Лабораторная работа №1

Мицеловский Артемий

Биктимиров Тимур

Коробов Иван

1.**Основное задание**

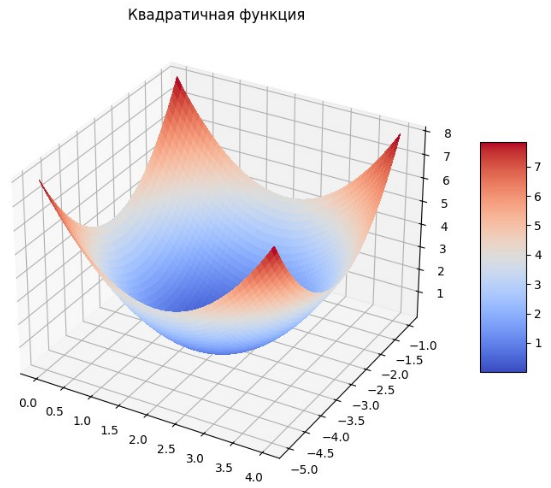
Для исследования были выбраны 2 функции — квадратичная с коэффициентами 2 и 3 (то есть (x — 2)^2 + (y + 3)^2), и функция Розенброка с коэффициентами 1 и 100: (1 — x)^2 + 100(x — y^2)^2. Большой коэффициент был выбран для наглядности.

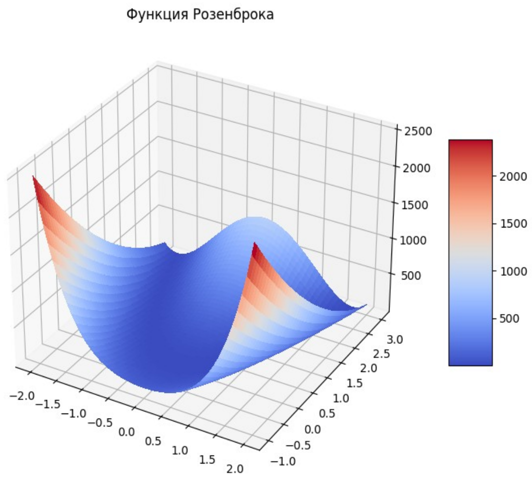
При исследовании зависимости количества итераций и вычислений от требуемой точности (р-я между 2 точками соседних итераций) стало понятно, что для функции Розенброка нужно меньшее расстояние между точками для достижения такой же абсолютной точности, чем для квадратичной функции (у них разная скорость 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

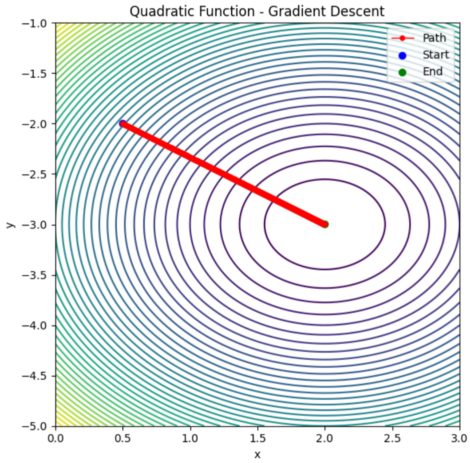
Для исследования зависимости точности от начальной точки был выбран достаточно примитивный подход — увеличивать или уменьшать значение по 2 осям на одну и ту же величину. Начальные точки эксперимента — сами минимумы функции. Естественно лучшая точность достигалась для них, и чем дальше мы от них уходили, тем в среднем был менее правдивым результат вычислений. Абсолютная точность для квадратичной функции росла быстрее с приближением

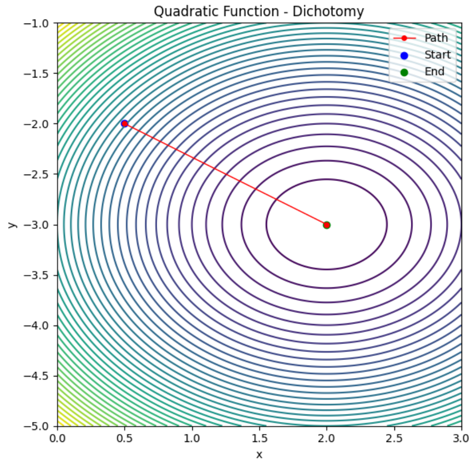
к минимуму, чем точность для функции Розенброка. Однако нужно учитывать, что для этого эксперимента была понижена точность (для точек), а при большой tolerance для получения достаточно точного с точки зрения математики результата достаточно выбрать любую точку, расчеты с которой не приведут к вычислительной 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. Ниже — сравнительная таблица.
ВАНЯ, СДЕЛАЙ ТАБЛИЦУ ПЖ!!!

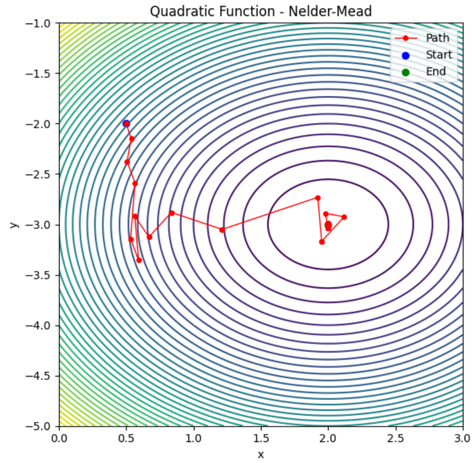
Графики функций и графики, иллюстрирующие работу методов:

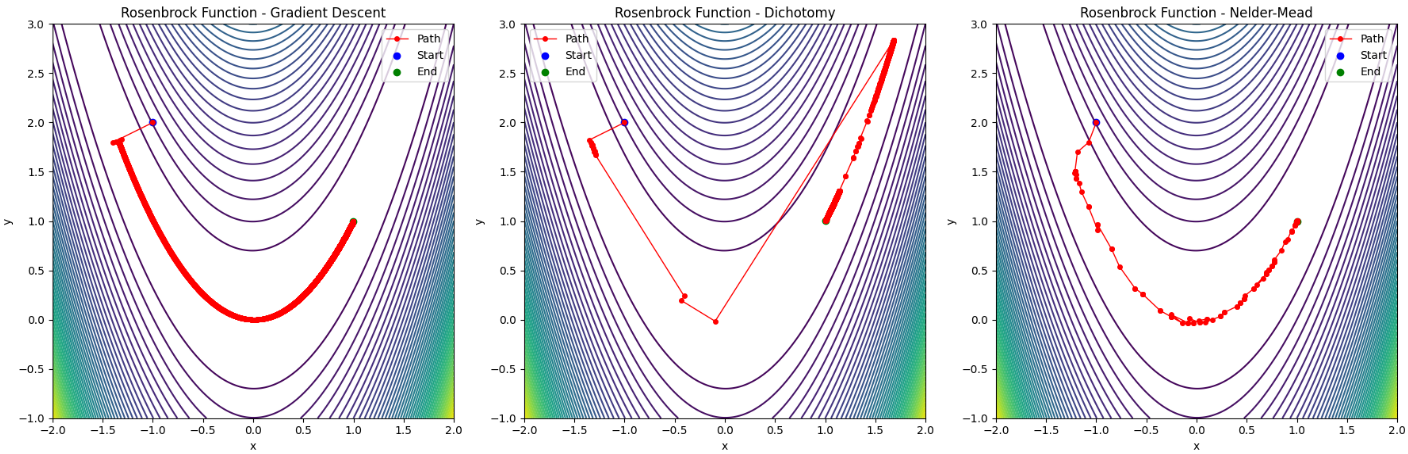

Квадратичная функция


Функция Розенброка


Quadratic Function - Gradient Descent
Path
Start
End
x
y


Quadratic Function - Dichotomy
Path
Start
End
x
y


Quadratic Function - Nelder-Mead
Path
Start
End
x
y



Общий вывод: в зависимости от конфигурации функции следует по разному рассчитывать точность и применять разные численные методы, анализируя их. Метод дихотомии точнее и эффективнее метода градиаентного спуска с постоянным шагом. Метод Нилдера-Мида в общем случае превосходит их обоих в точности и быстродействии.