# Методы нулевого порядка

лабораторная работа №1

**Коробов Иван Константинович**

**Мицеловский Артемий Антонович**

**Биктимиров Тимур Радикович**

---

M3234

# Содержание

# 1 Основное задание

$$f(x,y) \to \min_{x,y}$$

$$\alpha \ - \ learning\ rate$$

$$\bigtriangledown f(x,y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$$

$$\begin{cases} x = x - \alpha \cdot \frac{\partial f}{\partial x} \\ y = y - \alpha \cdot \frac{\partial f}{\partial x} \end{cases}$$

$$\|\overline{vec}\| = \sqrt{x^2 + y^2}$$

$$if \ \ (\| \bigtriangledown f(x,y)\| \ < \ \varepsilon \ || \ N \ > \ N_{\max}) \ \ break;$$

## 1.1 Метод градиентного спуска с постоянным шагом (learning rate)

$\alpha$ is constant, for example $\alpha = 10^{-3}$ or $\alpha = \dfrac{1}{2 \cdot \max(|A|, |B|)}$ for function $f(x,y) = Ax^2 + By^2$

## 1.2 Любой метод одномерного поиска и градиентный спуск на его основе (метод Дихотомии)

$$[a, b]$$

$$\text{mid} = \frac{a+b}{2}$$

$$\text{mid\_l}, \text{mid\_r} = \text{mid} \mp \frac{\varepsilon}{2}$$

$$\begin{cases} a = \text{mid} & \text{if } f((x,y) - \text{mid\_l} \cdot \bigtriangledown f(x,y)) \geq f((x,y) - \text{mid\_r} \cdot \bigtriangledown f(x,y)) \\ b = \text{mid} & \text{else} \end{cases}$$

$$if \ \ (a - b < \varepsilon) \ \ \alpha = \frac{a+b}{2}; \ break;$$

## 1.3 метод Нелдера-Мида. При этом используйте готовую реализацию в Python библиотеке scipy.optimize. Изучите возможности библиотеки scipy.optimize.

```
minimize(function, start_point, method='Nelder-Mead', options={'xatol': precision, 'disp': True},
↪  callback=callback)
```

К содержанию

## 1.4 Вывод

Для исследования были выбраны 2 функции — квадратичная с коэффициентами 2 и 3 (то есть $(x-2)^2+(y+3)^2$), и функция Розенброка с коэффициентами 1 и 100: $(1-x)^2+100(x-y^2)^2$. Большой коэффициент был выбран для наглядности.

При исследовании зависимости количества итераций и вычислений от требуемой точности (р-я между 2 точками соседних итераций) стало понятно, что для функции Розенброка нужно меньшее расстояние между точками для достижения такой же абсолютной точности, чем для квадратичной функции (у них разная скорость изменения градиента, для этого и взяли b=100). Дихотомия показала себя лучше, чем градиентный спуск с постоянной скоростью, но самым стабильным оказался метод Нелдера-Мида. Для простой функции в среднем дихотмия справлялась за меньшее число итераций и присвоений чем постоянный спуск, но при работе с функцией Розенброка, несмотря на свою относительную точность и небольшое количество итераций, часто совершала на порядок больше присвоений (вычислений значений).
Файл — *base/tolerance.txt*

Для исследования зависимости точности от начальной точки был выбран достаточно примитивный подход это — увеличивать или уменьшать значение по 2 осям на одну и ту же величину. Начальные точки эксперимента — сами минимумы функции. Естественно лучшая точность достигалась для них, и чем дальше мы от них уходили, тем в среднем был менее правдивым результат вычислений. Абсолютная точность для квадратичной функции росла быстрее с приближением к минимуму, чем точность для функции Розенброка. Однако нужно учитывать, что для этого эксперимента была понижена точность (для точек), а при большой tolerance для получения достаточно точного с точки зрения математики результата достаточно выбрать любую точку, расчеты с которой не приведут к вычислительной ошибки Python. Метод Нельдера Мида выдавал стабльно точный результат вне зависимости от начальной точки.
Файл — *base/start_points.txt*.

И наконец, для того, чтобы понять эффективность выбранных подходов в общем, был проведен финальный замер — исследование зависимости количества итераций и вычислений от требуемой абсолютной точности (разница между математически вычисленными и программными значениями). Точные значения для данных функций это (1, 1) и (2, 3) соответственно. Результаты замера вполне ожидаемы: градиентный спуск с постоянным шагом не смог справиться хорошо на вычислениях для квадратичной функции (теперь ведь точность мы определяем по другому), выполнение стопорилось на предельном количестве итераций, дихотомия сработала для обеих функций(хоть и менее точно чем gd для квадратичной), но на вычислениях для функции Розенброка произвела больше вычислений и итераций. Уменьшения шага на 2 порядка (изначальный — 0.001) в градиенте с постоянным спуском решило проблему с квадратичной функцией, это связано с ее конфигурацией. Однако программа стала вылетать слишком часто(overflow). Количество итераций для м. Нелдера-Мида по понятным причинам не было вычислено.
Файл — *base/precision.txt*.

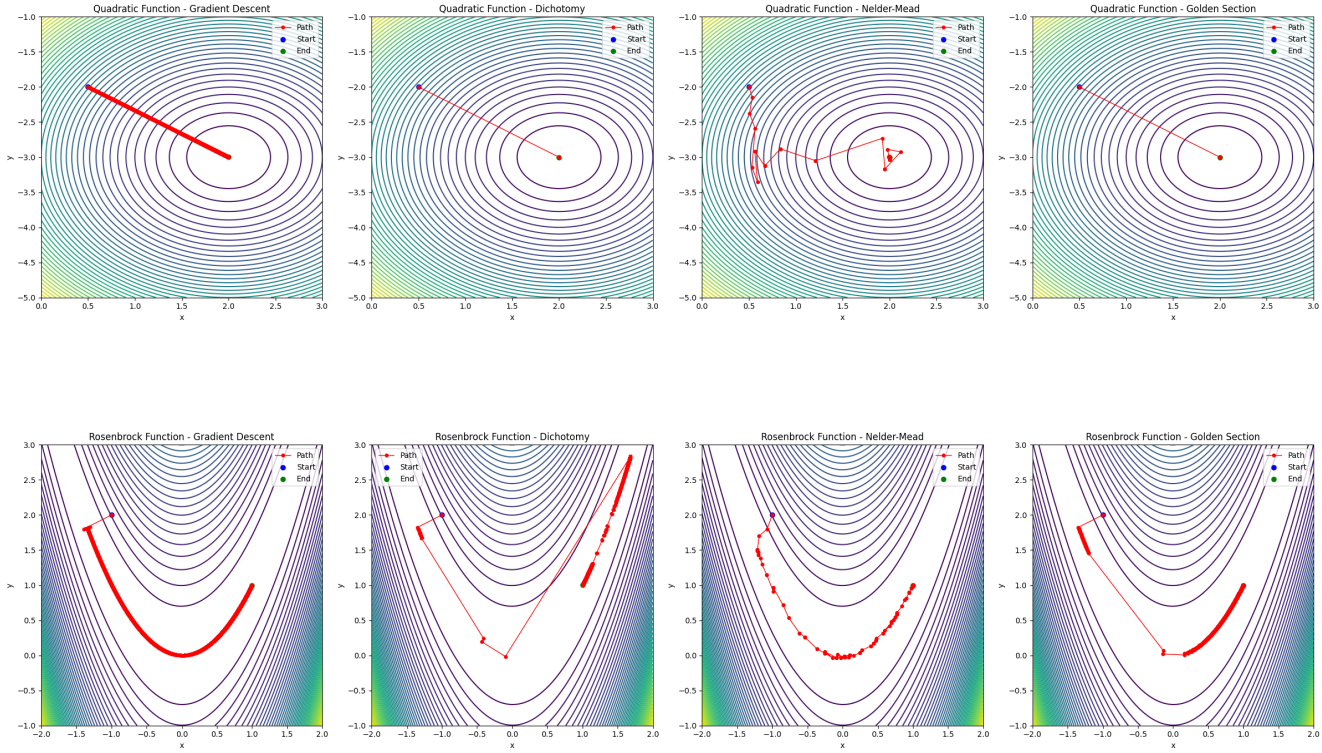
**К содержанию**

**Сравнительная таблица**

| Required Precision | Problem | Method | Iterations | Evaluations | Value 1 | Value 2 |
|---|---|---|---|---|---|---|
| 0.001 | Rosenbrock | GD | 16,310 | 16,310 | 0.9995533998195479 | 0.999105211496215 |
| 0.001 | Rosenbrock | Dichotomy | 5,923 | 177,690 | 0.9995526917178346 | 0.9991050690077696 |
| 0.001 | Rosenbrock | NM | - | - | 1.0001075060812563 | 1.0002075911897421 |
| 0.001 | Quadratic | GD | 1,000,000 | 1,000,000 | 1.9999999999999445 | -2.999999999999889 |
| 0.001 | Quadratic | Dichotomy | 2 | 60 | 2.001953125 | -3.0029296875 |
| 0.001 | Quadratic | NM | - | - | 1.9998623431763312 | -3.0003861299233128 |
| 0.0001 | Rosenbrock | GD | 22,073 | 22,073 | 0.9999553414450914 | 0.9999105061741808 |
| 0.0001 | Rosenbrock | Dichotomy | 9,689 | 406,938 | 0.9999552954709897 | 0.999910510644238 |
| 0.0001 | Rosenbrock | NM | - | - | 1.0000043858986165 | 1.0000106409916478 |
| 0.0001 | Quadratic | GD | 1,000,000 | 1,000,000 | 1.9999999999999445 | -2.999999999999889 |
| 0.0001 | Quadratic | Dichotomy | 2 | 84 | 2.0001220703125 | -3.00018310546875 |
| 0.0001 | Quadratic | NM | - | - | 2.000015209581492 | -2.9999794068082535 |
| 1e-05 | Rosenbrock | GD | 27,837 | 27,837 | 0.9999955334940125 | 0.9999910491347648 |
| 1e-05 | Rosenbrock | Dichotomy | 6,934 | 353,634 | 0.9999955319619693 | 0.9999910408829235 |
| 1e-05 | Rosenbrock | NM | - | - | 1.000000826301518 | 1.000001498936603 |
| 1e-05 | Quadratic | GD | 1,000,000 | 1,000,000 | 1.9999999999999445 | -2.999999999999889 |
| 1e-05 | Quadratic | Dichotomy | 2 | 102 | 1.9999847412109375 | -2.9999771118164062 |
| 1e-05 | Quadratic | NM | - | - | 1.999997219336365 | -3.000000539603813 |
| 1e-06 | Rosenbrock | GD | 33,602 | 33,602 | 0.9999995534369374 | 0.9999991050871078 |
| 1e-06 | Rosenbrock | Dichotomy | 9,402 | 564,120 | 0.9999995528365155 | 0.9999991034846515 |
| 1e-06 | Rosenbrock | NM | - | - | 1.0000000659152914 | 1.0000001145179755 |
| 1e-06 | Quadratic | GD | 1,000,000 | 1,000,000 | 1.9999999999999445 | -2.999999999999889 |
| 1e-06 | Quadratic | Dichotomy | 2 | 120 | 2.000001907348633 | -3.000002861022949 |
| 1e-06 | Quadratic | NM | - | - | 2.0000001353600485 | -3.0000001161410506 |

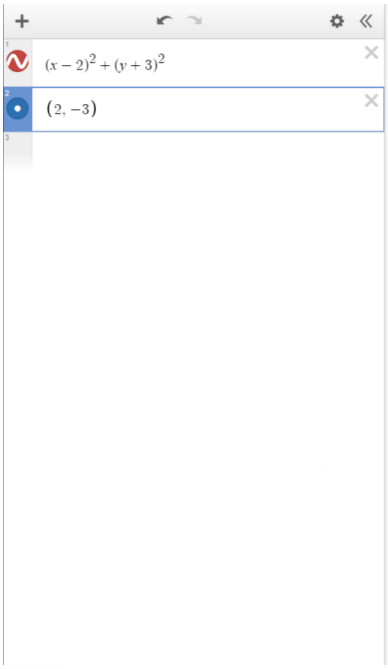
К содержанию

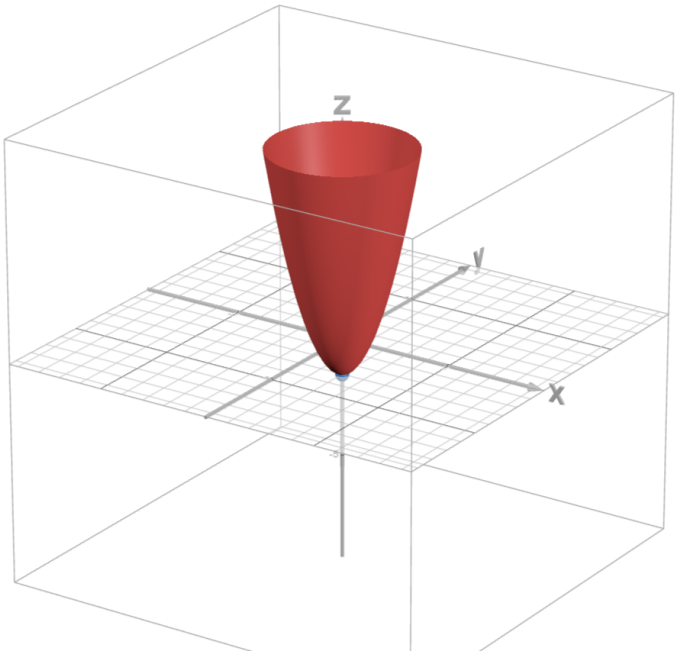
**Траектории движений (+ для Метода Золотого сечения из Доп. 1)**

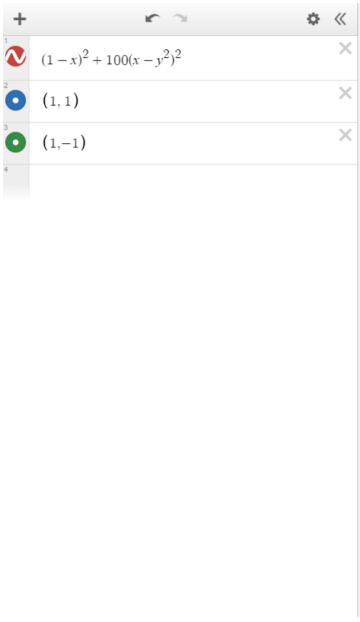


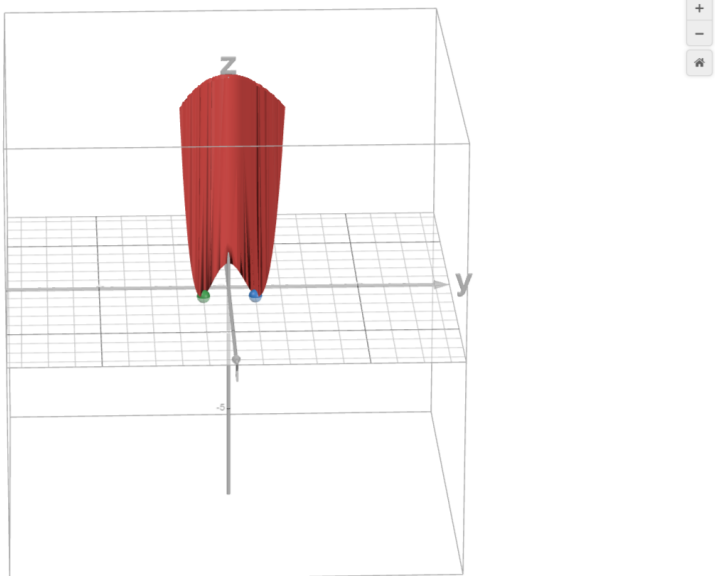

К содержанию

**Графики выбранных функций**




К содержанию

# 2 Дополнительное задание 1 (на выбор один из пунктов)

## 2.1 Ещй один метод одномерного поиска и градиентный спуск на его основе. (Метод Золотого сечения)

$$[a, b]$$

$$\text{golden\_ratio} = \frac{\sqrt{5}-1}{2}$$

$$c = b - \text{golden\_ratio} \cdot (b-a) \qquad d = a + \text{golden\_ratio} \cdot (b-a)$$

$$\begin{cases} a = c & \text{if } f((x,y) - c \cdot \bigtriangledown f(x,y)) \geq f((x,y) - d \cdot \bigtriangledown f(x,y)) \\ b = d & \text{else} \end{cases}$$

$$if\ \ (|c-d| < \varepsilon)\ \ \alpha = \frac{a+b}{2};\ break;$$

## 2.2 Вывод

Внедрение метода золотого сечения в стратегию градиентного спуска предоставляет значительные преимущества для поиска оптимального размера шага, ускоряя и повышая точность сходимости к минимуму функции потерь. Это особенно критично при работе с задачами оптимизации, где функции потерь обладают сложными ландшафтами. Неправильный выбор размера шага может замедлить или даже предотвратить достижение сходимости, в то время как метод золотого сечения минимизирует такой риск, оптимизируя процесс поиска без необходимости многократных вычислений функции, как это происходит в методе дихотомии.

Метод золотого сечения основан на принципе разделения отрезка на части в золотом соотношении, где пропорции между частями отрезка обладают уникальными свойствами, способствующими эффективности и точности поиска. Эта техника позволяет более точно и экономно приближаться к экстремуму функции, уменьшая объем вычислений и повышая эффективность градиентного спуска.

Кроме того, применение метода золотого сечения вносит уникальные преимущества в оптимизацию и поиск экстремумов, делая его неоценимым инструментом в сложных ситуациях, где требуется высокая точность и эффективность без прямой зависимости от градиентов функции. Это расширяет возможности градиентного спуска, делая комбинированный подход мощным решением для специалистов в области оптимизации и анализа данных, работающих с разнообразными и сложными оптимизационными задачами.

Таким образом, синтезируя метод золотого сечения с градиентным спуском, мы получаем стратегию, которая не только усиливает эффективность и точность нахождения глобального минимума, но и предоставляет гибкий и мощный инструмент для преодоления вызовов современных оптимизационных задач. Эта синергия подходов открывает новые горизонты в оптимизации, обеспечивая превосходную производительность и результаты.

К содержанию

# 3 Дополнительное задание 2

## 3.1 Исследуйте эффективность методов на функциях n переменных, в зависимости от размерности пространства n

Метод Нелдера-Мида обычно требует меньше всего итераций в сравнении с другими методами, особенно при малом числе итераций. Однако, по мере увеличения размерности пространства, количество итераций метода Нелдера-Мида растёт значительно быстрее, чем у других методов. Это происходит так как перемещение и деформирование симплекса очень быстро становится затратным на многомерных пространствах.

Метод одномерного поиска и градиентного спуска имеет промежуточное значение итераций, которое растет не так быстро, как у метода Нелдера-Мида, с увеличением размерности пространства. Исходя из этих данных, можно сделать вывод, что метод Нелдера-Мида может быть более выгодным для функций с меньшей размерностью, тогда как метод одномерного поиска дихотомией и градиентного спуска может быть более эффективным при большой размерности пространства.

## 3.2 Исследуйте эффективность методов на плохо обусловленных функциях двух переменных

Функция Гольдштейна – Прайса сложна для оптимизации из-за её нерегулярности и большого количества локальных минимумов.В связи с особенностями этой функции метод градиентного спуска вынужден на каждом шаге "проверять"каждый из этих минимумов, что приводит к большому количеству итераций. Дихотомия производила расчеты еще менее оптимально, чего нельзя сказать про метод Нелдера-Мида, он устроен так, что количество экстремум. Функция Растригина это еще одна сложная функция, имеющая большое количество локальных минимумов. Метода Нелдера-Мида показал себя чуть хуже, чем дихотомия, но намного лучше чем постоянный градиентный спуск.

Отсюда делаем вывод, что предпочтительным методом для плохообусловленных функций является алгоритм Нелдера-Мида из за его универсальности.

## 3.3 Исследуйте эффективность методов на функциях с зашумленными значениями и на мультимодальных функциях

Для шума используем:

```
random.normalvariate(0, 0.05)
```

На зашумленных функциях хуже всех себя показал метод градиентного спуска постоянной скорости (если не хардкодить learning rate). Дихотомия показала результаты лучше. Как и ожидалось, самым точным был метод Нелдера-Мида. Однако, нельзя не отметить, что для квадратичной функции отклонения траекторий методов было почти нулевым, когда для функции Розенброка траектория существенно отклонилась, хоть результаты и были близки к исходным.

К содержанию

## 3.4 Вывод

Исследование эффективности различных методов оптимизации в разнообразных сценариях выявляет важные инсайты, которые могут направлять выбор подхода в зависимости от конкретных условий задачи.

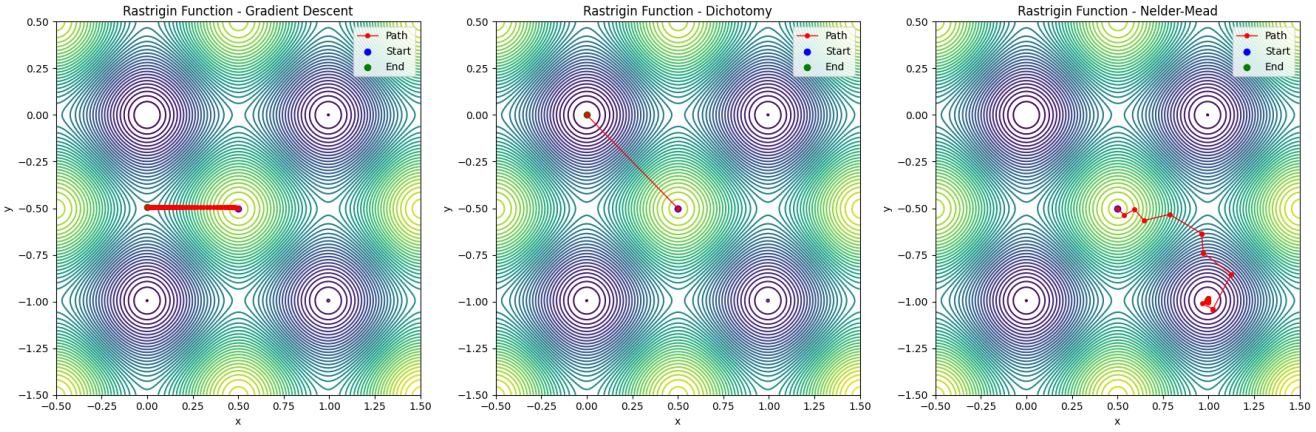
Во-первых, важно подчеркнуть, что нет универсального метода оптимизации, который был бы оптимальным для всех возможных задач. Выбор метода должен учитывать специфику задачи, включая размерность пространства, характеристики функции (например, наличие шума, мультимодальность, обусловленность) и вычислительные ограничения.

Метод Нелдера-Мида, благодаря своей гибкости и адаптивности, показывает выдающиеся результаты в условиях средней и низкой размерности пространства, а также при работе с плохо обусловленными и зашумлёнными функциями. Однако его эффективность снижается с увеличением размерности пространства из-за высоких вычислительных затрат.

Градиентный спуск и одномерный поиск показывают себя как более устойчивые методы при работе в условиях высокой размерности, хотя и могут быть менее эффективны в ситуациях с шумом или при наличии множества локальных минимумов. Их применение особенно оправдано, когда необходимы более простые и предсказуемые подходы, или когда вычислительные ресурсы ограничены.

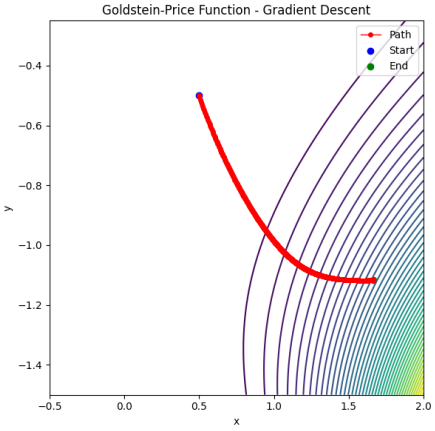
Таким образом, общий вывод подчёркивает значимость тщательного анализа задачи перед выбором метода оптимизации. Он также указывает на необходимость разработки и применения адаптивных алгоритмов, способных изменять свои стратегии в зависимости от текущих условий задачи, чтобы достигать высокой эффективности в широком спектре сценариев. В итоге, успешное решение оптимизационных задач требует глубокого понимания как математических основ методов, так и особенностей конкретной задачи, над которой ведётся работа.

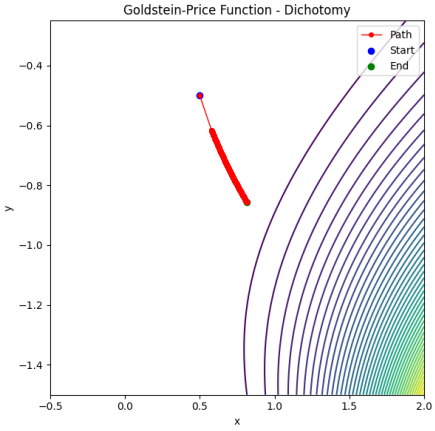
**Плохо обусловоенная функция (Растригина)**

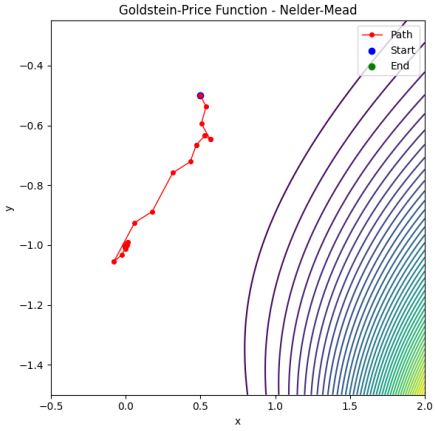



К содержанию

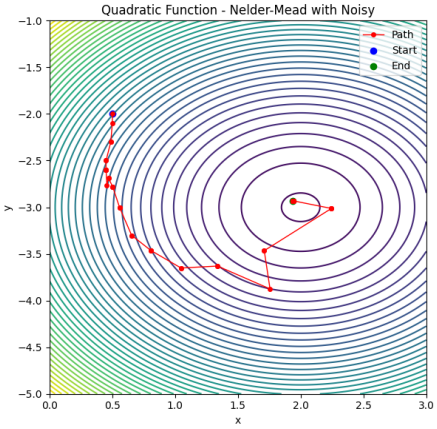
## Плохо обусловоенная функция (Гольдштейна)

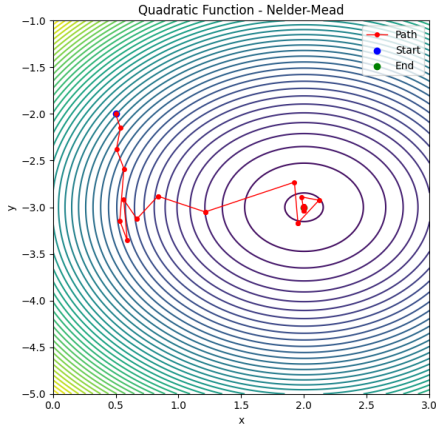



## Добавление шума




К содержанию