

Лабораторная работа №5

Модель эпидемии (SIR)

Астраханцева А. А.

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Реализация модели в xcos	7
5	Реализация модели с помощью блока Modelica в xcos	14
6	Реализация модели SIR в OpenModelica	18
7	Задание для самостоятельного выполнения в xcos	21
8	Задание для самостоятельного выполнения с помощью блока Modelica в xcos	25
9	Задание для самостоятельного выполнения в OpenModelica	27
10	Выводы	28

Список иллюстраций

4.1	Значения переменных β и ν	7
4.2	Блок со значением коэффициента β	8
4.3	Блок со значением коэффициента ν	8
4.4	Настройки для верхнего блока интегрирования	9
4.5	Настройки для среднего блока интегрирования	9
4.6	Настройки для нижнего блока интегрирования	10
4.7	Настройки времени моделирования	10
4.8	Настройки для мультиплексора	11
4.9	Настройки для регистрирующего устройства	11
4.10	Настройки для блока суммирования	12
4.11	Итоговый вид схемы	12
4.12	График распространения эпидемии	13
5.1	Параметры блока Modelica	15
5.2	Код для блока Modelica	16
5.3	Итоговый вид схемы	17
5.4	График распространения эпидемии	17
6.1	Создание нового класса в OpenModelica	18
6.2	Настройки моделирования в OpenModelica	19
6.3	Реализация класса в OpenModelica	19
6.4	График распространения эпидемии	20
7.1	Значения переменных β , ν и μ	22
7.2	Итоговый вид схемы	22
7.3	$\mu = 0.5$	23
7.4	Время моделирования - 60 секунд, $\mu = 0.5$	23
7.5	$\mu = 0.1$	23
7.6	$\mu = 0.9$	24
8.1	Параметры блока Modelica	25
8.2	Код для блока Modelica	26
8.3	$\mu = 0.5, \beta = 3$	26
8.4	$\mu = 0.1$	26
9.1	Реализация класса в OpenModelica	27
9.2	$\mu = 0.5, \beta = 3$	27

1 Цель работы

Приобретение навыков моделирования математических моделей с помощью средства имитационного моделирования Scilab, xcos и языка Modelica.

2 Задание

1. Реализовать модель в xcos
2. Реализовать модель с помощью блока Modelica 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.

3 Теоретическое введение

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. Типичная эволюция особи популяции описывается следующей диаграммой:

$S \rightarrow I \rightarrow R$

Считаем, что система замкнута, т.е. $N=S+I+R$.

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.

4 Реализация модели в xcos

Откроем окно Scilab, далее - инструменты - визуальное программирование xcos. 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 β и ν (рис. 4.1).

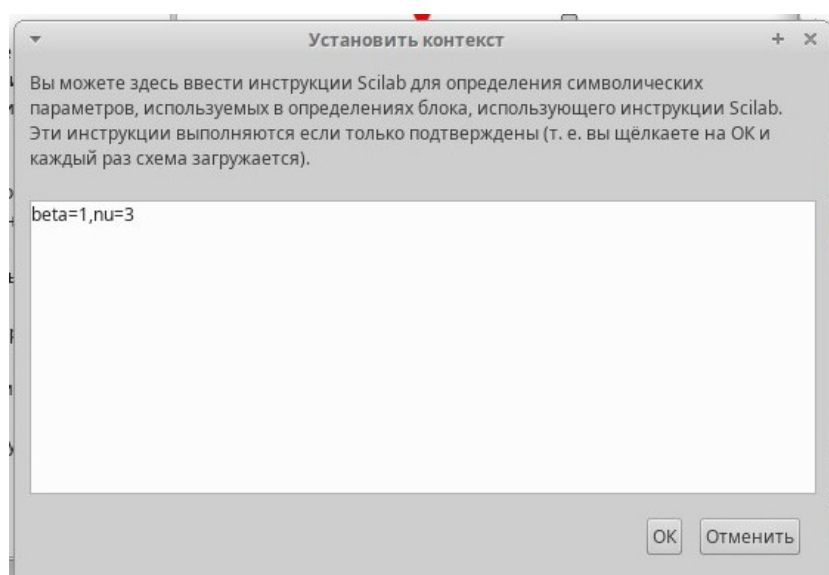


Рис. 4.1: Значения переменных β и ν

Далее начнем добавлять блоки: $GAINBLK_f$ — в данном случае позволяет задать значения коэффициентов β и ν (рис. 4.2 - 4.3).

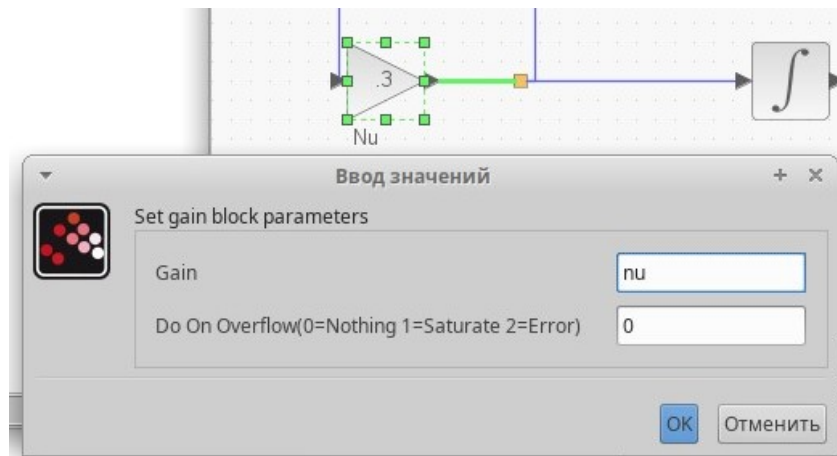


Рис. 4.2: Блок со значением коэффициента β

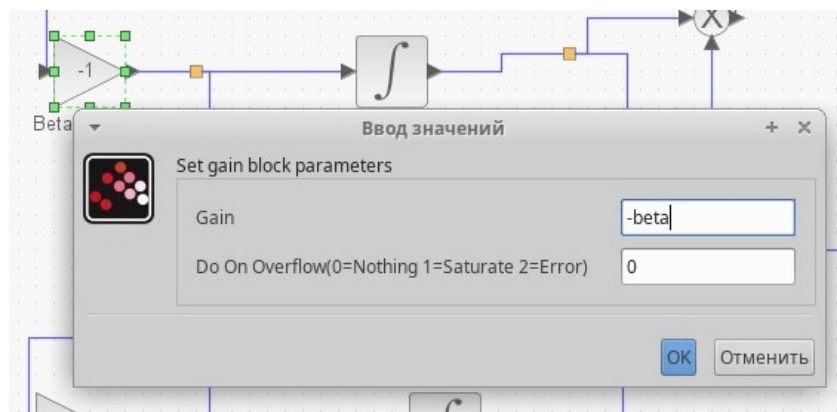


Рис. 4.3: Блок со значением коэффициента ν

Разместим блоки интегрирования - INTEGRAL_m: В параметрах верхнего и среднего блока интегрирования необходимо задать начальные значения

$$s(0) = 0, 999i(0) = 0, 001$$

(рис. 4.4 - 4.6).

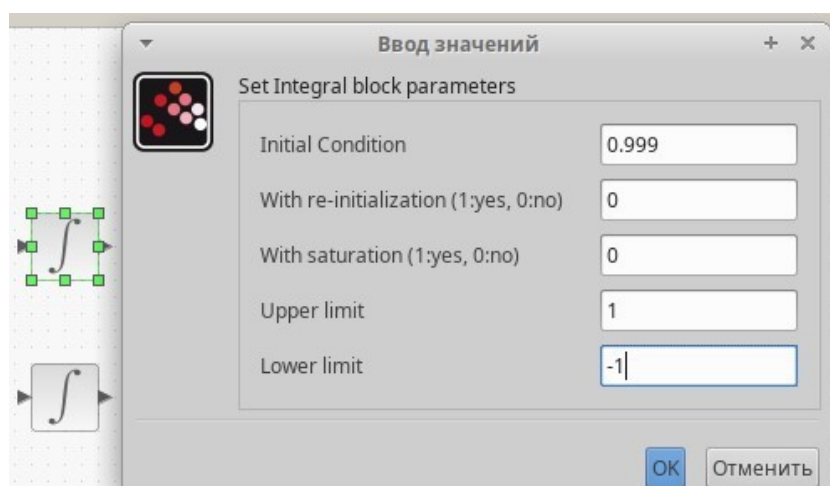


Рис. 4.4: Настройки для верхнего блока интегрирования

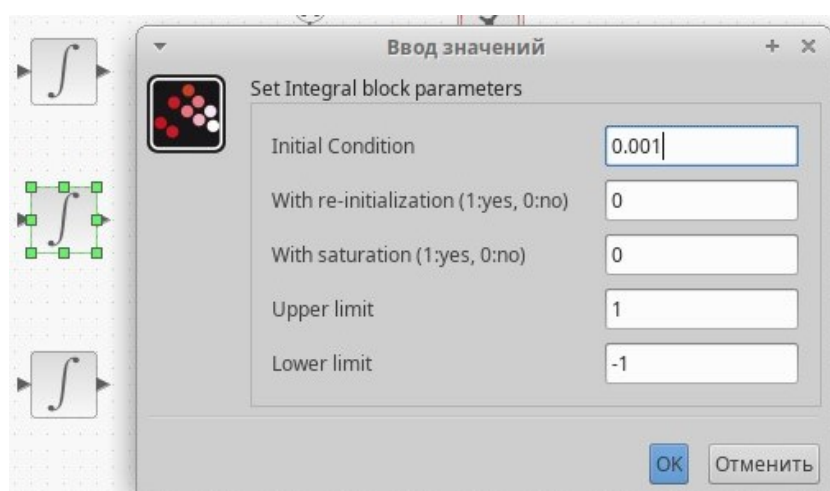


Рис. 4.5: Настройки для среднего блока интегрирования

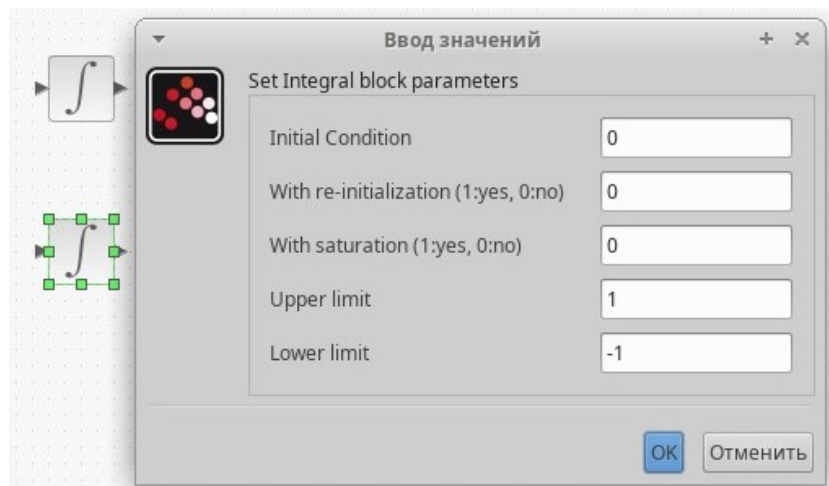


Рис. 4.6: Настройки для нижнего блока интегрирования

В меню “Моделирование -> Установка” необходимо задать конечное время интегрирования, равным времени моделирования (в данном случае 30) (рис. 4.7).

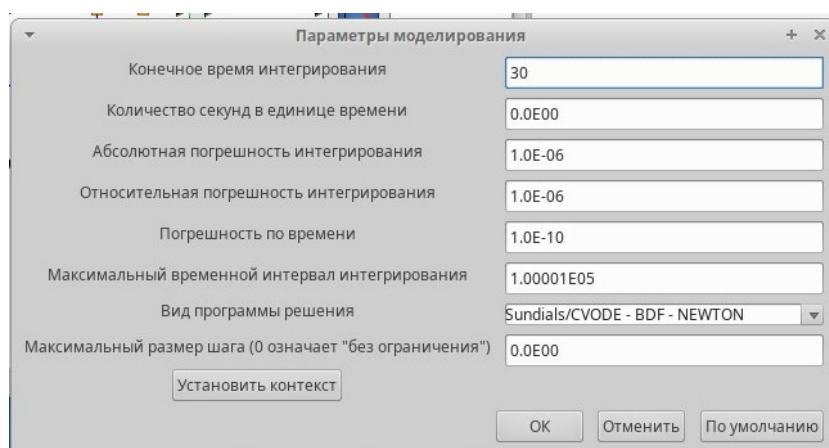


Рис. 4.7: Настройки времени моделирования

Для мультиплексора устанавливаем значение входных каналов равным 3 (рис. 4.8).

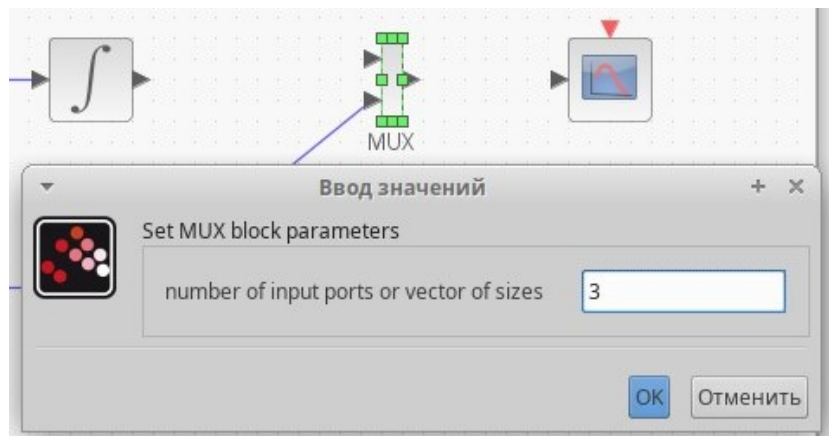


Рис. 4.8: Настройки для мультиплексора

Для регистрирующего устройства - блока *CSCOPE* - устанавливаем максимальное (1) и минимальное значение по оси Oy (рис. 4.9).

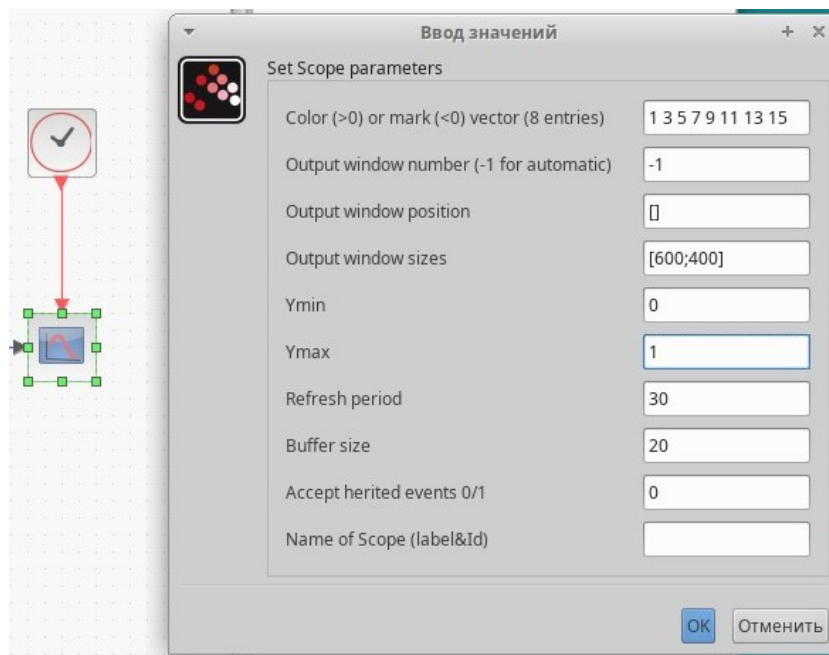


Рис. 4.9: Настройки для регистрирующего устройства

У блока суммирования изменяем значение на входе на -1 (рис. 4.10).

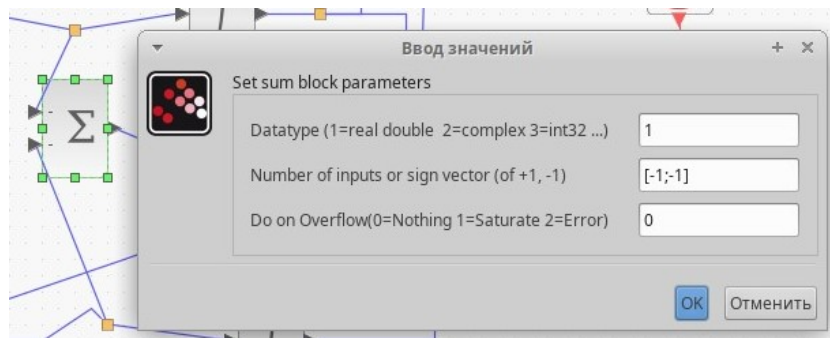


Рис. 4.10: Настройки для блока суммирования

Получаем такую схему (рис. 4.11).

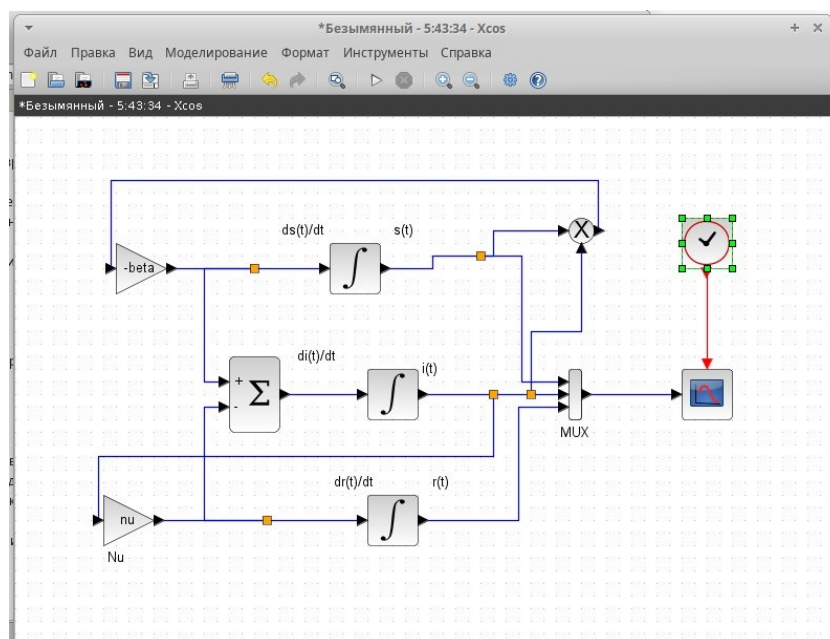


Рис. 4.11: Итоговый вид схемы

При запуске симуляции рисуется график распространения эпидемии (рис. 4.12).

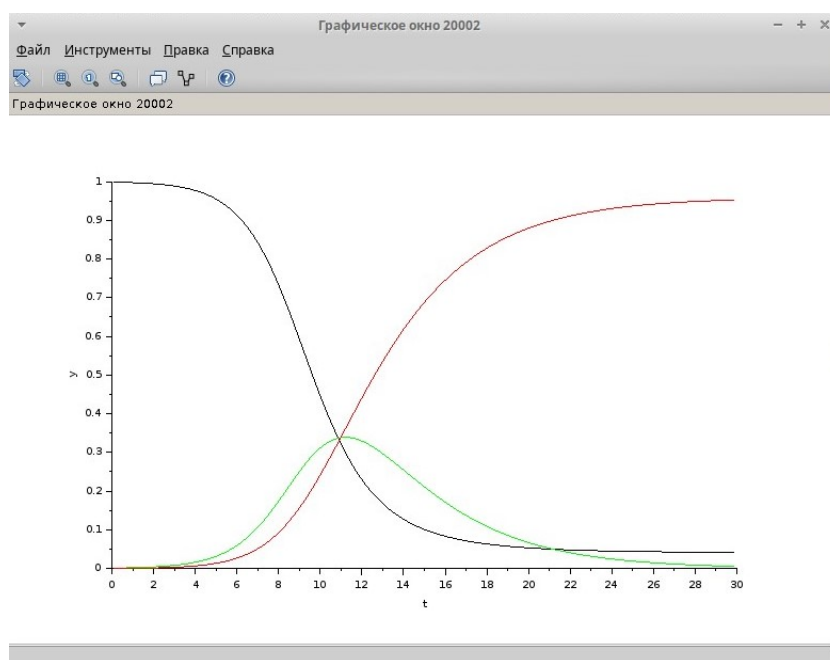


Рис. 4.12: График распространения эпидемии

5 Реализация модели с помощью блока Modelica в xcos

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. Параметры блока Modelica представлены на рис. 5.1. Переменные на входе (β , ν) и выходе (s , i , r) блока заданы как внешние (E).

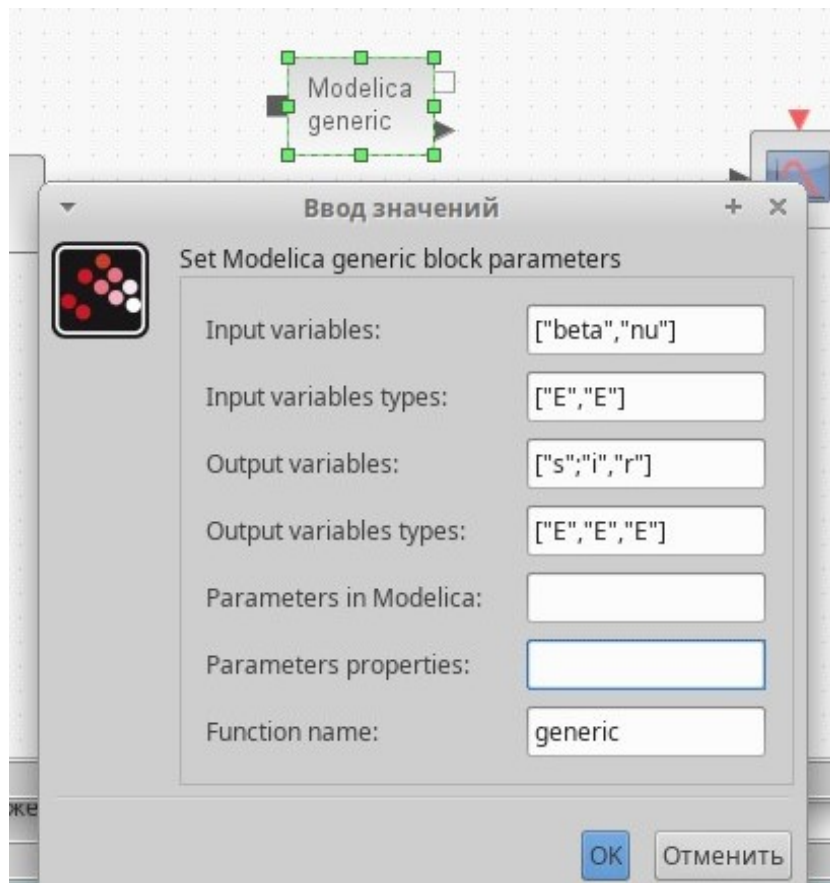


Рис. 5.1: Параметры блока Modelica

Код на языке Modelica: (рис. 5.2)

```
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;

```

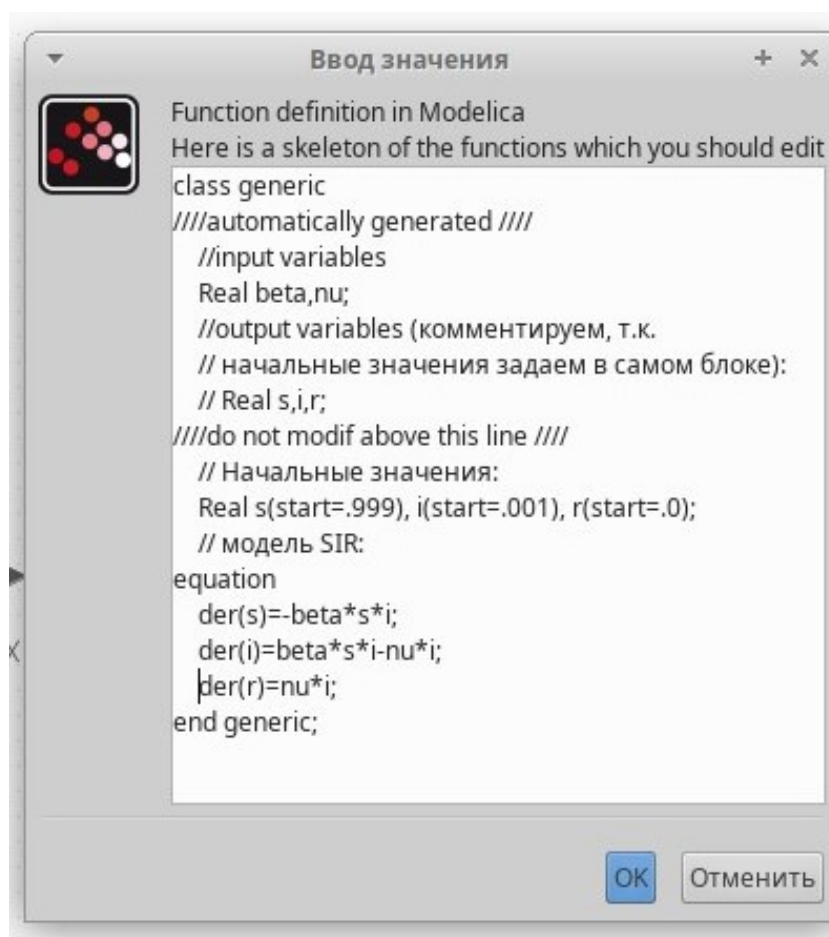


Рис. 5.2: Код для блока Modelica

Получаем такую схему (рис. 5.3).

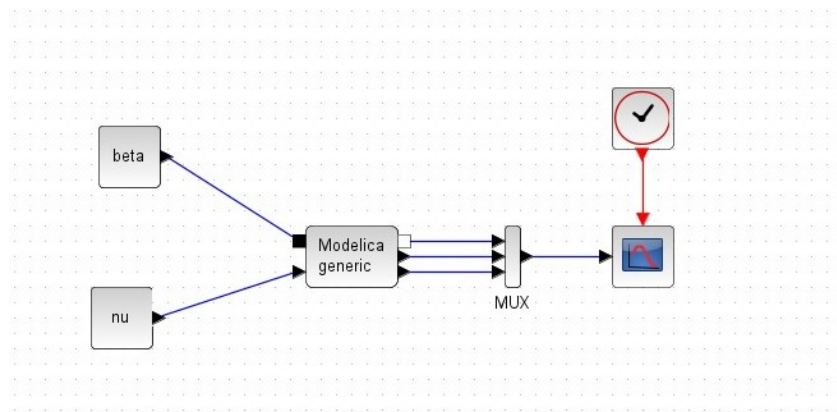


Рис. 5.3: Итоговый вид схемы

При запуске симуляции рисуется график распространения эпидемии (рис. 5.4).

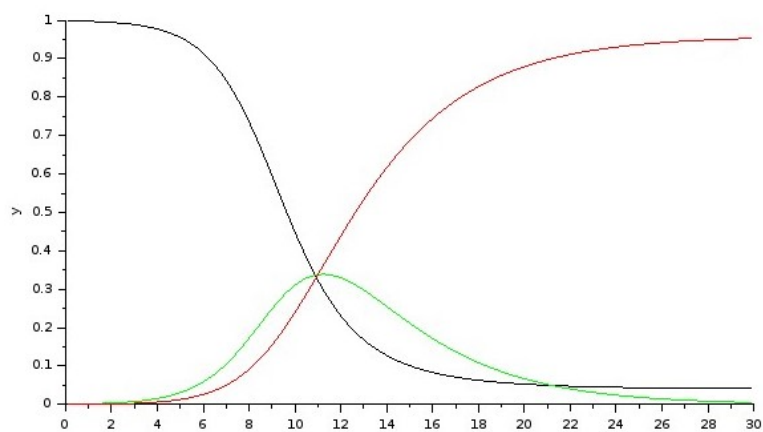


Рис. 5.4: График распространения эпидемии

6 Реализация модели SIR в OpenModelica

Открываем OMEdit, создаем новый класс (рис. 6.1).

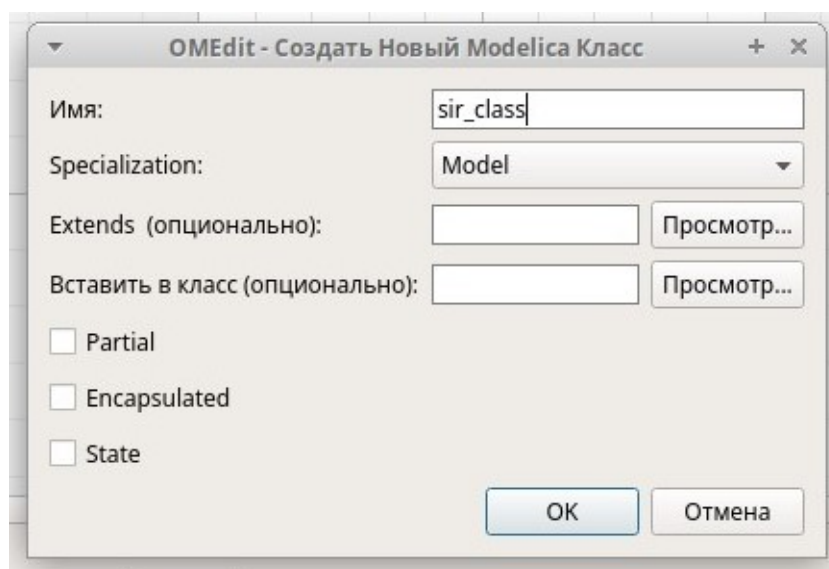


Рис. 6.1: Создание нового класса в OpenModelica

Задаем параметры моделирования, а именно - время - 30 секунд (рис. 6.2).

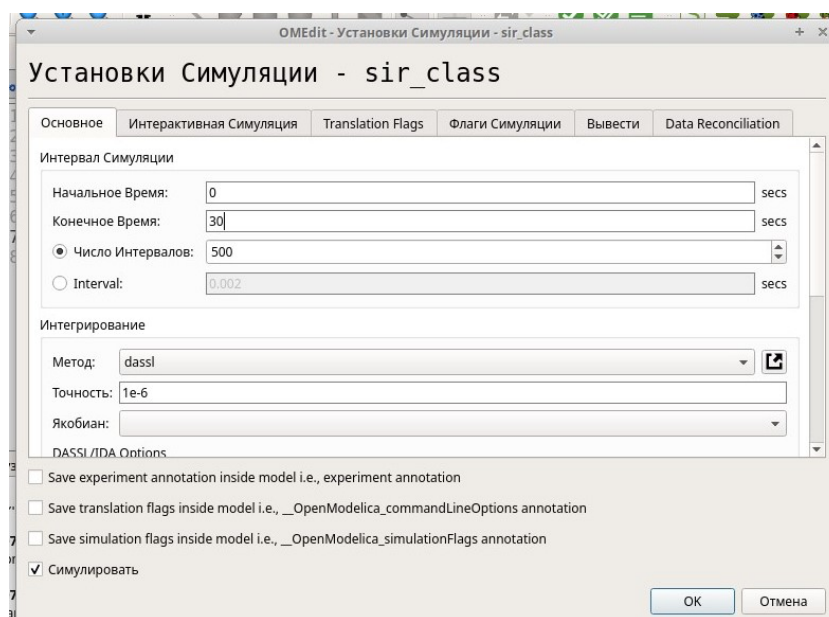


Рис. 6.2: Настройки моделирования в OpenModelica

Реализация класса (рис. 6.3).

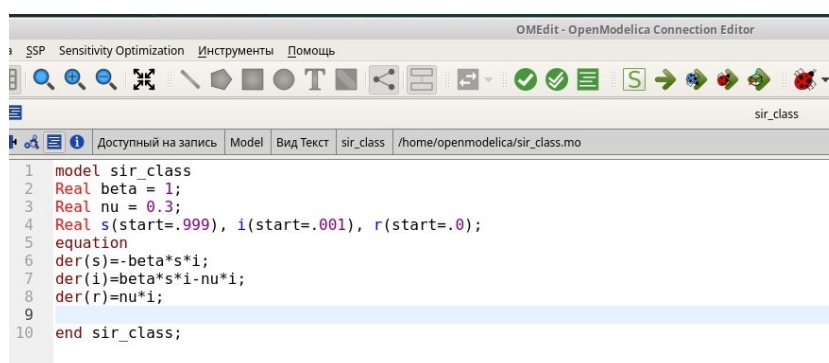


Рис. 6.3: Реализация класса в OpenModelica

При запуске симуляции рисуется график распространения эпидемии. Он аналогичен тем, что были построены с помощью xcos (рис. 6.4).

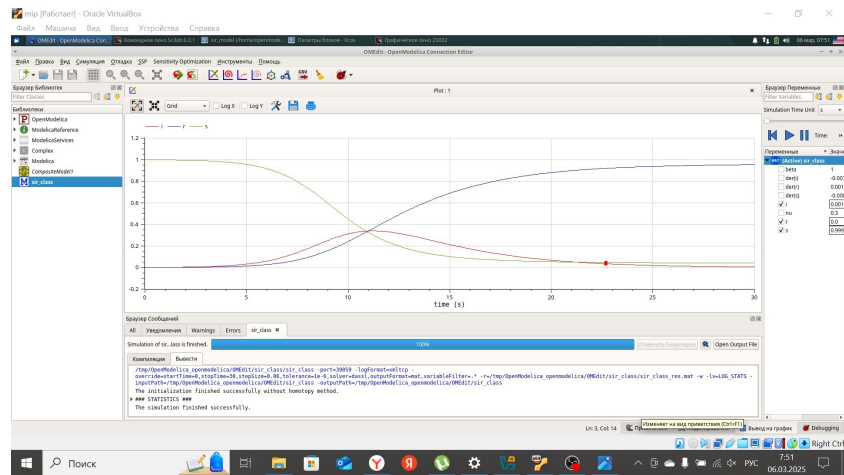


Рис. 6.4: График распространения эпидемии

7 Задание для самостоятельного выполнения в XCOS

В дополнение к предположениям, которые были сделаны для модели SIR, предположим, что учитываются демографические процессы, в частности, что смертность в популяции полностью уравнивает 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.

Зададим значения переменных β , ν и μ (рис. 7.1).

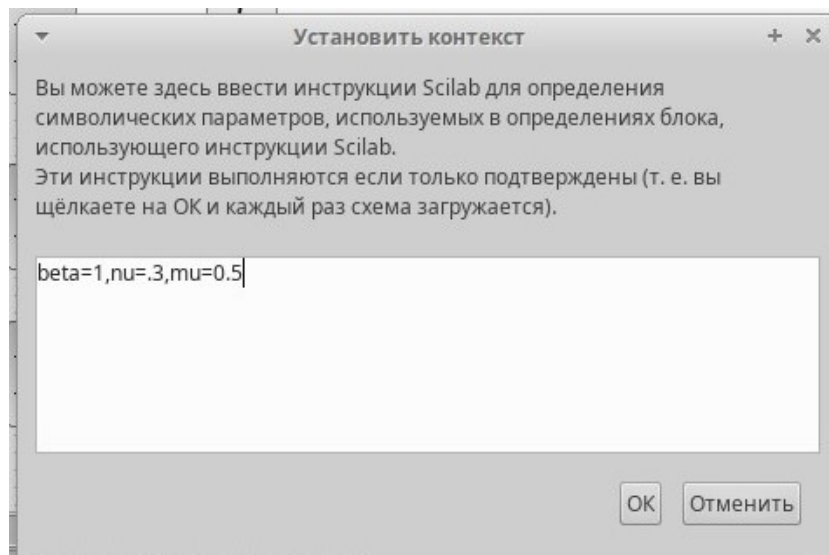


Рис. 7.1: Значения переменных β , ν и μ

Получаем такую схему (рис. 7.2).

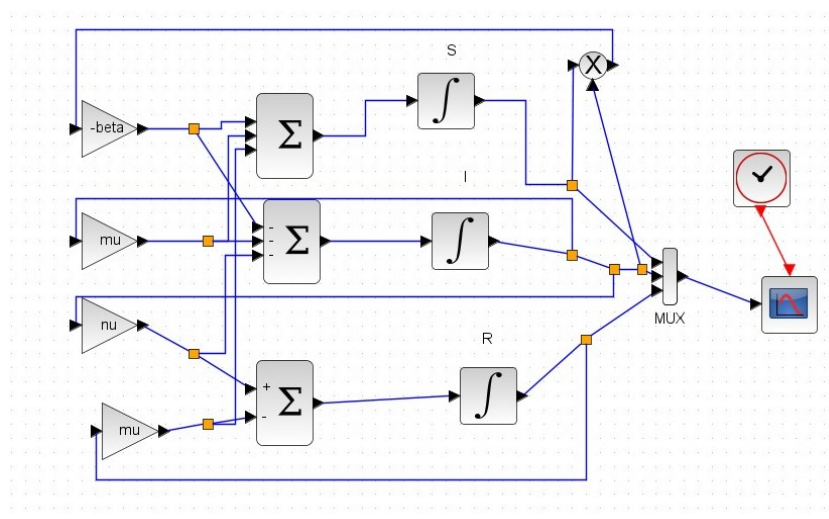


Рис. 7.2: Итоговый вид схемы

При запуске симуляции рисуется график распространения эпидемии (рис. 7.3).

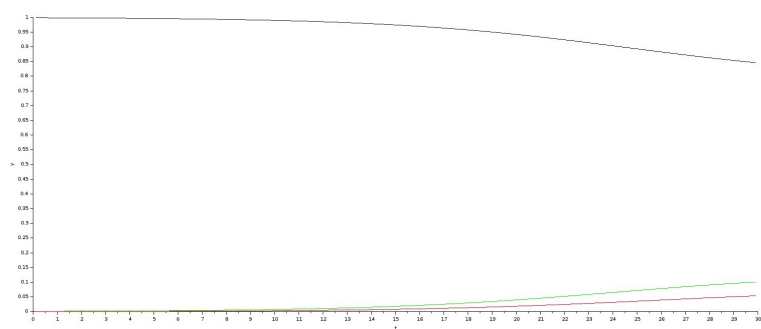


Рис. 7.3: $\mu = 0.5$

Далее я решила увеличить время моделирования до 60 секунд, оставив значение $\mu = 0.5$ (рис. 7.4).

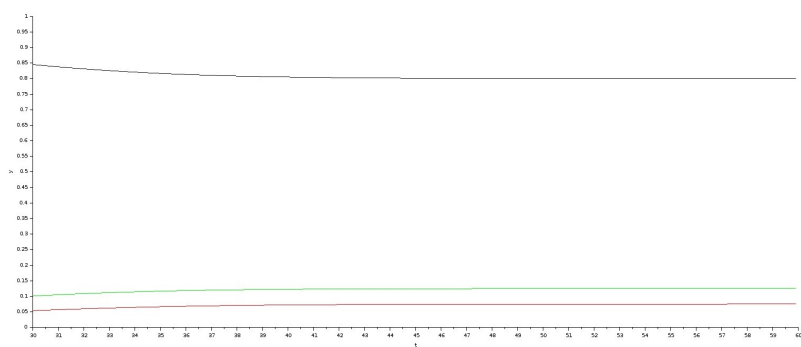


Рис. 7.4: Время моделирования - 60 секунд, $\mu = 0.5$

Другие варианты графиков распространения эпидемии (рис. 7.5 - 7.6).

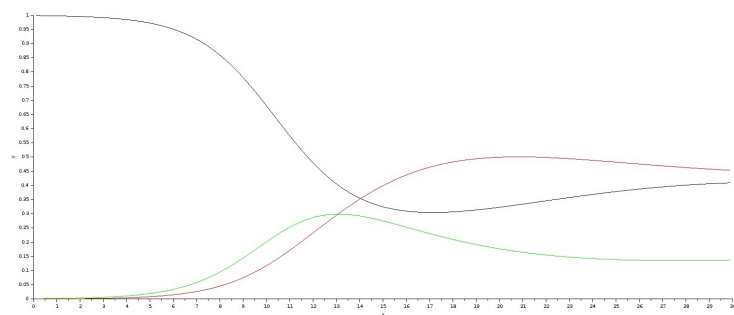


Рис. 7.5: $\mu = 0.1$

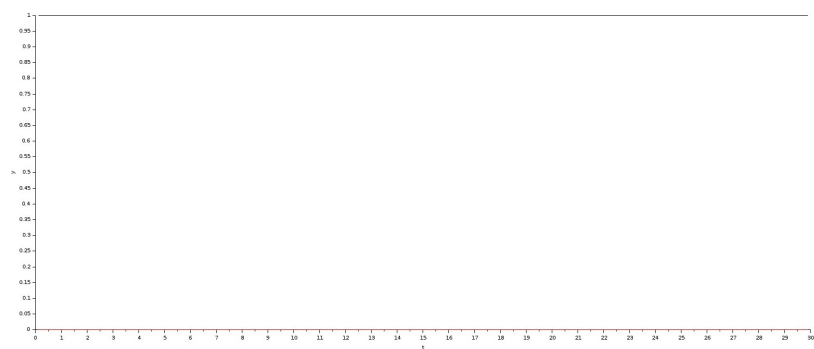


Рис. 7.6: $\mu = 0.9$

8 Задание для самостоятельного выполнения с помощью блока Modelica в xcos

Параметры блока Modelica представлены на рис. 8.1. Переменные на входе (β , ν) и выходе (s , i , r) блока заданы как внешние (E).

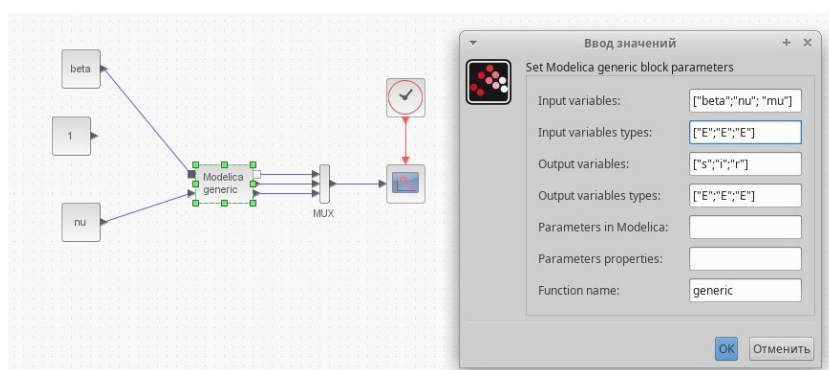


Рис. 8.1: Параметры блока Modelica

Код на языке Modelica: (рис. 8.2)

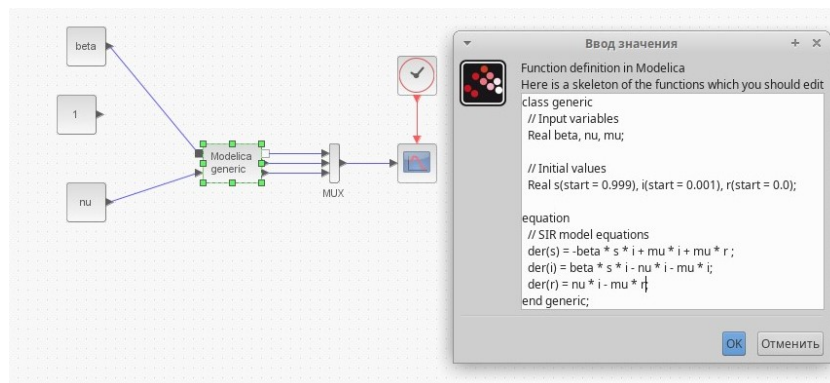


Рис. 8.2: Код для блока Modelica

Другие варианты графиков распространения эпидемии (рис. 8.3 - 8.4).

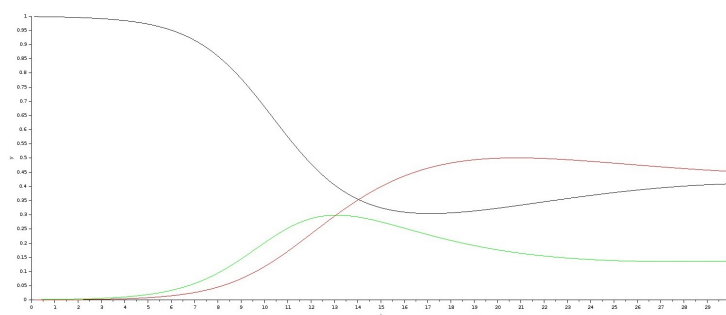


Рис. 8.3: $\mu = 0.5, \beta = 3$

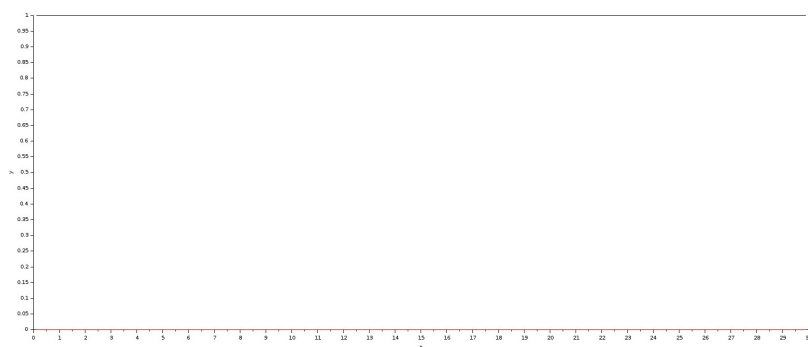


Рис. 8.4: $\mu = 0.1$

9 Задание для самостоятельного выполнения в OpenModelica

Реализация класса (рис. 9.1).

```
1 model sir_model_ex
2
3 Real beta = 1;
4 Real nu = 0.3;
5 Real mu = 0.5;
6 Real s(start=.999), i(start=.001), r(start=.0);
7 equation
8   der(s) = -beta * s * i + mu * i + mu * r;
9   der(i) = beta * s * i - nu * i - mu * i;
10  der(r) = nu * i - mu * r;
11
12 end sir_model_ex;
```

Рис. 9.1: Реализация класса в OpenModelica

При запуске симуляции рисуется график распространения эпидемии. Он аналогичен тем, что были построены с помощью xcos (рис. 9.2).

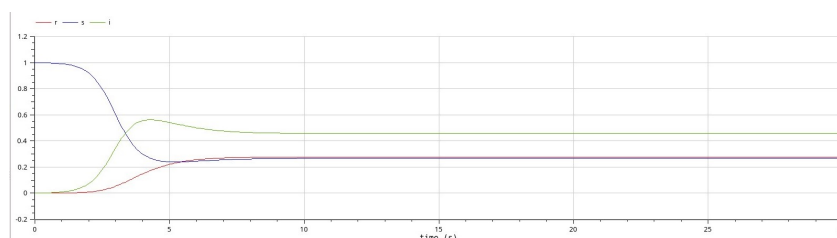


Рис. 9.2: $\mu = 0.5, \beta = 3$

10 Выводы

В ходе выполнения лабораторной работы я приобрела навыки моделирования математических моделей с помощью средства имитационного моделирования Scilab, xcos и языка Modelica.