Appendix

NetCDF data interpolation example code

```
@author: perepely
"""
#pip install netCDF4 #This package is necessary to install the netCDF4
#pip install cartopy # This function is required if cartopy program is not intalled before
#================================================================================
==
import netCDF4
from netCDF4 import Dataset
import os
import numpy as np
import cartopy.crs as ccrs
import cartopy.feature as cfeature
from cartopy.mpl.gridliner import LONGITUDE_FORMATTER, LATITUDE_FORMATTER
import matplotlib
import matplotlib.pyplot as plt
#plt.style.use("ggplot")
#================================================================================
==
nc_path                                                                    =
os.path.abspath(r'C:\Users\perepely\Models\Raw_data\Copernicus\Fresh\currents_hourly.nc'
);
dataset = Dataset(nc_path)
#Interrogate netCDF file
print(dataset.file_format)
print(dataset.dimensions.keys()) #dimensions
print(dataset.dimensions['time'])
print(dataset.variables.keys()) #variables
print(dataset.variables['vo'])
print(dataset.variables['uo'])
print(dataset.Conventions) # Get conventions attribute

attr=dataset.ncattrs() #find all NetCDF global attributes
for attr in dataset.ncattrs():
   print(attr, '=', getattr(dataset, attr))
#================================================================================
==
timestep=100;
fh = Dataset(nc_path, mode='r')
time=fh.variables['time']
jd = netCDF4.num2date(time[:],time.units)
lons = fh.variables['lon'][:]
lons = lons.data[:]
lats = fh.variables['lat'][:]
lats = lats.data[:]
North_vel = fh.variables['vo'][:]
```

```
North_vel = North_vel.data[:]
East_vel = fh.variables['uo'][:]
East_vel = East_vel.data[:]
fh.close()
#=========================================================================
==
# Load the Pandas libraries with alias 'pd'
import pandas as pd
# Read data from file 'filename.csv'
# (in the same directory that your python process is based)
# Control delimiters, rows, column names with read_csv (see later)
data                                                                    =
pd.read_csv(r'C:\Users\perepely\Models\Delft_FM\Boundary_Conditions\Long_boundary\Ma
riya_long_border2.csv')
# Preview the first 5 lines of the loaded data
data.head()

#=========================================================================
==
#Gives the correct id number of lat and lon of the closest point!
lat_idx_lis = []
lon_idx_lis = []
#from numpy import absolute as abs
for i in range(len(data['X'])):
    loni = data["X"][i]
    lati = data["Y"][i]
    my_point = {'name': 'My_point', 'lat': lati, 'lon': loni}

    # Find the nearest latitude and longitude
    lat_idx = np.abs(lats - my_point['lat']).argmin()
    lat_idx_lis.append(lat_idx)
    lon_idx = np.abs(lons - my_point['lon']).argmin()
    lon_idx_lis.append(lon_idx)

values_lat = lats[lat_idx_lis]
values_lon = lons[lon_idx_lis]

#iy = lat_idx
#ix = lon_idx
for i  in range(len(values_lat)):
    print ('Exact Location lat-lon:', data['Y'][i],data['X'][i])
    print ('Closest lat-lon:', values_lat[i], values_lon[i])

#gives all timesteps and all depth layers

NORTH_FINAL = []
for i in range(len(lon_idx_lis)):
    North_vel_dataserie = North_vel[:,:,lat_idx_lis,lon_idx_lis]
```

```
   NORTH_FINAL.append(North_vel_dataserie)
   ###########change
EAST_FINAL = []
for i in range(len(lon_idx_lis)):
   East_vel_dataserie = East_vel[:,:,lat_idx_lis,lon_idx_lis]
   EAST_FINAL.append(East_vel_dataserie)
#================================================================================
==
#PLOT map of North_vel
# Get some parameters for the Stereographic Projection
#Modify the range of i to obtain plots of several depths at an specific timestep
#Modify the number of plots that you require depending on the layer
for i in range(0,9):
   North_vel_plot=North_vel[timestep,i,:,:]
#Plot
   matplotlib.rcParams['figure.figsize'] = (10,10)
   proj=ccrs.Mercator()
   m = plt.axes(projection=proj)
# Put a background image on for nice sea rendering.
   m.stock_img()
   m.coastlines(resolution='110m')
   m.add_feature(cfeature.BORDERS)
   gl=m.gridlines(crs=ccrs.PlateCarree(), draw_labels=True,
            linewidth=2, color='gray', alpha=0.5, linestyle='--')
   gl.xformatter = LONGITUDE_FORMATTER
   gl.yformatter = LATITUDE_FORMATTER
   gl.xlabels_top = False
   gl.ylabels_right = False
#Plot data for North_vel
   plt.contourf(lons, lats, North_vel_plot, 60,
            transform=ccrs.PlateCarree())

   # Add Colorbar
   cbar = plt.colorbar()
   cbar.set_label(dataset.variables['vo'].units)
# Add Title
   plt.title('North Current Velocity Movement')
   plt.show()
#================================================================================
==
#PLOT of East_vel
# Get some parameters for the Stereographic Projection
#Modify the range of i to obtain plots of several depths at an specific timestep
#Modify the number of plots that you require depending on the layer
for i in range(0,9):
   East_vel_plot=East_vel[timestep,i,:,:]
#Plot
   matplotlib.rcParams['figure.figsize'] = (10,10)
```