Homework (A8B17CAS)

Problem Set 3

1 Assignment

P3-A Define MATLAB function `rot90` that takes a matrix (a list of lists) and rotates it counter-clockwise. *E.g.* an original matrix, `matA`, the output/transformed matrix, `matB = rot90[matA]`:

$$\texttt{matA} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad \texttt{matB} = \begin{bmatrix} 3 & 6 \\ 2 & 5 \\ 1 & 4 \end{bmatrix}. \tag{1}$$

<u>Task</u>: Write your definition of `rot90` to your notebook.
<u>A hint</u>: Take a look at MATHEMATICA command `Reverse`.

(**1 points**)

P3-B Assume a function $f(x) = x^5 e^{-x}$. Compute its first 4 derivatives. The function and their derivative are in Figure 1. Get the x-coordinates of the 5 shown points. You can use the command `FindRoot`. It requires a starting point. Use the starting points from the list/vector: `{5.1, 3, 1.5, 1, 0.5};`

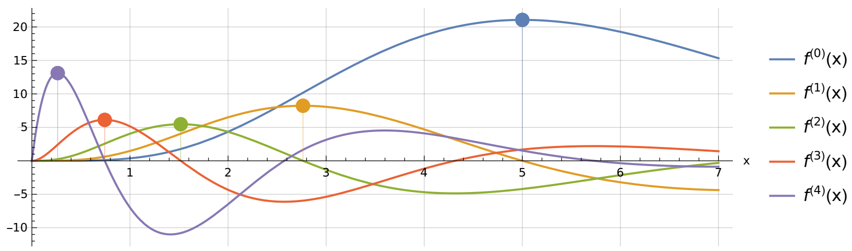


Figure 1: Function $f(x)$ and its first 4 derivatives.

<u>Task</u>: Get a list of x-coordinates of the 5 shown points. (Transform a list of rules that you get from `FindRoot` to a list of numbers).
<u>A hint</u>: Additional command that can be useful is `MapThread`, and maybe `NestList`.

(**2 points**)

P3-C Define a function `myMat[p_]` that for a given $p \geq 3$, $p \in \mathbb{Z}$ generates the following square matrix:

$$\begin{bmatrix} -p & 0 & \dots & 0 & & & \\ \vdots & \vdots & \ddots & \vdots & & \mathbf{R} & \\ \vdots & 0 & \dots & 0 & & & \\ -1 & 1 & \dots & 1 & 1 & \dots & 1 \\ 0 & 1 & \dots & 1 & 1 & \dots & 1 \end{bmatrix}. \tag{2}$$

E.g.

$$\texttt{myMat[5]=} \begin{bmatrix} -5 & 0 & 0 & 0 & 12 & 10 \\ -4 & 0 & 0 & 0 & 20 & 18 \\ -3 & 0 & 0 & 0 & 20 & 14 \\ -2 & 0 & 0 & 0 & 18 & 20 \\ -1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \ \texttt{myMat[7]=} \begin{bmatrix} -7 & 0 & 0 & 0 & 18 & 20 & 14 & 12 \\ -6 & 0 & 0 & 0 & 18 & 13 & 11 & 11 \\ -5 & 0 & 0 & 0 & 19 & 15 & 14 & 13 \\ -4 & 0 & 0 & 0 & 16 & 10 & 12 & 20 \\ -3 & 0 & 0 & 0 & 10 & 15 & 16 & 10 \\ -2 & 0 & 0 & 0 & 15 & 20 & 11 & 10 \\ -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{3}$$

Where $\mathbf{R} \in \mathbb{R}^{(p-1)\times(p-3)}$ is a matrix of random integers from a uniform distribution on a set $\{10, \ldots, 20\}$. Matrix of zeros, $\mathbf{O} \in \mathbb{R}^{(p-1)\times 3}$ and matrix of ones $\mathbf{1} \in \mathbb{R}^{2\times p}$.

Task: Write your definition of `myMat` to your notebook.

(**1 points**)

P3-D Assume $n \in \mathbb{N}$. Generate a list of n 2D points (a matrix with 2 columns and n rows). Denote the list as `ptsRhoPhi`, it is a list of the form $\{\{\rho_1, \varphi_1\}, \{\rho_2, \varphi_2\}, \ldots, \{\rho_n, \varphi_n\}\}$. Generate ρ_is as a random variable from the normal distribution $\mathcal{N}(10, 1)$ – `RandomVariate[ NormalDistribution[10, 1]]` and φ_is as a random variable from the continous uniform distribution $\mathcal{U}(0, 2\pi)$ – `RandomVariate[ UniformDistribution[0, 2 Pi[]]`.

Create matrices/lists `ptsXY` and `ptsXYSorted` such that:

- the list `ptsXY` contains pairs $\{x_i, y_i\}$, where $x_i = \rho_i \cos(\varphi_i)$ and $y_i = \rho_i \sin(\varphi_i)$.
- the list `ptsXYSorted` is sorted `ptsXY` by the angle of the point in plain.

Task: For a given n, create list `ptsRhoPhi`. Then create lists `ptsXY` and `ptsXYSorted`. Write the procedure to your notebook.
A hint: You can use *e.g.* `ArcTan` (with two arguments) to get the angle from x,y coordinates.

You can plot the resulting vectors using:

```
ListPlot[{ptsXY, ptsXYSorted}, Joined -> True, AspectRatio -> 1,
  AxesLabel -> {"x", "y"}, PlotLegends -> {"ptsXY", "ptsXYSorted"}]
```

Figure 2 shows an example of such a plot.

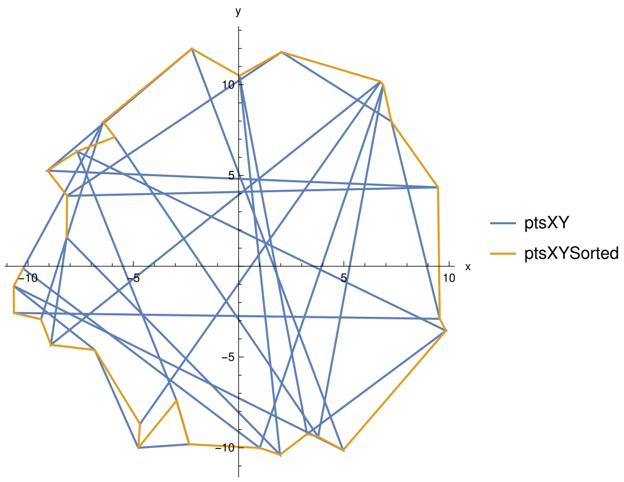


Figure 2: A plot of `ptsXY` and `ptsXYSorted` for $n = 30$.

(**2 points**)

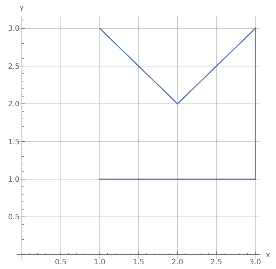
P3-E Assume a list of points' coordinates $[x, y]$ that form a polygon ($[x_1, y_1], [x_2, y_2], [x_3, y_3], [x_4, y_4], \ldots, [x_N, y_N]$). The last polygon line should be from the last point to the first point. Compute the perimeter and area of the polygon. The polygon area can be computed as[1]:

$$S = \frac{1}{2}\left(\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix} + \ldots + \begin{vmatrix} x_{N-1} & y_{N-1} \\ x_N & y_N \end{vmatrix} + \begin{vmatrix} x_N & y_N \\ x_1 & y_1 \end{vmatrix} \right).$$

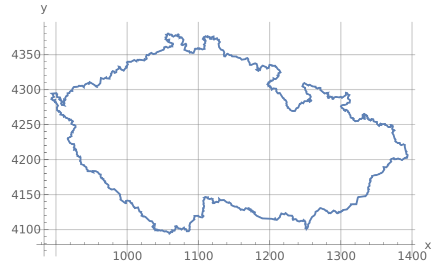
Where $|.|$ denotes determinant of a matrix.
Task: Determine perimeter and area of the two polygons in Figure 3.
A hint: Use `Partition` command.

```
ptsPolyg = {{1, 1}, {3, 1}, {3, 3}, {2, 2}, {1, 3}}
ListLinePlot[%, AspectRatio -> 1, AxesOrigin -> {0, 0},
  GridLines -> Automatic, AxesLabel -> {"x", "y"}]
 (*perimeter -- 6 + 2 Sqrt[2] *)
 (*area -- 3 *)
```

[1] https://mathodics.com/find-area-of-polygon-using-coordinates/



(a) Polygon for testing of your code.



(b) Border of Czech republic shown as a polygon.

Figure 3: Polygons in 2D.

```
Entity["Country", "CzechRepublic"]["FullCoordinates"];
GeoGridPosition[GeoPosition[%[[1]]], {"Mercator", "ReferenceModel" -> "WGS84",
  "StandardParallels" -> {48.5, 51.1}, "GridOrigin" -> {48.5, 51.1}}];
ptsPolyg = First@%;
ListLinePlot[%/1000, AxesLabel -> {"x", "y"}, GridLines -> Automatic]
```

(**2 points**)

P3-F Generate Pascal's triangle and its modified version. Do not use the `Binomial` command. Compute the rows iteratively so that your function takes one row (a list of numbers) and returns the next row of numbers. Use your function with the command `NestList` to get a list of the first few rows.

Define a second function: instead of the sum $x_1 + x_2$ of the above numbers x_1, x_2 returns $x_1 + 2x_2$. Use the function with command `NestList` to get the first few rows of such a triangle.

You can use the postfix command `// Column[#, Center] &;` to show the list of rows in a triangle form.

```
{1}
{1, 1}
{1, 2, 1}
{1, 3, 3, 1}
{1, 4, 6, 4, 1}
{1, 5, 10, 10, 5, 1}
{1, 6, 15, 20, 15, 6, 1}
```

```
{1}
{1, 1}
{1, 3, 1}
{1, 7, 5, 1}
{1, 15, 17, 7, 1}
{1, 31, 49, 31, 9, 1}
{1, 63, 129, 111, 49, 11, 1}
```

Figure 4: The Pascal's triangle and its modified version.

Task: Compute the first few rows of the ordinary Pascal's triangle and its modified version.
A hint: Useful commands to look at are `Partition` and `Join`.

(**2 points**)

2 Instructions

Complete all the assignments till

- (Fri) January 3 2024, 23:59.

Write your solutions into a notebook called `Problem3_yourusername.nb` and send it to `lukacjo1@fel.cvut.cz` with subject "CAS:HW3". All the problems shall be solved by the students individually. Do not use functions from MATHEMATICA Packages.

Contact us at `lukacjo1@fel.cvut.cz` with any questions. The team of teachers wishes you good luck in solving this.