«НАЦИОНАЛЬНЫЙ ИССЛЕДВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет информационных технологий и программирования

Кафедра компьютерных технологий

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

Методы одномерной оптимизации

Вариант 4

Выполнили студенты:

Ефимов Сергей Алексеевич
группа: M3237

Соколов Александр Андреевич
группа: M3234

Проверил:

Свинцов Михаил Викторович

г. Санкт-Петербург

Постановка задачи

Задача лабораторной работы – научиться реализовывать алгоритмы одномерной минимизации функции каждым из следуших способов:

1. Метод дихотомии
2. Метод золотого сечения
3. Метод Фиббоначи
4. Метод парабол
5. Комбинированный метод Брента

Также необходимо решить задачу аналитически и привести отчет по результатам сравнения проведенных вычислений

Ход работы

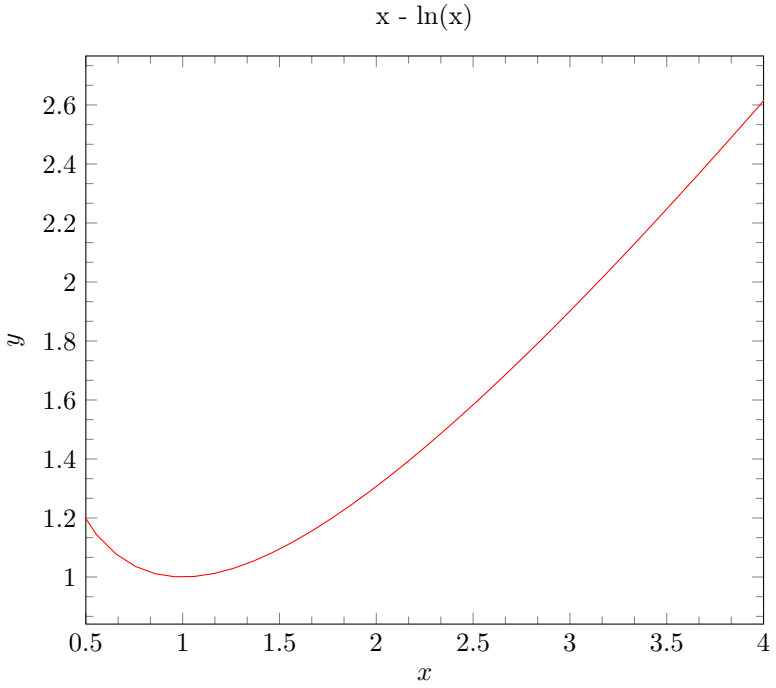
Исследуемая функция

$$f(x) = x - \ln(x)$$

Интервал исследования [0.5, 4]

Аналитическое решение

Для наглядности приведем график функции на заданном интервале:



Чтобы найти min функции необходимо найти ее критические точки(точки, в которых производная функции равна нуля $f'(x) = 0$):

$$f'(x) = (x - \ln(x))' = x' - \ln(x)' = 0$$

$$1 - \frac{1}{x} = 0 \Rightarrow x = 1$$

Итак, мы нашли критическую точку, теперь осталось сравнить три значения функции – в критической точке и в точках, являющиеся концами исследуемного отрезка. Пусть x критическая точка, a, b -начало и конец отрезка соответственно:

$$f(a) = 0.5 - \ln(0.5) \approx 0.5 + 0.6931 \approx 1.1931$$

$$f(b) = 4 - \ln(4) \approx 4 - 1.3863 \approx 2.6137$$

$$f(x) = 1 - \ln(1) = 1$$

Из приведенных выше вычислений можно утверждать, что min функции $f(x) = x - \ln(x)$ равен 1, при $x = 1$, (точка $M(1, 1)$)

Метод дихотомии

Давайте рассмотрим на примере нашей функции метод дихотомии, который основывается на методе приближений, сокращая интревал поиска таким образом, что сохраняется инвариант - с каждой интерацией отрезок сокращается, минимум находится в пределах отрезка.

Написав код к решению задачи данным способом(который можно увидеть на нашем github), мы имеем возможность привести таблицу выполнения данного алгоритма:

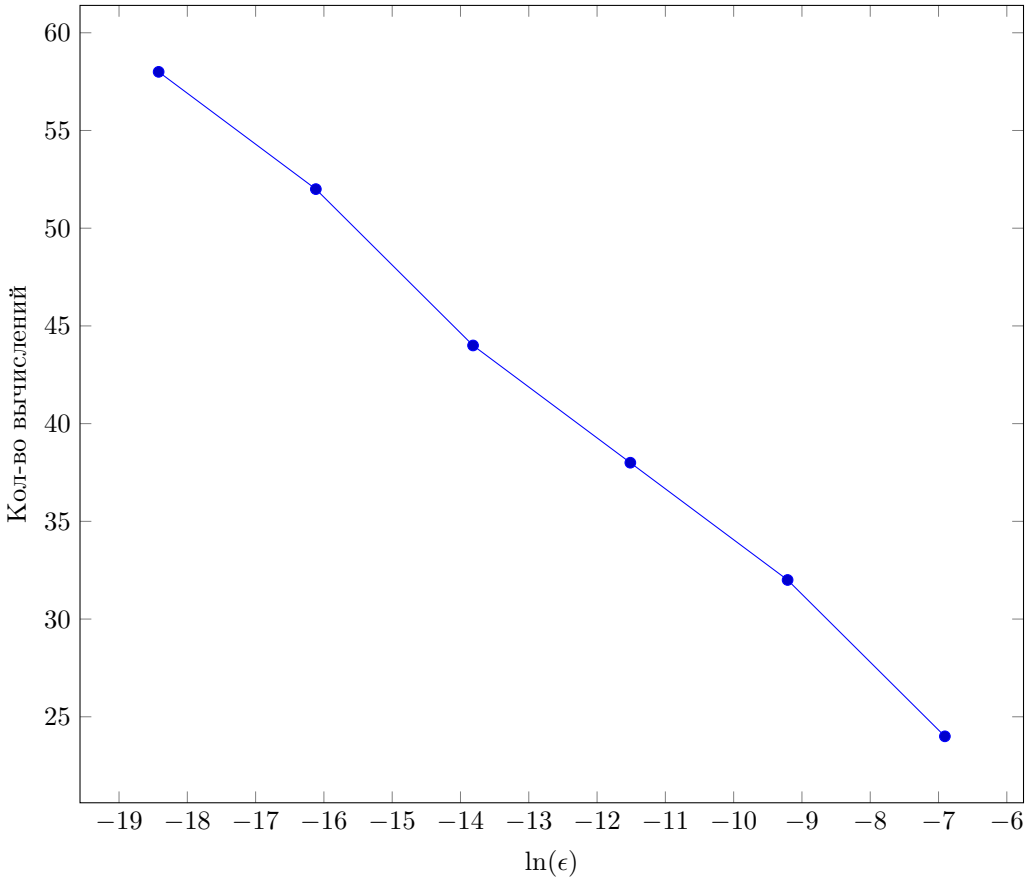
Текущий интервал	Соотношение отрезков	текущий минимум
[0.5, 4]	-	-
[0.5, 2.25]	1.999	1.375
[0.5 1.375]	1.999	0.9375
[0.973, 1.375]	1.999	1.156
[0.973, 1.156]	1.999	1.0468
[0.973, 1.0468]	1.999	0.992
[0.992, 1.0468]	1.999	1.019
[0.992, 1.019]	1.999	1.005
[0.992, 1.005]	1.999	0.999
[0.999, 1.005]	1.999	1.002
[0.999, 1.002]	1.999	1.001
[0.999, 1.001]	1.999	1.000

Исходя из данных таблицы можно утверждать, что каждую итерацию алгоритм сужает область поиска $\approx$ в 2 раза, что говорит о линейной сходимости данного алгоритма. Из этих соображений вытекает формула

$$|x_n - x_{min}| \leqslant \frac{1}{2^n} \cdot (b - a)$$

Чтобы оценить метод дихотомии давайте построим график зависимости кол-ва вычислений функции от логарифма точности:

График зависимости кол-ва вычислений функции от логарифма точности



Исходя из графика можно говорить о том, что кол-во вычислений данного метода линейно зависит от $\ln(\epsilon)$ причем чем больше $\ln(\epsilon)$, тем меньше кол-во вычислений, иными словами алгоритм делает больше итераций, если задаваемая точность выше, что логично

Итого:

- Преимущества
  - Метод является одним из простых в реализации
  - Обеспечивает грантированную сходимость
- Недостатки
  - Сходимость метода всегда постоянна т.е. Метод никогда не сойдется быстрее чем в наихудшем случае

Метод золотого сечения

Теперь рассмотрим метод золотого сечения. Принцип, который и дал название методу говорит о том, что мы будем сокращать интервал поиска нашего решения, причем делить текущий отрезок в прпорциях золотого сечения.

Давайте посмотрим на точки, которые мы получим на очередном этапе выполнения данного метода и проанализируем полученные данные:

Интервал	Соотношение отрезков	текущий минимум	x1	x2
[0.5, 4]	-	-	-	-
[0.5, 2.663]	1.618	1.582	1.326	1.837
[0.5, 1.837]	1.618	1.168	1.011	1.326
[0.5, 1.326]	1.618	0.913	0.816	1.011
[0.816, 1.326]	1.618	1.071	1.011	1.131
[0.816, 1.131]	1.618	0.973	0.936	1.011
[0.936, 1.131]	1.618	1.034	1.011	1.057
[0.936, 1.057]	1.618	0.996	0.982	1.011
[0.982, 1.057]	1.618	1.019	1.011	1.028
[0.982, 1.028]	1.618	1.005	1	1.011
[0.982, 1.011]	1.618	0.996	0.993	1
[0.993, 1.011]	1.618	1.002	1	1.004
[0.993, 1.004]	1.618	0.998	0.997	1
[0.997, 1.004]	1.618	1.001	1	1.001
[0.997, 1.001]	1.618	0.999	0.999	1
[0.999, 1.001]	1.618	1	1	1
[0.999, 1]	1.618	1	0.999	1
[0.999, 1]	1.618	1	1	1
[1, 1]	1.618	1	1	1

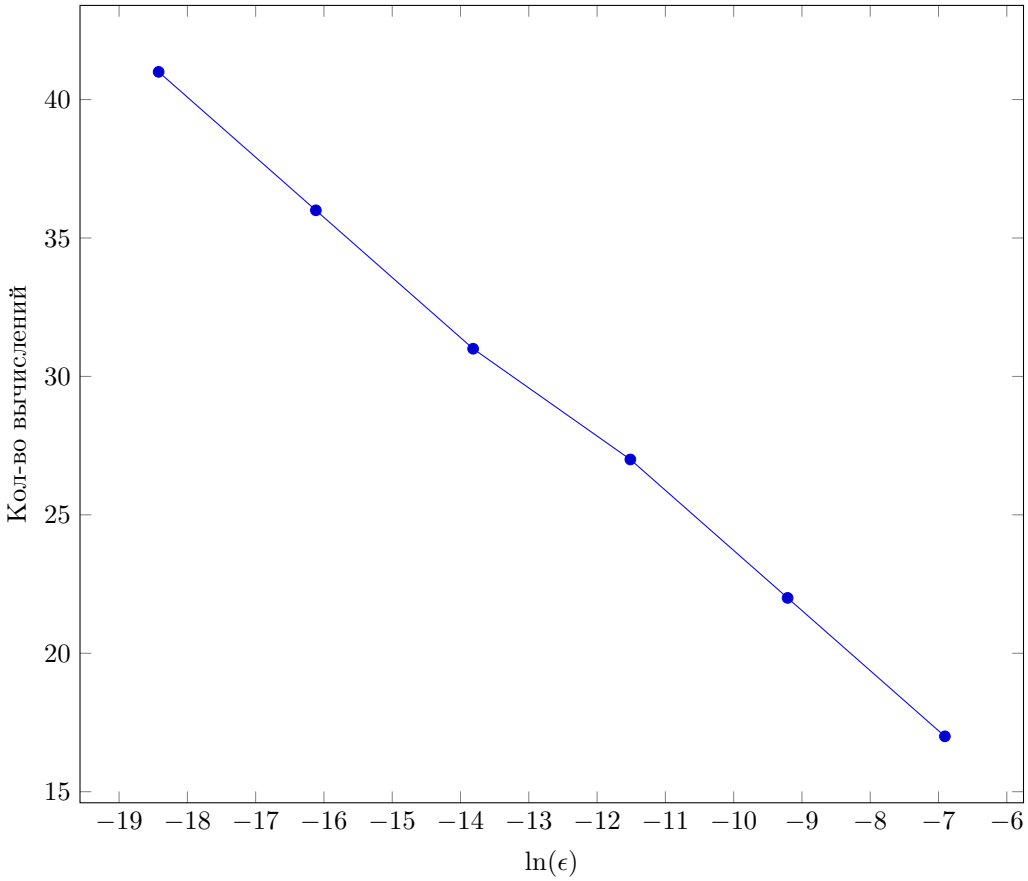
Посмотрев на таблицу можно говорить о следущих вещах:

1. Соотношение отрезков всегда одинаковое(пропорция золотого сечения) что говорит о его гарантированной сходимости и скорости сходимости порядка n(линейная сходимость)

2. Различия с методом дихотомии

Чтобы сделать некоторые выводы по методу, давайте приведем график зависимости кол-ва итераций от $\ln(\epsilon)$

График зависимости кол-ва вычислений функции от логарифма точности



На графике явно видна линейная зависимость. Но чем же тогда отличается метод дихотомии от метода золотого сечения? Стоит обратить внимание на эффективность данного алгоритма засчет деления отрезка в пропорциях золотого сечения.
Итого:

- Преимущества
  - Метод обладает неплохой эффективностью, выше чем у метода дихотомии, но существуют методы с более высокой эффективностью
- Недостатки
  - Метод выолняет большое кол-во итераций, поэтому велика возможность накопления ошибки, что не есть хорошо

Метод Фиббоначи

Очередной метод, который мы рассмотрим - метод Фиббоначи. Проанализированные ранее методы схожи с данным способом оптимизации. Его идея опять же заключается в последовательном делении отрезка поиска, НО есть

различия, данный метод делит текущий отрезок на подотрезки таким образом, что сохраняется равенство:

На отрезке $[a_k, b_k]$ имеем:

$$\lambda_k = a_k + \frac{F_{n-k-1}}{F_{n-k+1}} \cdot (b_k - a_k)$$

$$\mu_k = a_k + \frac{F_{n-k}}{F_{n-k+1}} \cdot (b_k - a_k)$$

Где $k = 1, 2, 3, 4, ..., n-1$ и n - Общее число вычислений, а λ_k, μ_k - вычисляемые точки на итерации метода.

Чтобы говорить о дальнейшем анализе метода, давайте построим таблицу значений:

Интервал	Соотношение отрезков	текущий минимум	λ_k	μ_k
[0.5, 4]	-	-	-	-
[0.5, 2.663]	1.618	1.582	1.326	1.837
[0.5, 1.837]	1.618	1.168	1.011	1.326
[0.5, 1.326]	1.618	0.913	0.816	1.011
[0.816, 1.326]	1.618	1.071	1.011	1.131
[0.816, 1.131]	1.618	0.973	0.936	1.011
[0.936, 1.131]	1.618	1.034	1.011	1.057
[0.936, 1.057]	1.618	0.996	0.982	1.011
[0.982, 1.057]	1.618	1.019	1.011	1.028
[0.982, 1.028]	1.618	1.005	1	1.011
[0.982, 1.011]	1.618	0.996	0.993	1
[0.993, 1.011]	1.618	1.002	1	1.004
[0.993, 1.004]	1.618	0.998	0.997	1
[0.997, 1.004]	1.618	1.001	1	1.001
[0.997, 1.001]	1.618	0.999	0.999	1
[0.999, 1.001]	1.618	1	1	1
[0.999, 1]	1.618	1	0.999	1
[0.999, 1]	1.618	1	1	1
[1, 1]	1.618	1	1	1

Можно заметить, что таблица занчений очень похожа на таблицу значений метода зологото сечения. Все различие с предыдущим подходом заключается в том, что коэфицент сокращения интервала поиска не статичен, он может меняться от итерации к итерации

Для большего кол-ва данных приведем график зависимости от логарифма вычислений

График зависимости кол-ва вычислений функции от логарифма точности

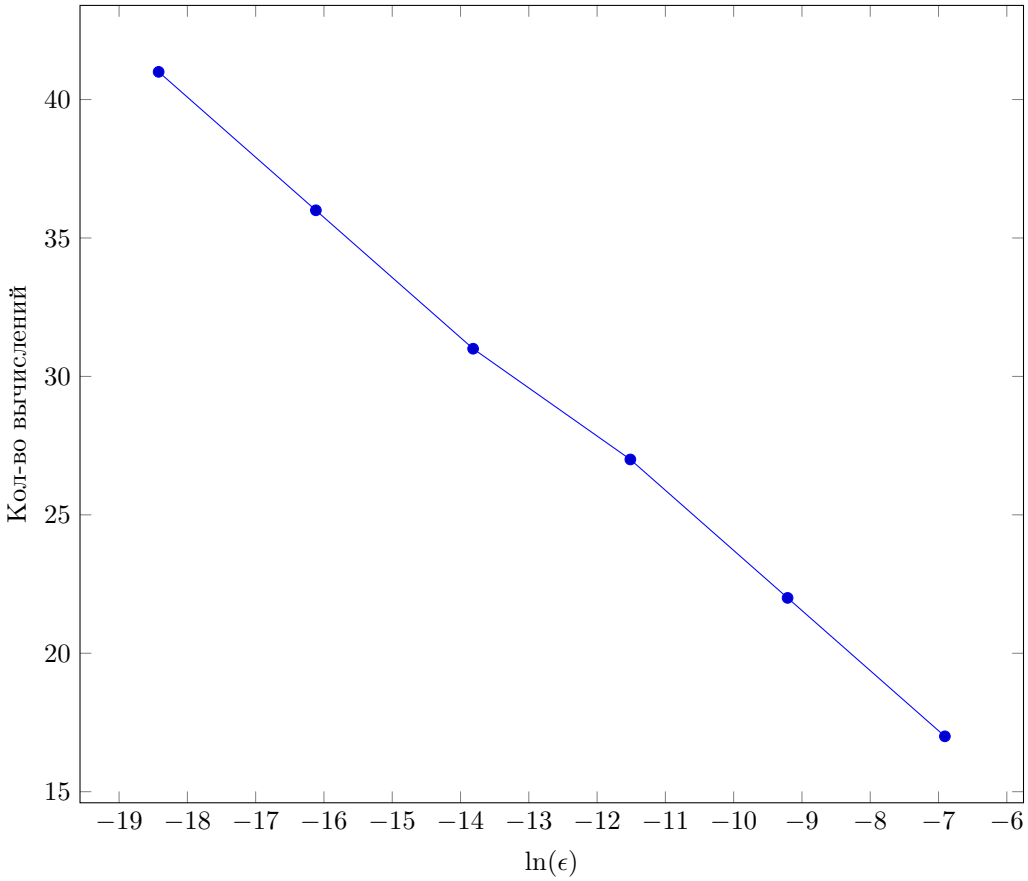


График получился аналогичный графику метода Золотого сечения. Можно подвести промежуточный итог:

- Преимущества
  - Метод является «надстройкой» метода золотого сечения, которая увеличивает эффективность алгоритма
- Недостатки
  - Необходимо знать n - кол-во итераций которое мы будем совершать

Метод парабол

Метод парабол принципиально другой метод оптимизации, нежели мы рассматривали ранее. Его основная идея заключается в нахождении квадратичной функции, с помощью которой мы попытаемся аппроксимировать минимизируемую функцию

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

Пусть имеются три точки $x_1 < x_2 < x_3$ такие, что интервал $[x_1, x_3]$ содержит точку минимума функции $f(x)$. Тогда коэффициенты аппроксимирующей параболы a, b, c могут быть найдены путем решения системы линейных уравнений:

$$ax_i^2 + bx_i + c_i = f(x_i), i = 1, 2, 3....$$

Минимум такой параболы равен:

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

Этот мимимум и бует минимумом исходной функции на данной итерации

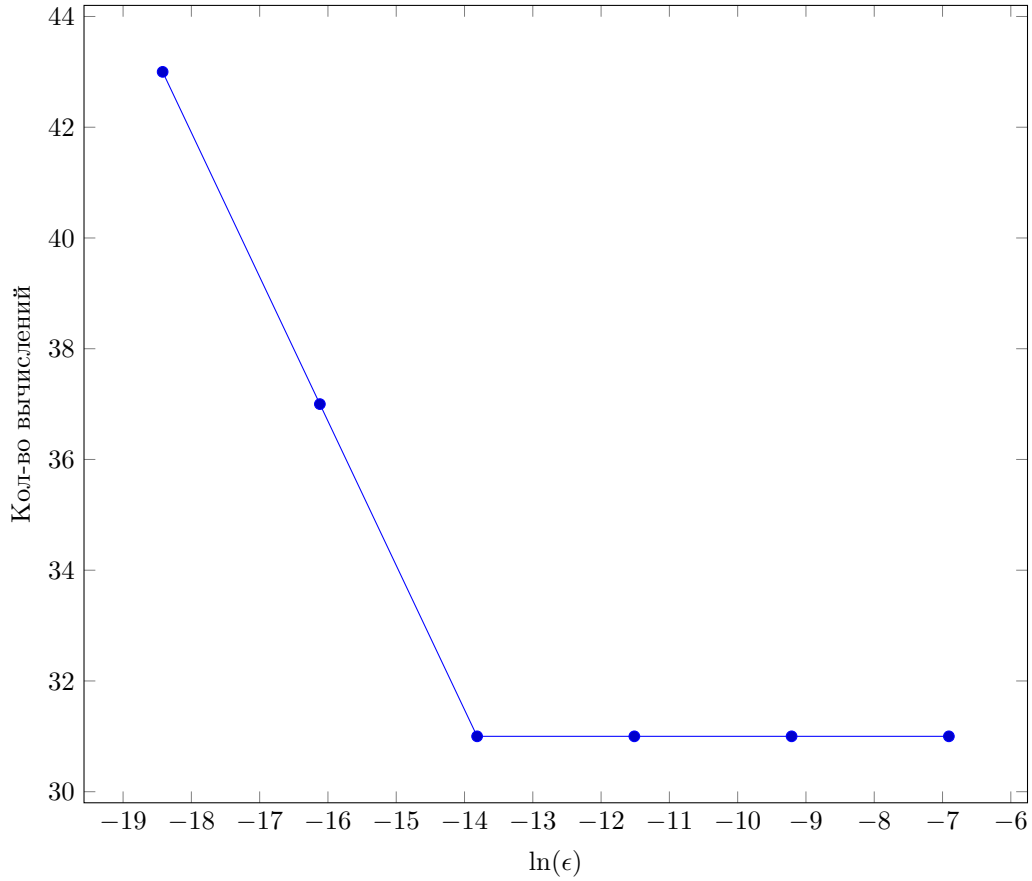
Построим таблицу значений данного метода:

Интервал	Соотношение отрезков	минимум параболы
[0.5, 4]	-	-
[0.5, 2.25]	2	1.375
[0.5, 1.218]	2.437	0.859
[0.912, 1.218]	2.343	1.065
[0.912, 1.027]	2.66	0.969
[0.912, 1.005]	1.23	0.958
[0.912, 1.001]	1.05	0.956
[0.912, 1]	1.008	0.956
[0.912, 1]	1.002	0.956
[1, 1]	∞	1

Стоит заметить, что данный метод совершил намного меньше итераций чем предыдущие.

Чтобы убедиться в этом построим график зависимости кол-ва вычилений от логарифма задаваемой точности:

График зависимости кол-ва вычислений функции от логарифма точности



Грфик говорит о квадратичной сходимости, которая дает свои плюсы и минусы

- Преимущества
  - Если необходима высокая точность решения, то данный метод хорошо подходит
  - Скорость сходимости. Алгоритм имеет квадратичную скорость сходимости
- Недостатки
  - Метод не надежен, Существуют функции на которых алгоритм дает сбой.

Комбинированный метод Брента

Данный метод является комбинированием 2 методов: метода золотого сечения и метода парабол. Метод парабол работает быстрее в малой окрестности решения, но может работать долго и неустойчиво на наччальных итерациях. Тут на помощь приходит метод золотого сечения. Более формально

метод отслеживает занчение в 6 точках: концы отрезка$(a, c), x, v, w, u$, где x – текущий минимум, w – второе снизу значению функции, v – предыдущее занчение w. Точка u – корень аппроксимирующей параболы, которая будет использоваться если:

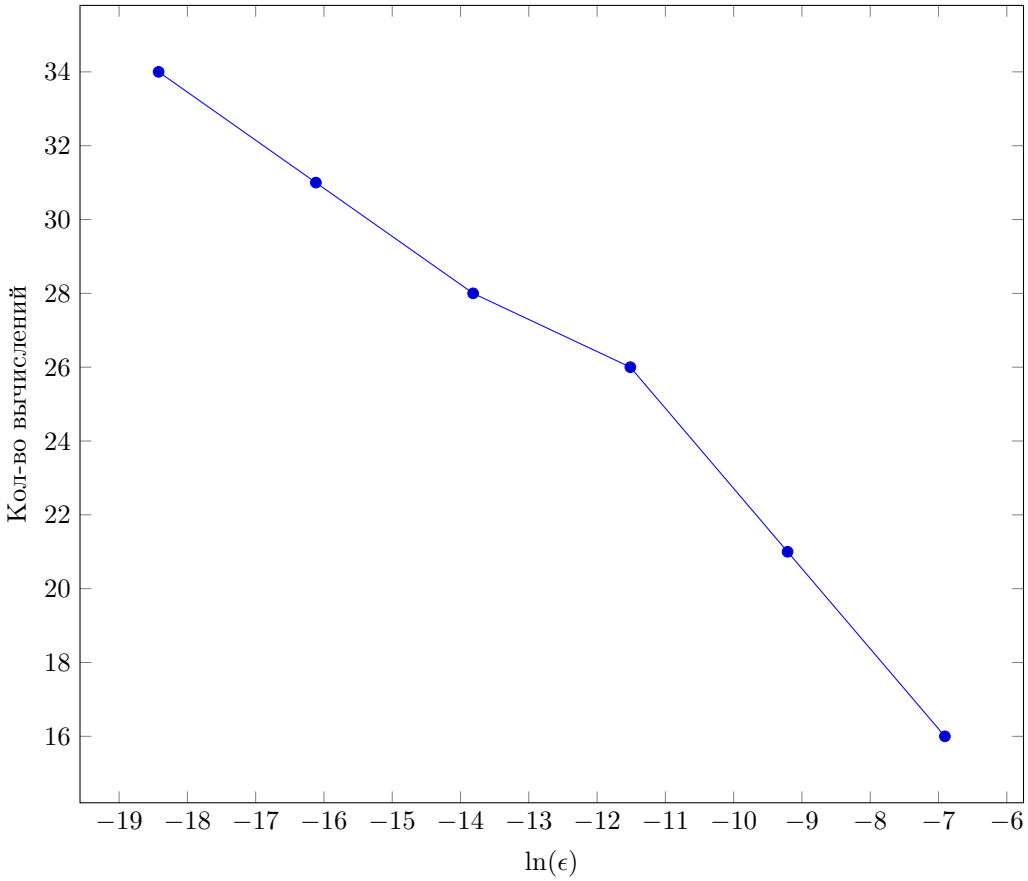
- $u \in [a, c]$
- u отстоит от точки x не более, чем на половину от длины предпредыдущего шага

Приведем таблицу значений нового метода:

Интервал	Соотношение отрезков	текущий минимум	x	w	v
[0.5, 4]	-	-	-	-	-
[0.5, 2.663]	1.618	1.582	1.326	2.663	2.663
[0.5, 2.152]	1.309	1.326	1.326	2.152	2.663
[0.5, 2.152]	1	1.326	1.326	2.152	2.152
[0.5, 1.837]	1.236	1.168	1.326	1.837	2.152
[0.5, 1.837]	1	1.168	1.326	1.837	1.837
[0.5, 1.326]	1.618	0.913	0.816	1.326	1.837
[0.816, 1.326]	1.618	1.071	0.993	0.816	1.326
[0.816, 1.02]	2.496	0.918	0.993	1.02	0.816
[0.993, 1.02]	7.55	1.007	1.001	0.993	1.02
[0.993, 1.001]	3.479	0.997	1	1.001	0.993
[0.996, 1.001]	1.519	0.998	1	1.001	0.993
[0.997, 1.001]	1.472	0.999	1	1.001	0.993
[0.998, 1.001]	1.411	1	1	0.998	1.001
[0.999, 1.001]	1.341	1	1	0.999	0.998
[0.999, 1.001]	1.267	1	1	0.999	0.999
[0.999, 1.001]	1.277	1	1	1.001	0.999
[1, 1.001]	1.267	1	1	1	1.001
[1, 1]	1.277	1	1	1	1

По колонке «Соотношение отрезков» можно отследить, что алгоритм чередует метод золотого сечения (1.618) и метод парабол. Давате взглянем на график зависимости кол-ва вычислений от лографима задаваемой точности:

График зависимости кол-ва вычислений функции от логарифма точности



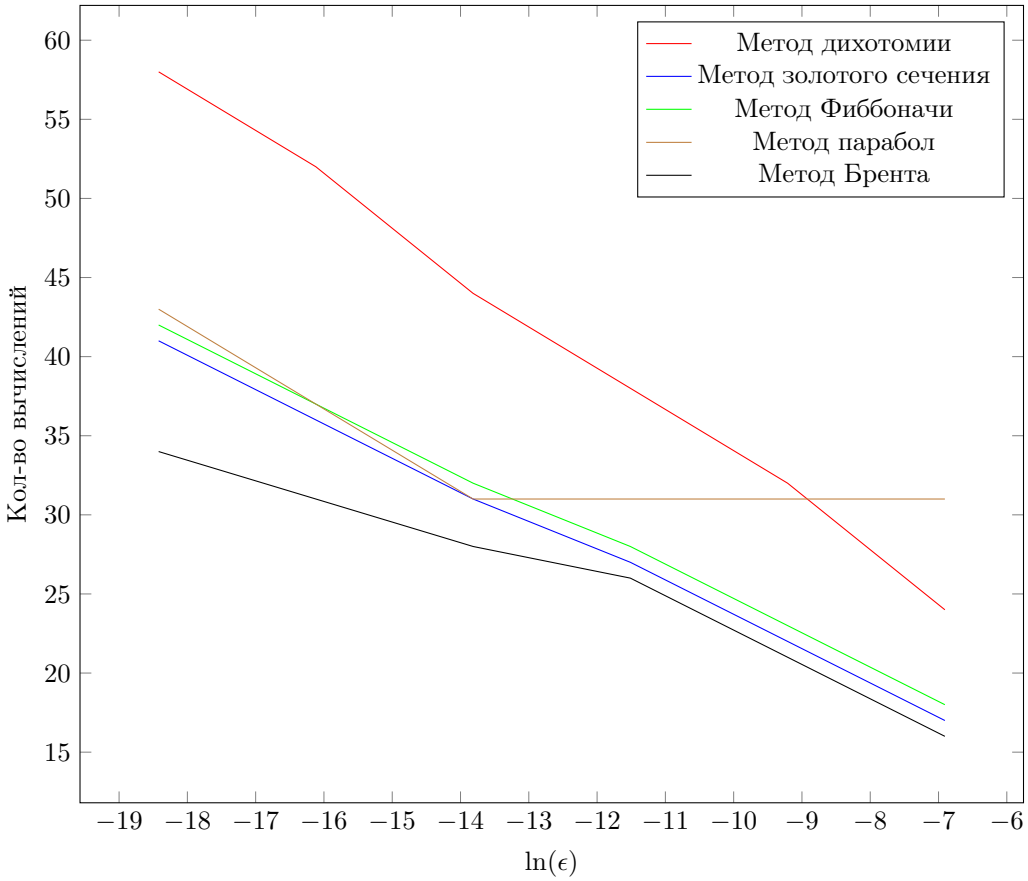
Поведение графика говорит о линейной сходимости данного алгоритма, так же можно утверждать, что метод сойдется гарантировано за конечное число итераций.
Итого:

- Преимущества
  - Обладает отличным контролем точности вычислений
  - Алгоритм гарантированно сходится за n итераций
- Недостатки
  - Скорость сходимости линейная

Сравнение

Приведем общий график эффективности функций для их сравнения

График зависимости кол-ва вычислений функции от логарифма точности



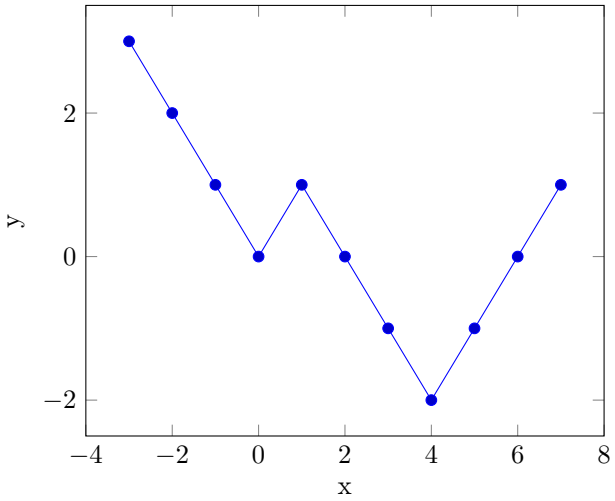
Проверка алгоритмов на унимодальность

Давайте рассмотрим функцию:

$$f(x) = \begin{cases} x < 1, f(x) = |x| \\ x \geqslant 1, f(x) = |x - 4| - 2 \end{cases}$$

Ее график:

График приведенной нами многомодальной функции:



По графику несложно заметить, что на отрезке $[-3, 7]$ функция является многомодальной. Запустив каждый из алгоритмов на данной функции, увидим, что результаты отличаются друг от друга, причем ни один из них не является верным. Почему же так происходит? Пусть дали изначальный отрезок на котором находится 2 локальных минимума, мы знаем, что для каждого вышеизложенного алгоритма выполянется инвариант: «алгоритм находит минимум функции, при условии что она унимодальна». Из данных рассуждений следует, что при очередной итерации алгоритм может шагнуть таким образом, что в области поиска окажется один локальный минимум, который не является глобальным, но т.к. алгоритм этого не поймет, для него это будет обычным нахождением минимума на отрезке.

Вывод

Из всего вышесказанного можно сделать вывод о том, что каждый из методов имеет свои преимущества и недостатки. Можно заметить некую закономерность: чем проще метод в релизации, тем меньше он применим в промышленной разработке, стоит отметить комбинированный метод Брента, который заключает в себе преимущества сразу двух методов, минимизируя их недостатки.

Рассмотрев множество предложенных нам методов, мы можем утверждать, что

- Если нам необходим метод, который выполняет поставленную задачу и прост в реализации, то как никто другой подходит метод дихотомии
- Методы золотого сечения и Фиббоначи имеют некое преимущество над методом дихотомии, которые дают прирост эффективности, а так же мало чем отличаются по сложности реализации, имея разные подходы к решению поставленной задачи
- Метод парабол принципиально новый метод оптимизации, реализованный через нахождение аппроксимирующей квадратичной функции

на каждой итерации. Такой метод дает прирост скорости сходимости, но теряется надежность метода, т.к. на начальных итерациях метод может делать слишком маленький шаг оптимизации, что сказывается на эффективности алгоритма

- И наконец комбинированный метод Брента – метод, который в зависимости от ситуации имеет возможность максимизировать эффективность каждой итерации, что говорит о наилучшей реализации из всех вышеперечисленных методов.