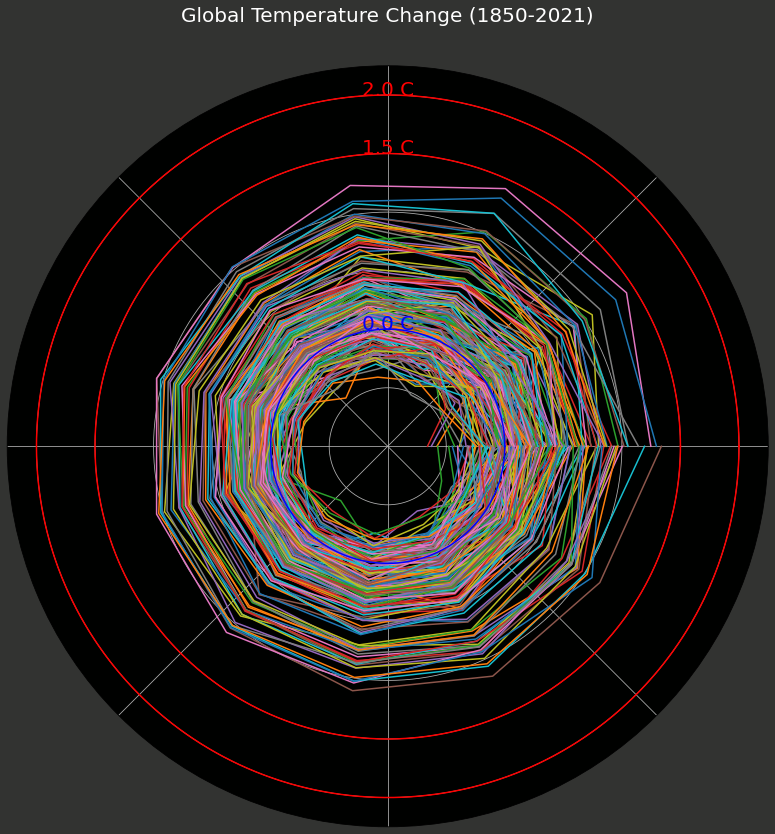
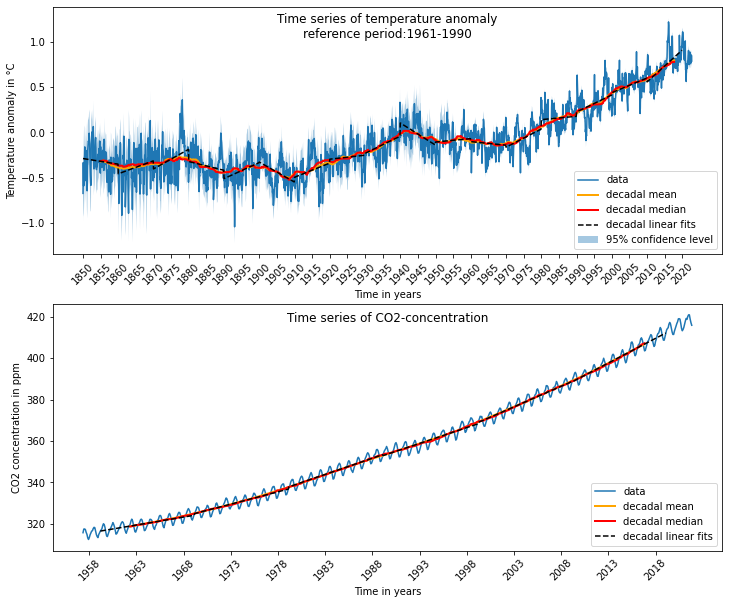
Physics of Climate Exercise 1

Jakob Schöll, Livia Eichmeyer, Florian Dznoff, Simeon Kraus





Annual grow rate of temperature in the decades

(obtained from decadal linear fits)

1850s: -0.006184 ppm/yr

1860s: 0.014405 ppm/yr

1870s: 0.021828 ppm/yr

1880s: -0.009696 ppm/yr

1890s: 0.017780 ppm/yr

1900s: -0.022225 ppm/yr

1910s: 0.015747 ppm/yr

1920s: 0.004409 ppm/yr

1930s: 0.019005 ppm/yr

1940s: -0.024541 ppm/yr

1950s: 0.003742 ppm/yr

1960s: -0.003890 ppm/yr

1970s: 0.020147 ppm/yr

1980s: 0.003682 ppm/yr

1990s: 0.018816 ppm/yr

2000s: 0.016271 ppm/yr

2010s: 0.035353 ppm/yr

Annual grow rate of CO2 concentration in the decades

(obtained from decadal linear fits)

1960s: 0.767081 °C/yr

1970s: 1.192464 °C/yr

1980s: 1.559946 °C/yr

1990s: 1.525992 °C/yr

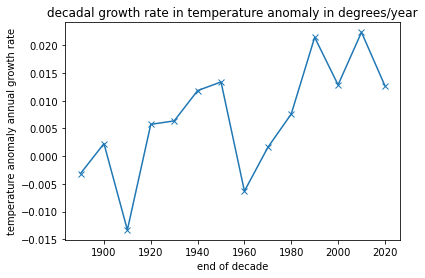
2000s: 1.989682 °C/yr

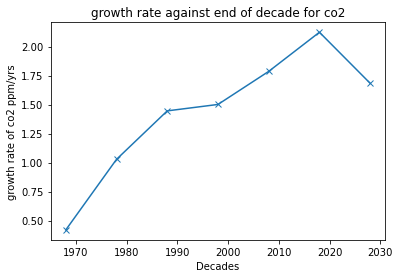
2010s: 2.378256 °C/yr

Temperature prediction for year 2050: 1.5 °C higher than average 1961-1990

(using quadratic fit for the data from 1970 on)

The growth rates of both temperature and CO2 are generally increasing with time:





Thermodynamic equilibrium of Earth Surface Atmosphere – as both temperature and CO2 are growing at increasing rates, thermodynamic equilibrium is shifting to higher mean temperatures. Increase in CO2 will augment greenhouse effect, creating a feedback effect that further increases warming.

E\_a changes – emissivity of atmosphere increases due to more trapping.

Code:

#!/usr/bin/env python

# coding: utf-8

# In[9]:

import matplotlib.pyplot as plt

import numpy as np

from scipy.optimize import curve\_fit

import pandas as pd

# In[3]:

def linear(x, a, b):

return a\*x + b

def quadratic(x,a,b,c):

return a\*x\*\*2 + b\*x + c

# # Data analysis

# In[5]:

#Load temperature data

t\_mean, t\_low\_conf, t\_up\_conf = np.loadtxt("HadCRUT.5.0.1.0.analysis.summary\_series.global.monthly.csv",

unpack=True, delimiter=",", skiprows=1, usecols=(1,2,3))

t\_date = np.loadtxt("HadCRUT.5.0.1.0.analysis.summary\_series.global.monthly.csv",

unpack=True, delimiter=",", skiprows=1, usecols=0, dtype=str)

#Unpack datestring

t\_year = np.array([])

t\_month = np.array([])

for timestr in t\_date:

t\_year = np.append(t\_year, int(timestr.split("-")[0]))

t\_month = np.append(t\_month, int(timestr.split("-")[1]))

#Decadal mean and median

t\_dec\_mean = np.array([])

t\_dec\_median = np.array([])

for i in range(0, len(t\_mean[120:])):

t\_dec\_mean = np.append(t\_dec\_mean, np.mean(t\_mean[i:120+i]))

t\_dec\_median = np.append(t\_dec\_median, np.median(t\_mean[i:120+i]))

#Fitting (linear fits of decades)

t\_popt\_array = np.array([])

t\_pcov\_array = np.array([])

t\_fit = np.array([])

for d in range(185,202): #from 1850s to 2010s

decade = d\*10

mask = np.logical\_and(t\_year>=decade, t\_year<(decade+10))

popt, pcov = curve\_fit(linear, np.linspace(decade, decade+10, 120, endpoint=False),

t\_mean[mask]) #Parameter a is then in °C/year

t\_popt\_array = np.append(t\_popt\_array, popt)

t\_pcov\_array = np.append(t\_pcov\_array, pcov)

t\_fit = np.append(t\_fit, linear(np.linspace(decade, decade+10, 120, endpoint=False), \*popt))

#Load CO2 data

c\_mean = np.loadtxt("co2\_mm\_mlo.txt", usecols=3)

c\_year, c\_month = np.loadtxt("co2\_mm\_mlo.txt", usecols=(0,1), unpack=True, dtype=int)

#Create datestring

c\_date = np.array([])

for i in range(0, len(c\_mean)):

c\_date = np.append(c\_date, "{0:4d}-{1:02d}".format(c\_year[i], c\_month[i]))

#Decadal mean and median

c\_dec\_mean = np.array([])

c\_dec\_median = np.array([])

for i in range(0, len(c\_mean[120:])):

c\_dec\_mean = np.append(c\_dec\_mean, np.mean(c\_mean[i:120+i]))

c\_dec\_median = np.append(c\_dec\_median, np.median(c\_mean[i:120+i]))

#Fitting (linear fits of decades)

c\_popt\_array = np.array([])

c\_pcov\_array = np.array([])

c\_fit = np.array([])

for d in range(196,202): #from 1960s-2010s

decade = d\*10

mask = np.logical\_and(c\_year>=decade, c\_year<decade+10)

popt, pcov = curve\_fit(linear, np.linspace(decade, decade+10, 120, endpoint=False),

c\_mean[mask]) #Parameter a is then in °C/year

c\_popt\_array = np.append(c\_popt\_array, popt)

c\_pcov\_array = np.append(c\_pcov\_array, pcov)

c\_fit = np.append(c\_fit, linear(np.linspace(decade, decade+10, 120, endpoint=False), \*popt))

#Print out grow rates

#Temperature

print("\nAnnual grow rate of temperature in the decades")

print("(obtained from decadal linear fits)")

dec = 1850

for a in t\_popt\_array[::2]:

print("{0:d}s: {1:9f} ppm/yr".format(dec, a))

dec=dec+10

#CO2

print("\nAnnual grow rate of CO2 concentration in the decades")

print("(obtained from decadal linear fits)")

dec = 1960

for a in c\_popt\_array[::2]:

print("{0:d}s: {1:9f} °C/yr".format(dec, a))

dec=dec+10

# # Plots

# In[6]:

#Plot

fig = plt.figure(figsize=(12,10))

ax = fig.add\_subplot(211)

x = np.linspace(0,len(t\_mean), len(t\_mean))

ax.set\_xticks(x[::60])

ax.set\_xticklabels(t\_year[::60].astype(int), rotation=45)

ax.plot(x, t\_mean, label="data")

ax.fill\_between(x, t\_up\_conf, t\_low\_conf, alpha=0.4, label="95% confidence level")

ax.plot(x[60:-60], t\_dec\_mean, linewidth=2, color="orange", label="decadal mean")

ax.plot(x[60:-60], t\_dec\_median, linewidth=2, color="red", label="decadal median")

#decadal mean/median plotted at time t is mean/median of period (t-5yr, t+5yr)

ax.plot(x[np.logical\_and(t\_year>=1850, t\_year<2020)], t\_fit, ls="--", color="black", label="decadal linear fits") #Fits

ax.set\_title("Time series of temperature anomaly\nreference period:1961-1990", y=0.85)

ax.set\_ylabel("Temperature anomaly in °C")

ax.set\_xlabel("Time in years")

ax.legend(loc="lower right")

#Plot

ax2 = fig.add\_subplot(212)

x = np.linspace(0,len(c\_mean), len(c\_mean))

ax2.set\_xticks(x[8::60])

ax2.set\_xticklabels(c\_year[8::60], rotation=45)

ax2.plot(x, c\_mean, label="data")

ax2.plot(x[60:-60], c\_dec\_mean, linewidth=2, color="orange", label="decadal mean")

ax2.plot(x[60:-60], c\_dec\_median, linewidth=2, color="red", label="decadal median")

#decadal mean/median plotted at time t is mean/median of period (t-5yr, t+5yr)

ax2.plot(x[np.logical\_and(c\_year>=1960, c\_year<2020)], c\_fit, ls="--", color="black", label="decadal linear fits") #Fits

ax2.set\_title("Time series of CO2-concentration", y=0.9)

ax2.set\_ylabel("CO2 concentration in ppm")

ax2.set\_xlabel("Time in years")

ax2.legend(loc="lower right")

# Temperature: Decaddal mean (and median) show an increasing trend (since the ~1970s), the slope is becoming steeper.

#

#

# CO2: Decadal mean (and median) of CO2 show an increasing trend

# ### C

# Increasing temperatures imply that a thermodynamic equilibrium does not exist anymore.

# ### D

# In[8]:

#Quadratic Fit to predict 2050

#We do a quadratic fit for all data from 1970 on

#Temperature

mask = t\_year >= 1970

t\_predict\_popt, t\_predict\_pcov = curve\_fit(quadratic, t\_year[mask] + t\_month[mask]/12, t\_mean[mask])

print("\nTemperature prediction for year 2050: {0:0.1f} °C higher than average 1961-1990".format(quadratic(2050, \*t\_predict\_popt)))

print("(using quadratic fit for the data from 1970 on)")

# ### E

# $$T\_{s}=(\frac{S\_0(1-A)}{2 \sigma})^{\frac{1}{4}}\cdot \frac{1}{(2- \varepsilon\_{a})}^{\frac{1}{4}}\\

# \frac{dT\_{s}}{d \varepsilon\_{a}}= (\frac{S\_0(1-A)}{2 \sigma})^{\frac{1}{4}}\cdot \frac{1}{4(2- \varepsilon\_{a})^{\frac{5}{4}}}= \frac{1}{4}\cdot \frac{T}{ \varepsilon\_{a}}\\

# \Delta T \approx dT,\ \Delta \varepsilon\_{a}\approx \varepsilon\_{a}\\

# \Rightarrow \frac{ \Delta T}{ \Delta \varepsilon\_{a}}=\frac{1}{4}\frac{T}{ \varepsilon}\\

# \Rightarrow \frac{ \Delta T}{T}= \frac{1}{4} \frac{ \Delta \varepsilon\_{a}}{ \varepsilon\_{a}}

# $$

#

# S doesn't change significantly. A might change because of melting Ice, but this can't be the driving factor. With increasing $CO\_2$ levels the emissivity $\epsilon$ increases, which leads to the observed change in temperature.

# In[10]:

#Extra Temperature Graph

hadcrut = pd.read\_csv(

"monthlyclimatedata.txt",

delim\_whitespace=True,

usecols=[0, 1],

header=None)

hadcrut['year'] = hadcrut.iloc[:, 0].apply(lambda x: x.split("/")[0]).astype(int)

hadcrut['month'] = hadcrut.iloc[:, 0].apply(lambda x: x.split("/")[1]).astype(int)

hadcrut = hadcrut.rename(columns={1: "value"})

hadcrut = hadcrut.iloc[:, 1:]

# In[12]:

hadcrut = hadcrut.set\_index(['year', 'month'])

hadcrut -= hadcrut.loc[1850:1900].mean()

hadcrut = hadcrut.reset\_index()

# In[13]:

fig = plt.figure(figsize=(14,14))

ax1 = plt.subplot(111, projection='polar')

ax1.axes.get\_yaxis().set\_ticklabels([])

ax1.axes.get\_xaxis().set\_ticklabels([])

fig.set\_facecolor("#323331")

ax1.set\_ylim(0, 3.25)

theta = np.linspace(0, 2\*np.pi, 12)

ax1.set\_title("Global Temperature Change (1850-2021)", color='white', fontdict={'fontsize': 20})

ax1.set\_facecolor('#000100')

years = hadcrut['year'].unique()

for year in years:

r = hadcrut[hadcrut['year'] == year]['value'] + 1

# ax1.text(0,0, str(year), color='white', size=30, ha='center')

ax1.plot(theta, r)

#Temperature rings

full\_circle\_thetas = np.linspace(0, 2\*np.pi, 1000)

blue\_line\_one\_radii = [1.0]\*1000

red\_line\_one\_radii = [2.5]\*1000

red\_line\_two\_radii = [3.0]\*1000

ax1.plot(full\_circle\_thetas, blue\_line\_one\_radii, c='blue')

ax1.plot(full\_circle\_thetas, red\_line\_one\_radii, c='red')

ax1.plot(full\_circle\_thetas, red\_line\_two\_radii, c='red')

ax1.text(np.pi/2, 1.0, "0.0 C", color="blue", ha='center', fontdict={'fontsize': 20})

ax1.text(np.pi/2, 2.5, "1.5 C", color="red", ha='center', fontdict={'fontsize': 20})

ax1.text(np.pi/2, 3.0, "2.0 C", color="red", ha='center', fontdict={'fontsize': 20})

# In[ ]: