

Отчет по лабораторной работе "Корни уравнения"

Студент группы А-07-22 Татарников Максим

НИУ МЭИ 2023

Содержание

1	Задание	2
2	Функции	2
3	Методы	2
3.1	Деление отрезка пополам	2
3.1.1	Описание метода	2
3.1.2	Блок-схема	3
3.1.3	Программный код	4
3.2	Метод хорд	4
3.2.1	Описание метода	4
3.2.2	Блок-схема	5
3.2.3	Программный код	6
3.3	Сравнение методов	7

1 Задание

Методом деления отрезка пополам и методом хорд для N значений погрешности ε ($0,1; 0,01; 0,001; \dots 1e-N, 1 \leq N \leq 10$) вычислить значение корня для двух заданных функций №4 и №7 на отрезке $[A, B] \subset (0, 2)$ и вывести их в виде таблицы (для каждой функции отдельную).

При вводе значений анализировать аномалии: A, B, N – числа и $0 < A < B < 2, 1 \leq N \leq 10$. Перед поиском корня обязательно проверять наличие корня на отрезке $[A, B]$ (разные знаки значения функции на концах отрезка).

Для самопроверки: 1) для каждой функции указан примерный корень для проверки написания заданной функции, и 2) корни, найденные разными методами с одинаковой точностью не должны отличаться больше, чем на эту точность.

2 Функции

$$f_1(x) = 1.5 - \frac{\sqrt{x} + \sqrt[3]{x}}{e^{3/2}} - x \quad (1)$$

$$f_2(x) = 0.5(2 - \sin \frac{1+x}{x} + \frac{1}{2} \ln \sqrt{x}) - x \quad (2)$$

3 Методы

3.1 Деление отрезка пополам

3.1.1 Описание метода

Рисунок 1 иллюстрирует метод половинного деления, который состоит в постепенном сужении отрезка поиска корня до заданной величины ε . На каждом шаге отрезок уменьшается вдвое.

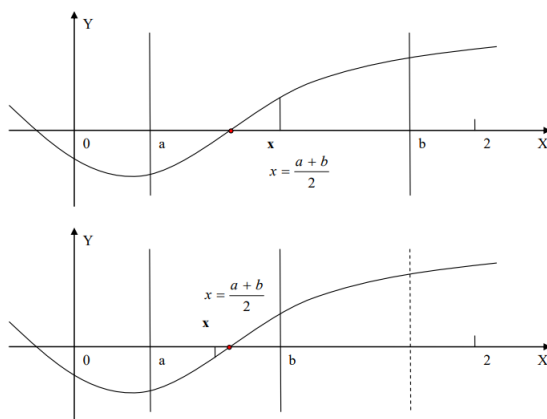


Рис. 1: Метод деление отрезка пополам

В начале каждой итерации находим середину нового отрезка $[a, b]$

$$x = \frac{a+b}{2}$$

Затем следует определить, с какой стороны от середины отрезка x находится корень x^* . Для этого достаточно сравнить знаки $f(x)$ и $f(b)$ или знаки $f(x)$ и $f(a)$.

Если знаки $f(x)$ и $f(b)$ не совпадают, то это означает, что $f(x)$ пересекает ось x на правом полуотрезке $[x, b]$. Следовательно, корня нет на левом полуотрезке $[a, x]$, и этот полуотрезок

можно отбросить, то есть можно перенести левую границу a в среднюю точку x (заменить значение приближения a на значение x).

Если же знаки $f(x)$ и $f(b)$ совпадают, то $f(x)$ пересекает ось x на левом полуотрезке $[a, x]$ и, следовательно, корня нет на правом полуотрезке $[x, b]$, и этот полуотрезок можно отбросить, то есть можно перенести правую границу b в среднюю точку x (заменить значение приближения b на значение x).

Итак, в результате выполнения итерации отрезок $[a, b]$ как и прежде, содержит единственный корень, но его длина стала меньше в два раза.

Вычисления следует прекратить, если на очередном шаге длина отрезка $[a, b]$ станет меньше ε . Тогда с точностью ε любая точка этого отрезка будет являться корнем уравнения $f(x)$, а середина этого отрезка – с точностью $\varepsilon/2$. Совпадение знаков $f(x)$ и $f(b)$ можно проверить, проверив неравенство

$$f(x) \cdot f(b) > 0$$

поскольку произведение двух чисел с одинаковыми знаками дает положительное значение.

3.1.2 Блок-схема

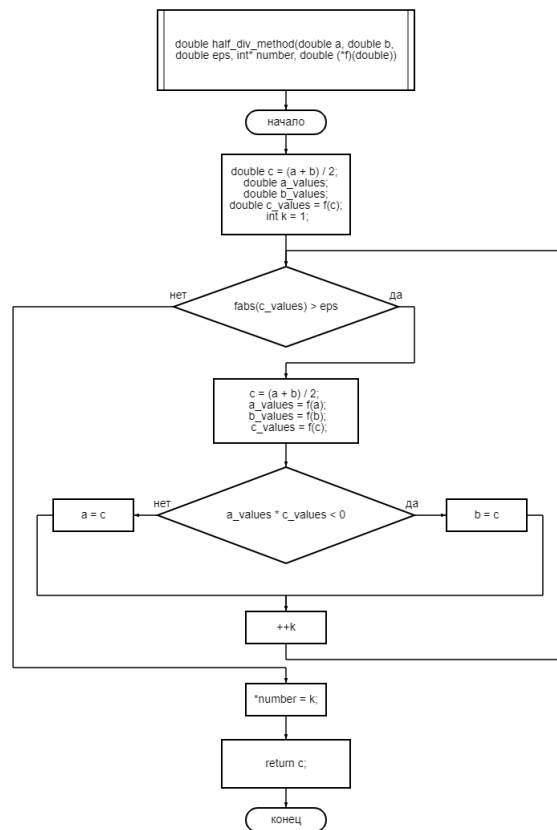


Рис. 2: Метод деление отрезка пополам

3.1.3 Программный код

```
double half_div_method(double a, double b, double eps, int* number, double (*f)(double))
{
    double c = (a + b) / 2;
    double a_values;
    double b_values;
    double c_values = f(c);
    int k = 1;
    while (fabs(c_values) > eps)
    {
        c = (a + b) / 2;
        a_values = f(a);
        b_values = f(b);
        c_values = f(c);
        if (a_values * c_values < 0) b = c;
        else a = c;
        ++k;
    }
    *number = k;
    return c;
}
```

Рис. 3: Метод деление отрезка пополам

3.2 Метод хорд

3.2.1 Описание метода

Этот метод вместе с методом бисекции относится к методам дихотомии – как отрезок делится на две части, но на этот раз неравных. Точка деления отрезка находится как точка пересечения отрезка, проведенного между точками $(a, f(a))$ и $(b, f(b))$, с осью OX по формуле:

Новые значения a и b вычисляются так же, как и в методе бисекции – в зависимости от знаков на границах новых отрезков $[a, x]$ и $[x, b]$.

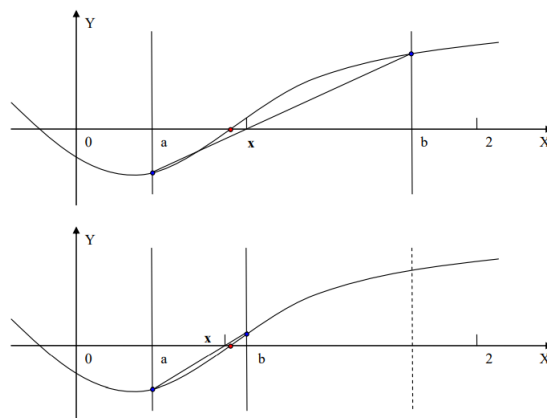


Рис. 4: Метод хорд

При выполнении итераций по методу хорд может оказаться, что к корню приближается только левая или только правая граница отрезка $[a, b]$. Поэтому в качестве меры близости к корню здесь следует применить величину перемещения границы при очередной итерации, которая равна:

$$d = \begin{cases} x - a, & \text{если корень справа от } x \text{ и перемещаем } a, \\ b - x, & \text{если корень слева от } x \text{ и перемещаем } b. \end{cases}$$

Необходимая точность будет достигнута при выполнении после очередной итерации неравенства:

$$|d| < \varepsilon$$

Полученное значение приближения x надо взять в качестве искомого значения корня.

3.2.2 Блок-схема

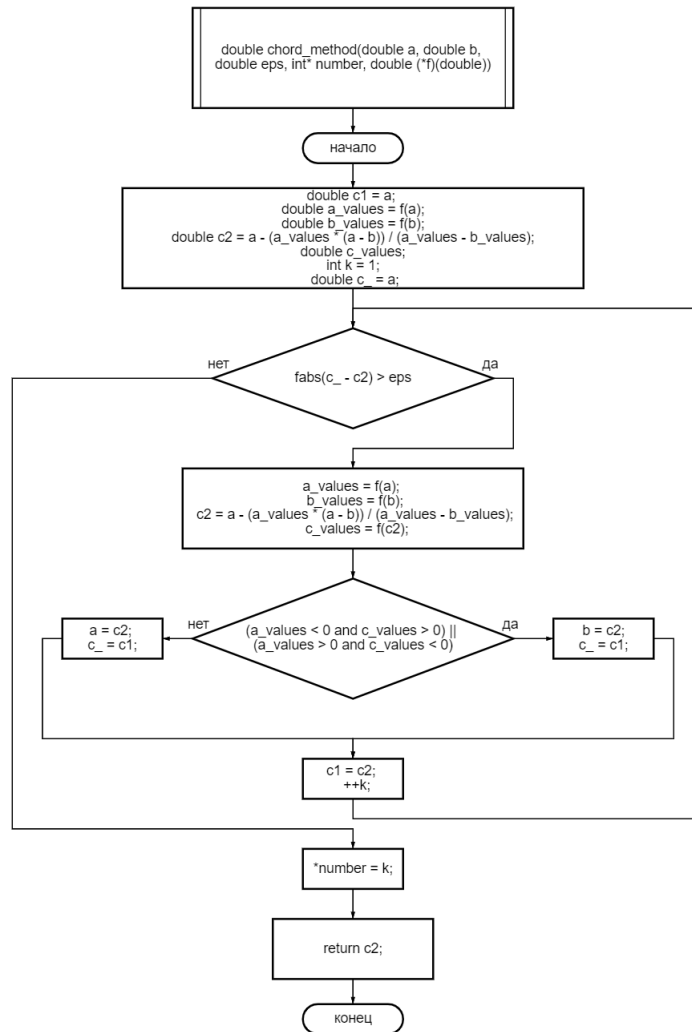


Рис. 5: Метод хорд

3.2.3 Программный код

```
double chord_method(double a, double b, double eps, int* number, double (*f)(double))
{
    double c1 = a;
    double a_values = f(a);
    double b_values = f(b);
    double c2 = a - (a_values * (a - b)) / (a_values - b_values);
    double c_values;
    int k = 1;
    double c_ = a;
    while (fabs(c_ - c2) > eps)
    {
        a_values = f(a);
        b_values = f(b);
        c2 = a - (a_values * (a - b)) / (a_values - b_values);
        c_values = f(c2);
        if ((a_values < 0 and c_values > 0) || (a_values > 0 and c_values < 0))
        {
            b = c2;
            c_ = c1;
        }
        else
        {
            a = c2;
            c_ = c1;
        }
        c1 = c2;
        ++k;
    }
    *number = k;
    return c2;
}
```

Рис. 6: Метод хорд

3.3 Сравнение методов

Различие двух методов состоит в том, что в случае деления отрезка пополам мы сравниваем значение функции с погрешностью, а в методе хорд сравнение предыдущего и нового полученного аргумента, что напоминает предел последовательности, когда деление пополам - предел функции.

Сравним 2 метода:

Введите n ($0 < n \leq 10$): 10					
Введите a,b ($0 < a < b < 2$): 0.1 1.9					
Функция №1					
Точность	Метод хорд		Попалам		
	корень	кол-во итераций	корень	кол-во итераций	
0.1000000000	1.04999940822	3	1.0000000000	1	
0.0100000000	1.04581120384	4	1.0421875000	8	
0.0010000000	1.04543964790	5	1.04570312500	10	
0.0001000000	1.04540660458	6	1.04548339844	14	
0.0000100000	1.04540366532	7	1.04540100098	17	
0.0000010000	1.04540340386	8	1.04540271759	21	
0.0000001000	1.04540338061	9	1.04540336132	24	
0.0000000100	1.04540337854	10	1.04540337473	28	
0.0000000010	1.04540337835	11	1.04540337808	30	
0.0000000001	1.04540337834	12	1.04540337829	34	
0.0000000000	1.04540337834	13	1.04540337833	38	
Функция №2					
Точность	Метод хорд		Попалам		
	корень	кол-во итераций	корень	кол-во итераций	
0.1000000000	0.66714218463	4	0.6625000000	5	
0.0100000000	0.65840570727	5	0.6625000000	5	
0.0010000000	0.65711550288	7	0.65722656250	11	
0.0001000000	0.65709815866	8	0.65711669922	15	
0.0000100000	0.65709603890	9	0.65709609985	19	
0.0000010000	0.65709577987	10	0.65709609985	19	
0.0000001000	0.65709574821	11	0.65709577799	25	
0.0000000100	0.65709574434	12	0.65709574446	29	
0.0000000010	0.65709574387	13	0.65709574362	32	
0.0000000001	0.65709574381	14	0.65709574383	34	
0.0000000000	0.65709574381	15	0.65709574381	37	

Рис. 7: Таблицы данных

Не трудно заметить, что количество итераций в методе хорд $\sim 2 - 2.5$ раза меньше, чем в методе деления отрезка пополам, из чего можно сделать вывод, что метод хорд эффективней.