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: **β = 1, ν = 0, 3, s(0) = 0, 999, i(0) = 0, 001, r(0) = 0**.

В меню моделирования устанавливаем переменные окружения (рис. 1)

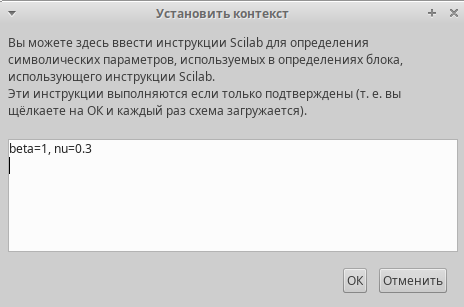


Рис. 1: beta, nu

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:

где – скорость заражения, – скорость выздоровления.

Реализованная модель эпидемии. Выходы трёх блоков интегрирования соединяем с мультиплексором.(рис. 2)

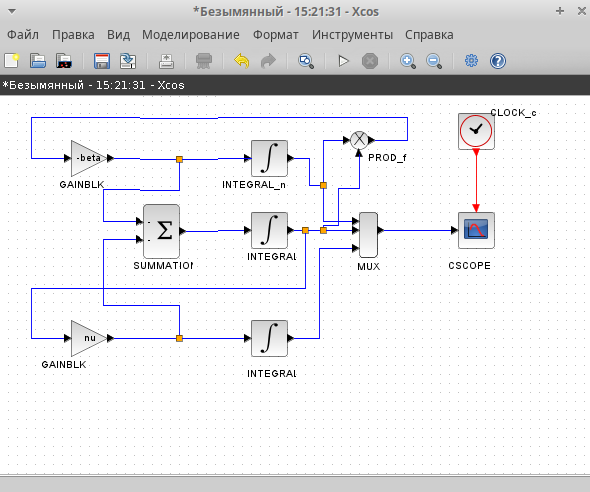


Рис. 2: Реализованная модель эпидемии

В параметрах верхнего блока интегрирования задаем значения s(0) = 0, 999, который отвечает за здоровых особей. (рис. 3)

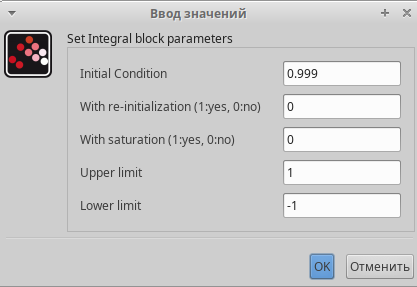


Рис. 3: Начальные значения для верхнего блока интегрирования

В параметрах среднего блока интегрирования задаем значения i(0) = 0, 001, который отвечает за переносчиков болезни. (рис. 4)

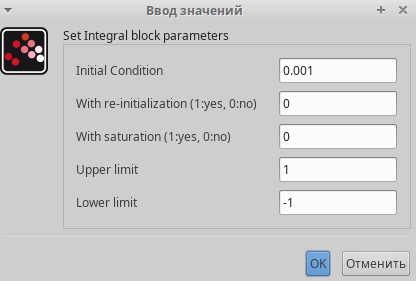


Рис. 4: Начальные значения для среднего блока интегрирования

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. 5)

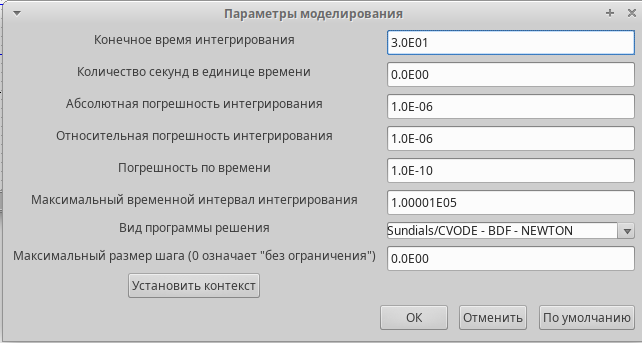


Рис. 5: Конечное время интегрирования

Результат моделирования представлен на (рис. 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.

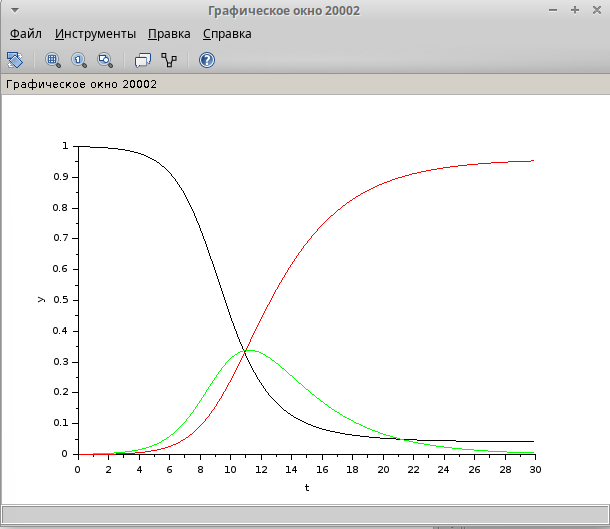


Рис. 6: Модель эпидемии при beta=1, nu=0.3

## 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. 7)

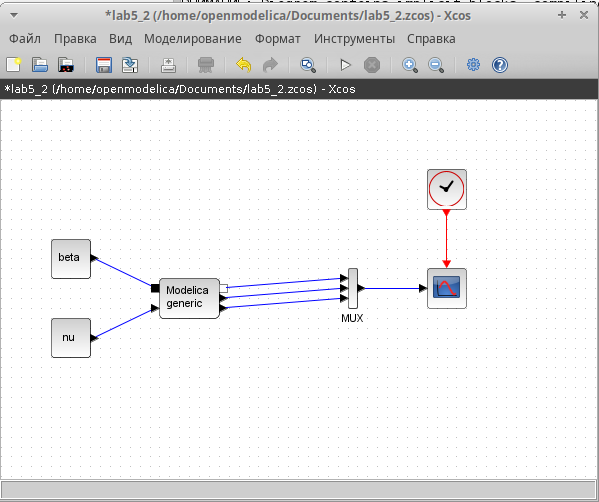


Рис. 7: Модель эпидемии

Указываем параметры для блока реализации. Переменные на входе (“beta”, “nu”) и выходе (“s”, “i”, “r”) блока заданы как внешние (“E”). (рис. 8)

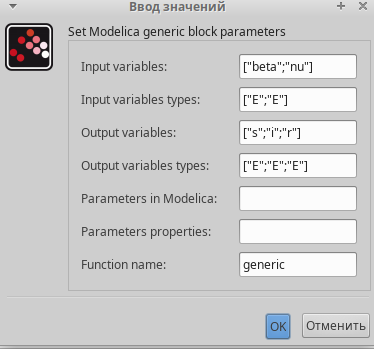


Рис. 8: Параметры блока реализации

Код на языке Modelica. Задаем переменные beta, nu. Указываем начальные значения для s, i, r и пишем систему уравнения. (рис. 9)

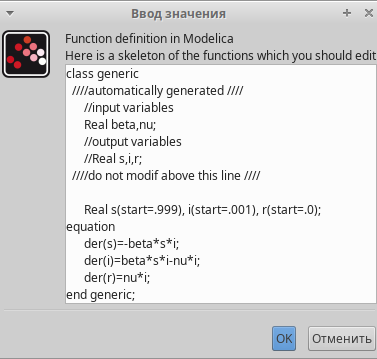


Рис. 9: Параметры блока реализации

Результат работы модели. Он идентичен с реализацией в xcos. (рис. 10)

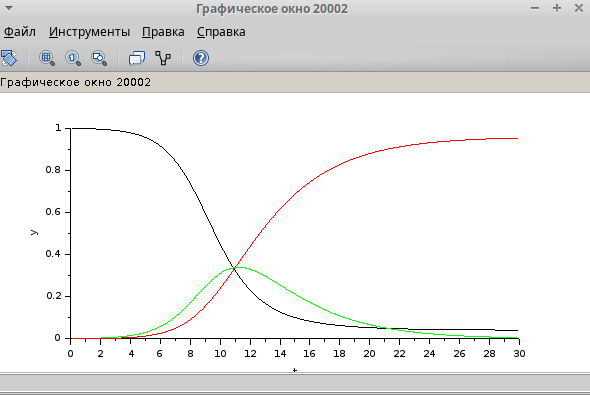


Рис. 10: Модель эпидемии Modelica

## 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. 11)

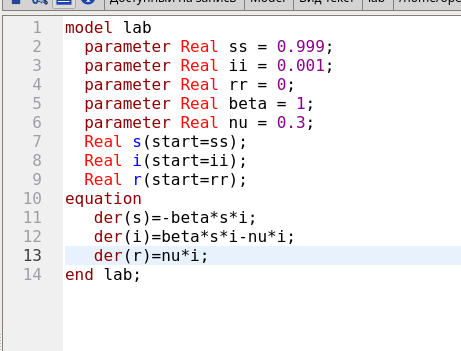


Рис. 11: Реализация модели эпидемии в OpenModelica

Результат модели. Результат идентичен с построением с помощью других способов, значит все выполнено правильно. (рис. 12)

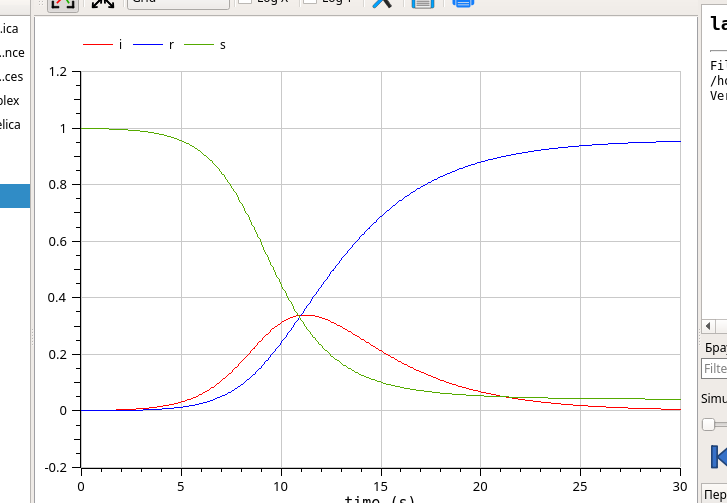


Рис. 12: Модель эпидемии в OpenModelica

## 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:

где — константа, которая равна коэффициенту смертности и рождаемости.

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

В меню моделирования устанавливаем переменные окружения. (рис. 13) Реализация с помощью xcos. (рис. 14)

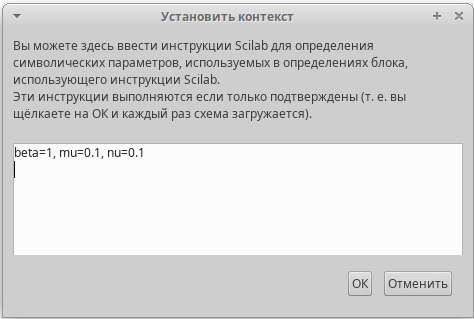


Рис. 13: Переменные окружения

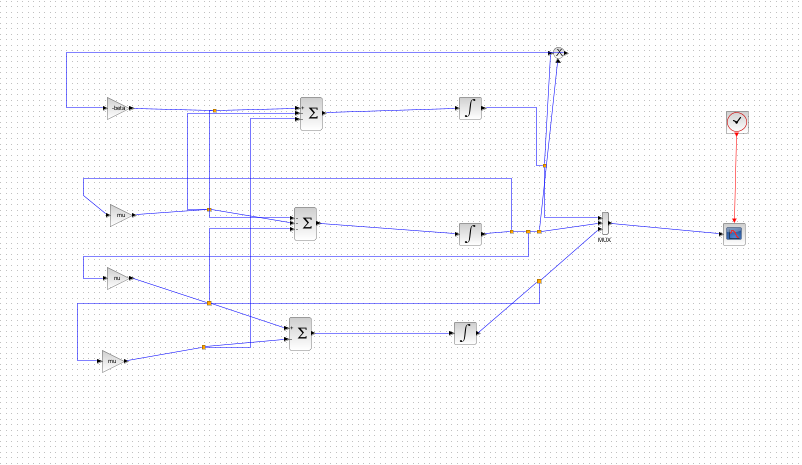


Рис. 14: Реализация модели эпидемии с учетом процесса рождения / гибели особей с помощью xcos

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

Результат моделирования представлен на (рис. 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.

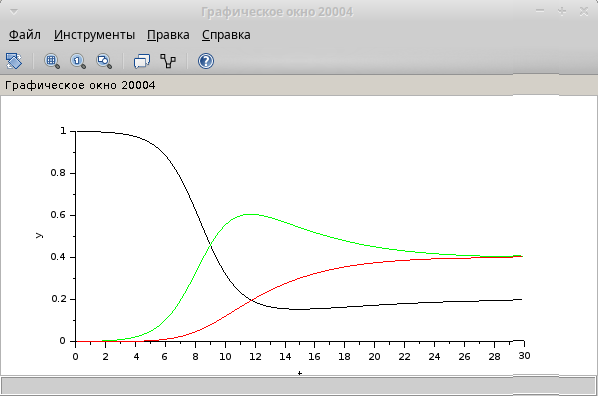


Рис. 15: Модель эпидемии при beta=1, nu=0.1, mu=0.1

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

Реализация с помощью блока Modelica в xcos. (рис. 16)

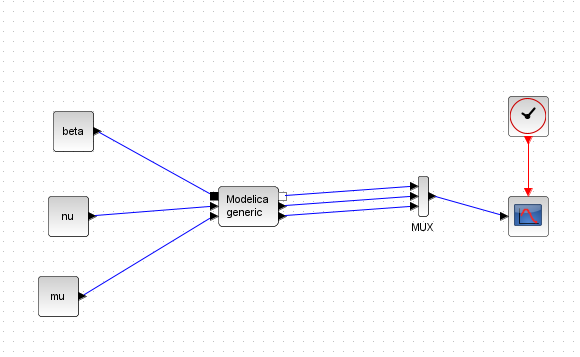


Рис. 16: Реализация модели эпидемии с учетом процесса рождения / гибели особей с помощью блока Modelica в xcos

Указываем параметры для блока реализации. Переменные на входе (“beta”, “nu”, “mu”) и выходе (“s”, “i”, “r”) блока заданы как внешние (“E”). (рис. 17)

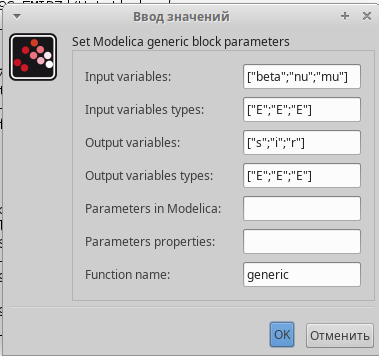


Рис. 17: Параметры блока реализации

Код на языке Modelica. Задаем переменные beta, nu, mu. Указываем начальные значения для s, i, r и пишем систему уравнения. (рис. 18)

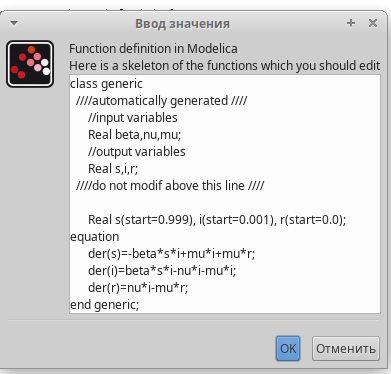


Рис. 18: Параметры блока реализации

Результат моделирования представлен на (рис. 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.

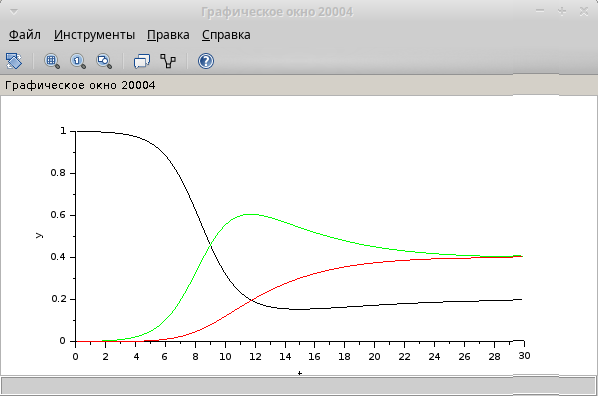


Рис. 19: Модель эпидемии при beta=1, nu=0.1, mu=0.1

## 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. 20)

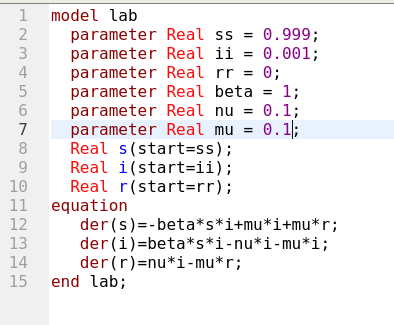


Рис. 20: Реализация модели с учетом процесса рождения / гибели особей эпидемии в OpenModelica

Результат модели. Результат идентичен с построением с помощью других способов, значит все выполнено правильно. (рис. 21)

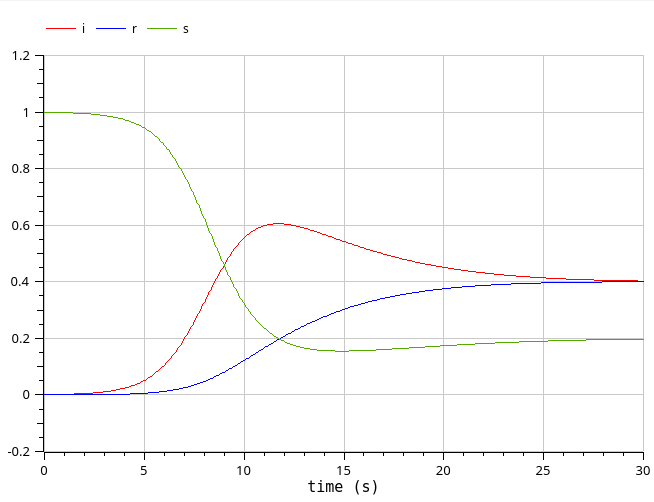


Рис. 21: Модель эпидемии с учетом процесса рождения / гибели особей в OpenModelica

## 4.7 Результаты на различных параметрах.

При mu=0.6, nu=0.1, beta=1 (рис. 22), (рис. 23)

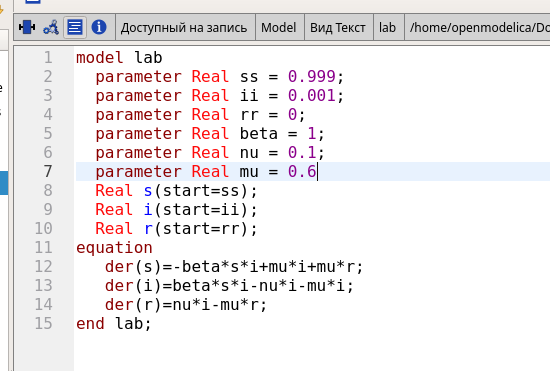


Рис. 22: Результаты на различных параметрах.

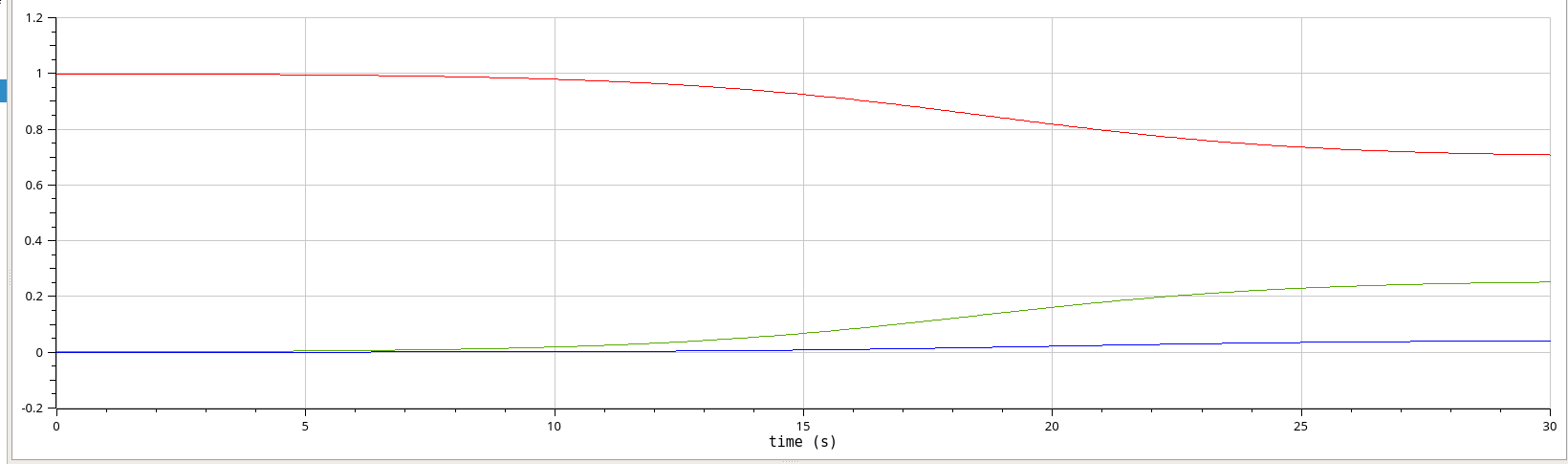


Рис. 23: Результаты на различных параметрах.

При mu=0.6, nu=0.6, beta=1 (рис. 24), (рис. 25)

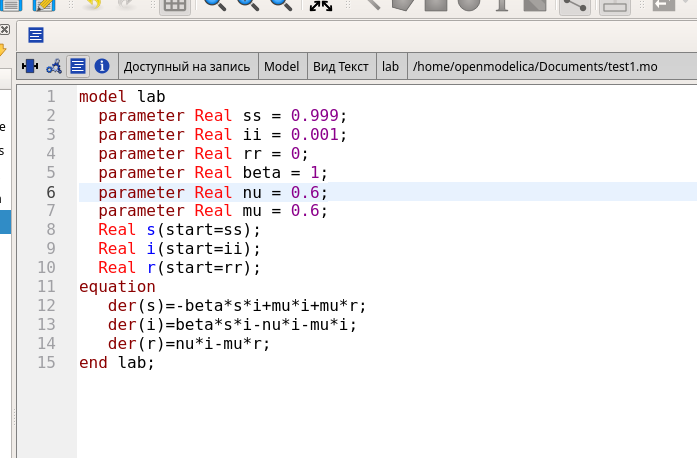


Рис. 24: Результаты на различных параметрах.

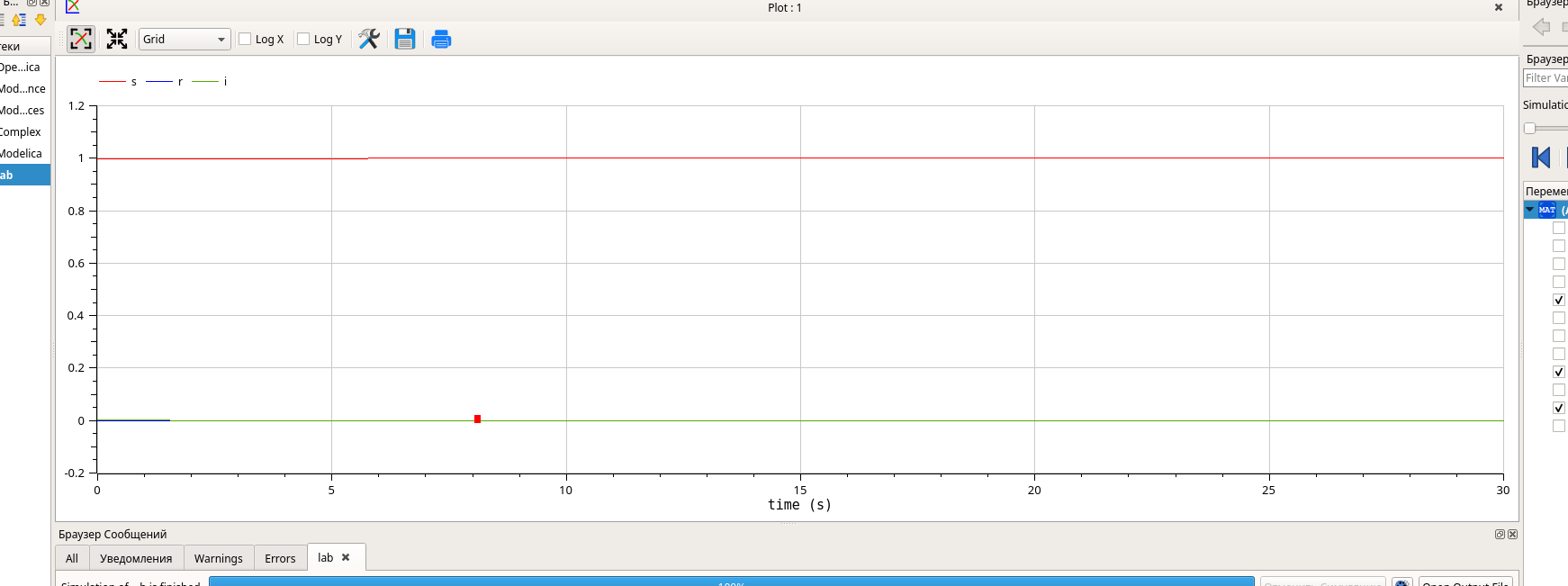


Рис. 25: Результаты на различных параметрах.

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