Homework 2

Student Information

Fill out the following fields:

In [ ]:
```
# Student name:
# Student e-mail:
# Student ID:
```

Assignment Information

Course	:	Introduction to Computational Physics (PHYS 220)
Semester	:	Spring 2019
Instructor	:	Dr. Zhibo Zhang
Instructor E-mail	:	Zhibo.Zhang@UMBC.edu (mailto:Zhibo.Zhang@UMBC.edu)
Teaching Assistant	:	Erick Edward Shepherd
Teaching Assistant E-mail	:	ErickShepherd@UMBC.edu (mailto:ErickShepherd@UMBC.edu)
Institution	:	University of Maryland, Baltimore County (UMBC) (www.umbc.edu)
Department	:	Department of Physics (www.physics.umbc.edu)
Assignment	:	Homework 2 of 7
Due Date	:	Tuesday, February 26th, 2019, 11:59 PM (EDT)
Assignment Value	:	6.67% of the course grade
Programming Language	:	Python 3

Assignment Description

A set of simple warmup exercises. Each of these problems should require nothing more than the standard Python syntax to solve: i.e., you do not need to import any modules or packages. This assignment also doubles as practice using Jupyter Notebooks.

Submission Instructions

Submit this assignment via the course Blackboard by the listed due date and time. If you have issues uploading the assignment to Blackboard, E-mail your submission to ErickShepherd@UMBC.edu (mailto:ErickShepherd@UMBC.edu) by the same deadline. Be sure to fill out the filename template:

- **`[Assignment] - [First_Name] [Last_Name].ipynb`**

Similarly, if you have to submit multiple files for this assigment, please put them all into a single zip folder using the same naming style.

This document was initially written as a Python 3 Jupyter Notebook and is intended to be read and completed as such. However, if you are instead reading the PDF version of this document, write your coded and commented solutions to each problem in a separate Python `.py` file following the same file naming style.

e.g., if your name is *John Smith* and you are submitting *Homework 4*, your submission should be either the Jupyter Notebook named **`Homework 4 - John Smith.ipynb`**, a Python `.py` file named **`Homework 4 - John Smith.py`**, or the folder named **`Homework 4 - John Smith.zip`**.

Additionally, remember to fill out the Student Information section prior to submission. If you are completing this assignment as a Python `.py` file, add similar documentation identifying the author in a comment at the beginning of the document.

- **Note:** You do not need to include the `.ipynb_checkpoints` directory or any files contained therein in your submission.

Table of Contents

Setup Code

No work is required at this point. However, subsequent problems depend on the imports and functions from this code cell. Re-run the below code cell if `NameError`s occur elsewhere in the document.

In [23]:

```
# Standard library imports.
import datetime

# Third party imports.
import matplotlib.pyplot as plt
import numpy as np
import netCDF4 as nc

reference_date = datetime.datetime(1800, 1, 1)


def load_data_from_nc():

    """

    Loads air temperature data from NetCDF file.

    """

    with nc.Dataset("air.mon.mean.nc", "r") as file_data:

        latitudes        = np.array(file_data.variables["lat"][:])
        longitudes       = np.array(file_data.variables["lon"][:])
        times            = np.array(file_data.variables["time"][:])
        air_temperatures = np.array(file_data.variables["air"][:])

    # Converts time data from Unix/POSIX time to Gregorian time.
    times = np.array([reference_date + datetime.timedelta(hours = x) for x in

    return latitudes, longitudes, times, air_temperatures


def load_data_from_npz():

    """

    Loads air temperature data from NumPy savez file.

    """

    with np.load("data.npz") as file_data:

        latitudes        = file_data["lats"][:]
        longitudes       = file_data["lons"][:]
        times            = file_data["time"][:]
        air_temperatures = file_data["air"][:]

    # Converts time data from Unix/POSIX time to Gregorian time.
    times = np.array([reference_date + datetime.timedelta(hours = x) for x in

    return latitudes, longitudes, times, air_temperatures


"""

Data fields:
```

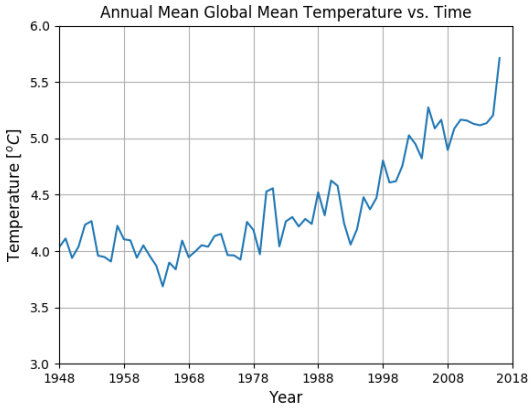
```
    Latitudes:
        Shape: (73,) -> (latitudes,)
        Range: -90 <= latitude <= 90
        Units: Degrees

    Longitudes:
        Shape: (144,) -> (longitudes,)
        Range: 0 <= longitude <= 357.5
        Units: Degrees

    Time:
        Shape: (829,) -> (time,)
        Range: 1297320.0 <= time <= 1902192.0
        Units: Hours since 1800-01-01 00:00:0.0 (UTC)
        Start: 1948-01-01 00:00:0.0 (UTC)
        End:   2017-01-01 00:00:0.0 (UTC)
        Notes: Each time stamp is spaced one month apart.

    Air temperature:
        Shape: (829, 73, 144) -> (time, latitudes, longitudes)
        Range: -73.78 <= temperature <= 41.749
        Units: Degrees centigrade

"""
```

Problem 1 (40%) - Simple Numpy Exercises

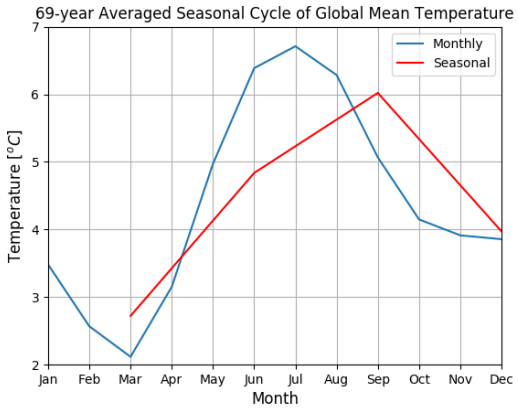
> **Problem value:** 40% of assignment grade.

Use the air temperature data to do the following exercises:

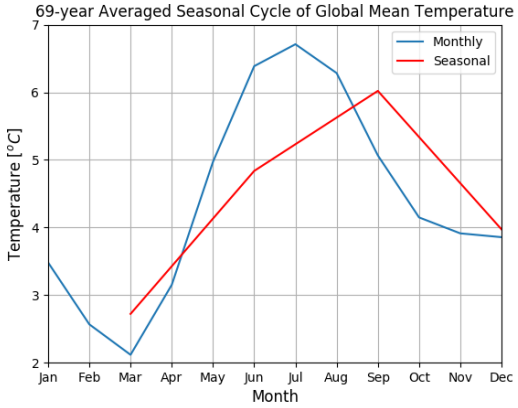
1. Reshape the air temperature array into a 4-dimentional array of shape `(years, months, latitudes, longitudes)`, or `(69, 12, 73, 144)`. Discard the last time value for the month of January, 2017.
2. Derive and plot the annual mean global mean temperature over the last 69 years. It should resemble the following:



3. Derive and plot the 69-year averaged seasonal cycle of global mean temperature. It should resemble the following:



4. Derive and plot the temperatures of the Summer months of the United States (June, July, August) over the last 69 years. It should resemble the following:



In [22]:

```
# Loads the data from the netCDF4 file.
latitudes, longitudes, times, air_temperatures = load_data_from_nc()

print("Dataset array shapes:")
print("Latitudes:        {}".format(latitudes.shape))
print("Longitudes:       {}".format(longitudes.shape))
print("Times:            {}".format(times.shape))
print("Air temperatures: {}".format(air_temperatures.shape))

# ---------------------------------
# Write your code below this point.
# ---------------------------------



# ---------------------------------
# Write your code above this point.
# ---------------------------------
```

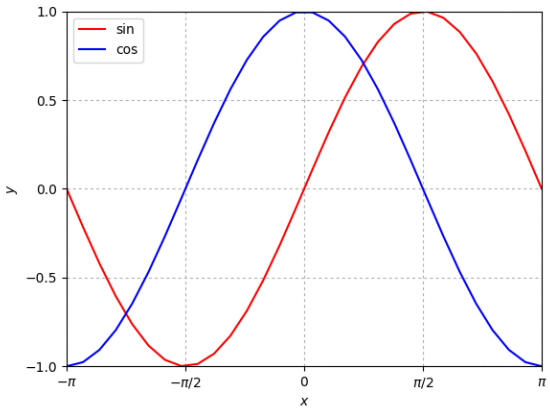
```
Dataset array shapes:
Latitudes:        (73,)
Longitudes:       (144,)
Times:            (829,)
Air temperatures: (829, 73, 144)
```

Problem 2 (20%) - Line Plot

> **Problem value:** 20% of assignment grade.

Plot the following figure using Numpy & Matplotlib. Your figure needs to closely resemble the figure below, including:

1. Proper labels and limits for the x-axis and y-axis utilizing $LaTeX$ where appropriate
2. Latex symbols for the x-axis ticks
3. A legend
4. Gridlines



In [24]:

```
x      = np.linspace(-np.pi, np.pi, 30)
cosine = np.cos(x)
sine   = np.sin(x)

plt.figure()

# ---------------------------------
# Write your code below this point.
# ---------------------------------



# ---------------------------------
# Write your code above this point.
# ---------------------------------

plt.show()
```

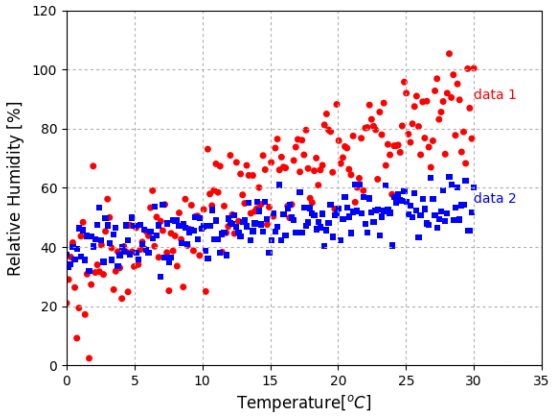
```
<Figure size 432x288 with 0 Axes>
```

Problem 3 (20%) - Scatter Plot

> **Problem value:** 20% of assignment grade.

Plot the following figure using Numpy & Matplotlib. Your figure needs to closely resemble the figure below, including:

1. Proper labels and limits for the x-axis and y-axis, using $LaTeX$ where appropriate
2. Annotations
3. Gridlines



In [13]:

```
"""

Glossary:
 * T:  Temperature
 * RH: Relative humidity

"""

T    = np.linspace(0,30,200)
dRH1 = 10.0
dRH2 = 5.0
RH1  = 2.0 * T + 30.0 + dRH1 * np.random.randn(T.size)
RH2  = 0.5 * T + 40.0 + dRH2 * np.random.randn(T.size)

plt.figure()

# ---------------------------------
# Write your code below this point.
# ---------------------------------



# ---------------------------------
# Write your code above this point.
# ---------------------------------

plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

Problem 4 (20%) - 2D Contour Plot

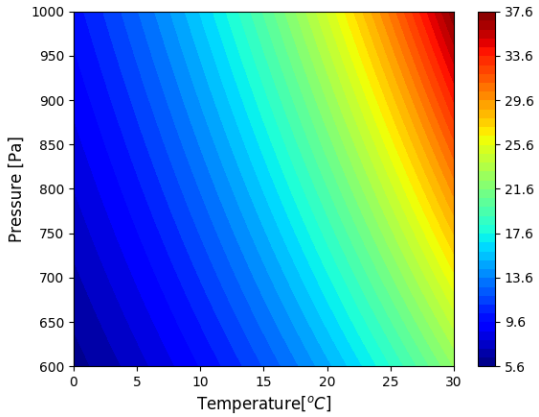
Problem value: 20% of assignment grade.

The relative humidity (RH) is a function of temperature (T) and pressure (P) as follows:

$$RH(T, P) = \frac{P}{100} e^{\frac{12T}{T+243}}$$

where, P (in Pascals) varies from 600 to 1000 and T (in Celsius) varies from 0° to 30°. Make a contour plot of RH as a function of T (x-axis) and P (y-axis). Your plot should resemble the below figure, including:

1. Proper labels and limits for the x-axis and y-axis utilizing $LaTeX$ where appropriate
2. A colorbar
3. A "jet" colormap



In [12]:
```
T = np.linspace(0, 30, 50)
P = np.linspace(600, 1000, 50)

RH = np.outer((P / 100), np.exp(12 * T / (T + 243)))

plt.figure()

# --------------------------------
# Write your code below this point.
# --------------------------------



# --------------------------------
# Write your code above this point.
# --------------------------------

plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

Co-authored by Dr. Zhibo Zhang and Erick Edward Shepherd.