

Факультет программной инженерии и компьютерной техники

Вычислительная математика

Лабораторная работа №2

Выполнил: Орехов Сергей Владимирович

Р32151

Преподаватель: Машина Екатерина Алексеевна

Санкт-Петербург

2023

Цель работы

Изучить численные методы решения нелинейных уравнений и их систем, найти корни заданного нелинейного уравнения/системы нелинейных уравнений, выполнить программную реализацию методов.

Листинг методов:

import sys  
  
import matplotlib.pyplot as plt  
import numpy as np  
import math  
  
  
def check\_interval(func, a, b):  
 print("f(a):%f", func(a))  
 print("f(b):%f", func(b))  
 if func(a) \* func(b) < 0:  
 return True  
 else:  
 return False  
  
  
def chords(func, a, b):  
 if check\_interval(func, a, b) == False:  
 return "Неправильный интервал!"  
  
 xnpy = np.linspace(a, b, 100)  
 ynpy = func(xnpy)  
 fig = plt.figure()  
 ax = fig.add\_subplot(1, 1, 1)  
 ax.spines['left'].set\_position('center')  
 ax.spines['bottom'].set\_position('center')  
 ax.spines['right'].set\_color('none')  
 ax.spines['top'].set\_color('none')  
 ax.xaxis.set\_ticks\_position('bottom')  
 ax.yaxis.set\_ticks\_position('left')  
 plt.plot(xnpy, ynpy, 'g')  
  
  
 eps = float(input("Введите эпсилон:\n").replace(",", "."))  
 x = 0  
 while abs(a - b) > eps:  
 fa = func(a)  
 fb = func(b)  
  
 x = (a \* fb - b \* fa) / (fb - fa)  
 fx = func(x)  
 if fa \* fx > 0 > fb \* fx:  
 a = x  
 elif fb \* fx > 0 > fa \* fx:  
 b = x  
 elif func(x) == 0:  
 plt.plot(x, func(x), 'ro')  
 plt.show()  
 return x  
 else:  
 sys.exit("В методе хорд произошла ошибка\n")  
 plt.plot(x, func(x), 'ro')  
 plt.show()  
 return x  
  
  
def simple\_iter(func, a, b):  
  
 if (check\_interval(func, a, b) == False):  
 return "Неправильный интервал!"  
  
 xnpy = np.linspace(a, b, 100)  
 ynpy = func(xnpy)  
 fig = plt.figure()  
 ax = fig.add\_subplot(1, 1, 1)  
 ax.spines['left'].set\_position('center')  
 ax.spines['bottom'].set\_position('center')  
 ax.spines['right'].set\_color('none')  
 ax.spines['top'].set\_color('none')  
 ax.xaxis.set\_ticks\_position('bottom')  
 ax.yaxis.set\_ticks\_position('left')  
 plt.plot(xnpy, ynpy, 'g')  
  
 eps = float(input("Введите эпсилон:\n").replace(",", "."))  
 deriv\_a = (func(a + eps / 100) - func(a)) / (eps / 100)  
 deriv\_b = (func(b + eps / 100) - func(b)) / (eps / 100)  
 if (deriv\_a > deriv\_b):  
 max\_deriv = deriv\_a  
 xi = a  
 else:  
 max\_deriv = deriv\_b  
 xi = b  
 lamda = -1 / max\_deriv  
 itercount = 0  
 while (abs(xi + lamda \* func(xi) - xi) > eps):  
 if itercount > 1000:  
 return xi + lamda \* func(xi)  
 xi = xi + lamda \* func(xi)  
 itercount += 1  
 plt.plot(xi + lamda \* func(xi), func(xi + lamda \* func(xi)), 'ro')  
 plt.show()  
 return xi + lamda \* func(xi)  
  
  
def sec(func, a, b):  
  
 if (check\_interval(func, a, b) == False):  
 return "Неправильный интервал!"  
  
 xnpy = np.linspace(a, b, 100)  
 ynpy = func(xnpy)  
 fig = plt.figure()  
 ax = fig.add\_subplot(1, 1, 1)  
 ax.spines['left'].set\_position('center')  
 ax.spines['bottom'].set\_position('center')  
 ax.spines['right'].set\_color('none')  
 ax.spines['top'].set\_color('none')  
 ax.xaxis.set\_ticks\_position('bottom')  
 ax.yaxis.set\_ticks\_position('left')  
 plt.plot(xnpy, ynpy, 'g')  
  
 eps = float(input("Введите эпсилон:\n").replace(",", "."))  
  
 xi\_2 = b  
 xi\_1 = b - abs(b-a)/8  
 xi = xi\_1 - func(xi\_1)\*(xi\_1-xi\_2)/(func(xi\_1)-func(xi\_2))  
  
 while abs(xi - xi\_1) <= eps:  
 tmp\_1 = xi  
 tmp\_2 = xi\_1  
 xi = xi\_1 - func(xi\_1)\*(xi\_1-xi\_2)/(func(xi\_1)-func(xi\_2))  
 xi\_2 = tmp\_2  
 xi\_1 = tmp\_1  
 plt.plot(xi, func(xi), 'ro')  
 plt.show()  
 return xi

Пример работы:

Пример 1:

Выберите уравнение или систему(s1 - система 1, s2 - система 2, f1 - уравнение 1, f2 - уравнение 2, f3 - уравнение 3):

f1

Выберите метод(хорд - cd, секущих - sc, простой итерации - si)cd

Выберите интервал(<a> <b>): 0 2

f(a):%f 0.0

f(b):%f 8.0

Неправильный интервал!

Пример 2:

Выберите уравнение или систему(s1 - система 1, s2 - система 2, f1 - уравнение 1, f2 - уравнение 2, f3 - уравнение 3):

f1

Выберите метод(хорд - cd, секущих - sc, простой итерации - si)sc

Выберите интервал(<a> <b>):-1 1

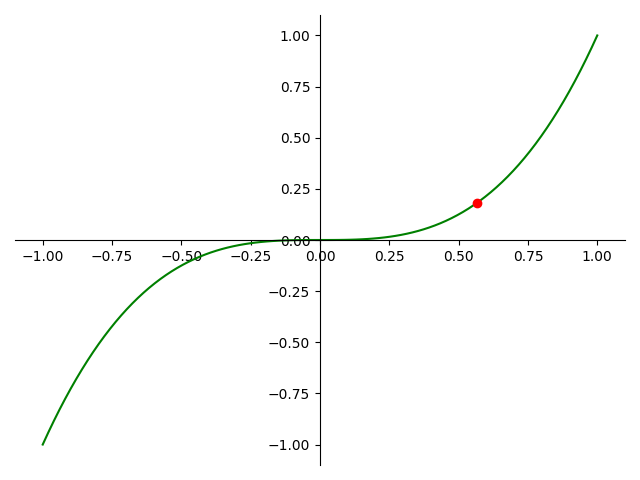
f(a):%f -1.0

f(b):%f 1.0

Введите эпсилон:

0.01

0.5675675675675675



Пример 3:

Выберите уравнение или систему(s1 - система 1, s2 - система 2, f1 - уравнение 1, f2 - уравнение 2, f3 - уравнение 3):

s1

x^2 + y^2 - 4 = 0

3x^2 - y = 0

Введите эпсилон:

0.01

Введите X0!

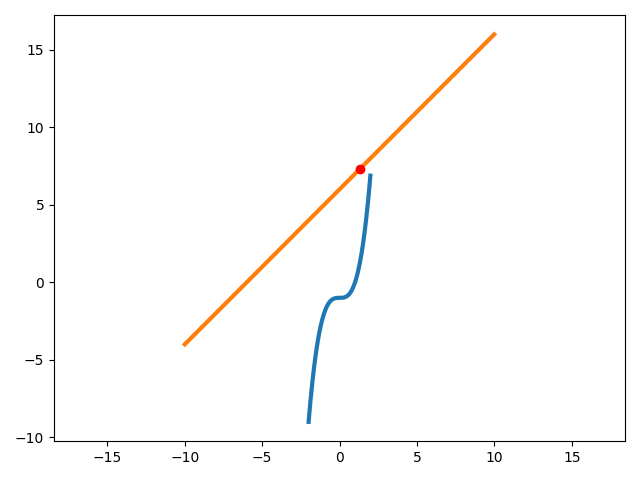
1

Введите Y0!

1

1.2570133745218284 7.0

X: 1.319716223782242 Y: 7.317888778568854



Вывод

В результате выполнения данной лабораторной работой я научился считать нелинейные уравнения и их системы.