Московский авиационный институт
(Национальный исследовательский университет)

Факультет прикладной математики и информатики

Курсовой проект
по курсу
«Фундаментальная информатика»
I семестр
Задание 4

Студент: Постнов А. В.
Группа: М8О-101Б-21
Руководитель: Титов В. К.
Оценка:

Дата: 23.12.21

Москва
2021г.

Задание

Составить программу на языке Си с процедурами решения трансцендентных алгебраических уравнений различными численными методами (итерации, Ньютона, хорд и половинного деления – дихотомии). Нелинейные уравнения оформить как параметры-функции, разрешив относительно неизвестной величины в случае необходимости. Применить каждую процедуру к решению трех уравнений. Если метод неприменим, дать математическое обоснование и графическую иллюстрацию.

Вариант 17:

$0.25 * x^3 + x - 1.2502 = 0$, корень на отрезке [0; 2]

Варианты, составленные самостоятельно:

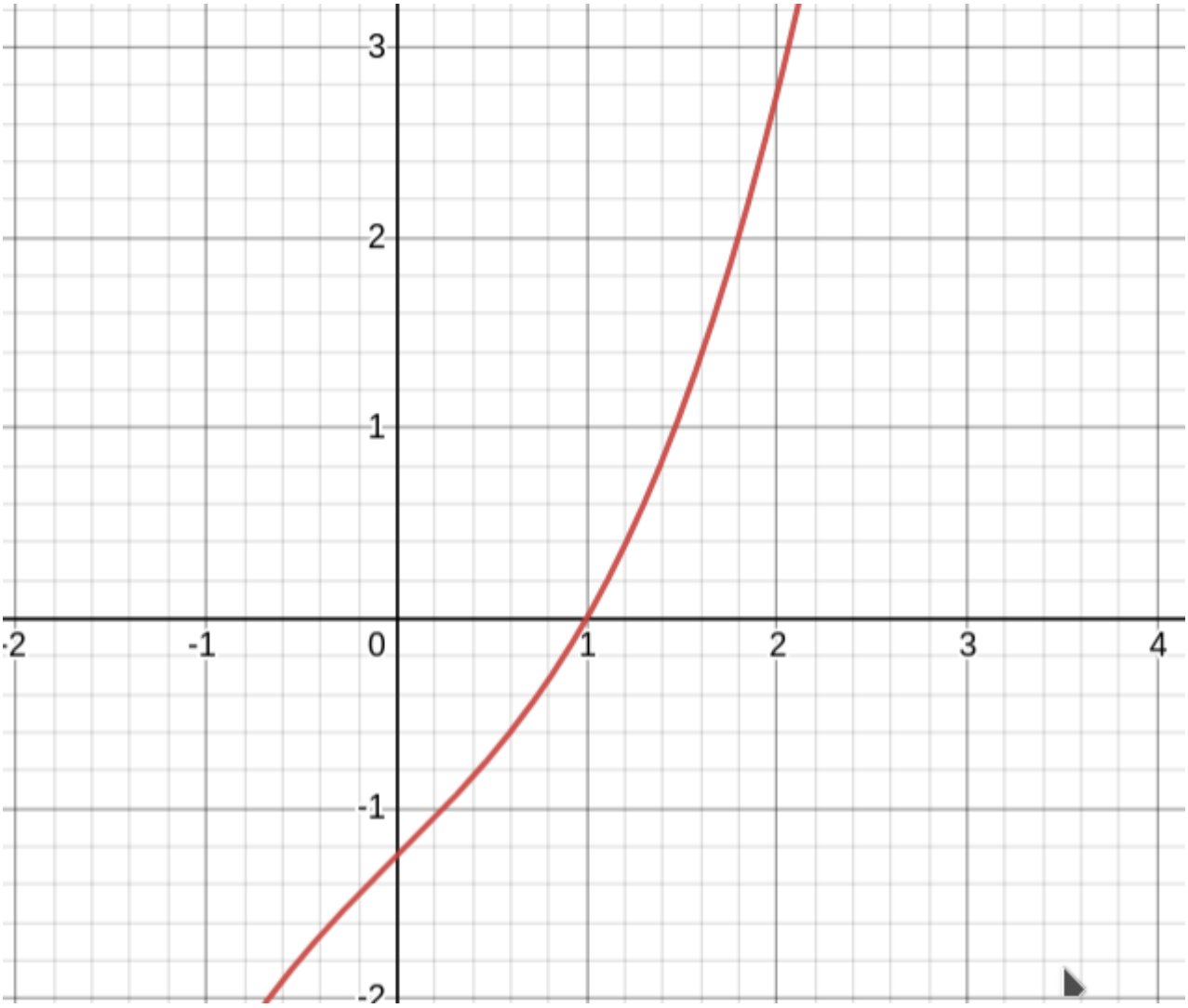
1) $x^2 + ln(x) - e * x = 0$, корень на отрезке [1.5; 3]

2) $cos(x) * ln(\frac{1}{x}) + e * x^2 - 1.5 = 0$, корень на отрезке [0.5; 0.9]

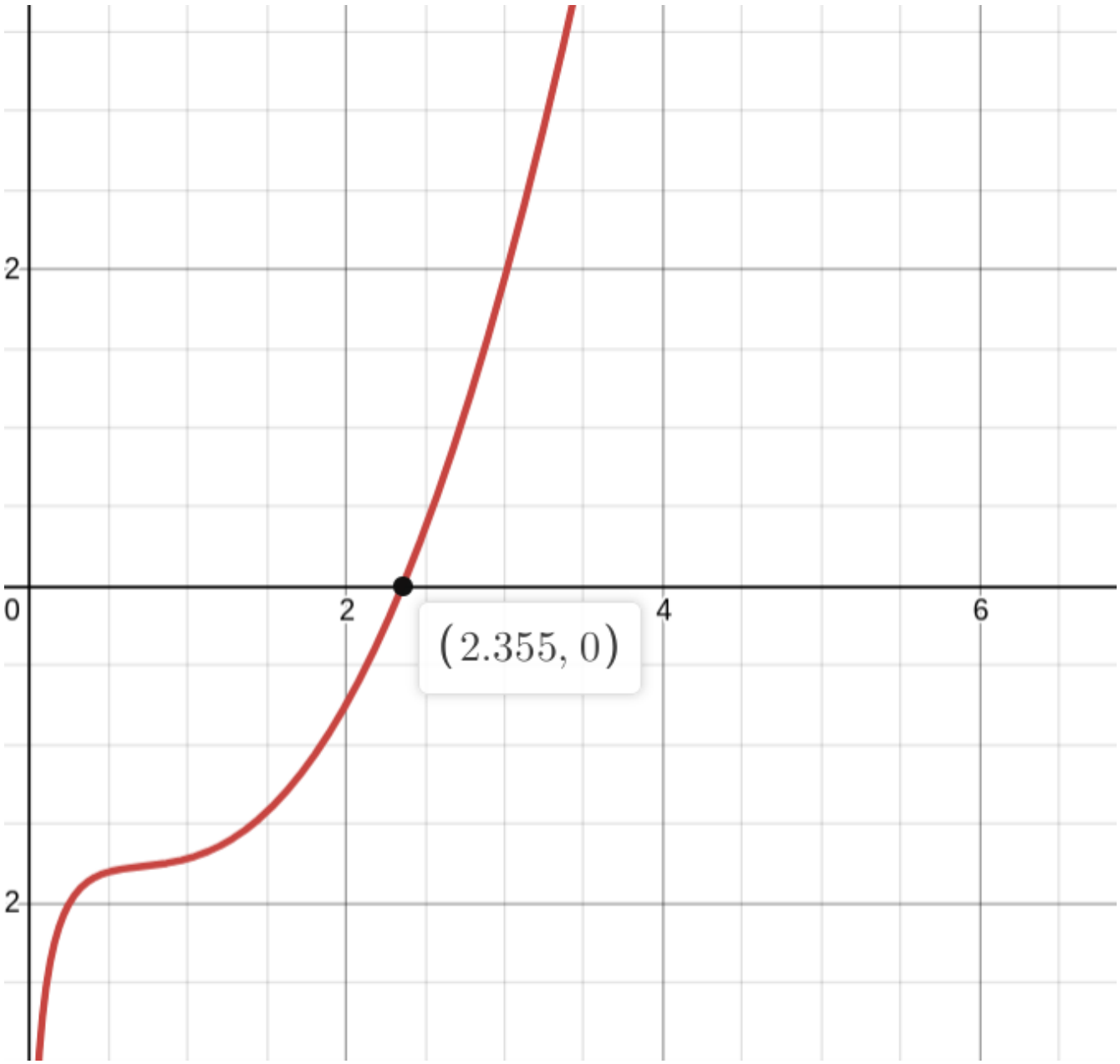
Решение

Построим графики функций:

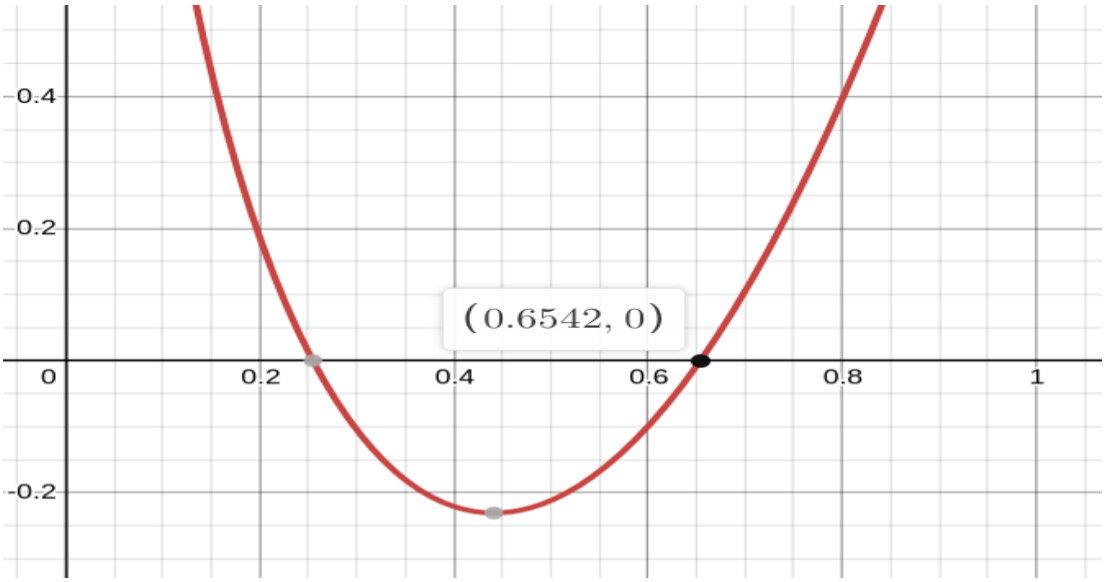
$$y = 0.25 * x^3 + x - 1.2502$$



$$y = x^2 + ln(x) - e * x$$


2
0
2
4
6
2
(2.355, 0)

$$y = cos(\mathrm{x}) * ln(\frac{1}{\mathrm{x}}) + \mathrm{e} * \mathrm{x}^2 - 1.5$$



Описание методов:

1) Дихотомия или метод деления пополам — метод вычисления корней уравнения $f(x) = 0$, основанный на пошаговом сужении промежутка, в котором находится единственный корень уравнения, пока не добиться заданной точности.

Возьмем две точки x_0 и x_1, в которых значения функции $f(x_0)$ и $f(x_1)$ имеют

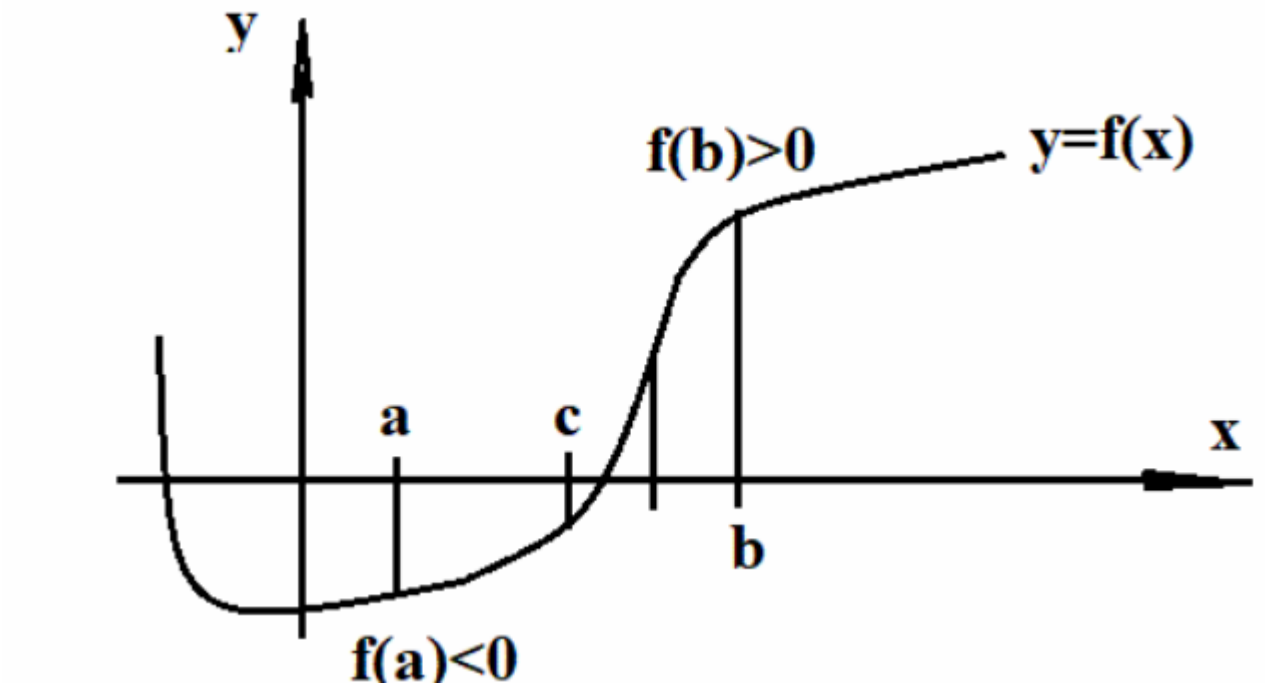
разные знаки. В этом случае между ними имеется хоть один корень функции $f(x)$.

Разделим промежуток между точками x_0 и x_1 пополам, обозначим середину отрезка

точкой x_2, которая равняется: $x_2 = \frac{x_0 + x_1}{2}$. Тогда $f(x_2) * f(x_1) >= 0$ или $f(x_2) * f(x_0) >= 0$.

Выбираем ту часть отрезка, на концах которого функция имеет разные знаки, и вновь делим полученный отрезок пополам. С каждым делением точность увеличивается вдвое. Деление «разнознакового» отрезка продолжаем, пока не сузится область нахождения корня функции, что поможет наконец найти его с большой степенью точности

Графическое представление метода:



Проверим, что уравнения принимают разные по знаку значения в концах отрезка:

$f_1(0) = -1.2502 \quad f_1(2) = 0.7498$

$f_2(1.5) \sim 2,2328878508 \quad f_2(3) \sim -0,25345777404$

$f_3(0.5) \sim -0,12730875524 \quad f_3(0.9) \sim 0,80715579864$

Условие выполняется.

2) Метод итераций:

Пусть дана функция $f(x)$ Заменим исходное уравнение $f(x)$ на эквивалентное $F(x) = x$. Выберем начальное приближение корня x_0. Тогда получим некоторое число $x_1 = F(x_0)$ Теперь подставляя вместо x_0 число x_1 получим $x_2 = F(x_1)$. Повторяя этот процесс, будем иметь последовательность чисел. Если последовательность сходящаяся, это будет происходить до тех пор, пока $|x_n - x_{n-1}| > eps$, где eps - машинное эпсилон. Функция будет возвращать x_n

Условие сходимости:
$|F'(x)| < 1$
Найдем $F_1(x),\ F_2(x),\ F_3(x)$

1. $0.25 * x^3 + x - 1.2502 = 0$

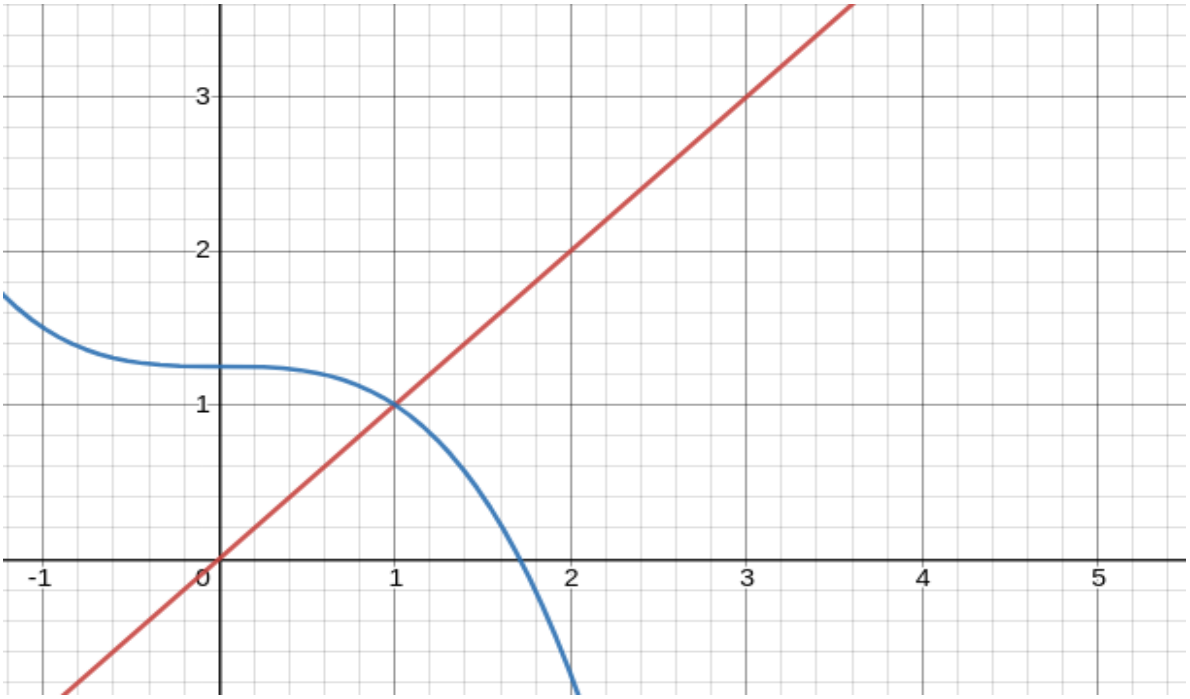
   $x = 1.2502 - 0.25 * x^3$, значит

$$F_1(x) = 1.2502 - 0.25 * x^3$$

Построим график функции $F_1(x)$

$$y_1 = 1.2502 - 0.25 * x^3$$

$$y_2 = x$$



Красная линия: $y_2 = x$

Синяя линия: $y_1 = 1.2502 - 0.25 * x^3$

Видно по графикам, что условие сходимости выполняется.

2. $x^2 + ln(x) - e * x = 0$

$$x^2 = e * x - ln(x)$$

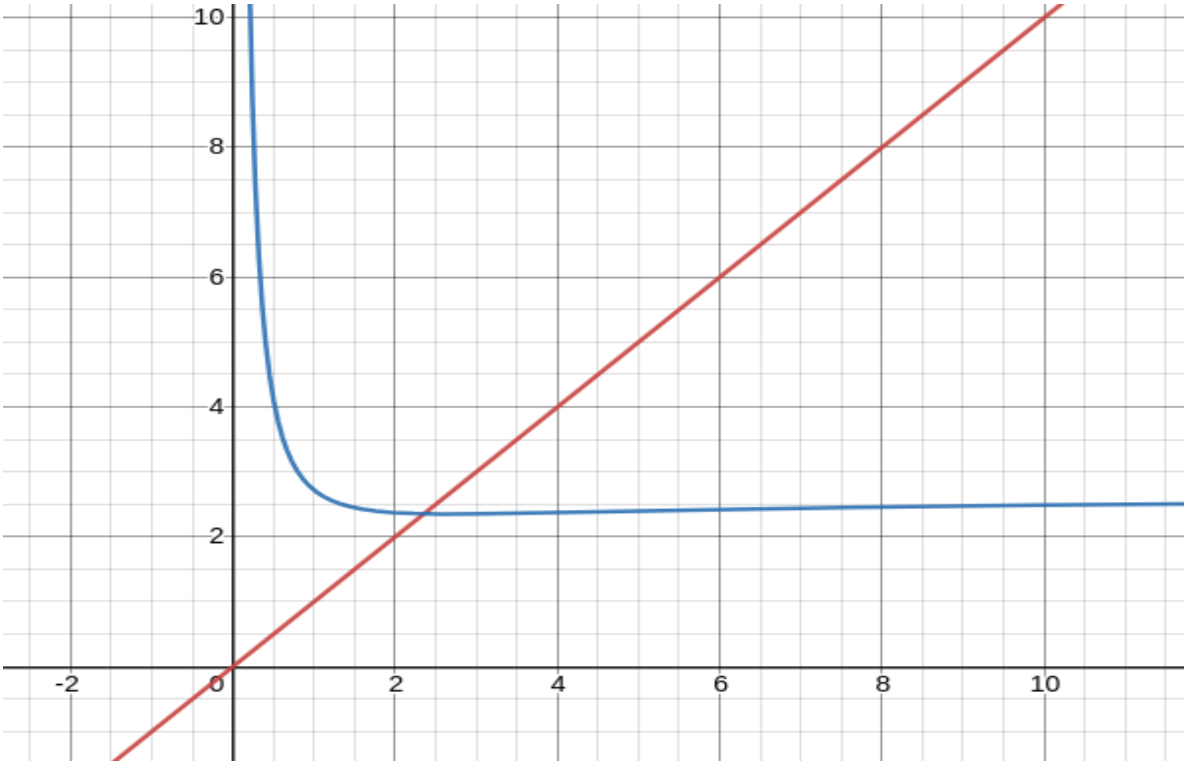
$$x = \frac{e * x - ln(x)}{x}$$

$$F_2(x) = \frac{e * x - ln(x)}{x}$$

Построим график функции $F_2(x)$

$$y_1 = \frac{e * x - ln(x)}{x}$$

$$y_2 = x$$



Красная линия: $y_2 = x$

Синяя линия: $y_1 = \frac{e * x - ln(x)}{x}$

Видно по графикам, что условие сходимости выполняется

3. $cos(x) * ln(\frac{1}{x}) + e * x^2 - 1.5 = 0$

$e * x^2 = 1.5 - cos(x) * ln(\frac{1}{x})$

$x = \frac{1.5 - cos(x) * ln(\frac{1}{x})}{x * e}$

$F_3(x) = \frac{1.5 - cos(x) * ln(\frac{1}{x})}{x * e}$

Построим график функции $F_2(x)$

$$y_1 = \frac{1.5 - cos(x) * ln(\frac{1}{x})}{x * e}$$

$$y_2 = x$$

Черная линия: $y_2 = x$

Синяя линия: $y_1 = \frac{1.5 - cos(x) * ln(\frac{1}{x})}{x * e}$

Видно по графикам, что условие сходимости выполняется(на заданном промежутке)

3) Метод Ньютона

Метод Ньютона – Суть метода состоит в разбиении отрезка на два отрезка с помощью касательной и выборе нового отрезка от точки пересечения касательной с осью абсцисс до неподвижной точки, на которой функция меняет знак и содержит решение.

Построение касательных продолжается до достижения необходимой точности. Член последовательности вычисляется таким образом:

$x = x - \frac{f(x)}{fp(x)}$, где $fp(x)$ – это производная функции.

Как в методе итераций образуется последовательность x_n, и изменения до тех пора

$|x_n - x_{n-1}| > eps$, где eps - машинное эпсилон

Вычислим производные функций:

$$Fp_1 = (f_1)' = (0.25 * x^3 + x - 1.2502)' = 0.75 * x^2 + 1$$

$$Fp_2 = (f_2)' = (x^2 + ln(x) - e * x)' = 2 * x + \frac{1}{x} - e$$

$$Fp_3 = (f_3)' = (cos(x) * ln(\frac{1}{x}) + e * x^2 - 1.5)' = -sin(x) * ln(\frac{1}{x}) - \frac{cos(x) * x}{x^2} + 2 * e * x$$

4) Метод хорд

В отличие от метода половинного деления, метод хорд предлагает, что деление рассматриваемого интервала будет выполняться не в его середине, а в точке пересечения хорды с осью абсцисс (ось - X). Следует отметить, что под хордой понимается отрезок, который проведен через точки рассматриваемой функции по концам рассматриваемого интервала. Рассматриваемый метод обеспечивает более быстрое нахождение корня, чем метод половинного деления, при условии задания одинакового рассматриваемого интервала.

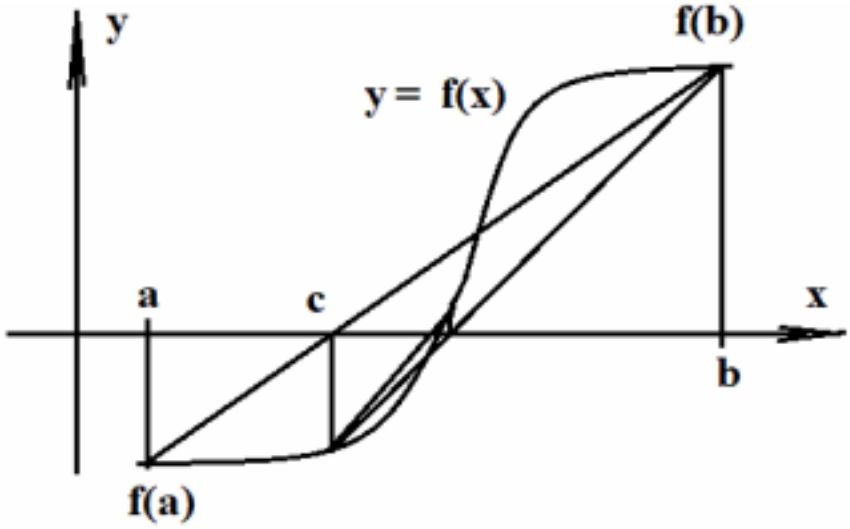
Тогда рассматриваемое c (благодаря которому будет сдвигаться граница) можно будет выразить из:

$\frac{f(b)}{b-c} = \frac{f(a)}{c-a}$ (a, b – границы, c – точка пересечения хорд)

$$c = \frac{f(a) * b - f(b) * a}{f(a) - f(b)}$$

Если $f(a) * f(b) > 0$, тогда a = c, иначе b = c;

Графическое представление метода:



Листинг программного кода

```
#include <stdio.h>

#include <cmath>

#include <locale.h>



const double eps = 0.000001;

double dichotomy(double f(double), double, double);

double iteration(double f(double), double, double);

double tangent(double f(double), double F(double), double, double);

double chord(double f(double), double, double);

double f1(double);

double F1(double);

double Fp1(double);

double f2(double);

double F2(double);
```

```
double Fp2(double);

double f3(double);

double F3(double);

double Fp3(double);



int main(void){

   setlocale(LC_ALL, "Rus");

   printf("Корень функции f1 методом деления пополам = %.5f\n",
dichotomy(f1,0.,2.));

   printf("Корень функции f1 методом итераций = %.5f\n", iteration(F1, 0.,
2.));

   printf("Корень функции f1 методом касательных = %.5f\n",
tangent(f1,Fp1,0.,2.));

   printf("Корень функции f1 методом хорд = %.5f\n\n", chord(f1,0.,2.));

   printf("Корень функции f2 методом деления пополам = %.5f\n",
dichotomy(f2,1.5,3.));

   printf("Корень функции f2 методом итераций = %.5f\n", iteration(F2, 1.5,
3.));

   printf("Корень функции f2 методом касательных = %.5f\n", tangent(f2,Fp2,1.5,
3.));

   printf("Корень функции f2 методом хорд = %.5f\n\n", chord(f2,1.5, 3.));

   printf("Корень функции f3 методом деления пополам = %.5f\n",
dichotomy(f3,0.5,0.9));

   printf("Корень функции f3 методом итераций = %.5f\n", iteration(F3, 0.5,
0.9));

   printf("Корень функции f3 методом касательных = %.5f\n", tangent(f3,Fp3,0.5,
0.9));

   printf("Корень функции f3 методом хорд = %.5f\n\n", chord(f3,0.5, 0.9));
```

```
    return 0;
}


double dichotomy(double f(double), double a, double b){
    double prevX = b, x = (a + b) / 2.;
    while (fabs(prevX - x) > eps){
        if (f(x) * f(a) > 0)
            a = x;
        else
            b = x;
        prevX = x;
        x = (a + b) / 2.;
    }
    return x;
}


double iteration(double f(double), double a, double b){
    double prevX = a, x = f(prevX);
    while (fabs(x - prevX) > eps){
        prevX = x;
        x = f(x);
    }
    return x;
}
```

```
double tangent(double f(double), double F(double), double a, double b){
   double prevX = (a + b / 2.), x = prevX - f(prevX) / F(prevX);
   while (fabs(prevX - x) > eps){
       prevX = x;
       x = prevX - f(prevX) / F(prevX);
   }
   return x;
}

double chord(double f(double), double a, double b){
   double prevX = b, ya = f(a), yb = f(b);
   double x = (ya * b - yb * a) / (ya - yb);
   while (fabs(prevX - x) > eps){
       if (ya * f(x) > 0)
           a = x;
       else
           b = x;
       ya = f(a), yb = f(b);
       prevX = x;
       x = (ya * b - yb * a) / (ya - yb);
   }
   return x;
}
```

```
double f1(double x){
   return 0.25 * x * x * x + x - 1.2502;
}

double F1(double x){
   return 1.2502 - 0.25 * x * x * x;
}

double Fp1(double x){
   return 0.75 * x * x + 1;
}


double f2(double x){
   return x * x + log(x) - exp(1.) * x;
}

double F2(double x){
   return ((exp(1.) * x - log(x)) / x);
}

double Fp2(double x){
   return (2 * x + 1 / x - exp(1.));
```

```
}


double f3(double x){
   return cos(x) * log(1./x) + exp(1.) * x * x - 1.5;
}


double F3(double x){
   return (1.5 - cos(x) * log(1./x)) / x / exp(1.);
}


double Fp3(double x){
   return -sin(x) * log(1./x) - cos(x) * x / (x * x) + 2 * exp(1.) * x;
}
```

Результат работы программы

Корень функции f1 методом деления пополам = 1.00011

Корень функции f1 методом итераций = 1.00011

Корень функции f1 методом касательных = 1.00011

Корень функции f1 методом хорд = 1.00011

Корень функции f2 методом деления пополам = 2.35458

Корень функции f2 методом итераций = 2.35458

Корень функции f2 методом касательных = 2.35458

Корень функции f2 методом хорд = 2.35458

Корень функции f3 методом деления пополам = 0.65415

Корень функции f3 методом итераций = 0.65415

Корень функции f3 методом касательных = 0.65415

Корень функции f3 методом хорд = 0.65415

Выводы

Нахождение корней трансцендентных уравнений является зачастую достаточно сложной задачей, не решаемой аналитически с помощью конечных формул. Кроме того, иногда на практике уравнение содержит коэффициенты, значения которых заданы приблизительно, так что говорить о точном решении уравнений в таких случаях вообще не имеет смысла. Поэтому задачи приближенного определения корней уравнения и соответствующей оценки их точности имеют большое значение.

В курсовом проекте были рассмотрены 4 численных метода решения трансцендентных уравнений – метод дихотомии, метод итераций, метод хорд и метод Ньютона.

Численные методы являются основным инструментом решения современных прикладных задач. Аналитическое решение той или иной задачи в виде отдельных формульных соотношений является скорее исключением, нежели правилом в силу сложного и приближенного характера исследуемых моделей. Вот почему численный анализ математических моделей является в настоящее время актуальным и наиболее эффективным аппаратом конструктивного исследования прикладных проблем.