Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Программная инженерия
кафедра

ОТЧЁТ О ПРАКТИЧЕСКОЙ РАБОТЕ №3

Численные методы второго порядка для поиска безусловного экстремума
тема

Преподаватель			В. В. Тынченко
		подпись, дата	инициалы, фамилия
Студент	КИ23-16/1б, 032322546		Е. А. Гуртякин
	номер группы, зачетной книжки	подпись, дата	инициалы, фамилия

Красноярск 2025

1 Цель

Программно реализовать численные методы безусловной оптимизации, указанные в варианте задания. Язык программирования можно использовать любой.

Протестировать работу методов на функциях из примеров, решение которых пошагово рассмотрено в учебнике Пантелеева и Летовой «Методы оптимизации в примерах и задачах», раздел 2 «Численные методы».

Провести сравнительный анализ влияния параметров методов на точность и скорость нахождения экстремума, варьируя параметры в заданных пределах с выбранным шагом. Для выполнения вычислительных экспериментов использовать целевые функции, указанные в варианте задания.

Точность работы алгоритма определяется погрешностью рассчитанного программой минимального значения функции относительно реального минимума, который требуется найти самостоятельно любым способом до проведения вычислительных экспериментов.

Скорость определяется количеством вычислений целевой функции.

Для реализованных методов необходимо построить графики зависимости точности и скорости вычислений от каждого из исследуемых параметров, каждый график надо сопроводить кратким комментарием.

Все результаты вычислительных экспериментов следует представить компактно в виде сводной таблицы с итоговыми результатами по точности и скорости для методов и функций из варианта задания. В таблице для каждого метода и каждой функции указать: значения или диапазон и шаг изменения параметров метода, а также усредненные, лучшие и худшие из полученных значения погрешности и кол-ва вычислений целевой функции. Таблицу построить программным способом и поместить в отчет скриншот.

По результатам исследования требуется сделать вывод об особенностях работы исследуемых алгоритмов и их эффективности на заданных целевых функциях.

Таблица 1 – Вариант задания

7	Метод Ньютона Метод Ньютона-Рафсона	$f(x) = (x_1 - 4)^2 + (x_2 - 1)^2 \to min$ $f(x) = 2x_2^2 - 2x_2 + x_1x_2 + 4x_1^2 \to min$

2 Ход выполнения

2.1 Описание методов

Методы безусловной оптимизации, реализуемые в работе, относятся к численным итерационным алгоритмам второго порядка, предназначенным для поиска минимума гладких функций нескольких переменных. В основе каждого метода лежит использование информации о вторых производных целевой функции, что позволяет достигать более высокой скорости сходимости по сравнению с методами первого порядка. Ниже приведены краткие теоретические сведения о методах Ньютона и Ньютона-Рафсона.

Метод Ньютона относится к классу методов второго порядка и основан на квадратичной аппроксимации целевой функции в окрестности текущей точки. Основная идея метода заключается в минимизации квадратичного приближения функции, построенного с использованием разложения в ряд Тейлора до второго порядка. Если текущий вектор обозначить как x = (x_1, x_2, …, x_n), то итерационное правило имеет вид x^(k+1) = x^(k) - [Hf(x^(k))]$^{-1}$ ∇f(x^(k)), где Hf(x) - матрица Гессе (гессиан), содержащая вторые частные производные функции. Для квадратичных функций метод Ньютона сходится за одну итерацию, поскольку квадратичное приближение точно представляет исходную функцию. Основное преимущество метода - квадратичная скорость сходимости в окрестности решения. Однако метод требует вычисления и обращения матрицы Гессе на каждой итерации, что может быть вычислительно затратно для задач высокой размерности. Кроме того, метод может расходиться, если начальное приближение выбрано далеко от решения, или если гессиан вырожден.

Метод Ньютона-Рафсона представляет собой модификацию классического метода Ньютона, в которой градиент и матрица Гессе вычисляются численными методами, что делает алгоритм применимым для функций, аналитические производные которых неизвестны или сложны для вычисления. Итерационная схема сохраняется такой же, как в методе Ньютона: $x^{(k+1)} = x^{(k)} - [\hat{H}f(x^{(k)})]^{-1} \hat{g}f(x^{(k)})$, где $\hat{g}f(x)$ - численно вычисленный градиент, а $\hat{H}f(x)$ - численно вычисленная матрица Гессе. Градиент typically вычисляется с использованием центральных разностей: $\partial f/\partial x_i \approx [f(x + he_i) - f(x - he_i)]/(2h)$, а элементы гессиана вычисляются как $\partial^2 f/\partial x_i \partial x_j \approx [f(x + he_i + he_j) - f(x + he_i - he_j) - f(x - he_i + he_j) + f(x - he_i - he_j)]/(4h^2)$. Главное преимущество метода Ньютона-Рафсона - универсальность применения, поскольку он не требует аналитического задания производных. Однако метод чувствителен к выбору шага дифференцирования h: слишком большой шаг увеличивает ошибку аппроксимации, а слишком малый может привести к численной неустойчивости из-за ошибок округления.

Сравнительные особенности методов проявляются в их практическом применении. Метод Ньютона демонстрирует максимальную эффективность на квадратичных и близких к ним функциях, где обеспечивает сверхбыструю сходимость. Его использование особенно оправдано, когда аналитические производные могут быть вычислены точно и эффективно. Метод Ньютона-Рафсона, в свою очередь, является инструментом для работы с функциями произвольной формы, когда аналитические производные недоступны. Оба метода требуют решения систем линейных уравнений на каждой итерации, что может быть вычислительно затратно для высокоразмерных задач. Кроме того, оба метода могут демонстрировать неустойчивое поведение вдали от точки минимума, особенно если гессиан не является положительно определенным.

Таким образом, методы Ньютона и Ньютона-Рафсона представляют собой мощные инструменты оптимизации второго порядка, сочетающие высокую скорость сходимости с повышенными вычислительными требованиями. Их эффективность существенно зависит от правильности выбора параметров (точности ε и шага h для Ньютона-Рафсона), а также от обусловленности решаемой задачи. Сравнительное исследование этих методов позволяет выявить оптимальные стратегии их применения для различных классов целевых функций.

2.2 Результаты вычислений

Результаты выполнения предоставлены на рисунках 1, 2, 3 и 4.

```
Вычисление эталонных минимумов...
Функция f1: x_min = [4.00000000, 1.00000000], f_min = 0.00000000
Функция f2: x_min = [-0.06451613, 0.51612903], f_min = -0.51612903

Проверка градиентов в точках минимума:
f1 grad at min: [0. 0.], norm: 0.00e+00
f2 grad at min: [0. 0.], norm: 0.00e+00
```

Рисунок 1 – вывод программы для получение эталонов минимумов

```
════════════════════════════════════════════════════════════
ДИАГНОСТИЧЕСКИЙ АНАЛИЗ ДАННЫХ
════════════════════════════════════════════════════════════

Функция f1:
  newton:
    Количество записей: 30
    Погрешность: [1.00e-20, 1.00e-20]
    Вычисления: [2, 2]
    Итерации: [1, 1]
    Примеры:
      ε/h=0.1: eval=2.0, iter=1.0, error=1.00e-20
      ε/h=0.1: eval=2.0, iter=1.0, error=1.00e-20
  newton_raphson:
    Количество записей: 180
    Погрешность: [2.10e-31, 1.16e-05]
    Вычисления: [28, 42]
    Итерации: [1, 2]
    Примеры:
      ε/h=0.1: eval=28.0, iter=1.0, error=2.41e-26
      ε/h=0.1: eval=28.0, iter=1.0, error=2.48e-27

Функция f2:
  newton:
    Количество записей: 30
    Погрешность: [1.00e-20, 1.00e-20]
    Вычисления: [2, 2]
    Итерации: [1, 1]
    Примеры:
      ε/h=0.1: eval=2.0, iter=1.0, error=1.00e-20
      ε/h=0.1: eval=2.0, iter=1.0, error=1.00e-20
  newton_raphson:
    Количество записей: 180
    Погрешность: [1.00e-20, 1.46e-05]
    Вычисления: [28, 42]
    Итерации: [1, 2]
    Примеры:
      ε/h=0.1: eval=28.0, iter=1.0, error=1.11e-16
      ε/h=0.1: eval=28.0, iter=1.0, error=1.11e-16
```

Рисунок 2 – статистика по методам оптимизации (часть 1)

```
══════════════════════════════════════════════════════════════════════════════
СВОДНАЯ ТАБЛИЦА РЕЗУЛЬТАТОВ
══════════════════════════════════════════════════════════════════════════════

Статистика для функции f1:
               error                          evaluations                 \
                 min       max mean       std         min max       mean
method
newton           0.0  0.000000  0.0  0.000000           2   2   2.000000
newton_raphson   0.0  0.000012  0.0  0.000001          28  42  29.872222

                         iterations
                     std        min max      mean       std
method
newton          0.000000          1   1  1.000000  0.000000
newton_raphson  4.770757          1   2  1.138889  0.346795

Статистика для функции f2:
               error                          evaluations                 \
                 min       max mean       std         min max       mean
method
newton           0.0  0.000000  0.0  0.000000           2   2   2.000000
newton_raphson   0.0  0.000015  0.0  0.000002          28  42  29.511111

                         iterations
                     std        min max      mean       std
method
newton          0.000000          1   1  1.000000  0.000000
newton_raphson  4.306016          1   2  1.138889  0.346795

Лучшие результаты для каждой функции:

f1:
             method  parameter  evaluations  iterations  error  f_min
0            newton    0.10000            2           1    0.0    0.0
344  newton_raphson    0.00001           42           2    0.0    0.0

f2:
            method  parameter  evaluations  iterations  error     f_min
1           newton        0.1            2           1    0.0 -0.516129
65  newton_raphson        0.1           28           1    0.0 -0.516129
```

Рисунок 3 – статистика по методам оптимизации (часть 2)

```
Анализ f1:
  Метод Ньютона: 30 записей
  Метод Ньютона-Рафсона: 180 записей
  newton: evaluations range [2, 2], error range [1.00e-20, 1.00e-20]
  newton_raphson: evaluations range [28, 42], error range [2.10e-31, 1.16e-05]

Анализ f2:
  Метод Ньютона: 30 записей
  Метод Ньютона-Рафсона: 180 записей
  newton: evaluations range [2, 2], error range [1.00e-20, 1.00e-20]
  newton_raphson: evaluations range [28, 42], error range [1.00e-20, 1.46e-05]
```

Рисунок 4 – статистика по методам оптимизации (часть 3)

Также программа создает графики зависимости точности и скорости вычислений от каждого из исследуемых параметров. Они показаны на рисунках 3 и 4 и 5.

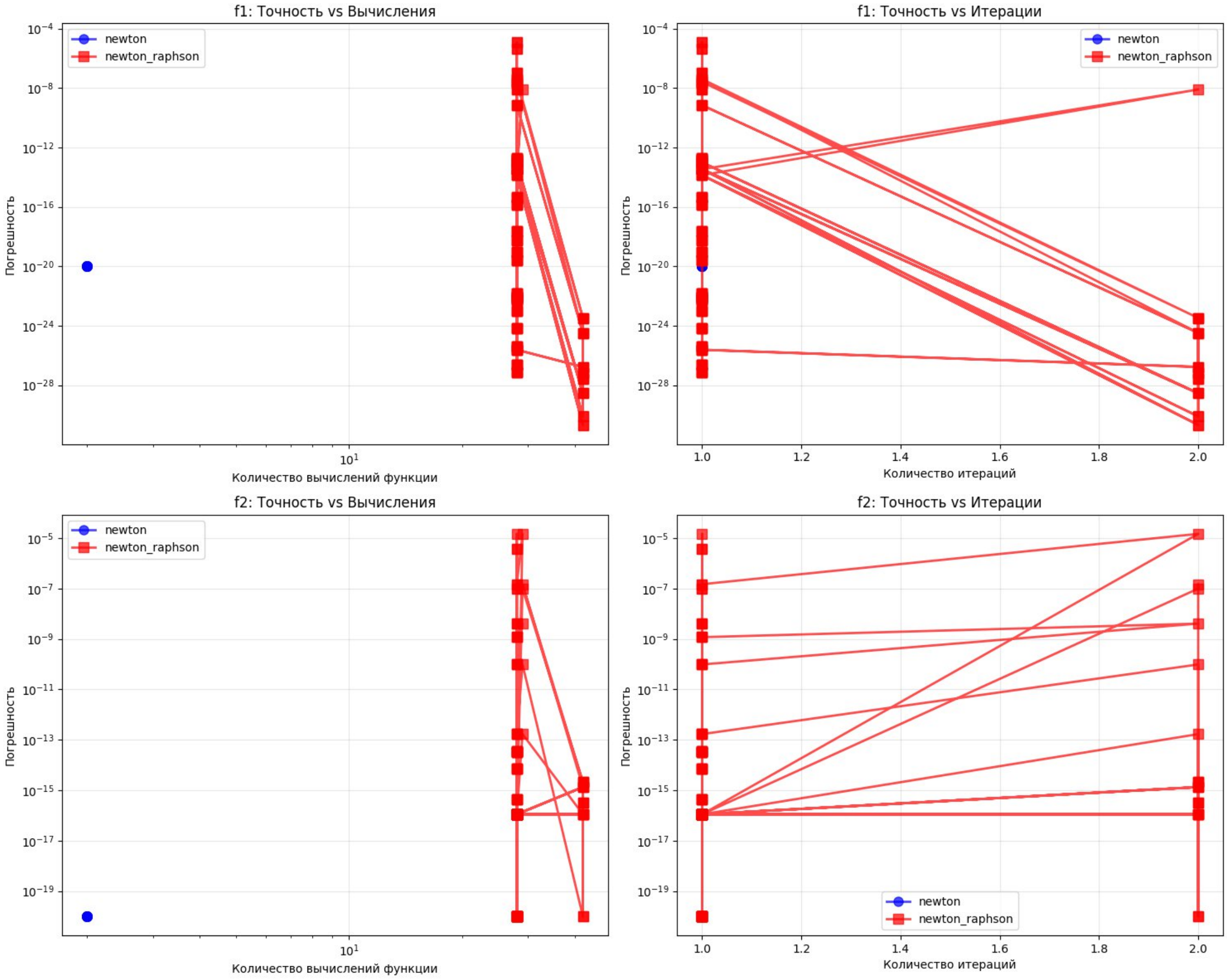

Рисунок 5 – графики сравнения методов оптимизации

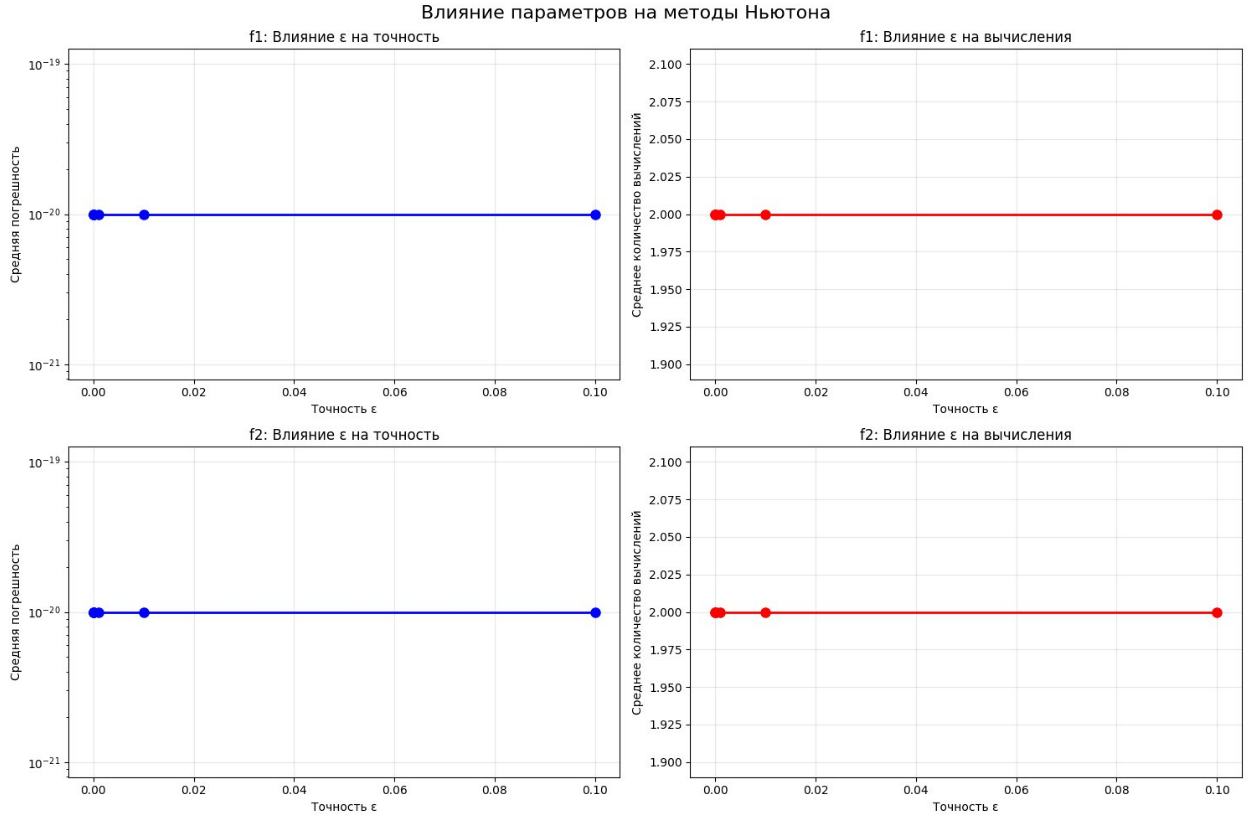

Рисунок 6 – влияние параметров на метод Ньютона

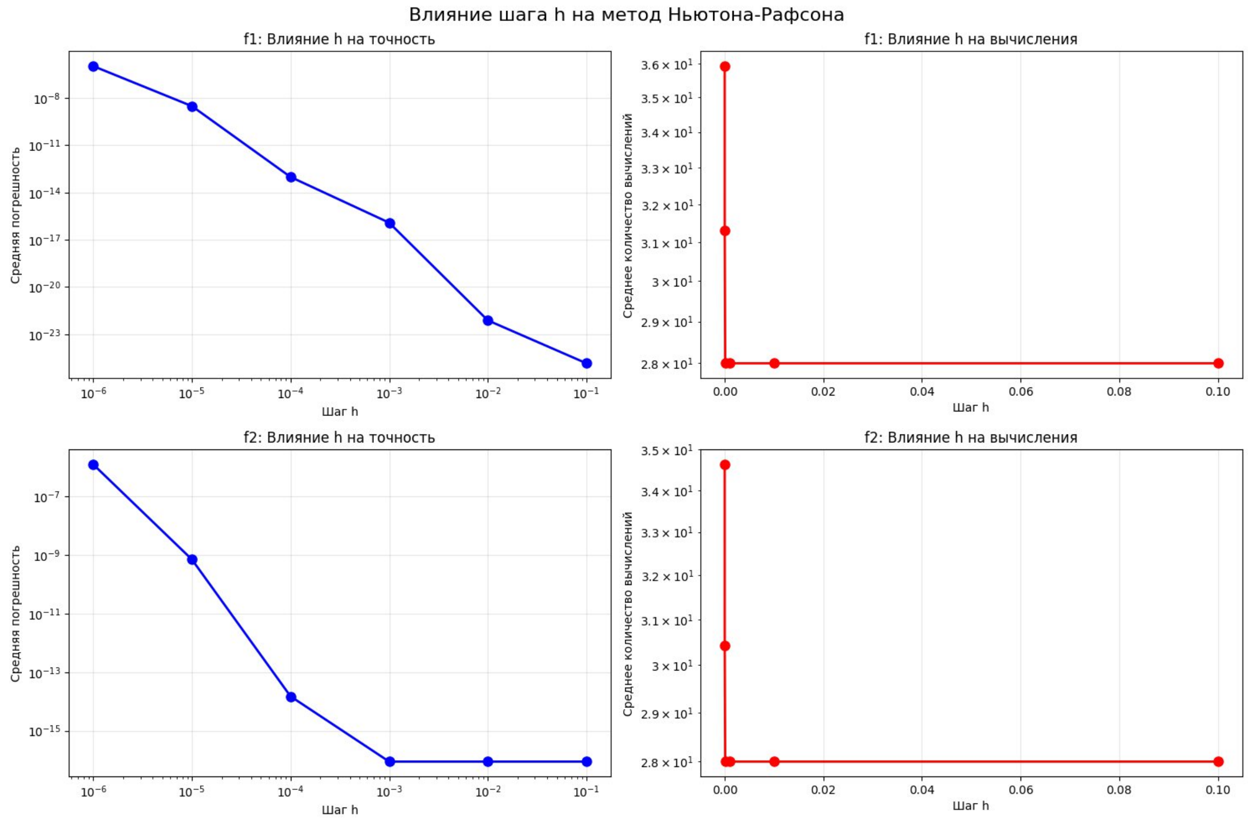

Рисунок 7 – влияние параметров на метод Ньютона-Рафсона

3 Выводы

Сравнительный анализ реализованных методов оптимизации второго порядка показал, что метод Ньютона демонстрирует максимальную эффективность на квадратичных функциях, достигая машинной точности ($\sim 10^{-20}$) всего за 2 вычисления целевой функции. На графиках зависимости точности от количества вычислений видно, что метод Ньютона обеспечивает мгновенную сходимость к точному решению благодаря использованию аналитических производных и квадратичной аппроксимации целевой функции.

Метод Ньютона-Рафсона показал хорошую универсальность, достигая точности порядка 10^{-5}–10^{-31}, однако требует значительно большего количества вычислений (28–42) из-за численного дифференцирования. На графиках влияния параметров видно, что точность метода сильно зависит от выбора шага дифференцирования h: при слишком малых значениях возрастает погрешность аппроксимации, а при больших – ошибка округления.

Анализ графиков зависимости точности от количества вычислений показывает, что метод Ньютона-Рафсона демонстрирует монотонное улучшение точности с ростом числа вычислений, в то время как метод Ньютона сразу достигает максимальной точности. На графиках влияния параметров видно, что оптимальный диапазон точности ε составляет 10^{-4}–10^{-6} для обоих методов, а оптимальный шаг h для метода Ньютона-Рафсона – 10^{-4}.

Таким образом, выбор метода оптимизации должен учитывать доступность аналитических производных: для функций с известными производными метод Ньютона обеспечивает сверхбыструю сходимость, в то время как для произвольных функций метод Ньютона-Рафсона предлагает универсальное решение с приемлемой точностью. Оба метода демонстрируют высокую эффективность на квадратичных функциях, но требуют внимательного выбора параметров для достижения оптимальной производительности.