COM INTRO 2017: Interpolation - Solution to practical

1. Running the script interp1.ksh retrieves data from MARS to the file out1.grib.

Using grib_ls shows the field to be the 1000 hPa temperature in spectral format:

```
%  grib_ls out1.grib
out1.grib
edition      centre       typeOfLevel  level        dataDate     stepRange    dataType     shortName    packingType  gridType
1            ecmf         isobaricInhPa  1000         20170222     0            an           t            spectral_complex  sh
1 of 1 grib messages in out1.grib

1 of 1 total grib messages in 1 files
```

Looking at Section 2 of the GRIB header with “grib_dump –O” shows:

```
======================   SECTION_2 ( length=32, padding=0 )    ======================
1-3       section2Length = 32
4         numberOfVerticalCoordinateValues = 0
5         pvlLocation = 255
6         dataRepresentationType = 50 [Spherical Harmonic Coefficients (grib1/6.table) ]
7-8       J = 1279
9-10      K = 1279
11-12     M = 1279
13        representationType = 1 [Associated Legendre Polynomials of the First Kind with
normalization such that the integral equals 1 (grib1/9.table) ]
14        representationMode = 2 [Spherical harmonics-complex packing (grib1/10.table) ]
```

The J, K and M keys here are the spectral truncation so this is data at a spectral resolution of T1279 (current HRES). This corresponds to the O1280 octahedral grid or ~9km resolution.

The MARS request has “resol = av” which means to get the data with the archived resolution without any additional post-processing (i.e., no transformation or interpolation).

2. The script interp2.ksh calls MARS to read this file with “resol = 319”:

```
mars<<EOF
read,
   source       = "out1.grib",
   resol        = 319,
   target       = "out2.grib"
EOF
```

With grib_ls, the output file out2.grib looks the same as out1.grib – the data are still in spectral format:

```
% grib_ls out2.grib
out2.grib
edition      centre       typeOfLevel  level        dataDate     stepRange    dataType     shortName    packingType  gridType
1            ecmf         isobaricInhPa  1000         20170222     0            an           t            spectral_complex  sh
1 of 1 grib messages in out2.grib

1 of 1 total grib messages in 1 files
```

But Section 2 of the GRIB header shows the differences:

```
======================   SECTION_2 ( length=32, padding=0 )    ======================
1-3       section2Length = 32
4         numberOfVerticalCoordinateValues = 0
5         pvlLocation = 255
6         dataRepresentationType = 50 [Spherical Harmonic Coefficients (grib1/6.table) ]
7-8       J = 319
9-10      K = 319
11-12     M = 319
13        representationType = 1 [Associated Legendre Polynomials of the First Kind with
normalization such that the integral equals 1 (grib1/9.table) ]
14        representationMode = 2 [Spherical harmonics-complex packing (grib1/10.table) ]
```

Now the J, K and M keys are set to 319 to indicate that the data are at the lower T319 spectral resolution. The spectral series has been further truncated. This would correspond to an O320 octahedral grid or ~36km resolution.

3. The script interp3.ksh calls MARS to again read data from the out1.grib file (the T1279 data) but now uses “grid = N400”:

```
mars<<EOF
read,
   source       = "out1.grib",
   grid         = N400,
   target       = "out3.grib"
EOF
```

Using grib_ls we see the data have been transformed to a reduced Gaussian grid (gridType = reduced_gg):

```
grib_ls out3.grib
out3.grib
edition      centre       typeOfLevel  level        dataDate     stepRange    dataType     shortName    packingType  gridType
1            ecmf         isobaricInhPa  1000         20170222     0            an           t            grid_simple  reduced_gg
1 of 1 grib messages in out3.grib

1 of 1 total grib messages in 1 files
```

Section 2 of the GRIB header shows this is a reduced (Ni = MISSING) Gaussian grid with N=400 lines between pole and equator (Nj=800 lines from pole to pole). It is a global grid.

```
======================   SECTION_2 ( length=1632, padding=0 )   ======================
1-3       section2Length = 1632
4         numberOfVerticalCoordinateValues = 0
5         pvlLocation = 33
6         dataRepresentationType = 4 [Gaussian Latitude/Longitude Grid (grib1/6.table) ]
7-8       Ni = MISSING
9-10      Nj = 800
11-13     latitudeOfFirstGridPoint = 89828
14-16     longitudeOfFirstGridPoint = 0
17        resolutionAndComponentFlags = 0 [00000000]
18-20     latitudeOfLastGridPoint = -89828
21-23     longitudeOfLastGridPoint = 359775
24-25     iDirectionIncrement = MISSING
26-27     N = 400
28        scanningMode = 0 [00000000]
29-32     padding_grid4_1 = 4 {
                      00, 00, 00, 00
                  } # pad padding_grid4_1
```

```
33-1632   pl = (800,1600) {
```

The pl array gives the number of longitude points at each latitude line. Because this starts with 18 this is an original style reduced Gaussian grid (the pl array of the octahedral grid starts at 20 and increases by 4 at each element to the equator).

4. The script interp4.ksh calls MARS to again read data from the out1.grib file (the T1279 data) but now uses “grid = F400”:

```
mars<<EOF
read,
   source       = "out1.grib",
   grid         = F400,
   target       = "out4.grib"
EOF
```

Using grib_ls we see the data have been transformed to a reduced Gaussian grid (gridType = regular_gg):

```
% grib_ls out4.grib
out4.grib
edition      centre       typeOfLevel  level        dataDate     stepRange    dataType     shortName    packingType  gridType
1            ecmf         isobaricInhPa  1000         20170222     0            an           t            grid_simple  regular_gg
1 of 1 grib messages in out4.grib

1 of 1 total grib messages in 1 files
```

Section 2 of the GRIB header shows this is a regular (Ni =1600) Gaussian grid with N=400 lines between pole and equator (Nj=800 lines from pole to pole). It is a global grid.

```
======================   SECTION_2 ( length=32, padding=0 )    ======================
1-3       section2Length = 32
4         numberOfVerticalCoordinateValues = 0
5         pvlLocation = 255
6         dataRepresentationType = 4 [Gaussian Latitude/Longitude Grid (grib1/6.table) ]
7-8       Ni = 1600
9-10      Nj = 800
11-13     latitudeOfFirstGridPoint = 89828
14-16     longitudeOfFirstGridPoint = 0
17        resolutionAndComponentFlags = 128 [10000000]
18-20     latitudeOfLastGridPoint = -89828
21-23     longitudeOfLastGridPoint = 359775
24-25     iDirectionIncrement = 225
```

```
26-27     N = 400
28        scanningMode = 0 [00000000]
```

There is no pl array in this case: all latitude lines have Ni=1600 longitude points.

5. The script interp5.ksh calls MARS to again read data from the out1.grib file (the T1279 data) but now uses “grid = O400”:

```
mars<<EOF
read,
   source       = "out1.grib",
   grid         = O400,
   target       = "out5.grib"
EOF
```

This is, of course, transforms the data to the O400 octahedral reduced Gaussian grid. Section 2 of the GRIB header is similar to that for the N400 data in out3.grib. The main difference is in the pl array which starts at 20.

6. The script interp6.ksh calls MARS to read data from the out5.grib file (the O400 data) but now uses “grid = 2.0/2.0”:

```
mars<<EOF
read,
   source       = "out5.grib",
   grid         = 2.0/2.0,
   target       = "out6.grib"
EOF
```

In this case, the O400 data is interpolated to a 2.0x2.0 degree regular lat-lon grid. Using grib_ls we see the gridType=regular_ll:

```
grib_ls out6.grib
out6.grib
edition      centre       typeOfLevel  level        dataDate     stepRange    dataType     shortName    packingType  gridType
1            ecmf         isobaricInhPa  1000         20170222     0            an           t            grid_simple  regular_ll
1 of 1 grib messages in out6.grib

1 of 1 total grib messages in 1 files
```

Section 2 of the GRIB header shows:

```
======================   SECTION_2 ( length=32, padding=0 )    ======================
```

```
1-3       section2Length = 32
4         numberOfVerticalCoordinateValues = 0
5         pvlLocation = 255
6         dataRepresentationType = 0 [Latitude/Longitude Grid (grib1/6.table) ]
7-8       Ni = 180
9-10      Nj = 91
11-13     latitudeOfFirstGridPoint = 90000
14-16     longitudeOfFirstGridPoint = 0
17        resolutionAndComponentFlags = 128 [10000000]
18-20     latitudeOfLastGridPoint = -90000
21-23     longitudeOfLastGridPoint = 358000
24-25     iDirectionIncrement = 2000
26-27     jDirectionIncrement = 2000
28        scanningMode = 0 [00000000]
```

7. The script interp7.ksh calls MARS to again read data from the out5.grib file (the O400 data) but now setting both grid and area keywords:

```
mars<<EOF
read,
   source       = "out5.grib",
   grid         = 0.5/0.5,
   area         = 1.5/-17.8/-21.0/8.3,
   target       = "out7.grib"
EOF
```

This interpolates the octahedral gridded data to a sub-area of a regular lat-lon grid at 0.5x0.5 degree resolution. Section 2 of the GRIB header shows:

```
======================   SECTION_2 ( length=32, padding=0 )    ======================
1-3       section2Length = 32
4         numberOfVerticalCoordinateValues = 0
5         pvlLocation = 255
6         dataRepresentationType = 0 [Latitude/Longitude Grid (grib1/6.table) ]
7-8       Ni = 54
9-10      Nj = 46
11-13     latitudeOfFirstGridPoint = 1500
14-16     longitudeOfFirstGridPoint = -18000
17        resolutionAndComponentFlags = 128 [10000000]
```

```
18-20     latitudeOfLastGridPoint = -21000
21-23     longitudeOfLastGridPoint = 8500
24-25     iDirectionIncrement = 500
26-27     jDirectionIncrement = 500
28        scanningMode = 0 [00000000]
```

The longitude and latitude of first and last grid points show the sub-area.

Note that the area retrieved is not the same as requested. MARS has adjusted the area boundaries to match an underlying 0.5x0.5 degree global grid containing a point at (0.0,0.0). The following is reported in the MARS output:

```
mars - WARN   - 20170223.115017 - Area not compatible with grid
mars - WARN   - 20170223.115017 - Area changed from 1.5/-17.8/-21/8.3 to 1.5/-18/-21/8.5
```

8. The script interp8.ksh calls MARS to again read data from the out5.grib file (the O400 data) but now setting the rotation keyword in addition to the grid and area keywords:

```
mars<<EOF
read,
   source       = "out5.grib",
   grid         = 0.5/0.5,
   area         = 1.5/-17.8/-21.0/8.3,
   rotation     = -32.5/10.0,
   target       = "out8.grib"
EOF
```

Using grib_ls we see the gridType reported at rotated_ll:

```
% grib_ls out8.grib
out8.grib
edition      centre       typeOfLevel  level        dataDate     stepRange    dataType     shortName    packingType  gridType
1            ecmf         isobaricInhPa  1000         20170222     0            an           t            grid_simple  rotated_ll
1 of 1 grib messages in out8.grib

1 of 1 total grib messages in 1 files
```

Section 2 of the GRIB header is:

```
=====================   SECTION_2 ( length=42, padding=0 )    =====================
1-3       section2Length = 42
4         numberOfVerticalCoordinateValues = 0
```

```
5         pvlLocation = 255
6         dataRepresentationType = 10 [Rotated Latitude/Longitude grid (grib1/6.table) ]
7-8       Ni = 53
9-10      Nj = 46
11-13     latitudeOfFirstGridPoint = 1500
14-16     longitudeOfFirstGridPoint = -17800
17        resolutionAndComponentFlags = 128 [10000000]
18-20     latitudeOfLastGridPoint = -21000
21-23     longitudeOfLastGridPoint = 8300
24-25     iDirectionIncrement = 500
26-27     jDirectionIncrement = 500
28        scanningMode = 0 [00000000]
29-32     zero =
33-35     latitudeOfSouthernPole = -32500
36-38     longitudeOfSouthernPole = 10000
39-42     angleOfRotationInDegrees = 0
```

In addition to the longitude and latitude of first and last grid points showing the sub-area (in the rotated frame) the latitude and longitude of the south pole is also reported.