|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | |  |  |  | | Министерство образования и науки РОССИЙСКОЙ ФЕДЕРАЦИИ | | |  Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА – Российский технологический университет»** | |
|  | |
|  | |
|  |  |

ИНСТИТУТ КИБЕРНЕТИКИ

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Курсовая работа

по курсу «Численные методы»

Тема: **Вычисление минимума функции, зависящей от интерполяционного многочлена таблично заданной функции**

Выполнил:

Студент 3-го курса

Петров С.В.

Группа: КМБО-03-17

МОСКВА 2020

**Задание**

Вычислить минимум функции на отрезке *[a, b]* с точностью ; *P(x)* – интерполяционный многочлен для функции *f(x)*, заданной таблично; *k = P(c)*.

Исходные данные:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | 1.05 | 1.15 | 1.25 | 1.35 |
| f(x) | 2.30 | 2.74 | 3.46 | 4.60 |
| *a* | *b* | *c* |  |  |
| 1.05 | 1.35 | 1.10 | 0.001 |  |

**План**

1. Теоретические сведения………………………………………………………………4
   1. Задача интерполяции………………………………………………………….4
   2. Многочлен Ньютона…………………………………………………………..5
2. Ход решения…………………………………………………………………………...6
3. Список литературы……………………………………………………………………10
4. Приложение (листинг программы)…………………………………………………..11

**Теоретические сведения**

Минимум функции – такая точка, что для всех точек в ее окрестности справедливо неравенство *f(x) ≥ f()*, – точка минимума функции

Одним из численных методов поиска минимума функции на отрезке является «метод золотого сечения». Он отличается высокой скоростью сходимостью и компактностью программной реализации.

Пусть *f(x)* задана и кусочно-непрерывна на [,], и имеет на этом отрезке только один локальный минимум. Золотое сечение, о котором упоминал ещё Евклид, состоит в разбиении интервала [L, ] точкой на две части таким образом, что отношение длины всего отрезка к его большей части равно отношению большей части к меньшей:

Таким образом, возьмем на отрезке две точки и , симметрично относительно границ делящие исходный отрезок в отношении золотого сечения:

Где

Если  *<*, мы должны сузить отрезок справа, т.е. новое значение *R =* , в противном случае = . Оставшаяся внутри нового отрезка точка является первым приближением к минимуму и делит этот отрезок в отношении золотого сечения. Таким образом, на каждой итерации приближения к минимуму нам нужно ставить только одну точку ( или ), в которой считать значение функции и сравнивать его с предыдущим. Условием выхода из итерационного процесса будет условие

.

Задача интерполяции

Пусть функция f(x) задана набором точек () на интервале [a, b]:

Задача: найти функцию F(x), принимающую в точках те же значения . Тогда, условие интерполяции:

*F() =*

При этом предполагается, что среди значений нет одинаковых. Точки называют узлами интерполяции.

Задача нахождения интерполяционной функции *F(x)* имеет много решений, так как через заданные точки можно провести бесконечно много кривых, каждая из которых будет графиком функции, для которой выполнены все условия интерполяции. Для практики важен случай интерполяции функции многочленами:

Искомый полином называется интерполяционным полиномом (многочленом).

При построении одного многочлена для всего рассматриваемого интервала [a, b] для нахождения коэффициентов многочлена необходимо решить систему уравнений, построенную на основе полинома . Данная система содержит n+1 уравнение, следовательно, с ее помощью можно определить n+1 коэффициент. Поэтому максимальная степень интерполяционного многочлена m=n, и многочлен принимает вид:

Многочлен Ньютона

Пусть функция *f(x)* задана с произвольным шагом, и точки таблицы значений пронумерованы в произвольном порядке. Разделенные разности нулевого порядка совпадают со значениями функции в узлах. Разделенные разности первого порядка определяются через разделенные разности нулевого порядка:

Разделенные разности второго порядка определяются через разделенные разности первого порядка:

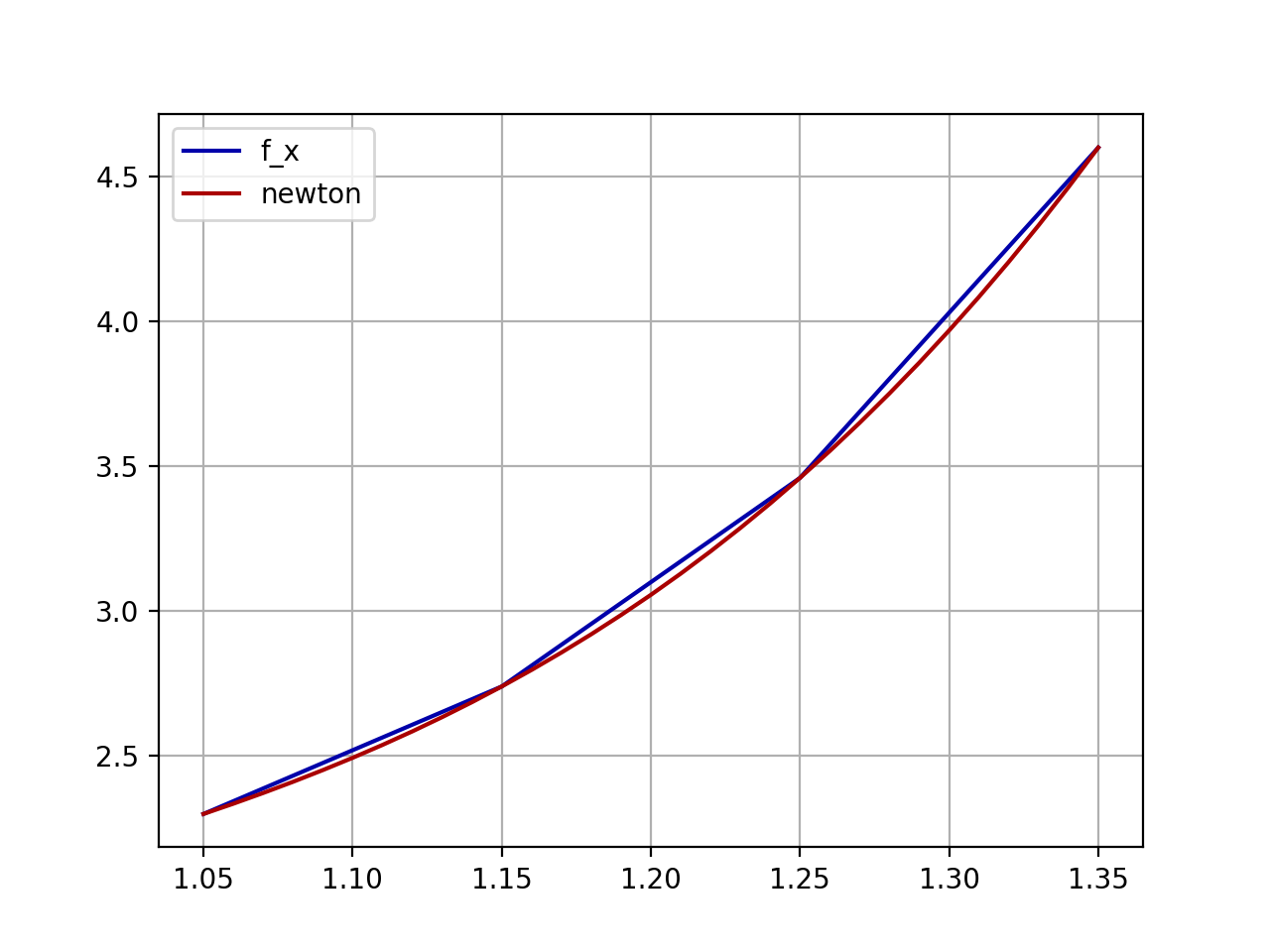
Разделенные разности k порядка определяются через разделенные разности k+1 порядка:

Используя понятие разделенной разности интерполяционный многочлен Ньютона можно записать в следующем виде:

**Ход решения**

Для поиска минимума функции *F(x)* необходимо сперва построить интерполяционный многочлен таблично заданной функции *f(x).*

Результат расчета интерполяционного многочлена Ньютона (синим цветом – график функции *f(x),* красным – многочлен Ньютона):



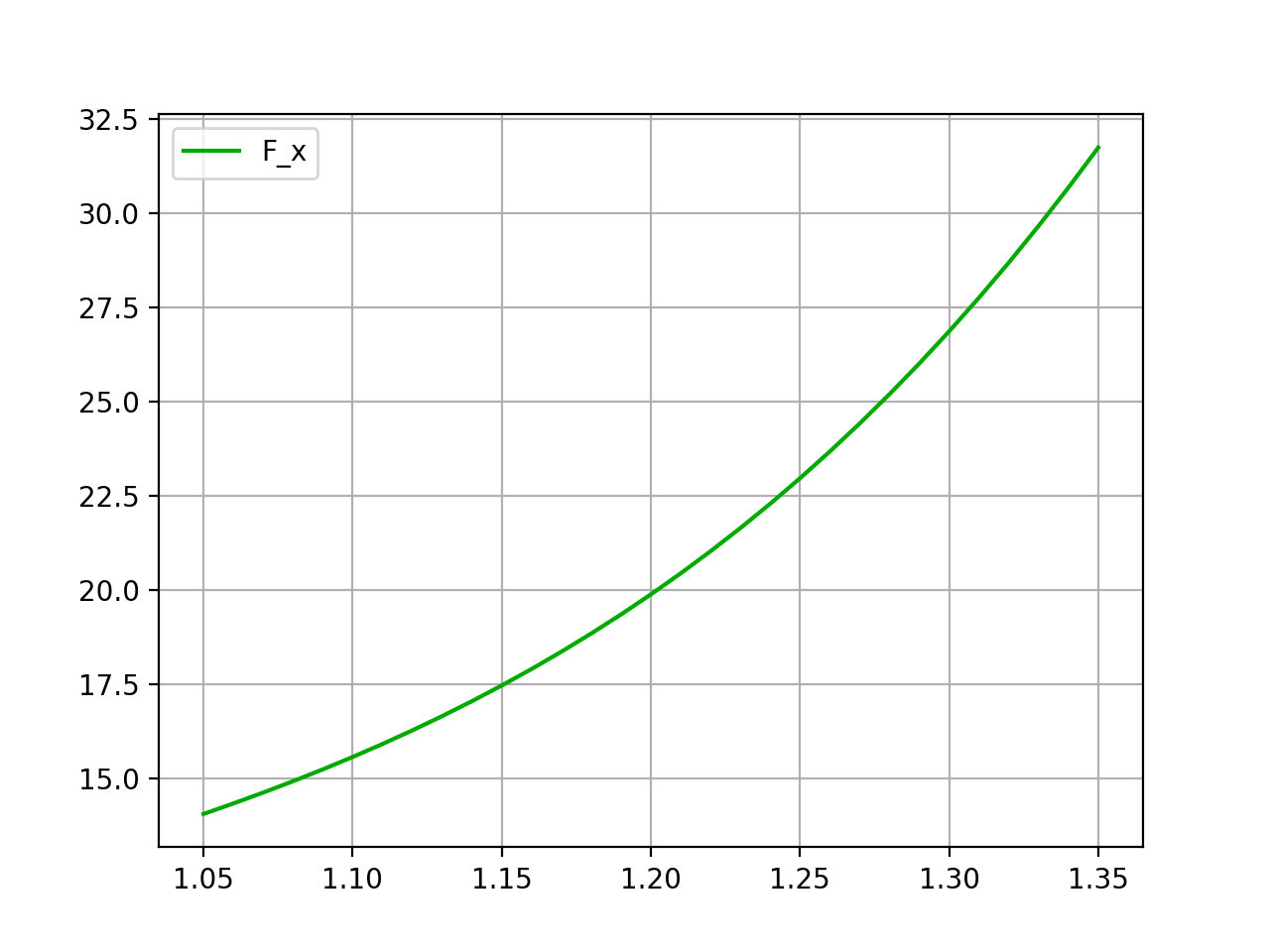
Проверим значения многочлена в узлах интерполяции:

|  |  |  |
| --- | --- | --- |
| x | f(x) |  |
| 1.05 | 2.30 | 2.3 |
| 1.15 | 2.74 | 2.74 |
| 1.25 | 3.46 | 3.46 |
| 1.35 | 4.6 | 4.6 |

Значение многочлена в точке c = 1.10: P(c) = 2.49375

Далее мы можем построить график функции *F(x)* на отрезке *[a, b]*

*(a =* 1.05, b = 1.35*):*



Многочлен Ньютона возрастает на [1.05, 1.35]. Убедимся в этом, построив его вручную (не полагаясь на результат программы):

*Получаем:*

где

где

*Получаем:*

*Получаем:*

Итог:

Этот многочлен положителен и возрастает на рассматриваемом отрезке. Покажем это:

У этого уравнения один вещественный корень: x ~ 0.59726

При x > 0.59726

При x <0.59726

Отсюда получаем, что на отрезке [1.05, 1.35]

Теперь найдем производную:

Получается, что многочлен возрастает и на нашем отрезке.

*k = P(c) = 2.49375 > 0, (a\*x + b) > 0* при любом x, следовательно, точка минимума – это точка a = 1.05. Значение минимума функции на отрезке [1.05, 1.35] совпадает с значением функции в точке 1.05 и равняется 14.0666.

Результат поиска точки минимума с помощью метода золотого сечения:

|  |  |  |  |
| --- | --- | --- | --- |
| Рассматриваемый отрезок | X1 | X2 | Результат шага |
| [1.05, 1.35] | 1.164589 | 1.2354101 | b = x2 |
| [1.05, 1.2354101] | 1.12082 | 1.1645898 | b = x2 |
| [1.05, 1.1645898] | 1.093769 | 1.1208203 | b = x2 |
| [1.05, 1.1208203] | 1.07705 | 1.0937694 | b = x2 |
| [1.05, 1.0937694] | 1.667184 | 1.0770509 | b = x2 |
| [1.05, 1.0770509] | 1.060332 | 1.667184 | b = x2 |
| [1.05, 1.0667184] | 1.056385 | 1.0603325 | b = x2 |
| [1.05, 1.0603325] | 1.053946 | 1.0563858 | b = x2 |
| [1.05, 1.0563858] | 1.052439 | 1.053946 | b = x2 |
| [1.05, 1.053946] | 1.051507 | 1.052439 | b = x2 |
| [1.05, 1.052439] | 1.050931 | 1.051507 | b = x2 |
| [1.05, 1.051507] | 1.050575 | 1.050931 | b = x2 |

В качестве ответа принимаем:

Значение функции в точке:

Абсолютная погрешность ответа: |1.050465 – 1.05| = 0.000465

Относительная погрешность ответа: (0.000465 / 1.05) = 0.000442

**Список литературы**

1. Т.В. Иванова. Численные методы в оптике. Учебное пособие. – СПб: Университет ИТМО, 2017.

2. Акулич И. Л. Математическое программирование в примерах и задачах: Учеб. пособие для студентов эконом. спец. вузов. — М.: Высш. шк., 1986.

3. Коршунов Ю. М. Математические основы кибернетики. — М.: Энергоатомиздат, 1972.

**Приложение**

Листинг программы на языке Python

# main.py

# kurs\_6

#

# Created by Сергей Петров on 02/05/2020.

# Copyright © 2020 SergioPetrovx. All rights reserved.

from numpy import arange

from matplotlib import pyplot

from math import sqrt, fabs

def function(x, p):

k = p(1.1)

a = 1.05

b = 1.35

return k\*(a\*x + b)\*p(x)

def f\_x(x):

if x == 1.05:

return 2.3

elif x == 1.15:

return 2.74

elif x == 1.25:

return 3.46

elif x == 1.35:

return 4.6

def newton(point, x):

def coeffs(x):

n = len(x)

y = list(map(f\_x, x))

k = [0] \* n

k[0] = y[0]

for j in range(1, n):

for i in range(0, n - j):

y[i] = (y[i + 1] - y[i]) / (x[i + j] - x[i])

k[j] = y[0]

return k

n = len(x)

k = coeffs(x)

ans = k[0]

p = 1

for i in range(1, n):

p = p \* (point - x[i - 1])

ans += k[i] \* p

return ans

def main():

eps = 0.001

a = 1.05

b = 1.35

args = [1.05, 1.15, 1.25, 1.35]

P = lambda x: newton(x, args)

x = [i for i in arange(args[0], args[-1], 0.01)]

y = [function(i, P) for i in x]

print(x)

print(y)

pyplot.grid(True)

pyplot.plot(x, y, color="#00AA00", label='F\_x')

pyplot.plot(args, [f\_x(i) for i in args], color="#0000AA", label='f\_x')

pyplot.plot(x, [P(i) for i in x], color="#AA0000", label='newton')

pyplot.legend()

pyplot.show()

print([P(i)for i in args])

print(P(1.1))

F = (sqrt(5) - 1) / 2

table = []

while fabs(b - a) > eps:

x1, x2 = a + (1 - F) \* (b - a), a + F\*(b - a)

table.append([a, b, x1, x2])

if function(x1, P) >= function(x2, P):

a = x1

else:

b = x2

print("Минимум функции: ", (a + b) / 2)

print("Значение функции: ", function((a+b) / 2, P))

print("Точное значение: ", function(1.05, P))

print(P((a+b) / 2))

for i in table:

print(i)

if \_\_name\_\_ == "\_\_main\_\_":

main()