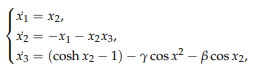
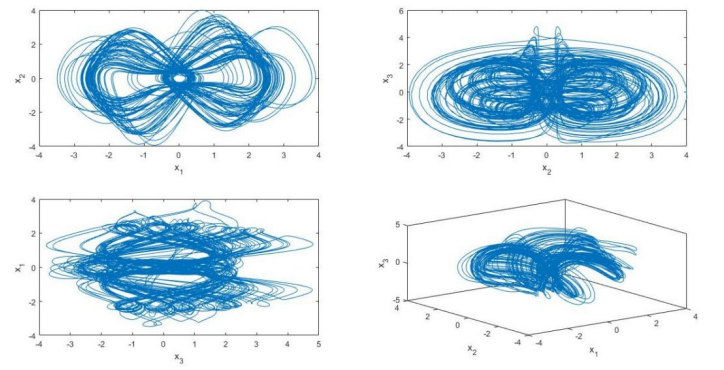
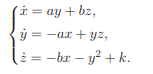
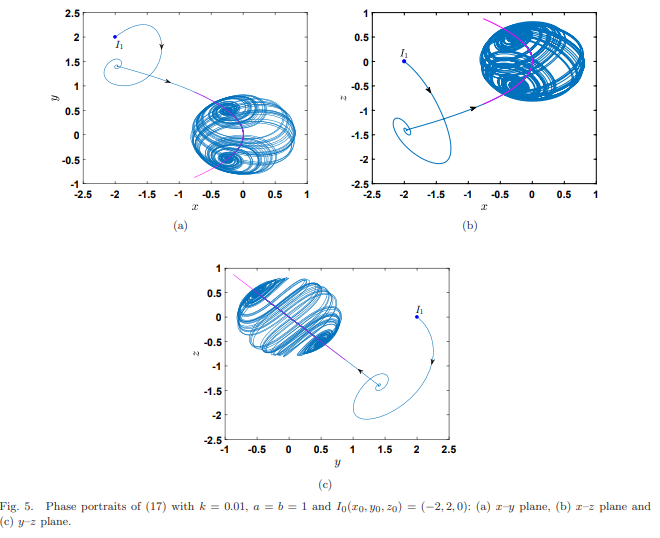
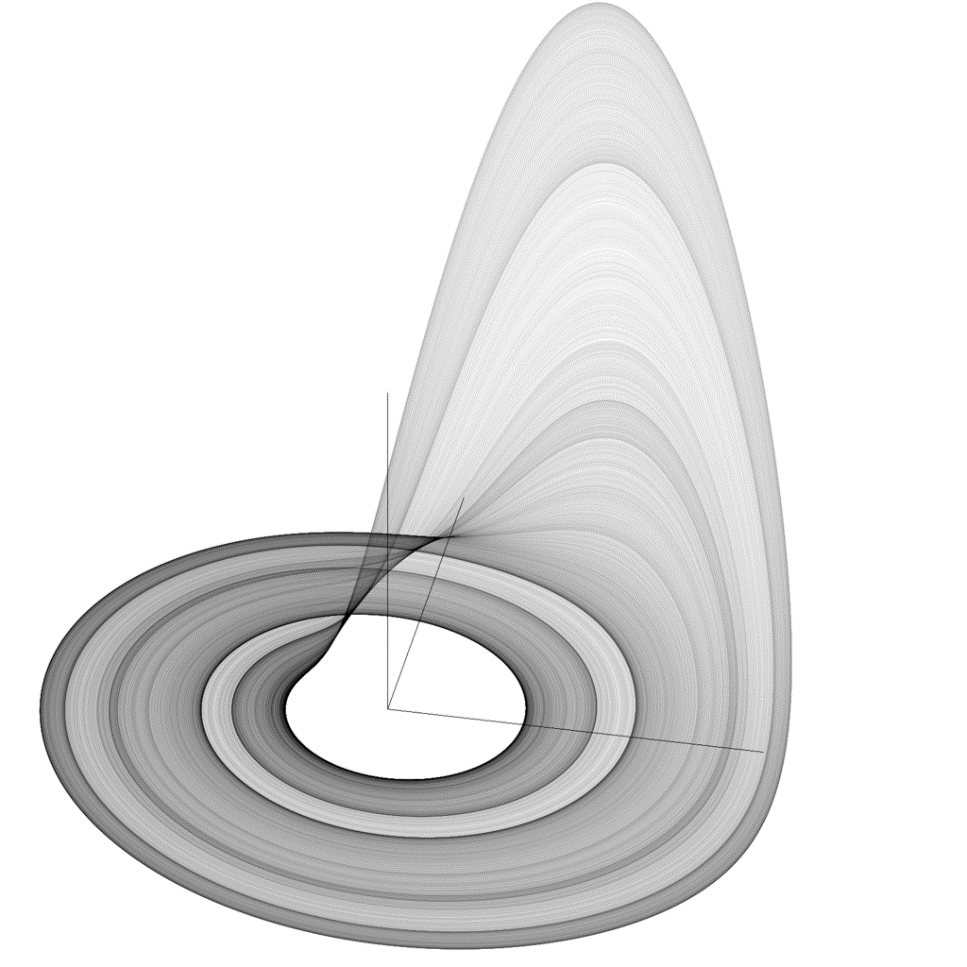
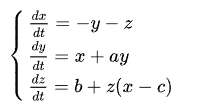
**Лаба 1:** Моделирование хаотической системы методами Эйлера, явной средней точки (explicit mid-point/RK2), КД и референсным методом DOPRI8. Построение фазового портрета, временной области, оценка разницы между реализациями 2-мя разными методами во временной области и в фазовом пространстве (Эйлер, EMP и КД сравниваются с референсным методом DOPRI8). Анализ аттракторов погрешности, полученных разными методами.

Вопросы: относятся ли системы, описываемые аттракторами погрешности, к разным классам?

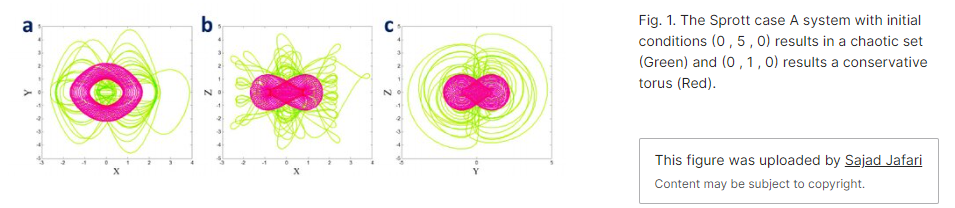


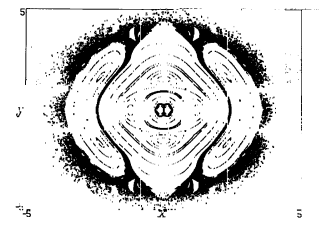


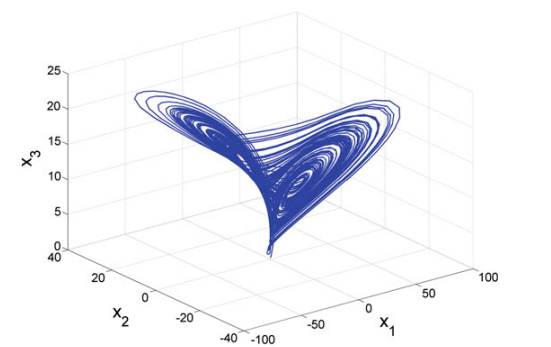
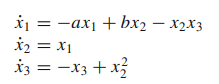
Если менять параметры, то можно получить более интересные формы, чем сфера

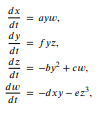
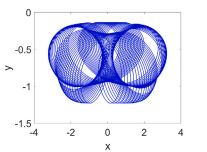
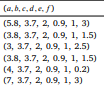


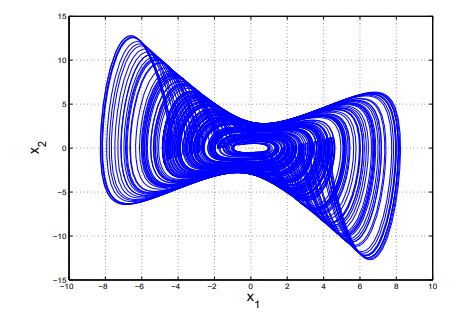
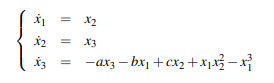




есть только сечение





//DOPRI-8 LORENZ

float z[3], y[3], X1[3], X2[3], X3[3];

int N = 3; // number of variables

float k[3][13]; // first index is a number of variables

int i, j, l;

for (i = 0; i < N; i++)

X1[i] = X[i];

for (i = 0; i < 13; i++) { // CHANGE RIGHT-SIDE FUNCTION HERE

k[0][i] = (a[0]\*(X1[1] - X1[0]));

k[1][i] = (X1[0]\*(a[1] - X1[2]) - X1[1]);

k[2][i] = (X1[0]\*X1[1] - a[2]\*X1[2]);

for (l = 0; l < N; l++)

X2[l] = 0;

for (j = 0; j < i + 1; j++)

for (l = 0; l < N; l++)

X2[l] += M[i + 1][j] \* k[l][j];

for (l = 0; l < N; l++)

X1[l] = X[l] + h \* X2[l];

}

for (l = 0; l < N; l++)

X2[l] = 0;

for (i = 0; i < N; i++)

X3[i] = X2[i];

for (i = 0; i < 13; i++)

for (l = 0; l < N; l++)

X2[l] += b[0][i] \* k[l][i];

for (i = 0; i < 13; i++)

for (l = 0; l < N; l++)

X3[l] += b[1][i] \* k[l][i];

for (l = 0; l < N; l++)

y[l] = X[l] + h \* X2[l];

for (l = 0; l < N; l++)

z[l] = X[l] + h \* X3[l];

for (l = 0; l < N; l++)

X[l] = y[l];

//EMP LORENZ

float X1[3];

X1[0] = X[0] + 0.5 \* h \* (a[0]\*(X[1] - X[0]));

X1[1] = X[1] + 0.5 \* h \* (X[0]\*(a[1] - X[2]) - X[1]);

X1[2] = X[2] + 0.5 \* h \* (X[0]\*X[1] - a[2]\*X[2]);

X[0] = X[0] + h \* (a[0]\*(X1[1] - X1[0]));

X[1] = X[1] + h \* (X1[0]\*(a[1] - X1[2]) - X[1]);

X[2] = X[2] + h \* (X1[0]\*X1[1] - a[2]\*X1[2]);

//Euler LORENZ

float X1[3];

int i;

X1[0] = X[0] + h \* (a[0]\*(X[1] - X[0]));

X1[1] = X[1] + h \* (X[0]\*(a[1] - X[2]) - X[1]);

X1[2] = X[2] + h \* (X[0]\*X[1] - a[2]\*X[2]);

for (i = 0; i < 3; i++)

{

X[i] = X1[i];

}