Санкт-Петербургский
Политехнический университет Петра Великого

Отчет по лабораторной работе №3

Решение задачи одномерной минимизации.

Студенты: Ерошкин Иван Игоревич
Губриенко Денис Дмитриевич
Чибышев Тимофей Андреевич

Преподаватель: Родионова Елена Александровна

Группа: 5030102/00401

Санкт-Петербург
2023

Содержание

1 Постановка задачи

$\sqsupset S = [a, b]$, где $a, b \in \mathbf{R} : a < b$
$\sqsupset f(x)$ определена на S :
• во-первых, $\exists! x_*-$ точка глобального минимума f на S;
• во-вторых, на $[a, x_*]$ f убывает, на $[x_*, b]$ возрастает.
$\sqsupset \epsilon \in \mathbf{R} : 0 < \epsilon < b - a$
Требуется вычислить $\tilde{x}_* \in S : |\tilde{x}_* - x_*| \leq \epsilon$

2 Методы решения задачи линейной минимизации

Мы рассмотрим три метода решения задачи линейной минимизации(метод Фибоначчи,равномерного поиска и пробных точек). Также мы сравним методы по числу обращений к вычислению функции $f(x)$, требуемому для достижения заданной точности.

2.1 Метод равномерного поиска

Описание. Задаются начальный интервал неопределенности $[a; b]$ и количество отрезков разбиения N. Вычисления производятся в $N-1$ равномерно удаленных друг от друга точках (при этом количество вызовов функции равно $N-1$). Путем сравнения величин $f(x_i), i = 1, ..., N-1$ находится точка x_j, в которой значение функции наименьшее. Искомая точка минимума x_* считается заключенной в интервале $[x_{j-1}, x_{j+1}]$ (рис. 2.1).

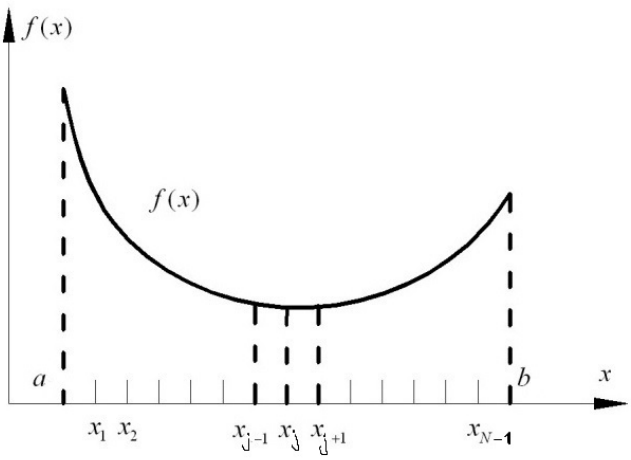


Рис. 2.1. Графическая иллюстрация метода равномерного поиска

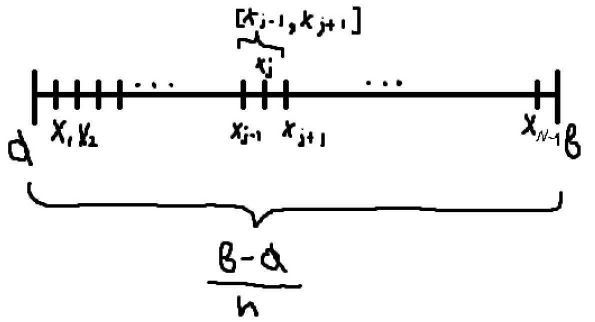
Алгоритм.
1. Зададим точность ϵ. Выберем число n и построим точки:
   $x_i = a + i \cdot h$, где $i = 1, \ldots, n-1$
2. Вычислим значение функции в этих точках и найдем минимальное из них. Пусть оно соответствует номеру j.
3. $f(x)$ - унимодальная функция, следовательно,
   $x^* \in [x_{j-1}, b]$ и $x^* \in [a, x_{j+1}] \implies x^* \in [x_{j-1}, x_{j+1}]$
   Тогда новый интервал неопределенности - $[x_{j-1}, x_{j+1}]$

Будем повторять шаги 1) – 3) до тех пор, пока $|x_{j+1} - x_{j-1}| \geq \epsilon$
В качестве решения можно выбрать любую точку конечного интервала неопределенности. Например, середину.

Замечания. «Давайте подумаем, что выгоднее, делить отрезок на 5 или на 50 частей?»

2.1.1 Теоретические выкладки для оценки количества вызовов функции

Рассмотрим каждую итерацию работы нашего алгоритма.
<u>1-шаг:</u>
Поделим отрезок на n частей. Заметим, что $n > 2$, ибо в противном получим бесконечный цикл. Вычислим длину интервала неопределенности: $|x_{j+1}^{(1)} - x_{j-1}^{(1)}| = |x_{j+1}^{(1)} - x_j^{(1)}| + |x_j^{(1)} - x_{j-1}^{(1)}| = \frac{2}{n}(b - a)$



Подробнее на рисунке.

2-шаг:

Вычислим длину интервала неопределенности на втором шаге:
$|x^{(2)}_{j+1} - x^{(2)}_{j-1}| = \frac{2^2}{n^2}(b-a)$

k-шаг:

Вычислим длину интервала неопределенности на k-ом шаге:
$|x^{(k)}_{j+1} - x^{(k)}_{j-1}| = \frac{2^k}{n^k}(b-a)$

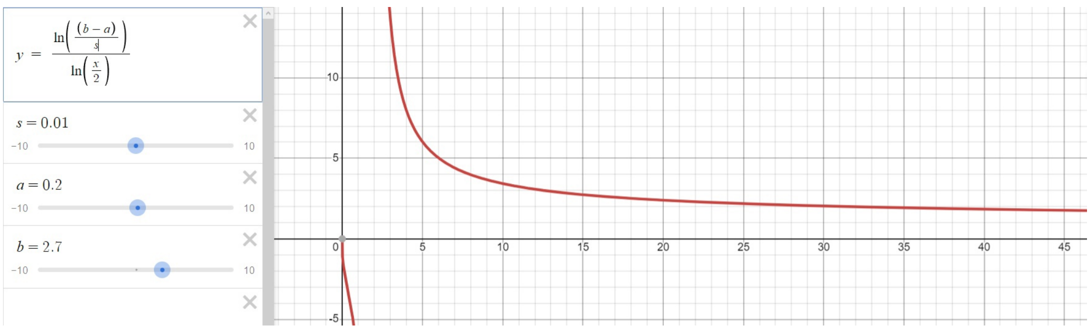
Предположим, что k-й шаг был последним. Значит, выполнено условие отсановки, то есть $|x^{(k)}_{j+1} - x^{(k)}_{j-1}| < \epsilon$, где k - шаг, то есть в этом случае количество итераций, необходимое для достижения заданной точности.
Произведем эквивалентные преобразования:

$$\frac{2^k}{n^k}(b-a) < \epsilon$$
$$(\frac{2}{n})^k < \frac{\epsilon}{(b-a)}$$

Так как $n > 2 \implies \frac{2}{n} < 1$. Также $\epsilon < b-a$, ибо в противном случае мы выйдем из цикла на первой же итераций. Тогда:

$$k \cdot ln(\frac{2}{n}) > ln(\frac{\epsilon}{b-a})$$
$$k \cdot (ln(2) - ln(n)) > ln(\epsilon) - ln(b-a)$$
$$k > \frac{ln(\epsilon) - ln(b-a)}{ln(2) - ln(n)} = \frac{ln(\frac{b-a}{\epsilon})}{ln(\frac{n}{2})}$$

Получаем оценку для количества итераций. Заметим, что $b-a$ задается в условии задачи. Точность ϵ также определяется до начала алгоритма, тогда $C = ln(\frac{b-a}{\epsilon})-$ константа. Следовательно, запишем оценки для k: $k_{low} = [\frac{ln(\frac{b-a}{\epsilon})}{ln(\frac{n}{2})}]$, $k_{up} = [\frac{ln(\frac{b-a}{\epsilon})}{ln(\frac{n}{2})}] + 1$, $[...]$ - взятие целой части. Далее на практике убедимся, что количество итераций будет равно следующему целому числу для $\frac{ln(\frac{b-a}{\epsilon})}{ln(\frac{n}{2})}$. Заметим, что при $n \in (2, 2e)$, то есть при $n = 3, 4, 5$; $\quad ln(\frac{n}{2}) < 1 \implies k > ln(\frac{b-a}{\epsilon})$. А при $n > 2e : ln(\frac{n}{2}) > 1 \implies k < ln(\frac{b-a}{\epsilon})$. Откуда можно сделать вывод, что при заданной точности ϵ и начальном отрезке $[a, b]$. Меньшее количество итераций будет получено при большем количестве отрезков разбиения. На рисунке представлен график функции количества итераций от количества отрезков разбиения. Можно наглядно убедиться в вышесказанном.

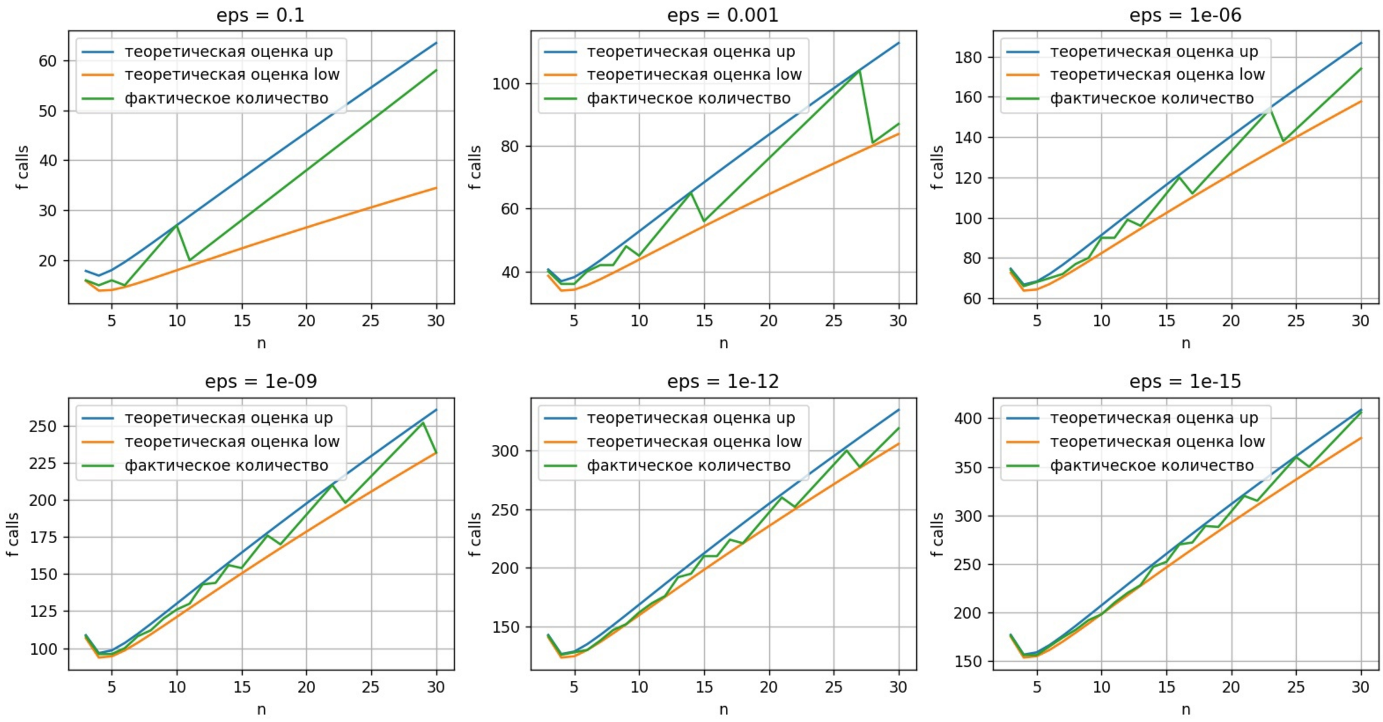


Теперь оценим количество вызовов функции. Так как на каждой итерации мы осуществляем $n-1$ вызов $\implies f_{calls} > (n-1) \cdot [k] = [\frac{ln(\frac{b-a}{\epsilon})}{ln(\frac{n}{2})}] \cdot (n-1)$. Так же оценим сверху $f_{calls} \leq (n-1) \cdot ([k]+1) = [\frac{ln(\frac{b-a}{\epsilon})}{ln(\frac{n}{2})} \cdot (n-1)] + n - 1$.
$]f(n) = \frac{ln(\frac{b-a}{\epsilon})}{ln(\frac{n}{2})} \cdot (n-1) \implies f'(n) = ln(\frac{b-a}{\epsilon}) \cdot \frac{ln(\frac{n}{2}) - \frac{n-1}{n}}{ln^2(\frac{n}{2})} \implies$ определим экстремум. Тогда $f'(n) = 0 \implies ln(\frac{n}{2}) = \frac{n-1}{n} \implies \frac{n}{2} = e^{1-\frac{1}{n}} \implies n \cdot e^{\frac{1}{n}} = 2e$, при этом $n > 2$ нетрудно заметить, что равенство будет достигаться при $n_* \in (5, 6)$. Заметим, что при $n < n_*, \quad f'(n) < 0$. А при $n > n_*, \quad f'(n) > 0 \implies n_*$ - это минимум.

Можем сделать вывод, что для количества отрезков разбиения, равного 5, количество вызовов функции будет меньше, чем для 50 отрезков. Оптиммальным же будет $n \approx 5$.

2.1.2 Сравнение теоретических выкладок с результатом на практике

Для сравнения программно посчитаем для разных точностей ($eps = 10^{-1}, 10^{-3}, 10^{-6}, 10^{-9}, 10^{-12}, 10^{-15}$) зависимость количества вызовов функции от числа отрезков разбиения от 3 до 30. Также нанесем на график полученную теоретическую оценку сверху и снизу.



Можем заметить, что график фактической оценки лежит между графиками для ограничения сверху и снизу. Также оптимальное количество отрезков разбиения $n \approx 5$.

2.2 Метод пробных точек

Описание. Задаются начальный интервал неопределенности $[a, b]$, который делим на 4 отрезка тремя пробными точками. Пробные точки: $x_i = a + \frac{b-a}{4} * i$, где $i = 1, 2, 3$.

Лемма. $f(x)$ - унимодальная функция на $[a, b]$.
Пусть $x_1, x_2 \in [a, b]$, $x_1 < x_2$.
Тогда
1) $f(x_1) \geq f(x_2) \Rightarrow x^* \notin [a, x_1]$
1) $f(x_1) \leq f(x_2) \Rightarrow x^* \notin [x_2, b]$

Алгоритм. Сравниваем значение функции в точках x_1 и x_2: $f(x_1) \leq f(x_2) \Rightarrow [a_{k+1}, b_{k+1}] = [a_{k+1}, x_2]$.
Такой переход делаем опираясь на лемму.

Если же условие в левой части не выполнено, то проводим следующее сравнение: $f(x_2) \leq f(x_3) \Rightarrow [a_{k+1}, b_{k+1}] = [x_1, x_3]$.

Если же и это условие не выполнено, то: $[a_{k+1}, b_{k+1}] = [x_2, b_{k+1}]$.
Будем повторять шаги выше до тех пор, пока $|x_{j+1} - x_{j-1}| \geq \epsilon$
В качестве решения можно выбрать середину конечного интервала неопределённости.

Замечания. Нетрудно заметить, что во всех возможных результатах сравнения мы сократили интервал неопределенности вдвое. А также мы можем заметить, что в любом исходе мы будем знать середину нового интервала неопределенности. Значит, вычисление функции в этой точке можно не производить повторно. И в итоге мы получим, что на следующей итерации число обращений к вычислению функции цели будет ≤ 2.

2.2.1 Теоретические выкладки для оценки количества вызовов функции

Для вывода теоретической формулы количества вызовов функции от точности в методе пробных точек одномерной оптимизации, предположим, что изначально отрезок $[a, b]$ делится на четыре равные части точками $x1$, $x2$ и $x3$, т.е. $a = x0 < x1 < x2 < x3 < x4 = b$.

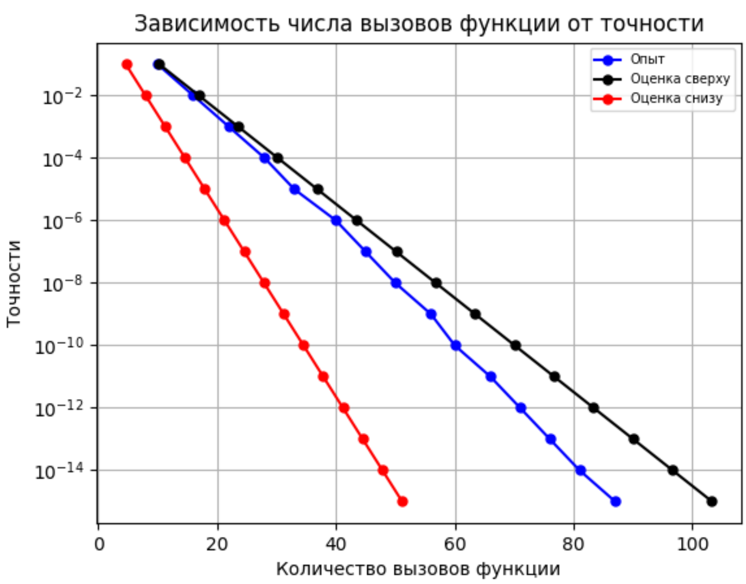
Выведем общее количество итераций для достижения заданной точности ϵ.

Будем пользоваться тем фактом, что во всех возможных результатах сравнения мы сократили интервал неопределенности вдвое, содержащий минимум.

При этом, чтобы достичь заданной точности ϵ, необходимо, чтобы длина текущего отрезка стала меньше или равной ϵ. Предположим, что на первой итерации длина отрезка равна $b - a$. Тогда на $k-$ итерации длина отрезка будет равна $\frac{(b-a)}{2^k}$.

Чтобы найти количество итераций, необходимых для достижения точности ϵ, решим неравенство $\frac{(b-a)}{2^k} \leq \epsilon$. Перепишем его в виде $2^k \geq \frac{(b-a)}{\epsilon}$. Тогда $k \geq \frac{\ln \frac{(b-a)}{\epsilon}}{\ln(2)}$.

Каждый вызов функции включает вычисление значения функции в одной точке. На каждой итерации мы используем три пробные точки, но вызываем функцию не более 2 раз (смотреть замечание). Таким образом,на каждой итерации функция могла быть вызвана 1 или 2 раза, тогда общее количество вызовов функции будет лежать в диапазоне : $1 * \frac{\ln \frac{(b-a)}{\epsilon}}{\ln(2)} \leq f_{calls} \leq 2 * \frac{\ln \frac{(b-a)}{\epsilon}}{\ln(2)}$.



2.3 Метод Фибоначи

Алгоритм

Предварительный этап

Пусть заданы: $[a_1, b_1]$ - начальный интервал неопределенности
ϵ - конечный интервал неопределенности
n - наименьшее число, такое что: $F_n > \dfrac{b_1 - a_1}{\epsilon}$, где F_n - числа Фибоначчи $(F_0 = F_1 = 1, F_i = F_{i-1} + F_{i-2})$
Положим: $\lambda_1 = a_1 + \dfrac{F_{n-2}}{F_n} \cdot (b_1 - a_1)$
$\mu_1 = a_1 + \dfrac{F_{n-1}}{F_n} \cdot (b_1 - a_1)$
Вычислим $f(\lambda_1)$ и $f(\mu_1)$
Положим $k = 1$

Основной этап

Первый шаг Если $f(\lambda_k) > f(\mu_k)$, то перейти ко второму шагу, иначе - к третьему
Второй шаг Положить $a_{k+1} = \lambda_k, b_{k+1} = b_k$
$\lambda_{k+1} = \mu_k, \mu_{k+1} = a_{k+1} + \dfrac{F_{n-k-1}}{F_{n-k}} \cdot (b_{k+1} - a_{k+1})$
Если $k = n - 1$, то перейти к пятому шагу
Иначе: вычислить $f(\mu_{k+1})$ и перейти к четвертому шагу
Третий шаг Положить $a_{k+1} = a_k, b_{k+1} = \mu_k$
$\mu_{k+1} = \lambda_k, \lambda_{k+1} = a_{k+1} + \dfrac{F_{n-k-2}}{F_{n-k}} \cdot (b_{k+1} - a_{k+1})$
Если $k = n - 1$, то перейти к пятому шагу
Иначе: вычислить $f(\lambda_{k+1})$ и перейти к четвертому шагу
Четвертый шаг $k = k + 1$, перейти к первому шагу
Пятый шаг $\dfrac{a_{k+1} + b_{k+1}}{2}$ - ответ

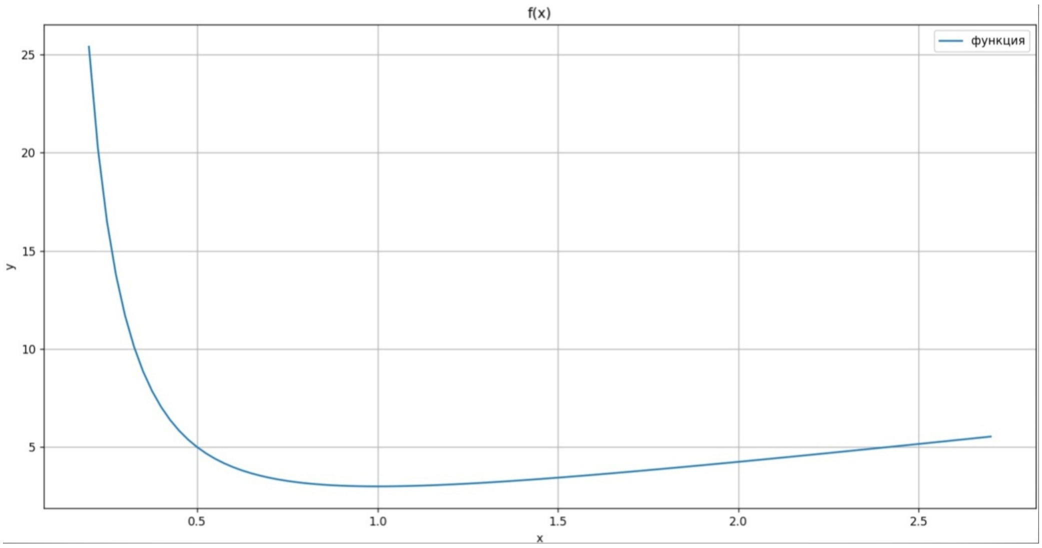
Теоретическая оценка количества обращений к функции На предварительном этапе было произведено **два обращения к функции**
На основном этапе будет проведено $n - 1$ итераций, в каждой из которых, кроме последней будет произведено **одно обращение к функции**
Таким образом будет совершено n обращений, где число n единственным образом определяется на предварительном этапе и зависит только от начального и конечного интервалов неопределенности.

Аналитическое решение задачи линейной минимизации.

Отрезок: $[0.2, 2.7]$

Функция: $f(x) = 2 \cdot x + \dfrac{1}{x^2}$

Проверим условия применимости наших методов, то есть убедимся в унимодальности нашей функции:

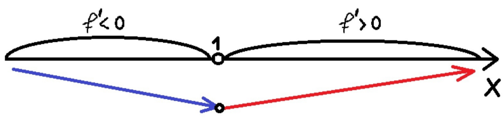


Невооруженным глазом можем заметить, что $\exists x_*$ на данном отрезке такая, что на $[a, x_*]$ функция убывает, а на $[x_*, b]$ функция возрастает, что позволяет судить о ее унимодальности. Из чего следует, что методы применимы.

Найдем минимум функции аналитически:

$$\frac{df(x)}{dx} = 2 - 2\frac{1}{x^3} = 0 \Rightarrow x_{extr} = 1 \Rightarrow f(x_{extr}) = 3$$

Проверим, является ли полученный экстремум точкой минимума:



Рассмотрим значение производной при $x < 1$:

$]x_1 = \frac{1}{2} \implies f'(x_1) = -14 < 0$

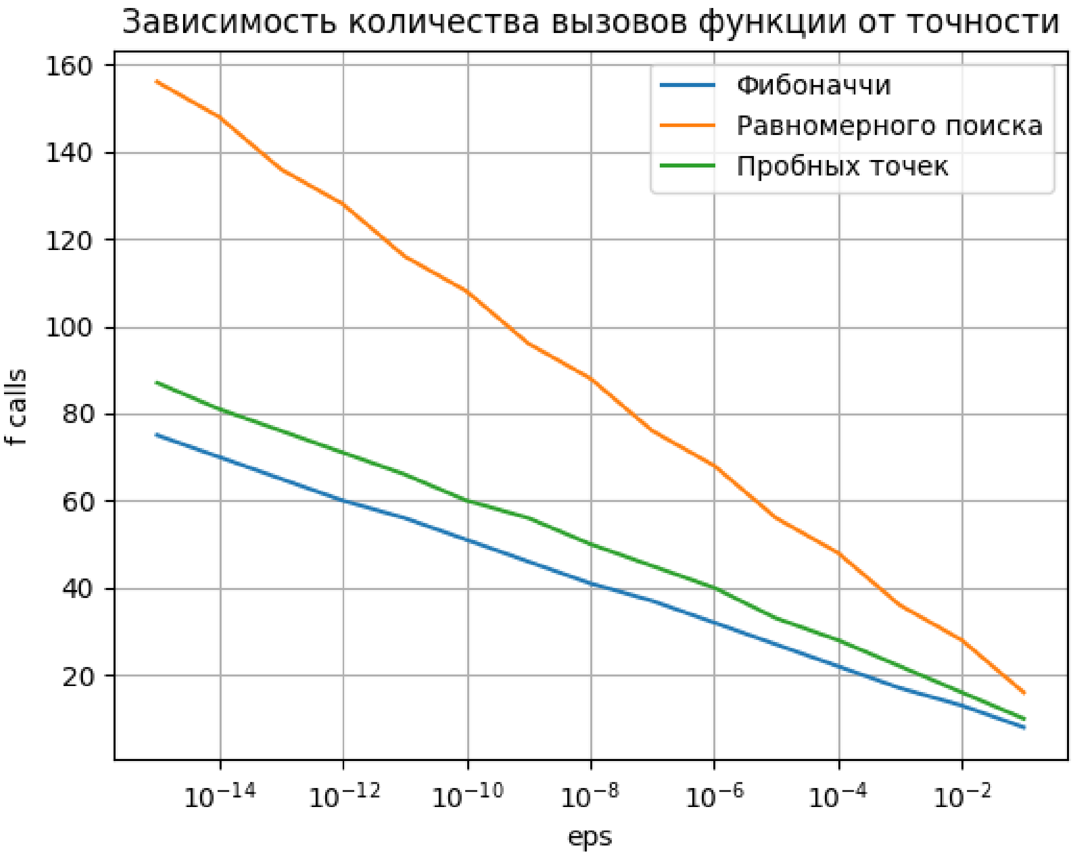
Рассмотрим значение производной при $x > 1$:

$]x_2 = 2 \implies f'(x_2) = \frac{3}{2} > 0$

Из чего можно сделать вывод о том, что в точке $x = 1$ функция достигает минимума $f(1) = 3$.

Таблица.

Сравнение методов одномерной минимизации по количеству вызовов функции от точности			
ϵ	Количество обращений к вызову функции		
	Метод равномерного поиска ($n = 5$)	Метод пробных точек	Метод Фибоначи
10^{-1}	16	10	8
10^{-2}	28	16	13
10^{-3}	36	22	17
10^{-4}	48	28	22
10^{-5}	56	33	27
10^{-6}	68	40	32



Сравнение. Достоинства и недостатки.
Благодаря вышеуказанной таблице и знанию алгоритмов рассматриваемых методов можем провести сравнение. Например, можем заметить, что методу равномерного поиска необходимо неибольшее количество вызовов функции для увеличения точности на порядок выше (по сравнению с другим методами), что обусловлено большим количеством вызовов функции на каждой итерации. В остальных же методах для устранения этого недостатка мы, например, жертвуем простотой алгоритма или требуем дополнительную память.

Метод Фибоначчи
Достоинства:
- Требует меньше всех обращений к функции

Недостатки:
- Необходимо хранить в памяти или вычислять на каждой итерации числа Фибоначчи

Метод Пробных точек
Достоинства:
- Простота реализации: метод пробных точек очень прост в реализации и понимании.
- Эффективность: при правильном выборе начального отрезка и числа пробных точек метод может быть эффективным в поиске минимума функции.
- Эффективность: дает достаточно хорошее количество вызовов функции по сравнению с равномерным поиском.

Недостатки:
- Низкая скорость сходимости: метод пробных точек может требовать большого числа итераций для достижения оптимального результата. Нежеле метод Фибоначи

Метод Равномерного поиска
Достоинства:
- Простота реализации: метод равномерного поиска очень прост в реализации и понимании.
- Гарантированная сходимость: метод равномерного поиска гарантирует сходимость к оптимальному решению, если у нас достаточно времени и точности.

Недостатки:
- Низкая скорость сходимости: метод равномерного поиска может быть медленным в случае, если мы ищем минимум на большом интервале или при большом количестве разбиений. Это может привести к невыгодному использованию вычислительных ресурсов и времени.