Лабораторная работа № 5

Имитационное моделирование

Королёв Иван

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Построение модели эпидемии (SIR) в xcos, с помощью блока Modelica и в OpenModelica.

2 Задание

1. Необходимо реализовать модель эпидемии в xcos
2. Необходимо реализовать модель эпидемии с помощью блока Modelica в xcos
3. Выполнить упражнение построения модели эпидемии в OpenModelica
4. Задание для самостоятельного выполнения. Требуется:

- 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 Теоретическое введение

Модель SIR предложена в 1927 г. (W. O. Kermack, A. G. McKendrick). С описанием модели можно ознакомиться, например в [1]. Предполагается, что особи популяции размера N могут находиться в трёх различных состояниях: * S (susceptible, уязвимые) — здоровые особи, которые находятся в группе риска и могут подхватить инфекцию; * I (infective, заражённые, распространяющие заболевание) — заразившиеся переносчики болезни; * R (recovered/removed, вылечившиеся) — те, кто выздоровел и перестал распространять болезнь (в эту категорию относят, например, приобретших иммунитет или умерших). Внутри каждой из выделенных групп особи считаются неразличимыми по свойствам.

4 Выполнение лабораторной работы

4.1 Реализация модели эпидемии в xcos

Зафиксируем начальные данные: **$\beta = 1, \nu = 0,3, s(0) = 0,999, i(0) = 0,001, r(0) = 0$**.

В меню моделирования устанавливаем переменные окружения (рис. 4.1)

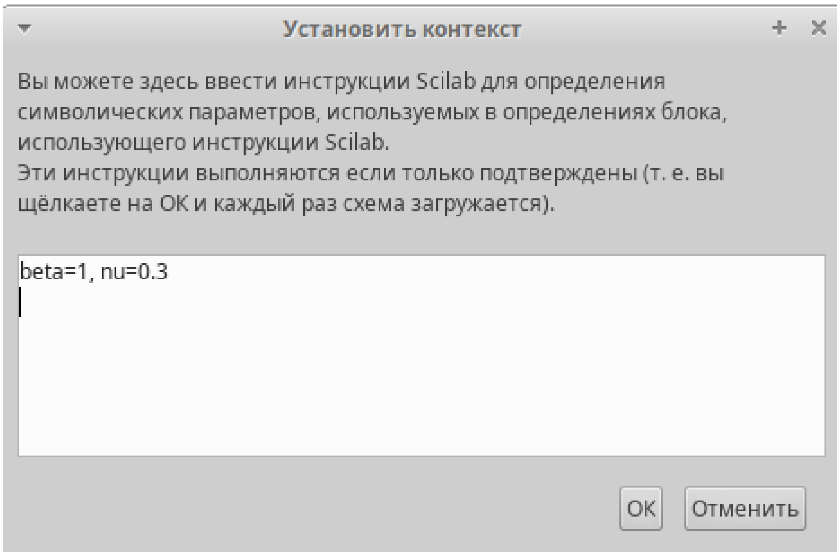


Рис. 4.1: beta, nu

Для реализации модели потребуется: * 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.

Добавляем эти блоки из палитры инструментов и строим с их помощью данную систему 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.

Реализованная модель эпидемии. Выходы трёх блоков интегрирования соединяем с мультиплексором.(рис. 4.2)

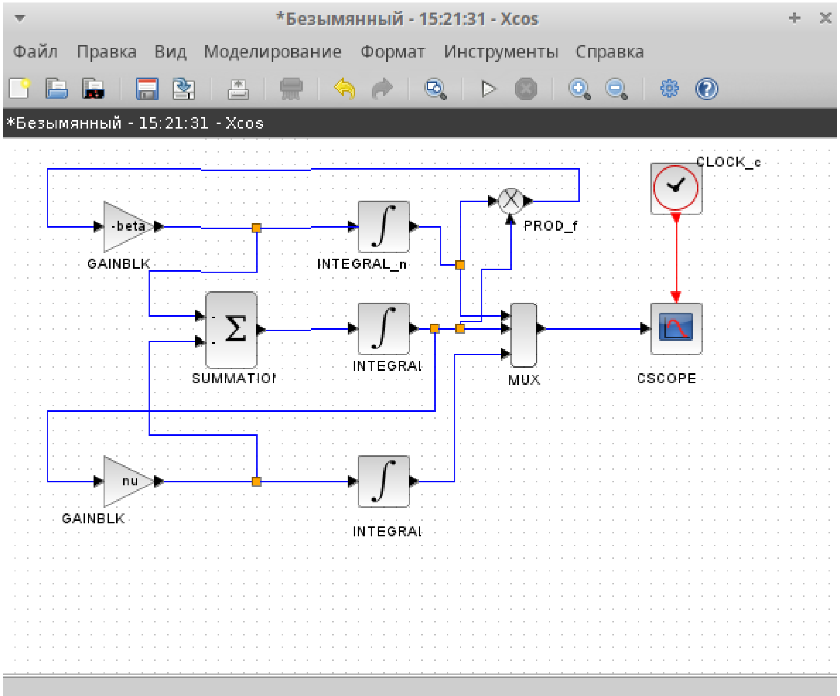


Рис. 4.2: Реализованная модель эпидемии

В параметрах верхнего блока интегрирования задаем значения s(0) = 0, 999, который отвечает за здоровых особей. (рис. 4.3)

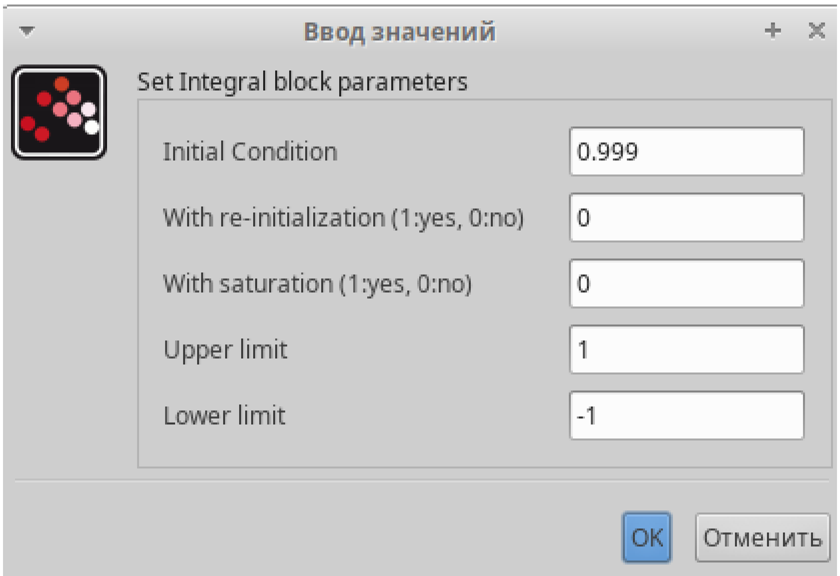


Рис. 4.3: Начальные значения для верхнего блока интегрирования

В параметрах среднего блока интегрирования задаем значения i(0) = 0, 001, который отвечает за переносчиков болезни. (рис. 4.4)

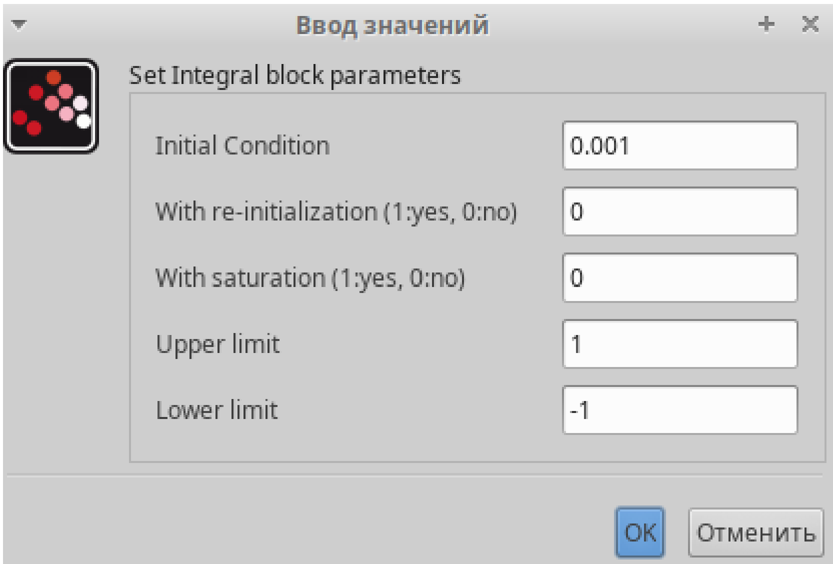


Рис. 4.4: Начальные значения для среднего блока интегрирования

В нижнем блоке интегрирования начальные значения по умолчанию заданы нулю, как в нашем условии. Данная часть отвечает за тех, кто имеет иммунитет.

Далее, устанавливаем конечное время интегрирования. Оно равно 30 (рис. 4.5)

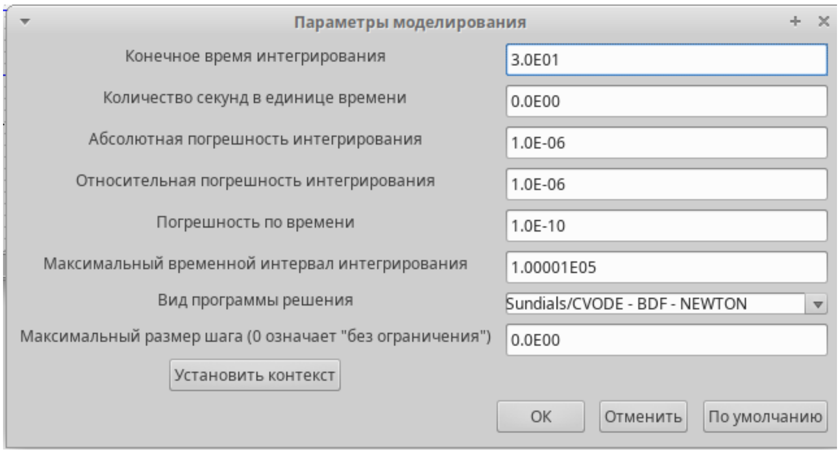


Рис. 4.5: Конечное время интегрирования

Результат моделирования представлен на (рис. 4.6), где черной линией обозначен график s(t) (динамика численности уязвимых к болезни особей), красная линия r(t) — динамику численности выздоровевших особей, наконец, зеленая линия i(t) — динамику численности заражённых особей. Пересечение трёх линий определяет порог эпидемии.

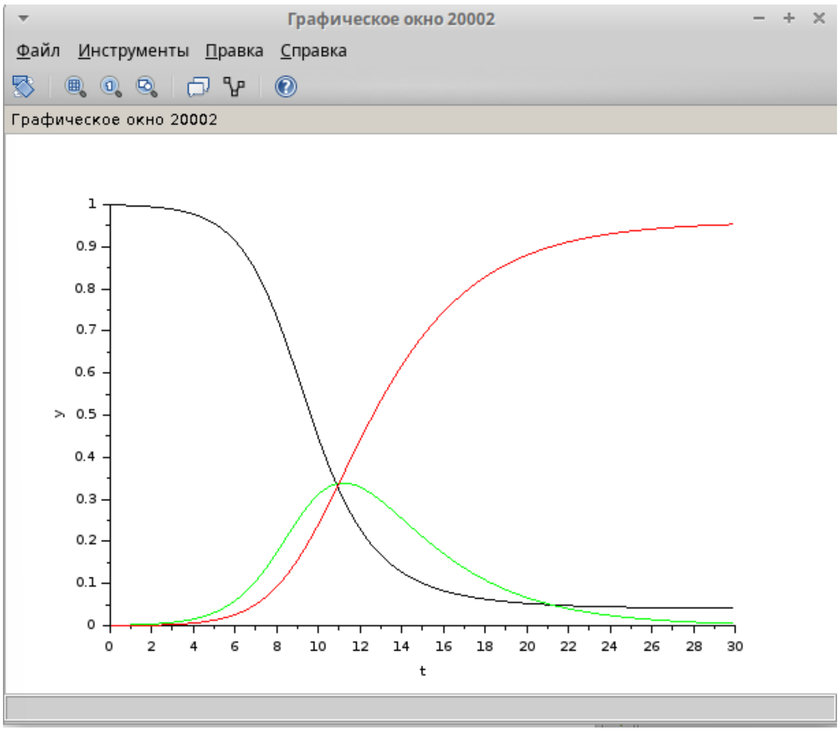


Рис. 4.6: Модель эпидемии при beta=1, nu=0.3

4.2 Реализация модели с помощью блока Modelica в xcos

В данном задании необходимо было реализовать такую же модель эпидемии при beta=1, nu=0.3, только с помощью блока Modelica в xcos. Для начала добавляем новый блок констант и блок реализации кода на Modelica. Таким образом выглядит наша модель (рис. 4.7)

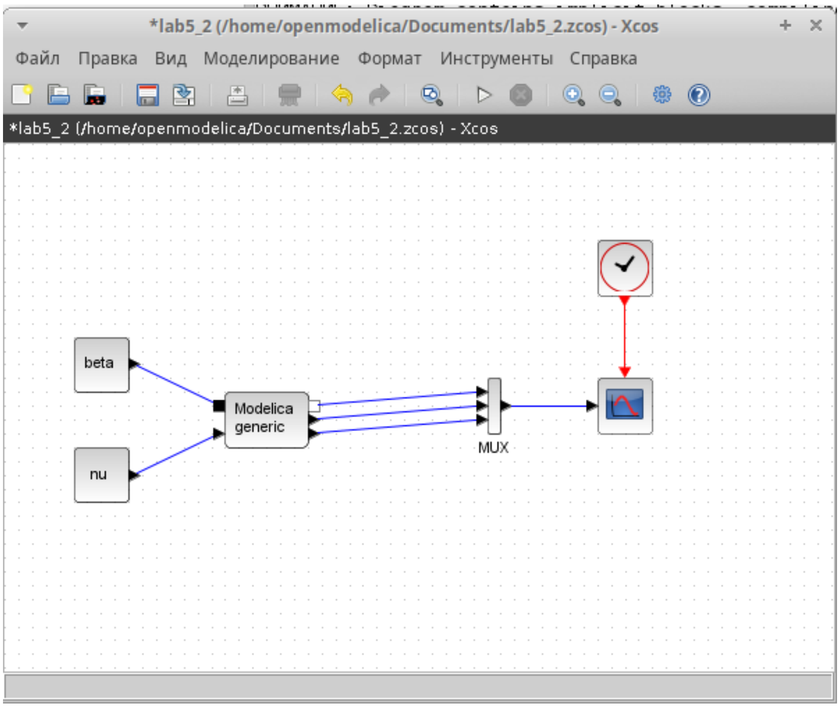


Рис. 4.7: Модель эпидемии

Указываем параметры для блока реализации. Переменные на входе (“beta”, “nu”) и выходе (“s”, “i”, “r”) блока заданы как внешние (“E”). (рис. 4.8)

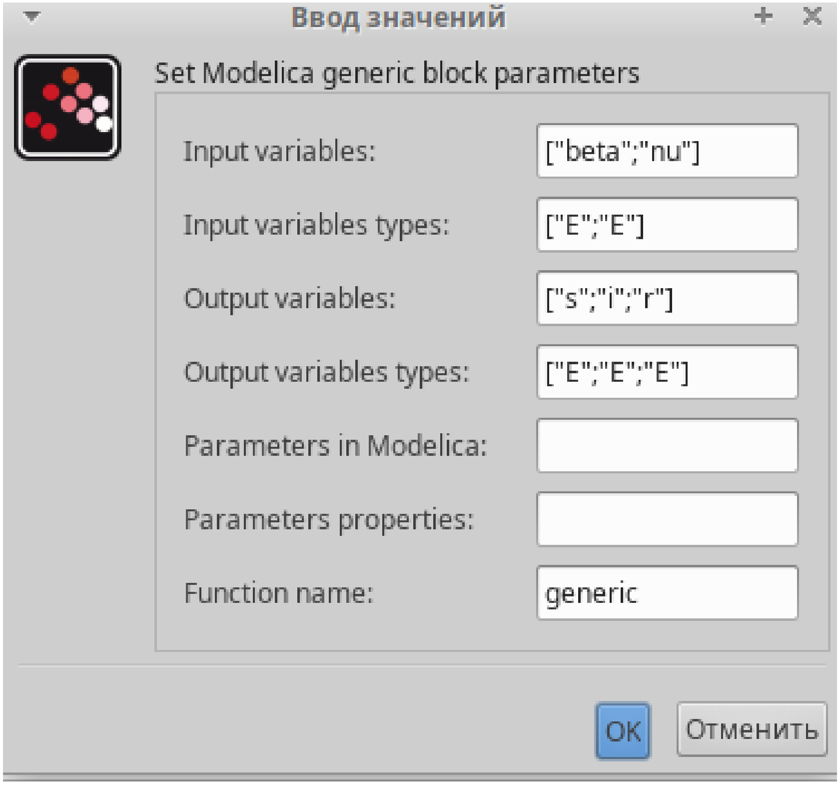


Рис. 4.8: Параметры блока реализации

Код на языке Modelica. Задаем переменные beta, nu. Указываем начальные значения для s, i, r и пишем систему уравнения. (рис. 4.9)

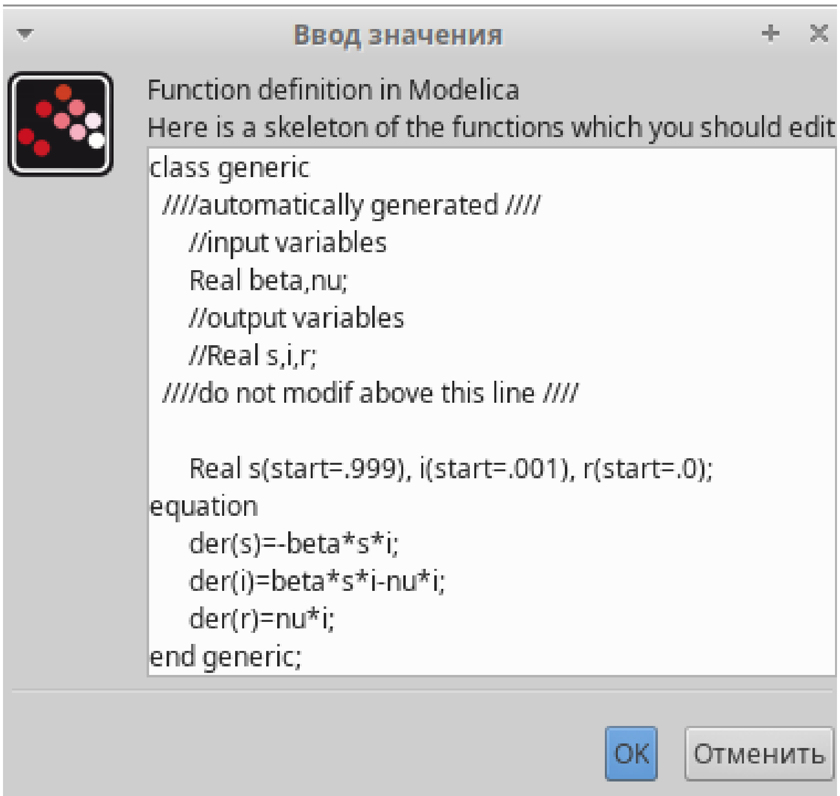


Рис. 4.9: Параметры блока реализации

Результат работы модели. Он идентичен с реализацией в xcos. (рис. 4.10)

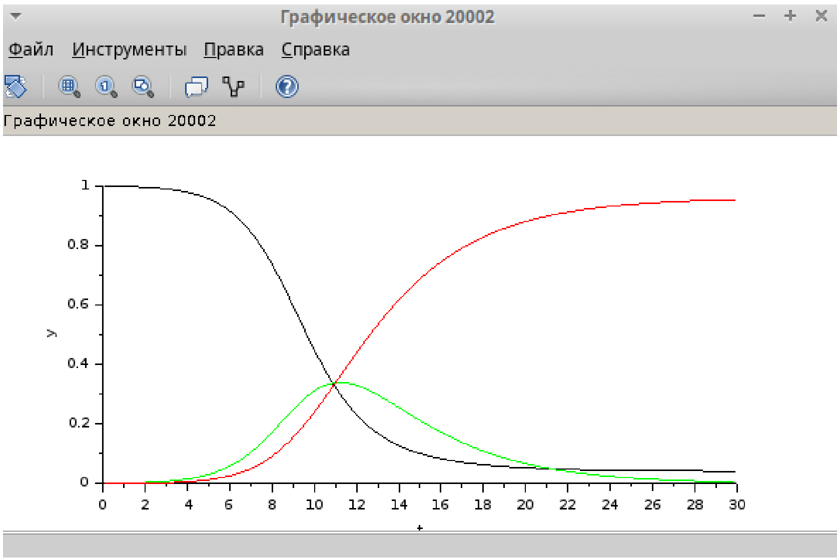


Рис. 4.10: Модель эпидемии Modelica

4.3 Выполнение упражнени построения модели эпидемии в OpenModelica

Код реализации модели эпидемии в OpenModelica. Задаем все начальные параметры с помощью parameter Real, как было в реализациях xcos. Записываем систему уравнения, реализация очень сильно схожа с реализацией с помощью блока Modelica в xcos (рис. 4.11)

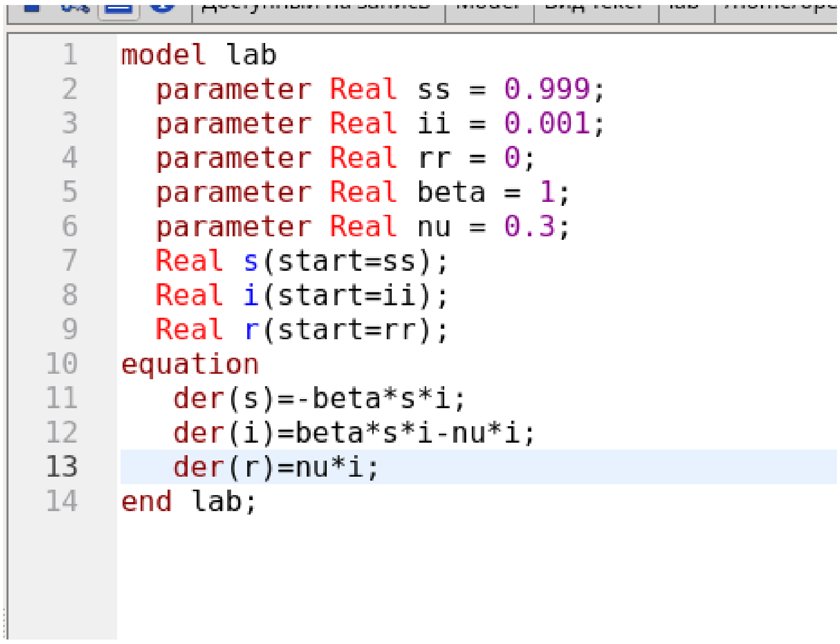


Рис. 4.11: Реализация модели эпидемии в OpenModelica

Результат модели. Результат идентичен с построением с помощью других способов, значит все выполнено правильно. (рис. 4.12)

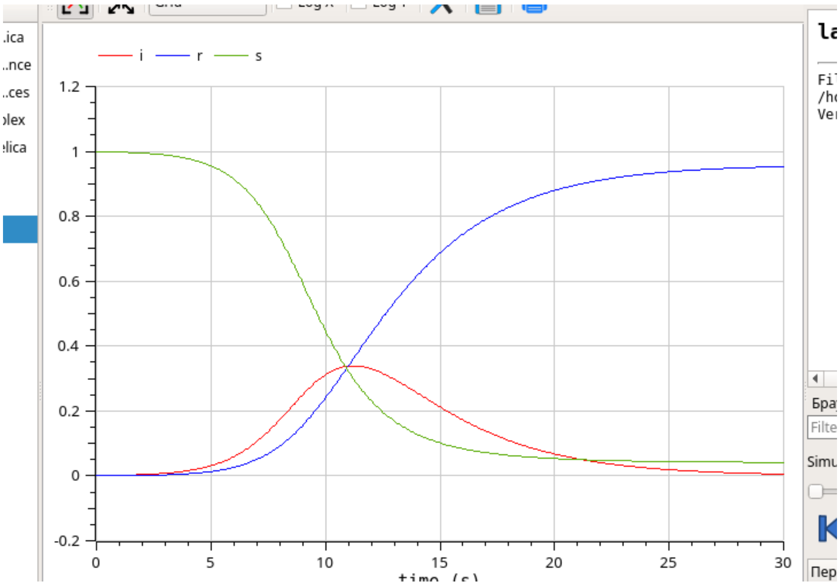


Рис. 4.12: Модель эпидемии в OpenModelica

4.4 Задание для самостоятельного выполнения. Реализация с помощью xcos

Необходимо реализовать такую же модель эпидемии, только с учётом процесса рождения / гибели особей в xcos (в том числе и с использованием блока Modelica), а также в OpenModelica.

Так выглядит система уравнения:

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

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).

В меню моделирования устанавливаем переменные окружения. (рис. 4.13) Реализация с помощью xcos. (рис. 4.14)

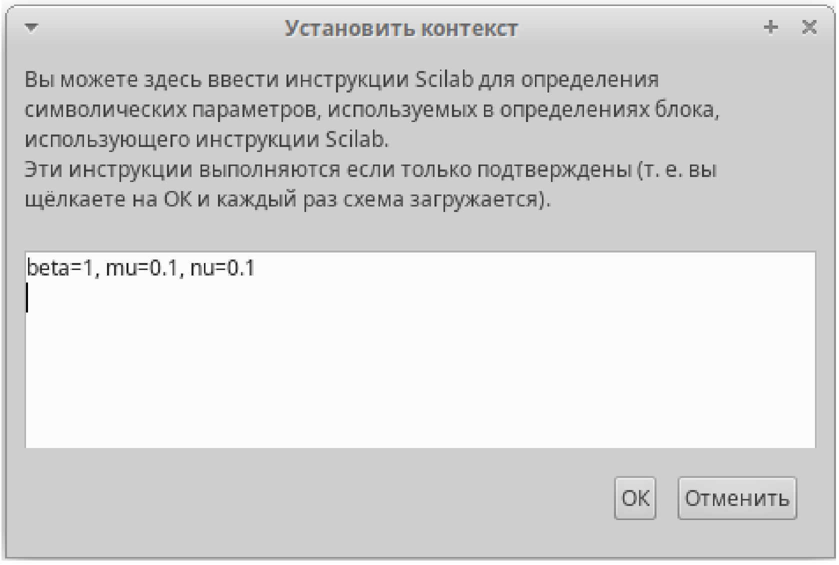


Рис. 4.13: Переменные окружения

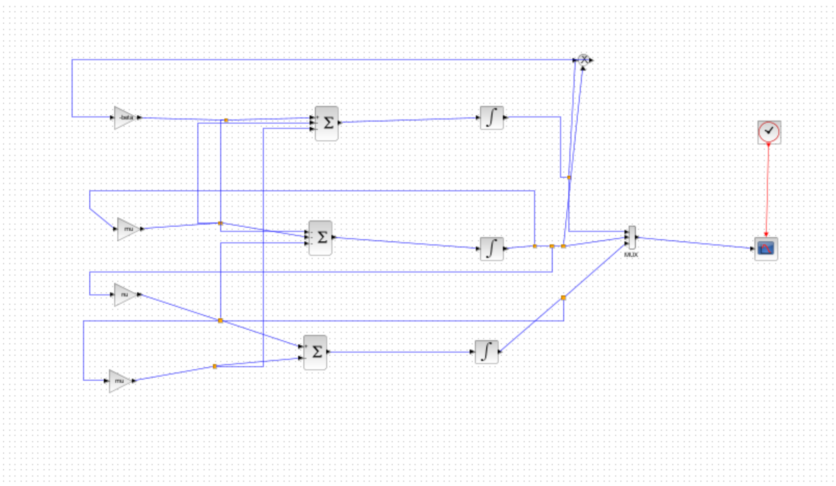


Рис. 4.14: Реализация модели эпидемии с учетом процесса рождения / гибели особей с помощью xcos

В параметрах блоков интегрирования нет изменений, указываем все начальные значения из предыдущих этапов выполнения.

Результат моделирования представлен на (рис. 4.15), где черной линией обозначен график s(t) (динамика численности уязвимых к болезни особей), красная линия r(t) — динамику численности выздоровевших особей, наконец, зеленая линия i(t) — динамику численности заражённых особей. Пересечение трёх линий определяет порог эпидемии.

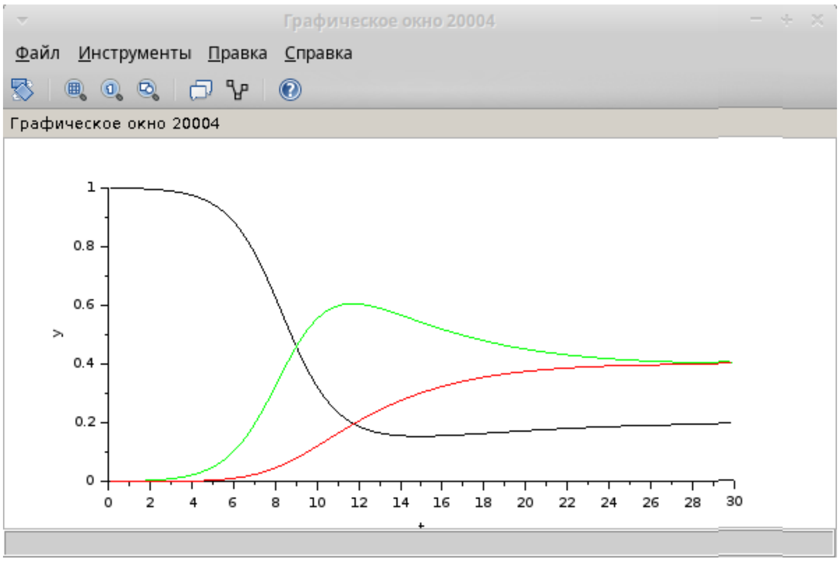


Рис. 4.15: Модель эпидемии при beta=1, nu=0.1, mu=0.1

4.5 Задание для самостоятельного выполнения. Реализация с помощью блока Modelica в xcos

Реализация с помощью блока Modelica в xcos. (рис. 4.16)

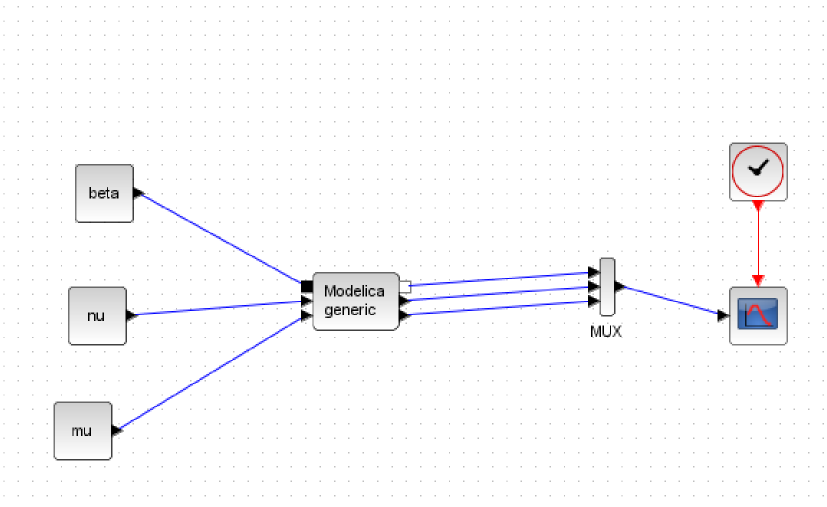


Рис. 4.16: Реализация модели эпидемии с учетом процесса рождения / гибели особей с помощью блока Modelica в xcos

Указываем параметры для блока реализации. Переменные на входе (“beta”, “nu”, “mu”) и выходе (“s”, “i”, “r”) блока заданы как внешние (“E”). (рис. 4.17)

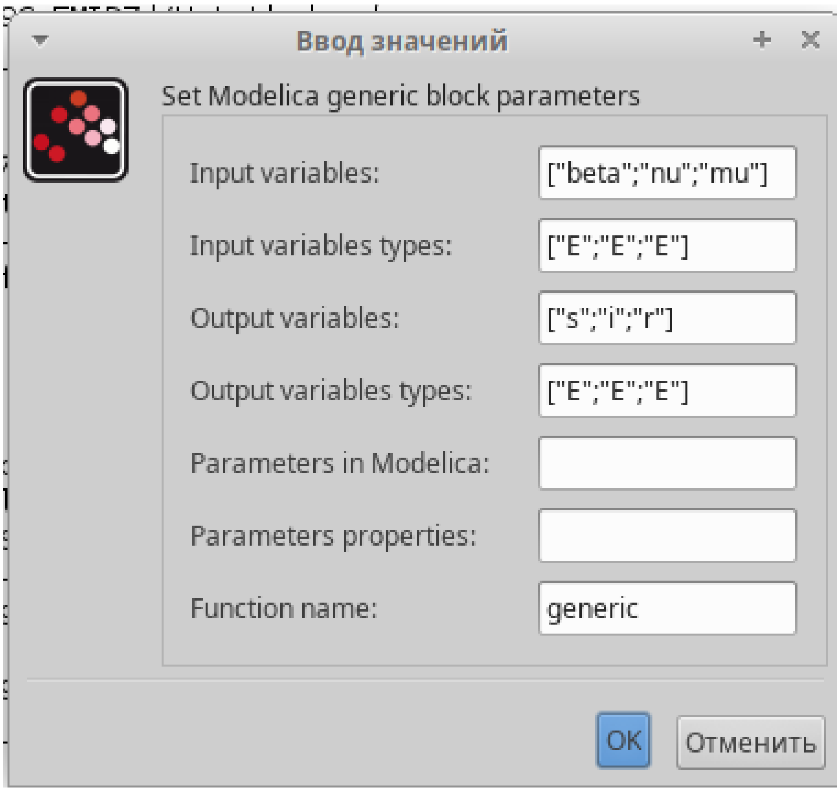


Рис. 4.17: Параметры блока реализации

Код на языке Modelica. Задаем переменные beta, nu, mu. Указываем начальные значения для s, i, r и пишем систему уравнения. (рис. 4.18)

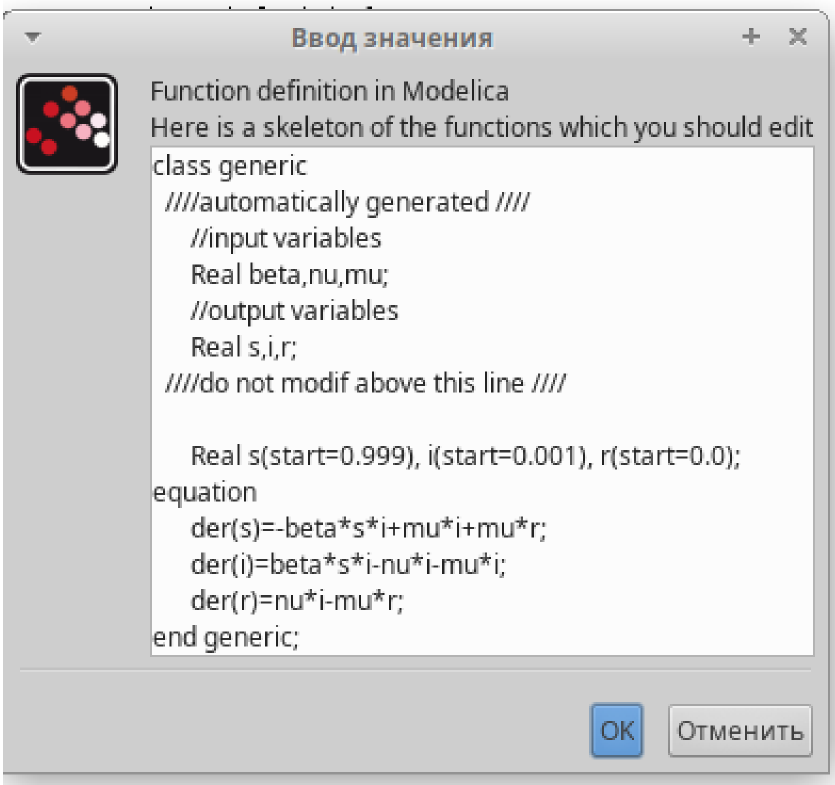


Рис. 4.18: Параметры блока реализации

Результат моделирования представлен на (рис. 4.19), где черной линией обозначен график s(t) (динамика численности уязвимых к болезни особей), красная линия r(t) — динамику численности выздоровевших особей, наконец, зеленая линия i(t) — динамику численности заражённых особей. Пересечение трёх линий определяет порог эпидемии.

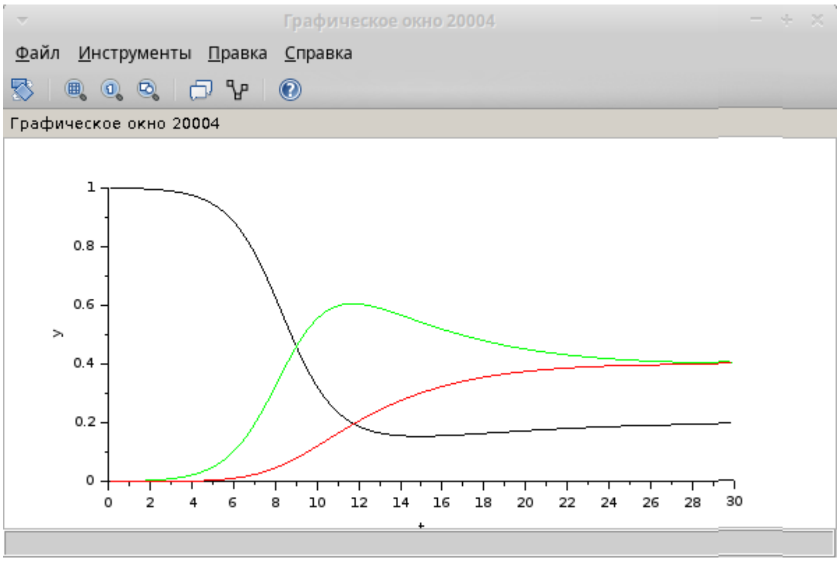


Рис. 4.19: Модель эпидемии при beta=1, nu=0.1, mu=0.1

4.6 Задание для самостоятельного выполнения. Реализация в OpenModelica

Код реализации модели эпидемии в OpenModelica. Задаем все начальные параметры с помощью parameter Real, как было в реализациях xcos. Записываем систему уравнения, реализация очень сильно схожа с реализацией с помощью блока Modelica в xcos (рис. 4.20)

```
model lab
  parameter Real ss = 0.999;
  parameter Real ii = 0.001;
  parameter Real rr = 0;
  parameter Real beta = 1;
  parameter Real nu = 0.1;
  parameter Real mu = 0.1;
  Real s(start=ss);
  Real i(start=ii);
  Real r(start=rr);
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 lab;
```

Рис. 4.20: Реализация модели с учетом процесса рождения / гибели особей эпидемии в OpenModelica

Результат модели. Результат идентичен с построением с помощью других способов, значит все выполнено правильно. (рис. 4.21)

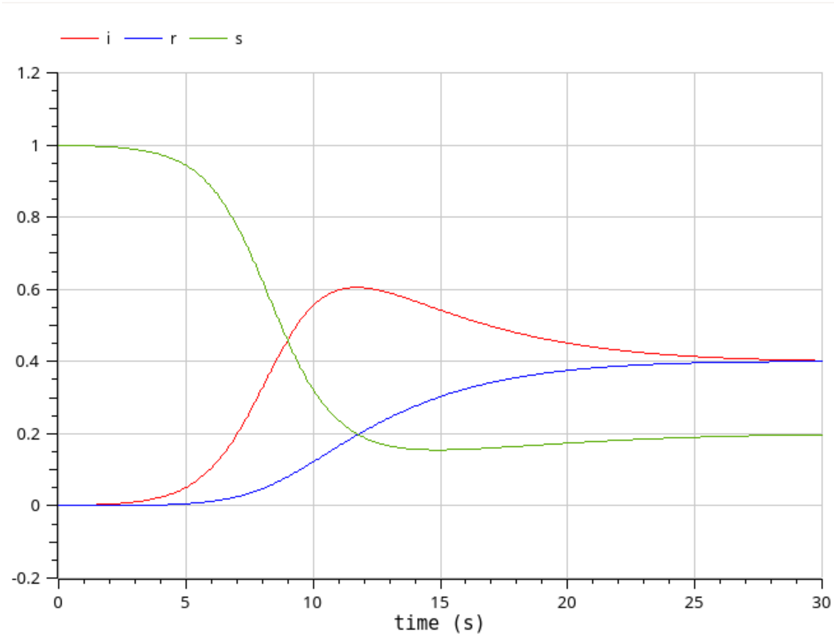
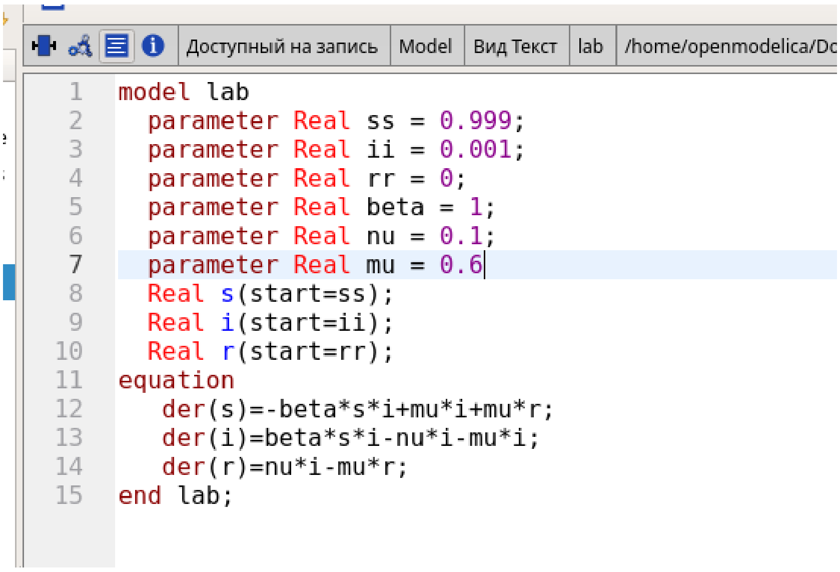


Рис. 4.21: Модель эпидемии с учетом процесса рождения / гибели особей в OpenModelica

4.7 Результаты на различных параметрах.

При mu=0.6, nu=0.1, beta=1 (рис. 4.22), (рис. 4.23)




```
model lab
  parameter Real ss = 0.999;
  parameter Real ii = 0.001;
  parameter Real rr = 0;
  parameter Real beta = 1;
  parameter Real nu = 0.1;
  parameter Real mu = 0.6
  Real s(start=ss);
  Real i(start=ii);
  Real r(start=rr);
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 lab;
```

Рис. 4.22: Результаты на различных параметрах.

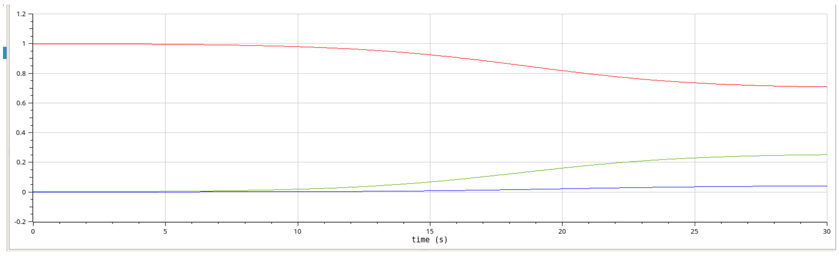


Рис. 4.23: Результаты на различных параметрах.

При mu=0.6, nu=0.6, beta=1 (рис. 4.24), (рис. 4.25)

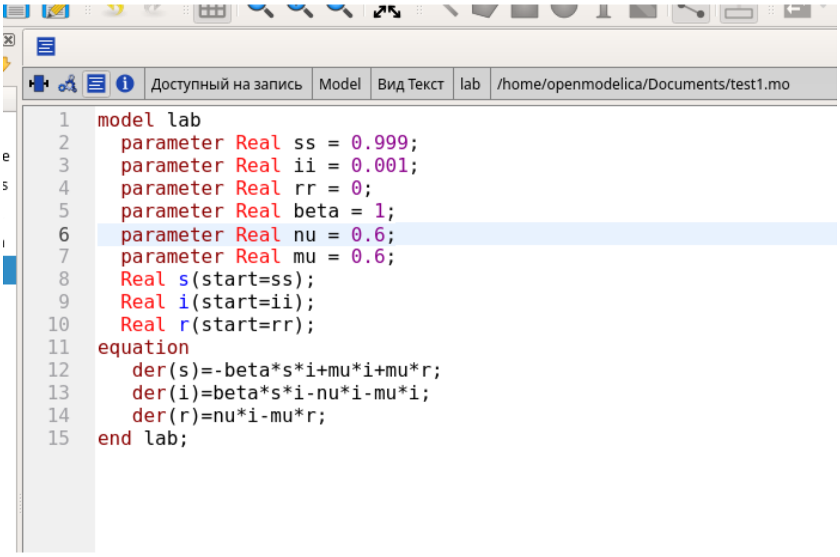


Рис. 4.24: Результаты на различных параметрах.

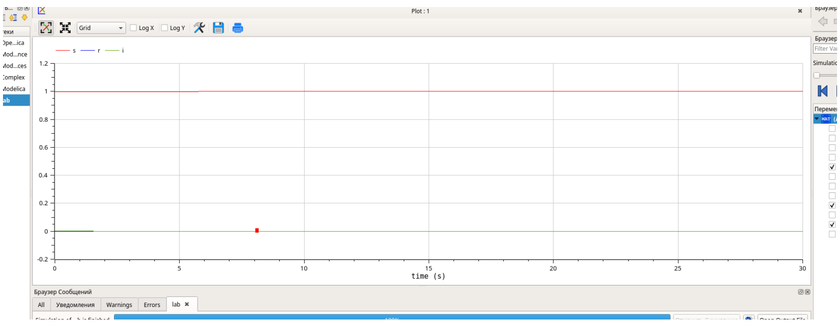


Рис. 4.25: Результаты на различных параметрах.

Исходя из анализа графиков, можно сделать вывод, что чем выше значение любого из параметров, тем быстрее система достигает стационарного состояния.

При высоком коэффициенте заражения □ система быстро проходит через пик развития эпидемии и достигает стационарного состояния.

5 Выводы

Построил модели эпидемии (SIR) в xcos, с помощью блока Modelica и в OpenModelica.

Список литературы