CDO - recent additions and other gems 💎

Ralf Müller

DKRZ Hamburg

February 15, 2022

DKRZ

❓ What to expect

Overview:

- select
- pack
- dcw
- selcircle
- grid handling 2.0

DKRZ

❓ What to expect

Overview:

- select
- pack
- dcw
- selcircle
- grid handling 2.0
- other gems
  - skewness, kurtosis
  - bottomvalue, topvalue
  - expr
  - ...

DKRZ

❓ What to expect

Overview:

- select
- pack
- dcw
- selcircle
- grid handling 2.0
- other gems
  - skewness, kurtosis
  - bottomvalue, topvalue
  - expr
  - ...

Questions + Diskussion:

- Bring your ideas and issues!

DKRZ

✂ select - jack of all trades

Selecting from datasets in the right order can be crucial for overall performance:

1. `cdo -selname,t -sellevidx,2/22 -seltimestep,4/12 ...` vs.
2. `cdo -seltimestep,4/12 -selname,t -sellevidx,2/22 ...`

Which one is better?

select - jack of all trades

Selecting from datasets in the right order can be crucial for overall performance:

1. `cdo -selname,t -sellevidx,2/22 -seltimestep,4/12 ...` vs.
2. `cdo -seltimestep,4/12 -selname,t -sellevidx,2/22 ...`

Which one is better?

Cutting data into pieces along all directions

cdo -select,<param>=...

code	date	day
dom	enddate	gridname
gridnum	hour	level
levidx	levrange	ltype
minute	month	season
startdate	steptype	timestep
timestepmask	timestep_of_year	year
zaxisname	zaxisnum	

DKRZ

select - jack of all trades

Selecting from datasets in the right order can be crucial for overall performance:

1. `cdo -selname,t -sellevidx,2/22 -seltimestep,4/12 ...` vs.
2. `cdo -seltimestep,4/12 -selname,t -sellevidx,2/22 ...`

Which one is better?

Cutting data into pieces along all directions

cdo -select,<param>=...

code	date	day
dom	enddate	gridname
gridnum	hour	level
levidx	levrange	ltype
minute	month	season
startdate	steptype	timestep
timestepmask	timestep_of_year	year
zaxisname	zaxisnum	

Example

```
cdo -select,name=w_*,levidx=3,timestep=1/33 <ifile> <ofile>
```

For performance, lets have a look at a real world scenario

pack - lossy compression for netcdf (a bit like GRIB)

What it does

- Computes data ranges of all variables in a file
- Calculates proper `add_offset` and `scale_factor` attributes
- Change data type to 16-bit integer (configurable with `-b` option)

pack - lossy compression for netcdf (a bit like GRIB)

What it does

- Computes data ranges of all variables in a file
- Calculates proper `add_offset` and `scale_factor` attributes
- Change data type to 16-bit integer (configurable with `-b` option)

Pros and Cons

+ Built-in feature of netcdf: tools should work with it
+ Flexible way to compress data as you like it
+ Can be combined with `-z zip`
- Memory intensive: Range is calculated across all variables
- Not variable specific
- Precision loss depends in the input range

DKRZ

pack - lossy compression for netcdf (a bit like GRIB)

What it does

- Computes data ranges of all variables in a file
- Calculates proper `add_offset` and `scale_factor` attributes
- Change data type to 16-bit integer (configurable with `-b` option)

Pros and Cons

+ Built-in feature of netcdf: tools should work with it
+ Flexible way to compress data as you like it
+ Can be combined with `-z zip`
- Memory intensive: Range is calculated across all variables
- Not variable specific
- Precision loss depends in the input range

Compression and data loss (tested with single topo field)

Integer type	**file size ration**	**incl. zip compression**	**rel. diff**
I8	3.9	5.7	0.99
I16	2	2.4	0.24
I32	1	1.1	5.e-6

⭕ selcircle - need something different than sellonlatbox (thx vera)?

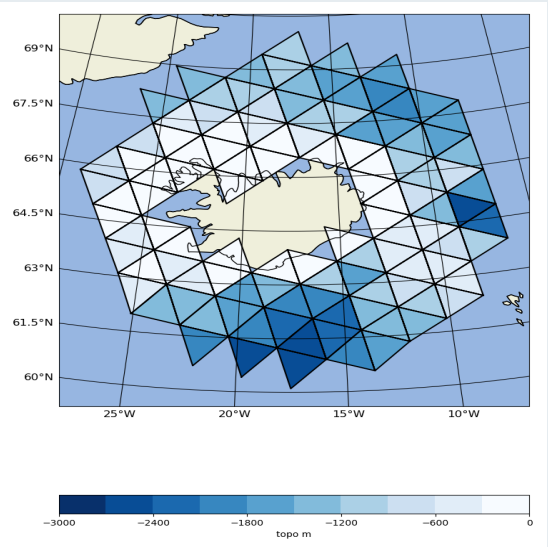
Sellonlatbox is well-known, but not suitable in all situations. Why not a cicrle around a given point?

selcircle - need something different than sellonlatbox (thx vera)?

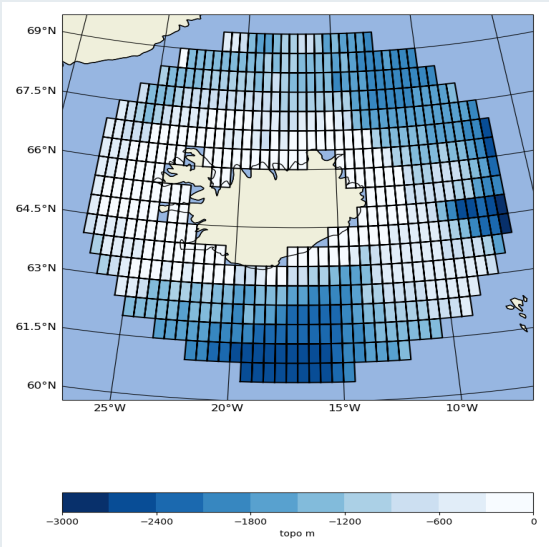
Sellonlatbox is well-known, but not suitable in all situations. Why not a cicrle around a given point?

cdo -f nc -selcircle,lon=-17,lat=65.11,radius=500km -setvrange,-100000,0 -topo,<grid>

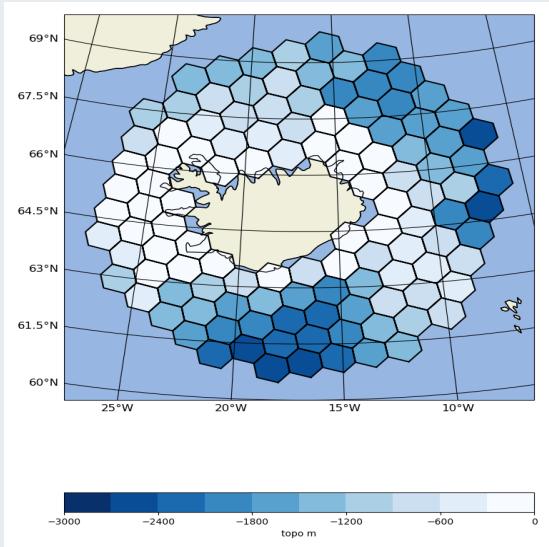
icon



lonlat



gme



Radius can be given in deg, rad, km and m

dcw - select countries on another level

Digital Chart of the World

This dataset has borders for countries 1992. Everyone can download it from here. DKRZ user can use it after `export DIR_DCW=/sw/rhel6-x64/gmt/dcw-1.1.1`

```
L: r=DE+FR+CH+AT+DK+IT; cdo -f nc -maskregion,dcw:${r} -topo,dcw:${r} out.nc
M: cdo -f nc -maskregion,dcw:DE -temp,icor2b6 DeIcon.nc
R: cdo -f nc -maskregion,dcw:SV -random,dcw:SV_0.005 sv.nc
```

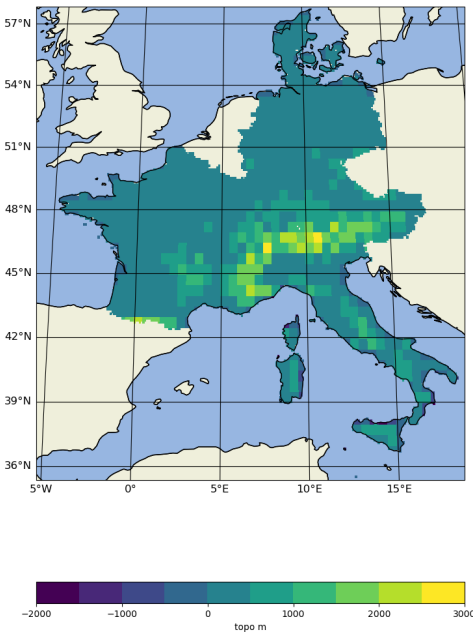
DKRZ

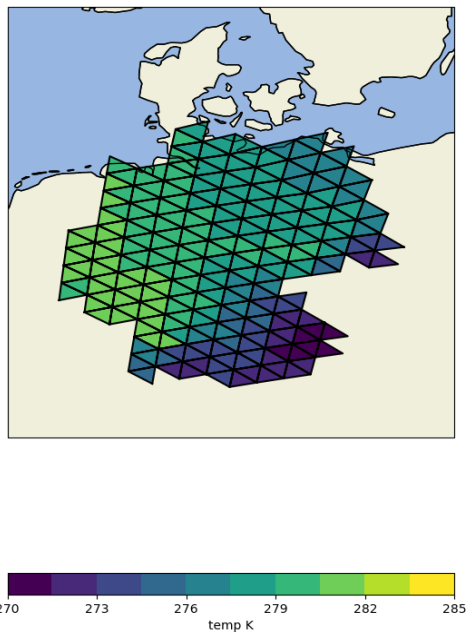
dcw - select countries on another level

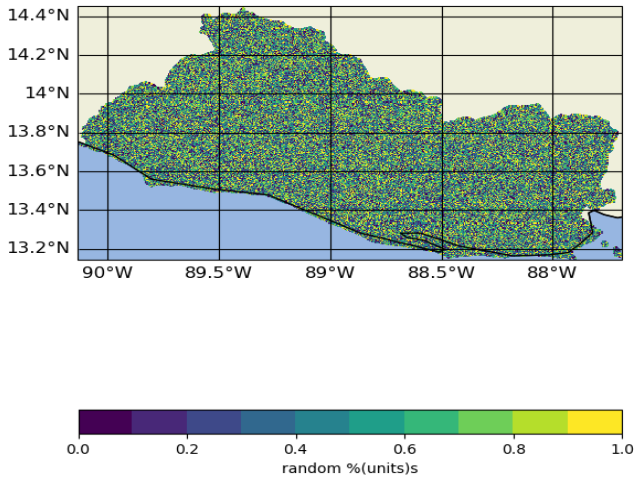
Digital Chart of the World

This dataset has borders for countries 1992. Everyone can download it from here. DKRZ user can use it after `export DIR_DCW=/sw/rhel6-x64/gmt/dcw-1.1.1`

```
L: r=DE+FR+CH+AT+DK+IT; cdo -f nc -maskregion,dcw:${r} -topo,dcw:${r} out.nc
M: cdo -f nc -maskregion,dcw:DE -temp,icor2b6 DeIcon.nc
R: cdo -f nc -maskregion,dcw:SV -random,dcw:SV_0.005 sv.nc
```







Watch out for `selregion` in release 2.0.4!

ICON - Riders of the lost grid 🤠

Indy, where is the Grid?

High resolution data usually comes without coordinates because they consume considerable diskspace. Still they are needed for many operations.

```
cdo sinfon dpp0016_ocean3D_u1000m.nc

File format : NetCDF4 zip
  -1 : Institut Source   T Steptype Levels Num    Points Num Dtype : Parameter name
   1 : MPIMET   git      v instant       3   1  14886338   1  F32z : to
   2 : MPIMET   git      v instant       3   1  14886338   1  F32z : so
Grid coordinates :
   1 : unstructured             : points=14886338
     grid : number=16  position=1
     uri : http://icon-downloads.mpimet.mpg.de/grids/public/mpim/0016/icon_grid_0016_R02B09_O.nc
     uuid : 375cb0cc-637e-11e8-9d6f-8f41a9b9ff4b
```

ICON - Riders of the lost grid 🤠

Indy, where is the Grid?

High resolution data usually comes without coordinates because they consume considerable diskspace. Still they are needed for many operations.

```
cdo sinfon dpp0016_ocean3D_u1000m.nc

File format : NetCDF4 zip
  -1 : Institut Source   T Steptype Levels Num    Points Num Dtype : Parameter name
   1 : MPIMET   git      v instant       3   1  14886338   1  F32z : to
   2 : MPIMET   git      v instant       3   1  14886338   1  F32z : so
Grid coordinates :
   1 : unstructured             : points=14886338
     grid : number=16  position=1
     uri : http://icon-downloads.mpimet.mpg.de/grids/public/mpim/0016/icon_grid_0016_R02B09_O.nc
     uuid : 375cb0cc-637e-11e8-9d6f-8f41a9b9ff4b
```

But what if ...

```
cdo -v sellonlatbox,160,210,-5,5 dpp0016_ocean3D_u1000m.nc result
cdo    sellonlatbox: Download horizontal grid file to ./icon_grid_0016_R02B09_O.nc
cdo    sellonlatbox: Processed 89318028 values from 2 variables over 1 timestep [829.81s 5485MB 1thread].
```

DKRZ

ICON - Riders of the lost grid 🤠

Indy, where is the Grid?

High resolution data usually comes without coordinates because they consume considerable diskspace. Still they are needed for many operations.

```
cdo sinfon dpp0016_ocean3D_u1000m.nc

File format : NetCDF4 zip
  -1 : Institut Source   T Steptype Levels Num    Points Num Dtype : Parameter name
   1 : MPIMET   git      v instant       3   1  14886338   1  F32z : to
   2 : MPIMET   git      v instant       3   1  14886338   1  F32z : so
Grid coordinates :
   1 : unstructured             : points=14886338
     grid : number=16  position=1
     uri : http://icon-downloads.mpimet.mpg.de/grids/public/mpim/0016/icon_grid_0016_R02B09_O.nc
     uuid : 375cb0cc-637e-11e8-9d6f-8f41a9b9ff4b
```

But what if ...

```
cdo -v sellonlatbox,160,210,-5,5 dpp0016_ocean3D_u1000m.nc result
cdo    sellonlatbox: Download horizontal grid file to ./icon_grid_0016_R02B09_O.nc
cdo    sellonlatbox: Processed 89318028 values from 2 variables over 1 timestep [829.81s 5485MB 1thread].
```

And now?

```
cdo -v sellonlatbox,160,210,-5,5 dpp0016_ocean3D_u1000m.nc result
cdo    sellonlatbox: Horizontal grid file used: ./icon_grid_0016_R02B09_O.nc
cdo    sellonlatbox: Processed 89318028 values from 2 variables over 1 timestep [15.39s 5487MB 1thread].
```

ICON - It's magic ...(don't know which kind though) 🪄

But wait! Now every user stores all grids?

Short: It depends

ICON - It's magic ...(don't know which kind though) 🪄

But wait! Now every user stores all grids?

Short: It depends

Long: CDO offers variables (A) `CDO_DOWNLOAD_PATH` and (B) `CDO_ICON_GRIDS` for that.

ICON - It's magic ...(don't know which kind though) 🪄

But wait! Now every user stores all grids?

Short: It depends
Long: CDO offers variables (A) `CDO_DOWNLOAD_PATH` and (B) `CDO_ICON_GRIDS` for that.

A: Where to store the downloads?

By setting the `CDO_DOWNLOAD_PATH` users can tell CDO where to store the grids. This can be a directory optimized for performance (like `/tmp`) or for sharing with other users (like a project folder)

DKRZ

ICON - It's magic ...(don't know which kind though) 🪄

But wait! Now every user stores all grids?

Short: It depends
Long: CDO offers variables (A) CDO_DOWNLOAD_PATH and (B) CDO_ICON_GRIDS for that.

A: Where to store the downloads?

By setting the CDO_DOWNLOAD_PATH users can tell CDO where to store the grids. This can be a directory optimized for performance (like /tmp) or for sharing with other users (like a project folder)

B: And at DKRZ?

ICON has lots of grids in use. Many of them are public because ICON is a production NWP model. DKRZ hosts them also in /pool:

```
export CDO_ICON_GRIDS=/pool/data/ICON
cdo -v sellonlatbox,160,210,-5,5 dpp0016_ocean3D_u1000m.nc result
cdo    sellonlatbox: Horizontal grid file used: /pool/data/ICON/grids/public/mpim/0016/icon_grid_0016_R02B09_O.nc
cdo    sellonlatbox: Processed 89318028 values from 2 variables over 1 timestep [15.78s 5487MB 1thread].
```

DKRZ

:n - select grids from grids

Short wrapup about grid handing in CDO

For CDO grids occur in different form:

:n - select grids from grids

Short wrapup about grid handing in CDO

For CDO grids occur in different form:

1. showcuts: global_1, gme98, F63, r1440x720, t63, ...

:n - select grids from grids

Short wrapup about grid handing in CDO

For CDO grids occur in different form:

1. showcuts: global_1, gme98, F63, r1440x720, t63, ...
2. text files: similar to the output of `cdo griddes` ...

:n - select grids from grids

Short wrapup about grid handing in CDO

For CDO grids occur in different form:

1. showcuts: global_1, gme98, F63, r1440x720, t63, ...
2. text files: similar to the output of `cdo griddes` ...
3. data files: model output, grid files (hopefully CF-conform or grib) ⇐

:n - select grids from grids

Short wrapup about grid handing in CDO

For CDO grids occur in different form:

1. showcuts: global_1, gme98, F63, r1440x720, t63, ...
2. text files: similar to the output of `cdo griddes` ...
3. data files: model output, grid files (hopefully CF-conform or grib) ⇐

Grid information from files are taken from the **first** grid that CDO can find:

```
Grid coordinates :
1 : unstructured             : points=15117  nvertex=3
                       clon : -3.141593 to 3.141593 radian
                       clat : -1.394108 to 1.552894 radian
                  available : cellbounds
                       uuid : f9f0d014-5735-11e3-bcac-e7ae25bb605f
2 : unstructured             : points=23195  nvertex=4
                       elon : -3.141593 to 3.141593 radian
                       elat : -1.402688 to 1.556312 radian
                  available : cellbounds
                       uuid : f9f0d014-5735-11e3-bcac-e7ae25bb605f
3 : unstructured             : points=8046  nvertex=6
                       vlon : -3.141593 to 3.141593 radian
                       vlat : -1.411733 to 1.570787 radian
                  available : cellbounds
                       uuid : f9f0d014-5735-11e3-bcac-e7ae25bb605f
```

DKRZ

:n - select grids from grids

What if you don't want the first grid?

1. Create an extra file: `selname`, `selgrid`, ...
2. Use this output for further steps: `setgrid`, `remap`, ...

As a bonus you get temporary files, which (a) cost diskspace and (b) need extra care for possible re-use and later cleanup.

:n - select grids from grids

What if you don't want the first grid?

1. Create an extra file: `selname`, `selgrid`, ...
2. Use this output for further steps: `setgrid`, `remap`, ...

As a bonus you get temporary files, which (a) cost diskspace and (b) need extra care for possible re-use and later cleanup.

No more extra files!

This is not needed anymore - Use the `:` instead to select the corresponding grid on-the-fly:

```
cdo -setgrid,icon-grid-158km.nc:2 -selname,vn icon-output.nc <ofile>
```

Please check the given grid number of your input files with operators like `sinfo` or `verifygrid` before using this feature!

This should be extra useful for high-resolution data since it improves parallel processing.

DKRZ

🙈 ℂomplex operators

CDO has operators to work with complex number:

- retocomplex: generate complex numbers out of real ones, imaginary part set to zero
- complextorect/recttocomplex: conversion complex to real and imaginary part and vice versa
- fourier: fft $\mathbb{C} \rightarrow \mathbb{C}$
- conj $\overline{z}$, sqr, sqrt, abs, arg, re: $\mathrm{Re}(z)$, im: $\mathrm{Im}(z)$
- add, sub, mul, div: basic arithmetics
- addc, subc, mulc, divc: arithgmetics with a real factor

DKRZ

🙈 ℂomplex operators

CDO has operators to work with complex number:

- retocomplex: generate complex numbers out of real ones, imaginary part set to zero
- complextorect/recttocomplex: conversion complex to real and imaginary part and vice versa
- fourier: fft $\mathbb{C} \rightarrow \mathbb{C}$
- conj $\overline{z}$, sqr, sqrt, abs, arg, re: $\mathrm{Re}(z)$, im: $\mathrm{Im}(z)$
- add, sub, mul, div: basic arithmetics
- addc, subc, mulc, divc: arithgmetics with a real factor

$\Longrightarrow$ Is is possible to generate fractals with CDO? ...🤔

🙈 ℂomplex can be beautiful

Remember the formular for the Mandelbrot set:

$$z_{n+1} = {z_n}^2 + c$$

Let's start with using the sphere as a local field of complex numbers:

```
gridtype  = lonlat
gridsize  = 6005001
xsize     = 3001
ysize     = 2001
xname     = lon
xlongname = "longitude"
xunits    = "degrees_east"
yname     = lat
ylongname = "latitude"
yunits    = "degrees_north"
xfirst    = -2
xinc      = 0.001
yfirst    = -1
yinc      = 0.001
```

DKRZ

🙈 ℂomplex can be beautiful

Remember the formular for the Mandelbrot set:

$$z_{n+1} = {z_n}^2 + c$$

Let's start with using the sphere as a local field of complex numbers:

```
gridtype  = lonlat
gridsize  = 6005001
xsize     = 3001
ysize     = 2001
xname     = lon
xlongname = "longitude"
xunits    = "degrees_east"
yname     = lat
ylongname = "latitude"
yunits    = "degrees_north"
xfirst    = -2
xinc      = 0.001
yfirst    = -1
yinc      = 0.001
```

```
cdo -f nc -expr,'z=clon(const)' -const,grid Real.nc
cdo -f nc -expr,'z=clat(const)' -const,grid Imag.nc
cdo -f nc4 recttocomplex Real.nc Imaginary.nc Complex.nc
```

DKRZ

🙈 ℂomplex can be beautiful

Remember the formular for the Mandelbrot set:

$$z_{n+1} = {z_n}^2 + c$$

Let's start with using the sphere as a local field of complex numbers:

```
gridtype  = lonlat
gridsize  = 6005001
xsize     = 3001
ysize     = 2001
xname     = lon
xlongname = "longitude"
xunits    = "degrees_east"
yname     = lat
ylongname = "latitude"
yunits    = "degrees_north"
xfirst    = -2
xinc      = 0.001
yfirst    = -1
yinc      = 0.001
```

```
cdo -f nc -expr,'z=clon(const)' -const,grid Real.nc
cdo -f nc -expr,'z=clat(const)' -const,grid Imag.nc
cdo -f nc4 recttocomplex Real.nc Imaginary.nc Complex.nc
```

```
def mandelNth(n,c)
  if 1 >= n then
    return c
  else
    mandelNM1 = mandelNth(n-1,c)
    @cdo.add(input: "-mul #{mandelNM1} #{mandelNM1} #{c}",
             output: "mandel_Iter#{n.to_s.rjust(3,'0')}.nc",
             options: '-L -f nc4', force: false)
  end
end
```

🙈 ℂomplex can be beautiful

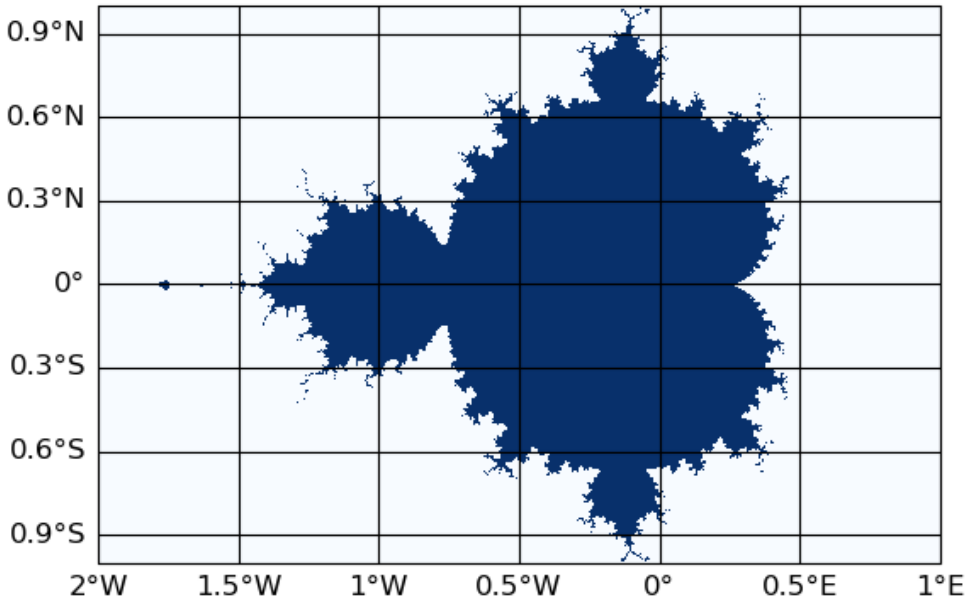
Remember the formular for the Mandelbrot set:

$$z_{n+1} = {z_n}^2 + c$$

Let's start with using the sphere as a local field of complex numbers:

```
gridtype  = lonlat
gridsize  = 6005001
xsize     = 3001
ysize     = 2001
xname     = lon
xlongname = "longitude"
xunits    = "degrees_east"
yname     = lat
ylongname = "latitude"
yunits    = "degrees_north"
xfirst    = -2
xinc      = 0.001
yfirst    = -1
yinc      = 0.001
```

```
cdo
cdo
cdo
 def
   i
   e
   e
 end
```




```
ɛ.nc

 #{c}",
)')}.nc",
```

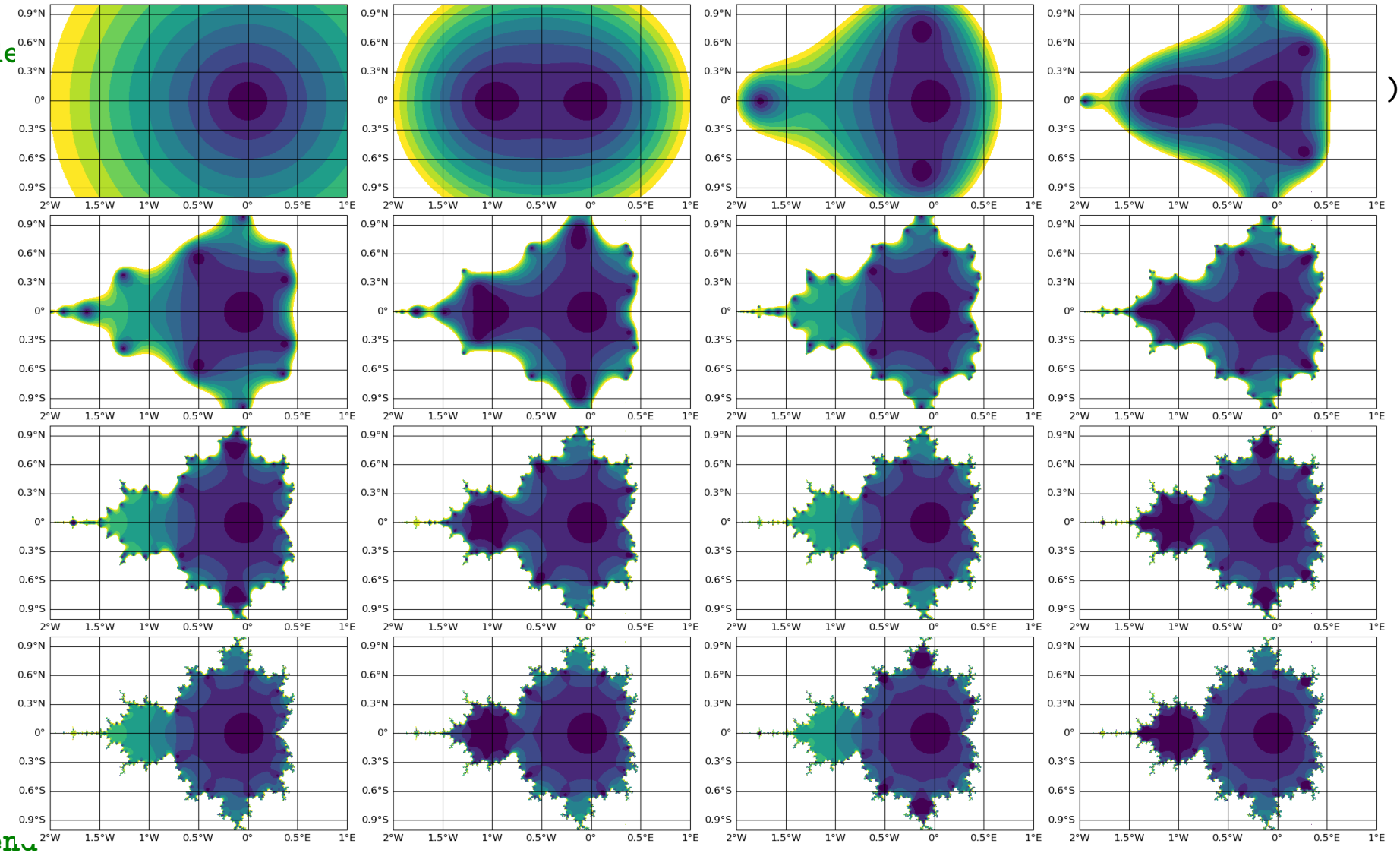
But what happens *inside*?

But what happens *inside*?

```ruby
def limitCto(z,limit=5.0)
  targetOutputFile = File.basename(z,File.extname(z))+'_limited'+File.extname(z)
  # dont re-compute things twice
  return targetOutputFile if File.exists?(targetOutputFile)

  # compute absolute value of z
  amplitude = @cdo.abs(input: z)

  # compute mask based on limitted amplitude
  # 1) positive mask for multiplication (0-1)
  maskPositive = @cdo.ltc(limit, input: amplitude)
  # 2) negative mask for later addition (5-0)
  maskNegative = @cdo.mulc(limit, input: "-gtc,#{limit} #{amplitude}")

  # mask real and imaginary part - not with missing value, but with the
  # limit since missing values are not possible with complex numbers atm
  zX, zY = @cdo.complextorect(input: z).split
  zXNew  = @cdo.add(input: [maskNegative, "-mul #{zX} #{maskPositive}"])
  zYNew  = @cdo.add(input: [maskNegative, "-mul #{zY} #{maskPositive}"])
  zNew   = @cdo.recttocomplex(input: [zXNew, zYNew], output: targetOutputFile)
  return zNew
end
```

DKRZ

But what happens *inside*?



Worth mentioning ...

bottomvalue/topvalue

depends on your position - useful or ocean data

verifygrid

analyze your grid in-depth: cell shapes, bounds, double coordinates ...

expr - coordinate functions

clon(), clat()	Longitude/Latitude
gridarea()	Grid cell area
clev()	Level coordinate
cdeltaz()	Upper minus lower level bound
ctimestep()	Timestep number (1 to N)
cdate()	Verification date as YYYYMMDD
ctime()	Verification time as HHMMSS.millisecond
cdeltat()	Difference between current and last timestep in seconds
cday(), cmonth(), cyear()	Day as DD, Month as MM, Year as YYYY
csecond(), cminute(), chour()	Second as SS.millisecond, Minute as MM, Hour as HH

Worth mentioning ...

skewness/kurtosis - statistics going nuts (thx dian)

For some poeple *normal* ensemble analysis is not enough. Therefore higher momentums **skewness** and **kurtosis** were included in CDO.
Both analyses are available for multiple operations:

- Ensemble: enskurt/ensskew
- Field: fldkurt/fldskew
- Gridbox: gridboxkurt/gridboxskew
- Meridional: merkurt/merskew
- Zonal: zonkurt/zonskew

Median ...or the 50th percentile

Like above: ensmedian fldmedian gridboxmedian mermedian zonmedian

🔫 Shoot!

```
1  (10*rand).to_i.times {
2    puts "Thank you for your attention!"
3  }
4  audience.select {|human|
5    human.has_questions?
6  }.each {|human|
7    human.ask!(please: true)
8  }
```

All plots were done with psyplot
More docu: FAQ and Operator News
For the curious: how to use emojis in latex 🙃

DKRZ