

# SCALE-LES USERS GUIDE

Version 1.1.0

Team SCALE  
UGC working group

平成 28 年 4 月 15 日

# 目次

<b>第 1 章 Overview</b>	<b>3</b>
1.1 はじめに	3
1.1.1 SCALE の特徴	3
1.1.2 SCALE-LES モデルの構成	3
1.1.3 SCALE のライブラリとモデル	5
1.2 表記上の注意	6
<b>第 2 章 Installation</b>	<b>8</b>
2.1 必要なシステム環境	8
2.1.1 本書で想定する環境	9
2.2 ライブラリ環境のインストール	10
2.3 SCALE のコンパイル	10
2.4 後処理ツールのコンパイル	12
<b>第 3 章 SCALE-LES Tutorial: Ideal case</b>	<b>14</b>
3.1 内容の説明	14
3.2 実行方法	14
3.2.1 下準備	14
3.2.2 初期値作成	15
3.2.3 モデル本体の実行	16
3.2.4 後処理と描画	17
3.3 MPI プロセス数の変更	18
<b>第 4 章 SCALE-LES Tutorial: Real case</b>	<b>21</b>
4.1 境界データの準備	22
4.2 地形・土地利用データの作成 : pp	24
4.3 初期値・境界値データの作成 : init	25
4.4 時間積分を行う : run	27
4.5 結果を描画する	31
<b>第 5 章 SCALE-LES Advance use</b>	<b>34</b>
5.1 実験設定の詳細	34
5.1.1 格子間隔	34
5.1.2 計算領域の大きさ	36
5.1.3 地図投影法と計算領域の位置	39
5.1.4 側面境界条件	41
5.1.5 積分時間と積分時間間隔の設定	42
5.2 物理モデルの利用	43

5.2.1	雲微物理スキームの設定 . . . . .	43
5.2.2	乱流スキームの設定 . . . . .	44
5.2.3	放射過程の設定 . . . . .	44
5.2.4	陸面・海洋上の地表面フラックスモデルの設定 . . . . .	45
5.2.5	都市モデルの設定 . . . . .	45
5.3	任意のデータを SCALE で使用する . . . . .	45
5.3.1	初期値・境界値データ . . . . .	45
5.4	ドメインネスティング実験 . . . . .	46
5.4.1	オフライン・ネスティング実験の方法 . . . . .	46
5.4.2	オンライン・ネスティング実験の方法 . . . . .	50
5.4.3	子ドメインにおける地形の取り扱い . . . . .	54
5.5	複数の実験を一括実行する：バルクジョブ機能 . . . . .	56
5.6	出力変数の追加・変更 . . . . .	59
5.7	Postprocess : netcdf2grads . . . . .	61
5.7.1	スーパーコンピュータ「京」での実行方法 . . . . .	64
5.7.2	バルクジョブ対応版の使用法 . . . . .	65
5.7.3	旧型 netcdf2grads の使用法 . . . . .	66
<b>付録 A ライブラリ環境のインストール</b>		<b>70</b>
A.1	インストール方法 (Linux - CentOS 6.6 編) . . . . .	71
A.2	インストール方法 (Linux - CentOS 7.1 編) . . . . .	73
A.3	インストール方法 (Linux - openSUSE 13.2 編) . . . . .	75
A.4	インストール方法 (Mac OS X 編) . . . . .	77
A.5	インストール方法 (スーパーコンピュータ「京」編) . . . . .	79
A.6	描画ツールのインストール . . . . .	79
<b>付録 B Namelist in run.conf</b>		<b>82</b>
<b>付録 C Variables for history output</b>		<b>94</b>
<b>付録 D SCALE に実装されている雲モデルの概要とそれぞれの選択方法</b>		<b>106</b>
D.1	雲微物理スキームの選択方法 . . . . .	107
D.2	実行方法 . . . . .	108
D.3	各雲物理スキームにのオプションと Namelist . . . . .	108

# 第1章 Overview

## 1.1 はじめに

本書は初めて SCALE モデルを利用するユーザー向けに、SCALE-LES モデルについて解説したユーザーズガイドです。本書は SCALE version 0.2.6 (SCALE-LES version 4.2.6) に対応した説明になっています。第1章で SCALE の概要について説明します。第2章で必要な環境、およびインストール方法を説明します。つづいて、第3章では理想実験、第4章では現実大気実験の簡単な例を挙げて実行方法を説明します。第2章から第4章まではひとつながりのチュートリアルになっており、前章の経験や結果に基づいて説明する部分がありますので注意してください。第5章ではチュートリアルでは説明しなかった実行方法の詳細、機能やツールについて説明します。

本書中の不明点やお気づきの点がございましたら、SCALE user's メーリングリスト `scale-user@riken.jp` までご連絡ください。

### 1.1.1 SCALE の特徴

SCALE は計算機を用いて気象・気候科学の問題にアプローチする際に、ユーザーが研究を進めやすいように事前処理から数値モデル計算、解析に至るまですべての過程を網羅する計算ライブラリの開発を目指したソフトウェアであり、SCALE は下記に挙げるような特徴を持つ。

- SCALE は、“Scalable Computing for Advanced Library and Environment” の略である。
- SCALE は、フリーソフトウェアである。「BSD-2 ライセンス」においてフリーソフトウェアとして提供されており、商用、非商用に関わらず自由な利用・改変が可能なソフトウェアである。
- SCALE には、SCALE-LES モデル、SCALE-GM といった組み上げ済みの数値モデルが含まれている。
- SCALE には、次節で説明する様々なコンポーネントが導入されており、必要に応じて取り替えることが可能性である。

ライセンスの詳細は、トップディレクトリ直下の“scale/LICENSE”のファイルに記述されている。SCALE の使用前に一読しておくこと。また SCALE の Web ページにも説明が記載されているので必要に応じて参照すること (<http://scale.aics.riken.jp/>)

### 1.1.2 SCALE-LES モデルの構成

現在、SCALE-LES モデルとして組み込まれているコンポーネントは下記のものである。詳細なモデル構成や差分化手法については、Team SCALE (2015)、Sato Y. and Tomita (2015a)、および Nishizawa S. and Tomita (2015) を参照されたい。

フレームワーク関係

- Cartesian グリッドシステム (Arakawa C-grid)
- Cartesian ベース MPI 通信
- Map-projection & Map-factors
- Nesting システム (1-way : 親 子のみ)
  - Option: オフライン実行とオンライン実行を選択可能
- 複数事例一括実行 システム (バルクジョブシステム)
- gtool ベース netcdf4 ファイル I/O
- 外部データ読み込みによる初期値・境界値データ作成
  - Option: WRF-ARW、NICAM、その他 GrADS フォーマットでの入力に対応

#### 力学コア関係

- 方程式系: 3次元完全圧縮流体方程式
- 数値解法: 陽解法と陰解法の両方を実装済み
  - Option: HE-VE、HE-VI、HI-VI スキームから選択可能
- 空間差分: 4次中央差分
- 時間差分: 3段ルンゲクッタスキーム (2次)
- 非負保証: FCT スキーム
- 数値フィルター: 4次 Hyper diffusion
- 地形: Terrain-following スキーム

#### 物理過程

- 乱流過程: 複数から選択可能
  - Smagorinsky-Lilly type のサブグリッドモデル (Smagorinsky (1963); Lilly (1962))
  - MYNN2.5 境界層モデル (Mellor and Yamada (1982), Nakanishi and Niino (2004))
- 雲微物理: 複数から選択可能
  - Kessler type バルクモデル (Kessler (1969))
  - 6-class single moment バルクモデル (Tomita (2008))
  - 6-class double moment バルクモデル (Seiki and Nakajima (2014))
  - ピン法雲モデル (Kentaro et al. (2010)) から選択可能
- 放射過程: MSTRN (Sekiguchi and Nakajima (2008))
- 地表面モデル

- 陸面モデル: バケツモデル (バルク交換係数は Beljaars and Holtslag (1991); Wilson (2001))
- 海洋モデル: (スラブモデル)
- 都市モデル: 単層キャノピーモデル (Kusaka et al. (2001))

上記に加えて、SCALE-LES モデル本体のドライバー、現実事例の計算に必要な地形や土地利用データを作成するツール、理想実験の初期値・境界値、もしくは外部モデルから初期値・境界値を作成するツールが提供されている。

次節で SCALE ライブラリの思想とモデルの関係について説明するが、SCALE-LES モデルの実行とは直接関係ないため、必要なければ読み飛ばしても構わない。

### 1.1.3 SCALE のライブラリとモデル

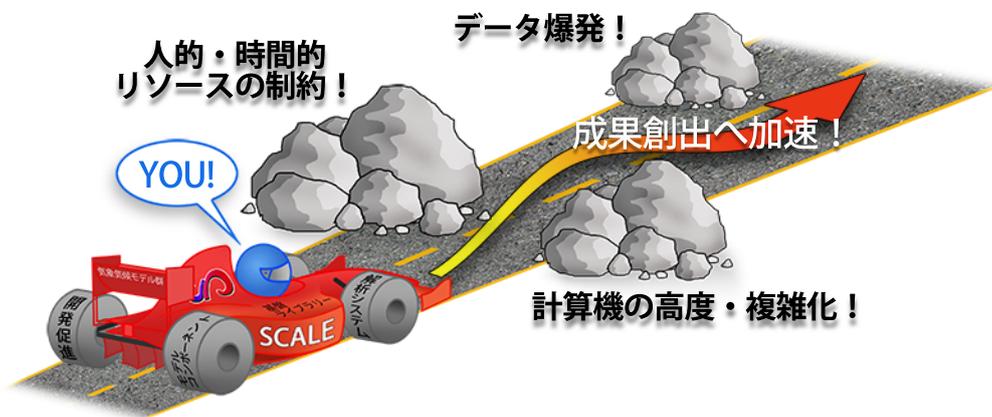


図 1.1: SCALE ライブラリのねらい

SCALE は、理化学研究所 計算科学研究機構 (RIKEN AICS) を中心に開発が進められている気象・気候科学向けのライブラリである。図 1.1 は SCALE のねらいを表したポンチ絵である。この図に示されるように、SCALE プロジェクトは、近い将来もしくは現在すでに研究の足かせとなっている諸問題を解決する、もしくは解決の手助けをすることを目指している。SCALE は次世代のスーパーコンピュータから汎用計算機に至るまで広く用いられる事を念頭において開発されており、気候・気象科学を専門とする科学者と計算機科学を専門とする科学者が共同で開発を行っている (Abe and Sato (2013))。そのため、スーパーコンピュータ「京」や Fujitsu FX10 等のスーパーコンピュータ上でも、ユーザーがチューニングすることなく高い計算効率で実行可能なようになっている。

SCALE ライブラリは、ユーザーにスーパーコンピュータ「京」をはじめとする並列計算機に対してチューニングされたモデルコンポーネントや解析システム、テストセット、そしてこれらに対する知見といった「開発環境」を提供する。

提供された開発環境利用の例として数値モデルの構築が挙げられる。その一例として、SCALE ライブラリを用いて構築した数値モデル、SCALE-LES モデルが本ライブラリセットに含まれている (図 1.2)。格子系、力学コア、そして物理過程といったモデル構築に必要な基本的コンポーネントは SCALE ライブラリが提供する。そのため、「SCALE-LES モデル」として新たに用意されたのは、これらのコンポーネントを動かすためのモデルドライバーや変数セットである。この SCALE-LES モデルは研究に利用できるものである、すでに層積雲の遷移過程を調べる研究などの実績があ

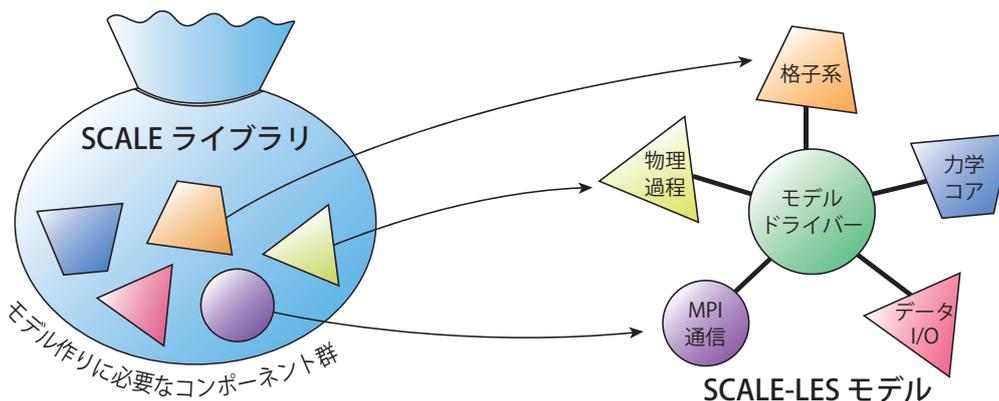


図 1.2: SCALE ライブラリと SCALE-LES モデルの関係

る (Sato Y. and Tomita (2014)、Sato Y. and Tomita (2015b))。ユーザーは、SCALE ライブラリを用いて SCALE-LES モデルと同様な方法によってユーザーは新たなモデルを作ることができる。

## 1.2 表記上の注意

本書中のシェルコマンド等は `bash` を想定して記述する。異なる環境下では適宜読み替えて対応すること。また、本書内では特に断りがない限り、下記の表記法に従うものとする。

コマンドラインのシンボル (`$`, `#`) は、コマンドの実行を示す。下記の表記の違いは、プログラムを実行する権限の違いを示している。

```
#      <- root 権限で実行するコマンド
$      <- ユーザ権限で実行するコマンド
```

権限の一時的な切り替えには `su` コマンドを用いる。{User\_Name} は実際のユーザ名に読み替えること。

```
$ su {User_Name}  <- {User_Name}のユーザー名でログイン
$ exit           <- {User_Name}のユーザー名でログインを終了
$ su -          <- root 権限に変更
#
```

コマンドオプションにハイフンを用いると、そのユーザでのログインを行う。用いない場合、権限のみの変更となる。またユーザ名を省略すると `root` でのログインを試す。ユーザの一時切り替えを終わるには、`exit` コマンドを用いる。文章表記のうち、ダブルスラッシュ (`//`) で始まる行は解説のためのもので、実際に記述する必要はない。

下記のように四角い囲みで区切られた部分は、コマンドラインのメッセージ部分であることを意味する。

```
----- コマンドラインのメッセージ
----- コマンドラインのメッセージ
----- コマンドラインのメッセージ
```

また、下記のように丸い囲みで区切られた部分は、エディタでファイルを編集する記述内容を表す、もしくはファイル内の記述を参照している部分である。

```
----- ファイル中の記述  
----- ファイル中の記述  
----- ファイル中の記述
```

\$ vi

上記のコマンドは、vi (汎用テキストエディタ) プログラムを実行する。他に、gedit、emacs などのテキストエディタがある。マニュアルでは vi コマンドの実行指示があるが、各自の使いやすいエディタへ適宜読み替えること。

## 第2章 Installation

本章では、SCALE-LES の使用方法と、SCALE チュートリアルを実行するための環境の準備について説明する。

### 2.1 必要なシステム環境

ここでは、SCALE のインストール、および実行に必要な環境について説明する。以下に説明するもの以外に、SCALE は、理化学研究所 計算科学研究機構のスーパーコンピュータ「京」、及び Fujitsu PRIME-HPC FX10 の環境に対応している。

必須のシステム環境

- システムハードウェア構成
  - CPU : 理想実験は物理コアが 2 コア以上、現実大気実験は 4 コア以上を搭載したシステムが望ましい。
  - Memory : 理想実験は 512MB 以上、現実大気実験は 2GB 以上のメモリ容量が搭載されたシステムが必要となる。
  - HDD : 実験内容によって変化するが、Tutorial の現実大気実験を実行するには約 3GB のディスク空き容量が必要となる。
- システムソフトウェア構成
  - OS : Linux OS、Mac OS X  
対応確認済み OS については、表 2.1 を参照のこと。
  - コンパイラ : C コンパイラ、Fortran  
Fortran コンパイラは Fortran2003 をサポートするコンパイラを必要とする。対応確認済みコンパイラについては、表 2.2 を参照のこと。
  - MPI ライブラリ : MPI1.0/2.0 に対応する MPI ライブラリを必要とする。対応確認済み MPI ライブラリについては、表 2.3 を参照のこと。
  - ファイル I/O ライブラリ : gzip、HDF5、NetCDF4 を必要とする。  
HDF5/NetCDF4 の代わりに、NetCDF3 でも動作するが、この場合は SCALE が提供する全ての機能が有効にならない可能性がある。

あると便利なシステム環境

- データ変換ツール : wgrib、wgrib2 や NCL 等があれば、SCALE で直接サポートしないデータを外部入力データとして利用できる。現実大気実験の Tutorial では、実際に wgrib2 を利用する。

- 描画環境：GrADS、GPhys/Ruby-DCL、ncview などがあれば計算結果を描画することができる。
- 演算性能評価：演算性能評価として PAPI ライブラリを使用が可能である。

表 2.1: 対応確認済み OS (全て x86-64 の 64bit 版)

OS 名	確認済みバージョン	備考
CentOS	6.5、6.6、7.0、7.1	左記以外も正常動作する可能性が高い
openSUSE	13.2	
SUSE Enterprise Linux	11.1、11.3	11.1 では GNU コンパイラは使用不可
fedora	20	左記以外も正常動作する可能性が高い
Vine Linux	6.2、6.3	
Mac OS X	10.10 Yosemite	

表 2.2: 対応確認済みコンパイラ

コンパイラ名	確認済みバージョン	備考
GNU (gcc/gfortran)	4.4.x、4.8.x	4.4.x ではコンパイル時に Warning が出ることがある
Intel (icc/ifort)	13.0.1、14.0.2、15.0.0	2013 年以降のバージョンであれば左記以外も正常動作する可能性が高い
PGI (gcc/pgfortran)	14.3	

表 2.3: 対応確認済み MPI ライブラリ

MPI ライブラリ名	確認済みバージョン	備考
openMPI	1.8.5	左記以外も正常動作する可能性が高い
Intel MPI	4.0 Update 3、4.1.0、5.0	2013 年以降のバージョンであれば左記以外も正常動作する可能性が高い
SGI MPT	2.05、2.09	Intel Compiler との組み合わせで確認済み

### 2.1.1 本書で想定する環境

本書の以降すべての説明において、下記の環境であることを前提として説明をすすめる。

- CPU：第 3 章は物理コアが 2 コア、第 4 章は 4 コアを搭載
- Memory：第 3 章は 512MB、第 4 章は 2GB
- OS：Linux OS x86-64 (CentOS 6.6、CentOS 7.1、openSUSE 13.2 のいずれか)
- コンパイラ：GNU コンパイラ (gcc/gfortran)
- MPI ライブラリ：openMPI (リポジトリ経由でのインストール)

## 2.2 ライブラリ環境のインストール

実行環境に合わせて、SCALE ライブラリのインストールに必要な環境、ライブラリのインストールを行う。詳細は、Appendix A 章を参照のこと。第 3 章、第 4 章のチュートリアルは、それらのライブラリ環境がインストールされていることを想定して進める。

描画ツールは何を使っても良いが、対応を確認している描画ツールについての詳細とインストール方法についての情報は Appendix A.6 を参照のこと。第 3 章および第 4 章のチュートリアルでは GrADS を使って描画する方法を紹介する。その他に Gphys を使って、結果のクイックビューを行う方法については、4.5 節にて説明する。

## 2.3 SCALE のコンパイル

以下の説明で使用した環境は次のとおりである。

- CPU: Intel Core i5 2410M 2 コア/4 スレッド
- Memory: DDR3-1333 4GB
- OS: CentOS 6.6 x86-64, CentOS 7.1 x86-64, openSUSE 13.2 x86-64

### ソースコードの入手

安定版ソースコードは、

<http://scale.aics.riken.jp/ja/download/index.html>

よりダウンロードできる。ソースコードの tarball ファイルを展開すると scale/ というディレクトリができる。{ver} には、“v020” 等のバージョン番号が入る。

```
$ tar -zxvf SCALE_{ver}.tar.gz
$ ls
scale/
```

### Makedef ファイルと環境変数の設定

SCALE はコンパイルするとき、独自の環境変数”SCALE\_SYS”で与えられた名称に対応した、Makedef ファイルを使用してコンパイルが行われる。Makdef ファイルは、scale/sysdep/内いくつかの計算機環境に対応するファイル (Makedef.\*\*\* ) が準備されており、これらの中から自分の環境にあったものを設定する。本書で対応を確認している環境について対応する Makedef ファイルを表 2.4 にまとめた。

ここでは、先述のとおり Linux OS、GNU コンパイラ、および openMPI を使用することを想定しているため、“Makedef.Linux64-gnu-ompi”が対応するファイルとなる。下記のコマンドによって環境変数の値を設定する。

```
$ export SCALE_SYS="Linux64-gnu-ompi"
```

---

<sup>1</sup>京や FX10 では、専用にチューニングされた富士通コンパイラ、及び MPI ライブラリを使用する。また、クロスコンパイルを行うので “fccpx/frtpx” や “mpicpx/mpifrtpx” といったコンパイラを用いる。

表 2.4: 環境例と対応する Makedef ファイル

OS/計算機	コンパイラ	MPI	Makedef ファイル
Linux OS x86-64	gcc/gfortran	openMPI	Makedef.Linux64-gnu-mpi
	icc/ifort	intelMPI	Makedef.Linux64-intel-mpi
	icc/ifort	SGI-MPT	Makedef.Linux64-intel-mpt
Mac OS X	gcc/gfortran	openMPI	Makedef.MacOSX-gnu-mpi
スーパーコンピュータ「京」	original <sup>1</sup>	original <sup>1</sup>	Makedef.K
Fujitsu PRIME-HPC FX10	original <sup>1</sup>	original <sup>1</sup>	Makedef.FX10

自分の環境に合致するものがなければ、既存ファイルをベースにして作成する。実行環境がいつも同じに決まっている場合は、常に同じ Makedef ファイルを使用するために、環境変数設定を `.bashrc` などの環境設定ファイルに記述しておくことと便利である。

チュートリアル通り進めている場合は、Appendix A 章で説明する以外に PATH の設定をする必要はないが、環境が異なる場合には上述に加えて次の設定が必要になる。HDF5 と NetCDF4 について、下記のように PATH を設定する。例えば、Intel コンパイラを利用して自分で `/opt` 下に HDF5 と NetCDF4 をインストールした場合を想定して、HDF5 は `/opt/hdf5`、`netcdf4` は `/opt/netcdf` にそれぞれインストールされている場合の例を示す。

```
$ export HDF5="/opt/hdf5"
$ export NETCDF4="/opt/netcdf"
```

Intel コンパイラ等を利用して自分で HDF5 をインストールした場合は、ライブラリのインストールパスが異なるので、`sysdep/Makdef.Linux64-intel-mpi` ファイルの記述を変更する必要がある。

```

  ~ ~ 中略 ~ ~
##### NetCDF library
NETCDF_INCLUDE ?= -I$(NETCDF4)/include
NETCDF_LIBS ?= -L$(NETCDF4)/lib -L$(HDF5)/lib
               -lnetcdff -lnetcdf -lhdf5_hl -lhdf5 -lm -lz
  ~ ~ 中略 ~ ~

```

上記のように、`-L$(HDF5)/lib64` の記述を `-L$(HDF5)/lib` に変更する。

## コンパイル

チュートリアル用のディレクトリに移動して、`make` コマンドによってコンパイルを行う。

```
$ cd scale/scale-les/test/tutorial/bin
$ make -j 4
```

`make` のあとの `"-j 4"` は、コンパイル時の並列数を示しており、例では 4 並列コンパイルを行うことを指示している。実行環境によっては並列数を増やすこともできる。この `make` によって SCALE ライブラリ、および SCALE-LES モデルのコンパイルが行われ、結果として

```
scale-les scale-les_init scale-les_pp
```

の 3 つの実行ファイルが生成されていればコンパイルは成功である。

下記のコマンドで作成された実行バイナリを消去できる。

```
$ make clean
```

ただし、この場合はコンパイルされたライブラリは消去されないため、全てのコンパイル済みファイルを消去したい場合は、

```
$ make allclean
```

とする。コンパイラを変更したり、コンパイルオプションを変更して再コンパイルする場合は、"allclean"を実行すること。

注意点

- SCALE ライブラリは、scale の TOP ディレクトリ直下の scale/scalelib/というディレクトリ内でコンパイルとアーカイブが行われ、".lib"という名前の隠しディレクトリとして bin/ディレクトリ内へコピーされる。
- Debug モードでコンパイルしたい場合は、"make -j 4 DEBUG=T"としてコンパイルする。
- 細かくコンパイルオプションを変更したい場合は、Makedef.\*\*\*のファイルを編集する。

## 2.4 後処理ツールのコンパイル

第 3 章、第 4 章のチュートリアルでは、GrADS を使用して図化する方法を紹介している。このとき、SCALE-LES の分割出力ファイル (history.\*\*\*\*\*.nc) を結合し、GrADS で直接読み込めるデータ形式へ変換する後処理ツール「net2g」を使用する。ここでは、この net2g のコンパイル方法について説明する。

net2g は SCALE 本体から独立したツールになっている (ただし Makedef ファイルを除く) ため、任意の場所へコピーしてコンパイルすることができるが、コンパイルには NetCDF ライブラリが必要であり、また並列実行するためには MPI ライブラリが必要である。従って、以降はこれらのライブラリがインストールされている環境であることを想定して進める。

まず、Makedef ファイル設定のために環境変数を設定する。SCALE 本体のコンパイル時と同じように、使用する環境にあった Makedef ファイルを選択し、下記のコマンドにて設定する。

```
$ export SCALE_SYS="Linux64-gnu-mpi"
```

その後、net2g のディレクトリに移動して、make コマンドによってコンパイルを行う。MPI ライブラリを用いた並列実行を行うためのバイナリは、下記のコマンドによって生成される。

```
$ cd scale/scale-les/util/netcdf2grads_h  
$ make -j 2
```

MPI ライブラリが無い場合など、逐次実行バイナリを生成するためには、

```
$ make -j 2 NOMPI=T
```

としてコンパイルを行う。結果として、”net2g”という名前の実行ファイルが生成されていればコンパイルは成功である。

下記のコマンドで作成された実行バイナリを消去できる。

```
$ make clean
```

## 第3章 SCALE-LES Tutorial: Ideal case

### 3.1 内容の説明

本章では、チュートリアルにおける1つめの実験として、SCALEを使った理想実験 (Ideal case) の実行方法を説明する。簡単な実験であるが、第2章で実行したSCALEのコンパイルが正常に完了しているかどうかのチェックも含めてぜひ実施してもらいたい。

ここでは、SCALEのコンパイルが正常に終了し、

```
scale-les/test/tutorial/bin
```

に `scale-les`、および `scale-les_init` が生成されており、

```
scale-les/util/netcdf2grads_h
```

に `net2g` が生成されているものとして説明を行う。これらに加えて、本章のチュートリアルでは、描画ツールとして GrADS を使用する。GrADS の詳細やインストール方法については、Appendix A.6 節を参照のこと。

ここで実行する理想実験は、「スコールライン」と呼ばれる積乱雲群を発生させる実験である。実験設定の概要を表 3.1 に示す。この実験は、積乱雲が発生する場合の典型的な大気の成層構造を表現した鉛直プロファイルを与え、対流圏下層に置いた初期擾乱から積乱雲が発達する様子を準2次元モデル実験する内容となっている。

このチュートリアルを実行するには、最低でも2コア/4スレッドの演算コアを持つCPU、512MB以上のメモリを搭載した計算機が必要である。本節の説明で使用した環境は次のとおりである。

- CPU: Intel Core i5 2410M 2.3GHz 2コア/4スレッド
- Memory: DDR3-1333 4GB
- OS: CentOS 6.6 x86-64, CentOS 7.1 x86-64, openSUSE 13.2 x86-64

### 3.2 実行方法

実行の流れとしては、下準備、初期値の作成、モデル本体の実行、後処理、そして描画といった順番で作業を進める。

#### 3.2.1 下準備

チュートリアル理想実験は、`scale-les/test/tutorial/ideal` のディレクトリにて実行するので、まずこのディレクトリに移動する。

```
$ cd scale-les/test/tutorial/ideal
```

表 3.1: チュートリアル理想実験の実験設定

項目	設定内容	備考
水平格子間隔	東西：500 m、南北：1000 m	東西-鉛直の面を切り取った準 2 次元実験である
水平格子点数	東西：40、南北：2	東西-鉛直の面を切り取った準 2 次元実験である
鉛直層数	97 層 (トップ：20 km)	下層ほど細かい層間隔をとったストレッチ設定である
側面境界条件	周期境界	東西、南北とも
積分時間間隔	5 sec	雲微物理スキームは 10 sec 毎
積分期間	3,600 sec	720 steps
データ出力間隔	300 sec	
物理スキーム	雲微物理モデルのみ使用	6-class single moment bulk model (tomita 2006)
初期鉛直プロファイル	GCSS Case1 squall-line	風のプロファイルは、Ooyama (2001) に基づいた鉛直シアを与える
初期擾乱	ウォームバブル	水平半径 4 km、鉛直半径 3 km の大きさを持つ最大プラス 3K の強度のウォームバブルを置く

次に、このディレクトリに対して、前章までに作成した SCALE の実行バイナリの静的リンクを張る。

```
$ ln -s ../bin/scale-les      ./
$ ln -s ../bin/scale-les_init ./
```

“scale-les” はモデル本体、“scale-les\_init” は初期値・境界値作成ツールである。もし、ここで説明するディレクトリとは異なる場所で実行している場合は、リンクを張る時のディレクトリ指定に注意すること。

### 3.2.2 初期値作成

ここでは、“scale-les\_init” を実行して初期値を作成する。“scale-les\_init” を実行する際には config ファイルを与える。例えば、“init\_R20kmDX500m.conf” のファイルには、表 3.1 に対応した実験設定が書き込まれており、この config ファイルの指示に従って scale-les\_init は大気の成層構造を計算し、ウォームバブルを設置する。

SCALE の基本的な実行コマンドは下記のとおりである。

```
$ mpirun -n [プロセス数] [実行バイナリ名] [config ファイル]
```

[プロセス数]の部分にはMPI並列で使いたいプロセス数を記述する。[実行バイナリ]には、scale-les や scale-les\_init が入る。そして、実験設定を記述した config ファイルを [config ファイル] の部分に指定する。ここでは、**2つのMPIプロセス**を用いて実行する。以降、これを「2-MPI並列」のように表現する。従って、init\_R20kmDX500m.conf を config ファイルとして与えて、2-MPI並列で scale-les\_init を実行する場合のコマンドはつぎようになる。

```
$ mpirun -n 2 ./scale-les_init init_R20kmDX500m.conf
```

実行が成功した場合には、コマンドラインのメッセージは下記のように表示される。

```
*** Start Launch System for SCALE-LES
TOTAL BULK JOB NUMBER = 1
PROCESS NUM of EACH JOB = 2
TOTAL DOMAIN NUMBER = 1
Flag of ABORT ALL JOBS = F
*** a single communicator
*** a single communicator
```

この実行によって、

```
“init.LOG.pe000000”
```

```
“init_00000000000.000.pe000000.nc”
```

```
“init_00000000000.000.pe000000.nc”
```

の3つのファイルが、現在のディレクトリ下に作成されているはずである。“init.LOG.pe000000”には、コマンドラインには表示されない詳しい実行ログが記録されている。実行が正常に終了している場合、このLOGファイルの最後に

```
+++++++ Stop MPI
*** Broadcast STOP signal
*** MPI is peacefully finalized
```

と記述される。

そして、“init\_00000000000.000.pe000000.nc”と“init\_00000000000.000.pe000001.nc”の2つのファイルが初期値ファイルである。計算領域全体を2つのMPIプロセスで分割し担当するため、2つのファイルが生成される。もし、4-MPI並列で実行すれば、4つの初期値ファイルが生成される。これらのファイル名の末尾が“.nc”で終わるファイルはNetCDF形式のファイルであり、Gphys/Ruby-DCLやncviewといったツールで直接読むことができる。

### 3.2.3 モデル本体の実行

いよいよ、モデル本体を実行する。初期値作成のときと同じように2-MPI並列だが、しかしconfigファイルは実行用の“run\_R20kmDX500m.conf”を指定する。

```
$ mpirun -n 2 ./scale-les run_R20kmDX500m.conf
```

本書の必要要件にあった計算機であれば、2分程度で計算が終わる。この実行によって、

```
“LOG.pe000000”
```

```
“history.pe000000.nc”
```

```
“history.pe000000.nc”
```

```
“monitor.pe000000”
```

の4つのファイルが、現在のディレクトリ下に作成されているはずである。“LOG.pe000000”には、コマンドラインには表示されない詳しい実行ログが記録されている。実行が正常に終了している場合、このLOGファイルの最後に

```
+++++++ Stop MPI
*** Broadcast STOP signal
*** MPI is peacefully finalized
```

と記述される。

そして、“history.pe000000.nc”と“history.pe000001.nc”の2つのファイルが計算経過のデータが記録された history ファイルである。このファイルも NetCDF 形式のファイルであり、2-MPI 並列で実行したため、やはり2つのファイルが生成される。“monitor.pe000000”は、計算中にモニタリングしている物理変数の時間変化を記録したテキストファイルである。

### 3.2.4 後処理と描画

ここでは、計算結果を描画するための後処理について説明する。本書のチュートリアルでは、NetCDF 形式の分散ファイルを1つのファイルにまとめ、ユーザーが解析しやすい Direct-Access の単純バイナリ形式 (GrADS 形式) に変換する方法を説明する。Gphys/Ruby-DCL を使うと分割ファイルのまま直接描画することができるが、この方法については4.5節を参照してもらいたい。

まず、2.4節でコンパイルした後処理ツール“net2g”を、現在のディレクトリへリンクを張る。

```
$ ln -s ../../../../util/netcdf2grads_h/net2g ./
```

もし、ここで説明するディレクトリとは異なる場所で実行している場合は、リンクを張る時のディレクトリ指定に注意すること。

net2g も実行方法は基本的に SCALE 本体と同じである。

```
$ mpirun -n [プロセス数] ./net2g [config ファイル]
```

net2g 専用の“net2g.conf”を config ファイルとして与えて、つぎのように実行する。

```
$ mpirun -n 2 ./net2g net2g.conf
```

net2g の実行にあたっては、SCALE 本体の実行時に使用した MPI プロセス数と同じか、その約数のプロセス数を用いて実行しなければならない。HDD の読み書き速度に依存するが、本書の必要要件にあった計算機であれば2分程度で計算が終わる。この実行によって、

“QHYD\_d01z-3d.ctl”、“U\_d01z-3d.ctl”、“W\_d01z-3d.ctl”

“QHYD\_d01z-3d.grd”、“U\_d01z-3d.grd”、“W\_d01z-3d.grd”

の6つのファイルが、現在のディレクトリ下に作成される。

これらのファイルはそれぞれ、3次元変数、U (水平風東西成分)、W (鉛直風)、QHYD (全凝結物混合比) について、分割ファイルを1つにまとめ、Direct-Access の単純バイナリ形式 (GrADS 形式) に変換された grd ファイルと GrADS に読み込ませるための ctl ファイルである。従って、この ctl ファイルを GrADS に読み込ませれば直ちに計算結果の描画が可能である。図 3.1 は、積分開始 1200 秒後における、U-W と QHYD についての鉛直断面図である。

“net2g.conf”の下記の行を編集することによって、net2g を用いて他の様々な変数の変換を行うことができる。

```
&VARI  
VNAME = "U", "W", "QHYD"  
/
```

この“VNAME”の項目を例えば、“PT”, “RH”と変更して実行すれば温位と相対湿度の変数について変換する。どの変数が出力されているのかを調べるには、NetCDF の ncdump ツールなどを使えば簡単に調べられる。net2g の詳しい使用方法は、5.7 を参照してほしい。

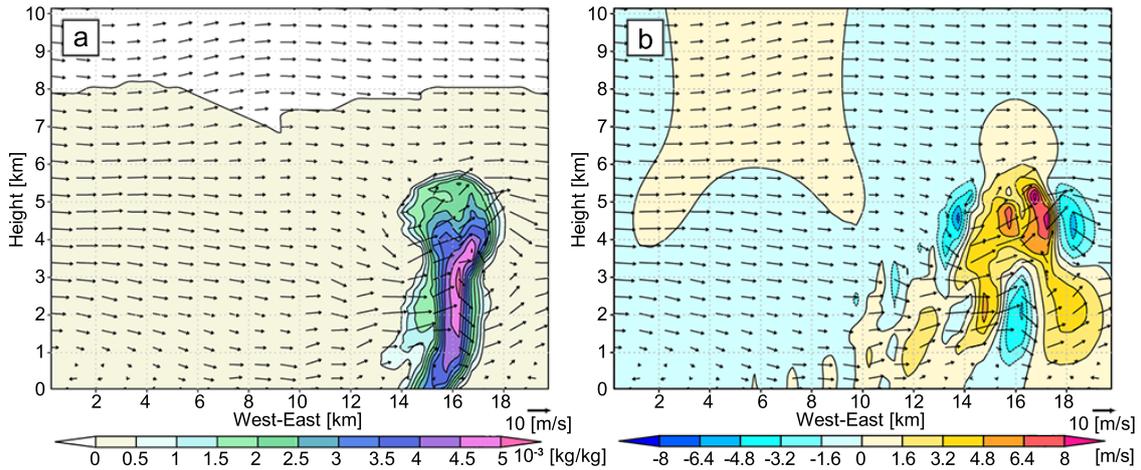


図 3.1: 積分開始後 1200 sec の  $Y=1$  km における東西-鉛直断面図 ; (a) のカラーシェードは全凝結物の混合比、(b) は鉛直速度をそれぞれ示す。ベクトルは東西-鉛直断面内の風の流れを表す。

### 3.3 MPI プロセス数の変更

今後、様々な実験を行う上で必須の設定変更である MPI プロセス数の変更方法について説明する。その他の設定方法については次節を参照して欲しい。

前節までに使用した `init_R20kmDX500m.conf` や `run_R20kmDX500m.conf` の `config` ファイルを編集することで、すべての実験設定の変更が行える。先に述べたとおり `SCALE` の入出力ファイルは、MPI プロセス毎に分割されている。そのため、MPI プロセス数を変更すると分割ファイル数も必ず変わることになる。従って、2-MPI 並列用に作成した初期値ファイルは、4-MPI 並列のモデル実行には使用できない。MPI プロセス数を変更するには、“`init_***.conf`”、“`run_***.conf`” の両方を編集・変更し、再度初期値作成から行わなければならない。

`init_R20kmDX500m.conf` を `vi` エディタ等で開くと、22 行目付近に“`PARAM_PRC`”という設定項目がある。チュートリアルでは 2-MPI 並列で実行するため、デフォルトでは下記に示すような設定が記述されている。“`PRC`”は“`Process`”を意味する。

```

&PARAM_PRC
  PRC_NUM_X = 2,
  PRC_NUM_Y = 1,
/

```

`SCALE` では水平 2 次元に領域を分割して並列計算する。“`PRC_NUM_X`”は X 方向（東西方向）の MPI 並列分割数、“`PRC_NUM_Y`”は Y 方向（南北方向）の MPI 並列分割数を指定する変数である。この `config` ファイルから、X 方向に 2 分割、Y 方向に 1 分割（分割なし）という設定であることがわかる。これはチュートリアルでは、Y 方向に均一な準 2 次元実験を行なっているため、X 方向にだけ MPI 並列している。全 MPI プロセス数は、 $PRC_{total} = PRC_X \times PRC_Y$  であるため、 $2 \times 1 = 2$  で全 MPI プロセス数は 2 プロセスである。次に格子点数の設定項目について説明する。

```
&PARAM_INDEX
KMAX = 97,
IMAX = 20,
JMAX = 2,
/
```

上記の“PARAM\_INDEX”の設定項目が格子点数を設定する項目である。KMAX、IMAX、JMAXは、鉛直層数、X方向の格子点数、Y方向の格子点数をそれぞれ意味する。この値は、MPIプロセス当たりの値であることに注意が必要である。つまり、計算領域全体での格子点数は、

$$\text{東西方向} : Grids_X = IMAX \times PRC_X$$

$$\text{南北方向} : Grids_Y = JMAX \times PRC_Y$$

と表現される。鉛直方向には分割しないため、KMAXがそのまま領域全体の鉛直層数を表す。従って、このチュートリアル全体の格子点数は、X方向（東西）に40点、Y方向（南北）に2点、そして鉛直に97層ということがわかる。

さて、上記の設定を変更して4-MPI並列で実行できるようにしてみる。注意する点は領域全体の格子点数を維持するように設定することである。今回は準2次元実験なので、X方向に4分割して4-MPI並列を達成する。この場合の設定方法は下記のとおりである。

```
&PARAM_INDEX
KMAX = 97,
IMAX = 10,
JMAX = 2,
/

&PARAM_PRC
PRC_NUM_X = 4,
PRC_NUM_Y = 1,
/
```

X方向に4分割を指定するため、PRC\_NUM\_X = 4と記述されている。そして、領域全体で40格子点とするために、IMAX = 10と記述されている。Y方向と鉛直方向には何も変更していない。**この変更を、init\_R20kmDX500m.confとrun\_R20kmDX500m.confの両方に施さなければならない。**そして、つぎのようにMPIコマンドに指定するプロセス数を“4”として、初期値作成、モデル実行の順で作業を進めれば、4-MPI並列で実行することができる。

```
$ mpirun -n 4 ./scale-les_init init_R20kmDX500m.conf
$ mpirun -n 4 ./scale-les run_R20kmDX500m.conf
```

計算領域（総演算量）を維持したままMPIプロセス数を2倍に増やすことによって、1つのMPIプロセスあたりの問題サイズ（演算量 per PRC）が1/2に減る。したがって、計算にかかる時間も理想的には半分になる<sup>1</sup>。実験機では、2-MPI並列のときチュートリアル時間積分に60 secかかっていたが、4-MPI並列にすることで同じ計算が32 secで終了できた。ここで説明したMPIプロセス数の変更を加えたサンプルファイルが、同じディレクトリ下の“sample”ディレクトリ内にinit\_R20kmDX500m.pe4.conf、run\_R20kmDX500m.pe4.confとして置いてあるので、うまく実行できない場合は参考にして欲しい。

<sup>1</sup>計算科学用語では、この変更、つまり総演算量一定でプロセスあたりの演算量を減らしていくことを“strong scaling”と呼ぶ。

この章の最後に

以上で、理想実験の最も簡単な実行方法についてのチュートリアルは終了である。ここでは、簡単な実行方法を MPI プロセス数の変更方法だけを説明したが、実際には解像度や計算領域を変更したり、放射過程や乱流過程といった他の物理過程を加えてみたり、雲微物理スキームを別のスキームに変更したりすることがあるだろう。これらの変更方法は、第 5 章 5.1 節に詳しく記載されているので適宜参考にして欲しい。このスコールラインの理想実験については、同じディレクトリ下の “sample” ディレクトリ内に、解像度設定、領域設定、そして使用する物理スキームについて変更を加えた config ファイルのサンプルが用意されているので、これらを参考にすれば、SCALE のシステムについて理解が深まることと思う。

また、SCALE には他にも理想実験セットが “scale-les/test/case” 以下に複数用意されているので、興味があれば他の理想実験にもチャレンジしてもよい。少々ディレクトリ構造がチュートリアルとは異なる部分もあるが、実行に関しては本章のチュートリアルと同じであるため、容易に実験できるだろう。

## 第4章 SCALE-LES Tutorial: Real case

このチュートリアルでは、図 4.2 に示した領域を対象とした現実大気実験を行う。現実大気実験用のチュートリアルは

```
scale/scale-les/test/tutorial/real/
```

の下で行う。以降の説明で $\{Tutrial\_DIR\}$ は、`scale/scale-les/test/tutorial/`を絶対 PATH で示すこととする。

SCALE-LES モデルの実行過程は、図 4.1 に示されるように

1. pp : 地形・海陸分布データの作成
2. init : 初期値・境界値データの作成
3. run : 時間積分を行う (モデル本体の実行)

といった手順で実行する。

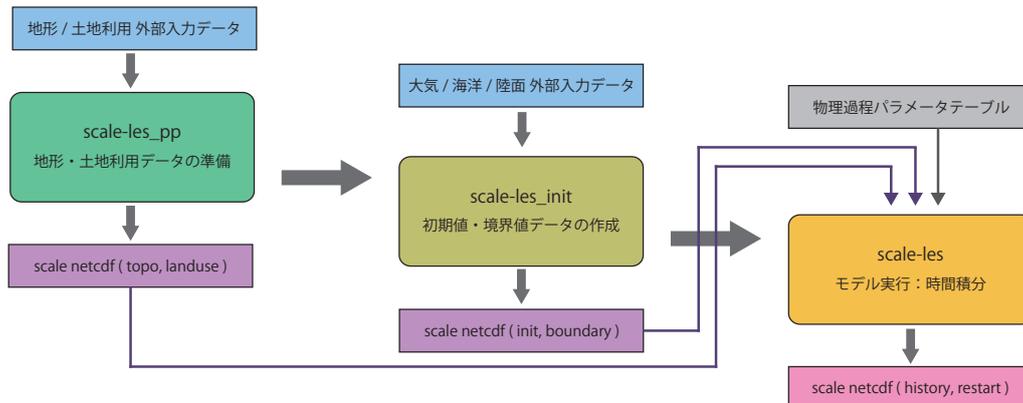


図 4.1: SCALE-LES モデルの実行過程

計算領域 (ドメイン) の設定は Table 4.1 のようになっている。チュートリアルは、SCALE の使い方を学ぶことが目的であり、短い時間で実行可能な設定にしている。領域モデルの実験設定として必ずしも適切な設定を選択しているとは限らないので (例えば、15km の水平解像度で積雲パラメタリゼーションなし) ご留意頂きたい。

このチュートリアルを実行するには、下記の条件を満たす計算機環境が必要である。

- CPU: 2 コア/4 スレッド以上の演算コアを持つ CPU (4 コア以上を推奨)
- Memory 容量: 4GB 以上をプログラムに割当可能 (8GB 以上を搭載した計算機を推奨)
- HDD 空き容量: 7GB 以上の空き容量

## Topography

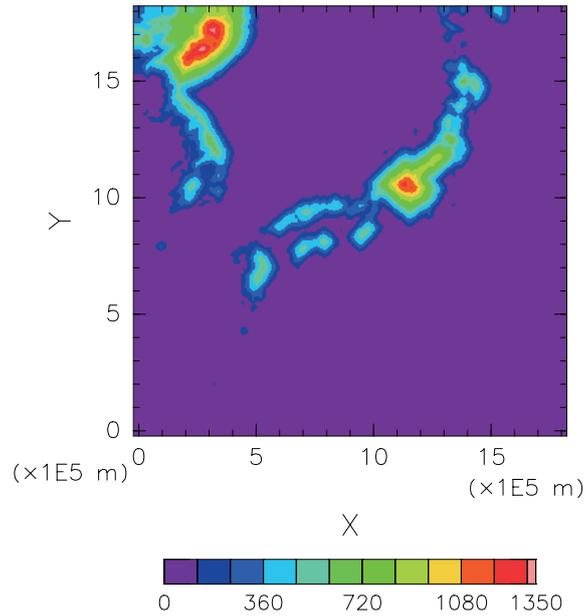


図 4.2: 計算領域 . カラーシェードは地形の標高を示す .

本節の説明で使用した環境は次のとおりである。

- CPU: Intel Xeon E5-2620 2.0GHz 6 コア (ただし 4 コアのみ使用)
- Memory: DDR3-1333 (4 channel, 32GB 搭載)
- OS: CentOS 6.6 x86-64, CentOS 7.1 x86-64, openSUSE 13.2 x86-64

表 4.1: 実験設定の概略

項目	設定
MPI プロセス分割 (東西 x 南北)	2 x 2 (合計 4 プロセス)
水平格子数 (東西 x 南北)	60 格子点 x 60 格子点
鉛直層数	36 層
水平格子間隔	$dx = dy = 15\text{km}$
積分期間	2014 年 8 月 10 日 00UTC ~ 12UTC (12 時間積分)
時間ステップ間隔	60 sec (720 steps)

### 4.1 境界データの準備

現実大気実験のシミュレーションを行う場合、SCALE 本体に加えて境界値データが必要になる。境界値データとしては下記が必要である。青字は必須の変数、その他は任意である。

- 地形データ (SCALE モデルの地形を準備する)

- 標高データ
- 土地利用データ
- 初期値境界値データ
  - 親モデルの緯度・経度
  - 3次元大気データ
    - \* 東西風速, 南北風速, 気温, 比湿 (相対湿度), 気圧, ジオポテンシャル高度
  - 2次元大気データ
    - \* 海面更正気圧, 地上気圧, 10m 東西風速, 10m 南北風速, 2m 気温, 2m 比湿 (相対湿度)
  - 2次元陸面データ
    - \* 親モデルの海陸マップ
    - \* 地表面温度
    - \* 土壌データの深さ情報, 土壌温度, 土壌水分 (体積含水率 or 飽和度)
  - 2次元海面データ 海面水温

#### 地形データと土地利用データ

標高データと土地利用データは実験設定に従って、SCALE のそれぞれの格子点における地形、海陸分布、土地利用を作成するために使用する。ユーザーが世界の任意の地域を対象とした計算ができるよう、フォーマット変換済みの標高データ USGS(U.S. Geological Survey) の GTOPO30 と、土地利用データ GLCCv2、を提供している。

##### 1. データのダウンロード

SCALE 用の地形・土地利用のデータを

[http://scale.aics.riken.jp/download/scale\\_database.tar.gz](http://scale.aics.riken.jp/download/scale_database.tar.gz)

より入手し、任意のディレクトリに展開しておく。

```
$ tar -zxvf scale_database.tar.gz
```

展開したディレクトリには、地形データと土地利用データが格納されている。

```
scale_database/topo/    <- 地形データ
scale_database/landuse/ <- 土地利用データ
```

##### 2. パスの設定

make を使った job script を使用する場合には、展開先のディレクトリを SCALE\_DB という環境変数に設定しておくことと便利である (以後、`${SCALE_DB}` と表記)。

```
$ export SCALE_DB=${path_to_directory_of_scale_database}/scale_database
```

## 大気・陸面・海面水温データ

初期値境界値データは 4-byte バイナリー (grads format) に変換すれば、任意のデータを読み込むことが可能である。チュートリアルでは NCEP FNL(Final) Operational Global Analysis data を使用する。

### 1. データのダウンロード

NCAR のサイト <http://rda.ucar.edu/datasets/ds083.2/> から、2014 年 8 月 10 日の grib2 フォーマットのデータ

```
fnl_20140810_00_00.grib2
fnl_20140810_06_00.grib2
fnl_20140810_12_00.grib2
fnl_20140810_18_00.grib2
```

を `${Tutorial_DIR}/real/tools/` の下にダウンロードする。

### 2. データフォーマットを grib から binary に変換

SCALE は 4byte バイナリー (grads format) の境界値データを読み込むことができる。

`${Tutorial_DIR}/real/tools/` 中にある `convert_grib2grads.sh` を実行。ただし、あらかじめ `wgrib2` がインストールされている必要がある。

```
$ sh convert_grib2grads.sh
```

成功すれば、下記のファイルが作成される。

```
$ ls FNL_output/**
FNL_output/201408/FNLatm_2014081000.grd
FNL_output/201408/FNLatm_2014081006.grd
FNL_output/201408/FNLatm_2014081012.grd
FNL_output/201408/FNLatm_2014081018.grd
FNL_output/201408/FNLland_2014081000.grd
FNL_output/201408/FNLland_2014081006.grd
FNL_output/201408/FNLland_2014081012.grd
FNL_output/201408/FNLland_2014081018.grd
FNL_output/201408/FNLsfc_2014081000.grd
FNL_output/201408/FNLsfc_2014081006.grd
FNL_output/201408/FNLsfc_2014081012.grd
FNL_output/201408/FNLsfc_2014081018.grd
```

## 4.2 地形・土地利用データの作成 : pp

pp ディレクトリへ移動し、現実実験のための地形データ、土地利用データを作成する。

```
$ cd ${Tutorial_DIR}/real/pp
```

pp ディレクトリの中には、pp.conf という名前のコンフィグファイルが準備されている。ドメインの位置や格子点数など、実験設定に合わせて、適宜 pp.conf を編集する必要があるが、チュートリアルではすでに Table 4.1 の設定に従って編集済みの pp.conf が用意されているためそのまま利用する。pp.conf の設定の中で特に注意すべき項目は、PARAM\_CONVERT である。

```
&PARAM_CONVERT
  CONVERT_TOPO = .true.,
  CONVERT_LANDUSE = .true.,
/
```

上記のように CONVERT\_TOPO と CONVERT\_LANDUSE が .true. となっていることが、それぞれ地形と土地利用の処理を行うことを意味している。詳細なコンフィグファイルの内容については、Appendix B を参照されたい。

次に、コンパイル済みのバイナリと入力データを pp ディレクトリへリンクする。

```
$ ln -s ../../bin/scale-les_pp ./
$ ln -s ${SCALE_DB}/topo ./
$ ln -s ${SCALE_DB}/landuse ./
```

今回は、Table 4.1 に示されているように、4 つの MPI プロセスを使用する設定なので次のように実行する。

```
$ mpirun -n 4 ./scale-les_pp pp.conf
```

実行にはおおよそ 15 秒を要する。ジョブが正常に終了すれば、topo\_d01.pe#####.nc と landuse\_d01.pe#####.nc というファイルが MPI プロセス数だけ、つまり 4 つずつ生成される (#####には MPI プロセスの番号が入る)。それぞれ、ドメインの格子点に内挿された地形と土地利用の情報が入っている。実行時のログは、pp\_LOG\_d01.pe000000 に出力されるので内容を確かめておくこと。

## OPTION

gpview がインストールされている場合、次のコマンドによって、作成された地形と土地利用データが正しく作成されているかどうか確認することが出来る。正しく作成されていれば、図 4.2 と同様の図ができる。

```
$ gpview topo_d01.pe00000*@TOP0 --aspect=1 --nocont
$ gpview landuse_d01.pe00000*@FRAC_LAND --aspect=1 --nocont
```

### 4.3 初期値・境界値データの作成：init

init では、SCALE 計算に必要な初期値・境界値データを作成する。まず、init ディレクトリへ移動する。

```
$ cd ${Tutorial_DIR}/real/init
```

init ディレクトリの中には、init.conf という名前のコンフィグファイルが準備されている。pp.conf と同様に、実験設定に合わせて、この init.conf を書き換える必要があるが、チュートリアル用の

init.conf ファイルは Table4.1 の設定にすでに合わせてある。初期値・境界値データの作成には前節で作成した地形・土地利用データを利用する。これは、init.conf の中で、下記のように相対 PATH を用いて参照するように設定されている。

```
&PARAM_TOPO
  TOPO_IN_BASENAME = "../pp/topo_d01",
  /
&PARAM_LANDUSE
  LANDUSE_IN_BASENAME = "../pp/landuse_d01",
  /
```

その他に init.conf の設定の中で特に注意すべき項目は、PARAM\_MKINIT\_REAL である。

```
&PARAM_MKINIT_REAL
  BASENAME_BOUNDARY = "boundary_d01",      境界値データの出力名
  FILETYPE_ORG = "GrADS",
  NUMBER_OF_FILES = 3,                    読み込むファイルの数
  BOUNDARY_UPDATE_DT = 21600.DO,          入力データの時間間隔
  INTERP_SERVC_DIV_NUM = 20,              内挿計算用のチューニングパラメータ
  PARENT_MP_TYPE = 3,
  USE_FILE_DENSITY = .false.,             親モデルの density データを使うか
  USE_FILE_LANDWATER = .true.,            親モデルの土壌水分データを使うか
  INTRP_LAND_SFC_TEMP = "mask",           親モデルの欠測値処理方法
  INTRP_LAND_TEMP = "fill",
  INTRP_LAND_WATER = "fill",
  INTRP_OCEAN_SFC_TEMP = "mask",
  INTRP_OCEAN_TEMP = "mask",
  /
```

FILETYPE\_ORG は入力する気象場データのファイルフォーマットに関するパラメータを設定しており、ここでは grads 形式のデータを読み込むことを指定している。詳細なコンフィグファイルの内容については、Appendix B を参照されたい。

次に、コンパイル済みのパイナリを init ディレクトリへリンクする。

```
$ ln -s ../../bin/scale-les_init ./
```

4.1 節で作成した入力データに、init ディレクトリの中に準備されている "gradsinput-link\_FNL.sh" を用いてリンクをはる。

```
$ sh gradsinput-link_FNL.sh
```

下記の grads 形式のファイルにリンクが張れば成功。

```

./FNLatm_00000.grd' -> './tools/FNL_output/201408/FNLatm_2014081000.grd
./FNLatm_00001.grd' -> './tools/FNL_output/201408/FNLatm_2014081006.grd
./FNLatm_00002.grd' -> './tools/FNL_output/201408/FNLatm_2014081012.grd
./FNLatm_00003.grd' -> './tools/FNL_output/201408/FNLatm_2014081018.grd
./FNLsfc_00000.grd' -> './tools/FNL_output/201408/FNLsfc_2014081000.grd
./FNLsfc_00001.grd' -> './tools/FNL_output/201408/FNLsfc_2014081006.grd
./FNLsfc_00002.grd' -> './tools/FNL_output/201408/FNLsfc_2014081012.grd
./FNLsfc_00003.grd' -> './tools/FNL_output/201408/FNLsfc_2014081018.grd
./FNLland_00000.grd' -> './tools/FNL_output/201408/FNLland_2014081000.grd
./FNLland_00001.grd' -> './tools/FNL_output/201408/FNLland_2014081006.grd
./FNLland_00002.grd' -> './tools/FNL_output/201408/FNLland_2014081012.grd
./FNLland_00003.grd' -> './tools/FNL_output/201408/FNLland_2014081018.grd

```

次に、陸面の変数を用意するのに必要なパラメータファイルにリンクをはる。

```
$ ln -s ../../../../data/land/* ./ <- 陸面スキーム用のパラメータファイル
```

準備が整ったら、4つのMPIプロセスを使用してinitを実行する。

```
$ mpirun -n 4 ./scale-les_init init.conf
```

実行にはおおよそ45秒を要する。正常にジョブが終了すれば、boundary\_d01.pe#####.ncとinit\_d01\_00019094400.000.pe#####.ncというファイルがMPIプロセス数だけ、つまり4つずつ生成される(#####にはMPIプロセスの番号が入る)。それぞれ、境界値データと初期値データが入っており、境界値データには複数の時刻のデータが1つのファイルに含まれている。初期値ファイルの名前のうち"00019094400.000"の部分は、モデル内で算出された実験開始時刻を表している。処理内容のログとして、init\_LOG\_d01.pe000000という名前でもログファイルも出力されるので内容を確認しておくこと。

## OPTION

gpviewがインストールされている場合、作成された初期値と境界値が正しく作成されているかどうかを確認することが出来る。正しく作成されていれば、図4.3と同じように描かれる。

```
$ gpvect --scalar --slice z=1500 --nocont --aspect=1 --range=0.002:0.016 \
--xintv=10 --yintv=10 --unit_vect init_d01_00019094400.000.pe00*@QV \
init_d01_00019094400.000.pe00*@MOMX init_d01_00019094400.000.pe00*@MOMY
```

## 4.4 時間積分を行う：run

run.confの準備

ここではいよいよSCALE-LESモデルを実行する。まず、runディレクトリへ移動する。

```
$ cd ${Tutorial_DIR}/real/run
```

Water Vapor mixing ratio  
(momentum x, momentum y)

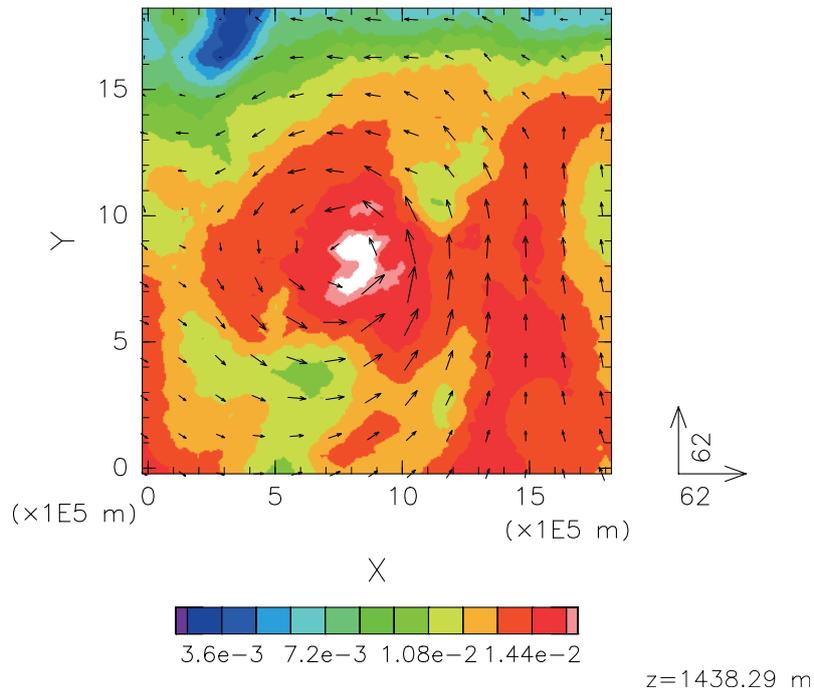


図 4.3: チュートリアル実験の高さ 1500m における初期場の様子 . カラーシェードは比湿 , ベクトルは水平運動量フラックスを表している .

run ディレクトリの中には、run.conf という名前のコンフィグファイルが準備されており、ドメインの位置や格子点数など、チュートリアル用の設定 ( Table4.1 ) に合わせてある。モデル本体の実行には事前に作成した地形・土地利用データや初期値・境界値データを利用する。これらのファイルの場所は、run.conf 内の TOPO\_IN\_BASENAME、LANDUSE\_IN\_BASENAME、RESTART\_IN\_BASENAME、および ATMOS\_BOUNDARY\_IN\_BASENAME で指定している。

```

&PARAM_TOPO
  TOPO_IN_BASENAME = "../pp/topo_d01",
/

&PARAM_LANDUSE
  LANDUSE_IN_BASENAME = "../pp/landuse_d01",
/

&PARAM_RESTART
  RESTART_OUTPUT = .false.,
  RESTART_IN_BASENAME = "../init/init_d01_00019094400.000",
/

&PARAM_ATMOS_BOUNDARY
  ATMOS_BOUNDARY_TYPE = "REAL",
  ATMOS_BOUNDARY_IN_BASENAME = "../init/boundary_d01",
  ATMOS_BOUNDARY_USE_VELZ = .true.,
  ATMOS_BOUNDARY_USE_QHYD = .false.,
  ATMOS_BOUNDARY_VALUE_VELZ = 0.0D0,
  ATMOS_BOUNDARY_UPDATE_DT = 21600.0D0,
/

```

run.conf の設定の中で時間積分に関する設定は、PARAM\_TIME の項目にある。

```

&PARAM_TIME
  TIME_STARTDATE = 2014, 8, 10, 0, 0, 0,      時間積分を開始する時刻
  TIME_STARTMS = 0.D0,
  TIME_DURATION = 12.0D0,                      積分期間
  TIME_DURATION_UNIT = "HOUR",                 TIME_DURATION の単位
  TIME_DT = 60.0D0,                            移流計算の時間ステップ
  TIME_DT_UNIT = "SEC",                        TIME_DT の単位
  TIME_DT_ATMOS_DYN = 15.0D0,                  移流計算以外の力学過程の時間ステップ
  TIME_DT_ATMOS_DYN_UNIT = "SEC",             TIME_DT_ATMOS_DYN の単位

  ~ ~ 中略 ~ ~

/

```

初期時刻 TIME\_STARTDATE は UTC で指定する。チュートリアルでは 2014 年 8 月 10 日 0 時 UTC に設定している。積分のための時間ステップは、上記の他、それぞれの物理スキーム毎に設定できるようになっている。

計算結果の出力に関する設定は PARAM\_HISTORY で行う。

```

&PARAM_HISTORY
HISTORY_DEFAULT_BASENAME = "history_d01",      出力するファイル名
HISTORY_DEFAULT_TINTERVAL = 1800.D0,          出力時間間隔
HISTORY_DEFAULT_TUNIT = "SEC",                出力時間間隔の単位
HISTORY_DEFAULT_TAVERAGE = .false.,
HISTORY_DEFAULT_DATATYPE = "REAL4",
HISTORY_DEFAULT_ZINTERP = .false.,           出力時に高さ面へ内挿するかどうか
HISTORY_OUTPUT_STEP0 = .true.,               初期時刻 (t=0) の値を出力するかどうか
/

```

上記の設定に従って、下記の HISTITEM に羅列された変数が出力される。HISTITEM ではオプション変数を加えることで、変数毎に、出力間隔を変更したり、平均値を出力したりすることも可能である。これらの説明は 5.6 を参照されたい。

```

&HISTITEM item="DENS" /      ! density (3D)
&HISTITEM item="MOMZ" /      ! vertical momentum (3D)
&HISTITEM item="MOMX" /      ! horizontal momentum-x (3D)
&HISTITEM item="MOMY" /      ! horizontal momentum-y (3D)
&HISTITEM item="RHOT" /      ! density * potential-temperature (3D)
&HISTITEM item="QV" /        ! mixing ratio for vapor (3D)
&HISTITEM item="QHYD" /      ! mixing ratio for hydrometeor (3D)
&HISTITEM item="T" /         ! temperature (3D)
&HISTITEM item="PRES" /      ! pressure (3D)
&HISTITEM item="U" /         ! horizontal wind component-x (3D)
&HISTITEM item="V" /         ! horizontal wind component-y (3D)
&HISTITEM item="W" /         ! vertical wind component (3D)
&HISTITEM item="PT" /        ! potential temperature (3D)
&HISTITEM item="RH" /        ! relative humidity (3D)
&HISTITEM item="PREC" /      ! precipitation (2D)
&HISTITEM item="OLR" /       ! out-going longwave radiation(2D)
&HISTITEM item="U10" /       ! horizontal wind component-x at 10m height(2D)
&HISTITEM item="V10" /       ! horizontal wind component-y at 10m height(2D)
&HISTITEM item="T2" /        ! temperature at 2m height (2D)
&HISTITEM item="Q2" /        ! mixing ratio for vapor at 2m height (2D)
&HISTITEM item="SFC_PRES" /   ! pressure at the bottom surface (2D)
&HISTITEM item="SFC_TEMP" /   ! temperature a the bottom surface (2D)
&HISTITEM item="LAND_SFC_TEMP" / ! temperature a the bottom surface for land model (2D)
&HISTITEM item="URBAN_SFC_TEMP"/ ! temperature a the bottom surface for urban model (2D)

```

その他に実験で使用される物理過程の設定は、PARAM\_TRACER ,PARAM\_ATMOS ,PARAM\_OCEAN ,PARAM\_LAND ,PARAM\_URBAN の項目に記述されているので、実行前にチェックすること。詳細なコンフィグファイルの内容については、Appendix B を参照されたい。

## 実行

コンパイル済みのバイナリを run ディレクトリへリンクする。

```
$ ln -s ../../bin/scale-les ./
```

陸面過程や放射過程のモデルを起動するためのパラメータファイルにリンクを張る。

```

$ ln -s ../../data/land/* ./ <- 陸面スキーム用のパラメータファイル
$ ln -s ../../data/rad/* ./ <- 放射スキーム用のパラメータファイル

```

準備が整ったら、4つのMPIプロセスを使用して scale-les を実行する。

```
$ mpirun -n 4 ./scale-les run.conf < /dev/null >&log&
```

実行にはおおよそ1~2時間を要する(評価環境では93分)。長時間の実行になるため、上記のように標準出力をファイルへ書き出すようにしてバックグラウンドで実行すると便利である。計算が開始されれば、処理内容のログとして、"LOG\_d01.pe000000"ファイルが生成されるので、例えば下記のようなコマンドで"LOG\_d01.pe000000"ファイルを参照すれば、どこまで計算が進んでいるかチェックすることができる。

```
$ tail -f LOG_d01.pe000000
```

正常にジョブが終了すれば、history\_d01.pe#####.nc と restart\_d01.pe#####.nc という名前のファイルがMPIプロセス数だけ、つまり4つつつ生成される(#####にはMPIプロセスの番号が入る)。historyファイルは計算結果の出力ファイルであり、HISTITEMに指定した変数のみ書き出される。restartファイルは対応する時刻を開始時刻として再計算を開始するための初期値ファイルである。次節でhistoryデータをGrADSで描画可能なバイナリーデータに変換して結果を確認する方法について説明する。

## 4.5 結果を描画する

SCALEモデルの出力ファイルは、MPIプロセス毎に計算領域が分割された状態で出力される。それぞれのファイルフォーマットは気候・予報(CF)メタデータ規約に対応したnetcdf4形式である<sup>1</sup>。ここでは、1) プロセス毎に分割されたnetcdfファイルをgradsで扱えるように1つのバイナリーファイルにまとめ、2) 作図して結果の確認を行う。

### GrADS バイナリーに変換

分割されたnetcdfからGrADSバイナリー変換するには、netcdf2grads\_hを使用する。詳細な使用法は5.7節を参照頂くこととして、ここでは最低限の手続きのみ説明する。まず、net2gディレクトリへ移動する。

```
$ cd ${Tutorial_DIR}/real/net2g
```

2.4節でコンパイルしたバイナリーファイルにリンクを張る。

```
$ ln -s ../../../../util/netcdf2grads_h/net2g ./
```

ここでは例として、2次元変数であるMSLP、PRECを、3次元変数として850hPa,500h,200hPa面のU、Vを変換する。2次元変数のためのconfigure fileはnet2g.2d.confに、3次元変数のためのconfigure fileはnet2g.3d.confに用意している。

netcdf2grads\_h実行時のプロセス数は、計算実行時に使用したプロセス数の約数である必要がある。ここでは、計算に用いたのと同じ4プロセスを使用することとする。

```
$ mpirun -n 4 ./net2g net2g.2d.conf
```

```
$ mpirun -n 4 ./net2g net2g.3d.conf
```

<sup>1</sup>gpviewがインストールされている場合、gpviewを使って作図することも出来る。gpviewならばhistoryデータを変換することなく直接作図することができるため、クイックチェックに適している。

両方とも数秒程度で終了する。成功すれば、下記のファイルが作成される。

```
MSLP_d01z-2d.ct1
MSLP_d01z-2d.grd
PREC_d01z-2d.ct1
PREC_d01z-2d.grd
U_d01z-3d.ct1
U_d01z-3d.grd
V_d01z-3d.ct1
V_d01z-3d.grd
```

#### 計算結果の確認

現在のバージョンの `netcdf2grads_h` では、SCALE の XY 格子座標で、`ctl` ファイルを作成する。今後、緯度経度座標で描画するための `ctl` ファイルを出力できるようにする予定であるが、現バージョンは未対応のため、緯度経度座標で作図するための `ctl` を別途用意している。

```
MSLP_d01z-2d_lcc.ct1
PREC_d01z-2d_lcc.ct1
U_d01z-3d_lcc.ct1
V_d01z-3d_lcc.ct1
```

計算結果確認用の図を作成するためのスクリプト `checkfig.gs` を使って作図する。

```
$ grads -blc checkfig.gs
```

成功すると、下記の図が作成される。なお、GrADS のバージョンによって文法が異なるので、Warning が出る場合は、適宜変更する。

```
real_mslp.png
real_prec.png
real_wind.png
```

計算と変換が成功していれば、下記と同じ図が描画される。

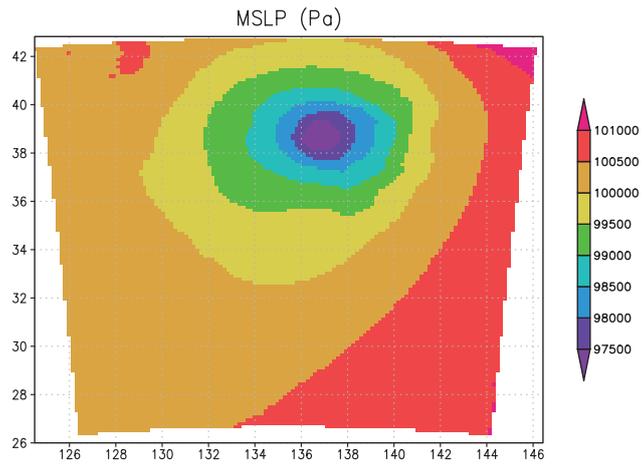


図 4.4: 計算開始から 12 時間後の海面更正気圧

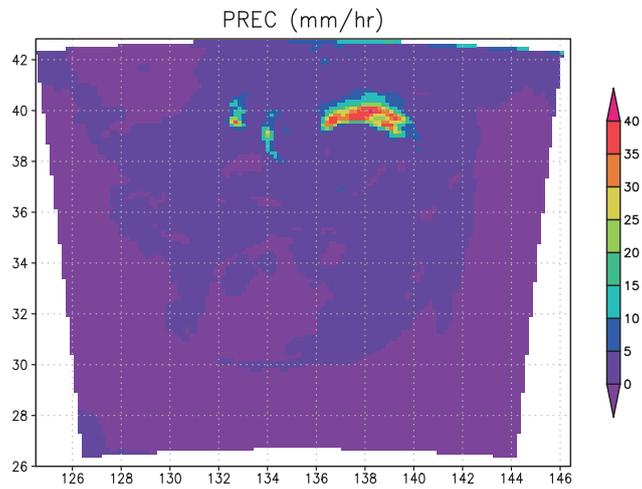


図 4.5: 計算開始から 12 時間後の降水フラックス

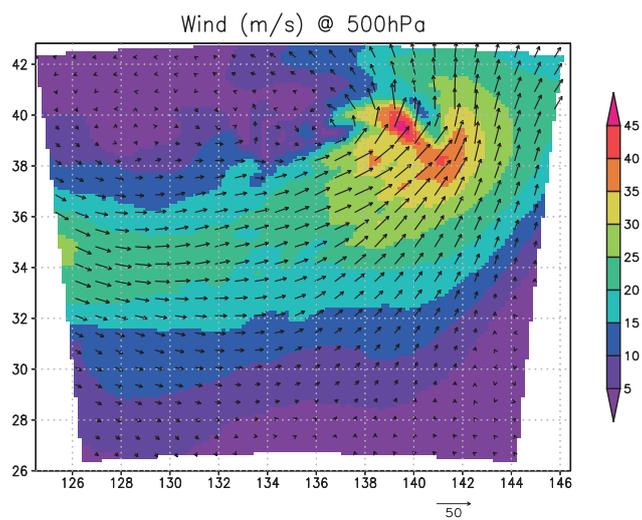


図 4.6: 計算開始から 12 時間後の 500hPa の風速と風ベクトル

## 第5章 SCALE-LES Advance use

### 5.1 実験設定の詳細

ここではチュートリアルで触れなかった実験設定の詳細について説明する。SCALEでサポートする全てを網羅してはいないが、格子間隔、計算領域、積分時間間隔、物理過程等の重要な設定方法についてはできる限り丁寧に説明するように心がけた。実験設定は全て、`pp_***.conf`、`init_***.conf`、および `run_***.conf` の `config` ファイルを編集することで設定できる。従って、以降の説明は全て `config` ファイルの編集方法について述べたものである。`config` ファイルの設定変数の詳細は付録 B に記載があるので、本節と併せて読み進めて欲しい。また以下の説明は、第3章と第4章のチュートリアルをすでに実行してきたものとして説明する。

#### 5.1.1 格子間隔

はじめに、この設定を行う上で注意しなければならない点を挙げる。以下で説明する **格子間隔の設定は、`pp_***.conf`、`init_***.conf`、`run_***.conf` の `config` ファイルの間で必ず一致させなければならない**。このことに注意して設定変更を進めてもらいたい。

SCALEでは、格子点の位置を均等間隔で設定することも、任意の格子点位置を直接指定することもできる。均等間隔で設定する場合は、`config` ファイルの `PARAM_GRID` の `DX`、`DY`、`DZ` の変数を指定することで、格子間隔を指定できる (`DX`、`DY`、`DZ` はそれぞれ、東西、南北、鉛直方向の格子間隔を意味する)。一方、直接格子点の位置を指定する場合は、`FZ`<sup>1</sup> などの格子点位置変数に対して直接に値を代入する。これらの変数は全て、倍精度浮動小数点変数で単位はメートル [m] である。

例として理想実験のチュートリアルの `run.conf` ファイル (`run_R20kmDX500m.conf`) を見てみると次のように記載されている。

---

<sup>1</sup>SCALEの格子系は Arakawa-C グリッド、および Lorenz グリッドであるため、水平方向と鉛直方向ともにスタッガード点 (1/2 ずれた点) に位置する変数がある。コントロールボリュームに対して中心点に位置する格子点を Center Point と呼び、コントロールボリュームの面に位置する格子点 (Center Point に対して 1/2 ずれている) を Face Point と呼ぶ。そして、これらの頭文字と方向を組み合わせると `CX`、`CY`、`CZ` や `FX`、`FY`、`FZ` と呼ぶ。

```

&PARAM_GRID
DX = 500.DO,
DY = 500.DO,
FZ(:) =
  80.00000000000000 ,
  168.00000190734863 ,
  264.80000610351567 ,
  ~ 中略 ~
  14910.428862936289 ,
  15517.262523292475 ,
  16215.121232702089 ,
  17017.658748523147 ,
  17940.576891717363 ,
  19001.932756390710 ,
  20222.492000765058 ,
BUFFER_DZ = 5000.DO,
BUFFFACT = 1.0D0,
/

```

DX、DY は“500.DO”と倍精度表記を用いて 500 m と指定されており、水平方向の格子間隔は 500 m の均等間隔に設定されていることがわかる。例えば、2 km の格子間隔で実験がしたければ、ここに“2000.DO”と記述すればよい。

一方、FX(:) の項目は鉛直方向の格子点位置を直接指定している。ここには示していないが、同じ config ファイル中の PARAM\_INDEX の項目に“KMAX = 97”として鉛直 97 層であると設定されているため、最下層から再上層まで 97 層分の位置が指定されている。これを見れば、最下層の高度は 80 m、最上層の高度は約 20 km であるとわかる。モデルの鉛直層とり方を変えたい場合は、これに習って KMAX で指定した層数に応じて高度をしていすればよい<sup>2</sup>。

## バッファ格子点の設定

SCALE では計算領域境界のすぐ内側に「バッファ格子点」を設定することができる。バッファ格子点では、指定された値や親モデルのデータにダンピングする、いわゆるナッジングが行われる。これは、モデル最上層で重力波が反射することを緩和するための「スポンジ層」として利用したり、現実大気実験を行う際に側面境界において親モデルと SCALE で計算される大気場が乖離することを防ぐための「緩和領域」として利用される。

バッファ格子点は、BUFFER\_DX、BUFFER\_DY、BUFFER\_DZ、および BUFFFACT の変数によって設定される。BUFFER\_DX、BUFFER\_DY、BUFFER\_DZ は、それぞれの方向のバッファ領域の幅（単位はメートル）を示す。バッファ領域は境界から内側に BUFFER\_DX 等で指定された幅で設置され、バッファ領域内の格子点はバッファ格子点として認識される。したがって、その分だけナッジングの影響を受けないで計算される自由領域は狭くなることに注意が必要である。水平方向には東西南北どの方向にも同様にバッファ領域が設定されるが、鉛直方向には計算領域の上端にのみバッファ領域が設定され、下端には設定されない。BUFFFACT はバッファ領域内の格子点の格子間隔に対するストレッチ係数である。下記の関係式によって、バッファ領域内の格子間隔（BDX）は決定される。

$$BDX_{i+1} = BDX_i^{BUFFFACT}$$

$$BDX_1 = DX_{default}$$

<sup>2</sup>独自の鉛直層設定の作成をサポートするツールとして、scale/scale-les/util/makevgrid のディレクトリ下に“make\_vgrid.f90”という Fortran プログラムといくつかのサンプル namelist が用意されている。これをコンパイルして実行すれば直ちに config ファイルに貼り付けて使用できるフォーマットになった鉛直層設定を作成できる。

ここで、 $i$  はバッファ領域内の格子点番号を表し、計算領域の内側から外側へ向かって番号が振られる。 $DX_{default}$  は PARAM\_GRID の項目で設定される元々の DX を意味する。 $i$  の最大値  $n_i$  は、 $\sum_{i=1}^n iBDX_i > BUFFER_{DX}$  の関係を達成する最小の数として決定される。したがって、“BUFFFACT = 1.0” ならば、格子間隔をストレッチせずに等間隔であることを意味し、“BUFFFACT = 1.2” ならば、計算領域の内側から境界へ向けて、格子点間距離がもともとの格子間隔から 1.2 倍の割合で広がっていくことを意味する。BUFFFACT の値を大きくすると、バッファ領域の幅が同じでもバッファ格子点として使用される格子点数は少なくなる。

例として理想実験のチュートリアル run.conf ファイル (run\_R20kmDX500m.conf; 上に表示したもの) を見てみると、BUFFER\_DZ = 5000.D0、BUFFFACT = 1.0D0 と指定されている。BUFFER\_DZ 以外には指定がないので、水平方向にはバッファ領域が設定されない (この実験は水平方向には周期境界条件になっておりバッファ領域は必要ない)。BUFFFACT = 1.0D0 となっているため、もとの格子間隔のままストレッチは行わない。最上層の高度が 20222 m と指定されており、鉛直方向のバッファ領域の幅は 5000 m と指定されているため、モデル計算領域のトップから高度 15222 m までがバッファ領域となる。鉛直層の高度指定によれば、92 層目の高度が 15517 m で、91 層目の高度が 14910 m であるため、92 層目まではバッファ領域だが、91 層目はバッファ領域ではない。繰り返しになるが、これらの設定は pp\_\*\*\*.conf、init\_\*\*\*.conf、run\_\*\*\*.conf のすべての config ファイルにおいて共通した設定になっていなければならない。

SCALE では、バッファ領域の大きさ、バッファ格子点の数について、まだ明確な指標を設定できていないが、鉛直方向 (計算領域トップ) のバッファ格子点は 5 点以上、水平方向 (側面境界付近) のバッファ格子点は 20 ~ 40 点程度を推奨している。実験設定や事例によっては、さらにバッファ格子点を増やしたり、ストレッチ係数を用いてバッファ領域を広げる処理が必要であったり、ここでは説明しなかったが ATMOS\_BOUNDARY\_taux、ATMOS\_BOUNDARY\_tauy といった設定項目を調整してバッファ領域のナッジング強度を強める必要があったりする。

### 5.1.2 計算領域の大きさ

ここで説明する計算領域の設定は、pp\_\*\*\*.conf、init\_\*\*\*.conf、run\_\*\*\*.conf の config ファイルの間で必ず一致させなければならない。このことに注意して設定変更を進めてもらいたい。

SCALE は MPI 並列プログラムであり、全計算領域は複数の MPI プロセスに分割されて並列実行される。計算領域は水平 2 次元 (東西方向 × 南北方向) に分割される。したがって計算領域の大きさは、MPI プロセス数と 1 つの MPI プロセスが担当する格子点数によって決まる。以降、X 方向は水平軸の東西方向、Y 方向は水平軸の南北方向をそれぞれ意味するものとする。

X 方向、および Y 方向の MPI プロセスの割付設定は、config ファイルの PARAM\_PRC の項目にある PRC\_NUM\_X、および PRC\_NUM\_Y よって指定される。また、1 つの MPI プロセスが担当する格子点数は、PARAM\_INDEX 項目にある KMAX、IMAX、JMAX の変数によって指定され、それぞれ鉛直方向、X 方向、Y 方向の格子点数を表す。したがって、X 方向の総格子点数は PRC\_NUM\_X × IMAX、Y 方向の総格子点数は PRC\_NUM\_Y × JMAX と表現される。また、計算に必要な総 MPI プロセス数は、PRC\_NUM\_X × PRC\_NUM\_Y となる。実行時に MPI コマンドに指定する MPI プロセス数は、この総 MPI プロセス数を指定しなければならない。この条件を満たさない場合は、LOG ファイルなどに

```
xxx total number of node does not match that requested. Check!
```

というメッセージが出力がされて計算が異常終了するので、設定、もしくは実行コマンドを見なおしてもらいたい。

ここまでの復習として、現実大気実験のチュートリアルで使用した `run.conf` ファイルを例にして、上述の内容を確認してみる。

```
&PARAM_PRC
  PRC_NUM_X = 2,
  PRC_NUM_Y = 2,
  PRC_PERIODIC_X = .false.,
  PRC_PERIODIC_Y = .false.,
/

&PARAM_INDEX
  KMAX = 36,
  IMAX = 60,
  JMAX = 60,
/
```

まず MPI プロセス数については `PRC_NUM_X = 2`、`PRC_NUM_Y = 2` と設定されているため、X 方向、Y 方向ともに 2 分割されており、総計として 4 つの MPI プロセスを使用した計算となることがわかる。一方、1 つの MPI プロセスあたりの格子点数については、`IMAX = 30`、`JMAX = 30` と指定されているため、総格子点数としては、X 方向、Y 方向ともに  $2 \times 60 = 120$  となり 120 格子点である。同じ config ファイル内に記載されている `PARAM_GRID` の項目の `DX`、`DY` はともに 15000 m (15 km) と指定されているため、 $120_{grid} \times 15_{km} = 1800_{km}$  の計算から、 $1800_{km} \times 1800_{km}$  の正方形の計算領域が設定されていることがわかる。

`PARAM_PRC` の項目の `PRC_PERIODIC_X`、`PRC_PERIODIC_Y` は側面境界条件に関する設定である。その詳細は次節で説明する。

#### 計算領域の変更例

ここでは、現実大気実験のチュートリアルの config ファイル (`run.conf`) を元にしていくつかの変更例を説明する。以下の変更例をもとに、自分の行いたい実験設定にあった config ファイルを作り上げてもらいたい。

##### a. MPI プロセス数はそのままに計算領域を 4 倍に広げる

この設定はとても簡単である。デフォルト設定は、X 方向、Y 方向ともに 2 つの MPI プロセスを使用し、格子点数はともに 30 点なので、全領域の格子点数は、 $(2 \times 60)_x \times (2 \times 60)_y = 14400$  点である。従って、計算領域を 4 倍に広げるためには、`IMAX`、`JMAX` の値をデフォルトに対して 2 倍の値に変更するだけである。これで、全領域の格子点数は、 $(2 \times 120)_x \times (2 \times 120)_y = 57600$  点となり、14400 点の 4 倍の計算領域になっている。この変更を施した config ファイルの例は次のとおりである。赤字の部分でデフォルトからの変更点を意味する。

```

&PARAM_PRC
  PRC_NUM_X = 2,
  PRC_NUM_Y = 2,
  PRC_PERIODIC_X = .false.,
  PRC_PERIODIC_Y = .false.,
/

&PARAM_INDEX
  KMAX = 36,
  IMAX = 120,
  JMAX = 120,
/

```

#### b. 1つのMPIプロセスあたりの格子点数はそのままに計算領域を4倍に広げる

先程は、MPIプロセス数を維持して計算領域を広げたが今度は、1つのMPIプロセスあたりの格子点数はそのままに、MPIプロセス数を増やすことで計算領域を4倍に広げる方法を説明する。1つのMPIプロセスあたりの格子点数（プロセスあたりの演算量）はそのままにMPIプロセス数を増やすことで計算領域を広げる（総演算量を増やす）。この場合、1つのCPUが担当する問題サイズは変わらないため、理想的には計算にかかる時間を増やすことなく計算領域を広げることができる<sup>3</sup>。

計算領域を4倍に広げるためには、PRC\_NUM\_X、PRC\_NUM\_Yの値をデフォルトに対して2倍の値に変更するだけである。これで、全領域の格子点数は、 $(4 \times 60)_x \times (4 \times 60)_y = 57600$ 点となり、14400点の4倍の計算領域になっている。このとき必要なMPIプロセス数は、 $4 \times 4 = 16$ プロセス、つまりデフォルトの4倍のMPIプロセス数が必要になる。この変更を施したconfigファイルの例は次のとおりである。赤文字の部分がデフォルトからの変更点を意味する。

```

&PARAM_PRC
  PRC_NUM_X = 4,
  PRC_NUM_Y = 4,
  PRC_PERIODIC_X = .false.,
  PRC_PERIODIC_Y = .false.,
/

&PARAM_INDEX
  KMAX = 36,
  IMAX = 60,
  JMAX = 60,
/

```

#### c. 計算領域はそのままに水平格子間隔を3 kmに変更する

現実大気実験チュートリアルでのデフォルト設定は、X方向、Y方向の総格子点数は120点で水平格子間隔が15 kmなので、計算領域は1800 km × 1800 kmの領域となっている。ここでは、この領域を維持したまま水平格子間隔を3 kmに狭める設定にトライする。水平格子間隔が15 kmから3 kmへ1/5だけ小さくなるので逆に1方向あたりの総格子点数は5倍、つまり600点必要になる。この600点をどのようにMPIプロセス数とプロセスあたりの格子点数として割り振るかはユーザーの環境や計算機リソース量によって異なる。たとえば、X方向、Y方向ともに10プロセスずつ、合

<sup>3</sup>計算科学用語では、この変更、つまりプロセスあたりの演算量一定で総演算量を増やしていくことを“weak scaling”と呼ぶ。

計で 100 プロセスを使えば、プロセスあたりの格子点数は 60 点となり、積分時間間隔が短くなることを無視すれば演算量はデフォルトと変わらない。

しかし、なかなか 100 プロセスを利用できる計算機を持っている環境にいる人は少ないだろう。そこで、デフォルトから各方向に 1 つずつ MPI プロセス数を増やし、 $3 \times 3 = 9$  プロセスを利用した設定を考えてみる。1 方向あたりの総格子点数は 600 点なので、プロセスあたりの格子点数は  $600 \div 3 = 200$  点となる。

ここでの変更で気をつけなければならないことは、バッファ領域の幅である。現実大気実験チュートリアル<sup>1</sup>のデフォルト設定ではバッファ領域は、計算領域トップと東西南北の側面境界に設定されており、側面境界のバッファ領域の幅は片側 300 km、つまり 15 km 格子間隔で 20 点のバッファ格子点が確保されている。水平格子間隔を 15 km から 3 km へ変更したので、このままでは 100 点もの格子点がバッファ領域に取られてしまう。SCALE では一般的に 20~40 点程度のバッファ格子点を設定するようにしているので、デフォルトと同じ 20 点になるように側面境界のバッファ領域の幅は片側 60 km と設定する。鉛直層設定は変更していないため、計算領域トップのバッファ領域については設定を変更する必要はない。

この変更を施した config ファイルの例は次のとおりである。赤文字の部分がデフォルトからの変更点を意味する。

```
&PARAM_PRC
  PRC_NUM_X = 3,
  PRC_NUM_Y = 3,
  PRC_PERIODIC_X = .false.,
  PRC_PERIODIC_Y = .false.,
/

&PARAM_INDEX
  KMAX = 36,
  IMAX = 200,
  JMAX = 200,
/

&PARAM_GRID
  DX = 3000.DO,
  DY = 3000.DO,
  FZ(:) = 80.841D0, 248.821D0, ... .. 1062.158D0,
          1306.565D0, 1570.008D0, ... .. 2845.575D0,
          ~ 中略 ~
          18387.010D0, 19980.750D0, ... .. 28113.205D0,
  BUFFER_DZ = 5000.DO,
  BUFFER_DX = 60000.DO, ! 20 buffer
  BUFFER_DY = 60000.DO, ! 20 buffer
/
```

### 5.1.3 地図投影法と計算領域の位置

計算領域の位置と投影法は、config ファイルの PARAM\_MAPPROJ の項目を編集することで設定できる。この設定も、pp\_\*\*\*.conf、init\_\*\*\*.conf、run\_\*\*\*.conf の config ファイルの間で必ず一致させなければならない。はじめに下記の例をもとに説明する。

```

&PARAM_MAPPROJ
MPRJ_basepoint_lon = 138.727778D0,
MPRJ_basepoint_lat = 35.360556D0,
MPRJ_type = 'MER',
/

```

まず MPRJ\_basepoint\_lat と MPRJ\_basepoint\_lon は、計算領域の中心の緯度・経度を表す。SCALE では、北緯を正、南緯を負の値として表現し、経度は0度を起点に右回りで表現するため、この設定では計算領域の中心が北緯 35.360556 度、東経 138.727778 度に位置することになる。この場所を中心に指定された大きさで、計算領域が設定される<sup>4</sup>。実際には SCALE 内部での格子点は実距離に基づいて格子点が配置されるので、投影法で設定されるのは、実距離に基づいた緯度・経度座標が計算される。この緯度・経度情報は、すべての SCALE の NetCDF 形式の出力ファイルに含まれている。

MPRJ\_type は、地図投影法の種類を表しており、MER はメルカトル図法を意味する。SCALE で現在選択できる地図投影法とその指定文字列は次のとおりである。

表 5.1: SCALE で選択できる地図投影法

地図投影法	MPRJ_type
地図投影なし (理想実験用)	NONE
ランベルト正角円錐図法	LC
極心平射図法 (ポーラーステレオ)	PS
メルカトル図法	MER
正距円筒図法	EC

メルカトル図法以外の投影法も、MPRJ\_type の指定を変更するだけで、上記と同じように使用可能であるが、ランベルト正角円錐図法の設定方法については、設定方法が異なるため以下で説明する。ここでは、現実大気実験チュートリアルで使用した run.conf ファイルを例に挙げる。

```

&PARAM_MAPPROJ
MPRJ_basepoint_lon = 135.220404D0,
MPRJ_basepoint_lat = 34.653396D0,
MPRJ_type = 'LC',
MPRJ_LC_lat1 = 30.00D0,
MPRJ_LC_lat2 = 40.00D0,
/

```

SCALE では “standard parallel type” の実装を採用しているため、投影を決定する上で 2 つの “standard latitude” の位置を指定する必要がある。2 つの standard latitude に挟まれた領域では、緯線・経線の長さの比が地球楕円体面上における長さの比と近くなるように調節される。従って、メルカトル図法の場合に比べて、standard latitude を設定する MPRJ\_LC\_lat1 と MPRJ\_LC\_lat2 の項目が追加されている。それぞれ、南側、北側の standard latitude の値を “degree” で指定する。

さらに下記のように MPRJ\_basepoint\_x と MPRJ\_basepoint\_y という変数を用いることで、地図投影中心と計算領域中心をずらすこともできる。

<sup>4</sup>デフォルトではメルカトル図法に基づいて緯度・経度を計算する際の基準とする緯度は、MPRJ\_basepoint\_lat の値が使用されるが、MPRJ\_M\_lat を用いて任意の緯度を指定することもできる。

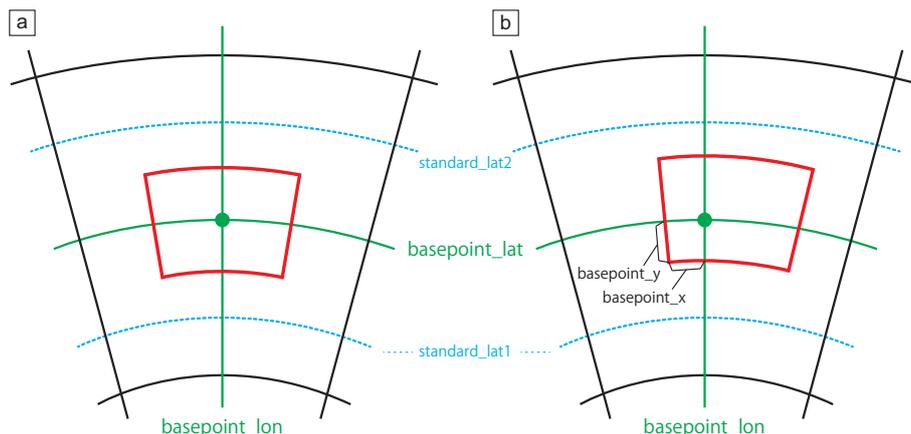


図 5.1: 投影中心と計算領域の関係 : (a) はデフォルト設定の場合、(b) は計算領域の位置を投影中心からずらした場合。赤線の矩形が計算領域を表す。

```

&PARAM_MAPPROJ
MPRJ_basepoint_lon = 135.220404D0,
MPRJ_basepoint_lat = 34.653396D0,
MPRJ_basepoint_x = 100.0D0,
MPRJ_basepoint_y = 100.0D0,
MPRJ_type = 'LC',
MPRJ_LC_lat1 = 30.00D0,
MPRJ_LC_lat2 = 40.00D0,
/

```

MPRJ\_basepoint\_x と MPRJ\_basepoint\_y は、地図投影中心の位置を、計算領域の南西端（左下角）からの距離で指定するパラメータで、単位はメートルである。これらを指定しない場合は、デフォルト設定として計算領域中心と地図投影中心の位置は一致する。上記の場合とデフォルト設定の場合を比較したものが図 5.1 である。図 5.1a はデフォルト設定で投影中心と計算領域中心が一致している場合、図 5.1b は、計算領域の位置を投影中心からずらした場合の関係を表している。図 5.1b では、計算領域の南西端から MPRJ\_basepoint\_x と MPRJ\_basepoint\_y で指定した距離だけ離れた位置に投影中心がある。

#### 5.1.4 側面境界条件

SCALE では、側面境界条件として「周期境界条件」と「外部入力データ指定」の 2 種類から選択できる。デフォルトの設定は東西方向、南北方向ともに周期境界条件となっている。主に理想実験では周期境界条件を使用し、現実大気実験では外部入力データ指定を使用することを想定している。設定方法によっては、他の側面境界条件も設定可能であるが、現在のところ想定外の使用方法であるためサポートできない。

側面境界条件の設定は、config ファイルの PARAM\_PRC の項目を編集することで設定できる。この設定は、pp\_\*\*\*.conf、init\_\*\*\*.conf、run\_\*\*\*.conf の config ファイルの間で必ず一致させなければならない。周期境界条件を設定したい場合は、デフォルト設定なので特に config ファイルに記述する必要はない。理想実験チュートリアル config ファイルの PARAM\_PRC の項目を見れば特に境界条件に関する記述がないことがわかるだろう。一方、外部入力データ指定を設定したい場合は、下

記のように “PRERIODIC” のスイッチを “false” に指定する。

```
&PARAM_PRC
  ~ 中略 ~
  PRC_PERIODIC_X = .false.,
  PRC_PERIODIC_Y = .false.,
/
```

外部入力データ指定の場合は、5.1.1 節で説明した必ず側面境界のバッファ領域を設定しなければならない。バッファ領域でどの変数に強制をかけるか（ダンピングするか）、またその場合の時定数などの設定は config ファイルの PARAM\_ATMOS\_BOUNDARY の項目で指定できる。現実大気実験のチュートリアル config ファイル (run.conf) を元にして一部の項目を説明する。

```
&PARAM_ATMOS_BOUNDARY
  ATMOS_BOUNDARY_TYPE = "REAL",
  ATMOS_BOUNDARY_IN_BASENAME = "../init/boundary_d01",
  ATMOS_BOUNDARY_USE_VELZ = .true.,
  ATMOS_BOUNDARY_USE_QHYD = .false.,
  ATMOS_BOUNDARY_VALUE_VELZ = 0.0D0,
  ATMOS_BOUNDARY_UPDATE_DT = 21600.0D0,
/
```

上 2 つの項目は、“REAL” が外部入力データを使用することを意味し、次の行の指定がその外部入力データのファイル名を指定している。上から 3 つ目の設定項目である “ATMOS\_BOUNDARY\_USE\_VELZ = .true.” は、鉛直速度に対して「強制をかける」ことを意味している。一方、“ATMOS\_BOUNDARY\_USE\_QHYD = .false.” として、凝結物の混合比に対しては逆に「強制をかけない」設定になっている。その次の項目の “ATMOS\_BOUNDARY\_VALUE\_VELZ” は、鉛直速度に対して強制をかける際、ここで指定した値、“0.0 m/s” へ近づくように強制をかけるという指定を意味する。最後の行の “ATMOS\_BOUNDARY\_UPDATE\_DT” は、外部入力データの更新間隔が 21600 秒であることを意味している。たとえば、6 時間間隔でデータが与えられている再解析データを外部入力データとして使用する場合にこの設定になる。

他にも、水平速度東西成分 (VELX)、水平速度南北成分 (VELY) や温位 (POTT) などに対して同様の設定項目が存在する。また、ダンピングの時定数を設定する ATMOS\_BOUNDARY\_TAUX や ATMOS\_BOUNDARY\_TAUY といった設定項目がある。更なる詳細については、付録 B を参照のこと。

### 5.1.5 積分時間と積分時間間隔の設定

例えば理想実験チュートリアルでは、積分時間は 1 時間とし、時間積分間隔として力学過程は 1.0 秒、雲物理過程は 10.0 秒で実行したが、積分時間を伸ばしたい場合や、計算にかかる時間を短くするために積分時間間隔を長くしたり、計算不安定を防ぐために積分時間間隔を短くすることがあるだろう。

積分時間と積分時間間隔の設定は、config ファイル run\_\*\*\*.conf の PARAM\_PRC の項目を編集することで設定できる。この項目はモデル本体の実行だけで有効である。理想実験チュートリアルで使った run\_R20kmDX500m.conf の例を示す。

```

&PARAM_TIME
TIME_STARTDATE = 0000, 1, 1, 0, 0, 0, (計算開始の日付:放射過程を用いる実験等で必要)
TIME_STARTMS = 0.00, (計算開始時刻 [mili sec])
TIME_DURATION = 3600.000, (積分時間 [単位は TIME_DURATION_UNIT で決定])
TIME_DURATION_UNIT = "SEC", (積分時間 TIME_DURATION の単位)
TIME_DT = 5.000, (移流のタイムステップ)
TIME_DT_UNIT = "SEC", (TIME_DT の単位)
TIME_DT_ATMOS_DYN = 1.000, (力学過程のタイムステップ)
TIME_DT_ATMOS_DYN_UNIT = "SEC", (TIME_DT_ATMOS_DYN の単位)
TIME_DT_ATMOS_PHY_MP = 10.000, (雲物理過程のタイムステップ)
TIME_DT_ATMOS_PHY_MP_UNIT = "SEC", (TIME_DT_ATMOS_PHY_MP の単位)
/

```

上記の各部分を変更することで積分時間や積分時間間隔を変更することができる。

## 5.2 物理モデルの利用

本節では雲物理モデルや乱流モデルなどの物理モデルの利用方法を説明する。チュートリアル実験で用いてきた物理モデルについて理解するために config ファイルの設定を再確認しながら説明をすすめる。ここでは例として、理想実験チュートリアルの sample ディレクトリ内にある run\_R20kmx20kmDX500m.conf を取り上げる。

### 5.2.1 雲微物理スキームの設定

チュートリアルでは、Tomita (2008) の 1-moment バルク法を用いていたが、SCALE には暖かい雲のみを考慮する 1-moment バルク法 (Kessler (1969)), 氷雲を含んだ 2-moment バルク法 (Seiki and Nakajima (2014)), 1-moment ビン法の 4 種類の雲微物理スキーム (Kentaroh et al. (2010)) も利用することが可能である。これらの雲微物理スキームの選択は、init.conf と run.conf に記載されている PARAM\_TRACER に含まれる “TRACER\_TYPE” 及び、PARAM\_ATMOS に含まれる “ATMOS\_PHY\_MP\_TYPE” によって設定する。両者は同じにする必要があり、選択する雲微物理スキームによって以下のように設定する。

KESSLER : 水雲のみの 1-moment バルク法 (Kessler 1969)

TOMITA08 : 氷雲を含む 1-moment バルク法 (Tomita 2008)

SN14 : 氷雲を含む 2-moment バルク法 (Seiki and Nakajima 2014)

SUZUKI10 : 1-moment ビン法 (Suzuki et al. 2010, 氷雲を含むか否かはオプションで選択)

このとき **TRACER\_TYPE** と **ATMOS\_PHY\_MP\_TYPE** は **init.conf**, **run.conf** で同一にする必要があることに注意されたい。SUZUKI10 以外を選択した場合は、init.conf、run.conf の TRACER\_TYPE と ATMOS\_PHY\_MP\_TYPE を変更するだけで実行可能である。一方 SUZUKI10 を選択した場合は、init.conf、run.conf の双方に

```

&PARAM_BIN
nbin = 33, (ビンの数)
ICEFLG = 1, (氷雲を考慮するか否か, 0->水雲のみ, 1->氷雲も含む)
/

```

を追記し、さらに init.conf に記載されている PARAM\_MKINIT に 「flg\_bin = .true. 」を追加する必要がある。

```

&PARAM_MKINIT
  ~ 中略 ~
  flg_bin = .true.
  ~ 中略 ~
/

```

この場合も、`init.conf`と`run.conf`に記載される`PARAM_BIN`は同一にする必要がある。SUZUKI10を選択した時には、`micpara.dat`という雲微物理の計算に必要なファイルが自動生成される。`micpara.dat`がすでに存在する場合は存在する場合はあるものを利用するが、`nbin`が変わると新たに作成しなければならない。`micpara.dat`に`nbin`の情報が記載されているが、もし`run.conf`に記載される`nbin`と`micpara.dat`に記載されている`nbin`が異なれば、

```

xxx nbin in inc_tracer and nbin in micpara.dat is different check!

```

というエラーメッセージを標準出力に出力して計算が落ちるようになっている。そのため、`nbin`を変更した際は、`micpara.dat`を消去して新たに作り直す必要がある（`micpara.dat`を消して再度SCALEをSUZUKI10を用いて実行すれば自動的に新しい`micpara.dat`が生成される）。

## 5.2.2 乱流スキームの設定

理想実験チュートリアルでは乱流スキームは導入されていなかったが、SCALEにはSmagorinsky typeの乱流スキーム (Smagorinsky (1963); Lilly (1962); Brown 1994; Scotti et al. 1994) と Mellor-Yamada Level 3の乱流スキーム (MYNN, Mellor and Yamada (1982), Nakanishi and Niino (2004)) 導入されている。これらを利用するには、`init.conf`と`run.conf`双方の`PARAM_ATMOS`に「`ATMOS_PHY_TB_TYPE`」を加える。

`ATMOS_PHY_TB_TYPE="SMAGORINSKY"` (Smagorinsky typeのサブグリッドモデル)

`ATMOS_PHY_TB_TYPE="MYNN"` (Mellor-Yamada Level 3のRANSモデル)

また`run.conf`の`PARAM_TIME`に

`TIME_DT_ATMOS_PHY_TB = 0.10D0`, (乱流スキームの時間ステップ)

`TIME_DT_ATMOS_PHY_TB_UNIT = "SEC"`, ( `TIME_DT_ATMOS_PHY_TB` の単位 )

を加える。これらを設定した上で、`run.sh`を実行することで、乱流スキームを考慮した計算が可能になる。地表面フラックスのバルク係数を定めるスキーム、放射スキーム、都市スキームなどに関しては、現実事例を対象とした4章を参照されたい。

## 5.2.3 放射過程の設定

SCALEには、放射過程としてMSTRN (Sekiguchi and Nakajima (2008)) が実装されている。ここでは、MSTRNの設定方法を説明する。

放射モデルを実行するには、各種外部データとパラメータテーブルが必要である。オゾンのプロファイルなどの外部データとパラメータテーブルは、“`scale-les/test/data/rad/`”の中に与えられている。放射過程を利用する実験を行う場合は、必ず“`scale-les/test/data/rad/`”下の外部データとパラメータテーブルが必要である。詳細に関しては随時説明を加えていく。

## 5.2.4 陸面・海洋上の地表面フラックスモデルの設定

SCALE には、陸面モデルとしてバケツモデル (バルク交換係数は Beljaars and Holtslag (1991); Wilson (2001) に基づく) が実装されている。また海洋上のフラックスモデルとしてスラブモデルが実装されている。ここでは、陸面モデルと海洋モデルの設定方法について説明する。

陸面モデルを実行するには、土地利用データとそれに対応するパラメータテーブルが必要である。パラメータテーブルは、“scale-les/test/data/land/” の中に “param.bucket.conf” というファイルで与えられている。中には 10 種類を超える土地利用区分とそれらに対応する粗度長などのパラメータが与えられている。陸面が存在する領域で実験を行う場合は、必ず “param.bucket.conf” のパラメータテーブルが必要である。詳細に関しては随時説明を加えていく。

## 5.2.5 都市モデルの設定

SCALE には、都市モデルとして単層キャノピーモデル (Kusaka et al. (2001)) が実装されている。ここでは都市モデルの設定方法について説明する。詳細に関しては随時説明を加えていく。

## 5.3 任意のデータを SCALE で使用する

### 5.3.1 初期値・境界値データ

現在、SCALE では下記のデータの読み込みとそれらに基づく初期値・境界値作成に対応している。

表 5.2: SCALE が読込に対応する外部入力データフォーマット

データ形式	対応状況	FILETYPE_ORG	備考
バイナリデータ	対応	GrADS	データ読み込み用の namelist を別途必要とする。
NICAM データ	対応	NICAM-NETCDF	netCDF 形式の LatLon データに対応する。
WRF データ	対応	WRF-ARW	“wrfout”、“wrfst” の両方に対応する。
SCALE データ	対応	SCALE-LES	history データのみ対応; latlon カタログを必要とする。

これらの使い分けは、初期値・境界値作成時、すなわち “scale-les\_init” の実行時の config ファイルの PARAM\_MKINIT\_REAL の項目中の FILETYPE\_ORG に表 5.2 に示した設定値を指定することで使い分ける。

最も汎用的に使用するデータ形式は「バイナリデータ」になることと思う。ここでいうバイナリデータとは、「4 バイト単精度浮動小数点のダイレクトアクセス方式、Fortran 型バイナリデータ」を指す。その主な使用方法は、第 4 章の現実大気実験チュートリアルで説明したとおりである。GRIB/GRIB2 のデータ形式は、チュートリアルで説明した方法に基づいて、バイナリデータ形式を経由して SCALE に読み込ませることができる。その他に任意のデータを境界値に使用したい場合は、バイナリデータ形式に変換することで読み込ませることができる。

SCALE データ形式は主にオフライン・ネスティング実験で使用される。詳細については、5.4.1 節を参照されたい。NICAM データ形式は、native の icosahedral grid system データではなく、緯度・経度座標に変換されたデータのみ読み込みに対応している。WRF データ形式についてはモデル出力データをそのまま使用することができる。これらの読み込み方法に関しては随時説明を加えていく。

## 5.4 ドメインネスティング実験

本節では、SCALE でネスティング実験を行う方法について説明する。ネスティング実験とは、図 5.2 に示すように、水平格子間隔の異なる複数の計算領域（ドメイン）を設定し、領域が重複するように入れ子（ネスト）構造にすることで、広領域かつ高解像度のドメインを設定する計算領域設定方法である。図 5.2 の例では、3つのドメインを用いた3段ネスティング構成になっている。外側のドメインは比較的粗い水平解像度であるが広い領域を取ることで大きな場の構造を表現することができる。逆に内側のドメインは、比較的狭い領域であるが細かい水平解像度を取ることで対象とする現象の細かい構造を表現することができる。ここでは、入れ子構造のうち、データを渡す側のドメインを「親ドメイン」、データを受ける側のドメインを「子ドメイン」と称する。

SCALE はオフライン・ネスティング実験とオンライン・ネスティング実験の両方をサポートしている。オフライン・ネスティング実験は、はじめに親ドメインだけで時間積分を行い、その計算結果の history データを用いて、子ドメイン用の初期値・境界値を作成する。その後子ドメインの時間積分を行う。オンライン・ネスティング実験は、親ドメインと子ドメインを同時に実行し、適宜計算途中のデータを親ドメインから子ドメインへ MPI 通信によって受け渡しすることで、子ドメインの時間積分を行う。オンライン・ネスティング実験が実行できる計算機リソースがあれば、オンラインで実行することを推奨する。それは、オンラインの場合、子ドメインの境界条件の更新間隔は親ドメインの時間積分間隔に一致するため、可能な限り細かい境界条件の更新間隔を得ることができる。また、中間ファイルも作成されずディスクリソースにも優しい。

ドメインネスティング実験を行う上で共通する実験設定の制限事項は、基本的に以下の2点だけである。

- 親ドメインの領域は子ドメインの領域を完全に内包しなければならない。
- 親ドメインの積分時間は子ドメインの積分時間を完全に内包しなければならない。

これに加えて、オンライン・ネスティング実験の場合、現在のところ親ドメインと子ドメインの積分時間は一致させなければならない。SCALE ではオンライン・ネスティングであっても、親ドメインと子ドメインの間で鉛直層数、鉛直層設定、地図投影法、そして物理スキームが異なっても構わない。

オフライン、オンラインに関わらず、ドメイン間の格子間隔比率 ( $DX_{parent}/DX_{child}$ ) にシステム上は制限はないが、この比率が大きすぎると計算結果の物理的なパフォーマンスが下がる可能性がある。本書では5倍以下で使用することを推奨する。

以降、まずは実行方法がわかりやすいオフライン・ネスティング実験から説明し、ついでオンライン・ネスティング実験について説明する。config ファイルの名前に特に指定はないが、ここでの表記としては、“`***.parent.conf`”と表記すれば、親ドメインの config ファイルを編集することを意味し、“`***.child.conf`”と表記すれば、子ドメインの config ファイルを編集することを意味する。

### 5.4.1 オフライン・ネスティング実験の方法

ここでは、親ドメインは解像度は荒いが広領域の外側ドメインで、子ドメインは領域は狭いが高解像度の内側ドメインであることを想定する。このとき、2段ネスティングのオフライン・ネスティング実験の実行過程は次のようになる。

1. 親ドメインの時間積分計算を行う。
2. 親ドメインの出力ファイル（history）を用いて子ドメインの初期値/境界値を作成する。

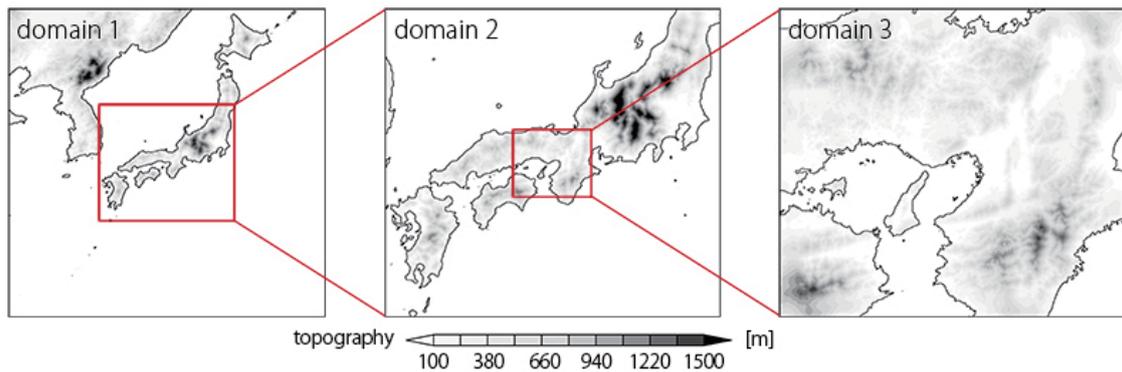


図 5.2: 日本の近畿地方を対象領域としたドメインネスティング設定の例. domain 1 が最外ドメインで domain 3 が最内ドメインである。赤い矩形と線は、それぞれの位置関係を示している。domain 1 の水平格子間隔は 7.5 km、domain 2 は 2.5 km、そして domain 3 は 0.5 km である。

### 3. 作成した初期値/境界値を用いて子ドメインの時間積分計算を行う。

以下、この流れに沿って説明を進める。親ドメインと子ドメインそれぞれについて、“pp.\*\*\*.conf”、“init.\*\*\*.conf”、そして“run.\*\*\*.conf” ファイルを事前に作成し、親ドメイン、子ドメインともに地形/土地利用データの作成を終え、親ドメインについては、初期値/境界値データの作成を終えていることを想定して説明を進める。ここで説明するオフライン・ネスティング実験の設定を記述した config ファイルがチュートリアルディレクトリの下、“tutorial/real/sample/offline\_nesting” に置いてあるので、説明を読み進める上で参考にしてもらいたい。

#### 親ドメインの時間積分計算を行う

基本的には通常のシングルドメインの計算と同じ方法で実行すればよいが、“run.conf” ファイルの設定について、次の 5 点の注意点・変更点がある。

- 親ドメインの history 出力間隔を適度に細かくとること。
- 親ドメインの history 出力変数に必要なものが揃っているか確認すること。
- 親ドメインの history 出力形式で ZINTERP は “false” に設定すること。
- 親ドメインの history 出力形式で STEP0 は “true” に設定すること。
- 親ドメインの計算領域を子ドメインへ伝える「カタログファイル」を出力すること。

この設定を “run.parent.conf” に適用すると下記ようになる。赤文字で示した部分が、上記の注意点・変更点に対応する部分である。

```

&PARAM_DOMAIN_CATALOGUE
  DOMAIN_CATALOGUE_OUTPUT = .true.,
/

&PARAM_HISTORY
  HISTORY_DEFAULT_BASENAME = "history",
  HISTORY_DEFAULT_TINTERVAL = 600.DO,
  HISTORY_DEFAULT_TUNIT = "SEC",
  HISTORY_DEFAULT_TAVERAGE = .false.,
  HISTORY_DEFAULT_DATATYPE = "REAL4",
  HISTORY_DEFAULT_ZINTERP = .false.,
  HISTORY_OUTPUT_STEP0 = .true.,
/

```

PARAM\_DOMAIN\_CATALOGUE の項目の “DOMAIN\_CATALOGUE\_OUTPUT” の変数がカタログファイルの出力設定である。もともとの config ファイルには項目自体がないこともあるので、その場合は自分で項目を加えて設定すること。カタログファイルは、“latlon\_domain\_catalogue.txt” というファイル名で出力される。この中には、MPI プロセス毎に分割して担当した計算領域の四隅の緯度・経度が記述されている。

HISTORY\_DEFAULT\_TINTERVAL の設定項目によって history データ出力間隔を指定する（単位は秒である）。指定値に任意性はあるが、子ドメインの側面境界条件の更新間隔として使用可能であると考えられる範囲で指定すること。親ドメイン、子ドメインの解像度、および実行環境のディスク空き容量にもよるが、おおよそ最大で 1 時間間隔、出来れば 10 分間隔の history データ出力間隔を指定することが多い。

また、history データを用いて初期値/境界値データを作成するために、下記の全ての変数を出力しなくてはならない。run.conf ファイルの “HISTORY\_ITEM” の項目を確認すること。

- T2, Q2, MSLP, DENS, MOMZ, MOMX, MOMY, RHOT
- QV, QC, QR, QI, QS, QG（親の雲微物理モデルに合わせて出力; 例えば Tomita08 なら全て）
- NC, NR, NI, NS, NG（親の雲微物理モデルに合わせて出力; 例えば Tomita08 なら不要）
- LAND\_SFC\_TEMP, URBAN\_SFC\_TEMP, OCEAN\_SFC\_TEMP
- OCEAN\_ALB\_LW, OCEAN\_ALB\_SW, LAND\_ALB\_LW, LAND\_ALB\_SW
- OCEAN\_TEMP, OCEAN\_SFC\_ZOM, LAND\_TEMP, LAND\_WATER

設定が完了すれば、“scale-les” を実行して親ドメインの時間積分計算を行う。

親ドメインの出力ファイルを用いて子ドメインの初期値/境界値を作成する

次に計算が終わった親ドメインの history データ出力を用いて、子ドメインの初期値/境界値を作成する。実行するプログラムは、通常の初期値/境界値作成と同じ “scale-les\_init” だが、“init.child.conf” を下記のように編集する。

```

&PARAM_MKINIT_REAL
BASENAME_BOUNDARY = "boundary",
BASENAME_ORG = "history",
FILETYPE_ORG = "SCALE-LES",
NUMBER_OF_TSTEPS = 25,
BOUNDARY_UPDATE_DT = 600.DO,
/

&PARAM_NEST
USE_NESTING = .true.,
OFFLINE = .true.,
OFFLINE_PARENT_PRC_NUM_X = 4,
OFFLINE_PARENT_PRC_NUM_Y = 4,
OFFLINE_PARENT_KMAX = 35,
OFFLINE_PARENT_IMAX = 40,
OFFLINE_PARENT_JMAX = 40,
OFFLINE_PARENT_LKMAX = 5,
LATLON_CATALOGUE_FNAME = "latlon_domain_catalogue.txt",
/

```

読み込む外部入力データのファイル名を指定する“BASENAME\_ORG”は、親モデルの history.pe\*\*\*.nc ファイルを読み込むので、“history”と指定する。また、この history ファイルは SCALE-LES モデルの出力データなので、“FILETYPE\_ORG”は、“SCALE-LES”と指定する。NUMBER\_OF\_TSTEPS には、history ファイルが持つ時間ステップ数を記述する（例として 25 が記述されているだけ）。BOUNDARY\_UPDATE\_DT には、時間ステップの時間間隔を指定する（単位は秒である）。つまり、親ドメインの HISTORY\_DEFAULT\_TINTERVAL の設定項目に一致する値を指定する。この説明では、親ドメインで 600 秒としたので、ここでも 600 秒を指定する。

PARAM\_NEST の項目は、ネスティング実験のために新たに加える項目である。もともとの config ファイルには項目自体がないので、自分で config ファイルに追記する。最初の 2 つの項目によって、オフライン・ネスティング実験であることが決定される。“OFFLINE\_PARENT\_”で始まる 6 つの設定変数は、親ドメインの設定を記述する変数である。親ドメインの対応する項目を参照して正しく設定すること。この例では、親ドメインは  $4 \times 4$  の MPI プロセス数を使用し、鉛直 35 層で、水平には 1 つの MPI プロセスあたり  $40 \times 40$  の格子点を持っており、陸面モデルは 5 層モデルであることを想定している。最後の“ATLON\_CATALOGUE\_FNAME”の項目は、親ドメインを実行した時に出力したカタログファイルを指定する。

設定の編集が完了すれば、“scale-les\_init”を実行して子ドメインの初期値/境界値を作成する。

```

xxx ERROR: REQUESTED DOMAIN IS TOO MUCH BROAD
xxx -- LONGITUDINAL direction over the limit

```

実行時に上記のようなメッセージが表示されて計算が止まる場合は、子ドメインの計算領域が親ドメインの計算領域の外側に取られている部分がある。この場合は、各ドメインの大きさや領域中心の設定を見直す必要がある。

作成した初期値/境界値を用いて子ドメインの時間積分計算を行う

初期値/境界値作成が終われば、子ドメインの時間積分計算を実行する。子ドメインの実行は、通常の現実大気実験と何も変わらないので、必要なデータの PATH が正しく config ファイルに記述されていることを確認してから、“scale-les”を実行すればよい。

1 点だけ設定を忘れやすい設定項目を挙げておく。

```
&PARAM_MKINIT_REAL
  ~ 中略 ~
  BOUNDARY_UPDATE_DT = 600.DO,
/
```

“run.child.conf” の BOUNDARY\_UPDATE\_DT を、初期値/境界値作成で使用した親ドメインの history データ出力間隔に合わせておくことを忘れないようにすること。オフライン・ネスティング実験の場合、現在のところこの設定に親ドメインと子ドメイン間で不整合あっても警告やエラーメッセージが発せられないまま、時間積分計算が進み、場合によっては正常終了してしまうため注意が必要である。

多段のオフライン・ネスティング実験を行いたい場合は、ここまでの過程を繰り返せばよい。つまり、子ドメインとして時間積分計算した結果を再度、親ドメインと見立てて、さらに内側の孫ドメインの初期値/境界値作成を行なえばよい。以上でオフライン・ネスティングの実行方法の説明を終える。

#### 5.4.2 オンライン・ネスティング実験の方法

オフライン・ネスティング実験では、各ドメインの計算を逐次的に実行する必要があったが、オンライン・ネスティング実験では全てのドメインの計算を同時に実行する。現在は、親ドメインから子ドメインへのみデータ受け渡しを行う、いわゆる“1-Way ネスティング”のみをサポートしている。オンライン・ネスティング実験でサポートするネスティング段数は、最大で 10 段までである。

SCALE のオンライン・ネスティング実験は、複数のドメインを逐次的に時間積分計算を進めるのではなく、並列的に時間積分計算を行う。図 5.3 に示すイメージ図のように、与えられた MPI プロセスを分割してそれぞれのドメインに分配し、各々のドメインが独立したモデルのように計算を進める。後ほど説明するが、複数のドメインを立ち上げるために実行時には“launch.conf”という起動用の config ファイルが別途必要になる。

ここでは、最も単純な 2 段ネスティングの例を説明する。親ドメインは解像度は荒いが広領域の外側ドメインで、子ドメインは領域は狭いが高解像度の内側ドメインであることを想定する。

オンライン・ネスティング実験を行う場合は、“scale-les”のモデル本体実行前に全てのドメインについて、地形/土地利用データの作成、及び初期値/境界値データの作成を事前に行っておく必要がある。従って、親ドメインと子ドメインそれぞれについて、“pp.\*\*\*.conf”、“init.\*\*\*.conf”、そして“run.\*\*\*.conf” ファイルを事前に作成し、親ドメイン、子ドメインともに地形/土地利用データの作成、及び初期値/境界値データの作成を終えていることを想定して説明を進める。ここで説明するオンライン・ネスティング実験の設定を記述した config ファイルがチュートリアルディレクトリの下、“tutorial/real/sample/online\_nesting” に置いてあるので、説明を読み進める上で参考にしてもらいたい。

#### config ファイルの編集

まず、親ドメイン、子ドメインそれぞれに“run.\*\*\*.conf” ファイルを編集する。

run.parent.conf の編集内容

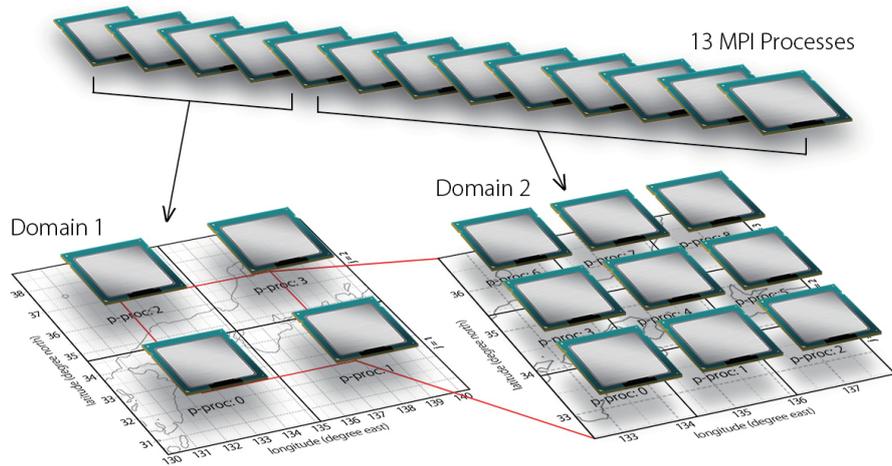


図 5.3: オンライン・ネスティング実験の MPI プロセス配分イメージ. 全部で 13 のプロセスを立ち上げ、これを適切に分配することで、Domain 1 は  $2 \times 2$  の 4-MPI 並列、Domain 2 は  $3 \times 3$  の 9-MPI 並列計算を行う。Domain 1 から Domain 2 へ MPI 通信によってデータを受け渡ししながら時間積分計算を進める。

```
&PARAM_NEST
USE_NESTING = .true.,
OFFLINE = .false.,
ONLINE_DOMAIN_NUM = 1,
ONLINE_IAM_PARENT = .true.,
ONLINE_IAM_DAUGHTER = .false.,
ONLINE_BOUNDARY_USE_QHYD = .true.,
ONLINE_AGGRESSIVE_COMM = .true.,
/
```

#### run.child.conf の編集内容

```
&PARAM_NEST
USE_NESTING = .true.,
OFFLINE = .false.,
ONLINE_DOMAIN_NUM = 2,
ONLINE_IAM_PARENT = .false.,
ONLINE_IAM_DAUGHTER = .true.,
ONLINE_BOUNDARY_USE_QHYD = .true.,
ONLINE_AGGRESSIVE_COMM = .true.,
/
```

上記の PARAM\_NEST の項目は、ネスティング実験のために新たに加える項目である。もともとの config ファイルには項目自体がないので、自分で config ファイルに追記する。最初の 2 つの項目によって、オンライン・ネスティング実験であることが決定される。“ONLINE\_”で始まる設定変数はオンライン・ネスティング実験専用の設定変数である。ONLINE\_DOMAIN\_NUM は、ドメインの ID 番号であり、外側ドメインから内側ドメインへ順番に番号を振っていく。ここでは、親ドメインは 1 番、子ドメインは 2 番と設定する。

ONLINE\_IAM\_PARENT と ONLINE\_IAM\_CHILD は各ドメインの役割を設定するパラメータである。これらの変数は、“In online nesting system, I am parent (or, I am child).” という意味で覚えれば設定を間違えることはない。少し脇道にそれるが、ここで説明している設定より複雑なものとして、図

5.2 のような 3 段ネスティング実験の場合の設定例を表 5.3 に示した。

表 5.3: 3 段ネスティング実験の設定例

ドメイン	ONLINE_DOMAIN_NUM	ONLINE_IAM_PARENT	ONLINE_IAM_CHILD
最外ドメイン	1	true	false
中間ドメイン	2	true	true
最内ドメイン	3	false	true

最外ドメインは親ドメインとしてのみ働き、最内ドメインは子ドメインとしてのみ働く。一方、中間ドメインは最外ドメインに対しては子ドメイン、最内ドメインに対しては親ドメインとして働くため両方共 “true” となる。

さて、config ファイルの編集内容の説明に戻る。ONLINE\_BOUNDARY\_USE\_QHYD は、「側面境界条件として親ドメインの凝結物の混合比を使うかどうか」を指定する設定変数である。外部入力データから側面境界条件を作成するときには通常使わないが、ネスティングの場合、ドメイン間の物理スキームの違いがなかったり、解像度もそれほど大きく離れていないため、側面境界から凝結物自体が移流して入ってくる設定も選択肢に入るだろう。側面境界付近で雲が立ちにくい問題を解決したり、親ドメインとの乖離を抑制したりする可能性がある。

最後の ONLINE\_AGGRESSIVE\_COMM はオンライン・ネスティング時のドメイン間通信に関する最適化変数である。通常は、“true” と設定して実行する。

#### launch ファイルの編集

オンライン・ネスティング実験の実行には、“run.\*\*\*.conf” の他に、起動用 config ファイル “launch.conf” が必要である。

```
$ vi launch.conf
```

などとして、適宜エディタをたちあげて新規ファイルを作成し、下記の内容を記述する。

```
&PARAM_LAUNCHER
NUM_DOMAIN = 2,
CONF_FILES = run.parent.conf,run.child.conf,
PRC_DOMAINS = 4,9,
/
```

図 5.3 のイメージを思い浮かべながら設定を確認してもらいたい。PARAM\_LAUNCHER の項目のうち、NUM\_DOMAIN = 2 が「2 つのドメインを起動する」ことを表しており、CONF\_FILES の項目に羅列されたファイル名は、各々のドメインで読み込む config ファイルを指定している。PRC\_DOMAINS は各々のドメインで使用する MPI プロセス数を指定する。PRC\_DOMAINS は、CONF\_FILES で羅列した順番で指定しなければならない。従ってこの場合、親ドメインは 4-MPI 並列、子ドメインは 9-MPI 並列で実行するように指定されている。ここで指定する MPI プロセス数は、各々の “run.\*\*\*.conf” で指定されている総 MPI プロセス数と合致させなければならない。この 2 段オンライン・ネスティング実験で使用する総 MPI プロセス数は、 $4 + 9 = 13$  プロセスとなる。

実行時には、シングルドメイン計算とは異なり、launch.conf を引数に指定し、計算全体で使用する MPI プロセス数を指定して実行する。

```
$ mpirun -n 13 ./scale-les launch.conf
```

実行にあたって注意することは、複数のドメインの計算を同時に実行するため、**ドメイン間で config ファイルに記述された出力ファイル名をドメイン毎に変更しなければならない**ことである。たとえば、“history.pe\*\*\*.nc” は、“history\_d01.pe\*\*\*.nc”、“history\_d02.pe\*\*\*.nc” といったようにドメイン毎に名前を変えながらどのドメインの出力データであるか判別がつくように config ファイルの記述を設定する。history ファイルのほかに、LOG ファイル、topo ファイル、landuse ファイル、boundary ファイル、init ファイル、restart ファイル、そして monitor ファイルの名前を変更しておく必要がある。

実行時に次のようなエラーメッセージが出力されて計算が異常終了することがある。

```
xxx region of daughter domain is larger than that of parent: SW search
```

```
xxx region of daughter domain is larger than that of parent: NE search
```

これは、子ドメインで設定された計算領域が親ドメインの計算領域よりも大きいことを意味するエラーメッセージである。“SW search” のエラーが出る場合は子ドメインの西側か南側が親ドメインの外側に出ており、“NE search” のエラーが出る場合は子ドメインの東側か北側が親ドメインの外側に出ていることを意味している。再度設定を確認し、地形・土地利用データ、および初期値/境界値作成からやり直すこと。

### MPI プロセスの分配ガイドライン

SCALE のオンライン・ネスティング実験は、図 5.3 のイメージ図で説明したように、MPI プロセスを分割し、複数のドメインに分配する。現在のところ、その分配割合はユーザーに委ねられているため、適切に MPI プロセスを分配しなければ余計な計算時間がかかってしまう。ここでは、適切に MPI プロセスを分配するためのガイドラインについて説明する。ガイドラインは、ドメイン毎に「**単位あたりの時間積分にかかる 1 プロセスあたりの演算量を揃える**」という単純なものである。ここでは、以下に示す 2 段オンライン・ネスティング実験を行う場合を想定し、ガイドラインに沿ったプロセス分配方法の例を示す。“domain 1” は外側の親ドメイン、“domain 2” は内側の子ドメインを意味する。

表 5.4: 2 段オンライン・ネスティング実験の設定想定

設定項目	domain 1	domain 2
計算領域	450 km × 450 km	200 km × 200 km
DX & DY (X,Y 同一設定)	3 km	1 km
鉛直層設定	40 層	60 層
積分時間間隔 (DT)	30 sec	10 sec
積分時間	3600 sec	3600 sec

このとき、親ドメインの水平方向の一辺の格子点数は、 $450\text{km} \div 3\text{km} = 150$  点であるので、総格子点数は  $X \times Y \times Z = 150 \times 150 \times 40 = 900,000$  点である。一方、子ドメインの水平方向の一辺の格子点数は、 $200\text{km} \div 1\text{km} = 200$  点であるので、総格子点数は  $200 \times 200 \times 60 = 2,400,000$  点である。1 つの時間ステップの積分を行うのにこれだけの格子点について計算を行わなければならない。

積分時間間隔は格子間隔に依存するためにドメイン毎に異なる。この例では、domain 1 は 30 sec だが、domain 2 は 10 sec であり、3 倍の差がある。したがって、同じ 30 sec という積分時間に対し

て domain 2 は 3 倍多くの時間ステップ、つまり 3 倍の演算量を要する。これらを考慮して、簡単なドメイン間の演算量比率 (Computation Rate) の指標を考えると下記の式で表される。

$$ComputationRate = \frac{Xgrd_{child} \times Ygrd_{child} \times Zgrd_{child} \times Ustep_{child}}{Xgrd_{parent} \times Ygrd_{parent} \times Zgrd_{parent} \times Ustep_{parent}}$$

ここで、Xgrd、Ygrd、Zgrd はそれぞれ X 方向、Y 方向、Z 方向の格子点数を表し、Ustep は単位時間積分に必要な時間ステップ数を表す。ここでの例をこの式に当てはまると、演算量比率は  $(2,400,000 \times 3) \div (900,000 \times 1) = 8$  であることがわかる。おおよそ、この割合にしたがって MPI プロセスをドメイン毎に分配すればよい。例えば domain 1 は 4 プロセス、domain 2 は 32 プロセスを使用し、全体で 36 プロセスを使用する設定が考えられる。この場合、例えば次のように設定することができる。

表 5.5: 2 段オンライン・ネスティング実験の MPI プロセス設定例

設定項目	domain 1	domain 2
MPI プロセス (X × Y)	2 × 2	4 × 8
水平格子点数 (IMAX × JMAX)	75 × 75	50 × 25

X 方向と Y 方向に分配するプロセス数には任意性が残るが、この例の domain 2 のように X と Y で大きくプロセス数が異なる場合には、X 方向の格子点数 (IMAX) の値が大きくなる設定を取ると計算機の演算性能を引き出しやすいと考えられる<sup>5</sup>。

この設定は一例であり、これ以外の方法で設定しても構わない。また、ここでは格子点数と積分時間間隔だけに着目して演算量比率を考えたが、実際の計算には様々な物理過程も含まれるだろうし、それらを Call する時間間隔もドメイン毎に異なるかもしれない。さらにドメイン内通信やドメイン間通信の MPI 通信にかかる時間も影響を及ぼす。SCALE におけるオンライン・ネスティングの実装において最も重要なことは、最内ドメインが時間積分を実行し続けることである。同じ設定で何度も実験を行うような場合には、上記の方法である程度の見通しをつけた上で、いくらかの微調整を行うことをおすすめする。以上でオンライン・ネスティングの実行方法の説明を終える。

### 5.4.3 子ドメインにおける地形の取り扱い

ネスティング実験を行う際、ドメイン間の格子間隔比率が大きい場合などに子ドメインのバッファ領域内で不整合が発生する可能性がある。バッファ領域内は親ドメインの計算結果を用いて一部の変数にダンピングがかかるが、地形の表現性が異なることで、子ドメインにとっては正しくない値へダンピングされる可能性がある。例えば、子ドメインでは斜面上の小さな谷として表現されている地形が、親ドメインでは格子間隔が荒く谷がなくスムーズな斜面として表現される場合が考えられる。

こういった不整合を無くすために、バッファ領域では親ドメインの地形を用い、内側領域では子ドメイン自身の地形を用いる「地形コピー」の機能が実装されている。この機能を使えば、図 5.4 に示すようにバッファ領域は完全に親ドメインに一致する地形で、内側に移る地形遷移領域内では親ドメインと子ドメインのミックス、それより内側では完全に子ドメインの地形という設定を構築することができる。以降、その設定方法と実行手順を説明する。基本的には、オフライン・ネスティングのフレームワークを利用して進める。ここで説明する地形コピーの設定を記述した “pp.d0\*.conf”

<sup>5</sup>SCALE では X 方向の Do Loop が最も内側ループになっているため、X 方向の回転数が多いとプリフェッチ機能が効果を発揮しやすく、メモリ性能へのプレッシャーが緩和される。ただし、京の場合のようにスレッド並列も併用するハイブリッド並列の場合には Y 方向の格子点数もある程度大きくしてスレッド間の演算量のインバランスを小さくする必要も出てくる。

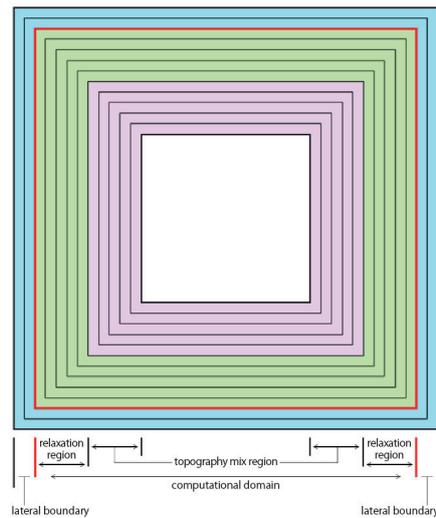


図 5.4: 地形コピーを適用した子ドメインの地形データ水平分布. 最外の水色の 2 層は側面境界で、それより内側の赤色の線で囲われた領域が計算領域である。緑色の部分がバッファ領域、桃色の部分が地形遷移領域、そして最内の白色の部分が子ドメインの地形をもつ領域である。地形遷移領域では外側から内側にかけて徐々に親ドメインの地形データから子ドメインの地形データへ遷移する。

ファイルがチュートリアルディレクトリの下、“tutorial/real/sample/online\_nesting” に置いてあるので、説明を読み進める上で参考にしてもらいたい。

まず親ドメインの “pp.d01.conf” ファイルを編集して、計算領域の大きさを子ドメインへ伝えるために緯度経度カタログファイルを出力するように設定する。具体的には、下記の記述を “pp.d01.conf” ファイルに追記する。

```
&PARAM_DOMAIN_CATALOGUE
  DOMAIN_CATALOGUE_FNAME = "latlon_domain_catalogue.txt",
  DOMAIN_CATALOGUE_OUTPUT = .true.,
/
```

その他の設定項目は通常通りで良い。編集ができれば親ドメインの地形データ作成を実行する（つまり、scale-les\_pp を実行する）。ここで、出力データは、“topo\_d01.pe\*\*\*.nc” というファイル名で保存されていると想定する。次に、子ドメインの “pp.d02.conf” ファイルを編集する。

```

&PARAM_CNVTOTOPO
  ~ 中略 ~
  CNVTOTOPO_copy_parent = .true.,
/

&PARAM_COPYTOPO
  COPYTOPO_IN_BASENAME = "topo_d01",
  COPYTOPO_ENTIRE_REGION = .false.,
  COPYTOPO_LINEAR_H = .true.,
/

&PARAM_NEST
  USE_NESTING = .true.,
  OFFLINE = .true.,
  OFFLINE_PARENT_PRC_NUM_X = 4,
  OFFLINE_PARENT_PRC_NUM_Y = 4,
  OFFLINE_PARENT_KMAX = 35,
  OFFLINE_PARENT_IMAX = 40,
  OFFLINE_PARENT_JMAX = 40,
  OFFLINE_PARENT_LKMAX = 5,
  LATLON_CATALOGUE_FNAME = "latlon_domain_catalogue.txt",
/

```

もともと config ファイルにある PARAM\_CNVTOTOPO の項目に、CNVTOTOPO\_copy\_parent = .true. という記述を加える。これは地形コピーの実行を指示するスイッチである。次の PARAM\_COPYTOPO は、地形コピーの設定項目群であり、すべて追記すること。1 つ目の COPYTOPO\_IN\_BASENAME は、親ドメインの地形データの PATH を指定する。ここでは、親ドメインの出力データは “topo\_d01.pe\*\*\*.nc” というファイル名でカレントディレクトリに保存されていると指定している。2 つ目の COPYTOPO\_ENTIRE\_REGION は、全領域でコピーするかどうかを決定するオプションである。このスイッチを true にすると、図 5.4 に示された桃色と白色の領域は無くなり、全て緑色の完全コピー領域になる。3 つ目の COPYTOPO\_LINEAR\_H は、地形遷移領域の遷移具合を調整するスイッチである。COPYTOPO\_LINEAR\_H が true だと線形プロファイルで遷移し、false だと指数関数プロファイルで遷移する。

地形遷移領域の幅は、デフォルト設定ではバッファ領域と同じ幅になる。バッファ領域の設定と同じ要領で、COPYTOPO\_TRANSITION\_DX、COPYTOPO\_TRANSITION\_DY、および COPYTOPO\_TRANSFACT の変数を使って任意の幅に設定することができる。

最後の PARAM\_NEST の項目はオフライン・ネスティング実験のフレームワークを利用するための設定項目であり、全て追記する必要がある。設定変数の詳しい説明は、5.4.1 節のオフライン・ネスティング実験の説明を参照してほしい。

config ファイルの編集が終われば、子ドメインの地形データ作成を実行する。3 つ以上のドメインがある場合は、上記の実行過程を外側ドメインから順に繰り返せばよい。

## 5.5 複数の実験を一括実行する：バルクジョブ機能

SCALE には「一括実行機能」、いわゆるバルクジョブ機能が備わっている。これは、パラメタスイープ実験、初期値アンサンブル実験や、Time Slice 気候実験など多数の実験を行う場合に便利な機能である。SCALE モデル本体の実行はもちろん、ドメインネスティング実験の場合でも利用できるし、地形・土地利用データ作成（地形コピーを利用しない場合のみ）、初期値/境界値作成、そして後処理ツールの net2g (netcdf2grads.bulk を使用) にも適用可能である。各プログラムの実行内容は異なっても構わないが、MPI 並列としての構造は共通していなければならない点に注意すること。1 つの計算事例をここでは「ジョブ」と呼ぶこととする。以下では、3 つの 2 段オンライン・

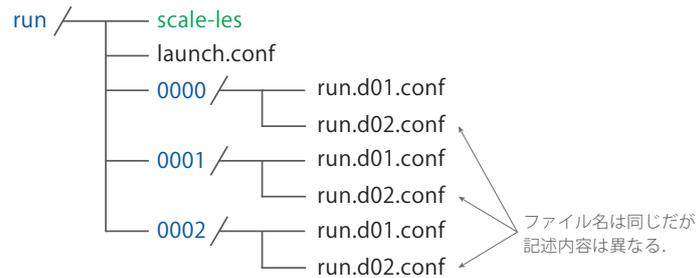


図 5.5: バルクジョブ機能を使って scale-les を実行する場合のディレクトリ構造. “0000” や “0001” はジョブ番号に対応する名前を持ったジョブディレクトリである。各ジョブディレクトリの中には、そこで実行する実験に関する config ファイルが置かれている。データパラメータテーブルなどのファイルやディレクトリの記述は割愛しているが、それらも必要に応じて適切に配置する必要がある。

ネスティング実験を一括に行う例をもとに説明する（積分期間、もしくは計算領域中心が異なっている 3 つのジョブを想定している）。

バルクジョブ実行するにあたって下記のことを事前に準備する必要がある。

- バルクジョブ用のディレクトリ構造
- 実験に必要なすべての config ファイル
- 実験に必要なすべての外部入力データ
- 実行用の launch.conf ファイル

まず、図 5.5 に示すようなディレクトリ構造を準備する。“0000” や “0001” といったディレクトリは、ジョブ番号に対応する名前を持ったジョブディレクトリである。ジョブディレクトリは必ず 4 桁の数字で、ジョブ番号はゼロから数え上げられる。これらのディレクトリの中には config ファイルが収められている。今回は 2 段オンライン・ネスティング実験を想定しているので、“run.d01.conf” と “run.d02.conf” の 2 つのファイルが準備されている。各ジョブディレクトリにある config ファイルの名前は同じにする必要があるが、内容は異なっても構わない。ただし、**ドメイン毎に使用する MPI プロセス数は全てのジョブで共通していなければならない**。config ファイル内にバルクジョブ機能用に追加する設定項目はないが、**入出力ファイルの PATH を適切に記述する必要がある**。以下にジョブ 0000 番の run.d01.conf の抜粋を示す。

```

&PARAM_IO
  IO_LOG_BASENAME = "0000/LOG_d01",
/

&PARAM_RESTART
  RESTART_OUTPUT = .true.,
  RESTART_OUT_BASENAME = "0000/restart_d01",
  RESTART_IN_BASENAME = "../init/0000/init_d01_00013046400.000",
/

&PARAM_TOPO
  TOPO_IN_BASENAME = "../pp/0000/topo_d01",
/

&PARAM_LANDUSE
  LANDUSE_IN_BASENAME = "../pp/0000/landuse_d01",
/

&PARAM_ATMOS_BOUNDARY
  ~ 中略 ~
  ATMOS_BOUNDARY_IN_BASENAME = "../init/0000/boundary_d01",
  ~ 以下略 ~
/

&PARAM_HISTORY
  HISTORY_DEFAULT_BASENAME = "0000/history_d01",
  ~ 以下略 ~
/

```

上記の config ファイルの設定例のうち、青色文字の部分は出力ファイルの指定、水色文字の部分は入力ファイルの指定である。図 5.5 を見てわかるように、実行バイナリ (scale-les) があるのは“run ディレクトリ”の下で、ジョブディレクトリも実行バイナリと同じ階層にある。従って、ジョブ 0000 番において、実行バイナリからみてデータを出力すべきディレクトリは、“0000/”の下である。そこで出力ファイル名の指定として“0000/\*\*\*”と記述している。

入力ファイルについても同様である。ここでは、run ディレクトリと同じ階層に pp ディレクトリや init ディレクトリがあり、その中にまたジョブディレクトリが作成してあって、それらの中に入力ファイルが保管されている状況を想定している。従って、run ディレクトリの下で実行される実行バイナリにとっては、“../pp/0000/\*\*\*”といった PATH になる。

バルクジョブ機能は、オンライン・ネスティング実験で利用した MPI プロセスを分割・分配する機能を使って実装されている。したがって、ジョブの起動のために“launch.conf”ファイルが必要になる。オンライン・ネスティング実験とバルクジョブ機能を併用して実行する今回のような場合も launch.conf ファイルは 1 つだけで良い。

```

&PARAM_LAUNCHER
  NUM_BULKJOB = 3,
  NUM_DOMAIN = 2,
  PRC_DOMAINS = 9,36,
  CONF_FILES = run.d01.conf,run.d02.conf,
/

```

上記がオンライン・ネスティング実験とバルクジョブ機能を併用して実行する場合の launch.conf ファイルの中身である。オンライン・ネスティング実験の場合の launch.conf ファイルに対して、NUM\_BULKJOB の設定項目を加えただけとなっている。ここで実行するジョブ数は 3 つであるので、

NUM\_BULKJOB に対して “3” と指定する。シングルドメイン実験としてバルクジョブ機能を利用する場合は、NUM\_DOMAIN = 1 と指定して、CONF\_FILES に 1 つだけ config ファイルを指定すればよい。実行時のコマンドは、

```
$ mpirun -n 135 ./scale-les launch.conf
```

となる。ここでは 1 ジョブあたり、 $9 + 36 = 45$  プロセス使用し、全体で 3 つのジョブを実行するので、総計で 135 プロセスを必要とする。

実行すると得られる LOG ファイルに、MPI プロセスを分割した時の情報が示されている。LOG ファイルを開いて最初の「SCALE ログ」のあとに下記のようなメッセージが出力される。

```
++++++ Start MPI
*** UNIVERSAL_COMM_WORLD : 0
*** total process [UNIVERSAL] : 135
*** my process ID [UNIVERSAL] : 36
*** master rank? [UNIVERSAL] : F
*** GLOBAL_COMM_WORLD : 3
*** total process [GLOBAL] : 45
*** my process ID [GLOBAL] : 36
*** master rank? [GLOBAL] : F
*** LOCAL_COMM_WORLD : 4
*** total process [LOCAL] : 9
*** my process ID [LOCAL] : 0
*** master rank? [LOCAL] : T
*** ABORT_COMM_WORLD : 0
*** master rank ID [each world] : 0
```

これらのうち、[LOCAL] と表記されている項目はドメイン内のプロセスグループ、[GLOBAL] と表記されている項目はネスティング・グループ、[MASTER] と表記された項目はジョブ・グループに関する情報である。したがって、この LOG メッセージから、当該ドメインは 12-MPI 並列で実行されており、オンライン・ネスティング実験は総計で 48 プロセス使用して実行され、バルクジョブ全体では 1488 プロセスが使用されているため、同時に 31 個のジョブが走っていたことがわかる。異常終了したときにも、この表記法に従ってメッセージが出力されるので、これを理解しているとバルクジョブ機能を使って多量に走らせている場合にも、どのプロセスがエラーを発生したのか即座に判断できる。ちなみに現在のバルクジョブ機能では、1 つのジョブがエラーを発生し、異常終了状態に入ると全てのジョブが異常終了する。以上でバルクジョブ機能の説明を終える。

## 5.6 出力変数の追加・変更

ここでは、“run.\*\*\*.conf” ファイルの中の出力変数の設定方法について説明する。history 出力ファイルへ新たに出力変数を追加するには、正式には次の 2 段階の手続きが必要である。

1. ソースファイル内の設定。対象の変数を history 出力するための設定で、この手続きにより、出力のための準備が行われる。
2. run.conf 内の設定。実際に history に出力するかどうかを指定。コンパイルし直すことなく、実験毎に変更可能。

頻繁に使用する予報変数や主要な変数については、すでに 1 の手続きは行われているため、2 の手続きだけを行えばよい。2 の手続きだけで出力可能な変数については、AppendixC にリストアップし

であるので、そちらを参照すること。これ以外の変数をユーザーが書き出したい場合は、1の手続きが必要であるがここでは割愛する。

history 出力ファイルへの変数の追加は、下記のフォーマットに従って run.conf に追記すればよい。

```
&HISTITEM ITEM = "[character]",
  BASENAME = "[character]",
  TINTERVAL= [real],
  TUNIT = "[character]",
  TAVERAGE = [logical],
  ZINTERP = [logical],
  DATATYPE = "[character]",
/
```

水色文字の部分はオプションである。[character] は任意の文字列、[real] は任意の浮動小数点値、[logical] は任意の論理変数 (.true. or .false. ) をそれぞれ指定することを意味する。デフォルト設定は、つぎに説明する&PARAM\_HISTORY の項目によって定められる。

```
&PARAM_HISTORY
  HISTORY_DEFAULT_BASENAME = "[character]",
  HISTORY_DEFAULT_TINTERVAL = [real],
  HISTORY_DEFAULT_TUNIT = "[character]",
  HISTORY_DEFAULT_TAVERAGE = [logical],
  HISTORY_DEFAULT_ZINTERP = [logical],
  HISTORY_DEFAULT_DATATYPE = "[character]",
  HISTORY_OUTPUT_STEP0 = [logical],
/
```

上記のとおり、変数毎の個別の設定群とデフォルト設定群の違いはあるが、各設定変数の意味は同じである。それらの各設定変数の説明は以下の通りである。

表 5.6: 出力ファイル設定変数の説明

設定変数	説明
ITEM	変数名。 Appendix C を参照
BASENAME	出力ファイル名 BASENAME_XXXXXX.nc となる。XXXXXX はノード番号
TINTERVAL	出力間隔
TUNIT	TINTERVAL で指定した出力間隔の単位
TAVERAGE	.false.=瞬間値、.true.=平均値として出力する；平均値の場合、出力タイミングの直前の TINTERVAL 間の平均値
DATATYPE	出力値の型；"REAL4","REAL8" など
ZINTERP	.false.=モデル面、.true.=Z 面 (FZ 面?、CZ 面?) の値として出力
HISTORY_OUTPUT_STEP0	.false.=初期時刻 (t=0) の値を出力、.true.=出力しない；&PARAM_HISTORY でのみ有効

&PARAM\_HISTORY の HISTORY\_DEFAULT\_TINTERVAL は、デフォルトの history 出力間隔を設定する変数だが、出力アイテム毎に出力間隔を設定することもできる。一部の変数で history 出力間隔を変更したい場合の記述例を次に挙げる。

```

&PARAM_HISTORY
  HISTORY_DEFAULT_BASENAME = "history_d03",
  HISTORY_DEFAULT_TINTERVAL = 3600.D0,
  HISTORY_DEFAULT_TUNIT = "SEC",
  HISTORY_DEFAULT_TAVERAGE = .false.,
  HISTORY_DEFAULT_DATATYPE = "REAL4",
  HISTORY_DEFAULT_ZINTERP = .false.,
  HISTORY_OUTPUT_STEPO = .true.,
/

&HISTITEM item="T" /
&HISTITEM item="PRES" /
&HISTITEM item="U" /
&HISTITEM item="V" /
  ~ 中略 ~

&HISTITEM item="RAIN", taverage=.true., tinterval=600.D0 /

```

上記のように設定すると、“RAIN”は600秒の出力間隔で、それ以外は3600秒の出力間隔となる。以上で、出力変数の追加・変更についての説明を終える。

## 5.7 Postprocess : netcdf2grads

本節では、SCALEの領域分割された history データ (history.\*\*\*.nc) を GrADS で読み込める形式に変換する後処理ツール “net2g” の実行方法について説明する。net2g のインストール方法については、2.4 節を参照すること。net2g の実行において注意すべき点は下記のとおりである。

- 使用する MPI プロセス数は、SCALE の本体の実行で使用した MPI プロセス数の約数でなければならない。
- history 出力形式は、“HIST\_BND = .false.” でなければならない (今後対応予定)。
- 2次元データと3次元データは同時に変換できない (今後対応予定)。
- 変換できるデータは history データのみ (今後対応予定)。

“HIST\_BND” の対応、2次元データと3次元データの同時実行、そして history データ以外のデータへの対応は今後早急に進める予定ではあるが、どうしても gpview 等の他のツールが使えず、データの描画に支障が出る場合は、「旧型 netcdf2grads」を使用することができる。旧型 netcdf2grads は、シングルプロセス実行しかできないが、上記の制限はない。旧型 netcdf2grads の概要については本節の最後に紹介する。

その他に注意すべき点は、net2g 実行時の MPI 並列数である。net2g は MPI 並列実行できるが、プログラム中の作業のほとんどが history ファイルからのデータ読み込みである。したがって、並列ファイルシステム上でない限り、一般に単一のディスク、もしくは単一のディスクアレイからデータを読み込む速度は、MPI プロセス数を増やしても向上しない。むしろ読み込み要求が立て込んで、作業効率が下がり結果として読み込み速度が落ちることもある。したがって、新しい環境で net2g を実行する場合、少し MPI プロセス数を変えて実行してみて最適な MPI プロセス数を予め知っておくことも必要である。

net2g の実行方法は、MPI 並列版としてコンパイルした場合は、

```
$ mpirun -n [procs] ./net2g net2g.conf
```

として実行する。[procs] には任意の MPI プロセス数を指定する。最後の “net2g.conf” は net2g の実行設定が記述された config ファイルである。一方、シングルプロセス版としてコンパイルした場合は mpi コマンドを使わずに、

```
$ ./net2g net2g.conf
```

とする。次に config ファイルの記述方法と設定内容について説明する。ここでは主要な設定項目を取り上げて説明する。ここで取り上げなかったオプションについては、“scale/scale-les/util/netcdf2grads\_h/” のディレクトリに入っている “net2g.all.conf” のサンプル config ファイルを参照してもらいたい。

### config ファイルサンプル：3次元データの変換

```
&LOGOUT
  LOG_BASENAME = "LOG_d01",
  LOG_ALL_OUTPUT = .false.,
/

&INFO
  START_TSTEP = 1,
  END_TSTEP = 121,
  INC_TSTEP = 1,
  DOMAIN_NUM = 1,
  CONFFILE = "../run/run.d01.conf",
  IDIR = "../run",
  Z_LEV_TYPE = "plev",
/

&VARI
  VNAME = "PT","U","V","W",
  TARGET_ZLEV = 850, 500, 200,
/
```

上記はある domain 1 のデータの 3 次元変数を、指定した気圧高度面へ変換して出力する場合の config ファイルの全記述内容の例を示している。各々の設定項目は次の説明のとおりである。

- LOGOUT の項目（この項目は必須ではない）
  - LOG\_BASENAME：デフォルトの LOG ファイル名 “LOG” を変更したいときに指定する。
  - LOG\_ALL\_OUTPUT：プロセス番号 0 番以外も LOG ファイルを出力させたい場合に “true” にする。デフォルト値は “false” である。
- INFO の項目
  - START\_TSTEP：変換を開始する netCDF データの最初の Time Step を指定する。最初のいくつかのステップを飛ばして変換したい場合に任意の値を指定する。デフォルト値は 1 である。
  - END\_TSTEP：変換を終了する netCDF データの Time Step を指定する。必ず指定する。
  - INC\_TSTEP：netCDF データを読み込む時の時間軸の増加量（インクリメント）を指定する。例えば 1 つ飛ばして変換したい場合に任意の値を指定する。デフォルト値は 1 である。
  - DOMAIN\_NUM：ドメイン番号を指定する。デフォルト値は 1 である。
  - CONFFILE：SCALE 本体を実行したときの run.\*\*\*.conf ファイルの PATH を指定する（ファイル名を含む）。

- IDIR : SCALE 本体の history 出力ファイルの PATH を指定する。
- Z\_LEV\_TYPE : 鉛直方向のデータ変換の種類を指定する。“original” は history データのまま出力、“plev” は気圧高度面に変換して出力、そして“zlev” は高度面データに変換する。“anal” を指定すると別途指定の簡易解析を行なって出力する(次項で説明)。デフォルト値は“plev”である。

- VARI の項目

- VNAME : 変換したい変数名を指定する。デフォルトでは、“PT”、“PRES”、“U”、“V”、“W”、“QHYD” が指定される。
- TARGET\_ZLEV : Z\_LEV\_TYPE に応じた変換高度を指定する。plev の場合は“hPa”、zlev の場合は“m”、そして original の場合は格子点番号で指定する。デフォルトでは、1000hPa、975hPa、950hPa、925hPa、900hPa、850hPa、800hPa、700hPa、600hPa、500hPa、400hPa、300hPa、250hPa、200hPa の 14 層が指定される。

### config ファイルサンプル : 3 次元データの変換 : 鉛直積算値を出す

```
&INFO
  ~ 中略 ~
  Z_LEV_TYPE = "anal",
/

&ANAL
  ANALYSIS = "sum"
/

&VARI
  VNAME = "QC","QI","QG"
/
```

上記は config ファイル記述例の抜粋である。Z\_LEV\_TYPE を“anal”と指定すると ANAL 項目を指定できるようになる。これは、鉛直次元に大して何からの簡易解析を施してデータ出力する機能である。この場合は、変換後のデータは必ず水平 2 次元データとなるため、VARI の項目の TARGET\_ZLEV は指定できない。下記に説明する以外の設定変数については、先の 3 次元変数の変換の場合と同じである。

- ANAL の項目

- ANALYSIS : 鉛直次元の簡易解析の種類を指定する。“max”を指定すると鉛直カラム中の最大値、“min”を指定すると最小値、“sum”を指定すると鉛直カラム積算値、そして“ave”を指定すると鉛直カラム平均値を算出する。デフォルト値は“ave”である。

### config ファイルサンプル : 2 次元データの変換

```
&INFO
  ~ 中略 ~
  Z_LEV_TYPE = "original",
/

&VARI
  VNAME = "T2","MSLP","PREC"
/
```

上記は config ファイル記述例の抜粋である。2次元データを変換する場合は、Z\_LEV\_TYPE を必ず “original” と指定する。また、2次元データなので VARI の項目の TARGET\_ZLEV は指定できない。その他の設定項目は3次元データ変換の場合と変わらない。

config ファイルサンプル：特殊な時間軸を持つデータの変換

```
&EXTRA
  EXTRA_TINTERVAL = 600.0,
  EXTRA_TUNIT = "SEC",
/

&VARI
  VNAME = "RAIN",
/
```

上記は config ファイル記述例の抜粋である。5.6 節の最後で説明した出力アイテム毎の設定で、HISTITEM の tinterval を設定し、一部のデータ変数が HISTORY\_DEFAULT\_TINTERVAL に従わない場合は、net2g では EXTRA の設定項目を上記のように設定することで対応できる。ここでは、5.6 節の最後で説明した設定に合わせて、2次元データの “RAIN” だけが 600 秒間隔で出力されていた場合の config ファイル設定例を示している。

### 5.7.1 スーパーコンピュータ「京」での実行方法

スーパーコンピュータ「京」などの大型計算機で並列計算を行った場合、出力ファイルの数が多く、それぞれのファイルのデータ容量も大きい。そのような場合には、手元のマシンへ history データをコピーするだけのディスク空き容量がなかったり、並列ファイルシステムではないために後処理に膨大な時間がかかり、現実的な時間スケールで解析が行えない可能性がある。こういった場合には、SCALE 本体の計算を行ったスーパーコンピュータで後処理も行ってしまうことをおすすめする。この目的のためにここでは、スーパーコンピュータ「京」で net2g を使用方法を簡単に説明する。

スーパーコンピュータ「京」には、JOB のサイズや実行形態に応じたいくつかの JOB グループがあるが、ここでは、もっとも簡単に実行を行える “micro JOB” を対象に実行方法を説明する。micro JOB では実行バイナリが “/scratch” ディレクトリの下に存在する必要があることに注意されたい。

/scratch/GROUP/USER/の下に、作業ディレクトリを用意する。そこに、

```
net2g          : copy executive file
net2g.conf     : copy configure file
output/       : link to directory with scale history files
bindata/      : create directory for grads file output
job.sh        : job script for K (option)
```

を用意する。net2g.conf の設定と job.sh の設定をする。net2g.conf の INFO 項目の IDIR と ODIR は、上記設定に合わせて適切に記述しなければならない。使用するノード数は、SCALE 本体の計算に使用したノード数の約数である必要がある。job.sh の中身は下記のような内容を記述する。各パラメタの詳細等は、スーパーコンピュータ「京」のユーザズマニュアル等を参照されたい。準備が整えば、job.sh を使って pjsub コマンドにて JOB を投入すればよい。実行時間が 30 分の制限時間に間に合わない場合は、一度に変換する変数の数を減らすか、鉛直層数を減らして実行する。もしくは

は Lustre の striping count の調整等を行なってもよい。

#### job.sh の記述例

```
#!/bin/bash -x
#PJM --rsc-list "rscgrp=micro"
#PJM --rsc-list "node=12"
#PJM --rsc-list "elapse=00:30:00"
#PJM -j
#PJM -s
#
. /work/system/Env_base
#
export PARALLEL=8
export OMP_NUM_THREADS=8
#
mpiexec -n 12 ./net2g n2g_d01-3d.conf
```

### 5.7.2 バルクジョブ対応版の使用方法

5.5 節で説明した「複数の実験を一括実行するバルクジョブ機能」を用いて SCALE を走らせた場合は、バルクジョブ対応版の net2g を利用するのが便利である。コンパイルは、通常版の net2g と同じである。“scale/scale-les/util/netcdf2grads\_bulk”の下で 2.4 節で説明したとおりの方法でコンパイルすれば、バルクジョブ対応版の net2g が生成される。

基本的な使用方法や制限事項も通常版の net2g と同じである。net2g に渡す config ファイルなどの記述も通常版と同じように記述すればよい。ただし、実行にあたっては“launch.conf”が必要になることと、5.5 節で説明したバルクジョブ実行時のディレクトリ構造を準備する必要がある。SCALE 本体をバルクジョブ機能で実行した場合にはディレクトリ構造はすでに準備されているため新たに用意する必要はない。net2g に渡す config ファイルだけ、各バルク番号のディレクトリ下に設置すればよい。以下に、launch.conf ファイルの記述例を挙げておく。

```
&PARAM_LAUNCHER
NUM_BULKJOB = 31,
NUM_DOMAIN = 2,
PRC_DOMAINS = 12,36,
CONF_FILES = net2g.3d.d01.conf,net2g.3d.d02.conf,
/
```

この例の場合、一度に 31 個のジョブを実行している。また 1 つのジョブは 2 段オンライン・ネスティング実験となっており、net2g の実行にあたっては domain 1 は 12-MPI 並列、domain 2 は 36-MPI 並列で実行される。ここで指定する MPI プロセス数は、SCALE 本体の実行時に使用した MPI プロセス数の約数でなければならない。

それぞれのドメインについて実行する net2g の config ファイルは、それぞれ“net2g.3d.d01.conf”と“net2g.3d.d02.conf”と指定さえている。この config ファイルは 31 個のバルク番号ディレクトリの中に収められてことを想定している。

この例では、1 つのジョブあたり、 $12 + 36 = 48$  プロセスを使用し、全体で 31 ジョブあるので総計で 1488 プロセスを必要とする。下記のコマンドのように実行する。

```
$ mpirun -n 1488 ./net2g launch.conf
```

### 5.7.3 旧型 netcdf2grads の使用方法

ここでは、シングルプロセス実行しかできないが、“HIST\_BND”の対応、そして history データ以外のデータにも対応する「旧型 netcdf2grads」の実行方法概略について説明する。このプログラムは、並列版/シングル版 net2g の整備が済み次第、サポート外となるため注意して欲しい。

netcdf2grads のソースファイルは scale/scale-les/util/netcdf2grads/にある。SCALE 本体とは独立なので、ディレクトリを任意の場所に移動/コピーして使用することが可能である。

#### コンパイル

##### Intel compiler

```
$ ifort -convert big_endian -assume byterecl -I${NETCDF4}/include  
-L${NETCDF4}/lib -lnetcdf -lnetcdf make_grads_file.f90 -o convine
```

##### gfortran

```
$ gfortran -frecord-marker=4 --convert=big-endian -I${NETCDF4}/include  
-L${NETCDF4}/lib -lnetcdf -lnetcdf make_grads_file1.f90 -o convine
```

#### 使用方法

実行時に、インタラクティブモードかサイレントモードかを選択することが出来る。

```
Interactive mode : './convine -i'  
Silent mode      : './convine -s'
```

#### インタラクティブモード

```
$ cd netcdf2grads/  
$ ./convine -i
```

と実行すると、下記のメッセージが出るので、指示通りに必要なファイルのパスを打ち込む。

```

path to configure file for run with the quotation mark
'${path to directory of configure file}/run.conf' <- configure ファイルへのパス
path to directory of history files with the quotation mark
'${path to directory of history files}/' <- SCALE-LES の出力ファイルのパス
path to directory of output files with the quotation mark
'./grads/' <- grads ファイルの出力先
start time of convert data
1 <- 任意の番号の時間から変換可能
end time of convert data
10 <- 変換を修了する時間
Input number of variable
0 -> all variable output from model
X <- 変換したい変数の数
Input variable
VARIABLE(PREC など) <- 変換したい変数名

```

青色文字の行は、ユーザーが入力する行である。うまく実行されれば、指定した出力先に ctl ファイルと grd ファイルが作成される。

#### サイレントモード

サイレントモードの場合は、あらかじめ、namelist.in に必要な情報を書き入れておく。

```

$ cd netcdf2grads/
$ ./convine -s

```

と実行すれば変換が始まる。

## 関連図書

- T. Maeda Abe, T. and M. Sato. Model checking stencil computations written in a partitioned global address space language. *In Proceedings of the 18th Workshop on High-Level Parallel Programming Models and Supportive Environments, Cambridge.*, pages 365–374, 2013.
- A. C. M. Beljaars and A. A. M. Holtslag. Flux parameterization over land surfaces for atmospheric models. *J. Appl. Meteor.*, 30:327341, 1991.
- S. Kentaroh, T. Nakajima, T. Y. Nakajima, and A. P. Khain. Aerosol effects of the condensation process on a convective cloud simulation. *Journal of the Atmospheric Sciences*, 67:1126–1141, 2010.
- E. Kessler. On the distribution and continuity of water substance in atmospheric circulation. *Meteorological Monograph*, 10:1–84, 1969.
- H. Kusaka, H. Kondo, Y. Kikegawa, and F. Kimura. A simple single-layer urban canopy model for atmospheric models: comparison with multi-layer and slab models. *Boundary-Layer Meteorol.*, 101:329–358, 2001.
- D. K. Lilly. On the numerical simulation of buoyant convection. *Tellus*, 14:148171, 1962.
- G. L. Mellor and T. Yamada. Development of a turbulence closure model for geophysical fluid problems. *Rev. Geophys. Space Phys.*, 20:851875, 1982.
- M. Nakanishi and H. Niino. An improved mellor-yamada level-3 model with condensation physics: Its design and verification. *Bound-Lay. Meteorol.*, 112:1–31, 2004.
- Y. Sato Y. Miyamoto Nishizawa S., H. Yashiro and H. Tomita. Influence of grid aspect ratio on planetary boundary layer turbulence in large-eddy simulations. *Geosci. Model Dev.*, 8:3393–3419, 2015. doi: 10.5194/gmd-8-3393-2015.
- H. Yashiro Y. Miyamoto Sato Y., S. Nishizawa and H. Tomita. Potential of retrieving shallow-cloud life cycle from future generation satellite observations through cloud evolution diagrams: A suggestion from a large eddy simulation. *SOLA*, 10:10–14, 2014. doi: sola.2014-003.
- H. Yashiro Y. Miyamoto Y. Kajikawa Sato Y., S. Nishizawa and H. Tomita. Impacts of cloud microphysics on trade wind cumulus: which cloud microphysics processes contribute to the diversity in a large eddy simulation? *Progress in Earth and Planetary Science*, 2:23, 2015a. doi: 10.1186/s40645-015-0053-6.
- S. Nishizawa H. Yashiro Y. Kajikawa R. Yoshida T. Yamaura Sato Y., Y. Miyamoto and H. Tomita. The horizontal distance of each cumulus and broadening distance of clouds determines shallow cloud cover. *SOLA*, 11:75–79, 2015b. doi: sola.2015-019.

- T. Seiki and T. Nakajima. Aerosol effects of the condensation process on a convective cloud simulation. *Journal of the Atmospheric Sciences*, 71:833–853, 2014.
- M. Sekiguchi and T. Nakajima. A k-distribution-based radiation code and its computational optimization for an atmospheric general circulation model. *J. of Quantitative Spectroscopy and Radiative Transfer*, 109:27792793, 2008.
- S. Shima, K. Kusano, A. Kawano, T. Sugiyama, and S. Kawahara. The super-droplet method for the numerical simulation of clouds and precipitation: a particle-based and probabilistic microphysics model coupled with a non-hydrostatic model. *Quarterly Journal of the Royal Meteorological Society*, 135(642).
- J. Smagorinsky. General circulation experiments with the primitive equations. *Mon. Weather Rev.*, 91:99164, 1963.
- Team SCALE. *The detail formulation of SCALE-LES*. RIKEN/AICS, 2015.
- H. Tomita. New microphysical schemes with five and six categories by diagnostic generation of cloud ice (special issue; the international workshop on high-resolution and cloud modeling, 2006). *Journal of the Meteorological Society of Japan. Ser. II*, 86:121–142, 2008. URL <http://ci.nii.ac.jp/naid/110007006135/>.
- D. K. Wilson. An alternative function for the wind and temperature gradients in unstable surface layers. *Bound-Lay. Meteorol.*, 99:151158, 2001.

## 付録A ライブラリ環境のインストール

ここでは、SCALE のインストールに必要なライブラリ環境のインストール方法について説明する。Linux をインストール後、各種プログラムのインストールにはコマンドライン端末を使う。本書で説明するライブラリ環境のインストールにあたっては、root 権限が必要になる。したがって、想定する環境は、ユーザーが root 権限を所持しているサーバーや、自前のデスクトップマシンといった環境である。別途サーバー管理者が存在し、root 権限を取得できない場合等は、必要な環境条件が整っているか問い合わせること。

また本節では、HDF5, NetCDF, MPI について GNU compiler でコンパイルされたライブラリの説明を行う。GNU compiler 以外の Intel compiler などを利用する場合は、各自でインストール方法を調べてインストールすること。

ここでインストールするライブラリ環境は、主に下記の 4 点である。

- GNU C/C++, fortran compiler
- HDF5 Library (<https://www.hdfgroup.org/HDF5/>)
- NetCDF Library (<http://www.unidata.ucar.edu/software/netcdf/>)
- Message Passing Interface (MPI) Library (openMPI 版、<http://www.open-mpi.org/>)

これらのインストール方法について、本書では下記の 5 種類の Operating System (OS) について説明する。

- Linux CentOS 6.6 x86-64
- Linux CentOS 7.1 x86-64
- Linux openSUSE 13.2 x86-64
- Apple Mac OS X 10.10 Yosemite
- スーパーコンピュータ「京」

他の OS ディストリビューション（下記参照）でも SCALE を利用可能だが、本書でサポートするのは上記の範囲とする。

動作確認済みの他の OS ディストリビューション

- Linux SUSE Enterprise Linux 11.1, 11.3 x86-64
- Linux Vine Linux 6.3 x86-64
- Linux Fedora 16 x86-64

## A.1 インストール方法 (Linux - CentOS 6.6 編)

以下の説明で使った環境は次のとおりである。

- CPU: Intel Core i5 2410M (sandybridge)
- Memory: DDR3-1333 4GB
- OS: CentOS 6.6 (kernel: 2.6.32-504.23.4.el6.x86\_64 )  
\* インストール時、"日本語"、"Desktop"、"Kdump 有り"を選択

### ライブラリのインストール

CentOS 6.6 では、一部のライブラリをエンタープライズ Linux 用の拡張パッケージ (EPEL) リポジトリからインストールする。そこで、はじめに EPEL リポジトリをシステムにインストールし登録する。CentOS 6.6 では、ソフトウェアのインストールに"yum" コマンドを利用する。すべての作業を行うまえに、下記のコマンドにてパッケージをアップデートしておくことをおすすめする。

```
# yum update
```

ルート権限で、下記のコマンドを実行することでリポジトリの登録が可能である。

```
# yum install epel-release
```

実行時のコマンドラインの様子は以下ようになる。インストール対象がリストされるので、確認して"y"をタイプして先へ進める。

```
読み込んだプラグイン:fastestmirror, refresh-packagekit, security
インストール処理の設定をしています
```

```
Loading mirror speeds from cached hostfile
```

```
* base: ftp.***.***.jp
```

```
* extras: ftp.***.***.jp
```

```
* updates: ftp.***.***.jp
```

```
依存性の解決をしています
```

```
- トランザクションの確認を実行しています。
```

```
- パッケージ epel-release.noarch 0:7-5 を インストール
```

```
- 依存性解決を終了しました。
```

```
依存性を解決しました
```

```
=====  
Package アーキテクチャー バージョン リポジトリ 容量
```

```
=====  
インストール中:
```

```
epel-release noarch 6-8 extras 14 k
```

```
トランザクションの要約
```

```
=====  
インストール 1 パッケージ
```

```
総ダウンロード容量: 14 k
```

```
インストール容量: 24 k
```

```
Is this ok (y/N): y
```

```
パッケージをダウンロードしています: epel-release-6-8.noarch.rpm 14 kB 00:00
```

```
rpm_check.debug を実行しています
```

```
トランザクションのテストを実行しています
```

```
トランザクションのテストを成功しました
```

```
トランザクションを実行しています
```

```
インストールしています : epel-release-6-8.noarch 1/1
```

```
Verifying : epel-release-6-8.noarch 1/1
```

```
インストール:
```

```
epel-release.noarch 0:6-8
```

```
完了しました!
```

\*この時点で、yum によるインストールに失敗する場合は、プロキシ設定等を含めた通信環境、yum リポジトリの登録状況等を再確認すること。

yum のグループインストール機能を用いて、開発ツール（ここでの対象は主に GNU compiler と make システム）をまとめてインストールする。

```
# yum groupinstall "development tools"
```

つづいて、グループインストールではインストールされないライブラリを個別に追加する。

```
# yum install zlib-devel
```

```
# yum install hdf5-devel hdf5-static
```

```
# yum install netcdf-devel netcdf-static
```

```
# yum install openmpi-devel
```

```
# yum install lapack lapack-devel
```

```
# yum install wgrib wgrib2
```

\* Lapack は、GNU compiler で SCALE をコンパイルする際にデフォルトで Lapack を利用する設定であるため、依存関係のあるライブラリとしてインストールが必要になる。

- \* wgrib、wgrib2 は、第 4 章：Tutorial: Real case で外部入力データのプレ処理を行うために使用する。
- \* ”yum -y install package name”として実行することで、インストール前の再確認をスキップできる。

## 環境変数の設定

ローカルシステムで MPI 並列プログラムを実行するために、OpenMPI ライブラリの環境変数設定を行う。ユーザ権限に移動して .bashrc をエディタで開き、

```
$ vi ~/.bashrc
```

下記をファイルの最後に追加して、環境変数の設定を記述する。

```
// ----- Add to end of the file -----  
# OpenMPI  
export MPI="/usr/lib64/openmpi"  
export PATH="$PATH:$MPI/bin"  
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$MPI/lib"
```

編集が終わったら、環境設定を有効にする。

```
$ . ~/.bashrc
```

## A.2 インストール方法 (Linux - CentOS 7.1 編)

以下の説明で使用した環境は次のとおりである。

- CPU: Intel Core i5 2410M (sandybridge)
- Memory: DDR3-1333 4GB
- OS: CentOS 7.1 (kernel: 3.10.0-229.7.2.el7.x86\_64)  
\* インストール時、”日本語”、”Gnome デスクトップ”、”Kdump 有り”を選択

### ライブラリのインストール

CentOS 7.1 では、一部のライブラリをエンタープライズ Linux 用の拡張パッケージ (EPEL) リポジトリからインストールする。そこで、はじめに EPEL リポジトリをシステムにインストールし登録する。CentOS 7.1 では、ソフトウェアのインストールに ”yum” コマンドを利用する。すべての作業を行うまえに、下記のコマンドにてパッケージをアップデートしておくことをおすすめする。

```
# yum update
```

ルート権限で、下記のコマンドを実行することでリポジトリの登録が可能である。

```
# yum install epel-release
```

実行時のコマンドラインの様子は以下ようになる。インストール対象がリストされるので、確認して”y”をタイプして先へ進める。

```
読み込んだプラグイン:fastestmirror, langpacks
base 3.6 kB 00:00
extras 3.4 kB 00:00
updates 3.4 kB 00:00
Loading mirror speeds from cached hostfile
* base: ftp.***.***.jp
* extras: ftp.***.***.jp
* updates: ftp.***.***.jp
依存性の解決をしています
- トランザクションの確認を実行しています。
- パッケージ epel-release.noarch 0:7-5 をインストール
- 依存性解決を終了しました。

依存性を解決しました

=====
Package アーキテクチャー バージョン リポジトリ 容量
=====
インストール中:
epel-release noarch 7-5 extras 14 k

トランザクションの要約
=====
インストール 1 パッケージ

総ダウンロード容量: 14 k
インストール容量: 24 k
Is this ok (y/d/N): y
Downloading packages:
extras/7/x86_64/prestodelta 7.6 kB 00:00
epel-release-7-5.noarch.rpm 14 kB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
インストール中 : epel-release-7-5.noarch 1/1
検証中 : epel-release-7-5.noarch 1/1

インストール:
epel-release.noarch 0:7-5

完了しました!
```

\*この時点で、yumによるインストールに失敗する場合は、プロキシ設定等を含めた通信環境、yumリポジトリの登録状況等を再確認すること。

yumのグループインストール機能を用いて、開発ツール(ここでの対象は主にGNU compilerとmakeシステム)をまとめてインストールする。

```
# yum groupinstall "development tools"
```

つづいて、グループインストールではインストールされないライブラリを個別に追加する。

```
# yum install hdf5-devel hdf5-static
# yum install netcdf-devel netcdf-static
```

```
# yum install netcdf-fortran-devel
# yum install openmpi-devel
# yum install lapack lapack-devel
# yum install wgrib wgrib2
```

\* fortran 用のモジュールファイルは別パッケージになっている。"netcdf-fortran-devel" のインストールを忘れないこと。

\* Lapack は、GNU compiler で SCALE をコンパイルする際にデフォルトで Lapack を利用する設定であるため、依存関係のあるライブラリとしてインストールが必要になる。

\* wgrib、wgrib2 は、第 4 章 : Tutorial: Real case で外部入力データのプレ処理を行うために使用する。

\* "yum -y install package name" として実行することで、インストール前の再確認をスキップできる。

## 環境変数の設定

ローカルシステムで MPI 並列プログラムを実行するために、OpenMPI ライブラリの環境変数設定を行う。ユーザ権限に移動して .bashrc をエディタで開き、

```
$ vi ~/.bashrc
```

下記をファイルの最後に追加して、環境変数の設定を記述する。

```
// ----- Add to end of the file -----
# OpenMPI
export MPI="/usr/lib64/openmpi"
export PATH="$PATH:$MPI/bin"
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$MPI/lib"
```

編集が終わったら、環境設定を有効にする。

```
$ . ~/.bashrc
```

## A.3 インストール方法 (Linux - openSUSE 13.2 編)

以下の説明で使用した環境は次のとおりである。

- CPU: Intel Core i5 2410M (sandybridge)
- Memory: DDR3-1333 4GB
- OS: openSUSE 13.2 (kernel: 3.16.7-21-desktop x86\_64)
  - \* インストール時、"日本語"、"Gnome Desktop" を選択

## ライブラリのインストール

openSUSE 13.2 では、一部のライブラリを外部リポジトリ (ocephaf's Home Project; <https://build.opensuse.org/project/show/home:ocephaf>) からインストールする。このため、まず home\_ocephaf リポジトリをシステムにインストールし登録する。このリポジトリには、grads、ncview、GMT、ncl、そして cdo といったツール群も含まれており便利である。

openSUSE 13.2 では、ソフトウェアのインストールに "zypper" コマンドを利用する。openSUSE では一般にユーザーが root ユーザーにスイッチすることを推奨しておらず、デフォルトのまま OS をインストールすると "su" コマンドによって root ユーザーにスイッチすることはできないので、"sudo" コマンドを利用してインストール作業を行う。すべての作業を行うまえに、下記のコマンドにてパッケージをアップデートしておくことをおすすめする。

```
# sudo zypper update
```

下記のコマンドを実行することでリポジトリの登録が可能である。

```
$ sudo zypper ar \\  
http://download.opensuse.org/repositories/home:/ocephaf/openSUSE_13.2/ \\  
home_ocephaf
```

\* 上記コマンド中の "\\" は、組版上の改行であることを意味する。実際は改行も "\\" の記述も必要ない。実行時のコマンドラインの様子は以下ようになる。

```
リポジトリ 'home_ocephaf' を追加しています ..... 完了  
リポジトリ 'home_ocephaf' を正常に追加しました  
有効: はい (Y)  
自動更新: いいえ (N)  
GPG チェック: はい (Y)  
URI: http://download.opensuse.org/repositories/home:/ocephaf/openSUSE_13.2/
```

\* この時点で、zypper によるインストールに失敗する場合は、プロキシ設定等を含めた通信環境、zypper リポジトリの登録状況等を再確認すること。

zypper のパターンインストール機能を用いて、基本開発ツール (ここでの対象は主に GNU compiler と make システム) をまとめてインストールする。

```
$ sudo zypper install --type pattern devel_basis
```

home\_ocephaf リポジトリを登録して最初のインストールの場合、下記のようにパッケージの署名鍵の信頼について問われることがある。"a" の「ずっと信頼」を選択して作業を進める。その後、インストール対象がリストされるので、確認して "y" をタイプして先へ進める。

```
鍵を拒否しますか (R)? 一時的に信頼しますか (T)?  
それとも今後ずっと信頼しますか (A)? [r/t/a/? 全てのオプションを表示] (r): a
```

つづいて、devel\_basis パッケージに含まれないライブラリを個別に追加する。

```
$ sudo zypper install gcc-fortran  
$ sudo zypper install hdf5-devel hdf5-devel-static  
$ sudo zypper install netcdf-devel netcdf-devel-static  
$ sudo zypper install netcdf-fortran-devel netcdf-fortran-static  
$ sudo zypper install lapack-devel lapack-devel-static
```

```
$ sudo zypper install openmpi-devel openmpi-devel-static
$ sudo zypper install wgrib wgrib2
```

\* Lapack は、GNU compiler で SCALE をコンパイルする際にデフォルトで Lapack を利用する設定であるため、依存関係のあるライブラリとしてインストールが必要になる。

\* wgrib、wgrib2 は、第 4 章：Tutorial: Real case で外部入力データのプレ処理を行うために使用する。

## 環境変数の設定

ローカルシステムで MPI 並列プログラムを実行するために、OpenMPI ライブラリの環境変数設定を行う。ユーザ権限に移動して .bashrc をエディタで開き、

```
$ vi ~/.bashrc
```

下記をファイルの最後に追加して、環境変数の設定を記述する。

```
// ----- Add to end of the file -----
# OpenMPI
export MPI="/usr/lib64/mpi/gcc/openmpi"
export PATH="$PATH:$MPI/bin"
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$MPI/lib64"
```

編集が終わったら、環境設定を有効にする。

```
$ . ~/.bashrc
```

## A.4 インストール方法 (Mac OS X 編)

### macports を用いたインストール

Apple Mac OS X での SCALE 実行環境を整備する方法について説明する。ここでは Mac OS X のパッケージマネージャの一つである macports を用いる方法を紹介する。その他の主要なパッケージマネージャとしては、homebrew が挙げられる。homebrew を利用しても環境は手軽に揃えられるので、興味のある方は利用してもらいたい。

まずは Apple の開発ツールである Xcode をインストールする。大元の gcc コンパイラを導入するために、必ずインストールする必要がある。最近の OS のバージョンのものは、App Store 経由で入手できる（無料）。古い OS では、インストールディスクから追加することが出来る。最近の OS の Xcode の場合、最初に以下の様な設定をターミナルから行う必要がある。

### コマンドラインツールのインストール

```
# xcode-select --install
```

### ライセンス条項の承認

```
# xcodebuild -license
```

次に macports 本体をインストールする。https://www.macports.org/  
macports と macports が管理するパッケージは/opt/local 以下に配置される。インストール時に .bash\_profile に、 /opt/local/bin へのパスが張られているので確認されたし。macports はコマンドラインから操作する。主要なコマンドは以下の通り。

インストール可能なソフトウェアを検索する

```
$ port search <検索文字>
```

ソフトウェアのインストール時に選択可能なオプション (variants) を確認する

```
$ port variants <アプリ名>
```

ソフトウェアのインストール (root 権限必要)

```
$ sudo port install <アプリ名> [variants]
```

ソフトウェアのアンインストール (root 権限必要)

```
$ sudo port uninstall <アプリ名> [variants]
```

macports 本体とパッケージカタログの更新 (root 権限必要)

```
$ sudo port selfupdate
```

パッケージの更新 (root 権限必要)

```
$ sudo port upgrade outdated
```

不要なパッケージ (activate されていない過去のバージョン等) の削除

```
$ sudo port -u uninstall
```

## gcc から NetCDF までのインストール

macports はパッケージの依存関係を解決してくれるが、必要な variants を備えたセットを作るには、順番にインストールしていく方が問題が少ない。以下に sudo port install していく順番と variants の設定を示す。今回は gcc4.9 の利用を想定している。

```
$ gcc49  
$ openmpi-gcc49 +threads  
$ hdf4 +gcc49 +szip  
$ hdf5 +gcc49 +szip +fortran +cxx +openmpi +threadsafe  
$ netcdf +gcc49 +openmpi +netcdf4 +hdf4  
$ netcdf-fortran +gcc49 +openmpi
```

最近の macports では、gcc と mpi ライブラリは select で選択する。この操作を行うと、gfortran 等の一般的な名前がエイリアスが作られてパスが通るようになる。

```
$ sudo port select --set gcc mp-gcc49  
$ sudo port select --set mpi openmpi-gcc49-fortran
```

SCALE は陰解法計算の部分で、数値計算ライブラリを利用するオプションがある。もし必要ならば、macports から ATLAS をインストールすることが出来る。

```
$ atlas +gcc49
```

## A.5 インストール方法 (スーパーコンピュータ「京」 編)

以下の説明で使用した環境は次のとおりである。

- 計算機: スーパーコンピュータ「京」
- 言語環境: K-1.2.0-18

### ライブラリについて

スーパーコンピュータ「京」では、SCALE のコンパイルに必要なライブラリが AICS ソフトウェアとして準備されている。詳細は、京ポータルサイトの「AICS ソフトウェア等」の項目、もしくは下記の Web ページを参照のこと。

<http://www-sys-aics.riken.jp/releasedsoftware/ksoftware/pnetcdf.html>

一般に、スーパーコンピュータ「京」における SCALE のコンパイルには、  
”/opt/aics/netcdf/k-serial-noszip/”下にある HDF5、NetCDF ライブラリを用いる。コンパイラや MPI ライブラリについてもスーパーコンピュータ「京」専用のコンパイラとライブラリを用いるため、特別にライブラリ環境を準備する必要はない。

\* コンパイル時に参照するライブラリの PATH は、SCALE コンパイル時に使用する”Makedef.K”に記述されているため、環境変数について特に設定する必要はない。

## A.6 描画ツールのインストール

SCALE の計算結果や、初期値/境界値データなどを描画するのに利用可能である描画ツールの例を挙げる。個人の好みでどのツールを使ってもよいし、出力形式を理解していれば、ここに挙げた以外のツールで解析・描画することももちろん可能である。

- Gphys / Ruby-DCL by 地球電腦倶楽部
  - URL : <http://ruby.gfd-dennou.org/products/gphys/>
  - 概略 : SCALE の出力ファイルは、MPI 並列の計算領域分割に従って MPI プロセスごとに NetCDF 形式の分割ファイルとして出力される。Gphys の”gpview”や”gpvect”といった描画ツールを使えば、分割ファイルを後処理なしに直接開いて描画することができる。
  - インストール方法 : 本書で使用した CentOS6、CentOS7 については、下記の Web ページにインストール方法が記載されている。  
<http://www.gfd-dennou.org/arch/davis/gfdnavi/doc/install/install-ruby-gphys.htm>  
Mac OS X におけるインストール方法は本節の末尾で説明する。
- Grid Analysis and Display System (GrADS) by COLA
  - URL : <http://iges.org/grads/>

- 概略:言わずと知れた描画ツール。SCALEのNetCDF形式の分割ファイルをそのまま読むことはできないため、SCALEで提供している出力データの後処理ツール”netcdf2grads\_h”を使用して分割ファイルを結合し、GrADSで読み込めるファイル形式に変換する必要がある。”netcdf2grads\_h”のインストール方法は、本書の第2.2章、使用方法は第3章、および第4章を参照のこと。
  - インストール方法:<http://iges.org/grads/downloads.html>を参照のこと。CentOS6、CentOS7ではEPELリポジトリを登録していればyumコマンドによって、openSUSE 13ではhome\_oecephafリポジトリを登録していればzypperコマンドによってインストールできる。
- Ncview: a netCDF visual browser by David W. Pierce
    - URL : [http://meteora.ucsd.edu/~pierce/ncview\\_home\\_page.html](http://meteora.ucsd.edu/~pierce/ncview_home_page.html)
    - 概略 : NetCDF形式ファイルのクイックビューアである。SCALEの分割ファイルを結合して描画することはできないが、分割ファイルを1つずつ描画してチェックすることはできる。
    - インストール方法:[http://meteora.ucsd.edu/~pierce/ncview\\_home\\_page.html](http://meteora.ucsd.edu/~pierce/ncview_home_page.html)を参照のこと。CentOS6、CentOS7ではEPELリポジトリを登録していればyumコマンドによって、openSUSE 13ではhome\_oecephafリポジトリを登録していればzypperコマンドによってインストールできる。

## COLUMN

---

### Mac OS XにおけるGphys / Ruby-DCLのインストール

macports-JPが作成しているリポジトリから、Mac OS X用のGPhysをインストールできる。以下のように、MacPorts-JPのリポジトリをチェックアウトして配置する。場所は好きな場所でもよい。

```
$ sudo svn co http://svn.sourceforge.jp/svnroot/macports-jp/trunk/ \
$ /opt/local/var/sources/svn.sourceforge.jp/svnroot/macports-jp/trunk
```

/opt/local/etc/macports/sources.confに、チェックアウトしたリポジトリを追加する。

```
file:///opt/local/var/sources/svn.sourceforge.jp/svnroot/macports-jp/trunk/dports
```

これで、Gphysと関連ライブラリをmacportsで一括インストールできるようになる。今回はruby1.9向けのrb19-gphysをインストールすることにする。

rubyのインストールと1.9の選択

```
$ sudo port install ruby19
$ sudo port select --set ruby ruby19
```

依存関係の解決を任せるかたちで、gphysのインストール

```
$ sudo port install rb19-gphys
```

インストールされた ruby1.9 版 gphys コマンドは、末尾に 1.9 が付いている。これを無くしたい場合は /opt/local/bin 内でシンボリックリンクを張る。以下では特に頻繁に使う 3 つを例に挙げる。

```
$ sudo ln -s gplist1.9 gplist
$ sudo ln -s gpvect1.9 gpvect
$ sudo ln -s gpview1.9 gpview
```

## 付録B Namelist in run.conf

### PARAM\_IO

名称	種類	初期値	説明
IO_LOG_BASENAME	文字列	"LOG"	ログファイルの接頭辞。
IO_LOG_ALLNODE	logical	.false.	全ノードログ出力するかどうか。

### PARAM\_CONST

名称	種類	初期値	説明
CONST_RADIUS	実数	6.37122d+6	惑星半径 (m) (デフォルトは地球)
CONST_OHM	実数	7.2920D-5	惑星の角速度 (1/s)
CONST_GRAV	実数	9.80665D0	重力加速度 ( $m/s^2$ )
CONST_RDRY	実数	287.04D0	乾燥気体の気体定数 (J/kg/K)
CONST_CPDRY	実数	1004.64D0	乾燥気体の定圧比熱 (J/kg/K)
CONST_LAPS	実数	6.5D-3	International Standard Atmosphere (ISA) の 気温減率 (K/m)
CONST_PSTD	実数	101325.D0	標準気圧 (Pa)
CONST_PREO0	実数	100000.D0	基準気圧 (Pa)
CONST_TSTD	実数	288.15D0	基準温度 (K)
CONST_THERMODYN_TYPE	文字列	"EXACT"	内部エネルギーの定義種類。SIMPLE は定数。 EXACT は温度依存。

PARAM.TIME

名称	種類	初期値	説明
TIME_STARTDATE	整数配列	0000, 1, 1, 0, 0, 0	積分実行時の初期時刻。
TIME_STARTMS	実数	0.0D0	初期時刻マイクロ秒。
TIME_DURATION	実数	0.0D0	実行する積分時間。
TIME_DT	実数	0.0D0	積分 1STEP に要する時間。
TIME_DT_ATMOS_DYN	実数	TIME_DT	力学スキームの時間差分値。 TIME_DT の約数である必要がある。
TIME_DT_ATMOS_PHY_MP	実数	TIME_DT	雲微物理スキームの時間差分値。 TIME_DT の倍数である必要がある。
TIME_DT_ATMOS_PHY_RD	実数	TIME_DT	放射スキームの時間差分値。 TIME_DT の倍数である必要がある。
TIME_DT_ATMOS_PHY_SF	実数	TIME_DT	地表面スキームの時間差分値。 TIME_DT の倍数である必要がある。
TIME_DT_ATMOS_PHY_TB	実数	TIME_DT	乱流スキームの時間差分値。 TIME_DT の倍数である必要がある。
TIME_DT_OCEAN	実数	TIME_DT	海洋スキームの時間差分値。 TIME_DT の倍数である必要がある。
TIME_DT_LAND	実数	TIME_DT	陸面スキームの時間差分値。 TIME_DT の倍数である必要がある。
TIME_DT_URBAN	実数	TIME_DT	都市スキームの時間差分値。 TIME_DT の倍数である必要がある。
TIME_DURATION_UNIT	文字列	"SEC"	積分時間単位。
TIME_DT_UNIT	文字列	"SEC"	積分 1STEP の時間単位。
TIME_DT_ATMOS_DYN_UNIT	文字列	TIME_DT_UNIT	力学スキームの時間単位。
TIME_DT_ATMOS_PHY_MP_UNIT	文字列	TIME_DT_UNIT	雲微物理スキームの時間単位。
TIME_DT_ATMOS_PHY_RD_UNIT	文字列	TIME_DT_UNIT	放射スキームの時間単位。
TIME_DT_ATMOS_PHY_SF_UNIT	文字列	TIME_DT_UNIT	地表面スキームの時間単位。
TIME_DT_ATMOS_PHY_TB_UNIT	文字列	TIME_DT_UNIT	乱流スキームの時間単位。
TIME_DT_OCEAN_UNIT	文字列	TIME_DT_UNIT	海洋スキームの時間単位。
TIME_DT_LAND_UNIT	文字列	TIME_DT_UNIT	陸面スキームの時間単位。
TIME_DT_URBAN_UNIT	文字列	TIME_DT_UNIT	都市スキームの時間単位。

\* 次ページへつづく

## PARAM.TIME : つづき

名称	種類	初期値	説明
TIME_DT_ATMOS_RESTART	実数	TIME_DURATION	大気のリスタートファイルを出力する時間間隔。
TIME_DT_ATMOS_RESTART_UNIT	文字列	TIME_DT_UNIT	大気のリスタートファイルを出力する時間間隔の時間単位。
TIME_DT_OCEAN_RESTART	実数	TIME_DURATION	海洋のリスタートファイルを出力する時間間隔。
TIME_DT_OCEAN_RESTART_UNIT	文字列	TIME_DT_UNIT	海洋のリスタートファイルを出力する時間間隔の時間単位。
TIME_DT_LAND_RESTART	実数	TIME_DURATION	地表面スキーム関係のリスタートファイルを出力する時間間隔。
TIME_DT_LAND_RESTART_UNIT	文字列	TIME_DT_UNIT	地表面スキーム関係のリスタートファイルを出力する時間間隔の時間単位。
TIME_DT_URBAN_RESTART	実数	TIME_DURATION	都市スキーム関係のリスタートファイルを出力する時間間隔。
TIME_DT_URBAN_RESTART_UNIT	文字列	TIME_DT_UNIT	都市スキームのリスタートファイルを出力する時間間隔の時間単位。

## PARAM.GRID

名称	種類	初期値	説明
GRID_IN_BASENAME	文字列		Grid を外部から与えるときの入力ファイル (省略時は DX, DY, DZ, BAFFFACT などから生成する)
GRID_OUT_BASENAME	文字列		Grid 情報の出力ファイル名
DX	実数	500.D0	X 方向の Grid の間隔 (m)
DY	実数	500.D0	Y 方向の Grid の間隔 (m)
DZ	実数	500.D0	Z 方向の Grid の間隔 (m)
BUFFER_DZ	実数	0.D0	Z 方向のダンピング層 (スポンジ層) の厚さ (m)
BUFFER_DX	実数	0.D0	X 方向のダンピング層 (スポンジ層) の厚さ (m)
BUFFER_DY	実数	0.D0	Y 方向のダンピング層 (スポンジ層) の厚さ (m)
BAFFFACT	実数	1.D0	ダンピング層での Grid の引き伸ばし度合い ( $(dx)_{i+1} = (dx)_i^{BAFFFACT}$ )

## PARAM.STATISTICS

名称	種類	初期値	説明
STATISTICS_checktotal	logical	.false.	値のチェックを行うかどうか。
STATISTICS_use_globalcomm	logical	.false.	全ノード通信を行うかどうか。

#### PARAM.RESTRAT

名称	種類	初期値	説明
RESTART_OUTPUT	logical	.false.	restart ファイルを出力するかどうか。
RESTART_OUT_BASENAME	文字列		書き出す restart ファイルの接頭辞。 RESTART_OUTPUT が真のときに有効。
RESTART_IN_BASENAME	文字列		読み込む restart ファイルの接頭辞。

#### PARAM.TOPO

名称	種類	初期値	説明
TOPO_IN_BASENAME	文字列		読み込む地形ファイルの接頭辞。

#### PARAM.LANDUSE

名称	種類	初期値	説明
LANDUSE_IN_BASENAME	文字列		読み込む土地利用ファイルの接頭辞。

#### PARAM.LAND\_PROPERTY

名称	種類	初期値	説明
LAND_PROPERTY_IN_FILENAME	文字列		読み込む土壌パラメータファイル名。

#### PARAM.PRC

名称	種類	初期値	説明
PRC_NUM_X	整数		X 方向に割り当てるプロセス数。
PRC_NUM_Y	整数		Y 方向に割り当てるプロセス数。
PRC_PERIODIC_X	logical		X 方向に周期境界とするかどうか。
PRC_PERIODIC_Y	logical		Y 方向に周期境界とするかどうか。

#### PARAM.INDEX

名称	種類	初期値	説明
KMAX	整数		大気鉛直層数。
IMAX	整数		プロセスあたりの X 方向の格子数。
JMAX	整数		プロセスあたりの Y 方向の格子数。

#### PARAM.LAND\_INDEX

名称	種類	初期値	説明
LKMAX	整数		陸面鉛直層数。

#### PARAM.URBAN\_INDEX

名称	種類	初期値	説明
UKMAX	整数		都市鉛直層数。

#### PARAM.LAND\_GRID

名称	種類	初期値	説明
LDZ	実数配列		陸面の鉛直層の層厚。鉛直層数分の設定が必要。

#### PARAM.URBAN\_GRID

名称	種類	初期値	説明
UDZ	実数配列		都市の鉛直層の層厚。鉛直層数分の設定が必要。

#### PARAM.MAPPROJ

名称	種類	初期値	説明
MPRJ_basepoint_lon	実数		計算領域の中心経度。
MPRJ_basepoint_lat	実数		計算領域の中心緯度。
MPRJ_type	文字列		計算領域の投影図法。
MPRJ_LC_lat1	実数		投影図法が LC の場合の参照緯度 1。
MPRJ_LC_lat2	実数		投影図法が LC の場合の参照緯度 2。

#### PARAM.TRACER

名称	種類	初期値	説明
TRACER_TYPE	文字列	"OFF"	トレーサの種類。通常、ATMOS_PHY_MP_TYPE と同じ。

#### PARAM.ATMOS

名称	種類	初期値	説明
ATMOS_DYN_TYPE	文字列	"OFF"	力学スキームの種類。(OFF、HEVE、HEVI、HIVI が選択可能)
ATMOS_PHY_MP_TYPE	文字列	"OFF"	雲微物理スキームの種類。(OFF、KESSLER、TOMITA08、SN14、SUZUKI10 が選択可能)
ATMOS_PHY_RD_TYPE	文字列	"OFF"	放射スキームの種類。(OFF、MSTRNX が選択可能)
ATMOS_PHY_SF_TYPE	文字列	"OFF"	地表面スキームの種類。(OFF、CONST、COUPLE が選択可能)
ATMOS_PHY_TB_TYPE	文字列	"OFF"	乱流スキームの種類。(OFF、SMAGORINSKY、MYNN が選択可能)

#### PARAM.OCEAN

名称	種類	初期値	説明
OCEAN_TYPE	文字列	"OFF"	海洋スキームの種類。(OFF、CONST が選択可能)

#### PARAM.LAND

名称	種類	初期値	説明
LAND_TYPE	文字列	"OFF"	陸面スキームの種類。(OFF、SLUB が選択可能)

## PARAM\_URBAN

名称	種類	初期値	説明
URBAN_TYPE	文字列	"OFF"	都市スキームの種類。(OFF、SLC が選択可能)

\*

## PARAM\_HIST

名称	種類	初期値	説明
HIST_BND	logical	.true.	Nesting 時に xy 方向の halo 領域を History ファイルに出力するか。

## PARAM\_HISTORY

名称	種類	初期値	説明
HISTORY_TITLE	文字列	SCALE-LES HISTORY OUTPUT	NetCDF 形式の History ファイルのタイトル
HISTORY_DEFAULT_BASENAME	文字列	'history'	History ファイルのファイル名(相対パス)
HISTORY_DEFAULT_TINTERVAL	実数	1.0-DP	History を出力する時間間隔。HISTORY_DEFAULT_TUNIT が単位となる。
HISTORY_DEFAULT_TUNIT	文字列	"SEC"	HISTORY_DEFAULT_TINTERVAL の単位 (SEC、HOUR、DAY が利用可能)
HISTORY_DEFAULT_TAVERAGE	logical	.false.	History に出力する際 HISTORY_DEFAULT_TINTERVAL で指定した時間間隔ごとの時間平均を出力するか。
HISTORY_DEFAULT_ZINTERP	logical	.false.	出力時に z 面への内挿をするかどうか
HISTORY_DEFAULT_DATATYPE	文字列	REAL4	出力データの precision
HISTORY_OUTPUT_STEPO	logical	.false.	初期時刻 (t=0) の値を出力するかどうか

## PARAM\_NEST

名称	種類	初期値	説明
USE_NESTING	logical	.false.	ネスティング機能を使うかどうか。
LATLON_CATALOGUE_FNAME	文字列		入力する親ドメインの latlon カタログのファイル名。
OFFLINE_PARENT_PRC_NUM_X	整数		親ドメインの MPI プロセス数 (X 方向)。OFFLINE が真のときのみ有効。
OFFLINE_PARENT_PRC_NUM_Y	整数		親ドメインの MPI プロセス数 (Y 方向)。OFFLINE が真のときのみ有効。
OFFLINE_PARENT_KMAX	整数		親ドメインの 1 プロセスあたりの格子点数 (Z 方向)。OFFLINE が真のときのみ有効。
OFFLINE_PARENT_IMAX	整数		親ドメインの 1 プロセスあたりの格子点数 (X 方向)。OFFLINE が真のときのみ有効。
OFFLINE_PARENT_JMAX	整数		親ドメインの 1 プロセスあたりの格子点数 (Y 方向)。OFFLINE が真のときのみ有効。
OFFLINE_PARENT_LKMAX	整数		親ドメインの 1 プロセスあたりの格子点数 (陸面モデル)。OFFLINE が真のときのみ有効。
OFFLINE	logical	.true.	オフラインかオンラインのどちらのシステムを使うか。USE_NESTING が真のときのみ有効。
ONLINE_DOMAIN_NUM	整数		ドメイン ID 番号。オンライン・ネスティングのときのみ有効。
ONLINE_IAM_PARENT	logical	.false.	当該ドメインが親ドメインとして働くかどうか。オンライン・ネスティングのときのみ有効。
ONLINE_IAM_DAUGHTER	logical	.false.	当該ドメインが子ドメインとして働くかどうか。オンライン・ネスティングのときのみ有効。
ONLINE_USE_VELZ	logical	.false.	鉛直速度をバッファ領域のダンピングに使用するかどうか。オンライン・ネスティングのときのみ有効。
ONLINE_NO_ROTATE	logical	.false.	水平速度成分に対して地図投影法の違いに基づく回転効果の変換を行うかどうか。オンライン・ネスティングのときのみ有効。
ONLINE_BOUNDARY_USE_QHYD	logical	.false.	凝結物の混合比をバッファ領域のダンピングに使用するかどうか。オンライン・ネスティングのときのみ有効。
ONLINE_AGGRESSIVE_COMM	logical	.false.	より安全にドメイン間の同期通信を行うかどうか。オンライン・ネスティングのときのみ有効。
ONLINE_SPECIFIED_MAXRQ	整数	1000	非同期 MPI 通信の同時 Request 数の最大値。オンライン・ネスティングのときのみ有効。
NEST_INTERP_LEVEL	整数	4	水平 2 次元内挿処理に使用する格子点数の指定 ; 1, 3, 4, 8, 12 のうちから指定する。オンライン・ネスティングのときのみ有効。

## PARAM\_ATMOS\_VARS

名称	種類	初期値	説明
ATMOS_VARS_CHECKRANGE	logical	.false.	N/A

## PARAM\_ATMOS\_REFSTATE

名称	種類	初期値	説明
ATMOS_REFSTATE_TYPE	文字列	UNIFORM	Reference state に何を用いるか。 (UNIFORM：一定値で与える。値はATMOS_REFSTATE_**によって決まる、INIT：初期値（リスタートの時はリスタート時の値）、FILE：ATMOS_REFSTATE_IN_BASENAME から読み出す、ISA：International Standard Atmosphere)
ATMOS_REFSTATE_IN_BASENAME	文字列		Reference state の入力ファイル名
ATMOS_REFSTATE_OUT_BASENAME	文字列		Reference state の出力ファイル名
ATMOS_REFSTATE_UPDATE_DT	実数	0.D0	Reference State をアップデートする時間間隔（0の時はReference stateが更新されない）

## PARAM.ATMOS.BOUNDARY

名称	種類	初期値	説明
ATMOS_BOUNDARY_TYPE	文字列	NONE	バッファ層でどの値に緩和させるか。(CONST:一定値(ATMOS_BOUNDARY_VALUE_**に緩和)、INIT:初期値に緩和、REAL:Nesting時、FILE:入力ファイル(ATMOS_BOUNDARY_IN_BASENAME)から読み込む))
ATMOS_BOUNDARY_IN_BASENAME	文字列		ATMOS_BOUNDARY_TYPE=FILEの時に読み込む外部ファイル
ATMOS_BOUNDARY_OUT_BASENAME	文字列		バッファ層で緩和させる値を出力する際の出力ファイル名(設定しない場合は出力されない)
ATMOS_BOUNDARY_USE_VELZ	logical	.false.	鉛直方向の運動量にダンピングを作用させるか否か。
ATMOS_BOUNDARY_USE_VELY	logical	.false.	Y方向の運動量にダンピングを作用させるか否か。
ATMOS_BOUNDARY_USE_VELX	logical	.false.	X方向の運動量にダンピングを作用させるか否か。
ATMOS_BOUNDARY_USE_POTT	logical	.false.	温位にダンピングを作用させるか否か。
ATMOS_BOUNDARY_USE_QV	logical	.false.	水蒸気混合比にダンピングを作用させるか否か。
ATMOS_BOUNDARY_USE_QHYD	logical	.false.	水物質の混合比にダンピングを作用させるか否か。
ATMOS_BOUNDARY_VALUE_VELZ	実数	0.D0	鉛直方向の運動量を緩和させる値(ATMOS_BOUNDARY_TYPE=.true.の時に有効)
ATMOS_BOUNDARY_VALUE_VELX	実数	0.D0	X方向の運動量を緩和させる値(ATMOS_BOUNDARY_TYPE=.true.の時に有効)
ATMOS_BOUNDARY_VALUE_VELY	実数	0.D0	Y方向の運動量を緩和させる値(ATMOS_BOUNDARY_TYPE=.true.の時に有効)
ATMOS_BOUNDARY_VALUE_POTT	実数	3.D0	温位を緩和させる値(ATMOS_BOUNDARY_TYPE=.true.の時に有効)

\*次ページへつづく

PARAM.ATMOS\_BOUNDARY : つづき

名称	種類	初期値	説明
ATMOS_BOUNDARY_VALUE_QV	実数	0.D0	水蒸気混合比を緩和させる値 ( ATMOS_BOUNDARY_TYPE=.true. の時に有効 )
ATMOS_BOUNDARY_VALUE_QHYD	実数	0.D0	水物質の混合比を緩和させる値 ( ATMOS_BOUNDARY_TYPE=.true. の時に有効 )
ATMOS_BOUNDARY_LINEAR_V	logical	.false.	N/A
ATMOS_BOUNDARY_LINEAR_H	logical	.true.	N/A
ATMOS_BOUNDARY_EXP_H	実数	2.D0	N/A

PARAM.ATMOS\_DYN

名称	種類	初期値	説明
ATMOS_DYN_NUMERICAL_DIFF_ORDER	実数	1.D0.	数値粘性のオーダーを4で割ったもの ( ATMOS_DYN_NUMERICAL_DIFF_ORDER =1 なら4次の数値粘性になる )
ATMOS_DYN_NUMERICAL_DIFF_COEF	実数	1.D-4	混合比以外の予報変数にかかる数値粘性の強さを決める無次元定数
ATMOS_DYN_NUMERICAL_DIFF_COEF_Q	実数	1.D-4	混合比にかかる数値粘性の強さを決める無次元定数
ATMOS_DYN_ENABLE_CORIOLIS	logical	.false.	コリオリ力を考慮するか
ATMOS_DYN_FLAG_FCT_RHO	logical	.false.	密度に FCT をかけるか
ATMOS_DYN_FLAG_FCT_MOMENTUM	logical	.false.	運動量に FCT をかけるか。
ATMOS_DYN_FLAG_FCT_T	logical	.false.	温位に FCT をかけるか。

PARAM.ATMOS\_PHY\_MP

名称	種類	初期値	説明
MP_DOPRECIPIATION	logical	.true.	水物質の重力による落下を考慮するか否か
MP_DONEGATIVE_FIXER	logical	.false.	雲物理スキームの前後で負値を0にするか否か。
MP_NTMAX_SEDIMENTATION	整数	1	水物質の重力落下の計算をする際に区切るステップ数

PARAM\_ATMOS\_PHY\_RD\_MSTRN

名称	種類	初期値	説明
ATMOS_PHY_RD_MSTRN_KADD	整数	0	放射スキームで計算する際にモデル上端から上に加える層数。
ATMOS_PHY_RD_MSTRN_GASPARA_IN_FILENAME	文字列	'PARAG.29'	放射スキームで用いる希ガスパラメータの入力ファイル名
ATMOS_PHY_RD_MSTRN_AEROPARA_IN_FILENAME	文字列	'PARAC.29'	放射スキームで用いるエアロゾルパラメータの入力ファイル名
ATMOS_PHY_RD_MSTRN_HYGROPARA_IN_FILENAME	文字列	'VARDATA.RM29'	放射スキームで用いる水蒸気と水物質パラメータの入力ファイル名
ATMOS_PHY_RD_MSTRN_NBAND	整数	29	MSTRNX で計算に用いる波長バンド数 ( ATMOS_PHY_RD_MSTRN_GAS (AERO, HYGRO) PARA_IN_FILENAME のバンド数と対応する必要がある )

PARAM\_ATMOS\_PHY\_RD\_PROFILE

名称	種類	初期値	説明
ATMOS_PHY_RD_PROFILE_TOA	実数	100.D0	放射スキームで用いる Top Of Atmospher (TOA) の高度 (km)
ATMOS_PHY_RD_PROFILE_USE_CLIMATOLOGY	logical	.true.	放射スキームに必要な大気のプロファイルに気候値を用いるかどうか。
ATMOS_PHY_RD_PROFILE_CIRA86_IN_FILENAME	文字列	./	cira.nc の相対パス
ATMOS_PHY_RD_PROFILE_MIPAS2001_IN_BASENAME	文字列		MIPAS2001 が保存されているディレクトリ名 ( 相対パス )
ATMOS_PHY_RD_PROFILE_USER_IN_FILENAME	文字列		放射スキームに用いる大気のプロファイルの入力ファイル名 ( 省略された場合は気候値が用いられる )
ATMOS_PHY_RD_PROFILE_USE_CO2	logical	.true.	放射計算で二酸化炭素を考慮するか
ATMOS_PHY_RD_PROFILE_USE_O3	logical	.true.	放射計算でオゾンを考慮するか
ATMOS_PHY_RD_PROFILE_USE_CO	logical	.true.	放射計算で一酸化炭素を考慮するか
ATMOS_PHY_RD_PROFILE_USE_N2O	logical	.true.	放射計算で一酸化二窒素を考慮するか
ATMOS_PHY_RD_PROFILE_USE_CH4	logical	.true.	放射計算でメタンを考慮するか
ATMOS_PHY_RD_PROFILE_USE_O2	logical	.true.	放射計算で酸素を考慮するか
ATMOS_PHY_RD_PROFILE_USE_CFC	logical	.true.	放射計算で CFC を考慮するか
DEBUG	logical	.false.	放射計算において DEBUG 情報を出力するか否か

## PARAM.OCEAN.PHY.SLAB

名称	種類	初期値	説明
OCEAN_PHY_SLAB_DEPTH	実数	10.D0	スラブモデルにおける海洋の深さ

## 付録C Variables for history output

### mod\_atmos\_phy\_ae\_driver

Item/Variable	Description	Unit
CCN	cloud condensation nuclei	

### mod\_atmos\_phy\_ch\_driver

Item/Variable	Description	Unit
Ozone	Ozone	PPM
CBMFX	cloud base mass flux	kg/m <sup>2</sup> /s

### mod\_atmos\_phy\_cp\_driver

Item/Variable	Description	Unit
CBMFX	cloud base mass flux	kg/m <sup>2</sup> /s

### mod\_atmos\_phy\_mp\_driver

Item/Variable	Description	Unit
RAIN	surface rain rate	kg/m <sup>2</sup> /s
SNOW	surface snow rate	kg/m <sup>2</sup> /s
PREC	surface precipitation rate	kg/m <sup>2</sup> /s
DENS_t_MP	tendency DENS in MP	kg/m <sup>3</sup> /s
MOMZ_t_MP	tendency MOMZ in MP	kg/m <sup>2</sup> /s <sup>2</sup>
MOMX_t_MP	tendency MOMX in MP	kg/m <sup>2</sup> /s <sup>2</sup>
MOMY_t_MP	tendency MOMY in MP	kg/m <sup>2</sup> /s <sup>2</sup>
RHOT_t_MP	tendency RHOT in MP	K * kg/m <sup>3</sup> /s
QV_t_MP	tendency QV in MP	kg/kg

**mod\_atmos\_phy\_rd\_driver**

Item/Variable	Description	Unit
SOLINS	solar insolation	$W/m^2$
COSZ	cos(solar zenith angle)	0 – 1
SFLX_LW_up	SFC upward longwave radiation flux	$W/m^2$
SFLX_LW_dn	SFC downward longwave radiation flux	$W/m^2$
SFLX_SW_up	SFC upward shortwave radiation flux	$W/m^2$
SFLX_SW_dn	SFC downward shortwave radiation flux	$W/m^2$
TOAFLX_LW_up	TOA upward longwave radiation flux	$W/m^2$
TOAFLX_LW_dn	TOA downward longwave radiation flux	$W/m^2$
TOAFLX_SW_up	TOA upward shortwave radiation flux	$W/m^2$
TOAFLX_SW_dn	TOA downward shortwave radiation flux	$W/m^2$
SLR	SFC net longwave radiation flux	$W/m^2$
SSR	SFC net shortwave radiation flux	$W/m^2$
OLR	TOA net longwave radiation flux	$W/m^2$
OSR	TOA net shortwave radiation flux	$W/m^2$
RADFLUX_LWUP	upward longwave radiation flux	$W/m^2$
RADFLUX_LWDN	downward longwave radiation flux	$W/m^2$
RADFLUX_LW	net longwave radiation flux	$W/m^2$
RADFLUX_SWUP	upward shortwave radiation flux	$W/m^2$
RADFLUX_SWDN	downward shortwave radiation flux	$W/m^2$
RADFLUX_SW	net shortwave radiation flux	$W/m^2$
TEMP_t_rd_LW	tendency of temp in rd(LW)	$K/day$
TEMP_t_rd_SW	tendency of temp in rd(SW)	$K/day$
TEMP_t_rd	tendency of temp in rd	$K/day$
RHOT_t_RD	tendency of RHOT in rd	$K.kg/m^3/s$

**mod\_atmos\_phy\_sf\_driver**

Item/Variable	Description	Unit
SFC_DENS	surface atmospheric density	$kg/m^3$
SFC_PRES	surface atmospheric pressure	$Pa$
SFC_TEMP	surface skin temperature (merged)	$K$
SFC_ALB_LW	surface albedo (longwave,merged)	0 – 1
SFC_ALB_SW	surface albedo (shortwave,merged)	0 – 1
SFC_ZOM	roughness length (momentum)	$m$
SFC_ZOH	roughness length (heat)	$m$
SFC_ZOE	roughness length (vapor)	$m$
MWFLX	w-momentum flux (merged)	$kg/m^2/s$
MUFLX	u-momentum flux (merged)	$kg/m^2/s$
MVFLX	v-momentum flux (merged)	$kg/m^2/s$
SHFLX	sensible heat flux (merged)	$W/m^2$
LHFLX	latent heat flux (merged)	$W/m^2$
GHFLX	ground heat flux (merged)	$W/m^2$
Uabs10	10m absolute wind	$m/s$
U10	10m x-wind	$m/s$
V10	10m y-wind	$m/s$
T2	2m temperature	$K$
Q2	2m water vapor	$kg/kg$
MSLP	mean sea-level pressure	$Pa$

mod\_atmos\_phy\_tb\_driver

Item/Variable	Description	Unit
TKE	turbulent kinetic energy	$m^2/s^2$
NU	eddy viscosity	$m^2/s$
Ri	Richardson number	<i>NIL</i>
Pr	Prantle number	<i>NIL</i>
MOMZ_t_TB	MOMZ tendency (TB)	$kg/m^2/s^2$
MOMX_t_TB	MOMX tendency (TB)	$kg/m^2/s^2$
MOMY_t_TB	MOMY tendency (TB)	$kg/m^2/s^2$
RHOT_t_TB	RHOT tendency (TB)	$K.kg/m^3/s$
QV_t_TB	QV tendency (TB)	$kg/kg$
QC_t_TB	QC tendency (TB)	$kg/kg$
QR_t_TB	QR tendency (TB)	$kg/kg$
QI_t_TB	QI tendency (TB)	$kg/kg$
QS_t_TB	QS tendency (TB)	$kg/kg$
QG_t_TB	QG tendency (TB)	$kg/kg$
SGS_ZFLX_MOMZ	SGS Z FLUX of MOMZ	$kg/m/s^2$
SGS_XFLX_MOMZ	SGS X FLUX of MOMZ	$kg/m/s^2$
SGS_YFLX_MOMZ	SGS Y FLUX of MOMZ	$kg/m/s^2$
SGS_ZFLX_MOMX	SGS Z FLUX of MOMX	$kg/m/s^2$
SGS_XFLX_MOMX	SGS X FLUX of MOMX	$kg/m/s^2$
SGS_YFLX_MOMX	SGS Y FLUX of MOMX	$kg/m/s^2$
SGS_ZFLX_MOMY	SGS Z FLUX of MOMY	$kg/m/s^2$
SGS_XFLX_MOMY	SGS X FLUX of MOMY	$kg/m/s^2$
SGS_YFLX_MOMY	SGS Y FLUX of MOMY	$kg/m/s^2$
SGS_ZFLX_RHOT	SGS Z FLUX of RHOT	$K * kg/m^2/s$
SGS_XFLX_RHOT	SGS X FLUX of RHOT	$K * kg/m^2/s$
SGS_YFLX_RHOT	SGS Y FLUX of RHOT	$K * kg/m^2/s$
SGS_ZFLX_QV	SGS Z FLUX of QV	$kg/m^2/s$
SGS_XFLX_QV	SGS X FLUX of QV	$kg/m^2/s$
SGS_YFLX_QV	SGS Y FLUX of QV	$kg/m^2/s$
SGS_ZFLX_QC	SGS Z FLUX of QC	$kg/m^2/s$
SGS_XFLX_QC	SGS X FLUX of QC	$kg/m^2/s$
SGS_YFLX_QC	SGS Y FLUX of QC	$kg/m^2/s$
SGS_ZFLX_QR	SGS Z FLUX of QR	$kg/m^2/s$
SGS_XFLX_QR	SGS X FLUX of QR	$kg/m^2/s$
SGS_YFLX_QR	SGS Y FLUX of QR	$kg/m^2/s$

\* 次ページへつづく

mod\_atmos\_phy\_tb\_driver : つづき

Item/Variable	Description	Unit
SGS_ZFLX_QI	SGS Z FLUX of QI	<i>kg/m2/s</i>
SGS_XFLX_QI	SGS X FLUX of QI	<i>kg/m2/s</i>
SGS_YFLX_QI	SGS Y FLUX of QI	<i>kg/m2/s</i>
SGS_ZFLX_QS	SGS Z FLUX of QS	<i>kg/m2/s</i>
SGS_XFLX_QS	SGS X FLUX of QS	<i>kg/m2/s</i>
SGS_YFLX_QS	SGS Y FLUX of QS	<i>kg/m2/s</i>
SGS_ZFLX_QG	SGS Z FLUX of QG	<i>kg/m2/s</i>
SGS_XFLX_QG	SGS X FLUX of QG	<i>kg/m2/s</i>
SGS_YFLX_QG	SGS Y FLUX of QG	<i>kg/m2/s</i>

mod\_atmos\_vars

Item/Variable	Description	Unit
DENS	density	$kg/m^3$
MOMZ	momentum z	$kg/m^2/s$
MOMX	momentum x	$kg/m^2/s$
MOMY	momentum y	$kg/m^2/s$
RHOT	rho * theta	$kg/m^3 * K$
QV	Water Vapor mixing ratio	$kg/kg$
QC	Cloud Water mixing ratio	$kg/kg$
QR	Rain Water mixing ratio	$kg/kg$
QI	Cloud Ice mixing ratio	$kg/kg$
QS	Snow mixing ratio	$kg/kg$
QG	Graupel mixing ratio	$kg/kg$
W	velocity w	$m/s$
U	velocity u	$m/s$
V	velocity v	$m/s$
PT	potential temp.	$K$
QDRY	dry air	$kg/kg$
QTOT	total water	$kg/kg$
QHYD	total hydrometeors	$kg/kg$
QLIQ	total liquid water	$kg/kg$
QICE	total ice water	$kg/kg$
LWP	liquid water path	$g/m^2$
IWP	ice water path	$g/m^2$
RTOT	Total gas constant	$J/kg/K$
CPTOT	Total heat capacity	$J/kg/K$
PRES	pressure	$Pa$
T	temperature	$K$
LWPT	liq. potential temp.	$K$
RHA	relative humidity(liq+ice)	%
RH	relative humidity(liq)	%
RHI	relative humidity(ice)	%
VOR	vertical vorticity	$1/s$
DIV	divergence	$1/s$
HDIV	horizontal divergence	$1/s$
Uabs	absolute velocity	$m/s$

\* 次ページへつづく

**mod\_atmos\_vars : つづき**

Item/Variable	Description	Unit
DENS_PRIM	horiz. deviation of density	$kg/m^3$
W_PRIM	horiz. deviation of w	$m/s$
U_PRIM	horiz. deviation of u	$m/s$
V_PRIM	horiz. deviation of v	$m/s$
PT_PRIM	horiz. deviation of pot. temp.	$K$
W_PRIM2	variance of w	$m^2/s^2$
PT_W_PRIM	resolved scale heat flux	$W/s$
W_PRIM3	skewness of w	$m^3/s^3$
TKE_RS	resolved scale TKE	$m^2/s^2$
ENGT	total energy	$J/m^3$
ENGP	potential energy	$J/m^3$
ENGK	kinetic energy	$J/m^3$
ENGI	internal energy	$J/m^3$

**mod\_land\_phy\_driver**

Item/Variable	Description	Unit
LAND_TEMP_t	tendency of LAND_TEMP	$K$
LAND_WATER_t	tendency of LAND_WATER	$m^3/m^3$
LAND_SFC_TEMP_t	tendency of LAND.SFC.TEMP	$K$
LAND_ALB_LW_t	tendency of LAND_ALB.LW	0 – 1
LAND_ALB_SW_t	tendency of LAND_ALB.SW	0 – 1
LP_WaterLimit	LAND PROPERTY maximum soil moisture	$m^3/m^3$
LP_WaterCritical	LAND PROPERTY critical soil moisture	$m^3/m^3$
LP_ThermalCond	LAND PROPERTY thermal conductivity for soil	$W/K/m$
LP_HeatCapacity	LAND PROPERTY heat capacity for soil	$J/K/m^3$
LP_WaterDiff	LAND PROPERTY moisture diffusivity in the soil	$m^2/s$
LP_ZOM	LAND PROPERTY roughness length for momentum	$m$
LP_ZOH	LAND PROPERTY roughness length for heat	$m$
LP_ZOE	LAND PROPERTY roughness length for vapor	$m$

**mod\_land\_vars**

Item/Variable	Description	Unit
LAND_TEMP	temperature at each soil layer	<i>K</i>
LAND_WATER	moisture at each soil layer	<i>m<sup>3</sup>/m<sup>3</sup></i>
LAND_SFC_TEMP	land surface skin temperature	<i>K</i>
LAND_ALB_LW	land surface albedo (longwave)	0 – 1
LAND_ALB_SW	land surface albedo (shortwave)	0 – 1
LAND_SFLX_MW	land surface w-momentum flux	<i>kg/m<sup>2</sup>/s</i>
LAND_SFLX_MU	land surface u-momentum flux	<i>kg/m<sup>2</sup>/s</i>
LAND_SFLX_MV	land surface v-momentum flux	<i>kg/m<sup>2</sup>/s</i>
LAND_SFLX_SH	land surface sensible heat flux	<i>J/m<sup>2</sup>/s</i>
LAND_SFLX_LH	land surface latent heat flux	<i>J/m<sup>2</sup>/s</i>
LAND_SFLX_GH	land surface ground heat flux	<i>J/m<sup>2</sup>/s</i>
LAND_SFLX_evap	land surface water vapor flux	<i>kg/m<sup>2</sup>/s</i>

**mod\_ocean\_phy\_driver**

Item/Variable	Description	Unit
OCEAN_TEMP_t	tendency of OCEAN_TEMP	<i>K</i>
OCEAN_SFC_TEMP_t	tendency of OCEAN_SFC_TEMP	<i>K</i>
OCEAN_ALB_LW_t	tendency of OCEAN_ALB_LW	0 – 1
OCEAN_ALB_SW_t	tendency of OCEAN_ALB_SW	0 – 1
OCEAN_SFC_ZOM_t	tendency of OCEAN_SFC_ZOM	<i>m</i>
OCEAN_SFC_ZOH_t	tendency of OCEAN_SFC_ZOH	<i>m</i>
OCEAN_SFC_ZOE_t	tendency of OCEAN_SFC_ZOE	<i>m</i>

**mod\_ocean\_vars**

Item/Variable	Description	Unit
OCEAN_TEMP	temperature at uppermost ocean layer	<i>K</i>
OCEAN_SFC_TEMP	ocean surface skin temperature	<i>K</i>
OCEAN_ALB_LW	ocean surface albedo (longwave)	0 – 1
OCEAN_ALB_SW	ocean surface albedo (shortwave)	0 – 1
OCEAN_SFC_ZOM	ocean surface roughness length (momentum)	<i>m</i>
OCEAN_SFC_ZOH	ocean surface roughness length (heat)	<i>m</i>
OCEAN_SFC_ZOE	ocean surface roughness length (vapor)	<i>m</i>
OCEAN_SFLX_MW	ocean surface w-momentum flux	<i>kg/m<sup>2</sup>/s</i>
OCEAN_SFLX_MU	ocean surface u-momentum flux	<i>kg/m<sup>2</sup>/s</i>
OCEAN_SFLX_MV	ocean surface v-momentum flux	<i>kg/m<sup>2</sup>/s</i>
OCEAN_SFLX_SH	ocean surface sensible heat flux	<i>J/m<sup>2</sup>/s</i>
OCEAN_SFLX_LH	ocean surface latent heat flux	<i>J/m<sup>2</sup>/s</i>
OCEAN_SFLX_WH	ocean surface water heat flux	<i>J/m<sup>2</sup>/s</i>
OCEAN_SFLX_evap	ocean surface water vapor flux	<i>kg/m<sup>2</sup>/s</i>

**mod\_urban\_phy\_driver**

Item/Variable	Description	Unit
URBAN_TR_t	tendency of URBAN_TR	<i>K</i>
URBAN_TB_t	tendency of URBAN_TB	<i>K</i>
URBAN_TG_t	tendency of URBAN_TG	<i>K</i>
URBAN_TC_t	tendency of URBAN_TC	<i>K</i>
URBAN_QC_t	tendency of URBAN_QC	<i>kg/kg</i>
URBAN_UC_t	tendency of URBAN_UC	<i>m/s</i>
URBAN_TRL_t	tendency of URBAN_TRL	<i>K</i>
URBAN_TBL_t	tendency of URBAN_TBL	<i>K</i>
URBAN_TGL_t	tendency of URBAN_TGL	<i>K</i>
URBAN_RAINR_t	tendency of URBAN_RAINR	<i>K</i>
URBAN_RAINB_t	tendency of URBAN_RAINB	<i>K</i>
URBAN_RAING_t	tendency of URBAN_RAING	<i>K</i>
URBAN_ROFF_t	tendency of URBAN_ROFF	<i>K</i>

**mod\_urban\_vars**

Item/Variable	Description	Unit
URBAN_TR	urban surface temperature of roof	<i>K</i>
URBAN_TB	urban surface temperature of wall	<i>K</i>
URBAN_TG	urban surface temperature of road	<i>K</i>
URBAN_TC	urban canopy air temperature	<i>K</i>
URBAN_QC	urban canopy humidity	<i>kg/kg</i>
URBAN_UC	urban canopy wind	<i>m/s</i>
URBAN_TRL	urban temperature in layer of roof	<i>K</i>
URBAN_TBL	urban temperature in layer of wall	<i>K</i>
URBAN_TGL	urban temperature in layer of road	<i>K</i>
URBAN_RAINR	urban rain strage on roof	<i>kg/m2</i>
URBAN_RAINB	urban rain strage on wall	<i>kg/m2</i>
URBAN_RAING	urban rain strage on road	<i>kg/m2</i>
URBAN_ROFF	urban runoff	<i>kg/m2</i>
URBAN_SFC_TEMP	urban grid average of temperature	<i>K</i>
URBAN_ALB_LW	urban grid average of albedo LW	0 – 1
URBAN_ALB_SW	urban grid average of albedo SW	0 – 1
URBAN_SFLX_MW	urban grid average of w-momentum flux	<i>kg/m2/s</i>
URBAN_SFLX_MU	urban grid average of u-momentum flux	<i>kg/m2/s</i>
URBAN_SFLX_MV	urban grid average of v-momentum flux	<i>kg/m2/s</i>
URBAN_SFLX_SH	urban grid average of sensible heat flux	<i>W/m2</i>
URBAN_SFLX_LH	urban grid average of latent heat flux	<i>W/m2</i>
URBAN_SFLX_GH	urban grid average of ground heat flux	<i>W/m2</i>
URBAN_SFLX_evap	urban grid average of water vapor flux	<i>kg/m2/s</i>

**scale\_atmos\_dyn**

Item/Variable	Description	Unit
DENS_t_phys	tendency of density due to physics	$kg/m^3/s$
MOMZ_t_phys	tendency of momentum z due to physics	$kg/m^2/s^2$
MOMX_t_phys	tendency of momentum x due to physics	$kg/m^2/s^2$
MOMY_t_phys	tendency of momentum y due to physics	$kg/m^2/s^2$
RHOT_t_phys	tendency of rho*theta temperature due to physics	$Kkg/m^3/s$
QV_t_phys	tendency of QV due to physics	$kg/kg$
QC_t_phys	tendency of QC due to physics	$kg/kg$
QR_t_phys	tendency of QR due to physics	$kg/kg$
QI_t_phys	tendency of QI due to physics	$kg/kg$
QS_t_phys	tendency of QS due to physics	$kg/kg$
QG_t_phys	tendency of QG due to physics	$kg/kg$
DENS_t_damp	tendency of density due to rayleigh damping	$kg/m^3/s$
MOMZ_t_damp	tendency of momentum z due to rayleigh damping	$kg/m^2/s^2$
MOMX_t_damp	tendency of momentum x due to rayleigh damping	$kg/m^2/s^2$
MOMY_t_damp	tendency of momentum y due to rayleigh damping	$kg/m^2/s^2$
RHOT_t_damp	tendency of rho*theta temperature due to rayleigh damping	$Kkg/m^3/s$
QV_t_damp	tendency of QV due to rayleigh damping	$kg/kg$
QC_t_damp	tendency of QC due to rayleigh damping	$kg/kg$
QR_t_damp	tendency of QR due to rayleigh damping	$kg/kg$
QI_t_damp	tendency of QI due to rayleigh damping	$kg/kg$
QS_t_damp	tendency of QS due to rayleigh damping	$kg/kg$
QG_t_damp	tendency of QG due to rayleigh damping	$kg/kg$

**scale\_atmos\_dyn (if CHECK\_MASS is defined in compilation process)**

Item/Variable	Description	Unit
MFLXZ	momentum flux of z-direction	$kg/m^2/s$
MFLXX	momentum flux of x-direction	$kg/m^2/s$
MFLXY	momentum flux of y-direction	$kg/m^2/s$
TFLXZ	potential temperature flux of z-direction	$K * kg/m^2/s$
TFLXX	potential temperature flux of x-direction	$K * kg/m^2/s$
TFLXY	potential temperature flux of y-direction	$K * kg/m^2/s$
ALLMOM_lb_hz	horizontally total momentum flux from lateral boundary	$kg/m^2/s$

**scale\_atmos\_dyn\_rk (if HIST\_TEND is defined in compilation process)**

Item/Variable	Description	Unit
DENS_t_adv cv	tendency of density (vert. advection)	$kg/m^3/s$
MOMZ_t_adv cv	tendency of momentum z (vert. advection)	$kg/m^2/s^2$
MOMX_t_adv cv	tendency of momentum x (vert. advection)	$kg/m^2/s^2$
MOMY_t_adv cv	tendency of momentum y (vert. advection)	$kg/m^2/s^2$
RHOT_t_adv cv	tendency of rho*theta (vert. advection)	$Kkg/m^3/s$
DENS_t_adv ch	tendency of density (horiz. advection)	$kg/m^3/s$
MOMZ_t_adv ch	tendency of momentum z (horiz. advection)	$kg/m^2/s^2$
MOMX_t_adv ch	tendency of momentum x (horiz. advection)	$kg/m^2/s^2$
MOMY_t_adv ch	tendency of momentum y (horiz. advection)	$kg/m^2/s^2$
RHOT_t_adv ch	tendency of rho*theta (horiz. advection)	$Kkg/m^3/s$
MOMZ_t_pg	tendency of momentum z (pressure gradient)	$kg/m^2/s^2$
MOMX_t_pg	tendency of momentum x (pressure gradient)	$kg/m^2/s^2$
MOMY_t_pg	tendency of momentum y (pressure gradient)	$kg/m^2/s^2$
MOMZ_t_ddiv	tendency of momentum z (divergence damping)	$kg/m^2/s^2$
MOMX_t_ddiv	tendency of momentum x (divergence damping)	$kg/m^2/s^2$
MOMY_t_ddiv	tendency of momentum y (divergence damping)	$kg/m^2/s^2$
MOMX_t_cf	tendency of momentum x (coliolis force)	$kg/m^2/s^2$
MOMY_t_cf	tendency of momentum y (coliolis force)	$kg/m^2/s^2$

**scale\_atmos\_phy\_mp\_kessler**

Item/Variable	Description	Unit
Vterm_QR	terminal velocity of QR	$m/s$

**scale\_atmos\_phy\_mp\_suzuki10**

Item/Variable	Description	Unit
aerosol	aerosol mass	$kg/m^3$

**scale\_atmos\_phy\_mp\_tomita08**

Item/Variable	Description	Unit
Vterm_QR	terminal velocity of QR	$m/s$
Vterm_QI	terminal velocity of QI	$m/s$
Vterm_QS	terminal velocity of QS	$m/s$
Vterm_QG	terminal velocity of QG	$m/s$

**scale\_atmos\_sub\_boundary**

Item/Variable	Description	Unit
DENS_BND	Boundary Density	<i>kg/m<sup>3</sup></i>
VELX_BND	Boundary velocity x-direction	<i>m/s</i>
VELY_BND	Boundary velocity y-direction	<i>m/s</i>
POTT_BND	Boundary potential temperature	<i>K</i>
VELZ_BND	Boundary velocity z-direction	<i>m/s</i>
QV_BND	Boundary QV	<i>kg/kg</i>
QC_BND	Boundary QC	<i>kg/kg</i>
QR_BND	Boundary QR	<i>kg/kg</i>
QI_BND	Boundary QI	<i>kg/kg</i>
QS_BND	Boundary QS	<i>kg/kg</i>
QG_BND	Boundary QG	<i>kg/kg</i>

**scale\_atmos\_sub\_boundary**

Item/Variable	Description	Unit
BND_ref_DENS	reference DENS	<i>kg/m<sup>3</sup></i>
BND_ref_VELZ	reference VELZ	<i>m/s</i>
BND_ref_VELX	reference VELZ	<i>m/s</i>
BND_ref_VELY	reference VELZ	<i>m/s</i>
BND_ref_POTT	reference VELZ	<i>K</i>

## 付録D SCALEに実装されている雲モデルの概要とそれぞれの選択方法

SCALE ライブラリには5種類の雲モデルが実装されている。氷雲の微物理過程を含まない1-moment バルクモデル (Kessler (1969))、氷雲の微物理過程を含む1-moment バルクモデル (Tomita (2008))、氷雲を含む2-moment バルクモデル (Seiki and Nakajima (2014))、1-moment ビン法 (Kentaroh et al. (2010))、及び、超水滴法 (Shima et al.) である。1-moment ビン法はオプションにより氷を含む場合とそうでない場合を切り替えられる。また超水滴法に関しては著作権の関係から公開されていない。使用したい場合はSCALEの開発者に連絡をされたい。これらの雲微物理モデルの違いは雲粒の粒径分布関数の表現方法にある。ここでは超水滴法を除く4種類の雲モデルの利用法について記述する。

### 1. 氷を含まない1-moment バルク法 Kessler (1969)

Kessler (1969) は後述の Tomita (2008) と同様に1-moment バルク法は粒径分布関数を粒径分布関数の3次のモーメント (質量) のみで表現する。Kessler (1969) は雲粒と雨粒のカテゴリを考慮し、雲粒混合比 (QC[kg/kg]) と雨粒混合比 (QR[kg/kg]) を予報する。

粒径分布関数はデルタ関数 (すなわち雲粒のサイズは全て同じと仮定する) で表現する。考慮する雲粒の成長過程は、雲粒生成 (飽和調整)、凝結・蒸発、衝突併合、落下である。

### 2. 氷を含む1-moment バルク法 Tomita (2008)

氷を含む1-moment バルク法は雲粒、雨粒、氷粒、雪片、あられの5カテゴリを考慮し、それぞれの混合比 (QC、QR、QI、QS、QG[kg/kg]) をそれぞれ予報する。粒径分布は雲物はデルタ分布 (半径10 $\mu$ m で一様)、その他は Marshall-Palmer 分布を仮定する。

考慮する成長過程は雲粒生成 (飽和調整)、凝結・蒸発、衝突併合、落下である。

### 3. 氷を含む2-moment バルク法 Seiki and Nakajima (2014)

氷を含む2-moment バルク法は混合比に加えて、粒径分布関数の0次のモーメント (数濃度) で表現する。

考慮する水物質のカテゴリは、雲粒、雨粒、氷粒、雪片、あられの5カテゴリであり、それぞれの質量混合比 (QC、QR、QI、QS、QG) と数密度混合比 (NC、NR、NI、NS、NG) を予報する。粒径分布は一般ガンマ関数で近似する。考慮する成長過程は雲粒生成、凝結・蒸発、衝突併合、分裂、落下である。

### 4. 1-moment ビン法 Kentaroh et al. (2010)

1-moment ビン法は粒径分布関数を差分化して陽に表現する。差分化された各粒径分布関数をビンと呼ぶためビン法と呼ばれる。各サイズのビンを質量のみで予報する (各ビンを質量及び数密度で表現するビン法を2-moment ビン法と呼ぶ)。考慮する水物質のカテゴリは雲粒、雨粒、氷粒、雪片、あられ、ひょうの6種類であり、考慮する成長過程は雲粒生成、凝結・蒸発、衝突併合、落下である。

上記の4種類の雲微物理は1～4の順に高精度になるが、その分計算コストも高くなる。次にこれらの雲物理スキームを用いて実験を行う方法を記述する。

## D.1 雲微物理スキームの選択方法

雲微物理スキームを選択するには、init.conf および run.conf の編集が必要である。まず init.conf をエディタで開き、TRACER\_TYPE を指定する（以下は氷を含む 2-moment バルク法を指定する場合の例である）。

```
&PARAM_TRACER
  TRACER_TYPE = 'SN14',
/
```

それぞれの雲微物理スキームを利用する際の TRACER\_TYPE は、次の表 D.1 に示すとおりである。

表 D.1: 雲微物理スキームの種類と設定値

雲微物理スキームの種類	TRACER_TYPE の設定値
氷を含まない 1-moment バルク法 ; Kessler (1969)	KESSLER
氷を含む 1-moment バルク法 ; Tomita (2008)	TOMITA08
氷を含む 2-moment バルク法 ; Seiki and Nakajima (2014)	SN14
1-moment ピン法 ; Kentaroh et al. (2010)	SUZUKI10

次に run.conf をエディタで開き TRACER\_TYPE と ATMOS\_PHY\_MP\_TYPE を指定する（以下は氷を含む 2-moment バルク法を指定する場合である）

```
&PARAM_TRACER
  TRACER_TYPE = 'SN14',
/

&PARAM_ATMOS
  ATMOS_PHY_MP_TYPE = "SN14",
/
```

この際、run.conf、init.conf に記述される TRACER\_TYPE と ATMOS\_PHY\_MP\_TYPE は全て同一でなければならない。

加えて、1-moment ピン法を用いる際は、init.conf に以下の項目を追加する必要がある。

```
&PARAM_MKINIT
  flg_bin = .true.
/

&PARAM_MKINIT_RF01
  PERTURB_AMP=0.02d0,
  RANDOM_LIMIT=5,
  RANDOM_FLAG=1
/

&PARAM_BIN
  nbin = 33,
  ICEFLG = 0,
  nccn = 0,
  kphase = 0,
/

&PARAM_SBMAERO
  FO_AERO = 1.D7,
  RO_AERO = 1.D-7,
  R_MAX = 1.D-6,
  R_MIN = 1.D-8,
  A_ALPHA = 3.D0,
  rhoa = 2.25D3,
/
```

それぞれの意味は「雲微物理スキームのオプションと Namelist」の章を参照されたい。

## D.2 実行方法

上記を設定して他の実験と同様に実行することで各雲微物理スキームを選択することができる。

## D.3 各雲物理スキームにのオプションと Namelist

現在執筆中...

## SCALE-LES USERS GUIDE [日本語版]

---

21, July, 2015. Version 1.0

06, November, 2015. Version 1.1

### 執筆・編集

---

Team SCALE ユーザーズガイド制作委員会 (UGC Working Group)

コアメンバー：足立 幸穂、佐藤 陽祐、富田 浩文、八代 尚、山浦 剛、吉田 龍二 (50 音順)

本書中に不明点やお気づきの点、ご要望がございましたら、SCALE user's メーリングリスト [scale-user@riken.jp](mailto:scale-user@riken.jp) までご連絡ください。