Национальный исследовательский университет ИТМО
Факультет ПИиКТ

Лабораторная работа №2

по курсу «*Вычислительная математика*»
Вариант «2бв»

Работу выполнил:
Глушков Даниил Григорьевич
Группа P3233

Преподаватель:
Перл О. В.

г. Санкт-Петербург
2022

1 Описание метода

1.1 Методы решения нелинейных уравнений

1.1.1 Метод хорд

Метод хорд состоит в многократном предположении корня уравнения, построения хорды к точке функции от предположенного значения и предположения корня в точке пересечения хорды с осью. Это выполняется до тех пор, пока не будет достигнута нужная точность.

1.1.2 Метод касательных

Метод касательных состоит в многократном предположении корня уравнения, построения касательной к функции в этой точке, нахождения пересечения касательной с осью ox и предположении корня в точке пересечения. Это выполняется до тех пор, пока не будет достигнута нужная точность.

1.2 Метод решения СНАУ: метод простой итерации

Метод начинается с приведения системы уравнений к виду:

$$y = \phi_1(x, y)$$

$$x = \phi_2(x, y)$$

После этого происходит предположение корней системы и вычисление с помощью этих корней нового предположения до тех пор, пока не будет достигнута нужная точность.

2 Расчетные формулы

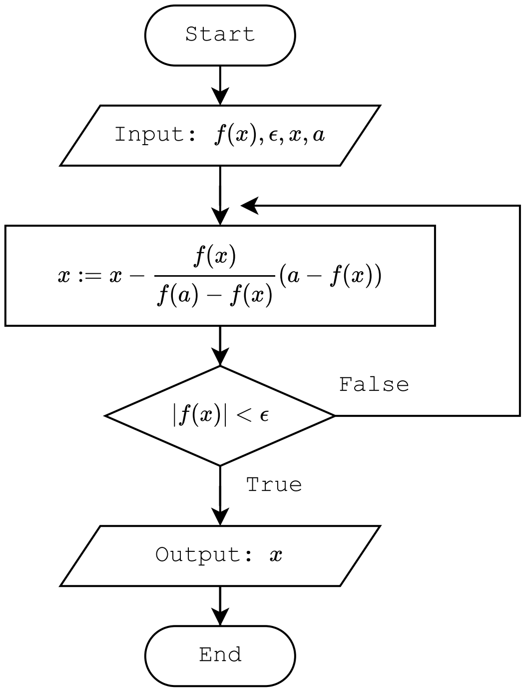
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f(a) - f(x_n)}(a - f(x_n))$$

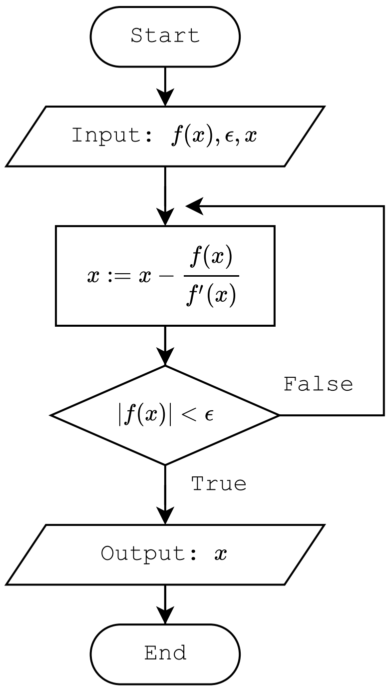
$$y_{n+1} = \phi_1(x_n, y_n)$$
$$x_{n+1} = \phi_2(x_n, y_n)$$

3 Блок-схема

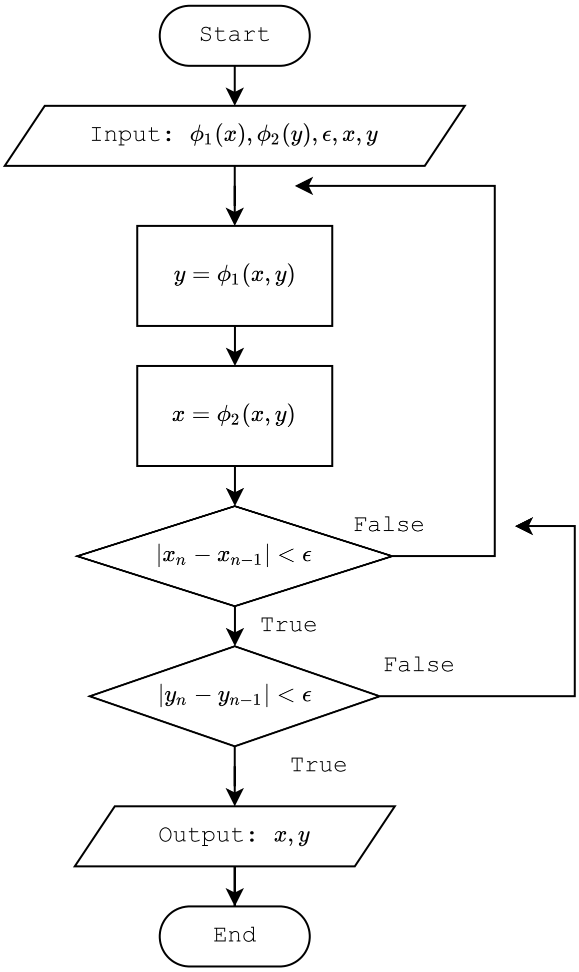
3.1 Метод хорд



3.2 Метод касательных



3.3 Метод простой итерации



4 Листинг реализации

Полный код лабораторной доступен на:
https://github.com/mrcat-pixel/compmath2.git

4.1 Метод хорд

```
public double calculate(Func function, double precision,
double begin, double end, int iterationCount) {
    double newBegin = begin - (function.calcFunc(begin)/(function.calcFunc(end)
        - function.calcFunc(begin)) * (end - begin));
    double diff = Math.abs(function.calcFunc(newBegin));

    if ( diff < precision || iterationCount > 50 ) return newBegin;
    return calculate(function, precision, newBegin, end, iterationCount + 1);
}
```

4.2 Метод касательных

```
public double calculate(@NotNull Func function, double precision, double approx,
int iterationCount) {
    double newApprox = approx - ( function.calcFunc(approx)/function.calcDer(approx) );
    double diff = Math.abs(function.calcFunc(newApprox));

    if ( diff < precision || iterationCount > 50 ) return newApprox;
    return calculate(function, precision, newApprox, iterationCount + 1);
}
```

4.3 Метод простой итерации

```
public AnswerPair calculate(EqSystem system, double precision,
double x, double y, int iterationCount) {
    double newX = system.x1(y);
    double newY = system.y2(x);

    double diffX = Math.abs(newX - x);
    double diffY = Math.abs(newY - y);

    if ((diffX < precision && diffY < precision) || iterationCount > 50)
        return new AnswerPair(newX, newY);

    return calculate(system, precision, newX, newY, iterationCount + 1);

}
```

5 Результаты работы программы

5.1 Пример 1: Система уравнений

```
complab2>syst
----------------------------------------
Choose an equation system to solve:
1: y = x^3;
   y = x^2 - 6.

2: y = 0.1x^3;
   y = x^2 - 0.5.
----------------------------------------
>1
Input precision:
>0.01
Input the initial approximation for x:
>1
Input the initial approximation for y:
>1
----------------------------------------
Starting the simple iteration method...
The solution is: x = -1,534810; y = -3,615462
complab2>
```

5.2 Пример 2: Уравнение 1

```
complab2>sing
----------------------------------------
Choose an equation to solve:
1: x + cos(x) - 0.67x^3 - 1 = 0
2: x^3 + 2x^2 - 5x - 5 = 0
3: -x^2 + 5 = 0
----------------------------------------
>1
Input precision:
>0.001
Input the initial approximation for Newton's method:
>1
Input the beginning of the interval for the Secant method:
>0.5
Input the end of the interval for the Secant method:
>1
----------------------------------------
Starting Newton's method...
The solution is: 0,922801
----------------------------------------
Starting the Secant method...
The solution is: 0,922642
----------------------------------------
The difference between the two solutions is: 0,000160
complab2>
```

5.3 Пример 2: Уравнение 2

```
complab2>sing
----------------------------------------
Choose an equation to solve:
1: x + cos(x) - 0.67x^3 - 1 = 0
2: x^3 + 2x^2 - 5x - 5 = 0
3: -x^2 + 5 = 0
----------------------------------------
>2
Input precision:
>0.001
Input the initial approximation for Newton's method:
>2
Input the beginning of the interval for the Secant method:
>1.5
Input the end of the interval for the Secant method:
>2
----------------------------------------
Starting Newton's method...
The solution is: 1,930805
----------------------------------------
Starting the Secant method...
The solution is: 1,930773
----------------------------------------
The difference between the two solutions is: 0,000032
complab2>
```

6 Вывод

1. Метод касательных хорошо применим для повышения точности корня уравнения, полученного другим методом. При этом он имеет значимые минусы. В частности, из-за использования касательных при $f'(x) = 0$ метод не найдет корень, и существуют условия, при которых он не сходится. Также эффективность метода снижается при недостаточном изначальном приближении корня.

2. Метод хорд применим для нахождения корня уравнения на определенном интервале. Также, как и для метода касательных, существуют условия, при которых метод не сходится, и также возникают проблемы при построении хорд при $f'(x) = 0$. Помимо этого, метод хорд в большинстве случаев требует больше итераций, чем метод касательных, но имеет свой плюс: не требует использования производной от функции при вычислении.

3. Метод простой итерации прост в понимании и реализации, однако на практике требует большое количество итераций для достижения достаточного приближения корня.