МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт цифровых технологий, электроники и физики

Кафедра вычислительной техники и электроники (ВТиЭ)

Лабораторная работа № 4

**Моделирование динамических систем.**

Выполнил студент 595 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Лаптев

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ П.Н. Уланов

Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2021

**Цель работы:** изучить и преобразовать дифференциальное уравнение аналитически, смоделировать систему численно с помощью программной реализации метода Рунге-Кутты четвёртого порядка.

**Задание 13.**

Определить на фазовой плоскости особые точки динамической системы, заданной дифференциальной моделью 2-го порядка, и их характер

**Решение:** Проводим обезразмеривание:

Откуда, . Обезразмеривание завершено.

Далее производим замену, чтобы получить систему уравнений первого порядка:

После чего, правая часть каждого уравнения системы приравнивается к нулю для нахождения «особых» точек.

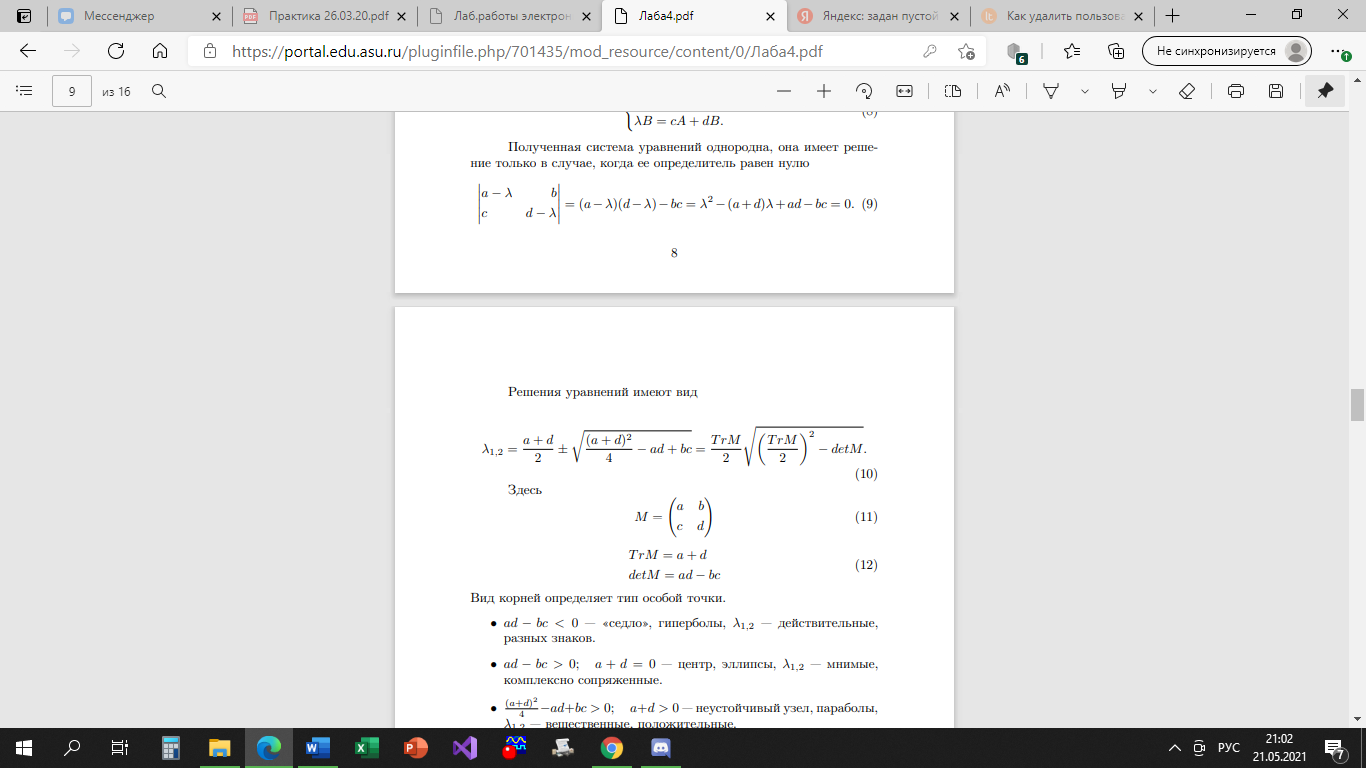
Откуда получаем две особые точки: .

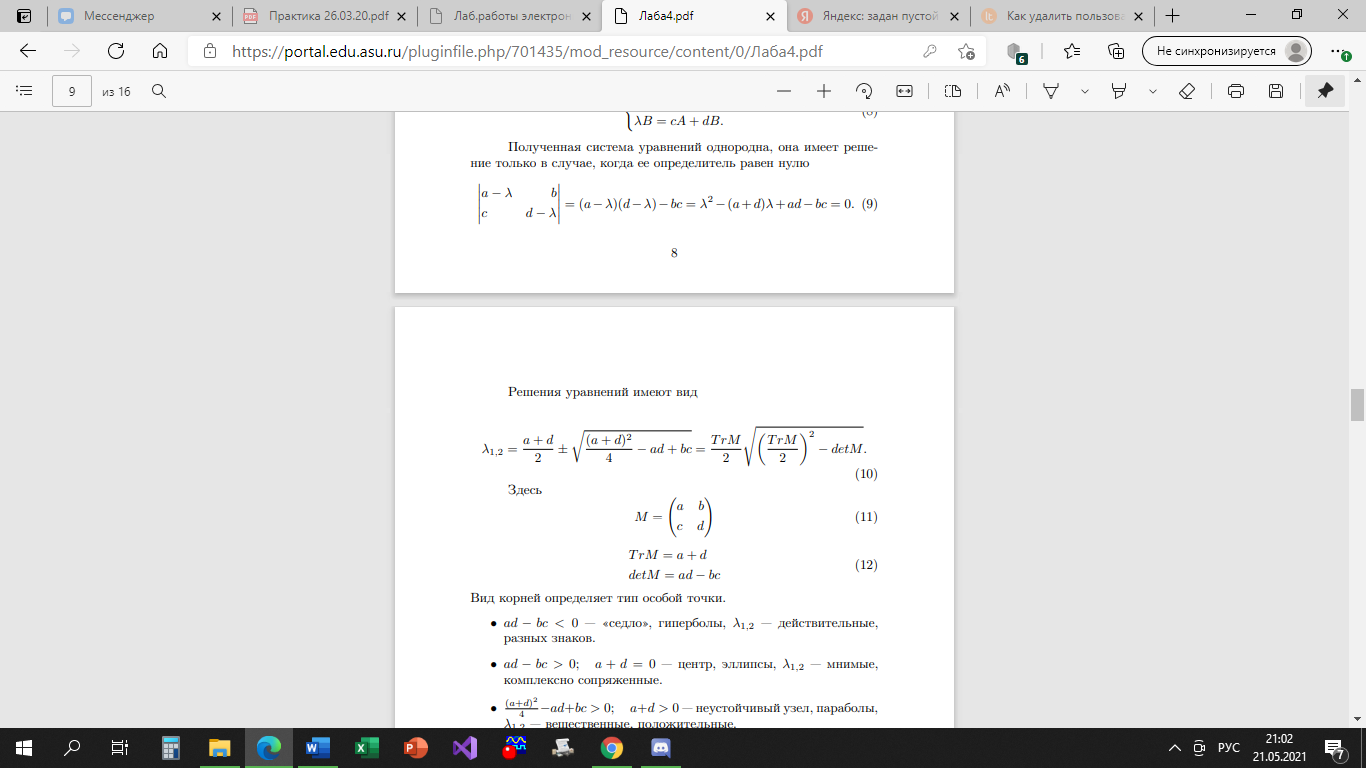
Чтобы определить типы особых точек надо найти коэффициенты a, b, c, d и решить квадратное уравнение:

. Откуда a = 0, b = 1, c = -2x-1, d = .

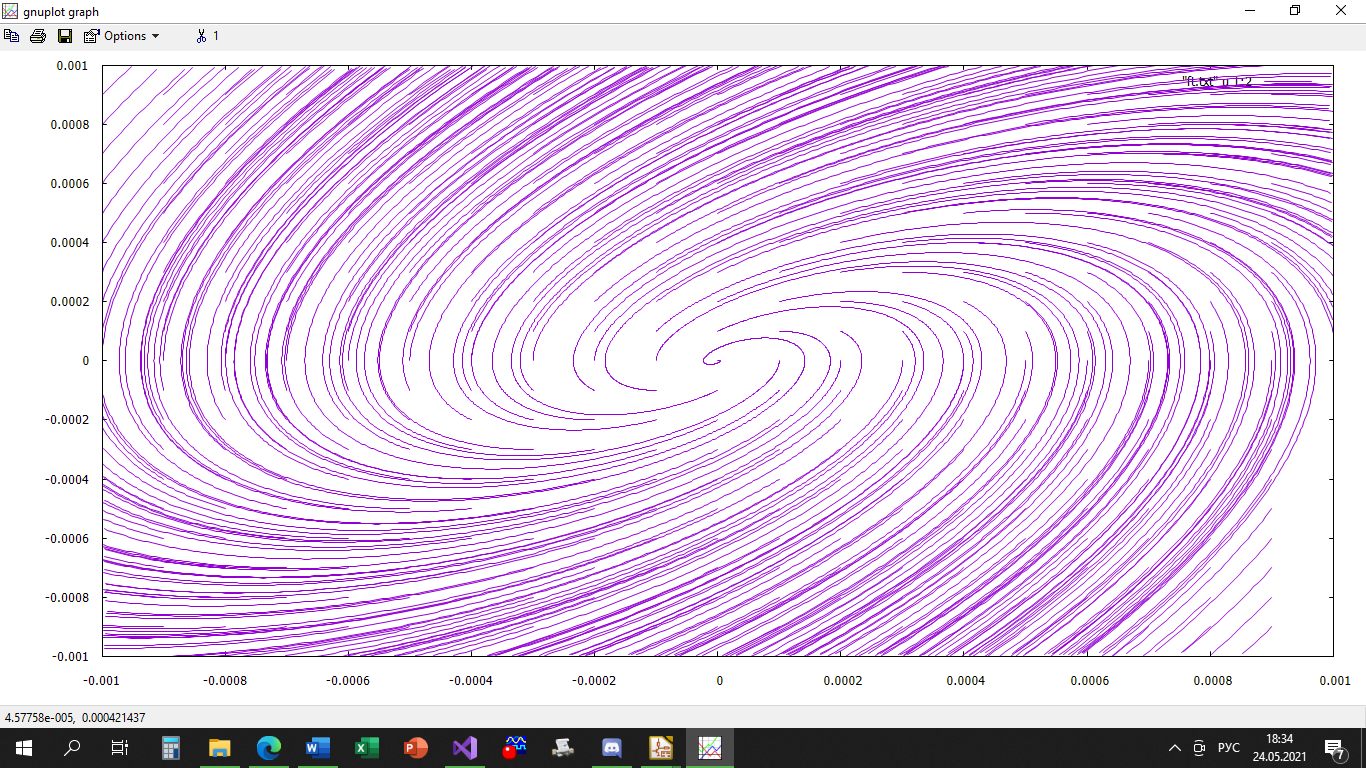
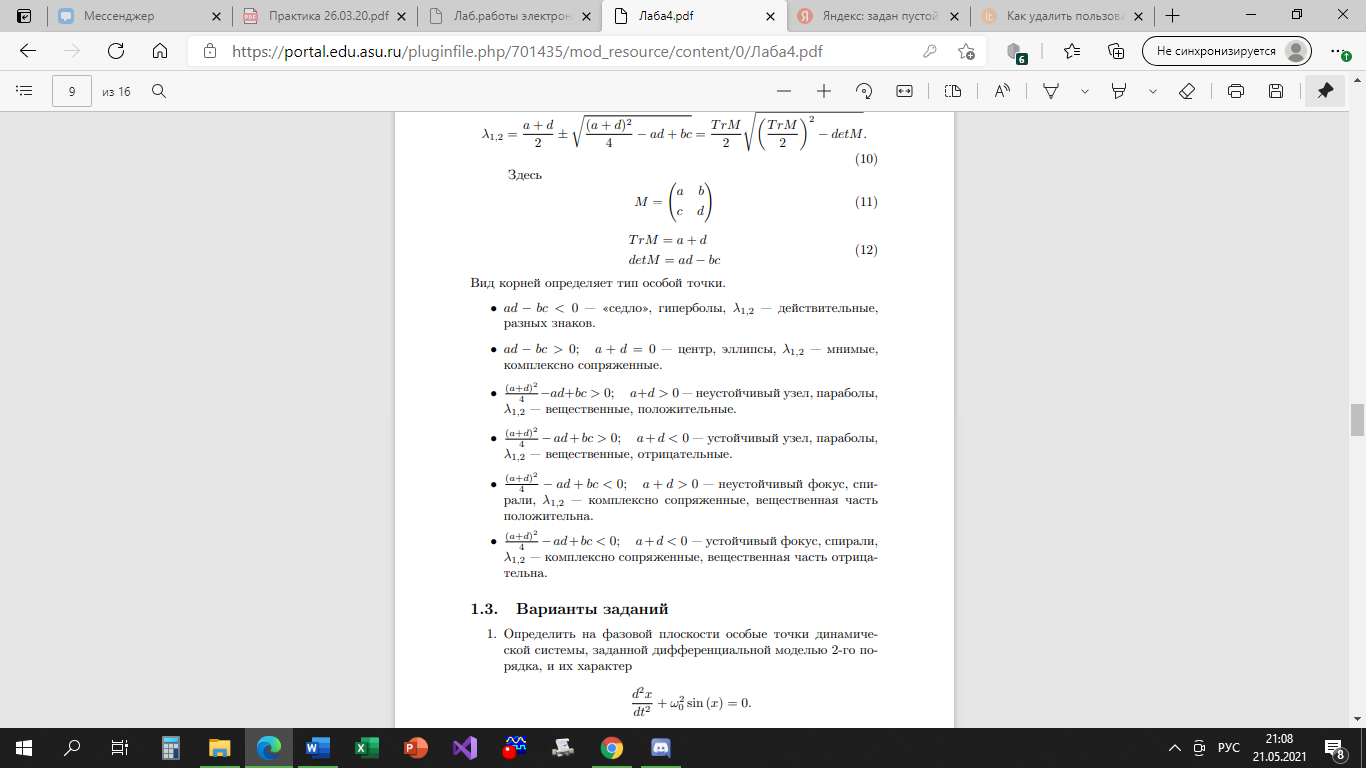
После преобразований получаем квадратное уравнение: .

Решения уравнения имеют вид:

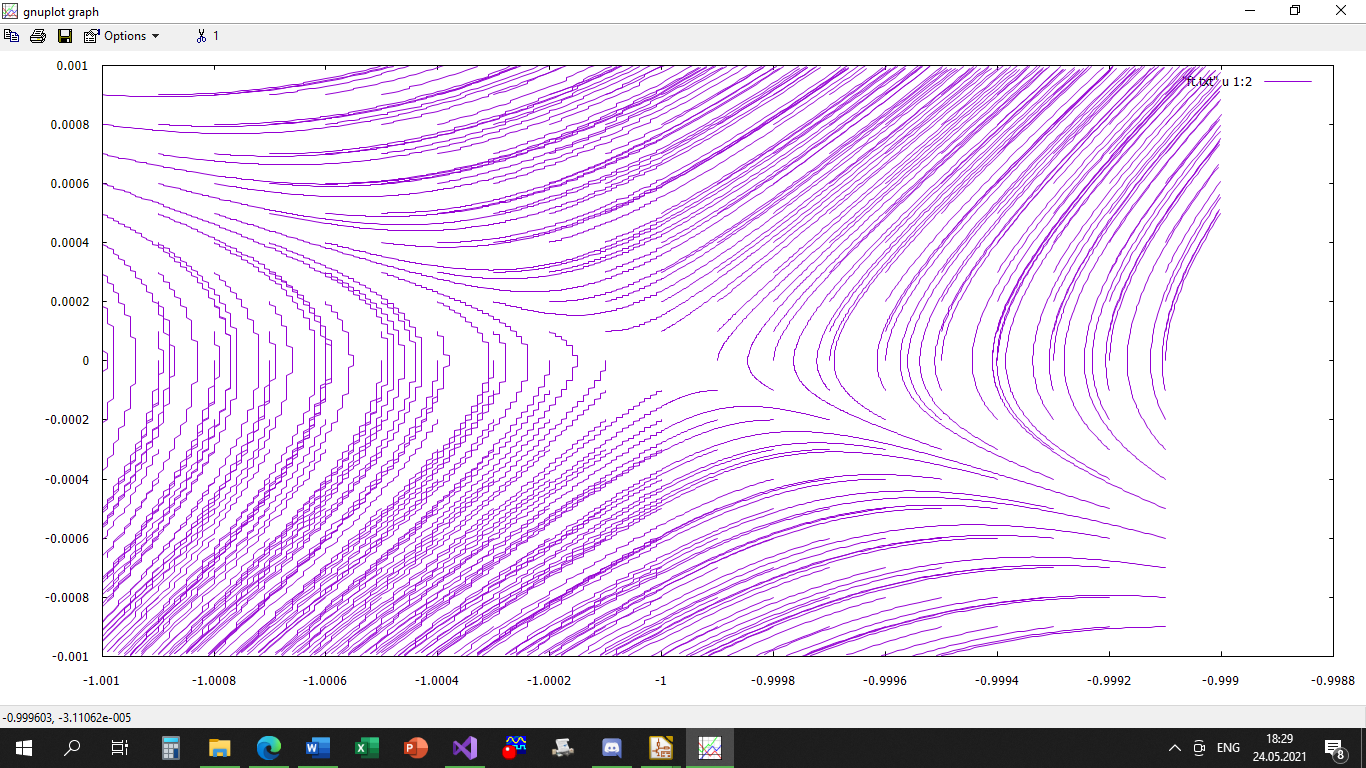


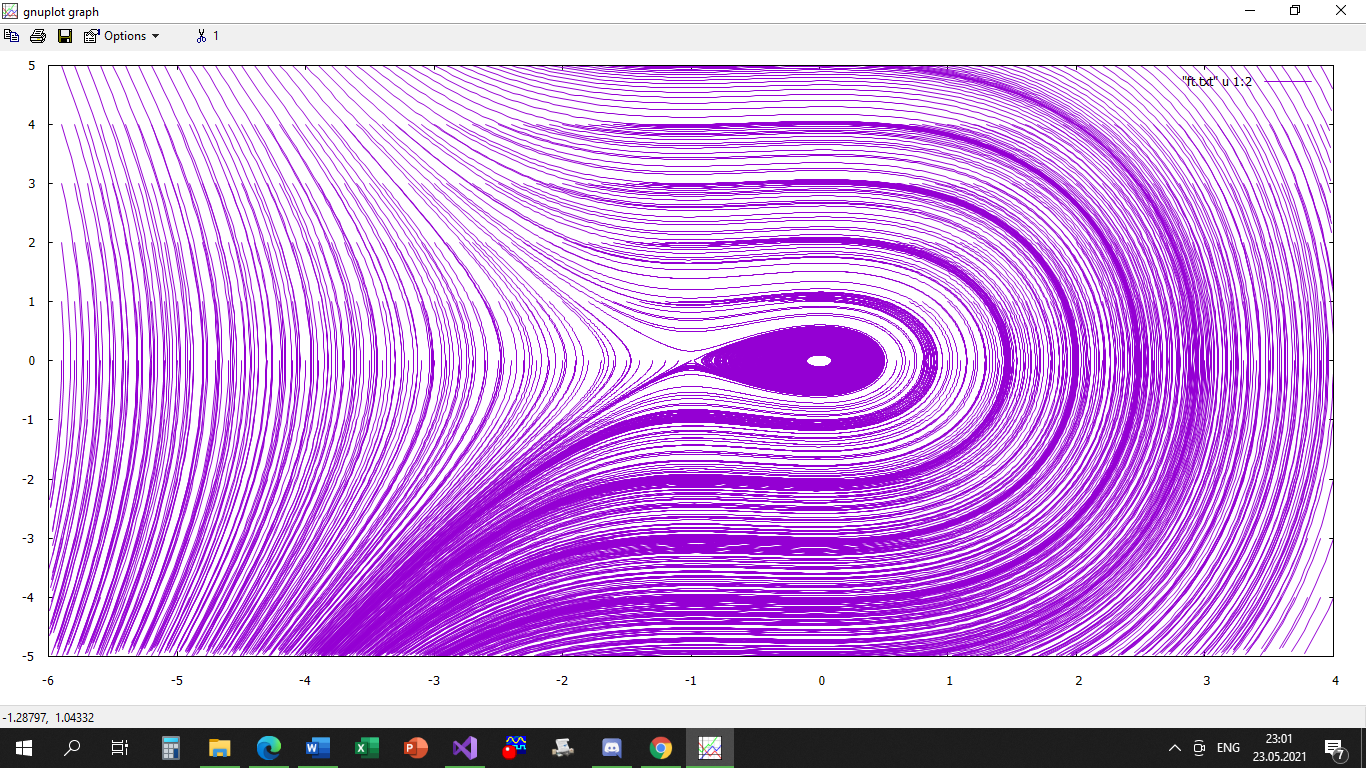


Тип особой точки определяется по его корням. В итоге, после подстановки своего уравнения и координат «особой» точки я получил типы точек: седло (для x = -1, y = 0), неустойчивый фокус (для x = 0, y = 0).



*Рис. 1. Фазовые траектории для «особой» точки типа «неустойчивый фокус», при delta = 1.*

*Рис. 3. Фазовые траектории для «особой» точки типа «седло», при delta = 1.*



*Рис. 3. Фазовый портрет.*

Для «неустойчивого фокуса» явственно видны характерные для этой точки закручивающиеся спирали, для «седла» на графике фазовых траекторий присутствуют траектории гипербол для всех четырёх направлений.

Также на фазовом портрете без особого труда можно заметить обе «особые» точки и их влияние друг на друга.

**Вывод:** в ходе выполнения лабораторной работы данное дифференциальное уравнение было преобразовано и смоделировано программно с применением метода Рунге-Кутты четвёртого порядка.

**Приложение**

**Код на С++**

#include <iostream>

#include <cmath>

#include <string>

#include <fstream>

double f(double, double, double); // Интерфейс функции f, это функция при x'

double g(double, double, double); // Интерфейс функции g, это при y'

using namespace std;

ofstream fout("ft.txt");

int main() {

int k = 0; //k - переменная-счетчик количества точек в графике, используется для ограничения

double t = 0, X, Y, h = 0.01; // t - время, X и Y - значения функций, h - шаг моделирования

double k1, k2, k4, k3, q1, q2, q4, q3; //служебные переменные

double X0 = -1, Y0 = 0; //С помощью X0 и Y0 задаются координаты особой точки для удобства

std::cout.precision(10); //Задается точность вывода чисел в потоке stdout

for (double j = -5; j <= 5; j += 0.01)

for (double i = -1; i <= 1; i += 0.1)

{//Циклы, двигающие начальную точку

X = X0 + j \* 10;

Y = Y0 + i \* 10; //Задание начальной точки моделирования

t = 0;

k = 0;

for (; (fabs(X - X0) <= 5 && fabs(Y - Y0) <= 5) && k++ < 1e5; t += h)

{//Цикл построения траектории

fout << X << ' ' << Y << "\n"; // Вывод координат точки

k1 = h \* g(t, X, Y);

q1 = h \* f(t, X, Y);

k2 = h \* g(t + h / 2.0, X + q1 / 2.0, Y + k1 / 2.0);

q2 = h \* f(t + h / 2.0, X + q1 / 2.0, Y + k1 / 2.0);

k3 = h \* g(t + h / 2.0, X + q2 / 2.0, Y + k2 / 2.0);

q3 = h \* f(t + h / 2.0, X + q2 / 2.0, Y + k2 / 2.0);

k4 = h \* g(t + h, X + q3, Y + k3);

q4 = h \* f(t + h, X + q3, Y + k3);

Y = Y + (k1 + 2.0 \* k2 + 2.0 \* k3 + k4) / 6.0;

X = X + (q1 + 2.0 \* q2 + 2.0 \* q3 + q4) / 6.0;

}

fout << "\n";

}

}

double f(double t, double x, double y)

{

return y;

}

double g(double t, double x, double y)

{

return 0.01 \* y - x \* x - x;

}