



Norwegian
Meteorological
Institute

Gridded data from many sources

A data-user's perspective

Heiko Klein

26.09.2014

Background

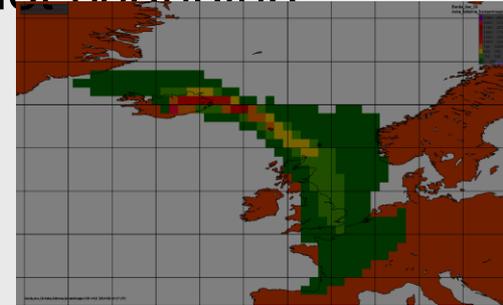
- MET used legacy format (felt) for gridded data since ~1980s
- Index 2d fields
- «unique» parameter table
- 2012 decided to use netcdf / CF for all gridded data internally
- Grib still allowed as exchange format
- Grib used in atmospheric models
- since autumn 2013: all new products are in netcdf
- since spring 2014: felt no longer needed in post-production

Model-input: e.g. Regional ocean models

- Need surface fields, usually U10, V10, (T, P)
- Usually not 100% overlap between atmospheric regional model and ocean model – merging of several atmospheric models required
- Model expects unified input
- Tasks:
 - Rename variable names (Uair, Vair, Tair, Pair)
 - Use unique units
 - Remove scaling-factors (in particular, U/V scaling factors change when changing projection – loss of precision)
 - Interpolate datasets / take care of land-sea mask
 - Join dataset
- => Heavy preprocessing required, independent of input-data

Model-input: e.g. Atmos. Dispersion models

- Needs model-level parameters: U, V, T, Q, CLWC, P, + surface (u10, v10, precip., ...)
- Some derived parameters (w, deaccumulated precip.)
- Model-domain = atmospheric model-domain
- Many domains (local, regional, global), but usually not changing too often (~ once a year)
- Emergency-models – response-time < 10minutes
- Tasks:
 - Rename variable names
 - Precalculation of derived parameters (on demand)
 - Hard-coding of model-domain (Fortran)



⇒ Preprocessing should be simple (Fortran), otherwise storing of specialized dataset for emergency model?

Model input: grib / netcdf

- Models which require heavy preprocessing of data usually don't care
- Emergency models need to read data as is, often can read many formats (grib and netcdf)
- Netcdf-advantage due to index, but
- Coordinate-systems are tough to read in Fortran
- Usually semi hard-coded dimensions
- Rely upon stable product – **no changes to dimensions and attributes, only change data**
- Grib seems often simpler – but is it really?

Real world, bad example of grib

·atmopl09190000_003.grib

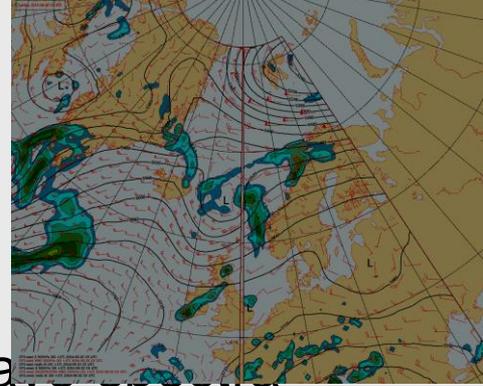
·latFirst	latLast	shortName	dataDate
·40	0	r	20140919
·0	-5	r	20140919
·40	0	t	20140919
·0	-5	t	20140919

·haven't found a general reader which can handle it good enough

·**Netcdf -> grib conversion simple**

·**Grib -> netcdf conversion can be impossible (What belongs together?)**

Visualization



- Maps, time-series, vert. cross-sections/sondes, wave spectra
- Products must show up within 2s on a meteorological workstation
- IO: ~50 fields / s
- Pre-defined units and legends – based on standard_name or variable-name
- Simple derived parameters are calculated on demand (variable-name based)
- Tasks:
 - Rename variable-names to those expected for presentation (or define several presentations)
 - Index data (grib, not netcdf)
 - Startup-delay for time-series: Extract time-series for all stations: ~minutes
 - Preprocess – data-duplication to obtain performance

Chunking?

• Maps

- `float air_temperature_2m(time, height1, y, x) ;`
- `air_temperature_2m:_ChunkSizes = 1, 1, 929, 719 ;`

• Time-Series (not used at MET)

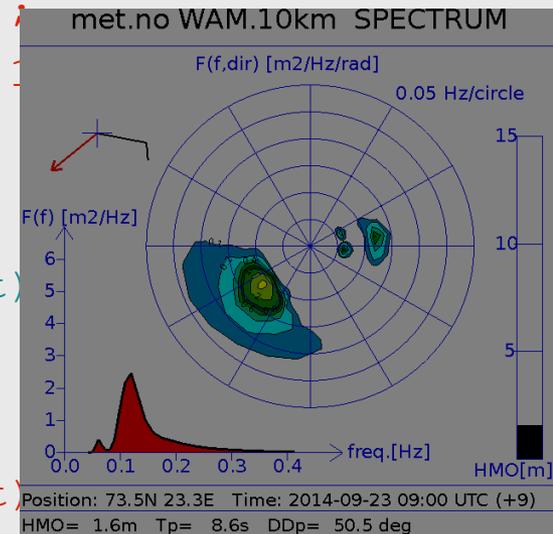
- `float air_temperature_2m(time, height1, y, x) ;`
- `air_temperature_2m:_ChunkSizes = 96, 1, 1, 1 ;`

• Maps (not used at MET)

- `float wave_spectra(time, angle, freq, lon, lat) ;`
- `wave_spectra:_ChunkSizes = 1, 1, 1, 929, 1, 1 ;`

• Wave-Spectra

- `float wave_spectra(time, angle, freq, lon, lat) ;`
- `wave_spectra:_ChunkSizes = 1, 36, 36, 1, 1 ;`



•=> chunking can increase performance, but chunking is also a «specialized» product

standard_name

- Very useful as documentation
- Good (enough) for discovery metadata
- Good first guess to add visualization hints – but needs often more tuning
- No way to add non-standard_names «metno_name»
- Often files with wrong standard_names
- Long standardization process (~3 weeks discussion, 3 weeks silence = eventual acceptance, 3-12 month until it appears on web-page) – too long for people working in the field
- (I have a list of ~1500 standard_names which I don't dare to send in.)
- I have local grib-extensions which no grib-reader can read.)

standard_name

- Several variables in one file might have same standard_name (e.g. pressure-level/model-level/surface)
- Not precise enough – often only correct together with dimensions / cell_method / bounds
- Usage of standard_name as variable name:
 - air_temperature_ml
 - air_temperature_6hour_max

Visualization grib / netcdf

- Map-visualization works well, grib requires indexing
- Usage of libraries to handle data-conversion and CF-coordinate system (C++: fimex,cdo; Java: netcdf-java)
- External tables (grib) / configuration (netcdf/ncml) needed in most cases
- **Netcdf can be used for most specialized, preprocessed products, e.g. wave-spectra (chunking), pre-calculated cross-sections**
- High-performance web-servers (<http://yr.no>) need still very special solutions (caching, parallelization, pre-calculations)

