Лабораторная работа 5

Имитационное моделирование

Оразгелдиев Язгелди

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Построить модель xcos и OpenModelica

2 Задание

– реализовать модель SIR с учётом процесса рождения / гибели особей в xcos (в том числе и с использованием блока Modelica), а также в OpenModelica; – 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

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

Рис. 3.1: СДУ

где beta — скорость заражения, nu — скорость выздоровления.

Зафиксируем начальные данные beta = 1, nu = 0.3, s(0)=0.999, i(0)=0.001, r(0)=0. В меню Моделирование, Задать переменные окружения зададим значения переменных beta и nu.

Для реализации модели (5.1) 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 beta и nu; – SUMMATION — блок суммирования; – PROD_f — поэлементное произведение двух векторов на входе блока.

Реализуем модель из примера

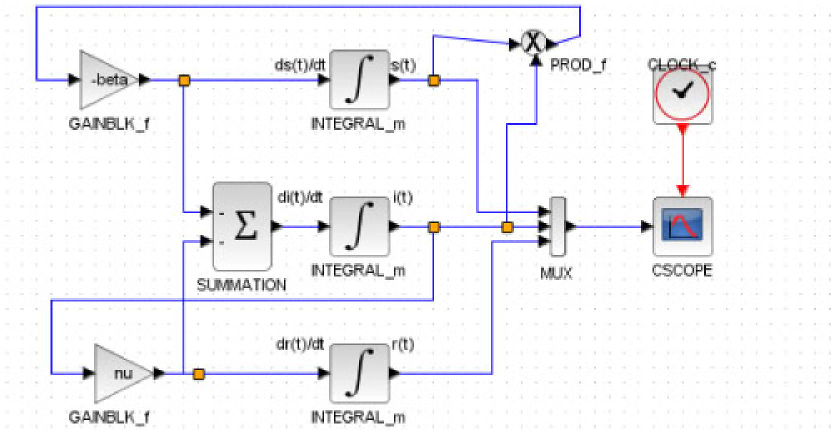

Рис. 3.2: Модель SIR

В параметрах верхнего и среднего блока интегрирования необходимо задать начальные значения s(0)=0.999, i(0)=0.001

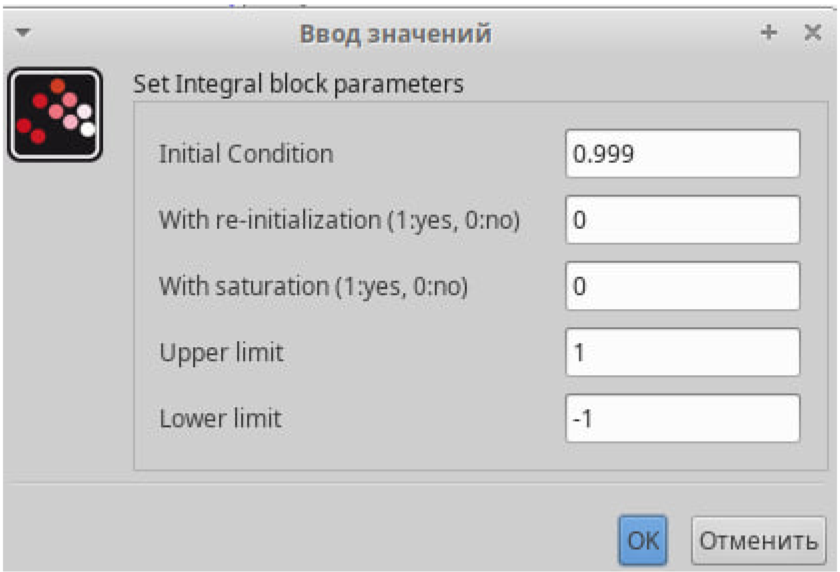

Рис. 3.3: Начальные значения в блоках интегрирования

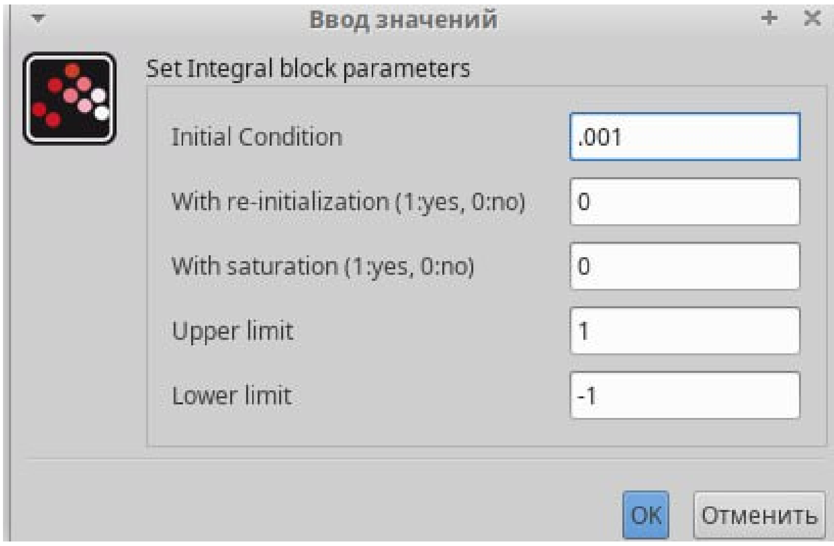

Рис. 3.4: Начальные значения в блоках интегрирования

В меню Моделирование, Установка зададим конечное время интегрирования, равным времени моделирования

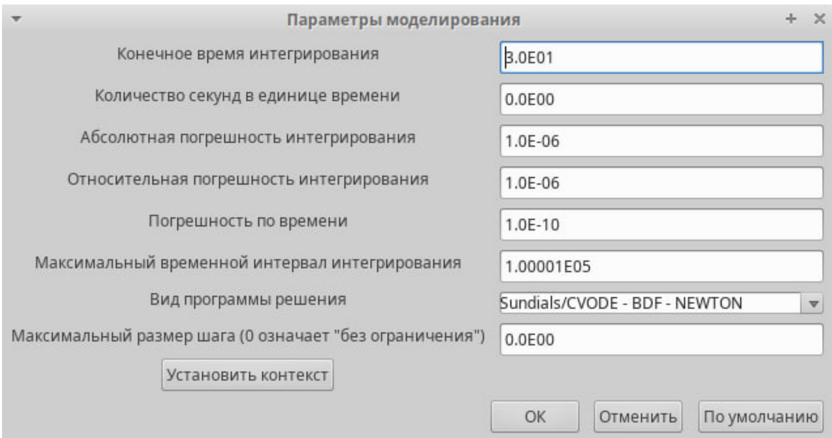

Рис. 3.5: Параметры моделирования

В итоге при запуске получится такой график

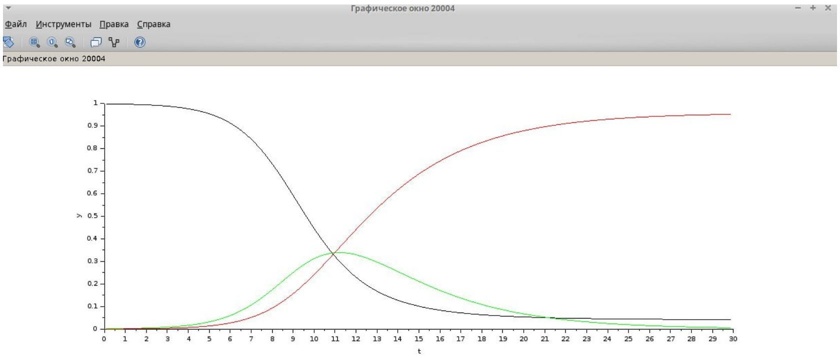

Рис. 3.6: Эпидемический порог модели SIR при $\beta = 1$, $\nu = 0.3$

Далее реализуем модель SIR с помощью блока Modelica в xcos. Для реализации модели с помощью языка Modelica помимо блоков CLOCK_c, CSCOPE, TEXT_f и MUX используем блоки CONST_m — задаёт константу; MBLOCK (Modelica generic) — блок реализации кода на языке Modelica. Задаём значения переменных beta и nu

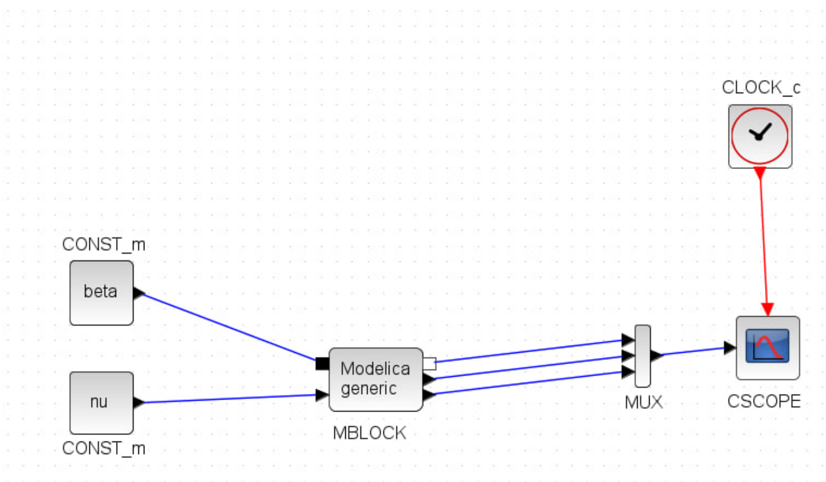

Рис. 3.7: Модель SIR в xcos с применением блока Modelica

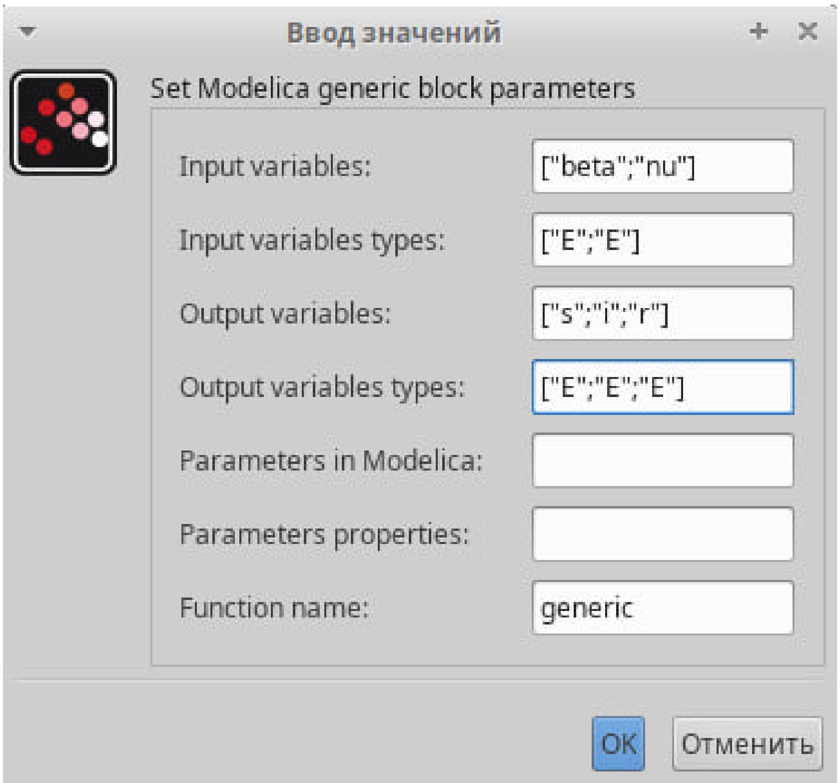

Рис. 3.8: Параметры блока Modelica для модели

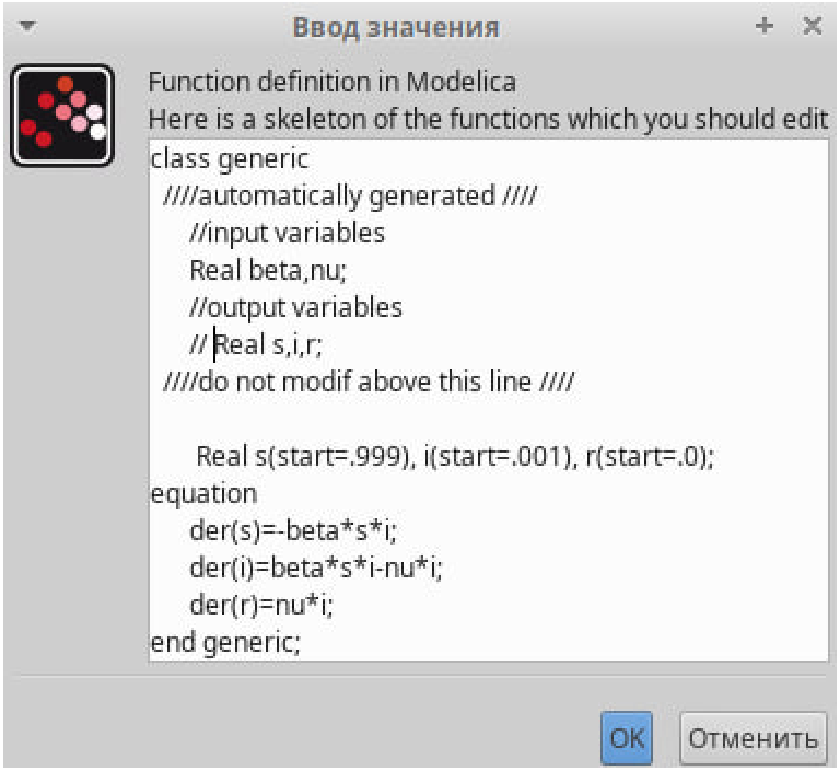

Рис. 3.9: Параметры блока Modelica для модели

В результате получаем такой график, построенный с помощью блока Modelica в xcos. Он идентичен прошлому графику

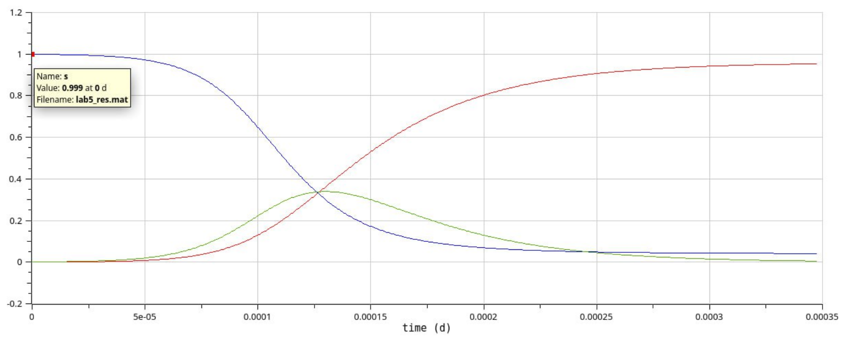

Рис. 3.10: График, построенный с помощью блока Modelica в xcos

Далее мы выполняем упражнение: реализуем модель SIR в OpenModelica.

Система схожа на обычную Modelica. Задаем параметры, начальные значения и систему дифф. уравнений.

```
model lab5

  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 lab5;
```

Рис. 3.11: Заданием параметров и начальных значений

Теперь реализуем симуляцию с конечным временем 30с.

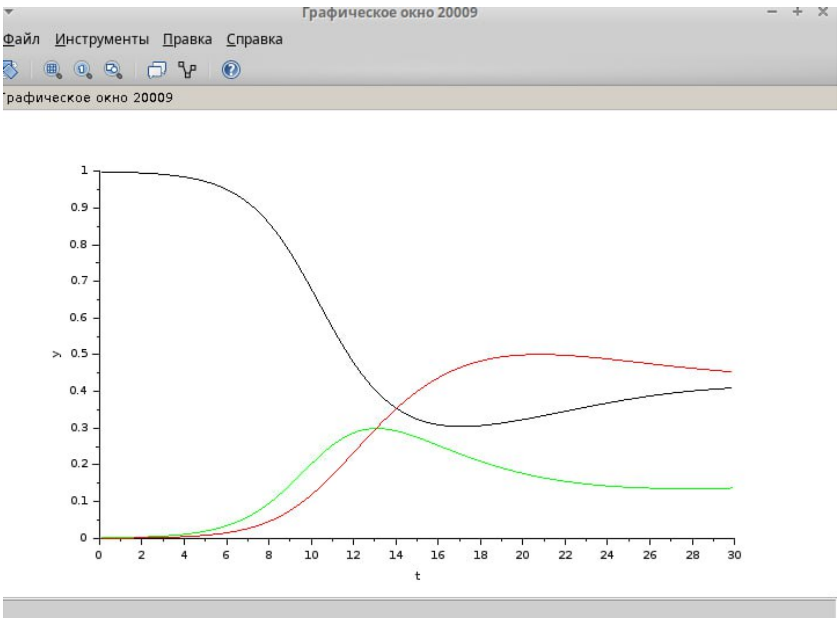

Рис. 3.12: модель SIR в OpenModelica

После выполнения упражнения выполним задание для самостоятельного выполнения. Предположим, что 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:

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

Рис. 3.13: СДУ из задания

Для начала реализуем модель в xcos. Тут нам понадобятся 3 блока суммирования и 4 блока констант. В итоге получаем такую схему:

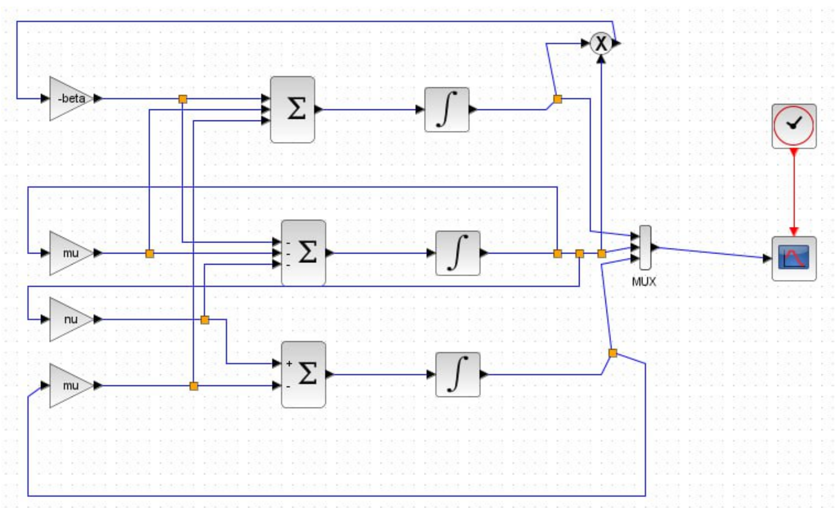

Рис. 3.14: Схема

Запускаем и получаем такой график

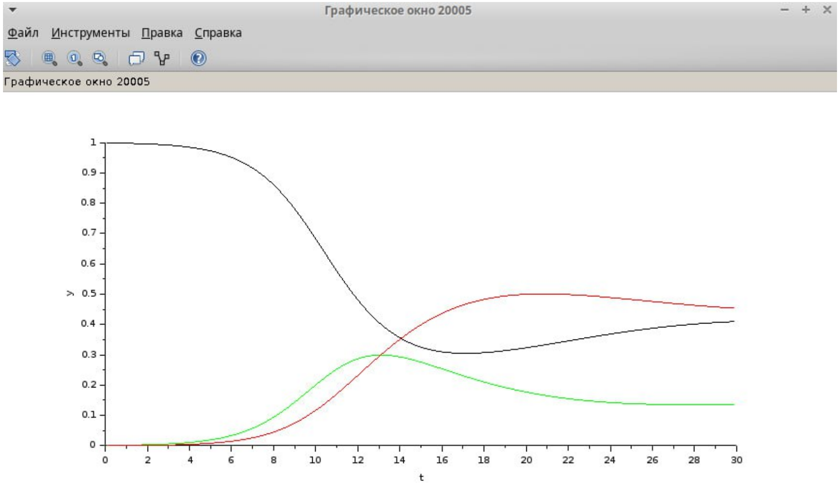

Рис. 3.15: График модели SIR в xcos

Теперь реализуем эту же модель с помощью блока Modelica

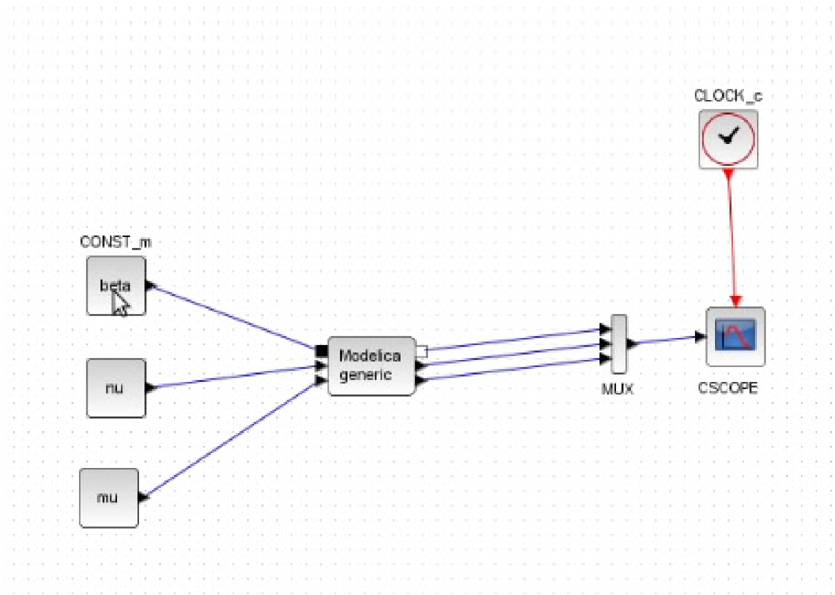

Рис. 3.16: Модель SIR в xcos с применением блока Modelica

Запускаем и получаем такой график как и в прошлом случаем

И последним этапом мы реализуем модель SIR в OpenModelica.

```
model l5t2

  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 = 0.1;

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

equation
  der(s) = -beta*s*i + mu*i +mu*i;
  der(i) = beta*s*i - nu*i;
  der(r) = nu*i - mu*r;

end l5t2;
```

Рис. 3.17: Синтаксис для симуляции

Запускаем симуляцию и получаем следующий график

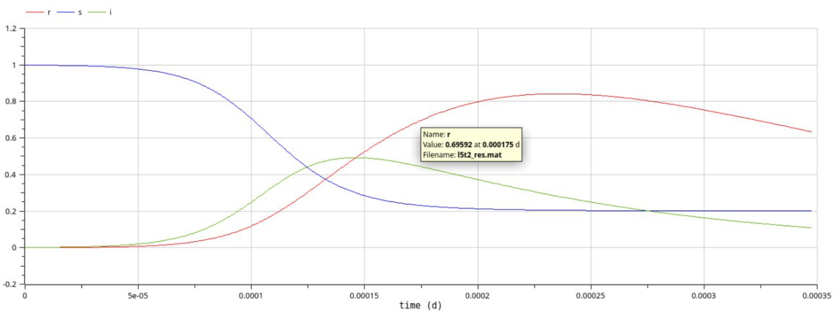

Рис. 3.18: Модель SIR в OpenModelica

Теперь мы пробуем задавать различные значения и выводить графики

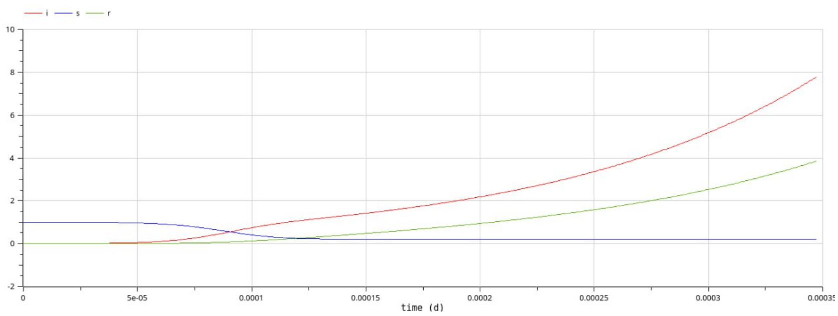

Рис. 3.19: Модель SIR в OpenModelica с разными параметрами

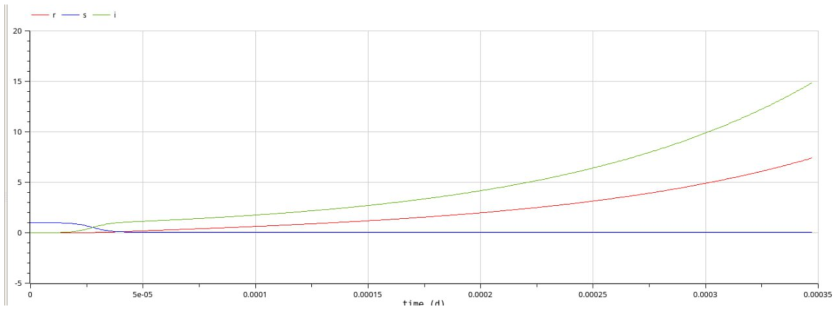

Рис. 3.20: Модель SIR в OpenModelica с разными параметрами

Стоит сделать вывод, что при повышении значения параметров система достигает стационарного состояния. При высоком коэффициенте заражения beta система быстро проходит через пик развития эпидемии и достигает стационарного состояния.

4 Выводы

В процессе выполнения лабораторной работы построил модель SIR в xcos и OpenModelica.