**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

по дисциплине «Методы численного анализа»

на тему: «Численные методы поиска экстремума унимодальной функции»

Выполнил: студент гр. ИП-22

Коваленко А.И.

Принял: преподаватель

Процкая М.А.

Гомель 2022

**Цель работы**: получить навыки поиска экстремумов унимодальных функций с использованием численных методов, выполнять графическую интерпретацию результатов.

***Задание 1***

Для функций по индивидуальному заданию:

-- найти координаты точки экстремума методом золотого сечения и методом Фибоначчи;

-- дать графическую интерпретацию результатов;

-- сравнить количество итераций при реализации каждого метода при разных значениях точности, сделать выводы об эффективности методов

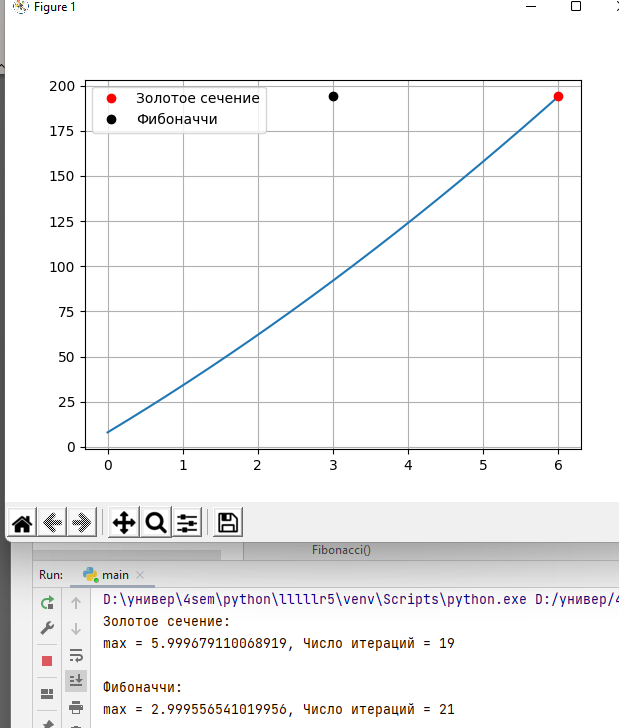
*Таблица 1*

|  |  |  |
| --- | --- | --- |
| Вариант | функция | интервал изоляции, точность |
| 7. |  | a=0 b=6  ε=0.1 |

Листинг программы:

**import** matplotlib.pyplot **as** plt  
**import** numpy **as** np  
  
  
**def** f(x):  
 **return** x\*\*2 +25\*x +8  
  
  
**def** FibonacciNumber(n):  
 **if** (n == 0):  
 **return** 0  
 **if** n **in** (1, 2):  
 **return** 1  
 **return** FibonacciNumber(n - 1) + FibonacciNumber(n - 2)  
  
  
**def** Fibonacci(a, b, eps):  
 n = 2  
 **while** (np.fabs(b - a) > eps):  
 x1 = a + (b - a) \* FibonacciNumber(n - 2) / FibonacciNumber(n)  
 x2 = a + (b - a) \* FibonacciNumber(n - 1) / FibonacciNumber(n)  
 y1 = f(x1)  
 y2 = f(x2)  
 n = n + 1  
 **if** (y1 < y2):  
 a = x1  
 x1 = x2  
 x2 = b - (x1 - a)  
 y1 = y2  
 y2 = f(x2)  
 **else**:  
 b = x2  
 x2 = x1  
 x1 = a + (b - x2)  
 y2 = y1  
 y1 = f(x1)  
 **return** (x1 + x2) / 2, n  
  
  
**def** GoldenRatio(a, b, eps):  
 n = 0  
 **while** (np.fabs(b - a) > eps):  
 n = n + 1  
 x1 = b - (b - a) / fi  
 x2 = a + (b - a) / fi  
 **if** (f(x1) <= f(x2)):  
 a = x1  
 **else**:  
 b = x2  
 **return** (a + b) / 2, n  
  
  
a = 0  
b = 6  
eps = 0.001  
n = 10  
fi = (1 + np.sqrt(5)) / 2  
min, nmin = GoldenRatio(a, b, eps)  
fmin, fnmin = Fibonacci(a, b, eps)  
print(**f"Золотое сечение:\nmax = {**min**}, Число итераций = {**nmin**}"**)  
print(**f"\nФибоначчи:\nmax = {**fmin**}, Число итераций = {**fnmin**}"**)  
x = np.arange(a, b, eps)  
plt.plot(x, f(x))  
plt.plot(min, f(min), **'ro'**, label=**'Золотое сечение'**, color=**"red"**)  
plt.plot(fmin, f(min), **'go'**, label=**'Фибоначчи'**, color=**"black"**)  
plt.grid()  
plt.legend()  
plt.show()

Результат выполнения



***Задание 2***

Для функций по индивидуальному заданию:

-- найти координаты точки экстремума функции методом Пауэла, дать графическую интерпретацию результатов. Значение ∆ выбрать самостоятельно (можно взять ∆=5);

-- сравнить полученный результат с аналитическим решением;

-- дать графическую интерпретацию результатов;

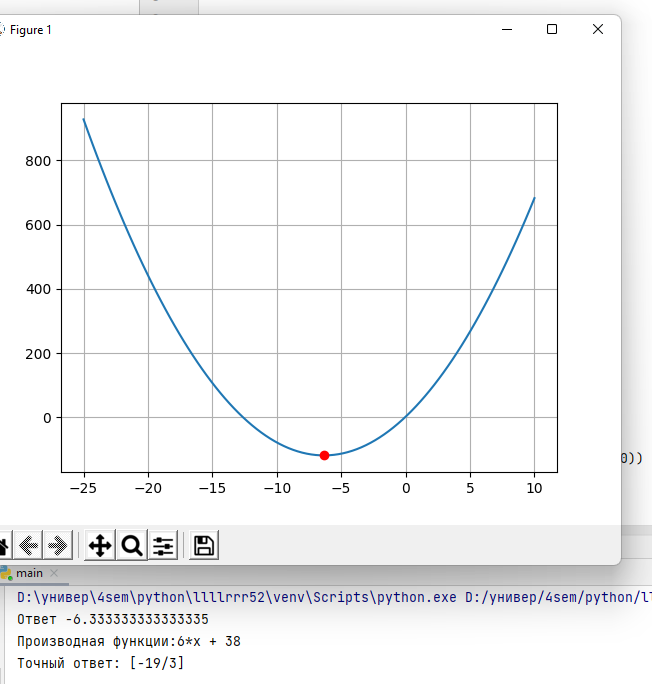
--вычислить количество итераций при достижении экстремума.

Таблица 2

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Функция f(x)= | Начальная точка | Точность |
| 7. |  | 8 | 0.2 |

Листинг программы:

**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**from** sympy **import** \*  
  
  
**def** f(x):  
 **return** 3 \* x\*\*2 + 38 \* x + 2  
  
  
**def** Powell(xn, delta):  
 x0 = xn  
 x1 = x0 + delta  
 x2 = 0  
 y0 = f(x0)  
 y1 = f(x1)  
 y2 = 0  
 **if** y0 < y1:  
 x2 = x0 - delta  
 **else**:  
 x2 = x0 + 2 \* delta  
 steps = 0  
 **while True**:  
 steps += 1  
 y0 = f(x0)  
 y1 = f(x1)  
 y2 = f(x2)  
 a0 = y0  
 a1 = (y1 - y0) / (x1 - x0)  
 a2 = (1 / (x2 - x1)) \* (((y2 - y0) / (x2 - x0)) - ((y1 - y0) / (x1 - x0)))  
 xz = ((x1 + x0) / 2) - (a1 / (2 \* a2))  
 **if** x0 >= x1 <= x2:  
 **if** (x1 - xz) < e:  
 **return** xz, steps  
 **elif** x1 >= x0 <= x2:  
 **if** (x0 - xz) < e:  
 **return** xz, steps  
 **else**:  
 **if** (x2 - xz) < e:  
 **return** xz, steps  
 **if** y1 <= y0 >= y2:  
 x0 = xz  
 **elif** y0 <= y1 >= y2:  
 x1 = xz  
 **else**:  
 x2 = xz  
  
  
**def** PowellAccurate():  
 xe = Symbol(**'x'**, real=**True**)  
 fone = 3\* xe \*\* 2 + 38 \* xe +1  
 derivative = fone.diff(xe)  
 result = solve(derivative, xe)  
 **return** derivative, result  
  
  
xn = 8  
e = 0.2  
result, stepsP = Powell(xn, 5)  
derivative, res = PowellAccurate()  
print(**"Ответ"**, result)  
print(**f"Производная функции:{**derivative**}\nТочный ответ: {**res**}"**)  
  
arrx = np.linspace(-25, 10, 100)  
arry = f(arrx)  
plt.plot(arrx, arry)  
plt.plot(result, f(result), **'ro'**, color=**"red"**)  
plt.grid()  
plt.show()



**Вывод:** получили навыки поиска экстремумов унимодальных функций с использованием численных методов, выполнили графическую интерпретацию результатов.