МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

Курсовая работа
по дисциплине «Алгоритмы решения задач оптимизации»
Тема: Решение задач одномерной безусловной оптимизации без использования и с использованием производных целевой функции

Студент гр. 0303	____________________	Болкунов В.О.
Студент гр. 0303	____________________	Калмак Д.А.
Преподаватель	____________________	Середа А.-В.И.

Санкт-Петербург
2024

Постановка задачи.

Исследовать методы решения задачи оптимизации аналитически заданных функций, выбрать методы решения и разработать вычислительные алгоритмы; разработать и протестировать программный модуль и провести вычислительные эксперименты.

Методы решения поставленной задачи.

Методы, использующие производные минимизируемой функции.

- **Метод средней точки (поиск Больцано)**

Пусть дана унимодальная, непрерывно дифференцируемая на интервале $\Delta_0 = (a, b)$ функция $f(x)$. Чтобы получить интервал меньшей длины для минимума функции, можно проанализировать значение производной в точке $x_0 = \frac{a+b}{2}$.

Возможно три варианта:

$f'(x_0) > 0$ новый интервал неопределенности $\Delta\ = (a, x_0)$

$f'(x_0) < 0$ новый интервал неопределенности $\Delta\ = (x_0, b)$

$f'(x_0) = 0$ x_0- точка минимума функции $f(x)$

Алгоритм поиска минимума унимодальной, непрерывно дифференцируемой функции на интервале $\Delta_0 = (a, b)$:

1. Интервал (a, b). Задать значения $\varepsilon > 0$ и $\delta > 0$.
2. Если $(b - a) \leq 2\varepsilon,$ то к 5.
3. Вычислить $x_0 = \frac{a+b}{2}$ и определить значение $f'(x_0)$.
4. $f'(x_0) > \delta$, тогда $b = x_0$и к 2.

   $f'(x_0) <- \delta$, тогда $a = x_0$и к 2.

   $|f'(x_0)| \leq \delta$, тогда x_0- искомая точка минимума функции $f(x)$ и к 5.

5. Процедура завершена. $x_0 = \frac{a+b}{2}$ - решение поставленной задачи. δ - заданная точность определения значений $f'(x)$ от нуля. (δ может быть, в частности, равно ε)

- **Метод хорд**

Пусть дана унимодальная, непрерывно дифференцируемая на интервале $\Delta_0 = (a, b)$ функция $f(x)$. Предполагаем, что производная $f'(x)$ в точках x=a и x=b существует. При этом справедливо, что $f'(a)f'(b) < 0$, обязательно $f'(a)$ <0 и $f'(b)$>0. Проведем прямую через точки $A(a, f'(a))$ и $B(b, f'(b))$, она обязательно пересечет внутри интервала $\Delta_0 = (a, b)$ ось х в точке x_0. Уравнение этой прямой: $y = f'(a) + \frac{f'(b)-f'(a)}{b-a}(x - a)$, тогда $x_0 = a - f'(a) + \frac{b-a}{f'(b)-f'(a)}$

$f'(x_0) > 0$ новый интервал неопределенности $\Delta = (a, x_0)$

$f'(x_0) < 0$ новый интервал неопределенности $\Delta = (x_0, b)$

$f'(x_0) = 0$ x_0- точка минимума функции $f(x)$

Алгоритм:

1. Интервал (a, b). Задать значения $\varepsilon > 0$ и $\delta > 0$.
2. Вычислить $f'(a)$ и $f'(b)$. Предполагаем, что $f'(a)$<0 и $f'(b)$>0. Если это не так, то процесс завершается и необходимо скорректировать интервал (a, b).
3. Определить значение $x_0 = a - f'(a)\frac{b-a}{f'(b)-f'(a)}$
4. Вычислить значение $f'(x_0)$
5. $f'(x_0) > \delta$, тогда $b = x_0$ и к 6.

   $f'(x_0) < -\delta$, тогда $a = x_0$ и к 6.

   $|f'(x_0)| \leq \delta$, тогда x_0- искомая точка минимума функции $f(x)$ и к 7.
6. Если $(b - a) > 2\varepsilon$, то к 3.

Если $(b - a) \le 2\varepsilon$, то положить $x_0 = \frac{a+b}{2}$ и к 7.

7. Процедура завершена. x_0 - решение поставленной задачи.

- **Метод Ньютона-Рафсона (метод касательных)**

Пусть дана унимодальная на интервале $\Delta_0 = (a, b)$ функция $f(x)$. Для метода касательных функция $f(x)$ должна быть дважды дифференцируема. В качестве основного элемента алгоритма используется аппроксимация $f'(x)$ линейной функцией, а именно касательной к графику $f'(x)$ в текущей точке. Очередным приближением к решению уравнения $f'(x) = 0$ считается точка пересечения этой касательной с осью OX.

Уравнение касательной к графику функции $f'(x)$ в точке $M_0(x_0, f'(x_0))$: $y - f'(x_0) = f''(x_0)(x - x_0)$. Касательная пересекает ось OX в точке $x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)}$. Уравнение касательной к графику функции $f'(x)$ в точке $M_1(x_1, f'(x_1))$: $y - f'(x_1) = f''(x_1)(x - x_1)$. Касательная пересекает ось OX в точке $x_2 = x_1 - \frac{f'(x_1)}{f''(x_1)}$. Получаем точки до тех пор, пока не приблизимся к x* - решению уравнения $f'(x) = 0$ с достаточной степенью точности. $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$, k=0, 1, 2… при этом необходимо $f''(x_k) \ne 0 \; \forall k$

Алгоритм:

1. Начать с точки x_0. Положить k = 0. Задать $\delta > 0$.
2. Вычислить $f'(x_k)$ и $f''(x_k)$.
3. Если $|f'(x_k)| \le \delta$, то x_k принимаем за искомую точку экстремума функции $f(x)$. Переходим к 6.
4. Вычислить значение $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$
5. Положим k = k + 1 и переходим к 2.
6. Процедура завершена. x_k - решение поставленной задачи.

Критерий окончания может быть другой. Например, близость к нулю разности двух последних членов последовательности $\{x_k\}$ $|x_k - x_{k-1}| \le \varepsilon$

Методы, не использующие производные минимизируемой функции.

Методы начинают свою работу с некоторого интервала $\Delta_0 = (x_0^{(0)}, x_0^{(a)})$, начального интервала неопределенности, содержащего точку минимума $f(x)$, эта функция предполагается унимодальной на этом интервале. Внутри интервала Δ_0, выберем каким-то образом две точки $x_0^{(1)}, x_0^{(2)}$. Возможны следующие три ситуации, представленные на рис. 1:

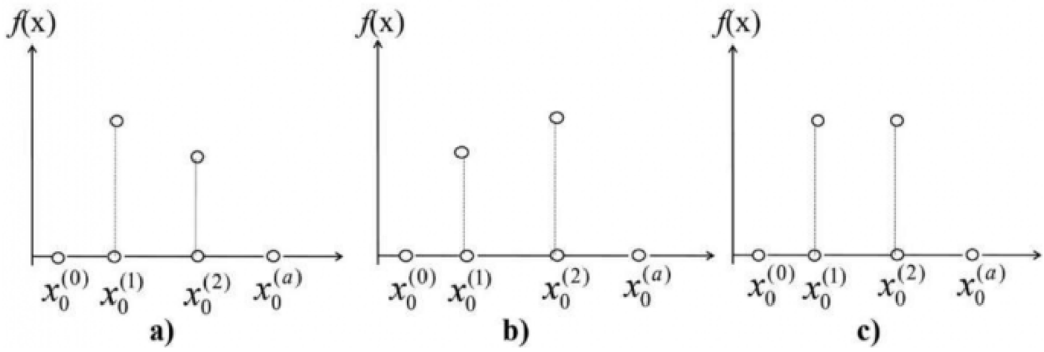

Рисунок 1 - Три ситуации выбранных точек $x_0^{(1)}, x_0^{(2)}$

a) точка минимума может быть только на интервале $(x_0^{(1)}, x_0^{(a)})$

b) точка минимума может быть только на интервале $(x_0^{(0)}, x_0^{(2)})$

c) точка минимума может быть только на интервале $(x_0^{(1)}, x_0^{(2)})$

И во всех трех случаях начальный интервал Δ_0 может быть уменьшен на конечную величину с получением нового интервала Δ_1, содержащегося в Δ_0.

- **Метод половинного деления**

Различные методы отличаются выбором положения пробных значений x внутри текущего интервала неопределенности. Целесообразно располагать две пробные точки симметрично относительно центра текущего интервала неопределенности.

Алгоритм:

1. Начать с интервала $\Delta = (a, b)$. Выбрать значение $\delta > 0$ - достаточно малое число и значение ε > 0 - допустимую погрешность в определении значения x* - оптимального решения задачи.
2. Вычислить |Δ|=b-a. Если $|\Delta| \leq 2\varepsilon$, то к 7.
3. Вычислить значение $c = \frac{(a+b)}{2}$. Определить значения $x_1 = c - \frac{\delta}{2}$ и $x_2 = c + \frac{\delta}{2}$. Вычислить значения $f(x_1)$ и $f(x_2)$.
4. Если $f(x_1)$<$f(x_2)$, то $b = x_2$
5. Если $f(x_1)$>$f(x_2)$, то $a = x_1$
6. Переходим к 2.
7. Процедура завершена. x* $= \frac{a+b}{2}$ - решение задачи с требуемой точностью.

- **Метод золотого сечения**

Отличается правило выбора пробных точек внутри текущего интервала неопределенности, при соблюдении которого на каждом шаге процесса (кроме первого) требуется строить только одну новую пробную точку. Исходный интервал неопределенности $\Delta_0 = (a, b)$. На каждом шаге строится интервал меньшей длины, содержащийся в предыдущем интервале. Для этого необходимо проанализировать значения минимизируемой функции в двух пробных точках, которые принадлежат текущему интервалу неопределенности Δ_k и расположенных симметрично относительно его середины. Отличается метод и двумя дополнительными условиями для последовательности вложенных интервалов, строящихся в процессе работы метода:

Для длин любых трех последовательно строящихся интервалов выполнено соотношение $|\Delta_{k-2}| = |\Delta_{k-1}| + |\Delta_k|$, $k = 2, 3,...$

Отношение длины последующего интервала к длине предыдущего интервала постоянно. Обозначим отношение через λ. $\frac{\lambda}{1} = \frac{1-\lambda}{\lambda}$ => $\lambda^2 + \lambda - 1 = 0$ => $\lambda = \frac{-1\pm\sqrt{5}}{2}$. Положительное $\lambda \simeq 0,618$.

Правило золотого сечения: на k-м шаге располагаем две пробные точки на расстоянии $0,382|\Delta_k|$ (1 - 0,618 = 0,382) от концов текущего интервала неопределенности Δ_k. За k шагов процесса, $k \geq 1$, потребуется k+2 вычислений функции $f(x)$. Длина интервала неопределенности на k-м шаге метода определяется как $|\Delta_k| = \lambda^k|\Delta_0|$, $k = 0, 1, 2,...$

Алгоритм:

1. Начать с интервала $\Delta = (a, b)$. Выбрать значение $\varepsilon > 0$. Положить $\lambda = 0.618$.
2. Вычислить $|\Delta| = b - a$. Если $|\Delta| \leq 2\varepsilon$, то к 6.
3. Вычислить $\delta = \lambda|\Delta|$. Определить значения $x_1 = b - \delta$ и $x_2 = a + \delta$. Вычислить значения $f(x_1)$ и $f(x_2)$.
4. Если $f(x_1)<f(x_2)$, $b = x_2$. Вычислить $|\Delta| = b - a$. Если $|\Delta| \leq 2\varepsilon$, то к 6. Иначе: положить $x_2=x_1$, $f(x_2)=f(x_1)$, определить $x_1 = b - \lambda|\Delta|$, вычислить значение $f(x_1)$, перейти к 4.
5. Если $f(x_1)>f(x_2)$, $a = x_1$. Вычислить $|\Delta| = b - a$. Если $|\Delta| \leq 2\varepsilon$, то к 6. Иначе: положить $x_1=x_2$, $f(x_1)=f(x_2)$, определить $x_2 = a + \lambda|\Delta|$, вычислить значение $f(x_2)$, перейти к 4.
6. Процедура завершена. $x^* = \frac{a+b}{2}$ - решение задачи с требуемой точностью.

Выполнение работы.

Выбор метода решения и разработка вычислительного алгоритма.

Выбор метода влияет не только на скорость сходимости к решению задачи минимизации функции, но и на объем вычислительной работы, которая необходима для решения задачи. Причем при минимизации функции объем вычислительной работы является важным пунктом при неизвестном аналитическом задании.

Если функция сложная и быстро изменяется, то лучше использовать методы, ориентированные на механическое деление на каждом шаге текущего интервала неопределенности по изначально заданному правилу, а не анализирующие специфику поведения функции. Среди таких методов есть метод золотого сечения и метод половинного деления. Метод золотого сечения может превосходить метод Ньютона-Рафсона, или метод касательных, по критерию объема вычислительной работы. Вместе с этим алгоритм метода золотого сечения в отличие от метода половинного деления требует лишь одну пробную точку при работе алгоритма, а также накладывает дополнительные условия. Метод золотого сечения зарекомендовал себя как надежный и простой для любых классов функций.

Если функция достаточно гладкая унимодальная, то стоит использовать методы, основанные на полиномиальной аппроксимации исследуемой функции. Один из таких методов - метод Ньютона-Рафсона, или метод касательных. Он обладает квадратичной скоростью сходимости, которая является наивысшей скоростью сходимости среди известных в настоящее время методов, но как упоминалось ранее, он может быть трудоемким. Метод Ньютона-Рафсона, или метод касательных, наиболее эффективен для решения уравнений, у которых график в окрестности корня имеет большую крутизну. Метод хорд является еще одним вариантом метода, основанного на полиномиальной аппроксимации исследуемой функции, и ориентирован на нахождение корня уравнения в случае, когда на границах интервала знаки производной различны. В случае квадратичной функции метод хорд гарантирует нахождение стационарной точки функции всего за одну итерацию.

Все рассмотренные методы могут находить лишь один из локальных минимумов, что является проблемой в случае многоэкстремальной функции, поскольку локальный минимум не обязательно будет глобальным минимумом. Метод Ньютона-Рафсона, или метод касательных, дает наивысшую скорость сходимости при хороших условиях, однако чтобы повысить стабильном метода используют комбинацию метода касательных и метода хорд, поскольку они дают приближения корня с противоположных сторон, один метод дает значение корня с недостатком, а другой - с избытком, поэтому их применяют в сочетании друг с другом, чтобы уточнение корня происходило быстрее. Наиболее перспективным вариантом является совмещение двух типов методов: без производной и с производной. В качестве комбинации выбран метод золотого сечения, как хорошо зарекомендовавший себя, а так же комбинация метода Ньютона-Рафсона, или метода касательных, и метода хорд, которые в комбинации дополняют друг друга.

Исходя из представленных методов, лучше всего использовать метод золотого сечения, поскольку он может выигрывать объем вычислительной работы, а так же показать более надежную практическую ценность на любых классах функций. Метод Ньютона-Рафсона, или метод касательных, показывает лучшую скорость сходимости, однако это может сопровождаться большим объемом вычислительной работы и рисками найти локальный, но не глобальный минимум. Для уменьшения риска используется комбинация двух методов: метода Ньютона-Рафсона, или метода касательных, и метода хорд. Для минимизации риска стоит использовать комбинацию двух типов методов безусловной одномерной оптимизации, а именно без производной и с производной. Поэтому как наиболее практичная используется комбинация метода золотого сечения с методами касательных и хорд.

Разработка программного модуля и его тестирование.

Для выбранных методов золотого сечения, хорд, и Ньютона-Рафсона, а также для комбинированного метода хорд и Ньютона-Рафсона и параллельного метода золотого сечения и комбинированного Н.-Р. разработаны вычислительные алгоритмы на языке Python 3, представленные в листинге 1. Программный модуль реализован с использованием библиотеки PyQt6, что позволяет пользователю работать с понятным графическим интерфейсом и упрощает взаимодействие.

Разработанные алгоритмы протестированы на функции $f(x) = x^4 - 2x^3 - 2x^2 + 2x - 4$, график функции представлен на рис. 1. На рис. 2-6 представлен пример работы соответствующих алгоритмов в программном модуле при поиске оптимальной точки.

Листинг 1. Вычислительные алгоритмы.

```
# Функция алгоритма оптимизации для метода золотого сечения
lmbd = ((-1 + sqrt(5)) / 2).evalf()
def goldSlice(f, epsilon, a=-1, b=1):
 X = [(a+b)/2]
 d = b - a
 while abs(d) > 2*epsilon:
   ld = lmbd * abs(d)
   x1 = b - ld
   x2 = a + ld
   fx1 = f(x1).evalf()
   fx2 = f(x2).evalf()
   if fx1 < fx2:
     b = x2
   else:
     a = x1
   d = b - a
   X.append((a+b)/2)
 return (a+b)/2, X

# Функция алгоритма оптимизации для метода хорд
def chords(f1, epsilon, delta, a=-1, b=1):
 f1a = f1(a).evalf()
 f1b = f1(b).evalf()
 if not (f1a < 0 and f1b > 0):
   raise Exception('incorrect interval')
 X = []
 while True:
   f1a = f1(a).evalf()
   f1b = f1(b).evalf()
   x = a - f1a * (b-a) / (f1b - f1a)
   X.append(x)
   f1x = f1(x).evalf()
   if f1x > delta:
     b = x
   elif f1x < -delta:
     a = x
```

```
    else:
      return x, X
    if abs(b - a) <= 2 * epsilon:
      return (a+b)/2, X


# Функция алгоритма оптимизации для метода Ньютона-Рафсона
def NewtonRafson(f1, f2, delta, x0=0):
 x = x0
 X = [x0]
 fx1 = f1(x).evalf()
 fx2 = f2(x).evalf()
 while abs(fx1) > delta:
   x = x - fx1 / fx2
   X.append(x)
   fx1 = f1(x).evalf()
   fx2 = f2(x).evalf()
 return x, X

# Комбинированный метод хорд и Ньютона-Рафсона
def combined_method(f1, f2, epsilon, delta, a, b, max_iter=1000):
   f1a = f1(a).evalf()
   f1b = f1(b).evalf()
   if not (f1a < 0 and f1b > 0):
       raise Exception('incorrect interval')

   X = []
   for _ in range(max_iter):
       f1a = f1(a).evalf()
       f1b = f1(b).evalf()

       # Выбор метода: касательных или хорд
       if f1a * f2(a).evalf() > 0:
           # Метод касательных для a
           a_new = a - f1a / f2(a).evalf()
       else:
           # Метод хорд для a
           a_new = a - f1a * (b - a) / (f1b - f1a)

       if f1b * f2(b).evalf() > 0:
           # Метод касательных для b
           b_new = b - f1b / f2(b).evalf()
       else:
           # Метод хорд для b
           b_new = b - f1b * (b - a) / (f1b - f1a)

       # Обновляем интервал
       if abs(f1(a_new).evalf()) < delta:
           a = a_new
       else:
           b = b_new
```

```
        # Добавляем приближение в список
        x = (a + b) / 2
        X.append(x)

        # Проверяем условия выхода
        if abs(f1(x).evalf()) < delta or abs(b - a) < 2 * epsilon:
            return x, X

# Параллельное выполнение методов золотого сечения и комбинированного
lmbd = ((-1 + sqrt(5)) / 2).evalf()
def combined_method_parallel(f, f1, f2, epsilon, delta, a, b):
    xmin1, steps1 = goldSlice(f, delta, a, b)
    xmin2, steps2 = combined_method(f1, f2, epsilon, delta, a, b)
    # Выбор наилучшего результата
    if f(xmin1) < f(xmin2):
        return xmin1, steps1
    else:
        return xmin2, steps2
```

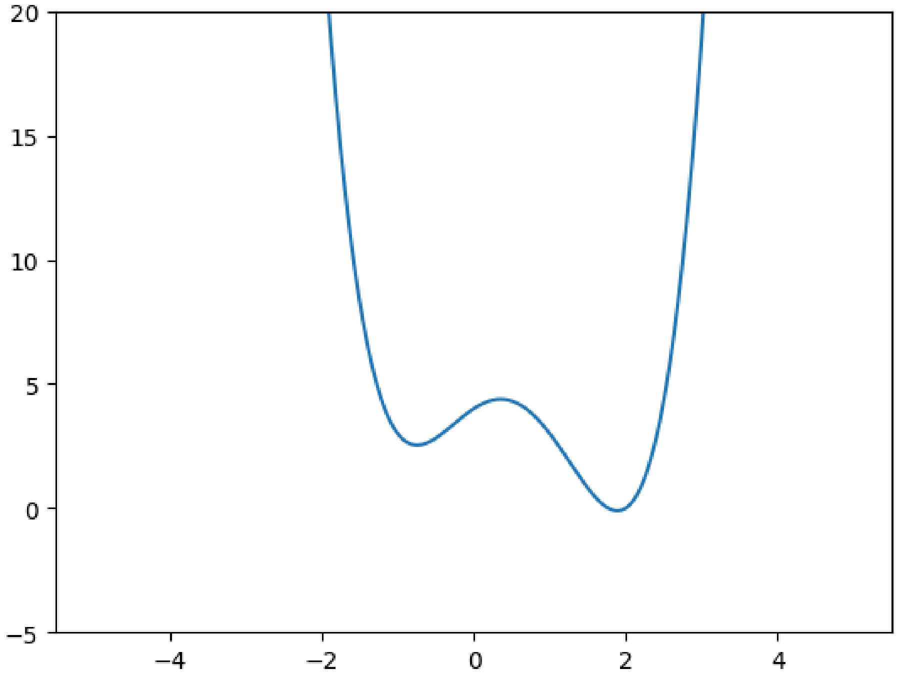

Рисунок 1. График исходной функции.

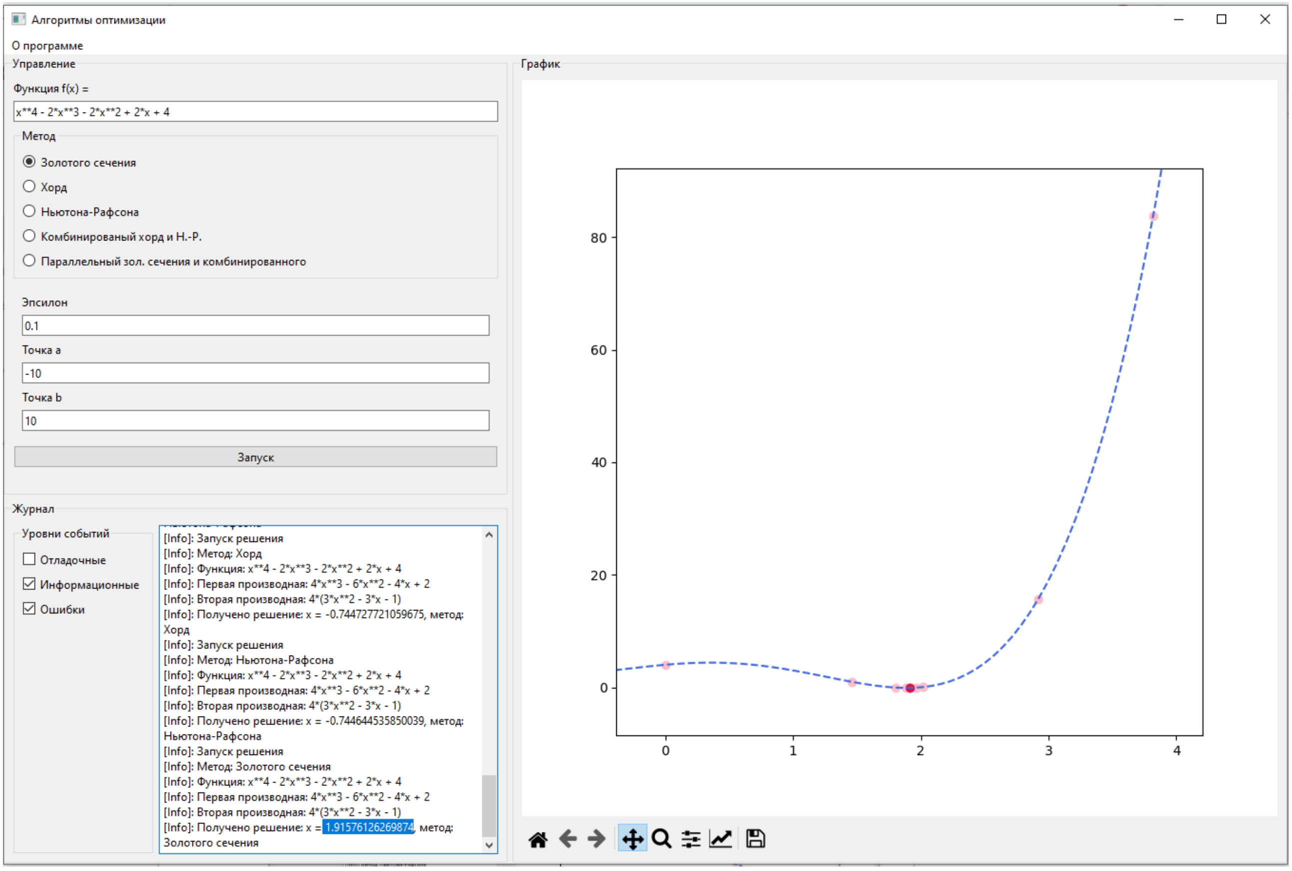

Рисунок 2. Результат работы программы для метода золотого сечения

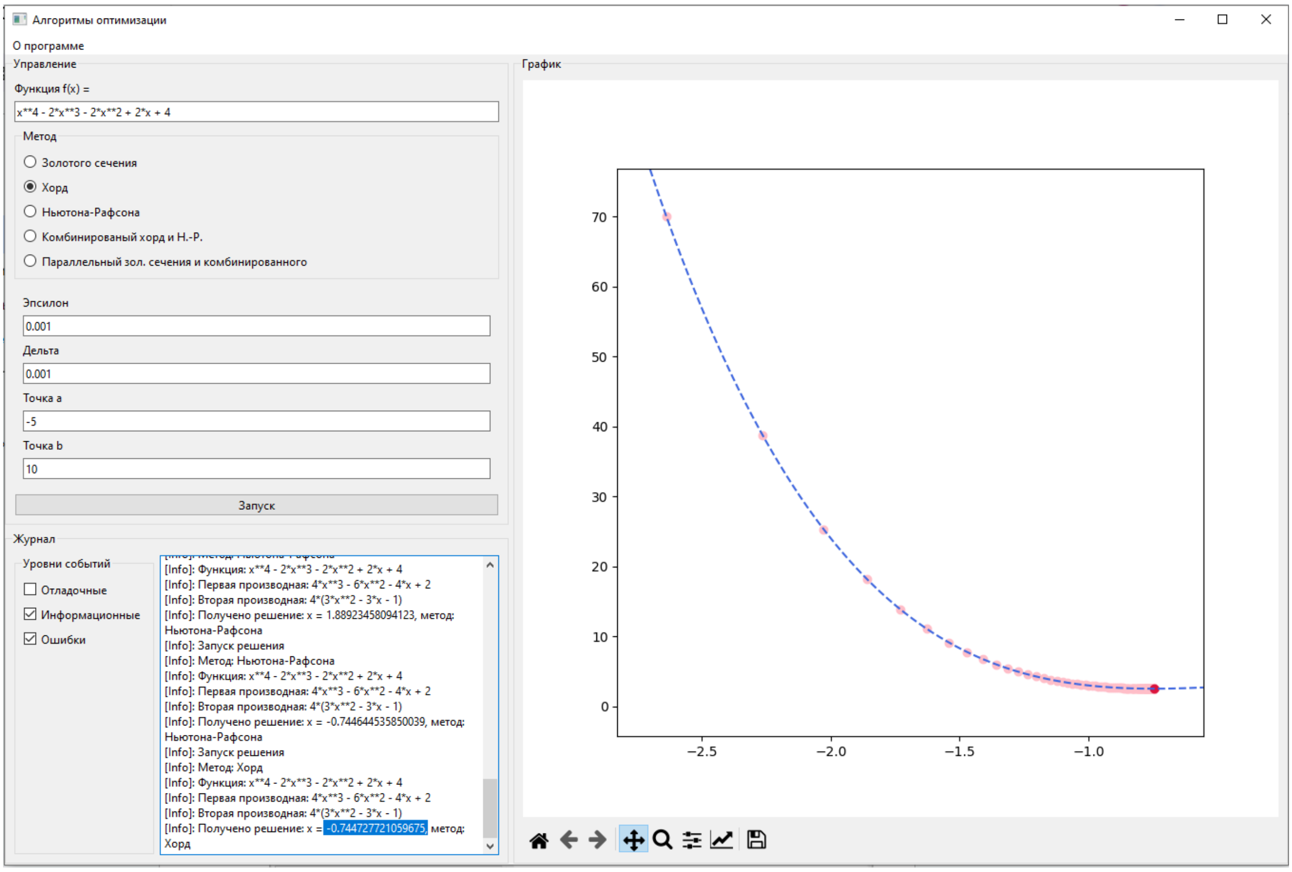

Рисунок 3. Результат работы программы для метода хорд

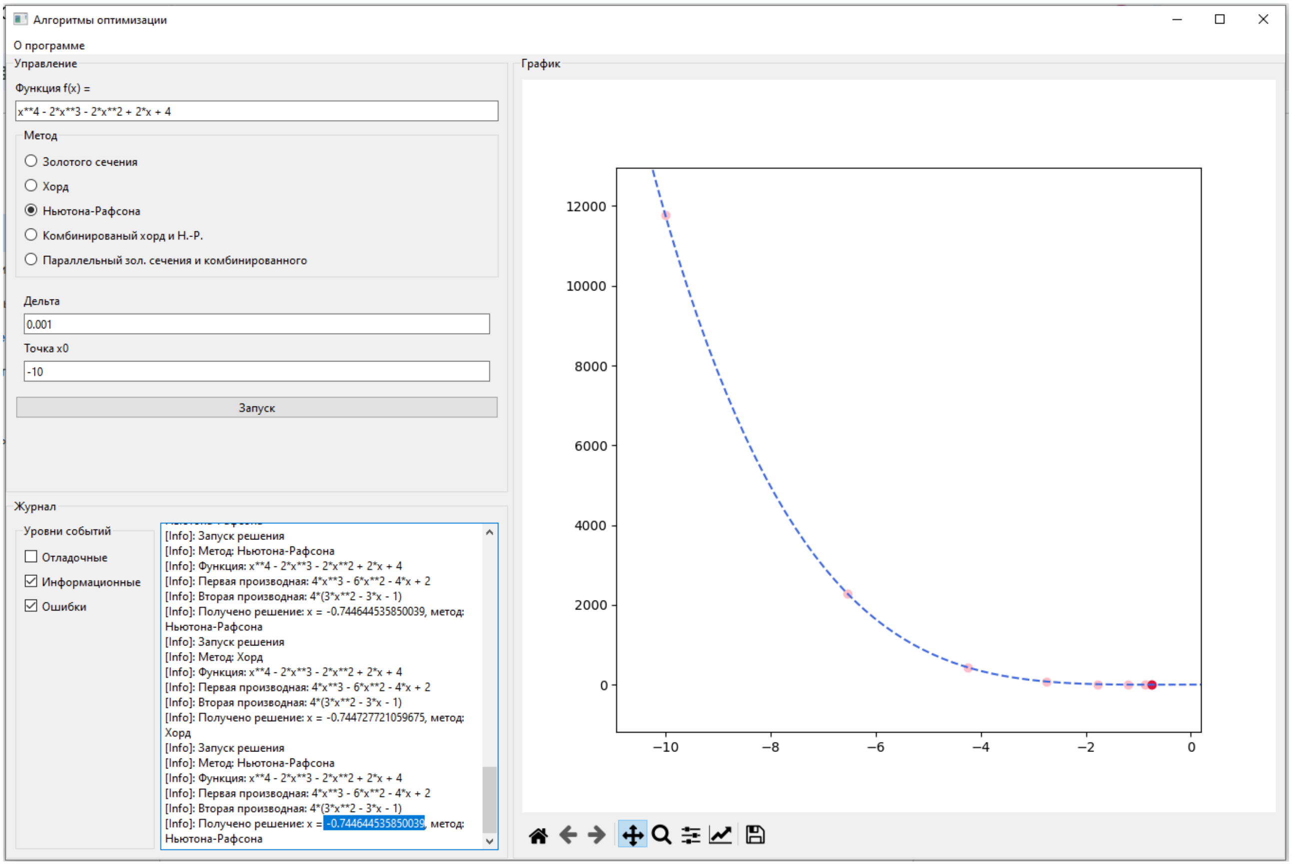

Рисунок 4. Результат работы программы для метода Ньютона-Рафсона

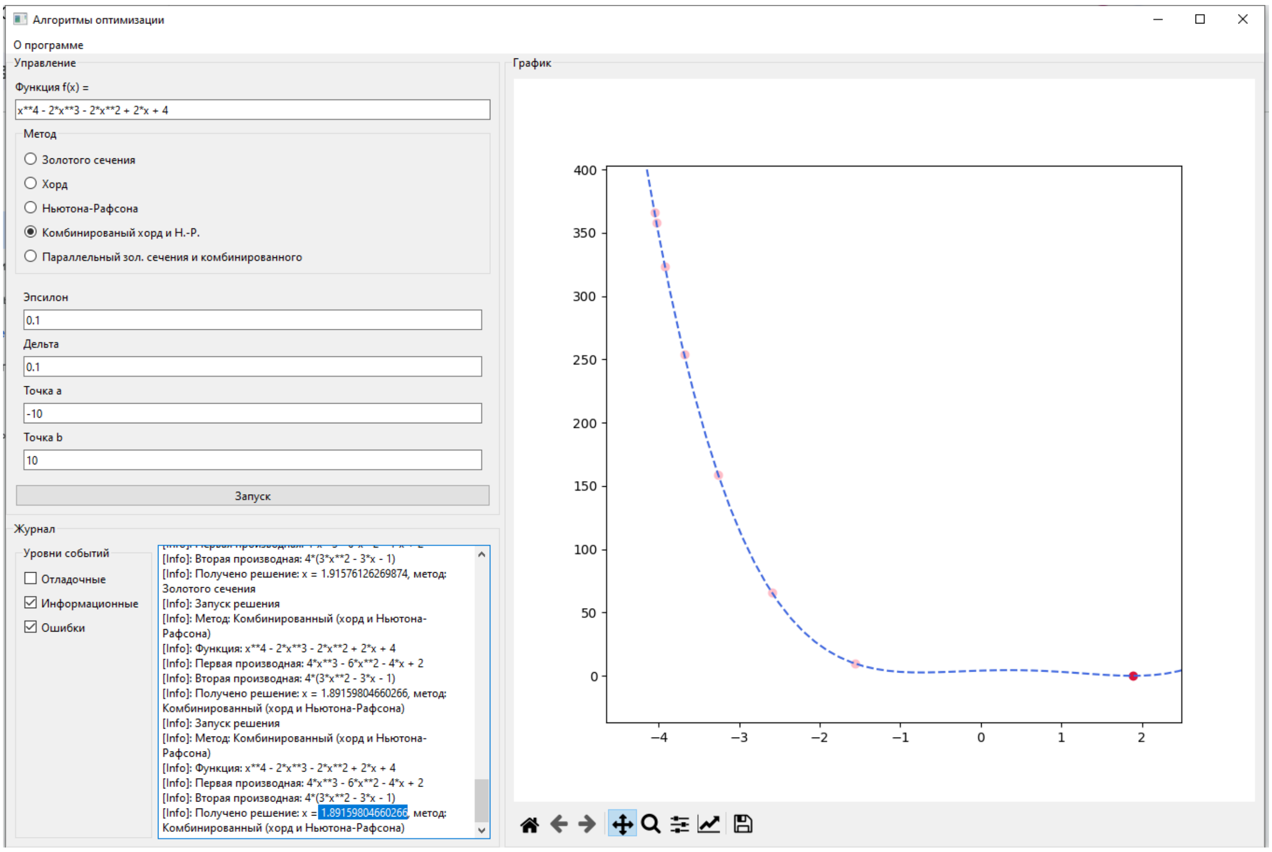

Рисунок 5. Результат работы программы для комбинированного метода хорд и Ньютона-Рафсона

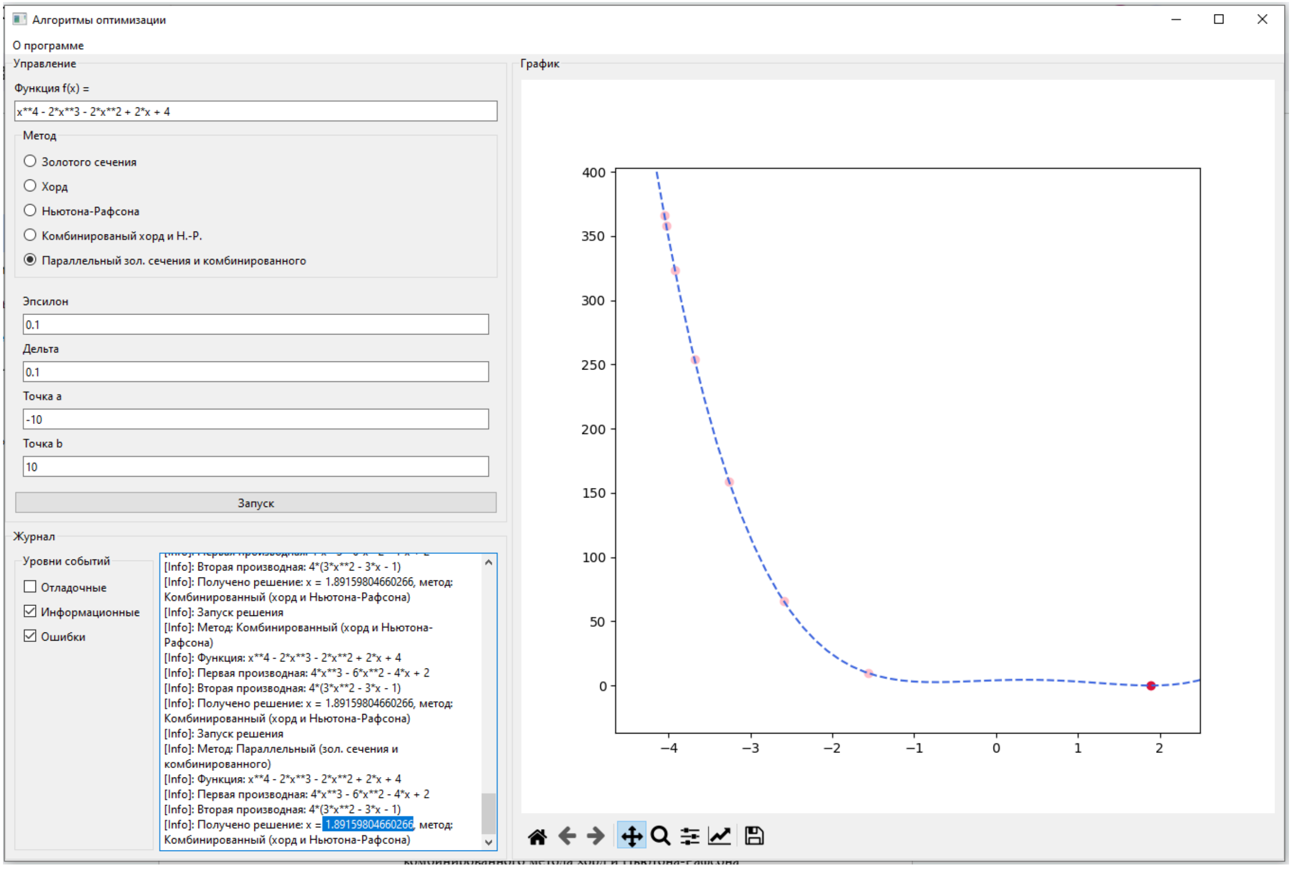

Рисунок 6. Результат работы программы для параллельного метода золотого сечения и комбинированного метода хорд и Ньютона-Рафсона

Реализованные алгоритмы находят глобальный минимум функции в зависимости от заданных начальных точек, как, например, алгоритм золотого сечения, представленный на рис. 2. В некоторых случаях они приходят к локальному минимуму, что можно наблюдать на рис. 3-4, которые соответствуют методам хорд и Ньютона-Рафсона. Данная проблема решается использованием комбинированного метода хорд и Ньютона-Рафсона, представленного на рис. 5. Для более надежного решения задач одномерной безусловной оптимизации используется параллельное выполнение методов золотого сечения и комбинированного, представленное на рис. 6.

Выводы.

Список литературы.

1. Середа А.-В.И. Лекция 1 Алгоритмы решения задач оптимизации, СПбГЭТУ “ЛЭТИ”, 2024.
2. Середа А.-В.И. Лекция 2 Алгоритмы решения задач оптимизации, СПбГЭТУ “ЛЭТИ”, 2024.
3. Эварт Т.Е., Троицкий А.В., Поздяев В.В Численные методы решения инженерных задач, 2014
4. С.И. Смуров, В.А. Таланова, С.П. Бобков Методичка 501, Глава 2. URL: https://dit.isuct.ru/IVT/sitanov/Literatura/M501.html
5. Захарова О.Н. Практическая работа 3 Теоретические сведения. URL: https://zaharova-olga.ucoz.net/ch_met/prakticheskaja_rabota-3.pdf

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД