Exercises on `numpy`, `scipy`, and `matplotlib`

1 Exercise 7: Numpy practice (5 points)

Start up Python (best to use Spyder) and use it to answer the following questions. Use the following imports:

```
import numpy as np
import scipy.linalg as la
import matplotlib.pyplot as plt
```

1. Choose a value and set the variable `x` to that value.
2. What is command to compute the square of `x`? Its cube?
3. Choose an angle θ and set the variable `theta` to its value (a number).
4. What is $\sin\theta$? $\cos\theta$? Angles can be measured in degrees or radians. Which of these are being used used?
5. Use the `np.linspace` function to create a row vector called `meshPoints` containing exactly 500 values with values evenly spaced between -1 and 1.
6. What expression will yield the value of the 53^{th} element of `meshPoints`? What is this value?
7. Produce a plot of a sinusoid on the interval $[-1, 1]$ using the command

   ```
   plt.plot(meshPoints,np.sin(2*pi*meshPoints))
   ```

   Please save this plot as a jpeg (.jpg) file and send it along with your work.

2 Vector and matrix operations: Exercise 8 (5 points)

Define the following vectors and matrices:

```
vec1 = np.array([ -1.,  4., -9.])
mat1 = np.array([[ 1.,  3.,  5.], [7., -9., 2.], [4., 6., 8. ]]
```

1. You can multiply vectors by constants. Compute

   ```
   vec2 = (np.pi/4) * vec1
   ```

2. The cosine function can be applied to a vector to yield a vector of cosines. Compute

```
vec2 = np.cos( vec2 )
```

3. You can add vectors and multiply by scalars. Compute

   ```
   vec3 = vec1 + 2 * vec2
   ```

4. The Euclidean norm of a matrix or a vector is available using `la.norm`. Compute

   ```
   la.norm(vec3)
   ```

5. You can do row-column matrix multiplication. Compute the product of `mat1` and `vec3` and set `vec4` equal to the result.

6. Compute the transpose of `mat1`.

7. Compute the determinant of `mat1`.

8. Compute the trace of `mat1`.

9. Find the smallest element in `vec1`.

10. What function would you use to find the value of `j` so that `vec1[j]` is equal to the smallest element in `vec1`?

11. What expression would you use to find the smallest element of the matrix `mat1`?

12. As you know, a magic square is a matrix all of whose row sums, column sums and the sums of the two diagonals are the same. (One diagonal of a matrix goes from the top left to the bottom right, the other diagonal goes from top right to bottom left.) Show by direct computation that if the matrix `A` is given by

    ```
    A=np.array([[17, 24,  1,  8, 15],
                [23,  5,  7, 14, 16],
                [ 4,  6, 13, 20, 22],
                [10, 12, 19, 21,  3],
                [11, 18, 25,  2,  9]])
    ```

    The matrix `A` has 5 row sums (one for each row), 5 column sums (one for each column) and two diagonal sums. These 12 sums should all be exactly the same, and you **could** verify that they are the same by printing them and "seeing" that they are the same. It is easy to miss small differences among so many numbers, though. **Instead**, verify that `A` is a magic square by constructing the 5 column sums and computing the maximum and minimum values of the column sums. Do the same for the 5 row sums, and compute the two diagonal sums. Check that these six values are the same. If the maximum and minimum values are the same, the flyswatter

principle says that all values are the same.
Hints: The function `np.diag` extracts the diagonal of a matrix, and the function `np.fliplr` extracts the other diagonal.

13. The function `np.random.rand` can be used to construct and fill vectors and matrices with random numbers. Use the help facility in Python to learn how to construct a 10×10 matrix named `M` filled with random numbers.

14. The colon and index notation can also be used to refer to a subset of elements of the array. With the *start:increment:finish* notation, we can refer to a range of indices.

    What commands would be needed to generate the four 5×5 sub-matrices of `M` in the upper left quarter `MUL`, the upper right quarter `MUR`, the lower left quarter `MLL`, and the lower right quarter `MLR`.

3 Exercise 9: Plotting (5 points)

1. Use `matplotlib.pyplot.plot` to produce a plot of the functions $f(x) = e^{-x/10}\sin(\pi x)$ and $g(x) = xe^{-x/3}$ over the interval $[0, 10]$. Include labels for the x- and y-axes, and a legend explaining which line is which plot. Save the plot as a `.jpg` ("Jpeg") file and send me a copy with your work.

2. The shape of a limaçon can be defined parametrically as

$$
\begin{aligned}
r &= r_0 + \cos\theta \\
x &= r\cos\theta \\
y &= r\sin\theta
\end{aligned}
$$

    When $r_0 = 1$, this curve is called a cardioid. Use this definition to plot the shape of a limaçon for $r_0 = 0.8$, $r_0 = 1.0$, and $r_0 = 1.2$. Be sure to use enough points that the curve is closed and appears smooth (except for the cusp in the cardioid). Use a legend to identify which curve is which. Save the plot as a `.pdf` file and send me a copy with your work.