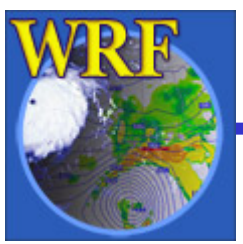
The WRF Preprocessing System: Overview and Installation

Michael Duda

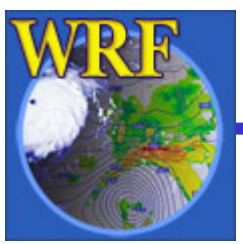


Purpose of this Lecture

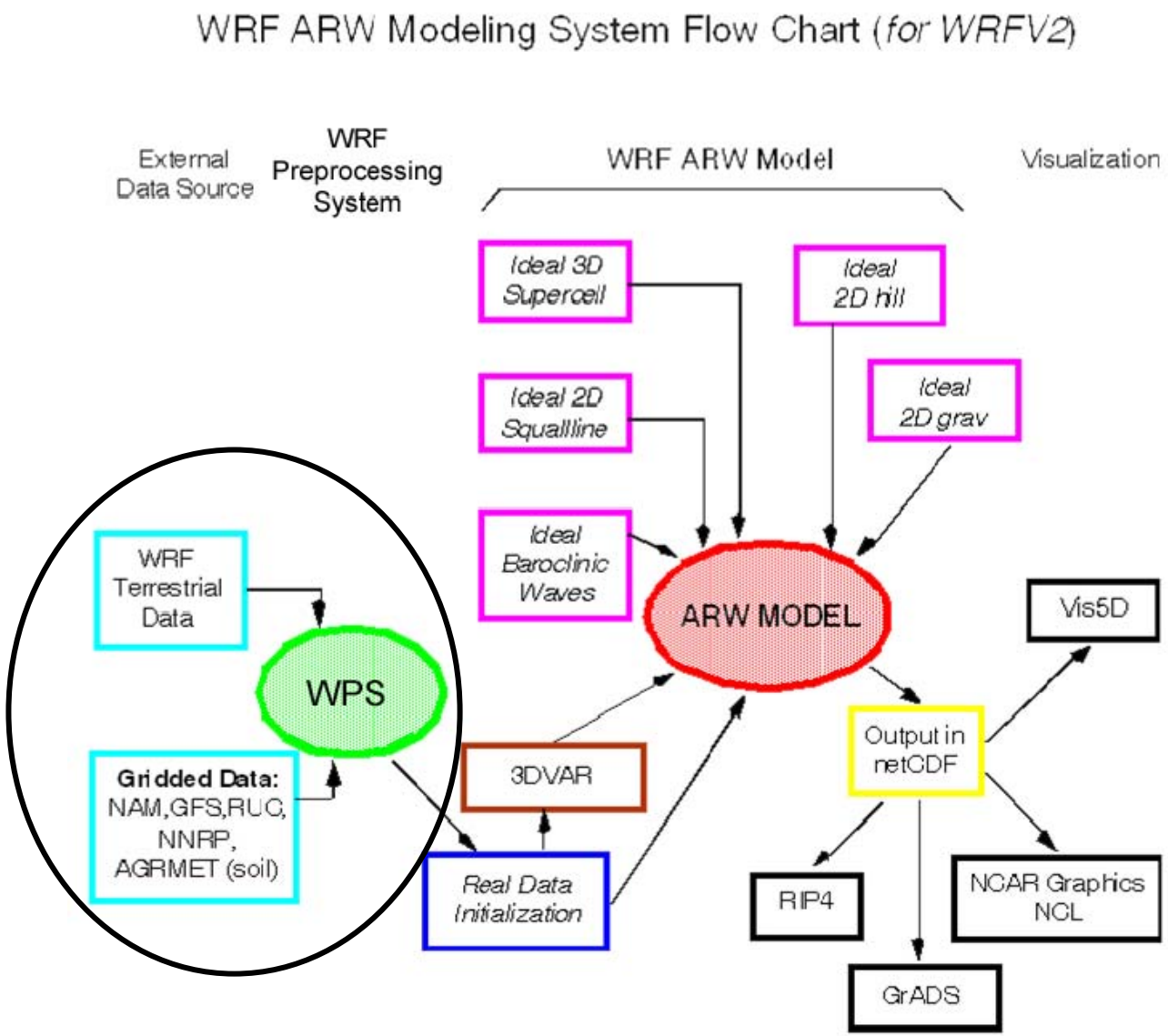
In this lecture, our goals are to:

1) Understand the purpose of the WPS
2) Learn what each component of the WPS does
3) Learn how install the WPS software

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



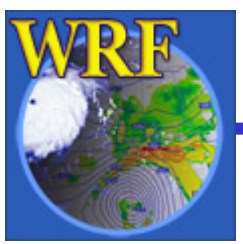
ARW Modeling System Flowchart



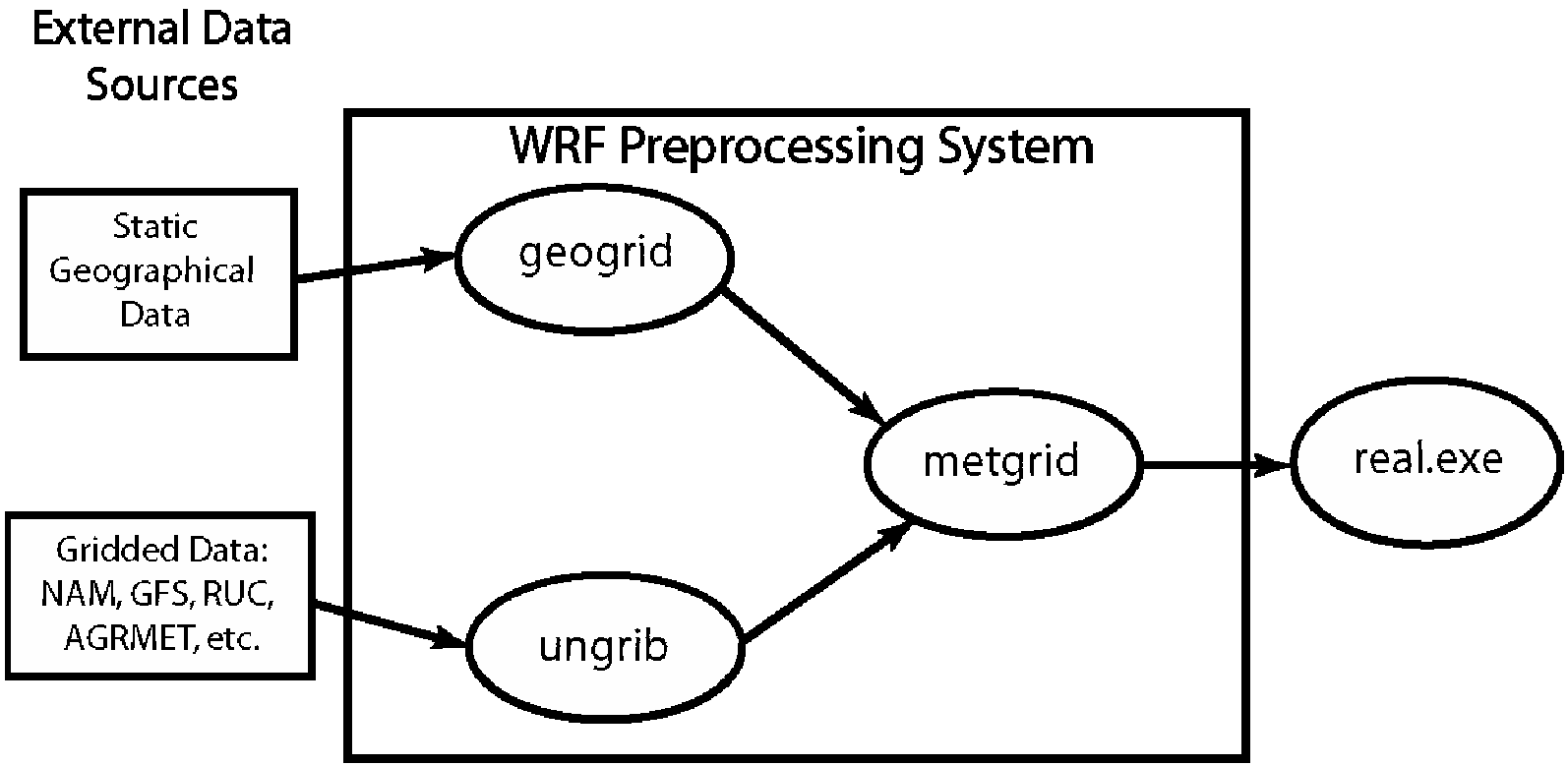
Purpose of the WPS

- Prepares input to ARW for real-data simulations:

1. Defines simulation domain and nested domains
2. Computes latitude, longitude, map scale factors, Coriolis parameters for 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

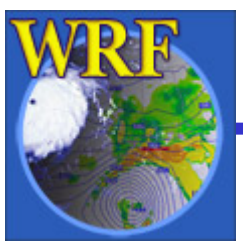


WPS Program Flowchart

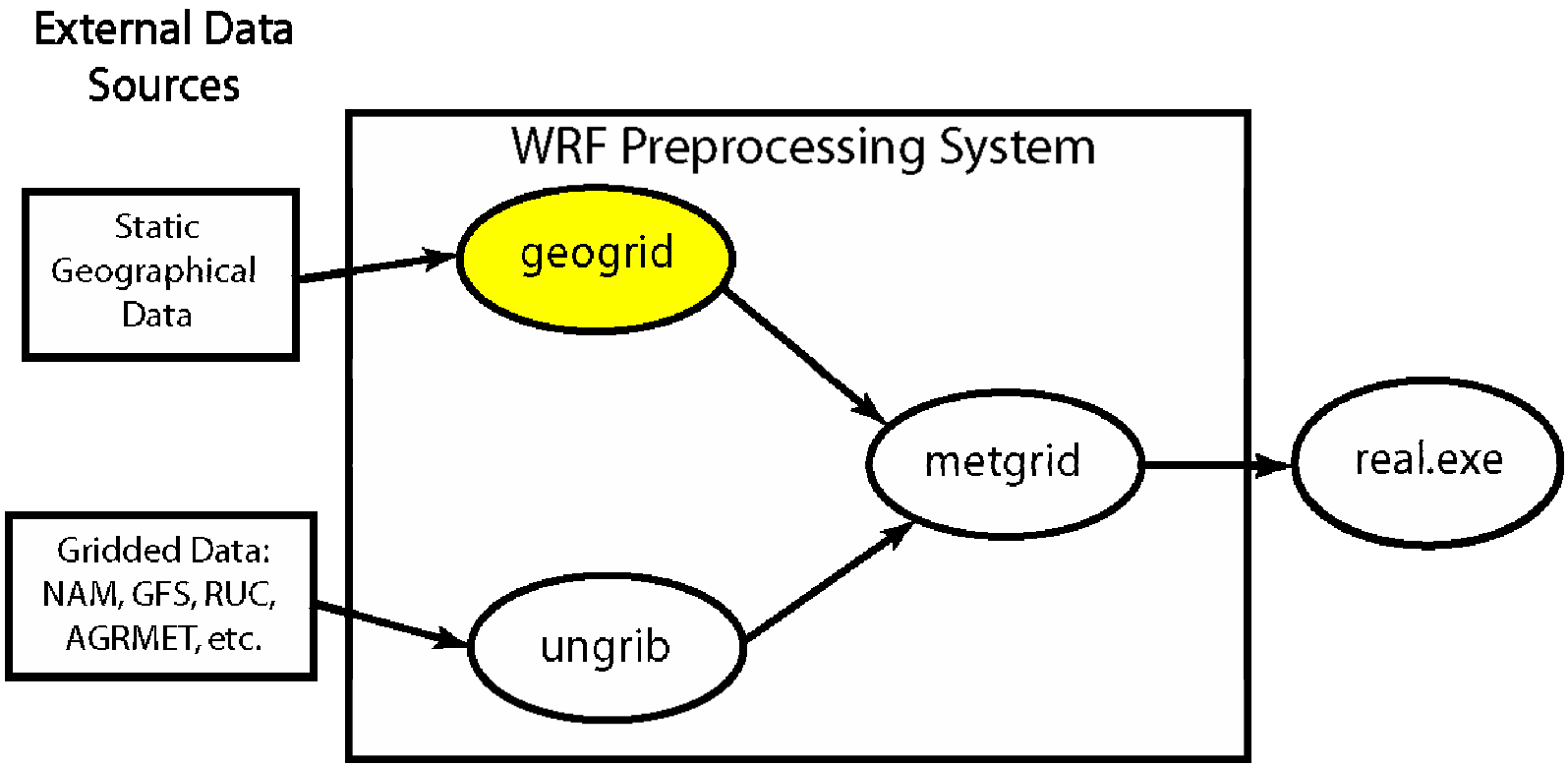


Function of WPS Components

- 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

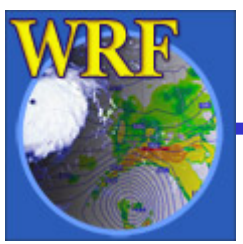


The *geogrid* program



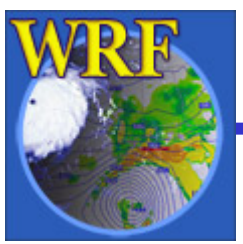
The *geogrid* program

- 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.

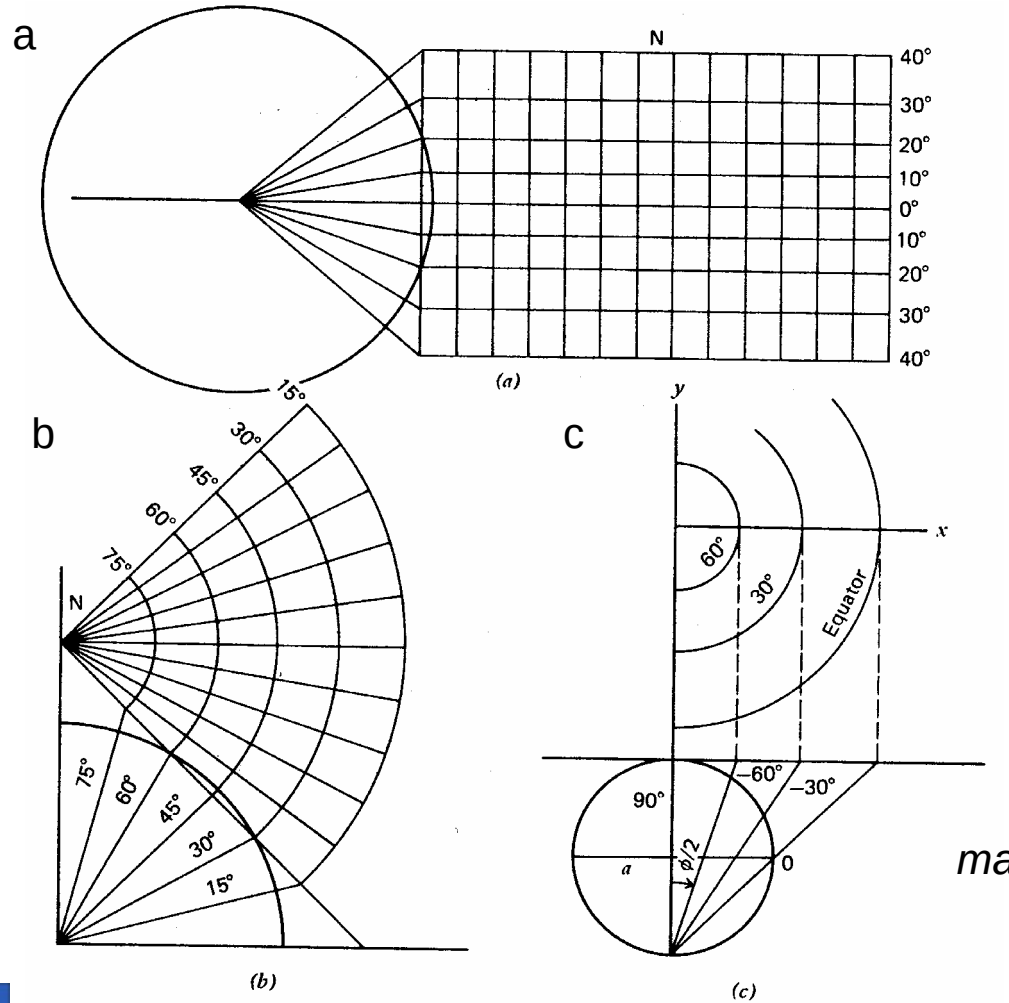


Geogrid: Defining model domains

- 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
- Map projections supported by ARW:
  1. Lambert conformal
  2. Mercator
  3. Polar stereographic



Aside: Map Projection Review



a. Mercator true at 20 N.

b. Lambert-Conformal true at 30 and 60 N.

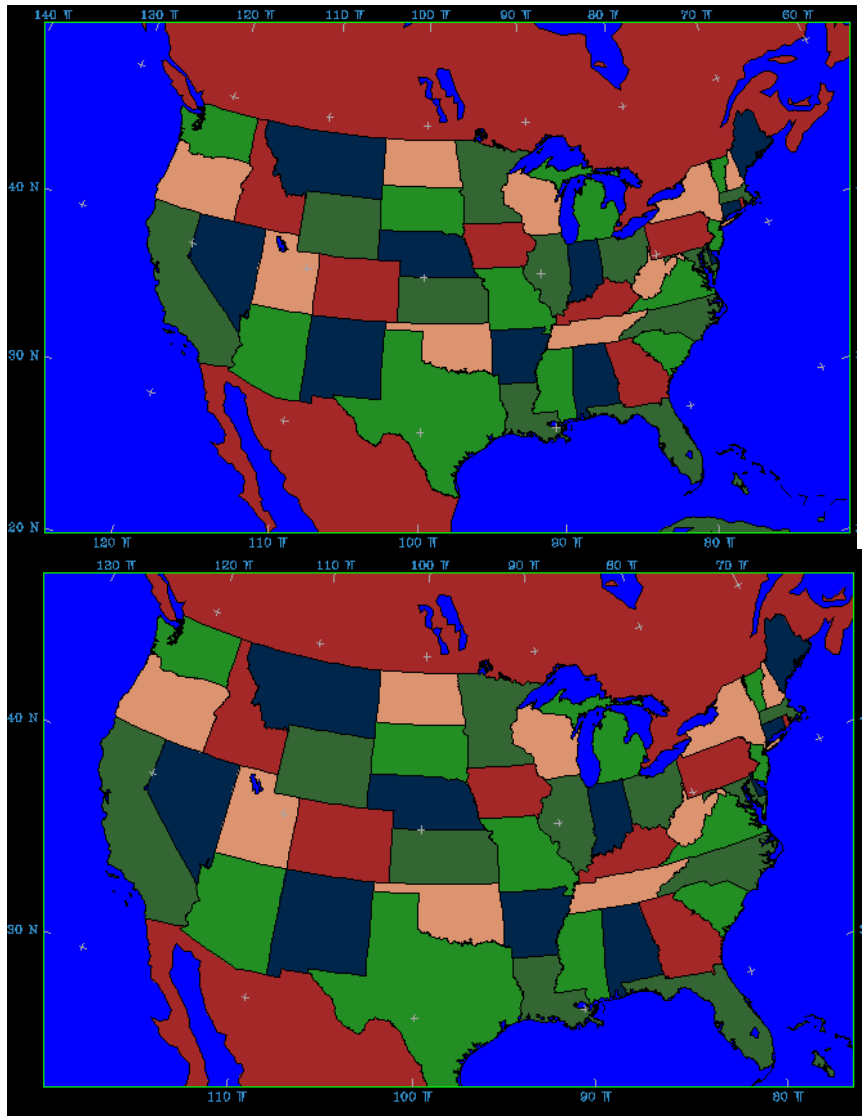
c. Polar stereographic true at 90 N

$$\textit{map scale factor} = \frac{\textit{distance on grid}}{\textit{distance on earth}}$$

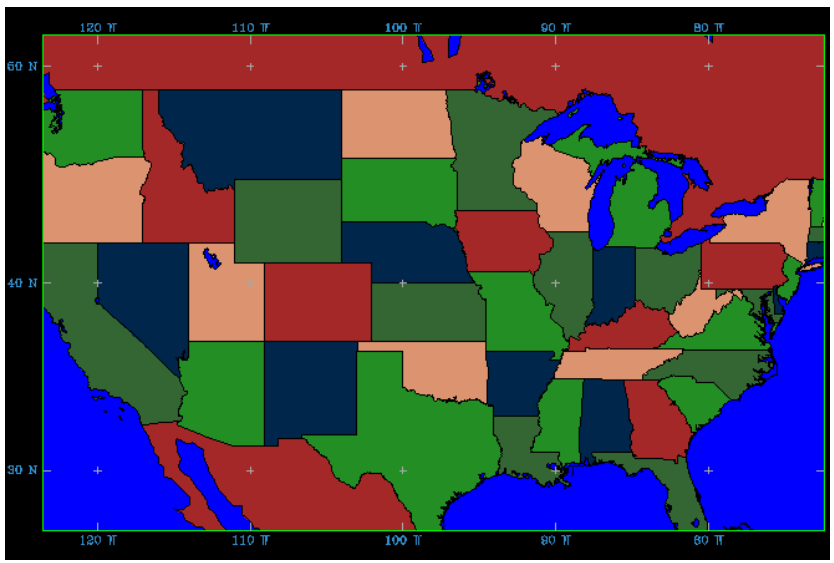
From *Numerical Prediction and Dynamic Meteorology* by Haltiner and Williams

Aside: Map Projection Examples

Lambert-Conformal

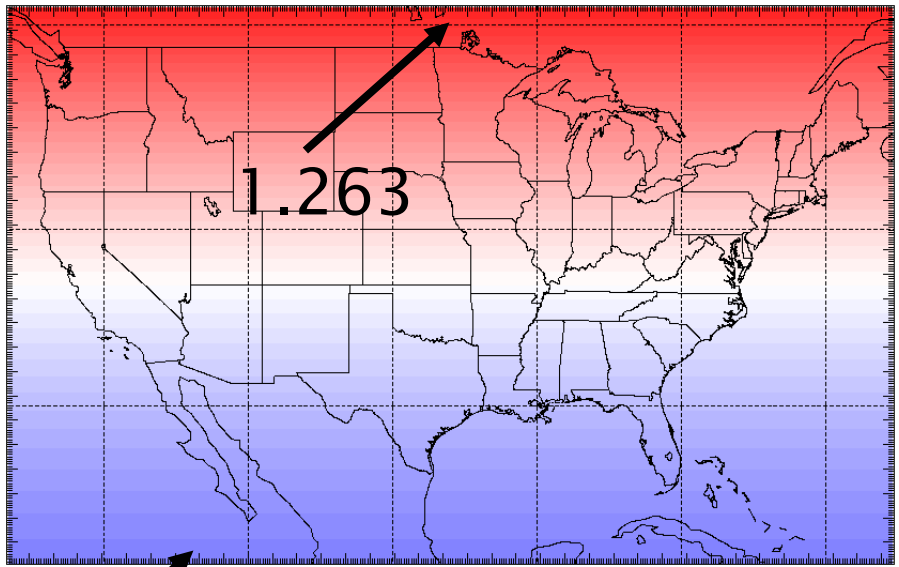


Mercator

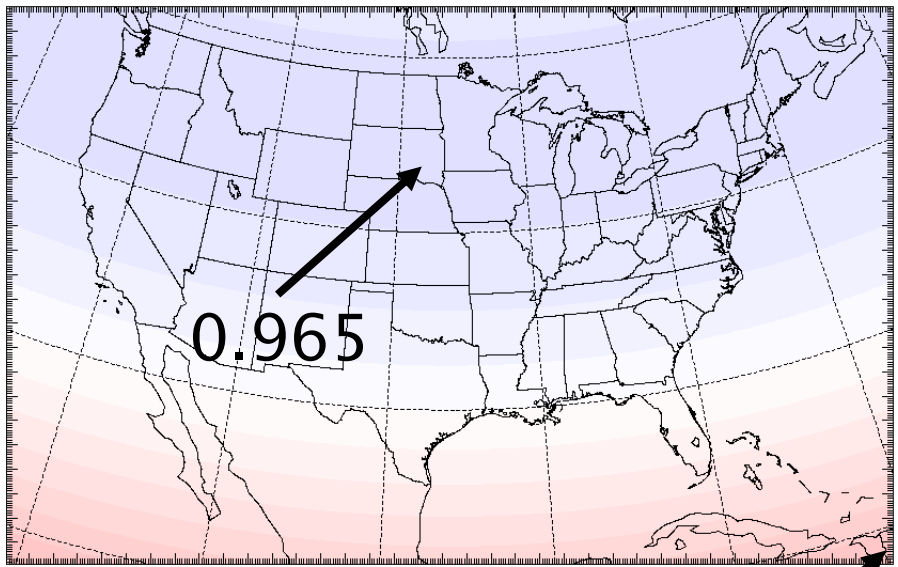


Polar stereographic

Aside: Why does projection matter?



Mercator projection

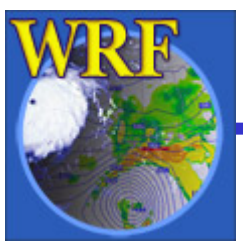


Lambert conformal projection

Above are the map scale factor fields for two 365x230 grid point domains; color scales are identical.

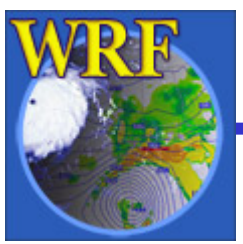
Geogrid: Defining Model Domains

- Define projection of domains with (at most) the following parameters
  - **MAP_PROJ**: 'Lambert', 'Mercator', or 'Polar'
  - **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*

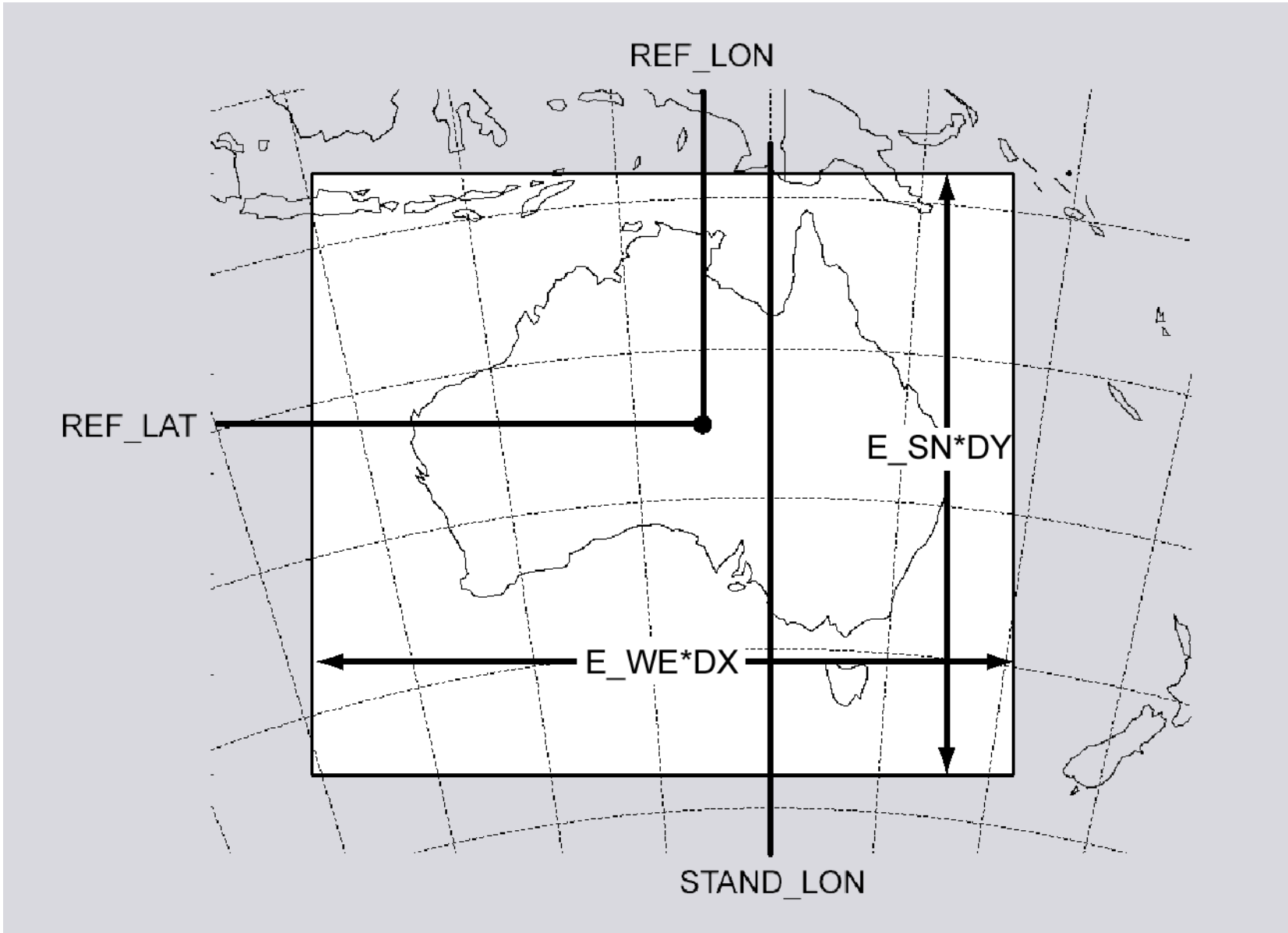


Geogrid: Defining Model Domains

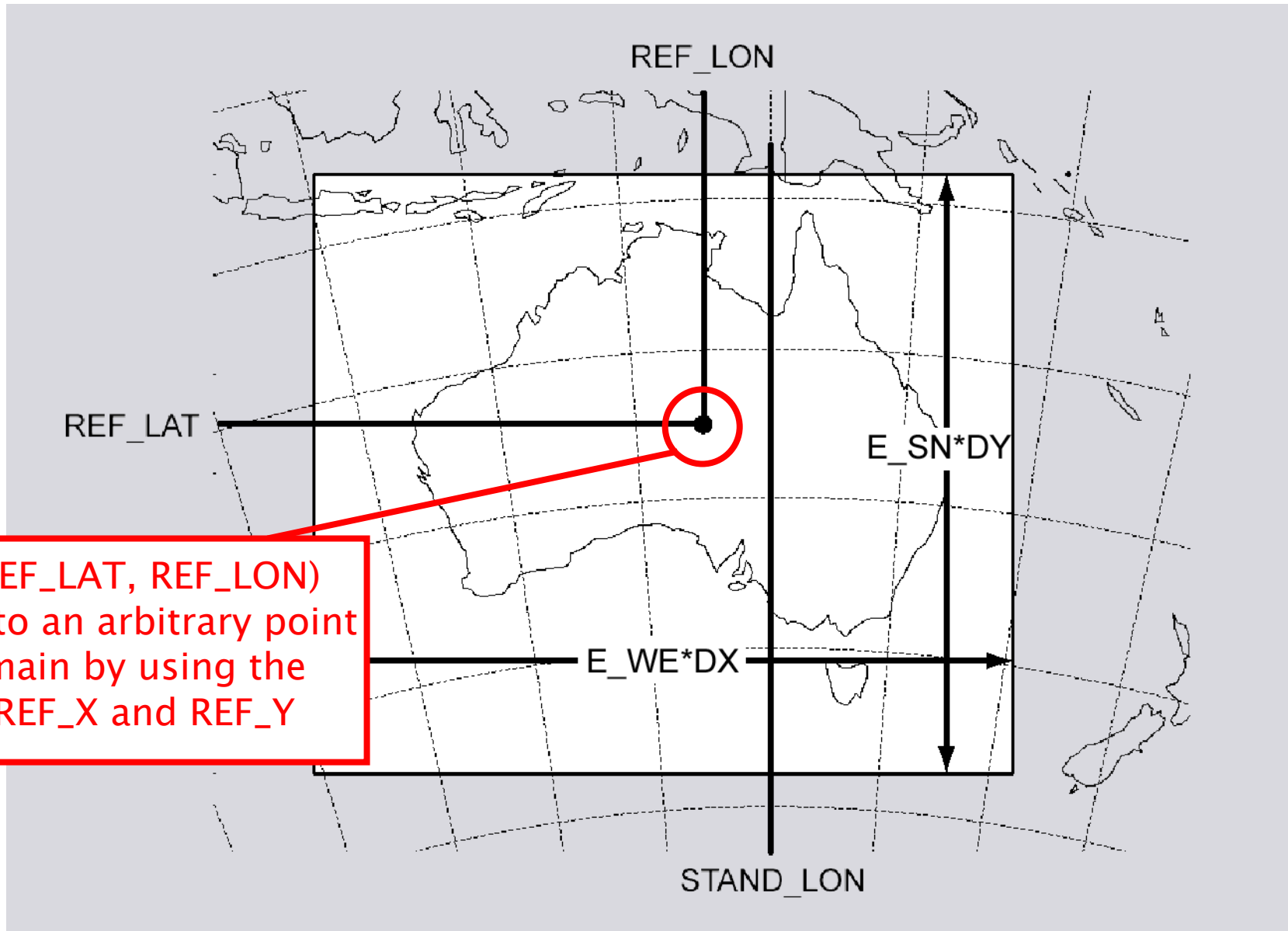
- Define the area covered (dimensions and location) by coarse domain using the following:
  - **REF_LAT, REF_LON**: The (lat,lon) of a known location in the domain (*by default, the center point of the domain*)
  - **DX, DY**: Grid distance where map factor = 1
  - **E_WE**: Number of grid points in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



Geogrid: Defining Model Domains



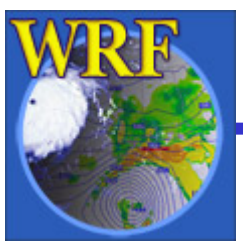
Geogrid: Defining Model Domains



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

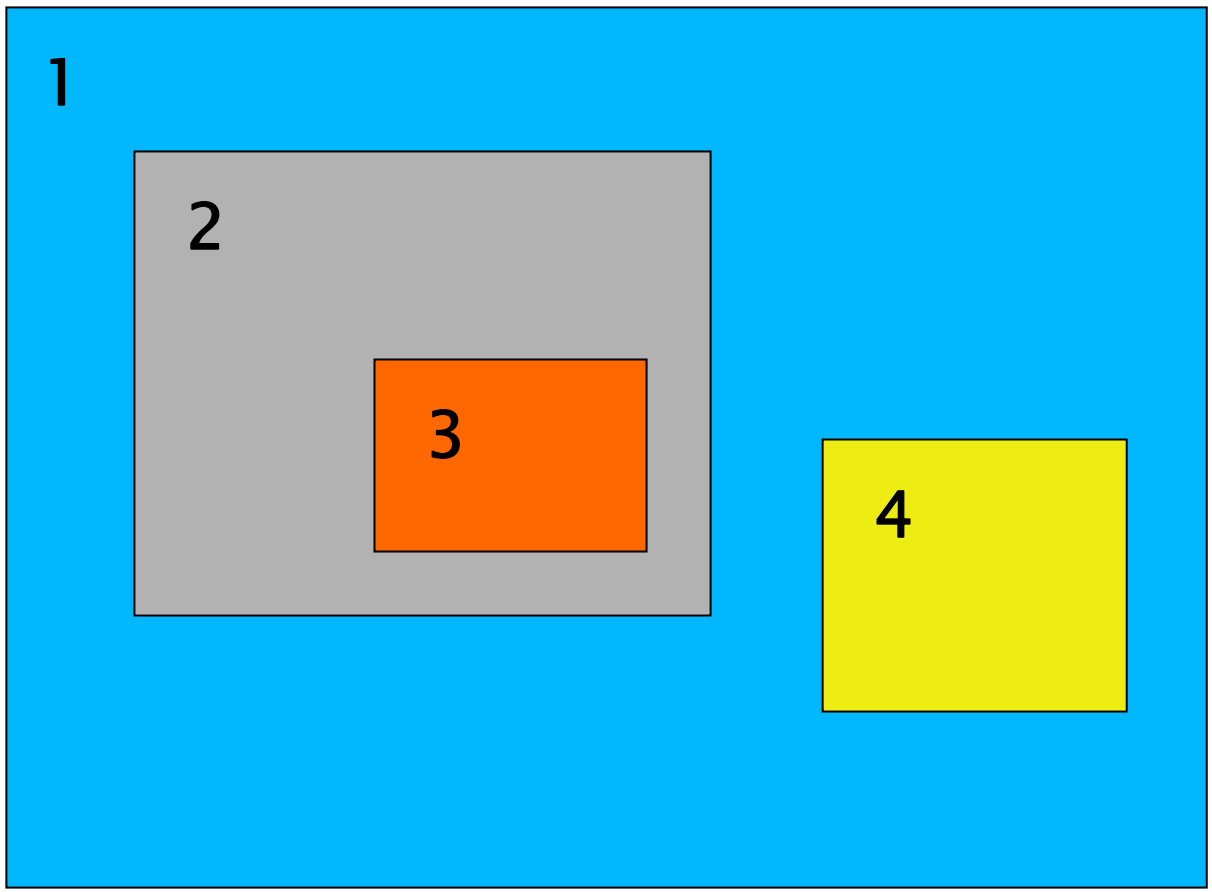
Geogrid: Nesting Basics

- A *nested domain* is a domain that is wholly contained within its *parent domain*, and may feed information back to its parent
  - A nested domain has exactly one parent
  - A domain (coarse or nested) may have one or more children
- In ARW, nests must not overlap in coverage!

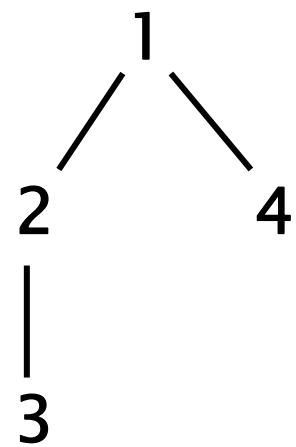


Geogrid: Nesting Example

Example configuration – 4 domains



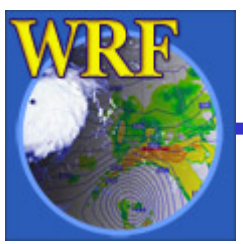
Each domain is assigned a *domain ID #*



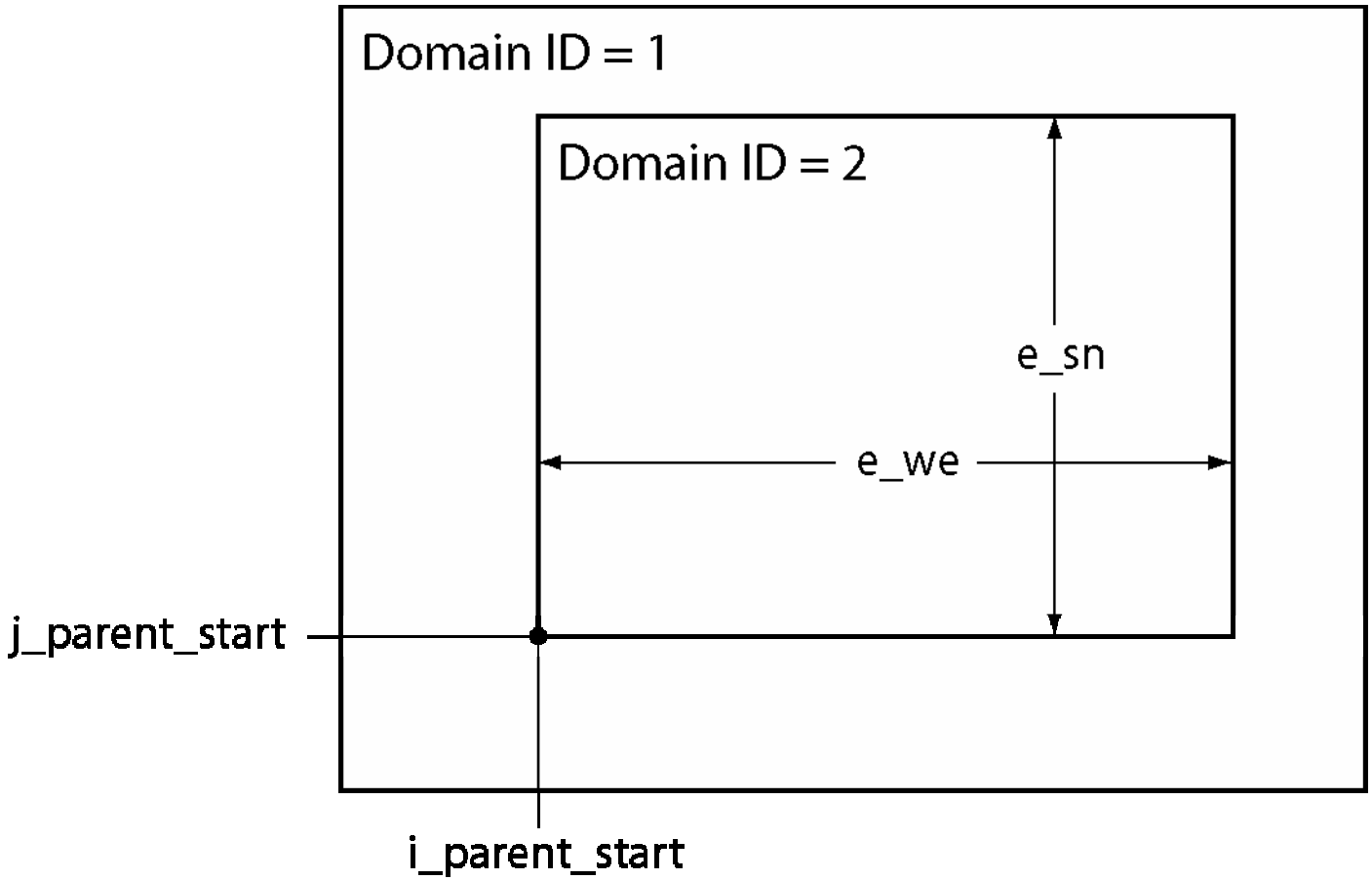
Nesting structure shown as a tree for the domains at left

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 grid points in west-east direction
  - **E_SN**: Number of grid points in south-north direction



Geogrid: Defining Nested Domains



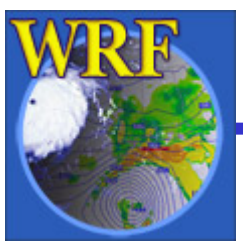
The grid spacing (*dx*) of domain 2 is determined by grid spacing of domain 1 and the *parent_grid_ratio*

e_we and e_*sn* must equal (*k**PARENT_GRID_RATIO)+1 for some positive integer *k*

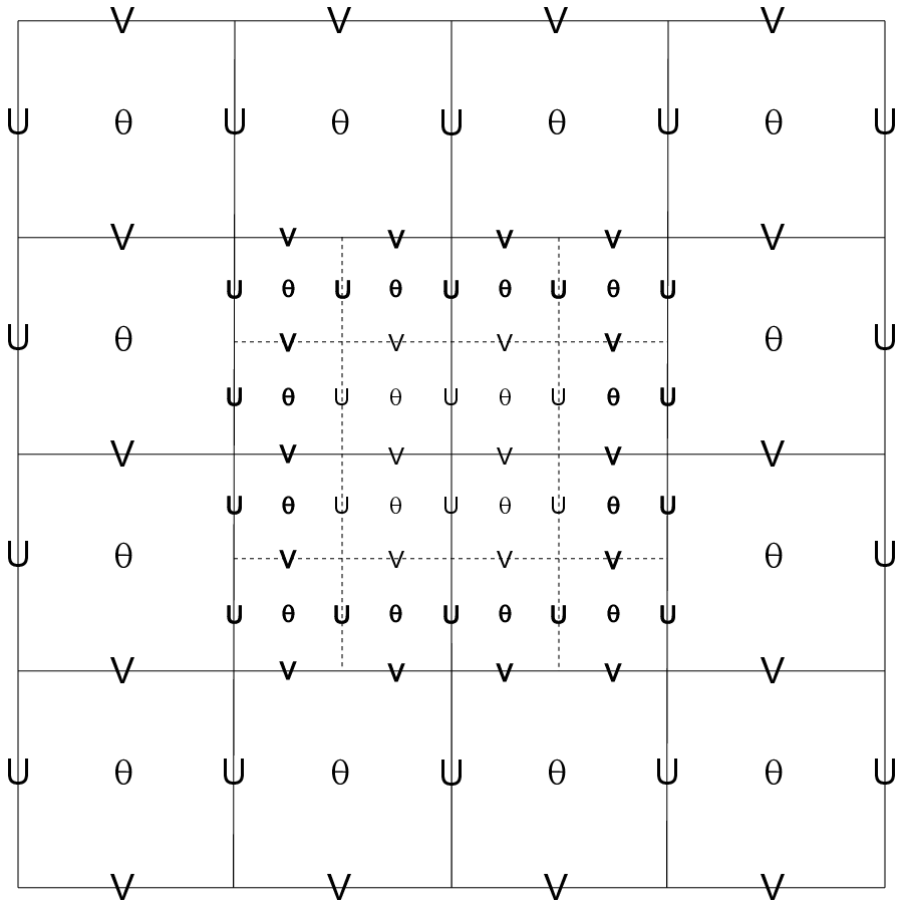
Aside: Nest Dimensions

Why must *e_we* and e_*sn* equal (*k**PARENT_GRID_RATIO)+1 for some positive integer *k?*

- In ARW, the corner points of nests must be coincident with the corner points of parent-domain grid cells



Aside: Nest Dimensions



PARENT_GRID_RATIO = 2

E_WE = 5 = 2*k+1

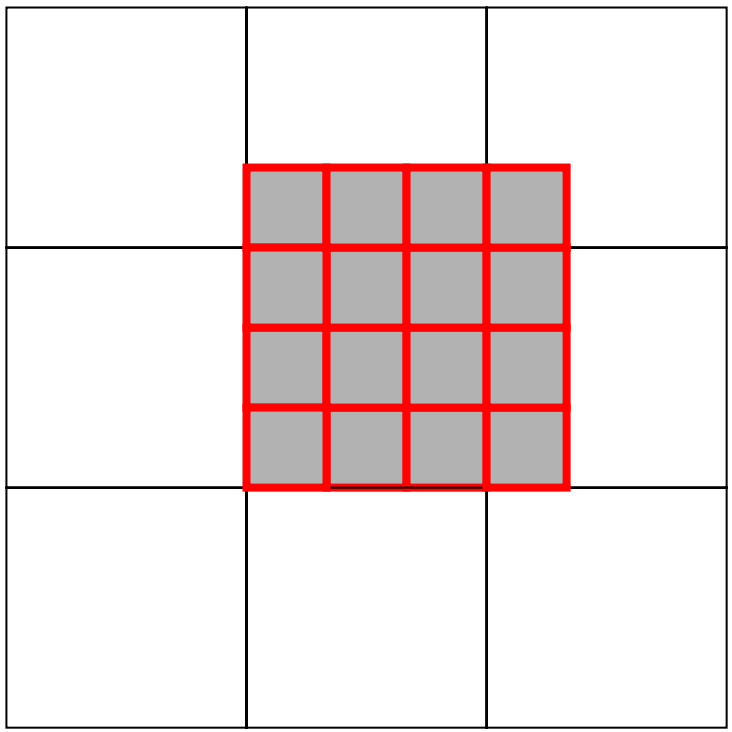
E_SN = 5 = 2*k+1

for k=2

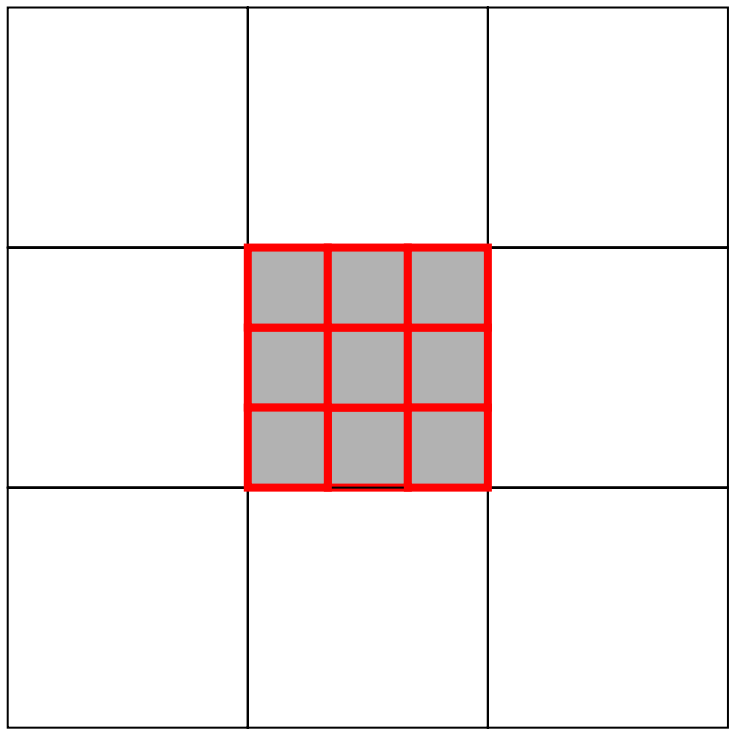
Geogrid will tell you if either E_WE or E_SN is bad.

Aside: Nest Dimensions

With a "bad" value for E_WE or E_SN, some coarse-domain grid cells will only be partially covered by the nested grid – bad for feedback



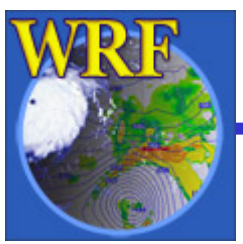
BAD



OK

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

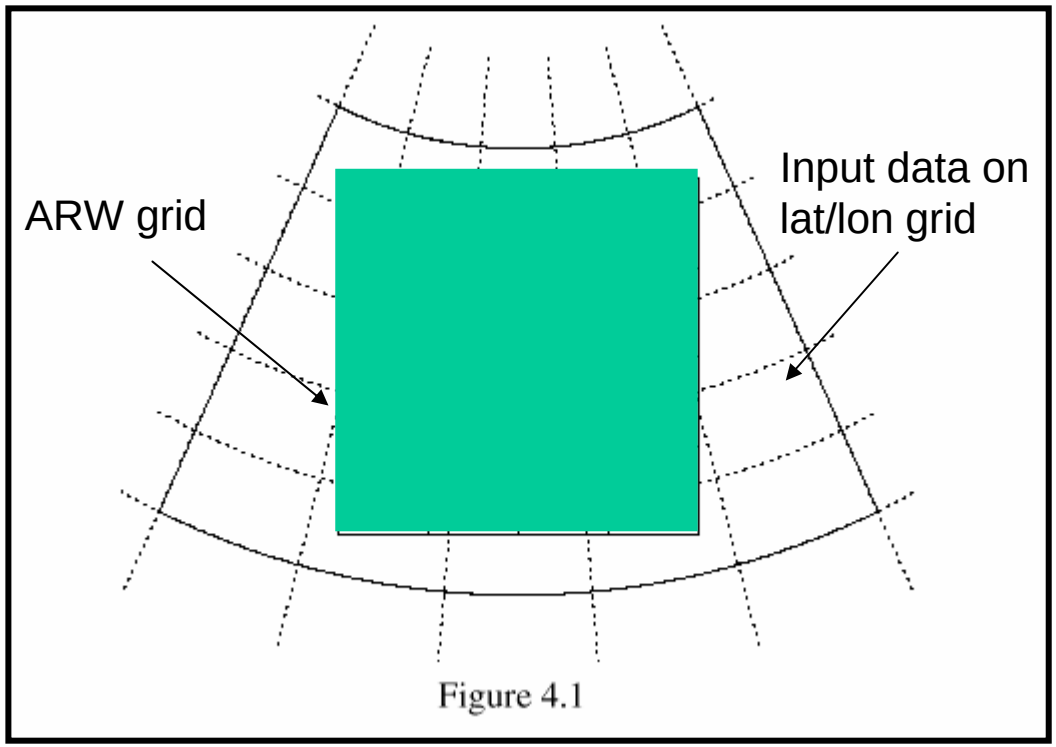
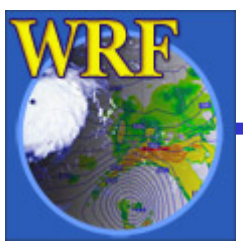


Figure 4.1

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

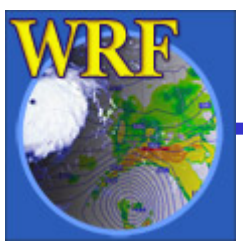
Geogrid: Interpolation Options

- Nearest neighbor
- 4-point bilinear
- 16-point overlapping parabolic
- 4-point average
- 16-point average
- Grid cell average



Geogrid: Program Flexibility

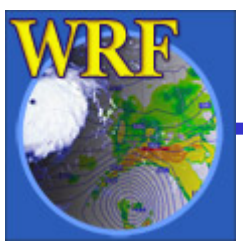
- *geogrid* is flexible enough to ingest and interpolate new static fields
  - handles either continuous or categorical fields
- New data sets must be written to simple binary format
- User needs to add an entry to the file `GEOGRID.TBL`



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)

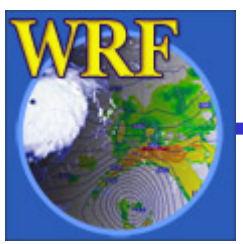
- Acceptable defaults exist in GEOGRID.TBL, so user will not generally need to edit the file (*but more on this in later lecture*!)



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/
==============================
```



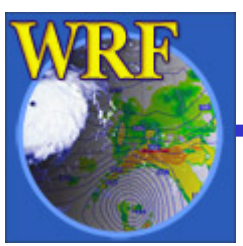
Geogrid: Program Flexibility

- 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
```



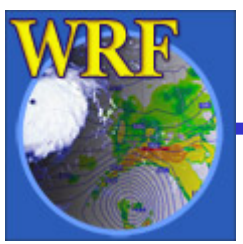
Geogrid: Program Flexibility

- 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 ARW User's Guide



Geogrid: Program Output

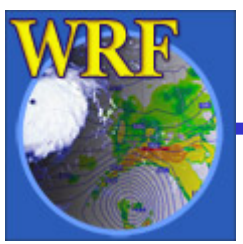
- The parameters defining each domain, plus interpolated static fields, are written using the WRF I/O API
  - One file per domain
- Filenames: geo_em.d0*n*.nc

  (where *n* is the domain ID #)
- Example:

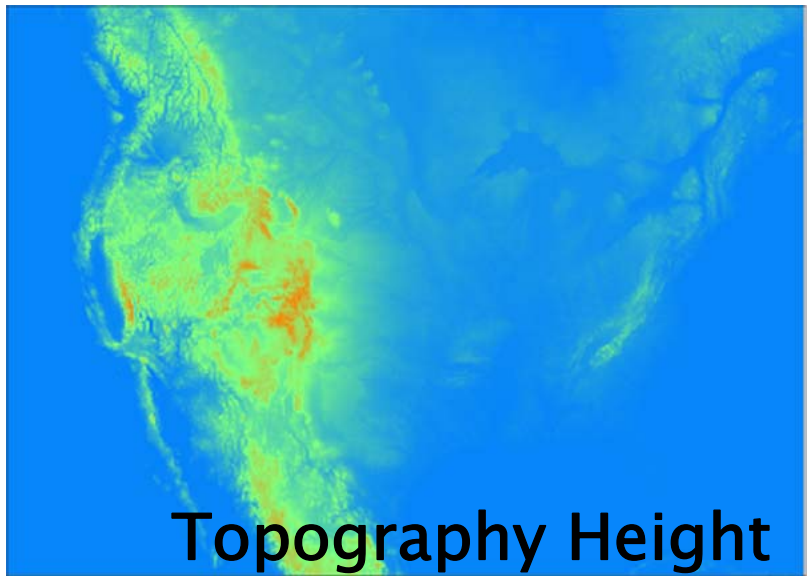
  geo_em.d01.nc

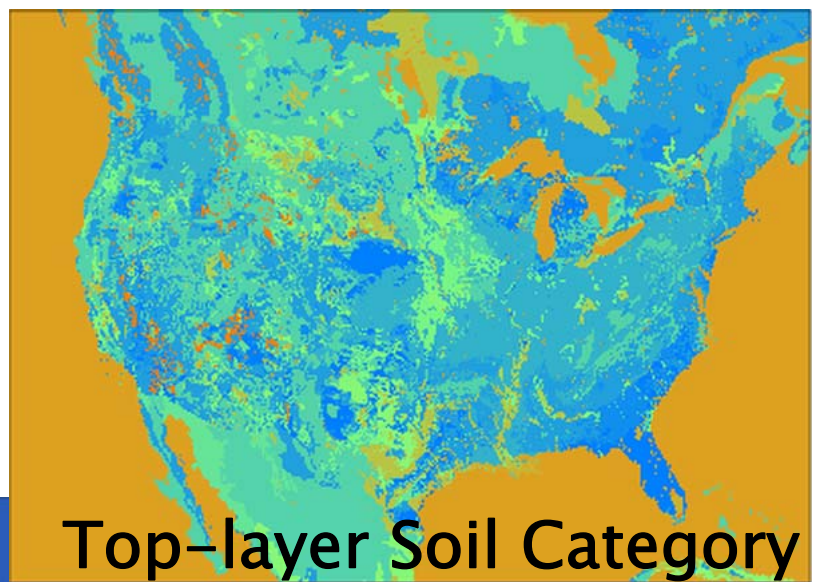
  geo_em.d02.nc (nest)

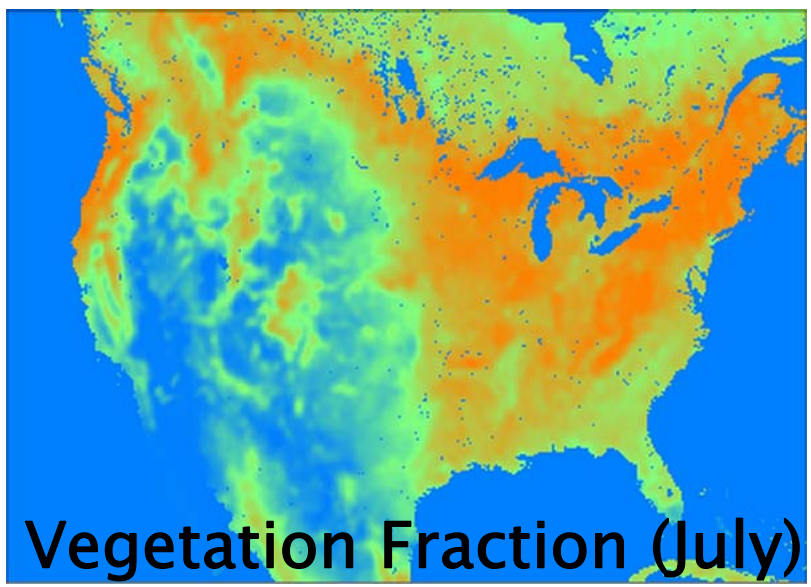
  geo_em.d03.nc (nest)



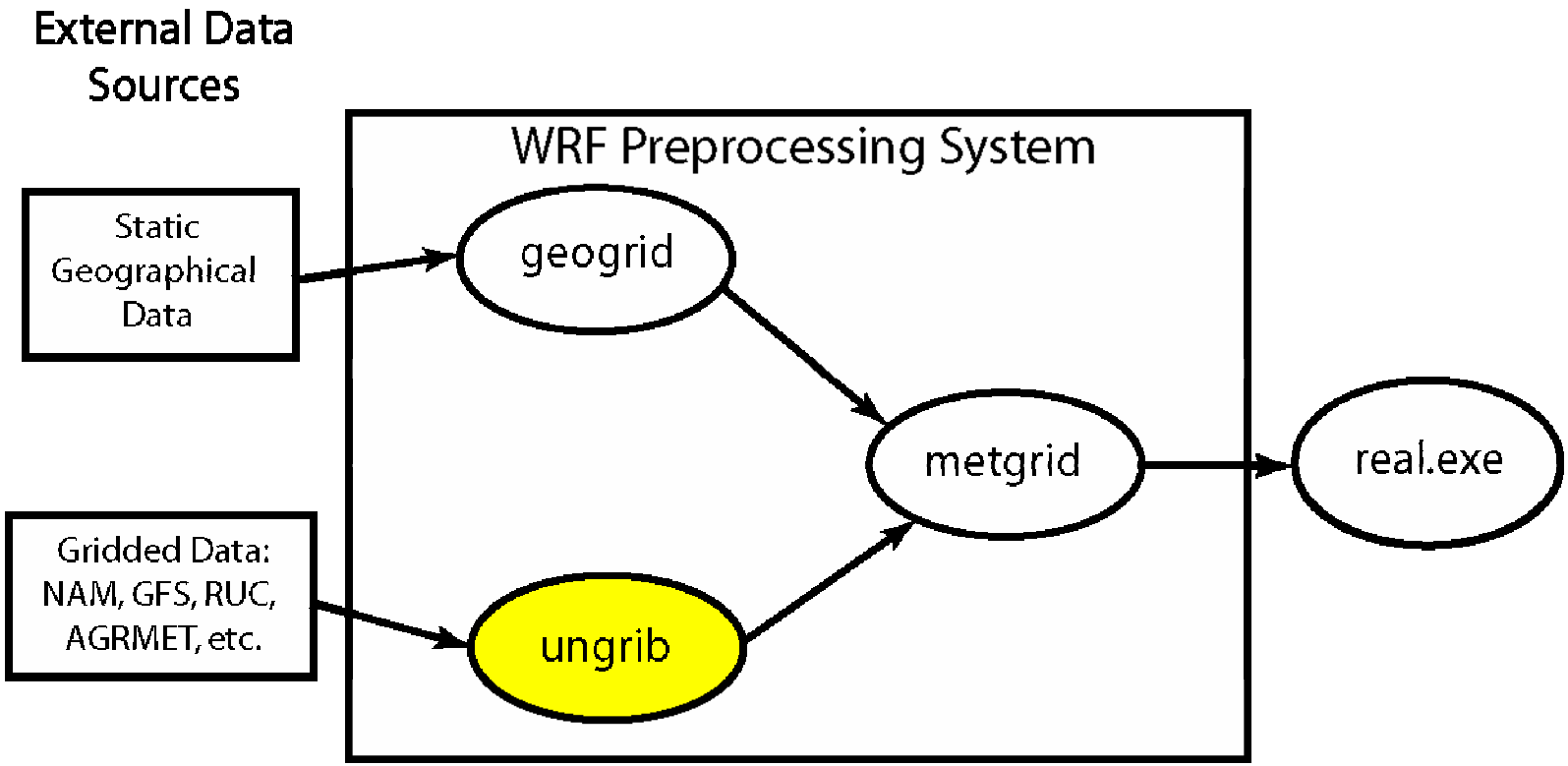
Geogrid: Example Output





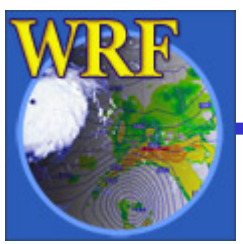


The *ungrib* program



The *ungrib* program

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

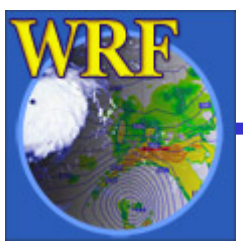


Ungrib: Vtables

How does ungrib know which fields to extract?

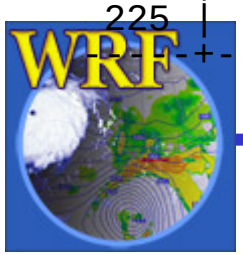
Using Vtables

- 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



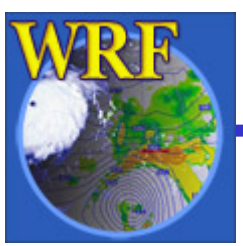
Ungrib: Example Vtable

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

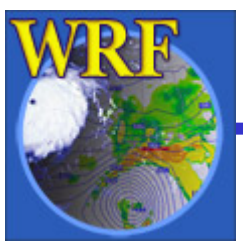


Ungrib: Vtables

What if a data source has no existing Vtable?

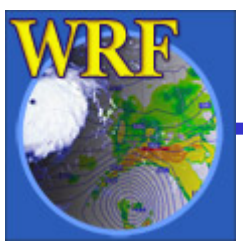
Create a Vtable

- 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, can 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



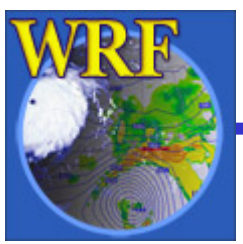
Ungrib: Program Output

- 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

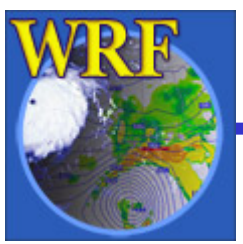
  FILE:2007-07-25_00



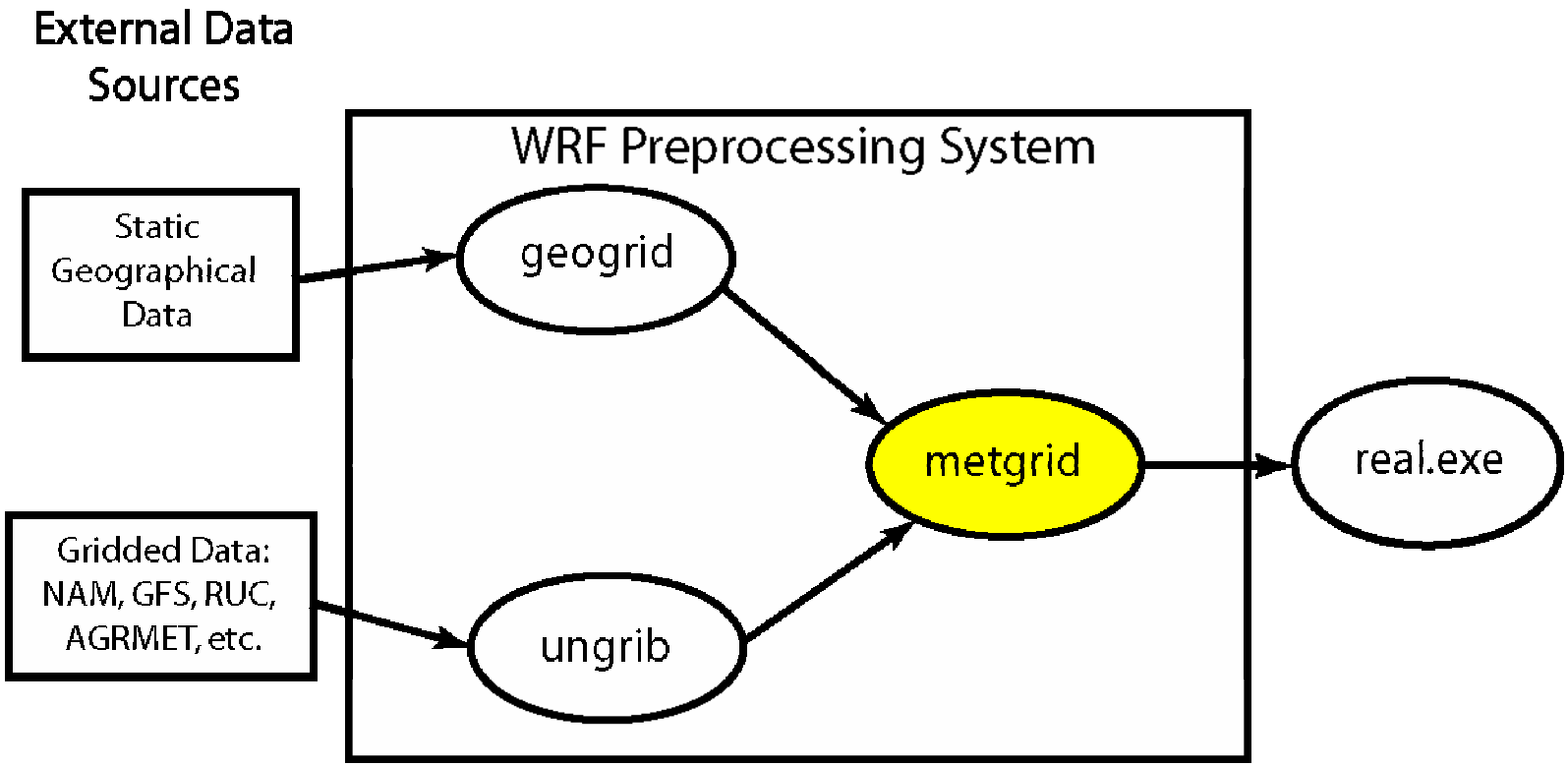
Ungrib: Obtaining GRIB Data

- 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

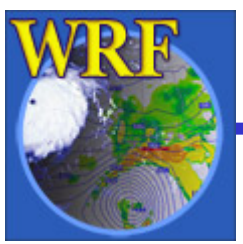


The *metgrid* program



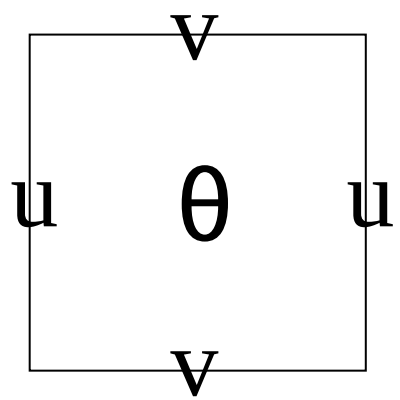
The *metgrid* program

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



Metgrid: Grid Staggering

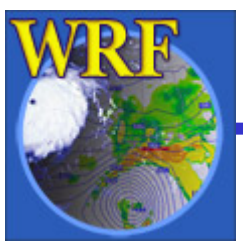
- For ARW, wind U-component interpolated to "u" staggering
- Wind V-component interpolated to "v" staggering
- Other meteorological fields interpolated to "θ" staggering by default (*can change this!*)



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

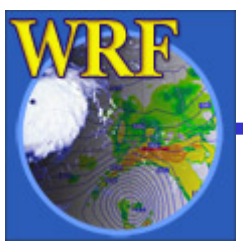
Metgrid: Interpolation Methods

- Nearest neighbor
- 4-point bilinear
- 16-point overlapping parabolic
- 4-point average
- 16-point average
- Grid-cell-average

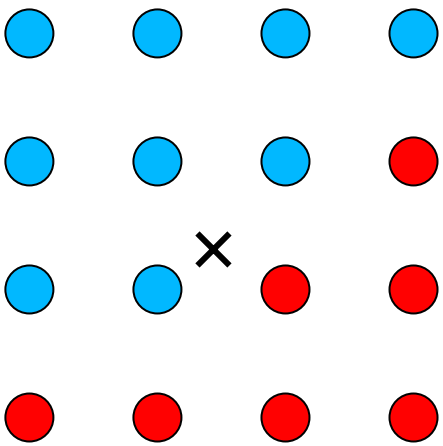


Metgrid: Masked Interpolation

- *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 (ex: LANDSEA)
  - Can rely on special values (ex: 1×10^{30}) in field itself to identify masked grid points



Metgrid: Masked Interpolation



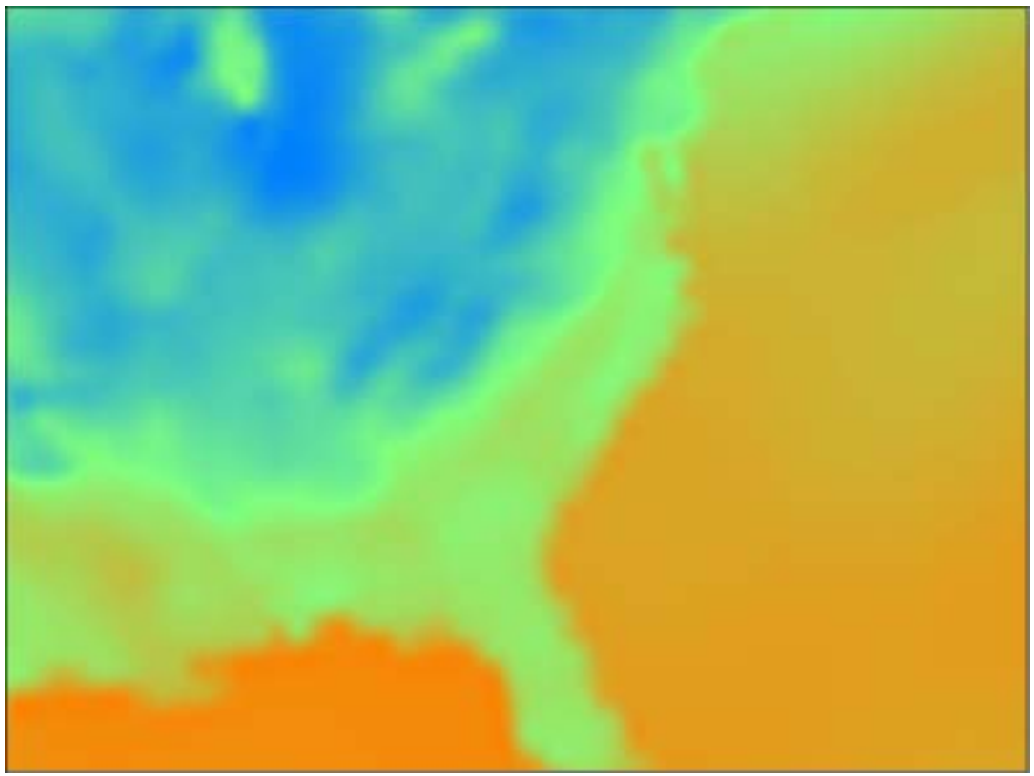
= valid source data

= masked/invalid data

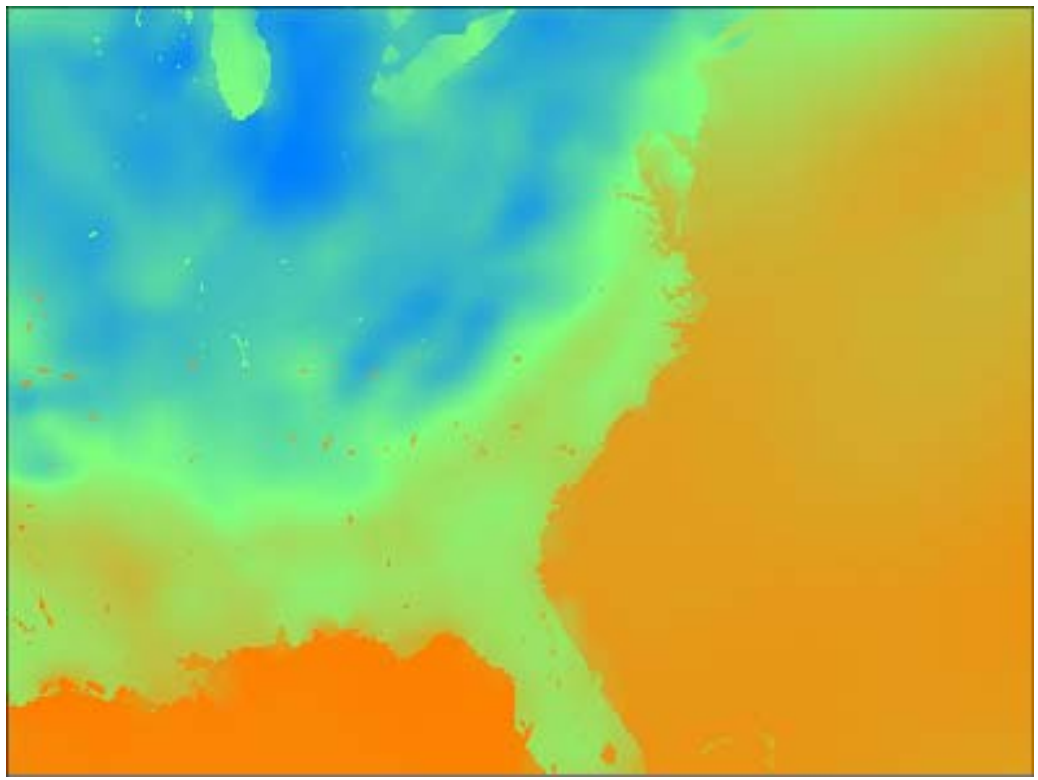
Suppose we need to interpolate to point X

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

Metgrid: Masked Interpolation



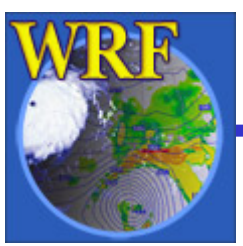
Skin temperature field interpolated from GFS 0.5-deg field with no mask using a sixteen-point interpolator.



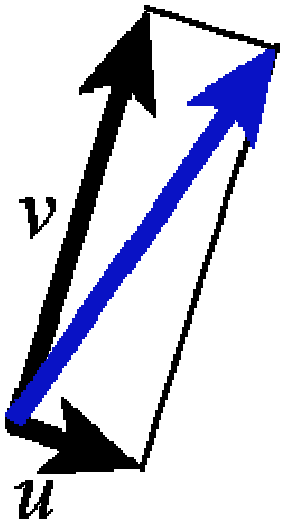
Skin temperature field interpolated using masks: GFS water points interpolated to model water points, GFS land points interpolated to model land points.

Metgrid: Wind Rotation

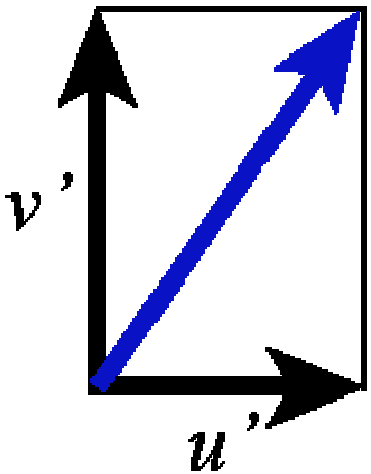
- Input wind fields (U-component + V-component) are either:
  - **Earth-relative**: U-component = westerly component; V-component = southerly component
  - **Relative to source grid**: U-component (V-component) parallel to source model x-axis (y-axis)
- WRF expects wind components to be relative to the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



Metgrid: Wind Rotation Example



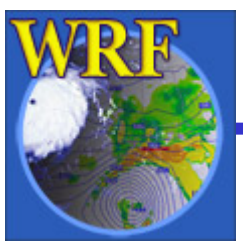
A wind vector, shown in terms of its U and V components with respect to the source grid.



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

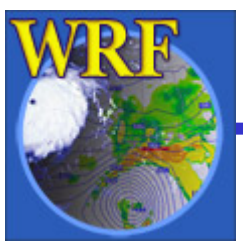
Metgrid: Constant Fields

- For short simulations, some fields may be constant
  - Ex: SST, sea-ice fraction
- Constant fields typically only available at one time
- Use namelist option `CONSTANTS_NAME` option to specify such fields:
  - `CONSTANTS_NAME = 'SST_FILE:2007-07-24_00'`



Metgrid: Program Flexibility

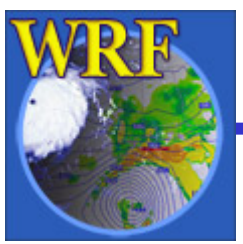
- *metgrid* is capable of interpolating both isobaric and native vertical coordinate data sets
- User may specify interpolation methods and related options in the `METGRID.TBL` file
- `METGRID.TBL` file similar in format to the file `GEOGRID.TBL`



Metgrid: Program Flexibility

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

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

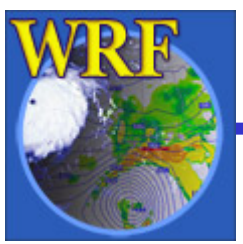


Metgrid: Program Output

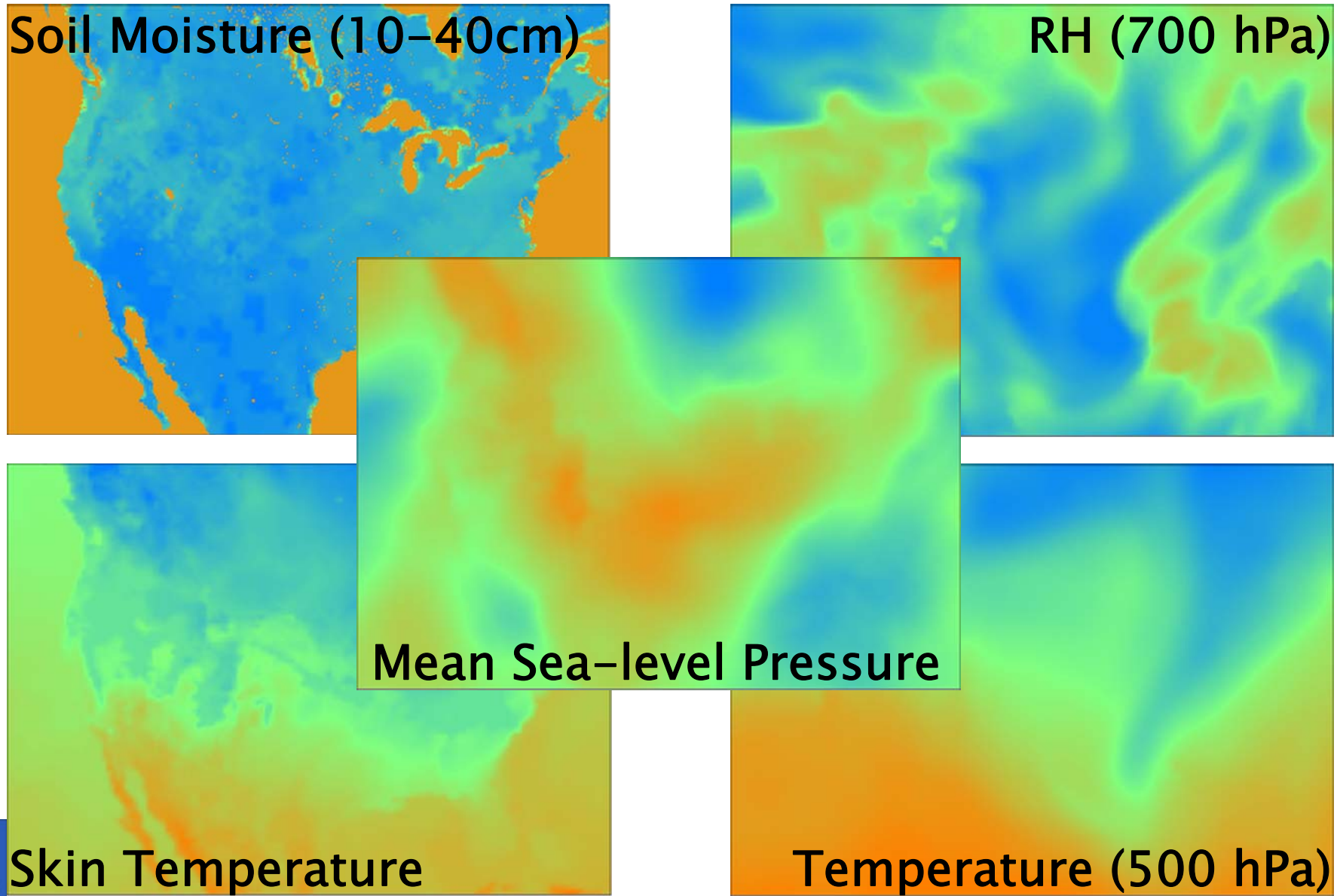
- One output file per domain per time
  - Except nests, which only have initial time!
- Files contain static fields from geogrid plus interpolated meteorological fields
- Filenames:

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

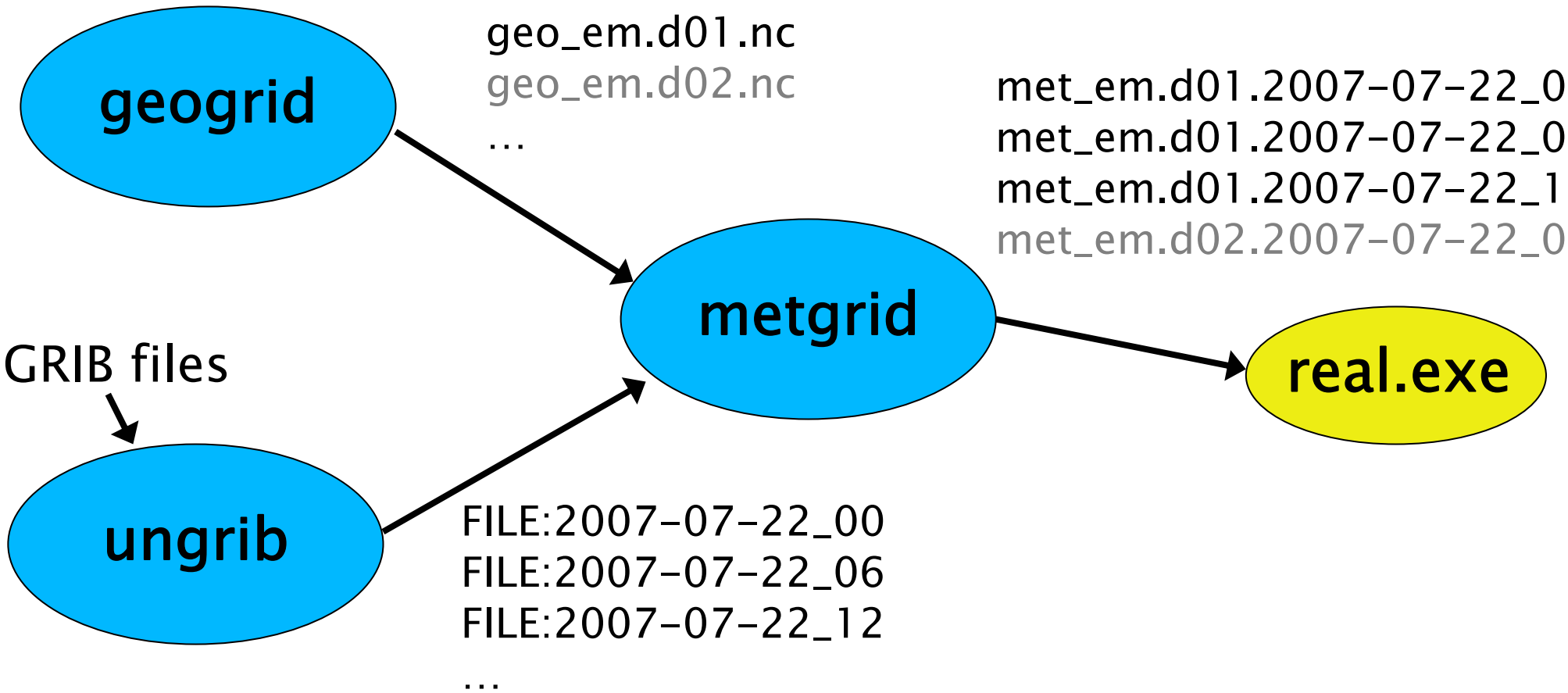
  (where *n* is the domain ID #)



Metgrid: Example Output

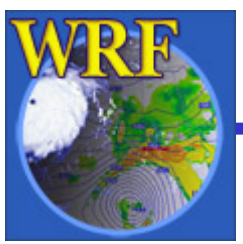


WPS Summary


geogrid
geo_em.d01.nc
geo_em.d02.nc
…
met_em.d01.2007-07-22_00
met_em.d01.2007-07-22_06
met_em.d01.2007-07-22_12
met_em.d02.2007-07-22_00
metgrid
real.exe
GRIB files
ungrib
FILE:2007-07-22_00
FILE:2007-07-22_06
FILE:2007-07-22_12
…

WPS Utility Programs

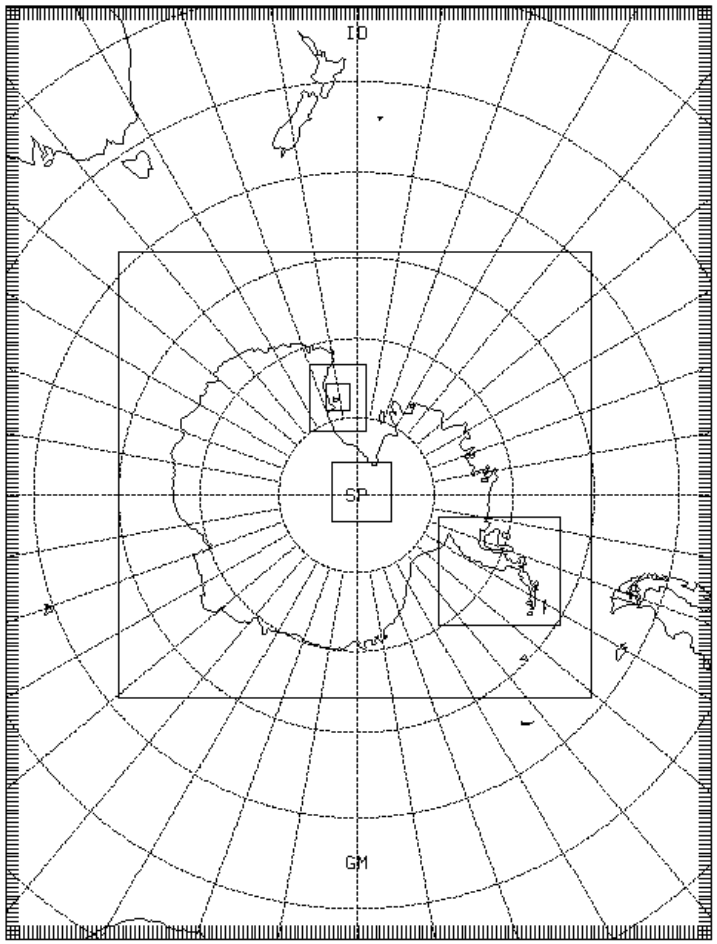
- Besides geogrid, ungrib, and metgrid, some simple utility programs are distributed with WPS:
  - For checking contents of intermediate format files
  - For listing contents of GRIB1 & GRIB2 files
  - To assist in locating domains
- Some programs use NCAR Graphics libraries for plotting -> NCAR Graphics must be installed



Utility: plotgrids.exe

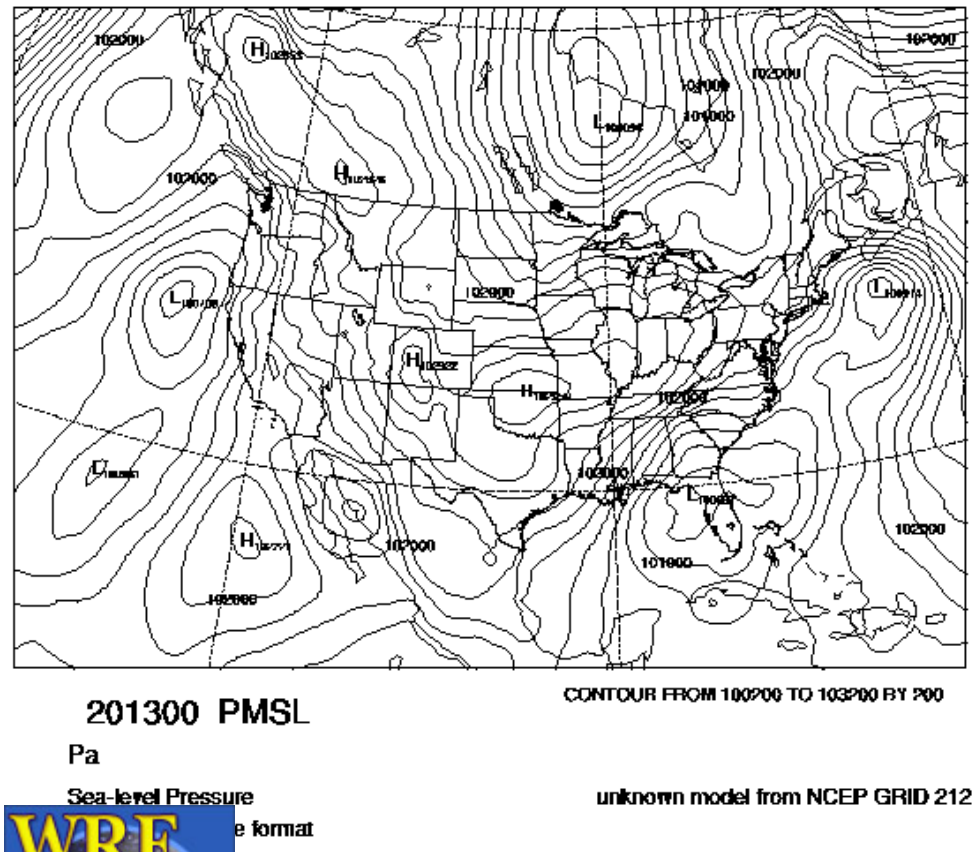
The plotgrids.exe program plots the location of grids defined in *namelist.wps*

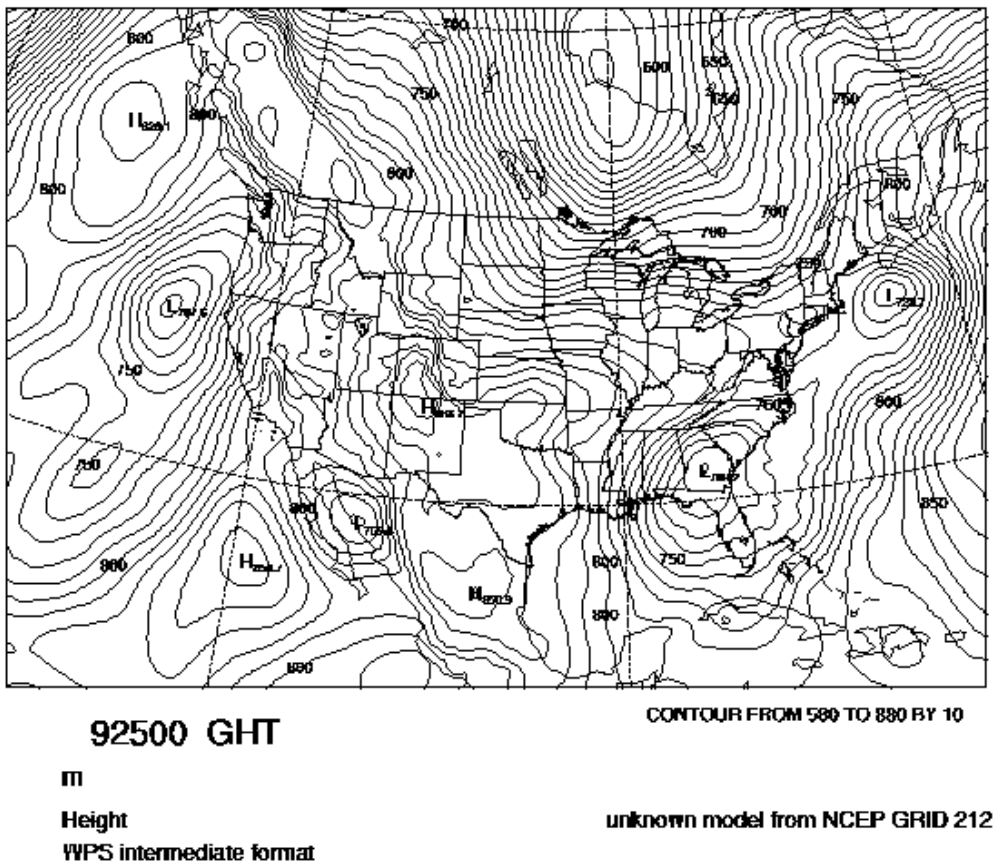
The plotgrids.exe program can be used to iteratively refine the locations of grids.



Utility: plotfmt.exe

The plotfmt.exe program plots the contents of ungrib intermediate-format files

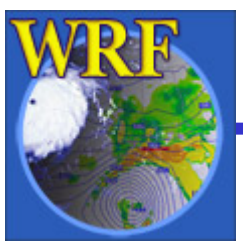




Utility: rd_intermediate.exe

The rd_intermediate.exe lists information about the fields found in an intermediate-format file

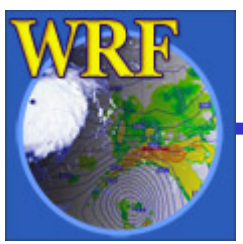
```
================================================
FIELD = TT
UNITS = K DESCRIPTION = Temperature
DATE = 2000-01-24_12:00:00 FCST = 0.000000
SOURCE = unknown model from NCEP GRID 212
LEVEL = 200100.000000
I,J DIMS = 185, 129
IPROJ = 1
  REF_X, REF_Y = 1.000000, 1.000000
  REF_LAT, REF_LON = 12.190000, -133.459000
  DX, DY = 40.635250, 40.635250
  TRUELAT1 = 25.000002
DATA(1,1)=295.910950
================================================
```



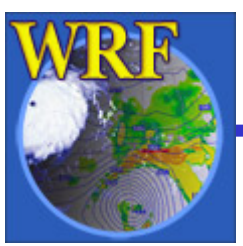
g1print.exe & g2print.exe

The g1print.exe & g2print.exe programs list the contents of a GRIB1 or GRIB2 file:

rec num	Prod Disc	Cat	Param num	Lvl code	Lvl one	Lvl two	Name	Time	Fcst hour
1	0	3	5	100	100000	0	HGT	2006-08-16_12:00:00	00
2	0	3	5	100	97500	0	HGT	2006-08-16_12:00:00	00
3	0	3	5	100	95000	0	HGT	2006-08-16_12:00:00	00
4	0	3	5	100	92500	0	HGT	2006-08-16_12:00:00	00
5	0	3	5	100	90000	0	HGT	2006-08-16_12:00:00	00
6	0	3	5	100	85000	0	HGT	2006-08-16_12:00:00	00
7	0	3	5	100	80000	0	HGT	2006-08-16_12:00:00	00
8	0	3	5	100	75000	0	HGT	2006-08-16_12:00:00	00
9	0	3	5	100	70000	0	HGT	2006-08-16_12:00:00	00
10	0	3	5	100	65000	0	HGT	2006-08-16_12:00:00	00



Installing the WPS



Installation: Basic Steps

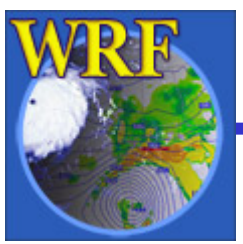
0) Check system requirements

1) Download source code

2) Download static terrestrial data

3) Install WRF

4) Install WPS



Basic Steps

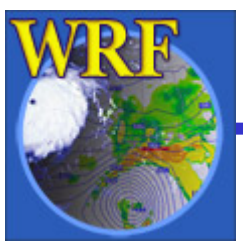
0) Check system requirements

1) Download source code

2) Download static terrestrial data

3) Install WRF

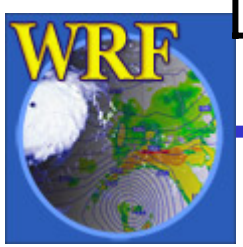
4) Install WPS



0) Check system requirements

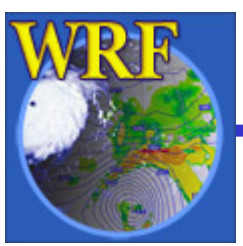
- Currently supported systems:

Architecture	OS	Compiler
IBM POWER-4/5	AIX	xlf90
Compaq Alpha	OSF1	f90
PC x86, x86_64	GNU/Linux 32-bit, 64-bit	PGI, Intel, g95, PathScale
SGI	IRIX	f90
SGI Altix	Linux	Intel
Sun	SunOS	f90
Apple G5	OSX	xlf90



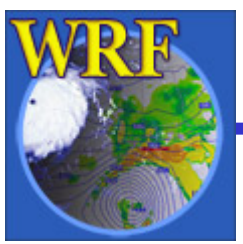
0) Check system requirements

- Required libraries
  - **NetCDF** (needed by WRF and WPS)
  - **NCAR Graphics** (*optional but recommended* – used by graphical utility programs)
- Optional libraries for GRIB2 support
  - **JasPer** (JPEG 2000 "lossy" compression library)
  - **PNG** ("lossless" compression library)
  - **zlib** (compression library used by PNG)



0) Check system requirements

- Installation of these libraries is *not* part of the WPS installation script
  - We recommend having a system administrator install the required libraries before installing WRF or WPS



Basic Steps

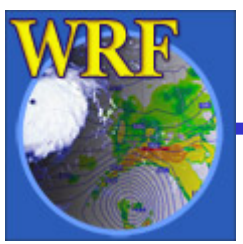
0) Check system requirements

1) Download source code

2) Download static terrestrial data

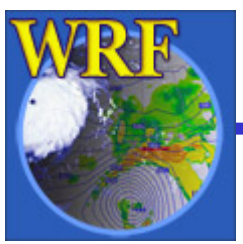
3) Install WRF

4) Install WPS



1) Download source code

- WPS is designed to work with WRF v2.2 and later
  - Users with earlier versions of WRF will need to upgrade to WRF v2.2
- Download source code from WRF download page:
  http://www.mmm.ucar.edu/wrf/users/download/get_source.html
  - WPSV2.2.TAR.gz – the WPS source code
  - WRFV2.2.TAR.gz – the WRF source code



Basic Steps

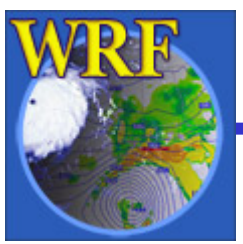
0) Check system requirements

1) Download source code

2) Download static terrestrial data

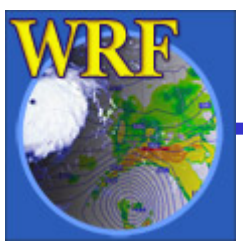
3) Install WRF

4) Install WPS



2) Download static terrestrial data

- The terrestrial fields interpolated by *geogrid* may be downloaded from same page as code
  - *Two options for data: low-res and all resolutions*
- Some terrestrial fields are available in several resolutions
- Since these data are static, they only need to be downloaded once
  - *Data can be shared by users on the same machine by placing files in a common directory!*



2) Download static terrestrial data

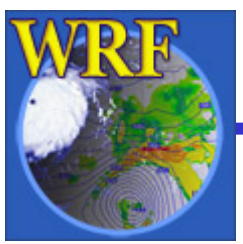
- The `geog.tar.gz` file (all resolutions) contains:

albedo_ncep – monthly surface albedo

greenfrac – monthly vegetation fraction

islope – slope index

landuse – land use category
(30”, 2’, 5’, and 10’ resolutions)



2) Download static terrestrial data

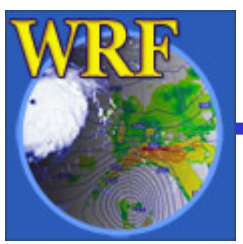
maxsnowalb – maximum snow albedo

soiltemp – annual mean deep soil temperature

soiltype_bot – bottom-layer soil type
(30", 2', 5', and 10' resolutions)

soiltype_top – top-layer soil type
(30", 2', 5', and 10' resolutions)

topo – topography height
(30", 2', 5', and 10' resolutions)

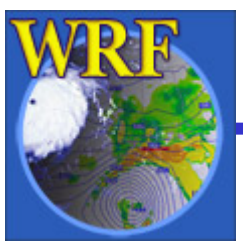


2) Download static terrestrial data

- Uncompress the data into a directory with ~10 GB of available space (264 MB for low-res only)!

```
> tar xzf geog.tar.gz
> rm -f geog.tar.gz
```

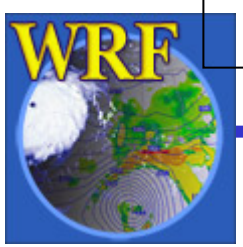
- *Since data files occupy 10 GB, placing data in a common location is recommended; all users can share this data*



2) Download static terrestrial data

- After uncompressing `geog.tar.gz`

```
> ls
geog
> ls geog
albedo_ncep     landuse_30s          …          …
greenfrac       landuse_5m           …          …
islope          maxsnowalb           …          …
landuse_10m     soiltemp_1deg        …          …
landuse_2m      soiltype_bot_30s     …          …
```



Basic Steps

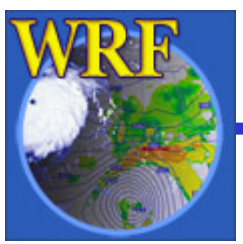
0) Check system requirements

1) Download source code

2) Download static terrestrial data

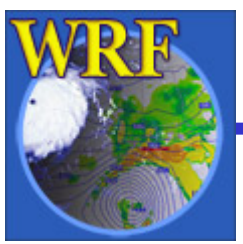
3) Install WRF

4) Install WPS



3) Install WRF

- WPS requires WRF v2.2 to be installed
  - WPS programs use WRF I/O API libraries to do file input and output
  - These I/O libraries are built when WRF is installed
- See ARW User's Guide for more detailed instructions on installing WRF v2.2



Basic Steps

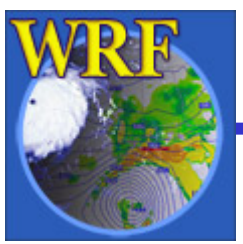
0) Check system requirements

1) Download source code

2) Download static terrestrial data

3) Install WRF

4) Install WPS

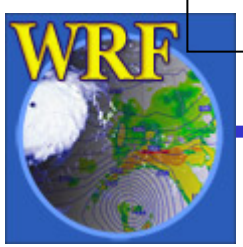


4) Install WPS

- Recall the directory where the WPS source code was downloaded (or copied) to
- Change to that directory and un-tar the code

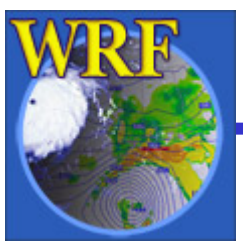
```
> gzip -d WPS.tar.gz
> tar xv WPS.tar
> ls
WPS
WRFV2
```

Assumed to exist! For simplicity, install WRFV2 in the same location as WPS



4) Install WPS

- Multiple users may share a single installation of the WPS; not every user needs to install
  - Each user will run WPS programs in their own working directories
  - WPS installation directory read-only, and output files created in user working directories
- Change to the WPS directory and configure WPS for installation



4) Install WPS

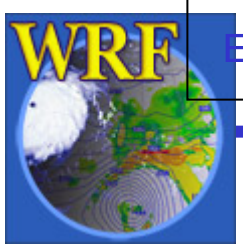
```
> cd WPS

> ./configure

Will use NETCDF in dir: /usr/local/netcdf-pgi
$JASPERLIB or $JASPERINC not found in environment, configuring to build
without grib2 I/O...
------------------------------------------------------------------------
Please select from among the following supported platforms.

   1.  PC Linux i486 i586 i686, PGI compiler serial, NO GRIB2
   2.  PC Linux i486 i586 i686, PGI compiler serial
   3.  PC Linux i486 i586 i686, PGI compiler DM parallel, NO GRIB2
   4.  PC Linux i486 i586 i686, PGI compiler DM parallel
   5.  PC Linux i486 i586 i686, Intel compiler   serial, NO GRIB2
   6.  PC Linux i486 i586 i686, Intel compiler   serial
   7.  PC Linux i486 i586 i686, Intel compiler   DM parallel, NO GRIB2
   8.  PC Linux i486 i586 i686, Intel compiler   DM parallel
   9.  PC Linux i486 i586 i686, g95 compiler,    serial, NO GRIB2
  10.  PC Linux i486 i586 i686, g95 compiler,    serial

Enter selection [1-10] :
```



4) Install WPS

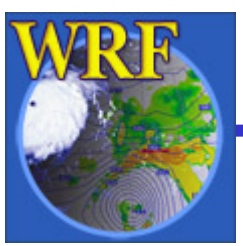
- When prompted by the configure script, select an option

```
Enter selection [1-4] : 1

------------------------------------------------------------

Configuration successful. To build the WPS, type: compile

------------------------------------------------------------
```



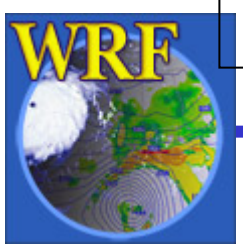
4) Install WPS

- If configuration was successful, compile:

```
> ./compile >&! compile.output
```

- After compilation, executables should exist in top-level WPS directory:

```
> ls -L *.exe
geogrid.exe
metgrid.exe
ungrib.exe
```



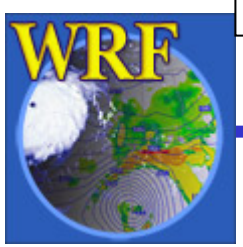
4) Install WPS

- If NCAR Graphics libraries are available:

```
> ./compile util >&! util.output
```

- After compilation, check for new executables:

```
> ls -L util/*.exe
avg_tsfc.exe        plotgrids.exe
g1print.exe         plotfmt.exe
g2print.exe         rd_intermediate.exe
mod_levs.exe
```



Questions?