

VRM TOOLS

Simplifying the Process of Creating and Using A Variable
Resolution Mesh in CESM

Patrick Callaghan, Paul Ullrich, Colin Zarzycki, Julio Bacmeister,
Peter Lauritzen, Adam Herrington

Erik Kluzek, Jim Edwards, Brian Eaton, et al....

In principle, the process of creating, installing, and using variable resolution grids in CESM is straight forward, just 5 simple steps.

5 Simple Steps:

1. Creating a grid
2. Generate mapping, domain, initial value datasets, etc...
3. Install new grid into CESM
4. Adjusting namelist parameters for stable model runs.
5. Further diagnostics and testing.

The problem was that for each step, there were hang-ups along the way which made the steps not-so-simple.

1. Creating a grid
 - SQuadGen is not difficult to use, but the user has to create an image defining the refinement region on a image editor with an implied relation between pixels and Lat/Lon locations.
 - HOMME had to be continuously ported to the current computing environment.
2. Generate mapping, domain, initial value datasets, etc...
 - What files are needed?, how to prepare them?, how to organize them?.

3. Install Grid into CESM

Installation consists of defining the grid structure and the relevant namelist default values in the CESM configuration files.

Historical Method:

Edit 7-9 XML files to install the grid for use with a given tag or release version.

The set of file that must be edited includes:

1. `$(CESM)/cime/config/cesm/config_grids.xml`
2. `$(CESM)/cime/src/drivers/mct/cime_config/config_component_cesm.xml`
3. `$(CESM)/components/cam/bld/config_files/horiz_grid.xml`
4. `$(CESM)/components/cam/bld/namelist_files/namelist_defaults.xml`
5. `$(CESM)/components/clm/bld/namelist_files/namelist_definition_clm4_5.xml`
6. `$(CESM)/components/clm/bld/namelist_files/namelist_defaults_clm4_5.xml`
7. `$(CESM)/cime/src/drivers/mct/cime_config/config_component.xml`

Note: This installation must be done tag-by-tag in the users sandbox. Once complete users can create new cases for the newly created grid:

```
./create_newcase -case MyCaseName -res neOnp4.MyGridName.ne30x4_mt12 --compset FHIST  
-mach cheyenne -run-unsupported
```

So easy a caveman can do it.



VRM Editor

Actions

VRM Edit Display

Grid Options

Grid Type: CubeSquared

Base Resolution: 30

Longitude Shift: 0.0

Rotate-X: 0.0

Rotate-Y: 0.0

Refinement Options

Refine Type: CUBIT

Refine Level: 0

Smoothing Options

Iterations: 0

NONE Length: 0

Grid Modifications

Tessellations: 0

Sub Cell Resolution: 0

Reverse Grid Orientation

Generate VarMesh

The image shows a software window titled 'VRM Editor'. The main area displays a world map with a grid overlay. The map is centered on the Atlantic Ocean, showing the Americas, Europe, and Africa. The grid lines are spaced at regular intervals. The map is surrounded by a coordinate system with latitude and longitude markers. The software interface includes a menu bar with 'VRM', 'Edit', and 'Display'. Below the menu bar are several sections of options: 'Grid Options', 'Refinement Options', 'Smoothing Options', and 'Grid Modifications'. Each section contains various controls like dropdown menus, spinners, and checkboxes. At the bottom right, there is a 'Generate VarMesh' button.

For users who wish to begin using the tools, there are 2 directories available in my scratch space:

```
/glade/scratch/patc/For_AMWG/  
/glade/scratch/patc/For_AMWG_DEMO/
```

- The first is a 'clean' copy of a directory containing the VRM_tools and a CESM tag (cesm2_2_alpha04c)
- The second is a copy in which I have gone through the process of creating, installing, and running a new grid.

The directories contain:

Models/	← contains a recent tag to use
VRM_Files/	← This is a repository where I store VRM grid data
VRM_NOTES-CESM2_2_ALPHA04C.	← Notes for processing my test grid SAM01_ne30x4
VRM_tools/	← The VRM toolkit

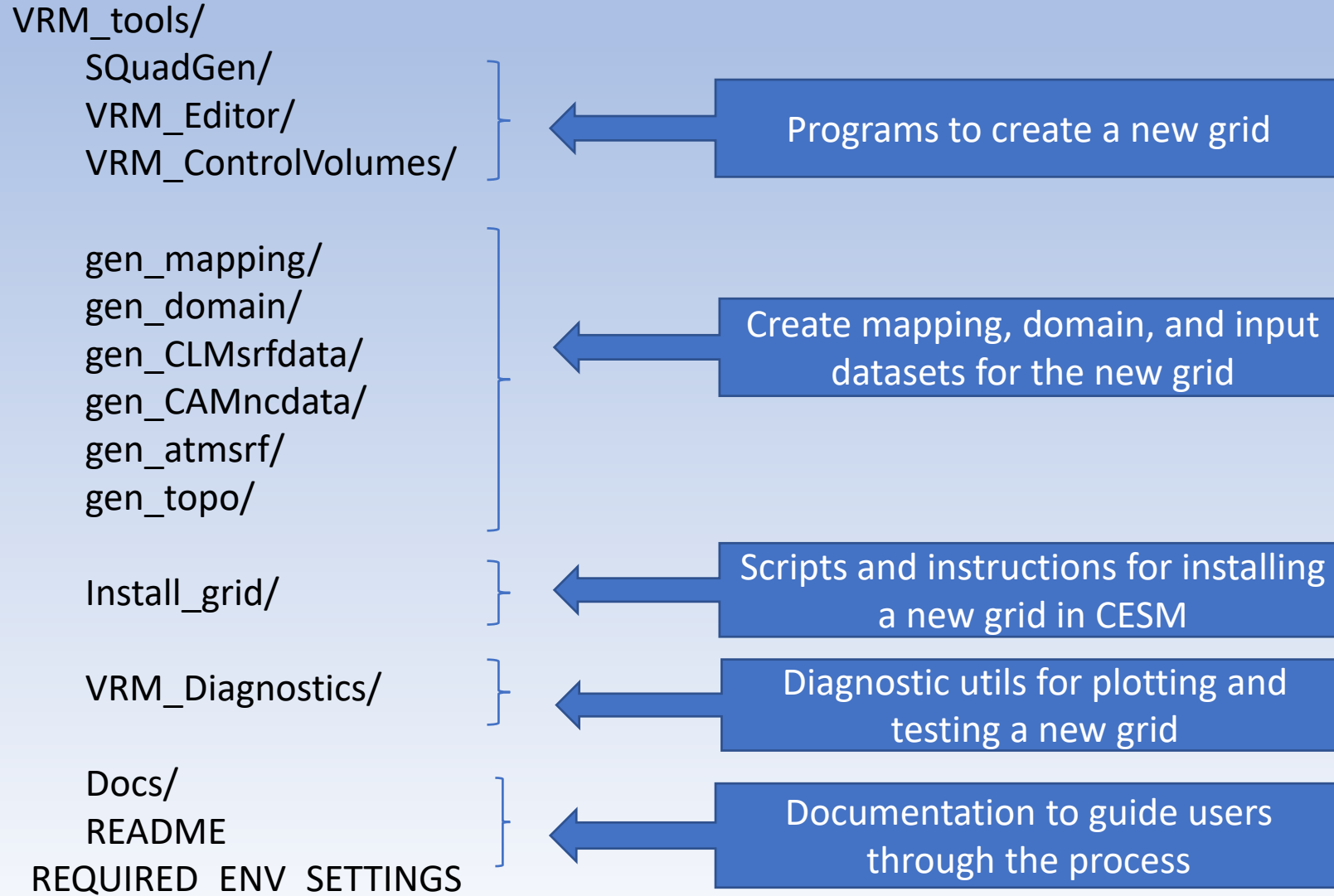
Mac Users can download a VRM Editor app:

```
cd VRM_tools/VRM_Editor/MAC_exe/  
scp Xfer_VRM_ForMac.tar my.mac.edu:~/Desktop
```

The tar file contains VRM_Editor.app that will run on current Mac OS's

Simplifying the Process of Creating and Using A Variable Resolution Mesh in CESM

Overview of the VRM tools directory:



1. Creating a new grid

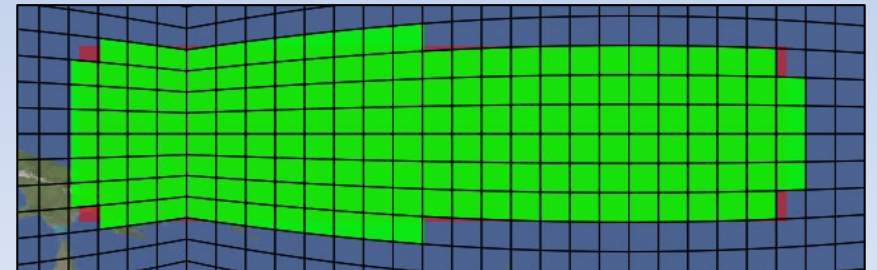
There are currently three programs to create a new grid:

1. **SQuadGen** -- Command line program
2. **VRM_Editor** -- Interactive GUI for SQuadGen
3. **Create_VRMgrid** -- Command line interface for VRM_Editor

These programs create the NE part of the cube-sphere grid and store the results in an EXODUS file.

SQuadGen

- To use SQuadGen, the user first creates a rectangular PNG file with an image editor in which brightness values indicate the region of refinement.
- Given a base resolution and a refinement level, this image is converted into a cube grid of discrete refinement levels.
- The then program uses a specified set of templates to fill in the regions where there is a transition between resolutions

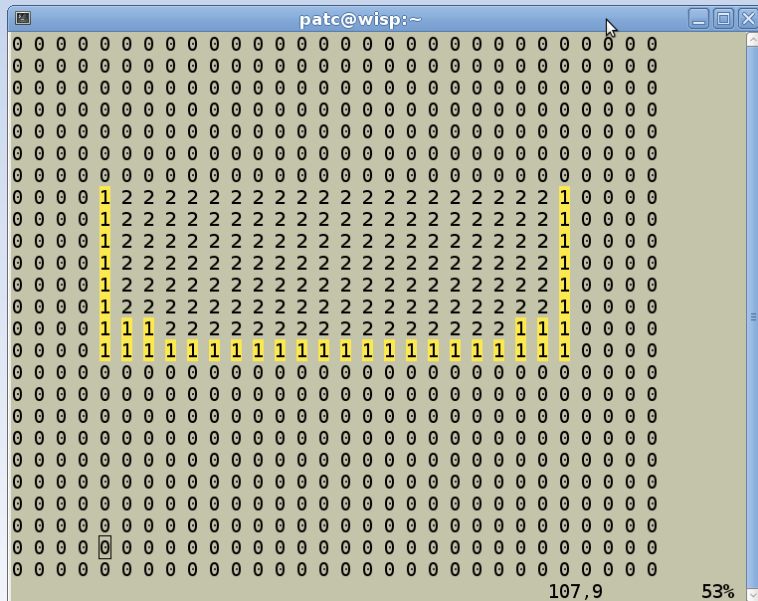


Each element for a NE grid is assigned an integer value indicating the level of refinement to perform.

1. Creating a new grid

SQuadGen

- SQuadGen has command line options to specify the refinement resolutions, template type, smoothing options, and rotation angles which control the resulting grid.
- Running the program with no options will produce a listing of the available options.



```
computer>./SQuadGen
```

Parameters:

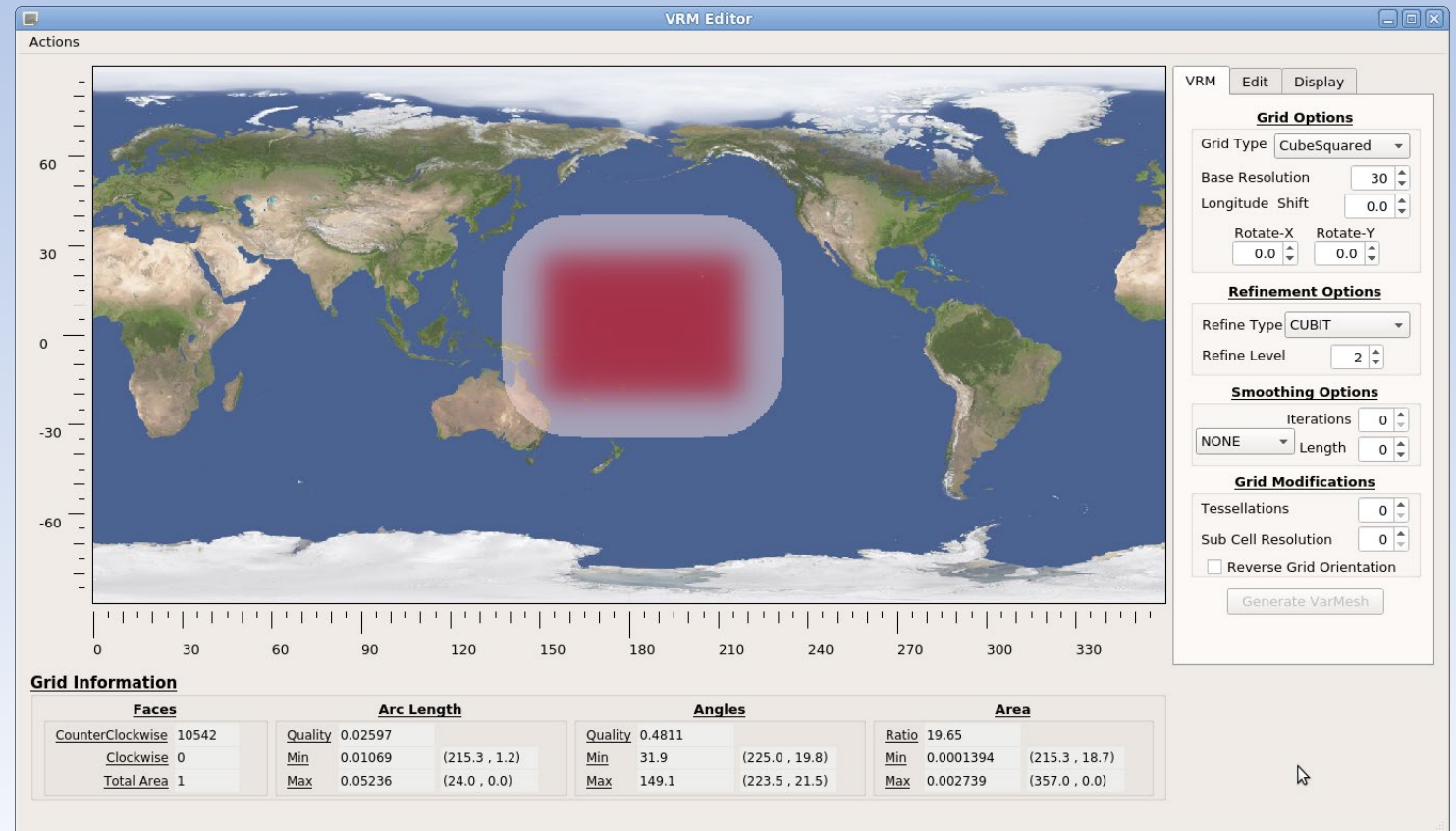
```
--grid_type <string> ["CS"] (Options: ICO | CS)
--refine_type <string> ["LOWCONN"] (Options: LOWCONN | CUBIT | LOWCONNOLD)
--refine_level <integer> [2]
--resolution <integer> [10]
--refine_file <string> ["" ]
--output <string> ["" ]
--loadcsrefinementmap <bool> [false]
--smooth_type <string> ["NONE"] (Options: NONE | SPRING | PRESSURE)
--smooth_dist <integer> [1] (Smooth distance, -1 = smooth entire mesh)
--smooth_iter <integer> [10]
--lon_base <double> [-180.000000]
--lat_base <double> [0.000000]
--x_rotate <double> [0.000000]
--y_rotate <double> [0.000000]
--tessellate <integer> [0]
--subcellres <integer> [0]
--invert <bool> [false]
--block_refine <bool> [false]
```

- The program also writes out the discrete refinement values in a text file that can be edited manually.
- Users can iteratively edit and feed these values back into SQuadGen to tailor the grid appearance as desired.

1. Creating a new grid

VRM Editor

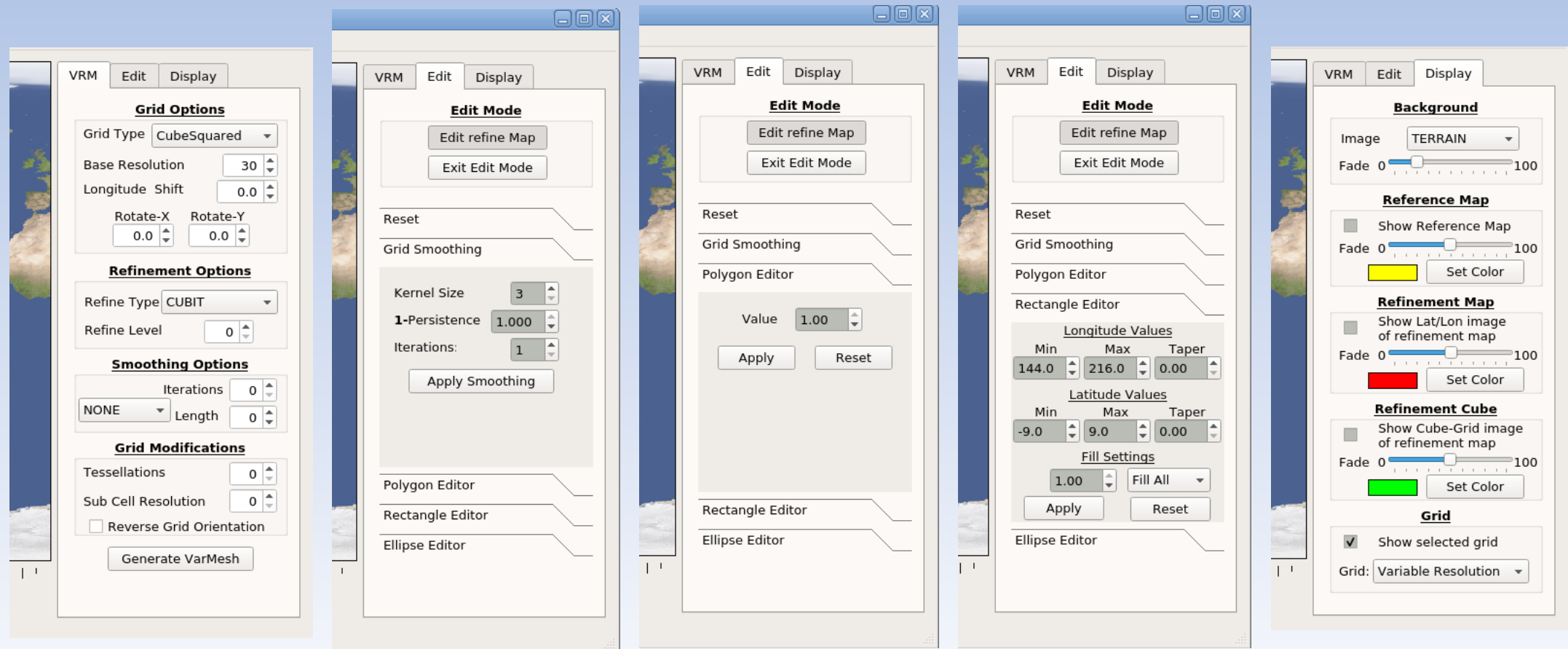
- VRM_Editor is a GUI interface for SQuadGen but it uses a different approach to define refinements.
- Rather than editing an image, the approach is to edit a map of continuous refinement values [0.,1.] on a (Lat,Lon) grid directly. Where values equal to 1, shown in red, define the desired region of refinement.
- These values are stored/read from a netCDF file with a defined (Lat,Lon) grid rather than relying on an *implied* (Lat,Lon) grid for a rectangular image.



1. Creating a new grid

VRM Editor

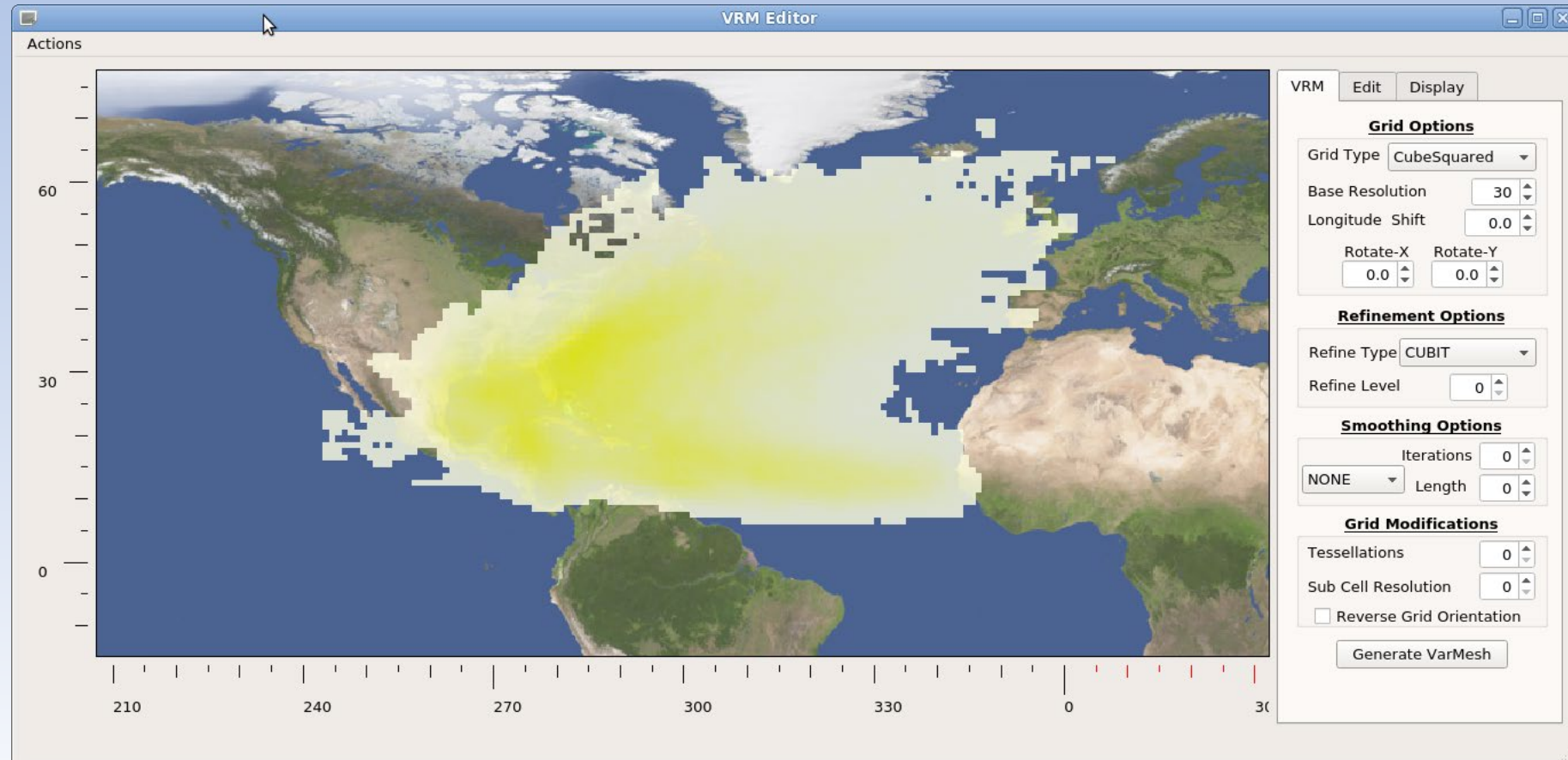
- In addition to all of the SQuadGen options, the editor has tools used to create and modify the refinement map.
- There is also set of options control the appearance of the displayed values in the editor.



1. Creating a new grid

VRM Editor

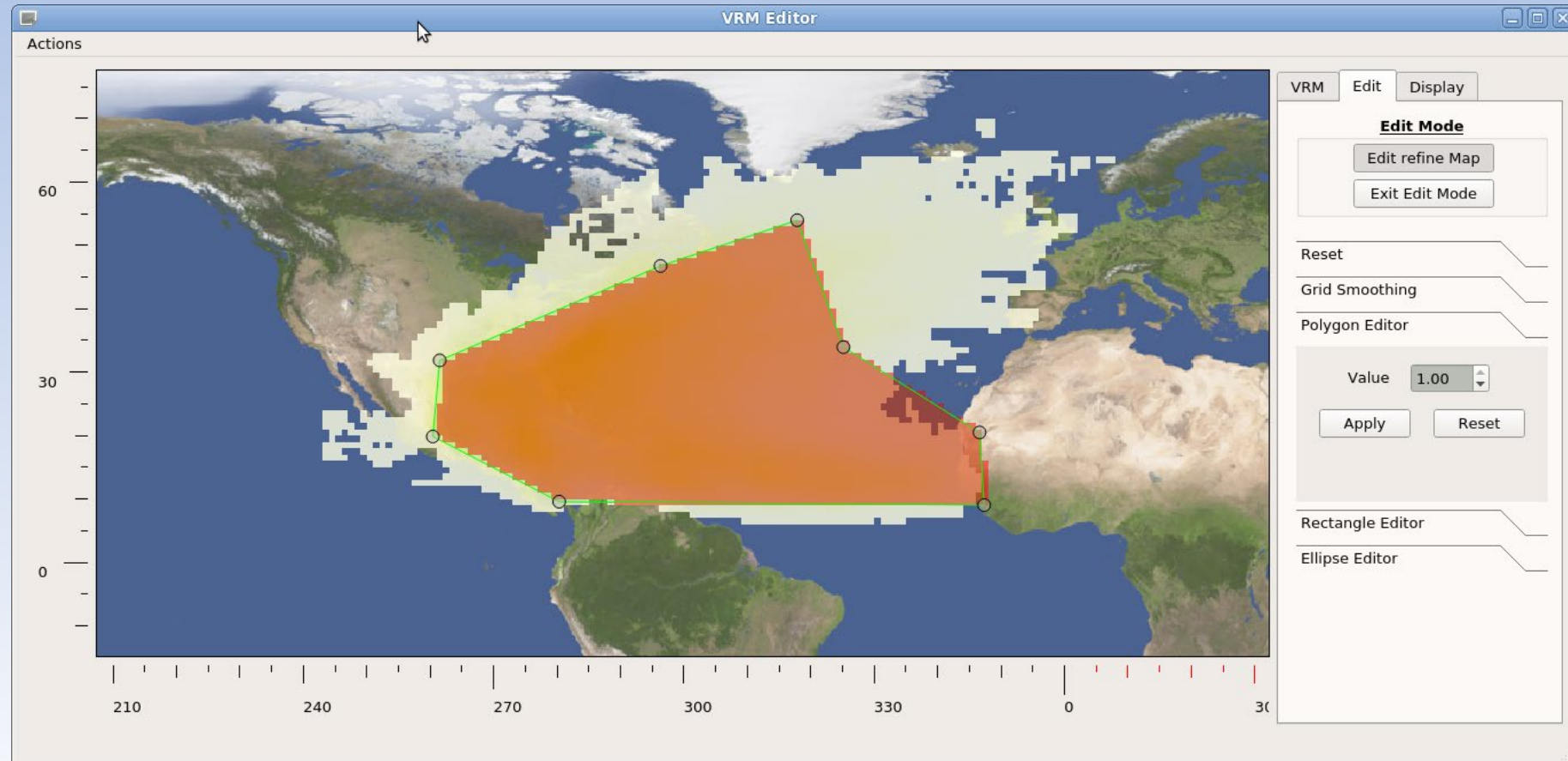
- Optionally, a reference map based on observed phenomena can be read in and used as a guide for creating refined regions.
- Values on a (Lat,Lon) grid are displayed with a yellow color gradient between min and max.
- The example here is a PDF of tropical storm tracks in the Atlantic basin.



1. Creating a new grid

VRM Editor

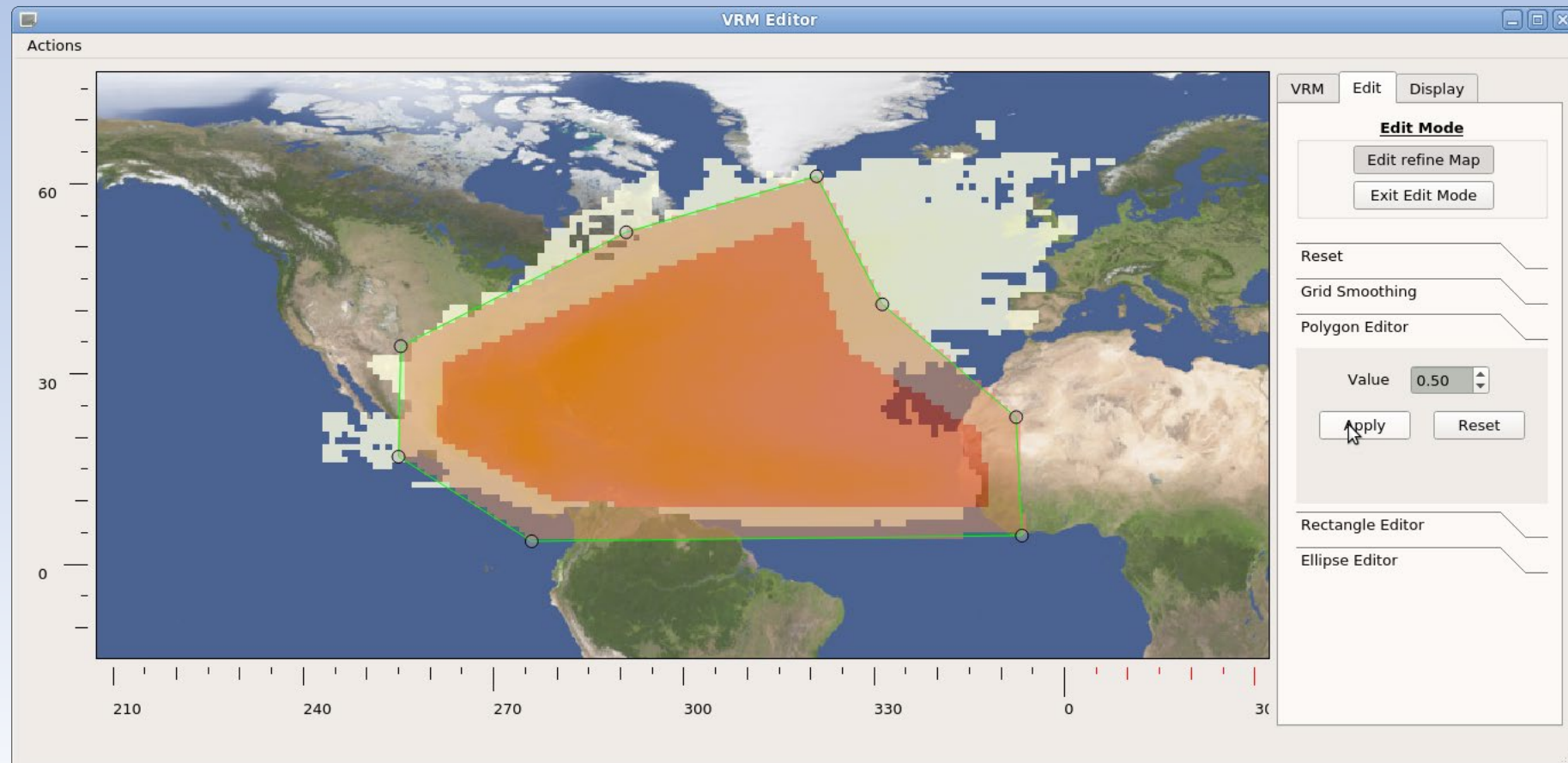
- For a refinement from ne30 to ne120 for example there are 2 refinement levels.
- The polygon editor can be used to first construct a ne120 region which encompasses the desired variability by setting a refinement value equal to 1.



1. Creating a new grid

VRM Editor

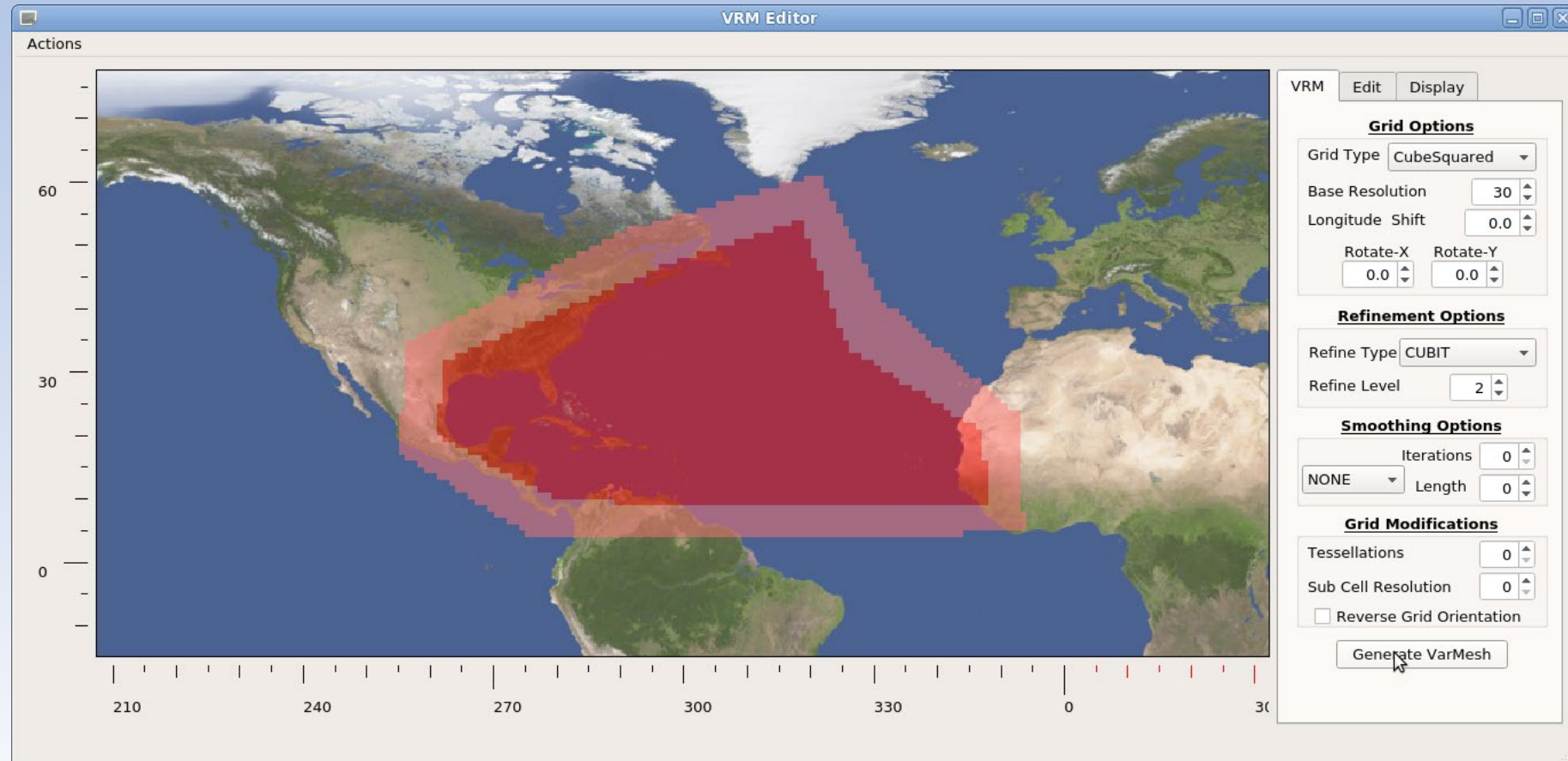
- An expanded perimeter is then added with a value of 0.5 to define a transition region which will correspond to a ne60 grid resolution.



1. Creating a new grid

VRM Editor

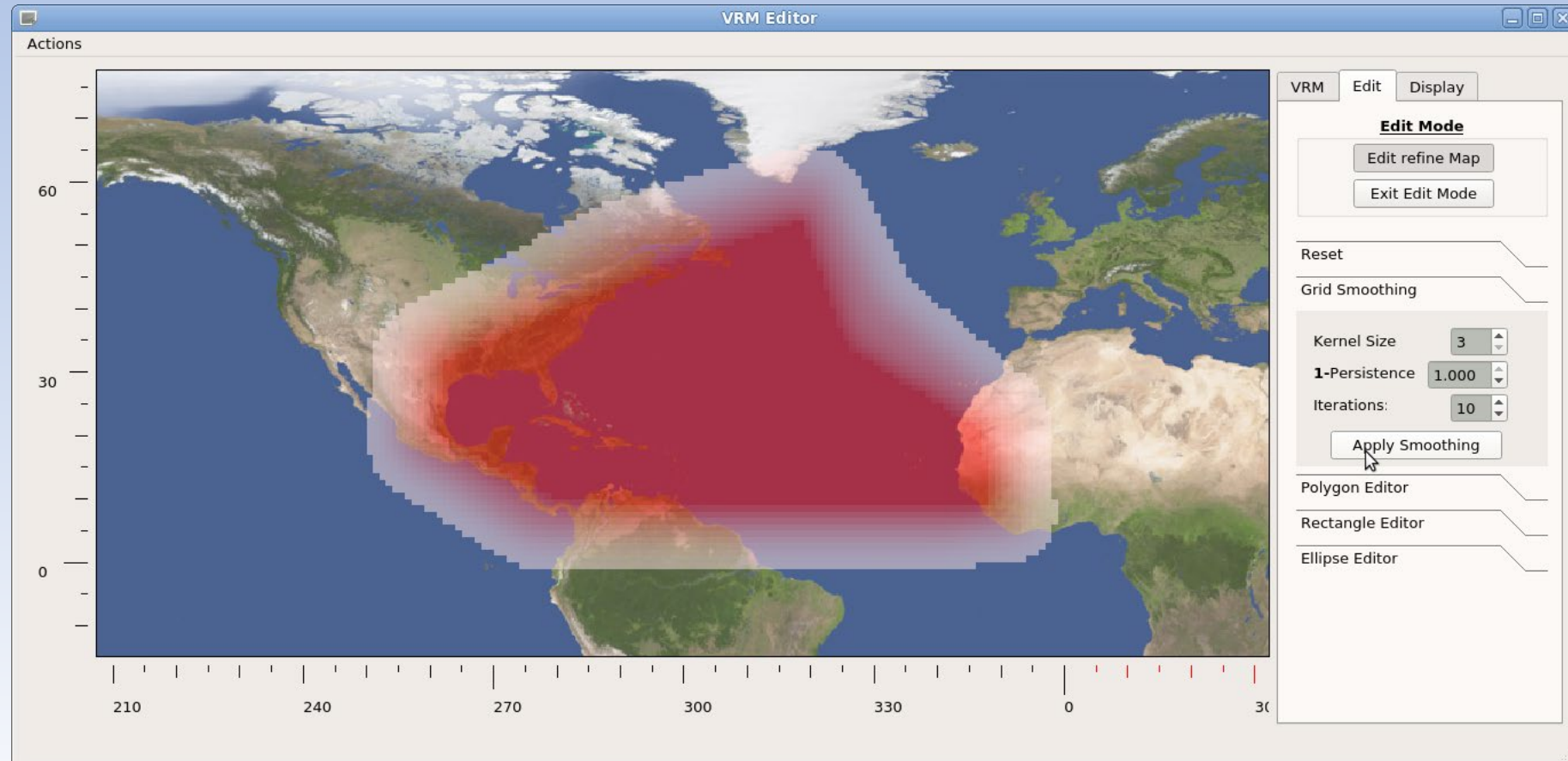
- The resulting refinement map contains incremental changes between values with some sharp vertices.



1. Creating a new grid

VRM Editor

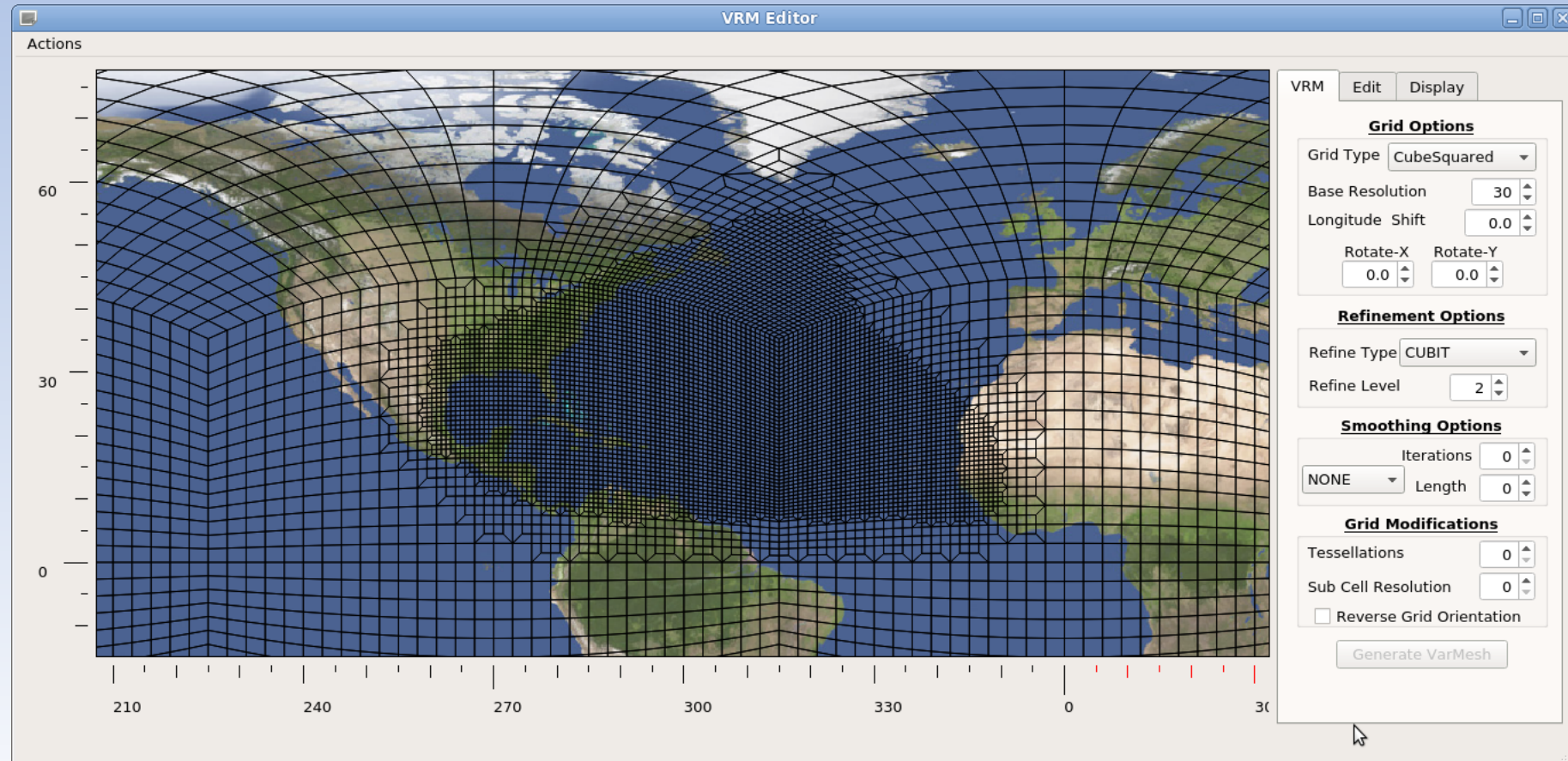
- Using the smoothing tool, these rough edges can be rounded to get a more satisfactory refinement map.



1. Creating a new grid

VRM Editor

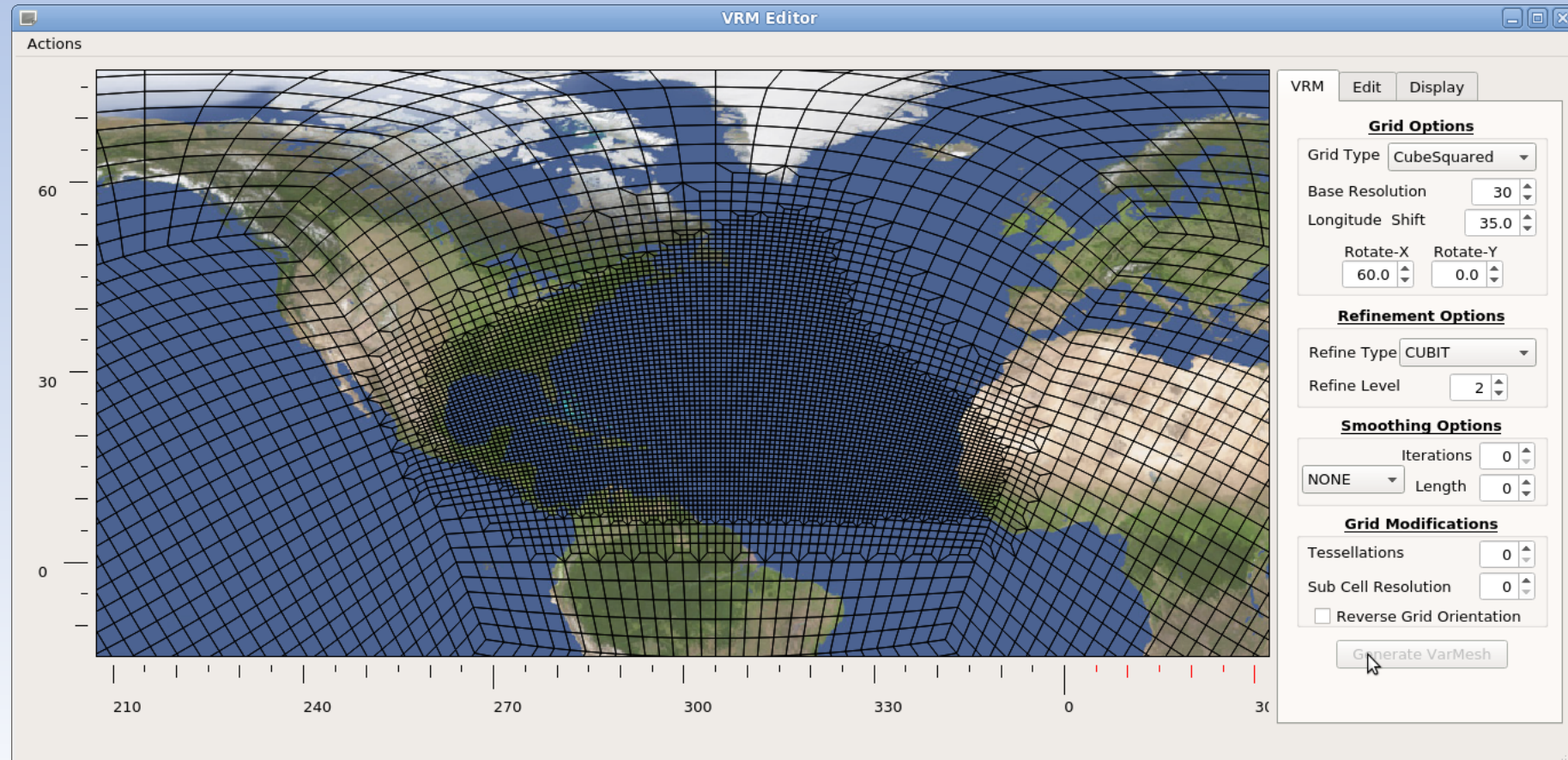
- Once the refinement map changes are accepted, the SQuadGen parameters are set and the resulting refined grid generated.



1. Creating a new grid

VRM Editor

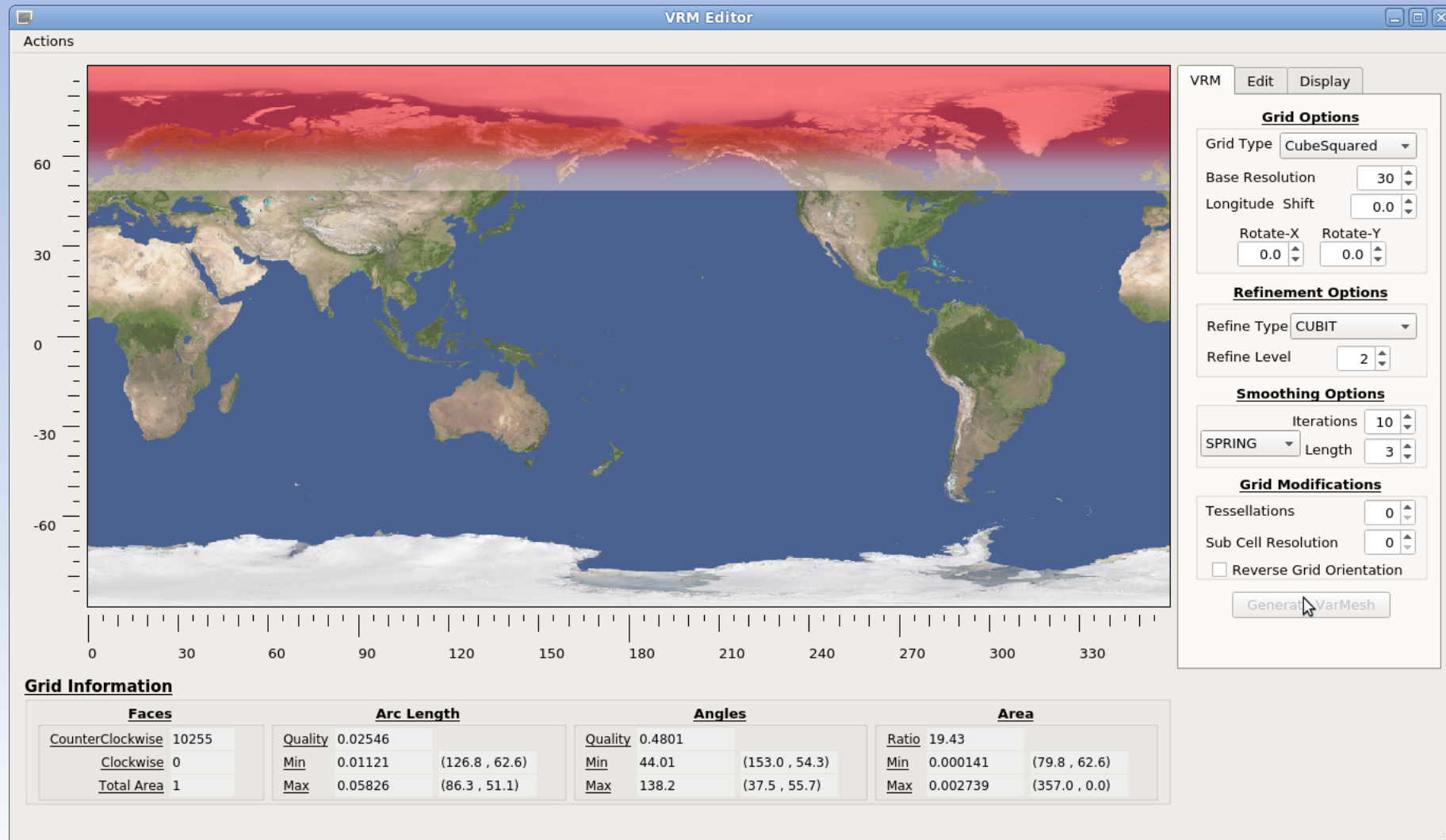
- As a general rule, the refined regions should be centered, as much as possible, on one of the cube faces.
- After adjusting values to rotate the base grid and selecting refinement and smoothing options, the final grid is ready to be saved.



Some Polar Examples

To create an ne30 grid which is refined to ne120 for Latitudes higher than 60N:

- The rectangle editor is used to create a refinement map with value 1 over the pole, which tapers down to 0 at 60N.



The screenshot shows the VRM Editor interface. The main window displays a global map with a red refinement map overlay. The refinement map is a gradient from red (value 1) at the pole to white (value 0) at 60N. The map is overlaid on a grid. The control panel on the right has the following settings:

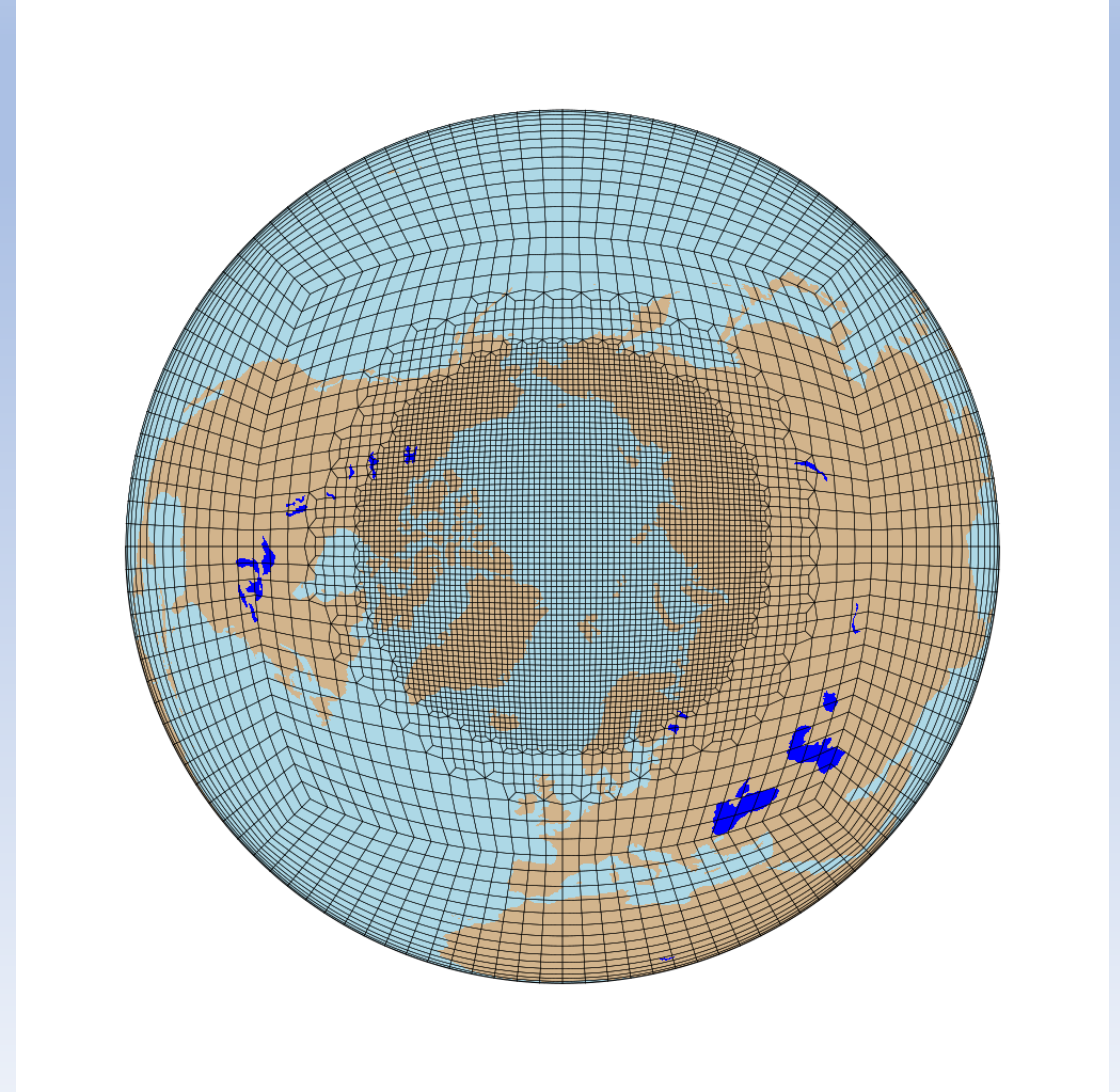
- Grid Options:** Grid Type: CubeSquared, Base Resolution: 30, Longitude Shift: 0.0, Rotate-X: 0.0, Rotate-Y: 0.0
- Refinement Options:** Refine Type: CUBIT, Refine Level: 2
- Smoothing Options:** Iterations: 10, SPRING, Length: 3
- Grid Modifications:** Tessellations: 0, Sub Cell Resolution: 0, Reverse Grid Orientation:

The **Grid Information** panel at the bottom provides the following data:

Faces		Arc Length		Angles		Area	
CounterClockwise	10255	Quality	0.02546	Quality	0.4801	Ratio	19.43
Clockwise	0	Min	0.01121 (126.8, 62.6)	Min	44.01 (153.0, 54.3)	Min	0.000141 (79.8, 62.6)
Total Area	1	Max	0.05826 (86.3, 51.1)	Max	138.2 (37.5, 55.7)	Max	0.002739 (357.0, 0.0)

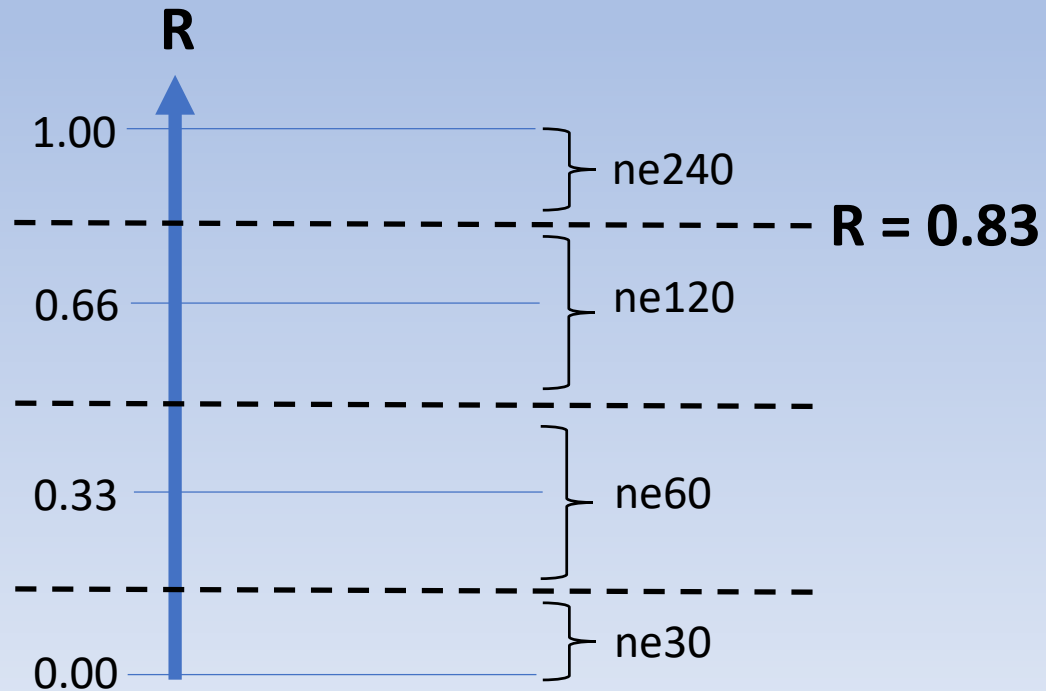
Some Polar Examples

- Setting the SQuadGen parameters for a ne30 base resolution, 2 refinements, and some grid smoothing creates a grid with a circumpolar refinement.



Some Polar Examples

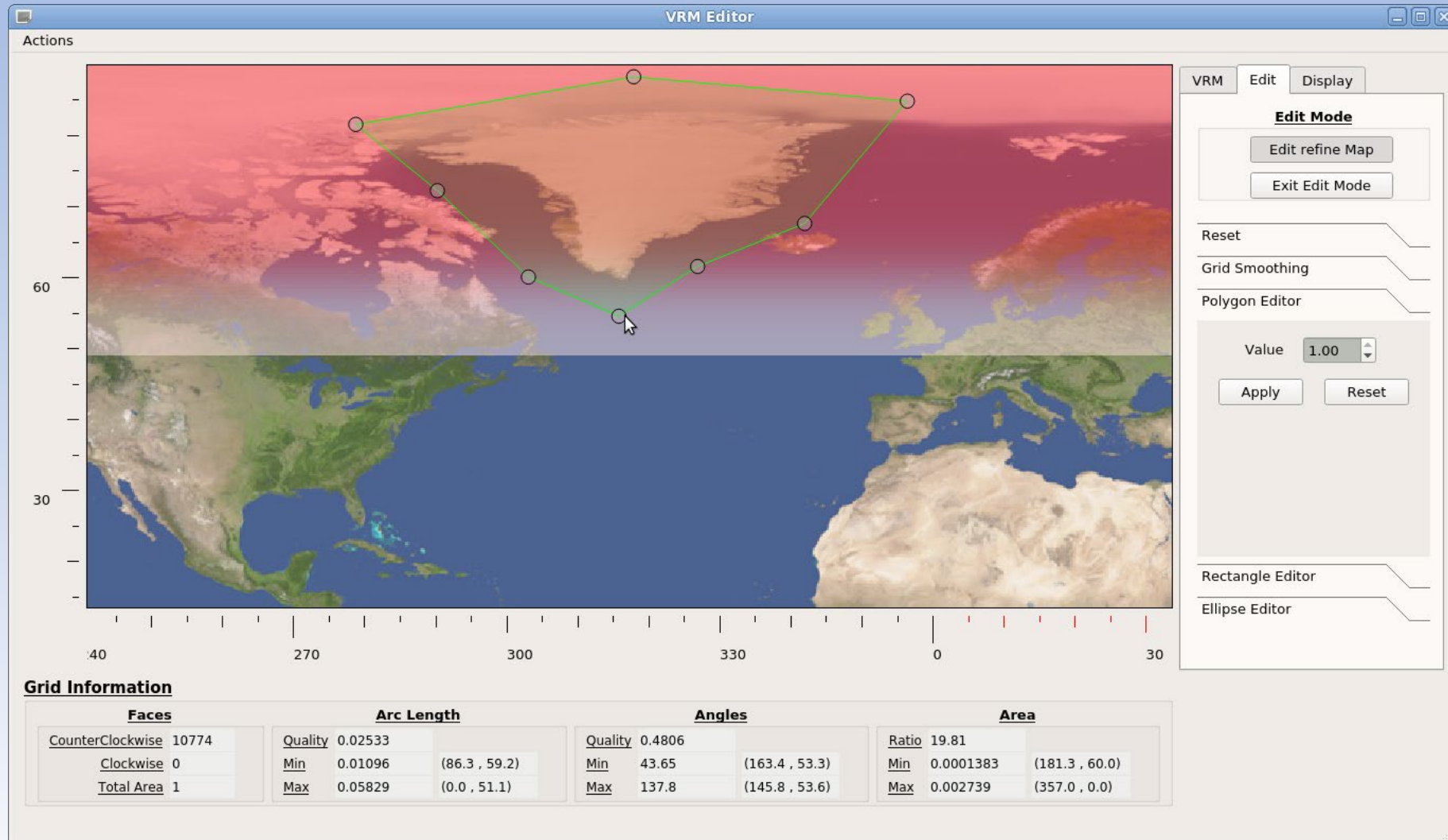
Now Consider a ne30 grid with a ne120 circumpolar refinement with a local refinement to ne240 over Greenland.



- The range of refinement values is discretized and assigned to refinement levels as shown.
- For the circumpolar part of the refinement, the rectangle edit tool is used in the same manner as before, but the maximum value in the rectangle is set to 0.82 (< 0.83).
- For the ne240 region, the polygon editor is then used to add a region where the values are set equal to 1.

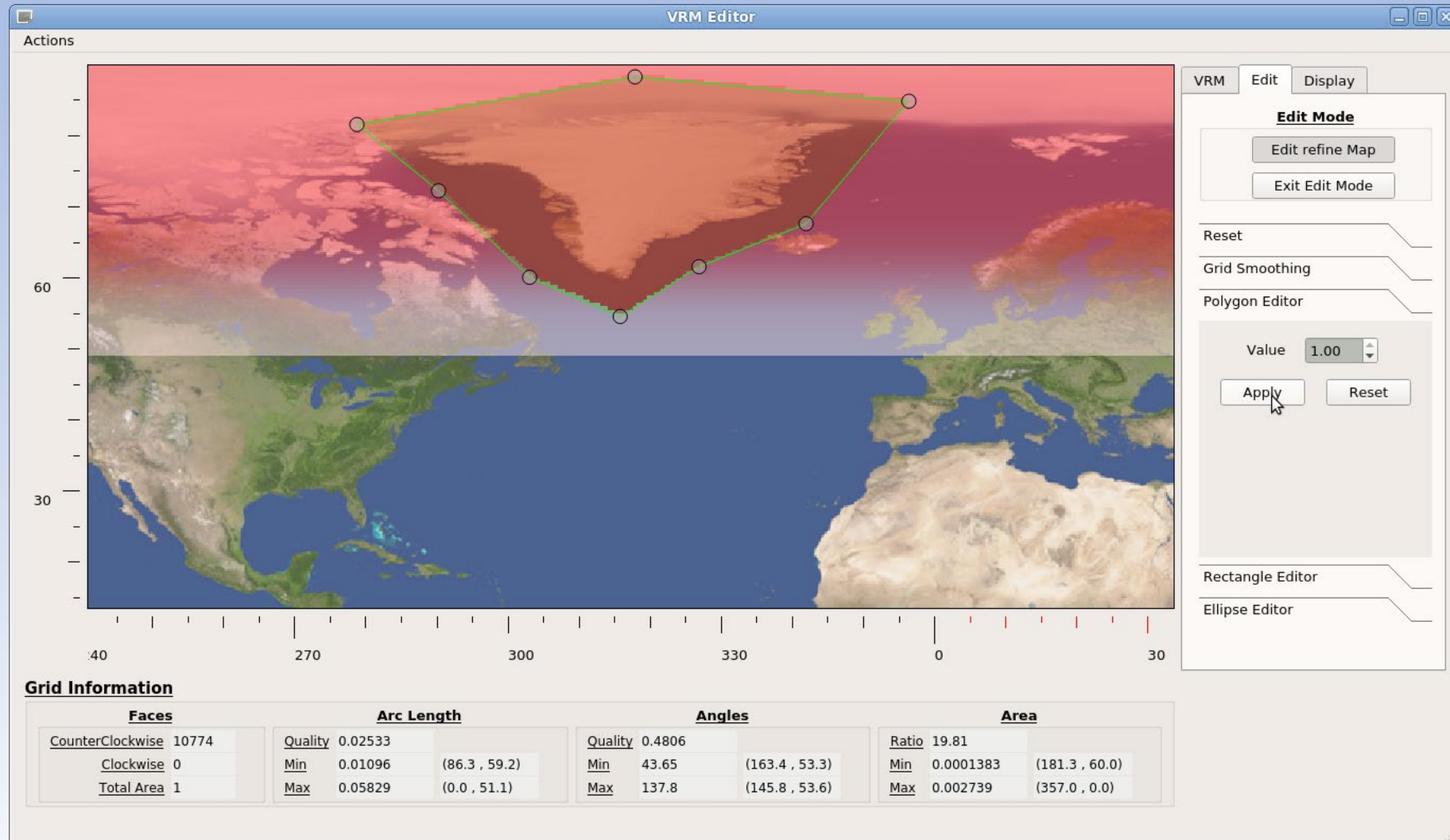
Some Polar Examples

The polygon vertices are moved to create a region which encompasses Greenland.



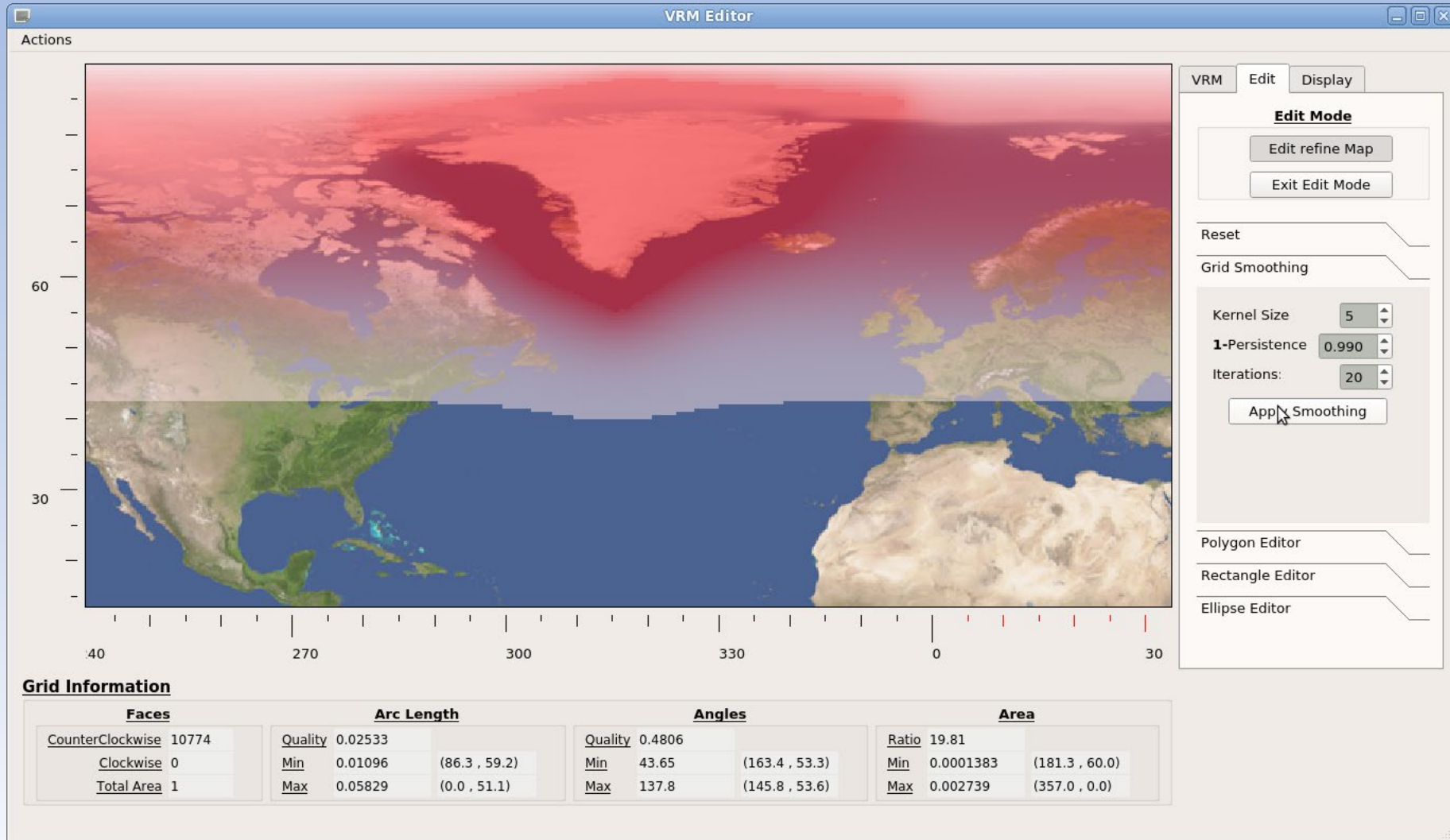
Some Polar Examples

After the polygonal region is added, there is a relatively sharp transition between $R=1$ and $R=0.82$ values.



Some Polar Examples

Apply smoothing to round the sharp edges.



The screenshot shows the VRM Editor software interface. The main window displays a polar map of the Arctic region with a grid overlay. The map is color-coded, with red and purple tones in the northern regions and green and blue tones in the southern regions. The grid lines are visible, and the map is centered on the North Pole. The interface includes a top menu bar with 'VRM', 'Edit', and 'Display' options. Below the menu bar, there are several toolbars and panels. The 'Edit Mode' panel is active, showing buttons for 'Edit refine Map' and 'Exit Edit Mode'. The 'Grid Smoothing' panel is also visible, with settings for 'Kernel Size' (5), '1-Persistence' (0.990), and 'Iterations' (20). An 'Apply Smoothing' button is present in this panel. The 'Polygon Editor', 'Rectangle Editor', and 'Ellipse Editor' panels are also visible. At the bottom of the interface, there is a 'Grid Information' section with a table of statistics.

Faces		Arc Length		Angles		Area	
CounterClockwise	10774	Quality	0.02533	Quality	0.4806	Ratio	19.81
Clockwise	0	Min	0.01096 (86.3, 59.2)	Min	43.65 (163.4, 53.3)	Min	0.0001383 (181.3, 60.0)
Total Area	1	Max	0.05829 (0.0, 51.1)	Max	137.8 (145.8, 53.6)	Max	0.002739 (357.0, 0.0)

Some Polar Examples

Then save the refinement map, set the SQuadGen parameters, and generate the grid

The screenshot displays the VRM Editor software interface. The main window shows a 3D visualization of a grid over a map of the world, with a polar region highlighted. The grid is composed of a dense network of lines, and the map shows continents and oceans. The interface includes a right-hand control panel with various options and a bottom 'Grid Information' panel.

Grid Options

- Grid Type: CubeSquared
- Base Resolution: 30
- Longitude Shift: 0.0
- Rotate-X: 0.0
- Rotate-Y: 0.0

Refinement Options

- Refine Type: CUBIT
- Refine Level: 3

Smoothing Options

- Iterations: 10
- SPRING Length: 3

Grid Modifications

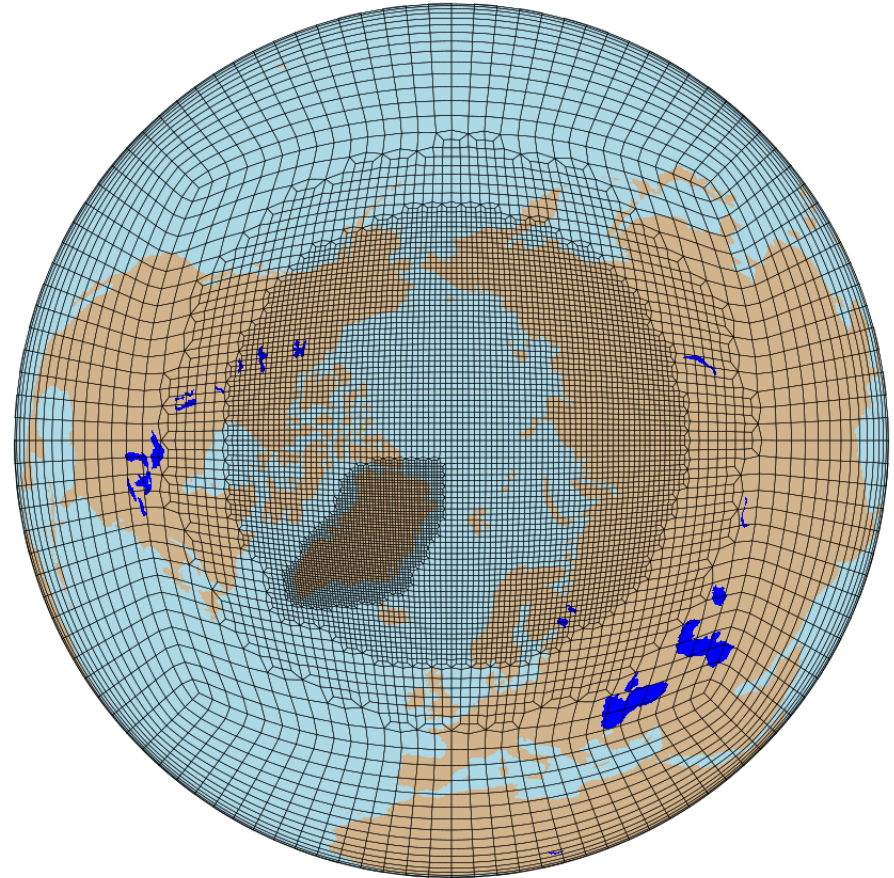
- Tessellations: 0
- Sub Cell Resolution: 0
- Reverse Grid Orientation

Grid Information

Faces		Arc Length		Angles		Area	
CounterClockwise	14073	Quality	0.02439	Quality	0.486	Ratio	80.53
Clockwise	0	Min	0.005436 (317.4, 55.3)	Min	41.34 (324.5, 48.7)	Min	3.401e-5 (317.4, 55.3)
Total Area	1	Max	0.05827 (360.0, 48.1)	Max	140.9 (318.8, 55.5)	Max	0.002739 (357.0, 0.0)

Some Polar Examples

- The final result is then a circumpolar refinement to ne120 over the North pole with a local refinement to ne240 over Greenland.



1. Creating a new grid

Create VRMgrid

- Create_VRMgrid is a command line interface that works very much like SQuadGen.
- Rather than providing a PNG image, the user provides a netCDF file containing the refinement map values.
- This file can be an output from the VRM_Editor, or a file created by some other means.

```
computer>./Create_VRMgrid
```

Input Parameters:

```
--refine_type <string> ["CUBIT"] (Options: CUBIT | LOWCONN | LOWCONNold)
--grid_type <string> ["CubeSquared"] (Options: Icosahedral | CubeSquared)
--smooth_type <string> ["NONE"] (Options: NONE | SPRING | PRESSURE)
--resolution <integer> [30]
--refine_level <integer> [0]
--tessellate <integer> [0]
--subcells <integer> [0]
--smooth_dist <integer> [0]
--smooth_iter <integer> [0]
--reverse_orient <bool> [false]
--x_rotate <double> [0.000000]
--y_rotate <double> [0.000000]
--lon_shift <double> [0.000000]
--refine_file <string> ["NULL"]
--output <string> ["NULL"]
```

Once the EXODUS file is created, transfer it to Cheyenne to create a repository:

Create \$(For_AMWG)/VRM_Files/**ne0np4.SAM01.ne30x4/**

Create \$(For_AMWG)/VRM_Files/**ne0np4.SAM01.ne30x4/grids/** ← **NOTE the naming conventions.**

Copy **SAM01_ne30x4_EXODUS.nc** to the grids/ directory.

VRM_NOTES-CESM2_2_ALPHA04C

```
=====
(1) Check out a recent CESM tag:
=====
cheyenne% mkdir cesm2_2_alpha04c
cheyenne% git clone https://github.com/escomp/cesm.git cesm2_2_alpha04c/
cheyenne% cd cesm2_2_alpha04c/
cheyenne% vi Externals.cfg <--- Verify the the cam tag is cam6_2_010 or higher.
                                (cime tag needs to be higher than cime.XXXXX)
cheyenne% ./manage_externals/checkout_externals

=====
(2) Get VRM_tools
=====
cheyenne% cp -r VRM_tools /glade/scratch/For_AMWG/

=====
(3) Build executables
=====
-----
Gen_ControlVolumes.exe
-----
cheyenne% cd VRM_tools/VRM_ControlVolumes/src/
cheyenne% module list
Currently Loaded Modules:
  1) ncenv/1.3  2) intel/18.0.5  3) ncarcompilers/0.5.0  4) mpt/2.19  5) netcdf/4.7.3
cheyenne% module load gnu
cheyenne% make
cheyenne% module load intel

-----
mksurfddata_map
-----
cheyenne% cd $(CESM)/components/clm/tools/mksurfddata_map/src
cheyenne% module load ncl
cheyenne% setenv LIB_NETCDF $NCAR_LDFLAGS_NETCDF
```

- The file contains the step-by-step notes from my installation of the SAM01_ne30x4 grid.
- The first 3 step are preliminaries for getting a CESM tag and building the executables that are needed for this case.

VRM_NOTES-CESM2_2_ALPHA04C

----- mkmapdata fix -----

```
cheyenne% cd ../../mkmapdata/  
cheyenne% cp mkmapdata.sh mkmapdata.sh-ORIG  
cheyenne% vi mkmapdata.sh  
>  
> module load esmf_libs/$esmfvers  
> module load intel/$irntelvers  
>#PFC FIX module load ncl <--- comment out this line  
> module load ncl/6.6.2 <--- add 6.6.2 so this script will work  
> module load nco  
>
```

----- gen_domains -----

```
cheyenne% cd $(CESM)/cime/tools/mapping/gen_domain_files/src  
cheyenne% ../../configure --macros-format Makefile --mpilib mpi-serial  
cheyenne% (source ./env_mach_specific.csh ; gmake)
```

----- interpic -----

```
cheyenne% module load ncl  
cheyenne% setenv LIB_NETCDF $NCAR_LDFLAGS_NETCDF  
cheyenne% setenv INC_NETCDF $NCAR_INC_NETCDF  
cheyenne% cd $(CESM)/components/cam/tools/interpic_new  
cheyenne% gmake
```

- For the developmental tags, there will still be problems that occur in the process.
- Typically when building the executables.
- Don't know why the mkmapdata script won't work unless the specific version is given, just know that this is what it took to get the processing script to run.
- Other problems typically are a result of not having the correct modules loaded.

VRM_NOTES-CESM2_2_ALPHA04C

```
=====  
(4) Create REPO for a new grid  
=====
```

```
cheyenne% mkdir /glade/scratch/patc/For_AMWG_DEMO/VRM_Files/ne0np4.SAM01.ne30x4  
cheyenne% mkdir /glade/scratch/patc/For_AMWG_DEMO/VRM_Files/ne0np4.SAM01.ne30x4/grids  
cheyenne% cp SAM01_ne30x4_EXODUS.nc  
                /glade/scratch/patc/For_AMWG_DEMO/VRM_Files/ne0np4.SAM01.ne30x4/grids/
```

```
=====  
(5) Generate Control Volumes  
=====
```

```
cheyenne% cd $(VRM_tools)/VRM_ControlVolumes/src  
cheyenne% vi input.nl  
>  
> &input_nl  
> GridPath = '/glade/scratch/patc/For_AMWG_DEMO/VRM_Files/ne0np4.SAM01.ne30x4/grids/'  
> GridName = 'SAM01_ne30x4'  
> /
```

```
cheyenne% ./Gen_ControlVolumes.exe input.nl  
cheyenne% ls /glade/scratch/patc/For_AMWG_DEMO/VRM_Files/ne0np4.SAM01.ne30x4/grids/  
  
    SAM01_ne30x4_EXODUS.nc SAM01_ne30x4_np4_LATLON.nc SAM01_ne30x4_np4_SCRIP.nc
```

Create a REPO Directory

- Create a directory ne0np4.SAM01.ne30x4
Where all grid data will be stored and maintained.
- Add your EXODUS file containing the NE part of the grid as a starting point.

Gen_ControlVolumes

- The process of generating the grid is completed by running this program.
- It adds the (NPxNP) GLL grid points for each element and stores the final grid description in SCRIP/LATLON files.
- When completed, the user has 3 separate files describing the grid.

VRM_NOTES-CESM2_2_ALPHA04C

```
=====  
(6) Generate Datasets.  
=====
```

```
-----  
set environment variables  
-----
```

```
cheyyenne% setenv VRM_CESM_TAG "/glade/scratch/patc/For_AMWG_DEMO/Models/cesm2_2_alpha04c/"  
cheyyenne% setenv VRM_REPO_PATH "/glade/scratch/patc/For_AMWG_DEMO/VRM_Files/"  
cheyyenne% setenv VRM_GRID_TAG "SAM01_ne30x4"  
cheyyenne% setenv VRM_GRID_NAME "ne0np4.SAM01.ne30x4"  
cheyyenne% setenv VRM_GRID_NCOL "113060"  
cheyyenne% setenv VRM_DATE "200309"
```

```
-----  
gen_mapping  
-----
```

```
cheyyenne% cd $(VRM_tools)/gen_mapping  
cheyyenne% ./MAKE_genMapping_scripts.csh  
cheyyenne% cd ne0np4.SAM01.ne30x4  
cheyyenne% module load ncl  
cheyyenne% qcmd -- 'sh gen_mapping_SAM01_ne30x4_01.sh >& LOG_SAM01_ne30x4_01' (~3 mins to run)  
cheyyenne% qcmd -- 'sh gen_mapping_SAM01_ne30x4_02.sh >& LOG_SAM01_ne30x4_02' (~4 mins to run)  
cheyyenne% qcmd -- 'sh gen_mapping_SAM01_ne30x4_03.sh >& LOG_SAM01_ne30x4_03' (~25 mins to run)
```

```
-----  
gen_domain  
-----
```

```
cheyyenne% cd $(VRM_tools)/gen_domain  
cheyyenne% ./MAKE_genDomains_scripts.csh  
cheyyenne% cd ne0np4.SAM01.ne30x4  
cheyyenne% qcmd -- 'sh genDomains_SAM01_ne30x4_gx1v7.sh >& LOG_SAM01_ne30x4_gx1v7' (~1 min to run)  
cheyyenne% qcmd -- 'sh genDomains_SAM01_ne30x4_tx0.1v2.sh >& LOG_SAM01_ne30x4_tx0.1v2' (~7 min to run)
```

Generate Datasets

- The user begins by setting some grid-dependent environment variables
- After that, the process is the same for each of the gen_XXXX directories:
 1. cd to the gen_XXXX directory
 2. Run a script which generates a grid directory containing the processing scripts that will generate the needed datasets.
 3. Verify the settings in each script and then run it.
 4. Check the output LOG file to verify that the program did not have any errors.

VRM_NOTES-CESM2_2_ALPHA04C

gen_CLMsrfdata

```
cheyyenne% cd $(VRM_tools)/gen_CLMsrfdata
cheyyenne% ./MAKE_genCLMsrfdata_script.csh
cheyyenne% cd ne0np4.SAM01.ne30x4
cheyyenne% vi genCLMsrfdata_SAM01_ne30x4.sh
> ##=====
> #PBS -N sub_genfsurdat_SAM01_ne30x4
> #PBS -A P93300642 <---- set the project number here!
> #PBS -l walltime=3:59:00
> #PBS -q premium
> #PBS -j oe
> #PBS -l select=4:ncpus=2:mpiprocs=2:mem=109GB
> #####
cheyyenne% qsub < genCLMsrfdata_SAM01_ne30x4.sh ( ~1 hour 10 min to run)
```

gen_CAMncdata

```
cheyyenne% cd $(VRM_tools)/gen_CAMncdata
cheyyenne% ./MAKE_interpic_script.csh
cheyyenne% cd ne0np4.SAM01.ne30x4
cheyyenne% module load ncl
cheyyenne% module load nco
cheyyenne% qcmd -- 'sh interpic_script_SAM01_ne30x4.sh >& LOG_SAM01_ne30x4' ( ~1 min to run)
```

Generate Datasets

- The CLM files take the longest to generate and the job must be submitted rather than run via qcmd.
- Generating true topography also takes a while to run. (I am using the quick way for now).

gen_atmsrf

```
cheyyenne% cd $(VRM_tools)/gen_atmsrf
cheyyenne% ./MAKE_genAtmsrf_script.csh
cheyyenne% cd ne0np4.SAM01.ne30x4
cheyyenne% module load ncl
cheyyenne% qcmd -- 'ncl gen_atmsrf_SAM01_ne30x4.ncl >& LOG_SAM01_ne30x4' ( ~1 min to run)
```

gen_topo

```
cheyyenne% cd $(VRM_tools)/gen_topo/Regrid_topo
cheyyenne% ./MAKE_RegridTopo_script.csh
cheyyenne% cd ne0np4.SAM01.ne30x4
cheyyenne% module load ncl
cheyyenne% qcmd -- 'ncl Regrid_topo_SAM01_ne30x4.ncl >& LOG_SAM01_ne30x4' ( ~1 min to run)
```

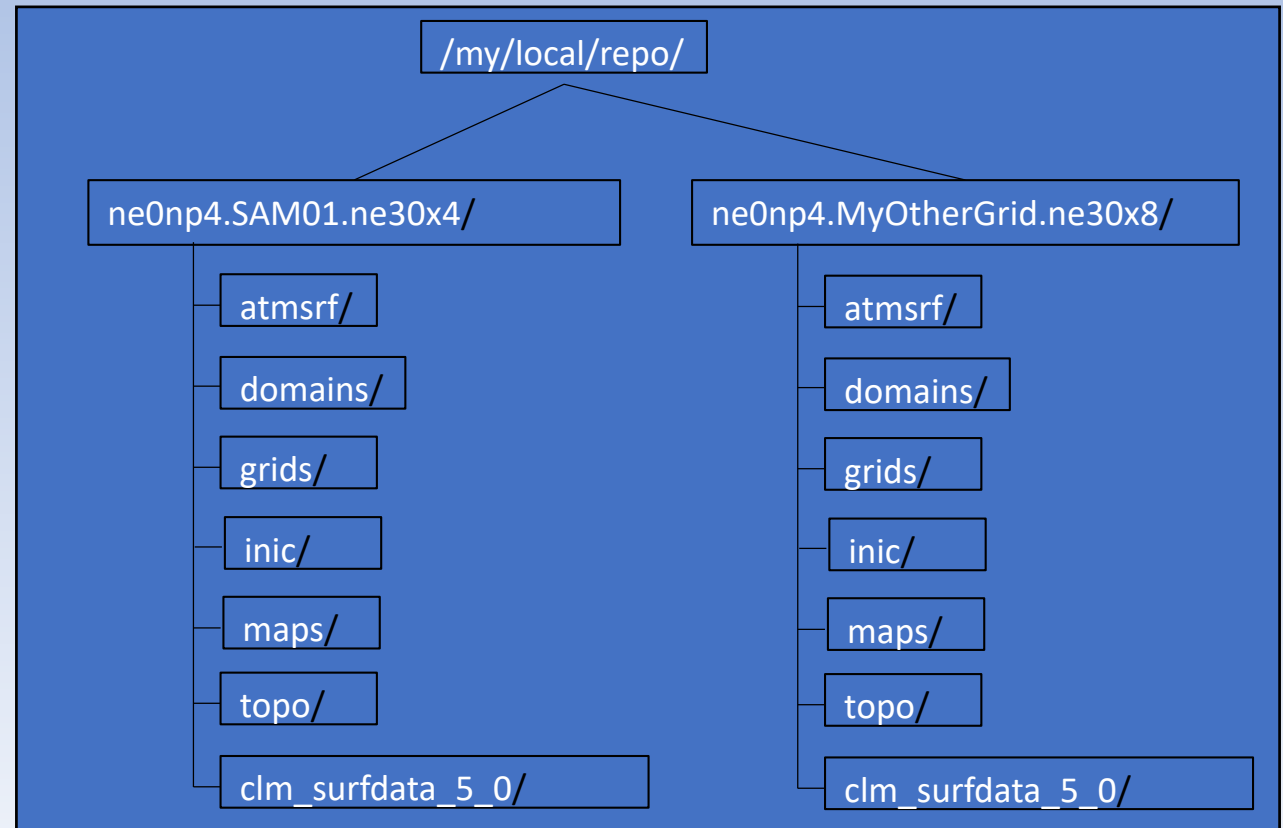

2. Generate Data

The data processing directories:

gen_mapping/
gen_domain/
gen_CLMsrfdata/
gen_CAMncdata/
gen_atmsrf/
gen_topo/

- Contain shell scripts, NCL, and FORTRAN programs which create all of the files needed by CESM.

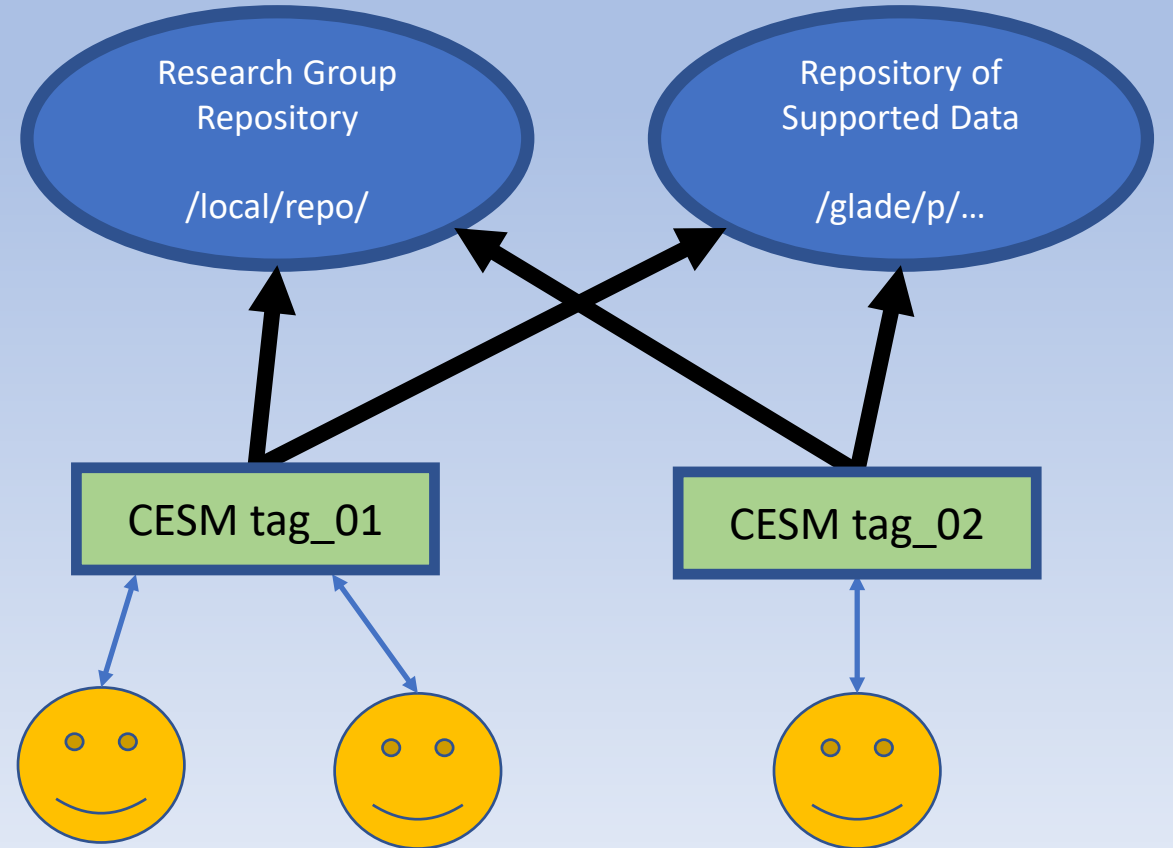
- When completed, the user has generated a repository of required values for each grid, stored with the following directory structure:



2. Generate Data

- The usage scenario envisioned for the next release is that a research group will have a common repository of grid information.
- This \$(REPO) will contain the VRM grids relevant to their research, which supplements the supported grid resolutions in CESM.
- A local repository of grids that can be readily shared by members of a research group using *compatible* CESM tags.
- The data in \$(REPO) will need to be updated periodically to keep up with CESM development, but not for every tag.

(Tags earlier than the most recent require a different installation process that is not as flexible as this.)



3. Install Grid into CESM

```
=====  
(7) Installation  
=====
```

```
cheyenne% cd $(VRM_tools)/Install_grid/NEWTAGS  
cheyenne% ./MAKE_install_script.csh  
cheyenne% vi Install_grid_SAM01_ne30x4.csh  
> set GRID_TAG = "SAM01_ne30x4"  
> set GRID_NAME = "ne0np4.SAM01.ne30x4"  
> set GRIDDESC = "An ne30np4 grid with a ne120np4 refinement over South America"  
>  
> set REPO_PATH = "/glade/scratch/patc/For_AMWG_DEMO/VRM_Files/"  
> set DATE = "200309"  
> set NCOL = "113060"  
cheyenne% chmod 744 Install_grid_SAM01_ne30x4.csh  
cheyenne% ./Install_grid_SAM01_ne30x4.csh
```

Install The New Grid

- In the same manner as generating datasets, the user just has to run a MAKE script.
- The edit that scripts to add a grid description for their new grid.
- Then run it.

This installation script adds 4 files to the \$(REPO) directory:

<code>config_grids.xml</code>	contains the grid description for the file
<code>shell_commands</code>	Sets the pelayout and the model timestep
<code>user_nl_clm</code>	Contains the landuse datasets.
<code>user_nl_cam</code>	Contains the &dyn_se_inparm namelist values

Adding these files completes the installation!

The only updates that should be needed for the repository are if there is are model updates to the XML file formats, the namelist values, or to the source input datasets for newer tags.

4. Adjusting Parameter for a stable 5 day run

Create a new Case

The user just need to add a `-user-mods-dir` argument to the `newcase` command to use the grid.

```
./create_newcase -case MyCaseName -res ne0np4.MyGridName.ne30x4_mt12 --compset FHIST -  
mach cheyenne -run-unsupported -user-mods-dir $(MyRepoPath)/ne0np4.MyGridName.ne30x4
```

In the case directory, the `user_nl_cam` has the defaults settings. These must be adjusted iteratively until the model runs stably.

For ne120 refinements, the values for a uniform ne120np4 grid will likely work unless the refinement created a small element in the transition region.

Higher resolution runs require a bit more effort. Once the values are established for a stable run, then they should be used to update the defaults in the `$(REPO)/user_nl_cam` file.

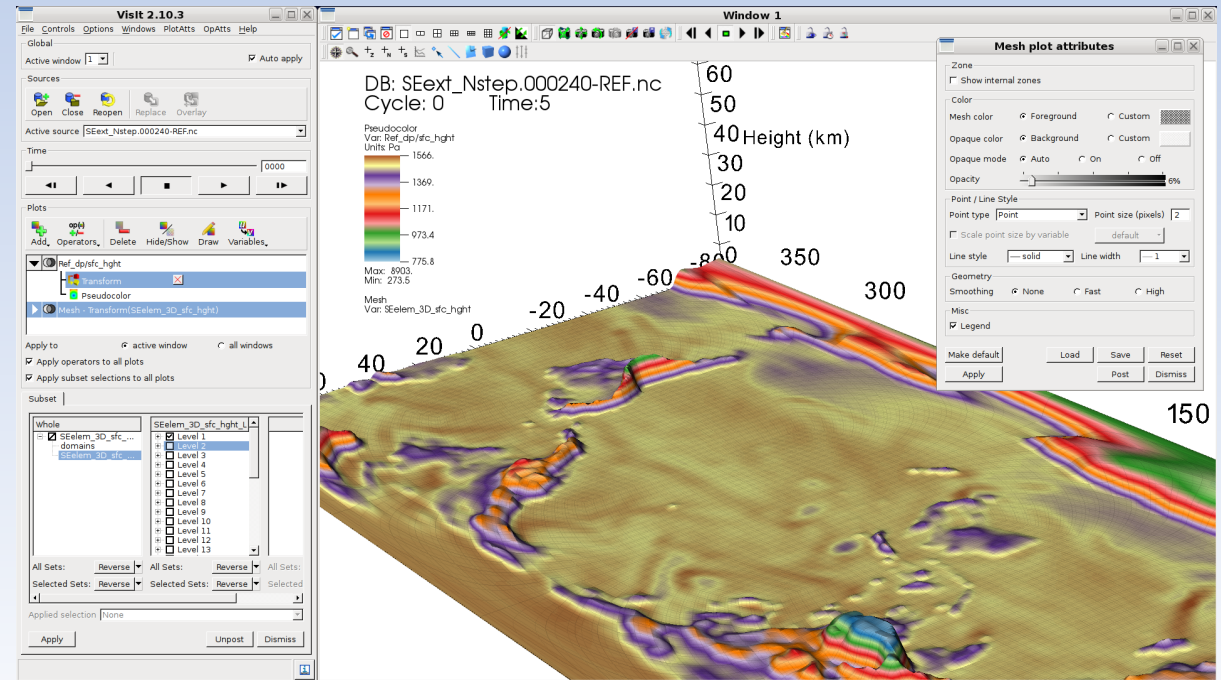
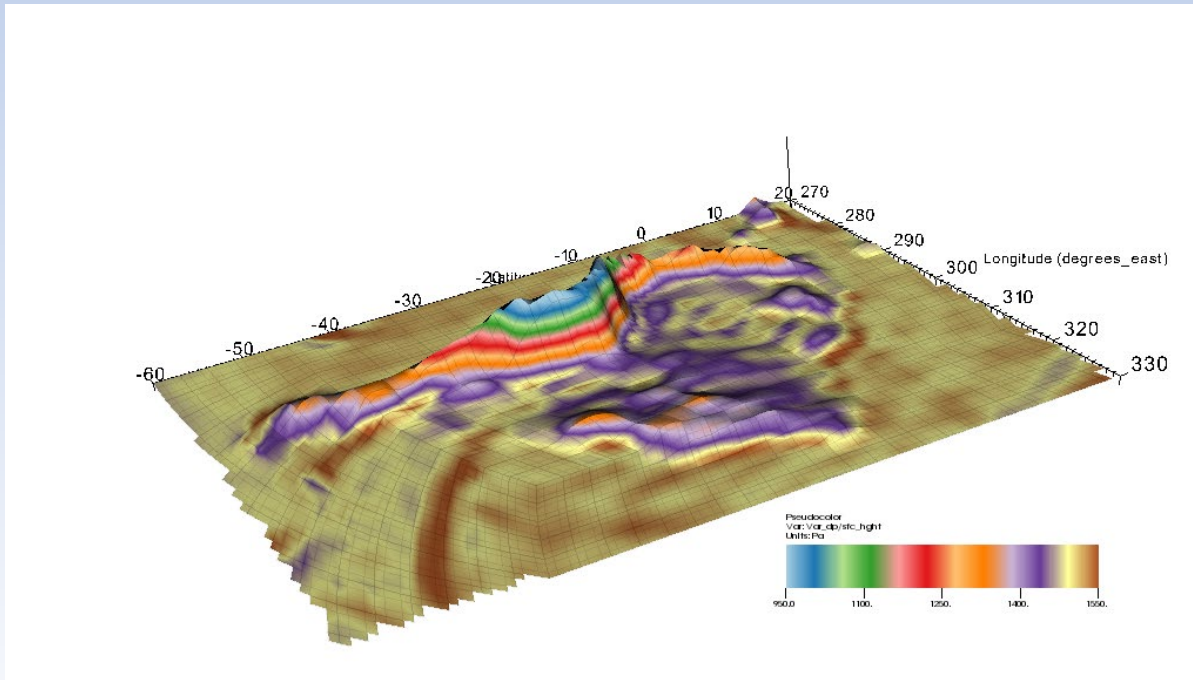
5. Further Diagnostic and Testing

VRM Diagnostics

The VRM_Diagnostics directory contains (**will contain**) utility programs and information useful for Variable resolution grids.

Plotting/
SEview/
VisIt_plugin/
Idealized(DCMIP)_tests?/

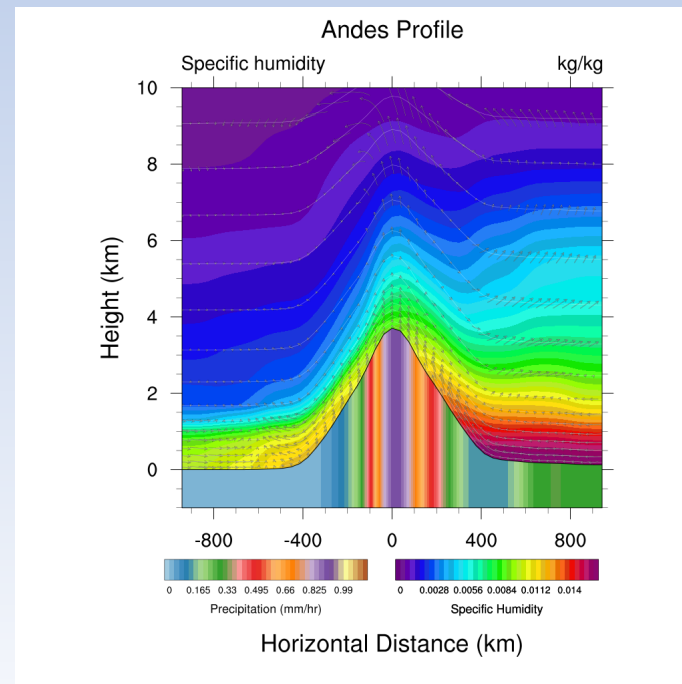
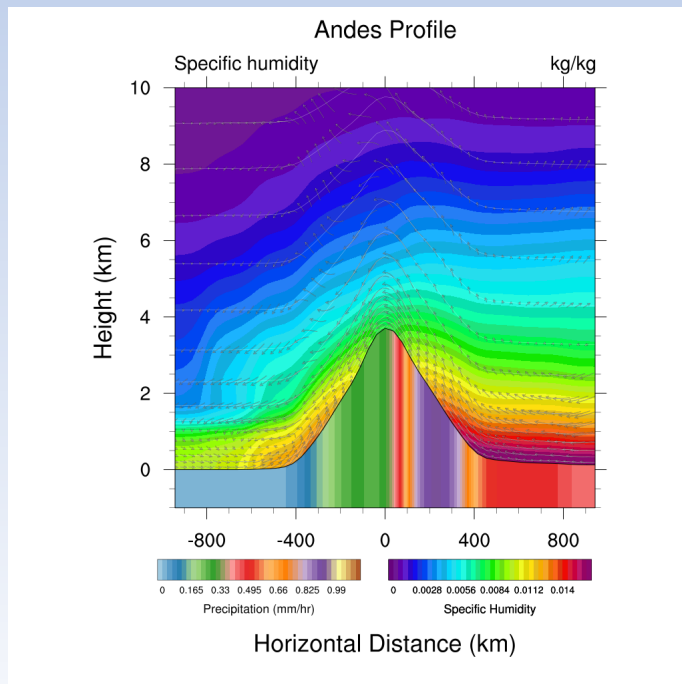
Contains ncl scripts for graphing EXODUS/SCRIP grids
Will eventually contain a simple viewer for the SE grids (like ncview)
Contains a plugin I wrote for the VisIt visualization program
Link to DCMIP tests & other idealized tests and guides
for running diagnostic testing of VRM grids



Further Development:

- Add NUOPC compatibility. This will add one more step to create an ESMF mesh file from the SCRIP file.
- Fix the remaining bug in CIME.
- Stop VRM_Editor from crashing randomly.

- Add a REFMAP smoothing option to VRM_Editor.
- Add more editing options to VRM_Editor.
- Make the processing more robust, not using hardwired source datasets.
- Add more diagnostic tests and viewing/analysis programs.



Summary:

A set of VRM_tools have been developed to greatly simplify the process of creating and using Variable Resolution Spectral Element grids.

The included documentation for the walks users through the process from the beginning to a working model grid.

SOON: will be available on https://github.com/ESCOMP/VRM_tools????? (VRM_toolkit?)

Once an EXODUS file has been created, the process of creating datasets and installing the grid in CESM consists of running 7 scripts.