Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №1**

**по дисциплине**  
 **«Методы оптимизации**»

Работу выполнил студент группы 34 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.М. Урюпин

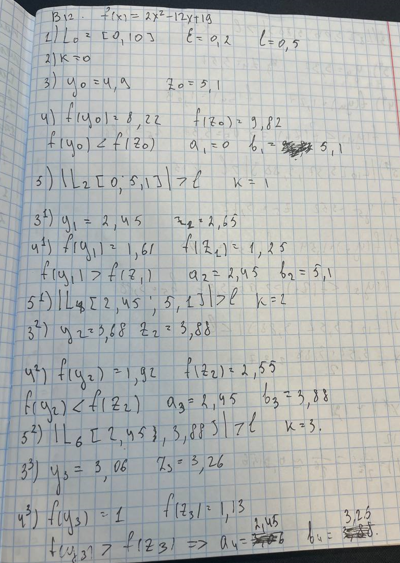
Направление подготовки: 02.03.03 Математическое обеспечение и администрирование информационных систем

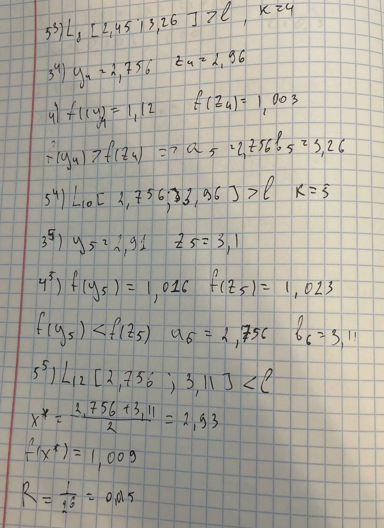
Отчет принял Преподаватель            А. С. Чёрная

**Постановка задачи**

Требуется найти точку минимума и значение в этой точке несколькими методами (методом дихотомии, методом золотого сечения, методом Фибоначчи. δ = 0.2, ε = 0.5) для функции:

**Решение 1.** Метод дихотомии.





**Код программы**

import math

import matplotlib.pyplot as plt

import numpy as np

def function(x):

    res = [(2 \* xi \*\* 2) - (12 \* xi) + 19 for xi in x]

    return res

def compute\_yz(a, b):

    y\_new = (a + b - delta) / 2

    z\_new = (a + b + delta) / 2

    return y\_new, z\_new

def update\_interval(y, z):

    if function([y])[0] <= function([z])[0]:

        l[1] = z

    else:

        l[0] = y

    return 2 \* (k + 1)

def calc\_accuracy(a, b):

    return abs(b - a) < ep

l = [0, 10]

delta = 0.2

ep = 0.5

k = 0

n = 0

print(f"""

Дана задача

Интервал неопределённости: {l}

Дельта: {delta}

Точность epsilon: {ep}

Сама функция: 2x^2 - 12x + 19

Начальное k: {k}

""")

while not calc\_accuracy(l[0], l[1]):

    y, z = compute\_yz(l[0], l[1])

    update\_interval(y, z)

    n = 2 \* (k + 1)

    k += 1

min\_point = (l[1] + l[0]) / 2

func\_val = function([min\_point])[0]

convergence = 1 / (2 \*\* (n / 2))

print(f"""

Решение

Исходная функция: 2x^2 - 12x + 19

Конечное k: {k}

Индекс конечного интервала неопределённости: {n}

Конечный интервал неопределённости: {l}

Точка минимума: {min\_point}

Значение функции в точке минимума: {func\_val}

Сходимость: {convergence}

""")

x\_range = np.linspace(-10, 10, 1000)

y\_values = function(x\_range)

plt.figure(figsize=(10, 6))

plt.plot(x\_range, y\_values, label='f(x)', color='purple', linewidth=2)

plt.scatter(min\_point, func\_val, color='red', s=100, label='Минимум', edgecolors='black', zorder=3)

plt.xlabel('X\*', fontsize=12)

plt.ylabel('f (x\*)', fontsize=12)

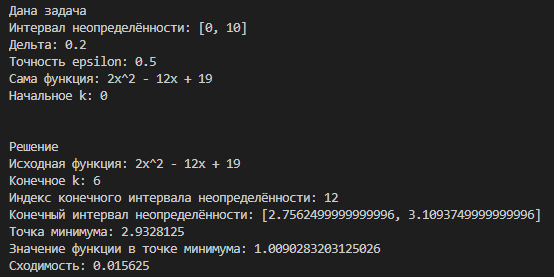
plt.title('График функции', fontsize=14, fontweight='bold')

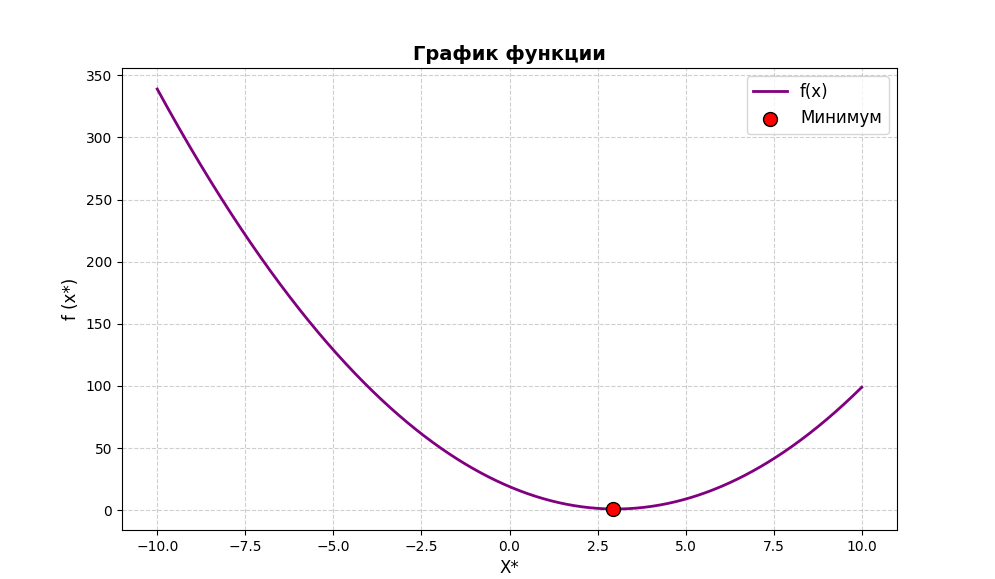
plt.legend(fontsize=12)

plt.grid(True, linestyle='--', alpha=0.6)

plt.show()

Вывод решения методом дихотомии:





**Решение 2.** Метод Фибоначчи.

**Код программы**

import math

import matplotlib.pyplot as plt

import numpy as np

f = lambda x: 2\*x\*\*2 - 12\*x + 19

e = 0.2

l = 0.5

a, b = 0, 10

F = [1, 1]

while F[-1] <= (b - a) / l:

    F.append(F[-1] + F[-2])

print("Элементы массива F:", F)

n = len(F) - 1

print(f"N =  {n}")

y = a + (b - a) \* F[n - 2] / F[n]

z = a + (b - a) \* F[n - 1] / F[n]

f1 = f(y)

f2 = f(z)

print(f"y{0} = {y:.5f}, z{0} = {z:.5f}")

print(f" [{a:.5f}, {b:.5f}]")

# print(F[n - 2])

# print(F[n - 1])

changes = {'a': [(a, 0)], 'b': [(b, 0)]}

k = 0

print(f"""

Дана задача

Интервал неопределённости: [{a}, {b}]

Дельта: {e}

Точность epsilon: {l}

Сама функция: 2x^2 - 12x + 19

Начальное k: {k}

""")

while k <= (n - 3):

    # print(f"Шаг {k}: [{a:.5f}, {b:.5f}]")

    # print(f"y{k} = {y:.5f}, z{k} = {z:.5f}")

    # print(f"f(y{k}) = {f1:.5f}, f(z{k}) = {f2:.5f}")

    print()

    if f1 > f2:

        a = y

        y = z

        f1 = f2

        z = a + F[n - k - 2] / F[n - k - 1] \* (b - a)

        f2 = f(z)

        changes['a'].append((a, k))

    else:

        b = z

        z = y

        f2 = f1

        y = a + F[n - k - 3] / F[n - k - 1] \* (b - a)

        f1 = f(y)

        changes['b'].append((b, k))

    k += 1

# print(f"y{k} = {y:.5f}, z{k} = {z:.5f}")

# print(f"f(y{k}) = {f1:.5f}, f(z{k}) = {f2:.5f}")

# print(f"[{a:.5f}, {b:.5f}]")

changes['b'].append((z, k))

y = (a + b) / 2

# print(y)

z = y + e

# print(z)

f1 = f(y)

f2 = f(z)

# print(f"fy6 = {f1:.5f}, fz6 = {f2:.5f}")

if (f1 <= f2):

    b = f2

else:

    a = f1

min\_point = (a + b) / 2

func\_val = f(min\_point)

convergence = 1 / F[n]

print(f"""

Решение

Исходная функция: 2x^2 - 12x + 19

Конечное k: {k}

Индекс конечного интервала неопределённости: {k}

Конечный интервал неопределённости: [{a}, {b}]

Точка минимума: {min\_point}

Значение функции в точке минимума: {func\_val}

Сходимость: {convergence}

""")

x\_range = np.linspace(-2, 12, 1000)

y\_values = f(x\_range)

plt.figure(figsize=(10, 6))

plt.plot(x\_range, y\_values, label='f(x)', color='purple', linewidth=2)

plt.scatter(min\_point, func\_val, color='red', s=100, label='Минимум', edgecolors='black', zorder=3)

plt.xlabel('X\*', fontsize=12)

plt.ylabel('f (x\*)', fontsize=12)

plt.title('График функции', fontsize=14, fontweight='bold')

plt.legend(fontsize=12)

plt.grid(True, linestyle='--', alpha=0.6)

plt.show()

Вывод решения методом Фибоначчи:

**Решение 2.** Метод золотого сечения.

**Код программы**

import math as mt

import matplotlib.pyplot as plt

import numpy as np

def function(*x*):

    """Функция, которая считает значение основной функции"""

    res = []

    for i in range(len(*x*)):

        res.append((2 \* mt.pow(*x*[i], 2)) - (12 \* 2 \* *x*[i]) + 19)

    return res

def y\_z():

    """Функция, которая находит новые значения y и z"""

    y = l[0] + num\_1 \* (l[1] - l[0])

    z = l[0] + l[1] - y

    return y, z

def fy\_fz(*y*, *z*, *count\_n\_for*, *n*):

    """Функция, которая считает значение функции в точках y и z"""

    fy = function([*y*])

    fz = function([*z*])

    if fy <= fz:

        l[1] = *z*

        y\_new = l[0] + l[1] - *y*

        z\_new = *y*

    else:

        l[0] = *y*

        y\_new = *z*

        z\_new = l[0] + l[1] - *z*

    if *count\_n\_for* == 0:

*n* = 2 \* (k + 1)

*count\_n\_for* += 1

    else:

*n* += 1

*count\_n\_for* += 1

    return y\_new, z\_new, *count\_n\_for*, *n*

def accuracy():

    """Функция, которая проверяет точность"""

    f = False

    if mt.fabs(l[1] - l[0]) < epsilon:

        f = True

    return f

num\_1 = (3 - mt.sqrt(5)) / 2

num\_2 = 1 - num\_1

l = [0, 10]

epsilon = 0.5

k = 0

count\_n\_for\_l = 0

n = 0

print()

print(f"""

Входные данные:

Интервал неопределённости l = {l}

Точность epsilon = {epsilon}

Сама функция: 2x^2 - 2x + 5/2

Начальное k: {k}

""")

y, z = y\_z()

while not accuracy():

    y, z, count\_n\_for\_l, n = fy\_fz(y, z, count\_n\_for\_l, n)

    k += 1

print()

print(f"""

Выходные данные:

Исходная функция: 2x^2 - 2x + 5/2

Точка min = {(l[1] + l[0]) / 2}

Значение функции в точке min = {function([(l[1] + l[0]) / 2])}

Интервал, в котором находится точка min: {l}

Индекс конечного интервала n = {n}

Сходимость = {mt.pow(num\_2, n - 1)}

Конечное k = {k}

Точность (по отношению к epsilon) = {mt.fabs(l[1] - l[0])}

""")

x\_range = np.linspace(-1, 9 ,1000)

plt.plot(x\_range, function(x\_range), *label*='f(x)')

plt.scatter((l[1] + l[0]) / 2, function([(l[1] + l[0]) / 2]), *color*='blue', *s*=20, *label*='Минимум')

plt.xlabel('X')

plt.ylabel('Y')

plt.legend()

plt.show()

Вывод решения методом золотого сечения:

**Вывод.** В лабораторной работе были рассмотрены три метода оптимизации: метод дихотомии, метод золотого сечения и метод Фибоначчи.