

WRF/WPS ユーザーガイド

はじめに

これは WRF/WPS という気象解析予報モデルプログラム及びその初期化プログラムに関して、色々なトラブルに見舞われつつインストール&使用していった時の覚え書きです。OS によりけりなインストールの覚え書きですが、まあソースからのコンパイル&インストールがほとんどなので、読み替えればだいたい UNIX 系 OS なら大丈夫かと。

現在の利用環境は以下の通り。

- * Fedora 10
- * Intel C/C++ Compiler 11.0
- * Intel Fortran Compiler 11.0

注意

このページでは、コマンドラインで実行するコード、ファイルに記述するコードなどが書かれています。コマンドラインで実行するコードについては、#で始まる場合はルート権限での実行とし、\$で始まる場合はユーザ権限で実行します。ファイルに記述するコードは `bash` の文法に則って記述します。コンパイラのインストールを除き、root 権限でインストールするプログラムは `/usr/local` 以下のエンタリにインストールしていきますが、後で混乱を避けるために専用ディレクトリを作成し、その中にインストールしていく方式をとっています。ユーザ権限でインストールするプログラムはユーザのホームディレクトリ直下にインストールします。ダウンロードしたファイルに対して作業するときは、`/tmp`で行います。ファイルの編集等は `vi`で行っていますが、`gedit`や `emacs`等々の普段使われているエディタで勿論 OK です。適宜読み替えて下さい。

目次

WRF/WPS のインストール

- * 各種インストールの準備
- * Intel Compiler 11.0
- * NetCDF 3.6.3
- * Szip 2.1
- * HDF 4.2r4
- * Udonits 1.12.9
- * G2clib 1.1.8
- * GrADS 2.0.a5
- * WRF 2.2.1
- * WPS 2.2.1
- * WRF2GrADS 2.0

WRF/WPS の使用 (サンプルモード)

- * WPS
- * WRF
- * WRF2GrADS

WRF/WPS の使用 (実用モード)

- * WRF のリコンパイル
- * WPS with DomainWizard
- * WRF
- * WRF2GrADS

WRF/WPS をインストール

各種インストール前準備

はじめに、WRF/WPS を使用するためのプログラムをインストールするために、開発ツールやライブラリが必要となる。以降のインストールにおける依存性を解決するソフトウェアを `apt-get` でインストールしておく。

まずソースのコンパイルに必要なプログラムのインストール、その確認をしておく。gcc の関連するデベロッパが入っていないと `icc` の出力チェックでエラーが発生するという報告があるので、適宜インストール。

```
# yum install gcc gcc-*  
# yum install csh  
# yum install readline readline-devel  
# yum install gd gd-devel
```

HDF をインストールするために構文解析器生成プログラムをインストールする必要がある。byacc はパブリックドメイン yacc 構文解析器生成プログラム、bison は GNU 汎用構文解析器生成プログラム、flex はスキャナ（テキストパターン認識器）生成ツールである。各システムで若干名称が違う可能性があるため、該当する構文解析器系のプログラムを確認すること。

```
# yum install byacc  
# yum install bison  
# yum install flex
```

画像関係の開発用ライブラリその他を導入する。これらも適宜確認、インストールする。

```
# yum install libjpeg libjpeg-devel  
# yum install libpng libpng-devel  
# yum install zlib zlib-devel  
# yum install freeglut freeglut-devel  
# yum install jasper jasper-devel  
# yum install ncurses ncurses-devel
```

Intel Compiler 11.0

C/C++コンパイラのインストール。Intel の非商用ライセンス版コンパイラを取得。Free Non-Commercial Download の Compiler から C/C++ Compiler をダウンロードする。英語で非商用目的の使用かどうか、サポートを得ることはできないことへの了解が求められるが、どちらも Yes。登録画面に移り、メールアドレスと所在国を入力。そのアドレスにライセンスファイルが送られてくる。（ここでは/tmp 以下に一時保存）

Fedora では Security-Enhanced Linux (SELinux)モジュールがデフォルトで起動している。Intel Compiler をインストールするには、SELinux を無効化していないとインストールプロセスが進まない。OS 起動時にカーネルにブートパラメータとして selinux=0 を渡すか、setenforce で SELinux を無効化しておく。

```
# setenforce 0 // SELinux を無効
# getenforce
Permissive

# setenforce 1 // SELinux を有効
# getenforce
Enforcing
```

恒常的に SELinux を無効とすることは望ましいことではないが、/etc/sysconfig/selinux にて有効・無効の変更は可能。

```
# vi /etc/sysconfig/selinux

// ----- Edit the file -----
SELINUX=enforcing // SELinux を有効として起動
SELINUX=permissive // SELinux を無効として起動
```

SELinux を無効化したら、次にソースファイルを/tmp 以下に解凍し、作成されたディレクトリに入る。

```
# cd /tmp
# tar zxvf l_cproc_p_11.0.081.tgz
# cd l_cproc_p_11.0.081
```

インストールプログラムを実行。

```
# ./install.sh
```

デフォルトの選択通り進めば問題ない。ライセンスナンバーは、メールで送られてきた番号を入力。

次に Fortran コンパイラをインストールする。C/C++コンパイラ同様、Free Non-Commercial Download から Intel Fortran Compiler をダウンロードする。質問内容は C/C++と全く同じ。メールアドレスと国を記入し、ライセンスファイルを貰う。

ソースファイルを/tmp 以下に解凍し、作成されたディレクトリに入る。

```
# cd /tmp
# tar zxvf l_cprof_p_11.0.081.tgz
# cd l_cprof_p_11.0.081
```

インストールプログラムを実行。

```
# ./install.sh
```

デフォルトの選択通り進めば問題ない。ライセンスナンバーは、メールで送られてきた番号を入力。

C/C++ Compiler とはライセンスナンバーが違うので注意。

環境設定のパスを通す。

```
# vi ~/.bashrc

// ----- Add to end of the file -----
# Intel Compiler
Intel="/opt/intel/Compiler/11.0/081/bin"
ICC="${Intel}/iccvars.sh"
IFORT="${Intel}/ifortvars.sh"
if [ -f ${ICC} ]; then
    . ${ICC} ia32
fi
if [ -f ${IFORT} ]; then
    . ${IFORT} ia32
fi
```

環境設定を有効にし、バージョンチェック、簡易なテストプログラムのコンパイル等が成功したら終了。

```
# . ~/.bashrc
# icc -v
Version 11.0 // バージョンのチェック
# ifort -v
Version 11.0 // バージョンのチェック
```

NetCDF 3.6.3

NetCDFはNetwork Common Data Formの略である。WRFその他のコンパイルにはこのライブラリが必須となるので、早めにインストールしておく。NetCDF Downloadsにアクセスしダウンロードする。解凍後、ソースディレクトリへ移動。

```
# cd /tmp
# tar zxvf netcdf-3.6.3.tar.gz
# cd netcdf-3.6.3
```

ソースをコンパイルする。インストール用スクリプト（ここではinstallというファイル名とする）をエディタで記述する。

```
# vi install

// ----- Edit the new file -----
#!/bin/bash

export CC=icc
export CXX=icpc
export FC=ifort
export F90=ifort
export CFLAGS="-O2"
export FFLAGS="-O -mp"
export CPPFLAGS="-DNDEBUG -DpgiFortran"

DIR="/usr/local/netcdf-3.6.3"

mkdir -p $DIR
./configure --prefix=$DIR
make test
make install
```

インストール用スクリプトを実行する。

```
# sh install
```

インストール完了後、環境設定を記述する。

```
# vi ~/.bashrc

// ----- Add to end of the file -----
# NetCDF
export NETCDF="/usr/local/netcdf-3.6.3"
export NCHOME="${NETCDF}"
export NETCDFHOME="${NETCDF}"
export PATH="${NETCDF}/bin:${PATH}"
export LD_LIBRARY_PATH="${NETCDF}/lib:${LD_LIBRARY_PATH}"
export MANPATH="${NETCDF}/man:${MANPATH}"
```

環境設定後、設定を有効にする。

```
# . ~/.bashrc
```

Szip 2.1

SZIP は HDF、GrADS のオプションインストールの時に必要となる。次の HDF と併せてインストールしておく。ダウンロードは HDF の FTP サイトから可能。

```
# cd /tmp
# tar zxvf szip-2.1.tar.gz
# cd szip-2.1
```

インストール用スクリプトを記述。

```
# vi install

// ----- Edit the new file -----
#!/bin/bash

export CC=icc
export CXX=icpc
export FC=ifort
export F90=ifort
export CFLAGS="-O2"
export FFLAGS="-O -mp"

DIR="/usr/local/szip-2.1"

mkdir -p $DIR
./configure --prefix=$DIR
make
make install
```

スクリプトを実行する。

```
# sh install
```

次に環境設定を行う。

```
# vi ~/.bashrc

// ----- Add to end of the file -----
# Szip
export SZIP="/usr/local/szip-2.1"
export LD_LIBRARY_PATH="${SZIP}/lib:${LD_LIBRARY_PATH}"
```

設定後、設定を有効にする。

```
# . ~/.bashrc
```

HDF 4.2r4

HDFは Hierarchical Data Format という NetCDF と同様データフォーマットライブラリの一つである。GrADS のオプションで HDF のファイルを扱えるようにする時に必要。HDF4 と HDF5 があるが、現時点では HDF4 のライブラリを必要とする。

HDF Home Page から HDF4 のダウンロードページに行き、Linux 用のソースファイルをダウンロードする。解凍後、ソースディレクトリに移る。

```
# cd /tmp
# tar zxvf HDF4.2r4.tar.gz
# cd HDF4.2r4
```

インストール用スクリプトを記述する。

```
# vi install

// ----- Edit the new File -----
#!/bin/bash

export CC=icc
export CXX=icpc
export F77=ifort
export FC=ifort

DIR="/usr/local/hdf-4.2r4"

mkdir -p $DIR
./configure --prefix=$DIR --with-szlib=/usr/local/szip-2.1 --disable-netcdf
make
make install
```

スクリプトを実行する。

```
# sh install
```

インストール後、bin 以下に nc*実行ファイルが置かれるが、これは netcdf のインストールで既にインストール済みである。パスの通し方によっては後々 nc*を実行したときに netcdf のものではなくこちらが実行されてしまい、上手く動かなくなる。old ディレクトリを作成し、実行ファイルを移しておく。

```
# cd /usr/local/hdf-4.2r4
# mkdir old
# mv bin/nc* old
```

また、次の ncarg インストール時に HDF のインクルードファイルの書き換えが必要になる。これをバックアップし、修正しておく。

```
# cp include/hdfi.h old
# vi include/hdfi.h

// ----- Edit lines: 962-965 -----
// rewrite
#include <sys¥types.h>
#include <sys¥stat.h>
#include <io.h>
#include <conio.h>
```

```
// to
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/io.h>
#include <ncurses.h>
```

次に環境設定を行う。

```
# vi ~/.bashrc

// ----- Add to end of the file -----
# HDF
export HDF="/usr/local/hdf-4.2r4"
export PATH="${HDF}/bin:${PATH}"
export LD_LIBRARY_PATH="${HDF}/lib:${LD_LIBRARY_PATH}"
```

設定後、設定を有効にする。

```
# . ~/.bashrc
```

Udunits 1.12.9

GrADS のインストール時にこのライブラリが必要となる。ダウンロードは Unidata から最新バージョンを選択。解凍し、ディレクトリへ移動。

```
# cd /tmp
# tar zxvf udunits-1.12.9.tar.gz
# cd udunits-1.12.9/src
```

インストール用スクリプトを記述。

```
# vi install

// ----- Edit the new file -----
#!/bin/bash

export CC=icc
export CXX=icpc
export FC=ifort
export F90=ifort
export CFLAGS="-O2"
export FFLAGS="-O"
export CPPFLAGS="-DNDEBUG -DpgiFortran"

DIR="/usr/local/udunits-1.12.9"

mkdir -p $DIR
./configure --prefix=$DIR
make
make install
```

スクリプトを実行する。

```
# sh install
```

次に環境設定を行う。

```
# vi ~/.bashrc

// ----- Add to end of the file -----
# Udunits
export UDUNITS="/usr/local/udunits-1.12.9"
export PATH="${UDUNITS}/bin:${PATH}"
export LD_LIBRARY_PATH="${UDUNITS}/lib:${LD_LIBRARY_PATH}"
```

設定後、設定を有効にする。

```
# . ~/.bashrc
```

G2clib 1.1.8

GrADS で grib2 ファイルを扱うために必要なライブラリ。ダウンロードは NCEP の GRIB2 ページから選択。

```
# cd /tmp
# tar xvf g2clib-1.1.8.tar
```

makefile のコンパイラ設定を修正。

```
# cd /tmp/g2clib-1.1.8
# vi makefile

// ----- Edit lines: 24-26 -----
CC=icc
LIB=libgrib2c.a
ARFLAGS=
```

make でコンパイル。成功すると、libgrib2c.a ファイルが作成される。/usr/local に移動させ、所有者を変更。

```
# make
# cd ..
# mv g2clib-1.1.8 /usr/local
# chown 0:0 -R /usr/local/g2clib-1.1.8
```

次に環境設定を行う。

```
# vi ~/.bashrc

// ----- Add to end of the file -----
# G2clib
export G2CLIB="/usr/local/g2clib-1.1.8"
export JASPERLIB="/usr/lib"
export JASPERINC="/usr/include"
```

設定後、設定を有効にする。

```
# . ~/.bashrc
```

GrADS 2.0.a5

正式名称は Grid Analysis and Display System。解析データの可視化ソフトである。

まず、各種リンカファイルに対し、`/usr/lib` 直下にシンボリックリンクを貼る。

```
# cd /usr/lib
# ln -s /usr/local/*/lib/*.a .
# ln -s /usr/local/*/*.a .
```

Downloading GrADS Software からソースファイルをダウンロード。解凍し、ディレクトリへ移動。

```
# cd /tmp
# tar zxvf grads-src-2.0.a5.tar.gz
# cd grads-2.0.a5
```

インストール用スクリプトを記述する。

```
# vi install

// ----- Edit the new file -----
#!/bin/bash

export CC=icc
export CXX=icpc
export FC=ifort
export F90=ifort
export CFLAGS="-O2"
export FFLAGS="-O -mp"
export CPPFLAGS="-I/usr/local/netcdf-3.6.3/include -I/usr/local/zip-2.1/include -I/usr/local/hdf-4.2r4/include -I/
usr/local/udunits-1.12.9/include -I/usr/local/g2clib-1.1.8"
export LIBS="-lnetcdf -lsz"

DIR="/usr/local/grads-2.0.a5"

mkdir -p $DIR
./configure --prefix=$DIR
make
make install
```

インストール用スクリプトを実行。ドキュメント等をソースディレクトリからコピーする。

```
# sh install
# cp -a data doc etc lib m4 /usr/local/grads-2.0.a5
```

次にスクリプトファイルをダウンロード、インストールする。`wget` を使用し、FTP サイトからデータを取得する。

```
# cd /usr/local/grads-2.0.a5
# mkdir scripts
# cd scripts
# wget ftp://grads.iges.org/grads/scripts/*
```

インストールが完了したら、環境設定を記述しておく。

```
# vi ~/.bashrc
```

```
// ----- Add to end of the file -----  
# GrADS  
export GRADS="/usr/local/grads-2.0.a5"  
export PATH="${GRADS}/bin:${PATH}"  
export GASCRP="${GRADS}/scripts"  
export GADDIR="${GRADS}/data"
```

環境設定を有効にしたら終了。

```
# . ~/.bashrc
```

WRF 2.2.1

Weather Research and Forecasting Model（気象解析予報モデル）。気象シミュレーション解析プログラム
の一種。

WRFはユーザー権限でインストールするため、差し支えなければrootの.bashrcを一般ユーザの.bashrc
にコピーする。rootと一般ユーザで設定が異なるならば、該当箇所をエディタでコピーすること。ファ
イルを丸ごとコピーした場合は所有者権限の変更も行っておく。

```
# cd /home/{USER_NAME}
# cp /root/.bashrc .
# chown {USER_NAME}:{GROUP_NAME} .bashrc
# su - {USER_NAME}
```

以降、一般ユーザ権限での作業となる。WRF Download から WRF model tar file をダウンロード。mkdir
でホームディレクトリ下に WRF 用のディレクトリを作成する。

```
$ mkdir ~/WRF
$ cd ~/WRF
$ tar zxvf /tmp/WRFV2.2.1.TAR.gz
$ cd WRFV2
```

WRFV2/arch/configure.defaults を編集する。「AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler
(single-threaded, no nesting)」の設定を基にする。該当行は line:6810-6937 となる。変更する箇所のみを
記載する。

```
$ vi arch/configure.defaults
// ----- Edit lines: 6810-6937 -----
CC      =   icc
CPP     =   ifort -E -free
```

設定の変更が終了したら configure を実行する。画面中設定は7を採用。configure が終わったらコンパ
イル。

```
$ ./configure
$ ./compile em_real
```

run ディレクトリに ndown.exe、real.exe、wrf.exe が作成されていればコンパイル成功。

WPS 2.2.1

WRF Preprocessing System の略。WRFSI に代わり、WRF Version2.2 から WPS が前処理プログラムとなる。WRFSI から大きく変更された点は、WPS の前に WRF をインストールする必要があること、GUI だけでなく CUI でも処理が可能になったこと等。ここでは簡単のために、WPS の GUI をインストールしない。

WRF Download のページから WPS をダウンロードする。また地理データとして `geog.tar.gz` もダウンロードすること。WRFSI のものと若干異なるので、WPS 用に新しくインストールする必要がある。

```
$ cd ~/WRF
$ tar zxvf /tmp/WPSV2.2.TAR.gz
$ tar zxvf /tmp/geog.tar.gz
$ mv geog GEOG
$ cd WPS
```

WRF 同様、WPS/arch/configure.defaults の該当箇所を修正。変更点のみ記載する。

```
$ vi arch/configure.defaults

// ----- Edit lines: 253-265 -----
CC          =      icc
SCC         =      icc
CPP         =      ifort -E -free -C -P
```

configure で設定。設定モードは 6 を選択する。設定後、コンパイルを実行。コンパイルには、WPS のディレクトリと同階層に "WRFV2" というディレクトリが存在し、その中に WRF がインストールされている必要がある。

```
$ ./configure
$ ./compile
```

geogrid.exe、ungrib.exe、metgrid.exe の 3 つが出力されていればコンパイル完了。

WRF2GrADS 2.0

WRFで出力されたファイルをGrADSで可視化する。この出力がうまくいけば、一通りの解析とその表示が可能になる。WRF Downloadの「WRF Post-Processing Software」から「Convert WRF model output in netCDF to GrADS format」にアクセスし、WRF2GrADSをダウンロードする。解凍後、ディレクトリへ移動。

```
$ cd ~/WRF
$ tar zxvf /tmp/wrf2grads.tar.gz
$ cd WRF2GrADS
```

インストールの設定を行う。Makefileの「linux flag (INTEL)」の箇所を修正。コメントアウトを解除し、ライブラリファイル、インクルードファイルへのパスを修正する。CPPやフラグも修正しておく。

```
$ vi Makefile

// ----- Edit lines: 29-34 -----
LIBNETCDF = -L/usr/local/netcdf-3.6.3/lib -lnetcdf -lm
INCLUDE = -I/usr/local/netcdf-3.6.3/include -I./
FC = ifort
FCFLAGS = -C -FR
CPP = ifort -E -free
CPPFLAGS = -I. -C -DRECL1
```

また、このままコンパイルすると「fortrl: severe (193): Run-Time Check Failure. The variable 'module_wrf_to_grads_util_mp_time_calc_\$HOUR1' is being used without being defined」という実行時エラーを起こす。これは初期化されていない変数を使用しているというエラー。詳細は「インテル(R) Visual Fortran コンパイラ 9.1 Windows* 版 リリースノート」の変更点のうちの該当箇所を参照（Linux版のリリースノートでは9.0、9.1ともにそのような変更点は記載されていない）。これを回避するために、module_wrf_to_grads_util.Fの1066-1068行目を修正する。

```
$ vi module_wrf_to_grads_util.F

// ----- Edit lines: 1066-1068 -----
// rewrite
hour1=hours
mins1=minutes
elseif ( it == 2) then
// to
elseif ( it == 2) then
hour1=hours
mins1=minutes
```

アルゴリズムやインデントの状態を考慮すると、書き順ミスによる単純なバグ?のようである。

修正が終わったらコンパイルを実行しても良いが、他のソースファイルまで実行可能になっているので色分けをしているときに見づらくなる。属性を変更してからコンパイル。

```
$ chmod 644 *
$ make
```

wrf_to_grads 実行ファイルが作成されていればコンパイル終了。

WRF/WPS を使用 (サンプルモード)

WPS

WPS を使うには FNL データを必要とする。WRF Download にある「WRF Preprocessing System test data」をダウンロードし、適当なディレクトリを作成して解凍。

```
$ mkdir -p ~/WRF/FNL/sample
$ cd ~/WRF/FNL/sample
$ tar zxvf /tmp/avn_data.tar.gz
$ cd ~/WRF/WPS
```

WPS は「geogrid.exe」→「ungrib.exe」→「metgrid.exe」の順でプログラムを走らせる。これらのプログラムの設定には、namelist.wps を一貫して使用する。

FNL データにアクセスするためには、シンボリックリンクを設定する。シンボリックリンクを貼るための C シェルスクリプトが link_grib.csh として提供されているのでこれを利用する。

```
$ ./link_grib.csh ../FNL/sample/*
```

処理する変数テーブルに対してシンボリックリンクを貼る。ここでは NCEP2 を使用する。

```
$ ln -sf ungrib/Variable_Tables/Vtable.NCEP2 Vtable
```

namelist.wps を以下のように編集する。初期設定からの変更部分を赤色で記載。

```
$ vi namelist.wps

// ----- Edit all Lines -----
&share
  wrf_core = 'ARW',
  max_dom = 1,
  start_date = '2000-01-24_12:00:00',
  end_date = '2000-01-25_00:00:00',
  interval_seconds = 21600,
  io_form_geogrid = 2,
/

&geogrid
  parent_id = 1,
  parent_grid_ratio = 1,
  i_parent_start = 1,
  j_parent_start = 1,
  e_we = 74,
  e_sn = 61,
  geog_data_res = '10m',
  dx = 30000,
  dy = 30000,
  map_proj = 'lambert',
  ref_lat = 34.83,
  ref_lon = -81.03,
  truelat1 = 30.0,
  truelat2 = 60.0,
  stand_lon = -90.0,
  geog_data_path = '../GEOG'
/
```

```
&ungrib
  out_format = 'WPS',
  prefix = 'NCEP2',
/

&metgrid
  fg_name = 'NCEP2',
  io_form_metgrid = 2,
/

&mod_levs
  press_pa = 201300 , 200100 , 100000 ,
             95000 , 90000 ,
             85000 , 80000 ,
             75000 , 70000 ,
             65000 , 60000 ,
             55000 , 50000 ,
             45000 , 40000 ,
             35000 , 30000 ,
             25000 , 20000 ,
             15000 , 10000 ,
             5000 , 1000
/
```

編集が終了したら、順序通りにプログラムを走らせる。

```
$ ./geogrid.exe
$ ./ungrib.exe
$ ./metgrid.exe
```

出力に成功したら、met_em.*.nc というファイル群が作成される。WRFではこのファイル群を使用する。

WRF

WRFはrunディレクトリでプログラムを実行する。ディレクトリに移動し、WPSで作成したファイルをこのディレクトリに置く。

```
$ cd ~/WRF/WRFV2/run
$ mv ~/WRF/WPS/met_em* .
```

WRFの設定はnamelist.inputで行う。namelist.inputに関する詳しい情報は、README.namelistを参照すること。以下、設定ファイルの情報を記す。初期設定からの変更点は赤で記載。

```
$ vi namelist.input

// ----- Edit all lines -----
&time_control
run_days           = 0,
run_hours          = 12,
run_minutes        = 0,
run_seconds        = 0,
start_year         = 2000, 2000, 2000,
start_month        = 01, 01, 01,
start_day          = 24, 24, 24,
start_hour         = 12, 12, 12,
start_minute       = 00, 00, 00,
start_second       = 00, 00, 00,
end_year           = 2000, 2000, 2000,
end_month          = 01, 01, 01,
end_day            = 25, 25, 25,
end_hour           = 00, 12, 12,
end_minute         = 00, 00, 00,
end_second         = 00, 00, 00,
interval_seconds   = 21600
input_from_file    = .true.,.false.,.false.,
history_interval   = 180, 60, 60,
frames_per_outfile = 1000, 1000, 1000,
restart            = .false.,
restart_interval   = 5000,
io_form_history    = 2
io_form_restart    = 2
io_form_input      = 2
io_form_boundary   = 2
debug_level        = 0
/

&domains
time_step          = 180,
time_step_fract_num = 0,
time_step_fract_den = 1,
max_dom            = 1,
s_we               = 1, 1, 1,
e_we               = 74, 112, 94,
s_sn               = 1, 1, 1,
e_sn               = 61, 97, 91,
s_vert             = 1, 1, 1,
e_vert             = 27, 28, 28,
num_metgrid_levels = 27
```

```

dx                = 30000, 3333, 1111,
dy                = 30000, 3333, 1111,
grid_id          = 1, 2, 3,
parent_id        = 0, 1, 2,
i_parent_start   = 0, 30, 30,
j_parent_start   = 0, 20, 30,
parent_grid_ratio = 1, 3, 3,
parent_time_step_ratio = 1, 3, 3,
feedback         = 1,
smooth_option    = 0
/

&physics
mp_physics       = 3, 3, 3,
ra_lw_physics    = 1, 1, 1,
ra_sw_physics    = 1, 1, 1,
radt             = 30, 30, 30,
sf_sfclay_physics = 1, 1, 1,
sf_surface_physics = 1, 1, 1,
bl_pbl_physics   = 1, 1, 1,
bldt             = 0, 0, 0,
cu_physics       = 1, 1, 0,
cudt             = 5, 5, 5,
isfflx           = 1,
ifsnow           = 0,
icloud           = 1,
surface_input_source = 1,
num_soil_layers  = 5,
ucmcall          = 0,
mp_zero_out      = 0,
maxiens          = 1,
maxens           = 3,
maxens2          = 3,
maxens3          = 16,
ensdim           = 144,
/

&fdda
/

&dynamics
w_damping        = 0,
diff_opt         = 1,
km_opt           = 4,
diff_6th_opt     = 0,
diff_6th_factor  = 0.12,
base_temp        = 290.
damp_opt         = 0,
zdamp            = 5000., 5000., 5000.,
dampcoef         = 0.01, 0.01, 0.01
khdif            = 0, 0, 0,
kvdif            = 0, 0, 0,
non_hydrostatic  = .true., .true., .true.,
pd_moist         = .false., .false., .false.,
pd_scalar        = .false., .false., .false.,

```

```

/

&bdy_control
spec_bdy_width      = 5,
spec_zone           = 1,
relax_zone          = 4,
specified            = .true., .false.,.false.,
nested              = .false., .true., .true.,
/

&grib2
/

&namelist_quilt
nio_tasks_per_group = 0,
nio_groups = 1,
/

```

設定が完了したら、初期値・境界値データを作成する。Vine Linux ではスタックサイズに 8192KB 制限があるので、real.exe を実行する前にスタックサイズを無制限にしておく。

```

$ ulimit -s unlimited
$ ./real.exe

```

初期値・境界値データが wrfbdy、wrfinput というファイルで作成される。この後、wrf.exe を実行してシミュレーションを行う。

```

$ ulimit -s unlimited
$ ./wrf.exe

```

計算範囲やタイムステップによって実行時間は大きく変化する。シミュレーションが終了すると、wrfout ファイルが作成される。

WRF2GrADS

WRFで作成されたファイルはNetCDFのファイル形式をとっている。この出力ファイルをGrADSで表示できるように変換する。WRFで作成したファイルをWRF2GrADSディレクトリに移動する。

```
$ cd ~/WRF/WRF2GrADS
$ mv ~/WRF/WRFV2/run/wrfout* .
```

GrADS変換プログラムを走らせるために、control_fileを編集する。先頭に空白文字が含まれる行は無視される。

```
$ vi control_file

// ----- Edit all lines -----
-2          ! number of times to put in GrADS file, negative means ignore the times
2000-01-24_12:00:00
2000-01-24_18:00:00
2000-01-25_00:00:00
end_of_time_list

                ! 3D variable list for GrADS file
                ! indent one space to skip
U            ! U Component of wind
V            ! V Component of wind
  UMET       ! U Component of wind - rotated (diagnostic)
  VMET       ! V Component of wind - rotated (diagnostic)
W            ! W Component of wind
  THETA      ! Theta
  TK         ! Temperature in K
  TC         ! Temperature in C
  TKE        ! TURBULENCE KINETIC ENERGY
P            ! Pressure (hPa)
Z            ! Height (m)
  QVAPOR     ! Vapor
  QCLOUD     ! Cloud Water
  QRAIN      ! Rain Water
  QICE       ! -
  QSNOW      ! -
  QGRAUP     ! -
  TKE_MYJ    ! TKE FROM MELLOR-YAMADA-JANJIC
PB           ! BASE STATE PRESSURE AT HALF LEVEL
TD           ! Dewpoint Temperature (diagnostic)
RH           ! Relative Humidity (diagnostic)
                ! Soil variables
  TSLB       ! SOIL TEMPERATURE
  SMOIS      ! SOIL MOISTURE
                ! below a list of fields from SI static and input files
  LANDUSEF   ! -
  SOILCTOP   ! -
  SOILCBOT   ! -
  SPECHUMD   ! -
  GREEN12M   ! -
  ALBDO12M   ! -
end_of_3dvar_list
  ACSNOM     ! ACCUMULATED MELTED SNOW
  ACSNOW     ! ACCUMULATED SNOW
  AKHS       ! SFC EXCH COEFF FOR HEAT
```

AKMS ! SFC EXCH COEFF FOR MOMENTUM
 CANWAT ! CANOPY WATER
 GLW ! DOWNWARD LONG WAVE FLUX AT GROUND SURFACE
 GSW ! DOWNWARD SHORT WAVE FLUX AT GROUND SURFACE
 HFX ! UPWARD HEAT FLUX AT THE SURFACE
 HGT ! Terrain Height
 IVGTYP ! VEGETATION TYPE
 ISLTYP ! SOIL TYPE
 LU_INDEX ! LAND USE CATEGORY
 MAPFAC_M ! Map scale factor on mass grid
 MU ! Perturbation dry air mass in column
 MUB ! base state dry air mass in column
 MU0 ! initial dry mass in column
 Q2 ! QV at 2 M
 QFX ! UPWARD MOISTURE FLUX AT THE SURFACE
 RAINC ! ACCUMULATED TOTAL CUMULUS PRECIPITATION
 RAINCV ! TIME-STEP CUMULUS PRECIPITATION
 RAINNC ! ACCUMULATED TOTAL GRID SCALE PRECIPITATION
 SFROFF ! SURFACE RUNOFF
 slvl ! sea level pressure
 SMSTAV ! MOISTURE VARIBILITY
 SNOW ! SNOW WATER EQUIVALENT
 SNOWC ! FLAG INDICATING SNOW COVERAGE (1 FOR SNOW COVER)
 SST ! SEA SURFACE TEMPERATURE
 T2 ! TEMP at 2 M
 TH2 ! POT TEMP at 2 M
 TMN ! SOIL TEMPERATURE AT LOWER BOUNDARY
 TSK ! SURFACE SKIN TEMPERATURE
 U10 ! U at 10 M
 U10M ! U at 10 M - rotated
 V10 ! V at 10 M
 V10M ! V at 10 M - rotated
 UDROFF ! UNDERGROUND RUNOFF
 VEGFRA ! VEGETATION FRACTION
 WEASD ! WATER EQUIVALENT OF ACCUMULATED SNOW
 XLAT ! LATITUDE, SOUTH IS NEGATIVE
 XLONG ! LONGITUDE, WEST IS NEGATIVE
 XLAND ! LAND MASK (1 FOR LAND, 2 FOR WATER)
 ! below a list of fields from SI static and input files
 ALBBCK ! -
 LANDMASK ! -
 PMSL ! -
 SLOPECAT ! -
 SHDMAX ! -
 SHDMIN ! -
 SNOALB ! -
 SOILHGT ! -
 ST000010 ! -
 ST010040 ! -
 ST040100 ! -
 ST100200 ! -
 SM000010 ! -
 SM010040 ! -
 SM040100 ! -
 SM100200 ! -

```

TOPOSTDV      ! -
TOPOSLPX      ! -
TOPOSLPY      ! -
XICE          ! -
end_of_2dvar_list

                ! All list of files to read here
                ! Indent not to read
                ! Full path OK

/DATA/wrfstatic_d01
/DATA/wrfstatic_d02
/DATA/wrf_real_input_em.d01.2000-01-24_12:00:00
/DATA/wrf_real_input_em.d02.2000-01-24_12:00:00
/DATA/real/wrfinput_d01

./wrfout_d01_2000-01-24_12:00:00

/DATA/b_wave/wrfout_d01_0001-01-01_00:00:00
/DATA/grav2d_x/wrfout_d01_0001-01-01_00:00:00
/DATA/hill2d_x/wrfout_d01_0001-01-01_00:00:00
/DATA/quarter_ss/wrfout_d01_0001-01-01_00:00:00
/DATA/squall2d_x/wrfout_d01_0001-01-01_00:00:00
/DATA/squall2d_y/wrfout_d01_0001-01-01_00:00:00
end_of_file_list

                ! Now we check to see what to do with the data
real          ! real (input/output) / ideal / static
1            ! 0=no map background in grads, 1=map background in grads
-1          ! specify grads vertical grid
                ! 0=cartesian,
                ! -1=interp to z from lowest h
                ! 1 list levels (either height in km, or pressure in mb)
1000.0
950.0
900.0
850.0
800.0
750.0
700.0
650.0
600.0
550.0
500.0
450.0
400.0
350.0
300.0
250.0
200.0
150.0
100.0

```

編集が終了したら、変換プログラムを走らせる。

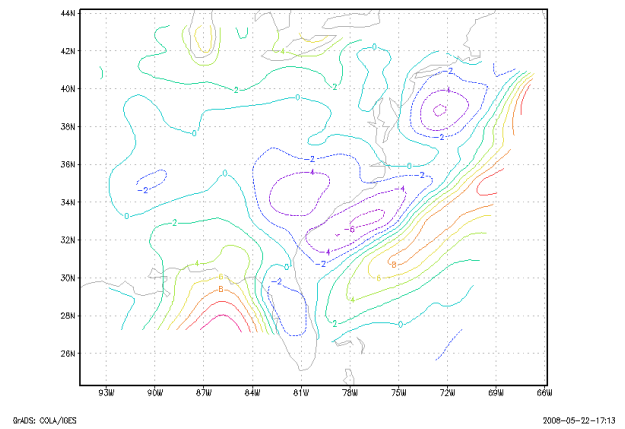
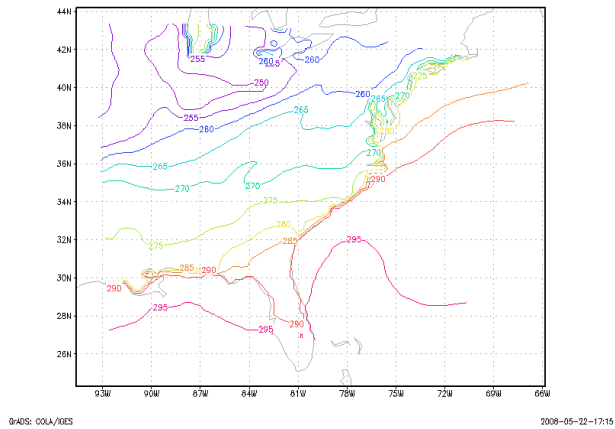
```
$ ./wrf_to_grads control_file sample
```

sample は出力名となる。出力が完了すると、GrADS 形式の sample.cti と sample.dat が作成される。

GrADS を起動し、出力を確認する。

```
$ grads  
ga-> open sample.ctl  
ga-> q file  
ga-> d sst  
ga-> c  
ga-> d u10m
```

設定通りに行うと、以下のような画像が出力される。



WRF/WPSの使用（実用モード）

WRFのリコンパイル

WRFにおいてネスティングが行えるようにリコンパイルする。通常、気象予報を行う時は対象領域に対し、それを大きく取り囲むように第一ネストを設定し、対象領域を第二ネストとして設定する。

WRFV2/arch/configure.defaults を編集する。「AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler (single-threaded, allows ARW nesting using RSL without MPI)」の設定を基にする。該当行は line:6939-7073 となる。変更する箇所のみを記載する。

```
$ vi arch/configure.defaults

// ----- Edit lines: 6939-7073 -----
CC      =   icc
CPP     =   ifort -E -free
CC_TOOLS = $(CC)
```

solve_em のコンパイルに必要なため、次の情報を solve_interface の情報の下に加えておく。

```
$ vi arch/configure.defaults

// ----- Edit lines: 7032-7033 -----
solve_interface.o : solve_interface.F
solve_em.o : solve_em.F

// ----- Edit lines: 7050-7051 -----
solve_interface.o          \
solve_em.o                 \
```

設定の変更終了後、サンプル等を実行したことがあればまず clean を実行し、その後 configure を実行する。画面中設定は 8 を採用。configure が終わったらコンパイル。

```
$. /clean
$. /configure
$. /compile em_real
```

run ディレクトリに ndown.exe、real.exe、wrf.exe が作成されていればコンパイル成功。

WPS with DomainWizard

WPS の設定をテキストファイルのみで行うのは実用的ではないので、DomainWizard という WPS 用 GUI を用いて設定を行う。DomainWizard は Java を利用しているため、当然 Java がインストールされていなければならない。Fedora では OpenOffice.org が自動インストールされており（初期インストール設定による）、OpenOffice.org に付属の Java Runtime Environment で問題はない。インストールされていない場合は、yum でインストールするか、OpenOffice.org のサイトから該当圧縮ファイルをダウンロードしインストールすること。

また、ここではサンプルモードの時と同様にテスト用として配布されている FNL データを用いるが、UCAR のサイトで配布されている NCEP FNL Operational Model Global Tropospheric Analyses から実用的な FNL データをダウンロードできる。ダウンロードにはユーザ登録が必要である。

DomainWizard は WRF Domain Wizard のページからダウンロードが可能である。ファイルをダウンロード後、解凍。展開されたファイルを実行する。

```
$ mkdir -p ~/WRF/DomainWizard
$ unzip /tmp/WRFDomainWizard.zip -d ~/WRF/DomainWizard
$ cd ~/WRF/DomainWizard
$ mkdir domains
$ sh run_DomainWizard
```

Java による GUI が起動する。最初の画面で、WPS、GEOG、設定されたファイルを置く domains ディレクトリの位置を絶対パスで指定する。

WRF Domain Wizard Configuration

Directory Configuration ---

...

WPS Programs: /home/{USER_NAME}/WRF/WPS

Geography: /home/{USER_NAME}/WRF/GEOG

Domains: /home/{USER_NAME}/WRF/DomainWizard/domains

ドメインの設定を行う。新しく作る場合は New Domain を選択し、Next ボタンをクリック。

ドメインの名前と説明の設定は適当。ここでは日付として Name/Description とともに 20000124 と入力する。Next ボタンをクリック。

ドメインを指定する世界地図が表示される。地図サイズが大きすぎる場合は、右上の Map 設定から Scale: 100% とする。テストのため、マウスを用いて日本の関東地方付近を適当に短形で囲む。Projection Options で Type: Lambert Conformal を選択し、Actions で Update Map を行う。すると、短形選択付近の地図だけが抜き出された形で表示される。Grid Options を次のように指定する。

- * Horizontal dimension X: 100
- * Horizontal dimension Y: 100
- * Grid point distance (km): 10
- * Geographic data resolution: 30s

右上の Nest タブをクリックし、Nested Domain Properties から New をクリック。New Nest の設定を次のように指定する。

- * Parent ID: 1
- * Grid spacing ratio to parent: 5
- * Geographic data resolution: 30s
- * (LLI) Left: 30
- * (URI) Right: 70

- * (URJ) Top: 70
- * (LLJ) Bottom: 30

設定が終わったら OK をクリックし、第 2 ネストが表示されることを確認する。Next ボタンをクリック。

namelist.input の設定を行う。GUI Editor と Text Editor の二つをタブで切り替えて編集できる。追加行については Text Editor で追加できる。以下、namelist.input の例。編集した部分を赤で記載している。DomainWizard 側が自動で編集している部分もあるので、GUI Editor での表示をよく読んで書き換えること。

```
&time_control
run_days      = 0,
run_hours     = 12,
run_minutes   = 0,
run_seconds   = 0,
start_year    = 2000, 2000,
start_month   = 01,  01,
start_day     = 24,  24,
start_hour    = 12,  12,
start_minute  = 00,  00,
start_second  = 00,  00,
end_year      = 2000, 2000,
end_month     = 01,  01,
end_day       = 25,  25,
end_hour      = 00,  00,
end_minute    = 00,  00,
end_second    = 00,  00,
interval_seconds = 21600,
input_from_file = .true., .true.,
history_interval = 180,  60,
frames_per_outfile = 1,  1,
restart       = .false.,
restart_interval = 5000,
io_form_history = 2,
io_form_restart = 2,
io_form_input  = 2,
io_form_boundary = 2,
debug_level   = 0,
/

&domains
time_step      = 60,
time_step_fract_num = 0,
time_step_fract_den = 1,
max_dom        = 2,
s_we           = 1,  1,
e_we           = 100, 201,
s_sn           = 1,  1,
e_sn           = 100, 201,
s_vert         = 1,  1,
e_vert         = 28, 28,
num_metgrid_levels = 27,
dx             = 10000, 2000,
dy             = 10000, 2000,
grid_id        = 1,  2,
```

```

parent_id      = 1,    1,
i_parent_start = 1,    30,
j_parent_start = 1,    30,
parent_grid_ratio = 1,    5,
parent_time_step_ratio = 1,    5,
feedback       = 1,
smooth_option  = 0,
/

&physics
mp_physics     = 3,    3,
mp_zero_out    = 2,
mp_zero_out_thresh = 1.e-8,
ra_lw_physics  = 1,    1,
ra_sw_physics  = 1,    1,
radt           = 30,   30,
sf_sfclay_physics = 2,    2,
sf_surface_physics = 2,    2,
bl_pbl_physics = 2,    2,
bltd          = 0,    0,
cu_physics     = 1,    0,
cudt          = 10,   10,
isfflx        = 1,
ifsnow        = 0,
icloud        = 1,
surface_input_source = 1,
num_soil_layers = 4,
maxiens       = 1,
maxens        = 3,
maxens2       = 3,
maxens3       = 16,
ensdim        = 144,
/

&fdda
/

&dynamics
dyn_opt       = 2,
rk_ord        = 3,
w_damping     = 0,
diff_opt      = 0,
km_opt        = 1,
damp_opt      = 0,
base_temp     = 290.,
zdamp         = 5000., 5000.,
dampcoef      = 0.01, 0.01,
khdif         = 0,    0,
kvdif         = 0,    0,
smdiv         = 0.1, 0.1,
emdiv         = 0.01, 0.01,
epssm         = 0.1, 0.1,
non_hydrostatic = .true., .true.,
time_step_sound = 4,    4,
h_mom_adv_order = 5,    5,

```

```

v_mom_adv_order      = 3,    3,
h_sca_adv_order      = 3,    5,
v_sca_adv_order      = 2,    3,
pd_moist              = .false., .false.,
pd_scalar             = .false., .false.,
pd_chem              = .false., .false.,
pd_tke                = .false., .false.,
/

&bdy_control
spec_bdy_width       = 5,
spec_zone            = 1,
relax_zone           = 4,
specified            = .true., .false.,
periodic_x           = .false., .false.,
symmetric_xs         = .false., .false.,
symmetric_xe         = .false., .false.,
open_xs              = .false., .false.,
open_xe              = .false., .false.,
periodic_y           = .false., .false.,
symmetric_ys         = .false., .false.,
symmetric_ye         = .false., .false.,
open_ys              = .false., .false.,
open_ye              = .false., .false.,
nested               = .false., .true.,
/

&grib2
/

&namelist_quilt
nio_tasks_per_group  = 0,
nio_groups            = 1,
/

```

編集が終わったら Next ボタンをクリック。

WPS を実際に実行するプロセスに入る。Grib Table Name は Vtable.GFS を選択し、Grib Files Dir は FNL データを置いたディレクトリを指定。Grib Files でサンプルの FNL データを全て加える。開始時間と終了時間は、2000-01-24-12:00:00 から 2000-01-25-00:00:00 で、時間間隔は 6(hr)である。指定したら、右の Run Preprocessors から geogrid、ungrib、metgrid の順をクリックする。全てが成功したら、Next ボタンをクリック。

生成されたファイルの確認画面。geo_em.*.nc が 2 つ、met_em.*.nc が 6 つ生成されていることを確認したら、Exit ボタンを押して終了。

WRF

WRFの実行はサンプルモードの時とさほど変わりはない。DomainWizardで生成したnamelist.input及びncファイルをrunディレクトリ下に置く。

```
$ cd ~/WRF/DomainWizard/domains/20000124
$ mv namelist.input geo_em.*.nc met_em.*.nc ~/WRF/WRFV2/run
$ cd ~/WRF/WRFV2/run
```

スタックサイズを無制限にし、プログラムを実行。

```
$ ulimit -s unlimited
$ ./real.exe
$ ./wrf.exe
```

WRFの実行は非常に時間がかかる。CPUプロセスに余裕をもてる時に実行するのが望ましい。

WRF2GrADS

サンプルモードの時とほぼ変わらない。WRF で作成されたファイルは NetCDF のファイル形式をとっているため、この出力ファイルを GrADS で表示できるように変換する。WRF で作成したファイルを WRF2GrADS ディレクトリに移動する。

```
$ cd ~/WRF/WRF2GrADS
$ mv ~/WRF/WRFV2/run/wrfout* .
```

GrADS 変換プログラムを走らせるために、control_file を編集する。先頭に空白文字が含まれる行は無視される。

```
$ vi control_file

// ----- Edit all lines -----
-1          ! number of times to put in GrADS file, negative means ignore the times
end_of_time_list

          ! 3D variable list for GrADS file
          ! indent one space to skip
U          ! U Component of wind
V          ! V Component of wind
  UMET      ! U Component of wind - rotated (diagnostic)
  VMET      ! V Component of wind - rotated (diagnostic)
W          ! W Component of wind
  THETA     ! Theta
  TK        ! Temperature in K
  TC        ! Temperature in C
  TKE       ! TURBULENCE KINETIC ENERGY
P          ! Pressure (hPa)
Z          ! Height (m)
  QVAPOR    ! Vapor
  QCLOUD    ! Cloud Water
  QRAIN     ! Rain Water
  QICE      ! -
  QSNOW     ! -
  QGRAUP    ! -
  TKE_MYJ   ! TKE FROM MELLOR-YAMADA-JANJIC
PB         ! BASE STATE PRESSURE AT HALF LEVEL
TD         ! Dewpoint Temperature (diagnostic)
RH         ! Relative Humidity (diagnostic)
          ! Soil variables
  TSLB      ! SOIL TEMPERATURE
  SMOIS     ! SOIL MOISTURE
          ! below a list of fields from SI static and input files
  LANDUSEF  ! -
  SOILCTOP  ! -
  SOILCBOT  ! -
  SPECHUMD  ! -
  GREEN12M  ! -
  ALBDO12M  ! -
end_of_3dvar_list
  ACSNOM    ! ACCUMULATED MELTED SNOW
  ACSNOW    ! ACCUMULATED SNOW
  AKHS      ! SFC EXCH COEFF FOR HEAT
  AKMS      ! SFC EXCH COEFF FOR MOMENTUM
  CANWAT    ! CANOPY WATER
```

GLW ! DOWNWARD LONG WAVE FLUX AT GROUND SURFACE
 GSW ! DOWNWARD SHORT WAVE FLUX AT GROUND SURFACE
 HFX ! UPWARD HEAT FLUX AT THE SURFACE
 HGT ! Terrain Height
 IVGTYP ! VEGETATION TYPE
 ISLTYP ! SOIL TYPE
 LU_INDEX ! LAND USE CATEGORY
 MAPFAC_M ! Map scale factor on mass grid
 MU ! Perturbation dry air mass in column
 MUB ! base state dry air mass in column
 MU0 ! initial dry mass in column
 Q2 ! QV at 2 M
 QFX ! UPWARD MOISTURE FLUX AT THE SURFACE
 RAINC ! ACCUMULATED TOTAL CUMULUS PRECIPITATION
 RAINCV ! TIME-STEP CUMULUS PRECIPITATION
 RAINNC ! ACCUMULATED TOTAL GRID SCALE PRECIPITATION
 SFROFF ! SURFACE RUNOFF
 slvl ! sea level pressure
 SMSTAV ! MOISTURE VARIBILITY
 SNOW ! SNOW WATER EQUIVALENT
 SNOWC ! FLAG INDICATING SNOW COVERAGE (1 FOR SNOW COVER)
 SST ! SEA SURFACE TEMPERATURE
 T2 ! TEMP at 2 M
 TH2 ! POT TEMP at 2 M
 TMN ! SOIL TEMPERATURE AT LOWER BOUNDARY
 TSK ! SURFACE SKIN TEMPERATURE
 U10 ! U at 10 M
 U10M ! U at 10 M - rotated
 V10 ! V at 10 M
 V10M ! V at 10 M - rotated
 UDROFF ! UNDERGROUND RUNOFF
 VEGFRA ! VEGETATION FRACTION
 WEASD ! WATER EQUIVALENT OF ACCUMULATED SNOW
 XLAT ! LATITUDE, SOUTH IS NEGATIVE
 XLONG ! LONGITUDE, WEST IS NEGATIVE
 XLAND ! LAND MASK (1 FOR LAND, 2 FOR WATER)
 ! below a list of fields from SI static and input files
 ALBBCK ! -
 LANDMASK ! -
 PMSL ! -
 SLOPECAT ! -
 SHDMAX ! -
 SHDMIN ! -
 SNOALB ! -
 SOILHGT ! -
 ST000010 ! -
 ST010040 ! -
 ST040100 ! -
 ST100200 ! -
 SM000010 ! -
 SM010040 ! -
 SM040100 ! -
 SM100200 ! -
 TOPOSTDV ! -
 TOPOSLPX ! -

```

TOPOSLPY      ! -
XICE          ! -
end_of_2dvar_list

                ! All list of files to read here
                ! Indent not to read
                ! Full path OK

/./DATA/wrfstatic_d01
/./DATA/wrfstatic_d02
/./DATA/wrf_real_input_em.d01.2000-01-24_12:00:00
/./DATA/wrf_real_input_em.d02.2000-01-24_12:00:00
/./DATA/real/wrfinput_d01

./wrfout_d02_2000-01-24_12:00:00
./wrfout_d02_2000-01-24_13:00:00
./wrfout_d02_2000-01-24_14:00:00
./wrfout_d02_2000-01-24_15:00:00
./wrfout_d02_2000-01-24_16:00:00
./wrfout_d02_2000-01-24_17:00:00
./wrfout_d02_2000-01-24_18:00:00
./wrfout_d02_2000-01-24_19:00:00
./wrfout_d02_2000-01-24_20:00:00
./wrfout_d02_2000-01-24_21:00:00
./wrfout_d02_2000-01-24_22:00:00
./wrfout_d02_2000-01-24_23:00:00
./wrfout_d02_2000-01-25_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
/./DATA/hill2d_x/wrfout_d01_0001-01-01_00:00:00
/./DATA/quarter_ss/wrfout_d01_0001-01-01_00:00:00
/./DATA/squall2d_x/wrfout_d01_0001-01-01_00:00:00
/./DATA/squall2d_y/wrfout_d01_0001-01-01_00:00:00
end_of_file_list

                ! Now we check to see what to do with the data
real          ! real (input/output) / ideal / static
1            ! 0=no map background in grads, 1=map background in grads
-1          ! specify grads vertical grid
                ! 0=cartesian,
                ! -1=interp to z from lowest h
                ! 1 list levels (either height in km, or pressure in mb)

1000.0
950.0
900.0
850.0
800.0
750.0
700.0
650.0
600.0
550.0
500.0
450.0
400.0
350.0
300.0

```

```
250.0  
200.0  
150.0  
100.0
```

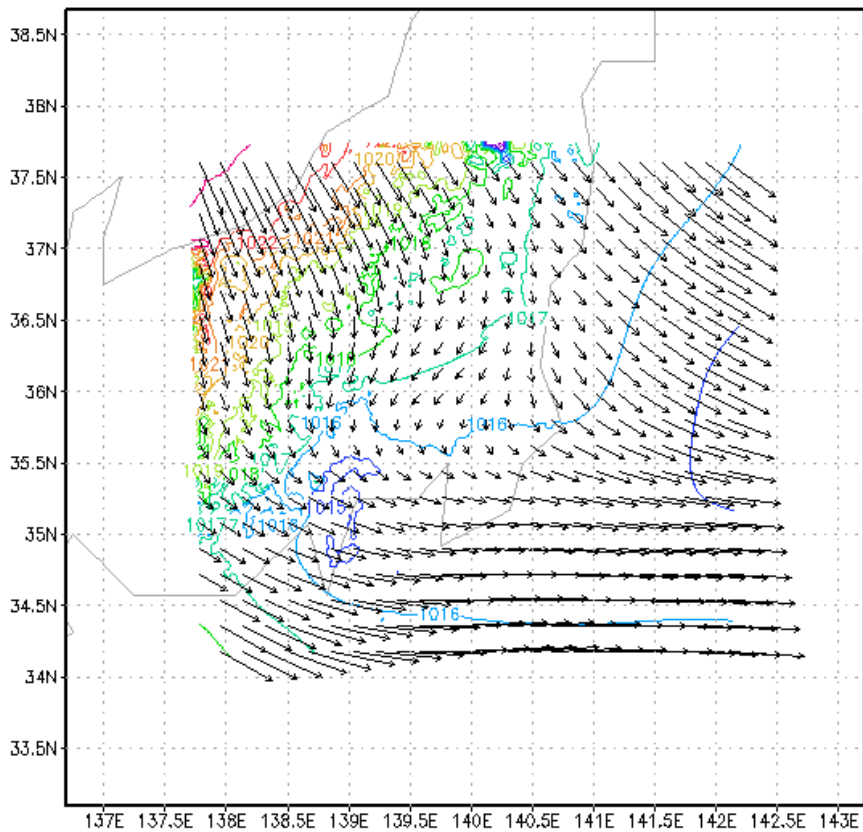
編集が終了したら、変換プログラムを走らせる。

```
$.wrf_to_grads control_file 20000124
```

出力が完了すると、GrADS形式の20000124.ctlと20000124.datが作成される。GrADSを起動し、出力を確認する。

```
$ grads  
ga-> open 20000124.ctl  
ga-> q file  
ga-> d slvl  
ga-> d skip(u10m,20,20);v10m
```

設定通りに行うと、以下のような画像が出力される。



GrADS: COLA/IGES

10

2009-03-21-16:52