cf-python_regridding

March 26, 2018

1 cf-python Regridding

- **Based on the ESMPy interface to the Earth System Modeling Framework (ESMF) library**
- **Coordinate systems: Spherical (regrids) or Cartesian (regridc)**
- **Regridding methods:**
  - **First order conservative** - conserves integral of field with respect to area
  - **Bilinear** - linear interpolation in 1-3 dimensions (2D for sperical coordinates)
  - **Higher order patch recovery** - polynomial based so better values/derivatives
  - **Nearest source to destination** - useful for integer fields such as land use
  - **Nearest destination to source** - multiple source points can contribute to one destination point
- **Global or regional source/destination grids in any combination**
- **Handles grids with 2D latitudes and longitudes including curvilinear, rotated pole and some tripolar grids**
- **Handles masking of both source and destination grids**

1.1 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] gregoria
               : 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), lon
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] gregoria
               : 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()
```