1.

import matplotlib.pyplot as plt

from scipy.optimize import curve\_fit

import numpy as np

x = [0, 1, 2, 3, 4]

y = [1, 1.8, 1.3, 2.0, 6.3]

def prbl(x, a, b, c):

return a + b \* x + c \* x \*\* 2

param, cov = curve\_fit(prbl, x, y)

x1 = np.linspace(min(x), max(x), 100)

y1 = prbl(x1, param[0], param[1], param[2])

plt.grid()

plt.scatter(x, y, label = "analytical solution")

plt.plot(x1, y1, 'r', label = "numerical solution")

plt.xlabel("x")

plt.ylabel("y")

plt.legend()

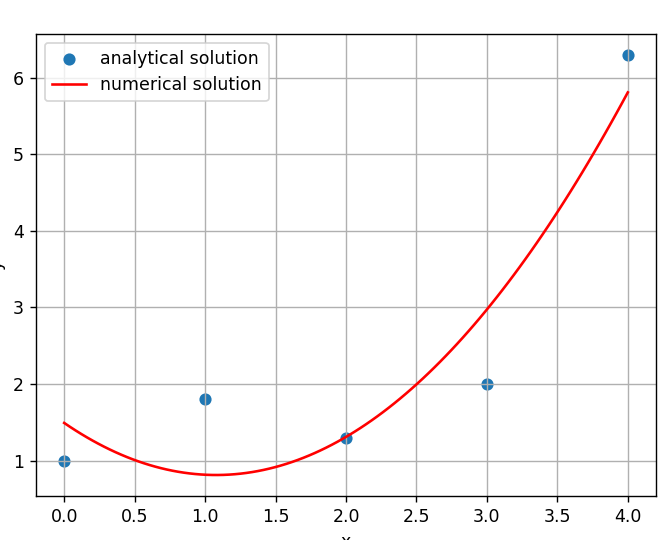
plt.show()

print("answers:")

print("a =", param[0], "b =", param[1], "c =", param[2])

answers:

a = 1.4914285774973124 b = -1.2628571501417563 c = 0.5857142872927956



2.

import matplotlib.pyplot as plt

from scipy.optimize import curve\_fit

import numpy as np

def line(x, a, b):

return a + b \* x

x = [6, 7, 7, 8, 8, 8, 9, 9, 10]

y = [5, 5, 4, 5, 4, 3, 4, 3, 3]

param, cov = curve\_fit(line, x, y)

x1 = np.linspace(min(x), max(x), 100)

y1 = line(x1, param[0], param[1])

plt.grid()

plt.scatter(x, y, label = "analytical solution")

plt.plot(x1, y1, 'r', label = "numerical solution")

plt.xlabel("x")

plt.ylabel("y")

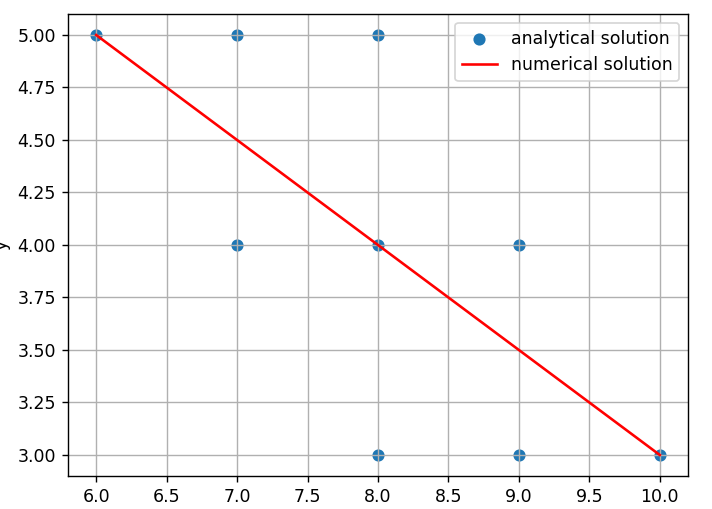
plt.legend()

plt.show()

print("answers:")

print("a =", param[0], "b =", param[1])

a = 8.000000000015271 b = -0.500000000003272



3.

import matplotlib.pyplot as plt

from scipy.optimize import curve\_fit

import numpy as np

def curve(a, b, x):

return a\*np.exp(b\*x)

x = [0, 1, 2, 3]

y = [1.05, 2.10, 3.85, 8.30]

param, cov = curve\_fit(curve, x, y)

x1 = np.linspace(min(x), max(x), 100)

y1 = curve(x1, param[0], param[1])

plt.grid()

plt.scatter(x, y, label = "analytical solution")

plt.plot(x1, y1, 'r', label = "numerical solution")

plt.xlabel("x")

plt.ylabel("y")

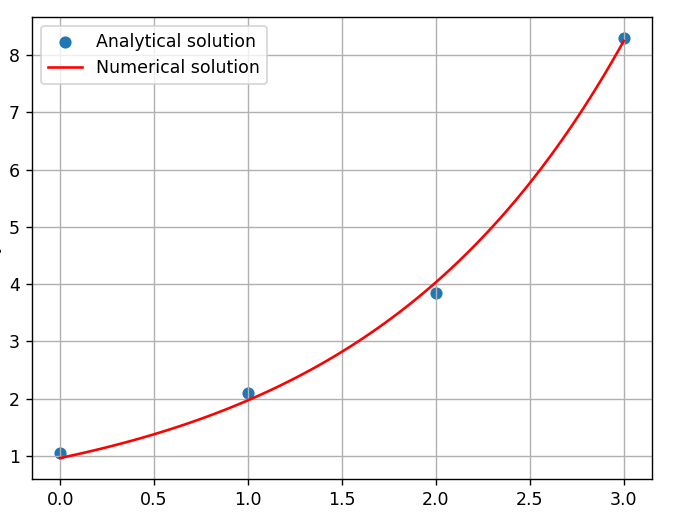
plt.legend()

plt.show()

print("answers:")

print("a =", param[0], "b =", param[1])

a = 0.9651136818052145 b = 0.7152344482141445



4.

import matplotlib.pyplot as plt

from scipy.optimize import curve\_fit

import numpy as np

x = [1, 2, 3, 4, 5]

y = [1.8, 5.1, 8.9, 14.1, 19.8]

def line(x, a, b):

return a \* x + b \* x \*\* 2

param, cov = curve\_fit(line, x, y)

x1 = np.linspace(min(x), max(x), 100)

y1 = line(x1, param[0], param[1])

plt.grid()

plt.scatter(x, y, label="analytical solution")

plt.plot(x1, y1, 'r', label="numerical solution")

plt.xlabel("x")

plt.ylabel("y")

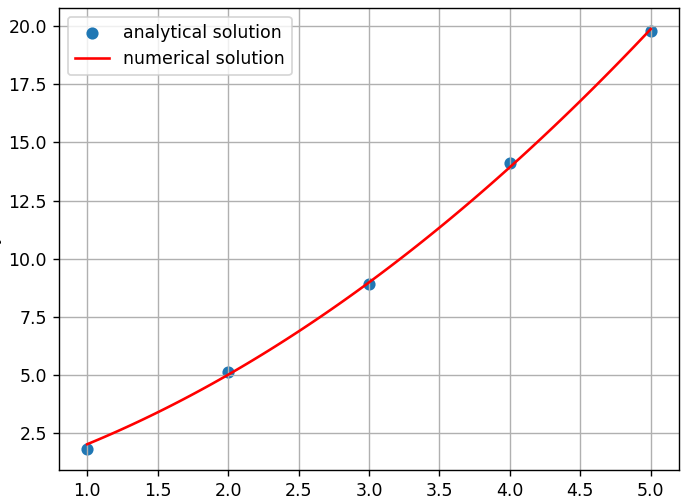
plt.legend()

plt.show()

print("answers:")

print("a =", param[0], "b =", param[1])

a = 1.5128571428582616 b = 0.49285714285603677



5.

import matplotlib.pyplot as plt

from scipy.optimize import curve\_fit

import numpy as np

def curve(x, a, b):

return a \* x + b / x

x = [1,2,3,4,5,6,7,8]

y = [5.4,6.3,8.2,10.3,12.6,14.9,17.3,19.5]

param, cov = curve\_fit(curve, x, y)

x1 = np.linspace(min(x), max(x), 100)

y1 = curve(x1, param[0], param[1])

plt.grid()

plt.scatter(x, y, label = "analytical solution")

plt.plot(x1, y1, 'r', label = "numerical solution")

plt.xlabel("x")

plt.ylabel("y")

plt.legend()

plt.show()

print("answers:")

print("a =", param[0], "b =", param[1])

a = 2.398526381328391 b = 3.0000772758586085

