Отчет по заданию

Вариант: М3.1 “Модель Ю.Пыха”

Выполнила: Брындина А.А.

Преподаватель: Есикова Н.Б.

305 группа

1. Аналитическое решение задачи

Нам дана система:

$$y'_1 = y_1\left[(3 - 4\mu) - y_1 - y_2 - y_3\right]$$

$$y'_2 = y_2\left(-E_2 + y_1 + 2\mu y_2\right)$$

$$y'_3 = y_3\left(-E_3 + y_2\right)$$

Параметры системы: E_i, μ; $y_i \geq 0$ при $E_i > 0$, $0 < \mu < \mu_0$

Найдем аналитически стационарные точки $y*$ как функции μ и E_i.

Получим:

$$y_1 = E_2 - 2\mu E_3$$

$$y_2 = E_3$$

$$y_3 = (3 - 4\mu) + 2\mu E_3 - E_2 - E_3$$

В качестве начальных условий возьмем:

$$y_1 = E_2 - 2\mu E_3 + \varepsilon$$

$$y_2 = E_3 + \varepsilon$$

$$y_3 = (3 - 4\mu) + 2\mu E_3 - E_2 - E_3 + \varepsilon$$

где $0 < \varepsilon < 1$. Выберем $\varepsilon = 0.5$

Воспользуемся классическим методом Рунге-Кутта.

Он описывается системой соотношения для $i = 0,\ldots, n - 1$

В нашем случае $n = 3$

$$y_{i+1} = y_i + h/6(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f\left(x_i, y_i\right)$$

$$k_2 = f\left(x_i + h/2, y_i + h/2k_1\right)$$

$$k_3 = f\left(x_i + h/2, y_i + h/2k_2\right)$$

$$k_4 = f\left(x_i + h, y_i + hk_3\right)$$

k_1, k_2, k_3, k_4- приближающие угловые коэффициенты.

Классический метод Рунге-Кутта имеет четвертый порядок точности. Он является явным и допускает расчёт с переменным шагом.

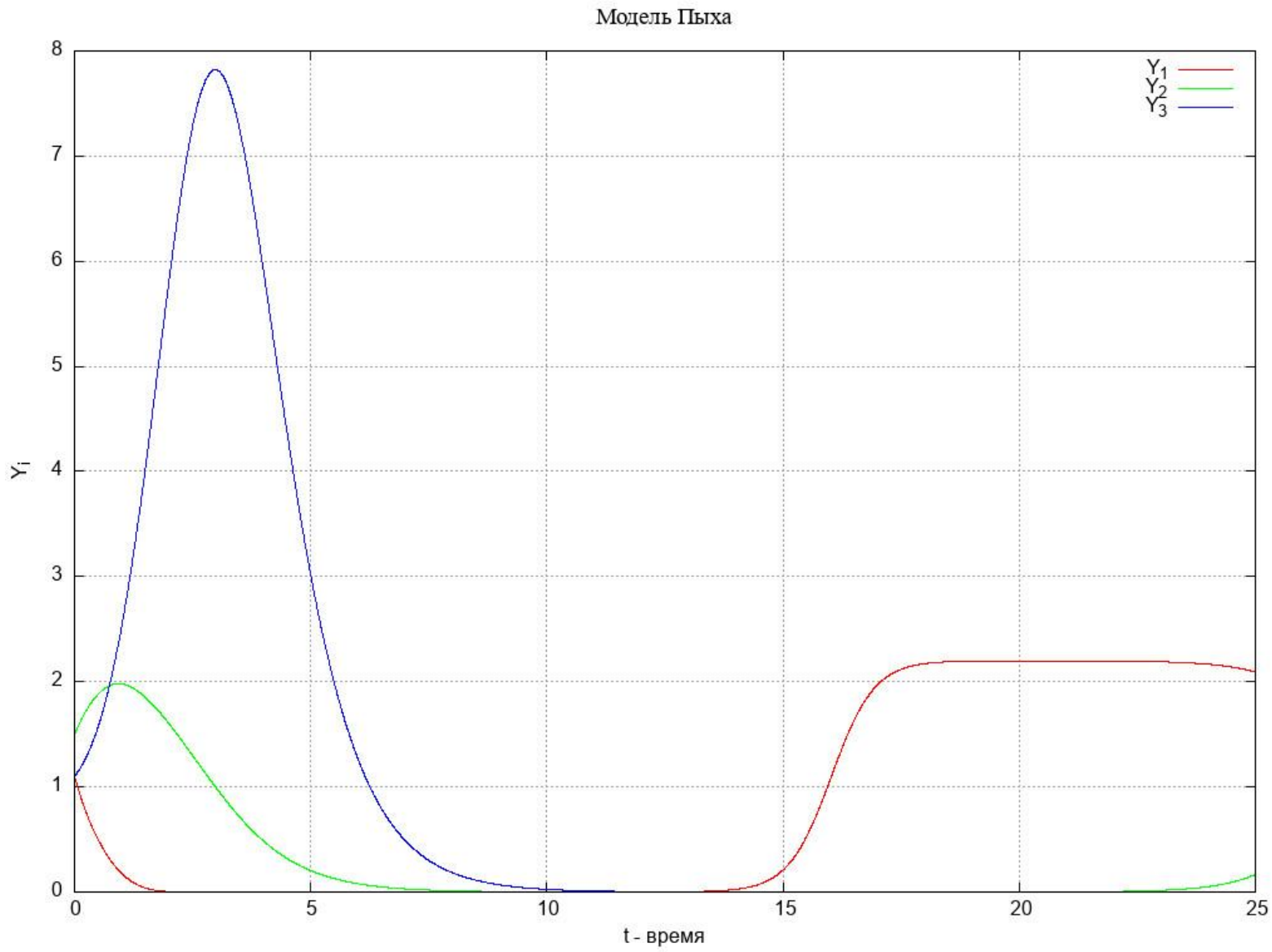
2. Построение графиков

Найти при $E_i = 1,\ i = 1, 2, 3$ такие значения μ, при которых возникают периодические и хаотические решения $y_i(t)$

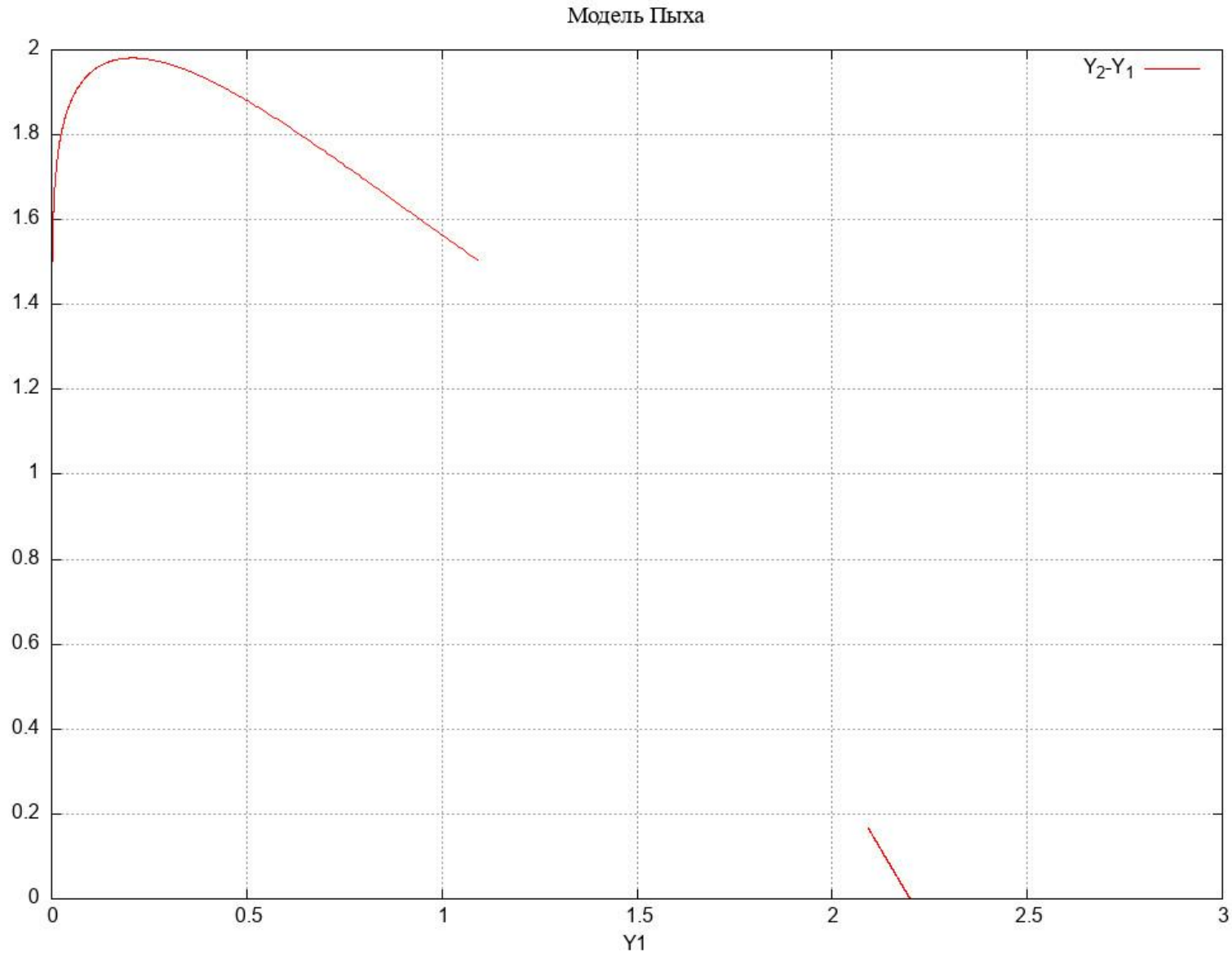
A. Хаотический режим

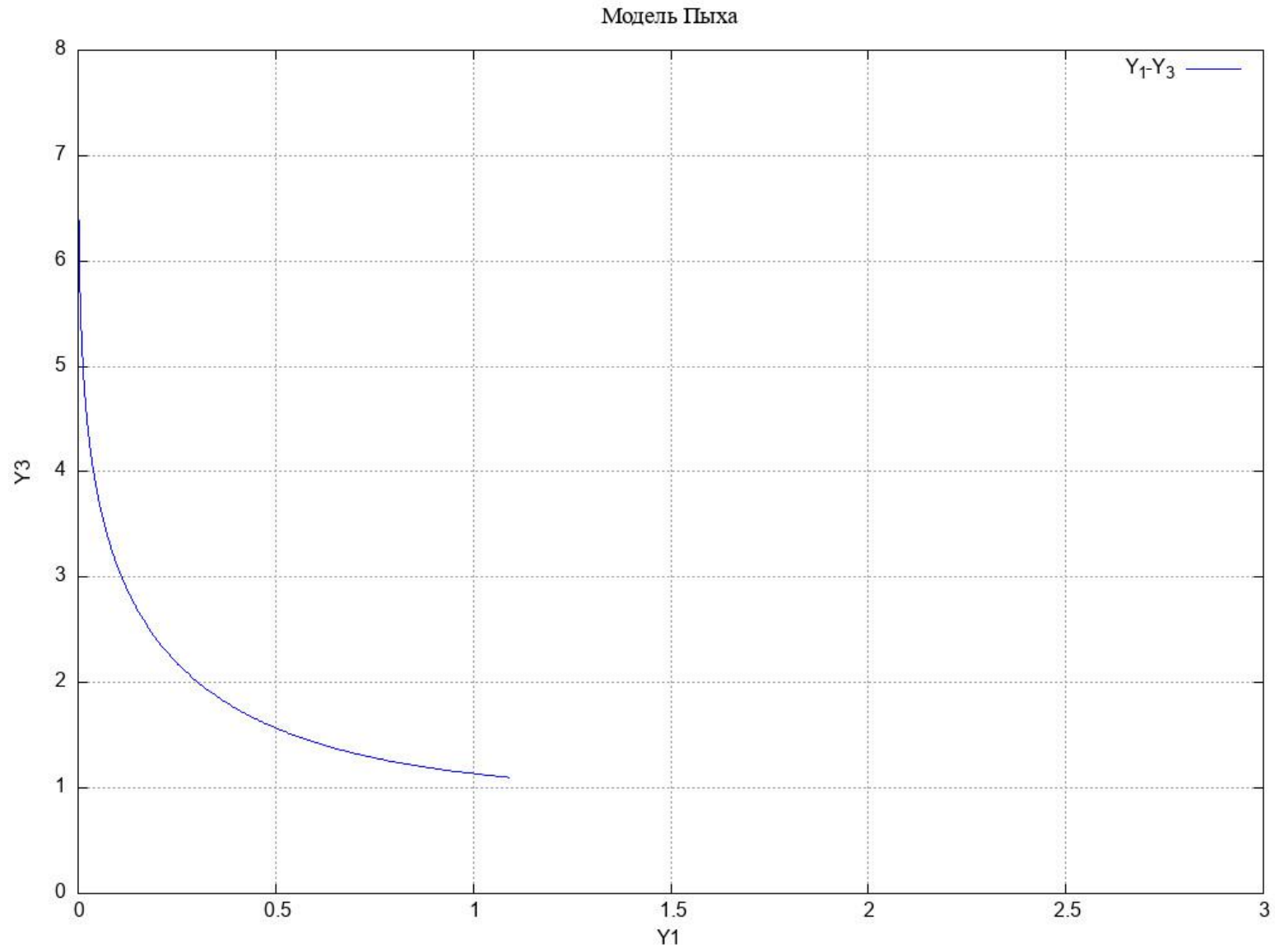
$\mu = 0.2$

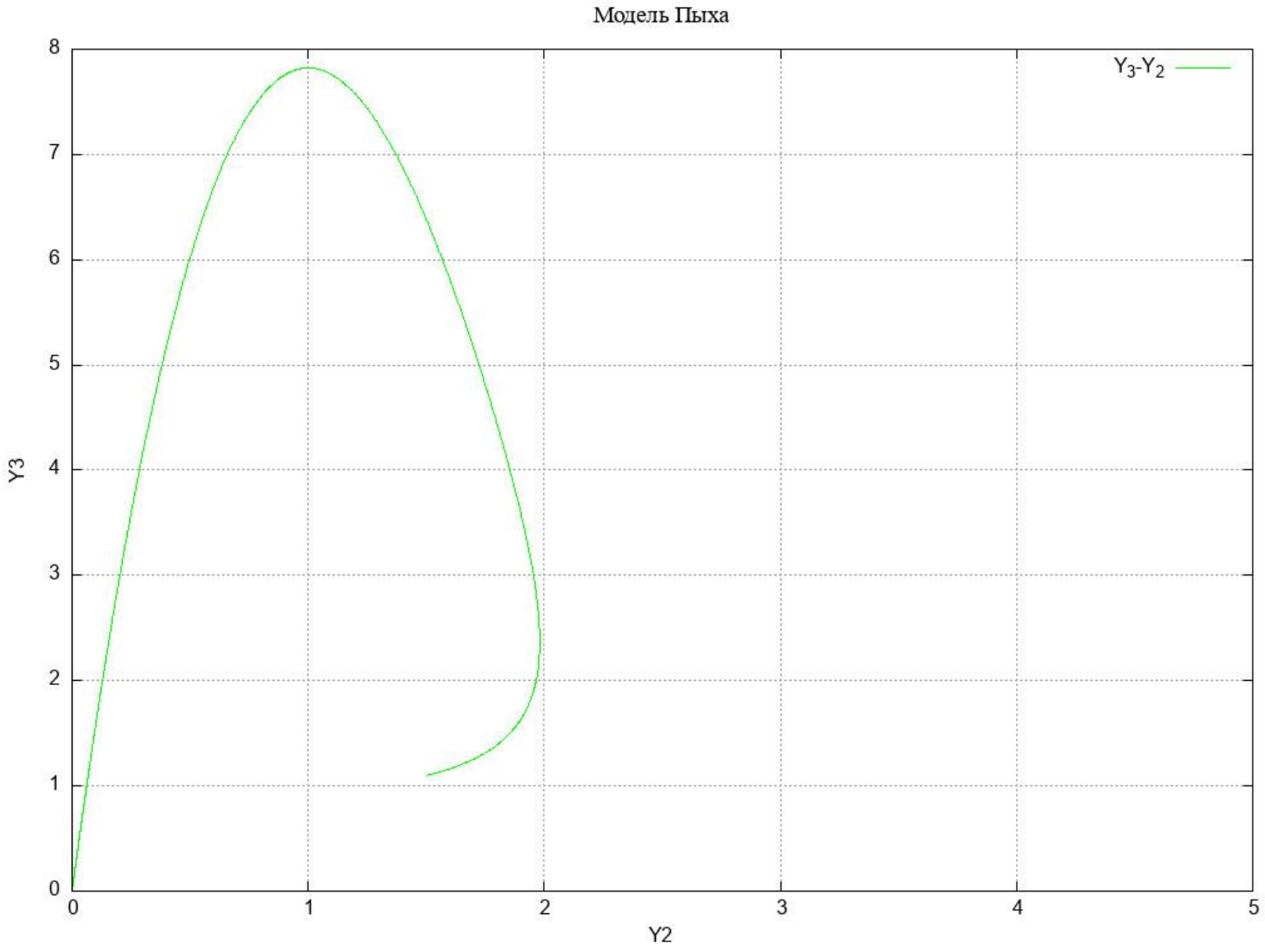
$Time = 25$, $h = 0.005$- временной промежуток и шаг по времени



Фазовые диаграммы


Модель Пыха
Y2-Y1
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5
3
Y1

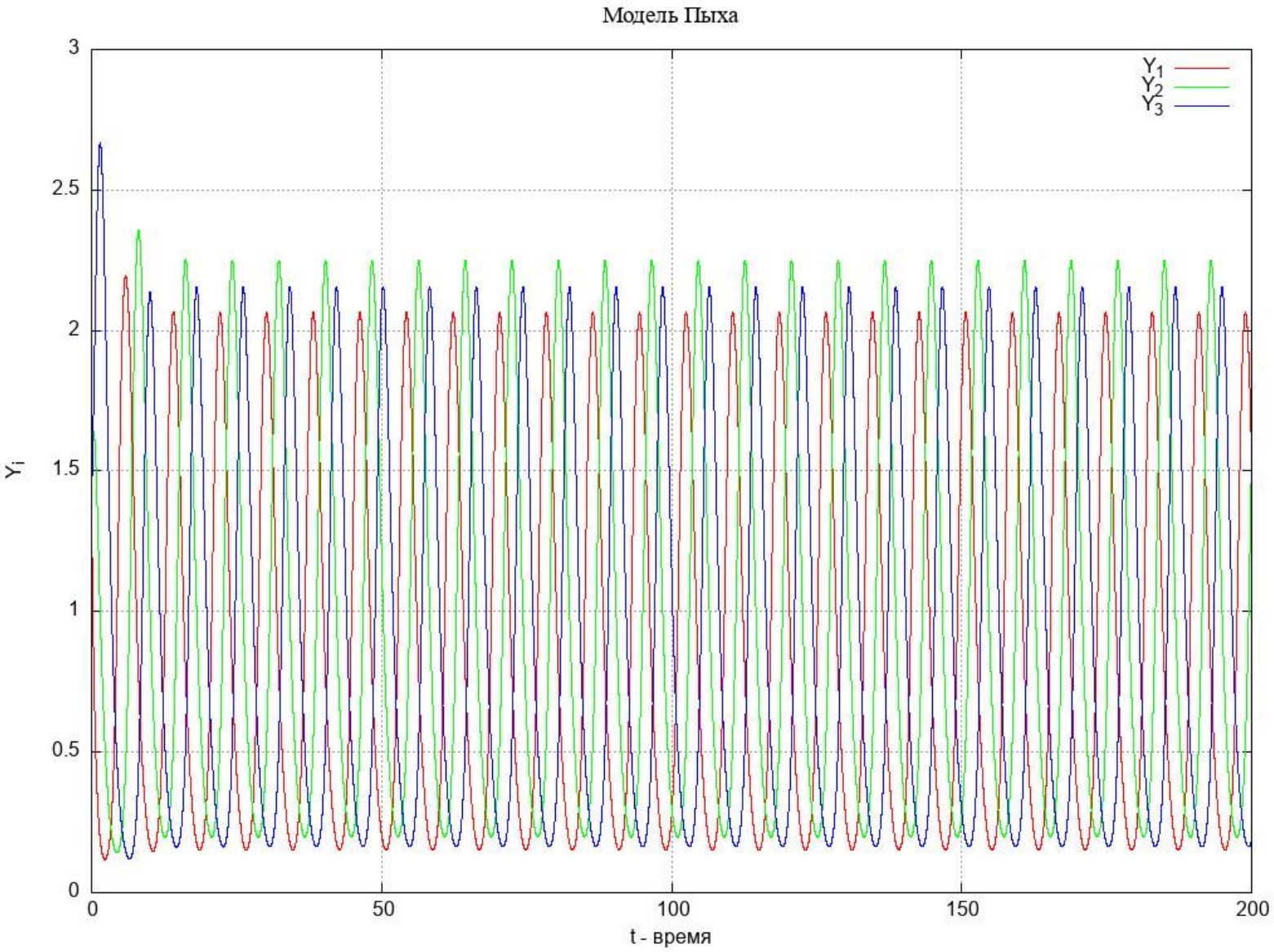

Модель Пыха
Y1-Y3
Y3
Y1
0
1
2
3
4
5
6
7
8
0
0.5
1
1.5
2
2.5
3



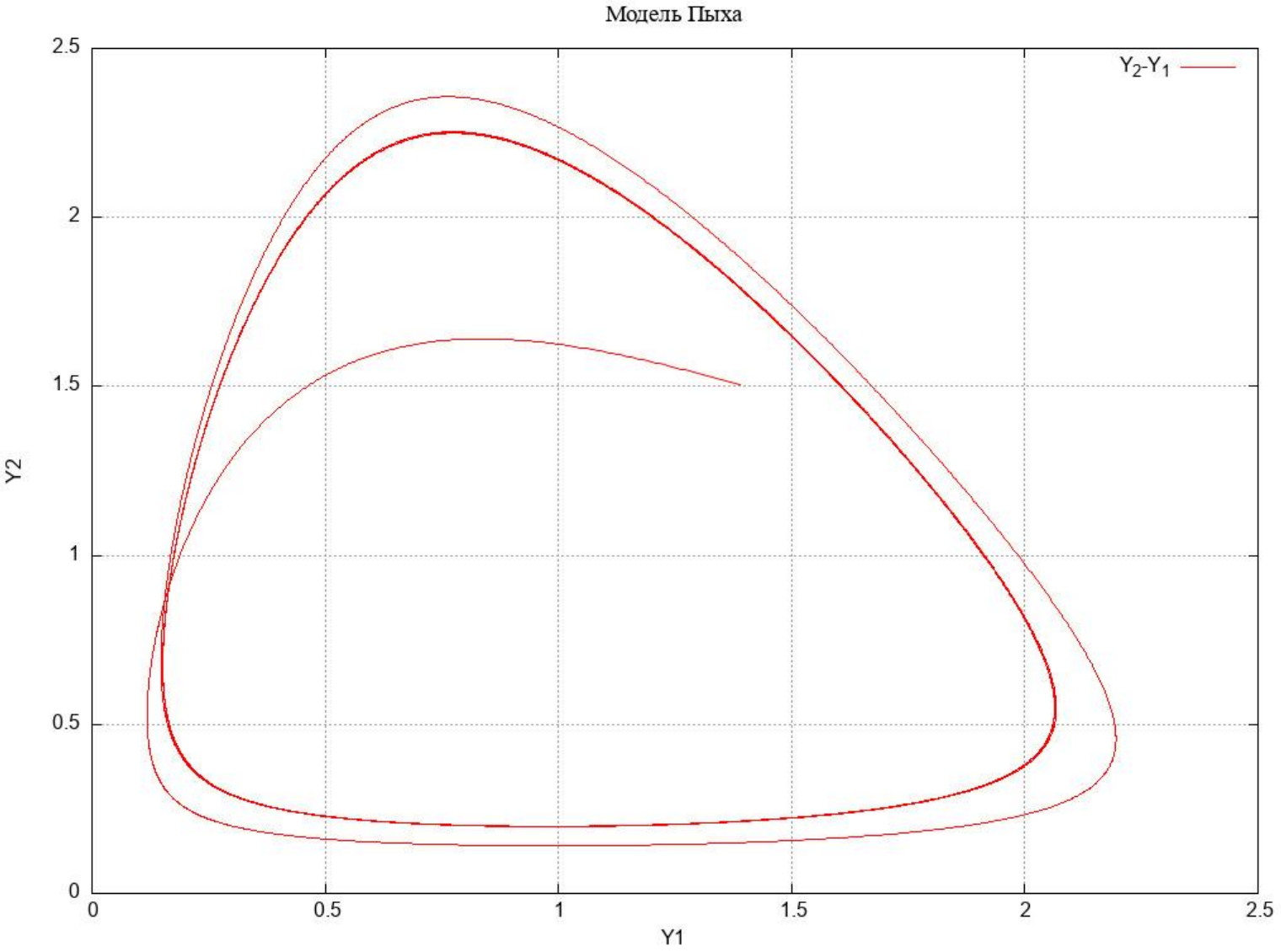
Б. Периодический режим

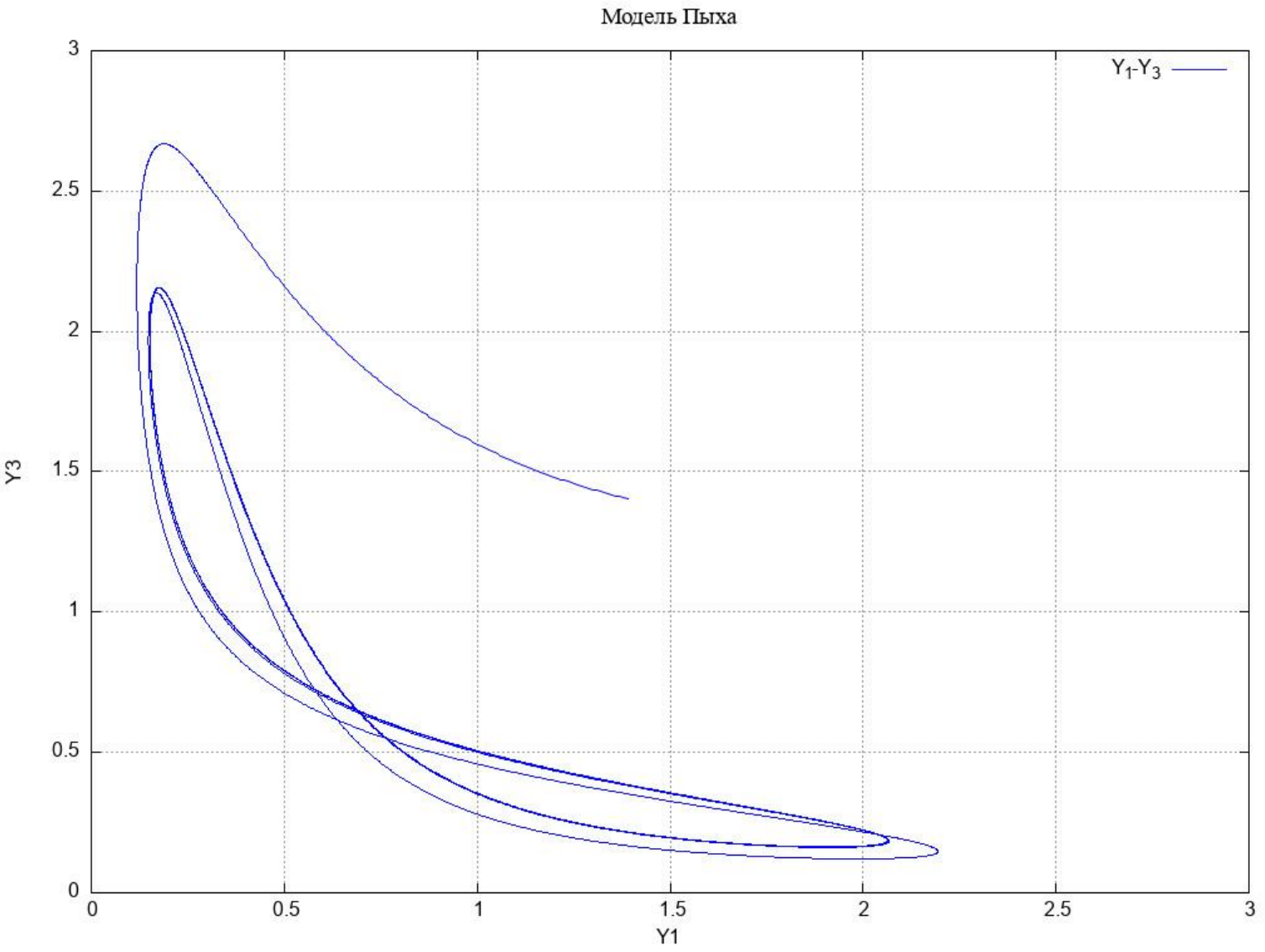
$\mu = 0.05$

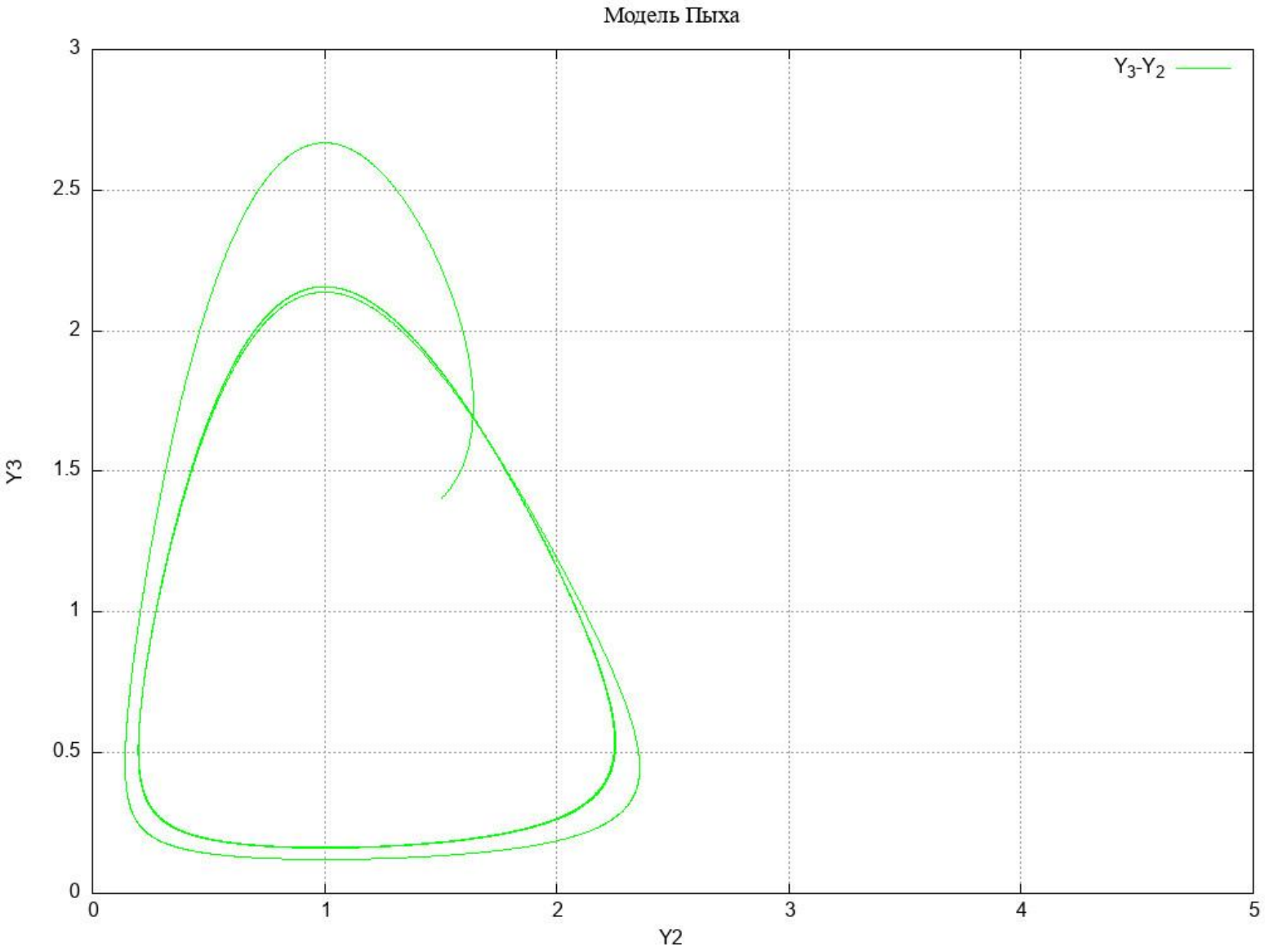
$Time = 200$, $h = 0.005$ **-** временной промежуток и шаг по времени



Фазовые диаграммы


Модель Пыха
Y2-Y1
Y2
Y1
0
0.5
1
1.5
2
2.5


Модель Пыха
Y1-Y3
Y3
Y1
0
0.5
1
1.5
2
2.5
3



ПРОГРАММА

```
#include <iostream>
#include <fstream>
```

```
#include <iomanip>

using namespace std;

const double eps = 0.5;
const int N = 3; // количество уравнений

// метод Рунге-Кутта
// h - шаг по времени
// E1,E2,E3 и m - параметры
double* Method(double E1, double E2, double E3, double m, double* Y, double
h) {
    double* NextY = new double[N];
    double* Arr = new double[N];
   // k1,k2,k3,k4 - приближающие угловые коэффициенты

    double* k1 = new double[N]; // в исходной точке
    k1[0] = Y[0] * ((3 - 4 * m) - Y[0] - Y[1] - Y[2]);  // посчитаем правую часть
уравнения
    k1[1] = Y[1] * (-E2 + Y[0] + 2 * m * Y[1]);
    k1[2] = Y[2] * (-E3 + Y[1]);
    Arr[0] = Y[0] + h / 2 * k1[0];
    Arr[1] = Y[1] + h / 2 * k1[1];
    Arr[2] = Y[2] + h / 2 * k1[2];

    // на половинном шаге
    double* k2 = new double[N];
    k2[0] = Arr[0] * ((3 - 4 * m) - Arr[0] - Arr[1]  - Arr[2]);
    k2[1] = Arr[1] * (-E2 + Arr[0] + 2 * m * Arr[1]);
    k2[2] = Arr[2] * (-E3 + Arr[1]);
    Arr[0] = Y[0] + h / 2 * k2[0];
    Arr[1] = Y[1] + h / 2 * k2[1];
    Arr[2] = Y[2] + h / 2 * k2[2];

    //тоже на половинном шаге, но по уточненному значению углового
коэффициента k2 вместо k1
    double* k3 = new double[N];
```

```
    k3[0] = Arr[0] * ((3 - 4 * m) - Arr[0] - Arr[1] - Arr[2]);
    k3[1] = Arr[1] * (-E2 + Arr[0] + 2 * m * Arr[1]);
    k3[2] = Arr[2] * (-E3 + Arr[1]);
    Arr[0] = Y[0] + h * k3[0];
    Arr[1] = Y[1] + h * k3[1];
    Arr[2] = Y[2] + h * k3[2];

    // на целом шаге по предыдущему значению k3
    double* k4 = new double[N];
    k4[0] = Arr[0] * ((3 - 4 * m) - Arr[0] - Arr[1]  - Arr[2]);
    k4[1] = Arr[1] * (-E2 + Arr[0] + 2 * m * Arr[1]);
    k4[2] = Arr[2] * (-E3 + Arr[1]);

    NextY[0] = Y[0] + h / 6 * (k1[0] + 2 * k2[0] + 2 * k3[0] + k4[0]);
    NextY[1] = Y[1] + h / 6 * (k1[1] + 2 * k2[1] + 2 * k3[1] + k4[1]);
    NextY[2] = Y[2] + h / 6 * (k1[2] + 2 * k2[2] + 2 * k3[2] + k4[2]);

    return NextY;

}


int main()
{

    double* Y = new double[N];
    double E1,E2,E3; // параметры const
    double m; // меняющийся параметр
    double h = 0.01; // шаг по времени
    double Time = 100; //
    cout << "Введите параметры\n";
    cout << "E1: ";
    cin >> E1;
    cout << "E2: ";
    cin >> E2;
    cout << "E3: ";
    cin >> E3;
```

```
    cout << "m: ";
    cin >> m;

    ofstream modelP;
    modelP.open("file.txt");
    // начальные условия Задачи Коши
    Y[0] = E2 - 2 * m * E3 + eps;
    cout << "Y1 = " << Y[0] << endl;
    Y[1] = E3 + eps;
    cout << "Y2 = " << Y[1] << endl;
    Y[2] = 3 - 4 * m + 2 * m * E3 - E2 - E3 + eps;
    cout << "Y3 = " << Y[2] << endl;

    modelP << setfill(' ');
    modelP << "# E1 = " << E1 << " E2 = " << E2 << " E3 = " << E3 << " m = "
<< m << " Eps = " << eps << endl;
    modelP << setw(10) << left << "# Time" << setw(20) << left << "Y_1" <<
setw(20) << left << "Y_2" << setw(20) << left << "Y_3" << endl;

    double t = 0;
    while (t < Time) {

            Y = Method(E1, E2, E3, m, Y, h);
            t += h;

        modelP << setfill(' ');
        modelP << setw(10) << left << t << setw(20) << left << Y[0] << setw(20)
<< left << Y[1] << setw(20) << left << Y[2] << endl;
    }
    return 0;
}
```