Дисциплина «Алгоритмы решения прикладных задач»
Рабочая тетрадь 1.
Метод половинного деления. Метод хорд

Теоретический материал

Метод половинного деления для нахождения квадратного корня и корня уравнения

Квадратный корень в программах встречается очень часто, при этом его вычисление достаточно трудоемко. Еще в 1950-х годах соответствующая операция была вынесена в специальный математический сопроцессор — центральный процессор отправлял в него запрос и пока выполнялись вычисления он успевал обрабатывать другие важные команды.

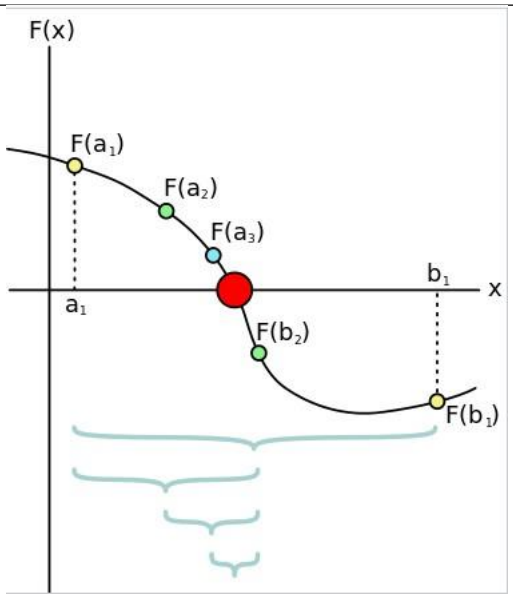
Метод половинного деления относится к серии алгоритмов, построенных по принципу «разделяй и властвуй». Он применяется для поиска корней уравнений. Допустим, есть у нас некоторая функция f(x), известно, что функция монотонна на некотором интервале [L,R]. Монотонность — обязательное требование для использования этого алгоритма, оно означает, что функция либо только возрастает на этом интервале, либо — убывает. В общем, на интервале нет перегибов функции, т.е. точек, в которых производная равна нулю.

Тогда, если на концах интервала функция имеет разные знаки — она обязательно пересекает горизонтальную ось, т.е. имеет корень.

Если f(L) * f(R) > 0 — значит функция на этом интервале корня не имеет.

Как же найти где именно находится этот корень? — опять же итеративно.

Возьмем точку посередине интервала (B = L + (R-L)/2) — по знаку f(B) можно определить где именно находится корень (правее этой точки или левее). Если f(L)*f(B) < 0 — то корень находится на интервале [L,B] при этом заменим R на B и повторим процесс. В противном случае корень находится на [B,R]. Вычисления продолжаются до тех пор, пока интервал поиска корня не сузится достаточно сильно, т.е. пока |R-L| > Eps (Eps – заданная погрешность, например 0.001). Схематически метод проиллюстрирован на рисунке ниже



Частный случай применения метода половинного деления – это поиск квадратного корня, для его вычисления достаточно решить уравнение типа x*x = A, т.е. f(x) = x*x - A = 0. Начальное значение интервала поиска — [1,A].

Программа для нахождения квадратного корня методом половинного деления представлена в примере 1.1.

Метод хорд

Метод хорд — итерационный численный метод приближённого нахождения корня уравнения.

Половинное деление не учитывает никаких свойств функции F(x), а эта функция может нести в себе очень полезную информацию. Метод хорд предполагает следующее. От точек, ограничивающих кривую (заданные концы отрезка L и R), строится хорда, затем определяется точка её пересечения с осью абсцисс, точка пересечения становится новой границей отрезка, после чего строится новая хорда. Итерационный процесс задается следующей формулой:

$$x_{n+1} = x_n - \frac{(b - x_n) f(x_n)}{f(b) - f(x_n)}$$

Точка пересечения касательной с осью X является точкой приближения (на рисунке точки приближения отмечены числами 1 и 2).

Пример 1.1

Задача:

Написать на языке C++ программу для поиска квадратного корня числа методом половинного деления

Решение:

```cpp
#include <iostream>

int main()
{
    double A, B, L, R;
    A = 16;
    B = 0;
    L = 1;
    R = A;
    int shag=0;
    while (R-L > 0.0001)
    {
        shag++;
        B = (L + R) / 2;
        if (B*B > A )
        {
            R = B;
        }
        else
        {
            L = B;
        }
        std::cout << shag;
        std::cout << " ";
        std::cout << B << std::endl;
    }
    std::cout << B << std::endl;
}
```

Ответ:

```
Консоль отладки Microsoft Visual Studio
1 8.5
2 4.75
3 2.875
4 3.8125
5 4.28125
6 4.04688
7 3.92969
8 3.98828
9 4.01758
10 4.00293
11 3.99561
12 3.99927
13 4.0011
14 4.00018
15 3.99973
16 3.99995
17 4.00007
18 4.00001
4.00001
```

Задание 1.1

Задача:

Решить методом половинного деления. Номер варианта соответствует номеру в списке (номеру в списке минус 15 для номеров 16-30 и номеру в списке минус 30 для номеров от 31 и далее). Оформить алгоритм на языке C++ (или каком-либо другом)

№ варианта	$f(x)$	№ варианта	$f(x)$
1	$e^{x-1} - x^3 - x$ $x \in [0,1]$	9	$0{,}25x^3 + x - 2$ $x \in [0,2]$
2	$x - \frac{1}{3 + \sin(3{,}6x)}$ $x \in [0,1]$	10	$\arccos\frac{1-x^2}{1+x^2} - x$ $x \in [2,3]$
3	$\arccos x - \sqrt{1 - 0{,}3x^3}$ $x \in [0,1]$	11	$3x - 4\ln x - 5$ $x \in [2,4]$
4	$\sqrt{1 - 0{,}4x^2} - \arcsin x$ $x \in [0,1]$	12	$e^x - e^{-x} - 2$ $x \in [0,1]$
5	$3x - 14 + e^x - e^{-x}$ $x \in [1,3]$	13	$\sqrt{1-x} - \mathrm{tg}x$ $x \in [0,1]$
6	$\sqrt{2x^2 + 1{,}2 - \cos x} - 1$ $x \in [0,1]$	14	$1 - x + \sin x - \ln(1+x)$ $x \in [0,2]$
7	$\cos\left(\frac{2}{x}\right) - 2\sin\left(\frac{1}{x}\right) + \frac{1}{x}$ $x \in [1,2]$	15	$x^5 - x - 0{,}2$ $x \in [1,2]$
8	$0{,}1x^2 - x\ln x$ $x \in [1,2]$		

Решение:

тет_1_зад_1.1.py > ...

```
# точность 0.0001
import math

R = 1
L = 0
B = 0

while R-L > 0.0001:
    B = L + ((R-L) / 2)
    if (math.acos(B) - math.sqrt(1 - 0.3 * B ** 3)) * (math.acos(R) - math.sqrt(1 - 0.3 * R ** 3)) < 0:
        L = B
    else:
        R = B
print(B)
print()
print((math.acos(B) - math.sqrt(1 - 0.3 * B ** 3)))
```

Ответ:

```
0.56292724609375
-1.3751447306420417e-06
```

Задание 1.2

Задача:

Решить задачу 1.1 методом хорд

Решение:

тет_1_зад_1.2.py > ...

```python
# точность 0.0001
import math

def f(x):
    return (math.acos(x) - math.sqrt(1 - 0.3 * x ** 3))

R = 1
L = 0
if f(R) > f(L):
    while f(L) > 0.0001:
        L = L - (((R - L) * (f(L))) / (f(R) - f(L)))
    print(L)
    print(f(L))

else:
    while abs(f(R)) > 0.0001:
        R = R - (((L - R) * (f(R))) / (f(L) - f(R)))
    print(R)
    print(f(R))
```

Ответ:

```
0.562938208495507
-1.3031897122206537e-05
```