Лабораторная работа №1

Методы нулевого порядка

Задачи:

1. Описание следующих методов:
   - метод градиентного спуска с постоянным шагом
   - методы одномерного поиска (метод дихотомии и метод золотого сечения) и градиентный спуск на их основе
   - метод Нелдера-Мида (использование готовой реализации в scipy.optimize)
2. Реализация вышеперечисленных методов
3. Исследование методов на эффективность на нескольких функциях:
   - зависимость количества итераций и вычислений значений минимизируемой функции и ее градиентов от желаемой точности
   - зависимость числа итераций от выбора стартовой точки
   - зависимость числа итераций от размерности пространства
   - зависимость числа итераций от обусловленности функции
   - зависимость числа итераций от зашумленности функции
   - поведение методов при наличии у функции нескольких оптимумов

Часть 1. Описания используемых методов

Методы одномерного поиска

Метод дихотомии, алгоритм:

1. Поиск значения функции f в точках $m = \frac{a+b}{2}, x_1 = \frac{a+m}{2}, x_2 = \frac{b+m}{2}$, где $a, b \ (a < b)$ – концы отрезка, на котором производится поиск минимума
2. Если $f(x_1) < f(m)$, то поиск (п. 1) на отрезке (a, m)
3. Если $f(x_2) > f(m)$, то поиск (п. 1) на отрезке (x_1, x_2)
4. Иначе поиск (п. 1) на отрезке (m, b)

Алгоритм продолжается до тех пор, пока не будет достигнута заданная точность

Метод золотого сечения, алгоритм:

1. Поиск значения функции f в точках $x_1 = b - \frac{b-a}{\varphi}, x_2 = a + \frac{b-a}{\varphi}$, где $a, b \ (a < b)$ – концы отрезка, на котором производится поиск минимума, $\varphi = \frac{1+\sqrt{5}}{2}$
2. Если $f(x_1) < f(x_2)$, то поиск на отрезке (a, x_2)
3. Если $f(x_1) \geq f(x_2)$, то поиск на отрезке (x_1, b)

Алгоритм продолжается до тех пор, пока не будет достигнута заданная точность. На каждом шаге можно использовать значение функции, вычисленное ранее, из-за чего происходит только одно вычисление функции за шаг.

Методы многомерного поиска

Метод градиентного спуска, алгоритм:

1. Вычисление $x_{(i+1)} = x_{(i)} - \lambda_i \nabla f$, где $\lambda_i = argmin_\lambda f(x_{(i)} - \lambda \nabla f(x_{(i)}))$, либо шаг λ может быть постоянным (тогда это градиентный спуск с постоянным шагом)
2. Проверка условия остановки – достижения заданной точности

Метод Нелдера-Мида, алгоритм:

1. Выбор $n + 1$ точки – симплекса n-мерного пространства
2. Нахождение среди выбранных точек:
   - x_h – точка, в которой значение функции максимально
   - x_g – точка, в которой значение функции следующее по величине

- x_l – точка, в которой значение функции минимально

3. Вычисление x_c – центра тяжести симплекса
4. Отражение x_h относительно x_c с коэффициентом отражения $\alpha > 0$: $x_r = (1 + \alpha)x_c - \alpha x_h$
5. Если $f(x_r) < f(x_l)$, то направление выбрано удачное и можно попробовать увеличить шаг (если значение функции на увеличится, производим «растяжение» на коэффициент растяжения).
   Если $f(x_l) < f(x_r) < f(x_g)$, то $x_h = x_r$
   Если $f\left(x_g\right) < f(x_r) < f(x_h)$, то меняем местами x_h и x_r
   Если $f(x_h) < f(x_r)$, то производим «сжатие» (если значение в новой точке не станет еще больше)
6. Проверка условия остановки – достижения заданной точности

При исследовании используется алгоритм Нелдера-Мида, реализованный в библиотеке scipy.optimize.

Помимо используемого метода библиотека scipy.optimize предоставляет и другие функции для минимизации (или максимизации) целевых функций, возможно, с учетом ограничений. Она включает в себя решатели нелинейных задач (с поддержкой алгоритмов локальной и глобальной оптимизации), линейное программирование, ограниченный и нелинейный метод наименьших квадратов, поиск корней и подгонку кривой.

Метод локальной минимизации Нелдера-Мида вызывается с помощью функции scipy.optimize.minimize, где аргумент method=‘Nelder-Mead’. Функция так же принимает минимизируемую функцию, стартовую точку и может принимать другие аргументы (например, для метода Нелдера-Мида может принимать значения коэффициентов отражения, растяжения и сжатия)

Часть 2. Исследование эффективности методов на выбранных функциях

2.0. Реализация методов

2.1. Исследование зависимости количества итераций и вычислений значений минимизируемой функции и ее градиентов от желаемой точности

В качестве функций, на которых будет проводиться исследование эффективности методов поиска, были взяты функции:

- $x^2 + y^2 – xy$ (рис. 1)
- $x^4 + y^2 + x^2y$ (рис. 2)
- $(1 – x)^2 + 100(y – x^2)^2$ – функция Розенброка (рис. 3, 4)

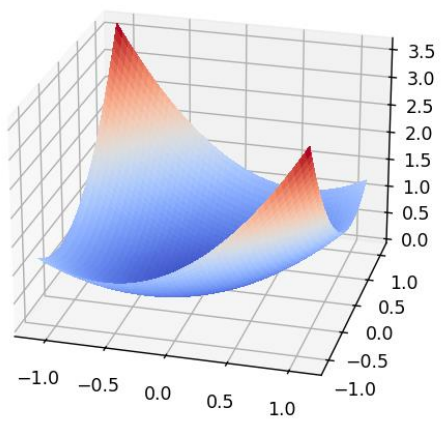


Рис. 1. График функции $x^2 + y^2 – xy$

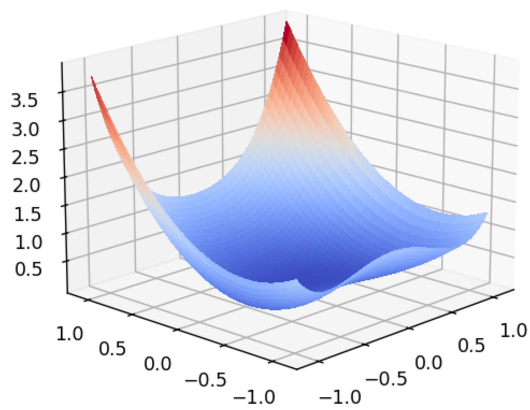


Рис. 2. График функции $x^4 + y^2 + x^2y$

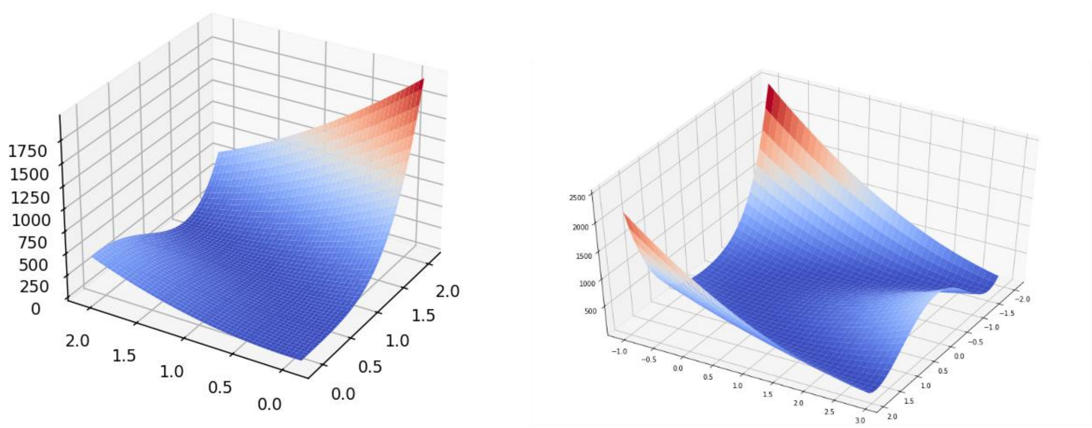
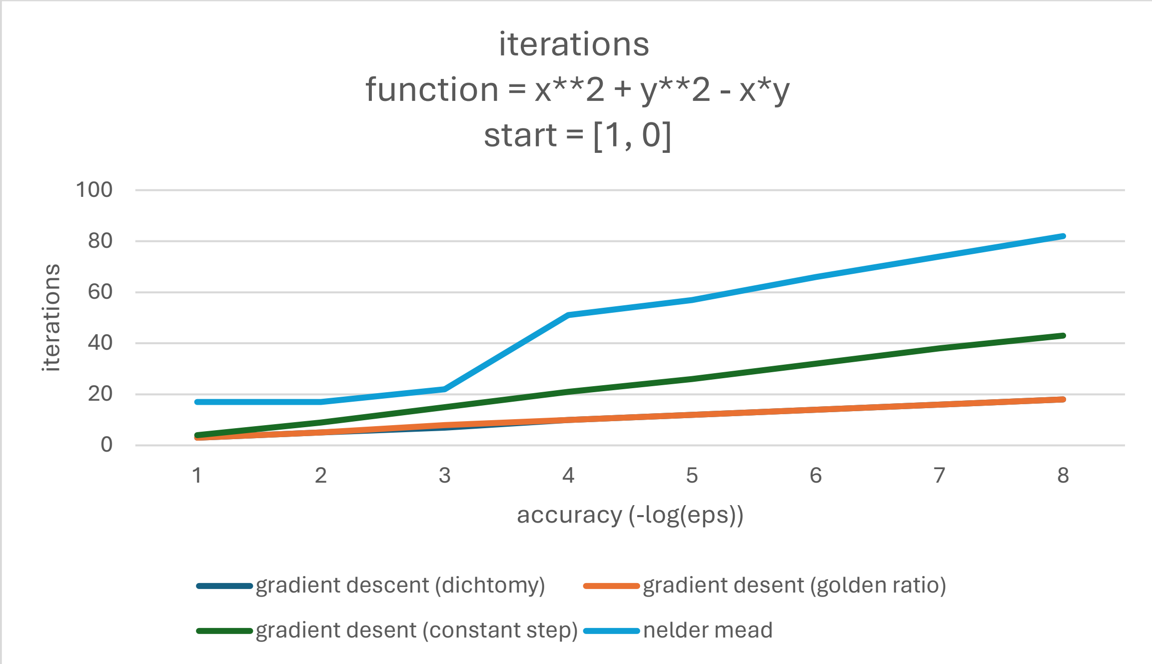
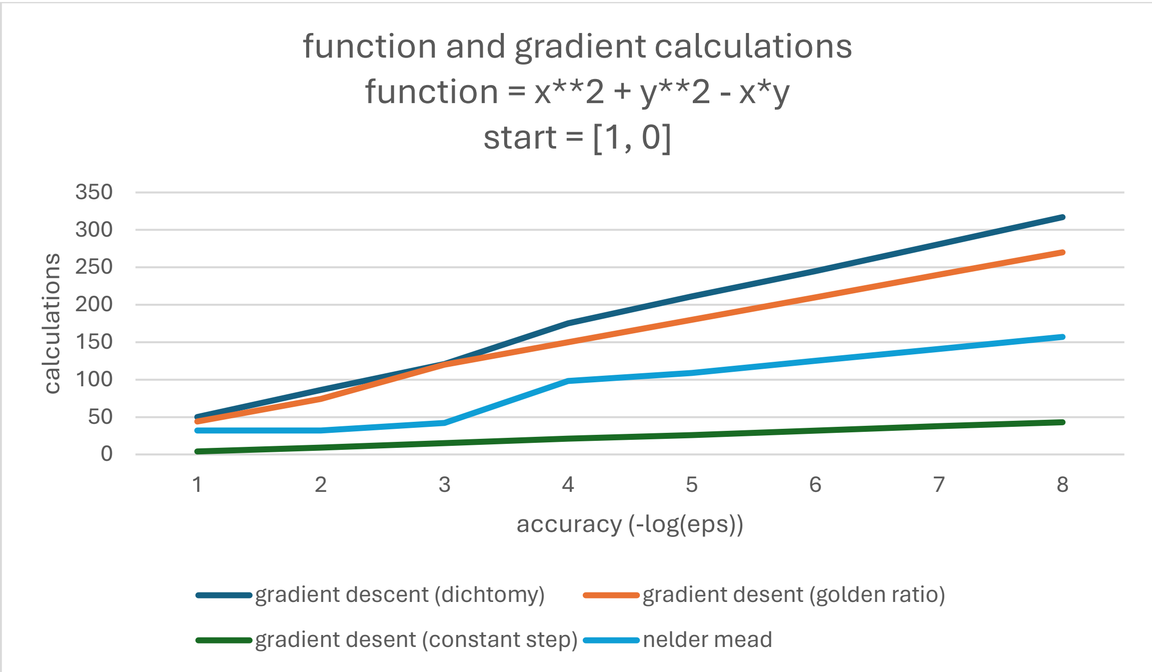
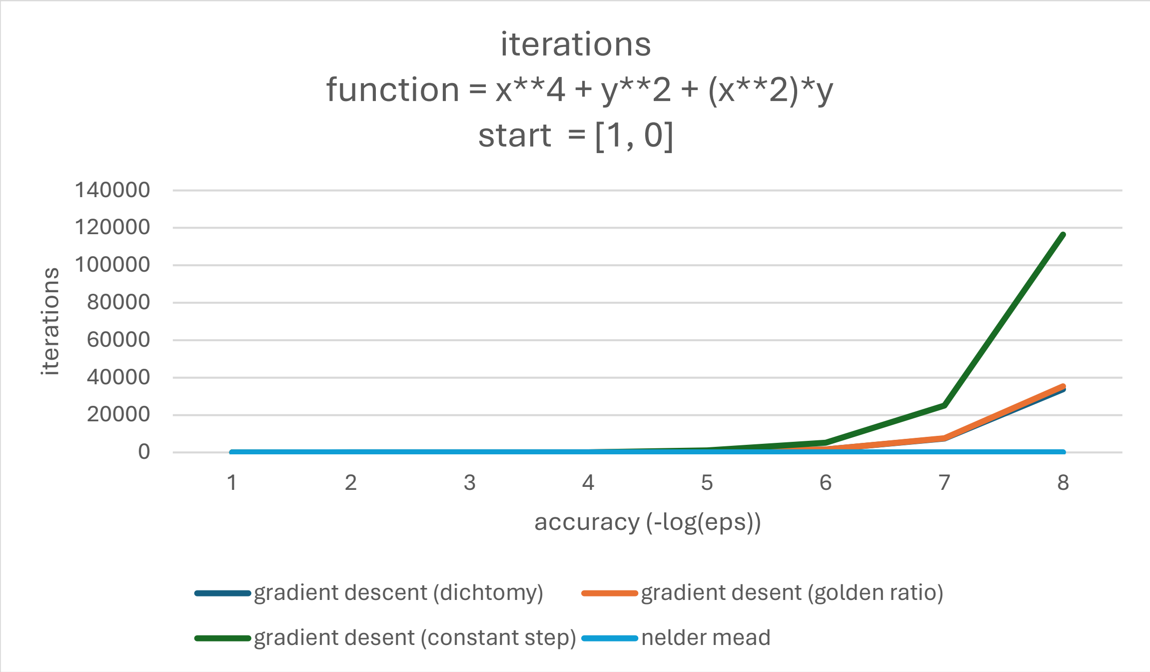
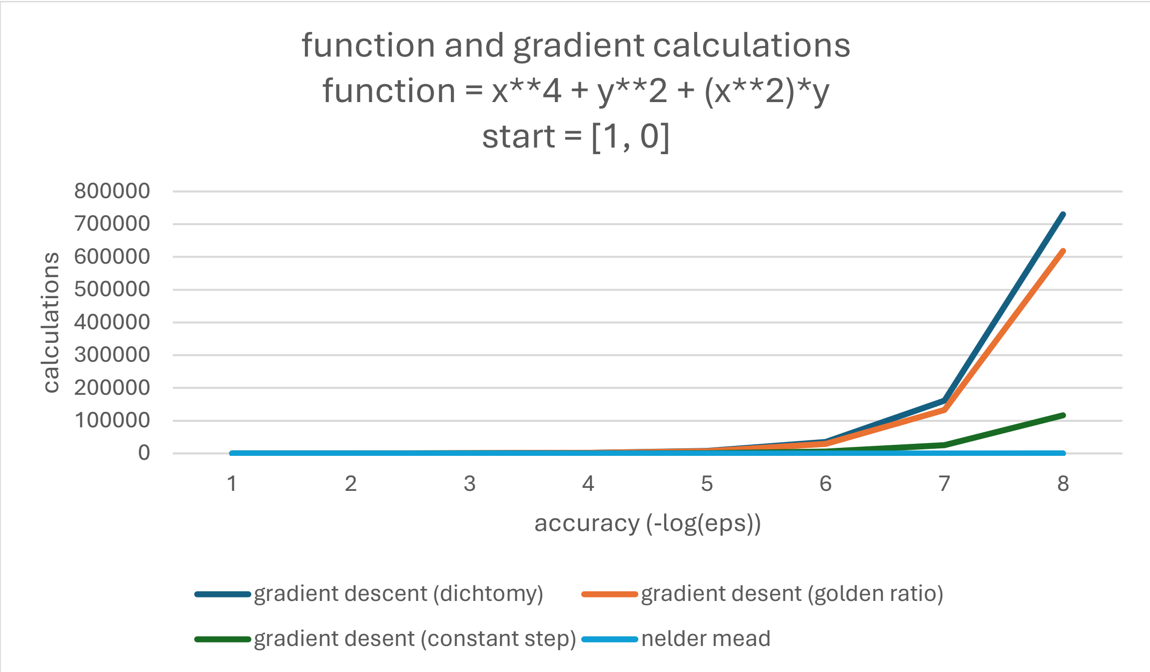


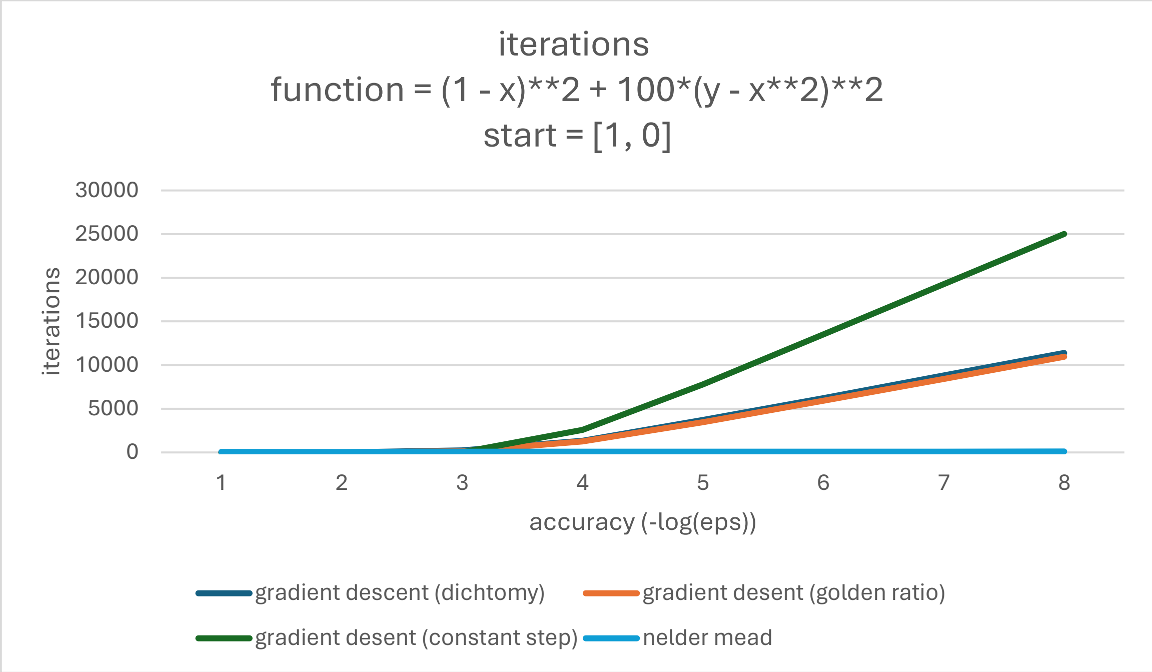
Рис. 3, 4. Графики функции $(1 – x)^2 + 100(y – x^2)^2$

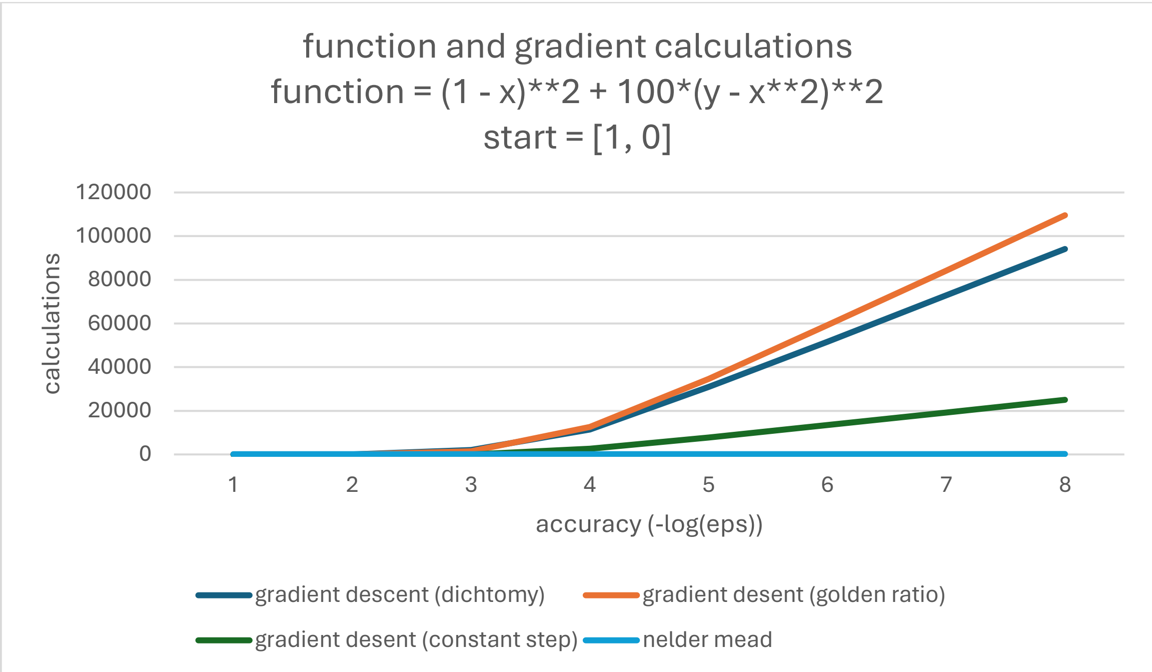

iterations
function = x**2 + y**2 - x*y
start = [1, 0]
iterations
100
80
60
40
20
0
1
2
3
4
5
6
7
8
accuracy (-log(eps))
gradient descent (dichtomy)
gradient desent (golden ratio)
gradient desent (constant step)
nelder mead


function and gradient calculations
function = x**2 + y**2 - x*y
start = [1, 0]
calculations
350
300
250
200
150
100
50
0
1
2
3
4
5
6
7
8
accuracy (-log(eps))
gradient descent (dichtomy)
gradient desent (golden ratio)
gradient desent (constant step)
nelder mead


iterations
function = x**4 + y**2 + (x**2)*y
start = [1, 0]
140000
120000
100000
80000
60000
40000
20000
0
iterations
1
2
3
4
5
6
7
8
accuracy (-log(eps))
gradient descent (dichtomy)
gradient desent (golden ratio)
gradient desent (constant step)
nelder mead


function and gradient calculations
function = x**4 + y**2 + (x**2)*y
start = [1, 0]
800000
700000
600000
500000
400000
300000
200000
100000
0
calculations
1
2
3
4
5
6
7
8
accuracy (-log(eps))
gradient descent (dichtomy)
gradient desent (golden ratio)
gradient desent (constant step)
nelder mead





Из графиков для рассматриваемых функций видно, что метод Нелдера-Мида мало зависит/не зависит от точности. В то же время, зависимость числа итераций от точности для методов градиентного спуска похожа на линейную/экспоненциальную (в зависимости от функции).Для градиентных методов с разными способами поиска минимума число итераций практический совпадает, но число вычислений функции у метода с дихотомией больше, чем у метода с золотым сечением, поскольку первый требует больше итераций вычисления для достижения нужной точности.

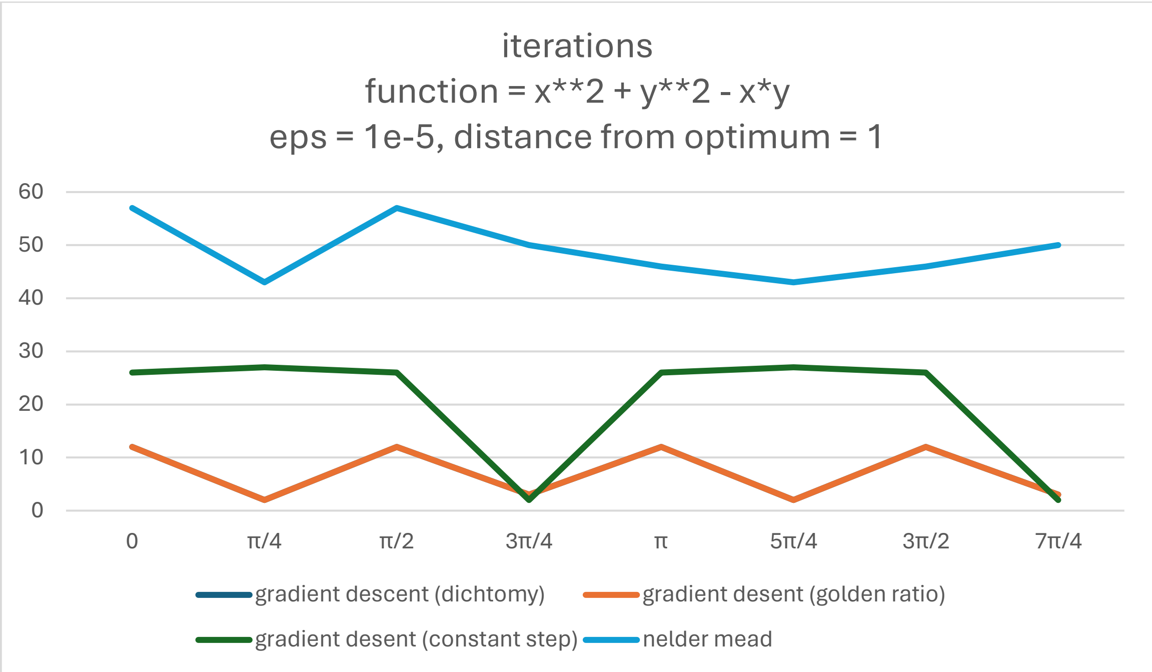
Несмотря на меньшее число итераций, количество вычислений функции и ее градиентов для метода градиентного спуска с поиском минимума на отрезке больше, чем для градиентного спуска с постоянным шагом, поскольку второй за одну итерацию высчитывает только значения градиента в точке. Однако, чтобы градиентный спуск оптимально работал, необходимо подобрать нужный шаг, что довольно сложно бывает на практике.

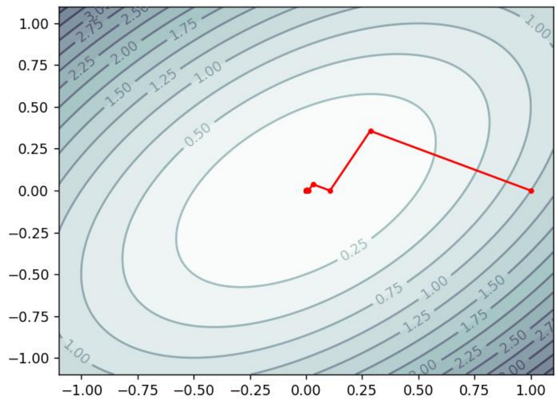
2.2. Исследование зависимости количества итераций от положения начальной точки

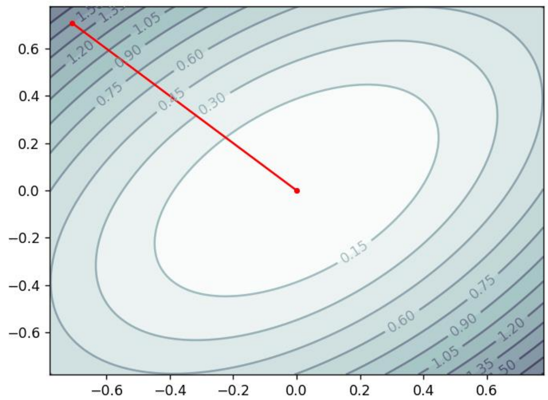
1. Зависимость количества итераций от угла наклона точки, относительно оси OX, на одинаковом расстоянии от точки экстремума

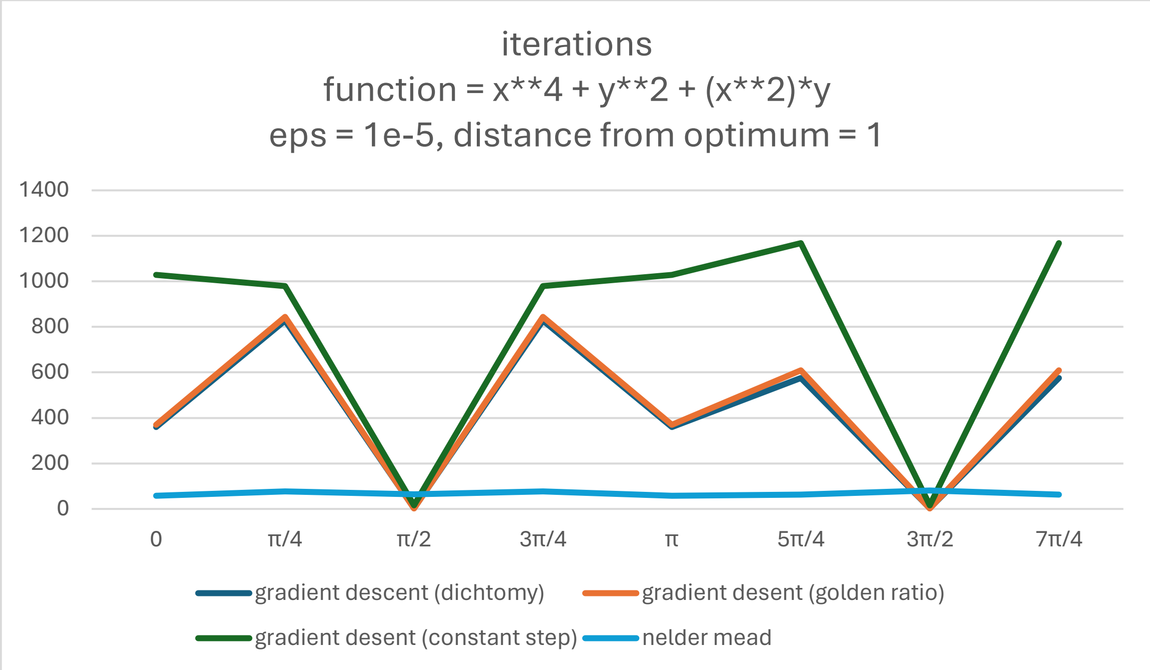
Для углов, при которых точка находится на плоскости симметрии функции (при этом минимум также находится на этой плоскости), поиск минимума происходит значительно быстрее.

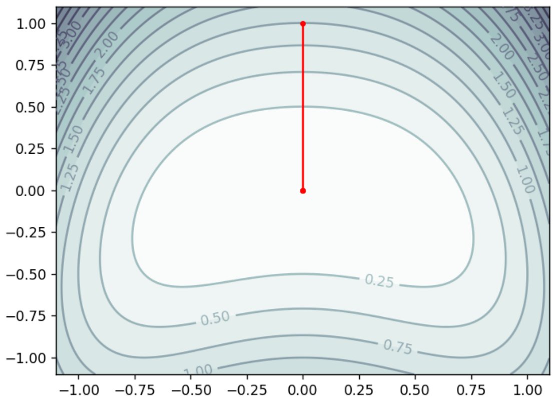
В общем случае для каждой функции зависимость от угла наклона точки уникальна и зависит от особенностей функции и её градиента.

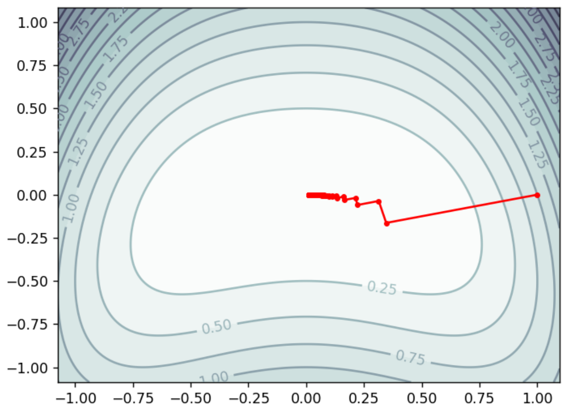


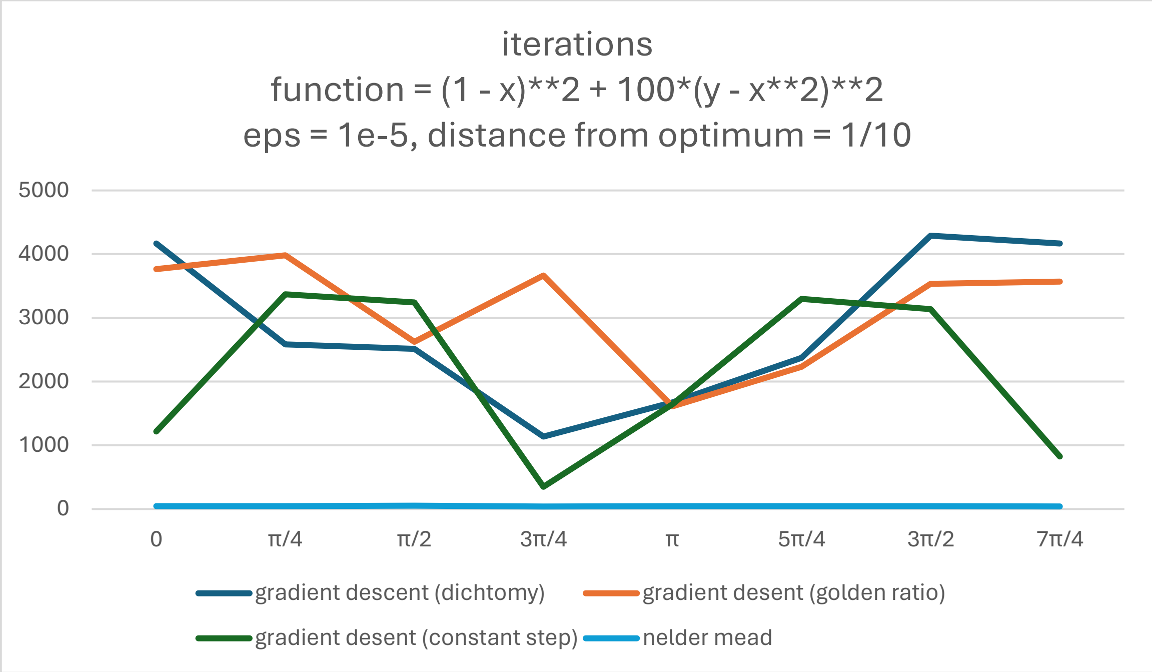





iterations
function = x**4 + y**2 + (x**2)*y
eps = 1e-5, distance from optimum = 1
1400
1200
1000
800
600
400
200
0
0
π/4
π/2
3π/4
π
5π/4
3π/2
7π/4
gradient descent (dichtomy)
gradient desent (golden ratio)
gradient desent (constant step)
nelder mead



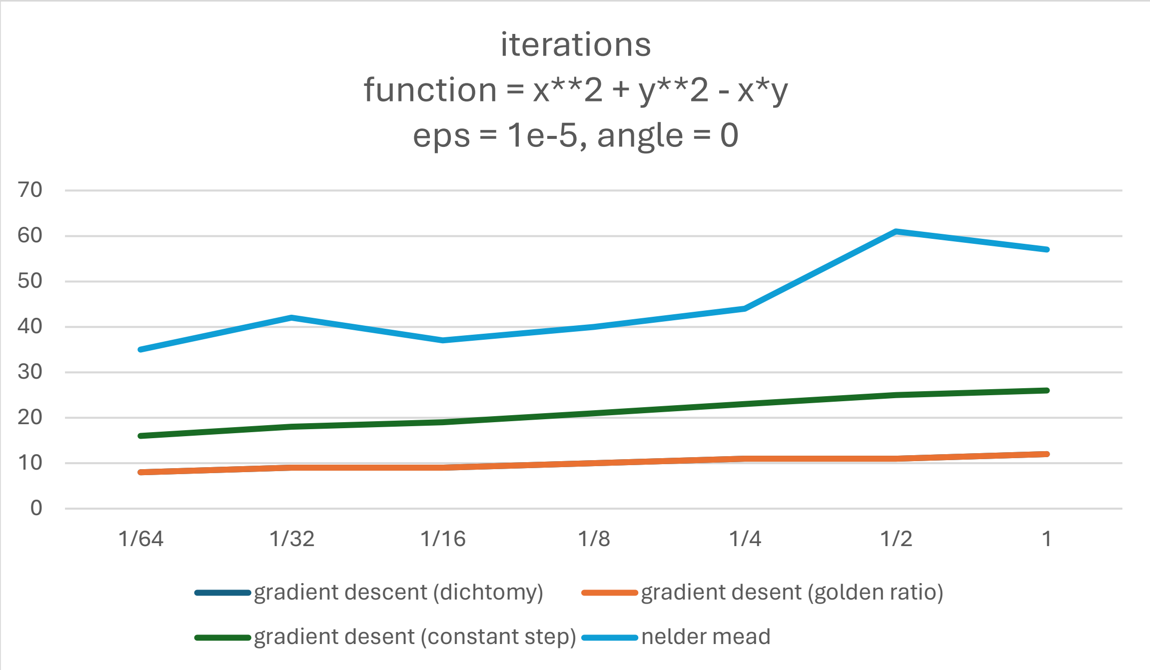




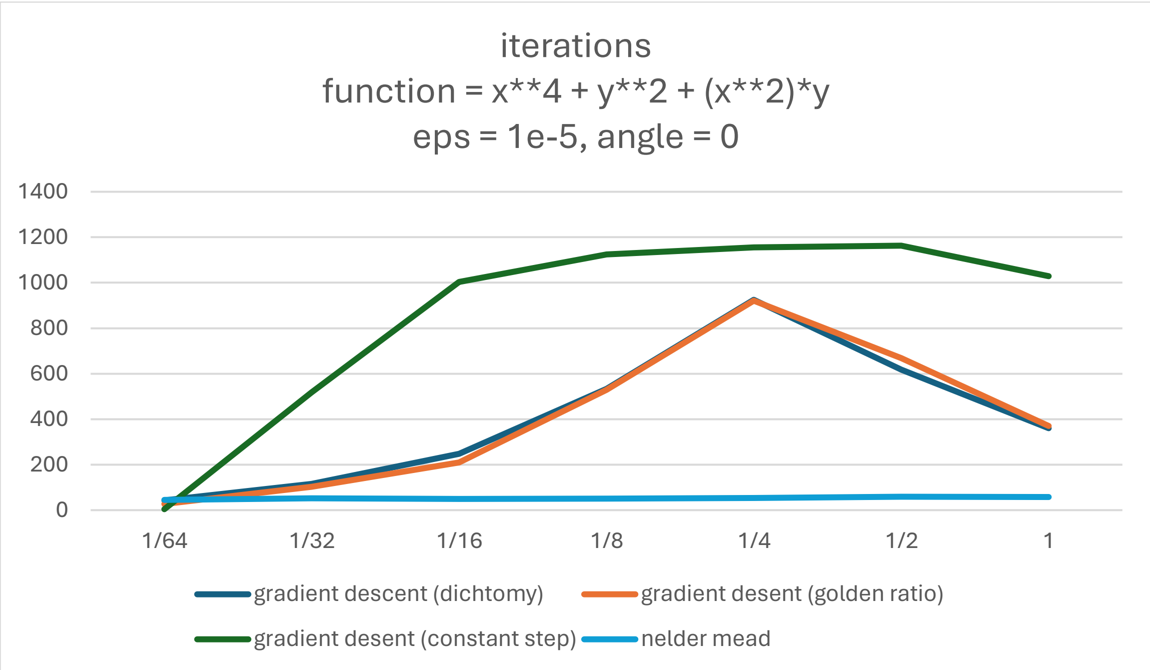
iterations
function = (1 - x)**2 + 100*(y - x**2)**2
eps = 1e-5, distance from optimum = 1/10
5000
4000
3000
2000
1000
0
0
π/4
π/2
3π/4
π
5π/4
3π/2
7π/4
gradient descent (dichtomy)
gradient desent (golden ratio)
gradient desent (constant step)
nelder mead

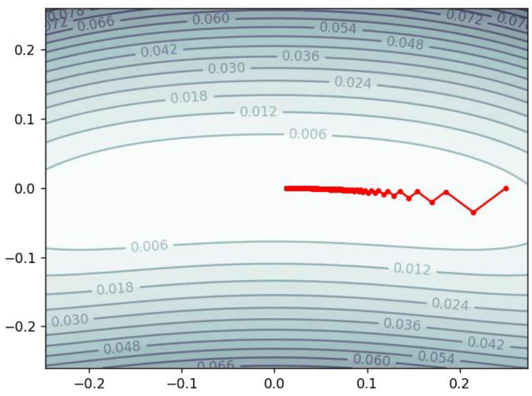
2. 1. Зависимость количества итераций от расстояния от точки экстремума при одинаковом угле наклона относительно оси OX

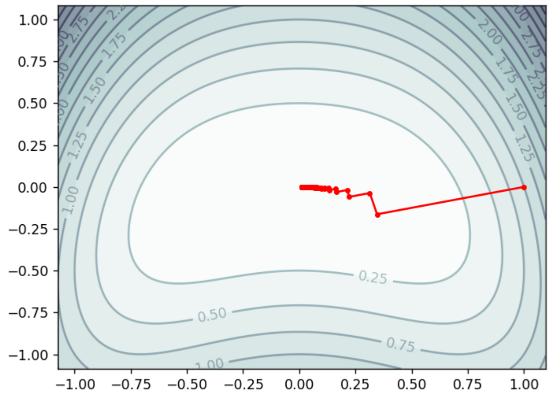
В данном случае поиск происходит в плоскости симметрии функции, поэтому расстояние незначительно влияет на количество итераций. Чем дальше стартовая точка находится от экстремума, тем больше будет очередной шаг и тем быстрее будет приближение к оптимуму.



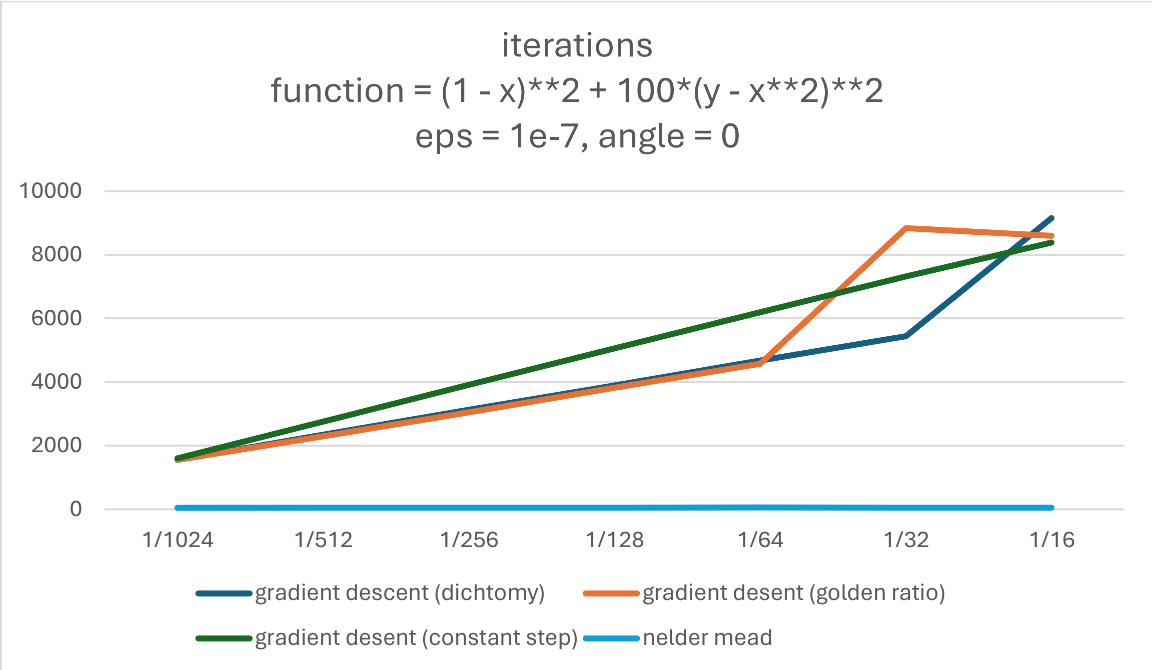
В этом же случае из-за того, что увеличение длины происходит не по оси симметрии, траектория поиска методом градиентного спуска может значительно меняться, из-за чего меняется и количество итераций

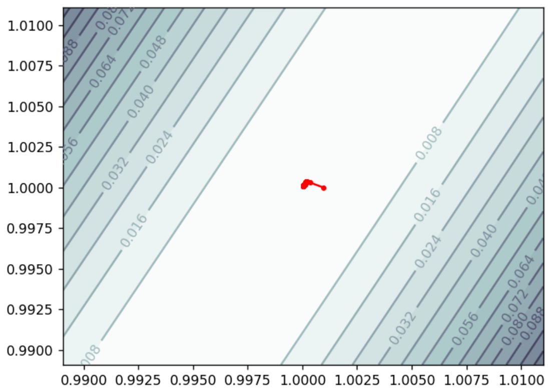


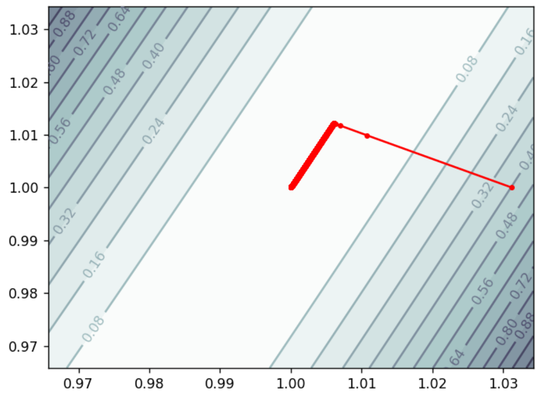




Для функции Розенброка наблюдается примерно линейная зависимость градиентного метода от расстояния от точки экстремума

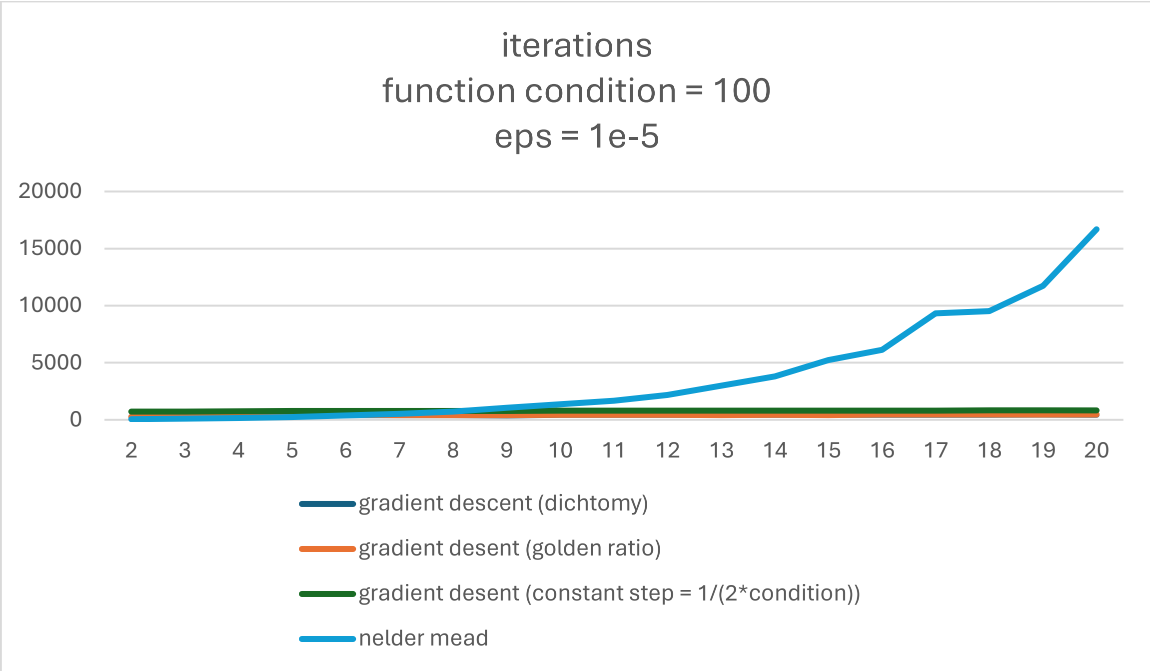






2.3. Исследование зависимости количества итераций от размерности пространства

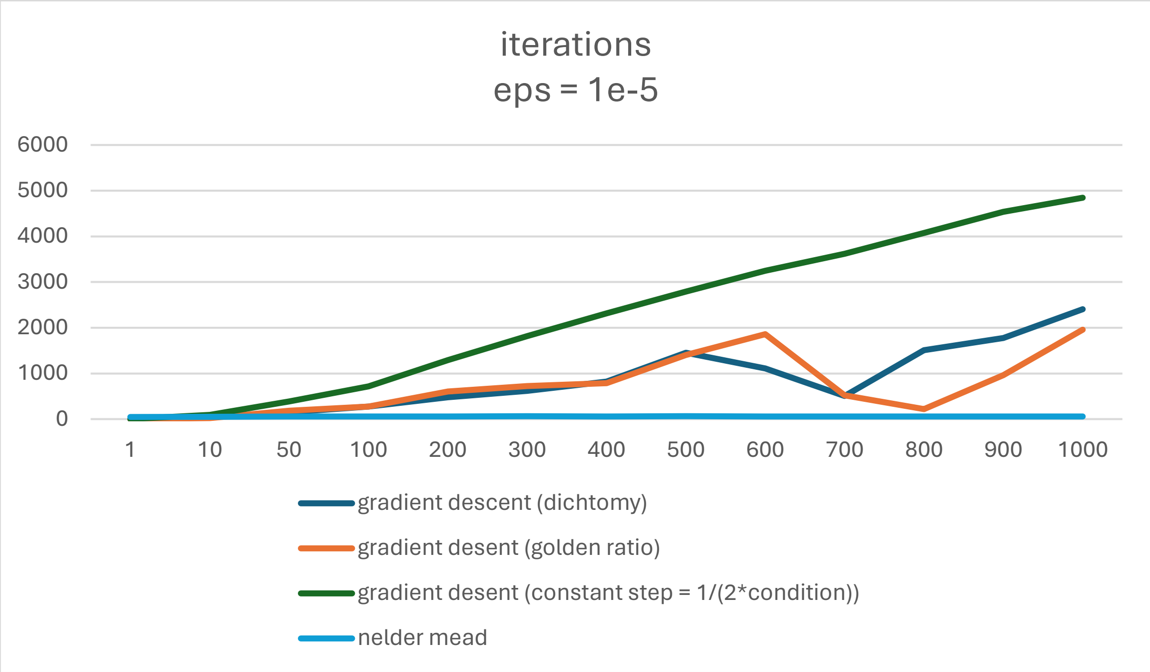
По графику видно, что метод градиентного спуска не зависит от размерности пространства, в то время как зависимость метода Нелдера-Мида похожа на экспоненциальную

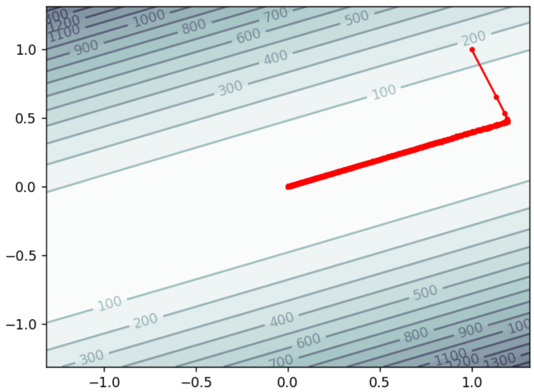


2.4. Исследование зависимости количества итераций от обусловленности функции

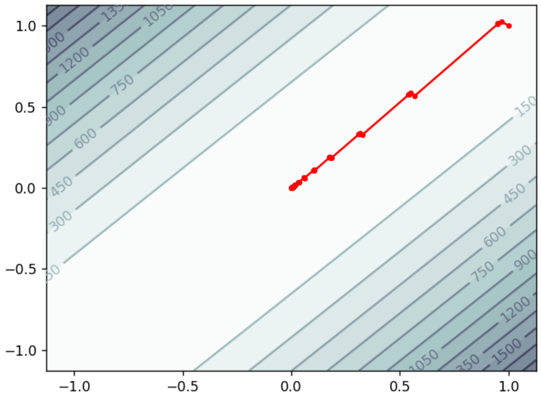
Из графика видно, что метод Нелдера-Мида не зависит от обусловленности функции, в то время как методы градиентного спуска зависят примерно линейно.

Для градиентных методов с поиском минимума одномерными методами наблюдается «провал» функции зависимости, что обуславливается особенностью генерируемых функций.





Градиентный спуск на основе метода золотого сечения при числе обусловленности 500 (потребовалось 1396 итераций)

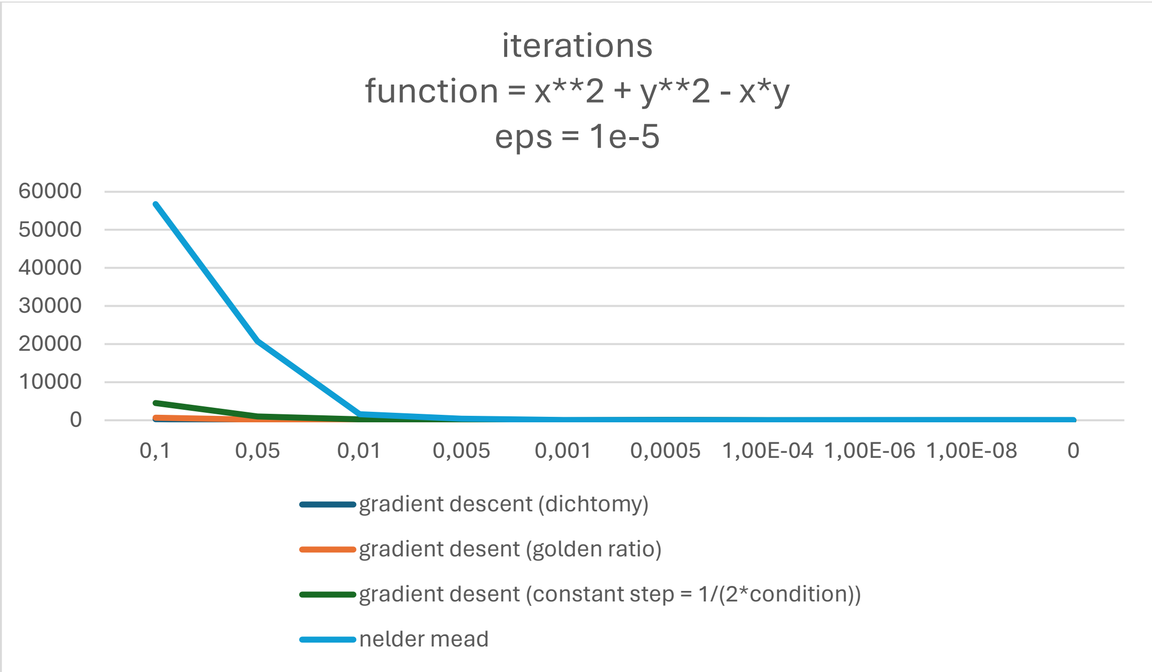


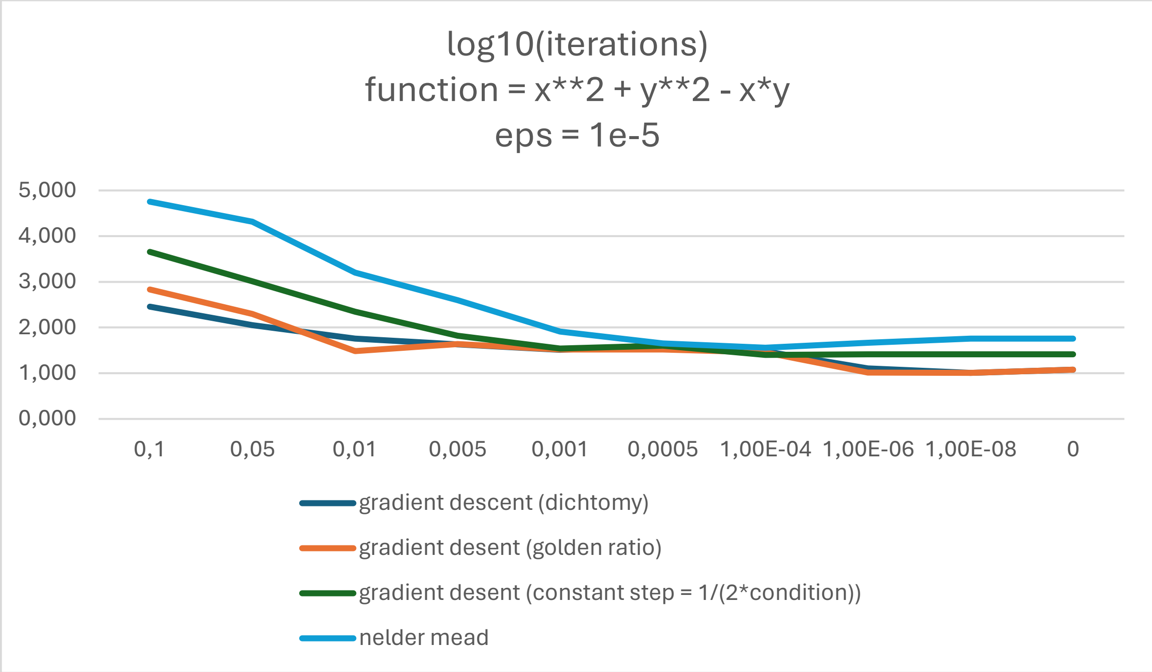
Градиентный спуск на основе метода золотого сечения при числе обусловленности 750 (потребовалась 51 итерация)

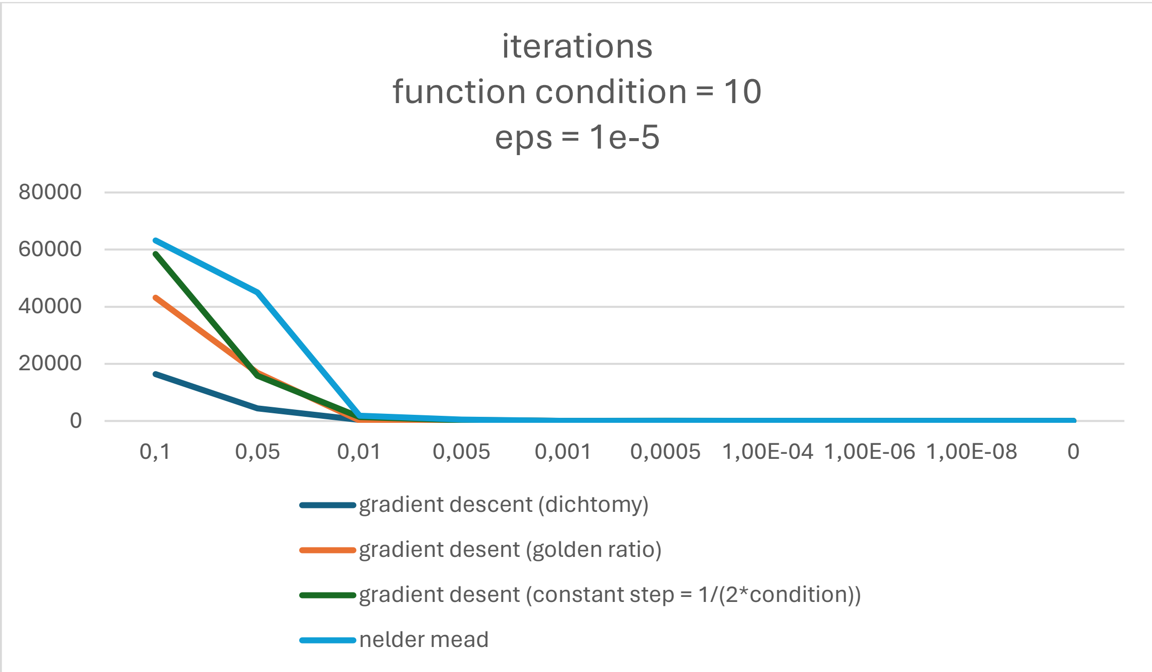
2.5. Исследование зависимости количества итераций от зашумленности функции

Из графиков видно, что для любого метода поиска время выполнения увеличивается при увеличении зашумленности функции. Например, для функции $x^2 + y^2 - xy$ метод Нелдера-Мида зависит от зашумленности сильнее, чем методы градиентного спуска. Однако при рассмотрении случайной квадратичной функции (с числом обусловленности 10) видно, что зависимость примерно одинаковая.

При слишком высокой зашумленности функции каждый из методов будет работать непредсказуемо, поиск может занять долгое время или вовсе не остановиться (если нет ограничения на число итераций)




log10(iterations)
function = x**2 + y**2 - x*y
eps = 1e-5
5,000
4,000
3,000
2,000
1,000
0,000
0,1
0,05
0,01
0,005
0,001
0,0005
1,00E-04
1,00E-06
1,00E-08
0
gradient descent (dichtomy)
gradient desent (golden ratio)
gradient desent (constant step = 1/(2*condition))
nelder mead

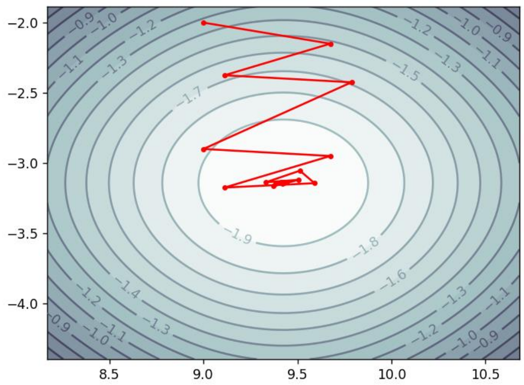

iterations
function condition = 10
eps = 1e-5
80000
60000
40000
20000
0
0,1
0,05
0,01
0,005
0,001
0,0005
1,00E-04
1,00E-06
1,00E-08
0
gradient descent (dichtomy)
gradient desent (golden ratio)
gradient desent (constant step = 1/(2*condition))
nelder mead

2.6. Исследование поведения методов на мультимодальных функциях

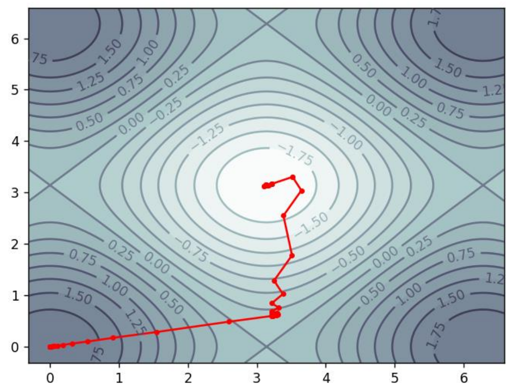
В качестве мультимодальной функции была взята $\cos(x) + \cos(y)$

Метод Нелдера-Мида в каждой из исследуемых точек находил точку локального минимума, поскольку сам метод не зависит от значений градиента и построен только на сравнении значений функции

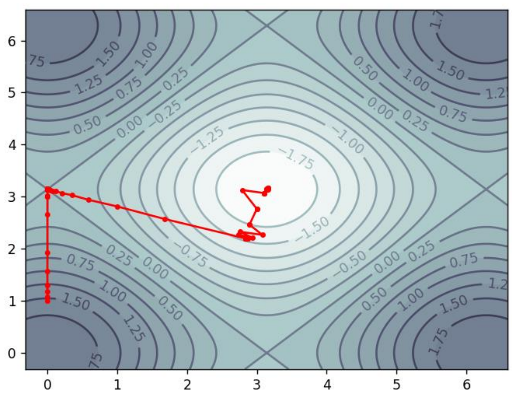
Запуск метода Нелдера-Мида с произвольной точки



Запуск метода Нелдера-Мида с точки максимума

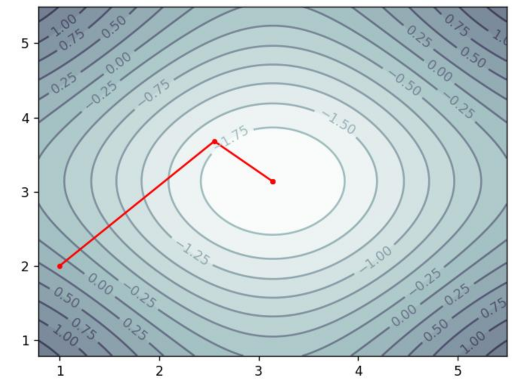


Запуск метода Нелдера-Мида, траектория проходит через седловую точку

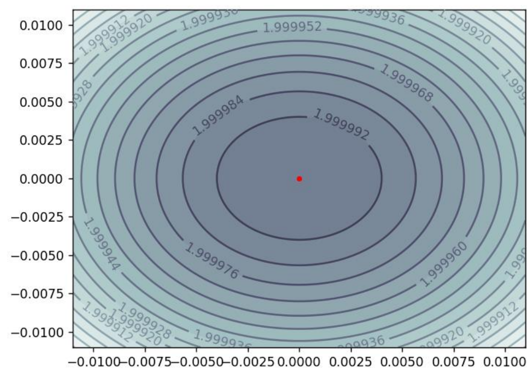


Методы градиентного спуска зависят от значения градиента в точке и, соответственно, находят ближайшую по направлению точку со значением градиента 0, которая необязательно является точкой минимума

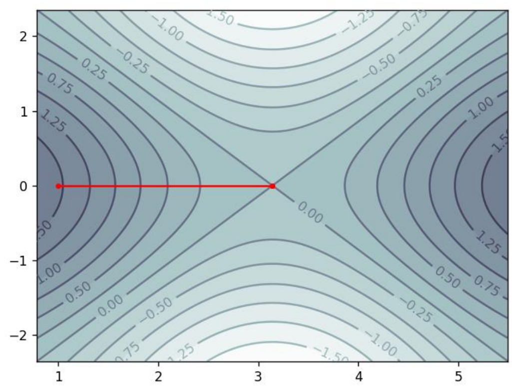
Запуск метода градиентного спуска с произвольной точки



Запуск метода градиентного спуска с точки максимума (в ней и остановился, так как градиент равен 0)



Запуск метода градиентного спуска, траектория проходит через седловую точку



Выводы

1. В зависимости от желаемой точности, количество итераций и количество вычислений функции растет
2. Работа методов градиентного спуска зависит от положения стартовой точки и обусловленности функции. Методы градиентного спуска не всегда ищут верное значение глобального минимума для мультимодальных функций
3. Работа метода Нелдера-Мида зависит от размерности пространства
4. Эффективность каждого из методов значительно уменьшается с увеличением зашумленности функции, методы в таких случая могут вести себя непредсказуемо.

Преимущества и недостатки методов

Метод	Преимущества	Недостатки
Градиентный спуск с постоянным шагом	Простота реализации (по сравнению с другими исследуемыми методами) Эффективность не зависит от размерности пространства Меньшее влияние зашумленности функции на эффективность по сравнению с методом Нелдера-Мида	Использование градиента при поиске (не для всех функций вычислить / вычислить точно) Сложность вычисления оптимальной длины шага Зависимость эффективности от выбора стартовой точки Зависимость эффективности от обусловленности функции Некорректная работа на мультимодальных функциях
Градиентный спуск, основанный на одномерном методе поиска	Эффективность не зависит от размерности пространства	Использование градиента при поиске (не для всех функций вычислить / вычислить точно)

	Меньшее влияние зашумленности функции на эффективность по сравнению с методом Нелдера-Мида	Зависимость эффективности от выбора стартовой точки Зависимость эффективности от обусловленности функции Некорректная работа на мультимодальных функциях
Метод Нелдера-Мида	Более высокая эффективность по сравнению с методами градиентного спуска Независимость эффективности от выбора стартовой точки Независимость эффективности от обусловленности функции Не требует вычисления градиента функции	Снижение эффективности при повышении размерности пространства Плохая работа (по сравнению с градиентными методами) на зашумленных функциях Некорректная работа на мультимодальных функциях (может найти локальный минимум)