NCEP's UNIFIED POST PROCESSOR (UPP)

Hui-Ya Chuang
(Hui-Ya.Chuang@noaa.gov)

Presented by Matthew Pyle
(Matthew.Pyle@noaa.gov)

Outline

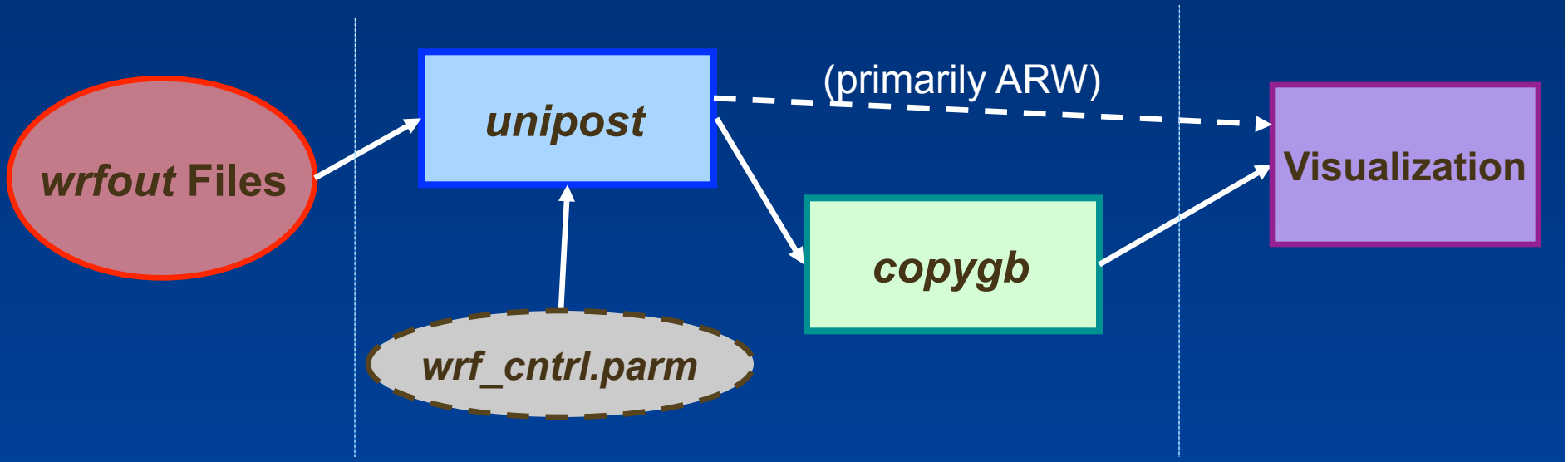
- Overview
- Components and Functions
- Sample fields generated
- Installation
- Running *unipost*
  - Controlling output generation
- Running *copygb*
  - Specifying target grid
- Visualization

The critical big picture overview

- Processes model output from both the NMM and the ARW dynamical cores.
- The Unified Post Processor (UPP) generates output in GRIB.
- The UPP enables product generation on any output grid.

Components of the UPP

The UPP has two components: unipost and copygb.



Functions and features of *unipost*

- Performs **vertical** interpolation onto isobaric and other non-model surfaces
- Computes diagnostic fields
- Destaggers wind onto mass points (ARW)
- An MPI-parallel code

Functions of *copygb*

- Performs destaggering (NMM only) and **horizontal** interpolation to a defined output grid
  - NOTE: many visualization packages cannot properly handle staggered grids
- Creates an output grid different than the model integration domain.

Ingesting WRF model output

- The unipost ingests WRF model output in netCDF or binary format using the WRF I/O package.
  - Users are encouraged to use netCDF-formatted model output for simplicity.
  - Binary I/O is quicker for large file sizes. DTC is working to improve binary support using MPI-IO.
  - One time per output file is best w/ sample UPP run scripts (frames_per_outfile=1 in WRF model namelist).

Ingesting WRF model output

- By default the WRF model will provide all fields that the unipost needs to ingest.
- The users' guide lists the fields read in by the unipost for both dynamical cores (by WRF Registry file variable names)
- Not a concern unless modifying the Registry.

Fields generated by the UPP

- The UPP currently outputs hundreds of possible fields.
  - Complete list in the Post Processing Utilities Chapter of the user guide
- Sample fields generated by UPP:

1) T, Z, humidity, wind, cloud water, cloud ice, rain, and snow on isobaric levels
2) SLP + shelter level T, humidity, and wind fields
3) Precipitation-related fields

Fields generated by the UPP

- Sample fields generated by UPP (cont.):

4) PBL-related fields
5) Diagnostic products
6) Radiative/Surface fluxes
7) Cloud related fields
8) Aviation products
9) Satellite look-alike products

UPP download and compile

Downloading the UPP source code

- The UPP source code can be obtained from:
  http://www.dtcenter.org/wrf-nmm/users/downloads
- The latest version available is:
  UPPV2.0.tar.gz
- Unpack the downloaded file:
  tar –zxvf UPPV2.0*.tar.gz*
- *cd* to newly created UPPV2.0/ directory

UPPV2.0 directory “important” contents

- **scripts/**: sample scripts for running UPP and generating graphics
- **parm/**: contains the files used to request output fields when running the unipost
- **clean, configure, compile**: scripts used in the build process

Compile source codes

- The build mechanism* follows the WRF model build paradigm:

./configure : respond to screen prompts about target computing platform

./compile >& compile_upp.log

* This build relies on the existence of a built WRF source directory

Compile source codes (cont.)

- If compilation is successful, these three executables will be present in exec/ :

  copygb.exe
  ndate.exe
  unipost.exe

- Currently have build options established for IBM (AIX) and Linux (PGI/Intel/Gnu compilers)

- The arch/configure.defaults file has compilation options for various platforms, and is where new computers or compilers might be added.

Running unipost and copygb

unipost needs three input files to run

In the sample run_unipost* scripts, these files are automatically generated or linked:

- *itag*: specifies details about WRF model output to process

wrfout_d01_2010-06-27_00:00:00 ← *WRF history filename*

netcdf ← *WRF output format (netcdf/binary)*

grib2 ← *extra line only if writing GRIB2 – currently broken so don't try this yet!*

2010-06-27_00:00:00 ← *validation time*

NMM ← *model name (NMM/NCAR)*

- *wrf_cntrl.parm*: control file specifying fields/levels to output in GRIB1

- *eta_micro_lookup.dat*: binary look-up table for Ferrier MP

unipost control file: *wrf_cntrl.parm (GRIB1)*

- Users specify which fields or which level(s) of fields to output by modifying control file, e.g.,

GRIB packing precision**

```
(PRESS ON MDL SFCS  ) SCAL=(6.0)
L=(11000 00000 00000 00000 00000 00000 00000…
(HEIGHT ON MDL SFCS ) SCAL=(6.0)
L=(11000 00000 00000 00000 00000 00000 00000…
```

Each column represents a single model/isobaric level:
"1" (or "2" - special case) = output, "0" = no output

Product description – unipost code keys on these character strings.

** larger values → more precision, but larger GRIB files.

unipost control file: *wrf_cntrl.parm*

- The included *wrf_cntrl.parm* file has entries for every possible output field.
- The users' guide "Fields produced by *unipost*" table more fully explain the character string abbreviations used in the control file.

- Generation of GRIB2 remains a work in progress (i.e., it isn't yet working correctly), but uses an XML file to request fields.

Outputting fields on different vertical coordinates

- *unipost* outputs on several vertical coordinates:
  - Native model levels
  - 47 isobaric levels
  - 15 flight/wind energy levels: 30, 50, 80, 100, …, 2743, 3658, 4572, 6000 m (above ground or above MSL)
  - 6 PBL layers: each averaged over a 30 hPa deep layer
  - 2 AGL radar levels: 1000 & 4000 m
- Except for AGL radar and isobaric levels, vertical levels are listed from the ground surface up in *wrf_cntrl.parm*.

Examples

- Output T every 50 hPa from 50 hPa to 1000 hPa:

(TEMP ON PRESS SFCS) SCAL=(4.0)

L=(00000 01001 01…)

2 5 7 10 20 30 50 70 75 100 125 150

Isobaric levels increase from left to right: 2, 5, 7, 10, 20, 30, 50, 70, then every 25 hPa from 75-1000 hPa.

Isobaric levels every 50 hPa:

L=(00000 01001 01010 10101 01010 10101 01010 10101 01010 10000 00000 00000 00000 00000)

Isobaric levels every 25 hPa:

L=(00000 01011 11111 11111 11111 11111 11111 11111 11111 10000 00000 00000 00000 00000)

Examples

- Output instantaneous surface sensible heat flux:

```
(INST SFC SENHEAT FX ) SCAL=( 4.0)
L=(10000 00000 00000 00000 00000 00000 00000 00000 00000 00000…)
```

- Output the U-wind component at the 5 lowest model levels:

```
(U WIND ON MDL SFCS  ) SCAL=( 4.0)
L=(11111 00000 00000 00                                    00000 00000…)
```

- Output U-wind com AGL:

```
(U WIND AT FD HEIGHT) S
L=(22200 00000 00000 00                                    00000 00000…)
```

For the flight/wind energy level fields:
- "2" requests AGL.
- "1" requests above mean sea level.

copygb target grid definition

- The generic command to run copygb and horizontally interpolate onto a new grid is:

  copygb.exe –xg"${grid}" in.grb out.grb

- Three options on how to specify the target $grid:

  1. Pre-defined NCEP standard grid number
  2. Grid navigation file created by *unipost* (NMM only)
  3. User-defined grid definition

Run *copygb* – Option 1

- Interpolate to a pre-defined NCEP standard grid (restrictive but simple)
  - For example, to interpolate onto NCEP grid 212:

    copygb.exe –xg212 in.grb out.grb

Descriptions of NCEP grids are available online:

http://www.nco.ncep.noaa.gov/pmb/docs/on388/tableb.html

Run *copygb* – Option 2

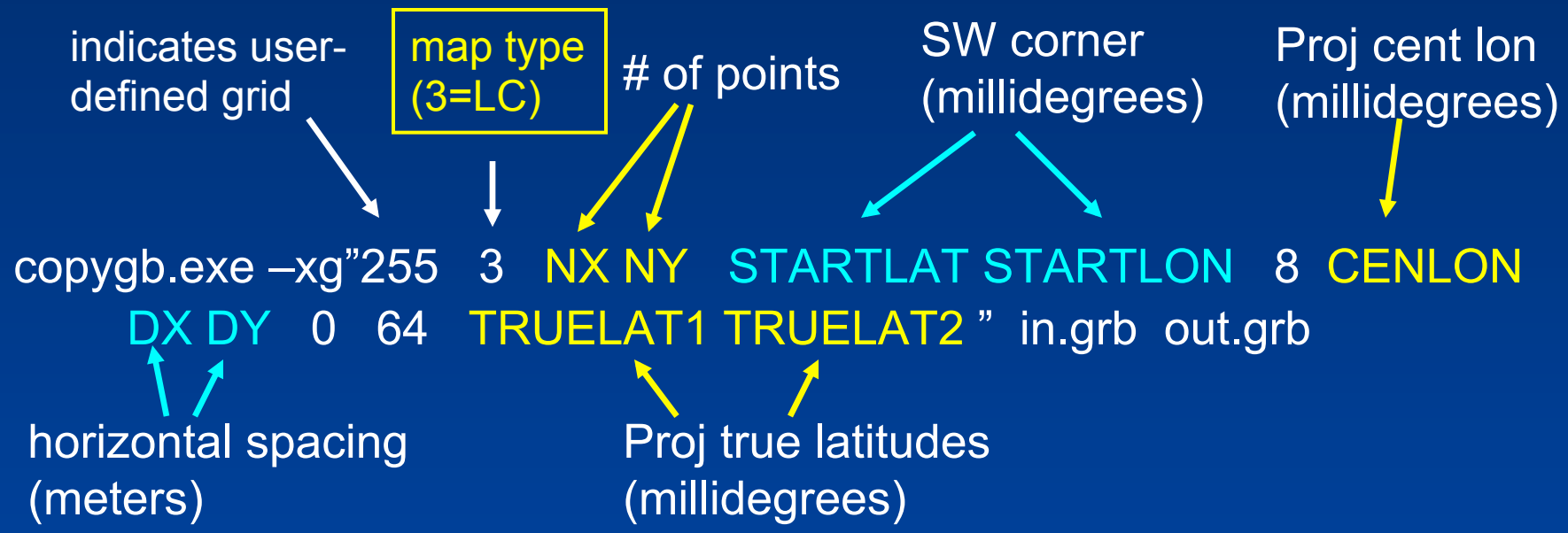
- Read in grid navigation file created by *unipost* (NMM only, simple, restrictive)
  - Running *unipost* on WRF-NMM output produces two ASCII files containing grid navigation information which is similar in domain and grid spacing to the model integration domain.
    - *copygb_gridnav.txt* for a Lambert Conformal grid
    - *copygb_hwrf.txt* for a regular Lat-Lon grid

For example:

```
read nav < 'copygb_gridnav.txt'
copygb.exe –xg”${nav}” in.grb out.grb
```

Run *copygb* – Option 3a

- Create a user-defined Lambert Conformal grid by specifying a full set of grid parameters (complicated but flexible).




```
copygb –xg"255  3  185  129  12190      -133459       8   -95000
    40635 40635  0  64  25000      25000"       in.grb  out.grb
```

Run *copygb* – Option 3b

- Create a user-defined Polar Stereographic grid by specifying a full set of grid parameters (complicated but flexible).

map type (5=STR)

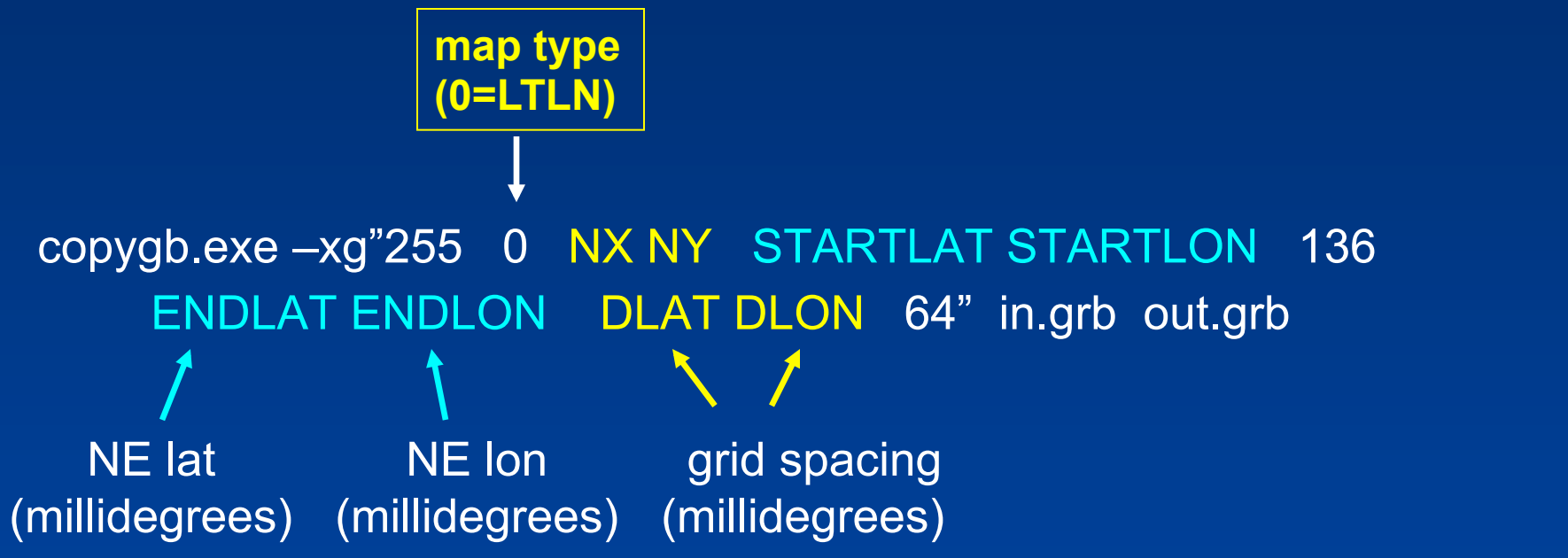
copygb.exe –xg"255 5 NX NY STARTLAT STARTLON 8 CENLON
DX DY 0 64" in.grb out.grb

Center flag (0=NH ; 1=SH)

copygb –xg"255 5 580 548 10000 -128000 8 -105000
15000 15000 0 64" in.grb out.grb

Run *copygb* – Option 3c

- Create a user-defined Latitude-Longitude grid by specifying a full set of grid parameters (complicated but flexible).



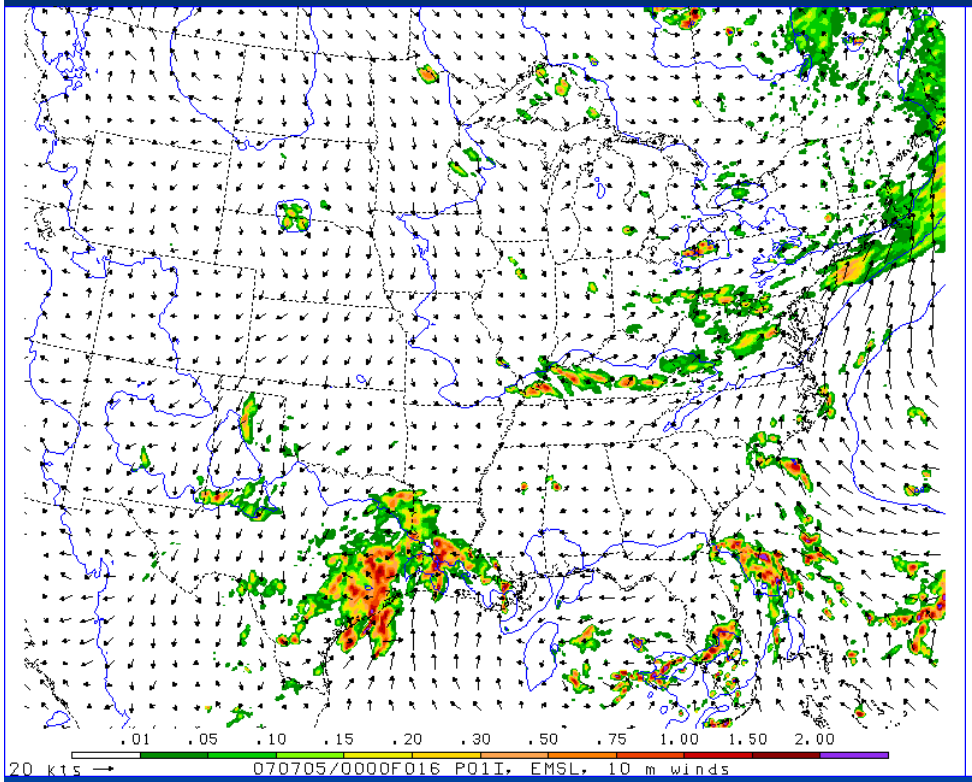

```
copygb –xg"255 0 401 401 10000 -130000 136
       50000 -90000 100 100 64" in.grb out.grb
```

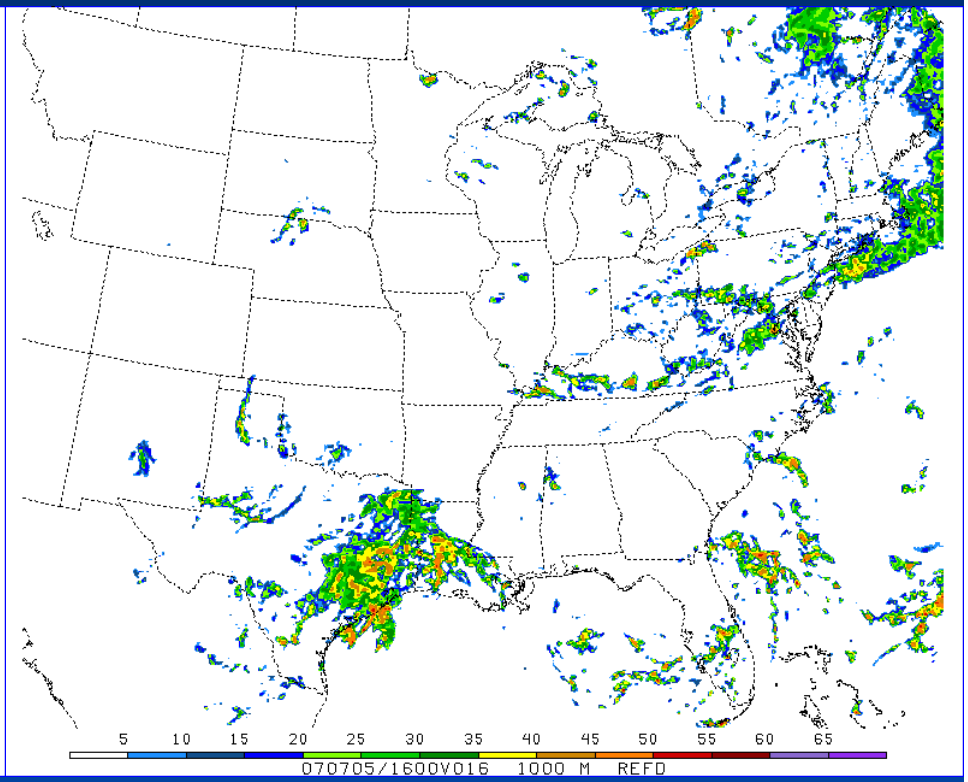
GRIB file visualization with GEMPAK

- The GEMPAK utility “nagrib” reads GRIB files from any non-staggered grid and generates GEMPAK-binary files that are readable by GEMPAK plotting programs
- GEMPAK can plot horizontal maps, vertical cross-sections, meteograms, and sounding profiles.
- Package download and user guide are available online: http://www.unidata.ucar.edu/content/software/gempak/index.html
- A sample script named *run_unipostandgempak* is included in scripts/ that can be used to run *unipost*, *copygb*, and then plot various fields using GEMPAK.
- Further details on this script and using GEMPAK are available in the user’s guide.

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 generate GrADS “control” files. The utilities grib2ctl and gribmap are available via: http://www.cpc.ncep.noaa.gov/products/wesley/grib2ctl.html

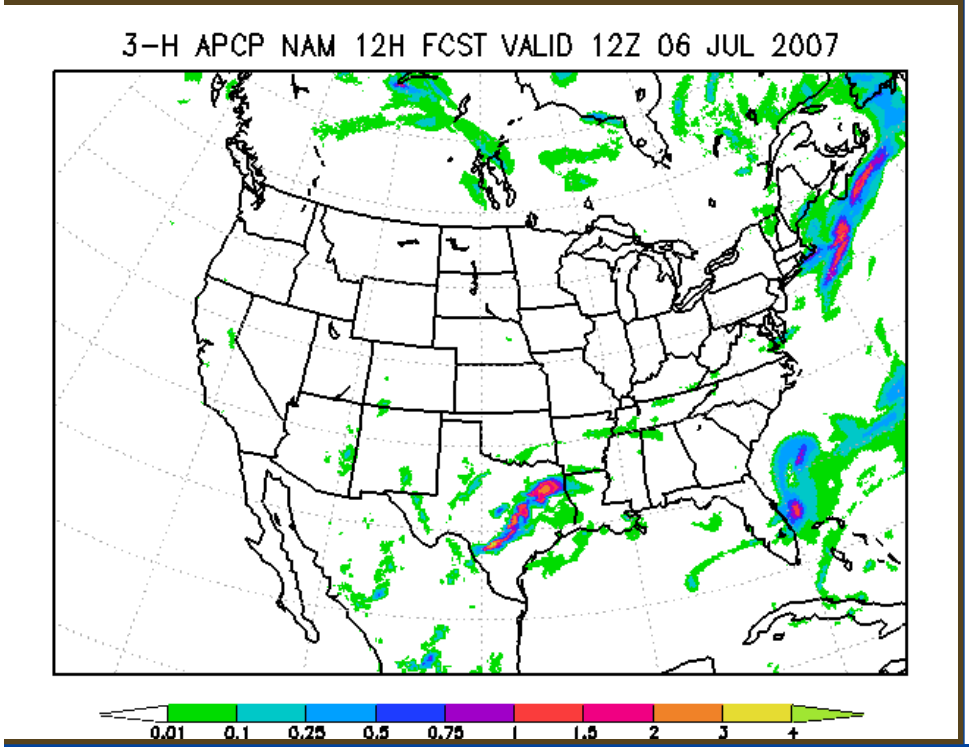
- Package download and user guide for GrADS are available online:

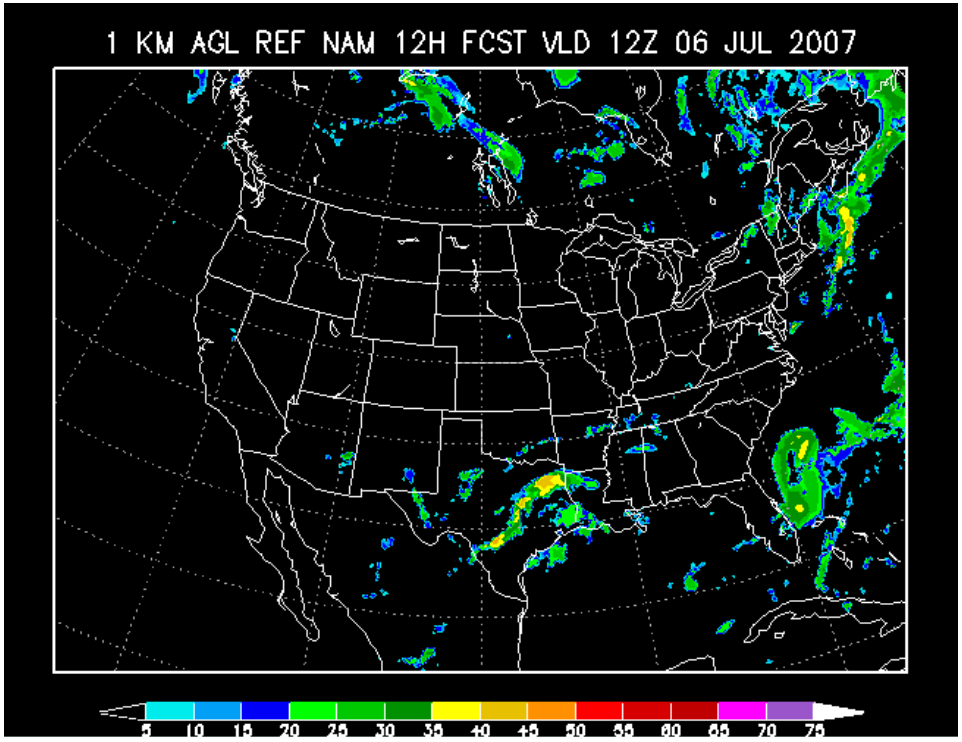
  http://grads.iges.org/grads/gadoc/

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

Forecast plotted with GrADS:

Precipitation and derived Radar reflectivity





Future plans

- Fix problem(s) in the generation of GRIB2 output.
- Continue adding new products to the released UPP code as they are developed, and expand code portability.

Questions???