

**Министерство науки и высшего образования Российской
Федерации**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

«Национальный исследовательский университет ИТМО»

Факультет информационных технологий и программирования

Прикладная математика

Лабораторная работа №1

Выполнили студенты группы № М32091

Фисенко Никита Данилович

Рустамов Марк Самирович

Санкт-Петербург 2023

Постановка задачи:

Обычному инженеру с дозиметром, оказавшемуся окруженным со всех сторон радиацией, необходимо понять, куда двигаться ему дальше, чтобы поскорее убраться от источника излучения. В его голове возникла модельная функция величины радиации от угла его первоначального движения: $y(x) = \sin(x) \cdot x^2$

Для решения задачи будут использованы следующие методы: дихотомии, золотого сечения, метод Фибоначи, метод парабол, комбинированный метод Брента.

Цели работы:

Реализовать методы дихотомии, золотого сечения, метод Фибоначи, метод парабол, комбинированный метод Брента на языке программирования Python.

Применить данные методы для определения оптимального направления движения.

Проанализировать результаты и сделать выводы о наиболее эффективном методе.

Теория:

Рассмотрим задачу одномерной оптимизации

$$\min_{x \in [a, b]} \{z = f(x)\} \quad (1)$$

Решить задачу - это значит найти точку $x^* \in [a, b]$, в которой достигается оптимальное значение. Делается предположение, что такая точка существует. Однако, часто бывает, что мы не можем решить задачу точно, и приходится использовать приближенное решение, которое близко к оптимальному.

Введем необходимое определение.

Функцию $f(x)$ называют унимодальной функцией на отрезке $[a, b]$, если существует такая точка $x_ \in [a, b]$, что функция $f(x)$ в полуинтервале $[a, x_*)$ убывает, а в полуинтервале $(x_*, b]$ возрастает.*

Пусть функция $f(x)$ унимодальна на отрезке $[a, b]$. Необходимо найти точку минимума функции на этом отрезке с заданной точностью ε . Однако довольно все методы одномерного поиска базируются на последовательном уменьшении интервала, содержащего точку минимума.

Возьмем внутри отрезка $[a_0, b_0]$ две точки x_1 и x_2 : $a_0 < x_1 < x_2 < b_0$, и вычислим значения функции в этих точках. Из свойства унимодальности функции можно сделать вывод о том, что минимум расположен либо на отрезке $[a_0, x_2]$, либо на отрезке $[x_1, b_0]$. Действительно, если $f(x_1) < f(x_2)$, то минимум не может находиться на отрезке $[x_2, b_0]$, а если $f(x_1) > f(x_2)$, то минимум не может находиться на отрезке $[a_0, x_2]$. Если же $f(x_1) = f(x_2)$, то минимум находится на интервале $[x_1, x_2]$.

Алгоритм заканчивается, когда длина интервала, содержащего минимум, становится меньше ε . Различные методы одномерного поиска отличаются выбором точек x_1 , x_2 . Об эффективности алгоритмов можно судить по числу вычислений функции, необходимому для достижения заданной точности.

Примечание:

Сравнение и реализация методов:

<https://github.com/russianZAK/applied-mathematics/blob/main/Lab%201/lab1.ipynb>

Тесты полиномов:

<https://github.com/russianZAK/applied-mathematics/tree/main/Lab%201/tests>

Теория методов:

1. Метод дихотомии

Точки x_1, x_2 выбираются на расстоянии $\delta < \varepsilon/2$ от середины отрезка:

$$x_1 = (a_i + b_i)/2 - \delta, \quad (2)$$

$$x_2 = (a_i + b_i)/2 + \delta \quad (3)$$

За одну итерацию интервал неопределенности уменьшается примерно в 2 раза. Значит, за n итераций длина интервала будет примерно равна $(b_0 - a_0)/2^n$. Для достижения точности ε потребуется приблизительно $\ln((b_0 - a_0)/\varepsilon)/\ln 2$ итераций. На каждой итерации функция вычисляется два раза.

2. Метод золотого сечения

Точки x_1, x_2 находятся симметрично относительно середины отрезка $[a_0, b_0]$ и делят его в пропорции золотого сечения, когда длина всего отрезка относится к длине большей его части также, как длина большей части относится к длине меньшей части:

$$\frac{b_0 - a_0}{b_0 - x_1} = \frac{b_0 - x_1}{x_1 - a_0} \quad (4)$$

Отсюда

$$x_1 = a_i + \frac{\sqrt{5} - 1}{2}(b_i - a_i) = a_i + 0.381966011 \cdot (b_i - a_i) \quad (5)$$

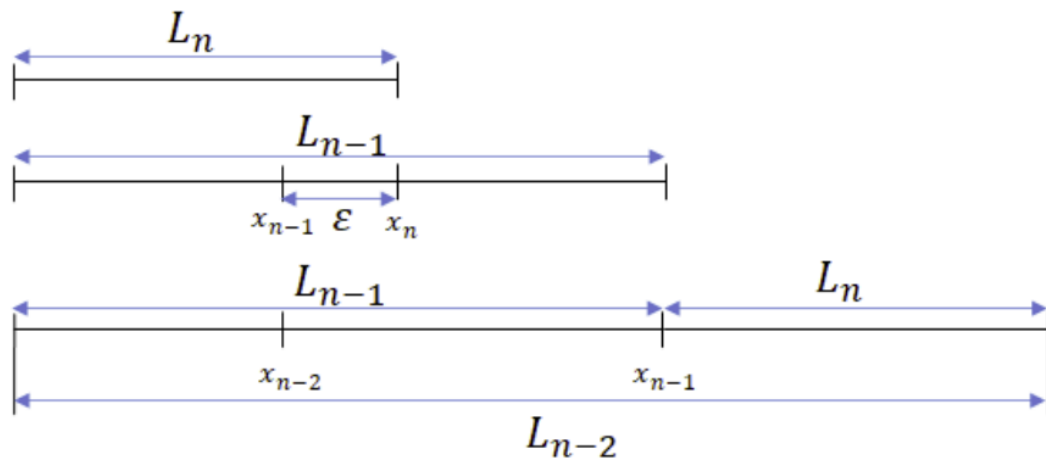
Аналогично для второй точки.

За одну итерацию интервал неопределенности уменьшается в $\frac{\sqrt{5}+1}{2} = 1.618..$ раз, однако на следующей итерации мы будем вычислять функцию только один раз, так как по свойству золотого сечения $\frac{x_2 - x_1}{b - x_1} = 0.381..$ и $\frac{b - x_2}{b - x_1} = 0.618...$ Для достижения точности ε потребуется $n \geq \frac{\ln((b_0 - a_0)/\varepsilon)}{\ln(\frac{\sqrt{5}+1}{2})}$.

3. Метод Фиббоначи

Это улучшение реализации поиска с помощью золотого сечения, служащего для нахождения минимума/максимума функции. Подобно методу золотого сечения, он требует двух вычислений функции на первой итерации, а на каждой последующей только по одному. Однако этот метод отличается от метода золотого сечения тем, что коэффициент сокращения интервала неопределенности меняется от итерации к итерации.

Предположим, нам нужно определить минимум как можно точнее, т.е. с наименьшим интервалом неопределенности, но при этом можно произвести только n вычислений функции.



Пусть у нас есть интервал неопределенности (x_1, x_3) , и нам известно значение функции $f(x_2)$, $x_2 \in (x_1, x_3)$.

Если можно вычислить функцию всего один раз в точке x_4 , то где следует ее поместить, чтобы получить минимально возможный интервал неопределенности? Заранее нам не известно, как ведет себя функция, и реализуется одна из двух ситуаций:

1. $x_4 \in (x_1, x_2)$
2. $x_4 \in (x_2, x_3)$

Т.е. неизвестно, какая из ситуаций будет иметь место, выберем x_4 таким образом, чтобы минимизировать максимальную из длин

$$\min\{\max\{(x_3 - x_4), (x_2 - x_1)\}\}.$$

Достигнуть этого можно, сделав эти длины равными, т.е.

$$(x_3 - x_4) = (x_2 - x_1).$$

Для этого нужно поместить x_4 внутри интервала (x_1, x_2) симметрично относительно точки x_2 .

Если окажется, что можно выполнить еще одно вычисление функции, то следует применить описанную процедуру к новому интервалу неопределенности.

Стратегия ясна: нужно поместить следующую точку внутри интервала, симметрично относительно уже находящейся там точки.

Остался один вопрос: как начать вычисления? Для ответа на этот вопрос, начнем с конца.

На n -ом вычислении n -ю точку стоит поместить симметрично по отношению к $(n-1)$ -й точке. Чтобы получить наибольшее уменьшение интервала на данном этапе, следует разделить пополам предыдущий интервал.

Обозначим за ε минимальную длину интервала неопределенности. Тогда

$$\begin{aligned} L_{n-1} &= 2L_n - \varepsilon \\ L_{n-2} &= L_{n-1} + L_n \\ L_1 & \end{aligned}$$

Числа Фибоначчи определяются соотношениями:

$$F_{n+1} = F_{n+1} + F_n, \quad n = 1, 2, \dots, \quad F_1 = F_2$$

n -е число Фибоначчи представимо в виде:

$$F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2} \right)^n - \left(\frac{1 - \sqrt{5}}{2} \right)^n \right], n = 1, 2, \dots \quad (6)$$

На начальном интервале вычисляются точки

$$x_1 = a_0 + \frac{F_n}{F_{n+2}}(b_0 - a_0), \quad (7)$$

$$x_2 = a_0 + \frac{F_{n+1}}{F_{n+2}}(b_0 - a_0) \quad (8)$$

где n выбирается исходя из точности и начальной длины интервала.

На k -м шаге метода будет получена тройка чисел a_k, b_k, x_k , локализирующая минимум $f(x)$, такая что

$$\Delta_k = b_k - a_k = (b_0 - a_0) \frac{F_{n-k+3}}{F_{n+2}}, \quad 1 \leq k \leq n, \quad a_1 = a_0, b_1 = b_0,$$

а точка $x_k, a_k < x_k < b_k$, с вычисленным значением

$$f(x_k) = \min_{1 \leq i \leq k} f(x_i)$$

совпадает с одной из точек

$$\begin{aligned} x_1 &= a_k + \frac{F_{n-k+1}}{F_{n-k+3}}(b_k - a_k) = a_k + \frac{F_{n-k+1}}{F_{n+2}}(b_0 - a_0), \\ x_2 &= a_k + \frac{F_{n-k+2}}{F_{n-k+3}}(b_k - a_k) = a_k + \frac{F_{n-k+2}}{F_{n+2}}(b_0 - a_0) \end{aligned}$$

расположенных на отрезке $[a_k, b_k]$ симметрично относительно его середины. При $k = n$ процесс заканчивается. В этом случае длина отрезка

$$\Delta_n = b_n - a_n = (b_0 - a_0)/F_{n+2},$$

а точки

$$\begin{aligned} x_1 &= a_n + \frac{F_1}{F_{n+2}}(b_0 - a_0), \\ x_2 &= a_n + \frac{F_2}{F_{n+2}}(b_0 - a_0) \end{aligned}$$

совпадают и делят отрезок пополам.

Следовательно

$$\frac{b_n - a_n}{2} = \frac{b_0 - a_0}{F_{n+2}} < \varepsilon.$$

Отсюда можно выбрать n из условия

$$\frac{b_0 - a_0}{\varepsilon} < F_{n+2}.$$

Отметим, что в методах золотого сечения и Фиббоначи нет необходимости вычислять значение функции f в крайних точках начального интервала. Это удобно, если оптимизируемая функция, например, имеет вертикальную асимптоту.

4. Метод парабол

В методе парабол предлагается аппроксимировать оптимизируемую функцию $f(x)$ с помощью квадратичной функции

$$p(x) = ax^2 + bx + c.$$

Для того, чтобы найти коэффициенты аппроксимируемой параболы a, b, c необходимо решить систему линейных уравнений

$$ax_i^2 + bx_i + c = f_i = f(x_i), \quad i = 1, 2, 3.$$

Для того, чтобы получить систему, используем три точки: $x_1 < x_2 < x_3$, $x_{min} \in [x_1, x_3]$. Решив эту систему, получим, что минимум такой параболы равен

$$u = -\frac{b}{2a} = x_2 - \frac{(x_2 - x_1)^2(f_2 - f_3) - (x_2 - x_3)^2(f_2 - f_1)}{2[(x_2 - x_1)(f_2 - f_3) - (x_2 - x_3)(f_2 - f_1)]}$$

Если $f_2 < f_1$ и $f_2 < f_3$, то точка m гарантированно попадает в интервал $[x_1, x_3]$. Таким образом, внутри интервала у нас определены две точки x_2 и u , с помощью сравнения значений функции f в которых можно сокращать интервалы поиска.

В отличие от перечисленных выше методов, метод парабол обладает суперлинейной скоростью сходимости. Однако, такая высокая скорость сходимости гарантируется только в малой окрестности точки минимума x_{min} . Однако, если начальное приближение не попадает в окрестность точки минимума, то сходимость никто не обещает. Также следует отметить, что на первой итерации метод парабол требует измерения значений функции f в крайних точках интервала оптимизации.

5. Комбинированный метод Брента

Метод золотого сечения представляет собой надежный способ оптимизации, который сходится за гарантированное число итераций, но обладает лишь линейной скоростью сходимости. Метод парабол работает быстрее в малой окрестности оптимального решения, но может работать долго и неустойчиво на начальных стадиях итерационного процесса. Поэтому на практике для решения задачи одномерной оптимизации используется метод Брента, который эффективно комбинирует эти две стратегии. В данном методе на каждой итерации отслеживаются значения в шести точках (не обязательно различных): a, c, x, w, v, u . Точки a, c задают текущий интервал поиска решения, x — точка, соответствующая наименьшему значению функции, w — точка, соответствующая второму снизу значению функции, v — предыдущее значение w . В отличие от метода парабол, в методе Брента аппроксимирующая парабола строится с помощью трех наилучших точек x, w, v (в случае, если эти три точки различны и значения в них также различны). При этом минимум аппроксимирующей параболы u принимается в качестве следующей точки оптимизационного процесса, если:

- u попадает внутрь интервала $[a, c]$ и отстоит от границ интервала не менее, чем на ε ;
- u отстоит от точки x не более, чем на половину от длины предыдущего шага.

Если точка u отвергается, то следующая точка находится с помощью золотого сечения большего из интервалов $[a, x]$ и $[x, c]$.

Приведем некоторые аргументы в пользу обозначенных условий приема минимума параболы u . Так как парабола на текущей итерации проводится через точки x, w, v , для которых не гарантируются соотношения $v < x < w$ или $w < x < v$, то минимум параболы может оказаться вне интервала $[a, c]$. Условие удаленности точки u от a и c не менее, чем на некоторый порог ε , позволяет избежать слишком маленьких шагов в оптимизации, которые могут свидетельствовать о локальном застопоривании метода парабол. Аналогично, ограничение на максимальную удаленность u от x позволяет избежать слишком больших шагов в оптимизации, которые могут соответствовать биениям в методе парабол. Использование в данном ограничении длины предыдущего шага, а не предыдущего, является эвристикой, эффективность которой подтверждается в экспериментах на больших базах задач оптимизации. Эта эвристика предлагает не штрафовать метод за текущий не слишком удачный маленький шаг в надежде на успешные шаги метода на следующих итерациях.

В сравнении этих методов по количеству вычислений функции и количеству итераций от разной точности можно выделить следующие особенности:

1. Метод дихотомии является наиболее простым и надежным методом, но он требует большого количества итераций для достижения высокой точности. Он гарантированно сходится к минимуму функции, но скорость сходимости медленная.
2. Метод золотого сечения и метод Фибоначчи сходятся быстрее, чем метод дихотомии, но все еще требуют относительно большого количества итераций для достижения высокой точности. Оба метода имеют одинаковую скорость сходимости.
3. Метод парабол имеет более быструю скорость сходимости, чем методы дихотомии, золотого сечения и Фибоначчи, но он может страдать от неустойчивости на плоских участках функции. Этот метод также может быть менее эффективным на функциях, которые не являются параболami.
4. Комбинированный метод Брента является наиболее эффективным методом оптимизации одномерных функций. Он использует идеи методов золотого сечения, парабол и дихотомии, чтобы быстро сходиться к минимуму функции. Этот метод обладает высокой скоростью сходимости, надежностью и эффективностью, особенно на сложных функциях.

Таким образом, для достижения высокой точности при оптимизации одномерных функций, наиболее эффективным методом является комбинированный метод Брента. Он обеспечивает быструю сходимость к минимуму функции, используя оптимальную комбинацию методов дихотомии, золотого сечения, Фибоначчи и парабол. Однако, если функция имеет специфические свойства, то использование более простых методов может быть более эффективным.

Все реализованные методы могут не найти локальный минимум многомодальной функции.

Реализованные методы являются методами оптимизации унимодальных функций, если функция является многомодальной, то есть имеет несколько локальных минимумов, то методы могут сойтись к одному из локальных минимумов, но не к глобальному.

Вывод:

В данной лабораторной работе были рассмотрены и применены различные методы оптимизации одномерных функций. Были реализованы методы дихотомии, золотого сечения, метод Фибоначи, метод парабол и комбинированный метод Брента на языке программирования Python.

Результаты анализа показали, что комбинированный метод Брента показал наилучшие результаты с точки зрения скорости сходимости и количества вычислений функции. Однако, если функция будет многомодальной, метод Брента не сможет найти локальный минимум.

Таким образом, наиболее эффективным методом оптимизации одномерной функции в данном случае является комбинированный метод Брента. Он позволяет найти оптимальное направление движения инженера с дозиметром, окруженного радиацией, быстро и с минимальным количеством вычислений функции.