Задание 1

При решении практических задач возникает потребность найти минимум функции. Ответить на этот вопрос легко, если знать, чему равна производная функции. Представим, что ничего, кроме возможности обращения к функции у нас нет. Т.е. имеется некоторый алгоритм, который позволяет при данном x найти $f(x)$, но мы не знаем, какой формулой задана $f(x)$.

Дадим несколько определений.

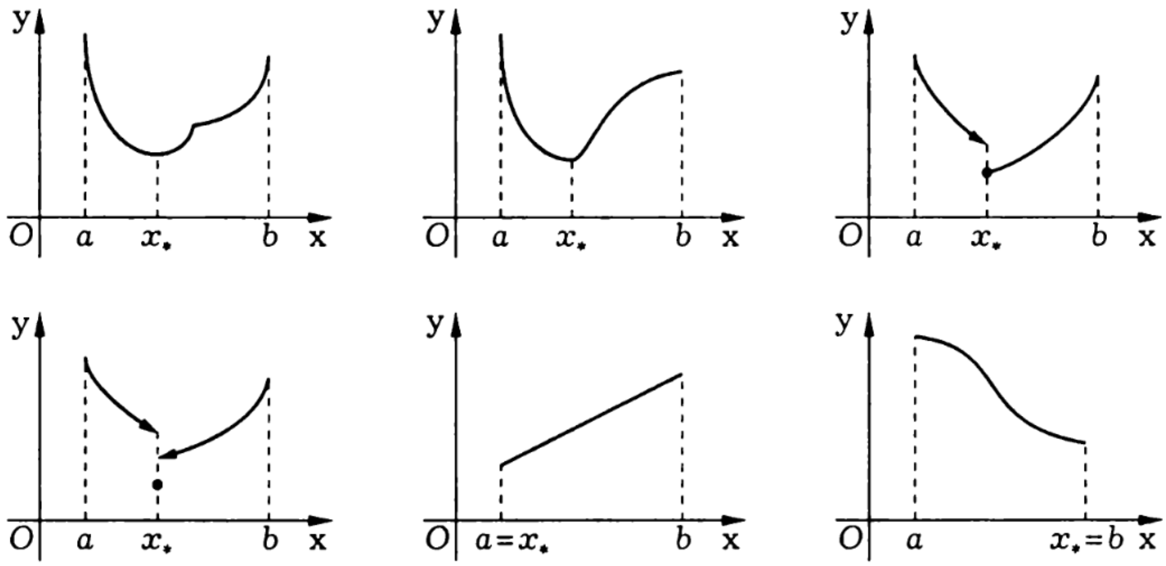
Функция $f(x)$ называется унимодальной на отрезке $[a, b]$, если она непрерывна на $[a, b]$ и существуют числа α, β $(a \leq \alpha \leq \beta \leq b)$ такие, что

1. $f(x)$ строго монотонно убывает на $[a, \alpha]$ (если $a < \alpha$)

2. $f(x)$ строго монотонно возрастает на $[\beta, b]$ (если $\beta < b$)

3. $f(x) = \inf\limits_{x \in [a,b]} f(x)$ на $[\alpha, \beta]$

При этом, вырожденные случаи совпадения одного или нескольких отрезков $[a, \alpha], [\alpha, \beta], [\beta, b]$ не исключаются. При этом, нас будут интересовать строго унимодальные функции (это случай $\alpha = \beta$). Ниже приведены примеры строго унимодальных функций.



Теорема 1 Пусть функция кусочно-непрерывна на отрезке $[a, b]$ и унимодальная на каждом отрезке, где она непрерывна. Тогда функция достигает минимума в точке $x_0 \in [a, b]$. Пусть точки $c < d$ принадлежат $[a, b]$. Если $f(c) \leq f(d)$, то $x_0 \in [a, d]$, иначе $x_0 \in [c, b]$.

Попробуйте доказать теорему самостоятельно. Рассмотрим далее несколько примеров.

Метод дихотомии

Метод относится с последовательным стратегиям. Задается начальный интервал неопределенности и требуемая точность. Алгоритм опирается на анализ функции в двух точках. Для этого от середины текущего интервала откладывается вправо и влево по $\frac{\delta}{2}$. Процесс поиска завершается, когда длина текущего интервала неопределенности меньше установленной точности.

1. Зададим интервал неопределенности $L = [a_0, b_0]$, точность $\varepsilon > 0$

2. Пусть $k = 0$

3. Вычислим $y_k = \frac{a_k + b_k - \varepsilon}{2}$, $z_k = \frac{a_k + b_k + \varepsilon}{2}$, $f(y_k)$, $f(z_k)$

4. Сравним $f(y_k)$ и $f(z_k)$:
   Если $f(y_k) <= f(z_k)$, тогда $a_{k+1} = a_k$, $b_{k+1} = z_k$
   Иначе - $a_{k+1} = y_k$, $b_{k+1} = b_k$

5. Вычислим $|L_{2(k+1)}|$ и проверим условие окончания:
   Если $|L_{2(k+1)}| \leq \varepsilon$, тогда $x^* \approx \frac{a_k + b_k}{2}$, алгоритм завершаем.
   Иначе - $k = k + 1$ и переходим к шагу 2.

Метод золотого сечения*

Метод относится с последовательным стратегиям. Задается начальный интервал неопределенности и требуемая точность. Алгоритм опирается на анализ функции в двух точках. В качестве точек выбираются точки золотого сечения. Процесс поиска завершается, когда длина текущего интервала неопределенности меньше установленной точности.

1. Зададим интервал неопределенности $L = [a_0, b_0]$, точность $\varepsilon > 0$
2. Пусть $k = 0$
3. Вычислим $y_0 = a_0 + \frac{3-\sqrt{5}}{2}(b_0 - a_0)$, $z_0 = a_0 + b_0 - y_0$
4. Вычислим $f(y_k)$, $f(z_k)$
5. Сравним $f(y_k)$ и $f(z_k)$:
   Если $f(y_k) <= f(z_k)$, тогда: $a_{k+1} = a_k$, $b_{k+1} = z_k$, $y_{k+1} = a_k + b_k - y_k$, $z_{k+1} = y_k$
   Иначе: $a_{k+1} = y_k$, $b_{k+1} = b_k$, $y_{k+1} = z_k$, $z_{k+1} = a_k + b_k - z_k$
6. Вычислим $\delta = a_{k+1} - b_{k+1}$ и проверим условие окончания:
   Если $|\delta| \leq \varepsilon$, тогда $x^* \approx \frac{a_{k+1}+b_{k+1}}{2}$, алгоритм завершаем.
   Иначе - $k = k + 1$ и переходим к шагу 4.

Аналитический этап

1. Придумайте функцию "средней"сложности, чтобы, изменив промежуток, можно было проверить все граничные условия и корректность метода.
2. Определите промежуток, на котором функция будет унимодальная.
3. Найти на этом отрезке точки локального экстремума и установить их тип.
4. Найти наибольшее/наименьшее значения на этом отрезке.
5. Выбрать один из алгоритмов (на выбор).
6. Подумать, почему данный алгоритм сходится при заданных условиях. (т.е. не будет такой ситуации, что метод будет работать бесконечно при заданной погрешности).
7. Определить, какое число итераций необходимо выполнить алгоритму при заданной погрешности ответа.
8. Повторите исследование на разных промежутках, в том числе на тех, где не выполнено условие, что функция унимодальна.

Дополнительные задания для алгоритма золотого сечения

1. Доказать, что выбор точек в алгоритме действительно образует золотое сечение промежутка неопределенности на каждой итерации.
2. Найти погрешность метода золотого сечения.

Практический этап

1. Реализовать алгоритм.
2. Добавить на графике положение точки предполагаемого экстремума на каждой итерации алгоритма.
3. Построить график изменения длины промежутка неопределенности с возрастанием числа итераций.

Задание 2

Пусть задана непрерывная функция $f(x)$ на отрезке $[a, b]$. Реализуйте алгоритм, который решает уравнение $f(x) = 0$.

Аналитический этап

1. Возьмите две произвольные функции (желательно брать такие, у которых можно доказать существование корней)
2. Найти, если это возможно, корни этого уравнения. Или докажите, что корень существует.

Практический этап

1. Реализуйте алгоритм нахождения корня для функции на данном отрезке $[a, b]$. Для начала используйте знание о том, что корень всегда есть.
2. Вспомните, какая теорема доказывает существование корня функции на отрезке.
3. Определите критерий остановки алгоритма.
4. Изучите, как при разных промежутках и критериях остановки меняется число итераций, которое делает ваш алгоритм.
5. Найдите среднеквадратичное отклонение численно найденного ответа от истинного на данном промежутке. (формула для СКО уже встроена в питон в библиотеке numpy)
6. Модифицируйте программу таким образом, чтобы в случае отсутствия корня ее поведение соответствовало ожиданиям.

Рекомендованная литература (доступ в электронную библиотеку есть тут https://lib.itmo.ru/resources)

1. «Васильев Ф.П.Методы оптимизации. Кн.1» (Васильев, Ф. П. Методы оптимизации : учебное пособие / Ф. П. Васильев. — Москва : МЦНМО, [б. г.]. — Книга 1 — 2011. — ISBN 978-5-94057-707-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/9304 (дата обращения: 16.12.2024). — Режим доступа: для авториз. пользователей. — С. 29.).
2. https://numpy.org/doc/2.1/reference/generated/numpy.std.html