Министерство науки и высшего образования Российской Федерации

Новосибирский государственный технический университет

Кафедра автоматизированных систем управления



**Лабораторная работа №5**

**«Расчет переходной функции численными методами»**

|  |  |
| --- | --- |
| Группа: АВТ-813  Студент:  Кинчаров Д. Д. | Преподаватель:  Достовалов Дмитрий Николаевич,  Заведующий кафедрой АСУ, доцент  кафедры Автоматизированных систем управления |

Новосибирск

2020

ЦЕЛЬ РАБОТЫ

Построить переходную характеристику системы с заданной передаточной функцией. Сравнить результаты расчета разными численными методами.

1. ПЕРЕДАТОЧНАЯ ФУНКЦИЯ, В СООТВЕТСТВИИ С ВАРИАНТОМ ЗАДАНИЯ.

ВАРИАНТ 30.

Согласно варианту передаточная функция выглядит следующим образом:

1. СТРУКТУРНАЯ СХЕМА В MATLAB.

На рисунке 1 представлена структурная схема заданной передаточной функции.

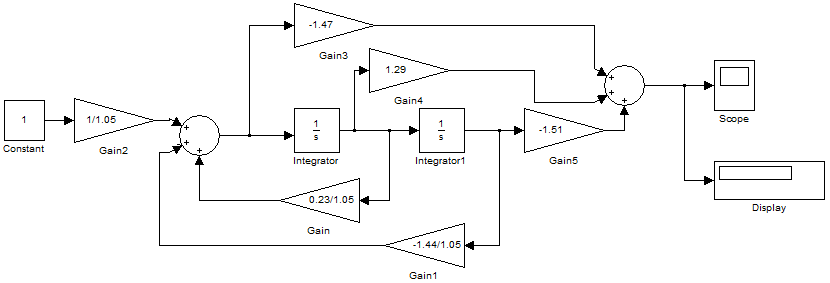


Рис. 1 – Структурная схема передаточной функции.

1. ГРАФИК ПЕРЕХОДНОЙ ХАРАКТЕРИСТИКИ ИЗ MATLAB.

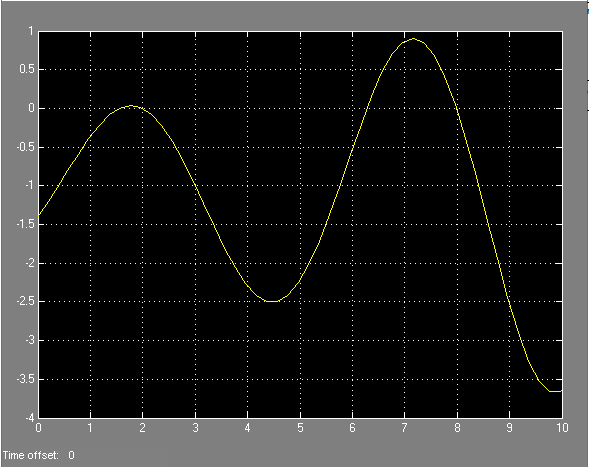
****

Рис. 2 – график переходной характеристики из Matlab.

1. ЗНАЧЕНИЯ И , ПОЛУЧЕННЫЕ В MATLAB.

Проверим систему на устойчивость. Характеристическое уравнение системы имеет вид:

Корни:

По корневому портрету получается неустойчевый фокус, значит система неустойчива.

Так как система неустойчива, мы берем любым. Пусть будет

Значение , его видно на рисунке 3 в блоке Display.

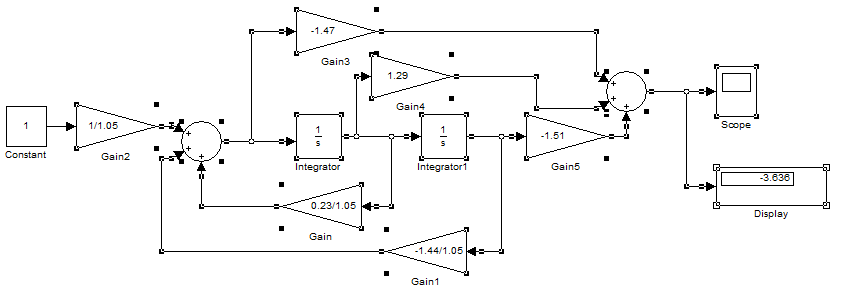


Рис. 3 – Структурная схема со значением y.

1. СИСТЕМА УРАВНЕНИЙ, ИСПОЛЬЗУЕМАЯ ДЛЯ РАСЧЕТА ПЕРЕХОДНОГО ПРОЦЕССА.
2. ЗНАЧЕНИЯ Y, ПОЛУЧЕННЫЕ С ПОМОЩЬЮ НАШИХ ПРОГРАММ. ВЫВОДЫ ОБ ОЦЕНКЕ ТОЧНОСТИ РАСЧЕТОВ.

Пусть шаг h = 0,001.

Полученное значение в нашей программе по методу Эйлера:

Полученное значение в нашей программе по методу Рунге-Кутты 4 порядка:

Полученное значение в Matlab:

**Абсолютная погрешность по методу Эйлера**:

**Относительная погрешность по методу Эйлера**:

**Абсолютная погрешность по методу Рунге-Кутты 4 порядка**:

**Относительная погрешность по методу Рунге-Кутты 4 порядка**:

Можем сделать вывод о том, что при относительной ошибке в , которая является незначительной, результат работы программы достаточно точен для метода Эйлера. Относительная ошибка по методу Рунге-Кутты 4 порядка составляет всего , что меньше результата по методу Эйлера. То есть метод Рунге-Кутты обладает более высокой точностью по отношению к методу Эйлера, что соответствует теории.

1. СКРИНШОТЫ ИЗ ПРОГРАММЫ

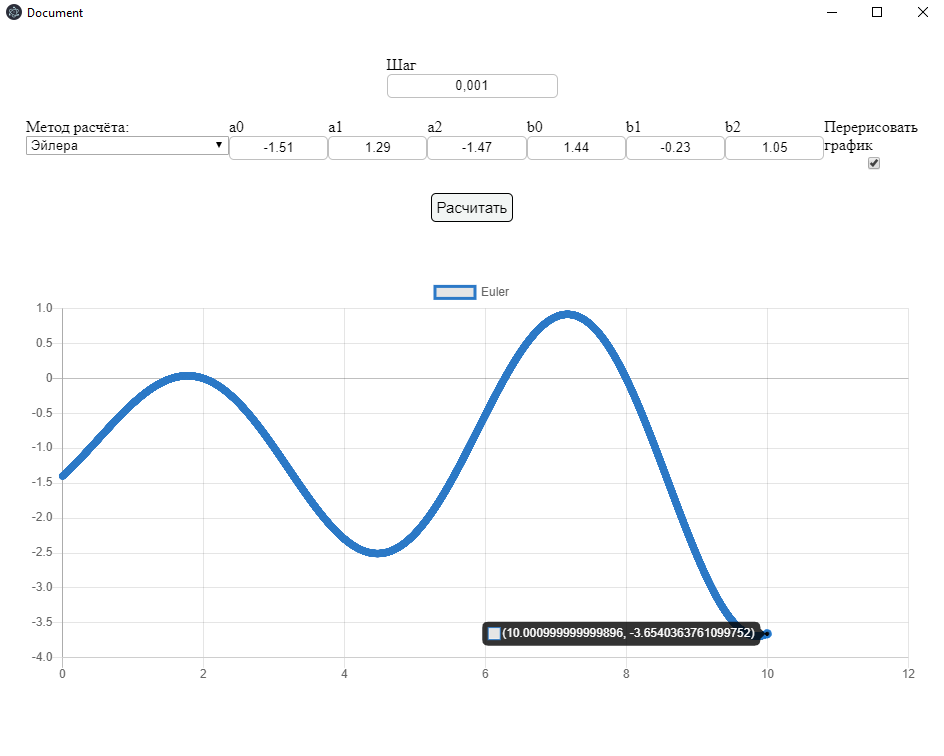
****

Рис.4 – Пример работы программы для метода Эйлера

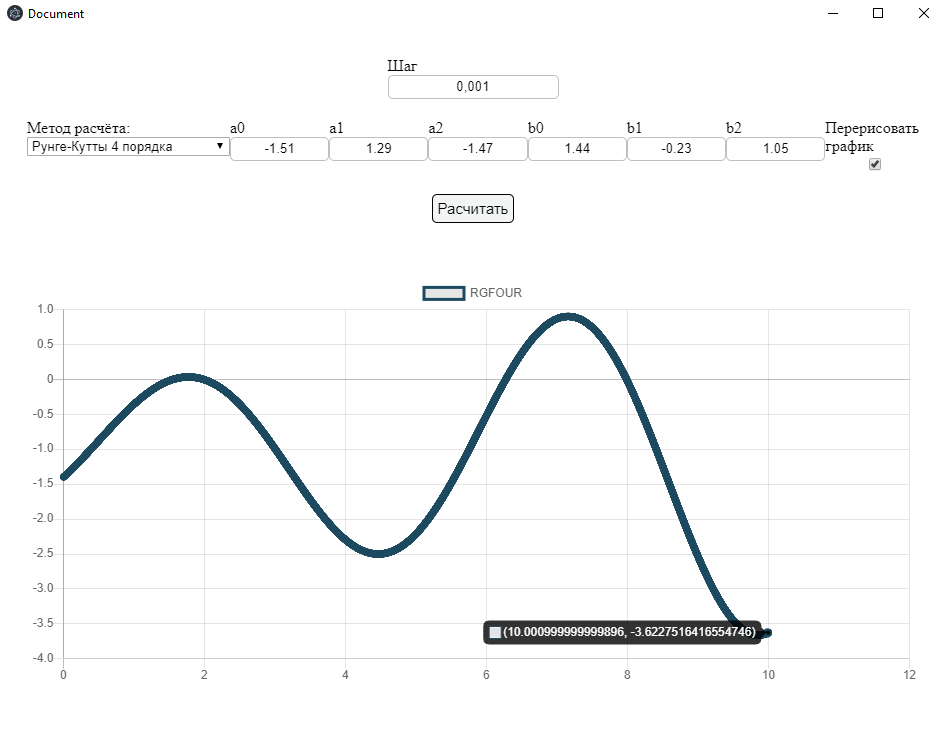


Рис.5 – Пример работы программы для метода Рунге-Кутты 4 порядка

1. МАТЕРИАЛЫ ПО ДОПОЛНИТЕЛЬНЫМ ЗАДАНИЯМ

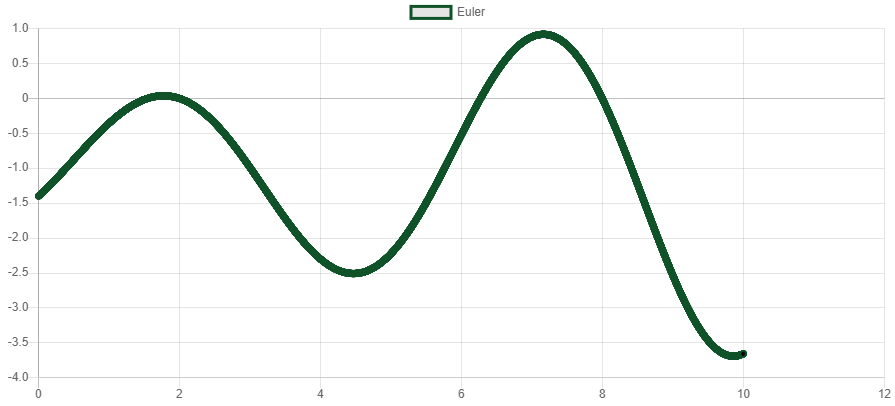


Рис.6 – График для метода Эйлера построенный нашей программой

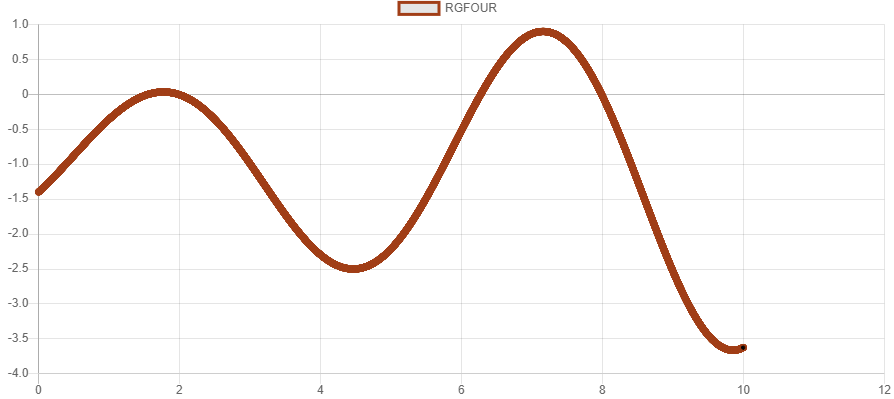


Рис.7 – График для метода Рунге-Кутты построенный нашей программой

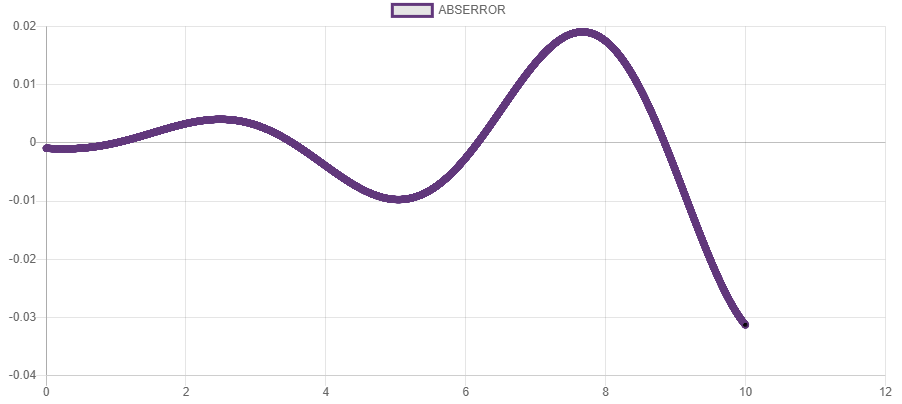


Рис.8 – График абсолютной погрешности для методов Эйлера и Рунге-Кутты построенный нашей программой (Рунге-Кутта – считаем, как точный)

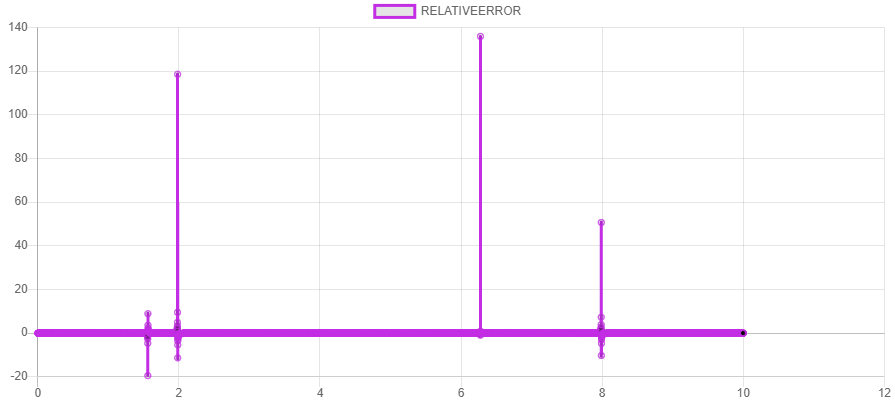


Рис.9 – График относительной погрешности для методов Эйлера и Рунге-Кутты построенный нашей программой (Рунге-Кутта – считаем, как точный)

На рисунке 9 заметны 4 выброса, появление которых можно объяснить тем, что возникает неопределенность из-за деления на 0. В этих точках график метода Рунге-Кутты пересекает ось абсцисс, то есть значение функции для этого метода в тех точках равно 0 (рис. 7).

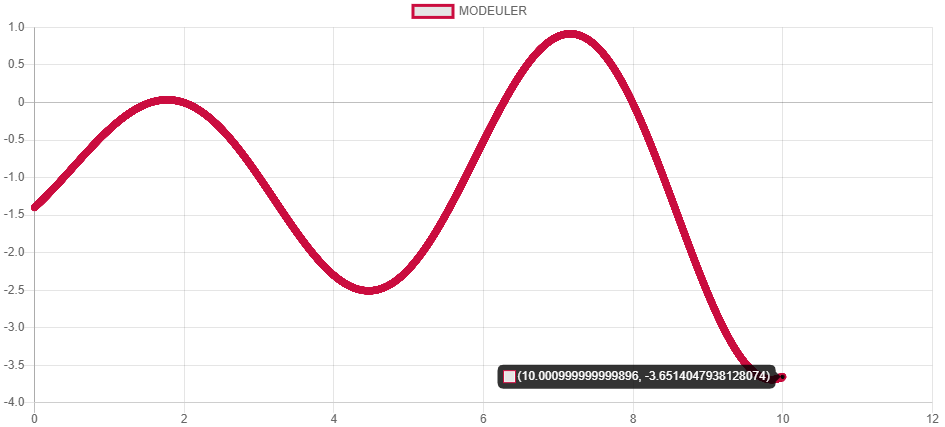


Рис.10 – График для дополнительного метода (модифицированный метод Эйлера) построенный нашей программой

Полученное значение в нашей программе по модифицированному методу Эйлера:

Полученное значение в Matlab:

**Абсолютная погрешность по методу модифицированному методу Эйлеру**:

**Относительная погрешность по модифицированному методу Эйлера**:

1. ПРИЛОЖЕНИЕ – ИСХОДНЫЕ КОДЫ ПРОГРАММЫ

*export* class DiffEquations

{

    constructor( coefs )

    {

        this.coefs = coefs

    }

    absoluteError( step ) {

        let firstResults = this.euler( step )

        let secondResults = this.rgFour( step )

        let results = []

*for*( const [ index, result ] of firstResults.entries() ) {

            results.push( { x: result.x, y: result.y - secondResults[ index ].y } )

        }

*return* results

    }

    relativeError( step ) {

        let absoluteErrorResults = this.absoluteError( step )

        let secondResults = this.rgFour( step )

        let results = []

*for*( const [ index, result ] of absoluteErrorResults.entries() ) {

            results.push( { x: result.x, y: result.y / secondResults[ index ].y } )

        }

*return* results

    }

    euler( step ) {

        let x1 = 0, x2 = 0, x3 = 0, y = 0, x = 0

        let results = []

*for*( let i = 0; i < 10001; i++ ) {

            x1 += step \* x2

            x2 += step \* x3

            x3 = 1 / this.coefs.b2 - this.coefs.b1 / this.coefs.b2 \* x2 - this.coefs.b0 / this.coefs.b2 \* x1

            y = this.coefs.a0 \* x1 + this.coefs.a1 \* x2 + this.coefs.a2 \* x3

            x += step

            results.push( { x, y } )

        }

*return* results

    }

    modEuler( step ) {

        let x1 = 0, x2 = 0, x3 = 0, y = 0, x = 0

        let results = []

*for*( let i = 0; i < 10001; i++ ) {

            x1 += step \* ( x2 + step / 2 \* x2 )

            x2 += step \* ( x3 + step / 2 \* x3 )

            x3 = 1 / this.coefs.b2 - this.coefs.b1 / this.coefs.b2 \* x2 - this.coefs.b0 / this.coefs.b2 \* x1

            y = this.coefs.a0 \* x1 + this.coefs.a1 \* x2 + this.coefs.a2 \* x3

            x += step

            results.push( { x, y } )

        }

*return* results

    }

    rgFour( step ) {

        let x1 = 0, x2 = 0, y = 0, x = 0

        let k1 = 0, k2 = 0, k3 = 0, k4 = 0, m1 =0, m2 = 0, m3 = 0, m4 =0

        let results = []

*for* ( let i = 0; i < 10001; i++ ) {

            m1 = x2

            m2 = x2 + step / 2 \* m1

            m3 = x2 + step / 2 \* m2

            m4 = x2 + step \* m3

            x1 += ( m1 + 2 \* m2 + 2 \* m3 + m4 ) \* step / 6

            k1 = ( -this.coefs.b0 \* x1 - this.coefs.b1 \* x2 + 1 ) / this.coefs.b2

            k2 =( -this.coefs.b0 \* ( x1 + step / 2 \* m1 ) -this.coefs.b1 \* ( x2 + step \* k1 / 2 ) + 1 ) / this.coefs.b2

            k3 = ( -this.coefs.b0 \* ( x1 + step / 2 \* m2 ) -this.coefs.b1 \* ( x2 + step \* k2 / 2 ) + 1 ) / this.coefs.b2

            k4 = ( -this.coefs.b0 \* ( x1 + step \* m3 ) -this.coefs.b1 \* ( x2 + step \* k3 ) + 1 ) / this.coefs.b2

            x2 += ( k1 + 2 \* k2 + 2 \* k3 + k4 ) \* step / 6

            y = this.coefs.a0 \* x1 + this.coefs.a1 \* x2 + this.coefs.a2 \* ( -this.coefs.b0 \* x1 - this.coefs.b1 \* x2 + 1 ) / this.coefs.b2

            x += step

            results.push( { x, y } )

        }

*return* results

    }

}