

# ОСНОВЫ СИСТЕМ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ



Маликов Р.Ф.



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
БАШКИРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМ. М.АКМУЛЛЫ**

**Р.Ф. Маликов**

**ОСНОВЫ СИСТЕМ  
КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ**

**УЧЕБНОЕ ПОСОБИЕ**

**Уфа 2008**

УДК

ББК 22.38я73

М13

Печатается по решению редакционно-издательского совета БГПУ

**Маликов Р.Ф.** Основы систем компьютерного моделирования: учеб. пособие. – Уфа: Изд-во БГПУ, 2008. - 265с.

Учебное пособие предназначено для студентов, обучающихся по специальностям 010400 – Физика, 030100 – Информатика, 230201- Информационные системы, 030500.06 - Профессиональное обучение (информатика, ВТ и компьютерные технологии), по направлениям 510400 –Физика, 511800 – Математика, компьютерные науки, для изучения основ компьютерного моделирования реальных объектов в системах компьютерной математики и технического моделирования.

Рецензенты:

д-р ф.-м. н., проф. В.П.Житников(УГАТУ),

д-р т.н., член-корр. АН РБ, проф. Р.А.Валиуллин (БГУ).

JSBN

© Изд-во БГПУ, 2008

© Р.Ф.Маликов, 2008

© Дизайн обложки, А.Куликов, 2008

## Содержание

<b>ВВЕДЕНИЕ .....</b>	<b>7</b>
<b>ГЛАВА 1. МОДЕЛИРОВАНИЕ ОБЪЕКТОВ В СИСТЕМЕ EXCEL .....</b>	<b>14</b>
1.1. Моделирование движения небесного тела под действием сил тяготения.....	14
1.2. Движение тела в поле силы тяжести Земли .....	18
1.3. Движение заряженной частицы в кулоновском поле .....	20
<b>ГЛАВА 2. МОДЕЛИРОВАНИЕ ОБЪЕКТОВ В СИСТЕМЕ DERIVE.....</b>	<b>23</b>
2.1. Общие сведения о системе Derive .....	24
2.2. Работа с окнами и графикой .....	29
2.3. Аналитические методы решения дифференциальных уравнений в среде Derive .....	34
2.3.1. Функции решения уравнений первого порядка.....	35
2.3.2. Функции решения уравнений второго порядка.....	39
2.4. Численные методы решения дифференциальных уравнений.....	46
<b>ГЛАВА 3. МОДЕЛИРОВАНИЕ ОБЪЕКТОВ В СРЕДЕ MAPLE .....</b>	<b>57</b>
3.1. Общие сведения о системе Maple .....	57
3.2. Примеры моделирования в системе Maple .....	60
<b>ГЛАВА 4. МОДЕЛИРОВАНИЕ ОБЪЕКТОВ В СРЕДЕ MATHCAD .....</b>	<b>69</b>
4.1. Общие сведения о математическом пакете MathCAD.....	70
4.2. Способы моделирования в MathCAD.....	71
<b>ГЛАВА 5. МОДЕЛИРОВАНИЕ В СИСТЕМЕ MATHEMATICA .....</b>	<b>80</b>
5.1. Общие сведения о системе Mathematica 4 .....	81
5.2. Графические возможности .....	89
5.3. Справочная система пакета Mathematica .....	95
5.4. Решение дифференциальных уравнений в символьном виде	110
5.5. Решение дифференциальных уравнений в численном виде	101
<b>ГЛАВА 6. МОДЕЛИРОВАНИЕ ОБЪЕКТОВ В СРЕДЕ MATLAB .....</b>	<b>107</b>
6.1. Общие сведения о системе MATLAB 6.0.....	107
6.2. Визуализация и графические средства .....	112
6.3. Решение обыкновенных дифференциальных уравнений в системе MATLAB.....	115
<b>ГЛАВА 7. ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ С ПОМОЩЬЮ СХЕМОТЕХНИЧЕСКОЙ СИСТЕМЫ VISSIM.....</b>	<b>123</b>
7.1. Знакомство с системой технического проектирования VisSim.....	123



7.2. Технологии моделирования дифференциальных уравнений	126
7.2.1. Решение дифференциальных уравнений	126
7.2.2. Моделирование дифференциальных уравнений первого и второго порядков с использованием блока variable	128
7.2.3. Схемотехнические модели, описанные типовыми звеньями и передаточными функциями	130
7.2.4. Модели матричного типа	134
7.2.5. Моделирование линейной системы дифференциальных уравнений	135
<b>ГЛАВА 8. СХЕМОТЕХНИЧЕСКОЕ МОДЕЛИРОВАНИЕ В СРЕДЕ SIMULINK</b>	<b>139</b>
8.1. Работа с основным пакетом расширения Simulink	140
8.2. Математическое моделирование вынужденных колебаний двухмассовой системы	147
<b>ГЛАВА 9. КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ЭЛЕКТРОННЫХ СХЕМ В СРЕДЕ ELECTRONICS WORKBENCH</b>	<b>157</b>
9.1. Знакомство с системой компьютерного моделирования Electronics Workbench 5.12.	157
9.2. Некоторые возможности электронной лаборатории	163
9.3. Индуктивность и емкость в цепи переменного тока	166
9.4. Источники вторичного питания (выпрямление с фильтрацией)	168
9.5. Исследование логических схем в системе Electronics Workbench	170
9.6. Виртуальный логический конвертор (Logic converter)	172
9.7. Решение дифференциальных уравнений	175
<b>ГЛАВА 10. КОМПЬЮТЕРНОЕ ПРОЕКТИРОВАНИЕ ВИРТУАЛЬНЫХ ПРИБОРОВ В СРЕДЕ LABVIEW</b>	<b>177</b>
10.1. Что такое LabVIEW?	177
10.2. Панели и палитры LabVIEW	182
10.3. Функции в системе LabVIEW	183
10.3.1. Формула	190
10.3.2. Масштабирование и отображение	190
10.3.3. Математическая обработка во временной области	191
10.4. Технология проектирования виртуальных приборов	192
10.5. Структуры, массивы и графические индикаторы среды LabVIEW	197
10.6. Функции численных методов решения дифференциальных уравнений	214
10.6.1. Решение ОДУ методом Эйлера	214
10.6.2. Решение ОДУ методом Рунге-Кутты 4-го порядка	215
10.6.3. Решение ОДУ методом Кэш-Капа 5-го порядка точно	

сти.....	216
10.6.4. Моделирование системы «Двойной маятник».....	217
<b>РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.....</b>	<b>219</b>
<b>ПРИЛОЖЕНИЕ 1.....</b>	<b>223</b>
<b>ПРИЛОЖЕНИЕ 2.....</b>	<b>234</b>
<b>ПРИЛОЖЕНИЕ 3 .....</b>	<b>246</b>

## ВВЕДЕНИЕ

Одно из направлений развития вычислительных технологий в настоящее время — это появление мощных математических пакетов, позволяющих максимально упростить процесс подготовки задачи, ее решения и анализа результатов. При использовании таких средств, как Maple, Mathcad, Mathematica или MATLAB, решение дифференциального или трансцендентного уравнения, аналитическое либо численное дифференцирование и интегрирование, обращение матрицы, решение проблемы собственных значений, вычисление предела, разложение в ряд и многие другие задачи решаются с помощью одной команды. Но, конечно, эту команду нужно правильно применить: надо корректно сформулировать задачу: знать, в каком виде искать решение и т. д.

Создание универсальных программных средств символьной математики стало, в последнее время, основой нового научного направления в информатике, которое получило название — *компьютерная алгебра*. В настоящее время наиболее известными и широко используемыми на практике являются такие математические системы, как Mathematica, Maple, Derive, Mathcad и др. Они позволяют решать задачи аналитическими и численными методами. Объем решаемых этими системами задач, а также время и точность их решения таковы, что поражают воображение самого взыскательного пользователя. Эти математические системы, в общей совокупности, представляют собой непревзойденные средства решения самых сложных математических задач.

***Зарождение и развитие систем компьютерной алгебры.*** Эру создания компьютерной символьной математики принято отсчитывать с начала 60-х годов. Именно тогда в вычислительной технике возникла новая ветвь компьютерной математики, названная компьютерной алгеброй. Речь шла о возможности создания компьютерных систем, способных осуществлять типовые алгебраические преобразования: подстановки в выражениях, упрощение выражений, операции со степенными многочленами (полиномами), решение линейных и нелинейных уравнений и их систем, вычисление их корней и т. д. При этом предполагалась возможность получения аналитических (символьных) результатов везде, где это только возможно.

К сожалению, книги по этому направлению были способны лишь отпугнуть обычного читателя и пользователя компьютера от изучения возможностей компьютерной алгебры в силу перенасыщенности их узкоспециальным теоретическим материалом и весьма специфического языка описания. Материал таких книг, возможно, интересен математикам, занимающимся разработкой систем компьютерной алгебры, но отнюдь не основной массе их пользователей. Большинство же пользователей заинтересовано в том, чтобы правильно выполнить конкретные аналитические преобразования, вычислить в символьном виде производную или первообразную заданной функции, разложить ее в ряд Тейлора или Фурье, провести аппроксимацию и т. д., а вовсе не в детальном и сложном математическом и логическом описании того, как это делается компьютером (или, точнее, его программистом). Здесь та же си-

туация, что и с телевизором, радиоприемником или факсом: большинство из нас пользуются этими аппаратами, вовсе не интересуясь тем, как именно они выполняют свои довольно сложные функции. Это тем более важно в связи с тем, что предметные области, представляющие интерес для пользователя (будь он математик, физик, биолог или химик), перегружены своим собственным математическим аппаратом. Словом, большинству пользователей нужны системы компьютерной алгебры в качестве простого и удобного инструмента для работы.

Поняв эту истину, многие западные фирмы приступили к созданию компьютерных систем символьной математики, ориентированных на широкие круги пользователей, не являющихся профессионалами в компьютерной алгебре. Учитывая невероятно большую сложность автоматизации решения задач в аналитическом виде (число математических преобразований и соотношений весьма велико, и некоторые из них неоднозначны в истолковании), первые подобные системы удалось создать лишь для больших ЭВМ. Но затем появились и системы, доступные для мини-ЭВМ.

Заметное развитие получили языки программирования для символьных вычислений Reduce, система muMath для малых ЭВМ, а в дальнейшем — интегрированные системы символьной математики для персональных компьютеров: Derive, MathCAD, Mathematica, Maple V и др.

В бывшем СССР большой вклад в развитие систем символьной математики внесла школа академика Глушкова. В конце 70-х годов были созданы малые инженерные ЭВМ класса «Мир», способные выполнять аналитические вычисления даже на аппаратном уровне. Был разработан и успешно применялся язык символьных вычислений «Аналитик». Эти работы отчасти превосходили развитие систем символьной математики. К огромному сожалению, они появились слишком рано для своего времени и не соответствовали «генеральной линии» развития советской вычислительной техники в те годы. Уклон в сторону развития больших ЭВМ серии ЕС, навязанный в СССР компьютерными чиновниками, отодвинул компьютеры «Мир» на задний план, а затем этот класс компьютеров просто прекратил свое существование и развитие.

К сожалению, в отрыве от мировой науки и серьезных источников финансирования наши работы (за исключением некоторых теоретических) в области компьютерной алгебры оказались малоэффективными — отечественных систем компьютерной алгебры для персональных компьютеров, доведенных до серийного производства и мировой известности, так и не было создано (впрочем, как и конкурентоспособных ПК на нашей элементной базе). Зато множество наших специалистов — как математиков, так и программистов — эмигрировали на Запад и приняли участие, порой весьма серьезное, в разработке западных систем символьной математики.

***Системы символьной математики для персональных компьютеров.*** На Западе решающий скачок в компьютеризации общества произошел с началом массового производства и внедрения ПК. Долгое время их ограниченные возможности не позволяли реализовать на них серьезные системы сим-

вольной математики. Но к началу 90-х годов ситуация стала заметно меняться к лучшему. С одной стороны, аппаратные возможности ПК стали резко возрастать по мере быстрой смены поколений микропроцессоров. В итоге по скорости счета и объему оперативного запоминающего устройства (ОЗУ) ПК стали обходить «большие» ЭВМ класса ЕС, а сейчас оставили их далеко позади. Это создало реальные предпосылки к развертыванию работ по разработке систем компьютерной алгебры. Впрочем, надо помнить, что разрыв в производительности между новейшими ПК и многопроцессорными супер-ЭВМ и в наши дни остается поразительно большим!

Среди разработчиков математических систем долгое время бытовало мнение о вторичной роли пользовательского интерфейса и главенствующем значении математических возможностей таких систем. В результате в прошлом пользовательский интерфейс многих математических систем отличался ущербной простотой и архаичностью. С переводом таких систем на ПК с графическими операционными системами класса Windows с таким подходом пришлось решительно кончать. Цветная графика высокого разрешения современных ПК резко повысила не только роль графического представления данных вычислений, но и привела к слиянию пользовательского интерфейса математических систем с интерфейсом современных графических операционных систем, таких как Windows 3.1/3.11/95/98/NT/2000/XP. Образцом для подражания повсеместно стал интерфейс пользователя массовых офисных программ — Microsoft Office.

Наибольшую известность получили три класса систем символьной математики: созданная на базе языка искусственного интеллекта Mu Lisp малая система Derive, одна из самых мощных и поныне привлекательных систем Maple V (ядро написано на языке C) и системы Mathematica 1 и 2. Позже на базе ядра системы Maple V символьные вычисления были реализованы в популярных числовых системах Mathcad – версии Mathcad 3.0/4.0/5.0/Plus5.0/6.0/Plus 6.0/7.0/Plus7.0/8.0/8.0PRO/2000PRO/2000Premium имеют изумительный пользовательский интерфейс и возможности, улучшающиеся от версии к версии. Блок символьной математики на базе ядра Maple V был добавлен и в одну из самых крупных матричных систем — MATLAB.

Система Derive и поныне привлекательна своими невзыскательными требованиями к аппаратным ресурсам ПК — это единственная система, которая работает даже на ПК класса IBM PC XT без жесткого диска. Более того, при решении задач умеренной сложности она показала более высокое быстродействие и большую надежность решения, чем первые версии систем Maple V и Mathematica. Системе Derive трудно конкурировать с этими системами всерьез — ни по обилию функций и правил аналитических преобразований, ни по возможностям машинной графики и удобству пользовательского интерфейса.

Хотя новейшая версия Derive 7 под Windows уже имеет современный интерфейс, он все же во многом уступает изысканному интерфейсу своих маститых конкурентов. А по возможности графической визуализации резуль-



татов вычислений Derive все еще далеко отстает от них. То же можно сказать и о новой системе символьной математики MuPAD 1.4.

Система Maple V — патриарх в семействе систем символьной математики. И поныне это весьма привлекательная система для математика-аналитика и научного работника. Даже в среде MS-DOS Maple V имеет неплохой интерфейс и превосходно организованную обширную базу данных помощи. Полнота ядра системы, хранящего более 2700 математических функций (у последней реализации Maple 6 их уже свыше 3000!) и правил их преобразования, вполне заслуживает большого уважения. Весьма привлекательное свойство этой системы — подробная встроенная помощь и множество примеров ко всем встроенным в нее функциям и прикладным пакетам. Эти примеры легко скопировать в окно редактирования системы и тут же решить.

Достойна восхищения и математическая графика системы Maple, в частности возможность изображения пересекающихся трехмерных фигур с функциональной окраской. Новейшие системы Maple V для Windows (реализации R5 и R6) по возможностям графики стоят на одном уровне с системами Mathematica 3/4. Считается, что они несколько превосходят системы Mathematica в части символьных преобразований, но такое превосходство на сегодня уже является весьма спорным.

К сожалению, фирма Waterloo Maple Inc. (Канада) - разработчик системы Maple V — больше блистала математической проработкой своего проекта, чем уровнем его коммерческой реализации. В силу этого система Maple V была доступна в основном узкому кругу профессионалов. Сейчас эта фирма работает совместно с более преуспевающей в коммерции и проработке пользовательского интерфейса математических систем фирмой MathSoft, Inc. — создательницей весьма популярных и массовых систем для численных расчетов Mathcad, ставших международным стандартом для технических вычислений. Пока, однако, математические возможности этих систем в области компьютерной алгебры намного уступают системам Maple V, Mathematica 2 и даже малютке Derive (не говоря уже о реализациях Mathematica 3 и 4).

Появление новых версий Mathematica 3 и 4 вновь резко поднимает планку оценки качества систем компьютерной алгебры. Наступает новый этап интеграции математических систем как друг с другом, так и с современными текстовыми и табличными процессорами, такими как Word и Excel из офисных пакетов Microsoft Office.

Очень интересны последние реализации матричных систем MATLAB 5.2/5.3, но это очень громоздкая система, последняя реализация которой — MATLAB 5.3.1 - занимает на жестком диске 1500 Мбайт памяти (даже Mathematica 4 требует на порядок меньше места). Система MATLAB создана фирмой MathWorks (США).

Сейчас уже ясно, что конкурентоспособные отечественные системы символьной математики у нас, в силу известной экономической ситуации, в обозримом будущем не появятся. Это печальное положение делает особенно актуальным освоение нашими учеными, педагогами и учащимися новейших западных систем компьютерной алгебры. К таковым и относятся системы

класса Mathematica — признанные мировые лидеры среди систем символьной математики, ориентированных на персональные компьютеры.

Существование большого количества информационных систем проектирования и моделирования (ИСПРиМ) позволяют их подразделить на системы компьютерной математики, технического и имитационного моделирования (рис.1.1).

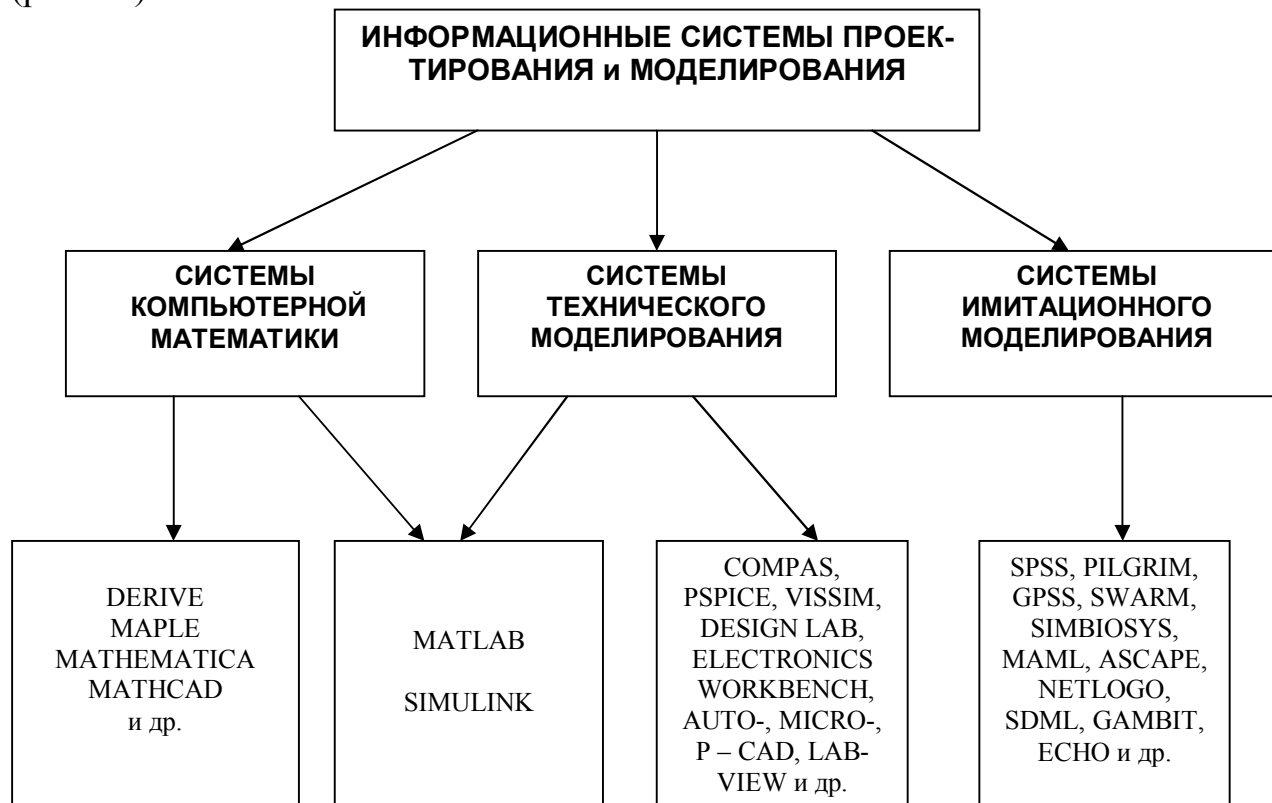


Рис. 1.1. Условная классификация информационных систем по типу решаемых задач

**Системы компьютерной математики.** К этим системам можно отнести пакеты Derive, Mathematica, MathCad, Maple, MatLAB и др. Эти пакеты разработаны различными фирмами и имеют свои особенности. Каждый из этих пакетов имеет свой интерфейс. В этих пакетах алгоритмизированы, систематизированы и заложены в виде процедур практически все известные методы аналитического и численного решения математических задач.

Все эти системы развиваются, в них вносятся дополнения, и разработчики этих систем предлагают новые модернизированные версии.

**Системы технического моделирования.** Наряду с развитием цифровых вычислительных машин формировалось направление аналоговых вычислительных машин (АВМ), с помощью которых решались различные физические и математические задачи. АВМ позволяли решать различные виды математических моделей, представленных в виде дифференциальных уравнений с помощью натурального схмотехнического моделирования. Аналоговые ЭВМ в настоящее время не разрабатываются. Однако появились технические информационные системы (компьютерные виртуальные конструкторы), в частности

Electronics Workbench, Simulink, Vissim, LabVIEW и др., решающие математические задачи с помощью схемотехнического моделирования.

Системы технического моделирования построены по принципу конструктора – из блоков. В системах технического моделирования можно решать как математические, так и инженерные задачи. В этих компьютерных системах можно собирать и конструировать виртуально любые электротехнические схемы с использованием компьютерных аналогов электротехнических и измерительных деталей, а также визуальное моделирование и конструирование инженерных, технических имитаторов электронных приборов и логических устройств. Более того, спроектированные и созданные виртуальные инженерные и производственные компьютерные объекты и установки можно использовать для натурального эксперимента и производственных испытаний в реальном масштабе времени.

**Системы имитационного моделирования.** В настоящее время активно разрабатываются системы имитационного моделирования: SimBioSys: C<sup>++</sup> - оболочки агентно-базового эволюционного моделирования в биологических и общественных науках; системы моделирования SWARM и его расширения MAML (Multi-Agent Modelling Language) для моделирования искусственного мира; пакеты Ascape (Agent Landscape) и RePast (Recursive Porous Agent Simulation Toolkit), написанные на платформе языка Java, для поддержки агентно-базового моделирования; информационные системы NetLogo и MI-MOSE (Micro- and Multilevel Modelling Software), предназначенные для создания имитационных моделей и технологий моделирования в общественных науках; SPSS, PilGrim, GPSS, Z-Tree для исследования экономических статистических явлений и процессов и др.

Знание и применение систем компьютерной математики, технического и имитационного моделирования позволяют модельщикам оперативно выбрать систему моделирования, построить адекватные модели, найти способы их решения, перейти к полномасштабному исследованию реального явления или процесса на модели, оценить решения моделей и представить поведение и закономерности изучаемого явления.

При компьютерном моделировании с помощью систем математического моделирования важен также субъективный фактор. Глубокое знание и освоение технологий математического моделирования в системах MathCAD, Maple, MatLAB и в других пакетах существенно влияет на оперативность решения математической модели реального объекта.

Таким образом, анализ показывает, что в настоящее время существует множество вариантов выбора методов и технологий компьютерного моделирования математических моделей. Освоение информационных систем и технологий математического и компьютерного моделирования предусмотрено Государственным образовательным стандартом многих специальностей и направлений, в частности: 010400 - Физика, 030100 Информатика, 030500.06 - Профессиональное обучение (информатика, ВТ и компьютерные технологии), 511800 - Математика, компьютерные науки, 080801- Прикладная информатика, 230201- Информационные системы.

Настоящее пособие состоит из десяти глав. Первая глава посвящена технологиям моделирования в офисной программе Excel. Использовать систему Excel офисного приложения Windows имеет смысл, если у исследователя на компьютере нет какой-нибудь из систем компьютерной математики. Применение системы Excel оправдано и с методической точки зрения. В частности, если же пользователь научится согласно алгоритму вычислять значения неизвестных функций на первом шаге итерации при численном моделировании, то он будет обладать знаниями и навыками составления алгоритма решения дифференциальных уравнений.

Вторая глава посвящается системе компьютерной алгебры Derive. Эта система играет важную роль при освоении основ компьютерного моделирования и систем компьютерной алгебры начального уровня. Она ориентирована на решение математических задач для школы и начальных курсов вузов. В этой главе приведено множество примеров по аналитическому и численному решению дифференциальных уравнений. В приложении 1 приведен практикум, где рассмотрено большое количество задач. Приведенные примеры решения позволят освоить технологию вычислений в системе Derive.

В следующих четырех главах рассмотрены системы компьютерной математики Maple, MathCAD, Mathematica, MatLAB. Здесь основное внимание уделено решениям систем дифференциальных уравнений как аналитическими, так и численными методами.

Последующие главы посвящены системам технического моделирования Vissim, Simulink, Electronics Workbench, LabVIEW. Эти системы предназначены для инженерных специальностей. Изложенные технологии компьютерного моделирования доступны пользователю среднего уровня подготовки по физике и математике, так как это демонстрируется на достаточно простых и доступных примерах. В приложении 2 приведен практикум по изучению основ электроники и узлов персонального компьютера.

В каждой главе описаны общие сведения об информационной системе и рассмотрены технологии компьютерного моделирования на задачах физики, информатики и математики.

Изучить в полной мере все системы компьютерного моделирования и технологии достаточно сложно в связи с ограниченностью по времени, однако знать об этих информационных системах и уметь использовать их в профессиональной деятельности – необходимое условие компетентности специалиста в соответствующей области знаний.

## ГЛАВА I. МОДЕЛИРОВАНИЕ ОБЪЕКТОВ В СИСТЕМЕ EXCEL

Электронные таблицы представляют собой класс специальных программ для ведения документации. Документ изображается на экране в виде таблицы, у которой именованы строки и столбцы. Каждая клетка может содержать текст, числа или формулу. С содержимым клеток можно производить арифметические, алгебраические и логические операции. Изменение содержимого одной из клеток автоматически ведет к изменению содержимого других клеток, связанных с ним логически или формулой. Таким образом, обработка данных происходит автоматически, результат получаем в виде готовых таблиц. При необходимости результат можно получить в виде графиков или же диаграмм.

Использовать систему Excel офисного приложения Windows для компьютерного моделирования имеет смысл, если у исследователя нет на компьютере какая-нибудь из систем компьютерной математики. Применение системы Excel оправдано и с методической точки зрения, она позволяет глубже понять пошаговую работу алгоритма вычислений для разностных уравнений. Если же пользователь научится согласно алгоритму вычислять значения неизвестных функций на первом шаге итерации, то он будет понимать работу алгоритма решения.

Технология решения (моделирования) физических объектов с помощью пакета Excel изложена также в работах [1, 2, 3].

### 1.1. Моделирование движения небесного тела под действием сил тяготения

Рассмотрим технологию моделирования в системе Excel на примере движения материального тела (планета) массой  $m$  под действием притягивающего неподвижного центра (Солнца). Например, любое материальное тело движущееся под действием силы притяжения небесного тела (в частности, поля тяготения Солнца).

**Задание на моделирование.** Построить орбиту малой планеты по ее координатам, рассчитанным с интервалом в 5 суток, если в перигелии она находится на расстоянии 0,5 а.е. от Солнца и имеет скорость 0,026 а.е./сут. Считать, что планета движется под действием притяжения только со стороны Солнца. Влияние других планет не учитывать.

**Дифференциальные уравнения движения** могут быть получены из закона Ньютона:

$$m \frac{d^2 \mathbf{r}}{dt^2} = \mathbf{F}_T$$

На (рис. 4.1) показана материальное тело и действующие на нее силы:  $\mathbf{F}_T = -G \frac{Mm}{r^3} \mathbf{r}$  сила тяжести направленная к центру  $O$ . Найдя проекции на координатные оси всех сил, действующих на материальную точку, получаем



следующую систему дифференциальных уравнений:

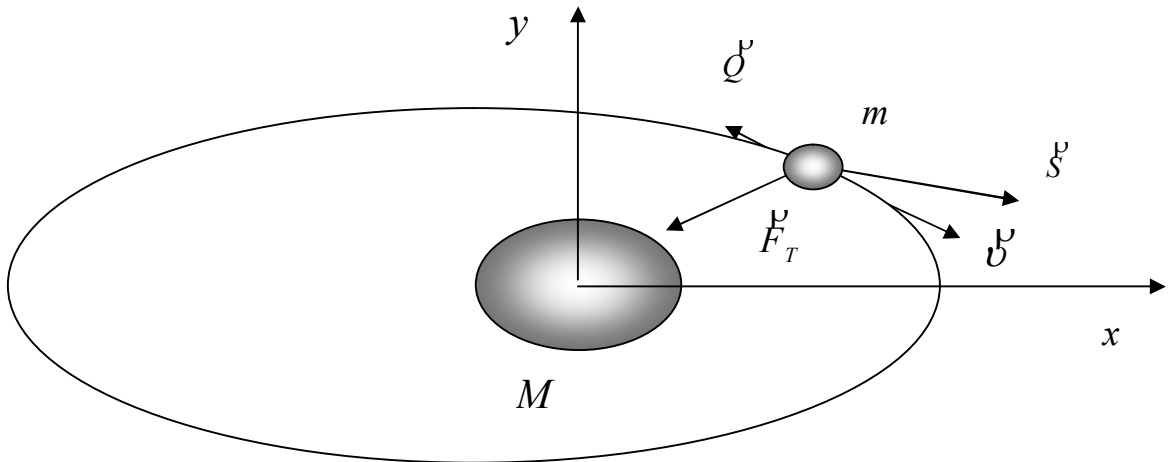


Рис. 4.1

$$\begin{aligned} \frac{dv_x}{dt} &= -GM \frac{x}{(x^2 + y^2)^{3/2}}, \\ \frac{dv_y}{dt} &= -GM \frac{y}{(x^2 + y^2)^{3/2}}, \\ \frac{dx}{dt} &= v_x, \quad \frac{dy}{dt} = v_y. \end{aligned} \quad (1.1)$$

с общими для всех случаев начальными условиями

$$v_x(0) = 0, x(0) = x_0, v_y(0) = v_0, y(0) = 0,$$

предполагающими, что материальное тело в начальный момент времени лежит на оси  $Ox$  в точке  $x_0$ , не имеет соответствующей скорости вдоль оси  $Ox$  и имеет начальную скорость лишь вдоль оси  $Oy$ , которая равна  $v_0$ .

**Решение.** Решение данной задачи средствами программирования приведены в [2, 3]. Рассмотрим формулы, позволяющие производить вычисления по решению уравнений (1.1) методом Эйлера

$$\begin{aligned} T_I &= T_{I-1} + DT; \\ R_I &= \text{КОРЕНЬ}(X_I * X_I + Y_I * Y_I); \\ A_{I, X} &= -GM * X_I / R_I^3; \\ A_{I, Y} &= -GM * Y_I / R_I^3; \\ V_{I, X} &= V_{I-1, X} - A_{I-1, X} * DT; \\ V_{I, Y} &= V_{I-1, Y} - A_{I-1, Y} * DT; \\ X_I &= X_{I-1} + V_{I-1, X} * DT; \\ Y_I &= Y_{I-1} + V_{I-1, Y} * DT; \end{aligned}$$

Приведенные формулы представляют собой готовую схему алгоритма для вычислений.

Введем свою систему единиц. Время измеряется в сутках, расстояние в а.е. (астрономических единицах), за единицу массы принята масса Солнца.

При таком выборе единиц числовые значения исходных данных таковы:  
 $X_0=0,5$ ;  $Y_0=0$ ;  $V_{0,x}=0$ ;  $V_{0,y}=0,026$ ;  $-GM=-1/58^2$ ;  $DT=1$ .

Заполняем электронную таблицу.

Первоначальный вид электронной таблицы приведен на рис. 4.1.

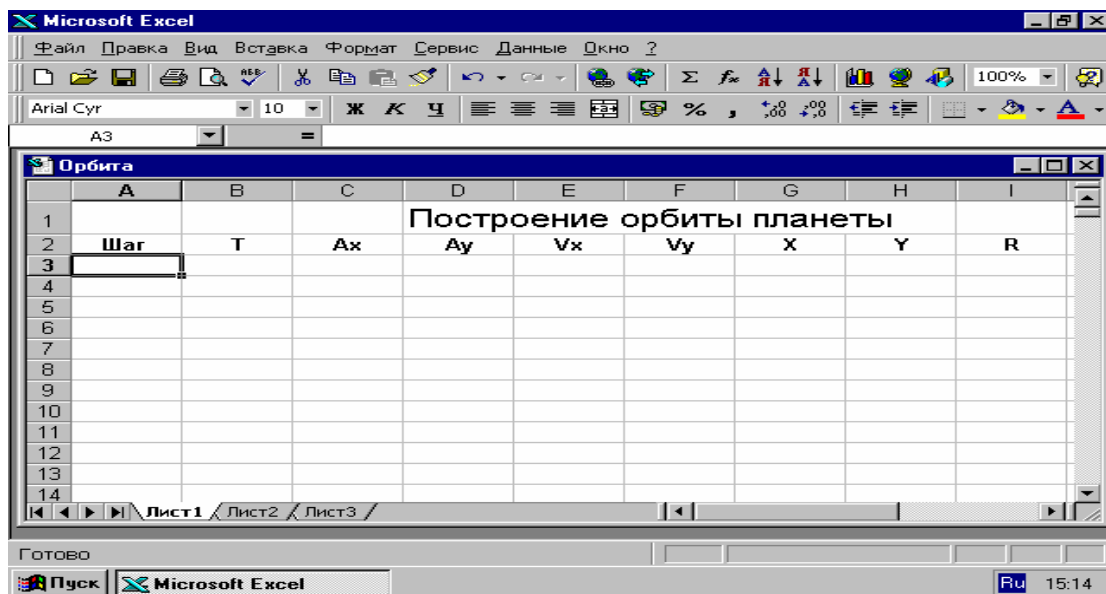


Рис. 4.2

Пояснения к заполнению электронной таблицы

1. В ячейку A3 внесено 1.
2. В ячейку A4 введена формула  $=A3+1$ .
3. В ячейку B3 внесено 0.
4. В ячейку B4 введена формула  $=B3+1$ .
5. В ячейку C3 введена формула  $=(-1/(58^2))*G3/(I3^3)$ , которая скопирована в ячейку C1.
6. В ячейку D3 введена формула  $=(-1/(58^2))*H3/(I3^3)$ , которая скопирована в ячейку D1.
7. В ячейку E3 внесено значение начальной скорости по x равная 0.
8. В ячейку E4 введена формула  $=E3+C3*1$ .
9. В ячейку F3 внесено значение начальной скорости по y равная 0,026.
10. В ячейку F4 введена формула  $=F3+D3*5$ .
11. В ячейку G3 внесено начальная координата по x равная 0,1.
12. В ячейку G4 введена формула  $=G3+E4*1$ .
13. В ячейку H3 внесено начальная координата по y равная 0.
14. В ячейку H4 введена формула  $=H3+F4*1$ .
15. В ячейку I3 введена формула  $=КОРЕНЬ(G3^2+H3^2)$ , которая скопирована в ячейку I1.
16. В ячейку J3 введена формула  $=КОРЕНЬ(C3^2+D3^2)$ , которая скопирована в ячейку J1.

Таким образом, мы заполнили две строки. В третьей строке введены начальные значения, в строке №4 формулы расчета. Выделяя строку №4 и рас-

тягивая мышью по вертикали, получаем расчетные значения на каждый момент времени.

Результаты вычислений приведены в таблице 2.

Таблица 2

	A	B	C	D	E	F	G	H	I	K
1										
2	Шаг	T	Ax	Ay	Vx	Vy	X	Y	R	A
3	1	0	-0,00119	0	0	0,026	0,5	0	0,5	0,001189
4	2	5	-0,0012	-0,00033	-0,00595	0,026	0,470273	0,13	0,487911	0,001249
5	3	10	-0,00109	-0,00067	-0,01196	0,024336	0,410458	0,251682	0,481476	0,001282
6	4	15	-0,00086	-0,00095	-0,01743	0,020985	0,323313	0,356607	0,481352	0,001283
7	5	20	-0,00055	-0,00112	-0,02174	0,016232	0,214624	0,437769	0,48755	0,001251
8	6	25	-0,00022	-0,00117	-0,02449	0,010618	0,092172	0,49086	0,499439	0,001192
9	7	30	7,75E-05	-0,00111	-0,02559	0,004762	-0,03578	0,514669	0,515911	0,001117
10	8	35	0,000313	-0,00099	-0,0252	-0,00081	-0,16179	0,510624	0,535643	0,001036
-	-	-	-	-	-	-	-	-	-	-
-	15	70	0,000649	-1,2E-05	-0,00488	-0,01913	-0,67652	0,012422	0,676639	0,000649
17										
18	16	75	0,00062	7,56E-05	-0,00163	-0,01919	-0,68467	-0,08351	0,689745	0,000625
19	17	80	0,000587	0,000154	0,001472	-0,01881	-0,67731	-0,17755	0,700196	0,000606
20	18	85	0,000549	0,000224	0,004404	-0,01804	-0,65529	-0,26774	0,707877	0,000593

Для построения орбиты планеты выделяем ячейки G2:H20. Из мастера диаграмм выбираем тип диаграммы “Точечная”, далее “Точечная диаграмма со значениями, соединенными сглаживающими линиями без маркеров”. В заголовке указываем название графика “Траектория орбиты”. Полученная траектория орбиты приведена на рис. 4.3.

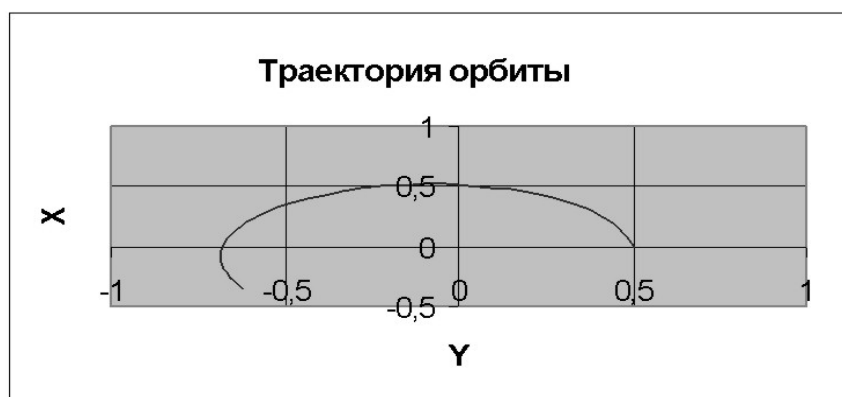


Рис. 4.3

Чтобы получить график удаления планеты в различные моменты итерации, выделяем ячейки I2:I20. Из мастера диаграмм выбираем тип диаграммы “Лепестковая”, далее “Лепестковая диаграмма является аналогом графика в полярной системе координат ...”. В заголовке указываем название диаграммы “Удаление планеты от Солнца”. Полученный график приведен на рис. 4.4.

Для построения графика зависимости ускорения от времени выделяем ячейки C2:D20. Из мастера диаграмм выбираем тип диаграмм “График”, далее “График отображает развитие процесса во времени или по категориям”. Выбираем “далее”, выпадает меню “диапазон данных”. В меню “диапазон

данных” выбираем “ряд”. В “подписи оси X” набираем =Лист1!\$B\$2:\$B\$20. Указатель мыши подводим к “добавить” и щелкаем левой кнопкой мыши, в “имя” набираем – A, в “значения” набираем =Лист1!\$I\$2:\$I\$20. Указатель мыши подводим к “далее” и щелкаем левой кнопкой мыши. В подписи данных вводим название диаграммы “График зависимости ускорения от времени”. Указатель мыши подводим к “готово”, щелкаем левой кнопкой мыши. График готов. Рис. 4.5. Увеличение количества шагов итерации дает более полное решение задачи.



Рис. 4.4.



Рис. 4.5

## 1.2. Движение тела в поле силы тяжести Земли

Использование электронных таблиц при моделировании физических процессов не единственная область. Применение электронных таблиц при проведении расчетов данных лабораторных работ позволяет сделать основной упор на постановку опыта (эксперимента) и анализ полученных результатов, т.е. формированию навыков, необходимых физику-экспериментатору.

В качестве примера приведем решение задачи «Движение тела в поле силы тяжести Земли».

**Постановка задачи.** Свинцовый шарик массой  $m = 0,54$  г падает в масле (плотность свинца  $\rho_c = 11,3 \cdot 10^3$  кг/м<sup>3</sup>, плотность масла  $\rho_m = 0,92 \cdot 10^3$  кг/м<sup>3</sup>, вязкость масла  $\mu_m = 0,95$  кг/(м·с)) с начальной скоростью  $v_0$ ,

направленной по вертикали. Постройте графики скорости, ускорения и перемещения шарика от времени. Варьируя  $v_0$ , установите с помощью компьютерной модели характерные особенности движения шарика.

**Математическая модель.** Будем считать, что на движущийся шарик в газовой или жидкой среде действуют сила трения, тяжести и архимедова сила. Величина силы сопротивления среды при малых скоростях пропорциональна скорости и имеет вид

$$F_{\text{сопр}} = k_1 v, \quad (1.2)$$

Для шарика  $k_1 = 6\pi\mu r$ , где  $\mu$  - динамическая вязкость среды,  $r$  - радиус шарика. Сила Архимеда  $F_a = \rho_{\text{сп}} V g = \rho_{\text{сп}} \pi r^3 g/3$

Закон Ньютона в направления движения шарика имеет вид

$$m \frac{d^2 y}{dt^2} = mg - F_c - F_A$$

с учетом (1.2) и силы Архимеда получим уравнения движения шарика в вязкой среде

$$\begin{aligned} \frac{dv}{dt} &= g - k_1 v / m - \rho_c V g / m, \\ \frac{dy}{dt} &= v \end{aligned} \quad (1.3)$$

Эти уравнения и составляют математическую модель движения тела в вязкой среде.

**Решение.** Введем обозначения и присвоим им значения и вычислим некоторые вспомогательные величины.

Обозначение величин и присваивание им значений

N	A	B	C	D	E	F	G	
1	$\rho_c =$	1,13E+04	кг/м <sup>3</sup>	$\mu_{\text{Дин Вязк}} =$	0,95	кг/(мс)	$m =$	5,40E-04 кг
2	$\rho_m =$	9,50E+02	кг/м <sup>3</sup>	$V_0 =$	-0,5	м/с		
3	$g =$	9,81	м/с <sup>2</sup>	$Dt =$	0,001	с		

Вычисление радиуса шарика, коэффициента сопротивления  $k_1 = 6\pi\mu r$ , объема шарика  $V_{\text{шар}} = m/\rho_c$  проведены в ячейках B4, B5 и E4

4	$r =$	0,00225117	м	$V_{\text{шар}} =$	4,778E-08	м <sup>3</sup>	
5	$k =$	0,04031193	кг/с				

Заполним шестую строку таблицы данных начальными данными. В седьмой строке получим первые значения скорости и ускорения. Остальные значения получаем, растягивая мышью, выделенную седьмую строку первых значений данных по вертикали.

ТАБЛИЦА ДАННЫХ						
N	dt, с	V, м/с	dV, м/с	y, м	dy, м/с	a, м/с <sup>2</sup>
6	0	-0,5	0,047052	0	-0,0005	4,63E+01



7	0,001	-0,45295	0,043539	-0,0005	-0,00045	4,28E+01
8	0,002	-0,40941	0,040289	-0,000952948	-0,00041	3,95E+01
9	0,003	-0,36912	0,037281	-0,001362357	-0,00037	3,65E+01
---	---	---	---	---	---	---
48	0,048	0,115072	0,001136	-0,001985586	0,000115	3,95E-01
49	0,049	0,116208	0,001051	-0,001870514	0,000116	3,10E-01
50	0,05	0,117258	0,000972	-0,001754306	0,000117	2,32E-01

Построение графиков можно провести аналогично предыдущему примеру

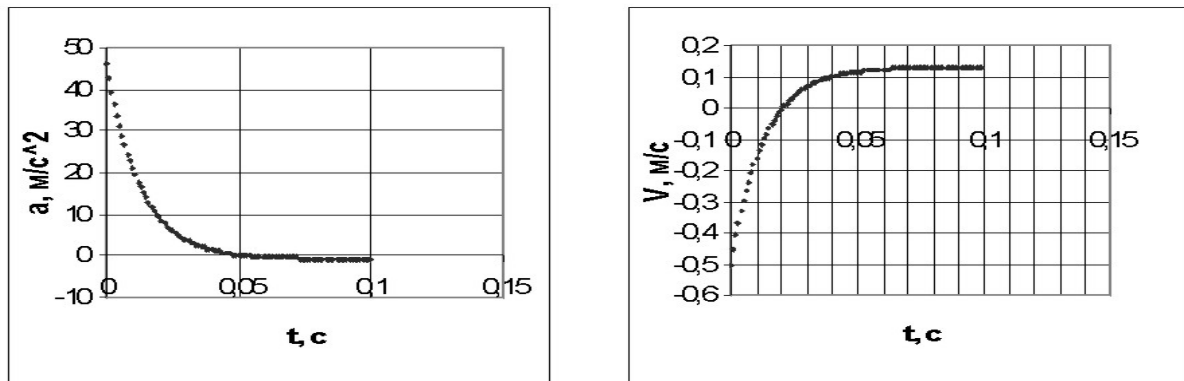


Рис. 4.6. Графики зависимости ускорения и скорости от времени

### 1.3. Движение заряженной частицы в кулоновском поле

**Постановка задачи и математическая модель.** Движение заряженной частицы с зарядом  $q$  и массой  $m$  в кулоновском поле другой частицы с зарядом  $Q$ , положение которой зафиксировано. Математическую модель можно получить из закона Ньютона

$$\vec{F} = m \frac{d\vec{v}}{dt} \quad (1.4)$$

и закона Кулона.

$$\vec{F} = - \frac{1}{4\pi\epsilon_0} \frac{Qq}{r^3} \vec{r} \quad (1.5)$$

В системе координат, начало которой привязано к телу с большой массой и зарядом  $Q$ , уравнения модели имеют вид:

$$\begin{aligned} \frac{dv_x}{dt} &= - \frac{Qq}{m} \frac{x}{(x^2 + y^2)^{3/2}}, \\ \frac{dv_y}{dt} &= - \frac{Qq}{m} \frac{y}{(x^2 + y^2)^{3/2}}, \\ \frac{dx}{dt} &= v_x, \quad \frac{dy}{dt} = v_y. \end{aligned} \quad (1.6)$$

Начальные условия определяются двумя параметрами: начальной скоростью и углом  $\alpha$ .

$$\begin{aligned} x(0) &= x_0, \quad y(0) = 0, \\ v_{x0} &= v_0 \cos \alpha, \quad v_{y0} = v_0 \sin \alpha, \end{aligned} \quad (1.7)$$

**Решение.** Задачу решим методом Эйлера. Формулы, по которым проводятся вычисления, имеют вид

$$\begin{aligned} T_I &= T_{I-1} + H; \\ X_I &= X_{I-1} + H * V_{I-1, X}; \\ Y_I &= Y_{I-1} + H * V_{I-1, Y}; \\ R_I &= \text{КОРЕНЬ}(X_I * X_I + Y_I * Y_I); \\ A_{I, X} &= -q * Q * X_I / R_I^3; \\ A_{I, Y} &= -q * Q * Y_I / R_I^3; \\ V_{I, X} &= V_{I-1, X} - A_{I-1, X} * H / M; \\ V_{I, Y} &= V_{I-1, Y} - A_{I-1, Y} * H / M; \end{aligned}$$

Введем обозначения и присвоим им значения и вычислим некоторые вспомогательные величины.

1. В ячейке С1 задано значение первоначального угла =B2\*3,14/180

1. В ячейках В2-В6 заданы входные параметры

1. В ячейке А11 задано начальное значение координаты X

1. В ячейке В11 задано начальное значение координаты Y

1. В ячейке С11 вычислено =B3\*COS(C2)

6. В ячейке D11 вычислено =B3\*SIN(C2)

Вычисления производятся в строке №12 по формулам

7. В ячейке А12 по формуле =A11+C11\*B\$7

5. В ячейке В12 по формуле =B11+D11\*B\$7

9. В ячейке С12 по формуле

$$=C11+(A11*B\$4*B\$5*B\$7)/(B\$6*\text{КОРЕНЬ}((A11^2+B11^2))^3)$$

10. В ячейке D12 по формуле

$$=D11+(B11*B\$4*B\$5*B\$7)/(B\$6*\text{КОРЕНЬ}((A11^2+B11^2))^3)$$

Остальные значения координат и скоростей получаем, растягивая мышью выделенную строку №12 значений данных по вертикали.

	A	B	C	D
1	Угол	1	0,017444	
2	скорость	10		
3	заряд 1	79		
4	заряд 2	2		
5	масса	4		
6	шаг	0,001		
7				
<b>10</b>	<b>x(i)</b>	<b>y(i)</b>	<b>vx(i)</b>	<b>vy(i)</b>
11	-1	0	0,999848	0,017444
12	-0,9995	8,72E-06	0,980098	0,017444

13	-0,99901	1,74E-05	0,960328	0,017444
14	-0,99853	2,62E-05	0,940539	0,017444
15	-0,99806	3,49E-05	0,920731	0,017445
16	-0,9976	4,36E-05	0,900904	0,017445
17	-0,99715	5,23E-05	0,881059	0,017446
18	-0,99671	6,11E-05	0,861196	0,017447
19	-0,99628	6,98E-05	0,841315	0,017448

Для построения траектории движения выделяем ячейки A11:A50 и B11:B50. Из мастера диаграмм выбираем тип диаграммы “Точечная”, далее “Точечная диаграмма со значениями, соединенными сглаживающими линиями без маркеров”. Полученная траектория движения заряженной частицы в кулоновском поле ядра приведена на рис.4.7.

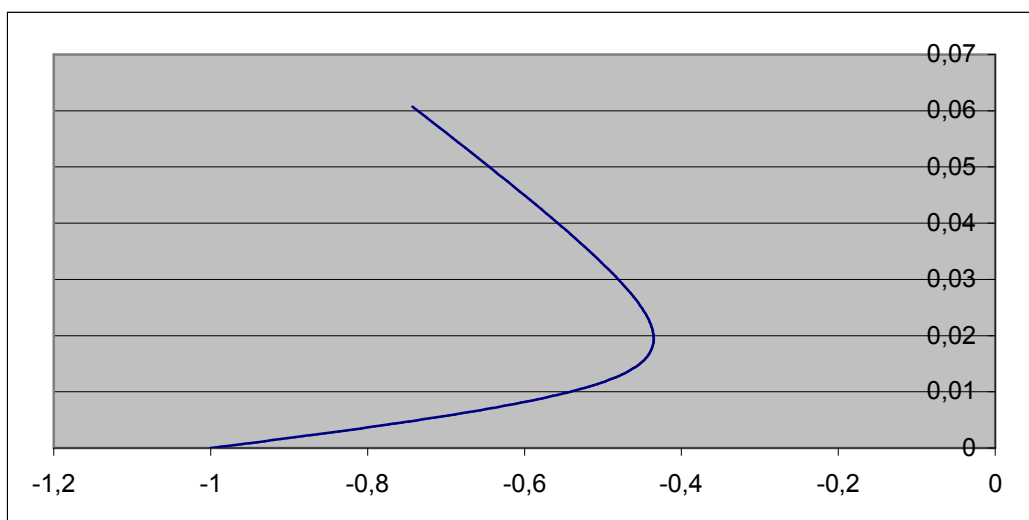


Рис. 4.7.

Аналогично можно использовать электронные таблицы при моделировании задач, приведенных в книгах [3, 4].

## ГЛАВА 2. МОДЕЛИРОВАНИЕ ОБЪЕКТОВ В СИСТЕМЕ DERIVE

Данная глава посвящена описанию основ работы с системой Derive при решении математических задач. В ней излагаются: алфавит системы, арифметические операторы, правила ввода, математических выражений, встроенные функции и функции пользователя. Основное внимание уделяется изложению функций и команд решения математических задач, а также организации диалога пользователя с системой Derive. Здесь рассмотрены не только функции системы и примеры их реализации, как это делается в большинстве книг, посвященных описанию универсальных программных средств символьной математики, но также методы, алгоритмы и компьютерные технологии решения математических задач. Это избавляет пользователя от возможных ошибок при решении задач с помощью функций данной системы.

В настоящее время нет стандарта на компьютерные технологии решения задач. Каждая система компьютерной алгебры имеет свой язык общения с пользователем.

При решении математических задач таким языком является алфавит, арифметические операторы, правила ввода математических выражений, набор функций и команд выполнения математических операций.

Система Derive обладает следующими особенностями:

- относительно проста в изучении;
- поддерживает интерактивный режим общения (запрос-ответ) пользователя с персональным компьютером (ПК);
- не предъявляет высоких требований к типу и техническим характеристикам компьютера (реализуется практически на любом современном ПК);
- обладает элементами интеллектуальной системы, при решении задач символьной математики;
- обеспечивает высокую производительность;
- устойчива к ошибкам вычислительного процесса;
- система Derive широко используется в образовательных учреждениях многих стран мира как современный инструмент автоматизации математических вычислений;
- применение системы помогает учащимся лучше осмысливать теоретические и прикладные стороны изучаемых дисциплин;
- обеспечивает высокую достоверность решения задач;
- самая дешевая из всех универсальных математических систем данного типа.

Опыт показывает, что Derive, по сравнению с другими системами, более легко внедряется в учебный процесс при изучении дисциплин широкого профиля — от естественно-научного цикла до специальных дисциплин.

Перечисленные достоинства Derive не дают основания утверждать, что это самая лучшая математическая система. Она, как и всякая другая система, имеет и недостатки.

К недостаткам системы можно отнести следующие:

- довольно слабые графические возможности;
- не поддерживается анимация;
- поддерживается небольшое количество операций (несколько сот), относительно других систем, например, таких как Maple (около 3000 операций), Mathematica (около 1000).

Достоинства и недостатки системы во многом субъективные. Их почувствует пользователь при решении практических задач или примеров, которых в книге достаточно много.

## 2.1. Общие сведения о системе

Derive — самая известная **система компьютерной математики** начального уровня. Она ориентирована на решение математических задач для школы и начальных курсов вузов. При запуске Derive появляется основное окно с заставкой системы (рис. 2.1). В основном окне видны строка заголовка, строка меню, панель инструментов и строка состояния системы (в нижней части окна).

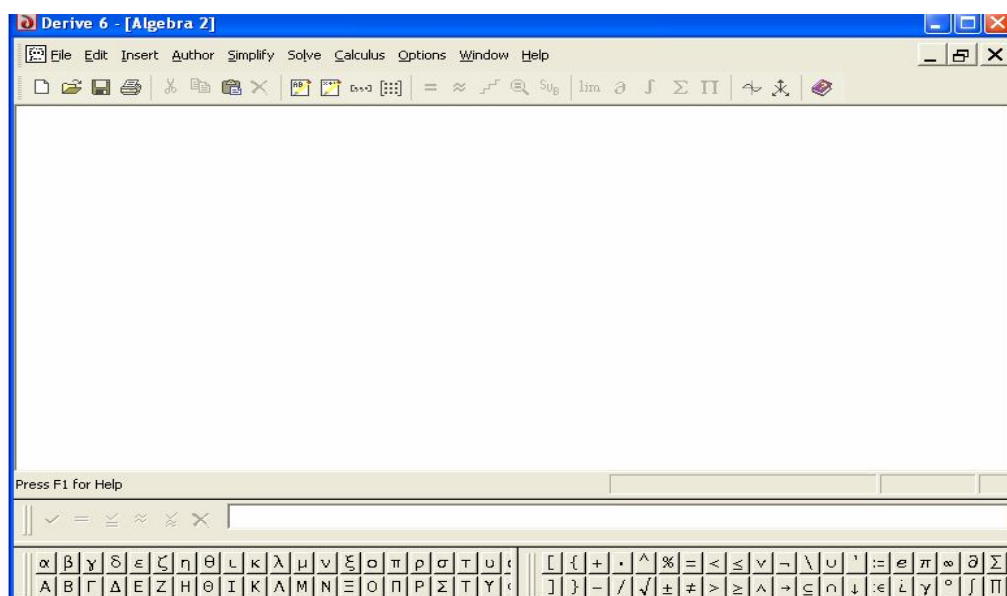


Рис.2.1. Основное окно системы Derive

Вычисления идут по схеме: “задал вопрос — получил ответ”. Для ввода математических выражений предназначено диалоговое окно Autor Expression (рис.2.2), которое открывается **командой** Author > Expression.

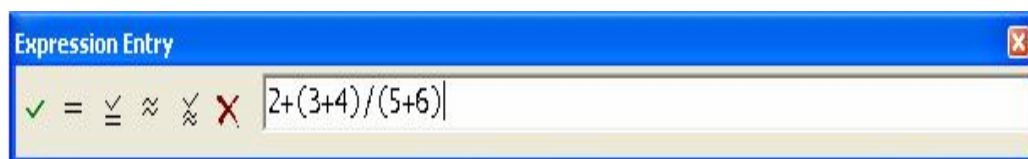


Рис.2.2. Диалоговое окно ввода выражений



Если после ввода выражения щелкнуть на кнопке Enter, выражение появится в окне выражений (рис.2.3) и будет выделено цветом — белым на сером фоне. Если после ввода выражения щелкнуть на кнопке Simplify, выражение будет вычислено и в окне выражений появится результат. Щелчок на кнопке Delete позволяет очистить диалоговое окно ввода выражений. Все введенные выражения и результаты их вычислений располагаются на отдельных строках и последовательно нумеруются символами вида #N, где N — номер строки.

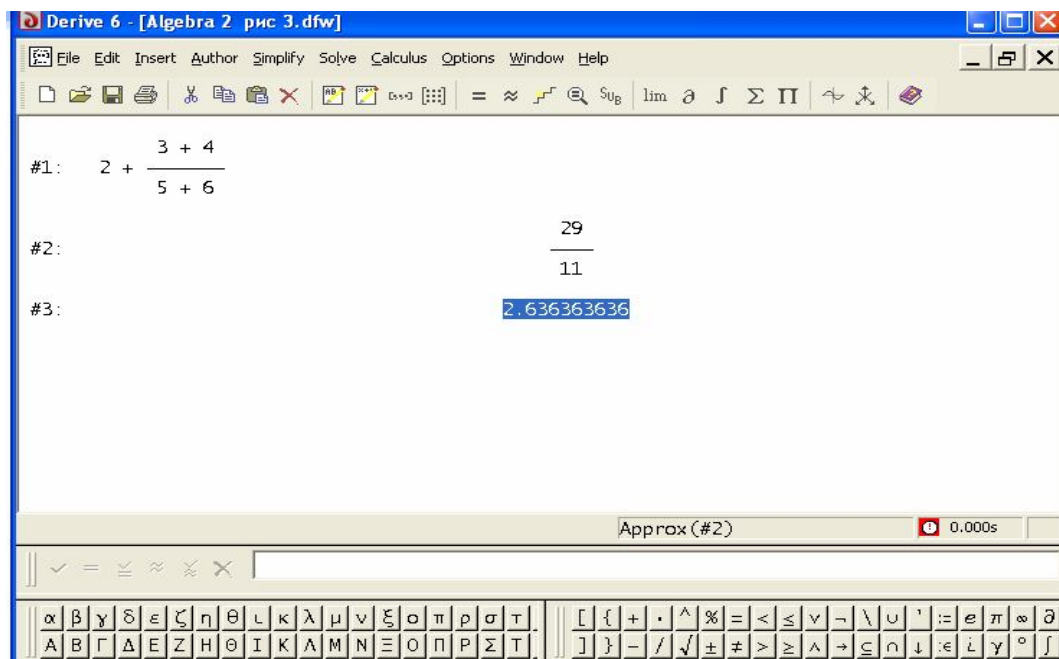


Рис.2.3. Пример ввода и вычисления арифметического выражения в символьном и численном виде

Для вычисления выражений в системе Derive имеется ряд других команд. Это команды Basic (результат в символьном виде), Display step (результат в упрощенном виде) и Approximate (результат в численном виде) меню Simplify. При выборе этих команд появляются диалоговые окна, в которых достаточно щелкнуть на кнопке ОК.

**Строка меню и интерфейс пользователя.** Интерфейс пользователя системы Derive позволяет выполнять большинство вычислений вообще без программирования, используя только команды меню. Строка меню (в версиях под MSDOS она располагается снизу) помимо обычных меню File, Edit, Window и Help имеет следующие специфические:

- Author — ввод математических выражений для их последующего преобразования;
- Insert – ввод объекта;
- Solve — решение математических уравнений;
- Calculus — вычисление производных, интегралов, пределов функций, сумм, произведений и разложение в ряд Тейлора;
- Options — задание различных параметров системы.

Строка меню обеспечивает доступ ко всем командам системы. Набор позиций может модифицироваться в зависимости от текущего состояния системы. Некоторые команды меню при этом могут быть недоступными или вообще отсутствовать. На панели инструментов размещен ряд кнопок для быстрого доступа к некоторым командам. Они перечислены ниже.

- Команды для работы с файлами:
  - New — открытие нового окна;
  - Open — вывод окна загрузки файла;
  - Save — запись файла под текущим именем.
- Команда печати содержимого окна с выражениями:
  - Print — вывод окна печати содержимого окна с выражениями.
- Команды ввода:
  - Insert text – ввод текста;
  - Author expression — ввод математических выражений;
  - Author vector — задание векторов указанного размера;
  - Author matrix — задание матриц указанного размера.
- Команды вычислений:
  - Simplify — упрощение выражений;
  - Approximate — вычисление выражений в числовой форме;
  - Display Step – вывод упрощенного выражения;
  - Solve — решение уравнений и неравенств (и систем с ними);
  - Substitute for variable — подстановки для переменных.
- Команды специальных вычислений:
  - Calculate limit — вычисление пределов функций;
  - Calculate derivative — вычисление производных функций;
  - Calculate integral — вычисление неопределенных и определенных интегралов;
  - Calculate sum — вычисление суммы рядов;
  - Calculate product — вычисление произведения рядов.
- Команды вывода графических окон:
  - 2D-plot window (Ctrl+2) — вывод окна двумерной графики;
  - 3D-plot window (Ctrl+3) — вывод окна трехмерной графики.

В нижней части окна Derive расположена строка состояния. В этой строке выводятся текстовые комментарии о состоянии выделенных выражений. На них надо обратить особое внимание, поскольку по этим комментариям можно судить о том, какие действия выполнялись над выражениями и из каких других выражений они получены. В окне выражений этого не видно.

**Редактирование выражений и документов.** Основные команды редактирования сосредоточены в меню Edit.

- Expression — вывод окна редактирования выделенного выражения;
- Annotation — вывод окна комментария для выделенного выражения;

Если в окне выражений выделено какое-либо выражение цветом, команда Expression доступна и ее выбор ведет к появлению описанного ранее окна Author Expression (см. рис. 2). В нем имеется панель специальных символов и строка редактирования. Каждое выражение может снабжаться текстовым

комментарием. Он вводится с помощью простого окна с полем ввода. Комментарий отображается в строке состояния, если выражение выделено.

Команды следующей группы служат для работы с буфером обмена:

- Copy Expressions — копирование выделенного выражения в буфер обмена;
- Select All — выделить все;
- Mark and Copy — выделение и копирование выражения.

**Ввод данных.** В меню Author представлены команды, необходимые для ввода данных:

- Expression — ввод математических выражений;
- Vector — задание вектора указанных размеров;
- Matrix — задание матрицы указанных размеров;
- Variable Value — задание величины переменной;
- Variable Domain — задание величины в интервале;
- Function Definition — определение функции.

Команда Expression, которая служит для ввода выражений, что уже описывалось (см. рис. 2.4). Для задания вектора используется команда Vector, которая вызывает появление окна для ввода единственного параметра — размера вектора. После ввода размера вектора появляется окно для ввода значений элементов вектор (рис.2.5), которых может быть от 1 до 100. Если размер вектора достаточно велик, то справа от полей ввода появляется вертикальная полоса прокрутки.

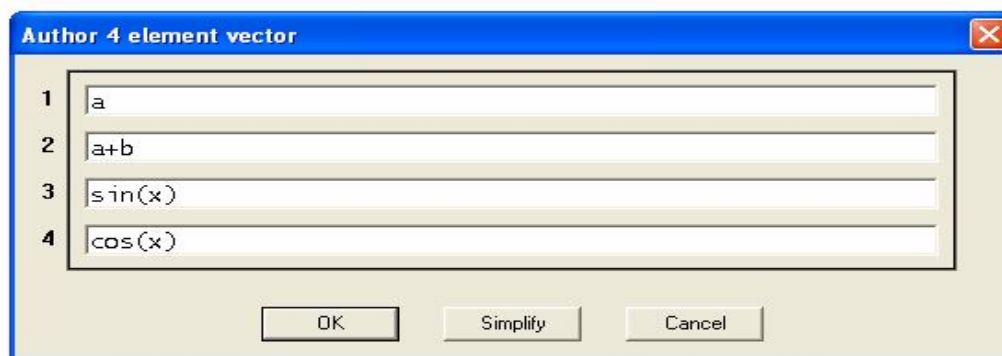


Рис. 2.4. Ввод элементов вектора

**Матрицей** называют двухмерную таблицу из  $m$  строк и  $n$  столбцов. Эти целые положительные числа определяют размер матрицы  $m \times n$ . Для задания матрицы надо выполнить команду Matrix и в появившемся окне ввести значения  $m$  и  $n$  (от 1 до 100) в поля Rows (строки) и Columns (столбцы). После ввода размера матрицы появляется окно для ввода элементов матрицы (рис. 6.). Заметим, что матрица с размером  $1 \times 1$  является единичным элементом, а матрица с одной строкой ( $m=1$ ) — вектором. Вектор может располагаться и вертикально, если  $m > 1$ , а  $n=1$ .

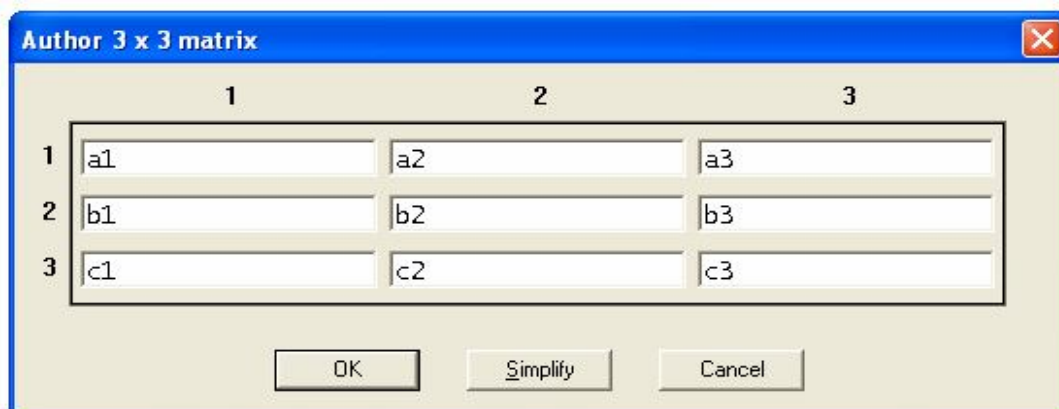


Рис.2.5. Окно для ввода элементов матрицы

Если размер матрицы достаточно велик, в окне ввода появляются вертикальные и горизонтальные полосы прокрутки. Элементы матриц могут быть любыми числами именами переменных и символьными выражениями (формулами).

Команда Variable Value служит для ввода (присвоения) величины какой-либо переменной. Для задания величины используется команда Variable Value, которая вызывает появление окна для ввода двух параметров — названия переменной и ее величины во второй строке (рис.2.6).

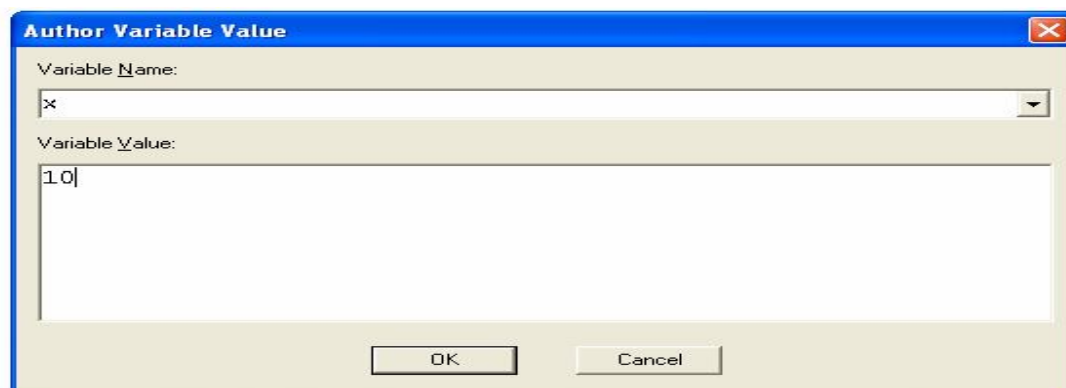


Рис.2.6. Окно для ввода (присвоения) величины какой-либо переменной

Если после ввода выражения щелкнуть на кнопке ОК, присвоенная величина переменной появится в окне выражений.

Команда Variable Domain задает величину в выбранном интервале. Для этого используется команда Variable Domain, которая вызывает появление окна для ввода названия переменной, вида выражения и соответственно этому необходимо задать интервал выражения какой-либо функции (рис. 2.7).

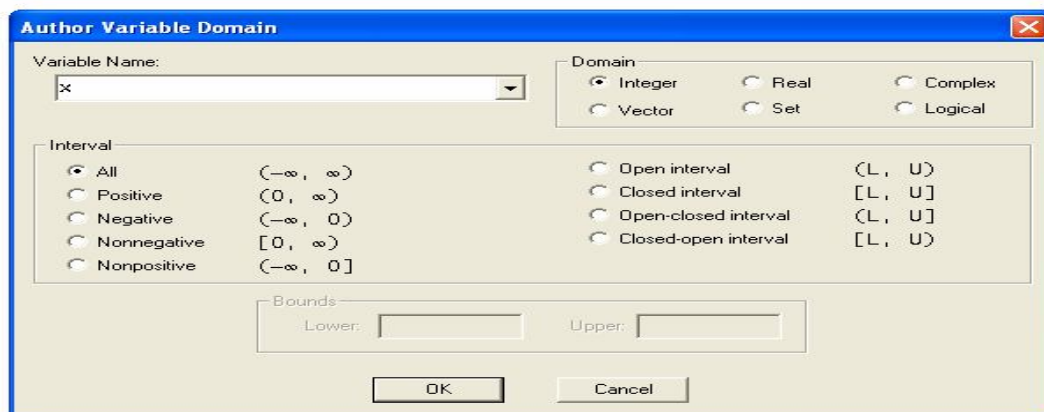


Рис.2.7. Окно для ввода переменной и задания интервала

Команда Function Definition —служит для определения функции переменных. Для функции переменной необходимо ввести название аргумента или названия функции в первой строке появившегося окна Function Definition. Затем нажать ОК и присвоенное значение функции переменной появится в окне выражений (рис. 2.8).

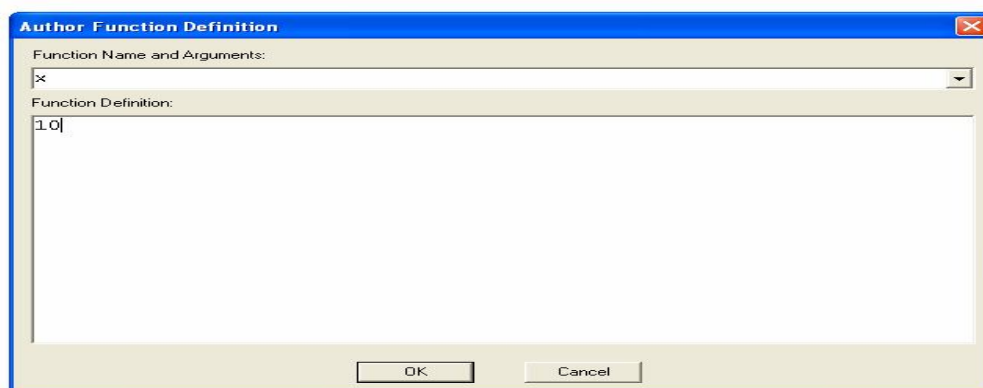


Рис.2.8. Окно для определения функции переменных

## 2.2. Работа с окнами и графикой

В меню Window расположены основные команды управления окнами и элементами пользовательского интерфейса:

- New Algebra View — создание дополнительного окна для выражений;
- New 2D-plot Window — создание дополнительного окна для двухмерной графики;
- New 3D-plot Window — создание дополнительного окна для трехмерной графики;
- Cascade — каскадное расположение окон;
- Tile Horisontally — расположение окон по горизонтали;
- Tile Vertically — расположение окон по вертикали;
- Arrange Icons — упорядочить расположение кнопок (значков) свернутых окон;
- Toolbar — отображение или скрытие панели инструментов;

- Status Bar — отображение или скрытие строки состояния.

Рассмотрим возможности задания и вывода окон.

**Создание нового окна для двухмерной графики.** Команда New 2D-plot Window создает новое окно для построения двухмерных графиков. Такое окно имеет собственную панель инструментов и кнопки свертывания, восстановления и закрытия окна. Имеет оно и свое главное меню, несколько отличное по набору позиций и команд от меню окна выражений.

Основными объектами окна двухмерной графики являются: строка заголовка, строка меню, панель инструментов, сам график, графический курсор (в виде креста +), координатная система, область выделения части графика (показана пунктирными линиями) и строка состояния, в которой отображаются координаты курсора (Cross), координаты точки пересечения осей (Center), которая находится в центре окна с графиком, и масштабные множители по горизонтальной и вертикальной осям (Scale).

Графический курсор можно перемещать клавишами управления курсором. Для увеличения шага перемещений надо нажимать клавиши управления курсором одновременно с клавишей Ctrl. Курсор служит для приближенного определения координат характерных точек графика и может иметь различные режимы перемещения. Они будут описаны чуть позже.

**Панель инструментов окна двухмерной графики** дает эффективные и быстрые средства построения и модификации графиков. При этом практически не требуется обращение к командам меню, хотя их набор несколько больше, чем набор команд, доступных с панели инструментов. Панель инструментов содержит шесть групп кнопок быстрого управления построением графиков:

- Копирование и печать графика:
  - Copy plot window — копирование графика в буфер обмена;
  - Print — печать графика;
- Управление построением графика:
  - Create annotation — создание комментария, располагаемого по месту графического курсора;
  - Plot expression — построение графика выделенного выражения;
  - Delete last plot — стирание последнего построенного графика;
- Центрирование и масштабирование графика:
  - Center on cross — центрирование графика относительно курсора;
  - Center on origin — центрирование графика по самому себе;
  - Set scale — установка масштаба;
  - Set range with box — развертывание выделенного графика на все окно.
- Расширение области графика:
  - Zoom both out (F10) — расширение области графика в обе стороны;
  - Zoom vertical out (F8) — расширение области графика по вертикали;
  - Zoom horizontal out (F6) — расширение области графика по горизонтали.
- Сжатие области графика:
  - Zoom both in (F9) — сжатие области графика в обе стороны;

- Zoom vertical in (F7) — сжатие области графика по вертикали;
  - Zoom horizontal in (F5) — сжатие области графика по горизонтали.
- Возвращение в окно выражений — Algebra window (Ctrl+1).

**Строка меню окна двумерной графики.** Работа с кнопками на панели инструментов довольно очевидна. Однако окно двумерных графиков имеет еще и собственную строку меню, в которой имеются следующие позиции:

- File — открытие окна, предварительный просмотр и печать графиков, выход из графического окна (для файлов графиков используется формат AcroSpin);
- Edit — создание и удаление комментария, стирание последнего графика и копирование графика целиком или выделенной части в буфер обмена;
- Set — центрирование, установка масштаба и размера графика;
- Plot — инициирование построения графика для выделенного выражения;
- Options — установка параметров (см. ниже);
- Windows — управление окнами;
- Help — команды справочной системы Derive.

Команды меню, за исключением команд меню Options, в основном аналогичны уже описанным командам меню главного окна. Поэтому ограничимся описанием команд меню Options:

- Axes — установка параметров координатных осей и надписей у них;
- Cross — установка параметров графического курсора (его наличия и цвета);
- Grids — установка параметров координатной сетки графика;
- Coordinate System — установка системы координат (Rectangular/Polar);
- Point — установка параметров точек;
- Plot color — установка цвета линий графиков;
- Background color — установка цвета фона графиков;
- Printing — установка параметров печати;
- Follow Mode — установка режима свободного перемещения графического курсора;
- Trace Mode — установка режима перемещения курсора по графику;
- Autoscal Mode — установка автоперемещения курсора.

Назначение большинства команд очевидно. Некоторые, например команды установки цвета, аналогичны уже описанным. Поэтому остановимся на тех командах, которые достаточно специфичны. После выбора команды Axes открывается диалоговое окно, показанное на рис.2.9.

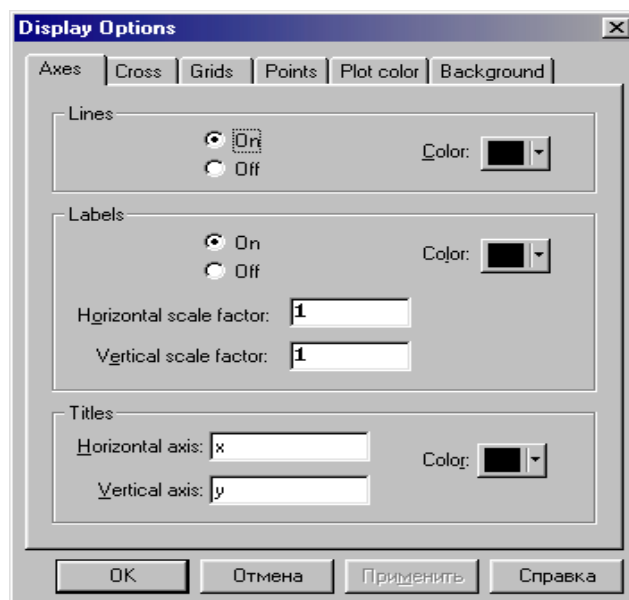


Рис.2.9. Окно установки параметров координатных осей

В этом окне в группе переключателей Lines можно задать отображение (On) или скрытие (Off) координатных осей, а в группе аналогичных переключателей Labels — и меток на них. В группе Titles можно задать также надписи у горизонтальной (поле Horizontal axis) и вертикальной (поле Vertical axis) осей. По умолчанию эти надписи представлены обозначениями осей — буквами x и y. Для всех элементов можно вывести окно установки параметров цвета, используя кнопки Color.

Еще одна команда, которую следует отметить, — Grids, которая выводит окно, позволяющее изменить параметры координатной сетки (рис.2.10). По умолчанию эта сетка представлена точками — местами пересечения линий сетки.

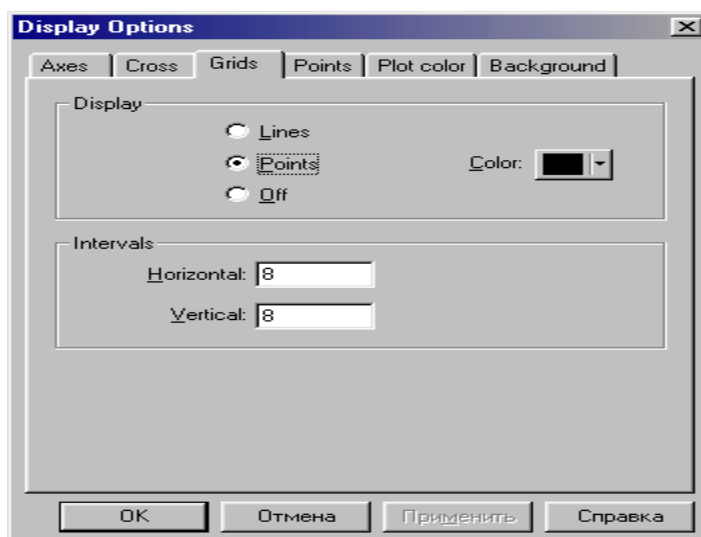


Рис. 2.10. Окно установки параметров координатной сетки

В этом окне с помощью переключателей группы Display можно задать параметры отображения сетки: в виде точек (Point), линий (Lines) или отсут-



стве сетки (Off). Кнопка Color позволяет вывести окно установки параметров цвета точек или линий сетки.

**Панель инструментов окна трехмерных графиков.** С помощью команд панели инструментов или строки меню вид графика можно существенно изменить. Так, на рис. 2.11 показан тот же график, но с другим масштабом и под другим углом зрения. Для изменения угла зрения использована кнопка с изображением глаза.

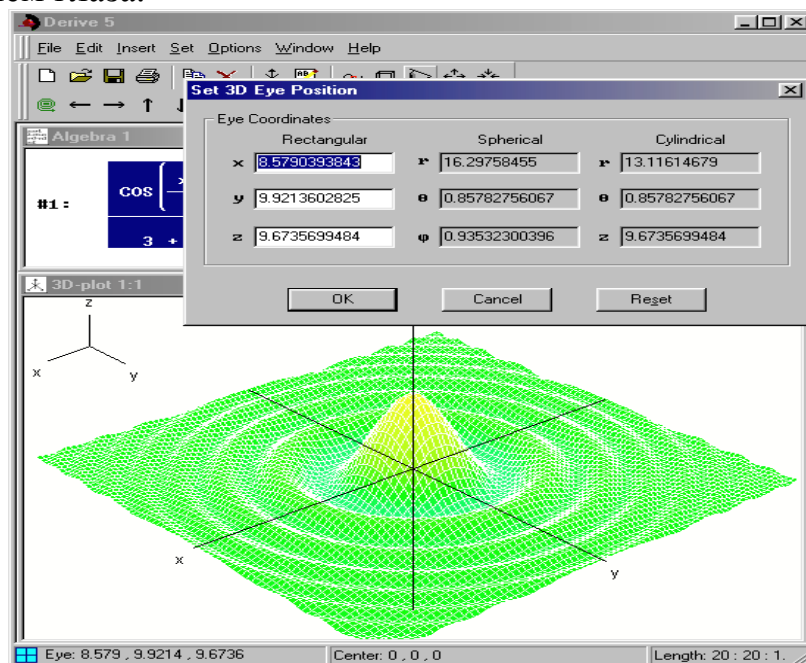


Рис. 2.11. График трехмерной поверхности с измененными масштабом и углом зрения

На рис. 2.11 видны строка меню и панель инструментов окна построения трехмерных графиков. Панели инструментов для трехмерной такие же, что и для двухмерной графики, отличие только в команде группы «Центрирование и просмотр графики». Где команда Set box center позволяет установить координаты  $x$ ,  $y$  и  $z$  центра графика. В режиме Auto координата  $z$  вычисляется автоматически. Команда Set viewpoint выводит окно, показанное в верхнем правом углу рис. 2.11. В нем нужно установить координаты точки, с которой ведется обзор трехмерной поверхности. Для изменения числа линий сетки на трехмерной поверхности служит команда Set grid panel.

**Строка меню окна трехмерной графики.** Команды меню окна трехмерной графики практически аналогичны командам меню окна двухмерной графики. Поэтому ограничимся описанием только отличающихся команд.

Команды меню Set:

- Center — установка координат центра;
- Eye — установка координат точки обзора;
- Grids — установка числа линий сетки по осям  $x$  и  $y$ ;
- Length — установка интервалов между линиями.

Команды меню Options:

- Axes — установка параметров координатных осей и надписей у них;

- Plot color — установка цвета линий графиков;
- Background color — установка цвета фона графиков;
- Printing — установка параметров печати;
- Remove Hidden Line — удаление невидимых линий.

Команда Axes выводит окно, показанное на рис. 2.12. В этом окне можно включить или выключить вывод координатных осей трехмерных графиков и задать надписи у них. По умолчанию заданы просто наименования осей x, y и z.

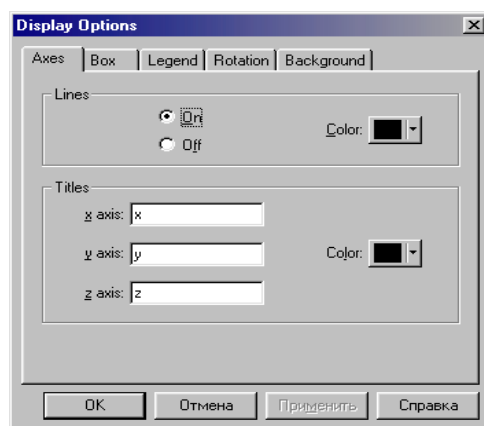


Рис.2.12. Окно параметров отображения осей Axes трехмерных графиков.

Команда Printing выводит окно установки параметров печати трехмерных графиков: Header and Footer — печать колонтитулов (верхних и нижних), Center plot — печать центральной точки графика и Black and white only — печать только в черно-белом цвете. Щелчок на кнопке Fonts выводит панель доступных шрифтов.

### 2.3. Аналитические методы решения дифференциальных уравнений в среде Derive

Система Derive позволяет решать аналитическими методами системы дифференциальных уравнений (ДУ) первого и второго порядков. Для этого используются два библиотечных файла: ode1.mth и ode2.mth. Первый из них содержит функции решения ДУ первого порядка, второй — второго порядка. Особенности решения ДУ аналитическими методами в Derive 5 являются следующие:

- решение ДУ, в зависимости от его вида, получается в виде функции  $\varphi(x,y)$  или уравнения  $\varphi(x, y) = 0$ ;
- ответ может содержать интегралы, которые система не может «взять»;
- решение может иметь специальные функции;
- ответ может быть представлен в вещественной и комплексной форме;
- если решение системой не получено, то откликом будет сообщение — «inapplicable» (означает — «неприменимо»).

Для получения решения в явном виде  $y = f(x)$  необходимо представить решение в виде уравнения и разрешить его относительно  $y$ . После тако-

го преобразования решение можно получить в графическом или табличном виде путем табулирования функции  $y = f(x)$ . При необходимости, специальные функции, содержащиеся в решении, можно разложить в степенной ряд, получив приближенное решение уравнения в аналитическом виде.

Система Derive содержит несколько десятков функций решения ДУ. Рассмотрим только те, которые наиболее часто используются на практике.

### 2.3.1. Функции решения уравнений первого порядка

**Функция DSOLVE1.** Эта функция позволяет решать уравнения первого порядка вида:

$$p(x, y) + q(x, y) * y' = 0.$$

Представляется функция в одной из следующих форм:

$$DSOLVE1(p, q, x, y, x_0, y_0);$$

$$DSOLVE1\_GEN(p, q, x, y, c).$$

Первая форма применяется для получения частного решения. Вторая — для получения общего решения. Символьные переменные в этих функциях имеют следующий смысл:

- $p, q$  — функции, содержащиеся в ДУ рассматриваемого вида ;
- $x_0, y_0$  — начальные условия решения ДУ;
- $c$  — произвольная постоянная.

В дальнейшем решения ДУ аналитическими методами, с помощью функций Derive 6, будем рассматривать на примерах.

**Пример 1.** Пусть необходимо найти частное и общее решение уравнения:

$$x/y * d*y / d*x + x*y + 1 = 0$$

при начальных условиях  $y(1) = 1$ . В нашем примере  $p = x*y + 1, q = x/y, x_0 = 1, y_0 = 1$ .

Функции DSOLVE запишутся в следующем виде:

$$DSOLVE1(x*y+1, x/y, x, y, 1, 1);$$

$$DSOLVE1\_GEN(x*y+1, x/y, x, y, c).$$

Технология решения задачи очень проста и состоит в выполнении следующих действий:

- набор и ввод одной из представленных функций;
- выполнение команды **Simplify** или **Approximate** (в первом случае ответ будет точным, во втором — численным с представлением чисел в естественной форме).

В результате решения будут получены следующие ответы:

- частное решение —  $(x*y-1) / x*y = -\ln(x)$ ;

- общее решение —  $1 / x*y = \ln(x) + c$ .

Ответ получен в виде уравнений. Для получения решения в явном виде их нужно разрешить относительно  $y$ . Ответ здесь очевиден:  $y = 1/x * (\ln(x) + 1), y = 1/x * (\ln(x) + c)$ .

Решение дифференциального уравнения показано на рис.2.13 и 2.14. На рис.3.15 приведены результаты решения уравнения при следующих значениях

$$p = x \cdot y + 1, q = y/x, x_0 = 1, y_0 = 1;$$

$$p = x \cdot \ln(y) + 1 = x/y, x_0 = 1, y_0 = 1;$$

$$p = xy + 1, q = 2, x_0 = 1, y_0 = 1;$$

Из полученного решения видно, что в первом случае решения нет (ответ «inapplicable»), во втором случае в решении содержится не вычисленный интеграл и, наконец, в третьем случае решение получено в комплексной форме, содержащей специальную функцию.

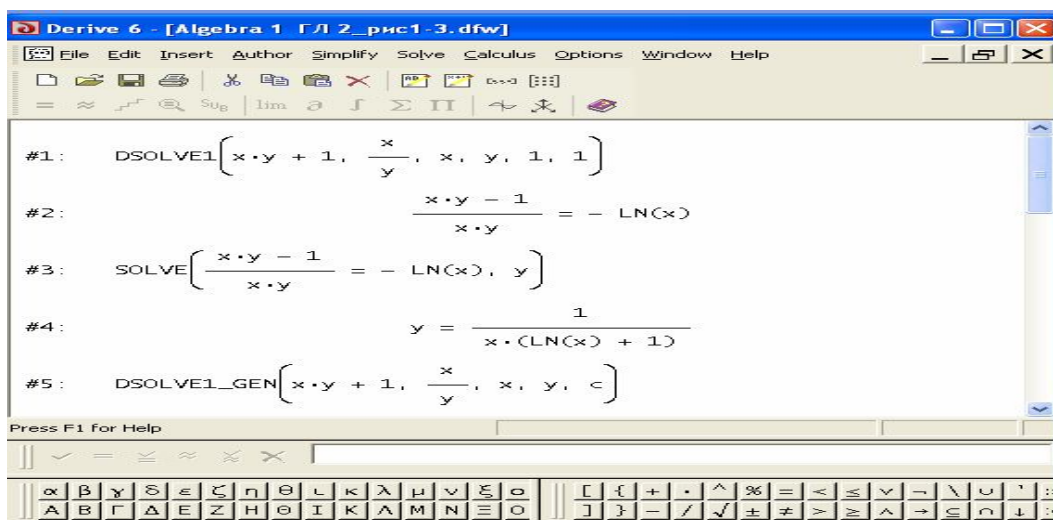


Рис.2.13. Решение дифференциальных уравнений в аналитическом виде (получение первого ответа)

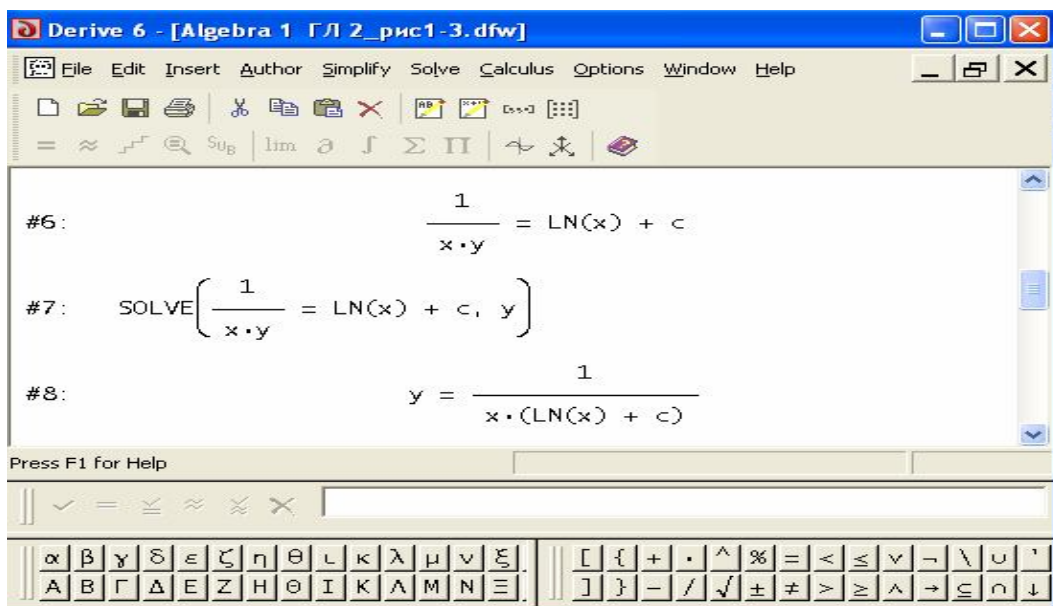


Рис.2.14. Решение дифференциальных уравнений в аналитическом виде (получение второго ответа)

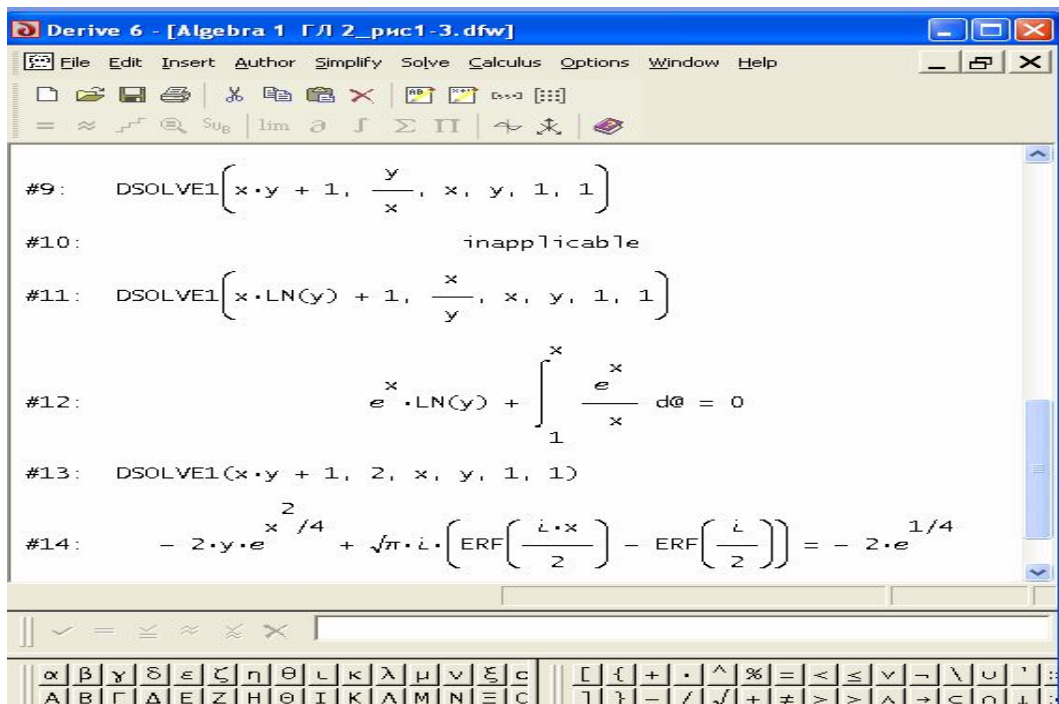


Рис.2.15. Решение дифференциальных уравнений при различных значениях  $p(x,y)$  и  $q(x,y)$

**Функция *LINEAR1*.** Функция решает ДУ вида:  $y' + p(x) * y = q(x)$  и имеет следующие две формы представления:

$LINEAR1(p, q, x, y, x0, y0);$

$LINEAR1\_GEN(p, q, x, y, c);$

Первая функция дает частное, а вторая общее решение. Символьные переменные в этих функциях имеют тот же смысл, что и в функции *DSOLVE*.

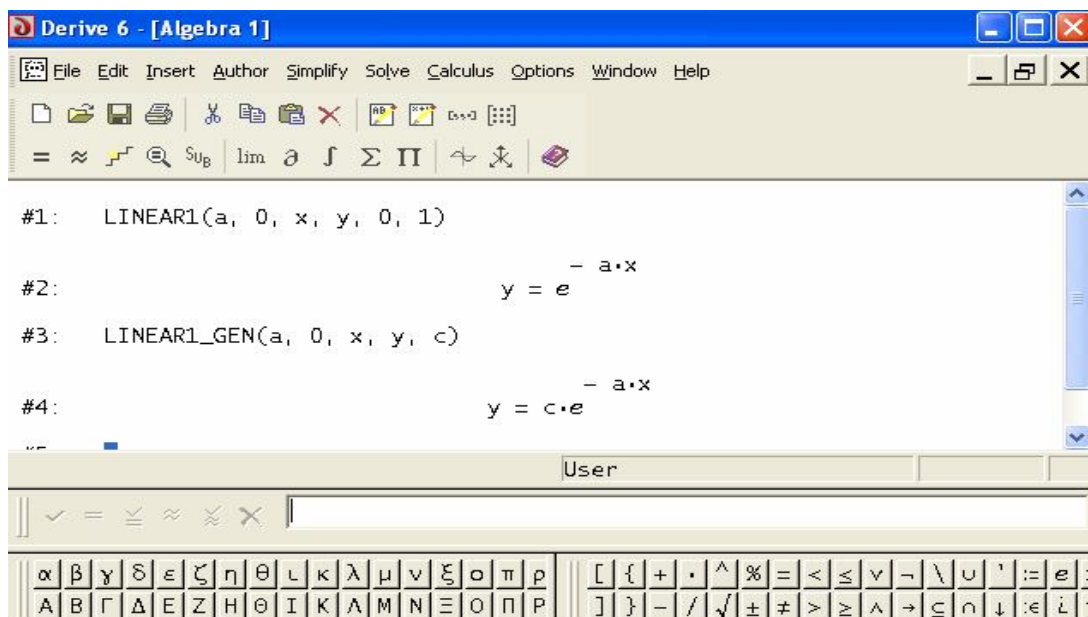


Рис.2.16. Решение уравнения  $dy(x)/dx + ay = 0$  с использованием функции *LINEAR1*

**Пример 2.** Пусть необходимо решить уравнение

$$d*y(x) / d*x + a*y = 0$$

при начальных условиях  $y(0) = 1$ .

В данном случае

$$p = a, q = 0, x_0 = 0, y_0 = 1$$

Тогда функции будут иметь вид:

LINEAR1 (a, 0, x, y, 0, 1);

LINEAR1\_GEN (a, 0, x, y, c);

Решение уравнения показано на рис.2.16. В данном случае решение получено в явном виде.

**Пример 3.** Необходимо решить ДУ  $y' + 2xy = x$  (найти частное и общее решения), если начальными условиями являются:  $y(0) = 1$ .

В данном случае  $p = 2x, q = x, x_0 = 0, y_0 = 1$ . Тогда функции

LINEAR1 будут иметь вид:

LINEAR1 (2x, x, x, y, 0, 1);

LINEAR1\_GEN (2x, x, x, y, c);

Решение уравнения показано на рис.2.17.

**Функция SEPARABLE.** Функция решает ДУ вида:  $y' = p(x)*q(y)$  и имеет следующие две формы представления:

SEPARABLE (p, q, x, y, x0, y0);

SEPARABLE\_GEN (p, q, x, y, c).

### Примечание

Обозначения переменных в этой функции те же, что и в функциях, рассмотренных выше.

Функция позволяет определить частное и общее решение дифференциального уравнения при начальных условиях  $y(x_0) = y_0$ .

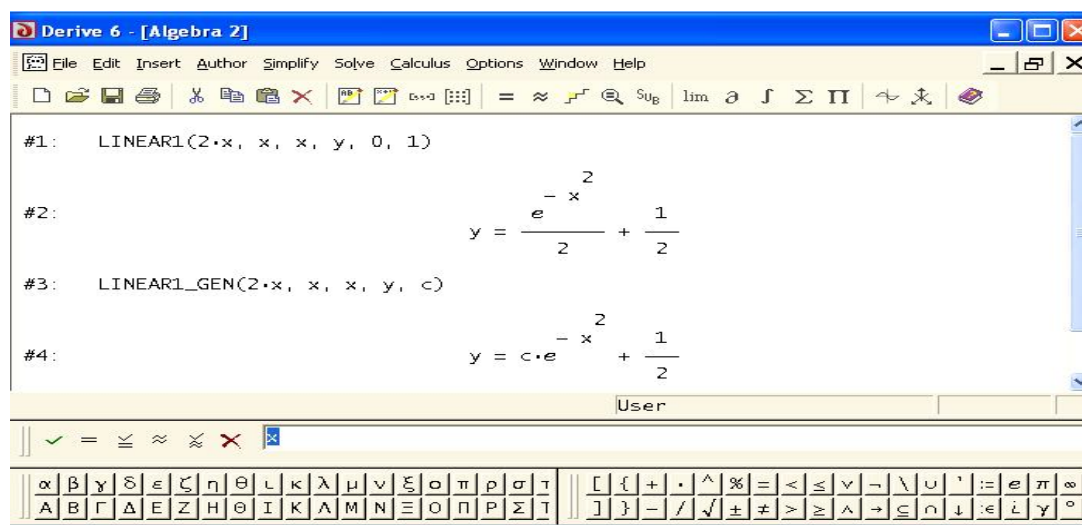


Рис.2.17. Решение уравнения  $y' + 2xy = x$  с использованием функции LINEAR1

**Пример 4.** Дифференциальное уравнение имеет вид:  $y' = x*y$ . Требуется

найти частное и общее решения при начальных условиях  $y(0) = 1$ . В данном случае переменные функций SEPARABLE будут иметь значения:  $p = x$ ,  $q = y$ ,  $x_0 = 0$ ,  $y_0 = 1$ , а функции запишутся в следующем виде:

SEPARABLE (p, q, x, y, x0, y0)

SEPARABLE\_GEN(p, q, x, y, c)

Решим теперь более сложную задачу.

Пример 5. Пусть дифференциальное уравнение имеет вид:  $y' - (x^2 - 1)(y^2 + 1)$ , начальными условиями являются:  $y(0) = 1$ . В данном случае переменные функций будут иметь значения:

$p = x^2 - 1$ ,  $q = y^2 + 1$ ,  $x_0 = 0$ ,  $y_0 = 1$ , а сами функции запишутся в следующем виде:

SEPARABLE ( $x^2 - 1$ ,  $y^2 + 1$ , x, y, 0, 1);

SEPARABLE\_GEN ( $x^2 - 1$ ,  $y^2 + 1$ , x, y, c).

Откликом реализации этих функций будут решения уравнения в неявном виде.

**Функция BERNOULLI.** Функция предназначена для решения дифференциального уравнения вида:  $y' + p(x)y = q(x)y^k$  и имеет следующие две формы представления:

BERNOULLI\_ODE (p, q, k, x, y, x0, y0);

BERNOULLI\_ODE\_GEN (p, q, k, x, y, c).

#### **Примечание**

Обозначения переменных в этой функции те же, что и в функциях, рассмотренных выше.

Первая форма функции дает частное решение при заданных начальных условиях  $x_0$  и  $y_0$ . Вторая — общее решение.

$K$  и  $c$  — это константы.

Пример 6. Пусть необходимо решить следующее дифференциальное уравнение:

$y' + (x + 1)y = x*y^k$  при начальных условиях  $y(0) = 1$  и значении  $k = 2$ . В данном случае

$p = x + 1$ ,  $q = x$ . Тогда функции будут иметь вид:

BERNOULLI\_ODE (x+1, x, k, x, y, 0, 1);

BERNOULLI\_ODE\_GEN (x+1, x, k, x, y, c).

После набора и ввода функций нажимается кнопка **Simplify** или **Approximate**, расположенные на панели инструментов. На экране будет сформировано аналитическое и численное решения уравнения при  $k = 2$ .

### **2.3.2. Функции решения уравнений второго порядка**

Решение дифференциальных уравнений второго порядка осуществляется функциями, которые содержатся в библиотечном файле ode2.mth. Технология решения уравнений практически не отличается от технологии решения уравнений первого порядка.



Ниже приводятся функции, которые позволяют наиболее просто решать уравнения второго порядка.

**Функция DSOLVE2.** Эта функция представляется в следующих трех формах:

DSOLVE2 (p, q, r, x, c1, c2);  
DSOLVE2\_BV (p, q, r, x, x0, y0, x2, y2);  
DSOLVE2\_IV (p, q, r, x, x0, y0, v0)

Все формы функции предназначены для решения уравнения вида:  $y'' + p(x)y' + q(x)y = r(x)$ . По виду уравнения, символьным переменным функции DSOLVE2 и примерам решения уравнений первого порядка понятно назначение символьных переменных в функциях DSOLVE2.

Рассмотрим примеры решения дифференциальных уравнений второго порядка указанного типа.

**Пример 7.** Пусть уравнение имеет вид:  $y'' + y' + y = x + 1$ . Необходимо решить дифференциальное уравнение с помощью приведенных выше функций при начальных условиях  $y(0) = 1$ ,  $y'(0) = 1$ . В данном примере символьные переменные функций имеют значения:  $p = 1$ ,  $q = 1$ ,  $r = x + 1$ ,  $x_0 = 0$ ,  $y_0 = 1$ ,  $x_1 = 0$ ,  $y_1 = 1$ .

Тогда функции решения дифференциального уравнения будут иметь вид:

DSOLVE2 (1, 1, x+1, x, c1, c2);  
DSOLVE2\_BV (1, 1, x+1, x, 0, 1, 0, 1);  
DSOLVE2\_IV (1, 1, x+1, x, 0, 1, 0, 1).

Обратим внимание на результаты решения по функциям DSOLVE2\_BV и DSOLVE2\_IV. Параметры этих функций одинаковы, но ответы разные. Какое же из этих решений верное? Найдем первую и вторую производные и убедимся, что результат решения будет равен правой части исходного уравнения (то есть  $x + 1$ ) и будут удовлетворены начальные условия  $y(0) = 1$ ,  $y'(0) = 1$ . Определив первую и вторую производные с помощью команды **Differentiate**, расположенной на панели инструментов, и вычислив сумму  $y'' + y' + y$ , убеждаемся, что эта сумма равна  $x + 1$ . Однако начальные условия не соответствуют заданным и имеют значения:

в случае функции DSOLVE2\_BV –  $y(0) = 1$ ,  $y'(0) = 5$  (вместо  $y'(0) = 1$ );  
в случае функции DSOLVE2\_IV –  $y(0) = 1$ ,  $y'(0) = 0$  (вместо  $y'(0) = 1$ );

Таким образом, задача Коши не решена. Почему? В чем наша ошибка? Или функции не предназначены для решения задачи Коши?

**Функция AUTONOMOUS\_CONSERVATIVE.** Функция находит решение дифференциального уравнения второго порядка  $y'' = q(y)$ . Она представляется в виде:

AUTONOMOUS\_CONSERVATIVE (q, x, y, x0, y0, v0).

Символьные переменные функции имеют следующие значения:

$q = f(x, y)$ ;

$x_0, y_0, y'_0$  – начальные условия  $y(x_0) = y_0, y'(x_0) = y'_0$ ;



$x, y$  – аргумент и искомая функции;

$$y' = v_0$$

**Пример 8.** Пусть требуется решить дифференциальное уравнение  $y = f(x, y)$ , если:

a)  $f(x, y) = y + 1$ ;

b)  $f(x, y) = x + 1$ ;

c)  $f(x, y) = x + y + 1$ .

Решение можно получить при нулевых начальных условиях, то есть  $y(0) = 0, y'(0) = 0$ .

В данном случае  $q_a = -y + 1, q_b = x + 1, q_c = x + y + 1, x_0 = 0, y_0 = 0, y' = 1$ . Тогда функции будут иметь вид:

a) `AUTONOMOUS_CONSERVATIVE(y+1, x, y, 0, 0, 0)`;

b) `AUTONOMOUS_CONSERVATIVE(x+1, x, y, 0, 0, 0)`;

c) `AUTONOMOUS_CONSERVATIVE(x+y+1, x, y, 0, 0, 0)`.

После команды **Simplify** отклик будет следующее решение:

a)  $x = \text{LN}(\sqrt{y \cdot (y + 2)}) + y + 1$ ;

b)  $x = \frac{\sqrt{2} \sqrt{y(x+1)}}{x+1}$

c)  $x = \text{LN}(\sqrt{y \cdot (y + 2 \cdot (x + 1))}) + y + x + 1 - \text{LN}(x + 1)$

**Функция LIOUVILLE.** Функция предназначается для получения общего решения дифференциального уравнения вида:

$$y'' + p(x) * y' + q(y) * (y')^2 = 0$$

Функция имеет вид:

$$\text{LIOUVILLE}(p, q, x, y, c1, c2).$$

Обозначения переменных здесь аналогичны переменным, используемым в предыдущих функциях.

**Пример 9.** Пусть требуется найти общее решение дифференциального уравнения:

$$y'' + y' + (y')^2 = 0.$$

В данном случае  $p = 1, q = -1$ . Тогда LIOUVILLE будет иметь вид:

$$\text{LIOUVILLE}(1, -1, x, y, c1, c2).$$

Откликом данной функции является следующее решение, в неявном виде:

$$e^{-y} + c2 * e^{-x} + c1 = 0.$$

Проверим его правильность. Для этого представим ответ в явном виде, разрешив полученное уравнение относительно  $y$ . Найдя первую и вторую производные, подставим их в исходное дифференциальное уравнение и убедимся, что имеет место тождество.

Функция LIOUVILLE позволяет получить частное решение, если заданы начальные условия. Убедимся в этом на нашем примере. Пусть начальными условиями будут  $y(0) = 1, y'(0) = 1$ . Тогда функция будет иметь вид:

$$\text{LIOUVILLE}(1, 1, x, y, 0, 1, 1).$$

Откликом при реализации этой функции будет следующий ответ:

$$e^y - e^{-x} = 0 \text{ или в явном виде: } y = -x. \text{ Подставив ответ в исходное диффе-}$$

ренциальное уравнение, получим тождество  $0 = 0$ .

Аналитические методы решения дифференциальных уравнений дают решения в виде точных или приближенных формул. Это позволяет:

- представлять решения в виде графика;
- выполнять любые математические преобразования над полученным решением (дифференцировать, интегрировать, находить пределы, раскладывать в ряд и т. П.);
- изучать физическое явление, представляя полученную формулу как его математическую модель.

Покажем возможности аналитических методов на примерах исследования динамики систем массового обслуживания (СМО).

**Пример 10.** Функционирование одноканальной СМО с отказами описывается следующим дифференциальным уравнением:  $p_0' + (\lambda + \mu)p_0 = \mu$ . Символы в уравнении имеют следующий смысл:

- $p_0$  — вероятность того, что СМО свободна от обслуживания (заявки отсутствуют);
- $\lambda$  — интенсивность потока заявок;
- $\mu$  — интенсивность обслуживания заявки.

По структуре уравнения видно, что для его решения можно воспользоваться функцией `LINEAR1(p, q, x, y, x0, y0)`. В нашей задаче определяется функция  $p(t)$ , то есть переменными  $x, y$  будут соответственно —  $t, p$ . Предположим, что при  $t = 0$  СМО свободна от заявок.

Тогда  $x_0 = t_0 = 0, y_0 = p(t_0) = 1$ . Подставляя переменные в выражение функции `LINEAR1`, получим:

`LINEAR1(λ+μ, μ, t, p, 0, 1)` .

После ввода этой функции и вызова команды **Simplify** получим частное решение в аналитическом виде. Упростив его с помощью пункта главного меню **Simplify** и функции **Expand**, получим ответ.

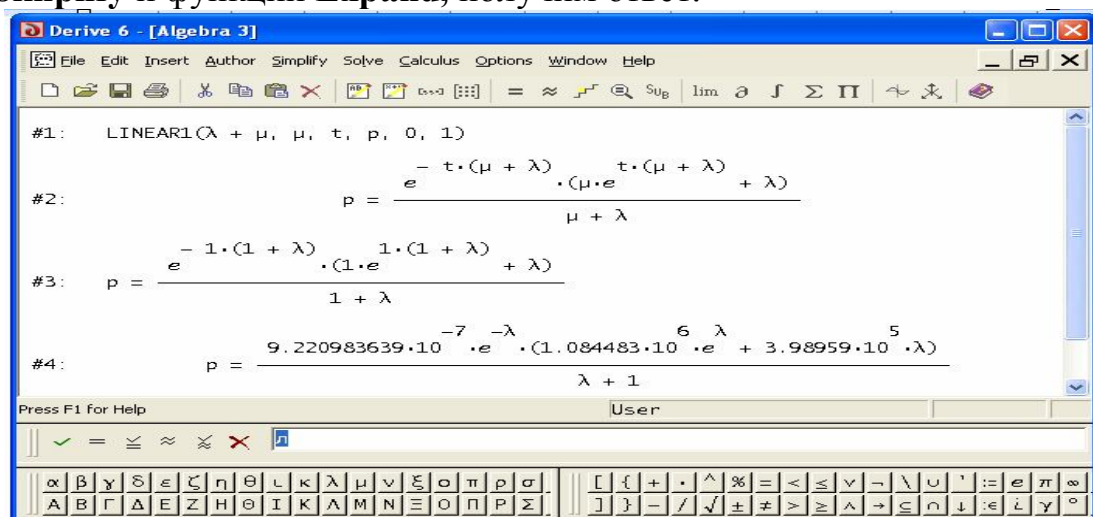


Рис.2.18. Частное решение уравнения `LINEAR1(λ+μ, μ, t, p, 0, 1)` в аналитическом виде (при  $\mu = 1$ )

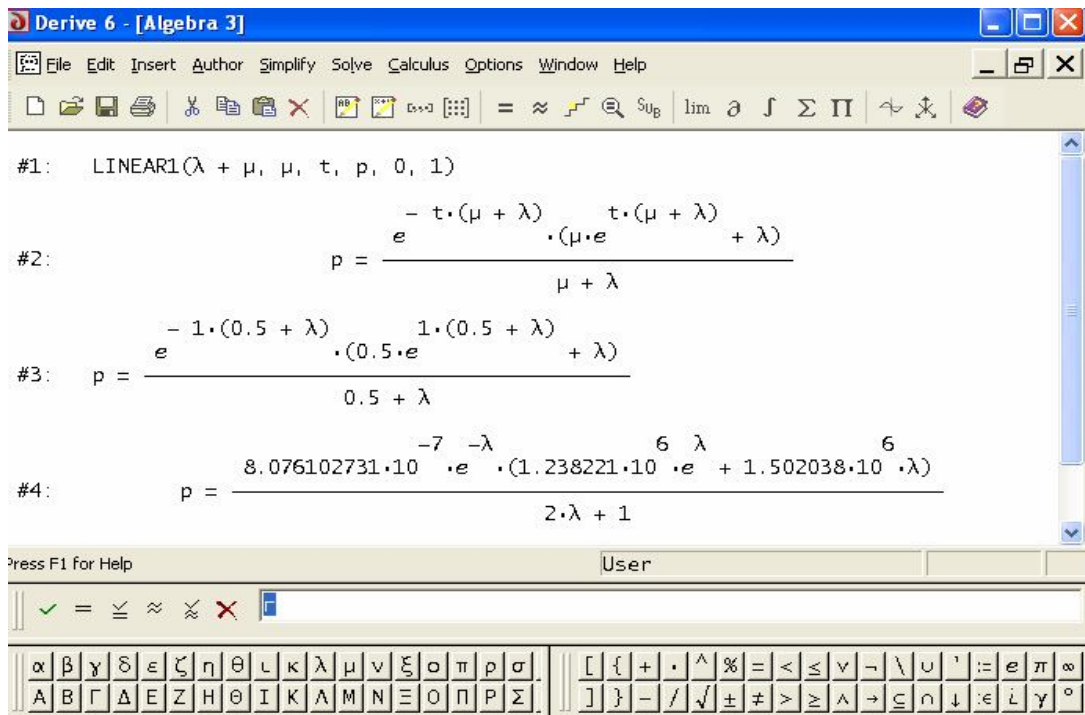


Рис.2.19. Частное решение уравнения LINEAR1 ( $\lambda + \mu$ ,  $\mu$ ,  $t$ ,  $p$ ,  $0$ ,  $1$ ) в аналитическом виде (при  $\mu = 0.5$ )

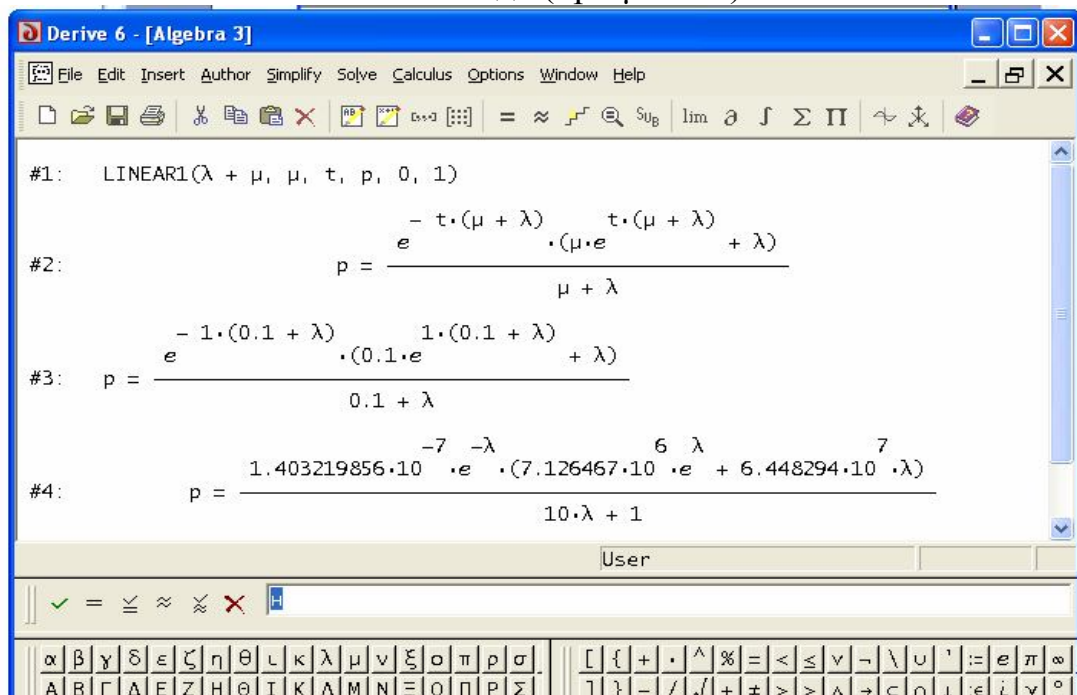


Рис.2.20. Частное решение уравнения LINEAR1 ( $\lambda + \mu$ ,  $\mu$ ,  $t$ ,  $p$ ,  $0$ ,  $1$ ) в аналитическом виде (при  $\mu = 0.1$ )

На рис.2.21 приведены графики функции  $p = f(t)$  при интенсивности потока заявок  $\lambda=1$  [1/час] и различных значениях  $\mu$ . Из формулы и графиков можно сделать ряд важных выводов. Например:

- длительность переходных процессов в системе мала;
- установившееся значение вероятности свободного состояния системы равно  $\mu / (\lambda + \mu)$

➤ если  $\lambda = \mu$ , то установившееся значение вероятности того, что система свободна от обслуживания равно 50%, то есть обслуживающий орган будет загружен лишь на половину рабочего времени и т.д.

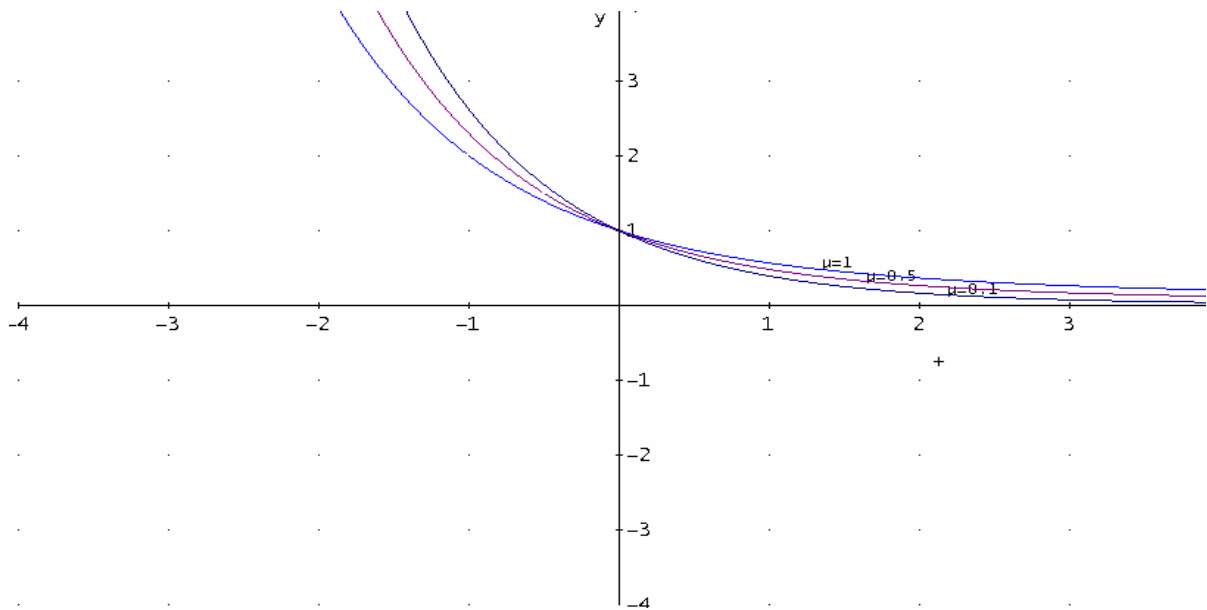


Рис.2.21. Графики функции  $p = f(t)$

**Пример 11.** Функционирование двухканальной системы массового обслуживания с отказами описывается следующим дифференциальным уравнением второго порядка:

$$p'' + (2\lambda + 3\mu)p' + (2\mu^2 + 2\lambda\mu + \lambda^2)p = 2\mu^2$$

Как и раньше, символ  $p$  означает вероятность того, что система в произвольный момент времени  $t$  свободна от обслуживания, символы  $\lambda$  и  $\mu$  обозначают соответственно интенсивность потока заявок и интенсивность их обслуживания. Необходимо определить вероятность того, что в произвольный момент времени  $t$  СМО свободна от обслуживания заявок (заявки отсутствуют).

Будем считать, что вначале (при  $t = 0$ ) заявок на обслуживание нет. Тогда начальными условиями решения задачи будут:

$$p(0) = 1, p'(0) = 1.$$

Из структуры уравнения видно, что его решение можно получить с помощью функции:

$$\text{DSOLVE2\_BV}(p, q, r, x, x_0, y_0, x_1, y_1).$$

В нашем случае  $p = (2\lambda + 3\mu)$ ,  $q = (2\mu^2 + 2\lambda\mu + \lambda^2)$ ,  $r = 2\mu^2$ ,  $x = t$ ,  $x_0 = 0$ ,  $y_0 = 1$ ,  $x_1 = 0$ ,  $y_1 = 1$ . Тогда функция DSOLVE2\_BV будет иметь вид:

$$\text{DSOLVE2\_BV}(2\lambda+3\mu, 2\mu^2 + 2\lambda\mu + \lambda^2, 2\mu^2, t, 0, 1, 0, 1).$$

Решение дифференциального уравнения в системе Derive 6 получить не сложно. После ввода функции DSOLVE2\_BV надо щелкнуть мышью по кнопке **Simplify** или **Approximate**, расположенных на панели инструментов. В результате на экране сформируется результат. В первом случае в виде точного решения, во втором — в виде численного в естественной форме представления чисел. Полезно упростить ответ с помощью пункта **Simplify** и

функции **Expand**. В конечном итоге получится следующее решение:

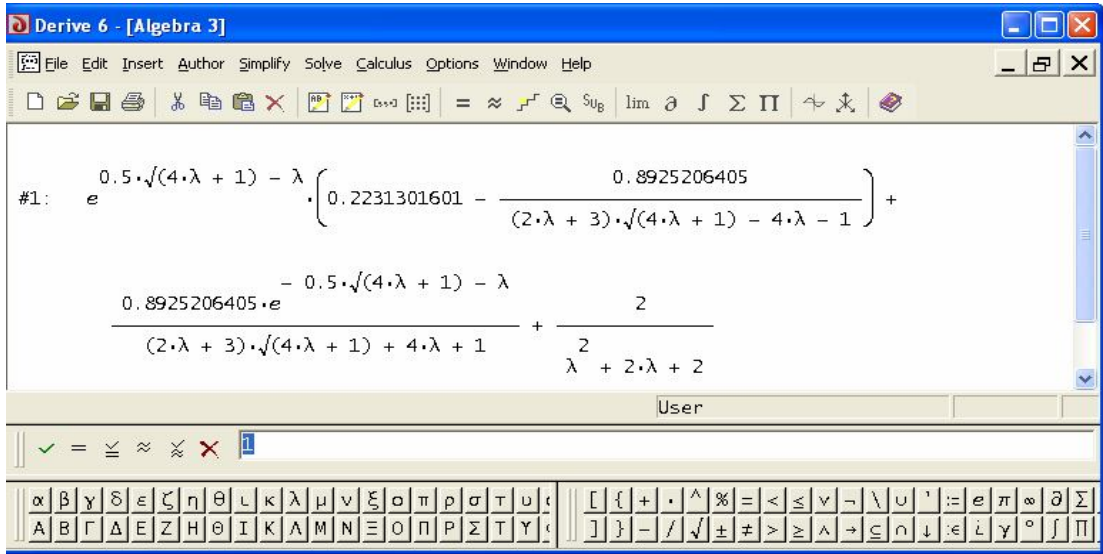


Рис.2.22.

При решении дифференциальных уравнений необходимо выбирать только ту решающую функцию, которая соответствует виду решаемого уравнения. Если выбрать иную функцию, то получить решение можно, но оно может быть ошибочным или полученным с большой погрешностью. Примером является решение нашего уравнения с помощью функции DSOLVE2\_IV. Эта функция отличается от функции DSOLVE2\_BV только видом начальных условий (вместо переменных  $x_i, y_i$  пишется переменная  $v_0$ ). Функция:

DSOLVE2\_IV ( $2\lambda+3\mu, 2\mu^2 + 2\lambda\mu + \lambda^2, 2\mu^2, t, 0, 1, 0, 1$ ) выдает решение, отличное от полученного с помощью функции DSOLVE2\_BV :

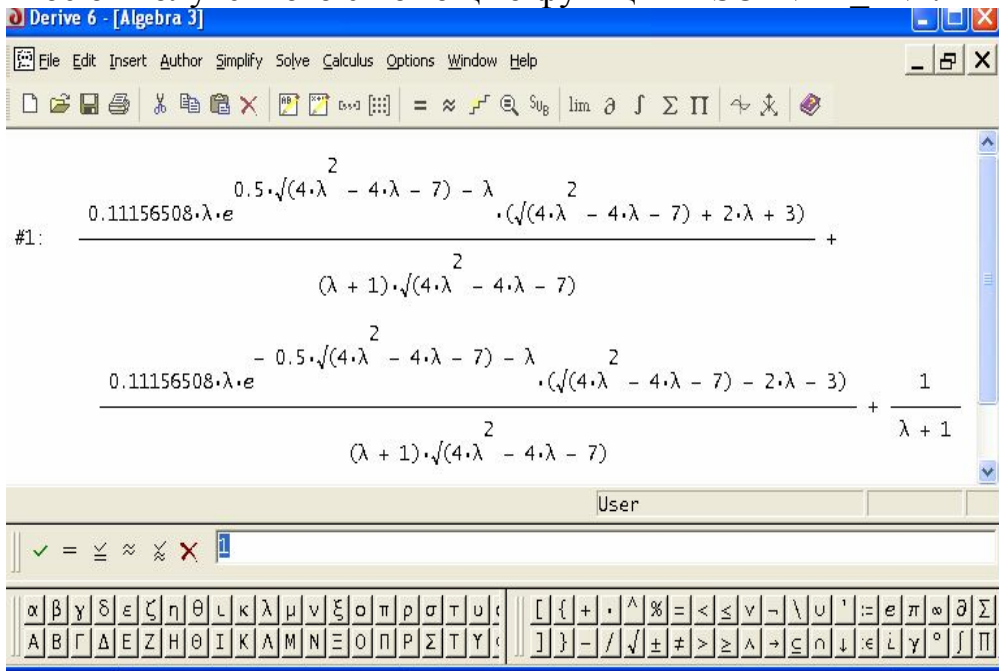


Рис.2.23

Разница в результатах решения показана на рис.2.24.

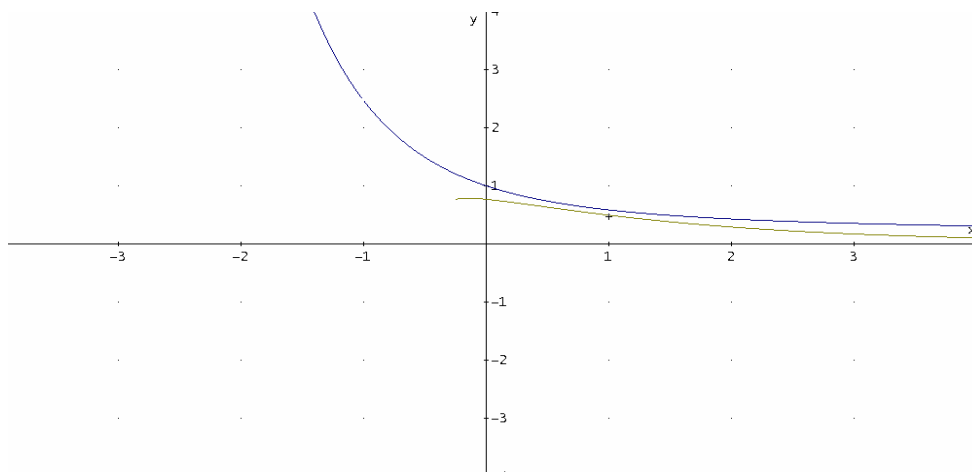


Рис.2.24. Зависимость вероятности свободного состояния от постоянной

## 2.4. Численные методы решения дифференциальных уравнений в среде Derive

Решение дифференциальных уравнений в Derive 6 осуществляется с помощью функций, находящихся в библиотечном файле `ode_appr.mth`. Файл содержит большое число различных функций решения таких уравнений. Рассмотрим основные из них.

**Функция EULER.** Функция предназначена для решения методом Эйлера дифференциального уравнения первого порядка  $y' = r(x, y)$  при начальных условиях  $y(x_0) = y_0$ . Функция представляется в следующем виде:

`EULER_ODE(r, x, y, x0, y0, h, n)`,

Где  $r(x, y)$  – правая часть дифференциального уравнения, разрешенного относительно производной;

$x, y$  – искомые неизвестные;

$x_0, y_0$  – начальные условия;

$h$  – шаг интегрирования;

$n$  – число итераций.

Откликом является решение в виде таблицы с числом строк, равном  $n$ , и шагом  $h$ .

Технологию решения уравнений рассмотрим на примерах.

**Пример 12.** Пусть необходимо решить дифференциальное уравнение:

$$\frac{dy}{dx} = \frac{x}{y} + 3e^{-2x}$$

при начальных условиях  $y(0) = 1$ . Выберем шаг интегрирования  $h = 0,2$  и число итераций  $n = 20$ .

В данном случае функция EULER будет иметь вид:



EULER\_ODE (x/y + 3e<sup>-2x</sup>, x, y, 0, 1, 0.2, 2) .

Технология решения задачи будет следующей:

➤ обращение к библиотечному файлу ODE\_APPR.MTH — **File | Load / ODE\_APPR;**

➤ набор и ввод правой части исходного уравнения (на экране в строке #2 формируется выражение:  $x/y + 3 e^{-2x}$ ).

➤ набор и ввод функции EULER (#2,x,y,0,1,0,2,20) (на экране формируется функция;

➤ щелчок мыши по кнопке **Approximate**, расположенной на панели инструментов (на экране формируется решение).

Процедуры решения уравнения показаны на рис.2.25.

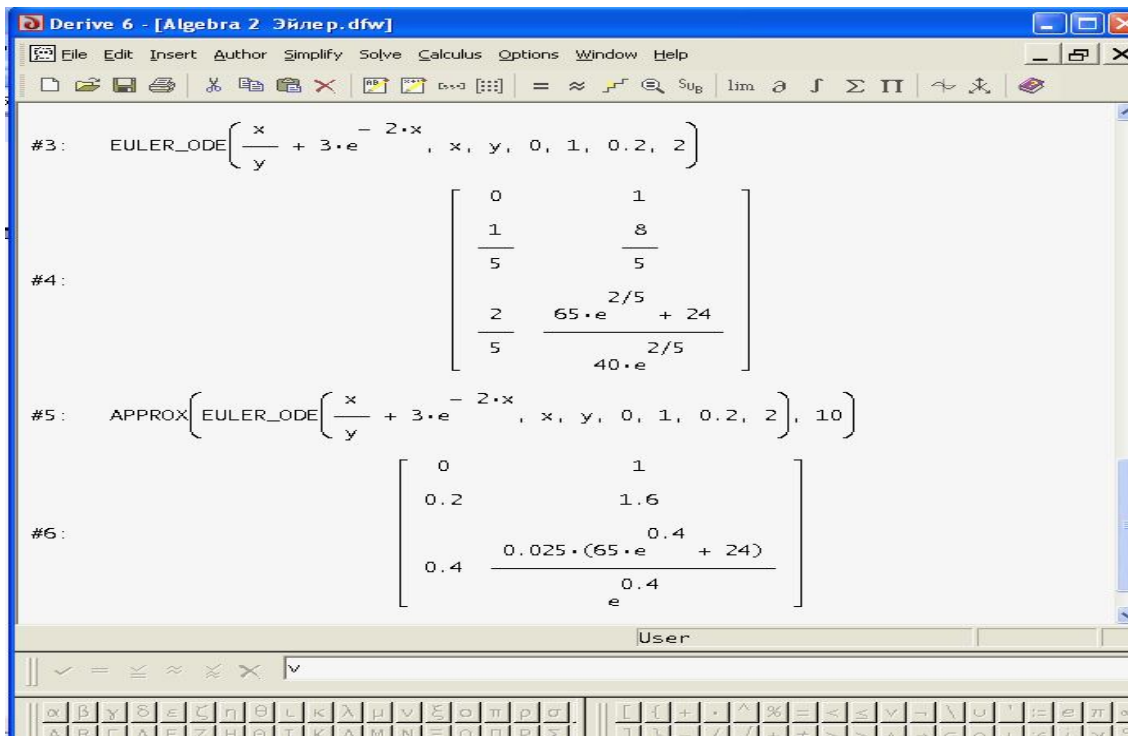


Рис.2.25. Решение дифференциального уравнения методом Эйлера

**Функция TAY\_ODE1.** Функция выдает решение в виде многочлена степени  $n$ . Функция имеет следующий вид:

TAYLOR\_ODE1(r, x, y, x0, y0, n);

$y' = f(x, y)$

Обозначения в этой функции те же, что и в функции EULER, символ  $n$  обозначает степень полинома, получаемого в результате решения уравнения.

**Пример 13.** Пусть необходимо решить дифференциальное уравнение из предыдущего примера. В этом случае функция будет иметь вид:

TAY\_ODE1(x/y + 3e<sup>-2x</sup>, x, y, 0, 1, n) .

Решение получим в виде многочлена степени  $n = 1, 2, 3$  путем редактирования функции в окне пользователя. Процедуры решения не нуждаются в комментариях. Они имеют вид, показанный на рис.2.26.

Ответами при  $n = 1, n = 2, n = 3$  соответственно являются:

$$1) x \cdot (3 + x) + 1;$$

$$2) -3 \cdot x^2 \cdot \text{LN}(e) - \frac{x \cdot x^2 \cdot (3 + x)}{2} + x \cdot (3 + x) + 1;$$

$$3) \left( \frac{2 \cdot 3 \cdot x^3 \cdot \text{LN}(e)^2}{3} \right) / \left( \frac{3 \cdot x \cdot x^3}{3} - 3 \cdot x^2 \right) \cdot \text{LN}(e) + x \cdot x^2 \cdot \left( \frac{2 \cdot 3^2 + 5 \cdot 3 \cdot x + 3 \cdot x^2}{6} \right) - \frac{x \cdot x^2 \cdot (3 + x)}{2} + x \cdot (3 + x) + 1;$$

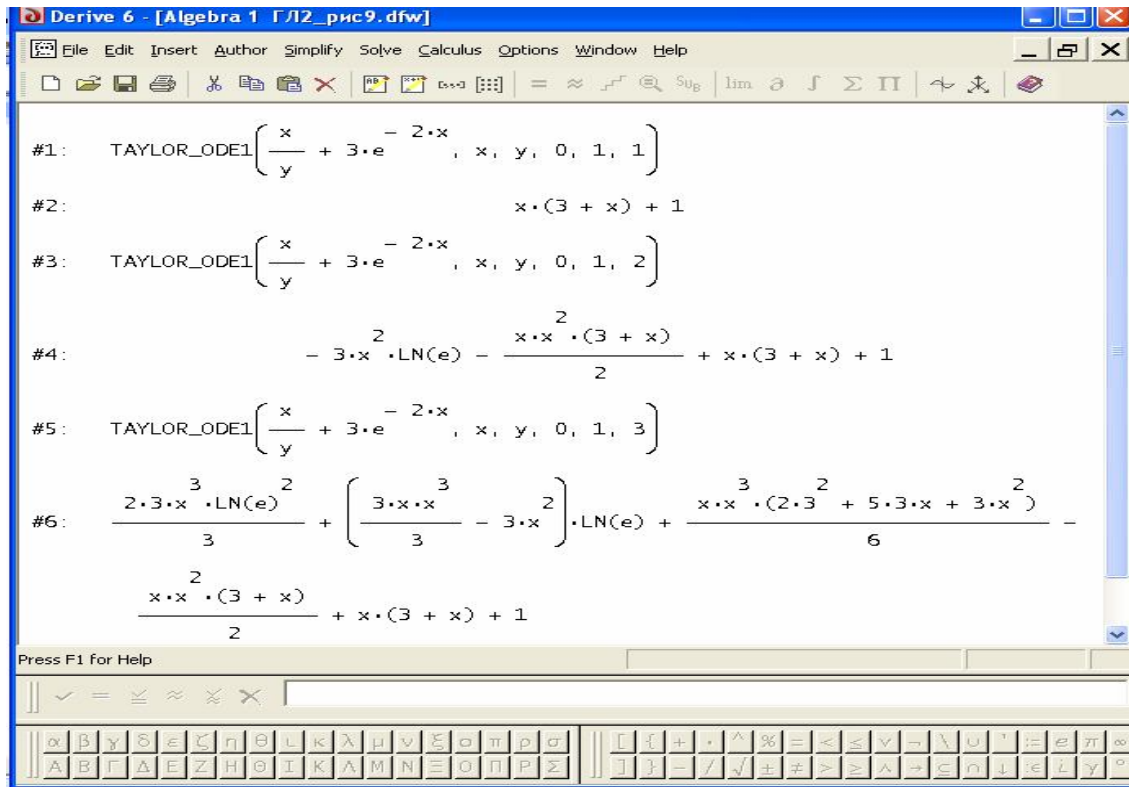


Рис.2.26. Решение дифференциального уравнения с помощью функции TAYLOR\_ODE1

Пользуясь функцией TAY\_ODE1, следует иметь в виду, что полученное решение в виде формулы пригодно в узком диапазоне аргумента, вблизи начальных условий.

**Функция PICARD.** Функция уточняет решение при известном приближении  $p(x)$ . Она имеет вид:

$$\text{PICARD}(f(x,y), p, x, y, x_0, y_0).$$

Обозначения переменных здесь очевидны.

**Пример 14.** Пусть решением уравнения

$$\frac{dy}{dx} = \frac{x}{y} + 3e^{-2x}$$

является решение первой степени  $3x + 1$ , полученный в примере 13. Уточним решение. В данном случае функция PICARD будет иметь вид:

$$\text{PICARD}(x/y + 3e^{-2x}, 3x + 1, x, y, 0, 1).$$

В результате решения уравнения получим следующий ответ, по-



казанный на рис. 2.27.

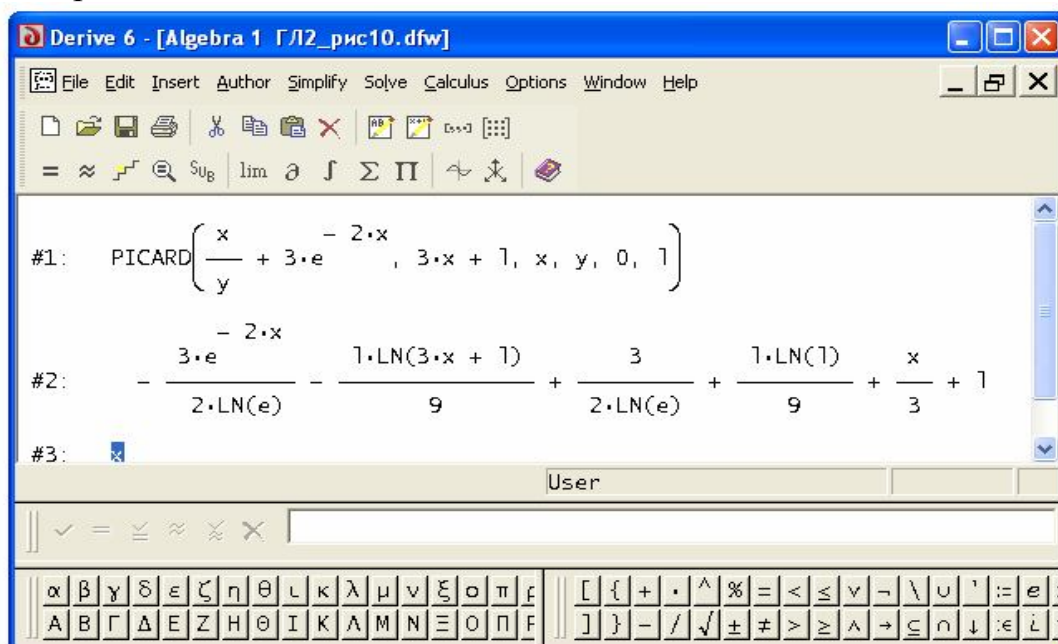


Рис.2.27. Использование функции PICARD

**Функция RK.** Система символьной математики Derive 6 содержит несколько десятков функций решения дифференциальных уравнений первого и второго порядков в символьном и числовом виде. Эти функции позволяют решать уравнения специального вида, с представлением решений в аналитическом виде, в виде чисел и числовых рядов. Наиболее важной является функция RK, которая дает решение дифференциальных уравнений и систем методом Рунге-Кутты.

Функция RK имеет вид:

RK (г, v, v0, h, n), где:

- г— вектор правых частей уравнений, разрешенных относительно производных (при этом система уравнений представляется в виде, когда каждое из уравнений является уравнением первого порядка);
- v— вектор неизвестных;
- v0— вектор начальных условий;
- h — шаг интегрирования;
- n— число итераций.

#### Примечание

Векторы представляются в виде квадратных скобок. Квадратные скобки опускаются, если решается дифференциальное уравнение первого порядка.

Откликом при реализации функции RK является матрица неизвестных с числом строк  $n$  и шагом  $h$ . Функция RK находится в файле ODE\_APPR.mth, к которому необходимо обратиться перед решением уравнений.

Технология решения дифференциальных уравнений с помощью этой функции следующая:

- обращение к файлу ODE\_APPR.mth— **File | Load |Utility | ODE\_APPR.mth;**

- ввод функции  $RK(r, v, v_0, h, n)$  (на экране формируется функция);
- выполнение команды **Approximate** (на экране формируется ответ в виде матрицы).

При решении дифференциальных уравнений высокого порядка целесообразно, перед вводом функции  $RK$ , ввести поочередно правые части всех уравнений. После ввода они будут расположены на экране в отдельных строках, помеченных символами  $\#k$ , где  $\#k$  ( $k=1,2,3,...,n$ ) номера строк, в которых находятся правые части всех  $n$  дифференциальных уравнений. Только после этого можно ввести функцию  $RK$ , записав вектор  $r$  в виде

$[\#1, \#2, \dots, \#n]$ .

**Пример 15.** Пусть необходимо решить следующее дифференциальное уравнение  $y'' + xy' - 2y + 1 = 0$  с начальными условиями  $y(0)=1, y'(0)=2$ .

Прежде чем обратиться к системе Derive 6, представим уравнение второго порядка в виде системы двух дифференциальных уравнений, каждое из которых первого порядка, разрешенное относительно производной. Такие преобразования выполняются методом подстановок. Обозначим  $y' = z$ . Тогда система дифференциальных уравнений будет иметь вид:

$$\begin{aligned} y' &= z; \\ z' &= -xz + 2y - 1. \end{aligned}$$

Теперь запишем функцию  $RK$ :  $RK([z, -x \cdot z + 2 \cdot y - 1], [x, y, z], [0, 1, 2], 0.2, 10)$  После ввода этой функции и вызова команды **Approximate** получим ответ (рис.2.28).

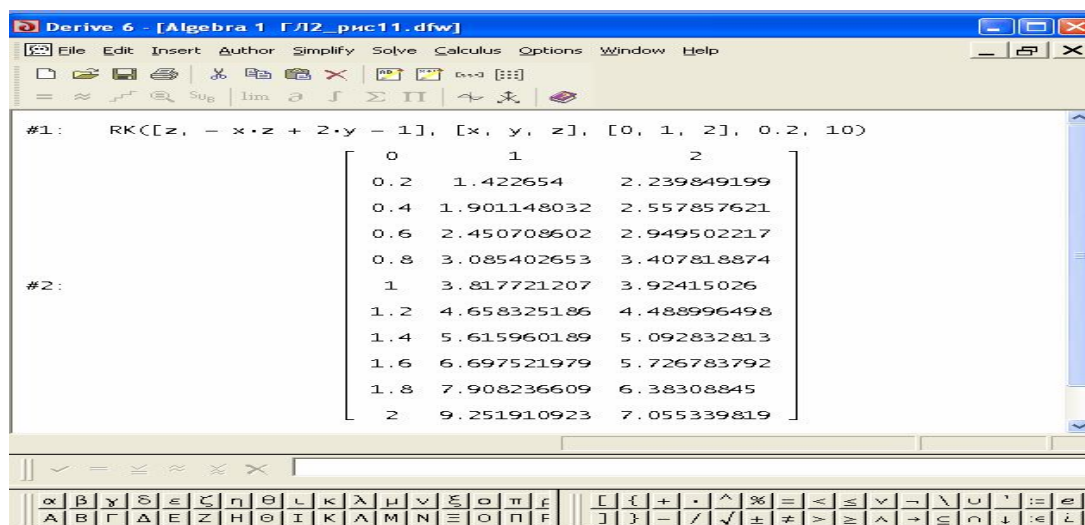


Рис.2.28. Решение дифференциального уравнения с помощью функции Рунге-Кутта

**Пример 16.** Пусть необходимо решить методом Рунге-Кутта следующую систему линейных дифференциальных уравнений:

$$\begin{aligned} dP_0(t)/dt &= -\lambda P_0(t) + \mu P_1(t); \\ dP_1(t)/dt &= -\mu P_1(t) + \lambda P_0(t); \end{aligned}$$

Начальные условия решения задачи и переменные имеют следующие значения:

$a = \lambda = 1, m = \mu = 0.5, t = 3$ . Шаг интегрирования  $h = 0,1$ ; число итераций  $n = 10, P_0 = x, P_1 = y$ .

Заданная система дифференциальных уравнений описывает функционирование одноканальной системой массового обслуживания с отказами. В данном случае  $\lambda$  это интенсивность поступления заявок в систему,  $\mu$  – это интенсивность обслуживания. Рациональной может быть следующая технология решения системы уравнений в среде Derive 6:

- запись обеих правых частей в систему уравнений и ввод; после ввода системы уравнений  $RK([- a \cdot x + m \cdot y, a \cdot x - m \cdot y], [t, x, y], [0, 1, 0], h, n)$  – на экране формируется функция RK с вектором правых частей системы дифференциальных уравнений);
- ввод с помощью команды **Simplify < Variable Substitution**, вызываемой нажатием кнопки на панели инструментов, значений переменных  $a, m, t, h, n$  (на экране формируется функция RK с численными значениями переменных);
- выполнение команды **Approximate** (на экране формируется решение в виде матрицы с  $n$  строками и четырьмя столбцами  $t, x, y$ ).

Описанное выше решение показано на рис.2.29.

$RK([- a \cdot x + m \cdot y, a \cdot x - m \cdot y], [t, x, y], [0, 1, 0], h, n)$

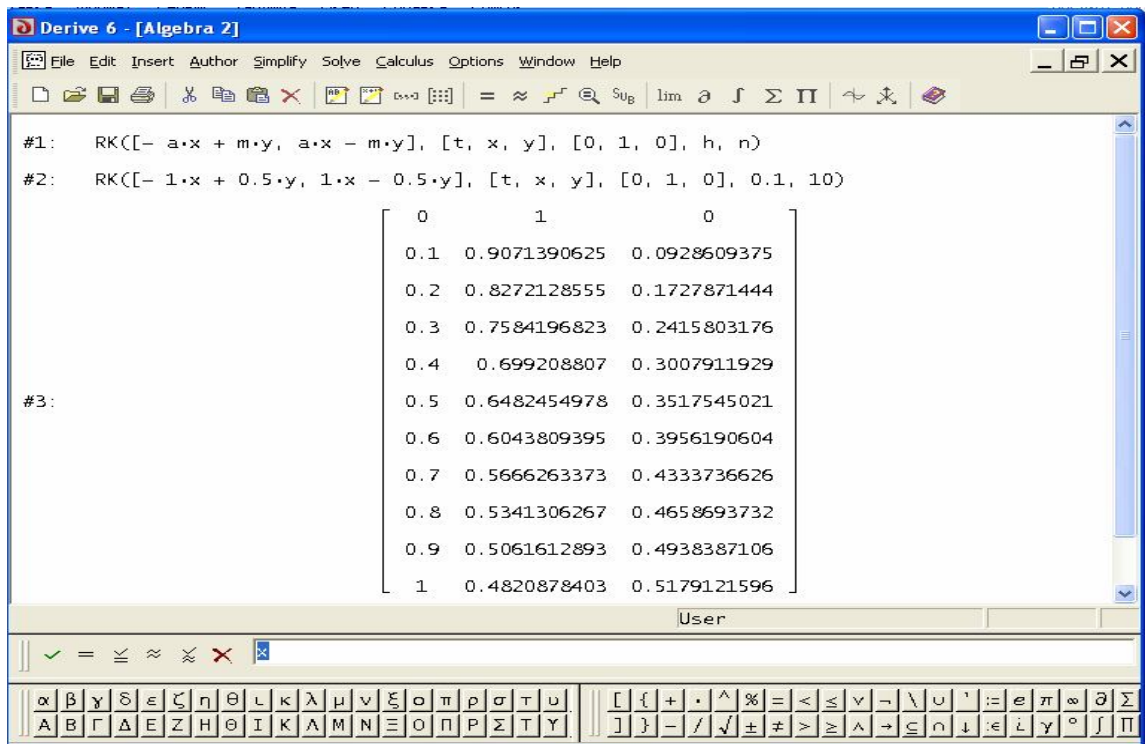


Рис.2.29. Определение вероятностей состояний системы массового обслуживания

Рассмотрим типичный пример решения дифференциального уравнения.

**Пример 17.** Дано дифференциальное уравнение  $y'' + 3y' + y = x \cdot e^x$ . Необходимо:

- найти общее и частное решения при начальных условиях  $(0), (1), (0), (1)$ ;
- найти решение в явном виде;
- представить решение в виде графика;
- проверить правильность решения задачи.

Решение задачи.

Определение общего и частного решения.

Структуре нашего уравнения соответствует функция DSOLVE2.

В данном случае —  $p(x) = 3$ ,  $q(x) = 1$ ,  $r(x) = x \cdot e^x$ . Тогда функция DSOLVE2 будет иметь вид:

➤ в случае общего решения —  $\text{DSOLVE2}(3, 1, x \cdot e^x, x, c1, c2)$ ;

➤ в случае частного решения —  $\text{DSOLVE2\_BV}(3, 1, x \cdot e^x, x, 0, 1, 0, 1)$ ;

ИЛИ  $\text{DSOLVE2\_IV}(3, 1, x \cdot e^x, x, 0, 1, 1)$ .

После набора и ввода функций необходимо щелкнуть мышью поочередно по кнопке **Simplify**. В результате получим:

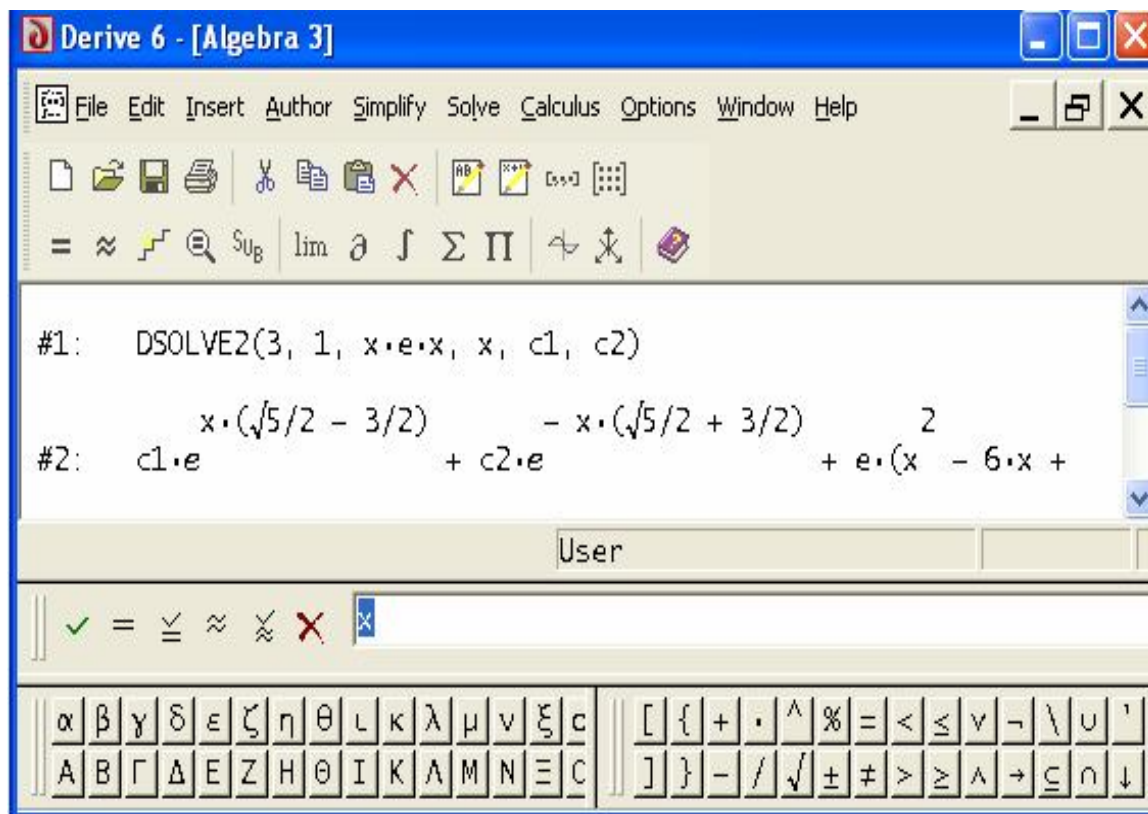


Рис.2.30. Общее решение

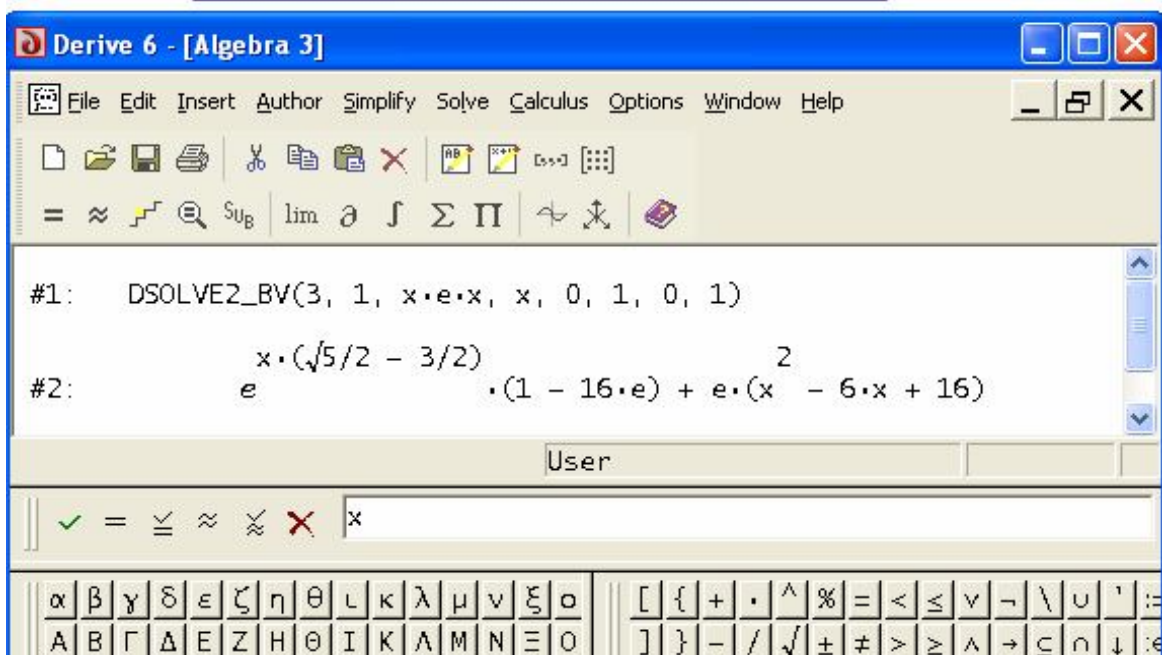


Рис.2.31. Частное решение

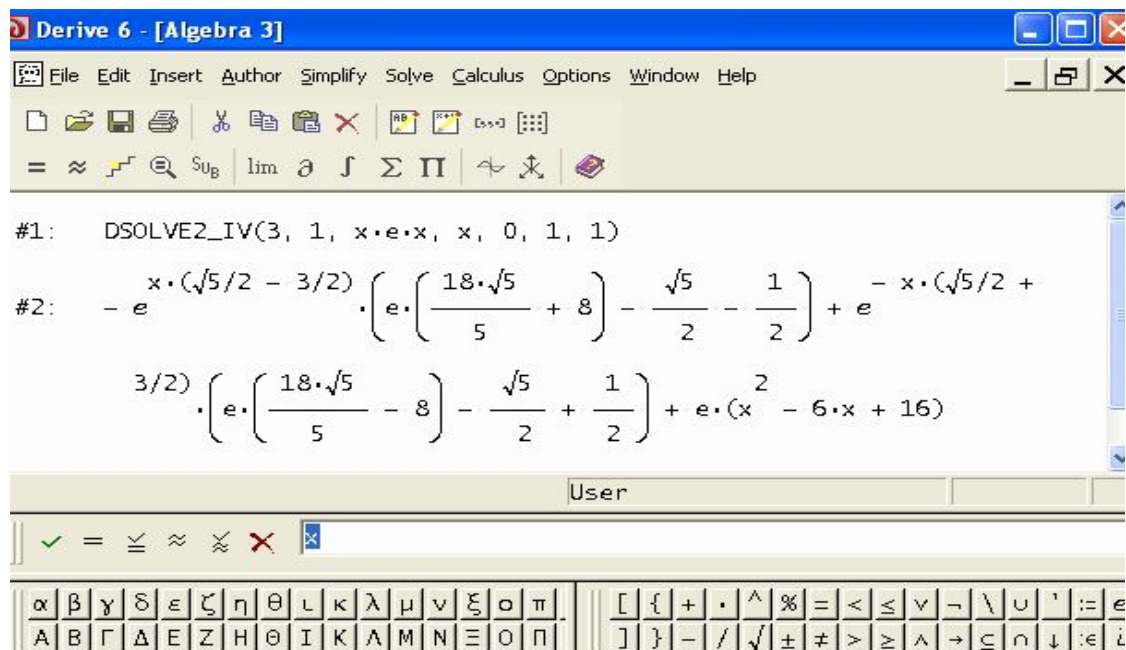


Рис.2.32. Частное решение

В нашем случае решение получено в явном виде. Представление решения в виде графика. Полученное решение является функцией  $y = \varphi(x)$ . Ее график получим с помощью функции PLOT обычным способом (рис. 2.33).

Решение дифференциального уравнения может быть получено в неявном виде. В этом случае его необходимо разрешить относительно искомой неизвестной. В противном случае нельзя проверить правильность решения задачи.



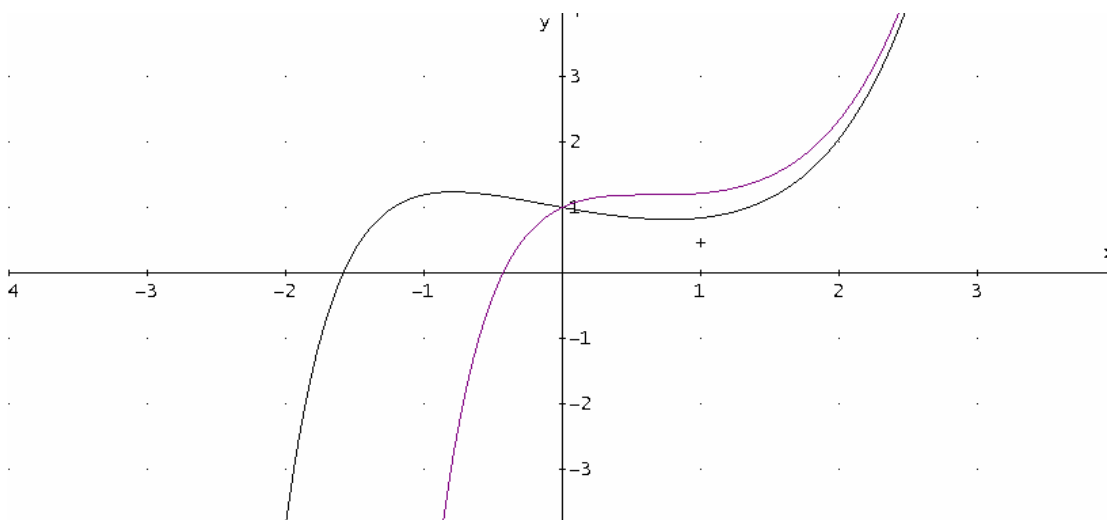


Рис.2.33. График решения дифференциального уравнения  $y'' + 3y' + y = x * e^x$

Рассмотренные нами функции аналитических методов решения дифференциальных уравнений позволяют получить решение уравнений первого и второго порядка. При анализе функционирования технических, информационных и других сложных систем приходится решать уравнения более высокого порядка. В этом случае пользуются численными методами. Между тем система Derive 6, как и другие универсальные математические средства символической математики, позволяет получить аналитические решения, используя приближенные методы, рассмотренные в настоящей главе. Для этого необходимо составить программу на языке системы.

Среди приближенных численных методов наибольшее применение получил метод Рунге-Кутты

**Пример 18.** Пусть необходимо решить методом Рунге-Кутты дифференциальное уравнение  $y''' + xy'' + yz - 1 = 0$  при начальных условиях:  $y(0) = y'(0) = y''(0) = 1$ . Представим уравнение в виде системы уравнений. Для этого обозначим:  $y' = x, x' = z$ . Подставляя новые переменные в исходное уравнение, получим следующую систему дифференциальных уравнений, каждое из которых первого порядка:

$$\begin{aligned} y' &= x; \\ x' &= z; \\ z' &= -xz - yz + 1. \end{aligned}$$

Запишем функцию RK в виде:

$$\text{RK}([x, z, -x \cdot z - y \cdot z + 1], [t, x, y, z], [0, 1, 1, 1], 0.1, 10).$$

В нашем случае будет получено решение в виде таблицы, состоящей из четырех столбцов и двадцати строк с шагом  $h = 0.1$ . Решение уравнения в диапазоне от 0 до 1 показано на рис. 2.34.

В системе Derive 6 имеется функция, выделяющая из матрицы A столбцы j и k. Она может быть использована для графического изображения решения дифференциального уравнения. Функция имеет следующий вид:

$$\text{EXTRACT\_2\_COLUMNS}(A, j, k).$$

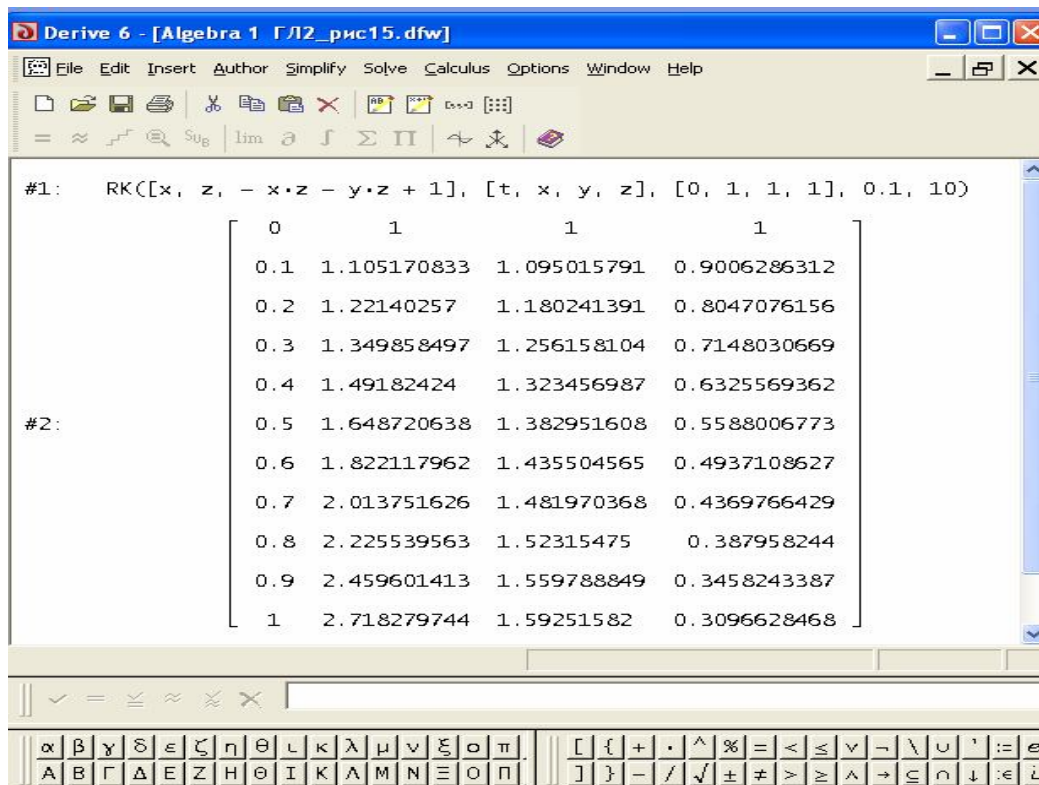


Рис.2.34. Решение системы дифференциальных уравнений методом Рунге-Кутта

Воспользуемся ею и представим наше решение в виде графика. Технология построения графика функции  $y = f(t)$  относительно проста и включает ряд действий.

➤ Образование матрицы, состоящей из двух столбцов  $t$  и  $y$ , которые в матрице решения нашего дифференциального уравнения находятся в первом и третьем столбцах. Пусть матрица решения уравнения находится в четвертой строке экрана (#4) из рис.2.34. Тогда функция выделения столбцов матрицы будет иметь вид:

EXTRACT\_2\_COLUMNS (#1, 1, 3) .

После ввода этой функции и нажатия кнопки **Approximate** (расположена на панели инструментов) на экране образуется матрица(рис.2.35).

➤ После выделения получившейся матрицы необходимо щелкнуть по кнопке 2D-plot window (расположена на панели инструментов).

В результате на экране формируется графическое окно 2D-plot с изображением координатной сетки.

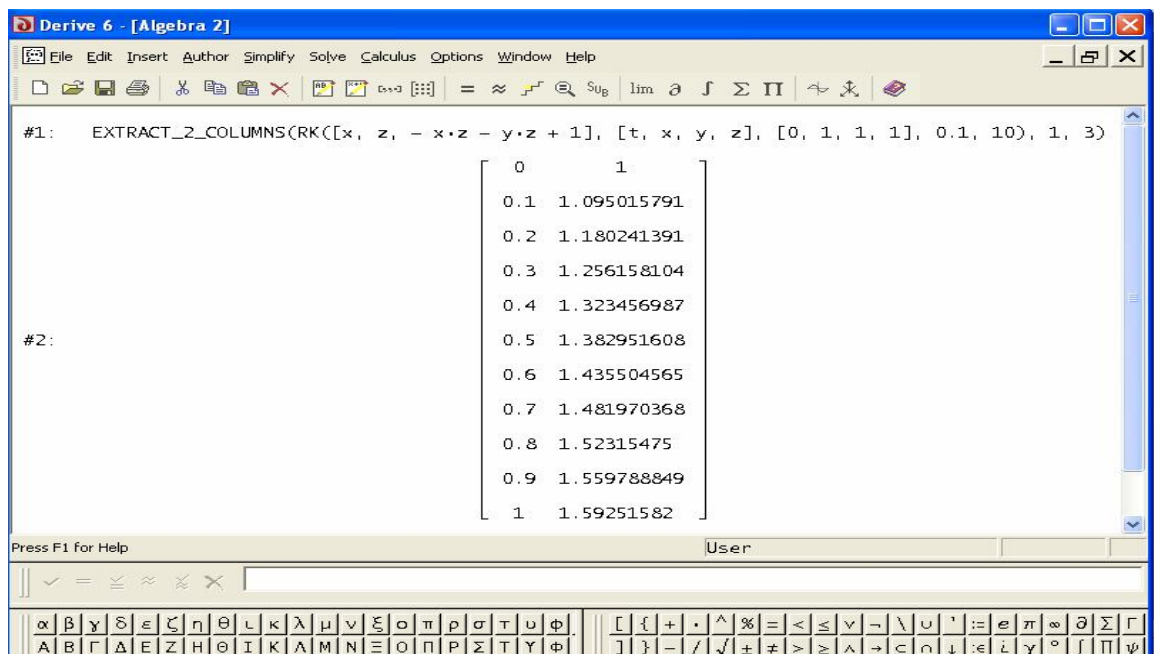


Рис. 2.35. Матрица по функции Рунге-Кутта

➤ Щелчок мышью по кнопке Plot Expression. На экране формируется график функции  $y = f(t)$ , являющейся решением дифференциального уравнения (рис. 2.36).

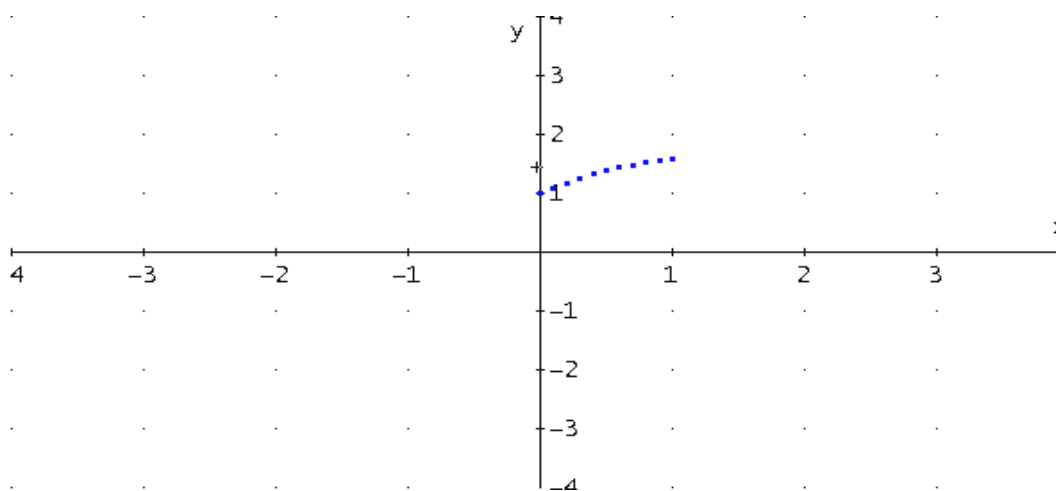


Рис.2.36. Решение дифференциального уравнения  $y''' + xy'' + yz - 1 = 0$

Достоинства и недостатки этой системы во многом субъективные. Их почувствует пользователь при решении практических задач или примеров, которые приведены в приложении 1.



## ГЛАВА 3. МОДЕЛИРОВАНИЕ ОБЪЕКТОВ В СРЕДЕ MAPLE

Сегодня на персональном компьютере можно решать задачи научно-технического характера, не прибегая к их кодированию на каком-нибудь алгоритмическом языке (Бейсик, Паскаль, СИ). Интегрированные программные системы автоматизации математических расчетов (Derive, Maple, Mathematica, MatLab, MathCAD и др.) позволяют решать поставленные задачи на входном языке, который максимально приближен к естественному математическому языку.

С каждой новой версией Maple добавляются новые математические пакеты расширения и применения для решения задач. Для изучения приемов работы и методик решения физических задач в системе Maple мы рекомендуем книги [1-3].

### 3.1. Общие сведения о системе Maple

Интерфейс данной системы адаптирован для работы пользователя, имеющего элементарные навыки работы с Windows-приложениями.

Под интерфейсом понимается не только легкое управление системой, как с клавишного пульта, так и с помощью мыши, но и просто набор необходимых символов, формул, текстовых комментариев с последующим запуском документов (Worksheets) в реальном времени. Запустив систему Maple из Windows, вы увидите на экране окно (рис.3.8). Над ним видна строка с основными элементами интерфейса. Опции главного меню, содержащиеся в этой строке, легко изучить самостоятельно с помощью опцию Help; некоторые из них очень похожи на стандартные опции, принятые в текстовых редакторах Windows.

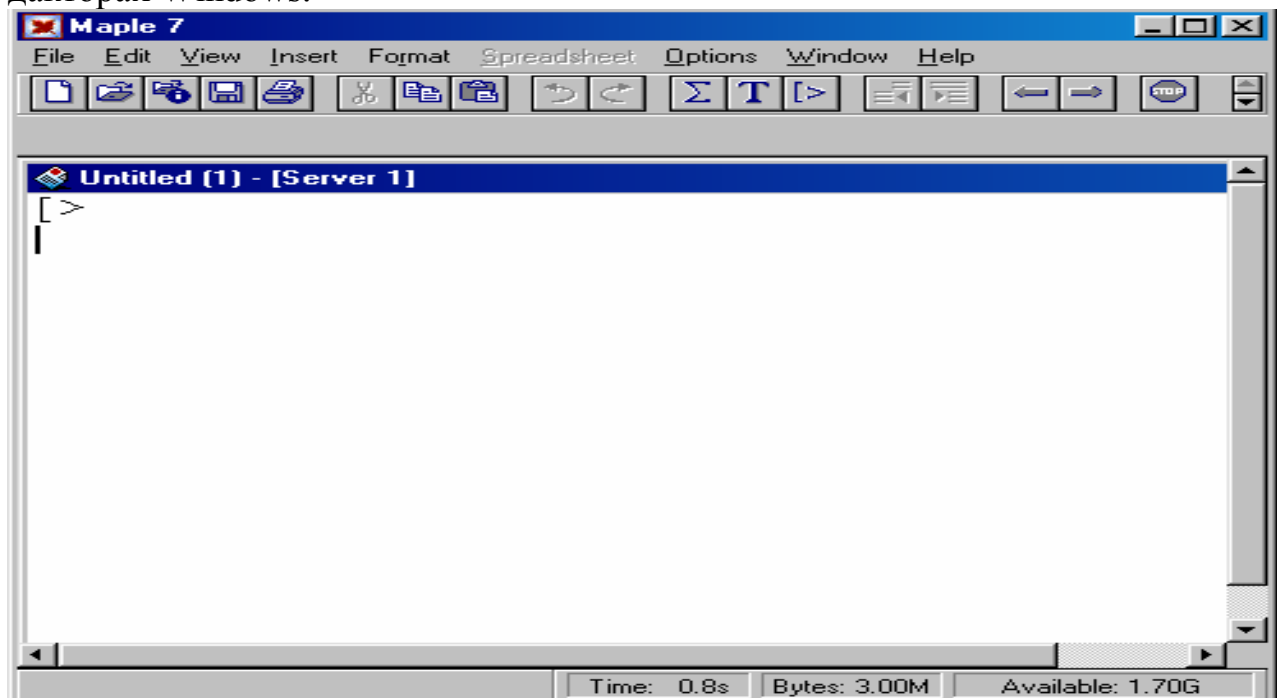


Рис. 3.8

Система Maple является мощной системой программирования, здесь реализованы три языка, так называемые язык входной – интерпретирующего типа похожего на Бейсик, язык реализации - им является универсальный язык программирования C, и язык процедурного программирования – Maple-язык. В этой системе можно проводить символьные и численные вычисления, численное и символьное решение уравнений, вычисление математических функций, возможности работы с матрицами, графической визуализации вычислений и многое другое.

Остановимся на применении системы Maple для решения дифференциальных уравнений. Для решения простых обыкновенных дифференциальных уравнений (ОДУ) и систем (ОДУ) используется процедура пакета Maple `dsolve`, которая в общем виде записывается следующим образом:

**`dsolve ( { ODE, nach }, { funcs }, type solution).`**

Здесь ODE – идентификатор, которому присваивается дифференциальное уравнение (система ОДУ) или само ДУ;

*nach* – начальные условия;

*funcs* – определяемые функции;

*type solution* – опция, задающая метод аналитического или численного решения. Этот параметр определяет тип используемого метода аналитического или численного решения ОДУ. При задании этого параметра в виде:

- *explicit* – выдает явное решение;
- *formal series* – решение в форме степенного многочлена;
- *series* – решение в виде ряда с порядком, которое указывается перед функцией `dsolve` в виде `Order:=8`;
- *integral transform* – решение на основе интегральных преобразований Лапласа, Фурье и др.;
- *numeric* – решение в численном виде.

Если мы не указываем этот параметр, то система сама выбирает предпочтительный метод решения по умолчанию.

Рассмотрим кратко процесс решения дифференциальных уравнений и моделирования физических задач в системе Maple [1-3] на примере уравнения

$$\frac{dy}{dx} = -ay . \quad (3.8)$$

Среда допускает произвольную запись нахождения решения. Она может быть оформлена через операторы присвоения, например

```
➤ ODE:=diff(y(x),x)=-a*y(x)
➤ Nach:=y(0)=5;
➤ dsolve({ODE,Nach},y(x));
```

(3.9)

или уравнения, начальные значения, определяемые функции, используемый метод решения записываются в одной строке, например

```
➤ dsolve({diff(y(x),x)=-a*y(x),y(0)=5},y(x));
```

(3.10)

возможна также их комбинация. Полученное решение может быть присвоено какому-либо оператору

➤ `F:=dsolve({diff(y(x),x)=-a*y(x),y(0)=5},y(x));`

Решение дифференциальных уравнений второго и более высоких порядков может быть описано с помощью функции `diff`, но в производной по аргументу ставится знак `$` и порядок производной. Например,

➤ `dsolve({diff(y(x),x$2)=-a*sin(x),y(0)=5,y(1)=0},y(x)); (3.10a)`

Решение систем дифференциальных уравнений может быть проведено различными методами выбором параметра *type solution* в явном виде, в виде разложения в ряд или ином другом виде. Чаще всего для начальных оценок используются различные численные методы.

Если мы в опции `dsolve` после типа решения *numeric* указываем `method = classical`, то система Maple выдаст решение с помощью одного из восьми версий классического метода. В системе Maple имеются и другие методы решения:

- `rkf45` - метод Рунге – Кутты –Фейхсберга 4-5-го порядка;
- `dverk78` - метод Рунге – Кутты 7-8-го порядка;
- `gear` - одношаговый метод Гира;
- `lsode` - одна из версий Ливенморского решателя дифференциальных уравнений;
- `taylorseries` – метод разложения в ряд Тейлора.

Ниже при решении систем дифференциальных уравнений численным методом мы не указывали метод численного решения, при этом система выбирает метод решения по умолчанию. Для построения решения дифференциальных уравнений в виде графиков может быть использована опция графического пакета **plots**.

➤ `with(plots);`

[`animate`, `animate3d`, `animatecurve`, `changecoords`, `complexplot`, `complexplot3d`, `conformal`, `contourplot`, `contourplot3d`, `coordplot`, `coordplot3d`, `cylinderplot`, `densityplot`, `display`, `display3d`, `fieldplot`, `fieldplot3d`, `gradplot`, `gradplot3d`, `implicitplot`, `implicitplot3d`, `inequal`, `listcontplot`, `listcontplot3d`, `listdensityplot`, `listplot`, `listplot3d`, `loglogplot`, `logplot`, `matrixplot`, `odeplot`, `pareto`, `pointplot`, `pointplot3d`, `polarplot`, `polygonplot`, `polygonplot3d`, `polyhedra_supported`, `polyhedraplot`, `replot`, `rootlocus`, `semilogplot`, `setoptions`, `setoptions3d`, `spacecurve`, `sparsematrixplot`, `sphereplot`, `surfdata`, `textplot`, `textplot3d`, `tubeplot` ]

➤ `odeplot(F,[x,y(x)],0..200);`

или использовать равнозначную запись

➤ `plots[odeplot](F,[x,y(x)],0..200);`

Графический пакет системы Maple содержит более пятидесяти графических функций процедур. Для визуализации результатов мы здесь использовали несколько графических функций.

В Maple имеется мощный инструментальный пакет **DEtools** решения дифференциальных уравнений и систем таких уравнений. Вызов этого пакета

осуществляется с помощью слова **with**.

> **with (DEtools) ;**

[*DEnormal, DEplot, DEplot3d, DEplot\_polygon, DFactor, DFactorLCLM, DFactorsols, Dchangevar, GCRD, LCLM, PDEchangecoords, RiemannPsols, Xchange, Xcommutator, Xgauge, abelsol, adjoint, autonomous, bernoullisol, buildsol, buildsym, canoni, caseplot, casesplit, checkrank, chinisol, clairautsol, constcoeffsols, convertAlg, convertsys, dalembertsol, dcoeffs, de2diffop, dfieldplot, diffop2de, dsubs, eigenring, endomorphism\_charpoly, equinv, eta\_k, eulersols, exactsol, expsols, exterior\_power, firint, firtest, formal\_sol, gen\_exp, generate\_ic, genhomosol, gensys, hamilton\_eqs, indicialeq, infgen, initialdata, integrate\_sols, intfactor, invariants, kovacicsols, leftdivision, liesol, line\_int, linearsol, matrixDE, matrix\_riccati, moser\_reduce, muchange, mult, mutest, newton\_polygon, normalG2, odeadvisor, odepde, parametricsol, phaseportrait, poincare, polysols, ratsols, redode, reduceOrder, reduce\_order, regular\_parts, regularsp, remove\_RootOf, riccati\_system, riccatisol, rifsimp, rightdivision, rtaylor, separablesol, solve\_group, super\_reduce, symgen, symmetric\_power, symmetric\_product, symtest, transinv, translate, untranslate, varparam, zoom*]

Как видно из вызова функций, в этом пакете заложено очень много возможностей решения и анализа, различных ОДУ и систем дифференциальных уравнений. Для детального рассмотрения других встроенных графических процедур системы Maple и ознакомлению с этими возможностями можно использовать работу [1].

Для решения и визуализации результатов вычислений дифференциальных уравнений можно также использовать графические функции этого пакета **DEplot** и **DEplot3d**, которые решают и сразу строят двухмерные и трехмерные графики решения дифференциальных уравнений.

## 3.2. Примеры моделирования в системе Maple

Рассмотрим ряд задач из практикума [4] и приведем их решения с помощью системы компьютерной математики Maple.

**Пример 1.** Движение небесного тела в гравитационном поле описывается системой уравнений

$$\begin{aligned}\frac{dv_x}{dt} &= -GM \frac{x}{(x^2 + y^2)^{3/2}}, \\ \frac{dv_y}{dt} &= -GM \frac{y}{(x^2 + y^2)^{3/2}}, \\ \frac{dx}{dt} &= v_x, \quad \frac{dy}{dt} = v_y.\end{aligned}\tag{3.11}$$

с начальными условиями, определенными двумя параметрами: начальной скоростью и углом  $\alpha$ .

$$\begin{aligned}
 x(0) &= x_0, \quad y(0) = 0, \\
 v_{x0} &= v_0 \cos \alpha, \quad v_{y0} = v_0 \sin \alpha.
 \end{aligned}
 \tag{3.12}$$

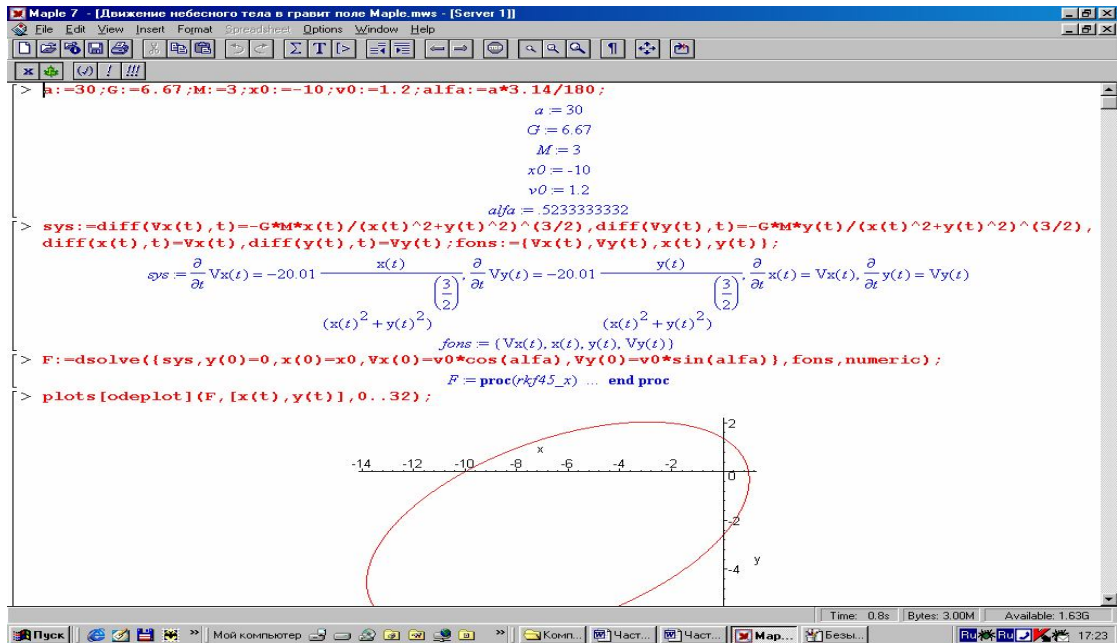
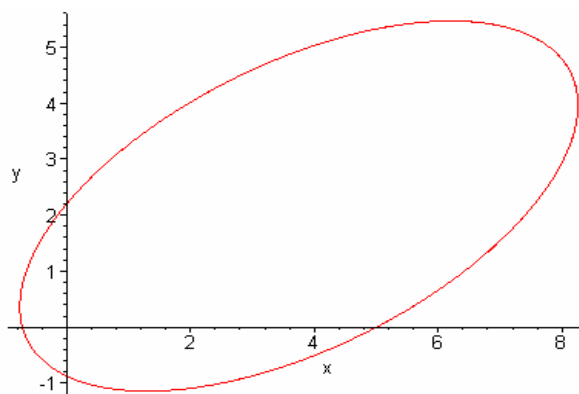


Рис.3.9

Листинг программы в Maple приведен на рис.3.9. Ниже проведено исследование движения небесного тела в зависимости от начальной скорости  $v_0$ .

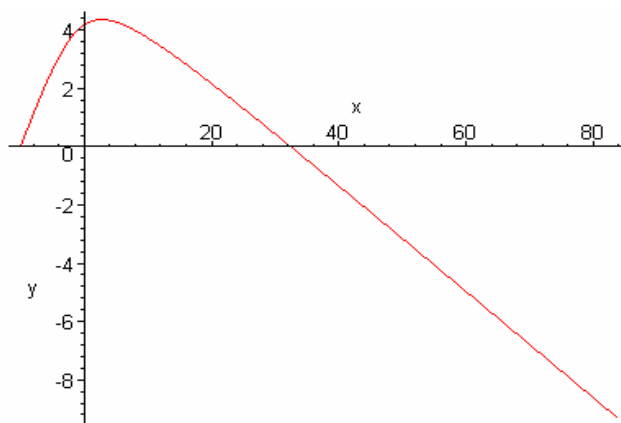
- $v_0 := 2; sys := diff(Vx(t), t) = -G * M * x(t) / (x(t)^2 + y(t)^2)^{3/2},$   
 $diff(Vy(t), t) = -G * M * y(t) / (x(t)^2 + y(t)^2)^{3/2},$   
 $diff(x(t), t) = Vx(t), diff(y(t), t) = Vy(t):$
- $fons := \{Vx(t), Vy(t), x(t), y(t)\}:$
- $F := dsolve(\{sys, y(0) = 0, x(0) = 5, Vx(0) = v_0 * \cos(alfa),$   
 $Vy(0) = v_0 * \sin(alfa)\}, fons, numeric):$
- $plots[odeplot](F, [x(t), y(t)], 0..200);$

$v_0 := 2$



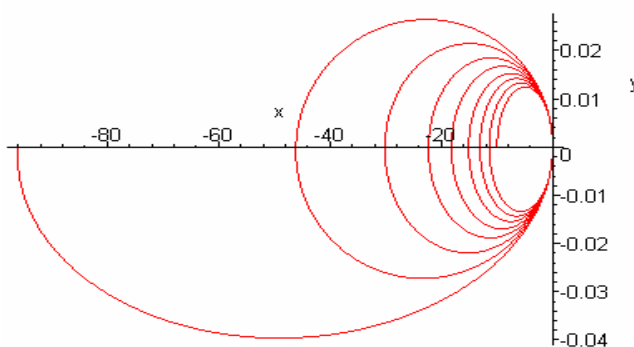
- `v0:=3.5;sys:=diff(Vx(t),t)=-G*M*x(t)/(x(t)^2+y(t)^2)^(3/2),`  
`diff(Vy(t),t)=-G*M*y(t)/(x(t)^2+y(t)^2)^(3/2),`  
`diff(x(t),t)=Vx(t),diff(y(t),t)=Vy(t);`
- `fons:={Vx(t),Vy(t),x(t),y(t)};`
- `F:=dsolve({sys,y(0)=0,x(0)=x0,Vx(0)=v0*cos(alfa),`  
`Vy(0)=v0*sin(alfa)},fons,numeric);`
- `plots[odeplot](F,[x(t),y(t)],0..30);`

$$v0 := 3.5$$



- `v0:=0.005;sys:=diff(Vx(t),t)=-G*M*x(t)/(x(t)^2+y(t)^2)^(3/2),`  
`diff(Vy(t),t)=-G*M*y(t)/(x(t)^2+y(t)^2)^(3/2),`  
`diff(x(t),t)=Vx(t),diff(y(t),t)=Vy(t):`  
`fons:={Vx(t),Vy(t),x(t),y(t)}:`  
`F:=dsolve({sys,y(0)=0,x(0)=x0,Vx(0)=v0*cos(alfa),`  
`Vy(0)=v0*sin(alfa)},fons,numeric):`  
`plots[odeplot](F,[x(t),y(t)],0..650);`

$$v0 := .005$$



**Пример 2. Движение заряженного тела в кулоновском поле ядра.**  
 Рассмотрим технологию решения задачи движения заряженной частицы в кулоновском поле ядра в системе MAPLE

- > `Q:=79;q:=2;m:=4;v0:=14;a:=10/180*3.14;`  

$$Q := 79$$

$$q := 2$$

$$m := 4$$

$$v0 := 14$$

`a := .1744444444`

`> yz0 := y(0) = 0, x(0) = -1, vx(0) = v0*cos(a), vy(0) = v0*sin(a);`

`yz0 := y(0) = 0, x(0) = -1, vx(0) = 13.78752359, vy(0) = 2.429854566`

`> sys := diff(vx(t), t) = Q*(q/m)*x(t) / (x(t)^2 + y(t)^2)^(3/2),  
diff(x(t), t) = vx(t), diff(vy(t), t) = (Q/m)*q*y(t) / (x(t)^2 + y(t)^2)^(3/2),  
diff(y(t), t) = vy(t); funcs := {x(t), vx(t), y(t), vy(t)};`

$$\text{sys} := \frac{\partial}{\partial t} vx(t) = \frac{79}{2} \frac{x(t)}{(x(t)^2 + y(t)^2)^{(3/2)}}, \frac{\partial}{\partial t} x(t) = vx(t), \frac{\partial}{\partial t} vy(t) = \frac{79}{2} \frac{y(t)}{(x(t)^2 + y(t)^2)^{(3/2)},$$

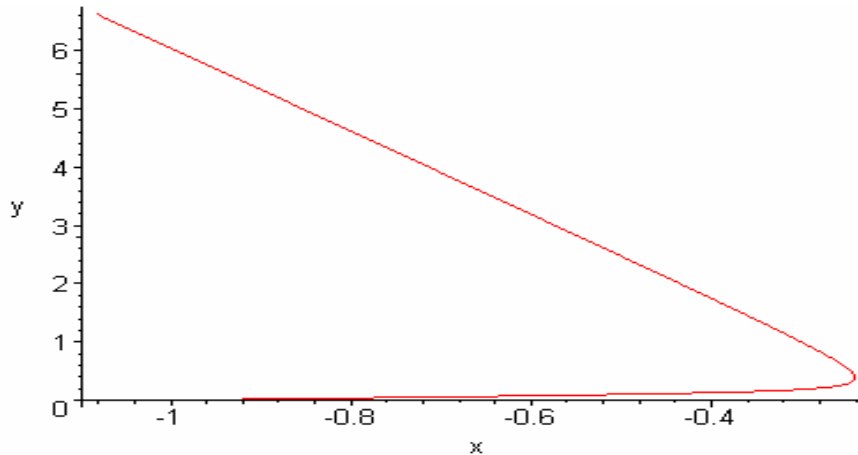
$$\frac{\partial}{\partial t} y(t) = vy(t)$$

`fons := {vx(t), x(t), y(t), vy(t)}`

`> F := dsolve({sys, yz0}, funcs, numeric);`

`F := proc (rkf45_x) ... end proc`

`> plots[odeplot](F, [x(t), y(t)], 0..0.5);`



### Пример 3. Двойной маятник

Колебания двойного маятника (на математическом маятнике с одной массой, подвешен другой с меньшей массой) описываются системой дифференциальных уравнений

$$\begin{aligned} \frac{d^2 x_1}{dt^2} &= -\omega^2(1 + 2\mu)x_1 + \mu\omega^2 x_2, \quad \dot{x}_1(0) = 0, \quad x_1(0) = x_{10}, \\ \frac{d^2 x_2}{dt^2} &= -\omega^2 x_2 + \omega^2 x_1, \quad \dot{x}_2(0) = x_2(0) = 0, \end{aligned} \quad (3.13)$$

где  $x_1$  и  $x_2$  - отклонения тяжелого и легкого маятников от положения равновесия,  $\mu = \frac{m}{M}$  - малое число, а  $m$  и  $M$  - массы легкого и тяжелого маятников,

$\omega^2 = \frac{g}{l}$ ,  $g$  - ускорение свободного падения,  $l$  - длина нити, на которой подвешен легкий маятник. Предполагается, что приведенные длины маятников

равны, т.е.  $L=l$ .

### Решение

```
> m1:=1; m2:=0.001; mu:=m2/m1; g:=3.8; l:=1; omega2:=g/l;  
    ml := 1
```

```
    m2 := .001
```

```
    μ := .001
```

```
    g := 9.8
```

```
    l := 1
```

```
    ω2 := 9.8
```

```
➤ x0:=1; apply(x1,t); apply(x2,t);
```

```
➤ apply(diff(x1(t),t),t); apply(diff(x1(t),t),t);  
    x0 := 1
```

```
    x1(t)
```

```
    x2(t)
```

$$\left(\frac{\partial}{\partial t} x1(t)\right)(t)$$
$$\left(\frac{\partial}{\partial t} x1(t)\right)(t)$$

```
➤ sys:=diff(x1(t),t$2)=-omega2*(1+2*mu)*x1(t)+mu*omega2*x2(t),  
    diff(x2(t),t$2)=-omega2*x2(t)+omega2*x1(t);
```

$$sys := \frac{\partial^2}{\partial t^2} x1(t) = -9.8196 x1(t) + .0098 x2(t), \frac{\partial^2}{\partial t^2} x2(t) = -9.8 x2(t) + 9.8 x1(t)$$

```
> nach:=x1(0)=x0,x2(0)=x0,D(x1)(0)=0,D(x2)(0)=0;
```

```
    nach := x1(0) = 1, x2(0) = 1, D(x1)(0) = 0, D(x2)(0) = 0
```

```
> fun:={x1(t),x2(t)};
```

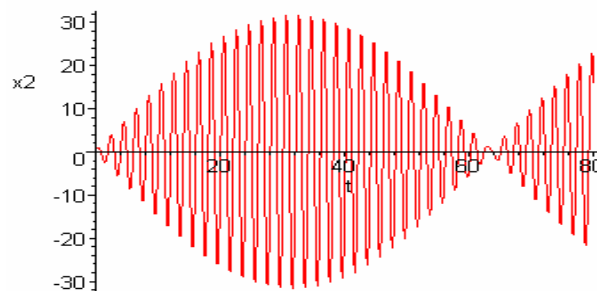
```
    fun := {x1(t), x2(t)}
```

```
> F:=dsolve({sys,nach},fun,numeric);
```

```
    F := proc(rkf45_x) ... end proc
```

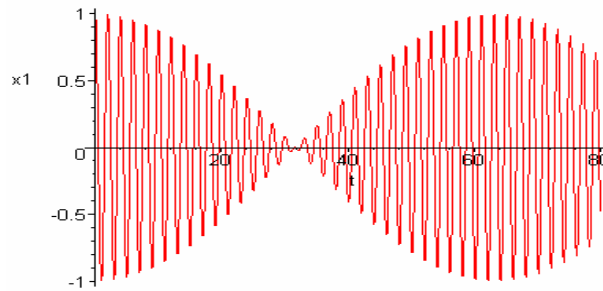
```
> with(plots):
```

```
> odeplot(F,[t,x2(t)],0..80,numpoints=25);
```

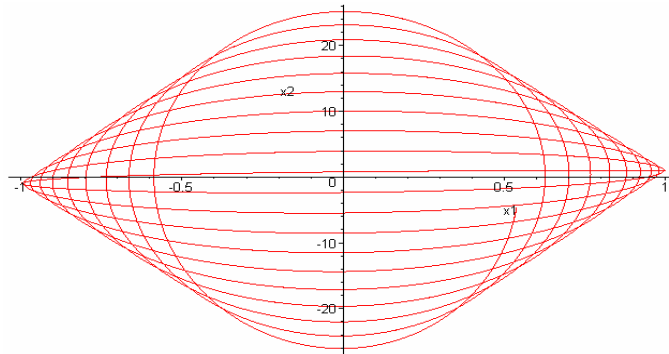


```
> odeplot(F,[t,x1(t)],0..80,numpoints=25);
```





```
> odeplot(F, [x1(t), x2(t)], -20..20);
```



**Пример 4. Выпрямление переменного напряжения с фильтрацией**  
описывается математической моделью

$$\frac{du(t)}{dt} = \frac{1}{rC} \left[ E(t) - \left( 1 + \frac{r}{R_H} \right) u(t) \right], \quad (3.14)$$

$$u(0) = 0.$$

Здесь  $C$  – емкость конденсатора фильтра,  $r$  – сопротивление схемы фильтра,  $R_H$  – сопротивление нагрузки,  $E(t)$  – вид выпрямленного напряжения после одно или двухполупериодного выпрямления.

```
> r:=1;R:=40;C:=0.1;
```

```
r:=1
```

```
R:=40
```

```
C:=.1
```

```
> E(t):=abs(sin(t));
```

```
E(t):=|sin(t)|
```

```
> pr:=diff(u(t),t)=1/r*C*(E(t)-(1+r/R)*u(t));fons:=u(t);
```

```
> F:=dsolve({pr,u(0)=0},fons,type=numeric,output=listprocedur  
e);
```

$$pr := \frac{\partial}{\partial t} u(t) = .1 |\sin(t)| - .1025000000 u(t)$$

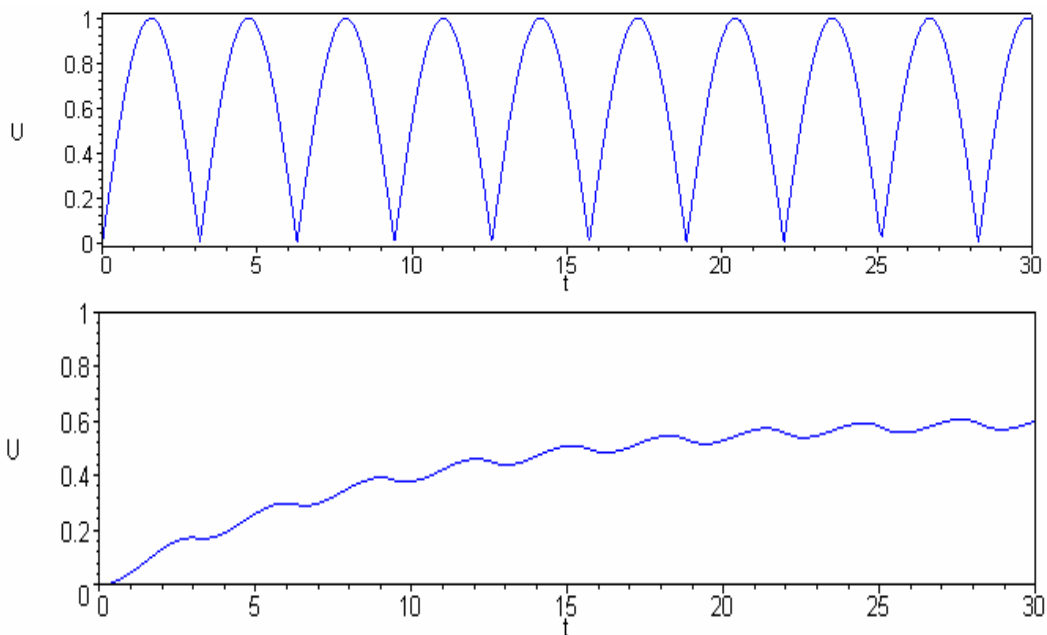
```
fons:=u(t)
```

```
F:= [t=(proc(t) ... end proc), u(t)=(proc(t) ... end proc)]
```

```
> Y:=subs(F,u(t));
```

```
Y:=proc(t) ... end proc
```

```
>plot(E(t),t=0..30,color=blue,labels=[t,U]);
plot(Y,0..30,0..1,color=blue,labels=[t,U]);
```



```
> r:=2;R:=40;C:=0.1;E(t):=((sin(t)+abs(sin(t)))/2);
      r:=2
      R:=40
      C:=.1
```

$$E(t) := \frac{1}{2} \sin(t) + \frac{1}{2} |\sin(t)|$$

- `pr:=diff(u(t),t)=1/r*C*(E(t)-(1+r/R)*u(t));fons:=u(t);`
- `F:=dsolve({pr,u(0)=0},fons,type=numeric,output=listprocedure);`

$$pr := \frac{\partial}{\partial t} u(t) = .02500000000 \sin(t) + .02500000000 |\sin(t)| - .05250000000 u(t)$$

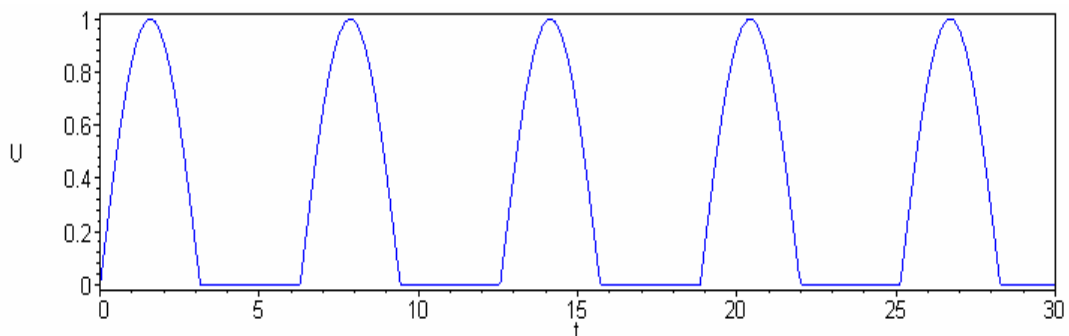
`fons := u(t)`

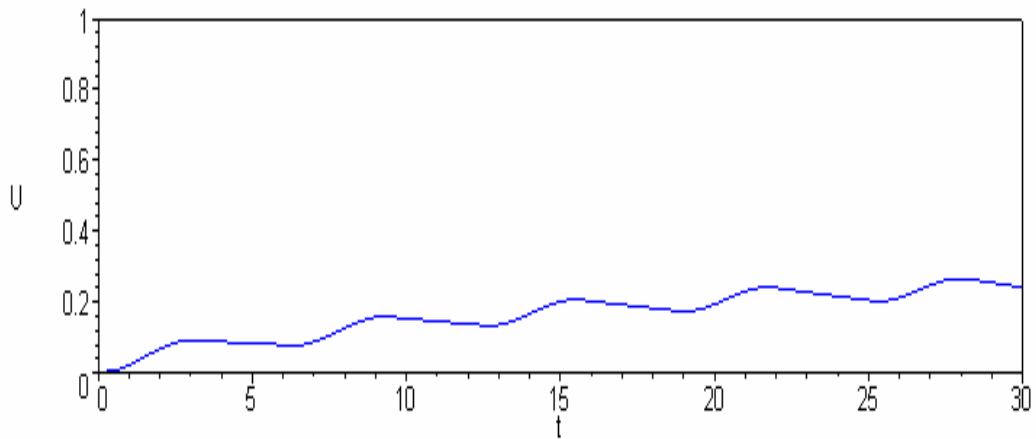
`F := [t=(proc(t) ... end proc), u(t)=(proc(t) ... end proc)]`

```
> Y:=subs(F,u(t));
```

`Y:=proc(t) ... end proc`

```
>plot(E(t),t=0..30,color=blue,labels=[t,U]);
plot(Y,0..30,0..1,color=blue,labels=[t,U]);
```





**Пример 5. Явление миража.** Ниже приведен пример с выбором классического метода *classical* численного решения задачи о мираже.

Математическая модель для явления миража имеет вид

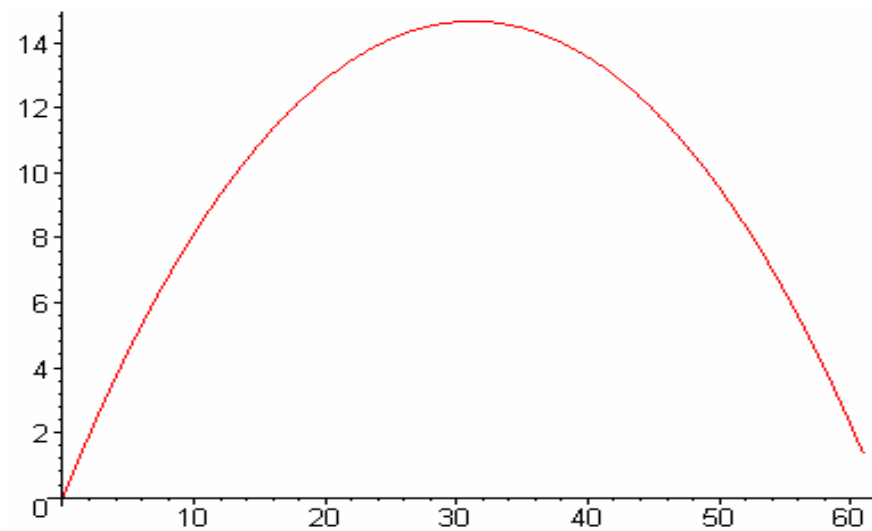
$$\frac{d}{dx} \left( \frac{dy}{dx} \right) = \frac{1}{l_0^2} (-\gamma n_0 + \gamma^2 y) \quad (3.15)$$

$$\frac{dy}{dx} \Big|_{x=0} = \text{ctg } \alpha_0 ,$$

здесь  $l_0 = n_0 \sin \alpha_0$  .

Листинг решения уравнений (3.15) в системе Maple:

```
> restart;
> n0:=3.0004;g:=0.02;
                                n0 := 1.0004
                                g := .02
> a0:=evalf(45*3.14/180);
                                a0 := .7850000000
> l0:=n0*sin(a0);
                                l0 := .7071079112
> sys:=diff(v(x),x)=(1/(10*10))*(-
g*n0+g*g*y(x),diff(y(x),x)=v(x);
                                sys :=  $\frac{\partial}{\partial x} v(x) = -.04001587211 + .0007999974432 y(x), \frac{\partial}{\partial x} y(x) = v(x)$ 
> y0:=v(0)=cot(a0),y(0)=0;
                                y0 := v(0) = 1.000796644, y(0) = 0
➤ p:=dsolve({sys,y0},{y(x),v(x)},numeric,method=classical);
                                p := proc(x_classical) ... end proc
> plots[odeplot](p,[x,y(x)],0..61);
```



**Рекомендации** при использовании системы Maple:

- 1) система различает прописные и строчные буквы;
- 2) необходимо строго соблюдать открытие и закрытие как фигурных, так и круглых скобок;
- 3) Необходимо аккуратно следить за знаками двоеточия и точки с запятой.

## ГЛАВА 4. МОДЕЛИРОВАНИЕ ОБЪЕКТОВ В СРЕДЕ MATHCAD

MathCAD является математическим редактором, позволяющим проводить разнообразные научные и инженерные расчеты, начиная от элементарной арифметики и заканчивая сложными реализациями численных методов. Пользователи MathCAD - это студенты, ученые, инженеры, разнообразные технические специалисты. Благодаря простоте применения, наглядности математических действий, обширной библиотеке встроенных функций и численных методов, возможности символьных вычислений, а также превосходному аппарату представления результатов (графики самых разных типов, мощных средств подготовки печатных документов и Web-страниц), MathCAD стал наиболее популярным математическим приложением. В соответствии с проблемами реальной жизни, математикам приходится решать одну или несколько из следующих задач:

- ввод на компьютере разнообразных математических выражений (для дальнейших расчетов или создания документов, презентаций, Web-страниц);
- проведение математических расчетов;
- подготовка графиков с результатами расчетов;
- ввод исходных данных и вывод результатов в текстовые файлы или файлы с базами данных в других форматах;
- подготовка отчетов работы в виде печатных документов;
- подготовка Web-страниц и публикация результатов в Интернете;
- получение различной справочной информации из области математики.

У разных фирм различны подходы к обозначению новых версий своих программных продуктов. MathSoft, Inc., к примеру, за какие-то пять лет породила добрый десяток новых версий популярной системы Mathcad — 3.0, 4.0, 5.0, Plus 4.0, 6.0, Plus 6.0, 7.0, Plus 7.0, 8.0, 8.0 PRO и даже Mathcad 2000 PRO/Premium. И почти каждый раз отмечала их новой цифрой, хотя революционными отличия этих версий друг от друга назвать трудно.

В этой среде описание записи и решения математических задач дается с помощью привычных математических формул и знаков. Такой же вид имеют и результаты вычислений. Поэтому среда MathCAD вполне оправдывает аббревиатуру CAD (Computer Aided Design), говорящую о принадлежности к сложным и продвинутым информационным системам проектирования и моделирования (ИСПРиМ).

Подробно изучить приемы работы в среде MathCAD можно по специальным руководствам. Мы же остановимся на применении пакета для решения обыкновенных дифференциальных уравнений. Отметим, что технология работы в данной среде проста и очень нравится студентам. “Общение” с MathCAD повышает интерес студентов к математике, информатике и физике. Студенты получают начальные, профессиональные знания и привыкают к профессиональному языку.

В этой среде описание записи и решения математических задач дается с помощью привычных математических формул и знаков. Такой же вид име-

ют и результаты вычислений. Поэтому среда MathCAD вполне оправдывает аббревиатуру CAD (Computer Aided Design), говорящую о принадлежности к сложным и продвинутым информационным системам проектирования и моделирования (ИСПРиМ).

## 4.1. Общие сведения о математическом пакете MathCAD

Интерфейс данного пакета адаптирован для работы пользователя, имеющего элементарные навыки работы с Windows-приложениями.

Под интерфейсом понимается не только легкое управление MathCAD, как с клавишного пульта, так и с помощью мыши, но и просто набор необходимых символов, формул, текстовых комментариев с последующим запуском документов (Worksheets) в реальном времени.

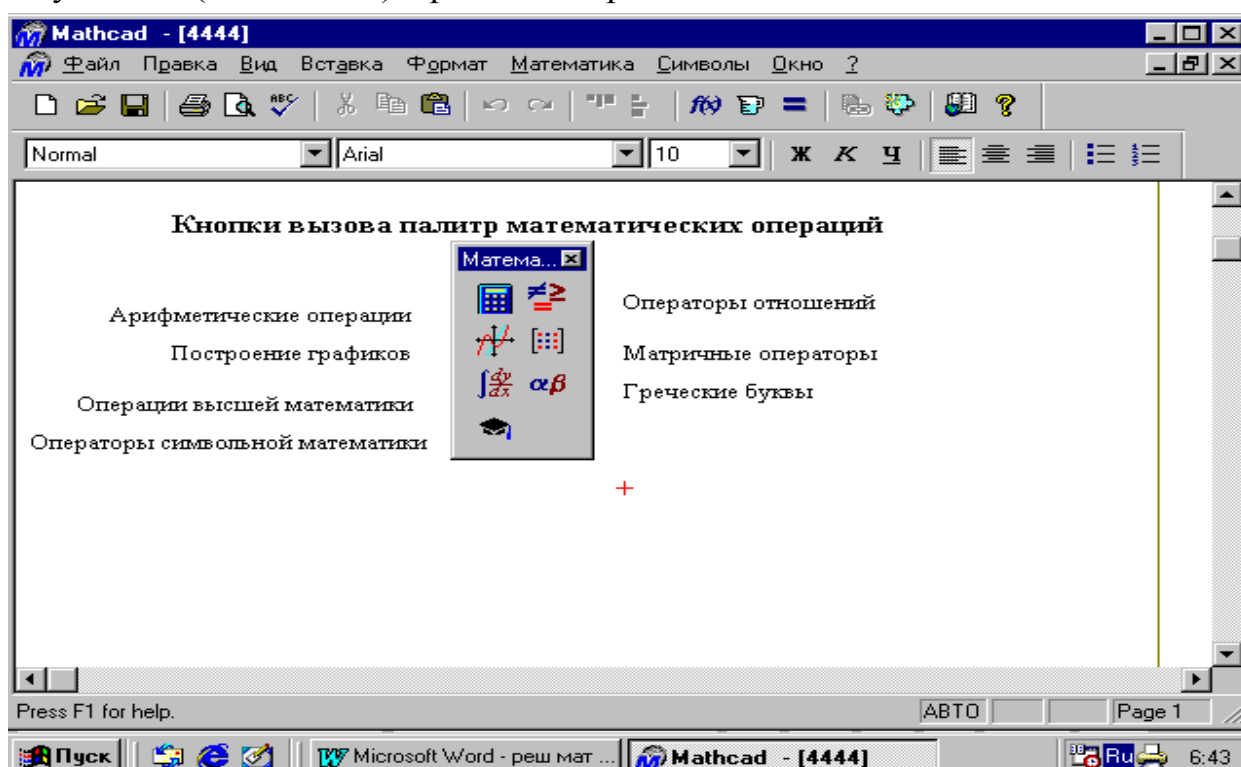


Рис. 4.10

Запустив систему MathCAD из Windows, вы увидите на экране окно (рис. 4.10). Над ним видна строка с основными элементами интерфейса. Опции главного меню, содержащиеся в этой строке, легко изучить самостоятельно; некоторые из них очень похожи на стандартные опции, принятые в текстовых редакторах Windows.

Работа с документами MathCAD обычно не требует обязательного использования возможностей главного меню, так как основные из них дублируются кнопками (пиктограммами) на панели инструментов, которые расположены в удобных, перемещаемых с помощью мыши наборных панелях-палитрах. Наборные панели появляются в окне редактирования документов при активизации кнопок. Они служат для вывода заготовок – шаблонов математических знаков (цифр, знаков арифметических операций, матриц, зна-

ков интеграла, производных, пределов и др.). Указатель мыши подводим к надписи “Вид” в главном меню, щелкаем левой кнопкой мыши; указатель подводим к “Панели инструментов” и щелкаем левой кнопкой мыши; выпадает следующее меню. Указатель мыши подводим к надписи “Математика” и щелкаем левой кнопкой мыши. Выпадают наборные панели по различным разделам математики.

## 4.2. Способы моделирования в MathCAD

Компьютерное моделирование в MathCAD можно провести разными способами:

- с использованием дискретных аналогов математической модели,
- с использованием готовых процедур пакета,
- с созданием собственных процедур (программированием).

Рассмотрим эти способы на примерах решения конкретных задач.

**Пример 1.** Решить задачу движения тела, брошенного под углом к горизонту и построить его график

Система дифференциальных уравнений, описывающая данный процесс, имеет вид

$$\begin{aligned} \dot{x} &= -kx, & x(0) &= v_0 \cos \alpha, & x(0) &= 0, \\ \dot{y} &= -ky - g, & y(0) &= v_0 \sin \alpha, & y(0) &= 0. \end{aligned} \quad (4.1)$$

**Решение с использованием разностных уравнений.** Для решения задачи вводим начальные значения и параметры в окне MathCAD.

$$\begin{aligned} v &:= 0.5 & g &:= 4.8 & a &:= 3.1415/4 \\ x_0 &:= 0 & y &:= 0 & h &:= 0.01 \\ i &:= 1 \dots 1000 \end{aligned}$$

Алгоритм решения дифференциальных уравнений, описывающих процесс движения тела брошенного под углом к горизонту, имеет вид

$$\begin{aligned} x_1 &:= v \cdot \cos(a) & y_1 &:= v \cdot \sin(a) \\ x_{i+1} &:= 2 \cdot x_i - x_{i-1} & y_{i+1} &:= 2 \cdot y_i - y_{i-1} - h^2 \cdot g \end{aligned}$$

Здесь мы использовали в качестве алгоритма разностную аппроксимацию второй производной левой части уравнений (4.1), в результате такой записи получаем решение. Явно это решение не видно. Чтобы показать решение проведем его визуализацию, через построение графика. Для этого указатель мыши подводим к пиктограмме “Построение графиков” и входим в него, активизируем “Декартов график”. Появляется шаблон для построения графика. На нем выделены метки. Указатель мыши подводим к нижней метке, активизируем. Набираем  $x_i$ . Появляются по горизонтали еще две метки, где мы должны указать интервалы построения графика. Указатель мыши подводим к левой метке, щелкая левой кнопкой мыши, активизируем и вводим левую границу 0. Указатель мыши подводим к правой метке, активизируем и вводим 300. Уводим указатель мыши к метке оси Y, активизируем

его и вводим  $y_i$ . Появляются метки нижней и верхней границ оси Y. В нижней набираем 0, в верхней 100. Отводим указатель мыши от шаблона для графиков, щелкаем левой кнопкой мыши. Появляется искомый график (рис.4.11).

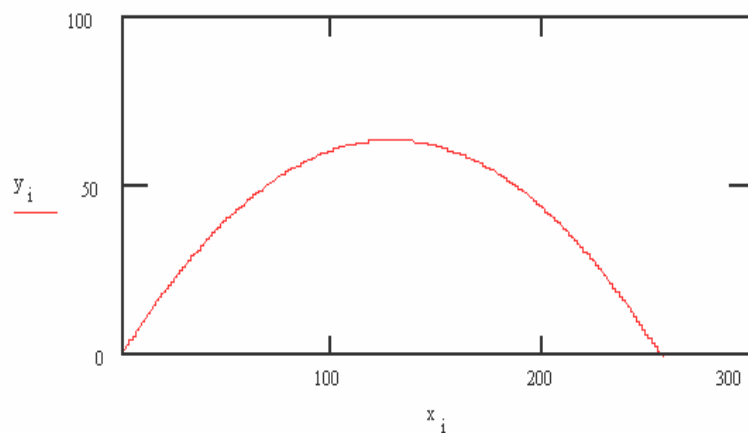
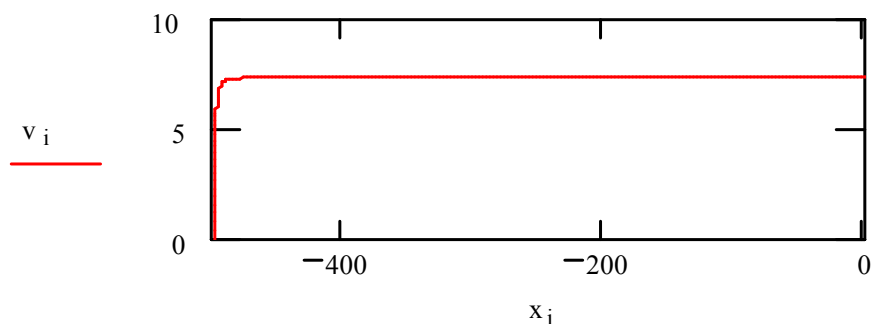


Рис.4.11

**Пример 2.** Листинг решения задачи падения парашютиста массой 70 кг, с высоты 500 м площадью парашюта 20 кв. м с использованием дискретного аналога (разностных уравнений) приведен на рис.4.12.

$$\begin{aligned}
 p &:= 1.29 & c &:= 1. & s &:= 20 & g &:= 9.8 & m &:= 70 \\
 k &:= 0.5 \cdot c \cdot s \cdot p & h &:= 0.04 & x_0 &:= -500 & v_0 &:= 0 \\
 i &:= 0 .. 2000 \\
 v_{i+1} &:= \left[ h \cdot g - k \cdot h \cdot v_i \cdot \left( \frac{v_i}{m} \right) \right] + v_i \\
 x_{i+1} &:= h \cdot v_i + x_i
 \end{aligned}$$



*Решение с использованием* Рис. 4.12 *процедур) MathCAD.* Чаще всего для решения ОДУ и системы обыкновенных уравнений используются встроенные опции (процедуры) **odesolve**, **rkfixed**, **Rkadapt**, **Radau** и другие.



**Пример 3.** Вернемся к задаче тела, брошенного под углом к горизонту. При использовании опции **odesolve** вначале вводятся постоянные параметры уравнения, затем после слова **Given**, дифференциальные уравнения и начальные условия записываются в привычном виде.

Визуализацию результатов проводим аналогично предыдущему примеру. В опции **odesolve** ( $t$ ,  $tMax$ )  $t$  – переменная интегрирования,  $tMax$  – конечная точка отрезка интегрирования. Листинг решения приведен на рис. 4.13.

$$v0 := 10 \quad g := 9.8 \quad \text{alfa} := \frac{\pi \cdot 40}{180} \quad tMax := \frac{2 \cdot v0 \cdot \sin(\text{alfa})}{g} \quad k := 0 \quad tMax = 1.312$$

Given

$$\frac{d^2}{dt^2}x(t) + k \cdot \frac{d}{dt}x(t) = 0 \quad x'(0) = v0 \cdot \cos(\text{alfa}) \quad x(0) = 0$$

$z1 := \text{Odesolve}(t, tMax)$

Given

$$\frac{d^2}{dt^2}y(t) + k \cdot \frac{d}{dt}y(t) + g = 0 \quad y'(0) = v0 \cdot \sin(\text{alfa}) \quad y(0) = 0$$

$z2 := \text{Odesolve}(t, tMax)$

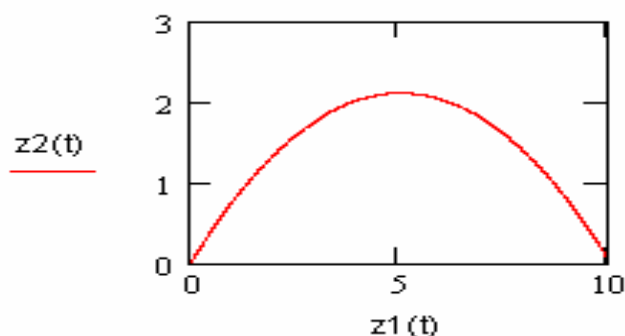


Рис.4.13

**Пример 4.** Задача о мираже. Миражом называется появление в атмосфере одного или несколько изображений отдаленных объектов (зданий, деревьев и т.п.). Оно возникает из-за полного внутреннего отражения света в атмосфере при необычном распределении плотности воздуха по вертикали. Явление миража часто наблюдается в пустыне. Математическая модель явления миража представлена в главе 3.

g := 0.00002    n0 := 1.0004    a0 := 30 ·  $\frac{\pi}{180}$

Given

$$\frac{d^2}{dx^2}y(t) = \frac{(-g \cdot n0 + g \cdot g \cdot y(x))}{10^2}$$

$$y(0) = 0 \quad y'(0) = \tan(a0)$$

y := Odesolve (x, 15000)

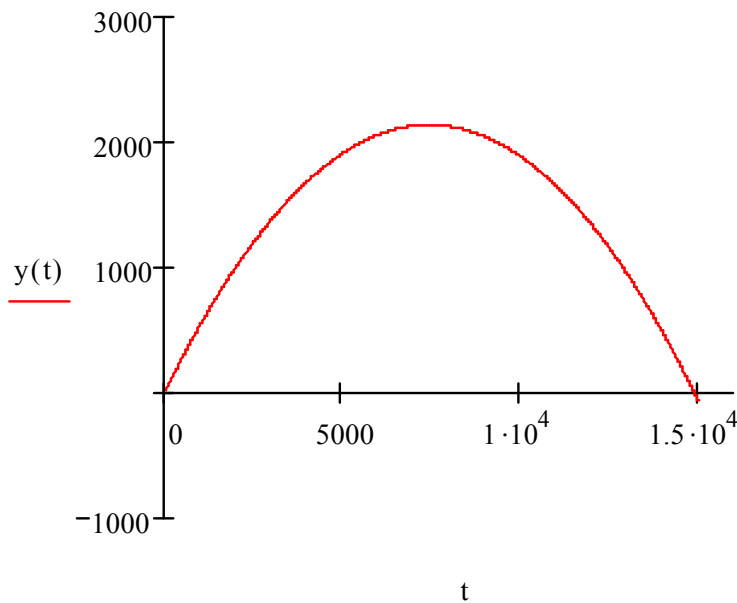


Рис.4.14

В этих листингах программы производная для начальных условий проставлена с помощью комбинации клавиш Ctrl +F7.

**Использование опции *rkfixed* (y, a, b, n, f).** Здесь воспользуемся встроенной опцией (процедурой) численного решения ОДУ по методу Рунге - Кутта с равномерным шагом **rkfixed** (y, a, b, n, f), где y – определяемая функция, [a, b] – интервал, где ищется решение, n – число точек разбиения интервала, f – правая часть уравнений. Эту опцию можно использовать для решения ОДУ или системы дифференциальных уравнений первого порядка.

**Пример 5.** Рассмотрим задачу о падающем парашютисте (рис.4.15).

p := 0.03    c := 4.    s := 20

g := 9.8    k := 0.5 · c · s · p    m := 70

$$\text{ORIGIN} := 1 \quad y := \begin{pmatrix} -500 \\ 0 \end{pmatrix} \quad f(t, y) := \begin{bmatrix} (y)_2 \\ g - k \cdot y_2 \cdot \frac{y_2}{m} \end{bmatrix}$$

$$z := \text{rkfixed} (y, 0, 25, 1000, f)$$

$$t := z^{\langle 1 \rangle} \quad x := z^{\langle 2 \rangle} \quad v := z^{\langle 3 \rangle}$$

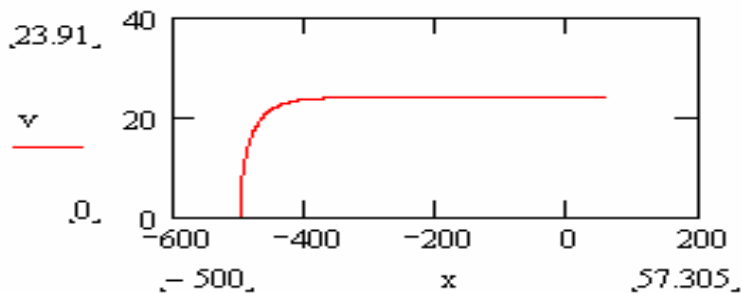


Рис. 4.15

Здесь ORIGIN=1 означает, что цикл начинается с единицы. Кроме этого проведено присвоение вычисленных значений к обозначениям сформулированной математической модели.

**Пример 6.** Рассмотрим задачу о выпрямлении тока с фильтрацией.

$$n := 5030 \quad h := 0.0001 \quad r := 3 \quad c := 0.05 \quad f := 30$$

$$w := 50 \quad E_0 := 1$$

$$E(t) := \begin{cases} E \leftarrow E_0 \cdot \sin(w \cdot t) \\ E \leftarrow |E| \quad \text{if } E < 0 \end{cases} \quad F(E, U) := \frac{1}{r \cdot c} \cdot \left[ E - \left( 1 + \frac{r}{f} \right) \cdot U \right]$$

$$z := 0$$

$$\text{Diff}(t, z) := F(E(t), z_0) \quad u := \text{rkfixed}(z, 0, n \cdot h, n, \text{Diff})$$

$$t := u^{\langle 0 \rangle} \quad U := u^{\langle 1 \rangle} \quad i := 1 \dots n$$

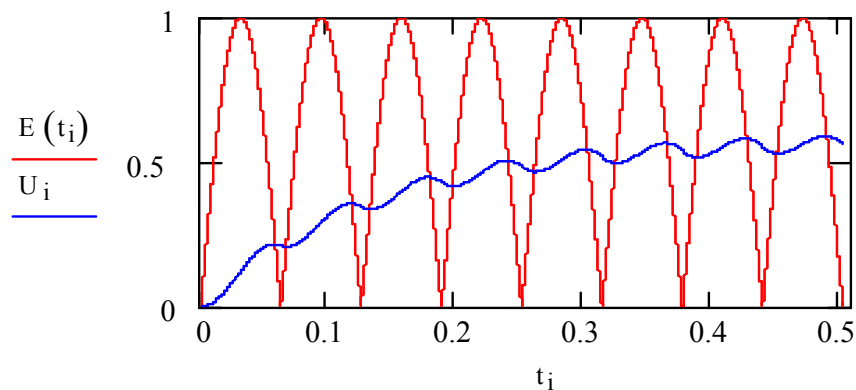


Рис.4.16

**Пример 7.** Уравнения движения для полета сверхзвукового самолета можно представить в виде

$$\begin{aligned} \frac{dv}{dy} &= \frac{F_T - Q - mg \sin \alpha}{m v \sin \alpha} = f(v, \alpha) \\ \frac{d\alpha}{dy} &= \frac{N - mg \cos \alpha}{m v^2 \sin \alpha} = \varphi(v, \alpha) \end{aligned} \quad (4.10)$$

Где сила сопротивления воздуха и подъемная сила определяется как

$$Q = \frac{1}{2} k_1 \rho S v^2 \quad N = \frac{1}{2} k_2 \rho S v^2$$

Здесь  $k_1$  – коэффициент сопротивления воздуха,  $\rho$  – плотность воздуха,  $S$  – площадь крыльев,  $k_2$  – коэффициент подъемной силы. Плотность воздуха подчиняется закону распределения Больцмана с учетом температуры

$$\rho = \rho_0 \exp\left(-\frac{mgy}{kT}\right) = \rho_0 \exp\left(-\beta \cdot y \frac{T_0}{T_0 - \alpha \cdot y}\right)$$

Сила тяги является функцией высоты и скорости и будем определять ее по полуэмпирической формуле

$$F_T = \left(1 + \frac{1}{2} \exp(-(0.001v - 1.5)^2)\right)(10^4 - 0.27y)$$

**Листинг решения.**

$$g := 9.8 \quad m := 8000 \quad k1 := 0.02 \quad k2 := 0.2 \quad p0 := 1.29 \quad k := 1.$$

$$s := 50 \quad T0 := 300 \quad k1 := 0.02 \quad n := 30000 \quad h := 0.01$$

$$b := \frac{0.125}{1000} \quad a := \frac{0.65}{100} \quad p(y) := p0 \cdot e^{\frac{-b \cdot y \cdot T0}{T0 - a \cdot y}}$$

$$Q(V, y) := \frac{1}{2} \cdot k1 \cdot p(y) \cdot s \cdot V^2 \quad N(V, y) := \frac{1}{2} \cdot k2 \cdot p(y) \cdot s \cdot V^2$$

$$Ft(V, y) := \left[1 + \frac{1}{2} \cdot e^{(0.001 \cdot V - 1.5)^2}\right] \cdot (10^4 - 0.27 \cdot y)$$

$$FUNV(V, y, \alpha) := \frac{Ft(V, y) - Q(V, y) - m \cdot g \cdot \sin(\alpha)}{m \cdot V \cdot \sin(\alpha)}$$

$$\text{FUNA}(V, y, \alpha) := \frac{N(V, y) - m \cdot g \cdot \cos(\alpha)}{m \cdot V^2 \cdot \sin(\alpha)}$$

$$D(t, z) := \begin{pmatrix} z_2 \cdot \cos(z_3) \\ z_2 \cdot \sin(z_3) \\ \text{FUNV}(z_2, z_1, z_3) \\ \text{FUNA}(z_2, z_1, z_3) \end{pmatrix} \quad \text{nach} := \begin{pmatrix} 0 \\ 0 \\ 450 \\ \frac{15 \cdot \pi}{180} \end{pmatrix}$$

UU := rkfixed(nach, 0, n·h, n, D)

t := UU<sup><0></sup>    x := UU<sup><1></sup>    y := UU<sup><2></sup>    V := UU<sup><3></sup>    α := UU<sup><4></sup>

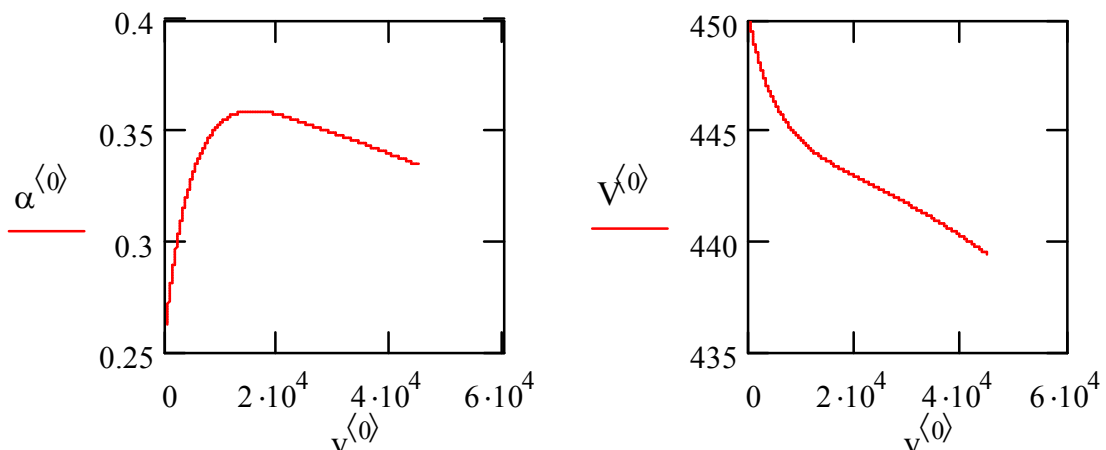


Рис.4.17

Опция **Rkadapt(y, a, b, n, f)** используется для численного решения ОДУ, но с переменным шагом. Эта опция используется для задач, имеющих быстро осциллирующие, определяемые функции.

Опция **Radau(y, a, b, n, f)** используется для численного решения жестких систем дифференциальных уравнений.

Параметры этих опций имеют аналогичный смысл, что и в опции **rkfixed**.

**Решение с помощью создания собственных процедур (программированием).** Ниже приведен листинг программы решения задачи движения тела, брошенного под углом к горизонту и с учетом сопротивления среды (падение парашютиста (рис.4.18 – 4.19)). Здесь при программировании для создания (x y) используется опция матрица из блока *Matrix*, для проведения вертикальных линий необходимо воспользоваться опцией *Add Line* из блока *Programming*.

## Движение тела, брошенного под углом к горизонту

$$g := 9.8 \quad n := 2000 \quad \text{alfa} = 40 \frac{\pi}{180} \quad v_0 := 10$$

$$t_{\text{Max}} = \frac{2 \cdot v_0 \sin(\text{alfa})}{g} \quad h := \frac{t_{\text{Max}}}{n}$$

$$x_0 := 0 \quad y_0 := 0 \quad k := -1$$

$$V_{x_0} := v_0 \cos(\text{alfa}) \quad V_{y_0} := v_0 \sin(\text{alfa})$$

$$F_x(V_x) := k \cdot V_x \quad F_y(V_y) := k \cdot V_y - g$$

```
(x y) :=
x0 ← x0
y0 ← y0
Vx0 ← v0 cos(alfa)
Vy0 ← v0 sin(alfa)
for i ∈ 0..(n-1)
  ax1 ← Fx(Vx1)
  ay1 ← Fy(Vy1)
  Vx1+1 ← Vx1 + h·ax1
  Vy1+1 ← Vy1 + h·ay1
  x1+1 ← x1 + h·Vx1
  y1+1 ← y1 + h·Vy1
  breakif (y1+1 < 0)
(x y)
```

stack(x<sup>T</sup>, y<sup>T</sup>) =

	0	1	2	3	4
0	0	5·10 <sup>-3</sup>	0.01	0.015	0.02
1	0	6·10 <sup>-3</sup>	5·10 <sup>-3</sup>	0.013	0.017

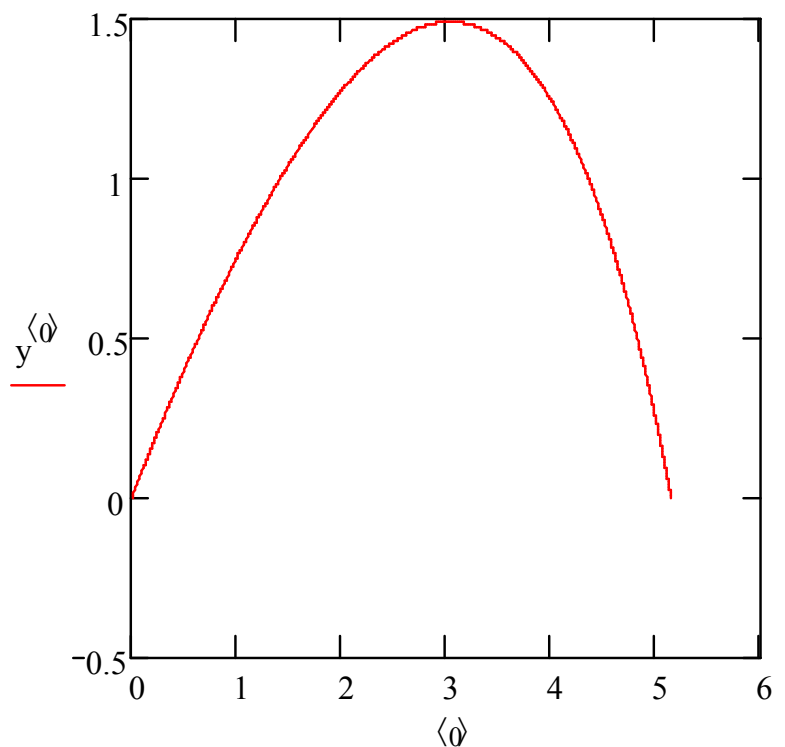


Рис.4.18

## Падение парашютиста (X убывает)

$n := 4000$   $h := 0.1$   $g := 9.8$   $v_0 := 0$   $m := 60$   $\mu := 0.95$   $r := 0.0025$   $c := 0.4$

$p := 1.29$   $S := 0.4$   $x_0 := n \cdot h$   $t_0 := 0$   $k1 := 6 \cdot \pi \cdot \mu \cdot r$   $k2 := 0.5 \cdot c \cdot S \cdot p$   $Fx(v) := g - \frac{(k1 \cdot v + k2 \cdot v \cdot v)}{m}$

```
(x v a t) :=
  x_0 ← x_0
  v_0 ← v_0
  a_0 ← g
  t_0 ← 0
  k2 ← 0.5 c · S · p
  for i ∈ 0..(n - 1)
    a_i ← g -  $\frac{(k1 \cdot v_i + k2 \cdot v_i \cdot v_i)}{m}$ 
    v_{i+1} ← v_i + h · a_i
    x_{i+1} ← x_i + h · (-v_i)
    t_{i+1} ← t_i + h
    (S ← 20) if (t_i > 3)
    k2 ← 0.5 c · S · p
    break if (x_{i+1} < 0)
  (x v a t)
```

$stack(x^T, v^T) =$

	0	1	2	3	4
0	400	400	399.902	399.706	399.412
1	0	0.98	1.96	2.94	3.92

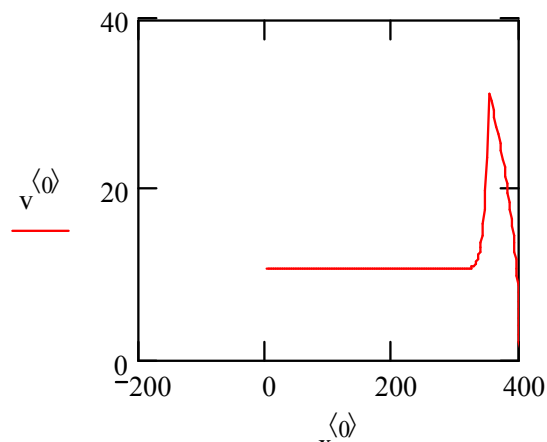
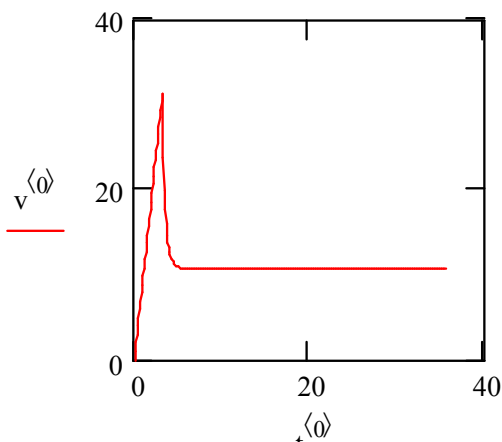


Рис.4.19

## ГЛАВА 5. МОДЕЛИРОВАНИЕ В СИСТЕМЕ MATHEMATICA

В 80-е годы возможностями символьной математики увлекся защитивший докторскую диссертацию Стивен Вольфрам (Stephen Wolfram) из США. Его интересы были столь серьезны, что он основал фирму Wolfram Research, Inc., приступившую к созданию проекта престижной математической системы Mathematica. Версия Mathematica 1.0 этой системы, появившаяся в 1988 г., уже устарела, и самой известной разработкой фирмы стала версия 2.0 системы Mathematica 2, появившаяся в 1991 г. У нас она впервые стала известна благодаря обзорам.

Цели нового проекта были достаточно амбициозными — разработка мощного и универсального ядра системы (Kernel), способного работать на различных компьютерных платформах, создание многофункционального языка программирования, ориентированного на математические приложения, подготовка современного пользовательского интерфейса и обширного набора прикладных пакетов и расширений системы (Packages), мощного языка программирования математических преобразований и вычислений. Система приобрела свойства адаптации и обучения новым математическим законам и закономерностям. В разработках систем Mathematica, наряду с головной фирмой Wolfram Research, Inc., принимали участие ряд других фирм и сотни специалистов высокой квалификации (в том числе математики и программисты). Среди них есть и представители пользующейся уважением и «спросом» за рубежом математической школы России. Системы Mathematica являются одними из самых крупных программных систем, они реализуют самые эффективные алгоритмы вычислений и имеют множество новинок. К их числу относится механизм контекстов, исключающий появление в программах побочных эффектов.

Общая структура систем Mathematica (всех версий) представлена на рис. 5.1.

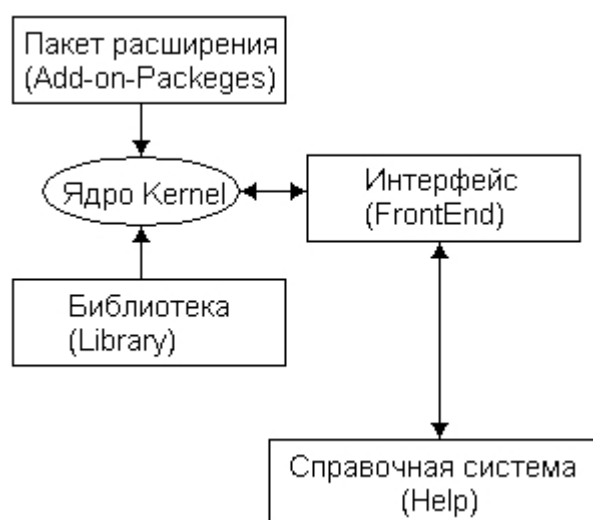


Рис. 5.1. Структура системы Mathematica

Центральное место в системах класса Mathematica занимает машинно-



независимое ядро математических операций — Kernel. Для ориентации системы на конкретную машинную платформу служит программный интерфейс процессор Front End. Именно он определяет, какой вид имеет пользовательский интерфейс системы. Интерфейсные процессоры систем Mathematica для всех платформ имеют свои нюансы, но особых различий с описанным интерфейсным процессором у них нет.

Объем ядра разных реализаций системы Mathematica из книги Стивена Вольфрама приведен в таблице:

Система	Mathematica 1	Mathematica 2	Mathematica 3	Mathematica 4
Число строк на языке C	150 000	350 000	600 000	800 000

Увеличение объема ядра в системе Mathematica 4 позволило перенести в ядро ряд функций из пакетов расширения. Ядро системы тщательно оптимизировано, что повысило скорость выполнения большинства команд.

Ядро сделано достаточно компактным с тем, чтобы любая функция из него вызывалась достаточно быстро. Для расширения набора функций служит библиотека (Library) и набор пакетов расширения (Add-on Packages). Пакеты расширений готовятся на собственном языке программирования систем Mathematica и являются главным средством расширения возможностей системы и их адаптации к решению конкретных классов задач пользователя. Кроме того, системы имеют встроенную электронную справочную систему — Help. Она содержит шесть электронных книг с «живыми» примерами, включая упомянутые во введении книги.

Ядро систем Mathematica 3/4 сделано независимым от компьютерной платформы, на которой может устанавливаться и работать система. Это гарантирует идентичность операторов и функций, используемых в системах Mathematica, устанавливаемых на разных ЭВМ — от ПК до суперкомпьютеров.

## 5.1. Общие сведения о системе Mathematica 4

Mathematica 4 рассчитан на автоматический запуск, для чего в его исходной директории имеется файл `autorun.inf`, считывание которого обеспечивает самозапуск. При этом на мгновение появляется окно, показанное на рис. 5.2.

Система Mathematica 4 чаще всего устанавливается на жесткий диск компьютера. Минимальные требования для установки системы Mathematica 4 следующие:

- IBM/Intel-совместимый компьютер с процессором класса не ниже 80386;
- операционная система Windows 95, 98 или NT;
- объем ОЗУ не менее 16 Мбайт, желательно 24 Мбайт при работе с Windows 95/ 98 и 32 Мбайт при работе с Windows NT;

- устройство чтения CD-ROM — для считывания инсталляционного диска;
- видеосистема класса SVGA;
- звуковая карта класса Sound Blaster для работы со звуком;
- все файлы системы Mathematica 4 занимают 156 Мбайт.

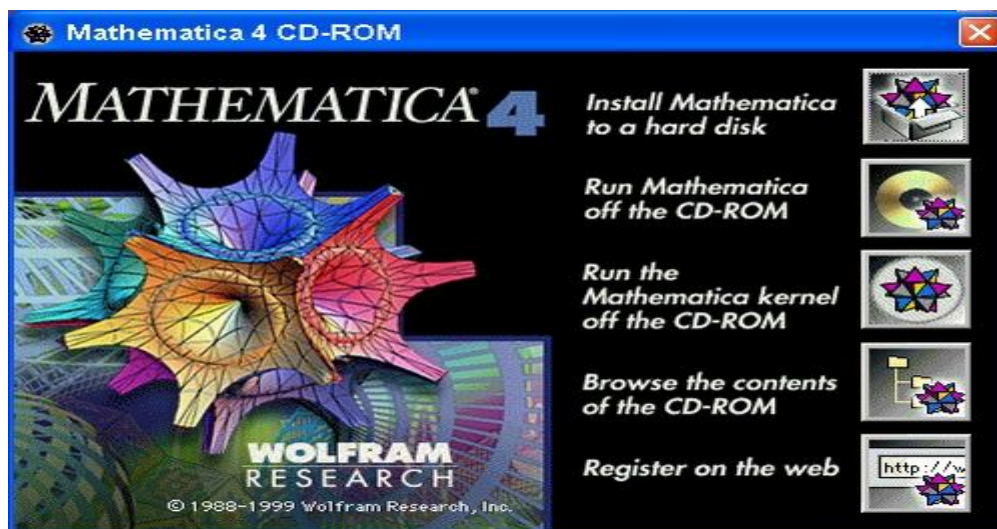


Рис. 5.2. Окно CD-ROM системы Mathematica 4

**Интерфейс системы.** Пользовательский интерфейс создается одной из двух основных частей системы — интерфейсным процессором (Front-end); вторая основная часть — ядро системы (Kernel). После загрузки интерфейсного процессора появляется скромная панель главного меню системы и пустое окно редактирования документов. В нем можно начинать вычисления, что и показано на рис. 5.3.

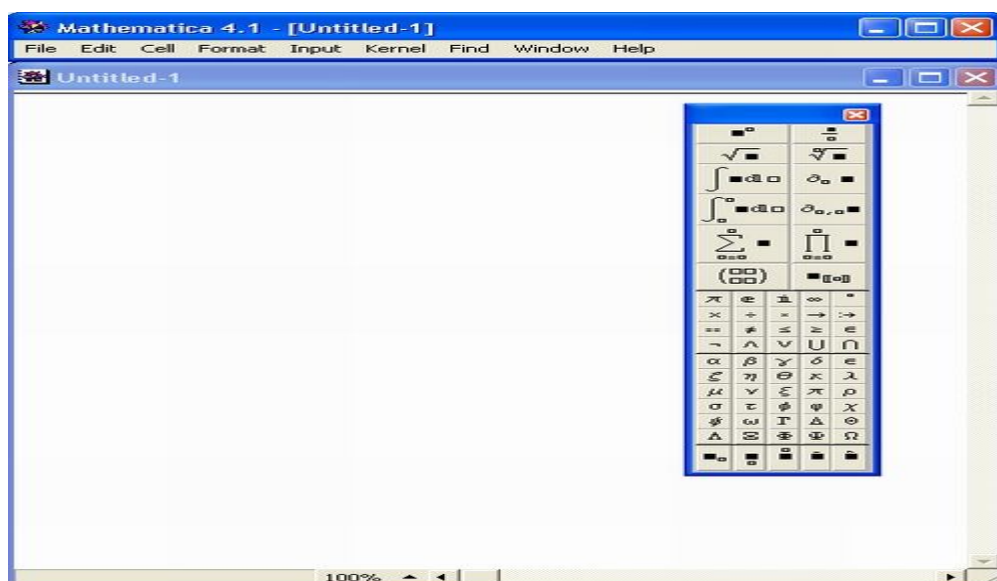


Рис. 5.3. Начало работы с системой Mathematica 4

В правой части экрана видна одна из палитр системы — палитра ввода математических выражений. Она появляется по умолчанию при обычной ин-

сталляции системы. Если этой палитры нет, то для ее вывода надо исполнить команду Basic-Input в подменю Palettes меню File. При этом запускается специальный файл, выводящий данную палитру. После этого она будет выводиться всякий раз при очередном запуске системы.

Палитру можно использовать для ввода спецзнаков и операторов, например возведения в степень, вычисления производной или интеграла и т. д. Примеры применения палитры тоже показаны на рис. 5.3. Обратите внимание, к примеру, на естественный вид интеграла в строке ввода 6. Шаблон интеграла вводится с помощью палитры BasicInput и затем заполняется. Мы не будем описывать этот процесс подробно, поскольку он вполне естественен и легко осваивается даже начинающими пользователями.

В общем случае для получения естественной формы шаблона математического выражения в строках ввода надо изменить их обычный формат InputForm (форма ввода) на формат StandardForm (стандартная форма). Для этого выберите вариант InputForm в подменю Convert To (Преобразовать в...) меню Cell.

Главное окно системы имеет крайне невзрачный вид, поскольку не содержит ничего, кроме строки заголовка и строки меню.

Справа и снизу большого окна редактирования находятся линейки прокрутки с характерными ползунками, управляемыми мышью. Они предназначены для скроллинга текстов больших документов, если последние не помещаются в видимой части окна. Положение ползунка приблизительно указывает место в документе, которое в данный момент отображается на экране.

В самом низу в начале линейки прокрутки имеется строка состояния (Status bar) с информацией о текущем режиме работы. Эта информация (если она есть в данный момент) полезна для оперативного контроля в ходе работы с системой.

Главное меню системы (см. рис. 5.4) содержит следующие позиции:

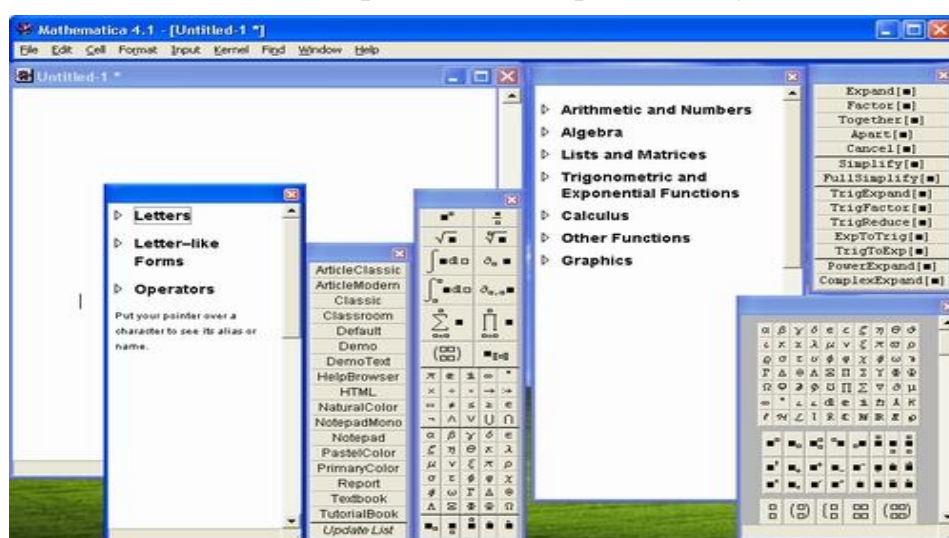


Рис. 5.4. Инструментальные палитры системы Mathematica 4

- File — работа с файлами: создание нового файла, выбор файла из каталога, закрытие файла, запись текущего файла, запись файла с изменением имени, печать документа и завершение работы;
- Edit — основные операции редактирования (отмена операции, копирование выделенных участков документа в буфер с их удалением и без удаления, перенос выделенных участков, их стирание);
- Cell — работа с ячейками (объединение и разъединение ячеек, установка статуса ячейки, открытие и закрытие);
- Format — управление форматом документов;
- Input — задание элементов ввода (графиков, матриц, гиперссылок и т. д.);
- Kernel — управление ядром системы;
- Find — поиск заданных данных;
- Window — операции с окнами и их расположением;
- Help — управление справочной системой.

Часть команд может быть в данный момент невыполнима — например, нельзя вычислить значение выражения, если его самого нет в окне редактирования или если ячейка с ним не выделена. Названия таких команд выделяются характерным серым расплывчатым шрифтом. Четкий шрифт, напротив, характерен для тех команд, которые в данный момент могут исполняться.

У многих программ интерфейс предусматривает вывод панелей с кнопками быстрого управления — уже привычными стали панели инструментов и панели форматирования. С одной стороны, эти панели упрощают работу, особенно для начинающих пользователей, но, с другой стороны, они загромождают экран, поэтому в этой системе отказались от вывода инструментальной панели с подобными кнопками.

Палитры, предназначенные для ввода математических спецзнаков, намного упрощают работу по подготовке документов. Общее число специальных математических знаков (греческих и латинских букв, операторов, функций и команд), вводимых с помощью палитр, составляет около 700. Многие знаки имеют альтернативные варианты ввода с применением комбинаций клавиш — их можно найти в справочной базе данных системы.

Рис. 5.4 наглядно показывает, что целесообразно пользоваться не более чем 2-3 панелями одновременно. Для удаления ненужных панелей в правом верхнем углу каждой из них расположены маленькие кнопки со знаком  $\times$ . Все панели максимально компактны и могут перетаскиваться мышью в наиболее удобное место экрана.

В версиях Mathematica 3/4 основным типом документов стали блокноты (notebooks). Им соответствуют файлы текстового формата с расширением .nb. Эти файлы могут редактироваться любым текстовым редактором, поддерживающим формат ASCII. Файлы содержат подробное описание документа с указаниями типов шрифтов, деталей оформления и местоположения различных объектов. Они завершаются довольно пространственным описанием того, что собой представляет notebook. Кроме того, система имеет ряд стан-

дартных пакетов расширения (в оригинале — Add-Ons), расположенных в каталоге ADDONS:

- Algebra — работа с полиномами, алгебраическими неравенствами, Гамильтоно-вой алгеброй и т. д.

- Calculus — символьные вычисления производных, интегралов и пределов функций, прямое и обратное преобразования Фурье и Лапласа, решение систем нелинейных уравнений, реализация инвариантных методов, решение дифференциальных уравнений в частных производных, нахождение полных интегралов и дифференциальных инвариантов нелинейных уравнений, аппроксимация Паде, вычисление эллиптических интегралов и работах векторами.

- DiscreteMath — вычисления из области дискретной математики, комбинаторики, вычислительной геометрии и теории графов, решение рекуррентных и разностных уравнений, операции с целыми числами и т. д.

- Geometry — функции для выполнения геометрических расчетов, создания правильных прямоугольников и многогранников, вращения геометрических фигур в плоскости и в пространстве.

- Graphics — построение графиков специального вида, геометрических фигур и поверхностей, графиков параметрически и неявно заданных функций, представления функций комплексного переменного, отображение ортогональных проекций трехмерных фигур, имитация теней, средства оформления графиков.

- LinearAlgebra — решение задач линейной алгебры, дополнительные векторные и матричные операции, задание ортогональных векторных базисов и т. д.

- Miscellaneuos — задание единиц измерения физических величин, данные о химических элементах, физические константы, географические данные и все прочее, не вошедшее в другие категории.

- NumberTheory — функции теории чисел.

- NumericalMath — реализация важнейших численных методов, аппроксимация данных и аналитических функций полиномами, сплайнами и тригонометрическими рядами, численное интегрирование и дифференцирование, решение дифференциальных уравнений, вычисление корней нелинейных уравнений, нахождение вычетов и разложений в комплексной плоскости и т. д. .

- Statistics — статистические функции для непрерывных и дискретных распределений, реализация линейной и нелинейной регрессии, вычисление параметров ряда распределений (особенно нормального), функции сглаживания и подгонки данных и т. д.

- Utilities — дополнительные утилиты для работы с бинарными файлами и памятью компьютера, поддержки языков, работы с системами класса AutoCAD и т.д.

Пакеты расширения содержат множество (полторы сотни) библиотечных файлов с расширениями .m, в каждом из которых определен ряд новых функций системы. Число функций в одном пакете расширений лежит в пре-

делах от нескольких функций до нескольких десятков, а общее число дополнительных функций и их вариантов достигает тысячи. С их помощью можно реализовывать новые алгоритмы решения математических задач и постоянно расширять возможности системы. Все библиотечные файлы подробно прокомментированы, что облегчает их использование пользователями, владеющими английским языком.

**Ядро системы Kernel.** В Mathematica 4 предусмотрена прямая работа с ядром системы в режиме командной строки. Такая работа возможна как после активизации ярлыка Mathematica 4 Kernel в папке системы. При этом работа с ядром возможна даже в MS-DOS.

При пуске ядра указанными способами появляется окно, показанное на рис. 5.3, в полностью развернутом виде. Лишь сообщения сверху окна указывают на то, что работает новейшая версия системы — Mathematica 4. В остальном работа с ядром напоминает работу в текстовом режиме системы Mathematica 1.0. Даже графики в данном случае строятся текстовыми символами, что наверняка умилило почитателей ретро в области вычислений.

При всей примитивности прямой работы с ядром она имеет ряд достоинств:

- работа происходит быстро;
- удобно знакомиться с назначением функций системы;
- простой формат ввода и вывода облегчает копирование;
- возможна работа в MS-DOS;
- требуются минимальные ресурсы ПК.

Таким образом, прямая работа с ядром может быть полезна при освоении системы Mathematica.

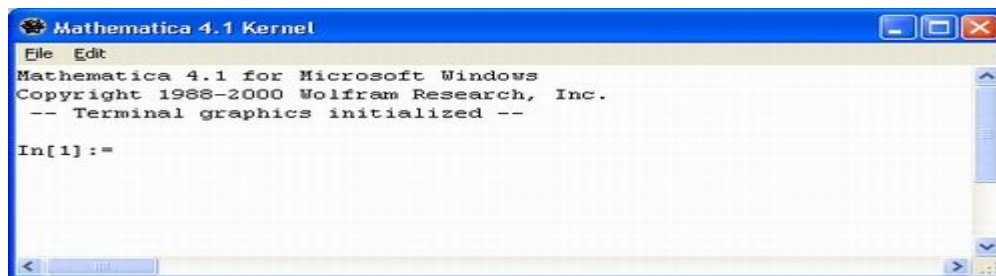


Рис. 5.5. Пример прямой работы с ядром системы Mathematica 4

Меню Kernel служит для управления действиями, проводимыми ядром системы над ячейками загруженного документа. Внешний вид меню Kernel и его подменю Evaluation показан на рис. 5.6.



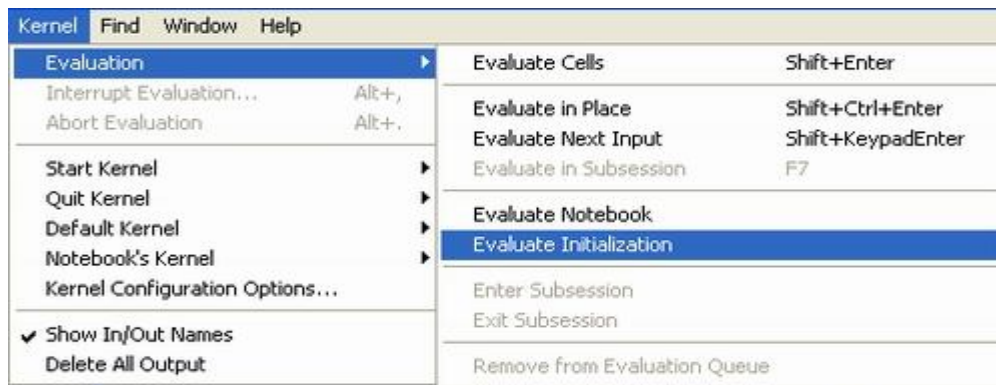


Рис. 5.6. Меню Kernel и его подменю Evaluation

Основные команды управления процессом вычислений сосредоточены в подменю Evaluation меню Kernel:

- Evaluate Cells (Shift+Enter) — вычисление выделенных ячеек;
- Evaluate in Place (Shift+Ctrl+Enter) — вычисление выделенного выражения «на месте»;
- Evaluate Next Input (Shift+Enter на цифровом поле клавиатуры) — вычисление следующей строки ввода;
- Evaluate in Subsession — вычисление выделенных ячеек в параллельном интерактивном сеансе работы с ядром;
- Evaluate Notebook — вычисление всех ячеек документа;
- Evaluate Initialization — вычисление инициализационных ячеек;
- Enter Subsection — запуск диалогового сеанса работы с ядром;
- Exit Subsection — завершение диалогового сеанса работы с ядром.

Данная группа команд управляет вычислением ячеек. Перед вычислением каждая ячейка оценивается по своим признакам. Команда Evaluate Cells оценивает все выделенные ячейки, вызывает их вычисление и помещает результат вычисления каждой ячейки сразу после нее. Это одна из наиболее распространенных команд. Следует помнить, что, казалось бы, естественное нажатие клавиши Enter вызывает лишь переход на новую строку, а не вычисление выделенных ячеек ввода. При управлении с клавиатуры вычисление выделенных ячеек происходит при одновременном нажатии клавиш Shift и Enter.

Особое внимание надо обратить на команду Evaluate in Place. Допустим, вы ввели в ячейку ввода, выражение

$$(2+3)/7$$

Выделите мышью выражение (2+3). Теперь, исполнив команду Evaluate in Place нажатием клавиш Ctrl+Shift+Enter, мы получим в строке ввода следующее:

$$5/7$$

Таким образом, выражение (2 + 3) было вычислено прямо в строке ввода, и на его месте появился результат — 5. Если теперь исполнить команду Evaluate Cells, то появится строка вывода с результатом:

$$5/7$$

Поскольку результат представлен дробно-рациональным числом, он

повторяет выражение в строке ввода, но в ином формате — в формате вывода.

Команда Evaluate Next Input позволяет последовательно вычислить ряд ячеек, расположенных под выделенной ячейкой. Если текстовый курсор находится в ячейке ввода, данная команда вычисляет эту ячейку. В противном случае она перемещает выделение на следующую ячейку. Последующее использование команды ведет к исполнению этой ячейки, затем к выделению следующей ячейки, ее вычислению и т. д. Таким образом, можно последовательно вызывать вычисление ячеек документа, используя эту команду дважды для каждой ячейки.

Команда Evaluate Notebook вычисляет все ячейки введенного документа сверху вниз. Это особенно полезно, если результаты вычисления последующих ячеек зависят от результатов вычисления предыдущих. При этом все ячейки переоцениваются, то есть выполняются заново с учетом всех возможных изменений их содержимого. Это напоминает работу с электронными таблицами, когда смена численного значения в одной ячейке автоматически меняет содержание всех других ячеек, использующих данные из данной ячейки.

Команда Evaluate Initialization вычисляет все ячейки, помеченные как инициализационные, то есть имеющие символ «т» над квадратной скобкой, обрамляющей ячейку. О задании такого признака говорилось ранее в разделе «Манипуляции с ячейками». Ячейки с указанным признаком выполняются этой командой без их выделения.

Следующие две команды меню Kernel управляют процессом текущих вычислений:

- Interrupt (Alt+,) — прерывание вычислений;
- Abort (Alt+. ) — полное прекращение вычислений.

Их действие вполне очевидно. Команда Interrupt служит для прерывания текущих вычислений. Эта команда при исполнении задает запрос о том, каким образом вы хотите прервать вычисления и сколько шагов вычислений надо еще сделать. Разумеется, можно и отменить прерывание. Команда Abort вызывает полное прекращение вычислений, так что их можно возобновить лишь с самого начала. Вместо результата выдается сообщение \$Abort.

Полезно запомнить клавиатурные комбинации для этих команд, поскольку «зависание» системы из-за чрезмерно большого времени исполнения неудачного алгоритма (например, глубокой рекурсии) не редкость. Кстати, в процессе таких вычислений команды прерывания доступны и из меню.

Используйте команды Interrupt и Abort, если вы загнали систему в режим «вечных вычислений» — заикливание. Это позволит вам благополучно выйти из этого режима, сохранив текущие данные и текущую программу (документ). Иные способы прерывания вычислений чреваты потерей текущих данных и документа.

## 5.2. Графические возможности



Система Mathematica обладает превосходными графическими возможностями — от построения двумерных и трехмерных графиков до синтеза сложных изображений (например цветных карт) и динамически изменяющихся поверхностей. Эти возможности задаются встроенными в ядро графическими функциями и дополняются средствами пакетов расширения.

Система Mathematica позволяет:

- Строить двумерные и трехмерные графики
- Получать информацию о графических объектах
- Перестройка и комбинирование графиков
- Проводить импорт графических изображений
- Вставить объекты
- Проводить синтез звуков

Обширные графические возможности достигаются при небольшом числе встроенных функций графики за счет их модификации с помощью опций и директив. Благодаря этому Mathematica позволяет строить практически любые виды графиков. Для просмотра и изменения опций графика можно (выделив ячейку с графиком) воспользоваться описанным ранее инспектором опций, в котором есть соответствующий раздел.

Рассмотрим графическую функцию Plot. Mathematica допускает следующие конструкции:

- `Plot[Sin[x], {x, 0, 20}]` — построение графика синусоиды;
- `g:=Plot [Sin [x], {x, 0, 20} ]` — задание объекта — графика синусоиды — с отложенным выводом;
- `g=Plot [Sin [x], {x, 0, 20} ]` — задание объекта — графика синусоиды — с немедленным выводом.

Начнем рассмотрение графических возможностей системы с построения простейших графиков функций одной переменной вида  $y = f(x)$  или просто  $f(x)$ . График таких функций строится на плоскости, то есть в двумерном пространстве. При этом используется прямоугольная (декартова) система координат. График представляет собой геометрическое положение точек  $(x, y)$  при изменении независимой переменной (абсциссы) в заданных пределах, например от минимального значения  $x_{\min}$  до максимального  $x_{\max}$  с шагом  $dx$ . По умолчанию строятся и линии координатной системы.

Для построения двумерных графиков функций вида  $f(x)$  используется встроенная в ядро функция Plot:

- `Plot [f, {x, xmin, xmax}]` — возвращает объект, представляющий собой график функции  $f$  аргумента  $x$  в интервале от  $x_{\min}$  до  $x_{\max}$ ;
- `Plot[{f1, f2, ...}, {x, xmin, xmax}]` — возвращает объект в виде графиков ряда функций  $f_i$ .

Функция Plot используется для построения одной или нескольких линий, дающих графическое представление для указанных функций  $f$ ,  $f_1$ ,  $f_2$  и т. д. На рис. 5.7 показано построение графика функции  $\sin(x)/x$  без использования каких-либо опций (точнее, с набором опций по умолчанию).

Тут виден как раз тот случай, когда масштаб графика по вертикали выбран системой неудачно — часть графика сверху просто отсекается. В боль-

шинстве же случаев применение функции Plot позволяет получить вполне «удобоваримый» график.

**Опции функции Plot.** Для вывода их списка надо использовать команду Options [Plot].

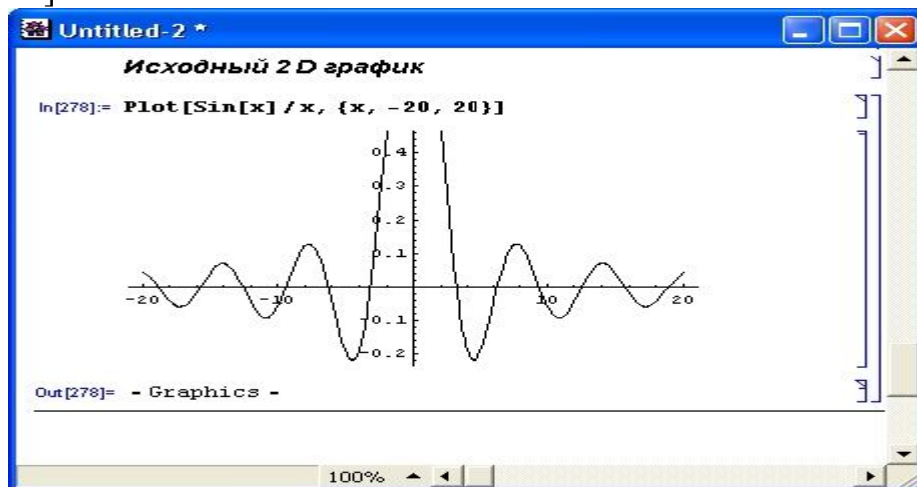


Рис. 5.7. Построение двумерного графика

Опции внутри графических функций задаются своим именем name и значением value в виде

name -> value

Наиболее распространённые символьные значения опций:

- Automatic — используется автоматический выбор;
- None — опция не используется;
- All — используется в любом случае;
- True — используется;
- False — не используется.

Многие опции могут иметь числовые значения. В сомнительных случаях рекомендуется уточнять форму записи опций и их значений по оперативной справочной системе. Рассмотрим примеры применения опций двумерной графики.

Мы уже отметили неудачный выбор масштаба в случае, представленном на рис. 5.7. Очевидно, этот недостаток графика легко исправить, введя коррекцию масштаба по оси y. Это и сделано в примере, показанном на рис. 5.8.

Для изменения масштаба использована опция PlotRange->{-0.25,1.2}. Нетрудно догадаться, что эта опция задает пределы отображения графика по вертикали от -0.25 до 1.2.

По умолчанию система строит графики, не указывая надписей ни по осям координат (кроме букв x и y), ни в верхней части графика. Такая надпись на графике по центру сверху называется титульной. Рис. 5.9 показывает построение графика с надписями у координатных осей.

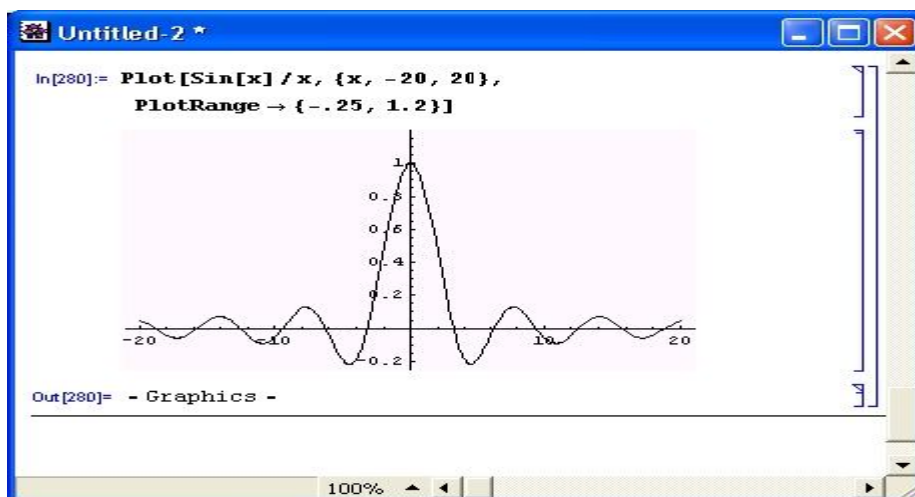


Рис. 5.8. График функции  $\sin(x)/x$  с масштабом, дающим его отображение в полном виде

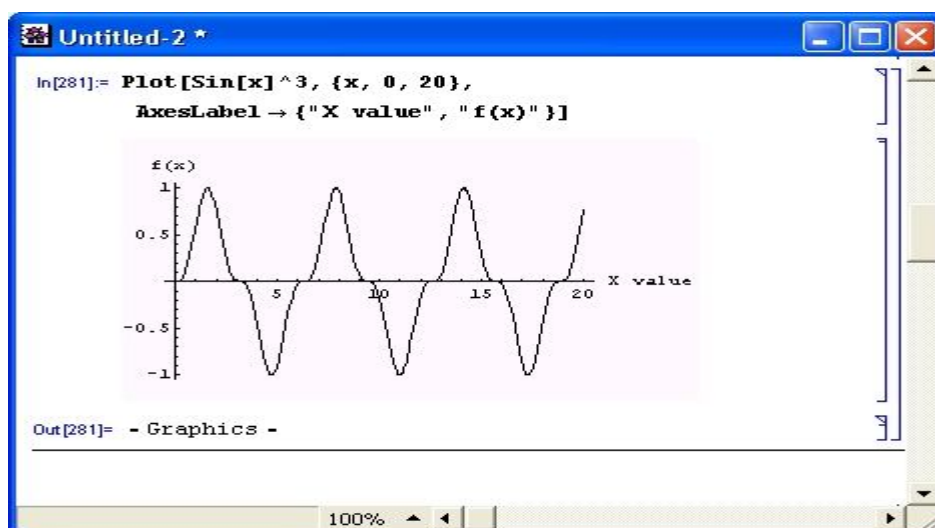


Рис. 5.9. График с надписями по координатным осям

Для создания таких надписей используется опция `Axes Label`. После нее указывается список, содержащий две надписи — одну для оси  $x$ , вторую — для оси  $y$ . Надписи указываются в кавычках. Таким образом, задание опции выглядит следующим образом: `AxesLabel-> {"X value", "f(x)}`.

С помощью опции `Axes` со значением `None` можно убрать с графика отображение осей. Вид получающегося при этом графика показан на рис. 5.10. При его построении, кроме удаления осей, использована опция `PlotLabel` для вывода указанной в качестве ее значения титульной надписи.

Рисунок 5.11 иллюстрирует недостаток одновременного представления нескольких графиков, создаваемого по умолчанию, — все графики построены одинаковыми линиями, и не сразу ясно, какой график к какой функции относится. Рис. 5.12 показывает возможности управления стилем линий (густотой черного цвета) графиков с помощью опции `PlotStyle`. Если желательно выделение линий разными цветами, удобно использовать в качестве значения опции `PlotStyle` список вида `{Hue [c1] , Hue [c2] , ...}`, где параметры  $c_1, c_2, \dots$  выбираются от 0 до 1 и задают цвет соответствующей кривой.

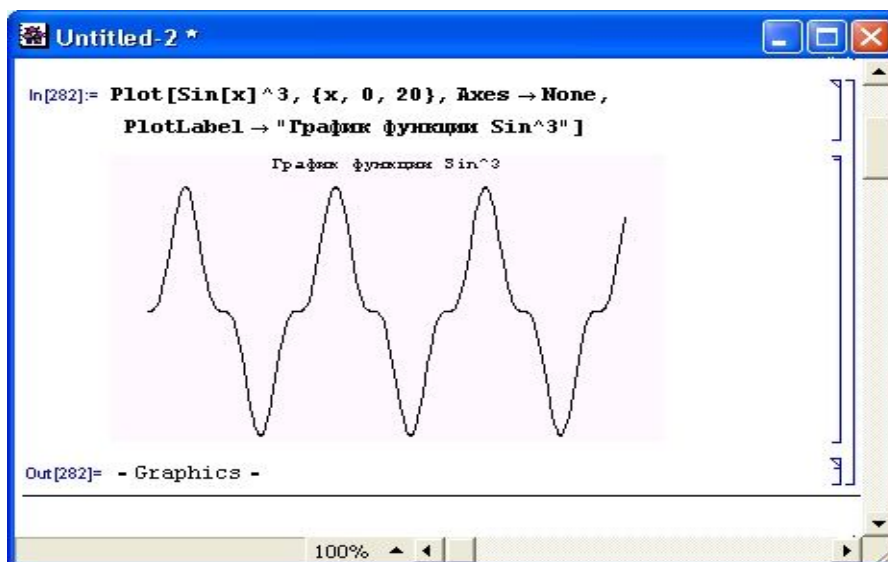


Рис. 5.10. График без координатных осей, но с титульной надписью

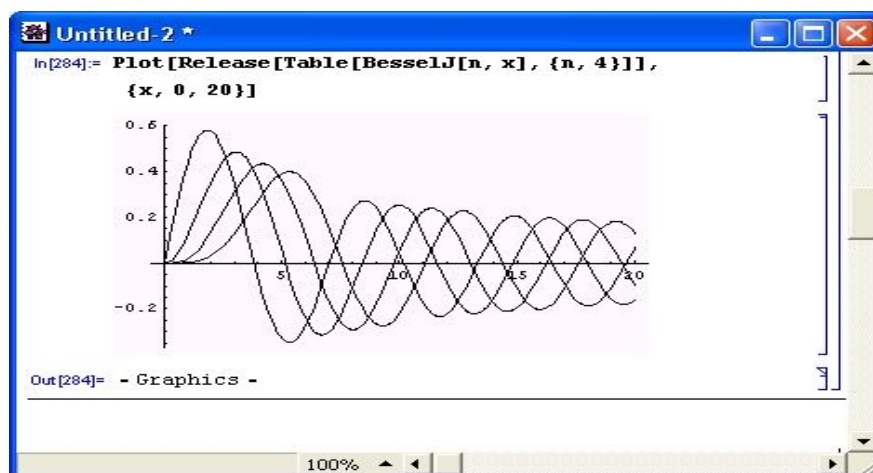


Рис. 5.11. Семейство функций Бесселя на одном графике

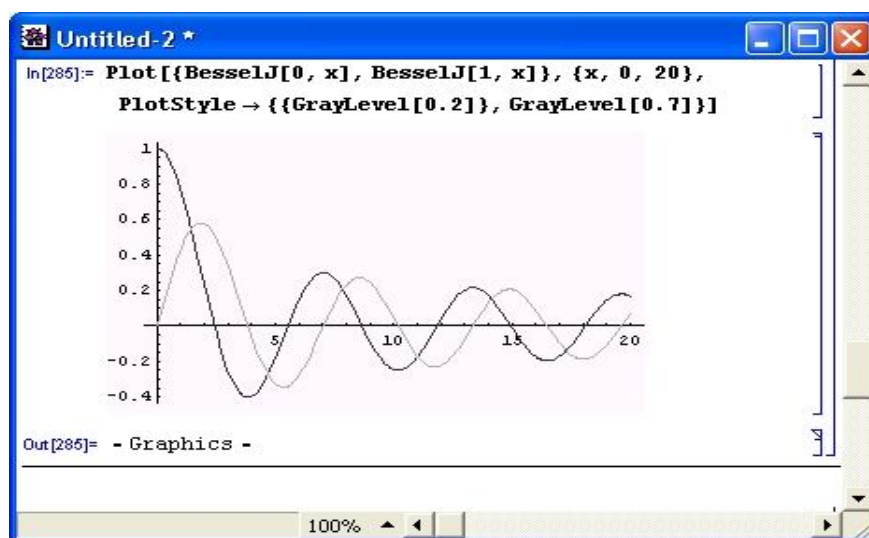


Рис. 5.12. Построение графиков линиями разного стиля

Применение других опций позволяет задавать массу свойств графиков, например цвет линий и фона, вывод различных надписей и т. д. Помимо

представленных примеров, полезно просмотреть и множество примеров построения двумерных графиков, приведенных в справочной системе Mathematica.

**Графическая визуализация решения уравнений.** В качестве примера на рис. 5.13 показана визуализация вычисления корней квадратного уравнения. В данном случае график функции явно указывает на существование двух действительных корней при  $x$ , близких к 0.2 и 2.3. Функция Nsolve без труда находит оба корня.

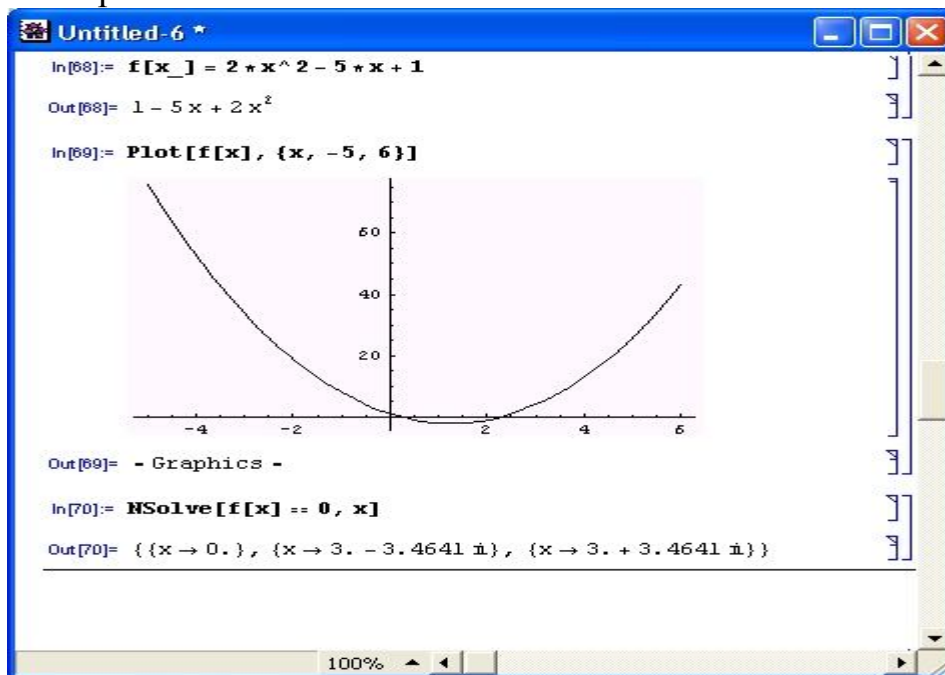


Рис. 5.13. Визуализация решения квадратного уравнения для случая двух действительных корней

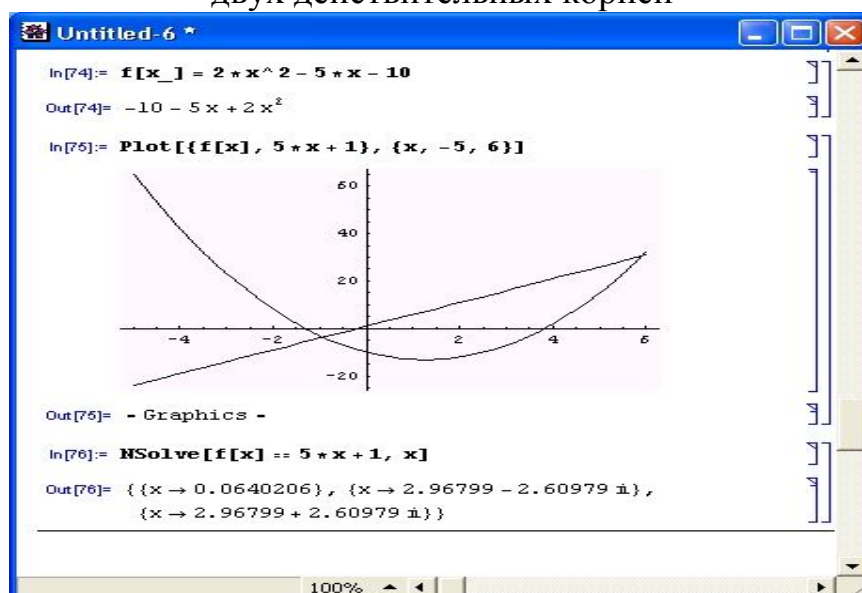


Рис. 5.14. Пример визуализации решения уравнения вида  $f(x) = 5x + 1$

Но вот на рис. 5.15 показан случай решения уравнения  $f(x) = \exp(x/2)$ . Графики функций ясно показывают, что парабола пересекается экспонентой в двух точках. Однако функция Nsolve отказывается решать такое уравнение

и выдает сообщение о том, что оно является трансцендентным.

Таким образом, в данном случае наличие графического решения говорит о необходимости смены функции, с помощью которой до сих пор решались уравнения. Подходящей в данном случае является функция FindRoot, которая отыскивает одно решение вблизи заданной начальной точки. Применив ее дважды, нетрудно получить оба корня данного уравнения.

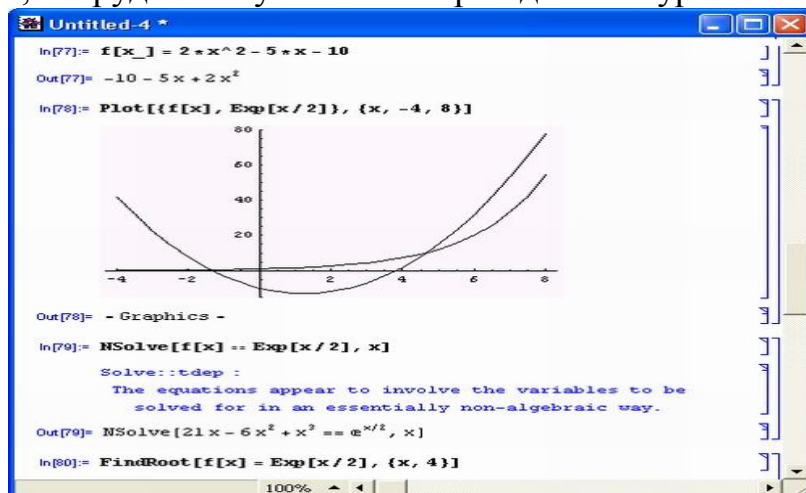


Рис. 5.15. Пример решения уравнения вида  $f(x) = \exp(x/2)$

Приведенные примеры далеко не исчерпывают проблему графической визуализации решения и выбора методов решения.

**Форматы изображений.** Mathematica 4 поддерживает свыше 20 различных форматов файлов. Возможность импорта графического файла с высоким разрешением в формате TIFF и последующего преобразования файла в формат JPG иллюстрирует рис. 5.16.

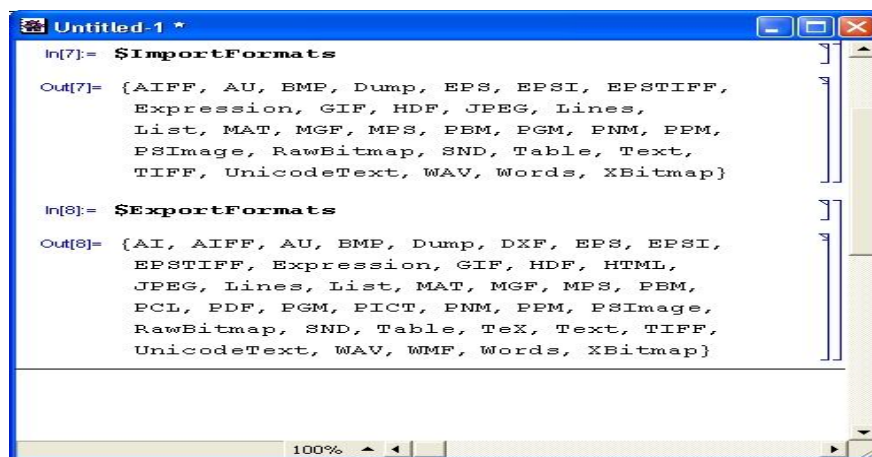


Рис. 5.16. Форматы ввода и вывода изображений, поддерживаемые системой Mathematica 4



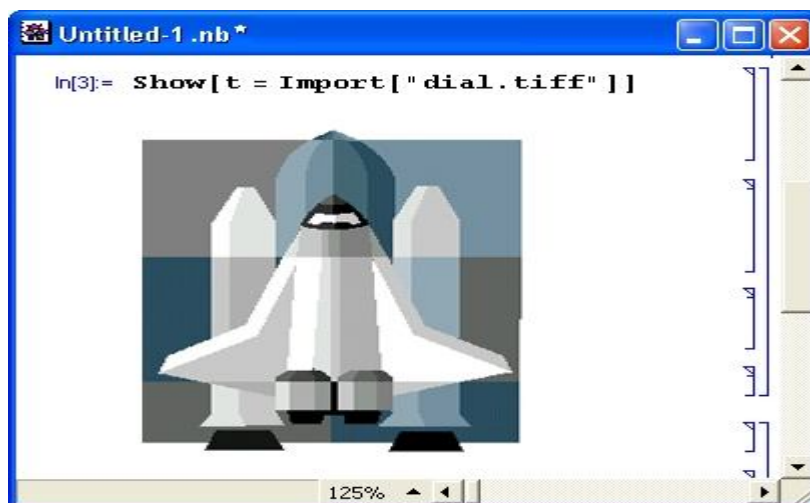


Рис. 5.17. Пример импорта файла в одном формате и экспорта того же файла в другом формате

### 5.3. Справочная система пакета Mathematica

Справочная база данных управляется командами, расположенными в меню Help (Справка). Его можно вызвать с помощью команды Help Browser (Shift+F1). Почти аналогично действует команда Find Selection Function (F1), которая служит для поиска заданной функции.

Справочная база данных стала очень удобной и содержит все лучшее из интерфейса справочных систем других математических пакетов. В частности, заимствована система иерархического поиска нужных данных (рис. 5.18), применяемая в системах класса Maple V.

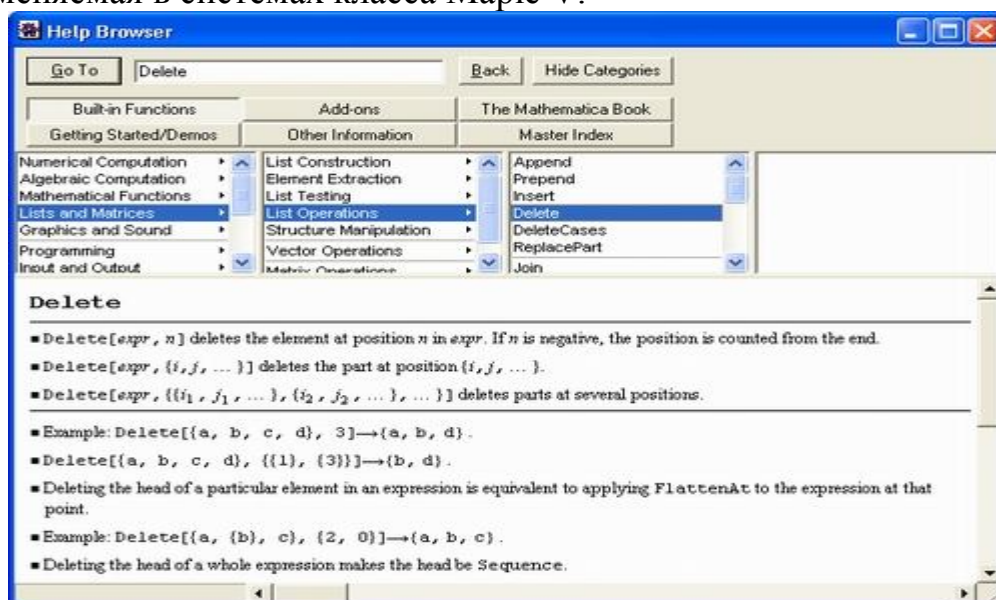


Рис. 5.18. Окно браузера справочной системы

В окне браузера справочной системы можно (активизацией соответствующей кнопки) выбрать следующие разделы справочной системы:

- Built-in Functions — встроенные функции;
- Add-ons — пакеты расширений;

- The Mathematica Book — электронная версия книги «The Mathematica Book», написанной разработчиком системы Стивеном Вольфрамом;
- Getting Started/Demos — руководство для начинающих и примеры;
- Other Information — другая информация;
- Master Index — справка по индексу (алфавитный указатель).

На рис. 5.18 наглядно показан поиск данных о встроенной функции, вычисляющей синус угла  $x$ .

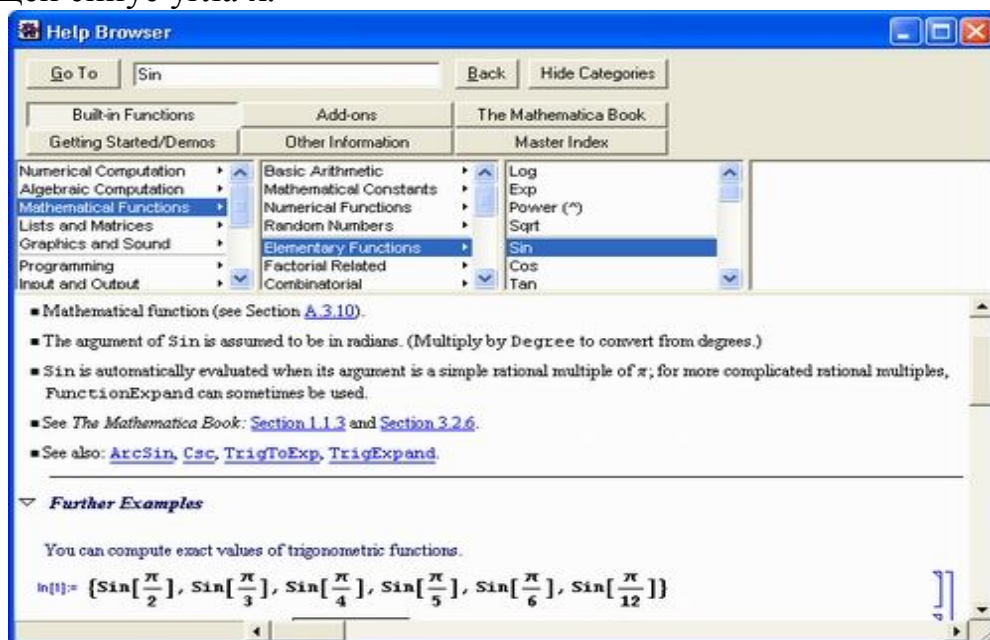


Рис. 5.19. Пример замены примера в ячейке 1п[1] и его вычисления

При этом возможен как прямой поиск по имени функции (в нашем случае Sin), так и поиск по контексту. В окнах справочной системы можно просматривать также страницы электронных книг, поставляемых с системой. Справочная система имеет гипертекстовые ссылки (подчеркнутые слова синего цвета), что позволяет расширять знакомство с объектами системы.

**Работа с примерами.** Практически по каждой функции приведен ряд примеров, которые открываются при активизации гиперссылки в виде треугольника с надписью Further Examples (вначале примеры скрыты). Примеры являются «живыми» в том смысле, что, не выходя из справочной системы, можно перенабрать содержимое любой ячейки ввода и тут же, вычислив ячейку, получить новый результат. К примеру, на рис. 5.19 показано, как список синусов, заданный ранее (см. рис. 5.18) строкой ввода In, заменен на построение графика функции Sin  $[x]$  (а строка ввода получила номер In).

Можно также, выделив ячейки примеров, перенести их содержимое в буфер командой Copy и затем разместить в текущем документе командой Paste. Такой пример можно редактировать и использовать для решения своих, близких к нему по сути задач.

**Работа с электронным учебником.** Для демонстрации возможностей системы служит электронный учебник Getting Started/Demos. На рис. 5.20 показан пример работы с ним — иллюстрируется построение документов в формате Notebook.



Электронный учебник содержит множество полезных применений системы Mathematica 4. Однако в целом он рассчитан на начальный уровень знакомства с системой. Учебник представляет материал по контексту.

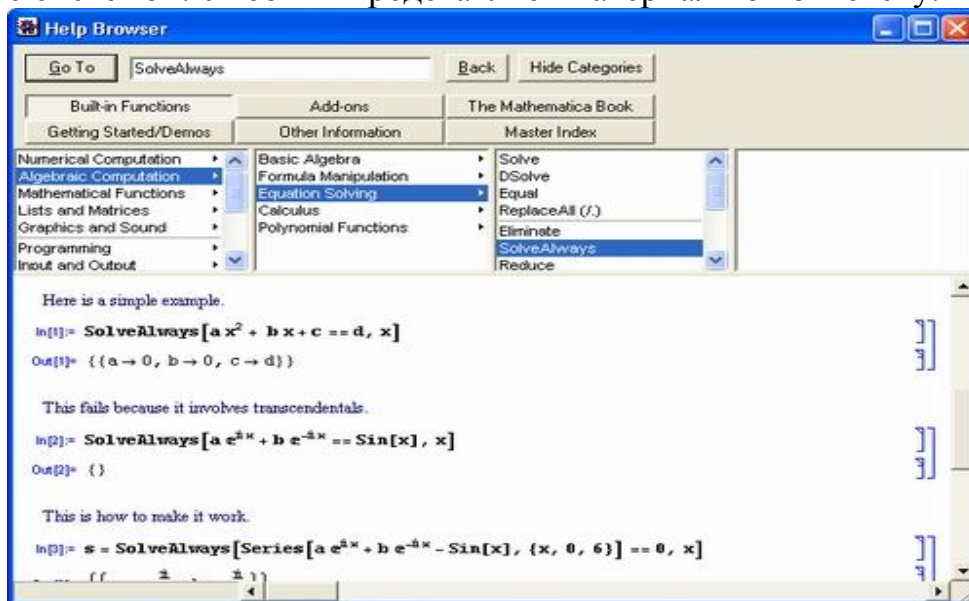


Рис. 5.20. Пример работы с электронным учебником

**Справка по пакетам расширения.** В систему Mathematica встроен ряд дополнительных пакетов расширения (Add-ons), содержащих массу полезных новых функций. Они служат для расширения функциональных возможностей системы в таких областях, как алгебра, геометрия, приближенные вычисления, дискретная математика, теория чисел, математическая статистика, линейная алгебра и т. д.

Доступ к ним возможен объявлением соответствующего пакета. На рис. 5.21 показан раздел справки по применению одного из таких пакетов для прямого и обратного преобразований Фурье.

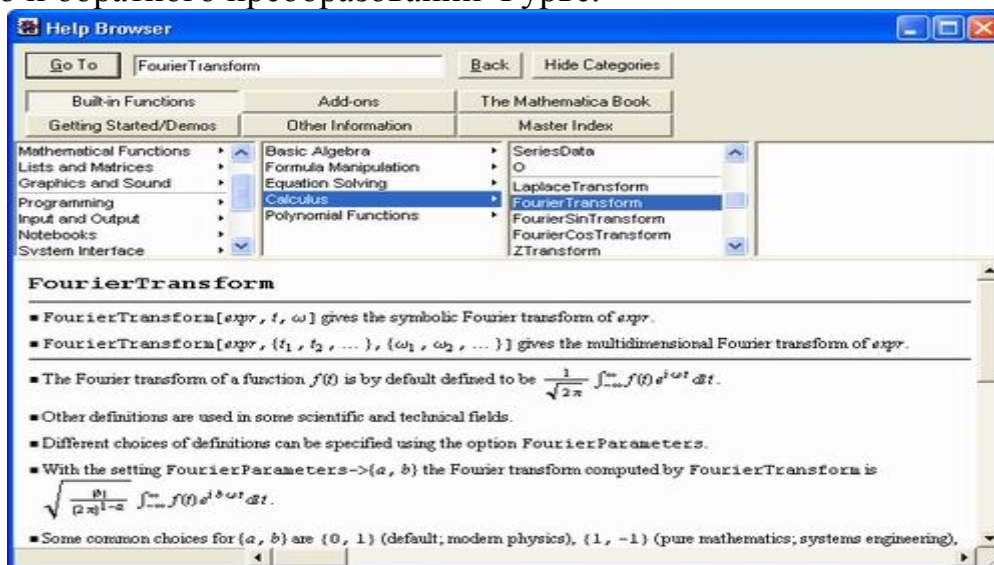


Рис. 5.21. Пример справки по преобразованию Фурье

Доступ к справке по пакетам расширения обеспечивается разделом Add-ons справочной системы. Фактически, справка повторяет материал книги от соответствующей версии системы [31,32].

Любой пример применения функции из пакета расширения также можно перенести в документ. Для этого нужно выделить соответствующую ячейку примера и с помощью команды Copy перенести ее в буфер промежуточного хранения. Затем с помощью команды Paste можно перенести содержимое буфера в документ. Для этого можно использовать и контекстно-зависимое меню, которое появляется при нажатии правой кнопки мыши.

**Работа с электронной книгой.** Раздел справки The Mathematica Book — это электронный вариант книги Стивена Вольфрама по соответствующей версии системы. Рисунок 5.22 иллюстрирует работу с электронной книгой — открыт раздел со списком ряда определенных интегралов (видно лишь начало списка).

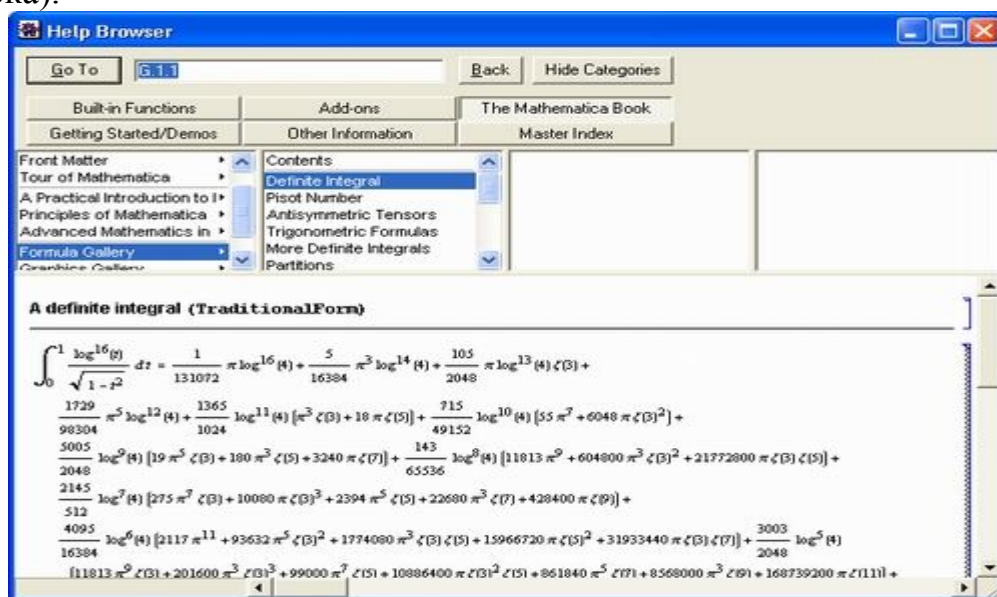


Рис.5.22. Раздел электронной книги со списком определенных интегралов

С первого взгляда трудно уловить отличие электронной книги «The Mathematica Book» («Математическая книга») от справки по пакетам расширения системы. Однако эти различия есть и заключаются в следующем:

- книга содержит большой объем чисто справочной информации (формулы, графики, примеры вычислений и т. д.);
- книга является систематическим руководством по применению системы;
- книга использует как встроенные функции, так и дополнительные функции из пакетов расширения.

Галерея формул (Formula Gallery), показанная на рис. 5.22, представляет собой весьма полное собрание математических формул. Любую формулу можно выделить и поместить в буфер обмена для использования в документах. Таким образом, электронная книга в состоянии заменить обычный математический справочник. Приходится еще раз сожалеть, что как и вся справочная база данных, электронная книга написана на английском языке.

Интересна также галерея рисунков (Graphics Gallery) электронной книги, дающая представление о графических возможностях системы. В элек-

тронной книге можно найти множество интересных и весьма эффектных примеров применения графики — например, построения карт или анимационных рисунков.

Электронная книга системы Mathematica являет собой наглядный пример развития электронных книг. Они характеризуются рядом новых качеств:

- красотой и наглядностью оформления;
- простотой поиска нужных сведений;
- возможностью применения действующих примеров;
- использованием гипертекстовых ссылок;
- применением наглядных средств анимации изображений;
- объединением с другими программными средствами;
- легкостью модификации.

При всех этих очевидных достоинствах совершенно ясно, что в ближайшие годы подобные электронные книги не заменят обычные, поскольку их стоимость неизмеримо выше, чем стоимость обычных книг, а удобства работы с электронными книгами нивелируются необходимостью многочасового сидения за экраном дисплея компьютера. К тому же обычные книги написаны на исконно русском и понятном нам языке, тогда как подавляющее большинство электронных книг остаются англоязычными.

**Просмотр другой информации.** Раздел Other Information (Другая информация) посвящен различным сведениям об интерфейсе системы Mathematica и данным по связи с ее разработчиком — фирмой Wolfram Research. На рис. 5.23 показана одна из страниц этого раздела, посвященная правилам ввода показателей степени.

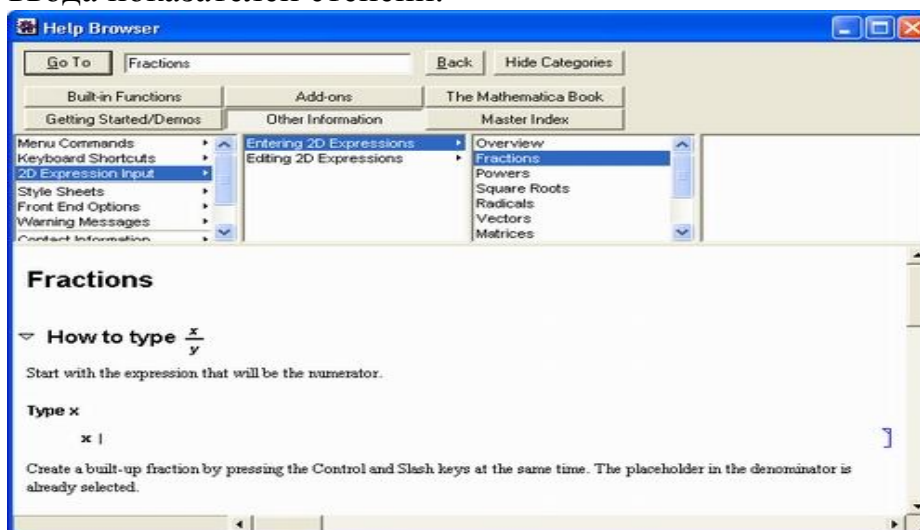


Рис. 5.23. Одна из страниц раздела Other Information справочной системы

В этом разделе справочной системы можно найти сведения обо всех командах главного меню системы, о правилах набора сложных выражений и об использовании в ячейках ввода записей математических выражений, предельно приближенных к естественной математической записи.

**Работа с алфавитным указателем.** Последний раздел справочной системы — алфавитный (или индексный) указатель Master Index. Если вы

знаете, какую команду или функцию ищете, то достаточно указать ее в поле у кнопки Go To (Перейти к...) и нажать эту кнопку. Можно также воспользоваться системой поиска по начальным буквам искомого слова (рис. 5.24).

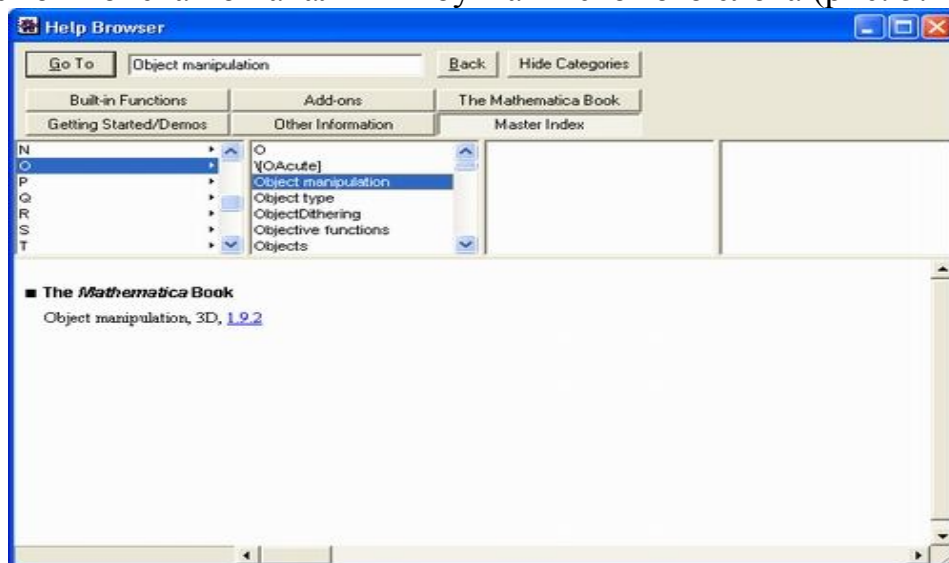


Рис. 5.24. Пример работы с алфавитным указателем

Алфавитный указатель ценен тем, что в него входят все команды и функции, опции и примитивы, причем не только встроенные, но и входящие в состав пакетов расширений.

## 5.4. Решение дифференциальных уравнений в символьном виде

Система Mathematica позволяет проводить различные операции алгебры, геометрии и математического анализа. Рассмотрим раздел «Решение дифференциальных уравнений».

Математические модели чаще всего представляются в виде дифференциальных уравнений. Дифференциальными принято называть уравнения, в состав которых входят производные функции  $y(x)$ , представляющей решение уравнения. Дифференциальные уравнения могут быть представлены в различной форме, например в общеизвестной форме Коши:

$$y'(x) = \text{eqn} = f(x, y).$$

Несколько дифференциальных уравнений образуют систему дифференциальных уравнений. Дифференциальные уравнения и системы дифференциальных уравнений могут быть линейными и нелинейными. Линейные уравнений обычно имеют решения в аналитическом виде. Нелинейные дифференциальные уравнения в общем случае аналитических решений не имеют, однако они могут быть решены приближенными численными методами.

Система компьютерной математики «Mathematica» позволяет находить как символьного, так и численного решения дифференциальных уравнений и их систем. Дифференциальные уравнения в символьном виде решаются с помощью функций:

`DSolve[eqn, y[x], x]` — дифференциальное уравнение `eqn` решается относительно функций `y[x]` по переменной `x`;

`DSolve[{eqn1, eqn2,...}, {y1 [x1,...],...}, {x1,...}]` – функция для решения системы дифференциальных уравнений.

У функции `DSolve` и ее численного варианта `NDSolve` есть пара опций, на которые следует обратить внимание:

`DSolveConstants` — опция к `DSolve`, определяющая постоянные интегрирования, которые будут использованы в результате;

`StartingStepSize` — опция к `NDSolve`, определяющая величину начального шага.

В решении дифференциальных уравнений встречаются постоянные интегрирования. По умолчанию они обозначаются как  $C[i]$ .

Приведем примеры решения дифференциальных уравнений в аналитических функциях:

```
DSolve [Derivative [1] [y] [x] == 2*a*x^3, y[x], x]
```

```
{{y[x] →  $\frac{ax^4}{2} + C[1]}$ }
```

```
DSolve [ {y1 ' [x] == 2 x, y2 ' [x] == 3 x}, {y1[x], y2[x]}, x ]
```

```
{{y1[x] →  $x^2 + C[1]$ , y2[x] →  $\frac{3x^2}{2} + C[2]}$ }
```

```
DSolve [ {y ' [x] == Sin [Ex], y[x], x]
```

```
{{y[x] →  $C[1] + x \text{Sin}[Ex]}$ }
```

```
DSolve [ {y ' [x] + y[x] == x, y[x], x]
```

```
{{y[x] →  $x - y'[x]}$ }
```

```
DSolve [ {y '' [x] - y ' [x] - 6 y[x] == 0, y[x], x]
```

```
{{y[x] →  $e^{-2x} C[1] + e^{3x} C[2]}$ }
```

```
DSolve [ {y '' [x] + 4 y ' [x] == 10 sin [2 x], y[x], x]
```

```
{{y[x] →  $-\frac{1}{4} e^{-4x} C[1] + C[2] - \text{Cos}[2x] - \frac{1}{2} \text{sin}[2x]}$ }
```

Как нетрудно заметить, аналитические решения дифференциальных уравнений могут содержать не только элементарные, но и специальные математические функции, что заметно расширяет возможности применения системы `Mathematica` в решении задач динамического моделирования.

## 5.5. Решение дифференциальных уравнений в численном виде

Дифференциальные уравнения могут быть решены в среде `Mathematica`. Численное решение дифференциального уравнения или системы ОДУ может быть найдено с помощью программирования на языке системы или с использованием встроенных функций `NDSolve`:

`NDSolve [eqns, y, {x, xmin, xmax}]`— ищет численное решение дифференциальных уравнений `eqns` относительно функции `y` независимой переменной `x` в интервале от `xmin` до `xmax`;

`NDSolve [eqns, {y1, y2,...}, {x, xmin, xmax }, MaxSteps → 3000]`— ищет численные решения относительно функций `y`.

`MaxSteps` — опция к `NDSolve`, которая определяет максимальное количество шагов.

Ниже приведены возможности применения системы Mathematica для математического моделирования. Здесь рассмотрены примеры решения систем дифференциальных уравнений с использованием технологий программирования и процедур численного решения пакета Mathematica. Результаты решения представлены в виде графиков.

**Пример 1.** Найти приближенное решение задачи Коши  $y' = \sin x - \cos x$ ,  $y(0) = 1$  на отрезке  $[0, 1]$  методом Эйлера с шагом 0.2.

Введем правую часть уравнения и шаг интегрирования

```
in[1]:=f[x,y]:=sin[x]-cos[y];
h=0.2;
```

Запишем формулы Эйлера

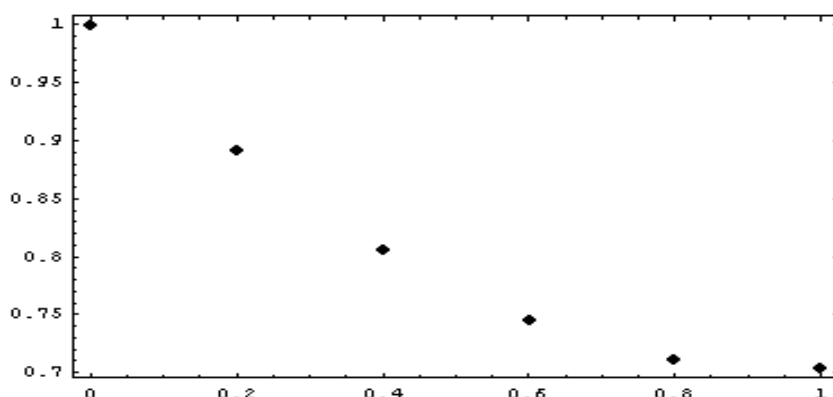
```
in[3]:=xe[0]=0;
      ye[0]=1;
      xe[i_]:=xe[i]=xe[i-1]+h;
      ye[i_]:=ye[i]=ye[i-1]+hf[xe[i-1],ye[i-1]]
```

и вычислим приближенные значения в узлах сетки

```
in[7]:= xlist=Table[xe[i], {i,0,5}]
Out[7]= {0, 0.2, 0.4, 0.6, 0.8, 1.}
in[8]:= ylist=Table[ye[i],{i,0,5}]
Out[8]={1, 0.89194, 0.806093, 0.745512, 0.711492, 0.703486}
```

Построим на графике полученное решение

```
in[9]:=gr1=ListPlot[Transpose[{xlist,ylist}],Plotstyle->PointSize[0.02],Frame->True]
```

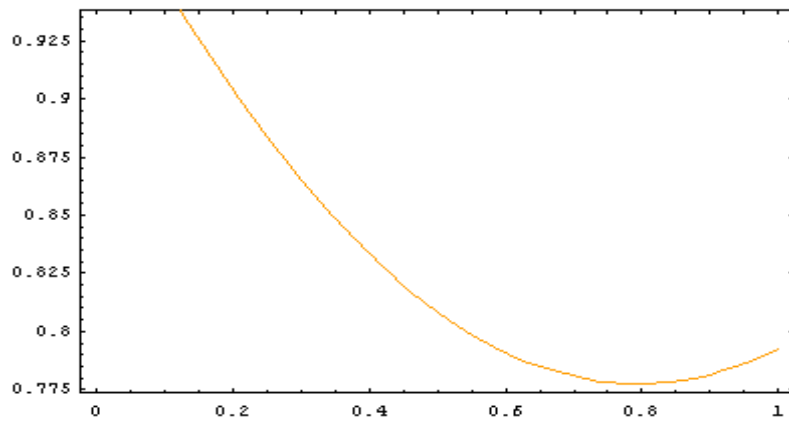


```
Out[9]=-Graphics-
```

Решим эту задачу Коши с помощью встроенной функции NDSolve и сравним результаты.

```
in[10]:=sol=NDSolve[{y'[x]==f[x,y[x]],y[0]==1},y,{x,0,1}]
Out[10]={{y->InterpolatingFunction[{{0.,1.}},<>]}}
in[11]:=gr2=Plot[Evaluate[y[x]/.sol],{x,0,1},Plotstyle->{Hue[0.1]},Frame->True]
```

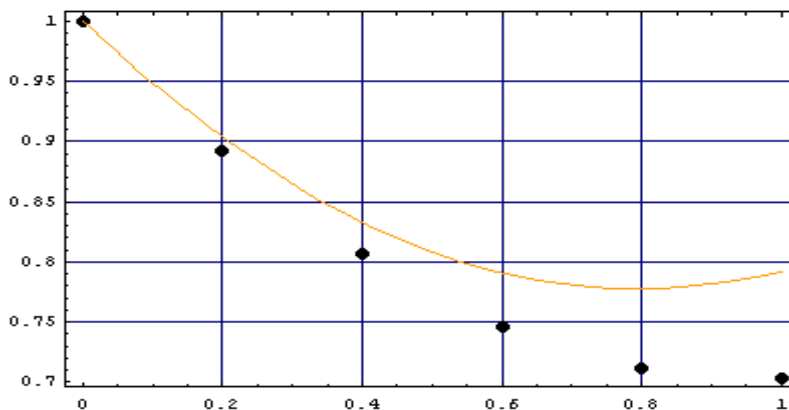




Out[11]=Graphics-

Нарисуем графики вместе

`in[12]:=Show[gr1,gr2,Gridlines→Automatic]`



Out[12]=Graphics-

При уменьшении шага интегрирования решение, найденное по формуле Эйлера, приближается к решению, найденному функцией NDSolve.

**Пример 2.** Тело, брошенное под углом к горизонту (система 4-х дифференциальных уравнений 1-го порядка)

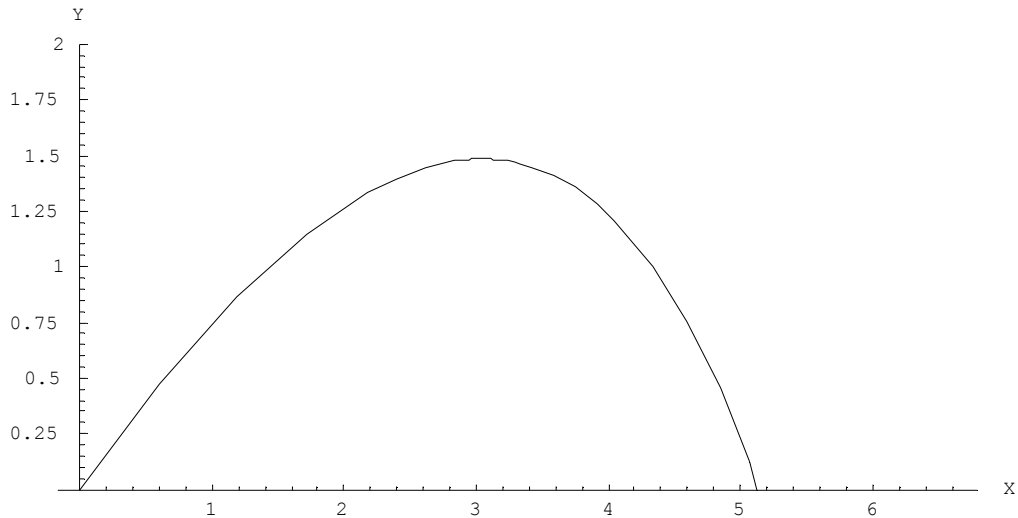
```
n = 2200; h:= 0.0005; v0 = 10; k = 1; g = 9.8; alfa = 40*π/180;
```

```
Fx[Vx_] := -k*Vx;
Fy[Vy_] := -k*Vy - g;
```

```
Sol = NDSolve[{Vx'[t] == Fx[Vx[t]], Vy'[t] == Fy[Vy[t]],
x'[t] == Vx[t], y'[t] == Vy[t], x[0] == 0, y[0] == 0,
Vx[0] == v0*Cos[alfa], Vy[0] == v0*Sin[alfa]}, {x, y, Vx, Vy},
{t, 0, 2}]
```

```
{{x → InterpolatingFunction[{{0., 2.}}, <>], y → InterpolatingFunction[{{0., 2.}}, <>]}
{{Vx → InterpolatingFunction[{{0., 2.}}, <>], Vy → InterpolatingFunction[{{0., 2.}}, <>]}}
```

```
ParametricPlot[Evaluate[{x[t], y[t]}/.sol], {t, 0, 2},
PlotRange → {0, 2}, AxesLabel → {"X", "Y"}]
```

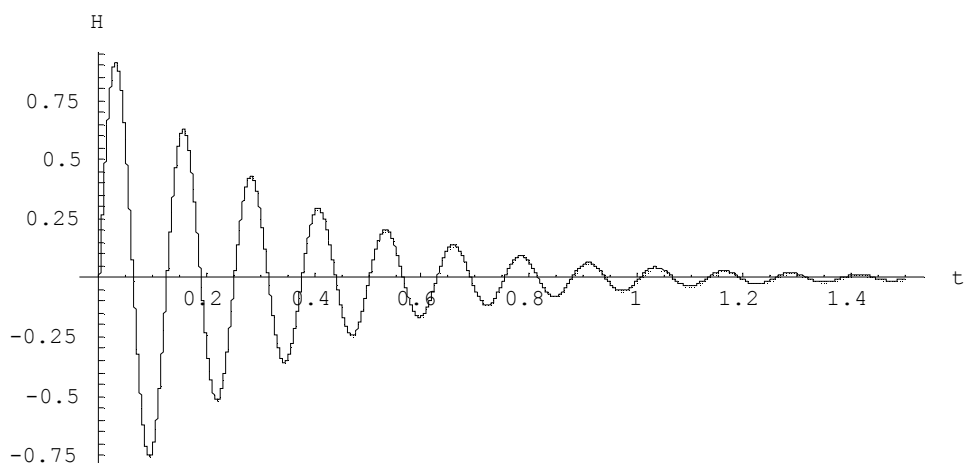


...Graphics ..

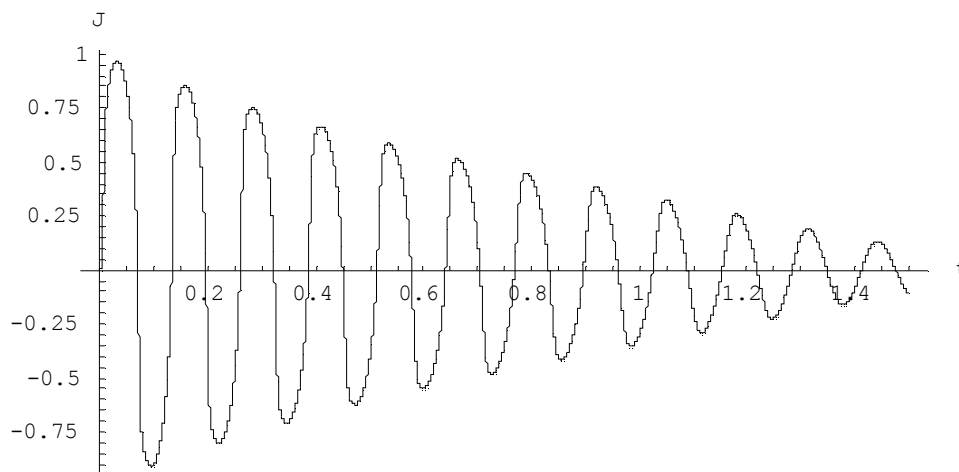
**Пример.3.** Листинг решения задачи о гистерезисе с выводом решения в виде графиков.

```
In[1]:= n=15000; h=0.0001; a=0.00006; b=3; w=50; g=0; v=500;
  H[t_]:=Exp[-b*t]*Sin[w*t]+g; F[H_,J_]:=v*(H-J^3+a*J);
  Sol =NDSolve[{J'[t]==F[H[t],J[t]]}, J[0]==0},{J},{t,0,n*h},
  MaxSteps -> 100000]
  {{J -> InterpolatingFunction [{{0., 1.5}, <>]}}
```

```
In[2]:= Graph1:=Plot[Evaluate[H[t]/. sol], {t, 0, n*h}, AxesLabel -> {"t", "H"},
  PlotRange -> All, PlotPoints -> 10000];
  Graph2:=Plot[Evaluate[J[t]/. sol], {t, 0, n*h}, AxesLabel -> {"t", "J"}, PlotRange
  -> All, PlotPoints -> 10000];
  Show[Graph1, Graph2]
```

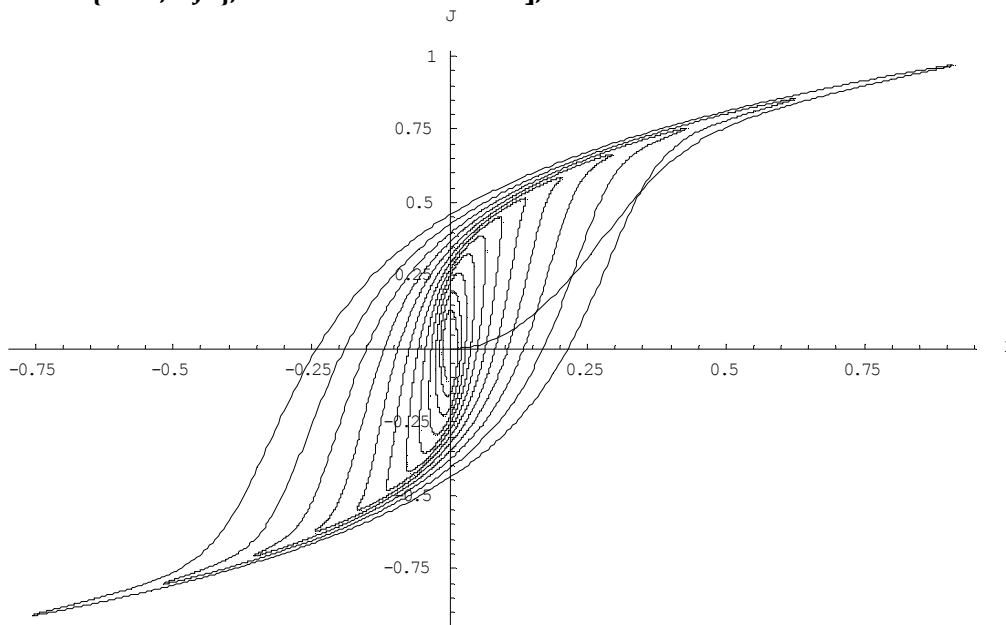






...Graphics ..

```
In[3]:= ParametricPlot[Evaluation[{H[t], J[t]} /.sol], {t, 0, n*h}, Plotrange -> All,
AxesLabel -> {"H", "J"}, PlotPoints -> 10000];
```



**Пример 4.** Листинг решения задачи о параметрическом маятнике с выводом решения в виде графиков. Математическая модель представлена в виде системы дифференциальных уравнений.

```
g=9.81; l = 5; l1 = 1/5; a = 30; w0 = Sqrt[g/l]; w1 = 2*w0;
```

```
x[t_]:= (1 + l1*Cos[w1*t]*Sin[Teta[t]]);
```

```
y[t_]:= (1 + l1*Cos[w1*t]*Cos[Teta[t]]);
```

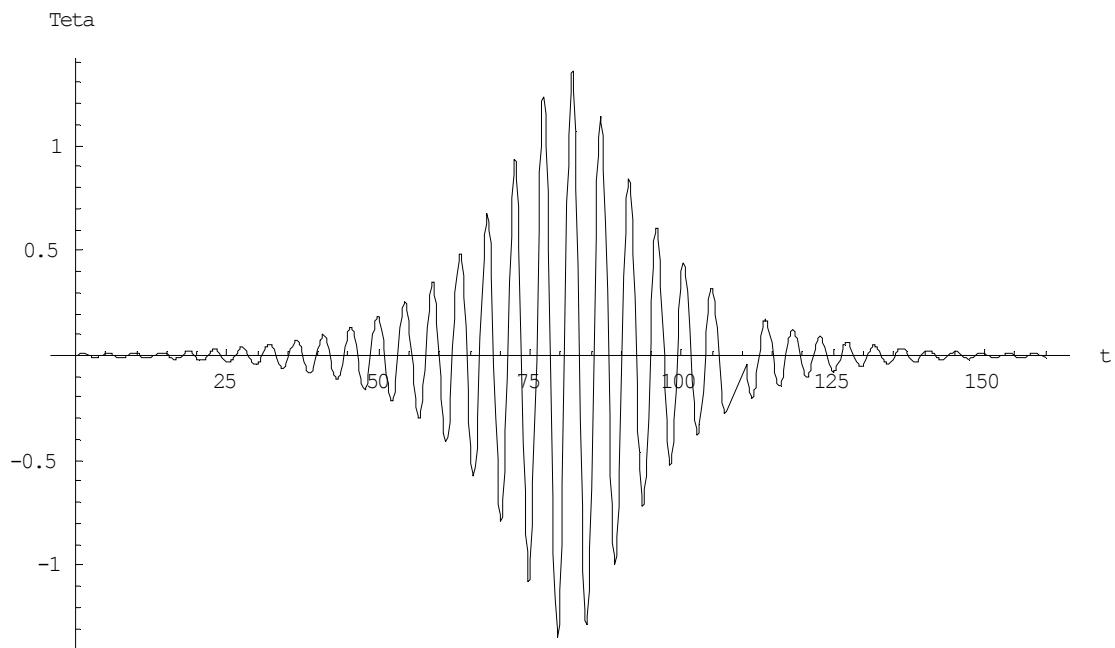
```
Sol=NDSolve[{u'[t] == (-g*Sin[Teta[t]])/(1 + l1*Cos[w1*t]), Teta'[t] == u[t],
```

```
Teta[0] == 0, u[0] == 0.01}, {Teta, u}, {t, 0, 160}, MaxSteps -> 3000]
```

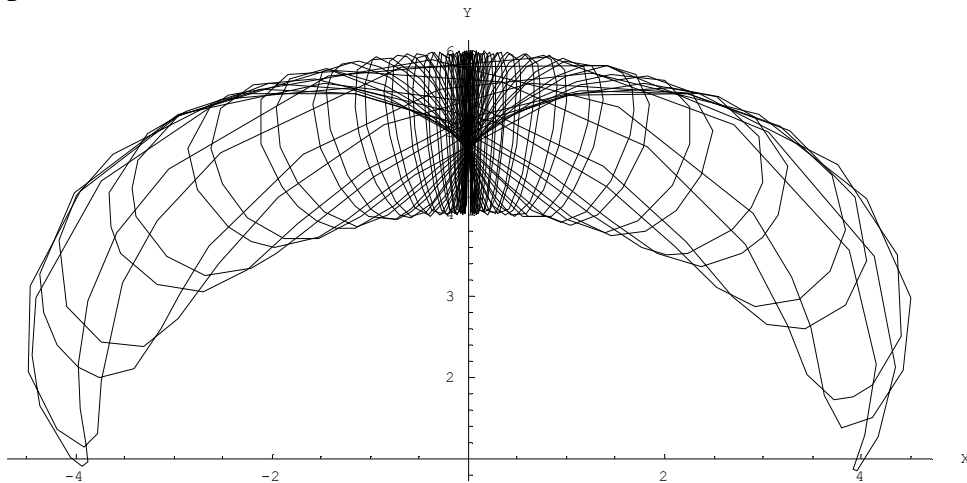
```
{{Teta -> InterpolatingFunction[{{0., 160.}}, <>], u -> InterpolatingFunction[{{0., 160.}}, <>]}}
```

```
Plot[Evaluate[Teta[t] /. sol], {t, 0, 160},
```

```
PlotRange -> All, AxesLabel -> {"t", "Teta"}];
```



```
PramericPlot[Evaluate[{x[t],y[t]}/.sol],{t,0,160},
PlotRange→All,AxesLabel→{"X","Y"}]
```



..Graphics ...

Для детального изучения системы Mathematica рекомендуем книги [1-6].

## ГЛАВА 6. МОДЕЛИРОВАНИЕ ОБЪЕКТОВ В СРЕДЕ MATLAB

Среди нескольких десятков математических пакетов (четыре из них перечислены ранее) особого внимания заслуживает система инженерных и научных расчетов MATLAB. В отличие от других пакетов, MATLAB в одинаковой мере ориентирован на применение как символических, так и численных методов. Включенное в MATLAB ядро Maple хорошо справляется с символическими вычислениями, а сам MATLAB и его многочисленные инструментарии прекрасно работают с числами.

Интересная особенность, реализованная разработчиками MATLAB, — это его способность встраиваться в другие приложения Windows и динамически обмениваться с ними данными.

Немаловажным фактором широкого распространения этого продукта является также ценовая политика компании MathWorks Inc. (производителя MATLAB), которая позволяет вузам приобретать MATLAB со значительными скидками. Это делает систему MATLAB наиболее подходящей средой для обучения студентов методам решения математических задач.

Система MATLAB предлагается разработчиками (фирма MathWorks, Inc.) как лидирующий на рынке, в первую очередь в системе военно-промышленного комплекса, в аэрокосмической отрасли и автомобилестроении, язык программирования *высокого уровня* для технических вычислений с большим числом стандартных пакетов. прикладных программ.

Система MATLAB вобрала в себя не только передовой опыт развития и компьютерной реализации численных методов, накопленный за последние три десятилетия, но и весь опыт становления математики за всю историю человечества. Около миллиона легально зарегистрированных пользователей уже применяют эту систему. Ее охотно используют в своих научных проектах ведущие университеты и научные центры мира. Популярности системы способствует ее мощное расширение Simulink, предоставляющее удобные и простые средства, в том числе визуальное объектно-ориентированное программирование, для моделирования линейных и нелинейных динамических систем, а также множество других пакетов расширения системы.

### 6.1. Общие сведения о системе MATLAB

Возможности версии MATLAB 6.0. Версия системы MATLAB 6 характеризуется целым рядом новых и важных возможностей:

- доведенное до более, чем 600 число встроенных функций и команд;
- новый интерфейс с набором инструментов для управления средой, включающий в себя окно команд (Command Window), окно истории команд (Command History), браузер рабочей области (Workspace Browser) и редактор массивов (Array Editor);
- новые инструменты, позволяющие при помощи мыши интерактивно редактировать и форматировать графики, оптимизировать их коды и затраты памяти на графические команды и атрибуты;

– улучшенные алгоритмы на основе оптимизированной библиотеки LAPACK;

новая библиотека FFTW (быстрых преобразований Фурье) Массачусетского технологического института Кембриджского университета (США);

– ускоренные методы интегральных преобразований;

– новые, более мощные и точные, алгоритмы интегрирования дифференциальных уравнений и квадратур;

– новые современные функции визуализации: вывод на экран двумерных изображений, поверхностей и объемных фигур в виде прозрачных объектов;

– новая инструментальная панель Camera для управления перспективой и ускорение вывода графики с помощью OpenGL;

– новый интерфейс для вызова Java-процедур и использования Java-объектов непосредственно из MATLAB;

– новые, современные инструменты проектирования графического пользовательского интерфейса;

– обработка (регрессия, интерполяция, аппроксимация и вычисление основных статистических параметров) графических данных прямо из окна графики;

– новое приложение MATLAB для системы разработки Visual Studio, позволяющее автоматически, непосредственно из Microsoft Visual Studio, преобразовывать Си и Си++ коды в выполняемые MATLAB файлы (MEX-файлы);

– интеграция с системами контроля версий кода, такими как Visual Source Safe;

– новый интерфейс (последовательный порт) для обмена данными с внешним оборудованием из MATLAB;

– новый пакет управления измерительными приборами (Instrument Control Toolbox) для обмена информацией с приборами, подключенными к Каналу общего пользования (GPIB, HP-IB, IEEE-488). Существует аналогичные международные МЭК (IEC) 625.1 и российские государственные стандарты. Несмотря на логическую и электрическую совместимость, международные и отечественные стандарты предполагают использование других разъемов. — Примеч. ред. или к шине VXI через адаптер VXI — GPIB (только в версиях для Windows и Sun Solaris) и последовательному интерфейсу RS-232, RS-422, RS-485 (также и для Linux-версии), в том числе в соответствии со стандартом VISA (Virtual Instruments Systems Application) (Применение виртуальных измерительных приборов);

– существенно обновленные пакеты расширения, в частности новые версии пакета моделирования динамических систем Simulink 4 и Real Time Workshop 4;

– интеграция с системами управления потребностями, например DOORS.

**Запуск системы MATLAB.** Запускается система MATLAB 6.0 из главного меню операционной системы Windows 98 со стандартным видом рабочего стола, подобным использованному в Windows 95. Для раскрытия глав-

ного меню активизируется кнопка Пуск (Start), расположенная внизу рабочего стола слева, или можно щелкнуть на значке с логотипом системы на рабочем столе Windows.

После запуска MATLAB на экране появляется основное окно системы MATLAB, показанное на рис.6.1.

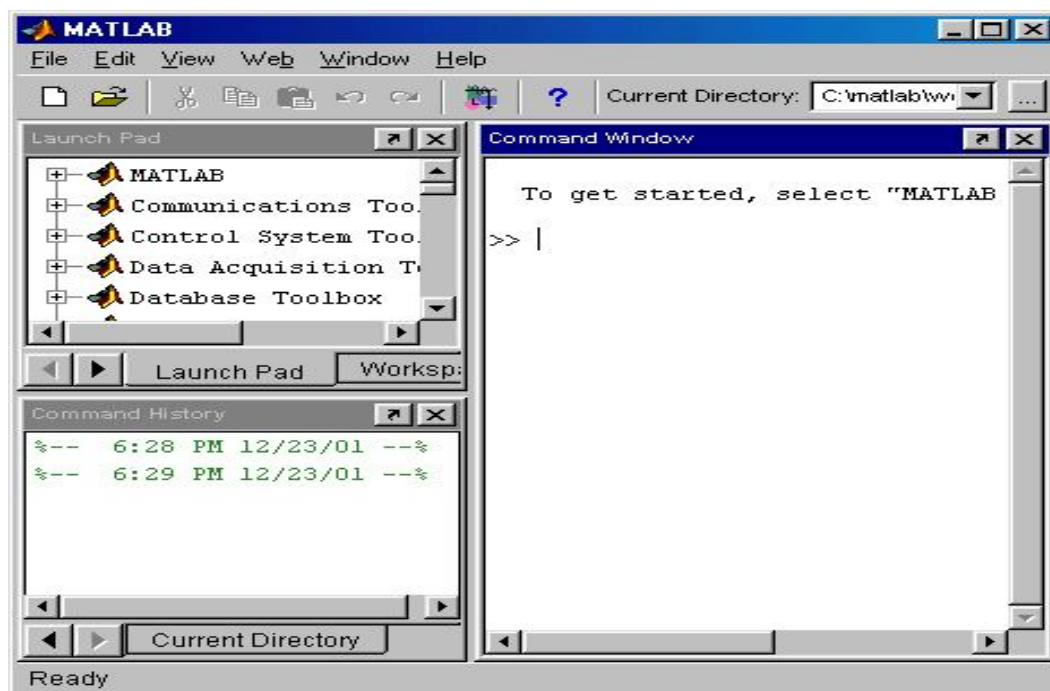


Рис. 6.1. Окно системы MATLAB после запуска и выполнения простых вычислений

Обычно это окно раскрыто не полностью и занимает часть рабочего стола. Вы можете раскрыть окно полностью. Система готова к проведению вычислений в командном режиме. При этом вы можете не обращать внимания на новации пользовательского интерфейса, привнесенного операционными системами Windows 95 и 98/Me/2000/NT4, в виде расширяемого окна и панели инструментов.

Для явного уточнения версии системы следует вывести окно с информацией о системе (команда About MATLAB (О MATLAB) в меню Help (Помощь)).

Сеанс работы с MATLAB принято именовать *сессией* (session). Сессия, в сущности, является текущим документом, отражающим работу пользователя с системой MATLAB.

В ней имеются строки ввода, вывода и сообщений об ошибках. Входящие в сессию определения переменных и функций, расположенные в рабочей области памяти, но не саму сессию, можно записать на диск (файлы формата .mat), используя команду save (Сохранить). Команда load (Загрузить) позволяет считать с диска данные рабочей области. Фрагменты сессии можно оформить в виде дневника с помощью команды diary (Дневник).

**Файловая система MATLAB.** Система MATLAB состоит из многих

тысяч файлов, находящихся в множестве папок. Полезно иметь представление о содержании основных папок, поскольку это позволяет быстро оценить возможности системы — например, узнать, какие операторы, функции или графические команды входят в систему.

В MATLAB особое значение имеют файлы двух типов — с расширениями `.mat` и `.m`. Первые являются бинарными файлами, в которых могут храниться значения переменных. Вторые представляют собой текстовые файлы, содержащие внешние программы, определения команд и функций системы. Именно к ним относится большая часть команд и функций, в том числе задаваемых пользователем для решения своих специфических задач. Нередко встречаются и файлы с расширением `.c` (коды на языке Си), файлы с откомпилированными кодами MATLAB с расширением `.tex` и другие. Исполняемые файлы имеют расширение `.exe`.

Особое значение имеет папка `MATLAB/TOOLBOX/MATLAB`. В ней содержится набор стандартных `m`-файлов системы. Просмотр этих файлов позволяет детально оценить возможности поставляемой конкретной версии системы. Ниже перечислены основные подпапки с этими файлами (деление на категории условно, на самом деле все подпапки находятся в общей папке `MATLAB/TOOLBOX/MATLAB`).

Подпапка команд общего назначения:

`General` — команды общего назначения: работа со справкой, управление окном MATLAB, взаимодействие с операционной системой и т. д.

Подпапки операторов, конструкций языка и системных функций:

`ops` — операторы и специальные символы;

`tang` — конструкции языка программирования;

`strfun` — строковые функции;

`iofun` — функции ввода/вывода;

`timefun` — функции времени и дат;

`datatypes` — типы и структуры данных.

Подпапки основных математических и матричных функций:

`elmat` — команды создания элементарных матриц и операций с ними;

`elfun` — элементарные математические функции;

`specfun` — специальные математические функции;

`matfun` — матричные функции линейной алгебры;

`datafun` — анализ данных и преобразования Фурье;

`polyfun` — полиномиальные функции и функции интерполяции;

`funfun` — функции функций и функции решения обыкновенных дифференциальных уравнений;

`soarfun` — функции разреженных матриц.

Подпайки команд графики:

`graph2d` — команды двумерной графики;

`graph3d` — команды трехмерной графики;

`specgraph` — команды специальной графики;

`graphics` — команды дескрипторной графики;

`uitools` — графика пользовательского интерфейса.

Полный состав файлов каждой подпапки (их список содержится в файле `сop-tents.m`) можно вывести на просмотр с помощью команды `help имя`, где `имя` — название соответствующей подпапки.

**Интерфейс системы.** Пользовательский интерфейс системы MATLAB многооконный и имеет ряд средств прямого доступа к различным компонентам системы. Бросается в глаза новый пункт меню — `Web`. Он дает прямой выход в Интернет. В панели инструментов добавлена позиция ввода ранее отмененной операции и меню просмотра файловой системы с кнопкой его открытия. Появились окна доступа к компонентам системы `Launch Pad/Workspace` (Панель запуска/ Рабочая область) и окно `Current Directory` (текущей папки). Разработчики системы ввели в позицию `View (Вид)` меню команду `Desktop Layout > Command Windows Only` (Только командное окно). Стоит ее исполнить, как вид окна системы будет очень напоминать добрый старый интерфейс версий MATLAB 5.\*

Если вы все же хотите вкусить прелести нового интерфейса, то исполните в той же позиции меню команду `Desktop Layout > Default` (Интерфейс по умолчанию). Там же вы найдете и другие возможности модификации вида интерфейса системы MATLAB 6.0.

При работе с MATLAB в командном режиме действует простейший строчный редактор. Его команды перечислены в табл. 2.1.

*Таблица 2.1. Команды строчного редактора MATLAB*

Комбинация клавиш	Назначение
<code>Ctrl+b</code>	Перемещение курсора вправо на один символ
<code>Ctrl+f</code>	Перемещение курсора влево на один символ
<code>Ctrl+r</code>	Перемещение курсора вправо на одно слово
<code>Ctrl+l</code>	Перемещение курсора влево на одно слово
<code>Home</code> или <code>Ctrl+a</code>	Перемещение курсора в начало строки
<code>End</code> или <code>Ctrl+e</code>	Перемещение курсора в конец строки
<code>Ctrl+pu</code> <code>Ctrl+n</code>	Перелистывание предыдущих команд вверх или вниз для подстановки в строку ввода
<code>Del</code> или <code>Ctrl+d</code>	Стирание символа справа от курсора
<code>Ctrl+h</code>	Стирание символа слева от курсора
<code>Ctrl+k</code>	Стирание до конца строки
<code>Esc</code>	Очистка строки ввода
<code>Ins</code>	Включение/выключение режима вставки
<code>PgUp</code>	Перелистывание страниц сессии вверх
<code>PgDn</code>	Перелистывание страниц сессии вниз

Полезно сразу усвоить некоторые команды управления окном командного режима:

`clc` — очищает экран и размещает курсор в левом верхнем углу пустого экрана.

`home` — возвращает курсор в левый верхний угол окна.

`echo <file_name> on` — включает режим вывода на экран текста Script-файла (файла-сценария).

`echo <file_name> off` — выключает режим вывода на экран текста Script-файла.

`echo <file_name>` — меняет режим вывода на противоположный.

`echo on all` — включает режим вывода на экран текста всех `m`-файлов.

`echo off all` — отключает режим вывода на экран текста всех `m`-файлов.

`more on` — включает режим постраничного вывода (полезен при просмотре больших `m`-файлов).

`more off` — отключает режим постраничного вывода (в этом случае для просмотра больших файлов надо пользоваться линейкой прокрутки).

В версии MATLAB 6.0 обе команды `clc` и `home` действуют аналогично — очищают экран и помещают курсор в левый верхний угол окна командного режима работы. Команды `echo` позволяют включать или выключать отображение текстов `m`-файлов при каждом обращении к ним. Как правило, отображение текста файлов сильно загромождает экран и часто не является необходимым. При больших размерах файлов начало их текста (листинга) убегает далеко за пределы области просмотра (текущего окна командного режима). Поэтому для просмотра длинных листингов файлов полезно включить постраничный вывод командой `more on`.

## 6.2. Визуализация и графические средства

В последнее время создатели математических систем уделяют огромное внимание визуализации решения математических задач. Говоря проще, это означает, что постановка и описание решаемой задачи и результаты решения должны быть предельно понятными не только тем, кто решает задачи, но и тем, кто в дальнейшем их изучает или просто просматривает. Большую роль в визуализации решения математических задач играет графическое представление результатов, причем как конечных, так и промежуточных. Визуализация постановки задачи в MATLAB решается применением приложения Notebook и назначением именам функций достаточно ясных имен (идентификаторов). А визуализация результатов вычислений достигается применением обширных средств графики, в том числе анимационной, а также использованием средств символьной математики.

Графические средства Handle Graphics (дескрипторная или описательная графика) позволяют создавать полноценные объекты графики высокого разрешения, как геометрического, так и цветового. Возможности этой графики поддерживаются объектно-ориентированным программированием, средства которого также имеются в языке программирования системы MATLAB.

Применение дескрипторной графики позволяет создавать и типовые элементы пользовательского интерфейса — кнопки, меню, информационные и инструментальные панели и т. д., то есть реализовать элементы визуально-ориентированного программирования. Графики выводятся отдельно от текстов в отдельных окнах. На одном графике можно представить множество



кривых, отличающихся цветом (при цветном дисплее) и отличительными символами (кружками, крестиками, прямоугольниками и т. д.). Графики можно выводить в одно или в несколько окон. Наконец, в статьях и книгах формата Notebook, реализованных при совместной работе системы MATLAB с популярным текстовым процессором Microsoft Word 95/97/2000, графики могут располагаться вместе с текстом, формулами и результатами вычислений (числами, векторами и матрицами, таблицами и т. д.). В этом случае степень визуализации оказывается особенно высокой, поскольку документы класса Notebook по существу являются превосходно оформленными электронными книгами с действующими (вычисляемыми) примерами.

Введен также ряд средств на основе графического интерфейса пользователя (GUI — Graphic User Interface), привычного для операционных систем Windows 95/98/Me/ 2000/NT4. Это панели инструментов, редактор и отладчик m-файлов, красочная демонстрация возможностей и т. д. Есть и возможность создавать свои средства пользовательского интерфейса.

С графическими возможностями вы можете познакомиться через учебные пособия [1-5] или через Help системы MATLAB в разделе **Graphics**. Здесь мы приводим 2 примера построения графиков с помощью процедур **plot(x,y)** и **plotyy(x,y1,x,y2)**;

**Пример 1.** Построение графика с помощью процедуры **plot(x,y)**  
`x = 0:0.02:20; % интервал значений по x`  
`y = 50*exp(-0.2*x).*sin(x); % функция`  
`figure(1); plot(x,y); % построение графика`

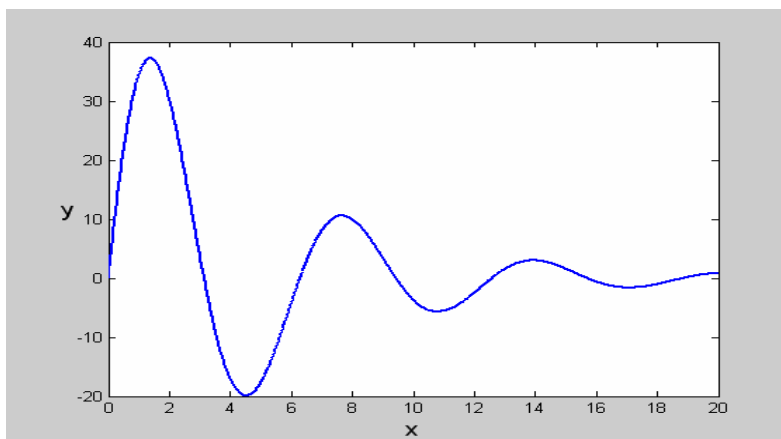


Рис.6.2

Здесь сохранение графика проводилось с помощью **Edit → Copy Figure** в окне **Figure № 1**

**Пример 2.** Выдача графиков с разными масштабами.

```
x = 0:0.01:20; % интервал значений по x
y1 = 200*exp(-0.2*x).*sin(x); % первая функция
y2 = 0.8*exp(-0.5*x).*sin(10*x); % вторая функция
figure(2);[AX,H1,H2] = plotyy(x,y1,x,y2,'plot'); % построение графиков
```

Для оформления графика можно использовать следующие операторы:

```
set(get(AX(1),'Ylabel'),'String','Left Y-axis'); % запись метки по оси у слева
set(get(AX(2),'Ylabel'),'String','Right Y-axis'); % запись метки по оси у справа
xlabel('Zero to 20 \musec.');
```

`% метка по оси x`  
`title('Labeling plotyy');` % обозначение графика  
`set(H1,'LineStyle','--');` % сплошная линия графика  
`set(H2,'LineStyle',':');` % пунктирная линия графика

Оформление внешнего вида графика (метки на осях, оглавление и др.) можно проводить также с помощью меню **Edit** в открытом окне нашего примера **Figure № 2**.

На рис.6.3. сохранение графика проводилось через **File** → **Export** в окне **Figure № 2** формате \*.emf.

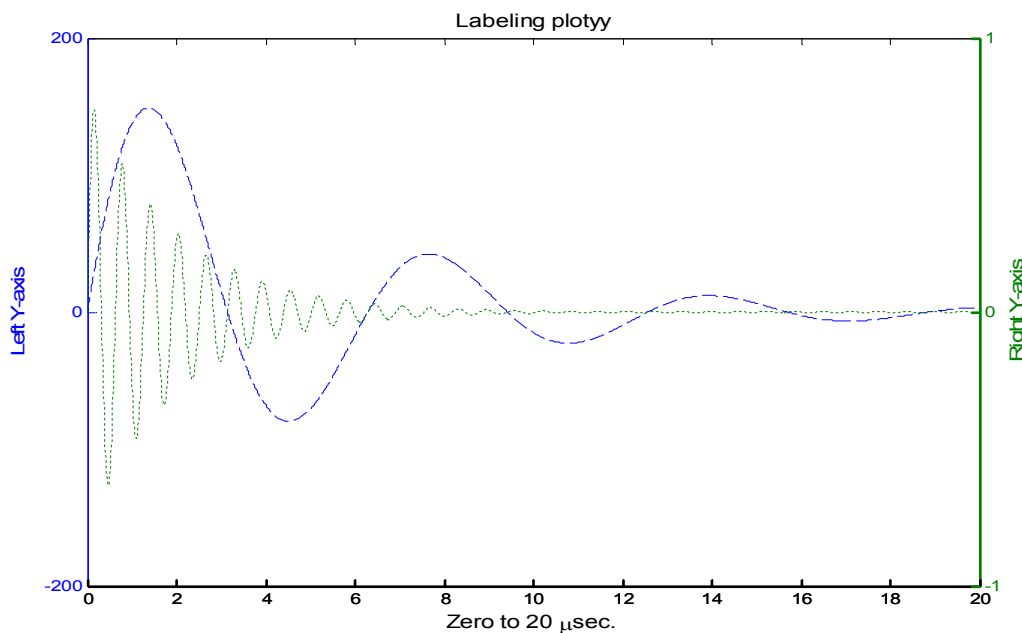


Рис.6.3

Сохранение можно проводить с различным видом расширения \*.emf, \*.bmp, \*.jpg, \*.eps и другими, в зависимости от необходимости использования в том или ином документе.

### 6.3. Решение обыкновенных дифференциальных уравнений в системе MATLAB

Анализ поведения многих систем и устройств в динамике, а также решение многих задач в теории колебаний и в поведении упругих оболочек обычно базируются на решении систем *обыкновенных дифференциальных уравнений* (ОДУ). Их, как правило, представляют в виде системы из дифференциальных уравнений первого порядка в форме Коши:

$$\frac{dy}{dt} = f(y, t)$$

с граничными условиями  $y(t_0, t_{end}, \mathbf{b})$ , где  $t_0, t_{end}$  — начальные и конечные точки интервалов. Параметр  $t$  не обязательно означает время, хотя чаще всего решение дифференциальных уравнений ищется во временной области. Вектор  $\mathbf{b}$  задает начальные и конечные условия.

Ниже коротко описаны численные методы решения обыкновенных дифференциальных уравнений (ОДУ) и некоторые вспомогательные функции, полезные для решения систем ОДУ. Дается представление о пакете расширения, решающем дифференциальные уравнения в частных производных.

Для решения систем ОДУ в MATLAB реализованы различные методы. Их реализации названы *решателями* ОДУ.

В этом разделе обобщенное название solver (решатель) означает один из возможных численных методов решения ОДУ. Решатели реализуют следующие методы решения систем дифференциальных уравнений, причем для решения жестких систем уравнений рекомендуется использовать только специальные решатели ode45s, ode23s, ode23t, ode23tb:

ode45 (f, interval, y0 [, options]) — одношаговые явные методы Рунге-Кутты 4-го и 5-го порядка. Это классический метод, рекомендуемый для начальной пробы решения. Во многих случаях он дает хорошие результаты;

ode23(f, interval, y0 [, options]) — одношаговые явные методы Рунге-Кутты 2-го и 4-го порядка. При умеренной жесткости системы ОДУ и низких требованиях к точности этот метод может дать выигрыш в скорости решения;

ode113 (f, interval, y0 [, options]) — многошаговый метод Адамса-Башворта-Мултона переменного порядка. Это адаптивный метод, который может обеспечить высокую точность решения

ode23tb (f, interval, y0 [, options]) — неявный метод Рунге-Кутты в начале решения и метод, использующий формулы обратного дифференцирования 2-го порядка в последующем

Несмотря на сравнительно низкую точность, этот метод может оказаться более эффективным, чем ode15s;

ode15s (f, interval, y0 [, options]) — многошаговый метод переменного порядка (от 1 до 5, по умолчанию 5), использующий формулы численного дифференцирования. Это адаптивный метод, его стоит применять, если решатель ode45 не обеспечивает решения;

ode23s (f, interval, y0 [, options]) — одношаговый метод, использующий модифицированную формулу Розенброка 2-го порядка. Может обеспечить высокую скорость вычислений при низкой точности решения жесткой системы дифференциальных уравнений;

ode23t (f, interval, y0 [, options]) — метод трапеций с интерполяцией. Этот метод дает хорошие результаты при решении задач, описывающих колебательные системы с почти гармоническим выходным сигналом;

bvp4c служит для проблемы граничных значений систем дифференци-

альных уравнений вида  $y' = f(t, y)$ ,  $F(y(a), y(b), p) = 0$  (краевая задача);

где  $y'$  нужен для решения систем параболических и эллиптических дифференциальных уравнений в частных производных, введен в ядро системы для поддержки новых графических функций Open GL, пакет расширения Partial Differential Equations Toolbox содержит более мощные средства.

Входными параметрами этих функций являются:

- $f$  – вектор функция для вычисления правой части уравнения или системы уравнений;
- $interval$  – массив из двух чисел, определяющий интервал интегрирования дифференциального уравнения или системы. Для получения решений в конкретные моменты времени  $t_0, t_1, \dots, t_{final}$  (расположенные в порядке уменьшения или увеличения) нужно использовать  $interval = [t_0 \ t_1 \dots \ t_{final}]$ ;
- $y_0$  – вектор начальных условий;
- $options$  – параметры управления ходом решения дифференциального уравнения или системы аргумент, создаваемый функцией `odeset` (еще одна функция — `odeget` или `bvpget` (только для `bvp4c`) — позволяет вывести параметры, установленные по умолчанию или с помощью функции `odeset` / `bvpset`);
- $p_1, p_2, \dots$  — произвольные параметры, передаваемые в функцию  $F$ .

Все решатели могут решать системы уравнений явного вида  $y' = f(t, y)$ . Решатели `ode15s` и `ode23t` способны найти корни дифференциально-алгебраических уравнений  $M(t)y' = f(t, y)$ , где  $M$  называется матрицей массы. Решатели `ode15s`, `ode23s`, `ode23t` и `ode23tb` могут решать уравнения неявного вида  $M(t, y) y' = f(t, y)$ . И наконец, все решатели, за исключением `ode23s`, который требует постоянства матрицы массы, и `bvp4c`, могут находить корни матричного уравнения вида  $M(t, y) y' = f(t, y)$ . `ode23tb`, `ode23s` служат для решения жестких дифференциальных уравнений. `ode15s` - жестких дифференциальных и дифференциально-алгебраических уравнений, `ode23t` - умеренно жестких дифференциальных и дифференциально-алгебраических уравнений.

$T, Y$  – матрица решений  $Y$ , где каждая строка соответствует времени, возвращенном в векторе-столбце  $T$ .

Перейдем к описанию функций для решения систем дифференциальных уравнений:

$[T, Y] = \text{solver}(@f, interval, y_0)$  — где вместо `solver` подставляем имя конкретного решателя — интегрирует систему дифференциальных уравнений вида  $y' = F(t, y)$  на интервале `interval` с начальными условиями  $y_0$ . `@f` — дескриптор (правая часть) ODE-функции. Каждая строка в массиве решений  $Y$  соответствует значению времени, возвращаемому в векторе-столбце  $T$ ;

$[T, Y] = \text{solver}(@f, interval, y_0, options)$  — дает решение, подобное описанному выше, но с параметрами, определяемыми значениями аргумента `options`, созданного функцией `odeset`;

$[T, Y] = \text{solver}(@f, interval, y_0, options, p_1, p_2, \dots)$  — дает решение, подобное описанному выше, передавая дополнительные параметры  $p_1, p_2, \dots$  в  $m$ -файл  $F$

всякий раз, когда он вызывается. Используйте options=[ ], если никакие параметры не задаются;

Параметры интегрирования (options) могут быть определены и в m-файле, и в командной строке с помощью команды odeset. Если параметр определен в обоих местах, определение в командной строке имеет приоритет.

Решатели используют в списке различные параметры. В приведенной ниже таблице они даны для решателей обычных (в том числе жестких) дифференциальных уравнений.

Специалистов в области численных методов решения дифференциальных уравнений в частных производных, может заинтересовать обширный пакет Partial Differential Equations Toolbox (PDETB). Хотя этот пакет является самостоятельным приложением и в ядро MATLAB не входит.

Параметры	ode45	ode23	Ode113	ode15s	ode23s
RelTol,AbsTol	+	+	+	+	+
OutputFcn, OutputSel, Refine, Stats	+	+	+	+	+
Events	+	+	+	+	+
MaxStep, InitialStep	+	+	+	+	+
Jacobian, JPattern, Vectorized	-	-	-	+	+
Jpattern, Vectorized	-	-	-	+	+
Mass	+	+	+	+	+
MStateDependence	+	+	+	+	-
MassConstant	-	-	-	+	-
MaxOrder, BDF	-	-	-	+	-

Способы решения дифференциальных уравнений в системе MATLAB также как и в других системах могут быть разные, в частности с помощью использования численных методов (программированием) и использованием готовых процедур самого пакета.

**Метод использования численных методов.** Рассмотрим задачу Коши:

$$\frac{dy}{dt} = -y^2 + \frac{1}{t^2}$$

$$y(1)=0$$

Для решения этого уравнения методом Эйлера создадим следующую функцию в M – файле.

```
% функция задающая решение по Эйлери
function [y,t]=eiler(a,b,n,y0)
h=(b-a)/n;
y(1)=y0;
for i=1:n
    t(i)=a+i*h;
```

```

end
for i=1:n-1
    y(i+1)=y(i)+h*(-y(i)*y(i)+1/(t(i)*t(i)));
end
end

```

Эта функция должна быть сохранена например в следующем каталоге :  
 C:\MATLAB6p5\toolbox\matlab\funfun, т.е. там где хранятся все функции системы MATLAB. При начале вызова этой функции необходимо удостовериться о наличии директории, где располагаются ваши М-файлы

Current Directory

т.е. в каком каталоге находится необходимая функция. Если необходимые функции хранятся в другом каталоге, то вы должны указать соответствующую директорию.

```

>>[y,t]=eiler(0.01,100,1000,0.1)
>>figure(1);plot(t,y);

```

Найдем численное решение дифференциального уравнения с помощью одного из решателей ode113, ode23, ode45 и др.

Они применяются для разных систем (мягкие, жесткие) и используют различные схемы численного интегрирования. Но для любой из них обязательно нужно задать систему, которую надо решить, начальные условия, интервал и шаг поиска решения.

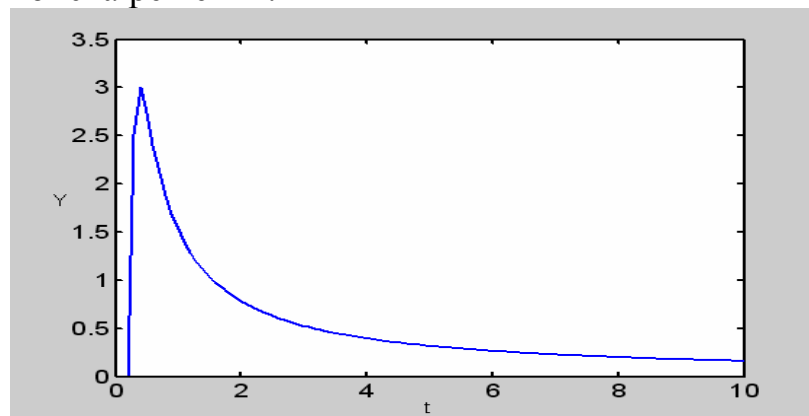


Рис.6.4

Систему уравнений нужно привести к нормальному виду и записать в файл функцию, которая по данным значениям аргумента  $x$  и всех функций  $y_1, y_2, \dots, y_n$  вычисляет производные в этой точке. Этот файл нужно назвать так, как называется функция (расширение дать  $m$ ), и поместить в какой-либо каталог, доступный MATLAB.

Например:

```

function f=ff(t,y)
f=-y(1)*y(1)+1/(t*t);
end

```

Решение

```
>>[y1,t1]=ode23(@ff,[0.2 10], 0.001)
```

```
>> figure(3);plot(y,t,':',y1,t1);
```

Результаты решения по методу Эйлера и с помощью решателя **ode23** представлены на рис. 6.5.

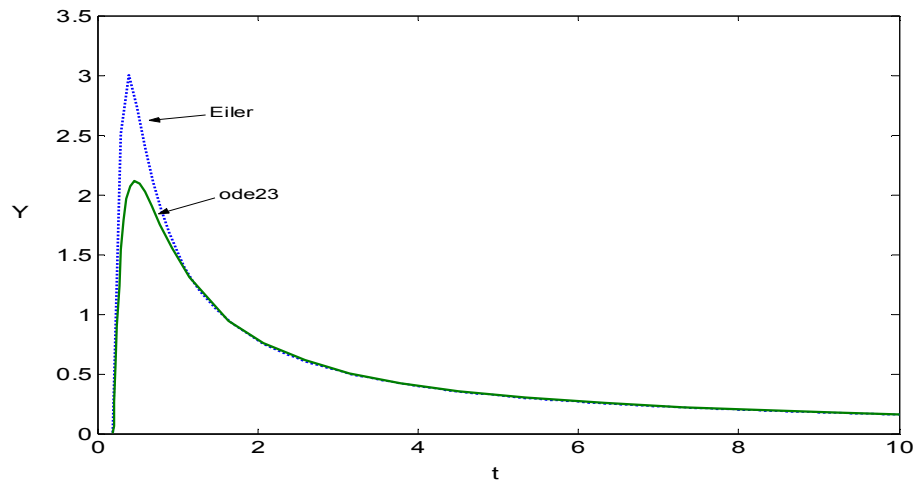


Рис.6.5

**Решение систем дифференциальных уравнений.** Рассмотрим задачу о маятнике Фуко. Математическую модель этой задачи можно представить в виде:

$$\begin{aligned}\frac{dy_1}{dt} &= 2y_1\omega + \omega^2 y_3 - g \frac{y_3}{L}, \\ \frac{dy_2}{dt} &= 2y_2\omega + \omega^2 y_4 - g \frac{y_4}{L}, \quad (3.58) \\ \frac{dy_3}{dt} &= y_1, \quad \frac{dy_4}{dt} = y_2.\end{aligned}$$

Здесь  $\omega$  - частота колебания маятника Фуко,  $L$  – длина маятника,  $g$  – значение ускорения свободного падения.

Создадим М-файл

```
function dy=fuko(t,y);  
g=9.8;  
L=50;  
w=0.04;  
dy=zeros(4,1)  
dy(1)=2*y(2)*w+w*w*y(3)-g*y(3)/L;  
dy(2)=-2*y(1)*w+w*w*y(4)-g*y(4)/L;  
dy(3)=y(1);  
dy(4)=y(2);
```

**end**

Решение находим с помощью решателя ode23 и визуализируем в виде графиков (рис.6.6)

```
>> [T,Y]=ode23(@fuko,[0 100],[0;0;1;0])  
>> figure(5); plot(T,Y,'-k');  
>> figure(6); plot(Y(:,3),Y(:,4),'k');
```

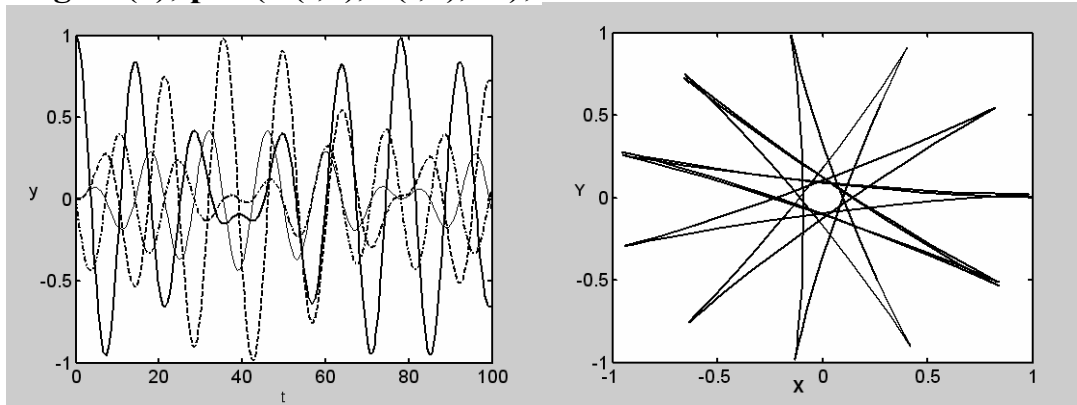


Рис.6.6

### Аналитическое решение дифференциальных уравнений.

Рассмотрим следующее уравнение

$$y''+2y'+y = x * \sin(2 * x)$$

```
>> clear all % очистили память  
eq='D2y+2*Dy+y-x*sin(2*x)'; % уравнение  
x0=-10; % начальная точка  
xk=10; % конечная точка  
y0=1; % начальное условие  
y10=-1; % начальное условие  
ic1=sprintf ('y (%d) =%d', x0,y0); % строка с начальным условием  
ic2=sprintf ('Dy (%d) =%d' , x0,y10); % строка с начальным услови-  
ем  
fprintf (' Дифференциальное уравнение : \n % s=0\n ', eq);  
fprintf (' Начальные условия : \n % s \n % s\n ', ic1, ic2);
```

Дифференциальное уравнение :

$$D2y+2*Dy+y-x*\sin(2*x)=0$$

Начальные условия :

$$y(0) = 1$$

$$Dy(0) = -1$$

Нахождение аналитического решения



```
>> [y]= dsolve('D2y=-2*Dy-y+x*sin(2*x)', 'y(0)=1', 'Dy(0)=0', 'x')
y = -4/25*cos(2*x)*x-4/125*cos(2*x)-3/25*x*sin(2*x)+22/125*sin(2*x)
+129/125*exp(-x)+21/25*exp(-x)*x
```

Для численного решения уравнения приведем наше уравнение 2-го порядка к системе двух уравнений 1-го порядка заменой:

Создадим М –файл

```
function dy=fff(x,y)
dy=zeros(2,1);
dy(1)=y(2);
dy(2)=-2*y(2)-y(1)+x*sin(2*x);
end
```

Для численного решения системы дифференциальных уравнений, кроме функции вычисления правых частей, нужно задать также начальные условия и массив точек, в которых будет вычисляться решение.

Начальные условия у нас есть — это переменные  $y_0$  и  $y_{10}$ . Решаем систему уравнений решателем ode23.

```
[X,Y]=ode23(@fff,[-20 20],[1;-1])
```

В результате получаем: массив аргументов  $X$  — это точки, в которых найдено решение, и  $Y$  — двумерный массив функций. Каждый столбец массива  $Y$  — это одна из переменных  $y_1, y_2, \dots, y_n$  в точках  $X$ . В нашем случае 1-й столбец  $Y$  — это функция  $y(x)$ , а 2-й — ее производная. Для построения графика нам нужен будет 1-й столбец массива  $Y$ .

```
figure(6);plot(Y(:,1));
```

Надписываем заголовок и метки осей выбранным шрифтом. На рис.6.7. сохранение графика проводилось через **File** → **Export** в окне **Figure № 6** формате \*.tif.

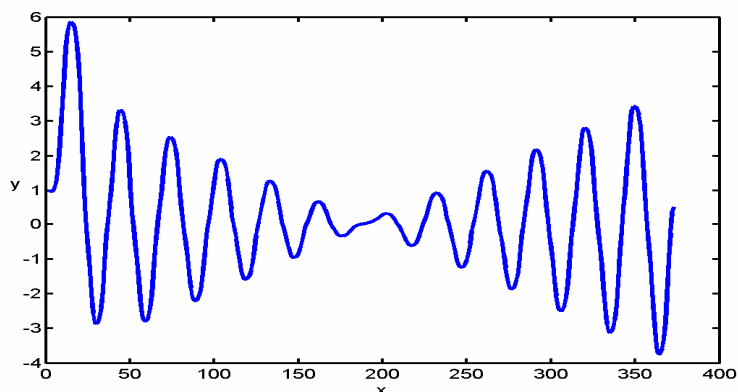


Рис.6.7

Для более глубокого ознакомления с системой MATLAB рекомендуем литературу [1-7].

## ГЛАВА 7. ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ ОБЪЕКТОВ С ПОМОЩЬЮ СИСТЕМЫ VISSIM

Отличительная особенность системы технического моделирования VisSim — ее явная ориентация на открытое математическое моделирование. Набор блоков по сравнению с системой Simulink для моделей в этой системе резко сокращен, и большая часть блоков ориентирована на реализацию именно математических и логических операций. Это позволяет создавать математически прозрачные модели с любыми описывающими их зависимостями, в том числе реализованными в системах компьютерной математики, например, таких известных и получивших массовое признание, как Mathcad и MATLAB.

Система VisSim поставляется и интегрируется с массовыми системами компьютерной математики Mathcad 2000/2001/20011/11 [4], благодаря чему она вполне доступна и имеется на ряде выпущенных в России CD-ROM. Интеграция VisSim 7.5 с математическими системами Mathcad и MATLAB открывает новые возможности выполнения самых серьезных вычислений, причем как в ходе моделирования, так и при обработке его результатов.

### 7.1. Знакомство с системой технического проектирования VisSim

Разработчиками VisSim рекламируется как самая быстрая система моделирования. И действительно, время моделирования систем и устройств в VisSim от 3 до 8 раз меньше, чем у других систем моделирования. Это большое достоинство системы, нередко оправдывающее переход к ее применению. Поэтому тема данного параграфа может быть актуальной. Запустив систему VisSim из Windows, вы увидите на экране окно (рис. 7.1). Над ним видна строка с основными элементами интерфейса. Опции главного меню, содержащиеся в этой строке, легко изучить самостоятельно; некоторые из них очень похожи на стандартные опции, принятые в текстовых редакторах Windows.

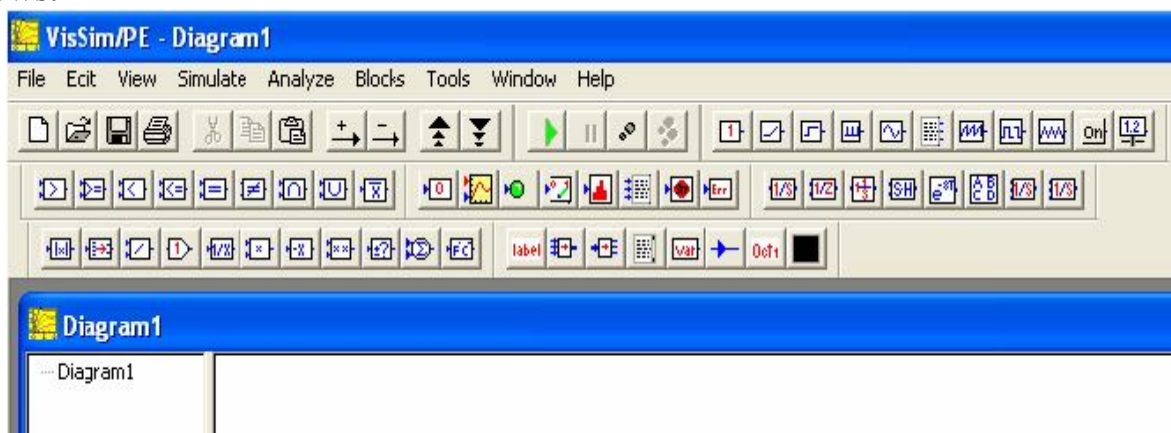


Рис. 7.1

Работа с документами VisSim обычно не требует обязательного использования возможностей главного меню, так как основные из них дублируются

кнопками на панели инструментов, которые расположены в удобных, перемещаемых с помощью мыши наборных панелях–палитрах. Наборные панели появляются в окне редактирования документов при активизации кнопок. Они служат для вывода заготовок – шаблонов математических знаков (цифр, знаков арифметических операций, матриц, знаков интеграла, производных, приделов и др.).

Система VisSim предназначена для решения задач математического моделирования, относящихся к следующим классам:

- линейные системы;
- нелинейные системы;
- непрерывные во времени системы;
- дискретные во времени системы;
- системы с изменяемыми во времени параметрами;
- гибридные системы;
- многоцелевые и многокомпонентные системы;
- одновходовые и одновыходные (одномерные) системы *SISO*;
- многовходовые и многовыходовые (многомерные) системы *MIMO*;

Система VisSim не имеет явной ориентации на какой-то класс моделирования. Это универсальная система, допускающая достаточно простое расширение и обеспечивающая легкую адаптацию под решение тех или иных конкретных задач пользователя. Тем не менее, можно считать, что наиболее удобна данная система для решения инженерных задач в области автоматического регулирования и управления, а также при моделировании различных физических, химических, экономических и прочих явлений и систем.

Для построения моделей в системе VisSim используются блоки, которые хранятся в библиотеке блоков и могут браться из нее, переноситься в окно модели и соединяться друг с другом.

Библиотека блоков, представленная в позиции Blocks (Блоки) меню (рис. 7.2) и инструментальными панелями, содержит следующие «тома»:

- **Animation** — блоки создания анимационных клипов;
- **Annotation** — блоки создания комментариев и определения переменных;
- **Arithmetic** — блоки арифметических и близких к ним операций;
- **Boolean** — блоки задания операций Булевой алгебры;
- **DDE** — блоки интерфейса;
- **Integration** — блоки задания операций интегрирования;
- **Linear Systems** — блоки задания параметров пространства состояний линейных систем и их передаточных функций;
- **MATLAB Interface** — блоки интеграции с матричной системой MATLAB;
- **Matrix Operations** — блоки задания матричных операций;
- **Nonlinear** — блоки нелинейных операций и создания нелинейных систем;
- **Optimization** — блоки задания операций оптимизации;
- **Random Generator** — блоки генерации случайных чисел;
- **Real Time** — блоки для систем реального времени;

- **Signal Consumer** — блоки регистрации, индикации и построения графиков сигналов;
- **Signal Producer** — блоки создания сигналов;
- **Time Delay** — блоки создания временной задержки;
- **Transcendental** — блоки задания трансцендентных математических функций;
- **General** — функции общего характера.

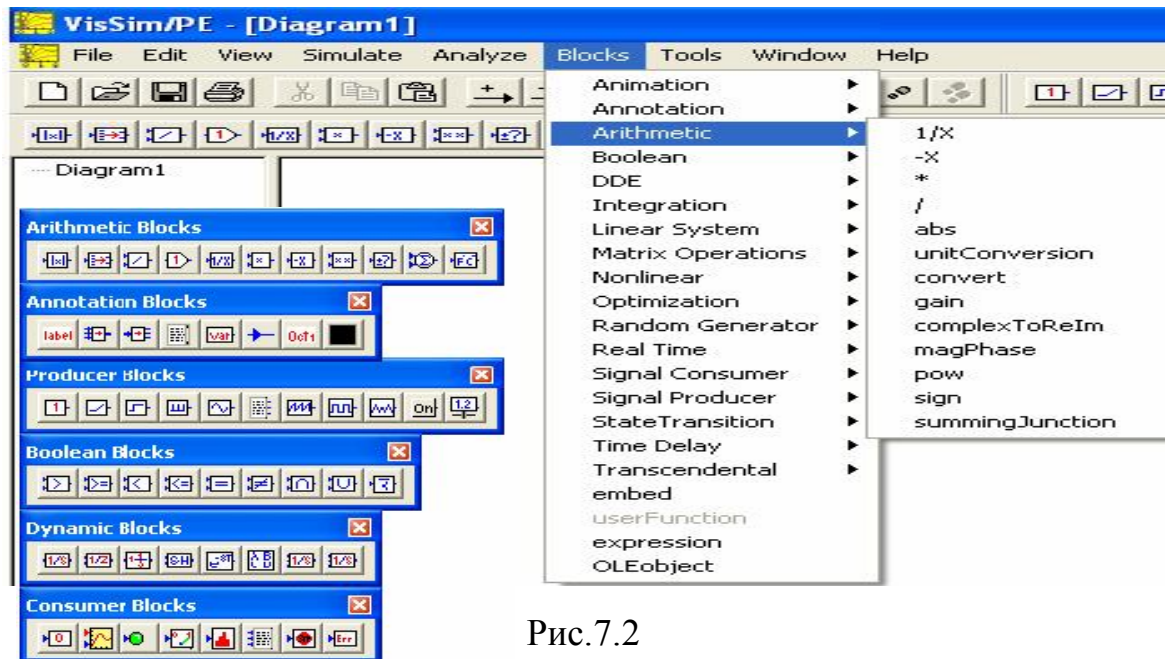


Рис.7.2

На рис.7.3 представлены блоки арифметических, интегро-дифференциальных и логических операций.

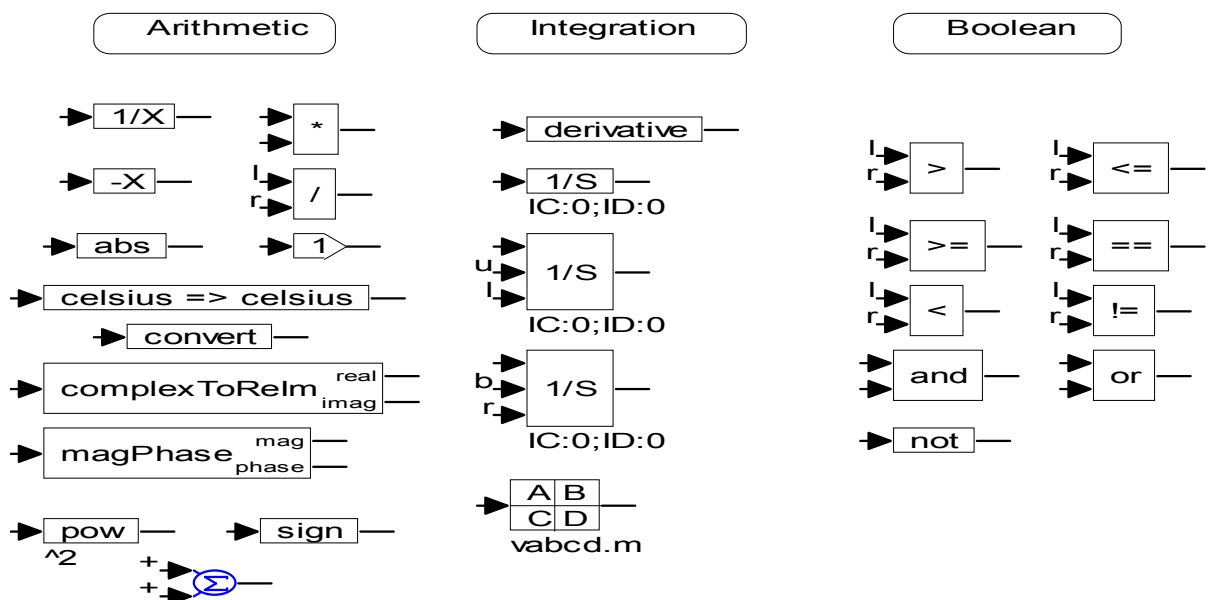


Рис.7.3.

## 7.2. Технологии моделирования дифференциальных уравнений

При разработке автоматических систем управления достаточно широко используются методы физического или математического моделирования. При физическом моделировании исследования проводятся на модели, физически однородной с объектом-оригиналом и отличающейся от него в некоторых случаях только количественно. При математическом моделировании исследования проводятся на модели, которая описывается такими же уравнениями, что и моделируемое устройство.

Создание математических моделей возможно с помощью аналоговых вычислительных устройств. Исторически первыми для этих целей использовались аналоговые устройства, достоинством которых по сравнению с цифровыми является простота модели и самого процесса моделирования, а также наглядность результатов при непрерывном воспроизведении заданной математической зависимости. Система Electron Workbench позволяет проводить моделирование различных устройств автоматики с сохранением наглядности и простоты аналоговых моделирующих устройств.

Более подробно для работы с системой VisSim мы рекомендуем книги [1-3] и сайт <http://model.exponenta.ru/help/0050.htm>.

Система Vissim позволяет проводить интегрирование в цифровой форме – которая является более устойчивой операцией, чем дифференцирование, здесь необходимо преобразовывать исходные дифференциальные уравнения (ДУ) в операторной форме таким образом, чтобы оператор Лапласа входил в уравнение только с отрицательными степенями (требуется разделить и левую и правую часть ДУ на оператор Лапласа в соответствующей степени). Это основной принцип построения моделей в программе VisSim.

### 7.2.1. Решение дифференциальных уравнений

Поскольку моделирование автоматических систем управления чаще всего сводится к решению дифференциальных уравнений, то в первую очередь рассмотрим устройство для их решения, которые в аналоговой вычислительной технике часто называют дифференциальными анализаторами.

В дифференциальных анализаторах реализуются два метода интегрирования дифференциальных уравнений. Один из них основан на повышении порядка производных искомой функции, а другой – на его понижении.

Достоинством метода понижения порядка, нашего наибольшее распространение на практике, является высокая точность, обусловленная применением интеграторов, что устраняет в определенной мере влияние различных флуктуаций, возникающих в схеме.

Рассмотрим технологию решения дифференциального уравнения порядка методом понижения порядка

$$a \cdot \frac{d^3 x}{dt^3} + b \cdot \frac{d^2 x}{dt^2} + c \cdot \frac{dx}{dt} + d \cdot x = \varphi(t).$$

Решим его относительно старшей производной

$$\frac{d^3 x}{dt^3} = f(t) - A_1 \cdot \frac{d^2 x}{dt^2} - A_2 \cdot \frac{dx}{dt} + A_3 \cdot x, \quad (7.1)$$

где

$$f(t) = \frac{\varphi(t)}{a}, \quad A_1 = \frac{b}{a}, \quad A_2 = \frac{c}{a}, \quad A_3 = \frac{d}{a}.$$

Структурная схема решения модели дифференциального уравнения представлена на рис.7.4.

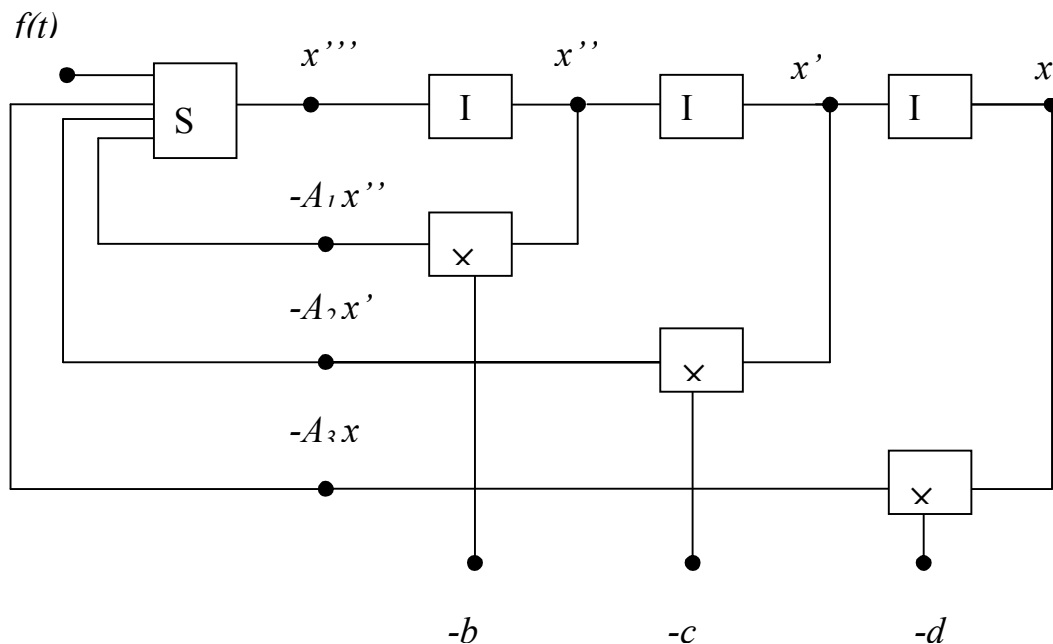


Рис. 7.4. Структурная схема решения дифференциального уравнения понижением порядка

В системе Vissim уравнение (7.1) используя преобразование Лапласа можно привести к виду

$$s^3 x = f - A_1 s^2 x - A_2 s^1 x - A_3 x,$$

тогда соответствующая блок-схема будет выглядеть, как показано на рис.7.5:

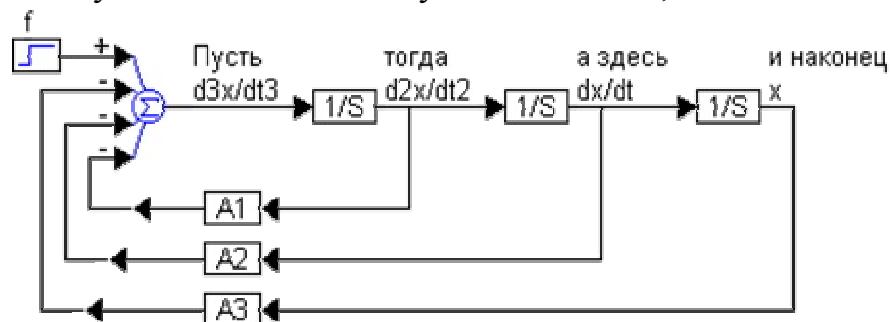


Рис.7.5

где  $x$  - координата системы, относительно которой решено ДУ - это может быть ошибка системы  $x(t)$  или воздействие на объект  $u(t)$ , или выходная координата -  $y(t)$ ,  $A_1, \dots, A_3$  - постоянные коэффициенты (если система не

является зависимой от параметра) - суммы и произведения постоянных времени  $T_j$ , а так же коэффициенты усиления  $K_n$ ,  $f$  - любое воздействие на систему - сигнал задания  $g(t)$ , или помехи  $f_j(t)$ .

### 7.2.2. Моделирование дифференциальных уравнений первого и второго порядков с использованием блока *variable*

Рассмотрим способ использования блока **variable** системы схемотехнического моделирования для решения дифференциальных уравнений на конкретных примерах.

**Пример 1.** Решение уравнений первого порядка

Рассмотрим уравнение первого порядка

$$\frac{dy}{dt} = -y + r(t), \quad r(t) = 1.$$

Для построения схемотехнической модели дифференциального уравнения, в левой части оставляем старшую производную, все остальные члены переносим в правую часть. Соберем схему с использованием суммы и значение присвоим переменной **ydot**, проведем интегрирование, результат просуммируем с начальным условием (в нашем расчете оно равно -7.2). Результат суммирования присвоим переменной **Y**, которую соединим осциллографом. На осциллографе будет выдан результат (Рис.7.6).

Using a 1/s (integrator) block to perform numerical integration.

Initial condition set externally.

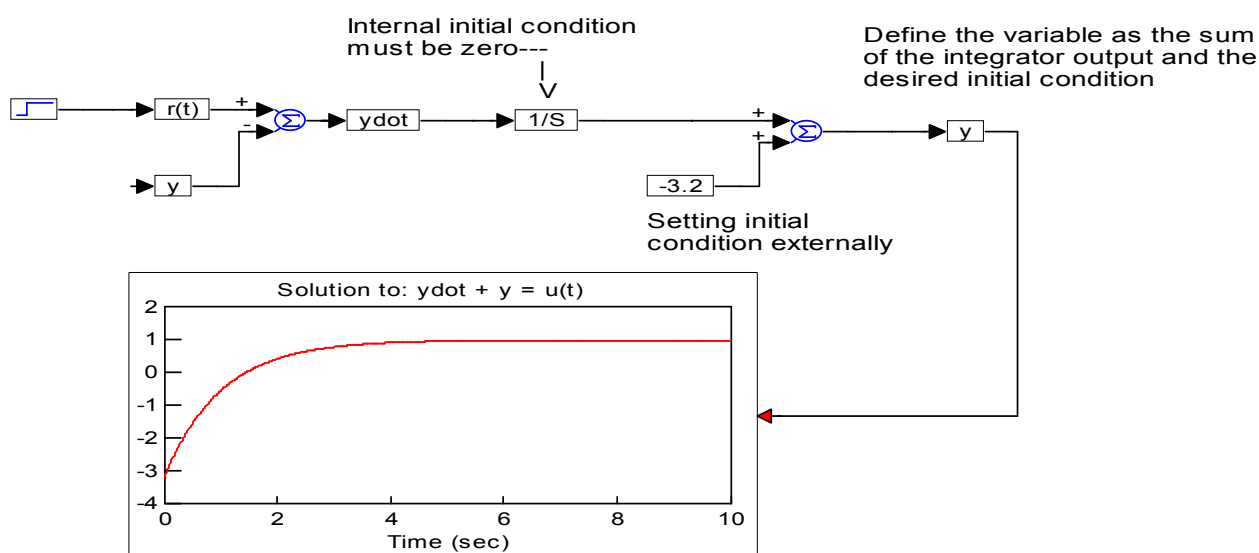


Рис.7.6

**Пример 2.** Решение уравнения второго порядка с помощью блока **variable** проводится аналогично примеру 1.

$$r(t) - y \frac{dy}{dt} - 2y = \frac{d^2 y}{dt^2}, \quad r(t) = 1.$$



Using a 1/s (integrator) block to perform numerical integration (nonlinear equation.)

Initial condition set externally.

$$r(t) - y \cdot \dot{y} - 2y = \ddot{y}$$

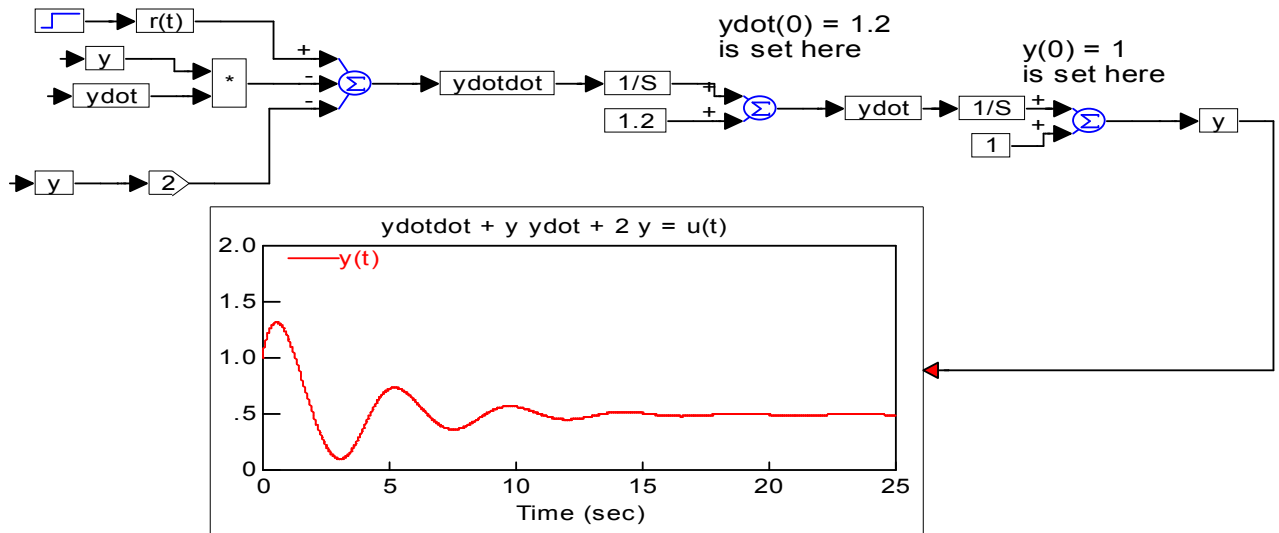


Рис.7.7

**Пример 3.** При решении уравнений можно опустить блок **variable**, в этом случае необходимо результаты операции присоединить соответствующим блокам.

Решение дифференциального уравнения  $\frac{d^2y}{dt^2} = -g \cdot \frac{dy}{dt} - y \cdot w^2$

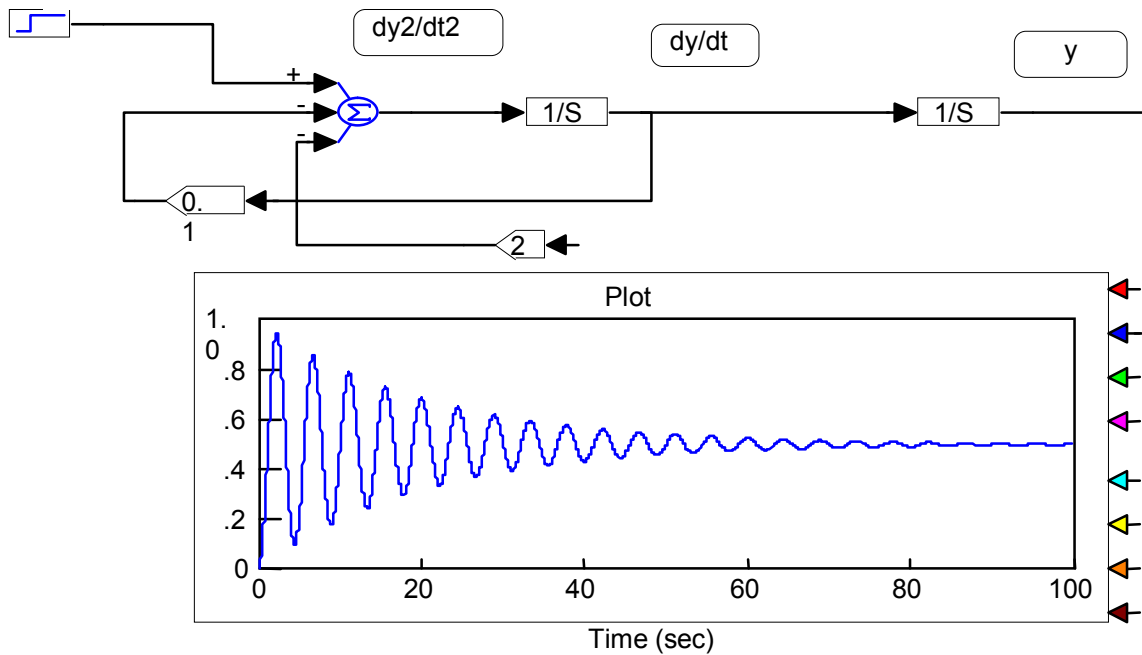


Рис.7.8

Рассмотрим этот метод на примере уравнения второго порядка



$$\frac{d^2 y}{dt^2} = -k \frac{dy}{dt} - \omega^2 y .$$

Листинг решения дифференциального уравнения представлен на рис.7.8.

**Пример 4.** Схмотехническое моделирование движения упругого мяча, брошенного под углом к горизонту.

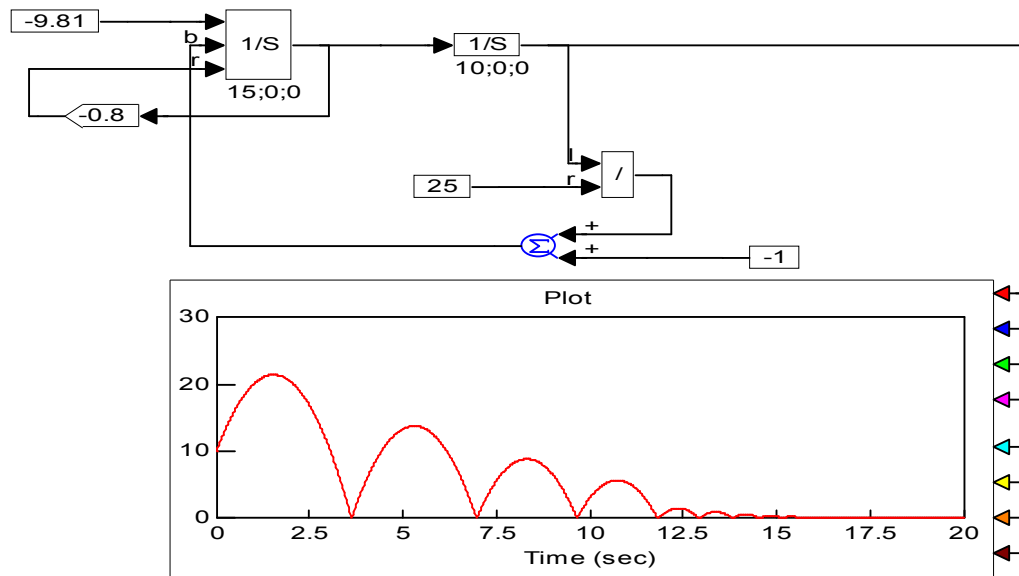


Рис.7.9

### 7.2.3. Схмотехнические модели, описанные типовыми звеньями и передаточными функциями

Пусть математическая модель представлена в виде дифференциального уравнения. Построение модели можно проводить по структурной схеме исследуемой системы, в этом случае дифференциальные уравнения должны быть представлены в виде типовых звеньев и передаточных функций.

С точки зрения входных воздействий, переменных состояния и выходных переменных такую математическую модель можно представить как преобразователь входных воздействий, начальных и граничных условий в переменные состояния и выходные величины. Математическая модель представляется в виде сочетаний некоторых элементов в виде математических преобразований (сложений, вычитаний, умножений, дифференцирование, интегрирование). Эти элементы получили название динамических звеньев. Обозначение динамического звена приведено на рис. 7.10.

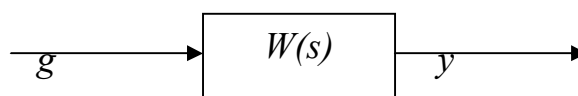
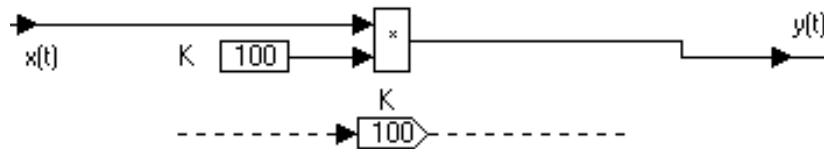


Рис.7.10

Динамические звенья отличаются друг от друга видом соответствующим им операций и могут быть описаны уравнениями в переменных состоя-

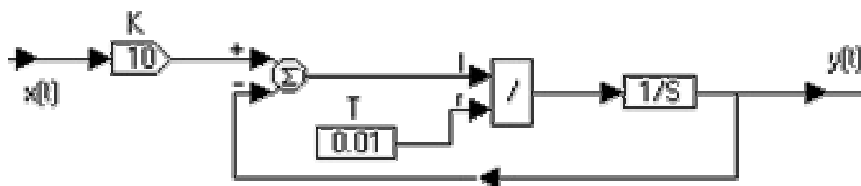
ниях или так называемыми передаточными функциями. Типовые динамические звенья и их схмотехнические решения приведены в таблице.

Тип звена	Передаточная функция	Уравнение вход-выход	Уравнения в переменных состояния
1. Пропорциональное звено	$W(s) = \frac{y(s)}{g(s)} = K$	$y(t) = Kg(t)$	



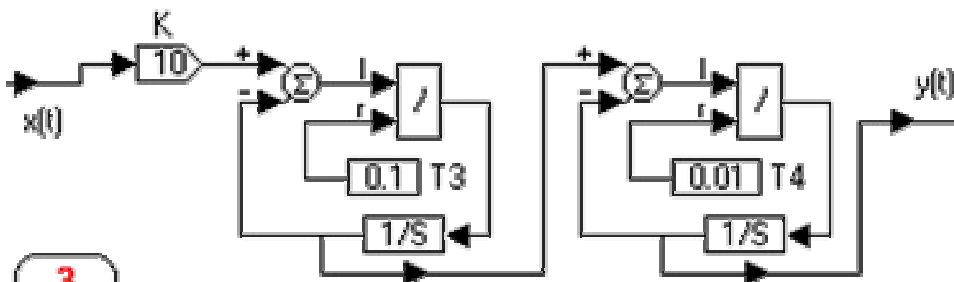
1

2. Инерционное (апериодическое) звено первого порядка	$W(s) = \frac{K}{Ts + 1}$	$T \frac{dy(t)}{dt} + y(t) = Kg(t)$	$\dot{x} = -\frac{1}{T}x + \frac{K}{T}g(t)$ $y = x$
---	---------------------------	-------------------------------------	--



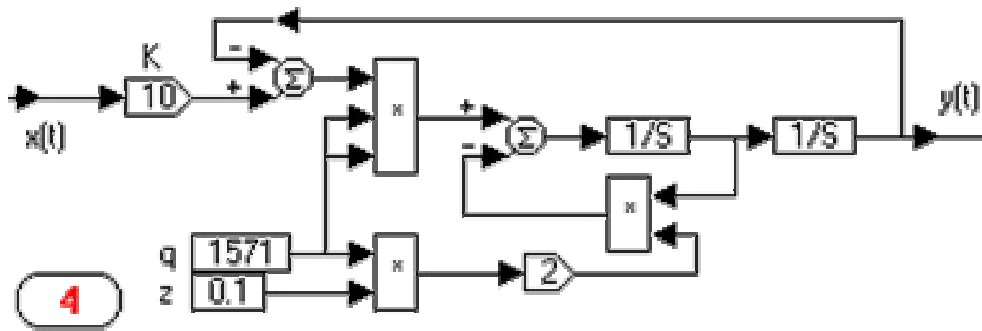
2

3. Апериодическое звено второго порядка	$W(s) = \frac{K_1}{T_1s + 1} \cdot \frac{K_2}{T_2s + 1}$		
---	--	--	--



3

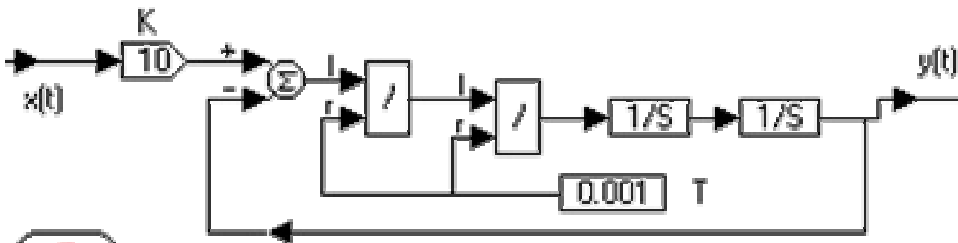
4. Колебательное звено второго порядка	$W(s) = \frac{K}{T^2s^2 + 2\xi Ts + 1}$ , $0 < \xi < 1$	$T^2 \frac{d^2y(t)}{dt^2} + 2\xi T \frac{dy(t)}{dt} + y(t) = Kg(t)$	
--	--	---	--



4

5. Консервативное звено

$$W(s) = \frac{K}{T^2 s^2 + 1}$$



5

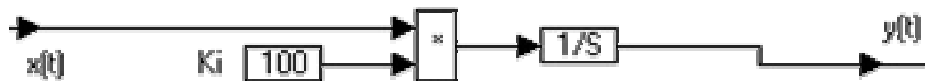
6. Интегрирующее звено (интегратор)

$$W(s) = \frac{K}{Ts}$$

$$T \frac{dy(t)}{dt} = Kx(t)$$

$$\dot{x} = \frac{K}{T} x(t)$$

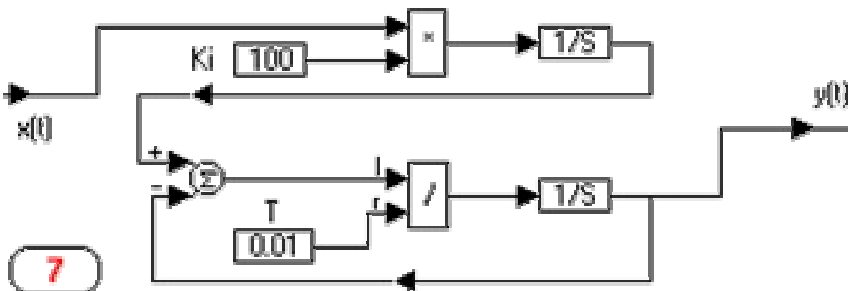
$$y = x$$



6

7. Интегрирующее звено с замедлением

$$W(s) = \frac{K}{(Ts + 1)s}$$

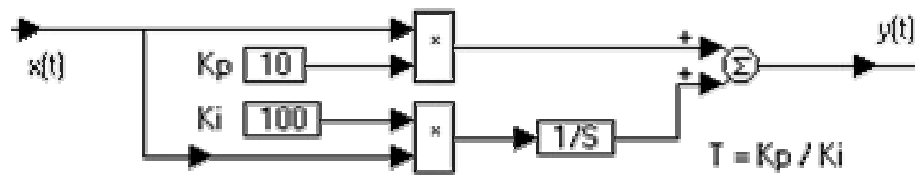


7

8. Изодромное звено

$$W(s) = \frac{K(Ts + 1)}{s}$$

$$\frac{dy(t)}{dt} = KT \frac{dg(t)}{dt} + g(t)$$



8

9. Дифференцирующее звено

$$W(s) = KT_s$$

$$y(t) = KT \frac{dg(t)}{dt}$$



9

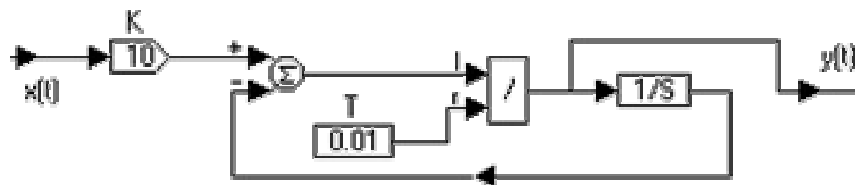
timeStep - шаг моделирования

10. Реальное дифференцирующее звено

$$W(s) = \frac{KT_1 s}{T_2 s + 1}$$

$$T_2 \frac{dy(t)}{dt} + y(t) = KT_1 \frac{dg(t)}{dt} \quad \& \quad -\frac{1}{T_2} x - \frac{KT_1}{T_2^2} g(t)$$

$$y = x + \frac{KT_1}{T_2} g$$



10

11. Интегродифференцирующее звено

$$W(s) = \frac{K(T_1 s + 1)}{T_2 s + 1}$$

$$T_2 \frac{dy(t)}{dt} + y(t) =$$

$$\& \quad -\frac{1}{T_2} x - \frac{KT_1 - T_2}{T_2^2} g(t)$$

$$= KT_1 \frac{dg(t)}{dt} + g(t)$$

$$y = x + \frac{KT_1}{T_2} g$$

В системе Vissim схемотехнические модели дифференциальных уравнений можно описать с помощью типовых динамических звеньев. Примеры решения приведены на сайте [4, 5].

Если система описана передаточными функциями высокого порядка, то можно либо воспользоваться блоком передаточная функция, либо составить одну из двух универсальных блок-схем (рис.7.11):

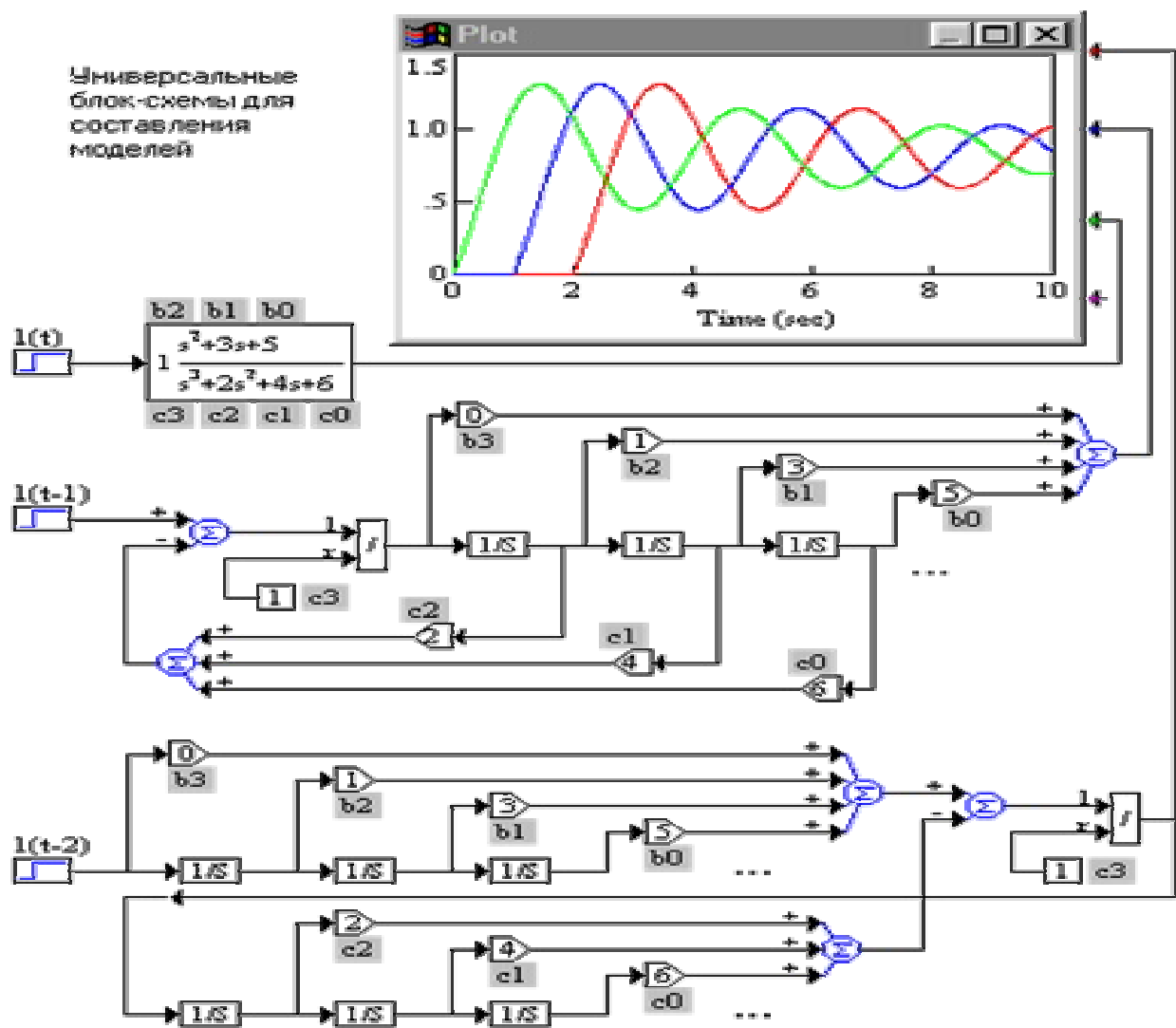


Рис.7.11. Универсальная схема для составления моделей [4]

#### 7.2.4. Модели матричного типа

Построение модели возможно по совокупности обыкновенных дифференциальных уравнений первого порядка, описывающей исследуемую систему типа Коши. Например, исследуемая система описана совокупностью дифференциальных уравнений первого порядка:

$$\left. \begin{aligned} \frac{dx_1}{dt} &= a_{11}x_1 + a_{21}x_2 + a_{31}x_3 + a_1f_1 \\ \frac{dx_2}{dt} &= a_{12}x_1 + a_{22}x_2 + a_{32}x_3 + a_2f_2 \\ \frac{dx_3}{dt} &= a_{13}x_1 + a_{23}x_2 + a_{33}x_3 + a_3f_3 \end{aligned} \right\},$$

тогда блок-схема системы уравнений будет выглядеть так, как показано на рис.7.12:

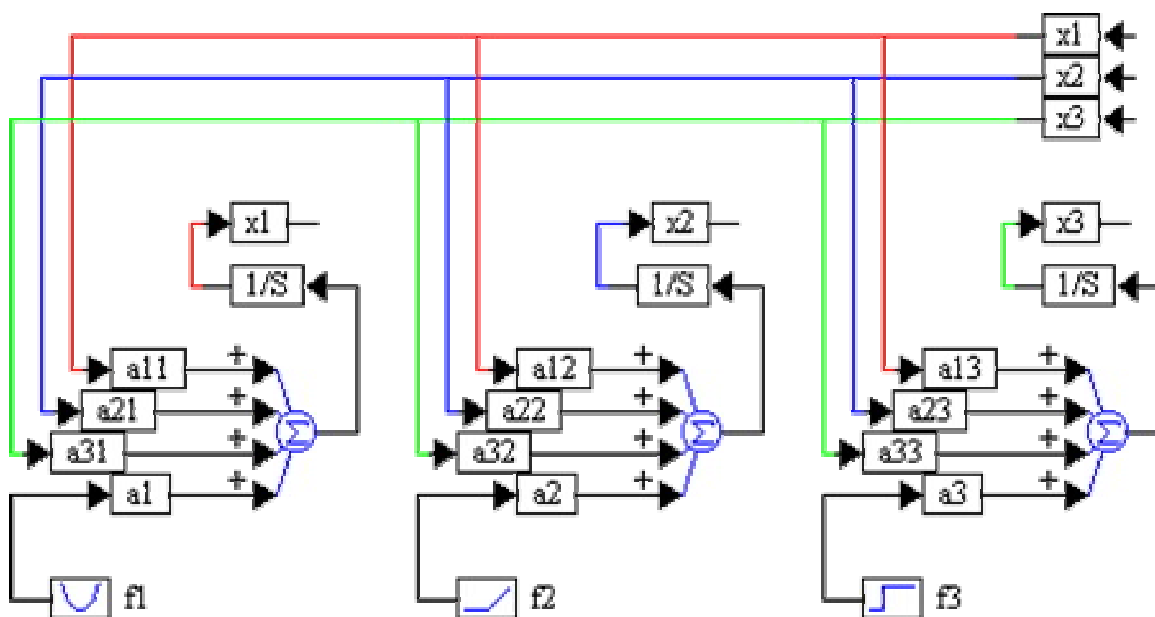


Рис.7.12

где  $x_1, x_2, x_3$  - собственные координаты системы,  $x(t)$  – ошибка системы  $u(t)$  – воздействие на объект, выходная координата -  $y(t)$ ,  $a_{11}, \dots, a_{33}$  - постоянные коэффициенты (если система не является зависимой от параметра) - суммы и произведения постоянных времени  $T_j$ , коэффициентов усиления  $K_n$ ; -  $f_1, f_2, f_3$  - воздействия на систему - сигнал задания  $g(t)$ , помехи  $f_j(t)$ .

### 7.2.5. Моделирование линейной системы дифференциальных уравнений

Кроме описанных методов существуют другие методы и технологии математического моделирования с помощью систем Vissim. Рассмотрим их.

Как известно, линейные системы могут быть описаны в стандартной векторной форме пространства состояний системы:

$$\frac{du}{dt} = Au + Bx,$$

$$y = Cu + Dx,$$

где  $x$  — вектор входных переменных;

$y$  — вектор выходных переменных;

$u$  — вектор переменных состояния (фазовых координат системы);

$A$  — матрица коэффициентов системы;

$B$  — матрица входных коэффициентов (матрица управления);

$C$  — матрица выходных коэффициентов;

$D$  — матрица коэффициентов пропорциональных каналов (матрица компенсации). Максимальная размерность пространства состояний, которую поддерживает VisSim, — 90 фазовых координат.

В системе Vissim, блок **stateSpace** (пространство состояний) предназначен для моделирования системы в пространстве состояний. Он может иметь много входов и выходов. Матрицы коэффициентов пространства со-

стояний определяются в подключаемых к блоку \*.m или \*.mat - файлах, которые могут быть созданы следующими способами.

1. С помощью библиотеки **VisSim/Analyze**, которая имеет функцию линеаризации фрагмента блок-схемы с одним входом и одним выходом. При этом можно получить матрицы A, B, C и D, которые определяются в процессе моделирования. Они могут быть либо выведены на экран, либо сохранены в \*.m-файле.

2. С помощью файла с расширением \*.m (m-файла) может быть создан текстовым редактором, например, редактором Блокнот или редактором m-файлов системы MATLAB.

Наиболее мощными средствами по работе с пространствами состояний обладает матричная система MATLAB. При подготовке m-файла в среде MATLAB надо придерживаться следующих правил:

- каждую новую матрицу надо описывать с новой строки;
- содержимое матриц заключается в квадратные скобки, после закрывающей скобки ставится точка с запятой;
- элементы матриц отделяются пробелами;
- строки матриц отделяются точкой с запятой;
- если матрица большая, то ее строки можно начинать с новой строки;
- в файле не следует использовать команды MATLAB. Особенности применения блока **stateSpace** рассмотрим на примере, представленном на рис. 7.13. Здесь вычисляется реакция непрерывной линейной системы второго порядка на скачок в момент  $t = 0$ .

Окно свойств блока представлено снизу и слева. В левой его части указывается тип файла с данными пространства состояния системы. Отсутствие птички у опции **Discrete** означает, что анализируется непрерывная система. В правой части окна показывается имя файла и указывается число переменных, входов и выходов. Кнопка **Select File...** открывает стандартное окно Windows для загрузки файла. А кнопка **Browse Data...** открывает окно текстового редактора Блокнот с текстом текущего (загруженного) файла, составленного по описанным выше правилам. Для моделируемой системы это окно показано снизу и справа в окне модели.

В окне свойств имеется также возможность задания через пробелы начальных условий (значений фазовых координат  $u$ ), которые будут установлены в матрице интеграторов [I/S] на первом шаге моделирования. Самое правое значение будет установлено на нижнем интеграторе в матрице. Неопределенные начальные значения принимаются равными 0.

Пример моделирования дискретной линейной системы второго порядка по заданному пространству ее состояний представлен на рис. 7.13.

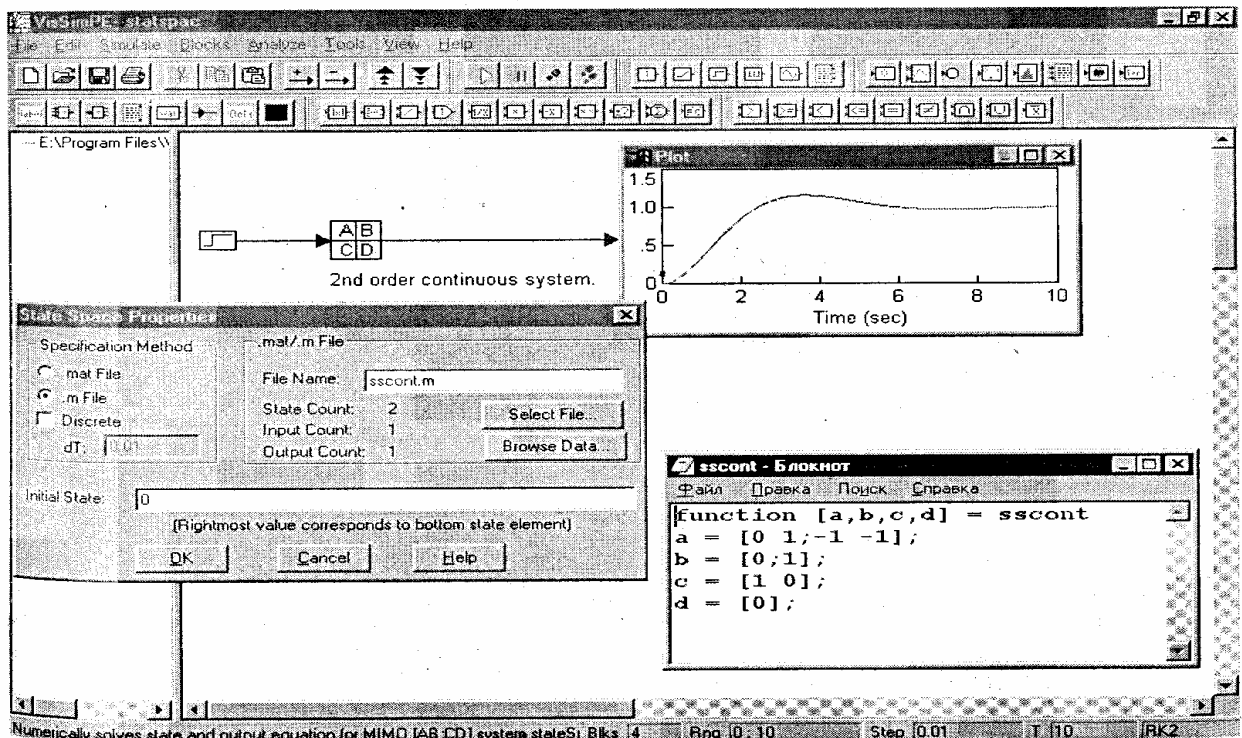


Рис. 7.13. Пример моделирования непрерывной линейной системы второго порядка

Для перехода к анализу дискретных систем в окне свойств блока **StateSpace** надо выставить птичку в опции **Diskrete** и задать шаг во времени  $\Delta T$ .

Рассмотрим алгоритм моделирования.

1. Выбираем блок **Step**. Затем щелкаем мышью **Blocks**, затем **linear system**, **StateSpace**. После этого щелкаем мышью около блока **Step**, появляется схема как на рис. 7.14.

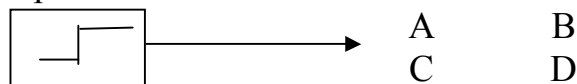


Рис. 7.14. Схема решения систем линейных уравнений

2. После этого нажимаем **Blocks**, **Signal Consumer**, **Plot**. Появляется графическая область. После нажатия зеленой кнопки “Go” (зеленый треугольник) появляется график как на рисунке 7.12.

3. Для анализа дискретной системы щелкаем правой кнопкой мыши на блоке A, B, C, D (появляется окно “**State Space Properties**”).

В нем в строке «**Initial State**» задаем начальное условие (например, 1), затем нажимаем “OK”, “Go”. После этого получится диаграмма, как на рисунке 7.15.



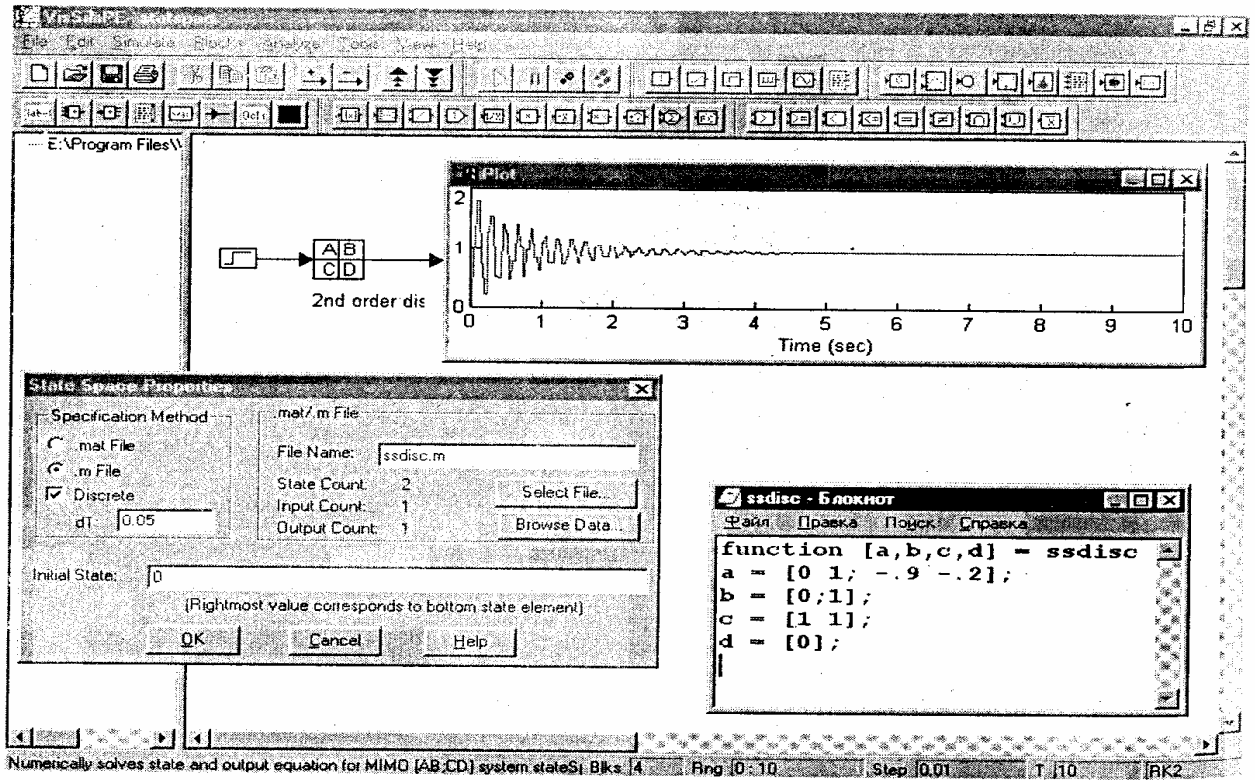


Рис. 7.15. Пример моделирования дискретной линейной системы второго порядка

## ГЛАВА 8. СХЕМОТЕХНИЧЕСКОЕ МОДЕЛИРОВАНИЕ В СИСТЕМЕ SIMULINK

В наши дни компьютерная математика получила должную известность и интенсивно развивается как передовое научное направление на стыке математики и информатики. Среди ряда современных СКМ выгодно выделяется математическая матричная лаборатория MATLAB. Она имеет мощные средства математически ориентированного программирования, диалога, графики и комплексной визуализации.

Популярности системы MATLAB способствует ее мощное расширение Simulink, предоставляющее пользователю удобные и простые средства, в том числе визуального объектно-ориентированного программирования для моделирования линейных и нелинейных динамических систем, а также множество других пакетов расширения системы. Некоторые из них являются наиболее доступными и эффективными инструментальными средствами для развития и применения новейших направлений науки и техники, таких, как средства анализа и обработки сигналов и изображений (включая новейшую вейвлет-технологию), методы и инструменты анализа, идентификации и моделирования систем, нечеткая логика, нейронные сети и т. д.

Simulink поступает к пользователям с более 100 встроенными блоками, в состав которых входят наиболее необходимые функции моделирования. Блоки сгруппированы в библиотеки в соответствии с их назначением: источники сигнала, приемники, дискретные, непрерывные, нелинейные, математика, функции и таблицы, сигналы и системы. В дополнение к обширному набору встроенных блоков Simulink имеет расширяемую библиотеку блоков благодаря функции создания пользовательских блоков и библиотек. Вы можете настраивать не только функциональность встроенных и пользовательских блоков, но также пользовательский интерфейс, используя значки и диалоги. Например, вы можете создать блоки для моделирования поведения специальных механических, электрических и программных компонентов, как, например, моторы, преобразователи, сервоклапаны, источники питания, энергетические установки, фильтры, шины, модемы, приемники или другие динамические компоненты. Однажды созданные пользовательские блоки могут быть сохранены в библиотеке блоков для использования в будущем. Любые пользовательские блоки или библиотеки блоков могут быть легально распространены в рабочих группах, переданы поставщикам и заказчикам как с исходным кодом, так и без него.

Новые и улучшенные возможности блоков:

- наряду с существовавшей ранее поддержкой скалярных и векторных сигналов обеспечена поддержка матричных сигналов многими блоками Simulink;
- блоки Product, Multiplication, Gain и Math Function теперь поддерживают матричные операции на матричных сигналах;
- Mix и Demux блоки теперь поддерживают мультиплексирование матричных сигналов;

- новый блок Reshape изменяет размер матрицы своего входного сигнала;
- блок Probe теперь по умолчанию выводит размер матрицы сигнала, подаваемого на вход;
- новый блок Bitwise Logical Operator (логические операции по битам) накладывает маску, инвертирует или производит логические операции с отдельными битами целочисленного сигнала без знака;
- четыре новых блока Look-Up Table (просмотра таблиц);
- новый Polynomial блок выводит полиномиальную функцию от входного сигнала.

## 8.1. Работа с основным пакетом расширения Simulink

**Доступ к пакету расширения Simulink.** Пакет блочного ситуационного моделирования Simulink является неразрывной частью системы MATLAB + Simulink. Он автоматически интегрируется с базовой системой MATLAB. Внешне это выражается появлением кнопки Simulink в панели инструментов (перед кнопкой ?) системы MATLAB. При нажатии этой кнопки открывается окно интегрированного браузера библиотек, показанное в левой части рис. 8.1. В окне браузера содержится дерево компонентов библиотек Simulink. Для просмотра того или иного раздела библиотеки достаточно выделить его мышью — в правой части окна Simulink Browser Library появится набор пиктограмм компонентов активного раздела библиотеки. На рис. 8.1 показан основной раздел библиотеки Simulink. Число разделов библиотеки различно у разных версий Simulink — у Simulink 5 оно возросло с 8 до 10. Новыми являются подразделы верификации моделей Model Verification и построения subsystems Subsystems. В остальном окна моделей версий Simulink 4 и 5 идентичны.

С помощью меню браузера или кнопок его панели инструментов можно открыть окно для создания новой модели или загрузить существующую. Работа с Simulink происходит на фоне открытого окна системы MATLAB, в котором нередко можно наблюдать за выполняемыми операциями — если их вывод предусмотрен программой моделирования.

**Запуск моделей Simulink из среды MATLAB.** Обычно Simulink запускается соответствующей кнопкой из панели инструментов окна MATLAB, что было описано выше, после чего все последующие действия выполняются в среде MATLAB+Simulink. Можно также запустить Simulink, исполнив в командной строке MATLAB команду

» simulink

Для вывода полного перечня команд Simulink надо выполнить команду

» help simulink

Дополнительную информацию можно получить, используя команды help blocks и help simdemos. Первая команда дает информацию об основных библиотеках Simulink и примерах применения S - функций, а вторая выводит список демонстрационных примеров. Запуск этих примеров дает возможность практически познакомиться с возможностями пакета Simulink и оце-

нить степень сложности систем и устройств, которые могут моделироваться с помощью этого пакета.

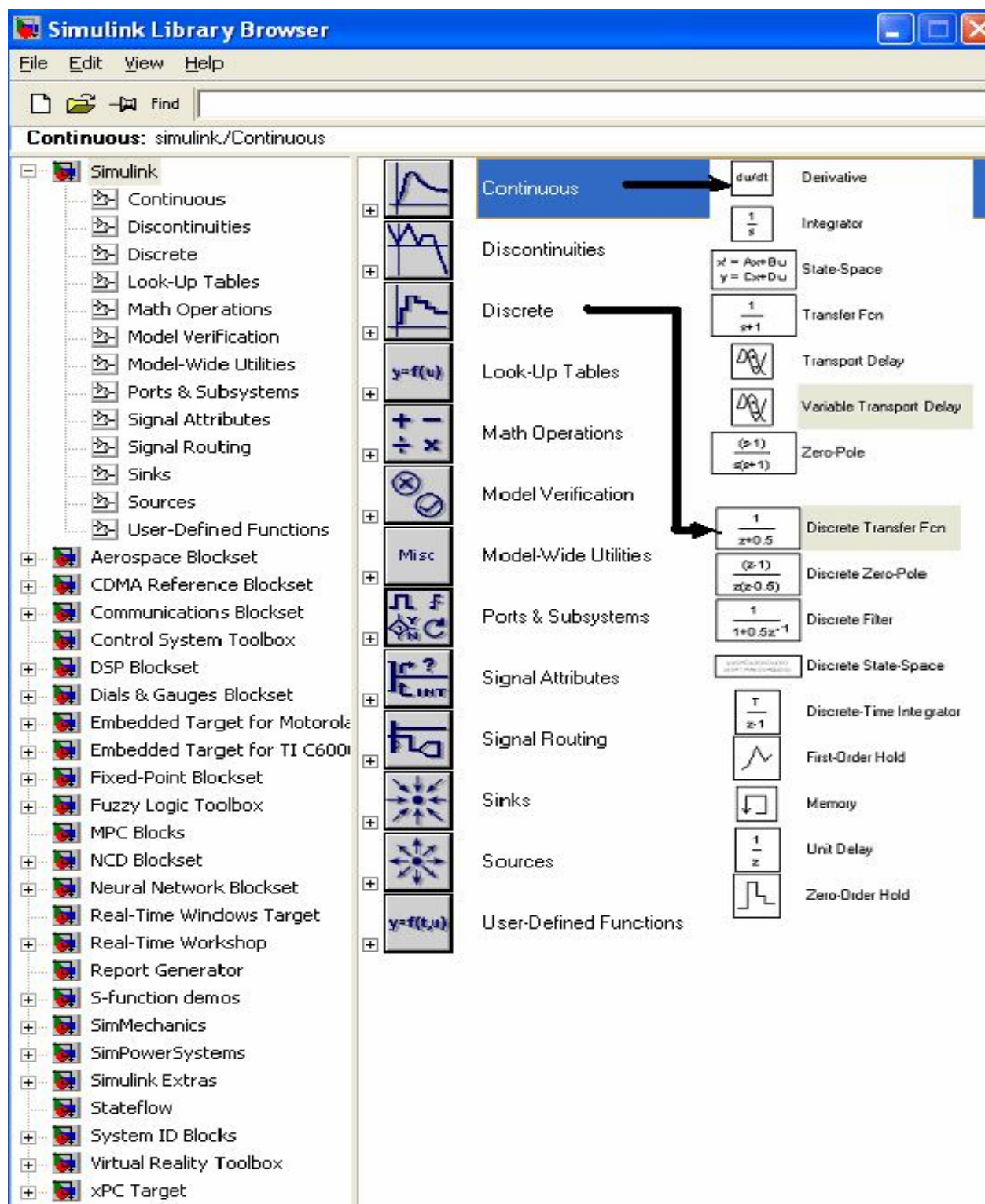


Рис. 8.1. Окно браузера библиотек Simulink

**Особенности интерфейса Simulink.** Интерфейс Simulink полностью соответствует стилю интерфейса типичных приложений Windows"95/98/NT/2000 (для Simulink 5 возможна работа и в Windows XP), в том числе интерфейсу системы MATLAB. В то же время он концептуально

строг, чтобы не досаждал пользователю многочисленными «излишествами» стандартного интерфейса Windows"95/98/NT/2000. Меню системы содержит следующие пункты:

- File – работа с файлами моделей и библиотек (их создание, сохранение, считывание и печать);
- Edit – операции редактирования, работа с буфером обмена и создание подсистем
- View – управление отображением панели инструментов и строки состояния;
- Simulation– управление процессом моделирования (старт, пауза, вывод окна настройки параметров моделирования);
- Format – операции форматирования модели (смена шрифтов, редактирование надписей, повороты блоков, использование тени от блоков операции с цветами линий блоков, их фоном и общим фоном;
- Tools – управление видом анализа (в линейной области и в режиме реального времени RTW).

Первые три пункта главного меню содержат общепринятые для Windows-приложений команды и операции, поэтому мы не будем их обсуждать. Остальные будут рассмотрены по мере знакомства с системой Simulink.

Как уже отмечалось, вместе с рабочим окном Simulink выводится окно с перечнем разделов основной библиотеки компонентов. Это окно – важная часть интерфейса Simulink. Оно открывает доступ к другим пакетам компонентов (Blocksets&Toolboxes) и примерам их применения (Demos). Это дает пользователю возможность постепенно знакомиться с новыми областями применения Simulink.

**Поиск и загрузка модели.** Даже незнакомый с Simulink пользователь может быстро оценить возможности этого пакета, воспользовавшись интересными и поучительными примерами, входящими в поставку Simulink. Они находятся в папке MATLAB/ TOOLBOX/ SIMULINK/ SIMDEMOS. Попутно отметим, что в папке MATLAB/ SIMULINK располагаются служебные файлы.

Для загрузки примеров используется (как обычно) команда **Open** в меню **File** браузера библиотеки Simulink. В обычном окне загрузки файлов с названием **Open** надо войти в указанную папку и выбрать подходящий пример.

При открытии нужного примера (в данном случае - файл lorenzs) появляется окно редактирования графической модели устройства (рис.8.2).

Как можно заметить из примера, графическая модель содержит ряд блоков. Каждый блок имеет наглядное общепринятое обозначение в виде прямоугольника, треугольника и т. д. Блоки имеют входы и выходы и описываются различными математическими зависимостями. Блоки соединяются друг с другом линиями со стрелками, причем стрелка указывает направление от выходов одних блоков ко входам других. Имеются также текстовые комментарии и средства для вывода подсказок и открытия окон справочной системы.

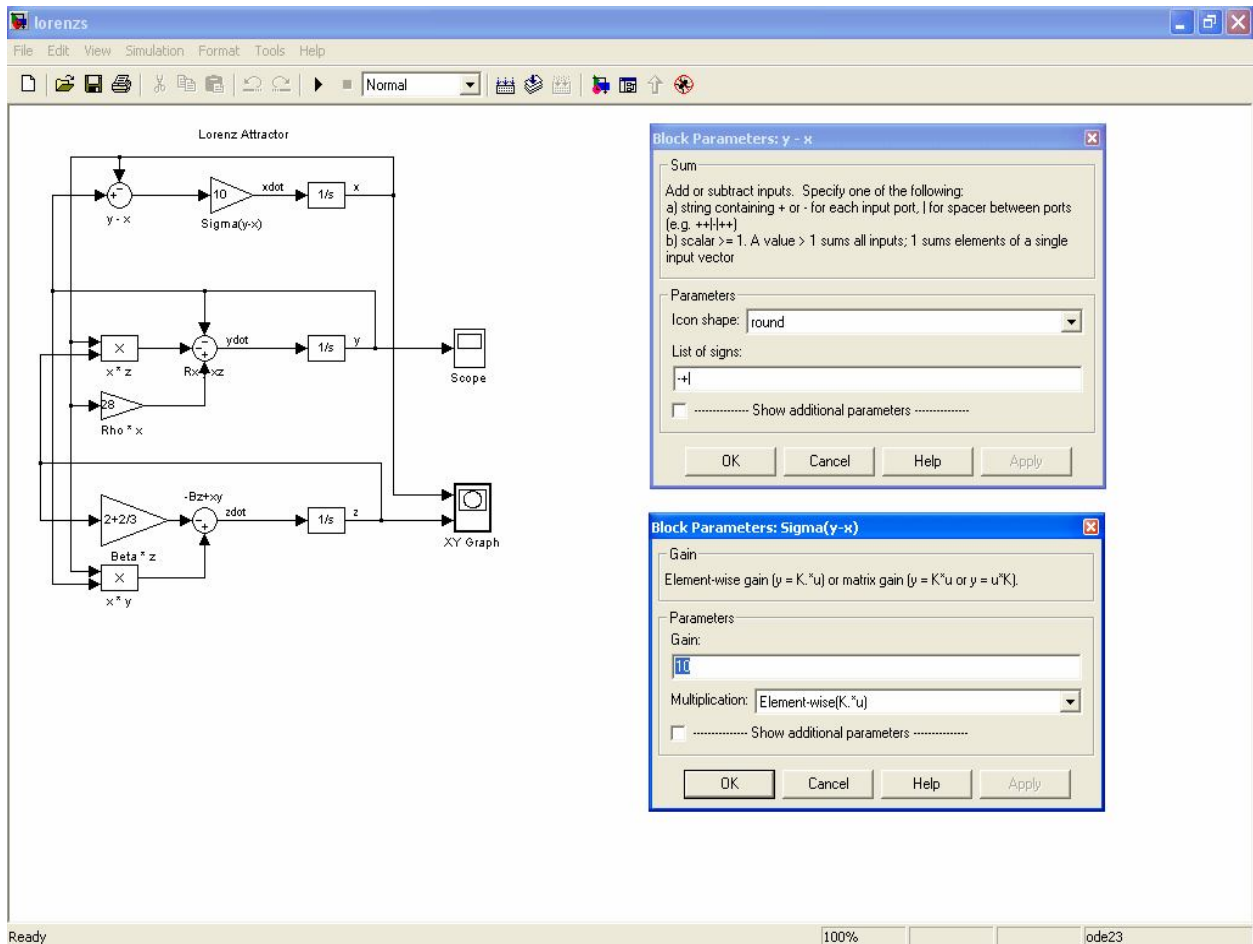


Рис. 8.2. Модель аттрактора Лоренца

**Установка параметров компонентов модели.** Для пользователя, имеющего хотя бы начальное представление о моделировании, вполне ясно, что любой компонент модели имеет какие-то параметры или хотя бы один параметр. Из самой графической модели не видно, какими параметрами обладает тот или иной компонент. Каким же образом узнать, что за параметры компонентов установлены в той или иной модели?

Для того чтобы вызвать окно модификации параметров компонента, нужно навести курсор мыши на изображение компонента и дважды щелкнуть левой кнопкой мыши. На рис.8.2 справа от модели показаны два таких окна для компонентов с именами  $y-x$  и  $\text{Sigma}(y-x)$ .

Кроме того, можно просто просмотреть параметры компонента. Для этого нужно, указав курсором мыши интересующий вас компонент, задержать курсор на 2-3 секунды. Под ним появится подсказка с указанием параметров этого компонента — на рис.8.2 показана подсказка для компонента  $1/s$ .

**Установка параметров моделирования.** Прежде чем запустить загруженную модель, стоит ознакомиться с установкой общих параметров моделирования. Для этого выполним команду Simulation Parameters... в меню Simulation окна Simulink. Появится окно установки параметров моделирования, показанное на рис.8.3.

Это окно имеет ряд вкладок с довольно большим числом параметров. Мы постепенно рассмотрим их все. Но пока остановимся на вкладке, открытой по умолчанию, — Solver (Решатель). Она позволяет установить параметры решающего устройства системы моделирования Simulink.

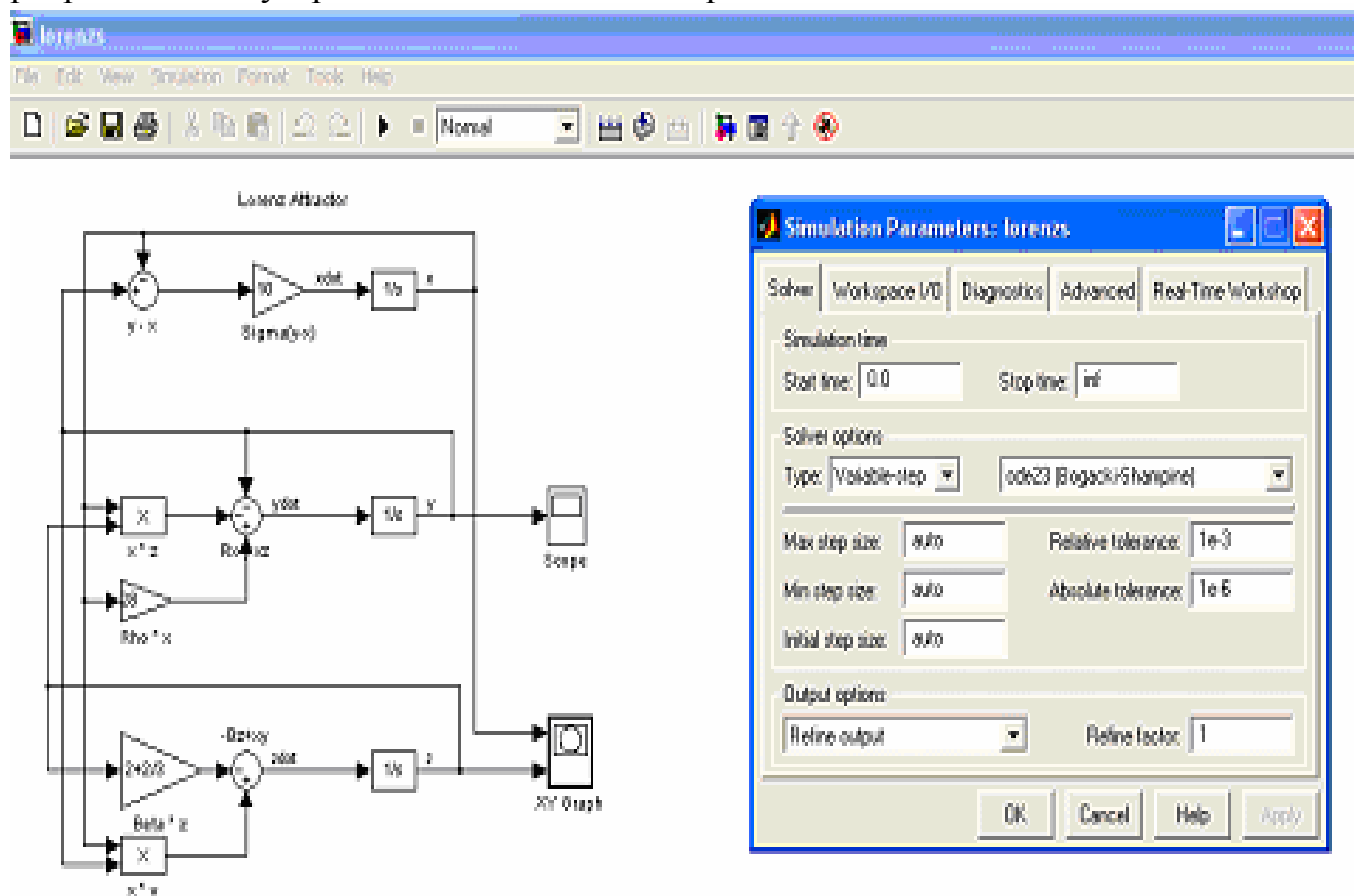


Рис. 8.3. Установка параметров модели аттрактора Лоренца

Важнейшим параметром решателя является время моделирования — Simulation time. Оно задается начальным временем Start time (обычно 0) и конечным временем Stop time (в нашем случае бесконечность inf). Равенство Stop time бесконечности означает, что моделирование будет происходить бесконечно долго, пока мы не прервем его. Однако в этом случае трудно получить различимые осциллограммы работы устройства, поэтому рекомендуется задавать конечные значения Stop time.

Стоит отметить, что время моделирования — величина довольно условная. Не следует думать, что Stop time=50 означает моделирование в течение 50 секунд. Точного соответствия между временем моделирования в секундах и заданным значением нет. Реальное время моделирования сильно зависит от быстродействия компьютера, на котором выполняется моделирова-



ние.

Первостепенное значение имеют две опции решателя в поле Solver options: тип решения и метод решения. Возможно два типа решения:

- Variable-step solvers — решение с переменным шагом;
- Fixed-step solvers — решение с фиксированным шагом.

Как правило, лучшие результаты дает решение с переменным шагом (обычно по времени, но не всегда). В этом случае шаг автоматически уменьшается, если скорость изменения результатов в процессе решения возрастает. И, напротив, если результаты меняются слабо, шаг решения автоматически увеличивается. Это исключает (опять-таки, как правило) расхождение решения, которое нередко случается при фиксированном шаге.

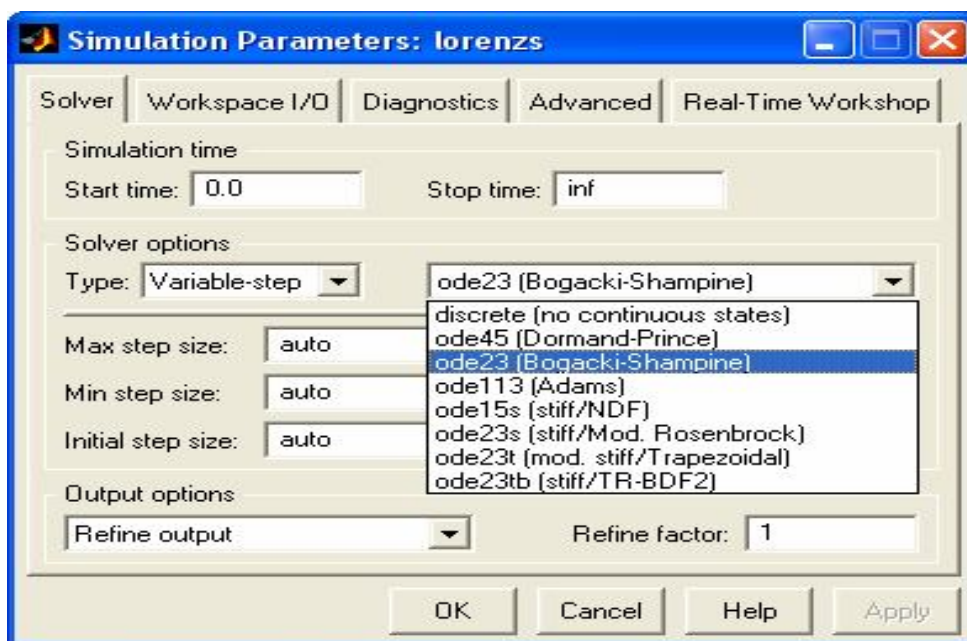


Рис. 8.4. Установка метода моделирования

Метод с фиксированным шагом стоит применять только тогда, когда фиксированный шаг обусловлен спецификой решения задачи, например, если ее цель заключается в получении таблицы результатов с фиксированным шагом. Этот метод дает неплохие результаты, если поведение системы описывается почти монотонными функциями. В противном случае шаг времени придется сильно уменьшать для описания наиболее быстрых участков изменения результатов моделирования, что ведет за собой значительное возрастание времени моделирования.

Вторая из указанных опций — выбор метода моделирования. Рисунок рис.8.4 показывает окно установки параметров моделирования отдельно с открытым меню выбора метода моделирования.

Для решения дифференциальных уравнений можно выбрать следующие методы: discrete (дискретный), ode45, ode23 (три варианта, включая метод Розенброка), rk45 (метод Дорманда — Принса), ode 113 (метод Адамса) и ode15s. Методы, в наименовании которых имеется слово stiff, служат для решения жестких систем дифференциальных уравнений.



Следующие три параметра — значения опции auto — обычно задаются автоматически, но в особых случаях их можно ввести явно:

- Max step size — максимальный шаг интегрирования системы однородных дифференциальных уравнений;
- Min step size — минимальный шаг интегрирования;
- Initial step size — начальный шаг интегрирования.

Важен и такой параметр моделирования, как точность интегрирования:

- Relative tolerance — относительная погрешность интегрирования;
- Absolute tolerance — абсолютная погрешность интегрирования.

По умолчанию они заданы, соответственно, равными  $10^{-3}$  и  $10^{-7}$ . Если, например, графики результатов моделирования выглядят составленными явно из отдельных фрагментов, это указывает на необходимость уменьшения указанных значений погрешности. Однако слишком малые погрешности могут вызвать значительное увеличение времени вычислений. Не оптимально выбранные значения погрешности (как очень малые, так и очень большие) могут вызвать неустойчивость и даже «зацикливание» процесса моделирования.

В конце панели инструментов Simulink находятся две важные кнопки управления моделированием. Одна из них, в виде черного треугольника (Start/Pause Simulation), запускает или приостанавливает начатый процесс моделирования, а другая, в виде черного квадрата (Stop), останавливает его. Все, что нужно для запуска моделирования — это нажать кнопку с изображением треугольника. Вместо кнопок можно использовать команды Start и Pause в меню Simulation окна модели.

Для моделирования поведения динамических систем, к которым относятся экипажи подвижного состава, используются ЭВМ. Существует большое количество алгоритмических языков, на которых может быть выполнено решение задачи. Выбор того или иного языка программирования зависит от многих условий. Часто решающую роль оказывает удобство программирования, наличие проверенных математических методов, легкость представления результатов моделирования. Такими особенностями обладает пакет MATLAB, содержащий в своем составе инструмент визуального моделирования SIMULINK.

SIMULINK сочетает в себе наглядность аналоговых машин и точность цифровых вычислительных машин. SIMULINK обеспечивает пользователю доступ ко всем возможностям пакета MATLAB, в том числе к большой библиотеке численных методов.

Подготовка задачи для моделирования в SIMULINK проводится в следующей последовательности:

- выбор расчетной схемы;
- составление системы уравнений, описывающих исследуемый процесс;
- приведение системы к виду, удобному для решения (разрешение относительно старших производных);
- определение начальных условий;
- составление структурной схемы;

- моделирование возмущающих функций;
- определение исходных данных;
- составление модели в среде SIMULINK;
- включение средств визуализации;
- тестирование;
- решение;
- анализ результатов;
- отчет.

## 8.2. Математическое моделирование вынужденных колебаний двухмассовой системы

**Математическая модель.** Для исследования влияния основных параметров экипажа на вертикальные колебания используют упрощенную модель с двумя степенями свободы, в которой две массы связаны упругими и диссипативными связями (рис.8.5). Такая модель описывает вертикальные колебания рельсовых экипажей с двухъярусным подвешиванием: магистральных локомотивов (электровозов и тепловозов) и пассажирских вагонов.

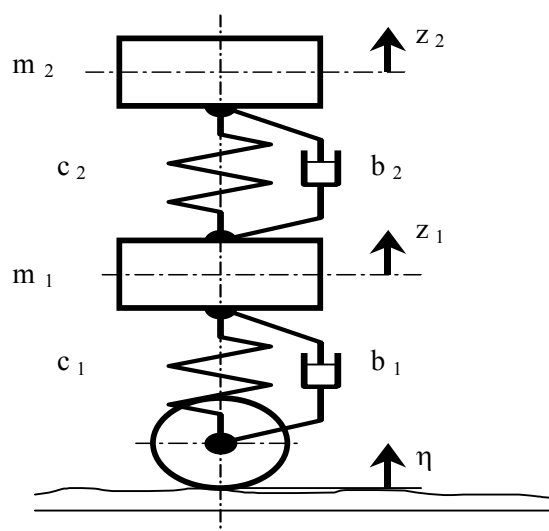


Рис.8.5. Расчетная схема

Уравнения движения рассматриваемой системы при наличии возмущения со стороны пути описывается следующими дифференциальными уравнениями:

$$m_1 \cdot \ddot{z}_1 + b_1 \cdot \dot{z}_1 + b_2 \cdot (\dot{z}_1 - \dot{z}_2) + c_1 \cdot z_1 + c_2 \cdot (z_1 - z_2) = b_1 \cdot \dot{\eta} + c_1 \cdot \eta; \quad (8.1)$$

$$m_2 \cdot \ddot{z}_2 + b_2 \cdot (\dot{z}_2 - \dot{z}_1) + c_2 \cdot (z_2 - z_1) = 0.$$

В уравнениях (8.1) введены следующие обозначения:

$m_1$  – обрессоренная масса тележки;

$m_2$  – масса кузова, приведенная к одной тележке;

$c_1, b_1$  – жесткость и демпфирование в первом ярусе подвешивания;

$c_2, b_2$  – жесткость и демпфирование во втором ярусе подвешивания;

$\eta(t)$  – возмущение со стороны пути;

$z_i, \dot{z}_i, \ddot{z}_i$  – обобщенные координаты и их производные по времени:  
Преобразуем уравнения движения к виду:

$$\ddot{z}_1 = -\frac{1}{m_1} (b_1 \cdot \dot{z}_1 + b_2 \cdot (\dot{z}_1 - \dot{z}_2) + c_1 \cdot z_1 + c_2 \cdot (z_1 - z_2) - b_1 \cdot \dot{\eta} - c_1 \cdot \eta) \quad (8.2)$$

$$\ddot{z}_2 = -\frac{1}{m_2} (b_2 \cdot (\dot{z}_2 - \dot{z}_1) + c_2 \cdot (z_2 - z_1))$$

В качестве возмущения  $\eta(t)$  используем неровность проф. Н.Н.Кудрявцева. Неровность хорошо описывает изменение прогиба вдоль рельсового звена. Модель неровности представляет собой сумму полуволны синусоиды частотой  $\omega$  и трех полуволн синусоиды частотой  $3\omega$ , уложенные на длине рельсового звена  $L$ . Амплитуды неровностей  $A_1, A_2$  выбираются в зависимости от типа и состояния пути.

$$\eta(t) = \left| A_1 \cdot \sin(\omega \cdot t) + A_2 \cdot \sin(3 \cdot \omega \cdot t) \right|, \quad (8.3)$$

где  $\omega = \frac{\pi}{L} \cdot V$  – частота возмущения,  $V$  – скорость движения.

**Построение модели в системе Simulink.** При запуске SIMULINK открываются два окна:

- пустое рабочее окно – заготовка для создания новой модели (*untitled*);
- окно библиотеки SIMULINK, содержащей набор основных разделов (*Library: simulink*).

Далее можно раскрыть необходимые разделы библиотеки SIMULINK (рис. 8.6). Поскольку моделируемая система довольно проста, покажем реализацию модели одного уровня (без вложенных подсистем).

Процесс построения модели в системе SIMULINK представляет последовательность выбора необходимых блоков из соответствующих библиотек и соединение их связями.

Обычно для моделирования динамической системы используют уравнения движения в виде (8.2).

Построение модели каждого уравнения начинаем с сумматора, имеющего столько входов, сколько членов содержит правая часть уравнение. Для первого уравнения это сумматор *Sum1* (рис. 8.6). Входы сумматора могут иметь как положительное значение, так и отрицательное. Используемые блоки рекомендуется именовать для облегчения последующей проверки и анализа.

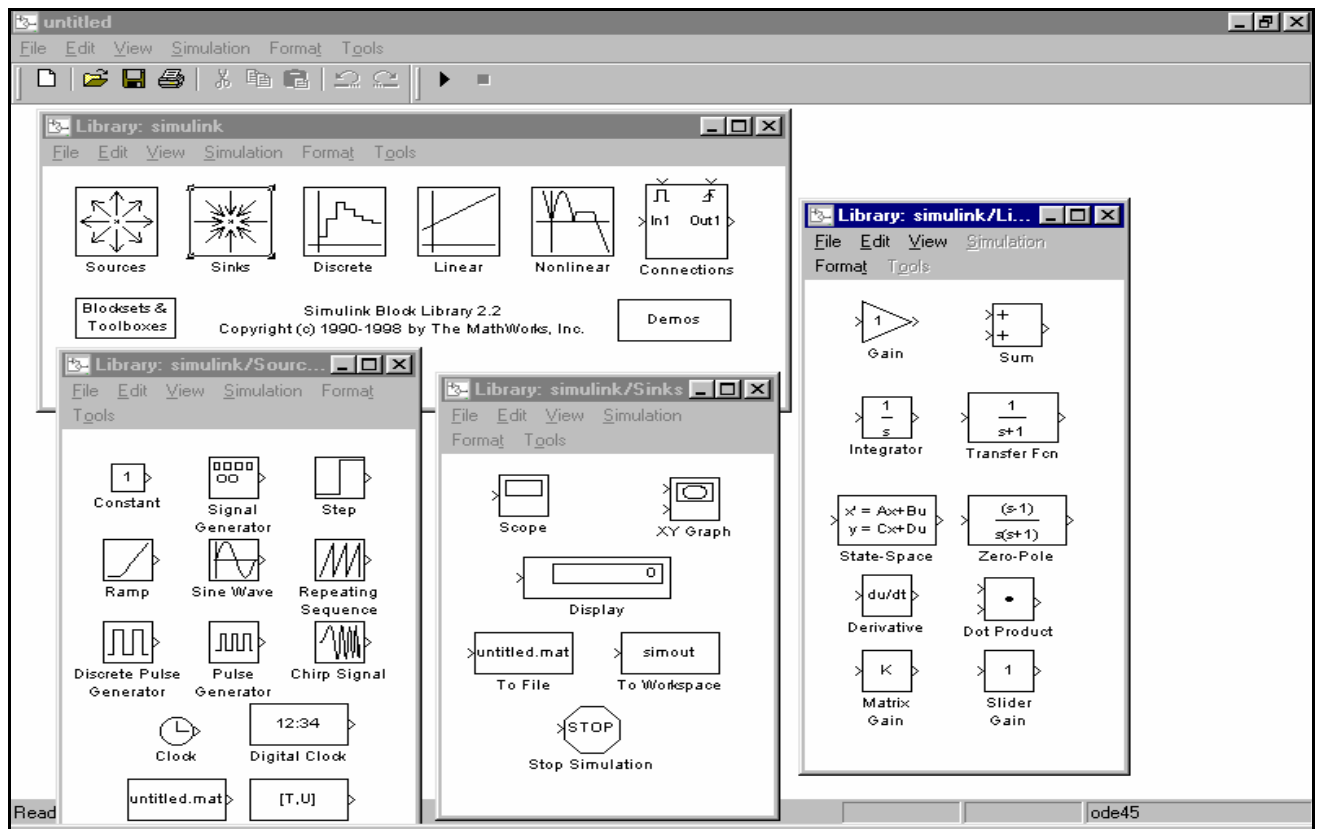


Рис. 8.6. Пример начала работы в SIMULINK

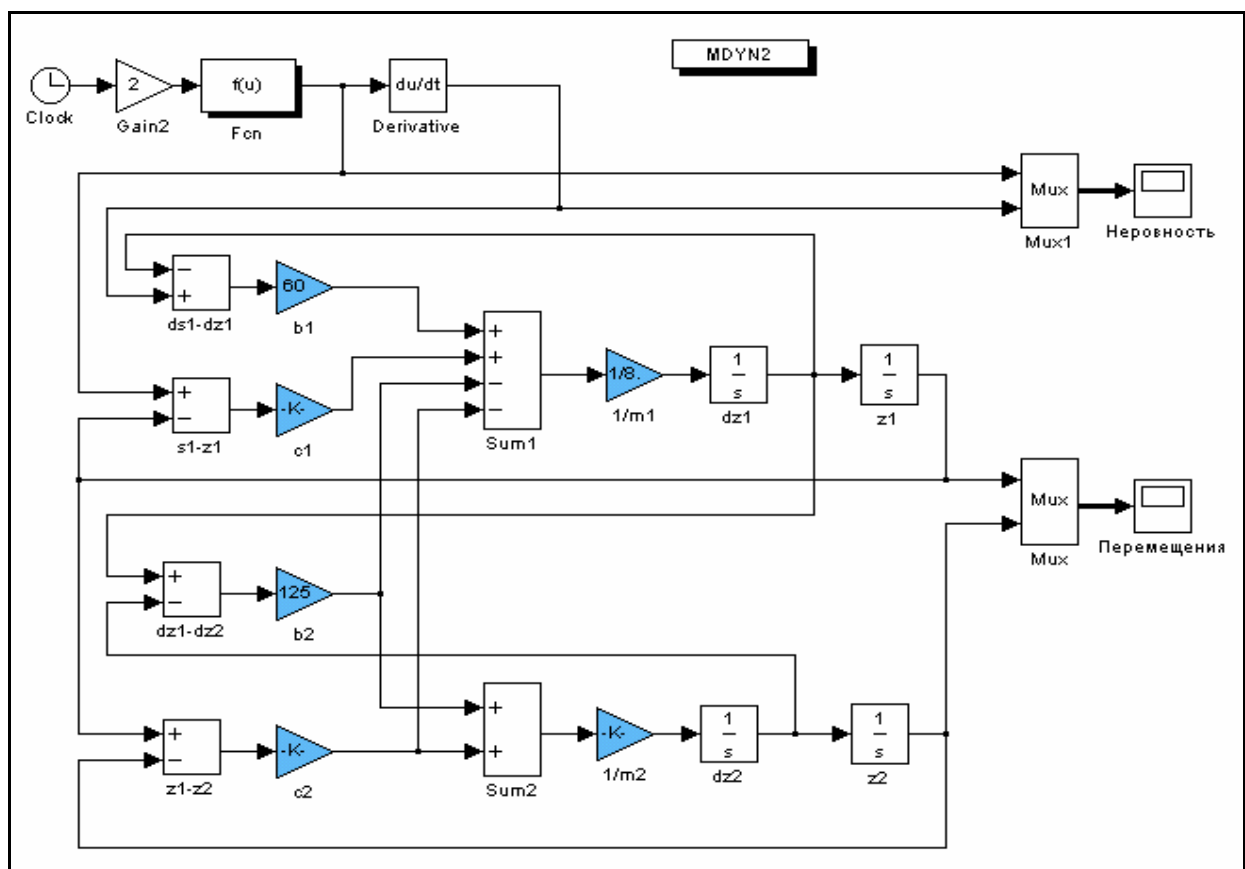


Рис. 8.7. Блок-схема модели

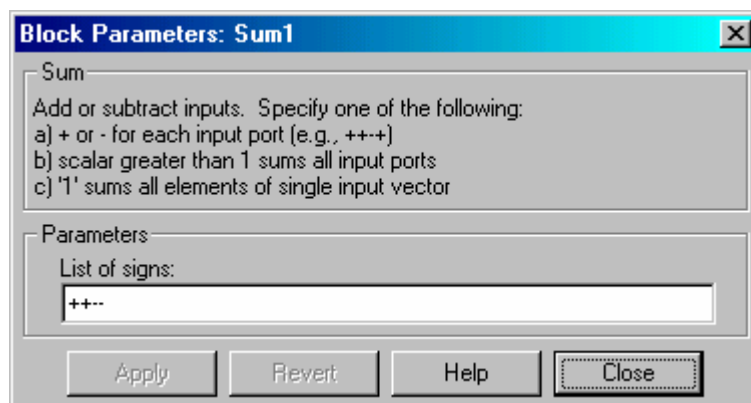


Рис. 8.8. Настройка сумматора

К выходу сумматора подключаем линейный преобразователь (множитель  $1/m1$ ), на выходе которого получаем значение второй производной. Далее включаем последовательно два интегратора ( $dz1, z1$ ), на выходе которых получаем значения первой производной и самой переменной.

В нашем случае имеем два уравнения, соответственно процесс повторяем для второго уравнения.

Далее устанавливаем связи между входами и выходами соответствующих блоков, применяя, где необходимо, дополнительные линейные преобразователи и сумматоры.

Для моделирования возмущения используем функциональный блок  $Fcn$ , в котором в аналитическом виде запишем зависимость перемещения неровности от времени. Блок имеет вход, на который через линейный преобразователь подается от "таймера" независимая переменная – время.

Для вычисления производной перемещения неровности используется блок дифференцирования.

После установки всех связей определяем необходимые коэффициенты в используемых преобразователях.

Для моделирования использованы следующие исходные данные:

амплитуда первой гармоники неровности	$A_1 = 0.005$	м;
амплитуда второй гармоники неровности	$A_2 = 0.002$	м;
длина рельсового звена	$L = 25$	м;
масса первого тела	$m_1 = 8.82$	т;
жесткость первого яруса	$c_1 = 7000$	кН/м;
демпфирования первого яруса	$b_1 = 60$	кН с/м;
масса второго тела	$m_2 = 25.8$	т;
жесткость второго яруса	$c_2 = 2600$	кН/м;
демпфирования второго яруса	$b_2 = 125$	кН с/м.

Зададим значения коэффициентов в соответствующих блоках (рис. 8.9).

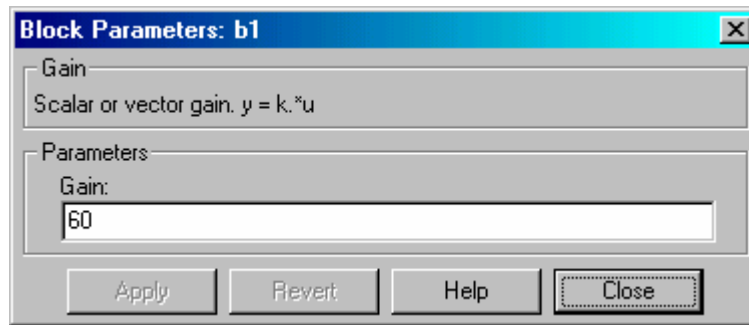


Рис. 8.9. Задание коэффициента линейного преобразователя

Настроим интеграторы, зададим нулевые начальные условия (рис. 8.10).

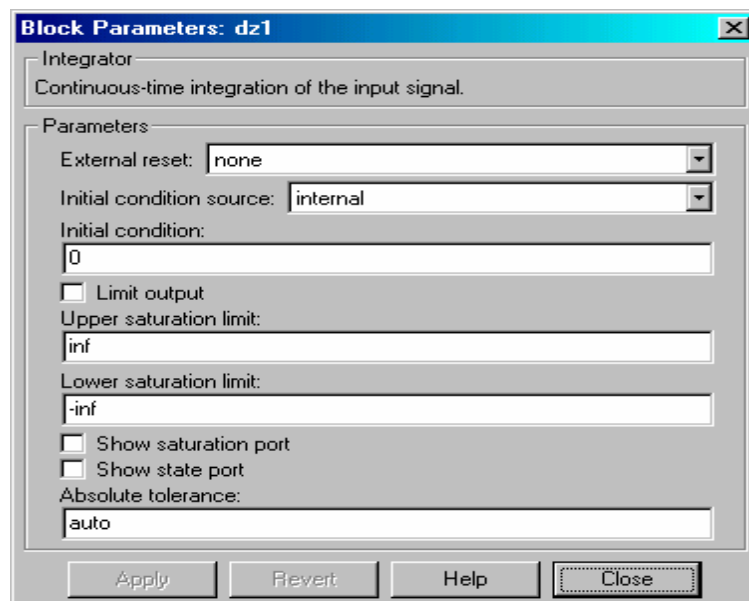


Рис. 8.10. Панель настройки интегратора

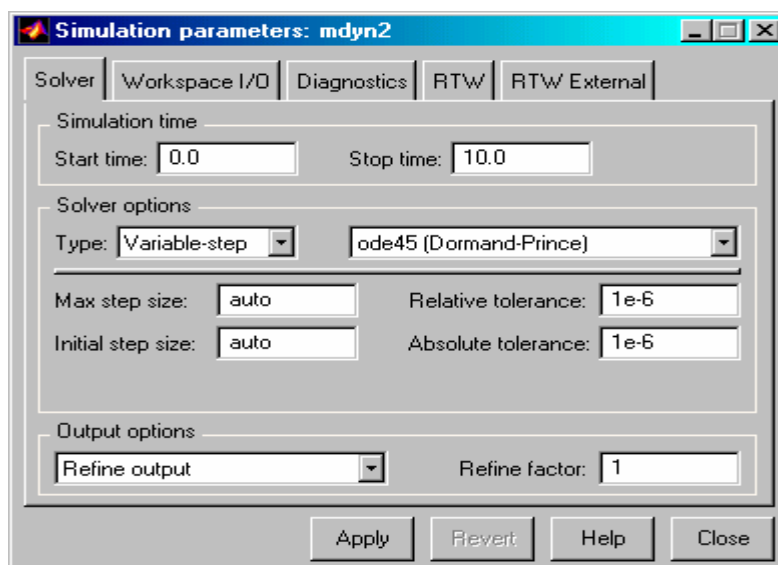


Рис. 8.11. Параметры интегрирования

Установим параметры интегрирования через меню (рис. 8.11).

Начало интегрирования – 0, конец интегрирования – 10. Используем метод *ODE45* с переменным шагом интегрирования.

В соответствии с формулой (3) запишем функцию возмущения в блоке *Fcn* (рис. 8.12).

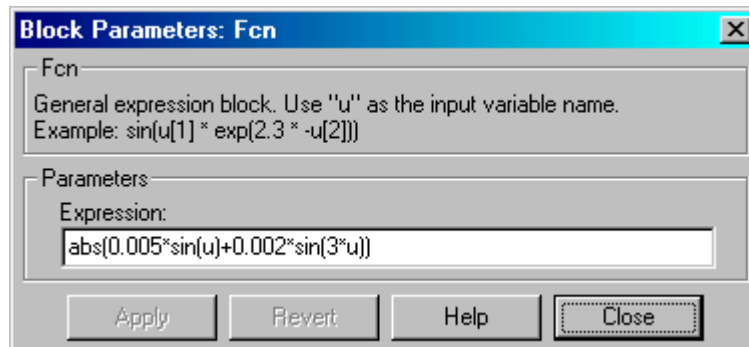


Рис. 8.12. Функция возмущения

Для наблюдения за процессами в модели установим "осциллограф" - блок *Scope*, обозначенный "*Перемещения*", для отображения изменения переменных во времени. Модель готова.

Приступаем к тестированию.

Проведем моделирование для заданных исходных данных.

Результаты моделирования представим в графическом виде (рис. 8.13).

При тестировании оцениваем соответствие модели поставленной задаче.

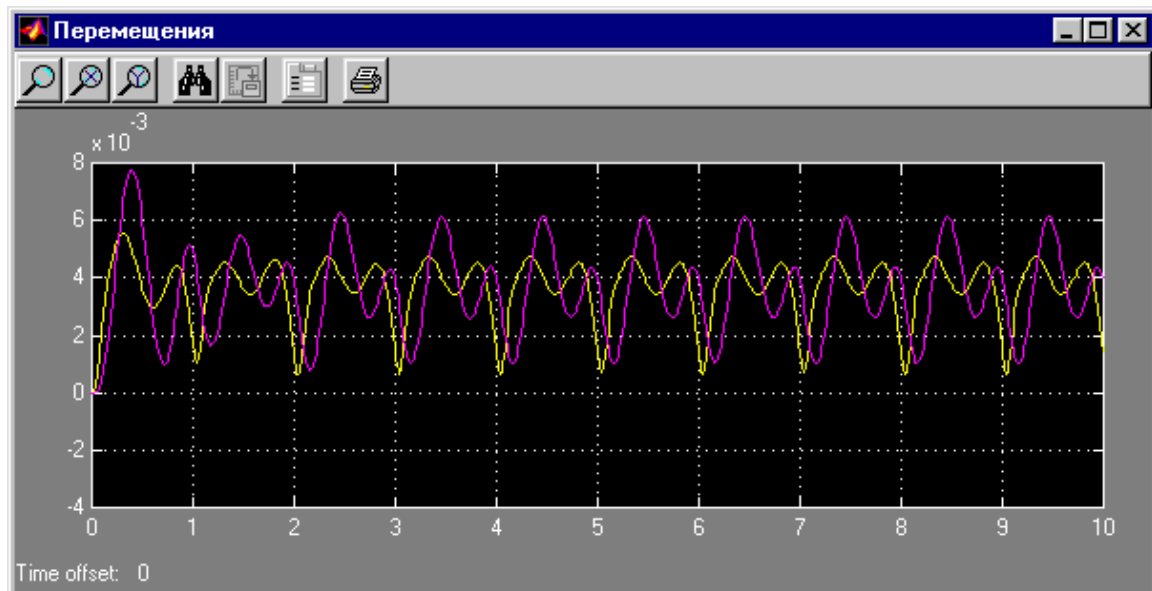


Рис. 8.13. Результаты моделирования – перемещения

**Использование подсистем.** Для моделирования простых систем достаточно использовать подход, описанный выше. При увеличении размерности для увеличения наглядности и удобства отладки выделяют логические

блоки – подсистемы. SIMULINK поддерживает многоуровневые системы. Для вложенных подсистем используют блок Subsystem (рис. 8.14). При раскрытии такого блока выводится окно с моделью соответствующей подсистемы (рис. 8.15, 8.16). Для передачи входных и выходных переменных используются так называемые порты подсистемы входные и выходные. В приведенном примере используются многоканальные порты – через один порт передается переменная и ее производная. Для создания «сложной переменной» (на блок-схеме такие переменные показаны жирными линиями) применяется блок *Mux*, для разделения – блок *Demux*.

Для большой разветвленной модели предпочтительно исходные данные собрать в одном месте, что облегчает настройку при исследовании влияния параметров системы. Один из вариантов – использование модуля присвоения начальных значений переменным MATLAB и использование в модулях SIMULINK имен этих переменных. В модели, показанной на рис. 10, в модуле *START* используется ссылка на подпрограмму в которой осуществляется присвоение начальных значений (рис. 8.17).

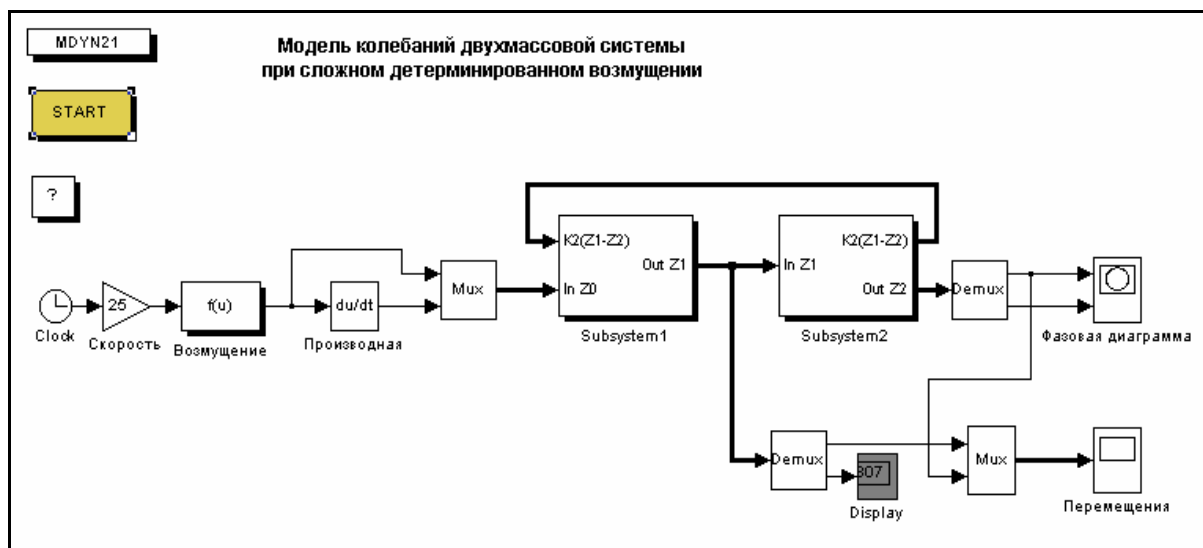


Рис. 8.14. Модель системы

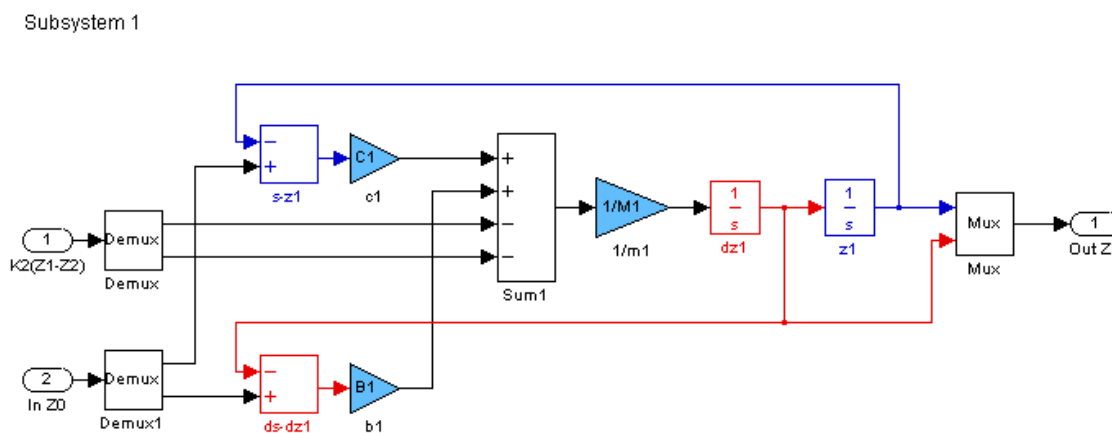


Рис. 8.15. Подсистема, описывающая колебания первого тела



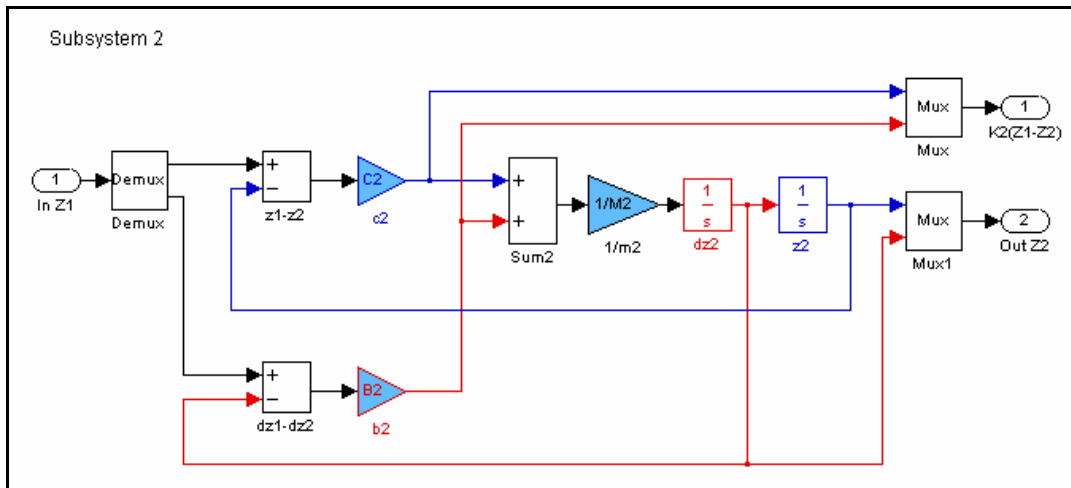


Рис. 8.16. Подсистема, описывающая колебания второго тела

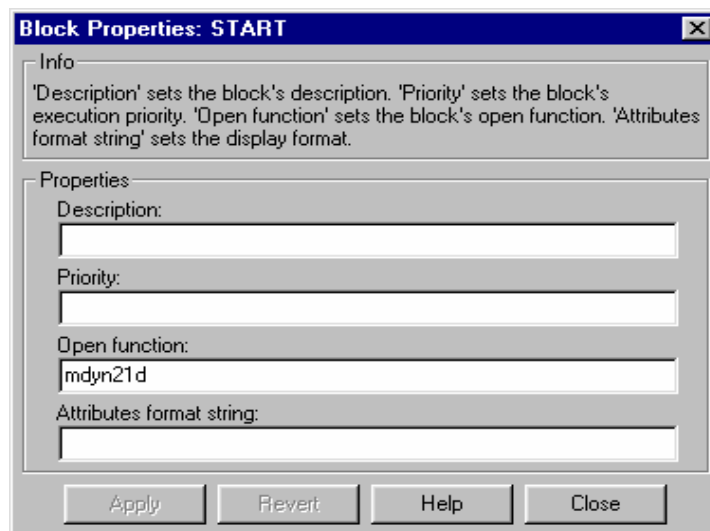


Рис. 8.17. Настройка запускающей программы

Подпрограмма задания начальных значений – обычный m-файл. Для данного примера в файле *MDYN21.m* задаются следующие значения:

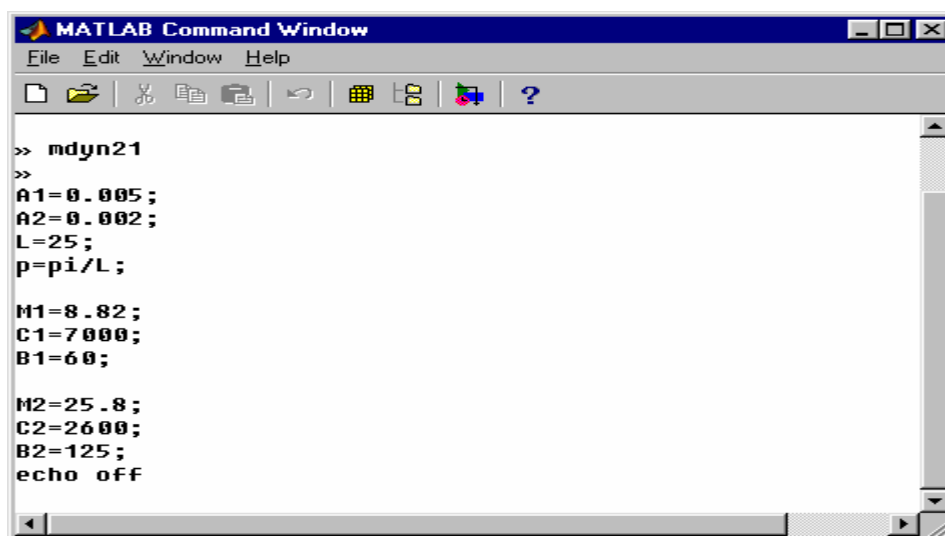
```
% MDYN21
% Start program
%
echo on
A1=0.005;
A2=0.002;
L=25;
p=pi/L;
M1=5.82;
C1=7000;
B1=60;
M2=25.8;
C2=2600;
B2=125;
```

echo off

```
x0 = [0; 0; 0; 0;];
```

```
[t,x]=sim('mdyn21',10,simset('InitialState',x0));
```

В приведенном тексте подпрограммы знак "точка с запятой" в конце строки запрещает вывод значений переменных. Для отображения введенных данных применены операторы «*echo on – echo off*». Выделенный фрагмент подпрограммы выводится в управляющее окно MATLAB (рис. 8.18). Для изменения исходных данных используется текстовый редактор. Запуск моделирующей программы в этом примере осуществляется двойным нажатием на блок *START*.



```
>> mdyn21
>>
A1=0.005;
A2=0.002;
L=25;
p=pi/L;
M1=8.82;
C1=7000;
B1=60;
M2=25.8;
C2=2600;
B2=125;
echo off
```

Рис. 8.18. Вывод в окне управляющей программы MATLAB

Для отображения фазовой диаграммы введем блок отображения *XY\_Graph*, обозначенный на блок-схеме "фазовая диаграмма". Результаты вывода показаны на рис. 8.19.

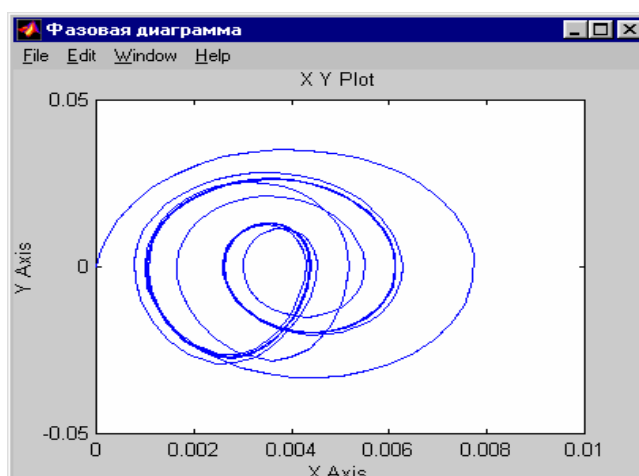


Рис. 8.19. Результаты моделирования – фазовая диаграмма

**Использование подпрограмм пользователя.** Для расширения возможностей моделирования в среде SIMULINK предусмотрена возможность

подключения подпрограмм пользователя, написанных на языке MATLAB. В предыдущей модели заменим функции неровности модулем "MATLAB Function" - "Неровность". Модель определяет ссылку на подпрограмму пользователя, в которой описана функция неровности от пути и ее производная (рис. 8.20). Такой подход позволяет использовать проверенные модули как при программировании в пакете MATLAB, так и при моделировании в среде SIMULINK.

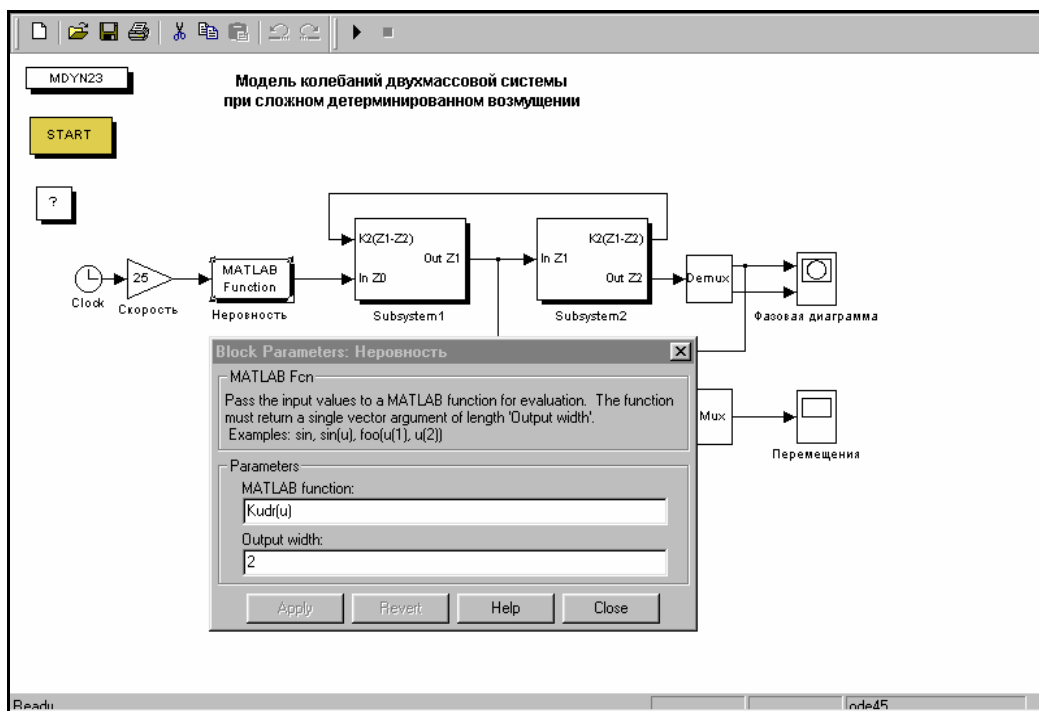


Рис. 8.20. Модель и настройка блока функции MATLAB

## ГЛАВА 9. КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ЭЛЕКТРОННЫХ СХЕМ В СРЕДЕ ELECTRONICS WORKBENCH

Научно-техническое проектирование является основным в развитии науки и техники. Одним из направлений является компьютерное схемотехническое моделирование электронных устройств. Использование интегрированных программных систем схемотехнического моделирования аналоговых и цифровых радиоэлектронных устройств (Micro-Cap V, DesignLab 5.0, APLAC 9.0, System View 1.9, Circuit Maker 7.0, Electronics Workbench, Vissim, LabVIEW) позволяют решать следующие задачи:

- создание модели принципиальной электрической схемы устройства и ее редактирование;
- расчет режимов работы модели;
- расчет частотных характеристик и переходные процессы модели;
- проведение оценки и анализа модели;
- наращивание библиотеки компонентов;
- представление данных в форме, удобной для дальнейшей работы;
- разработка печатных плат;
- подготовка научно-технических документов и многое другое.

Эта глава посвящена системе Electronics Workbench 1.12 и ее модернизированной системе Multisim, разработанной фирмой Interactive Image Technologies. Особенностью системы является наличие виртуальных контрольно-измерительных приборов, по внешнему виду и характеристикам приближенных к их промышленным аналогам. Система легко усваивается и достаточно удобна в работе.

В процессе знакомства с программой и примерами ее многочисленных применений можно убедиться, что весьма удобный пользовательский интерфейс Electronics Workbench действительно является серьезной предпосылкой для использования ее в учебных целях в качестве лабораторного практикума по таким предметам как физика (электричество), основы электротехники и электроники, основы вычислительной техники и автоматики.

Практика использования программы в таком качестве показала, что достаточно двух-трех часов предварительной подготовки, чтобы учащиеся, имеющие опыт работы в среде Windows, могли приступить к самостоятельной работе.

Подробно изучить приемы работы с системой можно по специальным пособиям и руководствам [1-6]. Мы ограничимся первоначальным знакомством с системой и моделированием электрических и логических схем из курса физики и информатики.

### 9.1. Знакомство с системой Electronics Workbench 5.12

Запустив интегрированный пакет Electronics Workbench 1.12, вы увидите диалоговое окно и окно редактирования (рис.9.1). Окно редактирования заполнено некоторыми компонентами. Диалоговое окно Electronics Workbench содержит поле меню, библиотеку компонентов и линейку контрольно-

измерительных приборов, расположенных в одном поле. Поле меню аналогично многим Windows-приложениям. Опции главного меню легко изучить самостоятельно.

Несколько более подробно остановимся на некоторых компонентах и контрольно-измерительных приборах. На рис. 9.1 в окне редактирования, начиная слева (сверху), приведены обозначения следующих компонентов и контрольно-измерительных приборов: заземление, батарея, источник постоянного тока, источник переменного синусоидального тока (эффективное значение тока, частота, фаза), источник переменного синусоидального напряжения (эффективное значение тока, частота, фаза), резистор, конденсатор, катушка (индуктивность), трансформатор, переключатель, электролитический конденсатор, конденсатор переменной емкости, катушка переменной индуктивности, диод, стабилитрон, светодиод, диодный мост, диод Шокли,  $n - p - n$  транзистор,  $p - n - p$  транзистор, далее 4 вида полевых транзисторов, вольтметр, амперметр, лампа накаливания (напряжение, мощность), светодиод (цвет свечения), мультиметр, осциллограф, измеритель амплитудно- и фазо-частотных характеристик.

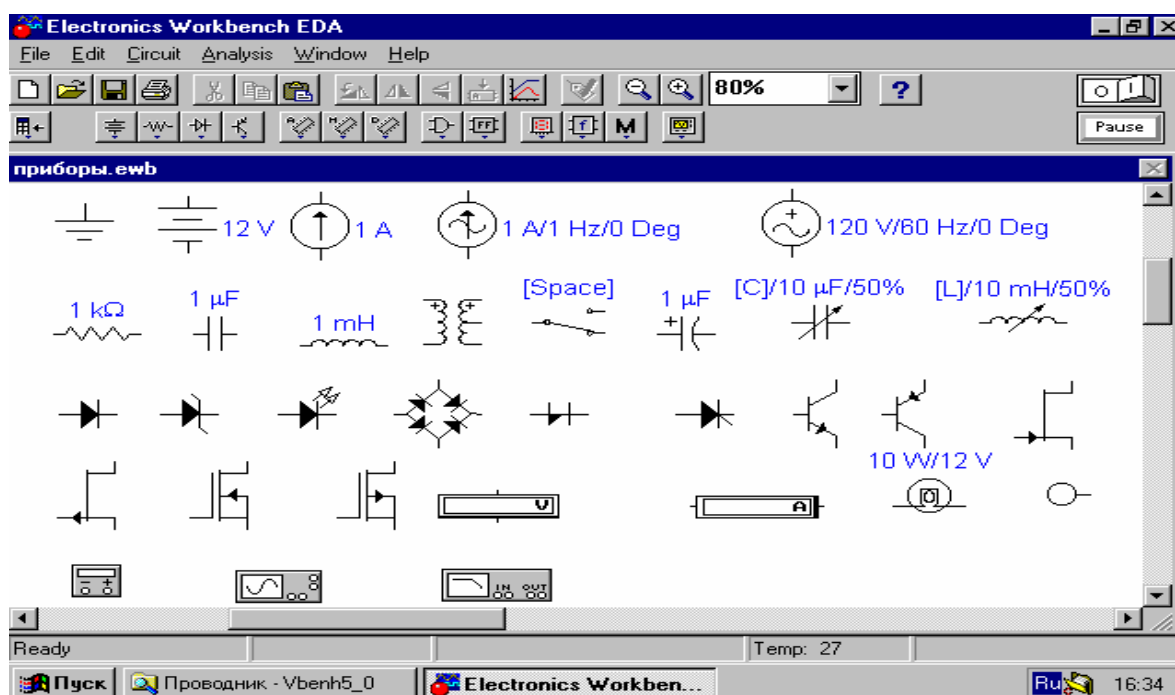


Рис. 9.1

Интерфейс пользователя состоит из полоски меню, панели инструментов и рабочей области. Полоса меню состоит из следующих компонентов: меню работы с файлами (File), меню редактирования (Edit), меню работы с цепями (Circuit), меню анализа схем (Analysis), меню работы с окнами (Window), меню работы с файлами справок (Help). Панель инструментов состоит из “быстрых кнопок”, имеющих аналоги в меню, кнопок запуска и приостановки схем, набора радиоэлектронных аналоговых и цифровых деталей, индикаторов, элементов управления и инструментов.

### Алгоритм технологии подготовки и запуска электрических схем

1. Выбор необходимых компонентов электрической схемы и расположение их в окне редактирования Electronics Workbench 1.12.

Для этого подводим указатель мыши к одной из пиктограмм библиотеки компонентов или линейке контрольно-измерительных приборов и щелкаем левой кнопкой мыши. Выпадает одна из выбранных групп компонентов. Для того чтобы выбрать необходимый, подводим указатель мыши к компоненту, нажимаем левую кнопку мыши (не опускаем кнопку), перемещаем компонент на окно редактирования, отпускаем кнопку.

2. Ввод и изменение параметров выбранных компонентов.

Подводим указатель мыши к компоненту в окне редактирования и щелкаем два раза левой кнопкой мыши. Выпадает меню, состоящее из нескольких опций. Рассмотрим два из них:

*Label* – необходим для написания обозначения компонента;

*Value* – необходим для простановки значений компонента.

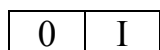
В контрольно-измерительных приборах, при необходимости, например, в вольтметрах и амперметрах, при внесении параметров в опции *Label*, указываем, для какого тока, постоянного или переменного; в *Mode* выбираем DC – для постоянного тока, AC – для переменного.

#### Соединение компонентов электрической схемы

После размещения компонентов и простановки параметров производится соединение их выводов проводниками. При этом необходимо учитывать, что к выводу компонента можно подключить только один проводник. Для выполнения подключения указатель мыши подводим к выводу компонента, и после появления жирной точки (указатель соединения) нажимаем левую кнопку мыши, и появляющийся при этом проводник протягиваем к выводу другого компонента до появления на нем такой же жирной точки, после чего кнопку мыши отпускаем, соединение готово. Если соединение нужно разорвать, указатель мыши подводим к одному из выводов компонента или к точке соединения и при появлении указателя соединения нажимаем левую кнопку, проводник отводим на свободное место рабочего поля, после чего кнопку отпускаем. Если необходимо вывод компонента подключить к имеющемуся на схеме проводнику, то из вывода компонента проводник указателем мыши подводим к указанному проводнику и после появления точки соединения кнопку мыши отпускаем. Отметим, что прокладка соединительных проводов производится автоматически, причем препятствия – компоненты и проводники огибаются по ортогональным направлениям (по горизонтали или вертикали).

#### Подключение электрической схемы к питанию

В правом верхнем углу диалогового окна расположена пиктограмма



0 – отключено питание; I – включено питание. После включения питания на контрольно-измерительных приборах регистрируются характеристики и значения собранной модели электрической схемы.

**Контрольно-измерительные приборы.** Панель измерительных приборов находится под полем меню рабочего экрана, и содержит:

1. Цифровой мультиметр
2. Функциональный генератор,
3. Двухканальный осциллограф,
4. Измеритель амплитудно-частотных и фазо-частотных характеристик,
5. Генератор слов (кодовый генератор),
6. 8-ми канальный логический анализатор,
7. Логический преобразователь.



Рис. 9.2

Общий порядок работы с приборами такой: иконка прибора курсором переносится на рабочее поле и подключается к исследуемой схеме. Для приведения прибора в рабочее (развернутое) состояние необходимо дважды щелкнуть курсором

на его иконке (рис 9.2).

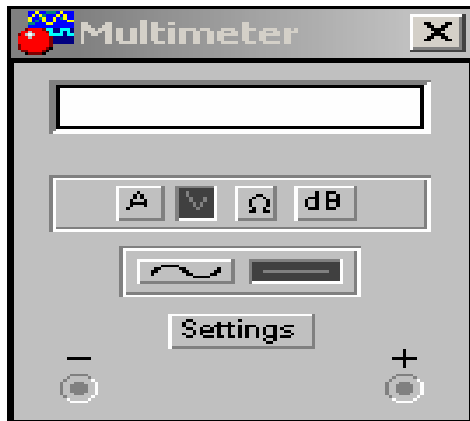


Рис. 9.3. Лицевая панель мультиметра

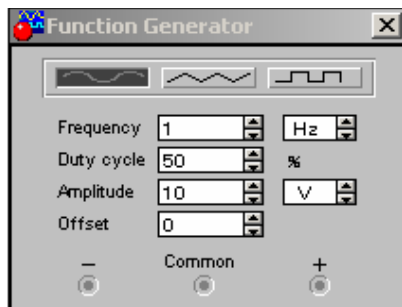
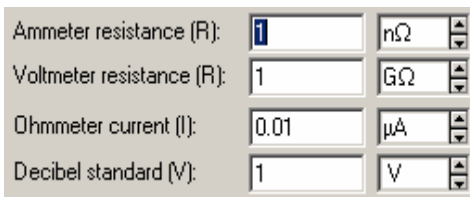


Рис. 9.4. Лицевая панель генератора

**Мультиметр.** На лицевой панели мультиметра (рис. 9.3) расположен дисплей для отображения результатов измерения, клеммы для подключения к схеме и кнопки управления:



-Выбор режима измерения тока, напряжения, сопротивления и ослабления (затухания)



- Выбор режима измерения переменного и постоянного тока

- - режим установки параметров мультиметра. После нажатия на эту кнопку открывается диалоговое окно на котором обозначено:

Ammeter resistance – внутреннее сопротивление амперметра.

Voltmeter resistance - входное сопротивление вольтметра.

Ohmmeter current - ток через контролируемый объект.

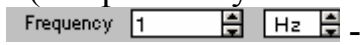
Decibel standard - установка эталонного напряжения  $V_1$  при измерении ослабления или усиления в децибелах (по умолчанию  $V_1 = 1 \text{ В}$ ).

**Функциональный генератор.** Лицевая панель генератора показана на рис. 9.4. Управление генератором осуществляется следующими органами

управления:



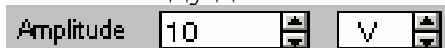
- выбор формы выходного сигнала: синусоидальный (выбрана по умолчанию), треугольный и прямоугольный;



- установка частоты выходного сигнала.



- установка коэффициента заполнения в %: для импульсных сигналов это отношение длительности импульса к периоду повторения – величина обратная скважности, для треугольных сигналов – соотношение между длительностями переднего и заднего фронта;

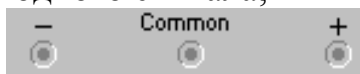


- установка амплитуды выходного сигнала;



- установка смещения (постоянной составляющей)

выходного сигнала;



- выходные зажимы; при заземлении клеммы СОМ (общий) на клеммах “-” и “+” получаем парафазный сигнал.

**Осциллограф.** Лицевая панель осциллографа представлена на рис. 9.5.

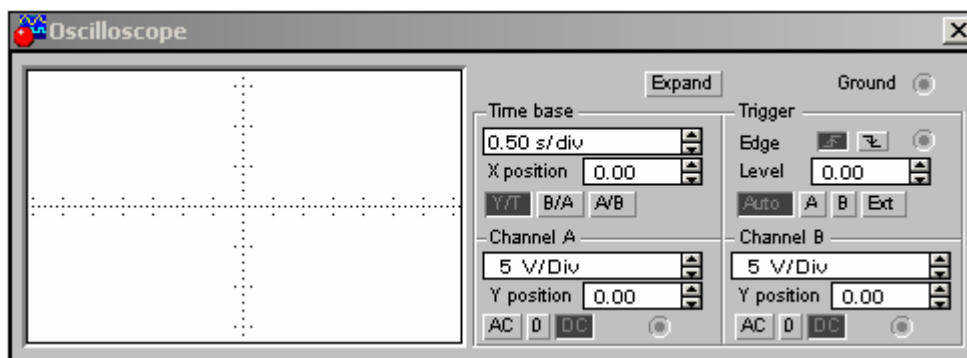


Рис. 9.5. Лицевая панель осциллографа.

Осциллограф имеет 2 канала А и В с разделенной регулировкой чувствительности в диапазоне от 10 мкВ/дел до 5 кВ/дел и регулировкой смещения по вертикали (Y POS). Выбор режима по входу осуществляется нажатием кнопок **AC** **0** **DC**.

- Режим **AC** предназначен для наблюдения только сигналов переменного тока
- В режиме **0** входной зажим замыкается на землю.
- В режиме **DC** (включено по умолчанию) можно проводить измерения как постоянного так и переменного тока.

Режим развертки выбирается кнопками **Y/T** **B/A** **A/B**.

- В режиме **Y/T** (обычный режим, включен по умолчанию) - реализуются следующие режимы развертки по вертикали - напряжение сигнала, по горизонтали время;
- В режиме **B/A** - по вертикали – сигнал канала В, по горизонтали - сигнал канала А
- В режиме **A/B** - по вертикали – сигнал канала А, по горизонтали - сигнал канала В



В режиме Y/T длительность развертки (TIME BASE) может быть задана в диапазоне от 0.1 нс/дел до 1 с/дел с возможностью установки смещения в тех же единицах по горизонтали, т.е. по оси X (X POS).

При нажатии кнопки EXPAND лицевая панель осциллографа существенно меняется – увеличивается размер экрана (рис. 9.6), появляется возможность прокрутки изображения по горизонтали и его сканирования с помощью вертикальных линий (синего и красного цвета), которые за треугольное ушко могут быть установлены в любое место экрана. При этом в индикаторных окошках под экраном приводятся результаты измерения напряжения, временных интервалов и их приращений.

Изображение можно инвертировать нажатием клавиши REVERSE и записать данные в файл нажатием кнопки SAVE. Возврат к исходному состоянию осциллографа – нажатием кнопки REDUCE.

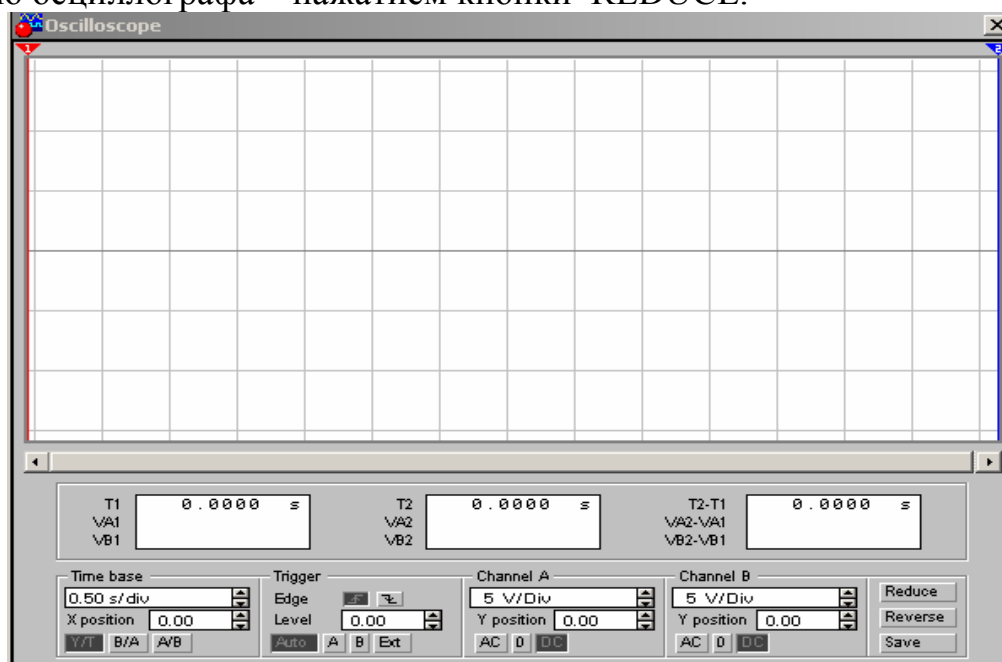


Рис. 9.6. Лицевая панель осциллографа в режиме EXPAND

**Измеритель АЧХ и ФЧХ.** Измеритель предназначен для анализа амплитудно-частотных (при нажатой кнопке MAGNITUDE, включена по умолчанию) и фазо-частотных (при нажатой кнопке PHASE) характеристик при логарифмической (кнопка LOG, включена по умолчанию) или линейной (кнопка LIN) шкале по осям Y и X (рис. 9.7). Настройка измерителя заключается в выборе пределов измерения коэффициента передачи и вариации частоты с помощью кнопок в окошках F – максимальное и I – минимальное значение.

Подключение прибора к исследуемой схеме осуществляется с помощью зажимов IN (вход) и OUT (выход). Левые клеммы зажимов подключаются соответственно к входу исследуемого устройства, а правые – к общей шине. К входу устройства необходимо подключить функциональный генератор или другой источник переменного напряжения, при этом каких-либо настроек в этих устройствах не требуются.

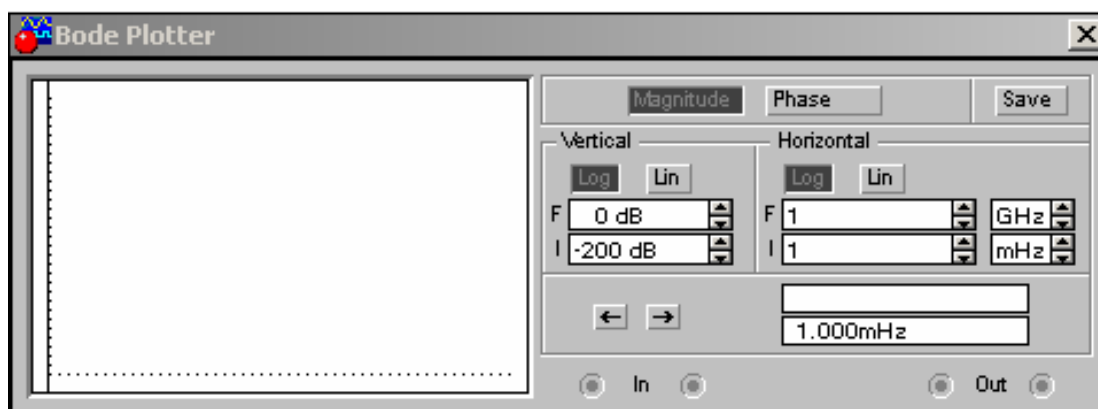


Рис. 9.7. Лицевая панель АЧХ и ФЧХ

## 9.2. Некоторые возможности электронной лаборатории

Прежде чем приступать к разработке или выполнению схемотехнического моделирования в системе Electronics Workbench, необходимо приобрести навыки работы в этом пакете. Поэтому рассмотрим несколько заданий.

**Задание 1.** Создать электронную модель однородного участка электрической цепи в электронной лаборатории Electronics Workbench 1.12 (рис. 9.8). Проверить справедливость закона Ома для этой цепи. Параметры компонентов цепи редактируются в окнах, открывающихся

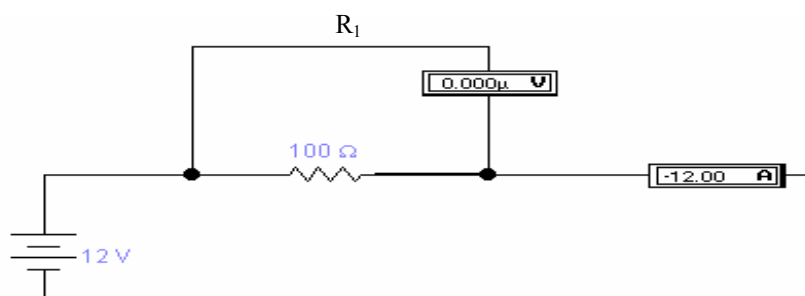


Рис.9.8.

двойным щелчком мыши на элементах схемы.

**Задание 2.** Изучить прибор «осциллограф». В режиме В/А осциллографа получить фазовую диаграмму

**Решение.** Для этого собираем схему, показанную на рис. 9.8.

Воспользуемся функциональным генератором для получения вольтамперной характеристики резисторов, подавая на резистор R1 пилообразное напряжение. Длительность первого полу периода установим равной 99% периода. Напряжение генератора будем подавать на канал А осциллографа с резистора R1. Напряжение, будем подавать на канал В осциллографа с дополнительного резистора R2.

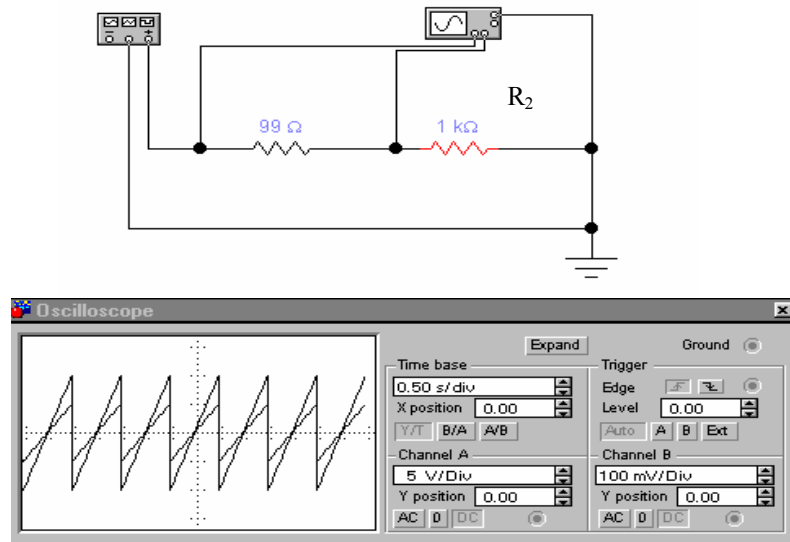


Рис. 9.9

**Задание 3.** Получить диаграмму в графическом формате.

**Решение.** Сохранение диаграммы в графическом формате и ее редактирование осуществляется с помощью программы Аналитические графики (Analysis Graphs) (рис. 9.10).

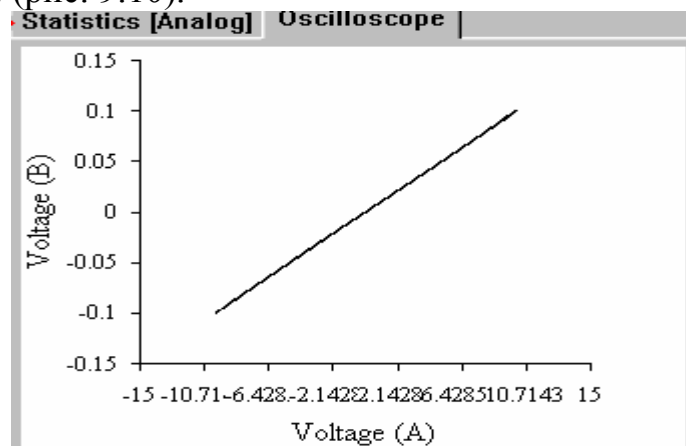
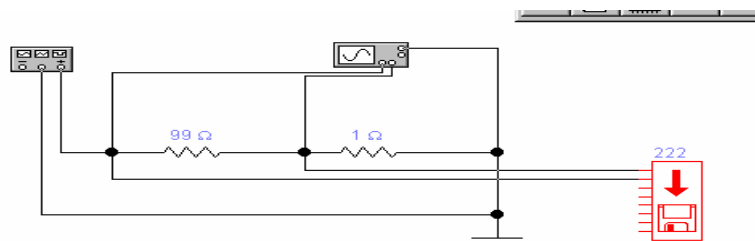


Рис. 9.10

Для установки параметров осей, графиков и линий сетки щелчком правой кнопки мыши вызывается контекстное меню, в которой надо выбрать команду Свойства.

**Задание 7.** Создать текстовый файл.

**Решение.** Для этого найти устройство записи данных **Write Data** в приложении Electronics Workbench (рис. 9.11). Это устройство позволяет сохранить в текстовом файле (данные из текстового файла приведены ниже) уровни потенциала в 8 точках измерения. Эти данные можно вставить в окно Описаний электронной лаборатории Electronics Workbench .



ASCII output data for Electronics Workbench circuit 222.ewb

Time	V1	V2
0.5312	0.731313	0.00731313
0.5712	1.53939	0.0153939
0.6112	2.34747	0.0234747
0.6512	3.15556	0.0315556
0.6912	3.96364	0.0396364
0.7312	4.77172	0.0477172
0.7712	5.5798	0.055798

Рис. 9.11

Результаты получаемые из текстового файла типа txt.

ASCII output data for Electronics Workbench circuit 222.ewb

Time	V1	V2
0.5312	0.731313	0.00731313
0.5712	1.53939	0.0153939
0.6112	2.34747	0.0234747
0.6512	1.15556	0.0315556
0.6912	1.96364	0.0396364
0.7312	7.77172	0.0477172
0.7712	1.5798	0.055798
0.8112	7.38788	0.0638788
0.8512	7.19596	0.0719596
0.8912	5.00404	0.0800404
0.9312	5.81212	0.0881212
0.9712	7.6202	0.096202
0.99	10	0.1

**Задание 1.** Копировать любую область окна Electronics Workbench

**Решение.** В Electronics Workbench можно имитировать и ручной режим проведения опыта: увеличивать напряжений на резисторе 100 Ом, уменьшая сопротивление переменного резистора нажатием клавиши R (или любую, назначенную пользователем) согласно определенному декременту. Показания приборов в этом случае просто вводятся с клавиатуры в окно **Description**. На Рис. 9.12 изображен скопированный в буфер обмена результат такого эксперимента.

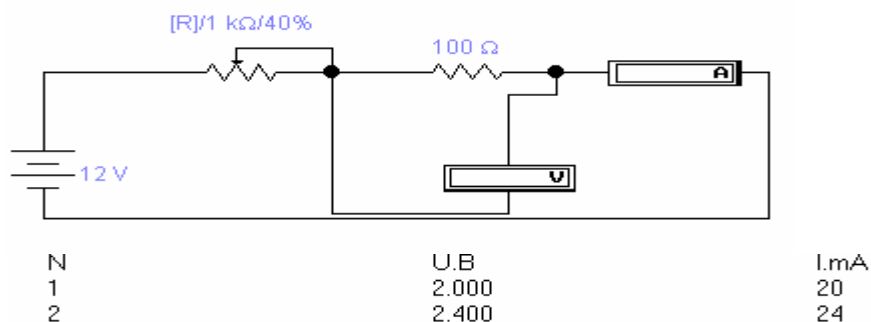


Рис. 9.12

Копирование любой области окна Electronics Workbench 1.12 производится с помощью команды **Копировать** как точечный рисунок (**Copy as Bitmap**), просмотр содержимого буфера обмена – с помощью команды Показать буфер обмена (**Show Clipboard**) в меню Правка (**Edit**). Копированный таким образом рисунок также можно вставить в текст Word.

### 9.3. Индуктивность и емкость в цепи переменного тока

**Цель:** Изучение процессов в электрической цепи переменного тока при наличии индуктивности и емкости.

**Оборудование:** Электронная лаборатория **Electronic Workbench 1.0c**

Приведем пример для самостоятельной лабораторной работы из книги [43]. Для более глубокого понимания процессов, происходящих в последова-

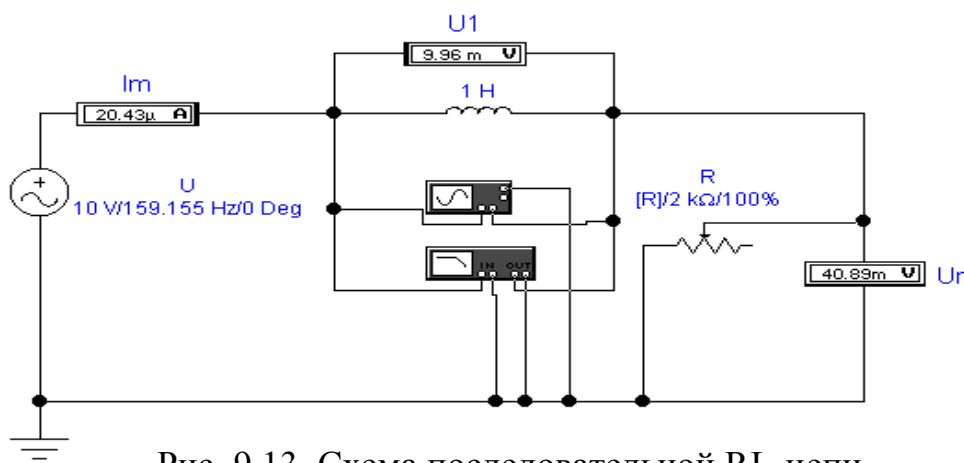


Рис. 9.13. Схема последовательной RL-цепи

тельной RLC-цепи, целесообразно рассмотреть последовательные RL- и RC- цепи. Для последовательной RL-цепи (рис. 9.13), выражения для мгновенного значения напряжения имеют вид:

$$I_m = \frac{U_m}{|Z|} = \frac{U_m}{\sqrt{R^2 + (wL)^2}}; \operatorname{tg} B_r = wL/R, B_r \leq 90^\circ.$$

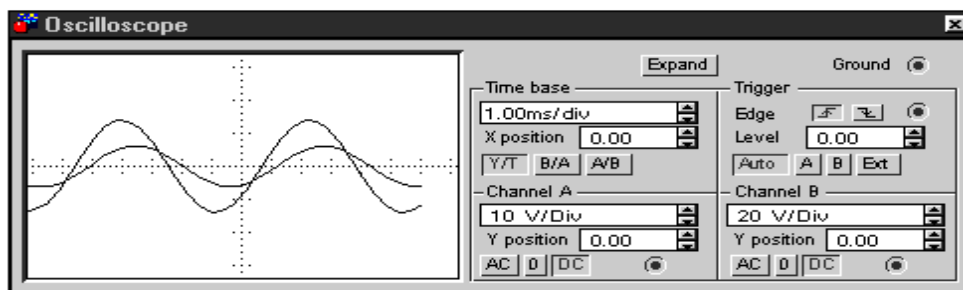


Рис. 9.14. Осциллограммы сигналов в последовательной RL-цепи

Модуль полного сопротивления RL-цепи при указанных на рис. 9.31 параметрах компонентов на частоте  $\omega = 2\pi F = 2\pi \cdot 159,155 \approx 1000$  рад/с равен

$$|Z| = \sqrt{1000^2 + (1000*1)^2} = 1414 \text{ Ом.}$$

Амплитуда и фаза тока:

$$I_m = U_m / |Z| = 10 / 1414 = 7,07 \text{ mA}; \quad \text{tg} B_r = (1000 * 1) / 1000 = 1, \quad B_r = 45^\circ.$$

Падения напряжения на сопротивлении и индуктивности

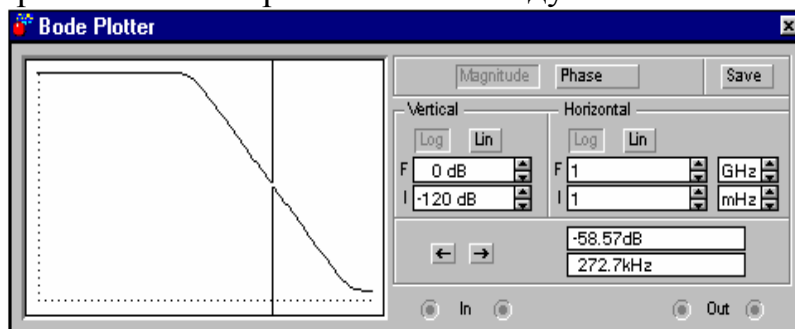


Рис. 9.15. Фазо-частотная характеристика последовательной RL-цепи

$U_r = I_m R = 0.00707 * 1000 = 7.07 \text{ V}$ ;  $U_l = I_m X_l = 0.0070 * 1000 = 7.07 \text{ V}$ , что совпадает с показаниями приборов на рис. 9.31. Для определения фазовых соотношений рассмотрим осциллограммы на рис. 9.15, откуда видно, что напряжение на входе RL- цепи (осциллограмма А, визирная линия): 1) опережает напряжение на сопротивление R (осциллограмма В, визирная линия); 2) и, соответственно, ток цепи на время  $T_2 - T_1 = 0.000785 \text{ с}$ . Поскольку период колебаний в данном случае равен  $1 / 157.155 = 0.0063 \text{ с}$ , то в градусах этот временной промежуток составит  $(360^\circ * 0.000785) / 0.0063 = 47.98^\circ$ , т.е. погрешность моделирования менее 0.05%.

Заметим, что с повышением частоты этот угол стремится к  $90^\circ$ , что удобно с помощью измерителя АЧХ-ФЧХ в режиме PHASE (режим ФЧХ). Результаты измерений показаны на рис. 9.16, откуда видно, что уже на частоте 12 кГц (В этой точке стоит визирная линия) ток в RL- цепи запаздывает относительно напряжения на ее входе более, чем на  $89^\circ$ .

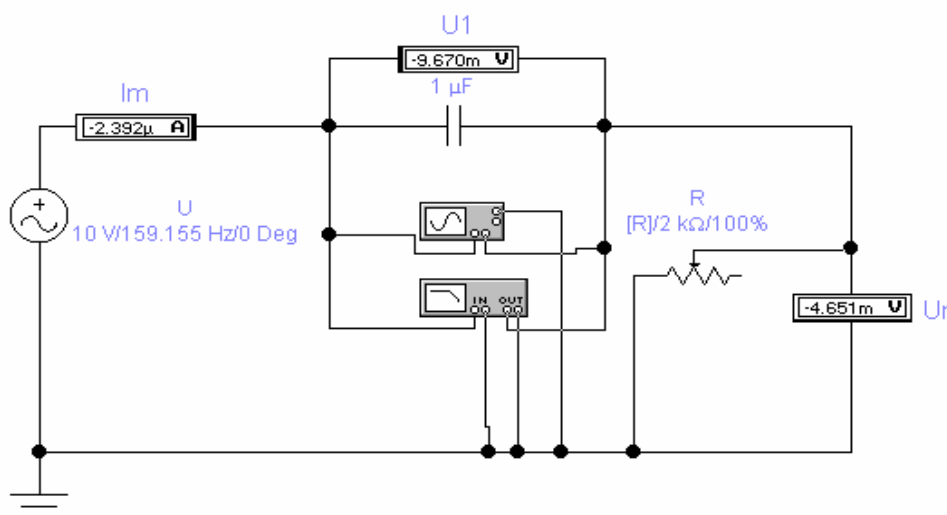


Рис. 9.16. Схема последовательной RC-цепи

Для последовательной RC-цепи (рис. 9.33) получаем:

$$I_m = U_m / |Z| = U_m / \sqrt{R^2 + (1 / \omega * C)^2},$$

$$\text{tg} B_r = -1 / \omega RC, \quad B_r \geq -90^\circ.$$

## 9.4. Источники вторичного питания (выпрямление с фильтрацией)

**1. Постановка задачи.** Рассмотрим задачу моделирования выпрямления электрического напряжения переменного тока и фильтрации с помощью системы Electronics Workbench. Цель фильтрации при выпрямлении – ослабить переменную составляющую выпрямленного напряжения. Здесь необходимо промоделировать работу фильтра выпрямителя.

На входе фильтра действует выпрямленное напряжение  $E(t)$  одно- или двухполупериодное, полученное из переменного синусоидального напряжения с помощью генератора сигналов (рис.9.17). Для ослабления переменной составляющей в схеме фильтра имеется реактивный элемент: конденсатор емкостью  $C=800$  мкф. Фильтр работает на нагрузку сопротивлением  $R_H=0,5$  Ком.

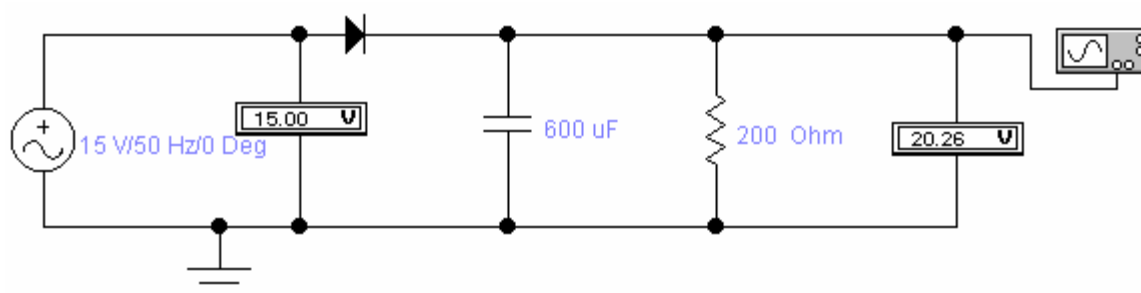


Рис. 9.17. Однополупериодный выпрямитель

Однополупериодную схему обычно применяют при выпрямленных токах до нескольких десятков миллиампер и в тех случаях, когда не требуется высокой степени сглаживания выпрямленного напряжения. Эта схема характеризуется низким коэффициентом использования мощности трансформатора.

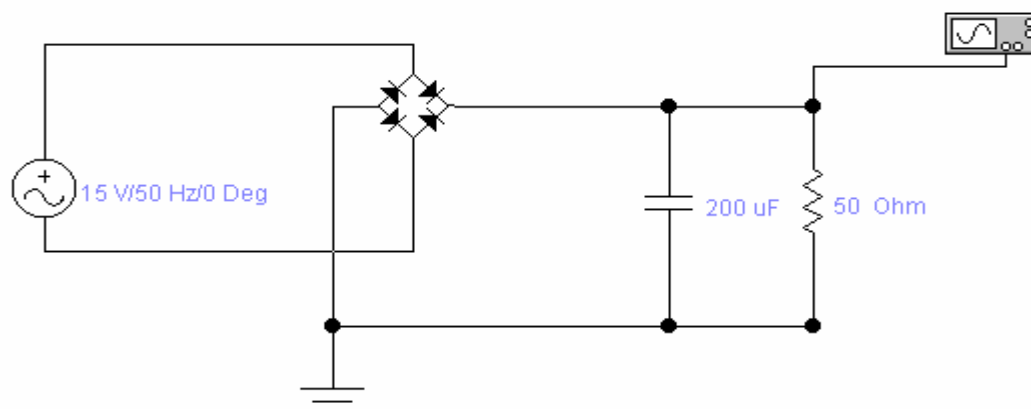


Рис. 9.18. Мостовая схема

Однофазная мостовая схема характеризуется высоким коэффициентом использования мощности и поэтому может быть рекомендована для использования в устройствах повышенной мощности при выходных напряжениях от десятков до сотен вольт, пульсации такие же, как и в предыдущей схеме.

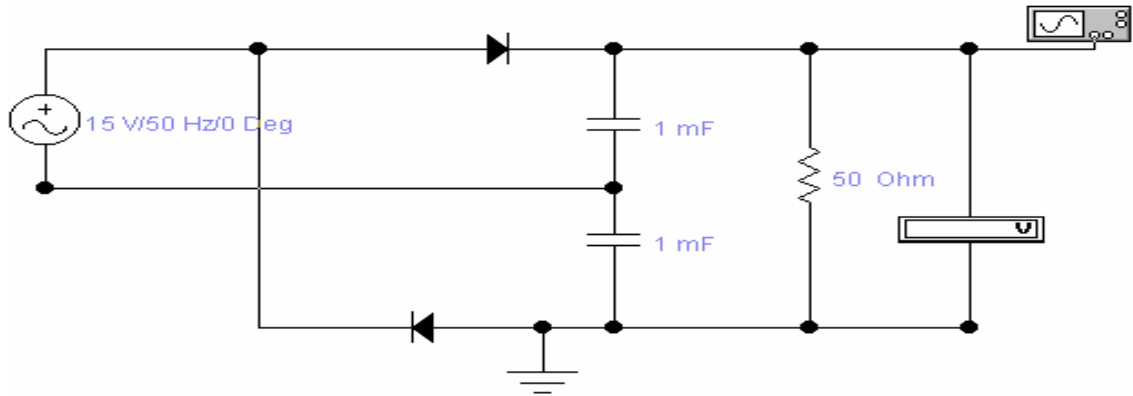


Рис. 9.19. Схема удвоения напряжения

Симметричная схема удвоения напряжения представляет собой последовательное соединение двух однополупериодных выпрямителей и применяется при высоких напряжениях (до 1...2 кВ) при небольших токах нагрузки. Пульсации на каждом конденсаторе схемы удвоения в 2 раза больше пульсаций на ее входе.

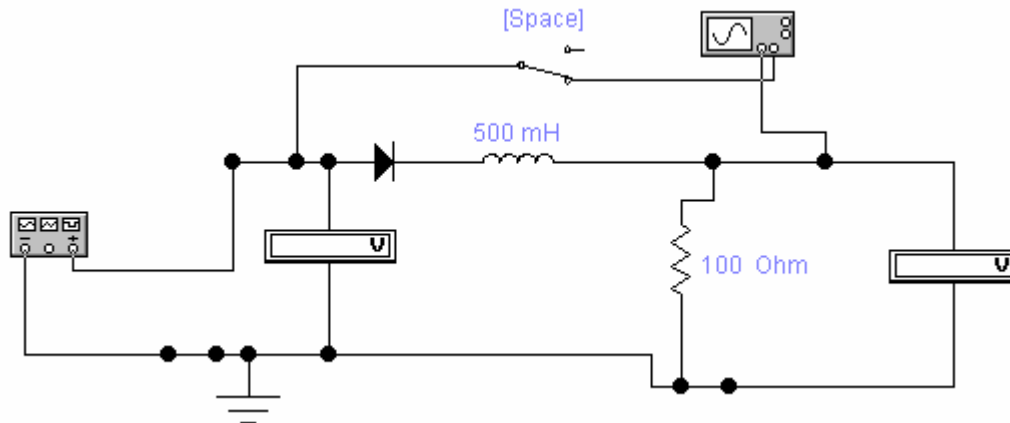


Рис. 9.20

## 2. Задания к лабораторной работе

2.1. Собрать схему выпрямления и фильтрации тока приведенным схемам. Провести моделирование выпрямления тока. Исследовать в зависимости от параметров фильтра как изменяется выходное выпрямленное значение напряжения.

2.2. Заменить выпрямитель двухполупериодным. Рассмотреть, насколько отличаются результаты моделирования.

2.1. Найти оптимальные параметры сопротивления и емкости, при котором получается полное выпрямление.

2.7. Зарисовать изображение выходного  $u(t)$ , для приведенных схем выпрямления.

**3. Результаты моделирования.** На рис. 9.21 изображено выходное  $u(t)$ , напряжение фильтра для однополупериодного выпрямителя. Выбор определенного напряжения  $E(t)$  на вход выпрямителя осуществляется с помощью генератора переменного напряжения. С помощью изменения параметров фильтра  $r$ ,  $C$  и нагрузки  $R_H$  можно оценить качество фильтрации.



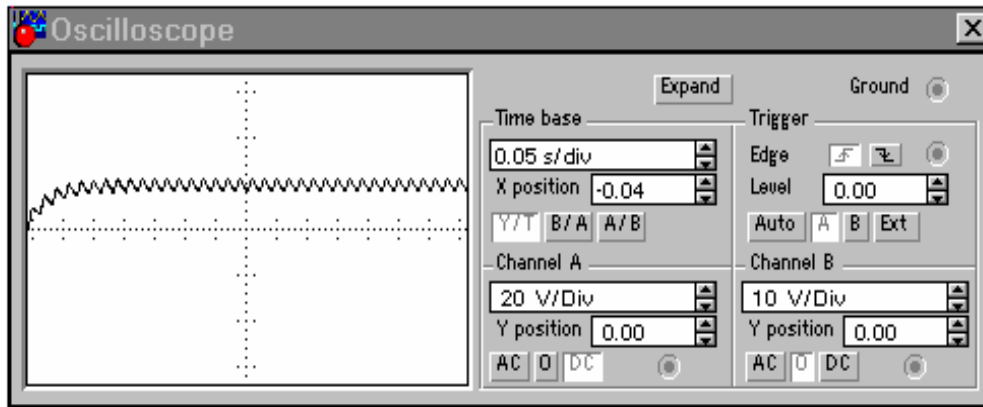


Рис. 9.21

## 9.5. Исследование логических схем в системе EWB

**Задача 1.** Проведите анализ логического устройства (рис.9.22): по функциональной схеме составьте структурную формулу, упростите ее, если это возможно.

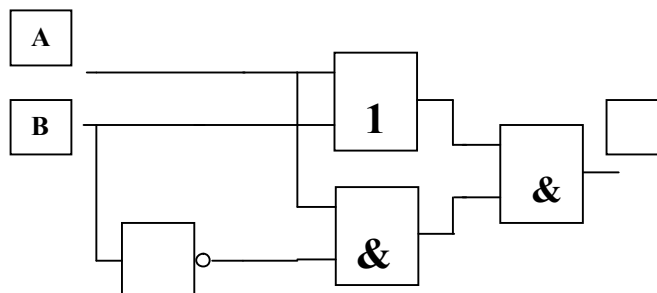


Рис. 9.22

*Решение.* 1. Составление логической функции для функциональной (логической) схемы. При составлении логической функции необходимо проследить пути движения потоков сигналов (рис.9.23).

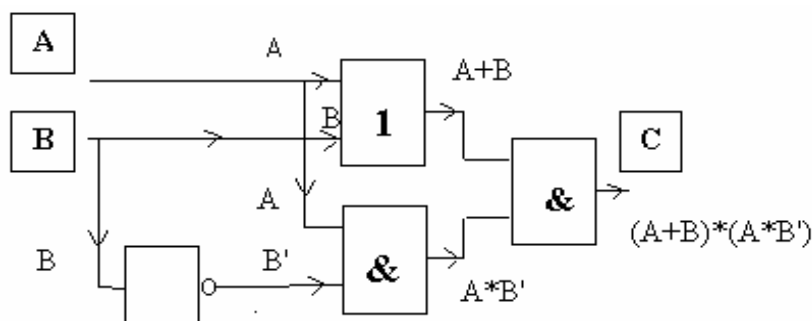


Рис.9.23

Ответ:  $(A+B) * (A * B')$ .

2. Проверка на избыточность функциональной схемы (упростить логическую функцию, т. е. преобразовать с помощью законов алгебры логики).

$(A+B) * (A * B')$  /Скобки для  $A * B'$  опускаем, так как перед скобками

тоже знак  $*$   $=(A+B)*A*B'=($  Для  $A*B'$  применяем закон коммутативности  $/$   
 $=(A+B)*B'*A= /$  Для  $(A+B)*B'$  применяем закон дистрибутивности  $/$   $=$   
 $((A*B')+(B*B'))*A= / B*B'=0/ = ((A*B')+0)*A=/$  Поглощение 0 при дизъ-  
 юнкции  $/$   $=(A*B')*A=/$  Скобки опускаем, применяем закон коммутативно-  
 сти  $/$   $=A*A*B'=/ A*A=A=/A*B'$ .

3. Проверяем справедливость логических преобразований. Для этого составляем таблицу истинности. В общем случае составляется две таблицы: для исходной и конечной логических функций. В этой задаче достаточно одной. Значения таблиц истинности  $A*B'$  и  $(A+B)*(A*B')$  равны, что доказывает справедливость логических преобразований.

A	B	A+B	B'	A*B'	(A+B)*(A*B')
0	0	0	1	0	0
0	1	1	0	0	0
1	0	1	1	1	1
1	1	1	0	0	0

4. По полученной логической функции составляем функциональную схему (рис.9.24).

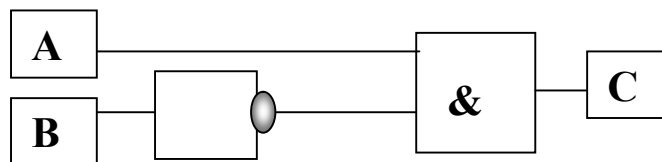


Рис. 9.24

**Задача 2.** Проведите синтез трехвходового логического устройства с выходной комбинацией 10011110 в таблице истинности.

*Решение.* 1. Составим таблицу истинности для данного логического устройства.

A	B	C	F(A,B,C)	
0	0	0	0	
0	0	1	0	
0	1	0	1	$A'*B*C'=1$
0	1	1	1	$A'*B*C=1$
1	0	0	0	
1	0	1	0	
1	1	0	1	$A*B*C'=1$
1	1	1	0	

Так как в таблице F единиц меньше чем нулей, то построим СДНФ:  $(A'*B*C')+(A'*B*C)+(A*B*C')$ .

2. Используя правила алгебры логики попробуем его упростить.  
 $(A'*B*C')+(A'*B*C)+(A*B*C')= [((A'*B)*C')+((A'*B)*C)]+(A*B*C')=$   
 $=[(A'*B)*(C'+C)]+(A*B*C')= / C'+C=1/ =[(A'*B)*1]+(A*B*C')=$   
 $=(A'*B)+(A*B*C')= (A'*B)+((A*C')*B)= (A'+(A*C'))*B=B*(A'+A*C).$

Упрощаем дальше, используя закон де Моргана.

$$\begin{aligned} & \mathbf{B*(A'+A*C')}=\mathbf{B*(A''*(A*C'))}'=\mathbf{B*(A*(A'+C'))}'= \\ & =\mathbf{B*((A*A')+(A*C))}'=\mathbf{A*A'=0, 0+(A*C)=A*C}=\mathbf{B*(A*C)}'=\mathbf{B*(A'+C')}= \\ & =\mathbf{B*A'+B*C'}. \end{aligned}$$

(Проверку можно осуществить с помощью таблиц истинности).

Ответ:  $\mathbf{B*A'+B*C'}$ .

3. По полученной структурной формуле построим функциональную схему (рис.9.25).

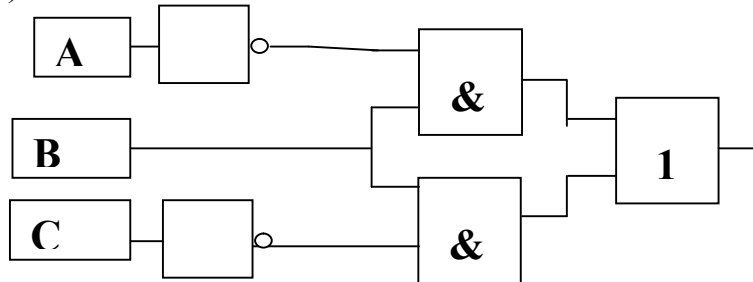


Рис.9.25

## 9.6. Виртуальный логический конвертор (LOGIC CONVERTER)

Для изучения назначения и принципа работы виртуального устройства логического конвертора (преобразователя) и знакомства с базовыми функциями логического конвертора задач воспользуемся программой электронной лаборатории Electronics Workbench. Для построения логических схем в библиотеке Logic Gates (логические элементы) предусмотрена возможность выбора логических элементов. На рис. 9.25 перечень выбора возможных логических элементов.

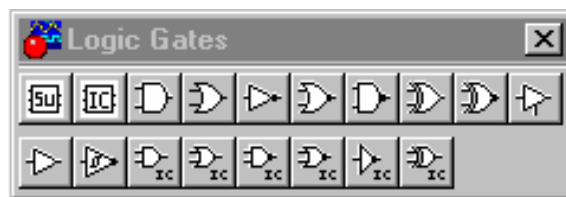


Рис. 9.25

На рис. 9.26 показаны обозначения, используемые в Electronics Workbench логических элементов: конъюнкции - И, дизъюнкции – или, отрицания –НЕ, 2 – И – НЕ, 2 –ИЛИ – НЕ.

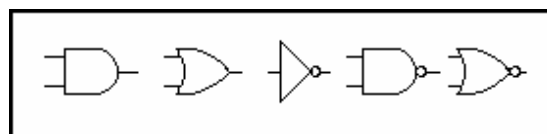


Рис. 9.26

В электронной лаборатории Electronics Workbench имеется виртуальное устройство. Логический конвертор (Logic Converter) позволяет осуществ-

влять 6 логических преобразований для логической функции с числом переменных от 1 до 8: представление таблицы истинности, собранной из логических элементов схемы; обращение таблицы истинности в логическую формулу (СДНФ); минимизацию СДНФ; обращение формулы в таблицу истинности; представление формулы в виде схемы в логическом базисе 2-И-НЕ. Логический конвертор выбирается из меню Instruments (рис. 9.27).



Рис. 9.27

Приведем описание технологии исследования логических схем с помощью логического конвертора (преобразователя).

1. Собираем логическую схему.
2. Подключаем исследуемую логическую схему к логическому конвертору (входов 8, выход один – расположен справа).
3. Открываем логический конвертор щелчком левой кнопкой мыши по иконке конвертора. На экране появляется меню Logic Converter.
4. Для получения таблицы истинности нажимаем



5. Для получения логической функции (структурной формулы) нажимаем



С помощью логического конвертора можно проводить не только анализ логических устройств, но их синтез. Приведем описание технологии синтеза логического устройства по выходной комбинации с помощью логического конвертора (преобразователя).

1. Раскрываем лицевую панель логического конвертора (рис.9.28).

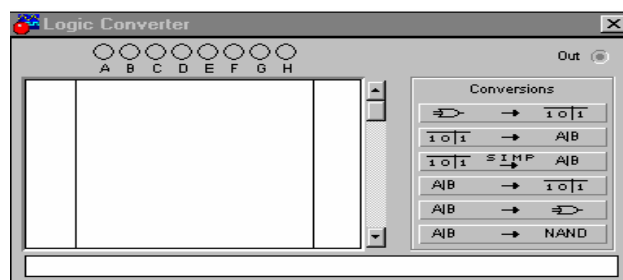


Рис.9.28

2. Активизируем курсором клеммы-кнопки А, В, ..Н (начиная с F), количество которых равно количеству входов синтезируемого устройства (количеству логических переменных).
3. Вносим необходимые изменения в столбец OUT и после нажатия на клавиши на панели преобразователя получаем результат в виде схемы на рабочем поле программы и логическую функцию в дополнительном дисплее.

**Задача 3.** Проведите анализ логического устройства (рис.9.29) по функциональной схеме с помощью Electronics Workbench.

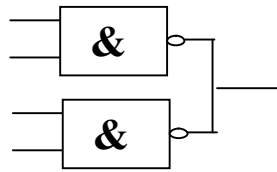


Рис.9.29

На рис. 9.30 решение задачи в Electronics Workbench. В приложении приведены некоторые лабораторные работы по изучению основ работы элементов ЭВМ.

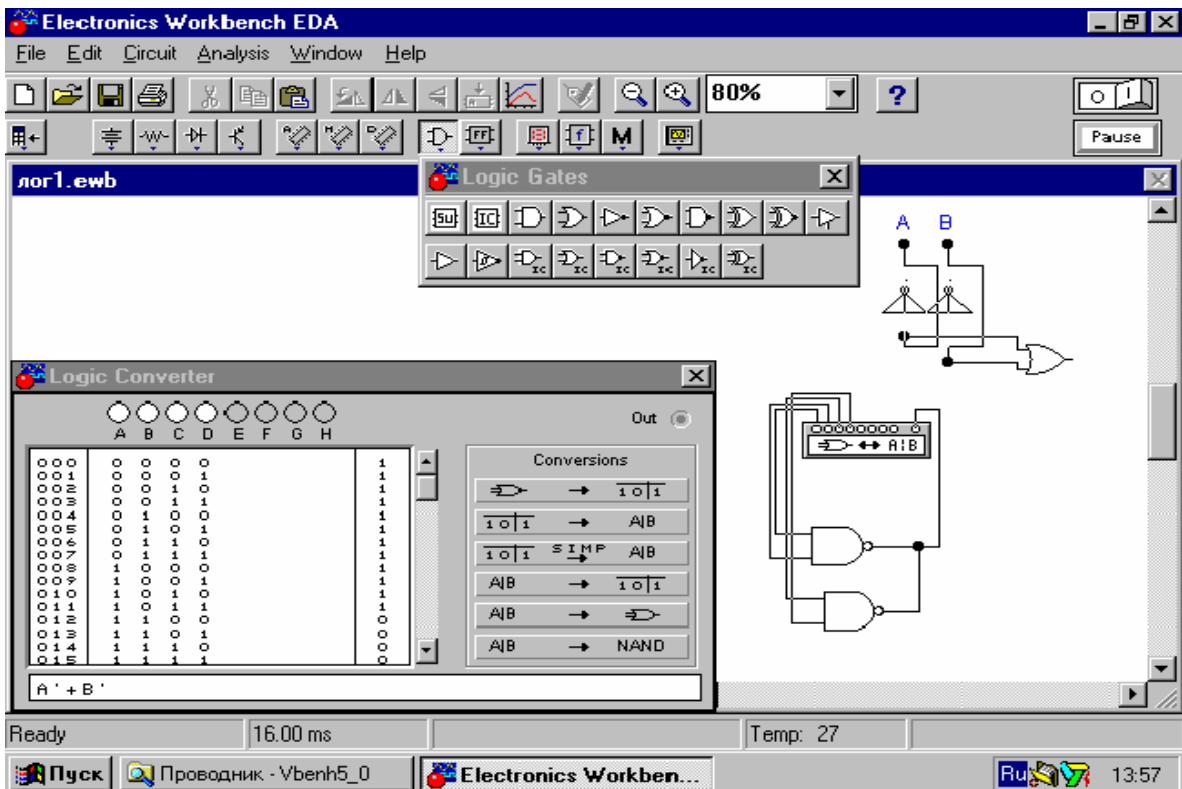


Рис.9.30

### Задания на моделирование

1. Исследуйте следующие функциональные схемы (рис.9.31).

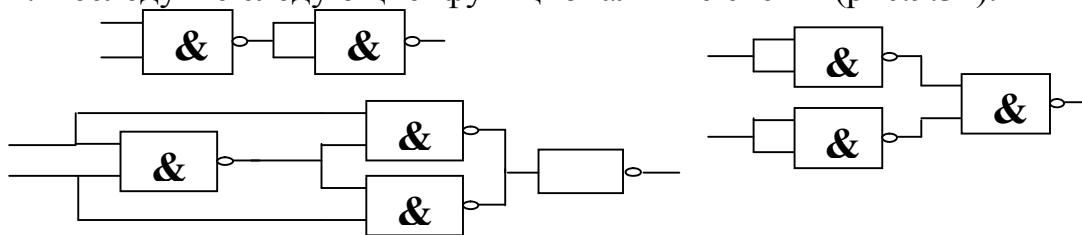


Рис.9.31

2. Исследуйте логическую схему и постройте функциональную логическую

- схему: а)  $B \cdot C' + A \cdot C$ .  
 б)  $A \cdot B' \cdot C + A \cdot B \cdot C' + A' \cdot B \cdot C$ .  
 в)  $A \cdot (B + C) \cdot (D + C)$ .

3. Проведите синтез логического устройства с выходной комбинацией:

- а) 00100111.  
 б) 01101001.  
 в) 0110100110010110.

## 9.7. Решение дифференциальных уравнений

В системе можно проводить решение дифференциальных уравнений с помощью схемотехнического моделирования.

Рассмотрим пример: уравнение свободного затухающего колебания, который можно записать в виде:

$$\frac{d^2 x}{dt^2} + K \frac{dx}{dt} + w_0^2 x = 0,$$

т.е. найти  $x = x(t)$  при заданных начальных условиях:

$$t = 0, v = v_0, x = x_0.$$

Это уравнение описывает затухание зарядов в колебательном контуре или колебания маятника с трением.

На рис. 9.32 представлены: сумматор, интегратор и инвертирующий усилитель в системе Electronics Workbench:

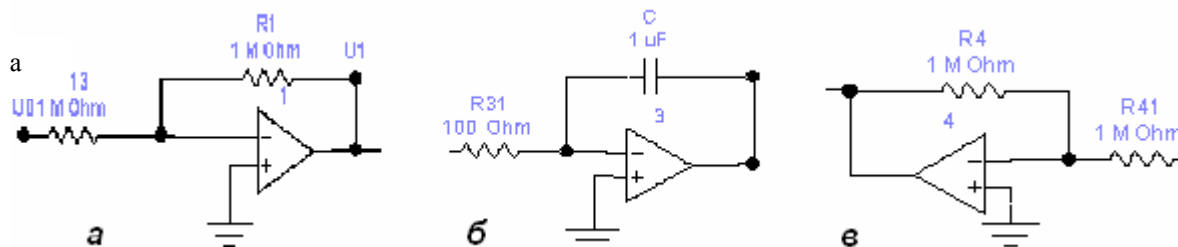


Рис. 9.32. Масштабирующие преобразователи: а – сумматор, б – интегратор, в - инвертирующий усилитель

Здесь S – блок суммирования (сумматор), I – блок интегрирования (интегратор). На рис. 9.33 представлена схема аналогового решения задачи последовательного колебательного контура, состоящей из сумматора на операционном усилителе (ОУ) 1, двух интеграторов на ОУ 2 и 3 и инвертирующего усилителя на ОУ 7. Позиционное обозначение резисторов принято в соответствии: порядковый номер резистора в цепи обратной связи совпадает с номером ОУ, токозадающие резисторы имеют двухзначный номер, первым идет номер соответствующего ОУ, вторым – порядковый номер резистора данной группы. Поскольку сигнал на выходе ОУ 2 имеет полярность, совпадающую с полярностью сигнала на входе сумматора на ОУ 1, то для обеспечения устойчивости модели выход интегратора на ОУ 2 соединен с входом сумматора

через инвертирующий усилитель на ОУ 7. Рис. 9.33 демонстрирует также результат моделирования.

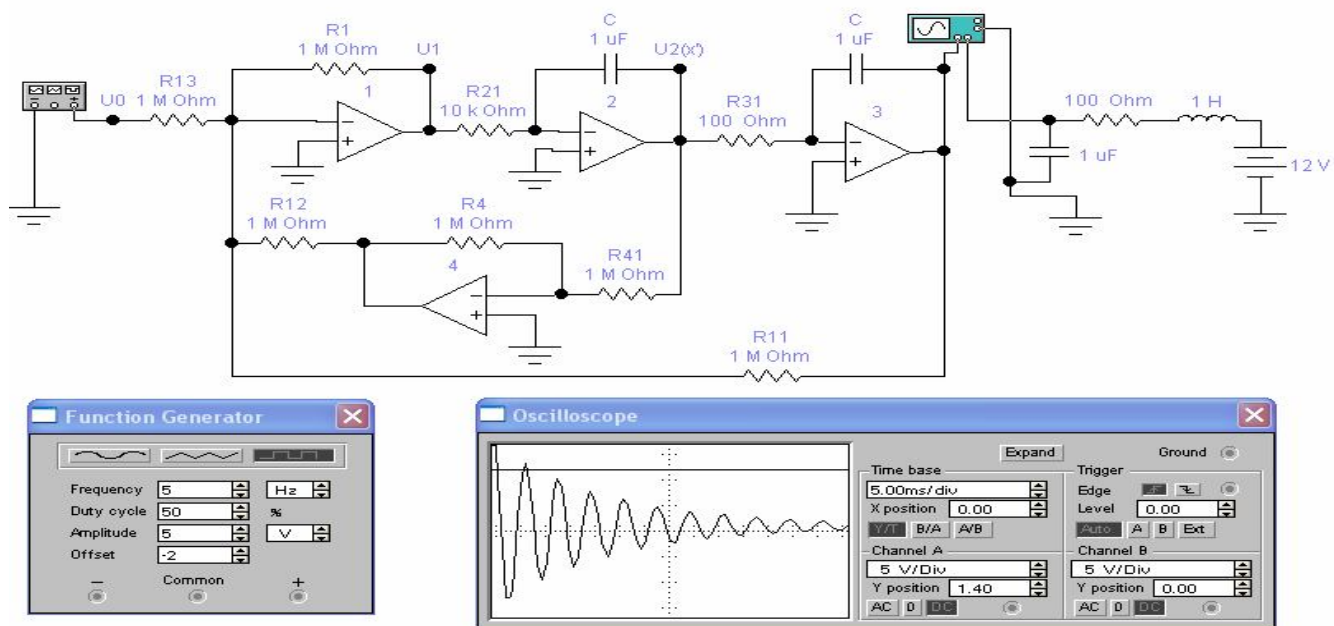


Рис. 9.33. Электронная модель и моделирование последовательного колебательного контура на решающих элементах

## ГЛАВА 10. КОМПЬЮТЕРНОЕ ПРОЕКТИРОВАНИЕ ВИРТУАЛЬНЫХ ПРИБОРОВ В СРЕДЕ LABVIEW

Разработчиком среды графического программирования LabVIEW является компания **National Instruments** (Остин, Техас, США). Заказчиками компании являются инженеры, ученые и технические специалисты, работающие в самом широком спектре отраслей и технологий.

### 10.1. Что такое LabVIEW?

Среда LabVIEW представляет собой высокоэффективную информационную систему, в которой можно создавать гибкие и масштабируемые приложения измерений, управления и тестирования с минимальными временными и денежными затратами. LabVIEW сочетает в себе гибкость традиционного языка программирования с интерактивной технологией Экспресс Виртуального программирования, которая включает в себя автоматическое создание кода, использование помощников при конфигурировании измерений, шаблоны приложений и настраиваемые Экспресс ВП. Благодаря этим особенностям и новички, и эксперты могут легко и быстро создавать приложения в LabVIEW.

#### Достоинства LabVIEW:

- Полноценный язык программирования
- Интуитивно понятный процесс графического программирования
- Широкие возможности сбора, обработки и анализа данных, управления приборами, генерации отчетов и обмена данных через сетевые интерфейсы
- Драйверная поддержка более 2000 приборов
- Возможности интерактивной генерации кода
- Шаблоны приложений, тысячи примеров
- Высокая скорость выполнения откомпилированных программ
- Обучение и техническая поддержка мирового уровня
- Совместимость с операционными системами Windows2000/NT/XP, Mac OS X, Linux и Solaris.

**Широта применения.** Приложения, написанные в LabVIEW, находят применение во всем мире в разнообразных отраслях промышленности:

- Автомобильная промышленность
- Телекоммуникации
- Аэрокосмическая промышленность
- Полупроводниковая промышленность
- Разработка и производство электроники
- Управление технологическими процессами
- Биомедицина

Благодаря своей гибкости и масштабируемости, LabVIEW может использоваться на всех этапах технологического процесса: от моделирования и разработки прототипов продуктов до широкомасштабных производственных



испытаний. Применение интегрированной среды LabVIEW для измерения сигналов, обработки результатов и обмена данными повысит производительность всего предприятия.

National Instruments является одним из пионеров в разработке и продвижении технологии виртуальных приборов, предназначенных для измерений, управления и автоматизации. Сегодня LabVIEW, являющаяся стандартом среды разработки виртуальных приборов, повсеместно используется для создания автоматизированных контрольно-измерительных и технологических систем в промышленности, научно-исследовательских центрах и учебных заведениях.

Виртуальные приборы (ВП) обычно состоят из компьютера, снабженного функциональными контрольно-измерительными модулями (например, плата ввода-вывода данных) и программного обеспечения, которые в комплексе образуют систему, идентичную традиционным приборам. Виртуальные приборы позволяют использовать всю вычислительную мощь, производительность, графические и сетевые возможности современных персональных и промышленных компьютеров. Определяющую роль в широком распространении технологии виртуальных приборов сыграли такие пионерские разработки National Instruments, как платформа графического программирования LabVIEW, система разработки приложений реального времени LabVIEW Real-Time, экспертная система для серийного контроля качества TestStand, а также многие другие, позволяющие инженерам и ученым создавать приборы, максимально точно отвечающие их нуждам.

Традиционные приборы, такие как осциллографы или генераторы сигналов, как правило, ориентированы на выполнение одной, в лучшем случае нескольких специфических задач, определяемых их производителем. Обычно пользователь не может существенно расширить функциональность своего оборудования. В свою очередь функциональность виртуальных приборов в основном определяется возможностями программного обеспечения, а значит, может быть легко адаптирована к конкретной задаче, решаемой инженером. Данная технология позволяет проводить адаптацию приборов без замены аппаратной части, используя возможности среды программирования, операционной системы и компьютерных технологий, что оказывается невозможным в случае использования традиционных приборов. Таким образом, именно гибкость технологии виртуальных приборов приводит к существенному расширению функциональности вашей системы при одновременном уменьшении ее стоимости.

**Программное обеспечение для виртуальных приборов.** Основой виртуального прибора является программное обеспечение, реализующее его функции. С помощью современных сред разработки виртуальных приборов, таких как LabVIEW, инженеры получают возможность создания приложений и пользовательского интерфейса, наилучшим образом ориентированных на выполнение какой-либо определенной задачи. Теперь именно инженер, по сути, конечный пользователь, определяет алгоритм осуществления сбора данных и синхронизацию, их обработку, представление и сохранение резуль-

татов.

Существенным преимуществом технологии виртуальных приборов, реализованной в LabVIEW, является ее модульность. В случае создания большой автоматизированной системы вы можете разработать отдельные виртуальные приборы для каждого из ее узлов, которые затем могут быть объединены в единый прибор, реализующий функции всей системы в целом. Подобная архитектура не только упрощает процесс разработки системы, но и позволяет проводить гибкую настройку каждого из ее узлов в отдельности посредством работы с соответствующим виртуальным прибором.

**LabVIEW – среда разработки виртуальных приборов.** На сегодняшний день LabVIEW является одной из наиболее популярных сред разработки виртуальных приборов, позволяющей ученым и инженерам осуществлять взаимодействие с большим набором оборудования и программного обеспечения. В LabVIEW вы имеете возможность размещения на лицевых панелях ваших виртуальных приборов (рис.10.2) таких элементов, как кнопки, переключатели, лампочки, ручки регулировки, циферблаты, графические панели и т.п., что позволяет воссоздавать внешний вид традиционных приборов.

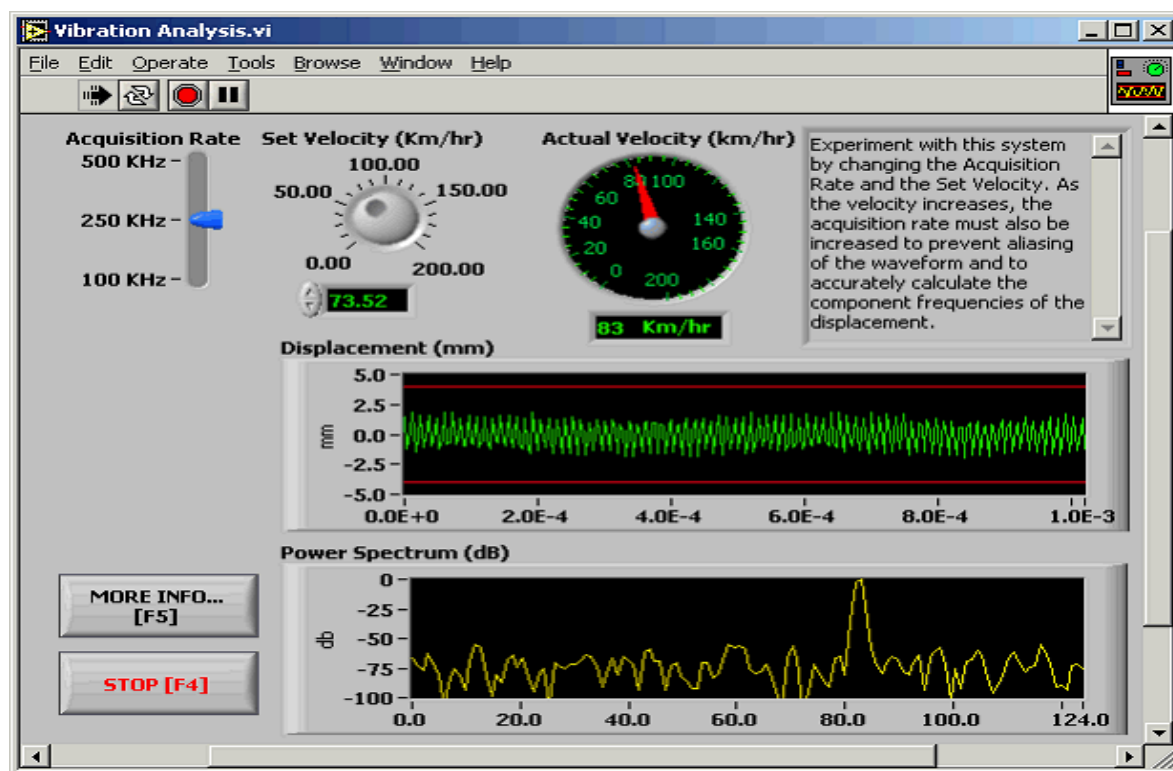


Рис.10.2. Лицевая панель виртуального прибора, созданного в среде LabVIEW

Логика работы виртуального прибора определяется созданной графической блок-диаграммой, каждый узел которой соответствует выполнению какой-либо функции или процедуры (рис.10.3). При этом представление программного кода в виде блок-диаграммы является интуитивно близким и понятным инженерам и ученым, привыкшим иметь дело с электронными схемами и т.п., а также позволяет осуществлять более гибкую и быструю разра-

ботку приложений.

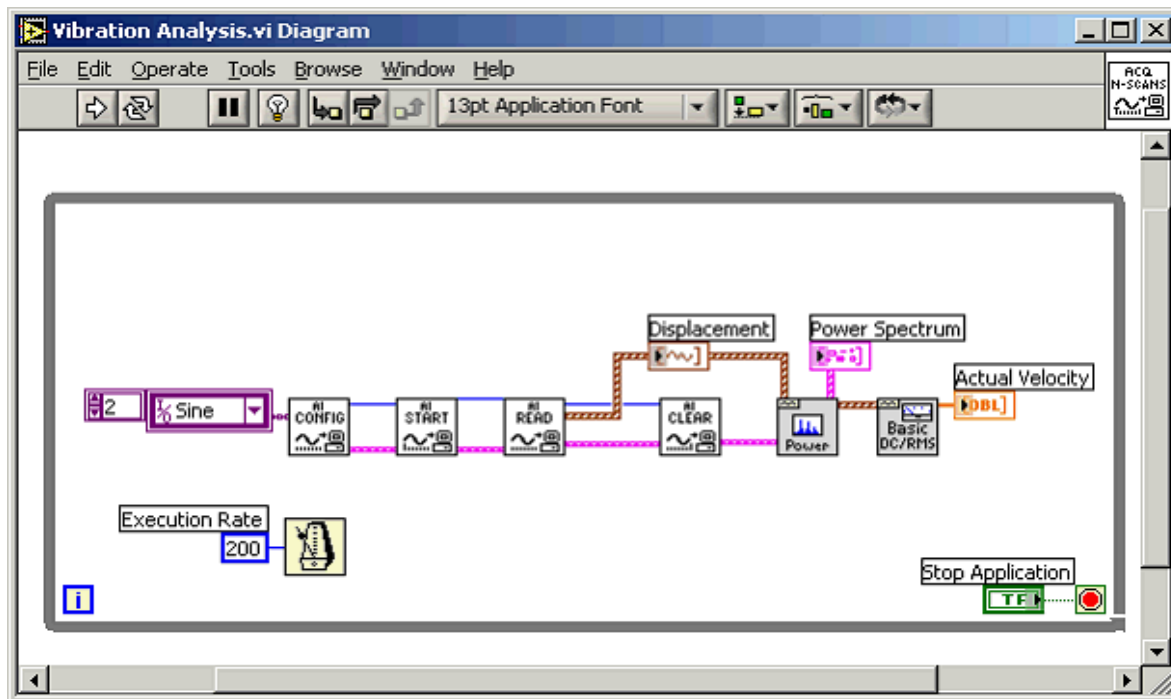


Рис.10.3. Блок-диаграмма виртуального прибора, созданного в среде LabVIEW

Среда LabVIEW, ориентированная на разработку виртуальных приборов, содержит в себе огромное количество инструментов, предназначенных для работы с различного рода устройствами и контрольно-измерительным оборудованием. Так, в состав среды входят библиотеки функций для работы с различными традиционными приборами, встроенными устройствами сбора данных, системами управления шаговыми и серво-двигателями, устройствами видеоввода, приборами, подключаемыми через интерфейсы GPIB/IEEE 488 и RS-232 и по шине USB, ПЛИС, системами реального времени и др. Кроме этого LabVIEW позволяет работать с VXI (VME eXtension for Instrumentation) и PXI (PCI eXtension for Instrumentation) устройствами, являющимися по сути инструментами, работающими на модифицированных шинах VME и CompactPCI.

LabVIEW является открытой средой программирования. Многие производители контрольно-измерительного оборудования поставляют свои устройства с соответствующими виртуальными приборами, написанными в среде LabVIEW. На интернет-сайте National Instruments (<http://www.ni.com>) можно найти более трех тысяч виртуальных приборов, созданных в среде LabVIEW и предназначенных для работы с приборами сторонних производителей. Кроме этого есть возможность работы в LabVIEW с различными устройствами, используя вызовы библиотек DLL, элементов ActiveX и кода написанного на C. Также в LabVIEW можно создавать свои библиотеки функций DLL, элементы ActiveX и EXE-файлы, которые могут быть вызваны в других средах разработки приложений. LabVIEW позволяет создавать виртуальные приборы, которые могут работать под всеми распространенными

операционными системами: все ОС семейства Windows, а также MacOS, Sun Solaris и Linux при полной переносимости кода. Кроме этого, можно создавать приложения, работающие под управлением операционной системы реального времени VenturCom ETS, установленной в устройствах реального времени производства National Instruments, а также приложения для ПЛИС.

С помощью LabVIEW создаются распределенные приложения, используя встроенные серверные инструменты. Данная особенность позволяет как создавать распределенные измерительные системы, так и распределять между компьютерами в сети сложные вычислительные процессы, а также осуществлять удаленный мониторинг и управление виртуальными приборами.

Для расширения функциональных возможностей виртуальных приборов LabVIEW содержит огромное количество инструментов, позволяющих осуществлять ввод и вывод сигналов, спектральный и корреляционный анализ, виброакустический анализ, статистический анализ, управление движением, анализ видео и графической информации и многое другое. Таким образом, с помощью LabVIEW на базе одного лишь устройства ввода-вывода можно реализовать множество приборов, таких как осциллографы, спектрометры, частотомеры, анализаторы импульсов, линии задержки, измерители частотного отклика систем, фильтры и т.п., что является несомненным преимуществом по сравнению с возможностями, предоставляемыми традиционными приборами.

**Виртуальные приборы в инженерных приложениях.** Использование виртуальных приборов предоставляет ряд преимуществ на всех этапах производственного технологического процесса, начиная от разработки и производства изделий и заканчивая контролем их качества. На этапе конструирования изделий инженерам приходится проводить тестирование прототипов и анализ полученных результатов в условиях постоянной модификации и модернизации объекта исследований. В этом случае использование виртуальных приборов позволит вам автоматизировать процедуру тестирования и существенно сократит время, затрачиваемое на разработку и адаптацию тестирующей системы, обеспечивая при этом высокую точность измерений.

Виртуальные приборы, разработанные в среде LabVIEW находят применение в науке и технике, во многих отраслях промышленности. Примерами таких приложений могут являться задачи проведения комплексных лабораторных измерений, тестирование электронных и механических систем автомобилей и летательных аппаратов, управление работой энергетических установок, тестирование систем уличного освещения, движения транспорта и даже установление места проведения оружейной стрельбы с точностью до десятков метров в масштабах города.

Технология виртуальных приборов находится в постоянном развитии как за счет изменений, происходящих в компьютерной технологии, так и за счет развития контрольно-измерительного оборудования.

Виртуальные приборы позволяют создавать гибкие и легко адаптируемые к новым условиям системы, которые как в настоящем, так и в будущем будут соответствовать требованиям надежности, точности и производитель-

ности. При этом на сегодняшний день наиболее мощным и надежным средством разработки виртуальных приборов является среда графического программирования LabVIEW производства компании National Instruments, предлагающая наиболее широкий спектр инструментов для работы с различного рода промышленным и контрольно-измерительным оборудованием.

National Instruments проводит и поддерживает ряд программ по внедрению современных компьютерных технологий в образовательную среду.

Оборудование National Instruments используется при проведении лабораторных практикумов, исследований, экспериментов и для дистанционного обучения. Более 100 учебных заведений России и стран ближнего зарубежья, а также российские производители учебной техники выбрали оборудование National Instruments и программное обеспечение для создания, модернизации и автоматизации учебных стендов и научных лабораторий.

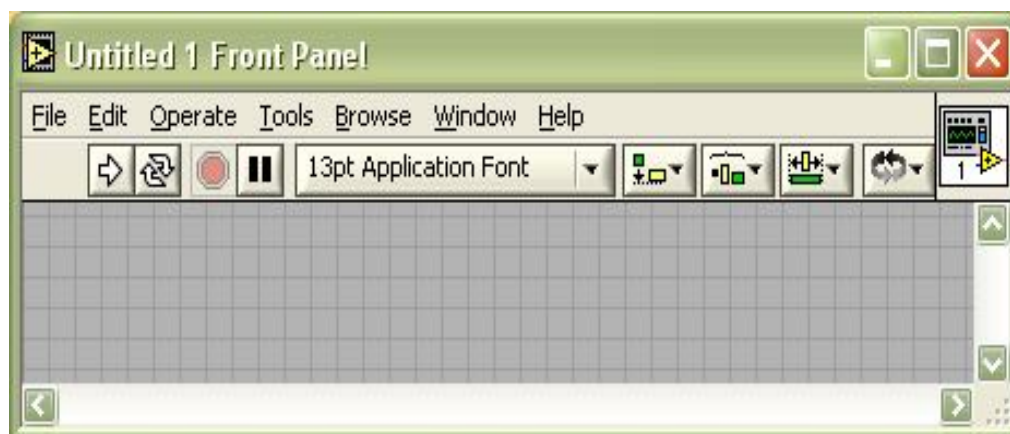
National Instruments предлагает ряд программ по оснащению вузов учебной техникой и программным обеспечением со скидками до 70%. Данные программы включают в себя:

- Оснащение вузов и их подразделений (факультетов, кафедр, лабораторий) программным обеспечением National Instruments
- Базовые практикумы по электротехническим дисциплинам
- Базовые практикумы по радиотехнике и связи
- Базовые практикумы по робототехнике, автоматическому управлению объектами и системами с обратной связью и проектирование с искусственным интеллектом
- Базовые многофункциональные системы сбора и обработки данных и лабораторные стенды
- Многоканальные системы виброакустической диагностики
- Контрольно-измерительные приборы, системы и лабораторные комплексы
- Лабораторные стенды для разработки систем промышленной автоматизации
- Базовые практикумы по фундаментальным дисциплинам.
- Дополнительные аксессуары для лабораторий и классов

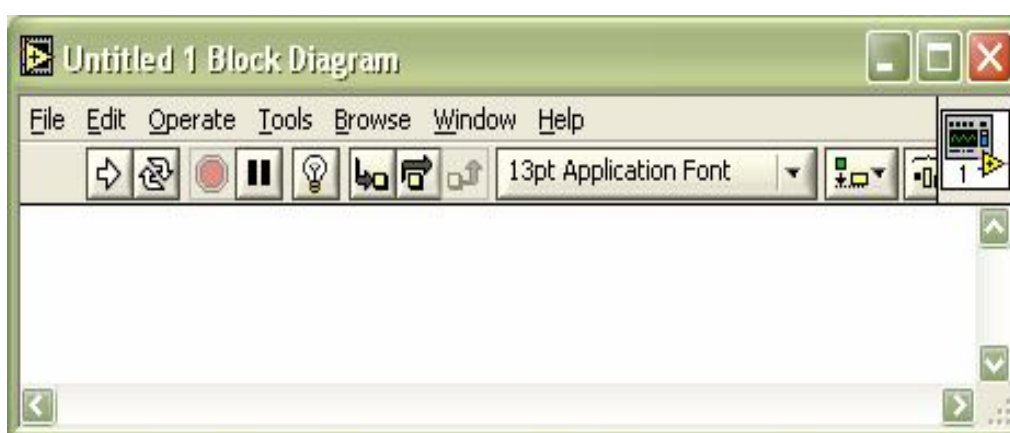
## 10.2. Панели и палитры LabVIEW

При запуске LabVIEW и выборе из меню стартового диалогового окна строки меню **Новый** => **Пустой ВП** (New => Blank VI) открываются два окна, содержащие лицевую панель (рис. 10.4а) и панель блок-диаграммы (рис. 10.4б) виртуального прибора (ВП). В правом верхнем углу каждой панели находится иконка, наложенная на соединительную панель ВП (последняя показана на лицевой панели). В верхней части каждого окна размещена традиционная для приложений Windows полоса главного меню с одинаковыми для обоих окон пунктами File, Edit, Operate, Tools, Browse, Windows и Help. Ниже полосы меню расположена полоса инструментальной панели, служащая для запуска и редактирования ВП. Полоса инструментальной панели окна блок-

диаграммы отличается дополнительными кнопками для отладки ВП.



а)



б)

Рис 10.4. Вид лицевой панели (а) и панели блок-диаграммы (б) виртуального прибора

Свободное пространство каждой панели образует рабочую область, снабженную горизонтальной и вертикальной полосами прокрутки. При построении ВП в рабочей области лицевой панели визуально размещаются элементы управления и индикации, формирующие интерфейс пользователя, а на панели блок-диаграммы составляется блок-диаграмма – графический исходный код ВП. Для одновременного отображения данных панелей в левой и правой половинах экрана целесообразно использовать меню **Окно => Панели слева и справа** (Windows => Tile Left and Right) или нажать «горячую» клавишу «Т». Клавиша становится «горячей» при нажатии одновременно с ней одной или более служебных клавиш. В данном случае должна быть нажата клавиша «Ctrl», далее такое сочетание обозначается <Ctrl-.Т>.

Построение ВП осуществляется с помощью трех вспомогательных палитр: палитры **Элементы управления** (Controls Palette), палитры **Функции** (Functions Palette) и палитры **Инструменты** (Tools Palette). Все перечисленные палитры можно вывести для постоянного или временного отображения и разместить в любом месте экрана. Вывод для постоянного отображения осу-

ществляется обычно с помощью разделов меню **Окно** (Window). Так, в частности, при активном окне лицевой панели с помощью строки **Показать палитру элементов управления** (Show Controls Palette) меню **Окно** (Window) на эту панель можно вывести палитру элементов, а при активном окне панели блок-диаграммы на нее можно вывести палитру функций, пользуясь строкой **Показать палитру функций** (Show Functions Palette) этого же меню. Для вывода палитры инструментов необходимо использовать строку **Показать палитру инструментов** (Show Tools Palette) меню **Окно** (Window).

Однако может оказаться, что пользователю будет более удобен временный вывод первых двух палитр, который реализуется как вызов контекстного меню каждой панели с помощью щелчка на ее рабочем пространстве правой кнопкой мыши (ПКМ). Выбор конкретного объекта из палитры элементов или палитры функций производится путем перемещения курсора мыши по разделам палитр. Выбранный объект берется из палитры с помощью щелчка левой кнопкой мыши (ЛКМ) и переносится в заданную область соответствующей панели, после чего фиксируется в этой области повторным щелчком ЛКМ (технология **Перенес и бросил** (Drag and Drop)). Эту же операцию можно выполнить с помощью щелчка ЛКМ на выбранном объекте, последующего удержания клавиши во время переноса объекта и отпускания клавиши в момент его фиксации. Такие объекты палитры функций, как **Структуры** (Structures), или строковые константы, перед фиксацией могут быть увеличены до необходимых размеров путем рисования модифицированным курсором мыши прямоугольного контура объекта при постоянно нажатой ЛКМ.

Временную версию палитры инструментов можно вывести с помощью щелчка ПКМ при нажатой клавише <Shift>.

Рассмотрим более подробно назначение пунктов главного меню, кнопок инструментальных панелей, палитр инструментов, элементов и функций.

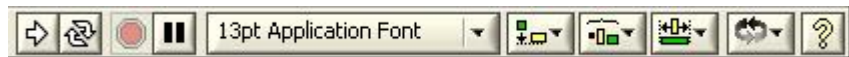
Выше были перечислены пункты главного меню, среди которых можно выделить пункты, встречающиеся в большинстве приложений Windows, такие как File, Edit, Windows, Help, и пункты, являющиеся характерными для LabVIEW, такие как Operate, Tools, Browse. Ниже в таблице приведено краткое описание функций пунктов главного меню.

<b>Файл</b> (File)	Используется для открытия новых или существующих ВП, закрытия, сохранения и печати ВП
<b>Правка</b> (Edit)	Применяется для редактирования панелей ВП, поиска объектов и удаления неисправных проводников с блок-диаграммы
<b>Управление</b> (Operate)	Реализует запуск и прерывание выполнения ВП, установку значений по умолчанию, соединение с удаленной панелью и изменение других опций ВП
<b>Инструменты</b> (Tools)	Используется для поиска ВП на диске, управления библиотеками ВП, управления соединением с удаленными ВП, управления публикацией панелей ВП в Web, кон-



	фигурации ВП и для выполнения ряда прикладных функций
<b>Просмотр</b> (Browse)	Используется для просмотра иерархии ВП
<b>Окно</b> (Window)	Используется для отображения окон LabVIEW и его палитр
<b>Справка</b> (Help)	Служит для получения информации об элементах и о функциях LabVIEW

Ниже показаны полосы инструментальных панелей на лицевой панели (рис. 10.5а) и на панели блок-диаграммы (рис. 10.5б).



а)



б)

Рис. 10.5. Инструментальные панели на лицевой панели (а) и на панели блок-диаграммы (б)

Все операции по созданию, редактированию и отладке ВП выполняются с помощью палитры **Инструменты** (Tools Palette) (рис. 10.6). При выборе определенного инструмента значок курсора мыши приобретает форму этого инструмента. При включенном автоматическом выборе инструмента наведение курсора на объект лицевой панели или блок-диаграммы LabVIEW приводит к автоматическому выбору соответствующего инструмента из палитры инструментов. Автоматический выбор инструментов включается нажатием кнопки **Автоматический выбор инструмента** (Automatic Tool Selection)



палитры инструментов или нажатием клавиш <Shift-Tab>.

Выбор любого другого инструмента приводит к отключению автоматического выбора инструмента. При этом можно циклически менять инструменты с помощью клавиши <Tab>. Для переключения между инструментом **Перемещение** и **Соединение** на блок-диаграмме или между инструментом **Перемещение** и **Управление** на лицевой панели достаточно нажать пробел.



Рис. 10.6

Пояснения по инструментам, входящим в палитру приведены в прило-



жении 3, раздел 10.2.1.

Палитра элементов лицевой панели (рис. 10.7) по умолчанию появляется в виде палитры Экспресс, содержащей наиболее часто используемые элементы. Для перехода к более привычному по предыдущим версиям LabVIEW виду палитры необходимо вызвать диалоговое окно **Опции** (Options) с помощью выбора одноименного раздела пункта **Инструменты** (Tools) главного меню или нажав кнопку **Option** на палитре элементов, а затем выбрать в верхнем раскрывающемся меню раздел **Палитры элементов управления или функций** (Controls/Functions Palettes) и выбрать в строке **Вид палитр** (Palette View) диалогового окна пункт **Advanced**.



Рис. 10.7



Рис. 10.8

В этом виде палитра элементов содержит подпалитры, приведенные в приложении 3, разделе 10.2.2.

Палитра функций панели блок-диаграммы (рис. 10.8.) по умолчанию также появляется в виде Экспресс и содержит ряд подпалитр Экспресс-ВП.

Переход к палитре с более широким набором функций на верхнем уровне осуществляется так же, как и описанный выше переход для палитры лицевой панели. В этом виде палитра функций панели блок-диаграммы содержит подпалитры, приведенные в приложении 3, раздел 10.2.2.

Из перечисленных подпалитр функций дополнительного рассмотрения, приведенных в приложении 3, разделе 10.2.3. заслуживает подпалитра **Анализ** (Analyze) как наиболее представительная.

Наиболее содержательными разделами этой подпалитры являются разделы **Обработка сигнала** (Signal Processing) и **Математика** (Mathematics).

Раздел функций **Обработка сигнала** включает подразделы **Генерация**

сигнала (Signal Generation), **Обработка сигнала во временной области** (Time Domain) и **Обработка сигнала в частотной области** (Frequency Domain), **Фильтры** (Filters) и **Окна** (Windows). Перечисленные подразделы функций рассмотрены соответственно в разделах 3.1.1-3.1.5 справочника.

Раздел функций **Математика** включает подразделы **Формула** (Formula), **Расчет одномерных и двумерных массивов** (1D and 2D Evaluation), **Численные методы и решение дифференциальных уравнений** (Calculus) (3.2.1), **Вероятность и статистика** (Probability and Statistics) (3.2.2), **Сглаживание данных** (Curve Fitting) (3.2.3), **Линейная алгебра** (Linear Algebra) и **операции с массивами** (Array Operations) (3.2.4), **Оптимизация** (Optimization) и **поиск нулей** (Zeroes) (3.2.5), а также **Специальные и числовые функции** (Special and Numeric Functions). После названий подразделов функций указаны номера разделов справочника, в которых эти функции рассмотрены более подробно.

Помимо разработки лицевой панели и блок-диаграммы самого ВП, важное значение имеет конфигурирование его входов-выходов и формирование графического представления ВП для последующего использования в других ВП в качестве подприбора (подпрограммы). Перечисленные функции выполняются с помощью иконки и соединительной панели, размещенных в правом верхнем углу панелей. Изображение, помещенное на иконке, придает разработанному ВП индивидуальность и в большинстве случаев несет информацию о его функциональном назначении. Соединительная панель определяет картину расположения входных и выходных терминалов, посредством которых происходит ввод и вывод данных при использовании ВП в качестве подпрограммы. Для того чтобы элементы лицевой панели могли обмениваться данными с ВП верхнего уровня, они должны быть подключены к терминалам соединительной панели.

В качестве примера показана иконка и соединительная панель ВП **Записать в файл табличного формата** (Write To Spreadsheet File), находящегося в подпалитре **Файловый ввод/вывод**.

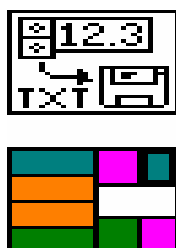


Рис.10.9

Вызов диалогового окна **Редактор иконки** (Icon Editor) для редактирования изображения иконки осуществляется с помощью строки **Редактировать иконку** (Edit Icon) контекстного меню иконки на лицевой панели. Вызов функций редактирования соединительной панели возможен с помощью строки **Показать соединительную панель** (Show Connector) того же меню.

### 10.3. Функции в системе LabView

В системе LabView предусмотрены числовые, логические, строковые функции, функции сравнения, функции работы с массивами и кластерами, функции времени и диалогов, функции и ВП ввода/вывода файлов.

Рассмотрим для примера числовые функции, которые используются для выполнения арифметических, тригонометрических, логарифмических и комплексных операций с числовыми данными и для преобразования типов числовых данных.

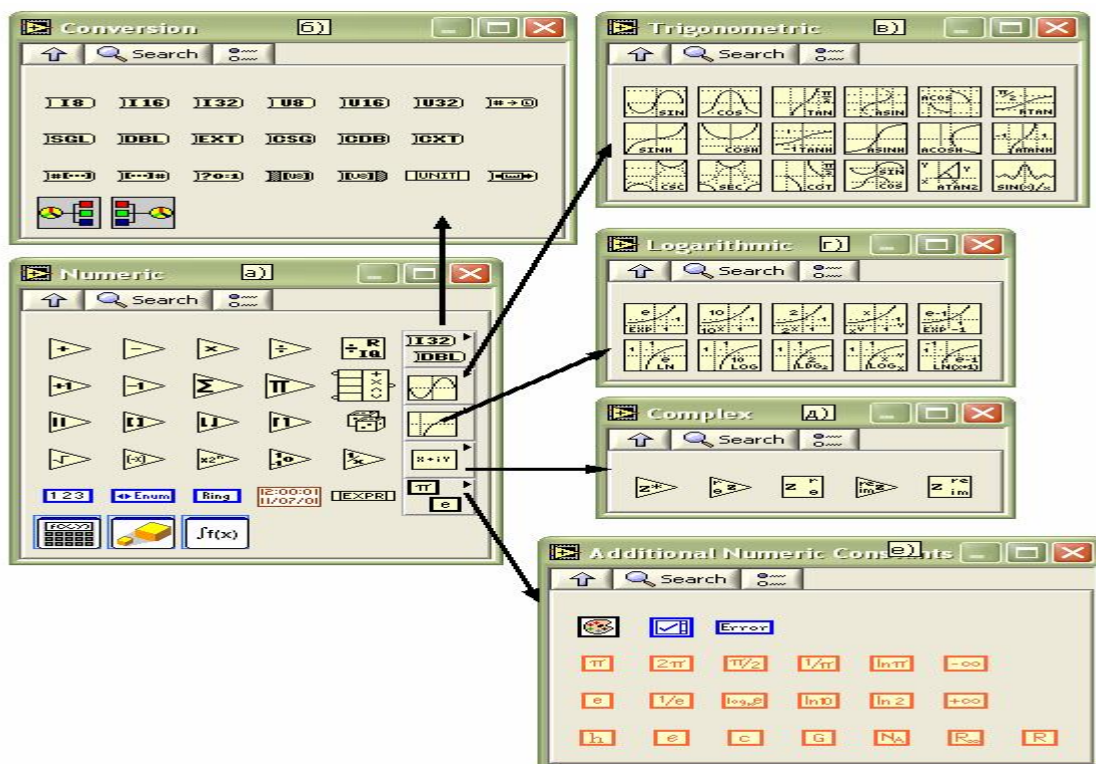


Рис. 10.10. Вид основной палитры (а)

и дополнительных подпалитр (б-е) числовых функций

На рис.10.10а показан вид основной палитры числовых функций и ряда дополнительных подпалитр: **Преобразование** (Conversion) (рис. 10.10б), **Тригонометрические функции** (Trigonometric) (рис. 10.10в), **Логарифмические функции** (Logarithmic) (рис. 10.10г), **Комплексные функции** (Complex) (рис. 10.10д) и **Дополнительные числовые константы** (Additional Numeric Constants) (рис. 10.10е).

В приложении 3 раздел 10.2.4 – 10.2.10 приведены таблицы этих функций и схемы их использования.

В состав палитры числовых функций входят Экспресс-ВП **Формула** (Formula), **Масштабирование и отображение** (Scaling and Mapping) и **Математическая обработка во временной области** (Time Domain Math).

### 10.3.1. Формула (Formula)

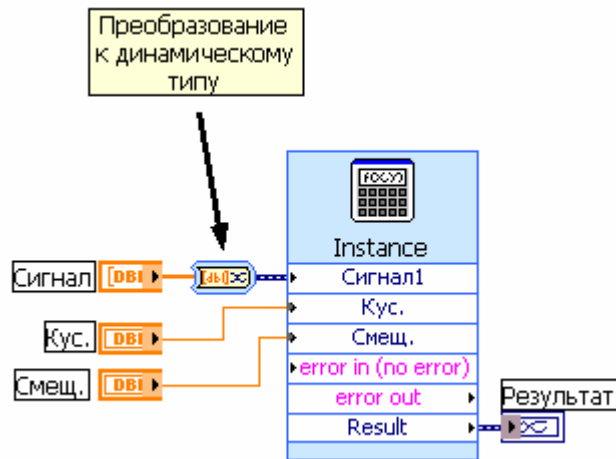



Рис. 10.11. Блок-диаграмма варианта подключения Экспресс-ВП



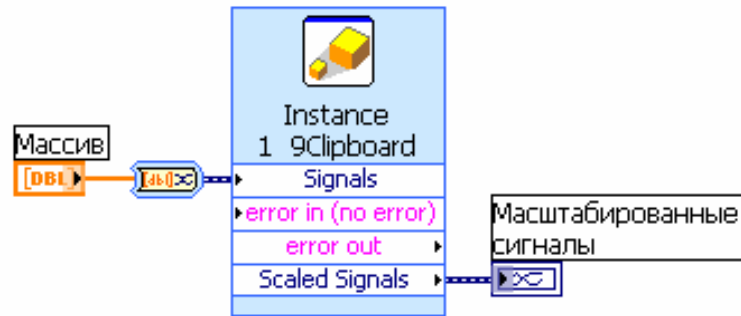
Рис. 10.12. Вид диалогового окна конфигурирования Экспресс-ВП **Формула** (Formula)

Экспресс-ВП **Формула** (Formula) (рис. 10.12) позволяет проводить математическую обработку входных данных, обеспечиваемую базовыми научными калькуляторами.

Входы и выходы Экспресс-ВП рассчитаны на подключение данных динамического типа. При подключении данных другого типа производится преобразование типов, в том числе и с помощью Экспресс-ВП **Преобразование в данные динамического типа** (Convert to Dynamic Data)  (рис. 10.11).

В Экспресс-ВП **Формула** используется функциональность следующих ВП: **Узел Выражение** (Expression Node), **Узел Формула** (Formula Node), **Оценка узла формулы** (Eval Formula Node), **Оценка строки формулы** (Eval Formula String), **Оценка массива значений параметрической функции** (Eval Multi-Variable Array).

### 10.3.2. Масштабирование и отображение



а)



б)

Рис. 10.13. Блок-диаграмма (а) и вид диалогового окна конфигурирования (б) Экспресс-ВП

Экспресс-ВП **Масштабирование и отображение** (Scaling and Mapping) (рис. 10.13) изменяет диапазон и характер отображения входных данных с помощью их масштабирования и выбора вида преобразования. Содержит следующие опции:

**Нормализовать** (Normalize) определяет масштаб и смещение, необходимые для преобразования данных так, чтобы их максимум был равен значению **Верхний пик** (Highest peak), а минимум - **Нижний пик** (Lowest peak).

**Линейное** (Linear ( $Y=mX+b$ )) устанавливает линейное преобразование значений.

**Логарифмическое** (Logarithmic) устанавливает логарифмическое преобразование входных значений. При этом параметр **начальное значение, дБ** (db reference) определяет начальную точку шкалы в децибелах.

**Интерполяционное** (Interpolated) определяет шкалу, основанную на



таблице значений, которые интерполируются линейно в целях получения масштабного параметра. Таблица значений задается в диалоговом окне **Определить сигнал** (Define Signal), которое вызывают с помощью кнопки **Определить таблицу** (Define Table). В Экспресс-ВП используется функциональность ВП **Масштаб и смещение осциллограммы** (Waveform Scale and Offset).

### 10.3.3. Математическая обработка во временной области (Time Domain Math)

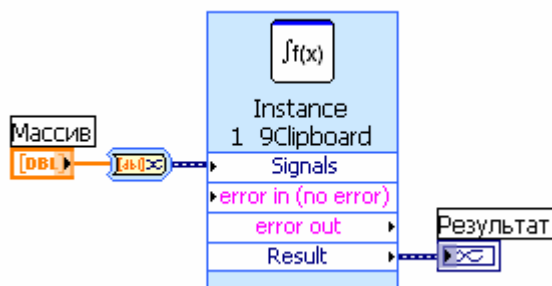


Рис. 10.14. Блок-диаграмма подключения Экспресс-ВП

Экспресс-ВП **Математическая обработка во временной области** (Time Domain Math) (рис. 10.14) выполняет одну из операций математической обработки сигналов во временной области.

Набор математических операций включает следующие (рис. 10.15): **Дифференциал** (Differential), **Разность** (Difference), **Интеграл** (Integral) и **Суммирование** (Summation). В Экспресс-ВП используется функциональность ВП **Производная  $x(t)$**  (Derivative  $x(t)$ ) и **Интеграл  $x(t)$**  (Integral  $x(t)$ ).

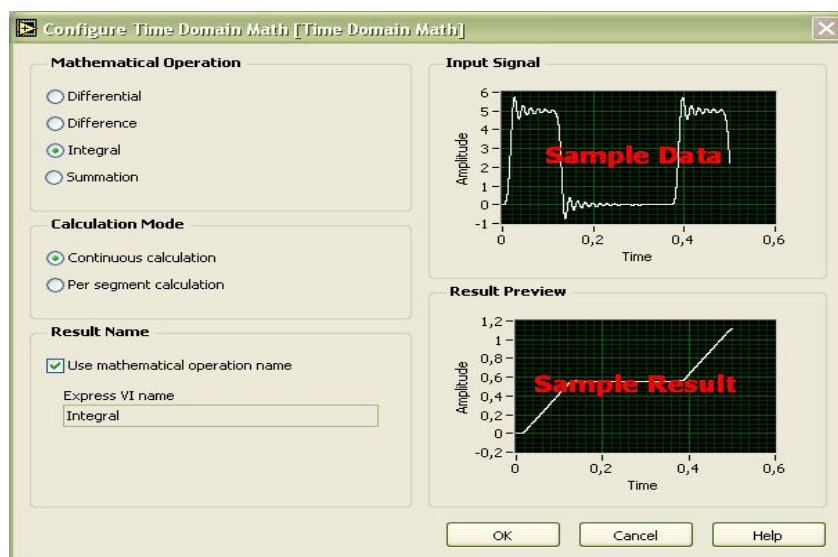


Рис. 10.15. Вид диалогового окна конфигурирования Экспресс-ВП **Математическая обработка во временной области** (Time Domain Math)

## 10.4. Технология проектирования

## виртуальных приборов


Для проектирования ВП в среде LabVIEW необходимо сформировать его лицевую панель и разработать блок-диаграмму. При формировании лицевой панели осуществляется выбор и установка на ней элементов управления и индикации из палитры элементов данной панели. Аналогично при разработке блок-диаграммы проходит выбор и установка на ней функциональных элементов и подприборов из палитры функций данной панели.


Установка каждого элемента на лицевой панели сопровождается появлением соответствующего **терминала** данных (terminal) на панели блок-диаграммы. Терминалы элементов управления представляют порты ввода информации в блок-диаграмму, а терминалы индикаторов – порты вывода информации из блок-диаграммы на лицевую панель (приложение 3, раздел 10.4.1). Для обработки введенной информации и программного управления параметрами и режимами работы элементов лицевой панели на панели блок-диаграммы размещаются необходимые константы, **функции** (Functions), **подприборы** (SubVI) и **структуры** (Structures), которые также имеют терминалы для ввода и вывода информации. Все перечисленные элементы представляют узлы (nodes) блок-диаграммы, которые соединяются с терминалами элементов управления и индикации и между собой линиями, называемыми **проводниками** (wires). В такой схеме через узлы в процессе обработки проходит **поток данных** (data flow), идущий по проводникам от входных терминалов к выходным. Узлы – это объекты на блок-диаграмме, которые имеют одно или более полей ввода/вывода данных и выполняют алгоритмические операции ВП. Они аналогичны операторам, функциям и подпрограммам текстовых языков программирования.

Таким образом, описанная технология формирования ВП является основой для **поточковой модели** обработки данных, когда поток данных входит (втекает) в узлы-источники, проходит через узлы обработки данных и выходит (вытекает) через узлы-приемники данных. При этом порядок обработки данных определяется целиком полнотой подхода данных к терминалам узлов. Такая концепция работы программы в LabVIEW существенно облегчает, по сравнению с текстовыми языками, разработку **многозадачных** и **многопоточных программ**.

Терминалы данных имеют прямоугольную форму и содержат буквенно-графическое обозначение, характеризующее тип и форму представления воспринимаемых ими данных. Таким образом, по виду терминала можно определить, является ли он источником или приемником данных, какие типы данных он воспринимает – числовые, логические или строковые, а для числовых является ли число целым или вещественным. Для определения подобных различий используются различия в толщине внешней рамки терминала и направлении треугольной стрелки внутри него, цвет терминала и буквенное или графическое обозначение. Кроме того, вид терминала можно определить и по содержанию контекстного меню.

В LabVIEW 7.0 введен еще один способ идентификации терминала, а

именно отображение иконки соответствующего элемента. Такое отображение включается с помощью опции **Отображать в виде иконки** (View as Icon) контекстного меню терминала. В качестве примера можно сравнить вид терминала числового элемента ввода данных в форме с плавающей запятой одинарной точности в его традиционном представлении  и в представлении

в виде иконки .

Однако представление в виде иконки приводит к потере пространства блок-диаграммы, что не всегда удобно.

Перечень и вид терминалов элементов ввода данных LabVIEW 7.0 с характеристикой типа, цвета и значения по умолчанию приведены в приложении 3, раздел 10.4.1.

Настройка параметров объектов лицевой панели (элементов управления и индикаторов) и терминалов блок-диаграммы производится с помощью контекстного меню опций и команд. Контекстное меню открывается с помощью щелчка ПКМ на объекте. Опции, входящие в состав контекстного меню, зависят от типа объекта. Вместе с тем в состав контекстного меню многих объектов входит ряд одинаковых пунктов. Сочетание общего и индивидуального можно показать на примере контекстных меню числового элемента управления (слева) и его терминала (рис. 10.16).

Краткие пояснения к пунктам контекстного меню приведены в приложении 3, раздел 10.4.2.

Для логических элементов управления специфическим является пункт контекстного меню **Механическое действие** (Mechanical Action), определяющий характер срабатывания логического элемента при его нажатии. Варианты срабатывания приведены в приложении 3 раздел 10.4.3.

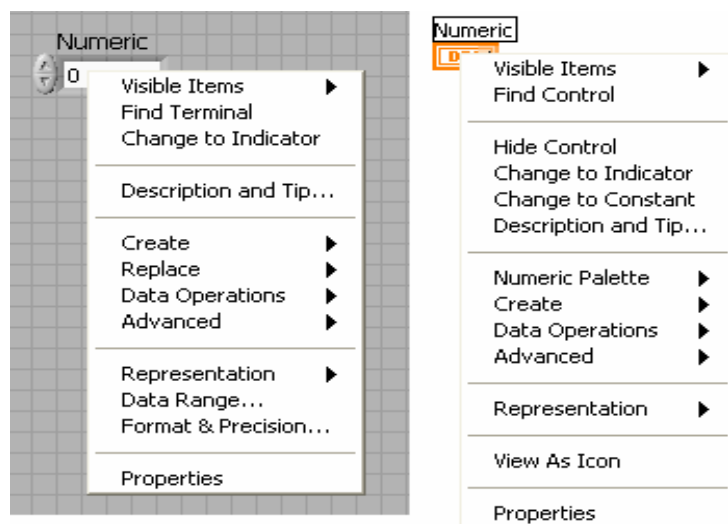





Рис. 10.16

В меню настроек строковых элементов управления и индикаторов специфическим является раздел, определяющий вид отображения содержимого этих элементов. Интерпретация строк данного раздела меню приведена в



приложении 3, раздел 10.4.4.

Наряду с терминалами элементов управления и отображения данных выделяют терминалы узлов, которые представляют поля ввода/вывода данных. Количество и тип терминалов узлов на панели блок-диаграммы можно определить по **соединительной панели** (connector panes) узла путем выбора строки контекстного меню **Видимые элементы => Терминалы** (Visible Items => Terminals). Так, например, схема терминалов функции **Сложить** (Add)  выглядит следующим образом: . Выходы на соединительной панели функции выделяются более толстым контуром.

При подводе инструмента соединения к терминалу узла на всех его выводах появляются короткие отрезки проводов (stubs), а выбранная область терминала начинает мерцать: . При этом цвет и толщина отрезков проводов характеризуют тип передаваемых через них данных.

Каждое соединение должно быть согласовано по типу выводов и по типу передаваемых и принимаемых данных. Согласование по типу выводов включает следующие аспекты:

- к линии передачи данных должен быть подключен только один источник;
- к одному источнику может быть подключено неограниченное число приемников данных;
- не допускается соединение одних приемников.

Нарушение правил соединения проявляется в сохранении пунктирной соединительной линии после окончания соединения и в разрыве кнопки запуска программы. Такое состояние в ряде случаев может быть легко устранено, например, путем соединения одних приемников с источником данных. В других случаях может потребоваться анализ типов соединяемых выводов элементов.

Перемещают и удаляют провода так же, как и любой другой объект блок-диаграммы. При этом одиночный щелчок инструментом перемещения выделяет прямолинейный участок провода (сегмент), двойной щелчок выделяет весь провод с изгибами до первого узла (ветвь), тройной щелчок выделяет все провода разветвленного соединения. Удаление всех поврежденных (broken) проводов осуществляется с помощью выбора опции **Удалить поврежденные проводники** (Remove Broken Wires) в меню **Правка** или с помощью «горячей» клавиши <Ctrl-B>.

При сборке блок-диаграммы ВП, помимо типа, необходимо знать функциональное назначение всех выводов элементов, входящих в состав прибора. Информацию о назначении выводов можно получить с помощью окна **Контекстная справка** (Context Help), вызываемого с помощью опции **Справка => Показать контекстную справку** (Help => Show Context Help), или с помощью инструмента соединения, подводимого к выводам. Назначение выводов подприбора можно определить, открыв его лицевую панель с помощью двойного щелчка на иконке подприбора или с помощью строки

**Открыть лицевую панель** (Open Front Panel) контекстного меню.

Состав и схему расположения терминалов самого ВП на соединительной панели можно просмотреть, с помощью выбора строки **Показать соединительную панель** (Show Connector) контекстного меню иконки ВП, находящейся в правом верхнем углу лицевой панели. Если после этого инструментом соединения последовательно выделять щелчком терминалы на соединительной панели, то на лицевой панели при этом пунктирной линией будут выделяться связанные с данным терминалом элементы. В режиме просмотра соединительной панели с помощью контекстного меню данной панели **Этот вывод является...** (This Connection Is...) можно определить тип вывода – **Обязательный** (Required), **Рекомендуемый** (Recommended) или **Необязательный** (Optional). Подписи входов указанных типов отличаются форматированием текста в окне контекстной справки. Далее в справочнике при описании функций подписи к обязательным входам на рисунках функций выделены полужирным шрифтом, а текст пояснений к рекомендуемым и необязательным входам отличается уменьшенным размером шрифта.

Можно провести и обратное действие – связать элемент лицевой панели с выводом терминала соединительной панели. Потребность в таких действиях возникает при формировании нового подприбора, когда необходимо определить его входы и выходы, то есть связать элементы управления и индикаторы с входными и выходными терминалами соединительной панели.

Сам новый подприбор может быть сформирован из фрагмента блок-диаграммы, разработанного ВП путем очерчивания инструментом перемещения прямоугольника, охватывающего элементы, которые должны быть включены в состав подприбора, и выбора строки меню **Правка => Создать подприбор** (Edit => Create SubVI). Если после формирования подприбора сохранить его в папке библиотеки пользователя (User.lib), то при новом запуске LabVIEW данная подпрограмма будет доступна в подпалитре **Библиотеки пользователя** (User Libraries) палитры функций панели блок-диаграммы.

Сохранение ВП возможно с помощью одной из опций меню **Файл**. Опция **Сохранить** (Save) позволяет оставить измененный ВП с сохранением имени и местоположения на диске или же новый ВП с указанием этих параметров в диалоговом окне. Опция **Сохранить как** (Save as) позволяет изменить имя ВП или его местоположение, при этом исходная версия ВП на диске не изменяется. При выборе опции **Сохранить с опциями** (Save with Options) выводится диалоговое окно, в котором пользователь может выбрать ряд вариантов.

Если сохранение ВП сопровождается вызовом диалогового окна, то пользователь может выбрать вариант сохранения в виде индивидуального файла или в составе библиотеки ВП.

Для отладки ВП в LabVIEW предусмотрены такие средства, как информационное окно **Список ошибок** (Error List), выводимое при нажатии неисправной кнопки **Запуск** (Run), кнопки пошагового выполнения, подсветка при выполнении программы (Execution Highlighting), инструменты **Пробник** (probe) и **Точки останова** (breakpoints).

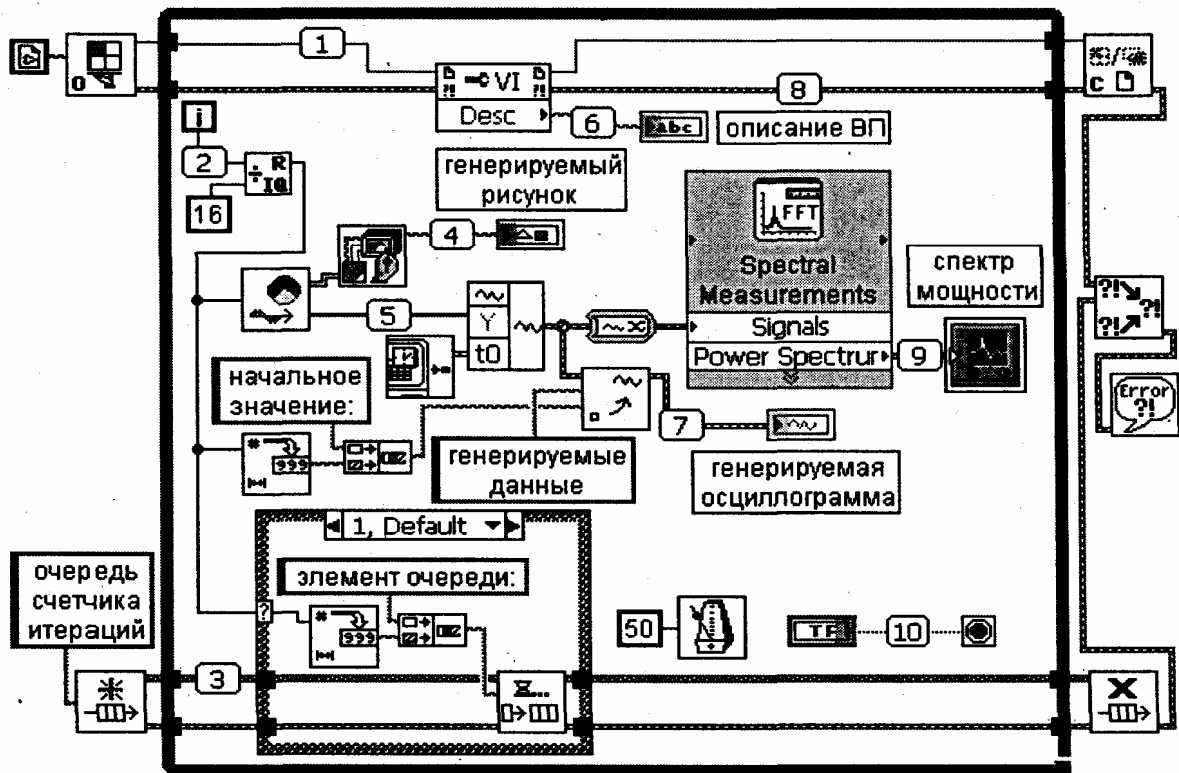


Рис. 10.17. Вид блок-диаграммы ВП **Использование вставляемых пробников** (Using Supplied Probes)

На рис. 10.17 показан вид блок-диаграммы ВП **Использование вставляемых пробников** (Using Supplied Probes) из набора примеров N1 Example Finder LabVIEW.

Установка пробников проводится с помощью выбора опций Probe или Custom Probe контекстного меню провода.

Ниже даны пояснения к пробникам различного типа.

1. **Пробник ссылки ВП (VI Refnum Probe)** показывает путь ВП, путь владельца и числовую ссылку.
2. **Пробник числа в форме 132 с условием (Conditional Signed32 Probe)** показывает числовые скалярные данные и обеспечивает останов по условию.
3. **Пробник ссылки очереди строк (String Queue Refnum Probe)** показывает различную информацию об очереди, такую, как имя и вставленные элементы.
4. **Пробник рисунка (Picture Probe)** показывает рисунок.
5. **Пробник массива чисел в форме DBL с условием (Conditional Double Array Probe)** показывает числовой массив данных и обеспечивает остановку по условию.
6. **Пробник строки с условием (Conditional String Probe)** - показывает строковые данные и обеспечивает останов по условию.
7. **Пробник аналоговой осциллограммы (Analog Waveform Probe)** - показывает график осциллограммы, набор данных и атрибуты осциллограммы.
8. **Пробник ошибки с условием (Conditional Error Probe)** показывает статус ошибки, код и описание. Обеспечивает остановку по условию

**9. Пробник данных динамического типа** (Dynamic Data Type Probe) показывает график динамических данных, набор данных и атрибуты.

**10. Пробник логических данных с условием** (Conditional Boolean Probe) показывает логические данные и обеспечивает остановку по условию.

### 10.5. Структуры, массивы и графические индикаторы среды LabVIEW

В составе палитры **Структуры** (Structures) среды LabVIEW можно выделить две группы структур, имеющих ряд общих черт. Первая группа – это структуры **Цикл с фиксированным числом итераций** (For Loop) и **Цикл по условию** (While Loop), вторая группа структура **Вариант** (Case Structure), структура **Стековая последовательность** (Stacked Sequence Structure) и структура **Открытая последовательность** (Flat Sequence Structure). Именно в таком сочетании происходит взаимная замена данных структур при выборе строки **Заменить на...** (Replace with...) из контекстного меню структуры.

Помимо перечисленных, в состав палитры входят также структуры: узел **Формула** (Formula Node), структура **Событие** (Event Structure), **Локальная переменная** (Local), **Глобальная переменная** (Global) и узел **Обратная связь** (Feedback Node).

Рассмотрим более подробно структуры LabVIEW в порядке их упоминания в предыдущих абзацах.

Структура **Цикл с фиксированным числом итераций** (For Loop) эквивалентна текстовому оператору for i = 0 to N - 1 do ...

При помещении структуры на панель блок-диаграммы ее контур в виде прямоугольника должен быть растянут так, чтобы охватить существующий код программы, который должен выполняться циклически заданное число раз, или так, чтобы позволить разместить в нем новый код программы. Если помещаемая в структуру или перемещаемая внутри структуры функция пересекается с ее границей, то граница автоматически расширяется. Данная опция может быть отключена для этой структуры путем снятия отметки строки Auto Grow в контекстном меню структуры или для всего приложения путем снятия отметки строки **Установить структуры с автоматическим расширением** (Place structures with Auto Grow enabled) диалогового окна **Опции** (Options) => **Новое и измененное в LabVIEW 7.0** (New and Changed in LabVIEW 7.0).

Количество циклов может задаваться с помощью константы или элемента управления, подключенных к **терминалу числа итераций** (count terminal) (прямоугольник в левом верхнем углу структуры с буквой N). Текущее число завершенных итераций цикла содержится в **терминале счетчика итераций** (iteration terminal).

В структуре цикла для передачи данных из одной итерации цикла в следующую могут быть установлены **Сдвиговые регистры** (Shift Register) (CP) (рис. 10.18).

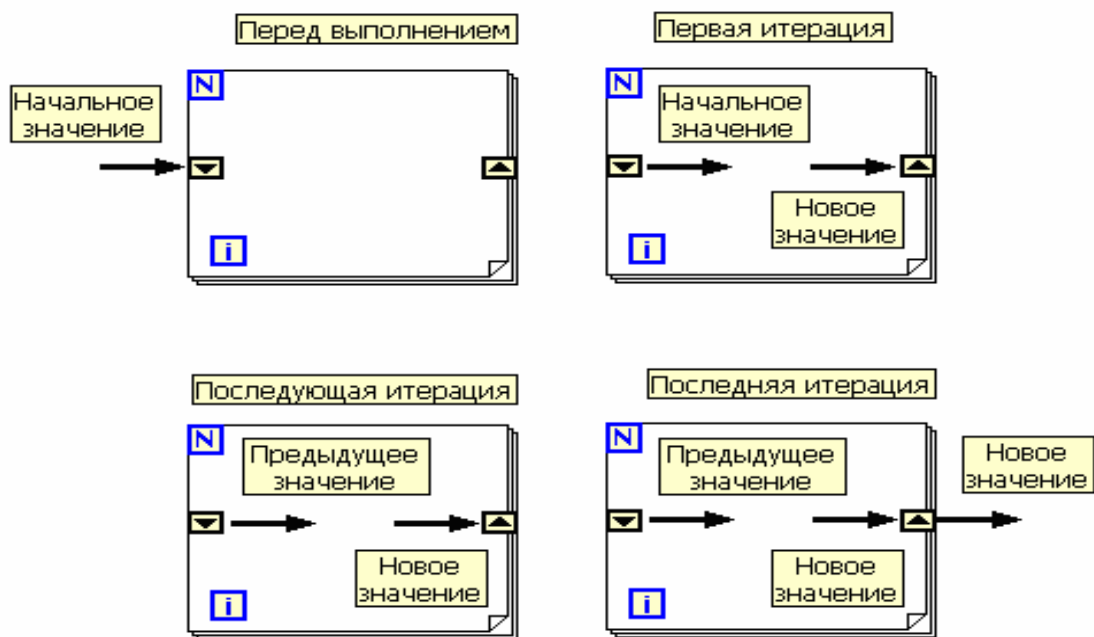


Рис. 10.18. Принцип работы сдвиговых регистров

Установка СР проводится на вертикальных сторонах структуры с помощью опции **Добавить сдвиговый регистр** (Add Shift Register), выбираемой из контекстного меню структуры при щелчке правой кнопкой мыши на одной из этих сторон. Тип и размерность СР определяются типом и размерностью данных, подключаемых к правому (входному) терминалу СР. Количество разрядов (тактов задержки) левого терминала СР можно изменить с помощью инструмента перемещения или с помощью опций **Добавить элемент** (Add Element), **Удалить элемент** (Remove Element) контекстного меню структуры. По умолчанию при первом запуске ВП начальные значения левых терминалов СР равны значению по умолчанию для типа данных регистра. Если эти терминалы не были подключены, то при последующих запусках в них будут находиться значения, оставшиеся от предыдущих запусков ВП. Для инициализации левых терминалов СР к ним необходимо подключить константу, элемент управления или функцию инициализации массивов.

Единичные СР могут быть преобразованы в тоннели с помощью опции **Заменить тоннелем** (Replace with Tunnel). После замены тоннель на правой вертикальной границе структуры цикла будет представлять терминал выхода данных из цикла, а тоннель на левой границе терминал входа данных. Терминал выхода данных, по умолчанию находящийся в состоянии **Включить индексирование** (Enable Indexing), служит точкой, в которой происходит накопление данных и повышение их размерности при их передаче за пределы структуры цикла после окончания ее выполнения. Так, в частности, скалярные данные преобразуются в одномерный массив, одномерный массив в двумерный массив и т. д. При выборе строки **Отключить индексирование** (Disable Indexing) контекстного меню терминала повышения размерности данных не происходит и за пределы структуры цикла передается последний элемент данных (рис. 10.19). Более подробно индексирование входных и вы-

ходных данных структуры цикла будет рассмотрено ниже при описании массивов.

Удаление структуры цикла без удаления содержащегося в ней кода проводится с помощью строки **Удалить цикл с фиксированным числом итераций** (Remove For Loop) контекстного меню структуры. Для удаления структуры цикла с содержимым необходимо, как обычно, выделить ее инструментом перемещения.

Примеры ВП с использованием структуры цикла с фиксированным числом итераций и сдвиговых регистров приведены на рис. 10.19 и 10.20.

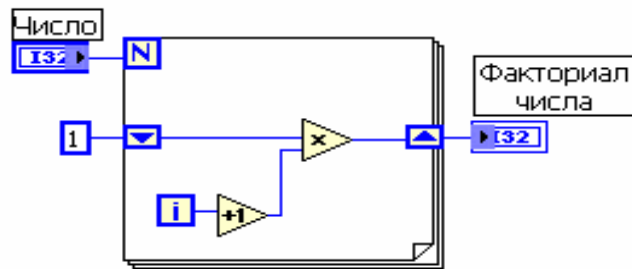


Рис. 10.19. Блок-диаграмма ВП вычисления факториала числа

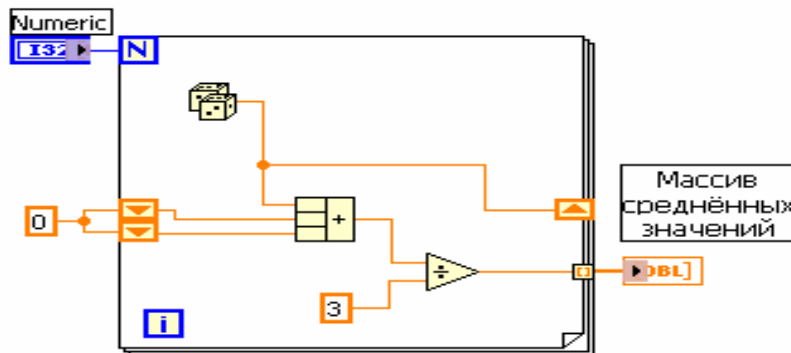


Рис. 10.20. Блок-диаграмма ВП усреднения последовательности случайных чисел

Помимо сдвигового регистра, в структуре цикла может устанавливаться структура узел **Обратная связь** (Feedback Node) (рис. 10.21). Узел **Обратная связь** автоматически появляется в циклах с фиксированным числом итераций и в циклах по условию при соединении поля вывода данных подприбора, функции или группы подприборов и функций с полем ввода тех же самых подприборов, функций или их групп. Узел **Обратная связь** может быть заменен сдвиговым регистром с помощью строки **Заменить сдвиговым регистром** (Replace with Shift Register) контекстного меню узла. Пример применения узла **Обратная связь** для формирования возрастающих знакопеременных чисел приведен на рис. 10.21.

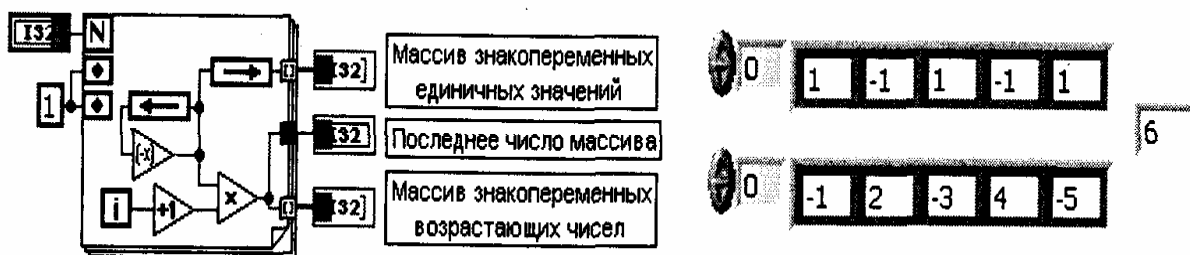


Рис. 10.21. Блок-диаграмма и лицевая панель ВП расчета знакопеременных чисел

Структура **Цикл по условию** (While Loop) эквивалентна следующему псевдокоду:

**do** {программа} **while** {условие} ...

Внутри структуры размещаются **терминал счетчика итераций** (iteration terminal) **i** и **терминал условия выхода из цикла** (conditional terminal). Вид структуры с терминалом условия по умолчанию приведен на рис. 10.22а. Код программы, размещенный в структуре, выполняется до подачи на терминал условия логической переменной ИСТИНА (TRUE) в структуре на рис. 10.22а или ЛОЖЬ (FALSE) в структуре на рис. 10.17б. Изменение варианта прекращения выполнения проводится с помощью строк **Остановить если истина** (Stop If True) или **Продолжить если истина** (Continue If True) контекстного меню терминала условия. Если терминал условия не подключен к какому-либо выходу, то цикл не будет выполняться. В данной структуре также могут быть установлены сдвиговые регистры и структуры узлов обратной связи. Входные и выходные терминалы данной структуры по умолчанию имеют состояние **Индексирование выключено** (Disable Indexing).

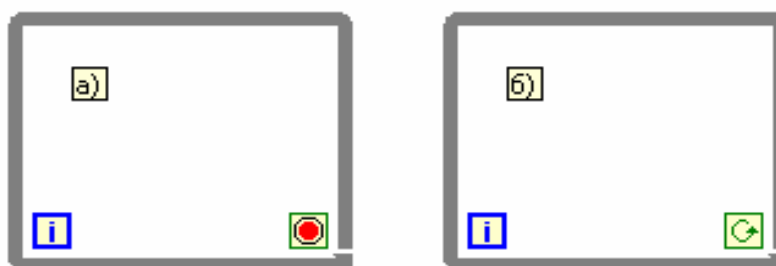


Рис. 10.22. Вид структуры **Цикл по условию** с различными режимами

Структура **Вариант** (Case Structure) аналогична операторам case или if-then-else в текстовых языках программирования. По умолчанию структура Вариант является логической и имеет два варианта: ИСТИНА (TRUE) и ЛОЖЬ (FALSE), выбираемые с помощью **терминала селектора структуры варианта**. Структура автоматически преобразуется в числовую или строковую при подключении соответственно числовой или строковой переменной к терминалу селектора. В этом случае структура может иметь практически неограниченное количество вариантов, начиная с нулевого. С помощью строк **Добавить вариант после** (Add Case After) или **Добавить вариант перед**



(Add Case Before) можно добавить новый вариант после или до текущего варианта. Одновременно можно наблюдать только один вариант (кадр) структуры. Переход между вариантами осуществляется с помощью **селектора структуры варианта**, расположенного на верхней стороне рамки структуры или контекстного меню структуры. Для использования структуры **Вариант** необходимо отметить **вариант по умолчанию** (Default).

Ввод и вывод данных в структуру **Вариант** проводится с помощью входных и выходных терминалов данных (тоннелей). Создание выходного терминала данных на одной поддиаграмме структуры приводит к его появлению на других поддиаграммах в том же самом месте границы структуры. До подключения данных к выходному терминалу во всех поддиаграммах он сохраняет белый цвет и воспринимается как ошибка создания структуры. Выходные терминалы должны иметь значения совместимых типов.

Структура **Последовательность** (Sequence Structure) используется для управления порядком выполнения узлов данных, которые не зависят друг от друга. Структура Последовательность выглядит как набор кадров и обеспечивает последовательное выполнение размещенных в ее кадрах фрагментов программ. Необходимость в такой структуре вызвана потоковым характером выполнения программ в LabVIEW, когда операции в узлах выполняются при поступлении данных на все входы узлов. При необходимости выполнения программы в ином порядке и используется структура последовательности.

В Lab VIEW 7.0 структура **Последовательность** наряду с известной по предыдущим версиям структурой **Стековая последовательность** (Stacked Sequence Structure) дополнена представлением в виде структуры **Открытая последовательность** (Flat Sequence Structure). Отличие указанных структур проявляется при увеличении числа кадров с помощью строк **Добавить кадр после** (Add Frame After) или **Добавить кадр перед** (Add Frame Before) (рис. 10.23).

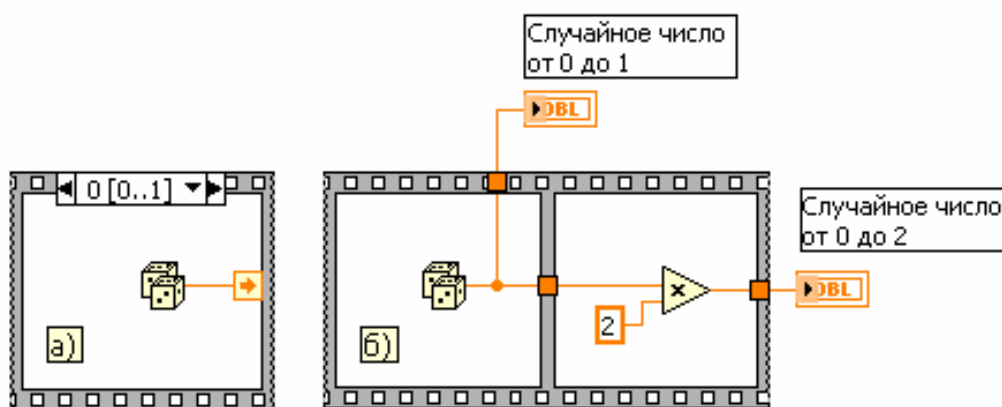


Рис. 10.23. Вид структуры Стековая последовательность (а) и структуры Открытая последовательность (б)

Так же как и в структуре **Вариант**, ввод и вывод данных производится с помощью входных и выходных терминалов, при этом данные из входного терминала доступны во всех кадрах. Для передачи данных внутри структуры

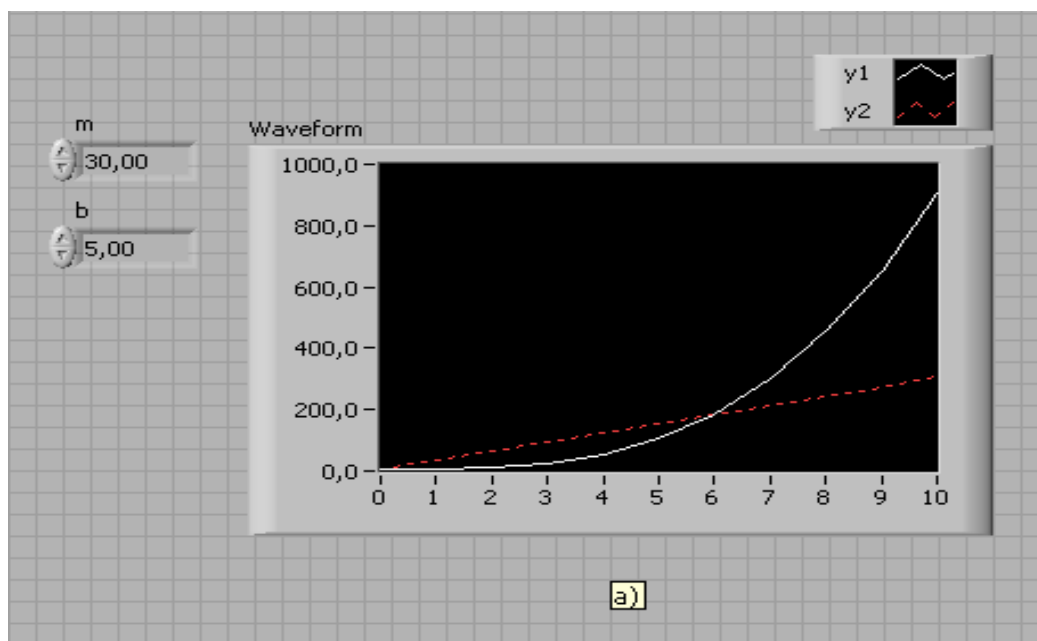


Стековая последовательность (рис. 10.23а) используется **терминал локальной переменной** (Sequence Local). Терминал локальной переменной создается с помощью строки **Добавить локальную переменную** (Add Sequence Local) контекстного меню границы структуры. В исходном состоянии терминал локальной переменной, появляющийся в текущем и других кадрах структуры, пуст. После подключения источника данных к локальной переменной в текущем и последующих кадрах появляется стрелка, указывающая направление передачи данных. Данные на выходах структуры Стековая последовательность появляются только после окончания ее выполнения.

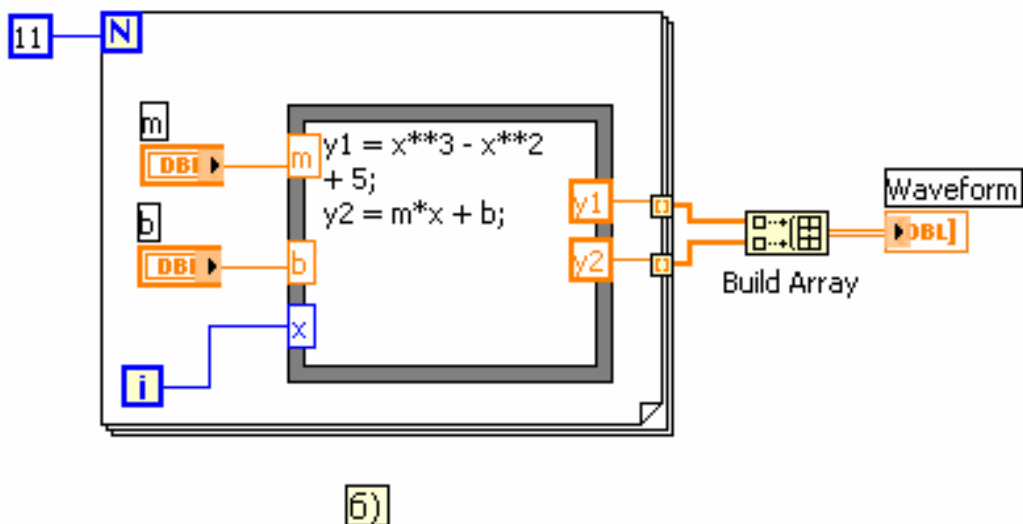
В отличие от этого в структуре **Открытая последовательность** (рис. 10.23б) данные между кадрами передаются без дополнительных переменных и выводятся на выход по мере выполнения кода кадров.

Структура узел **Формула** (Formula Node) представляет перестраиваемый по размеру прямоугольник, в котором можно в текстовом виде записывать математические выражения и операторы. Узел **Формула** целесообразно использовать при наличии множества переменных или при выполнении сложных расчетов. Формулы записываются с использованием функций и операторов, перечисленных в двух последующих таблицах. В первой таблице приведены имена и иконки аналогичных функций языка G. Запись каждой формулы должна заканчиваться символом «;». Имена входных и выходных переменных вводят соответственно во входные и выходные терминалы, формируемые с помощью строк **Добавить вход** (Add Input) и **Добавить выход** (Add Output) контекстного меню структуры. Имена вписываются в зачерненный прямоугольник непосредственно после вызова терминала. При записи имен необходимо соблюдать условие одинаковости регистра.

Список функций узла Формула приведен в таблице приложения 3, раздел 10.5.1. Примеры записи операторов в узле Формула приведены на рис. 10.24.



(а) лицевая панель



(б) – панель блок-диаграммы

Рис. 10.24. Примеры записи операторов в узле **Формула**

Структура **Событие** (Event Structure) имеет одну или более поддиаграмм или вариантов событий, из которых только один выполняется при обращении к структуре. Структура **Событие** ожидает наступления события на лицевой панели, после чего выполняет соответствующий вариант в целях обработки события. С помощью контекстного меню структуры можно добавить новые варианты событий или определить вид обрабатываемого события. Подключение значения к терминалу времени ожидания, находящемуся в левом верхнем углу структуры, позволяет задать величину интервала ожидания события структурой в миллисекундах. По умолчанию значение на входе этого терминала равно 1, что соответствует отсутствию времени ожидания.

Структура **Локальная переменная** (Local Variable) используется для обеспечения доступа к элементам управления или индикации из более чем одной точки блок-диаграммы. Локальные переменные всегда содержат действительные значения объектов лицевой панели, с которыми они ассоциированы. Локальную переменную можно устанавливать в режим как записи, так и чтения данных. Таким образом, с помощью локальной переменной можно записывать данные в элемент управления или считывать данные с индикатора. Помимо этого, для одного элемента управления можно определить несколько локальных переменных, установленных в режим записи или считывания. Таким способом можно управлять параллельными циклами с помощью одной переменной.

Наиболее просто локальная переменная элемента управления или индикатора может быть создана с помощью строки **Создать => Локальная переменная** (Create => Local Variable) контекстного меню этих элементов или их терминалов. Локальная переменная также может быть установлена из палитры **Структуры**, однако при этом ее необходимо связать с определенным элементом управления или индикатором с помощью опции **Выбрать элемент** (Select Item) контекстного меню локальной переменной. Выбираемый

элемент должен иметь ярлык.

Структура **Глобальная переменная** (Global Variable) используется для передачи данных между одновременно работающими ВП. Глобальная переменная является встроенным элементом Lab VIEW, имеющим лицевую панель, но не имеющим диаграммы. В зависимости от направления передачи и типа данных на **лицевой** панели глобальной переменной устанавливаются один или несколько элементов управления, которые идентифицируются своими ярлыками. Таким образом, лицевая панель глобальной переменной является контейнером, с помощью которого различные ВП могут обмениваться данными.

Глобальную переменную выбирают из палитры **Структуры**. После установки на блок-диаграмме с помощью контекстного меню или двойного щелчка на терминале глобальной переменной необходимо открыть ее лицевую панель, установить необходимые элементы управления и индикаторы и сформировать их ярлыки. Выбор ярлыка из списка в строке **Выбрать элемент** (Select Item) контекстного меню терминала глобальной переменной позволяет связать эту переменную с соответствующим элементом управления или индикатором.

**Массив** LabVIEW – это набор индексированных данных одного типа. Он может иметь любую размерность и содержать до  $2^{31}$  элементов на размерность. Элементом массива может быть любой тип данных, за исключением массива, таблицы или графика. Доступ к элементам осуществляется с помощью индексов. Значения индексов лежат в диапазоне от 0 до N-1, где N – количество элементов массива.

В LabVIEW массивы могут быть созданы как вручную на лицевой панели или на панели блок-диаграммы, так и программно. На лицевой панели могут быть созданы массивы элементов управления или индикаторов, на панели блок-диаграммы – массивы констант. Программно массивы создаются с помощью структур и соответствующих функций.

Для формирования массивов на лицевой панели необходимо разместить на ней **шаблон массива** (array shell) из подпалитры **Массив и кластер** (Array & Cluster) палитры элементов управления (рис. 10.25а). При этом терминал массива имеет черный цвет и отображает пустые скобки. В **окно отображения элемента** может быть помещен **объект данных** – элемент управления или индикатор в соответствии с типом формируемого массива, за исключением типов, перечисленных выше. Помещение объекта сопровождается мерцанием оболочки, а его фиксация в окне приводит к присвоению терминалу массива цвета, типа и надписи, присущих помещенному объекту. После задания типа массива он может использоваться для ввода или вывода данных.

Аналогичным образом создается массив констант на блок-диаграмме. Для создания массива констант необходимо разместить на диаграмме **шаблон массива констант** (Array Constant) из палитры **Массив** (Array) и поместить в него константу необходимого типа.

На рис. 10.25 показан вид массивов различного типа на лицевой панели

и соответствующих терминалов на блок-диаграмме: массив числовых элементов управления (б), массив логических индикаторов (в), массив элементов управления строкового типа (г) и двумерный массив индикаторов целых чисел (д). Для программного формирования и обработки массивов используются рассмотренные выше структуры **Цикл с фиксированным числом итераций** (For Loop) и **Цикл по условию** (While Loop), имеющие в своем составе индексную переменную **i**.

Если элементы массива формируются в структуре цикла с фиксированным числом итераций, то их преобразование в массив происходит в терминале вывода данных, находящемся по умолчанию в режиме **Включить индексирование** (Enable Indexing) (рис. 10.26). В этом режиме терминал вывода данных представляет пустотелый двойной квадрат. Формирование массива сопровождается изменением толщины провода с данными после его выхода из структуры. Выключение режима индексирования осуществляется с помощью строки **Отключить индексирование** (Disable Indexing) контекстного меню терминала. При этом из структуры будет выводиться только последний элемент.

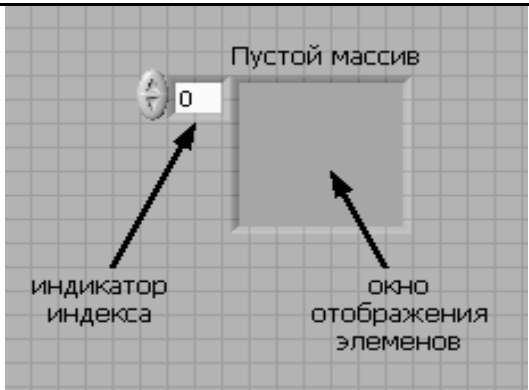

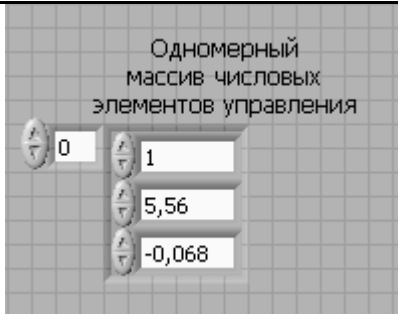
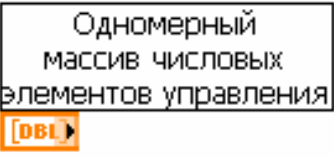
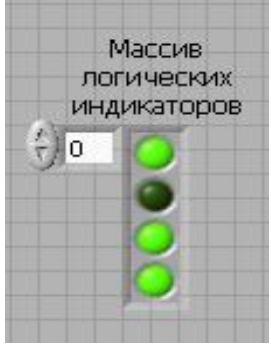
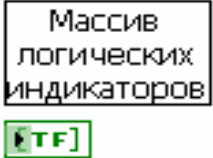
Рис.	На лицевой панели	На блок-диаграмме
а)		
б)		
в)		



Рис. 10.25. Вид массивов на лицевой панели и соответствующих терминалов на блок-диаграмме

Описанный режим автоматического индексирования и аккумулярования массивов на границе цикла путем добавления одного нового элемента в каждом повторении цикла называется автоиндексированием (auto-indexing).

При вводе массива в структуру цикла терминал ввода данных по умолчанию также находится в состоянии **Включить индексирование** (рис. 10.26). В данном режиме терминал передает в цикл по одному элементу в каждую итерацию. В этом случае структура автоматически определяет размерность массива и нет необходимости задавать значение терминала числа итераций. При подаче на вход нескольких массивов разной длины структура настраивается на самый короткий массив. То же самое происходит и при одновременно подключенном терминале числа итераций.

Если точку входа перевести в состояние **Отключить индексирование**, массив будет вводиться в цикл целиком (рис. 10.26). При этом, естественно, способность автоиндексирования теряется и значение числа циклов должно быть задано.

На рис. 10.26 видно, что при выводе одномерного массива из цикла в режиме автоиндексирования он преобразуется в двухмерный массив. Массив такой же размерности может быть создан на основе скалярных переменных при использовании структуры из внутреннего и внешнего циклов. Увеличение размерности массива, созданного оператором на лицевой панели или в блок-диаграмме, может быть выполнено с помощью строки **Добавить размерность** (Add Dimension) контекстного меню элемента управления / отображения **индекса массива**. Такое же действие может быть выполнено и инструментом перемещения.

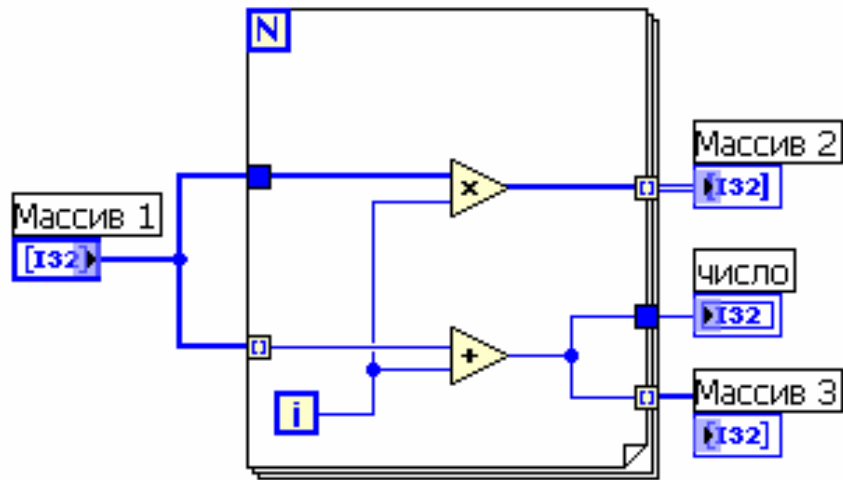


Рис. 10.26. Варианты ввода и вывода массивов в структуре цикла с фиксированным числом итераций

В структуре **Цикл по условию** ситуация с индексацией противоположная, то есть по умолчанию терминалы ввода и вывода данных находятся в состоянии **Отключить индексирование**.

Наряду с описанными выше элементами индикации значений массивов, представляющими упорядоченные наборы элементов индикации соответствующих скаляров, образующих массив, в LabVIEW для отображения массивов и наборов числовых данных применяются специальные графические индикаторы – **Графики** (Graph).

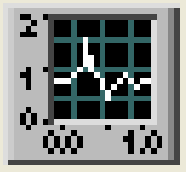
В состав палитры Графики входят следующие графические индикаторы (рис. 10.27).



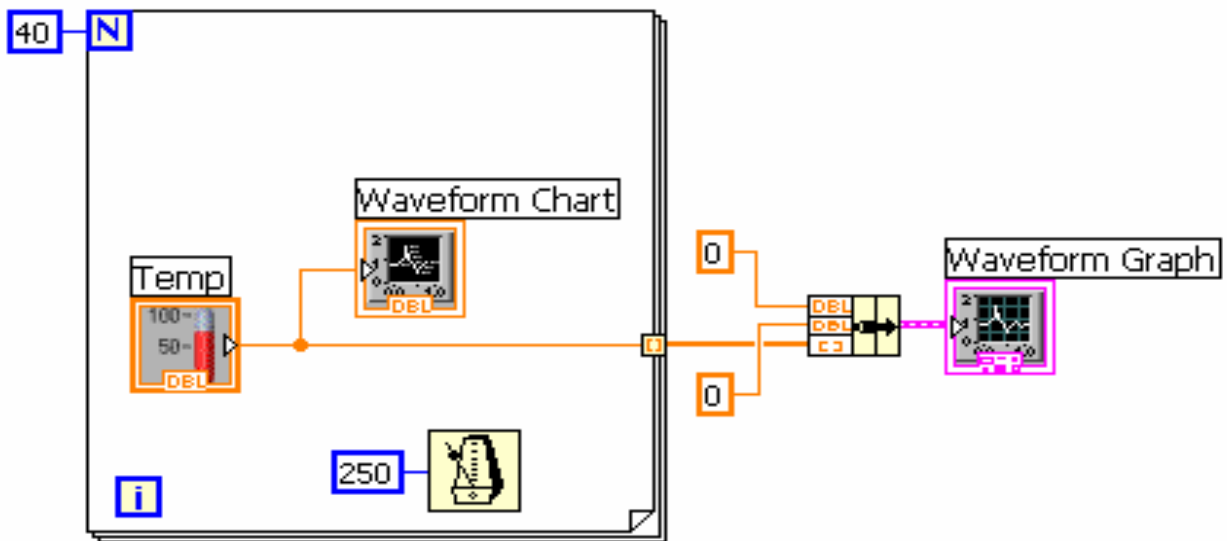
Рис. 10.27. Палитра Графики

Изображение	Описание
	<p><b>Развертка осциллограммы (Waveform Chart)</b> – графический индикатор, имитирующий работу самописца. Поступающие данные нумеруются по оси абсцисс целыми числами. Поскольку индикатор запоминает всю подаваемую на него информацию в виде отдельных чисел, он устанавлива-</p>

ется внутри структур **Цикл с фиксированным числом итераций** и **Цикл по условию** (рис. 10.23). Индикатор может быть многолучевым и многоэкранным. Для отображения двух и более наборов данных от разных источников необходимо объединить их в кластер с помощью функции **Объединить (Bundle)**. Стирание информации производится с помощью строки **Операции с данными (Data Operations)** => **Очистить развертку (Clear Chart)** контекстного меню графика



**График осциллограммы (Waveform Graph)** – графический индикатор, имитирующий работу осциллографа. Он принимает данные в виде массива чисел и отображает их с равномерным шагом. Для отображения двух и более массивов данных они должны быть объединены в двумерный массив с помощью функции **Сформировать массив (Build Array)**. Для отображения графика с заданной начальной точки и с заданным шагом предусмотрено формирование кластера из трех элементов с помощью функции **Объединить (Bundle)**, на верхний вход которой подается начальное смещение, а на средний - шаг отображения (рис. 10.28).



a)

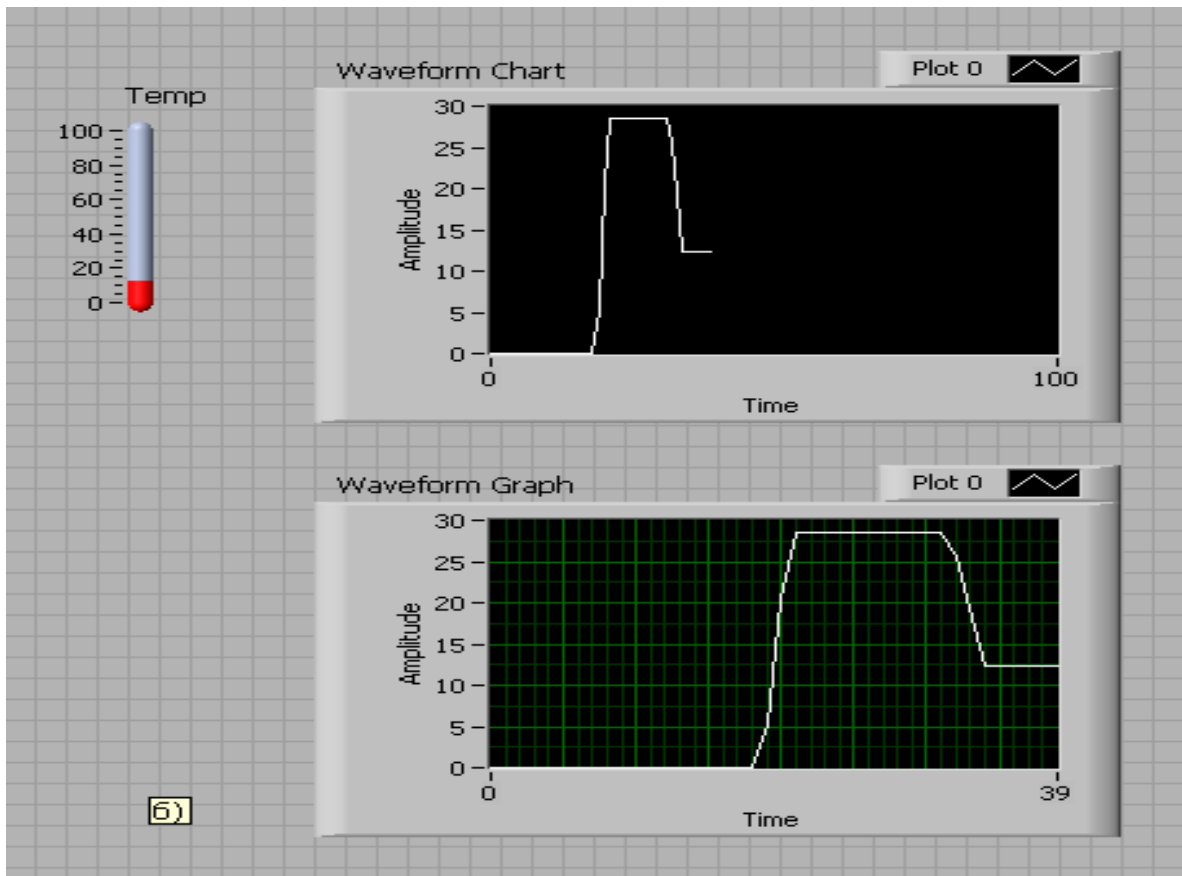
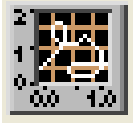


Рис. 10.28. Блок-диаграмма ВП контроля температуры с индикаторами Развертка температуры и График температуры. (а) – лицевая панель ВП, (б) – панель блок-диаграммы ВП

Изображение	Описание
	<p>Двухкоординатный график (XY Graph) – графический индикатор, позволяющий отображать функциональные зависимости. Для отображения на данном индикаторе массива точек с произвольными координатами по осям необходимо сформировать массив кластеров или объединить два массива координат X и Y в кластер. С целью отображения двух и более графиков необходимо использовать функцию <b>Сформировать массив (Build Array)</b> для формирования массива кластеров (рис. 10.29).</p>



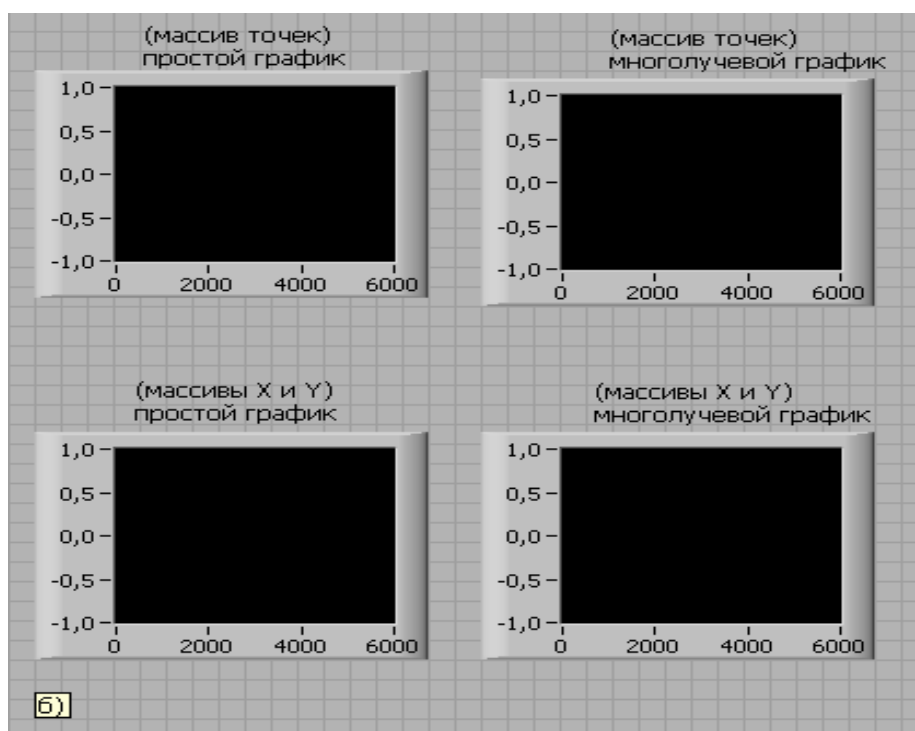
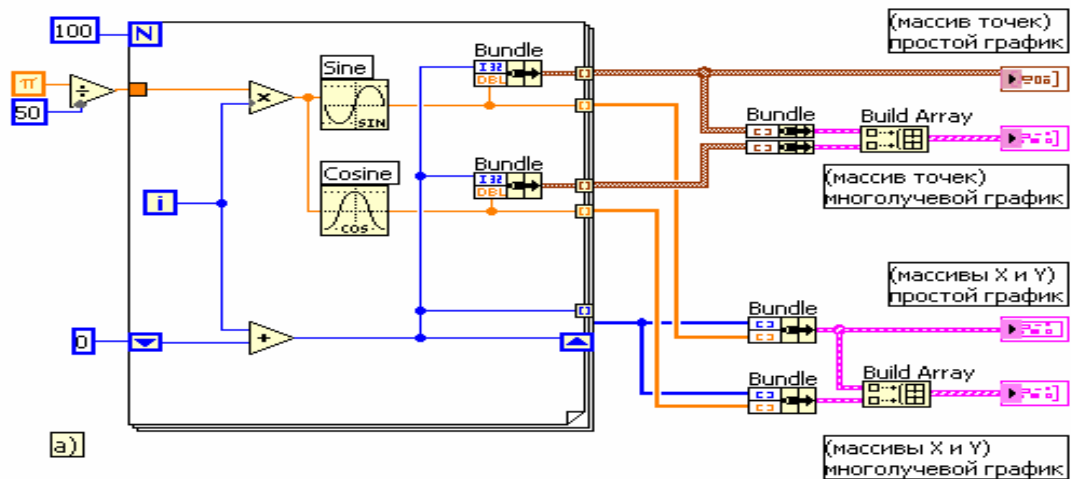
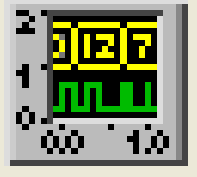


Рис. 10.29. Блок-диаграмма ВП XY Graph из набора примеров NI Example Finder. (а) – лицевая панель ВП, (б) – панель блок-диаграммы ВП

Изображение	Описание
	Экспресс-ВП. Двухкоординатный график (XY Graph)
	Графические индикаторы <b>Развертка интенсивности</b> (Intensity Chart) и <b>График интенсивности</b> (Intensity Graph) служат для отображения двумерных массивов данных в виде графиков интенсивности такой же размерности. В исходном состоянии на <b>шкале интенсивностей</b> (Ramp) отображается 3 цвета: черный, соответствующий нижней половине диапазона, синий, соответствующий верхней

	<p>половине диапазона, и белый, соответствующий превышению диапазона. Однако при включении режима <b>интерполяции</b> (Interpolate Color) в шкале интенсивностей градации интенсивности при переходе от черного к белому становятся плавными. При желании изменить палитру цветов необходимо сформировать <b>таблицу цветов</b> (Color Table), то есть рассчитать массив цифровых значений цветов, которые будут соответствовать градациям величины данных, и подключить таблицу к <b>узлу свойств</b> графического индикатора (Property Node) установленным свойством <b>таблица цветов</b> (Color Table).</p>
	<p>Графический индикатор <b>График цифровой осциллограммы</b> (Digital Waveform Graph) служит для отображения массива целых чисел в виде диаграмм логических сигналов, соответствующих двоичным разрядам чисел. Для работы индикатора необходимо сформировать кластер, содержащий начальное значение, шаг, отображаемые данные и число портов.</p>

Все перечисленные выше графические индикаторы имеют в меню Настройки более обширный раздел **Видимые элементы** (Visible Items) и дополнительные строки, связанные с настройкой осей. Так, в частности, в состав меню **Видимые элементы** этих индикаторов входят следующие разделы (рис. 10.30): **Ярлык** (Label), **Заголовок** (Caption), **Панель редактирования графика** (Plot Legend), **Панель редактирования шкалы** (Scale Legend), **Палитра элементов управления графиком** (Graph Palette), **Панель редактирования курсора** (Cursor Legend), **Линейка прокрутки** (Scrollbar), **Шкала X** (X Scale), **Шкала Y** (Y Scale).

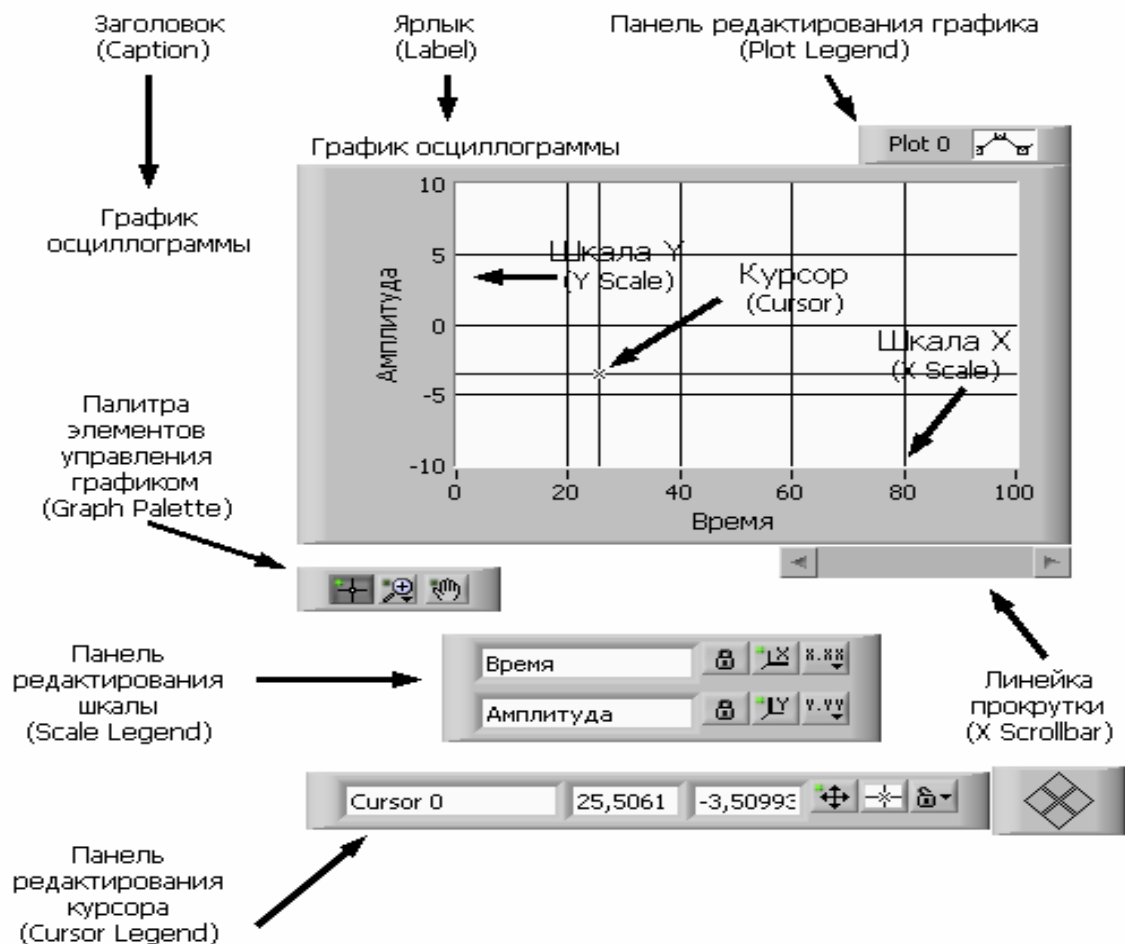


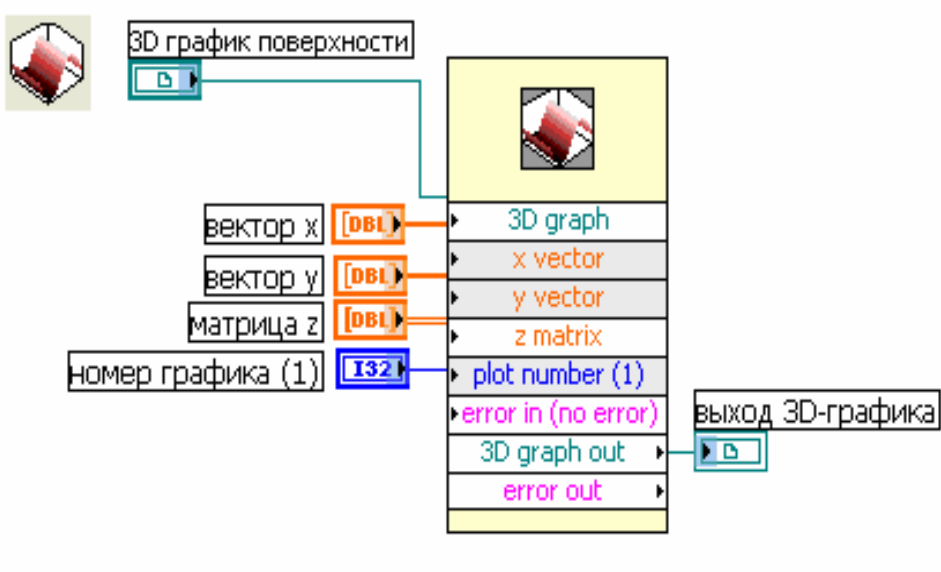
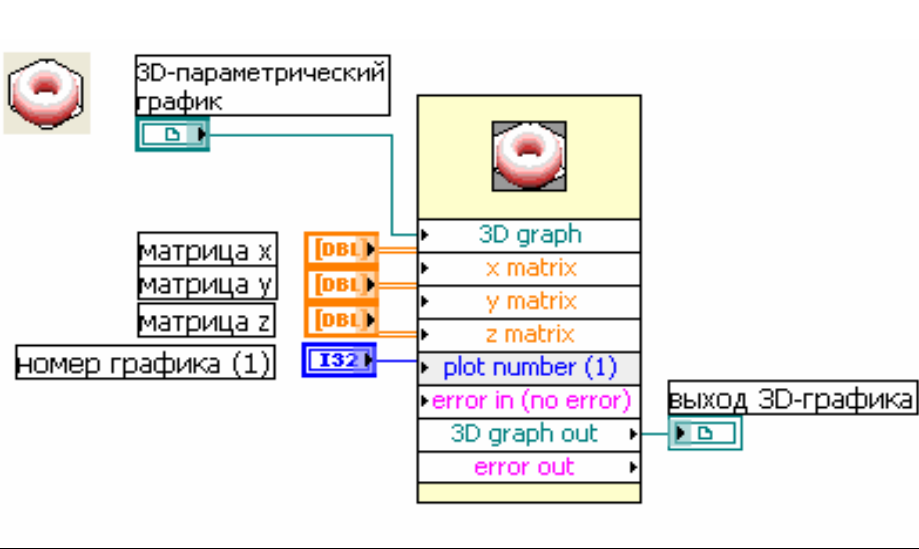
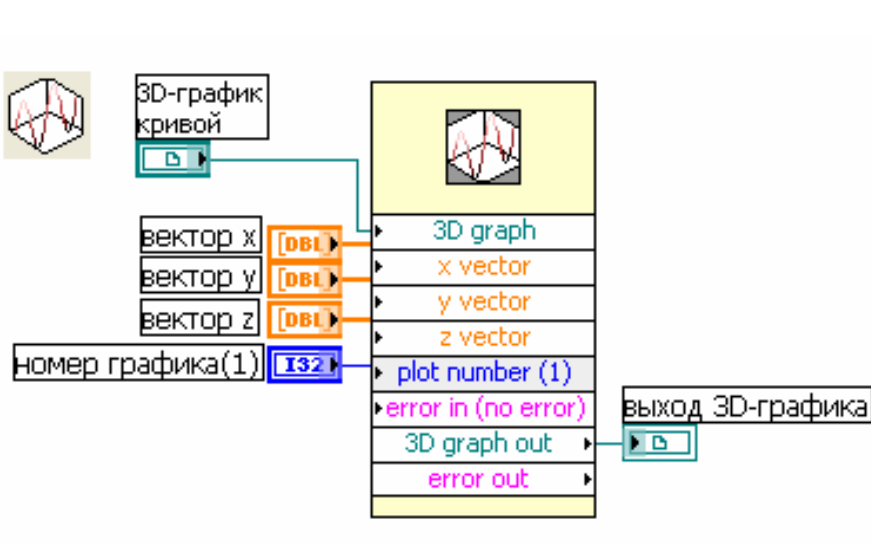
Рис. 10.30. Вид графика осциллограммы с панелями и атрибутами

Меню **Видимые элементы** графического индикатора **Развертка осциллограммы** отличается тем, что на месте строки **Панель редактирования курсора** находится строка **Цифровой индикатор**.

**Панель редактирования графика** может перестраиваться по размеру в вертикальном направлении для настройки параметров набора графиков. Панель редактирования графика имеет свое контекстное меню, которое позволяет настраивать тип графиков, их цвет, ширину и тип линии, тип точек и вид их соединения.

**Панель редактирования шкалы** позволяет фиксировать или сбрасывать в исходное состояние масштаб по осям, установленный пользователем с помощью одного из инструментов, входящих в состав **палитры элементов управления графиком**. Помимо этого, она позволяет настраивать параметры осей: формат, точность, характер расположения меток, видимость<sup>1</sup> оси и подписи, цвет сетки. Эти и ряд других параметров могут быть также установлены и из контекстного меню осей индикатора.

В состав палитры **Графики** входят также трехмерные графики. Установка таких графиков на лицевой панели сопровождается появлением на блок-диаграмме связки из **ссылки (Refnum)** на элемент управления ActiveX **CWGraph3D** и соответствующего ВП.

Изображение	Описание
 <p>3D график поверхности</p> <p>вектор x [0x1]</p> <p>вектор y [0x1]</p> <p>матрица z [0x1]</p> <p>номер графика (1) [I32]</p> <p>3D graph</p> <p>x vector</p> <p>y vector</p> <p>z matrix</p> <p>plot number (1)</p> <p>error in (no error)</p> <p>3D graph out</p> <p>error out</p> <p>выход 3D-графика</p>	<p>Графический индикатор <b>3D - график поверхности</b> (3D Surface Graph) предназначен для отображения двумерных данных в виде поверхности на трехмерном графике</p>
 <p>3D-параметрический график</p> <p>матрица x [0x1]</p> <p>матрица y [0x1]</p> <p>матрица z [0x1]</p> <p>номер графика (1) [I32]</p> <p>3D graph</p> <p>x matrix</p> <p>y matrix</p> <p>z matrix</p> <p>plot number (1)</p> <p>error in (no error)</p> <p>3D graph out</p> <p>error out</p> <p>выход 3D-графика</p>	<p>Графический индикатор <b>3D-параметрический график</b> (3D Parametric Graph) отображает данные с помощью трех двумерных массивов, характеризующих проекции графика на плоскости x, y и z</p>
 <p>3D-график кривой</p> <p>вектор x [0x1]</p> <p>вектор y [0x1]</p> <p>вектор z [0x1]</p> <p>номер графика (1) [I32]</p> <p>3D graph</p> <p>x vector</p> <p>y vector</p> <p>z vector</p> <p>plot number (1)</p> <p>error in (no error)</p> <p>3D graph out</p> <p>error out</p> <p>выход 3D-графика</p>	<p>Графический индикатор <b>3D-график кривой</b> (3D Curve Graph) используется для отображения данных в виде пространственной кривой</p>

Настройка графиков осуществляется с помощью диалогового окна **Свойства: CWGraph3D Control**, вызываемого с помощью строки

CWGraph3D => Свойства контекстного меню графика.

## 10.4. Функции численных методов решения дифференциальных уравнений

### 10.4.1. Решение ОДУ методом Эйлера

Виртуальный прибор, построенный согласно рис.10.31 находит решение обыкновенных дифференциальных уравнений (ОДУ) с начальными условиями, используя метод Эйлера.

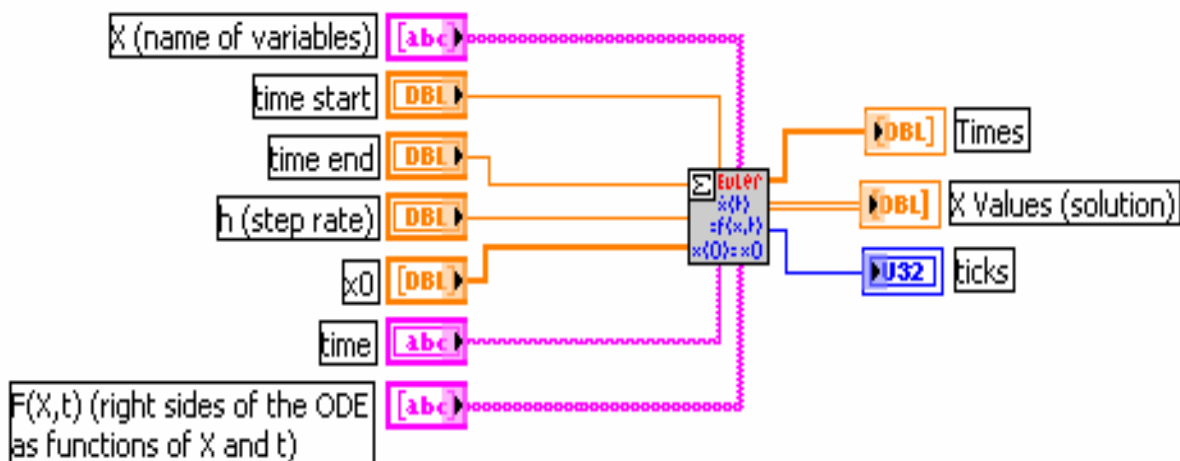


Рис.10.31

- Вход **X** представляет массив строк с именами переменных.
- Вход **начальное время** (time start) задает начальную точку решения ОДУ. По умолчанию его значение равно 0.
- Вход **конечное время** (time end) определяет конечную точку временного интервала решения. По умолчанию его значение равно 1,0.
- Вход **h** задает величину фиксированного шага. По умолчанию его значение равно 0,1.
- Вход **X0** определяет вектор начальных условий  $x[10] \dots x[n0]$ . Между компонентами X0 и X существует однозначное соответствие.
- Вход **время** (time) представляет строку, задающую переменную времени. По умолчанию в качестве переменной используется t.
- Вход **F(X,t)** задает одномерный массив строк, представляющих правые части дифференциальных уравнений.
- Выход **интервалы времени** (Times) отображает массив, представляющий интервалы времени. Метод Эйлера использует одинаковые временные интервалы между **начальным временем** и **конечным временем**.
- Выход **значения X** (X Values) отображает двумерный массив вектора решений  $x[10], \dots, x[n]$ . Верхний индекс соответствует интервалам вре-

мени, определенным в массиве **интервал времени**, нижний индекс - элементам  $x[10] \dots x[n]$ .

Стандартная форма записи системы линейных дифференциальных уравнений первого порядка включает запись самой системы и ее начальных условий:

$$\begin{cases} \dot{x}_1(t) = f_1(x_1(t), \dots, x_n(t), t) \\ \vdots \\ \dot{x}_n(t) = f_n(x_1(t), \dots, x_n(t), t) \end{cases} \quad \begin{cases} x_1(t_0) = x_{10} \\ \vdots \\ x_n(t_0) = x_{n0} \end{cases}$$

Предполагается, что функции  $f, -f_n$ , начальные условия  $x_{10}, \dots, x_{n0}$  и начальный момент времени заданы. Необходимо определить вид зависимостей  $x^*(t), \dots, X_n(t)$ .

Используя обозначения  $F = (f_1, \dots, f_n)$ ,  $X = (x_1(t), \dots, x_n(t))$  и  $X_0 = (x_{10}, \dots, x_{n0})$ , систему уравнений можно записать в векторной форме

$$\begin{cases} \dot{X}(t) = F(X(t), t) \\ X(t_0) = X_0 \end{cases}$$

Метод Эйлера является наиболее простым методом решения обычных дифференциальных уравнений (ОДУ). Его суть заключается в использовании при пошаговом интегрировании зависимой переменной двух первых членов ряда Тейлора. Таким образом, итерационная процедура вычисления значений  $X(t)$  при начальном моменте времени  $t_0$  и достаточно малом шаге интегрирования  $h$  может быть записана в следующем виде:

$$X(t_0 + h) = X(t_0) + hF(X(t_0), t_0);$$

$$X(t_0 + 2h) = X(t_0 + h) + hF(X(t_0 + h), t_0 + h).$$

Этот процесс останавливается, когда **начальное время +  $nh < \text{конечное время}$** , где **конечное время** представляет правую конечную точку исследуемого процесса.

#### 10.4.2. Решение ОДУ методом Рунге-Кутты 4-го порядка

ВП (рис. 10.32) решает обыкновенные дифференциальные уравнения (ОДУ) с начальными условиями, используя метод Рунге-Кутты. Метод Рунге-Кутты работает с фиксированным шагом, обеспечивая вместе с тем более высокую точность, чем метод Эйлера. Функции одноименных входов и выходов идентичны рассмотренному выше ВП **Решение ОДУ методом Эйлера** (ODE Euler Method).

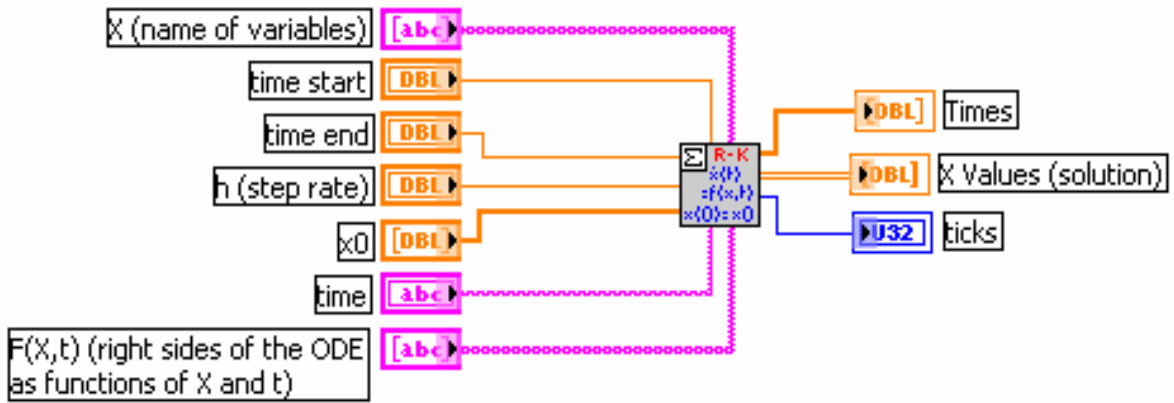


Рис.10.32

Метод Рунге-Кутта имеет 4-й порядок, и итерационные расчетные выражения выглядят следующим образом:

$$\begin{aligned}
 K_1 &= hF(X(t_n), t_n) & K_2 &= hF\left(X(t_n) + \frac{K_1}{2}, t_n + \frac{h}{2}\right) \\
 K_3 &= hF\left(X(t_n) + \frac{K_2}{2}, t_n + \frac{h}{2}\right) & K_4 &= hF(X(t_n) + K_3, t_n + h) \\
 X(t_{n+1}) &= X(t_n) + \frac{K_1}{6} + \frac{K_2}{3} + \frac{K_3}{3} + \frac{K_4}{6}
 \end{aligned}$$

Этот процесс останавливается, когда **начальное время + nh < конечное время**, где **конечное время** представляет правую конечную точку исследуемого процесса.

### 10.4.3. Решение ОДУ методом Кэш-Капа 5-го порядка точности

ВП представленная на рис.10.33 решает ОДУ с начальными условиями с помощью метода Кэш-Капа. Метод Кэш-Капа работает с адаптивным шагом и в вычислительном отношении более эффективен по сравнению с методами Эйлера и Рунге-Кутта.

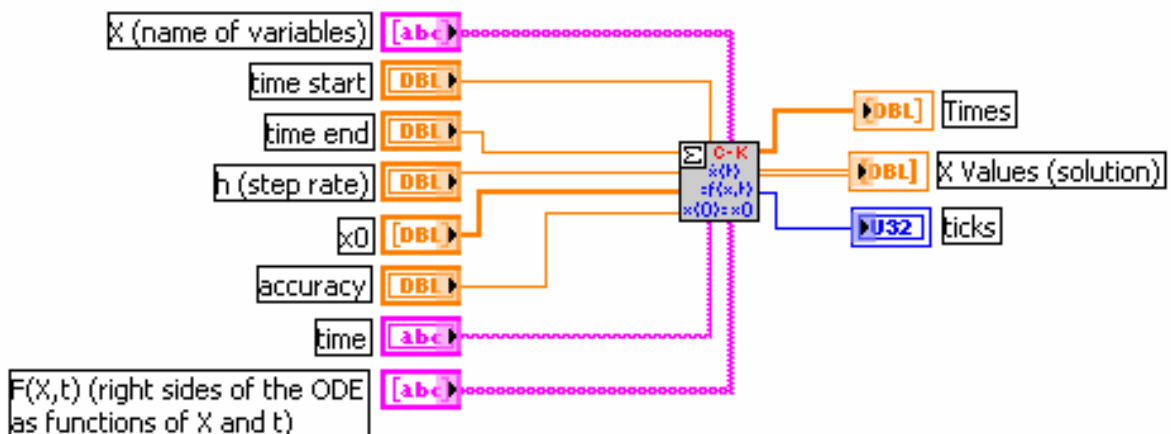
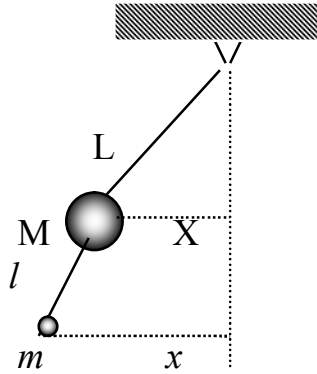


Рис.10.33

### 10.4.4. Моделирование системы «Двойной маятник»

**1. Описание явления колебаний.** Представим себе два маятника примерно с одинаковым периодом колебаний, из которых один тяжелый (например, люстра), а другой легкий. Легкий маятник подвешен к тяжелому так, как это показано на рис.10.29.



Если отклонить слегка тяжелый маятник и опустить его, то колебания его внезапно прекращаются. Одновременно легкий маятник приходит в быстрое движение, которое спустя некоторый промежуток времени прекращается, что сопровождается возникновением колебаний тяжелого маятника. Такие чередующиеся колебания двух колебательных систем, сопровождающиеся обменом энергией, носят название биений.

Рис. 10.34

**2. Математическая модель (дифференциальные уравнения движения).** Колебания двойного маятника изучаются в курсе теоретической механики. Поэтому не будем проводить математическое описание этого явления, а сразу приведем дифференциальные уравнения для случая малых колебаний. Процесс описывается системой дифференциальных уравнений

$$\begin{aligned} \frac{dv_X}{dt} &= -\omega^2(1+2\mu)X + \mu\omega^2 X, \quad \frac{dX}{dt} = v_X, \\ v_X(0) &= 0, \quad X(0) = X_0 \\ \frac{dv_x}{dt} &= -\omega^2 x + \omega^2 X, \quad \frac{dx}{dt} = v_x, \\ v_x(0) &= 0, \quad x(0) = 0 \end{aligned}, \quad (10.1)$$

где  $X$  и  $x$  - отклонения тяжелого и легкого маятников от положения равновесия,

$\mu = \frac{m}{M}$  - малое число, а  $m$  и  $M$  - массы легкого и тяжелого маятников,

$\omega^2 = \frac{g}{l}$ ,  $g$  - ускорение свободного падения, а  $l$  - длина легкого маятника.

Предполагается, что приведенные длины маятников равны, т.е.  $L=l$ .

**3. Задание на моделирование.** Построить виртуальный прибор позволяющий проводить исследования по изучению колебаний системы «Двойной маятник»

**4. Результаты моделирования.** Ниже приведены вид виртуального прибора (рис.10.35) и панель блок-диаграммы (рис.10.36)



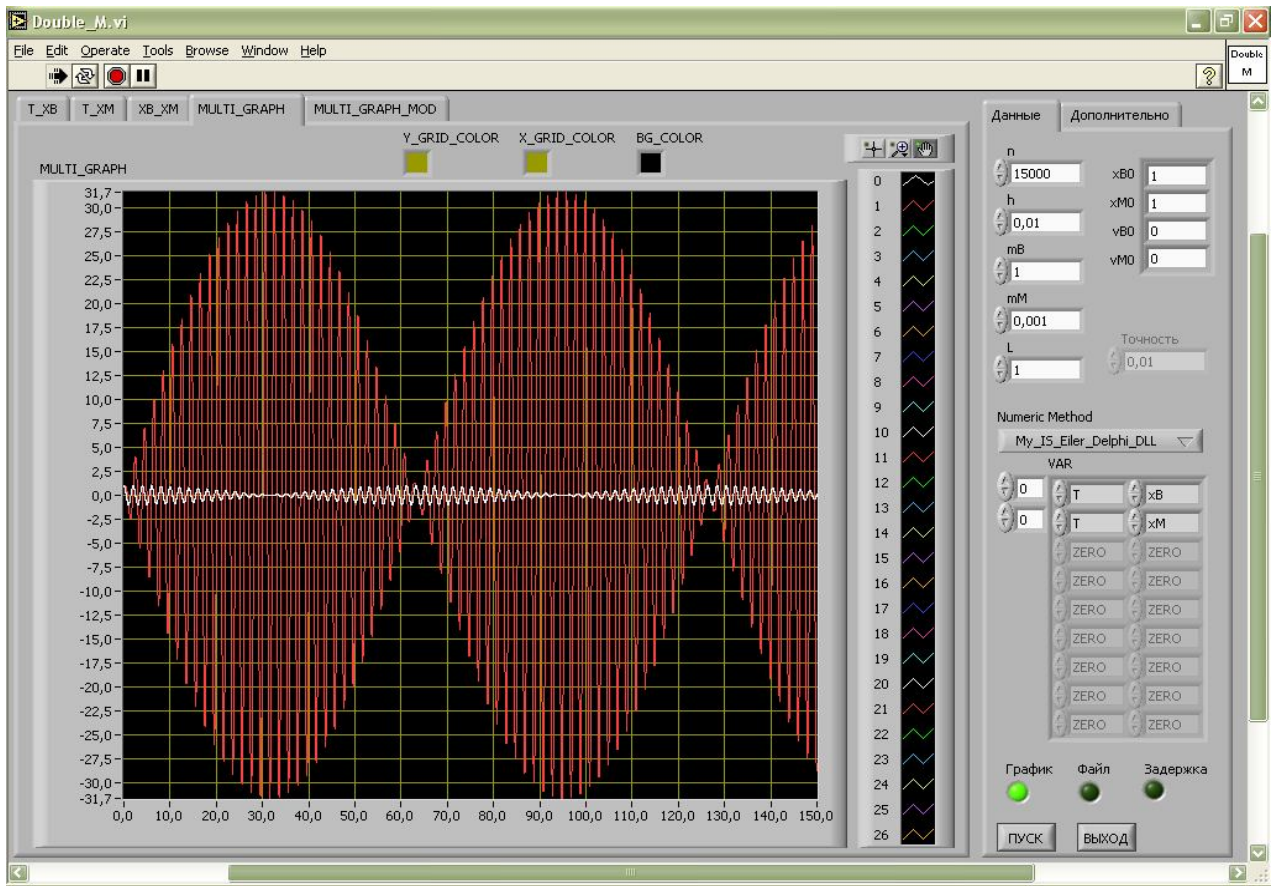


Рис. 10.35. Вид лицевой панели

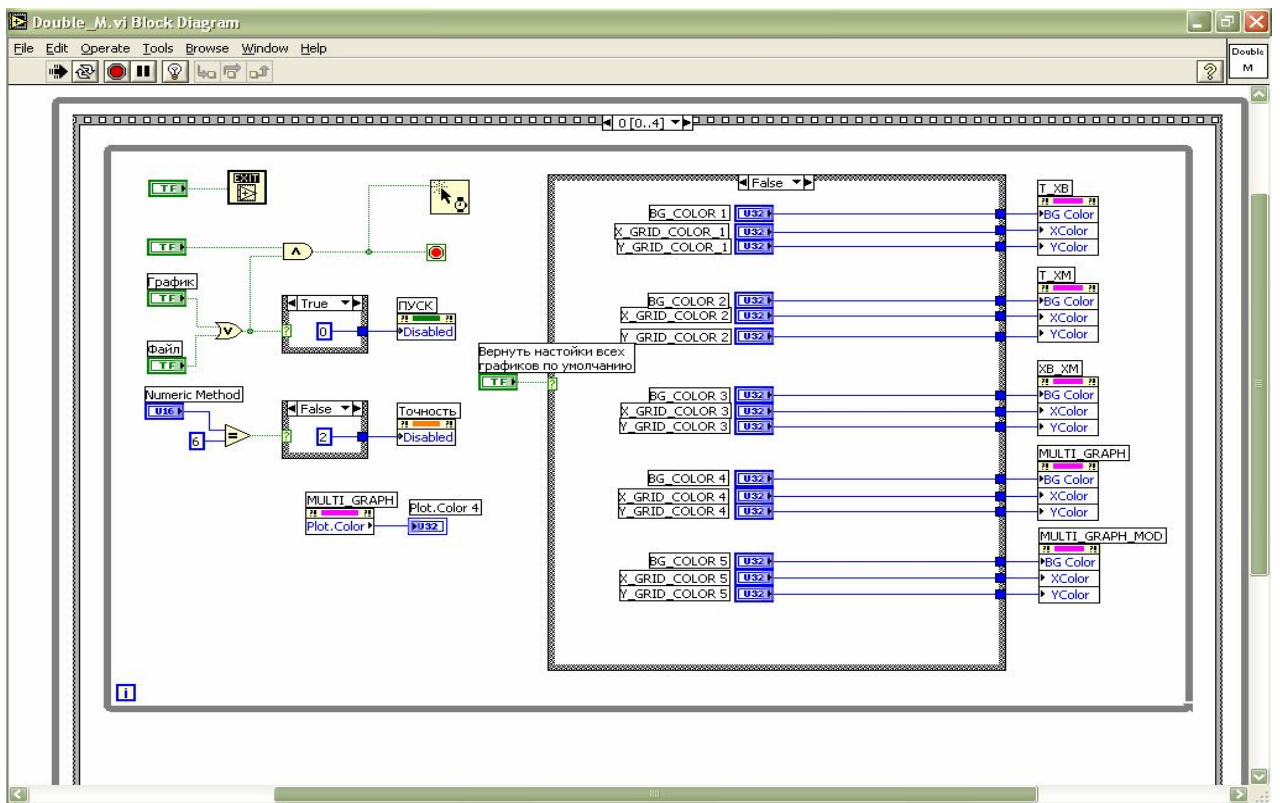


Рис. 10.36. Вид панели блок-диаграммы

## Рекомендуемая литература

### К главе 1

1. Каганов В.И. Компьютерные вычисления в средах Excel и MathCAD.- М.: Горячая линия – Телеком, 2003. – 328с
2. Маликов Р.Ф., Сулейманов Р.Р. Информатика классная и внеклассная. Учебно-методическое пособие. –Уфа: Изд-во «Гилем».- 2004. – 365с.
3. Маликов Р.Ф. Практикум по компьютерному моделированию физических явлений и объектов. – Уфа: Изд-во БГПУ, 2004. – 236 с.
4. Маликов Р.Ф. Основы математического моделирования. Учебное пособие. – Уфа: Изд-во БГПУ, 2005. – 134с.

### К главе 2

1. Дьяконов В. П. Справочник по системе символьной математики Derive. - М.: СК Пресс, 1998.
2. Лобанова О. В. Практикум по решению задач в математической системе Derive. Алгебра. Геометрия. Математический анализ. Математическая статистика. Теория вероятностей. – М.: Изд. <Финансы и статистика>, 1999.
3. Половко А.М. Derive для студента. – СПб.БХВ – Петербург, 2005.
4. Дьяконов В.П. Справочник по применению системы Derive.: – М . Наука, Физматлит— 1996.
5. Дьяконов В. П. Справочник по системе символьной математики Derive. – М.: СК-ПРЕСС.— 1998.
6. [www.edu.severodvinsk.ru](http://www.edu.severodvinsk.ru)
7. [www.ito.bitpro.ru](http://www.ito.bitpro.ru)
8. [www.cdo.susu.ac.ru/journal/numero1/product/Slivina.html](http://www.cdo.susu.ac.ru/journal/numero1/product/Slivina.html)
9. [www.pereplet.ru/obrazovanie/stsoros/1161.html](http://www.pereplet.ru/obrazovanie/stsoros/1161.html)
10. [www.bsu.by/Converences/cas99/Abstracts/Section1/Dyakonov/Dyakonov.htm](http://www.bsu.by/Converences/cas99/Abstracts/Section1/Dyakonov/Dyakonov.htm)
11. <http://dev.sinin.nsc.ru/publication.xml?h=sinin:isr:istcpr:3327>
12. [http://intz.imm.uran.ru/int\\_z263/izone263/pub/izone8.htm](http://intz.imm.uran.ru/int_z263/izone263/pub/izone8.htm)
13. <http://www.cdo.susu.ac.ru/journal/numero1/product/Slivina.html>
14. <http://www.mycomp.com.ua/>
15. <http://www.2devochki.ru/67/15642/1.html>
16. <http://www.allbooks.ru/abbook/5934551396.shtml>.

### К главе 3

1. Дьяконов В. П. Maple 7: Учебный курс. – СПб.: Питер, 2002.
2. Семененко М.Г. Введение в математическое моделирование. – М: Солон-Р, 2002. – 112с.
3. Алексеев Е.Р., Чеснокова О.В. Решение задач вычислительной математики в пакетах Mathcad 11, MATLAB 7, Maple 9. –М.: НТ Пресс, 2006. – 496с.

4. Маликов Р.Ф. Практикум по компьютерному моделированию физических явлений и объектов. – Уфа: Изд-во БГПУ, 2004. – 236 с.
5. Маликов Р.Ф. Основы математического моделирования: учеб. пособие. – Уфа: Изд-во БГПУ, 2005. – 134с.

#### **К главе 4**

1. Поршнев С.В. Компьютерное моделирование физических процессов с использованием пакета MathCAD: Учеб. пособие. – М: Горячая линия – Телеком, 2002. –252с.
2. Поршнев С.В. Компьютерное моделирование физических систем с использованием пакета MathCAD: учеб.пособие. – М.: Горячая линия –Телеком, 2004. -319с.
3. Семенов М.Г. Введение в математическое моделирование. – М: Солон-Р, 2002. -112с.
4. Семенов М.Г. Математическое моделирование в MathCAD. – М: Альтекс-А, 2003. -206с.
5. Дьяконов В. П., Абраменкова И. В. Mathcad 8 PRO в математике, физике и Internet. – М.: Нолидж, 1996.
6. Алексеев Е.Р., Чеснокова О.В. Решение задач вычислительной математики в пакетах Mathcad 11, MATLAB 7, Maple 9. –М.: НТ Пресс, 2006. – 496с.

#### **К главе 5**

1. Дьяконов В. П. Mathematica 4.1/4.2/5.0 в математических и научно-технических расчетах. – М.: СОЛОН-Пресс, 2004.
2. Дьяконов В. П. Mathematica 4.0: Учебный курс. – М.: СОЛОН-Пресс, 2000.
3. Муравьев В., Бурланков Д. Практическое введение в пакет МАТЕМАТИКА.
4. Тарасевич Ю. Математическое и компьютерное моделирование. Вводный курс. – М.: Едиториал-УРСС, 2001.
5. Маликов Р.Ф. Основы математического моделирования: учеб. пособие. – Уфа: Изд-во БГПУ, 2005. – 134с.
6. Маликов Р.Ф. Практикум по компьютерному моделированию физических явлений и объектов. – Уфа: Изд-во БГПУ, 2004. – 236 с.

#### **К главе 6**

1. Иглин С.П. Математические расчеты на базе MATLAB: учеб. пособие. – СПб., 2005.
2. Алексеев Е.Р., Чеснокова О.В. Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9: Самоучитель. – М.: НТ Пресс, 2006. – 496 с.
3. Алексеев Е.Р., Чеснокова О.В. MATLAB 7: Самоучитель. – М.: НТ Пресс, 2006. – 464 с.

4. Коткин Г.Л., Черкасский В.С. Компьютерное моделирование физических процессов с использованием MATLAB: учеб. пособие / Новосиб. ун-т. – Новосибирск, 2001. – 173 с.
5. Дьяконов В. П., Абраменкова И. В. MATLAB. Обработка сигналов и изображений. – СПб.: Питер, 2002.
6. Дьяконов В. П., Круглое В. В. MATLAB. Анализ, идентификация и моделирование систем: Специальный справочник. – СПб.: Питер, 2002.
7. Дьяконов В. П., Круглое В. В. Математические пакеты расширения MATLAB. Специальный справочник. – СПб.: Питер, 2001.

### **К главе 7**

1. Дьяконов В. П. VisSim+Mathcad+MATLAB. Визуальное математическое моделирование. – М.: СОЛОН-Пресс, 2004. – 384 с.
2. Клиначев Н. В. Моделирование систем в программе VisSim. Справочная система. 2001
3. Дьяконов В. П. Компьютерная математика. Теория и практика. – М.: Нолидж, 2001. – 1296с.
4. <http://www.vissim.nm.ru>
5. [http://model.exponenta.ru/bt/bt\\_fed.html](http://model.exponenta.ru/bt/bt_fed.html)

### **К главе 8**

1. Дьяконов В.П. MATLAB 6/6.1/6.5. Simulink 6.1. Основы применения: Полное руководство пользователя. – М.: Солон-Пресс, 2002.
2. Дьяконов В.П. MATLAB 6/6.1/6.5. Simulink 4/5. в математике и моделировании. Полное руководство пользователя. – М.: Солон-Пресс, 2003. – 576с.
3. Дьяконов В.П. Simulink 6. Специальный справочник. – СПб.: Питер, 2002.
4. Дьяконов В. П. 6.5 SP1/7 + Simulink 5/6. Основы применения. «Библиотека профессионала». – М.: СОЛОН-Пресс, 2005.- 800 с.
5. Дьяконов В. П. Компьютерная математика. Теория и практика. – М.: Нолидж, 2001. – 1296с.
6. <http://www.aec.neva.ru>
7. <http://www.tula.net/tgpu/new/New/informatic/g7.htm>
8. <http://www.midenk.ru/info/new/s.htm>
9. <http://www.matlab.ru>

### **К главе 9**

1. Карлашук В. И. Обучающие программы. – М.: Солон-Р, 2001. – 528с.
2. Карлашук В. И. Электронная лаборатория на IBM PC. Программа Electronics Workbench.– М.: Солон-Р, 2000.- 504с.
3. Жарников Д. Радиолобитель//Ваш компьютер. 1999, № 12.
4. Баскаков С.И. Радиотехнические цепи и сигналы. – М. Высш. шк., 1988.
5. Ходяков И. А. Mathcad 6/0 и Electronics Workbench 5.12 в средней школе // Информатика и образование. 1999, №6.

6. Практикум по компьютерному моделированию электрических схем и явлений / Сост. Р. Ф. Маликов, У. Ш. Шаяхметов. – Уфа: Изд-во БГПУ, 2002. –32.
7. Практикум по изучению основ электроники и вычислительной техники в Electronics Workbench. / Сост. Сулейманов Р.Р., Маликов Р.Ф. – Уфа: Изд-во БашГПУ. - 2003. - 30с.
8. Прокубовская А.О., Окуловская А.Г. Компьютерное моделирование электронных устройств: учеб. пособие. – Екатеринбург: Изд-во УГППУ, 2001. – 103с.
9. Internet: <http://workbench.online.kg/>
10. В. Разевиг. Электронная лаборатория <http://softline.perm.ru /interactive/articles/art1.htm>
11. Цвеляя И. А. Использование системы компьютерного моделирования Electronics Workbench при изучении электро—технических дисциплин в неинженерных ВУЗах. <http://bitpro.aha.ru/ITO/ITO98/2/TSVELAJA.html>

### **К главе 10**

1. Загидуллин Р.Ш. LabView в исследованиях и разработках. – М.: Горячая линия – Телеком, 2005. -352с.
2. Бутырин П.А., Васьковская Т.А., Каратаев В.В., Матеркин С.В. Автоматизация физических исследований и эксперимента: компьютерные измерения и Виртуальные Приборы на основе LabVIEW 7 Express (30 лекций). – М.: Приборкомплект, 2006.
3. Батоврин В.К., Бессонов А.С., Мошкин В.В. "LabVIEW: практикум по электронике и измерительной технике". – М.: Приборкомплект, 2006.
4. Суранов А.Я.. "LabVIEW 7 Express: справочник по функциям". – М.: Приборкомплект, 2006.
5. Батоврин В.К., Бессонов А.С., Мошкин В.В. "LabVIEW : практикум по основам измерительных технологий"– М.: Приборкомплект, 2006.
6. Джеффри Тревис. "LabVIEW для всех". – М.: Приборкомплект, 2006.
7. Пейч Л.И., Точилин Д.А., Поллак Б.П. "LabVIEW для новичков и специалистов". – М.: Горячая линия – Телеком, 2004. -321с.
8. [www.ni.com](http://www.ni.com), [www.labview.ru](http://www.labview.ru), [www.edulib.ru](http://www.edulib.ru), [www.yandex.ru](http://www.yandex.ru), [www.mathcad.com](http://www.mathcad.com), [www.kirianov.orc.ru](http://www.kirianov.orc.ru), [www.exponenta.ru](http://www.exponenta.ru).

Практикум по вычислениям в системе Derive

Тема 1. Вычисление производных функций

**Вычислите производную функции**  $\text{SIN}(a \cdot x^2)$ .

Форма записи производной :  $\text{DIF}(u, x)$ .

Для того, чтобы вычислить производную функции, необходимо записать ее следующим образом:  $\text{DIF}(\text{SIN}(a \cdot x^2), x)$ . Затем нажать Simplify и на экране выведется:

$$2 \cdot a \cdot x \cdot \text{COS}(a \cdot x^2).$$

**Вычислите аналогичным способом производные следующих функций:**

- |                                 |                                      |
|---------------------------------|--------------------------------------|
| 1) $\text{LN}(\text{Cos}(x))$ ; | 6) $\text{Sin}(x) / \text{Cos}(x)$ ; |
| 2) $\text{Cos}(x^2 + 2)$ ;      | 7) $1 / \text{Sin}(x)^2$ ;           |
| 3) $5 \text{Sin}(3 - x^2)$ ;    | 8) $\text{Cos}(x) + y^2$ ;           |
| 4) $(ax + by)^3$ ;              | 9) $\text{Sin}(x^2 + 6)$ ;           |
| 5) $\text{Sin}(x + y)^2$ ;      | 10) $(2ax - by)^2$ .                 |

Тема 2. Вычисление интегралов

**Вычислите интеграл функции**  $x / (a^2 + b^2 \cdot x^4)$ .

Форма записи интегралов :

$\text{INT}(u, x)$ ,

$\text{INT}(u, x, c)$ , где  $c$  – это константа

$\text{INT}(u, x, a, b)$ , где  $a$  и  $b$  – переменные.

Для того, чтобы вычислить интеграл функции, необходимо записать ее следующим образом:  $\text{DIF}(\text{SIN}(a \cdot x^2), x)$ . Затем нажать Simplify и на экране выведется:

$$0.5 \cdot \text{ATAN}(b \cdot x^2 / a)$$

---


$$a \cdot b$$

**Вычислите аналогичным способом следующие интегралы функции:**

- |                          |                              |
|--------------------------|------------------------------|
| 1) $1 / x^2$ ;           | 6) $2 / x^3 + 3$ ;           |
| 2) $x / (2 + a^2) - 1$ ; | 7) $a / b + (x^2 + 1)$ ;     |
| 3) $a^3 + x / 0.5$ ;     | 8) $x / 2 + a \cdot b$ ;     |
| 4) $(a + b)^2 / 2$ ;     | 9) $x^3 - 4 \cdot (a + b)$ ; |
| 5) $x^2 / (a + b)$ ;     | 10) $x^2 / 2a \cdot b$ .     |

Тема 3. Вычисление пределов функций

**Вычислите предел функции**  $(x + h)^2 - x^2 / ((x + h) - x)$ .

Форма записи предела :  $\text{LIM}(u, x, a)$ .

Для того, чтобы вычислить предел функции, необходимо записать его следующим образом:  $\text{LIM}(((x + h)^2 - x^2) / ((x + h) - x), h, 0)$ . Затем нажать Simplify и на экране выведется:

2·x.

**Вычислите аналогичным способом следующие пределы функции:**

- |                           |                       |
|---------------------------|-----------------------|
| 1) $(a \cdot x / (x+1));$ | 6) $2 / x^3 + 2/x;$   |
| 2) $x / \sin(x);$         | 7) $x / y + (x^2+1);$ |
| 3) $x^2 + y^2;$           | 8) $x / 2 + 3y^2;$    |
| 4) $(a+x)^2 / 2;$         | 9) $x^2 / y;$         |
| 5) $x^2 / (a+3);$         | 10) $y + x^2/2.$      |

#### Тема 4. Вычисление производений членов ряда

**Вычислите произведение членов ряда  $n^2$**

Форма записи вычисления произведения членов ряда :

PRODUCT(u, n)

PRODUCT(u, n, k, m), где k и m – это значения от k до m.

Для того, чтобы вычислить произведение членов ряда, необходимо записать его следующим образом: PRODUCT( $n^2$ , n). Затем нажать Simplify и на экране выведется:

$$(n-1)!^2.$$

**Вычислите аналогичным способом произведения членов ряда следующих функций:**

- |               |                 |
|---------------|-----------------|
| 1) $k^2 + 1;$ | 6) $2 / n^3;$   |
| 2) $x^2 / 2;$ | 7) $x + 2k;$    |
| 3) $n^3 + 2;$ | 8) $n^2 / 2x;$  |
| 4) $(k+x)^2;$ | 9) $x^2 / n;$   |
| 5) $n / x^2;$ | 10) $2y + x^2.$ |

#### Тема 5. Представление ряда в виде вектора

**Представьте ряд в виде вектора  $x^2$ .**

Форма записи вектора :

VECTOR(u, k, n), где n – это значение, до которого необходимо представить ряд.

VECTOR(u, k, m, n)

VECTOR(u, k, m, n, s), где s – это значение шага.

Для того, чтобы представить ряд в виде вектора, необходимо записать его следующим образом: VECTOR( $x^2$ , x, 5). Затем нажать Simplify и на экране выведется:

$$[1, 4, 9, 16, 25].$$

**Представьте аналогичным способом ряд в виде вектора следующих функций:**

- |               |                   |
|---------------|-------------------|
| 1) $x^2;$     | 6) $2 + (1/k^3);$ |
| 2) $1 / x^2;$ | 7) $2k + k^2;$    |
| 3) $k^3 + 2;$ | 8) $1 / k^2;$     |

4)  $(x+2)^2$ ;  
 5)  $5 / x^2$ ;

9)  $k^2 / 1$ ;  
 10)  $3k + x^2$ .

### Тема 6. Вычисление суммы членов ряда

**6. Вычислите сумму ряда  $1/2^n$ .**

Форма записи вычисления суммы ряда:

SUM(u, n)

SUM(u, n, k, m).

Для того, чтобы вычислить сумму ряда, необходимо записать функцию следующим образом: SUM( $1/2^n$ , n). Затем нажать Simplify и на экране выведется:

$-2^{1-n}$ .

**Вычислите аналогичным способом сумму членов ряда следующих функций:**

1)  $1 / 2^n$ ;

6)  $2 / k^3$ ;

2)  $1 / k^2$ ;

7)  $2x + k$ ;

3)  $n^3 + 2$ ;

8)  $k^2 / 2x$ ;

4)  $(k+2)^2$ ;

9)  $k^2 / 1$ ;

5)  $k / x^2$ ;

10)  $2k + x^2$ .

### Тема 7. Возведение в степень

**Возведите в степень  $x^2$  в 5 рядов**

Форма записи возведения в степень функции:

TABLE(u, k, n), где n – это конечное значение возводимое в степень

TABLE(u, k, m, n)

TABLE(u, k, m, n, s), где s – это значение шага.

Для того, чтобы возвести в степень число, необходимо записать функцию следующим образом: TABLE( $x^2$ , x, 5). Затем нажать Simplify и на экране выведется:

• 1 1 •

• 2 4 •

• 3 9 •

• 4 16 •

• 5 25 •

**Возведите в степень аналогичным способом:**

1)  $(x+1)^2$ ;

6)  $1 / k^3$ ;

2)  $x^2 / 2$ ;

7)  $x + 2k$ ;

3) Sin (x);

8)  $1 / k^2$ ;

4)  $k^2$ ;

9)  $x^2 / 1$ ;

5)  $k / x^2$ ;

10)  $k + x^2$ .



## Тема 8. Разложение функций в ряд Тейлора

**Разложите функцию в ряд Тейлора  $e^x$  до 5 степени.**

Форма записи разложения функции в ряд: TAYLOR(u, x, a, n), где n – это конечная степень в которую необходимо возвести при разложении ряда.

Для того, чтобы разложить функцию в ряд Тейлора, необходимо записать функцию следующим образом: TAYLOR(#e^x, x, 0, 5). Затем нажать Simplify и на экране выведется:

$$\frac{x^5}{120} + \frac{x^4}{24} + \frac{x^3}{6} + \frac{x^2}{2} + x + 1.$$

**Разложите аналогичным способом в ряд Тейлора следующие функции:**

- |                    |                      |
|--------------------|----------------------|
| 1) $c * \sin(x)$ ; | 6) $2x / k^3$ ;      |
| 2) $x^2 + 1$ ;     | 7) $2k * 3x$ ;       |
| 3) $\sin(x) / 2$ ; | 8) $\cos(x) / x^2$ ; |
| 4) $1 / x^2$ ;     | 9) $1 + x^2$ ;       |
| 5) $2 / x^2$ ;     | 10) $x^2 + 2k$ .     |

## Тема 9. Аналитические методы решения дифференциальных уравнений первого порядка

### Функция DSOLVE1

**1. Решите уравнение первого порядка:  $2 \cdot x \cdot y + (1 + x^2) \cdot y' = 0$**

Вид уравнения:  $p(x, y) + q(x, y) \cdot y' = 0$ .

Форма записи:

DSOLVE1(p, q, x, y, x0, y0), где p, q — функции, содержащиеся в ДУ рассматриваемого вида, x0, y0 — начальные условия решения ДУ;

DSOLVE1\_GEN(p, q, x, y, c), где c — произвольная постоянная.

Первая форма применяется для получения частного решения. Вторая — для получения общего решения.

Для решения уравнения, необходимо записать его следующим образом:

DSOLVE1(2·x·y, 1 + x^2, x, y, 0, 1). Затем нажать Simplify и на экране выведется:

$$x^2 \cdot y + y = 1.$$

**Решите аналогичным способом уравнения первого порядка с помощью функции DSOLVE1:**

- |                                   |                                    |
|-----------------------------------|------------------------------------|
| 1) $2xy + (3+x) \cdot y' = 0$ ;   | 6) $2xy + (1+x^2) \cdot y' = 0$ ;  |
| 2) $3xy + (1+2y) \cdot y' = 0$ ;  | 7) $3xy + (3+x^2) \cdot y' = 0$ ;  |
| 3) $2x + (2+x^2) \cdot y' = 0$ ;  | 8) $5x + (2+xy) \cdot y' = 0$ ;    |
| 4) $4xy + (1+x^2) \cdot y' = 0$ ; | 9) $5xy + (1+x^2) \cdot y' = 0$ ;  |
| 5) $2x + (2+x^2) \cdot y' = 0$ ;  | 10) $4xy + (1+x^2) \cdot y' = 0$ . |

### Функция LINEAR1

**2. Решите уравнение первого порядка:  $b \cdot y(x) / b \cdot x + a \cdot y = 0$ , при начальных условиях  $y(0) = 1$ .**

В данном случае:

$$p = a, q = 0, x_0 = 0, y_0 = 1$$

Вид уравнения:  $y' + p(x) \cdot y = q(x)$ .

Форма записи:

LINEAR1 (p, q, x, y, x0, y0);

LINEAR1\_GEN (p, q, x, y, c).

Первая форма применяется для получения частного решения. Вторая — для получения общего решения.

Для решения уравнения, необходимо записать его следующим образом:

LINEAR1 (a, 0, x, y, 0, 1), LINEAR1\_GEN (a, 0, x, y, c).

Затем нажать Simplify и на экране выведется:

$$y = e^{-a \cdot x} - \text{частное решение.}$$

$$y = c \cdot e^{-a \cdot x} - \text{общее решение.}$$

**Решите аналогичным способом уравнения первого порядка с помощью функции *LINEAR1*:**

1)  $y' + 2x^2 \cdot 2y = 0$ ;

6)  $y' + 5x \cdot 2y = 0$ ;

2)  $y' + 3x \cdot 3y = 0$ ;

7)  $y' + 6x^2 \cdot 2y = 0$ ;

3)  $y' + 4x \cdot 2y = 0$ ;

8)  $y' + 7x \cdot 3y = 0$ ;

4)  $y' + 5x^2 \cdot 2y = 0$ ;

9)  $y' + x^2 \cdot 5y = 0$ ;

5)  $y' + 4x^2 \cdot y = 0$ ;

10)  $y' + 2x \cdot 4y = 0$ .

### **Функция *SEPARABLE***

**3. Решите уравнение первого порядка:  $y' = (x^2 - 1) \cdot (y^2 + 1)$ , при начальных условиях  $y(0) = 1$ .**

В данном случае:

$$p = x^2 - 1, q = y^2 + 1, x_0 = 0, y_0 = 1.$$

Вид уравнения:  $y' = p(x) \cdot q(y)$ .

Форма записи:

SEPARABLE (p, q, x, y, x0, y0);

SEPARABLE\_GEN (p, q, x, y, c).

Первая форма применяется для получения частного решения. Вторая — для получения общего решения.

Для решения уравнения, необходимо записать его следующим образом:

SEPARABLE ( $x^2 - 1, y^2 + 1, x, y, 0, 1$ ),

SEPARABLE\_GEN ( $x^2 - 1, y^2 + 1, x, y, c$ ).

Затем нажать Simplify и на экране выведется:

$$\frac{y}{2y+1} - \frac{1}{2y+1} = x(2x-1) \quad - \text{частное решение.}$$

$$\frac{y}{2y+1} = x(2x-1) + c \quad - \text{общее решение.}$$

**Решите аналогичным способом уравнения первого порядка с помощью функции *SEPARABLE*:**

1)  $y' = (x+1) \cdot (y^2 - 2)$ ;

6)  $y' = (2x^2+1) \cdot (y^2 - 2)$ ;

- |                               |                                |
|-------------------------------|--------------------------------|
| 2) $y' = (2x-1) * (y^2+3);$   | 7) $y' = (x^2+5) * (2y^2-4);$  |
| 3) $y' = (3x^2+1) * (y^2-2);$ | 8) $y' = (2x^2+1) * (y^2+3);$  |
| 4) $y' = (x^2+1) * (y^2+2);$  | 9) $y' = (x+1) * (2y^2-5);$    |
| 5) $y' = (x+4) * (y^2-3);$    | 10) $y' = (2x^2+1) * (y^2+4);$ |

### **Функция *BERNOULLI***

**2. Решите уравнение первого порядка:**  $y' + (x + 1)y = x*y^k$ , при начальных условиях  $y(0) = 1, k = 2$ .

В данном случае:

$$p = x + 1, q = x$$

#### **Примечание**

Обозначения переменных в этой функции те же, что и в функциях, рассмотренных выше. Первая форма функции дает частное решение при заданных начальных условиях  $x_0$  и  $y_0$ . Вторая — общее решение.

$k$  и  $c$  — это константы.

Вид уравнения:  $y' + p(x)y = q(x)*y^k$

Форма записи:

`BERNOULLI_ODE (p, q, k, x, y, x0, y0);`

`BERNOULLI_ODE_GEN (p, q, k, x, y, c).`

Первая форма применяется для получения частного решения. Вторая — для получения общего решения.

Для решения уравнения, необходимо записать его следующим образом:

`BERNOULLI_ODE ( x+1, x, k, x, y, 0, 1),` `BERNOULLI_ODE_GEN (x+1, x, k, x, y, c).` Затем нажать `Simplify` и на экране выведутся ответы при  $k = 2$ .

**Решите аналогичным способом уравнения первого порядка с помощью функции *BERNOULLI*:**

- |                                    |                                      |
|------------------------------------|--------------------------------------|
| 1) $y' + (3x - 1) * y = 2x * y^k;$ | 6) $y' + (2x^2+4) * y = 3x * y^k;$   |
| 2) $y' + (x^2 + 2) * y = x * y^k;$ | 7) $y' + (4x^2+2) * y = 2x * y^k;$   |
| 3) $y' + (2x+2) * y = 4x * y^k;$   | 8) $y' + (5x^2 + 1) * y = x * y^k;$  |
| 4) $y' + (x^2+5) * y = x * y^k;$   | 9) $y' + (x^2+3) * y = 3x * y^k;$    |
| 5) $y' + (3x^2+2) * y = 2x * y^k;$ | 10) $y' + (x^2 + 4) * y = 3x * y^k.$ |

### **Тема 10. Аналитические методы решения дифференциальных уравнений второго порядка**

#### **Функция *DSOLVE2***

**1.Решите уравнение второго порядка:**  $y''+y'+y = x+ 1$ , при начальных условиях  $y(0) = 1, y'(0) = 1$ .

В данном случае:  $p = 1, q = 1, r = x + 1, x_0 = 0, y_0 = 1, x_1 = 0, y_1 = 1$ .

Вид уравнения:  $y'' + p(x)*y' + q(x)*y = r(x)$ .

Форма записи:

`DSOLVE2 (p, q, r, x, c1, c2);`

`DSOLVE2_BV (p, q, r, x, x0, y0, x2, y2);`

DSOLVE2\_IV (p, q, r, x, x0, y0, v0)

Первая форма применяется для получения общего решения. Вторая и третья — для получения частного решения.

Для решения уравнения, необходимо записать его следующим образом:

DSOLVE2 (1, 1, x+1, x, c1, c2), DSOLVE2\_BV (1, 1, x+1, x, 0, 1, 0, 1), DSOLVE2\_IV (1, 1, x+1, x, 0, 1, 0, 1).

Затем нажать Simplify и на экране выведется:

$e^{-0.5x} \cdot (c1 \cdot \cos(0.8660254037 \cdot x) + c2 \cdot \sin(0.8660254037 \cdot x)) + x$  — общее решение.

$e^{-0.5x} \cdot \cos(0.8660254037 \cdot x) + x$  — частное решение.

$e^{-0.5x} \cdot (\cos(0.8660254037 \cdot x) - 0.5773502691 \cdot \sin(0.8660254037 \cdot x)) + x$  — частное решение.

## 2. Решите аналогичным способом уравнения второго порядка с помощью функции *DSOLVE2*:

1)  $y'' + 2y' + y = x+2$ ;

6)  $y'' + y' + y = x+1$ ;

2)  $y'' + y' + 3y = x^2 + 1$ ;

7)  $y'' + y' + 2y = x^2 + 1$ ;

3)  $y'' + y' + y = x+1$ ;

8)  $y'' + y' + 5y = 2x+1$ ;

4)  $y'' + y' + 4y = x+5$ ;

9)  $y'' + y' + 2y = x+3x^2 + 1$ ;

5)  $y'' + y' + 3y = x^2 + 1$ ;

10)  $y'' + y' + 4y = x^2 + 2x^2$ .

### Функция *AUTONOMOUS\_CONSERVATIVE*

#### 3. Решите уравнение второго порядка: $y'' = f(x, y)$ , если:

d)  $f(x, y) = y + 1$ ;

e)  $f(x, y) = x + 1$ ;

f)  $f(x, y) = x + y + 1$ .

Решение можно получить при нулевых начальных условиях, то есть  $y(0) = 0, y'(0) = 0$ .

В данном случае  $q_a = -y + 1, q_b = x + 1, q_c = x + y + 1, x_0 = 0, y_0 = 0, y' = 1$ .

Вид уравнения:  $y'' = q(y)$ , при этом  $q = f(x, y)$ ;

$x_0, y_0, y'_0$  — начальные условия  $y(x_0) = y_0, y'(x_0) = y'_0$ ;

$x, y$  — аргумент и искомая функции;

$y' = v_0$

Форма записи:

AUTONOMOUS\_CONSERVATIVE (q, x, y, xO, yO, v0).

Для решения уравнения, необходимо записать его следующим образом:

a) AUTONOMOUS\_CONSERVATIVE(y+1, x, y, 0, 0, 0);

b) AUTONOMOUS\_CONSERVATIVE(x+1, x, y, 0, 0, 0);

c) AUTONOMOUS\_CONSERVATIVE(x+y+1, x, y, 0, 0, 0).

Затем нажать Simplify и на экране выведется:

a)  $x = \ln(\sqrt{y(y+2)} + y + 1)$ ;

$$b) x = \frac{\sqrt{2} \cdot \sqrt{y \cdot (x + 1)}}{x + 1}$$

$$c) x = \text{LN}(\sqrt{y \cdot (y + 2 \cdot (x + 1))}) + y + x + 1 - \text{LN}(x + 1)$$

**Решите аналогичным способом уравнение второго порядка с помощью функции *AUTONOMOUS\_CONSERVATIVE*:**

$y'' = f(x, y)$ , при:

- |                 |                 |
|-----------------|-----------------|
| 1) $2y + 1$ ;   | 6) $x + 6$ ;    |
| 2) $2x^2 + 1$ ; | 7) $2x^2 + 1$ ; |
| 3) $3x + 1$ ;   | 8) $2x + 1$ ;   |
| 4) $x + 5$ ;    | 9) $x + 3$ ;    |
| 5) $x^2 + 1$ ;  | 10) $x^2 + 2$ . |

### **Функция *LIOUVILLE***

**4. Решите уравнение второго порядка:  $y'' + y' + (y') = 0$ .**

В данном случае  $p = 1$ ,  $q = -1$ .

Вид уравнения:  $y'' + p(x) \cdot y' + q(y) \cdot (y')^2 = 0$

Форма записи: *LIOUVILLE* ( $p, q, x, y, c1, c2$ ).

Для решения уравнения, необходимо записать его следующим образом:

*LIOUVILLE*(1,-1,x,y,c1,c2).

Затем нажать *Simplify* и на экране выведется:  $e^{-y} + c2 \cdot e^{-x} + c1 = 0$ .

**Решите аналогичным способом уравнение второго порядка с помощью функции *LIOUVILLE*:**

$y'' + y' + (y') = 0$ , при:

- |                       |                      |
|-----------------------|----------------------|
| 1) $p = 5, q = -1$ ;  | 6) $p = 3, q = 3$ ;  |
| 2) $p = 2, q = -2$ ;  | 7) $p = 2, q = 4$ ;  |
| 3) $p = -4, q = -1$ ; | 8) $p = 3, q = -2$ ; |
| 4) $p = -3, q = -1$ ; | 9) $p = -3, q = 6$ ; |
| 5) $p = 6, q = -1$ ;  | 10) $p = 7, q = 1$ . |

## **Тема 11. Численные методы решения дифференциальных уравнений**

### **Функция *EULER***

**1. Решите уравнение первого порядка:  $y' = 26/(3+(x+y)^2)$**

Вид уравнения:  $y' = r(x, y)$ .

Форма записи: *EULER\_ODE*( $r, x, y, x0, y0, h, n$ ), где  $r(x, y)$  – правая часть дифференциального уравнения, разрешенного относительно производной;

$x, y$  – искомые неизвестные;

$x0, y0$  – начальные условия;

$h$  – шаг интегрирования;

$n$  – число итераций.

Для решения уравнения, необходимо записать его следующим образом:

EULER\_ODE (26/(3+(x+y)<sup>2</sup>), x, y, 0, 1, 0.2, 2) .

Затем нажать Approximate и на экране выведется:

0     1  
0.2   2.3  
0.4   2.862162162

**Решите аналогичным способом уравнения первого порядка с помощью функции *EULER*:**

- |                                |                              |
|--------------------------------|------------------------------|
| 1) $y' = 12 / (x+y)^2$ ;       | 6) $y' = 8 / (x+y)^2$ ;      |
| 2) $y' = 6 / (5 + (x+y)^2)$ ;  | 7) $y' = 14 / (2+(x+y)^2)$ ; |
| 3) $y' = 13 / (2 + (x+y)^2)$ ; | 8) $y' = 10 / (x+y)^2$ ;     |
| 4) $y' = 20 / (x+y)^2$ ;       | 9) $y' = 17 / (1+(x+y)^2)$ ; |
| 5) $y' = 15 / (8+(x+y)^2)$ ;   | 10) $y' = 30 / (x+y)^2$ .    |

### **Функция *TAY\_ODE1***

**2. Решите уравнение первого порядка из предыдущего примера.:**

$$y' = 26/(3+(x+y)^2)$$

Вид уравнения:  $y' = r(x, y)$ .

Форма записи: TAYLOR\_ODE1(r, x, y, x0, y0, n), где  $r(x, y)$  – правая часть дифференциального уравнения, разрешенного относительно производной;

x, y – искомые неизвестные;

x0, y0 – начальные условия;

n – число итераций.

Для решения уравнения, необходимо записать его следующим образом:

$$\text{TAY\_ODE1}(26/(3+(x+y)^2), x, y, 0, 1, n) .$$

Затем нажать Approximate и на экране выведется:

$$\frac{26 \cdot x}{y^2 + 3} + 1 - \text{ответ при } n = 1$$

**Решите аналогичным способом уравнения первого порядка с помощью функции *TAY\_ODE1*:**

- |                                |                                 |
|--------------------------------|---------------------------------|
| 1) $y' = 10 / (1 + (x+y)^2)$ ; | 6) $y' = 5 / (6 + (x+y)^2)$ ;   |
| 2) $y' = 9 / (2 + (x+y)^2)$ ;  | 7) $y' = 4 / (7 + (x+y)^2)$ ;   |
| 3) $y' = 8 / (3 + (x+y)^2)$ ;  | 8) $y' = 3 / (8 + (x+y)^2)$ ;   |
| 4) $y' = 7 / (4 + (x+y)^2)$ ;  | 9) $y' = 2 / (9 + (x+y)^2)$ ;   |
| 5) $y' = 6 / (5 + (x+y)^2)$ ;  | 10) $y' = 1 / (10 + (x+y)^2)$ . |

### **Функция *PICARD***

**3. Решите уравнение первого порядка.:** Пусть решением уравнения  $dy(x, y) / dx = x/y + 3e^{-2x}$ , является решение первой степени  $3x + 1$ ,

Вид уравнения:  $y' = r(x, y)$ .

Форма записи: PICARD(f(x,y),p,x,y,x0,y0).

Для решения уравнения, необходимо записать его следующим образом:

PICARD(x/y+3e<sup>-2x</sup>,3x+1,x,y,0,1) .

Затем нажать Approximate и на экране выведется:

$$\frac{-2 \cdot x}{3 \cdot e} + \frac{1 \cdot \text{LN}(3 \cdot x + 1)}{9} + \frac{3}{2 \cdot \text{LN}(e)} + \frac{1 \cdot \text{LN}(1)}{9} + \frac{x}{3} + 1$$

Решите аналогичным способом уравнение первого порядка с помощью функции **PICARD**, при этом пусть решением уравнения  $dy(x, y) / dx = x/y + 3e^{-2x}$ , является решение первой степени:

- |               |                |
|---------------|----------------|
| 1) $2x + 3$ ; | 6) $4x + 3$ ;  |
| 2) $3x + 2$ ; | 7) $x + 8$ ;   |
| 3) $4x + 1$ ; | 8) $7x + 1$ ;  |
| 4) $5x + 2$ ; | 9) $3x + 4$ ;  |
| 5) $6x + 1$ ; | 10) $5x + 3$ . |

### Функция RK

**4. Решите уравнение второго порядка:**  $y'' + xy' - 2y + 1 = 0$ , при начальных условиях  $y(0)=1, y'(0)=2$ .

В данном случае:  $y' = z$ . Тогда система дифференциальных уравнений будет иметь вид:

$$\begin{aligned} y' &= z ; \\ z' &= -xz + 2y - 1. \end{aligned}$$

Вид уравнения: RK (r, v, v0, h, n), где:

r — вектор правых частей уравнений, разрешенных относительно производных (при этом система уравнений представляется в виде, когда каждое из уравнений является уравнением первого порядка);

v — вектор неизвестных;

v0 — вектор начальных условий;

h — шаг интегрирования;

n — число итераций.

Форма записи: RK (r, v, v0, h, n).

Для решения уравнения, необходимо записать его следующим образом:

$$\text{RK}([z, -x \cdot z + 2 \cdot y - 1], [x, y, z], [0, 1, 2], 0.2, 10) .$$

Затем нажать Approximate и на экране выведется:

0	1	2
0.2	1.422654	2.239849199
0.4	1.901148032	2.557857621
0.6	2.450708602	2.949502217
0.8	3.085402653	3.407818874
1	3.817721207	3.924150260
1.2	4.658325186	4.488996498
1.4	5.615960189	5.092832813
1.6	6.697521979	5.726783792

1.8 7.908236609 6.383088450

2 9.251910923 7.055339819

**Решите аналогичным способом уравнения первого порядка с помощью функции *RK*:**

1)  $y'' + xy' - 1y + 5 = 0;$

2)  $y'' + xy' - 2y + 9 = 0;$

3)  $y'' + xy' - 3y + 8 = 0;$

4)  $y'' + xy' - 4y + 7 = 0;$

5)  $y'' + xy' - 5y + 6 = 0;$

6)  $y'' + xy' - 0,5y + 10 = 0;$

7)  $y'' + xy' - 6y + 4 = 0;$

8)  $y'' + xy' - 7y + 3 = 0;$

9)  $y'' + xy' - 8y + 2 = 0;$

10)  $y'' + xy' - 9y + 1 = 0.$



## Лабораторный практикум по изучению узлов ЭВМ в системе Electronics Workbench

### Цифровой компаратор

**Цель:** Изучение назначения устройства и принцип работы цифрового компаратора.

**Оборудование:** Электронная лаборатория Electronics Workbench.

#### Краткая теория

Цифровые компараторы (от английского compare – сравнивать) выполняет сравнение двух чисел А, В одинаковой разрядности, заданных в двоичном или двоично-десятичном коде. В зависимости от схемного исполнения компараторы могут определять равенство  $A=B$  или неравенства  $A<B, A>B$ . Результат сравнения отображается в виде логического сигнала на одноименных выходах, в случае выполнения условия на выходе 1.

Цифровые компараторы применяются для выявления нужного числа (слова) в цифровых последовательностях, для выполнения условных переходов.

Схемы одноразрядных компараторов приведены на рис. П.2.1.

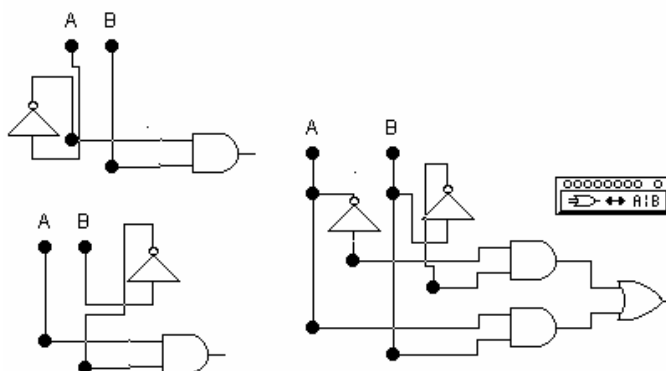


Рис. П.2.1

Операциям сравнения ( $A<B, A=B, A>B$ ) соответствуют структурные формулы ( $A' * B, A' * B' + A * B, A * B'$ ).

#### Контрольные вопросы и задания.

1. Какие функции выполняет цифровой компаратор, в каких устройствах он может быть использован?
2. Подсоединив схемы к логическому конвектору исследуйте приведенные схемы.
3. Составьте схему устройства объединяющее все три компаратора.
4. Составьте схемы устройств, удовлетворяющее условиям:  $A \leq B, A \neq B, A \geq B$ .
5. Исследуйте составленные схемы устройств.
6. Составьте структурные формулы и таблицы истинности для составленных выше логических схем цифровых компараторов.

## Устройство контроля четности

**Цель:** Изучение назначения и принцип работы устройства контроля четности.

**Оборудование:** Электронная лаборатория Electronics Workbench.

### Краткая теория

Операция контроля четности двоичных чисел позволяет повысить надежность передачи и обработки информации. Ее сущность заключается в суммировании по модулю 2 всех разрядов с целью выяснения четности числа, что позволяет выявить наиболее вероятную ошибку в одном из разрядов двоичной последовательности. Например, если при передаче кода 1001 произойдет сбой во втором разряде, то на приемном пункте получим код 1101 – такую ошибку определить в общем случае затруднительно. Если же код относится к двоично-десятичному (способ кодирования десятичных чисел, при котором каждая цифра представляется четырьмя двоичными разрядами – двоичной тетрадой).

Обнаружение ошибок путем введения дополнительного бита четности происходит следующим образом. На передающей стороне передаваемый код анализируется и дополняется контрольным битом до четного или нечетного числа единиц в суммарном коде. Соответственно суммарный код называется четным или нечетным. В случае нечетного кода дополнительный бит формируется таким образом, чтобы сумма всех единиц в передаваемом коде, включая контрольный бит, была нечетной. При контроле четности все наоборот. Например, в числе 0111. Число единиц нечетно. Поэтому при контроле нечетности дополнительный код должен быть нулем, а при контроле четности – единицей. На практике чаще всего используется контроль нечетности, поскольку он позволяет фиксировать полное пропадание информации (случай нулевого кода во всех информационных разрядах). На приемной стороне производится проверка кода четности. Если он правильный, то прием разрешается, в противном случае включается сигнализация ошибки или посылается передатчику запрос на повторную передачу.

Схема формирования бита четности для четырехразрядного кода приведена на рис. Она содержит четыре элемента Иключающее ИЛИ, выполняющее функции сумматоров по модулю 2 (без переноса) и состоит из трех ступеней. На первой ступени попарно суммируются все биты исходного кода на входах А, В, С, D. На второй ступени анализируются сигналы первой ступени и устанавливается четность или нечетность суммы входного кода. На третьей ступени полученный результат сравнивается с контрольным сигналом на входе E, задающим вид используемого контроля, в результате чего на выходе F формируется дополнительный пятый бит четности сопровождающий информационный сигнал в канале передачи.

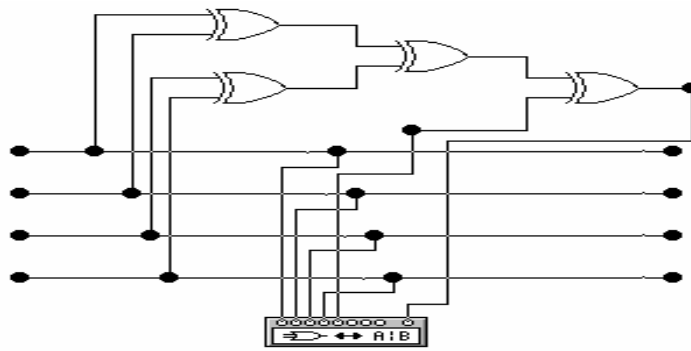


Рис. П.2.3.

Результаты моделирования приведены на рис.П.2.4.

	A	B	C	D	E	F	G	H	Out
000	0	0	0	0	0				0
001	0	0	0	0	1				1
002	0	0	0	1	0				1
003	0	0	0	1	1				0
004	0	0	1	0	0				1
005	0	0	1	0	1				0
006	0	0	1	1	0				0
007	0	0	1	1	1				1
008	0	1	0	0	0				0
009	0	1	0	0	1				0
010	0	1	0	1	0				0
011	0	1	0	1	1				1
012	0	1	1	0	0				0
013	0	1	1	0	1				1
014	0	1	1	1	0				1
015	0	1	1	1	1				0

$A'B'C'D'E + A'B'C'D'E' + A'B'CD'E' + A'B'CD'E + A'BC'D'$

Рис.П.2.4

### Контрольные вопросы и задания

1. Какое назначение имеют формирователи кода четности, где они могут быть использованы?
2. Постройте схему формирователя бита четности трехразрядного (пятиразрядного) кода.
3. Проанализируйте работу составленных схем формирователей битов четности.

### Мультиплексоры и демультиплексоры

**Цель:** Изучение назначения и принцип работы устройств мультиплексора и демультиплексора.

**Оборудование:** Электронная лаборатория Electronics Workbench.

### Краткая теория

Назначение мультиплексоров (от английского multiplex – многократный) – коммутировать в заданном порядке сигналы, поступающие с нескольких входных шин в одну выходную. У мультиплексора может быть, например, 16 входов и один выход. Это означает, что если к этим входам присоединить 16 источников цифровых сигналов – генераторов последова-

тельных цифровых слов, то байты от любого из них можно передавать на единственный выход. Для выбора любого из 16 каналов необходимо иметь 4 входа селекции ( $2^4=16$ ), на которые подается двоичный адрес канала. Так, для передачи данных от канала номер 9 на входах селекции необходимо установить код 1001. В силу этого мультиплексоры часто называют селекторами или селекторами-мультиплексорами.

На рис. П.2.5 приведена схема двухканального мультиплексора, состоящего из элементов ИЛИ, НЕ и двух элементов И.

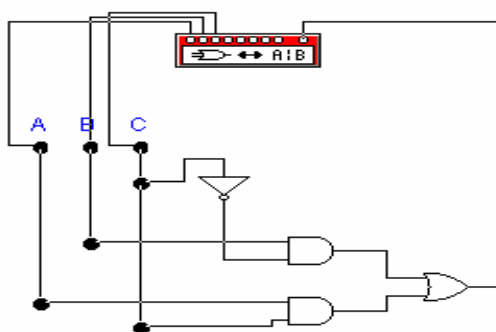


Рис. П.2.5. Схема двухканального мультиплексора

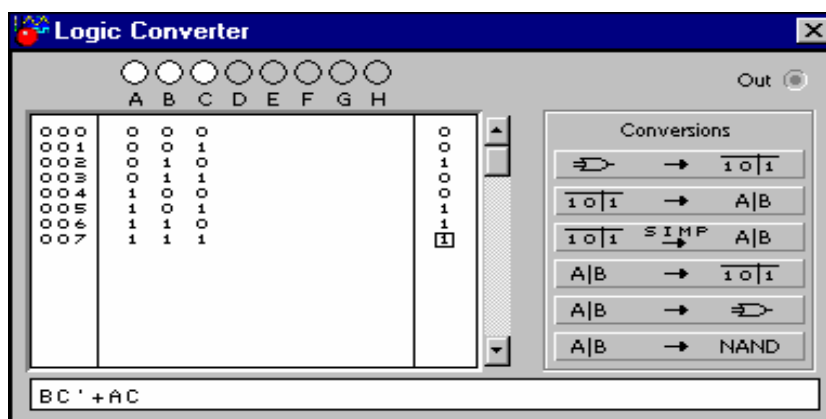


Рис. П.2.6

Результаты моделирования двухканального мультиплексора с помощью логического конвертора показаны на рис. П.2.6. Здесь его выходной сигнал описывается структурной формулой  $B \cdot C' + A \cdot C$ , т.е. сигнал из канала A проходит на выход при адресном входе  $C=1$ , а из канала B - при  $C=0$ , что и соответствует логике работы мультиплексора.

Демультимплексоры в функциональном отношении противоположны мультиплексорам. С их помощью сигналы с одного информационного входа распределяются в требуемой последовательности по нескольким выходам. Выбор нужной выходной шины, как и в мультиплексоре, обеспечивается установкой соответствующего кода на адресных входах. При  $m$  адресных входах демультимплексор может иметь до  $2^m$  выходов.

Принцип работы демультимплексора поясним с помощью схемы на рис.П.2.7.

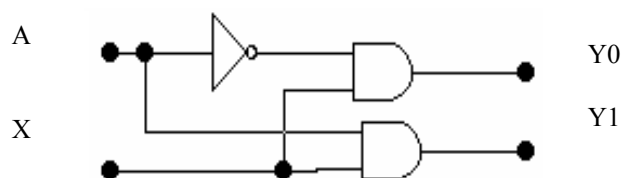


Рис.П.2.7. Схема демультиплексора

Схема содержит два элемента элемента И и один элемент НЕ. На схеме: X - информационный вход, A - вход адреса, Y0, Y1 - выходы.

Если  $A=0$  сигнал информационного входа передается на выход Y0, а при  $A=1$  - на выход Y1.

### Контрольные вопросы и задания

1. Что такое мультиплексор, каково его назначение?
2. Что такое демультиплексор, для решения каких задач его можно применить?
3. Придумайте схему трехканального мультиплексора?
4. Придумайте схему трехвыходного демультиплексора?

### Арифметические сумматоры

**Цель:** Изучение назначения и принцип работы устройств полусумматора и сумматора. Знакомство с базовыми элементами полусумматора и полного сумматора из библиотеки EWB.

**Оборудование:** Электронная лаборатория Electronics Workbench.

### Краткая теория

Арифметические сумматоры являются составной частью так называемых арифметико-логических устройств (АЛУ) микропроцессоров. В программе EWB арифметические сумматоры представлены в библиотеке Digital двумя базовыми устройствами: полусумматорами и полными сумматорами. Они имеют следующие назначения выводов: A, B – входы слагаемых,  $\Sigma$  - результат суммирования,  $C_0$  – выход переноса,  $C_i$  - вход переноса. N – разрядный сумматор создается на базе одного полусумматора и n-1 полных сумматоров. На рис.П.2.7 приведено исследование полусумматора.

На рис.П.2.8 приведена схема подключения полного сумматора к логическому конвертору и трехразрядный сумматор.

### Контрольные вопросы и задания.

1. Чем отличается полусумматор от полного сумматора.
2. Выясните внутреннюю структуру полного сумматора, пользуясь схемой подключения к логическому конвертору, аналогично приведенному анализу полусумматора.

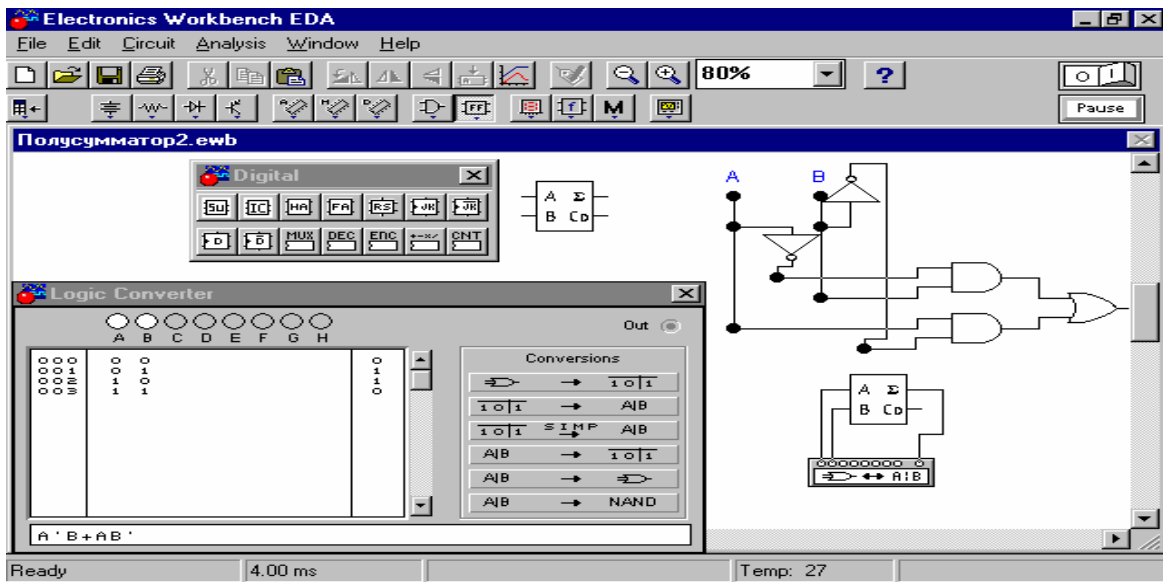


Рис. П.2.7

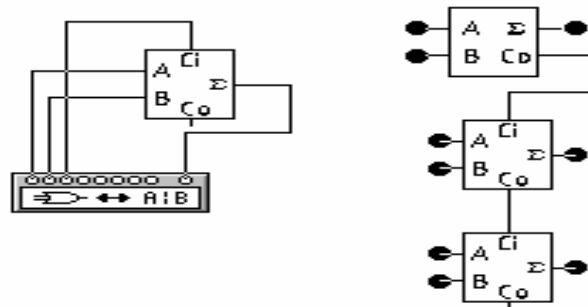


Рис. П.2.8

3. Исследуйте выходы переносу полусумматоров и полного сумматора.
4. Исследуйте приведенный трехразрядный сумматор последовательно подключая выходы к логическому конвертору.

### Виртуальный генератор слова (Word Generator)

**Цель:** Изучение назначения и принцип работы виртуального генератора слова. Знакомство с базовыми функциями виртуального генератора слов. Анализ работы сумматора с помощью виртуального генератора слов.

**Оборудование:** Электронная лаборатория Electronics Workbench.

### Краткая теория

Внешний вид виртуального генератора слова и лицевая панель приведены на рис. П.2.9. Генератор (его еще кодовым генератором) предназначен для генерации 16 разрядных двоичных слов. Кодовые комбинации необходимо задавать в шестнадцатиричном коде.

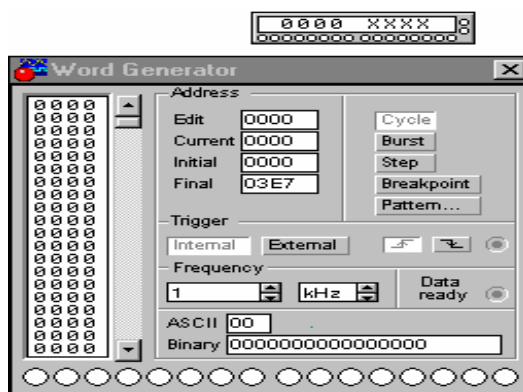


Рис.П.2.9

Каждая комбинация заносится с помощью клавиатуры, номер редактируемой ячейки фиксируется в окошке EDIT блока ADDRESS. Всего таких ячеек и следовательно, комбинаций – 2048. В процессе работы генератора в отсеке ADDRESS индицируется номер текущей ячейки (CURRENT), ячейки инициализации или начала работы (INITIAL) и конечной ячейки (FINAL). Выдаваемые на 16 выходов (В нижней части генератора) кодовые комбинации индицируются в текстовом (ASCII) и двоичном коде (BINARY).

Сформированные слова выдаются на 16 расположенных в нижней части виртуального прибора клемм-индикаторов:

- В пошаговом (при нажатии кнопки STEP), циклическом (при нажатии кнопки CYCLE) или с выбранного слова до конца (при нажатии клавиши BURST) при заданной частоте посылок (установка – нажатиями кнопок в окнах FREQUENCY);
- При внутреннем (при нажатии кнопки INTERNAL) или внешнем запуске (при нажатии кнопки EXTERNAL по готовности данных (клемма DATA READY), рядом расположена клемма для подключения канала синхронизации);
- При запуске по переднему или заднему фронту.

На клемму CLK выдается выходной синхронизирующий импульс. К органам управления относится также кнопка BREAK POINT – прерывание работы генератора в указанной ячейке. При нажатии на кнопку PATTERN выпадает меню (рис. П.2.10), где:

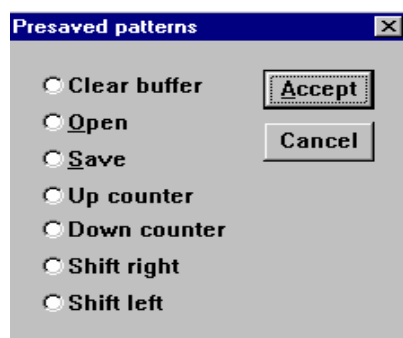


Рис. П.2.10

- Clear buffer – стереть содержимое буфера (содержимое буфера экрана);
- Open – загрузить кодовые комбинации (из файла с расширением .dp);
- Save – записать все набранные на экране комбинации в файл (.dp);
- Up counter - заполнить буфер экрана кодовыми комбинациями, начиная с 0 в нулевой ячейке и далее с прибавлением 1 в каждой последующей ячейке;
- Down counter – заполнить буфер кодовыми комбинациями, начиная с FFFF в нулевой ячейке и далее с уменьшением на 1 в каждой последующей ячейке;
- Shift right – заполнить каждые четыре ячейки комбинациями 1-2-4-8 со смещением их в следующих четырех ячейках вправо;
- Shift left – тоже самое, но со смещением влево.

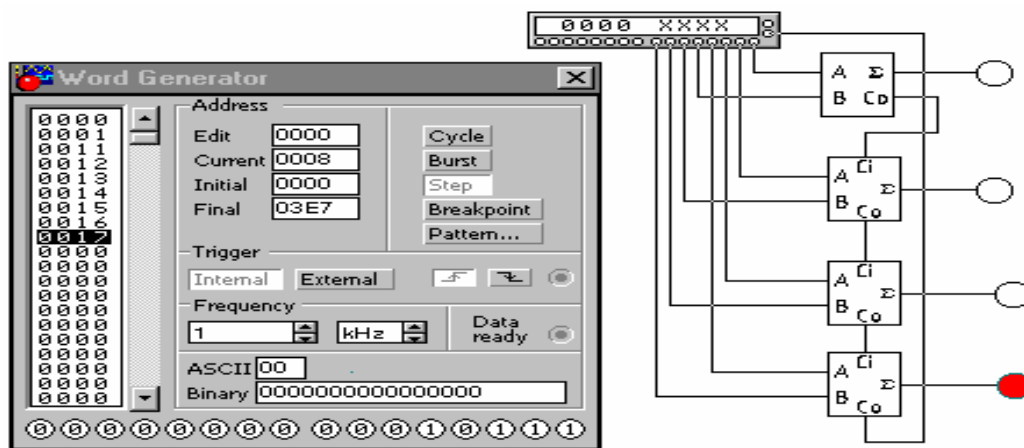


Рис. П.2.11

**Задание.** Исследовать 4-х разрядный сумматор (рис. П.2.11) с помощью виртуального генератора слов.

1. Собираем 4-х разрядный сумматор, состоящий из одного полусумматора и 3-х полных сумматоров.
2. Из панели Indicators выбираем 4 индикатора и подключаем к выходам сумматора (при сигнале 1 – зажигаются).
3. Выходы виртуального генератора слов подключаем входам сумматора; четыре первых разряда соответствуют первому слагаемому, четыре последующих разряда – второму слагаемому.
4. В лицевой панели генератора вводим шестнадцатиразрядные числа 0, 1, 11, 12, 13, 14, 15, 16, 17.
5. Запустить генератор слов в режиме Step. Проанализировать работу сумматора.

**Контрольные вопросы и задания.**

1. Объяснить предназначение виртуального генератора слов.
2. Перечислить основные функции генератора слов.
3. Объяснить, почему выбраны шестнадцатиразрядные числа 0, 1, 11, 12, 13, 14, 15, 16, 17 для анализа работы сумматора.



4. Постройте 5-разрядный сумматор и проанализируйте с помощью генератора слов.
5. Какие шестнадцатиразрядные числа необходимо ввести для анализа работы сумматора?
6. Как обеспечить вычитание двух чисел?

### Счетчик

**Цель:** Изучение назначения и функции устройства счетчик. Знакомство с принципом работы устройства счетчик.

**Оборудование:** Электронная лаборатория Electronics Workbench.

### Краткая теория

Счетчиком называют устройство, сигналы, на выходе которого отображают число импульсов, поступивших на счетный выход. Триггер может служить примером простейшего счетчика. Такой счетчик считает до двух. Счетчик, образованный цепочкой из  $m$  триггеров, может подсчитать в двоичном коде  $2^m$  импульсов. Каждый из триггеров такой цепочки называют разрядом счетчика. Число  $m$  определяет количество разрядов двоичного числа, которое может быть записано в счетчик. Число  $K_{сч}=2^m$  называют коэффициентом (модулем) счета.

Информация снимается с прямых и (или) инверсных выходов всех триггеров. В паузах между входными импульсами триггеры сохраняют свои состояния, т. е. счетчик запоминает число входных импульсов.

Нулевое состояние всех триггеров принимается за нулевое состояние счетчика в целом. Остальные состояния складываются по числу поступивших входных импульсов. Когда число входных импульсов  $N_{вх} > K_{сч}$ , происходит переполнение, после чего счетчик возвращается в нулевое состояние и цикл повторяется. Коэффициент счета, таким образом, характеризует число входных импульсов, необходимое для одного цикла и возвращения в исходное состояние.

Счетчики различаются числом и типами триггеров, способами связей между ними, кодом, организацией счета и другими показателями. Цифровые счетчики классифицируются по следующим параметрам:

- Коэффициент счета – двоичные; двоично-десятичные или с другим основанием счета; с произвольным постоянным и переменным (программируемым) коэффициентом счета;
- Направление счета – суммирующие, вычитающие и реверсивные ( );
- Способ организации внутренних связей – с последовательным, параллельным или комбинированным переносом, кольцевые.

Классификационные признаки независимы и могут встречаться в различных сочетаниях: например, суммирующие счетчики бывают как с последовательным, так и с параллельным переносом, могут иметь двоичный, десятичный и иной коэффициент счета.

Схема четырехразрядного двоичного счетчика с последовательным пере-

носом на D – триггерах приведена на рис. П.2.12.

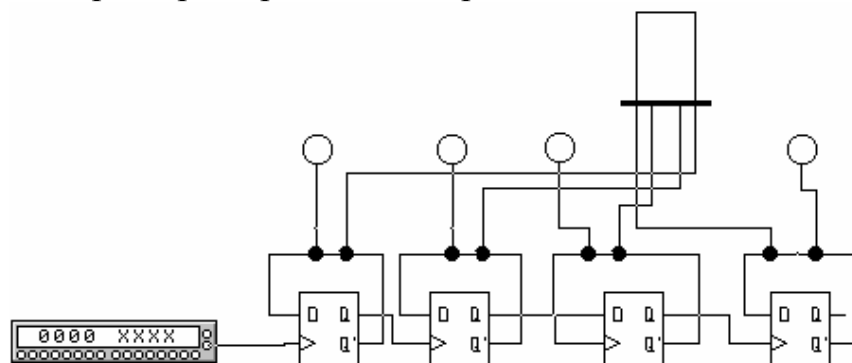


Рис. П.2.12

На вход счетчика подаются импульсы с выхода синхросигналов генератора слова, которые генерируются при каждом нажатием клавиши STEP. Каждый триггер счетчика осуществляет деление на 2, сигнал переноса передается последовательно от одного разряда к другому. Состояние разрядов счетчиков в двоичном коде индицируются логическим пробником (индикатором), а в десятичном – семисегментным индикатором.

#### Контрольные вопросы и задания.

1. Что такое счетчик, какие функции он может выполнять?
2. Назовите типы счетчиков и их возможные применения.
3. Смоделируйте приведенную выше схему и проанализируйте работу счетчика.

### Регистр

**Цель:** Изучение назначения и функций регистр. Знакомство с принципом работы регистр.

**Оборудование:** Электронная лаборатория Electronics Workbench.

#### Краткая теория

Регистры – устройства для временного хранения и преобразования информации в виде много разрядных двоичных чисел. Регистры наряду со счетчиками и запоминающими устройствами являются наиболее распространенными устройствами цифровой техники. При сравнительной простоте регистры обладают большими функциональными возможностями. Они используются в качестве управляющих и запоминающих устройств, генераторов и преобразователей кодов, счетчиков, делителей частоты, узлов временной задержки. Элементами структуры регистров являются триггеры D- или JK- типа с динамическим или статическим управлением. Одиночный триггер может запоминать (регистрировать) один разряд (бит) двоичной информации. Такой триггер можно считать одноразрядным регистром. Занесение информации в регистр называют операцией ввода или записи. Выдача информации к внешним устройствам характеризует операцию вывода или считывания. Запись информации в регистр не требует его предварительного обнуления.

Все регистры в зависимости от функциональных свойств подразделяются на две категории – накопительные (регистры памяти, хранения) и сдви-

гающие. В свою очередь, сдвигающие регистры делятся по способу ввода и вывода информации на параллельные и последовательно-параллельные и комбинированные, по направлению передачи (сдвига) информации – на односторонние и реверсивные.

На рис. П.2.13 показана схема простейшего четырехразрядного регистра на D – триггерах, в котором информация заносится последовательно, начиная с младшего разряда.

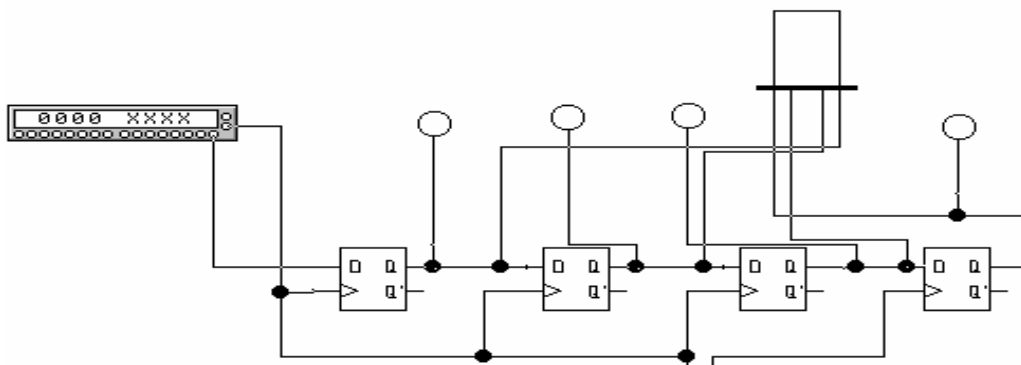


Рис. П.2.13

### Контрольные вопросы и задания.

1. Что такое регистр, какие функции он может выполнять?
2. Назовите типы регистров и их возможные применения.
3. Смодулируйте приведенную выше схему и проанализируйте работу регистра.

### Оперативное запоминающее устройство

**Цель:** Изучение назначения и функций оперативного запоминающего устройства. Знакомство с принципом работы оперативного запоминающего устройства.

**Оборудование:** Электронная лаборатория Electronics Workbench.

### Краткая теория

Оперативные запоминающие устройства (ОЗУ) являются неотъемлемой частью микропроцессорных систем различного назначения. ОЗУ делятся на два класса: статические и динамические. В статических ОЗУ запоминание информации производится на триггерах, а в динамических – на конденсаторах емкостью 0,5 пФ. Длительность хранения информации в статических ОЗУ не ограничена, тогда как в динамических ОЗУ она ограничена временем саморазряда конденсатора, что требует специальных средств регенерации и дополнительных затрат времени на этот процесс.

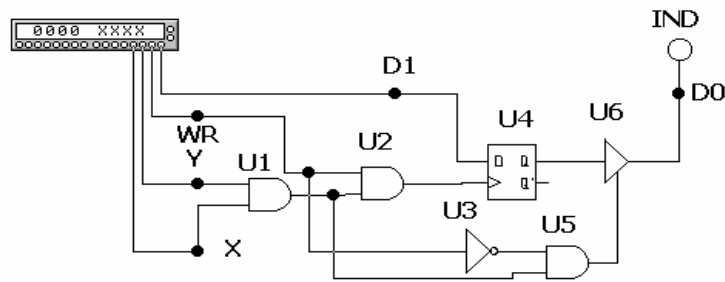


Рис. П.2.14

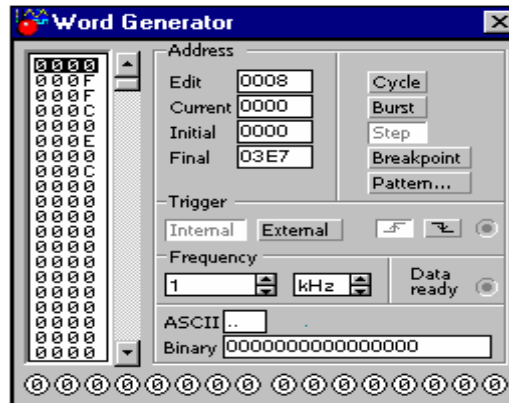


Рис. П.2.15

На рис. П.2.14 показана ячейка статического ОЗУ на D-триггере и вспомогательных логических элементах. Информационный вход ячейки подключен к шине данных D1 одного из разрядов, ее выход – к соответствующей шине D0 через элемент с тремя состояниями U6. Ячейка выбирается сигналами Y=1, X=1, поступающих с дешифратора адреса. При записи в ячейку памяти на D1 устанавливается 1 или 0, на входе WR/RD' – сигнал 1, в результате чего срабатывают элементы U1, U2. Положительный перепад сигнала с элемента U2 поступает на тактовый вход D-триггера U4 и в нем записывается 1 или 0 в зависимости от уровня сигнала на его D-входе. При чтении на входе WR/RD' устанавливается 0, при этом срабатывают элементы U1, U3, U5 и на вход РАЗРЕШЕНИЕ ВЫХОДА буферного элемента U6 поступает разрешающий сигнал, в результате чего сигнал с Q-выхода D-триггера передается на разрядную шину D0, состояние которой индицируется логическим пробником IND. Для проверки функционирования ячейки памяти используется генератор слова (рис. П.2.15), выходной код которого соответствует указанным режимам работы ячейки.










Заметим, что запоминающие устройства статического типа отличаются высоким быстродействием и в компьютерах используются в качестве так называемой кэш-памяти.

### Контрольные вопросы и задания.

1. Какие типы памяти существуют?
2. Чем отличается динамическая память от статической?
3. Смоделируйте и проанализируйте работу ОЗУ, схема которого приведена выше.

## Инструменты, палитры, элементы виртуального конструктора LabVIEW



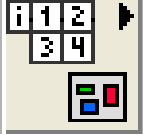

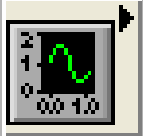

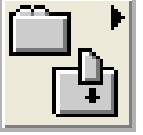
### 10.2.1. Краткие пояснения по инструментам

	<p>Инструмент <b>Управление</b> (Operate Value, «палец») используется для изменения значений элементов управления или ввода текста. При работе со строковыми элементами управления вид инструмента изменяется на следующий:</p>
	<p>Инструмент <b>Перемещение</b> (Position/Size/Select, «стрелка») служит для выбора, перемещения или изменения размеров объектов. Для изменения размеров в LabVIEW 7.0 используются подвижные прямоугольные элементы, появляющиеся в зависимости от допустимого направления изменения в центре сторон или на углах контура объекта при установке инструмента <b>Перемещение</b> внутри этого контура</p>
	<p>Инструмент <b>Редактирование текста</b> (Edit Text, «буква») используется для ввода и редактирования текста и создания свободных меток. При создании текстовых элементов вид инструмента изменяется:</p>
	<p>Инструмент <b>Соединение</b> (Connect Wire, «катушка») применяется для соединения объектов на блок-диаграмме. Он также используется для условного (невидимого) подключения элементов управления и индикаторов лицевой панели к терминалам соединительной панели ВП</p>
	<p>Инструмент <b>Контекстное меню объекта</b> (Object Shortcut Menu) вызывает контекстное меню соответствующего объекта при щелчке на нем ЛКМ</p>
	<p>Инструмент <b>Быстрая прокрутка окна</b> (Scroll Window) используется для просмотра окна без обращения к полосам прокрутки</p>
	<p>Инструмент <b>Контрольная точка</b> (Set/Clear Breakpoint) позволяет размещать и удалять контрольные точки на ВП, функциях, узлах, проводниках данных, структурах и приостанавливать в них выполнение программы</p>
	<p>Инструмент <b>Установка отладочных индикаторов</b> (Probe Data) позволяет наблюдать данные в проводниках блок-диаграммы при выполнении ВП</p>
	<p>Инструмент <b>Получить цвет</b> (Get Color, «пипетка») служит для копирования цвета с последующей вставкой с помощью инструмента <b>Установить цвет</b></p>



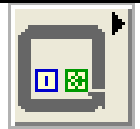


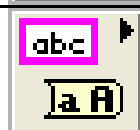
Инструмент **Установить цвет** (Set Color) предназначен для изменения цвета объекта. Он также отображает текущие цвета переднего и заднего плана

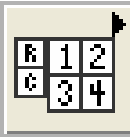
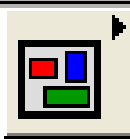


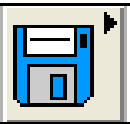

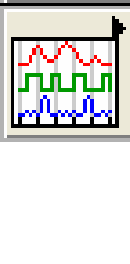


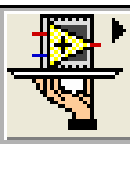
10.2.2. Палитра элементов содержит следующие подпалитры:

	<b>Числовые элементы</b> (Numeric). Элементы подпалитры используются в качестве источников или приемников числовых данных
	<b>Логические элементы</b> (Boolean). Подпалитра содержит набор различных переключателей, кнопок и индикаторов, имитирующих действие лампочек и светодиодов. Все элементы могут находиться в двух состояниях, отображающих два состояния логической функции: ИСТИНА (True) и ЛОЖЬ (False)
	<b>Строка и путь</b> (String & Path). Элементы подпалитры представляют типы данных, которые содержат последовательность литер, символов, массивов.
	<b>Массив и кластер</b> (Array & Cluster). Подпалитра содержит структуры, которые позволяют создавать массивы или кластеры элементов. Массивы и кластеры представляют упорядоченное множество элементов соответственно одного или различных типов. Элементами массива могут быть числовые или логические элементы, строки или кластеры. Тип элементов массива определяется типом данных, помещаемых из палитры элементов в шаблон массива.
	<b>Лист и таблица</b> (List & Table). Элементы подпалитры представляют собой управляющие или управляемые элементы, позволяющие заносить или отображать буквенную, символьную и цифровую информацию в виде набора строк или ячеек.
	<b>График</b> (Graph). Подпалитра содержит набор объектов, которые применяются для отображения временных или функциональных зависимостей реальных или расчетных сигналов.
	<b>Кольцевой список и перечень</b> (Ring & Enum). Элементы подпалитры представляют собой специальные числовые объекты, которые ставят в соответствие 16-битовым целым числам без знака строки, рисунки или то и другое.
	<b>Контейнеры</b> (Containers). Элементы подпалитры представляют объекты, внутри которых могут размещаться элементы управления и индикации, лицевые панели приборов и элементы ActiveX.


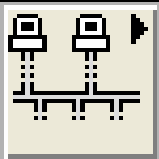

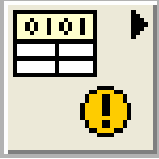
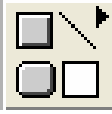


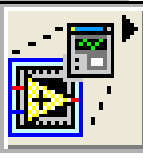
	<b>Ввод - вывод (I/O).</b> Подпалитра содержит элементы управления, используемые для установки параметров плат ввода-вывода данных.
	<b>Элементы диалога (Dialog Controls).</b> Подпалитра содержит набор элементов, поддерживающих диалоговые функции.
	<b>Классические элементы (Classic Controls).</b> Подпалитра содержит палитры элементов, выполненных в стиле предыдущих версий LabVIEW.
	<b>Ссылка (Refnum).</b> Подпалитра содержит идентификаторы, которые связаны с открытым приложением или файлом.
	<b>Оформление (Decorations).</b> Подпалитра служит для размещения на лицевой панели разнообразных графических элементов: линий, стрелок, рамок различной формы, надписей и т. п.
	<b>Выбор элемента управления (Select a Control).</b> Подпалитра служит для выбора элемента управления, отсутствующего в палитре.
	<b>Элементы пользователя (User Controls).</b> Подпалитра служит для выбора элементов управления, созданных пользователем и помещенных им в папку LabVIEW7.0\user.lib.
	<b>Экспресс (Express).</b> Подпалитра содержит палитры с наиболее часто применяемыми элементами.

**10.2.3.** Палитра функций панели блок-диаграммы содержит следующие основные подпалитры:

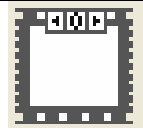
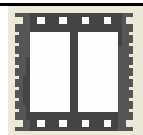
	Подпалитра <b>Структуры (Structures)</b> содержит набор структур, управляющих выполнением ВП. Перечень структур приведен в последующей таблице
	Подпалитра <b>Числовые (Numeric)</b> содержит полный набор математических функций, функций преобразования форматов чисел и набор констант.
	Подпалитра <b>Логические (Boolean)</b> включает набор функций <b>И (AND)</b> , <b>ИЛИ (OR)</b> , <b>Исключающее ИЛИ (Exclusive Or)</b> , <b>НЕ (Not)</b> и логические константы.
	Подпалитра <b>Строковые (String)</b> содержит ряд функций обработки строковых переменных, функции взаимного преобразования чисел и строк, а также строковые константы.

	<p>Подпалитра <b>Массив</b> (Array) включает функции выделения или замены элементов массивов, выделения подмассивов, транспонирования, сдвига, изменения размерности и т. п.</p>
	<p>Подпалитра <b>Кластер</b> (Cluster) включает функции формирования кластеров и массивов кластеров, выделения элементов из кластеров, а также кластерную константу.</p>
	<p>Функции подпалитры <b>Сравнение</b> (Comparison) формируют логическую переменную в зависимости от результата сравнения входных переменных или позволяют определять соотношение чисел, нахождение числа в заданном диапазоне, тип числа или тип символа.</p>
	<p>Функции подпалитры <b>Время и диалоги</b> (Time &amp; Dialog) позволяют задавать или определять временные интервалы или текущее время, формировать сообщения об ошибках и контролировать активность лицевой панели.</p>
	<p>Функции подпалитры <b>Файловый ввод/вывод</b> (File I/O) выполняют файловые операции записи и считывания данных.</p>
	<p>Подпалитра <b>Измерения NI</b> (NI Measurements) содержит наборы функций работы с платами ввода/вывода данных, ввода изображений, управления двигателями</p>
	<p>Подпалитра <b>Осциллограмма</b> (Waveform) включает разделы <b>Аналоговая осциллограмма</b> (Analog Waveform), <b>Цифровая осциллограмма</b> (Digital Waveform), <b>Файловый ввод/вывод осциллограмм</b> (Waveform File I/O), <b>Измерения параметров осциллограмм</b> (Waveform Measurements) и <b>Генерация осциллограмм</b> (Waveform Generation). Функции подпалитры.</p>
	<p>Подпалитра <b>Анализ</b> (Analyze) включает разделы <b>Операции с осциллограммами</b> и <b>Измерения параметров осциллограмм</b>, содержание которых идентично одноименным разделам функции <b>Осциллограмма</b>, а также разделы <b>Подготовка осциллограмм</b> (Waveform Conditioning), <b>Контроль осциллограмм</b> (Waveform Monitoring), подразделы <b>Поточечная обработка</b> (Point By Point), <b>Обработка сигналов</b> (Signal Processing) и <b>Математика</b> (Mathematics)</p>
	<p>Подпалитра <b>Связь с приборами</b> (Instrument I/O) включает разделы <b>Канал общего пользования</b> (GPIB или IEEE 488), <b>Последовательный интерфейс</b> (Serial) и библиотеку интерфейсов VISA (Virtual Instrument Software Architecture)</p>
	<p>Функции подпалитры <b>Управление приложением</b> (Application Control) позволяют программно управлять приложением (LabVIEW) или ВП на локальном или удаленном компьютере, изменять вид меню, останавливать ВП или выгружать LabVIEW.</p>



	Функции подпалитры <b>Графики и звук</b> (Graphics & Sound) позволяют строить трехмерные графики, производить запись/чтение графических файлов, формировать различные объекты на рисунке и осуществлять запись/чтение звуковых файлов.
	Подпалитра <b>Коммуникация</b> (Communication) содержит функции связи между приложениями Windows с использованием технологии .NET и ActiveX, а также функции сетевого обмена данными по протоколам TCP/IP, UDP, Data Socket, SMTP E-mail, IrDA и функцию System Exec.
	Функции подпалитры <b>Создание отчетов</b> (Report Generation) позволяют формировать отчеты в виде HTML-страниц
	Подпалитра <b>Расширенные</b> (Advanced) включает узел <b>Вызов библиотечной функции</b> (Call Library Function Node), узел <b>Интерфейсный код</b> (Code Interface Node (CIN)) и подразделы <b>Управление устройством ввода</b> (Input Device Control), <b>Манипуляция данными</b> (Data Manipulation), <b>Синхронизация</b> (Synchronization), <b>Ввод/вывод данных через порты</b> (Port I/O) и <b>Доступ к реестру Windows</b> (Windows Registry Access Vis).
	Подпалитра <b>Оформление</b> (Decorations) содержит такие графические элементы, как <b>Свободная метка</b> (Free Label), <b>Линия</b> (Line), <b>Ровный кадр</b> (Flat Frame)
	Подпалитра <b>Выбрать ВП</b> (Select a VI) позволяет выбрать ВП, не содержащийся в палитре функций, с помощью диалогового окна <b>Choose the VI to Open</b>
	Подпалитра <b>Библиотеки пользователя</b> (User Libraries) открывает доступ к ВП из библиотек пользователя, расположенных в папке LabVIEW 7.0\user.lib
	Подпалитра <b>Экспресс</b> (Express) открывает доступ к подпалитрам Экспресс-ВП <b>Вход</b> (Input), <b>Анализ сигнала</b> (Signal Analysis), <b>Выход</b> (Output), <b>Управление выполнением</b> (Execution Control), <b>Арифметика и сравнение</b> (Arithmetic Comparison) и <b>Обработка сигнала</b> (Signal Manipulation)

В состав подпалитры **Структуры** (Structures) входят следующие элементы:

	Структура <b>Стековая последовательности</b> (Stacked Sequence Structure) позволяет управлять последовательностью выполнения отдельных фрагментов кода путем их размещения в кадрах данной структуры
	Структура <b>Открытая последовательности</b> (Flat Sequence Structure). отличается от предыдущей возможностью передачи данных между кадрами без вспомогательной переменной и возможностью вывода данных из любого кадра структуры

	Структура <b>Вариант</b> (Case Structure) управляет выполнением одного из двух или более фрагментов кода и при выборе по условию аналогична оператору <b>if-then-else</b> текстовых языков, а при выборе по значению числовой или строковой переменной аналогична оператору <b>case</b>
	<b>Цикл с фиксированным числом итераций</b> (For Loop) осуществляет заданное число итераций при выполнении кода внутри данной структуры
	<b>Цикл по условию</b> (While Loop) осуществляет итерационное выполнение кода внутри данной структуры до выполнения заданного условия
	Узел <b>Формула</b> (Formula Node) позволяет включить фрагмент кода в текстовом представлении
	Структура <b>Событие</b> (Event Structure) ожидает наступления заданных событий на лицевой панели и производит их обработку
	<b>Глобальная переменная</b> (Global) используется для передачи данных между ВП на одном компьютере
	<b>Локальная переменная</b> (Local) используется для передачи данных между элементами управления или индикаторами без применения проводов
	Узел <b>Обратная связь</b> (Feedback Node) используется для передачи значений между итерациями структур циклов

#### 10.2.4. Таблица палитры числовых функций

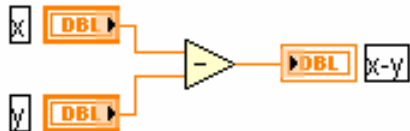
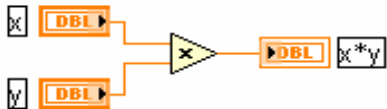
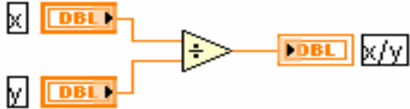
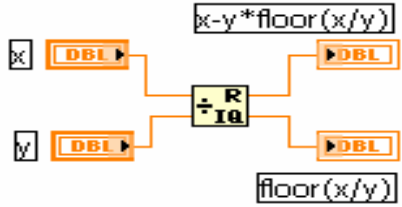
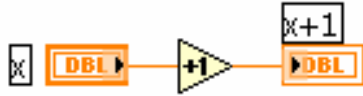
Вид блок-диаграммы ВП	Название функции
<p>а) </p>	<p><b>10.2.4.1. Сложить (Add).</b> Функция рассчитывает сумму входов. Если к входам функции подключаются две осциллограммы или два набора значений с динамическим типом данных, то рядом с функцией появляются терминалы <b>вход ошибки</b> (error in) и <b>выход ошибки</b> (error out). Не допускается суммирование двух значений меток времени. Функция является полиморфной, поэтому</p>
<p>б) </p>	
<p>в) </p>	
<p>г) </p>	
<p></p>	

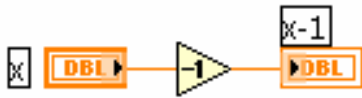

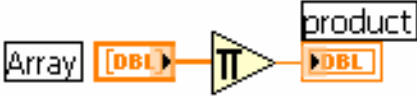
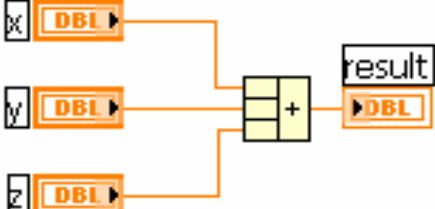

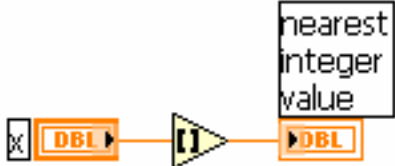

д)

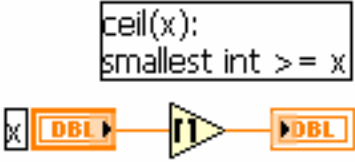

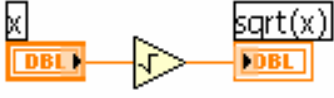
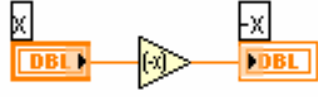
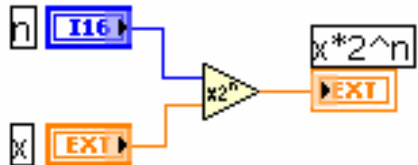
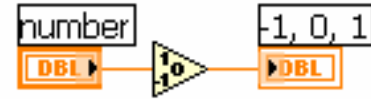

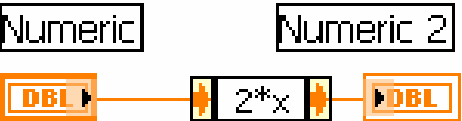
Рис.П.2.1





е)

входы могут быть числовыми скалярами, массивами или кластерами чисел, массивами кластеров чисел, отметками времени и т. д. На рисунках показаны варианты суммирования двух скаляров, скаляра и массива, двух массивов, отметки времени и скаляра, двух кластеров, двух осциллограмм (рис. П.1).

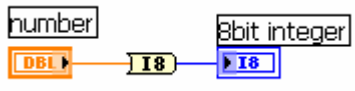
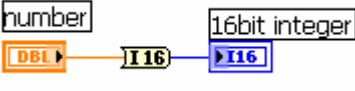
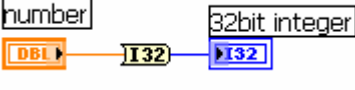
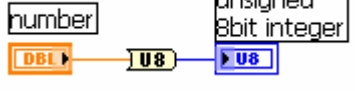
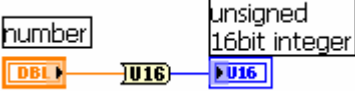
Вид блок-диаграммы ВП	Название функции
	<p><b>10.2.4.2. Вычесть (Subtract)</b>            Функция рассчитывает разность входов. Вычитание значений двух меток времени дает числовое значение (интервал времени), а вычитание числового значения из значения метки времени дает значение метки времени. Недопустимо вычитание метки времени из числового значения</p>
	<p><b>10.2.4.3. Умножить (Multiply)</b>            Функция возвращает произведение входов</p>
	<p><b>10.2.4.4. Разделить (Divide)</b>            Функция возвращает частное от деления значений на входах</p>
	<p><b>10.2.4.5. Частное и остаток (Quotient &amp; Remainder)</b>            Функция рассчитывает целое <b>частное</b> (<math>\text{floor}(x/y)</math>) и <b>остаток</b> (<math>x-y*\text{floor}(x/y)</math>) от деления <math>x</math> на <math>y</math></p>
	<p><b>10.2.4.6. Инкремент (Increment)</b>            Функция возвращает значение входа, увеличенное на 1</p>

	<p><b>10.2.4.7. Декремент (Decrement)</b>  Функция возвращает значение входа, уменьшенное на 1</p>
	<p><b>10.2.4.8. Сложить элементы массива (Add Array Elements)</b>  Функция возвращает сумму всех элементов входного <b>числового массива</b> (numeric array)</p>
	<p><b>10.2.4.9. Перемножить элементы массива (Multiply Array Elements)</b>  Функция возвращает произведение всех элементов входного <b>числового массива</b> (numeric array)</p>
	<p><b>10.2.4.10. Составная арифметика (Compound Arithmetic)</b>  Функция позволяет выполнять арифметические операции сложения, вычитания, умножения и деления с произвольным количеством числовых величин. Вид операции выбирается с помощью строки <b>Изменить режим (Change Mode)</b> контекстного меню функции.  Знак выхода и каждого входа может быть изменен путем выбора опции <b>Инвертировать (Invert)</b> контекстного меню</p>
	<p><b>10.2.4.11. Абсолютное значение (Absolute Value)</b> Функция возвращает абсолютное значение входа</p>
	<p><b>10.2.4.12. Округление до ближайшего целого (Round To Nearest)</b>  Функция округляет входное значение до ближайшего • целого. Если входное значение находится посередине между двумя целыми (например, 1,5 или 2,5), то функция возвращает ближайшее четное значение (2)</p>
 <p>floor(x): largest int ≤ x</p>	<p><b>10.2.4.13. Округление до меньшего целого (Round To -Infinity)</b>  Функция усекает входное значение до меньшего целого значения. Например, если входное значение равно 3,8, то результат будет равен 3. Если на входе -3,8, то результат будет равен -4</p>

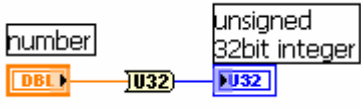
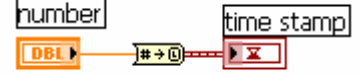
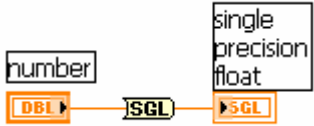
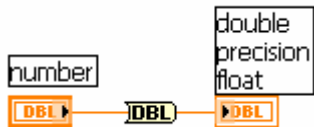
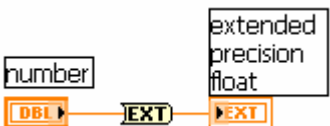
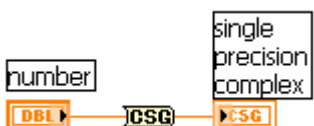



	<p><b>10.2.4.14. Округление до большего целого (Round To +infinity)</b>  Функция округляет входное значение до большего целого. Например, если входное значение равно 3,1, то результат будет равен 4. Если на входе -3,1, то результатом будет -3</p>
	<p><b>10.2.4.15. Случайное число в диапазоне (0-1) (Random Number (0-1))</b>  Функция генерирует случайные числа с равномерным амплитудным распределением в диапазоне от 0 до 1</p>
	<p><b>10.2.4.16. Квадратный корень (Square Root)</b> Функция рассчитывает квадратный корень входного значения. Если входное значение отрицательное, то возвращается значение NaN</p>
	<p><b>10.2.4.17. Отрицание (Negate)</b>  Функция изменяет знак входной величины на противоположный</p>
	<p><b>10.2.4.18. Масштабирование по степени числа 2 (Scale By Power Of 2)</b>  Функция умножает x на число 2, возведенное в степень p. Если p является числом с плавающей запятой, то функция округляет p перед масштабированием x (0,5 округляется до 0; 0,51 округляется до 1). Если x целое, то эта функция эквивалентна арифметическому сдвигу</p>
	<p><b>10.2.4.19. Знак (Sign)</b>  Функция возвращает значение -1, если входное число отрицательное, возвращает 0 - если оно равно 0, и возвращает 1, если число положительное</p>
	<p><b>10.2.4.20. Обратная величина (Reciprocal).</b> Функция делит 1 на входное значение</p>
	<p><b>10.2.4.21. Узел Выражение (Expression Node).</b> Узел <b>Выражение</b> используется для расчета выражений или уравнений, которые содержат единственную переменную. В формулах могут</p>

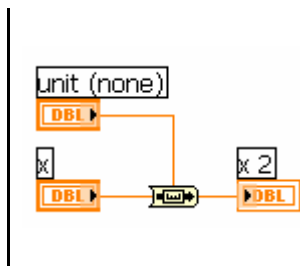
	использоваться следующие встроенные функции: abs, acos, acosh, asin, asinh, atari, atanh, ceil, cos, cosh, cot, esc, exp, expm 1, floor, getexp, getman, int, intrz, In, Inp1, log, Iog2, max, min, mod, rand, rem, sec, sign, sin, sine, sinh, sqrt, tan, tanh. Узел Выражение воспринимает только точку в качестве десятичного разделителя
	<b>10.2.4.22. Числовая константа</b> (Numeric Constant)
	<b>10.2.4.23. Константа перечисления</b> (Enum Constant)
	<b>10.2.4.24. Кольцевая константа</b> (Ring Constant)
	<b>10.2.4.25. Константа отметки времени</b> (Time Stamp Constant)

### 10.2.5. Функции преобразования в LabVIEW

Вид блок-диаграммы ВП	Название функции
	<b>10.2.5.1. В байтовое целое число (To Byte Integer).</b> Функция преобразует входное число в восьмибитовое целое в диапазоне от -128 до 127
	<b>10.2.5.2. В целое слово (To Word Integer).</b> Функция преобразует входное число в 16-битовое целое в диапазоне от -32,768 до 32,767
	<b>10.2.5.3. В длинное целое число (To Long Integer).</b> Функция преобразует входное число в 32-битовое целое в диапазоне от $-(2^{31})$ до $(2^{31})-1$
	<b>10.2.5.4. В байтовое целое без знака (To Unsigned Byte Integer).</b> Функция преобразует входное число в восьмибитовое целое число без знака в диапазоне от 0 до 255
	<b>10.2.5.5. В целое слово без знака (To Unsigned Word Integer).</b> Функция преобразует входное число в 16-битовое целое число без знака в диапазоне от 0 до 65 535



	<p><b>10.2.5.6. В длинное целое число без знака (To Unsigned Long Integer).</b> Функция преобразует входное число в 16-битовое целое число без знака в диапазоне от 0 до <math>(2^{32})-1</math></p>
	<p><b>10.2.5.7. В метку времени. (To Time Stamp).</b> Функция преобразует входное число в метку времени</p>
	<p><b>10.2.5.8. В число с плавающей запятой с одинарной точностью (To Single Precision Float).</b> Функция преобразует входное число в число с плавающей запятой с одинарной точностью</p>
	<p><b>10.2.5.9. В число с плавающей запятой с двойной точностью (To Double Precision Float).</b> Функция преобразует входное число в число с плавающей запятой с двойной точностью</p>
	<p><b>10.2.5.10. В число с плавающей запятой с расширенной точностью (To Extended Precision Float).</b> Функция преобразует входное число в число с плавающей запятой с расширенной точностью</p>
	<p><b>10.2.5.11. В комплексное число с одинарной точностью (To Single Precision Complex).</b> Функция преобразует входное число в комплексное число с плавающей запятой с одинарной точностью</p>
	<p><b>10.2.5.12. В комплексное число с двойной точностью (To Double Precision Complex)</b> Функция преобразует входное число в комплексное число с плавающей запятой с двойной точностью</p>
	<p><b>10.2.5.13. В комплексное число с расширенной точностью (To Extended Precision Complex).</b> Функция преобразует входное число в комплексное число с плавающей запятой с расширенной точностью</p>
	<p><b>10.2.5.14. Преобразовать размерность (Convert Unit).</b> Функция преобразует физическое число (размерное) в безразмерное число и наоборот</p>



**10.2.5.15. Изменить базовые единицы (Cast Unit Bases).** Функция изменяет базовые единицы, связанные с входом  $x$ , на базовые единицы, связанные с входом **размерность** (unit), и возвращает результаты на выходной терминал

**10.2.6. Таблица функций подпалитры Тригонометрические функции (Trigonometric):**

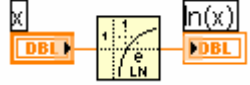
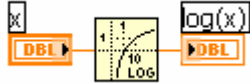
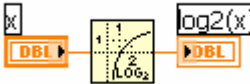
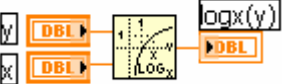

	<p><b>10.2.6.1. Синус (Sine).</b> Функция рассчитывает синус входного значения <math>x</math> (рад)</p>
	<p><b>10.2.6.2. Косинус (Cosine).</b> Функция рассчитывает косинус входного значения <math>x</math> (рад)</p>
	<p><b>10.2.6.3. Тангенс (Tangent).</b> Функция рассчитывает тангенс входного значения <math>x</math> (рад)</p>
	<p><b>10.2.6.4. Арксинус (Inverse Sine).</b> Функция рассчитывает значение арксинуса <math>x</math>. Результат выражен в радианах <math>x</math></p>
	<p><b>10.2.6.5. Арккосинус (Inverse Cosine).</b> Функция рассчитывает значение арккосинуса <math>x</math>. Результат выражен в радианах <math>x</math></p>
	<p><b>10.2.6.6. Арктангенс (Inverse Tangent)</b> Функция рассчитывает значение арктангенса <math>x</math>. Результат выражен в радианах <math>x</math></p>
	<p><b>10.2.6.7. Гиперболический синус (Hyperbolic Sine)</b> Функция рассчитывает гиперболический синус <math>x</math></p>
	<p><b>10.2.6.8. Гиперболический косинус (Hyperbolic Cosine).</b> Функция рассчитывает гиперболический косинус <math>x</math></p>
	<p><b>10.2.6.9. Гиперболический тангенс (Hyperbolic Tangent).</b> Функция рассчитывает гиперболический тангенс <math>x</math></p>
	<p><b>10.2.6.10. Гиперболический арксинус (Inverse Hyperbolic Sine).</b> Функция рассчитывает гиперболический арксинус <math>x</math></p>
	<p><b>10.2.6.11. Гиперболический арккосинус (Inverse Hyperbolic Cosine).</b> Функция рассчитывает гиперболический арккосинус <math>x</math></p>
	<p><b>10.2.6.12. Гиперболический арктангенс (Inverse Hyperbolic Tangent).</b> Функция рассчитывает</p>



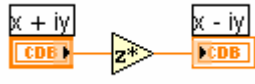
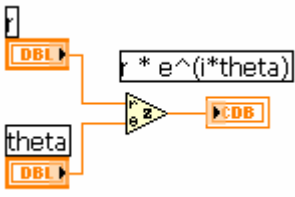
	ет гиперболический арктангенс $x$
	<b>10.2.6.13. Косеканс (Cosecant).</b> Функция рассчитывает косеканс <b>Входного</b> значения $x$ (рад)
	<b>10.2.6.14. Секанс (Secant).</b> Функция рассчитывает секанс <b>Входного</b> значения $x$ (рад)
	<b>10.2.6.15. Котангенс (Cotangent).</b> Функция рассчитывает котангенс <b>Входного</b> значения $x$ (рад)
	<b>10.2.6.16. Синус и косинус (Sine &amp; Cosine)</b> Функция рассчитывает синус и косинус <b>Входного</b> значения $x$ (рад)
	<b>10.2.6.17. Арктангенс (2 входа) (Inverse Tangent (2 Input)).</b> Функция рассчитывает арктангенс отношения $y/x$ , выраженного в радианах $x$ . Эта функция может рассчитывать арктангенс для углов в любом квадранте плоскости $x$ - $y$ , в то время как функция <b>Inverse Tangent</b> рассчитывает арктангенс только в двух квадрантах
	<b>10.2.6.18. Функция Sin(x)/x (Sine)</b> Функция рассчитывает значение $\sin(x)/x$ , где значение $x$ выражено в радианах

**10.2.7. Таблица функций подпалитры  
Логарифмические функции (Logarithmic):**

<b>Вид блок-диаграммы ВП</b>	<b>Название функции</b>
	<b>10.2.7.1. Экспонента (Exponential)</b> Функция рассчитывает значение числа $e$ , возведенного в степень $x$
	<b>10.2.7.2. Степень числа 10 (Power Of 10)</b> Функция рассчитывает значение числа 10, возведенного в степень $x$
	<b>10.2.7.3. Степень числа 2 (Power Of 2)</b> Функция рассчитывает значение числа 2, возведенного в степень $x$
	<b>10.2.7.4. Степень числа x (Power Of X)</b> Функция рассчитывает значение числа $x$ , возведенного в степень $y$
	<b>10.2.7.5. Функция Exp(x) – 1 (Exponential (Arg) -1).</b> Функция рассчитывает уменьшенное на 1 значение числа $e$ , возведенного в степень $x$ . При очень малых $x$ данная

	<p>функция является более точной по сравнению с функцией <b>Exponential</b>, у которой единица вычитается на выходе</p>
	<p><b>10.2.7.6. Натуральный логарифм (Natural Logarithm).</b> Функция рассчитывает натуральный логарифм числа <math>x</math></p>
	<p><b>10.2.7.7. Логарифм по основанию 10 (Logarithm Base 10)</b> Функция рассчитывает десятичный логарифм числа <math>x</math></p>
	<p><b>10.2.7.8. Логарифм по основанию 2 (Logarithm Base 2)</b> Функция рассчитывает логарифм числа <math>x</math> по основанию 2</p>
	<p><b>10.2.7.9. Логарифм по основанию x (Logarithm Base X)</b> Функция рассчитывает логарифм числа <math>y</math> по основанию <math>x</math></p>
	<p><b>10.2.7.10. Функция Ln(x+1) (Natural Logarithm (Arg +1))</b> Функция рассчитывает натуральный логарифм увеличенного на 1 значение числа <math>x</math>. При значениях <math>x</math>, близких к 0, данная функция является более точной по сравнению с добавлением 1 к <math>x</math> в функции <b>Натуральный логарифм</b></p>




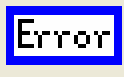
**10.2.8. Таблица функций подпалитры **Комплексные функции (Complex)**:**

Вид блок-диаграммы ВП	Название функции
	<p><b>10.2.8.1. Комплексно-сопряженное значение (Complex Conjugate).</b> Функция формирует комплексно-сопряженное значение <math>x-iy</math> для входного значения <math>x+iy</math>. <math>x + iy</math> может быть комплексным числом, массивом или кластером комплексных чисел, массивом кластеров комплексных чисел и т. д.</p>
	<p><b>10.2.8.2. Преобразование значений в полярных координатах в комплексное значение (Polar To Complex).</b> Функция формирует комплексное значение из двух значений, заданных в полярных координатах</p>













	<p><b>10.2.8.3. Преобразование комплексного значения в значения полярных координат (Complex To Polar).</b> Функция разделяет комплексное значение на два значения полярных координат</p>
	<p><b>10.2.8.4. Преобразование значений в декартовых координатах в комплексное значение (Re/Im To Complex).</b> Функция формирует комплексное значение из двух значений, заданных в декартовых координатах</p>
	<p><b>10.2.8.5. Преобразование комплексного значения в значения декартовых координат (Complex To Re/Im).</b> Функция разделяет комплексное значение на два значения декартовых координат</p>



















**10.2.9. Перечень констант подпалитры Дополнительные числовые константы (Additional Numeric Constants):**







Изображение	Описание	Изображение	Описание
	<b>Число <math>\pi</math></b> (3,1415926535897932)		<b>Число <math>2\pi</math></b> (6,28318530717958650)
	<b>Число <math>\pi/2</math></b> (1,5707963267948966)		<b>Число <math>1/\pi</math></b> (0,318309886183790670)
	<b>Натуральный логарифм числа <math>\pi</math></b> (1,1447298858494002)		<b>Основание натурального логарифма (число <math>e</math>)</b> (2,7182818284590452)
	<b>Значение <math>1/e</math></b> (0,36787944117144232)		<b>Десятичный логарифм числа <math>e</math></b> (0,43429448190325183)
	<b>Натуральный логарифм числа 10</b> (2,3025850929940597)		<b>Натуральный логарифм числа 2</b> (0,69314718055994531)
	<b>Отрицательная бесконечность</b>		<b>Положительная бесконечность</b>
	<b>Постоянная Планка</b> (6,62606876e-34) (Дж •с)		<b>Гравитационная постоянная</b> (6,673e-11)(Н-м <sup>2</sup> /кг <sup>2</sup> )
	<b>Скорость света</b> (2,99792458e8) м/с		<b>Число Авогадро</b> (6,02214199e23)(1/моль)
	<b>Элементарный заряд электрона</b> (1,602176462e-19) Кл		<b>Постоянная Ридберга</b> (1,0973731568549e7)1/м

	<b>Константа цвета</b> (Color Box Constant)		<b>Молярная газовая постоянная</b> (8,314472)Дж/(моль·К)
	<b>Кольцевая константа символов окна списков</b> (Listbox Symbol Ring Constant) используется для передачи символов в пункты элемента управления <b>Окно списков</b> (Listbox). Передача символа производится с помощью подключения константы к входу свойства <b>пункт символов</b> (Item Symbols) <b>узла свойств</b> (Property Node) элемента управления <b>Окно списков</b>		
	<b>Кольцевая константа ошибки</b> (Error Ring Constant) содержит список ошибок, возникающих при использовании памяти, работе в сети, печати и файловом вводе-выводе. Константа позволяет заменить код ошибки, формируемый функцией, на ее словесное описание		


### 10.2.10. Математические функции в LabVIEW








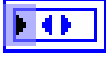














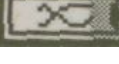
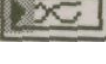

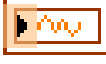




Функции узла	Имя аналогичной функции языка G	Числовая функция	Описание функции
abs(x)	Absolute Value		Абсолютное значение $x$
acos(x)	Inverse Cosine		Арккосинус $x$ , рад
acosh(x)	Inverse Hyperbolic Cosine		Гиперболический арккосинус $x$ , рад
asin(x)	Inverse Sine		Арксинус $x$ , рад
asinh(x)	Inverse Hyperbolic Sine		Гиперболический арксинус $x$ , рад
atan(x)	Inverse Tangent		Арктангенс $x$ , рад
atanh(x)	Inverse Hyperbolic Tangent		Гиперболический арктангенс $x$ , рад
ceil(x)	Round to +Infinity		Округление до большего целого
cos(x)	Cosine		Косинус $x$
cosh(x)	Hyperbolic Cosine		Гиперболический косинус $x$
cot(x)	Cotangent		Котангенс $x$ , рад
csc(x)	Cosecant		Косеканс $x$

exp(x)	Exponential		Экспонента $x$
expm1 (x)	Exponential (Arg) - 1		Экспонента $x-1$
floor(x)	Round to -Infinity		Округление до меньшего целого $x$
getexp(x)	Mantissa & Exponent (Data Manipulation)		Экспонента числа $x$ в случае его представления числом по основанию 2
getman(x)	Mantissa & Exponent (Data Manipulation)		Мантисса числа $x$ в случае его представления числом по основанию 2
int(x)	Round To Nearest integer		Округление до ближайшего целого $x$
intrz(x)	Round Toward 0		Округление до меньшего целого между $x$ и 0
ln(x)	Natural Log		Натуральный логарифм $x$
lnp1(x)	Natural Logarithm (Arg +1)		Натуральный логарифм $x+1$
log(x)	Logarithm Base 10		Десятичный логарифм $x$
log2(x)	Logarithm Base 2		Двоичный логарифм $x$
max(x,y)	Max& Min		Максимальное из $x,y$
min(x,y)	Max& Min		Минимальное из $x,y$
mod(x,y)	Quotient & Remainder		Остаток $x$ при округлении частного отделения $x/y$ до меньшего целого
rand	Random Number (0- 1)		Случайное число от 0 до 1
rem(x)	Remainder		Остаток $x$ при округлении частного отделения $x/y$ до ближайшего целого
pow(x,y)	Power Of X		Возведение $x$ в степень $y$
sec(x)	Secant		Секанс $x$





sign(x)	Sign		Знаковая функция: равна 1 при $x > 0$ , равна 0 при $x = 0$ , равна -1 при $x < 0$
sin(x)	Sine		Синус $x$
sinc(x)	Sinc		(Синус $x$ )/ $x$
sinh(x)	Hyperbolic Sine		Гиперболический синус $x$
sqrt(x)	Square Root		Квадратный корень $x$
tan(x)	Tangent		Тангенс $x$
tanh(x)	Hyperbolic Tangent		Гиперболический тангенс $x$

**10.4.1.** Перечень и вид терминалов элементов ввода данных LabVIEW 7.0 с характеристикой типа, цвета и значения по умолчанию приведены в Таблице

Элементы		Тип данных	Цвет	Значение по умолчанию
управления	индикации			
		Числовой с плавающей запятой одинарной точности	Оранжевый	0,0
		Числовой с плавающей запятой двойной точности	Оранжевый	0,0
		Числовой с плавающей запятой расширенной точности	Оранжевый	0,0
		Комплексный с плавающей запятой одинарной точности	Оранжевый	0,0+i0,0
		Комплексный с плавающей запятой двойной точности	Оранжевый	0,0+i0,0
		Комплексный с плавающей запятой расширенной точности	Оранжевый	0,0+i0,0
		Числовой байтовый со знаком	Синий	0
		Числовой двухбайтовый со знаком	Синий	0
		Числовой четырехбайтовый со знаком	Синий	0

		Числовой байтовый без знака	Синий	0
		Числовой двухбайтовый без знака	Синий	0
		Числовой четырехбайтовый без знака	Синий	0
		Нумерованный	Синий	-
		<64,64>-битовая отметка времени		Дата и время (локальное)
		Логический	Зеленый	FALSE
		Строковый	Розовый	
		Массив - включает тип данных в квадратные скобки и принимает цвет этого типа данных	Различный	
		Кластер - включает разные типы данных. Кластерный тип данных имеет коричневый цвет если все элементы кластера числовые, и розовый, если элементы разных типов	Коричневый или розовый	-
				
		Путь	Морской волны	<Не путь>
		Динамический (Dynamic) - (Express Vis) - включает данные, связанные с сигналом и атрибутами, которые обеспечивают информацию о сигнал такую как имя сигнала или дат; и время получения данных	Фиолетовый	
		Осциллограмма - кластер элементов, которые содержат дату, начальное время и интервал выборок сигнала	Коричневый	-
		Ссылка	Морской волны	-
		Вариант - включает имя элемента управления или	Фиолетовый	-



		индикатора, информацию о типе данных и сами данные		
		I/O name - передает имя канала платы сбора данных (DAQ), ресурсное имя VISA или логическое имя IVI с целью их конфигурации	Фиолетовый	-
		Рисунок - отображает рисунок, который содержат линии, окружности, текст и другие графические элементы	Синий	-

#### 10.4.2. Пояснения к пунктам контекстного меню

<b>Видимые объекты (Visible Items)</b>	Позволяет показать или скрыть определенные элементы оформления, такие как ярлыки, заголовки, полосы прокрутки или соединительные терминалы
<b>Найти терминал (Find Terminal)</b>	LabVIEW находит и выделяет на блок-диаграмме терминал элемента лицевой панели
<b>Найти элемент управления (Find Control)</b>	LabVIEW находит и выделяет на лицевой панели элемент управления, имеющий заданный терминал на блок-диаграмме
<b>Заменить на индикатор (Change to Indicator)</b>	Позволяет заменить элемент управления лицевой панели на индикатор. Для индикатора аналогичный пункт выполняет обратное действие
<b>Сделать элемент управления невидимым (Hide Control)</b>	Позволяет сделать элемент управления невидимым. Для невидимого элемента аналогичный пункт меню выполняет обратное действие
<b>Описание и подсказка (Description and Tip)</b>	Позволяет описать объект и сделать подсказку к нему
<b>Создать (Create)</b>	Позволяет создать локальную переменную, узел свойств или ссылку
<b>Заменить (Replace)</b>	Позволяет входить в палитры панелей и заменять выбранный объект другим
<b>Операции с данными (Data Operations)</b>	Позволяет устанавливать текущие значения как значения по умолчанию, удалять, копировать и вставлять данные, устанавливать соединение DataSocket
<b>Дополнительно (Advanced)</b>	Позволяет производить тонкую настройку элементов управления и индикаторов
<b>Свойства (Properties)</b>	Позволяет выбрать диалоговое окно свойств объекта







Следующие пункты данного контекстного меню являются специфическими для числовых элементов:



<b>Представление (Representation)</b>	Позволяет выбирать и устанавливать тип представления числовых данных
<b>Диапазон данных (Data Range)</b>	Позволяет настраивать определенный действующий диапазон числовых значений и приращений данных
<b>Формат и точность (Format &amp; Precision)</b>	Позволяет выбирать формат и точность представления данных

Представление данных в LabVIEW включает следующие типы:

 - повышенной точности	 - двойной точности	 - 32-битовый
 - целочисленный 32-битовый со знаком	 - целочисленный 16-битовый со знаком	 - целочисленный 8-битовый со знаком
 - целочисленный 32-битовый без знака	 - целочисленный 16-битовый без знака	 - целочисленный 8-битовый без знака
 - комплексный повышенной точности	 - комплексный двойной точности	 - комплексный 32-битовый

#### 10.4.3. Варианты срабатывания режимов.

Изображение	Режим	Действие
	<b>Включить при нажатии (Switch When Pressed)</b>	Значение изменяется при каждом щелчке по элементу инструментом управления
	<b>Включить при отпускании (Switch When Released)</b>	Значение изменяется после отпускания кнопки мыши при каждом щелчке по элементу инструментом управления
	<b>Включить до отпускания (Switch Until Released)</b>	Значение изменяется при щелчке и автоматически возвращается в исходное состояние после отпускания кнопки мыши
	<b>Сработать при нажатии (Latch When Pressed)</b>	Значение изменяется при щелчке и автоматически возвращается в исходное состояние после однократного считывания виртуальным прибором нового

		значения
	<b>Сработать при отпускании</b> (Latch When Released)	Значение изменяется после отпускания кнопки мыши и автоматически возвращается в исходное состояние после однократного считывания виртуальным прибором нового значения
	<b>Сработать до отпускания</b>	Значение изменяется при щелчке и возвращается в исходное состояние после однократного считывания виртуальным прибором нового значения или после отпускания кнопки мыши

В меню настроек строковых элементов управления и индикаторов специфическим является раздел, определяющий вид отображения содержимого этих элементов.

**10.4.4.** Интерпретация строк данного раздела меню приведена в таблице.

<b>Нормальное отображение</b> (Normal Display)	Позволяет отображать содержимое в том виде, в котором оно вводится с клавиатуры	
<b>"\" Кодированное отображение</b> (Codes Display)	Позволяет принимать и отображать символы, которые являются неотображаемыми. Скрытые символы появляются в виде обратного слэша, за которым следует соответствующий код. Ниже в таблице приведены слэш-коды LabVIEW.	
	<b>Код</b>	<b>Выполнение в LabVIEW</b>
	\OO-\\FF	Шестнадцатеричное значение восьмибитового символа, буквы должны быть заглавными.
	\\b	Знак возврата на один символ назад (ASCII BS, эквивалент \\08)
	\\f	Знак смещения формы (ASCII FF, эквивалент \\0C)
	\\n	Новая строка (ASCII LF, эквивалент \\0A)
	\\r	Знак возврата каретки (ASCII CR, эквивалент 0D)
	\\t	Табуляция (ASCII HT, эквивалент \\09)
	\\s	Пробел (эквивалент \\20)
\\\\	Обратный слэш (ASCII \\, эквивалент \\5S)	
<b>Скрытое отображение</b> (Password Display)	Переводит элемент управления или отображения в режим показа «звездочки» вместо каждого введенного символа	

<b>Шестнадцатеричное отображение</b> (Hex Display)	Преобразует строку алфавитно-цифровых знаков в шестнадцатеричные символы
<b>Ограничить до одной строки</b> (Limit to Single Line)	Выбор функции приводит к блокированию ввода символа «возврат каретки», и строка всегда остается единственной. Ввод символа «возврат каретки» приводит к автоматическому завершению ввода текста
<b>Обновлять значение во время ввода</b> (Update Value While Typing)	Выбор функции приводит к обновлению данных во время набора текста. Если эта строка не выбрана, то вводимый текст передается от элемента управления строками к его терминалу на блок-диаграмме только после щелчка мышью за пределами окна строки или по кнопке <input checked="" type="checkbox"/>

### 10.5.1. Список операторов узла **Формула** (по приоритету):

Символ	Значение
**	Возведение в степень
+, -, !,  , ++, -	Унарный плюс, унарный минус, логическое НЕ, дополнение бита, пред- и постинкремент, пред- и постдекремент
*, /, %	Умножение, деление, модуль (остаток)
+ и -	Сложение и вычитание
>> и <<	Арифметический сдвиг вправо и сдвиг влево
>, <, >=, <=	Больше, меньше, больше или равно, меньше или равно
!= и ==	Неэквивалентность и эквивалентность
&	Битовое И
^	Битовое исключающее ИЛИ
	Битовое ИЛИ
&&	Логическое И
	Логическое ИЛИ
?:	Условное выражение
= оп =	Присваивание Укороченная операция и присваивание оп (оператора) может быть +, -, *, /, », «, &, М, % **

