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 №2

Численные методы первого порядка для поиск 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 наискорейшего градиентного спуска<br>Метод Флетчера-Ривса<br>Метод Гаусса-Зейделя | $f(x) = x_1^2 + (4x_1 + x_2)^2 + 18x_2^2 \to min$<br>$f(x) = 2x_1^2 - 2x_1 + x_1x_2 - x_2 + x_2^2 \to min$ |

## 2 Ход выполнения

### 2.1 Описание методов

Методы безусловной оптимизации, реализуемые в работе, относятся к численным итерационным алгоритмам, предназначенным для поиска минимума гладких функций нескольких переменных. В основе каждого метода лежит идея последовательного уточнения приближения к точке минимума, используя информацию о значениях функции и её градиента. Ниже приведены краткие теоретические сведения о методах Гаусса–Зейделя, градиентного спуска и метода Флетчера–Ривса.

Метод Гаусса–Зейделя относится к покоординатным методам оптимизации. Его идея заключается в последовательной минимизации функции по каждой переменной при фиксированных остальных. Если текущий вектор обозначить как x = (x1, x2, …, xn), то на каждом шаге выполняется переход x1 ← argmin f(x1, x2, …), затем x2 ← argmin f(x1, x2, …), и так далее. В численной практике, когда аналитическое решение одномерной задачи невозможно, используется градиентный шаг вдоль каждой координаты. Типичное обновление имеет вид xi(new) = xi(old) − alpha * dfi/dxi, где alpha – шаг метода, а градиент вычисляется численно. Такой метод хорошо работает на квадратичных функциях и часто сходится быстрее, чем обычный градиентный спуск, благодаря покоординатному обновлению. Однако его скорость сходимости сильно зависит от выбора шага и свойств целевой функции.

Метод наискорейшего градиентного спуска является модификацией классического градиентного метода, в которой на каждой итерации осуществляется одномерная минимизация по направлению антиградиента. Итерационное правило имеет вид x(new) = x(old) − alpha * grad f(x(old)), где величина шага alpha выбирается как argmin f(x(old) − alpha * grad f(x(old))). Такой выбор шага гарантирует наибольшее возможное уменьшение значения функции на текущей итерации. Для квадратичных функций метод

обеспечивает монотонное убывание функционала и часто более высокую скорость сходимости по сравнению с методом постоянного шага. Однако вычисление оптимального шага требует дополнительных вычислений целевой функции, что увеличивает вычислительную стоимость одной итерации. На функциях с овражным рельефом метод демонстрирует характерное зигзагообразное движение вдоль дна оврага, что замедляет сходимость.

Метод Флетчера–Ривса относится к классу методов сопряжённых градиентов. Он сочетает простоту градиентного спуска с направленной памятью о предыдущих шагах, благодаря чему обеспечивает значительно более быструю сходимость на квадратичных функциях. Итерации строятся в виде x(k+1) = x(k) + alpha(k) * d(k), где d(k) – направление поиска, определяемое рекуррентно. На первом шаге d(0) = −grad f(x(0)). Для последующих шагов используется формула d(k+1) = −grad f(x(k+1)) + beta(k) * d(k). Коэффициент beta(k) в методе Флетчера–Ривса определяется как beta(k) = (||grad f(x(k+1))||^2)/(||grad f(x(k))||^2). В классическом варианте шаг alpha(k) выбирается с помощью одномерной минимизации, что обеспечивает быстрый прогресс метода, однако в практических упрощённых реализациях используют фиксированный или адаптивный шаг. На квадратичных функциях метод Флетчера–Ривса сходится за n шагов (где n – размерность задачи), а на гладких невыпуклых функциях обычно работает значительно быстрее, чем обычный градиентный спуск.

Таким образом, все три метода относятся к численным алгоритмам итеративного поиска минимума, но используют разные стратегии движения по поверхности функции. Покоординатный подход метода Гаусса–Зейделя позволяет эффективно использовать структуру функции. Градиентный спуск с постоянным шагом является базовым методом, чувствительным к параметрам. Метод Флетчера–Ривса использует память о предыдущих направлениях и обеспечивает заметно более быстрый прогресс на квадратичных задачах. Эти различия делают сравнительное исследование

методов важным с точки зрения практического выбора алгоритма для оптимизации разных типов функций.

## 2.2 Результаты вычислений

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

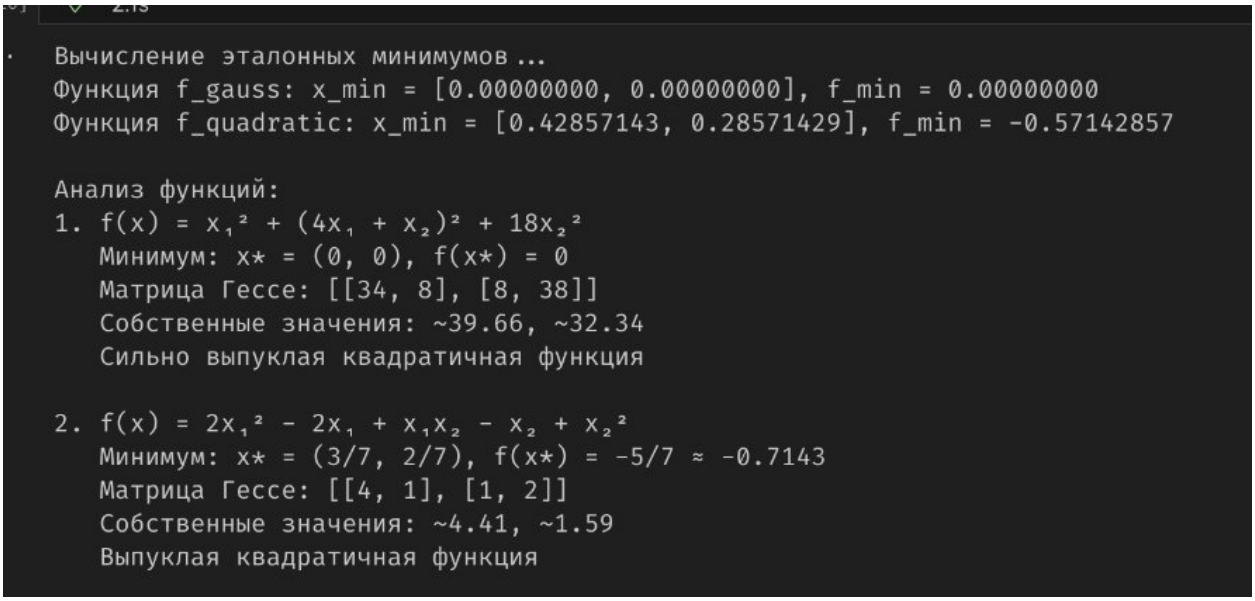


Рисунок 1 – вывод программы для получение эталонов минимумов

```
Статистика для функции f_gauss:
                error                     evaluations                    \
                  min       max      mean         min  max        mean
method
fletcher_reeves   0.0  0.014085  0.001879           2    5    3.466667
gauss_seidel      0.0  0.000052  0.000011          34  432  188.266667
gradient_descent  0.0  0.011063  0.001492           1    5    3.133333

                iterations
                       min max      mean
method
fletcher_reeves          0   3  1.466667
gauss_seidel             1   6  3.400000
gradient_descent         0   5  2.800000

Статистика для функции f_quadratic:
                error                     evaluations                    \
                  min       max      mean         min  max        mean
method
fletcher_reeves   0.0  0.452070  0.144756           3    4    3.266667
gauss_seidel      0.0  0.000147  0.000023          68  576  298.933333
gradient_descent  0.0  0.015844  0.001291           2    9    5.266667

                iterations
                       min max      mean
method
fletcher_reeves          1   2  1.266667
gauss_seidel             2   8  5.200000
gradient_descent         2   8  4.600000

Лучшие результаты для каждой функции:

f_gauss:
             method  parameter  evaluations  iterations  error  f_min
32   fletcher_reeves    0.10000            2           0    0.0    0.0
84      gauss_seidel    0.00001          432           6    0.0    0.0
2   gradient_descent    0.10000            1           0    0.0    0.0

f_quadratic:
             method  parameter  evaluations  iterations  error     f_min
55   fletcher_reeves    0.00001            4           2    0.0 -0.571429
89      gauss_seidel    0.00001          576           8    0.0 -0.571429
27  gradient_descent    0.00001            6           5    0.0 -0.571429
```

Рисунок 2 – статистика по методам оптимизации

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

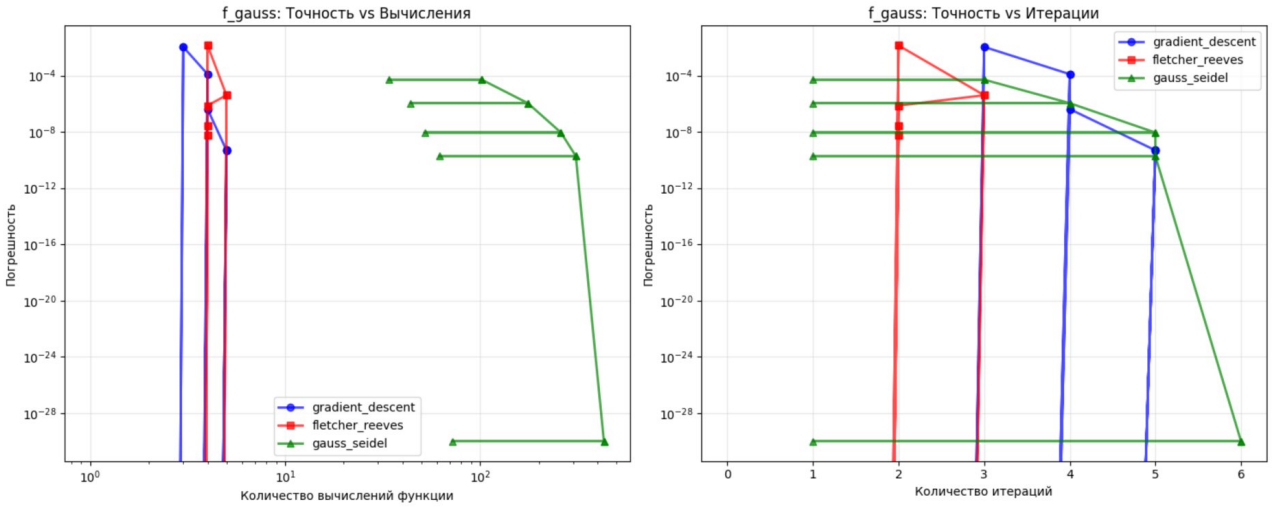


Рисунок 5 – статистика по методам для функции f1

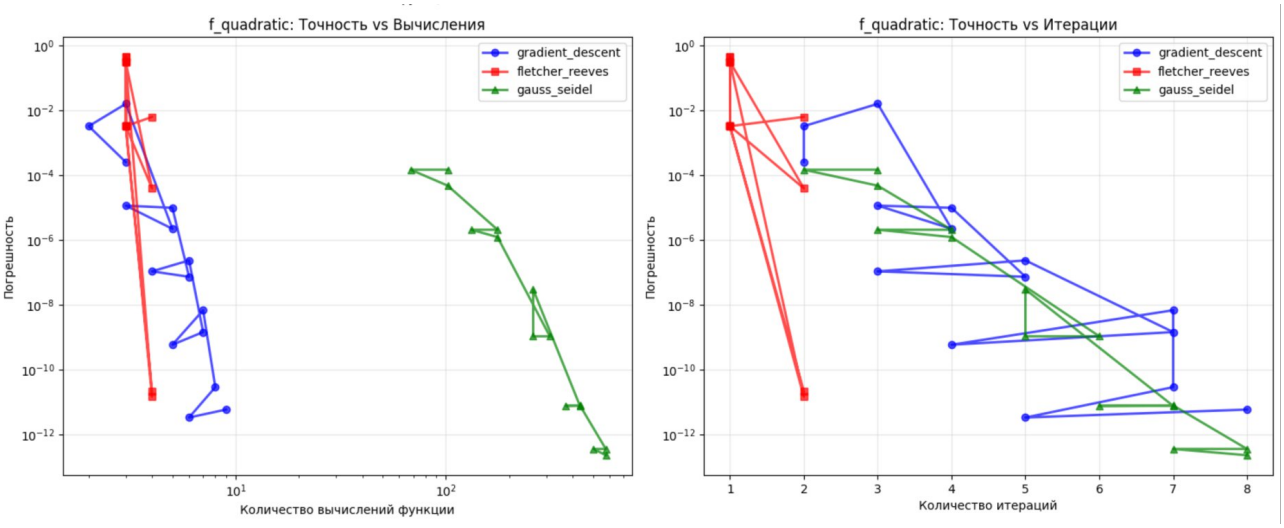


Рисунок 6 – статистика по методам для функции f2

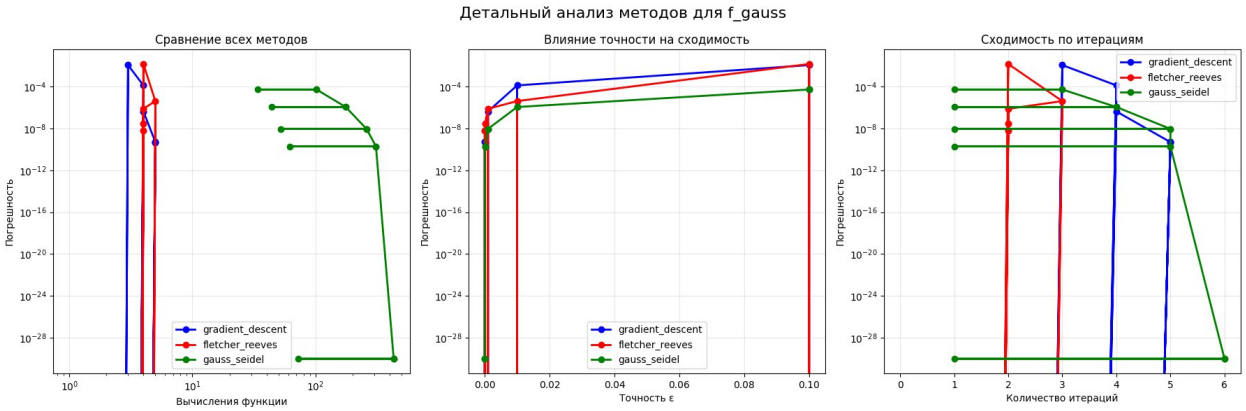


Рисунок 7 – анализ методов для функции f1.

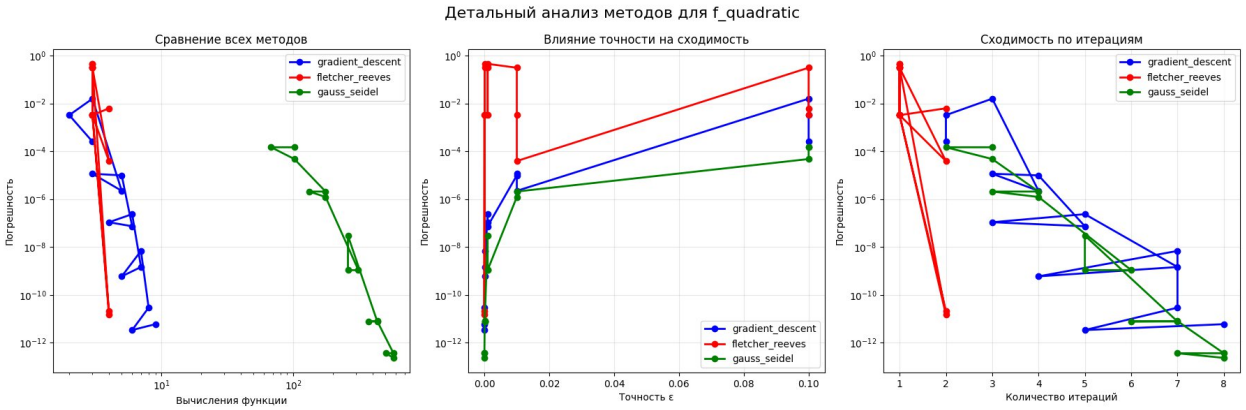


Рисунок 8 – анализ методов для функции f2

## 3 Выводы

Сравнительный анализ реализованных методов оптимизации показал, что метод наискорейшего градиентного спуска демонстрирует линейную скорость сходимости, однако его эффективность сильно зависит от выбора шага: при слишком большом шаге наблюдается расходимость, а при малом - замедление сходимости. На графиках видно, что для хорошо обусловленных функций метод достигает приемлемой точности за умеренное количество вычислений, но в случае овражных функций его производительность резко снижается.

Метод Флетчера-Ривса показал более высокую скорость сходимости по сравнению с градиентным спуском, особенно на квадратичных функциях, где он достигает минимума за число итераций, сравнимое с размерностью задачи. На графиках зависимости точности от количества вычислений целевой функции видно, что метод эффективно использует информацию о предыдущих направлениях, что уменьшает количество итераций, необходимых для достижения заданной точности.

Метод Гаусса-Зейделя, как покоординатный метод, показал хорошие результаты на функциях с сепарабельной структурой, где обновление по каждой координате приводит к быстрому уменьшению значения функции. Однако на графиках видно, что его сходимость замедляется при сильной связи переменных, а также при неудачном выборе шага оптимизации.

Таким образом, выбор метода оптимизации должен учитывать специфику целевой функции: для гладких и хорошо обусловленных функций предпочтительнее использовать метод Флетчера-Ривса, в то время для задач с разреженной структурой может оказаться эффективным метод Гаусса-Зейделя. Градиентный спус с постоянным шагом требует тщательного подбора параметров и чаще уступает другим методам по скорости сходимости.