

Звіт до лабораторної роботи №4
з предмету "Моделі інтелектуальних систем"

Дровольського Ярослава, ІПС-42

Варіант №2

Matplotlib was installed in lab3, so we don't need to do it now

Import Matplotlib and NumPy

```
In [2]: import matplotlib.pyplot as plt
import numpy as np
```

Дано таку систему рівнянь:

$$\begin{cases} 6.2x_1 + 6.6x_2 = 83 \\ -9.6x_1 + 13.8x_2 = 72 \\ -13.2x_1 + 5.7x_2 = 305 \end{cases}$$

Перетворимо її (виразимо x_2 через x_1 у кожному рівнянні):

$$\begin{cases} x_2 = \frac{83-6.2x_1}{6.6} \\ x_2 = \frac{72+9.6x_1}{13.8} \\ x_2 = \frac{305+13.2x_1}{5.7} \end{cases}$$

```
In [14]: def f1(x):
return (83 - 6.2*x)/6.6

def f2(x):
return (72 + 9.6*x)/13.8

def f3(x):
return (305 + 13.2*x)/5.7

f1_text = '$6.2 x_{1} + 6.6 x_{2} = 83$'
f2_text = '$-9.6 x_{1} + 13.8 x_{2} = 72$'
f3_text = '$-13.2 x_{1} + 5.7 x_{2} = 305$'
```

Завдання 1

```
In [94]: import matplotlib.ticker as ticker

%matplotlib inline

def task_1():
    x = np.linspace(-32, 10, 100)

    fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(20, 5))

    axes[0].plot(x, f1(x), color='#2596be', linestyle='dotted', linewidth=2)
    axes[1].plot(x, f2(x), color='#009933', linestyle='dashed', linewidth=2)
    axes[2].plot(x, f3(x), color='#cc99ff', linestyle='dashdot', linewidth=2)

    # add titles for plot and for each of axes; set ylim for each of axes
    axes[0].set(ylim=(-5,20), title=f1_text)
    axes[1].set(ylim=(-5,20), title=f2_text)
    axes[2].set(ylim=(-5,20), title=f3_text)
    fig.suptitle('Each line on separate plot \n', fontsize=16, y=1)

    # format ticks on Y axis
```

```

axes[0].yaxis.set_ticks(range(-5, 20, 5))
axes[1].yaxis.set_ticks(range(-5, 20, 5))
axes[2].yaxis.set_ticks(range(-5, 20, 5))
formatter = ticker.NullFormatter()
axes[1].yaxis.set_major_formatter(formatter)
axes[2].yaxis.set_major_formatter(formatter)

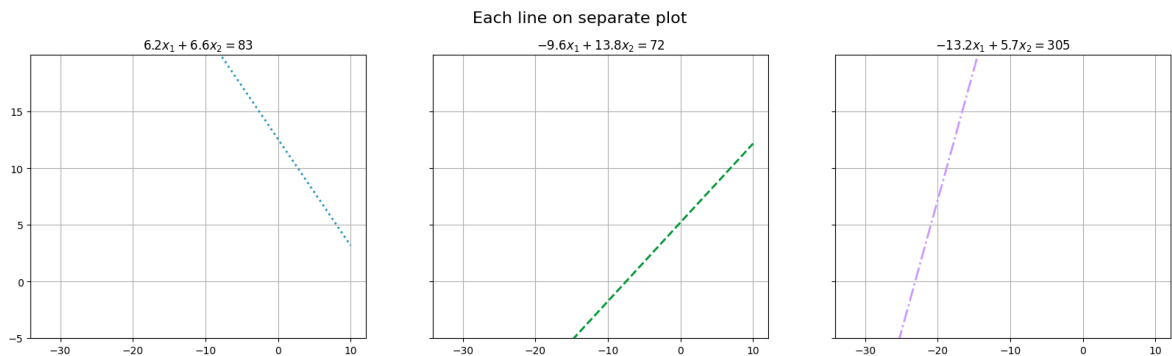
# add grid
axes[0].grid()
axes[1].grid()
axes[2].grid()

# save image
plt.savefig('plot1.jpg')
plt.savefig('plot1.png')
plt.savefig('plot1.svg')

#plt.tight_layout()
plt.show()

```

task_1()



Завдання 2

In [93]: `import numpy as np`

`%matplotlib inline`

```

# find intersection of lines with values y1(x1) and y2(x2)
# where x1 is vector of independent values, and y1 is vector of values of functi
# each of vectors x1, y1, x2, y2 must contain at least 2 elements
# source: https://stackoverflow.com/a/65144877
def find_intersection(x1, y1, x2, y2):
    # Line 1 passing through points p1 and p2
    p1 = [x1[0], y1[0]]
    p2 = [x1[-1], y1[-1]]

    # Line 2 passing through points p3 and p4
    p3 = [x2[0], y2[0]]
    p4 = [x2[-1], y2[-1]]

    # Line 1 dy, dx and determinant
    a11 = (p1[1] - p2[1])
    a12 = (p2[0] - p1[0])
    b1 = (p1[0]*p2[1] - p2[0]*p1[1])

    # Line 2 dy, dx and determinant

```

```

a21 = (p3[1] - p4[1])
a22 = (p4[0] - p3[0])
b2 = (p3[0]*p4[1] - p4[0]*p3[1])

# Construction of the linear system
# coefficient matrix
A = np.array([[a11, a12],
              [a21, a22]])

# right hand side vector
b = -np.array([b1,
              b2])

return np.linalg.solve(A,b)

def task_2():
    # fill values of functions
    x = np.linspace(-32, 10, 200)
    y1 = f1(x)
    y2 = f2(x)
    y3 = f3(x)

    plt.figure(figsize=(20, 16), dpi=300) # change plot size (in inches) and plo
    ax = plt.axes()
    ax.grid(color='grey', linestyle='solid', linewidth=0.2) # set grid lines

    # set axis labels and plot title
    ax.set_xlabel('$x_{1}$', fontsize=16)
    ax.set_ylabel('$x_{2}$', fontsize=16)
    ax.set_title('All lines in one figure', fontsize=20, fontweight='bold')

    # plot equations
    ax.plot(x, y1, color='#2596be', linestyle='dotted', label='f[1]', linewidth=
    ax.plot(x, y2, color='#009933', linestyle='dashed', label='f[2]', linewidth=
    ax.plot(x, y3, color='#cc99ff', linestyle='dashdot', label='f[3]', linewidth

    # add Label for each equation line along line. Note, coordinates is in terms
    plt.text(-25, 32, '$f[1]: 6.2 x_{1} + 6.6 x_{2} = 83$', rotation = -18, font
    plt.text(-10, -4, '$f[2]: -9.6 x_{1} + 13.8 x_{2} = 72$', rotation = 15, fon
    plt.text(0, 55, '$f[3]: -13.2 x_{1} + 5.7 x_{2} = 305$', rotation = 40, font

    # find intersection between equation lines
    int_12 = find_intersection(x, y1, x, y2)
    int_13 = find_intersection(x, y1, x, y3)
    int_23 = find_intersection(x, y2, x, y3)

    # fill area between three lines. Based on specific VALUES, it is NOT GENERAL
    x_fill_area = np.linspace(int_23[0], int_12[0], 100)
    plt.fill_between(x_fill_area, f2(x_fill_area), np.minimum(f1(x_fill_area), f

    # paint intersection points, arrows for them, labels for them
    ax.scatter(int_12[0], int_12[1], color='#ff9933', marker='o', linewidths=[7]
    ax.scatter(int_13[0], int_13[1], color='#0039e6', marker='o', linewidths=[7]
    ax.scatter(int_23[0], int_23[1], color='#663300', marker='o', linewidths=[7]
    ax.arrow(int_23[0], 16, 0, int_23[1]-15+1.3, width=0.1, color='black')
    ax.arrow(int_12[0], 30, 0, int_12[1]-30+1.3, width=0.1, color='black')
    ax.arrow(int_13[0], 50, 0, int_13[1]-50+1.3, width=0.1, color='black')
    plt.text(int_23[0], 17, 'The intersection point of \n f[2] and f[3]', rotati
    plt.text(int_13[0], 51, 'The intersection point of \n f[1] and f[3]', rotati

```

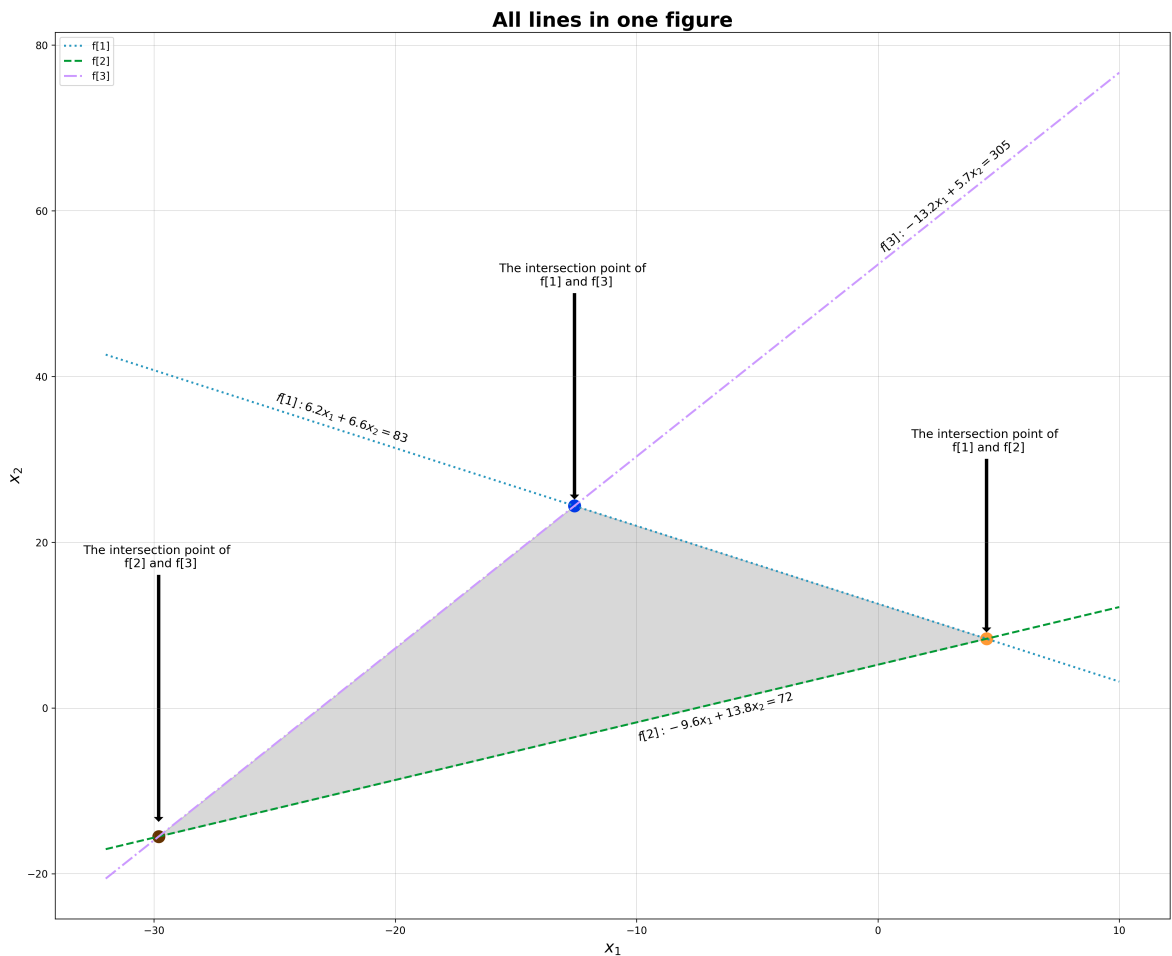
```
plt.text(int_12[0], 31, 'The intersection point of \n f[1] and f[2]', rotati

plt.legend()

# save image
plt.savefig('plot2.jpg')
plt.savefig('plot2.png')
plt.savefig('plot2.svg')

plt.show()
```

task_2()



Порівняння зображень

- **Вид вмісту файлу:** у першому завданні (де три графіки поруч) зображення обрізане точно по тексту; а в другому завданні наявне біле обрамлення навколо рисунку.
- **Якість** без прибиження у всіх однакова. При прибиженні у svg-файлі якість не втрачається, оскільки це векторний формат. А при прибиженні якість png-файлу краща, ніж jpg.
- Без прибиження помітних **артефактів** не виявлено. У jpg-файлі при прибиженні видно чорні пікселі навколо тексту та ліній.
- **Розмір:** Будемо порівнювати по другому завданню, оскільки там картинка більша і тому порівняння буде якісніше. jpg-файл важить найбільше (732 Kб), далі png - 522 Kб, а svg - 52 Kб.

Висновок: кращим з-поміж двох растрових форматів (png та jpg) для зберігання рисунків є png: він якісніший та має менший розмір.