# ＷＲＦ　マニュアル

(ＷＲＦＶ２__ＷＰＳ用)

＊チュートリアル

第二版　２００９年3月３０日　　修正：小田僚子
初　版　２００７年3月３０日　　文責：小田僚子 ・ 尤　龍涛

# 目次

## ～はじめに～

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_WPS 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！**

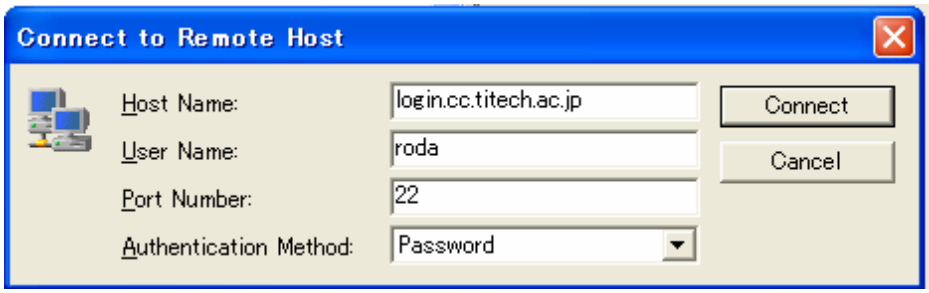
＊ 大岡山スパコンのアカウントを取得する。
（東工大学術国際情報センターの HP から申請書を DL できる。http://www.gsic.titech.ac.jp/）

＊ 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 password:　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。

  （vi エディタ内で **>esc >:q!** 上書きせずに quit。編集して上書きする場合は **>esc >: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、

- **csh**

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。（付録.D 参照）

# ＊PGI コンパイラのバージョンアップ

**～ WRF を正常に動作させるために、TSUBAME の PGI コンパイラをバージョンアップしよう！**

WRF では pgf6-1.3 以上でなければ正常に動作しない。センターでは、PGI7.2-4 が用意されている（2008/11/14 現在）。以下の通り、環境変数を設定する。

- **setenv PGI /usr/apps/isv10/pgi**
- **setenv PATH /usr/apps/isv10/pgi/linux86-64/7.2-4/bin:$PATH**
- **setenv LD_LIBRARY_PATH /usr/apps/isv10/pgi/linux86-64/7.2-4/libso:$LD_LIBRARY_PATH**

＊毎回、上記設定をするのは面倒なので、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。（付録 D 参照）

＊なるべく work にインストールすること。（home にあまり要領の大きなものは置かない）

**********

**ただし**、この PGI7.2-4 は従来の PGU6.1-2 と比較して最適化機能が強化されたために、一時的に使用するメモリ量が増加しているため、interactive node でのメモリ制限により、以下のコンパイルエラーによってコンパイルに失敗する場合がある。

PGF90-F-0007-Subprogram too large to compile at this optimization level (xxxx.f90: xxx)

PGF90/x86-64 Linux 7.2-4: compilation aborted

これに対処するために、コンパイル専用のキューが用意されている。

コンパイル時に、以下の interactive 内の特定ノードにログインする。

- **ssh –Y –t roda@login.cc.titech.ac.jp nolimit　（ここで、roda は各自の USER-ID に変更する）**

これは、通常の interactive node で設けられているメモリ制限がない。

ログイン後、6 時間が経過すると強制終了となる。

コンパイル用のキューなので、プログラムの実行はできないことに注意！！

実行は通常の interactive node などで行うこと。

つまり、コンパイル終了後は、

- **exit**

でコンパイル専用キューから出て、それから通常の作業（実行等）を行うこと。

# 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、atmos の

/work/database/WRF/Source_Codes_and_Graphics_Software

に入れてある。

2006 年 12 月に、WRF ARW model は v2.1.2 から v2.2 に update され、WRFSI から The WRF Preprocessing System (WPS)を用いるようになった。

## 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

  **WPS + 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

  **WPS + 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。

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。

WRFSI

WRFSI は WPS の前バージョン。WPS を用いる場合（今回）は必要ない。
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。

＊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。

## GUI（画像表示）

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。（今のところ使っていないので、ここでは GUI での操作については詳しく触れない。）

- ssh –Y –l roda login.cc.titech.ac.jp
- setenv DISPLAY 172.20.25.24:0.0 （←自分の PC の IP アドレスを打つこと）
- setenv LANG C
- qmon &

画面上に GUI が現れる。

## Let's Get Started

WRF ARW System の標準的な処理の流れは以下の通りである。


しかし、WPS のコンパイルの成功は、ARW モデルの成功による。この理由は、２つのプログラムで WRF I/O API のような共通ルーチンを用いているからである。そのため、ソースコードを入手するとまず WRFV2 をコンパイルし、それから WPS をコンパイルする。

## 1.1 WRFV2・WPS のインストール

### Get Source Code

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

注＞最初にダウンロードする際、登録を要求される（無料）が、次回からは登録した e-mail address を入力するだけでダウンロード可能となる。

ダウンロードするもの

- 最新の WRF ARW tar file
- 最新の WPS tar file

上記のものを自分の working directory (WRF/)に入れる。

### Unpack the Code

WRF ディレクトリ上で、TAR file を開く。

- **cd WRF**
- **gunzip WRFV2.2.TAR.gz**
- **gunzip WPSV2.2.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 WRFV2.2.TAR**
- **tar –xvf WPSV2.2.TAR**

この後、新しく WRFV2/、WPS/ ディレクトリが作成されているはず。

### Examine the Source Code

- **cd WRFV2**

中身を確認すると、下記のような感じになっている。

- **ls -all**

```
-rw-r--r--      1 roda user 18506 Dec 19 04:12 Makefile
-rw-r--r--      1 roda user  8192 Dec 23 02:51 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 Dec 23 02:57 Registry
drwxr-xr-x    2 roda user  4096 Dec 23 02:57 arch
-rwxr-xr-x     1 roda user  1597 Dec 19 04:12 clean
-rwxr-xr-x     1 roda user  9028 Oct 14 02:53 compile
-rwxr-xr-x     1 roda user 12097 Dec 14 22:35 configure
drwxr-xr-x    2 roda user  4096 Dec 23 02:57 dyn_em
```

```
drwxr-xr-x     2 roda user    4096 Dec 23 02:57 dyn_exp
drwxr-xr-x     2 roda user    4096 Dec 23 02:56 dyn_nmm
drwxr-xr-x    15 roda user    4096 Dec 23 02:56 external
drwxr-xr-x     2 roda user    4096 Dec 23 02:56 frame
drwxr-xr-x     2 roda user    4096 Dec 23 02:57 inc
drwxr-xr-x     2 roda user    4096 Dec 23 02:56 main
drwxr-xr-x     2 roda user    4096 Dec 23 02:56 phys
drwxr-xr-x     2 roda user    4096 Dec 23 02:57 run
drwxr-xr-x     2 roda user    4096 Dec 23 02:56 share
drwxr-xr-x    12 roda user    4096 Dec 23 02:56 test
drwxr-xr-x     4 roda user    4096 Dec 23 02:56 tools
```

README ファイルはコードやモデルの setup・run の仕方など有益な情報が書かれている。
ソースコードディレクトリは以下が含まれている。

```
 dyn_em/
 dyn_nmm/
 dyn_exp/
 external/
 frame/
 inc/
 main/
 phys/
 share/
 tools/
```

スクリプトには以下がある。

```
 clean
 compile
 configure
makefile
Registry/
arch/
run/
test/
```

## 1.2 WRFV2 のコンパイル

現在 TSUBAME に搭載されているコンパイラ PGI7.2-4 では、interactive node において使用した場合、コンパイルに失敗する。そこで、コンパイル専用のノードに移動して、コンパイルを行うこと。

＊ 本マニュアル「PGI コンパイラのバージョンアップ」のページ（p.6）を参照すること。

### Configure WRFV2

- cd ../WRF/WRFV2
- ./configure

自分のコンピュータで使用可能なリストが現れる。シングルで動かすかマルチ（並列）で動かすか、また、ネスティングを許すか許さないかが選択されるので、目的にあわせて注意深く選ぶこと。以下は、TSUBAME で現れるリストである。

checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /home1/usr2/roda/netcdf-3.6.2
PHDF5 not set in environment. Will configure WRF for use without.
$JASPERLIB or $JASPERINC not found in environment, configuring to build without grib2 I/O...
---
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)
4. PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher (Single-threaded, RSL, Allows nesting)
5. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler (single-threaded, no nesting)
6. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler (single threaded, allows nesting using RSL without MPI)
7. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler (OpenMP)
8. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler SM-Parallel (OpenMP, allows nesting using RSL without MPI)
9. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort+icc compiler DM-Parallel (RSL, MPICH, allows nesting)
10. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort+gcc compiler DM-Parallel (RSL, MPICH, allows nesting)
11. PC Linux x86_64 (IA64 and Opteron), PathScale 2.1 or higher (Single-threaded, no nesting)
12. PC Linux x86_64 (IA64 and Opteron), PathScale 2.1 or higher DM-Parallel (RSL_LITE, PathScale MPICH, No periodic LBCs)
13. Cray XT3 Catamount/Linux x86_64 (Opteron), PGI 5.2 or higher DM-Parallel (RSL_LITE, MPICH, Allows nesting, Periodic in X only )

Enter selection [1-13] :

今回は、最も基本的（singre-threaded, no nesting）な「1」を選ぶことにする。なお、並列での計算を行う場合は「２」を選ぶ。

- 1

configure.wrf file が作成される。

ここで、TSUBAME の場合、configure.wrf の中のコンパイラーを pgf95 と pgcc に改める。

```
FC    =       mpif90
LD    =       mpif90
CC    =       mpicc -DMPI2_SUPPORT -DFSEEKO64_OK
SCC   =       gcc
SFC   =       pgf90
```

上記の部分の CC と SCC を以下のように書き換える！！

```
CC    =       mpicc -DFSEEKO64_OK
SCC   =       pgcc   -DFSEEKO64_OK
SFC   =       pgf95
```

**ネスティングを行う場合**

ネスティングオプションを持っているかどうか確認すること。たいていは、MPI/RSL/RSL_LITE でネスティングをサポートしている。

TSUBAME の場合、RSL を使用すること。（configure の時に、２を選ぶ）。

## Compile WRFV2

Intel のようないくつかのコンピュータは、以下の環境変数を設定する必要がある。

- setenv WRF_EM_CORE1
- ./compile

以下のようなメッセージが現れる。

```
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_esmf_exp
        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 を動かす場合、以下のように打つ（ideal case ではない）。

- **./compile em_real >& compile.log**

これはしばらく時間がかかるので、ひたすら待つ。（1 時間くらい）
コンパイルが成功したかどうか、compile.log を見て確認する。もし成功していたら、main/ ディレクトリに以下のような実行ファイルが生成される。

-rwxr-xr-x　1 roda user 16873997 Jan　4 17:48 ndown.exe
-rwxr-xr-x　1 roda user 16816379 Jan　4 17:48 nup.exe
-rwxr-xr-x　1 roda user 13490787 Jan　4 17:48 real.exe
-rwxr-xr-x　1 roda user 16416917 Jan　4 17:48 wrf.exe

ここで、
ndown.exe　＞one-way ネスティングで用いられる
nup.exe　　＞WRF-Var で用いられる
real.exe　　＞real case のための WRF 初期設定
wrf.exe　　＞WRF モデル計算
これらの実行ファイルは、main/ ディレクトリから run/ と test/em_real/ ディレクトリ（実際に run するところ）にリンクされている。

**Ideal case**

- **./compile em_b_wave >& compile.log**

real.exe が ideal.exe に変わる以外、ほとんど real case と同じ。
実行ファイルは、run/ と test/em_b_wave/ ディレクトリ（傾圧波の場合）にリンクされている。

もしほかの idealized case に変更したい場合は以下のように打つ。

- **./clean -a**

その後、再度 configure と compile を行う。

## 1.3 WPS のコンパイル

**Configure & Compile WPS**

次のステップは WPS のコンパイルである。
もしも Compile WRF が成功していなく、上記４つの.exe ファイルが生成されていないなら、ここに進むべきではない。

## Examine the WPS Source Code

自分で作った WPS ディレクトリに移動する。

- **cd WPS**

このディレクトリには、以下のようなファイルがある。

```
-rw-r--r--      1 roda user 4786 Dec 22 05:27 README
drwxr-xr-x   2 roda user 4096 Dec 23 06:47 arch
-rwxr-xr-x     1 roda user 1672 Sep    9 09:50 clean
-rwxr-xr-x     1 roda user 3349 Sep 13 02:11 compile
-rwxr-xr-x     1 roda user 4257 Jul 20 04:47 configure
drwxr-xr-x   5 roda user 4096 Dec 23 06:47 geogrid
-rwxr-xr-x     1 roda user 1328 Nov 26 05:29 link_grib.csh
drwxr-xr-x   4 roda user 4096 Dec 23 06:47 metgrid
-rw-r--r--      1 roda user 1101 Dec 23 06:47 namelist.wps
-rw-r--r--      1 roda user 1650 Dec 23 06:47 namelist.wps-all_options
drwxr-xr-x   8 roda user 4096 Dec 23 06:47 test_suite
drwxr-xr-x   4 roda user 4096 Dec 23 06:47 ungrib
drwxr-xr-x   3 roda user 4096 Dec 23 06:47 util
```

README ファイルには、コードやモデルの設定・実行の仕方などの有益な情報が含まれている。
ソースコードディレクトリは以下が含まれている。

```
geogrid/
metgrid/
ungrib
util/
```

スクリプトには以下がある。

```
clean
compile
configure
link_grib.csh
arch/
namelist.wps/
```

## Environment Variavle -NETCDF

NETCDF の環境変数を定義する。既に .cshrc などで定義している場合には再度行わなくて良い。

- **setenv NETCDF /home1/usr2/roda/netcdf-3.6.2**（自分の home ディレクトリ）

## Configure WPS

- **./configure**

自分のコンピュータで使用可能なリストが現れる。シングルで動かすかマルチ（並列）で動かすか、また、ネスティングを許すか許さないかが選択される。以下は、TSUBAME で現れるリストである。

Will use NETCDF in dir: /home1/usr2/roda/netcdf-3.6.2
$JASPERLIB or $JASPERINC not found in environment, configuring to build without grib2 I/O...
---
Please select from among the following supported platforms.

1. PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher, serial, NO GRIB2
2. PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher, serial
3. Cray XT Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher, DM parallel, NO GRIB2
4. PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher, DM parallel, NO GRIB2
5. PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher, DM parallel
6. PC Linux x86_64 (IA64 and Opteron), PathScale compiler 2.1 or higher, serial, NO GRIB2
7. PC Linux x86_64 (IA64 and Opteron), PathScale compiler 2.1 or higher, DM parallel, NO GRIB2
8. PC Linux x86_64, g95 compiler, serial, NO GRIB2
9. PC Linux x86_64, g95 compiler, serial

Enter selection [1-9] :

今回は、最も基本的（singre-threaded, no nesting）な「1」を選ぶことにする。なお、並列での計算を行う場合は「4」を選ぶ。(←WRFv2.2 では 3 だったが、WRFv2.2.1 では 4)

- **1**

configure.wps 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。
中身を確認し、WRFV2 パスが WRF_DIR=../WRFV2 でなければ configure.wps ファイル中の左記の部分を正しいパスに書き直す。
注＞NCAR_Graphics を使う場合、以下の部分を変更すること（NCAR_Graohics については付録 A を参照すること）。必ず WPS をコンパイルする前に変更すること。

```
NCARG_LIBS    =    -L$(NCARG_ROOT)/lib -lncarg -lncarg_gks -lncarg_c ¥
                   -L/usr/X11R6/lib64 -lX11
```

また、環境変数を設定する。

- **setenv NCAR_Graphics /home1/usr2/roda/NCAR_Graphics/ncarg-4.4.1**
- **setenv PATH $NCARG_ROOT/bin:$PATH**
- **setenv MANPATH $NCARG_ROOT/man:$MANPATH**

## Compile WPS

- ./compile >& compile.log

コンパイルが成功したかどうか、compile.log を見て確認する。もし成功していたら、以下のような実行ファイルが生成される。

少し待つが、それほど時間はかからない。5 分くらい。

lrwxrwxrwx 1 roda user 23 Jan 5 16:36 geogrid.exe -> geogrid/src/geogrid.exe

lrwxrwxrwx 1 roda user 23 Jan 5 16:36 metgrid.exe -> metgrid/src/metgrid.exe

lrwxrwxrwx 1 roda user 23 Jan 5 16:36 ungrib.exe -> ungrib/src/ungrib.exe

ここで、

geogrid.exe ＞静的データを生成する

metgrid.exe ＞WRFV2 への入力データを生成する

ungrid.exe ＞GRIB データを解凍する

いくつかの utilities が util/ ディレクトリにリンクされている。

lrwxrwxrwx 1 roda user 16 Jan 5 16:37 avg_tsfc.exe -> src/avg_tsfc.exe

lrwxrwxrwx 1 roda user 25 Jan 5 16:36 g1print.exe -> ../ungrib/src/g1print.exe

lrwxrwxrwx 1 roda user 25 Jan 5 16:36 g2print.exe -> ../ungrib/src/g2print.exe

lrwxrwxrwx 1 roda user 16 Jan 5 16:37 mod_levs.exe -> src/mod_levs.exe

lrwxrwxrwx 1 roda user 23 Jan 5 16:36 rd_intermediate.exe -> src/rd_intermediate.exe

ここで

avg_tsfc.exe ＞intermediate file から日平均表面温度を計算する。WRF において、5-layer soil model（sf_surface_physics=1）を用いることを推奨する。

g1print.exe ＞GRIB1 ファイルの内容 List

g2print.exe ＞GRIB2 ファイルの内容 List

mod_levs.exe ＞intermediate file における 3-d files から過剰な level を取り除く。

plotfmt.exe ＞intermediate file をプロットする。(NCAR Graphics に依存する －もしこれがなければ、plotfmt.exe は正確に compile できないだろう。)

plodgrids.exe ＞domain graphics を作成する。geogrid.exe を実行する前に領域を構成する優れたツールである。

rd_intermediate.exe ＞intermediate file を読む。

＊ Utilities についての詳細な情報は Users' Guide の Chapter 3 に書かれている。

＊ NCAR Graphics を自分のシステムに組み込んでいないならば問題ではないが、これはとても便利で自由に使えるものなので、これをインストールすることをお勧めする。なお、インストールしている場合は、下記のとおり、実際に計算を行う前に領域を確認することができる。

＊ 2008/11/14 現在、

NCARG_LIBS = -L$(NCARG_ROOT)/lib -lncarg -lncarg_gks -lncarg_c ¥

-L/usr/X11R6/**lib64** -lX11

を変更しても、plotgrids.exe は作成されない（原因は解明できていない。ただ、ここはシミュレーションの上で重要ではないので、大きな問題はない)。

**領域の確認**

- ./util/plotgrids.exe

gmeta というファイルが生成されているはず。
作成した領域を見る。(x-window を使えるようにしておくこと)

- idt gmeta

右図のような絵が表示されるはず。（なんか、idt コマンド
使えないので、以下のコマンドを打つ）

- ctrans -d ps.mono gmeta > gmeta.ps (モノクロ)
- ctrans -d ps.color gmeta > gmeta.ps (カラー)

- convert -trim gmeta.ps gmeta.png

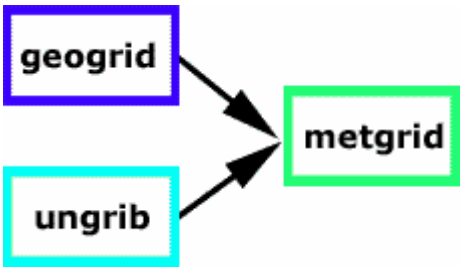
（ps 形式から png 形式に変換）

ここまでできたら、WPS と WRFV2 を実行する準備が整った。

## 1.4 WPS の実行

### Running WPS

WPS は WRFV2 で使われる入力データを作成するために用いられる。




geogrid と ungrib はいくつかの場所で実行されうる。
geogrid は陸生データ（静的）を生成する。
ungrib は GRIB 気象データを解凍し、intermediate file format に入れ込む。
metgrid は水平方向の測定データを自分のモデル領域に補間する。
metgrid からの出力は WRFV2 への入力として用いられる。

### Get Terrestrial Input Data

以下のページから陸生データを入手する。

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

解像度の異なる陸生データも入手しといたほうが良い。（上記ページにリンクがある）
自分でデータ保存用のフォルダを作成し、解凍して保存しておく。
/home1/usr2/roda/WRF/WPS/WPS_Geog/　←私はここ。

### 1.4.1 Geogrid.exe

**Run Geogrid.exe**

namelist.wps file を例として、以下のように書き換える（これは今回のモデル領域に対しての例)。Geogrid は領域を設定するだけのものなので、領域以外の項目（計算開始時間など）はここでは関係ない。

max_dom = 1,
e_we = 75,
e_sn = 70,
map_proj = 'mercator',
ref_lat = 25.00,
ref_lon = －85.00,
truelat1 = 0.0,
truelat2 = 0.0,
stand_lon = －85.00,
geog_data_path = /home1/usr2/roda/WRF/WPS/WPS_Geog/geog （←自分で WPS_GEOG データを置いたところに設定する）

上記のように namelist.wps を編集した後、まずこれから自分が作成しようとする領域かどうかを確認できる。これをするためには、NCAR Graphics を自分のシステムにインストールしておく必要がある。領域の確認の方法は付録に記す。

静的データを作成するために、上記の領域に対する geogrid.exe を動かす。

- **./geogrid.exe**

DM（distributed memory）parallel system で動かす場合は、mpirun コマンドが必要である。4 並列する場合は以下のようになる。

- **mpirun –np 4 geogrid.exe**

実行している間、以下のような表示がされ、”Successful completion of geogrid”が最後に表示される。

Parsed 11 entries in GEOGRID.TBL
Processing domain 1 of 1
Processing XLAT and XLONG
Processing MAPFAC
Processing F and E
Processing ROTANG
Processing LANDUSEF
Calculating landmask from LANDUSEF (WATER = 16)
Processing HGT_M
Processing HGT_U
Processing HGT_V

Processing SOILTEMP
Processing SOILCTOP
Processing SOILCAT
Processing SOILCBOT
Processing ALBEDO12M
Processing GREENFRAC
Processing SNOALB
Processing SLOPECAT
Processing SLOPECAT
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Successful completion of geogrid. !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
静的データが生成されているか、確認する。
-rw-r--r-- 1 roda user 2133896 Jan 10 16:21 geo_em.d01.nc

nudump utility を使って、このファイルの内容を確認できる。

- **ncdump -h geo_em.d01.nc**

このファイルは support している graphical tool を使っても見ることができる。

### 1.4.2 Ungrib.exe

**Run Ungrib.exe**

以下のように namelist.wps を書き換える（これまで既に 72 時間分のデータを入手している（Katrina）が、ここでは 24 時間だけ実行する。より長い時間実行することもできる。）

start_date = '2005-08-28_00:00:00',
end_date = '2005-08-29_00:00:00',
interval_seconds = 21600

*Download* ＞これを GRIB1 データに入力する。また、異なるディレクトリにテストデータを置く。（各ケースごとにそれぞれ異なるディレクトリを作って保存しておくことを推奨する）。今回は、
/home1/usr2/roda/WRF/DATA/
に、解凍していないデータを入れる。

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

このデータは、global GPS/AVN GRIB1 データで、6 時間毎に得られている（2005082800 から 2005083100）。

＊ input data を使用する前に確認することをお勧めする。WPS/util ディレクトリにある g1print.exe と g2grid.exe は、この作業を行うのに適した tool である。他の便利な tool”wgrib”は以下にある。

http://www.cpc.ncep.noaa.gov/products/wesley/wgrib.html

このデータを link_grib.csh を使って link する。

- **./link_grib.csh /home1/usr2/roda/WRF/DATA/avn_050828**

＊ link_grib.csh はスクリプトであり、UNIX コマンドの ln と似ているが、同じフォーマットではない。

- link_grib.csh root_name_of_data_files

  e.g. ./link_grib.csh ../DATA/avn

- ln all_files_to_be_linked link_to

  e.g. ln ../DATA/avn*

以下の link が生成されているはず。

```
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAA -> /home1/usr2/roda/WRF/DATA/avn_050828_00_00
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAB -> /home1/usr2/roda/WRF/DATA/avn_050828_00_06
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAC -> /home1/usr2/roda/WRF/DATA/avn_050828_00_12
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAD -> /home1/usr2/roda/WRF/DATA/avn_050828_00_18
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAE -> /home1/usr2/roda/WRF/DATA/avn_050828_00_24
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAF -> /home1/usr2/roda/WRF/DATA/avn_050828_00_30
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAG -> /home1/usr2/roda/WRF/DATA/avn_050828_00_36
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAH -> /home1/usr2/roda/WRF/DATA/avn_050828_00_42
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAI -> /home1/usr2/roda/WRF/DATA/avn_050828_00_48
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAJ -> /home1/usr2/roda/WRF/DATA/avn_050828_00_54
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAK -> /home1/usr2/roda/WRF/DATA/avn_050828_00_60
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAL -> /home1/usr2/roda/WRF/DATA/avn_050828_00_66
lrwxrwxrwx  1 roda user  42 Jan 12 13:02 GRIBFILE.AAM -> /home1/usr2/roda/WRF/DATA/avn_050828_00_72
```

Vtable に link する（今回のデータは GFS/AVN データなので、GFS Vtable を使う）。

- **ln –sf ungrib/Variable_Tables/Vtable.GFS Vtable**

（提供しているほかの Vtable を確認するには、/ungrib/Variable_Vtables を確認すること）

intermediate を生成するために、ungrid を実行する。

- **./ungrib.exe >&** ungrib.log （注＞並列でも mpirun ではなく左のように実行する）

以下のファイルが生成される。

```
-rw-r--r--  1 roda user 38869928 Jan 12 13:52 FILE:2005-08-28_00
-rw-r--r--  1 roda user 38869928 Jan 12 13:52 FILE:2005-08-28_06
-rw-r--r--  1 roda user 38869928 Jan 12 13:52 FILE:2005-08-28_12
-rw-r--r--  1 roda user 38869928 Jan 12 13:52 FILE:2005-08-28_18
-rw-r--r--  1 roda user 38869928 Jan 12 13:52 FILE:2005-08-29_00
```

＊ 実行している最中に作られる ungrib.log は、input file にかかわる field に関する重要な情報を含んでいる。

これらのファイルの中にはどのような情報が入っているかを見るには２つの方法がある。

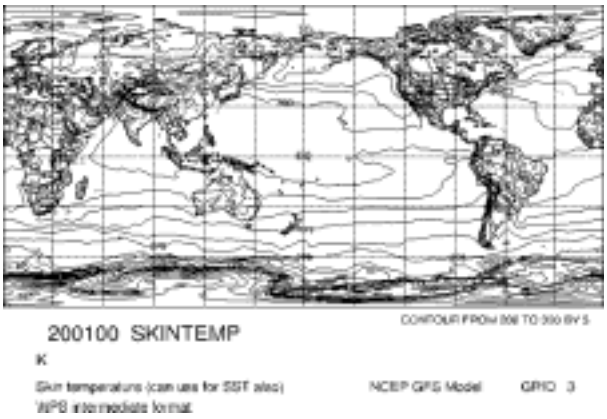
1. util/rd_intermadiate.exe を使ってファイルを実行する。データについての情報（txt 形式）が生成される。

- **./util/rd_intermediate.exe FILE:2005-08-28_00**

2. NCAR_Graphics が使える環境（つまり、util/plotfmt.exe がある）ならば、このコマンドを使って intermediate files を実行することができる。Output は gmeta file で、ctrans コマンドを使って表示できる。

- **./util/plotfmt.exe FILE:2005-08-28_00**

右図は、いくつか生成される図のうちのひとつ。




### 1.4.3 Metgrid.exe

**Run Metgrid.exe**

WPS の最終 step に進むことが出来る。

- **./metgrid.exe**
- **mpirun –np 4 metgrid.exe** （並列の場合）

```
Processing domain 1 of 1.
Processing 2005-08-28_00
      FILE
  Processing 2005-08-28_06
      FILE
  Processing 2005-08-28_12
      FILE
  Processing 2005-08-28_18
      FILE
  Processing 2005-08-29_00
      FILE
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!    Successful completion of metgrid.    !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

以下のファイルが生成される。

```
-rw-r--r--   1 roda user   5932836 Jan 12 19:01 met_em.d01.2005-08-28_00:00:00.nc
-rw-r--r--   1 roda user   5932836 Jan 12 19:01 met_em.d01.2005-08-28_06:00:00.nc
-rw-r--r--   1 roda user   5932836 Jan 12 19:01 met_em.d01.2005-08-28_12:00:00.nc
-rw-r--r--   1 roda user   5932836 Jan 12 19:01 met_em.d01.2005-08-28_18:00:00.nc
-rw-r--r--   1 roda user   5932836 Jan 12 19:01 met_em.d01.2005-08-29_00:00:00.nc
```

ncdump を使って、これらのファイルの内容を確認できる。

- **ncdump –h met_em.d01.2005-08-28_00:00:00.nc**

また、GrADS を使って描画できる。（…と user's guide には書いてあるが、この.nc ファイルは地図情

報を持っていないために、gradsnc を使っても描画できなかった）

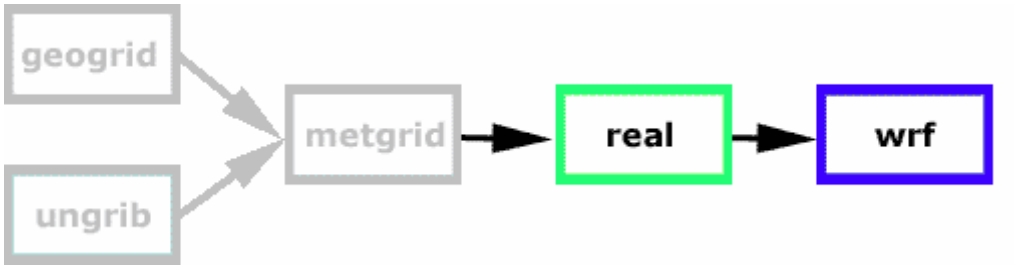
- /home1/usr2/roda/grads-1.9b4/bin/gradsnc
- sdfopen met_em.d01.2005-08-28_00:00:00.nc

**ここまで出来たら、WRFV2 を動かす準備が完了！！**

## 1.5 WRFV2 の実行

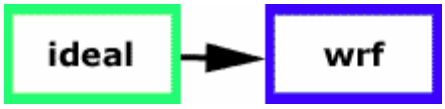
**Running WRFV2**

real case を実行するためのフローを下に示す。




“real” は WRF モデルにインプットする初期値および境界値を生成する。

ideal case を実行するためのフローは以下の通りである。




“ideal” は WRF モデルにインプットする初期値および境界値を生成する。
“ideal” は先のプログラム（real のフローで書かれているグレーのボックス部）に依存しない。

### 1.5.1 Run.exe

**Run Real.exe**

WRFV2 ディレクトリに戻り、run/もしくは test/em_real/ディレクトリに移動する。
ここでは、test/em_real を選択する。

- cd test/em_real

このディレクトリには、以下のファイルが入っているはず。

```
lrwxrwxrwx   1 roda user     22 Jan   4 17:49 CAM_ABS_DATA -> ../../run/CAM_ABS_DATA
lrwxrwxrwx   1 roda user     25 Jan   4 17:49 CAM_AEROPT_DATA -> ../../run/CAM_AEROPT_DATA
lrwxrwxrwx   1 roda user     23 Jan   4 17:49 ETAMPNEW_DATA -> ../../run/ETAMPNEW_DATA
lrwxrwxrwx   1 roda user     21 Jan   4 17:49 GENPARM.TBL -> ../../run/GENPARM.TBL
lrwxrwxrwx   1 roda user     21 Jan   4 17:49 LANDUSE.TBL -> ../../run/LANDUSE.TBL
-rw-r--r--   1 roda user   3227 Dec 20 10:25 README.grid_fdda
lrwxrwxrwx   1 roda user     25 Jan   4 17:49 README.namelist -> ../../run/README.namelist
-rw-r--r--   1 roda user   5166 Dec 20 10:25 README.obs_fdda
```

```
lrwxrwxrwx   1 roda user     19 Jan   4 17:49 RRTM_DATA -> ../../run/RRTM_DATA
lrwxrwxrwx   1 roda user     22 Jan   4 17:49 SOILPARM.TBL -> ../../run/SOILPARM.TBL
lrwxrwxrwx   1 roda user     21 Jan   4 17:49 VEGPARM.TBL -> ../../run/VEGPARM.TBL
lrwxrwxrwx   1 roda user     22 Jan   4 17:49 grib2map.tbl -> ../../run/grib2map.tbl
lrwxrwxrwx   1 roda user     21 Jan   4 17:49 gribmap.txt -> ../../run/gribmap.txt
-rw-r--r--   1 roda user   1762 Feb 19   2005 landFilenames
-rwxr-xr-x   1 roda user   4663 Dec 19 04:13 namelist.input
-rw-r--r--   1 roda user   5647 Dec 15 06:28 namelist.input.chem
-rwxr-xr-x   1 roda user   5732 Dec 19 04:13 namelist.input.grid_fdda
-rwxr-xr-x   1 roda user   5902 Dec 19 04:13 namelist.input.jan00
-rwxr-xr-x   1 roda user   5901 Dec 15 06:28 namelist.input.jun01
-rwxr-xr-x   1 roda user   5655 Dec 19 04:13 namelist.input.obs_fdda
-rwxr-xr-x   1 roda user   4804 Dec 19 04:13 namelist.input.si
-rwxr-xr-x   1 roda user   6108 Dec 19 04:13 namelist.input.wps
lrwxrwxrwx   1 roda user     20 Jan   4 17:49 ndown.exe -> ../../main/ndown.exe
lrwxrwxrwx   1 roda user     18 Jan   4 17:49 nup.exe -> ../../main/nup.exe
lrwxrwxrwx   1 roda user     25 Jan   4 17:49 ozone.formatted -> ../../run/ozone.formatted
lrwxrwxrwx   1 roda user     29 Jan   4 17:49 ozone_lat.formatted -> ../../run/ozone_lat.formatted
lrwxrwxrwx   1 roda user     30 Jan   4 17:49 ozone_plev.formatted -> ../../run/ozone_plev.formatted
lrwxrwxrwx   1 roda user     19 Jan   4 17:49 real.exe -> ../../main/real.exe
-rw-r--r--   1 roda user  10240 Dec   9 16:40 run_1way.tar
-rw-r--r--   1 roda user  20480 Jan 20   2006 run_2way.tar
-rw-r--r--   1 roda user  10240 Jan 20   2006 run_restart.tar
lrwxrwxrwx   1 roda user     17 Jan   4 17:49 tr49t67 -> ../../run/tr49t67
lrwxrwxrwx   1 roda user     17 Jan   4 17:49 tr49t85 -> ../../run/tr49t85
lrwxrwxrwx   1 roda user     17 Jan   4 17:49 tr67t85 -> ../../run/tr67t85
lrwxrwxrwx   1 roda user     25 Jan   4 17:49 urban_param.tbl -> ../../run/urban_param.tbl
lrwxrwxrwx   1 roda user     18 Jan   4 17:49 wrf.exe -> ../../main/wrf.exe
```

**namelist.input**

以下のように編集する。

```
run_days = 0,
run_hours = 0,
start_year = 2005,
start_month = 08,
start_day = 28,
start_hour = 00,
end_year = 2005,
end_month = 08,
end_day = 29,
end_hour = 00,
interval_seconds = 21600
history_interval = 180,
time_step = 180,
max_dom = 1,
s_we = 1,
e_we = 75,
s_sn = 1,
e_sn = 70,
```

s_vert = 1,　　dx = 30000,

e_vert = 28,　　dy = 30000,

num_metgrid_levels = 27

＊ README.namelist には namelist 変数のついての説明が書かれている。

＊ namelist.input は real.exe、wrf.exe の両方で使われる。

以下に、em_real ディレクトリ内にあるファイルの説明を記す。

| | |
|---|---|
| ETAMPNEW_DATA | Variables for the Ferrier (new Eta) microphysics scheme, (binary file). |
| CAM_ABS_DATA<br>CAM_AEROPT_DATA<br>ozone.formatted<br>ozone_lat.formatted<br>ozone_plev.formatted | CAM radiation variables (binary files).<br>Details available in **phys/module_ra_cam.F** |
| RRTM_DATA | Data for the 16 longwave spectral bands used in RRTM (binary file).<br>Details available in **phys/module_ra_rrtm.F** |
| tr49t67<br>tr49t85<br>tr67t85 | Data for the GFDL radiation scheme (binary files).<br>Details available in **phys/module_ra_gfdleta.F** |
| LANDUSE.TBL | Land surface parameters required by the LSM models. The file is an ASCII table.<br>Details available in *phys/module_physics_init.F* |
| GENPARM.TBL<br>SOILPARM.TBL<br>VEGPARM.TBL<br>urban_param.tbl | Soil and vegetation parameters required by the Noah LSM model. The files are ASCII tables.<br>Details available in **phys/module_sf_noahlsm.F** and **phys/module_sf_urban.F** |
| grib2map.tbl<br>gribmap.txt | Tables required for creating GRIB1 and GRIB2 output. |
| namelist.input<br>namelist.input.jan00<br>namelist.input.jun01<br>namelist.input.wps<br>namelist.input.SI | Sample namelists to set up and run real.exe and wrf.exe<br>namelist.input will always be used by real.exe and wrf.exe<br>.jan00 and .jun01 have been set up for our default test cases<br>.wps and .SI are samples of different setups required depending on the preprocesssor used.<br>Details of all the namelist options are available in the README.namelist file. |

**WRFSI Users**

まだ WRFSI を使っている場合は、以下のラインを、namelist_input の time_control section（namelist.input.SI を見る）に付け加える。

＞auxinput1_inname = “wrf_real_input_em.d<domain>.<date>”

**Vertical Levels for your case**

WPS ソフトウェアの導入部までに、鉛直補間は real.exe によって形成される。鉛直レベルは以下の namelist パラメータによって支配されている。

**e_vert ; eta_levels**

最低でも e_vert を設定すること。もしこの変数のみ設定したら、コードは下端と上端において薄い層の鉛直レベルを自動的に設定する。これは全ての領域で設定されなければならないが、全ての領域で同じでなければならない。

e.g. e_vert = 28, 28

あるいは e_vert と eta_levels の両方を設定することによって、eta levels を指定することができる。eta_levels は 1.0～0.0 の間でなければならず、e_vert の数と対応していなければならない。

e.g. e_vert = 28, 28
eta_levels = 1.000, 0.990, 0.978, 0.964, 0.946,
0.922, 0.894, 0.860, 0.817, 0.766,
0.707, 0.644, 0.576, 0.507, 0.444,
0.380, 0.324, 0.273, 0.228, 0.188,
0.152, 0.121, 0.093, 0.069, 0.048,
0.029, 0.014, 0.000

＊ WRFSI User はモデルの上端を設定することに関し、SI input data に影響を与えない。次を参照のこと＞ *SI interpolation section*

**Link**

WPS で生成された met_em_d01*にリンクする。

- **ln –sf /home1/usr2/roda/WRF/WPS/met_em.d01.2005-08-* .**（注＞*と.の間にスペースあり！）
- **ln –sf /home1/usr2/roda/WRF/WPS/met_em.d02.2005-08-* .**

＊ WRFSI User は WRFSI に生成されている wrf_real_input_em.d01*をリンクすること。

**Run Real.exe**

シングルで動かす場合は、以下を打つだけでよい。

- **./real.exe**

並列で動かす場合は、以下のように打つ。

- **mpirun –np 4 real.exe**

**Check you output**

正しく実行されたかを確認する。
最後に ***SUCCESS COMPLETE REAL_EM INIT*** と表示されていれば成功している。

### 1.5.2 Wrf.exe

**MPI runs**

rsl.out*と rsl.error*の log file を確認する（並列の数だけこれらの log file が作られる）。
通常 rsl.out.0000 と rsl.error.0000 にたいていの情報が含まれているが、もし実行が失敗していたら、エラーメッセージはその他のファイルのどこかにあるかもしれない。log file の最後に ***SUCCESS COMPLETE REAL_EM INIT*** と表示されていれば成功している。

実行がうまくいっていたら、以下のファイルが生成される。
-rw-r--r-- 1 roda user 10056744 Jan 12 20:25 wrfbdy_d01
-rw-r--r-- 1 roda user 7076584 Jan 12 20:25 wrfinput_d01
これらは境界条件（wrfbdy_d01）と初期条件（wrfinput_d01）であり、wrf.exe への入力に必要である。
netCDF の ncdump コマンドでこれらのファイルに書かれる出力時間を確認できる。

- **ncdump –v Times wrfbdy_d01**

ここで、Times は wrfbdy_d01 ファイルの中の変数である。

これらのファイルが生成されていれば、ARW WRF モデル（wrf.exe）に進む準備が整った。

**Run Wrf.exe**

**MPI runs**

rsl.out*と rsl.error*ファイルがまだ run ディレクトリにあるなら、それをどこか違う場所に移すか削除する。wrf.exe も同じく rsl.out*と rsl.error*というファイルを作成する。
削除する場合は以下のように打つ。

- **rm rsl.***

（上記のように打つと、削除するか確認しないで消してしまうので、慣れないうちは rm –i rsl.*と打ち、削除しても良いかの確認文を出力させるようにした方が賢明）

このテストランではデフォルトの physical options を用いることにするので、namelist.input を再度編集しなくても良い。もし physical options を変更したいなら、この段階で namelist.input を編集する。
＊ 異なる surface scheme を用いて実行する場合、これまで用いてきた surface scheme に依存する input data が real.exe によって生成されているため、real.exe を再度実行する必要がある。他の physical schemes についてはこの必要はない。

**Run Wrf.exe**

シングルで動かす場合は、以下を打つだけでよい。

- **./wrf.exe**

並列で動かす場合は、以下のように打つ。

- **mpirun –np 4 wrf.exe**

**Check you output**

正しく実行されたかを、log file で確認する。

最後に

***wrf：SUCCESS COMPLETE WRF*** と表示されていれば成功している。

real.exe が正しく出来ていれば、この表記が見られるはずである。

（d01 2005-08-29_00:00:00 wrf: SUCCESS COMPLETE WRF）

**MPI runs**

rsl.out*と rsl.error*の log file を確認する（並列の数だけこれらの log file が作られる）。

通常 rsl.out.0000 と rsl.error.0000 にたいていの情報が含まれているが、もし実行が失敗していたら、エラーメッセージはその他のファイルのどこかにあるかもしれない。

実行がうまくいっていたら、以下のファイルが生成される。

-rw-r--r--　1 roda user 68758172 Jan 14 12:17 wrfout_d01_2005-08-28_00:00:00

2005082800 から 2005082900 までの 24 時間分がこのファイルにあるはず。wrfout_d01 ファイルに何が書かれているかも簡単に確認するためには、ncdump を用いる。

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

***Congratulations☆　WRF モデルのテストラン終了！！***

## 1.6 その他

① この後、自分のケースを試す場合は、WRF/WRFV2 ディレクトリ&WRF/WPS 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 C を参照すること。

# 2．GrADS による描画　(現在は ARW-POST に変更)

GrADS は、2008 年 2 月現在、ARW-POST という描画ソフトに変更になっている。GrADS（ARW-POST）以外にもいくつかの描画ソフトが推奨されているので、自分で判断して好きなものを使ってください。以下、参考までに、GrADS のインストール方法などを掲載します。

## 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 : wq** 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、

- **/home1/usr2/roda/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
…
…
```

# 付録A．NCAR Graphics のインストール

詳細は http://ngwww.ucar.edu/ng/installsrc.html#WhatSystems を参考にすること。
ホームディレクトリに NCAR Graphics のソースコードをダウンロードし、解凍する。

- **gunzip ncarg-4.4.1.src.tar.gz**
- **tar –xvf ncarg-4.4.1.src.tar**

ncarg-4.4.1 というディレクトリが作成される。ここで、環境変数を設定する。

- **setenv NCARG /home1/usr2/roda/NCAR_Graphics/ncarg-4.4.1**
- **setenv FC pgf95**
- **cd ncarg-4.4.1**
- **./Configure –v**

システムに対応していなければ、エラーメッセージが出てすぐに quit してしまう。そうでなければ、いくつかの質問に答える形で進んでいく。
環境変数の設定をするところで、default では /usr/local/ncarg/lib などになっているが、そのようなところを自分のところ（/home1/usr2/roda/NCAR_Graphics/ncarg-4.4.1/lib など。以下参照）に変更する。また X11 の設定では、TSUBAME では/usr/X11R6/lib64 にする。また、HDF のオプションをつけるかを聞かれるところでは no にする。それ以外はそのままで良い。

＞/home1/usr2/roda/NCAR_Graphics/ncarg-4.4.1/bin
＞/home1/usr2/roda/NCAR_Graphics/ncarg-4.4.1/lib
＞/home1/usr2/roda/NCAR_Graphics/ncarg-4.4.1/include
＞/home1/usr2/roda/NCAR_Graphics/ncarg-4.4.1/man
＞/home1/usr2/roda/NCAR_Graphics/ncarg-4.4.1/tmp
＞/usr/X11R6/lib64
＞/usr/X11R6/include

- **make Everything >& make-output &**

バックグラウンドで実行しているが、フォアグラウンドで見てみたければ以下のようにする。

- **tail –f make-output**

エラーが出て止まるようならば、Restarting the installation を参考に、reinstall する。
ここまで出来たら、環境変数の設定をする。

- **setenv NCARG_ROOT /home1/usr2/roda/NCAR_Graphics/ncarg-4.4.1**
- **setenv PATH $NCARG_ROOT/bin:$PATH**
- **setenv MANPATH $NCARG_ROOT/man:$MANPATH**

次に、installation のテストをする。
NCAR Graphics Fundamentals を参考にするのが良いが、簡単にテストするためには以下のことを行う。

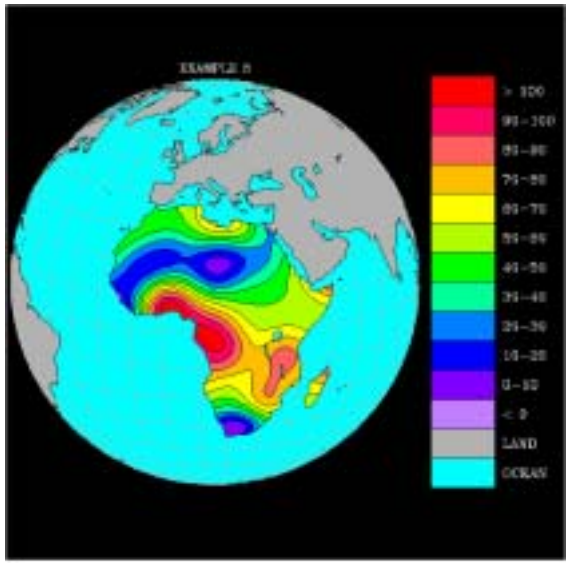
- **ncargex cpex08**

NCAR Graphics Fortran　Example <cpex08>
Copying cpex08.f
Copying cpexcc.f

```
Compiling and linking...
pgf95 -O -o cpex08 cpexcc.f cpex08.f -L/home1/usr2/roda/NCAR_Graphics/ncarg-4.4.1/lib
-L/usr/X11R6/lib64 -lncarg -lncarg_gks -lncarg_c -lX11 -lXext
cpexcc.f:
cpex08.f:
Executing <cpex08>...
PLOT TITLE WAS EXAMPLE 8
INTEGER WORKSPACE USED                120
    REAL WORKSPACE USED               400
    AREA MAP SPACE USED             93250
FORTRAN STOP
Metafile file is named cpex08.ngcm.
```

- **ctrans –d X11 cpex08.ngcm**

最初のコマンドは cpex08.f というファイルを自分のワーキングディレクトリにコピーし、それをコンパイル、リンク、実行し、cpex08.ngcm というグラフィックファイルを生成するものである。２つ目のコマンドは X11 スクリーンに画像を表示するものである。X ウィンドウを使うので、まず DISPLAY の設定（GUI（画像表示）参照）を行っておくこと。
右図のような画像が表示される。なお、何も表示されず黒い画像が表示される場合は、画面をクリックすると図が表示される。



③ 直接 ps ファイルにするには以下のようにする。

- **ncargex -W ps cpex08**

cpex08.ps という名前のファイルが作成される（右図のような背景黒・文字白の図）
また、以下のようにすると、ncgm を介して ps が作成される。

- **ctrans -d ps.color cpex08.ngcm | psblack > cpex08.ps**　（背景黒・文字白）
- **ctrans -d ps.color cpex08.ngcm | pswhite > cpex08.ps**　（背景白・文字黒）

④ ps から png などに変換するには、以下のようにする。

- **convert –trim cpex08.ps cpex08.png**

# 付録Ｂ．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。 |

**Run gs file**

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 test.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
```

シングルクォーテーション（‘ ’）内のコマンドを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
set mpdset mres　国境を表す
draw title *TEST*　*TEST* というタイトルをつける

# 付録Ｃ．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 4S070143 –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 4S070143）。投入にはこ

の番号が必要。番号は、年度毎（申請毎）に変わるので注意！

-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

# 付録Ｄ．.cshrc の作成

ホームディレクトリの下に、.cshrc（bash を使う場合は.bashrc）を設定しておくと、いちいち環境変数をコマンド上で設定しなくても、シェルを起動する度に自動的に読み込んでくれるので便利。
ここでは、WRF で使用している環境変数をまとめた設定ファイル「.cshrc」を示す。以下に示すのは、このマニュアル内で出て来た環境変数をまとめたものである。bash を使用している場合は、環境変数の設定の仕方が違うので、bash 用に書き換えて作成すること。（csh では setenv であるのに対し、bash では export となる）
ホームディレクトリの下に、もともと.cshrc（.bashrc）が存在する場合は、それに書き加えれば良い。作成されていない場合は、自分で.cshrc（.bashrc）という名前のファイルを作成する。
なお、.cshrc（.bashrc）は隠しファイルなので、単に「ls」としても確認できない。「ls –all」などとすれば確認できる。

```
setenv NETCDF /home1/usr2/roda/netcdf-3.6.2
setenv WRF_EM_CORE1
setenv NCARG_ROOT /home1/usr2/roda/NCAR_Graphics/ncarg-4.4.1-2
setenv PATH $NCARG_ROOT/bin:$PATH
setenv MANPATH $NCARG_ROOT/man:$MANPATH
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"
setenv PGI /work/roda/pgi
setenv PATH /work/roda/pgi/linux86-64/6.2/bin:$PATH
setenv LD_LIBRARY_PATH /work/roda/pgi/linux86-64/6.2/libso:$LD_LIBRARY_PATH

alias oda "setenv DISPLAY 172.20.25.24:0.0"
alias gradsc /home1/usr2/roda/grads-1.9b4/bin/gradsc
alias gradsnc /home1/usr2/roda/grads-1.9b4/bin/gradsnc
alias WPS "cd /home1/usr2/roda/WRF/WPS"
alias WRFV2 "cd /home1/usr2/roda/WRF/WRFV2"
alias em_real "cd /home1/usr2/roda/WRF/WRFV2/test/em_real"
```

＊ alias とは、「別名」という意味の英語で、「alias 別名　正式名」のように設定すれば、別名を入力しただけで正式名の作業をしてくれるというもの。
上の例では、「WPS」と入力すると、「cd /home1/usr2/roda/WRF/WPS」のように働く。