

# SCALE USERS GUIDE

Version 5.1.5

Team SCALE  
UGC working group

平成 29 年 6 月 16 日



# 目次

<b>第 1 章</b>	<b>概要</b>	<b>3</b>
1.1	はじめに	3
1.1.1	SCALE の特徴	3
1.1.2	SCALE-RM の構成	4
1.2	表記上の注意	7
<b>第 2 章</b>	<b>インストール</b>	<b>8</b>
2.1	必要なシステム環境	8
2.2	ライブラリ環境のインストール	9
2.3	SCALE のコンパイル	10
2.3.1	ソースコードの入手	10
2.3.2	Makedef ファイルと環境変数の設定	10
2.3.3	コンパイル	11
2.4	後処理ツール (net2g) のコンパイル	12
<b>第 3 章</b>	<b>動作確認と基本的な操作について</b>	<b>13</b>
3.1	概要	13
3.2	実行方法	13
3.2.1	前準備	13
3.2.2	初期値作成	14
3.2.3	モデル本体の実行	15
3.3	後処理と描画	15
3.4	この章の最後に	17
<b>第 4 章</b>	<b>簡単な現実実験の方法</b>	<b>19</b>
4.1	概要	19
4.2	入力データ (境界データ) の準備	19
4.3	実験セットの準備	22
4.4	地形データの作成 : pp	23
4.5	初期値・境界値データの作成 : init	25
4.6	シミュレーションの実行 : run	28
4.7	結果を描画する : net2g	31
<b>第 5 章</b>	<b>各種設定</b>	<b>35</b>
5.1	対象計算領域の設定	35
5.1.1	MPI プロセス数の設定	35
5.1.2	水平・鉛直格子数の設定	36
5.1.3	水平・鉛直格子間隔の設定	37

5.1.4	緩和領域とナッジングの設定	38
5.2	地図投影法と計算領域の位置の設定	40
5.3	地形の設定	41
5.4	様々な初期値・境界値データの作成方法	43
5.5	積分時間と積分時間間隔の設定	48
5.6	出力変数の追加・変更方法	49
5.7	力学スキームの設定	50
5.7.1	数値解法の設定	50
5.7.2	時間・空間差分スキームの設定	50
5.8	物理スキームの設定	52
5.8.1	雲微物理スキームの設定	52
5.8.2	乱流スキームの設定	53
5.8.3	放射スキームの設定	54
5.8.4	地表面 (大気下端境界) の設定	56
5.8.5	海洋モデルの設定	58
5.8.6	陸面モデルの設定	60
5.8.7	都市モデル (大気-都市面フラックス) の設定	61
5.9	ポスト処理	61
5.10	リスタート計算の方法	66
5.11	領域ネスティング実験の方法	67
5.11.1	子領域における地形の取り扱い	68
5.11.2	オフライン・ネスティング実験の方法	70
5.11.3	オンライン・ネスティング実験の方法	74
5.12	複数の実験を一括実行するバルクジョブの設定	78
5.13	設定ファイルと実験に必要なファイル一式を用意する	80
<b>付録 A ライブラリ環境のインストール</b>		<b>85</b>
A.1	インストール方法 (Linux - CentOS 6.6-6.8 編)	86
A.2	インストール方法 (Linux - CentOS 7.1-7.2 編)	88
A.3	インストール方法 (Linux - openSUSE 13.2 編)	90
A.4	インストール方法 (Mac OS X 編)	92
A.5	インストール方法 (スーパーコンピュータ「京」 編)	95
A.6	描画ツールのインストール	95
<b>付録 B 設定ファイル”run.conf”のネームリストと出力変数リスト</b>		<b>97</b>
<b>付録 C よくある質問とその回答 : FAQ</b>		<b>98</b>

# 第1章 概要

## 1.1 はじめに

本書は初めて領域気象気候モデル (SCALE-RM) を利用する人に向けた解説書です。気象気候ライブラリー SCALE version 5.1.5 に対応した説明を記載します。現在、SCALE を利用するため、領域モデル SCALE-RM と全球モデル SCALE-GM が用意されていますが、後者は、力学コアの整備のみされています。そのため、本版では、SCALE-RM の使い方についてのみ詳しく述べています。SCALE-GM については、次版で詳しく記載される予定です。

第1章で SCALE の概要について、第2章で必要な環境、およびインストール方法について説明します。続いて、第3章で基本的な操作方法、第4章で現実大気実験の実行方法について簡単な例を示しながら説明します。第2章から第4章まではひと繋がりチュートリアルとなっており、SCALE-RM を初めて使うユーザは一通り通読することをお勧めします。第5章以降は、設定の変更、機能やツールの説明が記載されています。

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 (Scalable Computing for Advanced Library and Environment) は計算機を用いて気象・気候科学計算を行う上で、研究を進めやすいようにプレ処理から数値シミュレーション、ポスト処理、解析に至るまですべての過程を網羅する気象・気候数値計算ライブラリを目指したソフトウェアである。下記に挙げるような特徴を持つ。

- SCALE は、「BSD-2 ライセンス」のもとオープンソース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・再配布が可能である。
- SCALE には、SCALE-RM(SCALE-Regional Model)、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、行いたい実験に合わせて選択利用することが可能である。
- SCALE で提供されている物理過程は、他の数値モデルへ組み込んで使用することも可能である。

ライセンスの詳細は、トップディレクトリ直下の `scale-5.1.5/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 ページにもソフトウェアの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/>)。

本節の以下では SCALE の思想とモデルの関係について説明するが、SCALE-RM 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。



## SCALE のライブラリとモデルの関係について

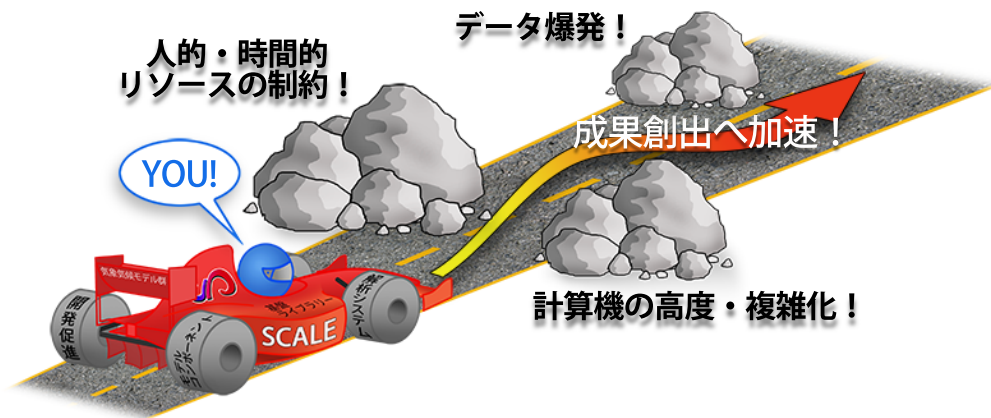


図 1.1: SCALE のねらい

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・気候科学計算向けのライブラリである。図 1.1 に SCALE の思想の概念図を示す。この図に示されるような諸問題に対応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小型 PC クラスタ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。そのため、スーパーコンピュータ「京」や富士通 FX10 等のスーパーコンピュータに加え、インテル機のような汎用計算機でも計算効率がでることを目指して設計されている。

SCALE を利用した数値モデルとして、SCALE-RM が含まれている (図 1.2)。並列プロセスの管理、ファイル入出力、プロセス間通信や格子情報の設定は SCALE が提供する。大気流体の支配方程式を解く部分（流体力学コア、力学過程）と雲微物理や大気放射のような諸物理過程を解く部分も SCALE が提供する。SCALE-RM は入力された大気状態のデータ（予報変数）を保持し、SCALE が提供する機能を組み合わせ、各コンポーネントを適切に呼び出すことで時間発展計算を行っている。ユーザは必要に応じて利用するコンポーネントを選択し、組み合わせでシミュレーションを行うことができる。

### 1.1.2 SCALE-RM の構成

現在、SCALE-RM には下記のコンポーネントが実装されている。SCALE-RM ではすべてのコンポーネントが利用可能である。詳細なモデル構成や差分化手法については、[Team SCALE \(2015\)](#)、[Sato et al. \(2015\)](#)、および [Nishizawa et al. \(2015\)](#) を参照されたい。

#### フレームワーク

- 距離座標に基づいた三次元カーテシアン格子系
- MPI 通信を用いた二次元領域分割
- 各種地図投影法

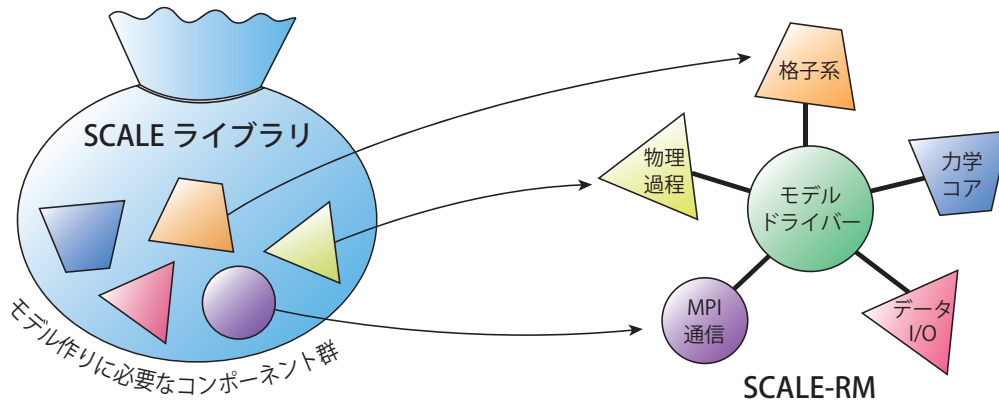


図 1.2: SCALE と SCALE-RM の関係

- ネスティングシステム (1 way : 親領域 → 子領域)
  - オンライン実行 (複数ドメインの計算を同時に実行)
  - オフライン実行 (外側ドメインの計算終了後に、その結果を用いて内側ドメインの計算を行う)
- 複数事例一括実行 システム (バルクジョブシステム)
- CF 規約<sup>\*1)</sup>に基づく netCDF ファイル I/O
  - netCDF3 または netCDF4 形式
- 理想実験のための初期値データ生成
- 外部データ読み込みによる標高・土地利用区分データの変換作成
- 外部データ読み込みによる初期値・境界値データ変換作成
  - WRF-ARW<sup>\*2)</sup>、GrADS<sup>\*3)</sup>フォーマットでの入力に対応

#### 力学コア関係

- 方程式系: 3次元完全圧縮非静力学方程式系
- 数値解法: 陽解法と陰解法の両方を実装
  - 水平陽解法-鉛直陽解法
  - 水平陽解法-鉛直陰解法
- 空間差分: フラックス形式
  - 2次中央差分
  - 4次中央差分

\*1) <http://cfconventions.org/>

\*2) <http://www.wrf-model.org/>

\*3) <http://cola.gmu.edu/grads/>

- 6次中央差分
- 3次風上差分
- 5次風上差分
- 時間差分:(詳細は、[Team SCALE \(2015\)](#) を参照のこと)
  - Heun 型 3次ルンゲクッタスキーム
  - [Wicker and Skamarock \(2002\)](#) の 3段ルンゲクッタスキーム
  - 4次ルンゲクッタスキーム
- 非負保証:
  - フラックス修正法 (Flux Corrected Transport, FCT; [Zalesak, 1979](#))
  - [Koren \(1993\)](#) フィルター (3次風上差分スキーム使用時のみ)
- 数値フィルター: 4次超粘性・拡散
- 地形: 地形に沿った座標系

#### 物理過程

- 乱流過程: 複数から選択可能
  - [Smagorinsky \(1963\)](#) & [Lilly \(1962\)](#) 型のサブグリッドモデル ([Brown et al. \(1994\)](#) と [Scotti et al. \(1993\)](#) による補正)
  - [Deardorff \(1980\)](#) サブグリッドモデル
  - [Mellor and Yamada \(1982\)](#); [Nakanishi and Niino \(2004\)](#) による MYNN2.5 境界層モデル
- 雲微物理: 複数から選択可能
  - [Kessler \(1969\)](#) による 3-class 1 モーメントバルクモデル
  - [Tomita \(2008\)](#) による 6-class 1 モーメントバルクモデル
  - [Seiki and Nakajima \(2014\)](#) による 6-class 2 モーメントバルクモデル
  - [Suzuki et al. \(2010\)](#) によるビン法モデル
- 放射過程: [Sekiguchi and Nakajima \(2008\)](#) による相関 k 分布法ブロードバンド大気放射伝達モデル
- 地表面モデル
  - 陸面モデル: 熱拡散・バケツモデル
  - 海洋モデル: 初期値固定・外部データ入力・スラブモデル
  - 都市モデル: [Kusaka et al. \(2001\)](#) による単層キャノピーモデル
  - バルク交換係数 (陸面および海面): [Beljaars and Holtslag \(1991\)](#); [Wilson \(2001\)](#) による普遍関数によるバルク法もしくは [Uno et al. \(1995\)](#) による Louis 型バルク法

## 1.2 表記上の注意

本書中では、Unix での bash 上での実行を想定して記述している。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
```

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を表す。

```
----- 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
```

また、本書では、ネームリストとその項目についての記述を以下のように表記するものとする。

ネームリスト **[namelist]**

ネームリストの中の項目 (**item\_of\_namelist**)



## 第2章 インストール

本章では、SCALE と SCALE-RM のコンパイル及び、実行に必要な環境について説明する。

### 2.1 必要なシステム環境

#### 推奨のシステム環境

- システムハードウェア構成  
必要なハードウェアは実験設定に依存するが、ここでは第3章と第4章のチュートリアル実行に必要なスペックを示す。
  - 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必要(倍精度浮動小数点使用時)。
  - HDD : 現実大気実験には約3GBのディスク空き容量が必要。
- 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、出力ファイルサイズが大きくなる。

#### あると便利なシステム環境

- データ変換ツール : wgrib、wgrib2やNCLにより、SCALE-RMで読込可能な入力データの作成が可能である。チュートリアルの現実大気実験では、wgribを使用する。
- 描画環境 : GrADS<sup>\*1)</sup>、GPhys/Ruby-DCL<sup>\*2)</sup>、ncview<sup>\*3)</sup>など。
- 演算性能評価 : PAPIライブラリ<sup>\*4)</sup>が使用可能。

\*1) <http://cola.gmu.edu/grads/>

\*2) <https://www.gfd-dennou.org/arch/ruby/products/gphys/>

\*3) [http://meteora.ucsd.edu/~piece/ncview\\_home\\_page.html](http://meteora.ucsd.edu/~piece/ncview_home_page.html)

\*4) <http://icl.utk.edu/papi/>

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、6.8、7.0、7.1、7.2	
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	
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、4.9.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を推奨

SCALE のソースコードは Fortran2003 規格に基づく機能を利用している。そのため、Fortran2003 の基本的な機能をサポートしたコンパイラが必要となる。例えば GNU gfortran の場合、バージョン 4.3 以前では Fortran2003 規格のサポートが不十分であるため、SCALE の利用環境として用いることは出来ない。

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、1.10.3	
Intel MPI	4.0 Update 3、4.1.0、5.0	2013 年以降のバージョンを推奨
SGI MPT	2.05、2.09	Intel Compiler との組み合わせで確認済み

以上で説明したもの以外に、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 の環境でも動作確認されている。

## 2.2 ライブラリ環境のインストール

SCALE ライブラリに必要な各種ライブラリ (Fortran および C コンパイラ、MPI、NetCDF、HDF5 等) のインストールを行う。詳細は、付録 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前提として進める。

描画ツールとして、第 3 章、第 4 章のチュートリアルではクイックビューのため GPhys を使用する。また、GrADS を使った結果の描画方法についても紹介する。これらの描画ツールの詳細とインストール方法については付録 A.6 を参照のこと。

## 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
- GNU C/C++、Fortran compiler (付録 A 章参照)

### 2.3.1 ソースコードの入手

最新のリリース版ソースコードは、

<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-5.1.5/` というディレクトリができる。

```
$ tar -zxvf scale-5.1.5.tar.gz
$ ls ./
scale-5.1.5/
```

### 2.3.2 Makedef ファイルと環境変数の設定

SCALE はコンパイルするとき、環境変数 “SCALE\_SYS” に設定した Makedef ファイルを使用してコンパイルが行われる。Makedef ファイルは、`scale-5.1.5/sysdep/` 内に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。動作確認済みの環境と対応する Makedef ファイルを表 2.4 に示す。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各自作成する。

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-ompi
	icc/ifort	intelMPI	Makedef.Linux64-intel-impi
	icc/ifort	SGI-MPT	Makedef.Linux64-intel-mpt
Mac OS X	gcc/gfortran	openMPI	Makedef.MacOSX-gnu-ompi
スーパーコンピュータ「京」	fccpx/frtpx	mpiccpx/mpifrtpx	Makedef.K
Fujitsu PRIME-HPC FX10	fccpx/frtpx	mpiccpx/mpifrtpx	Makedef.FX10

Linux OS、GNU コンパイラ、openMPI を使用する場合には、“Makedef.Linux64-gnu-ompi” が対応するファイルとなる。別の環境でインストールを行っている場合には、適宜、表 2.4 に従って読み替えること。下記の通り、環境変数を設定する。

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

実行環境が常に同じであるならば、環境変数の設定を `.bashrc` などの環境設定ファイルに記述しておくとう便利である。

各種ライブラリのインストールを付録 A 章に従って行った場合を除き、下記の PATH の設定が必要となる。HDF5 と netCDF4 については、下記のように環境変数を設定する。例えば、Intel コンパイラを利用して、HDF5 を `/opt/hdf5`、netCDF4 を `/opt/netcdf` にそれぞれインストールした場合の例を示す。適宜、各自の環境に応じて読み替えて設定すること。

```
$ export HDF5="/opt/hdf5"
$ export NETCDF4="/opt/netcdf"
$ export NETCDF_INCLUDE="-I/opt/netcdf/include"
$ export NETCDF_LIBS="-L/opt/hdf5/lib64 -L/opt/netcdf/lib64 \
-lnetcdf -ldf5_h1 -lhdf5 -lm -lz"
```

### 2.3.3 コンパイル

SCALE-RM ソース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-5.1.5/scale-rm/src
$ make -j 4
```

`make` のあとの `"-j 4"` は、コンパイル時の並列数 (例では 4 並列) を示しており、実行環境によって並列数を指定すれば良い。コンパイルが成功すると下記 3 つの実行ファイルが `scale-5.1.5/bin` 以下に生成される。

```
scale-rm  scale-rm_init  scale-rm_pp
```

また、コンパイルをやり直したい場合など、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、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。コンパイル環境、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 は、`scale` の TOP ディレクトリ直下の `scale-5.1.5/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る。SCALE-RM は、コンパイルを実行したディレクトリの下に `".lib"` という名前の隠しディレクトリの中にコンパイルされたオブジェクトファイルが置かれる。
- 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 (net2g) のコンパイル

SCALE-RM の出力ファイルは、ノードごとに分割されて出力される。SCALE では、これら出力ファイル (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」を提供している。第 3 章、第 4 章のチュートリアルでも使用する。ここでは、net2g のコンパイル方法について説明する。

まず、SCALE 本体のコンパイル時と同様に、使用する環境にあった Makedef ファイル設定のための環境変数を設定する (2.3.3 節コンパイル参照)。次に、net2g のディレクトリに移動し、make する。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-5.1.5/scale-rm/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。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 動作確認と基本的な操作について

### 3.1 概要

本章では、SCALE-RM を用いて一連の実験を行うための基本的な操作について、チュートリアル用に準備した理想実験のテストケースを題材に説明する。

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兼ねているのでぜひ実施してもらいたい。

本章では、SCALEのコンパイルが正常に終了し、すでに下記のファイルが生成されているものとして説明を行う。

```
scale-5.1.5/bin/scale-rm
scale-5.1.5/bin/scale-rm_init
scale-5.1.5/scale-rm/util/netcdf2grads_h/net2g
```

これらに加えて、描画ツールとしてGrADSを使用する。オプションとして、gpviewは、結果の確認用に利用することができる。GrADSおよびgpview(GPhys)との詳細やインストール方法については、付録A.6節を参照のこと。

### 3.2 実行方法

実験の実行は、前準備、初期値作成、シミュレーション実行、後処理、そして描画の順番で作業を進める。

#### 実験設定

チュートリアル理想実験として、積雲対流の理想実験を用意している。この実験では、積乱雲が発生するときの典型的な大気のプロファイルと対流圏下層に初期擾乱を与え、積乱雲が発達する様子を準2次元モデルで実験する内容となっている。実験設定を表3.1に示す。

#### 3.2.1 前準備

チュートリアル理想実験は、scale-rm/test/tutorial/idealのディレクトリにて実行する。このディレクトリに移動し、scale-5.1.5/binにある実行バイナリへの静的リンクを張る。

```
$ cd scale-rm/test/tutorial/ideal
$ ln -s ../../../../bin/scale-rm ./
$ ln -s ../../../../bin/scale-rm_init ./
```

“scale-rm”はモデル本体、“scale-rm\_init”は初期値・境界値作成ツールである。

表 3.1: 本章での理想実験の実験設定

項目	設定内容	備考
MPI プロセス数	東西 : 2、南北 : 1	計 2 プロセスでの並列計算を行う
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 <sup>a</sup>	
鉛直層数	97 層 (トップ : 20 km)	下層ほど細かい層厚をもつストレッチ設定である
側面境界条件	周期境界	東西、南北境界とも
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, 2008)
初期鉛直プロファイル	GCSS Case1 squall-line (Redelsperger et al., 2000)	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

<sup>a</sup>現在は 2 次元実験を行うための枠組みは用意されていないが、Y 方向に同じ値をもつ初期値を与える事で 2 次元実験に相当する実験を行うことが可能である。この場合、ハロの格子数と同じ数の格子数を Y 方向に設定する必要がある。ハロの必要格子数については 5.7.2 参照。

### 3.2.2 初期値作成

初期値の作成は、`scale-rm_init` に設定ファイルを与えて実行する。`init_R20kmDX500m.conf` には、表 3.1 に対応した実験設定が書き込まれている。この設定ファイルを `scale-rm_init` に与えて実行することで、設定ファイルの指示に従って大気の成層構造を計算し、初期擾乱が設定される。

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] [設定ファイル]
```

[プロセス数] の部分には MPI 並列で使用したいプロセス数を記述する。[実行バイナリ] には、`scale-rm` や `scale-rm_init` が入る。そして、実験設定を記述した設定ファイルを [設定ファイル] の部分に指定する。例えば、`init_R20kmDX500m.conf` は 2-MPI 並列 (2 つの MPI プロセス) を使って計算するように指定されている。`scale-rm_init` を実行するコマンドは次のように記述する。

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

実行が成功した場合、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-RM
*** Execute preprocess? : T
*** Execute model? : F
*** a single communicator
*** a single communicator
*** End Launch System for SCALE-RM
```

この実行によって、

```
init_LOG.pe000000
init_00000101-000000.000.pe000000.nc
init_00000101-000000.000.pe000001.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る。ログファイル (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。ファイル名の pe の後の数字は、MPI のプロセス番号を示している。例では2つの MPI プロセスを利用しているが、デフォルト設定では0番目のプロセス (マスターランク) のログファイルだけが出力される。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

```
+++++++ Finalize MPI...
+++++++ MPI is peacefully finalized
```

と記述される。

init\_00000101-000000.000.pe000000.nc と init\_00000101-000000.000.pe000001.nc の2つのファイルは初期値ファイルであり、それぞれおおよそ 100KB のファイルサイズになる。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 ncvview 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

プロセス並列数は、初期値作成のときと同じ数を指定する。設定ファイルには run\_R20kmDX500m.conf を指定する。

```
$ mpirun -n 2 ./scale-rm 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.pe000001.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。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

```
+++++++ Finalize MPI...
+++++++ MPI is peacefully finalized
```

と記述される。history.pe000000.nc と history.pe000001.nc の2つのファイルが計算結果が記録された history ファイルであり、それぞれおおよそ 1.4MB のファイルサイズになる。2-MPI 並列で実行したため、2つのファイルが生成されており、ファイル形式は NetCDF である。

### 3.3 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。

まず、2.4節でコンパイルした後処理ツールnet2gへリンクを張る。

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

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 [設定ファイル]
```

net2g専用のnet2g.confを設定ファイルとして与えて、次のように実行する。

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

エラーメッセージがなく、下記のメッセージだけが標準出力へ表示されていれば正常に変換完了である。

```
+++ MPI COMM: Corrective Finalize
```

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。この実行によって、下記6つのファイルが、実行ディレクトリ下に作成される。

```
QHYD_d01z-3d.ctl
QHYD_d01z-3d.grd
U_d01z-3d.ctl
U_d01z-3d.grd
W_d01z-3d.ctl
W_d01z-3d.grd
```

これらのファイルはそれぞれ、分割ファイルを1つにまとめ、U（水平風東西成分）、W（鉛直風）、QHYD（全凝結物の質量比）の変数についてダイレクトアクセスの単純バイナリ形式（GrADS形式）に変換した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。

計算がうまくいっているかを確認するため、GrADSスクリプトcheckfig\_ideal.gsを使って作図する。なお、GrADSのバージョンによっては文法が異なるので、Warningが出る場合は適宜GrADSスクリプトを変更する。

```
$ grads -blc checkfig_ideal.gs
```

作図が成功すると、下記の図が生成される。

```
ideal_QHYD.png
ideal_W.png
```

計算と変換が成功していれば、図3.1と同じ図が描画される。

他の変数についてもバイナリデータに変換したい場合には、net2g.confの[VARI]の(VNAME)に必要な変数を追加すればよい。

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

history ファイルに出力されている変数は、netCDF の `ncdump` などを使えば簡単に調べることができる。net2g の詳しい使用方法は、[5.9](#) を参照してほしい。

### 3.4 この章の最後に

本章では、スコールラインの理想実験を例に、SCALE の実行方法について説明した。次ステップとして、解像度や計算領域、MPI プロセス数の変更放射過程や乱流過程、雲微物理スキームといった物理過程の変更を試すことをお勧めする。これらの変更方法は、[第 5 章](#)に記載されている。

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、使用する物理スキームについて変更を加えた設定ファイルのサンプルが用意されているので、これらも参考となる。また、SCALE には各種理想実験セットが “scale-rm/test/case” 以下に複数用意されている。実験設定によってはテストケースに特化したソースコードの入れ替えを行っている場合があるため、これらの理想実験セットでは設定ファイルのあるディレクトリで `make` コマンドを実行する必要がある。初期値作成と実行の手順は基本的に本章のチュートリアルと同じである。



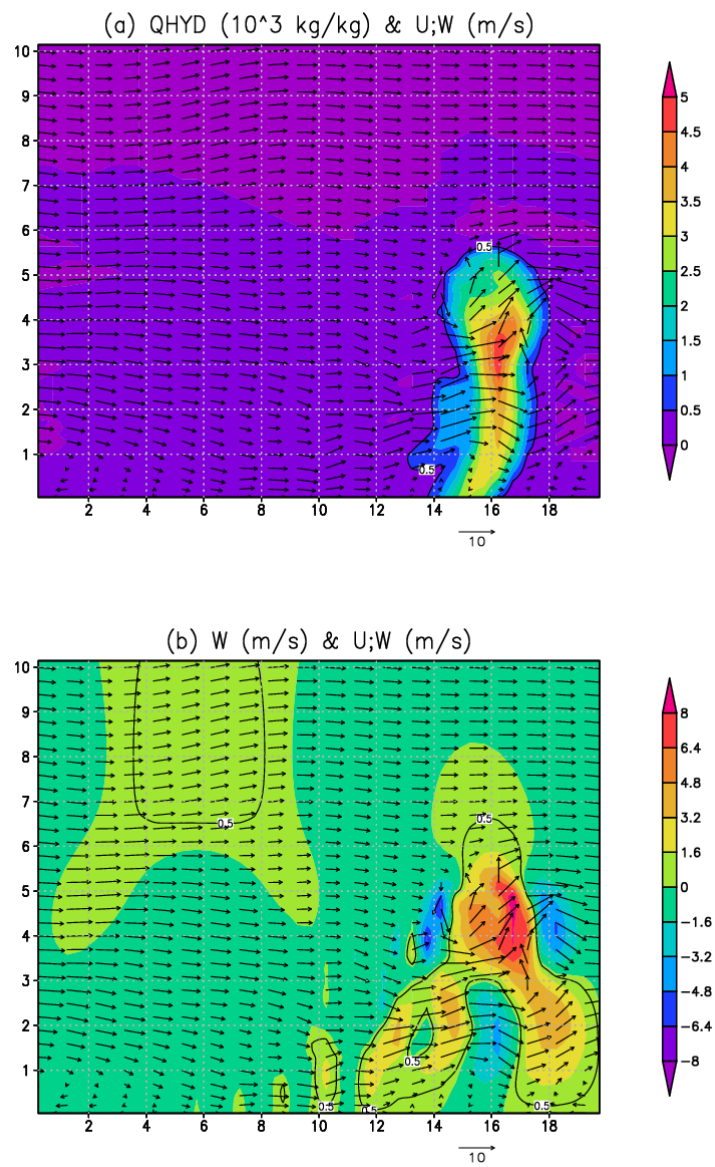


図 3.1: 積分開始後 1200 sec (20 minute) の  $Y=750\text{m}$  における東西-鉛直断面図; (a) のカラーシェードは全質量に対する凝結物の質量比、(b) は鉛直速度をそれぞれ示す。ベクトルは東西-鉛直断面内の風の流れを表す。

## 第4章 簡単な現実実験の方法

### 4.1 概要

本章では、チュートリアルとして準備した現実大気実験の基本的な実行手順を習得する。現実大気実験は、次の流れ (図 4.1) に従って実行する。

1. 入力データの準備 (基本各自で準備。チュートリアルでは tools/ で行う。)
2. pp : 地形データの作成
3. init : 初期値・境界値データの作成
4. run : シミュレーションの実行
5. net2g : 出力データの netCDF から GrADS 形式への変換 (オプション)

これ以降の説明では、scale-5.1.5/scale-rm/test/tutorial/の絶対パスを $\${Tutorial\_DIR}$ と示すこととする。

チュートリアルの現実大気実験の計算領域 (ドメイン) の設定は表 4.1 のようになっている。図 4.2 に対象領域を示す。このチュートリアルは、SCALE-RM 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ご留意頂きたい (例えば、20km の水平解像度で積雲パラメタリゼーションなし)。

### 4.2 入力データ (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-RM 本体に与える境界値データが必要になる。境界値作成のための外部入力データとして表 4.2 のデータが必要である。青字は必須の変数、その他は任意である。

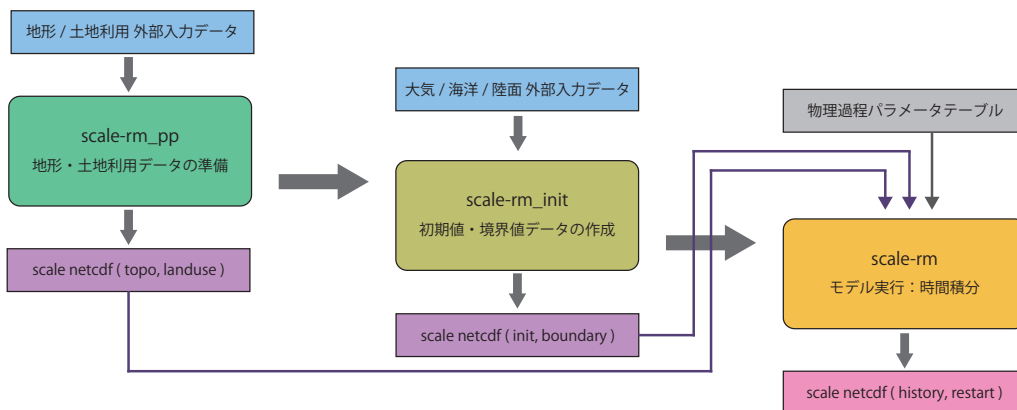


図 4.1: SCALE-RM モデルの実行過程

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)	90 格子点 x 90 格子点
鉛直層数	36 層
水平格子間隔	$dx = dy = 20\text{km}$
積分期間	2007 年 7 月 14 日 18UTC~15 日 00UTC (6 時間積分)
時間ステップ間隔	90 sec (240 steps)

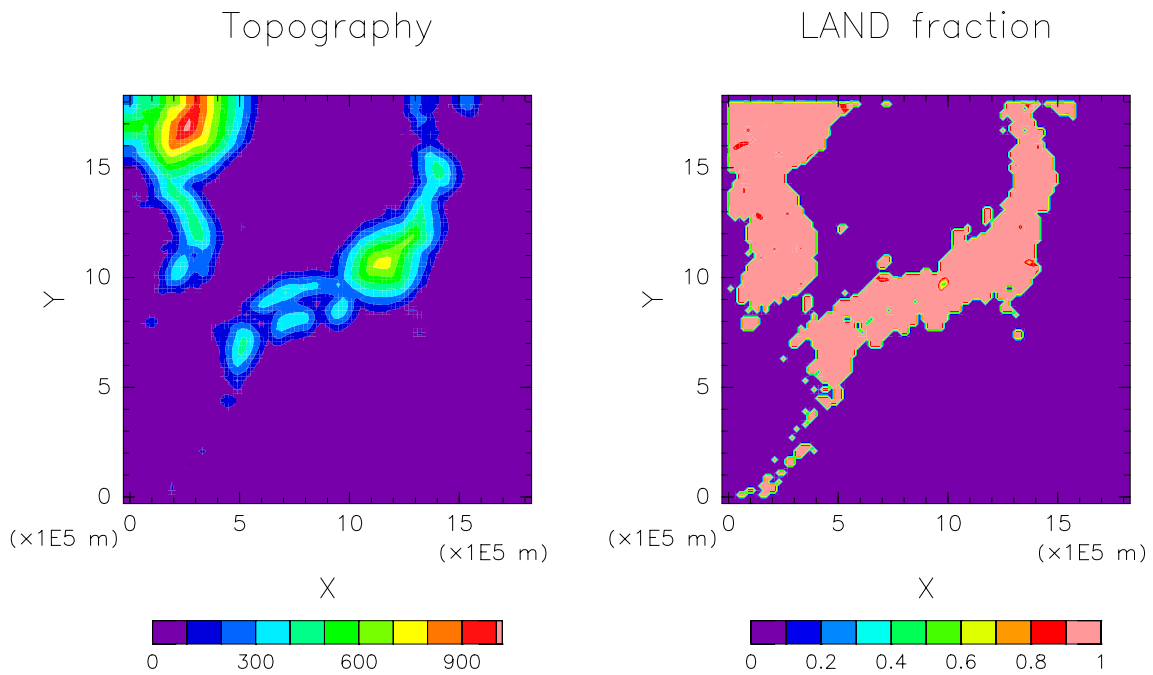


図 4.2: 計算領域の地形と海陸分布。

### 標高データと土地利用区分データ

標高データと土地利用区分データは実験設定に従って、SCALE-RM のそれぞれの格子点における標高、海陸比率、湖比率、都市被覆率、植生比率、土地 (植生) 利用区分を作成するために使用する。ユーザが全球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、土地利用区分データ GLCCv2 を提供している。

#### 1. データのダウンロード

SCALE-RM 用の標高・土地利用区分のデータを

[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。展開したディレクトリには、標高データと土地利用区分データが格納されている。

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

表 4.2: 現実大気実験に必要な外部入力データ

SCALE-RM の地形と土地利用を作成するための元データ
<ul style="list-style-type: none"> <li>標高データ</li> <li>土地利用区分データ</li> </ul>
SCALE-RM の初期値境界値を作成するための外部入力データ (一般的には GCM データ)
<ul style="list-style-type: none"> <li>親モデルの緯度・経度情報 <ul style="list-style-type: none"> <li>(3次元大気データ) <ul style="list-style-type: none"> <li>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</li> </ul> </li> <li>(2次元大気データ) <ul style="list-style-type: none"> <li>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)</li> </ul> </li> <li>(2次元陸面データ) <ul style="list-style-type: none"> <li>親モデルの海陸マップ</li> <li>地表面温度 (Skin temp)</li> <li>親モデル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)</li> </ul> </li> <li>(2次元海面データ) <ul style="list-style-type: none"> <li>海面水温 (Skin temp がある場合は省略可)</li> </ul> </li> </ul> </li> </ul>

```
scale_database/topo/    <- 標高データ
scale_database/landuse/ <- 土地利用区分データ
```

## 2. パスの設定

現実大気実験の実験に必要なファイル一式の準備には、make を用いた「実験セット一式作成ツール」を用いる。このツールを利用するためには、標高・土地利用区分データの tar ファイルの展開先ディレクトリを、SCALE\_DB という環境変数に設定しておくことが必須である (以後、`SCALE_DB` と表記)。

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

ここで、`path_to_directory_of_scale_database` は、データベースがあるディレクトリである。

## 大気・陸面・海面水温データ

初期値境界値データは 4byte バイナリ (GrADS 形式、以降 “binary 形式” と表記する) 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。基本的に、バイナリデータはユーザ自身が用意する。チュートリアルでは NCEP FNL(Final) Operational Global Analysis data を使用方法を示す。あらかじめ `wgrib` をインストールしておく<sup>\*1)</sup>。

### 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、2007 年 7 月 14 日 18 時から一日分の grib1 フォーマットのデータ

\*1) <http://www.cpc.ncep.noaa.gov/products/wesley/wgrib.html>

```
fnl_20070714_18_00.grib1
fnl_20070715_00_00.grib1
fnl_20070715_06_00.grib1
fnl_20070715_12_00.grib1
```

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 形式から binary 形式に変換

``${Tutorial_DIR}/real/tools/``にある `convert_grib2grads_FNLgrib1.sh` を実行。

```
$ cd `${Tutorial_DIR}/real/tools/
$ sh convert_grib2grads_FNLgrib1.sh
```

成功すれば、下記のファイルが作成される。

```
$ ls FNL_output/**
FNL_output/200707/FNLatm_2007071418.grd
FNL_output/200707/FNLatm_2007071500.grd
FNL_output/200707/FNLatm_2007071506.grd
FNL_output/200707/FNLatm_2007071512.grd
FNL_output/200707/FNLland_2007071418.grd
FNL_output/200707/FNLland_2007071500.grd
FNL_output/200707/FNLland_2007071506.grd
FNL_output/200707/FNLland_2007071512.grd
FNL_output/200707/FNLsfc_2007071418.grd
FNL_output/200707/FNLsfc_2007071500.grd
FNL_output/200707/FNLsfc_2007071506.grd
FNL_output/200707/FNLsfc_2007071512.grd
```

### 4.3 実験セットの準備

現実大気実験は、理想実験よりも実行手続きや必要なファイルが多く、プレ処理 `pp`、初期値作成 `init`、シミュレーション実行 `run` で使用する設定ファイル (`***.conf`) 内の実験設定を統一する必要がある。準備段階でのファイルの不足や設定の不一致はモデルが正常に動かない原因となる。これを回避するため、現実実験の実行に必要なファイル一式を用意するためのツール「実験セット一式作成ツール」が用意されている。まずはじめに、以下のディレクトリに移動し、次の手続きにより現実大気実験チュートリアルのためのファイル一式を用意する。

```
$ cd `${Tutorial_DIR}/real/
$ ls
Makefile : 実験セット一式作成のための Makefile
README   : 実験セット一式作成ツールに関する README
USER.sh  : 実験設定の記述
config/  : 実験セット一式作成一つのためのファイル (ユーザは基本書き換えない)
sample/  : サンプル USER.sh スクリプト
data/    : チュートリアルのためのツール類
```



```
tools/      : チュートリアル用の入力大気データ用意のためのツール（基本各自で準備）
$ make
$ ls experiment/      : make により追加
  init/
  net2g/
  pp/
  run/
```

make を実行すると、USER.sh に記述された設定に従って、experiment ディレクトリの下に実験セットが作成される。実験セット一式準備ツールに関する詳しい説明については、第 5.13 節を参照いただきたい。なお、sample ディレクトリにはネスティングの際に利用できるファイルが用意されており、必要に応じて参考にされたい。

## 4.4 地形データの作成：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を作成する。

```
$ cd ${Tutorial_DIR}/real/experiment/pp/
$ ls
  pp.d01.conf
  scale-rm_pp
```

pp ディレクトリの中には、pp.d01.conf という名前の設定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.d01.conf を編集する必要があるが、チュートリアルではすでに表 4.1 の設定に従って編集済みの pp.d01.conf が用意されているため、そのまま利用する。

pp.d01.conf のネームリストのうち、領域に関する設定は [PARAM\_PRC]、[PARAM\_INDEX]、[PARAM\_GRID] で行っている。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 プロセスを使用する設定となっている。1 つの MPI プロセスあたりの格子点数については、(IMAX = 45)、(JMAX = 45) と指定されているため、X 方向、Y 方向の総格子点数は、ともに  $2 \times 45 = 90$  である。計算ドメインの大きさは、[PARAM\_GRID] の (DX, DY) はともに 20000 m (20 km) と指定されており、一辺の長さが  $90 \times 20$  km より、 $1800$  km  $\times$   $1800$  km の正方形の計算領域が設定されている。

```

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

&PARAM_INDEX
  KMAX = 36,
  IMAX = 45,
  JMAX = 45,
/

&PARAM_GRID
  DX = 20000.0,
  DY = 20000.0,
  FZ(:) = 80.841, 248.821, 429.882, 625.045, 835.409, 1062.158,
          1306.565, 1570.008, 1853.969, 2160.047, 2489.963, 2845.575,
          3228.883, 3642.044, 4087.384, 4567.409, 5084.820, 5642.530,
          6243.676, 6891.642, 7590.074, 8342.904, 9154.367, 10029.028,
          10971.815, 11988.030, 13083.390, 14264.060, 15536.685, 16908.430,
          18387.010, 19980.750, 21698.615, 23550.275, 25546.155, 28113.205,
  BUFFER_DZ = 5000.0,
  BUFFER_DX = 400000.0,
  BUFFER_DY = 400000.0,
/

```

さらに、scale-rm\_pp 専用の設定として [PARAM\_CONVERT] の項目がある。(CONVERT\_TOPO) と (CONVERT\_LANDUSE) に .true. を設定すると、標高データと土地利用区分データのそれぞれの処理が行われる。

```

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

```

また、[PARAM\_CNVTPOPO\_GTOPO30] の中の (GTOPO30\_IN\_DIR) と [PARAM\_CNVLANDUSE\_GLCCv2] の中の (GLCCv2\_IN\_DIR) は、標高データと土地利用区分データの場所を示しており、環境変数 \${SCALE\_DB} で設定されたデータベースが指定されている。

```

&PARAM_CNVTPOPO_GTOPO30
  GTOPO30_IN_CATALOGUE = "GTOPO30_catalogue.txt",
  GTOPO30_IN_DIR = "./topo/GTOPO30/Products",
/

&PARAM_CNVLANDUSE_GLCCv2
  GLCCv2_IN_CATALOGUE = "GLCCv2_catalogue.txt",
  GLCCv2_IN_DIR = "./landuse/GLCCv2/Products",
  limit_urban_fraction = 0.3D0,
/

```

今回は、表 4.1 に示すとおり、4つの MPI プロセスを使用する設定であるので次のように実行し、

地形データを作成する。

```
$ mpirun -n 4 ./scale-rm_pp pp.d01.conf
```

ジョブが正常に終了している場合は、ログファイル (pp\_LOG\_d01.pe000000) の最後に

```
+++++++ Finalize MPI...
+++++++ MPI is peacefully finalized
```

と出力される。また、topo\_d01.pe#####.nc (約 180KB のファイルサイズ) と landuse\_d01.pe#####.nc (約 220KB のファイルサイズ) 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、各格子点における標高、海陸比率、湖比率、都市被覆率、植生比率、土地 (植生) 利用区分の情報が入っている。

## 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*@TOPO --aspect=1 --nocont
$ gpview landuse_d01.pe00000*@FRAC_LAND --aspect=1 --nocont
```

## 4.5 初期値・境界値データの作成 : init

init ディレクトリでは、SCALE-RM 計算に必要な初期値・境界値データを作成する。

```
$ cd ${Tutorial_DIR}/real/experiment/init
$ ls
  init.d01.conf
  init.launch.conf
  param.bucket.conf
  scale-rm_init
```

ディレクトリの中には、init.d01.conf という名前の設定ファイルが準備されている。

他に init.launch.conf というファイルも作成されているが、ここでは使用しない。pp.d01.conf と同様に、実験設定に合わせて、この init.d01.conf を書き換える必要があるが、チュートリアル用の init.d01.conf ファイルは表 4.1 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データを利用する。これは、init.d01.conf の中で、下記のように相対パス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.d01.conf` の設定の中で特に確認してほしいのは、`[PARAM_MKINIT_REAL_ATMOS]`、`[PARAM_MKINIT_REAL_OCEAN]`、`[PARAM_MKINIT_REAL_LAND]` の内容である。

```

&PARAM_MKINIT_REAL_ATMOS
NUMBER_OF_FILES = 2,
FILETYPE_ORG = "GrADS",
BASENAME_ORG = "namelist.grads_boundary.FNL.grib1",
BASENAME_BOUNDARY = "boundary_d01",
BOUNDARY_UPDATE_DT = 21600.0,
PARENT_MP_TYPE = 3,
USE_FILE_DENSITY = .false.,

/

&PARAM_MKINIT_REAL_OCEAN
..... 略 .....
INTRP_OCEAN_SFC_TEMP = "mask",
INTRP_OCEAN_TEMP = "mask",

/

&PARAM_MKINIT_REAL_LAND
..... 略 .....
USE_FILE_LANDWATER = .true.,

INTRP_LAND_TEMP = "mask",
INTRP_LAND_WATER = "fill",
INTRP_LAND_SFC_TEMP = "fill",

/

```

- ← 読み込むファイルの数
- ← 表 5.2 に示す設定値を指定
- ← 境界値データの出力名
- ← 入力データの時間間隔
- ← 親モデルの大気密度データを使うか
- ← SST の欠測値処理方法
- ← SST の欠測値処理方法
- ← 親モデルの土壌水分データを使うか
- ← 土壌温度の欠測値処理方法
- ← 土壌水分の欠測値処理方法
- ← 地表面温度の欠測値処理方法

`(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ため “grads” と指定している。詳細な設定ファイルの内容については、第 5.4 節を参照されたい。

第 4.2 節用意した、バイナリ形式に変換した入力データ (FNL) を作業ディレクトリにリンクを張る。リンクをはるためのスクリプトが、`/${Tutorial_DIR}/real/data` の “gradsinput-link\_FNL.sh” に用意されている。

```

$ cp ../../data/gradsinput-link_FNL.sh ./
$ sh gradsinput-link_FNL.sh

```

下記のリンクが作成されていれば成功である。

```

FNLatm_00000.grd -> ../tools/FNL_output/200707/FNLatm_2007071418.grd
FNLatm_00001.grd -> ../tools/FNL_output/200707/FNLatm_2007071500.grd
FNLatm_00002.grd -> ../tools/FNL_output/200707/FNLatm_2007071506.grd
FNLatm_00003.grd -> ../tools/FNL_output/200707/FNLatm_2007071512.grd
FNLLand_00000.grd -> ../tools/FNL_output/200707/FNLLand_2007071418.grd
FNLLand_00001.grd -> ../tools/FNL_output/200707/FNLLand_2007071500.grd
FNLLand_00002.grd -> ../tools/FNL_output/200707/FNLLand_2007071506.grd
FNLLand_00003.grd -> ../tools/FNL_output/200707/FNLLand_2007071512.grd
FNLsfc_00000.grd -> ../tools/FNL_output/200707/FNLsfc_2007071418.grd
FNLsfc_00001.grd -> ../tools/FNL_output/200707/FNLsfc_2007071500.grd
FNLsfc_00002.grd -> ../tools/FNL_output/200707/FNLsfc_2007071506.grd
FNLsfc_00003.grd -> ../tools/FNL_output/200707/FNLsfc_2007071512.grd

```

次に、GrADS 形式のバイナリデータを SCALE で読み込むための namelist ファイルを init ディレクトリへリンクする。

```
$ ln -s ../../data/namelist.grads_boundary.FNL.grib1 ./
```

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-rm_init init.d01.conf
```

正常にジョブが終了すると、

```

$ ls
  boundary_d01.pe000000.nc
  boundary_d01.pe000001.nc
  boundary_d01.pe000002.nc
  boundary_d01.pe000003.nc
  init_d01_20070714-180000.000.pe000000.nc
  init_d01_20070714-180000.000.pe000001.nc
  init_d01_20070714-180000.000.pe000002.nc
  init_d01_20070714-180000.000.pe000003.nc
  init_LOG_d01.pe000000

```

が作成され、init\_LOG\_d01.pe000000 の最後に

```

+++++++ Finalize MPI...
+++++++ MPI is peacefully finalized

```

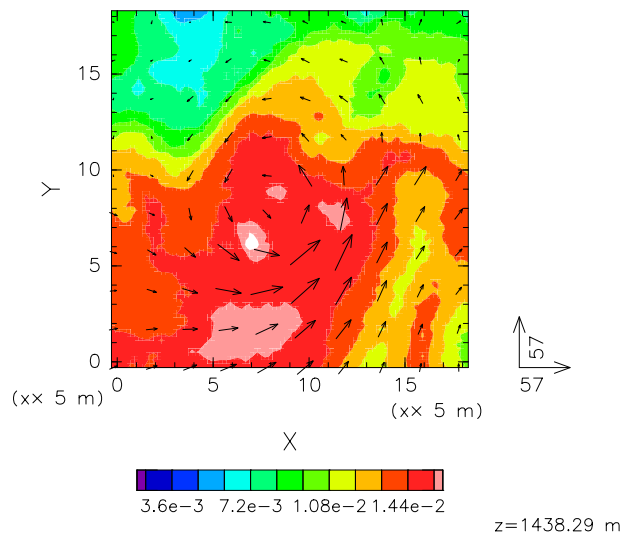
と出力される。boundary\_d01.pe#####.nc は境界値データで約 5.8MB のファイルサイズとなり、init\_d01\_20070714-180000.000.pe#####.nc は初期値データで約 3.5MB のファイルサイズとなる。init\_LOG\_d01.pe000000 はログファイルである。##### は MPI プロセス番号を表している。

## 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_20070714-180000.000.pe00*@QV \
  init_d01_20070714-180000.000.pe00*@MOMX init_d01_20070714-180000.000.pe00*@MOMY
```

Vapor mass to total mass (Specific humidity)(momentum x,



/home1/gleo/share/SCALE/bin/gpvect 2016-09-01

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。

## 4.6 シミュレーションの実行：run

### run.conf の準備

run ディレクトリへ移動する。

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

run ディレクトリの中には、run.d01.conf という名前の設定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 (表 4.1) に合わせて設定されている。他に run.launch.conf というファイルも作成されているが、ここでは使用しない。

モデル本体の実行には事前に作成した地形データや初期値・境界値データを利用する。これらのファイルの指定は、run.d01.conf の下記部分で設定している。

```

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

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

&PARAM_RESTART
  RESTART_RUN = .false.,
  RESTART_OUTPUT = .true.,
  RESTART_OUT_BASENAME = "restart_d01",
  RESTART_IN_BASENAME = "../init/init_d01_20070714-180000.000",
/

&PARAM_ATMOS_BOUNDARY
  ATMOS_BOUNDARY_TYPE = "REAL",
  ATMOS_BOUNDARY_IN_BASENAME = "../init/boundary_d01",
  ATMOS_BOUNDARY_START_DATE = 2007, 7, 14, 18, 0, 0,
  ATMOS_BOUNDARY_UPDATE_DT = 21600.0,
  ATMOS_BOUNDARY_USE_DENS = .true.,
  ATMOS_BOUNDARY_USE_VELZ = .true.,
  ATMOS_BOUNDARY_USE_QHYD = .false.,
  ATMOS_BOUNDARY_VALUE_VELZ = 0.0,
  ATMOS_BOUNDARY_ALPHAFACD_DENS = 1.0,
  ATMOS_BOUNDARY_LINEAR_H = .false.,
  ATMOS_BOUNDARY_EXP_H = 2.0,
/

```

run.d01.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。初期時刻は、**(TIME\_STARTDATE)** に UTC で指定する。チュートリアルでは 2007 年 7 月 14 日 18 時 UTC に設定している。積分時間は **(TIME\_DURATION)** で与える。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。

```

&PARAM_TIME
  TIME_STARTDATE = 2007, 7, 14, 18, 0, 0, ← 時間積分を開始する時刻
  TIME_STARTMS = 0.D0,
  TIME_DURATION = 6.0D0, ← 積分期間
  TIME_DURATION_UNIT = "HOUR", ← TIME_DURATION の単位
  TIME_DT = 90.0D0, ← トレーサー移流計算の時間ステップ
  TIME_DT_UNIT = "SEC", ← TIME_DT の単位
  TIME_DT_ATMOS_DYN = 45.0D0, ← トレーサー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

  ..... 略 .....

/

```



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 = 3600.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_STEPO = .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) に列挙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="MSLP" /      海面更正気圧
&HISTITEM item="PREC" /     降水強度 (2次元)
&HISTITEM item="OLR" /      外向き赤外放射 (2次元)
&HISTITEM item="U10" /      地表 10m での X 方向水平速度成分 (2次元)
&HISTITEM item="V10" /      地表 10m での Y 方向水平速度成分 (2次元)
&HISTITEM item="T2" /       地表 2m での温度 (2次元)
&HISTITEM item="Q2" /       地表 2m での水蒸気比湿 (2次元)
&HISTITEM item="SFC_PRES" /  地表気圧 (2次元)
&HISTITEM item="SFC_TEMP" / バルクの地表面温度 (2次元)
&HISTITEM item="DENS" /     密度 (3次元)
&HISTITEM item="QV" /       水蒸気比湿 (3次元)
&HISTITEM item="QHYD" /     全凝結物の全質量に対する比 (3次元)
&HISTITEM item="PRES" /     圧力 (3次元)
&HISTITEM item="U" /        X 方向水平速度成分 (3次元)
&HISTITEM item="V" /        Y 方向水平速度成分 (3次元)
&HISTITEM item="T" /        温度 (3次元)
&HISTITEM item="W" /        鉛直方向速度成分 (3次元)
&HISTITEM item="Uabs" /     風速 (3次元)
&HISTITEM item="PT" /       温位 (3次元)
&HISTITEM item="RH" /       相対湿度 (3次元)

```

その他、実験で使用されるスキームの設定として、力学過程の設定は、[&PARAM\_ATMOS\_DYN] 物理過程の設定は、[PARAM\_TRACER, PARAM\_ATMOS, PARAM\_OCEAN, PARAM\_LAND, PARAM\_URBAN] に記述されている。詳細な内容については、第 5.7 節、5.8 節を参照されたい。

## シミュレーションの実行

実行に必要なファイルとして、下記が用意されている。

```
$ ls
```

```
MIPAS PARAG.29  PARAPC.29  VARDATA.RM29  cira.nc
                                : 放射スキーム用のパラメータファイル
run.d01.conf      : 設定ファイル
param.bucket.conf : 陸面スキーム用のパラメータファイル
scale-rm         : SCALE-RM 実行バイナリ
run.launch.conf  : ネスティング計算用の launch ファイル
                  (チュートリアルでは使用しない)
```

準備が整ったら、4-MPI 並列により SCALE-RM を実行する。

```
$ mpirun -n 4 ./scale-rm run.d01.conf >& log &
```

実行にはある程度時間を要するため（上記のマシンで計算完了までに 10~20 分を要した）、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"が生成される。さらに、ジョブが正常に終了すると、"LOG\_d01.pe000000"に

```
+++++++ Finalize MPI...
+++++++ MPI is peacefully finalized
```

と出力され、下記のファイルが作成される。

```
$ ls
history_d01.pe000000.nc
history_d01.pe000001.nc
history_d01.pe000002.nc
history_d01.pe000003.nc
```

それぞれ、おおよそ 23MB のファイルサイズとなる。モデルの出力ファイル (history\_d01.pe#####.nc) は、MPI プロセス毎に計算領域が分割されて、(HISTITEM) で指定した出力変数が出力される。#####は MPI プロセス番号を表している。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。

## 4.7 結果を描画する : net2g

ここでは、プロセス毎に分割された netCDF 形式の出力ファイル (history.\*\*.nc <sup>\*2)</sup> を GrADS で読み込めるように 1 つのバイナリファイルにまとめる netcdf2grads (略して、net2g) の使い方と、変換した GrADS バイナリデータを使って結果の確認を行う。

### GrADS バイナリに変換

プロセスごとに分割された netCDF 形式の history ファイルから GrADS バイナリ変換するには、net2g を使用する。詳細な使用法は 5.9 節を参照頂くこととし、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。

<sup>\*2)</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。

```
$ cd ${Tutorial_DIR}/real/experiment/net2g
$ ls
net2g -> ../../../../util/netcdf2grads_h/net2g
net2g.2D.d01.conf
net2g.3D.d01.conf
```

中には設定ファイルとバイナリファイルがあり、バイナリファイルは 2.4 節のコンパイルにより作成された実行ファイルにリンクが貼られている。

ここでは例として、2次元変数の MSLP、PREC の変換と、3次元変数の U、V を 850hPa、500hPa、200hPa 面で抽出して変換する手順について説明する。2次元変数のための設定ファイルは `net2g.2D.d01.conf` に、3次元変数のための設定ファイルは `net2g.3D.d01.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、計算に用いた時と同じ 4 プロセスを使用する。`net2g` では 2次元変数と 3次元変数を同時に変換することはできないため、別々に実行する。

```
$ mpirun -n 4 ./net2g net2g.2D.d01.conf
$ mpirun -n 4 ./net2g net2g.3D.d01.conf
```

エラーメッセージがなく、下記のメッセージだけが標準出力へ表示されていれば正常に変換完了である。

```
+++ MPI COMM: Corrective Finalize
```

成功すれば、下記のファイルが作成される。`**ctl` は SCALE-RM の XY 格子座標の `ctl` ファイル、`**lccr.ctl` は緯度経度座標で作図するための `ctl` ファイルである。

```
MSLP_d01z-2d.ctl
MSLP_d01z-2d.grd
MSLP_d01z-2d_lccr.ctl
PREC_d01z-2d.ctl
PREC_d01z-2d.grd
PREC_d01z-2d_lccr.ctl
PRES_d01z-3d.ctl
PRES_d01z-3d.grd
PRES_d01z-3d_lccr.ctl
U_d01z-3d.ctl
U_d01z-3d.grd
U_d01z-3d_lccr.ctl
V_d01z-3d.ctl
V_d01z-3d.grd
V_d01z-3d_lccr.ctl
```

#### 計算結果の確認

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

```
$ cp ../../data/checkfig_real.gs ./  
$ grads -blc checkfig_real.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。

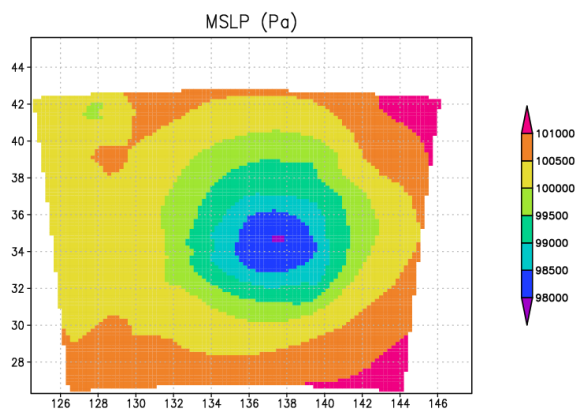


図 4.4: 計算開始から 6 時間後の海面更正気圧

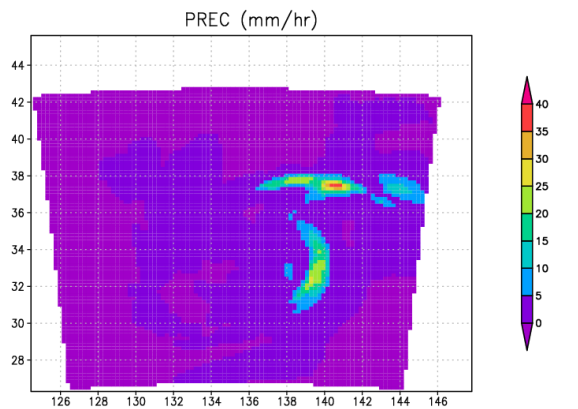


図 4.5: 計算開始から 6 時間後の降水フラックス

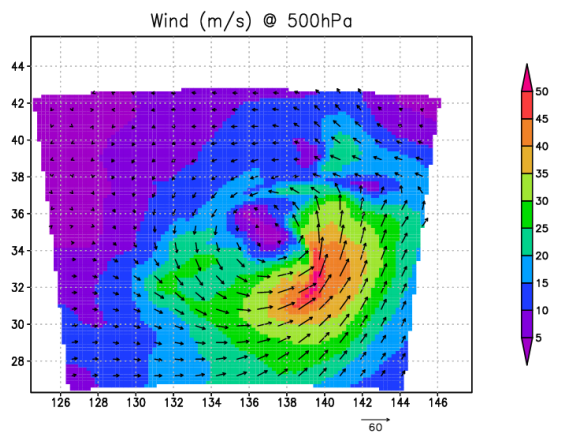


図 4.6: 計算開始から 6 時間後の 500hPa の風速と風ベクトル

## 第5章 各種設定

この章では、チュートリアルから発展して、基本的な様々な設定ができるように、各種設定を網羅的に記述している。各節で閉じており、辞書代わりに使ってほしい。

### 5.1 対象計算領域の設定

まず、SCALE-RM での対象計算領域の格子点数と MPI プロセスの関係を整理しておく。計算領域は、水平格子間隔と格子点数および MPI プロセス数を指定することで決定されるようになっている。図 5.1 は、計算領域、水平格子間隔、格子数、及び MPI プロセス数の関係を示している。水平方向に 2 次元の領域分割を行うことで並列化がなされている。

格子点数および MPI プロセス数は、`[PARAM_INDEX]` 内の `(IMAX, JMAX)`、`[PARAM_PRC]` 内の `(PRC_NUM_X, PRC_NUM_Y)` で設定する。

図 5.1 に示すように、計算領域全体が、X 方向に `(PRC_NUM_X)` 個、Y 方向に `(PRC_NUM_Y)` 個に分割され、それぞれが 1 つの MPI プロセスによって担当される。

それぞれの MPI プロセスは、`(IMAX, JMAX, KMAX)` 個の格子を受け持つ。ここで、注意すべきことは、「指定する格子点数は各プロセスが受け持つ値」であり、計算領域全体の格子点数ではないことである。すなわち、計算領域は、水平格子間隔、格子点数とともに MPI プロセス数にも依存する。

以上の関係から、計算領域全体のそれぞれの方向の格子点数および総格子点数は、

$$\text{領域内 X 方向の格子数} = \text{IMAX} \times \text{PRC\_NUM\_X} \quad (5.1)$$

$$\text{領域内 Y 方向の格子数} = \text{JMAX} \times \text{PRC\_NUM\_Y} \quad (5.2)$$

$$\text{領域内の総格子数} = (\text{IMAX} \times \text{PRC\_NUM\_X}) \times (\text{JMAX} \times \text{PRC\_NUM\_Y}) \times (\text{KMAX})$$

となる。ここで、`(KMAX)` は、鉛直方向の格子点数であり、`[PARAM_INDEX]` 内の項目で指定される。

また、領域全体の大きさは、式 (5.1)、(5.2) を使って、

$$X \text{ 方向の領域の長さ} = X \text{ 方向の格子点数} \times \text{DX}$$

$$Y \text{ 方向の領域の長さ} = Y \text{ 方向の格子点数} \times \text{DY}$$

となる。ここで、`(DX, DY)` は、後述するように `[PRAM_GRID]` で指定されるものである。解像度と領域の大きさが決まっている場合、MPI プロセス数が決まると、逆算して MPI プロセスあたりの格子点数が決まる。

次節以降では、MPI プロセス数、格子数、格子間隔、それぞれの設定方法について詳しく説明する。また、本節内の設定は、基本的に `pp.conf`、`init.conf`、`run.conf` の設定ファイル間で一致させなければならないことに注意が必要である。

#### 5.1.1 MPI プロセス数の設定

MPI プロセス数は、設定ファイルの `[PARAM_PRC]` で指定する。SCALE-RM の入出力ファイルは、MPI プロセス毎に分割されているため、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

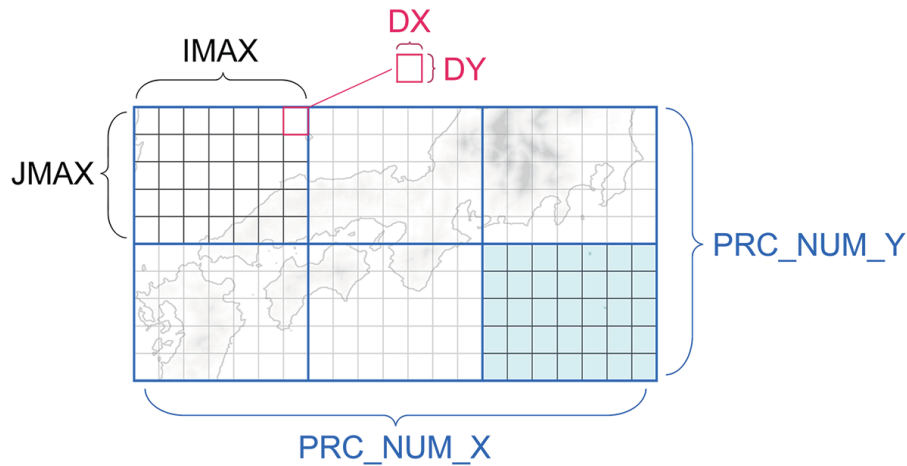


図 5.1: 計算領域に対する、水平格子間隔 (DX, DY)、MPI プロセスあたりの格子数 (IMAX, JMAX)、MPI プロセス数 (PRC\_NUM\_X, PRC\_NUM\_Y) の関係。水色領域は、ある 1 つの MPI プロセスが担当する領域。

ことになる。従って、例えば、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、pp.conf、init.conf、run.conf のすべてを編集・変更し、pp、init から行う必要がある。

```
&PARAM_PRC
  PRC_NUM_X = 2, ; X 方向 (東西方向) の MPI 並列分割数
  PRC_NUM_Y = 1, ; Y 方向 (南北方向) の MPI 並列分割数
/
```

総 MPI プロセス数は、 $PRC\_NUM\_X \times PRC\_NUM\_Y$  となり、上記の例では、X 方向に 2 分割、Y 方向に 1 分割 (分割なし) の 2-MPI 並列ということになる。

実行時に MPI コマンドに指定する MPI プロセス数は、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、下記のメッセージが標準出力に出力されて計算は行われず、直ちに終了する。

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

### 5.1.2 水平・鉛直格子数の設定

格子数の設定は、設定ファイル (\*\*\*.conf) の [PARAM\_INDEX] で行う。以下で設定する水平格子数の値は、1 つの MPI プロセスあたりの値であることに注意が必要である。

```
&PARAM_INDEX
  KMAX = 97, ; 鉛直層数
  IMAX = 20, ; プロセスあたりの X 方向の格子点数
  JMAX = 25, ; プロセスあたりの Y 方向の格子点数
/
```



### 5.1.3 水平・鉛直格子間隔の設定

SCALE-RM では、第 5.1.4 節で述べる緩和領域を除き、水平格子間隔は等間隔でのみ設定可能である。鉛直格子間隔については、任意に定義することが可能である。すべての方向について等間隔で設定する場合には、以下のように設定ファイルの [PARAM\_GRID] の (DX, DY, DZ) にそれぞれ、東西、南北、鉛直方向の格子間隔を指定する。単位は [m] である。

```
&PARAM_GRID
  DX = 500.DO, ; X 方向（東西方向）の格子間隔
  DY = 500.DO, ; Y 方向（南北方向）の格子間隔
  DZ = 500.DO, ; Z 方向（鉛直方向）の格子間隔
/
```

以下に、鉛直方向での任意の格子点位置を指定する場合の設定を示す。鉛直方向は、ローレンツ格子を採用しており、速度成分定義格子点とスカラー定義格子点が半格子分ずれた食い違い格子になっている。ここでは、スカラー量を定義している格子点をセンターポイントと呼び、半格子ズレた格子点をフェイスポイントと呼ぶ。

直接格子点の位置を指定する場合は、フェイスポイントの位置を [PARAM\_GRID] 中の (FZ(:)) で配列として与えればよい<sup>\*1)</sup> (図 5.2 参照)。また、(FZ(:)) で指定する値の数は、鉛直層数 ([PARAM\_INDEX] の (KMAX)) と一致している必要がある。例として理想実験のチュートリアルの設定ファイルの例を下記に示す。

```
&PARAM_GRID
  DX = 500.DO,           X 方向の格子間隔 (等間隔)[m]
  DY = 500.DO,           Y 方向の格子間隔 (等間隔)[m]
  FZ(:) =                Z 方向のフェイスポイントの位置 [m]
    80.0000000000000000 ,
    168.00000190734863 ,
    264.80000610351567 ,
    ~ 中略 ~
    14910.428862936289 ,
    15517.262523292475 ,
    16215.121232702089 ,
    17017.658748523147 ,
    17940.576891717363 ,
    19001.932756390710 ,
    20222.492000765058 ,
  BUFFER_DZ = 5000.DO,   第 5.1.4 節参照
  BUFFFACT = 1.0D0,      第 5.1.4 節参照
/
```

なお、指定された鉛直高度は標高が 0 m の場所（海上）での鉛直格子点位置として設定され、標高がある場所では地形に沿った座標系によって適切に処理される。

格子点位置は任意に設定できるが、設定によっては計算不安定につながる。鉛直層の設定については、作成をサポートするツールが、scale-5.1.5/scale-rm/util/makevgrid/ ディレクトリの中に make\_vgrid.f90 という Fortran プログラムといくつかのサンプルネームリストが用意されているので、参考にされたい。ツールをコンパイルして実行すれば直ちに設定ファイルに貼り付けて使用できる (FZ(:)) の値が作成される。

<sup>\*1)</sup> 指定の際には、シミュレーションの計算精度 (モデルのコンパイル時に指定した浮動小数点の精度。デフォルトでは倍精度) を用いることが望ましい。

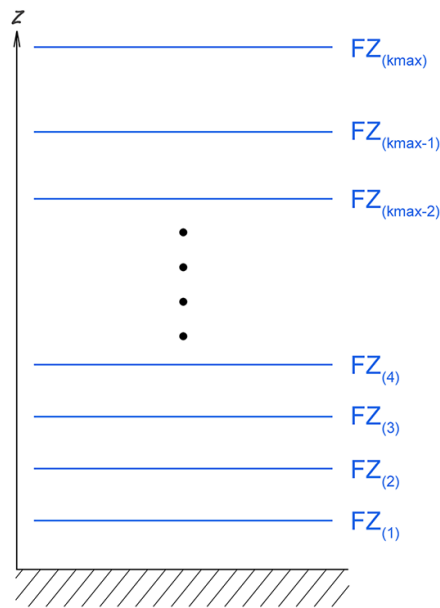


図 5.2: SCALE-RM の鉛直格子のフェイスポイントの定義点。[PARAM\_GRID] で (FZ) を指定する時は、地表面を除いた計算領域下端第 1 層目上端の位置から  $k = 1$  として与える。

#### 5.1.4 緩和領域とナッジングの設定

モデル最上層では音波や重力波の非物理的な反射が起こる。また、側面境界では現実大気実験を行う際に境界条件として与えられる入力データと対象領域の計算結果の間に値の不一致が起こる。これを回避するため、「緩和領域」を設ける。

SCALE-RM では計算領域の境界のすぐ内側に緩和領域を設定することができる (図 5.3)。緩和領域の格子では、計算された値を指定された値 (境界値データ、親領域のデータなど) に対してある時定数で近づける。以下これをナッジングと呼ぶ (上層における緩和はレイリーダンピングと呼ばれることが多い)。緩和領域の幅は、設定ファイルの [PARAM\_GRID] の中で設定する。以下に例を示す。設定はすべての設定ファイルにおいて共通していなければならない。

```
&PARAM_GRID
BUFFER_DZ = 5000.DO,      ; Z 方向 (モデルトップから下向き方向) の緩和領域の幅 (参考値) [m]
BUFFER_DX = 300000.DO,   ; X 方向 (東西方向) の緩和領域の幅 (参考値) [m]
BUFFER_DY = 300000.DO,   ; Y 方向 (南北方向) の緩和領域の幅 (参考値) [m]
BUFFFACT = 1.DO,        ; 緩和領域内の格子間隔に対するストレッチ係数 (デフォルトは 1.0)
/
```

水平方向には東西南北の四方境界に緩和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緩和領域が設定され、下端には設定されない。緩和領域は、計算領域内に設定されるため、ナッジングの影響を受けない領域 (緩和領域を除いた範囲) は計算領域よりも狭くなることに注意が必要である。

緩和領域の格子数  $ibuff$  は、

$$\sum_{n=1}^{ibuff} BDX(n) \geq BUFFER\_DX$$

の関係を満たす最小の整数で自動的に計算される。従って、緩和領域の大きさ  $BUFFER_X (= \sum_{n=1}^{ibuff} BDX(n))$

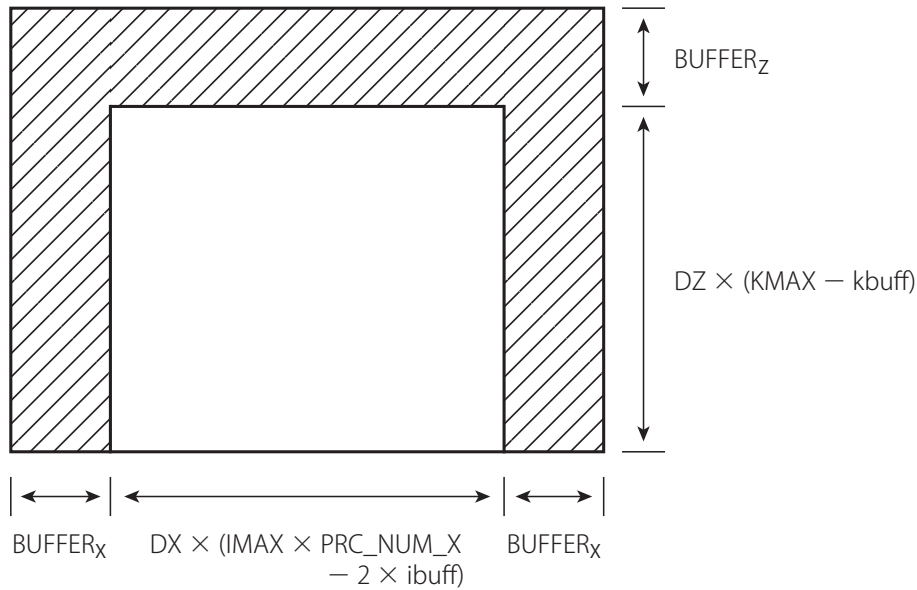


図 5.3: 計算領域における緩和領域の配置：斜線部分が緩和領域を意味する。図は XZ 断面だが、Y 方向にも同一の配置である。

は必ずしも (**BUFFER\_DX**) と一致するわけではないことに注意が必要である。結局、緩和領域を除いた計算領域の大きさは、

$$DX \times (IMAX \times PRC\_NUM\_X - 2 \times ibuff)$$

となる。

ここでは、X 方向の説明をしたが、Y 方向、Z 方向も同様である。ただし、Z 方向 については下端には緩和領域が取られないため、緩和領域を除いた計算領域の大きさは、モデル上端の緩和領域の格子数を **kbuff** として

$$DZ \times (KMAX - kbuff)$$

となる。

一般に、緩和領域の大きさや緩和格子点の数のとり方については、解く問題に依存するため明確な指標はない。SCALE-RM では、鉛直方向（計算領域トップ）の緩和格子点は 5 点以上、水平方向（側面境界付近）の緩和格子点は 20~40 点程度を推奨している。実験設定や事例によっては、さらに緩和格子点を増やしたり、後述のストレッチ係数を用いて緩和領域を広げたり、緩和の時定数を調整したりする必要があるだろう。時定数とは、目標値との差が  $1/e$  になるまでの時間である。緩和の時定数は、**[PARAM\_ATMOS\_BOUNDARY]** 中の (**ATMOS\_BOUNDARY\_taux**, **ATMOS\_BOUNDARY\_tauy**, **ATMOS\_BOUNDARY\_tauz**) によって秒単位で設定する。デフォルトの値は  $10\Delta t$  であり、これは、10 タイムステップで  $1/e$  になることに相当する。タイムステップについては、第 5.5 節を参照のこと。

#### 緩和領域の格子間隔をストレッチさせる

緩和領域の格子間隔は、基本的に **[PARAM\_GRID]** 中の (**DX**, **DY**, **DZ**) で指定した通りであるが、(**BUFFFACT**) に 1 以上の値を設定することで、ストレッチさせることも可能である。ただし、格子間隔を等間隔で指定した場合、この (**BUFFFACT**) の設定は、X、Y、Z 方向すべてに適用される。それぞれの方向に別々に設定したい場合は、(**BUFFFACT\_X**, **BUFFFACT\_Y**, **BUFFFACT\_Z**) を指定する。Z

方向の層レベルを任意の格子点位置に指定する場合、すなわち、(FZ(:)) を与える場合 (第 5.1.3 節参照のこと) にはストレッチの設定は Z 方向には適用されない。

緩和領域内の格子間隔 (BDX) は次の通り決定される。

$$BDX(n) = DX \times BUFFFACT^n$$

ここで、 $n$  は緩和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う番号である。緩和領域の格子間隔は、(BUFFFACT=1.0) ならば内部領域と同じであり、(BUFFFACT=1.2) ならば内側から外側 (境界) に向かって 1.2 倍の割合で広がっていく。(BUFFFACT) はいくつに設定しても良いが、計算の安定性を考慮すると 1.0 から 1.2 が推奨される。

緩和領域の大きさ  $BUFFER_x$  は、

$$BUFFER_x = DX \times \frac{BUFFFACT^{1_{buff}} - 1}{BUFFFACT - 1}$$

となる。緩和領域の幅 (BUFFER\_DX) が同じでも、(BUFFFACT) の値を大きくすると緩和領域の格子数は少なくなる。

## 5.2 地図投影法と計算領域の位置の設定

SCALE-RM では、まず実距離に基づいた格子点が配置され、その格子点位置と緯度経度基準点の情報を元に、それぞれの投影法を用いた際の各格子点での緯度・経度座標が計算される。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る。

計算領域の位置と投影法は、設定ファイルの (PARAM\_MAPPROJ) の項目を編集することで設定できる。この設定も、基本的に pp.conf、init.conf、run.conf の設定ファイル間で一致させなければならない。まずはじめに、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',
/
```

(MPRJ\_basepoint\_lat, MPRJ\_basepoint\_lon) は、それぞれ基準点の緯度・経度を表す。デフォルトでは基準点は計算領域の中心である。SCALE-RM 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。この設定例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。

表 5.1: SCALE で選択できる地図投影法

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

(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-RM で現在選択できる地図投影法とその指定文字列は表 5.1 のとおりである。メルカトル図法を用いた場合、投射する円筒に接する基準緯線は (MPRJ\_M\_lat) で設定する。値の単位は [degree] である。通常、基準緯線は赤道にとることが多いが、基準緯線に近いほど歪みが少なく正確に記述できるので、(MPRJ\_M\_lat) を指定しない場合は (MPRJ\_basepoint\_lat) が基準緯線として用いられる。

投影法の中でも利用頻度が高いランベルト正角円錐図法の設定について以下に説明する。ここでは、現実大気実験チュートリアルで使用した run.d01.conf ファイルを例に挙げる。

```
&PARAM_MAPPROJ
MPRJ_basepoint_lon = 135.220404,
MPRJ_basepoint_lat = 34.653396,
MPRJ_type = 'LC',
MPRJ_LC_lat1 = 30.0,
MPRJ_LC_lat2 = 40.0,
/
```

SCALE-RM では、2 標準緯線型の投影方法を採用している。両標準緯線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。南側、北側の標準緯線をそれぞれ (MPRJ\_LC\_lat1, MPRJ\_LC\_lat2) で設定する。値の単位は [degree] である。

さらに下記のように (MPRJ\_basepoint\_x, 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ができる。

```
&PARAM_MAPPROJ
MPRJ_basepoint_lon = 135.220404,
MPRJ_basepoint_lat = 34.653396,
MPRJ_basepoint_x = 100.0,
MPRJ_basepoint_y = 100.0,
MPRJ_type = 'LC',
MPRJ_LC_lat1 = 30.0,
MPRJ_LC_lat2 = 40.0,
/
```

(MPRJ\_basepoint\_x, 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、値の単位は [m] である。これらを指定しない場合、地図投影中心は計算領域中心に設定される。地図投影中心を指定しない場合とした場合を比較したものを図 5.4 に示す。図 5.4(a) は、デフォルト設定で地図投影中心と計算領域中心が一致している場合、図 5.4(b) は、地図投影中心をずらすよう指定した場合の関係を表している。図 5.4(b) では計算領域の南西端から (MPRJ\_basepoint\_x, MPRJ\_basepoint\_y) で指定した距離だけ離れた位置に投影中心が設定される。

## 5.3 地形の設定

SCALE-RM では地形に沿った座標系を採用している。この座標系はモデル下端の格子面を与えられた標高に対して沿うように与える。許容される最大の地形傾斜角度  $\theta_{\max}$  [radian] は次の式で計算される。

$$\theta_{\max} = \arctan(\text{RATIO} \times \text{DZ}/\text{DX})$$

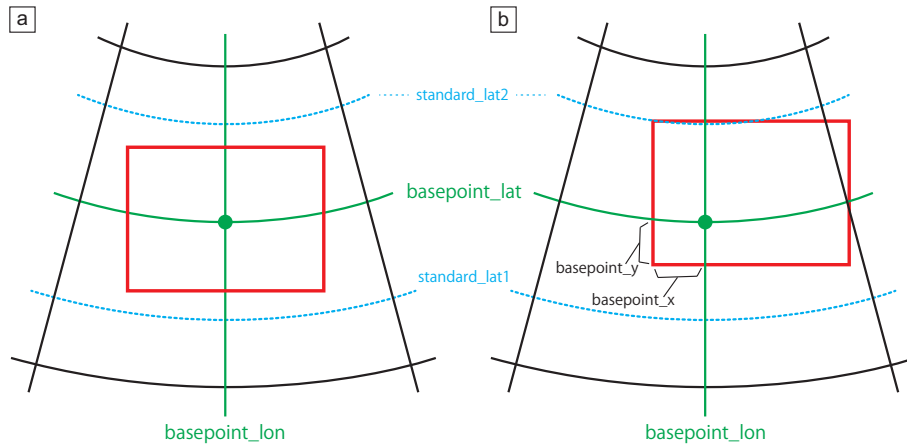


図 5.4: 投影中心と計算領域の関係：(a) はデフォルト設定の場合、(b) は投影中心の位置を計算領域中心からずらした場合。赤線の矩形が計算領域を表す。

ここで、DZ、DX は、それぞれ鉛直と水平の格子間隔である。上記の計算式から分かるように、許容される最大傾斜角度は空間解像度に応じて変わる。

SCALE-RM では RATIO のデフォルト値を 1.0 に設定している。RATIO の設定を、1.0 よりも大きくすれば地形がより細かく、1.0 よりも小さくすれば地形がより粗く表現される。ただし RATIO を 1.0 よりも大きくした場合、計算が途中で破綻する危険性が高くなる。

外部入力する標高データを実験設定に合わせて変換する作業は、scale-rm\_pp プログラムで行ない、詳細は実行時に使用する設定ファイル (pp.conf) の [PARAM\_CNVTPO] の中で設定する。以下に例を示す。

```
&PARAM_CNVTPO
CNVTPO_name = "GTOPO30",           ; 使用する地形データ名
CNVTPO_smooth_maxslope_ratio = 1.0, ; 許容する傾斜の DZ/DX に対する倍率
CNVTPO_smooth_local = .true.,     ; 最大傾斜角度を超えた格子のみ平滑化を行うかどうか
CNVTPO_copyparent = .false.,      ; 緩和領域に親ドメインの地形をコピーするかどうか
/
```

SCALE-RM では変換に使用する外部入力標高データベースとして、GTOPO30、または国土地理院による高精度地形データ (DEM50M) をサポートしている。

標高データは設定した水平格子に合わせて面積重み付け平均される。この変換された標高データの時点で、隣り合う格子との標高差を傾斜角として計算する。最大傾斜角度を超える傾斜角があった場合、それを最大傾斜角度以下になるように、反復計算を用いてラプラシアンフィルターでの平滑化を実行する。

このとき、最大傾斜角度を超えた格子のみ平滑化を行うか、計算領域全体で行うかを選択することができる。前者は、最大傾斜角度以内のシャープな地形構造を残すことができるので、細かな地形表現を望む場合に選択する。平滑化フィルターは、ガウシアンフィルターを選択することも出来る。こちらを用いた場合はラプラシアンフィルタより平滑化された地形になる。

(CNVTPO\_copyparent) は、ネスティング計算のための設定項目である。一般的に、多段ネスティング計算を行う場合、子ドメインのほうが空間解像度が細かいため、地形もシャープに表現される。このとき、親ドメインと子ドメインの地形表現が異なるために、子ドメインの緩和領域で用いられる親ドメインの気象データと不整合を起こすことがある。これを回避するために、(CNVTPO\_copyparent)



を `.true.` とすることで、親ドメインの地形を子ドメインの緩和領域にコピーすることができる。親ドメインが存在しない場合は `(CNVTOPO_copyparent)` を必ず `.false.` に設定しなければならない。`(CNVTOPO_copyparent)` を利用する場合の設定については、第 5.11.1 節で詳しく説明する。

## 5.4 様々な初期値・境界値データの作成方法

SCALE-RM では初期値・境界値作成において、表 5.2 に示す外部入力データの読み込みに対応している。

表 5.2: SCALE が読み込みに対応する外部入力データフォーマット

データ形式	FILETYPE_ORG	備考
SCALE データ	SCALE-RM	history データのみ対応。latlon カタログを必要とする。
バイナリデータ	GrADS	データ読み込み用のネームリストを別途必要とする。
WRF データ	WRF-ARW	“wrfout”、“wrfst” の両方に対応する。

外部入力する初期値・境界値データを変換する作業は、`scale-rm_init` プログラムで行ない、詳細は実行時に使用する設定ファイル (`init.conf`) の中で設定する。

このうち、入力データフォーマットの指定は、`[PARAM_MKINIT_REAL_***]` の `(FILETYPE_ORG)` で指定する。表 5.2 に設定するための値を示す。

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、5.11.2 節を参照されたい。

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。ただし、SCALE の境界値データ作成に必要な出力変数が全て出力されている必要がある。

バイナリデータ形式とは、「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。GRIB/GRIB2 データは、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に変換することで SCALE-RM に読み込ませることができる。その他にも、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れば読み込ませることができる。下記では、バイナリデータの読み込み手続きについて説明する。

### バイナリデータの読み込み

入力データフォーマットの指定は、`init.conf` の `[PARAM_MKINIT_REAL_***]` で設定する。



```

&PARAM_RESTART
  RESTART_OUTPUT = .true.,
  RESTART_OUT_BASENAME = "init_d01",
/

&PARAM_MKINIT_REAL_ATMOS
  NUMBER_OF_FILES = 2,
  FILETYPE_ORG = "GrADS",
  BASENAME_ORG = "namelist.grads_boundary.FNL.grib1",
  BASENAME_BOUNDARY = "boundary_d01",
  BOUNDARY_UPDATE_DT = 21600.0,
  PARENT_MP_TYPE = 3,
  USE_FILE_DENSITY = .false.,
/

&PARAM_MKINIT_REAL_OCEAN
  NUMBER_OF_FILES = 2,
  FILETYPE_ORG = "GrADS",
  BASENAME_ORG = "namelist.grads_boundary.FNL.grib1",
  INTRP_OCEAN_SFC_TEMP = "mask",
  INTRP_OCEAN_TEMP = "mask",
/

&PARAM_MKINIT_REAL_LAND
  NUMBER_OF_FILES = 2,
  FILETYPE_ORG = "GrADS",
  BASENAME_ORG = "namelist.grads_boundary.FNL.grib1",
  USE_FILE_LANDWATER = .true.,
  INTRP_LAND_TEMP = "mask",
  INTRP_LAND_WATER = "fill",
  INTRP_LAND_SFC_TEMP = "fill",
/

```

バイナリデータを読み込むときは、(FILETYPE\_ORG) に"GrADS"を設定する。

(NUMBER\_OF\_FILES) は、読み込むファイルの数である。入力ファイルが時間方向に複数に分かれている場合は、"ファイル名.XXXXX.grd"のように時間の早い方から順に5桁の数字をつける。scale-rm\_init では、00000 から (NUMBER\_OF\_FILES) の数だけファイルを読み込む。ファイルが一つしかない場合は、"ファイル名.grd"でよい。"ファイル名"は後ほど紹介するネームリストファイル内で指定する。(BOUNDARY\_UPDATE\_DT) は、入力データの時間間隔である。

(BASENAME\_BOUNDARY) は、SCALE-RM の格子の値に変換された境界値データの出力先ファイル名の頭である。また、初期値データの出力先は [PARAM\_RESTART] の (RESTART\_OUT\_BASENAME) で設定する。

SCALE-RM では、ctl ファイルの代わりに、データのファイル名とデータ構造の情報を含むネームリストファイル (namelist.grads\_boundary\*\*) を用意する。このネームリストファイルへのパスは (BASENAME\_ORG) に指定する。

以上の設定は、[PARAM\_MKINIT\_REAL\_ATMOS] で指定されたものが [PARAM\_MKINIT\_REAL\_OCEAN]、[PARAM\_MKINIT\_REAL\_LAND] にも適用されるが、別途指定されている場合には、情報が上書きされる。

(USE\_FILE\_DENSITY) は FILETYPE\_ORG="SCALE-RM"用のオプションなので、バイナリデータ読み込みのときは常に .false. とする。(PARENT\_MP\_TYPE) は親モデルの水物質カテゴリーのタイプであるが、バイナリデータ読み込みのときは常に 3 に設定する。

土壌水分の設定については、親モデルから与える方法と、領域全体に一定条件を与える方法の2種類ある。親モデルから与える場合には、3次元の土壌水分データを用意する必要がある。領域全体に一定条件を与える場合には、`init.conf` の `[PARAM_MKINIT_REAL_LAND]` に `USE_FILE_LANDWATER = .false.` を追加すれば良い。また、土壌水分の条件は、土壌空隙率に対する水が占める割合を `INIT_LANDWATER_RATIO` で与える。デフォルト値は 0.5 である。また、単位体積に対する空隙率は土地利用に応じて変わる。

```
&PARAM_MKINIT_REAL_LAND
USE_FILE_LANDWATER = .false.   土壌水分をファイルから読むかどうか。デフォルトは.true.
INIT_LANDWATER_RATIO = 0.5    USE_FILE_LANDWATER=.false. の場合、
                               空隙率に対する水が占める割合。
..... 略.....
/
```

バイナリデータは GrADS で読み込める形式でユーザが用意する (<http://cola.gmu.edu/grads/gadoc/aboutgriddeddata.html#structure>)。データのファイル名とデータ構造を SCALE-RM に教えるためのネームリストファイル (`namelist.grads_boundary**`) の一例を下記に示す。

```
#
# Dimension
#
&nml_grads_grid
  outer_nx = 360, ; 大気データの x 方向の格子数
  outer_ny = 181, ; 大気データの y 方向の格子数
  outer_nz = 26,  ; 大気データの z 方向の層数
  outer_nl = 4,   ; 土壌データの層数
/

#
# Variables
#
&grdvar item='lon', dtype='linear', swpoint=0.0d0, dd=1.0d0 /
&grdvar item='lat', dtype='linear', swpoint=90.0d0, dd=-1.0d0 /
&grdvar item='plev', dtype='levels', lnum=26,
  lvars=100000,97500,...(省略)...,2000,1000, /
&grdvar item='MSLP', dtype='map', fname='FNLsfc', startrec=1, totalrec=6 /
&grdvar item='PSFC', dtype='map', fname='FNLsfc', startrec=2, totalrec=6 /
&grdvar item='U10', dtype='map', fname='FNLsfc', startrec=3, totalrec=6 /
&grdvar item='V10', dtype='map', fname='FNLsfc', startrec=4, totalrec=6 /
&grdvar item='T2', dtype='map', fname='FNLsfc', startrec=5, totalrec=6 /
&grdvar item='RH2', dtype='map', fname='FNLsfc', startrec=6, totalrec=6 /
&grdvar item='HGT', dtype='map', fname='FNLatm', startrec=1, totalrec=125 /
&grdvar item='U', dtype='map', fname='FNLatm', startrec=27, totalrec=125 /
&grdvar item='V', dtype='map', fname='FNLatm', startrec=53, totalrec=125 /
&grdvar item='T', dtype='map', fname='FNLatm', startrec=79, totalrec=125 /
&grdvar item='RH', dtype='map', fname='FNLatm', startrec=105, totalrec=125, knum=21 /
&grdvar item='llev', dtype='levels', lnum=4, lvars=0.05,0.25,0.70,1.50, /
&grdvar item='lsmask', dtype='map', fname='FNLland', startrec=1, totalrec=10 /
&grdvar item='SKINT', dtype='map', fname='FNLland', startrec=2, totalrec=10 /
&grdvar item='STEMP', dtype='map', fname='FNLland', startrec=3, totalrec=10,
  missval=9.999e+20 /
&grdvar item='SMOISVC', dtype='map', fname='FNLland', startrec=7, totalrec=10,
  missval=9.999e+20 /
```

大気データの格子数を `outer_nx`, `outer_ny`, `outer_nz` で指定し、土壌データ (`STEMP`, `SMOISVC`) の層数を `outer_nl` に指定する。

QV や RH のデータは、他の大気データよりも低い層までしか提供されていない場合がある。その場合、データがある層の数を `knum` で指定するが、それより上の層の QV の値の見積もり方について、2 種類の手法が用意されている。デフォルトは `upper_qv_type = "ZERO"` である。

```
&PARAM_MKINIT_REAL_GrADS
  upper_qv_type = "ZERO"    "ZERO": QV=0
                           "COPY": データがある最上層の RH をコピー
/
```

`[grdvar]` は、各変数のデータの与え方に応じて必要な変数が異なる。

表 5.3: `[grdvar]` の変数

grdvar の項目	説明	備考
item	変数名	表 5.4 より選択
dtype	データ形式	"linear", "levels", "map" から選択
<b>(dtype) が "linear" の場合のネームリスト ("lon", "lat" 専用)</b>		
swpoint	スタートポイントの値	
dd	増分	
<b>(dtype) が "levels" の場合のネームリスト ("plev", "llev" 専用)</b>		
lnum	レベルの数 (層数)	
lvars	各層の値	
<b>(dtype) が "map" の場合のネームリスト</b>		
fname	ファイル名の頭	
startrec	変数 (item) のレコード番号	t=1 の時刻の値
totalrec	全変数一時刻あたりのレコード長	
knum	3次元データの z 方向の層数	(オプション) <code>outer_nz</code> と異なる場合。 RH と QV のみ使用可。
missval	欠陥値の値	(オプション)

表 5.4 のうち、体積含水率は土の体積  $V$  の中に占める水の体積  $V_w$  の割合 ( $V_w/V$ )、飽和度は  $V$  の中に占める間隙の体積  $V_v$  に対する水の体積  $V_w$  の割合 ( $V_w/V_v$ ) である。`[PARAM_MKINIT_REAL_LAND]` の `(USE_FILE_LANDWATER)` が `.true.` の場合、`SMOISVC` か `SMOISDS` のどちらかを用意する必要がある。

表 5.4: [grdvar] の (item) の変数リスト

必須変数	変数名	説明	単位	(dtype)
*	lon	経度データ	[deg.]	linear, map
*	lat	緯度データ	[deg.]	linear, map
*	plev	気圧データ	[Pa]	levels, map
*	HGT	高度 (ジオポテンシャル) データ	[m]	map
*	U	東西風速	[m/s]	map
*	V	南北風速	[m/s]	map
*	T	気温	[K]	map
*	QV	比湿 (RH がある場合は省略可)	[kg/kg]	map
*	RH	相対湿度 (QV がある場合は省略可)	[%]	map
	MSLP	海面更正気圧	[Pa]	map
	PSFC	地上気圧	[Pa]	map
	U10	10m 東西風速	[m/s]	map
	V10	10m 南北風速	[m/s]	map
	T2	2m 気温	[K]	map
	Q2	2m 比湿 (RH2 がある場合は省略可)	[kg/kg]	map
	RH2	2m 相対湿度 (Q2 がある場合は省略可)	[%]	map
	TOPO	GCM の地形	[m]	map
	lsmask	GCM の海陸分布	0:海 1:陸	map
*	SKINT	地表面温度	[K]	map
*	llev	土壌の深さ情報	[m]	levels
*	STEMP	土壌温度	[K]	map
(*)	SMOISVC	土壌水分 (体積含水率)	[-]	map
(*)	SMOISDS	土壌水分 (飽和度)	[-]	map
*	SST	海面温度 (SKINT がある場合は省略可)	[K]	map

## 5.5 積分時間と積分時間間隔の設定

積分時間やタイムステップは、実験の目的や設定によって適切に設定する必要がある。空間解像度を変えた場合はそれに応じたタイムステップを設定する必要があり、同じ解像度でも計算不安定を防ぐためにタイムステップを短くすることもある。

積分時間とタイムステップの設定は、設定ファイル `run.conf` の `[PARAM_PRC]` の項目を編集することで設定できる。

```

&PARAM_TIME
TIME_STARTDATE = 2014, 8, 10, 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.DO,                    計算開始時刻 [mili sec]
TIME_DURATION = 12.0DO,                  積分時間 [単位はTIME_DURATION_UNIT で設定]
TIME_DURATION_UNIT = "HOUR",             TIME_DURATION の単位
TIME_DT = 60.0DO,                       時間積分のタイムステップ
TIME_DT_UNIT = "SEC",                   TIME_DT の単位
TIME_DT_ATMOS_DYN = 30.0DO,              力学過程計算のタイムステップ
TIME_DT_ATMOS_DYN_UNIT = "SEC",          TIME_DT_ATMOS_DYN の単位
TIME_DT_ATMOS_PHY_MP = 60.0DO,           雲物理過程計算のタイムステップ
TIME_DT_ATMOS_PHY_MP_UNIT = "SEC",       TIME_DT_ATMOS_PHY_MP の単位
TIME_DT_ATMOS_PHY_TB = 60.0DO,           乱流過程計算のタイムステップ
TIME_DT_ATMOS_PHY_TB_UNIT = "SEC",       TIME_DT_ATMOS_PHY_TB の単位
TIME_DT_ATMOS_PHY_RD = 600.0DO,          放射過程計算のタイムステップ
TIME_DT_ATMOS_PHY_RD_UNIT = "SEC",       TIME_DT_ATMOS_PHY_RD の単位
TIME_DT_ATMOS_PHY_SF = 60.0DO,           大気下端境界 (フラックス) 過程計算のタイムステップ
TIME_DT_ATMOS_PHY_SF_UNIT = "SEC",       TIME_DT_ATMOS_PHY_SF の単位
TIME_DT_OCEAN = 300.0DO,                 海面過程計算のタイムステップ
TIME_DT_OCEAN_UNIT = "SEC",              TIME_DT_OCEAN の単位
TIME_DT_LAND = 300.0DO,                  陸面過程計算のタイムステップ
TIME_DT_LAND_UNIT = "SEC",               TIME_DT_LAND の単位
TIME_DT_URBAN = 300.0DO,                 都市過程計算のタイムステップ
TIME_DT_URBAN_UNIT = "SEC",              TIME_DT_URBAN の単位
TIME_DT_ATMOS_RESTART = 21600.DO,        リスタートファイル (大気) の出力間隔
TIME_DT_ATMOS_RESTART_UNIT = "SEC",      TIME_DT_ATMOS_RESTART の単位
TIME_DT_OCEAN_RESTART = 21600.DO,        リスタートファイル (海面) の出力間隔
TIME_DT_OCEAN_RESTART_UNIT = "SEC",      TIME_DT_OCEAN_RESTART の単位
TIME_DT_LAND_RESTART = 21600.DO,         リスタートファイル (陸面) の出力間隔
TIME_DT_LAND_RESTART_UNIT = "SEC",       TIME_DT_LAND_RESTART の単位
TIME_DT_URBAN_RESTART = 21600.DO,        リスタートファイル (都市) の出力間隔
TIME_DT_URBAN_RESTART_UNIT = "SEC",      TIME_DT_URBAN_RESTART の単位
/

```

(`TIME_DT`) は、時間積分計算におけるタイムステップであり、 $\Delta t$  と呼ばれることが多い。トレーサー移流計算のタイムステップとして使われるほか、すべての物理過程計算のタイムステップの基本単位となる。タイムステップは、計算不安定を起こさないように格子間隔を移流速度で割った値が取りうる最少値よりも小さな値を設定する。

力学変数の時間積分のためのタイムステップは移流速度ではなく音速で制約されるため、一般には上記のタイムステップよりも小さくとる必要がある。この時間間隔は、(`TIME_DT_ATMOS_DYN`) で設定し、(`TIME_DT`) の約数である必要があることに注意すること。(`TIME_DT_ATMOS_DYN`) の値は、計算安定性のため主に時間積分スキームに依存し、(`ATMOS_DYN_TINTEG_SHORT_TYPE`) が RK4 の場合には最少格子間隔 (HE-VI 利用時には水平の最少格子間隔) を 420 m/s で割った値が、RK3 の場合には 840 m/s で割った値が目安となる。

それぞれの物理過程のタイムステップは各過程が与えるテンデンシーが更新されるタイミングを表す。各過程は利用される場合、モデルのセットアップ時に一度計算され、テンデンシーが設定され

る。その後、各過程に設定した時間間隔でテンデンシーが更新される。時間間隔は (TIME\_DT) の倍数であることが望ましい。時間間隔を指定しない場合は、(TIME\_DT) と同じ値が設定される。

海面・陸面・都市モデルを利用する場合は、大気側で下端境界（フラックス）過程の計算は行われな  
い。大気は、各地表面過程が計算した状態値・フラックスを海陸フラクションや都市被覆率に応じて重  
み付け平均された値を受け取る。このとき、(TIME\_DT\_ATMOS\_PHY\_SF) は受け取った値を HISTORY  
出力に渡す時間間隔にのみ用いられるので、(TIME\_DT\_OCEAN), (TIME\_DT\_LAND), (TIME\_DT\_URBAN)  
と同じか、その約数に設定しておく。

リスタートファイルの出力間隔と、各過程の計算間隔についても気をつけなければいけない。上  
で示したとおり、各過程のテンデンシーは必ずセットアップ時に更新される。そのため、リスタート  
ファイルの出力間隔がすべての利用する過程の計算時間間隔の倍数になっていない場合、通して時間  
積分を行った場合と途中で停止してリスタートを行った場合の結果に差が生まれる。時間間隔を指定  
しない場合は途中で出力はされず、(TIME\_DURATION) と同じ時刻、すなわちシミュレーションの最  
後に出力される。リスタート計算の設定の詳細は第 5.10 節を参照のこと。

## 5.6 出力変数の追加・変更方法

計算結果の出力ファイルと出力形式の設定、及び、出力する変数の追加は、[PARAM\_HISTORY] と  
[HISTITEM] で行う。まず、出力ファイルとデフォルトの出力形式の設定を、run.conf の [PARAM\_HISTORY]  
で行う。

```
&PARAM_HISTORY
HISTORY_DEFAULT_BASENAME = "history_d01", ; 出力ファイル名の頭。
HISTORY_DEFAULT_TINTERVAL = 3600.0, ; 出力の時間間隔。
HISTORY_DEFAULT_TUNIT = "SEC", ; HISTORY_DEFAULT_TINTERVAL の単位。
HISTORY_DEFAULT_TAVERAGE = .false., ; .false.: 瞬間値、.true.: 平均値。
HISTORY_DEFAULT_ZINTERP = .false., ; 鉛直内挿するかどうか。
; .false.: モデル面出力。
; .true.: Z 面に内挿した値を出力。
HISTORY_DEFAULT_DATATYPE = "REAL4", ; 出力データの型。REAL4, REAL8 など。
HISTORY_OUTPUT_STEP0 = .true., ; 初期時刻 (t=0) の値を出力するかどうか。
; .true.: 出力、.false.: 出力しない。
/
```

(HISTORY\_DEFAULT\_TUNIT) の単位は、

"MSEC", "msec", "SEC", "sec", "s", "MIN", "min", "HOUR", "hour", "h", "DAY", "day"  
より選択可能である。

HISTORY\_DEFAULT\_TAVERAGE = .true. とし、平均値での出力を設定した場合、出力するタイミ  
ング直前の HISTORY\_DEFAULT\_TINTERVAL 間の平均値が出力される。

次に、出力する変数の設定を [HISTITEM] で行う。[HISTITEM] は変数毎に設定するため、出力した  
い変数の数だけ追加することになる。また、それぞれの変数の出力形式は、基本的に [PARAM\_HISTORY]  
の設定に準ずるが、変数毎に設定を変更することも可能である。

```
&HISTITEM
```

```
ITEM = "RAIN",          変数名。出力可能な変数は付録 B を参照
BASENAME = "rain_d01", (オプション) HISTORY_DEFAULT_BASENAME に同じ。
TINTERVAL= "600.0",    (オプション) HISTORY_DEFAULT_TINTERVAL に同じ。
TUNIT = "SEC",         (オプション) HISTORY_DEFAULT_TINTERVAL に同じ。
TAVERAGE = .true.,    (オプション) HISTORY_DEFAULT_TAVERAGE に同じ。
ZINTERP = .false.,    (オプション) HISTORY_DEFAULT_DATATYPE に同じ。
DATATYPE = "REAL4",   (オプション) HISTORY_DEFAULT_ZINTERP に同じ。
/
```

(オプション)の項目は、変数 (**ITEM**) にのみ適用される。上記では、明示的にすべての設定を書いているが、(**HISTORY\_DEFAULT\_\*\*\***) と同じ設定であればそれらが適用されるので明記する必要はない。例えば、上記の [**PARAM\_HISTORY**] の設定に、下記の [**HISTITEM**] の設定を組み合わせた場合には、`history_d01.xxxxxx.nc` に 4 バイト実数で、3600 秒毎に T, U, V の瞬間値が出力される。また、RAIN が、600 秒の出力間隔で、前 600 秒間の平均値が出力される。

```
&HISTITEM ITEM = "T" /
&HISTITEM ITEM = "U" /
&HISTITEM ITEM = "V" /
&HISTITEM ITEM = "RAIN", TINTERVAL = "600.0", TAVERAGE = .true. /
```

## 5.7 力学スキームの設定

### 5.7.1 数値解法の設定

時間方向の数値解法選択は、設定ファイルの [**PARAM\_ATMOS**] の (**ATMOS\_DYN\_TYPE**) で行う。

```
&PARAM_ATMOS
ATMOS_DYN_TYPE = "HEVI", ; 表 5.5 より選択。
/
```

表 5.5: 数値解法の選択肢

設定名	スキームの説明	備考
HEVE	水平陽解法-鉛直陽解法	
HEVI	水平陽解法-鉛直陰解法	実大気実験ではこちらを推奨

### 5.7.2 時間・空間差分スキームの設定

時間・空間差分スキームの設定は、設定ファイルの [**PARAM\_ATMOS\_DYN**] で設定する。また、空間スキームによってはハロの数の設定も必要である。下記の設定は推奨設定である。その他、スキームと各種設定については表 5.6 を参照のこと。また、時間スキームや空間差分スキーム変更時には、数値安定性が変わるため、タイムステップも考慮する必要がある。タイムステップについては、第 5.5 節を参照のこと。



```

&PARAM_ATMOS_DYN
ATMOS_DYN_TINTEG_SHORT_TYPE = RK4,           ; 時間スキームより選択
ATMOS_DYN_TINTEG_TRACER_TYPE = RK3WS2002,   ; 時間スキームより選択
ATMOS_DYN_FVM_FLUX_TYPE = CD4,              ; 空間スキームより選択
ATMOS_DYN_FVM_FLUX_TRACER_TYPE = UD3KOREN1993, ; 空間スキームより選択
ATMOS_DYN_FLAG_FCT_TRACER = .false.,        ; FCT スキームを利用するかどうか
ATMOS_DYN_NUMERICAL_DIFF_COEF = 1.D-2,
ATMOS_DYN_NUMERICAL_DIFF_COEF_TRACER = 0.DO,
ATMOS_DYN_enable_coriolis = .true.,
/

```

表 5.6: 力学スキームの設定

設定名	スキーム名	
時間スキーム		
RK3	3次3段ルンゲクッタスキーム (Heun)	
RK3WS2002	Wicker and Skamarock (2002) 3段ルンゲクッタスキーム	
RK4	4次4段ルンゲクッタスキーム	
空間スキーム		最小のハロ格子数
CD2	2次中央差分	1
CD4	4次中央差分	2
CD6	6次中央差分	3
UD3	3次風上差分	2
UD5	5次風上差分	3
UD3KOREN1993	3次風上差分 + Koren(1993) フィルター	2

トレーサー移流については、何らかの非負保証スキームを使う事が望ましい。(UD3KOREN1993) は非負保証スキームであるが、それ以外は非負が保証されない。つまり (ATMOS\_DYN\_FVM\_FLUX\_TRACER\_TYPE) として (UD3KOREN1993) 以外を選択した場合は、(ATMOS\_DYN\_FLAG\_FCT\_TRACER) を .true. として FCT を利用するのがよい。

ハロの格子数はデフォルトで2となっている。従って、空間スキームが CD4、UD3 以外の場合、上記の表に記したハロの数の設定が必要である。設定は、[PARAM\_INDEX] の中の (IHALO, JHALO) で行う。

例えば、空間スキームとして5次風上差分を設定した場合、

```

&PARAM_INDEX
IHALO = 3,
JHALO = 3,
/

```

のように設定する。

## 5.8 物理スキームの設定

### 5.8.1 雲微物理スキームの設定

雲微物理スキームの選択は、`init.conf` と `run.conf` のうち、`[PARAM_TRACER]` の `(TRACER_TYPE)`、及び、`[PARAM_ATMOS]` の `(ATMOS_PHY_MP_TYPE)` で設定する。

このとき `(TRACER_TYPE)` と `(ATMOS_PHY_MP_TYPE)` は普通は同じものを設定し、かつ、`init.conf`、`run.conf` で同一の設定とする必要がある。ただし、`(ATMOS_PHY_MP_TYPE)` を OFF とするときは、`(TRACER_TYPE)` は何を設定してもよいが、乾燥大気の場合には `TRACER_TYPE = "DRY"` とするのがよい。雲微物理スキームを呼び出すタイミングは、`[PARAM_TIME]` で設定するが、これについては第 5.5 節を参照のこと。以下に、氷雲を含む 1 モーメントバルク法を用いるときの設定を示す。

```
&PARAM_ATMOS
  ATMOS_PHY_MP_TYPE = "TOMITA08", ; 表 5.7 より選択。
/

&PARAM_TRACER
  TRACER_TYPE = "TOMITA08",          ATMOS_PHY_MP_TYPE と同じ名前。
/
```

表 5.7: 雲微物理スキームの設定

設定名	スキームの説明	文献
OFF	雲微物理による相変化を計算しない	
KESSLER	水雲のみの 1 モーメントバルク法	<a href="#">Kessler (1969)</a>
TOMITA08	氷雲を含む 1 モーメントバルク法	<a href="#">Tomita (2008)</a>
SN14	氷雲を含む 2 モーメントバルク法	<a href="#">Seiki and Nakajima (2014)</a>
SUZUKI10	1 モーメントビン法 (氷雲を含むか否かはオプションで選択)	<a href="#">Suzuki et al. (2010)</a>

#### 1. 氷を含まない 1 モーメントバルク法 [Kessler \(1969\)](#)

1 モーメントバルク法は粒径分布関数を 3 次のモーメント (質量) のみで表現する。このスキームでは雲粒と雨粒のカテゴリを考慮し、全質量に対する雲粒の質量比 ( $QC[\text{kg}/\text{kg}]$ ) と雨粒の質量比 ( $QR[\text{kg}/\text{kg}]$ ) を予報する。

粒径分布はデルタ関数、すなわち雲粒のサイズは全て同じと仮定することで表現する。雲粒と雨粒の半径はそれぞれ  $8\mu\text{m}$ 、 $100\mu\text{m}$  と与えられている。

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 モーメントバルク法 [Tomita \(2008\)](#)

このスキームでは [Kessler \(1969\)](#) と同じく、粒径分布関数を 3 次のモーメント (質量) のみで表現する。雲粒、雨粒、氷粒、雪片、あられの 5 カテゴリを考慮し、それぞれの質量比 ( $QC$ 、 $QR$ 、 $QI$ 、 $QS$ 、 $QG[\text{kg}/\text{kg}]$ ) を予報する。

粒径分布については雲粒と氷粒をデルタ分布 (それぞれ半径  $8\mu\text{m}$ 、 $40\mu\text{m}$ )、その他は Marshall-Palmer 分布を仮定して表現する。

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 モーメントバルク法 [Seiki and Nakajima \(2014\)](#)

氷を含む 2 モーメントバルク法は粒径分布関数を質量に加えて、0 次のモーメント（個数）で表現する。雲粒、雨粒、氷粒、雪片、あられの 5 カテゴリを考慮し、それぞれの質量比（QC、QR、QI、QS、QG）と数密度（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して表現する。

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 モーメントビン法 [Suzuki et al. \(2010\)](#)

1 モーメント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。雲粒、雨粒、氷粒、雪片、あられ、ひょうの 6 カテゴリを考慮し、各粒径ビンの質量比を予報する。

粒径分布は陽に与えられ、ビン数のとりかたによって分布の表現精度が異なる。

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 から 4 の順に高精度になるが、その分計算コストも高くなる。

SUZUKI10 以外を選択する場合は、`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が、**SUZUKI10 を選択する場合は**、以下のように `init.conf`、`run.conf` の双方に下記を追加する必要がある。

```
&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)
/
```

この場合も、`init.conf`、`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` の 1 行目に `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を行わず終了す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-RM 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)。

## 5.8.2 乱流スキームの設定

乱流スキームの選択は、`init.conf` と `run.conf` 中の `[PARAM_ATMOS]` 中の `(ATMOS_PHY_TB_TYPE)` で以下のように設定する。乱流スキームが呼び出されるタイミングは、`[PARAM_TIME]` で設定するが、これについては第 5.5 節を参照のこと。

```
&PARAM_ATMOS
  ATMOS_PHY_TB_TYPE = "MYNN", ; 表 5.8 より選択。
/
```

表 5.8: 乱流スキームの設定

設定名	スキームの説明	文献
OFF	乱流の計算を行わない	
SMAGORINSKY	Smagorinsky 型のサブグリッドモデル	<a href="#">Smagorinsky (1963)</a> ; <a href="#">Lilly (1962)</a> ; <a href="#">Brown et al. (1994)</a> ; <a href="#">Scotti et al. (1993)</a>
D1980	Deardorff(1980) サブグリッドモデル	<a href="#">Deardorff (1980)</a>
MYNN	MYNN Level 2.5 乱流モデル	<a href="#">Mellor and Yamada (1982)</a> ; <a href="#">Nakanishi and Niino (2004)</a>

### 5.8.3 放射スキームの設定

放射スキームの選択は、init.conf と run.conf 中の [PARAM\_ATMOS] の (ATMOS\_PHY\_RD\_TYPE) で設定する。放射スキームが呼び出されるタイミングは、[PARAM\_TIME] で設定するが、これについては第 5.5 節を参照のこと。

```
&PARAM_ATMOS
  ATMOS_PHY_RD_TYPE = "MSTRNX", ; 表 5.9 より選択。
/
```

表 5.9: 放射スキームの選択肢

設定名	スキームの説明	文献
OFF または NONE	放射スキームを使用しない	
MSTRNX	mstrnX	<a href="#">Sekiguchi and Nakajima (2008)</a>
WRF	mstrnX(長波)+Dudhia(短波)	<a href="#">Dudhia (1989)</a>

放射計算のための太陽放射量は、モデル実行の日付および時刻設定と、モデルの計算領域の緯度経度に従って計算される。理想実験のために、緯度経度や時刻を固定することも出来る。また太陽定数も変更できる。これらは [PARAM\_ATMOS\_SOLARINS] で設定する。

```

&PARAM_ATMOS_SOLARINS
  ATMOS_SOLARINS_constant = 1360.250117      太陽定数 [W/m2]
  ATMOS_SOLARINS_fixedlatlon = .false.        放射計算の緯度経度を固定するかどうか
  ATMOS_SOLARINS_fixeddate = .false.         放射計算の日時を固定するかどうか
  ATMOS_SOLARINS_lon = 135.221               ATMOS_SOLARINS_fixedlatlon=.true. の場合の経度
                                              [deg.]
  ATMOS_SOLARINS_lat = 34.653                ATMOS_SOLARINS_fixedlatlon=.true. の場合の緯度
                                              [deg.]
  ATMOS_SOLARINS_date = -1,-1,-1,-1,-1,-1,   ATMOS_SOLARINS_fixeddate=.true. の場合の年月日
                                              時刻 [Y,M,D,H,M,S]
/

```

[PARAM\_MAPPROJ] の (MPRJ\_basepoint\_lon,MPRJ\_basepoint\_lat) が設定されている場合には、(ATMOS\_SOLARINS\_lon, ATMOS\_SOLARINS\_lat) には、(MPRJ\_basepoint\_lon, MPRJ\_basepoint\_lat) がそれぞれ与えられる。[PARAM\_MAPPROJ] の説明は、第 5.2 節参照。

実験設定によっては、モデルトップの高度が 10-20 km と低いことがしばしばある。そのため放射計算ではモデルトップとは別の最上層高度を設定し、モデルトップより上空を何層で表現するか設定するようになっている。放射用最上層を何 km にとるかは放射スキーム・パラメータファイル依存であるが、例えば MSTRNX ではデフォルトのパラメータテーブルが想定する放射用最上層は 100km である。追加される高度はデフォルトの場合 10 層で表現する。すなわち、モデルトップが 22km であれば、放射スキーム内では 7.8km×10 層が追加されて計算される。これらは MSTRNX なら [PARAM\_ATMOS\_PHY\_RD\_MSTRN] で設定する。

```

&PARAM_ATMOS_PHY_RD_MSTRN
  ATMOS_PHY_RD_MSTRN_KADD = 10                モデルトップから TOA までの間の
                                              大気層数
  ATMOS_PHY_RD_MSTRN_TOA = 100.0            放射計算で考慮する大気最上層
                                              (TOA) の高さ [km](パラメータ
                                              ファイルに依存)
  ATMOS_PHY_RD_MSTRN_nband = 29             波長帯ピン数 (パラメータファ
                                              イルに依存)
  ATMOS_PHY_RD_MSTRN_nptype = 11            エアロゾルの種類の数 (パラメ
                                              タファイルに依存)
  ATMOS_PHY_RD_MSTRN_nradius = 6            エアロゾル粒径ピンの数 (パラメ
                                              タファイルに依存)
  ATMOS_PHY_RD_MSTRN_GASPARA_IN_FILENAME = "PARAG.29"  気体吸収パラメータの入力ファ
                                              イル
  ATMOS_PHY_RD_MSTRN_AEROPARA_IN_FILENAME = "PARAPC.29"  エアロゾル吸収・散乱パラメータ
                                              の入力ファイル
  ATMOS_PHY_RD_MSTRN_HYGROPARA_IN_FILENAME = "VARDATA.RM29"  エアロゾル粒径パラメータの入力
                                              ファイル
  ATMOS_PHY_RD_MSTRN_ONLY_QCI = .false.     雨・雪・あられを放射計算で考慮
                                              するかどうか
/

```

MSTRNX を実行するには、放射計算のためのパラメータテーブルが必要である。デフォルトでは太陽放射から赤外放射までを 29 バンド 111 チャンネルに分割し、雲・エアロゾル粒子は 11 種類、粒径を 6 ピンで表した時のテーブルを用いている。パラメータファイル (3 種類) は、

```
scale-rm/test/data/rad/
```

に用意されている。[PARAM\_ATMOS\_PHY\_RD\_MSTRN] でファイルのディレクトリと名前を指定する。

放射計算のために追加された層の気温・気圧プロファイルは外部から与える必要がある。また二酸化炭素やオゾン等の気体濃度プロファイルも必要である。プロファイルは気候値またはユーザが準備した ASCII 形式の入力データを用いることが出来る。

気候値の場合、SCALE-RM では気温・気圧については CIRA86 <sup>\*2)</sup> (Committee on Space Research; NASA National Space Science Data Center, 2006)、気体種については MIPAS2001 (Remedios et al., 2007) をデータベースとして用意している。気候値プロファイルについても、モデル実行の日付および時刻設定とモデルの計算領域の緯度経度に従って計算され、日射量計算のために固定した緯度経度や時刻を用いた場合は、それに従う。読み込むファイルは、

```
scale-rm/test/data/rad/
```

に用意されており、`[PARAM_ATMOS_PHY_RD_PROFILE]` でファイルのディレクトリと名前を指定する。

ユーザが準備するプロファイルの場合は、任意の層数の高度 [m]、気圧 [Pa]、気温 [K]、水蒸気量 [kg/kg]、オゾン濃度 [kg/kg] を ASCII ファイルで記述して指定する。それ以外の気体濃度はすべて 0 として扱われ、時間変化も考慮しない。このユーザ指定ファイルの例は、

```
scale-rm/test/data/rad/
```

に用意されており、`[PARAM_ATMOS_PHY_RD_PROFILE]` で (`ATMOS_PHY_RD_PROFILE_use_climatology = .false.`) と設定し、ファイルのディレクトリと名前を指定する。

```
&PARAM_ATMOS_PHY_RD_PROFILE
  ATMOS_PHY_RD_PROFILE_use_climatology = .true.           CIRA86 と MIPAS2001 の気候値を利用
                                                           するかどうか
  ATMOS_PHY_RD_PROFILE_CIRA86_IN_FILENAME = "cira.nc"     CIRA86 (NetCDF format) ファイル
  ATMOS_PHY_RD_PROFILE_MIPAS2001_IN_BASENAME = "."        MIPAS2001 (ASCII format) のディレク
                                                           トリ
  ATMOS_PHY_RD_PROFILE_USER_IN_FILENAME = ""             気候値を利用しない場合に用いるユーザ
                                                           指定ファイル (ASCII format)
  ATMOS_PHY_RD_PROFILE_USE_CO2 = .true.                  false の場合、強制的に CO2 濃度を常に
                                                           0 に設定する
  ATMOS_PHY_RD_PROFILE_USE_O3 = .true.                   false の場合、強制的に O3 濃度を常に 0
                                                           に設定する
  ATMOS_PHY_RD_PROFILE_USE_N2O = .true.                  false の場合、強制的に N2O 濃度を常に
                                                           0 に設定する
  ATMOS_PHY_RD_PROFILE_USE_CO = .true.                   false の場合、強制的に CO 濃度を常に 0
                                                           に設定する
  ATMOS_PHY_RD_PROFILE_USE_CH4 = .true.                  false の場合、強制的に CH4 濃度を常に
                                                           0 に設定する
  ATMOS_PHY_RD_PROFILE_USE_O2 = .true.                   false の場合、強制的に O2 濃度を常に 0
                                                           に設定する
  ATMOS_PHY_RD_PROFILE_USE_CFC = .true.                  false の場合、強制的にすべての CFC 濃
                                                           度を常に 0 に設定する
/
```

気候値・ユーザ指定プロファイルのいずれもモデルの高度・層数の設定とは独立に高度・層数が指定され、実行時にはモデル各層の値に内挿される。ただし、想定する最高高度（放射計算で考慮する大気最上層 (TOA) の高さ）がプロファイルよりも高い場合、外挿となるので注意されたい。

<sup>\*2)</sup> <http://catalogue.ceda.ac.uk/uuid/4996e5b2f53ce0b1f2072adadaeda262>

### 5.8.4 地表面 (大気下端境界) の設定

大気下端境界のフラックス計算スキームの選択は、init.conf と run.conf 中の [PARAM\_ATMOS] の (ATMOS\_PHY\_SF\_TYPE) で以下のように設定する。

```
&PARAM_ATMOS
  ATMOS_PHY_SF_TYPE = "COUPLE", ; 表 5.10 より選択。
/
```

海面・陸面・都市モデルを用いない場合の大気下端境界は、理想実験のための仮想的な地表面であることを想定している。この大気下端境界のフラックス計算を行うスキームが呼び出されるタイミングは、[PARAM\_TIME] で設定する。時間設定の詳細については第 5.5 節を参照のこと。

海面・陸面・都市モデルを用いる場合には、(ATMOS\_PHY\_SF\_TYPE) は COUPLE に設定する。

表 5.10: 大気下端境界の選択肢

設定名	スキームの説明
NONE	地表面フラックスを計算しない (海面・陸面・都市モデルの実行設定に応じて COUPLE に変更される)
OFF	地表面フラックスを計算しない
CONST	地表面フラックスを任意の値に固定
BULK	地表面フラックスをバルクモデルで計算
COUPLE	海面・陸面・都市モデルそれぞれが計算するフラックスを受け取る

#### 一定値設定の場合

(ATMOS\_PHY\_SF\_TYPE = "CONST") を選択した場合は、run.conf で下記を設定することにより、任意の値に固定することが可能である。下記の値はデフォルトの設定を示す。

```
&PARAM_ATMOS_PHY_SF_CONST
  ATMOS_PHY_SF_FLG_MOM_FLUX = 0           0: バルク係数を一定にする
                                           1: 摩擦速度を一定にする
  ATMOS_PHY_SF_U_minM = 0.0E0           絶対速度の下限值 [m/s]
  ATMOS_PHY_SF_Const_Cm = 0.0011E0     運動量に対する一定バルク係数値
                                           (ATMOS_PHY_SF_FLG_MOM_FLUX = 0 のとき有効)
  ATMOS_PHY_SF_CM_min = 1.0E-5         運動量に対するバルク係数の下限値
                                           (ATMOS_PHY_SF_FLG_MOM_FLUX = 1 のとき有効)
  ATMOS_PHY_SF_Const_Ustar = 0.25E0     一定摩擦係数値 [m/s]
                                           (ATMOS_PHY_SF_FLG_MOM_FLUX = 1 のとき有効)
  ATMOS_PHY_SF_Const_SH = 15.E0         一定地表面顕熱フラックス値 [W/m2]
  ATMOS_PHY_SF_FLG_SH_DIURNAL = .false. 顕熱フラックスに日変化をつけるか否か [logical]
  ATMOS_PHY_SF_Const_FREQ = 24.E0       顕熱フラックスに日変化を付けるときのサイクル [hour]
  ATMOS_PHY_SF_Const_LH = 115.E0        一定地表面潜熱フラックス値 [W/m2]
/
```



## バルク設定の場合

(`ATMOS_PHY_SF_TYPE = "BULK"`) を選択した場合は、任意の地表面温度に対応したフラックスをバルクモデルに従って計算する。このとき、粗度は海面粗度の計算スキームを利用しているが、後述するように海面粗度は定数を与えることができ、また蒸発効率を決める係数も任意の値を設定出来るため、海面に限定せず、陸面を想定した理想実験も行うことが出来る。

海面粗度を計算するスキームは、`run.conf` 中の [`PARAM_ROUGHNESS`] の (`ROUGHNESS_TYPE`) で以下のように設定する。

```
&PARAM_ROUGHNESS
ROUGHNESS_TYPE = "MOON07", ; 表 5.11 より選択。
/
```

表 5.11: 海面粗度スキームの選択肢

設定名	スキームの説明	文献
MOON07	経験式に基づく値 (時間発展あり)、デフォルト	<a href="#">Moon et al. (2007)</a>
MILLER92	経験式に基づく値 (時間発展なし)	<a href="#">Miller et al. (1992)</a>
CONST	定数を与える	

また、フラックス計算に使用するバルク交換係数の計算スキームは `run.conf` 中の [`PARAM_BULKFLUX`] の (`BULKFLUX_TYPE`) で以下のように設定する。

```
&PARAM_BULKFLUX
BULKFLUX_TYPE = "B91W01", ; 表 5.12 より選択。
/
```

表 5.12: バルク交換係数スキームの選択肢

設定名	スキームの説明	文献
B91W01	経験式に基づく値、デフォルト	<a href="#">Beljaars and Holtslag (1991)</a> ; <a href="#">Wilson (2001)</a>
U95	経験式に基づく値、( <a href="#">Louis (1979)</a> ) の改良版	<a href="#">Uno et al. (1995)</a>

### 5.8.5 海洋モデルの設定

海面過程は海面の状態量の更新と大気-海面間のフラックス計算の2つに大別される。これらの過程を計算するタイミングはどちらも [`PARAM_TIME`] で設定する。時間設定の詳細については第 5.5 節を参照のこと。



## 海面スキーム

海面の状態量（主に海面温度）の更新を担う海面スキームの選択は、init.conf と run.conf 中の [PARAM\_OCEAN] の (OCEAN\_TYPE) で設定する。[PARAM LANDUSE] で入力・設定された土地利用分布に海面が含まれている場合、(OCEAN\_TYPE = NONE または OFF) を選択することは出来ない。この条件を満たさない場合は、下記のメッセージが LOG ファイルなどに出力されて計算は行われず、直ちに終了する。

```
xxx Ocean fraction exists, but ocean components never called. STOP.
```

```
&PARAM_OCEAN
OCEAN_TYPE = "CONST", ; 表 5.13 より選択。
/
```

表 5.13: 海面スキームの選択肢

設定名	スキームの説明
NONE または OFF	海面モデルを利用しない
CONST	初期値のまま固定
FILE	外部ファイルから与える (時間変化可能)
SLAB	スラブ海面モデル

OCEAN\_TYPE = "FILE" を選択した場合は、init.conf と run.conf で外部入力ファイルの設定が必要である。この場合、与えられた外部ファイルの空間分布と時系列に応じて、海面温度は変化する。

```
&EXTITEM
basename = "../init/output/ocean_d01", ; 入力ファイル
varname = "OCEAN_TEMP", ; "OCEAN_TEMP" と書く。
step_limit = 1800,
step_fixed = -1,
enable_periodic_year = .false.,
enable_periodic_month = .false.,
enable_periodic_day = .false.,
/
```

OCEAN\_TYPE = "SLAB" を選択した場合は、init.conf と run.conf でスラブ混合層の深さを設定することができる。この場合、大気-海面間の熱フラックスの移動に応じて、スラブ混合層の温度は時間発展する。

```
&PARAM_OCEAN_PHY_SLAB
OCEAN_PHY_SLAB_DEPTH = 10.0D0, ; デフォルト設定 [m]
/
```

海面のアルベドは、どの海面スキームを選択しても同じ計算スキームが適用され、太陽天頂角に応じたアルベドが計算される。

## 大気-海面フラックス

海面スキームが設定されると、大気-海面間で必要な状態量やフラックスが交換されることになる。

大気-海面間フラックスは、複数あるバルクスキームのいずれかを用いて計算される。また、バルクスキーム内で利用される海面粗度長計算についても、複数のスキームが選択できる。海面粗度を計算するスキームは、run.conf 中の `[PARAM_ROUGHNESS]` の `(ROUGHNESS_TYPE)` で設定する。フラックス計算に使用するバルク交換係数の計算スキームは run.conf 中の `[PARAM_BULKFLUX]` の `(BULKFLUX_TYPE)` で設定する。これらの詳細については、第 5.8.4 節参照のこと。

### 5.8.6 陸面モデルの設定

陸面過程についても、海面過程と同じく陸面の状態量の更新と大気-陸面間のフラックス計算の 2 つに大別される。これらの過程が呼び出されるタイミングはどちらも `[PARAM_TIME]` で設定する。時間設定の詳細については第 5.5 節を参照のこと。

#### 陸面スキーム

陸面の状態量（主に陸面温度と土壤温度、土壤水分量）の更新を担う陸面スキームの選択は、init.conf と run.conf 中の `[PARAM_LAND]` の `(LAND_TYPE)` で以下のように設定する。`[PARAM_LANDUSE]` で入力・設定された土地利用分布に陸面が含まれている場合、`(LAND_TYPE = NONE または OFF)` を選択することは出来ない。この条件を満たさない場合は、下記のメッセージが LOG ファイルなどに出力されて計算は行われず、直ちに終了する。

```
xxx Land fraction exists, but land components never called. STOP.
```

```
&PARAM_LAND
LAND_TYPE = "SLAB", ; 表 5.14 より選択。
/
```

表 5.14: 陸面スキームの選択肢

設定名	スキームの説明
NONE または OFF	陸面モデルを利用しない
SLAB	熱拡散モデル+バケツモデル
CONST	SLAB で土壤温度、土壤水分量、陸面温度を更新しない

`(LAND_TYPE = "SLAB")` または `(LAND_TYPE = "CONST")` を選択した場合は、土地利用区分に応じたアルベド、粗度長等のパラメータテーブルと、土地利用区分分布の入力が必要である。パラメータテーブルは、

```
scale-rm/test/data/land/param.bucket.conf
```

に用意されている。

## 大気-陸面フラックス

陸面スキームが設定されると、大気-陸面間で必要な状態量やフラックスが交換されることになる。大気-陸面間フラックスの計算スキームは、陸面スキームに対応して選択される。`(LAND_TYPE = "SLAB")` または `(LAND_TYPE = "CONST")` を選択した場合は、海面または理想実験用の地表面で用いられるバルクスキームと同じものが利用され、`run.conf` 中の `[PARAM_BULKFLUX]` の `(BULKFLUX_TYPE)` で設定する。これらの詳細については、第 5.8.4 節参照のこと。

### 5.8.7 都市モデル (大気-都市面フラックス) の設定

都市過程は都市の状態量の更新と大気-都市面間のフラックス計算の 2 つに大別される。これらの過程を計算するタイミングはどちらも `[PARAM_TIME]` で設定する。時間設定の詳細については第 5.5 節を参照のこと。

#### 都市スキームおよび大気-都市面間フラックス

都市面の状態量（主に都市表面温度と水分量）の更新、また大気-都市面間フラックスの計算を担う都市スキームの選択は、`init.conf` と `run.conf` 中の `[PARAM_URBAN]` の `(URBAN_TYPE)` で以下のように設定する。`[PARAM LANDUSE]` で入力・設定された土地利用分布に都市が含まれている場合、`(URBAN_TYPE = NONE または OFF)` を選択することは出来ない。この条件を満たさない場合は、下記のメッセージが LOG ファイルなどに出力されて計算は行われず、直ちに終了する。

```
xxx Urban fraction exists, but urban components never called. STOP.
```

```
&PARAM_URBAN
  URBAN_TYPE = "SLC", ; 表 5.15 より選択。
/
```

表 5.15: 都市スキームの選択肢

設定名	スキームの説明	文献
NONE または OFF	都市モデルを利用しない	
SLC	単層キャノピーモデル	Kusaka et al. (2001)

## 5.9 ポスト処理

本節では、SCALE-RM の後処理ツール `net2g` について説明する。`net2g` は、SCALE-RM で出力されたノード毎に分割された `netCDF` 形式 `history` データ (`history.***.nc`) を結合し GrADS で読み込めるバイナリ形式に変換するツールである。分割データを結合することで、GrADS で描画できるだけでなく、Fortran プログラムなどで解析しやすいデータになる。`net2g` は MPI 並列実行も可能であり、単一プロセス実行に比べて変換に要する時間を短縮できる。その他、`net2g` の機能として次のものが利用できる。

- モデル面から任意の鉛直高度座標、鉛直気圧座標への内挿
- 3次元変数に対する鉛直の積算値、平均値、最大値、最小値の出力
- 3次元変数を鉛直層ごとに分割したファイルとしての出力
- 時間ステップごとに分割したファイルとしての出力

特に大規模計算結果においては、鉛直層や時間ステップごとに分割することで扱いやすいサイズのデータを出力することができるため便利である。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。ただし、現行の net2g においては下記の制約がある。

- net2g に使用する MPI プロセス数は、SCALE-RM 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。
- SCALE-RM 実行時の history 出力形式は、[PARAM\_HIST] の (HIST\_BND) が `.false.` でなければならない。
- 2次元データと3次元データは同時に変換できない。
- 変換できるデータは history データのみである。

実行時に注意すべき点は、net2g 実行時の MPI プロセス数を大きくしすぎないことである。net2g は実行内容のほとんどがデータ入出力であるため、MPI プロセス数を増やしすぎるとストレージへの読み込み/書き込み要求が多くなり、結果として、効率が下がり実行速度が落ちることもある。特に大規模計算結果を手元のマシンで変換する際は注意が必要である。

net2g の実行は MPI 並列を用いて、

```
$ mpirun -n [プロセス数] ./net2g net2g.conf
```

とする。最後の net2g.conf は net2g の実行設定が記述された設定ファイルである。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。エラーメッセージがなく、次のメッセージだけが標準出力へ表示されて終了すれば変換完了である。

```
+++ MPI COMM: Corrective Finalize
```

次に、3次元変数と2次元変数を変換する場合における設定ファイルの記述方法について説明する。それぞれ、scale-5.1.5/scale-rm/util/netcdf2grads\_h/ にあるサンプル設定ファイル“net2g.3d.conf”と“net2g.2d.conf”に基づいて説明する。本節では主要な設定項目だけを取り上げる。ここで触れなかったオプションについては、“scale-5.1.5/scale-rm/util/netcdf2grads\_h/” サンプル設定ファイル“net2g.all.conf”を参照してもらいたい。

## 設定ファイルサンプル：3次元変数の変換

```

&LOGOUT
LOG_BASENAME = "LOG_d01_3d",
LOG_ALL_OUTPUT = .false.,
/

&INFO
TIME_STARTDATE = 2000, 1, 1, 0, 0, 0,
START_TSTEP = 1,
END_TSTEP = 25,
DOMAIN_NUM = 1,
CONFFILE = "../run/run.d01.conf",
IDIR = "../run",
Z_LEV_TYPE = "plev",
MAPPROJ_ctl = .true.
/

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

```

上記はある domain の 3次元変数を、気圧高度面へ内挿して出力する場合の例を示している。各々の設定項目は次のとおりである。

- **[LOGOUT]** (このネームリストは必須ではない)
  - **(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]**
  - **(TIME\_STARTDATE)** : 変換する NetCDF データの最初の日時
  - **(START\_TSTEP)** : 変換する 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)** : 変換する NetCDF データの最後の Time Step を指定する。必ず指定すること。
  - **(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-RM 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-RM 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” はモデル面、“plev” は気圧面に内挿、そして “zlev” は高度面に内挿して出力する。“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。
  - **(MAPPROJ\_ctl)** : pdef を使った投影図法に対応した ctl ファイルを出力するかどうか。
- **[VARI]**

- (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。

#### 設定ファイル変更例：3次元変数の変換：鉛直積算値を出す

```

&INFO
  ~ 中略 ~
  Z_LEV_TYPE = "anal",
  ZCOUNT = 1,
/

&ANAL
  ANALYSIS = "sum",
/

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

```

上記は、簡易解析を利用する場合の設定ファイル変更例の抜粋である。他の設定項目については先の3次元変数の変換の場合と同じである。(Z\_LEV\_TYPE)を“anal”と指定することで、3次元変数の鉛直次元に対する簡易解析が有効になり、[ANAL]を指定できるようになる。このとき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(TARGET\_ZLEV)は指定できない。また、2次元データ出力を指定するため、[INFO]の(ZCOUNT)を必ず“1”と指定する。

- [ANAL]

- (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。

## 設定ファイルサンプル：2次元変数の変換

```

&LOGOUT
LOG_BASENAME = "LOG_d01_2d",
/

&INFO
TIME_STARTDATE = 2000, 1, 1, 0, 0, 0,
START_TSTEP = 1,
END_TSTEP = 25,
DOMAIN_NUM = 1,
CONFFILE = "../run/run.d01.conf",
IDIR = "../run",
ZCOUNT = 1,
MAPPROJ_ctl = .true.
/

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

```

上記はある domain の 2 次元変数をデータ形式変換する場合の設定例である。2 次元変数の変換を指定するため、[INFO] の (ZCOUNT) を必ず“1”と指定する。また、2 次元データなので [VARI] の (TARGET\_ZLEV) は指定できない。その他の設定項目は 3 次元変数変換の場合と変わらない。

## 設定ファイル変更例：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",
/

```

上記は設定ファイル変更例である。その他の項目は先の 3 次元や 2 次元変数の場合と同じである。SCALE-RM 実行時に、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、一部の出力変数が (HISTORY\_DEFAULT\_TINTERVAL) に従わない場合、net2g では [EXTRA] を上記のように設定することで対応できる。この例では、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 秒間隔で出力された場合の例を示している。net2g では、異なる出力時間間隔を同時に取り扱うことはできないため、複数の異なる出力間隔が存在する history データにおいては、それぞれに独立して net2g を走らせる必要がある。

## 大型並列計算機での実行

スーパーコンピュータ「京」などの大型計算機で大規模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容量がなかったり、後処理に膨大な時間がかかることがある。こういった場合には、SCALE-RM 本体の計算を行ったスーパーコンピュータ上で後処理も行なってしまうことをおすすめする。

スーパーコンピュータ「京」では、2.3.2節で説明されている環境変数の設定がされていれば、net2gはそのまま make コマンドでコンパイルできる。ジョブの投入・実行方法については、スーパーコンピュータ「京」のユーザーズマニュアル等を参照されたい。net2gの実行方法や制約、注意点については通常のマシンで実行する場合と何ら変わらない。

## 5.10 リスタート計算の方法

計算機のジョブ実行時間に制限がある場合や、長期積分時に途中で計算が異常終了してしまった場合など、リスタート計算を行うことで、ひと続きの計算を分割して実行することが可能である。リスタート計算の手続きは、下記の2ステップに分けられる。

1. リスタート計算のためのリスタートファイルを出力する。
2. リスタートファイルを読んで、続きの計算（リスタート計算）を行う。

リスタート計算時に次の計算のためのリスタートファイルを出力することも可能である。リスタートファイルの出力とリスタート計算実行のための設定は、設定ファイル run.conf の [PARAM\_RESTART] と [PARAM\_TIME] で設定する。

```

&PARAM_RESTART
RESTART_RUN = .true.,           リスタート計算かどうかの指定。
                                明記されていない場合は .false. が設定される。
                                .true.: リスタート計算、.false.: 通常計算。

RESTART_IN_BASENAME = "restart1_d01_20070715-000000.000",
                                入力する初期値ファイルの指定。

RESTART_OUTPUT = .true.,       リスタートファイルを出力するかどうか。
                                .true.: 出力する、.false.: 出力しない。
RESTART_OUT_BASENAME = "restart2_d01",
                                リスタートファイルのファイル名の頭。
                                この後ろに、出力時の年月日時刻が追加される。

/

&PARAM_TIME
TIME_STARTDATE = 2007, 7, 15, 00, 0, 0,   リスタート計算開始時刻
TIME_STARTMS = 0.DO,                     計算開始時刻 [mili sec]
TIME_DURATION = 12.0D0,                   積分時間 [単位はTIME_DURATION_UNIT で設定]
TIME_DURATION_UNIT = "HOUR",              TIME_DURATION の単位
..... 略 .....

TIME_DT_ATMOS_RESTART = 21600.DO,         リスタートファイル (大気) の出力間隔
TIME_DT_ATMOS_RESTART_UNIT = "SEC",       TIME_DT_ATMOS_RESTART の単位
TIME_DT_OCEAN_RESTART = 21600.DO,         リスタートファイル (海洋) の出力間隔
TIME_DT_OCEAN_RESTART_UNIT = "SEC",       TIME_DT_OCEAN_RESTART の単位
TIME_DT_LAND_RESTART = 21600.DO,         リスタートファイル (陸面) の出力間隔
TIME_DT_LAND_RESTART_UNIT = "SEC",        TIME_DT_LAND_RESTART の単位
TIME_DT_URBAN_RESTART = 21600.DO,        リスタートファイル (都市) の出力間隔
TIME_DT_URBAN_RESTART_UNIT = "SEC",       TIME_DT_URBAN_RESTART の単位

/

```

上記の例では、restart1\_\*\*\*という入力ファイルからリスタート計算を開始し、6時間(21600 sec)に1回、リスタートファイルをrestart2\_\*\*\*という名前で出力する設定となっている。

(TIME\_DT\_ATMOS\_RESTART), (TIME\_DT\_OCEAN\_RESTART), (TIME\_DT\_LAND\_RESTART), (TIME\_DT\_URBAN\_RESTART) が指定されていない場合には、積分時刻の最終時刻 (TIME\_DURATION) に



ファイルが作成される。`(RESTART_RUN)` を `.false.` として `(RESTART_IN_BASENAME)` に `scale-rm_init` で作成した初期値ファイル (`init_***`) を指定すれば通常計算が実行され、`(RESTART_RUN)` を `.true.` として `(RESTART_IN_BASENAME)` にリスタートファイルを指定すればリスタート計算が実行される。なお、SCALE-RM は `(RESTART_IN_BASENAME)` で指定されているファイル名でリスタート計算かどうかを判断しているわけではないので、ファイル名はなんでも良い。チュートリアル、および、本書解説では便宜上、`scale-rm_init` で作成した初期値ファイルの名前を `init_***.nc` とし、リスタートファイルのファイル名を `restart_***.nc` としているが、これに限らない。

計算開始時刻と積分時間は、リスタート計算の計算開始時刻と積分時間を設定する。つまり、リスタート計算のためのリスタートファイルを出力した実験とは独立した計算と考えれば良い。ただし、使用するスキームを変更する場合には、計算に必要な入力変数が足りないことがあるので、注意が必要である。

さらに、現実大気実験の場合、初期値以外に `scale-rm_init` で作成した境界値ファイルが必要である。

`[PARAM_ATMOS_BOUNDARY]` の `(ATMOS_BOUNDARY_START_DATE)` に境界値ファイル `boundary_***.nc` の初期時刻を設定しておくことで、リスタート計算時に適切な時刻のデータを読み込むことが可能である。`(ATMOS_BOUNDARY_START_DATE)` の設定がない場合には、境界値ファイル `boundary_***.nc` の最初のデータから読み込まれてしまうので注意が必要である。

<pre>&amp;PARAM_ATMOS_BOUNDARY   ATMOS_BOUNDARY_TYPE = "REAL",   ATMOS_BOUNDARY_IN_BASENAME = "../init/output/boundary_d01",   ATMOS_BOUNDARY_START_DATE = 2010, 7, 14, 18, 0, 0,   ATMOS_BOUNDARY_UPDATE_DT = 21600.D0, /</pre>	<p>現実実験の場合は"REAL"。 境界値データのファイル名の頭。 境界値データの初期時刻。 境界値データのデータ間隔。</p>
--	---

## 5.11 領域ネスティング実験の方法

ネスティングとは、水平格子間隔の異なる複数の計算領域を領域が重複するように入れ子（ネスト）構造に設定する計算領域設定方法である。図 5.5 では、3つの領域を用いた3段ネスティングの例を示している。外側の領域は、大きな空間スケールの現象を表現するため、低い水平解像度で広い領域を設定し、内側の領域は、細かい現象を解像するために、狭い範囲であるが高い水平解像度に設定する。内側の領域は、外側の領域の計算結果を境界値データとして用いる。ここでは、入れ子構造のうち、データを渡す側の領域を「親領域」、データを受ける側の領域を「子領域」と呼ぶことにする。

ネスティングの手法として、下記の種類に分けることができる。

- 実行方法

- オンライン・ネスティング

- 計算の途中で、親領域と子領域の計算結果を逐次やりとりしながら親領域と子領域の計算を同時に実行する方法。

- オフライン・ネスティング

- まず最初に親領域の時間積分を行う。その後、親領域の出力結果を用いて子領域用の初期値・境界値を作成し、子領域の計算を行う方法。

- データの受け渡し方法

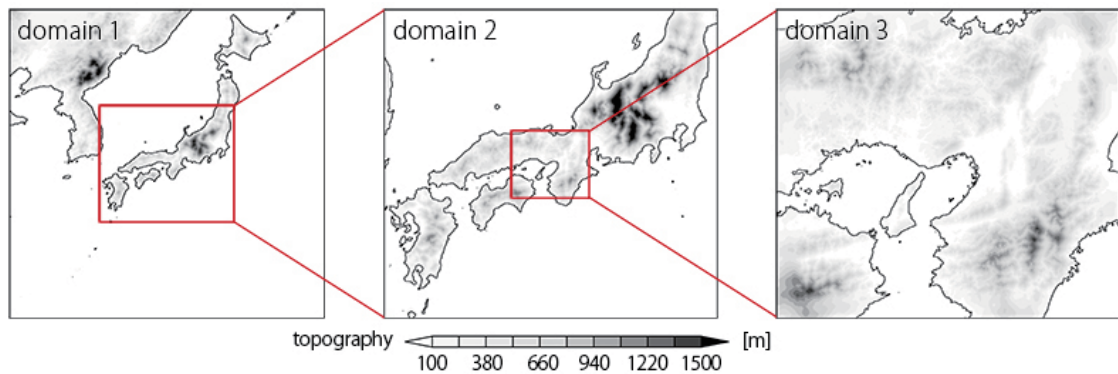


図 5.5: 日本の近畿地方を対象領域とした領域ネスティング設定の例。domain 1 が最外領域で domain 3 が最内領域である。赤い矩形と線は、それぞれの位置関係を示している。domain 1 の水平格子間隔は 7.5 km、domain 2 は 2.5 km、そして domain 3 は 0.5 km である。

#### 一方向ネスティング

親領域のデータは子領域に受け渡されるが、子領域の計算結果は親領域に渡されない。つまり、データの流れは親領域から子領域に向かう一方通行。親領域は子領域の結果の影響を受けない。

#### 双方向ネスティング

親領域のデータは子領域へ、子領域の計算結果は親領域にフィードバックされる。親領域の計算に、子領域の計算結果が反映される。(オンライン・ネスティング時のみ適用可能であるが、SCALE-RM v5.1.5 では実装されていない。)

オンラインとオフラインの違いは、子領域の境界条件に与えられる親領域の情報の更新頻度にある。SCALE-RM 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。オンラインの場合、子領域の境界条件の更新は親領域のタイムステップ ( $\Delta t$ ) 毎に行われるが、オフラインの場合、親領域の history ファイルの出力間隔に依存する。

親領域と子領域の間の格子間隔の比 ( $DX_{d01}/DX_{d02}$ ) は、オフライン、オンラインに関わらずシステム上の制限はないが、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。SCALE-RM では 5 倍以下で使用することを推奨する。

本節、以降の説明では、親領域の設定ファイルを `***.d01.conf`、子領域の設定ファイルを `***.d02.conf` と表記する。

### 5.11.1 子領域における地形の取り扱い

ネスティング実験では、一般的に、親領域と子領域の間で空間解像度が異なるため地形の解像度も異なる。子領域の緩和領域 (第 5.1.4 節) では、親領域の大気データにナッジングを行うが、2つの領域間で地形の表現が異なると、ナッジングに必要な大気データが存在しないことがある。その場合、外挿などにより大気データを見積もることになるが、見積もりが悪いと不整合が生じる。こういった不整合を無くすため、SCALE-RM では、ネスティング計算を行う場合、子領域の緩和領域では親領域の地形をコピーする「地形コピー」機能が推奨されている。この機能を使えば、図 5.6 に示すように、子領域の緩和領域は完全に親領域の地形と一致させることができる。さらに、地形の解像度を内側にいくに従って徐々に高めていくため、緩和領域の内側に地形遷移領域を設定し、そこでは親領域

と子領域の地形を重み付けした値を与える。地形遷移領域の幅は、デフォルト設定では緩和領域と同じ幅である。さらに内側の内部領域では完全に子領域自身の解像度に応じた地形を与える。実験セット一式準備ツール (第 5.13 節) を利用する場合には地形コピー機能は自動的に適用される設定となっている。

ここで説明する地形コピーの設定を記述した `pp.d0*.conf` ファイルは、サンプル設定ファイル `#{Tutorial_dir}/real/sample/USER.online-nesting.sh` を `USER.sh` に置き換えて、実験セット準備ツールを実行した場合に作成される。説明を読み進める上で参考にしてもらいたい。以降、具体的な設定方法と実行手順を説明する。

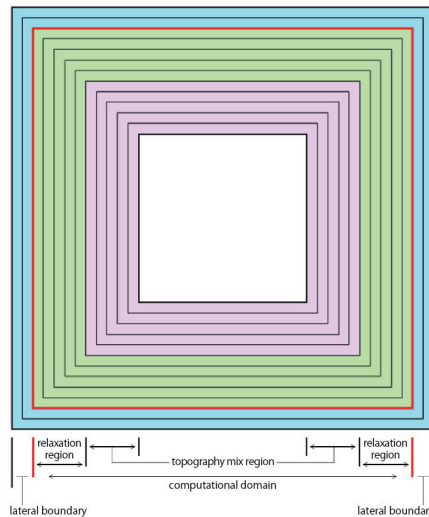


図 5.6: 地形コピーを適用した子領域の地形データ水平分布。最外の水色の格子 (HALO 領域。格子数は水平差分スキームによって異なる。) は側面境界で、それより内側の赤色の線で囲われた領域が計算領域である。緑色の部分は緩和領域、桃色の部分は地形遷移領域、そして最内の白色の部分が子領域の地形をもつ領域である。地形遷移領域 (桃色) では外側から内側にかけて徐々に親領域の地形データから子領域の地形データへ遷移する。

### 地形コピー機能のための設定

まず親領域の地形データ作成時 (`scale-rm_pp`) に計算領域の大きさを子領域へ伝えるためにカタログファイルを出力するよう設定する。具体的には、`pp.d01.conf` に下記の設定が必要となる。

```
&PARAM_DOMAIN_CATALOGUE
  DOMAIN_CATALOGUE_FNAME = "latlon_domain_catalogue.d01.txt",   カタログファイルのファイル名
  DOMAIN_CATALOGUE_OUTPUT = .true.,                             カタログファイルを出力するか
                                                                どうか
/
```

その他の設定項目は通常通りで良い。編集ができれば親領域の地形データ作成を実行する。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ことを想定する。次に、子領域の `pp.d02.conf` ファイルを編集する。

```

&PARAM_NEST
USE_NESTING = .true.,
OFFLINE = .true.,
OFFLINE_PARENT_PRC_NUM_X = 2,
OFFLINE_PARENT_PRC_NUM_Y = 2,
OFFLINE_PARENT_KMAX = 36,
OFFLINE_PARENT_IMAX = 45,
OFFLINE_PARENT_JMAX = 45,
OFFLINE_PARENT_LKMAX = 5,
LATLON_CATALOGUE_FNAME = "latlon_domain_catalogue.d01.txt",
/

&PARAM_CNVTPOPO
  ~ 中略 ~
CNVTPOPO_copy_parent = .true.,
/

&PARAM_COPYTOPO
COPYTOPO_IN_BASENAME = "topo_d01",
COPYTOPO_ENTIRE_REGION = .false.,

COPYTOPO_LINEAR_H = .true.,
/

```

親領域のPRC\_NUM\_X  
 親領域のPRC\_NUM\_Y  
 親領域のKMAX  
 親領域のIMAX  
 親領域のJMAX  
 親領域のLKMAX  
 親領域のカタログファイル

地形コピー機能を適用するかどうか

親領域の地形データの頭  
 子領域の全域に親領域の地形を  
 コピーするかどうか

地形コピーを用いるには、[PARAM\_CNVTPOPO] の (CNVTPOPO\_copy\_parent) に .true. を設定する。(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 5.6 に示されたすべての領域で親領域の地形が完全にコピーされる。(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、親領域の地形と子領域の地形の混合割合が線形的に変化するのに対し、.false. だと指数関数的に変化する。

### 地形の作成

地形コピー機能を使用する場合 (つまり、ネスティング計算の場合)、子領域は親領域の地形データ作成時に出力されるカタログデータを必要とするため、地形データの作成は、親領域から順番に実行する。3 つ以上の領域がある場合も、手続きは同様である。

```

$ mpirun -n [プロセス数] ./scale-rm_pp pp.d01.conf
$ mpirun -n [プロセス数] ./scale-rm_pp pp.d02.conf
$ mpirun -n [プロセス数] ./scale-rm_pp pp.d03.conf

```

### 5.11.2 オフライン・ネスティング実験の方法

オフラインネスティング実験を行う上での実験設定の制限事項は、以下の 2 点である。

- 子領域の計算範囲は、親領域の計算範囲の内側に位置している必要がある。
- 子領域の積分期間は、親領域の積分期間と同じもしくはそれより短い必要がある。

また、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. 親領域の時間積分計算を行う。
2. 親領域の history 出力ファイルを用いて子領域の初期値/境界値を作成する。
3. 作成した初期値/境界値を用いて子領域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。親領域と子領域それぞれについて、`pp.***.conf`、`init.***.conf`、そして `run.***.conf` ファイルを事前に作成し、親領域、子領域ともに地形・土地利用データの作成 (`scale-rm_pp`) が、親領域については、初期値/境界値データの作成 (`scale-rm_init`) が終わっ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設定ファイルは、サンプル設定ファイル `#{Tutorial_dir}/real/sample/USER.offline-nesting-parent.sh` および `#{Tutorial_dir}/real/sample/USER.offline-nesting-child.sh` をそれぞれ `USER.sh` に置き換えて、実験セット一式準備ツールを実行すると作成される。説明を読み進める上で参考にしてもらいたい。

### 親領域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場合と同じ方法で実行すればよいが、`run.***.conf` の設定で次の5点に注意する必要がある。

- 子領域の計算に必要な変数全てを、親領域の計算時に history 出力する。
- 親領域の history 出力間隔を適度に細かくとること。
- 親領域の history 出力データは、モデル面のデータを出力すること。
- 親領域の計算領域を子領域へ伝える「カタログファイル」(以下参照のこと)を出力する。
- (子領域の計算開始時刻が親領域と同じ場合) 親領域の history 出力データに `t=0` の値を含めること。

この設定を `run.d01.conf` に適用すると下記のようなになる。青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_FNAME = "latlon_domain_catalogue_d01.txt",
DOMAIN_CATALOGUE_OUTPUT = .true.,
/

&PARAM_HISTORY
HISTORY_DEFAULT_BASENAME = "history",
HISTORY_DEFAULT_TINTERVAL = 900.DO,
HISTORY_DEFAULT_TUNIT = "SEC",

HISTORY_DEFAULT_TAVERAGE = .false.,
HISTORY_DEFAULT_DATATYPE = "REAL4",
HISTORY_DEFAULT_ZINTERP = .false.,
HISTORY_OUTPUT_STEP0 = .true.,
/

```

カタログファイルのファイル名  
カタログファイルを出力。

history データの出力時間間隔。  
HISTORY\_DEFAULT\_TINTERVAL  
の単位。

モデル面データを出力。  
t=0 の値を出力に含める。

カタログファイルの出力設定を `.true.` にすると、`latlon_domain_catalogue_d01.txt` というカタログファイルが出力される。実験セット準備ツールを使用した場合、同ファイルが `pp` ディレクトリに出力されているので、そちらを参照すること。この中には、親領域の計算で各 MPI プロセスが担

当する計算領域の四隅の緯度・経度が記述されている。`(HISTORY_DEFAULT_TINTERVAL)` は history データの出力間隔を示し、子領域の側面境界条件を更新したい時間間隔に設定する。短い時間間隔でデータを出力する場合には、ディスクの空き容量にも注意が必要である。その他、`[PARAM_HISTORY]` の各項目の詳細は、第 5.6 節を参照のこと。

また、子領域の初期値/境界値データ作成に必要な変数全てを `run.d01.conf` ファイルの `[HISTITEM]` に追加しておく必要がある。オフライン・ネスティングに必要な変数は、下記の通りである。設定が完了したら、`scale-rm` を実行して親領域の時間積分計算を行う。

```
T2, Q2, MSLP, DENS, MOMZ, MOMX, MOMY, RHOT
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
(親の雲微物理モデルに合わせて出力; 例えば Tomita08 なら全て)
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 なら不要)
NC, NR, NI, NS, NG
```

親領域の出力ファイルを用いて子領域の初期値/境界値を作成する

次に、計算が終わった親領域の history データを用いて、子領域の初期値/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-rm_init` だが、`init.d02.conf` を下記のように設定する。



```

&PARAM_NEST
USE_NESTING = .true.,
OFFLINE = .true.,
OFFLINE_PARENT_PRC_NUM_X = 2,
OFFLINE_PARENT_PRC_NUM_Y = 2,
OFFLINE_PARENT_KMAX = 36,
OFFLINE_PARENT_IMAX = 45,
OFFLINE_PARENT_JMAX = 45,
OFFLINE_PARENT_LKMAX = 5,
LATLON_CATALOGUE_FNAME = "latlon_domain_catalogue_d01.txt",

/

&PARAM_MKINIT_REAL_ATMOS
NUMBER_OF_TSTEPS = 25,

FILETYPE_ORG = "SCALE-RM",
BASENAME_ORG = "history_d01",

BASENAME_BOUNDARY = "boundary_d01",
BOUNDARY_UPDATE_DT = 900.DO,

/

&PARAM_MKINIT_REAL_OCEAN
NUMBER_OF_TSTEPS = 25,

BASENAME_ORG = "history_d01",

FILETYPE_ORG = "SCALE-RM",

/

&PARAM_MKINIT_REAL_LAND
NUMBER_OF_TSTEPS = 25,

BASENAME_ORG = "history_d01",

FILETYPE_ORG = "SCALE-RM",

/

```

run.d01.conf のPRC\_NUM\_X  
run.d01.conf のPRC\_NUM\_Y  
run.d01.conf のKMAX  
run.d01.conf のIMAX  
run.d01.conf のJMAX  
run.d01.conf のLKMAX  
親領域を実行した時に作成した  
カタログファイル

history ファイル内の時間ステッ  
プ数  
run.d01.conf の  
HISTORY\_DEFAULT\_BASENAME  
history ファイルの出力時間間  
隔 (単位は"SEC")

history ファイル内の時間ステッ  
プ数  
run.d01.conf の  
HISTORY\_DEFAULT\_BASENAME

history ファイル内の時間ステッ  
プ数  
run.d01.conf の  
HISTORY\_DEFAULT\_BASENAME

SCALE-RM の出力データから初期値境界値を作成する場合は、(FILETYPE\_ORG) に"SCALE-RM"を指定する。(BOUNDARY\_UPDATE\_DT) は、基本的に、親領域の設定ファイル (run.d01.conf) の (HISTORY\_DEFAULT\_TINTERVAL) と同じ設定を記述する。[PARAM\_NEST] の項目は、ネスティング実験のための設定項目である。オフライン・ネスティングでは、(USE\_NESTING=.true., OFFLINE=.true.) と設定する。(OFFLINE\_PARENT\_\*\*\* ) で始まる 6 つの設定変数は親領域の設定を記述する変数である。親領域の設定ファイル (run.d01.conf) を参照して正しく設定すること。

設定の編集が完了したら、scale-rm\_init を実行し、子領域の初期値/境界値を作成する。実行時に下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、子領域の計算領域が親領域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、各領域の大きさや領域中心の設定を見直す必要がある。

```

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

```

## 作成した初期値/境界値を用いて子領域の時間積分計算を行う

初期値/境界値作成が終わったら、子領域の計算 (scale-rm) を実行する。子領域の実行は、通常の現実大気実験と同じである。1点だけ注意すべき点として、run.d02.conf の [PARAM\_ATMOS\_BOUNDARY] の (ATMOS\_BOUNDARY\_UPDATE\_DT) が初期値/境界値作成で使用した親領域の history データ出力間隔に合っているか確認すること。現在のところ、この設定に親領域と子領域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、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。

```
&PARAM_ATMOS_BOUNDARY
  ~ 中略 ~
  ATMOS_BOUNDARY_UPDATE_DT = 900.DO,
/
```

多段のオフライン・ネスティング実験を行いたい場合は、以上の過程を繰り返せばよい。つまり、子領域として時間積分計算した結果を再度、親領域と見立てて、さらに内側の孫領域の初期値/境界値作成を行えばよい。

### 5.11.3 オンライン・ネスティング実験の方法

オンライン・ネスティング実行時の制約として、親領域と子領域の積分時間は一致している必要があり、かつ、親領域の時間ステップは子領域の時間ステップの倍数でなければならない。一方、親領域と子領域の間で鉛直層数、鉛直レベル、地図投影法、物理スキームは一致している必要はない。

オンライン・ネスティング実験では全ての領域の計算を同時に実行する。現在は、親領域から子領域に対してデータの受け渡しを行う一方向ネスティングのみをサポートしている。また、サポートするネスティングの段数は、最大で10段までである。

SCALE-RMのオンライン・ネスティング実験は、複数の領域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 5.7 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領域に分配し、各々の領域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、複数の領域を立ち上げるために実行時には launch.conf という起動用の設定ファイルが別途必要になる。

オンライン・ネスティングに必要な設定は複数ある他、設定の不具合があると計算が正常に行われない。設定の詳細は以下の説明を参照することとし、実際の設定ファイル (\*\*.conf) は第 5.13 節で説明する実験セット作成サポートツールで作成すること。

以下では、最も単純な2段ネスティングの例を示しながら、オンライン・ネスティング実験のための設定について説明する。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設定ファイルは、サンプル設定ファイル \${Tutorial\_dir}/real/sample/USER.online-nesting.sh を USER.sh に置き換えて、実験セット準備ツールを実行した場合に作成される。説明を読み進める上で参考にしてもらいたい。また、すでに、親領域、子領域ともに地形/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ものとして説明を進める。それぞれの領域の地形データの作成手順は、第 5.11.1 節に示したとおりであり、初期値/境界値データの作成は、通常計算 (シングル領域計算) の場合と同じである。

#### 設定ファイルの編集

まず、親領域と子領域のそれぞれの設定ファイル (run.\*\*\*.conf) において、オンライン・ネスティングのための設定を追加する。[PARAM\_NEST] は、ネスティング実験のための設定項目である。



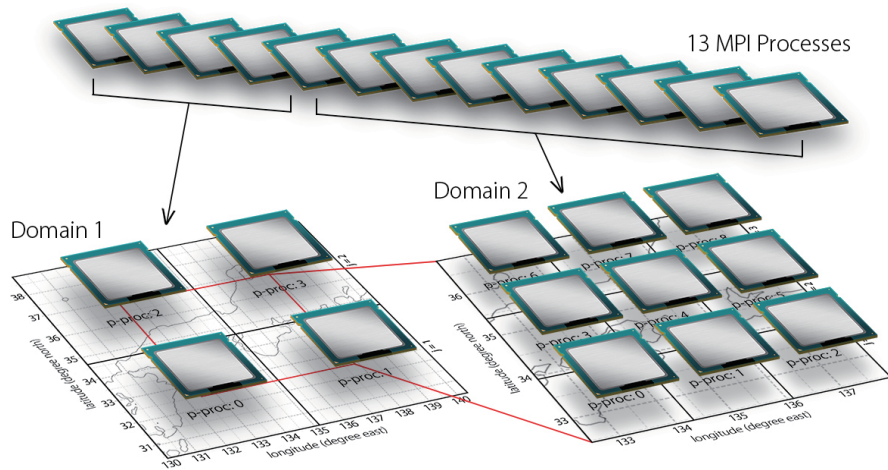


図 5.7: オンライン・ネスティング実験の MPI プロセス配分イメージ. 全部で 13 のプロセスを立ち上げ、これを適切に分配することで、Domain 1 は  $2 \times 2$  の 4-MPI 並列、Domain 2 は  $3 \times 3$  の 9-MPI 並列計算を行う。Domain 1 から Domain 2 へ MPI 通信によってデータを受け渡ししながら時間積分計算を進める。

#### run.d01.conf の編集内容

```
&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.,
/
```

オンラインネスティング時には、.true. とする  
オンラインネスティング時には、.false. とする  
領域の番号。外側から 1 番。

#### run.d02.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.,
/
```

オンラインネスティング時には、.true. とする  
オンラインネスティング時には、.false. とする  
領域の番号。外側から 1 番。

(**USE\_NESTING = .true.**, **OFFLINE = .false.**) 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、領域の ID 番号であり、外側領域から内側領域へ順番に番号を振っていく。ここでは、親領域は 1 番、子領域は 2 番と設定する。(ONLINE\_IAM\_PARENT) と (ONLINE\_IAM\_DAUGHTER) は、各領域が複数のネスティング領域の中でどこに位置するかを示している。namelist の名前から明らかなように、ONLINE\_IAM\_PARENT=.true. であれば、子領域にデータを受け渡し、ONLINE\_IAM\_DAUGHTER=.true. であれば、境界値データは親領域から受け取るように

なる。N 段ネスティング実験の場合の設定例を表 5.16 に示す。

表 5.16: N 段ネスティング実験の設定例

領域	ONLINE_DOMAIN_NUM	ONLINE_IAM_PARENT	ONLINE_IAM_CHILD
最外領域	1	.true.	.false.
中間領域	2~(N-1)	.true.	.true.
最内領域	N	.false.	.true.

最外領域は親領域としてのみ働き、最内領域は子領域としてのみ働く。一方、中間領域は最外領域に対しては子領域、最内領域に対しては親領域として働くため両方共 .true. となる。

(ONLINE\_BOUNDARY\_USE\_QHYD) は、「側面境界条件として親領域の凝結物の値を使うかどうか」を指定する。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、領域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、親領域で計算した凝結物を子領域の境界条件として与えることが可能である。流入境界付近での雲や降水の生成の遅れの影響を小さくすることが期待される。

#### 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、起動用設定ファイル launch.conf が必要である。

```
&PARAM_LAUNCHER
NUM_DOMAIN = 2,           領域の数
PRC_DOMAINS = 4, 16,     それぞれの領域で使用する MPI プロセス数 (領域の数
                           だけ必要)
CONF_FILES = run.d01.conf, run.d02.conf, それぞれの領域の設定ファイル (領域の数だけ必要)
/
```

(PRC\_DOMAINS) と (CONF\_FILES) の記載順は対応している必要がある。上記の例の場合、親領域は 4-MPI 並列、子領域は 16-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、各々の領域の設定ファイル (run.\*\*\*.conf) で指定されている総 MPI プロセス数 (PRC\_NUM\_X×PRC\_NUM\_Y) と一致させなければならない。

実行時には、シングル領域計算とは異なり、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。例えば、上記の例だとプロセス数は 20 となる。

```
$ mpirun -n [プロセス数] ./scale-rm launch.conf
```

実行にあたって注意することは、複数の領域の計算を同時に実行するため、出力ファイル (history ファイル, LOG ファイル, restart ファイルなど) の書き出し先が重複しないように設定する必要がある。例えば、history ファイルは history\_d01.pe\*\*\*.nc, history\_d02.pe\*\*\*.nc といったように領域毎にファイル名を変えることで、どの領域の出力データであるか判別がつくように指定する。出力ファイルのほか、入力ファイルである topo ファイル, landuse ファイル, boundary ファイル, init ファイル の名前も注意が必要である。

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。これは、子領域で設定された計算領域が親領域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領域の西側か南側が親領域の外側に出ており、“NE search” の

エラーが出る場合は子領域の東側か北側が親領域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。

```
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
```

### MPI プロセスの分配ガイドライン

オンライン・ネスティング実験は、図 5.7 に示した通り、複数の領域間で MPI プロセスを共有しない。つまり、それぞれの MPI プロセスは、どれか 1 つのネスティング領域の一部を担当することになる。このため、ユーザは、使用可能な MPI プロセス数のうち、各領域の計算にいくつの MPI プロセスを割り当てるかを定める必要がある。割り当て配分のバランスが悪いと、待ち時間が発生し、計算時間が余計にかかってしまう。これを避けるためには、領域毎に、時間積分にかかる 1 プロセスあたりの計算量（ここでは格子数とタイムステップ数の積として定義）を揃えればよい<sup>\*3)</sup>。具体的な見積もり方法は下記の通りである。

ここでは N 個の領域 (N 段ネスティング) を考える。n 番目の領域の X 方向、Y 方向、Z 方向の格子数をそれぞれ  $IMAX_n$ 、 $JMAX_n$ 、 $KMAX_n$  と表し、時間積分のタイムステップ ( $TIME\_DT$ ) を  $DT_n$  と表すことにする。この時、一番外側領域 ( $n=1$ ) の時間積分のタイムステップ  $DT_1$  を基準とし、この時間を積分するのに必要な n 番目の領域の計算ステップ数は、

$$TSTEP\_n = DT\_1 / DT\_n$$

と表される。領域全体での計算量は、領域が持つ格子数を掛けて

$$OPR\_n = IMAX\_n \times JMAX\_n \times KMAX\_n \times TSTEP\_n$$

と見積もられる。n 番目の領域に配分する MPI プロセス数の目安は、全 MPI プロセス数を  $MPI\_total$  として

$$MPI\_total \times \frac{OPR\_n}{\sum_{m=1}^N OPR\_m} \tag{5.3}$$

と見積もることができる。

X 方向 と Y 方向に分配するプロセス数 ( $PRC\_NUM\_X$ ,  $PRC\_NUM\_Y$ ) には任意性が残るが、 $IMAX$  と  $JMAX$  の違いが小さくなるように設定する方が HALO 領域を少なくすることが出来るため、計算機の演算性能を引き出しやすいと考えられる<sup>\*4)</sup>。

以上の説明では、格子点数と積分時間のタイムステップだけを考慮して演算量比率を考えたが、実際の計算では、物理過程の計算時間間隔も領域毎に異なる可能性があり、領域内通信や領域間通信の MPI 通信にかかる時間の違いも計算時間に影響を及ぼす。オンライン・ネスティングの設定では、最も計算負荷が高い領域 (通常は最内領域) で MPI 通信のための待ち時間が最小となるようにプロセスを分配するのが効率的であることが多い。大規模計算や長期積分、繰り返し行うような実験の場合には、上記の方法で効率的な配分を見積もり、いくらかの微調整を行うことを勧める。

<sup>\*3)</sup> 正確を期すなら演算量を見積もる必要がある。

<sup>\*4)</sup> 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 方向の格子点数を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性も出てくる。

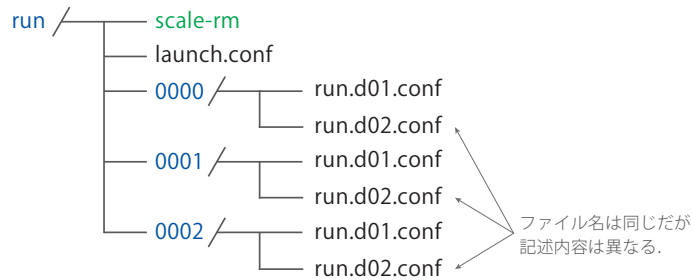


図 5.8: バルクジョブ機能を使って `scale-rm` を実行する場合のディレクトリ構造. “0000” や “0001” はジョブ番号に対応する名前のディレクトリ (ジョブディレクトリ) である。各ジョブディレクトリの中には、サブジョブの実行に必要なすべてのファイルを用意する。

## 5.12 複数の実験を一括実行するバルクジョブの設定

SCALE-RM 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。バルクジョブ機能とは、複数の独立した実験を 1 つのジョブとして実行する機能である。これは、パラメタスイープ実験、初期値アンサンブル実験、タイムスライス気候実験など、似たような設定を大量に行う場合に便利な機能である。

バルクジョブ機能は、モデル本体 (`scale-rm`) の実行はもちろん、地形・土地利用データ作成 (地形コピー (第 5.11.1 節) を利用しない場合に限る) (`scale-rm_pp`)、初期値/境界値作成 (`scale-rm_init`) の他、ネスティング実験にも適用可能である。

以下の説明では、1 のバルクジョブに含まれる独立した実行命令を「サブジョブ」と呼ぶこととする。ここでは、3 つの 2 段オンライン・ネスティング実験を例に説明する (積分期間、もしくは計算領域中心が異なっている 3 つのサブジョブを想定している)。各サブジョブの設定内容のうち、`launch.conf` で指定される設定 (`launch.conf` の `[PARAM_LAUNCHER]` の `(NUM_DOMAIN, PRC_DOMAINS, CONF_FILES)`; 第 5.11.3 節参照) は共通していなければならないが、それ以外の設定内容 (例えば、積分時間や使用するスキーム、1 つの MPI プロセスあたりの格子数など) は異なっても構わない。

バルクジョブ機能は、オンライン・ネスティングで利用した MPI プロセスを分割・分配する機能を拡張したものである。従って、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場合も `launch.conf` ファイルは 1 つで良い。

```
&PARAM_LAUNCHER
NUM_BULKJOB = 3,           サブジョブの数
NUM_DOMAIN = 2,          ネスティング領域の数
PRC_DOMAINS = 9, 36,
CONF_FILES = run.d01.conf, run.d02.conf,
/
```

上記は、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` ファイルの例である。サブジョブで設定しているオンライン・ネスティング実験の用の `launch.conf` ファイルに、`(NUM_BULKJOB)` の項目を加えればよい。それ以外の設定の詳細は、第 5.11.3 節の通りである。シングルドメイン実験 (ネスティングを使用しない) の場合は、`(NUM_DOMAIN = 1)` と指定し、`(CONF_FILES)` に設定ファイルを 1 つ指定すればよい。

次に、バルクジョブの実行にあたり、まず、サブジョブの数だけディレクトリ (ジョブディレクトリと呼ぶ) を用意する (図 5.8 の 0000/ 0001/ 0002/)。ディレクトリ名は、サブジョブのジョブ

番号を4桁の数字で表したもので、ジョブ番号はゼロから設定される。それぞれのジョブディレクトリには、各サブジョブの実験に必要なすべてのファイル(設定ファイル、入力ファイル、出力ファイルを格納するディレクトリなど)を用意する。各ジョブディレクトリ内の設定ファイルは、通常の設定と同じであるが、適切な入力ファイルが設定されているかどうか、注意が必要である。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 ~
/

```

図5.8のとおり、ジョブディレクトリは実行バイナリと同じ階層にある。つまり、設定ファイルは各ジョブディレクトリの下にあるが、入力ファイルや出力先のディレクトリは、実行バイナリの位置から見た相対パスを記述する必要がある。従って、ジョブ0000番の実験は、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/であり、出力ファイル名は0000/\*\*となる。例えばジョブディレクトリを付けるのを忘れてhistory\_d01とするなど、すべてのサブジョブで同じファイル名を設定してしまうと、同じファイルにすべてのサブジョブが書き込みを行うため、データが消失してしまうことに注意すること。

バルクジョブの実行は、全サブジョブを実行するのに必要なMPIプロセス数を指定し、

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

と行う。例では、一つのサブジョブあたり、 $9 + 36 = 45$  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ファイルを開くと、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。下記、ドメイン1のプロセス0からの出力例である。



```

++++++ Start MPI
*** UNIVERSAL_COMM_WORLD : 0           ; 実行環境によって値が異なる
*** total process [UNIVERSAL] : 135
*** my process ID [UNIVERSAL] : 36
*** master rank? [UNIVERSAL] : F
*** GLOBAL_COMM_WORLD : 3             ; 実行環境によって値が異なる
*** total process [GLOBAL] : 45
*** my process ID [GLOBAL] : 36
*** master rank? [GLOBAL] : F
*** LOCAL_COMM_WORLD : 4              ; 実行環境によって値が異なる
*** 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、[UNIVERSAL] と表記されている項目はジョブグループに関する情報である。LOCAL グループは GLOBAL グループに包含され、さらに GLOBAL グループは UNIVERSAL グループに包含される。total process は各グループの全プロセス数、my process ID はあるグループで見た時のプロセス番号を表している。

この例では、total process [UNIVERSAL] は 135 と表記され、実行時に指定したとおり、全体で 135 のプロセスが起動したことがわかる。また、total process [GLOBAL] は 45 であり、これは 1 サブジョブあたり 45 プロセスを使用したことを表している。この例はドメイン 1 における LOG メッセージであるため、total process [LOCAL] は 9 で正しい。ドメイン 2 の LOG メッセージを確認した場合は、36 と表記される。LOG ファイルや history ファイルの番号に対応するプロセス番号は、my process ID [UNIVERSAL] である。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。現在のバルクジョブ機能では、1つのジョブが異常終了となると、全てのジョブが強制終了されることに注意すること。

## 5.13 設定ファイルと実験に必要なファイル一式を用意する

現在、実験のための設定ファイル\*\*\*.conf は pp, init, run 用にそれぞれ用意することになっているが、本章で説明した通り、3つの設定ファイルで共通するネームリストが、不一致である場合モデルが動かない。設定ミスを防ぎ、簡易に設定ファイルを用意するためのサポートツールが下記に用意されている。

```

$ cd $Tutorial_DIR/real/
$ ls
  Makefile : 実験セット一式作成のための Make ファイル
  README   : 実験セット一式作成ツールに関する README
  USER.sh  : 実験設定の指定
  config/  : 実験セット一式作成一つのためのファイル (ユーザは基本書き換えない)
  sample   : USER.sh のサンプルスクリプト

```

これは、初期設定として現実大気実験のチュートリアルに合わせた設定となっているが、ユーザが任意の設定に変更することが可能な仕組みとなっている。本章で説明した箇所の多くを USER.sh で設定できるようになっている。

また、`sample/` 以下に、いくつかの実験設定を想定したサンプルスクリプトが用意されている。必要に応じて、`USER.sh` にコピーの上、使用していただきたい。

```
$ ls sample/
USER.default.sh           : USER.sh と同じ。チュートリアル実大気実験用。(シングルメイン用)
USER.offline-nesting-child.sh : オフラインネスティングの子領域用。
USER.offline-nesting-parent.sh : オフラインネスティングの親領域用。
USER.online-nesting.sh     : オンラインネスティング用。
```

## ツールの使い方

使い方は README に書かれているように、以下の通りである。

1. ユーザが希望する任意の実験設定に従って、`USER.sh` を編集する。
2. `make` コマンドを実行する。
3. 実験に必要な設定ファイル一式が `experiment` ディレクトリ以下に作成される。

現在は、`tutorial` の設定になっているので、実際には以下のようにチュートリアル設定を残しておくことを勧める。

```
$ mv experiment/ tutorial/ : (すでに experiment ディレクトリがある場合)
$ cp USER.sh USER_tutorial.sh
... USER.sh を編集 ...
$ make
$ cp -rL experiment 任意の場所/ : SCALE のソースがあるディレクトリ以外の場所も可
```

## USER.sh の編集

まず、サンプルプログラムの中、最も想定する実験設定に近いスクリプトを `USER.sh` に上書きコピーする。スクリプトの中を見ると、まず、ドメインの数を指定する `NUM_DOMAIN` がある。その下には、設定ファイルのネームリストで設定する項目が並んでいるので、それぞれの項目の設定を書き込んでいけば良い。"# required parameters for each domain" というコメントがあるところにはドメインの数だけ設定変数をスペースで区切って用意する。この時、`NUM_DOMAIN` で設定したネスティングドメインの数と変数の数が不一致だとメッセージが出力され、実験セットが作成されずに終了してしまうので必ず一致させること。`USER.sh` にはない項目については、デフォルトの設定で出力されるので `experiment` ディレクトリ以下に作成された設定ファイルを直接変更すればよい。

## 関連図書

- A. C. M. Beljaars and A. A. M. Holtslag. Flux parameterization over land surfaces for atmospheric models. *J. Appl. Meteor.*, 30:327341, 1991.
- A. R. Brown, S. H. Derbyshire, and P. J. Mason. Large-eddy simulation of stable atmospheric boundary layers with a revised stochastic subgrid model. *Quarterly Journal of the Royal Meteorological Society*, 120:1485–1512, 1994.
- Committee on Space Research; NASA National Space Science Data Center. Cospar international reference atmosphere (cira-86): Global climatology of atmospheric parameters, 2006.
- J. W. Deardorff. Stratocumulus-capped mixed layers derived from a three-dimensional model. *Boundary-Layer Meteorology*, 18:495–527, 1980.
- J. Dudhia. Numerical study of convection observed during the winter monsoon experiment using a mesoscale twodimensional model. *J. Atmos. Sci.*, 46:3077–3107, 1989.
- E. Kessler. On the distribution and continuity of water substance in atmospheric circulation. *Meteorological Monograph*, 10:1–84, 1969.
- Barry Koren. *A robust upwind discretization method for advection, diffusion and source terms*. Centrum voor Wiskunde en Informatica Amsterdam, 1993.
- 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*, 14:148–171, 1962.
- J.-F. Louis. A parametric model of vertical eddy fluxes in the atmosphere. *Boundary-Layer Meteorol.*, 17:187–202, 1979. doi: 10.1007/BF00117978.
- G. L. Mellor and T. Yamada. Development of a turbulence closure model for geophysical fluid problems. *Rev. Geophys. Space Phys.*, 20:851–875, 1982.
- M. J. Miller, A. C. M. Beljaars, and T. N. Palmer. The sensitivity of the ecmwf model to the parameterization of evaporation from the tropical oceans. *Journal of Climate*, 5(5):418–434, 1992.
- I.-J. Moon, I. Ginis, T. Hara, and B. Thomas. A physics-based parameterization of air-sea momentum flux at high wind speeds and its impact on hurricane intensity predictions. *Mon. Weather Rev.*, 135:2869–2878, 2007. doi: 10.1175/MWR3432.1.
- 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.



- S. Nishizawa, H. Yashiro, Y. Sato, Y. Miyamoto, and H. Tomita. Influence of grid aspect ratio on planetary boundary layer turbulence in large-eddy simulations. *Geosci. Model Dev.*, 8(10): 3393–3419, 2015. doi: 10.5194/gmd-8-3393-2015.
- Katsuyuki V Ooyama. A dynamic and thermodynamic foundation for modeling the moist atmosphere with parameterized microphysics. *Journal of the atmospheric sciences*, 58(15):2073–2102, 2001.
- J-L Redelsperger, PRA Brown, F Guichard, C How, M Kawasima, S Lang, Th Montmerle, K Nakamura, K Saito, C Seman, et al. A gcss model intercomparison for a tropical squall line observed during toga-coare. i: Cloud-resolving models. *Quarterly Journal of the Royal Meteorological Society*, 126(564):823–863, 2000.
- J. J. Remedios, R. J. Leigh, A. M. Waterfall, D. P. Moore, H. Sembhi, I. Parkes, J. Greenhough, M. P. Chipperfield, and D. Hauglustaine. Mipas reference atmospheres and comparisons to v4.61/v4.62 mipas level 2 geophysical data sets. *Atmos. Chem. Phys. Discuss.*, 7:9973–10017, 2007. doi: doi:10.5194/acpd-7-9973-2007.
- Y. Sato, S. Nishizawa, H. Yashiro, Y. Miyamoto, Y. Kajikawa, 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(1):1–16, 2015. doi: 10.1186/s40645-015-0053-6.
- A. Scotti, C. Meneveau, and D. K. Lilly. Generalized smagorinsky model for anisotropic grids. *Physics of Fluids A*, 5:2306–2308, 1993.
- 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*, 109:2779–2793, 2008.
- J. Smagorinsky. General circulation experiments with the primitive equations. *Mon. Weather Rev.*, 91:99–164, 1963.
- K. Suzuki, T. Nakajima, T. Y. Nakajima, and A. P. Khain. A study of microphysical mechanisms for correlation patterns between droplet radius and optical thickness of warm clouds with a spectral bin microphysics cloud model. *Journal of the Atmospheric Sciences*, 67(4):1126–1141, 2010.
- Team SCALE. *The detail formulation of SCALE-RM*. 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.
- I Uno, X-M Cai, DG Steyn, and S Emori. A simple extension of the louis method for rough surface layer modelling. *Boundary-Layer Meteorology*, 76(4):395–409, 1995.

- Louis J Wicker and William C Skamarock. Time-splitting methods for elastic models using forward time schemes. *Monthly Weather Review*, 130(8):2088–2097, 2002.
- D. K. Wilson. An alternative function for the wind and temperature gradients in unstable surface layers. *Bound-Lay. Meteorol.*, 99:151–158, 2001.
- S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.*, 31:335–362, 1979. doi: 10.1016/0021-9991(79)90051-2.

## 付 録 A ライブラリ環境のインストール

SCALE のインストールに必要なコンパイラやライブラリ環境のインストール方法について説明する。ここでの記載内容は、こちらのテスト環境でのインストールプロセスを示しているものであって必ずしも全く同じとは限らない。うまくいかない場合には、それぞれのツール・ライブラリの開発元に直接問い合わせること。

Linux をインストール後、各種プログラムのインストールはコマンドライン端末にて行う。本書で説明するライブラリ環境のインストールでは、root 権限が必要になる。従って、想定する環境は、ユーザが root 権限を所持しているかサーバやデスクトップマシンである。別途サーバ管理者が存在し、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サーバ管理者に問い合わせること。

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コンパイラおよびライブラリ環境は、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 6.6-6.8 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 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 wgrib wgrib2
```

SCALE は陰解法計算の部分で、数値計算ライブラリ Lapack <sup>\*1)</sup> を利用するオプションがある。もし必要ならば、Lapack もインストールすること。

\*1)<http://www.netlib.org/lapack/>

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

\* `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` のように “-y” オプションをつけて実行することで、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 7.1-7.2 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: 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 wgrib wgrib2
```

SCALE は陰解法計算の部分で、数値計算ライブラリ Lapack <sup>\*2)</sup> を利用するオプションがある。もし必要ならば、Lapack もインストールすること。

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

\* 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。

\* 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

---

\*2) <http://www.netlib.org/lapack/>

## 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 (ocefpaf's Home Project; <https://build.opensuse.org/project/show/home:ocefpaf>) からインストールする。このため、まず home\_ocefpaf 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 では一般にユーザが root ユーザ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:/ocefpaf/openSUSE_13.2/ \\  
home_ocefpaf
```

\*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_ocefpaf' を追加しています ..... 完了  
リポジトリ 'home_ocefpaf' 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:/ocefpaf/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\_ocefpaf 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 openmpi-devel openmpi-devel-static  
$ sudo zypper install wgrib wgrib2
```

SCALE は陰解法計算の部分で、数値計算ライブラリ Lapack <sup>\*3)</sup> を利用するオプションがある。もし必要ならば、Lapack もインストールすること。

```
$ sudo zypper install lapack-devel lapack-devel-static
```

\* 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、最初に以下のよう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 (root 権限必要)

```
# sudo xcodebuild -license
```

---

\*3) <http://www.netlib.org/lapack/>

次に macports 本体をインストールする。<https://www.macports.org/> 必要なパッケージインストーラーをダウンロードし、インストールを進める。

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い。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。この例では gcc4.9 を利用する。

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

macports では複数のコンパイラと MPI ライブラリをインストール出来るため、その中で利用するものを選択する必要がある。今回の場合、gcc と MPI ライブラリが該当する。この操作を行うと、gfortran 等の一般的な名前でもエイリアスが作られて PATH が通るようになる。

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

インストールした NetCDF ライブラリを用いるときの PATH の設定は以下の通りである。 .bash\_profile をエディタで開き、

```
$ emacs ~/.bash_profile
```

下記をファイルに追加して、環境変数の設定を記述する。

```
export NETCDF_INCLUDE="-I/opt/local/include"  
export NETCDF_LIBS="-L/opt/local/lib -lnetcdff -L/opt/local/lib -Wl,\  
-headerpad_max_install_names -lnetcdf"
```

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
```

追加する環境変数の設定は以下の通りである。

```
export LAPACK_LIBS="-L/opt/local/lib -llapack -lcblas -lf77blas -latlas"
```

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

GPhys は Ruby のパッケージ管理システム RubyGems を通してインストールできる。詳細な情報については、(<https://www.hdfgroup.org/HDF5/>) を参照されたい。まず必要であれば ruby のインストールを macports を用いて行う。この例では Ruby2.1 を利用することにする。

```
$ sudo port install ruby21  
$ sudo port select --set ruby ruby21
```

次に macports を用いて、GPhys に必要なライブラリをインストールする。

```
$ sudo port install fftw-3  
$ sudo port install gsl  
$ sudo port install C-DCL6
```

最後に RubyGems を用いて、GPhys をインストールする。

```
$ sudo gem install gphys
```

## Mac OS X での実行時の注意点

Mac OS X を用いて SCALE-RM プログラムを実行すると、実行時に「アプリケーション”scale-rm”へのネットワーク受信接続を許可しますか？」というダイアログが出ることがあります。これはコンパイルしたバイナリがマシンをまたいだ MPI 通信をするかファイアウォール機能が確認するためです。コンパイルし直すたびに MPI 並列数の分だけダイアログが出てきてしまいますが、今のところ表示を回避するためには「システム環境設定」の「セキュリティとプライバシー」項目で「ファイアウォール」のタブを選択し、プログラムの実行時にファイアウォールを切る方法しかありません。

## 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 地球電脳倶楽部
  - 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 : 地球電脳倶楽部の Web ページに、主な OS でのインストール方法についての解説がある。  
<https://www.gfd-dennou.org/library/ruby/tutorial/install/index-j.html>  
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/library/cc-env/rpm-dennou/index.html.ja>  
Mac OS X におけるインストール方法は第 A.4 節で説明している他、<https://www.gfd-dennou.org/library/ruby/products/macports/index-j.html> でも解説されている。
- Grid Analysis and Display System (GrADS) by COLA
  - URL : <http://cola.gmu.edu/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://cola.gmu.edu/grads/downloads> 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、openSUSE13ではhome\_osefpaf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。GrADS本体のインストールし、バイナリが保存されている場所にPATHを通す。次に、描画に必要なスクリプト集(.gs)を<ftp://cola.gmu.edu/grads/scripts>からダウンロードする。例えばwgetを用いる場合は  
「`wget ftp://cola.gmu.edu/grads/scripts/*.gs`」  
スクリプトを保存した場所に環境変数GASCRPを設定する。  
例えばスクリプトを/usr/local/grads/script/に保存した場合は  
「`export GASCRP="/usr/local/grads/script"`」
- Nview: 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ーである。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、openSUSE13ではhome\_osefpaf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。



## 付 録 B 設定ファイル”run.conf”のネームリストと出力変数リスト

設定ファイル”run.conf”のネームリスト

SCALE-RM ドキュメントページ

Configuration List の NAMELIST Parameters

<http://scale.aics.riken.jp/doc/5.1.5/namelist.html>

history 出力できる変数一覧

SCALE-RM ドキュメントページ

Configuration List の History Variables

<http://scale.aics.riken.jp/doc/5.1.5/history.html>

## 付録C よくある質問とその回答：FAQ

ここでは、よくある質問を具体的な練習問題と共に列挙し、その後、それぞれの問題に対する回答を示す。まずは自分で答を考えることにより、より理解が深まることを期待している。

### 練習問題

1. 計算領域を変えず、MPI 並列数を変更したい  
第4章の現実実験チュートリアルの設定ファイル\*\*`.conf`について、4-MPI 並列の設定を6-MPI 並列に変更する。(参考：5.1.1 節、5.1.2 節)
2. MPI 並列数を変えず、計算領域を変更したい  
第4章の現実実験チュートリアルの設定について、MPI 並列数を変更せず、計算領域を  $x$  方向に  $4/3$  倍に拡大、 $y$  方向は  $2/3$  倍に縮小する。(参考：5.1.2 節)
3. 計算領域を変えず、水平格子間隔を変更したい  
第4章の現実実験チュートリアルの設定について、計算領域を変えず、水平格子間隔を 5km に変更する。(参考：5.1.2 節、5.1.3 節、5.1.4 節、5.5 節)
4. 計算領域の位置を変更したい  
第4章の現実実験チュートリアルの設定について、計算領域の大きさは変えず、中心位置を経度 139 度 45.4 分、緯度 35 度 41.3 分に変更する。(参考：5.2 節)
5. 積分時間を変更したい  
第4章の現実実験チュートリアルの設定について、6 時間積分から 12 時間積分に変更する。(参考：5.5 節)
6. 出力変数の追加と出力間隔を変更したい  
第4章の現実実験チュートリアル`の計算出力の設定で、出力の時間間隔を 30 分に変更し、地表面での下向き短波放射と上向き短波放射の出力を追加する。(参考：5.6 節、付録 B)`
7. リスタート計算をしたい  
第4章の現実実験チュートリアル`の計算について、最初に 3 時間動かして停止し、リスタートファイルから再開してさらに 3 時間計算し、計 6 時間計算する。(参考：5.10 節、5.4)`

## 回答

### 1. 計算領域は変えず、MPI 並列数を変更したい

変更箇所は、[PARAM\_INDEX] 内の (IMAX, JMAX)、[PARAM\_PRC] 内の (PRC\_NUM\_X, PRC\_NUM\_Y) である。下記 3 つの式を満たしていれば正解である。

$$\text{MPI 並列数} = (\text{PRC\_NUM\_X}) \times (\text{PRC\_NUM\_Y}) = 6$$

$$X \text{ 方向の格子数} = (\text{IMAX} \times \text{PRC\_NUM\_X}) = 90$$

$$Y \text{ 方向の格子数} = (\text{JMAX} \times \text{PRC\_NUM\_Y}) = 90$$

### 2. MPI 並列数は変えず、計算領域を変更したい

MPI プロセスあたりの格子数を  $n$  倍にすれば、領域全体の格子数も  $n$  倍となる。変更箇所は、[PARAM\_INDEX] 内の (IMAX, JMAX) のみである。赤文字の部分がデフォルトからの変更点を意味する。

```
&PARAM_INDEX
  KMAX = 36,
  IMAX = 60,      (オリジナル設定は IMAX = 45)
  JMAX = 30,      (オリジナル設定は JMAX = 45)
/
```

### 3. 計算領域は変えず、水平格子間隔を変更したい

MPI 並列数を変えない場合、変更箇所は [PARAM\_GRID] の (DX, DY) と [PARAM\_INDEX] 内の (IMAX, JMAX) である。

```
&PARAM_PRC
  PRC_NUM_X = 2,
  PRC_NUM_Y = 2,

&PARAM_INDEX
  KMAX = 36,
  IMAX = 180,     (オリジナル設定は IMAX = 45)
  JMAX = 180,     (オリジナル設定は JMAX = 45)
/

&PARAM_GRID
  DX = 5000.DO,  (オリジナル設定は DX = 20000.DO)
  DY = 5000.DO,  (オリジナル設定は DY = 20000.DO)
/
```

MPI 並列数も変更している場合には、下記の関係を満たしていれば正解である。

$$X \text{ 方向の格子数} = (\text{IMAX} \times \text{PRC\_NUM\_X}) = 360 \quad (\text{C.1})$$

$$Y \text{ 方向の格子数} = (\text{JMAX} \times \text{PRC\_NUM\_Y}) = 360$$

これに加えて、力学変数の時間積分のためのタイムステップ

(TIME\_DT\_ATMOS\_DYN) と (ATMOS\_DYN\_TINTEG\_SHORT\_TYPE) の調整も必要である (第 5.5 節参照)。

緩和領域も、格子間隔の 20 倍から 40 倍となるよう変更する。

```
&PARAM_PRC
  BUFFER_DX = 100000.DO, (オリジナル設定はBUFFER_DX = 400000.DO,)
  BUFFER_DY = 100000.DO, (オリジナル設定はBUFFER_DY = 400000.DO,)
/
```

#### 4. 計算領域の位置を変更したい

計算領域の中心位置の座標を下記の通り、変更すれば良い。ここで、単位を度で設定する必要があることに注意すること。例えば、139 度 45.4 分 = 139 + 45.5/60 度。

```
&PARAM_MAPPROJ
  MPRJ_basepoint_lon = 139.7567D0, (オリジナル設定は MPRJ_basepoint_lon = 135.220404D0,)
  MPRJ_basepoint_lat = 35.6883D0, (オリジナル設定は MPRJ_basepoint_lat = 34.653396D0,)
  MPRJ_type = 'LC',
  MPRJ_LC_lat1 = 30.00D0,
  MPRJ_LC_lat2 = 40.00D0,
/
```

#### 5. 積分時間の変更したい

```
&PARAM_TIME
  TIME_STARTDATE = 2007, 7, 14, 18, 0, 0,
  TIME_STARTMS = 0.DO,
  TIME_DURATION = 12.00D0, (オリジナル設定は 6.00D0,)
  TIME_DURATION_UNIT = "HOOR",
/
```

また、scale-rm\_init の境界値を 12 時間以上用意している必要がある。第 5.4 節を参照し、(NUMBER\_OF\_FILES) の数を 3 に変更する。

#### 6. 出力変数の追加と出力時間間隔の変更したい

例えば、出力時間間隔を 30 分に変更したい時には、[PARAM\_HISTORY] の中の (HISTORY\_DEFAULT\_TINTERVAL) を下記のように変更する。また、出力変数を追加したい時には、[HISTITEM] の中の (ITEM) に追加したい変数を指定する。変数名に関しては、B 章を参照のこと。以下では、地表面での上向き、下向き短波フラックスを追加した例である。

```
&PARAM_HISTORY
  HISTORY_DEFAULT_BASENAME = "history_d01",
  HISTORY_DEFAULT_TINTERVAL = 1800.DO, (オリジナル設定は3600.DO,)
  HISTORY_DEFAULT_TUNIT = "SEC",
/

&HISTITEM item="SFLX_SW_up" /      追加
&HISTITEM item="SFLX_SW_dn" /      追加
```

#### 7. リスタート計算をしたい

最初の 3 時間分の計算設定は下記の通りである。3 時間目にリスタートファイルを出力する。

```

&PARAM_TIME
TIME_STARTDATE = 2007, 7, 14, 18, 0, 0,
TIME_STARTMS = 0.DO,
TIME_DURATION = 3.0DO,          3時間以上であれば良い。
TIME_DURATION_UNIT = "HOUR",
....(省略)....
TIME_DT_ATMOS_RESTART = 10800.DO,
TIME_DT_ATMOS_RESTART_UNIT = "SEC",
TIME_DT_OCEAN_RESTART = 10800.DO,
TIME_DT_OCEAN_RESTART_UNIT = "SEC",
TIME_DT_LAND_RESTART = 10800.DO,
TIME_DT_LAND_RESTART_UNIT = "SEC",
TIME_DT_URBAN_RESTART = 10800.DO,
TIME_DT_URBAN_RESTART_UNIT = "SEC",
/

&PARAM_RESTART
RESTART_RUN = .false.,          追加。なくても良い。
RESTART_OUTPUT = .true.,       (オリジナル設定は.false.,)
RESTART_IN_BASENAME = "../init/init_d01_20070714-180000.000",
RESTART_OUT_BASENAME = "restart_d01",      追加
/

&PARAM_ATMOS_BOUNDARY
ATMOS_BOUNDARY_TYPE = "REAL",
ATMOS_BOUNDARY_IN_BASENAME = "../init/output/boundary_d01",
ATMOS_BOUNDARY_START_DATE = 2010, 7, 14, 18, 0, 0,      追加。なくてもよい。
ATMOS_BOUNDARY_UPDATE_DT = 21600.DO,
/

```

(TIME\_DURATION) が 3 時間に設定されている場合には、リスタートファイルは積分終了時に出力されるため、(TIME\_DT\_ATMOS\_RESTART)、(TIME\_DT\_OCEAN\_RESTART)、(TIME\_DT\_LAND\_RESTART)、(TIME\_DT\_URBAN\_RESTART) の設定は必要ない。また、(TIME\_DURATION) が 3 時間以上に設定されている場合には、(TIME\_DT\_ATMOS\_RESTART)、(TIME\_DT\_OCEAN\_RESTART)、(TIME\_DT\_LAND\_RESTART)、(TIME\_DT\_URBAN\_RESTART) に 3 時間、もしくは 10800 (sec) の約数かつ (TIME\_DT) の倍数を指定している必要がある。

3 時間目から 6 時間目までのリスタート計算のための設定は下記の通りである。

```

&PARAM_TIME
  TIME_STARTDATE = 2007, 7, 14, 21, 0, 0,
  TIME_STARTMS = 0.DO,
  TIME_DURATION = 3.0D0,
  TIME_DURATION_UNIT = "HOUR",
/

&PARAM_RESTART
  RESTART_RUN = .true.,
  RESTART_OUTPUT = .true.,
  RESTART_IN_BASENAME = "restart_d01_20070714-210000.000",
  RESTART_OUT_BASENAME = "restart2_d01",
/

&PARAM_ATMOS_BOUNDARY
  ATMOS_BOUNDARY_TYPE = "REAL",
  ATMOS_BOUNDARY_IN_BASENAME = "../init/output/boundary_d01",
  ATMOS_BOUNDARY_START_DATE = 2010, 7, 14, 18, 0, 0,
  ATMOS_BOUNDARY_UPDATE_DT = 21600.DO,
/

```

3 時間以上であれば良い。

必須  
あってもなくてもよい。  
必須  
あってもなくてもよい。

必須

# SCALE-RM USERS GUIDE [日本語版]

## 執筆・編集

---

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

足立 幸穂、安藤 和人、佐藤 陽祐、富田 浩文、西澤 誠也、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。

Copyright ©Team SCALE, RIKEN AICS, 2016. All rights reserved.