OPeNDAP access of grids

netCDF/OPeNDAP are the perfect file format/web service for gridded data. Gridded data exist in various flavours of complexity.

Grid topology

- **Orthogonal grid**
  - An orthogonal lat-lon grid: climate data products often appear in this topology. There is a separate lat vector (e.g. -180:1:180 deg) and a separate lon vector (e.g. -90:1:90 deg) that span a part of the globe. Support for these grids is always available. Example: http://opendap.deltares.nl/thredds/catalog/opendap/deltares/FEWS-IPCC/catalog.html

    The *minimal* netCDF dump of this grid type looks like

    ```
    dimensions:
            lon = 3 ;
            lat = 5 ;
    variables:
            single VARIABLE(lon,lat), shape = [3 5]
                    VARIABLE:grid_mapping = "wgs84"
    \\ lat-lon vectors with mapping info
            single lon(lon), shape = [3]
            single lat(lat), shape = [5]
            int32 wgs84([]), shape = [1]
    ```

  - An orthogonal x-y grid: national data products often appear in this topology. There is a separate x vector (e.g. 0:100:280000 m) and a separate y vector (e.g. 624800:-100:300000) that span a part in a local projection. Support to plot these grids in lat-lon space is less often available, as it assumes that the client can deal with the two methods to prescribe lat-lon coordinates. The first, implicit option is assuming that the client can do the conversion from the local projection to the a WGS84 lat-lon projection. The CF convention describes how to add projection meta-data to the netCDF file (grid_mapping attribute). In addition, the data producer can transform the x and y vectors to complete lat and lon grids, prescribing the WGS84 position of each grid cell separately. The CF convention describes how to attach these global coordinates to the dataset (coordinates attribute). OpenEarth recommend to supply both options in netCDF files. This option is in fact the same as considering the orthogonal x-y grid as a curvi-linear lat-lon grid, as discussed next. Example: http://opendap.deltares.nl/thredds/catalog/opendap/tno/ahn100m/catalog.html with clearly defined local projection (Dutch RD) and with full lat-lon matrices included.
    http://opendap.deltares.nl/thredds/catalog/opendap/knmi/NOAA/mom/1990_mom/5/catalog.html without clearly defined local projection (a polar stereographic one) but with full lat-lon matrices included.

    The *minimal* netCDF dump of this grid type looks lile

    ```
    dimensions:
            x = 3 ;
            y = 5 ;
    variables:
            single VARIABLE(x,y), shape = [3 5]
                    VARIABLE:coordinates = "lat lon" \\ here we optionally connect the full lat-lon
    grids
                    VARIABLE:grid_mapping = "epsg" \\ here we choose the x-y sticks
    \\ x-y vectors with mapping info
            single x(x), shape = [3]
                    x:grid_mapping = "epsg"
            single y(y), shape = [5]
                    y:grid_mapping = "epsg"
            int32 epsg([]), shape = [1]
    \\ optional: add full lat-lon matrices as if it were a curvi-linear grid
            single lon(x,y), shape = [3 5]
                    lon:grid_mapping = "wgs84"
            single lat(x,y), shape = [3 5]
                    lat:grid_mapping = "wgs84"
            int32 wgs84([]), shape = [1]
    ```

- **Curvi-linear grid**: plotting support for curvi-linear grids depends on the client. Curvi-linear grids often exist, key examples are: (i) individual satellite images that are orthogonal in pixel space, but fully warped in lat-lon space and (ii) curvi-linear coastal circulation models that are orthogonal in grid index space, but that have been warped to fit local coast lines.
  - A curvi-linear lat-lon mesh: Example: remote sensing swath
    http://data.nodc.noaa.gov/opendap/ghrsst/L2P/AVHRR19_L/NEODAAS/2010/348/20101214-AVHRR19_L-NEODAAS-L2P-14dec100129_wsst.8bit-v01.nc.bz2

The *minimal* netCDF dump of this grid type looks like

```
dimensions:
        j = 3 ;
        i = 5 ;
variables:
        single VARIABLE(col,row), shape = [3 5]
                VARIABLE:coordinates = "lat lon" \\ here we connect the full lat-lon grids
                VARIABLE:grid_mapping = "wgs84"
\\ mandatory: add full lat-lon matrices as if it were a curvi-linear grid
        single lon(col,row), shape = [3 5]
                lon:grid_mapping = "wgs84"
        single lat(col,row), shape = [3 5]
                lat:grid_mapping = "wgs84"
        int32 wgs84([]), shape = [1]
```

- A curvi-linear x-y mesh: This kind of data is the regular output type of regional ocean circulation models. The CF convention allows to attach only one set of coordinates, either the lat-lon matrices or the x-y matrices. Example:

The *minimal* netCDF dump of this grid type looks like

```
dimensions:
        j = 3 ;
        i = 5 ;
variables:
        single VARIABLE(col,row), shape = [3 5]
                VARIABLE:coordinates = "lat lon" \\ we can connect only one set of full coordinate
matrices
\\                VARIABLE:coordinates2 = "x y"  \\ the other we can't connect within the CF
standards
                VARIABLE:grid_mapping = "epsg"
\\ define full lat-lon matrices
        single lon(col,row), shape = [3 5]
                lon:coordinates = "x y"
                lon:grid_mapping = "wgs84"
        single lat(col,row), shape = [3 5]
                lat:coordinates = "x y"
                lat:grid_mapping = "wgs84"
        int32 wgs84([]), shape = [1]
\\ define full x-y matrices (extra)
        single x(col,row), shape = [3 5]
                x:grid_mapping = "epsg"
        single y(col,row), shape = [3 5]
                y:grid_mapping = "epsg"
        int32 epsg([]), shape = [1]
```

- **Unstructured grid**: Support for unstructured grids has not been standardized. A proposals can be found at the Deltares netCDF wiki

Single grid vs. tiled grids

For large areas with high resolutions often single netCDF files become too big. Also when there are large parts with _Fillvalue, netCDF3 files can become too large. netCDF4 of course solves this issue by allowing for internal zipping of variables, but not all applications can handle netCDF4 yet (arcGis 9.2, R for windows, ncBrowse,...). So tiling of the full coverage into sub netCDF files is a solution.

- Single grid coverages: one file contains all the data of the entire space covered by the dataset.
- A set of tiled grid coverages: one file contains only the data of a subset of the entire space covered by the dataset. The entire space is . Example:
  http://opendap.deltares.nl/thredds/catalog/opendap/rijkswaterstaat/jarkus/grids/catalog.html
  http://opendap.deltares.nl/thredds/catalog/opendap/rijkswaterstaat/vaklodingen/catalog.html
  http://opendap.deltares.nl/thredds/catalog/opendap/rijkswaterstaat/kustlidar/catalog.html
  For Matlab Openearth developed a crawler that can gets the names of all netCDF files that comprise a tile by interpreting the THREDDS catalog.xml

```
>> url = 'http://opendap.deltares.nl/thredds/catalog/opendap/rijkswaterstaat/vaklodingen/catalog.xml'
>> L = opendap_catalog(url)
>> disp(L{1})
ans = http://opendap.deltares.nl/thredds/dodsC/opendap/rijkswaterstaat/vaklodingen/vaklodingenKB138_0706.nc
>> nc_dump(L{1})
```

Time dependency

Most 'GIS'-minded people produce grids without a time dimension, but simply with some kind of time annotation. This does not easily allow to automatic processing of temporal data. Instead, model output, spatially binned remote sensing products and repeated depth sounding products do produce datasets with a clear and explicit time dimension.

- 'GIS' map. Example
  http://opendap.deltares.nl/thredds/catalog/opendap/rijkswaterstaat/vaklodingen/catalog.html
- 'GIS' map with added dummy time dimension. OpenEarth recommends that also 'GIS' maps always insert a dummy time dimension so 'GIS' data can be processed like model data or remote sensing products. Example: single remote sensing map valid at a specific moment in time.
  http://opendap.deltares.nl/thredds/catalog/opendap/knmi/NOAA/mom/1990_mom/5/catalog.html
  For datasets gathered over a longer period of time the CF convention allows to also to specify time bounds (begin and end time) rather than a specific 'snap-shot' time.
- Full spatio-temporal datasets with explicit time dimension. Example:
  http://opendap.deltares.nl/thredds/catalog/opendap/rijkswaterstaat/vaklodingen/catalog.html

Password restrictions

- without: example
  http://opendap.deltares.nl/thredds/catalog/
- with password. Not all client support password protection. All internet browsers do, so you can always browse pass-word protected datasats. The snctools toolbox shipped with OpenEarthTools allows for password encryption also of direct data request from Matlab. A 26 line open source piece of java code is used for this.
  http://matroos.deltares.nl:8080/thredds/