Лабораторная работа №5

Модель эпидемии (SIR)

Джахангиров Илгар Залид оглы

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Джахангиров Илгар Залид оглы
- студент
- Российский университет дружбы народов
- [1032225689@pfur.ru]

Цель работы

Построить модель SIR в *xcos* и OpenModelica.

Задание

1. Реализовать модель SIR в в *xcos*;
2. Реализовать модель SIR с помощью блока Modelica в в *xcos*;
3. Реализовать модель SIR в OpenModelica;
4. Реализовать модель SIR с учётом процесса рождения / гибели особей в xcos (в том числе и с использованием блока Modelica), а также в OpenModelica;
5. Построить графики эпидемического порога при различных значениях параметров модели (в частности изменяя параметр μ);
6. Сделать анализ полученных графиков в зависимости от выбранных значений параметров модели.

Задача о распространении эпидемии описывается системой дифференциальных уравнений:

$$\begin{cases} \dot{s} = -\beta s(t) i(t); \\ \dot{i} = \beta s(t) i(t) - \nu i(t); \\ \dot{r} = \nu i(t), \end{cases}$$

где β – скорость заражения, ν – скорость выздоровления.

Зафиксируем начальные данные:

$\beta = 1,\ \nu = 0,3, s(0) = 0,999,\ i(0) = 0,001,\ r(0) = 0.$

В меню Моделирование, Установить контекст зададим значения переменных β и ν (рис.

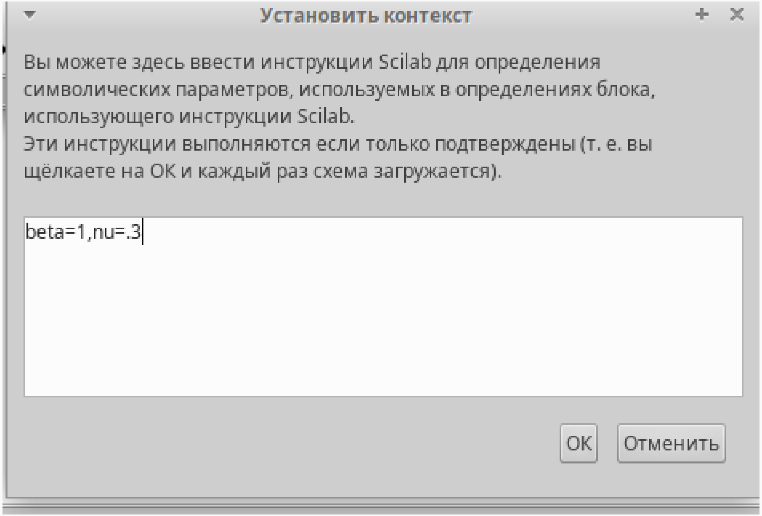


Figure 1: Задание переменных окружения в xcos

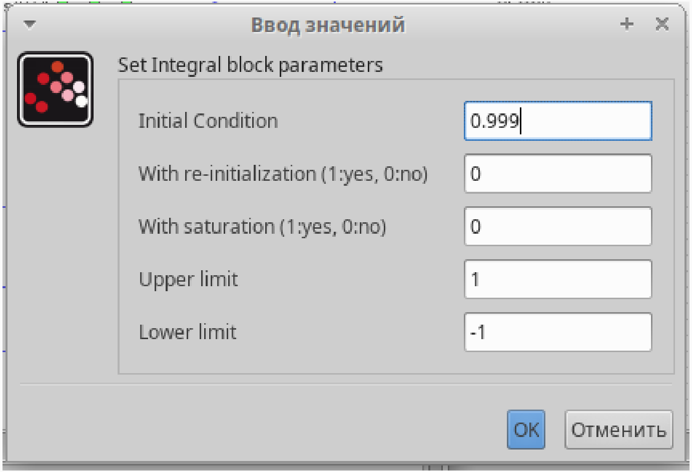


Figure 3: Задание начальных значений в блоках интегрирования

Выполнение лабораторной работы

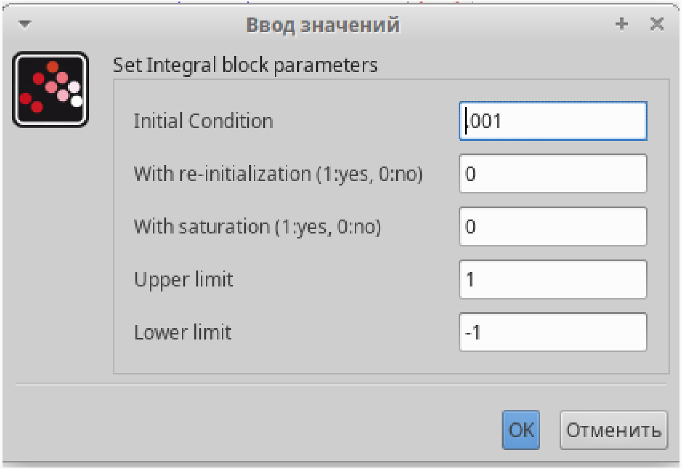


Figure 4: Задание начальных значений в блоках интегрирования

В меню Моделирование, Установка зададим конечное время интегрирования, равным

Эпидемический порог модели SIR при $\beta = 1, \nu = 0.3$

Figure 5: Эпидемический порог модели SIR при $\beta = 1, \nu = 0.3$

Готовая модель SIR представлена на рис.

Для реализации модели SIR с помощью языка Modelica помимо блоков `CLOCK_c`, `CSCOPE`, `TEXT_f` и `MUX` требуются блоки `CONST_m` — задаёт константу; `MBLOCK` (Modelica generic) — блок реализации кода на языке Modelica. Задаём значения переменных β и ν .

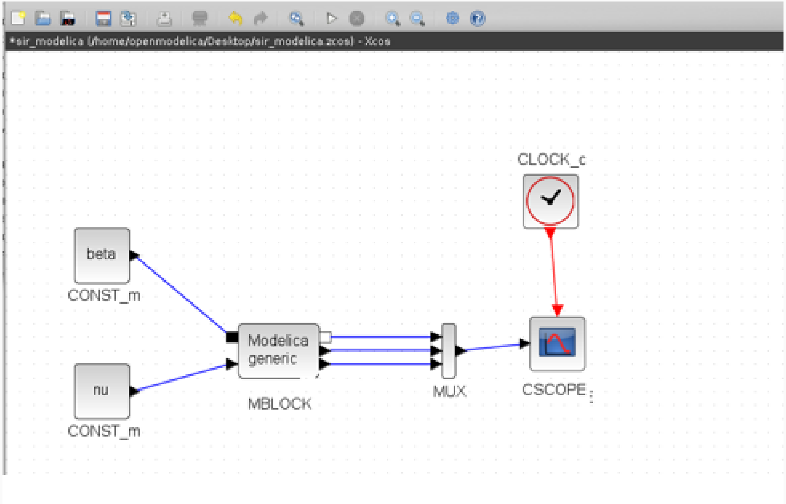
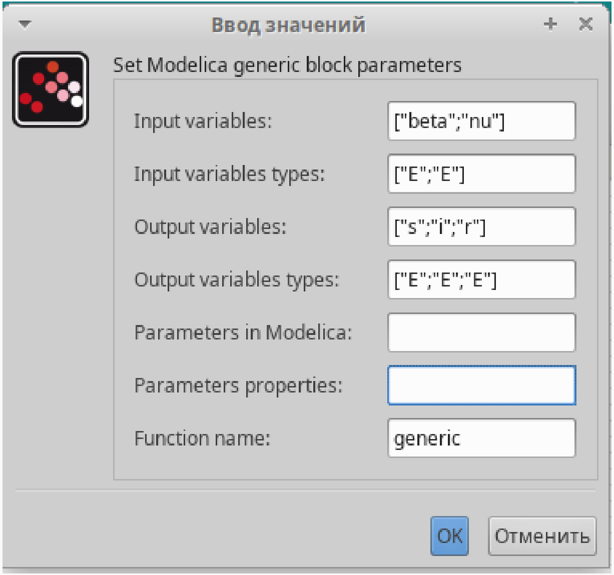


Figure 6: Модель SIR в xcos с применением блока Modelica


Ввод значений
Set Modelica generic block parameters
Input variables:
["beta";"nu"]
Input variables types:
["E";"E"]
Output variables:
["s";"i";"r"]
Output variables types:
["E";"E";"E"]
Parameters in Modelica:
Parameters properties:
Function name:
generic
OK
Отменить

Выполнение лабораторной работы

Function definition in Modelica

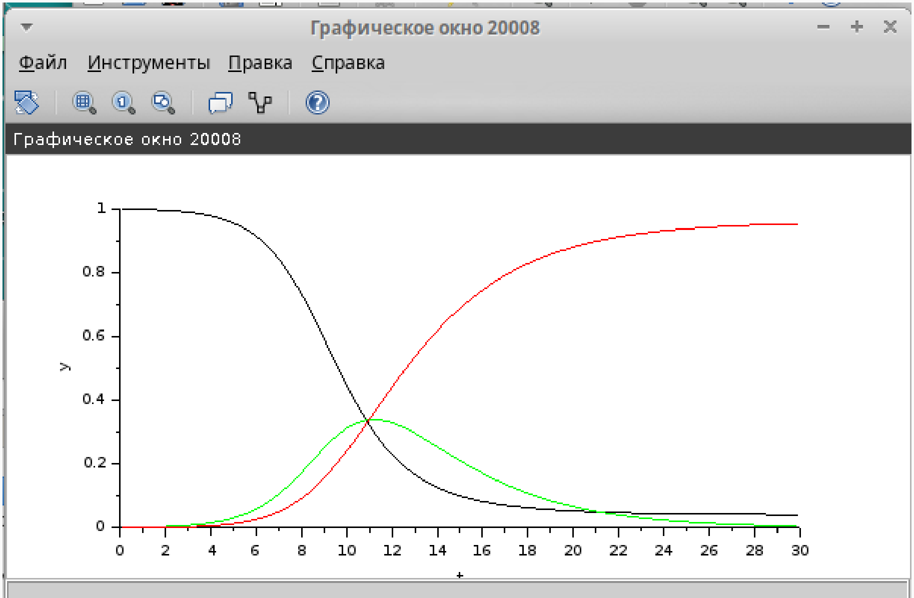
Here is a skeleton of the functions which you should edit

```
class generic
 ////automatically generated ////
    //input variables
    Real beta,nu;
    //output variables (комментируем, т.к.
    // начальные значения задаем в самом блоке):
    //Real s,i,r;
 ////do not modif above this line ////

  // начальные значения:
   Real s(start=.999),  i(start=.001), r(start= 0):

   // модель  SIR:
equation
    der(s)=-beta*s*i:
    der(i)=beta*s*i-nu*i:
    der(r)=nu*i:
end generic;
```

Выполнение лабораторной работы



Упражнение

В качестве упражнения нам надо построить модель SIR на OpenModelica. Синтаксис почти такой же как и на Modelica. Нужно задать параметры, начальные значения и систему дифференциальных уравнений.

```
parameter Real I_0 = 0.001;
parameter Real R_0 = 0;
parameter Real S_0 = 0.999;
parameter Real beta = 1;
parameter Real nu = 0.3;
parameter Real mu = 0.5;

Real s(start=S_0);
Real i(start=I_0);
Real r(start=R_0);
```

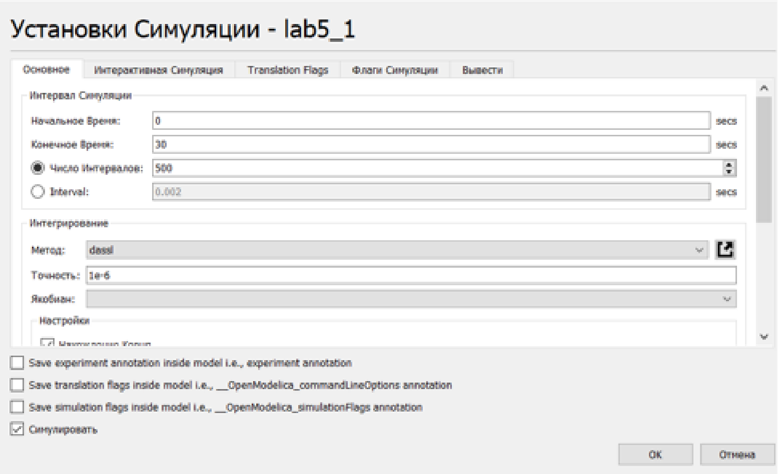
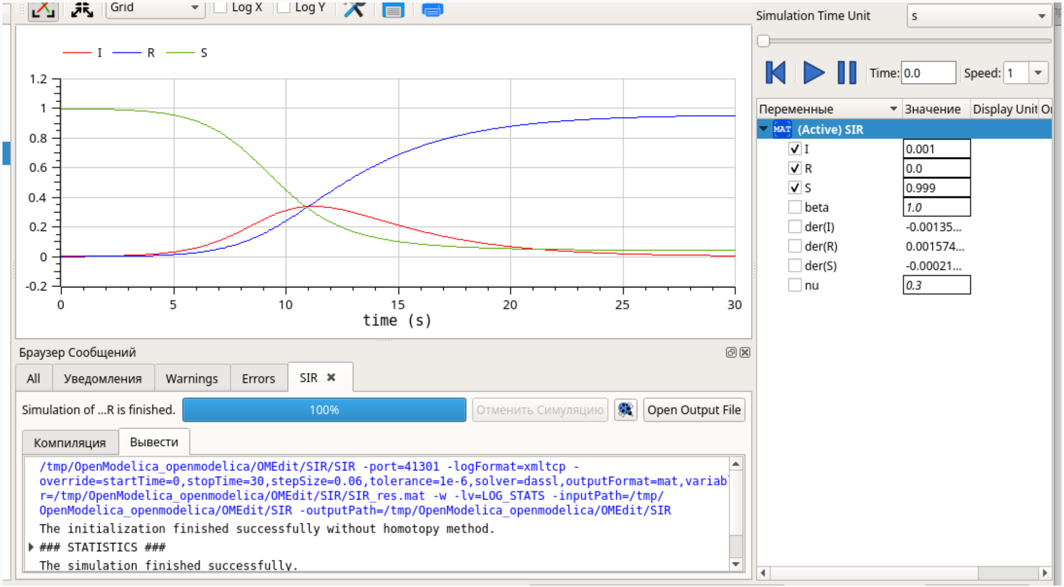


Figure 10: Установка симуляции в OpenModelica

В результате получаем следующий график. Он идентичен предыдущим графикам

Выполнение лабораторной работы



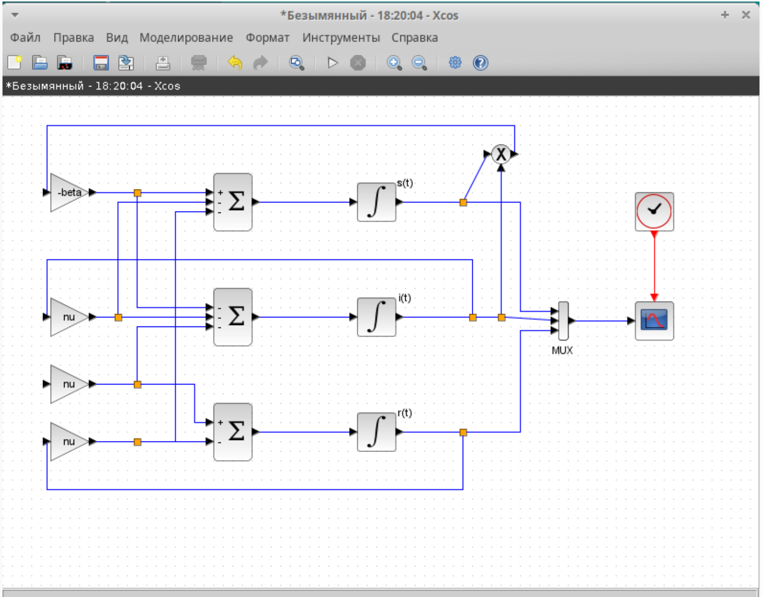
Предположим, что в модели SIR учитываются демографические процессы, в частности, что смертность в популяции полностью уравновешивает рождаемость, а все рожденные индивидуумы появляются на свет абсолютно здоровыми. Тогда получим следующую систему уравнений:

$$\begin{cases} \dot{s} = -\beta s(t)i(t) + \mu(N - s(t)); \\ \dot{i} = \beta s(t)i(t) - \nu i(t) - \mu i(t); \\ \dot{r} = \nu i(t) - \mu r(t), \end{cases}$$

где μ — константа, которая равна коэффициенту смертности и рождаемости.

Реализуем эту модель в *xcos*. Тут нам понадобятся три блока суммирования и 4 блока констант (добавляется константа ν).

Выполнение лабораторной работы



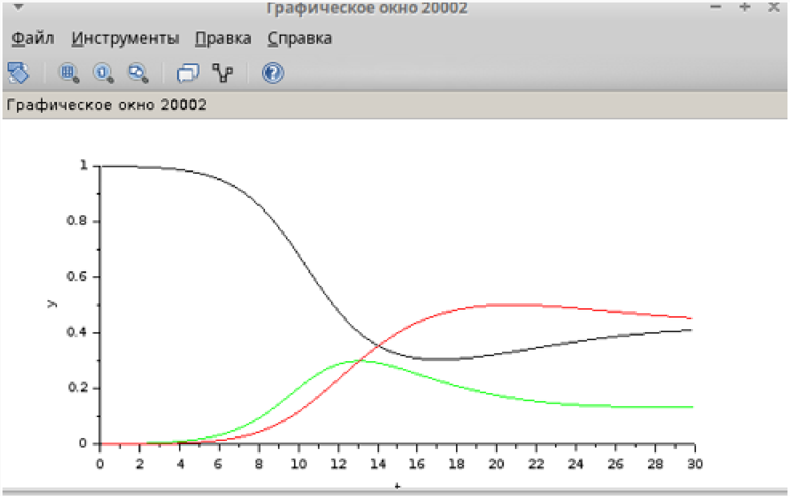


Figure 13: График модели SIR с учетом демографических процессов

![Модель SIR с учетом демографических процессов в xcos с применением блока Modelic(image/19.png)

Параметры блока Modelica представлены на рис.. Переменные на входе (“beta”, “nu”, “mu”) и выходе (“s”, “i”, “r”) блока заданы как внешние (“E”).

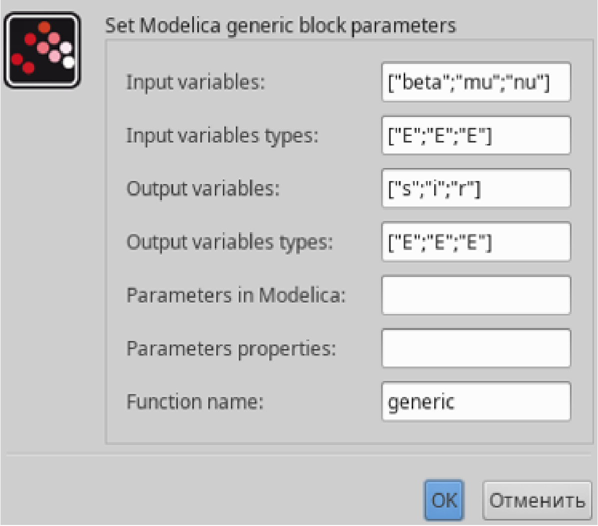
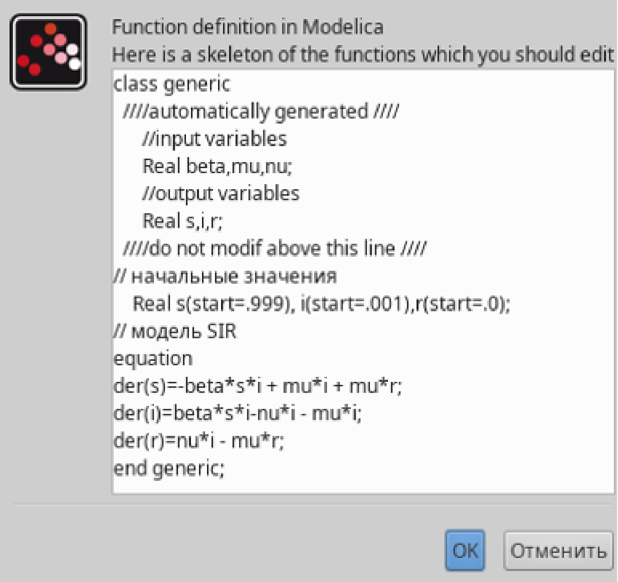


Figure 14: Параметры блока Modelica для модели SIR с учетом демографических процессов


Function definition in Modelica
Here is a skeleton of the functions which you should edit
class generic
////automatically generated ////
//input variables
Real beta,mu,nu;
//output variables
Real s,i,r;
////do not modif above this line ////
// начальные значения
Real s(start=.999), i(start=.001),r(start=.0);
// модель SIR
equation
der(s)=-beta*s*i + mu*i + mu*r;
der(i)=beta*s*i-nu*i - mu*i;
der(r)=nu*i - mu*r;
end generic;
OK
Отменить

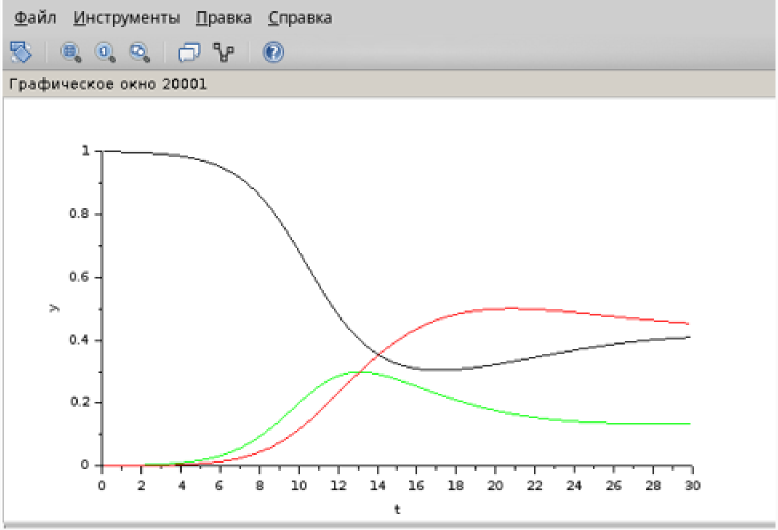
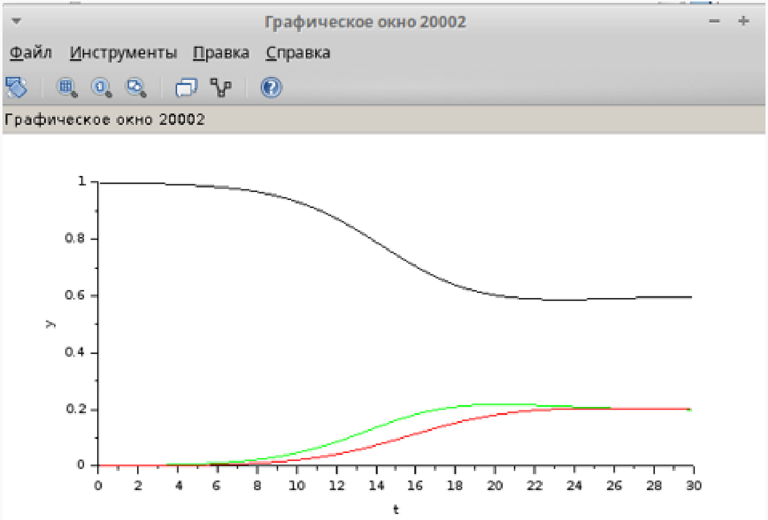


Figure 16: График модели SIR с учетом демографических процессов

Выполнив симуляцию, получим следующий график



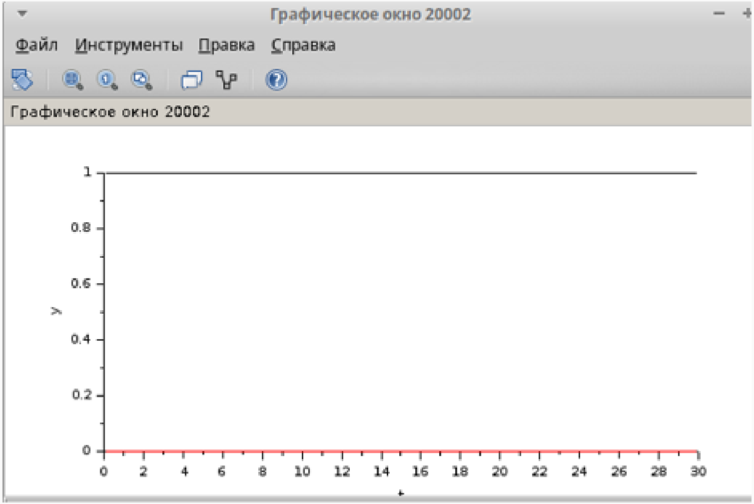


Figure 18: График модели SIR с учетом демографических процессов

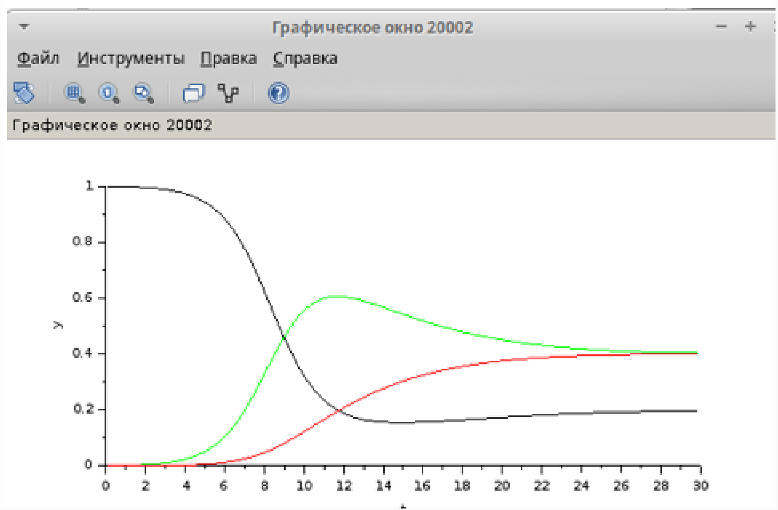


Figure 19: График модели SIR с учетом демографических процессов

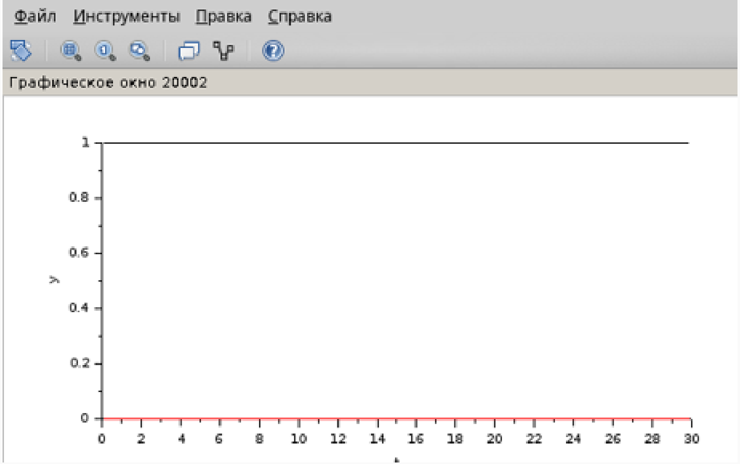
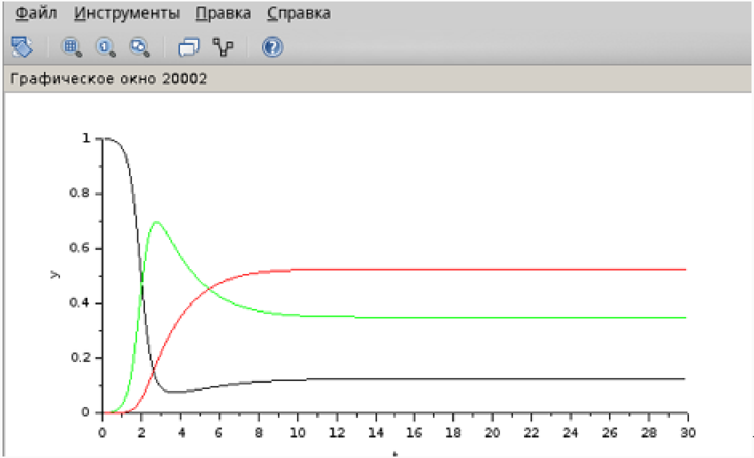


Figure 20: График модели SIR с учетом демографических процессов

3) $\beta = 4, \nu = 0.3, \mu = 0.2$



}

Исходя из анализа графиков, можно сделать вывод, что чем выше значение любого из параметров, тем быстрее система достигает стационарного состояния. При высоком коэффициенте заражения β система быстро проходит через пик развития эпидемии и

В процессе выполнения данной лабораторной работы был построен модель SIR в *xcos* и OpenModelica.

спасибо за внимание