

# NCEP's WRF POST PROCESSOR

Hui-Ya Chuang



# Outline

- Overview
- Sample fields generated by WRF post package
- Derivation of commonly used fields
- Break
- Installation
- Controlling what to output
- Visualization



# Introduction I

- NCEP's WRF post processor can post process model output from WRF NMM and WRF ARW.
- NCEP uses WRF post processor as the common post processor so that forecasts from different models can be compared and verified fairly.



# Introduction II - IO

- WRF post reads in model output in either binary or netcdf format using WRF IO API.
- Users are encouraged to use netcdf format. NCEP uses binary output for speed.
- Output is on NCEP standard or user-defined grids in NWS & WMO standard GRIB format, which can be read by NCEP's de-gribber, GEMPAK, or GrADS.



# Introduction III – Platform portability

- WRF post has been developed and run on NCEP's IBM with MPI.
- WRF post was also ported to run on linux thanks to Bob Rozumalski, Ligia Bernardet, Dusan Jovic and Meral Demirtas.



# Introduction IV – Different components of WRF post package

- Two components of WRF post package:
  - 1) wrfpost: a) perform vertical interpolations onto pressure and other levels (parallized for 3D computation) ; b) horizontally interpolate velocity onto mass points for ARW only; c) compute diagnostic fields.
  - 2) copygb: performs horizontal interpolations and de-staggering. Note that most graphics packages can not handle staggered grids.



# Introduction IV – Different components of WRF post package - Continued

- Computational domain of WRF NMM is on Arakawa-E rotated lat/lon, so copygb utility is needed to convert model output onto non-staggered grid.
- Computational domain of WRF ARW is on Arakawa-C grid. However, because wrfpost interpolates all velocity fields onto mass points for ARW and hence convert model output onto an A grid, no further horizontal interpolations are needed (i.e., no need to run copygb).



# Fields generated by WRF post package

- WRF post package currently outputs 288 fields. Complete list can be found in Table 1 of your user guide (p7-8) or online:

<http://wwwt.emc.ncep.noaa.gov/mmb/papers/chuang/2/wrfpost.txt>

- Sample fields generated by WRF post package:
  - 1) Temperature, height, humidity, 3D wind, turbulent kinetic energy, exchange coefficients, cloud water, cloud ice, rain, and snow on 47 isobaric levels (8 levels above 75 mb and then from 75 to 1000 mb every 25 mb);
  - 2) Shelter level temperature, humidity, and wind fields;





# Fields generated by WRF post package

## - Continued

- 3) Accumulated and instantaneous precipitation: total, convective, and grid scale;
- 4) Radar reflectivity, visibility, and precip types;
- 5) Vorticity and geostrophic stream function;
- 6) PBL height; 6 layers of PBL AGL 30 mb layer-averaged temperature, humidity, and wind;
- 7) Surface wind stress, drag coefficient, roughness length, friction velocity;
- 8) SLP (two types);



# Fields generated by WRF post package

## - Continued

9) Instantaneous and time-averaged surface fluxes: sensible, latent, ground, downward and upward shortwave and longwave;

10) Soil temperature, moisture, and types;

11) Cloud fraction as well as cloud top/bottom pressure, height, and temperature for total, convective, and grid-scale;

12) Aviation products including in-flight icing and ceiling;

13) Brightness temperature.



# Computation of atmospheric isobaric fields

- Vertical interpolation of height, temperature, specific humidity, vertical velocity, horizontal winds, and turbulent kinetic energy from model level to pressure level fields is **linear in  $\ln(p)$** .
- WRF NMM model does not output height fields, so the WRF post processor derives WRF NMM model level heights by integrating virtual temperature hydrostatically from bottom up.



# Computation of underground isobaric fields

- Underground vertical and horizontal wind components are specified to be the same as those at the first atmospheric model layer above ground.
- Underground temperature is reduced by assuming constant virtual potential temperature from the temperature averaged over the second and the third model levels above the surface.
- Underground humidity fields are computed so as to maintain RH averaged over the second and third model levels from the ground.



# Derivation of sea level pressure type I: standard NCEP SLP

- Ground and sea level temperatures are extrapolated from the temperature at the lowest atmospheric layer by assuming a constant lapse rate of 6.5 K/KM.
- Compute  $\tau = R_d T_v / g$  at ground and sea level and then apply Shuell correction to both  $\tau$ s. The basic principal of Shuell correction is to make sure that  $\tau$  at both sea level and ground do not exceed a critical value.
- Standard NCEP SLP is then derived as follows:

$$SLP = PSFC \times \exp\left(\frac{2 \times HSFC}{\tau(SFC) + \tau(SLP)}\right)$$



# Derivation of sea level pressure type II: membrane NCEP SLP

- Compute underground virtual temperatures by horizontally relaxing virtual temperatures on pressure levels:

$$\nabla^2 T_v = 0.$$

- The nine-point successive over-relaxation formula is used to solve the above Laplace's Eq. numerically.
- Once all underground virtual temperatures are generated, the hydrostatic equation is integrated downward to obtain sea level pressure.



# Computation of simulated radar reflectivity

- Two algorithms are used to compute simulated radar reflectivity depending on the microphysics (MP) option used in the model run:
  - 1) Ferrier MP scheme: derived by Ferrier to be consistent with assumptions made in Ferrier MP scheme. A maximum of 80 dBZ and a minimum of -20 dBZ are applied. More information can be found in Ferrier's 94 JAS publication;
  - 2) Other MP schemes: adopted from RIP4. More information can be found online:

<http://www.mmm.ucar.edu/wrf/users/docs/ripug.htm>



# Derivation of visibility

- Warner-Stoelinga algorithm is used to compute visibility for all cores and physics packages.
- This algorithm first computes extinction coefficients ( $\beta$ ) for each hydrometeor species:

$$\beta = A \times C^\lambda$$

,where  $A$  and  $\lambda$  are empirical coefficients. All the  $\beta$ s are summed to yield a single  $\beta$ . The visibility is then calculated using the formulation:

$$vis = \min(20km, -\ln \varepsilon / \beta)$$





# Shelter level fields and PBL height

- Shelter level fields and PBL height are direct output from WRF model, not interpolated or diagnosed in the WRF post.
- This ensures that these fields are derived within the model based on surface and PBL physics consistent with your model runs.



# Computation of other fields

- Computation of many other fields can be found in ETA post documentation online:

<http://www.emc.ncep.noaa.gov/mmb/papers/chuang/1/OF438.html>

- Keep in mind that there are some differences between WRF and ETA post when looking through documentation. In addition to differences in vertical coordinate, they are mostly how to ingest model output and set up constants. One example is that WRF post no longer reads in namelist FCSTDATA.
- Eta post documentation will soon be updated to become WRF post documentation.



# Download

- The tar file wrfpost\_v2.0.tar containing all the source code, scripts, and libraries is available via DTC site:  
<http://www.dtcenter.org/wrf-nmm/users/downloads>
- Un-tarring the tar file creates 4 directories, a configure file, and two master makefiles for two supported platforms:
  - 1) src/: source code;
  - 2) scripts/: sample scripts for running post and graphics packages;
  - 3) lib/: libraries used by wrfpost and copygb;
  - 4) parm/: control files used by wrfpost;
  - 5) configure: to set up proper makefiles based on user's platform and path names for IO libs;
  - 6) makefile: to compile all the lib and src.



# Installation I – Compile source codes

- Configure your makefiles by executing the file configure. Users will be prompted to specify:
  - 1) platform: enter “1” for LINUX or “2” for IBM;
  - 2) path name of your netcdf utility;
  - 3) path name of your WRF model source code.
- Compile all the libraries and source codes by executing the master makefile in the top directory.



# Installation II – Run wrfpost

- wrfpost needs three input files:
  - 1) itag: read in via unit 5 to provide information on
    - a) model output file name in first line,
    - b) format of model output in second line (netcdf or binary),
    - c) forecast verifying time in WRF format in third line,
    - d) model name in fourth line (NMM or NCAR);
  - 2) wrf\_cntrl.parm: control file to let users specify which fields to output;
  - 3) eta\_micro\_lookup.dat: look-up table containing MP coefficients used by Ferrier scheme.



# Installation III – Run wrfpost – Sample script run\_wrfpost

```
#!/bin/sh  
set -aeux  
export tmmark=tm00
```

*creating file itag in the script*

```
cat > itag <<EOF  
wrfout_d01_2005-04-27_00:00:00  
netcdf  
2005-04-27_00:00:00  
NMM  
EOF
```

← *file name of WRF history file*

← *format of WRF output*

← *validation time*

← *model name (NMM or NCAR)*

```
rm -f fort.*
```

```
ln -sf ../parm/wrf_cntrl.parm fort.14
```

← *linking to control file*

```
ln -sf griddef.out fort.110
```

```
ln -sf ../parm/eta_micro_lookup.dat .
```

← *linking to Ferrier's lookup table*

```
../exec/wrfpost.exe < itag > outpost_wrf
```

← *execute WRF post*

# Installation IV – Description of wrfpost control file wrf\_cntrl.parm

*specifying grid number*

KGTYPE\*\*\*\*\*15\*\*\*\*\*:(00255)\*\*\*\*\*START OF THIS OUTPUT

IMDLTY \*15\* :(00089)

DATSET \*A6\* :(WRFPRS)

*GRIB packing precision*

(PRESS ON MDL SFCS ) SCAL=( 3.0)

L=(11000 00000 00000 00000 00000 00000 00000 00000 00000

(HEIGHT ON MDL SFCS ) SCAL=(-5.0)

L=(11000 00000 00000 00000 00000 00000 00000 00000 00000

*switch to specify which level of field to output with "1" being yes*

*abbreviated name used in post source code for each field*

# Installation V – Use wrfpost control file to output a field

- To output a desired field:
  - 1) look through field names in Table 1 of WRF post user guide (p7-8) to see if WRF post produces this field;
  - 2) If yes, note the corresponding abbreviated name in the 2<sup>nd</sup> column of the Table 1 and look for it in wrf\_cntrl.parm;
  - 3) If it is already listed in wrf\_cntrl.parm, make sure that the switch is turned onto “1”;
  - 4) If it is not listed in wrf\_cntrl.parm, add this field to the control file. To add land/sea mask:

```
(LAND SEA MASK ) SCAL=( 3.0)
```

```
L=(10000 00000 00000 00000 00000 00000 00000 00000 00000)
```





# Installation VI – Use wrfpost control file to output fields on multiple levels

- wrfpost outputs fields on many different types of vertical coordinates:
  - 1) native model vertical levels;
  - 2) 47 pressure levels: 2, 5, 7, 10, 20, 30, 50, 70 mb, then 75 to 1000 mb every 25 mb;
  - 3) 7 flight levels above MSL in m: 914, 1524, 1829, 2134, 2743, 3658, 6000;
  - 4) 6 layers of 30 mb averaged PBL layers;
  - 5) 2 AGL level: 1000 and 4000 m for radar reflectivity.
- Except for AGL and pressure levels, all the other vertical levels are counted from bottom to top in wrf\_cntrl.parm.



# Installation VI – Use wrfpost control file to output fields on multiple levels - Cont

- To output temperature fields at 75 and 125 mb:

(TEMP ON PRESS SFCS ) SCAL=( 3.0)

L=(00000 00010 10000 00000 00000 00000 00000 00000 00000)

- To output 30 mb PBL mean U from 30 to 60 mb and then from 90 to 120 mb AGL:

(U WIND IN BNDRY LYR ) SCAL=( 3.0)

L=(01010 00000 00000 00000 00000 00000 00000 00000 00000)



# Installation VII – Change the number or values of output pressure levels

- Modify specification of variable LSM in the file CTLBLK.comm to change the number of pressure levels:  
PARAMETER (LSM=47)
- Modify specification of SPL array in the subroutine POSTDATA.f to change the values of pressure levels:

```
DATA SPL/200.,500.,700.,1000.,2000.,3000.  
&,5000.,7000.,7500.,10000.,12500.,15000.,17500.,20000.
```



# Installation VIII – run copygb

- To use copygb to perform horizontal interpolations:

```
copygb -xg"${grid}" in.grb out.grb
```

Two ways to specify  $\{\text{grid}\}$  :

1)  $\{\text{grid}\}$  = NCEP standard grid number:

```
copygb -xg212 in.grb out.grb
```

2)  $\{\text{grid}\}$  = '255 INT(18), where INT(18) is an array containing user-defined grid navigation information:

```
copygb -xg"255 3 109 91 37748 -77613 8 -71000 10379 9900 0  
64 42000 42000 " in.grb out.grb
```



# Installation VIII – run copygb - Continued

- To find a NCEP standard grid that matches your output domain, look up the grid specs for all NCEP grids online: <http://www.nco.ncep.noaa.gov/pmb/docs/on388/tableb.html>
- Details on how to specify INT(18) for a user-defined grid can be found in the subroutine w3fi63 in your lib/w3lib.
- Examples on how to use copygb to interpolate onto a user-defined grid is shown in the sample scripts run\_wrfpostandgempak and run\_wrfpostandgrads .
- The instructions on how to run copygb can be found in the file copygb.doc that comes with the copygb source code.



# GRIB file visualization with GEMPAK

- GEMPAK has an utility named “nagrib” that reads GRIB files on any non-staggered grids and then generates GEMPAK-binary files that are readable by all GEMPAK programs
- GEMPAK can plot horizontal contours, cross-sections, meteograms, and sounding profiles.
- Package download and user guide are available online:

<http://my.unidata.ucar.edu/content/software/gempak/index.html>

- A sample script named run\_wrfpostandgempak is included in scripts/ that can be used to run wrfpost, copygb, and then plot various fields using GEMPAK.



# Sample script run\_wrfpostandgempak

```
nagrrib<< EOF
GBFILE=wrfnmm${fhr}.tm00
GDOUTF=wrfnmm${fhr}.grd
MAXGRD=3000
CPYFIL=gds
OUTPUT=t
run
exit
EOF
```

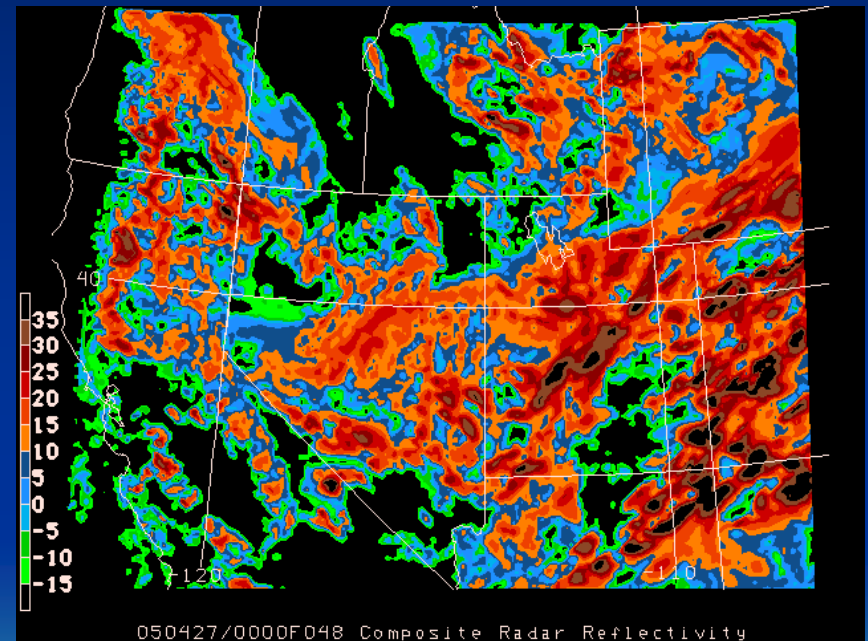
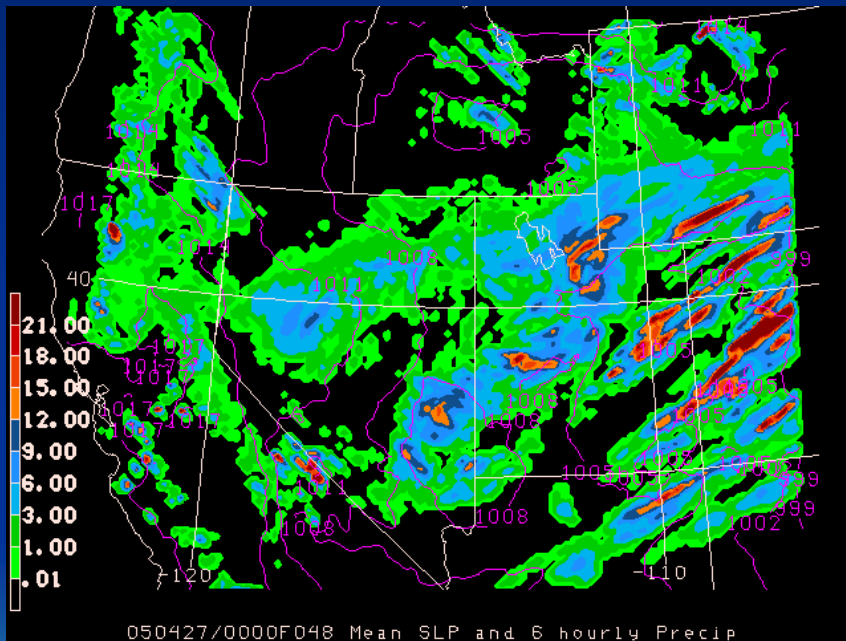
converting GRIB to GEMPAK binary

```
gdplot<< EOF
GDFILE = wrfnmm${fhr}.grd
GDATTIM = LAST
GLEVEL = 0
GVCORD = none
GFUNC = p06m
CTYPE = f
DEVICE = GIF|Sfcmmap${fhr}.gif
```

plot 6hr precipitation

save the plot to GIF file

# WRF NMM forecast plotted with GEMPAK : Precipitation and derived Radar reflectivity





# GRIB file visualization with GrADS

- GrADS also has utilities to read GRIB files on any non-staggered grids and then generates GrADS control files. The utilities grib2ctl and gribmap are available via Wesley Ebisuzaki's web site:

<http://www.cpc.ncep.noaa.gov/products/wesley/grib2ctl.html>

- Package download and user guide for GrADS are available online:

<http://grads.iges.org/grads/gadoc/>

- A sample script named run\_wrfpostandgrads is included in scripts/ that can be used to run post, copygb, and then plot various fields using GrADS.



# Sample script run\_wrfpostandgrads

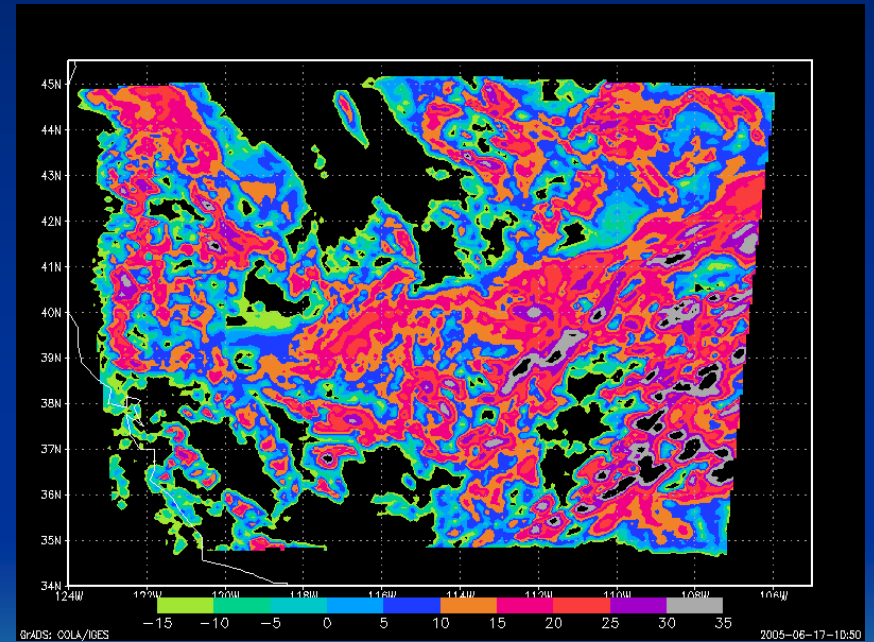
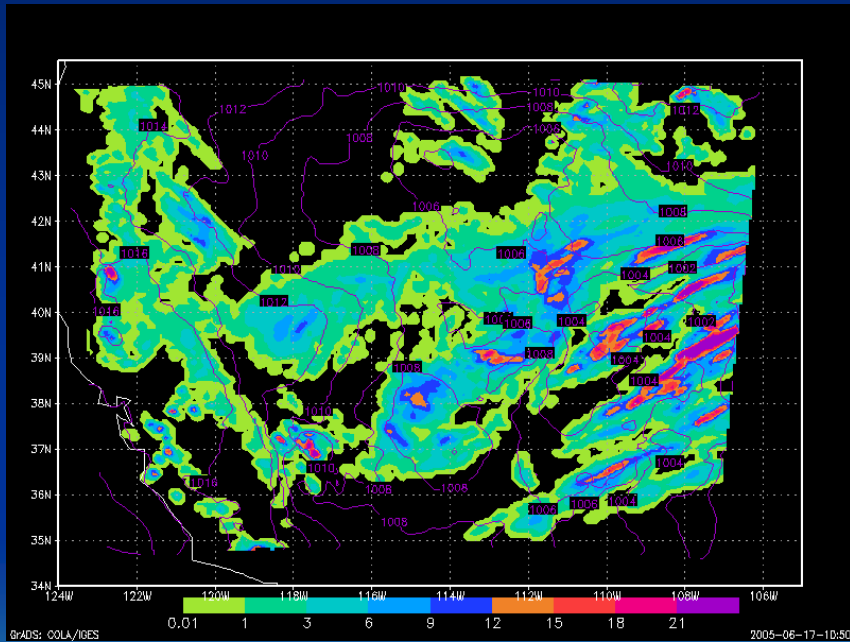
```
grib2ctl.pl -verf wrfnmm${fhr}.tm00 > wrfnmm${fhr}.ctl ← create ctrl file  
gribmap -i wrfnmm${fhr}.ctl ← create index file
```

```
cat > plotgrads << EOF  
'open wrfnmm${fhr}.ctl'  
'set gxout shaded'  
'd APCPsfc' ← plot precipitation  
'cbar'  
'printim Sfcmap${fhr}_GrADS.gif gif' ← save plot to gif
```

```
xgrads -blc "run plotgrads"
```



# WRF NMM forecast plotted with GrADS: Precipitation and derived Radar reflectivity



# WRF NMM and ARW fields ingested by WRF post

- A list of fields (as named in Registry file) that are read in by WRF post for both WRF NMM and WRF ARW can be found in tables 2 (p7-15) and 3 (p7-16) in your user guide or online:

<http://wwwt.emc.ncep.noaa.gov/mmb/papers/chuang/2/wrfpost.txt>

- It is important to make sure that users have all these fields in the model output so that WRF post can compute and output each field properly.
- To have these fields in your model output, users will need to modify Registry file and then re-compile WRF model source code. John will talk about how to modify Registry file.



# Some tips and suggestions

- To reduce the size of the GRIB file, users can modify the control file `wrf_cntrl.parm` as mentioned earlier to only output desired fields.
- If a field in your GRIB file does not have physical values, it is likely that you don't have required fields in your model output. For example, if your vorticity fields look unreasonable, you may not have velocity or grid resolution fields in your model output.



# Future plan

- WRF post processor is constantly updated to add more fields and to make bug fixes. NCEP and DTC will work together to distribute the new version to users ASAP.
- WRF post will soon be modified to post process GFS output to achieve the goal of having a common post processor at EMC.



# Observed and simulated brightness temperature

