cf-python Regridding

- **Based on the Earth System Modeling Framework (ESMF) library**
- **Spherical (regrids) or Cartesian (regridc)**
- **Regridding methods:**
  - **First order conservative**
  - **Bilinear**
  - **Higher order patch recovery**
  - **Nearest source to destination**
  - **Nearest destination to source**
- **Global or regional grids in any combination**
- **Grids with 2D latitudes and longitudes including some tripolar**
- **Handles masking of both source and destination**

Spherical regridding (regrids)

In [1]:

```
# Plot images inline - not needed in Python
%matplotlib inline
```

In [2]:

```
# Import the cf and cfplot packages
import cf
import cfplot as cfp
```

In [3]:

```
# Read in ncas_data/precip_2001-2010_low_res.nc and inspect the field
f = cf.read_field('ncas_data/precip_2010.nc')
print f
```

```
long_name:precipitation field summary
-------------------------------------
Data           : long_name:precipitation(long_name:time(12), latitude(145), longitude(53)) mm
Axes           : long_name:time(12) = [2010-01-16T00:00:00Z, ..., 2010-12-16T00:00:00Z] gregori
an
               : latitude(145) = [-90.0, ..., 90.0] degrees_north
               : longitude(53) = [-33.75, ..., 63.75] degrees_east
```

In [4]:

```
# Read in ncas_data/model_precip_DJF_means_low_res.nc and inspect the field
g = cf.read_field('ncas_data/model_precip_DJF_means_low_res.nc')
print g
```

```
long_name:precipitation field summary
-------------------------------------
Data           : long_name:precipitation(long_name:t(1), long_name:Surface(1), latitude(73), lo
ngitude(27)) mm/day
Cell methods   : long_name:t: mean
Axes           : long_name:t(1) = [1996-07-16T00:00:00Z] 360_day
               : long_name:Surface(1) = [0.0]
               : latitude(73) = [-90.0, ..., 90.0] degrees_north
               : longitude(27) = [-33.75, ..., 63.75] degrees_east
```

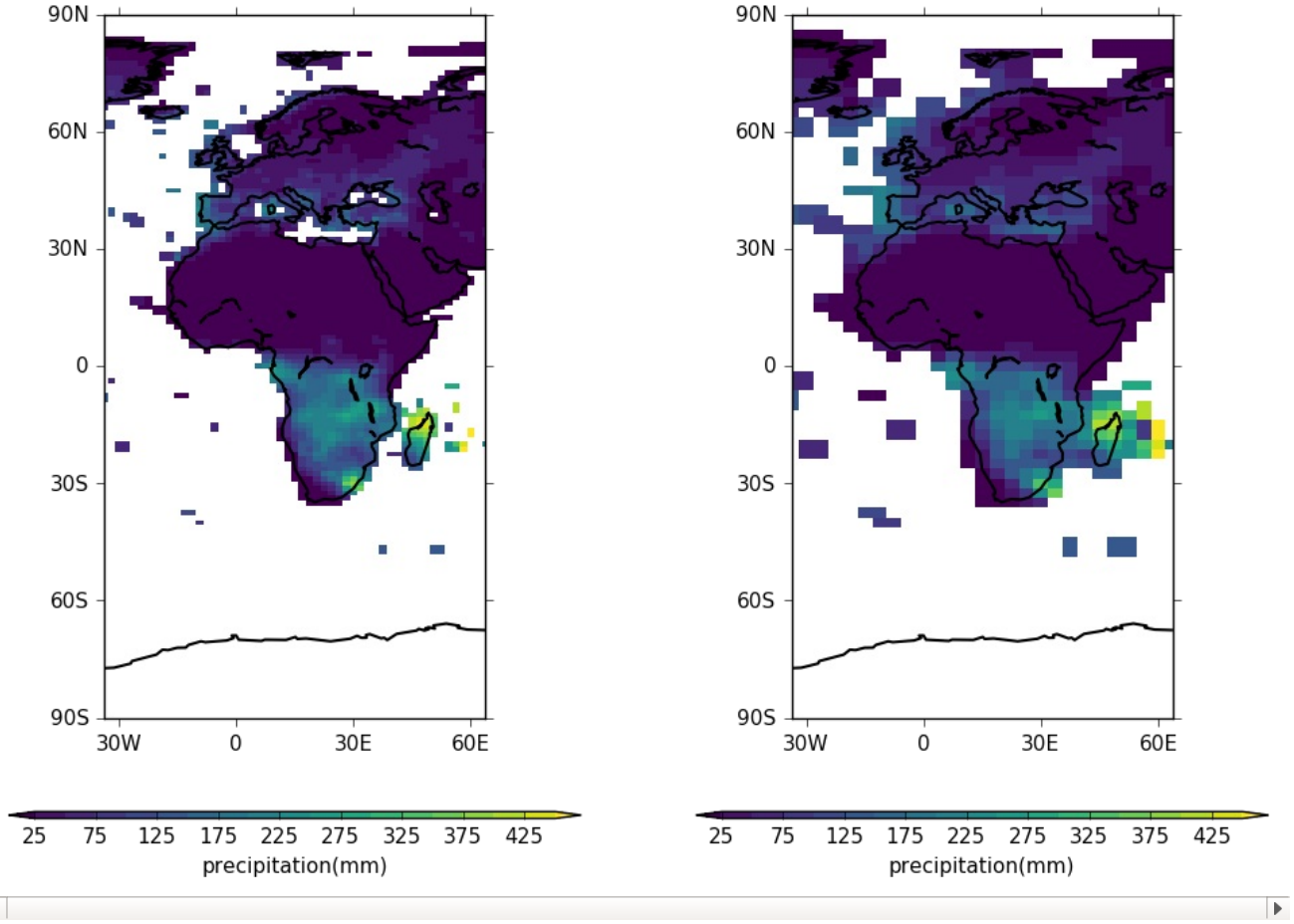
In [5]:

```
# Regrid the first field to the grid of the second and inspect the results
h = f.regrids(g, method='conservative')
print h
```

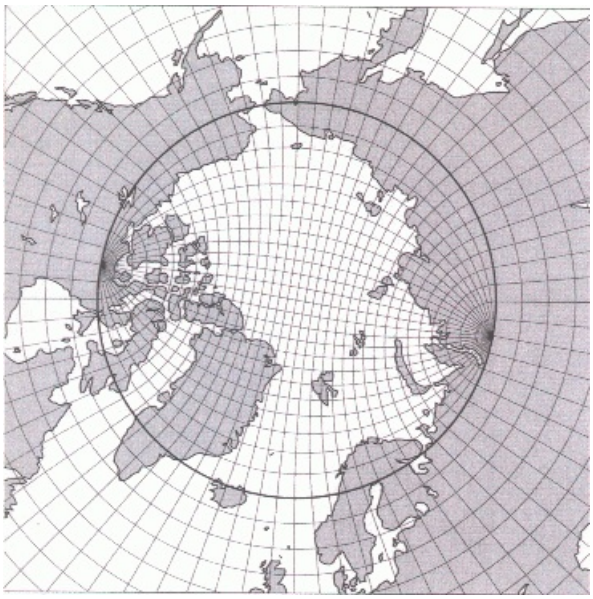
```
long_name:precipitation field summary
---------------------------------------
Data           : long_name:precipitation(long_name:time(12), latitude(73), longitude(27)) mm
Axes           : long_name:time(12) = [2010-01-16T00:00:00Z, ..., 2010-12-16T00:00:00Z] gregori
an
               : latitude(73) = [-90.0, ..., 90.0] degrees_north
               : longitude(27) = [-33.75, ..., 63.75] degrees_east
```

In [6]:

```
# Plot before and after
cfp.gopen(rows=1, columns=2)
cfp.gpos(1)
cfp.con(f[0], blockfill=True, lines=False, colorbar_label_skip=2)
cfp.gpos(2)
cfp.con(h[0], blockfill=True, lines=False, colorbar_label_skip=2)
cfp.gclose()
```



Regridding from a tripolar grid



In [7]:

```
# Read in ncas_data/tripolar.nc and select and inspect a field
f = cf.read_field('ncas_data/tripolar.nc')
print f
```

```
sea_surface_height_above_geoid field summary
--------------------------------------------
Data           : sea_surface_height_above_geoid(time(1), ncdim%y(332), ncdim%x(362)) m
Cell methods   : time: mean (interval: 2700 s)
Axes           : time(1) = [1978-09-06T00:00:00Z] 360_day
               : ncdim%y(332)
               : ncdim%x(362)
Aux coords     : time(time(1)) = [1978-09-06T00:00:00Z] 360_day
               : longitude(ncdim%y(332), ncdim%x(362)) = [[72.5, ..., 72.989151001]] degrees_ea
st
               : latitude(ncdim%y(332), ncdim%x(362)) = [[-84.2107086182, ..., 50.0109405518]]
degrees_north
```

In [8]:

```
# Read in ncas_data/model_precip_DJF_means.nc and inspect the field
g = cf.read_field('ncas_data/model_precip_DJF_means.nc')
print g
```

```
long_name:precipitation field summary
-------------------------------------
Data           : long_name:precipitation(long_name:t(1), long_name:Surface(1), latitude(145), l
ongitude(192)) mm/day
Cell methods   : long_name:t: mean
Axes           : long_name:t(1) = [1996-07-16T00:00:00Z] 360_day
               : long_name:Surface(1) = [0.0]
               : latitude(145) = [-90.0, ..., 90.0] degrees_north
               : longitude(192) = [0.0, ..., 358.125] degrees_east
```

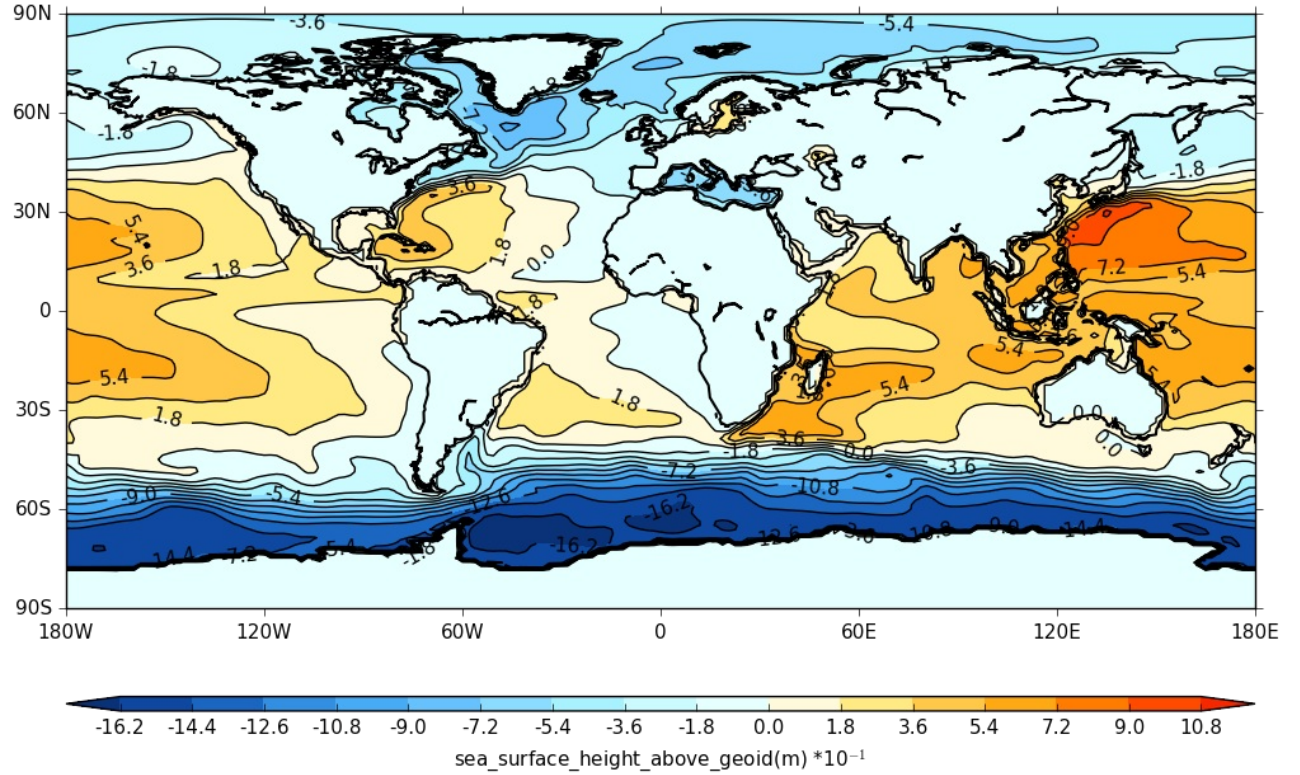
In [9]:

```
# Regrid the field on the tripolar grid to the regular lat-long grid
h = f.regrids(g, method='bilinear', src_axes={'X': 'ncdim%x', 'Y': 'ncdim%y'}, src_cyclic=True)
print h
```

```
sea_surface_height_above_geoid field summary
--------------------------------------------
Data           : sea_surface_height_above_geoid(time(1), latitude(145), longitude(192)) m
Cell methods   : time: mean (interval: 2700 s)
Axes           : time(1) = [1978-09-06T00:00:00Z] 360_day
               : latitude(145) = [-90.0, ..., 90.0] degrees_north
               : longitude(192) = [0.0, ..., 358.125] degrees_east
Aux coords     : time(time(1)) = [1978-09-06T00:00:00Z] 360_day
```

In [10]:

```
# Plot the regridded data
cfp.con(h)
```



Higher order patch recovery versus bilinear regridding

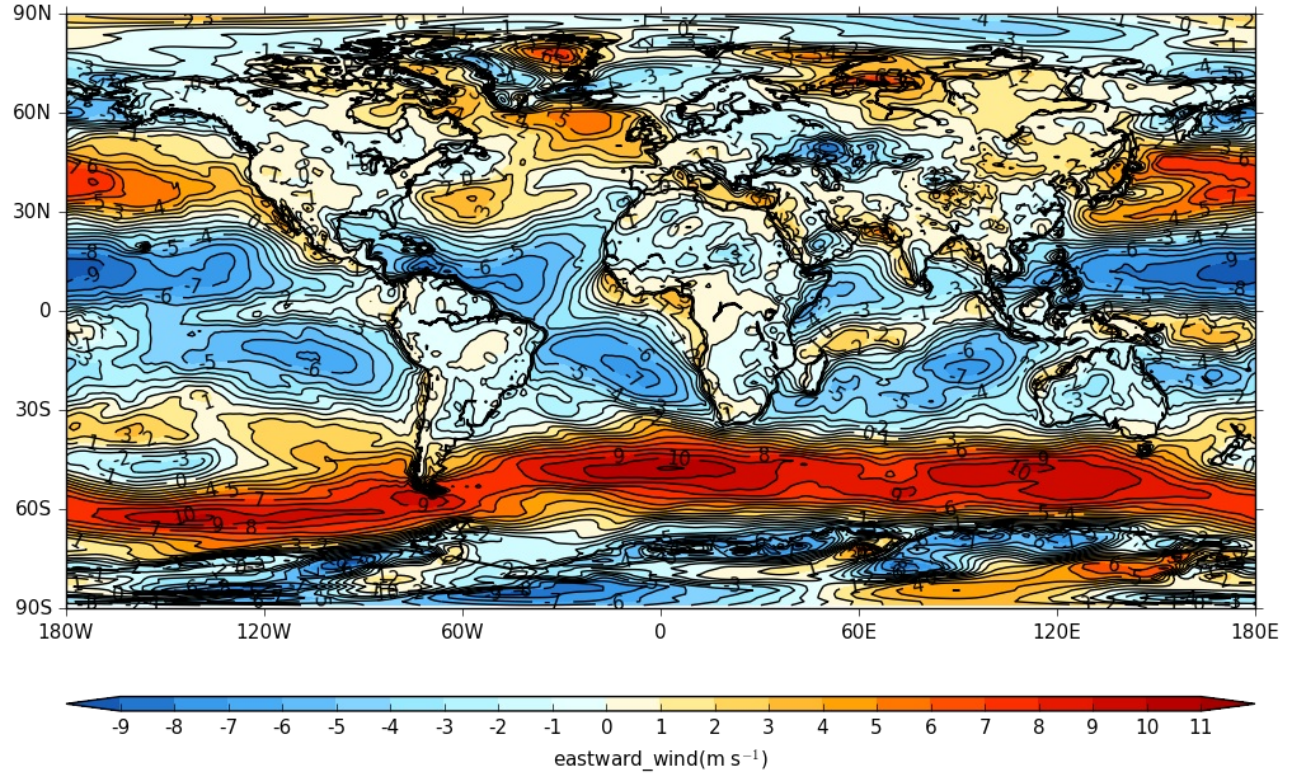
In [11]:

```
# Read in ncas_data/data5.nc and inspect the field
f = cf.read_field('ncas_data/data5.nc').subspace[0, 0]
print f
```

```
eastward_wind field summary
---------------------------
Data           : eastward_wind(time(1), pressure(1), latitude(160), longitude(320)) m s**-1
Axes           : time(1) = [1987-03-15T00:00:00Z]
               : pressure(1) = [1000.0] mbar
               : latitude(160) = [89.1415176392, ..., -89.1415176392] degrees_north
               : longitude(320) = [0.0, ..., 358.875] degrees_east
```

In [12]:

```
cfp.con(f)
```



In [13]:

```
# Read in ncas_data/model_precip_DJF_means_low_res.nc and inspect the field
g = cf.read_field('ncas_data/model_precip_DJF_means_low_res.nc')
print g
```

```
long_name:precipitation field summary
--------------------------------------
Data           : long_name:precipitation(long_name:t(1), long_name:Surface(1), latitude(73), lo
ngitude(27)) mm/day
Cell methods   : long_name:t: mean
Axes           : long_name:t(1) = [1996-07-16T00:00:00Z] 360_day
               : long_name:Surface(1) = [0.0]
               : latitude(73) = [-90.0, ..., 90.0] degrees_north
               : longitude(27) = [-33.75, ..., 63.75] degrees_east
```

In [14]:

```
# Regrid the first field to the grid of the second using bilinear interpolation
h = f.regrids(g, method='bilinear')
print h
```

```
eastward_wind field summary
---------------------------
Data           : eastward_wind(time(1), pressure(1), latitude(73), longitude(27)) m s**-1
Axes           : time(1) = [1987-03-15T00:00:00Z]
               : pressure(1) = [1000.0] mbar
               : latitude(73) = [-90.0, ..., 90.0] degrees_north
               : longitude(27) = [-33.75, ..., 63.75] degrees_east
```

In [15]:

```
# Regrid the first field to the grid of the second using higher order patch recovery
i = f.regrids(g, method='patch')
print i
```

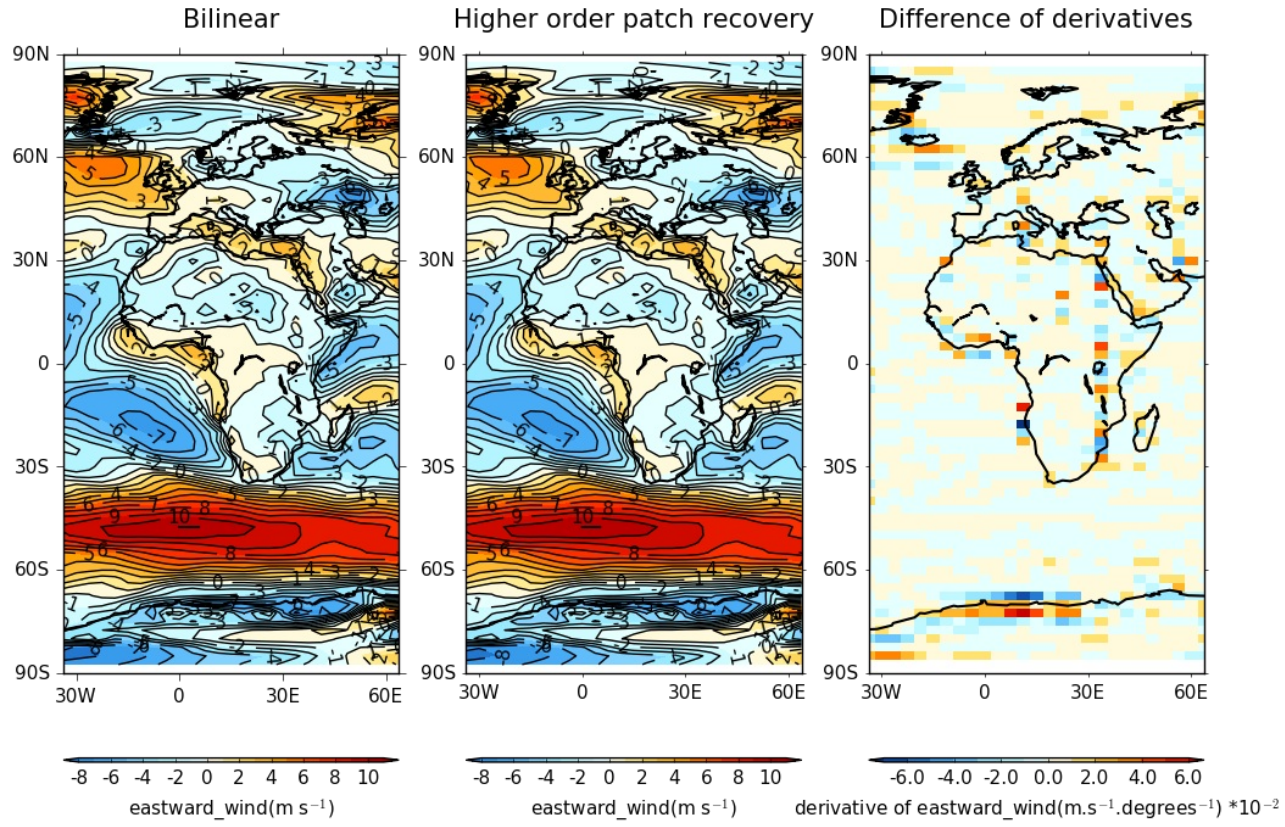
```
eastward_wind field summary
---------------------------
Data           : eastward_wind(time(1), pressure(1), latitude(73), longitude(27)) m s**-1
Axes           : time(1) = [1987-03-15T00:00:00Z]
               : pressure(1) = [1000.0] mbar
               : latitude(73) = [-90.0, ..., 90.0] degrees_north
               : longitude(27) = [-33.75, ..., 63.75] degrees_east
```

In [17]:

```
# Find the y derivatives of the regridded fields
deriv_h = h.derivative('Y')
deriv_h.units = 'm.s-1.degrees-1'
deriv_i = i.derivative('Y')
deriv_i.units = 'm.s-1.degrees-1'
```

In [18]:

```
# Plot the regridded fields and the differences between the derivatives
cfp.gopen(rows=1, columns=3)
cfp.gpos(1)
cfp.con(h, colorbar_label_skip=2, title='Bilinear')
cfp.gpos(2)
cfp.con(i, colorbar_label_skip=2, title='Higher order patch recovery')
cfp.gpos(3)
cfp.con(deriv_i - deriv_h, blockfill=True, lines=False, colorbar_label_skip=2, title='Difference of derivati
ves')
cfp.gclose()
```



Regridding an integer field using the nearest neighbour method

In [19]:

```
# Read in ncas_data/regions_low_res.nc and inspect the field
f = cf.read_field('ncas_data/regions.nc')
print f
```

```
region field summary
--------------------
Data           : region(latitude(145), longitude(53))
Axes           : latitude(145) = [-90.0, ..., 90.0] degrees_north
               : longitude(53) = [-33.75, ..., 63.75] degrees_east
```

In [20]:

```
# Read in ncas_data/model_precip_DJF_means_low_res.nc and inspect the field
g = cf.read_field('ncas_data/model_precip_DJF_means_low_res.nc')
print g
```

```
long_name:precipitation field summary
-------------------------------------
Data           : long_name:precipitation(long_name:t(1), long_name:Surface(1), latitude(73), lo
ngitude(27)) mm/day
Cell methods   : long_name:t: mean
Axes           : long_name:t(1) = [1996-07-16T00:00:00Z] 360_day
               : long_name:Surface(1) = [0.0]
               : latitude(73) = [-90.0, ..., 90.0] degrees_north
               : longitude(27) = [-33.75, ..., 63.75] degrees_east
```

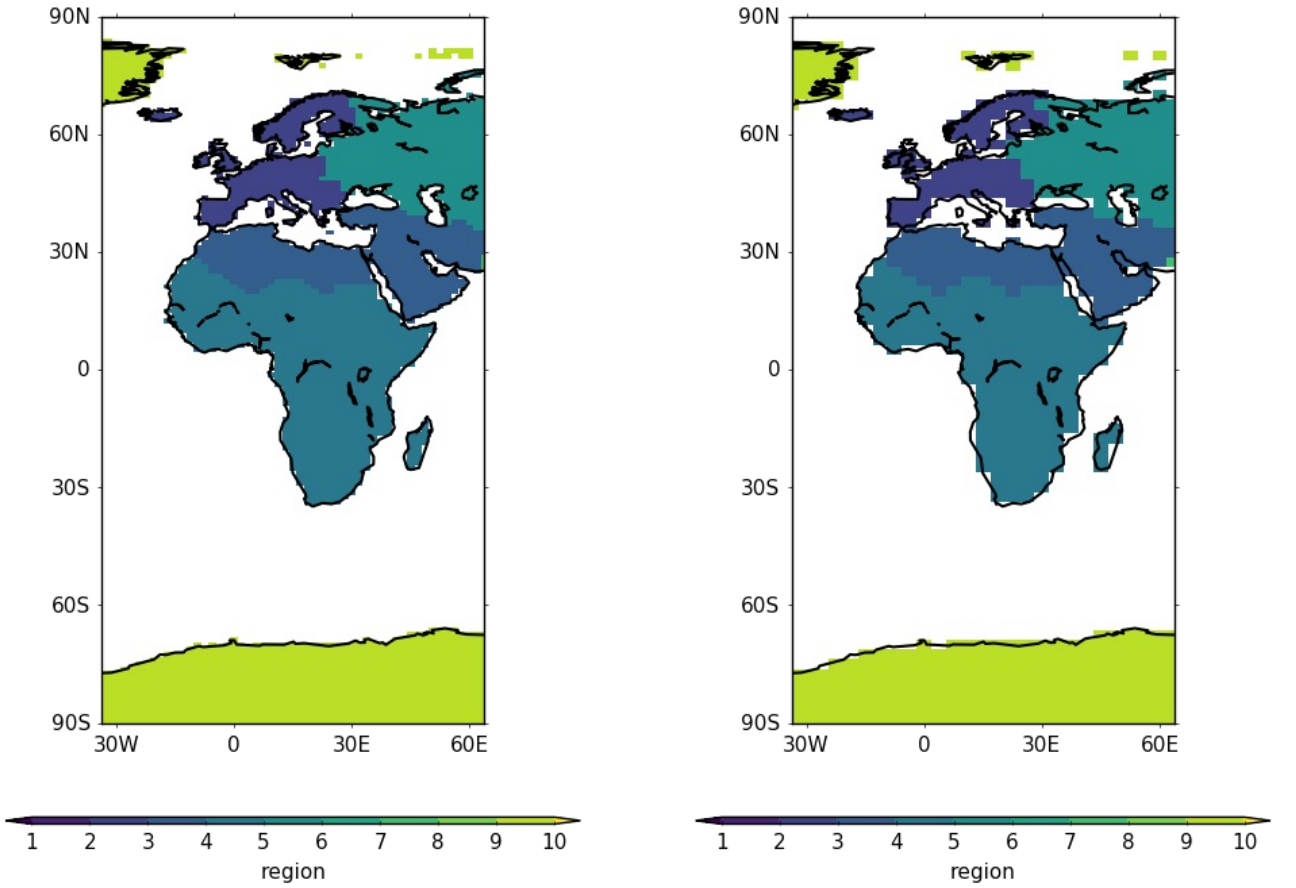
In [21]:

```
# Regrid regions to model grid using nearest source to destination regridding and inspect the result
h = f.regrids(g, method='nearest_stod')
print h
```

```
region field summary
--------------------
Data           : region(latitude(73), longitude(27))
Axes           : latitude(73) = [-90.0, ..., 90.0] degrees_north
               : longitude(27) = [-33.75, ..., 63.75] degrees_east
```

In [22]:

```
# Plot before and after
cfp.gopen(rows=1, columns=2)
cfp.levs(min=1, max=10, step=1)
cfp.gpos(1)
cfp.con(f, blockfill=True, lines=False)
cfp.gpos(2)
cfp.con(h, blockfill=True, lines=False)
cfp.gclose()
```



Cartesian regridding (regridc)

In [23]:

```
# Read in ncas_data/precip_1D_yearly.nc and inspect the field
f = cf.read_field('ncas_data/precip_1D_yearly.nc')
print f
```

```
long_name:precipitation field summary
-------------------------------------
Data           : long_name:precipitation(long_name:time(10), long_name:latitude(1), long_name:l
ongitude(1)) mm
Cell methods   : long_name:time: mean long_name:latitude: long_name:longitude: mean
Axes           : long_name:time(10) = [1981-07-02T00:00:00Z, ..., 1990-07-02T00:00:00Z] gregori
an
               : long_name:latitude(1) = [0.0] degrees_north
               : long_name:longitude(1) = [0.0] degrees_east
```

In [24]:

```
# Read in ncas_data/precip_1D_monthly.nc and inspect the field
g = cf.read_field('ncas_data/precip_1D_monthly.nc')
print g
```

```
long_name:precipitation field summary
-------------------------------------
Data           : long_name:precipitation(long_name:time(120), long_name:latitude(1), long_name:
longitude(1)) mm
Cell methods   : long_name:latitude: long_name:longitude: mean
Axes           : long_name:time(120) = [1981-01-16T00:00:00Z, ..., 1990-12-16T00:00:00Z] gregor
ian
               : long_name:latitude(1) = [0.0] degrees_north
               : long_name:longitude(1) = [0.0] degrees_east
```

In [25]:

```
# Regrid the first field to the grid of the second linearly and summarize the resulting field
h = f.regridc(g, axes='T', method='bilinear')
print h
```

```
long_name:precipitation field summary
-------------------------------------
Data           : long_name:precipitation(long_name:time(120), long_name:latitude(1), long_name:
longitude(1)) mm
Cell methods   : long_name:time: mean long_name:latitude: long_name:longitude: mean
Axes           : long_name:time(120) = [1981-01-16T00:00:00Z, ..., 1990-12-16T00:00:00Z] gregor
ian
               : long_name:latitude(1) = [0.0] degrees_north
               : long_name:longitude(1) = [0.0] degrees_east
```

In [26]:

```
# Plot before and after
cfp.gopen(rows=1, columns=2)
cfp.gpos(1)
cfp.lineplot(f, marker='o', color='red')
cfp.gpos(2)
cfp.lineplot(h, marker='o', color='blue')
cfp.gclose()
```