

NASA Space Apps Challenge 2019: Playing with IRIS and Cartopy

Documentation

- <https://scitools.org.uk/iris/docs/latest/> (<https://scitools.org.uk/iris/docs/latest/>)
- <https://scitools.org.uk/cartopy/docs/latest/> (<https://scitools.org.uk/cartopy/docs/latest/>)

Import the libraries

```
In [297]: import cartopy.crs as ccrs
import cartopy.feature as cfeat
import cartopy.io.shapereader as shpreader
import iris
import iris.quickplot as qplt
import iris.analysis.cartography as icart
import matplotlib.pyplot as plt
import numpy as np
```

Sea levels today

```
In [298]: # https://sealevel.nasa.gov/data/data-search/?keyword=MEASURES-PROJECT&Collection[] = Ocean

# https://podaac-tools.jpl.nasa.gov/drive/files/allData/merged_alt/L4/cdr_grid/
ssh_grids_v1812_1992100212.nc
fname = "../Data/ssh_grids_v1812_1992100212.nc"
cubes = iris.load(fname)
ssh_1992 = cubes[1][0]

# https://podaac-tools.jpl.nasa.gov/drive/files/allData/merged_alt/L4/cdr_grid/
ssh_grids_v1812_2018113012.nc
fname = "../Data/ssh_grids_v1812_2018113012.nc"
cubes = iris.load(fname)
ssh_2018 = cubes[1][0]

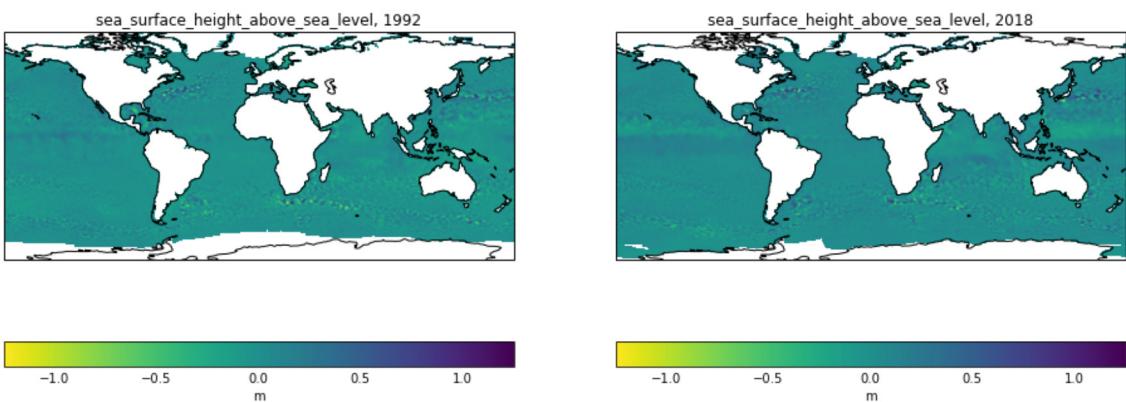
# Is 2018 data absolute, or relative to 1992?
# chg_2018_1992 = ssh_2018 - ssh_1992
```

```
In [299]: fig = plt.gcf()
fig.set_size_inches(16,8)

ax1 = fig.add_subplot(1,2,1)
qplt.pcolormesh(ssh_1992, cmap='viridis_r', vmin=-1.25, vmax=1.25)
ax1 = plt.gca()
ax1.coastlines()
plt.title('sea_surface_height_above_sea_level, 1992')

ax2 = fig.add_subplot(1,2,2)
qplt.pcolormesh(ssh_2018, cmap='viridis_r', vmin=-1.25, vmax=1.25)
ax2 = plt.gca()
ax2.coastlines()
plt.title('sea_surface_height_above_sea_level, 2018')

plt.show()
```



```
In [300]: # Set up our 3 locations
# We'll take sea level to be the average over a 2deg lat-long square

lat_res = 1
lon_res = 1

lon_loc = 'Thames Estuary 51.533229, 1.3472113'
lon_lat = 51.53
lon_lon = 1.35

syd_loc = 'Sydney coast -33.847794, 151.342051'
syd_lat = -33.85
syd_lon = 151.34

ny_loc = 'New Jersey Bight 40.379992, -73.5939137'
ny_lat = 40.38
ny_lon = 360.00 - 73.59
```

```
In [301]: # Set up the data table for sea levels at our 3 locations
```

```
sl_locs_dt = np.dtype([('Location', np.unicode_, 16),
                      ('2018', np.float64, 1), ('2099', np.float64, 1),
                      ('2199', np.float64, 1), ('2299', np.float64, 1)])
sl_locs = np.array([('Thames Estuary', 0.0, 0.0, 0.0, 0.0),
                    ('Sydney coast', 0.0, 0.0, 0.0, 0.0),
                    ('New Jersey Bight', 0.0, 0.0, 0.0, 0.0)], dtype=sl_locs_d
t)
```

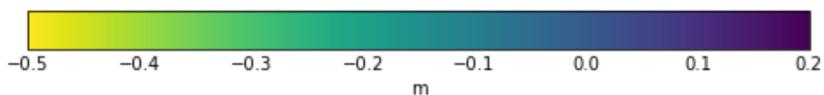
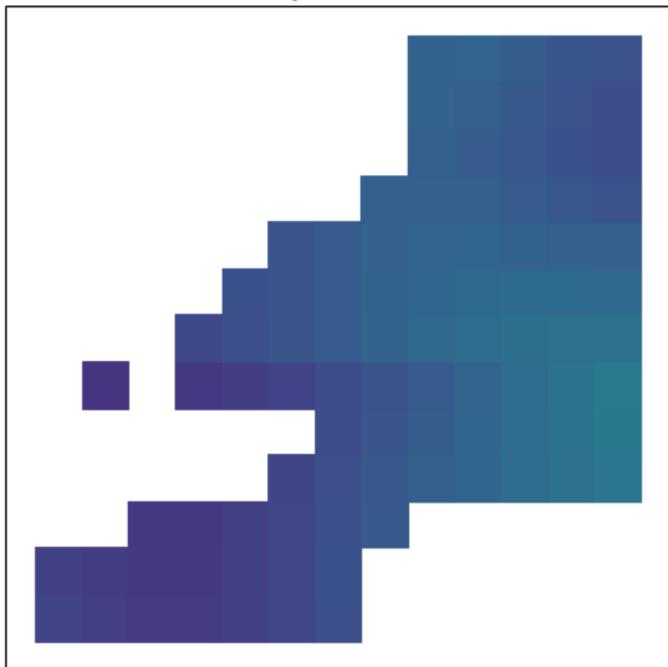
```
In [310]: fig = plt.gcf()
fig.set_size_inches(8,10)

# London
location_constraint = iris.Constraint(
    latitude = lambda cell: lon_lat - lat_res <= cell <= lon_lat + lat_res,
    longitude = lambda cell: lon_lon - lon_res <= cell <= lon_lon + lon_res)
location_cube = ssh_2018.extract(location_constraint)

grid_areas = icart.area_weights(location_cube)
sl_locs[0]['2018'] = location_cube.collapsed(['longitude', 'latitude'], iris.analysis.MEAN, weights=grid_areas).data

qplt.pcolormesh(location_cube, cmap='viridis_r', vmin=-0.5, vmax=0.2)
plt.title('sea_surface_height_above_sea_level, 2018\n'+lon_loc)
plt.show()
```

sea_surface_height_above_sea_level, 2018
Thames Estuary 51.533229,1.3472113



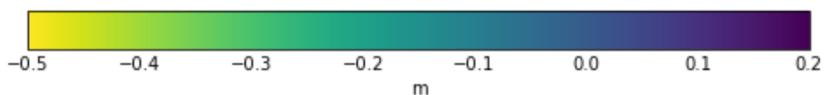
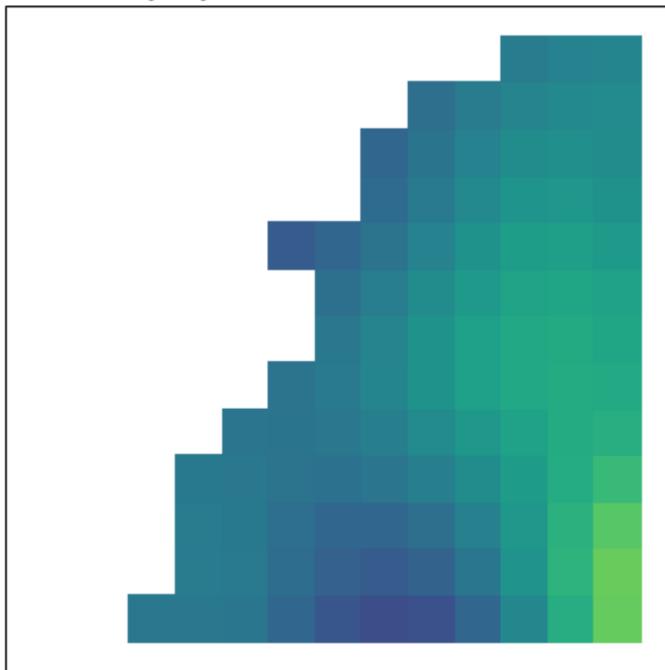
```
In [303]: fig = plt.gcf()
fig.set_size_inches(8,10)

# Sydney
location_constraint = iris.Constraint(
    latitude = lambda cell: syd_lat - lat_res <= cell <= syd_lat + lat_res,
    longitude = lambda cell: syd_lon - lon_res <= cell <= syd_lon + lon_res)
location_cube = ssh_2018.extract(location_constraint)

grid_areas = icart.area_weights(location_cube)
sl_locs[1]['2018'] = location_cube.collapsed(['longitude', 'latitude'], iris.analysis.MEAN, weights=grid_areas).data

qplt.pcolormesh(location_cube, cmap='viridis_r', vmin=-0.5, vmax=0.2)
plt.title('sea_surface_height_above_sea_level, 2018\n'+syd_loc)
plt.show()
```

sea_surface_height_above_sea_level, 2018
Sydney coast -33.847794, 151.342051

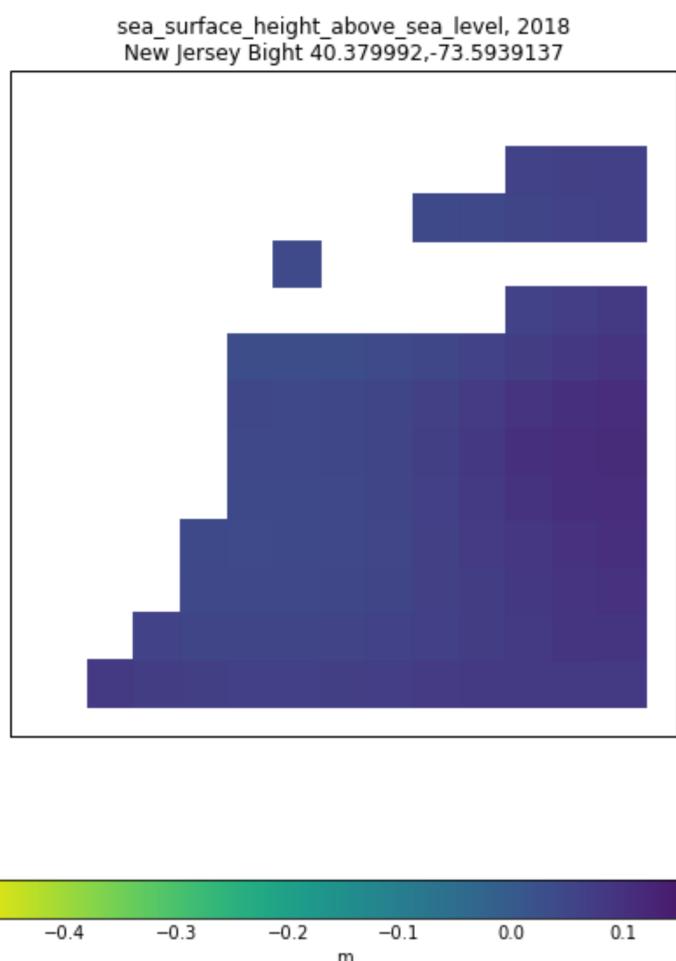


```
In [304]: fig = plt.gcf()
fig.set_size_inches(8,10)

# NY
location_constraint = iris.Constraint(
    latitude = lambda cell: ny_lat - lat_res <= cell <= ny_lat + lat_res,
    longitude = lambda cell: ny_lon - lon_res <= cell <= ny_lon + lon_res)
location_cube = ssh_2018.extract(location_constraint)

grid_areas = icart.area_weights(location_cube)
sl_locs[2]['2018'] = location_cube.collapsed(['longitude', 'latitude'], iris.analysis.MEAN, weights=grid_areas).data

qplt.pcolormesh(location_cube, cmap='viridis_r', vmin=-0.5, vmax=0.2)
plt.title('sea_surface_height_above_sea_level, 2018\n'+ny_loc)
plt.show()
```



Sea levels in the future

```
In [ ]: # https://esgf-node.llnl.gov/search/cmip5/
# Project = CMIP5
# Model = HADGEM3-ES
# Experiment Family = RCP
# Ensemble = r1i1p1
# Variable Long Name = Sea Surface Height Above Geoid (4171)

fname = "../Data/zos_Omon_HadGEM2-ES_rcp26_r1i1p1_209912-219911.nc"
cubes = iris.load(fname)
sl_2099_26 = cubes[0][0,:,:]

fname = "../Data/zos_Omon_HadGEM2-ES_rcp26_r1i1p1_219912-229911.nc"
cubes = iris.load(fname)
sl_2199_26 = cubes[0][0,:,:]

fname = "../Data/zos_Omon_HadGEM2-ES_rcp26_r1i1p1_229912-229912.nc"
cubes = iris.load(fname)
sl_2299_26 = cubes[0][0,:,:]

fname = "../Data/zos_Omon_HadGEM2-ES_rcp45_r1i1p1_209912-219911.nc"
cubes = iris.load(fname)
sl_2099_45 = cubes[0][0,:,:]

fname = "../Data/zos_Omon_HadGEM2-ES_rcp45_r1i1p1_219912-229911.nc"
cubes = iris.load(fname)
sl_2199_45 = cubes[0][0,:,:]

fname = "../Data/zos_Omon_HadGEM2-ES_rcp45_r1i1p1_229912-229912.nc"
cubes = iris.load(fname)
sl_2299_45 = cubes[0][0,:,:]

fname = "../Data/zos_Omon_HadGEM2-ES_rcp85_r1i1p1_209912-219911.nc"
cubes = iris.load(fname)
sl_2099_85 = cubes[0][0,:,:]

fname = "../Data/zos_Omon_HadGEM2-ES_rcp85_r1i1p1_219912-229911.nc"
cubes = iris.load(fname)
sl_2199_85 = cubes[0][0,:,:]

fname = "../Data/zos_Omon_HadGEM2-ES_rcp85_r1i1p1_229912-229912.nc"
cubes = iris.load(fname)
sl_2299_85 = cubes[0][0,:,:]
```

```
In [306]: fig = plt.gcf()
fig.set_size_inches(20,12)
fig.suptitle('Sea Surface Height, RCP2.6 in 2099 2199 2299\nRCP4.5\nRCP8.5', fontsize=16)

ax11 = fig.add_subplot(3,3,1)
qplt.pcolormesh(sl_2099_26, cmap='plasma_r', vmin=-1, vmax=6)
ax11 = plt.gca()
ax11.coastlines()

ax12 = fig.add_subplot(3,3,2)
qplt.pcolormesh(sl_2199_26, cmap='plasma_r', vmin=-1, vmax=6)
ax12 = plt.gca()
ax12.coastlines()

ax13 = fig.add_subplot(3,3,3)
qplt.pcolormesh(sl_2299_26, cmap='plasma_r', vmin=-1, vmax=6)
ax13 = plt.gca()
ax13.coastlines()

ax21 = fig.add_subplot(3,3,4)
qplt.pcolormesh(sl_2099_45, cmap='plasma_r', vmin=-1, vmax=6)
ax21 = plt.gca()
ax21.coastlines()

ax22 = fig.add_subplot(3,3,5)
qplt.pcolormesh(sl_2199_45, cmap='plasma_r', vmin=-1, vmax=6)
ax22 = plt.gca()
ax22.coastlines()

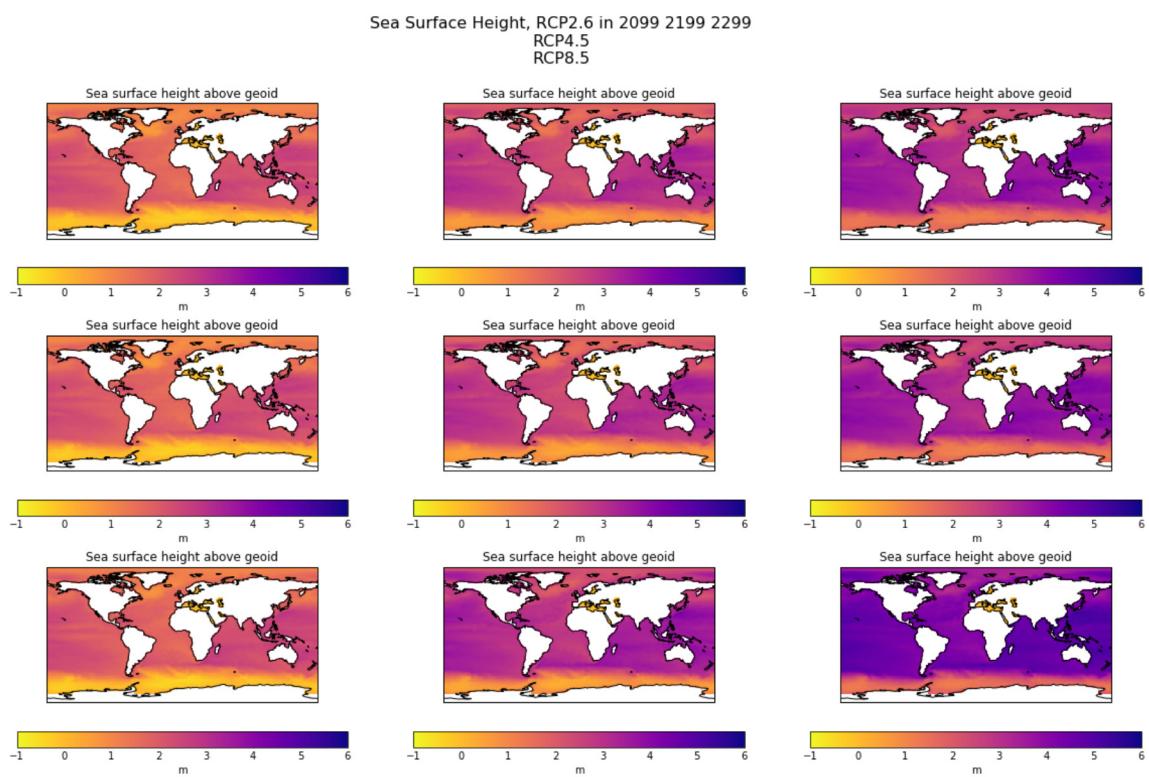
ax23 = fig.add_subplot(3,3,6)
qplt.pcolormesh(sl_2299_45, cmap='plasma_r', vmin=-1, vmax=6)
ax23 = plt.gca()
ax23.coastlines()

ax31 = fig.add_subplot(3,3,7)
qplt.pcolormesh(sl_2099_85, cmap='plasma_r', vmin=-1, vmax=6)
ax31 = plt.gca()
ax31.coastlines()

ax32 = fig.add_subplot(3,3,8)
qplt.pcolormesh(sl_2199_85, cmap='plasma_r', vmin=-1, vmax=6)
ax32 = plt.gca()
ax32.coastlines()

ax33 = fig.add_subplot(3,3,9)
qplt.pcolormesh(sl_2299_85, cmap='plasma_r', vmin=-1, vmax=6)
ax33 = plt.gca()
ax33.coastlines()

plt.show()
```



```
In [309]: # London
location_constraint = iris.Constraint(
    latitude = lambda cell: lon_lat - lat_res <= cell <= lon_lat + lat_res,
    longitude = lambda cell: lon_lon - lon_res <= cell <= lon_lon + lon_res)

location_cube = sl_2099_85.extract(location_constraint)
grid_areas = icart.area_weights(location_cube)
sl_locs[0]['2099'] = location_cube.collapsed(['longitude', 'latitude'], iris.analysis.MEAN, weights=grid_areas).data

location_cube = sl_2199_85.extract(location_constraint)
grid_areas = icart.area_weights(location_cube)
sl_locs[0]['2199'] = location_cube.collapsed(['longitude', 'latitude'], iris.analysis.MEAN, weights=grid_areas).data

location_cube = sl_2299_85.extract(location_constraint)
grid_areas = icart.area_weights(location_cube)
sl_locs[0]['2299'] = location_cube.collapsed(['longitude', 'latitude'], iris.analysis.MEAN, weights=grid_areas).data

# Sydney
location_constraint = iris.Constraint(
    latitude = lambda cell: syd_lat - lat_res <= cell <= syd_lat + lat_res,
    longitude = lambda cell: syd_lon - lon_res <= cell <= syd_lon + lon_res)

location_cube = sl_2099_85.extract(location_constraint)
grid_areas = icart.area_weights(location_cube)
sl_locs[1]['2099'] = location_cube.collapsed(['longitude', 'latitude'], iris.analysis.MEAN, weights=grid_areas).data

location_cube = sl_2199_85.extract(location_constraint)
grid_areas = icart.area_weights(location_cube)
sl_locs[1]['2199'] = location_cube.collapsed(['longitude', 'latitude'], iris.analysis.MEAN, weights=grid_areas).data

location_cube = sl_2299_85.extract(location_constraint)
grid_areas = icart.area_weights(location_cube)
sl_locs[1]['2299'] = location_cube.collapsed(['longitude', 'latitude'], iris.analysis.MEAN, weights=grid_areas).data

# NY
location_constraint = iris.Constraint(
    latitude = lambda cell: ny_lat - lat_res <= cell <= ny_lat + lat_res,
    longitude = lambda cell: ny_lon - lon_res <= cell <= ny_lon + lon_res)

location_cube = sl_2099_85.extract(location_constraint)
grid_areas = icart.area_weights(location_cube)
sl_locs[2]['2099'] = location_cube.collapsed(['longitude', 'latitude'], iris.analysis.MEAN, weights=grid_areas).data

location_cube = sl_2199_85.extract(location_constraint)
grid_areas = icart.area_weights(location_cube)
sl_locs[2]['2199'] = location_cube.collapsed(['longitude', 'latitude'], iris.analysis.MEAN, weights=grid_areas).data

location_cube = sl_2299_85.extract(location_constraint)
grid_areas = icart.area_weights(location_cube)
sl_locs[2]['2299'] = location_cube.collapsed(['longitude', 'latitude'], iris.analysis.MEAN, weights=grid_areas).data
```

```
In [308]: # Sea surface heights in 2018, 2099 2199 2299 under RCP8.5 'worst case' scenario  
o  
sl_locs
```

```
Out[308]: array([('Thames Estuary', 0.00768565, 1.71223205, 2.9450562, 4.14300942),  
 ('Sydney coast', -0.12476984, 2.25440075, 3.31024461, 4.64952831),  
 ('New Jersey Bight', 0.06914707, 1.19410283, 2.32413902, 3.82559856)],  
 dtype=[('Location', '
```

That's all folks!