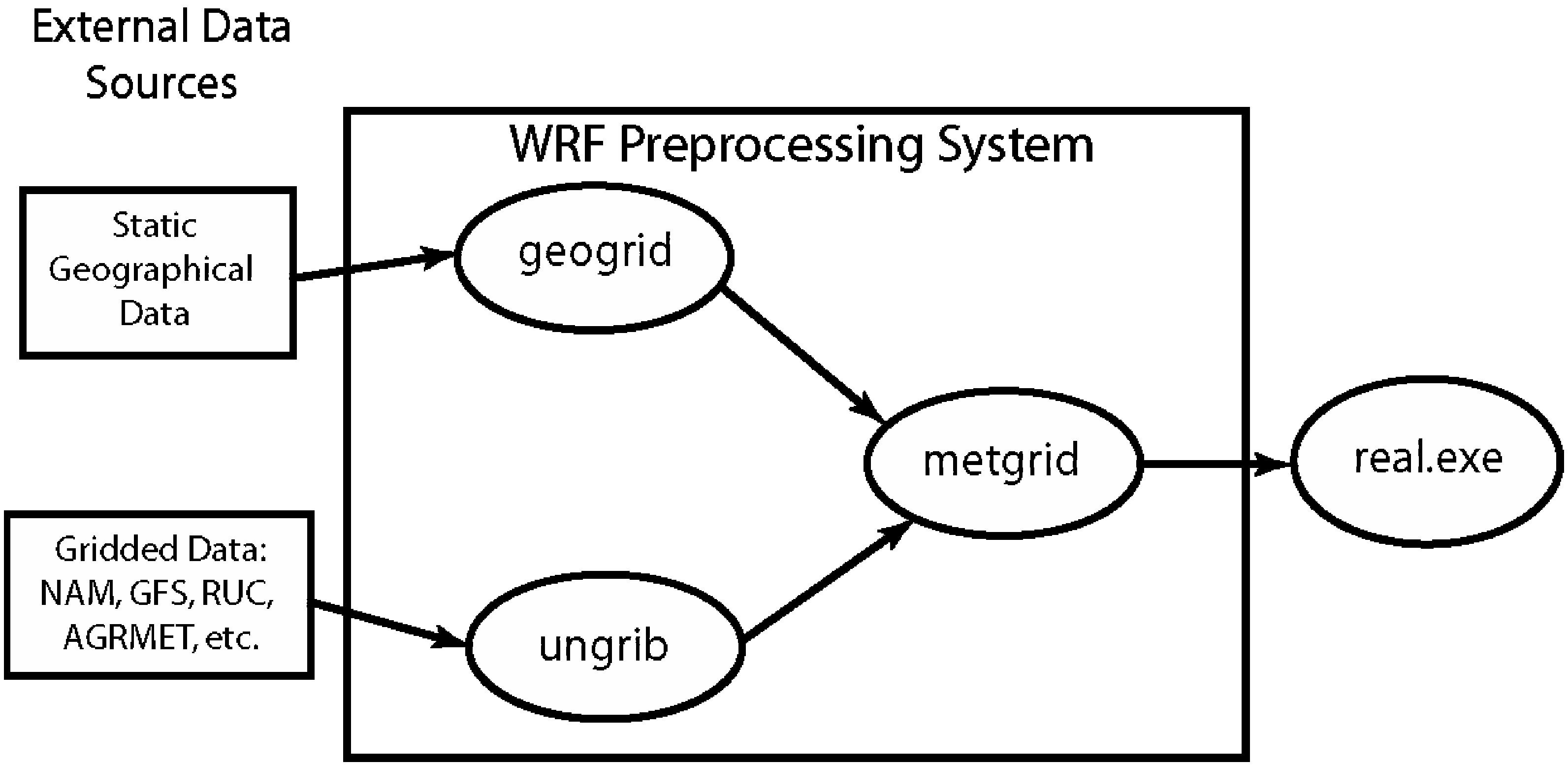
Running the WRF Preprocessing System

Michael Duda

Review

- Briefly recall the data flow among programs:



Review

- geogrid (think geographical)
  - Define size/location of model domains and interpolate static terrestrial fields to simulation grids
- ungrib
  - Extract meteorological fields from GRIB files
- metgrid (think meteorological)
  - Horizontally interpolate meteorological fields (from ungrib) to simulation grids (defined by geogrid)

Overview

- How to run through the WPS for basic cases
  - Standard test case with single met. data source
  - Typical case with multiple met. data sources
- Advanced features of the WPS
  - The GEOGRID.TBL file
  - Ingesting new static fields
  - The METGRID.TBL file
  - "Managing" meteorological fields
- WPS utility programs

Running geogrid.exe

Basic steps to run *geogrid*

1) Edit `namelist.wps`
   - define projection and domain locations
   - specify path to static terrestrial data

A note about editing `namelist.wps`:

When running the WPS program *<program_name>*, it is only necessary to set variables in the sections &*share* and &*<program_name>*

Running geogrid.exe

2) Run *geogrid.exe*

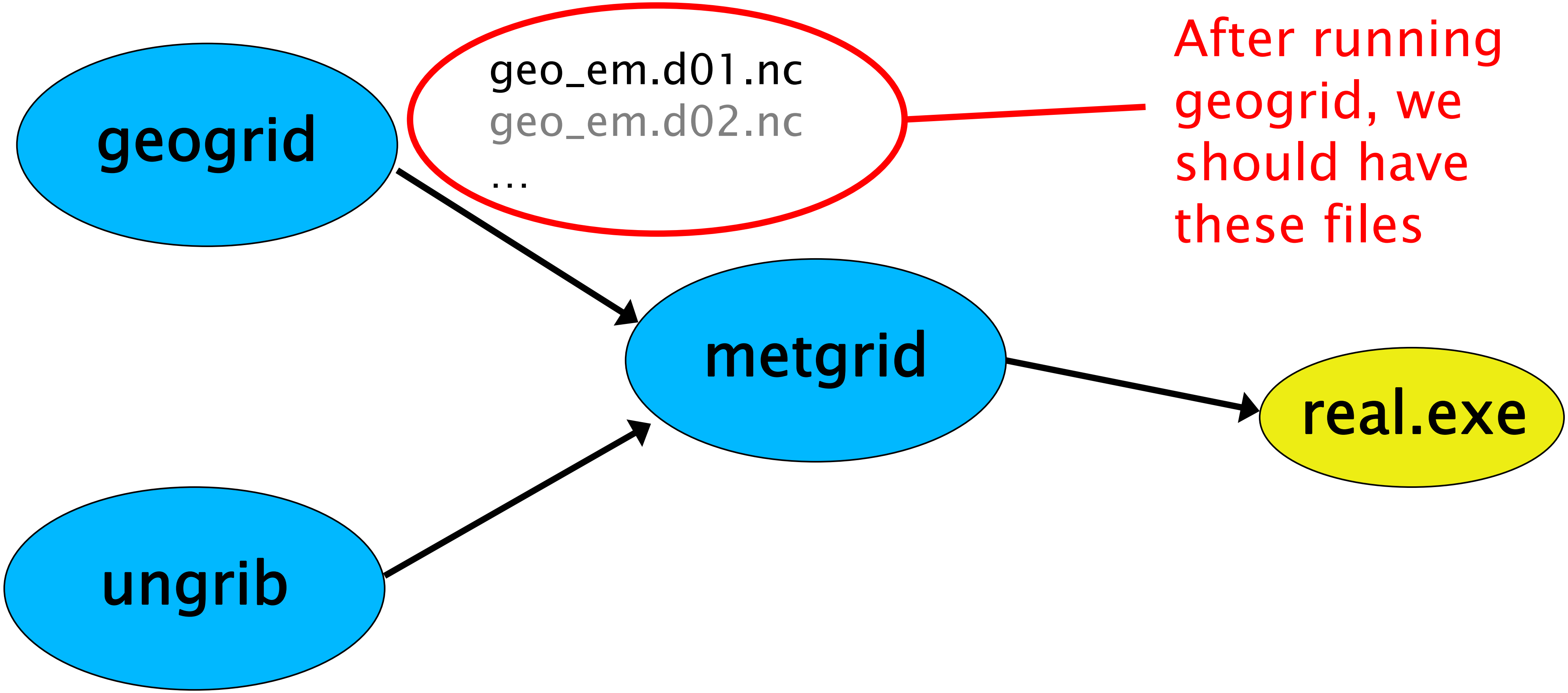
3) Check geogrid output

- Did geogrid run successfully?

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!  Successful completion of geogrid.        !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

- Do `geo_em.d0N.nc` files exist?
- Are the domains in their expected locations?

Running geogrid.exe



After running geogrid, we should have these files

Running ungrib.exe

Basic steps to run *ungrib*

1) Edit `namelist.wps`
   - specify starting and ending times for domains
   - specify interval of available data
2) Link the proper Vtable to the file `Vtable`
3) Link first-guess GRIB files to `GRIBFILE.AAA`, `GRIBFILE.AAB`, …

Running ungrib.exe

Basic steps to run *ungrib* (cont.)

4) Run *ungrib.exe*

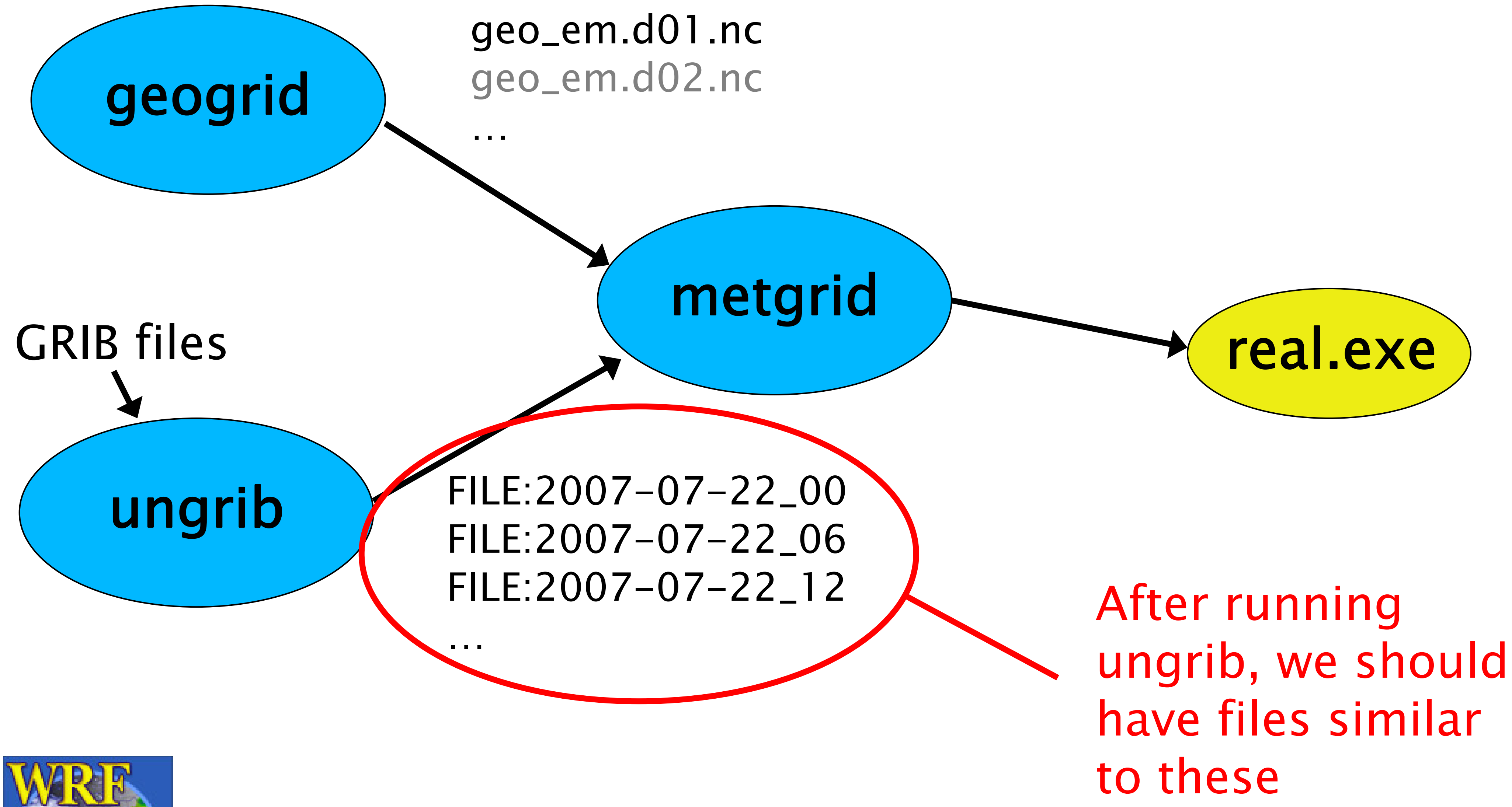
5) Check ungrib output

- Did ungrib run successfully?

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!  Successful completion of ungrib.        !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

- Do `FILE:`*`YYYY-MM-DD_HH`* files exist?
- Are all of the expected fields in the ungrib output files?

Running ungrib.exe



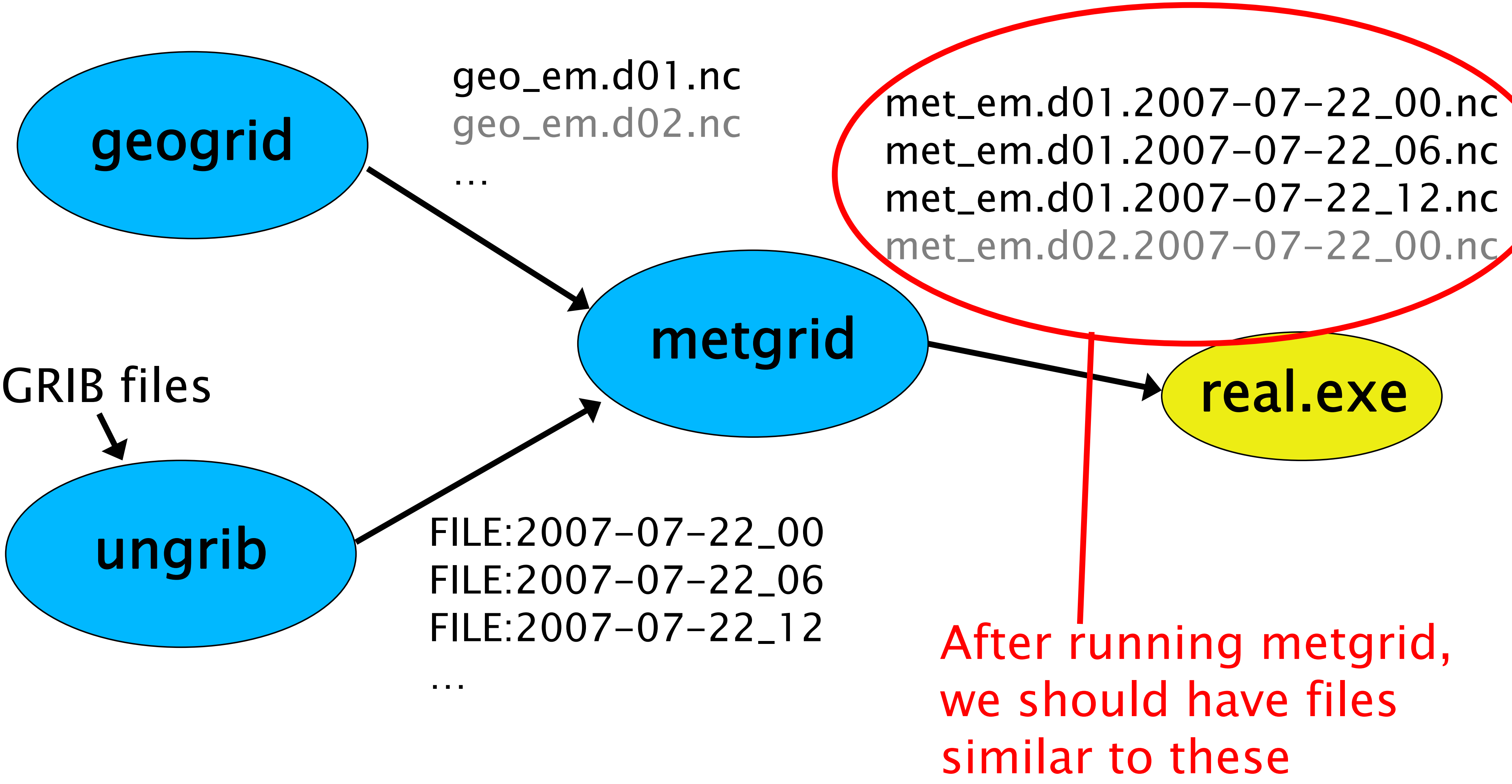
After running ungrib, we should have files similar to these

Running metgrid.exe

Basic steps to run *metgrid*

1) Edit `namelist.wps`
   - specify starting and ending times for all grids
   - specify path and prefix of ungrib output
2) Run *metgrid.exe*
3) Check metgrid output
   - Did metgrid run successfully?
   - Do `met_em.d0N.YYYY-MM-DD_HH.nc` files exist?

Running metgrid.exe



After running metgrid, we should have files similar to these

Running WPS: Summary

- The basic steps to running each WPS program can be summarized as:
  - Set variables in the `&share` and `&<`*`program name`*`>` sections in the `namelist.wps` file
    - E.g., for metgrid, edit `&share` and `&metgrid` sections
  - For ungrib, link `Vtable` and `GRIBFILE.???` files
  - Run the program executable
  - Check that the proper output files exist and contain good data

DEMONSTRATION: Basic test case

- For this demonstration:
  - Assume we’re given a specification for domains
  - We will only use a single source of GRIB data (1-degree GFS)
  - Basically, we’ll just run each component to see what files are created during a successful WPS run

DEMONSTRATION: Typical case

- What new things will we do?
  - We need to come up with domain specifications “from scratch”
  - The meteorological data come in three pieces: 3-d fields, surface fields, and fixed fields
    - AWIP data
  - We also want to use a separate SST field

DEMONSTRATION: Summary

- What steps did we take?

  1) Edit `&geogrid` namelist and iteratively refine the location of our coarse domain and nests

  2) Set dates in `&share` namelist, and run ungrib.exe separately for each piece of data, changing the prefix in the `&ungrib` namelist each time

  3) List all data sources in the `&metgrid` namelist before running metgrid

Overview

- How to run through the WPS for basic cases
  - Standard test case with single met. data source
  - Typical case with multiple met. data sources
- Advanced features/use of the WPS
  - The GEOGRID.TBL file
  - Ingesting new static fields in geogrid
  - The METGRID.TBL file
  - “Working with” meteorological fields in metgrid
- WPS utility programs

The GEOGRID.TBL File

- GEOGRID.TBL is the file that determines which fields are interpolated by geogrid
  - Generally, user will want all of the default fields, so few reasons to edit GEOGRID.TBL
  - When new data sources are involved, or when the default treatment of fields is inadequate, user will want to edit GEOGRID.TBL
  - Each *entry* in GEOGRID.TBL corresponds to one data source

Example: GEOGRID.TBL Entries

```
===============================
name = VEG_CATEGORY
        priority = 1
        dominant_only = VEG_CAT
        dest_type  = categorical
        z_dim_name = veg_cat
        interp_option = default:nearest_neighbor
        abs_path      = default:/data/duda/MODIS/
===============================
name = SOILCTOP
        dominant = SOILCAT
        priority = 1
        dest_type = categorical
        z_dim_name = soil_cat
        interp_option =      2m:sixteen_pt
        interp_option =     10m:sixteen_pt
        rel_path=      2m:soiltype_top_2m/
        rel_path=     10m:soiltype_top_10m/
===============================
```

Entry for the field
“VEG_CATEGORY”

Entry for the field
“SOILCTOP”

New Field in GEOGRID.TBL

There are three basic types of new data to be added through the GEOGRID.TBL file:

1) Completely new fields
   - fields that were previously not processed by geogrid
2) Different resolution data sets for an existing field
   - e.g., Adding a 100-meter resolution topography data set
3) Alternative sources for a field that must be used in addition to an existing source
   - e.g., A new soil category data set exists, but covers only southern Colorado

GEOGRID.TBL: Data Type 1

Completely new fields:

For a new field, simply add an entry in GEOGRID.TBL for that field.

```
===============================
name = MY_NEW_FIELD_NAME
   priority = 1
   dest_type  = continuous # continuous or categorical?
   interp_option = four_pt
   abs_path      = /data/duda/mydata/
===============================
```

GEOGRID.TBL: Data Type 2

Different resolution data sets for an existing field:

Specify the path to the new data set and which interpolation methods should be used for the new resolution in the existing entry for that field.

```
===============================
name = HGT_M
     priority = 1
     dest_type = continuous
     smooth_option = smth-desmth
     interp_option =         30s:special(4.0)+four_pt
     interp_option =      my_res:four_pt
     interp_option =     default:four_pt
     rel_path=      30s:topo_30s/
     rel_path=   my_res:new_topo_directory/
     rel_path=  default:topo_2m/
===============================
```

GEOGRID.TBL: Data Type 3

Alternative sources for a field that must be used in addition to an existing source :

Add a new entry for the field that has the same name as the field's existing entry, but make priority of new entry higher.

```
===============================
name = HGT_M
      priority = 2
      dest_type = continuous
      interp_option = default:four_pt
      rel_path      = default:some_path/
===============================
name = HGT_M
     priority = 1
     dest_type = continuous
     interp_option = default:four_pt
     rel_path      = default:topo_2m/
===============================
```

Ingesting new static fields

- To add a new data source, need to

1) Write the data in the proper binary format

   – See Chapter 3: “Writing Static Data to the Geogrid Binary Format”

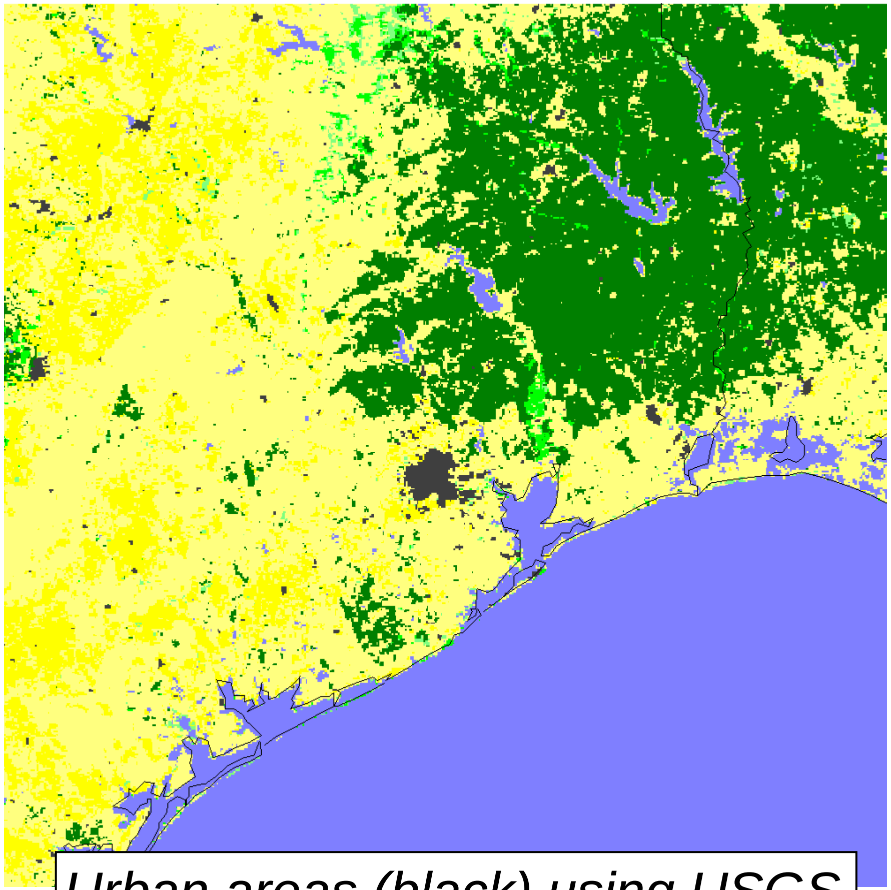
2) Create an “index” metadata file to define projection and dimensions of data

3) Add entry for the data in the GEOGRID.TBL file

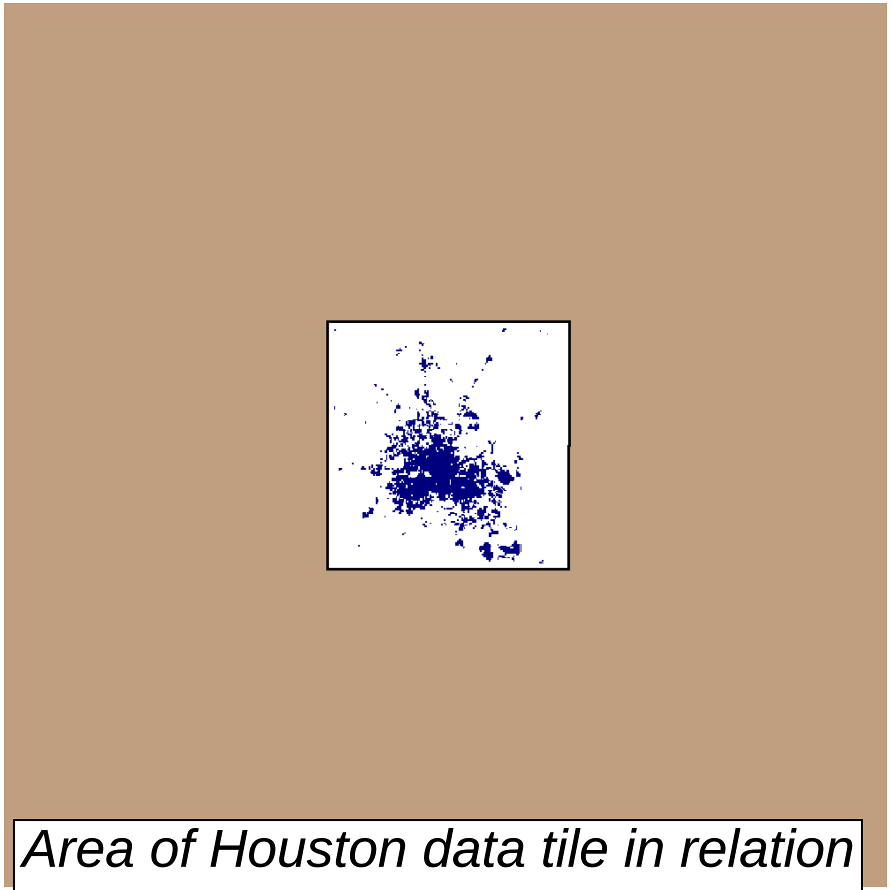
4) Run geogrid.exe

Example: Houston LU Data Set

- 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 LU Data Set

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 LU Data Set

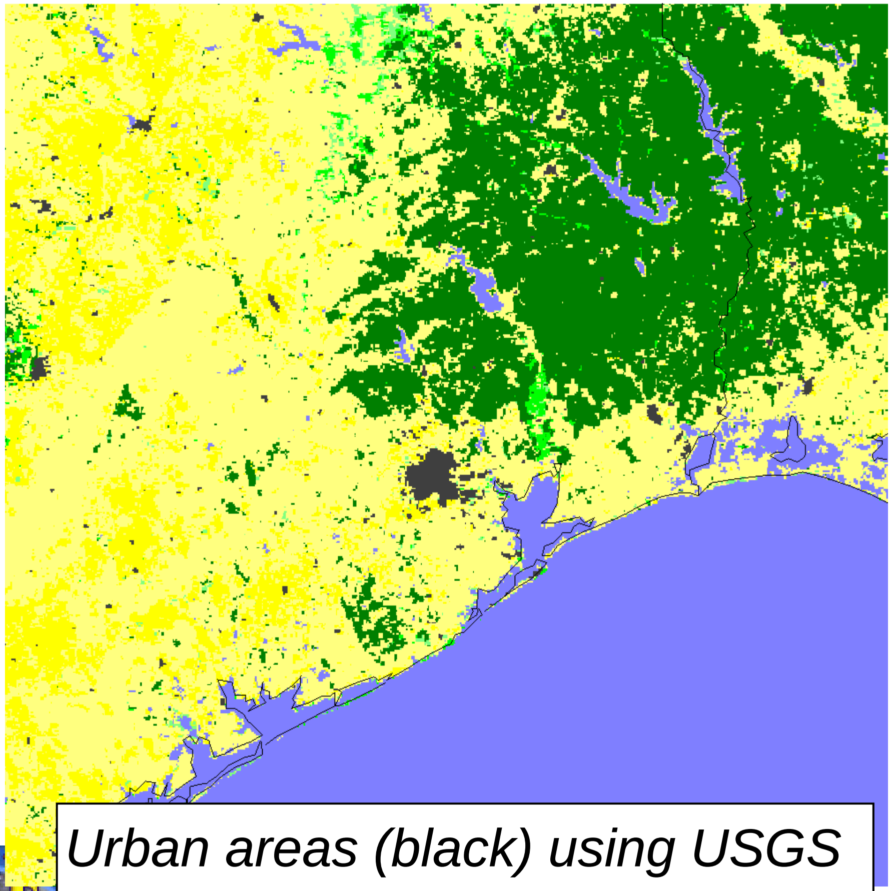
3) Define an entry for the data in GEOGRID.TBL

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

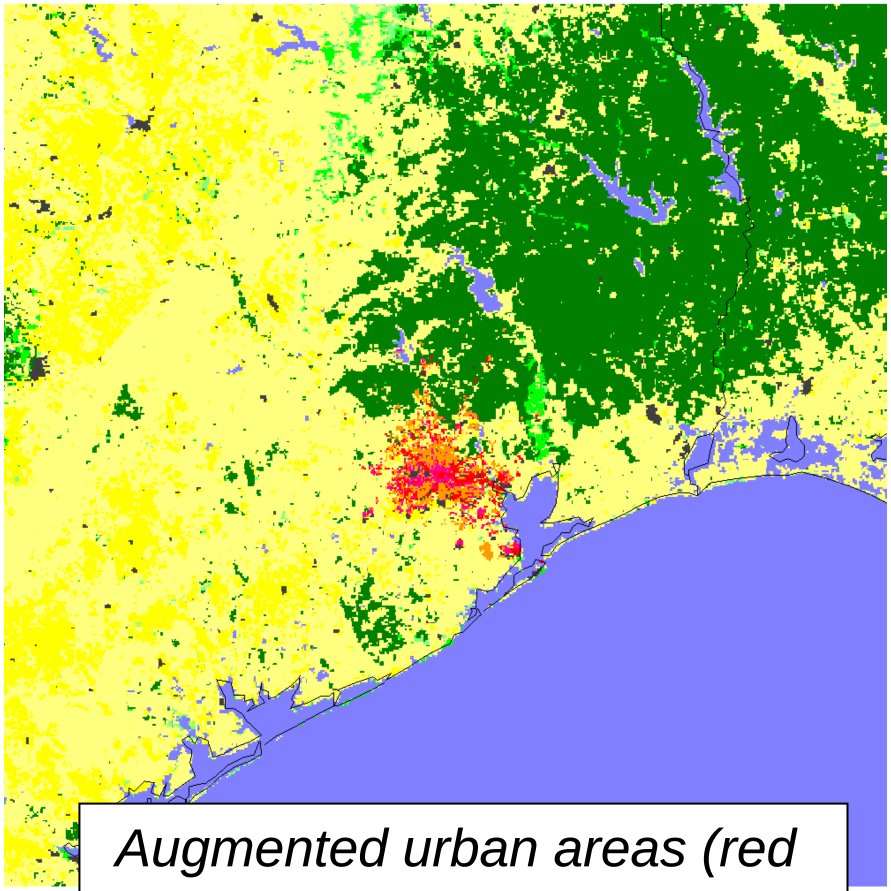
Example: Houston LU Data Set

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

The METGRID.TBL File

The METGRID.TBL file controls how time-varying fields are interpolated

- Unlike GEOGRID.TBL, METGRID.TBL *does not determine which fields will be processed*, only *how to process them*
- Every field in intermediate files will be interpolated
  - If no entry in METGRID.TBL for a field, a default interpolation scheme (<u>nearest neighbor</u>) will be used

The METGRID.TBL File

- 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 (and other options) can be set in METGRID.TBL
  - Interpolation options can depend on the source of a field

Ingesting New Fields in Metgrid

- Suppose we have a 1000x1000 domain over Houston (dx=500 m)
  - This is the same domain as in the urban land use example

- Meteorological data come from 1-degree GFS
  - *Note that we will be interpolating 1-degree data onto a 500-m grid!*

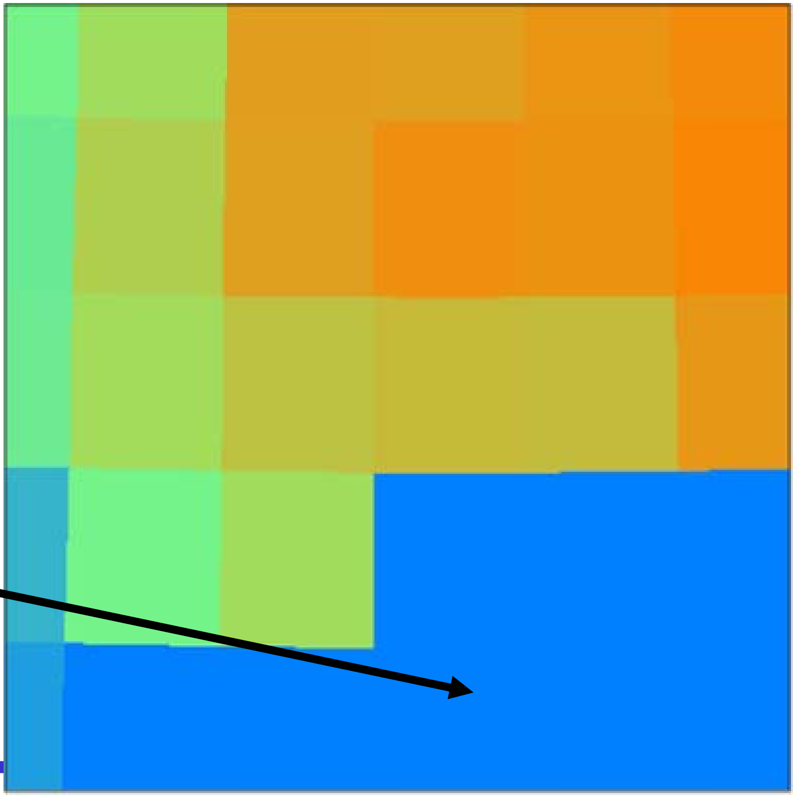
- Also suppose that there is no METGRID.TBL entry for some new soil moisture field, SM000010

Ingesting New Fields in Metgrid

- Initially, run metgrid.exe and get the 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)

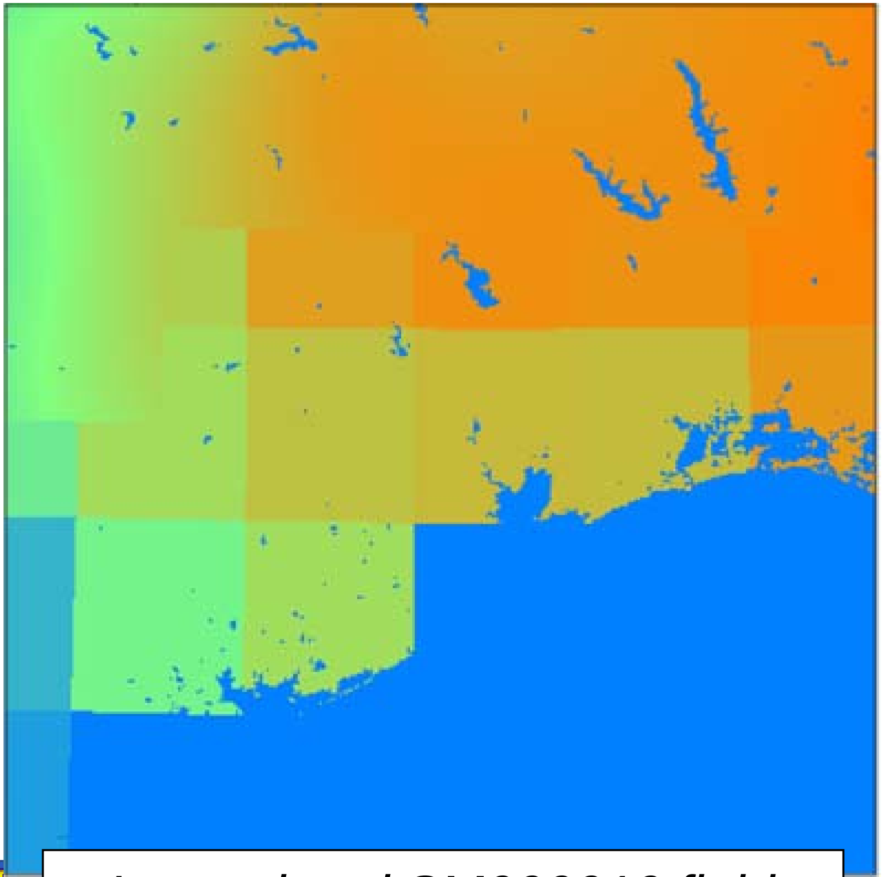
Ingesting New Fields in Metgrid

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

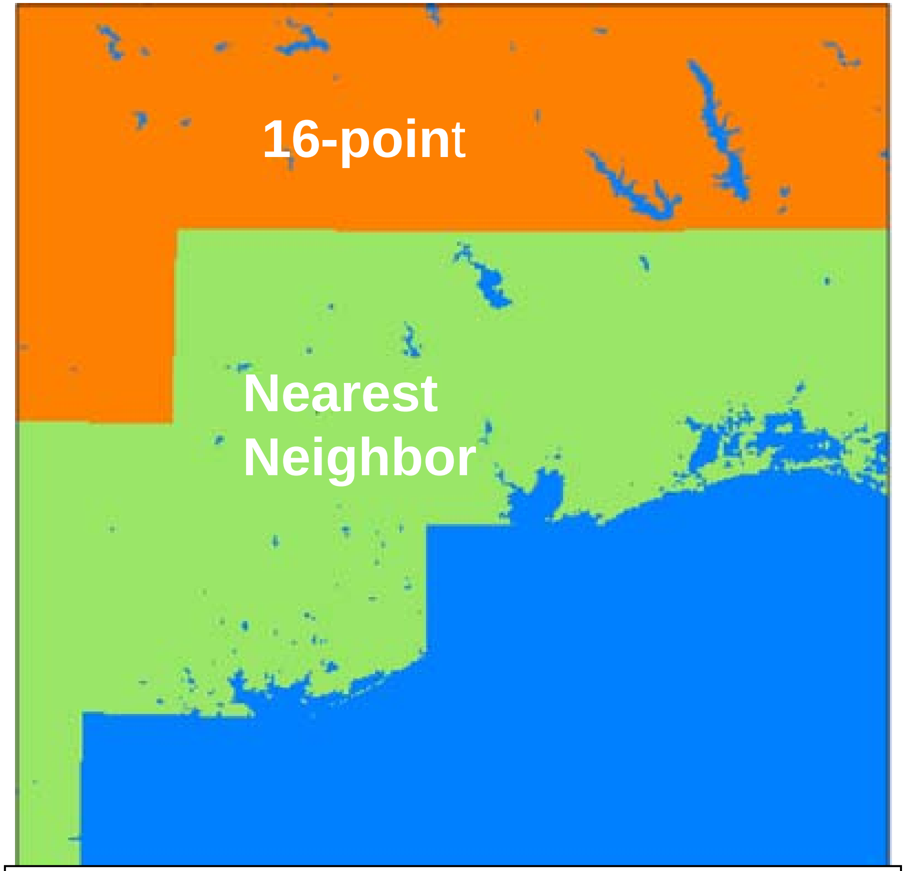
```
===========================
name = SM000010
masked = water
interp_mask = LANDSEA(0)
interp_option = sixteen_pt + nearest_neighbor
fill_missing = 0.
===========================
```

Ingesting New Fields in Metgrid

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



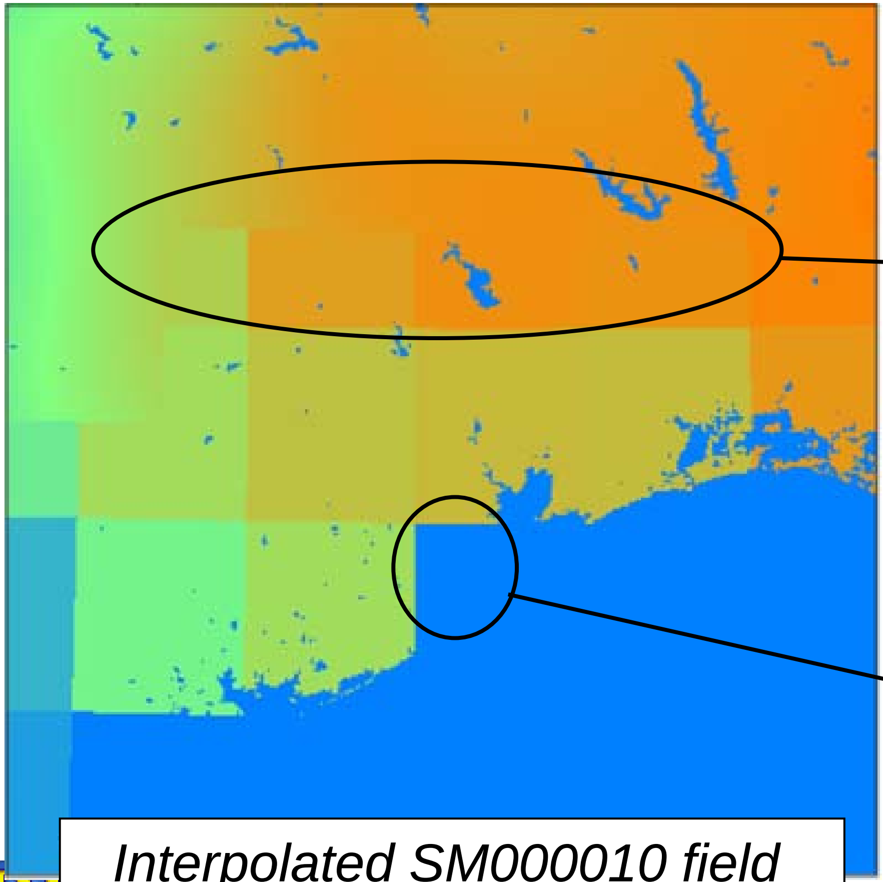
Interpolated SM000010 field (sixteen_pt + nearest_neighbor)



Which interpolator was used at each model grid point

Ingesting New Fields in Metgrid

- 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)

Ingesting New Fields in Metgrid

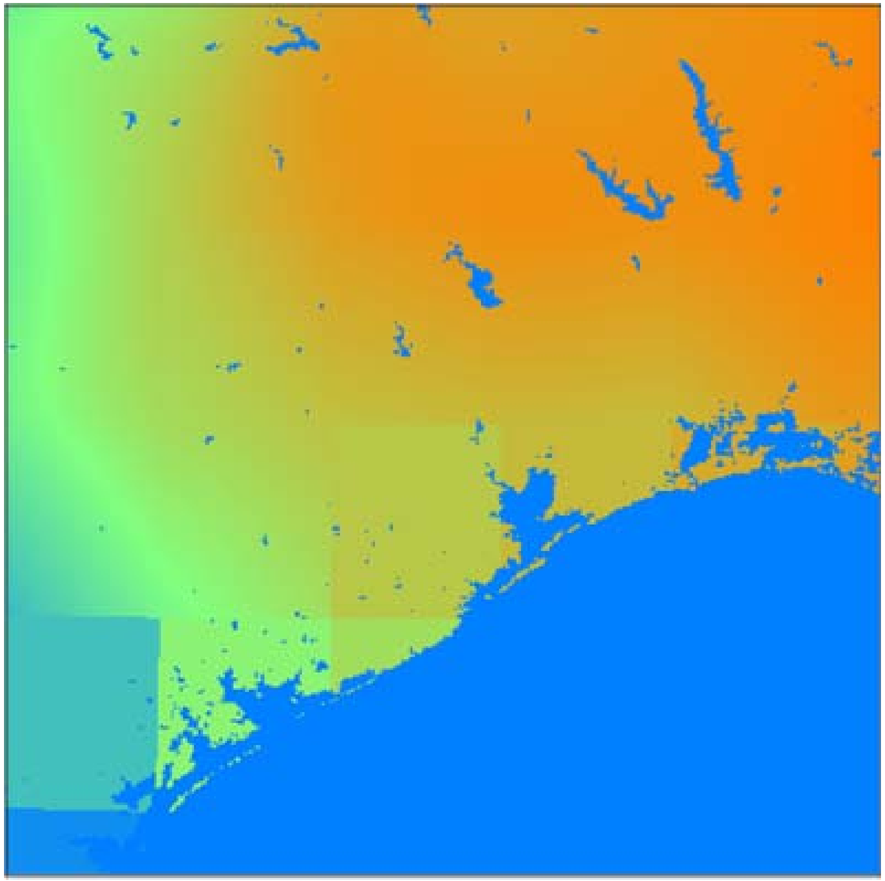
- Update the METGRID.TBL entry for SM000010

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

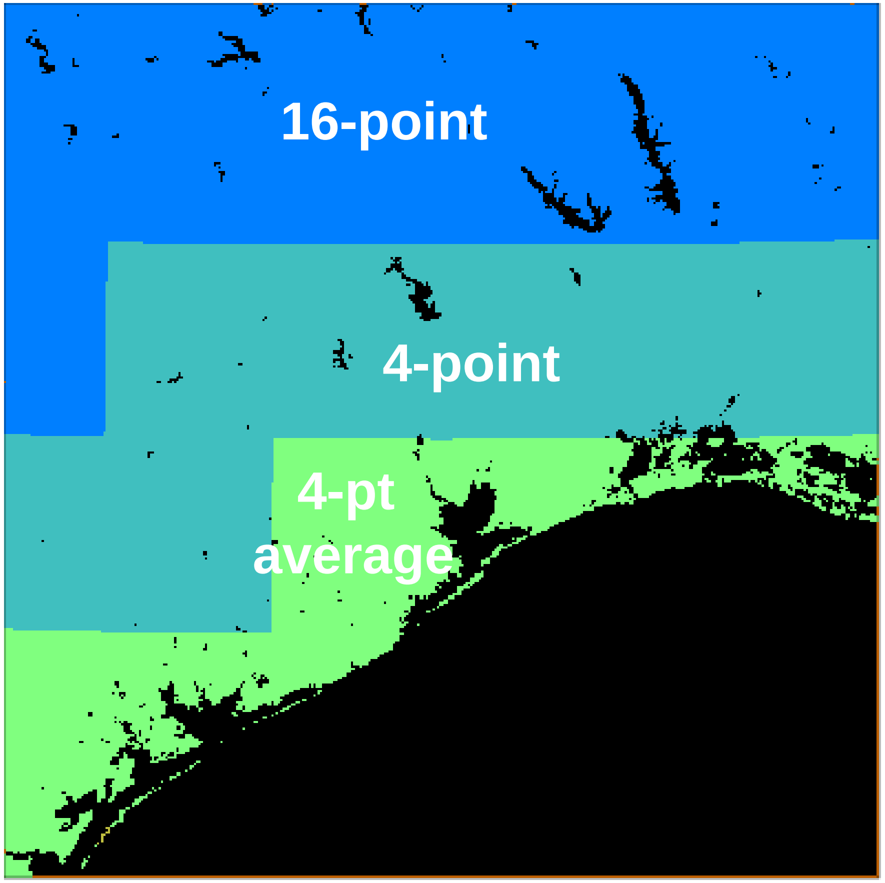
- If 16-pt doesn't work, then try 4-pt before reverting to a 4-point average
  - Note that 4-point average will work anywhere nearest_neighbor would (missing/masked values not counted in the average)

Ingesting New Fields in Metgrid

- The resulting field, below-left:



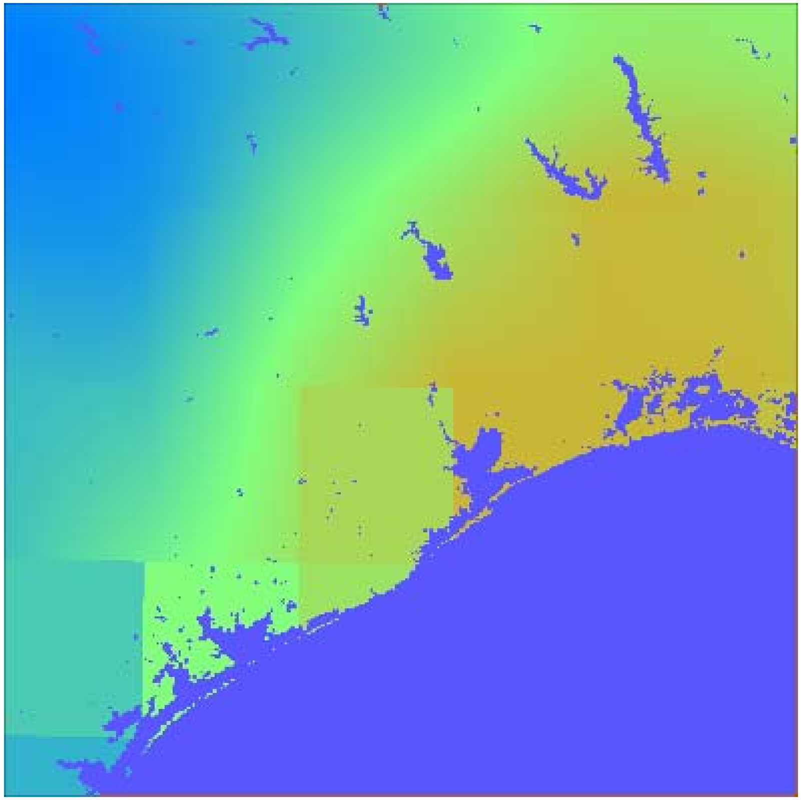
Interpolated SM000010 field (sixteen_pt + four_pt + average_4pt)



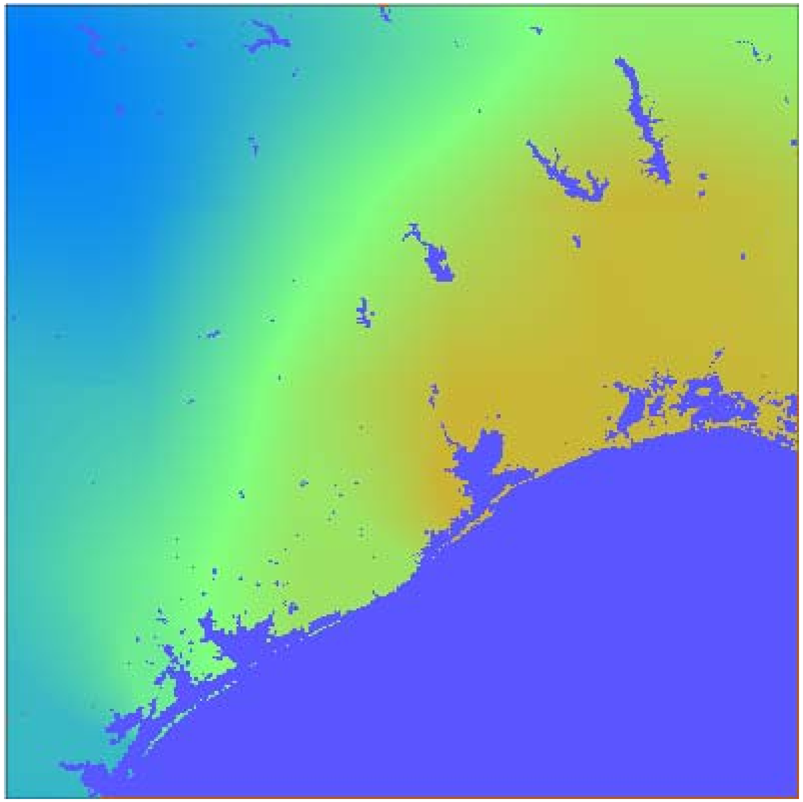
Which interpolator was used at each model grid point

Ingesting New Fields in Metgrid

- By using **`wt_average_4pt`** instead of **`average_4pt`**:



sixteen_pt + four_pt + average_4pt



sixteen_pt + four_pt + wt_average_4pt

NB: The figures above are from a different time from previous slides!

METGRID.TBL: Real-time System Example

- Suppose we have a real-time system that:
  - Uses GFS for initial and boundary conditions
  - When possible (i.e., if the files are available soon enough) uses *soil moisture* and *soil temperature* fields from AGRMET

- In our system, it may occasionally happen that the AGRMET files are not ready when we want to start our WRF run
  - Because system is real-time, we want to proceed using just the GFS land surface fields!

METGRID.TBL: Real-time System Example

- We already know how to run ungrib on multiple sources of data to get

  GFS:YYYY-MM-DD_HH

  and

  AGRMET:YYYY-MM-DD_HH

  intermediate files, and specify

  fg_name = ‘GFS’, ’AGRMET’,

  in the `&metgrid` namelist record to use both files

METGRID.TBL: Real-time System Example

Without further changes, what happens if:

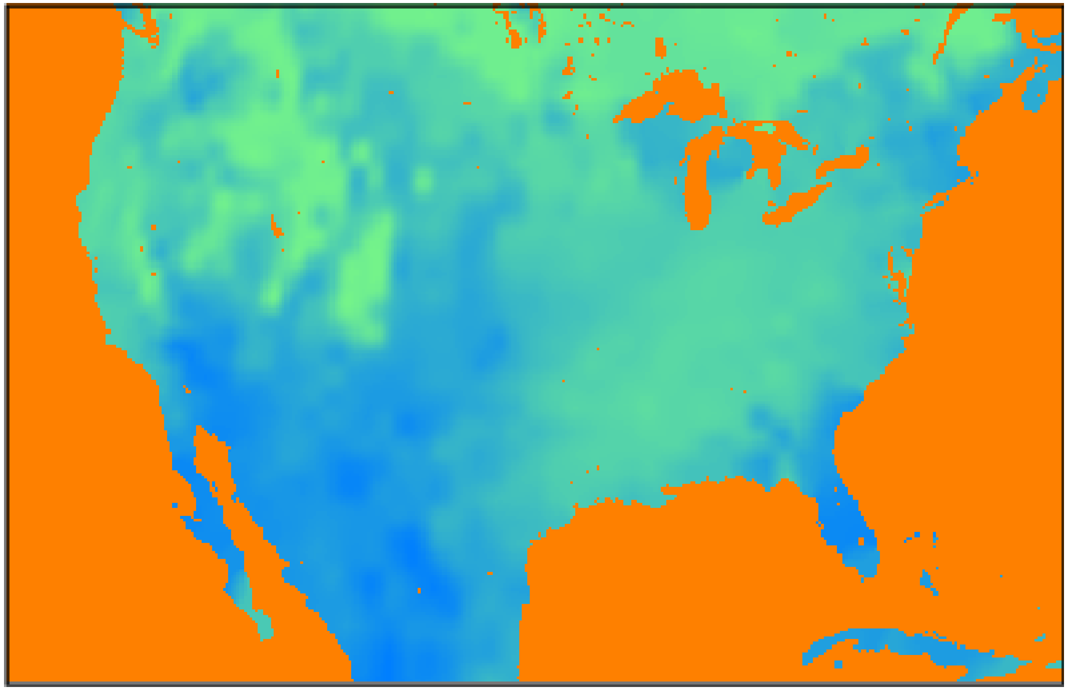
1) Only GFS data are available when we run metgrid

– Metgrid runs and warns that no AGRMET data files were found:

```
Processing 2006-04-01_00
     GFS
     AGRMET
 WARNING: Couldn't open file AGRMET:2006-04-01_00 for
 input.
```

METGRID.TBL: Real-time System Example

And the 0-10 cm soil moisture field (SM000010) looks like:

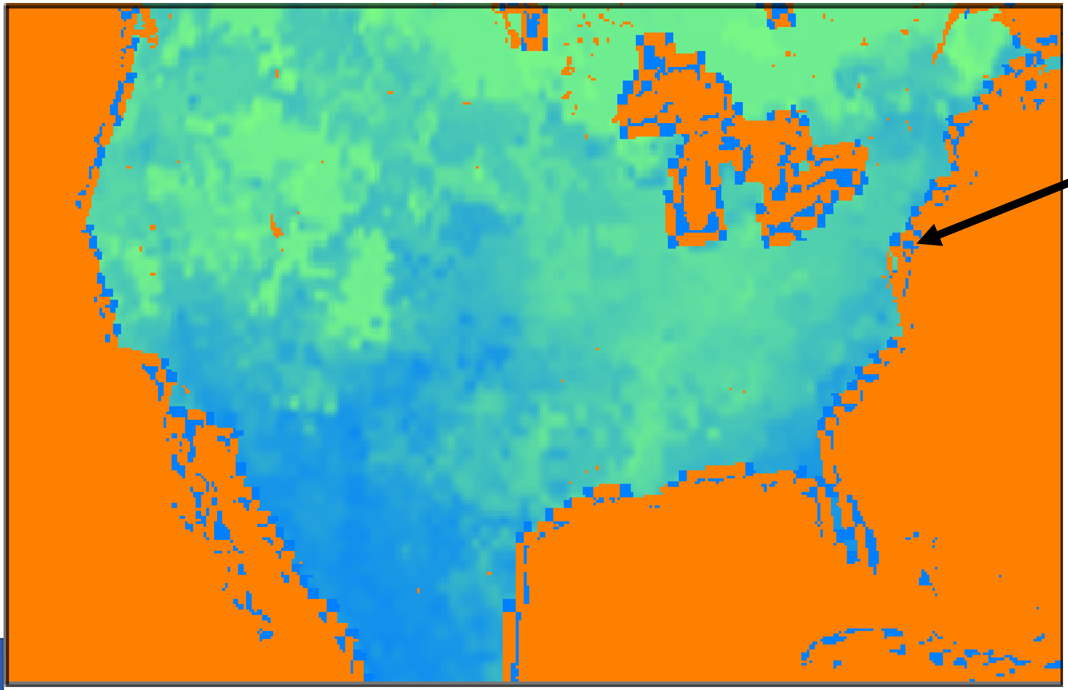


METGRID.TBL: Real-time System Example

However, what happens if:

2) Both GFS and AGRMET files are available when we run metgrid?

Our SM000010 field looks like:



We get unreasonable values with magnitude ~1E30 near land-water boundaries!

METGRID.TBL: Real-time System Example

What went wrong?

In both Vtable.GFS and Vtable.AGRMET, the land-sea mask field is named LANDSEA

- In METGRID.TBL, our entry for SM000010 says:

```
========================================
name=SM000010
interp_option=sixteen_pt+four_pt+wt_average_4pt+search
masked=water
interp_mask=LANDSEA(0)
fill_missing=1.
flag_in_output=FLAG_SM000010
========================================
```

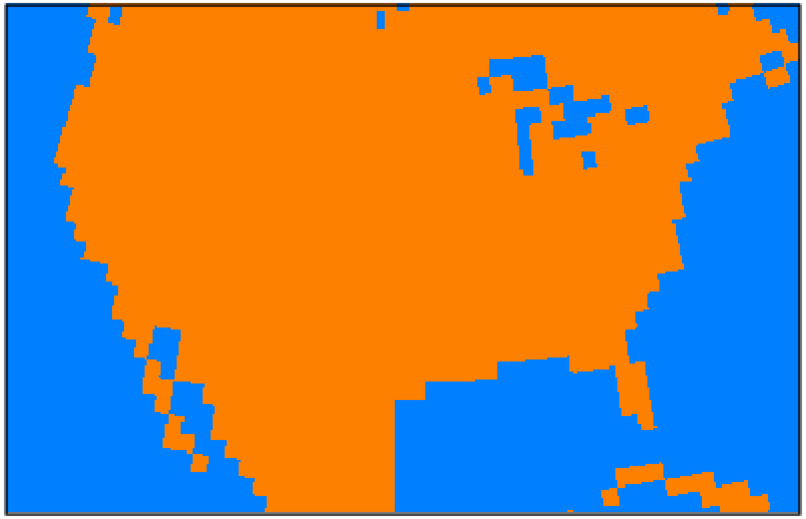
METGRID.TBL: Real-time System Example

```
========================================
name=SM000010
interp_option=sixteen_pt+four_pt+wt_average_4pt+search
masked=water
interp_mask=LANDSEA(0)
fill_missing=1.
flag_in_output=FLAG_SM000010
========================================
```

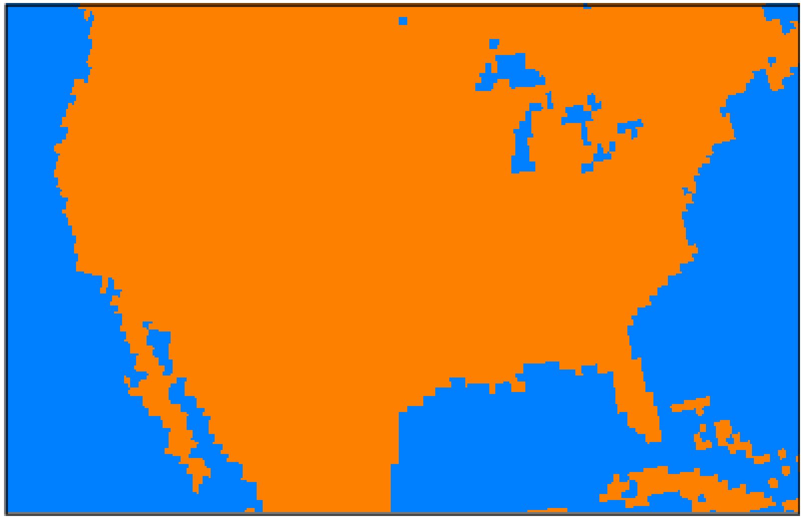
After metgrid reads in LANDSEA from GFS file *to use as an interpolation mask*, it ignored the LANDSEA field from AGRMET *for use as a mask*.

METGRID.TBL: Real-time System Example

When metgrid interpolated SM000010, it used the GFS landmask for a field masked by the AGRMET landmask!



GFS LANDSEA field



AGRMET LANDSEA field

METGRID.TBL: Real-time System Example

Solution:

- Rename LANDSEA to *AGR_LAND* in Vtable.AGRMET
- Rename LANDSEA to *GFS_LAND* in Vtable.GFS
- Create separate entries in METGRID.TBL

  one for GFS SM000010 field

  another for AGRMET SM000010 field

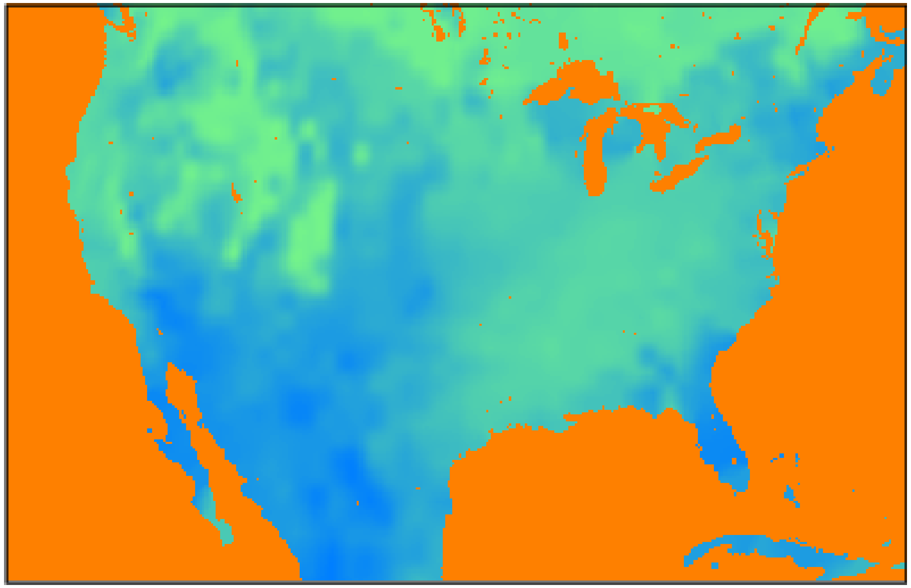
METGRID.TBL: Real-time System Example

```
========================================
name=SM000010; from_input=GFS
interp_option=sixteen_pt+four_pt+wt_average_4pt+search
masked=water
interp_mask=GFS_LAND(0)
fill_missing=1.
flag_in_output=FLAG_SM000010
========================================
```

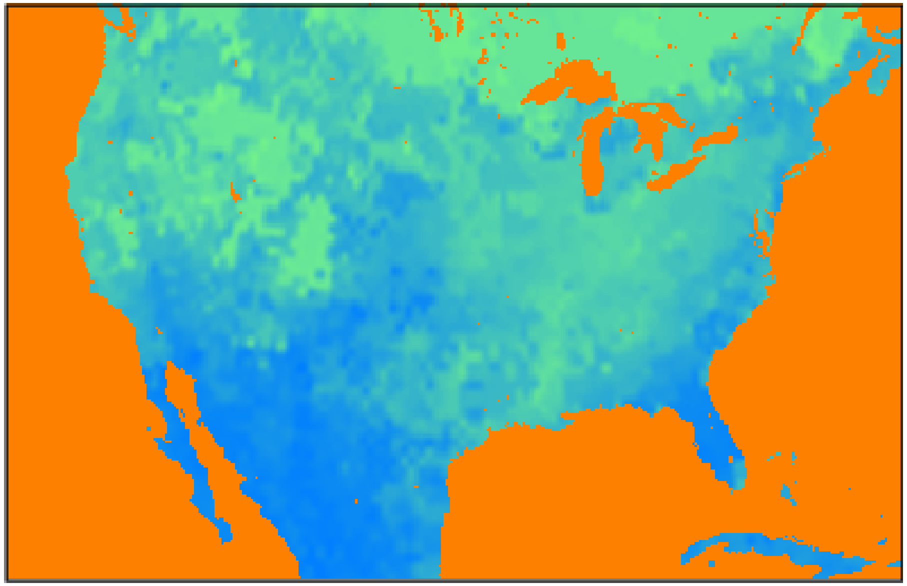
```
========================================
name=SM000010; from_input=AGRMET
interp_option=sixteen_pt+four_pt+wt_average_4pt+search
masked=water
interp_mask=AGR_LAND(-1.E30)
fill_missing=1.
flag_in_output=FLAG_SM000010
========================================
```

METGRID.TBL: Real-time System Example

With modified Vtables and METGRID.TBL:



The SM000010 field when only GFS files are available



The SM000010 field when both GFS and AGRMET files are available

WPS Utility Programs

The utility programs that come with WPS can be helpful when diagnosing problems with WPS output

- Users are encouraged to make use of these utilities to examine WPS input and output files

LIVE DEMONSTRATION OF UTILITIES

WPS Pitfalls

Some common pitfalls to look out for:

1) All 3-d fields must have same number of levels in metgrid

```
WRF_DEBUG: Warning DIM             4 , NAME num_metgrid_levels REDIFINED
  by var GHT            27           26  in wrf_io.F90 line         2347
  ERROR: Error in ext_pkg_write_field
```

– *This is usually corrected by ensuring that all 3-d meteorological fields have surface level data*

WPS Pitfalls

2) When using a regional data set (e.g., NAM), ensure that model domain is completely covered by the data

- *Points of missing data in domain will cause real.exe to fail*

3) For native vertical coordinate data sets (e.g., RUCb), ensure that pressure and geopotential height fields are available

Questions?