Introduction to the New WRF Preprocessing System

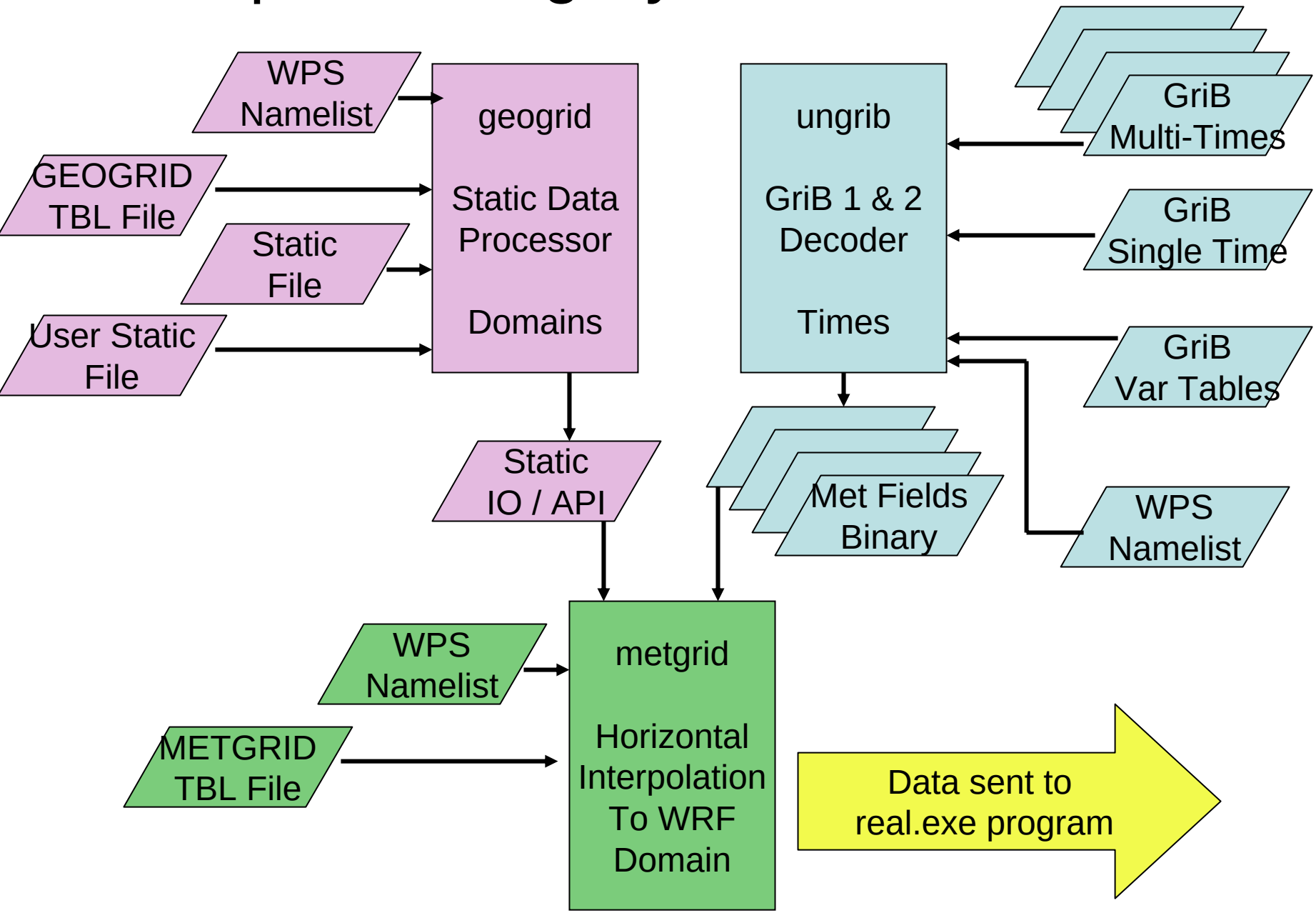
Michael Duda
James Bresch
Dave Gill

Wei Wang
Jordan Powers
Jimy Dudhia
Kevin Manning

WRF Preprocessing System (WPS) Key Design Goals

- Ease of use
  - Avoid command-line parameters and environment variables
  - All configuration through namelists and tables
- Expandability and flexibility
  - Handle new data sets
  - Allow for flexibility in how data are interpolated
- Large domain performance – time and memory
  - Timing should scale favorably with domain size
  - Must be able to process large domains
- Use WRF I/O API for input and output of intermediate files
- Work with either ARW or NMM cores

Preprocessing System Overview



Outline of Topics

geogrid

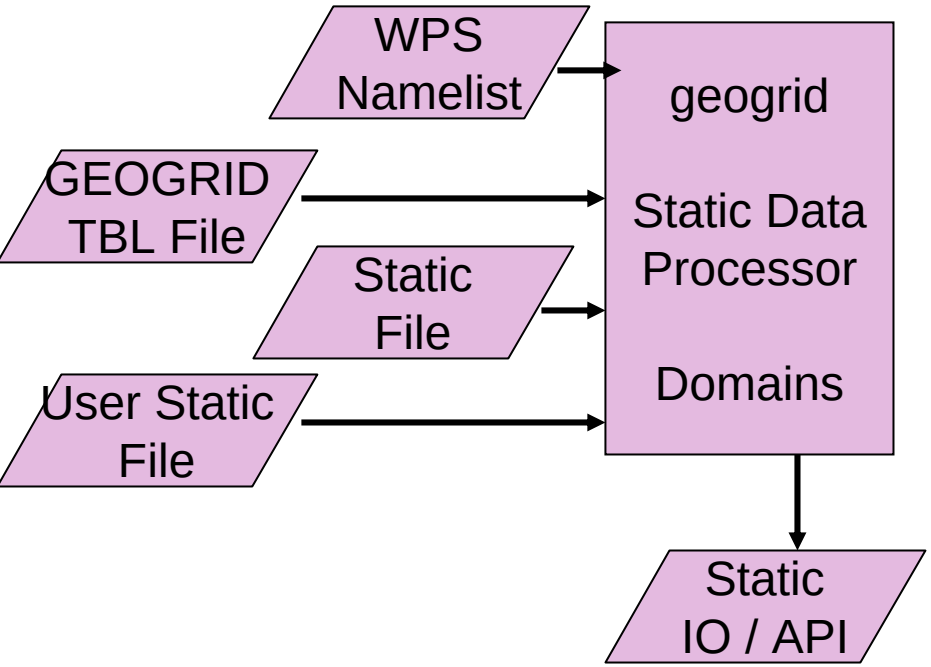
- purpose
- installing and running (live demo!)
- timing and memory performance
- output field comparison: SI and TERRAIN
- importing new data sources
- Domain Wizard

ungrib

metgrid

real

The geogrid program



geogrid is the WPS counterpart to grid_gen (SI) and TERRAIN (MM5)

Purposes of geogrid

- Define dimensions and location of model grids
- Interpolate static geographical data to grids

Outline of Topics

geogrid
- purpose
- installing and running (live demo!)
- timing and memory performance
- output field comparison: SI and TERRAIN
- importing new data sources

ungrib

metgrid

real

Steps to install and run geogrid

1. Obtain static geographical data
2. Configure and compile geogrid
3. Edit namelist to define parent domain and nests
4. (OPTIONAL!) Edit GEOGRID.TBL to modify handling of data – only needed for special circumstances
5. Run geogrid.exe to produce geo_em (or geo_nmm) files

Steps 3 & 5 will generally only be used after initial installation and setup

After running geogrid.exe, domain files appear in current directory (or, if specified, an alternate directory)

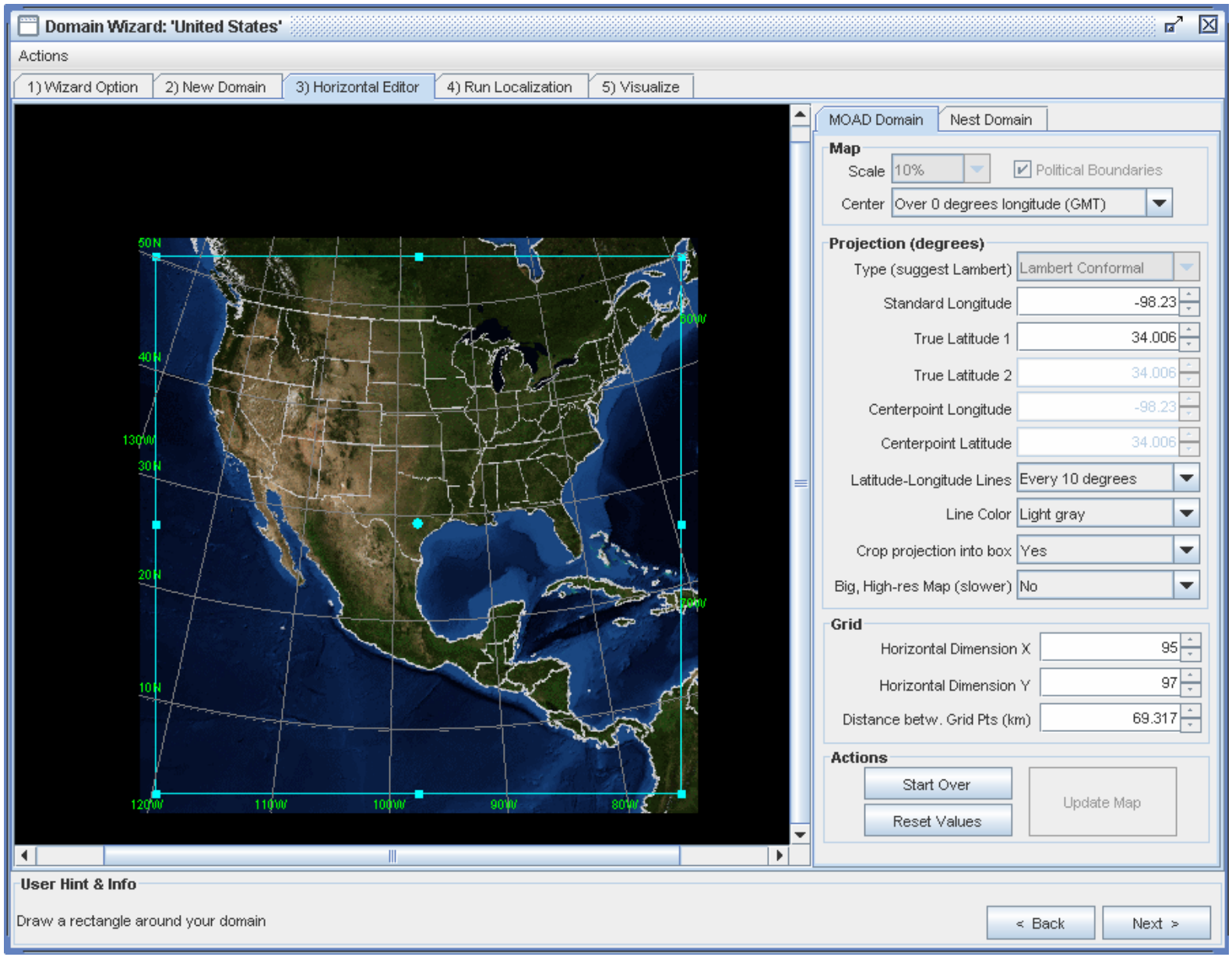
GEOGRID.TBL can be used to tune the way that input data are handled for a particular domain

- *Typically, the user should not need to modify this file!*

Domain Wizard

- Developed by Jeff Smith and Paula McCaslin at ESRL
- Provides graphical interface to WPS
- Java-based; can run from website or as standalone program
- www.wrfportal.org
- Flash demonstration at www.wrfportal.org/flash/domainwiz_demo.html

Domain Wizard (cont.)



Outline of Topics

geogrid
- purpose
- installing and running (live demo!)
- timing and memory performance
- output field comparison: SI and TERRAIN
- importing new data sources

ungrib

metgrid

real

Running geogrid: timing performance*

Grid Descr.	# Doms	Sizes	DX	grid_gen (mm:ss)	geogrid (mm:ss)
Gulf Coast	1	400x250	4 km	33:21	01:04
T92 (CONUS)	1	355x223	15 km	26:11	00:31
Antarctica + southern ocean	2	165x218 331x313	60 km 20 km	84:55	01:05
South-eastern US	1	1480 x1220	1.5 km	--:--	05:53

** tests run on a PC running FC4; 2 GB mem; 3.4 GHz Xeon; PGI 6.1 compilers*

geogrid: Distributed memory capable

- geogrid may be run on a distributed memory platform
  - Compile with MPI libraries
- Two possibilities for io_form_output
  - Ex: io_form_output=102 – one output file per processor, NetCDF
  - Output fields identical regardless of number of CPUs
- Main advantages: process domains with very large memory requirements, reduce runtime

geogrid: Distributed memory (cont.)

- Using io_form_output = 10X (e.g., 101, 102), size of domain theoretically limited by aggregate memory space of all CPUs!
  - Each CPU writes only its part of domain to a file; no need to collect domain on a single CPU for output
- Performance scaling*

Grid Descr.	Size	2 CPUs (mm:ss)	4 CPUs (mm:ss)	8 CPUs (mm:ss)	16 CPUs (mm:ss)
South-eastern US	2220 x1830	16:27	08:12	04:19	03:19

** Tests run on NCAR's IBM "bluesky" using* io_form_output=101 *(binary)*

Outline of Topics

geogrid
- purpose
- installing and running (live demo!)
- timing & memory performance
- output field comparison: grid_gen and TERRAIN
- importing new data sources

ungrib

metgrid

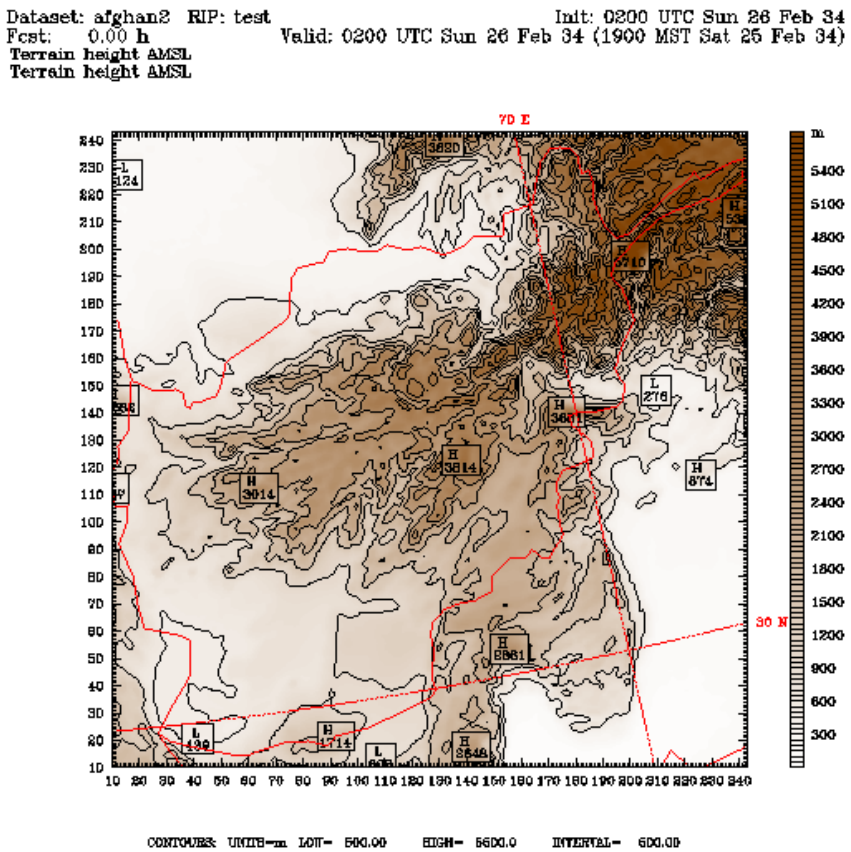
real

geogrid: output comparisons

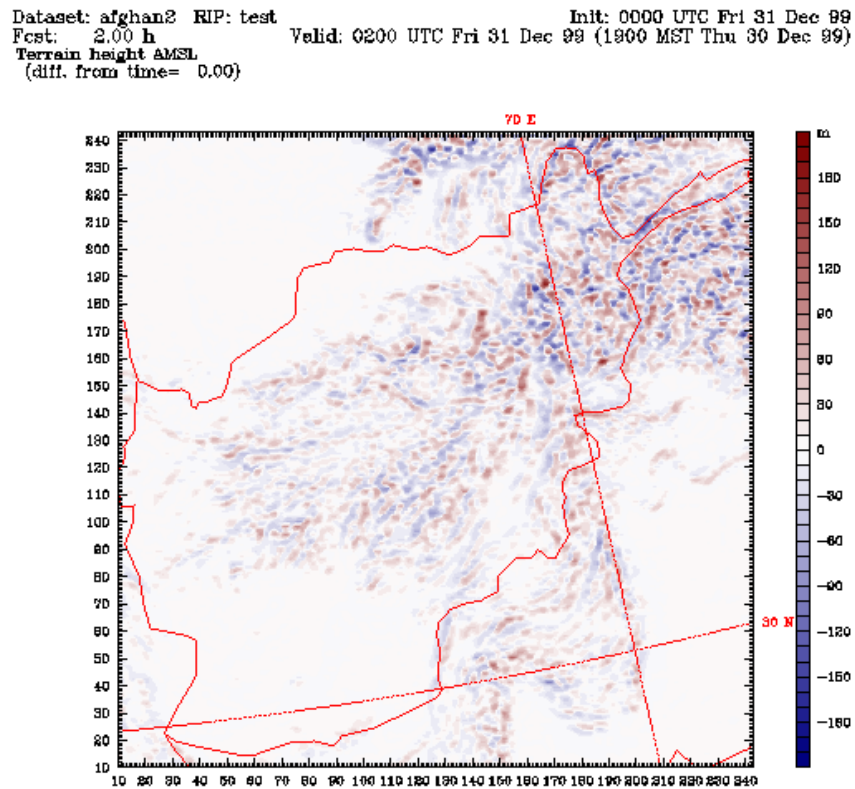
- Topography height and land use index are *very* similar to MM5 TERRAIN output
  - We know how TERRAIN computes these fields, and can employ the same methods in geogrid through the GEOGRID.TBL file
- Topography and land use index differ slightly from SI
  - Perhaps due to intermediate interpolations and smoothing?

Example: T44 Afghanistan domain

- WPS topography field matches MM5 TERRAIN almost exactly
- WPS differs sometimes significantly from SI



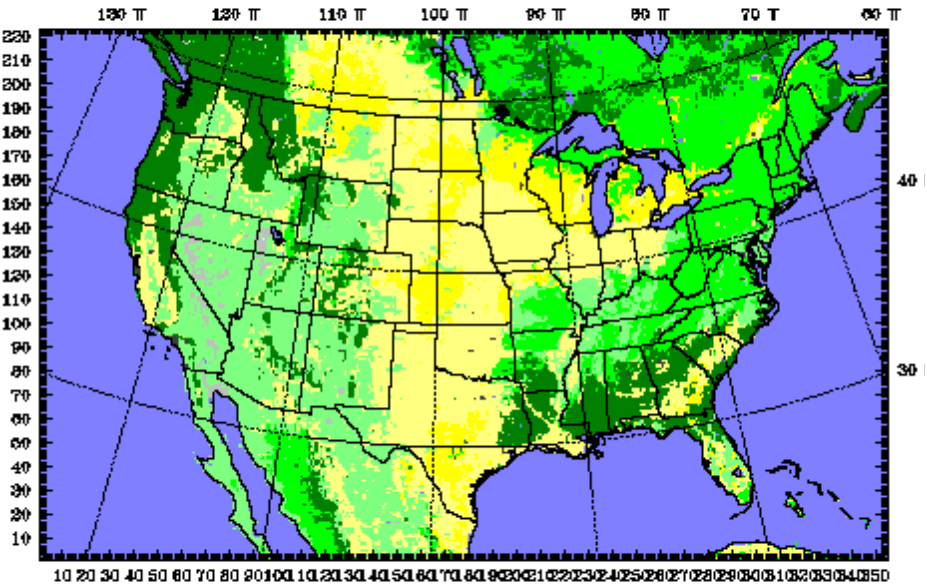
HGT field from WPS



Difference from SI

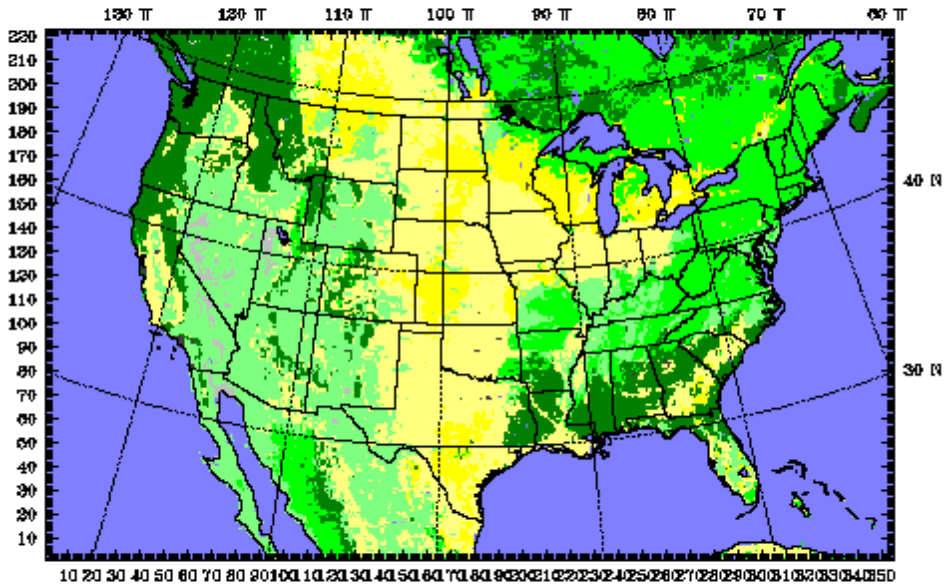
Example: CONUS, 360x230 22 km Land use categories from 2’ input

aset: conus1 RIP: test
: 0.00 h Init: 1400 UTC Thu 27 Oct
Valid: 1400 UTC Thu 27 Oct 33 (0800 MDT Thu 27 Oct
d use category



MM5 Land Use

Dataset: conus1 RIP: test
Fcst: 0.00 h Init: 1400 UTC Thu 27 Oct
Valid: 1400 UTC Thu 27 Oct 33 (0800 MDT Thu 27 Oct



WPS Land Use

Outline of Topics

geogrid
- purpose
- installing and running (live demo!)
- timing & memory performance
- output field comparison: SI and TERRAIN
- importing new data sources

ungrib

metgrid

real

geogrid: using new data sources

- To add a new data source, need to

  1) Place data in proper binary format

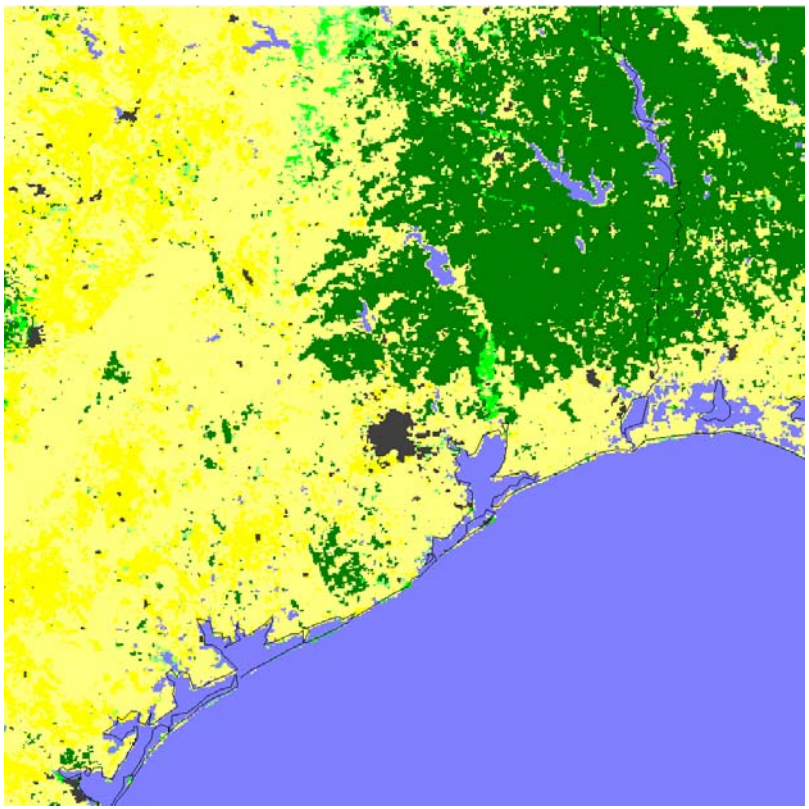
  2) Create an “index” file to define projection and dimensions of data

  3) Specify data in GEOGRID.TBL file

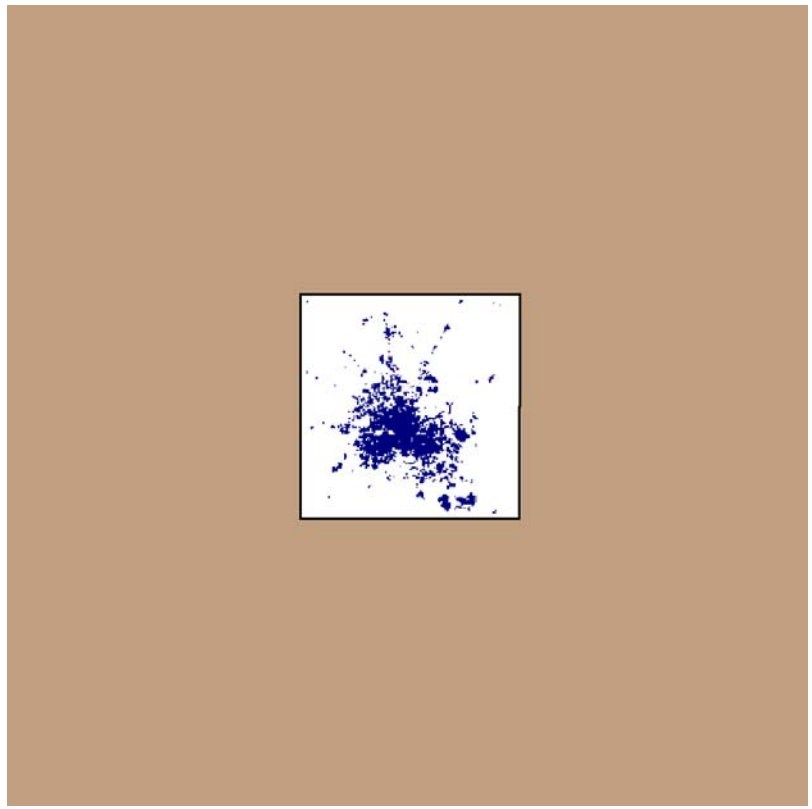
- Definition of binary format:
  - Data must fill a rectangular array; missing data can be given a null/missing value
  - Data can be projected: regular lat/lon, Lambert conformal, Mercator, polar steroegraphic
  - Array written to file in row-major order, beginning at bottom row
  - In file, use 1, 2, 3, or 4 bytes per element
  - All elements scaled to integers by constant scale factor

Example: Houston Urban LU Dataset

- Given dataset for new Houston urban land use categories
  - Regular lat/lon projection, 30” resolution; categories 31, 32 & 33



Urban areas (black) using USGS 24-category data set



Area of Houston data tile in relation to model domain; white=missing data and blue=valid data

Example: Houston Urban LU Dataset

To make use of the new data, we do the following:

1) Write the data to the binary format used by geogrid
2) Create an index file for the data

type=categorical
category_min=31; category_max=33
projection=regular_ll
dx=0.00833333; dy=0.00833333
known_x=1.0; known_y=1.0
known_lat=29.3375
known_lon=-95.9958333
wordsize=1
tile_x=157; tile_y=143; tile_z=1
missing_value = 0.
units="category"
description="3-category urban LU"

Example: Houston Urban LU Dataset

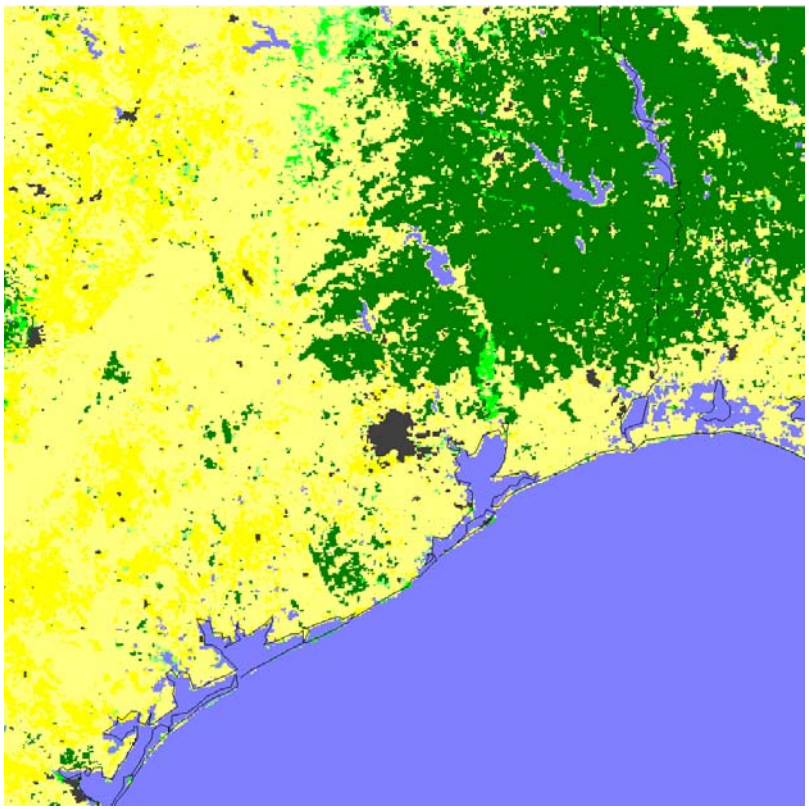
3) Define an entry for the data in GEOGRID.TBL

```
==============================
name=LANDUSEF
        priority=2
        dest_type=categorical
        z_dim_name=land_cat
        interp_option = default:nearest_neighbor
        abs_path = default:/users/duda/Houston/
==============================
```

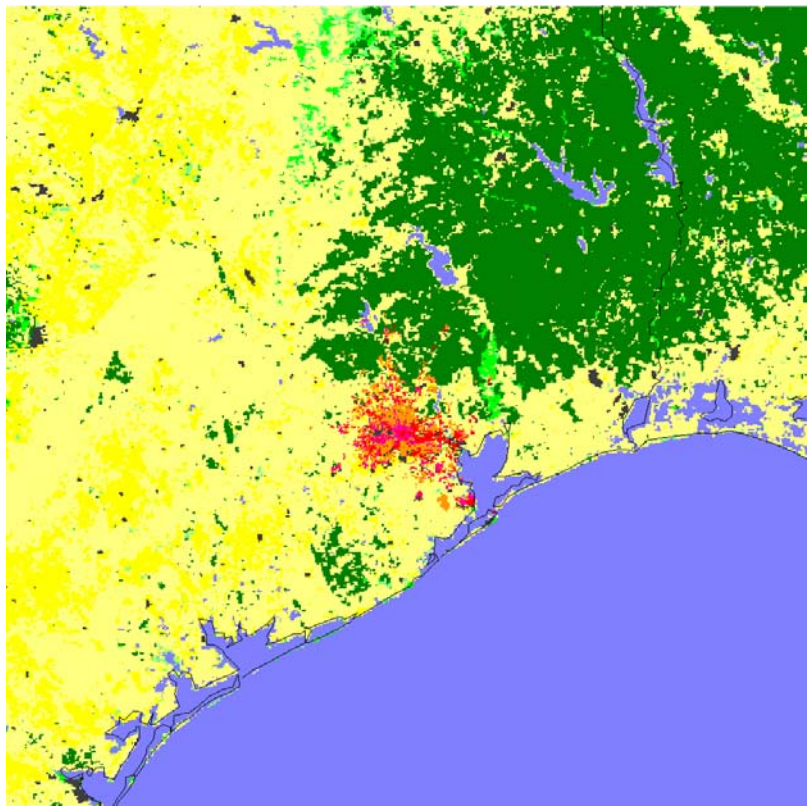
Example: Houston Urban LU Dataset

4) Run geogrid.exe

Any gridpoints covered by Houston data will use it; otherwise default USGS data will be used



Urban areas (black) using USGS 24-category data set



Augmented urban areas (red shades) using new LU data set

Outline of Topics

geogrid

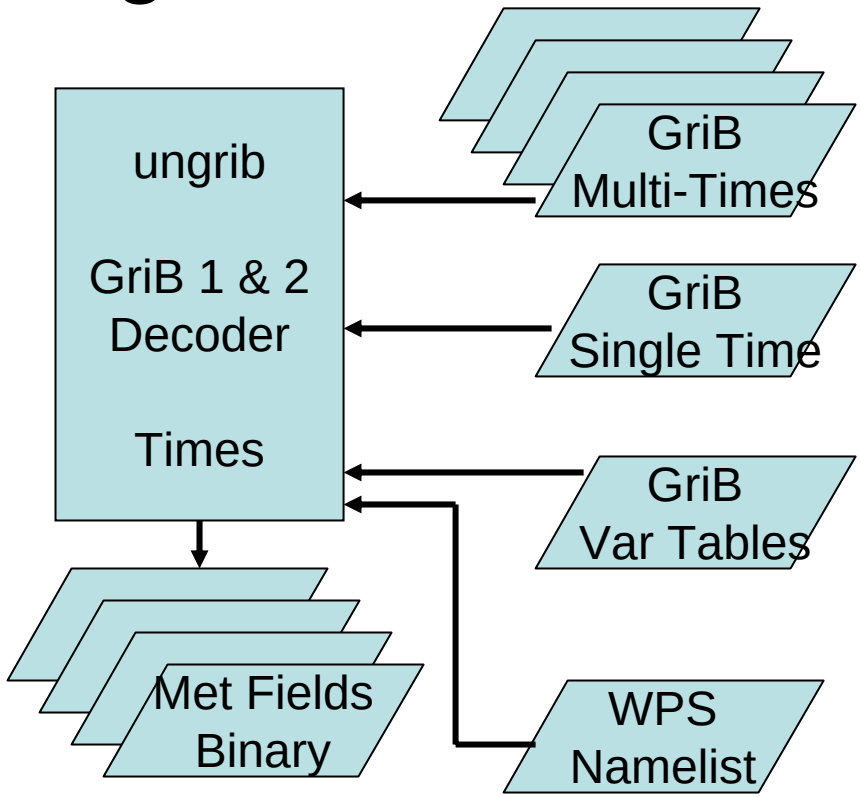
ungrib
- purpose
- installing and running (live demo!)
- additional features

metgrid

real

The ungrib program

ungrib is the WPS counterpart to grib_prep (SI) and pregrid (MM5)



Purposes of ungrib

- extract meteorological fields from GRIB files
- write fields to intermediate format accepted by metgrid

Outline of Topics

geogrid

ungrib
- purpose
- installing and running (live demo!)
- additional features

metgrid

real

Steps to install and run ungrib

1. Install jasper libraries (needed for GRIB2 compression)
2. Configure and compile ungrib
3. Create (or link to) appropriate Vtable
4. Edit namelist to define time range to process (optionally, output format)
5. Link GRIB files to GRIBFILE.AAA … GRIBFILE.ZZZ
6. Run ungrib.exe

Vtables are similar to those in grib_prep (SI) and pregrid (MM5)

- GRIB2 Vtables include extra parameters
- only need to create once per input source

Outline of Topics

geogrid

ungrib
- purpose
- installing and running (live demo!)
- additional features

metgrid

real

Additional ungrib features

- Backward compatibility
  - Can write to intermediate format used by REGRID or hinterp
- Supports GRIB1 and GRIB2
  - Combine with backward compatibility to use GRIB2 data with SI or MM5
- Supports input files with 1-minute resolution; up to 17576 input files

Outline of Topics

geogrid

ungrib

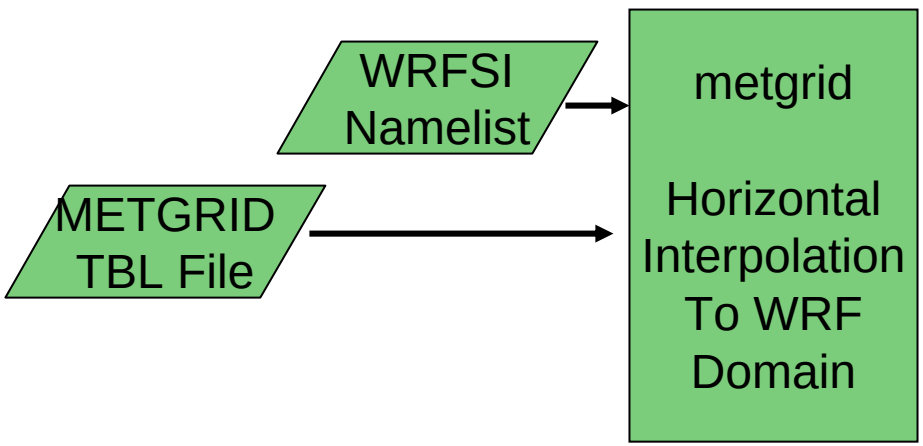
metgrid
- purpose
- installing and running (live demo!)
- timing and memory performance
- tuning output fields with METGRID.TBL

real

The metgrid program

Purposes of metgrid

- Horizontally interpolate meteorological data to model grids
- Rotate winds to model grid U and V
- Provide sorted 3-d fields to real.exe



metgrid is the WPS counterpart to hinterp (SI) and regrid (MM5)

Outline of Topics

geogrid

ungrib

metgrid
- purpose
- installing and running (live demo!)
- timing and memory performance
- tuning output fields with METGRID.TBL

real

Steps to install and run metgrid

1. Configure and compile metgrid
2. Locate ungribbed data and domain files from geogrid
3. Edit namelist to define time ranges and # nests
4. (OPTIONAL!) Edit METGRID.TBL to modify handling of data – only needed for special circumstances
5. Run metgrid.exe to produce met_em files

Outline of Topics

geogrid

ungrib

metgrid
- purpose
- installing and running (live demo!)
- timing and memory performance
- tuning output fields with METGRID.TBL

real

Running metgrid: timing performance*

Grid Descr.	# Doms	Sizes	DX	hinterp (sec/time)[1]	metgrid (sec/time)[1]
T92 (CONUS)	1	355x223	15 km	1.81	11.35
Antarctica	1	331x313	20 km	2.30	11.41

[1] Average number of seconds per output time using HINTERP_METHOD=1 (4pt)

** tests run on a PC running FC4; 2 GB mem; 3.4 GHz Xeon; PGI 6.1 compilers*

Future metgrid work: implement namelist option to eliminate computations over interior of domain for later output times

- *This should provide very substantial improvements to processing time, especially for long simulations*

metgrid: Distributed memory capable

- Like geogrid, metgrid may be compiled with MPI libraries for distributed memory machines
- Two possibilities for io_form_input and io_form_output
  - Ex: io_form_output=102 – one output file per processor, NetCDF
  - Output fields identical regardless of number of CPUs

metgrid: Distributed memory (cont.)

- Primary reason for running metgrid in parallel is larger memory
  - Using io_form=10x, domain size limited by aggregate memory of all CPUs

Grid size	1 CPU (sec)[1]	2 CPUs (sec)[1]	4 CPUs (sec)[1]	8 CPUs (sec)[1]
1480x1220	317	176 [1.8]	116 [2.7]	70 [4.5]

[1] Average number of seconds per output time [speedup over 1 CPU]

  - For improved runtime, manually parallelize over time periods (e.g., process one time period on each CPU)

metgrid: memory performance

- Optionally, metgrid can shift fields from memory to disk when they are not in use
  - Memory only needs to be large enough to hold one 3-d array
  - Disk is slower than memory -- performance decreases, but can handle very large domains

Outline of Topics

geogrid

ungrib

metgrid
- purpose
- installing and running (live demo!)
- timing and memory performance
- tuning output fields with METGRID.TBL

real

Ingesting new fields in megrid

- Every field in ungribbed files will be interpolated
  - If no entry in METGRID.TBL for a field, a default interpolation scheme (nearest neighbor) will be used
- Suitable entries in METGRID.TBL are provided for common fields
  - *Thus, many users will rarely need to edit METGRID.TBL*
- When necessary, different interpolation methods (an other options) can be set in METGRID.TBL

Example using METGRID.TBL

- Suppose we have a 1000x1000 domain over Houston (dx=500m)
- Meteorological data come from GFS
  - *Note that we will be interpolating 1-degree data onto a 500-m grid!*
- Also suppose that there is no METGRID.TBL entry for soil moisture SM000010

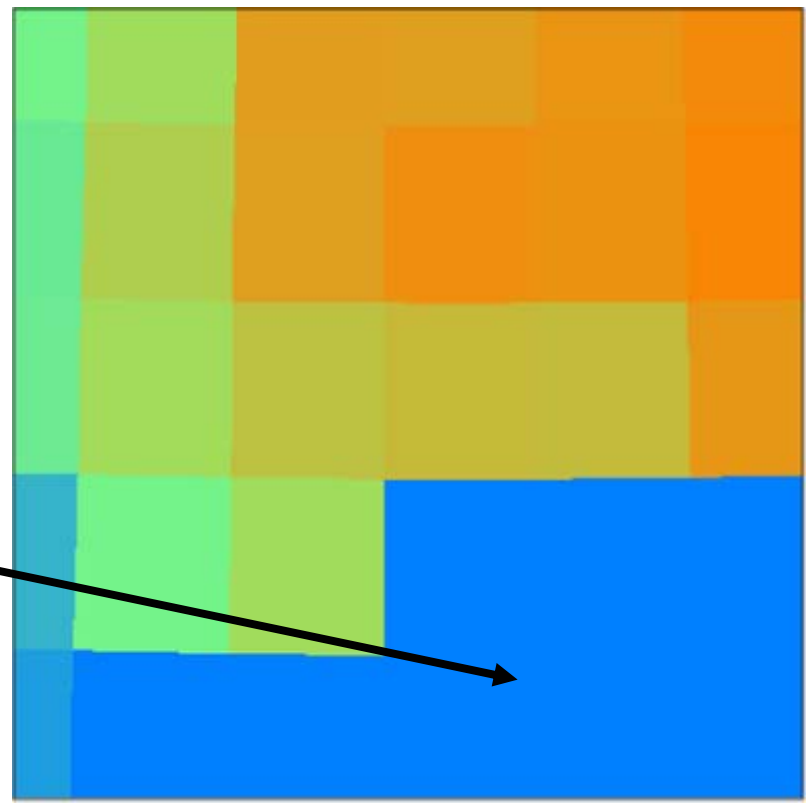
Example using METGRID.TBL (cont)

- Initially, run metgrid.exe and get a message:

INFORM: Entry in METGRID.TBL not found for field SM000010. Default options will be used for this field!

- Resulting field looks like

GFS puts -1E30 in water areas (LANDSEA=0)



Example using METGRID.TBL (cont)

- We add an intial entry in METGRID.TBL for SM000010:

==========

name = SM000010

masked = water

interp_mask = LANDSEA(0)

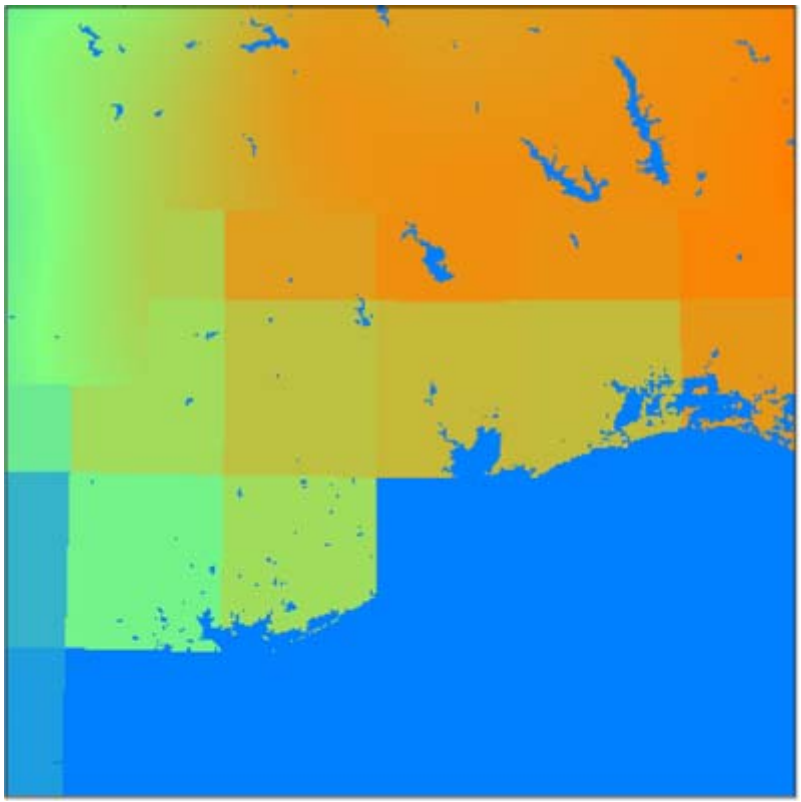
interp_option = sixteen_pt + nearest_neighbor

fill_missing = 0.

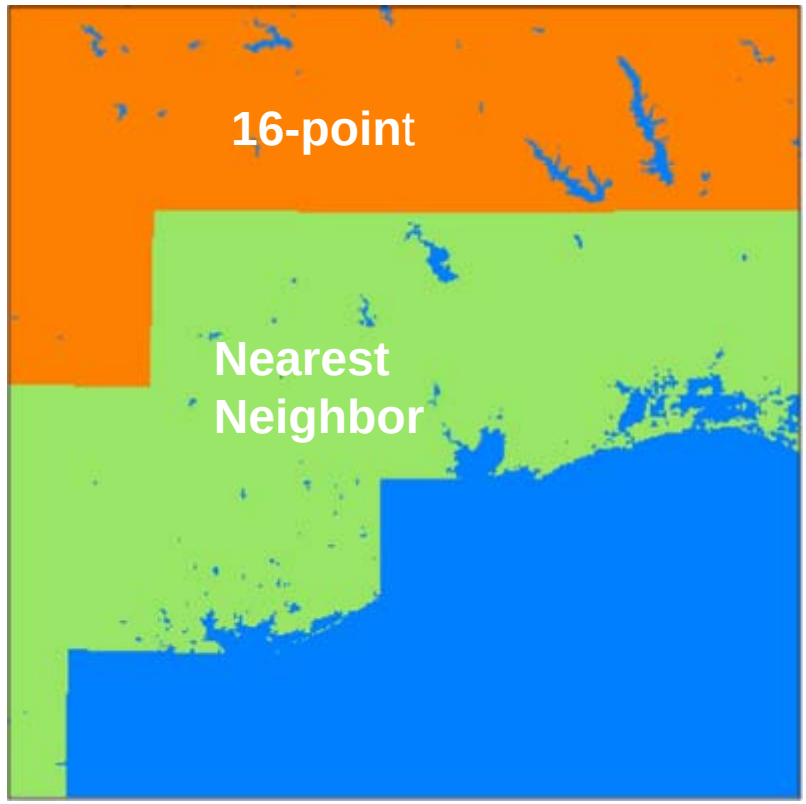
==========

Example using METGRID.TBL (cont)

- Running metgrid.exe again, the SM000010 field now looks like



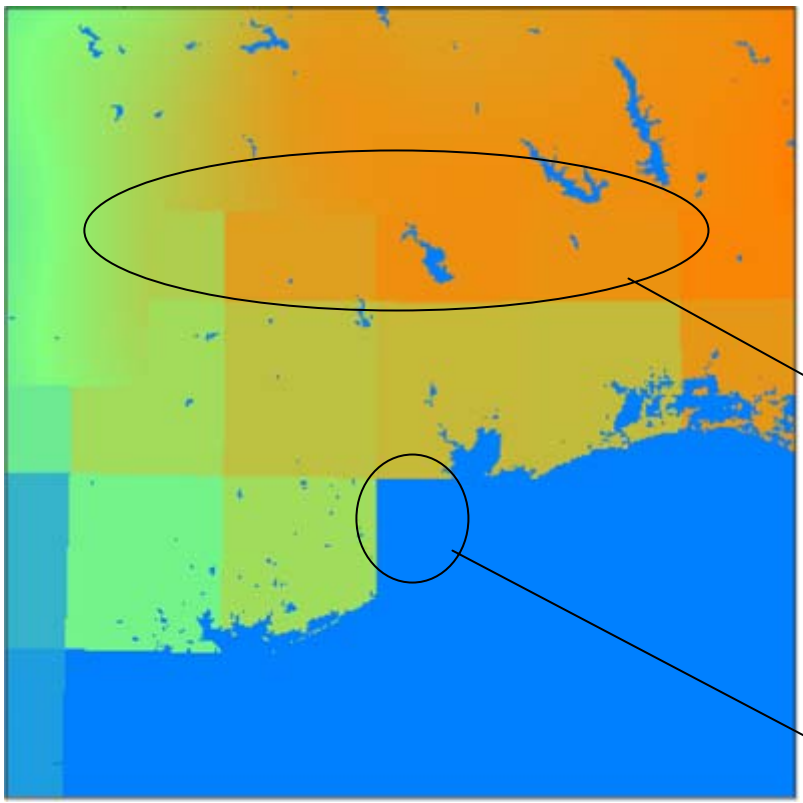
Interpolated SM000010 field (sixteen_pt + nearest_neighbor)



Which interpolator was used at each model grid point

Example using METGRID.TBL (cont)

- The interpolated field looks “blocky” near the coastline



Interpolated SM000010 field (sixteen_pt + nearest_neighbor)

Should be sufficient data to use 4-point interpolation in these areas

Model grid points here should be adjacent to at least one valid GFS point (though not nearest)

Example using METGRID.TBL (cont)

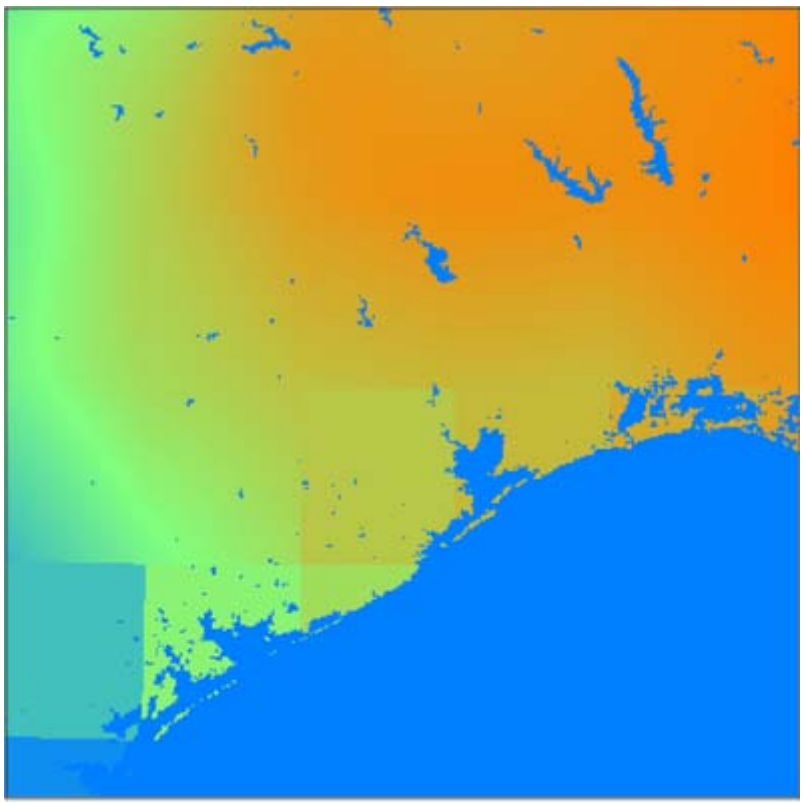
- Update the METGRID.TBL entry for SM000010

```
==============================
name = SM000010
masked = water
interp_mask = LANDSEA(0)
interp_option = sixteen_pt + four_pt + average_4pt
fill_missing = 0.
==============================
```

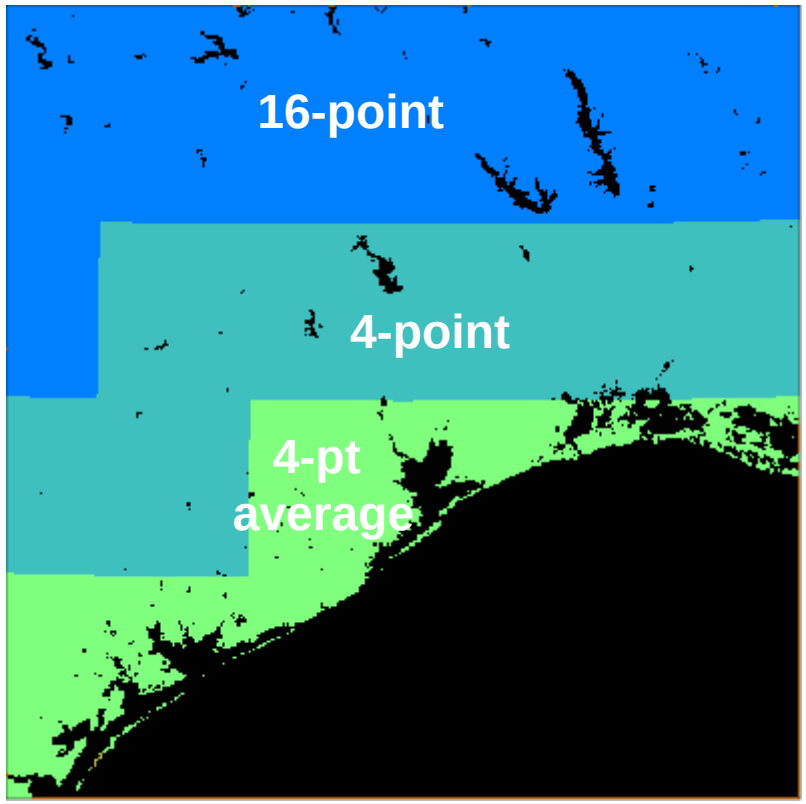
- If 16-pt doesn’t work, then try 4-pt before reverting to a 4-point average
  - Note that 4-point average will work anywhere nearest_neighbor would (missing/masked values not counted in the average)

Example using METGRID.TBL (cont)

- The final field, below-left:



Interpolated SM000010 field (sixteen_pt + four_pt + average_4pt)



Which interpolator was used at each model grid point

Outline of Topics

geogrid

ungrib

metgrid

real

- running real.exe (live demo!)

real.exe

- Vertical interpolation handled in real.exe
- Only ARW currently, working with NCEP to get real_nmm.exe updated
- Input from metgrid can be either isobaric or generalized vertical coordinate (RUC only so far)
- New fields in the metgrid file must be included in WRF Registry

real.exe

- Serial and DM parallel, nest-ready, linear and higher order vertical interp
- Extra unused 3d arrays are carried around in WRF, this needs to be fixed
- Backward comaptible – can use output from either SI vinterp and WPS metgrid

real.exe

- Small changes to namelist: input name and number of vertical levels on input

```
&time_control
 auxinput1_inname = "met_em.d<domain>.<date>"
 /

 &domains
 num_vert_levels  = 40
 /
```

Questions?