ORF 363/COS 323 | Problem set 5 | Fall 2023, Princeton University

Instructor: A.A. Ahmadi

TAs: Beneventano, Chaudhry, Hua, Li, Lok

Due on Thursday, November 9, 2023, at 1:30pm EST, on Gradescope

For all problems that involve coding, please include your code. Our instructions are given with MATLAB in mind (recommended language), but you are free to use a different language.

Problem 1: Princeton's development plans

In the file `princetoncampus.png`, you can see a bird's eye view of our campus before the recent constructions with 6 residential colleges, and Dillon gym, marked by crosses.

Open this image using the following code once you've added the image to your Matlab path:

```
colleges = imread('princetoncampus.jpeg');
imshow(colleges,'InitialMagnification',50);
hold on
```

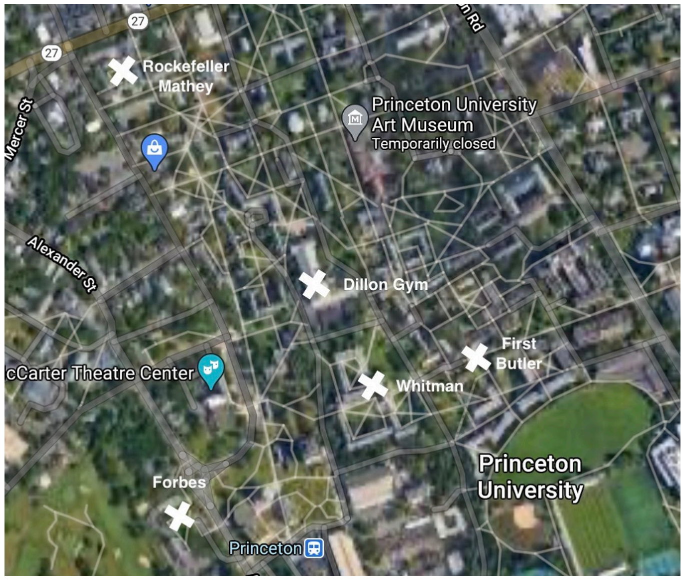
For Python users, you can use the following code in order to be consistent with Matlab.

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.image import imread

colleges = imread('princetoncampus.jpeg')
colleges_flipped = np.flipud(colleges)
plt.imshow(colleges_flipped, origin='lower')
```

We have placed a grid (a coordinate system) on the image with $(0,0)$ in the lower left corner. For those of you who want to see this grid, you can do so by copying the code given in `plotgrid.m/plotgrid.py` into Matlab/Python, but this is not mandatory.

On this grid, the colleges have the following coordinates: $z_1 = (8, 34.5)$ for the regroupment Rockfeller-Mathey, $z_2 = (12, 4)$ for Forbes, $z_3 = (25, 13)$ Whitman, $z_4 = (32, 15)$ for the regroupment First-Butler. Dillon gym has coordinates $z_5 = (21, 20)$.



Part 1: Unconstrained optimization

Princeton University wants to understand if the current location of Dillon gym is convenient for undergraduate students. To answer this question, the Princeton Trustees would like to know where a gym that minimizes the sum of distances squared to the residential colleges would be. The location of this gym is the solution to the following optimization problem:

$$\min_x \sum_{i=1}^{4} ||x - z_i||^2. \tag{1}$$

If the location of Dillon gym is considered "sufficiently" close to the optimal location, then they will not rebuild the gym; if it isn't, then they will.

1. Solve this problem using CVX. Report the optimal solution x^* and the optimal value. To visualize where x^* is on the map, use the function `Circledraw.m` to plot a small circle (e.g. of radius 0.5) around the point. What building does this correspond to?

   Remark: To plot a circle, download the file `Circledraw.m` into your Matlab directory. The command `Circledraw(x,y,r,`color')` will draw a circle of center $(x, y)$, of radius $r$ and of color `color` (this could be ``r'` for red for example or ``magenta'`). Make sure you have opened the image in Matlab as said previously. The function `plot(x,y)` will *not* work here as we are working with a different coordinate system.

   For Python users, you can use the following code in order to be consistent with Matlab.

```python
from matplotlib.patches import Circle

scale = 22
circle = Circle((x*scale, y*scale), r*scale, color='red',
    fill=False)
plt.gca().add_patch(circle)
plt.axis('off')
plt.show()
```

2. You are tired of using CVX as a black-box and want to implement your own solver. In this part of the question, we will solve (1) by implementing two variants of gradient descent. For each implementation, use a random point in $[0, 1] \times [0, 1]$ as a starting point and stop the iterations when either $||\nabla f(x_k)|| < 10^{-5}$ or the iteration count has reached 100. (The function f here is the objective function.)

   (a) Implement steepest descent with exact line search (a.k.a. the line minimization rule). How many steps does convergence take? Why?

   (b) Implement steepest descent with constant stepsize for three different values of the stepsize: $\alpha = \frac{1}{2}$, $\alpha = \frac{1}{4}$ and $\alpha = \frac{2}{9}$. Plot the iterates you obtain on three different figures. For each step size, do the iterates (i) converge to the global minimum, (ii) oscillate, or (iii) diverge? Why?[1]
   Hint: To make the analysis simpler, you may want to perform a change of variables so the minimum of f gets shifted to the origin in the new coordinates.

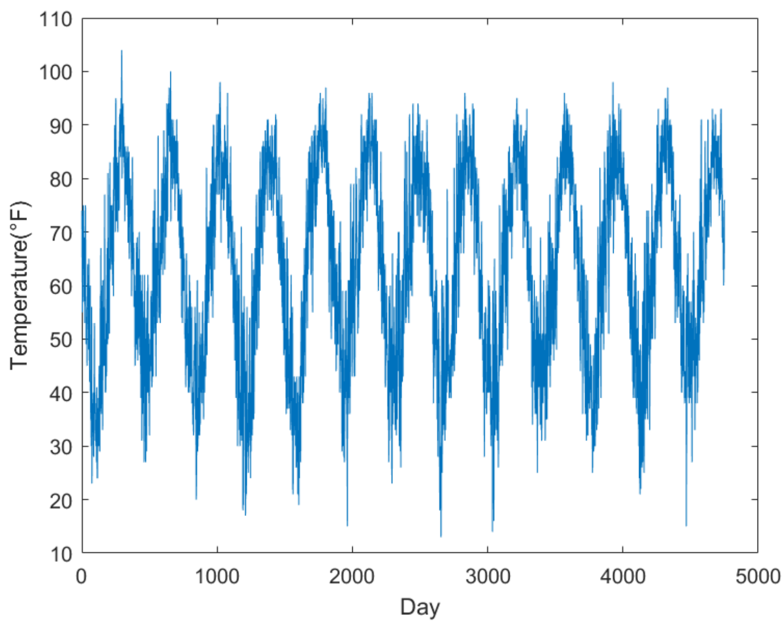
Part 2: Constrained optimization

The state of New Jersey has recently enacted a law that requires all public buildings to be at a maximum distance of 8 from a fire hydrant. There is only one available fire hydrant for the gym at location $(35, 28)$. Write down a new optimization problem (call it problem (2)) that resolves (1) with the fire hydrant constraint. Prove that the optimal solution to (2) must be attained on the boundary of the feasible set. Use CVX and report the optimal solution and the optimal value. On the same figure, plot the optimal solution to (2), the optimal solution to (1), and the boundary of the feasible set of (2). Does the plot make sense?

[1]If you are wondering how rigorous your answer should be, we remind you that the answer to the question "why?" in mathematics is *always* a proof ;)

Problem 2: Orbit of the Earth around the Sun and Temperatures in NYC

In the file `TemperatureNewYork.csv`, you are given the maximum daily temperatures in Central Park, New York over 13 years (starting on 10/01/2010).[2] To load the data, you can use the function `readmatrix` in Matlab or `np.loadtxt` in Python. This is what the data looks like:



The goal of the question is to estimate the duration of the Earth's orbit around the Sun, using only temperature data in New York City. To do this, we wish to fit the following model to the data

$$T(t) = a \sin\left(\frac{2\pi}{T_E}t + \Phi\right) + bt + c,$$

where $T(t)$ is the temperature on day t, and our parameters to be found are a, b, c, Φ and T_E. The last parameter T_E corresponds to the length of the Earth's orbit in days. Essentially, we are postulating that the temperature is a periodic function of time, with a given phase and period, to which we have added an offset and growth term. We saw in class that the problem of fitting such a model to data is a nonlinear least-squares problem, for which the Gauss-Newton method is a popular algorithm.

Write down the nonlinear least squares problem that minimizes the sum of the squares of the residuals between the data and the prediction from the model over the 13 years. Apply 50 iterations of the Gauss-Newton method to the problem starting with the initial condition

[2] This data was obtained from `http://www.ncdc.noaa.gov/cdo-web/`.

$(a, T_E, \Phi, b, c) = (120, 360, \pi, 0, 0)$.[3] Plot the data we gave you and the best fit that you found on the same figure.

Hint: You may want to use *function handles* in Matlab for the implementation of the Gauss-Newton algorithm. These are used to create and evaluate functions in a simple and efficient manner. Here is an example:

```
>>f=@(x,y,z) x^2+2*y+1/z;
>>f(1,1,1)
ans=
4
```

In Python, you can use *lambda functions* to create function handles, but it is totally fine if you write normal functions. Here is an example:

```
f = lambda x, y, z: x**2 + 2*y + 1/z
```

(a) What is the period T_E^* that you obtain for the revolution of the Earth around the Sun? How far is it from the true value? (You can use Wikipedia e.g. to get this number.)

(b) What do this particular model and dataset tell you about global warming?[4]

Problem 3: Finding a Lyapunov function for a linear dynamical system

Consider the linear system in two dimensions given by

$$x_{k+1} = Ax_k,$$

where

$$A = \begin{pmatrix} 0 & 2 \\ \frac{1}{4} & 0 \end{pmatrix}, x_k = \begin{pmatrix} w_k \\ v_k \end{pmatrix} \in \mathbb{R}^2,$$

and $k = 0, 1, \ldots$ is the index of time. We want to prove that the system is "stable"; i.e., that for all $x_0 \in \mathbb{R}^2$, we have $x_k \to 0$ as $k \to \infty$.

Prove this property by finding an appropriate Lyapunov function for this system. (Hint: you can restrict your search to functions of the form $V(w, v) = aw^2 + bv^2$, where $a, b \in \mathbb{R}$.)

[3] As the problem is nonconvex, the success of the algorithm is highly sensitive to the way the initial conditions are chosen. This is why we start the algorithm with a reasonable guess for the values of our parameters. Feel free to play around with these initial conditions.

[4] You should be careful not to form serious opinions about an important issue like global warming based on one limited experiment.