Data visualization course

**Laboratory work 1**

# Drawing plots via Matplotlib

Code example for polar coordinates plot:

**import** numpy **as** np

**import** matplotlib**.**pyplot **as** plt

r **=** np**.**arange**(**0**,** 2**,** 0.01**)**

theta **=** 2 **\*** np**.**pi **\*** r

ax **=** plt**.**subplot**(**111**,** projection**=**'polar'**)**

ax**.**plot**(**theta**,** r**)**

ax**.**set\_rmax**(**2**)**

ax**.**set\_rticks**([**0.5**,** 1**,** 1.5**,** 2**])** # less radial ticks

ax**.**set\_rlabel\_position**(-**22.5**)** # get radial labels away from plotted line

ax**.**grid**(True)**

ax**.**set\_title**(**"A line plot on a polar axis"**,** va**=**'bottom'**)**

plt**.**show**()**

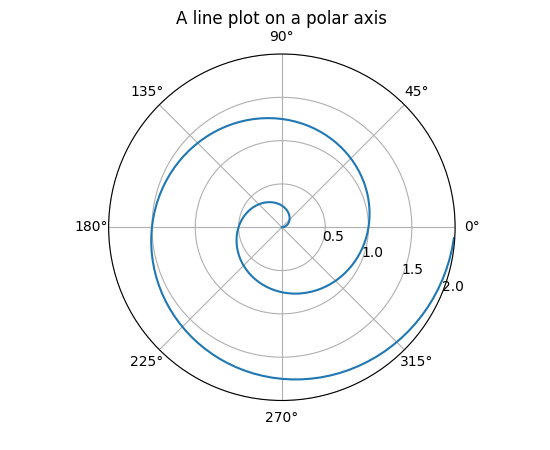


Figure 1– Polar axis plot

Code example for surface plot:

**from** mpl\_toolkits**.**mplot3d **import** Axes3D

**import** matplotlib**.**pyplot **as** plt

**from** matplotlib **import** cm

**from** matplotlib**.**ticker **import** LinearLocator**,** FormatStrFormatter

**import** numpy **as** np

fig **=** plt**.**figure**()**

ax **=** fig**.**gca**(**projection**=**'3d'**)**

# Make data.

X **=** np**.**arange**(-**5**,** 5**,** 0.25**)**

Y **=** np**.**arange**(-**5**,** 5**,** 0.25**)**

X**,** Y **=** np**.**meshgrid**(**X**,** Y**)**

R **=** np**.**sqrt**(**X**\*\***2 **+** Y**\*\***2**)**

Z **=** np**.**sin**(**R**)**

# Plot the surface.

surf **=** ax**.**plot\_surface**(**X**,** Y**,** Z**,** cmap**=**cm**.**coolwarm**,**

linewidth**=**0**,** antialiased**=False)**

# Customize the z axis.

ax**.**set\_zlim**(-**1.01**,** 1.01**)**

ax**.**zaxis**.**set\_major\_locator**(**LinearLocator**(**10**))**

ax**.**zaxis**.**set\_major\_formatter**(**FormatStrFormatter**(**'%.02f'**))**

# Add a color bar which maps values to colors.

fig**.**colorbar**(**surf**,** shrink**=**0.5**,** aspect**=**5**)**

plt**.**show**()**

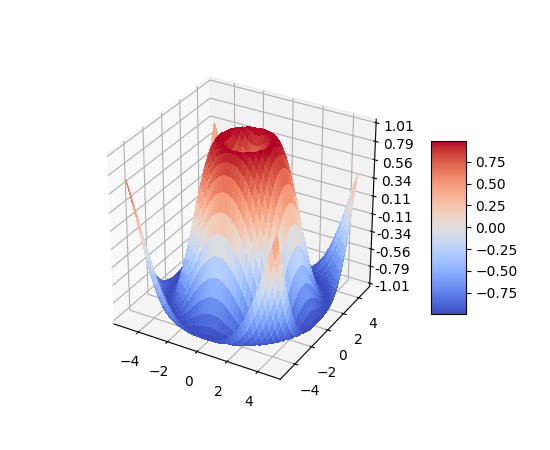


Figure 2 – Surface plot

Code example for bar chart:

**import** numpy **as** np

**import** matplotlib**.**pyplot **as** plt

**from** matplotlib**.**ticker **import** MaxNLocator

**from** collections **import** namedtuple

n\_groups **=** 5

means\_men **=** **(**20**,** 35**,** 30**,** 35**,** 27**)**

std\_men **=** **(**2**,** 3**,** 4**,** 1**,** 2**)**

means\_women **=** **(**25**,** 32**,** 34**,** 20**,** 25**)**

std\_women **=** **(**3**,** 5**,** 2**,** 3**,** 3**)**

fig**,** ax **=** plt**.**subplots**()**

index **=** np**.**arange**(**n\_groups**)**

bar\_width **=** 0.35

opacity **=** 0.4

error\_config **=** **{**'ecolor'**:** '0.3'**}**

rects1 **=** ax**.**bar**(**index**,** means\_men**,** bar\_width**,**

alpha**=**opacity**,** color**=**'b'**,**

yerr**=**std\_men**,** error\_kw**=**error\_config**,**

label**=**'Men'**)**

rects2 **=** ax**.**bar**(**index **+** bar\_width**,** means\_women**,** bar\_width**,**

alpha**=**opacity**,** color**=**'r'**,**

yerr**=**std\_women**,** error\_kw**=**error\_config**,**

label**=**'Women'**)**

ax**.**set\_xlabel**(**'Group'**)**

ax**.**set\_ylabel**(**'Scores'**)**

ax**.**set\_title**(**'Scores by group and gender'**)**

ax**.**set\_xticks**(**index **+** bar\_width **/** 2**)**

ax**.**set\_xticklabels**((**'A'**,** 'B'**,** 'C'**,** 'D'**,** 'E'**))**

ax**.**legend**()**

fig**.**tight\_layout**()**

plt**.**show**()**

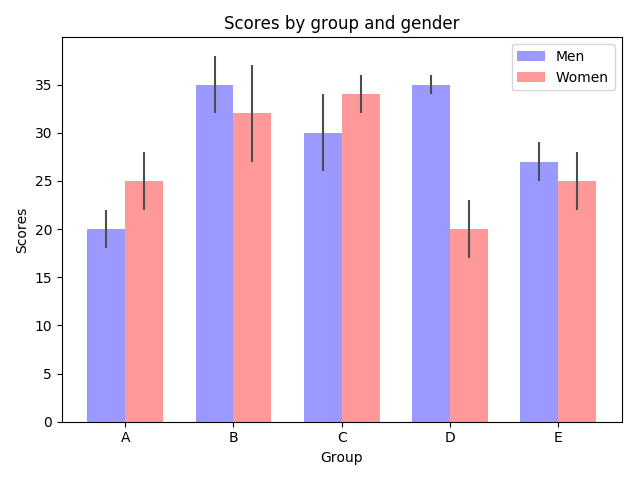
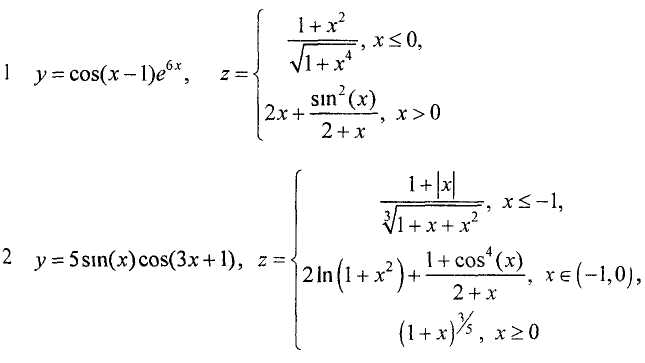


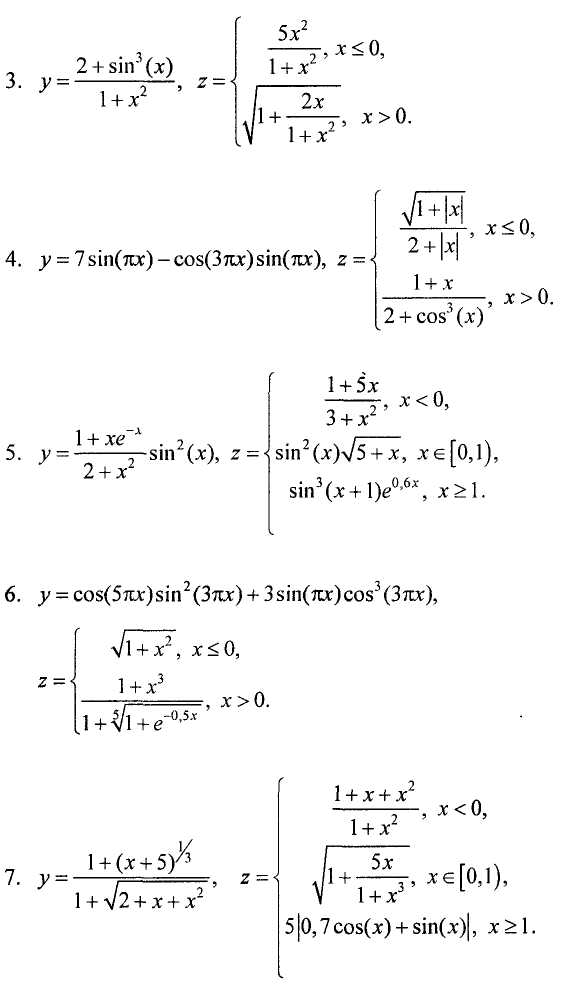
Figure 3 – Bar chart

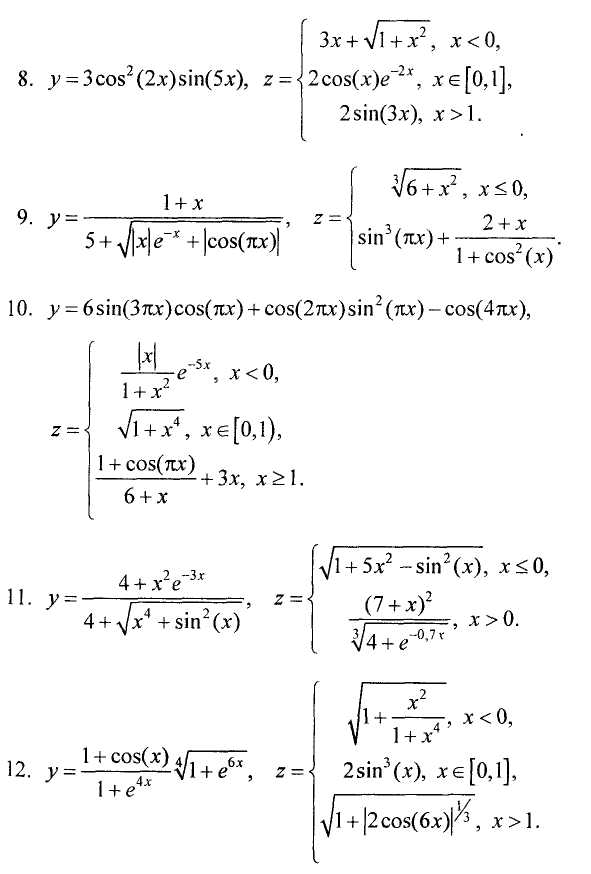
# Tasks

Plot graphs of functions, surfaces and bar charts. On all graphs, sign the axes, display the grid, and the legend. For 2D plots, write the equation of the graphs as text on the plot.

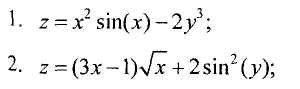
**Plot graphs of functions:**

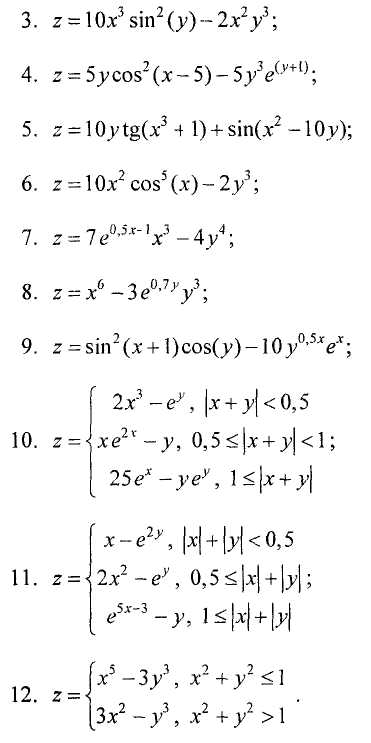




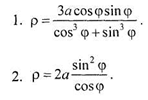


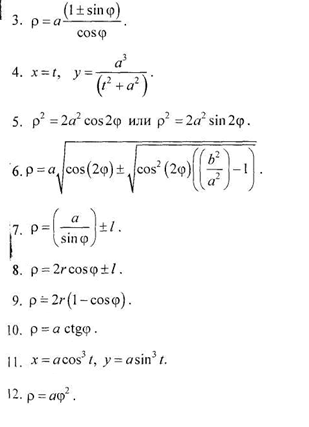
**2. Plot surfaces**



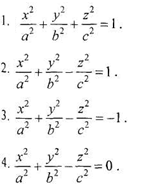


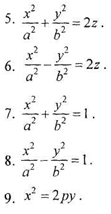
**3. Plot surface plot**





**4. Build surfaces of the 2nd order. a, b, c are constants**





**5. Based on the data from the tables, build 2d and 3d bar charts**

1. Population, million people.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1900 | 1913 | 1929 | 1938 | 1950 | 1960 | 1970 | 1980 | 1990 | 2000 |
| USA | 76,4 | 97,6 | 122,2 | 130,5 | 153 | 176 | 200,5 | 227 | 247 | 277 |
| Germany | 45,7 | 54,7 | 58,7 | 62,3 | 67 | 72 | 77 | 78,5 | 79 | 82 |
| France | 40,8 | 41,8 | 42 | 42 | 42 | 46 | 50,5 | 54 | 56,5 | 59 |
| Japan | 44 | 51,6 | 63,2 | 71,8 | 83 | 93 | 104 | 116,8 | 123,5 | 127 |
| USSR | 123 | 158 | 171,5 | 186,5 | 205,5 | 226,5 | 247 | 258,5 | 290 | 290 |

2. Number of employed in the global economy, mln people.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1900 | 1913 | 1929 | 1938 | 1950 | 1960 | 1970 | 1980 | 1990 | 2000 |
| Germany | 18,5 | 23,5 | 25 | 26,5 | 29 | 31 | 34 | 35 | 37 | 38,5 |
| France | 20 | 20 | 20 | 19,5 | 19 | 21 | 23 | 25 | 26,5 | 27,5 |
| Great Britain | 16,5 | 18,5 | 20 | 20,5 | 22,5 | 24 | 25 | 25,5 | 26 | 26,5 |
| Italy | 15 | 16,5 | 17 | 18 | 18,5 | 20 | 22 | 24 | 24,5 | 25 |

3. Industrial production: added value, in national currency prices of 2000, billion dollars

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1900 | 1913 | 1929 | 1938 | 1950 | 1960 | 1970 | 1980 | 1990 | 2000 |
| Germany | 29 | 51 | 59 | 478 | 93 | 244 | 420 | 510 | 575 | 625 |
| France | 28 | 46 | 57 | 52 | 63 | 93 | 190 | 275 | 310 | 355 |
| Great Britain | 53 | 73 | 84 | 105 | 130 | 180 | 245 | 265 | 300 | 335 |
| USSR | 40 | 70 | 80 | 105 | 205 | 480 | 725 | 935 | 1000 | 545 |

4. Global agricultural production: added value in 2000 prices, billion dollars.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1900 | 19131 | 1929 | 1938 | 1950 | 1960 | 1970 | 1980 | 1990 | 2000 |
| USA | 43 | 56 | 69 | 76,5 | 93,5 | 105 | 128,5 | 146 | 157,5 | 175 |
| Germany | 16 | 19 | 20 | 21,5 | 23 | 29 | 37 | 40,5 | 46,5 | 52,5 |
| France | 21,5 | 22 | 22,5 | 23 | 23,5 | 29,5 | 47 | 53 | 65 | 76,5 |
| Italy | 13,5 | 14,5 | 16 | 17 | 18,5 | 30,5 | 42 | 44,5 | 49 | 56 |
| USSR | 37 | 50,5 | 58,8 | 63 | 75 | 81,5 | 87,5 | 98 | 120 | 100 |

5. Global merchandise export, in 2000 prices, billion dollars.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1900 | 1913 | 1929 | 1938 | 1950 | 1960 | 1970 | 1980 | 1990 | 2000 |
| Germany | 21,5 | 54 | 58 | 64,1 | 36,5 | 87,5 | 185 | 385 | 600 | 710 |
| France | 22 | 28,5 | 40,5 | 40 | 31,5 | 62,5 | 140 | 235 | 330 | 420 |
| Great Britain | 38,5 | 54,5 | 73 | 76 | 66 | 105 | 160 | 235 | 320 | 400 |
| Belgium | 12,2 | 15,5 | 18,4 | 16,8 | 12,3 | 27,5 | 63 | 112 | 176 | 214 |

**Requirements for submitting the laboratory work**

1. Tasks are divided by students. You make one graph of a function, surface, polar, 2nd order surface, bar chart. 1 student makes 5 graphs.
2. Task number is the student number in the university list (alphabetical). If you don't have your number, take it in a cycle. For example - the 1st, 6th, 11th students take the first bar chart task, the 2nd, 7th and 12th - the second bar chart, and so on.
3. To submit the laboratory work, you need to send the completed tasks to the email [ruslan.babudzhan@infiz.khpi.edu.ua](mailto:ruslan.babudzhan@infiz.khpi.edu.ua)
4. The completed task includes Python code and a report.
5. Please exclusively English for report, code comments and plot labels.
6. The report submission format is one of the following:
   1. A .doc/.docx file;
   2. A link to a document in Office 365;
   3. A link to Google Docs.
7. The report should contain the text of the task and the obtained graphs. Do not insert code snippets into the report.
8. The format for submitting code is one of the following:
   1. A .py file and requirements.txt with your version of matplotlib;
   2. A Jupyter notebook (.ipynb file);
   3. A link to Google Collaboratory;
   4. A link to GitHub with a \*.py file and requirements.txt;
   5. A link to GitHub with a Jupyter notebook file.
9. If you are submitting code through a Jupyter notebook, specify the installation of your version of matplotlib in the first cell. For example:

!pip install matplotlib==3.8.3