Лабораторная работа 5. Модель эпидемии (SIR)

5.1. Математическая модель

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. Типичная эволюция особи популяции описывается следующей диаграммой:

$$S \to I \to R.$$

Считаем, что система замкнута, т.е.

$$N = S + I + R.$$

Если предположить, что каждый член популяции может контактировать с каждым, то задача о распространении эпидемии описывается системой дифференциальных уравнений:

$$\begin{cases} \dot{s} = -\beta s(t) i(t); \\ \dot{i} = \beta s(t) i(t) - \nu i(t); \\ \dot{r} = \nu i(t). \end{cases} \tag{5.1}$$

где β — скорость заражения, ν — скорость выздоровления.

В представленной модели сделано предположение, что все особи популяции могут равновероятно заразиться со скоростью β.

Первое уравнение описывает динамику численности уязвимых к болезни особей: заражённая особь с некоторой скоростью β заражает уязвимую особь. Таким образом, инфицированная особь вступает в контакт и может передавать болезнь другим со скоростью βN. При этом доля контактов с инфицированными составляет $s(t)/N$. Число новых инфицированных в единицу времени составляет $\beta N (s(t)/N) i(t) = \beta s(t) i(t)$.

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

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

5.2. 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*, *Задать переменные окружения* зададим значения переменных β и ν (рис. 5.1).

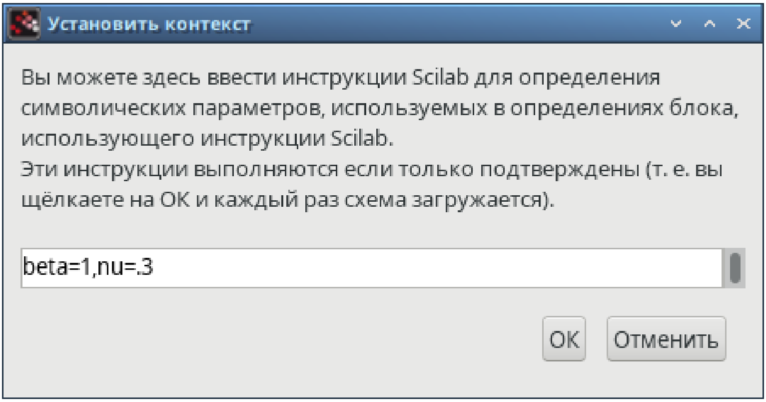

Рис. 5.1. Задать переменные окружения в xcos

Для реализации модели (5.1) потребуются следующие блоки xcos:
– `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:

$$y(t) = \int\limits_{t_0}^{t} u(t)dt + y_0;$$

– `GAINBLK_f` — в данном случае позволяет задать значения коэффициентов β и ν;
– `SUMMATION` — блок суммирования;
– `PROD_f` — поэлементное произведение двух векторов на входе блока.

Готовая модель SIR представлена на рис. 5.2.

Первое уравнение модели (5.1) задано верхним блоком интегрирования, блоком произведения и блоком задания коэффициента β. Блок произведения соединён с выходами верхнего и среднего блоков интегрирования и блоком коэффициента β, что реализует математическую конструкцию $s(t)i(t)\beta$.

Третье уравнение модели (5.1) задано нижним блоком интегрирования и блоком задания коэффициента ν. Для реализации математической конструкции $\nu i(t)$ соединяем выход среднего блока интегрирования и вход блока задания коэффициента ν, а результат передаём на вход нижнего блока интегрирования.

Средний блок интегрирования и блок суммирования определяют второе уравнение модели (5.1), которое по сути является суммой правых частей первого и третьего уравнений (5.1). Для реализации соединяем входы верхнего и нижнего блоков интегрирования с входами блока суммирования, меняя при этом в его параметрах оба знака на минус. Выход блока суммирования соединяем с входом среднего блока интегрирования.

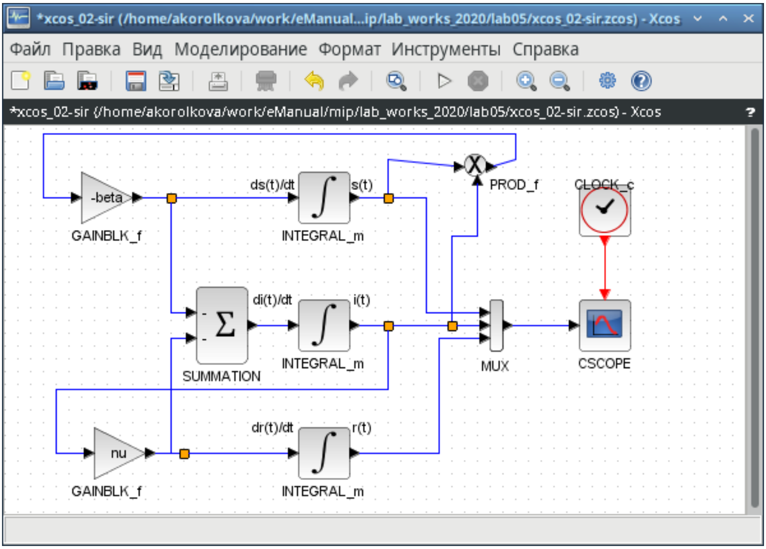

Рис. 5.2. Модель SIR в xcos

Выходы трёх блоков интегрирования соединяем с мультиплексором.

В параметрах верхнего и среднего блока интегрирования необходимо задать начальные значения $s(0) = 0,999$ и $i(0) = 0,001$ (рис. 5.3).

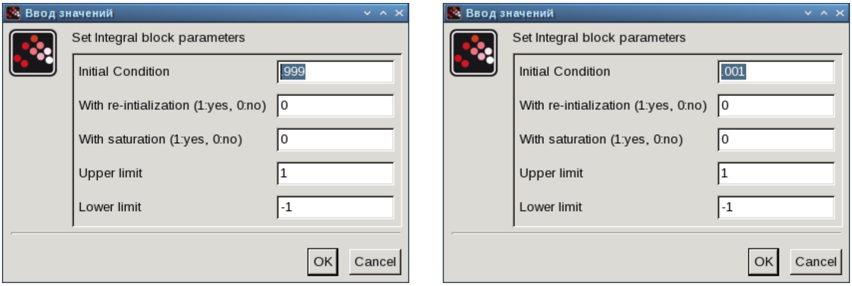

Рис. 5.3. Задать начальные значения в блоках интегрирования

В меню *Моделирование*, *Установка* необходимо задать *конечное время интегрирования*, равным времени моделирования (в данном случае 30, см. рис. 5.4).

Результат моделирования представлен на рис. 5.5, где сплошной линией обозначен график $s(t)$ (динамика численности уязвимых к болезни особей), пунктирная линия определяет $r(t)$ — динамику численности выздоровевших особей, наконец, пунктирная с точкой линия определяет $i(t)$ — динамику численности заражённых особей. Пересечение трёх линий определяет порог эпидемии.

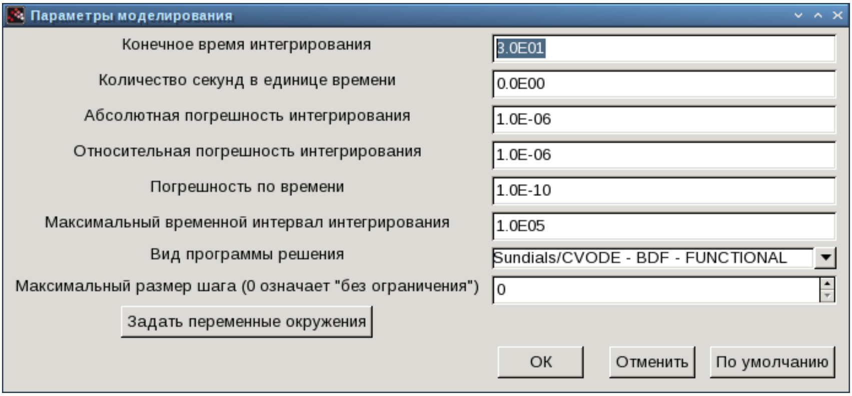

Рис. 5.4. Задать конечное время интегрирования в xcos

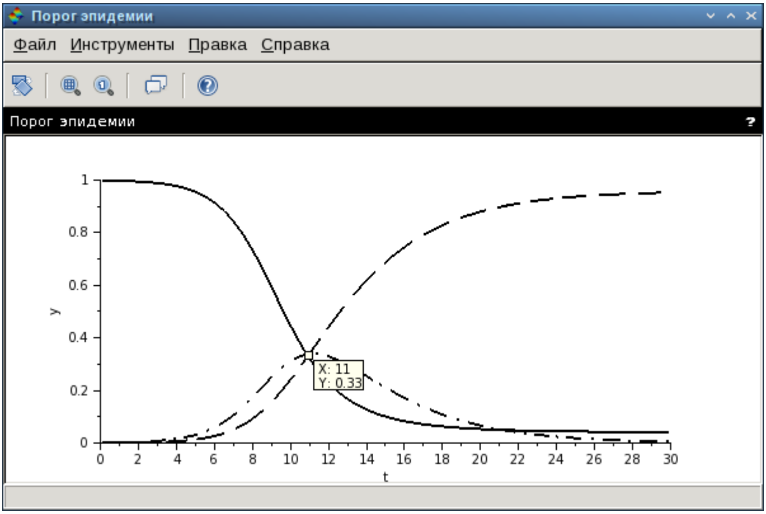

Рис. 5.5. Эпидемический порог модели SIR 5.1 при $\beta = 1, \nu = 0.3$

5.3. Реализация модели с помощью блока Modelica в xcos

Готовая модель SIR представлена на рис. 5.6.

Для реализации модели (5.1) 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. Задаём значения переменных β и ν (см. рис. 5.1).

Параметры блока Modelica представлены на рис. 5.7. Переменные на входе (“beta”, “nu”) и выходе (“s”, “i”, “r”) блока заданы как внешние (“E”).

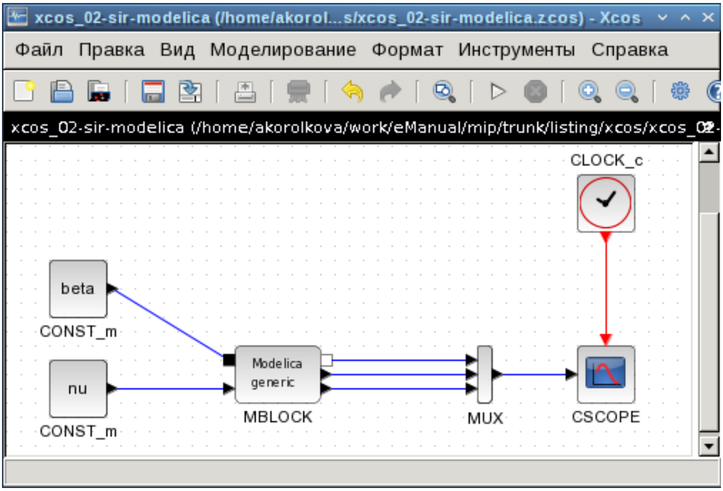

Рис. 5.6. Модель SIR в xcos с применением блока Modelica

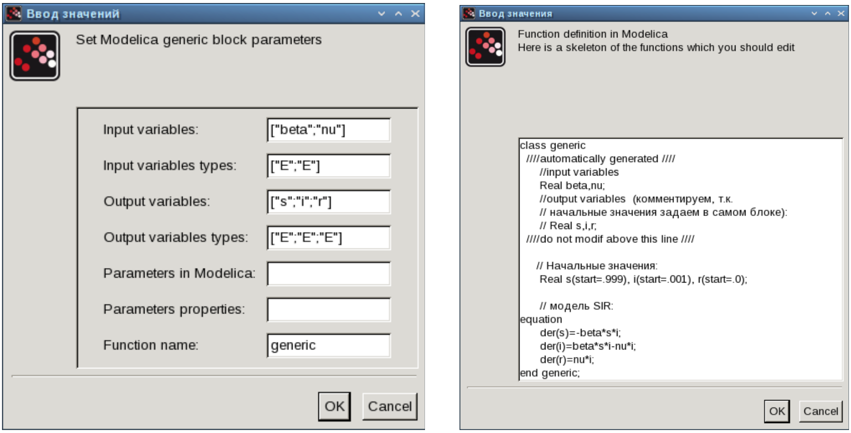

Рис. 5.7. Параметры блока Modelica для модели (5.1)

Код на языке Modelica:

```
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;
```

Результат моделирования совпадёт с рис. 5.5.

Упражнение Реализуйте модель SIR в OpenModelica.

5.4. Задание для самостоятельного выполнения

В дополнение к предположениям, которые были сделаны для модели SIR (5.1), предположим, что 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:

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

где μ — константа, которая равна коэффициенту смертности и рождаемости.

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.