ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

ПРИБЛИЖЕННЫЕ МЕТОДЫ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

(Вариант 8)

Выполнил студент 3 курса ПМиИ
Ковшов Максим

Постановка задачи:Исследовать функцию $f(x) = 2\sin(3x) - 1.5x$ и решить уравнение $f(x) = 0$.

I. Найти промежуток, содержащий наименьший положительный корень уравнения $f(x) = 0$, для которого выполняются достаточные условия сходимости одного из итерационных методов;

II. Получить приближенное решение (с точностью 10^{-7}) методами:

1) *методом Ньютона (метод касательных)*

$$x_0 = a, \quad x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad f(a)f''(a) > 0;$$

2) *методом хорд*

$$x_0 = a, \quad x_{k+1} = x_k - \frac{f(x_k)}{f(b) - f(x_k)}(b - x_k), \quad f(b)f''(b) > 0;$$

3) *методом секущих*

$$x_0, x_1 \in [a,b], \quad x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(x_{k-1})}(x_k - x_{k-1});$$

4) *конечноразностным методом Ньютона*

$$x_0 \in [a,b], \quad x_{k+1} = x_k - \frac{h \cdot f(x_k)}{f(x_k + h) - f(x_k)}, \quad h > 0 \text{ — малый параметр;}$$

5) *методом Стеффенсена*

$$x_0 \in [a,b], \quad x_{k+1} = x_k - \frac{f^2(x_k)}{f\left(x_k + f(x_k)\right) - f(x_k)};$$

6) *методом простых итераций*

$$x_0 \in [a,b], \quad x_{k+1} = x_k - \tau f(x_k),$$

Если $f'(x) > 0$, то $0 < \tau < \dfrac{2}{\min\left(f'(x)\right)}$.

Для оценки погрешности приближенного решения, полученного любым методом, может использоваться неравенство

$$\left|x_k - x^*\right| < \frac{\left|f(x_k)\right|}{m}, \quad m = \min_{[a,b]}\left|f'(x)\right|.$$

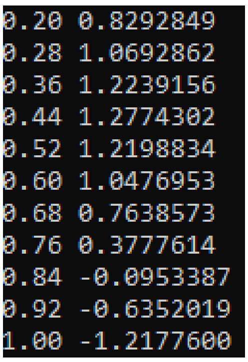
Результаты расчетов

$a = 0{,}2\ ;\ b = 1\ ;\ n = 10$

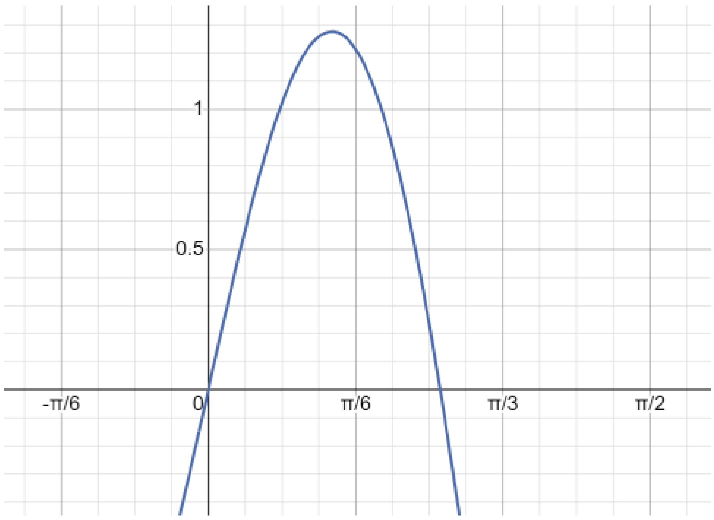
Таблица значений функции (см. программу 1 в приложении)

x	$f(x)$
0.2	0.8292849
0.28	1.0692862
0.36	1.2239156
0.44	1.2774302
0.52	1.2198834
0.60	1.0476953
0.68	0.7638573
0.76	0.3777614
0.84	-0.0953387
0.92	-0.6352019
1.0	-1.2177600

График функции



```
0.20 0.8292849
0.28 1.0692862
0.36 1.2239156
0.44 1.2774302
0.52 1.2198834
0.60 1.0476953
0.68 0.7638573
0.76 0.3777614
0.84 -0.0953387
0.92 -0.6352019
1.00 -1.2177600
```



Построив график функции, определяем, что уравнение имеет корень, который находится в интервале $0{,}8 < x < 1$

Уточним значение корня с требуемой точностью 10^{-7}, пользуясь методами 1–6.

<u>Метод Ньютона (метод касательных)</u>. Для корректного использования данного метода необходимо определить поведение второй производных функции $f(x)$ на интервале уточнения корня и правильно выбрать начальное приближение x_0.

Для функции *f(x)* имеем: *f'(x)*=$6\cos(3x) - 1{,}5$, *f''(x)=-18sin(3x)*. Видим, что вторая производная отрицательна во всей области определения функции, поэтому в качестве начального приближения можно взять правую границу интервала, т.е. $x_0 = 0{,}8$ Дальнейшие вычисления проводятся по формуле $x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_k)}$. Итерации завершаются при выполнении условия $|x_{k+1} - x_k| < \varepsilon$.

```
Metod Kasatelnih
0 0.82547555
1 0.82485927
2 0.82485893
3 0.82485893
```

<u>Метод хорд</u>. Вычисления проводятся по формуле $x_{k+1} = x_k - \dfrac{f(x_k)}{f(b) - f(x_k)}(b - x_k)$.

Итерации завершаются при выполнении условия $|x_{k+1} - x_k| < \varepsilon$.

```
Metod Chord
0 0.82205419
1 0.82455879
2 0.82482701
3 0.82485554
4 0.82485857
5 0.82485889
6 0.82485893
```

<u>Метод секущих.</u> В качестве начальных точек зададим: $x_0 = 0{,}8$ и $x_0 = 1$. Дальнейшие вычисления проводятся по формуле $x_{k+1} = x_k - \dfrac{f(x_k)}{f(x_k) - f(x_{k-1})}(x_k - x_{k-1})$. Итерации завершаются при выполнении условия $|x_{k+1} - x_k| < \varepsilon$.

```
Metod Secant
0 0.82205419
1 0.82455879
2 0.82485969
3 0.82485893
4 0.82485893
```

<u>Конечноразностный метод Ньютона.</u> В качестве начального приближения берем $x_0 = 0{,}8$. Выбираем параметр $h = 0{,}05 > 0$. Вычисления проводятся по формуле $x_{k+1} = x_k - \dfrac{f^2(x_k)}{f\left(x_k + f(x_k)\right) - f(x_k)}$.

```
Metod Newtona Konechnoraznostni
0 0.82429915
1 0.82483663
2 0.82485803
3 0.82485889
4 0.82485893
```

<u>Метод Стеффенсена.</u> В качестве начального приближения берем $x_0 = 0{,}8$. Вычисления проводятся по формуле $x_{k+1} = x_k - \dfrac{f^2(x_k)}{f\left(x_k + f(x_k)\right) - f(x_k)}$.

```
Metod Steffensena
0 0.82260503
1 0.82483573
2 0.82485893
3 0.82485893
```

<u>Метод простых итераций.</u> Выбираем $x_0 = 0{,}8$. Вычисления проводятся по формуле $x_{k+1} = x_k - \tau f(x_k)$. Выбираем $\tau = -0{,}2$.

```
Metod SimpleItteration
0 0.83018527
1 0.82353429
2 0.82517860
3 0.82478120
4 0.82487780
5 0.82485435
6 0.82486004
7 0.82485866
8 0.82485899
9 0.82485891
```

Итоговая таблица

Метод решения	Выбранный интервал $[a,b]$	Полученное решение	Количество итераций
1. Метод Ньютона (метод касательных)	[0,8 ; 1]	0,82485893	4
2. Метод хорд	[0,8 ; 1]	0,82485893	7
3. Метод секущих	[0,8 ; 1]	0,82485893	5
4. Конечноразностный метод Ньютона	[0,8 ; 1]	0,82485893	5
5. Метод Стеффенсена	[0,8 ; 1]	0,82485893	5
6. Метод простых итераций	[0,8 ; 1]	0,82485891	10

Выводы: Скорости сходимости всех методов для данной функции примерно одинаковы, кроме метода простых итераций – у него она хуже.

Приближенным решением уравнения является $x \approx$0,8248589
Все исходные тексты программ приводятся в Приложении.

ПРИЛОЖЕНИЕ

Программы нахождения корня всеми способами и построения таблицы значений функции

```cpp
#include <iostream>
#include <functional>
#include <cmath>
#include <iomanip>
#define ACCURACY pow(10, -7)

double function(double x)
{
    return (2 * std::sin(3 * x) - 1.5 * x);
}

double derivative1(double x)
{
    return (6 * std::cos(3*x) - 1.5);
}

double derivative2(double x)
{
    return (-18 * std::sin(3 * x));
}

void createStartTable()
{
    double startX = 0.2;
    double endX = 1.0;
    double step = 0.08;

    for (double x = startX; x <= endX; x += step)
    {
        double result = function(x);
        std::cout << std::fixed << std::setprecision(2) << x << " ";
        std::cout << std::fixed << std::setprecision(7) << result << "\n";
    }
}

```

```cpp
void metodKasateln()
{
    std::cout << "Metod Kasatelnih" << std::endl;
    double x0 = 0.8;
    for (int i = 0; true; i++)
    {
        double x1 = x0 - function(x0) / derivative1(x0);
        if (std::abs(x1 - x0) < ACCURACY)
        {
            std::cout << i << " " << std::fixed << std::setprecision(8) << x1 << "\n";
            break;
        }
        std::cout << i << " " << std::fixed << std::setprecision(8) << x1 << "\n";
        x0 = x1;
    }
}

void metodChord()
{
    std::cout << "Metod Chord" << std::endl;
    double a = 0.8;
    double b = 1;
    double x0 = a;
    for (int i = 0; true; i++)
    {
        double x1 = x0 - function(x0) * (b - x0) / (function(b) - function(x0));
        if (std::abs(x1 - x0) < ACCURACY)
        {
            std::cout << i << " " << std::fixed << std::setprecision(8) << x1 << "\n";
            break;
        }
        std::cout << i << " " << std::fixed << std::setprecision(8) << x1 << "\n";
        x0 = x1;
    }
}

void metodSecant()
{
    std::cout << "Metod Secant" << std::endl;
    double x0 = 0.8;
    double x1 = 1;
    for (int i = 0; true; i++)
    {
        double x2 = x1 - function(x1) * (x1 - x0) / (function(x1) - function(x0));
        if (std::abs(x2 - x1) < ACCURACY)
        {
            std::cout << i << " " << std::fixed << std::setprecision(8) << x2 << "\n";
            break;
        }
        std::cout << i << " " << std::fixed << std::setprecision(8) << x2 << "\n";
        x0 = x1;
        x1 = x2;
    }
}

void metodNewton()
{
    std::cout << "Metod Newtona Konechnoraznostni" << std::endl;
    double x0 = 0.8;
    double h = 0.05;
    for (int i = 0; true; i++)
    {
        double x1 = x0 - function(x0) * h / (function(x0 + h) - function(x0));
        if (std::abs(x1 - x0) < ACCURACY)
        {
            std::cout << i << " " << std::fixed << std::setprecision(8) << x1 << "\n";
            break;
        }
        std::cout << i << " " << std::fixed << std::setprecision(8) << x1 << "\n";
        x0 = x1;
    }
}
```

```cpp
void metodSteffensena()
{
    std::cout << "Metod Steffensena" << std::endl;
    double x0 = 0.8;
    for (int i = 0; true; i++)
    {
        double x1 = x0 - std::pow(function(x0),2) / (function(x0+function(x0)) - function(x0));
        if (std::abs(x1 - x0) < ACCURACY)
        {
            std::cout << i << " " << std::fixed << std::setprecision(8) << x1 << "\n";
            break;
        }
        std::cout << i << " " << std::fixed << std::setprecision(8) << x1 << "\n";
        x0 = x1;
    }
}

void metodSimpleItteration()
{
    std::cout << "Metod SimpleItteration" << std::endl;
    double x0 = 0.8;
    double t = -0.2;
    for (int i = 0; true; i++)
    {
        double x1 = x0 - function(x0) * t;
        if (std::abs(x1 - x0) < ACCURACY)
        {
            std::cout << i << " " << std::fixed << std::setprecision(8) << x1 << "\n";
            break;
        }
        std::cout << i << " " << std::fixed << std::setprecision(8) << x1 << "\n";
        x0 = x1;
    }
}
int main()
{
    createStartTable();
    metodKasateln();
    metodChord();
    metodSecant();
    metodNewton();
    metodSteffensena();
    metodSimpleItteration();
}
```