**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Систем автоматизированного проектирования**

отчет

**по лабораторной работе №1**

**по дисциплине «Моделирование нелинейных динамических систем»**

Тема: Моделирование хаотической системы методами Эйлера, явной средней точки, КД и **референсным методом DOPRI8**

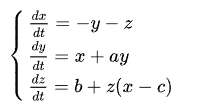
|  |  |  |
| --- | --- | --- |
| Студентка гр. 1302 |  | Марзаева В.И. |
| Студент гр. 1302 |  | Новиков Г.В. |
| Студентка гр. 1302 |  | Романова О.В. |
| Преподаватель |  | Бабкин И.А. |

Санкт-Петербург

2024

**Цель работы**

Моделирование хаотической системы (аттрактора Рёсслера) методами Эйлера, явной средней точки (explicit mid-point/RK2), КД и референсным методом DOPRI8. Построение фазового портрета, временной области, оценка разницы между реализациями 2-мя разными методами во временной области и в фазовом пространстве (Эйлер, EMP и КД сравниваются с референсным методом DOPRI8). Анализ аттракторов погрешности, полученных разными методами.



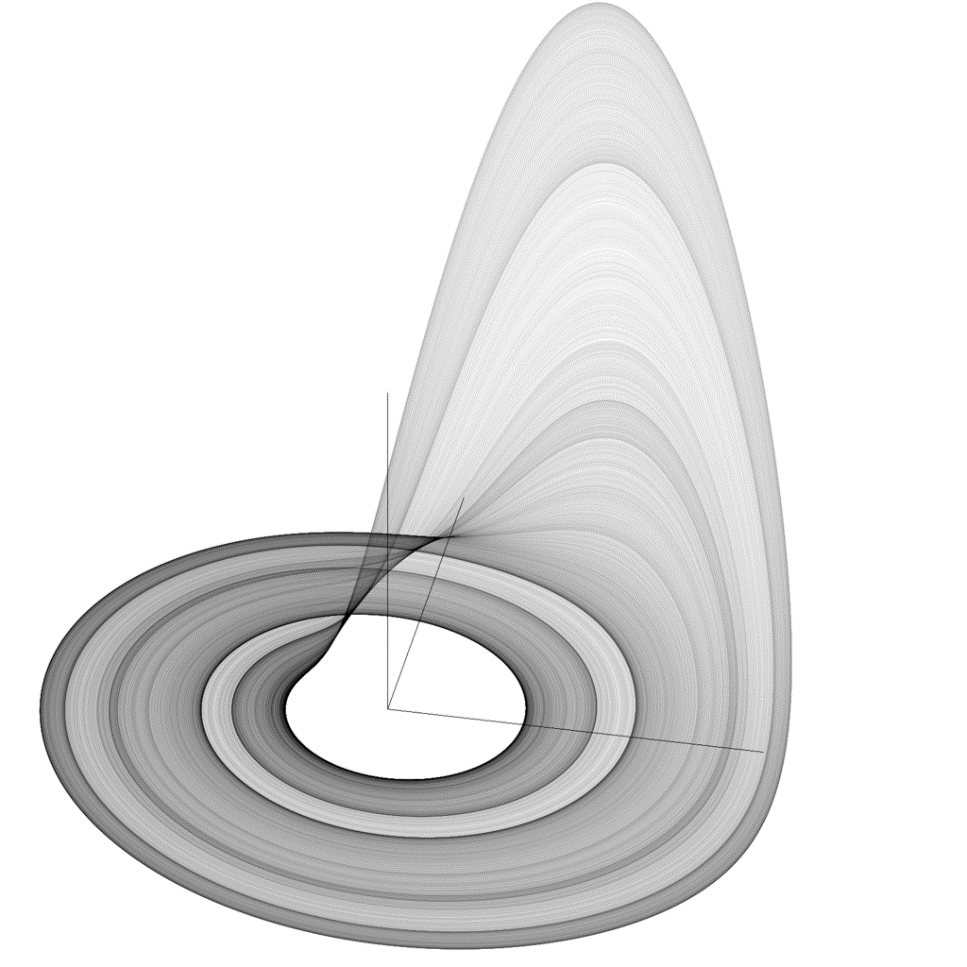


Рис. . Аттрактор Рёсслера

**Работа программы**

Шаг , количество точек .

**Метод Эйлера**

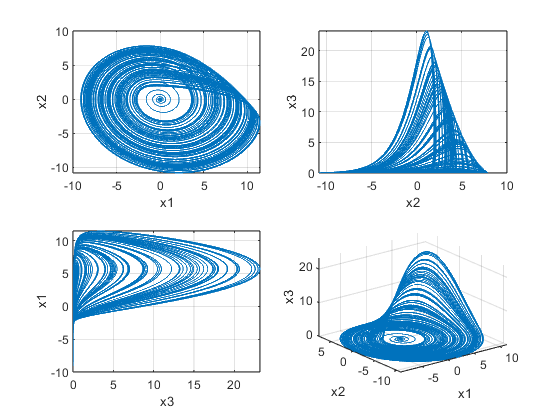


Рис. 2. Фазовый портрет

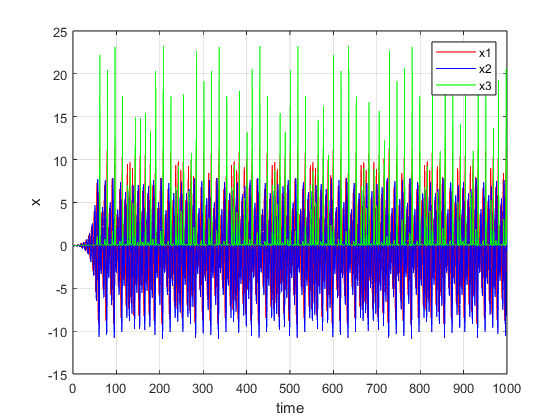


Рис. 3. Значения переменных во временной области

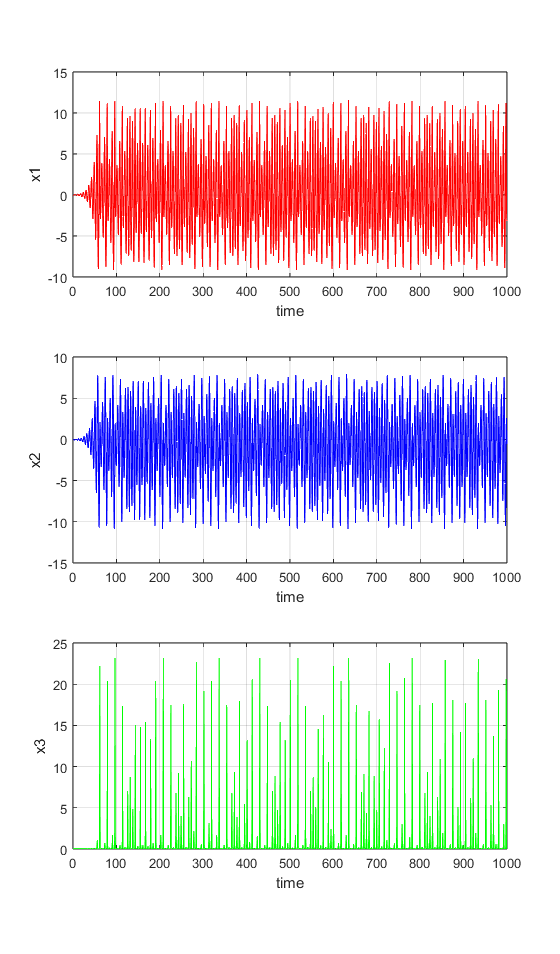


Рис. . Значения переменных во временной области

**Метод явной средней точки**

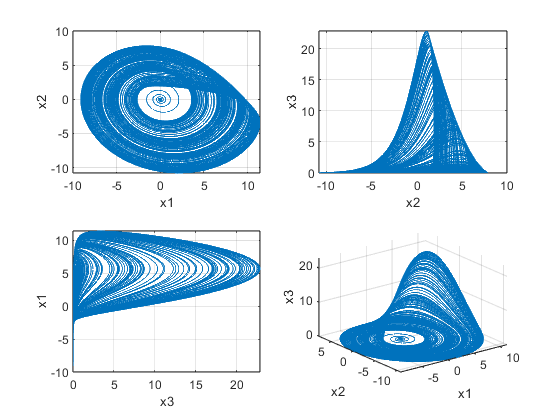


Рис. 5. Фазовый портрет

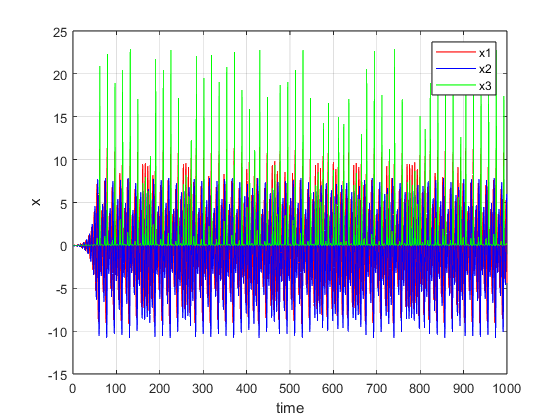


Рис. 6. Значения переменных во временной области

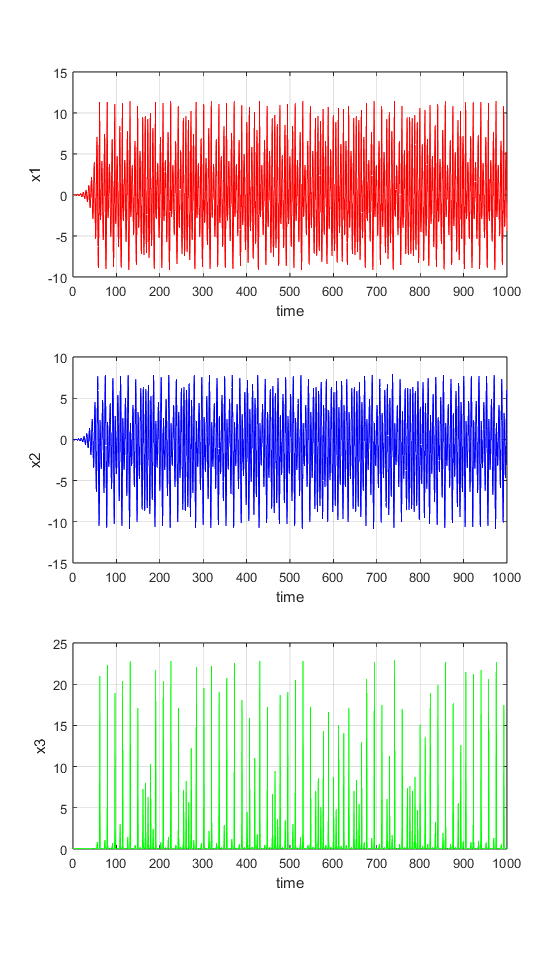


Рис. 7. Значения переменных во временной области

**Метод КД**

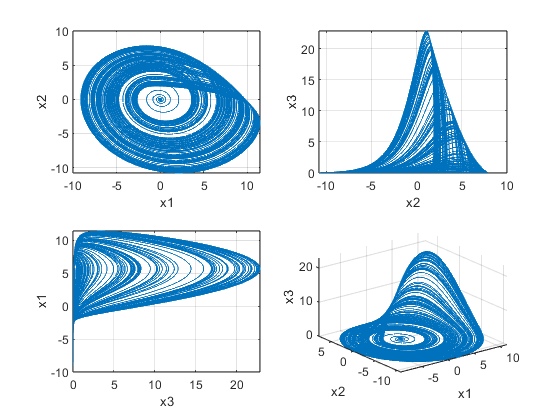


Рис. 8. Фазовый портрет

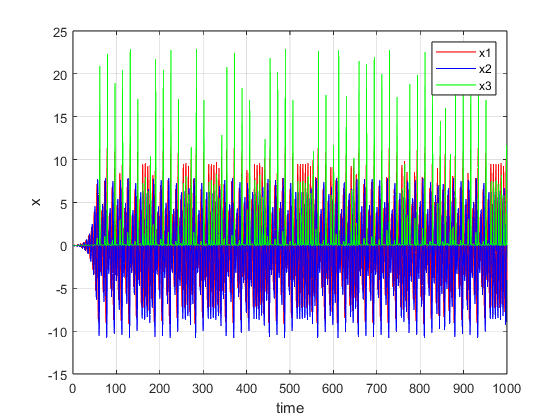


Рис. 9. Значения переменных во временной области

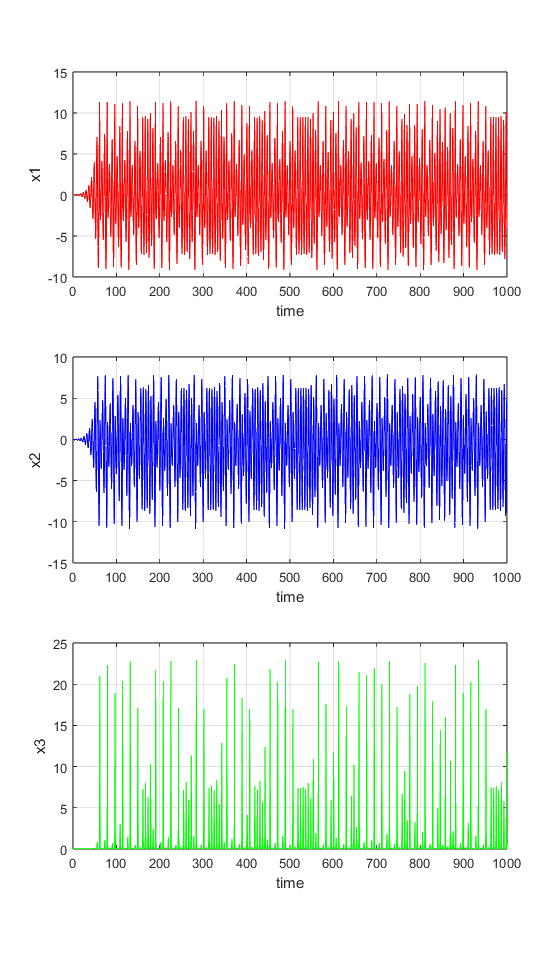


Рис. 10. Значения переменных во временной области

**Метод DOPRI8**

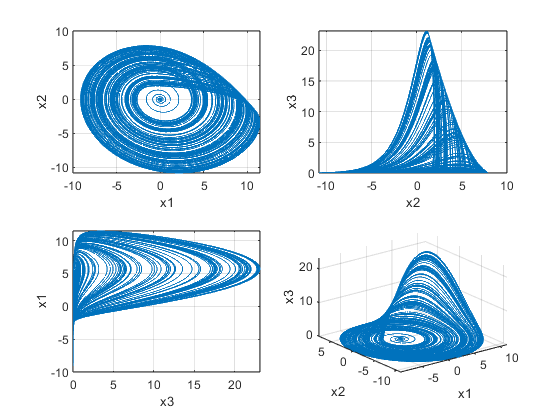


Рис. 11. Фазовый портрет

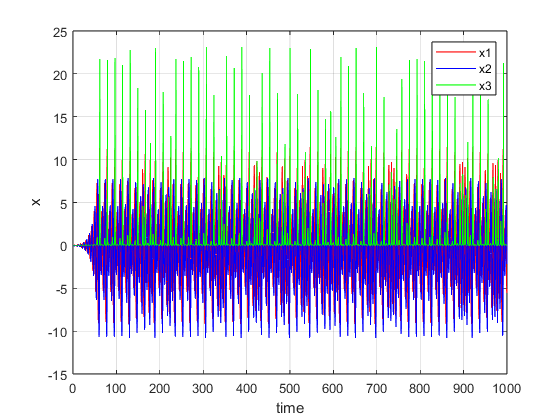


Рис. 12. Значения переменных во временной области

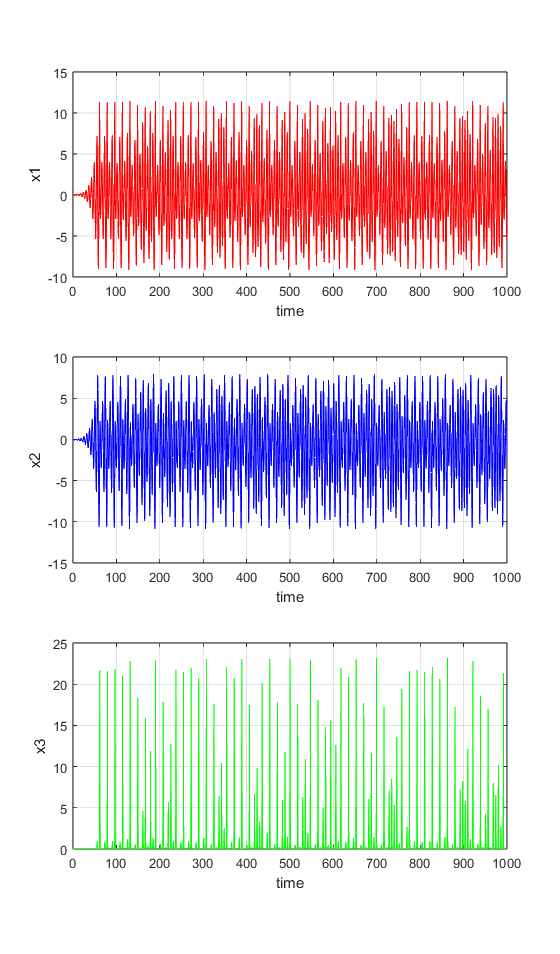


Рис. 13. Значения переменных во временной области

**Порядок методов**

Сравним методы с эталонным методом DOPRI 8 и определим их порядок точности.

**Порядок метода Эйлера**

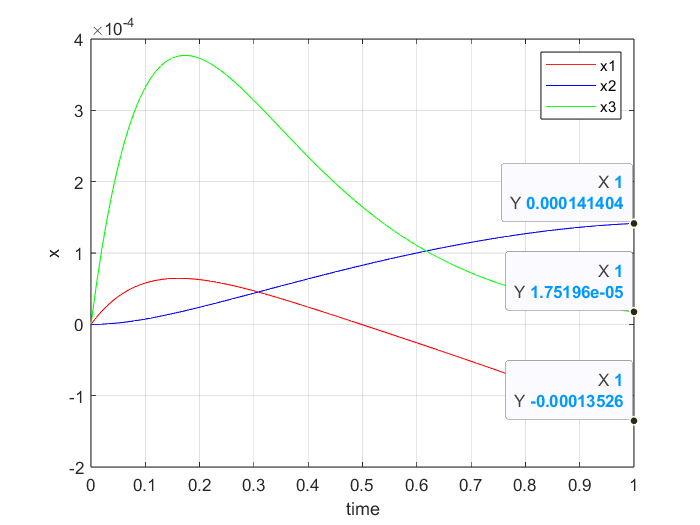


Рис. . Метод Эйлера,

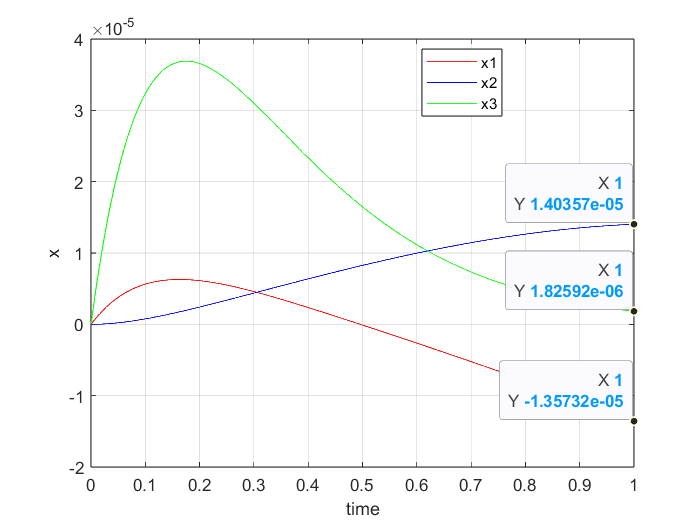


Рис. . Метод Эйлера,

При уменьшении h в 10 раз ошибка уменьшается в 10 раз, следовательно порядок точности первый.

**Порядок метода средней точки**

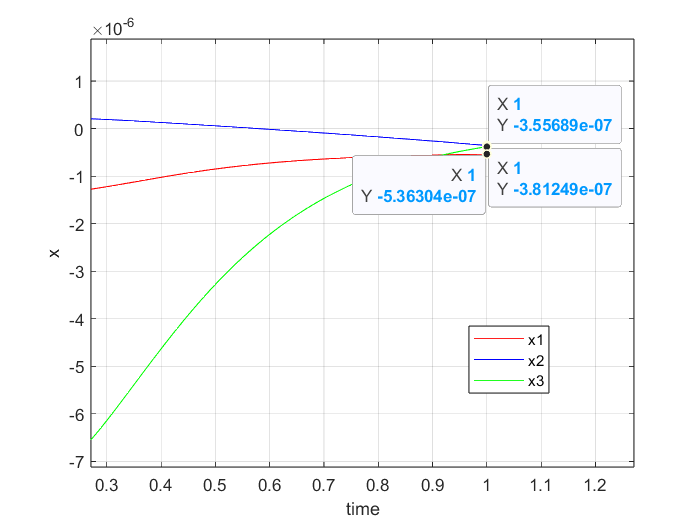


Рис. . Метод средней точки,

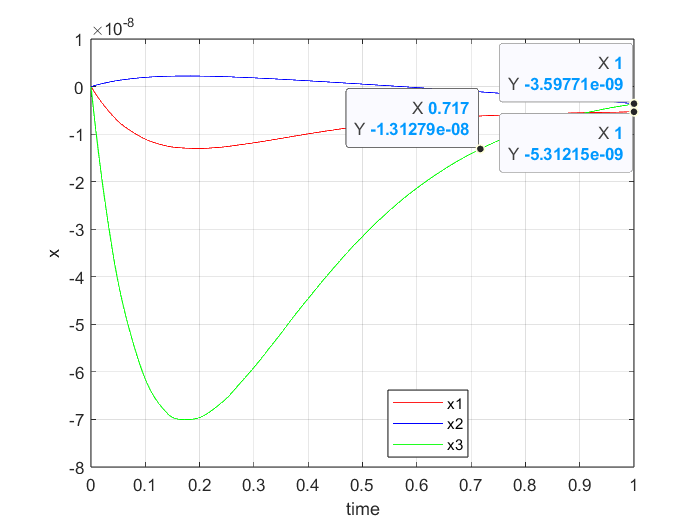


Рис. . Метод средней точки,

При уменьшении h в 10 раз ошибка уменьшается примерно в 100 раз, следовательно порядок точности второй.

**Порядок метода КД**

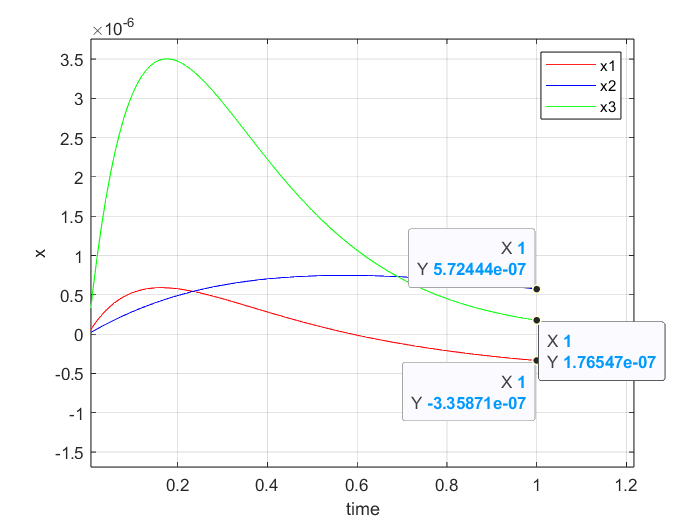


Рис. . Метод КД,

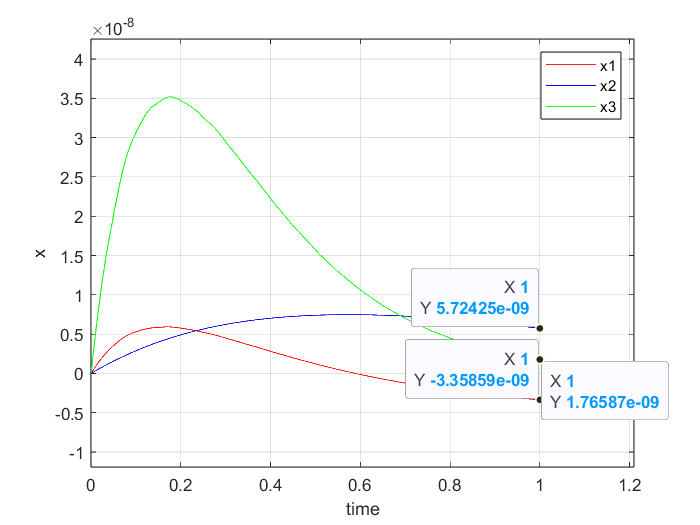


Рис. . Метод КД,

При уменьшении h в 10 раз ошибка уменьшается в 100 раз, следовательно порядок точности второй.

**Код программы**

f.m:

function [Y] = f(X)

% dx1 / dt = Y(1)

% dx2 / dt = Y(2)

% dx2 / dt = Y(3)

a = 0.2;

b = 0.2;

c = 5.7;

Y = zeros(1, 3);

Y(1) = -X(2) - X(3);

Y(2) = X(1) + a \* X(2);

Y(3) = b + X(3) \* (X(1) - c);

end

f\_ode.m:

function [Y] = f\_ode(t, X)

Y = f(X);

Y = Y';

end

lab1.m:

clear

X0 = [0 0 0];

h = 1e-3;

n = 1e3;

X = solve\_euler(X0, n, h);

% X = solve\_midpoint(X0, n, h);

% X = solve\_cd(X0, n, h, 0.5);

% X = solve\_dopri8(X0, n, h);

% plot\_3d\_time(h, X);

% plot\_3d\_phase(X);

compare\_to\_dopri(X, X0, n, h);

plot\_3d\_phase.m:

function [] = plot\_3d\_phase(X)

figure

subplot(2, 2, 1)

plot(X(:,1), X(:,2));

xlabel('x1')

ylabel('x2')

grid on

subplot(2, 2, 2)

plot(X(:,2), X(:,3));

xlabel('x2')

ylabel('x3')

grid on

subplot(2, 2, 3)

plot(X(:,3), X(:,1));

xlabel('x3')

ylabel('x1')

grid on

subplot(2, 2, 4)

plot3(X(:,1), X(:,2), X(:,3));

xlabel('x1')

ylabel('x2')

zlabel('x3')

grid on

end

plot\_3d\_time.m:

function [] = plot\_3d\_time(h, X)

t = 0 : h : (length(X) - 1) \* h;

figure

plot(t, X(:, 1), 'r', t, X(:, 2), 'b', t, X(:, 3), 'g');

legend('x1', 'x2', 'x3')

grid on

xlabel('time')

ylabel('x')

figure

subplot(3, 1, 1)

plot(t, X(:, 1), 'r');

grid on

xlabel('time')

ylabel('x1')

subplot(3, 1, 2)

plot(t, X(:, 2), 'b');

grid on

xlabel('time')

ylabel('x2')

subplot(3, 1, 3)

plot(t, X(:, 3), 'g');

grid on

xlabel('time')

ylabel('x3')

end

solve\_euler.m:

function [Y] = solve\_euler(X, n, h)

Y = zeros(n + 1, 3);

Y(1, 1:3) = X;

for i = 2 : n + 1

Y\_prev = Y(i - 1, :);

Y(i, :) = Y\_prev + h \* f(Y\_prev);

end

end

solve\_midpoint.m:

function [Y] = solve\_midpoint(X, n, h)

Y = zeros(n + 1, 3);

Y(1, 1:3) = X;

for i = 2 : n + 1

Y\_prev = Y(i - 1, :);

[F] = f(Y\_prev + h / 2 \* f(Y\_prev));

Y(i, :) = Y\_prev + h \* F;

end

end

solve\_cd.m:

function [Y] = solve\_cd(X, n, h, s)

h1 = s \* h;

h2 = (1 - s) \* h;

a = 0.2;

b = 0.2;

c = 5.7;

Y = zeros(n + 1, 3);

Y(1, 1:3) = X;

for i = 2 : n + 1

% Euler

Y\_prev = Y(i - 1, :);

Y\_mid = zeros(1, 3);

Y\_mid(1) = Y\_prev(1) + h1 \* (-Y\_prev(2) - Y\_prev(3));

Y\_mid(2) = Y\_prev(2) + h1 \* (Y\_mid(1) + a \* Y\_prev(2));

Y\_mid(3) = Y\_prev(3) + h1 \* (b + Y\_prev(3) \* (Y\_mid(1) - c));

% D-method

Y(i, 3) = (Y\_mid(3) + h2 \* b) / (1 - h2 \* Y\_mid(1) + h2 \* c);

Y(i, 2) = (Y\_mid(2) + h2 \* Y\_mid(1)) / (1 - a \* h2);

Y(i, 1) = Y\_mid(1) + h2 \* (-Y(i, 2) - Y(i, 3));

end

end

solve\_dopri8.m:

function [Y] = solve\_dopri8(X, n, h)

A = [

0 0 0 0 0 0 0 0 0 0 0 0

0.0555556 0 0 0 0 0 0 0 0 0 0 0

0.0208333 0.0625 0 0 0 0 0 0 0 0 0 0

0.03125 0 0.09375 0 0 0 0 0 0 0 0 0

0.3125 0 -1.17188 1.17188 0 0 0 0 0 0 0 0

0.0375 0 0 0.1875 0.15 0 0 0 0 0 0 0

0.0479101 0 0 0.112249 -0.0255057 0.0128468 0 0 0 0 0 0

0.016918 0 0 0.387848 0.0359774 0.19697 -0.172714 0 0 0 0 0

0.0690958 0 0 -0.634248 -0.161198 0.13865 0.940929 0.211636 0 0 0 0

0.183557 0 0 -2.46877 -0.291287 -0.026473 2.84784 0.281387 0.123745 0 0 0

-1.21542 0 0 16.6726 0.915742 -6.05661 -16.0036 14.8493 -13.3716 5.13418 0 0

0.258861 0 0 -4.77449 -0.435093 -3.04948 5.57792 6.15583 -5.0621 2.19393 0.134628 0

0.822428 0 0 -11.6587 -0.757622 0.713974 12.0758 -2.12766 1.99017 -0.234286 0.175899 0

];

b = [0.0417475 0 0 0 0 -0.0554523 0.239313 0.703511 -0.75976 0.660563 0.158187 -0.23811 0.25];

Y = zeros(n + 1, 3);

Y(1, 1:3) = X;

for j = 2 : n + 1

k = zeros(length(b), 3);

for i = 1 : length(k)

dy = 0;

for m = 1 : i - 1

dy = dy + A(i, m) \* k(m);

end

dy = dy \* h;

k(i, :) = f(Y(j - 1, :) + dy);

end

dy = zeros(1, 3);

for i = 1 : length(k)

dy = dy + k(i, :) \* b(i);

end

dy = dy \* h;

Y(j, :) = Y(j - 1, :) + dy;

end

end

compare\_to\_dopri.m:

function [] = compare\_to\_dopri(Y, X0, n, h)

% Y\_dopri = solve\_dopri8(X0, n, h);

[t, Y\_dopri] = ode78(@f\_ode, [0 : h : n \* h], X0);

Y\_diff = Y - Y\_dopri;

plot\_3d\_time(h, Y\_diff);

% plot\_3d\_phase(Y\_diff);

end

**Выводы**

В лабораторной работе было сделано моделирование хаотической системы методами Эйлера, явной средней точки (explicit mid-point/RK2), КД и референсным методом DOPRI8, были построены фазовые портреты, временные области, оценка разницы между реализациями 2-мя разными методами во временной области и в фазовом пространстве (Эйлер, EMP и КД сравниваются с референсным методом DOPRI8) и произведен анализ аттракторов погрешности, полученных разными методами. Также мы узнали, что порядок точности первый у метода Эйлера, потому что при уменьшении h в 10 раз ошибка уменьшается в 10 раз при сравнении методов с эталонным методом DOPRI8 и второй порядок точности у методов средней точки и КД.