

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ ТА  
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

**Лабораторна робота №2**  
**з дисципліни «Алгоритми та методи обчислення»**  
**на тему: «Розв'язання рівнянь з одним невідомим»**

**Варіант 21**

Виконав:  
студент 3-го курсу,  
гр. КВ-41,  
Яковенко Максим

Київ – 2016

## Завдання для лабораторної роботи

Для заданого рівняння відповідно до варіанту (табл. 2.4) виконати відокремлення коренів та обчислити значення констант  $m_i$  або  $q$  (залежно від методу уточнення коренів, що використовується). Розв'язати задане рівняння з точністю  $\varepsilon_i = \varepsilon_{(i-1)} / 10^{-3}$ ,  $i = 1, 2, \dots, 4$ ;  $\varepsilon_0 = 0.01$ . Результати подати у вигляді трьох таблиць

Для першого кореня рівняння побудувати порівняльну таблицю швидкості збігання методу ітерації та іншого методу, заданого за варіантом.

Звіт з лабораторної роботи має містити вихідний текст програми, три таблиці з результатами та висновки.

### Варіант 21:

21	$3 / (2 + \cos x) - x / 4 = 0$	I,Б
----	--------------------------------	-----

### Текст програми:

#### Result.h

```
#pragma once
```

```
struct Result
```

```
{
```

```
    double x;
```

```
    double clarRoot;
```

```
    int count;
```

```
    Result(void);
```

```
    Result(Result& source);
```

```
    Result& operator=(Result& source);
```

```
    ~Result(void);
```

```
};
```

#### methods.h

```
#pragma once
```

```
#define _USE_MATH_DEFINES
```

```
#include <cmath>
```

```
#include "Result.h"
```

```

class Method

{

private:

public:

    typedef double (*type_func)(double x);

    Method(void);

    ~Method(void);

    Result Iteration(double m, double M, double x, double eps, double(*fx)(double x));

    Result Bisection(double a, double b, double eps, double(*fx)(double x));

};

```

## methods.cpp

```

#include "methods.h"

Method::Method(void) { };

Method::~~Method(void) { };

Result Method::Iteration(double m, double M, double x, double eps, double(*fx)(double x))
{
    double x_n, lambda, q;

    int count = 0;

    Result resInf;

    lambda = 1/M;

    q = 1 - m/M;

    do {

x_n = x;

x = x_n - lambda*fx(x_n);

        ++count;

    } while(abs(x_n - x) > (1 - q)*eps/q);

    resInf.x = x;

    resInf.clarRoot = abs(x - x_n);

    resInf.count = count;

    return resInf;

}

Result Method::Bisection(double a, double b, double eps, double(*fx)(double x))

```

```

{

    double x, c;

    int count = 0;

    Result resInf;

    while(abs(b - a) >= 2*eps) {

        c = (a + b)/2;

        if((fx(a)*fx(c)) < 0)

            b = c;

        if((fx(b)*fx(c)) < 0)

            a = c;

        ++count;

    }

    x = (a + b)/2;

    resInf.x = x;

    resInf.clarRoot = (b - a) * 0.5;

    resInf.count = count;

    return resInf;

}

```

## Result.cpp

```

#include "Result.h"

Result::Result(Result& source)

{

    x = source.x;

    clarRoot = source.clarRoot;

    count = source.count;

}

Result& Result::operator=(Result& source)

{

    if (this == &source)

        return source;

}

```

```

        x = source.x;

        clarRoot = source.clarRoot;

        count = source.count;

        return source;
    }

Result::~~Result(void) {}

Result::Result(void) {}

```

## main.cpp

```

#include <iostream>

#include <stdio.h>

#include "methods.h"

#include "math.h"

#define N 4

using namespace std;

double fx(double x)
{
    return 3/(2 + cos(x)) - (x/4);
}

double minus_fx(double x)
{
    return -3/(2+cos(x))+(x/4);
}

int main()
{
    double x;

    double eps;

    double a, b;

    int i;

    Method *method = new Method();

    Result result, result2;

    cout<<"\t\tIteration Method"<<endl;

    cout<<endl;

```

```

x = 5;

cout<<"\tEps\t"<<"\t"<<" Root's value\t"<<"\tEstimation accuracy"<<endl;

for(i = 0, eps = 1e-2; i <= N; ++i, eps /= 1e3) {

    result = method->Iteration(0.7, 1.3, x, eps, minus_fx);

    printf("%.14lf \t %.15lf \t %.15lf\n", eps, result.x, result.clarRoot);

}

cout<<endl;

x = 8;

cout<<"\tEps\t"<<"\t"<<" Root's value\t"<<"\tEstimation accuracy"<<endl;

for(i = 0, eps = 1e-2; i <= N; ++i, eps /= 1e3) {

    result = method->Iteration(0.2, 1.1, x, eps, fx);

    printf("%.14lf \t %.15lf \t %.15lf\n", eps, result.x, result.clarRoot);

}

cout<<endl;

x = 9.5;

cout<<"\tEps\t"<<"\t"<<" Root's value\t"<<"\tEstimation accuracy"<<endl;

for(i = 0, eps = 1e-2; i <= N; ++i, eps /= 1e3) {

    result = method->Iteration(0.05, 1.5, x, eps, minus_fx);

    printf("%.14lf \t %.15lf \t %.15lf\n", eps, result.x, result.clarRoot);

}

cout<<endl;

cout<<"\tBisection Method"<<endl;

cout<<endl;

a = 4;

b = 6;

cout<<"\tEps\t"<<"\t"<<" Root's value\t"<<"\tEstimation accuracy"<<endl;

for(i = 0, eps = 1e-2; i <= N; ++i, eps /= 1e3) {

    result = method->Bisection(a, b, eps, fx);

    printf("%.14lf \t %.15lf \t %.15lf\n", eps, result.x, result.clarRoot);

}

cout<<endl;

```

```

a = 7;

b = 9;

cout<<"\tEps\t"<<"\t"<<" Root's value\t"<<"\tEstimation accuracy"<<endl;

for(i = 0, eps = 1e-2; i <= N; ++i, eps /= 1e3) {

    result = method->Bisection(a, b, eps, minus_fx);

    printf("%.14lf \t %.15lf \t %.15lf\n", eps, result.x, result.clarRoot);

}

cout<<endl;

a = 9;

b = 11;

cout<<"\tEps\t"<<"\t"<<" Root's value\t"<<"\tEstimation accuracy"<<endl;

for(i = 0, eps = 1e-2; i <= N; ++i, eps /= 1e3) {

    result = method->Bisection(a, b, eps, minus_fx);

    printf("%.14lf \t %.15lf \t %.15lf\n", eps, result.x, result.clarRoot);

}

cout<<endl;

cout<<"\tComparison Table "<<endl;

cout<<endl;

x = 5;

a = 4;

b = 6;

cout<<"\tEps\t"<<"\t"<<"Iteration method\t"<<"Bisection method"<<endl;

for(i = 0, eps = 1e-2; i <= N; ++i, eps /= 1e3) {

    result = method->Iteration(0.7, 1.3, x, eps, minus_fx);

    result2 = method->Bisection(a, b, eps, fx);

    printf("%.14lf \t %d \t\t %d\n", eps, result.count, result2.count);

}

return 0;

}

```

## Таблиці результатів:

Iteration Method		
Eps	Root's value	Estimation accuracy
0.0100000000000000	5.076288466164677	0.007953904021086
0.0000100000000000	5.082010287777901	0.000007819503556
0.0000000100000000	5.082015985274184	0.000000007827792
0.0000000000100000	5.082015990977792	0.000000000007836
0.000000000000001	5.082015990983502	0.000000000000008
Eps	Root's value	Estimation accuracy
0.0100000000000000	8.478380423460933	0.000285214743760
0.0000100000000000	8.478420288575240	0.000000527996772
0.0000000100000000	8.478420362322275	0.000000000976666
0.0000000000100000	8.478420362458689	0.000000000001807
0.000000000000001	8.478420362458941	0.000000000000000
Eps	Root's value	Estimation accuracy
0.0100000000000000	10.056880378461907	0.000022816319582
0.0000100000000000	10.056880455139151	0.000000076677244
0.0000000100000000	10.056880455396561	0.000000000257410
0.0000000000100000	10.056880455397428	0.000000000000002
0.000000000000001	10.056880455397428	0.000000000000000

  

Bisection Method		
Eps	Root's value	Estimation accuracy
0.0100000000000000	5.085937500000000	0.007812500000000
0.0000100000000000	5.082008361816406	0.000007629394531
0.0000000100000000	5.082015983760357	0.000000007450581
0.0000000000100000	5.082015990985383	0.000000000007276
0.000000000000001	5.082015990983514	0.000000000000007
Eps	Root's value	Estimation accuracy
0.0100000000000000	8.476562500000000	0.007812500000000
0.0000100000000000	8.478416442871094	0.000007629394531
0.0000000100000000	8.478420369327068	0.000000007450581
0.0000000000100000	8.478420362465840	0.000000000007276
0.000000000000001	8.478420362458941	0.000000000000007
Eps	Root's value	Estimation accuracy
0.0100000000000000	10.054687500000000	0.007812500000000
0.0000100000000000	10.056877136230469	0.000007629394531
0.0000000100000000	10.056880451738834	0.000000007450581
0.0000000000100000	10.056880455398641	0.000000000007276
0.000000000000001	10.056880455397426	0.000000000000007

  

Comparison Table		
Eps	Iteration method	Bisection method
0.0100000000000000	3	7
0.0000100000000000	11	17
0.0000000100000000	19	27
0.0000000000100000	27	37
0.000000000000001	35	47
Press any key to continue . . .		



## **Висновки:**

В ході виконання лабораторної роботи ми досліджували методи розв'язання рівнянь з одним невідомим.

Для знаходження коренів рівняння необхідно спочатку відокремити, тобто визначити певний відрізок, що містить в собі лише один корінь рівняння. Потім відокремленні корені уточнюються відповідними способами.

В даному варіанті методами уточнення кореня були: ітераційний метод та метод пропорційного поділу відрізка(метод бісекції).

Обидва методи уточнення коренів є досить точними, тому ми порівнювали їх за кількістю ітерацій необхідних для знаходження кореня заданої точності. Метод ітерації виявився швидшим від методу бісекції, що видно з результатів поданих у таблиці 3. Проте метод ітерації можна використовувати лише для рівнянь, придатних до ітерації, і, часто, необхідно знаходити чи певним чином зводити рівняння до придатного до ітерації, що ускладнює процес уточнення кореня.