COAMPS Grids, Projections, and Output Files in COAMPS-OS v1.X

Outline

I. COAMPS Model Basics
   A. Sigma Levels
   B. Vertical Grids
   C. Horizontal Grids
   D. Map Projections

II. COAMPS Flat Files
   A. File Naming and Location
   B. Available Fields
   C. Data Record
   D. Data Format

Model Basics - Sigma Levels

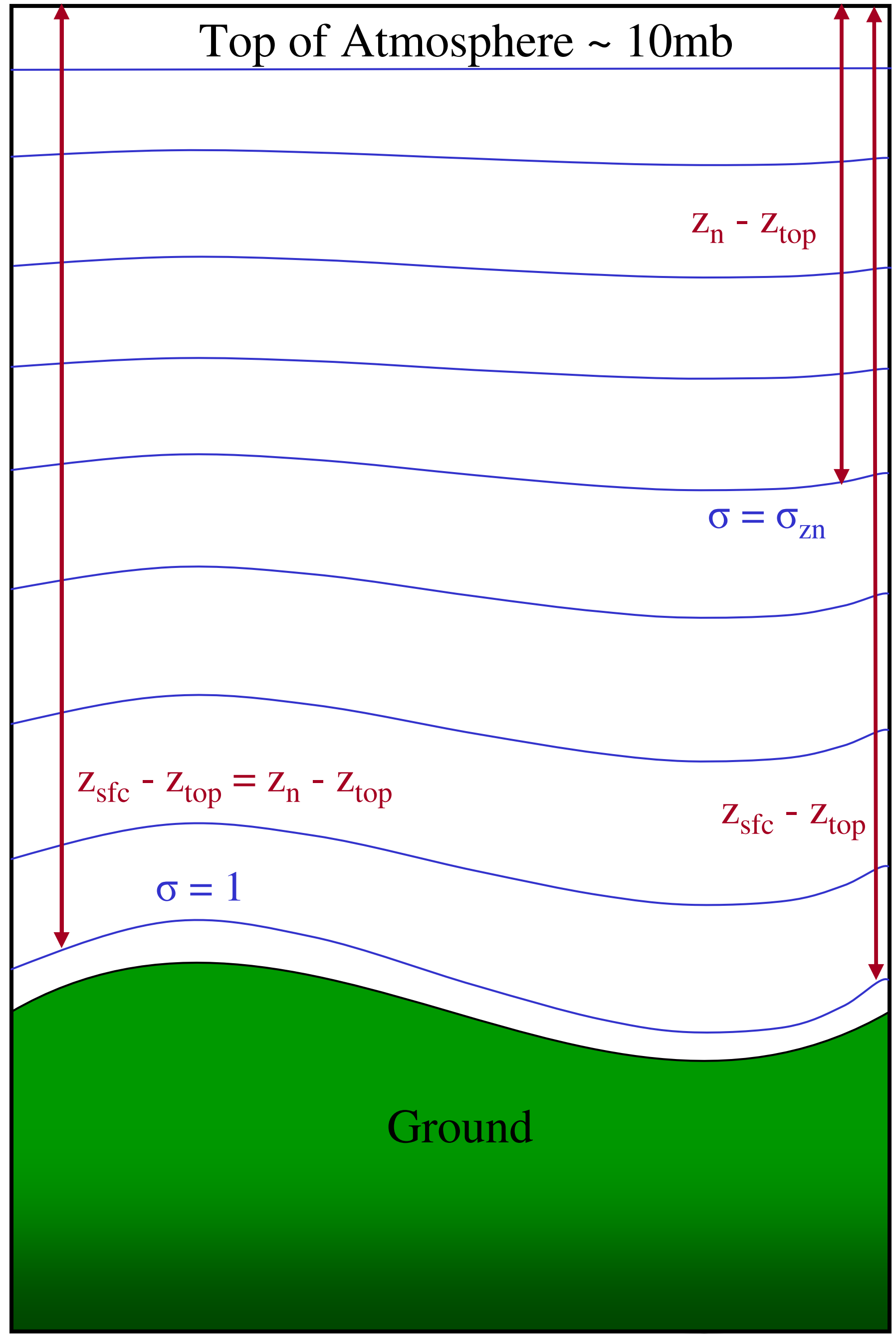
COAMPS uses *sigma-z coordinates* as the vertical coordinate. Sigma levels are terrain-following, non-dimensional, normalized height values, defined as

$$\sigma \equiv \frac{z - z_{top}}{z_{sfc} - z_{top}},$$

where z_{sfc} is surface height and z_{top} is the depth of the atmosphere or model domain.

The vertical index in COAMPS is from the top down, so level 1 represents the top of the model atmosphere.

Near the earth's surface, the model's vertical levels slope along with the terrain; at the top of the atmosphere, the vertical levels approach level surfaces. The number of vertical levels can vary, however, the default is 30.



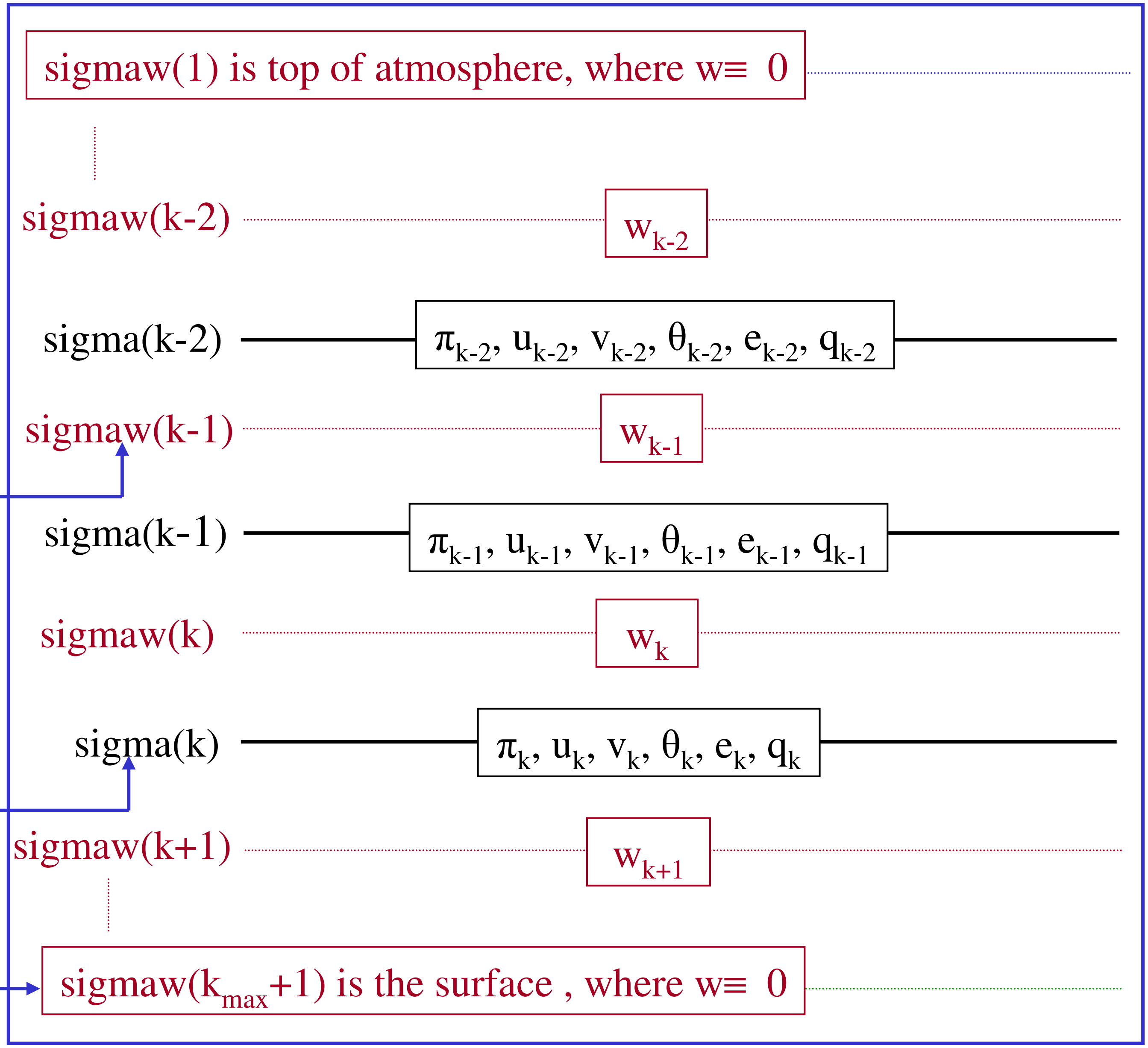
Model Basics - Vertical Grids

The vertical grid is also staggered, specifically for the calculation of vertical motion. Vertical staggering must be accounted for in output

Vertical motion, w, is computed on *sigmaw* levels, where k is the total number of vertical levels in the model.

All other prognostic variables are computed on levels halfway between the sigmaw levels.

The top and bottom of the model domain are both sigmaw levels.



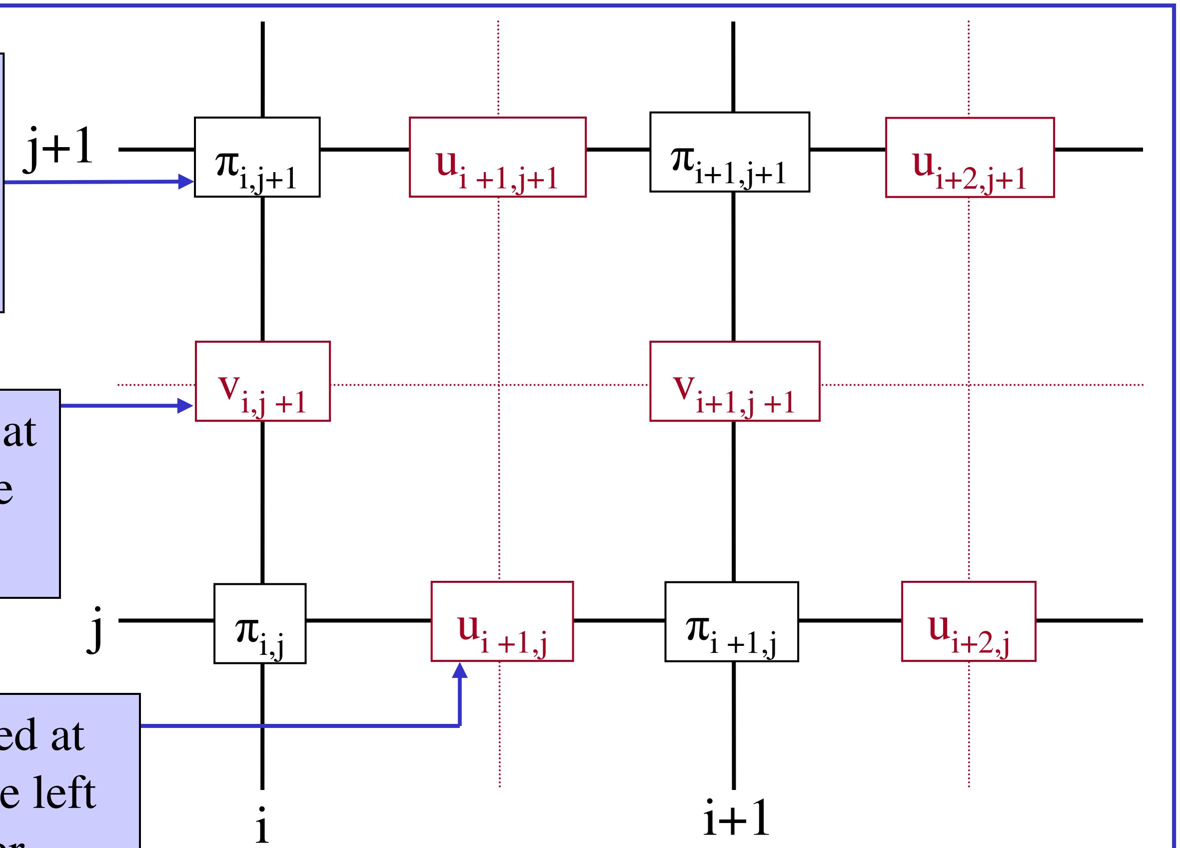
Model Basics - Horizontal Grids

The horizontal grid in COAMPS is a system of staggered grid points. This staggering (Arakawa-Lamb Scheme C) of momentum and mass variables improves the overall forecast accuracy, specifically in the treatment of geostrophic adjustment and linear convection.

Scalar variables (denoted by π) are computed at the center of the boxes formed by the mesh. These are the common model grid points.

V wind components are computed at points half the grid distance above and below the grid box center.

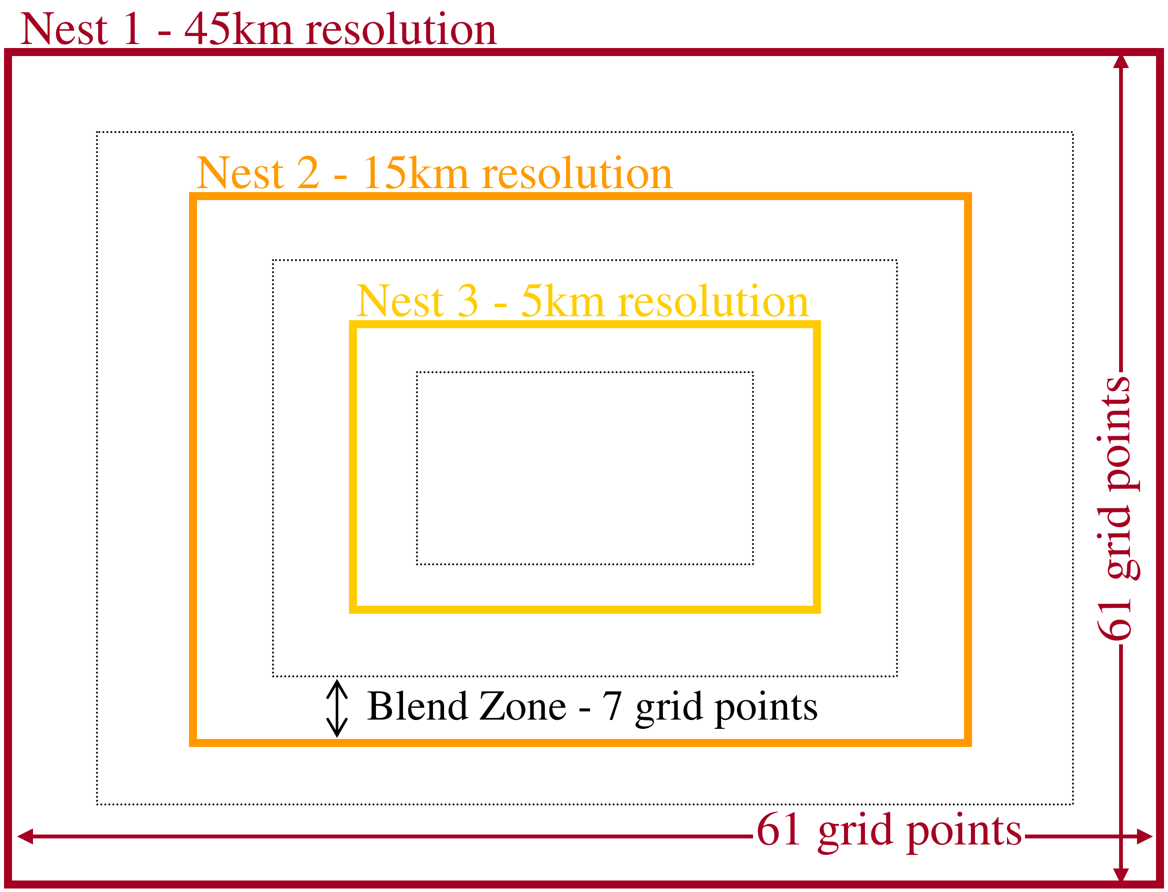
U wind components are computed at points half the grid distance to the left and right of the grid box center.



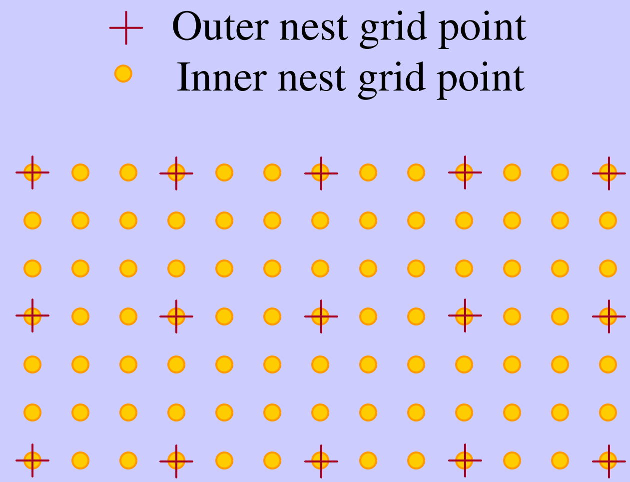
Staggering of the U and V fields must be accounted for in the sigma-level output.

Horizontal Resolution

A Sample Nest Setup



For computational efficiency, COAMPS is designed to nest with a factor of three. For simplicity, each parent grid point corresponds to an inner nest grid point.



A 7-grid point *blend zone* at the boundary of each nests blends the information in the outer nests (or background fields) into the interior. Interior nests cannot extend into this zone.

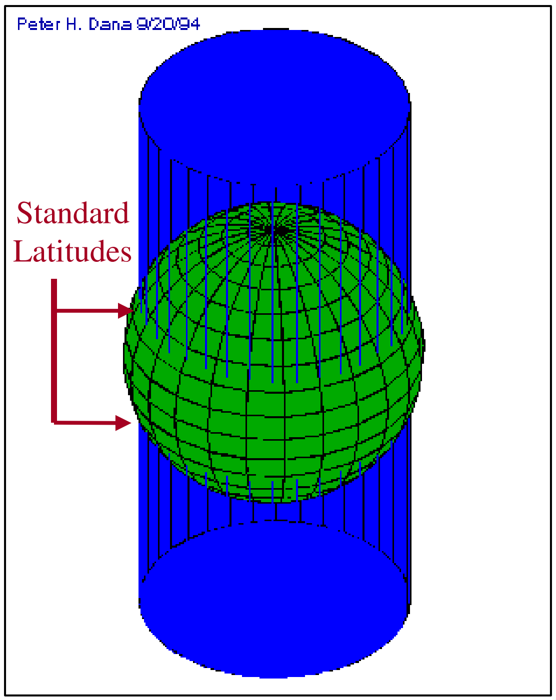
Each interior nest should have a number of grid points that is a multiple of 3, plus 1. For instance, 37 is the minimum number of grid points, followed by 40, 43, 46, 49...

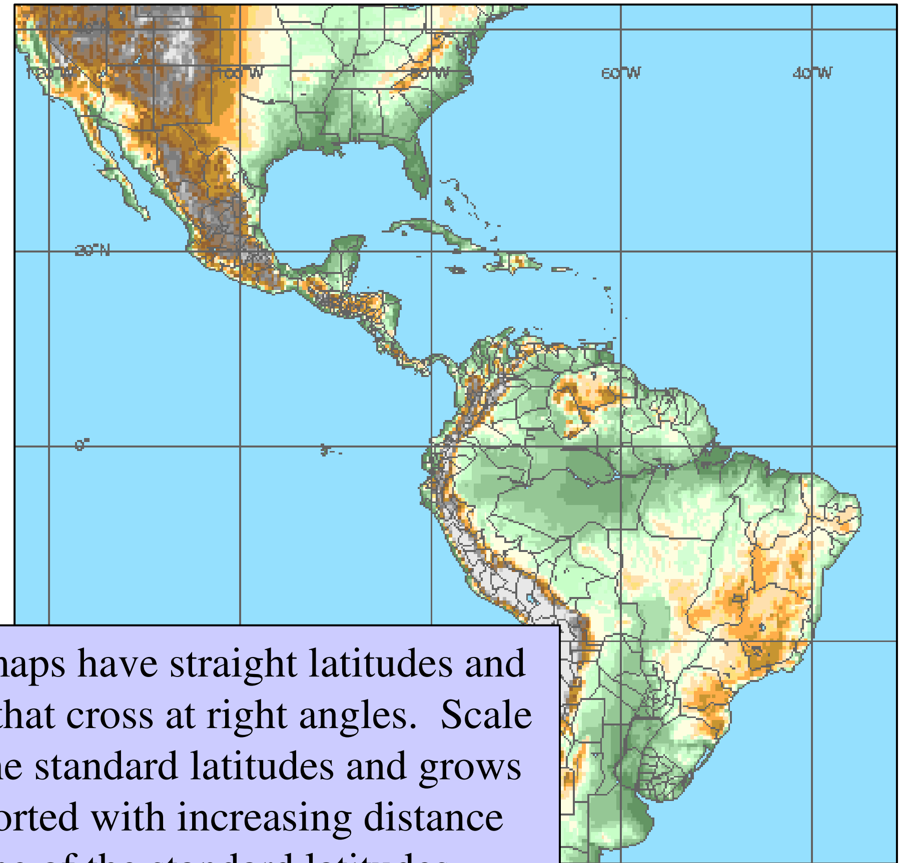
Choosing an Appropriate Map Projection

Projections are chosen to minimize distortion of both visualization and the computational grids with respect to the curved surface of the earth; this increases computational efficiency.

Cylindrical Projection

Projection of a spherical surface onto a cylinder that is tangent to or slices through the earth.





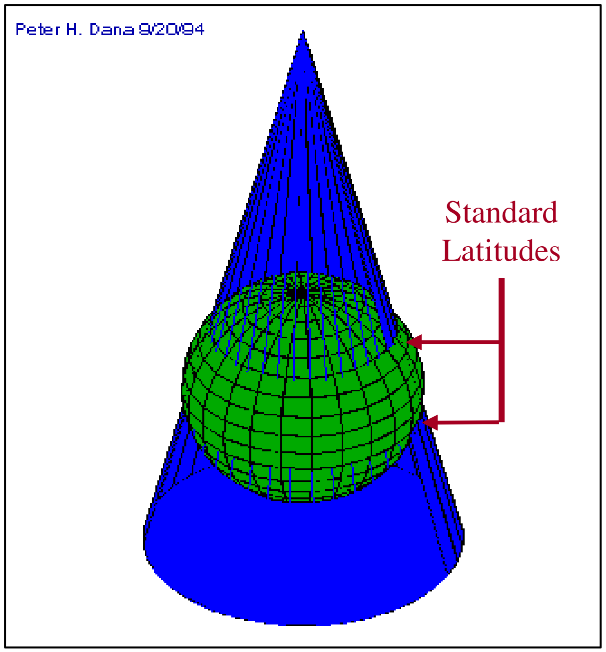
Mercator maps have straight latitudes and longitudes that cross at right angles. Scale is true at the standard latitudes and grows more distorted with increasing distance from one of the standard latitudes.

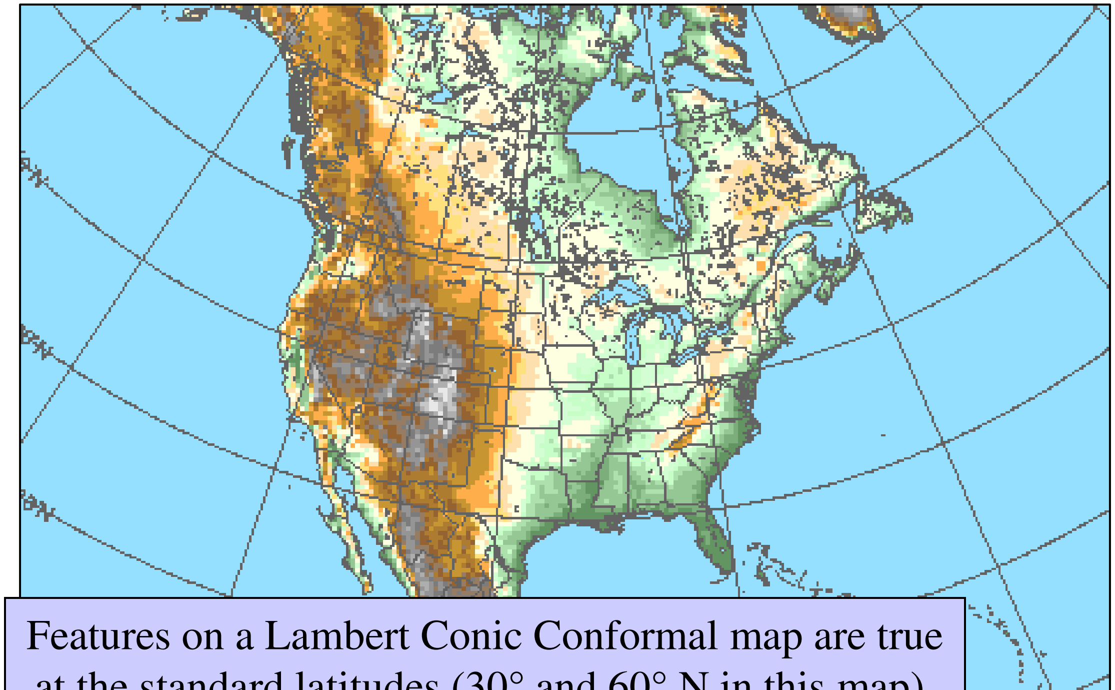
In this case, a *Mercator* projection, the cylinder slices through the earth at two latitudes equidistant from the equator.

Choosing an Appropriate Map Projection

Projections are chosen to minimize distortion of both visualization and the computational grids with respect to the curved surface of the earth; this increases computational efficiency.

Conical Projection
Projection of a spherical surface onto a cone that is tangent to or slices through the earth.





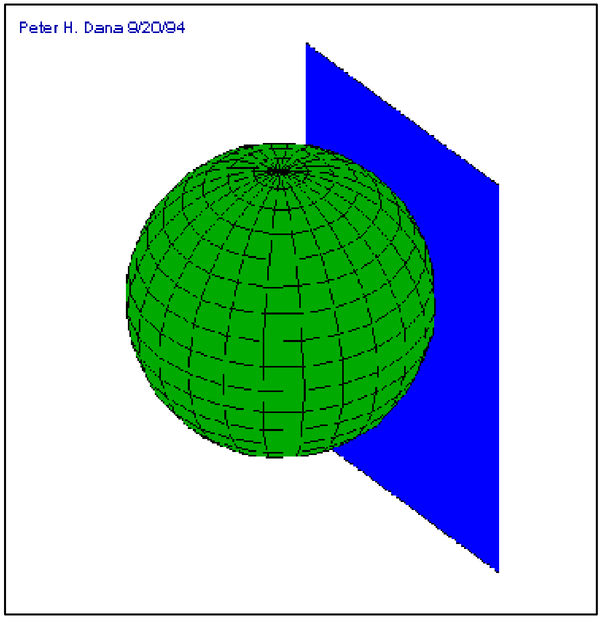
Features on a Lambert Conic Conformal map are true at the standard latitudes (30° and 60° N in this map). Distortion increases with increasing distance from one of the standard latitudes.

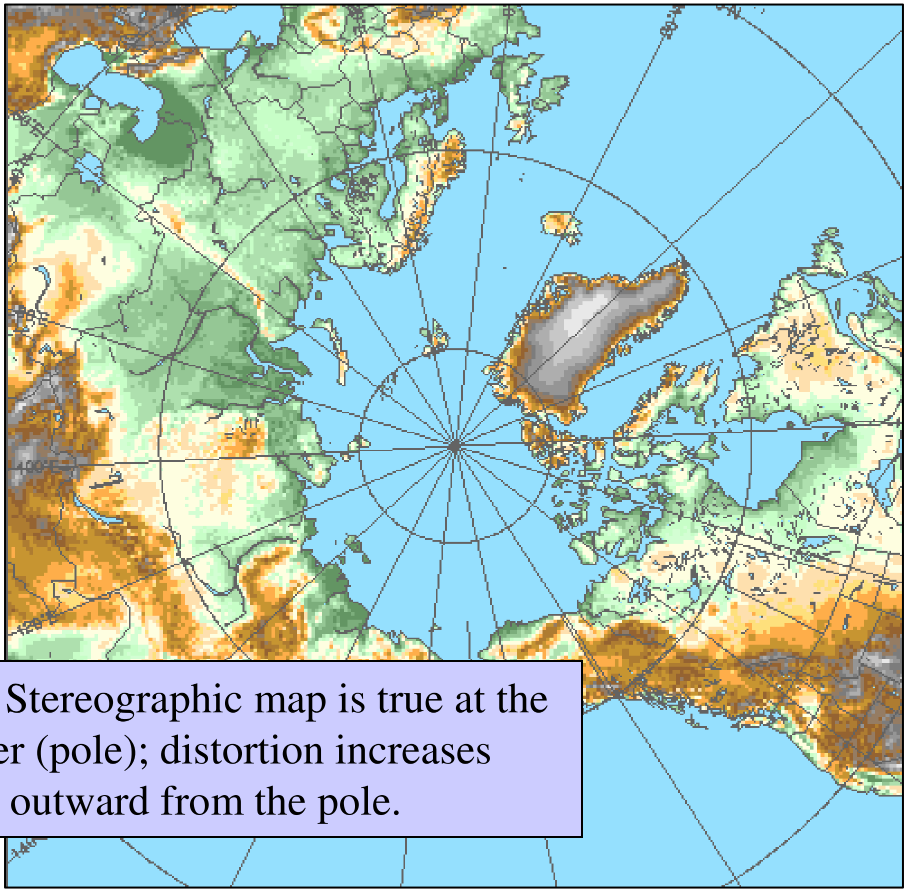
In this case, a *Lambert Conic Conformal* projection, the cone slices through the earth at two latitudes, known as standard latitudes.

Choosing an Appropriate Map Projection

Projections are chosen to minimize distortion of both visualization and the computational grids with respect to the curved surface of the earth; this increases computational efficiency.

Azimuthal Projection
Projection of a spherical surface onto a plane that is tangent to or slices through the earth.





A Polar Stereographic map is true at the center (pole); distortion increases outward from the pole.

When the plane is tangent to a pole, the resulting projection is *Polar Stereographic*.

Choosing an Appropriate Map Projection

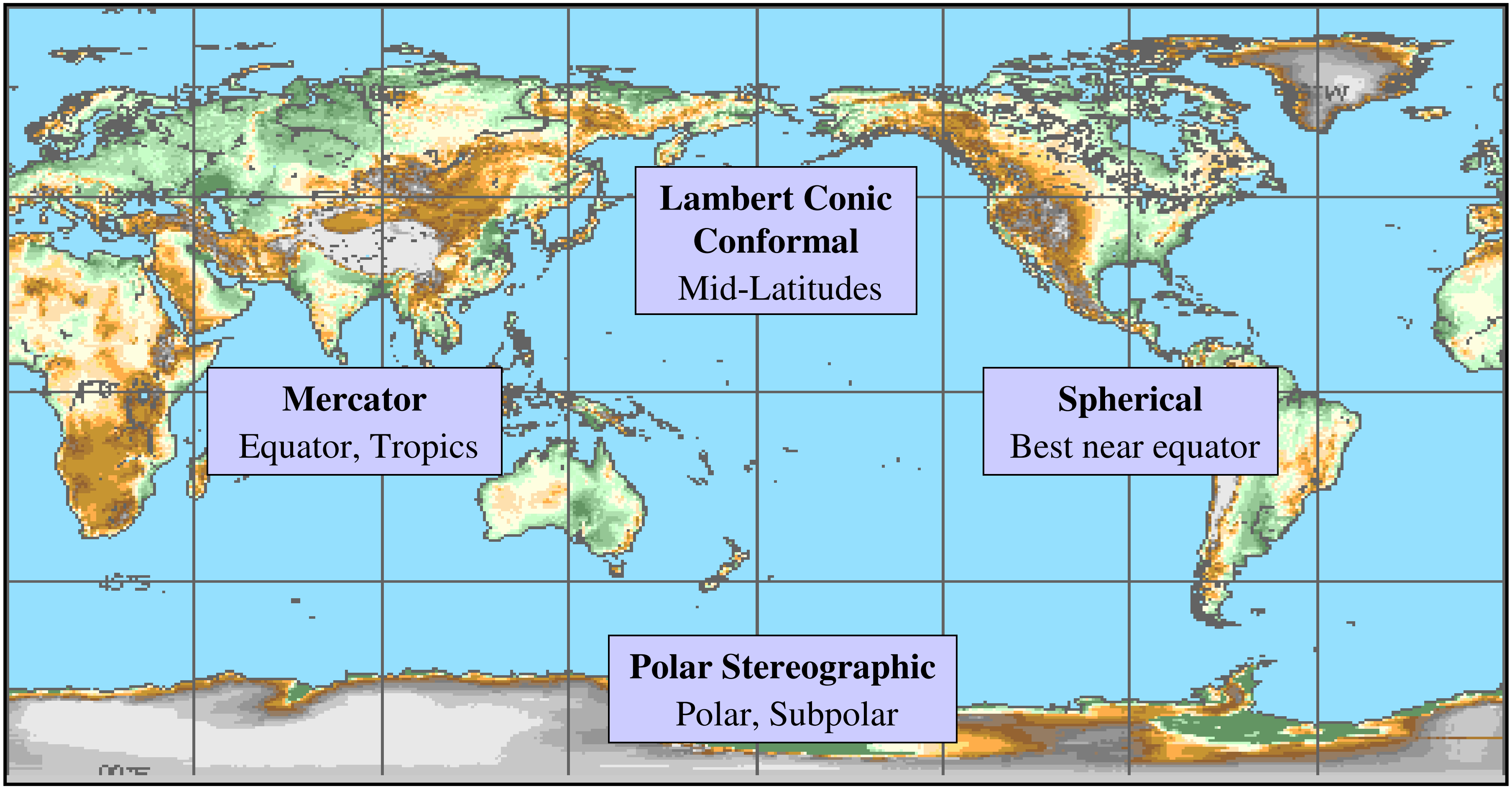
Projections are chosen to minimize distortion of both visualization and the computational grids with respect to the curved surface of the earth; this increases computational efficiency.

Unprojected Coordinates

Considers latitude and longitude coordinates as a rectangular coordinate system.

This projection is known as a *Spherical* projection. Maps with this projection are increasingly distorted away from the equator; considering latitude and longitude as rectangular coordinates may be useful for other applications of COAMPS output.

Choosing an Appropriate Map Projection



Lambert Conformal - Appropriate for domains extending between 18° and 64°. A grid of this projection may not cross the equator.

Spherical – Scale, distance, area and shape all distorted; may have value for other applications.

Mercator - Appropriate for domains near the equator; recommended poleward model domain extent should be less than 45°.

Polar Stereographic - Appropriate for latitudes above 60°.

Flat Files - Location and Names

The files use a 64-character naming convention. Files contain one level except for sigma level files. Sigma level files contain the data from all model sigma levels.

nnnnnn_ttt_llllll_kkkkkk_gfppppxpppp_yyyymmddhh_hhhhmmss_vvvvvvv

1 6 10 17 24 36 47 56 64

nnnnnn: Field identifier, see next slide

ttt: Level type

zht: constant height surface (m)
pre: constant pressure surface (mb)
sfc: surface (kkkkk=lllll=00000)
msl: mean sea level (kkkkk=lllll=00000)
sig: sigma surface

llllll: Level for which the field is valid

For sigma level fields - the bottom level

kkkkk: Thickness of level, if applicable

For sigma level fields - the top level

g: Grid identifier

f: Descriptor fluid type

a: atmosphere, o: ocean

ppppxpppp: Grid dimensions

Example: 0081x0121 represents 81 grid points in the x-direction and 121 in the y-direction

yyyymmddhh: 10-digit date-time-group

hhhhmmss: 8-digit forecast time

h: hours (0000-9999)
m: minutes (00-59)
s: seconds (0-59)

vvvvvvv: File Type

analinc: Analysis increment file
bndyfld: Boundary file
climfld: Climatology file
fcstfld: Forecast file
infofld: Ascii information file
obsdata: Observation file

Examples - Filenames

The files use a 64-character naming convention. Files contain one level except for sigma level files. Sigma level files contain the data from all model sigma levels.

vpress_pre_000700_000000_2a0061x0061_2003030612_00060000_fcstfld

Contains the grid 2 forecast of vapor pressure at 700-mb for time step 6:00:00. The basetime of the model run is 1200Z on 12 March 2003. The grid contains 61 grid points in the x-direction and 61 grid points in the y-direction.

slpres_msl_000000_000000_3a0043x0076_2001122100_00203000_fcstfld

Contains the grid 3 forecast of sea level pressure for time step 20:30:00. The basetime of the model run is 0000Z on 21 December 2001. The grid contains 43 grid points in the x-direction and 76 grid points in the y-direction.

snomix_sig_031050_000010_1a0121x0088_1998021412_00000000_fcstfld

Contains the grid 1 analysis field for snow mixing ratio. The basetime of the model run is 1200Z on 14 February 1998. The grid contains 121 grid points in the x-direction and 88 grid points in the y-direction. The file contains data on sigma levels beginning at 31050 meters to 10 meters.

Flat Files - Surface/Special Level Field Identifier List

nnnnnn	Field Name	Units	nnnnnn	Field Name	Units
airtmp	air temperature	K	**sehflx**	sensible heat flux	W/m^2
albedo	albedo	fraction	**totflx**	total heat flux	W/m^2
bkrich	bulk Richardson number	numeric	**ttlpcp**	total precipitation	kg/m^2
ceilht	cloud ceiling height	m	**ttlsac**	accumulated snow	kg/m^2
cldbas	cloud base height	m	**snowdp**	snow depth	mm
cltopt	cloud top temperature	K	**solflx**	solar radiation	W/m^2
conpac	accum. convective precipitation	kg/m^2	**soltmp**	deep soil temperature	K
conpcp	bucket convective precipitation	kg/m^2	**stapac**	accum. stable precipitation	kg/m^2
datahd	COAMPS data record	ascii	**stapcp**	stable precipitation	kg/m^2
evapdh	evaporative duct height	m	**stresu**	wind stress grid-relative u-comp	kg/m^2
frzzht	freezing level	m	**stresv**	wind stress grid-relative v-comp	kg/m^2
grdtmp	ground/sea temperature	K	**terrht**	terrain height	m
grdwet	ground wetness	fraction	**trpres**	terrain pressure	mb
htindx	heat index	K	**ttlrad**	total radiation	W/m^2
lahflx	latent heat flux	W/m^2	**ttlsnw**	total snow	kg/m^2
latitu	latitude	degree	**ttstar**	temperature surface scaling	K
lndsea	land/sea flag	numeric	**ustrue**	wind stress true u-component	kg/m^2
lonflx	long wave radiation	W/m^2	**uustar**	wind surface scaling	m/s
longit	longitude	degree	**uutrue**	true wind u-component	m/s
nradfl	net radiation	W/m^2	**uuwind**	wind u-component	m/s
ocndep	ocean depth	m	**visibl**	visibility	km
qqstar	mixing ratio surface scaling	kg/kg	**vpress**	vapor pressure	mb
pblzht	boundary layer height	m	**vstrue**	wind stress true u-component	kg/m^2
roughl	roughness length	m	**vvtrue**	true wind v-component	m/s
seaice	ice coverage	fraction	**vvwind**	wind v-component	m/s
seatmp	sea temperature	K	**wstres**	wind stress	kg/m^2

Flat Files - Sigma Level Field Identifier List

nnnn	Field Name	Units	nnnn	Field Name	Units
cldmix	cloud mixing ratio	g/g	**snomix**	snow mixing ratio	g/g
emixht	eddy mixing coeff. - heat	m^2/s^3	**ttlprs**	pressure	mks
emixmt	eddy mixing coeff. - momentum	mm/s	**turbke**	turbulent kinetic energy	m^2/s^2
icemix	ice mixing ratio	g/g	**uuwind**	wind u-component	m/s
perprs	pressure perturbation	mb	**vvwind**	wind v-component	m/s
pottmp	potential temperature	K	**wvapor**	mixing ratio	kg/kg
ranmix	rain mixing ratio	g/g	**wwwind**	wind vertical velocity	mb/s

Flat Files - Pressure Level Field Identifier List

nnnn	Field Name	Units	nnnn	Field Name	Units
absvor	absolute vorticity	s^{-1}	**ttlrad**	total radiation	W/m^2
airtmp	air temperature	K	**uutrue**	true wind u-component	m/s
diverg	divergence	s^{-1}	**uuuinc**	wind u-component increment	m/s
dwdpin	dewpoint depression increment	K	**uuwind**	grid-relative wind u-component	m/s
dwptdp	dewpoint depression	K	**vpress**	vapor pressure	mb
geopht	geopotential height	gpm	**vvtrue**	true wind v-component	m/s
ghtinc	geopotential height increment	gpm	**vvvinc**	wind v-component increment	m/s
relhum	relative humidity	percent	**vvwind**	grid-relative wind v-component	m/s
tmpinc	air temperature increment	K	**wwwind**	wind vertical velocity	mb/s
ttlcvr	total cloud coverage - relh	percent			

Flat Files - Height Surface Field Identifier List

nnnn	Field Name	Units
absvor	absolute vorticity	s^{-1}
airtmp	air temperature	K
cldmix	cloud mixing ratio	g/g
diabht	diabatic heating	K/s
diverg	divergence	s^{-1}
dwptdp	dewpoint depression	K
emixht	eddy mixing coeff. - heat	m^2/s
emixmt	eddy mixing coeff. - momentum	m^2/s
evapcl	evaporative cooling	K/s
gmrefr	modified refractivity gradient	km^{-1}
icemix	ice mixing ratio	g/g
perpot	perturbation potential temp	K
perprs	perturbation pressure	mb

nnnn	Field Name	Units
ranmix	rain mixing ratio	g/g
relhum	relative humidity	percent
smixco	Smagorinsky eddy mixing coeff.	m^2/s
snomix	snow mixing ratio	g/g
ttlrad	radiative heating rate	K/s
turbke	turbulent kinetic energy	$(m/s)^2$
turbml	turbulent mixing length	m^2/s
uutrue	true wind u-component	m/s
uuwind	grid-relative wind u-component	m/s
vpress	vapor pressure	mb
vvtrue	true wind v-component	m/s
vvwind	grid-relative wind v-component	m/s
wwwind	wind vertical velocity	mb/s

Flat Files - The COAMPS Data Record

Each COAMPS run produces an ascii file containing grid and projection information for the current date-time-group. The file prefix is “datahd”.

Array Elements

Array Elements	
1-29	Coarse Domain General Information
30-500	Nested Domain Information
501-800	Vertical Grid Spacing Information
801-1000	Sigma Level Information
1001-1200	Pressure Level Information
1301-1800	Idealized Sounding Information

Significant Elements (1-29)

2: Number of vertical levels (kka)
3: Projection type
4: Standard latitude 1
5: Standard latitude 2
6: Standard longitude
7: Reference latitude
8: Reference longitude
11: Number of nests

Significant Elements (30-500)

30: Number of x-gridpoints
31: Number of y-gridpoints
34: Reference i-point
35: Reference j-point
37: x-gridspacing
38: y-gridspacing
39: SW corner latitude
40: SW corner longitude
41: SE corner latitude
42. SE corner longitude
43: NE corner latitude
44: NE corner longitude
45: NW corner latitude
46: NW corner longitude
For each nest, elements repeat every 30 entries.

Significant Elements (501-800)

501 through (501+kka):
Thickness of sigma layers

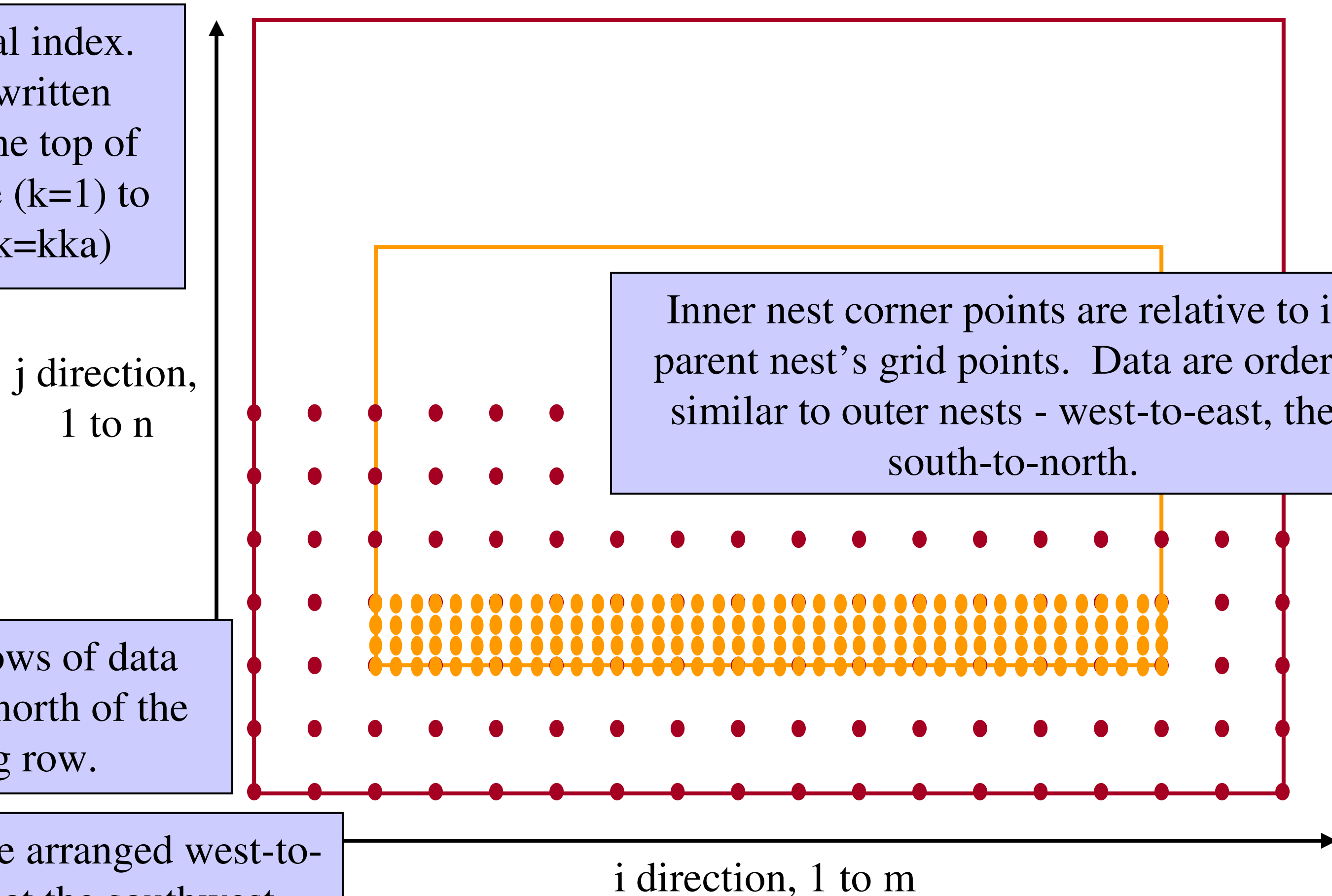
Significant Elements (801-1000)

801 through (801+kka):
Sigma level values for all prognostic variables but vertical motion

Flat Files - Data Format

2D and 3D COAMPS flat files are IEEE 32-bit unformatted files, written using FORTRAN. Example code to read these files appear on the following slide. Routines to convert between latitude/longitude and grid i/j space are also available.

k is the vertical index. 3D files are written beginning at the top of the atmosphere (k=1) to the surface (k=kka)



Inner nest corner points are relative to its parent nest's grid points. Data are ordered similar to outer nests - west-to-east, then south-to-north.

Subsequent rows of data follow, to the north of the preceding row.

Rows of data are arranged west-to-east, starting at the southwest corner of the domain.

Flat Files - Data Format

Example FORTRAN code to read fields

```
parameter (lend=m*n*k)       ! 3D sigma volume; k=1 for 2D fields
real d(m*n*k)
character*164 filename
open (unit=33, file=filename, form='unformatted', access='direct',
status='old')
read (33) (d(i),i=1,lend)
close (33)
```

Example C code to read fields

```
FILE *fp = 0;
int lend = m*n*k;          /* 3D sigma volume; k=1 for 2D fields */
char *filename;
float *d = 0;
d = ( char * ) calloc ( 1, lend * sizeof ( float ) );
fp = fopen ( filename, “r” );
fread ( d, 1, lend * sizeof ( float ), fp );
fclose ( fp );
```