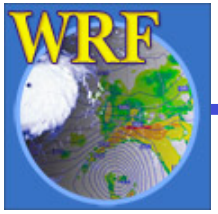
WPS – Description of General Functions

Michael Duda



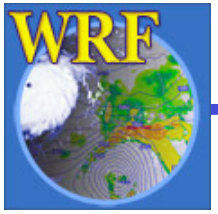
Purpose of this Lecture

In this lecture, our goals are to:

1) Understand the purpose of the WPS

2) Learn what each component of the WPS does

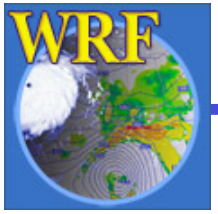
The details of *actually running* the WPS will be covered in a second lecture!



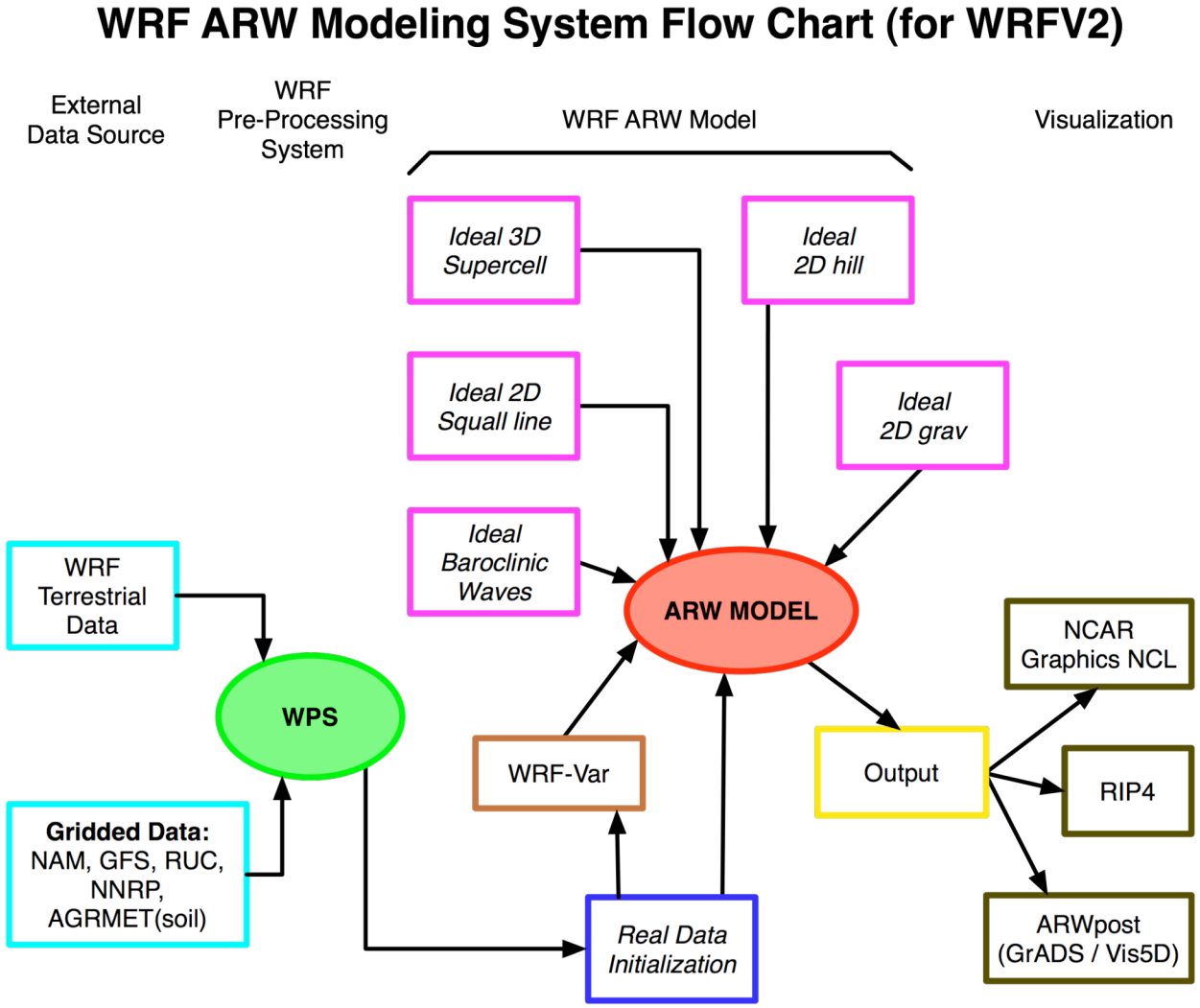
Purpose of the WPS

The purpose of the WPS is to prepare input to WRF for real-data simulations:

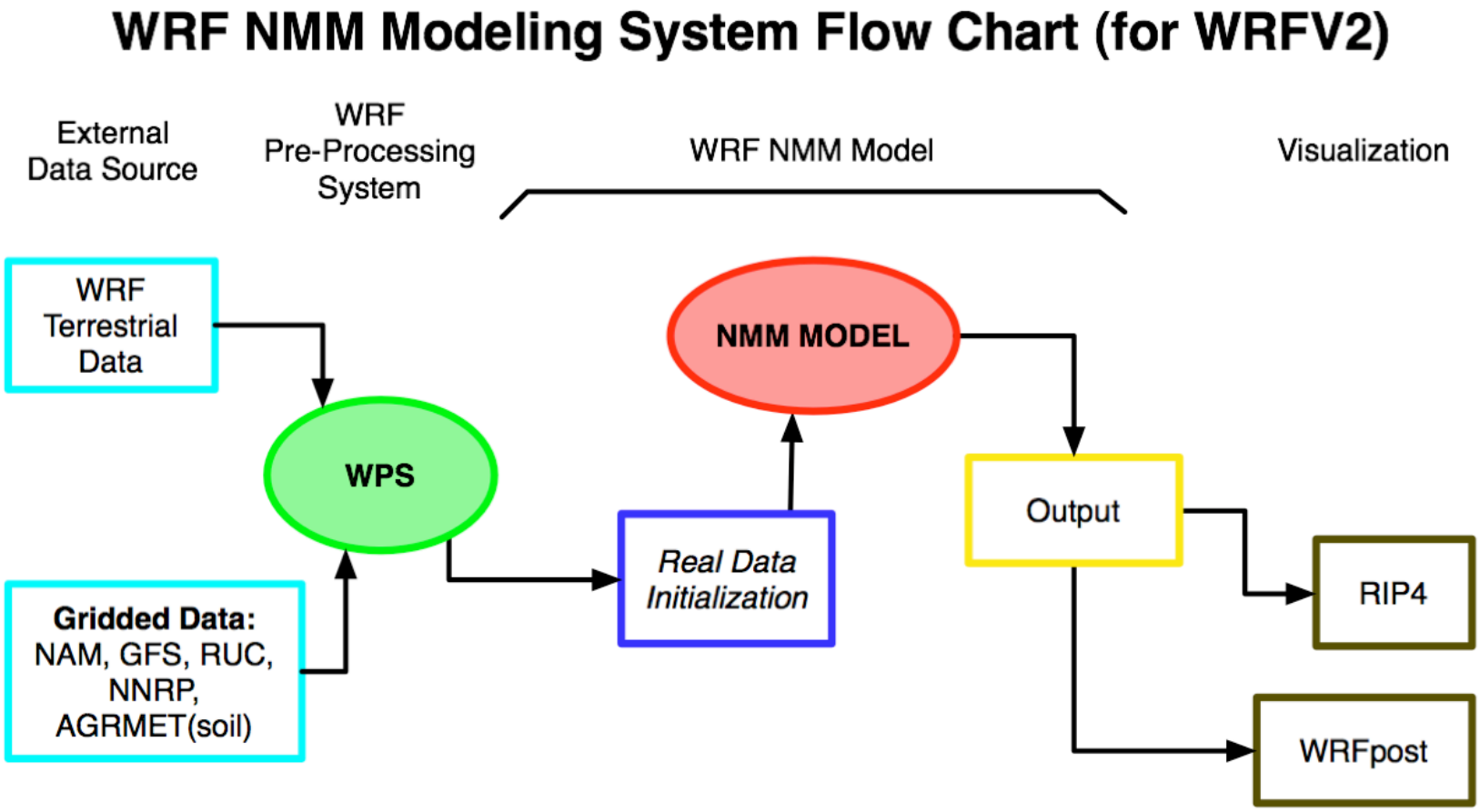
1. Defines simulation domain and nested domains
2. Computes latitude, longitude, map scale factors, Coriolis parameters at every grid point
3. Interpolates time-invariant terrestrial data to simulation grids (e.g., terrain height and soil type)
4. Interpolates time-varying meteorological fields from another model onto simulation domains



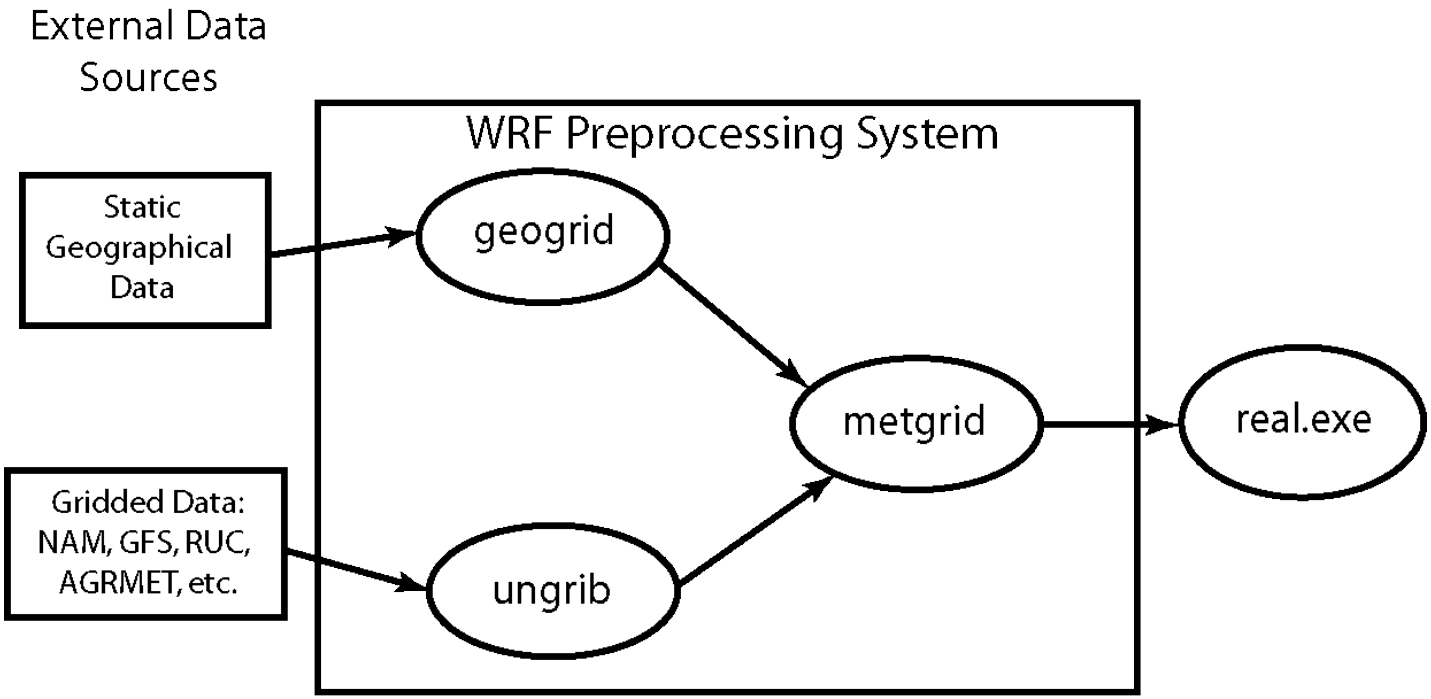
ARW Modeling System Flowchart



NMM Modeling System Flowchart



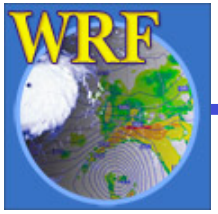
WPS Program Flowchart



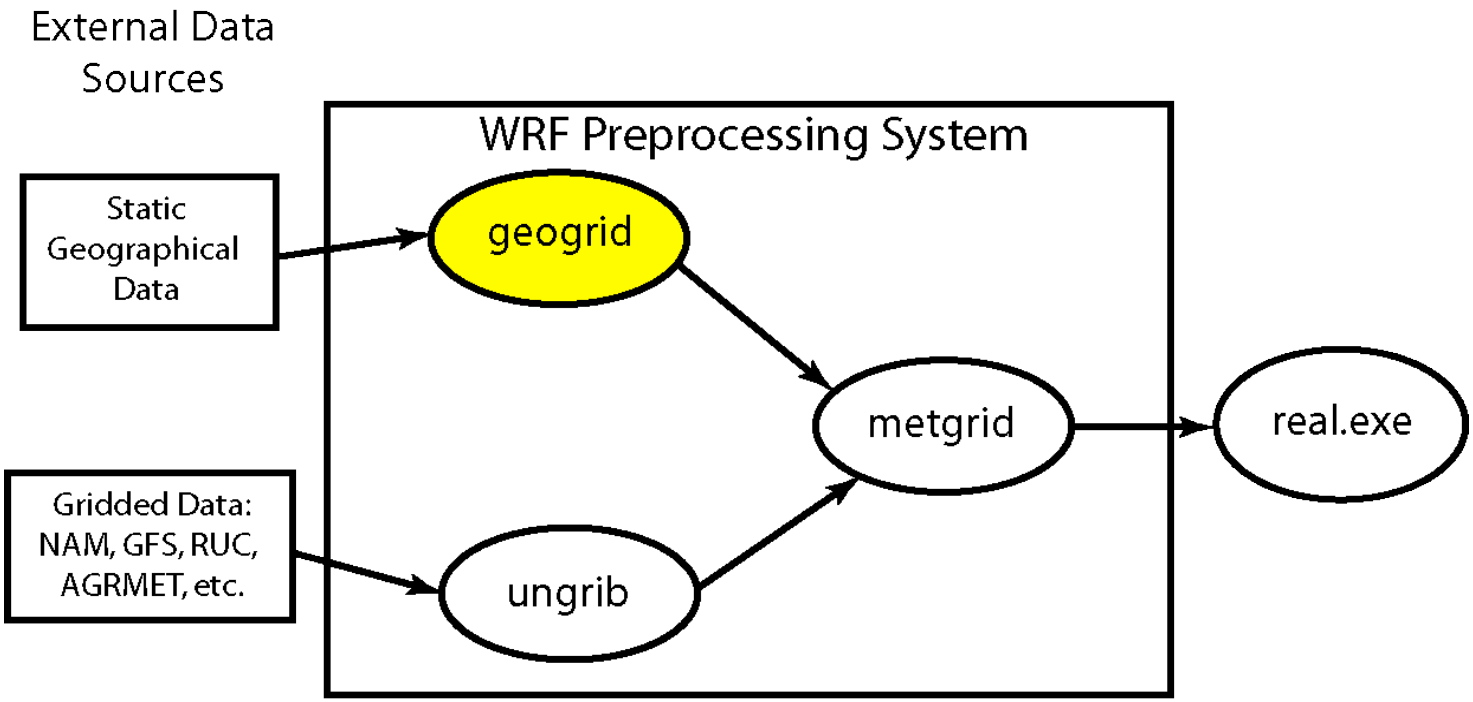
Function of WPS Components

- 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 to simulation grids

NB: Vertical interpolation takes place within real.exe

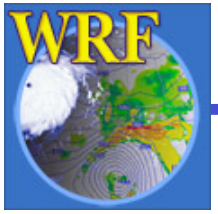


The *geogrid* program



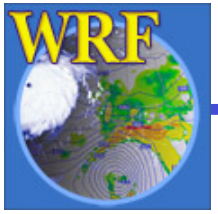
The *geogrid* program

- Define projection, location, and dimensions of simulation domains, including nested domains
- Compute latitude, longitude, map scale factor, and Coriolis parameters at each grid point
- Horizontally interpolate static terrestrial data to each grid point
  - Topography height, land use category, soil type, vegetation fraction, monthly surface albedo, etc.



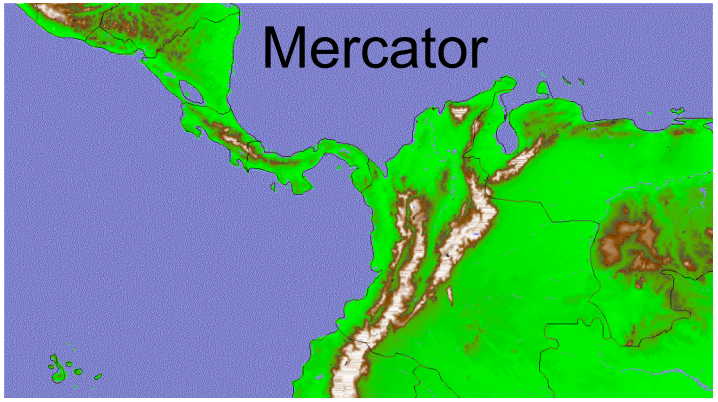
Geogrid: Defining model domains

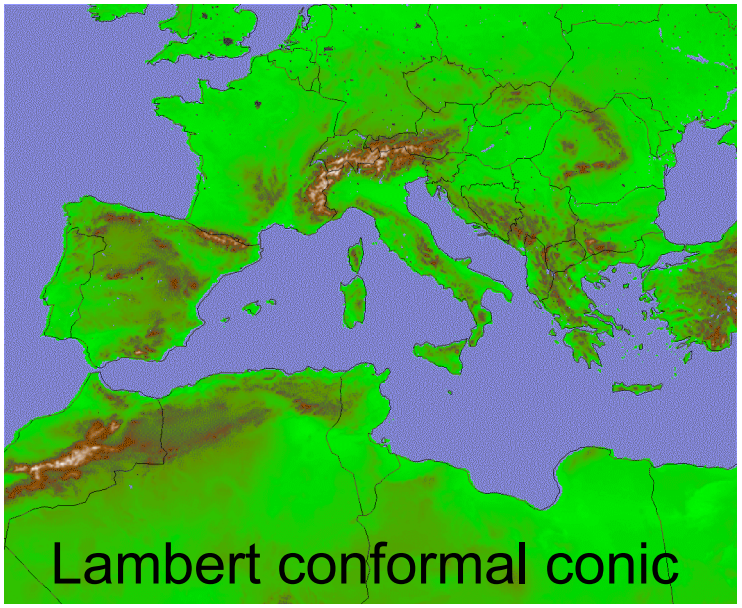
- First, we must choose a map projection to use for the domains
  - The real earth is (roughly) an ellipsoid
  - But WRF computational domains are defined by rectangles in the plane – maps
- NMM uses a rotated lat/lon projection
- ARW can use one of the following projections:
  1. Lambert conformal
  2. Mercator
  3. Polar stereographic

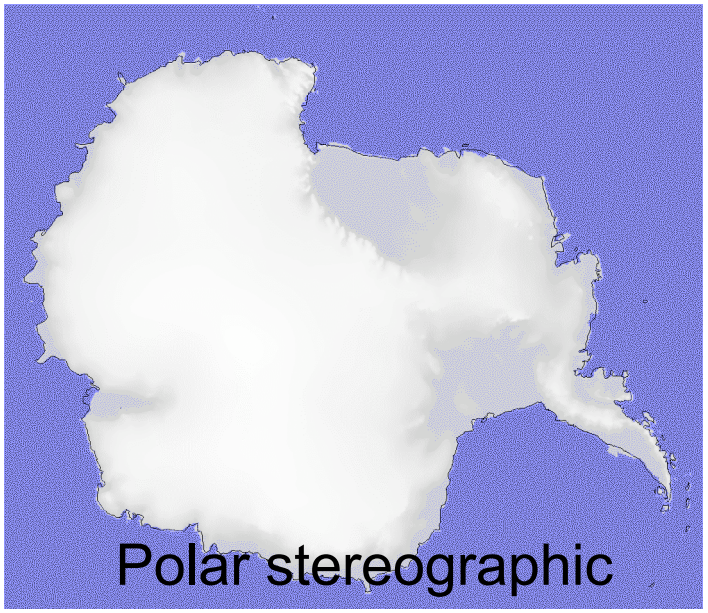


Why does ARW support 3 projections?

Rather than rotate the area of the model domain to an optimal position w.r.t. the projection plane, a projection appropriate to the domain location is chosen

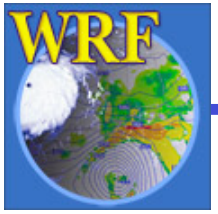






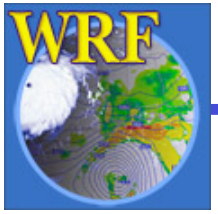
Geogrid: Defining Model Domains

- Define projection of domains with (at most) the following parameters
  - **MAP_PROJ**: 'lambert', 'mercator', 'polar', or 'rotated_ll'
  - **TRUELAT1**: First true latitude
  - **TRUELAT2**: Second true latitude (only for Lambert conformal)
  - **STAND_LON**: The meridian parallel to y-axis
- All parameters reside in the file *namelist.wps*

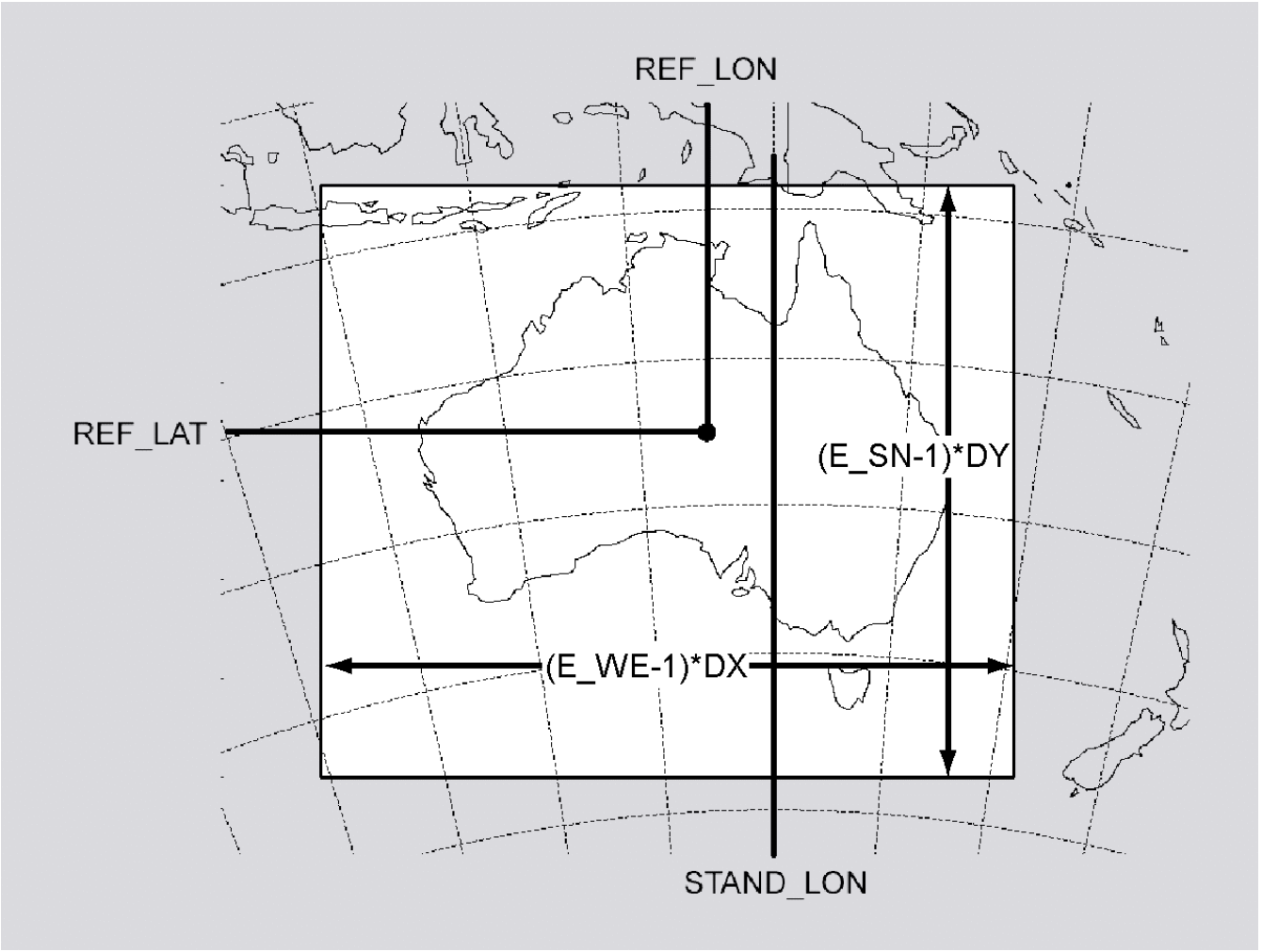


Geogrid: Defining Model Domains

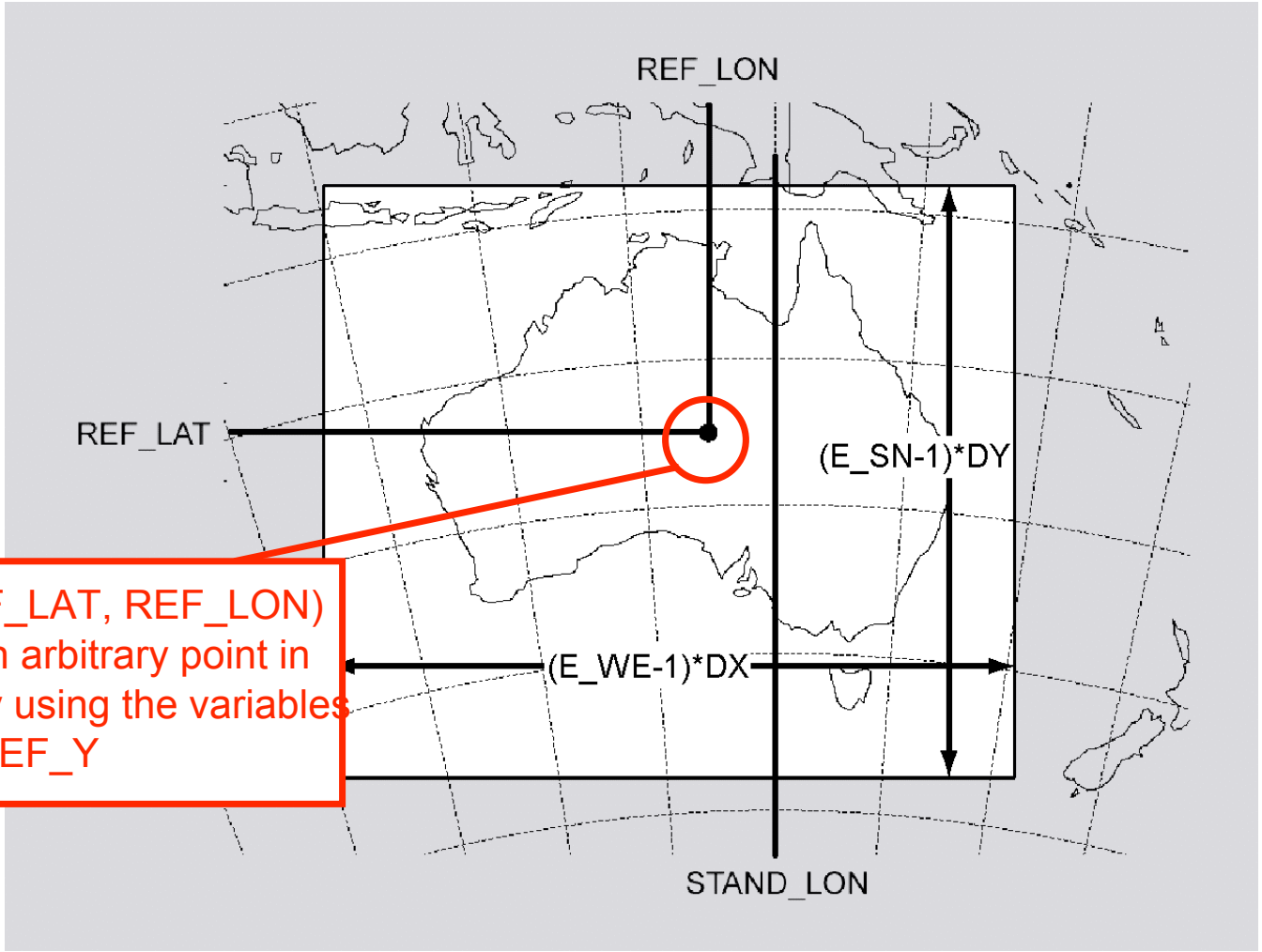
- Define the area covered (dimensions and location) by coarse domain using the following:
  - **REF_LAT**, **REF_LON**: The (lat,lon) of a known location in the domain (*by default, the center point of the domain*)
  - **DX**, **DY**: Grid distance where map factor = 1
  - **E_WE**: Number of velocity points in west-east direction for ARW; number of mass points in odd rows for NMM
  - **E_SN**: Number of velocity points in south-north direction for ARW; number of rows for NMM



Geogrid: Defining ARW Domains

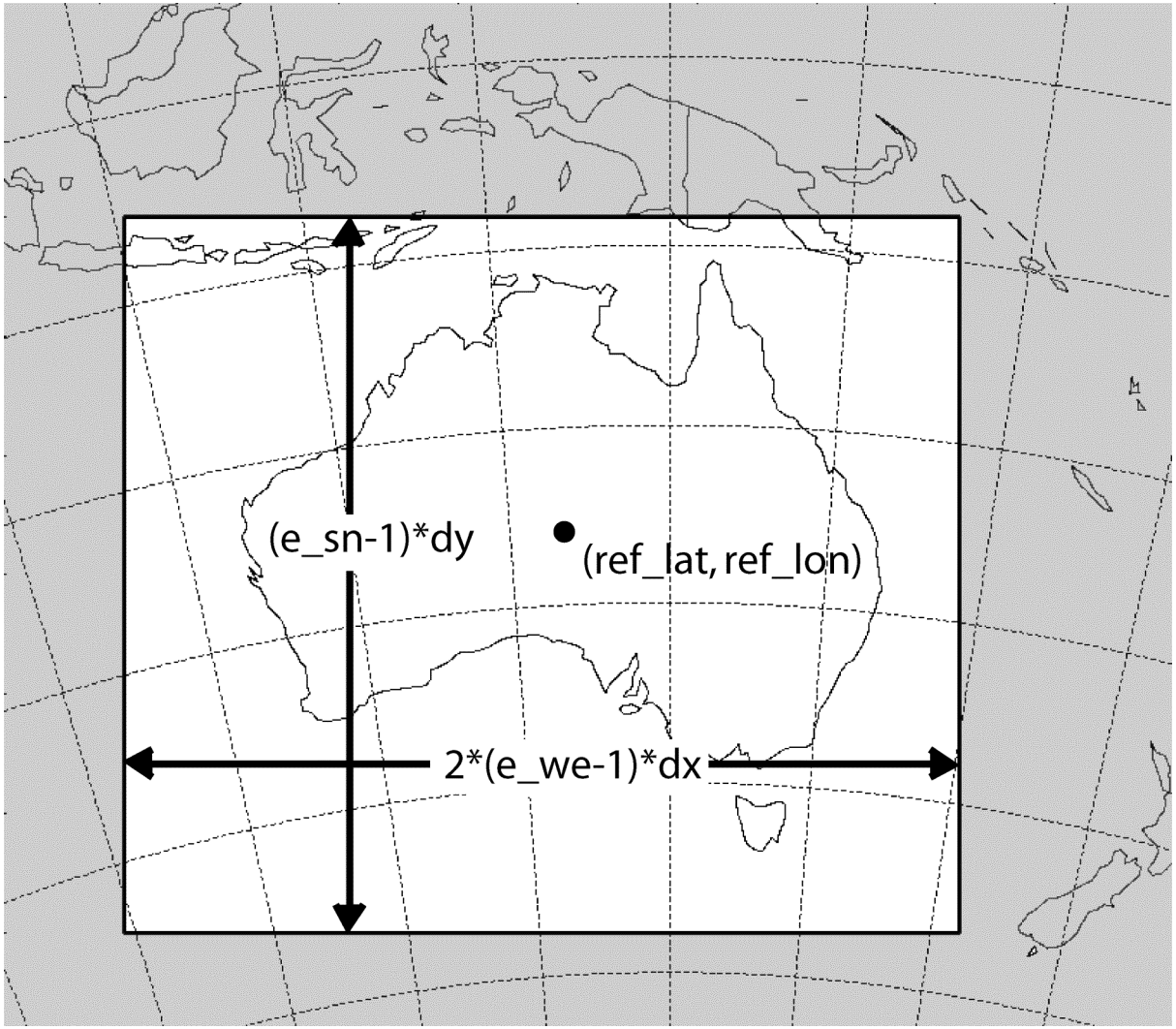


Geogrid: Defining ARW Domains



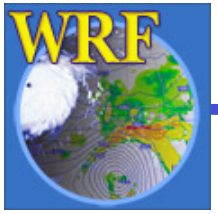
In ARW, (REF_LAT, REF_LON) can refer to an arbitrary point in the domain by using the variables REF_X and REF_Y

Geogrid: Defining NMM Domains



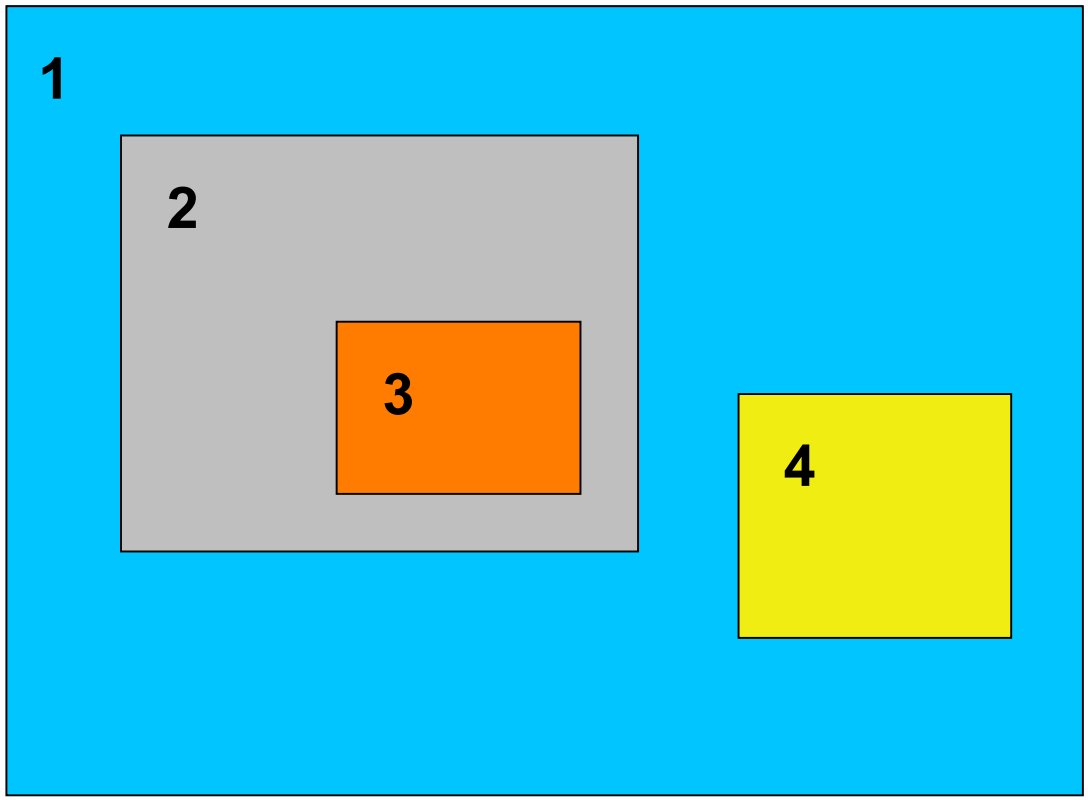
Geogrid: Nesting Basics

- A *nested domain* is a domain that is wholly contained within its *parent domain* and that receives information from its parent, and that may also feed information back to its parent
  - A nested domain has exactly one parent
  - A domain (coarse or nested) may have one or more children
- Nests *on the same nesting level* must not overlap in coverage!

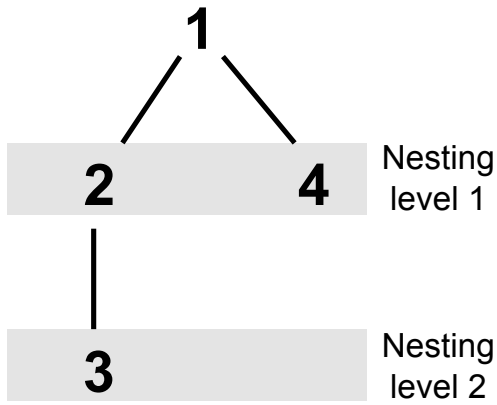


Geogrid: Nesting Example

Example configuration – 4 domains



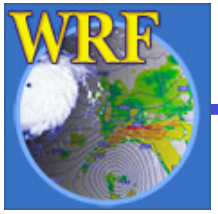
Each domain is assigned a *domain ID #*



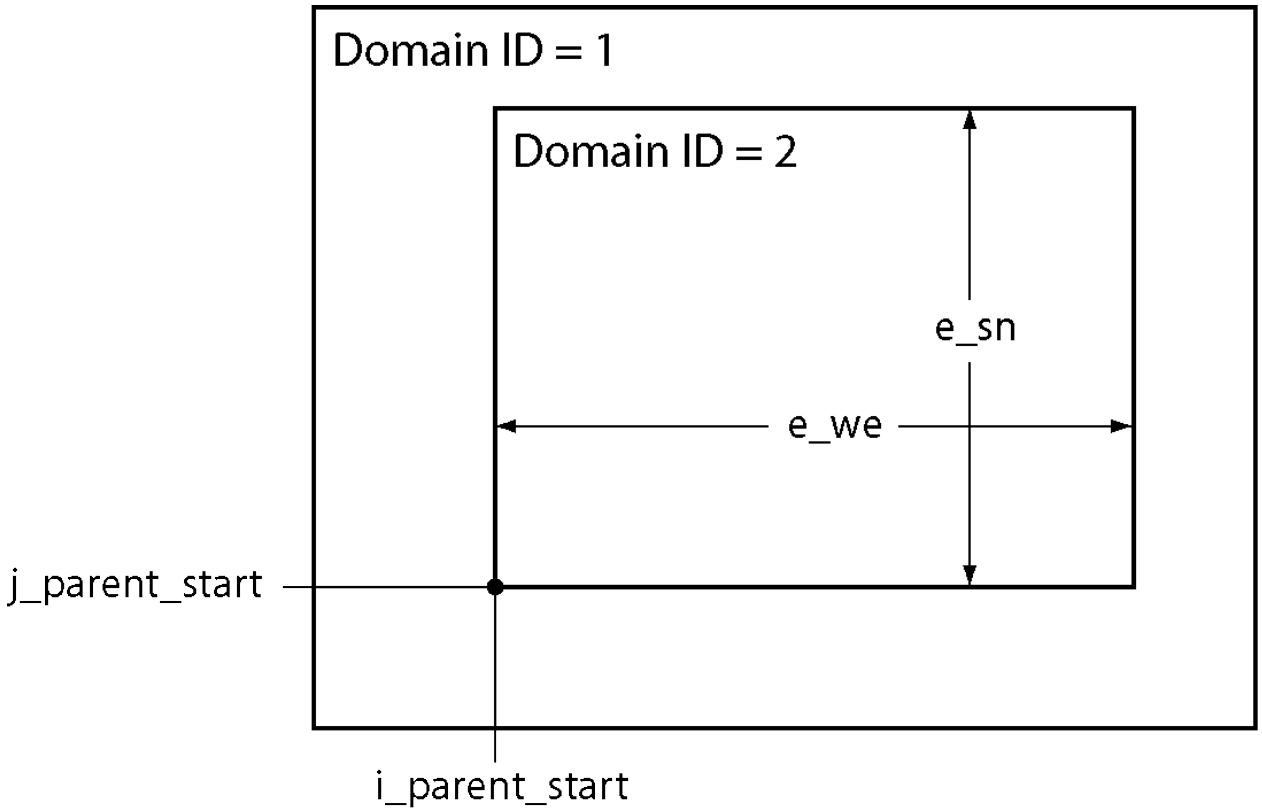
Nesting structure shown as a tree for the domains at left

Geogrid: Defining Nested Domains

- Define the dimensions and location of nested domains using:
  - **PARENT_ID**: Which domain is the parent?
  - **PARENT_GRID_RATIO**: What is the ratio between grid spacing in parent to grid spacing in this nest?
  - **I_PARENT_START**: *i*-coordinate in parent of this nest's lower-left corner
  - **J_PARENT_START**: *j*-coordinate in parent of this nest's lower-left corner
  - **E_WE**: Number of velocity points in west-east direction for ARW
  - **E_SN**: Number of velocity points in south-north direction for ARW



Geogrid: Defining Nested Domains

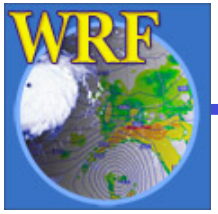


The grid spacing (*dx*) of domain 2 is determined by grid spacing of domain 1 and the *parent_grid_ratio*

NB: For NMM, the parent_grid_ratio is always 3!

Geogrid: Interpolating Static Fields

- Given definitions of all computational grids, interpolate terrestrial, time-invariant fields
  - Terrain height
  - Land use categories
  - Soil type (top & bottom layer)
  - Annual mean soil temperature
  - Monthly vegetation fraction
  - Monthly surface albedo



Geogrid: Interpolating Static Fields

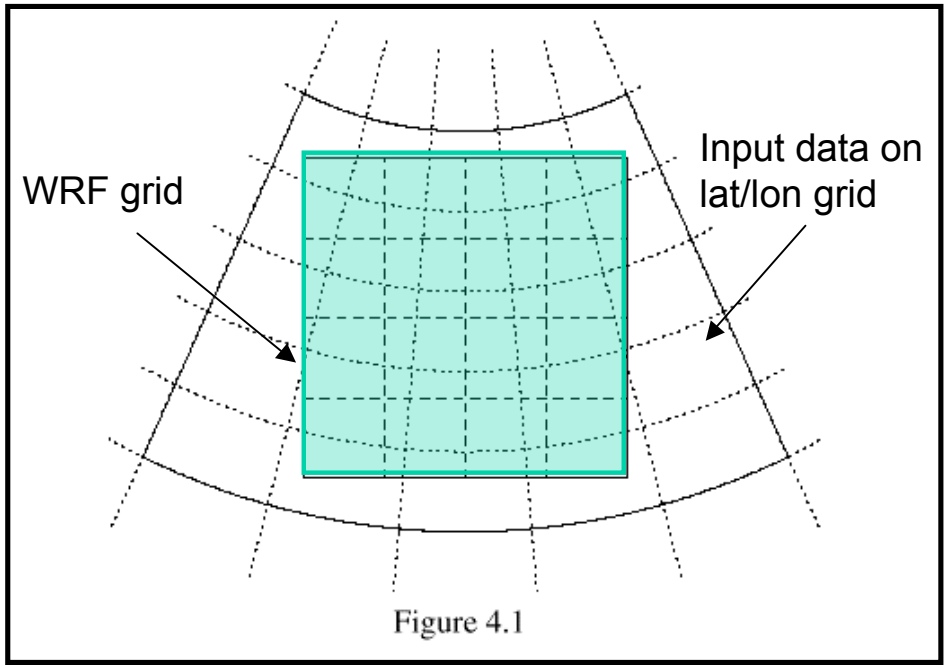
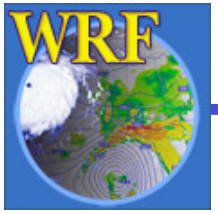


Figure 4.1

In general, source data are given on a different projection from the model grid.

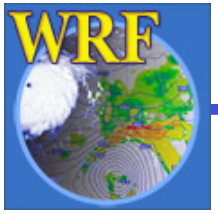
Geogrid: Interpolation Options

- 4-point bilinear
- 16-point overlapping parabolic
- 4-point average (weighted or unweighted)
- 16-point average (weighted or unweighted)
- Grid cell average
- Nearest neighbor
- Breadth-first search



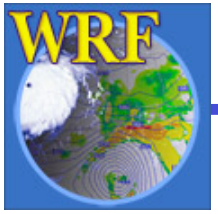
Why have so many interpolation options?

- Different interpolators work best for different fields
  - Some interpolators preserve positive definiteness
  - Some interpolators produce “smoother” fields
  - Some interpolators are best suited for discrete or categorical fields
- Having a choice of how to interpolate fields is good!
  - We’ll see in the second lecture how several different options can be used for different regions of the same field



Geogrid: Program Flexibility

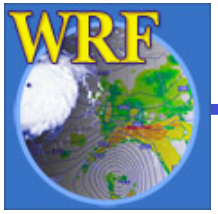
- *geogrid* is flexible enough to ingest and interpolate new static fields
  - handles either continuous or categorical fields
- New data sets must be written to simple binary format
- User needs to add an entry to the file `GEOGRID.TBL`



Geogrid: Program Flexibility

- The GEOGRID.TBL file determines
  1. Which fields will be produced by geogrid
  2. What sources of data will be used
  3. How the data will be interpolated/smoothed
  4. Any derived fields (e.g., dominant cat., df/dx)

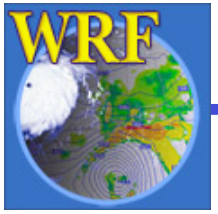
- Acceptable defaults exist in GEOGRID.TBL, so user will not generally need to edit the file (*but more on this in later lecture*!)



Geogrid: Program Flexibility

- Format of GEOGRID.TBL file is simple text, with specifications of the form <*keyword*>=<*value*>
- Example entry for new landuse data set:

```
==========================================
name=LANDUSEF
        priority=2
        dest_type=categorical
        z_dim_name=land_cat
        interp_option=30s:nearest_neighbor
        abs_path=30s:/users/duda/Houston/
==========================================
```



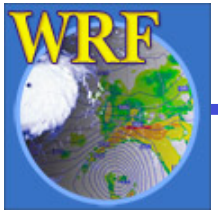
Geogrid: Program Flexibility

- The `GEOGRID.TBL` file also allows user to change which interpolation methods are used for each field
- Example:

  `interp_option=sixteen_pt`

  or

  `interp_option=four_pt+average_4pt`



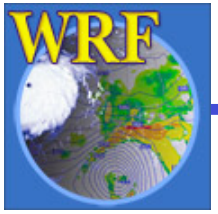
Geogrid: Program Flexibility

- Other options in the GEOGRID.TBL include smoothing options and slope calculation
- Example:

```
smooth_option=smth-desmth

smooth_passes=2
```

- More complete information on the available options may be found in Chapter 3 of the User's Guide



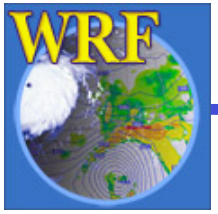
Geogrid: Program Output

- The parameters defining each domain, plus interpolated static fields, are written using the WRF I/O API
  - One file per domain for ARW
  - One file per *nesting level* for NMM
- Filenames: geo_em.d0*n*.nc, or

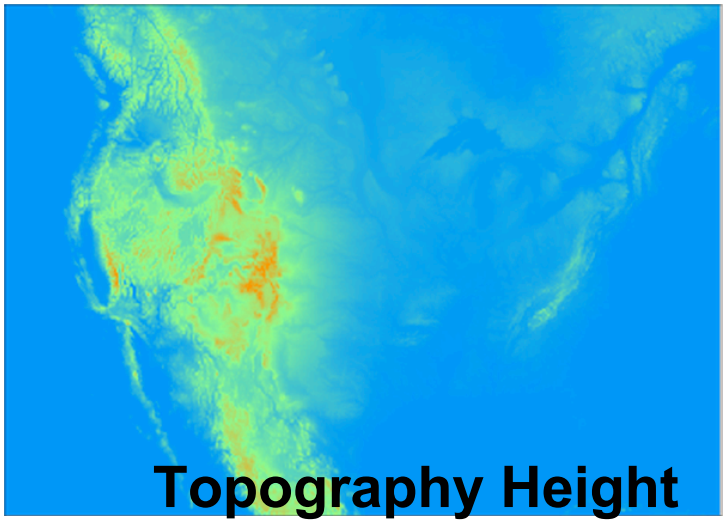
  geo_nmm.d01.nc, geo_nmm_nest.l0*k*.nc

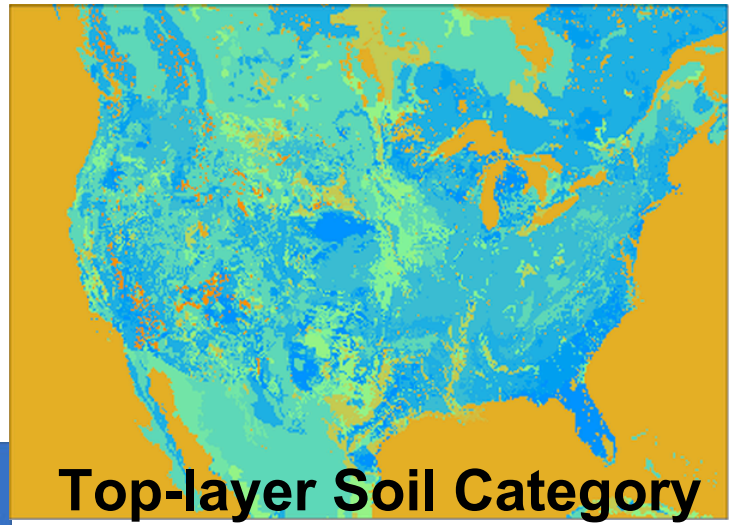
  (where *n* is the domain ID # and *k* is the nest level)
- Example:

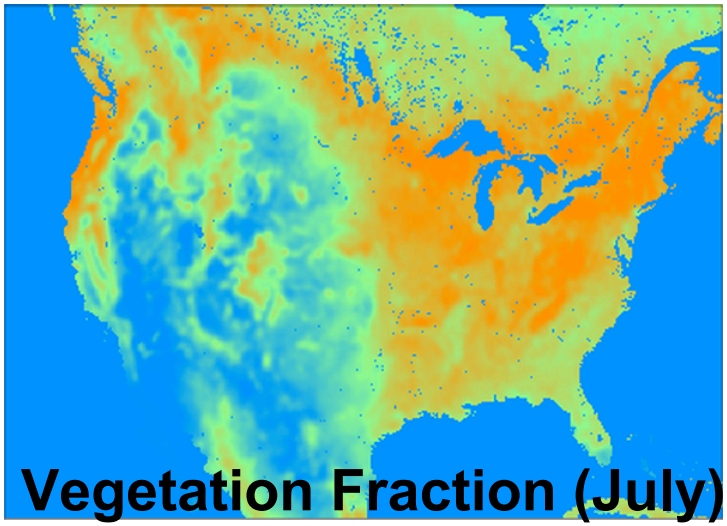
geo_em.d01.nc	geo_nmm.d01.nc
geo_em.d02.nc (nest)	geo_nmm_nest.l01.nc (nest level)
geo_em.d03.nc (nest)	geo_nmm_nest.l02.nc (nest level)



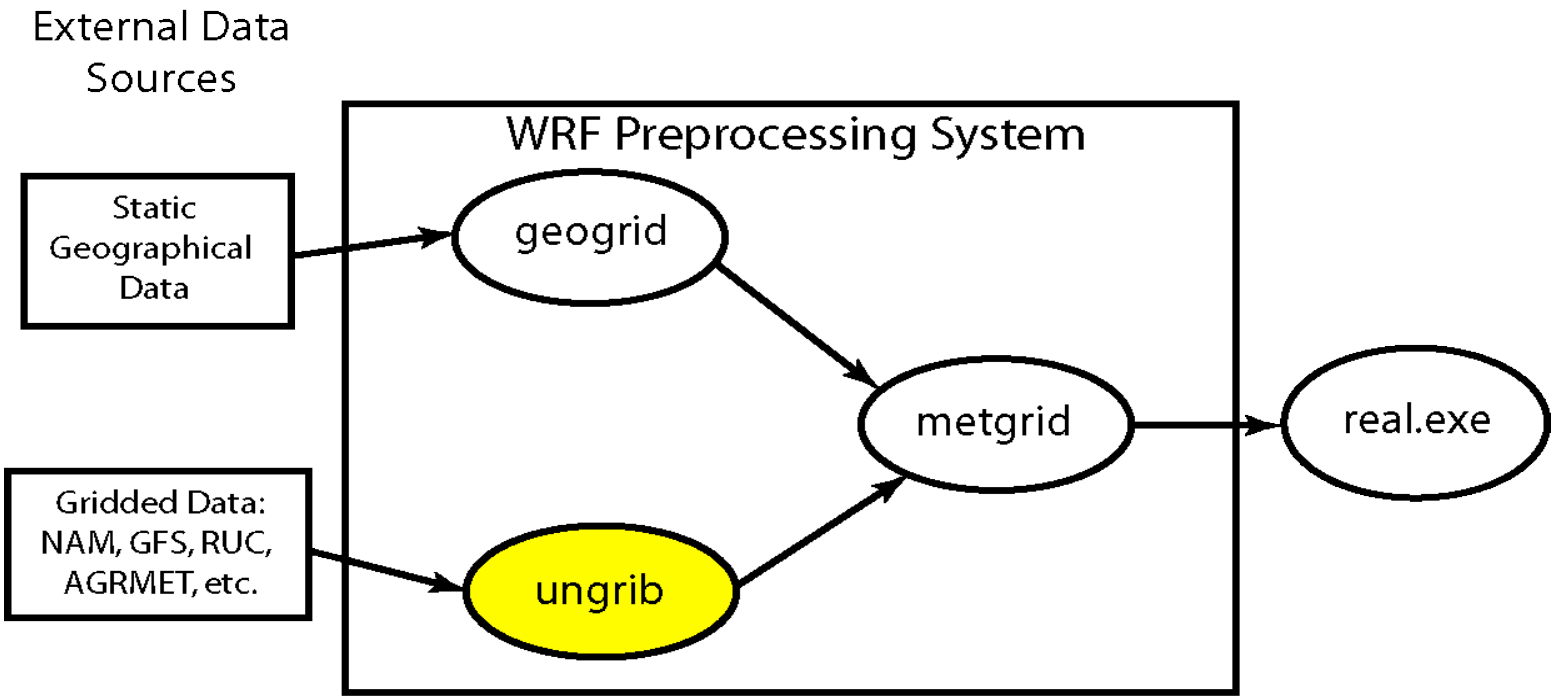
Geogrid: Example Output





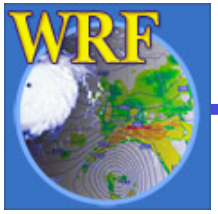


The *ungrib* program



The *ungrib* program

- Read GRIB Edition 1 and GRIB Edition 2 files
- Extract meteorological fields
- If necessary, derive required fields from related ones
  - E.g., Compute RH from specific humidity
- Write requested fields to an intermediate file format

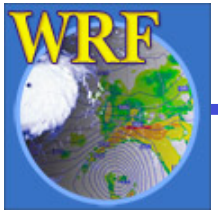


Ungrib: Vtables

How does ungrib know which fields to extract?

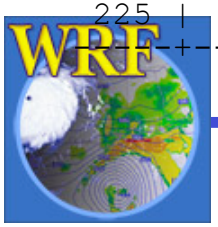
Using Vtables

- Vtables are files that give the GRIB codes for fields to be extracted from GRIB input files
- One Vtable for each source of data
- Vtables are provided for: NAM 104, NAM 212, GFS, AGRMET, and others



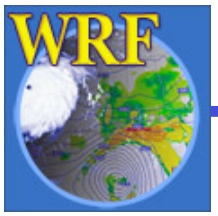
Ungrib: Example Vtable

GRIB1 Param	Level Type	From Level1	To Level2	UNGRIB Name	UNGRIB Units	UNGRIB Description
11	100	*		T	K	Temperature
33	100	*		U	m s-1	U
34	100	*		V	m s-1	V
52	100	*		RH	%	Relative Humidity
7	100	*		HGT	m	Height
11	105	2		T	K	Temperature at 2 m
52	105	2		RH	%	Relative Humidity at 2 m
33	105	10		U	m s-1	U at 10 m
34	105	10		V	m s-1	V at 10 m
1	1	0		PSFC	Pa	Surface Pressure
130	102	0		PMSL	Pa	Sea-level Pressure
144	112	0	10	SM000010	kg m-3	Soil Moist 0-10 cm below grn layer (Up)
144	112	10	40	SM010040	kg m-3	Soil Moist 10-40 cm below grn layer
144	112	40	100	SM040100	kg m-3	Soil Moist 40-100 cm below grn layer
144	112	100	200	SM100200	kg m-3	Soil Moist 100-200 cm below gr layer
85	112	0	10	ST000010	K	T 0-10 cm below ground layer (Upper)
85	112	10	40	ST010040	K	T 10-40 cm below ground layer (Upper)
85	112	40	100	ST040100	K	T 40-100 cm below ground layer (Upper)
85	112	100	200	ST100200	K	T 100-200 cm below ground layer (Bottom)
91	1	0		SEAICE	proprtn	Ice flag
81	1	0		LANDSEA	proprtn	Land/Sea flag (1=land,2=sea in GRIB2)
7	1	0		HGT	m	Terrain field of source analysis
11	1	0		SKINTEMP	K	Skin temperature (can use for SST also)
65	1	0		SNOW	kg m-2	Water equivalent snow depth
223	1	0		CANWAT	kg m-2	Plant Canopy Surface Water
224	1	0		SOILCAT	Tab4.213	Dominant soil type category
225	1	0		VEGCAT	Tab4.212	Dominant land use category



Ungrib: GRIB2 Vtable Entries

metgrid Description	GRIB2 Discp	GRIB2 Catgy	GRIB2 Param	GRIB2 Level
Temperature	0	0	0	100
U	0	2	2	100
V	0	2	3	100
Relative Humidity	0	1	1	100
Height	0	3	5	100
Temperature at 2 m	0	0	0	103
Relative Humidity at 2 m	0	1	1	103
U at 10 m	0	2	2	103
V at 10 m	0	2	3	103
Surface Pressure	0	3	0	1
Sea-level Pressure	0	3	1	101
Soil Moist 0-10 cm below grn layer (Up)	2	0	192	106
Soil Moist 10-40 cm below grn layer	2	0	192	106
Soil Moist 40-100 cm below grn layer	2	0	192	106
Soil Moist 100-200 cm below gr layer	2	0	192	106
Soil Moist 10-200 cm below gr layer	2	0	192	106
T 0-10 cm below ground layer (Upper)	0	0	0	106
T 10-40 cm below ground layer (Upper)	0	0	0	106
T 40-100 cm below ground layer (Upper)	0	0	0	106
T 100-200 cm below ground layer (Bottom)	0	0	0	106
T 10-200 cm below ground layer (Bottom)	0	0	0	106
Ice flag	0	2	0	1
Land/Sea flag (1=land, 0 or 2=sea)	2	0	0	1
Terrain field of source analysis	2	0	7	1
Skin temperature (can use for SST also)	0	0	0	1
Water equivalent snow depth	0	1	13	1
Dominant soil type cat.(not in GFS file)	2	3	0	1
Dominant land use cat. (not in GFS file)	2	0	198	1

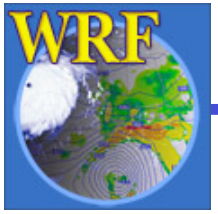


Ungrib: Vtables

What if a data source has no existing Vtable?

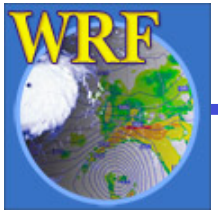
Create a Vtable

- Get a listing of GRIB codes for fields in the source
  - Check documentation from originating center or use utility such as *wgrib*
- Use existing Vtable as a template
- Check documentation in Chapter 3 of the Users' Guide for more information about Vtables



Ungrib: Intermediate File Format

- After extracting fields listed in Vtable, ungrib writes those fields to intermediate format
- For meteorological data sets not in GRIB format, the user may write to intermediate format directly
  - *Allows WPS to ingest new data sources*; basic programming required of user
  - Simple intermediate file format is easily read/written



Ungrib: Program Output

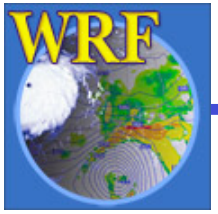
- Output files named *FILE:YYYY-MM-DD_HH*
  - *YYYY* is year of data in the file; *MM* is month; *DD* is day; *HH* is hour
  - All times are UTC
- Example:

  FILE:2007-07-24_00

  FILE:2007-07-24_12

  FILE:2007-07-25_00

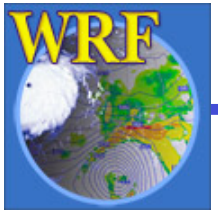
ungrib can also write intermediate files in the MM5 or WRF SI format! *(To allow for use of GRIB2 data with MM5, for example)*



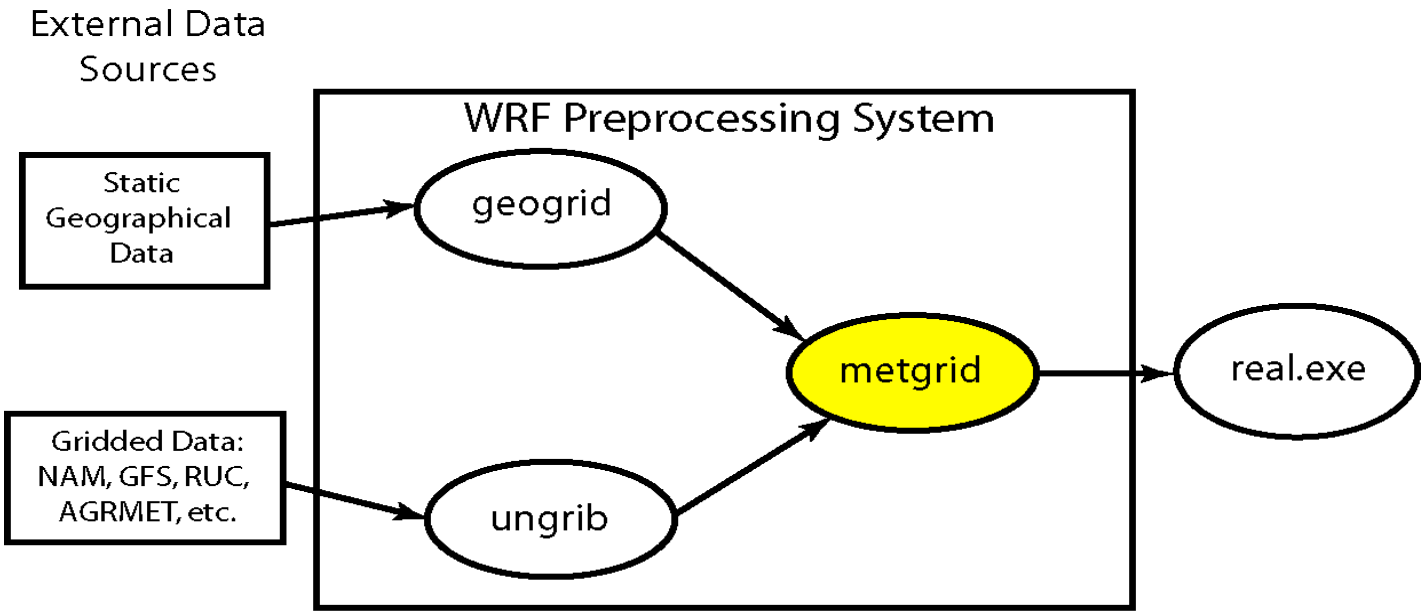
Ungrib: Obtaining GRIB Data

- Where does one get GriB data?
  - User's responsibility
  - Some free data are available from NCAR and NCEP. See
  - http://www.mmm.ucar.edu/wrf/users/

    > *Download*
    - Some NCEP data in the past year
    - NCEP operational data available daily

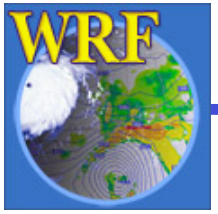


The *metgrid* program



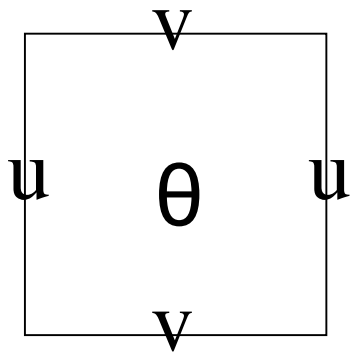
The *metgrid* program

- Horizontally interpolate meteorological data (*extracted by ungrib*) to simulation domains (*defined by geogrid*)
  - Masked interpolation for masked fields
- Rotate winds to WRF grid
  - i.e., rotate so that U-component is parallel to *x*-axis, V-component is parallel to *y*-axis



Metgrid: ARW Grid Staggering

- For ARW, wind U-component interpolated to “u” staggering
- Wind V-component interpolated to “v” staggering
- Other meteorological fields interpolated to “θ” staggering by default (*can change this!*)



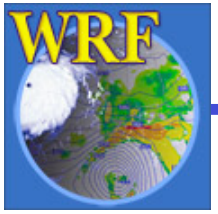
A single ARW grid cell, with “u”, “v”, and “θ” points labeled.

Metgrid: NMM Grid Staggering

- For NMM, wind U- and V-components interpolated to “V” staggering
- Other meteorological fields interpolated to “H” staggering by default (*can change this!*)

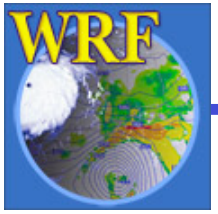
H	V	H	V	H
V	H	V	H	V
H	V	H	V	H

An NMM grid showing “V”, and “H” points.



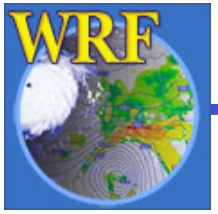
Metgrid: Interpolation Options

- 4-point bilinear
- 16-point overlapping parabolic
- 4-point average (weighted or unweighted)
- 16-point average (weighted or unweighted)
- Grid cell average
- Nearest neighbor
- Breadth-first search

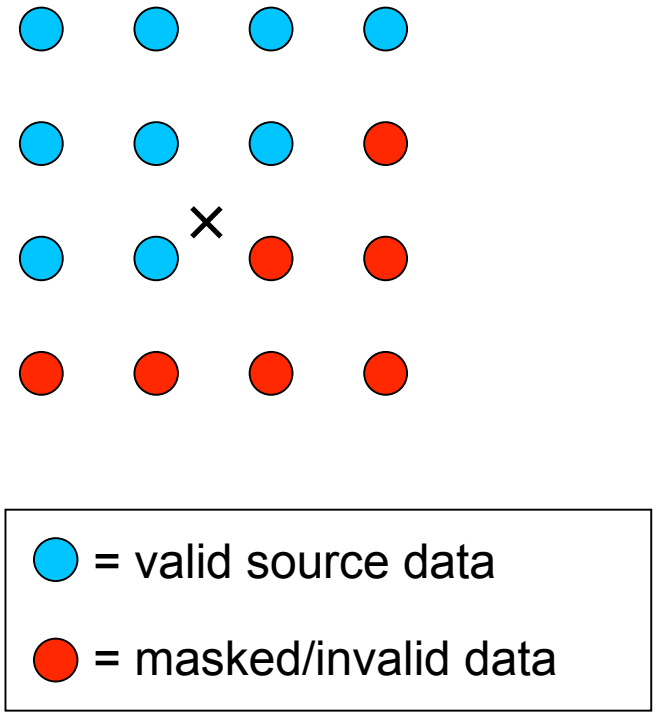


Metgrid: Masked Interpolation

- *Masked fields* may only have valid data at a subset of grid points
  - Ex: SST field only valid on water points
- When metgrid interpolates masked fields, it must know which points are invalid (masked)
  - Can use separate mask field (e.g., LANDSEA)
  - Can rely on special values (e.g., 1×10^{30}) in field itself to identify masked grid points



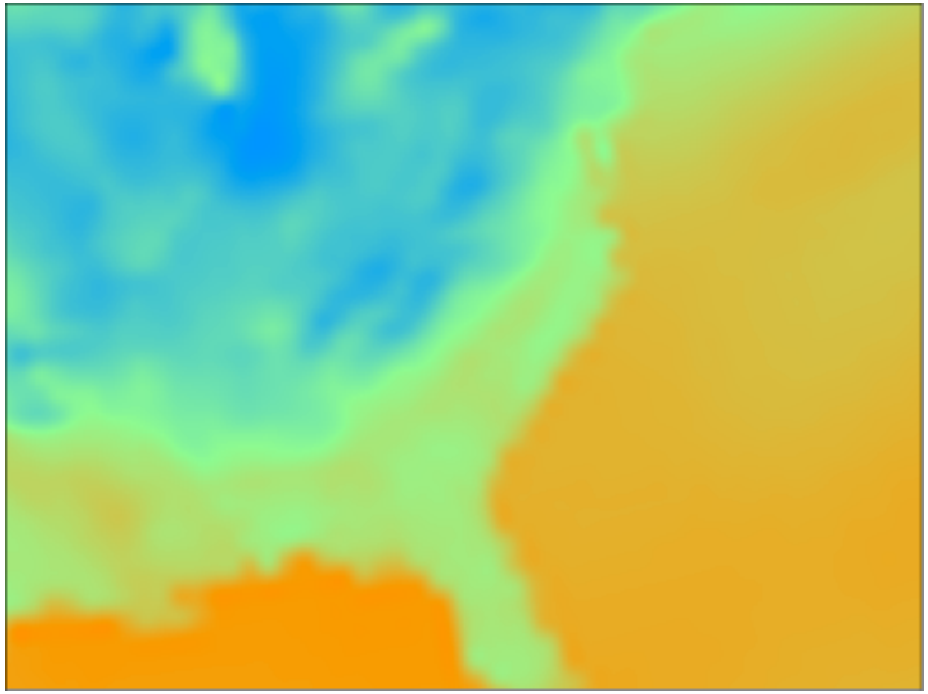
Metgrid: Masked Interpolation



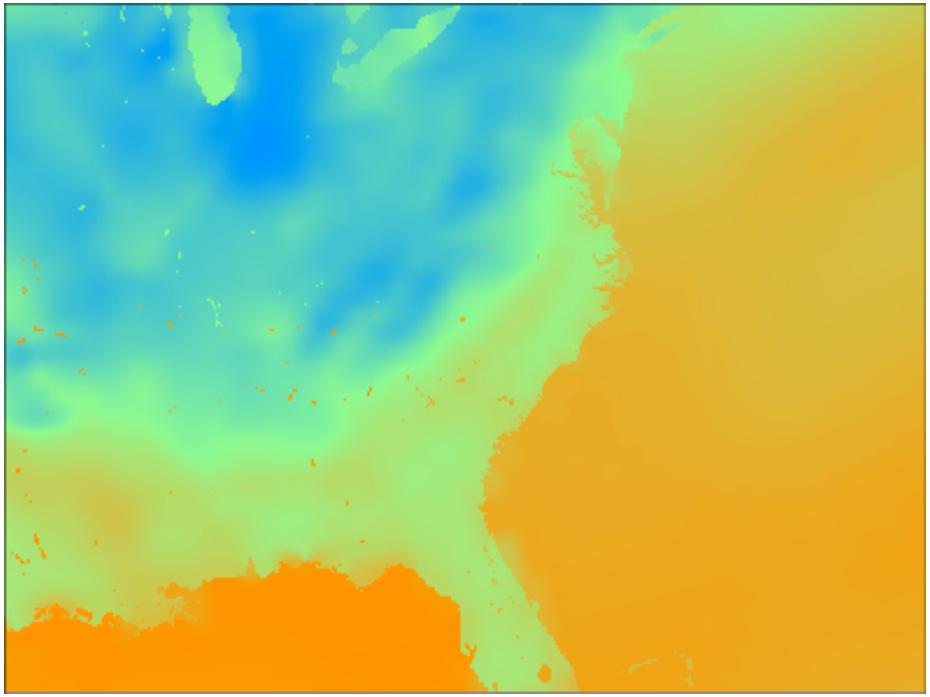
Suppose we need to interpolate to point X

- Using red points as valid data can give a bad interpolated value!
- Masked interpolation only uses valid blue points to interpolate to X

Example: Masked Interpolation



Skin temperature field interpolated from GFS 0.5-deg field with no mask using a sixteen-point interpolator.

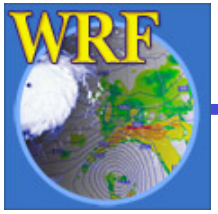


Skin temperature field interpolated using masks: GFS water points interpolated to model water points, GFS land points interpolated to model land points.

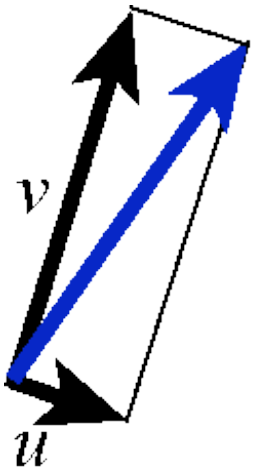
Metgrid: Wind Rotation

- Input wind fields (U-component + V-component) are either:
  - **Earth-relative**: U-component = westerly component; V-component = southerly component
  - **Relative to source grid**: U-component (V-component) parallel to source model x-axis (y-axis)

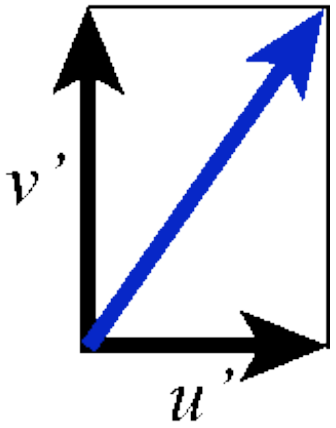
- WRF expects wind components to be relative to the simulation grid



Metgrid: Wind Rotation Example



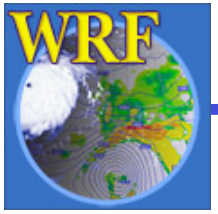
A wind vector, shown in terms of its U and V components with respect to the source grid.



The same vector, in terms of its U and V components with respect to the WRF simulation grid.

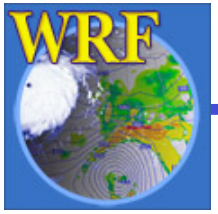
Metgrid: Constant Fields

- For short simulations, some fields may be constant
  - E.g., SST or sea-ice fraction
- Use namelist option `CONSTANTS_NAME` option to specify such fields:
  - `CONSTANTS_NAME = 'SST_FILE:2007-07-24_00'`



Metgrid: Program Flexibility

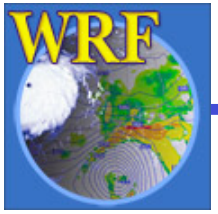
- *metgrid* is capable of interpolating both isobaric and native vertical coordinate data sets
- User may specify interpolation methods and related options in the `METGRID.TBL` file
  - `METGRID.TBL` file similar in format to the file `GEOGRID.TBL`



Metgrid: Program Flexibility

- Example METGRID.TBL entry (for “soil moisture 0-10 cm”)

```
===================================
name=SM000010
interp_option=sixteen_pt+four_pt+average_4pt
masked=water
interp_mask=LANDSEA(0)
fill_missing=1.
flag_in_output=FLAG_SM000010
===================================
```



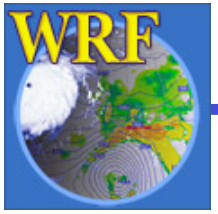
Metgrid: Program Output

- For coarse domain, one file per time period
  - In ARW, we also get the first time period for all nested grids
- Files contain static fields from geogrid plus interpolated meteorological fields
- Filenames:

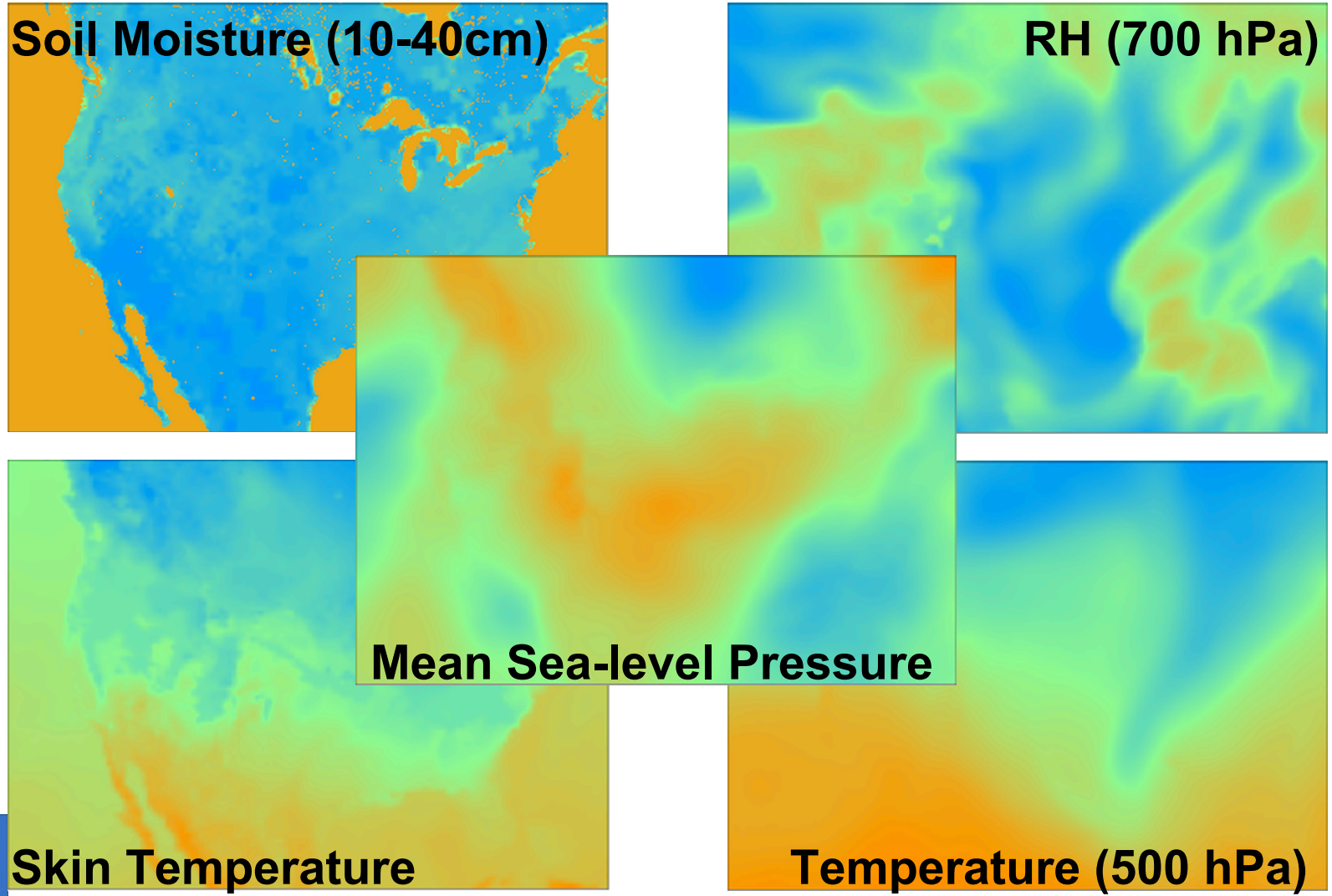
  ARW: `met_em.d0`*`n`*`.`*`YYYY-MM-DD_HH`*`.nc`

  (where *n* is the domain ID #)

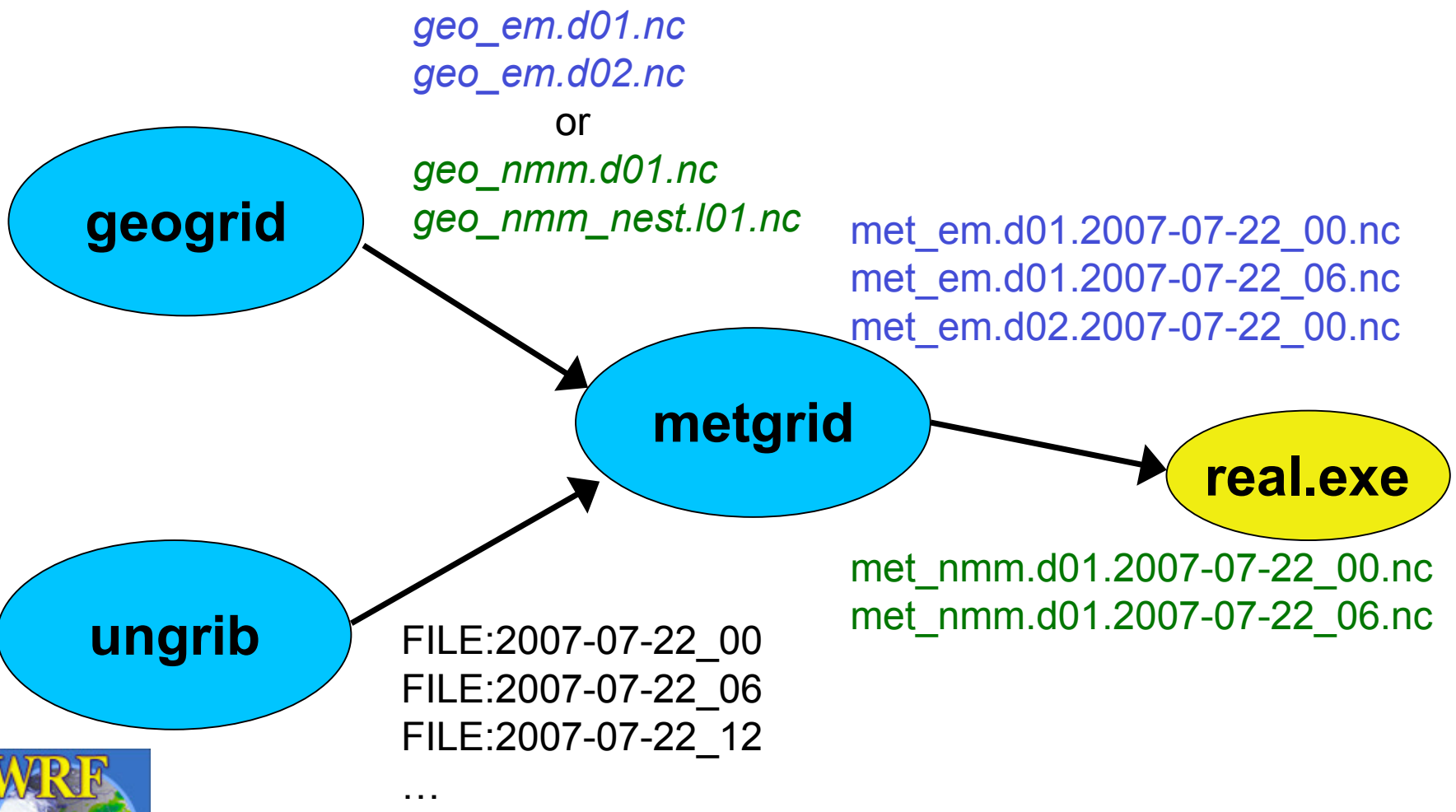
  NMM: `met_nmm.d01.`*`YYYY-MM-DD_HH`*`.nc`



Metgrid: Example Output



WPS Summary


geo_em.d01.nc
geo_em.d02.nc
or
geo_nmm.d01.nc
geo_nmm_nest.l01.nc
geogrid
met_em.d01.2007-07-22_00.nc
met_em.d01.2007-07-22_06.nc
met_em.d02.2007-07-22_00.nc
metgrid
real.exe
met_nmm.d01.2007-07-22_00.nc
met_nmm.d01.2007-07-22_06.nc
ungrib
FILE:2007-07-22_00
FILE:2007-07-22_06
FILE:2007-07-22_12
…

Questions?