

Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та комп'ютерних технологій

Звіт

про виконання лабораторної роботи № 4

з курсу “Алгоритмізація та програмування”

«Розв'язування нелінійних рівняння ітераційним методом Ньютона»

Виконав:

ст. гр. ФЕІ-11

Стасів Петро

Перевірив:

доц. Хвищун І.О.

Львів-2021

Звіт

Мета: у середовищі Delphi написати програму яка буде шукати корінь нелінійного рівняння ітераційним методом Ньютона.

Виконання лабораторної роботи:

1) Описуємо функції які будуть використані в програмі:

```
function GetFunctionRes(x : double) : double;
begin
    GetFunctionRes := 3 * x - 4 * Ln(x) - 5;
end;

function GetFunctionDerivative(x : double) : double;
begin
    const d = Eps / 1000.0;

    GetFunctionDerivative := (GetFunctionRes(x + d) -
        GetFunctionRes(x)) / d;
end;

function GetFunctionDerivative2(x : double) : double;
begin
    const d = Eps / 1000.0;

    GetFunctionDerivative2 := (GetFunctionDerivative(x + d) -
        GetFunctionDerivative(x)) / d;
end;
```

2) Цикл обрахунку кореня

```
var iterations := 0;

while True do

begin

    if iterations > MaxIterations then

        break;

    var d := GetFunctionRes(x) / GetFunctionDerivative(x);

    x := x - d;

    if Abs(d) <= Eps then

        break;

    iterations := iterations + 1;

end;
```

Тестування:

```
Enter interval:
2 4
Enter result tolerance:
1e-3
Approximate root is: 3.22995944378425E+0000 found in 2 iterations

Enter interval:
2 4
Enter result tolerance:
1e-7
Approximate root is: 3.22995943972793E+0000 found in 4 iterations

Enter interval:
2 10
Enter result tolerance:
1e-8
Approximate root is: 3.22995943972793E+0000 found in 4 iterations
```

Текст програми:

uses

System.SysUtils;

const MaxIterations = 1024;

var Eps : **double** = 1e-5;

function GetFunctionRes(x : **double**) : **double**;

begin

GetFunctionRes := 3 * x - 4 * Ln(x) - 5;

end;

function GetFunctionDerivative(x : **double**) : **double**;

begin

const d = Eps / 1000.0;

GetFunctionDerivative := (GetFunctionRes(x + d) -
GetFunctionRes(x)) / d;

end;

function GetFunctionDerivative2(x : **double**) : **double**;

begin

const d = Eps / 1000.0;

GetFunctionDerivative2 := (GetFunctionDerivative(x + d) -
GetFunctionDerivative(x)) / d;

```

end;

begin

    var a := 0.0;

    var b := 0.0;

    while True do

        begin

            Writeln('Enter interval: ');

            Readln(a, b);

            Writeln('Enter result tolerance: ');

            Readln(Eps);

            var x := a;

            if GetFunctionRes(x) * GetFunctionDerivative2(x) < 0.0
then
                begin

                    x := b;

                    if GetFunctionRes(x) * GetFunctionDerivative2(x) < 0.0
then

                        Writeln('For the specified interval result is not
guaranteed!');

                    end;

                end;

```

```

var iterations := 0;

while True do
begin
    if iterations > MaxIterations then
        break;

    var d := GetFunctionRes(x) / GetFunctionDerivative(x);

    x := x - d;

    if Abs(d) <= Eps then
        break;

    iterations := iterations + 1;

end;

Writeln('Approximate root is: ', x, ' found in ',
iterations, ' iterations', #13#10);

end;

end.

```

Додаткове завдання, програма на C++:

```

#include <iostream>

#include <cmath>

```

```
constexpr int MaxIterations = 1024;

double Eps = 1e-5;

double GetFunctionRes(const double x)
{
    return 3 * x - 4 * std::log(x) - 5;
}

double GetFunctionDerivative(const double x)
{
    const double d = Eps / 1000.0;
    return (GetFunctionRes(x + d) - GetFunctionRes(x)) / d;
}

double GetFunctionDerivative2(const double x)
{
    const double d = Eps / 1000.0;
    return (GetFunctionDerivative(x + d) -
            GetFunctionDerivative(x)) / d;
}

int main()
{
    double a = 0.0;
    double b = 0.0;
```

```

while (1)
{
    std::cout << "\nEnter interval: ";

    std::cin >> a >> b;

    std::cout << "\nEnter result tolerance: ";

    std::cin >> Eps;

    double x = a;

    if (GetFunctionRes(x) * GetFunctionDerivative2(x)
< 0.0)
    {
        x = b;

        if (GetFunctionRes(x) *
GetFunctionDerivative2(x) < 0.0)

            std::cout << "\nFor the specified
interval result is not guaranteed!";

    }

    int iterations = 0;

    while(1)
    {

```



```

        if (iterations > MaxIterations)

            break;

        double d = GetFunctionRes(x) /
GetFunctionDerivative(x);

        x = x - d;

        if (std::abs(d) <= Eps)

            break;

        ++iterations;

    }

    std::cout << "\nApproximate root is: " << x << "
found in: " << iterations << " iterations\n\n";

}

return(0);

}

```

Код лабораторної: <https://github.com/ptrstasiv/Lab4.git>

Висновок: при виконанні даної лабораторної роботи ми ознайомилися з особливостями чисельного розв'язання нелінійних рівнянь ітераційним методом Ньютона. Реалізувавши цей алгоритм в середовищі Delphi.