Lab 1 Report

python 绘图使用

王世炟

PB20151796

April 8, 2023

Part 1: 实验要求

本次实验要求使用 pylab/matplotlib 库进行绘图，绘制 $1 \leq x \leq 100$ 范围内 $f(x) = x, g(x) = \log(x), h(x) = 1.05^x, k(x) = \frac{100}{x}$ 四个函数的图表。所有曲线绘制在同一张图表中，要求绘制的图表包含图标题和坐标轴标题，实验代码中需要设置字体大小、数字大小、坐标轴刻度大小以及标记点大小。曲线的图表需要对每一条曲线进行标注。

Part 2: 实验环境

本次实验的实验环境为：

- Windows10 64 位
- IDE：VS Code
- python==3.9.5
- jupyter==1.0.0
- matplotlib==3.5.1
- numpy==1.23.0

Part 3: 实验内容

3.1 实验设计

首先要搞清楚 matplotlib 画函数图像的原理。在 matplotlib.pyplot(以下简称 plt) 中，存在这样的一个函数 plt.plot(), 它按照输入的 X，Y 值进行绘图，将每一个点按照先后顺序用折线连接起来。所以基本的想法是生成一个区间内足够多的等距点，用折线模拟曲线。而其他的要求（字体、标注等）可以通过 plt 中的函数实现。

3.2 算法流程

1. 导入 matplotlib.pyplot 以及 numpy 库
2. 生成 x 的值
3. 定义函数 $f(x), g(x), h(x), k(x)$
4. 绘制曲线、标注曲线、设置坐标轴刻度大小范围及其字体大小、设置网格线、设置标题、设置图例
5. 显示图像

3.3 核心代码解释

导入库：

```
import matplotlib.pyplot as plt
import numpy as np
```

生成 x 的值、定义 $f(x), g(x), h(x), k(x)$：

```
x = np.arange(1, 100, 0.01)                    # 生成x的取值
g, h, k = np.log2(x), 1.05**x, 100/x           # g(x),h(x),k(x) 的值
```

设置字体和数字大小：

```
plt.rcParams.update({"font.size": 12, "axes.labelsize": 14}) # 设置字体和数字大小
```

画出曲线，利用 plt.plot() 函数，并设置相应的参数（前两个参数为 x 和 y 值，color 参数代表颜色，label 参数会在后面的图例中显示，linestyle 参数代表曲线的样式，markersize 代表标记点的大小）：

```
plt.plot(x, x, color='r', label='f(x)=x', markevery=x1 , marker='o',
    markersize=5)                # f(x)=x
plt.plot(x, g, color='c', label='g(x)=log(x)',
         markevery=x1 , marker='s', linestyle='—', markersize=5)
                                              # g(x)=log(x)
plt.plot(x, h, color='b', label='h(x)=1.05^x',
         markevery=x1 , marker='*', linestyle=':', markersize=5)
                                               # h(x)=1.05^x
plt.plot(x, k, color='orange', label='k(x)=100/x',
         markevery=x1 , marker='D', linestyle='-.', markersize=5)
```

标注曲线，利用 plt.annotate() 函数，该函数并设置相应的参数（第一个参数为标注的文字，xy 参数为被注释点的坐标，xytext 为注释文本的坐标，arrowprops 为箭头的属性）：

```
    # 标注曲线
    plt.annotate('f(x)=x', xy=(40, 40), xytext=(35, 60),xycoords='data',
        arrowprops=dict(facecolor='red', shrink=0.05))

    plt.annotate('g(x)=log(x)', xy=(90, 7), xytext=(85, 25),xycoords='data'
        ,arrowprops=dict(facecolor='c', shrink=0.05))

    plt.annotate('h(x)=1.05^x', xy=(60, 20), xytext=(55, 40),xycoords='data
        ',arrowprops=dict(facecolor='b', shrink=0.05))

    plt.annotate('k(x)=100/x', xy=(3, 60), xytext=(10, 70),xycoords='data',
        arrowprops=dict(facecolor='orange', shrink=0.05))
```

设置图像中的特定字体大小：

```
    x_scale = plt.MultipleLocator(10)
    y_scale = plt.MultipleLocator(20)
    ax = plt.gca()
    ax.xaxis.set_major_locator(x_scale)  # 设置x轴刻度大小
    ax.yaxis.set_major_locator(y_scale)  # 设置y轴刻度大小
    plt.xlim((0, 100))                   # 设置x轴刻度范围
    plt.ylim((0, 140))                   # 设置y轴刻度范围
    plt.xticks(fontsize=10)              # 设置x轴刻度字体大小
    plt.yticks(fontsize=10)              # 设置y轴刻度字体大小

    ax.set_xlabel(..., fontsize=15)      # 设置xlabel的字体大小
    ax.set_ylabel(..., fontsize=15)      # 设置ylabel的字体大小
    plt.xlabel('x', loc='right')         # 设置xlabel
    plt.ylabel('y', loc='top')           # 设置ylabel

    plt.grid(axis='both', alpha=0.5)     # 显示网格线
    plt.title('Function Image', loc='center', fontsize=13)  # 图片标题设置
```

设置图例：

```
    # 图例设置
    plt.legend(bbox_to_anchor=(0, 1.06, 1, 0.2), loc='lower left',
               mode='expand', framealpha=0.5, ncol=2)
```

显示图片：

```
    # 显示图片
    plt.show()
```

Part 4: 实验结果

运行上述代码可得到以下结果图像

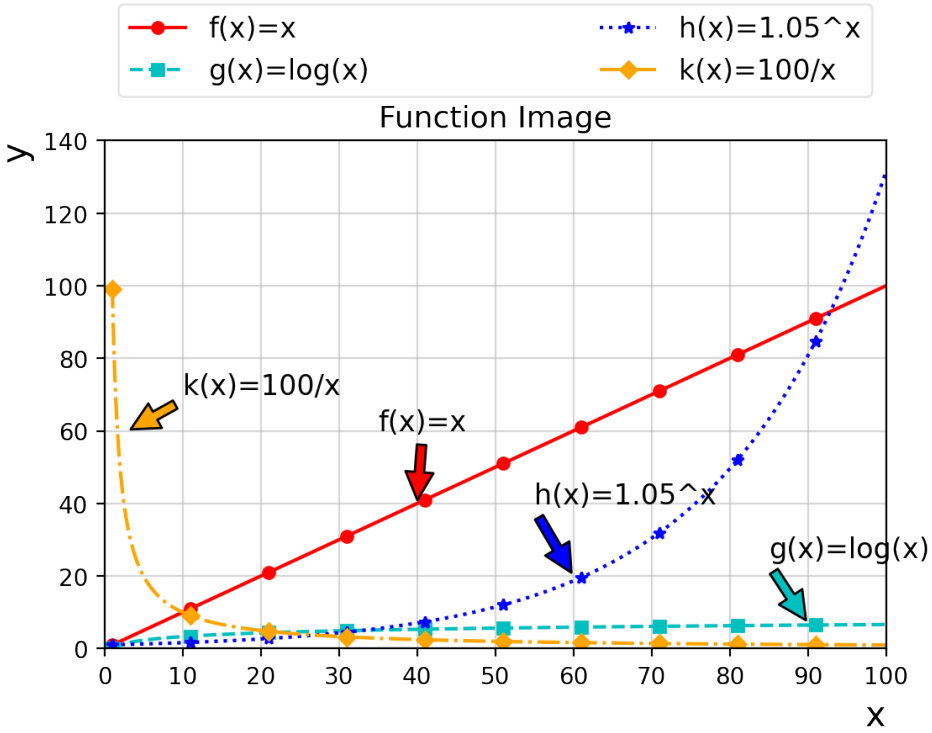

图 1: Output

Part 5: 实验总结

通过本次实验，学会了如何使用 python 进行绘图，熟悉了常用的函数以及其参数。对于 matplotlib 的使用更加熟练。