**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра КСУ**

отчет

**по идз №1**

**по дисциплине «ПРОЕКТИРОВАНИЕ**

**ОПТИМАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ»**

Тема: Решение дифференциальных уравнений

**Вариант 12**

|  |  |  |
| --- | --- | --- |
| Студент гр. 4491 | Пономарев Д.А. |  |
| Преподаватель | Ветчинкин А.С. |  |

Санкт-Петербург

2018

**Решение дифференциальных уравнений**

**Исходные данные**

При выполнении задания необходимо найти аналитическое решение уравнения путем ручных преобразований, выполнить численное решение, построить графики переходных процессов, соответствующие аналитическому и численному решениям, оценить время переходного процесса и оценить влияние выбора шага интегрирования или настроек параметров точности солверов. Для численного решения уравнений требуется разработка MATLAB скриптов, в которых реализуется метод Эйлера и применяются солверы ODE45 и ODE23S.

Исходные данные заданы в таблице 1.

Таблица 1. Исходные данные к заданию

|  |  |
| --- | --- |
| Вариант | Матрица уравнения |
| 12 |  |

**Аналитическое решение**

1. Запишем исходную систему уравнений:
2. Выполним преобразование по Лапласу и приведем к виду:
3. Воспользуемся методом Крамера:

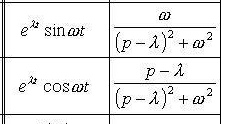
Тогда:

4. Приведем к к табличному виду.

Поскольку порядок 3 и есть комплексные корни, будем раскладывать на следующие дроби:

где I – то номер переменной состояния, – корень ХП, w – частота периодической составляющей.

В таком виде удобно будет воспользоваться таблицей преобразования Лапласа.



Пример расчета для :

Введем следующие обозначения

Тогда:

Получаем систему уравнений:

1.000\* + 0.000\* + 1.000\* = 0.000

1.327\* + 2.380\* + 1.346\*= -3.000

0.440\* + 1.556\* + 6.120\*= -4.000

Методом Крамера получаем коэффициенты:

= 0.360, = -1.257, = -0.360

В результате расчетов были получены следующие дроби:

Выполняем обратное преобразование по таблице:

Расчет времени переходного процесса

Результаты построения графиков функции представлены на рисунке 1.

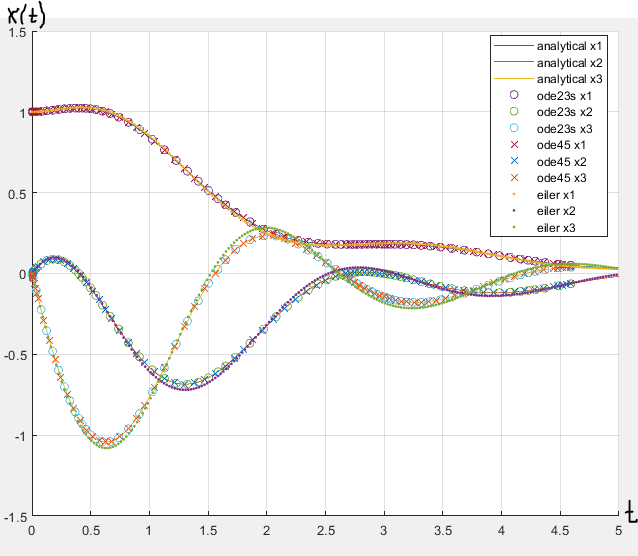


Рисунок 1 – Графики функции

На графике численного решения методом Эйлера шаг интегрирования выбран 0.005 сек.

Код программы, строящей графики:

|  |
| --- |
| clear; close all; clc;    % Input data variant12  A = [0 1 0; 1 0 2; -3 -4 -2];  B = [1 0 0];  T = 4.59; % Transient response time  % Analytical plots  figure; grid on; hold on;  lambda1 = -0.6732;  lambda2 = -0.6536;  w = 2.3805;  rootsABC = [-0.256356, 0.272450, 1.256356]  plot(t, rootsABC(3)\*exp(t\*lambda2) + rootsABC(1)\*cos(w\*t).\*exp(lambda1\*t) + rootsABC(2)\*sin(w\*t).\*exp(lambda1\*t))  rootsABC = [0.821148, 0.426843, -0.821148]  plot(t, rootsABC(3)\*exp(t\*lambda2) + rootsABC(1)\*cos(w\*t).\*exp(lambda1\*t) + rootsABC(2)\*sin(w\*t).\*exp(lambda1\*t))  rootsABC = [0.359829, -1.257274, -0.359829]  plot(t, rootsABC(3)\*exp(t\*lambda2) + rootsABC(1)\*cos(w\*t).\*exp(lambda1\*t) + rootsABC(2)\*sin(w\*t).\*exp(lambda1\*t))  % Create plot ode45, ode23s  [t, x] = ode23s('idz1\_fun', [0 T], B);  plot(t, x, 'o')  [t, x] = ode45('idz1\_fun', [0 T], B);  plot(t, x, 'x')    % Create plot Eiler  x1\_graph = []; x2\_graph = []; x3\_graph = []; t\_graph = [];  x1 = 1; x2 = 0; x3 = 0; t = 0;  dt = 0.005\*T;  while t < 5  x1\_graph = [x1\_graph x1];  x2\_graph = [x2\_graph x2];  x3\_graph = [x3\_graph x3];  t\_graph = [t\_graph t];  dx1 = (A(1, 1)\*x1 + A(1,2)\*x2 + A(1,3)\*x3)\*dt;  dx2 = (A(2, 1)\*x1 + A(2,2)\*x2 + A(2,3)\*x3)\*dt;  dx3 = (A(3, 1)\*x1 + A(3,2)\*x2 + A(3,3)\*x3)\*dt;  x1 = x1 + dx1;  x2 = x2 + dx2;  x3 = x3 + dx3;  t = t + dt;  end  plot(t\_graph, x1\_graph, '.' , ...  t\_graph, x2\_graph, '.' , ...  t\_graph, x3\_graph, '.')    % Other options  legend('analytical x1', 'analytical x2', 'analytical x3', ...  'ode23s x1', 'ode23s x2', 'ode23s x3', ...  'ode45 x1', 'ode45 x2', 'ode45 x3', ...  'eiler x1', 'eiler x2', 'eiler x3') |

|  |
| --- |
| function dxdt = idz1\_fun(t, x)  A = [0 1 0; 1 0 2; -3 -4 -2];  dxdt = [A(1, 1)\*x(1) + A(1,2)\*x(2) + A(1,3)\*x(3);...  A(2, 1)\*x(1) + A(2,2)\*x(2) + A(2,3)\*x(3);...  A(3, 1)\*x(1) + A(3,2)\*x(2) + A(3,3)\*x(3)];  end |