Лабораторная работа №5

Имитационное моделирование

Александрова Ульяна Вадимовна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Целью данной работы является получение навыков создания модели эпидемии (SIR) при помощи утелит Sci-Lab и OpenModelica.

2 Задание

1. Проделать пример из методического материала;
2. Проделать упражнение;
3. Выполнить задание для самостоятельной работы.

3 Теоретическое введение

Модель SIR предложена в 1927 г. (W. O. Kermack, A. G. McKendrick).

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).

Если предположить, что каждый член популяции может контактировать с каждым, то 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.

Первое уравнение описывает динамику численности уязвимых к болезни особей: заражённая особь с некоторой скоростью заражает уязвимую особь.

Третье уравнение описывает динамику выздоровления заражённой особи: с некоторой скоростью инфицированная особь выздоравливает.

Второе уравнение описывает динамику численности заражённых особей: разность числа заражённых особей и числа выздоровевших особей.

4 Выполнение лабораторной работы

4.1 Реализация модели в xcos

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

В меню Моделирование, задаю переменные окружения. Для создания модели, используем следующие блоки:

- `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.

Перед моделированием, настроим параметры моделирования (рис. 4.1).

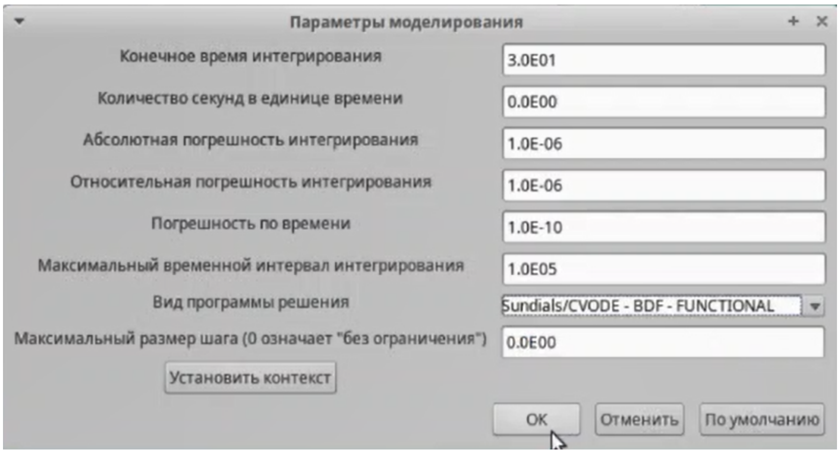

Рис. 4.1: Параметры моделирования

Готовая модель выглядит следующим образом (рис. 4.2).

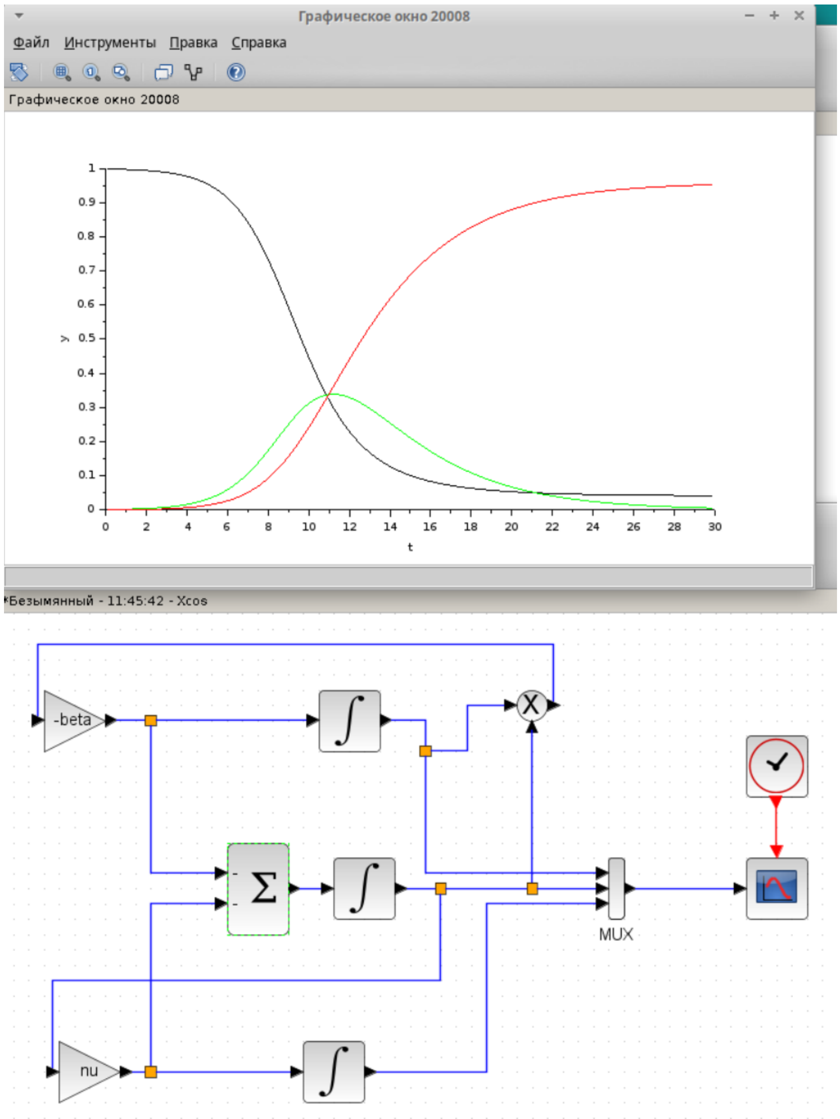

Рис. 4.2: График модели эпидемии в xcos

На этом графике красным цветом обозначена численность выздоровевших, зеленым - численность зараженных, а черным - уязвимых к болезни. Мы видим, что в какой-то момент времени все графики перескаются в одной точке - пике эпидемии, когда число зараженных макисмально. После прохождения этой точки, число выздоровевших многократно увеличивается.

4.2 Реализация модели с помощью блока Modelica в xcos

Теперь составим эту же модель через блок Modelica. Настроим его и напишем код для блока:

```
////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;
```

Готовая модель аналогична предыдущей (рис. 4.3).

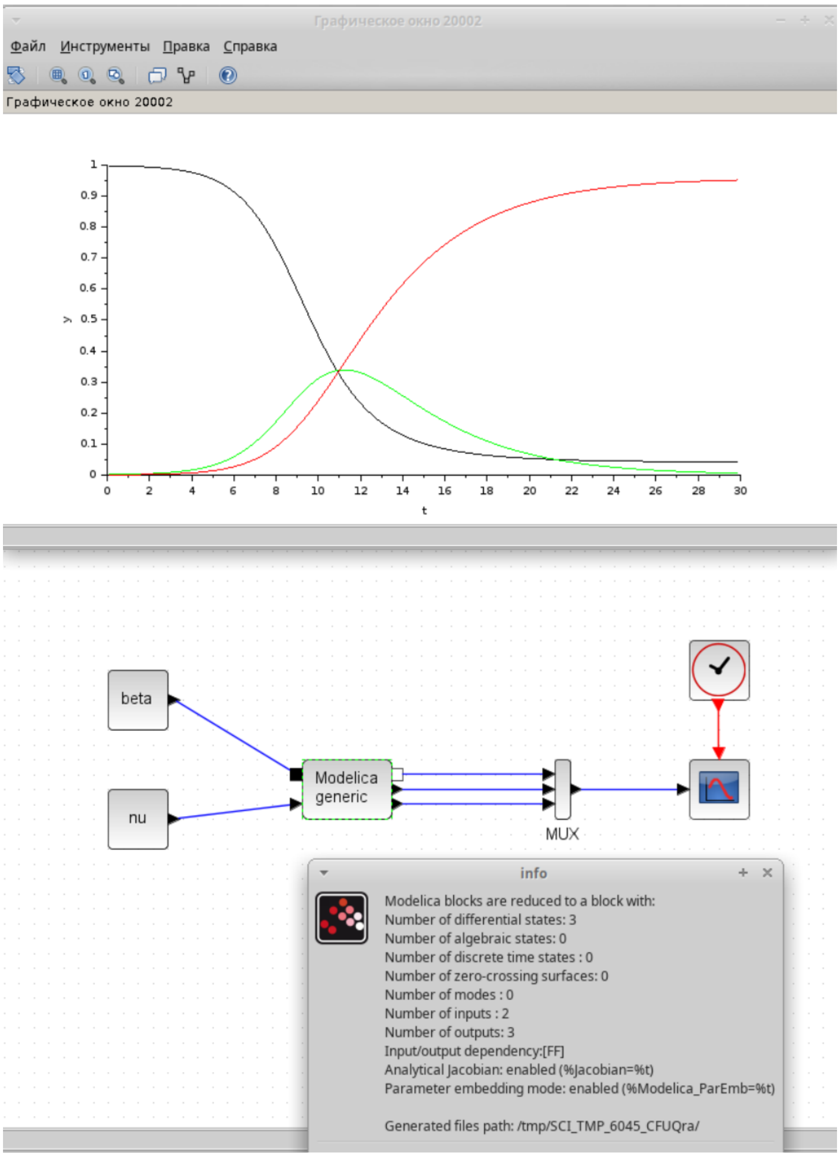

Рис. 4.3: График модели эпидемии с блоком

4.3 Реализация модели SIR в OpenModelica

Открываю OMEdit и создаю новый файл SIR.mo для реализации модели. В окне “Вид текста”, расписываю модель (рис. 4.4).

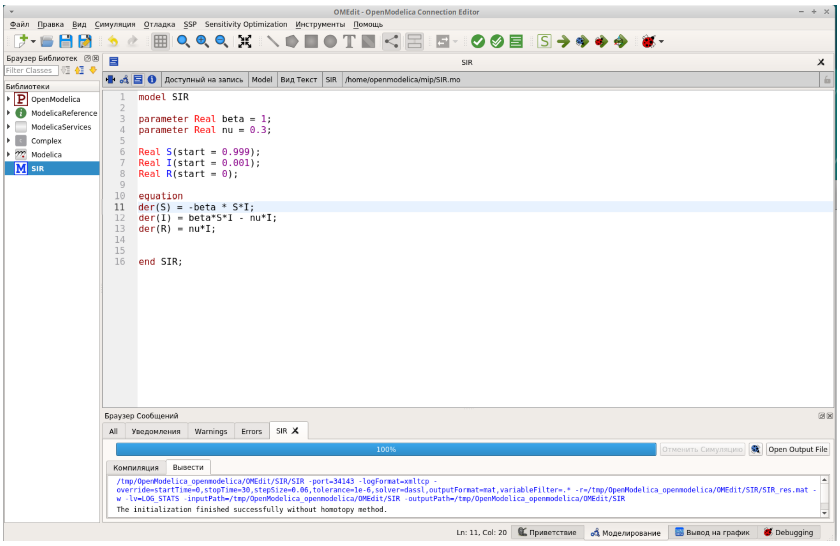

Рис. 4.4: Листинг программы в OpenModelica

График модели идентичен предыдущим моделям (рис. 4.5).

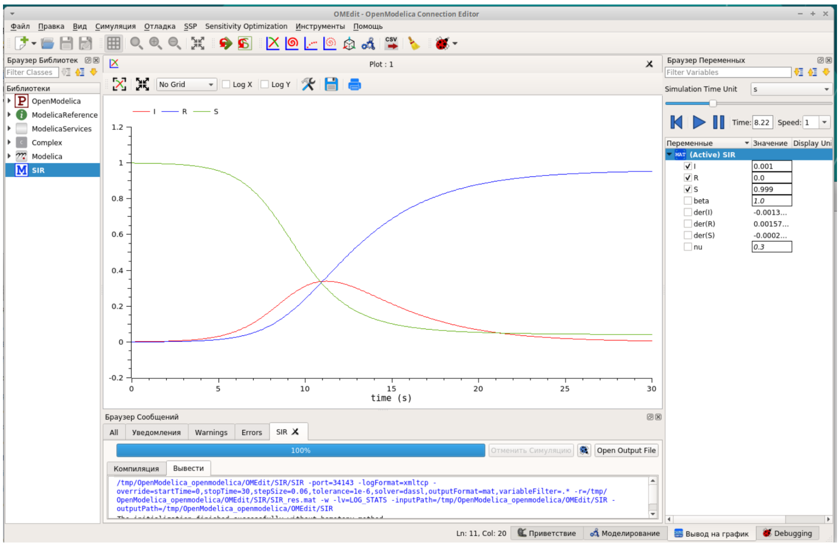

Рис. 4.5: График модели в OpenModelica

4.4 Задание для самостоятельного выполнения

В дополнение к предположениям, которые были сделаны для модели SIR, предположим, что учитываются демографические процессы, в частности, что смертность в популяции полностью уравновешивает рождаемость, а все рожденные индивидуу мыпоявляются на свет абсолютно здоровыми. Тогда получим следующую систему уравнений:

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

Сначала посмотрим систему в xcos при mu = 0.1 (рис. 4.6).

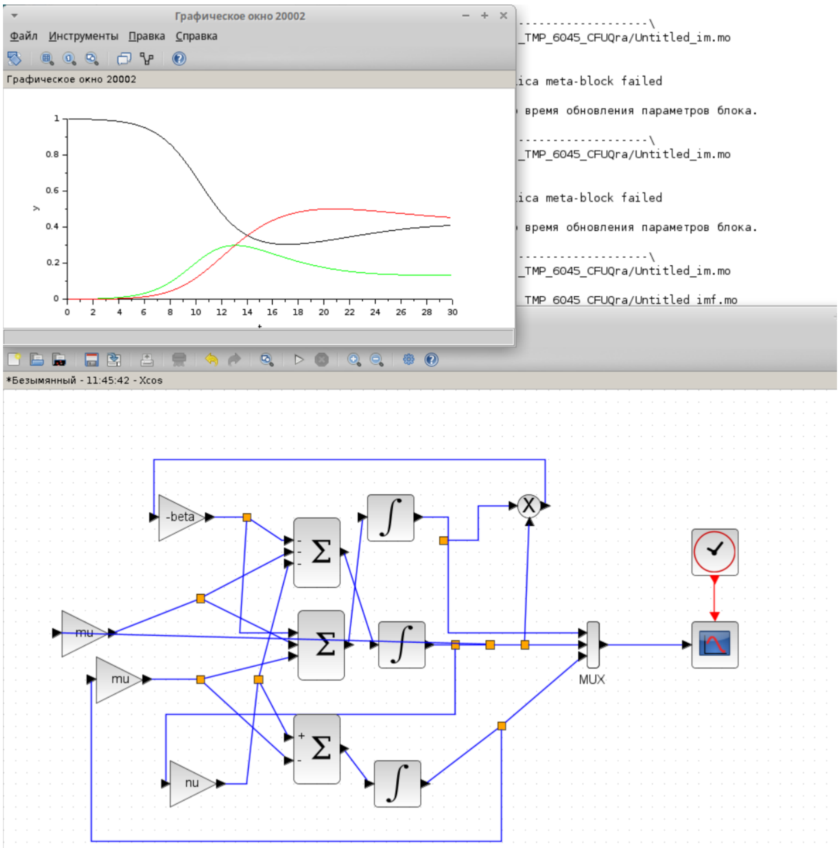

Рис. 4.6: График модели в Xcos, mu = 0,1

Мы можем заметить, что после пика эпидемии, число выздоровевших и уязвимых к болезни уравнивается.

Теперь положим, что mu = 0.3 (рис. 4.7).

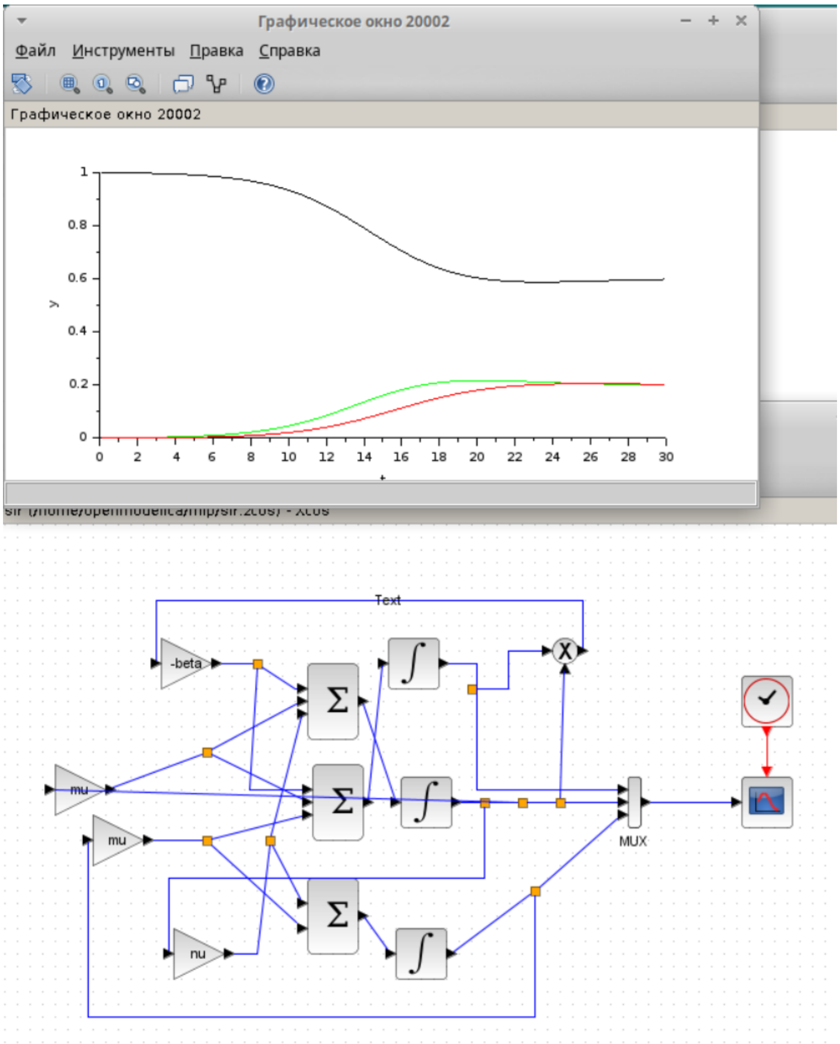

Рис. 4.7: График модели в Xcos, mu = 0,3

Мы можем наблюдать еще более сильное выравнивание всех показателей и сглаживание пика эпидемии.

Если mu = 0, то график не меняется в сравнении с нашей прошлой моделью (рис. 4.8).

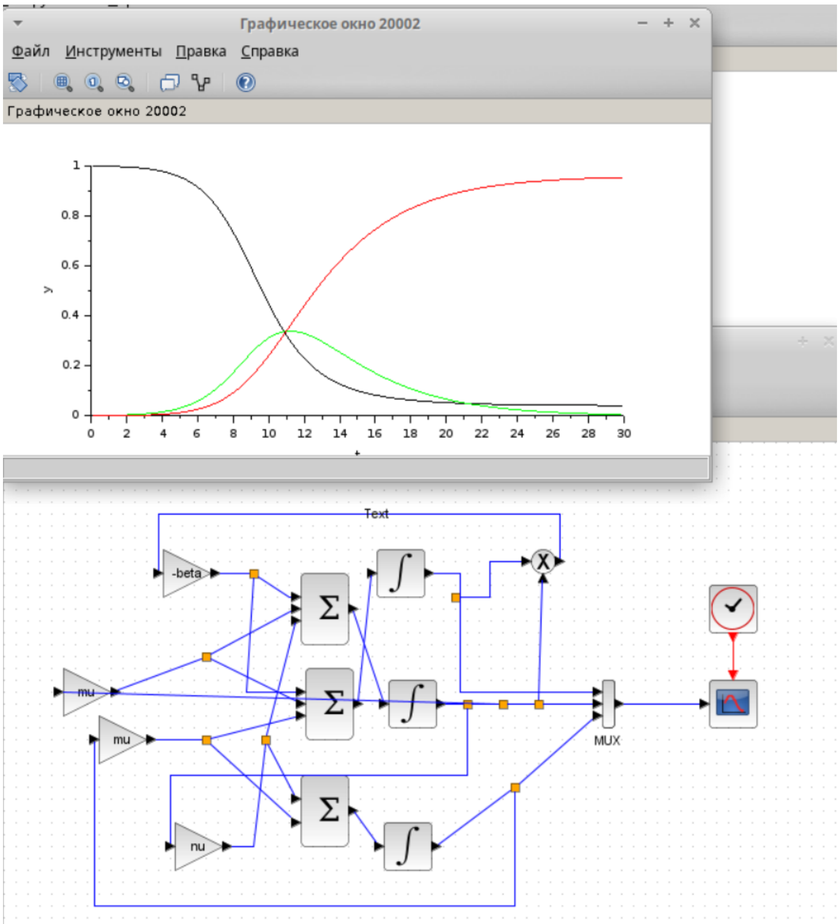

Рис. 4.8: График модели в Xcos, mu = 0

Теперь построим график через блок Modelica. Настроим показатели (рис. 4.9), (рис. 4.10).

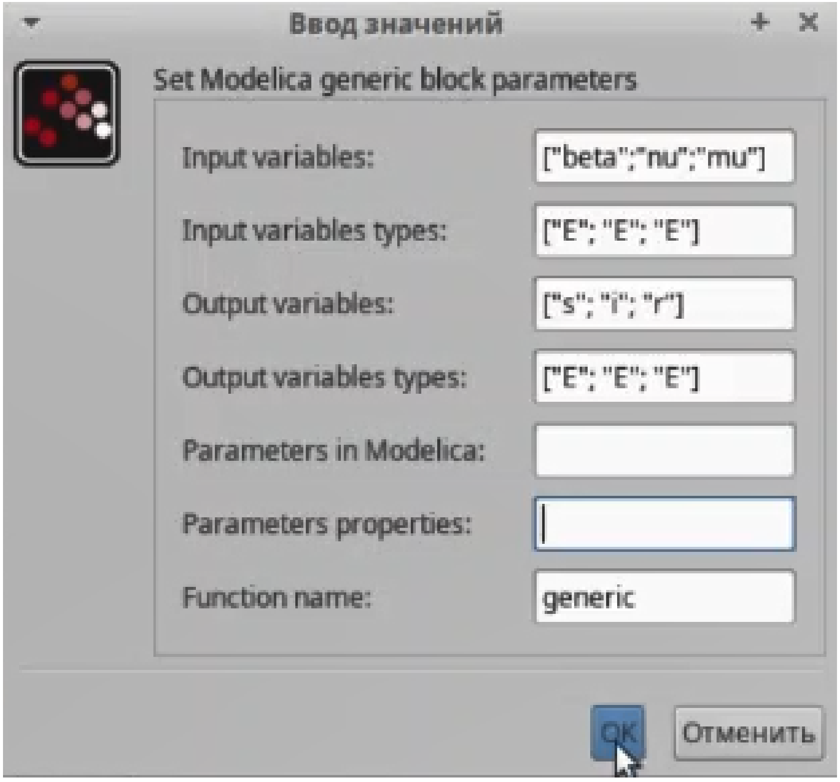

Рис. 4.9: Ввод значений

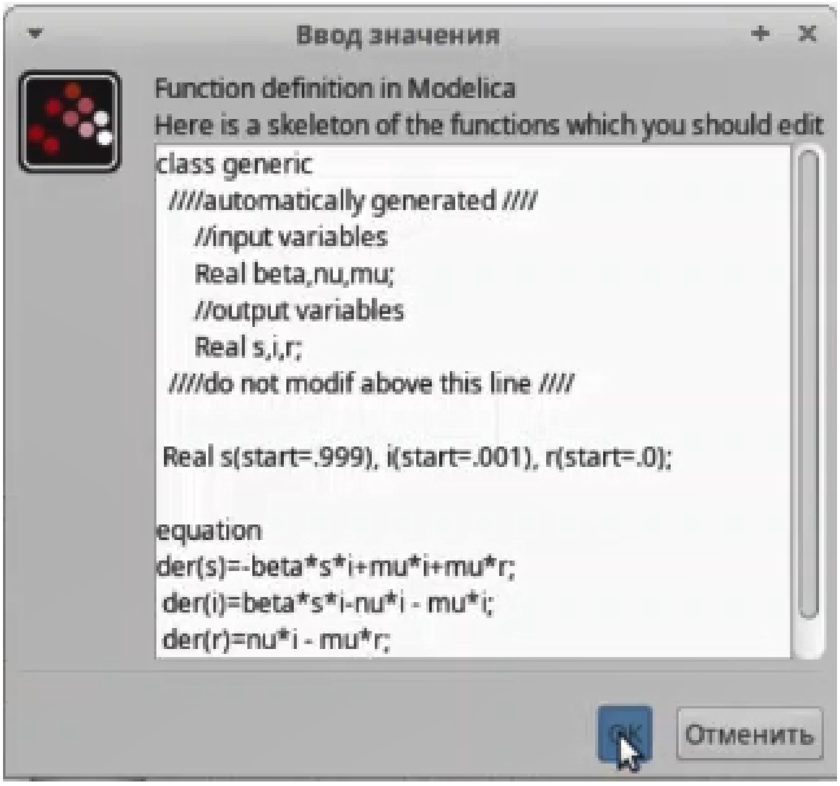

Рис. 4.10: Ввод значений. Код

Симулируем график (рис. 4.11).

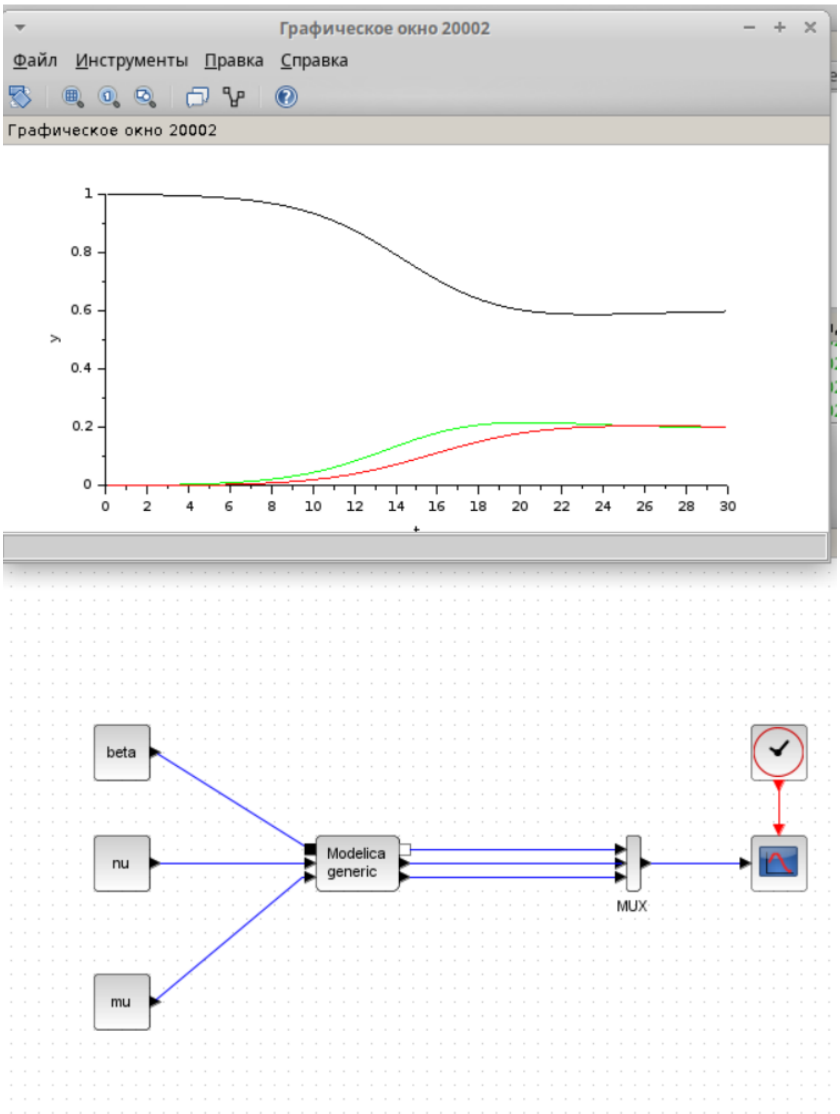

Рис. 4.11: График модели через блок, mu = 0,3

Проделаем все те же шаги через OpenModelica. Напишу код:

“‘model SIR

parameter Real beta = 1; parameter Real nu = 0.3; parameter Real mu = 0.5;

Real s(start = 0.999); Real i(start = 0.001); Real r(start = 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;

end SIR;

“‘

Запустим симуляцию (рис. 4.12).

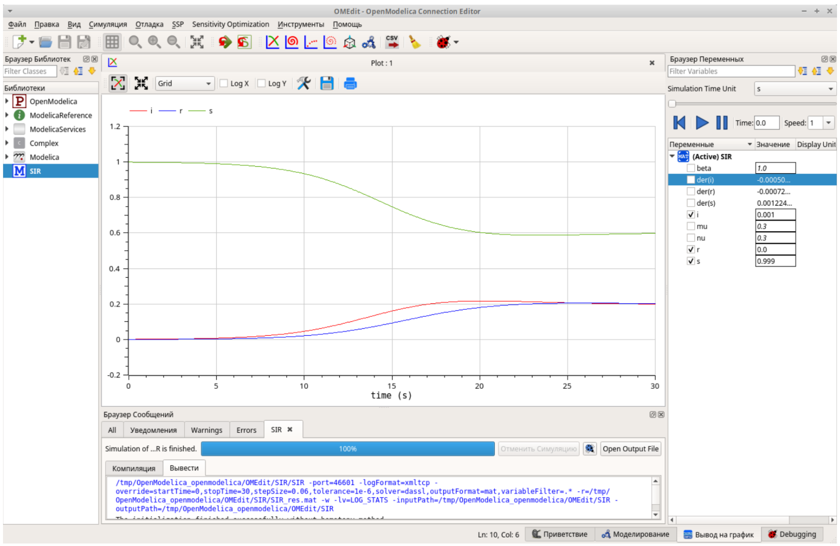

Рис. 4.12: График модели через OpenModelica, mu = 0,3

5 Выводы

Я построила модель эпидемии, используя разные утилиты.