Xcube Grid Layouts

Technical Note  
Draft  
Norman Fomferra, 20.11.2018

## Problem Description

We want to layout data cube grids in terms of coverage and number of grid cells. These grids may be global or sub-grids to a larger parent grid. The requirements are:

* **Easily combine sub-grids into a larger parent grid**. The grid layout shall support sub-grids whose cells boundaries exactly match the boundary of one single cell of the parent grid.
* **Easily combine sub-grids of different spatial resolutions**. The grid layout shall support multiple resolutions in a way that any higher resolution cell will always be fully contained by one single lower resolution cell.
* **Easily provide lower resolution versions of both the parent grid and its sub-grids**. The grid layout shall allow chunking the level grids into smaller tiles, where ideally grid sizes should be divisible by tile sizes without remainder. The grid layout shall allow for a maximum number of resolution levels also for the sub-grids. Each level shall have the same spatial coverage.

## Definitions

Let a parent grid's position and coverage be given by *xmin*, *ymin* and *xmax* = *xmin* + *cx*, *ymax* = *ymin* + *cy* in any geodetic coordinate system. The graticule on which sub-grids shall be aligned is defined by the distances *dx* and *dy*:



We assume to grid's geodetic coordinate system is projected or rectified such that Δ*x* = Δy = const. To layout a suitable parent grid, we want the spatial resolution *r* given in geodetic distance units to be a rational number, so we can use it to subdivide a given global or regional coverage *c* of the parent grid and yield an integral number of grid cells *h*:



We also want to generate multi-resolution pyramids, therefore for any resolution *rz* for pyramid level *z*, we postulate

1. with r0 being the lowest resolution of the pyramid.

The total number of grid cells *hz* at level *z* is then

1. where h0 is the number of grid cells at level z = 0 with lowest resolution.

At lowest level *z*=0 with lowest resolution *r*0, are the smallest possible number of grid cells which are necessarily uneven

The distance *d* between lines of a graticule that aligns all sub-grids is



If we demand the same tiling scheme for all sub-grids using a suitable tile or chunk size *t*, sub-grids must be aligned using a possibly larger distance



where suitable tile sizes are given by

Note that in most cases *d* by (7) will be too large so that aligning small sub-grid regions using this graticule will be too coarse and yield inappropriately positioned and sized sub-regions.

## Algorithm

Given a target resolution *rT* for a sub-grid and a coverage *c* of the parent grid we search for best *rz* that are close to *rT* within a given tolerance Δ*rT*. We also want the level number *z* to be as large as possible (and the number of grid cells on pyramid level *h0* to be as small as possible), that is, we want the distance of graticule lines to be as small as possible. This way, we can adjust sub-grid regions with minimum deviation from the actual desired region. We find best *z* and *rz* by testing all possible integer values *h* in the range of *h* = [*hmin*, *hmax*] that is given by target grid coverage and the target resolution:

We get *h*0 and *z* from *h* because of (4) and then *r*0 from (2) with *h*0.

We want *rz*  - *rT* to be small and *z* to be large. With *r*0 and *z* we can define a cost function, for example

that yield the best values for *r*0 and *z* with *r*0, *h*0 defining the parent grid's layout with respect to *c*.

Now we can align sub-grids using (1) with (6) or (7).

## Tool Implementation

## Example

*rT* = 1/12°

Δ*rT* = 10% *rT*

*c* = 180°

hmin = 1963

hmax = 2400

r

TODO …