Extracting Global Warming Levels from the UK Climate Projections

May 21, 2021

1 Abstract

This notebook sets out the methodology for extracting a 21-year time slice centred on the year at which members of UKCP Global (60km) reach a certain global warming level. It demonstrates how you can extract data from the UK Climate Projections' suite of products including UKCP Regional (12km) for locations in the UK (and Europe) as well as UKCP Global (60km) for locations across the globe.

Author: Fai Fung and Alex Chamberlain-Clay

Organisation: Met Office

Language: Python 3.6.8 Required libraries: datetime, numpy, pathlib, iris, scipy

2 Contents

3 Introduction

The Bank of England benchmark scenarios include a set of data based on the latest UK Climate Projections (Lowe et al, 2018). One set of benchmark data for the UK is provided as global warming levels: these are based on UKCP Regional (12km) and are a set of 12 regional climate model simulations at a spatial resolution of 12 km and include daily data. These 12 regional climate model simulations were driven by 12 global climate models (GCMs) with the same model setup; the GCMs are a subset of UKCP Global (60km) .

The simulations were run for a high emission scenario (RCP 8.5). Global warming level data can be extracted from UKCP Regional by finding the year at which global mean temperatures are reached and including 10 years of data either side resulting in 21 years of data. Global mean temperatures are calculated from the subset of UKCP Global (60km). Further details about these datasets can be found in Lowe et al (2018) and Fung et al (2018).

The following Python 3 code lays out the steps for extracting the 21-year time-slice for:

- UKCP Regional (12km) - this is only required if you're planning to use data for UK locations. This requires sections 6 and 7 only.
- UKCP Global (60km) - this is only required if you're planning to use data for non-UK locations. You requires sections 6 and 8 only.

Instructions on how to download these datasets are in the next section.

4 Data download instructions

To extract daily time series underpinning the benchmark scenarios, you will need to download:

- Global mean surface temperatures from UKCP Global (60km) - see here.
- Daily data from UKCP Regional (12km) - see here.

All data is subject to the Open Government Licence and is available from the CEDA Data Archive. You can access them via

- The CEDA Catalogue. You can peruse and download datasets directly on the CEDA Catalogue web pages.
- The CEDA FTP site using the file transfer protocol (FTP). For further details see STFC-CEDA help pages.
- The OpenDAP API. Details are on the STFC-CEDA web pages.

Note that all of options require registration which you can do on the CEDA registration page.

5 References

Lowe et al (2018) UKCP18 Science Overview Report, Met Office. Available at here.

Fung et al (2018) UKCP Guidance:Data availability, access and formats, Met Office. Available at here.

6 Identifying global warming 21-year time-slice

Import libraries

[8]:
```
# standard libraries
import datetime
import numpy as np
from pathlib import Path
```

```
# scientific libraries
import iris
from scipy.signal import savgol_filter
```

The following values are the adjustments to account for UKCP Global (60km) only including data for 1900-2100. pdwarming adjusts 1980-2000 anomalies to preindustrial values (1850-1900) based on HadCRUT5.

[18]:
```
pdymin = 1981  # baseline start year
pdymax = 2000  # baseline end year
pdwarming = 0.606  # adjustment from HadCRUT5 based on difference between␣
↪1850-1900 and 1980-2000 average
```

Set the global warming level of interest, here we set it at 3.1 degrees Celsius above preindustrial levels

[19]:
```
Tlims = [3.1]
```

Set location of UKCP Global (60km) mean surface air temperature file

[20]:
```
file_gmst = Path('tas_rcp85_land-gcm_global_60km_01_mon_189912-209911.nc')
```

Find the range of years for the global warming level for one ensemble member. Note that there are 12 members of UKCP Regional.

Load one of the corresponding members of UKCP Global (60km) available on the CEDA Archive (e.g. https://data.ceda.ac.uk/badc/ukcp18/data/land-gcm/global/60km/rcp85/01/tas/mon/latest. Note that the UKCP Regional (12km) members include ensemble member ids of 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 and 15 which correspond to the same ensemble member id of UKCP Global (60km).

Load file using Iris

[21]:
```
gcm = iris.load_cube(str(file_gmst), iris.Constraint(ensemble_member=1))
```

Calculate mean global surface temperature

[22]:
```
grid_areas = iris.analysis.cartography.area_weights(gcm)
gcmts = gcm.collapsed(['longitude', 'latitude'],iris.analysis.MEAN,␣
↪weights=grid_areas)
```

Calculate annual mean tempeatures

[23]:
```
Tdec = np.zeros([200]) # dictionary for annual means (dec to november)

for i in range(len(Tdec)):
    Tdec[i] = np.mean(gcmts.data[i*12 : 12+i*12])

Ydec = 1899 + np.asarray(range(200))  # years for dec-nov annual mean
```

```
Tcal = np.zeros([199]) # dictionary for annual means (jan to dec)
for i in range(len(Tcal)):
    Tcal[i] = np.mean(gcmts.data[1 + i * 12 : 1 + 12 + i * 12])
Ycal = 1900 + np.asarray(range(199)) # years for calendar annual means
```

Smooth the data of year-to-year natural variability

[24]:
```
window = 21  # years of window
poly = 3  # polynomial fitting

# smooth the data
Tcals = savgol_filter(Tcal, window, poly)
Tdecs = savgol_filter(Tdec, window, poly)

# only include years during baseline period
pdID = np.where((Ycal >= pdymin) & (Ycal <= pdymax))
```

Adjust absolute warming using HadCRUT5

[25]:
```
Tann = Tcals
TannBC = Tcals - np.mean(Tcals[pdID]) + pdwarming
```

Find start and end year of time slice centreed on global warming level

[26]:
```
# find year that global warming level is crossed
years = np.interp(Tlims, TannBC, Ycal).round()

print('GWL:', Tlims[0])
print('Year at which GWL is crossed:', int(years[0]))
```

```
GWL: 3.1
Year at which GWL is crossed: 2048
```

7 Extracting global warming levels data from UKCP Regional (12km) for UK locations

Set location of corresponding UKCP Regional file which can be downloaded from the CEDA Archive (e.g. https://data.ceda.ac.uk/badc/ukcp18/data/land-rcm/uk/12km/rcp85/01/tas/day/latest)

[27]:
```
file_regional = Path('land-rcm')
filelist_regional = sorted(file_regional.rglob('*.nc'))
filelist_regional
```

[27]:
```
[PosixPath('land-rcm/tas_rcp85_land-rcm_uk_12km_01_day_19801201-19901130.nc'),
 PosixPath('land-rcm/tas_rcp85_land-rcm_uk_12km_01_day_19901201-20001130.nc'),
 PosixPath('land-rcm/tas_rcp85_land-rcm_uk_12km_01_day_20001201-20101130.nc'),
 PosixPath('land-rcm/tas_rcp85_land-rcm_uk_12km_01_day_20101201-20201130.nc'),
```

```
 PosixPath('land-rcm/tas_rcp85_land-rcm_uk_12km_01_day_20201201-20301130.nc'),
 PosixPath('land-rcm/tas_rcp85_land-rcm_uk_12km_01_day_20301201-20401130.nc'),
 PosixPath('land-rcm/tas_rcp85_land-rcm_uk_12km_01_day_20401201-20501130.nc'),
 PosixPath('land-rcm/tas_rcp85_land-rcm_uk_12km_01_day_20501201-20601130.nc'),
 PosixPath('land-rcm/tas_rcp85_land-rcm_uk_12km_01_day_20601201-20701130.nc'),
 PosixPath('land-rcm/tas_rcp85_land-rcm_uk_12km_01_day_20701201-20801130.nc')]
```

Load UKCP Regional, RCP 8.5 simulation files

[28]:
```
cubelist = iris.cube.CubeList()
for filename in filelist_regional:
    cube_load = iris.load_cube(str(filename))
    cubelist.append(cube_load)
```

Create one data cube

[29]:
```
cube = cubelist.concatenate_cube()
```

Extract the time slice associated with the GWL from the corresponding UKCP Global member

Set the window

[33]:
```
window_size = 10  # size each side of centerpoint - total years is 2 x␣
↪window_size + 1
maxyear = years[0] + window_size
minyear = years[0] - window_size
constraint_year = iris.Constraint(year=lambda cell: maxyear >= cell >= minyear)

# check that the time slice does not exceed simulation years, i.e. not beyond␣
↪2080
if maxyear > 2080:
    maxyear = 2080
```

Extract the required 21-year time slice

[34]:
```
cube_target_gwl = cube.extract(constraint_year)
```

Save file for future use

[35]:
```
file_out = 'tas_gwl3.1_land-rcm_uk_12km_01_day.nc'
iris.save(cube_target_gwl, file_out, unlimited_dimensions=[],
          local_keys=('plot_label', 'label_units', 'description', 'level'),
          netcdf_format='NETCDF4_CLASSIC', fill_value=1e20)
```

8 Extracting global warming levels data from UKCP Global (60km) for locations around the globe

Set location of corresponding UKCP Global file which can be downloaded from the CEDA Archive (e.g. for precipitation https://data.ceda.ac.uk/badc/ukcp18/data/land-gcm/global/60km/rcp85/01/pr/day/latest)

[38]:
```
file_global = Path('land-gcm')
filelist_global = sorted(file_global.rglob('*.nc'))
filelist_global
```

[38]:
```
[PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_18991201-19091130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_19091201-19191130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_19191201-19291130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_19291201-19391130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_19391201-19491130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_19491201-19591130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_19591201-19691130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_19691201-19791130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_19791201-19891130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_19891201-19991130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_19991201-20091130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_20091201-20191130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_20191201-20291130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_20291201-20391130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_20391201-20491130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_20491201-20591130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_20591201-20691130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_20691201-20791130.nc'),
```

```
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_20791201-20891130.nc'),
 PosixPath('land-gcm/pr_rcp85_land-
gcm_global_60km_01_day_20891201-20991130.nc')]
```

Load UKCP Global, RCP 8.5 simulation files

[39]:
```
cubelist = iris.cube.CubeList()
for filename in filelist_global:
    cube_load = iris.load_cube(str(filename))
    cube_load.attributes['creation_date'] = []
    cubelist.append(cube_load)
```

Create one data cube

[40]:
```
cube = cubelist.concatenate_cube()
```

Extract the time slice associated with the GWL from the corresponding UKCP Regional member

Set the window

[41]:
```
window_size = 10  # size each side of centerpoint - total years is 2 x␣
↪window_size + 1
maxyear = years[0] + window_size
minyear = years[0] - window_size
constraint_year = iris.Constraint(year=lambda cell: maxyear >= cell >= minyear)

# check that the time slice does not exceed simulation years, i.e. not beyond␣
↪2080
if maxyear > 2080:
    maxyear = 2080
```

Extract the required 21-year time slice

[42]:
```
cube_target_gwl = cube.extract(constraint_year)
```

Extract region of interest, e.g. New York State covering 40.5-45.017°N and 71.85°-79.767°W

[66]:
```
constraint_latitudes = iris.Constraint(latitude=lambda cell: 40.5 < cell.point␣
↪< 45.017)
constraint_longitudes = iris.Constraint(longitude=lambda cell:  -79.767 < cell.
↪point < -71.85)
cube_region_gwl = cube_target_gwl.extract(constraint_latitudes &␣
↪constraint_longitudes)
```

Save file for future use

[68]:
```
file_out = 'tas_gwl3.1_land-gcm_nystate_60km_01_day.nc'
iris.save(cube_region_gwl, file_out, unlimited_dimensions=[],
```

```
        local_keys=('plot_label', 'label_units', 'description', 'level'),
        netcdf_format='NETCDF4_CLASSIC', fill_value=1e20)
```