Введение

Имитационное моделирование (от англ. Simulation) – это распространенная разновидность аналогового моделирования, реализуемого с помощью набора математических инструментальных средств, специальных имитирующих компьютерных программ и технологий программирования.

Имитационной моделью называется спец. программный комплекс, который позволяет имитировать деятельность какого либо сложного объекта. Имитационную модель нужно создавать. Для этого необходимо спец. программное обеспечение – ***система моделирования*** (simulation system). Специфика такой системы определяется технологией работы, набором языковых средств, сервисных программ и приемов моделирования.

В данной работе такой системой моделирования является подсистема Simulink приложения MatLab (матричная лаборатория), которое является мощным математическим пакетом.

Данные методические указания предназначены в помощь студентам при выполнении лабораторных работ по курсу «Математическое и имитационное моделирование».

В методических указаниях содержатся краткие теоретические сведения, необходимые при выполнении той или иной работы, задания и примеры выполнения лабораторных работ. Лабораторные работы выполняются с помощью системы моделирования Simulink, поэтому методические указания содержат сведения о системе моделирования Simulink и знакомят студентов с некоторыми особенностями работы в ней.

Тема 1. Знакомство с пакетом MatLab и системой моделирования Simulink

MATLAB – это высокоэффективное программное обеспечение для научных и инженерных расчетов. Одно из основных достоинств пакета состоит в том, что для работы пользователю достаточно знать о нем ровно столько, сколько требует решаемая задача.

С точки зрения пользователя MATLAB представляет собой богатейшую библиотеку функций. Для облегчения поиска библиотека разбита на разделы. Те функции, которые носят наиболее общий характер и используются наиболее часто, входят в состав ядра MATLAB. Те же, которые являются специфическими, включены в состав специализированных разделов, которые называются *Toolboxes (Инструменты)*.

Лабораторная работа № 1

Сведения о пакете MatLab и системе моделирования Simulink

Особое место среди инструментальных приложений пакета MATLAB занимает система визуального моделирования SIMULINK. Её можно рассматривать, как самостоятельный продукт фирмы Math Works, однако работает он только при наличии ядра MATLAB и использует многие функции, входящие в его состав.

Начало работы

Чтобы запустить MATLAB, надо начать с команд ***пуск – программмы*** – *MATLAB – MATLAB* ***NN*** (где NN – название версии) или нажать соответствующий ярлык или пиктограмму. После этого на экране на некоторое время, зависящее от быстродействия ПК, появляется титульная заставка системы MATLAB (рис. 1.1). Оно вскоре сменяется основным окном системы, в центре которого располагается командное окно (рис. 1.2).

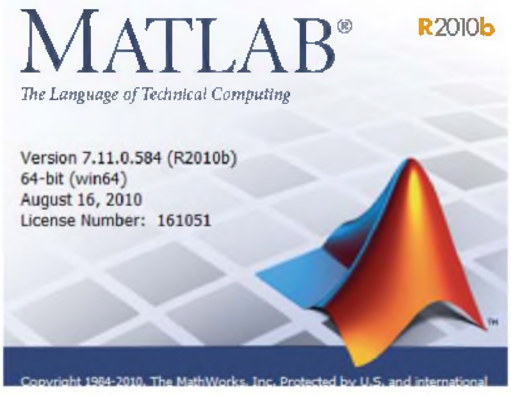

Рис. 1.1. Титульная заставка системы MATLAB

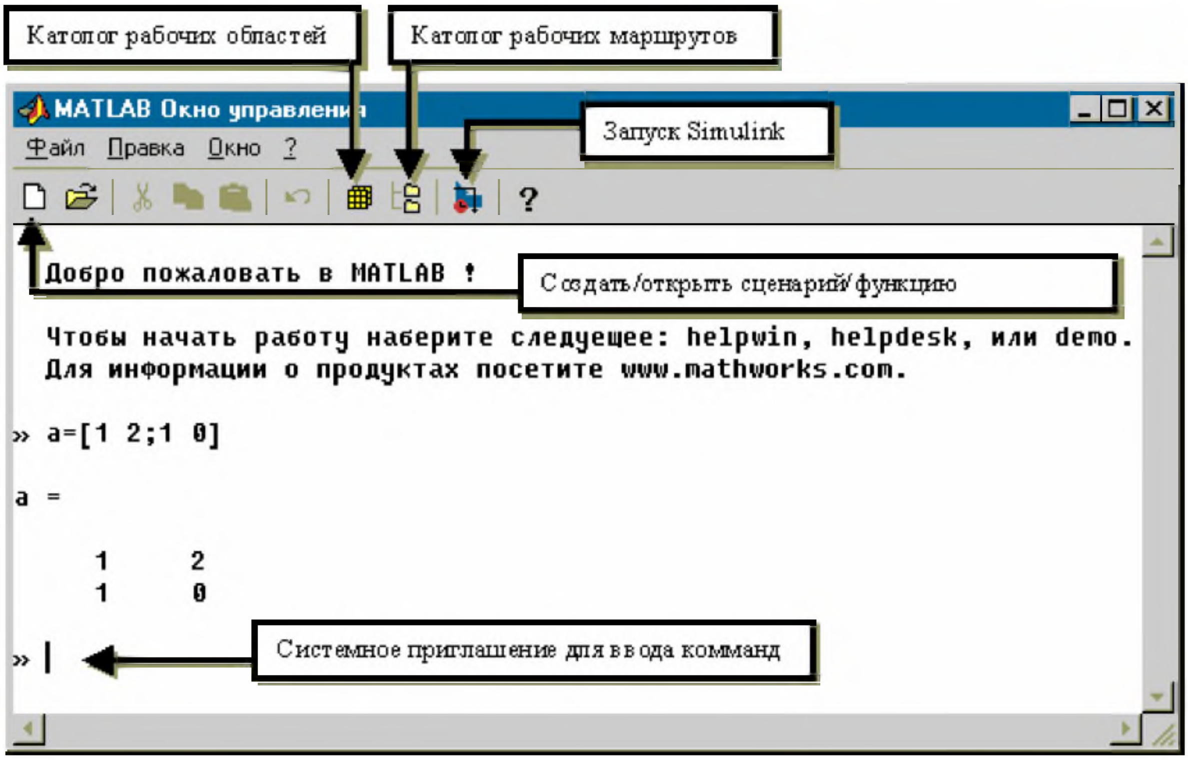

Рис. 1.2. Пример командного окна MATLAB

Запустить SIMULINK можно одним из 3 способов:

- щелкнув по соответствующей кнопке  панели инструментов командного окна MATLAB;
- введя команду Simulink в активной строке командного окна;
- выбрав команду New ► Model (создать ► модель) в меню File.

Использование первого и второго способов приводит к открытию окна просмотра библиотеки Simulink, а при выборе команды New ► Model кроме него открывается еще и пустое окно для создания S-модели.

Библиотеки блоков SIMULINK

Список разделов библиотеки Simulink представлен в основном окне просмотра в виде дерева (рис.1.3.). Структура библиотеки Simulink:

❖ Основная библиотека:

Разделы основной библиотеки:

Блоки, входящие в разделы;

❖ Расширения основной библиотеки, относящиеся к наборам инструментов MATLAB.

Нижний уровень иерархии образуют собственно блоки SIMULINK, которые и играют роль кирпичиков при построении S-модели. Чтобы вставить блок в S-модель, необходимо нажать кнопку мыши на графической или текстовой метке блока и, не отпуская кнопку мыши, переместить его в окно блок-диаграммы.

Строка меню окна блока-диаграммы содержит кроме общеизвестных меню ещё два:

Tools – инструменты;

Simulation – моделирование.

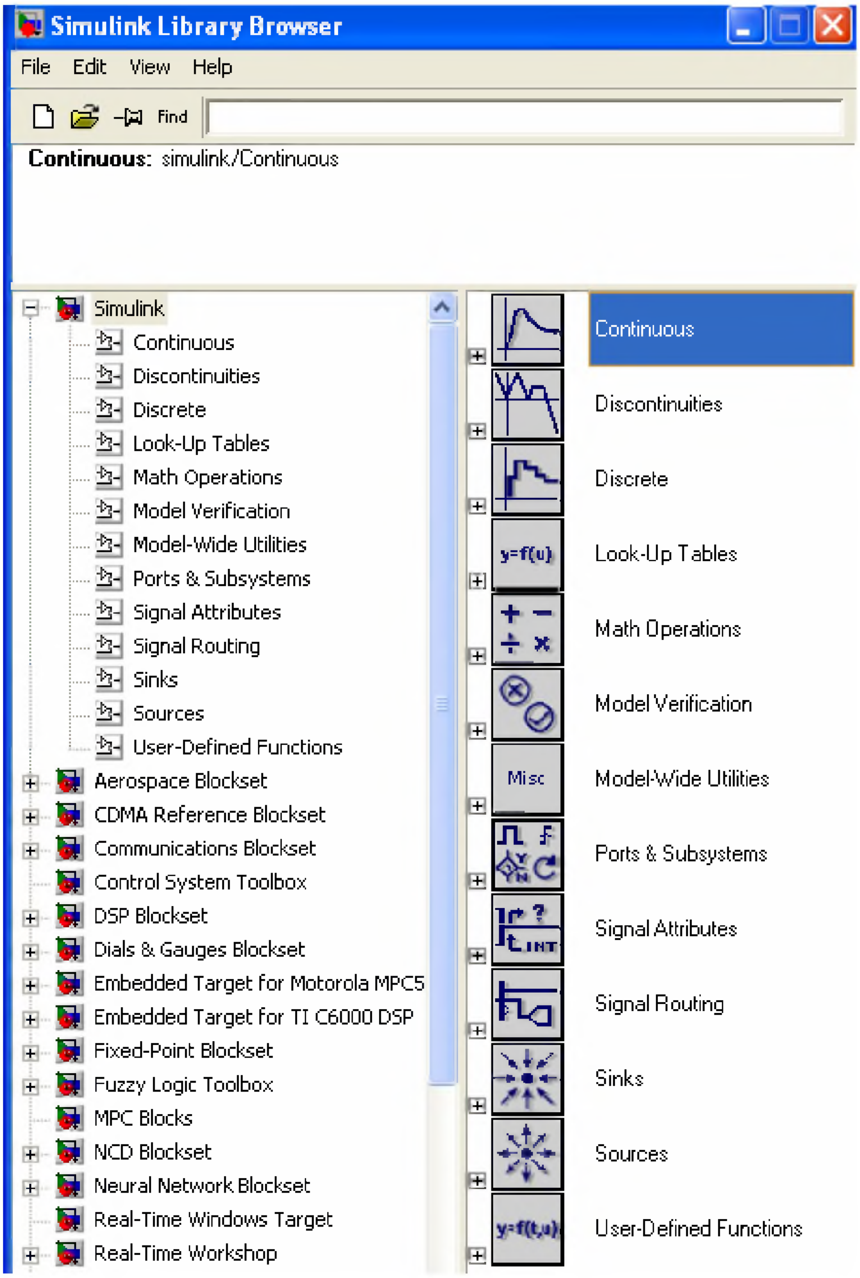

Рис.1.3. Основная библиотека блоков SIMULINK

Демонстрация возможностей

Чтобы получить представление о возможностях MATLAB и о том, что такое модель, разработанная с помощью SIMULINK, можно воспользоваться демонстрационными средствами MATLAB (команда demo в активной строке командного окна).

Задание к лабораторной работе № 1

1. Знакомство с возможностями MATLAB

1.1. Наберите команду ***demo*** в активной строке командного окна и нажмите ***Enter***. Откроется окно демонстрации (рис. 1.4). В левой части окна нажмите на + перед *MATLAB*. После того, как развернется список папок, можно по очереди выбирать папки и посмотреть, какие возможности пакет MATLAB предлагает пользователям.

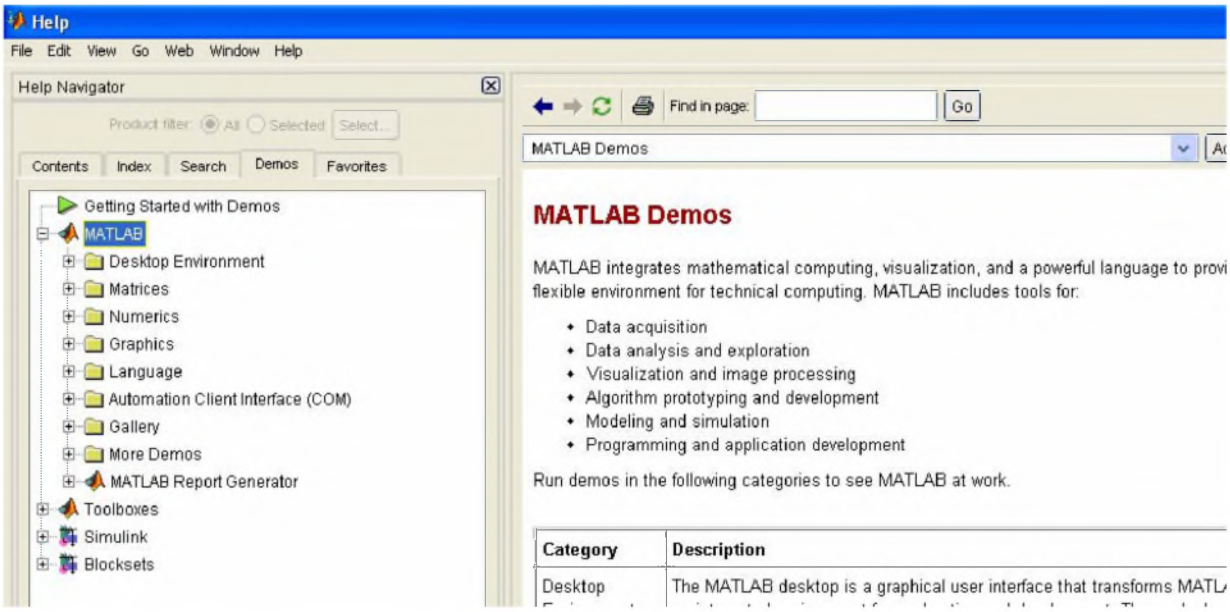

Рис. 1.4. Демонстрационные возможности MATLAB

1.2. Нажмите + перед Matrices и запустите демонстрацию Basic Matrix Operations (например, двойным щелчком). Первым слайдом откроется картинка с логотипом пакета MATLAB (рис. 1.5.).

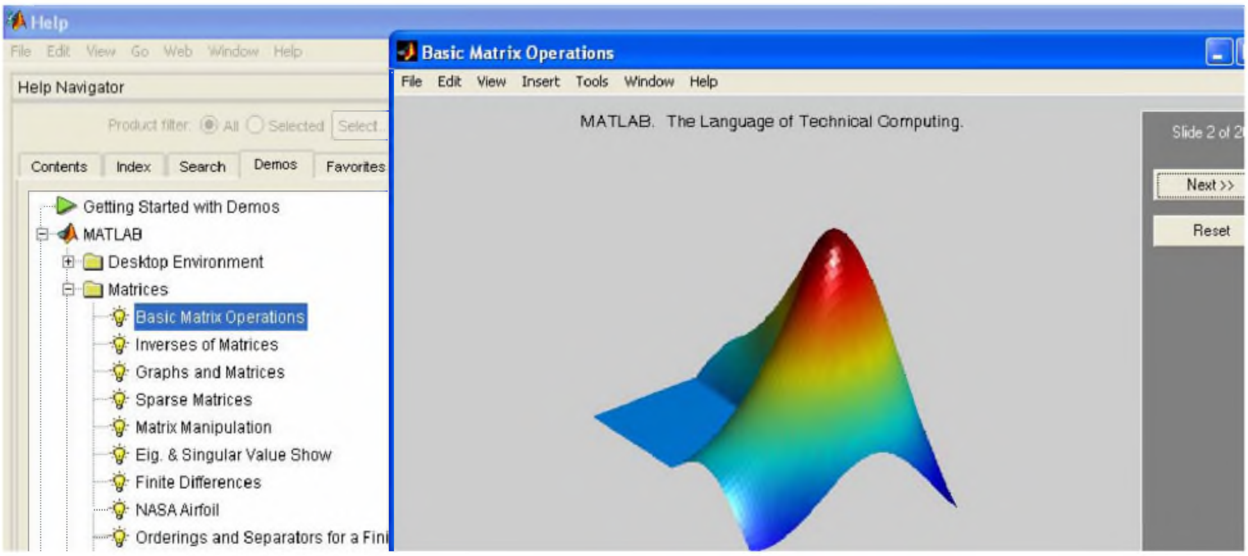

Рис.1.5. Демонстрация основных матричных операторов

1.3. По очереди посмотрите все файлы, предложенные демонстрацией возможностей MATLAB, особенное внимание уделите разделам Матрицы, Графики и Галерея, где можно увидеть интересные примеры различных графиков (рис. 1.6).

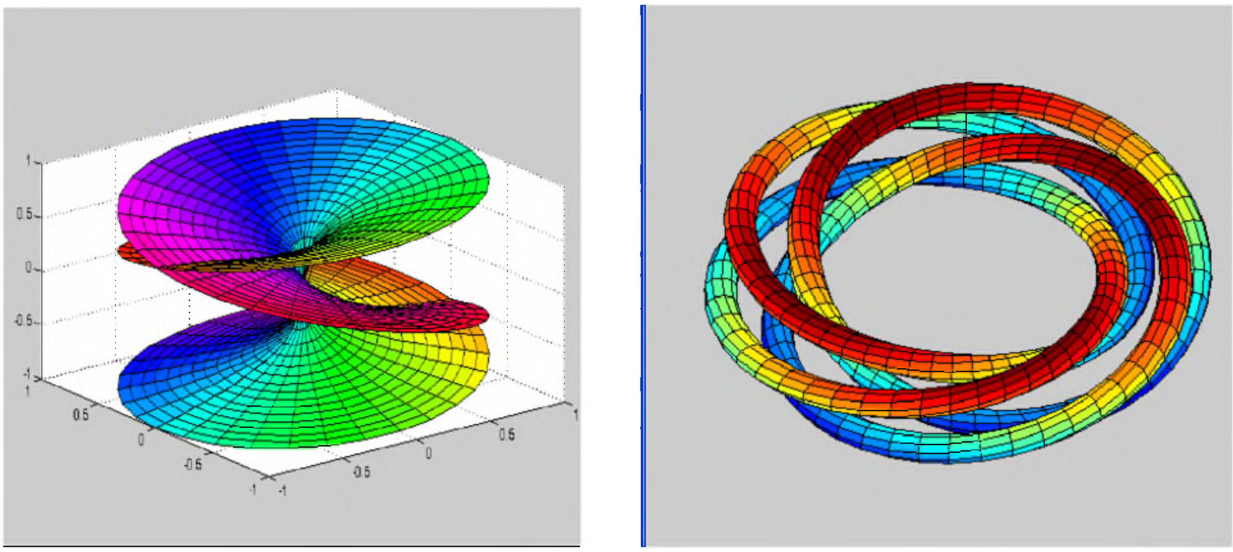

Рис.1.6. Примеры графиков сложных функций

2. Знакомство с возможностями SIMULINK

2.1. Сверните в левой части демонстрационного окна папку MATLAB и разверните папку SIMULINK. Выберите из списка папку General – Главная (рис. 1.7).

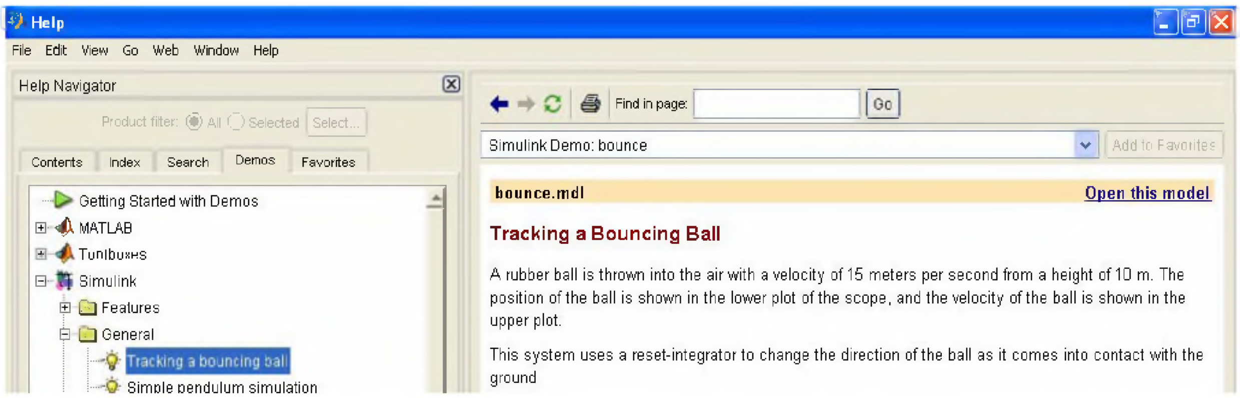

Рис. 1.7. Главные примеры из SIMULINK

2.2. Выберите пример «Траектория прыгающего мяча», откройте модель двойным щелчком или командой «Открыть модель». Откроется окно модели с двумя окнами для графиков. Запустите модель на выполнение, нажав кнопку со стрелкой ▶. В окнах для графиков появятся сами графики. Верхний – изменение скорости, нижний – изменение траектории со временем (рис. 1.8). В окне модели слева можно внести изменения и посмотреть, как это отразится на графиках. Можно изменить гравитационную постоянную и посмотреть, как изменится траектория прыгающего мяча например, для Луны или Юпитера, – для этого надо двойным щелчком открыть блок ***Gravity***, изменить константу, нажать ОК и запустить модель на выполнение снова. Аналогично можно изменить коэффициент упругости в блоке ***Elasticity*** и посмотреть на изменения.

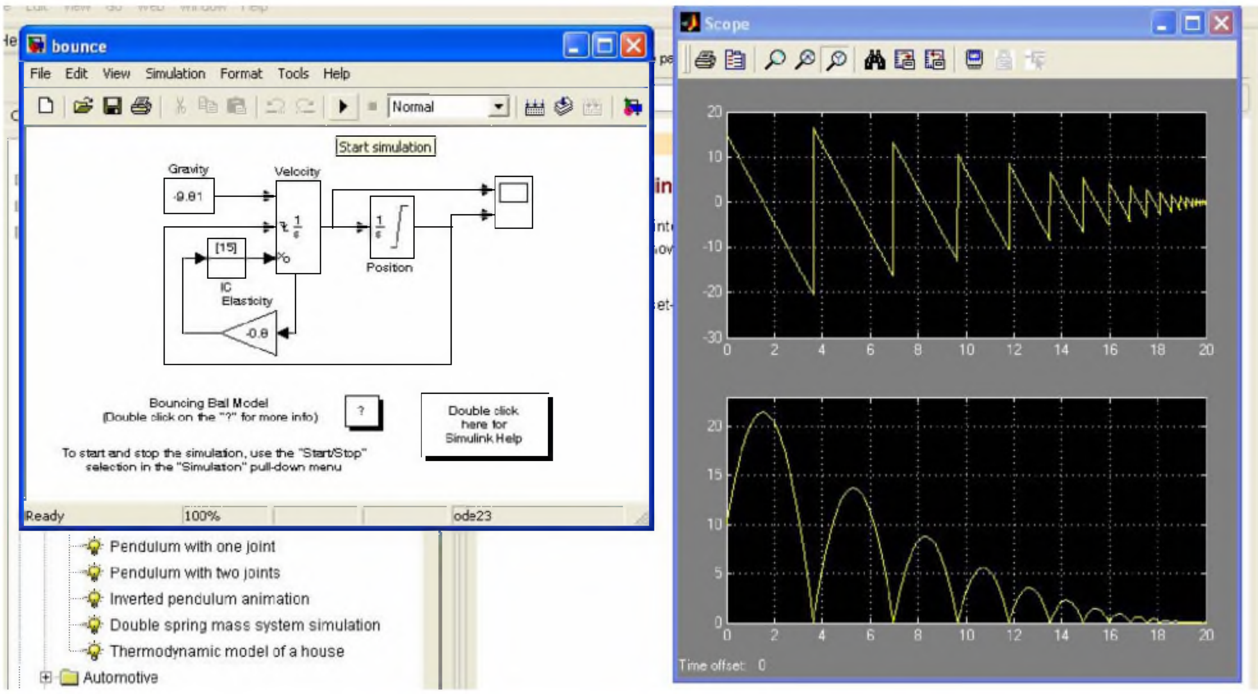

Рис. 1.8. Модель «Траектория прыгающего мяча»

2.3. Ознакомьтесь с моделью «Термодинамическая модель дома», тоже находящейся в Главной папке SIMULINK (рис. 1.9). В этой модели можно поменять температуру окружающей среды, среднюю температуру в доме (в Фаренгейтах) и последить на графиках, как будет меняться температура, расход электроэнергии и плата за электричество.

2.4. Самостоятельно ознакомьтесь с другими моделями из демонстрации SIMULINK, среди которых есть весьма интересные.

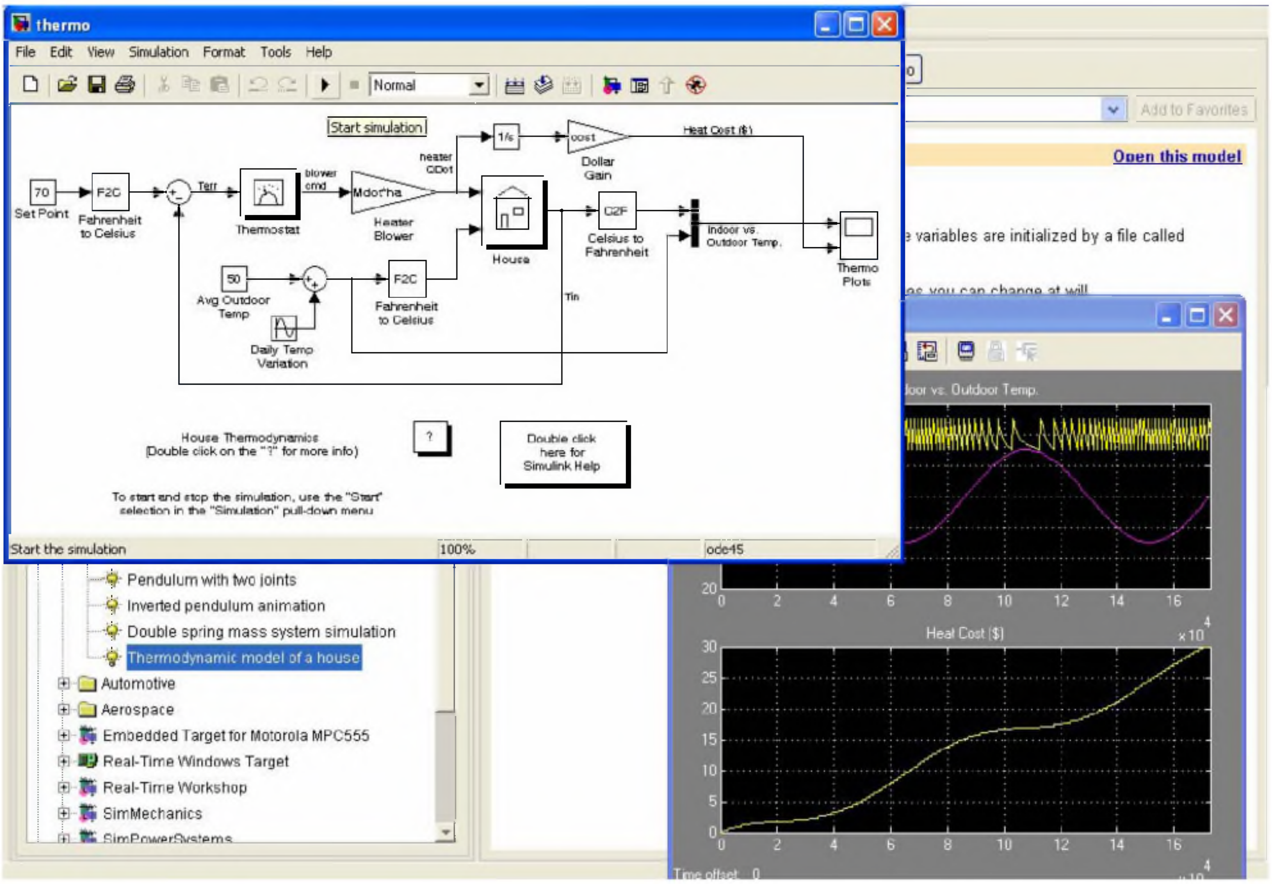

Рис.1.9. Термодинамическая модель дома

3. Знакомство с блоками Sources и Sinks из SIMULINK

3.1. Закройте окно демонстрации и откройте новую модель SIMULINK. При этом слева откроется список библиотек блоков (см. рис. 1.3), а с правой стороны – пустое окно для новой модели. Разработка моделей средствами Simulink (в дальнейшем S-моделей) основана на технологии drag-and-drop.

3.2. Откройте двойным щелчком библиотеку ***Sinks*** – Получатели и переместите в окно S-модели блоки ***Display*** – Дисплей, показывающий последнее значение входящего сигнала, и ***Scope*** – График-«осциллограф», показывающий изменение сигнала во времени.

3.3. Откройте двойным щелчком библиотеку ***Sources*** – Источники и переместите в окно S-модели блок ***Clock*** (часы). Соедините эти блоки стрелочкой, как показано на рис. 1.10. Щелчком по стрелочке ▸ запустите модель на выполнение, на графике появится изменение сигнала. Если не весь график отображается, то можно нажать на значок  для изменения масштаба.

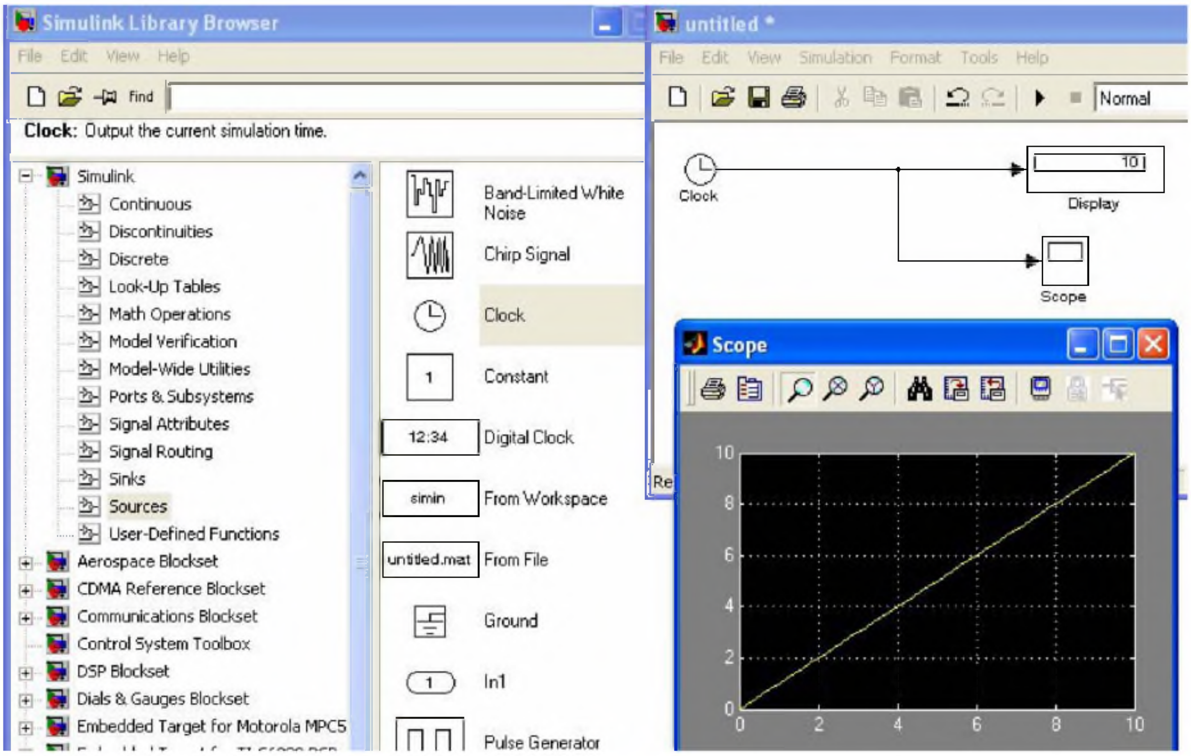

Рис. 1.10. Знакомство с блоками ***Получатели*** и ***Источники***

3.4. Замените блок ***Clock*** на блок ***Digital Clock,*** посмотрите, как изменится график в этом случае. Дважды щелкните по блоку, откроется окно параметров. Измените шаг по времени с 1 на 2. Посмотрите, что получится (рис. 1.11).

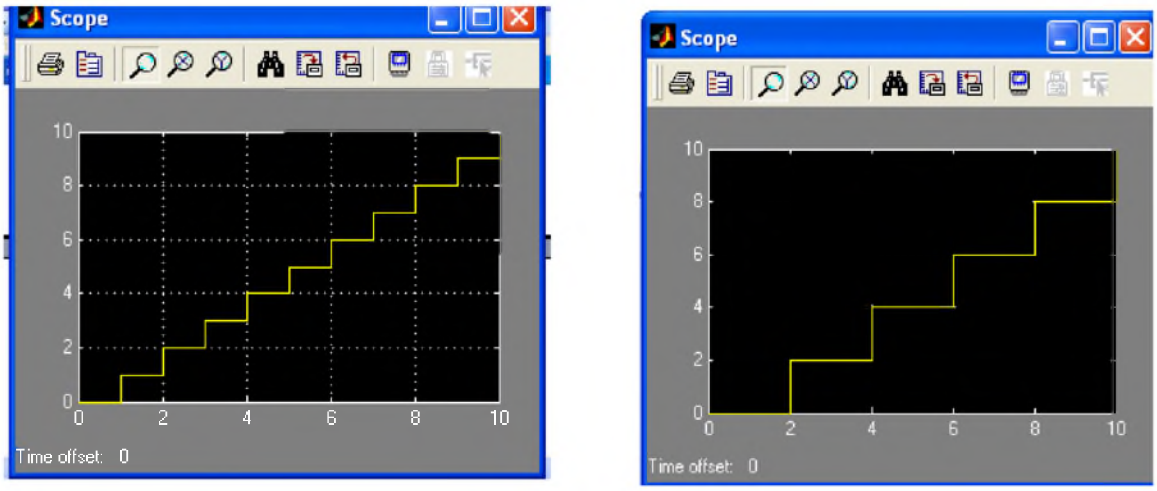

Рис.1.11. Графики блока ***Digital Clock*** с шагом 1 и шагом 2

3.5. По очереди поставьте в качестве блоков-источников другие блоки, где есть возможность, поменяйте параметры, посмотрите результаты.

Тема 2. Расчеты площадей методом Монте-Карло

Метод Монте-Карло (или метод статистических испытаний) можно определить как **метод моделирования случайных величин с целью вычисления характеристик их распределений**. Суть состоит в том, что результат испытаний зависит от некоторой случайной величины, распределенной по заданному закону. Поэтому результат каждого отдельного испытания носит случайный характер (как правило, составляется программа для осуществления одного случайного испытания). Проведя серию испытаний, получают **выборку**. Полученные статистические данные обрабатываются и представляются в виде численных оценок интересующих исследователя величин (характеристик системы).

Теоретической основой метода Монте-Карло являются предельные теоремы теории вероятностей. Они гарантируют высокое качество статистических оценок при весьма большом числе испытаний. Метод статистических испытаний применим для исследования как стохастических, так и детерминированных систем. Практическая реализация метода Монте-Карло невозможна без использования компьютера.

Лабораторная работа № 2

Первая модель – квадратура круга

Применим метод статистических испытаний к задаче, многие века занимающей умы многих математиков – нахождению квадратуры (площади) круга.

Рассмотрим решение задачи нахождения площади круга, ограниченной уравнением окружности с радиусом r = 5: $(x-2)^2+(y-3)^2=25$. Впишем этот круг в квадрат размером 10×10 (рис. 2.1). Центр окружности находится в точке с координатами (2; 3). Любая точка внутри квадрата должна удовлетворять неравенствам: $-3 < \boldsymbol{x} < 7$; $-2 < \boldsymbol{y} < 8$.

Для получения выборки будем исходить из того, что все точки в этом квадрате могут появляться с одинаковой вероятностью, т.е. ***x*** и ***y*** распределены равномерно. Проведя ***n*** испытаний, т.е. получив некоторое случайное множество точек, принадлежащих квадрату, подсчитаем количество точек ***m***, попавших в круг (рис. 2.2). Тогда оценку площади круга можно получить из соотношения: $S_{кр} = S_{кв} \cdot \frac{m}{n} = 100 \cdot \frac{m}{n}$.

Построим имитационную модель средствами Simulink. Сначала выберем источники равномерно распределенных чисел – датчики **Uniform-RandomNumber,** для моделирования координат ***x*** и ***y***, потом проверим, попадает ли точка с координатами (***x,y***) внутрь круга, т.е. удовлетворяют ли координаты неравенству $(x-2)^2+(y-3)^2 \le 25$. Неравенство зададим с помощью блока **Fcn** – Функция пользователя. Подсчитаем количество точек, попавших в круг, с помощью блока **Discrete-TimeIntegrator**, а потом посчитаем площадь круга блоком **Fcn1,** при этом общее количество точек получим с блока **Clock**. Результат выведем в блок **Display** (рис. 2.3).

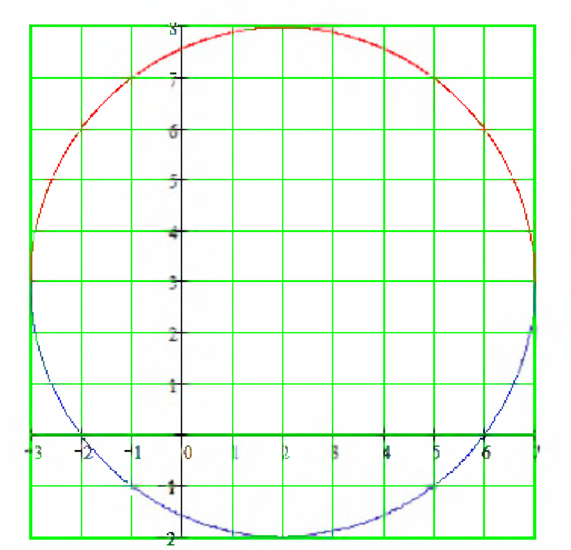

Рис. 2.1. Круг, вписанный в квадрат

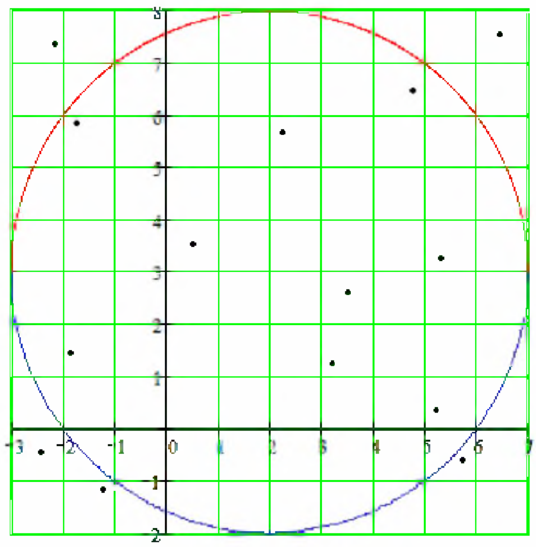

Рис. 2.2. Точки, случайно разбросанные по квадрату

Задание к лабораторной работе № 2

1. Построение модели

1.1. Соберите модель, приняв за образец схему на рис. 2.3.

1.2. Задайте параметры для блоков **UniformRandomNumber**, как показано на рис. 2.4, 2.5. Обратите внимание, что, кроме минимума и максимума, должны быть еще и разные начальные зерна (**Initial seed**).

1.3. Введите в блок **Fcn** функцию для проверки принадлежности точки (***x,y***) кругу. Блок **Mux** (смеситель) превращает координаты ***x*** и ***y*** в вектор ***u*** с индексами 1 и 2, поэтому левая часть неравенства примет вид $(u[1]-2)\text{^}2+(u[2]-3)\text{^}2$, где в квадратных скобках индексы элементов вектора ***u***.

1.4. Введите в блок **Fcn1** функцию для расчета площади круга. Функции опять предшествует блок **Mux** (смеситель), поэтому переменные в функции будут иметь имя ***u*** с номерами в той последовательности, в какой они введены в блок **Mux**: $u[1]*u[2]/u[3]$.

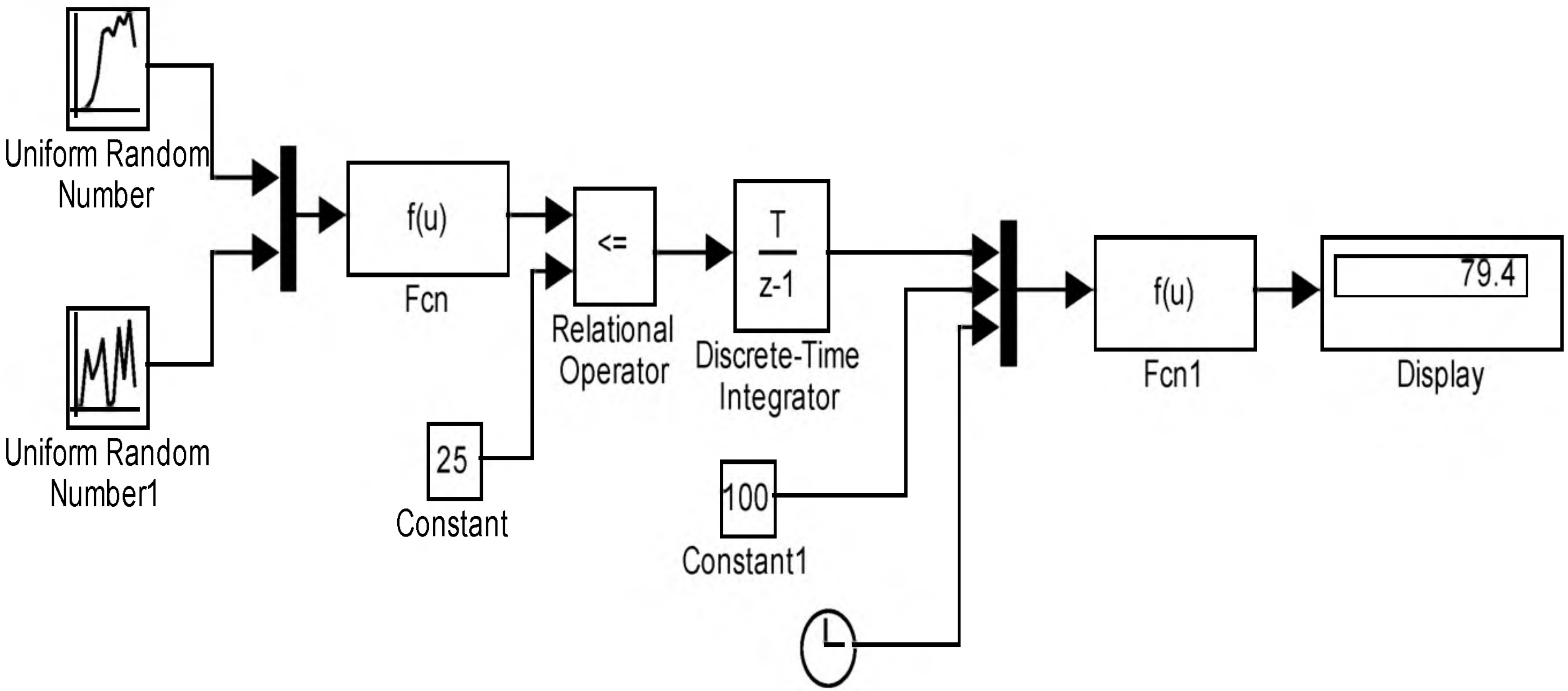

Рис. 2.3. Модель расчета площади круга

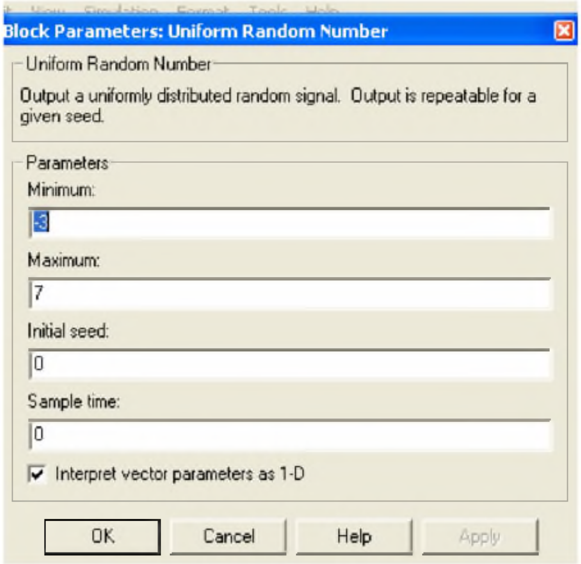

Рис. 2.4. Настройки для получения координаты ***x***

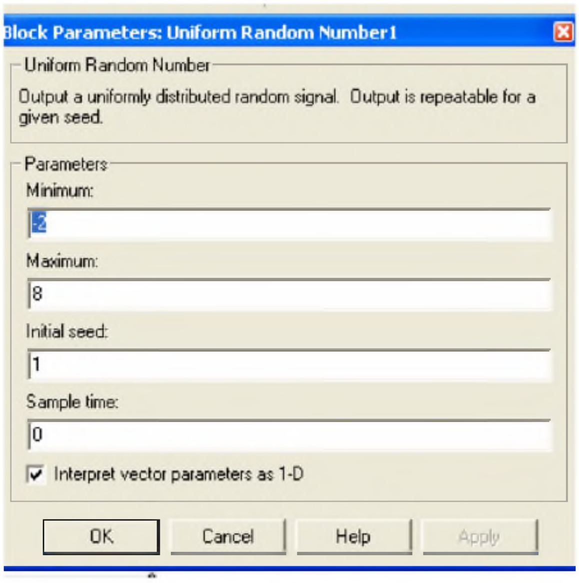

Рис. 2.5. Настройки для получения координаты ***y***

2. Настройка модели

2.1. Настройте собранную модель с помощью пункта меню **Simulation** → **SimulationParameters.** На вкладке **Solver** (Решатели) выберите Фиксированный шаг размером 1 (рис. 2.6), а на вкладке **Advanced** (Дополнительно) укажите, что *Логические сигналы* необходимо **off** (выключить) (рис. 2.7).

2.2. Запустите на выполнение построенную модель.

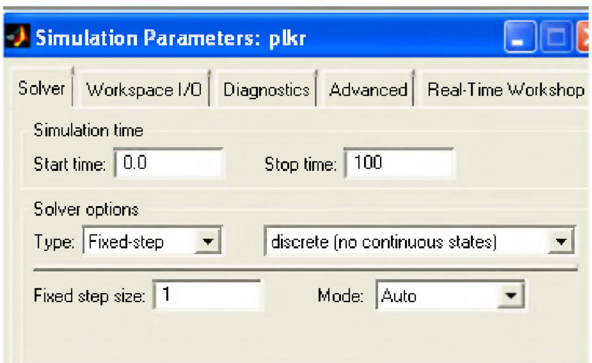

Рис. 2.6. Настройка параметров модели (вкладка Решатели)

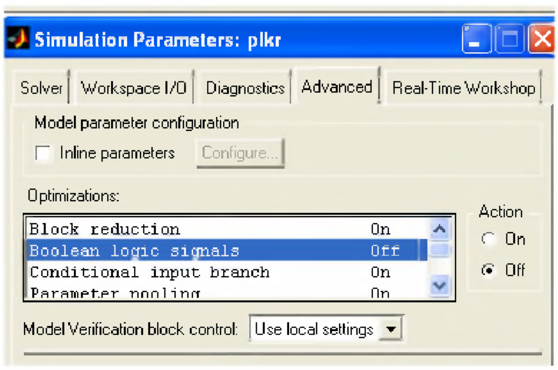

Рис. 2.7. Настройка параметров модели (вкладка Дополнительно)

2.3. Поменяйте количество точек, которое моделируете, с помощью параметра **Stop time** (см. рис. 2.6).

3. Подготовка статистического эксперимента

Запуск модели – это всего лишь один прогон, на основании которого нельзя сделать выводы о результате – вполне может быть, что сосчитанная один раз площадь круга очень далека от истинного значения. Чтобы можно было сделать выводы о результате имитационного моделирования, необходимо провести серию прогонов, причем для разного количества точек ***n***. Прогоны будут отличаться друг от друга последовательностями случайных чисел, из которых формировались координаты точек. Этого можно добиться, меняя начальное зерно (**Initial seed**). Единственное требование к начальным зернам – одно из зерен должно быть равно **0**, иначе между результатами двух датчиков **UniformRandomNumber** возникает линейная зависимость, и точки перестают быть равномерно распределенными.

3.1. Добавьте в модель дополнительные графические блоки для контроля за вычислениями (рис. 2.8). Блок **XY Graph** предназначен для контроля за разбросом точек на квадрате – за их равномерной распределённостью. А блок **Scope** показывает изменение значения площади круга в зависимости от количества точек.

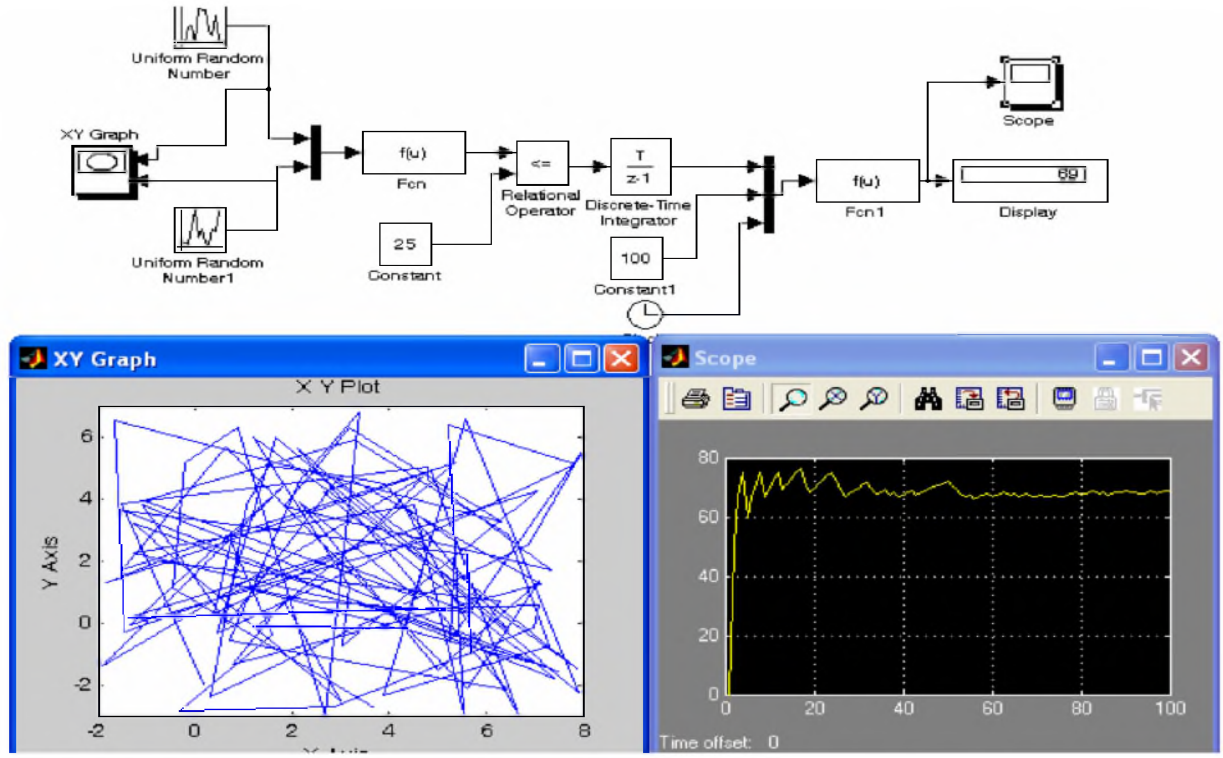

Рис. 2.8. Модель расчета площади круга с дополнительной графикой

Например, если при настройке параметров датчиков **Uniform Random Number** начальные зерна сделать ненулевыми, скажем, для получения ***x*** начальное зерно равно 1, а для получения ***y*** – 2, или оба начальных зерна сделать одинаковыми, то график **XY Graph** сразу же покажет, что такие зерна недопустимы, да и ответ будет далек от истинного (рис. 2.9, 2.10).

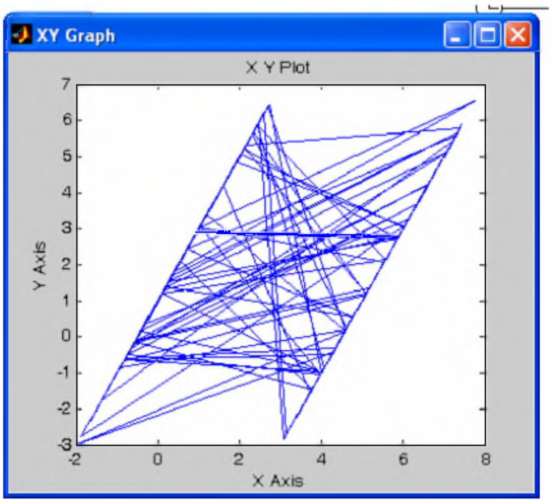

Рис. 2.9. График распределения точек (***x,y***) при обоих ненулевых начальных зернах

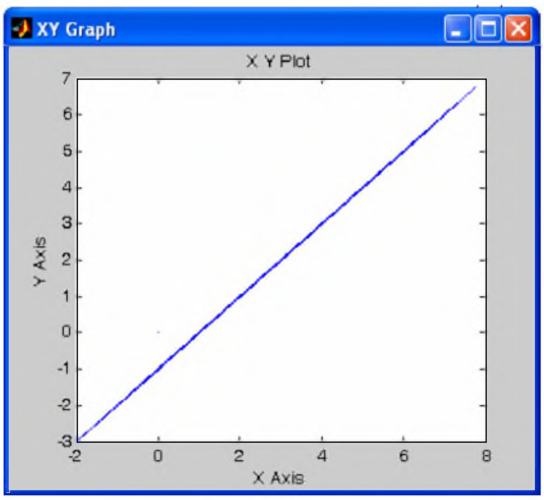

Рис. 2.10. График распределения точек ***(x,y)*** при одинаковых начальных зернах

3.2. Заготовьте в электронных таблицах **Excel** таблицу для занесения результатов статистических испытаний – так называемый план экспериментов, примерно как показано на рис. 2.11.

Оценки площади круга $S_{кр}$							
номер прогона	Объем испытаний (n)						
	100	500	1000	5000	10000	50000	100000
1 (1,0)							
2 (0,2)							
3 (5,0)							
4 (...)							
5 (...)							
...							
Среднее							
Дисперсия							

Рис. 2.11. Таблица для занесения результатов экспериментов

3.3. Проведите запланированные эксперименты, не забывая для каждого прогона менять начальные зерна в блоках **Uniform Random Number.** Если в одном прогоне при небольшом количестве испытаний графики показывают правильность протекания вычислений, то для большего количества испытаний графические блоки можно отключить, чтобы ускорить вычисления.

После заполнения таблицы сделайте выводы.

Лабораторная работа № 3

Расчет «неберущихся» интегралов

Конечно, смешно высчитывать площадь круга методом Монте-Карло, но применение метода Монте-Карло для вычисления площадей может применяться для вычисления так называемых «неберущихся» интегралов – т.е. интегралов, для которых первообразная функция не может быть выражена через элементарные функции. В таких случаях определенный интеграл рассчитывается численными методами с помощью производных с определенными погрешностями, так что применение метода Монте-Карло вполне оправдано (особенно если возможность взятия производной тоже под вопросом).

Пусть нужно сосчитать определенный интеграл в пределах от a до b от функции y = f(x). Для упрощения будем брать такие функции y = f(x), график которых в пределах интервала (a,b) лежит выше оси OX. Тогда интеграл будет равен площади фигуры, лежащей под кривой y = f(x) (рис.2.12).

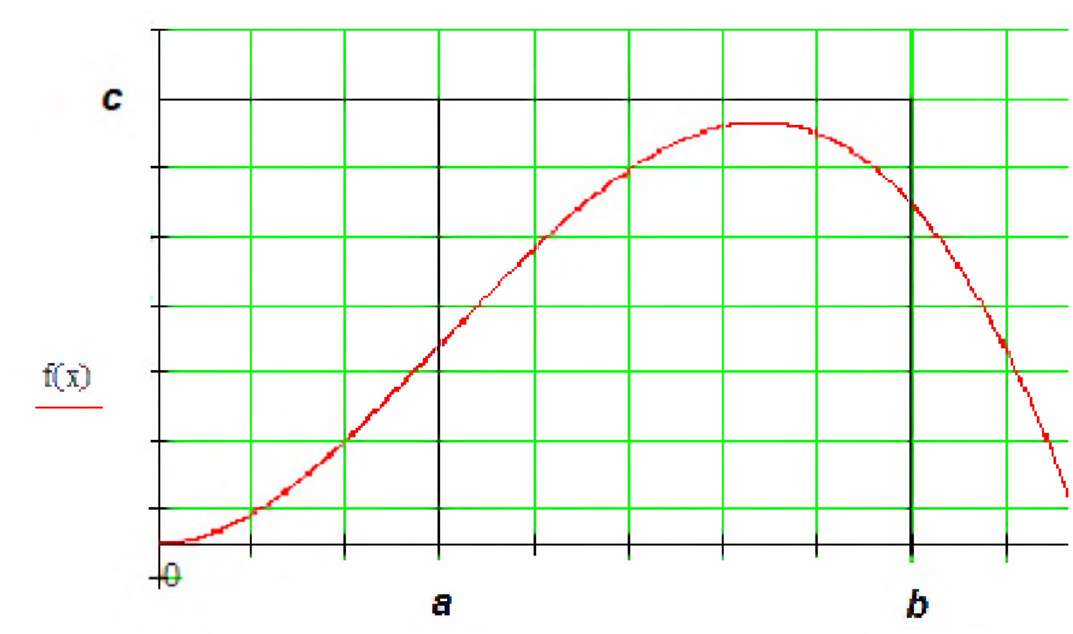

Рис. 2.12. Пример кривой, интеграл которой надо посчитать

Воспользуемся таким же подсчетом точек и пропорциями площадей, что и в прошлой лабораторной работе. Будем моделировать равномерно распределенные точки в прямоугольнике, координаты будут лежать в пределах a<x<b, а 0<y<c, где c – координата по оси OY, равная или больше max(f(x)). Тогда интеграл $\int_a^b f(x)dx = S_{пр} \cdot \frac{N}{n}$, где n – общее количество точек, *N* – количество точек, попавших под кривую *f(x)*, $S_{пр}$ – площадь прямоугольника, на который производится «бросание» точек.

Задание к лабораторной работе № 3

1. Построение модели

1.1. Соберите модель, используя за образец схему на рис. 2.13. Задайте параметры для блоков **UniformRandomNumber** для своей задачи по образцу прошлой модели. Следите за начальными зернами.

1.2. Задайте в блоке **Fcn** функцию для проверки принадлежности точки ***(x,y)*** области под графиком функции.

1.3. Введите в блок **Fcn1** функцию для расчета площади фигуры.

2. Настройка модели

2.1. Настройте свою модель аналогично модели для расчета площади круга.

2.2. Запустите на выполнение построенную модель.

3. Подготовка статистического эксперимента

3.1. Заготовьте в электронных таблицах Excel таблицу для занесения результатов статистических испытаний.

3.2. Проведите запланированные эксперименты, не забывая для каждого прогона менять начальные зерна в блоках **Uniform Random Number.**

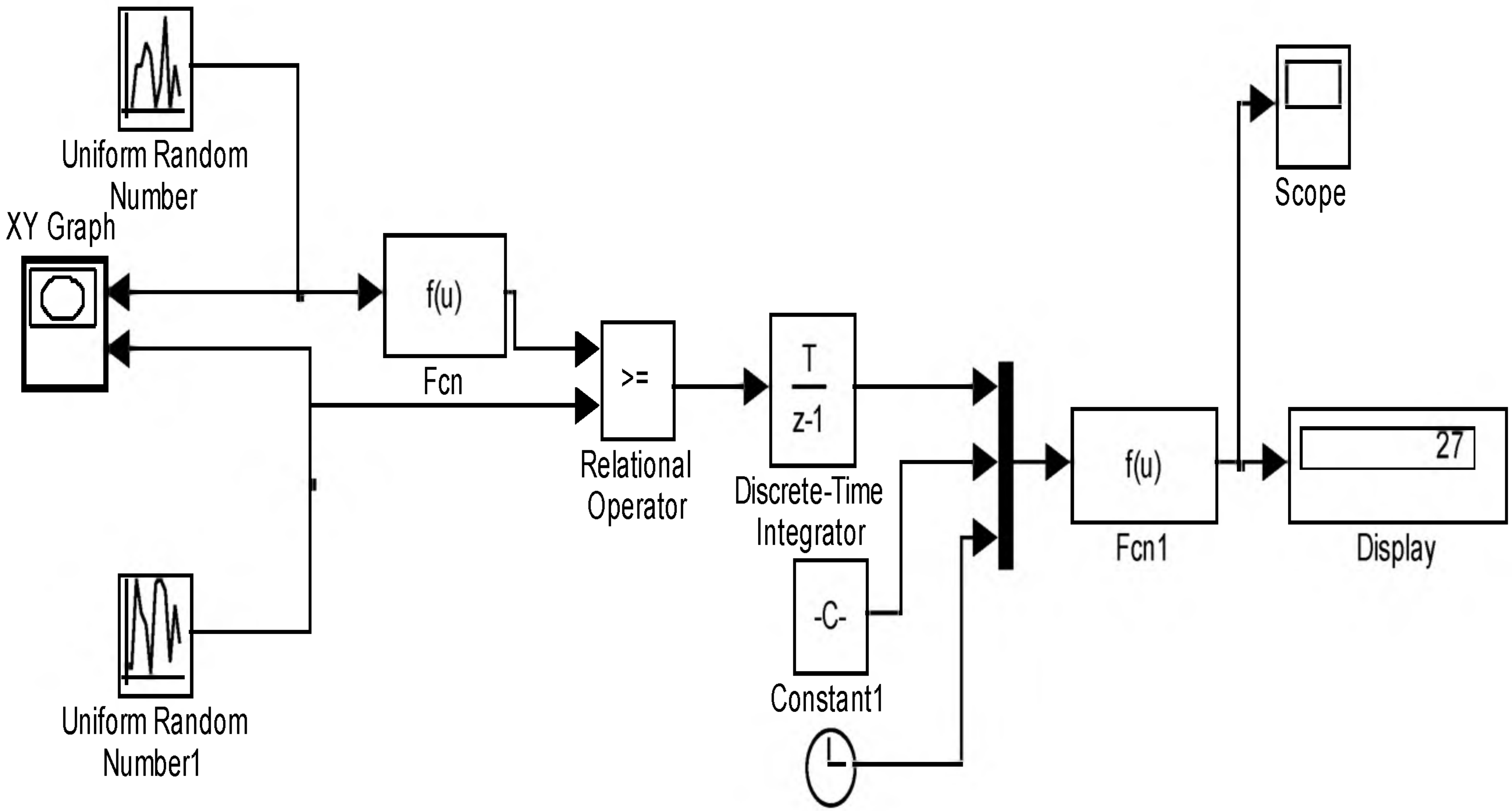

Рис. 2.13. Пример схемы модели для подсчета интегралов