Лабораторная работа 5

Модель эпидемии (SIR)

Туем Гислен

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

построить модель SIR в xcos и в OpenModelicaв xcos.

# 2 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); Сделать анализ полученных графиков в зависимости от выбранных значений параметров модели.

# 3 Выполнение лабораторной работы

Задача о распространении эпидемии описывается системой дифференциальных уравнений:

s’=−βs(t)i(t); i’=βs(t)i(t)−νi(t); r’=νi(t),

где β- скорость заражения, ν- скорость выздоровления.

# 4 Реализация модели в xcos

Зафиксируем начальные данные: β = 1, ν = 0, 3, s(0) = 0, 999, i(0) = 0, 001, r(0) = 0.(рис. 1).

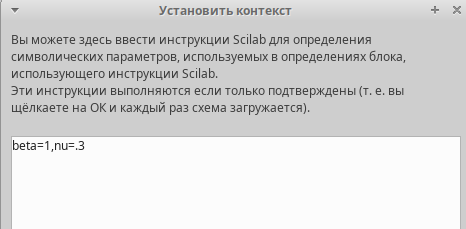


Рис. 1: задать переменные окружения в xcos

Для реализации модели потребуются следующие блоки xcos: – CLOCK\_c — запуск часов модельного времени; – CSCOPE — регистрирующее устройство для построения графика; – TEXT\_f — задаёт текст примечаний; – MUX — мультиплексер, позволяющий в данном случае вывести на графике сразу несколько кривых; – INTEGRAL\_m — блок интегрирования – GAINBLK\_f — в данном случае позволяет задать значения коэффициентов β и ν; – SUMMATION — блок суммирования; – PROD\_f — поэлементное произведение двух векторов на входе блока.(рис. 2).

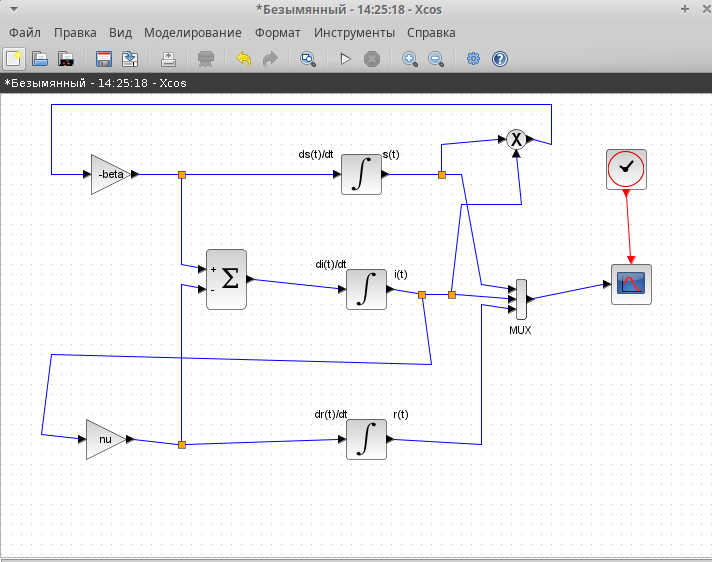


Рис. 2: Модель SIR в xcos

В параметрах верхнего и среднего блока интегрирования необходимо задать начальные значения s(0) = 0, 999 и i(0) = 0, 001 (рис. 3,4).

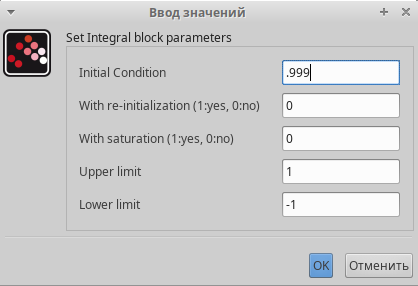


Рис. 3: Задать начальные значения в блоках интегрирования

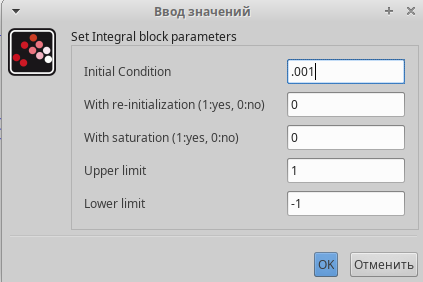


Рис. 4: Задать начальные значения в блоках интегрирования

В меню Моделирование, Установка необходимо задать конечное время интегри- рования(рис. 5).

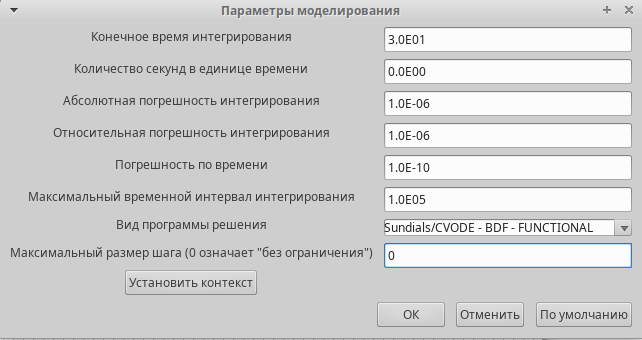


Рис. 5: Задать конечное время интегрирования в xcos

Результат моделирования представлен на (рис. 6)

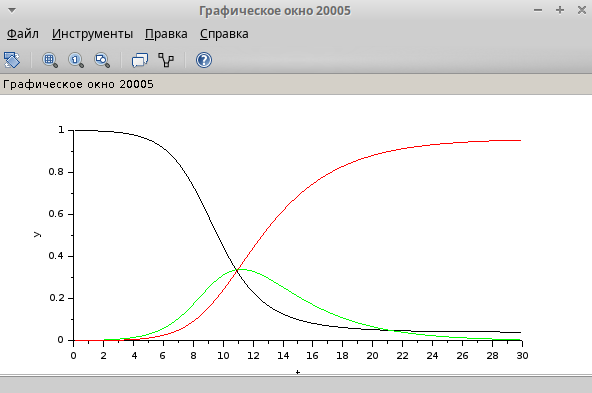


Рис. 6: Эпидемический порог модели SIR

# 5 Реализация модели с помощью блока Modelica в xcos

Готовая модель SIR представлена на (рис. 7). Для реализации модели 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.

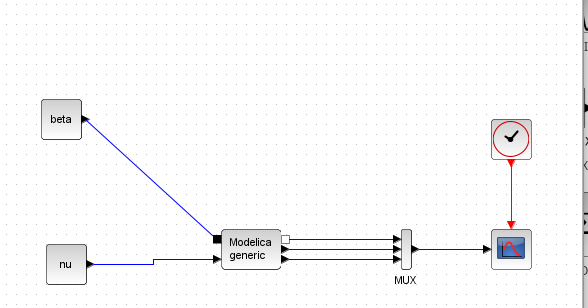


Рис. 7: Модель SIR в xcos с применением блока Modelica

Параметры блока Modelica представлены на (рис. 8,9). Переменные на входе (“beta”,“nu”) и выходе (“s”, “i”, “r”) блока заданы как внешние (“E”).

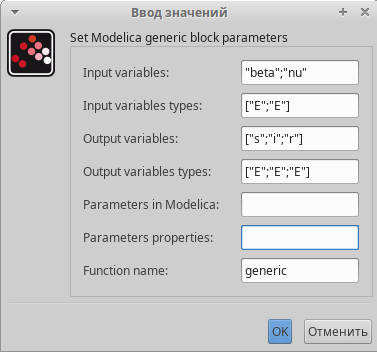


Рис. 8: Параметры блока Modelica для модели

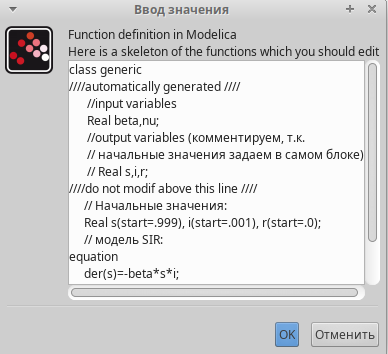


Рис. 9: Параметры блока Modelica для модели

## 5.1 Код на языке Modelica

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;

## 5.2 Результат моделирования (рис. 10)

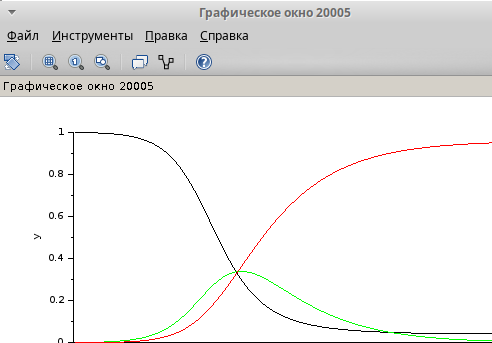


Рис. 10: Результат моделирования

# 6 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.

model lab  
  
 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;  
   
   
 Real s(start=S\_0);  
 Real i(start=I\_0);  
 Real r(start=R\_0);  
   
equation  
 der(s)=-beta\*s\*i;  
 der(i)=beta\*s\*i-nu\*i;  
 der(r)=nu\*i;  
  
  
end lab;

Результат модель SIR в OpenModelica(рис. 11).

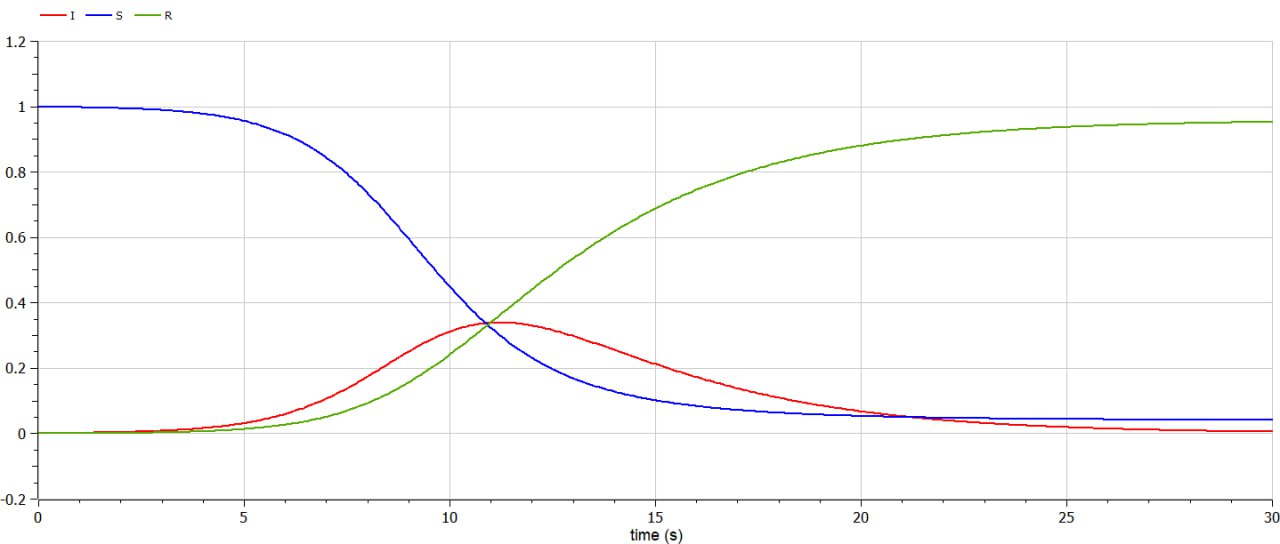


Рис. 11: Результат модель SIR в OpenModelica

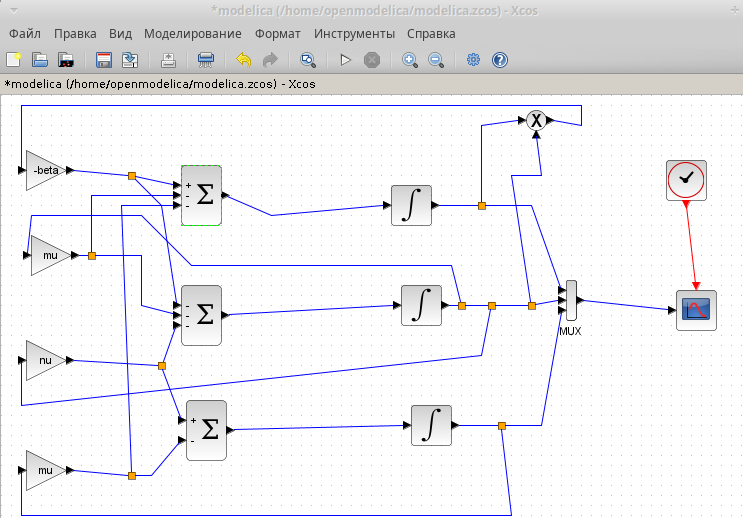
# 7 Задание для самостоятельного выполнения

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

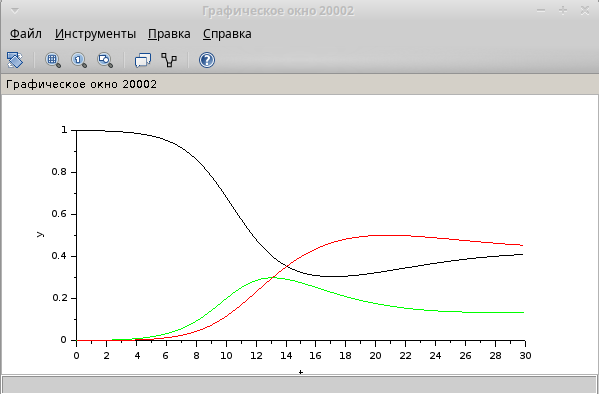
s’= −βs(t)i(t) + μ(N − s(t)); i’= βs(t)i(t) − νi(t) − μi(t); r’= νi(t) − μr(t),

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

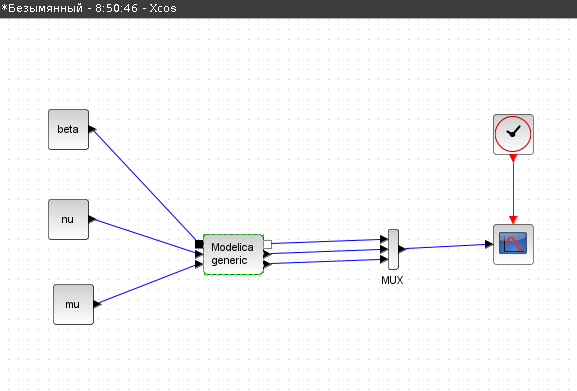
Реализуем эту модель в xcos. Тут нам понадобятся три блока суммирования и 4 блока констант (добавляется константа ν)



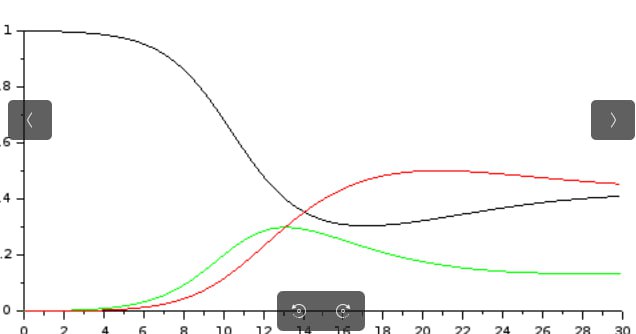
Результат модель (рис. **¿fig:021?**).



Теперь реализуем модель SIR с учетом демографических процессов в xcos с помощью блоков Modelica (рис. **¿fig:022?**).



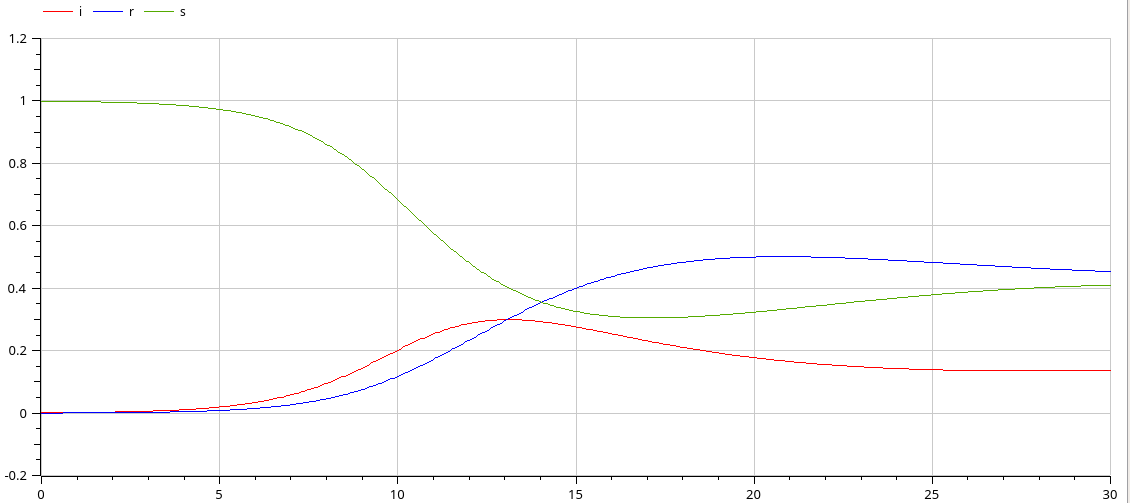
Результат модель (рис. **¿fig:037?**).



Реализуем модель SIR с учетом демографических процессов на OpenModelica.

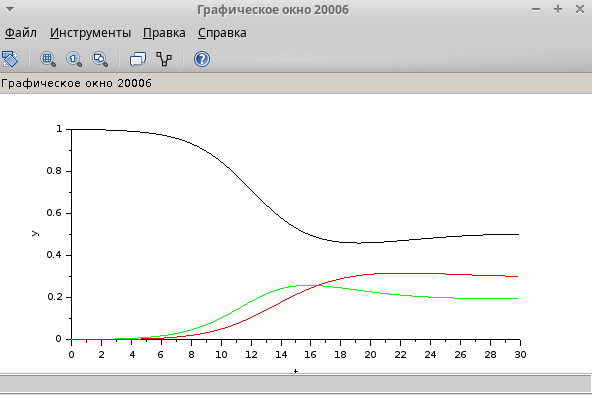
parameter Real I\_0 = 0.001;  
 parameter Real R\_0 = 0;  
 parameter Real S\_0 = 0.999;  
 parameter Real N = 1;  
 parameter Real beta = 1;  
 parameter Real nu = 0.3;  
 parameter Real mu = 0.5;  
   
 Real s(start=S\_0);  
 Real i(start=I\_0);  
 Real r(start=R\_0);  
   
equation  
 der(s)=-beta\*s\*i + mu\*i + mu\*r;  
 der(i)=beta\*s\*i-nu\*i - mu\*i;  
 der(r)=nu\*i - mu\*r;

Выполнив симуляцию, получим следующий график (рис. **¿fig:030?**).

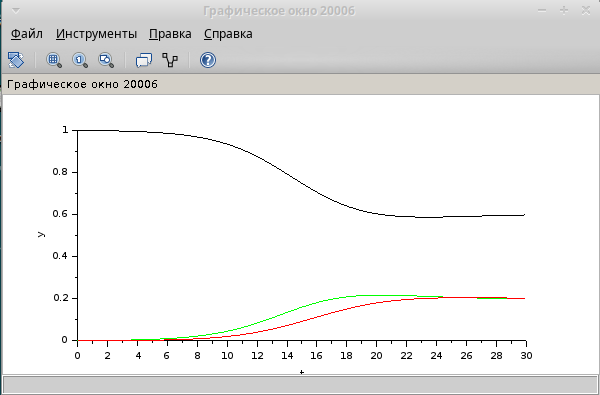


Теперь построим графики при разных значениях параметров.

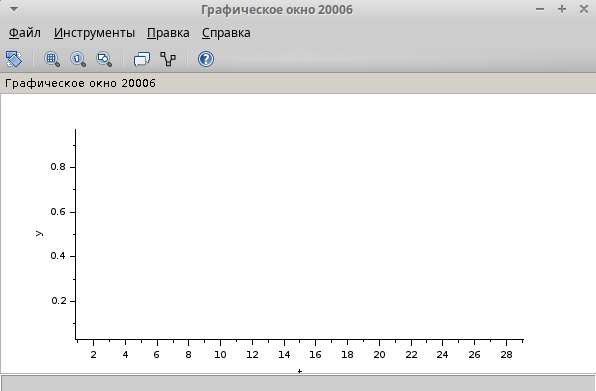
1. β=1, ν=0.3,μ=0.2(рис. **¿fig:038?**)



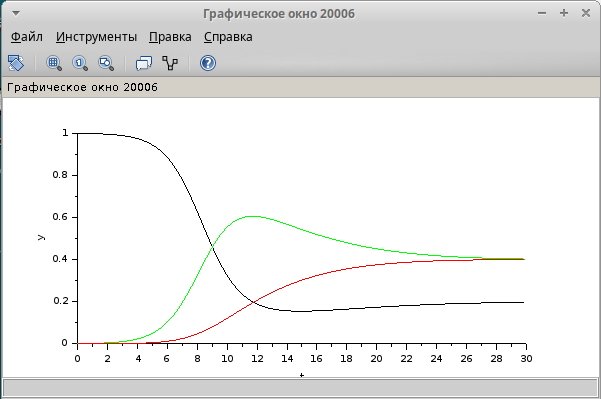
1. β=1, ν=0.3,μ=0.3(рис. **¿fig:039?**)



1. β=1, ν=0.3,μ=0.8(рис. **¿fig:040?**)



1. β=1, ν=0.1,μ=0.1(рис. **¿fig:042?**)



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 достигает стационарного состояния.

# 8 Выводы

В процессе выполнения данной лабораторной работы была построена модель SIR в xcos и OpenModelica.