ＷＲＦ　マニュアル

(ＷＲＦＶ２__ＷＲＦＳＩ用)

＊チュートリアル

初版　２００７年3月２９日

文責：小田僚子 ・ 尤　龍涛

目次

～はじめに～

WRFSI は WPS の前バージョンで、2007 年 3 月現在では世界的に WPS の使用に以降してきています。しかしながら、WRFSI もまだ使用でき、また、私たちが最初に WRF を始めたときに書き進めたマニュアルを無駄にしたくなかったので、一応 WRFSI 用のマニュアルとしてまとめておくことにしました。

私たちのように UNIX に使い慣れていない人のためにも、できる限り細かい事項をすべて書き込んで、“この通りにやれば誰でも WRF を動かせる！”というマニュアルを作るつもりです。自分が実際 WRF を動かしてみて引っかかったところなど、特に重点的に書いていきます。

コマンドは**緑色**で書き、解説と区別しています。基本的なコマンドの説明はしていませんが、このような基本コマンドの説明などは、神田研 HP（http://www.cv.titech.ac.jp/~kandalab/ja/index.html）にも掲載されていますし、参考書にも細かく丁寧に解説されていますので、そちらを参照してください。

この WRF マニュアル（WRFV2_WRFSI 用） ＊チュートリアルは、WRF ホームページに掲載されている onlone tutorial（http://www.unidata.ucar.edu/software/netcdf/index.html）を和訳したものです。パスの設定などは、TSUBAME（東工大スパコン）用に修正を加えています。

WRF は約半年に一度バージョンアップされています。また、バグなどがあった場合にはそれらの情報が随時 WRF ホームページ（http://www.mmm.ucar.edu/wrf/users/）にて更新されているので、度々ホームページを訪れるようにしてください。

＊TSUBAME への接続

～ TSUBAME（東工大スパコン）で WRF を使ってみよう！

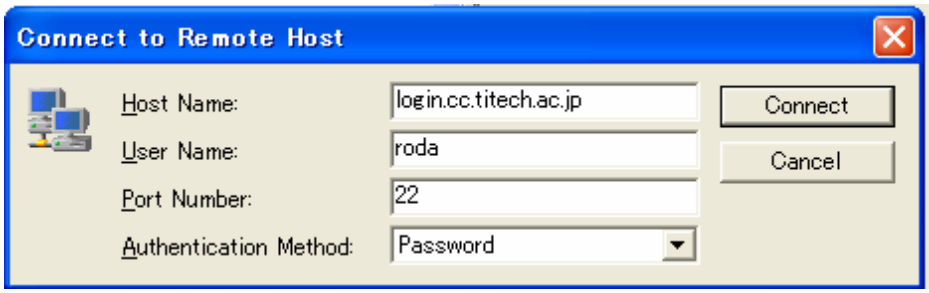
＊ 大岡山スパコンのアカウントを取得する。
（東工大学術国際情報センターの HP から申請書を DL できる。http://www.gsic.titech.ac.jp/）

＊ センターから折り返し、ログイン名とパスワードが送付されてくるので、早速スパコンにアクセスしてみる。

＜自分の PC からアクセスする方法＞

① SSHSecureShellClient というフリーのソフトウェアをネットから、もしくは使っている人から入手する。（SSH に対応している Tera Term などでも良いが、SSHSecureShellClient は SSH のウィンドウから FTP ソフトをワンクリックで立ち上げることができるので便利）
SSHSecureShellClient-3.2.9.exe

② SSHSecureShellClient をインストールできたら、ダブルクリックして起動し、以下の画面のように Host Name などを打ち込む。　もちろん、User Name は自分のものを入力すること。これでコネクトすると、次に Password を聞かれるので、センターからもらったパスワードを入力する。エラーがでたら、センターに問い合わせよう（内線 2035）。



③ スパコンにアクセスできたら、まずは初期パスワードの変更をする。
パスワードは以下のルールに従い、設定する。なお、パスワードの有効期限は 24 週なので、有効期限が切れた際は再度設定しなおす必要がある。

- 8 文字の英数字（大文字・小文字の区別あり）や符号を組み合わせる。
- 英数字のみで設定する場合は、最低 1 文字は大文字を使用する。
- ユーザー名と同一のパスワードは設定しない。
- 英単語等、辞書に載っているものは設定しない。

パスワードは以下の手順で変更する。

➢ **passwd**

Changing password for user user001
Enter login(LDAP) password :　旧パスワード
New UNIX passwoed:　新パスワード
Retype new UNIX password:　新パスワード
LDAP password information changed for user001
passwd: all authentication tokens updated successfully.

X-window（画像などを表示するためのツール）を使うことを考えて、以下のようにログインしなおす。

➢ **ssh –Y –l roda login.cc.titech.ac.jp　（←自分のユーザー名を打つ）**

＊NetCDF

～ NetCDF（配列型データフォーマットの一種）を扱えるようにしよう！

WRF では NetCDF 形式を使用している。そのため事前に NetCDF をインストールする。

以下からダウンロード。

http://www.unidata.ucar.edu/software/netcdf/index.html

- **gunzip netcdf.tar.gz**
- **tar xvf netcdf.tar**
- **cd netcdf-3.6.2**
- **make distclean**
- **./configure –disable-cxx --prefix=/home1/usr2/roda/netcdf-3.6.2** （自分の home ディレクトリにインストールする）
- **vi Makefile**

  この中で、INSTALL = /usr/bin/install -c

  であることを確認する。もし違っていたら、上記のように書き換える。
- **:q!** （上書きせずに quit。編集して上書きする場合は :wq）
- **make check**
- **make install**

  このとき、prefix した場所（今回の場合は /home1/usr2/roda/netcdf-3.6.2）に bin, lib, include, man といったディレクトリができていることを確認。

Environment Variable -NetCDF

NetCDF の環境変数を定義する。

通常、NETCDF libraries は　/usr/local/netcdf　（/home1/usr2/roda/netcdf）に設定する。

- **setenv NETCDF /home1/usr2/roda/netcdf-3.6.2**（自分の home ディレクトリにインストールする）

ここで、setenv を入力してコマンドが違うというメッセージが出てきたら、csh ではない。

- **echo $SHELL**

と入力し、自分のシェルを確かめる。シェルを変更するには、

- **chsh**

を入力する。そうすると以下のようなメッセージが現れる。

```
Changing login shell for roda.
Password:
Enter the new value, or press return for the default.
  Login Shell [/usr/bin/csh]: /usr/bin/csh　（変更したいシェルを直接入力する）
Shell changed.
```

＊NetCDF は常にパスが通っている必要があるので、csh（上記）を使用している場合は .cshrc に、bash を使用している場合には .bashrc に環境変数を設定しておくと便利。

１．WRF テストラン

最初に、WRF を動かすことができるかどうか、テストランを行う。

http://www.mmm.ucar.edu/wrf/OnLineTutorial/index.htm

に、詳細な online tutorial が掲載されているので、それらを参照にする。

以下に、上記のＨＰに掲載されている内容を、コメントを加えつつ記載していく。

なお、必要なソースコードは現在の時点（2005/5）で最新のものをまとめて、atmos の

/work/database/WRF/Source_Codes_and_Graphics_Software

に入れてある。（念のため、ＣＤにも焼いておきますね。）

Program Flow

必要に応じて、以下のプログラムをダウンロードする。

- 理想的なケースのみをシミュレーションする場合

  WRF-ARW Model + Post Processing
- 実際のケースをシミュレーションする場合

  WRFSI + WRF-ARW Model + Post Processing
- 影響変動値を考慮した実際のケースをシミュレーションする場合

  WRFSI + WRF-Var + WRF-ARW Model + Post Processing

（もし WRF-Var を使うことを考えている場合は、最初に WRF-Var Online Tutorial で扱い方を学んでください）

Documentation

Users' Guide

WRF OnLine Tutorial は全て User's Guide に書かれている。User's Guide は半年毎に更新されるので、WRF ARW model を使うための最新の情報を入手できる。Model を走らせる前に、この User's Guide をダウンロードしておくこと。

WRF ARW Technical Note

このノートには以下の内容が含まれている。

- ARW model の方程式、打切り（discretization）、初期設定、ネスティングの概要
- モデルの中で利用可能な Physical Options の概要
- WRF-Var の概要

Bi-Annual Tutorial Presentation

全てのスライドは、最新のバージョンを掲載している。

WRFSI

このページから、WRFSI の説明文書だけでなく、ソフトウェアもダウンロードできる。

WRF-Var

WRF-Var の説明文書と WRF-Var OnLine Tutorial へのリンクについては、このページを参照すること。

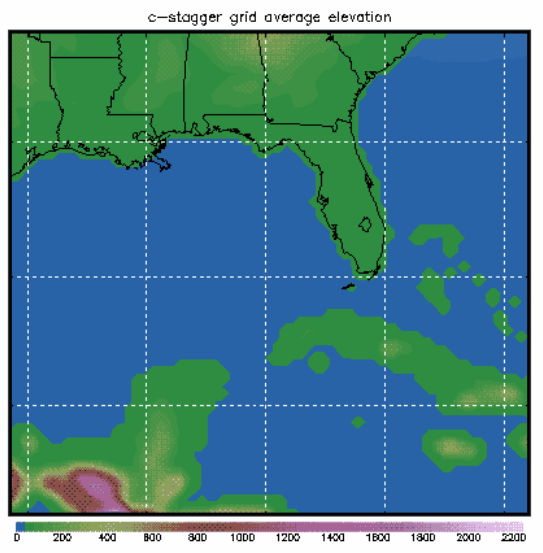
＊HP（WRF ARW Users Pages）でも、より多くの情報や説明文書を見ることができる。

Case Study

ここでは練習として、Hurricane Katrina (August 28, 2005) についてシミュレーションしてみる。
Case Study を動かすため、またデータをダウンロードするために十分なスペースのある場所に、新しい working directory を作る。例えば、

- mkdir WRF
- cd WRF

入力データとして、global AVN data を用いる。
この case study の領域は右図に示す通りである。



Introduction

WRFSI は ARW モデルを構成する上での重要で必要な最初の step である。SI は WRF において３つの不可欠な部分を提供している。

- ３次元グリッドの定義
- 陸、水、植物の“静的な”表面特性の指定
- 外部モデルデータを領域へ挿入することによる、初期および水平境界条件 files の提供

Get Terrestrial Input Data

下記 web sites にある terrestrial data をダウンロードする。
（atmos の/work/database/WRF/Source_Codes_and_Graphics_Software にあります。）
http://wrfsi.noaa.gov/release/
http://www.mmm.ucar.edu/wrf/users/download/get_source.html

Run WRFSI with GUI

もし、GUI を実行したいならば、以下のように入力する。X-window を使う。
eXodus を実行しておく。

- ssh –X login.cc.titech.ac.jp
- export DISPLAY=172.20.25.24:0.0 （自分の PC の IP アドレスを打つこと）
- LANG=C
- export LANG

- qmon &
- cd /home1/usr2/roda/WRF/wrfsi/
- ./wrf_tools

画面上に GUI が現れる。詳細は、WRFSI web site を参照すること。

NCL_COMMAND

GUI を用いる場合、この環境変数を設定する。GUI を用いて、terrestrial input data の画像を作ることができるようになる。

NCL_COMMAND=/home1/usr2/roda/ncl;export NCL_COMMAND

1.1 環境変数の設定

Set Environment Variables

WRFSI をインストール・実行する前に用意する必要のある環境変数を記述する。
環境変数のいくつかは、もし設定しないならば初期値を持つが、設定する方が、どこにそれがあるのかがわかるので、理解が深まる。
環境変数を定義するか、初期値を使うか、どちらにしても、後で参照するために、
/home1/usr2/roda/WRF/wrfsi/の中の file config_paths の中で、それらを見つけることができる。
これらの環境変数を正しく設定することは、SI プログラムを動かすときに重要である。Run scripts と source code がデータにアクセスするためにこれらの環境変数を打ち込む。
Input GriB file があるところの directory は、環境変数が定義されないことに注意する。namelist file that the grib_prep プログラムを使うところで定義される。

SOURCE_ROOT

source root directory へのパス

- SOURCE_ROOT=/home1/usr2/roda/WRF/wrfsi;export SOURCE_ROOT

INSTALLROOT

install root directory へのパス

- INSTALLROOT=/home1/usr2/roda/WRF/wrfsi;export INSTALLROOT

EXR_DATAROOT

degribbed（媒体の）file がある directory へのパス

- EXT_DATAROOT=/home1/usr2/roda/WRF/wrfsi/extdata;export EXT_DATAROOT

TEMPLATES

stemplate directory を置きたい場所。この directory は、自分自身が扱うケースを構築するとき修正するために必要な SI namelist file を含んでいる。GUI を用いない SI を動かすときだけ関連する。

- TEMPLATES=/home1/usr2/roda/WRF/wrfsi/templates;export TEMPLATES

DATAROOT & MOAD_DATAROOT

自分のケースを動かしたい場所。

DARAROOT は複数の sundirectory（MOAD_DATAROOT）を含む最上部の directory に置くことができる。仮にそこにひとつの MOAD_DATAROOT しかないとき、MOAD_DATAROOT はまた DATAROOT directory と同じとすることもできる。

- **DATAROOT=/home1/usr2/roda/WRF/wrfsi/domains;export DATAROOT**

ただし、wrfsi/data directory は設定しないこと。

MOAD_DATAROOT は single case を動かす directory である。この環境変数について設定されている初期値はない。一般的に以下のように定義する。（後で定義するので、ここでは無視してよし）

- **MOAD_DATAROOT=/home1/usr2/roda/WRF/wrfsi/domains/your-case-name;export MOAD_DATAROOT**

GEOG_DATAROOT

この環境変数は、自分の terrestorial input data（terrain,landuse,etc.）の場所を指定するものである。つまり、この場所（ここでは GEAG）に、最初にダウンロードしたソースコードを入れておかなくてはいけない。

atmos に置いておいたデータのうち、

/Source_Codes_and_Graphics_Software/WRF_Standard_Initialization/WRF_Standard_Initialization_input_data 内にあるフォルダ（９個）を、

/home1/usr2/roda/WRF/wrfsi/extdata/GEOG に置く。

GEOG フォルダは自分で作成する。

- **cd /home1/usr2/roda/WRF/wrfsi/extdata**
- **mkdir GEOG**
- **GEOG_DATAROOT=/home1/usr2/roda/WRF/wrfsi/extdata/GEOG;export GEOG_DATAROOT**

1.2 WRFSI のインストール

Get Source Code

最初に述べたように、必要なソースコードは現在の時点で最新のものをまとめて、atmos の /work/database/WRF/Source_Codes_and_Graphics_Software に入っている。

WRFSI の source code は、HP からもダウンロードできる。

http://wrfsi.noaa.gov/release/

atmos の中の Source_Codes_and_Graphics_Software フォルダ内にある、

WRF_Standard_Initialization フォルダを自分の PC 上にダウンロードしておく。

WRF という working directory を作る（最初からこのマニュアルを読んで、既に作ってあれば必要ありません）。

- **mkdir WRF**

ここに、WRF_Standard_Initialization/ WRF_Standard_Initialization_program_tar_file にある、wrfsi_v2.1.2.tar.gz を置く。

Unpack the Code

WRF ディレクトリ上で、TAR file を開く。

- **cd WRF**
- **gunzip wrfsi_v2.1.2.tar.gz**

（随時バージョンは更新されていくので、最新のものを使うようにしていく）

TAR file を解凍する。

- **tar –xvf wrfsi_v2.1.tar**

この後、新しく wrfsi/ ディレクトリが作成されているはず。

Examine the Source Code

- **cd wrfsi**

中身を確認すると、下記のような感じになっている。

- **ls -all**

```
total 156
drwxr-xr-x    9 roda user   4096 Mar 10 08:08   .
drwxr-xr-x    3 roda user   4096 May 25 19:04   ..
-rw-r--r--    1 roda user 15975 Jun 19   2004   CHANGES
-rw-r--r--    1 roda user 11166 Mar 12   2003   HOW_TO_RUN.txt
-rw-r--r--    1 roda user  4405 Jan 18   2003   INSTALL
-rw-r--r--    1 roda user  4101 Dec 11   2003   Makefile
-rw-r--r--    1 roda user 30421 Jan 28   2005   README
-rw-r--r--    1 roda user 12661 Mar  8   2005   README.wrfsi_nl
drwxr-xr-x    7 roda user   4096 Aug 23   2005   data
drwxr-xr-x    2 roda user   4096 Mar  6 07:18   etc
drwxr-xr-x    7 roda user   4096 Aug  3   2005   extdata
drwxr-xr-x    3 roda user   4096 Aug  3   2005   graphics
drwxr-xr-x    5 roda user   4096 Mar 10 08:11    gui
-rwxr-xr-x    1 roda user 27153 Mar 10 08:08    install_wrfsi.pl
drwxr-xr-x   12 roda user   4096 Aug  3   2005   src
drwxr-xr-x    3 roda user   4096 Aug 23   2005   util
```

CHANGES は 最近更新されたすべての code の紹介が書かれている。
HOW_TO_RUN.txt・INSTALL・README・README.wrfsi_nl は、プログラムのインストールや実行に関する有益な情報が書かれている。実行する前に、一読した方がよい。
install_wrfsi.pl は、WRFSI をコンパイルまたはインストールする perl script である。

Install WRFSI

- perl install_wrfsi.pl

画面に以下のようなメッセージが現れる。

Routine: Install_wrfsi
Path to perl: /usr/bin/perl
--path_to_netcdf not specified, attempting to determine...
netCDF path found from environment variable.
Do you want to install the WRF SI graphical user interface? [y | n]:

ここで、" n "を選択すると WRFSI のみインストール。" y "を選択すると、GUI もインストールする。
ここでは、" y " を選択することにする。(例；次のように表示される。" n "、" y ")

- y

インストールに成功すると、下記ディレクトリに以下のような実行ファイルが出来ているはず。

- cd /home1/usr2/roda/WRF/wrfsi/bin/

-rwxr-xr-x 1 roda user 870534 Jul 12 16:01 grib_prep.exe
-rwxr-xr-x 1 roda user 1901081 Jul 12 16:02 gridgen_model.exe
-rwxr-xr-x 1 roda user 1760763 Jul 12 16:01 hinterp.exe
-rwxr-xr-x 1 roda user 1647080 Jul 12 16:02 siscan
-rwxr-xr-x 1 roda user 1754324 Jul 12 16:02 staticpost.exe
-rwxr-xr-x 1 roda user 2018603 Jul 12 16:01 vinterp.exe

また、下記の makefile が作られていることを確認しておく。

- cd /home1/usr2/roda/WRF/wrfsi/src/include

makefile_alpha.inc.in

＊ もし、install に失敗したら…
ARW OnLine Tutorial に掲載されている、考えられる error の対処法を参照すること。

1.3 WRFSI の実行

Run WRFSI Manually

1.3 .1 STEP1

STEP1: Locallize model domain and create static files

ここでは、対象とする領域を設定し、WRF を動かすために必要な静的ファイルを作る。
この step は、各モデル領域設定のために一度だけ必要となる。
Locarization のために必要とされる script; /home1/usr2/roda/WRF/wrfsi/etc/window_domain_rt.pl

Namelist; wrfsi.nl

ここでテストケースを Online Tut と呼ぶことにする。

- **cd domains**
- **mkdir OnlineTut**

MOAD_DATAROOT を設定する。

- **MOAD_DATAROOT=/home1/usr2/roda/WRF/wrfsi/domains/OnlineTut;export MOAD_DATAROOT**

TEMPLATE を作成する。

- **cd /home1/usr2/roda/WRF/wrfsi/templates**
- **cp –r default OnlineTut**
- **chmod –R u+w OnlineTut**
- **cd OnlineTut**

wefsi.nl の編集

wrfsi.nl は step1（localization）と step3（interpolation of data）で用いる。
wrfsi.nl に示される赤の部分は step1 で、紫の部分は step3 に関連する。
今の段階で両方とも編集しておくことをお勧めする。
Namelist 変数の詳細は、README.wrfsi_nl を参照すること。

1. hgridspec を編集することにより、扱うモデル領域を設定する。今回は hurricane Katrina を扱う。
2. ここでポイントは、terrestrial data へのパスが通っていること、つまり環境変数が正しく設定されていることである。sfcfiles をチェックすること。
3. interp_control 部分で、データの内挿を設定する。INIT_ROOT と LBC_ROOT がこのケースに対して正しく設定されているか確認する。AVN データを用いるとき、これらのパラメータは両方とも AVN に設定する必要がある。もし違うモデルを動かすときに sigma lebel を変更する場合、ここの段階で LEVELS も変更する必要がある。今回のケースの場合は、初期値 31 levels の設定のまま使用する。
4. si_path が intermediate fails（/home1/usr2/roda/WRF/wfsi/extdata/）のところに設定されているかを確認する。

一度、このモデル構成について編集すると、モデル領域を適用できる準備が整う。

- **cd /home1/usr2/roda/WRF/wrfsi/**
- **./etc/window_domain_rt.pl –w wrfsi –t /home1/usr2/roda/WRF/wrfsi/templates/OnlineTut**

ここで実行すると、このような画面が表示される。
log file（/home1/usr2/roda/WRF/wrfsi/domains/OnlineTut/log/localize_domain.log.date）で実行が成功したかを確認する。
以下の static file が作成されていれば成功している。

- **/home1/usr2/roda/WRF/wrfsi/domains/OnlineTut/static/static.wrfsi.d01**

ncdump –h /home1/usr2/roda/WRF/wrfsi/OnlineTut/static/static.wrfsi.d01 を使用することができ、このファイルに書かれた場所を見ることができる。

このファイルが作成できれば、GRIB ファイルを入力する deGrib へ進める。

1.3 .2 STEP2

STEP2: DeGrib GRIB files

全ての GRIB fails を deGrib する必要がある。各データセットについて一度する必要がある。

Script needs to deGrib GRIB file: /home1/usr2/roda/WRF/wrfsi/etc/grib_prep.pl

Namelist: /home1/usr2/roda/WRF/wrfsi/extdata/static/grib_prep.nl

通常は自分の考えるケースを動かす場合は、それに対応した GRIB を得る必要がある。今回のケースである、AVN test data for the Katrina case は web から入手できる。

Katrina case

- Global AVN/GFS (90.0 to -90.0 by 1.0 latitude & 0.0 to 360.0 by 1.0 longitude)
- List of field available in these files
- Start date: 2005082800
- End date: 2005083100
- Frequency: 6 hourly

このデータをダウンロードし、別の directory に置く。(今後、各ケースの GRIB file を異なる directory に置いておき、それぞれの directory で動かす方が良い)

/home1/usr2/roda/WRF/wrfsi/extdata/static/grib_prep.nl の編集

重要なのは gpinput_defs の部分だけである。

AVN データがあるとき、3rd path in the SRCPATH パラメータは、GRIB input file が置かれている directory を指定して設定されていなければならない。ここでは、Katrina という directory を作り、GRIB file を置くことにする。

- **mkdir Katrina (/home1/usr2/roda/WRF/wrfsi 上に)**
- **tar xvfz Katrina_AVN_input.tar.gz**

script を動かす準備をする。

- **cd /home1/usr2/roda/WRF/wrfsi/**
- **./etc/grib_prep.pl –s 2005082800 -l 72 –t 6 AVN**

実行が終了したら、/home1/usr2/roda/WRF/wrfsi/extdata/log/gp_AVN.200508280.log を確認する。

もし成功していたら、/home1/usr2/roda/WRF/wrfsi/extdata/extprd に intermediate files が作られている。

```
-rw-rw-rw-   1 roda user 38867544 May 30 22:11 AVN:2005-08-28_00
-rw-rw-rw-   1 roda user 38867544 May 30 22:11 AVN:2005-08-28_06
-rw-rw-rw-   1 roda user 38867544 May 30 22:11 AVN:2005-08-28_12
-rw-rw-rw-   1 roda user 38867544 May 30 22:11 AVN:2005-08-28_18
-rw-rw-rw-   1 roda user 38867544 May 30 22:11 AVN:2005-08-29_00
-rw-rw-rw-   1 roda user 38867544 May 30 22:11 AVN:2005-08-29_06
-rw-rw-rw-   1 roda user 38867544 May 30 22:11 AVN:2005-08-29_12
-rw-rw-rw-   1 roda user 38867544 May 30 22:11 AVN:2005-08-29_18
-rw-rw-rw-   1 roda user 38867544 May 30 22:11 AVN:2005-08-30_00
```

-rw-rw-rw- 1 roda user 38867544 May 30 22:11 AVN:2005-08-30_06
-rw-rw-rw- 1 roda user 38867544 May 30 22:11 AVN:2005-08-30_12
-rw-rw-rw- 1 roda user 38867544 May 30 22:11 AVN:2005-08-30_18
-rw-rw-rw- 1 roda user 38867544 May 30 22:11 AVN:2005-08-31_00

ここまでできたら、step1（localization）で作成されている WRF model 領域に内挿する準備が整った。

1.3 .3 STEP3

STEP3: Interpolate meteorological data

ここでは、モデル領域に気象データを挿入する。hinterp と vinterp の両方が実行される。

Script needed: /home1/usr2/roda/WRF/wrfsi/etc/wrfprep.pl

Namelist: wrfsi.nl

＊ step1 で既に wrfsi.nl を編集していれば、ここで編集する必要はない。

＊ もし、wrfprel.pl を実行する前に、inter control もしくは si paths の部分を変更したいならば、/home1/usr2/roda/WRF/wrfsi/domains/OnlineTut/static/wrfsi.nl を編集しなければならない。/home1/usr2/roda/WRF/wrfsi/templates/OnlineTut/wrfsi.nl をコピーしない。

script を動かす準備をする。

- **cd /home1/usr2/roda/WRF/wrfsi/**
- **./etc/wrfprep.pl –s 2005082800 -f 72 –t 6**

これは wrfsi.nl の時間設定部分（filetimespec）であるが、コマンドでも上書きすることができる。（-s=start time, -f=forecast length, -t=interval）

実行画面として以下が表示される。

Routine: wrfprep.pl

INSTALLROOT = /home1/usr2/roda/WRF/wrfsi

MOAD_DATAROOT = /home1/usr2/roda/WRF/wrfsi/domains/OnlineTut

Start time: 2005/08/28 00:00:00

End time: 2005/08/31 00:00:00

CYCLE.0524000000072

ictime = 2005-08-28_00

lbctime = 2005-08-28_06

lbctime = 2005-08-28_12

lbctime = 2005-08-28_18

lbctime = 2005-08-29_00

lbctime = 2005-08-29_06

lbctime = 2005-08-29_12

lbctime = 2005-08-29_18

lbctime = 2005-08-30_00

lbctime = 2005-08-30_06

lbctime = 2005-08-30_12

lbctime = 2005-08-30_18
lbctime = 2005-08-31_00
/home1/usr2/roda/WRF/wrfsi/domains/OnlineTut/siprd/hinterp.d01.2005-08-31_00:00:00
/home1/usr2/roda/WRF/wrfsi/domains/OnlineTut/siprd/wrf_real_input_em.d01.2005-08-31_00:00:0
0

実行が完了したら、/home1/usr2/roda/WRF/wrfsi/domains/OnlineTut/log にある log file を確認する。以下が出来ていたらよい。

2005082800.hinterp
2005082800.vinterp
2005082800.wrfprep

成功すると、/home1/usr2/roda/WRF/wrfsi/domains/OnlineTut/siprd に以下の wrf_real_inut files が作成されている。

-rw-r--r-- 1 roda user 185 Jun 1 18:15 CYCLE.0524000000072
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-28_00:00:00
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-28_06:00:00
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-28_12:00:00
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-28_18:00:00
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-29_00:00:00
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-29_06:00:00
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-29_12:00:00
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-29_18:00:00
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-30_00:00:00
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-30_06:00:00
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-30_12:00:00
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-30_18:00:00
-rw-r--r-- 1 roda user 2821104 Jun 1 18:14 hinterp.d01.2005-08-31_00:00:00
-rw-r--r-- 1 roda user 368 Jun 1 18:14 hinterp.global.metadata
-rw-r--r-- 1 roda user 4338960 Jun 1 18:14 wrf_real_input_em.d01.2005-08-28_00:00:00
-rw-r--r-- 1 roda user 4338576 Jun 1 18:14 wrf_real_input_em.d01.2005-08-28_06:00:00
-rw-r--r-- 1 roda user 4338576 Jun 1 18:14 wrf_real_input_em.d01.2005-08-28_12:00:00
-rw-r--r-- 1 roda user 4338576 Jun 1 18:14 wrf_real_input_em.d01.2005-08-28_18:00:00
-rw-r--r-- 1 roda user 4338576 Jun 1 18:14 wrf_real_input_em.d01.2005-08-29_00:00:00
-rw-r--r-- 1 roda user 4338576 Jun 1 18:14 wrf_real_input_em.d01.2005-08-29_06:00:00
-rw-r--r-- 1 roda user 4338576 Jun 1 18:14 wrf_real_input_em.d01.2005-08-29_12:00:00
-rw-r--r-- 1 roda user 4338576 Jun 1 18:14 wrf_real_input_em.d01.2005-08-29_18:00:00
-rw-r--r-- 1 roda user 4338576 Jun 1 18:14 wrf_real_input_em.d01.2005-08-30_00:00:00
-rw-r--r-- 1 roda user 4338576 Jun 1 18:14 wrf_real_input_em.d01.2005-08-30_06:00:00
-rw-r--r-- 1 roda user 4338576 Jun 1 18:15 wrf_real_input_em.d01.2005-08-30_12:00:00

-rw-r--r-- 1 roda user 4338576 Jun 1 18:15 wrf_real_input_em.d01.2005-08-30_18:00:00
-rw-r--r-- 1 roda user 4338576 Jun 1 18:15 wrf_real_input_em.d01.2005-08-31_00:00:00

ここまでできたら、WRF ARM モデルを実行する準備が整った。

1.4 WRFV2 のインストール

Introduction

WRF ARM model は完全圧縮・非静水モデル（静水 option 付）である。鉛直座標は静水圧に基づく地形座標（σ座標)、格子は荒川 C 格子で、高次の数値モデルである。2 次・3 次の Runge-Kutta 法および水平・鉛直両方向に対して 2 次～6 次の移流スキームを含む。音波・重力波に対して細かな time-split で使用できる。力学過程はスカラー変数を保存する。

WRF ARF model code はいくつかの初期プログラム（ideal.exe, real.exe)、数値積分プログラム（wrf.exe)、one-way nesting をするためのプログラム（ndown.exe）を持つ。WRF ARF model Version2.1 はさまざまな能力に対応している。これらは以下の事項を含んでいる。

- Real-data and idealized simulations
- Various lateral boundary condition options for both real-data and idealized simulations
- Full physics options
- Non-hydrostatic and hydrostatic (runtime option)
- One-way, two-way nesting and moving nest
- Applications ranging from meters to thousands of kilometers

Software Requirement

- Fortran 90 or 95 and c compiler
- perl 5.04 or better
- もし MPI や OpenMP が要求されるなら、MPI もしくは OpenMP libraries を必要とする
- WRF I/O API は netCDF, PHD5, GriB 1 形式に対応しており、そのためこれらのうちどれか１つの librarie が WRF の実行やコンパイルする場所のコンピュータ上で利用可能である必要がある。

Get Source Code

ソースコードを以下の HP からダウンロードする。すでにダウンロードしている場合は必要ない。

http://www.mmm.ucar.edu/wrf/users/download/get_source.html

ユーザー登録するとダウンロードできる仕組みになっている。メールアドレスを入力するだけの簡単な登録で、１度登録すると、次回からはメールアドレスを入力するだけでダウンロードページに進むことができる。

最新の WRF TAR file を working directory (WRF/)に置く。

Unpack the Code

TAR file を解凍する。

- **tar –xvf WRFV2.1.1.TAR**

これにより、WRFV2/ という新しい directory が作成される。

Examine the Source Code

- **cd WRFV2**

この directory 内には、以下の file と directory が存在する。

```
-rw-r--r--    1 roda user 16421 Oct 21   2005 Makefile
-rw-r--r--    1 roda user  7250 Nov  9   2005 README
-rw-r--r--    1 roda user  7238 Oct  5   2005 README.NMM
-rw-r--r--    1 roda user  2548 May 18   2004 README_test_cases
drwxr-xr-x    2 roda user  4096 Apr 11 16:00 Registry
drwxr-xr-x    2 roda user  4096 Nov  9   2005 arch
drwxr-xr-x    2 roda user  4096 Nov  9   2005 chem
-rwxr-xr-x    1 roda user  1078 May 24   2005 clean
-rwxr-xr-x    1 roda user  6913 Oct 21   2005 compile
-rwxr-xr-x    1 roda user 10636 May  7   2005 configure
drwxr-xr-x    2 roda user  4096 Apr 20 20:01 dyn_em
drwxr-xr-x    2 roda user  4096 Nov  9   2005 dyn_exp
drwxr-xr-x    2 roda user  4096 Nov  9   2005 dyn_nmm
drwxr-xr-x   12 roda user  4096 Nov  9   2005 external
drwxr-xr-x    2 roda user  4096 Apr 11 16:02 frame
drwxr-xr-x    2 roda user  4096 Jun  1 16:44 inc
drwxr-xr-x    2 roda user  4096 Jun  1 16:44 main
drwxr-xr-x    2 roda user  4096 Apr 11 16:13 phys
drwxr-xr-x    2 roda user  4096 Jun  1 16:44 run
drwxr-xr-x    2 roda user  8192 Jun  1 16:44 share
drwxr-xr-x   11 roda user  4096 Nov  9   2005 test
drwxr-xr-x    3 roda user  4096 Apr 11 16:01 tools
```

＊ README file には code についての有益な情報とモデル実行と設定方法が書かれている。

＊ ソースコード directory は以下の通りである。

chem/	Directory containing modules for chemistry (not currently supported)
dyn_em/	Directory containing modules for dynamics in WRF ARW core
dyn_nmm/	Directory containing modules for dynamics in WRF NMM core
dyn_exp/	Directory for a 'toy' dynamical core
external/	Directory containing external packages, such as those for IO, time keeping and MPI
frame/	Directory containing modules for WRF framework
inc/	Directory containing include files
main/	Directory for main routines, such as wrf.F, and all executables after install

phys/ Directory for all physics modules
share/ Directory containing mostly modules for WRF mediation layer and WRF I/O
tools/ Directory containing tools

＊Script は以下の通りである。

clean Script to clean created files, executables
compile Script for compiling WRF code
configure Script to configure the configure.wrf file for compile

＊ Makefile Top-level makefile
＊ Registry/ Directory for WRF Registry file
＊ arch/ Directory where compile options are gathered
＊ run/ Directory where one may run WRF
＊ test/ Directory that contains 7 test case directories, may be used to run WRF

Environment Variable –NETCDF

NETCDF 環境変数を設定する。(既に行っている場合は再度設定する必要はないが、一度 logout している場合は再度設定しなくてはならない。)

- **NETCDF=/home1/usr2/roda;export NETCDF**

1.5 WRFV2 のコンパイル

Configure WRFV2

- **./configure**

ここで、TSUBAME でサポートしているプラットフォームの一覧が表示される。

checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /home1/usr2/roda
PHDF5 not set in environment. Will configure WRF for use without.

Please select from among the following supported platforms.

1. PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher (Single-threaded, no nesting)
2. PC Linux x86_64 (IA64 and Opteron), PGI 5.2 or higher, DM-Parallel (RSL, MPICH, Allows nesting)
3. PC Linux x86_64 (IA64 and Opteron), PGI 5.2 or higher DM-Parallel (RSL_LITE, MPICH, Allows nesting, No periodic LBCs)

Enter selection [1-3] :

今回は、最も基本的（singre-threaded, no nesting）な 1 を選ぶことにする。

- 1

configure.wrf file が作成される。必要であれば、このファイルにある compile options/paths を編集すること。

Compile WRFV2

- ./compile

Usage:

compile wrf　compile wrf in run dir (NOTE: no real.exe, ndown.exe, or ideal.exe generated)

or choose a test case (see README_test_cases for details) :

compile em_b_wave

compile em_grav2d_x

compile em_hill2d_x

compile em_quarter_ss

compile em_real

compile em_squall2d_x

compile em_squall2d_y

compile exp_real

compile nmm_real

compile -h　　help message

WRF ARW real data case を compile するので、em_real を選択する。

- ./compile em_real >& compile.log

各種理想実験、予報実験毎にロードモジュールを作り分ける仕組みになっている。テストデータを使って予報してみるので、em_real を選択する。少し時間がかかる。

compile.log ファイルを確認する。成功していれば、main/ directory に以下の実行ファイルが作成されている。

-rwxr-xr-x　1 roda user 13612035 Jul 13 15:11 ndown.exe

-rwxr-xr-x　1 roda user 10909533 Jul 13 15:11 real.exe

-rwxr-xr-x　1 roda user 13237353 Jul 13 15:11 wrf.exe

ndown.exe は one-way nesting で使用される。

real.exe は real data cases についての初期化

wrf.exe は WRF model integration

これらの実行ファイルは main/ から run/, test/em_real へリンクされている。

1.6 WRFV2 の実行

1.6.1 Real.exe

Run real.exe

run/ もしくは test/em_real directory へ移動する（どちらでも良い）。ここでは後者を選択する。

- **cd test/em_real**

ここには以下のファイルがある。

```
lrwxrwxrwx 1 roda user 23 Jul 13 15:11 ETAMPNEW_DATA
-> ../../run/ETAMPNEW_DATA
lrwxrwxrwx 1 roda user 21 Jul 13 15:11 GENPARM.TBL -> ../../run/GENPARM.TBL
lrwxrwxrwx 1 roda user 21 Jul 13 15:11 LANDUSE.TBL -> ../../run/LANDUSE.TBL
lrwxrwxrwx 1 roda user 25 Jul 13 15:11 README.namelist -> ../../run/README.namelist
lrwxrwxrwx 1 roda user 19 Jul 13 15:11 RRTM_DATA -> ../../run/RRTM_DATA
lrwxrwxrwx 1 roda user 22 Jul 13 15:11 SOILPARM.TBL -> ../../run/SOILPARM.TBL
lrwxrwxrwx 1 roda user 21 Jul 13 15:11 VEGPARM.TBL -> ../../run/VEGPARM.TBL
lrwxrwxrwx 1 roda user 21 Jul 13 15:11 gribmap.txt -> ../../run/gribmap.txt
-rw-r--r-- 1 roda user 1762 Feb 19 2005 landFilenames
-rw-r--r-- 1 roda user 5485 Jun 1 17:03 namelist.input
-rwxr-xr-x 1 roda user 5489 Jun 20 2005 namelist.input.jan00
-rwxr-xr-x 1 roda user 5488 Jun 20 2005 namelist.input.jun01
lrwxrwxrwx 1 roda user 20 Jul 13 15:11 ndown.exe -> ../../main/ndown.exe
lrwxrwxrwx 1 roda user 19 Jul 13 15:11 real.exe -> ../../main/real.exe
-rw-r--r-- 1 roda user 10240 Sep 17 2004 run_1way.tar
-rw-r--r-- 1 roda user 10240 Oct 18 2005 run_restart.tar
lrwxrwxrwx 1 roda user 17 Jul 13 15:11 tr49t67 -> ../../run/tr49t67
lrwxrwxrwx 1 roda user 17 Jul 13 15:11 tr49t85 -> ../../run/tr49t85
lrwxrwxrwx 1 roda user 17 Jul 13 15:11 tr67t85 -> ../../run/tr67t85
lrwxrwxrwx 1 roda user 18 Jul 13 15:11 wrf.exe -> ../../main/wrf.exe
```

OnLine Tutorial case について、namelist.input を編集する。
要約すると、以下の点を修正することになる。

Start date: 2005082800
End date: 2005083100
Interval: 21600 sec (6 hours)
Grid points in EW direction: 75
Grid points in SN direction: 70
Number of vertical levels: 31
Grid distance: 30km

同じ namelist.input が real.exe, wrf.exe で用いられる。

Link

WRFSI で作成された wrf_real_input* files を WRFV2/test/em_run/ directory にリンクする。

- **ln –sf ../../../WRF/wrfsi/domains/OnlineTut/siprd/wrf_real_input_em.d01.2005-08-* .**

Run

- **./real.exe**

single processor machine（今回はこれ）の場合は、./real.exe と入力する。
DM(distributed memory) parallel systems の場合、mpirun command が必要になる場合がある。例えば、Linux cluster では、4CPU の MPI code を実行するためのコマンドは

- **mpirun –np 4 ./real.exe**

となる。

Check you output

rsl.out*と rsl.error*を確認する（各 processor について、rsl.out と rsl.error が１つできるはず）。
rsl.out.0000 と rsl.error.0000 は重要な情報を含んでいる。もし実行が失敗していたら、これらのファイルのうちの１つにエラーメッセージがある可能性がある。
real.exe の実行が成功しているならば、rsl.out.0000 の最後に
SUCCESS COMPLETE REAL_EM INIT
が表示される。
全てうまくいけば、以下に示す２つのファイルが作成されている。

境界条件（wrfbdy_d01）と初期条件（wrfinput_d01）は、wrf.exe へ入力するために必要である。
netCDF ncdump command を用いることで、これらのファイルに書かれている output times を確認することもできる。

- **ncdump –v Times wrfbdy_d01**

Times は wrfbdy_d01 ファイルの変数である。
これらのファイルが作成されていれば、WRF ARW model を実行する（wrf.exe）準備が整った。

1.6.2 Wrf.exe

Run wrf.exe

rsl.out*と rsl.error*が run directory にあるならば、wrf.exe でも log file としてこれらのファイルネームを使うため、他のところに移動するか、削除すること。

Edit

namelist.input を OnLine Tutorial case について編集する。

Run

- **PATH=”&PATH”:$HOME/bin;export PATH**
- **./wrf.exe**

single processor machine（今回はこれ）の場合は、./wrf.exe と入力する。
DM(distributed memory) parallel systems の場合、mpirun command が必要になる場合がある。例えば、Linux cluster では、4CPU の MPI code を実行するためのコマンドは

- **mpirun –np 4 ./wrf.exe**

となる。

Check you output

rsl.out*と rsl.error*を確認する（各 processor について、rsl.out と rsl.error が１つできるはず）。

rsl.out.0000 と rsl.error.0000 は重要な情報を含んでいる。もし実行が失敗していたら、これらのファイルのうちの１つにエラーメッセージがある可能性がある。
real.exe の実行が成功しているならば、rsl.out.0000 の最後に
SUCCESS COMPLETE WRF
が表示される。
全てうまくいけば、新しい wrfout_d01 file が作成される。

2005082800～2005083100 まで 25 時間分がこのファイルにできる。ncdump を用いて wrfout_d01 file があることを確認する。

- **ncdump –h wrfout_d01_2005-08-28_00:00:00**

WRF OnLine Tutorial case の実行に成功！！

① この後、自分のケースを試す場合は、WRF/WRFV2 ディレクトリで

- **clean -a**

を行う。
Clean しなければ、以前の情報がそのまま引き継がれてしまうので注意すること！

② 今回はインタラクティブ処理（コマンドの実行）をしていたが、計算量が多くなるとメモリ制限などによりインタラクティブ処理では間に合わなくなる。その場合は、バッチ処理システムを利用する。巻末の付録Ｂを参照すること。

Graphics

現在、４つの graphical package がサポートされている。

- NCL
- RIP4
- WRF-to-Grads
- WRF-to-vis5d

これらのインストール方法や使い方の詳細は、WRF Users Page を参照すること。

私たちは GrADS を使用している。
次の章で、GrADS の使用方法を述べる。

２．GrADS による描画

2.1 WRF2GrADS のインストール・実行

Install wrf2grads

インストール方法は WRF2GrADS のページを参照してください。

ダウンロード済みの Source_Codes の中から、wrf2grads.tar.gz ファイルを/home1/usr2/roda/ディレクトリの元にコピーする。それで、gunzip と tar コマンドで解凍する。

- **gunzip wrf2grads.tar.gz**
- **tar –xvf wrf2grads.tar**

このとき、WRF2GrADS というディレクトリが作られる。

- **cd WRF2GrADS**

WRF2GrADS 内には以下のファイルがある。

Makefile
README
control_file
control_file_height
control_file_pressure
module_wrf_to_grads_netcdf.F
module_wrf_to_grads_util.F
wrf_to_grads.F

Edit Makefile & control_file

WRF2GrADS を使用する前に、**一回 README ファイルを読んでください。**

Makefile からコンピュータに合う環境を編集する。

- **vi Makefile**

例えば、TSUBAME は PGI 環境なので、以下の部分に着目する。

```
……
# linux flags (PGI)

#LIBNETCDF = -L/usr/local/netcdf/lib -lnetcdf -lm
#INCLUDE = -I/usr/local/netcdf/include -I./
#FC = pgf90
#FCFLAGS = -g -C -Mfree
#FCFLAGS = -fast -Mfree
#CPP = /usr/bin/cpp
#CPPFLAGS = -I. -C -traditional -DRECL4
……
```

LIBNETCDF から CPPFLAGS までの前の「#」を削除する。
続いて、赤色部分を編集し、最後に上書き保存する（**Esc : w q** と打つ）。

```
……
# linux flags (PGI)

LIBNETCDF = -L/home1/usr2/roda/netcdf-3.6.2/lib -lnetcdf -lm
INCLUDE = -I/home1/usr2/roda/netcdf-3.6.2/include -I./
FC = pgf90
FCFLAGS = -g -C -Mfree
FCFLAGS = -fast -Mfree
CPP = /usr/bin/cpp
CPPFLAGS = -I. -C -traditional -DRECL4
……
```

- **make**

wrf_to_grads という実行ファイルが作成されれば完了。

コントロールファイルを編集する。

- **vi control_file**

set times to be processed
- set variables to be processed
- define the input file
- specify if the input is real/ideal/static data
- set levels to interpolate too

```
7                    ! number of times to put in GrADS file, negative means ignore the times
2005-08-28_00:00:00   (←出力ファイルの最小時間間隔は 03:00:00)
2005-08-28_12:00:00
2005-08-29_00:00:00
2005-08-29_12:00:00
2005-08-30_00:00:00
2005-08-30_12:00:00
2005-08-31_00:00:00
end_of_time_list
                     ! 3D variable list for GrADS file
                     ! indent one space to skip（１マス空けると出力されない）
……
/DATA/real/wrfinput_d01

/home1/usr2/roda/WRF/WRFV2/test/em_real/wrfout_d01_2005-08-28_00:00:00

 /DATA/b_wave/wrfout_d01_0001-01-01_00:00:00
 /DATA/grav2d_x/wrfout_d01_0001-01-01_00:00:00
 ……
```

Run wrf2grads

- **./wrf_to_grads control_file test -v**（注＞test の部分は自分で好きな名前で良い）

成功したら Gracefull STOP が表示され、test.ctl・test.dat が作成されている。

```
 ........
 writing out variable, time                7              3
 time 2005-08-30_00:00:00, output variable Z
 getting data for HGT
 writing out variable, time                8              3
 time 2005-08-30_00:00:00, output variable HGT


 -------------------------------------
              Gracefull STOP
 -------------------------------------
```

2.2 GrADS のインストール・実行

Install GrADS

ダウンロード済みの Source_Codes の中から grads-src-1.9b4 をダウンロードする。

- **./configure**
- **make**
- **make　install**

ssh モードで、実行する。

Set Environment Variables

環境変数を設定する。

- **setenv NETCDF /home1/usr2/roda/netcdf-3.6.2**
- **setenv GADDIR /home1/usr2/roda/grads-1.9b4/data**
- **setenv GASCRP /home1/usr2/roda/grads-1.9b4**
- **setenv GAUDFT /home1/usr2/roda/grads-1.9b4/data**
- **setenv PATH $PATH":$GADDIR"**

eXodus を開く。

WRF2GrADS ディレクトリ内で、

- **/home/usr/myname/grads-1.9b4/bin/gradsc**

Xwindow が開かれる。
ga> というのが新しいコマンドライン。
ここに GrADS 用のコマンドを打ち、画像を表示させる。
以下に表示の一例を示す。
詳しくは、GrADS マニュアル（英語版）を参照すること。なお、日本語による説明もインターネットで検索するといくつか引っかかるので、そちらも参考にすると良い。巻末の付録Ｂには、よく使用する簡単なコマンドについて記載している。

➢ /home1/usr2/roda/grads-1.9b4/bin/gradsc

```
Grid Analysis and Display System (GrADS) Version 1.9b4
Copyright (c) 1988-2005 by Brian Doty and IGES
Center for Ocean-Land-Atmosphere Studies (COLA)
Institute for Global Environment and Society (IGES)
GrADS comes with ABSOLUTELY NO WARRANTY
See file COPYRIGHT for more information
Config: v1.9b4 32-bit little-endian lats

Issue 'q config' command for more information.

Landscape mode? (no for portrait): y
GX Package Initialization: Size = 11 8.5
ga>open test.ctl
ga>query file
ga>d hgt            標高
ga>c                clean
ga>d u;v            風
ga>d tc             温度
ga>d p              気圧
…
…
```

簡単なコマンドの説明は付録Aを参照。

付録A．GrADSコマンドの簡単な説明

open test.ctl	ファイルを展開する。
query file	利用できるコマンドをリストする
reset	open test.ctl　以下全部リセットする。
reinit	open から入力しなおす。

gs ファイルの実行

もっと便利に使うためには、gs ファイルを予め作ったほうがよい。この gs ファイルは秀丸を利用して編集することができる。gs ファイルの書き方は WRF 2 GrADS のいくつかの例文を参照してください。

ga>　run test.gs

test.gs ファイルは下記の test.gs ファイルの例を参照してください。

よりきれいな図を出すには：

ga>　enable print test.emf	test という emf ファイルを開く。
ga>　xxxxx　（下記参照）	操作
ga>　print	画像を書き込む
ga>　disable	emf ファイルを閉じる

例えば、xxxxx で次の操作ができる。

```
set t 5
set cint 1
set grads off
set grid off
d tc
```

emf ファイルについては、gv32.exe というフリーソフトを使って wmf というファイルに変換する。wmf ファイルから他形式（.gif, .png など）に変換する。

grads を使ってアニメを出力することができない。gif ファイルに変換してから、他ソフト（gif アニメなど）を利用してください。

test.gs ファイルの例：

```
'reinit'
'open gg16w2n2d1.ctl'
'set grads off'
'set gxout shaded'
'set grid off'
```

```
'set mpdset mres'
'set map 1 1 5'
'set display color white'

say 'Create gif images as well (1=yes ; 0=no)'
pull ans
frame = 1

'q file'
rec=sublin(result,5)
_endtime=subwrd(rec,12)

runscript = 1
dis_t = 1

while(runscript)

'set t ' dis_t
'q dims'
rec=sublin(result,5)
_analysis=subwrd(rec,6)
say 'Time is ' _analysis

'c'
'enable print gg16w2n2d1-tsk'frame'.emf'
'set grads off'
'set gxout shaded'
'set cint 2'
'set clevs -8 -6 -4 -2 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32'
'set ccols 58 56 55 54 45 44 43 38 37 36 34 33 32 31 21 22 23 24 25 26 27 29 '
'd tsk-273.16'
'set strsiz .2'
'set string 1 l 6'
'draw string 2.7 7.35    SURFACE SKIN TEMPERATURE (C)'
'set strsiz .15'
'set string 1 l 3'
'draw string 8.4 7.2 ' _analysis
'run cbar.gs'
```

```
if(ans)
'print'
'disable'
frame=frame+1
endif
pull dummy

if ( dis_t=_endtime )
 runscript=0
endif
dis_t = dis_t + 1
endwhile
```

一つずつ入力してみれば、各コマンドの意味がわかるのでやってみてください。

なお、図を整理するときに良く使われるコマンドを以下に記します。

set arrxcl 0.6 20　　風速ベクトル　（0.6 inch 、20m/s を意味する）

set grads off　　grads という名前を図に表さない。

set grid off　　図にグリッドを表さない

set map 1 1 5　　マップの色、線型、線のサイズ

付録Ｂ．バッチ処理

① TSUBAME でバッチ処理システムを利用する。バッチ処理は、投入したジョブがシステムによってスケジューリングされ、他のジョブの影響を受けることなく効率よく計算できる。チュートリアルではインタラクティブ処理（コマンドの実行）をしていたが、計算量が多くなるとメモリ制限などによりインタラクティブ処理では間に合わなくなる。その場合は、バッチ処理システムを利用する。大規模並列（32 並列以上＆メモリ 2GB 以上）の計算を行う場合には課金申請が必要になるので、まずは先生に相談する。詳しくは TSUBAME マニュアルを参照すること。

② バッチ処理の手順は以下の通り。

1．ジョブの作成
2．ジョブの投入
3．ジョブの状態確認
4．ジョブの結果確認

C-1　ジョブの作成

ジョブとは、バッチ処理で実行したいコマンドを記述したシェルスクリプトのこと。
実際に WRFV2 で投入するスクリプトを以下に示す。これは wrf.exe のスクリプトであるが、real.exe でも同様のスクリプトを用意する。
なお、**自分の PC で txt 作成（秀丸など）により.sh を作らずに vi エディタなどで直接ファイルを作成すること！**

```
#!/bin/sh
#
#wrf.exe
#
FILE=./result_`printf "%03d" ${MPIRUN_RANK}`_${HOSTNAME}.log

cd /home1/usr2/roda/WRF/WRFV2/test/em_real
./wrf.exe >& ${FILE}
```

実行時間を計測する場合は、最後の行の先頭に　Time　をつける

```
Time ./wrf.exe >& ${FILE}
```

C-2　ジョブの投入

スクリプトをジョブとしてキューに投入する。投入には n1ge コマンドを使う。
まずはスクリプトに実行権を与え、それから投入する。

- **chmod a+x wrf.sh**
- **n1ge –g 1S060156 –N test_case – q sla1 –mpi 128:8 –mem 4 –rt 180 wrf.sh**

以上は性能保障サービス（sla1 キュー）への投入例である。
課金申請をすると、課金グループに ID number が割り当てられる（ここでは 1S060156）。投入にはこ

の番号が必要。

-g　課金グループの番号

-N　ジョブの名前（自分で適当につける）

-q　キューの種類

-mpi　CPU 数　（CPU 数だけでも良いが、今回は

-mem　各プロセスのメモリサイズ（単位：Gbyte）性能保障サービスでは 1Gbyte までしか使用できないので、それ以上になる場合は-mem で指定する必要がある。

-rt　性能保障サービスの実行時間。指定しない場合は 30 分のみの計算となり、それ以上の計算になる場合は落ちてしまう。

最後に、スクリプト名を入力する。

C-3　ジョブの状態確認

投入したジョブの状態を確認するには qstat コマンドを使用。

- **qstat –u roda**

```
job-ID  prior   name       user    state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
1096938 0.50500 LOGIN      roda    r     03/21/2007     15:39:28 interactive@tgg075001          1
```

ジョブ ID　1096938 のジョブ（LOGIN）が interactive キューで実行（state が r）されている。

- **qjobs -f**

で 2 時間以内に終了したジョブが表示される。

投入したジョブを、終了を待たずに削除する場合は、qdel コマンドでジョブ ID を指定して削除する。

- **qdel 1096938**

C-4　ジョブの結果確認

ジョブが終了すると、そのジョブの実行結果ファイルが得られる。結果ファイルは通常 2 つある。1 つはジョブ実行時に標準出力された内容を格納されたファイルで、もう 1 つは標準エラー出力に出力された内容が格納されたファイルである。

標準出力ファイル名は「ジョブ名.o ジョブ ID」（例；ジョブ名.o1096938）で、標準エラー出力は「ジョブ名.e ジョブ ID」（例；ジョブ名.e1096938）である。ジョブ ID はシステムによって割り当てられる一意の識別子。ファイルの標準的な出力先は、ジョブが実行される現在の作業ディレクトリになる。

＊ バッチ実行予報は、GSIC ホームページで確認できるので、それを参考にどのキューを使用するか決めると良い。

http://spinner.cs.ucsb.edu/batchq/bqcluster.php?resource=tsubame