Звіт з предмету Чисельні методи
Пустовіта Дмитра
КА-32

1. Умова Задачі

Знайти всі дійсні корені рівняння $f(x) = 2x^3 - 4x^2 - x + 2$

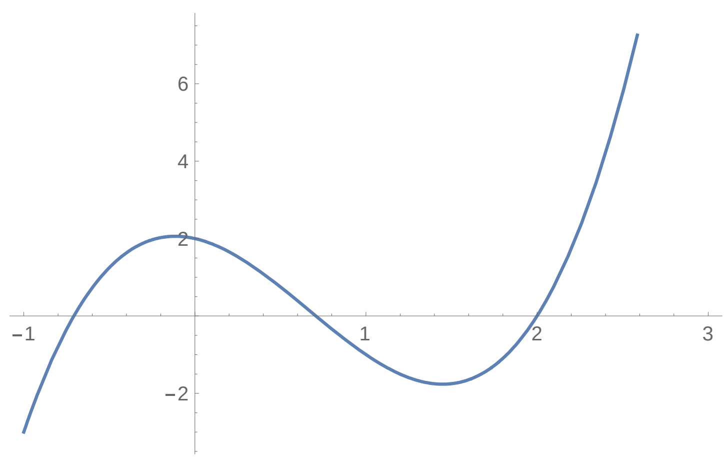
2. Математичне розв'язання задачі

Маємо рівняння $f(x) = 2x^3 - 4x^2 - x + 2$
Знайдемо спочатку раціональні розв'язки рівняння з допомогою схеми Горнера. Шукатимемо Серед дільників вільного члена(± 1 та ± 2)

	2	-4	-1	2
2	2	0	-1	0

Отримали що 2 є коренем f(x) та $f(x) = (x-2)(2x^2-1)$, тобто функція f(x) має 3 дійсні корені (більше мати вона не може)

$$\begin{cases} x_1 = -\frac{\sqrt{2}}{2} \\ x_2 = \frac{\sqrt{2}}{2} \\ x_3 = 2 \end{cases}$$



Так як $0 < \sqrt{2} < 2$ маємо інтервали :

$$\begin{cases} -1 < x_1 < 0 \\ 0 < x_2 < 1 \\ 1 < x_3 < 3 \end{cases}$$

3. **Результати роботи** (для кореня $-\frac{\sqrt{2}}{2}$)

```
bisection method:
iteration 1
a=-1 ,b=0
f(a)=-3 ,f(b)=2
iteration 2
a=-1 ,b=-0.5
f(a)=-3 ,f(b)=1.25
iteration 3
a=-0.75 ,b=-0.5
f(a)=-0.34375 ,f(b)=1.25
iteration 4
a=-0.75 ,b=-0.625
f(a)=-0.34375 ,f(b)=0.574219
iteration 5
a=-0.75 ,b=-0.6875
f(a)=-0.34375 ,f(b)=0.146973
.
.
.
iteration 13
a=-0.707275 ,b=-0.707031
f(a)=-0.00129125
,f(b)=0.000578284
iteration 14
a=-0.707153 ,b=-0.707031
f(a)=-
0.000356361
,f(b)=0.00057828
4
iteration 15
a=-0.707153 ,b=-
0.707092
f(a)=-
0.000356361
,f(b)=0.00011099
2
iteration 16
```

```
a=-0.707123 ,b=-0.707092
f(a)=-0.000122677
,f(b)=0.000110992
```
root is -0.707108

chord method:

```
iteration 1
a=-1 ,b=0
f(a)=-3 ,f(b)=2
iteration 2
a=-1 ,b=-0.6
f(a)=-3 ,f(b)=0.728
iteration 3
a=-0.921888 ,b=-0.6
f(a)=-2.04461 ,f(b)=0.728
iteration 4
a=-0.837371 ,b=-0.6
f(a)=-1.1417 ,f(b)=0.728
.
.
.
iteration 25
a=-0.707118 ,b=-0.707066
f(a)=-8.25134e-005
,f(b)=0.000316
iteration 26
a=-0.707118 ,b=-0.707076
f(a)=-8.25134e-005
,f(b)=0.000233491
iteration 27
a=-0.707118 ,b=-0.707087
f(a)=-8.25134e-005
,f(b)=0.00015098
iteration 28
a=-0.707118 ,b=-0.707098
f(a)=-8.25134e-005
,f(b)=6.84682e-005
```
root is -0.707109

Newton method:

```
iteration 1
x(1)= -0.5
f(X1)=1.25
iteration 2
x(2)= -0.777778
f(X2)=-0.58299
iteration 3
x(3)= -0.711917
f(X3)=-0.0370216
iteration 4
x(4)= -0.707131
f(X4)=-
0.000189202
```
**root is -
0.707107**

4.Код програми

```
// NonLineEq.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "fstream"
#include "string"
#include "math.h"
#include "iostream"

#define epsilon 0.00001
static int a0, a1, a2, a3;

using namespace std;
ofstream fout("out.txt");

void error(int n){
	switch (n)
	{
	case 1: fout << "error, a>b ";
			break;
	case 2:
		fout << "error, interval is so small";
		break;
	case 3:
		fout << "error, root is not in interval";
		break;
	case 4: fout << "df / dx is small, newton is not working";

	default: fout << "unckown error ";
		break;
	}
	exit(n);
}
```

```
double func(double x)
{
	return a3 * pow(x, 3) + a2 * pow(x, 2) + a1*x + a0;
}

double dfunc(double x){
	return a3 * 3 * pow(x, 2) + 2 * a2*x + a1;
}

void correct(int a, int b, double eps){
	if (a > b) error(1);
	if (b - a < eps) error(2);
	if (func(a)*func(b) > 0) error(3);
}

double Chord(double a, double b, double eps, double f(double)){
	double d, c = (a*f(a) - b*f(b)) / (f(a) - f(b));
	int iter=0;
	fout << endl << "chord method: " << endl;
	while ((fabs(f(c)) > eps)){
		iter++;
		fout << "iteration " << iter << endl;
		fout << "a=" << a << " ,b=" << b << endl;
		fout << "f(a)=" << f(a) << " ,f(b)=" << f(b) << endl;
		if (f(a)*f(c) < 0)
			b = c;
		else a = c;
	 d = (a*f(a) - b*f(b)) / (f(a) - f(b));
		if (fabs(d - c) < eps) return d;
		c = d;
	}
	return c;
}
double Chord(double a, double b, double eps, double f(double)){
	double c;
	int iter=0;
	fout << endl << "chord method: " << endl;
	while (fabs(a - b) > eps){
		iter++;
		c = (a*f(a) - b*f(b)) / (f(a) - f(b));
		fout << "iteration " << iter << endl;
		fout << "a=" << a << " ,b=" << b << endl;
		fout << "f(a)=" << f(a) << " ,f(b)=" << f(b) << endl;
		if (f(a)*f(c) < 0)
			b = c;
		else if (fabs(f(c)) < eps)
			return c;
		else a = c;
	}
	return a;
}

double Newton(double a, double b, double eps, double f(double), double df(double)){
	double d,c = (a + b) / 2;
	fout << endl << "Newton method: " << endl;
	int iter=0;
	while (fabs(f(c)) > eps){
		iter++;
		if (fabs(df(c)) < eps) error(4);
		fout << "iteration " << iter << endl;
		fout << "x(" << iter << ")= " << c << endl;
		fout << "f(X" << iter << ")=" << f(c) << endl;

		d = c - f(c) / df(c);
```

```
			if (fabs(d - c) < eps) return d;
			c = d;
		}
		return c;
}

int _tmain(int argc, _TCHAR* argv[])
{
		int a, b;
		double root;
		ifstream fin("in.txt");
		fin >> a3 >> a2 >> a1 >> a0 >> a >> b;
		correct(a,b,epsilon);

		fout << "root is " << Bisec(a, b, epsilon, func)<< endl;
		fout << "root is " << Chord(a, b, epsilon, func)<< endl;
		fout << "root is " << Newton(a, b, epsilon, func, dfunc) << endl;
		fin.close();
		fout.close();
		system("Pause");
		return 0;
}
```

4. Висновок

Для кореня $-\frac{\sqrt{2}}{2}$ методом Н'ютона вдалося досягти кореня за найменшу кількість ітерацій(4), Методом Хорд за найбільшу – 28. Метод бісекцій зробив 16 ітерацій. Для всіх коренів складено порівняльну таблицю.

Корінь	Кількість ітерацій		
	Методом бісекції	Методом Хорд	Методом Н'ютона
$-\frac{\sqrt{2}}{2}$	16	28	4
$\frac{\sqrt{2}}{2}$	17	21	3
2	1	46	1