Assignment #6

Math 2604: Mathematical Programming in Python

https://people.sc.fsu.edu/~jburkardt/classes/python_2025/assignment06/assignment06.pdf

---

Instructions: Choose 3 of the following problems to work on. Submit your responses as Python text files, with the extension `.py`. Each file should include your name and the problem number.

- *Problem 6.0:* In class, we discussed counting the number of ways of achieving a specific score in football. We assumed points were gained in the following ways:

  - 2 points for a safety;
  - 3 points for a field goal;
  - 6 points for a touchdown with no conversion;
  - 7 points for a touchdown with 1 point conversion;
  - 8 points for a touchdown with 2 point conversion.

  If we don't care about the order in which points are made, there are 11 ways of making a score of 14. But suppose, instead, we do care about the order. So 6 points followed by 8 points is counted separately from 8 points followed by 6. How many ways are there of making a score of 14 points if order matters? Here's a way to determine this. Create a `ways` array big enough to hold 0 through 14. Let `ways[0]=1`. Now, for each score from 1 to 14, do the following:

  ```
  if ( 0 <= score - 2 ) ways[score] = ways[score] + ways[score-2]
  if ( 0 <= score - 3 ) ways[score] = ways[score] + ways[score-3]
  ...and so on for 6, 7 and 8
  ```

- *Problem 6.1:* In class, we talked about how to use the `numpy` function `meshgrid()` to create X and Y coordinate matrices from x and y vectors, so that we could compute the values of $Z(X,Y)$ in a single statement. Create x and y vectors of length 33 over the range -8 to 8. Use `meshgrid()` to create the X and Y matrices, and then evaluate the functions $R = \sqrt{X^2 + Y^2}$ and $Z = \sin(R)/R$. Python will probably complain about a division by zero. We talked about how to fix the Z matrix to get rid of the bad 0/0 value. We could now do a contour plot. But instead, I'd like you to try a 3D surface plot. So issue the commands

  ```
  from mpl_toolkits.mplot3d import Axes3D
  fig = plt.figure ( )
  ax = fig.add_subplot ( 111, projection='3d' )
  ax.plot_surface ( X, Y, Z, edgecolor = 'none' )
  plt.show ( )
  ```

  If you can't get the 3D plot to work, then just do a contour plot of the data with `plt.contourf()`.

- *Problem 6.2:* In assignment 4, we computed the area of the polygon whose vertices had the following (x,y) coordinates:

  ```
  x  y
  0, 0
  3, 0
  3, 3
  2, 3
  2, 1
  1, 1
  1, 2
  0, 2
  ```

By plotting this data, it is easy to see that this polygon is contained in a square of side 3, and that it has area 6 units. Let's pretend this is a hard problem, and try to estimate the area using rejection. To do this, we need a special function that can determine whether a point is inside a polygon.

```
from matplotlib.path import Path
polygon = Path ( v )
```

Now we can generate a random point xy in the $[0, 3] \times [0, 3]$ square using `np.random.uniform()`, and if it is inside the polygon, we increment a counter k:

```
  if ( polygon.contains_point ( xy ) ):
    k = k + 1
```

After you have created n points, and counted k of them inside the polygon, your estimate for the area of the polygon is $area \approx 9 * k/n$. See if you can make this work! If so, you can now estimate the area of any region defined by a polygon.

- *Problem 6.3:* Suppose that three flies each land at random positions inside a circular plate of radius 1. Suppose we write A to symbolize the area of the triangle formed by their positions. Use Heron's formula to determine this area. Use simulation to repeat this experiment 1000 times, and estimate the mean and standard deviation of A.

- *Problem 6.4:* In the game of "odd man out", a group of n people each flip a coin. If 1 person flips heads and all the others have tails, or 1 person flips tails and all the others have heads, that person loses and the game is over. Otherwise, they flip again.

  Suppose $n = 7$ people play this game. Estimate the number of times they must flip a coin before someone becomes the odd man out.

- *Problem 6.5:* Consider a modified version of a duel Anne has accuracy 0.25, meaning that, on any shot, Anne will hit her target with probabiity 0.25. Similarly, Barbara has accuracy 0.30. In a typical duel, the particpants alternate taking a single shot at each other, until one is hit. However, in this version, Anne goes first, firing once. Unless she hits, Barbara now has two chances to fire at Anne. Unless she hits, Anne now gets three shots at Barbara, and so on. This continues until someone is hit. Estimate the chances that Anne or Barbara will win in this modified duel.

- *Problem 6.6:* Run the SIR model on a 10×10 grid with a probability of transmission $p = 0.2$ and a sickness length of 4 days. Choose one infected person at a random position in your grid. Run your simulation for 50 days. Make a plot of the values of S(t), I(t) and R(t) over the 50 day period.

- *Problem 6.7:* For the problem of vaccinating the wolves, suppose that the park buys a second trap, so that now two wolves are trapped every night. What is the average number of days required before all wolves will be vaccinated. When you select two wolves at random, be careful. `np.random.randint()` could return two values that are the same. It would be better to use `np.random.choice()`, which includes the r `eplace = False` option.

- *Problem 6.8:* Consider a 2D random walk starting at the origin. On each step, the walker at position (i,j) randomly moves to $(i + 1, j)$, $(i - 1, j)$, $(i, j + 1)$ or $(i, j - 1)$. Measure the distance of the walker from the origin using the Euclidean measurement $d(i, j) = \sqrt{i^2 + j^2}$. Stop the walk as soon as the walker has reached a distance of 10 or more units from the origin, and note the number of steps that were taken. Perform this experiment 1000 times, and report the average number of steps taken.

- *Problem 6.9:* We discussed in class how to use the `matplotlib` function `plt.fill()` to create a simple image of Mario, using a small set of colored squares. Use the same ideas to create a minion: