Отчёт по лабораторной работе №5

Дисциплина: Имитационное моделирование

Ганина Таисия Сергеевна, НФИбд-01-22

Содержание

# 1 Цель работы

Выполнить задания и получить практические навыки работы со средствами моделирования xcos, Modelica и OpenModelica. Рассмотреть модель эпидемии (SIR).

# 2 Задание

1. Реализовать имитационную модель эпидемии в xcos;
2. Реализовать имитационную модель эпидемии в Modelica;
3. Реализовать имитационную модель эпидемии в OpenModelica (упражнение);
4. 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).

Внутри каждой из выделенных групп особи считаются неразличимыми по свойствам. Типичная эволюция особи популяции описывается следующей диаграммой:

*S->I->R*

Считаем, что система замкнута, т.е. N=S+I+R. [1].

# 4 Выполнение лабораторной работы

## 4.1 Повторим пример из лабораторной работы

### 4.1.1 Реализация модели в xcos

Зайдя в среду моделирования Xcos начала выполнять учебный пример. В начале во вкладке “Моделирование” открыла “Установить контекст” и задала переменные , (рис. [1](#fig:001)).

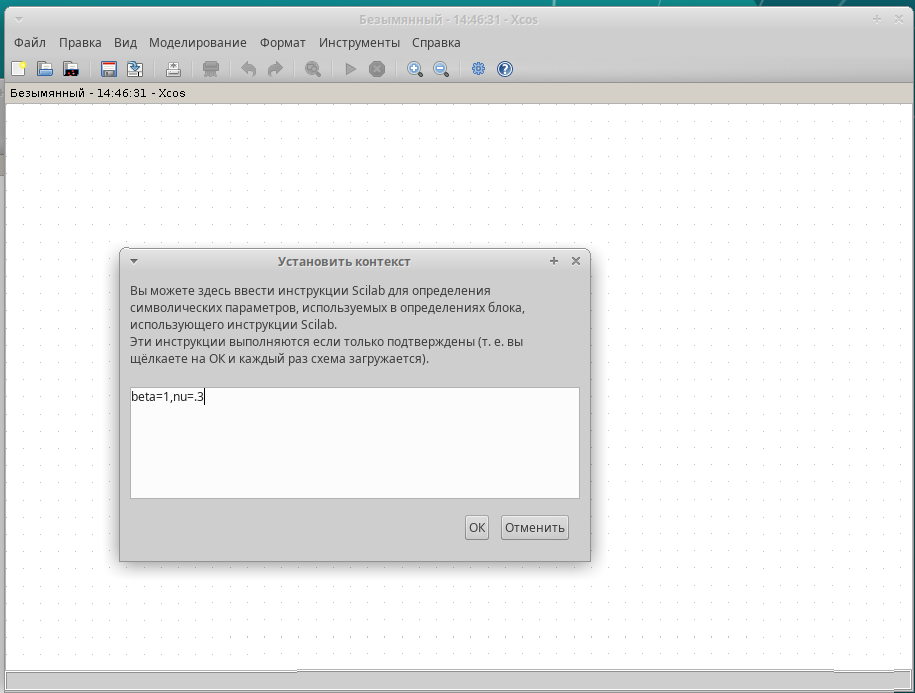


Figure 1: Установить контекст для учебного примера в Xcos

Далее я реализовала модель при помощи следующих блоков xcos(рис. [2](#fig:002)):

* 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 и ;
* SUMMATION — блок суммирования;
* PROD\_f — поэлементное произведение двух векторов на входе блока.

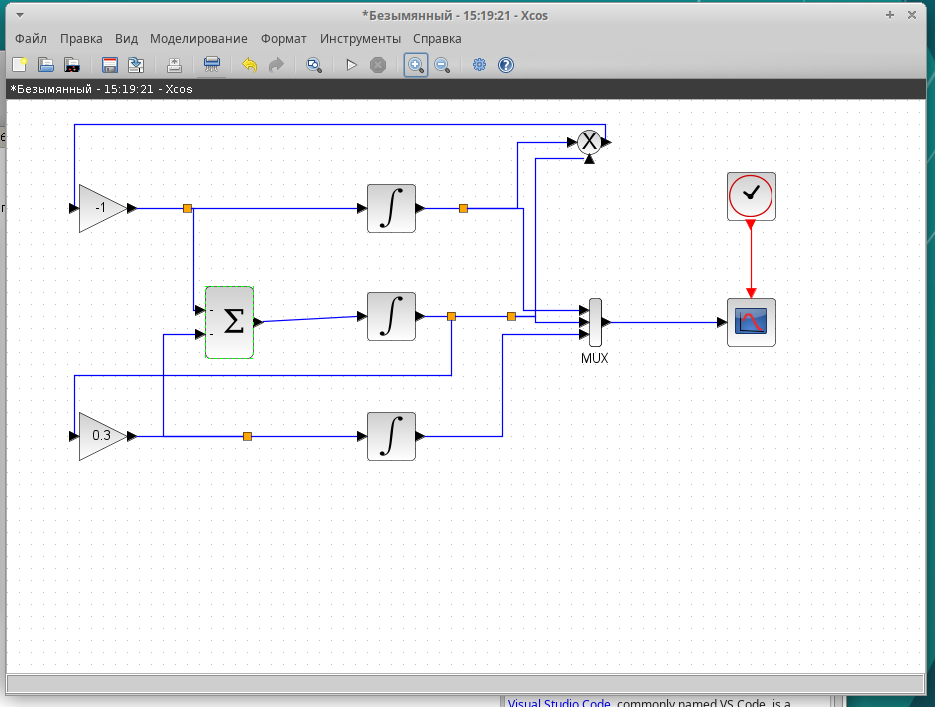


Figure 2: Модель SIR в xcos

Более подробно про вводимые мной значения (рис. [3](#fig:003), [4](#fig:004), [5](#fig:005), [6](#fig:006), [7](#fig:007), [8](#fig:008), [9](#fig:009), [10](#fig:010), [11](#fig:011), [12](#fig:012))

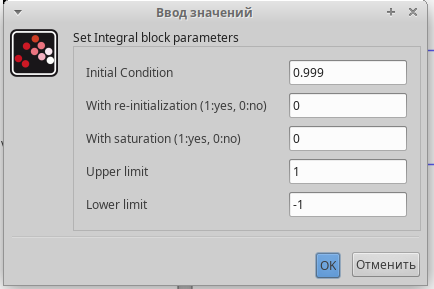


Figure 3: Задать начальные значения в верхнем блоке интегрирования

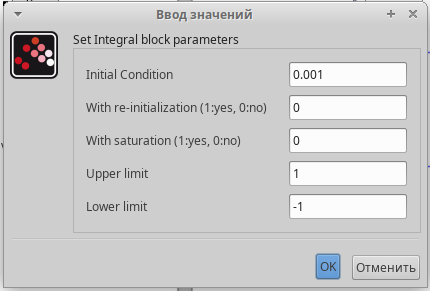


Figure 4: Задать начальные значения в среднем блоке интегрирования

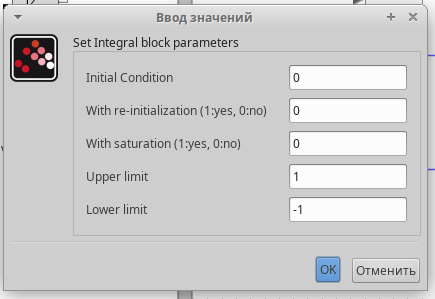


Figure 5: Задать начальные значения в нижнем блоке интегрирования

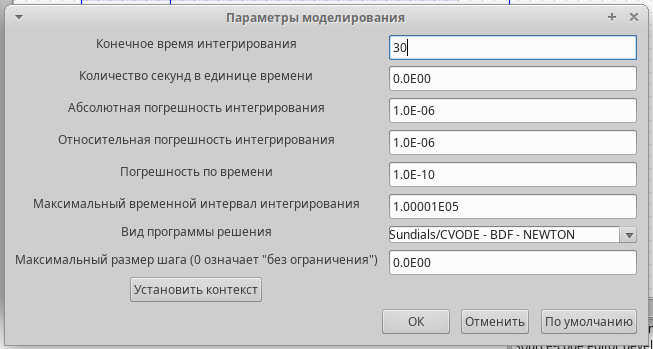


Figure 6: Задать конечное время интегрирования в xcos = 30

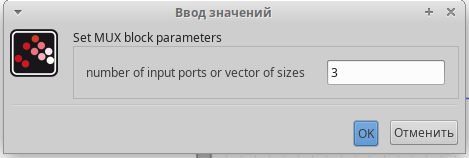


Figure 7: Параметры MUX

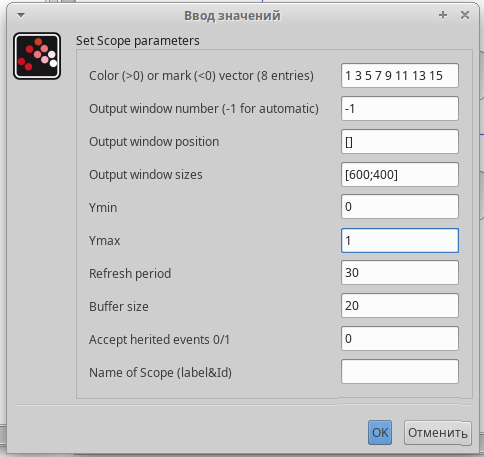


Figure 8: Редактирование параметров блока Scope

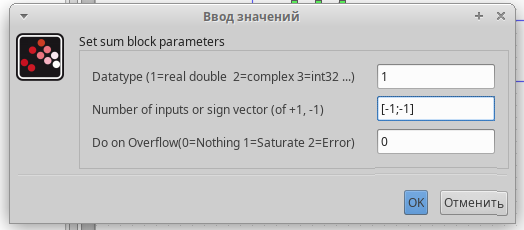


Figure 9: Задание параметров суммы

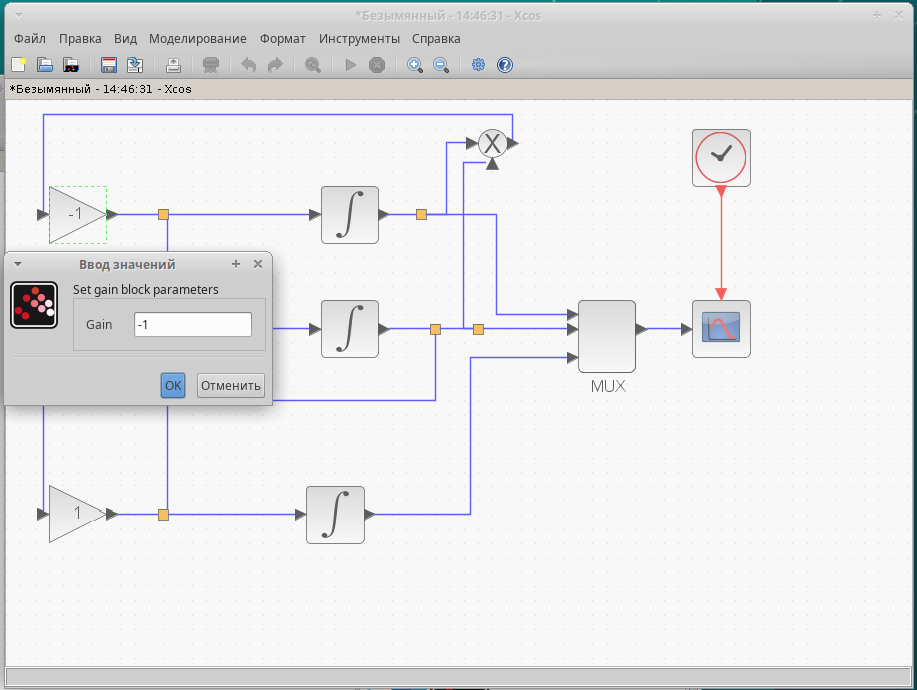


Figure 10: Задаю параметр = -1

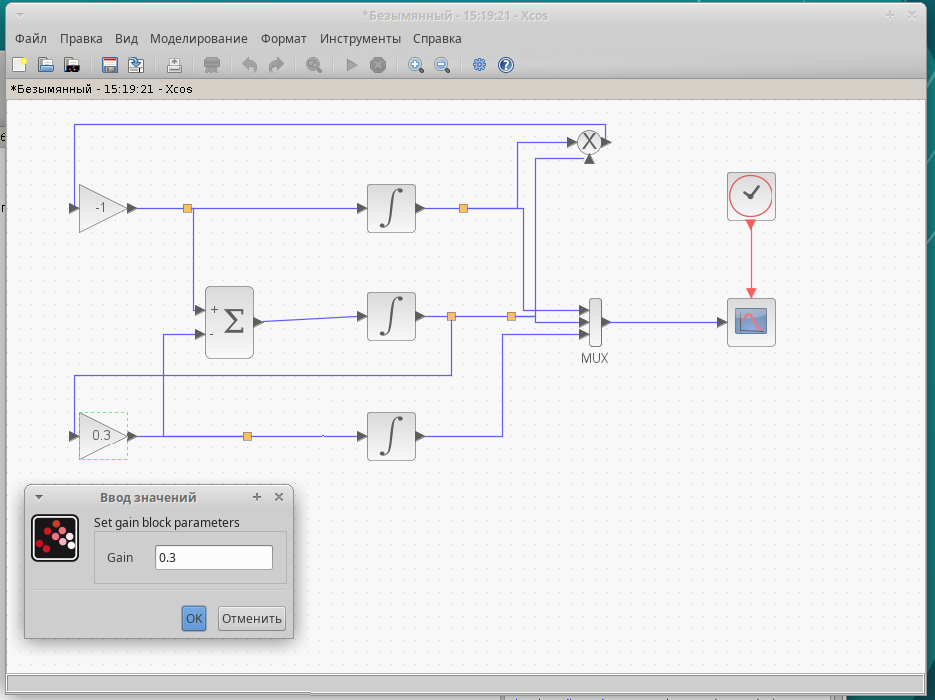


Figure 11: Задаю параметр = 0.3

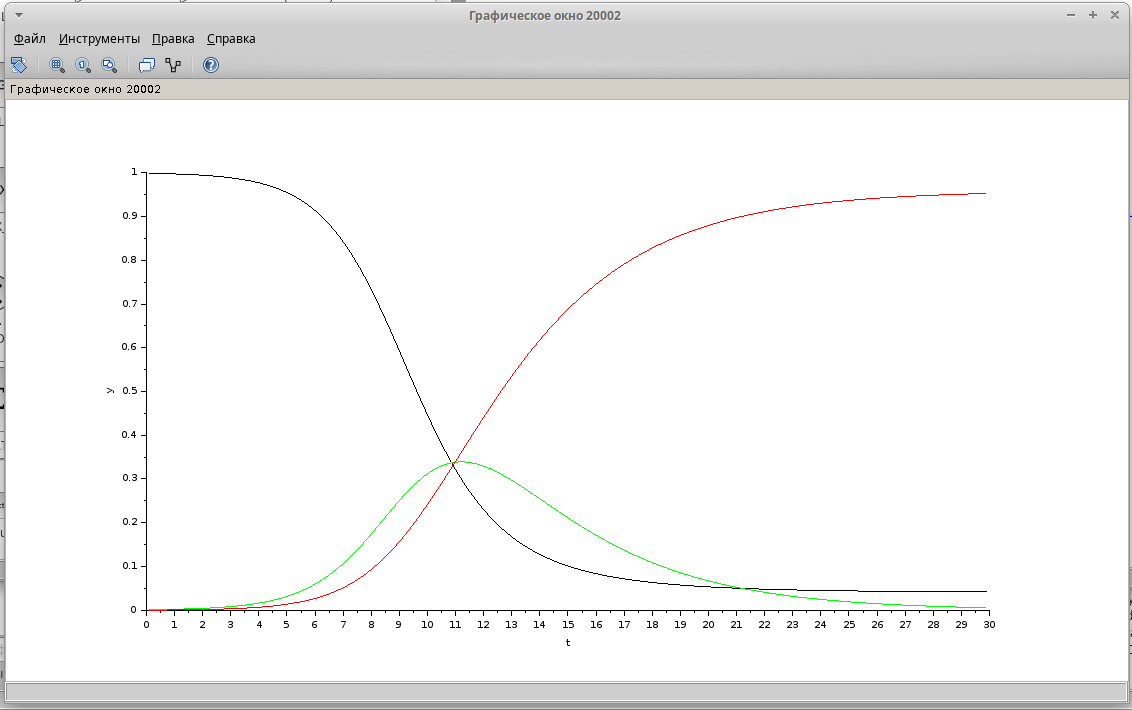


Figure 12: Эпидемический порог модели SIR, график построен в Xcos

### 4.1.2 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.

В начале задаем необходимые параметры такие же, как и в прошлый раз (рис. [1](#fig:001), [6](#fig:006)).

После этого я перешла к новым параметрам (рис. [13](#fig:013), [14](#fig:014)):

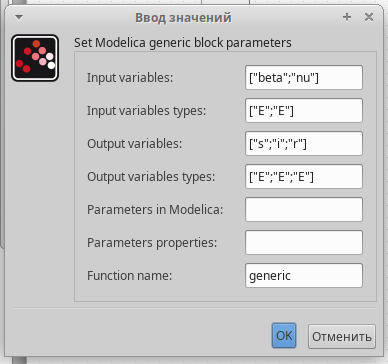


Figure 13: Параметры блока Modelica для модели. Ввод значений

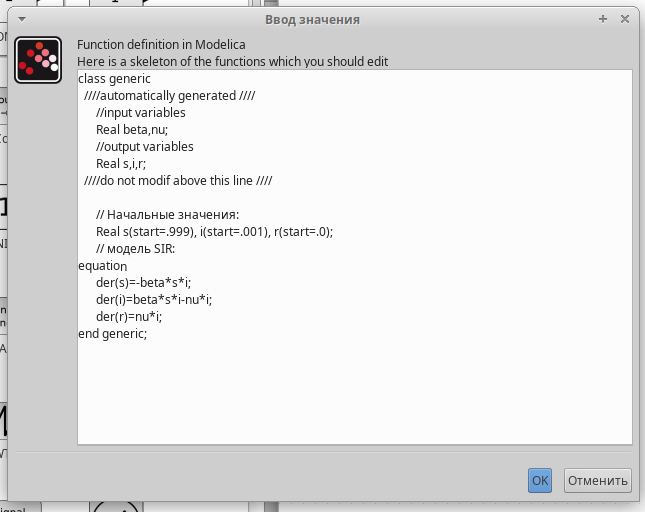


Figure 14: Параметры блока Modelica для модели. Ввод значений - функция

Пишу значения в блоки CONST\_m (рис. [15](#fig:015), [16](#fig:016)):

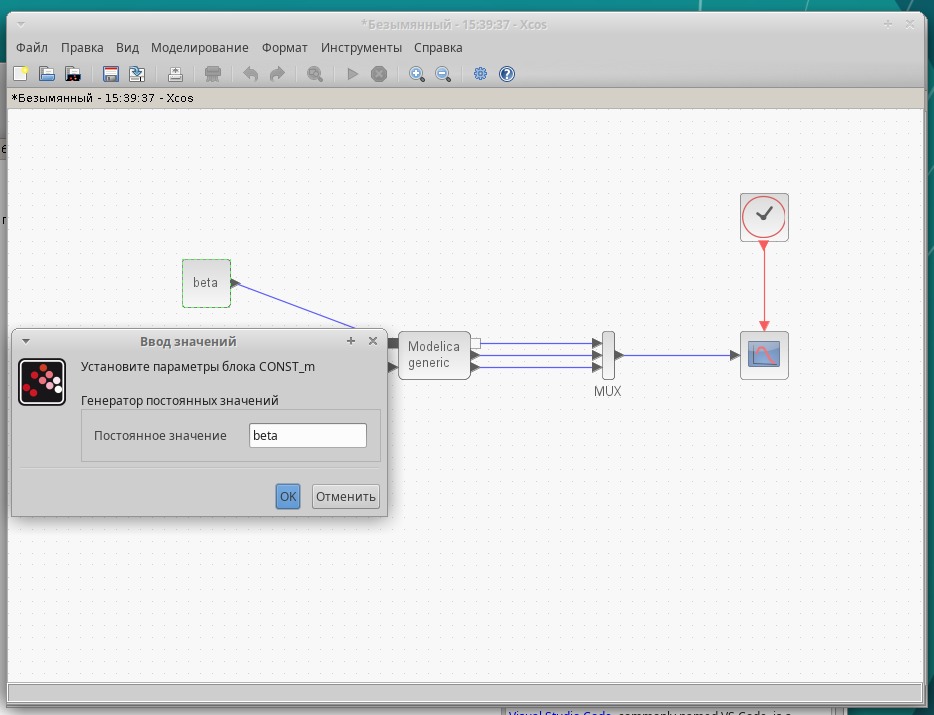


Figure 15: CONST\_m

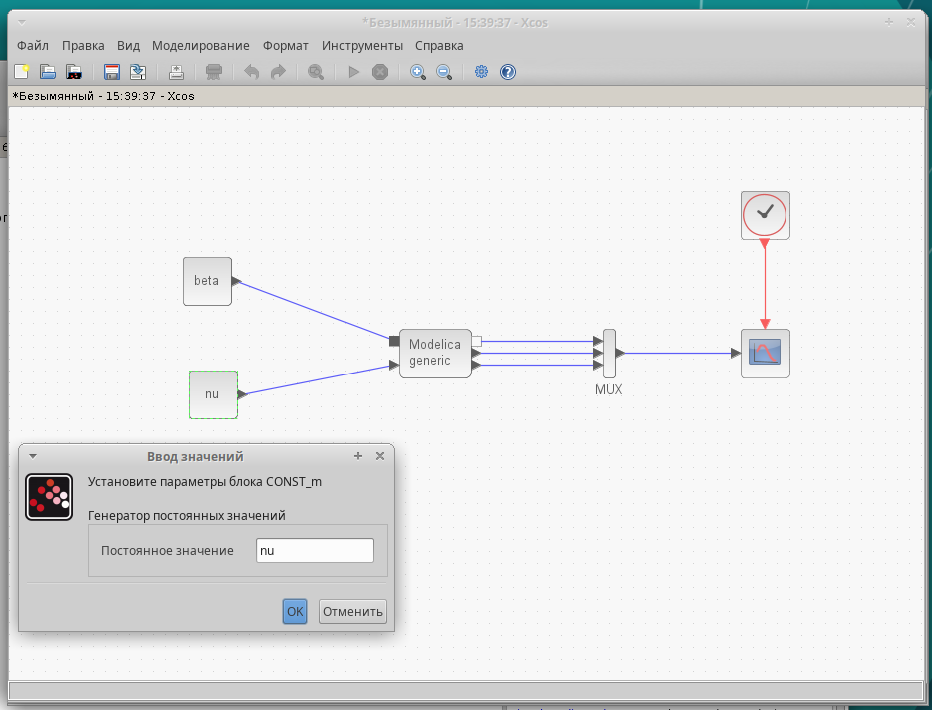


Figure 16: CONST\_m

Итоговая модель выглядит так (рис. [17](#fig:017), [18](#fig:018)):

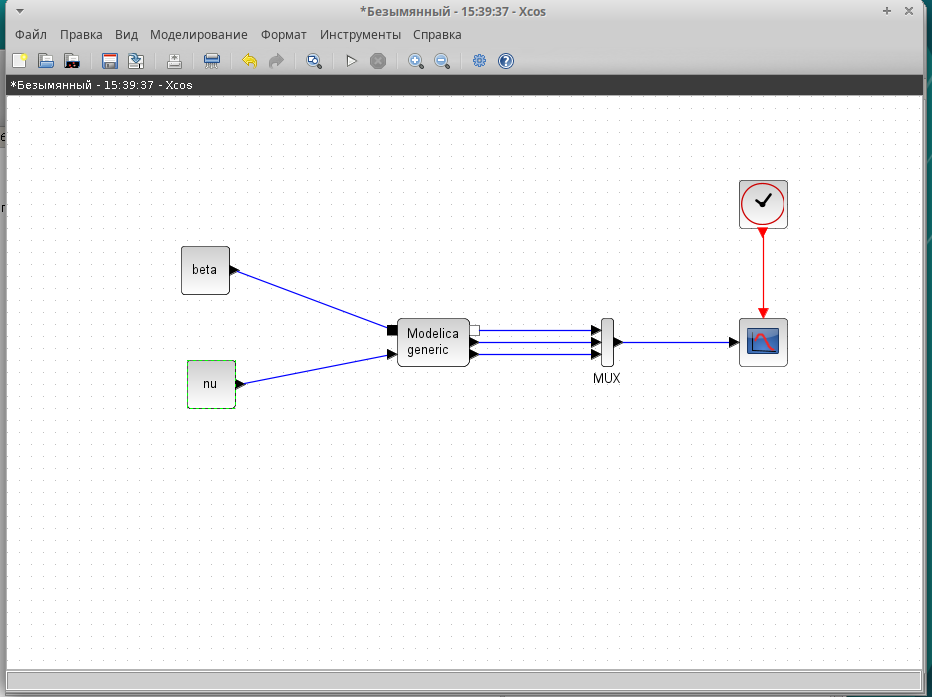


Figure 17: Модель с помощью блока Modelica в xcos

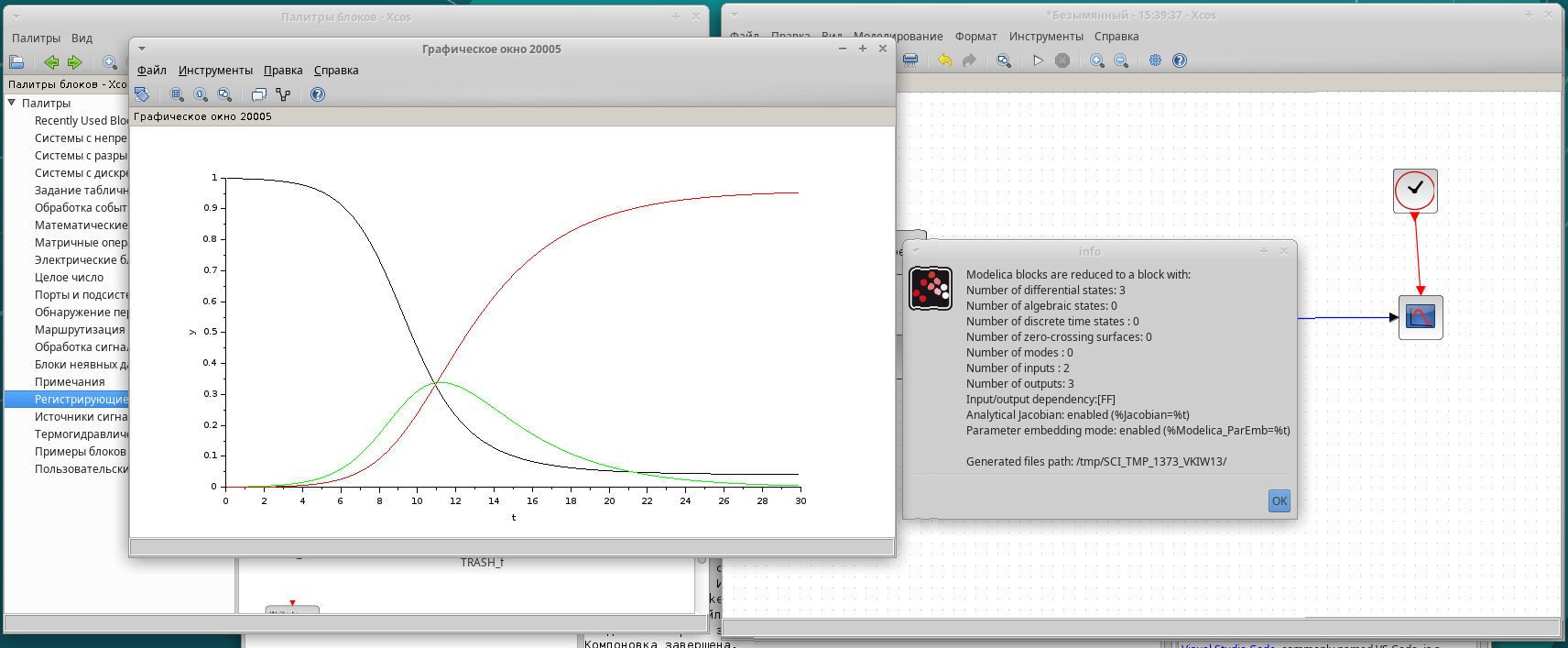


Figure 18: Эпидемический порог модели SIR, график построен с помощью блока Modelica в xcos

## 4.2 Упражнение. Реализация модели SIR в OpenModelica

Для начала я создала класс, а после написала код (рис. [19](#fig:019), [20](#fig:020)).

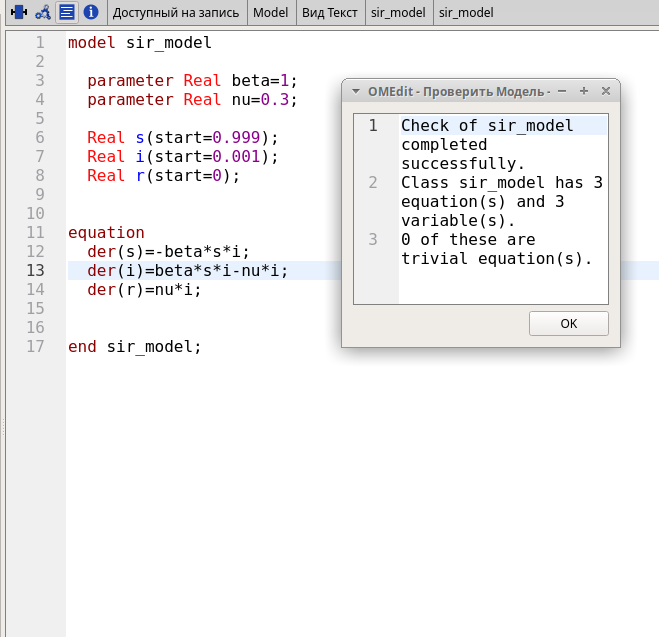


Figure 19: Код для модели SIR в OpenModelica

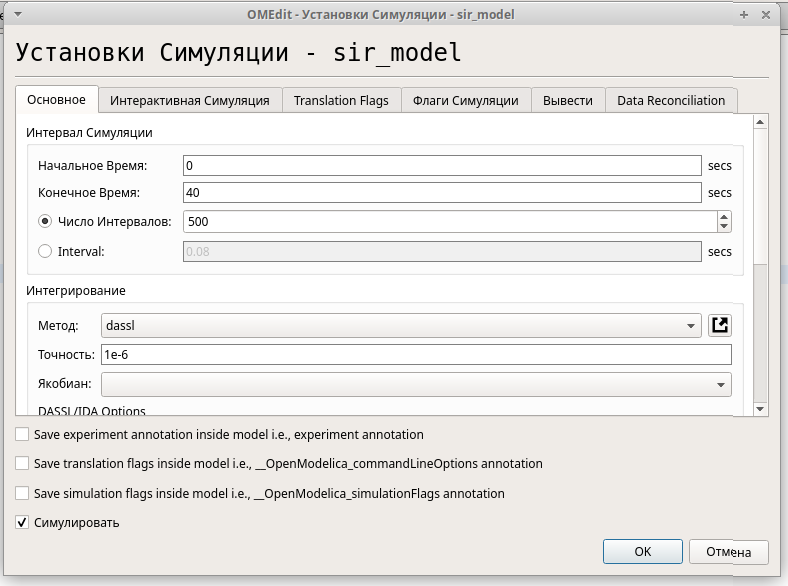


Figure 20: Установки симуляции (конечное время = 40)

Результат моделирования выглядит вот так (абсолютно такой же, что и предыдущие – значит, код верный) (рис. [21](#fig:021))

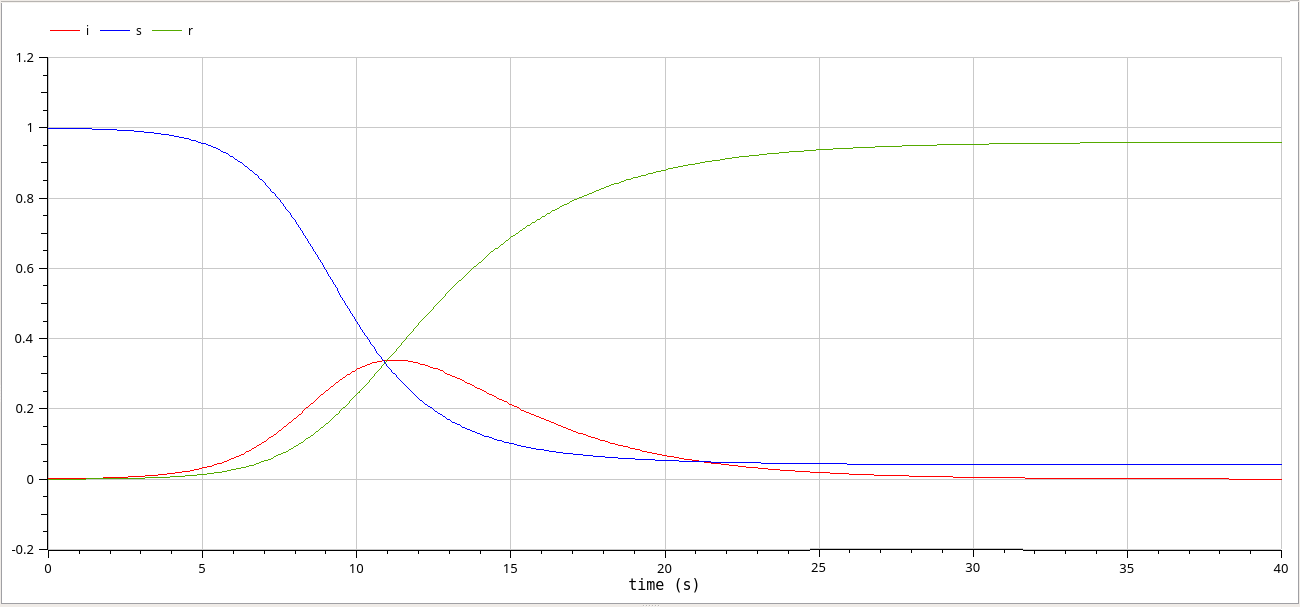


Figure 21: Эпидемический порог модели SIR, график построен в OpenModelica

## 4.3 Задание для самостоятельной работы

Предположим, что в модели SIR 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:

где — константа, которая равна коэффициенту смертности и рождаемости.

### 4.3.1 Реализация модели в xcos

По примеру уже выполненного задания строим модель (рис. [22](#fig:022), [23](#fig:023), [24](#fig:024), [25](#fig:025), [26](#fig:026), [27](#fig:027)).

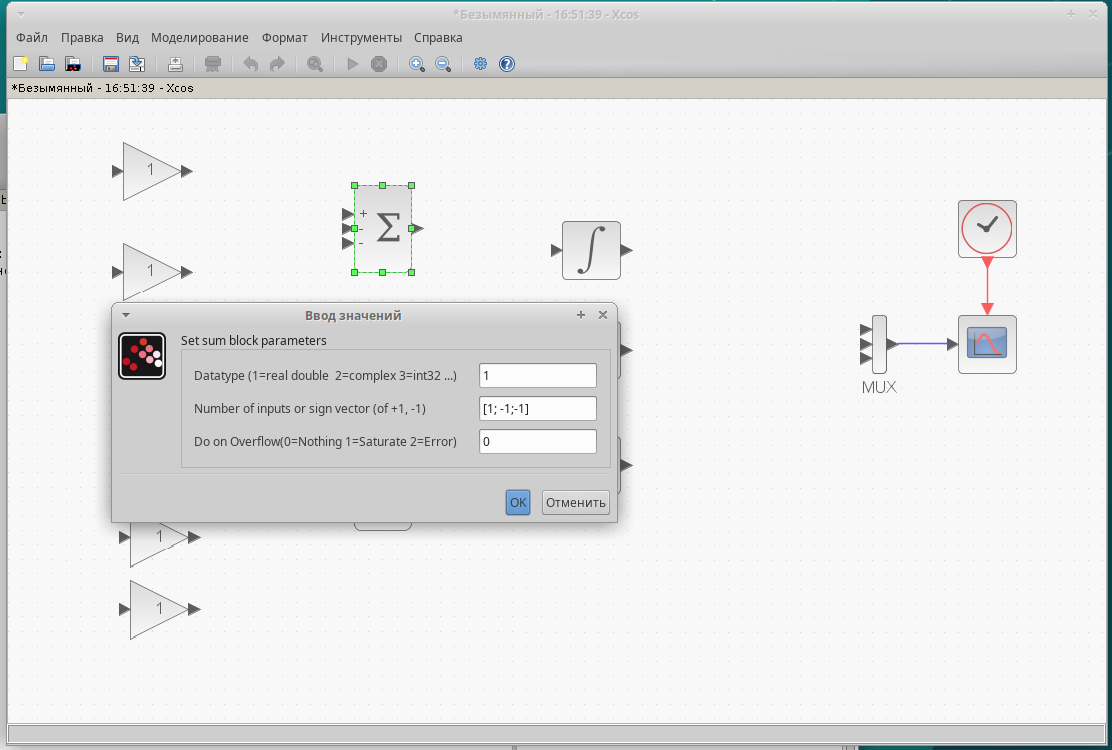


Figure 22: Задание параметров суммы (одной из сумм)

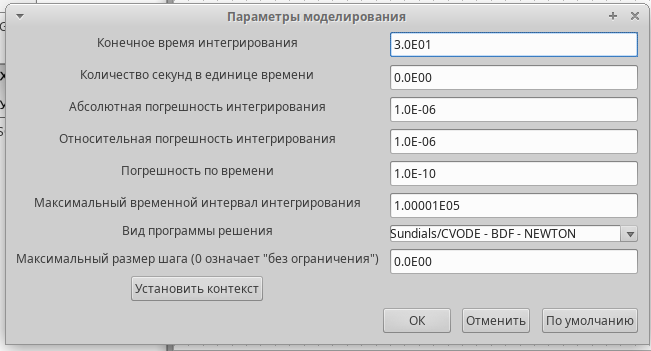


Figure 23: Задать конечное время интегрирования в xcos = 30

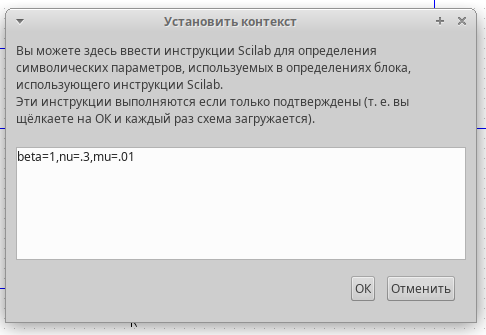


Figure 24: Установить контекст для самостоятельного примера в Xcos

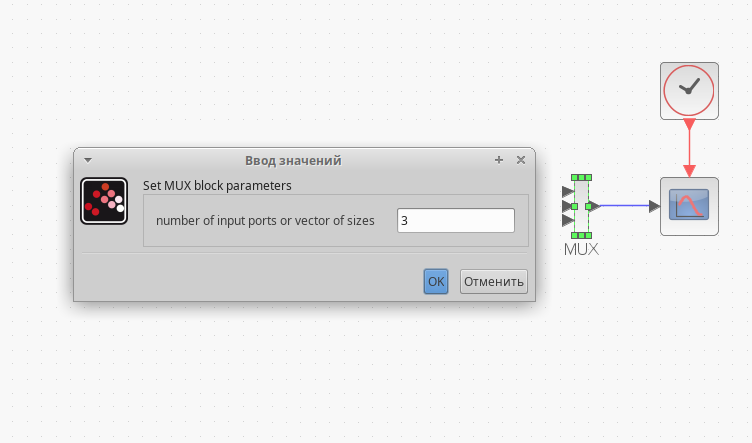


Figure 25: Параметры MUX

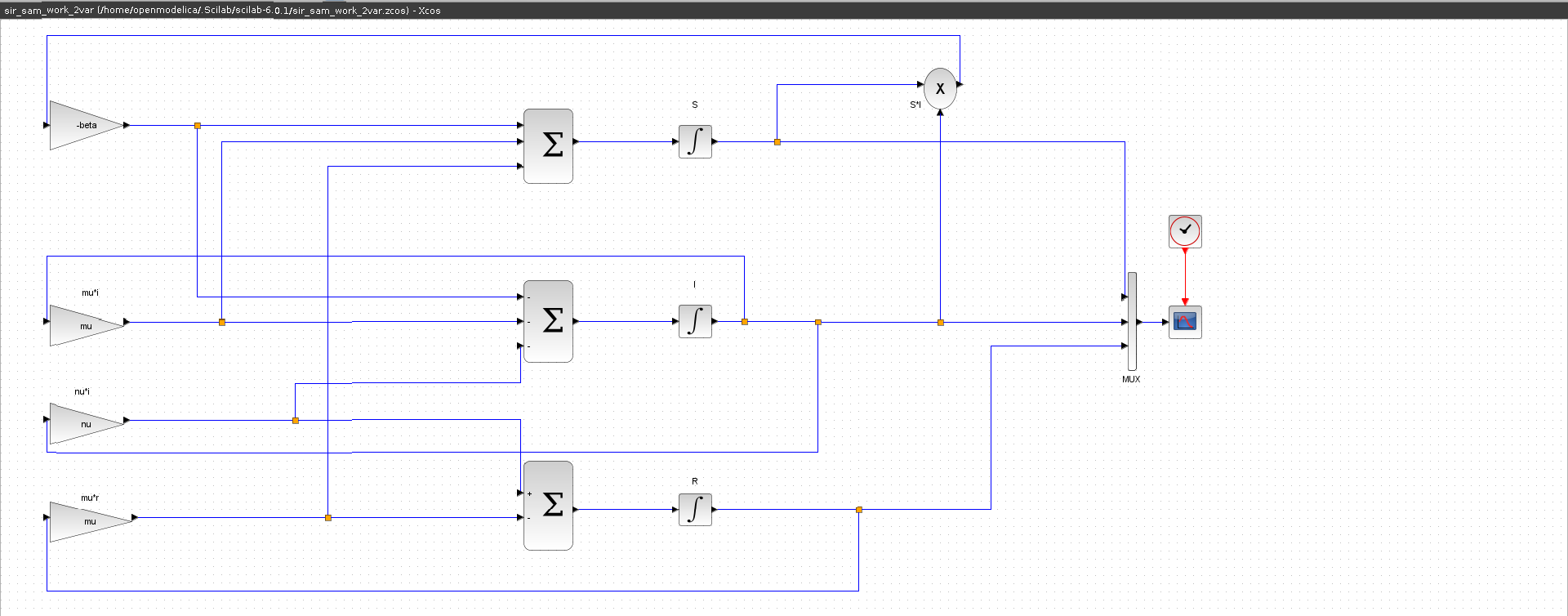


Figure 26: Модель SIR (с параметром ) в xcos

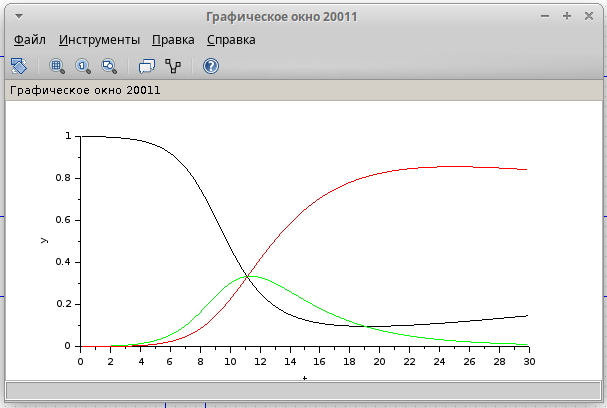


Figure 27: Эпидемический порог модели SIR (с параметром ), график построен в Xcos

### 4.3.2 Реализация модели с помощью блока Modelica в xcos (2 варианта)

В задании формула выглядит следующим образом:

Но в начале мы говорили, что , значит, можно вывести следующее:

Используем это для построения модели (рис. [28](#fig:028), [29](#fig:029), [30](#fig:030), [31](#fig:031), [32](#fig:032)):

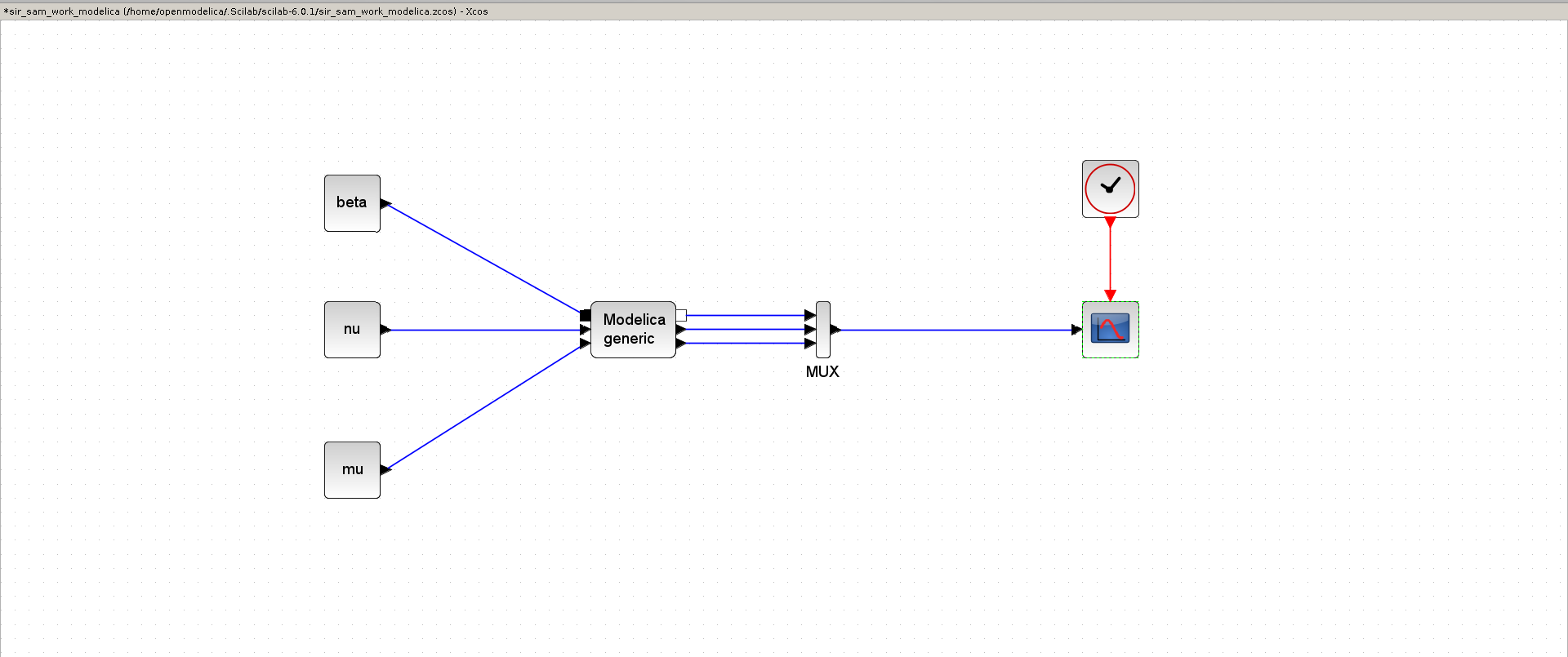


Figure 28: Модель с помощью блока Modelica (с параметром ) в xcos

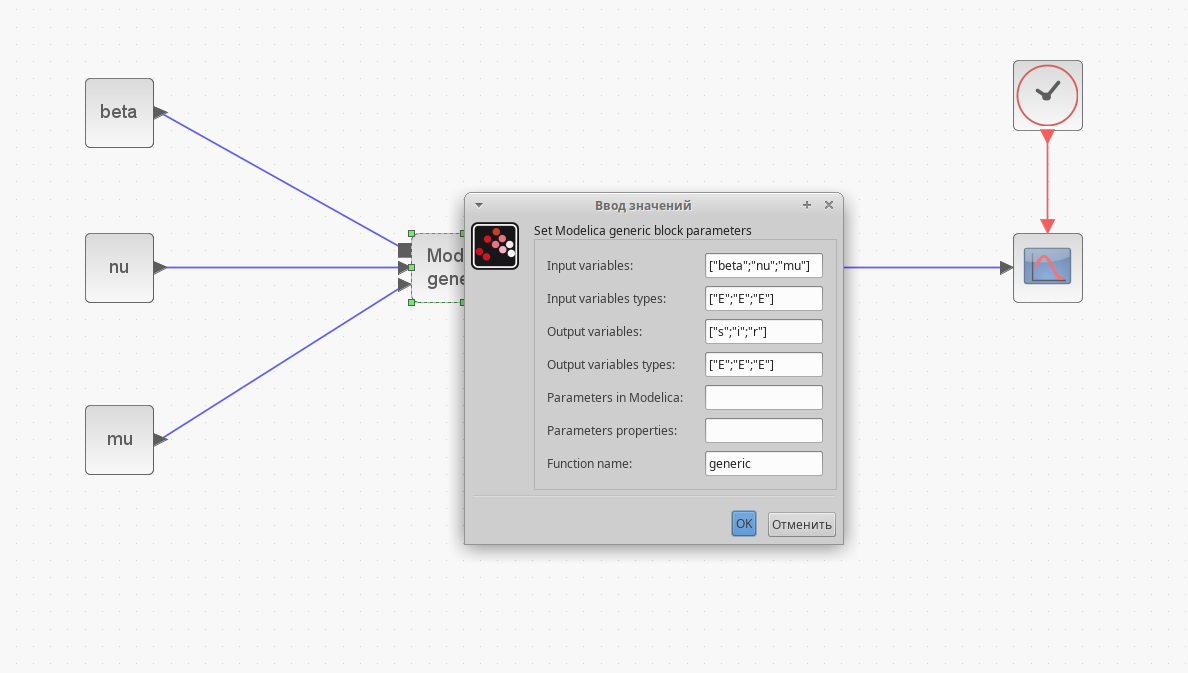


Figure 29: Параметры блока Modelica для модели (с параметром ). Ввод значений

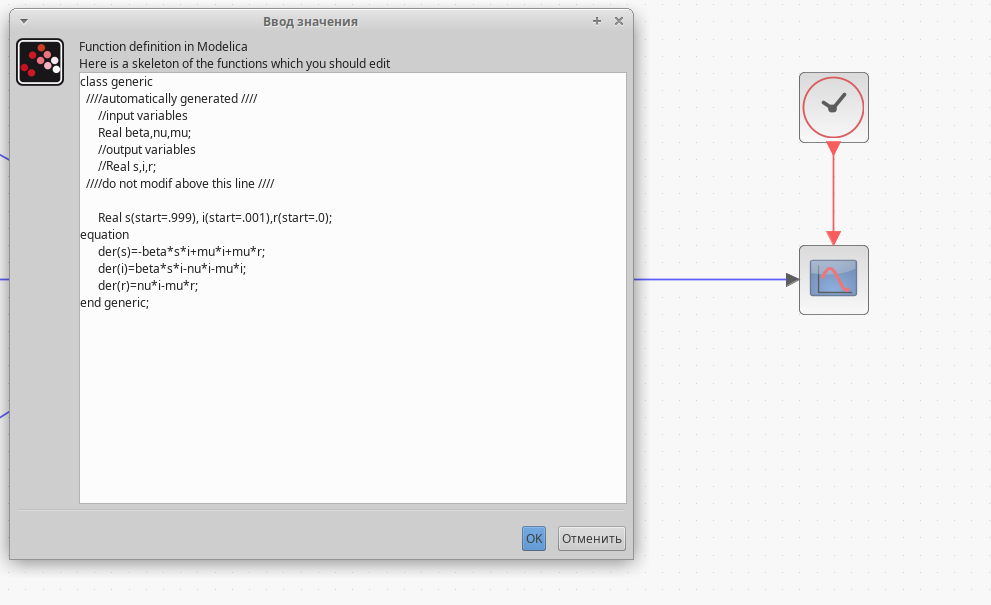


Figure 30: Параметры блока Modelica для модели (с параметром ). Ввод значений - функция

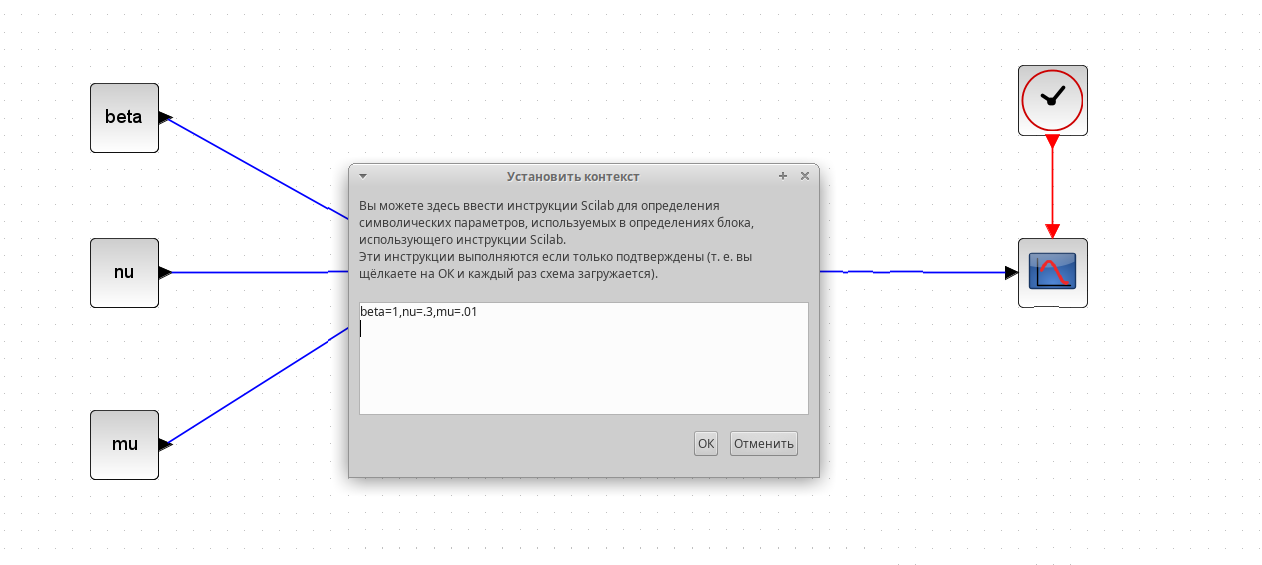


Figure 31: Установить контекст в Xcos (с параметром )

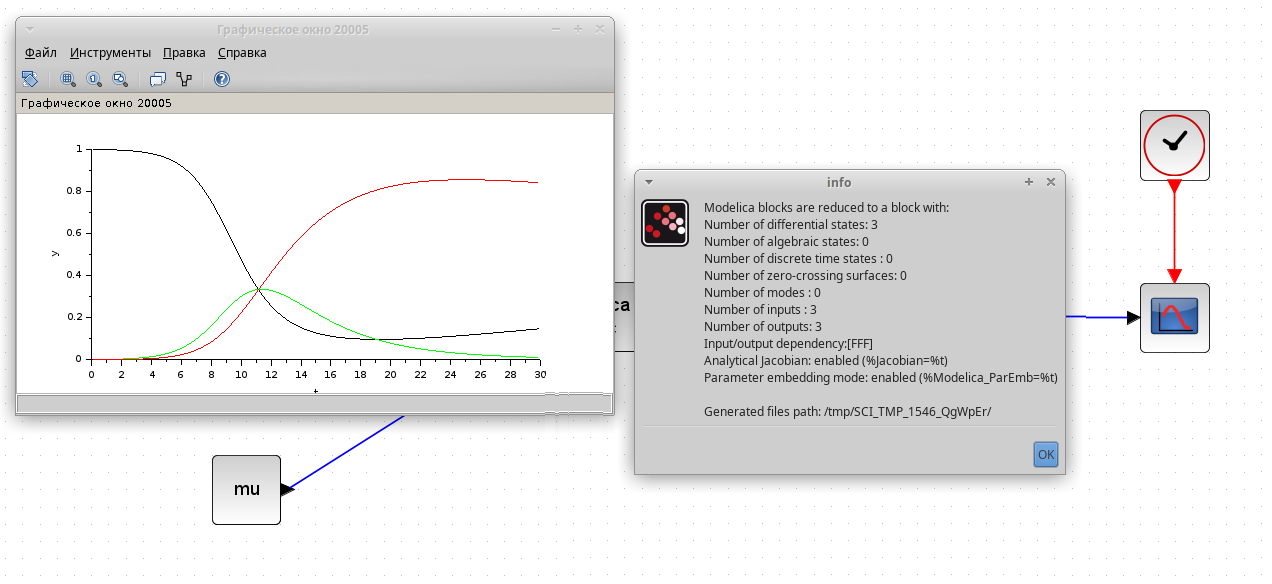


Figure 32: Эпидемический порог модели SIR (с параметром ), график построен с помощью блока Modelica в xcos

И давайте рассмотрим **второй вариант**, который реализуется без раскрытия скобок (рис. [33](#fig:033), [34](#fig:034), [35](#fig:035), [36](#fig:036), [37](#fig:037)).

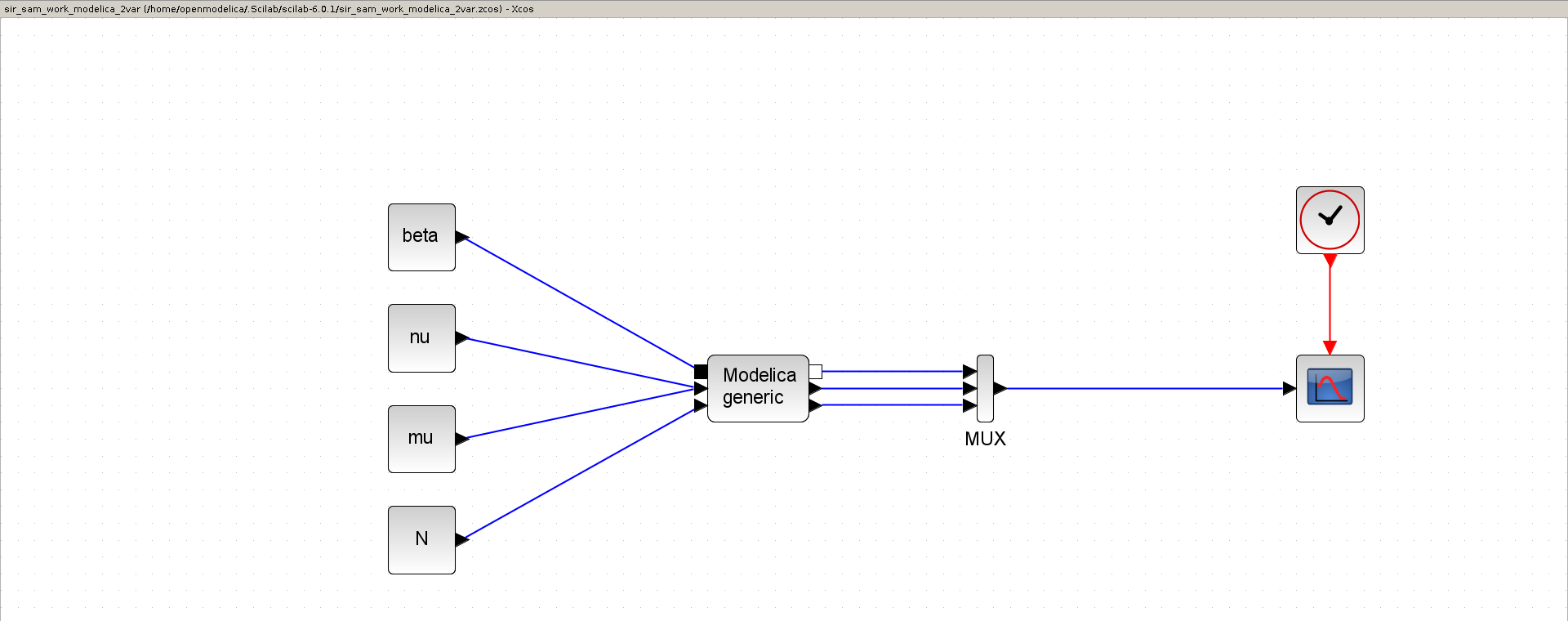


Figure 33: Модель с помощью блока Modelica (с параметром и N) в xcos

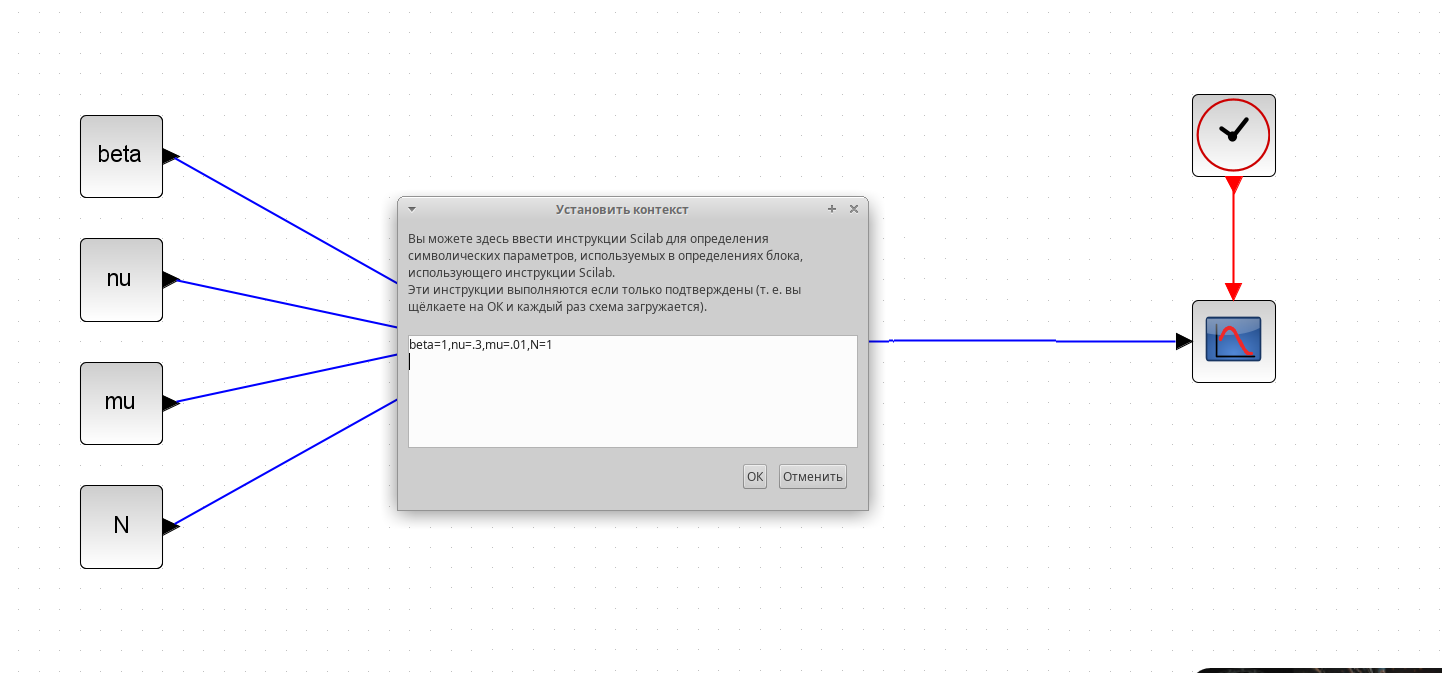


Figure 34: Установить контекст в Xcos (с параметром и N)

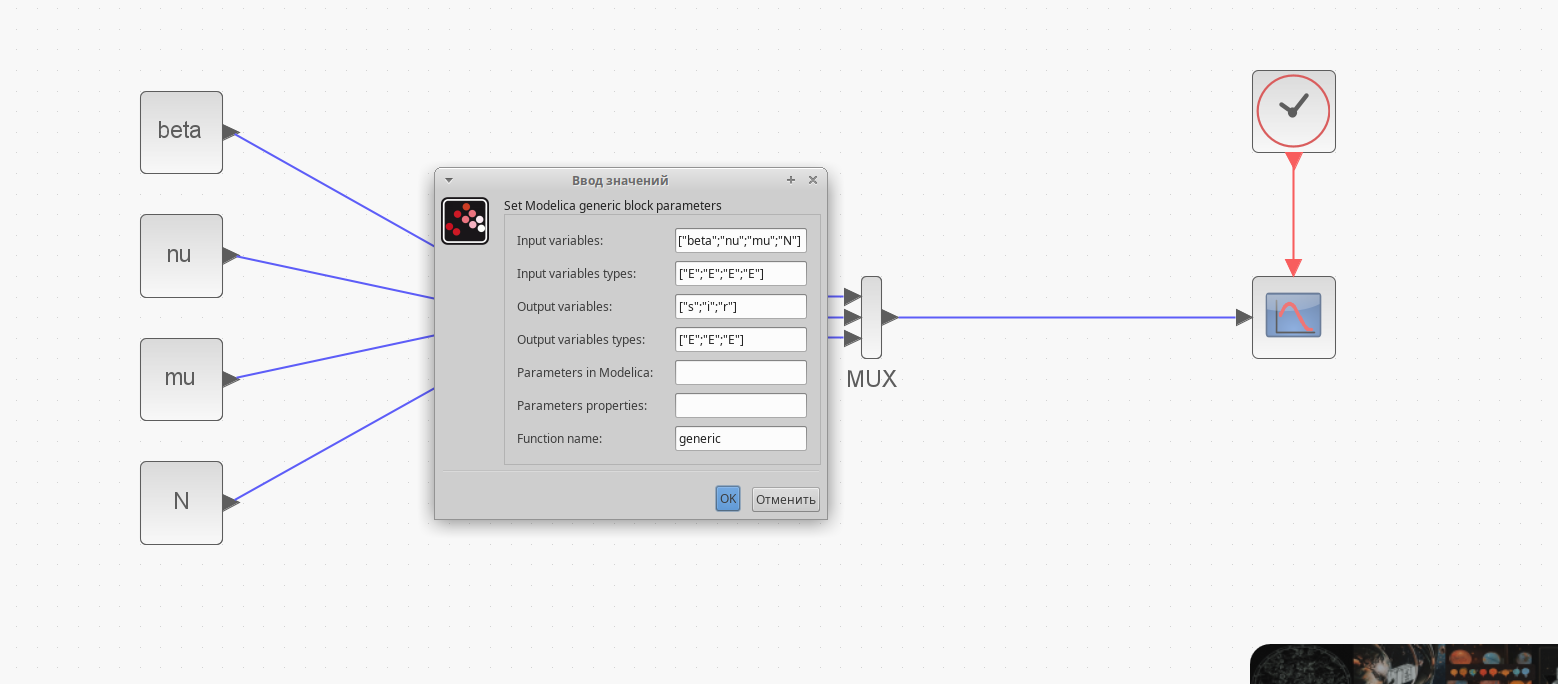


Figure 35: Параметры блока Modelica для модели (с параметром и N). Ввод значений

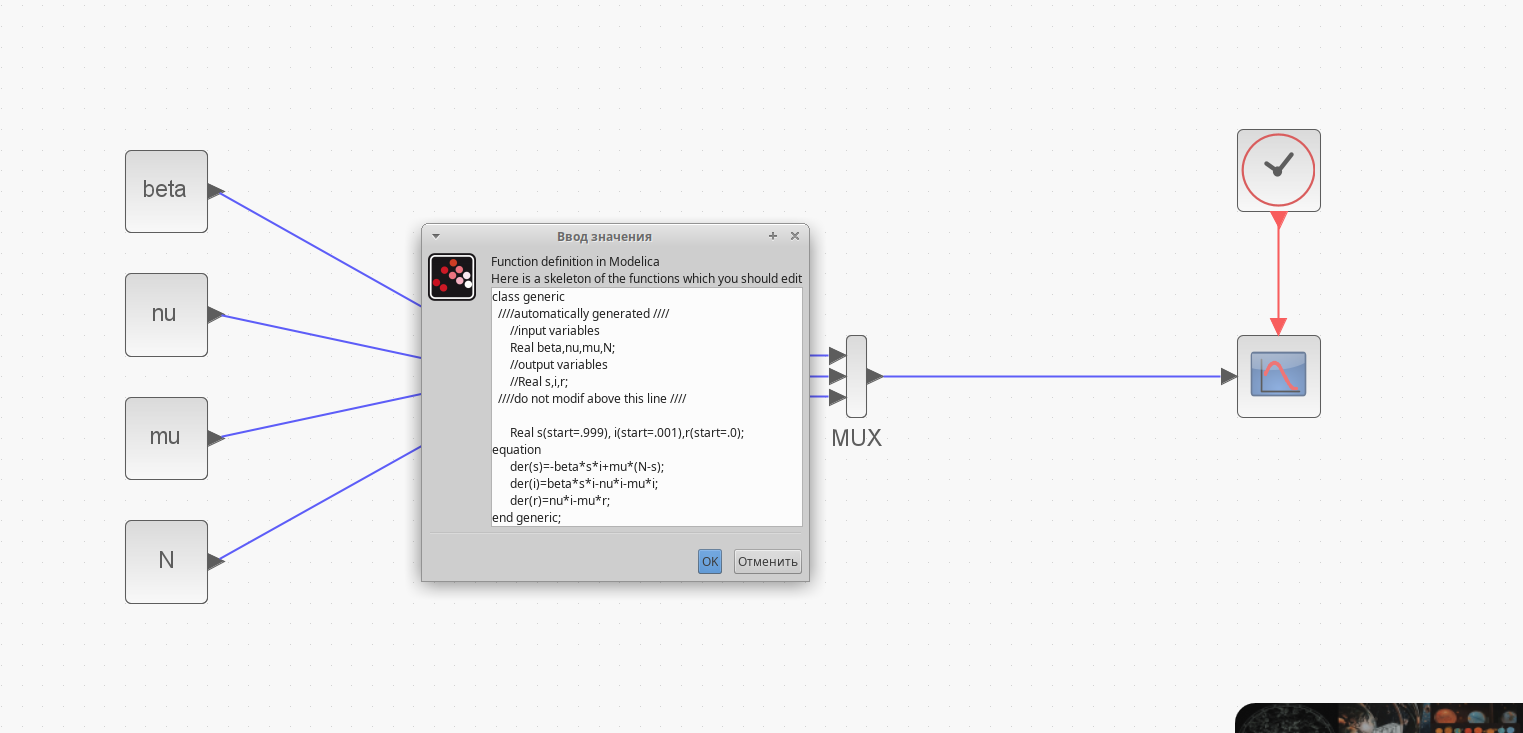


Figure 36: Параметры блока Modelica для модели (с параметром и N). Ввод значений - функция



Figure 37: Эпидемический порог модели SIR (с параметром и N), график построен с помощью блока Modelica в xcos

### 4.3.3 Реализация модели в OpenModelica

Реализуем модель SIR с учетом демографических процессов на OpenModelica.

Для начала я создала класс, а после написала код (рис. [38](#fig:038), [39](#fig:039)).

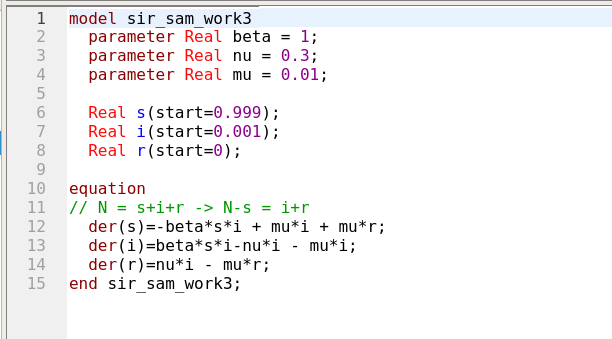


Figure 38: Код для модели SIR в OpenModelica (с параметром )

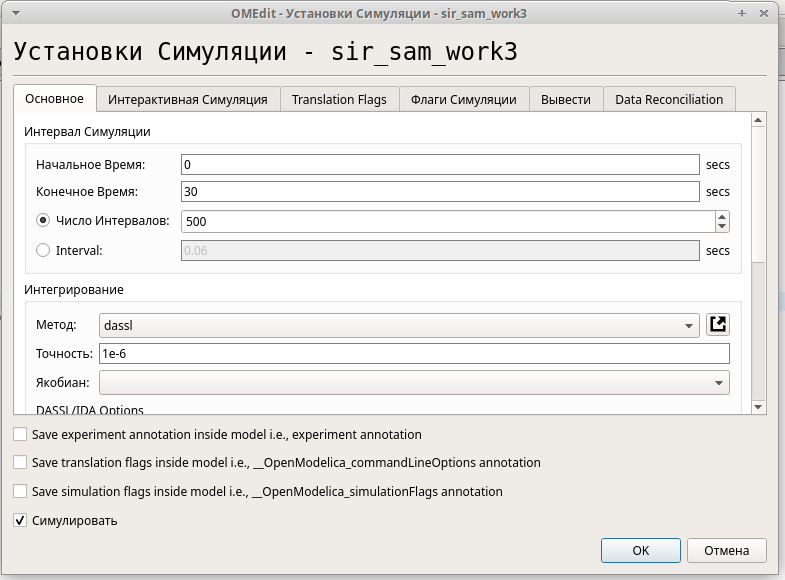


Figure 39: Установки симуляции (конечное время = 30)

Результат моделирования выглядит вот так (абсолютно такой же, что и предыдущие – значит, код верный) (рис. [40](#fig:040))

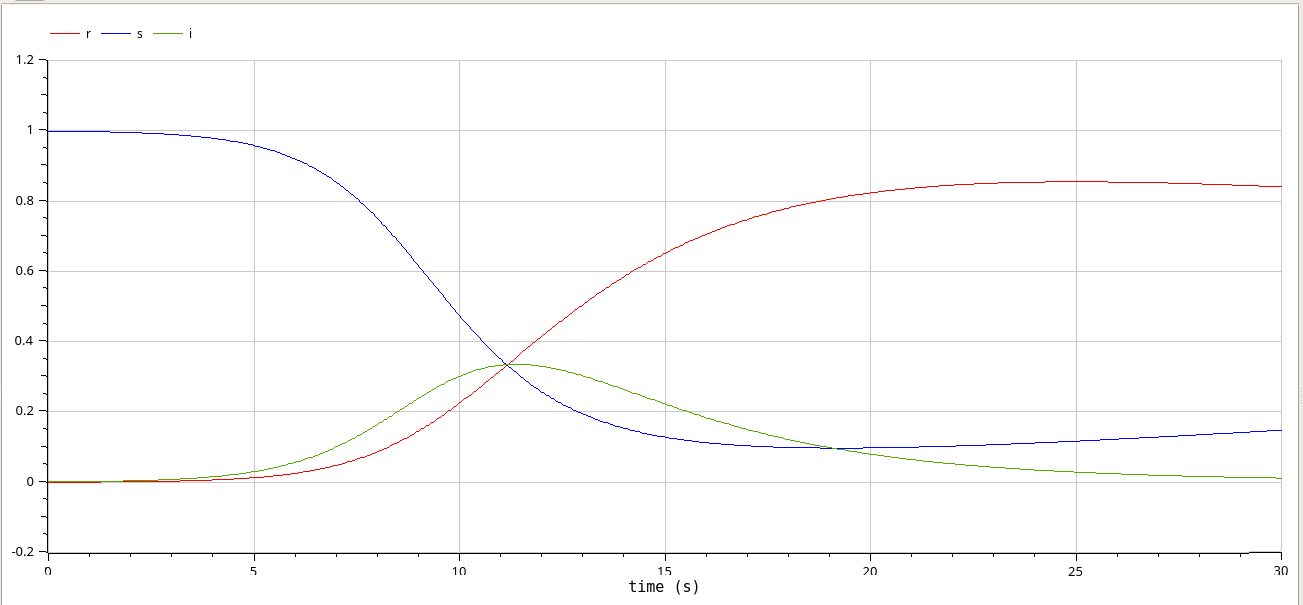


Figure 40: Эпидемический порог модели SIR (с параметром ), график построен в OpenModelica

И давайте рассмотрим **второй вариант**, который реализуется без раскрытия скобок (рис. [41](#fig:041), [42](#fig:042))

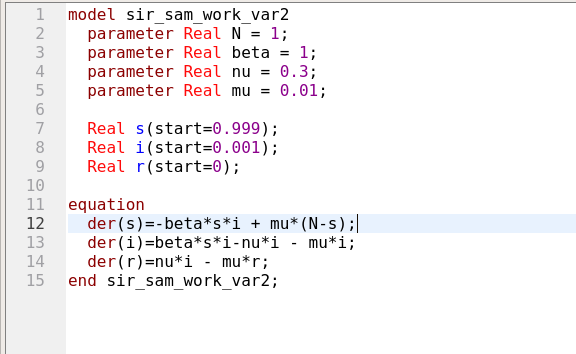


Figure 41: Код для модели SIR в OpenModelica (с параметром и N)

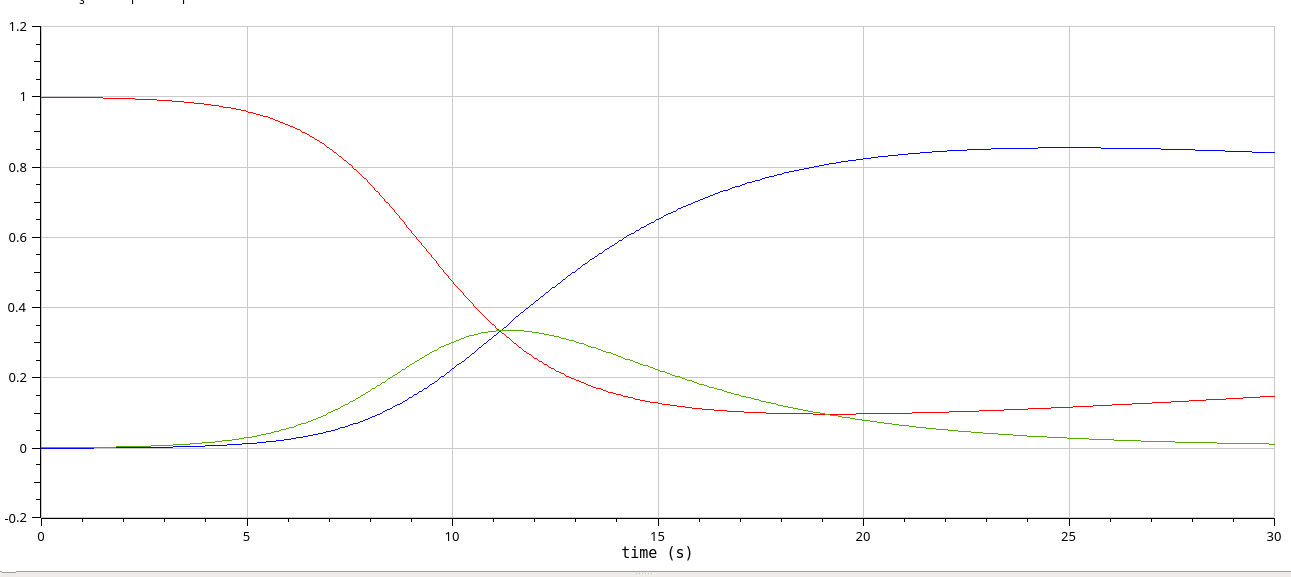


Figure 42: Эпидемический порог модели SIR (с параметром и N), график построен в OpenModelica

### 4.3.4 Графики эпидемического порога при различных значениях параметров модели

Далее я буду приводить примеры графиков с различными параматрами (рис. [43](#fig:043), [44](#fig:044), [45](#fig:045), [46](#fig:046), [47](#fig:047), [48](#fig:048))

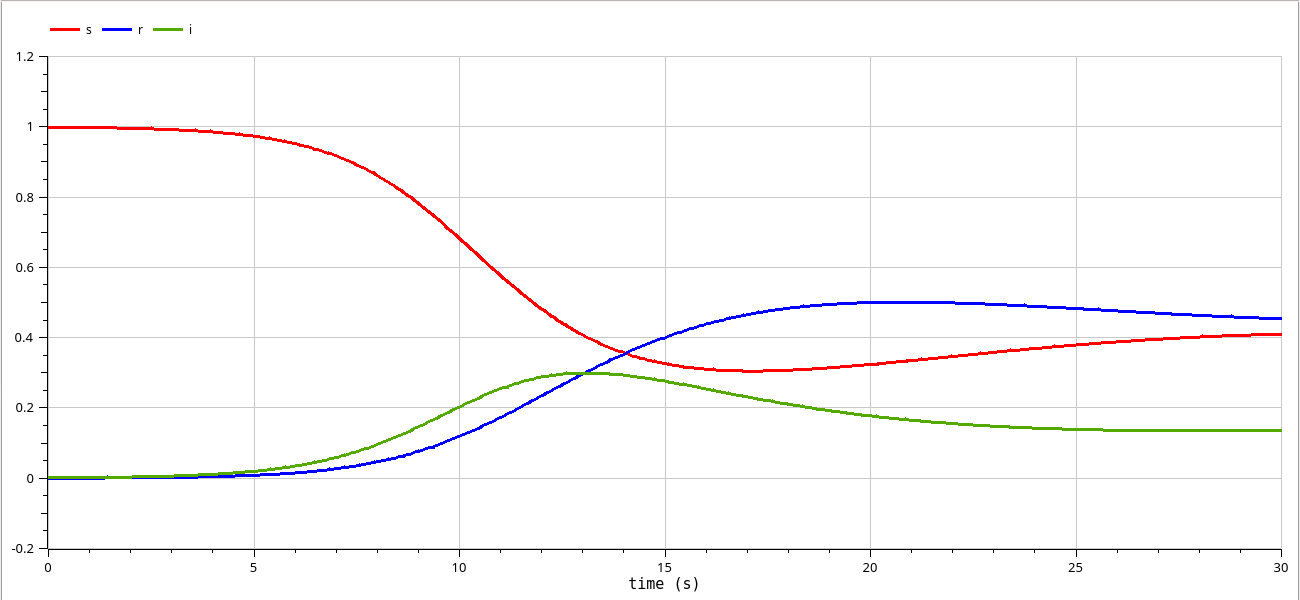


Figure 43: = 1, = 0.3, = 0.1

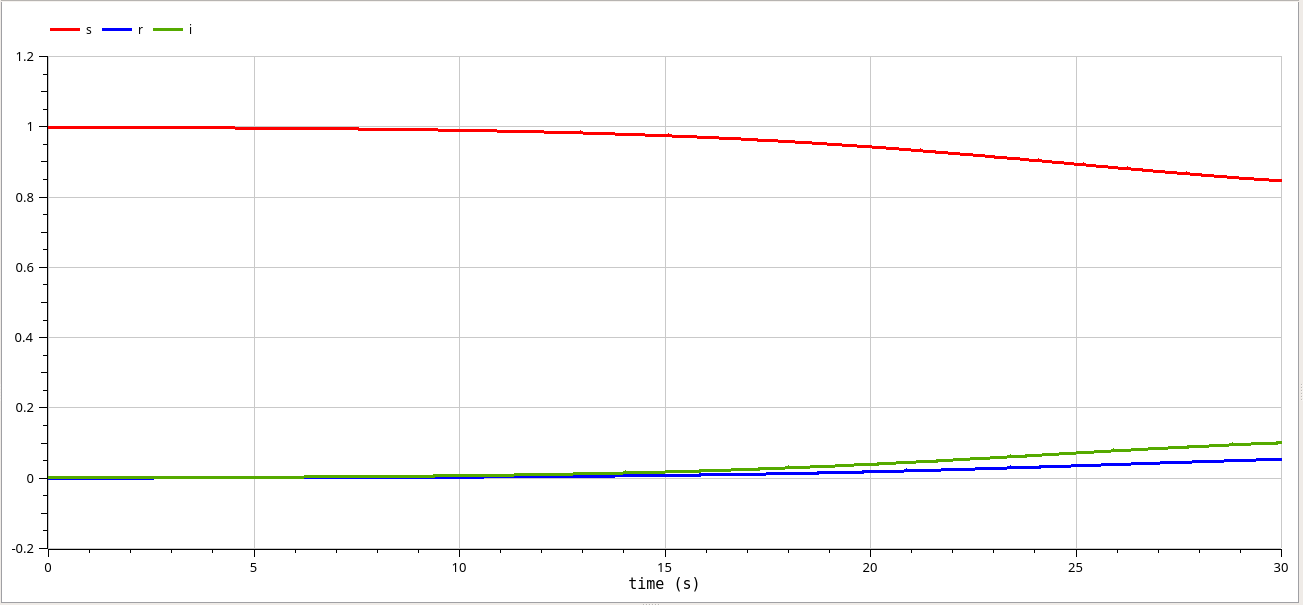


Figure 44: = 1, = 0.3, = 0.5

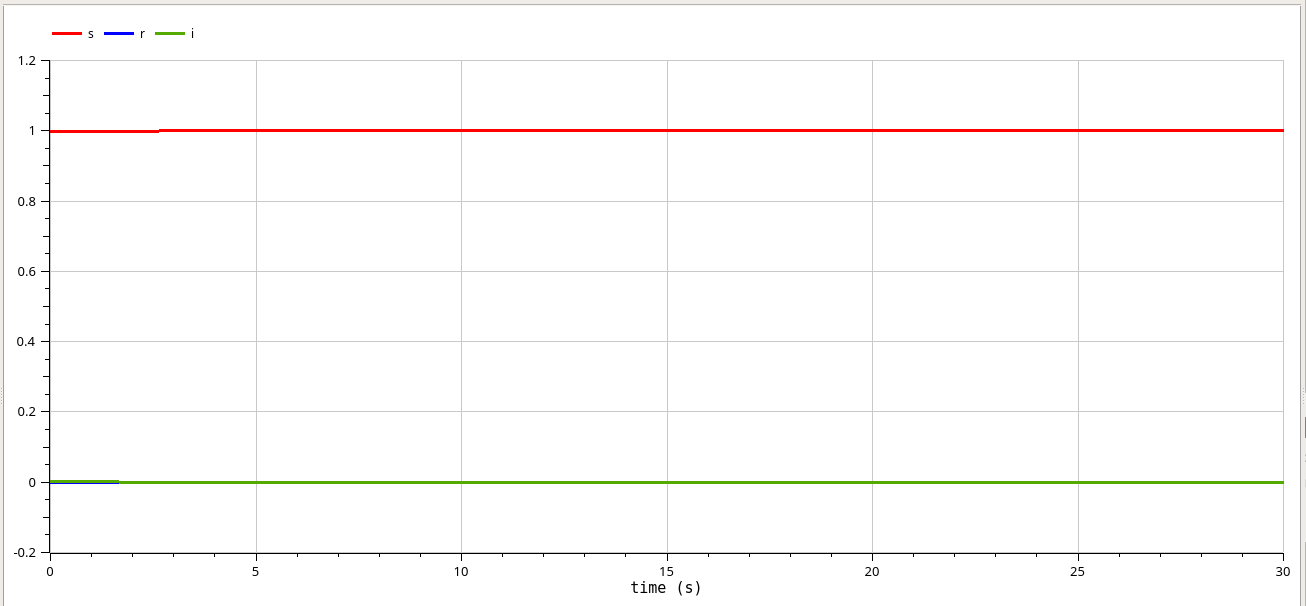


Figure 45: = 1, = 0.3, = 1

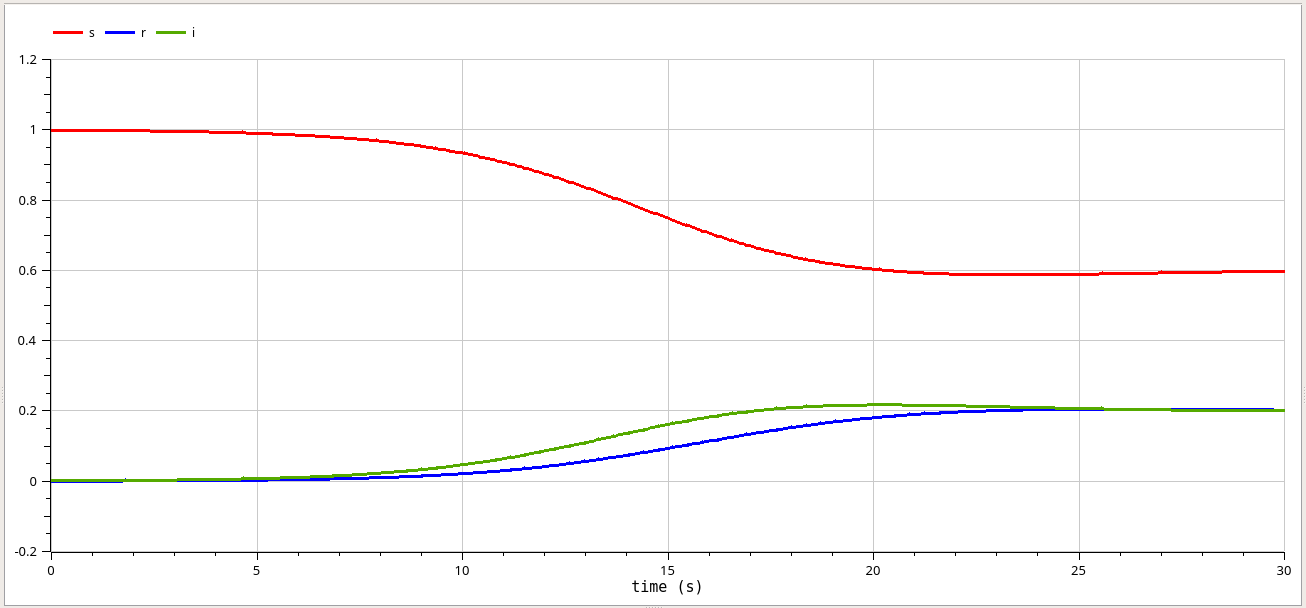


Figure 46: = 1, = 0.3, = 0.3

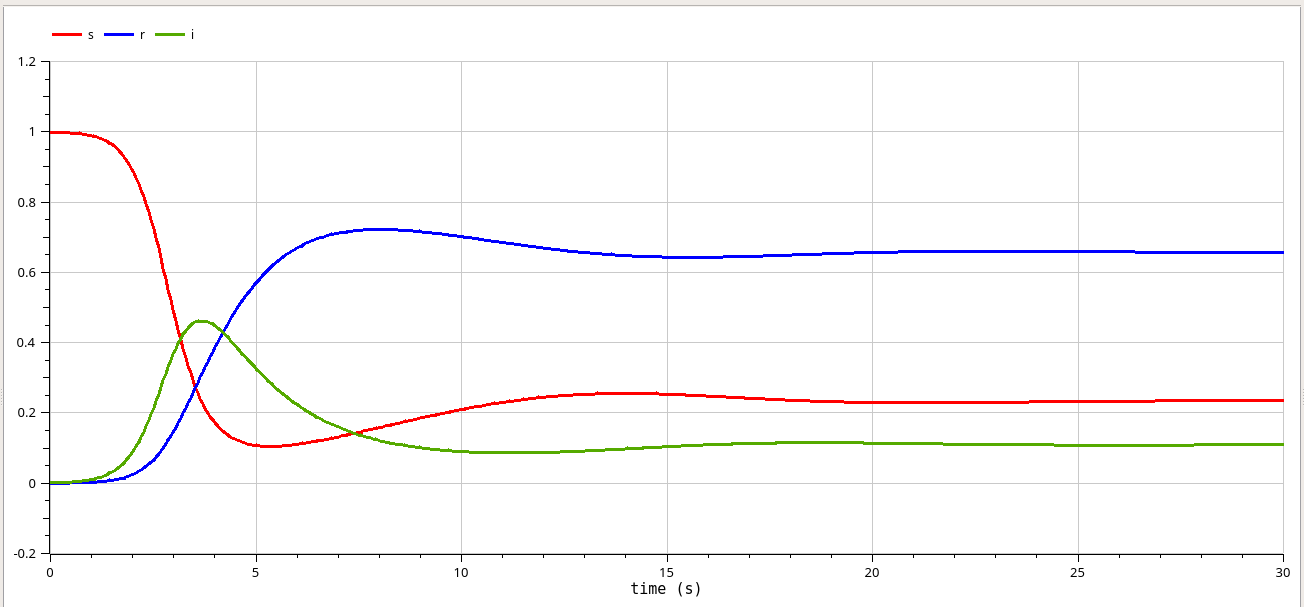


Figure 47: = 3, = 0.6, = 0.1

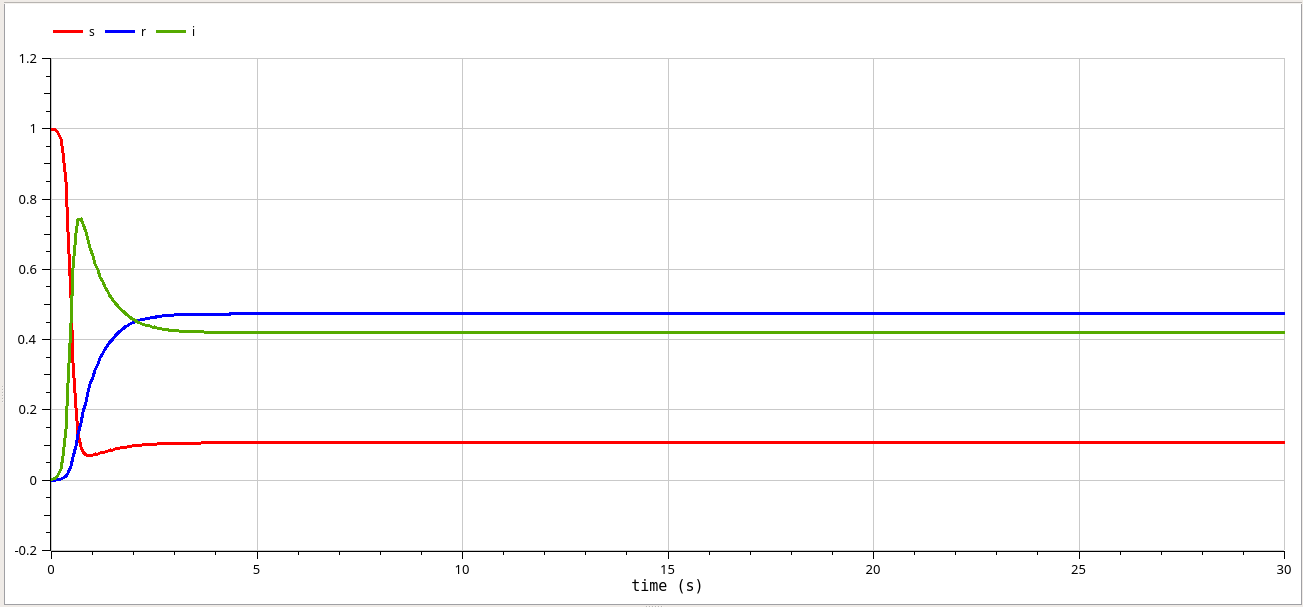


Figure 48: = 16, = 0.9, = 0.8

Приведу некоторые размышления по поводу графиков:

**Изменение (скорости заражения):**

Параметр (скорость заражения) оказывает существенное влияние на динамику эпидемии. Чем выше , тем быстрее распространяется болезнь и тем больше людей заражается.

* **= 3:** Высокая скорость заражения приводит к быстрому распространению эпидемии. Число инфицированных быстро растет, достигает пика, а затем снижается.
* **= 1:** Низкая скорость заражения приводит к медленному или незначительному распространению эпидемии. Число инфицированных остается низким или постепенно увеличивается.

**Изменение (скорости выздоровления):**

Более высокая скорость выздоровления способствует быстрому снижению числа инфицированных после пика.

* **Изменение (коэффициента смертности и рождаемости):**

В самых базовых моделях SIR (Susceptible - восприимчивые, Infected - инфицированные, Recovered - выздоровевшие), которые часто используются для начального анализа, рождаемость и смертность (и, следовательно, параметр ) обычно не учитываются. Это делается для упрощения модели и сосредоточения внимания на динамике распространения инфекции. В таких моделях население считается постоянным.

* **= 1:** Высокий коэффициент приводит к быстрому обороту популяции, что предотвращает распространение эпидемии.
* **= 0.5:** Средний коэффициент позволяет эпидемии распространяться медленно.
* **= 0.1:** Низкий коэффициент позволяет эпидемии распространяться быстрее, но в совокупности с изменениями других коэффициентов быстрый рост сопровождается и быстрым ростом “переболевших”.

**= 3, = 0.6, = 0.1**

* Начальная популяция восприимчивых быстро уменьшается, поскольку болезнь распространяется.
* Число инфицированных быстро растет, достигая пика, а затем постепенно снижается.
* Число выздоровевших увеличивается и стабилизируется на определенном уровне.
* Система достигает состояния равновесия, где популяция восприимчивых стабилизируется на уровне около 0.25, инфицированных - на уровне около 0.1, а выздоровевших - на уровне около 0.65.

**= 1, = 0.3, = 1**

* Число восприимчивых остается неизменным.
* Число инфицированных остается очень низким.
* Число выздоровевших также остается неизменным.
* Болезнь не может распространиться.

**= 1, = 0.3, = 0.5**

* Число восприимчивых немного уменьшается.
* Число инфицированных остается низким, но постепенно увеличивается.
* Число выздоровевших также постепенно увеличивается.

**= 1, = 0.3, = 0.1**

* Начальная популяция восприимчивых значительно уменьшается.
* Число инфицированных быстро растет, достигая пика, а затем снижается.
* Число выздоровевших увеличивается и стабилизируется на определенном уровне.
* Система достигает состояния равновесия, где популяция восприимчивых стабилизируется на уровне около 0.4, инфицированных - на уровне около 0.17, а выздоровевших - на уровне около 0.45.

# 5 Выводы

Я получила практические навыки работы со средствами моделирования xcos, Modelica и OpenModelica. Была рассмотрена модель эпидемии (SIR).

# Список литературы

1. Руководство к выполнению упражнения [Электронный ресурс]. URL: <https://esystem.rudn.ru/mod/resource/view.php?id=1223346>.