

**ПРИОРИТЕТНЫЙ НАЦИОНАЛЬНЫЙ ПРОЕКТ «ОБРАЗОВАНИЕ»
РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**А.М. УМНОВ, В.А. ТУРИКОВ,
М.Н. МУРАТОВ, В.П. МИЛАНТЬЕВ**

**СОВРЕМЕННЫЕ
ЧИСЛЕННО-АНАЛИТИЧЕСКИЕ ПАКЕТЫ
ДЛЯ СЛОЖНЫХ ИНЖЕНЕРНО-ФИЗИЧЕСКИХ
ВЫЧИСЛЕНИЙ**

Учебное пособие

Москва

2008

**«Создание комплекса инновационных образовательных программ
и формирование инновационной образовательной среды,
позволяющих эффективно реализовывать государственные интересы РФ
через систему экспорта образовательных услуг»**

Экспертное заключение –

доктор физико-математических наук, профессор *Е.Ю. Щетинин*

Умнов А.М., Туриков В.А., Муратов М.Н., Милантьев В.П.

Современные численно-аналитические пакеты для сложных инженерно-физических вычислений: Учеб. пособие. – М.: РУДН, 2008. – 154 с.

Пособие предназначено для студентов первого года обучения в магистратуре. Освоение данного курса позволит им быстро перейти от начального знакомства с программами компьютерной алгебры к некоему минимальному (стартовому) уровню, позволяющему проводить профессиональные вычисления и моделирование сложных физических и технических систем. В пособии представлены три основных профессиональных численно-аналитических пакета – Maple, MATLAB и Mathematica. Пособие не является систематическим учебником по этим пакетам, поэтому детальное изложение языков выходит за его рамки. После необходимого минимума начальных сведений о пакетах дальнейшее обучение происходит на конкретных примерах расчетов и моделирования, причем в некоторых из них затрагиваются вопросы, находящиеся на переднем крае современных исследований.

Учебное пособие выполнено в рамках инновационной образовательной программы Российского университета дружбы народов, направление «Комплекс экспортноориентированных инновационных образовательных программ по приоритетным направлениям науки и технологий», и входит в состав учебно-методического комплекса, включающего описание курса, программу и электронный учебник.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
ТЕМА 1. ОСНОВЫ АНАЛИТИЧЕСКОГО ПРОГРАММИРОВАНИЯ И ОБРАБОТКИ ДАННЫХ. СИСТЕМЫ КОМПЬЮТЕРНОЙ АЛГЕБРЫ. ПАКЕТ MAPLE	8
1.1. Историческая справка	8
1.2. Основные сведения о Maple и начальные навыки работы	9
1.3. Пример расчета физической системы в Maple	24
ТЕМА 2. ПАКЕТ MATLAB – НАЧАЛЬНЫЕ СВЕДЕНИЯ	45
2.1. Введение	45
2.2. Начальные навыки работы с MATLAB	48
2.3. Решение задачи о туннелировании в MATLAB	57
ТЕМА 3. ПАКЕТ MATHEMATICA. СРАВНИТЕЛЬНЫЙ АНАЛИЗ ТРЕХ ЧИСЛЕННО-АНАЛИТИЧЕСКИХ ПАКЕТОВ MAPLE, MATLAB И MATHEMATICA	62
3.1. Историческая справка и общие сведения о Mathematica	62
3.2. Начальное знакомство с системой Mathematica	63
3.3. Решение задачи о туннелировании	69
3.4. Сопряжения систем компьютерной алгебры	73
ТЕМА 4. ПРИМЕРЫ ВЫЧИСЛЕНИЙ И МОДЕЛИРОВАНИЯ СИСТЕМ С ПОМОЩЬЮ ЧИСЛЕННО-АНАЛИТИЧЕСКИХ ПАКЕТОВ	77
4.1. Имитационное моделирование методом Монте-Карло модельной системы броуновского движения: случайные блуждания на прямой	77
4.2. Случайные блуждания на прямой: упрощенный случай	92
4.3. Вычисление статистической суммы модели Изинга и сравнение с известными точными выражениями	94

4.4. Регулярное и стохастическое движение в асимметричном вихре	110
4.4.1. Введение	110
4.4.2. Постановка задачи	112
4.4.3. Регулярное движение	112
4.4.4. Тепловое движение	113
4.4.5. Геометрия системы и движение частиц	114
4.4.6. Программная реализация	115
ТЕМА 5. ПРИМЕР МОДЕЛИРОВАНИЯ В МАТЛАВ ТЕХНИЧЕСКОЙ СИСТЕМЫ: МОДЕЛЬ АЛГОРИТМА ОЦЕНИВАНИЯ ОШИБОК ИНС И GPS НА ОСНОВЕ ФИЛЬТРА КАЛЬМАНА	120
5.1. Постановка проблемы и обзор математических средств, пригодных для решения поставленной задачи	119
5.2. Алгоритмы адаптивной фильтрации	132
5.3. Оптимальный фильтр Винера	123
5.4. Алгоритм LMS	125
5.5. Детерминированная задача оптимизации	128
5.6. Алгоритм RLS	129
5.7. Алгоритм Кальмана	131
5.8. Описание алгоритма и алгоритмов выбора начальных значений	133
5.9. Описание входных и выходных параметров	136
5.10. Комментированный текст программы в MATLAB	139
5.11. Описание параметров работоспособности фильтра и примеры применения	140
ЗАКЛЮЧЕНИЕ	145
ОПИСАНИЕ КУРСА И ПРОГРАММА	153

Введение

«При изучении наук примеры полезнее правил» – это знаменитое утверждение И. Ньютона как нельзя лучше подходит для изучения численно-аналитических пакетов и вообще программирования. Посмотрев на библиографию к данному курсу, можно обратить внимание на обилие учебников и пособий, посвященных описанию этих программ. Ощущается явная избыточность количества таких пособий и это притом, что библиография явно неполная и включает в себя в большинстве русскоязычные наименования! Учитывая также, что в каждом пакете имеются превосходно составленные встроенные справочные пособия (Help), становится непонятным наличие такого количества во многом дублирующих друг друга учебников, фактически представляющих собой расширенные толкования Help. Имеется некоторое количество учебников, посвященных именно применению численно-аналитических пакетов, и данный курс относится к этой категории. Авторы никогда не изучали компьютерные программы по учебникам, а шли от практических задач, используя пакеты программ как инструменты, осваивая их по мере необходимости с помощью Help. Этим своим опытом мы и хотели поделиться, составив данное пособие.

Помимо основополагающего принципа И. Ньютона, приведенного в начале, в данном учебнике используется еще один методический прием, авторство которого принадлежит нам и проверено нами на практике. Он состоит в том, что при изучении наук чрезвычайно полезно решать одни и те же задачи разными способами. Нет ничего лучше для проясненного понимания материала и закрепления связанных с ним навыков. В данном случае речь идет о решении одной и той же задачи с помощью разных программных пакетов.

Кроме того, в построении данного пособия использован широко известный принцип «от простого к сложному», т.е., начав с элементарного знакомства с численно-аналитическими пакетами, переходим к простей-

шим примерам, далее решаем одну и ту же нетривиальную задачу из квантовой механики с помощью трех программ. Причем задача разбирается досконально, в малейших деталях, как в плане физики, так и в плане программного воплощения. Далее также весьма подробно разобрано несколько примеров (в одном из пакетов – Maple; реализация в других пакетах будет предложена как одна из лабораторных работ). Один из этих примеров – модель Изинга – представляет собой нерешенную проблему современной теоретической физики, т.е. здесь читатель соприкоснется с проблемами, находящимися на острие научных исследований.

Следует отметить, что принцип «от простого к сложному» в применении к обучению в современных условиях, нуждается в следующей корректировке: на определенном этапе следует резко ускорить это восхождение к сложному, предложив обучаемому разобраться сразу в достаточно сложной задаче (пока только разобраться в готовом решении, а не решать). Здесь напрашивается аналогия с одним из методов обучения плаванию, когда не умеющего плавать бросают на глубокую воду и предоставляют выплыть самому.

Именно нечто подобное мы предлагаем читателю в пятой теме. Там досконально разобрана очень актуальная и близкая к реальности (хотя версия имитационная) техническая задача, воплощенная в программу в MATLAB. Читателю необходимо разобраться в самой задаче, в алгоритме и тексте программы и воспроизвести результаты. В лабораторном практикуме ему будет предложено развить эту программу.

Наконец, хотелось бы еще раз напомнить, что при освоении вычислений и моделирования с помощью программных пакетов идти надо не от них, а от существа решаемых задач. Заблуждением является мнение, что использование пакетов может в чем-то компенсировать ваше недостаточное понимание материала. Как раз наоборот: мощные вычислительные возможности программ в ряде случаев приоткроют перед вами такие ас-

пекты задачи, понять и использовать которые вы сможете, только если являетесь хорошим физиком или инженером и глубоко владеете предметом.

В Теме 1 кратко изложены основные сведения о численно-аналитических пакетах вообще и о Maple в частности. Описано все необходимое для начала работы с этой программой, даны элементарные примеры математических и физических расчетов. Рассмотрена нетривиальная модельная задача квантовой механики – вычисление коэффициента прохождения через одномерный прямоугольный потенциальный барьер, которая тщательно разобрана во всех деталях. В Теме 2 алгоритм начального знакомства с программой, реализованный в предыдущей теме для Maple, применен к пакету MATLAB, а в Теме 3 – к Mathematica. Кроме того, в Теме 3 по ходу изложения внимание уделяется практическому сравнению различных пакетов компьютерной алгебры, а в конце темы даются некоторые общие сведения о возможностях их сопряжения между собой. Тема 4 посвящена конкретным примерам расчета и моделирования физических систем: модель броуновского движения, модель одномерного случайного блуждания и модель Изинга, регулярное и стохастическое движение в асимметричном вихре. В Теме 5 рассмотрена актуальная техническая задача фильтрации и сопряжения сигналов навигационных систем глобального позиционирования ИНС и ГЛОНАСС. После изложения теоретических сведений об адаптивных фильтрах приводится описание алгоритма имитационного моделирования системы фильтрации и сопряжения, приводится текст программы, примеры результатов моделирования и их обсуждение.

Тема 1. Основы аналитического программирования и обработки данных. Системы компьютерной алгебры. Пакет Maple

1.1. Историческая справка

Время начала создания компьютерной символьной математики [1] относится к началу 60-х годов. Именно тогда в вычислительной технике возникла новая ветвь компьютерной математики, не совсем точно 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. Далее 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, 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: Maple, Derive, MathCAD, Mathematica и др. [3, с. 4].

Наибольшую известность получили три класса систем символьной математики: созданная на базе языка искусственного интеллекта Lisp малая система Derive, одна из самых мощных и поныне привлекательных систем Maple (ядро написано на языке C) и системы Mathematica [3, с. 5]. Позже на базе ядра системы Maple символьные вычисления были реализо-

ваны в популярных числовых системах Mathcad – версии Mathcad имеют продвинутый пользовательский интерфейс и возможности, улучшающиеся от версии к версии. Блок символьной математики на базе ядра Maple был добавлен и в одну из самых крупных систем – MATLAB.

1.2. Основные сведения о Maple и начальные навыки работы

Maple – система компьютерной математики, рассчитанная на широкий круг пользователей. До недавнего времени ее называли системой компьютерной алгебры, что указывало на особую роль символьных вычислений и преобразований, которые способна осуществлять эта система. Но такое название сужает сферу применения системы. На самом деле она способна выполнять быстро и эффективно не только символьные, но и численные расчеты, причем сочетает это с превосходными средствами графической визуализации и подготовки электронных документов.

Maple – типичная интегрированная система. Она объединяет в себе:

- мощный язык программирования (он же язык для интерактивного общения с системой);
- редактор для подготовки и редактирования документов и программ;
- современный многооконный пользовательский интерфейс с возможностью работы в диалоговом режиме;
- мощную справочную систему со многими тысячами примеров;
- ядро алгоритмов и правил преобразования математических выражений;
- численный и символьный процессоры;
- систему диагностики;
- библиотеки встроенных и дополнительных функций;
- пакеты функций сторонних производителей и поддержку некоторых других языков программирования и программ.

Следует отметить, что последние версии Maple имеют новый интерфейс, однако он далеко не всегда оправдывает себя в плане соотношения используемые ресурсы – качество. Видимо, по этой причине версия Maple 10 предоставляет возможность запуска программы в режиме так называемого «классического интерфейса». Поскольку практически все функции и ресурсы не зависят от интерфейса и доступны в «классическом» режиме, в данном пособии рассматривается только последний вариант. Практика показывает, что реальные вычисления происходят в нем быстрее, что связано с меньшей загрузкой оперативной памяти и процессора вспомогательными функциями.

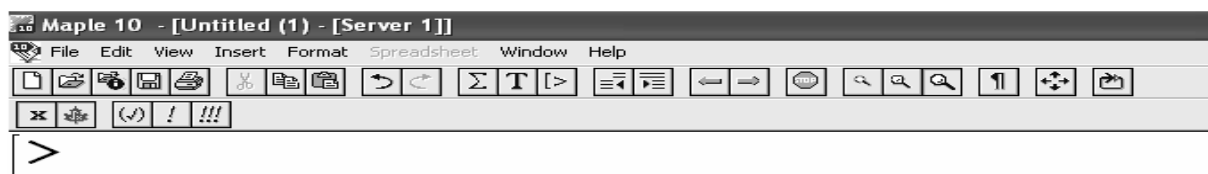


Рис. 1.1. «Классический» интерфейс Maple 10

Как у всех приложений под Windows, интерфейс Maple имеет ряд характерных элементов, видимых на рисунке 1.1 и перечисленных ниже:

- строка заголовка (сверху);
- строка главного меню;
- главная панель инструментов;
- контекстная панель инструментов, вид которой зависит от режима работы с Maple;
- окно ввода и редактирования документов; строка состояния (в самом низу окна).

Пользовательский интерфейс Maple позволяет готовить документы, содержащие одновременно текстовые комментарии, команды входного языка (с возможным преобразованием их в естественную математическую

форму), результаты вычислений в виде обычных математических формул и графические данные. Это обеспечивает понятное представление исходных данных и результатов вычислений, а также удобство их повторного использования. В основе пользовательского интерфейса Maple лежит графический многооконный интерфейс операционной системы Windows. Управление системой Maple возможно с помощью главного меню, панелей инструментов и палитр, а также «горячих» клавиш. Поддерживаются также многие возможности мыши, присущие приложениям под Windows. Важно отметить и прекрасно реализованную справочную систему Maple. Преодолев первые трудности общения с системой, пользователь быстро осваивает систему справки, которая позволяет без какой-либо бумажной документации получить исчерпывающую информацию о любом операторе, функции или пакете (на английском языке). На каждой странице справки находятся по несколько примеров, причем их можно скопировать и перенести в редактор или в окно исполняемых документов системы. Это способствует быстрому обучению пользователя. Пользователь Maple (как и ряда других математических систем) работает с документами, которые являются одновременно описаниями алгоритмов решения задач, программами и результатами их исполнения. Все данные команды и результаты размещаются в соответствующих ячейках. Графические построения выполняются как в ячейках документа, так и в отдельных окнах, и имеют свои меню для оперативного управления параметрами.

Основой для работы с символьными преобразованиями в Maple является ядро системы. Оно содержит сотни базовых функций и алгоритмов символьных преобразований. В новых реализациях объем ядра достигает 6–7 Мбайт. Имеется также основная библиотека операторов, команд и функций. Многие встроенные в нее функции, как и функции ядра, могут использоваться без какого-либо объявления, другие нуждаются в объявлении. Кроме того, имеется ряд подключаемых пакетов (packages).

Дополнительные функции из пакетов могут применяться после объявления подключения пакета с помощью команды `with(name)`, где `name` – имя применяемого пакета. Общее число функций, с учетом встроенных в ядро и размещенных в пакетах, в системе приближается к 3000. Это означает, что большинство задач может решаться в режиме прямого диалога с системой без использования каких-либо программных средств.

Например, пакет `linalg` – линейная алгебра – представляет собой совокупность следующих основных функций:

```
> with(linalg);
```

```
Warning, the protected names norm and trace have been redefined and unprotected
```

```
[BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian,
addcol, addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout,
blockmatrix, charmat, charpoly, cholesky, col, coldim, colspace, colspan,
companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det,
diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects,
entermatrix, equal, exponential, extend, ffgausselim, fibonacci, forwardsub,
frobenius, gausselim, gaussjord, geneqns, genmatrix, grad, hadamard, hermite,
hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse,
ismith, issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve,
matadd, matrix, minor, minpoly, mulcol, mulrow, multiply, norm, normalize,
nullspace, orthog, permanent, pivot, potential, randmatrix, randvector, rank,
ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith,
stackmatrix, submatrix, subvector, sumbasis, swapcol, swaprow, sylvester, toeplitz,
trace, transpose, vandermonde, vecpotent, vectdim, vector, wronskian ]
```

Maple способна решить огромное число задач вообще без какого-либо программирования в общепринятом смысле этого понятия. Достаточно лишь описать алгоритм решения задачи и разбить его на отдельные вопросы, на которые система Maple способна дать ответы. Более того, есть тысячи задач, алгоритмы решения которых уже реализованы в виде функций и команд системы. Тем не менее, это вовсе не означает, что в Maple нельзя программировать. На самом деле Maple поддерживает три собственных языка: входной, реализации и программирования.

Maple имеет входной язык сверхвысокого уровня, ориентированный на решение математических задач практически любой сложности. Он служит для задания системе вопросов или, говоря иначе, задания входных данных для последующей их обработки. Это язык интерпретирующего типа и по своей идеологии напоминает Бейсик. Входной язык имеет большое число заранее определенных математических и графических функций, а также обширную библиотеку, подключаемую по мере необходимости.

Имеет Maple и свой язык процедурного программирования – Maple-язык. Этот язык имеет вполне традиционные средства структурирования программ: операторы циклов, операторы условных и безусловных переходов, операторы сравнения, логические операторы, команды управления внешними устройствами, функции пользователя, процедуры и т.д. Он также включает в себя все команды и функции входного языка, ему доступны все специальные операторы и функции. Многие из них являются весьма серьезными программами, например символьное дифференцирование, интегрирование, разложение в ряд Тейлора, построение сложных трехмерных графиков и т.д.

Не следует путать входной язык и язык программирования системы (Maple-язык) с языком ее реализации. Им является один из самых лучших и мощных универсальных языков программирования – Си. На нем написано ядро системы, содержащее тщательно оптимизированные процедуры. Большинство же функций, которые содержатся в пакетах, написаны на Maple-языке, благодаря чему их можно модифицировать и даже писать свои собственные библиотеки. По разным оценкам, лишь от 5 до 10% средств Maple создано на языке реализации – все остальное написано на Maple-языке. Таким образом, система имеет развитые возможности для расширения и адаптации к задачам пользователя. Для подготовки программ на языке Maple могут использоваться внешние редакторы, но система имеет и свой встроенный редактор, вполне удовлетворяющий требова-

ниям большинства пользователей. Он открывается командами New и Open в меню File. Этот редактор можно использовать для редактирования файлов программ или математических выражений.

Синтаксис структурных операторов языка Maple напоминает смесь Бейсика и Паскаля. Это облегчает знакомство с ним тем, кто имеет хотя бы начальный опыт программирования на этих языках. По близким к Бейсику правилам (и при помощи общепринятых математических сокращений) выполняется и ввод математических выражений в диалоговом режиме работы с системой.

Перечислим кратко основные возможности Maple.

Интерфейс:

- работа со многими окнами;
- вывод графиков в отдельных окнах или в окне документа;
- представление выходных и входных данных в виде естественных математических формул;
- задание текстовых комментариев различными шрифтами;
- возможность использования гиперссылок и подготовки электронных документов;
- удобное управление с помощью клавиатуры через главное меню и инструментальную панель;
- управление с помощью мыши.

Символьные и численные вычисления:

- дифференцирование функций;
- численное и аналитическое интегрирование;
- вычисление пределов функций;
- разложение функций в ряды;
- вычисление сумм и произведений;
- интегральные преобразования Лапласа, Фурье и др.;
- дискретные Z-преобразования;

- прямое и обратное быстрое преобразование Фурье;
- работа с кусочно-заданными функциями.

Работа с уравнениями в численном и символьном виде:

- решение систем линейных и нелинейных уравнений;
- решение систем дифференциальных уравнений;
- символьное вычисление рядов;
- работа с рекуррентными функциями;
- решение трансцендентных уравнений;
- решение систем с неравенствами.

Работа с функциями:

- вычисление значений всех элементарных функций;
- вычисление значений большинства специальных математических функций;
- пересчет координат точек между различными координатными системами;
- задание функций пользователя.

Линейная алгебра:

- свыше ста операций с векторами и матрицами;
- решение систем линейных уравнений;
- формирование специальных матриц и их преобразования;
- вычисление собственных значений и собственных векторов матриц;
- поддержка быстрых векторных и матричных алгоритмов пакета программ NAG.

Графическая визуализация результатов вычислений:

- построение графиков многих функций;
- различные типы осей (с линейным и логарифмическим масштабом);
- графики функций в декартовой и полярной системах координат;

- специальные виды графиков (точки массивов, векторные графики, диаграммы уровней и др.);
- системы координат, определяемые пользователем;
- графики, представляющие решения дифференциальных уравнений;
- графики трехмерных поверхностей с функциональной закраской;
- построение пересекающихся в пространстве объектов;
- задание пользователем окраски графиков;
- импорт графиков из других пакетов и программных систем;
- анимация графиков;
- создание и проигрывание анимационных файлов.

Программирование:

- встроенный язык процедурного программирования;
- простой и типичный синтаксис языка программирования;
- обширный набор типов данных;
- типы данных, задаваемых пользователем;
- средства отладки программ;
- мощные библиотеки функций;
- задание внешних функций и процедур;
- поддержка языков программирования С и Fortran;
- возможность записи формул в формате LaTeX.

Отметим также некоторые специальные возможности:

- новый пакет для поддержки языка XML;
- поддержка новейшего стандарта записи математической информации – языка MathML 2.0;
- поддержка протокола TCP/IP, обеспечивающего динамический удаленный доступ к данным, например, для финансового анализа в реальном масштабе времени или данных о погоде;
- дополнительные пакеты (Maple PowerTools), доступные через Интернет, поддерживающие анализ методом конечных элементов

(РЕМ), нелинейную оптимизацию и статистику, а также три новых пакета: вычисления для новичков, теоретическая физика и программирование;

- возможность работы с курсом университетского математического образования, загружаемого через Интернет.

Для получения начальных навыков работы с системой рассмотрим конкретные примеры. После загрузки и запуска системы можно начать диалог с ней с использованием операторов и функций (с параметрами) для создания и вычисления математических выражений. Во избежание грубых ошибок рекомендуется вначале любой программы помещать команду `restart`, которая снимает определения со всех использованных ранее переменных и позволяет начать вычисления «с чистого листа». Ниже представлен реальный диалог с системой при решении простейших арифметических задач и построении графика функций $\sin(ax)/x$, $\cos(ax)$. Уже из этого простого примера видны особенности диалога с Maple и синтаксиса ее входного языка, то есть языка, на котором системе задаются вопросы. Диалог идет в стиле: «задал вопрос, получил ответ». Знак `>` является знаком приглашения к заданию вопроса. Мигающая вертикальная черта `|` – маркер ввода (курсор).

Ввод выражений (вопросов) задается по правилам, давно принятым для строчных редакторов. Они хорошо известны, и мы не будем на них останавливаться подробно. Отметим лишь специфический момент: знак фиксации конца выражения `;` (точка с запятой) указывает, что результат его вычисления должен быть выведен на экран, а знак `:` (двоеточие) отменяет вывод и может использоваться лишь как знак разделителя при записи нескольких выражений в одной строке. Клавиши перемещения курсора позволяют передвигаться по ранее введенным строкам на экране.

Нижеприведенная строка иллюстрирует несколько базовых моментов. Операторы могут располагаться на одной строке, и в этом случае они

выполняются все сразу (в порядке их написания) при одном нажатии клавиши ввода. Знак := представляет собой оператор присваивания, причем он применяется как для присваивания значения переменной (a), выражения (g), так и функции (f).

```
> restart: a:=2: f:=x->sin(a*x)/x; g:=cos(b*x);
      f := x ->  $\frac{\sin(ax)}{x}$ 
      g := cos(bx)
```

Важно понимать принципиальное отличие заданных типов объектов f и g . Если f – это истинная функция, заданная в виде процедуры, точнее, оператора (переменная x переводится с помощью знака $->$, изображающего стрелку, в некоторое выражение), то g – строго говоря, вообще не функция, а лишь выражение, содержащее неопределенный символ x . Однако и в таком виде g можно использовать как полноценную функцию, только процедуры «вывода», т.е. вычисления значений в конкретных точках, различны для f и g . Если для первой достаточно написать $f(2)$, то для второй необходимо использовать оператор подстановки `subs`:

```
> f(2); subs(x=2,g);
       $\frac{1}{2} \sin(4)$ 
      cos(2b)
```

Для вычисления в привычном виде (в виде десятичного числа с мантиссой и порядком) надо воспользоваться функцией `evalf` – эта функция обеспечивает вычисление символьного выражения, заданного ее параметром (имя функции происходит от слова *evaluate*, а буква f в конце символизирует *float-point* – вычисления «с плавающей запятой»):

```
> evalf(f(2)); evalf(subs({x=2,b=1},g));
      -0.3784012476
      -0.4161468365
```

(в случае g для вычисления конкретного значения необходимо, помимо x , подставить также значение для b , по этой причине внутри оператора `subs`

использованы фигурные скобки для обозначения списка условий). Используем определенные функции для построения их графиков (внутри оператора построения графика `plot` также использованы фигурные скобки для обозначения списка функций):

```
> plot({f(x), subs(b=2,g)}, x=-15..15, color=black);
```

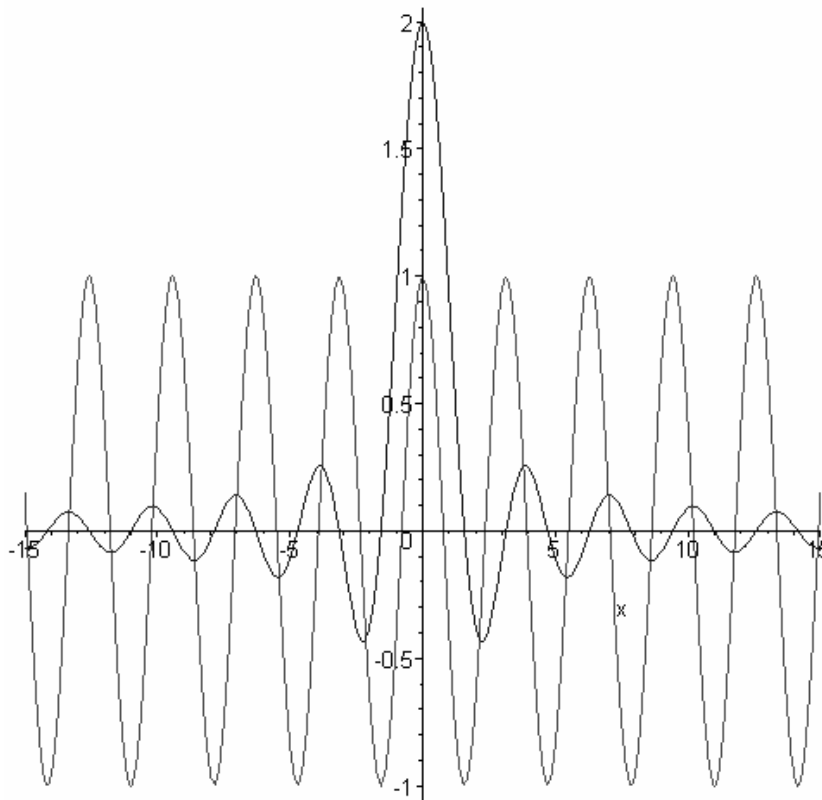


Рис. 1.2. Пример построения графиков функций от одной переменной в Maple

Обратим еще раз внимание, что здесь f и g используются по-разному: если первая записана в обычном для функции виде $f(x)$, то вторая записана просто как g , а ее зависимость от переменной x подразумевается, т.к. ранее символу g было присвоено значение $\cos(bx)$. Другой распространенный оператор, оператор равенства $=$, используется для задания равенств и логических условий (например, $a = b$), указания областей изменения переменных (например, $x=-15..15$ означает формирование диапазона изменения

x от -15 до 15) и определения значений параметров в функциях и командах (например, `color=black` для задания черного цвета у линий графиков).

Аналогично, как график обычной функции в декартовой системе координат, можно построить график трехмерной поверхности. В данном случае задана функция двух переменных $z(x, y) = \sin(x + y)$ и ее график строится с использованием графической функции `plot3d`. Правила задания пределов изменения переменных x и y соответствуют описанным выше.

```
> z:=sin(x+y):plot3d(z,x=-2..2,y=-2..2);
```

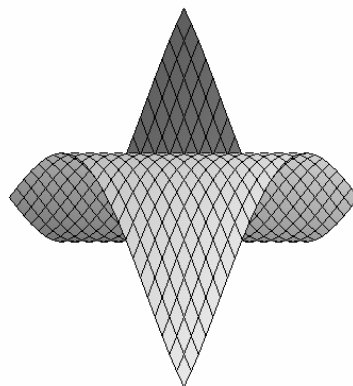


Рис. 1.3. Пример построения графика функции от двух переменных

Необходимо помнить, что сила системы Maple состоит прежде всего в возможности выполнения аналитических (символьных) вычислений. Рассмотрим некоторых примеры.

Требуется найти емкость трех последовательно включенных емкостей (конденсаторов) C_1 , C_2 и C_3 произвольной величины. Из электротехники известно, что суммарная емкость C_0 определяется уравнением:

$$\frac{1}{C_0} = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} . \quad (1.1)$$

Задав это уравнение и присвоив ему некоторое имя (например, *eq*), далее достаточно использовать функцию решения алгебраических уравнений `solve`, чтобы найти значение C_0 в общей аналитической форме:

```
> eq:=1/C0=1/C1+1/C2+1/C3:R:=solve(eq,C0);
```

$$R := \frac{C1 C2 C3}{C2 C3 + C1 C3 + C1 C2}$$

Результат может быть получен и в численном виде для конкретных значений C_1 , C_2 и C_3 :

```
> C1:=2:C2:=1:C3:=4:R;evalf(%);
```

$$\frac{4}{7}$$

```
0.5714285714
```

(знак % означает последнее предыдущее выражение). Отметим еще некоторые тонкие моменты, связанные с применением оператора solve. Выполним следующую командную строку:

```
> restart:eq:=1/C0=1/C1+1/C2+1/C3:R:=solve({eq},C0);
```

$$R := \{ C0 = \frac{C1 C2 C3}{C2 C3 + C1 C3 + C1 C2} \}$$

Отличие от предыдущего случая состоит лишь в наличии фигурных скобок, в которые заключено имя решаемого уравнения *eq*. Благодаря этому, как видно из результата выполнения оператора, теперь R (имя, которым мы обозначили процедуру решения уравнения) представляет собой массив (в данном случае размерности 1). В этом легко убедиться, выполнив оператор:

```
> R[1];
```

$$C0 = \frac{C1 C2 C3}{C2 C3 + C1 C3 + C1 C2}$$

Отметим, что предыдущие операции не определили значения символа C_0 :

```
> C0;
```

$$C0$$

Можно «извлечь» выражение C_0 , воспользовавшись оператором решения алгебраических уравнений solve:

```
> solve(R[1],C0);
```

$$\frac{C1 C2 C3}{C2 C3 + C1 C3 + C1 C2}$$

Полученное значение выражения C_0 можно закрепить за этим символом, если «утвердить» результат процедуры решения уравнения (которой мы дали имя R) с помощью оператора assign:

```
> assign(R):C0;
```

$$\frac{C1 C2 C3}{C2 C3 + C1 C3 + C1 C2}$$

Вычисления производных и интегралов в символьном виде являются наиболее интересными из простейших применений систем символьной математики. Ниже показаны примеры с применением функции diff для вычисления производной и int для вычисления неопределенных и определенных интегралов:

```
> diff(sin(x)^2+cos(x)^3,x);
```

$$2 \sin(x) \cos(x) - 3 \cos(x)^2 \sin(x)$$

```
> int(sin(x),x); int(sin(x),x=0..Pi);
```

$$-\cos(x)$$

$$2$$

Возможны случаи, когда интеграл выражается через специальные функции, а если это невозможно, то возвращается в символьном виде:

```
> q1:=int(exp(-x^2),x=-1..1); q2:=int(exp(-x^3),x=-1..1);
```

$$q1 := \operatorname{erf}(1) \sqrt{\pi}$$

$$q2 := \int_{-1}^1 e^{(-x^3)} dx$$

К такому же результату приводит функция **Int** – так называемая инертная форма функции **int**. Инертная форма служит для вывода записи интеграла в естественной математической форме, но с отложенным выводом результата вычислений. Это один из путей наглядного представления входных выражений, причем инертные формы имеют не только интегралы, но практически все функции. Все инертные функции имеют имена, начинающиеся с большой буквы, тогда как обычные функции имеют имена, начинающиеся с маленькой буквы.

```
> q3:=Int(sin(x),x=0..Pi);
```

$$q3 := \int_0^{\pi} \sin(x) dx$$

Во всех случаях можно посчитать численные значения:

```
> evalf(q1); evalf(q2); evalf(q3);  
1.493648266  
2.149415600  
2.000000000
```

Отметим в заключение, что настоящее пособие ни в коей мере не является систематическим учебником Maple, MATLAB или Mathematica, поэтому детальное изложение языка выходит за его рамки. Так, слова solve, diff и int с их аргументами являются именами встроенных в систему функций, возвращающих символьные значения результатов, и таких функций с их вариантами система Maple имеет около трех тысяч. При этом многие функции подчас могут применяться во многих случаях и имеют массу параметров и директив для уточнения направлений решения и расширения областей применения. В то же время, задача данного пособия состоит в том, чтобы читатель после его изучения мог самостоятельно использовать указанные системы для решения практических задач, не используя специальные систематические пособия. Это возможно в силу следующих важных обстоятельств:

- мало кто на практике использует из всей этой массы функций более чем несколько десятков;
- названия и формы представления многих функций интуитивно предсказуемы;
- наконец, система имеет превосходную справочную базу данных, с помощью которой можно легко разобраться с синтаксисом любой функции.

Задача настоящего пособия и состоит в предоставлении и разъяснении важнейших, базовых сведений и приемов, овладев которыми (с обязательным изучением примеров), читатель сможет работать самостоятельно.

1.3. Пример расчета физической системы в Maple

Цель данной лекции состоит в построении в Maple конкретного примера расчета модельной квантовой системы: прохождение частицы через одномерный прямоугольный потенциальный барьер [2, с. 101–102]. На этой задаче можно многому научиться. Детальный разбор такой задачи и доведение ее «до числа» позволит существенно продвинуться в изучении Maple и выйти на некий минимальный уровень практического владения системой для расчета и моделирования физических систем.

Теоретические сведения. Рассмотрим частицу массы m , способную перемещаться по оси x под действием некоторого потенциала $U(x)$ [2, с. 84]. Уравнение Шредингера записывается в виде:

$$i\hbar \frac{\partial}{\partial t} \Psi(x, t) = \left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + U(x) \right) \Psi(x, t). \quad (1.2)$$

Потоки частиц моделируются стационарными состояниями. Если E есть энергия стационарного состояния (энергия частицы), то волновая функция имеет вид:

$$\Psi(x, t) = \Phi(x) e^{-\frac{iE}{\hbar} t}, \quad (1.3)$$

причем функция $\Phi(x)$ является решением стационарного уравнения Шредингера (штрих обозначает дифференцирование по координате)

$$\Phi'' + \frac{2m}{\hbar^2} (E - U(x)) \Phi = 0. \quad (1.4)$$

Требуется найти решения данного уравнения, ограниченные, непрерывные и дифференцируемые во всем интервале $(-\infty, +\infty)$.

Рассмотрим случай прямоугольного потенциального барьера, который описывается кусочно-постоянной потенциальной функцией

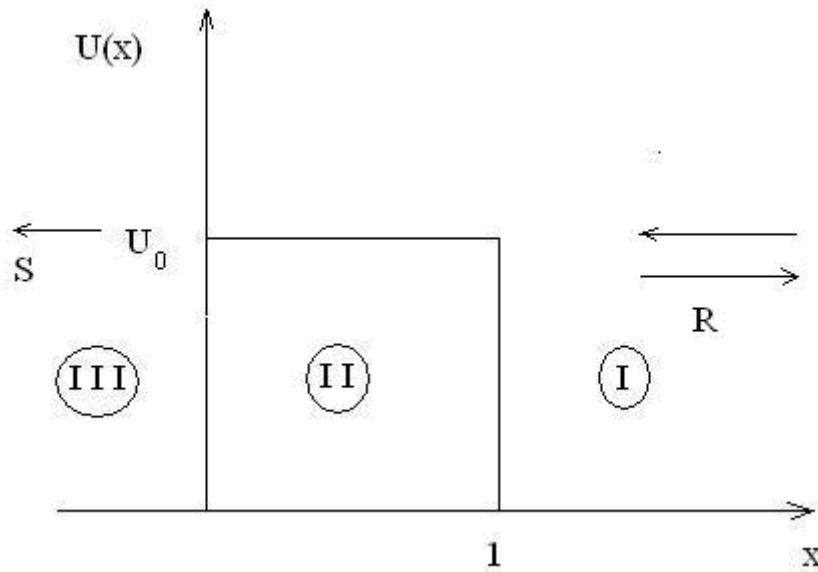


Рис. 1.4. Прямоугольный потенциальный барьер

$$U(x) = \begin{cases} 0, & x \leq 0; \\ U_0, & 0 < x < 1; \\ 0, & x \geq 1 \end{cases} \quad (1.5)$$

Весь интервал x разбивается на три области, пронумерованные на рисунке 1.4. В каждой области частица описывается своей волновой функцией, обозначим их как $\Phi_1(x)$, $\Phi_2(x)$, $\Phi_3(x)$. Они удовлетворяют уравнениям

$$\begin{aligned} \Phi_1'' + \frac{2m}{\hbar^2} E \Phi_1 &= 0, \\ \Phi_2'' + \frac{2m}{\hbar^2} (E - U_0) \Phi_2 &= 0, \\ \Phi_3'' + \frac{2m}{\hbar^2} E \Phi_3 &= 0. \end{aligned} \quad (1.6)$$

Проведем предварительный анализ данной задачи, тех проблем, которые встретятся при ее рассмотрении, и составим план ее решения. На таком предварительном этапе также возможно использование аналитического пакета, например, Maple. Для этого попытаемся получить общие реше-

ния дифференциальных уравнений (1.6). Однако необходимо конкретизировать знак величины E , а также знак разности $E - U_0$. Ясно, что E , будучи энергией частицы, не может быть отрицательной. Что касается разности, то она может быть и отрицательной, и положительной, в зависимости от того, меньше или больше величина E высоты потенциального барьера (1.5), изображенного на рисунке 1.4. Предположим для определенности, что $E < U_0$. Кроме того, необходимо уточнить и положительный знак U_0 (не следует забывать, что если для нас что-либо и очевидно, то программа этого знать не может!). Отразим все это в операторе *assume*, который описывает свойства переменных и соотношения между ними. Опишем после этого уравнения (1.6), придав им имена, например, *eq1*, *eq2*, *eq3*, и воспользуемся оператором *dsolve* (без каких-либо дополнительных опций) для получения общих решений этих уравнений. Последние будут выданы программой лишь в том случае, если они выражаются через элементарные или хотя бы специальные функции. Все вышеописанное реализуется в следующей программе.

```
> restart:
```

```
> assume (U0>0, E>0, E<U0) :
```

```
> eq1:=diff(phi1(x), x$2) + (2*m/h^2)*E*phi1(x) ;
```

```
eq2:=diff(phi2(x), x$2) + (2*m/h^2)*(E-U0)*phi2(x) ;
```

```
eq3:=diff(phi3(x), x$2) + (2*m/h^2)*E*phi3(x) ;
```

$$eq1 := \left(\frac{d^2}{dx^2} \phi_1(x) \right) + \frac{2 m E \phi_1(x)}{h^2}$$

$$eq2 := \left(\frac{d^2}{dx^2} \phi_2(x) \right) + \frac{2 m (-U_0 + E) \phi_2(x)}{h^2}$$

$$eq3 := \left(\frac{d^2}{dx^2} \phi_3(x) \right) + \frac{2 m E \phi_3(x)}{h^2}$$

```
> dsolve(eq1) ; dsolve(eq2) ; dsolve(eq3) ;
```

$$\phi_1(x) = _C1 \sin\left(\frac{\sqrt{2} \sqrt{m} \sqrt{E} x}{h}\right) + _C2 \cos\left(\frac{\sqrt{2} \sqrt{m} \sqrt{E} x}{h}\right)$$

$$\begin{aligned}\phi_2(x) &= {}_C1 e^{\left(\frac{\sqrt{2} \sqrt{m} \sqrt{U_0 - E} x}{h}\right)} + {}_C2 e^{\left(-\frac{\sqrt{2} \sqrt{m} \sqrt{U_0 - E} x}{h}\right)} \\ \phi_3(x) &= {}_C1 \sin\left(\frac{\sqrt{2} \sqrt{m} \sqrt{E} x}{h}\right) + {}_C2 \cos\left(\frac{\sqrt{2} \sqrt{m} \sqrt{E} x}{h}\right)\end{aligned}$$

Отметим, что величины, описанные с помощью оператора *assume*, в дальнейшем отображаются в виде с тильдой: $E\sim, U_0\sim$.

Общие решения уравнений получены, и они выражаются через элементарные функции. Мы приходим к следующим выводам. Решения в областях 1 и 3 удобно выразить не через E , а через параметр

$$\varepsilon = \frac{2m}{\hbar^2} E \quad (1.7)$$

(сравните с [2, с. 84]). Эти решения представляют собой линейные комбинации с неопределенными коэффициентами функций $\sin(\sqrt{\varepsilon}x)$ и $\cos(\sqrt{\varepsilon}x)$. Но поскольку общее решение исходного уравнения (1.2) ищется в виде (1.3), т.е. в виде экспонент с мнимым показателем, будем считать их линейными комбинациями функций $e^{i\sqrt{\varepsilon}x}$ и $e^{-i\sqrt{\varepsilon}x}$ (это возможно ввиду полной произвольности коэффициентов). Таким образом, в области 1 решение имеет вид

$$\Phi_1(x) = C e^{-i\sqrt{\varepsilon}x} + R e^{i\sqrt{\varepsilon}x}, \quad (1.8)$$

а в области 3 такой вид

$$\Phi_3(x) = S e^{-i\sqrt{\varepsilon}x} + D e^{i\sqrt{\varepsilon}x}. \quad (1.8)$$

Вспоминая, что, согласно (1.3), для получения общей волновой функции Ψ эти стационарные функции надо умножить на $e^{-\frac{iE}{\hbar}t}$. Таким образом получим:

$$\begin{aligned}\Psi_1(x) &= C e^{-i\left(\sqrt{\varepsilon}x + \frac{E}{\hbar}t\right)} + R e^{-i\left(\sqrt{\varepsilon}x - \frac{E}{\hbar}t\right)} \\ \Psi_3(x) &= S e^{-i\left(\sqrt{\varepsilon}x + \frac{E}{\hbar}t\right)} + D e^{-i\left(\sqrt{\varepsilon}x - \frac{E}{\hbar}t\right)}\end{aligned} \quad (1.9)$$

Итак, функции Ψ_1 и Ψ_2 представляют собой суперпозиции волн, распространяющихся в отрицательном направлении оси OX (с амплитудами C и S) и в положительном направлении (с амплитудами R и D). Иными словами, имеются волны (частицы), приходящие «из бесконечности» слева и справа, которые частично отражаются от барьера, частично проходят (туннелируют) сквозь него, и такие отраженные и преломленные волны накладываются на исходные. Вычислить коэффициент прохождения через барьер в такой ситуации затруднительно. Необходимо «очистить» картину, а именно: пусть имеется поток частиц, приходящих из бесконечности, например, справа, причем амплитуда этого потока нормирована на единицу. Слева от барьера тогда будет распространяться преломленная волна слева направо (с амплитудой S), а справа от него, помимо падающей, будет также отраженная волна с амплитудой R (как изображено с помощью стрелок на рисунке 1.4). Таким образом, в формулах (1.7) – (1.9) должно быть $C = 1$; $D = 0$, т.е. функции (1.7), (1.8) должны иметь вид

$$\begin{aligned}\Phi_1(x) &= e^{-i\sqrt{\epsilon}x} + R e^{i\sqrt{\epsilon}x} \\ \Phi_3(x) &= S e^{-i\sqrt{\epsilon}x}\end{aligned}\tag{1.10}$$

В этом случае квадраты амплитуд S^2, R^2 будут представлять собой коэффициенты прохождения и отражения соответственно, причем должен выполняться закон сохранения числа частиц

$$S^2 + R^2 = 1.\tag{1.11}$$

Соотношение (1.11) должно получиться автоматически и будет являться одним из критериев правильности проделанных вычислений.

Добиться выполнения выражений (1.10) можно, наложив некоторые дополнительные краевые условия, например, такие:

$$\Phi_1(0) + i\Phi_1\left(\frac{\pi}{2\sqrt{\epsilon}}\right) = 2$$

$$\Phi_3(0) - i\Phi_3\left(\frac{\pi}{2\sqrt{\varepsilon}}\right) = 0. \quad (1.12)$$

Помимо этого, должны удовлетворяться граничные условия непрерывности функций и их первых производных на границах областей 1, 2 и 3, т.е. в точках $x = 0, x = 1$:

$$\Phi_1(1) = \Phi_2(1); \Phi_2(0) = \Phi_3(0); \Phi_1'(1) = \Phi_2'(1); \Phi_2'(0) = \Phi_3'(0) \quad (1.13)$$

Итого имеется три уравнения второго порядка (1.6), в решения которых входят шесть произвольных констант. С другой стороны, имеются два краевых условия (1.12) и четыре краевых условий (1.13) – итого шесть условий. Следовательно, решение не будет содержать произвольных констант, т.е. будет полностью определенным. В частности, как указывалось выше, соотношение (1.11) должно будет выполняться безусловно, так как у нас не остается свободных констант для его удовлетворения.

На данном этапе анализа задачи, после выполнения предварительных пробных расчетов (в нашем случае это общие решения дифференциальных уравнений) и сделанных из них выводов, надо сформулировать план решения задачи и определить, что именно мы ожидаем от применения аналитических пакетов. В данном случае необходимо сделать следующее:

а) решить систему уравнений (1.6) с краевыми условиями (1.12), (1.13);

б) из полученных функций Φ_1, Φ_3 выделить выражения для S, R и возвести в квадрат их модули – это и будут коэффициенты прохождения и отражения от барьера;

в) проверить правильность вычислений, посчитав сумму квадратов коэффициентов S, R , которая, согласно (1.11), должна равняться единице;

г) проанализировать полученное выражение для коэффициента прохождения, в первую очередь физический интерес представляет исследова-

ние его зависимости от энергии частицы E , сравнить с известными выражениями (задача учебная!);

д) для полноты картины построить график волновых функций во всей области, проверить «сшивание» функций и их первых производных, проанализировать поведение волновой функции внутри барьера.

На каких же этапах будет полезно и/или необходимо применение аналитического пакета? В пункте а), в принципе, можно обойтись и без него, проведя «ручные» вычисления, поскольку общие решения содержат только элементарные функции. Вспомнив, однако, что функций у нас три, каждая из которых содержит по две произвольные константы, а краевых условий шесть, становится ясно, что вычисления предстоят весьма громоздкие, а чем более они громоздки, тем больше вероятность ошибиться. И это для такой простейшей задачи, как одномерный и прямоугольный потенциальный барьер! Она, конечно, может быть решена вручную, что и было сделано задолго до появления аналитических пакетов и вообще компьютеров. Интересно, тем не менее, что даже в таком фундаментальном курсе квантовой механики, как [2], детальный вывод коэффициента прохождения не приводится, автор пишет [2, с.101]: «Ограничимся тем, что дадим результат вычисления коэффициента прохождения:

$$T = |S|^2 = \begin{cases} \frac{4\varepsilon(\varepsilon - U_0)}{4\varepsilon(\varepsilon - U_0) + U_0^2 \sin^2(\sqrt{\varepsilon - U_0} L)}, & \text{если } \varepsilon > U_0 \\ \frac{4\varepsilon(U_0 - \varepsilon)}{4\varepsilon(U_0 - \varepsilon) + U_0^2 \operatorname{sh}^2(\sqrt{U_0 - \varepsilon} L)}, & \text{если } \varepsilon < U_0 \end{cases} \quad (1.14)$$

(здесь L – длина барьера, в нашем случае $L=1$). То есть вычисления возможны, но настолько громоздки и «рутинны», что А. Мессиа не посчитал целесообразным включение их в изложение своего курса. А немного более сложные (и реалистичные) задачи практически уже не позволяют проведения ручных вычислений. Наш учебный пример приобретает, таким образом, большое значение как некая базовая задача, на которой необходимо

«обкатать» технологию применения аналитических пакетов и который можно будет впоследствии обобщать на более сложные и интересные случаи.

В пунктах б), в) также возможно в принципе обойтись без компьютера, но ввиду громоздкости, аналитический пакет позволит провести вычисления проще и быстрее. Что же касается графиков, то преимущества применения компьютеров для их построения и анализа совершенно очевидны.

Итак, в случае нашей задачи применение «компьютерной алгебры» позволяет произвести вычисления с неизмеримо меньшими затратами труда и времени, чем «вручную» и является практически необходимым. Причем вычисления будут проводиться точно, т.е. будет задействована аналитическая компонента пакета, без использования его численной части.

Начало программы практически идентично приведенной выше программе для предварительного анализа: описание знаков E и $E - U_0$, задание уравнений. Отличие состоит в том, что теперь необходимо задать также граничные условия (1.12), (1.13), причем удобно всю совокупность этих условий обозначить каким-либо именем, например *ics*. Также имеет смысл дать некоторое имя системе дифференциальных уравнений, например *ode*. Кроме того, зададим для определенности значения массы и постоянной Планка: $m = 1/2$, $\hbar = 1$ и высоты барьера $U_0 = 40$ (для сопоставления с графиком, приведенном в [2, с.102], который соответствует такому значению). Опишем также параметр ε (1.7).

```
> restart:
> U0:=40:m:=1/2:h:=1:
> assume (E>0, E<U0) :
> epsilon:= (2*m/h^2)*E;
```

$\varepsilon := E\sim$

```
> eq1:=diff(phi1(x),x$2)+(2*m/h^2)*E*phi1(x);
eq2:=diff(phi2(x),x$2)+(2*m/h^2)*(E-U0)*phi2(x);
eq3:=diff(phi3(x),x$2)+(2*m/h^2)*E*phi3(x);
```

$$eq1 := \left(\frac{d^2}{dx^2} \phi_1(x) \right) + E \sim \phi_1(x)$$

$$eq2 := \left(\frac{d^2}{dx^2} \phi_2(x) \right) + (E \sim - 40) \phi_2(x)$$

$$eq3 := \left(\frac{d^2}{dx^2} \phi_3(x) \right) + E \sim \phi_3(x)$$

```
> ode:=eq1,eq2,eq3:
```

```
> ics := phi1(1)=phi2(1),phi2(0)=phi3(0),
```

```
D(phi1)(1)=D(phi2)(1),D(phi2)(0)=D(phi3)(0),phi1(0)+I*phi1(Pi/
2/sqrt(epsilon))=2, phi3(0)-I*phi3(Pi/2/sqrt(epsilon))=0 ;
```

```
ics := phi1(1) = phi2(1), phi2(0) = phi3(0), D(phi1)(1) = D(phi2)(1), D(phi2)(0) = D(phi3)(0),
```

$$\phi_1(0) + \phi_1\left(\frac{\pi}{2\sqrt{E\sim}}\right)I = 2, \phi_3(0) - \phi_3\left(\frac{\pi}{2\sqrt{E\sim}}\right)I = 0$$

Применим теперь ключевой оператор нашей программы *dsolve*, производящий точное решение дифференциальных уравнений, но в отличие от его применения на предварительном этапе без каких-либо опций (тогда в результате получили общие решения с неопределенными коэффициентами), теперь этот оператор будет использован в режиме нахождения решений с граничными условиями. Совершенно необходимо дать решению какое-либо имя, например *R*, если мы хотим далее использовать это решение, а не просто посмотреть на его распечатку. Кстати говоря, такая распечатка не является необходимой, и можно было бы поставить в конце оператора двоеточие, чтобы избежать отображения громоздкого выражения на экране. Однако, из методических соображений, поставим в конце точку с запятой и выведем решение на экран, поскольку на практике всегда именно так и приходится делать с тем, чтобы убедиться, что оно действительно получено, и посмотреть на него, что называется «вживую». Далее можно вернуть курсор на строчку оператора, поменять точку с запятой на двоеточие и выполнить оператор еще раз, в результате распечатка исчезнет. Кроме

того, распечатка решения служит иллюстрацией к вышеприведенным рассуждениям о громоздкости задачи и преимуществах применения компьютерной алгебры перед ручными вычислениями.

> R:=dsolve({ode,ics});

$$\begin{aligned}
R := \{ & \phi 3(x) = 2 \sqrt{E\sim} \sqrt{-E\sim + 40} \left(e^{(\sqrt{-E\sim + 40})^2} \sin(\sqrt{E\sim} x) \right) / (\\
& \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(3\sqrt{-E\sim + 40})} \cos(\sqrt{E\sim}) I - \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(3\sqrt{-E\sim + 40})} \sin(\sqrt{E\sim}) \\
& + e^{(3\sqrt{-E\sim + 40})} E\sim \cos(\sqrt{E\sim}) + e^{(3\sqrt{-E\sim + 40})} E\sim \sin(\sqrt{E\sim}) I \\
& + \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(\sqrt{-E\sim + 40})} \cos(\sqrt{E\sim}) I - \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(\sqrt{-E\sim + 40})} \sin(\sqrt{E\sim}) \\
& - e^{(\sqrt{-E\sim + 40})} E\sim \cos(\sqrt{E\sim}) - e^{(\sqrt{-E\sim + 40})} E\sim \sin(\sqrt{E\sim}) I \\
& - 20 I \sin(\sqrt{E\sim}) e^{(3\sqrt{-E\sim + 40})} + 20 I \sin(\sqrt{E\sim}) e^{(\sqrt{-E\sim + 40})} \\
& - 20 \cos(\sqrt{E\sim}) e^{(3\sqrt{-E\sim + 40})} + 20 \cos(\sqrt{E\sim}) e^{(\sqrt{-E\sim + 40})}) + 2 I \sqrt{E\sim} \sqrt{-E\sim + 40} \\
& \left(e^{(\sqrt{-E\sim + 40})^2} \cos(\sqrt{E\sim} x) \right) / \left(\sqrt{E\sim} \sqrt{-E\sim + 40} e^{(3\sqrt{-E\sim + 40})} \cos(\sqrt{E\sim}) I \right. \\
& - \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(3\sqrt{-E\sim + 40})} \sin(\sqrt{E\sim}) + e^{(3\sqrt{-E\sim + 40})} E\sim \cos(\sqrt{E\sim}) \\
& + e^{(3\sqrt{-E\sim + 40})} E\sim \sin(\sqrt{E\sim}) I + \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(\sqrt{-E\sim + 40})} \cos(\sqrt{E\sim}) I \\
& - \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(\sqrt{-E\sim + 40})} \sin(\sqrt{E\sim}) - e^{(\sqrt{-E\sim + 40})} E\sim \cos(\sqrt{E\sim}) \\
& - e^{(\sqrt{-E\sim + 40})} E\sim \sin(\sqrt{E\sim}) I - 20 I \sin(\sqrt{E\sim}) e^{(3\sqrt{-E\sim + 40})} \\
& \left. + 20 I \sin(\sqrt{E\sim}) e^{(\sqrt{-E\sim + 40})} - 20 \cos(\sqrt{E\sim}) e^{(3\sqrt{-E\sim + 40})} + 20 \cos(\sqrt{E\sim}) e^{(\sqrt{-E\sim + 40})} \right) \\
, \phi 2(x) = & \sqrt{E\sim} \left(e^{(\sqrt{-E\sim + 40})^2} \left(\sqrt{-E\sim + 40} - \sqrt{E\sim} I \right) e^{(\sqrt{-E\sim + 40} x)} I \right) / (\\
& \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(3\sqrt{-E\sim + 40})} \cos(\sqrt{E\sim}) I - \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(3\sqrt{-E\sim + 40})} \sin(\sqrt{E\sim}) \\
& + e^{(3\sqrt{-E\sim + 40})} E\sim \cos(\sqrt{E\sim}) + e^{(3\sqrt{-E\sim + 40})} E\sim \sin(\sqrt{E\sim}) I \\
& + \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(\sqrt{-E\sim + 40})} \cos(\sqrt{E\sim}) I - \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(\sqrt{-E\sim + 40})} \sin(\sqrt{E\sim}) \\
& - e^{(\sqrt{-E\sim + 40})} E\sim \cos(\sqrt{E\sim}) - e^{(\sqrt{-E\sim + 40})} E\sim \sin(\sqrt{E\sim}) I \\
& - 20 I \sin(\sqrt{E\sim}) e^{(3\sqrt{-E\sim + 40})} + 20 I \sin(\sqrt{E\sim}) e^{(\sqrt{-E\sim + 40})} \\
& - 20 \cos(\sqrt{E\sim}) e^{(3\sqrt{-E\sim + 40})} + 20 \cos(\sqrt{E\sim}) e^{(\sqrt{-E\sim + 40})}) + (\sqrt{-E\sim + 40} + \sqrt{E\sim} I) \\
& \sqrt{E\sim} \left(e^{(\sqrt{-E\sim + 40})^2} e^{(-\sqrt{-E\sim + 40} x)} I \right) / \left(\sqrt{E\sim} \sqrt{-E\sim + 40} e^{(3\sqrt{-E\sim + 40})} \cos(\sqrt{E\sim}) I \right. \\
& - \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(3\sqrt{-E\sim + 40})} \sin(\sqrt{E\sim}) + e^{(3\sqrt{-E\sim + 40})} E\sim \cos(\sqrt{E\sim}) \\
& + e^{(3\sqrt{-E\sim + 40})} E\sim \sin(\sqrt{E\sim}) I + \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(\sqrt{-E\sim + 40})} \cos(\sqrt{E\sim}) I \\
& \left. - \sqrt{E\sim} \sqrt{-E\sim + 40} e^{(\sqrt{-E\sim + 40})} \sin(\sqrt{E\sim}) - e^{(\sqrt{-E\sim + 40})} E\sim \cos(\sqrt{E\sim}) \right)
\end{aligned}$$

$$\begin{aligned}
& - e^{(\sqrt{-E\sim+40})} E\sim \sin(\sqrt{E\sim}) I - 20 I \sin(\sqrt{E\sim}) e^{(3\sqrt{-E\sim+40})} \\
& + 20 I \sin(\sqrt{E\sim}) e^{(\sqrt{-E\sim+40})} - 20 \cos(\sqrt{E\sim}) e^{(3\sqrt{-E\sim+40})} + 20 \cos(\sqrt{E\sim}) e^{(\sqrt{-E\sim+40})} \\
\phi 1(x) = & - (-2 e^{(\sqrt{-E\sim+40})} E\sim \cos(\sqrt{E\sim})^2 + 2 e^{(3\sqrt{-E\sim+40})} E\sim \cos(\sqrt{E\sim})^2 \\
& + e^{(\sqrt{-E\sim+40})} E\sim - e^{(3\sqrt{-E\sim+40})} E\sim - \sqrt{E\sim} \sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})} I \\
& + 40 I \cos(\sqrt{E\sim}) \sin(\sqrt{E\sim}) e^{(\sqrt{-E\sim+40})} - 40 \cos(\sqrt{E\sim})^2 e^{(3\sqrt{-E\sim+40})} \\
& - 2 \cos(\sqrt{E\sim}) \sqrt{E\sim} \sqrt{-E\sim+40} e^{(3\sqrt{-E\sim+40})} \sin(\sqrt{E\sim}) \\
& - 2 I \cos(\sqrt{E\sim}) e^{(\sqrt{-E\sim+40})} E\sim \sin(\sqrt{E\sim}) \\
& - 2 \cos(\sqrt{E\sim}) \sqrt{E\sim} \sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})} \sin(\sqrt{E\sim}) + 40 \cos(\sqrt{E\sim})^2 e^{(\sqrt{-E\sim+40})} \\
& - \sqrt{E\sim} \sqrt{-E\sim+40} e^{(3\sqrt{-E\sim+40})} I + 2 I \sqrt{-E\sim+40} e^{(3\sqrt{-E\sim+40})} \sqrt{E\sim} \cos(\sqrt{E\sim})^2 \\
& + 2 I \sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})} \sqrt{E\sim} \cos(\sqrt{E\sim})^2 \\
& + 2 I \cos(\sqrt{E\sim}) e^{(3\sqrt{-E\sim+40})} E\sim \sin(\sqrt{E\sim}) - 40 I \cos(\sqrt{E\sim}) \sin(\sqrt{E\sim}) e^{(3\sqrt{-E\sim+40})} \\
\sin(\sqrt{E\sim} x) / & (2 \sqrt{-E\sim+40} e^{(3\sqrt{-E\sim+40})} \sqrt{E\sim} \cos(\sqrt{E\sim})^2 \\
& + 2 I \cos(\sqrt{E\sim}) \sqrt{E\sim} \sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})} \sin(\sqrt{E\sim}) - 20 I e^{(3\sqrt{-E\sim+40})} \\
& + 40 I \cos(\sqrt{E\sim})^2 e^{(3\sqrt{-E\sim+40})} + 2 \cos(\sqrt{E\sim}) e^{(3\sqrt{-E\sim+40})} E\sim \sin(\sqrt{E\sim}) \\
& + 2 \sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})} \sqrt{E\sim} \cos(\sqrt{E\sim})^2 + e^{(3\sqrt{-E\sim+40})} E\sim I \\
& - 40 I \cos(\sqrt{E\sim})^2 e^{(\sqrt{-E\sim+40})} - 2 \cos(\sqrt{E\sim}) e^{(\sqrt{-E\sim+40})} E\sim \sin(\sqrt{E\sim}) \\
& - 40 \cos(\sqrt{E\sim}) \sin(\sqrt{E\sim}) e^{(3\sqrt{-E\sim+40})} + 40 \cos(\sqrt{E\sim}) \sin(\sqrt{E\sim}) e^{(\sqrt{-E\sim+40})} \\
& + 2 I e^{(\sqrt{-E\sim+40})} E\sim \cos(\sqrt{E\sim})^2 - 2 I e^{(3\sqrt{-E\sim+40})} E\sim \cos(\sqrt{E\sim})^2 \\
& - \sqrt{E\sim} \sqrt{-E\sim+40} e^{(3\sqrt{-E\sim+40})} \\
& + 2 I \cos(\sqrt{E\sim}) \sqrt{E\sim} \sqrt{-E\sim+40} e^{(3\sqrt{-E\sim+40})} \sin(\sqrt{E\sim}) - e^{(\sqrt{-E\sim+40})} E\sim I \\
& + 20 I e^{(\sqrt{-E\sim+40})} - \sqrt{E\sim} \sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})}) + (\\
2 I \cos(\sqrt{E\sim}) \sqrt{E\sim} \sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})} \sin(\sqrt{E\sim}) - 40 I e^{(3\sqrt{-E\sim+40})} \\
& + 40 I \cos(\sqrt{E\sim})^2 e^{(3\sqrt{-E\sim+40})} + e^{(3\sqrt{-E\sim+40})} E\sim I \\
& + 2 \cos(\sqrt{E\sim}) e^{(3\sqrt{-E\sim+40})} E\sim \sin(\sqrt{E\sim}) + 40 \cos(\sqrt{E\sim}) \sin(\sqrt{E\sim}) e^{(\sqrt{-E\sim+40})} \\
& - 40 I \cos(\sqrt{E\sim})^2 e^{(\sqrt{-E\sim+40})} + 2 I e^{(\sqrt{-E\sim+40})} E\sim \cos(\sqrt{E\sim})^2 \\
& - 2 I e^{(3\sqrt{-E\sim+40})} E\sim \cos(\sqrt{E\sim})^2 \\
& + 2 I \cos(\sqrt{E\sim}) \sqrt{E\sim} \sqrt{-E\sim+40} e^{(3\sqrt{-E\sim+40})} \sin(\sqrt{E\sim}) \\
& + 2 \sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})} \sqrt{E\sim} \cos(\sqrt{E\sim})^2 - 2 \cos(\sqrt{E\sim}) e^{(\sqrt{-E\sim+40})} E\sim \sin(\sqrt{E\sim}) \\
& - e^{(\sqrt{-E\sim+40})} E\sim I - \sqrt{E\sim} \sqrt{-E\sim+40} e^{(3\sqrt{-E\sim+40})} \\
& + 2 \sqrt{-E\sim+40} e^{(3\sqrt{-E\sim+40})} \sqrt{E\sim} \cos(\sqrt{E\sim})^2 + 40 I e^{(\sqrt{-E\sim+40})}
\end{aligned}$$

$$\begin{aligned}
& -40 \cos(\sqrt{E\sim}) \sin(\sqrt{E\sim}) e^{(3\sqrt{-E\sim+40})} - \sqrt{E\sim} \sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})} \cos(\sqrt{E\sim} x) \\
& / (2\sqrt{-E\sim+40} e^{(3\sqrt{-E\sim+40})} \sqrt{E\sim} \cos(\sqrt{E\sim})^2 \\
& + 2I \cos(\sqrt{E\sim}) \sqrt{E\sim} \sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})} \sin(\sqrt{E\sim}) - 20I e^{(3\sqrt{-E\sim+40})} \\
& + 40I \cos(\sqrt{E\sim})^2 e^{(3\sqrt{-E\sim+40})} + 2 \cos(\sqrt{E\sim}) e^{(3\sqrt{-E\sim+40})} E\sim \sin(\sqrt{E\sim}) \\
& + 2\sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})} \sqrt{E\sim} \cos(\sqrt{E\sim})^2 + e^{(3\sqrt{-E\sim+40})} E\sim I \\
& - 40I \cos(\sqrt{E\sim})^2 e^{(\sqrt{-E\sim+40})} - 2 \cos(\sqrt{E\sim}) e^{(\sqrt{-E\sim+40})} E\sim \sin(\sqrt{E\sim}) \\
& - 40 \cos(\sqrt{E\sim}) \sin(\sqrt{E\sim}) e^{(3\sqrt{-E\sim+40})} + 40 \cos(\sqrt{E\sim}) \sin(\sqrt{E\sim}) e^{(\sqrt{-E\sim+40})} \\
& + 2I e^{(\sqrt{-E\sim+40})} E\sim \cos(\sqrt{E\sim})^2 - 2I e^{(3\sqrt{-E\sim+40})} E\sim \cos(\sqrt{E\sim})^2 \\
& - \sqrt{E\sim} \sqrt{-E\sim+40} e^{(3\sqrt{-E\sim+40})} \\
& + 2I \cos(\sqrt{E\sim}) \sqrt{E\sim} \sqrt{-E\sim+40} e^{(3\sqrt{-E\sim+40})} \sin(\sqrt{E\sim}) - e^{(\sqrt{-E\sim+40})} E\sim I \\
& + 20I e^{(\sqrt{-E\sim+40})} - \sqrt{E\sim} \sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})} \}
\end{aligned}$$

Получившееся выражение имеет структуру $R = \{\Phi_3(x) = \dots; \Phi_2(x) = \dots; \Phi_1(x) = \dots\}$. Далее можно применить оператор присваивания *assign* к имени, под которым у нас фигурирует решение – R . В результате произойдет присваивание всем величинам, входящим в R , их значений, фигурирующих в этом массиве. Т.е. $\Phi_1(x), \Phi_2(x), \Phi_3(x)$ обретут свои выражения согласно распечатанному массиву R . А тогда, имея в виду (1.10), можно получить амплитуду прошедшей сквозь барьер волны S , просто подставив $x=0$ в $\Phi_3(x)$, используя уже знакомый нам оператор *subs*. Далее вычисляем коэффициент прохождения как квадрат модуля S , умножая S на его комплексно сопряженное (оператор *conjugate*).

> **assign(R) :**

> **S:=simplify(subs(x=0, phi3(x))) ;**

$$\begin{aligned}
S := & 2I \sqrt{E\sim} \sqrt{-E\sim+40} e^{(\sqrt{-E\sim+40})} / (\sqrt{E\sim} \sqrt{-E\sim+40} e^{(2\sqrt{-E\sim+40})} \cos(\sqrt{E\sim}) I \\
& - \sqrt{E\sim} \sqrt{-E\sim+40} e^{(2\sqrt{-E\sim+40})} \sin(\sqrt{E\sim}) + e^{(2\sqrt{-E\sim+40})} E\sim \cos(\sqrt{E\sim}) \\
& + e^{(2\sqrt{-E\sim+40})} E\sim \sin(\sqrt{E\sim}) I + \sqrt{E\sim} \sqrt{-E\sim+40} \cos(\sqrt{E\sim}) I \\
& - \sqrt{E\sim} \sqrt{-E\sim+40} \sin(\sqrt{E\sim}) - E\sim \cos(\sqrt{E\sim}) - E\sim \sin(\sqrt{E\sim}) I \\
& - 20I \sin(\sqrt{E\sim}) e^{(2\sqrt{-E\sim+40})} + 20I \sin(\sqrt{E\sim}) - 20 \cos(\sqrt{E\sim}) e^{(2\sqrt{-E\sim+40})} \\
& + 20 \cos(\sqrt{E\sim})
\end{aligned}$$

>> **T:=simplify(S*conjugate(S)) ;**

$$T := \frac{E_{\sim} (E_{\sim} - 40) e^{(2\sqrt{40-E_{\sim}})}}{-100 - 100 e^{(4\sqrt{40-E_{\sim}})} + 200 e^{(2\sqrt{40-E_{\sim}})} + e^{(2\sqrt{40-E_{\sim}})} E_{\sim}^2 - 40 e^{(2\sqrt{40-E_{\sim}})} E_{\sim}}$$

Обратим внимание на оператор *simplify*, который очень часто употребляется в Maple. Как следует из его названия, он позволяет упрощать выражения, что бывает необходимо не только из эстетических соображений, но и для возможности продолжения вычислений, которые иначе, выражаясь компьютерным языком, «затыкаются». Читателю рекомендуется, например, выполнить последний оператор без применения *simplify* и сравнить полученные выражения.

Итак, коэффициент прохождения T получен. Аналогично можно получить коэффициент отражения как квадрат модуля амплитуды R , которую, согласно (1.10), можно извлечь из $\Phi_1(x)$ следующим образом:

$$R = \Phi_1(0) - 1. \quad (1.15)$$

Однако мы не можем использовать в программе имя R , которое уже занято. Обозначим поэтому амплитуду отраженной волны малой буквой r , а квадрат ее модуля как $r2$:

```
> r:=simplify(subs(x=0,phi1(x))-
1):r2:=simplify(evalc(r*conjugate(r)));
```

$$r2 := -\frac{100 (e^{(2\sqrt{40-E_{\sim}})} - 1)^2}{-100 - 100 e^{(4\sqrt{40-E_{\sim}})} + 200 e^{(2\sqrt{40-E_{\sim}})} + e^{(2\sqrt{40-E_{\sim}})} E_{\sim}^2 - 40 e^{(2\sqrt{40-E_{\sim}})} E_{\sim}}$$

Теперь все готово для проверки соотношения (1.11):

```
> simplify(r2+T);
```

1

(попробуйте опять выполнить этот оператор без *simplify*, в виде $r2+T$ и сравните результаты!)

Сравним полученный коэффициент прохождения T с известным из литературы выражением (1.14). Опишем формулу (1.14) для случая $\varepsilon < U_0$ под именем Q . Равны ли между собой Q и T ? Применим *simplify* к их разности:

```
> Q:=4*epsilon*(U0-epsilon)/(4*epsilon*(U0-epsilon)+U0^2*sinh(sqrt(U0-epsilon))^2);
```

$$Q := \frac{4 E \sim (40 - E \sim)}{4 E \sim (40 - E \sim) + 1600 \sinh(\sqrt{40 - E \sim})^2}$$

```
> simplify(T-Q);
```

$$\frac{-100 (E \sim - 40) E \sim (-4 e^{(2\sqrt{40-E \sim})} \cosh(\sqrt{40-E \sim})^2 + 2 e^{(2\sqrt{40-E \sim})} + 1 + e^{(4\sqrt{40-E \sim})})}{((40 e^{(2\sqrt{40-E \sim})} E \sim + 100 - e^{(2\sqrt{40-E \sim})} E \sim^2 - 200 e^{(2\sqrt{40-E \sim})} + 100 e^{(4\sqrt{40-E \sim})}) (E \sim^2 - 40 E \sim - 400 \cosh(\sqrt{40-E \sim})^2 + 400)}$$

Ожидаемого нуля для $T-Q$ не получилось. Тем не менее, поскольку мы надеемся, что вычисления проведены правильно, выражения T и Q должны совпадать! Подставим в T и Q вместо E одно и то же значение $E < U_0$:

```
> evalf(subs(E=38,T)); evalf(subs(E=38,Q));
```

0.04829096249

0.04829096249

Значения совпадают. Более того, анализируя структуру выражений, можно заметить, что они зависят от разности U_0 (в нашем примере 40) и E , которая стоит под корнем. При вычислениях мы предполагали $E < U_0$. Можно предположить, что при $E > U_0$, поскольку разность $E - U_0$ сменит знак, появится мнимая единица и экспоненты в нашем выражении для T дадут тригонометрические функции, и наше вычисленное T перейдет в выражение (1.14) для случая $E > U_0$. Действительно, легко видеть, что в (1.14) оба случая представляют собой практически одно и то же выражение, которые переходят одно в другое при смене знака $E - U_0$. Проверим, подставив в T и Q вместо E одно и то же значение $E > U_0$:

```
> evalf(subs(E=50,T)); evalf(subs(E=50,Q));
```

0.9996578700 + 0.1157016412 10⁻¹⁰ I

0.9996578700

Выражения, как и следовало ожидать, совпали. Отметим, что небольшой «дефект» в первом выражении, возникший из-за применения численных расчетов в функции *evalf*, никакого значения не имеет, поскольку равен

10^{-10} , т.е. нулю. Наберемся теперь смелости и построим зависимости T и Q от E на одном графике в диапазоне от 0 до $4U_0$. Оператор построения графиков *plot* имеет много опций, которые можно посмотреть в Help, набрав в командной строке знак вопроса и то слово, которым вы интересуетесь:

```
> ?plot
```

Мы используем лишь небольшое число из возможностей оператора построения графиков. Поскольку строятся две кривые на одном графике, их имена необходимо поместить в массив с помощью фигурных скобок. Далее надо указать переменную, откладываемую по горизонтальной оси и диапазон ее изменения (отметьте, что применяются две, а не три точки!) Поскольку мы ожидаем, что обе кривые совпадут, имеет смысл построить их по-разному, например, одну в виде сплошной, а другую в виде точечной линии. Это можно сделать, используя опцию *style*, перечисление параметров в которой соответствует порядку перечисления выражений в фигурных скобках. Наконец, с помощью опции *title* напишем название графика.

```
> plot({T,Q}, E=0..4*U0, style=[line,point], title="Коэффициент  
прохождения в зависимости от энергии");
```

График показан на рисунке 1.5. Полученный график в точности совпадает с [2, с. 102]. Таким образом, нам удалось получить нетривиальный результат, приведенный в [2] без вывода и доказательства. Это сделано с минимальными затратами труда и времени лишь благодаря применению численно-аналитического пакета компьютерной алгебры.

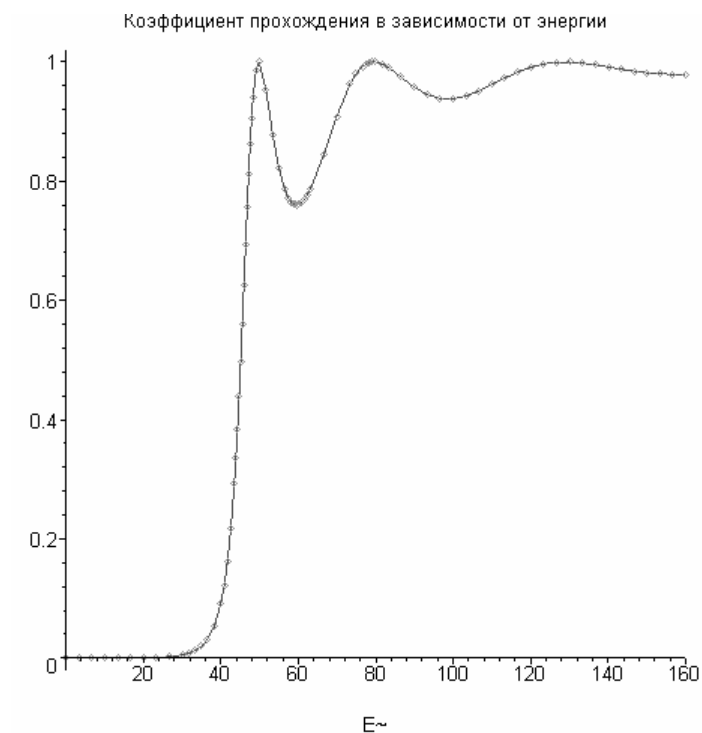


Рис. 1.5. Изменение коэффициента прохождения в зависимости от энергии для потенциального барьера, показанного на рисунке 1.4. Принято $U_0 = 40$

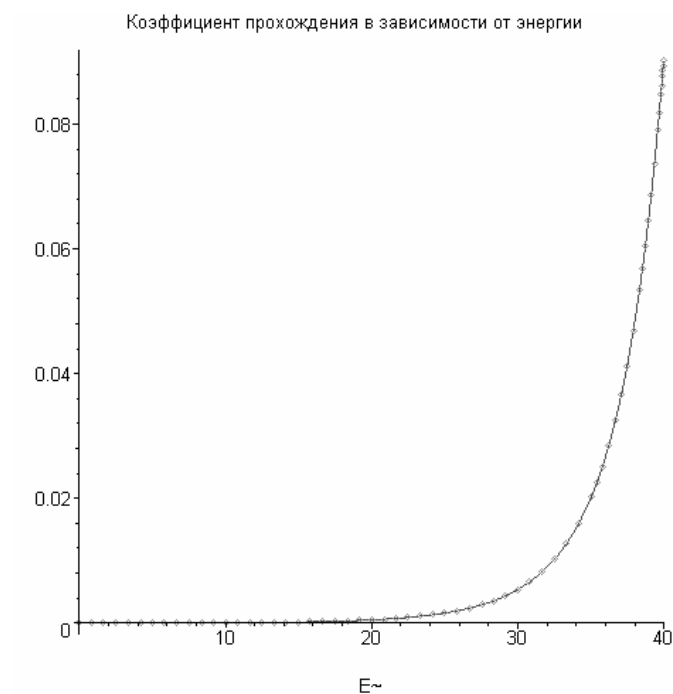


Рис. 1.6. Изменение коэффициента прохождения в зависимости от энергии для потенциального барьера на рисунке 1.4, в диапазоне $0 < E < U_0$. Принято $U_0 = 40$

Можно более подробно посмотреть график в диапазоне $E < U_0$:

```
>plot({T,Q},E=0..U0,style=[line,point],title="Коэффициент про-
хождения в зависимости от энергии");
```

Графики T и Q совпадают. Теперь нет никакого сомнения: полученное нами выражение для коэффициента прохождения совпадает с известным из литературы выражением (1.14), причем для частиц с энергиями как меньшими, так и большими высоты барьера. Но почему же не удалось упростить выражения так, чтобы разность $T-Q$ обратилась в ноль? Оператор *simplify* сам по себе не справился с этой задачей, что нередко случается в Maple. Иногда такие трудности удается обойти с помощью дополнительных опций в *simplify* (посмотрите в Help), иногда имеет смысл комбинировать *simplify* с такими операторами, как *evalc* (от слов evaluate complex – явное выделение действительной и мнимой части из комплексного выражения), *evalf*, *expand*, *coeff* и т.п. (ознакомьтесь с этими операторами в Help). Читателю предоставляется возможность попытаться добиться обращения разности $T-Q$ в ноль, комбинируя опции и операторы. Здесь добьемся этого по-другому: «поможем» оператору *simplify*, заменив в выражении для Q гиперболический синус (*sinh*) на его явное выражение через экспоненты $sh(\alpha) = (e^\alpha - e^{-\alpha})/2$:

```
> Q:=4*epsilon*(U0-epsilon)/(4*epsilon*(U0-
epsilon)+U0^2*((exp(sqrt(U0-epsilon))-exp(-sqrt(U0-
epsilon)))/2)^2);
```

$$Q := \frac{4 E \sim (40 - E \sim)}{4 E \sim (40 - E \sim) + 1600 \left(\frac{1}{2} e^{(\sqrt{40 - E \sim})} - \frac{1}{2} e^{(-\sqrt{40 - E \sim})} \right)^2}$$

```
> simplify(T-Q);
```

0

Наконец, построим графики получившихся (стационарных) волновых функций (1.10) (точнее, квадратов их модулей). Воспользуемся операторами *evalf*, *evalc*, *simplify* для приведения выражений к максимально уп-

рощенному виду и для вычисления всех входящих в них коэффициентов. Подставим $E = 38$, т.е. значение немного ниже высоты барьера.

```
> EE:=38:
> f3:=simplify(evalf(evalc(subs(E=EE,phi3(x))*conjugate(subs(E=EE,phi3(x))))));
f1:=simplify(evalf(evalc(subs(E=EE,phi1(x))*conjugate(subs(E=EE,phi1(x))))));
f2:=simplify(evalf(evalc(subs(E=EE,phi2(x))*conjugate(subs(E=EE,phi2(x))))));
```

$$f3 := 0.04829096238$$

$$f1 := 0.06697212262 + 3.769473828 \cos(6.164414003 x)^2 + 1.009163835 \sin(6.164414003 x) \cos(6.164414003 x)$$

$$f2 := 0.2414548124 e^{(2.828427124 x)} - 0.4346186622 + 0.2414548123 e^{(-2.828427124 x)}$$

Как и следовало ожидать, решение внутри барьера выражается через экспоненты, а вне барьера – через тригонометрические функции. Естественно, в области слева от барьера после всех упрощений получилась константа. Напомним, что это не сама волновая функция, а квадрат ее модуля, т.е. плотность вероятности, а из (1.10) ясно, что квадрат модуля функции Φ_3 равен S^2 , т.е. коэффициенту прохождения, значение которого при $E = 38$ мы уже получали выше. Построим графики этих функций, причем здесь в отличие от предыдущего применения оператора *plot* имеем не только несколько функций, но и у каждой свой интервал изменения переменной. Как построить такой график – ясно из следующей строчки.

```
> plot({[x, f1, x=1..3],[x, f2, x=0..1],[x, f3, x=-1..0]});
```

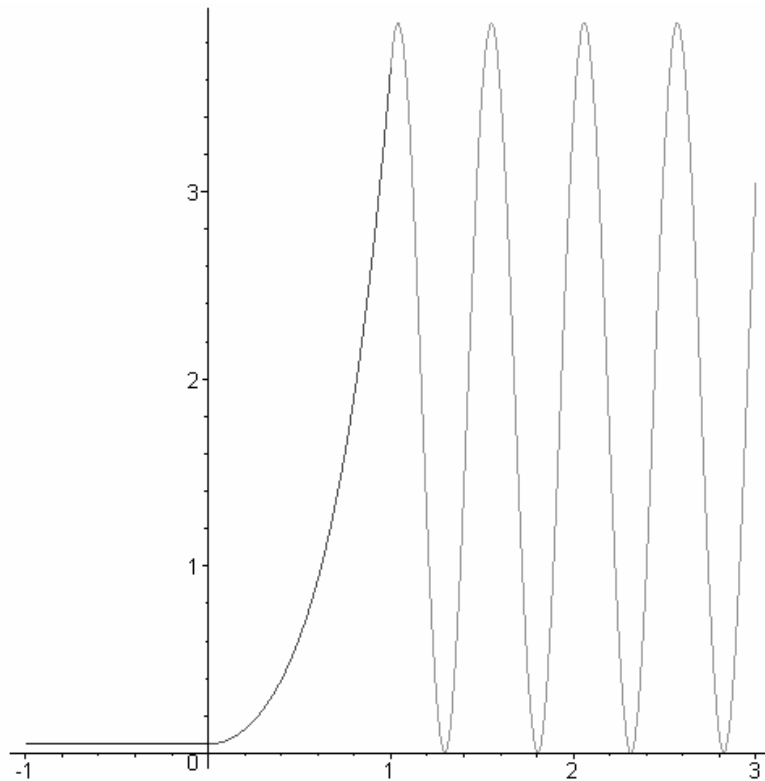


Рис. 1.7. Плотность вероятности для частицы с энергией $E = 38$ при высоте барьера $U_0 = 40$. Барьер локализован на участке от 0 до 1

Список литературы к теме 1

Использованная литература:

1. Сдвижков О.А. Математика на компьютере: Maple 8. – Солон-пресс, 2003. – 244 с.
2. Мессиа Альберт. Квантовая механика. – М.: Наука, 1978. – 480 с.
3. Дьяконов В.П., Maple 9.5/10 в математике, физике и образовании. Серия «Библиотека профессионала.» – М.: Изд-во «Солон», 2006. – 720 с.

Рекомендуемая литература:

1. Васильев А.Н. Maple 8. Самоучитель. – М.: Диалектика, Вильямс, 2003.

2. Алексеев Е.Р., Чеснокова О.В. Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9. – М.: Изд-во «ИТ Пресс», 2006. – 496 с.
3. Гандер В., Гржебичек И. Решение задач в научных вычислениях с применением Maple и MATLAB. – М.: Изд-во «Вассамедина», 2005. – 520 с.

Интернет-ресурсы к теме 1

1. Аладьев Виктор. Система компьютерной алгебры Maple: Библиотека новых программных средств.
www.lib.grsu.by/library/data/links/aladiev/Download/Maple_Lib.htm
2. Руководство пользователя MAPLE.
<http://exponenta.ru/soft/Maple/manson/poweredition/0.asp>
3. Области применения систем компьютерной алгебры и конкретные примеры.
<http://exponenta.ru/educat/systemat/systemat.asp>.

Контрольные вопросы к теме 1

1. Каковы были первые системы компьютерной алгебры?
2. Что представляет собой интегрированная система?
3. Что такое ядро системы?
4. Какие можете назвать дополнительные пакеты (библиотеки), загружаемые в Maple командой *with*?
5. Какие языки и каких уровней использованы в Maple?
6. Что такое XML?
7. Какими способами можно задавать функции в Maple?
8. Постройте графики однополостного и двуполостного гиперболоидов в Maple.

9. Напишите программу в Maple для решения какой-либо конкретной электрической цепи с помощью правил Кирхгофа.
10. Напишите программу в Maple для решения квантовой задачи для частицы в одномерном потенциальном ящике.
11. Решите в Maple какое-либо уравнение Бесселя и постройте график его решений для различных значений параметров.

Тема 2. Пакет MATLAB – начальные сведения

2.1. Введение

MATLAB представляет собой одну из старейших, тщательно проработанных и проверенных временем систем автоматизации математических расчетов, построенную на расширенном представлении и применении матричных операций. Это нашло отражение в названии системы MATrix LABoratory – матричная лаборатория. Интересно отметить, что даже обычные числа и переменные в MATLAB рассматриваются как матрицы размера 1×1 , что дает единообразные формы и методы проведения операций над обычными числами и массивами. Данная операция обычно называется векторизацией. Векторизация обеспечивает и упрощение записи операций, производимых одновременно над всеми элементами векторов и матриц, и существенное повышение скорости их выполнения [1, с. 6]. Это также означает, что большинство функций может работать с аргументами в виде векторов и матриц. При необходимости векторы и матрицы преобразуются в массивы, и значения вычисляются для каждого их элемента. Однако синтаксис языка программирования системы продуман настолько тщательно, что эта ориентация почти не ощущается теми пользователями, которых не интересуют непосредственно матричные вычисления.

Матрицы широко применяются в сложных математических расчетах, например при решении задач линейной алгебры и математического моделирования статических и динамических систем и объектов. Они являются основой автоматического составления и решения уравнений состояния динамических объектов и систем. Примером может служить расширение MATLAB – Simulink. Это существенно повышает интерес к системе MATLAB, вобравшей в себя лучшие достижения в области быстрого решения матричных задач [1].

Однако в настоящее время MATLAB далеко вышла за пределы специализированной матричной системы и стала одной из наиболее мощных универсальных интегрированных систем компьютерной алгебры. Слово «интегрированная» указывает на то, что в этой системе объединены удобная оболочка, редактор выражений и текстовых комментариев, вычислитель и графический программный процессор. В новой версии используются такие мощные типы данных, как многомерные массивы, массивы ячеек, массивы структур, массивы Java и разреженные матрицы, что открывает возможности применения системы при создании и отладке новых алгоритмов матричных и основанных на них параллельных вычислений и крупных баз данных. В целом MATLAB – это уникальная коллекция реализаций современных численных методов компьютерной математики, созданных за последние три десятка лет. Она вобрала в себя опыт, правила и методы математических вычислений, накопленные за тысячи лет развития математики. Это сочетается с мощными средствами графической визуализации и даже анимационной графики. Систему с прилагаемой к ней обширной документацией вполне можно рассматривать как фундаментальный многотомный электронный справочник по математическому обеспечению ЭВМ от массовых персональных компьютеров до супер-ЭВМ.

Система MATLAB была разработана Молером (С.В. Moler) и с конца 70-х гг. широко использовалась на больших ЭВМ. В начале 80-х гг. Джон Литл (John Little) из фирмы MathWorks, Inc. разработал версии системы PC MATLAB для компьютеров класса IBM PC, VAX и Macintosh. В дальнейшем были созданы версии для рабочих станций Sun, компьютеров с операционной системой UNIX и многих других типов больших и малых ЭВМ. Сейчас свыше десятка популярных компьютерных платформ могут работать с системой MATLAB. К расширению системы были привлечены крупнейшие научные школы мира в области математики, программирования и естествознания. Одной из основных задач системы было предостав-

ление пользователям мощного языка программирования, ориентированного на математические расчеты и способного превзойти возможности традиционных языков программирования, которые многие годы использовались для реализации численных методов. При этом особое внимание уделялось как повышению скорости вычислений, так и адаптации системы к решению самых разнообразных задач пользователей.

Возможности MATLAB весьма обширны, а по скорости выполнения задач система нередко превосходит своих конкурентов. Она применима для расчетов практически в любой области науки и техники. Например, она очень широко используется при математическом моделировании механических устройств и систем, в частности в динамике, гидродинамике, аэродинамике, акустике, энергетике и т.д. Этому способствует не только расширенный набор матричных и иных операций и функций, но и наличие пакета расширения (toolbox) Simulink, специально предназначенного для решения задач блочного моделирования динамических систем и устройств, а также десятков других пакетов расширений.

Говоря о графических возможностях системы, следует отметить, что 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. 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/XP, графики могут распола-

гаться 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.

Полезно знать, что в начале запуска автоматически выполняется команда `matlabrc`, которая исполняет загрузочный файл `matlabrc.m` и файл `startup.m`, если таковой существует. Эти файлы выполняют начальную настройку терминала системы и задают ряд ее параметров. В частности, могут быть заданы пути доступа к другим файлам, необходимым для корректной работы системы MATLAB. Таким образом, опытные пользователи могут выполнить настройку системы под свои запросы. Однако в большинстве случаев особой необходимости в этом нет. Поскольку указанные файлы имеют текстовый формат, их легко просмотреть с помощью какого-либо текстового редактора или с помощью команды `type` в командном режиме работы MATLAB. 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).

2.2. Начальные навыки работы с MATLAB

Ознакомимся с работой MATLAB на практике. При запуске MATLAB появляется командное окно, в левой части которого возможно появление областей Command History – Перечень предыдущих команд, Current

Directory – Браузер локального диска и др. (их можно вызывать или убирать с помощью опций в меню «Desktop»). Причем обычно в Command History по умолчанию загружается последний файл, с которым работали перед выключением системы.

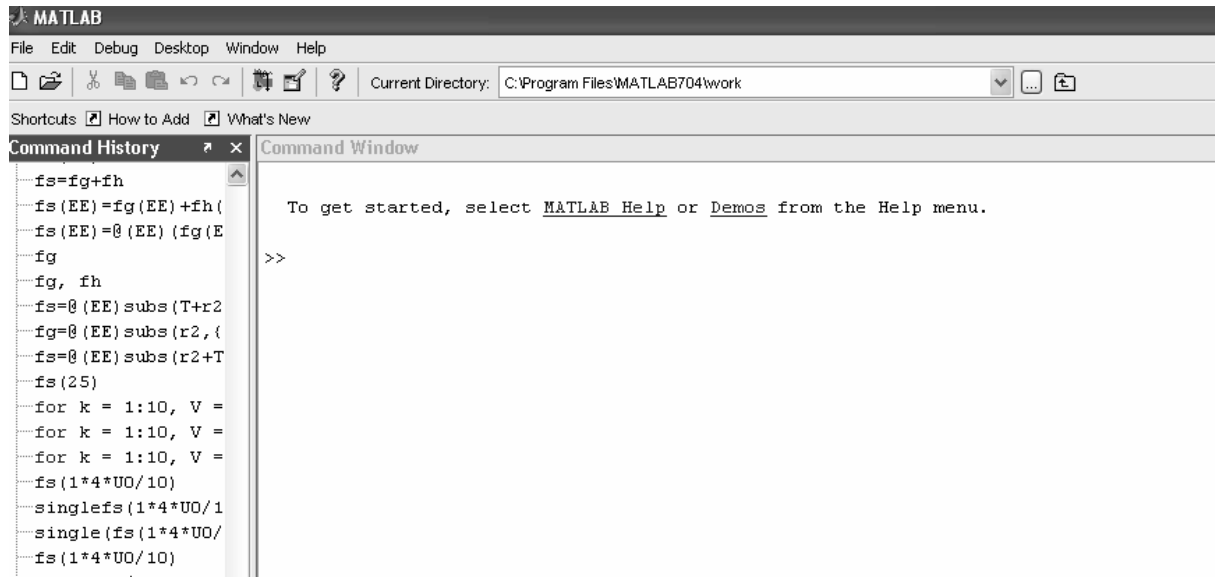


Рис. 2.1. Окно системы MATLAB после запуска

Например, на рисунке 2.1 изображено командное окно, которое появилось после запуска MATLAB в меню Пуск. Во время предыдущего использования были произведены простые действия, которые видны в левой части: переменным *a* и *b* были присвоены значения 2 и 3. Эти команды отобразились в Command History, но это не означает, что они выполнены, в чем легко убедиться, если попытаться использовать эти переменные:

```
>> a
??? Undefined function or variable 'a'.

>> a+b
??? Undefined function or variable 'a'.

>> b
??? Undefined function or variable 'b'.
```

Система не понимает, что мы от нее хотим. Загружен только список предыдущих команд, а сама Рабочая область (Workspace) – нет. Такая же

ситуация, кстати, имеет место и в Maple, там при загрузке файла загружается текст программы, все команды которого надо выполнить, чтобы присвоить переменным и функциям их значения. Так же можно поступить в MATLAB, если вызывать предыдущие команды (по одной) стрелкой курсора вверх-вниз и выполнять их (этот прием создатели MATLAB явно заимствовали из теперь уже почти забытой DOS):

```
>> a=2;b=3;
>> a, b
a =
    2
b =
    3.
```

Обратим внимание: в MATLAB также используется точка с запятой для разделения операторов, но совершенно иначе, можно даже сказать, с противоположным значением, чем в Maple. Если в последнем точка с запятой означает команду выполнить оператор и распечатать результат его выполнения, то в MATLAB наоборот – это команда НЕ печатать! При этом команда «печатать» в MATLAB – это отсутствие какого-либо знака после оператора. Но тогда возникает проблема разделения операторов, записанных в одной строке, что делается с помощью запятой, как продемонстрировано в вышеприведенном примере.

Такого рода различия между системами носят формально-синтаксический характер, и говорить о том, какой язык лучше и удобнее – сугубо дело вкуса и индивидуальных предпочтений. Но есть более глубокие различия между языками, которые носят, на наш взгляд, принципиальный характер. Вспомним, что до выполнения команды присваивания значений переменных a и b , система (MATLAB) вообще не допускала какого-либо использования этих символов, даже в ответ на простейшую команду « a » следовало «Undefined function or variable 'a'». А как поступит

в такой ситуации (т.е. до загрузки или предварительного выполнения каких-либо команд) Maple? Вот так:

```
> a; a+b; t:=a-b;
```

$$\begin{array}{c} a \\ a + b \\ t := a - b \end{array}$$

То есть в Maple любые символы по умолчанию предполагаются переменными (вообще говоря, комплексными), с которыми можно производить действия, определяя одни через другие. В MATLAB, как сразу же убеждается в этом любой, приступающий к его изучению, необходимо все описывать. Это, на наш взгляд, является неким «рудиментом», пришедшим из алгоритмических языков и делает язык MATLAB менее наглядным и удобным в применении, чем Maple. Можно сказать, что Maple намного более соответствует духу системы аналитических вычислений и компьютерной алгебры.

Продолжая первичное знакомство с MATLAB, отметим, что необязательно выполнять по одной все команды из файла (с расширением *.mat*), который вы хотите загрузить. Его можно загрузить с помощью команды «Open» из меню File (обычная процедура открытия файла в Windows). Интересно, что при выполнении этой процедуры вы не заметите никаких видимых изменений в командном окне. Изменения будут состоять в том, что в память загрузится рабочая область (Workspace), т.е. выполнятся все команды из загружаемого файла, в чем можно убедиться, выполняя какие-либо действия с переменными, которые там фигурируют.

В любой момент можно очистить Перечень команд, Командное окно и Рабочую область с помощью команд из меню Edit.

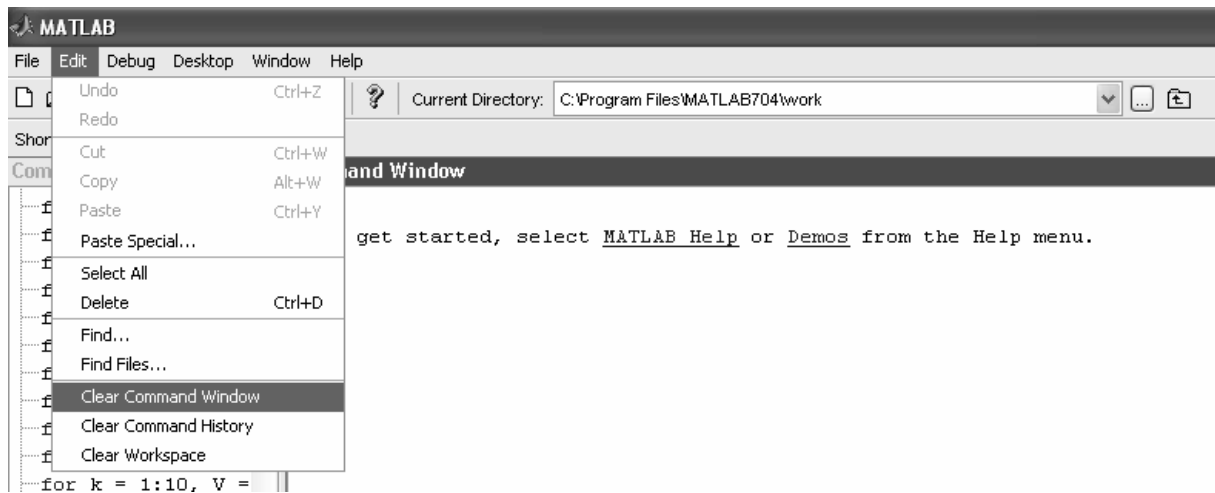


Рис. 2.2. Меню Edit в MATLAB

Обратим внимание, что выполнение команд «Clear Command Window» и «Clear Command History» не приводит к выгрузке рабочего файла из памяти и можно продолжать вычисления дальше. Лишь команда «Clear Workspace» очищает память и является аналогом команды «restart» в Maple.

Можно также загрузить рабочий файл, если запустить MATLAB, произведя двойной щелчок по значку этого файла. В этом случае в Command History сразу появится команда о загрузке файла (оператор load):

```
load('C:\DOCUME~1\admin\0016~1\matlab.mat').
```

Отсюда понятен синтаксис оператора load: в скобках и в одинарных кавычках «прописан путь» к вашему файлу. Такую команду можно напечатать и выполнить в рабочем окне и вручную, что также приведет к загрузке файла.

Читателю настоятельно рекомендуется проверить все вышеописанные процедуры на практике. А именно произвести следующие действия:

- запустите MATLAB из меню Пуск;
- очистите все с помощью команд «Clear Command Window», «Clear Command History» и «Clear Workspace» из меню Edit;

- произведите какие-либо действия в рабочем окне, например, вычисления, разобранные ниже в этой лекции и/или простейшие операции присваивания и арифметические действия;
- сохраните созданный вами файл с помощью команды «Save Workspace As» из меню File и закройте программу;
- снова запустите несколько раз MATLAB и загрузите ваш файл различными способами, описанными выше.

Для первичного знакомства с MATLAB, следуя принятому в данном пособии принципу производить одни и те же вычисления разными способами, повторим те простейшие операции, которыми иллюстрировалось изложение Maple в Теме 1. И здесь на первом же шаге мы сталкиваемся с тем принципиальным отличием от Maple, о котором говорилось выше:

```
>> a=2, b=3, f=sin(a*x)/x, g=cos(b*x)
a =
    2
b =
    3
??? Undefined function or variable 'x'.
```

Если оператор присваивания для a прошел без проблем, то попытка описать функции, даже в форме простых выражений, зависящих от переменной x , провалилась. Система не принимает переменную x , требуя ее описания. Ясно, что мы не можем продвинуться дальше, не обращаясь к Help.

Войдя в меню Help, выберем раздел MATLAB Help, в котором можно выбрать одну из вкладок представления справочного материала. Выберем вкладку Contents, которая представляет собой полное оглавление содержимого системы в виде «дерева». К нужному нам разделу об описании функций приходим по пути: MATLAB – Mathematics – Functions – Representing Functions in MATLAB. Попадаем на страницу, на которой изложено, как описывать функции.

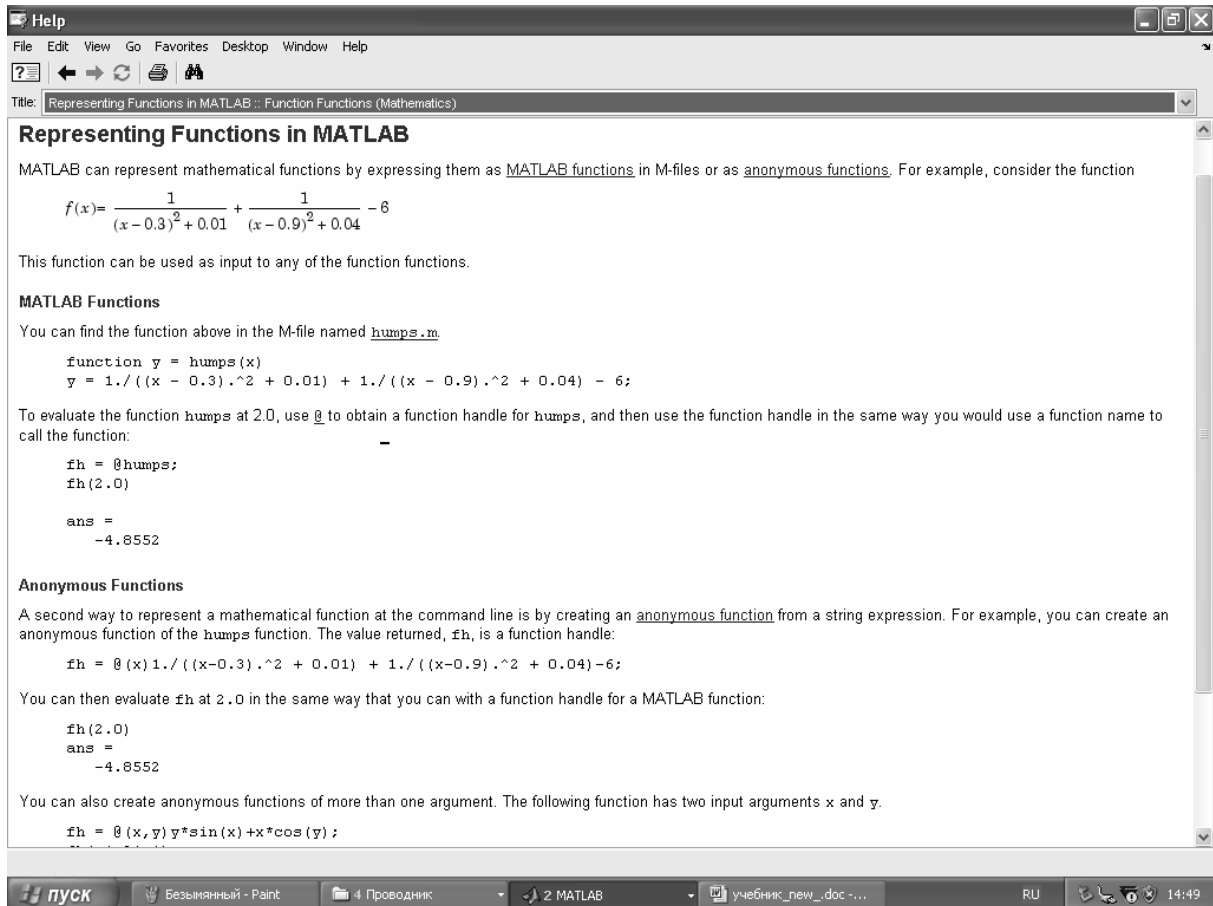


Рис. 2.3. Пользование справкой в MATLAB

Нам нужны в данном случае не функции, записанные в *m*-файлах (MATLAB Functions), а так называемые Anonymous Functions, под этим несколько странным названием фигурируют функции, задаваемые пользователем в ходе выполнения программы (в Maple это делается с помощью оператора $x \rightarrow$). Таким образом, наши функции можно задать так:

```
>> f=@(x) sin(a*x)/x, g=@(x) cos(b*x)
f =
    @(x) sin(a*x)/x
g =
    @(x) cos(b*x).
```

Теперь функции f и g существуют в нашей рабочей области, в чем можно убедиться, вызвав их значения при каких-либо значениях x :

```
>> f(2), g(3)
```

```
ans =  
-0.3784
```

```
ans =  
-0.9111
```

или построив график (выдается в отдельном окне)

```
>> fplot(f, [-5 5]).
```

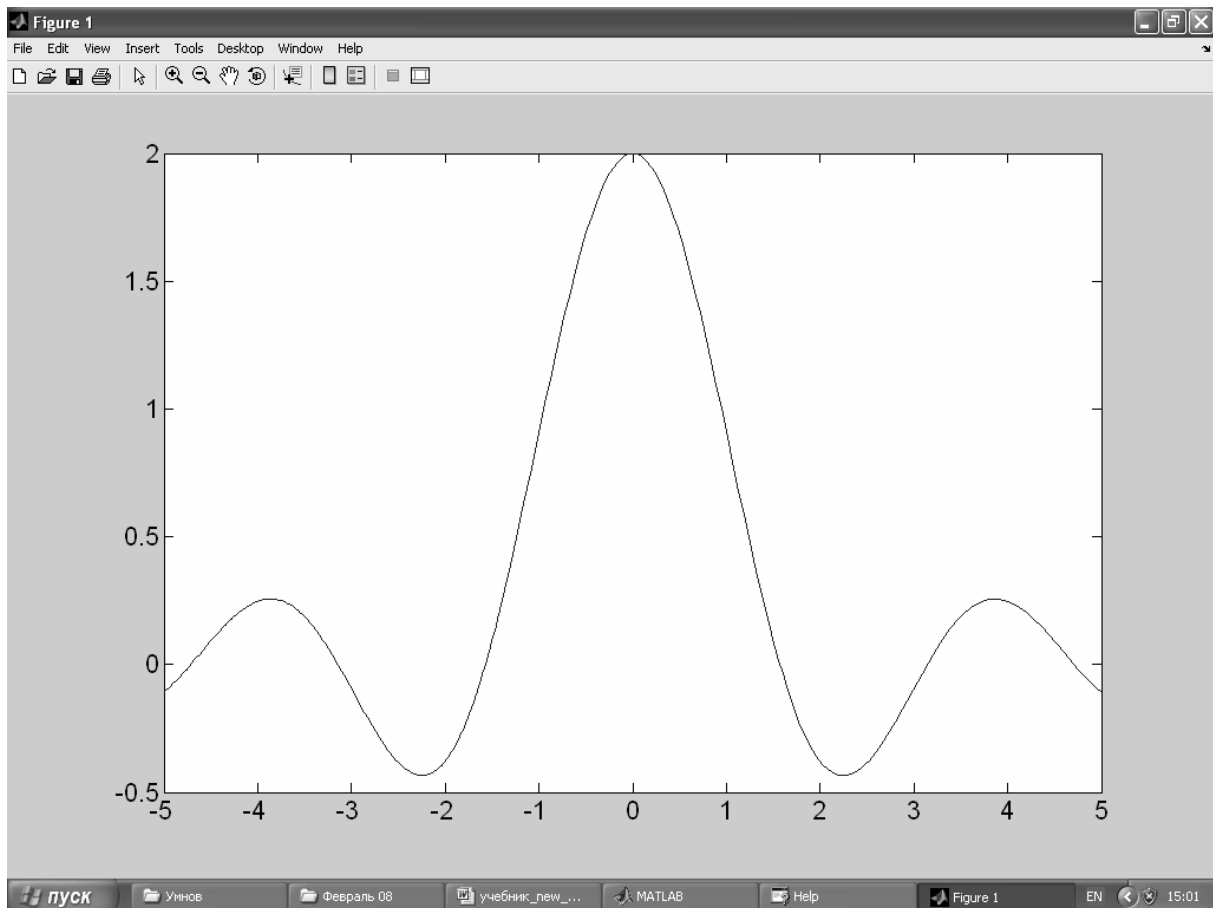


Рис. 2.4. Пример построения графика функции от одной переменной в MATLAB

Проведем также вычисление в MATLAB примера нахождения емкости трех последовательно включенных емкостей (конденсаторов) C_1 , C_2 и C_3 произвольной величины, разобранный в теме 1 в рамках Maple.

```
>> R=solve('1/c0=1/c1+1/c2+1/c3', 'c0')
```

```
R =
```

```

c1*c2*c3/(c2*c3+c1*c3+c1*c2)
>> c1=2; c2=1; c3=4;
>> subs(R)
ans =
    0.5714

```

Обратим внимание, что здесь оператор *subs*, примененный к имени *R*, под которым фигурирует решение уравнения, действует так же, как оператор *assign* в Maple.

Примеры на дифференцирование и интегрирование, разобранные в теме 1, в MATLAB выглядят так (перед началом вычислений необходимо описать переменные, которые будут входить в выражения, как символические, с помощью оператора *syms*):

```

>> syms x
>> diff(sin(x)^2+cos(x)^3,'x')
ans =
2*sin(x)*cos(x)-3*cos(x)^2*sin(x)
>> int(sin(x),x), int(sin(x),0,pi)
ans =
-cos(x)
ans =
2.

```

Интегралы, выражающиеся через специальные функции и «неберущиеся»:

```

>> q1=int(exp(-x^2),-1,1),q2=int(exp(-x^3),-1,1)
q1 =
erf(1)*pi^(1/2)
Warning: Explicit integral could not be found.
> In sym.int at 58
q2 =
int(exp(-x^3),x = -1 .. 1)
>> single(q1), single(q2)
ans =
    1.4936
ans =

```


2.1494.

(сравните с численными значениями, полученными в Теме 1 с помощью Maple). Обратим внимание, что система предупреждает о том, что интеграл q_2 не может быть вычислен в явном виде. Тем не менее, с помощью оператора *single* (который в данном случае действует так же, как *evalf* в Maple) он может быть найден численно (по поводу *single* и других операторов, имеющих отношение к типам данных, см. раздел Programming – Data Types в Help).

2.3. Решение задачи о туннелировании в MATLAB

Наконец, повторим в MATLAB пример расчета прохождения частицы через одномерный прямоугольный потенциальный барьер [2, с. 101-102], разобранный в Теме 1. Программа вместе с результатами выполнения операторов имеет следующий вид.

```
>> syms E positive; syms epsilon x m h U0;
>>ode='D2f1+(2*m/h^2)*E*f1=0,D2f2+(2*m/h^2)*(E-
U0)*f2=0,D2f3+(2*m/h^2)*E*f3=0'
ode =
D2f1+(2*m/h^2)*E*f1=0,D2f2+(2*m/h^2)*(E-
U0)*f2=0,D2f3+(2*m/h^2)*E*f3=0
>>ics='f1(1)=f2(1),Df1(1)=Df2(1),f2(0)=f3(0),Df2(0)=Df3(0),f1(
0)+i*f1(pi/2/sqrt(E))=2,f3(0)-i*f3(pi/2/sqrt(E))=0'
ics =
f1(1)=f2(1),Df1(1)=Df2(1),f2(0)=f3(0),Df2(0)=Df3(0),f1(0)+i*f1
(pi/2/sqrt(E))=2,f3(0)-i*f3(pi/2/sqrt(E))=0
>> [f1,f2,f3] = dsolve(ode, ics,'x');
>> S=subs(f3,{x,U0,m,h},{0,40,1/2,1});
>> T=S*conj(S);
>> fh=@(EE)subs(T,{E},{EE});
>> fh(38), fh(50)
ans =
0.0483 + 0.0000i
```

ans =

0.9997 - 0.0000i

Отметим, что здесь, в отличие от программы в Maple, система уравнений с граничными условиями сначала решена с общими значениями U_0, m, \hbar , а затем в них подставлены значения 40, 1/2 и 1 соответственно (разумеется, то же самое можно было бы сделать и в Maple). Громоздкая распечатка выражений для волновых функций на этот раз, конечно, опущена. В результате подстановки $E = 38$ и $E = 50$, для коэффициента прохождения T получили значения, совпавшие с результатом Maple. График $T(E)$ также имеет знакомый вид:

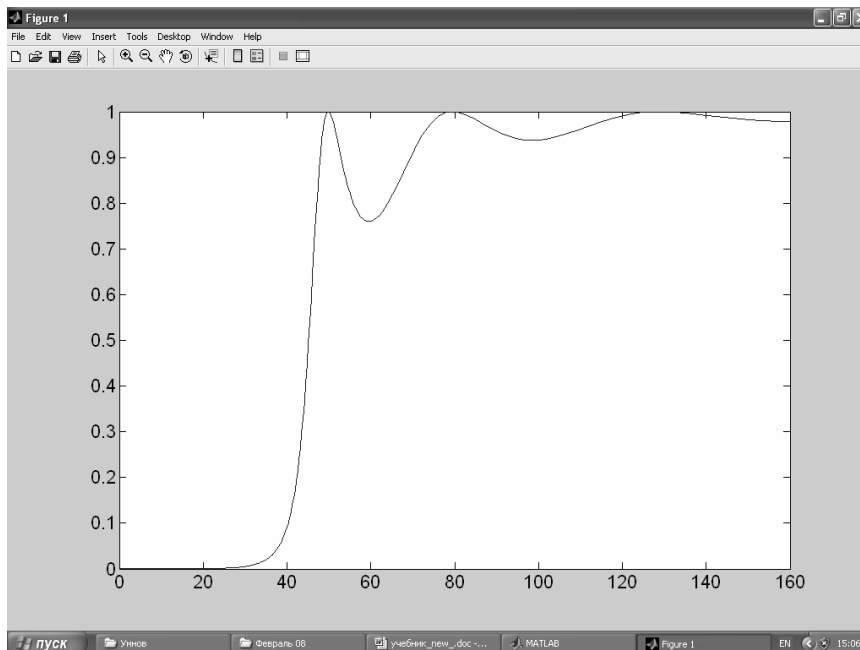


Рис. 2.5. Изменение коэффициента прохождения в зависимости от энергии для потенциального барьера, показанного на рисунке 1.4, полученное в MATLAB

Проверим также соотношение (1.11), для чего вычислим коэффициент отражения (обозначим, как и выше, через r_2). Функция fs представляет собой сумму коэффициентов прохождения и отражения. Для ознакомления с циклами в MATLAB, произведем проверку не путем построения графика,

а составим простейший цикл, выводя значения fs для пяти различных значений аргумента E :

```
>> r=subs(f1,{x,U0,m,h},{0,40,1/2,1})-1;  
>> r2=r*conj(r);  
>> fs=@(EE)subs(r2+T,{E},{EE});  
>> N=5; for k = 1:N, V = fs(k*4*40/N), end  
V =  
    1.0000  
V =  
    1.0000 + 0.0000i  
V =  
    1.0000 + 0.0000i  
V =  
    1.0000 - 0.0000i  
V =  
    1.0000 + 0.0000i.
```

Список литературы к теме 2

Использованная литература:

1. Алексеев Е.Р., Чеснокова О.В. МАТЛАБ 7. Самоучитель. – М.: Изд-во «ИТ Пресс», 2006 г. – 464 с.
2. Мессиа Альберт Квантовая механика. – М.: Наука, 1978. – 480 с.

Рекомендованная литература:

1. Мартынов Н.Н. МАТЛАБ 7. Элементарное введение. – М: Кудиц-Образ, 2005 г. – 416 с.
2. Поршнева С.В. МАТЛАБ 7. Основы работы и программирования. Учебник. – М.: Изд-во «Бином. Лаборатория знаний», 2006 г. – 320 с.
3. Плохотников К.Э., Волков Б.И., Задорожный С.С., Антонюк В.А./ Редактор К.Э. Плохотников. Методы разработки курсовых работ. Моделирование, вычисления, программирование на С++ и МАТЛАБ, виртуа-

лизация, образцы лучших студенческих курсовых работ. – М.: Изд-во «Солон», 2006. – 320 с.

Интернет-ресурсы к теме 1

1. Области применения систем компьютерной алгебры и конкретные примеры. <http://exponenta.ru/educat/systemat/systemat.asp>.
2. Руководство пользователя MATLAB.
http://exponenta.ru/soft/matlab/potemkin/book/matlab/chapter0/0_0.asp
3. Краткое описание функционала MATLAB.
<http://itc.ua/article.phtml?ID=21405&IDw=29&pid=18>
4. Что такое MATLAB на Wikipedia
<http://ru.wikipedia.org/wiki/MATLAB>.
5. Применение MATLAB. <http://matlab.exponenta.ru/mltb/default.php>.

Контрольные вопросы к теме 2

1. Кем разработана система MATLAB?
2. Что такое формат Notebook?
3. Что исполняют загрузочный файл matlabrc.m и файл startup.m?
4. Какие можете назвать toolbox в MATLAB?
5. Какие языки и каких уровней использованы в MATLAB?
6. В чем принципиальные отличия работы в интерфейсах Maple и MATLAB?
7. Какими способами можно задавать функции в MATLAB?
8. Постройте графики однополостного и двуполостного гиперboloидов в MATLAB.
9. Напишите программу в MATLAB для решения какой-либо конкретной электрической цепи с помощью правил Кирхгофа.
10. Напишите программу в MATLAB для решения квантовой задачи для частицы в одномерном потенциальном ящике.

11. Решите в MATLAB какое-либо уравнение Бесселя и постройте график его решений для различных значений параметров.
12. Каково назначение символа @ в MATLAB?

Тема 3. Пакет Mathematica. Сравнительный анализ трех численно-аналитических пакетов Maple, MATLAB и Mathematica

3.1. Историческая справка и общие сведения о 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 [1, с. 4]. 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. В разработках систем Mathematica, наряду с головной фирмой Wolfram Research, Inc., принимали участие ряд других фирм и сотни специалистов высокой квалификации. Система Mathematica всегда рассматривалась как мировой лидер среди компьютерных систем символьной математики для ПК, обеспечивающих не только возможности выполнения сложных численных расчетов с выводом их результатов в самом изысканном графическом виде, но и проведение особо трудоемких аналитических вычислений и преобразований. Версии системы имеют современный пользовательский интерфейс и позволяют готовить документы в форме Notebooks («записных книжек»). Они объединяют исходные данные, описание алгоритмов решения задач, программ и результатов решения в самой разнообразной форме (математические формулы, числа, векторы, матрицы, графики) [1].

С самого начала большое внимание уделялось графике, в том числе динамической, и даже возможностям мультимедиа – воспроизведению динамических изображений и синтезу звуков с поддержкой звуковой платы (аудиоадаптера). Набор функций графики и меняющих их действие опций и директив весьма разнообразен. Графика всегда была отличительной чертой систем Mathematica и обеспечивала им лидерство среди систем компьютерной математики.

3.2. Начальное знакомство с системой Mathematica

Ознакомимся с работой системы на практике. Запустим программу из меню Пуск. Автоматически откроется новое рабочее поле в виде автономного окна (в Mathematica оно называется Notebook – блокнот). Можно открыть второе и большее количество таких окон. На рисунке 3.1 открыто два окна, и в них проведены простейшие вычисления.

Самое главное, что надо знать о Mathematica в отношении сравнения ее с другими системами компьютерной алгебры: ее язык внешне достаточно сильно отличается от языка Maple, но по внутреннему «устройству» и по приемам работы с интерфейсом весьма похож на него. Так что пользователь, владеющий, допустим, навыками работы с Maple, не испытает принципиальных неудобств при переходе на Mathematica, в отличие от MATLAB, который потребует от него значительной перестройки. А именно: в Mathematica, так же как в Maple, вы набираете, оператор в командной строке и для его выполнения нажимаете Shift–Enter (отличие от Maple и MATLAB, где нажимается просто Enter – сравнение, конечно, не в пользу Mathematica, но при длительном пользовании это входит в привычку).



Рис. 3.1. Окно системы Mathematica после запуска, открытия второго окна и выполнения простейших вычислений

А далее, в отличие от MATLAB и подобно Maple, если вы захотите повторить какое-то вычисление, то можно установить курсор вставки в соответствующей строке щелчком и нажать Shift–Enter. Это же можно сделать иначе: установите I-образный курсор на квадратную скобку справа от формулы (курсор при этом изменит свой вид) и щелкните один раз. Скобка «почернеет». Тем самым выделена ячейка, содержащая нужную формулу. Теперь достаточно нажать Shift – Enter. Нужное вычисление будет выполнено. При желании выделенные ячейки можно копировать и размножать обычными для систем с графическим интерфейсом приемами (кнопками или меню). Повторим, все перечисленное является совершенно естественным для пользователя Maple и составляет одно из основных неудобств при его переходе на MATLAB. А вот что очень непривычно для тех, кто привык иметь дело с Maple и является одним из проявлений различий в языках: а) аргументы функций заключаются в квадратные скобки; б) имена функций, встроенных в систему Mathematica, начинаются с заглавных букв.

Как обычно в Windows, для сохранения из меню File выбирается пункт Save As и записывается протокол проведенных расчетов в файл (для Mathematica 5 с расширением .nb – от слова Notebook).

Как видно из рисунка 3.1, арифметические действия в системе Mathematica изображаются обычным образом: +, -, *, ^, но вместо знака умножения можно набирать пробел. Подобно MATLAB и в отличие от Maple, точка с запятой означает команду выполнить оператор и распечатать результат его выполнения, а команда «печатать» – это отсутствие какого-либо знака после оператора.

Внешне отличаются, а по сути применяются точно так же, многие операторы Maple, например *evalf*: в Mathematica это просто буква *N*:

```
N[Pi/2]
```

```
1.5708
```

```
N[33/7-Sqrt[2]]
```

```
3.30007.
```

Или, например, оператор подстановки (*subs* в Maple) имеет весьма непривычный и громоздкий вид:

```
r/.{x→w,y→u}
```

```
u+w.
```

Из следующей строчки проясняется синтаксис этого оператора:

```
1+x^2+x^4/.x→2
```

```
21.
```

Интересно, что некоторые операторы имеют такой же вид, что и в Maple, но в их действиях есть различия. Например, это относится к *simplify*: в обеих системах это оператор, упрощающий выражения, но, видимо, понятие «упрощение» создатели двух систем понимают несколько по-разному. Так, в Mathematica разложим степень разности на отдельные слагаемые с помощью оператора *Expand*, а затем применим к результату *Simplify*:

```
q:=Expand[(x-y)^3]
```

$$x^3 - 3x^2y + 3xy^2 - y^3$$

```
Simplify[q]
```

$$(x-y)^3$$

А вот что в такой ситуации сделает Maple:

```
> q:=expand((x-y)^3);
```

$$q := x^3 - 3x^2y + 3xy^2 - y^3$$

```
> simplify(q);
```

$$x^3 - 3x^2y + 3xy^2 - y^3$$

Т.е. *expand* «сработал» так же, а упрощение в Maple понимается как разложение, а в Mathematica – наоборот, как представление в свернутом виде. Кстати, в Mathematica, помимо обычного *Simplify*, есть еще «более мощный» оператор *FullSimplify*. Например, в нижеследующей ситуации *Simplify* не справляется, а его усиленный аналог – да:

```
Simplify[Gamma[z] Gamma[1-z]]
```

$$\text{Gamma}[1-z] \text{Gamma}[z]$$

```
FullSimplify[Gamma[z] Gamma[1-z]]
```

$$\pi \text{Csc}[\pi z].$$

Впрочем, в Maple с этой задачей упрощения произведения гамма-функций справится и обычный оператор:

```
> q:=GAMMA(z)*GAMMA(1-z);
```

$$q := \Gamma(z) \Gamma(1-z)$$

```
> simplify(q);
```

$$\frac{\pi}{\sin(\pi z)}$$

Для первичного знакомства с системой, читателю настоятельно рекомендуется проработать все пункты превосходно написанного Tutorial из меню Help в Mathematica. Как неоднократно указывалось, подробное изложение языков и структуры систем выходит за рамки данного пособия. Ограничимся тем, что разберем примеры, решенные выше в Maple и MATLAB.

Функции описываются с помощью знака :=

```
a=2; f[x_] := Sin[a*x]/x; g[y_] := Sin[b*y]
```

```
f[1]
```

```
Sin[2]
```

```
g[3]
```

```
Sin[3 b].
```

Причем $x_$, $y_$ означают произвольные переменные под именами x , y . Численные значения выражений и подстановка параметров:

```
N[f[1]]
```

```
-0.378401
```

```
N[g[3]/.b→Pi/9]
```

```
0.909297.
```

Построение графиков описанных функций:

```
Plot[{f[x], g[x]/.b→2}, {x, -15, 15}]
```

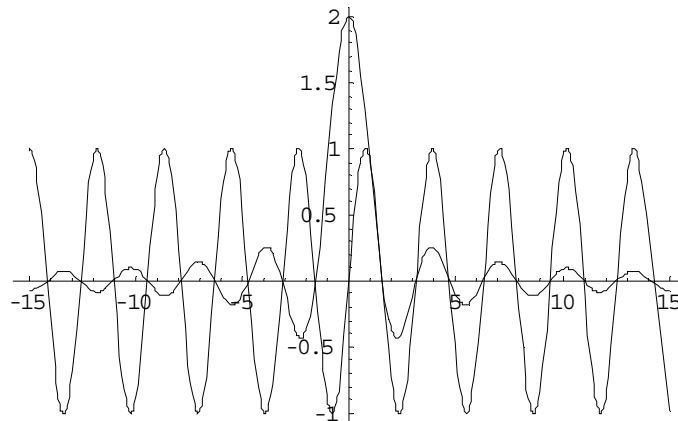


Рис. 3.2. Пример построения графика функций в Mathematica

Построим график функции $\sin(x+y)$:

```
Plot3D[Sin[x+y], {x, -2, 2}, {y, -2, 2}].
```

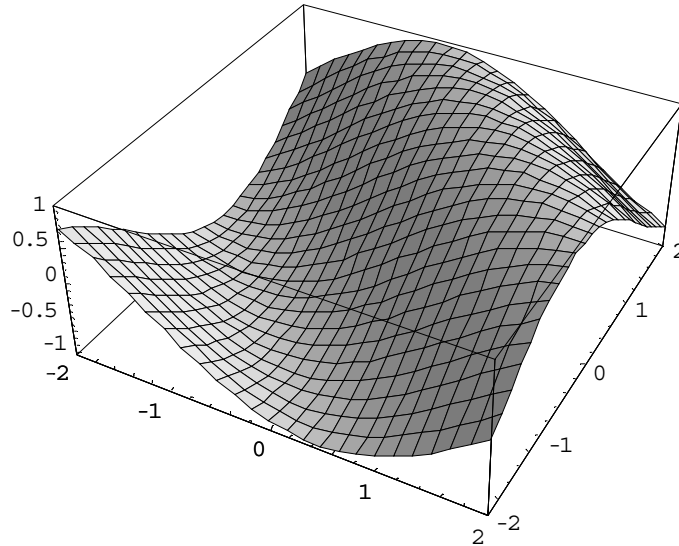


Рис. 3.3. Пример построения графика функции от двух переменных

Решим задачу (1.1) о нахождении емкости трех последовательно включенных емкостей (конденсаторов) C_1 , C_2 и C_3 произвольной величины, разобранный в Maple и MATLAB.

`eq=1/C0==1/C1+1/C2+1/C3;r=Solve[eq,C0]`

$$\left\{ \left\{ C0 \rightarrow \frac{C1 C2 C3}{C1 C2 + C1 C3 + C2 C3} \right\} \right\}$$

`r/.{C1→2,C2→1,C3→4}`

$$\left\{ \left\{ C0 \rightarrow \frac{4}{7} \right\} \right\}$$

`N[r/.{C1→2,C2→1,C3→4}]`

$$\left\{ \left\{ C0 \rightarrow 0.571429 \right\} \right\}$$

Операцию присваивания полученных в процедуре решения функции (в данном случае одной – $C0$) можно провести следующим образом:

`C0=C0/.r`

$$\left\{ \frac{C1 C2 C3}{C1 C2 + C1 C3 + C2 C3} \right\}$$

`s=C0/.{C1→2,C2→1,C3→4}`

$$\left\{ \frac{4}{7} \right\}$$

Видно, что теперь C_0 определена как функция трех переменных. Результат подстановки в нее значений емкостей 2, 1 и 4 обозначен как s . С ним можно производить действия:

N[s]+1

{1.57143}

Вычисление производных и интегралов:

D[Sin[x]^2+Cos[x]^3,x]

2 Cos[x] Sin[x] - 3 Cos[x]^2 Sin[x]

Integrate[Sin[x],x]

-Cos[x]

Integrate[Sin[x],{x,0,Pi}]

2.

Неберущиеся интегралы в численном виде:

q1=Integrate[Exp[-x^2],{x,-1,1}]

$\sqrt{\pi}$ Erf[1]

q2=Integrate[Exp[-x^3],{x,-1,1}]

$-\frac{1}{3} \text{ExpIntegralE}\left[\frac{2}{3}, -1\right] - (-1 + (-1)^{2/3}) \text{Gamma}\left[\frac{4}{3}\right] - \frac{1}{3} \text{Gamma}\left[\frac{1}{3}, 1\right]$

N[q1]

1.49365

N[q2]

2.14942 - 1.11022 Γ 10^{-16} д

Интересно отметить, что в случае интеграла $q2$, в отличие от Maple и MATLAB, система Mathematica пыталась как-то вычислить его, преобразовав к сумме гамма-функций и интегральной экспоненты.

3.3. Решение задачи о туннелировании

Наконец, следуя нашему принципу обучения посредством решения одних и тех же задач разными способами, повторим в Mathematica пример расчета прохождения частицы через одномерный прямоугольный потенци-

альный барьер [2, с. 101–102], разобранный в Теме 1. Программа имеет следующий вид (вместо E используем V).

```
u0=40;m=1/2;h=1;
```

```
(V > 0) True
```

```
(V < u0) True
```

Обратим внимание на формат оператора, присваивающего свойства переменным – аналог оператора *assume* в Maple. Далее описываем величину ε и уравнения:

```
epsilon=(2*m/h^2)*V;
```

```
eq1=f1''[x]+(2 m/h^2) V f1[x] 0
```

```
Vf1[x] + f1''[x] == 0
```

```
eq2=f2''[x]+(2 m/h^2) (EE-u0) f2[x] 0
```

```
(-40 + V) f2[x] + f2''[x] == 0
```

```
eq3=f3''[x]+(2 m/h^2) EE f3[x] 0
```

```
Vf3[x] + f3''[x] == 0
```

Очень важное наблюдение: в Mathematica есть три вида «знаков равенства»: обычный знак равенства $=$, двоеточие со знаком равенства $:=$ и двойной знак равенства $==$. Первый является оператором присваивания для символов (переменных), второй – оператором присваивания для функций, а третий является просто знаком равенства в соотношениях и уравнениях (т.е. когда он непосредственно ничего не присваивает). В Maple имеется лишь два таких знака, причем $:=$ – это оператор присваивания и для символов, и для уравнений а $=$ – это знак равенства в уравнениях.

```
r=DSolve[{eq1,eq2,eq3,f1[1] f2[1],f1'[1] f2'[1],f2[0] f3[0],f2'[0] f3'[0],f1[0]+I*f1[Pi/2/Sqrt[epsilon]]==2,f3[0]-I*f3[Pi/2/Sqrt[epsilon]]==0},{f1[x],f2[x],f3[x]},x]
```

{{f1[x]→

$$\begin{aligned} & (\sqrt{40-v} \sqrt{v} \operatorname{Cos}[\sqrt{v}] \operatorname{Cos}[\sqrt{v} x] + e^{2\sqrt{40-v}} \sqrt{40-v} \sqrt{v} \operatorname{Cos}[\sqrt{v}] \operatorname{Cos}[\sqrt{v} x] + i v \operatorname{Cos}[\sqrt{v}] \operatorname{Cos}[\sqrt{v} x] - i e^{2\sqrt{40-v}} v \operatorname{Cos}[\sqrt{v}] \operatorname{Cos}[\sqrt{v} x] + \\ & -40 \operatorname{Cos}[\sqrt{v} x] \operatorname{Sin}[\sqrt{v}] - 40 e^{2\sqrt{40-v}} \operatorname{Cos}[\sqrt{v} x] \operatorname{Sin}[\sqrt{v}] + i \sqrt{40-v} \sqrt{v} \operatorname{Cos}[\sqrt{v} x] \operatorname{Sin}[\sqrt{v}] + i e^{2\sqrt{40-v}} \sqrt{40-v} \sqrt{v} \operatorname{Cos}[\sqrt{v} x] \operatorname{Sin}[\sqrt{v}] - \\ & v \operatorname{Cos}[\sqrt{v} x] \operatorname{Sin}[\sqrt{v}] + e^{2\sqrt{40-v}} v \operatorname{Cos}[\sqrt{v} x] \operatorname{Sin}[\sqrt{v}] - 40 \operatorname{Cos}[\sqrt{v}] \operatorname{Sin}[\sqrt{v} x] + 40 e^{2\sqrt{40-v}} \operatorname{Cos}[\sqrt{v}] \operatorname{Sin}[\sqrt{v} x] - \\ & i \sqrt{40-v} \sqrt{v} \operatorname{Cos}[\sqrt{v}] \operatorname{Sin}[\sqrt{v} x] - i e^{2\sqrt{40-v}} \sqrt{40-v} \sqrt{v} \operatorname{Cos}[\sqrt{v}] \operatorname{Sin}[\sqrt{v} x] + v \operatorname{Cos}[\sqrt{v}] \operatorname{Sin}[\sqrt{v} x] - e^{2\sqrt{40-v}} v \operatorname{Cos}[\sqrt{v}] \operatorname{Sin}[\sqrt{v} x] + \\ & \sqrt{40-v} \sqrt{v} \operatorname{Sin}[\sqrt{v}] \operatorname{Sin}[\sqrt{v} x] + e^{2\sqrt{40-v}} \sqrt{40-v} \sqrt{v} \operatorname{Sin}[\sqrt{v}] \operatorname{Sin}[\sqrt{v} x] + i v \operatorname{Sin}[\sqrt{v}] \operatorname{Sin}[\sqrt{v} x] - i e^{2\sqrt{40-v}} v \operatorname{Sin}[\sqrt{v}] \operatorname{Sin}[\sqrt{v} x]) / \\ & ((-20 i + 20 i e^{2\sqrt{40-v}} + \sqrt{40-v} \sqrt{v} + e^{2\sqrt{40-v}} \sqrt{40-v} \sqrt{v} + i v - i e^{2\sqrt{40-v}} v) (\operatorname{Cos}[\sqrt{v}] + i \operatorname{Sin}[\sqrt{v}])) \end{aligned}$$

$$f2[x] \rightarrow \frac{e^{\sqrt{40-v}-\sqrt{40-v}x} \sqrt{v} (40 i + 40 i e^{2\sqrt{40-v}x} - \sqrt{40-v} \sqrt{v} + e^{2\sqrt{40-v}x} \sqrt{40-v} \sqrt{v} - i v - i e^{2\sqrt{40-v}x} v) (\operatorname{Cos}[\sqrt{v}] - i \operatorname{Sin}[\sqrt{v}])}{\sqrt{40-v} (20 - 20 e^{2\sqrt{40-v}} + i \sqrt{40-v} \sqrt{v} + i e^{2\sqrt{40-v}} \sqrt{40-v} \sqrt{v} - v + e^{2\sqrt{40-v}} v)}$$

$$f3[x] \rightarrow \frac{2 e^{\sqrt{40-v}} (-40+v) \sqrt{v} (\operatorname{Cos}[\sqrt{v}] - i \operatorname{Sin}[\sqrt{v}]) (\operatorname{Cos}[\sqrt{v} x] - i \operatorname{Sin}[\sqrt{v} x])}{\sqrt{40-v} (-20 i + 20 i e^{2\sqrt{40-v}} + \sqrt{40-v} \sqrt{v} + e^{2\sqrt{40-v}} \sqrt{40-v} \sqrt{v} + i v - i e^{2\sqrt{40-v}} v)}$$

Нельзя не обратить внимания, насколько компактнее, по сравнению с Maple, система Mathematica выдала решение! А ведь это то же самое решение, в чем мы сейчас убедимся. Это является проявлением уже упоминавшейся несколько большей оптимальности расчетов в Mathematica.

Выполняем операторы присваивания:

$$f1[x] = f1[x] /. r; f2[x] = f2[x] /. r; f3[x] = f3[x] /. r;$$

(структура операторов получилась несколько странная с точки зрения логики, но в каждой системе иногда надо подстраиваться под общее строение и синтаксис языка).

Вычисляем коэффициент прохождения (напомним, имея в виду (1.10), можно получить амплитуду прошедшей сквозь барьер волны S , просто подставив $x=0$ в $\Phi_3(x)$ и далее вычисляя коэффициент прохождения как квадрат модуля S , умножая S на его комплексно сопряженное (оператор *conjugate*):

$$S = f3[x] /. x \rightarrow 0$$

$$\left\{ -\frac{2 e^{\sqrt{40-V}} (-40+V) \sqrt{V} (\text{Cos}[\sqrt{V}] - i \text{Sin}[\sqrt{V}])}{\sqrt{40-V} \left(-20 i + 20 i e^{2\sqrt{40-V}} + \sqrt{40-V} \sqrt{V} + e^{2\sqrt{40-V}} \sqrt{40-V} \sqrt{V} + i V - i e^{2\sqrt{40-V}} V \right)} \right\}$$

`T=N[S Conjugate[S]];`

Проверим, получили ли мы тот же результат, что и раньше, для чего вычислим коэффициент прохождения для энергии $V=38$:

`T/.V→38`

`{0.048291- 4.67843×10-18 i}`

С помощью оператора *Plot* получаем хорошо знакомый график для коэффициента прохождения в зависимости от энергии

`Plot[T/.V→Eх,{Eх,0,4*у0}]`

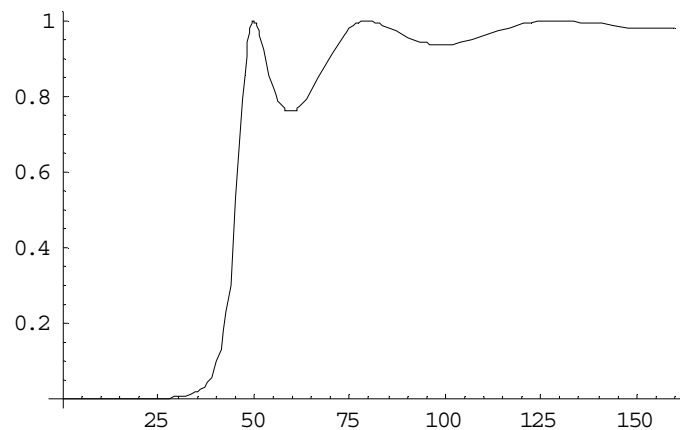


Рис. 3.4. Изменение коэффициента прохождения в зависимости от энергии для потенциального барьера, показанного на рисунке 1.4, полученное в Mathematica

Наконец, проверим еще раз соотношение (1.11), для чего вычислим коэффициент отражения (обозначим, как и выше, через r_2). Функция ft представляет собой сумму коэффициентов прохождения и отражения. Так же как это было сделано в MATLAB, для ознакомления с циклами в Mathematica, произведем проверку не путем построения графика, а составим

простейший цикл, выводя значения ft для пяти различных значений аргумента E :

```
r=(f1[x]-1)/.x→0; r2=N[r Conjugate[r]];
ft[q_]:=N[(T+r2)/.V→q];
nn=5;For[k=1,k<=nn,Print[ft[k*4*u0/nn]];k++]
{1. +0. 7M}
{1. + 2.77556 × 10-17 i}
{1. - 6.7987 × 10-17 i}
{1. - 3.92523 × 10-17 i}
{1. - 2.19427 × 10-17 i}
```

3.4. Сопряжение систем компьютерной алгебры

В последние годы разработчики математических систем уделяют огромное внимание их интеграции и совместному использованию. Это не только расширяет класс решаемых каждой системой задач, но и позволяет подобрать для них самые лучшие и наиболее подходящие инструментальные средства. Решение сложных математических задач сразу на нескольких системах существенно повышает вероятность получения корректных результатов, так как математики, как и математические системы, способны ошибаться, особенно при некорректной постановке задач неопытными пользователями.

С системой MATLAB могут интегрироваться такие популярные математические системы, как Mathcad, Maple V и Mathematica. Есть тенденция и к объединению математических систем с современными текстовыми процессорами. Так, новое средство одних из последних версий MATLAB – Notebook позволяет готовить документы в текстовом процессоре Word со вставками в виде документов MATLAB и результатов вычислений, представленных в численном, табличном или графическом виде. Таким образом, становится возможной подготовка «живых» электронных книг, в которых демонстрируемые примеры могут быть оперативно изменены. Так,

вы можете менять условия задач и тут же наблюдать изменение результатов их решения. В версиях MATLAB 6 и выше предусмотрены также улучшенные средства для экспорта графики в слайды презентационной программы Microsoft PowerPoint.

В MATLAB задачи расширения системы решаются с помощью специализированных пакетов расширения – наборов инструментов (Toolbox). Многие из них содержат специальные средства для интеграции с другими программами, поддержки объектно-ориентированного и визуального программирования, для генерации различных приложений. В состав системы MATLAB входит ядро системы Maple. Оно используется пакетами расширения Symbolic Math Toolbox и Extended Symbolic Math Toolbox, благодаря которым в среде MATLAB стали доступны принципиально новые возможности символьных и аналитических вычислений. Новые свойства системе MATLAB придала ее интеграция с программной системой Simulink, созданной для моделирования динамических систем и устройств, заданных в виде системы блоков. Базируясь на принципах визуально-ориентированного программирования, Simulink позволяет выполнять моделирование сложных устройств с высокой степенью достоверности и с прекрасными средствами представления результатов. Помимо естественной интеграции с пакетами расширения Symbolic Math и Simulink, MATLAB интегрируется с десятками мощных пакетов расширения. В свою очередь, многие другие математические системы, например Mathcad и Maple, допускают установление объектных и динамических связей с системой MATLAB, что позволяет использовать в них эффективные средства MATLAB для работы с матрицами.

Список литературы к теме 3

Использованная литература:

1. Шмидский Я.К. Mathematica 5. Самоучитель. – М.: Диалектика, 2004.

2. Мессиа Альберт Квантовая механика.– М.: Наука, 1978. – 480 с.

Рекомендованная литература:

1. Эдвардс Ч.Г., Пенни Д.Э. Дифференциальные уравнения и краевые задачи: моделирование и вычисление с помощью Mathematica, Maple и MATLAB. 3-е издание. – Киев: Диалектика-Вильямс, 2007.
2. Дьяконов В.П. Mathematica 4.1/4.2/5.0 в математических и научно-технических расчетах. – М.: СОЛОН-Пресс, 2004.

Интернет-ресурсы к теме 3

1. Кулешов А.А., Земсков С В., Позняк Ю.В. Электронное учебное пособие по высшей математике на базе системы Mathematica.
2. Научный форум. Что лучше, MATLAB, MathCad или Maple?
<http://www.lib.mexmat.ru/forum/viewtopic.php?t=6803&postdays=0&postorder=asc&start=0&sid=3f5a631f553046a6a49d431ef10c64d2>.
3. Области применения систем компьютерной алгебры и конкретные примеры. <http://exponenta.ru/educat/systemat/systemat.asp>.

Контрольные вопросы к теме 3

1. Кем разработана система Mathematica?
2. Чем отличаются форматы Notebook в Mathematica и MATLAB?
3. Что всегда было отличительной чертой систем Mathematica и обеспечивало им лидерство среди систем компьютерной математики?
4. Какие виды знаков равенства используются в Mathematica?
5. Каков синтаксис операторов присваивания в Mathematica?
6. В чем принципиальные отличия и сходства языков и систем Maple, MATLAB и Mathematica?
7. Какими способами можно задавать функции в Mathematica?

8. Постройте графики однополостного и двуполостного гиперboloидов в Mathematica.
9. Напишите программу в Mathematica для решения какой-либо конкретной электрической цепи с помощью правил Кирхгофа.
10. Напишите программу в Mathematica для решения квантовой задачи для частицы в одномерном потенциальном ящике.
11. Решите в Mathematica какое-либо уравнение Бесселя и постройте график его решений для различных значений параметров.

Тема 4. Примеры вычислений и моделирования систем с помощью численно-аналитических пакетов программ

4.1. Имитационное моделирование методом Монте-Карло модельной системы броуновского движения: случайные блуждания на прямой

Броуновское движение является типичным примером так называемой стохастической системы. Это явление было открыто в 1827 году английским ботаником Брауном (1773–1858). Это движение тем оживленнее, чем выше температура и чем меньше вязкость среды. Его едва удастся обнаружить в глицерине, а в газах оно, напротив, чрезвычайно интенсивно. Оно представляет собой хаотичное движение достаточно крупной (по сравнению с молекулами и видимой под микроскопом) частицы, находящейся в жидкой среде, происходящее под действием случайных ударов по ней со стороны молекул. Последние находятся в состоянии теплового движения с характерной скоростью $v_i \approx \sqrt{3k_B T / m}$, где k_B – постоянная Больцмана, T – температура среды в абсолютной шкале, m – масса молекулы. Направление движения молекулы и момент ее столкновения с броуновской частицей являются случайными величинами. Иными словами, в каждый момент времени неизвестно, столкнется ли с ней молекула, а если столкнется, то неизвестно, с какой стороны она «ударит». Вообще говоря, неизвестно и точное значение величины скорости молекулы. Тем не менее, можно оценить параметры вероятностных распределений этих величин. Например, в отношении направления скорости молекулы чаще всего обоснованным является предположение о равновероятном (равномерном) распределении. Частота столкновения молекул с частицей зависит от их концентрации (т.е. от плотности среды), от их скоростей и от геометрического размера частицы. Что касается величины скорости, то хорошим приближением часто является распределение Максвелла, т.е. нормальное распре-

деление с дисперсией, определяемой температурой среды. Но можно и предположить, что все скорости одинаковы и равны v_i .

Рассмотрим предельно простую модель, когда частица может двигаться только вдоль прямой, и она испытывает столкновения со стороны молекул справа или слева. Можно возразить, что эта одномерная модель слишком примитивна и имеет мало отношения к действительности. Однако не надо забывать, что нашей целью является моделирование, которое включает в себя собственно составление модели и реализацию ее в виде компьютерной программы. Следовательно, если вы составите сразу предельно реалистичную, а, следовательно, сложную модель, то вам придется сразу составлять и сложную программу. На этом двухступенчатом пути в таком случае вы с большой вероятностью совершите ошибку, или же столкнетесь с проблемой, которую не сможете решить или потратите неоправданно много времени на ее решение. Поэтому правильным подходом является такой: сначала составляете простую, пусть даже далекую от реальности, модель. Программная реализация этой модели будет, тем не менее, содержать достаточно много из того, что сохранится и в дальнейшем. Далее надо определить, в каких аспектах ваша простая модель далека от реальности, и составить другую, более реалистичную модель. Вам будет неизмеримо проще внести изменения в программу, чтобы приспособить ее к новой модели, чем сразу «с нуля» составлять программу для сложной модели.

Более того, часто оказывается, что выводы, сделанные на основе простой модели, сами по себе оказываются весьма реалистичными и объясняют сущность явления.

Итак, упрощенная одномерная модель броуновского движения, или модель случайных блужданий вдоль прямой, включает в себя следующее. Частица (материальная точка) массы M испытывает столкновения со стороны молекул (других материальных точек) массы m (ясно, что $m \ll M$). В

каждый момент времени столкновение может произойти с вероятностью p , причем если столкновение произойдет, то с равной вероятностью оно будет справа или слева. Скорость молекулы равна u . Столкновения молекул с частицей упругие. При отсутствии столкновений частица свободно движется вдоль прямой, т.е. на нее не действуют никакие силы. Нас интересует координата частицы в зависимости от времени и вычисление среднего по времени отклонения частицы от начала координат.

В принципе, в вышеприведенном абзаце приведено полное описание модели. Однако прежде чем составлять программу, необходимо выполнить еще один этап: разобраться в физических процессах, которые протекают в модели, написать соответствующие формулы и уравнения и продумать, как эти процессы будут воплощены в программе. Важно понимать, что ни один самый совершенный программный пакет на это не способен, это под силу только живому разуму.

В данном случае основной физической процесс, определяющий поведение нашей модели, представляет собой упругое столкновение двух тел. Масса первого тела M , его скорость до столкновения обозначим V , масса второго тела m , скорость до столкновения u . После столкновения скорости \tilde{V} и \tilde{u} (в программе будем обозначать Vt и ut). Все эти скорости могут иметь положительные и отрицательные значения, причем положительное значение соответствует движению слева направо вдоль оси, а отрицательное – справа налево. Необходимо найти скорости после столкновения, т.е. выразить их через скорости до него. Из механики известно, что это проще сделать, используя законы сохранения энергии и импульса:

$$\begin{aligned}
 M V^2 + m u^2 &= M \tilde{V}^2 + m \tilde{u}^2; \\
 M V + m u &= M \tilde{V} + m \tilde{u} .
 \end{aligned}
 \tag{4.1}$$

Уравнения (4.1) простые и могут быть решены относительно \tilde{V} и \tilde{u} непосредственно. Однако для их решения также применим Maple:

> restart:

```

> assume (M>0, m>0) :
> eq1 := M*V^2+m*u^2=M*Vt^2+m*ut^2; eq2 := M*V+m*u=M*Vt+m*ut;
      eq1 := M~ V^2 + m~ u^2 = M~ Vt^2 + m~ ut^2
      eq2 := M~ V + m~ u = M~ Vt + m~ ut
> R := solve ({eq1, eq2}, {ut, Vt});
      R := {ut = u, Vt = V}, {ut = -\frac{u M~ - m~ u - 2 M~ V}{m~ + M~}, Vt = \frac{-V m~ + M~ V + 2 m~ u}{m~ + M~}}

```

Нас интересует значение скорости частицы после столкновения, скорость молекулы \tilde{u} никакого интереса не представляет. Итак, получено, что

$$\tilde{V} = \frac{V(M-m)}{M+m} + \frac{2m}{M+m}u. \quad (4.2)$$

Формулу (4.2) можно преобразовать с целью выделить отношение m/M , которое в силу того, что $m \ll M$ является малым, и по нему можно разложить:

$$\tilde{V} = V \frac{\left(1 - \frac{m}{M}\right)}{\left(1 + \frac{m}{M}\right)} + \frac{2 \frac{m}{M}}{\left(1 + \frac{m}{M}\right)} u \approx V + 2(u-V) \frac{m}{M} \quad (4.3)$$

(ограничились линейными по m/M членами).

В реальности m значительно меньше M , так что имеет смысл использовать упрощенный вариант формулы (4.3). Однако для составляемой учебной программы это не имеет значения, поскольку и точная формула достаточно проста и позволит провести вычисления в случае, если число шагов не будет слишком велико. В случае большого числа шагов выбор более простой, но физически обоснованной формулы может иметь решающее значение для практической выполнимости вычислений. Имеет смысл делать прикидки быстродействия различных вариантов, используя ту же программу, в которой будут производиться расчеты. Например, в Maple в правой нижней части командного окна имеется счетчик времени. Проведем эксперимент: выполним простые, но многократные вычисления по двум вариантам формулы, например, присваивание значений некоторо-

му массиву для достаточно большого числа значений. Первый вариант (по точному варианту формулы):

```
> M:=100:m:=1:u:=10:
> for j from 1 to 200000 do t[j]:=j*(1-
m/M)/(1+m/M)+2*(m/M)*u/(1+m/M):od:
```

Второй вариант (по приближенному варианту):

```
> M:=100:m:=1:u:=10:
> for j from 1 to 200000 do t[j]:=j+2*(u-j)*(m/M):od:
```

Время выполнения этих программ с 200000 шагами каждая: первой – 6,4 секунды, второй – 2,2 секунды. Различие очень существенное, так что если масса молекулы действительно намного меньше массы частицы, лучше использовать более простую приближенную формулу.

Поскольку модель вероятностная, программа будет включать в себя использование случайных чисел для моделирования вероятностных распределений. Генераторы случайных (точнее, псевдослучайных) чисел имеются во всех без исключения численных и численно-аналитических программах. Разобраться в способах использования этих генераторов не составляет труда, вызывая соответствующие разделы Help. Так, в Maple можно описать процедуру генерирования (целых) случайных чисел в заданном диапазоне (например, от 1 до 100):

```
> restart:
> q:=rand(1..100);
      q:=proc() (proc() option builtin; 391 end proc)(6, 100, 7) + 1 end proc
> q();q();q();q();
```

93

45

96

6

Как видим, вызов процедуры производится с помощью оператора *имя процедуры()*. Однако есть одно, но очень существенное «но». Попро-

буйте выполнить ту же программу, начиная с *restart*, потом еще и еще раз. Можно заметить, что генерируются в точности одни и те же, якобы «случайные», числа! Ясно, что так использовать генератор случайных чисел в составе моделирующей программы нельзя, т.к. при различных «прогонах» («runs» в англоязычной литературе) будут получаться одинаковые результаты и никакого усреднения не получится. Это является проявлением того, что генерируются не истинно случайные (в компьютере это невозможно), а псевдослучайные числа. Они представляют собой результат вычисления по некоторому, вполне определенному, но очень запутанному алгоритму. Т.е. оказывается, что случайные числа, необходимые для моделирования, сами представляют собой результат некоторого моделирования!

Выход заключается в том, что указанный алгоритм генерирования псевдослучайных чисел можно запускать, начиная с разных моментов, и этот стартовый момент можно связать, например, с текущим показанием часов компьютера. Это делается с помощью оператора *randomize()*, который в вышеприведенной программе надо выполнить между *restart* и $q=rand(1..100);$. Читателю предоставляется возможность проверить, что тогда при различных прогонах, начиная с *randomize()*, генерируются разные числа. Эта процедура называется «посев» последовательности псевдослучайных чисел (*seed* по-английски).

Приступим к составлению программы. Вначале, помимо *randomize*, загрузим библиотеку *stats* (вычисления статистических характеристик) с помощью оператора *with*. Зададим значения параметров модели M, m, u, p . Зададим T – количество элементарных временных шагов, т.е. общее время, в течение которого будет функционировать модель. Вычислим и сохраним под некоторыми именами те части формулы (4.2), которые не будут изменяться со временем: $c_1 = \frac{(M - m)}{M + m}$ и $c_{2u} = \frac{2m}{M + m}u$. Единственная величина в (4.2), которая изменяется со временем – это скорость частицы, ввиду по-

стоянных ударов со стороны молекул, передающих частице свой импульс. Кроме того, зададим две процедуры генерирования случайных чисел: одно в интервале от 1 до 100, другое – бинарное (1 или 2).

```
> restart:
> with(stats):randomize():
> T:=1000:
> M:=100:m:=1:u:=10:p:=0.75:
> c1:=(M-m)/(M+m):c2u:=u*2*m/(M+m):
> ra:=rand(1..100): roll_1:= rand(1..2):
```

Основными динамическими величинами в нашей модели являются две: координата частицы X и ее скорость V . Моделирование заключается в том, чтобы вычислить значения координаты и скорости в каждый из T моментов времени. Таким образом, X и V представляют собой массивы длиной T . Опишем их как векторы. В начальный момент положим координату и скорость равными нулю, кроме того, на всякий случай занулим вначале и все остальные элементы массивов:

```
> X:=vector(T):V:=vector(T):
> for k from 1 to T do V[k]:=0:X[k]:=0:od:
```

Согласно динамическим законам, координата и скорость в момент времени t определяются значениями координаты и скорости в предшествующий момент времени. Если интервал времени между последовательными моментами равен Δt , тогда имеем из кинематики

$$\begin{aligned} V(t + \Delta t) &= V(t) + a(t)\Delta t \\ X(t + \Delta t) &= X(t) + V(t)\Delta t + a(t)\Delta t^2 / 2 \end{aligned} \quad (4.4)$$

Необходимо как-то конкретизировать Δt . Будем считать, что этот элементарный промежуток времени достаточно мал, но при этом он значительно больше, чем характерное время взаимодействия (столкновения) молекулы и частицы. Такого рода предположение о малом промежутке времени или расстоянии/объеме является стандартным в статистической и молекулярной физике. Поэтому вместо первого уравнения (4.4) будем

предполагать, что скорость $V(t + \Delta t) = V(t)$, т.е. не меняется, но в случае, если происходит столкновение с молекулой, скорость частицы мгновенно меняется на новое значение (4.3). Что касается второго уравнения (4.4), то по той же причине (мгновенность взаимодействия, т.е. мгновенность действия ускорения a) в нем не будет последнего слагаемого.

Для удобства положим $\Delta t = 1$, это означает, что время в нашей модели будет измеряться в единицах Δt . Теперь все готово для запуска основной части нашей программы: цикл по времени от 1 до T , на каждом шаге которого будет осуществляться расчет очередных значений массивов X и V на основе предыдущих значений. При этом на каждом шаге с помощью случайных чисел решаются два вопроса: а) будет ли на этом шаге столкновение с молекулой; б) если будет, то где – справа или слева; вот для чего нам понадобятся две последовательности случайных чисел.

```
> for j from 1 to T-1 do
X[j+1]:=evalf(X[j]+V[j]):
if(ra()/100<p) then
if(roll_1(=1) then V[j+1]:=V[j]*c1+c2u:
else V[j+1]:=V[j]*c1-c2u:fi
else V[j+1]:=V[j]:fi:
od:
```

Надо помнить, что весь этот текст написан в одном командном поле программы (выполняется одним нажатием Enter). Разделение по строкам сделано искусственно с помощью клавиши пробела для большей наглядности и понимания структуры программы. Сделаем несколько пояснений. Оператор $x[j+1]:=evalf(x[j]+v[j]):$ вычисляет новое значение координаты по второму уравнению (4.4) без последнего члена. Далее используется случайное целое число ra , которое ранее было нормировано на диапазон от 0 до 100. Но поскольку вероятность столкновения с молекулой p у нас задается в долях единицы, то случайное число используется в виде $ra/100$, т.е. нормируется от 0 до 1. Имеем два вложенных друг в друга опе-

ратора *if*: внешний сравнивает вероятность столкновения p с $ra/100$, и если окажется, что $p < ra/100$, это означает, что столкновение с молекулой произойдет. Тогда выполняется второй (внутренний) оператор *if*, в котором решается («разыгрывается»), с какой стороны оно произойдет: если случайное число $roll_1=1$, то считаем, что молекула «ударилась» слева, что соответствует положительному знаку ее скорости u . В противном случае ($roll_1=2$) знак скорости молекулы будет отрицательным, т.е. она столкнется с частицей справа. В обоих случаях во внутреннем операторе *if* используется точный вариант формулы (4.3), но с разными знаками u .

Если же окажется, что $p > /100$, то столкновения на данном шаге не будет, и выполняется опция *else* внешнего оператора *if*: $v[j+1] := v[j]$, т.е. скорость остается без изменения.

Если основной цикл выполнен без сбоев, это означает, что массивы X и V получены. Для наглядной визуализации построим их графики. Впрочем, для броуновской частицы интерес представляет только ее координата. Для построения точечного графика (т.е. графика по отдельным заданным точкам) в Maple необходимо представить данные в формате «список данных» (list). Формат «vector», в котором у нас представлен массив X , не будет принят оператором построения графиков *plot* (убедитесь в этом). Создадим массив под именем $X_$ в виде двойного списка аргумент–значение функции:

```
> x_ := [[n, X[n]] $n=1..T] :
```

Эквивалентная форма (*seq* – от sequence – последовательность):

```
> x_p := {seq([t, X[t]], t=1..T)} :
```

Вот что представляют собой элементы $X_$:

```
> x_[1]; x_[10];
```

```
[1, 0]
```

```
[10, 0.7464728909]
```

Теперь можно применить *plot*:

```
> plot(x_, x=1..T, style=point, symbol=circle);
```

Впрочем, можно построить точечный график, используя специально предназначенный для этого оператор *pointplot*, входящий в состав графической библиотеки *plots* (которую необходимо загрузить оператором *with*). Формат, приемлемый для него, – тот же самый двойной список.

```
> with(plots);
```

```
> pointplot(x_);
```

В результате получаем график координаты частицы в зависимости от времени, т.е. «историю» ее случайных блужданий при данном прогоне программы. Можно повторить программу, начиная или с *restart* или с *randomize*, тогда получим другие результаты. На рисунке 4.1 показано несколько получившихся графиков.

Несмотря на то, что для броуновской частицы интерес представляет только ее координата, для исследования самой модели и анализа ее адекватности, построим также график массива скорости частицы V :

```
> V_ := [[n, V[n]] $n=1..T]:
```

```
> plot(V_, x=1..T, style=line);
```

Типичный график показан на рисунке 4.2.

Из графика видно, что скорость частицы является негладкой (не дифференцируемой) функцией. Этого и следовало ожидать, исходя из предположения, заложенного в модель, о мгновенном характере изменения скорости при ударе со стороны молекулы.

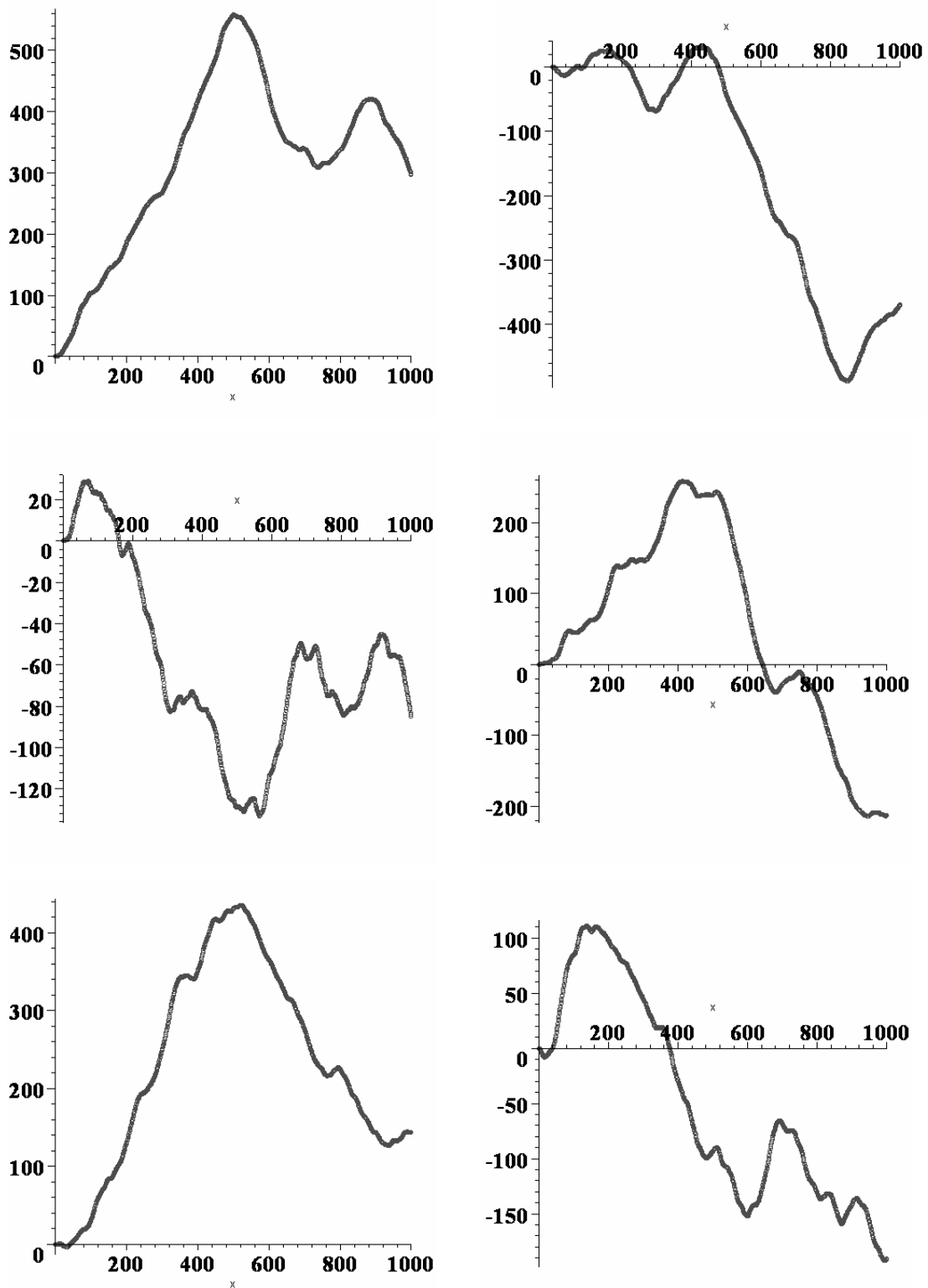


Рис. 4.1. Изменения во времени координаты одномерного движения броуновской частицы, полученные при различных прогонах программы

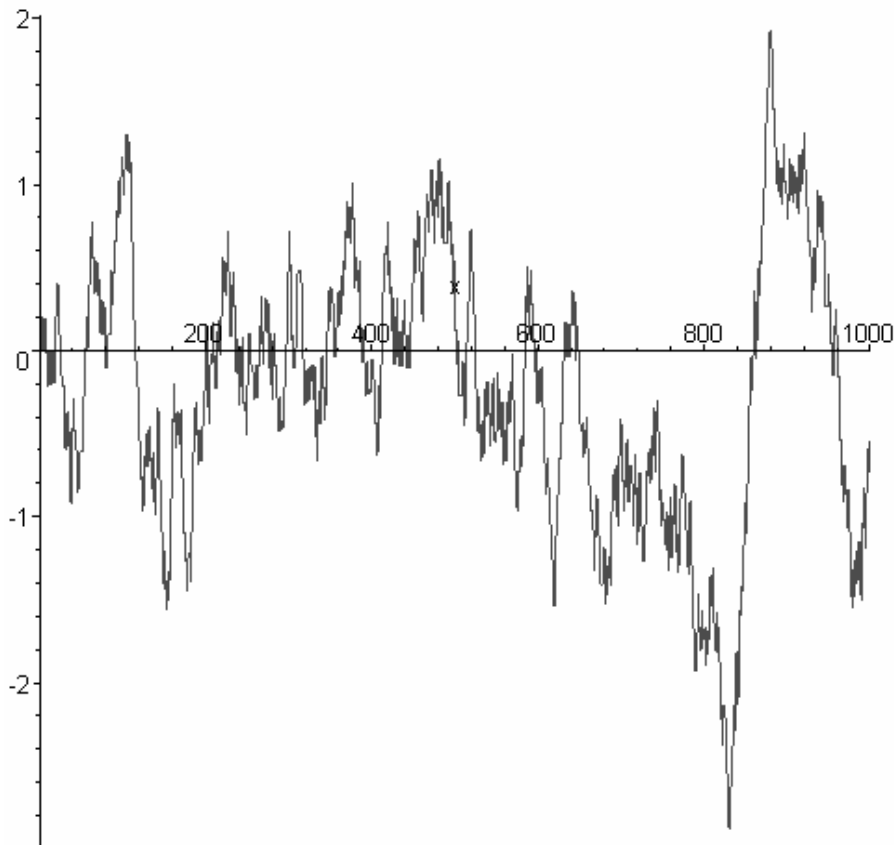


Рис. 4.2. Изменение скорости одномерного движения броуновской частицы

Здесь настает момент, когда необходимо остановиться и задаться вопросом: что, собственно говоря, сделано и с какой целью, т.е. что полезного из всего этого можно извлечь? Имеет ли смысл получать различные графики, подобные изображенным на рисунке 4.1, и анализировать их? Например, при первом прогоне, как видно из рисунка 4.1, график имеет восходящий тренд, это означает, что частица двигалась в сторону положительных значений координаты, а при втором и шестом прогонах она, наоборот, ушла налево. Может быть, это не случайно? Что все это означает? Ответ: само по себе – абсолютно ничего. Если бы мы создали модель и программу и ограничились получением нескольких графиков, как на рисунке 4.1, то мы бы зря потратили время. Ввиду вероятностного характера протекающих в модели процессов, смысл имеют не отдельные результаты

прогонов программы, а усредненные характеристики по многим прогонам. В этом состоит суть так называемого метода Монте-Карло. Несмотря на то, что слово «метод» употреблено в единственном числе, на самом деле это целая категория совершенно разнообразных методов и подходов, которые можно охарактеризовать следующим образом. Если вы в процессе вычислений или моделирования хотя бы один раз использовали случайные числа не для вспомогательных целей, а по сути ваших расчетов, то можно считать, что вы использовали метод Монте-Карло. Ясно поэтому, что наша модель является типичной стохастической моделью в рамках подхода Монте-Карло. Поэтому необходимо прогонять ее не один и не шесть раз, как на рисунке 4.1, а много, очень много раз, чтобы «набрать статистику» и получить надежные (состоятельные, по терминологии математической статистики) средние значения интересующих нас величин.

В случае броуновского движения интерес обычно представляет, насколько далеко от первоначального положения отклонилась частица по прошествии определенного времени T , причем по абсолютной величине, направление значения не имеет. Мы должны, следовательно, прогнать программу, например, 10000 раз (количество прогонов обозначим MC) и по окончании каждого прогона запоминать конечную координату частицы $X(T)$. Таким образом, получим массив длиной MC (обозначим его XMC). Разумеется, выполнять столько раз программу следует не вручную. Заключим основной цикл предыдущей программы в еще один внешний цикл длиной MC . Время наблюдения за системой T на этот раз сократим до 10. Новая программа имеет вид:

```
> restart:
> with(stats):randomize():
> T:=10:
> MC:=10000:
> M:=100: m:=1: u:=10: p:=0.75:
```

```

> c1:=evalf((M-m)/(M+m)):c2u:=evalf(u*2*m/(M+m)):
> X:=vector(T):V:=vector(T):XMC:=vector(MC):
> for i from 1 to MC do XMC[i]:=0:od:
> ra:=rand(1..100): roll_1 := rand(1..2):
> for ijk from 1 to MC do
for k from 1 to T do V[k]:=0:X[k]:=0:od:
for j from 1 to T-1 do X[j+1]:=evalf(X[j]+V[j]):
q1:=ra()/100:q2:=roll_1():if(q1<p) then if(q2=1) then
V[j+1]:=V[j]*c1+c2u:else V[j+1]:=V[j]*c1-c2u:fi else
V[j+1]:=V[j]:fi: od:XMC[ijk]:=X[T]:od:

```

Итак, элементы полученного массива *XMC* представляют собой координаты частицы в момент времени *T*, полученные при разных прогонах программы. Вычислим среднее значение элементов этого массива и их квадратов. Средние значения можно вычислить с помощью оператора *describe[mean]*, входящего в библиотеку *stats*. Но, подобно оператору *plot*, требуется преобразовать *XMC* к формату простого (одинарного) списка:

```

> xx := [ XMC[n] $n=1..MC]:xx2 := [ XMC[n]^2 $n=1..MC]:
> describe[mean](xx);describe[mean](xx2);
-0.01508749946
5.414696333

```

Как и следовало ожидать, среднее значение смещения близко к нулю, поскольку вероятность ударов молекул слева и справа одинакова. Что касается среднего квадрата смещения, то он, очевидно, зависит от параметров системы, и при увеличении числа прогонов программы, его величина будет стремиться к определенному значению (а среднее смещение все ближе будет приближаться к нулю). Читателю предлагается убедиться в этом, увеличивая число прогонов *MC*.

Наконец, настал момент, когда необходимо критически оценить сделанное и определить, в чем построенная модель плоха и как ее можно совершенствовать. Один момент очевиден: для сравнения с реальными сис-

темами размерность пространства, в которой движется частица, должна равняться трем. Кроме того, существенным недостатком модели является то, что в промежутках между столкновениями частица движется свободно (т.е. на нее не действуют никакие силы). В реальности частица подвержена вязкому трению со стороны среды [1]. Броуновская частица будет двигаться по зигзагообразному пути, удаляясь постепенно от начальной точки. Вычисления показывают, что значение среднего квадрата смещения броуновской частицы $r^2 = x^2 + y^2 + z^2$ описывается формулой:

$$\langle r^2 \rangle = 6kT B t, \quad (4.5)$$

где B – подвижность частицы, которая обратно пропорциональна вязкости среды и размеру частицы. Эта формула, называемая формулой Эйнштейна, была со всей возможной тщательностью подтверждена экспериментально французским физиком Жаном Перреном (1870–1942). На основе измерения параметров движения броуновской частицы Перрен получил значения постоянной Больцмана и число Авогадро, хорошо согласующиеся в пределах ошибок измерений со значениями, полученными другими методами.

Итак, мы не можем сравнить нашу модель с экспериментом и с теорией, сопоставив полученное значение среднего квадрата смещения с формулой (4.4). Модель необходимо сделать ближе к реальности, что будет проведено в лабораторной работе. Здесь же отметим еще раз, что программа упрощенной модели составит основу для программы будущей модели. Тем не менее, для иллюстрирования методики сравнения результатов имитационного моделирования с теоретическими и/или экспериментальными данными рассмотрим в следующем пункте упрощенный вариант блуждания по прямой, для которого имеется *точное* теоретическое описание.

4.2. Случайные блуждания на прямой: упрощенный случай

Рассмотрим стохастическое блуждание точки вдоль прямой «в чистом виде», т.е. вне всякой связи с динамикой. Допустим, что точка в каждый момент времени может с одинаковой вероятностью сместиться на одну единицу длины вправо или влево. Тогда имеется точный результат [2, с. 10–11]: вероятность того, что в момент времени T координата частицы будет равна k , дается следующим выражением:

$$P_T(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} (\cos \varphi)^T e^{-i\varphi k} d\varphi . \quad (4.6)$$

Для имитационного моделирования системы модифицируем программу п. 4.1. Ясно, что теперь программа будет существенно проще: не будет массива скорости, достаточно использовать только одно бинарное случайное число, расчет нового значения координаты будет выполняться по совсем простой формуле $X[j+1] = X[j] \pm 1$. Программа имеет вид:

```
> restart:
> with(stats):randomize():
> T:=10:
> MC:=100000:
> X:=vector(T):XMC:=vector(MC):
> for i from 1 to MC do XMC[i]:=0:od:
> roll_1 := rand(1..2):
> for ijk from 1 to MC do
for k from 1 to T do X[k]:=0:od:
for j from 1 to T-1 do q2:=roll_1(): if(q2=1) then
X[j+1]:=X[j]+1:else X[j+1]:=X[j]-1:fi: od: XMC[ijk]:=X[T]:od:
> xx := [ XMC[n] $n=1..MC]:xx2 := [ XMC[n]^2 $n=1..MC]:
> evalf(describe[mean](xx));evalf(describe[mean](xx2));
```

0.0002200000000

8.990640000

Первый результат понятен: ввиду равноправия левого и правого, среднее смещение равно нулю. Второе значение – средний квадрат смещения – явно равно 9. Вычисляя с большим количеством прогонов, т.е. увеличивая MC еще на порядок – до 10^6 , получаем еще более точное значение

$$8.998896000$$

Соответствует ли полученное значение среднего квадрата смещения истинному распределению вероятностей (4.6)? Вычислим вероятности по формуле (4.6), имея в виду, что в вышеприведенной программе мы фактически рассматривали не T , а $T-1$ шагов блужданий:

```
> for j from -(T-1) to T-1 do w[j]:=evalf(int(cos(phi)^(T-1)*exp(-I*phi*j), phi=-Pi..Pi)/2/Pi);od:
```

Любое распределение вероятностей полного (т.е. всех возможных) набора событий должно быть нормировано на единицу, что необходимо всегда обязательно проверять, вычисляя сумму всех значений вероятностей:

```
> sum(w[jjk], jjk=-(T-1)..T-1);
```

$$1.000000000$$

Итак, под именем w вычислено точное распределение вероятностей координат частицы после $T-1$ случайных шагов. Вычислим те же вероятности по результатам нашего имитационного моделирования. Для этого для каждого значения координаты, необходимо посчитать, сколько раз среди наших MC прогонов, после $T-1$ шагов, получалось это значение координаты, и разделить на MC :

```
> for y from -(T-1) to T-1 do S[y]:=0:for jj from 1 to MC do
if(xMC[jj]=y) then S[y]:=S[y]+1:fi:
od:P[y]:=evalf(S[y]/MC):od:
```

По терминологии математической статистики, P – это так называемые выборочные значения вероятностей (фактически выборочные частоты), т.е. полученные из выборки конечного объема, в данном случае MC .

Сравним выборочные вероятности для объема $MC=100000$ и точные, распечатав их в формате *list* (простой список):

```
> [P[a] $a=0..T-1];
```

```
[0., 0.2468900000, 0., 0.1667400000, 0., 0.06917000000, 0., 0.01734000000, 0.,  
0.001850000000]
```

```
> [w[a] $a=0..T-1];
```

```
[0., 0.2460937500, 0., 0.1640625000, 0., 0.07031250000, 0., 0.01757812500, 0.,  
0.001953125000]
```

Видим, что имеется хорошее совпадение выборочного и точного распределений. По закону больших чисел, при увеличении объема выборки выборочные значения должны все более приближаться к истинным. Читателю предоставляется убедиться в этом, увеличив MC на порядок.

Наконец, вычислим средний квадрат координаты по точному распределению вероятностей w :

```
> sum(w[jjk] * jjk^2, jjk=-(T-1) .. T-1);
```

```
9.000000000
```

Оно действительно равно 9, как мы и получили в нашем имитационном эксперименте!

4.3. Вычисление статистической суммы модели Изинга и сравнение с известными точными выражениями

В качестве еще одного примера применения численно-аналитических пакетов компьютерной алгебры для вычисления сложных систем рассмотрим модель Изинга. Это определенная модель двумерной решетки, для которой точное выражение для статистической суммы и задача о фазовом переходе была впервые решена Л. Онсагером в 1944 г. [4, с. 541–549]. Этот результат часто называют одним из самых выдающихся достижений теоретической физики в 20 веке.

Рассматриваемая модель представляет собой плоскую квадратную решетку, состоящую из N узлов ($N = L^2$, где L – сторона квадрата), в каждом из которых находится «спин», с осью, перпендикулярной к плоскости решетки. Спин может иметь две противоположные ориентации, так что общее число всевозможных конфигураций спинов в решетке равно 2^N .

Для описания различных конфигураций поступают следующим образом. С каждым узлом решетки (с координатами k, l) свяжем переменную σ_{kl} , принимающие два значения ∓ 1 , соответствующие двум возможным ориентациям спина. Если ограничиться только учетом взаимодействия между соседними спинами, то энергия конфигурации может быть записана в виде:

$$E(\sigma) = -J \sum_{k,l=1}^L (\sigma_{kl} \sigma_{k,l+1} + \sigma_{kl} \sigma_{k+1,l}) . \quad (4.7)$$

Здесь параметр J определяет энергию взаимодействия пары соседних спинов, равную $-J$ и $+J$ соответственно для одинаковых и противоположных ориентаций спинов.

Основная задача состоит в вычислении статистической суммы

$$Z = \sum_{(\sigma)} e^{-E(\sigma)/k_B T} = \sum_{(\sigma)} \exp \left\{ K \sum_{k,l=1}^L (\sigma_{kl} \sigma_{k,l+1} + \sigma_{kl} \sigma_{k+1,l}) \right\} , \quad (4.8)$$

где $K = J/k_B T$, k_B – постоянная Больцмана.

Несмотря на кажущуюся простоту, в случае размерности больше единицы, задача относится к числу сложнейших и показывает, что сложная для вычислений система может возникнуть в относительно простой исходной ситуации. Знаменитое решение Онсагера является не вполне точным, а асимптотическим выражением, справедливым в пределе больших N (т.е. для бесконечной решетки). Оно имеет вид

$$Z = 2^N (1-x^2)^{-N} \prod_{p,q=0}^L \left[(1+x^2)^2 - 2x(1-x^2) \left(\cos \frac{2\pi p}{L} + \cos \frac{2\pi q}{L} \right) \right]^{1/2} , \quad (4.9)$$

где $x = th(K)$.

Что касается действительно точного выражения, справедливого для конечной решетки, то оно также было получено Онсагером совместно с Кауфман, и называется решением Кауфман [5, 6]. Однако в отличие от (4.9) для конечного случая требуется задание конкретных граничных условий. Обычно используют так называемые циклические граничные условия, когда крайние (справа – слева и сверху – внизу) спины предполагаются взаимодействующими друг с другом. Пример решетки с $N=16$ показан на рисунке 4.3.

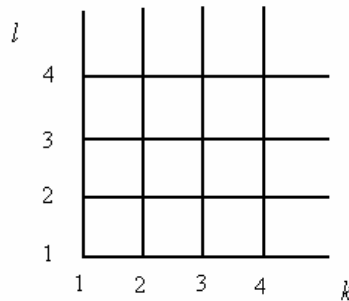


Рис. 4.3. Квадратная решетка спинов размера 4 x 4 ($L = 4, N = L^2 = 16$) с циклическими граничными условиями

В этом случае точное выражение для статистической суммы, полученное Кауфман, имеет вид суммы четырех слагаемых:

$$Z = \frac{1}{2} \left(Z^{(1)} + Z^{(2)} + Z^{(3)} + Z^{(4)} \right) ;$$

$$Z^{(1)} = \left(e^{2K} - e^{-2K} \right)^{\frac{N}{2}} \prod_{j=0}^{L-1} \left(e^{\frac{L}{2} \beta_{2j+1}} + e^{-\frac{L}{2} \beta_{2j+1}} \right) ;$$

$$Z^{(2)} = \left(e^{2K} - e^{-2K} \right)^{\frac{N}{2}} \prod_{j=0}^{L-1} \left(e^{\frac{L}{2} \beta_{2j+1}} - e^{-\frac{L}{2} \beta_{2j+1}} \right) ;$$

$$Z^{(3)} = \left(e^{2K} - e^{-2K} \right)^{\frac{N}{2}} \prod_{j=0}^{L-1} \left(e^{\frac{L}{2} \beta_{2j}} + e^{-\frac{L}{2} \beta_{2j}} \right) ;$$

$$Z^{(4)} = \left(e^{2K} - e^{-2K} \right)^{\frac{N}{2}} \prod_{j=0}^{L-1} \left(e^{\frac{L}{2} \beta_{2j}} - e^{-\frac{L}{2} \beta_{2j}} \right), \quad (4.10)$$

где параметры β_j являются решениями уравнений: при $0 < j < 2L$

$$e^{\beta_j} + e^{-\beta_j} = \frac{(e^{2K} + e^{-2K})^2}{(e^{2K} - e^{-2K})} - 2 \cos\left(\frac{\pi j}{L}\right) \quad (4.11)$$

и при $j = 0$

$$e^{\beta_0} = e^{2K} \frac{(e^K - e^{-K})}{(e^K + e^{-K})}. \quad (4.12)$$

Задачей данной лекции является демонстрация того, как с помощью системы компьютерной алгебры можно как непосредственно рассчитать статистическую сумму модели Изинга двумерной квадратной решетки, так и по формулам (4.10) – (4.12) и, сравнив полученные выражения, убедиться в справедливости решения Кауфман (разумеется, лишь для нескольких небольших значений L). Кроме того, также для нескольких L вычислим точные выражения для статистической суммы в случае наличия внешнего магнитного поля. Как известно, задача Изинга на простой квадратной решетке в магнитном поле до сих пор не решена, так что сравнивать с точным выражением, подобно (4.10) – (4.12), будет невозможно. Здесь мы соприкасаемся с нерешенной проблемой, находящейся на переднем крае исследований.

Но прежде, следуя принятому в данном курсе принципу «от простого к сложному», разберем более простой случай одномерной задачи Изинга. Она решена точно как в отсутствие, так и в присутствии магнитного поля. Статистическая сумма имеет вид

$$Z = \sum_{(\sigma)} e^{-E(\sigma)/k_B T} = \sum_{(\sigma)} \exp\left\{ K \sum_{k,l=1}^L (\sigma_k \sigma_{k+1}) \right\}, \quad (4.13)$$

причем согласно циклическому граничному условию $\sigma_{L+1} = \sigma_1$. Статистическая сумма при наличии магнитного поля:

$$Z = \sum_{(\sigma)} e^{-E(\sigma)/k_B T} = \sum_{(\sigma)} \exp \left\{ K \sum_{k,l=1}^L (\sigma_k \sigma_{k+1}) + h \sum_{k=1}^L (\sigma_k) \right\}, \quad (4.14)$$

где $h = H/k_B T$, H – напряженность магнитного поля, k_B – постоянная Больцмана.

Точное выражение для статистической суммы (4.14) может быть вычислено достаточно просто и имеет вид

$$Z = \left[e^K ch(h) + e^{-K} \sqrt{1 + e^{4K} sh^2(h)} \right]^N + \left[e^K ch(h) - e^{-K} \sqrt{1 + e^{4K} sh^2(h)} \right]^N \quad (4.15)$$

При отсутствии магнитного поля, полагая в (4.15) $h=0$, имеем:

$$Z = \left[e^K + e^{-K} \right]^N + \left[e^K - e^{-K} \right]^N. \quad (4.16)$$

В начале программы зададим длину цепочки спинов (начнем с наименьшего $N=2$) и вычислим выражение (4.16):

> restart;

> N:=2;

> Z := (exp(K) + exp(-K)) ^ N + (exp(K) - exp(-K)) ^ N;

$$Z := (e^K + e^{(-K)})^2 + (e^K - e^{(-K)})^2$$

Далее вычислим (4.13):

> S := sum(sum(product(abs(sigma[m]), m=1..N) * exp(K) ^ (sum(sigma[k] * sigma[k+1], k=1..N-1) + sigma[N] * sigma[1]), sigma[1] = -1..1), sigma[2] = -1..1);

$$S := 2 (e^K)^2 + \frac{2}{(e^K)^2}$$

В этом выражении искусственно введено произведение абсолютных величин $sigma[1]$ и $sigma[2]$ для того, чтобы исключить из сумм те слагаемые, в которых хотя бы одна из $sigma[1]$, $sigma[2]$ равна нулю, чтобы

можно было использовать диапазон суммирования $-1..1$. Он включает в себя $-1, 0, 1$, а нужно только $-1, 1$. Итак, под именем Z у нас точное выражение (4.16), а под именем S – непосредственно вычисленное по (4.13). Сравним их:

```
> simplify(S-Z);
```

0

Они совпадают, следовательно, (4.16) является правильным выражением в случае $N=2$. Можно проверить и для других значений N , но тогда надо не только переопределить N , но также добавить дополнительные суммирования в операторе, вычисляющем статистическую сумму по (4.13). Например, для $N=3$:

```
> restart:
```

```
> N:=3:
```

```
> Z:=(exp(K)+exp(-K))^N+(exp(K)-exp(-K))^N;
```

$$Z := (e^K + e^{(-K)})^3 + (e^K - e^{(-K)})^3$$

```
> S:=sum(sum(sum(product(abs(sigma[m]),m=1..N)*exp(K)^(sum(sigma[k]*sigma[k+1],k=1..N-1)+sigma[N]*sigma[1]),sigma[1]=-1..1),sigma[2]=-1..1),sigma[3]=-1..1);
```

$$S := 2(e^K)^3 + \frac{6}{e^K}$$

```
> simplify(S-Z);
```

0

Теперь можно сделать шаг вперед и проверим выражение (4.15), сравним его с (4.14). Сделаем это для $N=4$.

```
> restart:
```

```
> N:=4:
```

```
> Z:=(exp(K)*(exp(h)+exp(-h))/2+exp(-K)*sqrt(1+exp(4*K)*((exp(h)-exp(-h))/2)^2))^N+(exp(K)*(exp(h)+exp(-h))/2-exp(-K)*sqrt(1+exp(4*K)*((exp(h)-exp(-h))/2)^2))^N;
```

$$Z := \left(\frac{1}{2} e^K (e^h + e^{(-h)}) + e^{(-K)} \sqrt{1 + e^{(4K)} \left(\frac{1}{2} e^h - \frac{1}{2} e^{(-h)} \right)^2} \right)^4 + \left(\frac{1}{2} e^K (e^h + e^{(-h)}) - e^{(-K)} \sqrt{1 + e^{(4K)} \left(\frac{1}{2} e^h - \frac{1}{2} e^{(-h)} \right)^2} \right)^4$$

```
> S := sum (sum (sum (sum (product (abs (sigma [m] ), m=1..N) * exp (K) ^ (sum (s
igma [k] * sigma [k+1] , k=1..N-
1) + sigma [N] * sigma [1] ) * exp (h) ^ sum (sigma [r] , r=1..N) , sigma [1] =-
1..1) , sigma [2] =-1..1) , sigma [3] =-1..1) , sigma [4] =-1..1) ;
```

$$S := \frac{(e^K)^4}{(e^h)^4} + \frac{4}{(e^h)^2} + 4 + \frac{2}{(e^K)^4} + 4 (e^h)^2 + (e^K)^4 (e^h)^4$$

```
> simplify (S-Z) ;
```

0

Освоив одномерный случай как в отсутствии, так и в присутствии магнитного поля и имея полную уверенность в том, что наша программа правильно работает в этом случае, можно обобщить ее на двумерную квадратную решетку. Непосредственно вычисляем статистическую сумму по формуле (4.8) и сравниваем с решением Кауфман (4.10) – (4.12). Начнем с наименьшей решетки $L=2$, т.е. размера 2×2 .

```
> restart:with(linalg) :
```

```
> L:=2 :
```

```
> for j from 1 to 2*L-1 do be-
```

```
ta_[j] := arccosh((ek^2+1/ek^2)^2/(ek^2-1/ek^2)/2-
```

```
cos(Pi*j/L)) : od :
```

```
> beta_[0] := ln( ek^2*(sqrt(ek^2) -
```

```
1/sqrt(ek^2)) / (sqrt(ek^2)+1/sqrt(ek^2)) :
```

```
> c := (ek^2-1/ek^2)^(L^2/2) :
```

Здесь вычислили величины β_j по формулам (4.11), (4.12). Обратим внимание, что (4.11) было решено относительно β_j , используя обратную гиперболическую функцию *arccosh*. Кроме того, здесь вместо $\exp(K)$, используем символ *ek*. Это сделано для ускорения расчетов, поскольку при

больших размерностях использование явного выражения $\exp(K)$ заставляет программу проводить слишком громоздкие преобразования. Получив результаты, зависящие от ek , всегда можно с помощью оператора *subs* подставить явное выражение. Такой прием часто может оказаться весьма полезным. Разумеется, такого рода трудности выясняются на практике, при попытках вычислить сначала обычным образом.

Далее вычисляем слагаемые формулы (4.10) и всю сумму:

```
>z1:=c*product (exp (L*beta_ [2*j1+1] /2) +1/exp (L*beta_ [2*j1+1] /2) ,
j1=0..L-1) :
> z2:=c*product (exp (L*beta_ [2*j2+1] /2) -
1/exp (L*beta_ [2*j2+1] /2) , j2=0..L-1) :
>z3:=c*product (exp (L*beta_ [2*j3] /2) +1/exp (L*beta_ [2*j3] /2) , j3=0
..L-1) :
> z4:=c*product (exp (L*beta_ [2*j4] /2) -
1/exp (L*beta_ [2*j4] /2) , j4=0..L-1) :
>a1:=simplify (z1, symbolic) : a2:=simplify (z2, symbolic) : a3:=simpli
fy (z3, symbolic) : a4:=simplify (z4, symbolic) :
> Z:=simplify (a1+a2+a3+a4) /2;
```

$$\begin{aligned}
Z := & 4 (1 + 2 ek^4 + 16 ek^8 \\
& + \sqrt{ek^8 + 2 ek^4 + 1 + 2 ek^6 - 2 ek^2} \sqrt{ek^8 + 2 ek^4 + 1 - 2 ek^6 + 2 ek^2} \\
& + 12 ek^{12} \sqrt{ek^8 + 2 ek^4 + 1 + 2 ek^6 - 2 ek^2} \sqrt{ek^8 + 2 ek^4 + 1 - 2 ek^6 + 2 ek^2} \\
& + 7 ek^8 \sqrt{ek^8 + 2 ek^4 + 1 + 2 ek^6 - 2 ek^2} \sqrt{ek^8 + 2 ek^4 + 1 - 2 ek^6 + 2 ek^2} \\
& + 2 ek^4 \sqrt{ek^8 + 2 ek^4 + 1 + 2 ek^6 - 2 ek^2} \sqrt{ek^8 + 2 ek^4 + 1 - 2 ek^6 + 2 ek^2} + 62 ek^{16} \\
& + 14 ek^{12} + 16 ek^{24} + 14 ek^{20} \\
& + ek^{24} \sqrt{ek^8 + 2 ek^4 + 1 + 2 ek^6 - 2 ek^2} \sqrt{ek^8 + 2 ek^4 + 1 - 2 ek^6 + 2 ek^2} \\
& + 2 ek^{20} \sqrt{ek^8 + 2 ek^4 + 1 + 2 ek^6 - 2 ek^2} \sqrt{ek^8 + 2 ek^4 + 1 - 2 ek^6 + 2 ek^2} \\
& + 7 ek^{16} \sqrt{ek^8 + 2 ek^4 + 1 + 2 ek^6 - 2 ek^2} \sqrt{ek^8 + 2 ek^4 + 1 - 2 ek^6 + 2 ek^2} + ek^{32} \\
& + 2 ek^{28}) / (\\
& (ek^8 + 2 ek^4 + \sqrt{ek^8 + 2 ek^4 + 1 + 2 ek^6 - 2 ek^2} \sqrt{ek^8 + 2 ek^4 + 1 - 2 ek^6 + 2 ek^2} + 1)^2 \\
& ek^8) .
\end{aligned}$$

Для непосредственного вычисления статистической суммы (4.8) за основу можно взять соответствующий оператор из программы для одномерной модели:

```
>S:=sum(sum(sum(sum(product(product(abs(sigma[m,n])),m=1..L),n=1..L)*ek^(sum(sum(sigma[k,l]*sigma[k,l+1]+sigma[k,l]*sigma[k+1,l],k=1..L-1),l=1..L-1)+sum(sigma[L,l]*sigma[L,l+1]+sigma[L,l]*sigma[1,l],l=1..L-1)+sum(sigma[k,L]*sigma[k+1,L]+sigma[k,L]*sigma[k,1],k=1..L-1)+sigma[L,L]*sigma[L,1]+sigma[L,L]*sigma[1,L]),sigma[1,1]=-1..1),sigma[1,2]=-1..1),sigma[2,1]=-1..1),sigma[2,2]=-1..1);
```

$$S := 2 ek^8 + 12 + \frac{2}{ek^8}$$

```
> simplify(S-Z);
```

0

Итак, выражения совпали, что говорит о правильности формулы Кауфман для $L=2$. Аналогично можно проверить для больших размерностей. Так, при $L=3$ непосредственное вычисление по (4.8) дает:

```
> L:=3:
```

```
>S:=sum(sum(sum(sum(sum(sum(sum(product(product(abs(sigma[m,n])),m=1..L),n=1..L)*ek^(sum(sum(sigma[k,l]*sigma[k,l+1]+sigma[k,l]*sigma[k+1,l],k=1..L-1),l=1..L-1)+sum(sigma[L,l]*sigma[L,l+1]+sigma[L,l]*sigma[1,l],l=1..L-1)+sum(sigma[k,L]*sigma[k+1,L]+sigma[k,L]*sigma[k,1],k=1..L-1)+sigma[L,L]*sigma[L,1]+sigma[L,L]*sigma[1,L]),sigma[1,1]=-1..1),sigma[1,2]=-1..1),sigma[2,1]=-1..1),sigma[2,2]=-1..1),sigma[1,3]=-1..1),sigma[2,3]=-1..1),sigma[3,1]=-1..1),sigma[3,2]=-1..1),sigma[3,3]=-1..1);
```

$$S := 198 ek^2 + \frac{102}{ek^6} + \frac{144}{ek^2} + 48 ek^6 + 18 ek^{10} + 2 ek^{18}$$

а при $L=4$ (случай решетки, изображенной на рисунке 4.3):

```
> L:=4:
```

```

>S:=sum (sum (sum (sum (sum (sum (sum (sum (sum (sum (sum (sum (sum
(sum (product (product (abs (sigma [m,n] ), m=1..L), n=1..L) *ek^
(sum (sum (sigma [k,l] *sigma [k,l+1]+sigma [k,l] *sigma [k+1,l] ,
k=1..L-1), l=1..L-1) +
sum (sigma [L,l] *sigma [L,l+1]+sigma [L,l] *sigma [1,l] , l=1..L-
1)+sum (sigma [k,L] *sigma [k+1,L]+sigma [k,L] *sigma [k,1] , k=1..L-
1) +
sig-
ma [L,L] *sigma [L,1]+sigma [L,L] *sigma [1,L] ), sigma [1,1]= -
1..1), sigma [1,2]= -1..1), sigma [2,1]= -1..1), sigma [2,2]= -
1..1), sigma [1,3]= -1..1), sigma [2,3]= -1..1), sigma [3,1]= -
1..1), sigma [3,2]= -1..1), sigma [3,3]= -1..1), sigma [1,4]= -
1..1), sigma [2,4]= -1..1), sigma [3,4]= -1..1), sigma [4,1]= -
1..1), sigma [4,2]= -1..1), sigma [4,3]= -1..1), sigma [4,4]= -1..1);
S := 20524 + \frac{13568}{ek^4} + 13568 ek^4 + 6688 ek^8 + 424 ek^{16} + 1728 ek^{12} + \frac{2}{ek^{32}} + \frac{6688}{ek^8}
+ 32 ek^{24} + 64 ek^{20} + 2 ek^{32} + \frac{1728}{ek^{12}} + \frac{424}{ek^{16}} + \frac{32}{ek^{24}} + \frac{64}{ek^{20}}

```

Приведем еще результат для $L=5$

$$\begin{aligned}
S := & \frac{14800}{ek^{26}} + 74500 ek^{22} + \frac{314300}{ek^{18}} + 3140 ek^{30} + \frac{1024150}{ek^{14}} + \frac{82750}{ek^{22}} + 1014900 ek^{14} \\
& + \frac{5276500}{ek^6} + \frac{2645740}{ek^{10}} + 850 ek^{34} + \frac{2470}{ek^{30}} + 16300 ek^{26} + 5230300 ek^6 \\
& + 7431800 ek^2 + \frac{7413900}{ek^2} + 311800 ek^{18} + 2696080 ek^{10} + 50 ek^{42} + 2 ek^{50} \\
& + 100 ek^{38}
\end{aligned}$$

В этих случаях также сравнение с формулой Кауфман дает полное совпадение. Интересно отметить, однако, что с увеличением размерности вычисления по формулам Кауфман (4.10) – (4.12) становятся все более трудоемкими и занимают намного больше времени, чем непосредственное вычисление (4.8)! Впрочем, и последнее становится проблематичным уже при $L=7$, при использовании компьютера с обычными ресурсами, поскольку тогда общее количество вариантов конфигураций спинов, которые необходимо просуммировать, равно 2^{49} .

Построим графики полученных статистических сумм в зависимости от $K = J/K_B T$ (т.е. от обратной температуры). Для этого, как говорилось выше, подставим вместо символического ek явную экспоненту:

```
> f2 := subs (ek=exp (K) , S2) ; f3 := subs (ek=exp (K) , S3) ; f4 := subs (ek=exp (K) , S4) ; f5 := subs (ek=exp (K) , S5) :
```

$$f2 := 2 (e^K)^8 + 12 + \frac{2}{(e^K)^8}$$

$$f3 := 198 (e^K)^2 + 18 (e^K)^{10} + \frac{144}{(e^K)^2} + \frac{102}{(e^K)^6} + 2 (e^K)^{18} + 48 (e^K)^6$$

и применим оператор построения графиков, например, для $L=3$:

```
> plot ({f3}, K=0..0.6) ;
```

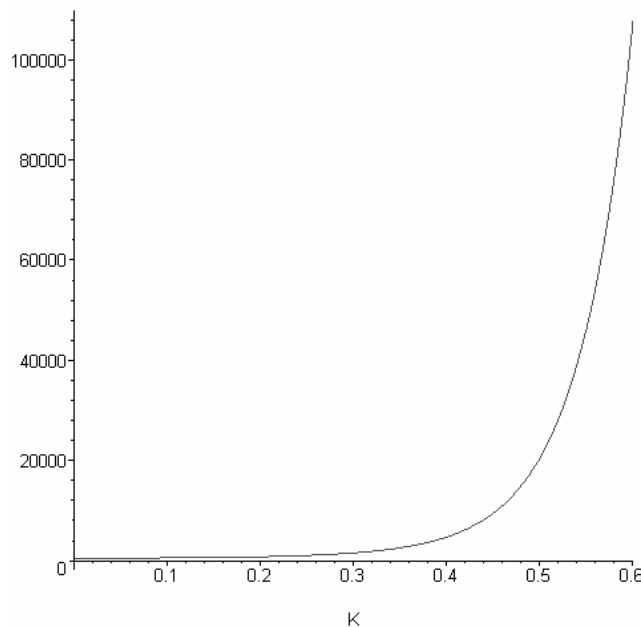


Рис. 4.4. Зависимость точной статистической суммы 3 x 3 квадратной решетки от $K = J/K_B T$

Видно, что статистическая сумма монотонно возрастает и не имеет сингулярностей (можно убедиться в этом, строя график в различных диапазонах). Т.е. фазовый переход в решетке размером 3 x 3 невозможен (что неудивительно для такой малой и простой системы). Как известно из теории модели Изинга [4, с. 548], фазовый переход возможен в случае бесконечной квадратной решетки, т.е. при $L \rightarrow \infty$. Статистическая сумма в этом

случае дается выражением (4.9). Вычислим это выражение для нескольких L и построим графики. Строго говоря, решение Онсагера является асимптотическим и совпадает с реальной статистической суммой лишь в пределе $L \rightarrow \infty$, т.е. тем точнее, чем больше L . Тем не менее, формально его можно вычислить даже при $L=2$.

Воплотим формулу (4.9) в программе, произведем вычисления для значений L от 4 до 18, подставим в полученные выражения вместо $X = th(K)$ и построим графики их логарифмов в зависимости от K . Результат приведен на рисунке 4.5. Видно, что статистические суммы имеют сингулярности в одной и той же точке. На графике верхней кривой $L=18$ может сложиться впечатление, что сингулярности нет, однако если построить тот же график с «большим разрешением», то можно убедиться, что сингулярность есть, как видно из рисунка 4.6.

```
> for j from 4 to 18 do L:=j:N:=L^2: Z[j]:=2^N*(1-X^2)^(-
N)*product(product((1+X^2)^2-2*X*(1-X^2)*
(cos(2*Pi*p/L)+cos(2*Pi*q/L)))^(1/2),p=0..L),q=0..L):
Z_[j]:=subs(X=tanh(K),Z[j]):od:
>plot({ln(Z_[4]),ln(Z_[5]),ln(Z_[6]),ln(Z_[7]),ln(Z_[8]),ln(Z_[
9]),ln(Z_[10]),ln(Z_[11]),ln(Z_[12]),ln(Z_[13]),ln(Z_[14]),ln(
Z_[15]),ln(Z_[16]),ln(Z_[17]),ln(Z_[18])},K=0..1.2);
```

Положение сингулярности согласуется со значением $X = X_C = \sqrt{2} - 1 \approx 0,414$, известным из литературы [4, с. 548].

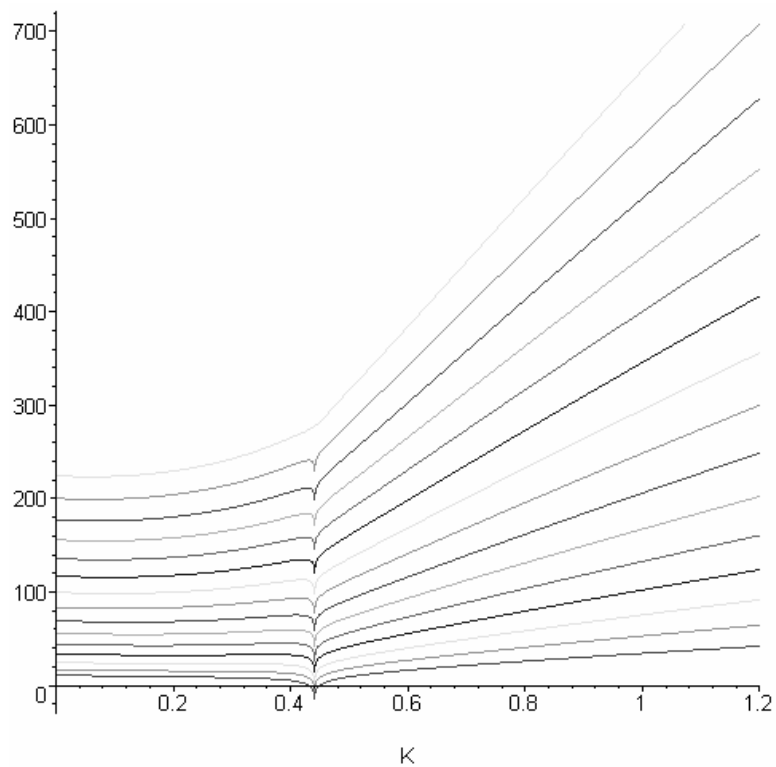


Рис. 4.5. Зависимость от $K = J / K_B T$ логарифма статистической суммы квадратной решетки по формуле Онсагера (4.9), для L от 4 (нижняя кривая) до 18 (верхняя кривая)

```
> plot({ln(Z_[18])}, K=0.4..0.6);
```

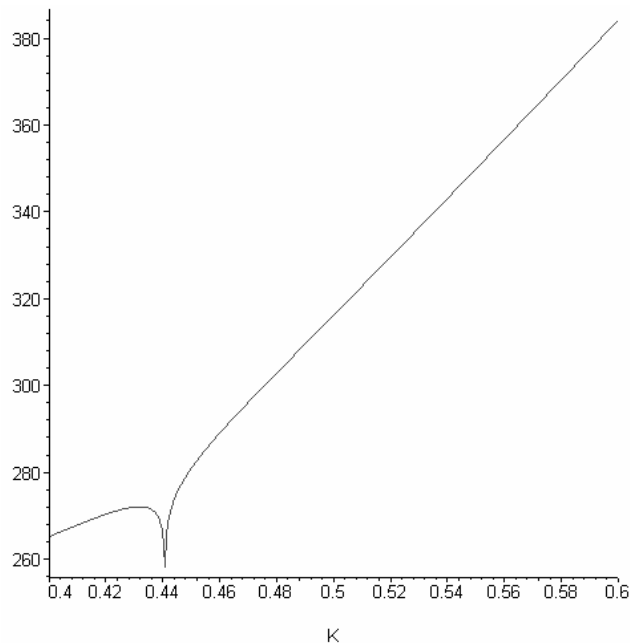


Рис. 4.6. Логарифм статистической суммы для $L=4$ квадратной решетки, полученный по формуле Онсагера (4.9) вблизи сингулярности

Как уже неоднократно указывалось, формула Онсагера (4.9) неправильно описывает статистическую сумму для решеток малых размеров. Построим на одном графике соответствующие кривые, чтобы наглядно увидеть, в чем состоит расхождение.

```
> plot({ln(Z_[4]), ln(f4), ln(Z_[5]), ln(f5)}, K=0..4);
```

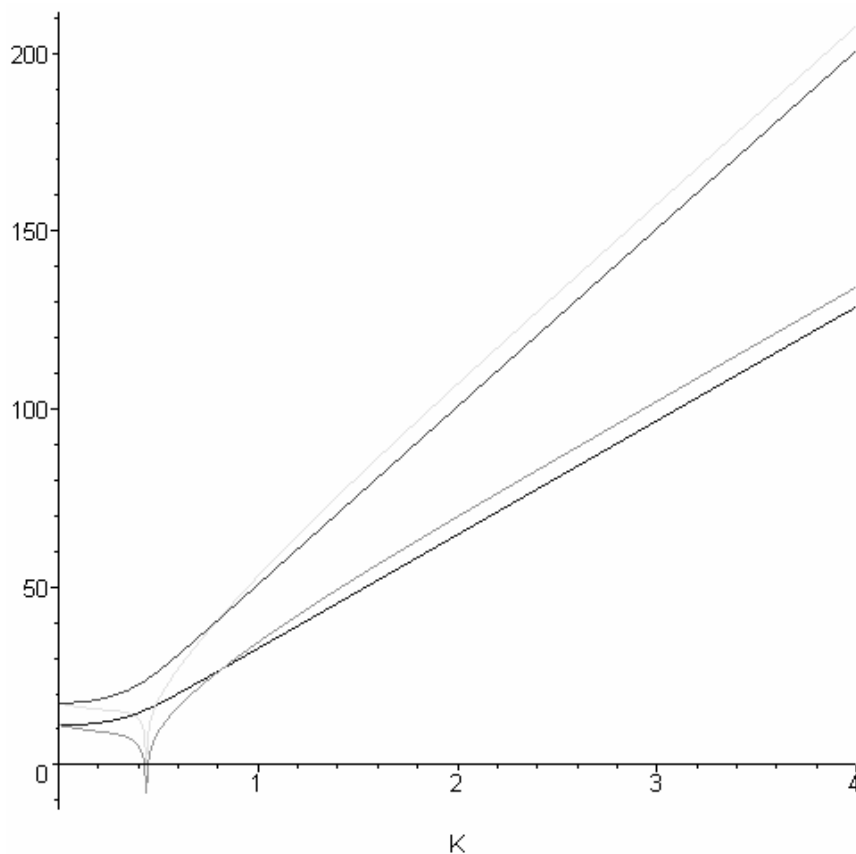


Рис. 4.7. Логарифм статсуммы для $L=4$ (внизу) и $L=5$ (вверху) квадратной решетки, вычисленный по формуле Онсагера (4.9) и точные результаты

Видно, что расхождение существенное, причем кривые качественно различны при малых K , т.е. при высоких температурах (вспомним, что $X = J / K_B T$). При больших K (низких температурах) расхождение несущественно. Программа компьютерной алгебры позволяет просматривать полученные функции с разной степенью разрешения:

```
> plot({ln(Z_[5]), ln(f5)}, K=9..10);
```

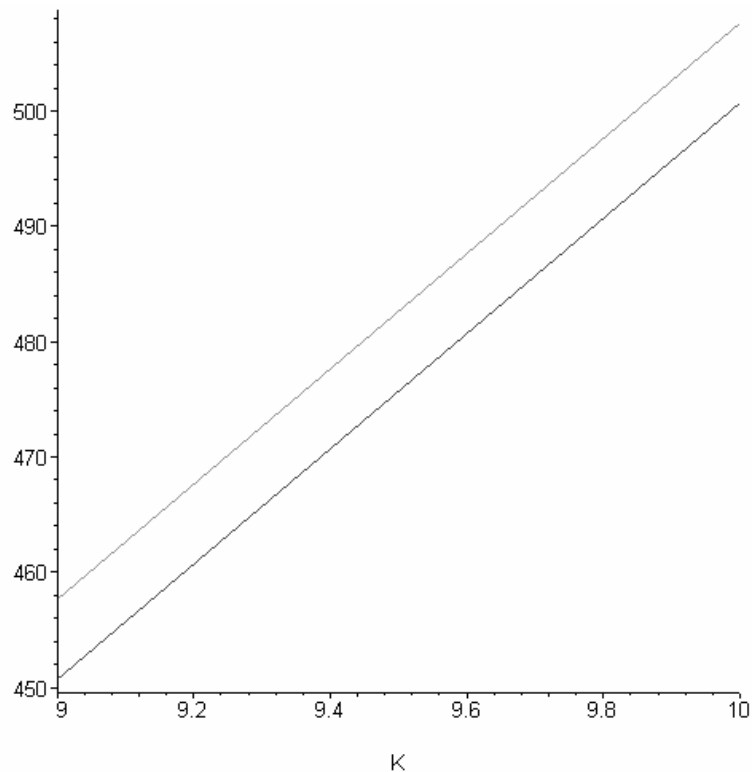


Рис. 4.8. Логарифмы статистической суммы для квадратной решетки размера $L=5$, вычисленные по формуле Онсагера (4.9) (верхняя кривая), и точный результат для низких температур

Наконец, применим нашу программу для вычисления статистической суммы квадратной решетки во внешнем магнитном поле. Как указывалось, это является одной из важнейших нерешенных проблем теоретической физики. Сделать это мы сможем, разумеется, лишь для самых малых размеров решетки. Нижеследующая программа вычисляет такую статсумму для решетки размера 3 x 3. Обратите внимание, что единственное отличие от программы без магнитного поля выделено более крупным шрифтом.

```
> restart: L:=2:
> S2:=sum(sum(sum(sum(product(product(abs(sigma[m,n]),m=1..L),n=
1..L)*ek^(sum(sum(sigma[k,l]*sigma[k,l+1]+
sigma[k,l]*sigma[k+1,l], k=1..L-1),l=1..L-1)+
sum(sigma[L,l]*sigma[L,l+1]+sigma[L,l]*sigma[1,l],l=1..L-
1)+sum(sigma[k,L]*sigma[k+1,L]+sigma[k,L]*sigma[k,l],k=1..L-
1)+sigma[L,L]*sigma[L,1]+sigma[L,L]*sigma[1,L]
```

```
) *eh^(sum(sum(sigma[k,l],k=1..L),l=1..L)),sigma[1
,1]=-1..1),sigma[1,2]=-1..1),sigma[2,1]=-1..1),sigma[2,2]=-
1..1);
```

$$S2 := \frac{ek^8}{eh^4} + \frac{4}{eh^2} + 4 + \frac{2}{ek^8} + 4eh^2 + ek^8 eh^4 .$$

Подставим вместо символических ek , eh явные выражения:

```
> f2:=subs({ek=exp(k),eh=exp(h)},S2);
```

$$f2 := \frac{(e^k)^8}{(e^h)^4} + \frac{4}{(e^h)^2} + 4 + \frac{2}{(e^k)^8} + 4(e^h)^2 + (e^k)^8 (e^h)^4$$

и построим график от двух переменных:

```
> plot3d(ln(f2),k=0..1.4,h=0..2);
```

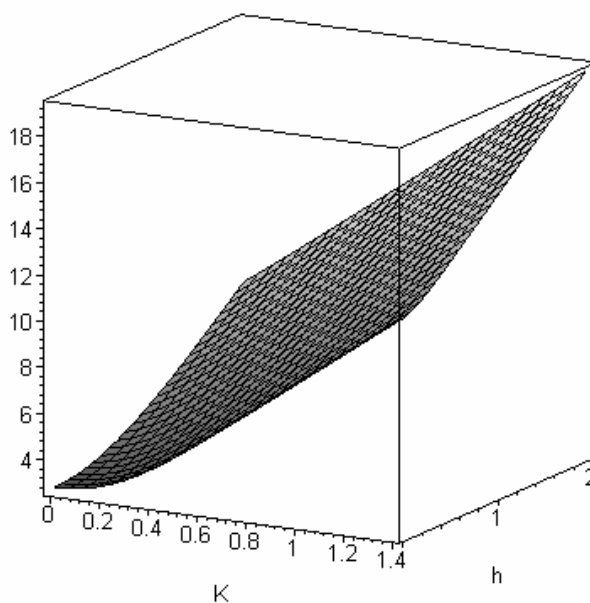


Рис. 4.9. Логарифм статистической суммы для квадратной решетки размера $L=5$ внешним магнитным полем H (здесь $h=H/k_B T$)

Аналогично для $L=3$ вычисления дают:

$$\begin{aligned}
f_3 := & \frac{18 (e^K)^6}{(e^h)^5} + \frac{6 (e^K)^6}{(e^h)^3} + \frac{45}{(e^K)^6 e^h} + \frac{45 (e^K)^2}{e^h} + \frac{36 (e^K)^2}{(e^h)^3} + \frac{36}{(e^K)^2 (e^h)^3} + \frac{(e^K)^{18}}{(e^h)^9} \\
& + \frac{36 (e^h)^3}{(e^K)^2} + \frac{9 (e^K)^{10}}{(e^h)^7} + 9 (e^K)^{10} (e^h)^7 + \frac{18 (e^K)^2}{(e^h)^5} + \frac{45 e^h}{(e^K)^6} + \frac{36}{(e^K)^2 e^h} \\
& + 6 (e^K)^6 (e^h)^3 + 45 (e^K)^2 e^h + \frac{6}{(e^K)^6 (e^h)^3} + \frac{36 e^h}{(e^K)^2} + 36 (e^K)^2 (e^h)^3 \\
& + 18 (e^K)^6 (e^h)^5 + \frac{6 (e^h)^3}{(e^K)^6} + 18 (e^K)^2 (e^h)^5 + (e^K)^{18} (e^h)^9
\end{aligned}$$

Можно вычислить так называемую спонтанную намагниченность, которую для модели Изинга обычно оценивают по формуле:

$$M_0 = \lim_{N \rightarrow \infty} \frac{1}{N} \left(\frac{\partial \ln Z}{\partial H} \right)_{H=0} \quad (4.17)$$

Нам недоступно выражение при больших N , поэтому оценим спонтанную намагниченность для конечных случаев $L=2; 3$:

```
>evalf(subs(h=0,diff(ln(f2),h)));evalf(subs(h=0,diff(ln(f3),h))
);
0.
0.
```

Таким образом, для конечных решеток с $L=2; 3$ спонтанная намагниченность отсутствует.

Можно обобщить программу для вычисления еще более интересного случая трехмерной модели Изинга, что будет проведено в лабораторной работе.

4.4. Регулярное и стохастическое движение в асимметричном вихре

4.4.1. Введение

В отличие от стандартных задач диффузии и теплопроводности, относящихся к классической математической физике, проблема модификации процессов переноса в движущихся средах является до сих пор нерешенной в полном объеме. Движение среды меняет не только эффективную

скорость диффузии, но и характер зависимости диффузионного потока (характерного смещения) от времени. Более сильная (по сравнению с обычной для стандартной диффузии корневой) зависимость от времени называется супердиффузией, а более слабая – субдиффузией. Как правило, неоднородное движение среды приводит к супердиффузии, что необходимо иметь в виду при многочисленных практических приложениях этой задачи, например, для расчетов распространения выбросов ядовитых веществ в атмосфере или загрязнений в океане, миграции примесей в плазменном или химическом реакторе и многих других.

Возможно ли течение, реализующее режим субдиффузии в отсутствие специальных «ловушек» для диффундирующих частиц, пока неизвестно. Идея постановки данной работы состояла в том, что в условиях течения по несимметричной области, когда ширина узкого слоя оказывается порядка длины свободного пробега частицы, может быть реализован и режим субдиффузии. Так это или нет, предстоит выяснить в результате проведения предлагаемого численного эксперимента.

Рассмотрим модель пробных частиц (dye particles). Частицы в этой модели испытывают воздействие среды, но не изменяют ее параметров. В общем случае поток частиц имеет вид:

$$\vec{\Gamma} = n\vec{V} - D\nabla n \quad (4.18)$$

где первое слагаемое представляет конвективный поток частиц плотности n в поле скоростей \vec{V} , второе слагаемое соответствует диффузионному переносу с коэффициентом диффузии D :

$$D = \frac{\langle (\vec{r}^i - \vec{r}_0^i)^2 \rangle}{t - t_0}, \quad (4.19)$$

где радиус-вектор \vec{r}^i (\vec{r}_0^i) обозначает положение i -ой частицы в момент времени t (t_0), а угловые скобки означают усреднение по ансамблю пробных частиц:

$$\langle \dots \rangle = \frac{1}{N} \sum_N \dots, \quad (4.20)$$

где N – полное количество пробных частиц.

4.4.2. Постановка задачи

Ставится модельная задача нахождения усредненного потока частиц (4.18), находящихся в тепловом (стохастическом) и регулярном (вихревом) движении. Движение происходит в прямоугольной области, показанной на рисунке 4.10. Регулярное движение осуществляется в виде вихря вдоль границы. В центре прямоугольника существует зона захвата (на рисунке 4.10 отмечена внутренним прямоугольником), в которой происходит только тепловое движение. Поскольку область вихря асимметрична, то согласно условию непрерывности (см. ниже), в нижней зоне (зона II на рисунке 4.10) вихревое движение осуществляется с большей скоростью, чем в верхней (зона I на рисунке 4.10). Расположение зоны захвата является геометрическим параметром задачи.

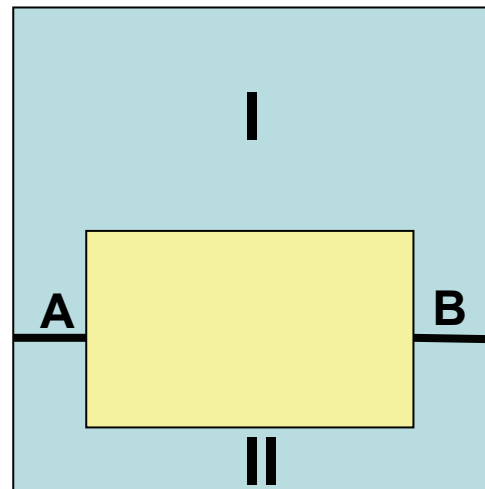


Рис. 4.10

4.4.3. Регулярное движение

Для расчета линий тока в асимметричном вихре необходимо рассчитать уравнение непрерывности. Непрерывность линий тока описывается уравнением

$$\operatorname{div} \vec{V}^r = 0,$$

которое сводится в двумерном случае к уравнению Лапласа для потенциала скорости $\varphi(x, y)$:

$$\begin{cases} \vec{V}^r = -\nabla \varphi \\ \Delta \varphi = 0 \end{cases}, \quad (4.21)$$

а компоненты скорости имеют вид:

$$\begin{cases} V_x^r = -\frac{\partial \varphi}{\partial x} \\ V_y^r = -\frac{\partial \varphi}{\partial y} \end{cases} \quad (4.22)$$

Кроме того, необходимо задать ненулевой ротор

$$\text{rot } \vec{V}^r = \text{const} \neq 0, \quad (4.23)$$

чтобы описать вихревое движение. Для того чтобы совместить уравнения (4.21) и (4.23), область вихря разбивается на две части и задача на регулярное движение решается в два этапа: для верхней половины вихря I и для нижней половины II (рисунок 4.10). По отдельности движение в этих областях будет потенциальным и описывается уравнением Лапласа (4.21) с заданными граничными условиями на границах разделения этих областей А и В на перепад функции $\varphi(x,y)$:

$$\begin{cases} \varphi_A^I - \varphi_B^I = \varphi_1, & \text{in } I \\ \varphi_A^{II} - \varphi_B^{II} = \varphi_2, & \text{in } II \end{cases} \quad (4.24)$$

Сшивка решений на границах А и В происходит с помощью подбора граничного условия (4.24) из условия непрерывности скорости (4.22). Граничные условия на функцию φ на остальных границах накладываются таким образом, что регулярное движение частиц (4.22) не выносит частицы из зоны вихря.

4.4.4. Тепловое движение

Тепловое движение описывается моделью стохастического рассеяния при столкновениях частиц на случайный угол $\alpha = \text{random}(0, 2\pi)$:

$$\begin{cases} V_x^{st} = V^{st} \cos(\alpha) \\ V_y^{st} = V^{st} \sin(\alpha) \end{cases}. \quad (4.25)$$

Частота столкновений и амплитуда теплового движения (температура) являются параметрами задачи. Полная скорость представляется в виде:

$$\vec{V} = \vec{V}^r + \vec{V}^{st}. \quad (4.26)$$

Конвективный член в (4.18) зануляется, т.к. средние по времени значения $\langle \vec{V} \rangle = \langle \vec{V}^r + \vec{V}^{st} \rangle = 0$, и остается лишь диффузионный член. При этом расчетное время должно в несколько раз превышать время оборота частицы в вихревом движении.

4.4.5. Геометрия системы и движение частиц

Зона движения состоит из двух прямоугольников, один из которых вложен в другой, как показано на рисунке 4.11. Вихрь вращается внутри большего прямоугольника $b_x \times b_y$, не заходя внутрь меньшего. Т.е. во внешности малого прямоугольника, происходит и регулярное вихревое движение, и тепловое, а внутри него происходит только тепловое движение.

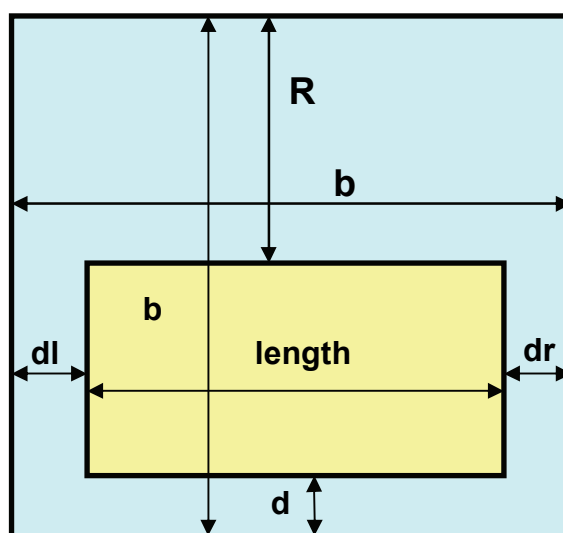


Рис. 4.11

Обозначения, приведенные на рисунке 4.11, используются в расчетной программе. Для описания движения многих частиц и расчета эффективной диффузии необходимо задавать источники и стоки на границах. По горизонтальной координате накладываются периодические граничные условия, т.е. потеря частицы с одной стороны приводит к появлению новой частицы с другой стороны. По вертикальной координате накладывается граничное условие постоянного потока. Для определенности средний градиент плотности частиц направлен снизу вверх, а поток частиц сверху вниз.

4.4.6. Программная реализация

Программа **DifAV** (Diffusion on Asymmetric Vortex) реализована в системе MATLAB с помощью инструментария PDEtools. Расчет диффузии состоит из двух этапов. **На первом этапе** рассчитывается уравнение Лапласа (4.18) в заданной геометрии. Геометрические параметры задачи (рис. 4.11), количество точек сетки N и граничное условие (4.21) для уравнения Лапласа для верхней $\Delta\varphi_{up}$ (φ_1) и нижней $\Delta\varphi_{down}$ (φ_2) областей движения можно задавать на панели управления под именем vortex (см. рис. 4.12). Решение представляется двумерной функцией $\varphi(x,y)$, линии уровня которой также показаны на панели vortex (рис. 4.12). Для улучшения сходимости решения уравнения Лапласа (4.18) острые углы внутреннего прямоугольника сглажены окружностью радиуса 0.05 в единицах рисунка ($b_x=b_y=1$). Частота линий уровня характеризует скорость частицы, согласно системе (4.19). Регулярное движение, таким образом, в верхней широкой части вихря медленное, а в нижней части и по бокам быстрое.

На втором этапе рассчитывается движение частиц в полях (4.19) и (4.22). Параметры расчета можно менять на панели particles (рис. 4.13). На этой панели выдается также количество частиц (particles number), расчетное время (Time) и положение частиц на рисунке линий уровня вихря. Регулярное и тепловое движения можно независимо включать и выключать.

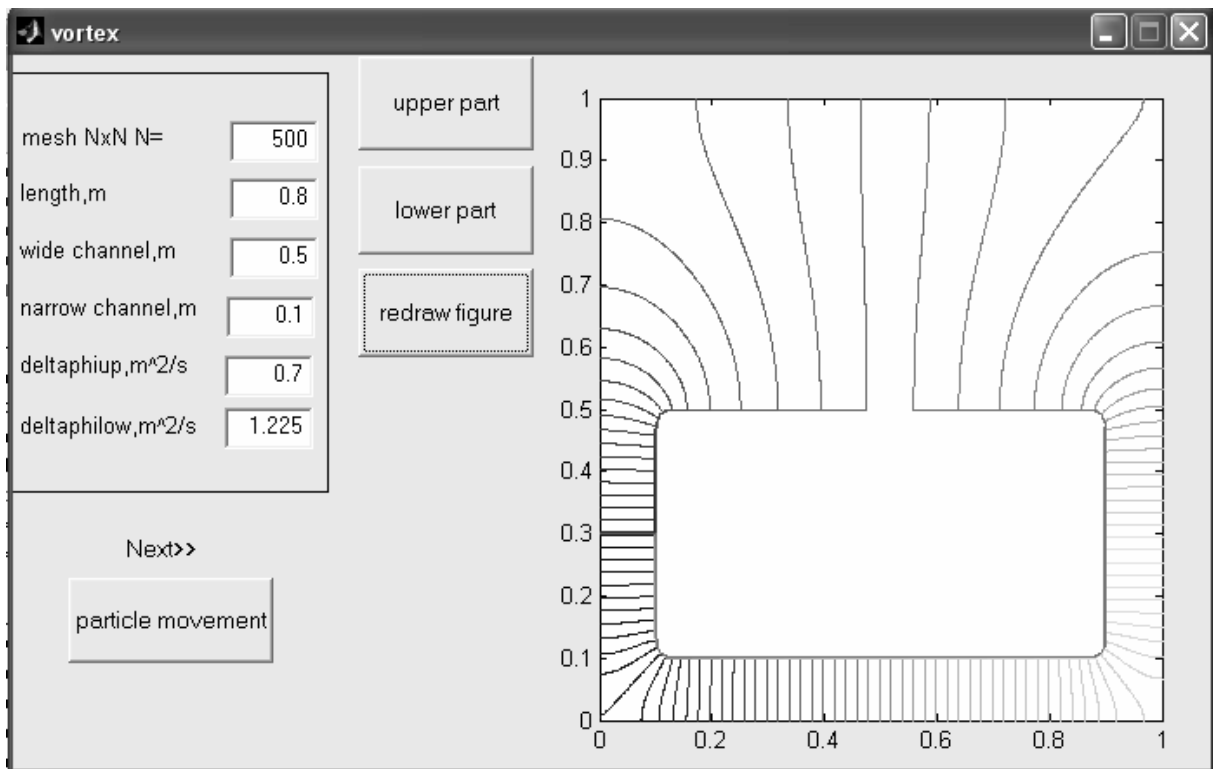


Рис. 4.12

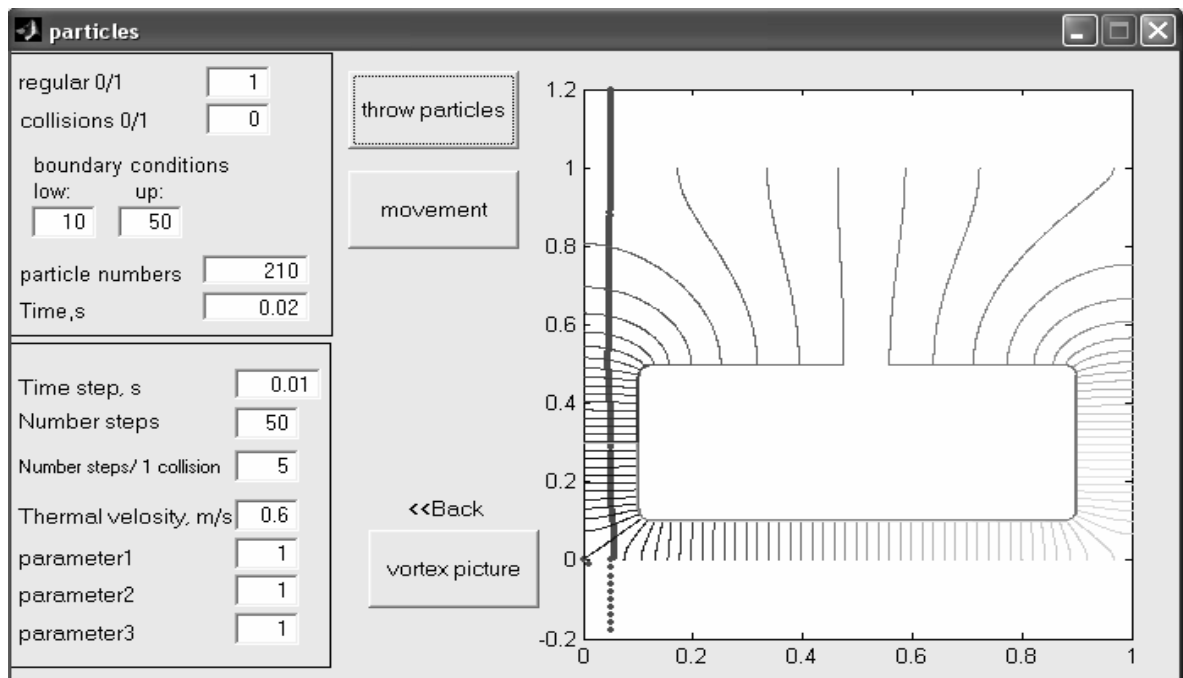


Рис. 4.13

Список литературы к теме 4

Использованная литература:

1. Броуновское движение
physics.nad.ru/Physics/Cyrillic/brow_txt.htm
2. Прикладная комбинаторная математика. Сборник статей под редакцией Э.Беккенбаха. – М.: Мир, 1968. – 362 с.
3. Ландау Л.Д., Лифшиц Е.М. Статистическая физика. Часть 1. – М.: Наука, 1976.
4. Castening B. Pedestrian Solution of the Two-Dimensional Ising Model
<http://xxx.lanl.gov/cond-mat/0104398v5>
5. Castening B. Symplifying Kaufman’s Solution of the Two-Dimensional Ising Model
<http://xxx.lanl.gov/cond-mat/0111380v2>

Рекомендованная литература:

1. Васильев А.Н. Maple 8. Самоучитель. – М.: Диалектика, Вильямс, 2003.
2. Алексеев Е.Р., Чеснокова О.В. Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9. – М.: Изд-во «НТ Пресс», 2006. – 496 с.
3. Гандер В., Гржебичек И. Решение задач в научных вычислениях с применением Maple и MATLAB. – М.: Изд-во «Вассамедина», 2005. – 520 с.

Интернет-ресурсы к теме 4

Броуновское движение

physics.nad.ru/Physics/Cyrillic/brow_txt.htm

elementy.ru/trefil/30

ru.wikipedia.org/wiki/Броуновское_движение

Решение для двумерной модели Изинга

<http://xxx.lanl.gov/cond-mat/0104398v5>

Контрольные вопросы к теме 4

1. Напишите программу сравнения быстродействия вычислений факториалов точно и с помощью формулы Стирлинга в Maple.
2. То же в программах MATLAB и Mathematica. Сравните быстродействие трех программ.
3. Сравните быстродействие трех программ, решая одну и ту же краевую задачу для какого-либо гипергеометрического уравнения.
4. С помощью генератора случайных чисел смоделируйте распределение Пуассона и нормальное распределение.
5. Приведите пример, когда неправильное положение оператора посева последовательности случайных чисел *randomize* приводит к абсолютным повторам в вычислениях.
6. Видоизмените программу о случайных блужданиях, предположив, что вероятность удара со стороны молекулы распределена во времени по Пуассону.
7. Напишите программу для вычисления точной статистической суммы модели Изинга для кубической решетки размером $3 \times 3 \times 3$.
8. То же во внешнем магнитном поле.
9. С помощью пакета CurveFitting в Maple подберите аппроксимации для кривых Онсагера, изображенных на рисунке 4.5.

Тема 5. Пример моделирования в MATLAB технической системы: модель алгоритма оценивания ошибок ИНС и GPS на основе фильтра Кальмана

5.1. Постановка проблемы и обзор математических средств, пригодных для решения поставленной задачи

Инерциальные навигационные системы составляют ядро навигационных комплексов современных самолетов в силу своей универсальности и автономности.

Инерциальные системы по сравнению с радионавигационными системами, обладают следующими преимуществами [8]:

- непрерывность определения местоположения и скорости подвижного объекта;
- абсолютная помехоустойчивость;
- практически неограниченная область применения;
- слабая зависимость точности навигационной информации от маневрирования подвижного объекта;
- максимальный объем информации (местоположение, путевая скорость, направление и местная вертикаль);
- высокая точность измерения курса и местной вертикали; автономность.

Принципиальный недостаток ИНС состоит в том, что в результате неизбежных инструментальных (технологических) погрешностей изготовления инерциальных датчиков со временем накапливается ошибка в определении местоположения подвижного объекта.

Это приводит к технико-экономической целесообразности коррекции ИНС с помощью способов другой физической природы, не страдающих главным пороком ИНС – накоплением погрешности со временем. Причем коррекция может быть периодической, а не непрерывной.

Здесь рассматривается метод повышения точности ИНС на основе использования позиционной и скоростной информации от GPS–ГЛОНАСС приёмника. Подобная задача активно изучается в настоящее время многими авторами, например в [9], где реализованы алгоритмы оценивания ошибок в определении координат и прогноза этих ошибок в случае пропадания сигналы системы GPS–ГЛОНАСС, а также предложена методика коррекции углов ориентации на основе учета моделей ошибок и использования вариантов фильтра Кальмана. Анализ экспериментов, выполненных в этой работе, показал, что адаптивный алгоритм сглаживания скачков показаний системы GPS–ГЛОНАСС позволяет на основе скалярного алгоритма прогноза поддерживать навигацию в случае пропадания сигнала системы GPS–ГЛОНАСС длительностью до 30 сек. И что такой подход более эффективен, чем традиционный фильтр Кальмана.

Существуют два основных подхода к комплексированию сигналов ИНС и GPS–ГЛОНАСС – централизованный (замкнутый) метод и каскадный (разомкнутый) метод.

При реализации первого метода на вход алгоритма оценивания поступают от GPS–ГЛОНАСС информация о расстоянии между приемником и спутниками, а также скорость ее изменения. Кроме того, от ИНС также поступает информация о текущих навигационных параметрах.

Алгоритм оценивания (фильтр Кальмана) реализует совместную модель ошибок ИНС и GPS–ГЛОНАСС для определения текущих навигационных параметров. Основным преимуществом централизованного подхода является возможность расширения полосы пропускания приемника GPS–ГЛОНАСС без существенного увеличения шумов измерений. Кроме того, коррекция может осуществляться по измерениям только одного спутника.

При коррекции по схеме каскадного типа в качестве измерений используются параметры, полученные от GPS: координаты и скорость движения объекта. В отличие от первого метода комплексирования, в этом

подходе необходимо осуществлять решение навигационной задачи непосредственно в приемнике GPS, что требует необходимость обозрения как минимум 4-х спутников.

При сравнении этих двух подходов следует отметить, что в случае применения комплексирования для подвижных объектов, особенно высокоманевренных, вероятность потери нескольких спутников является большой. То есть количество спутников может быть меньше четырех. В этом случае первый тип комплексирования имеет преимущество, потому что может осуществлять коррекцию и в случае измерения, поступающего только с одного спутника. Однако время пропадания спутников для такого рода объектов составляет только несколько секунд, в течение которых возможно осуществлять прогноз поведения ошибок ИНС. Использование каскадной схемы позволяет сформировать удобный простой и надежный алгоритм оценивания, удобный для реализации в бортовом вычислителе. Таким образом, наиболее целесообразным является использование каскадной схемы комплексирования.

Наиболее распространенным методом оценивания вектора состояния является фильтр Кальмана. В постановке задачи оптимальной фильтрации требуется, чтобы входные и измерительные шумы являлись белыми. В случае цветного шума используются формирующие фильтры, на вход которых поступает белый шум.

Для реализации фильтра Кальмана требуется априорная информация о корреляционных матрицах входных и измерительных шумов (Q и R). Однако в реальных приложениях такая информация, как правило, задана неточно. Это может привести к расходимости алгоритма фильтрации.

Реализация вычислительной схемы по алгоритму Кальмана является сложной при больших значениях размерности вектора состояния [10]. Чтобы предотвратить указанные трудности, разработана целая серия модифи-

каций фильтра Кальмана. Так, для сокращения объема вычислений предложено использовать редуцированный фильтр Кальмана.

Существует ряд модификаций фильтра Кальмана, позволяющих осуществлять оценку вектора состояния в отсутствии точной априорной информации о статистике входных и измерительных шумов (Q и R).

Такие алгоритмы называют адаптивными фильтрами Кальмана [10, 11, 12]. Для построения адаптивных фильтров можно использовать свойства обновляемой последовательности.

Существующие в литературе алгоритмы фильтрации работоспособны при больших неопределенностях априорной информации. Использование адаптивных фильтров Кальмана на базе анализа ковариационной матрицы обновляемой последовательности целесообразно для случая появления непредсказуемых скачков измерительного шума GPS–ГЛОНАСС в случае резкой смены «созвездия» наблюдаемых спутников.

Определим ограничение на точность, оценивания ее в случае использования фильтра Кальмана. Точность оценивания ограничена уровнем входных шумов в описании модели системы. Таким образом, основное направление в повышении точности оценивания при использовании фильтра Кальмана зависит от представления модели системы в более детерминированной форме. Однако такое представление возможно при абсолютном знании протекающих в системе процессов, что является на практике редким явлением и зависит от масштаба времени описания модели.

Кроме того, возможно использовать скалярный подход, позволяющий осуществить оценивание вектора состояния индивидуально для каждой компоненты. Скалярный алгоритм является менее чувствительным в отношении точности описания математической модели и статистических характеристик входного шума, благодаря адаптивной зависимости коэффициента усиления от текущих ошибок оценивания [12].

Перед тем как осуществить выбор алгоритма фильтрации и комплексирования, проведем более подробное описание существующих алгоритмов.

5.2. Алгоритмы адаптивной фильтрации

При поиске оптимальных алгоритмов обработки сигнала неизбежно приходится опираться на некоторые *статистические модели* сигналов и шумов. Чаще всего при формировании этих моделей используются концепции линейности, стационарности и нормальности (гауссовости). Однако перечисленные принципы далеко не всегда выполняются на практике, а от адекватности выбранной модели в значительной мере зависит качество приема сигнала. Возможным решением проблемы является использование адаптивных фильтров, которые позволяют системе подстраиваться под статистические параметры входного сигнала, не требуя при этом задания каких-либо моделей.

Появившись в конце 1950-х годов, адаптивные фильтры прошли большой путь, превратившись из экзотической технологии, применявшейся преимущественно в военных целях, в «ширпотреб», без которого сейчас была бы немыслима работа модемов, сотовых телефонов и многого другого. Существует большое количество адаптивных алгоритмов, различающихся вычислительной сложностью, особенностями поведения, используемыми исходными данными и структурами самих адаптирующихся систем. Здесь рассмотрим несколько основных алгоритмов, более подробные сведения можно найти в [1–4].

Общая структура адаптивного фильтра показана на рисунке 1. Входной дискретный сигнал $x(k)$ обрабатывается дискретным фильтром, в результате чего получается выходной сигнал $y(k)$. Этот выходной сигнал сравнивается с *образцовым* сигналом $d(k)$, разность между ними образует сигнал ошибки $e(k)$. Задача адаптивного фильтра – минимизировать ошиб-

ку воспроизведения образцового сигнала. С этой целью блок адаптации после обработки каждого отсчета анализирует сигнал ошибки и дополнительные данные, поступающие из фильтра, используя результаты этого анализа для подстройки параметров (коэффициентов) фильтра.

Возможен и иной вариант адаптации, при котором образцовый сигнал не используется. Такой режим работы называется *слепой адаптацией* (blind adaptation) или *обучением без учителя* (unsupervised learning). В этом случае требуется некоторая информация о структуре полезного входного сигнала (например, знание типа и параметров используемой модуляции) и при этом очевидно, что слепая адаптация является более сложной вычислительной задачей, чем адаптация с использованием образцового сигнала. Может показаться, что алгоритмы с использованием образцового сигнала лишены практического смысла, поскольку выходной сигнал должен быть заранее известен. Однако есть целый ряд практических задач, при решении которых образцовый сигнал оказывается доступен, а у нас при комплексировании таковым мы будем считать сигнал GPS и его экстраполяцию, в случае отсутствия GPS-сигнала. В ряде случаев полезным сигналом является не выходной сигнал фильтра, а *сигнал ошибки*, то есть разность между образцовым сигналом и выходным сигналом адаптивного фильтра.

В качестве фильтра в структуре, показанной на рисунке 5.1, используются либо нерекурсивный (нерекурсивный – неизменяемый от такта к такту) либо рекурсивный цифровые фильтры. Алгоритм адаптации вносит в систему обратную связь, вследствие чего адаптивная система в целом может стать неустойчивой. Еще один класс адаптивных систем – *нейронные сети* (neural networks), которые в определенной степени моделируют работу нервной системы живых организмов.

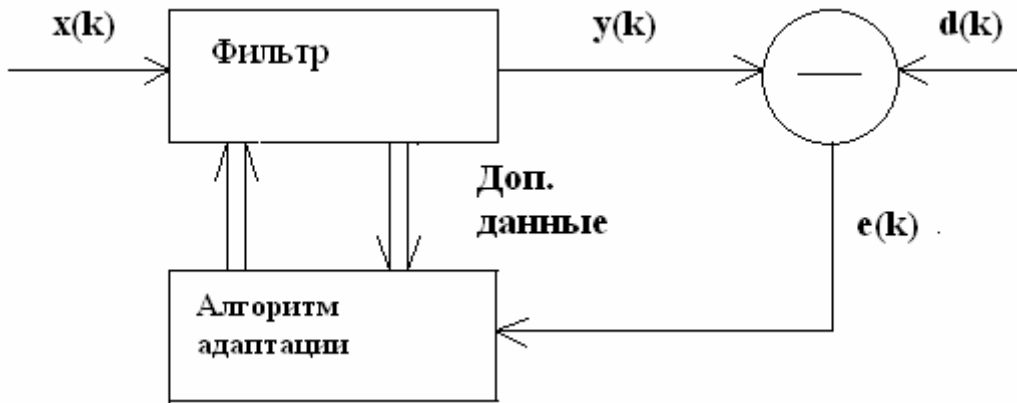


Рис. 5.1. Общая структура адаптивного фильтра

Здесь рассмотрим несколько адаптивных алгоритмов с использованием образцового сигнала, часто применяемых на практике в различных системах обработки информации. Для упрощения математических выкладок предположим, что сигналы и фильтры являются вещественными. Однако результирующие формулы легко обобщаются на случай комплексных сигналов и фильтров.

5.3. Оптимальный фильтр Винера

Подход к задаче оптимальной фильтрации может быть как статистическим, так и детерминированным. Сначала рассмотрим статистический вариант.

Пусть входной дискретный, случайный сигнал $x(k)$ обрабатывается не рекурсивным дискретным фильтром порядка N , коэффициенты которого могут быть представлены вектор-столбцом $w = [w_0, w_1, \dots, w_N]^T$. Здесь символ T означает транспонирование вектора-строки в вектор-столбец. Теперь выходной сигнал фильтра можно записать

$$y(k) = u^T(k)w, \quad (5.1)$$

где $u(k) = [x(k), x(k - 1), \dots, x(k - N)]^T$ – вектор-столбец содержимого линии задержки фильтра на k -м шаге.

Кроме того, имеется образцовый (также случайный) сигнал $d(k)$. Ошибка воспроизведения образцового сигнала равна

$$e(k) = d(k) - y(k) = d(k) - u^T(k)w \quad (5.2)$$

Необходимо найти такие коэффициенты фильтра w , которые обеспечивают максимальную близость выходного сигнала фильтра к образцовому, т.е. минимизируют ошибку $e(k)$. Поскольку $e(k)$ также является случайным процессом, в качестве меры ее величины разумно принять средний квадрат. Таким образом, оптимизируемый функционал выглядит так:

$$J(w) = \overline{e^2(k)} \rightarrow \min$$

(Здесь черта означает статистическое осреднение дискретной, случайной величины, т.е. вычисление математического ожидания или средневзвешенной суммы из $N+1$ слагаемых.) Квадрат ошибки равен

$$e^2(k) = (d(k) - u^T(k)w)^2 = d^2(k) - 2d(k)u^T(k)w + w^T u(k)u^T(k)w$$

(конечно, здесь операции выполняются с помощью скалярного произведения векторов).

Статистически усредняя это выражение, получаем следующее:

$$J(w) = \overline{e^2(k)} = \overline{d^2(k)} - \overline{2d(k)u^T(k)w} + \overline{w^T u(k)u^T(k)w} \quad (5.3)$$

Входящие в полученную формулу усредненные величины имеют следующий смысл:

$\overline{d^2(k)} = \sigma_d^2$ – средний квадрат образцового сигнала;

$\overline{d(k)u^T(k)} = p^T$ – транспонированный вектор-столбец p взаимных корреляций между k -м отсчетом образцового сигнала и содержимого линии задержки фильтра. Если рассматриваемые случайные процессы $x(k)$ и $d(k)$ являются совместно стационарными, вектор взаимных корреляций не зависит от номера шага k ;

$\overline{u(k)u^T(k)} = R$ – корреляционная матрица сигнала, имеющая размер $(N + 1) \times (N + 1)$.

Для стационарного, случайного процесса корреляционная матрица имеет вид матрицы Теплица, т.е. на ее диагоналях стоят одинаковые величины:

$$R = \begin{bmatrix} R_x(0) & R_x(1) & \dots & R_x(N) \\ R_x(1) & R_x(0) & \dots & R_x(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ R_x(N) & R_x(N-1) & \dots & R_x(0) \end{bmatrix}$$

Здесь $R_x(m) = \overline{x(k)x^T(k-m)}$ – корреляционная функция (КФ) случайного процесса $\{x(k)\}$.

С учетом введенных обозначений (5.3) принимает следующий вид

$$J(w) = \sigma_d^2 - 2p^T w + w^T R w. \quad (5.4)$$

Данное выражение представляет собой квадратичную форму относительно w и потому для положительно определенной матрицы R , что чаще всего имеет место, функция (5.4) имеет единственный минимум, для нахождения которого необходимо приравнять нулю вектор градиента:

$$\text{grad } J(w) = -2p + 2Rw = 0.$$

Отсюда получаем искомое решение для оптимальных коэффициентов фильтра:

$$w = R^{-1} p. \quad (5.5)$$

Такой фильтр называется **фильтром Винера**. Подстановка (5.5) в (5.4) дает минимально достижимую дисперсию сигнала ошибки:

$$\overline{e^2(k)}_{\min} = \sigma_d^2 - p^T R^{-1} p. \quad (5.6)$$

Несложно также показать, что $\overline{e(k)y(k)} = 0$ и $\overline{e(k)x(k)} = 0$. Т.е. сигнал ошибки для фильтра Винера не коррелирован с входным и выходным сигналами фильтра.

5.4. Алгоритм LMS

Один из наиболее распространенных адаптивных алгоритмов основан на поиске минимума целевой функции (5.3) методом наискорейшего спуска (вариант метода градиентного спуска при выборе наилучшего шага на каждой итерации спуска). При использовании данного способа оптимизации вектор коэффициентов фильтра $w(k)$ должен рекурсивно обновляться следующим образом:

$$w(k+1) = w(k) - \frac{\mu}{2} \text{grad}J(w(k)) = w(k) + \mu p - \mu R w(k) \quad (5.7)$$

где μ – положительный коэффициент, называемый размером шага. Подробный анализ сходимости данного процесса приведен, например, в [6]. Скорость сходимости зависит от разброса собственных чисел корреляционной матрицы R – чем меньше отношение наибольшего и наименьшего ее собственных значений, тем быстрее сходится итерационный процесс.

Для расчета градиента необходимо знать значения матрицы R и вектора p . Но на практике могут быть доступны лишь *оценки* этих значений, получаемые по входным данным. Простейшими такими оценками являются *мгновенные* значения корреляционной матрицы и вектора взаимных корреляций, получаемые без какого-либо усреднения:

$$\begin{aligned} \hat{R}(k) &= u(k)u^T(k), \\ \hat{p}(k) &= d(k)u(k) \end{aligned}$$

При использовании данных оценок формула (5.7) принимает следующий вид:

$$w(k+1) = w(k) + \mu d(k)u(k) - \mu u(k)u^T(k)w(k) = w(k) + \mu u(k)(d(k) - u^T(k)w(k)) \quad (5.8)$$

Выражение, стоящее в скобках, согласно (5.2) представляет собой разность между образцовым сигналом и выходным сигналом фильтра на k -м шаге, т.е. ошибку фильтрации $e(k)$. С учетом этого выражение для рекурсивного обновления коэффициентов фильтра оказывается очень простым:

$$w(k+1) = w(k) + \mu u(k)e(k) \quad (5.9)$$

Алгоритм адаптивной фильтрации, основанный на формуле (9), получил название LMS (Least Mean Square, метод наименьших квадратов). Можно получить ту же формулу и несколько иным образом: используя вместо градиента *статистически усредненного* квадрата ошибки $\overline{e^2(k)}$ градиент его *мгновенного значения* $e^2(k)$.

Основным достоинством алгоритма LMS является предельная вычислительная простота – для подстройки коэффициентов фильтра на каждом шаге нужно выполнить $N + 1$ пар операций «умножение–сложение». Платой за простоту является медленная сходимость и повышенная, по сравнению с минимально достижимым значением (6), дисперсия ошибки в установившемся режиме – коэффициенты фильтра всегда флуктуируют вокруг оптимальных значений (5), что и увеличивает уровень выходного шума.

Существует большое число модификаций алгоритма LMS, направленных на ускорение сходимости либо на уменьшение числа арифметических операций. Ускорение сходимости может быть достигнуто за счет улучшения используемой оценки градиента, а также за счет преобразования входного сигнала с целью сделать его отсчеты некоррелированными. Уменьшение вычислительной сложности может быть достигнуто, в частности, за счет использования в (5.3) не самих сигнала ошибки и содержимого линии задержки фильтра, а лишь их *знаков*. Это позволяет полностью избавиться от операций умножения при обновлении коэффициентов фильтра. В целом следует отметить, что требования ускорения сходимости и сокращения вычислительных затрат являются противоречивыми.

5.5. Детерминированная задача оптимизации

Рассматривая статистическую задачу оптимизации, мы считали входной сигнал *случайным процессом* и минимизировали *средний квадрат*

ошибки воспроизведения образцового сигнала. Однако возможен и иной подход, не использующий статистические методы.

Пусть, как и раньше, обработке подвергается последовательность, состоящая из K отсчетов $x(k)$, коэффициенты нерекурсивного фильтра образуют вектор-столбец w , а отсчеты образцового сигнала равны $d(k)$. Выходной сигнал фильтра определяется формулой (5.1), а ошибка воспроизведения образцового сигнала – формулой (5.2).

Теперь оптимизационная задача формулируется так: необходимо найти такие коэффициенты фильтра w , чтобы норма ошибки воспроизведения образцового сигнала была минимальной:

$$J(w) = \sum_{k=0}^{K-1} |e(k)|^2 \rightarrow \min. \quad (5.10)$$

Для решения задачи в выражениях (5.1) и (5.2) перейдем к матричной записи, получая формулы для векторов-столбцов выходного сигнала y и для ошибки воспроизведения входного сигнала e :

$$y = U^T w, \quad e = d - U^T w. \quad (5.11)$$

Здесь d – вектор-столбец отсчетов образцового сигнала, а U – матрица, столбцы которой представляют собой содержимое линии задержки фильтра на разных тактах

$$U = [u(0), u(1), \dots, u(K-1)].$$

Выражение (5.10) для нормы ошибки можно переписать в матричном виде следующим образом:

$$J(w) = e^T e \rightarrow \min. \quad (5.12)$$

Подставляя (5.11) в (5.12), получаем

$$J(w) = (d - U^T w)^T (d - U^T w) = d^T d - w^T U d - d^T U^T w + w^T U U^T w.$$

Для нахождения минимума необходимо вычислить градиент данного функционала и приравнять его нулю:

$$\text{grad } J(w) = -2U d + 2U U^T w = 0.$$

Отсюда легко получается искомое оптимальное решение:

$$w = (UU^T)^{-1}Ud. \quad (5.13)$$

В формуле (5.13) прослеживается близкое родство с формулой (5.5), описывающей оптимальный в статистическом смысле фильтр Винера. Действительно, если учесть, что $(UU^T)^{-1}/K$ дает оценку корреляционной матрицы сигнала, полученную по одной реализации сигнала путем временного усреднения, а Ud/K является аналогичной оценкой взаимных корреляций между образцовым сигналом и содержимым линии задержки фильтра, то формулы (5.5) и (5.13) совпадут.

5.6. Алгоритм RLS

В принципе, в процессе приема сигнала можно на каждом очередном шаге пересчитывать коэффициенты фильтра непосредственно по формуле (5.13), однако это связано с неоправданно большими вычислительными затратами. Действительно, размер матрицы U постоянно увеличивается и, кроме того, необходимо каждый раз заново вычислять обратную матрицу $(UU^T)^{-1}$. Сократить вычислительные затраты можно, если заметить, что на каждом шаге к матрице U добавляется лишь один новый столбец, а к вектору d – один новый элемент. Это дает возможность организовать вычисления *рекурсивно*. Соответствующий алгоритм называется *рекурсивным методом наименьших квадратов* (Recursive Least Square, RLS).

Подробный вывод формул, описывающих алгоритм RLS, можно найти, например, в [2, 3], здесь, следуя [4], приведем лишь основные идеи. При использовании алгоритма RLS производится рекурсивное обновление оценки обратной корреляционной матрицы $P = (UU^T)^{-1}$, а вывод формул основывается на следующем матричном тождестве:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1},$$

где A и C – квадратные невырожденные матрицы (необязательно одинаковых размеров), а B и D – матрицы совместимых размеров. Применение этой формулы для рекурсивного обновления обратной корреляционной

матрицы P в сочетании с исходной формулой (5.13) для коэффициентов оптимального фильтра дает следующую последовательность шагов адаптивного алгоритма RLS.

1. При поступлении новых входных данных $u(k)$ производится фильтрация сигнала с использованием текущих коэффициентов фильтра $w(k-1)$ и вычисление величины ошибки воспроизведения образцового сигнала:

$$y(k) = u^T(k)w(k-1); \quad (5.14)$$

$$e(k) = d(k) - y(k).$$

2. Рассчитывается вектор-столбец *коэффициентов усиления* (следует отметить, что знаменатель дроби в следующих двух формулах является скаляром, а не матрицей):

$$K(k) = \frac{P(k-1)u(k)}{1 + u^T(k)P(k-1)u(k)}. \quad (5.15)$$

3. Производится обновление оценки обратной корреляционной матрицы сигнала:

$$P(k) = P(k-1) - K(k)u^T(k)P(k-1). \quad (5.16)$$

4. Наконец, производится обновление коэффициентов фильтра

$$w(k) = w(k-1) + K(k)e(k). \quad (5.17)$$

Начальное значение вектора w обычно принимается нулевым, а в качестве исходной оценки матрицы P используется диагональная матрица вида $CE\sigma^2$, где скаляр $C \gg 1$ (в [2] рекомендуется $C \geq 100$).

В критериях ошибок (5.10) и (5.12) значениям ошибки на всех временных тактах придается одинаковый вес. В результате, если статистические свойства входного сигнала со временем изменяются, это приводит к ухудшению качества фильтрации. Чтобы дать фильтру возможность отслеживать нестационарный входной сигнал, можно применить в (5.10)

экспоненциальное забывание, при котором вес прошлых значений сигнала ошибки экспоненциально уменьшается

$$J(w) = \sum_{i=0}^{K-1} \lambda^{K-1-i} |e(k)|^2, 0 < \lambda \leq 1.$$

В этом случае формулы (15), (16) принимают следующий вид

$$K(k) = \frac{P(k-1)u(k)}{\lambda + u^T(k)P(k-1)u(k)};$$

$$P(k) = \frac{1}{\lambda} (P(k-1) - K(k)u^T(k)P(k-1)).$$

Главным достоинством алгоритма RLS является быстрая сходимость. Однако достигается это за счет значительно более высокой (по сравнению с алгоритмом LMS) вычислительной сложности. Согласно [2] при оптимальной организации вычислений для обновления коэффициентов фильтра на каждом такте требуется $(2.5N^2 + 4N)$ пар операций «умножение–сложение».

5.7. Алгоритм Кальмана

Прежде чем рассматривать применение алгоритма Кальмана для решения задачи адаптивной фильтрации, напомним формулировку задачи фильтрации случайного процесса с известными динамическими свойствами, для решения которой фильтр Кальмана изначально предназначен. Цель фильтра Кальмана – минимизировать дисперсию оценки векторного дискретного случайного процесса $x(k)$, изменяющегося во времени следующим образом:

$$x(k+1) = \Phi(k)x(k) + v(k),$$

где $\Phi(k)$ – матрица перехода, $v(k)$ – случайный вектор (шум процесса), имеющий нормальное распределение с корреляционной матрицей $Q_p(k)$.

Для наблюдения доступен линейно преобразованный процесс $y(k)$, к которому добавляется шум наблюдения

$$y(k) = H(k)x(k) + m(k),$$

где $H(k)$ – матрица наблюдения, $m(k)$ – шум наблюдения, представляющий собой случайный вектор, имеющий нормальное распределение с корреляционной матрицей $Q_m(k)$.

Поиск алгоритма для рекурсивного обновления оценки процесса $\hat{x}(k)$ дает следующую последовательность формул:

$\hat{y}(k) = C(k)\Phi(k)\hat{x}(k-1)$ – прогнозируемое значение наблюдаемого сигнала;

$e(k) = y(k) - \hat{y}(k)$ – невязка между прогнозируемым и реально наблюдаемым значениями;

$K(k) = P(k-1)C^T(k)(C(k)P(k-1)C^T(k) + Q_m(k))^{-1}$ – калмановский коэффициент усиления;

$\hat{x}(k) = \Phi(k)\hat{x}(k-1) + K(k)e(k)$ – обновление оценки процесса $x(k)$;

$P(k) = \Phi(k)[P(k-1) - K(k)C(k)P(k-1)]\Phi^T(k) + Q_m(k)$ – обновление оценки корреляционной матрицы ошибок фильтрации.

При использовании фильтра Кальмана для решения задачи адаптивной фильтрации отслеживаемым процессом является *вектор коэффициентов оптимального фильтра w* .

Предположим, что детерминированных изменений коэффициентов не происходит, поэтому матрица перехода Φ является единичной: $\Phi(k) = E$. В качестве матрицы наблюдения выступает вектор содержимого линии задержки фильтра $u(k)$. Таким образом, выходной сигнал фильтра представляет собой прогнозируемое значение наблюдаемого сигнала, а в качестве самого наблюдаемого сигнала выступает образцовый сигнал адаптивного фильтра $d(k)$. Шум наблюдения в данном случае является ошибкой воспроизведения образцового сигнала, а матрица Q_m превращается в скалярный параметр – средний квадрат сигнала ошибки. Как отмечается в [3], величина этого параметра слабо влияет на поведение алгоритма. Рекомендуемые значения – $(0.001...0.01)\sigma_d^2$.

Если фильтруется стационарный случайный процесс, коэффициенты оптимального фильтра являются постоянными и можно принять $Q_m = 0$. Чтобы дать фильтру возможность отслеживать медленные изменения статистики входного сигнала, в качестве Q_m может использоваться диагональная матрица.

В результате приведенные выше формулы принимают следующий вид

$y(k) = u^T(k) \hat{w}(k-1)$ – выходной сигнал фильтра (прогнозируемое значение образцового сигнала);

$e(k) = d(k) - y(k)$ – ошибка фильтрации;

$K(k) = \frac{P(k-1)u(k)}{u^T(k)P(k-1)u(k) + Q_m}$ – кальмановский коэффициент усиления;

$\hat{w}(k) = \hat{w}(k-1) + K(k)e(k)$ – обновление оценки коэффициентов фильтра $w(k)$;

$P(k) = P(k-1) - K(k)u^T(k)P(k-1) + Q_p$ – обновление оценки корреляционной матрицы ошибок оценивания.

Начальное значение вектора w обычно принимается нулевым, а в качестве исходной оценки матрицы P используется диагональная матрица вида CE , где E – единичная матрица.

Сравнивая формулы, описывающие алгоритмы *RLS* и Кальмана, легко заметить их сходство [7]. Вычислительная сложность и качественные параметры двух алгоритмов также оказываются весьма близкими. Разница заключается лишь в исходных посылах, использовавшихся при выводе формул, и в трактовке параметров алгоритмов. В некоторых источниках алгоритмы *RLS* и Кальмана применительно к адаптивной фильтрации отождествляются.

5.8. Описание алгоритма и алгоритмов выбора начальных значений

В основу положен алгоритм RLS (описание в п. 5.6, формулы (5.14) – (5.17), рис. 5.1). Обработка каждой из трех координат производится независимо (скалярный адаптивный алгоритм). Блок-схема алгоритма приведена на рисунке 5.2.

Вначале производится задание численных параметров (описание см. ниже). Основной входной параметр – размерность вектора задержки (обозначен буквой N выше и L в тексте программы). Далее в качестве первых L значений скорректированного сигнала просто задаются значения сигнала ИНС.

Производится вычисление вспомогательных массивов алгоритма. Последние представляют собой векторы размерности L ($u_{t_inv_short}$, $u_{ins_inv_short}$, $u_{gps_inv_short}$, u_{posf}) или L (u_t , u_{ins} , u_{gps}) с компонентами, равными значениям времени или сигналов ИНС, GPS. Иными словами, это векторы задержки, но не только для входного сигнала (роль которого в данном случае играет ИНС), как в классических адаптивных алгоритмах, но также для эталонного сигнала (GPS) и скорректированного. Основное назначение этих массивов – использование в алгоритме прогноза сигнала GPS при паузах последнего. Они также используются во вспомогательном фильтре сглаживания сигнала.

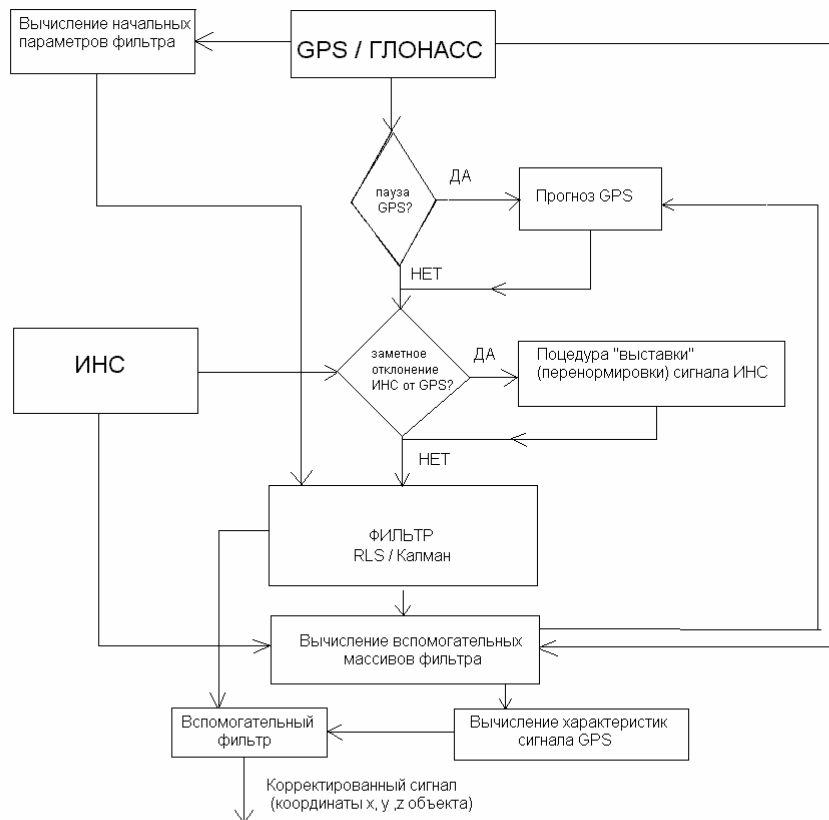


Рис. 5.2. Блок-схема алгоритма

Начальное значение вектора коэффициентов фильтра w принимается нулевым, а в качестве исходной оценки обратной корреляционной матрицы сигнала P используется диагональная матрица вида C/σ^2 , где $C = 100$. Для этой цели производится вычисление среднеквадратичного отклонения для вспомогательных массивов u_{gps} , тем самым вычисляются σ для каждой из трех координат.

Далее производится включение алгоритма. Отличие функционирования алгоритма в режиме наличия сигнала GPS и в режиме его отсутствия заключается в том, что при паузе GPS производится прогноз последнего, который затем используется при обработке сигналов в фильтре, как в обычном режиме. Кроме того, при наличии GPS производится (при необходимости) процедура «выставки» (точнее, перенормировки) сигнала ИНС

при его значительном отклонении от GPS, чего никогда не делается во время паузы.

Прогноз сигнала GPS во время паузы производится путем обработки предшествующих значений GPS и ИНС. Для этой цели используются вспомогательные массивы u_{gps} , u_{ins} . Точнее, производится процедура экспоненциальной регрессии соответствующих предшествующих сигналов, что дает прогностическое значение GPS в данный момент времени.

Перенормировка сигнала ИНС при его значительном отклонении от GPS представляет собой весьма простую процедуру вычитания из всех последующих значений ИНС разности ИНС–GPS в тот момент времени, когда последняя превысила некий допустимый порог ($prec_vist$).¹

Далее производятся вычисления массивов фильтра по формулам (5.14 – 5.17), в результате получаем основной результат – скорректированные значения координат объекта pos_f . Вычисляются обновленные значения вспомогательных массивов.

Далее производятся вычисления основных статистических параметров массива последних L значений сигнала GPS (математического ожидания m и среднеквадратичного отклонения d), после чего производится вычисление прогностического значения сигнала ($posf_res$). Это прогностическое значение впоследствии при необходимости может использоваться как некая «суррогатная» альтернатива pos_f или использоваться для сравнения.²

Сигнал $posf_res$ используется в следующей процедуре – вспомогательном фильтре сглаживания «полюсов». Полюса представляют собой ар-

¹ В данной имитационной версии алгоритма производится одномоментное вычитание поправки из всех последующих значений ИНС. Однако это без труда переносится и на режим реального времени, когда поправка запоминается и вычитается из значений ИНС, когда они появляются.

² По сути эта процедура представляет собой некий очень грубый фильтр, который нельзя называть адаптивным, поскольку он, в отличие от фильтра RLS/Кальман, не подстраивается под реальные статистические характеристики входного сигнала.

тефакты вычислительной процедуры фильтра (точечные расходимости), которые возникают редко, но практически неустранимы и непредсказуемы. Вспомогательный фильтр производит отсечку значений `posf`, когда последние слишком и резко отклоняются от `m`, и заменяет их на «суррогатные» значения `posf_res`.

Наконец, производится обновление линии задержки на основе нового, вновь поступившего, значения ИНС. На этом алгоритм замыкается.

5.9. Описание входных и выходных параметров

В данной имитационной программе производится считывание сразу всех значений сигналов из файла `inposb.dat`. Пример файла (первые несколько строк) приведен в конце раздела 5.10. В результате формируются входные массивы сигналов ИНС и GPS: `pos_ins(rows, 1:3)` и `gps_pos(rows, 1:3)`. Длина массивов `rows` (т.е. длительность наблюдения за объектом в секундах) определяется программой.

Входные параметры фильтра, задаваемые пользователем:

- глубина линии задержки `L` (натуральное число);
- пороговое значение для процедуры перенормировки ИНС `prec_vist` (положительное число); оно представляет собой предельную допустимую (относительную) погрешность между ИНС и GPS, когда включается процедура «выставки» ИНС;
- пороговое значение для процедуры сглаживания полюсов `prec` (число из интервала $(0,1)$); оно представляет собой предельную допустимую (относительную) погрешность между скорректированным сигналом и средним GPS, когда включается процедура отсечки и сглаживания полюсов.

Выходные массивы данных:

- `posf(rows, 1:3)` массив трех координат скорректированного сигнала
- для сравнения можно использовать массив `posf_res(rows, 1:3)`.

5.10. Комментированный текст программы в MATLAB

```

% matlab kalman filter
% source code has been simplificated
format short e
clear posf a f_x f_y f_z ex ey ez stx sty stz pos_true pos_ins gps_pos

gps_off=140;
gps_on=180;

file=fopen('insposbb.dat','r');
tmp=fscanf(file,'%f');
fclose(file);
[rows,cols]=size(tmp);
rows=rows/10;

for i=1:rows,
    pos_true(i,1:4)= tmp(10*(i-1)+1:10*(i-1)+4)';
    pos_ins(i,1:3) = tmp(10*(i-1)+5:10*(i-1)+7)';
    gps_pos(i,1:3) = tmp(10*(i-1)+8:10*(i-1)+10)';
end;
for i=2:rows,
    if (pos_true(i,4)==int32(pos_true(i,4)));
    else
        gps_pos(i,1:3) = gps_pos(i-1,1:3)      +pos_ins(i,1:3)-pos_ins(i-1,1:3);%надо
ли это?
    end;
end;
%-----
plot3(pos_true(:,1),pos_true(:,2),pos_true(:,3),...
    gps_pos(:,1),gps_pos(:,2),gps_pos(:,3),...
    pos_ins(:,1),pos_ins(:,2),pos_ins(:,3));

legend('True','GPS','INS');
title('Input trajectoty')
ylabel('East(m)')
xlabel('North(m)')
zlabel('Up(m)')
grid on
pause
close
%-----
er1(1:rows,1:3) =pos_true(1:rows,1:3)-pos_ins(1:rows,1:3);
er2(1:rows,1:3) =pos_true(1:rows,1:3)-gps_pos(1:rows,1:3);
for i=1:rows,
    er1n(i)= sqrt(er1(i,1)*er1(i,1)+er1(i,2)*er1(i,2)+er1(i,3)*er1(i,3));
    er2n(i)= sqrt(er2(i,1)*er2(i,1)+er2(i,2)*er2(i,2)+er2(i,3)*er2(i,3));
end
plot(pos_true(1:80,4),er2n(1:80));

```

```

title('GPS Errors');
xlabel('Time (sec)');
ylabel('Error (m)');
pause
close
plot(pos_true(1:80,4),er1n(1:80));
title('INS Errors');
xlabel('Time (sec)');
ylabel('Error (m)');
pause
close
%-----

[rows,cols]=size(pos_ins);
muu=1;

%ZADANIYE PARAMETROV ALGORITMA
L=25; prec_vist=0.1; prec=0.1;

%ZADANIYE RAZMERNOSTEY MASSIVOV FILTRA
w=zeros(L+1,3); u=zeros(L+1,3); e=zeros(rows,3); P1=zeros(L+1,L+1);
II_1=zeros(L+1,L+1); K1=zeros(L+1,1);
P2=zeros(L+1,L+1); II_2=zeros(L+1,L+1); K2=zeros(L+1,1);P3=zeros(L+1,L+1);
II_3=zeros(L+1,L+1); K3=zeros(L+1,1);
ww=zeros(L+1,1); posf=zeros(rows,3); posf_res=zeros(rows,3);

%ZADANIYE PERVIH L ZNACHENIY FILTROVANNOGO SIGNALA
posf(1:L,1:3)=pos_ins(1:L,1:3); posf_res(1:L,1:3)=pos_ins(1:L,1:3);

%ZADANIYE NACHALNIH ZHACHENIY MASSIVOV FILTRA
u_ins=zeros(L+1,3);u_gps=zeros(L+1,3);ut=zeros(L+1,1);u_ins_inv=zeros(L+1,3);u_gps
_inv=zeros(L+1,3);ut_inv=zeros(L+1,1);
u_ins_inv_short=zeros(L,3);u_gps_inv_short=zeros(L,3);ut_inv_short=zeros(L,1);

for j = 1:L+1 u_ins(j,1:3)=pos_ins(L+1-j+1,1:3); u_gps(j,1:3)=gps_pos(L+1-j+1,1:3);
ut(j)=pos_true(L+1-j+1,4); end
for j = 1:L+1 u_ins_inv(j,1:3)=u_ins(L+1-j+1,1:3); u_gps_inv(j,1:3)=u_gps(L+1-
j+1,1:3); ut_inv(j)=ut(L+1-j+1); end
for j = 1:L u_posf(j,1:3)=posf(L-j+1,1:3); end
for j = 1:L u_posf_inv(j,1:3)=u_posf(L-j+1,1:3); end
for j = 1:L+1 w(j,1:3) = 1/(L+1); u(j,1:3)=pos_ins(L+1-j+1,1:3); ww(j,1:3)=0;end

%VICHISLENIYE NACHALNIH PARAMETROV FILTRA, OPTIMALNIH DLYA
DANNOGO SIGNALA (PO
%REZULTATAM OBRABOTKI PERVIH L ZNACHENIY)
C1=100/std(u_gps(:,1))^2;C2=100/std(u_gps(:,2))^2;C3=100/std(u_gps(:,3))^2;

%ZADANIYE NACHALNIH ZHACHENIY OBRATNOY KORRELYATSIONNOY
MATRITSI "P" FILTRA
for i = 1:L+1 P1(i,i)=C1; P2(i,i)=C2; P3(i,i)=C3; end

```

```

h1 = 1 + u(:,1)*P1*u(:,1); h2 = 1 + u(:,2)*P2*u(:,2); h3 = 1 + u(:,3)*P3*u(:,3);
for i = 1:L+1 II_1(i,i)=1/h1; II_2(i,i)=1/h2; II_3(i,i)=1/h3; end

%VKLUCHENIYE ALGORITMA
warning off MATLAB:polyfit:RepeatedPointsOrRescale;
for k=L+1:rows-1
    if( (pos_true(k,4)>gps_off) & (pos_true(k,4)<gps_on))
%VICHISLENIYA PRI OTSUTSTVII GPS
        for j = 1:3
            if( abs((pos_ins(k-1,j)-gps_pos(k-1,j))/pos_ins(k-1,j)) < prec )
                uu=u_gps_inv_short(:,j);
                fi = polyfit( ut_inv_short,log(uu),2);
                uf = exp(1).^polyval(fi,ut_inv_short);
                gps_pos(k,j)=interp1(ut_inv_short,uf,ut_inv(L+1),'linear','extrap');
%
gps_pos(k,j)=interp1(ut_inv_short,u_gps_inv_short(:,j),ut_inv(L+1),'linear','extrap');
            else
                gps_pos(k,j)= ( posf_res(k-1,j) + gps_pos(k-1,j) )/2 ;
            end
        end
    end
%KONETS VICHISLENIY PRI OTSUTSTVII GPS
    else

%PROTSEDURA "VISTAVKI" (PERENORMIROVKI) SIGNALA INS PRI EGO
ZNACHITELNOM
%OTKLONENII OT GPS
        for j = 1:3
            if( abs((pos_ins(k,j)-gps_pos(k,j))/abs(pos_ins(k,j))) > prec_vist )
                delta=pos_ins(k,j)-gps_pos(k,j);
                for jj=k:rows
                    pos_ins(jj,j)=pos_ins(jj,j)-delta;
                end
            else
                end
            end
        end
    end

%FILTR KALMANA
    posf(k,1)=u(:,1)*ww(:,1); posf(k,2)=u(:,2)*ww(:,2); posf(k,3)=u(:,3)*ww(:,3);
    e(k,1:3)=gps_pos(k,1:3) - posf(k,1:3);
    h1 = 1 + u(:,1)*P1*u(:,1); h2 = 1 + u(:,2)*P2*u(:,2); h3 = 1 + u(:,3)*P3*u(:,3);
    for i = 1:L+1 II_1(i,i)=1/h1; II_2(i,i)=1/h2; II_3(i,i)=1/h3; end
    K1 = II_1*P1*u(:,1); K2 = II_2*P2*u(:,2); K3 = II_3*P3*u(:,3);
    for i=1:L+1
        ww(i,1) = ww(i,1) + K1(i)*e(k,1); ww(i,2) = ww(i,2) + K2(i)*e(k,2); ww(i,3) =
ww(i,3) + K3(i)*e(k,3);
    end
    P1 = P1 - K1*u(:,1)*P1; P2 = P2 - K2*u(:,2)*P2; P3 = P3 - K3*u(:,3)*P3;

%OBNOVLENIYE VSPOMOGATELNIH MASSIVOV

```

```

for j=1:L+1 u_ins(j,1:3)=pos_ins(k+1-j+1,1:3); u_gps(j,1:3)=gps_pos(k+1-j+1,1:3);
ut(j)=pos_true(k+1-j+1,4); end
for j=1:L+1 u_ins_inv(j,1:3)=u_ins(L+1-j+1,1:3); u_gps_inv(j,1:3)=u_gps(L+1-j+1,1:3);
ut_inv(j)=ut(L+1-j+1); end
for j=1:L u_ins_inv_short(j,1:3)=u_ins_inv(j,1:3);
u_gps_inv_short(j,1:3)=u_gps_inv(j,1:3); ut_inv_short(j)=ut_inv(j); end
for j=1:L u_posf(j,1:3)=posf(k-j+1,1:3); end
for j=1:L u_posf_inv(j,1:3)=u_posf(L-j+1,1:3); end

%VICHILENIYE KHARASTERISTIK SIGNALA GPS (POSLEDNIH L ZNACHENIY)
for j = 1:3
    m = mean(u_gps(:,j)); d = std (u_gps(:,j)); delta=sign(gps_pos(k,j)-m);
posf_res(k,j)=gps_pos(k,j)-delta*d*muu;
    % posf(k,j)=(posf_res(k,j)+posf(k,j))/2; posf(k,j)=(gps_pos(k,j)+posf(k,j))/2;
end

%OBNOVLENIYE LINII ZADERJKI
for i=1:L+1
    u(i,1:3) = pos_ins(k+1-i+1,1:3);
end

%VSPOMOGATELNIY FILTR DLYA SGLAJIVANIYA VOZMOJNIH "POLUSOV"
for j = 1:3
    m = mean(u_gps(:,j));
    if( abs( (posf(k,j)-m)/m ) > prec )
        posf(k,j) = posf_res(k,j);
    end
end

%KONETS ALGORITMA
end
posf(rows,1:3)=posf(rows-1,1:3);

%for i=2:rows,
% if (pos_true(i,4)>gps_off) & (pos_true(i,4)<gps_on)
% gps_pos(i,1:3)=[0 0 0];
% end;
%end;

subplot(311);
plot(pos_true(:,4),pos_true(:,1),pos_true(:,4),pos_ins(:,1),pos_true(:,4),posf(:,1),pos_true(:,4),gps_pos(:,1));
title('Position: X Y Z');
legend('True','INS','Filtered','GPS');
subplot(312);
plot(pos_true(:,4),pos_true(:,2),pos_true(:,4),pos_ins(:,2),pos_true(:,4),posf(:,2),pos_true(:,4),gps_pos(:,2));
subplot(313);
plot(pos_true(:,4),pos_true(:,3),pos_true(:,4),pos_ins(:,3),pos_true(:,4),posf(:,3),pos_true(:,4),gps_pos(:,3));

```

```

pause
close

%-----
er1(1:rows,1:3) =pos_true(1:rows,1:3)-pos_ins(1:rows,1:3);
er2(1:rows,1:3) =pos_true(1:rows,1:3)-gps_pos(1:rows,1:3);
erf(1:rows,1:3) =pos_true(1:rows,1:3)-posf(1:rows,1:3);
for i=2:rows,
    if (pos_true(i,4)>gps_off) & (pos_true(i,4)<gps_on)
        er2(i,1:3)=[0 0 0];
    end;
end;

for i=1:rows,
    er1n(i)= sqrt(er1(i,1)*er1(i,1)+er1(i,2)*er1(i,2)+er1(i,3)*er1(i,3));
    er2n(i)= sqrt(er2(i,1)*er2(i,1)+er2(i,2)*er2(i,2)+er2(i,3)*er2(i,3));
    erf_n(i)= sqrt(erf(i,1)*erf(i,1)+erf(i,2)*erf(i,2)+erf(i,3)*erf(i,3));
end;

subplot(311)
plot(pos_true(:,4),er1(:,1),pos_true(:,4),er2(:,1),pos_true(:,4),erf(:,1));
title('Errors');
legend('INS','GPS','Filtering');
subplot(312)
plot(pos_true(:,4),er1(:,2),pos_true(:,4),er2(:,2),pos_true(:,4),erf(:,2));
subplot(313)
plot(pos_true(:,4),er1(:,3),pos_true(:,4),er2(:,3),pos_true(:,4),erf(:,3));
xlabel('Time (sec)');
pause
close

plot(pos_true(:,4),er1n(:),pos_true(:,4),er2n(:),pos_true(:,4),erfn(:));
title('Trajectory Errors');
legend('INS','GPS','Filtering');
pause

close
%-----

```

Пример файла inposb.dat. (первые несколько строк), из которого производится считывание (по столбцам) всех значений сигналов

```

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 -5.256897 -11.935277 -
7.555930
5.047140 6.014947 6.539573 0.100000 5.051447 6.014832 6.553451 -5.256897 -11.935277 -
7.555930
10.340481 12.323306 13.302540 0.200000 10.351656 12.313794 13.333108 -5.256897 -
11.935277 -7.555930
15.880765 18.925959 20.287519 0.300000 15.897750 18.890825 20.329712 -5.256897 -
11.935277 -7.555930

```


21.669404 25.824590 27.491851 0.400000 21.691040 25.760259 27.539151 -5.256897 -
11.935277 -7.555930
27.707724 33.020780 34.913010 0.500000 27.732178 32.924199 34.936767 -5.256897 -
11.935277 -7.555930
33.996998 40.516045 42.548537 0.600000 34.019838 40.379892 42.518181 -5.256897 -
11.935277 -7.555930
40.538453 48.311847 50.396032 0.700000 40.561536 48.125663 50.295706 -5.256897 -
11.935277 -7.555930
47.333270 56.409594 58.453155 0.800000 47.364391 56.172686 58.292680 -5.256897 -
11.935277 -7.555930
54.382587 64.810644 66.717622 0.900000 54.425978 64.531032 66.520095 -5.256897 -
11.935277 -7.555930
61.687505 73.516305 75.187202 1.000000 61.749413 73.188564 74.953589 43.853184
89.305123 73.126352
69.249085 82.527846 83.859715 1.100000 69.331244 82.153972 83.585484 43.853184
89.305123 73.126352
77.068356 91.846490 92.733029 1.200000 77.173205 91.440795 92.423211 43.853184
89.305123 73.126352
85.146311 101.473422 101.805057 1.300000 85.281459 101.045205 101.463869 43.853184
89.305123 73.126352

5.11. Описание параметров работоспособности фильтра и примеры применения

Типичный пример действия фильтра в режиме наличия сигнала GPS приведен на рисунке 5.3, а в режиме паузы GPS – на рисунке 5.4 (значения входных параметров $L=25$, $\text{prec_vist}=\text{prec}=0.1$). Тестирование при различных значениях L показали, что меньшие значения этого параметра иногда позволяют добиваться несколько лучших результатов (рис. 5.5), однако в целом на основании испытаний рекомендуется значение $L=25$. Дальнейшее увеличение этого параметра в ряде случаев приводит к искаженным результатам и не рекомендуется.

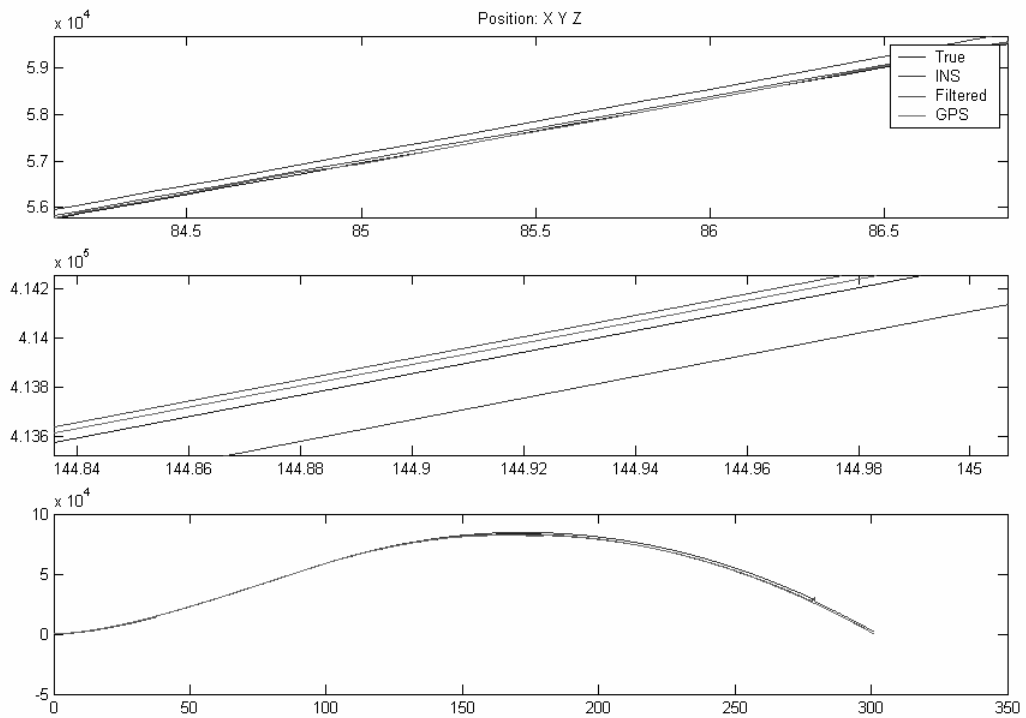


Рис. 5.3. Пример действия фильтра в режиме наличия сигнала GPS

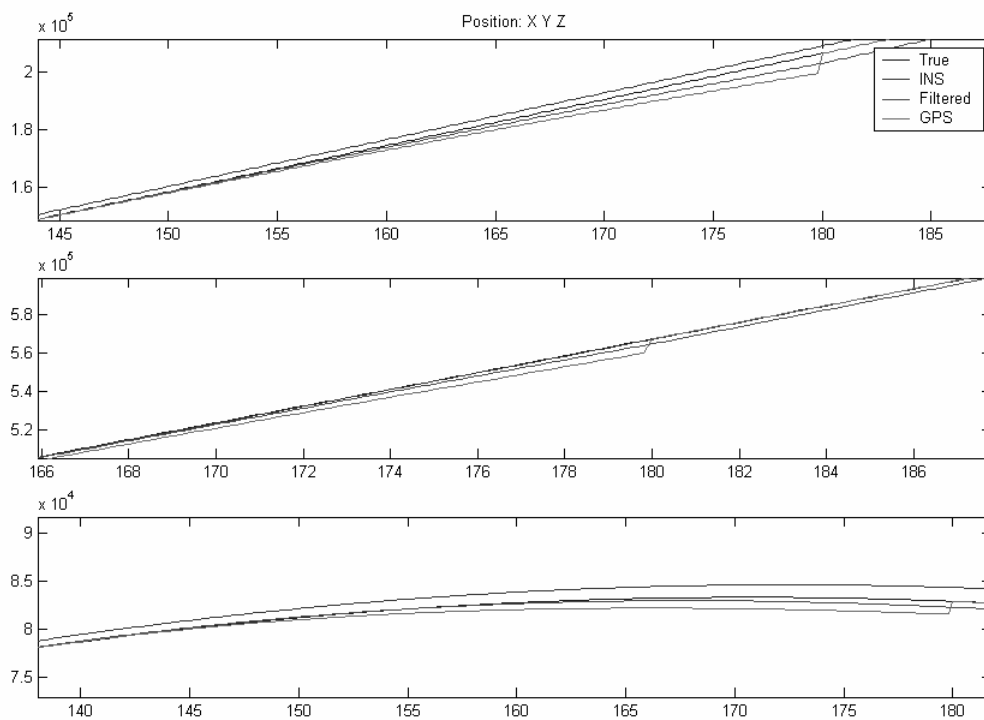


Рис. 5.4. Пример действия фильтра в режиме паузы

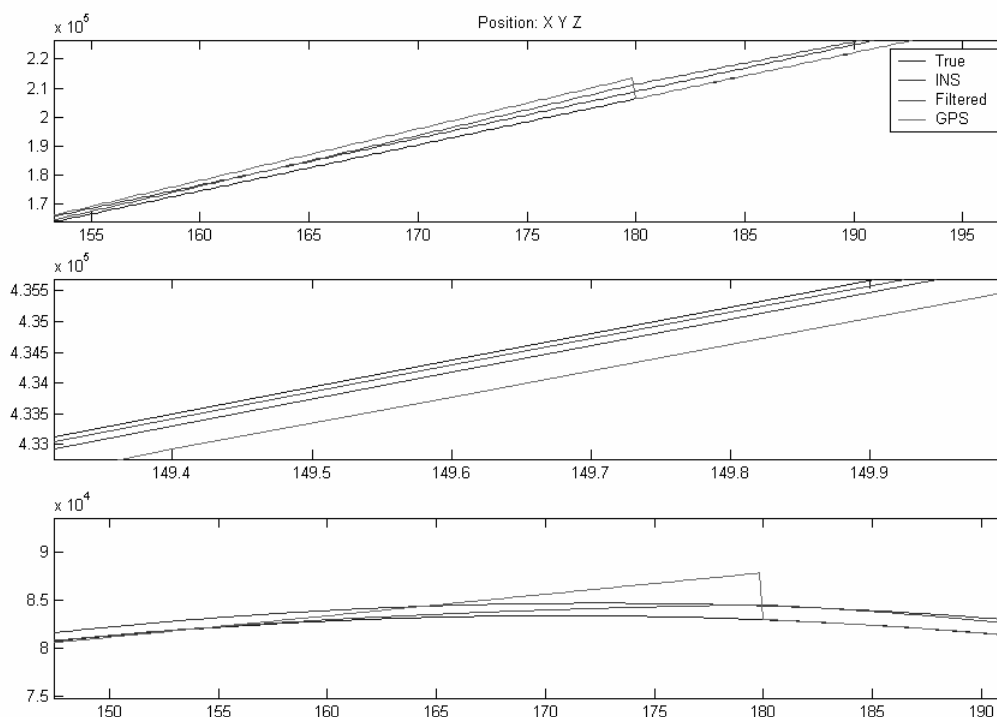


Рис. 5.5. Тестирование при различных значениях L

Значение параметра `prec_vist` определяется свойствами (качеством) используемой ИНС аппаратуры. Естественно, что критическое значение этот параметр приобретает только в случае сильного ухода (дрейфа) сигнала ИНС. Примеры на рисунках 5.6, 5.7, 5.8 показывают, что в таких условиях (третья координата на указанных рисунках) слишком частая перенормировка ИНС (`prec_vist=0.1` на рисунке 5.6 и `prec_vist=0.15` на рисунке 5.7) может приводить к очень плохим результатам во время паузы GPS. Более редкая перенормировка (`prec_vist=0.3` на рисунке 5.8) позволяет поправить ситуацию и «держать» паузу. Приведен также пример редкой (`prec_vist=0.9` на рисунке 5.9)

Параметр `prec` не может быть слишком малым, иначе это приводит к дискредитации работы фильтра, особенно во время пауз (пример на рисунке 5.10). Пример возникновения полюсов (очень большое значение

prec_vist) приведен на рисунке 5.11. Рекомендуемые значения – $0.05 \div 0.1$.

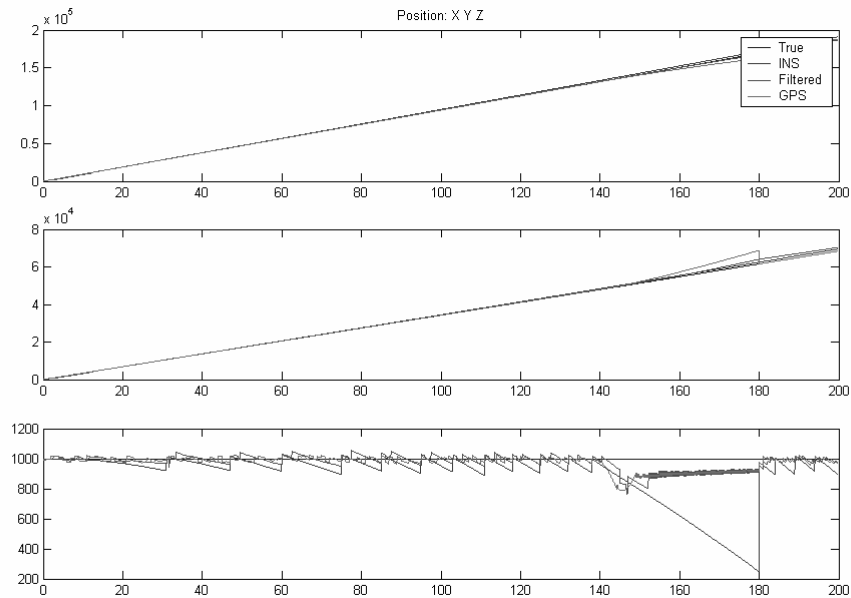


Рис. 5.6. Тестирование при $prec_vist=0.1$

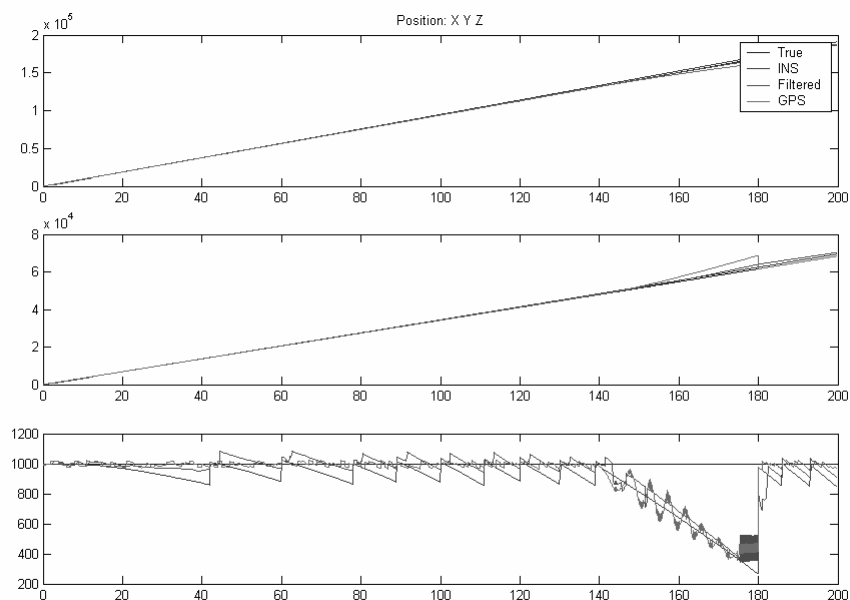


Рис. 5.7. Тестирование при $prec_vist=0.15$

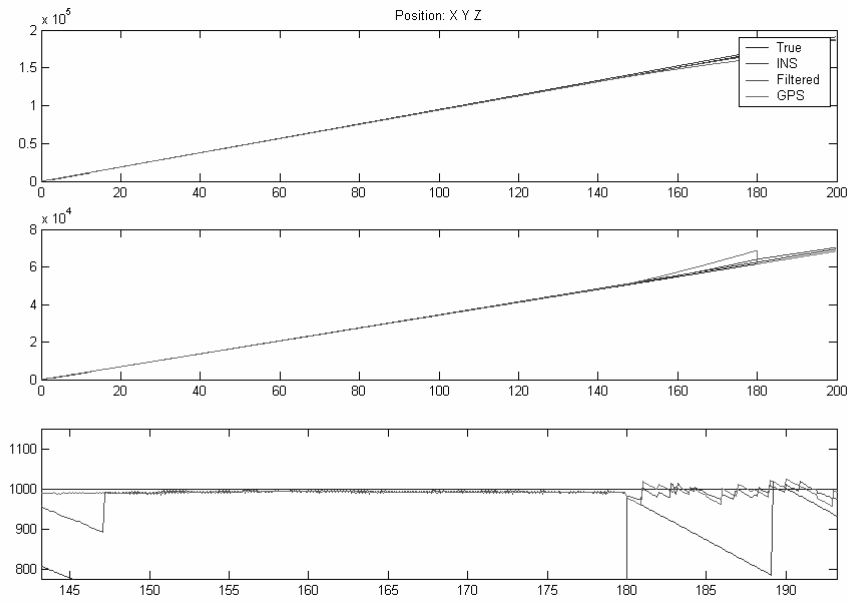


Рис. 5.8. Тестирование при $prec_vist=0.3$

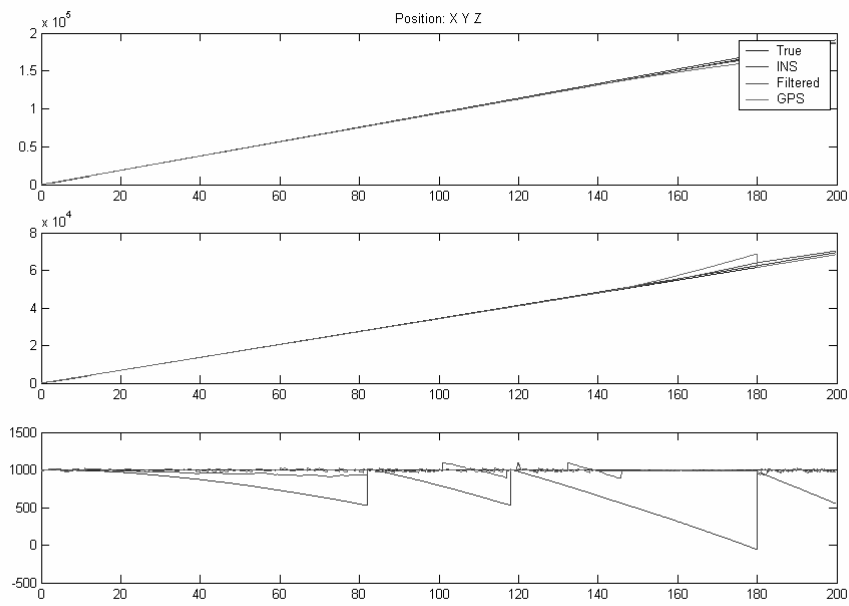


Рис. 5.9. Тестирование при $prec_vist=0.9$

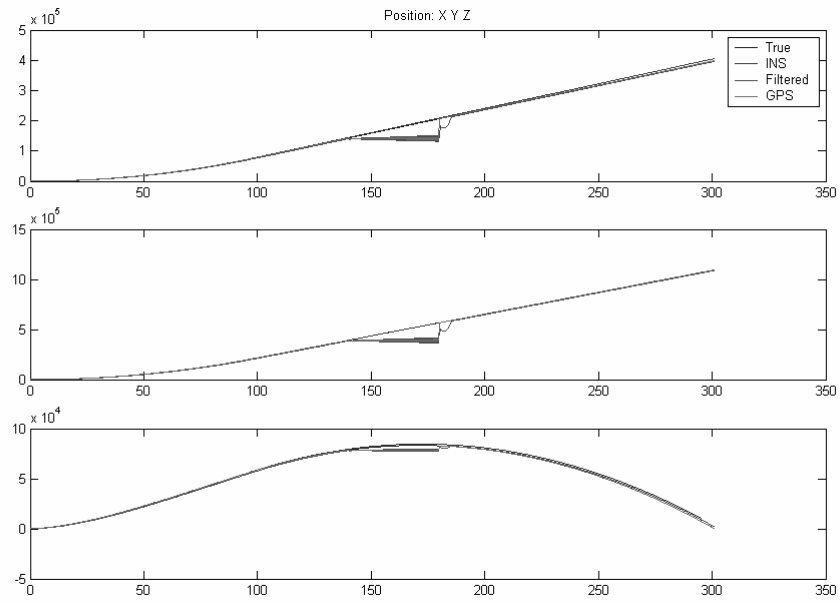


Рис. 5.10. Пример сбоя работы фильтра во время пауз

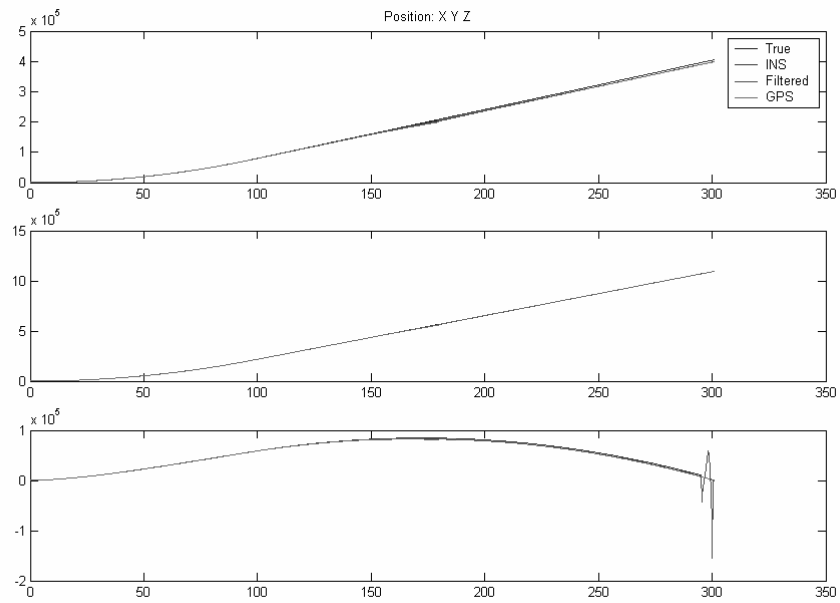


Рис. 5.11. Пример возникновения полюсов (очень большое значение `prec_vist`)

Список литературы к теме 5

Использованная литература:

1. Уидроу Б., Стирнэ С.Д. Адаптивная обработка сигналов. – М.: Радио и связь, 1989. – 440 с.
2. Адаптивные фильтры: Пер. с англ./ Под ред. К.Ф.Н. Коуэна и П.М. Гранта. – М.: Мир, 1988. – 392 с.
3. Naykin S., Adaptive Filter Theory. – Prentice Hall, 2002. – 936 p.
4. Сергиенко А.Б. Алгоритмы адаптивной фильтрации: особенности реализации в MATLAB. В журнале «Exponenta Pro», 2003, №1, с. 18–28.
5. Прокис Дж. Цифровая связь. М.: Радио и связь, 2000. – 800 с.
6. Васильев Ф.П. Численные методы решения экстремальных задач. – М.: Наука, 1988. – 552 с.
7. Первозванский А.А. Курс теории автоматического управления. – М.: Наука, 1986. – 616 с.
8. Шкирятов В.В. Радионавигационные системы и устройства. – М.: Радио и связь, 1994. – 160 с.
9. Шамси Баша. Комплексирование ИНС/GPS–ГЛОНАСС с целью коррекции углов ориентации подвижного объекта. Дисс. на соиск. уч. степ. к.т.н. – М.: Моск. гос. техн. ун-т им Н.Э.Баумана, 2000. – 114 с.
10. Кузовков Н.Т., Салычев О.С. Инерциальная навигация и оптимальная фильтрация. – М.: Машиностроение, 1982. – 215 с.
11. Кузовков Н.Т., Карабанов С.В., Салычев О.С. Непрерывные и дискретные системы управления и методы идентификации. - М.: Машиностроение, 1978. – 221 с.
12. Salychev O.S. Inertial Systems in Navigation and Geophysics. – Moscow: Bauman MSTU Press, 1998. – 352 p.

Рекомендованная литература:

1. Адаптивные фильтры: Пер. с англ./ Под ред. К.Ф.Н. Коуэна и П.М. Гранта. – М.: Мир, 1988. – 392 с.
2. Сергиенко А.Б. Алгоритмы адаптивной фильтрации: особенности реализации в MATLAB. В журнале «Exponenta Pro», 2003, №1, с. 18–28.

Интернет-ресурсы к теме 5

Сергиенко А.Б. Алгоритмы адаптивной фильтрации: особенности реализации в MATLAB. В журнале «Exponenta Pro», 2003, №1, с. 18–28.

<http://exponenta.ru/educat/>

Контрольные вопросы к теме 5

1. Каковы недостатки инерциальных навигационных систем?
2. Что такое обновляемая последовательность?
3. Что представляют собой сигналы на рисунках 5.3?
4. Каковы особенности фильтров Винера, LMS, RLS и Кальмана?
5. Что такое полюса в фильтрах и отчего они возникают?
6. Что вычисляется в следующем цикле программы?

```
for i=1:rows,  
    er1n(i)= sqrt(er1(i,1)*er1(i,1)+er1(i,2)*er1(i,2)+er1(i,3)*er1(i,3));  
    er2n(i)= sqrt(er2(i,1)*er2(i,1)+er2(i,2)*er2(i,2)+er2(i,3)*er2(i,3));  
end
```

7. Объясните смысл следующей цепочки операторов программы:

$$P1 = P1 - K1 * u(:,1)' * P1; \quad P2 = P2 - K2 * u(:,2)' * P2; \quad P3 = P3 - K3 * u(:,3)' * P3;$$

Заключение

В случае если читатель досконально разобрал и проработал все приведенные в настоящем пособии примеры, т.е. понял, с одной стороны, их физическое содержание, а с другой – их программную реализацию, – следовательно, он достиг некоторого минимального уровня, позволяющего профессионально применять численно-аналитические пакеты для вычислений и моделирования. Дальнейший этап обучения состоит в выполнении лабораторных работ. Необходимо, однако, помнить, что достигнутый уровень является минимальным и читателю нужно теперь совершенствоваться самостоятельно посредством решения практических задач. Напомним в этой связи еще один известный принцип, согласно которому пределов для совершенствования не существует.

ОПИСАНИЕ КУРСА И ПРОГРАММА

Кредит: 4

Описание курса

Цель курса – обеспечение базовой подготовки студентов для работы с системами аналитических вычислений (системами компьютерной алгебры) в различных математических, физических и инженерных приложениях; изучение основ работы с различными системами (Maple, MATLAB, Mathcad); проведение анализа преимуществ и недостатков каждого из рассматриваемых пакетов; ознакомление студентов с разработками выполненными в перечисленных выше системах; обучение студентов современным методам моделирования физических процессов; изучение принципов организации и проведения математического моделирования; разработка алгоритмов, создание студентами моделей и решение сложных физических и инженерных задач.

Содержание курса – Системы компьютерной алгебры. Применение систем компьютерной алгебры в науке, технологиях и образовании. Разработка алгоритмов для символьных вычислений и их обработка на компьютере. Системы символьной (аналитической) математики — Maple 11, Mathcad 14, MATLAB 7. Toolboxes. Интерфейсы. Ядро системы. Библиотеки. Системы символьной математики и языки программирования высокого уровня. Численные методы решения прикладных задач и их реализация в современных пакетах Maple 11, Mathcad 14, MATLAB 7. Методы решения обыкновенных дифференциальных уравнений, интегральных уравнений. Разностные схемы решения уравнений математической физики. Визуализация результатов вычислений. Примеры

применения систем компьютерной алгебры в научно-технических расчетах. Примеры применения систем компьютерной алгебры в образовании.

Организационно-методическое построение курса. Курс состоит из лекций и практических занятий, предусмотрено выполнение курсовых работ и контрольных заданий. В программе курса предусмотрено выполнение большого объема практических работ, связанных с решением задач с использованием систем компьютерной алгебры. Изложение курса базируется на большинстве разделов курсов информатики, высшей математики (высшая алгебра, математический анализ, обыкновенные дифференциальные уравнения, уравнения в частных производных, методы математической физики), курсов общей физики и теоретической физики (электродинамика, статистическая физика, квантовая физика), входящих в учебный план обучения бакалавра классического университета по направлению подготовки – физика.

Лекции построены на принципе «от простого к сложному» и реализуют непрерывную подготовку в рамках учебной программы. Каждая лекция сопровождается демонстрацией иллюстративного видеоматериала (презентаций, слайдов и т.п.) с использованием демонстрационных программ.

Первым этапом практических занятий является обсуждение общей концепции, выполняемой студентами работы и проверкой преподавателем самостоятельной работы каждого студента, результатом которой является допуск (или недопуск) к выполнению работы. На втором этапе студент выполняет практическую работу, обрабатывает полученные результаты и составляет отчет согласно методическим требованиям. На третьем этапе обсуждаются отчеты студентов по выполненной теме с каждым студентом индивидуально, так и в составе всей группы.

Для контроля и закрепления студентами полученных знаний проводятся практические занятия (9 практических работ), 1 курсовая работа или 1 контрольное практическое задание (по выбору). Предусмотрены: промежуточная аттестация в середине семестра, зачет по практическим занятиям и по окончании курса – экзамен.

Освоив курс, студент должен: иметь представление о наиболее часто используемых в физике и инженерных приложениях современных системах компьютерной алгебры; овладеть методами математического и численного моделирования; научиться разрабатывать алгоритмы и численные модели; уметь выбирать систему компьютерной алгебры, наиболее подходящую для решения поставленной задачи; получить навыки решения физических и инженерных задач с использованием систем компьютерной алгебры.

Общие правила выполнения практических и самостоятельных работ

В рамках читаемого курса студенты выполняют практические работы, количество которых определено учебным планом по дисциплине, а также выполняют самостоятельную исследовательскую работу – курсовую работу (или контрольное практическое задание).

Практическое учебное занятие проводится в профилированных учебных лабораториях с целью углубления знаний и приобретения навыков решения физических и инженерных задач.

Объектом изучения при проведении практических занятий является конкретный физический объект, рассматриваемый в предметной области изучаемой учебной дисциплины. Объект изучения (эффект, явление, образец, установка) может быть представлен в виде модели - математической, графической, знаковой и т.д.

Курсовая работа (контрольное практическое задание) представляет собой самостоятельную учебную работу – первое научное исследование, в котором студентом раскрываются теоретические и практические проблемы избранной темы.

Курсовая работа должна представлять законченный материал, в котором представлены результаты, раскрывающие закономерности взаимосвязи между изучаемыми явлениями, процессами и содержать авторское видение и решение поставленной задачи.

Объектом исследований при выполнении курсовой работы является изучение физических явлений или процессов, при этом самостоятельно привлекаемые методы моделирования, алгоритмы либо подходы теоретического изучения, преимущественно основанные на знаниях полученных студентом в предшествующий период обучения или изучаемых самостоятельно. Результаты курсовой работы могут являться прологом будущей дипломной работы.

Контрольное практическое задание – самостоятельная учебная работа, направленная на анализ и решение поставленной задачи.

Контрольное практическое задание представляет, законченный материал, в котором представлены результаты самостоятельной работы студента по разработке и постановке новых упражнений практикума и связаны с совершенствованием постановки математического моделирования, разработки новых алгоритмов и программ и содержат авторское видение и решение поставленной задачи.

Объектом исследований при выполнении контрольного практического задания являются существующие и вновь создаваемые модели физических процессов и явлений.

Организация практических занятий

Общие правила:

- Практические работы выполняются студентами согласно установленного в начале семестра расписания.
- Практическая работа выполняется индивидуально.
- Количество практических работ, выполняемых за учебное занятие, не превышает одну работу.
- Перенос выполнения назначенной практической работы допускается только в пределах расписания по согласованию с преподавателем.

К выполнению работы не допускаются учащиеся, которые:

- не подготовились для выполнения практической работы;
- грубо нарушают правила внутреннего распорядка в лаборатории;
- опоздали к началу занятий;
- не защитили отчет по предыдущей работе;
- пропустили два и более занятий без уважительной причины.

Организация практических занятий включает:

- самостоятельную внеаудиторную подготовку студента в соответствии с методическими рекомендациями;
- первичный контроль преподавателем степени подготовленности каждого студента к выполнению практической работы;
- выполнение всех заданий (упражнений) практической работы в полном объеме;
- оформление отчета (см. структуру отчета и правила его оформления) и его защиту в сроки, установленные преподавателем;
- учет преподавателем текущего и итогового рейтингов каждого из студентов по результатам выполнения и защиты им отдельных лабораторных работ.

Студент имеет право:

- получить необходимые для выполнения практической работы методические материалы в бумажном или электронном видах;
- проводить практические работы по оригинальным методикам при предварительном согласовании их с преподавателем;
- выполнить практической работу, пропущенную по уважительной причине, в часы, согласованные с преподавателем.

Студент обязан:

- Самостоятельно, согласно методическим рекомендациям, подготовиться к выполнению практической работы и получить допуск к ее выполнению по результатам краткого опроса в начале занятий;
- По выполнении работы студент предъявляет результаты преподавателю. Они сохраняются в виде электронного файла на рабочем компьютере преподавателя. Копия файла остается у студента для оформления отчета (см. правила оформления отчётов);
- При пропуске занятия подготовиться к следующей по расписанию работе. Дату выполнения пропущенной работы необходимо согласовать с преподавателем.

Студент несет ответственность:

- за пропуск практических занятий по неуважительной причине;
- за неподготовленность к выполнению работы;
- за несвоевременную сдачу отчетов и их защиту;
- за порчу имущества и нанесение материального ущерба лаборатории.

Преподаватель несет ответственность:

- за качественную постановку и проведение практических работ;
- за поддержание рабочей дисциплины и порядка в лаборатории;

Преподаватель имеет право:

- отстранять от проведения практических работ студентов, нарушающих установленные выше правила;

- требовать от студентов пропустивших занятия разрешения из деканата факультета на посещение последующих практических занятий;
- проводить перед началом практических работ контрольный опрос студентов;
- вносить в случае необходимости (из-за отказа оборудования или вычислительных средств и т. п.) частичные изменения в программу практической работы.

Категорически запрещено:

- **Устанавливать или изменять** программное обеспечение на компьютерах лаборатории;
- **Вскрывать** системные блоки;
- **Копировать на внешние носители** программное обеспечение без согласование с преподавателем.

***Организация выполнения курсовых работ
или контрольных практических заданий***

О необходимости выполнения курсовой работы студента информируют на первой лекции и предлагают либо выбрать тему работы из списка, либо заранее самостоятельно обдумать и предложить тему будущей работы.

Подготовка и защита курсовой работы, а также ее оценка в учебной программе каждого направления определяется большой ролью этого вида подготовки специалиста в общей системе учебных программ, нацеленных на фундаментальность и систематичность образования. Для завершения и защиты курсовой работы студенту предоставляется в конце семестра специально 1 неделя, свободная от аудиторных занятий. В течение 1 установочной недели, отведенной на выбор курсов, утверждается тема и научный руководитель курсовой работы. Темы курсовых работ находятся

на портале кафедр. Студент, консультируясь с преподавателями кафедры, за которой закреплено руководство курсовыми работами, определяет тему своей курсовой работы.

Выполнение курсовых работ происходит по следующей схеме:

- выбор темы;
- поиск литературы и ее изучение;
- проведение предварительного анализа, аналитических оценок, математического моделирования;
- написание первого варианта работы и представление его научному руководителю;
- исправление и доработка текста на основе замечаний руководителя;
- представление окончательного текста курсовой работы и ее защита перед учебной комиссией кафедры.

Отчетность (ведомость) по курсовой работе сдается в учебную часть в строго определенные сроки (1 неделя).

Разработка программного обеспечения, необходимого для проведения вычислительного эксперимента, осуществляется в научно-учебной «Лаборатории вычислительного эксперимента и систем автоматизации физических измерений», оснащенного всем необходимым оборудованием и программным обеспечением.

В процессе работы над курсовым проектом текущее состояние работы еженедельно обсуждается с преподавателем, курирующим ее выполнение.

Обработка результатов практической части работы, и подготовка отчета по курсовому проекту может быть выполнена студентом в дисплей-классе центра прикладных информационных технологий университета, оснащенного практически всеми необходимыми аппаратными и программными ресурсами.

Защита курсовых работы осуществляется перед экспертной учебной комиссией, назначаемой зав. кафедрой на специальном заседании научного семинара кафедры и представляет собой публичное научное сообщение по результатам работы должным образом оформленное с использованием мультимедийных средств. Лучшие работы могут быть рекомендованы в качестве докладов на конференциях или публикации в научных журналах.

Инновационная составляющая курса

В последнее десятилетие системы компьютерной алгебры находят все более широкое применение в различных областях науки, становятся все более популярными в образовании. Эти системы являются мощными инструментами при моделировании широкого спектра различных процессов и явлений: от физических до социальных. Существует огромное количество литературы, Интернет-ресурсов, библиотек, посвященных различным пакетам аналитических вычислений. Системы компьютерной алгебры лишены недостатков, присущих численному моделированию и вычислительному эксперименту, связанных с ошибками округления, устойчивостью метода, сходимостью. Реализация предлагаемого УМК будет способствовать повышению уровня подготовки магистров, расширению их научного кругозора. Выпускники магистратуры будут способны самостоятельно применять современные методы математического моделирования, применять научно обоснованные и проработанные подходы для изучения сложных физических процессов и явлений с использованием систем компьютерной алгебры, осуществлять самостоятельную постановку и решение сложных инженерных и физических задач, что значительно повысит их творческий и научный

потенциал. Особое внимание в УМК уделено практическому аспекту подготовки, столь необходимому научному исследователю.

Инновационная форма подачи учебного материала в курсе по отношению к учебникам традиционной формы определяется тем, что курс одновременно содержит в себе научно теоретический и методический материал по соответствующей учебной дисциплине, включающий описание практических работ, которые по сути своей являются научным исследованием.

Освоение курса связано с большой практической составляющей, с выполнением самостоятельных индивидуальных исследований в форме курсовых работ и/или контрольного задания, согласно перечня предложенных актуальных тем, являющихся прологом выполнения магистерской диссертационной работы. Кроме того, результатом выполнения курсовых работ или контрольных заданий будут являться новые разработанные студентами практические работы. Таким образом, в данный курс заложен элемент саморазвития.

В реализуемой магистерской программе все УМК построены по единой форме расположения и организации материала, позволяют соотносить их содержание в общем контексте подготовки специалистов. Использование единого подхода к представлению учебно-методической информации дает возможность отобразить, с одной стороны, существующие межпредметные взаимосвязи, а с другой – динамику развития отдельных тем и их сочетание, что особенно актуально при выполнении самостоятельных научных исследований (курсовые и дипломные работы, магистерская диссертация). Выполнение практических работ и курсовых работ позволяют получить практические навыки по разделам изучаемой дисциплины.

Предлагаемая для изучения курса литература в основе своей имеется в наличии в библиотечном фонде РУДН, в случае отсутствия литературы в

библиотечном фонде электронные версии доступны в локальной сети на сайте магистерской программы (<http://vlab.sci.pfu.edu.ru>). Характерной особенностью данного курса является работа студентов с Интернет-ресурсами, в которых представлен поистине неисчерпаемый материал по различным системам компьютерной алгебры.

Разрабатываемый УМК является плодом совместной работы коллектива авторов, состоящих из преподавателей РУДН и ведущих специалистов крупнейшего научного центра РФ «Курчатовский институт». Такой подход позволяет рассмотреть последние достижения в области вычислительного эксперимента, обеспечить проведение НИРС с использованием современных методов численного моделирования.

Авторы курса выделили две основные цели написания:

1. Разработка разделов, являющихся на сегодняшний день наиболее актуальными для студентов физических и инженерных специальностей.
2. Практическая составляющая курса, нацеленная на изучение и освоение методов и принципов постановки математического и численного моделирования в физике и инженерных науках с использованием систем компьютерной алгебры.

Предлагаемый учебный курс является принципиально инновационным как в области содержания, так и технологии организации педагогического процесса.

Обязательная литература

1. *В.П. Дьяконов Maple 9.5/10 в математике, физике и образовании. Серия «Библиотека профессионала. Из-во «Солон», 2006. 720 с.*
2. *Е.Р. Алексеев, О.В.Чеснокова Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9. Из-во «ИТ Пресс», 496 с.*

3. В.П. Дьяконов *Mathcad 11/12/13 в математике: справочник*. Из-во «Горячая линия - Телеком», 2007. 958 с.
4. Д.В. Кирьянов *Mathcad 14*. Из-во ВХВ, 2007. 704 с.
5. А.М. Половко *Mathematica*. Из-во ВХВ, 2007. 368 с.
6. Е. Макаров. *Инженерные расчеты в Mathcad 14*. Из-во «Питер», 2007. 592 с.
7. К.Э. Плохотников, Б.И. Волков, С.С. Задорожный, В.А. Антонюк
Редактор: К.Э. Плохотников. *Методы разработки курсовых работ. Моделирование, вычисления, программирование на С/С++ и MATLAB, виртуализация, образцы лучших студенческих курсовых работ*. Из-во «Солон», 2006. 320 с.
8. В. Гандер, И. Гржебичек *Решение задач в научных вычислениях с применением Maple и MATLAB*. Из-во «Вассамедина», 2005. 520 с.

Интернет-ресурсы

1. Виктор Аладьев. Система компьютерной алгебры Maple:
Библиотека новых программных средств.
www.lib.grsu.by/library/data/links/aladiev/Download/Maple_Lib.htm
2. Научный форум. Что лучше, MatLab, MathCad или Maple?
<http://www.lib.mexmat.ru/forum/viewtopic.php?t=6803&postdays=0&postorder=asc&start=0&sid=3f5a631f553046a6a49d431ef10c64d2>.
3. Руководство пользователя MAPLE.
<http://exponenta.ru/soft/Maple/manson/poweredition/0.asp>
4. Руководство пользователя MATHCAD.
<http://exponenta.ru/soft/Mathcad/UsersGuide/0.asp>
5. Руководство пользователя MATLAB.
http://exponenta.ru/soft/matlab/potemkin/book/matlab/chapter0/0_0.asp

6. *Краткое описание функционала MATLAB.*
<http://itc.ua/article.phtml?ID=21405&IDw=29&pid=18>
7. *Что такое MATLAB на Wikipedia -*
<http://ru.wikipedia.org/wiki/MATLAB>.
8. *Области применения систем компьютерной алгебры и конкретные примеры.* <http://exponenta.ru/educat/systemat/systemat.asp>.
9. *Применение MATLAB.* <http://matlab.exponenta.ru/mltb/default.php>.
10. *EqWold. Мир математических уравнений.*
<http://eqworld.ipmnet.ru/ru/software.htm>
11. *Математическое моделирование в естественных науках.*
http://mathmod.aspu.ru/?id=6&sub_id=1
12. *Вычислительные методы и программирование.* <http://num-meth.srcc.msu.su>

Условия и критерии выставления оценок: От студентов требуется посещение лекций и семинарских занятий, обязательное участие в аттестационных испытаниях. Особо ценится активная работа при выполнении практических работ, а также качество выполнения контрольных работ и экзаменационных эссе. Именно творческий подход к выполнению практических работ и курсовых работ являются главным критерием высокой экзаменационной оценки.

Для успешного выполнения каждой практической работы студент должен внимательно изучить учебно-методические материалы (пособия и литературу, рекомендованную для выполнения практических работ), уметь изложить изученный материал и быть готовым к выполнению. Важным этапом в итоговой аттестации студента является обязательное выполнение и сдача самостоятельного контрольного практического задания (или курсовой работы). Студент не допускается к итоговому экзамену, если он набрал менее 50% баллов. При набранных 60% баллов студент **может**

автоматически получить оценку «3». Более высокая оценка может быть получена только на итоговом экзамене.

Балльная структура оценки:

За выполнение обязательных практических работ (8 работ в течение семестра) – 40 баллов. За каждую практическую работу – 5 баллов (1 балл – допуск к выполнению работы, 1 балл – выполнение, 3 балла – оформление и сдача работы).

В итоговом подсчете на практические работы – 40 баллов (зачет по лабораторному практикуму).

Промежуточный контроль знаний – 10 баллов.

Самостоятельная контрольная практическая работа – 20 баллов (всего).

Итоговый экзамен – 30 баллов.

Всего – 100 баллов.

При выборе курсовой работы по данному курсу, контрольное практическое задание снимается, максимальный балл за курсовую работу – 30 баллов засчитывается в итоговую сумму баллов.

Шкала оценок:

A (5+) - $93 \leq 100$ баллов;

B (5) - $84 \leq 92$ баллов;

C (4) - $74 \leq 83$ баллов;

D (3+) - $63 \leq 73$ баллов;

E (3) - $51 \leq 62$ баллов;

FХ (2+) - $31 \leq 50$ баллов;

F (2) - $0 \leq 30$ баллов.

		Неуд		3		4	5	
кредит	Сумма	F	FX	E	D	C	B	A
	Баллов	2	2+	3	3+	4	5	5+
2	100	0≤30	31≤50	51≤62	63≤73	74≤83	84≤92	93≤100

Пояснение оценок:

A – выдающийся ответ

B – очень хороший ответ

C – хороший ответ

D – достаточно удовлетворительный ответ

E – отвечает минимальным требованиям удовлетворительного ответа

FX – означает, что студент может добрать баллы только до минимального удовлетворительного ответа

F – неудовлетворительный ответ (либо повтор курса в установленном порядке, либо основание для отчисления).

Требования к оформлению отчета самостоятельных работ (лабораторных, курсовых работ и контрольных практических заданий)

Структура отчета

Объем отчета по практической работе не должен превышать 8 стр. Объем отчета по курсовой работе не должен превышать 18 стр. Текст набирается на компьютере и печатается на принтере. Требования по содержанию разделов и оформлению отчета изложены ниже.

Отчет (требования к содержанию разделов лабораторного отчета)

включает:

- Титульный лист;
- Аннотация;

Представляет собой краткое (несколько предложений) содержание работы, включающее цель работы, объект исследований, используемый метод, диапазон варьируемых параметров эксперимента, основной результат, погрешности измерений.

1. Введение.

Содержит краткое теоретическое рассмотрение изучаемого явления и краткий обзор литературы по изучаемому вопросу;

2. Описание вычислительного эксперимента.

В данном разделе, относящемся к постановке эксперимента, необходимо дать описание параметров эксперимента, используемых численных методов и алгоритмов, предварительную оценку точности используемых методов.

3. Проведение вычислительного эксперимента.

Описываются особенности программ (кодов), созданных для проведения численных экспериментов, диагностики вычислительного эксперимента. Способы оптимизации.

4. Результаты вычислительного эксперимента.

Приводятся результаты вычислительного эксперимента измерений в виде таблиц и/или графиков.

5. Анализ результатов измерений, включающий:

Обсуждение полученных результатов; определение основных результатов. Указывается область варьируемых параметров эксперимента, в которых был получен результат. Проводится сравнение полученных результатов с предварительными аналитическими оценками и теоретическими выводами, обсуждаются их расхождения, и делается попытка их объяснения.

6. Выводы и заключение;

Отмечается суть выполненной работы, делаются выводы. Приветствуются оригинальные подходы к решению поставленной задачи,

критические замечания по улучшению методики проведения эксперимента.

7. Список используемой литературы.

В разделе указывается используемая при выполнении работы основная и дополнительная литература.

Общие положения по оформлению

Электронная копия отчета и ее бумажный вариант должна быть в MS Word. При оформлении границы полей, шрифты, параметры абзацев, вставки (номера страниц, рисунки, графики и т.п.) задаются с помощью панели инструментов или меню (Файл - Параметры страницы; Формат – Шрифт, Абзац, Регистр; Вставка – Номера страниц, Рисунок).

Титульные листы отчетов лабораторной работы и курсовой представлены в Приложение А. Шрифт - Times New Roman Cyr, размер - 14 пт, регистры – указаны в примере, выравнивание абзацев - по центру.

Разделы и подразделы должны иметь заголовки. Заголовки разделов располагают с левого края строки без точки в конце. Текст раздела отделяется от текста двумя межстрочными интервалами. Переносы в заголовках не допускаются. Каждый раздел рекомендуется начинать с новой страницы.

Для нумерованных заголовков разделов отчета шрифт - Times New Roman, размер - 12 пт, Bold, выравнивание по левому краю.

Пример:

- Раздел 1,
 - Подраздел 1.1,
 - пункт 1.1.1,
 - подпункт 1.1.1.1.

Основной текст - формат А-4 (297x210), ориентация - книжная. Границы полей: верхнее, нижнее – 2 см; левое– 2.5 см, правое -1.5см,

шрифт - Times New Roman, размер - 12 пт, интервал - полуторный (Word), автоматический перенос слов в границах полей, выравнивание по ширине. Формулы и символы набираются в редакторе формул Equation. Ссылки в тексте заключены в квадратные скобки.

Нумерация страниц (меню Вставка - Номера страниц). Положение - внизу страницы, выравнивание - по центру, без нумерации первой страницы - титульного листа.

Пример оформления отчета показан в методических рекомендациях.

Графический материал и таблицы

Иллюстрации: схемы, чертежи, графики, диаграммы, фотоснимки, рисунки следует располагать по тексту непосредственно после первого упоминания или на следующей странице, если в указанном месте они не помещаются.

Все иллюстрации в тексте должны быть со ссылками. Допустима как сквозная нумерация рисунков, так и нумерация в пределах раздела («согласно рисунка 4» - при сквозной нумерации, либо «согласно рисунка 3.4» при нумерации в пределах раздела). Номер следует размещать под иллюстрацией посередине после слова «Рис.».

Рисунки могут быть выполнены с помощью ЭВМ или от руки. На графиках, выражающих количественные зависимости (экспериментальные, расчетные), должна быть координатная сетка. Стрелки на осях координат в этом случае не ставятся. Цифры располагают ниже оси абсцисс и левее оси ординат. Обозначения физических величин и единиц измерения приводят через запятую с внешней стороны оси по центру или с противоположного относительно начала координат края. Масштаб координатной сетки (линейный, логарифмический масштаб или иной) выбирается из соображений удобства представления результатов. На одном графике допустимо представление семейства функциональных зависимостей. При этом кривые отображаются сглаженными линиями различного типа или

цвета, допускается вводить обозначение параметра, при котором получены данные результаты. Экспериментальные точки отмечаются на графике символами.

Результаты измерений и расчетов также целесообразно представлять в виде таблиц. Все таблицы в тексте должны быть с ссылками. Допустима как сквозная нумерация таблиц, так и нумерация в пределах раздела («согласно таблицы 4» - при сквозной нумерации, либо «согласно таблицы 3.4» при нумерации в пределах раздела). Таблицы следует располагать по тексту непосредственно после первого упоминания или на следующей странице, если в указанном месте они не помещаются. Номер таблицы следует размещать в правом верхнем углу после слова “Таблица” над заголовком таблицы. Если в работе одна таблица, её не нумеруют. Слово “Таблица” и заголовок начинаются с прописной буквы, точка в конце заголовка не ставится. Заголовки граф таблицы должны начинаться с прописных букв.

Математические символы и формулы

Уравнения и формулы в тексте располагаются в отдельной строке со свободными строками выше и ниже. Если уравнение не умещается в одну строку, оно должно быть перенесено после математических знаков (+, -, x) с их обязательным повторением в новой строке. Пояснение значений, символов и числовых коэффициентов следует приводить непосредственно по тексту или под формулой в той же последовательности, как и в формуле. Значение каждого символа и числового коэффициента следует давать с новой строки, первую строку пояснения начинают со слова “где” без двоеточия. Формулы и уравнения в работе следует нумеровать по выбранному способу нумерации сквозному или в пределах раздела.

Написание обозначений единиц физических величин

При написании числовых значений величин и их обозначений используются согласно системы измерений СИ или СГС. Между

последней цифрой числа и обозначением единицы физической величины следует оставлять пробел, исключение составляют знаки, поднятые над строкой. 15 В, 20 А, 12 Вт, 4,3 Дж, 35°, 1 10⁻³ Торр, 25%. Не допускается перенос обозначения единиц на следующую строку. Единицы измерений, названные в честь выдающихся ученых, обозначают с большой буквы, например: В (Вольт), Гц (Герц), Па (Паскаль).

Обозначения единиц измерений величин, представимых в виде произведения или частного, следует отделять точкой или косой чертой например: А·м Вт/(м² К). Десятичные кратные и дольные единицы представляют в виде: кГц (килогерц), МВт (мегаватт), мВт (милливатт), мкс (микросекунда), мс (миллисекунда).

Список использованных литературных источников

Список использованных источников составляют в порядке появления ссылок в тексте или в алфавитном порядке. Ссылки следует приводить в форме указания порядкового номера по списку источников, выделенного квадратными скобками, например, [28]. При ссылке на формулу или рисунок и т.п. из первоисточника следует указывать номера страниц, например [18, с.704]. Допускается приводить ссылки на литературу в подстрочном примечании.

Примеры библиографических описаний:

Монография (учебник, справочник) центрального издательства при числе авторов не более трех:

1. Б.Б. Кадомцев Коллективные явления в плазме Москва.: Наука, 1988. 304 с.

Монография (учебник, справочник) центрального издательства при числе авторов больше трех и наличии редактора (редакторов):

2. Конструирование экранов и СВЧ-устройств / А.М. Чернушенко, Б.В. Петров, Л.Г. Малорацкий и др.; Под. ред. А.М. Чернушенко Москва: Радио и связь, 1990. 351 с.

Отдельный том многотомного издания:

3. Савельев И.В. Курс общей физики. Т.1. Механика. Молекулярная физика: Учеб. пособие для студентов вузов. 2-е изд., перераб. М.: Наука, 1982.432 с.

Вузовские учебные пособия:

4. М.В. Кузелев, А.А. Рухадзе, П.С. Стрелков Плазменная релятивистская СВЧ-электроника: Учеб. пособие / Москва.: Издательство МГТУ им.Н.Э.Баумана, 2002. 543 с.

Периодические издания:

6. R. Benattar, C. Galas, P. Ney X-UV Index of refraction of dense and hot plasmas // Journal of X-ray Science and Technology. 1995. № 5. p.p. 249-260.

7. Взаимодействие электронного пучка с плазмой / И.Ф. Харченко, Я.Б. Файнберг, Р.Н. Николаев и др. // ЖЭТФ 1960. Т. 38, вып. 3. С. 685-692.

Материалы конференций:

9. А.С. Постникова, Б.В. Шишкин Система автоматизации для построения изображения объектов в терагерцовом диапазоне частот // Образовательные, научные и инженерные приложения в среде Labview и технологии National Instruments: Сборник трудов. междунар. науч.-практ. конф. / Москва. Издательство Российского университета дружбы народов, 2006. С. 259-262.

10. ECR plasmas and ECR Ion Sources / A.Girard, C.Lecot, G.Melin // 27th EPS Conference on Contr. Fusion and Plasma Phys. / Budapest, 2000. vol.24B(2000), p.p. 85-88.

Академическая этика

В курсовой работе и контрольных практических заданиях, используемые выдержки, идеи других авторов снабжаются сносками и отражаются в списке используемой литературы. Не допустимо включать в

свою работу выдержки из работ без указания на это, пересказывать чужую работу близко к тексту без отсылки к ней, использовать чужие идеи без указания первоисточников, включая электронные версии, распространяемые в Интернет. Все случаи плагиата должны быть исключены. В конце работы, согласно общих требований по оформлению отчетов самостоятельной работы, дается исчерпывающий список всех использованных источников.

ТЕМЫ ЛЕКЦИЙ И ПРАКТИЧЕСКИХ РАБОТ

1 СЕМЕСТР (сентябрь-январь)

Неделя 1: Лекция. Основы аналитического программирования и обработки данных. Системы компьютерной алгебры. Пакет Maple. Понятие об аналитических пакетах. История создания и развития. Основные сведения о пакете Maple: основы языка, структура подпрограмм и библиотек. Примеры моделирования простых физических и технических систем.

Литература:

Обязательная:

1. В.П. Дьяконов *Maple 9.5/10 в математике, физике и образовании. Серия «Библиотека профессионала. Из-во «Солон», 2006. 720 с.*
2. Е.Р. Алексеев, О.В. Чеснокова *Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9. Из-во «НТ Пресс», 2006. 496 с.*
3. Виктор Аладьев. *Система компьютерной алгебры Maple: Библиотека новых программных средств.*
www.lib.grsu.by/library/data/links/aladiev/Download/Maple_Lib.htm

Дополнительная:

1. *А. Н. Васильев Maple 8. Самоучитель, М.: Диалектика, Вильямс, 2003.*
2. *О.А. Сдвижков Математика на компьютере: Maple 8, Солон-пресс, 2003.*

Неделя 2: Лекция. Пакет MatLab – начальные сведения. Основные сведения о пакете Maple: выражения, командное окно, графика, основы языка, структура подпрограмм и библиотек. Примеры моделирования простых физических и технических систем. Система коллекций алгоритмов MATLAB (Toolboxes). Основные сведения о Simulink.

Литература:

Обязательная:

1. *Алексеев Е.Р., Чеснокова О.В. MATLAB 7. Самоучитель. Издательство "НТ Пресс", 2006г. 464 с.*
2. *Е.Р. Алексеев, О.В. Чеснокова Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9. Из-во «НТ Пресс», 2006. 496 с.*
3. *К.Э. Плехотников, Б.И. Волков, С.С. Задорожный, В.А. Антонюк Редактор: К.Э. Плехотников. Методы разработки курсовых работ. Моделирование, вычисления, программирование на C/C++ и MATLAB, виртуализация, образцы лучших студенческих курсовых работ. Из-во «Солон», 2006. 320 с.*

Дополнительная:

1. *Н.Н. Мартынов. Matlab 7. Элементарное введение. М: "Кудиц-Образ", 2005г, 416 с.*
2. *Поршнев С.В. MATLAB 7. Основы работы и программирования. Учебник. Издательство "Бином. Лаборатория знаний" 2006г. 320 с.*

Неделя 3: Лекция. Пакет MathCad. Сравнительный анализ трех численно-аналитических пакетов Maple, MatLab и MathCad. Основные сведения о пакете MathCad. Сравнение функциональных возможностей пакетов Maple, MatLab и MathCad. Варианты сопряжения пакетов между собой.

Литература:

Обязательная:

1. *Е.Р. Алексеев, О.В. Чеснокова Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9. Из-во «ИТ Пресс», 2006. 496 с.*
2. *В.П. Дьяконов Mathcad 11/12/13 в математике: справочник. Из-во «Горячая линия - Телеком», 2007. 958 с.*
3. *Д.В. Кирьянов Mathcad 14. Из-во ВХВ, 2007. 704 с.*
4. *Е. Макаров. Инженерные расчеты в Mathcad 14. Из-во «Питер», 2007. 592 с.*
5. *В.Говорухин, В.Цибулин. Компьютер в математическом исследовании: Maple, MATLAB, LaTeX. Питер. 2001.*

Интернет-ресурс

Stefan Steinhaus. Comparison of mathematical programs for data analysis. 2007. // <http://www.scientificweb.de/ncrunch/>

Дополнительная:

В.Очков. Физические и экономические величины в Mathcad и Maple. М.: Финансы и статистика, 2002.

Неделя 4: Лекция. Основы теории управления - 1. Динамические и частотные характеристики САУ. Анализ и синтез САУ методом корневого годографа. Описание систем в пространстве состояний. Передаточные

функции. Устойчивость линейных систем. Синтез оптимального управления с полной обратной связью. Фильтр Калмана и другие фильтры.

Литература:

Обязательная:

1. В. Гандер, И. Гржебичек *Решение задач в научных вычислениях с применением Maple и MATLAB. Из-во «Вассамедина», 2005. 520 с.*
2. Андриевский Б.Р., Фрадков А.Л. *Избранные главы теории автоматического управления с примерами в системе MATLAB. – СПб.: Наука, 1999.*

Дополнительная:

Г.А.Доррер. Основы теории управления. Красноярск: СибГТУ, 2003. 228 с.

Неделя 5: Лекция. Основы теории управления - 2. Устойчивость линейных систем. Синтез оптимального управления с полной обратной связью. Фильтр Калмана и другие фильтры. Примеры реализации систем управления в MatLab.

Литература:

Обязательная:

1. Андриевский Б.Р., Фрадков А.Л. *Избранные главы теории автоматического управления с примерами в системе MATLAB. – СПб.: Наука, 1999.*

Дополнительная:

1. *Г.А.Доррер. Основы теории управления. Красноярск: СибГТУ, 2003. 228 с.*

Неделя 6: Лекция. Симуляторы для применения в моделировании.

Общие понятия о симуляторах и генераторах данных. Случайные числа.

Генераторы случайных чисел. Применение случайных чисел и приемы работы с ними. Симуляторы в Simulink. Примеры.

Литература:

Обязательная:

1. В.Дьяконов. *Simulink 4. Специальный справочник*. СПб: Питер, 2002. 528 с.

Дополнительная:

1. Дж.Дэбни, Т.Харман. *Simulink 4. Секреты мастерства*. Издательство: Бином. Лаборатория знаний, 2003 г. 404 с.

Неделя 7: Лекция. Проектирование систем управления и реализация в численно-аналитических пакетах. Control System Toolbox - коллекция алгоритмов MATLAB для моделирования, анализа и проектирования систем автоматического управления. Сравнение традиционных методов передаточных функций и современных методов пространства состояний. Fuzzy Logic Toolbox – коллекция алгоритмов для использования методов нечеткой логики. Model Based Calibration Toolbox – гибкий инструмент для адаптивного моделирования систем.

Литература:

Обязательная:

1. Е.В.Никульчев. *Пособие "Control System Toolbox" MathWorks: Documentation (Release 14) \ Control System Toolbox // www.exponenta.ru*
2. Е.В.Никульчев. *Пособие "Control System Toolbox" MathWorks: Documentation (Release 14) \ Fuzzy Logic Toolbox // www.exponenta.ru*
3. Е.В.Никульчев. *Пособие "Control System Toolbox" MathWorks: Documentation (Release 14) \ Model Based Calibration Toolbox // www.exponenta.ru*

Дополнительная:

1. *Алексеев Е.Р., Чеснокова О.В. MATLAB 7. Самоучитель.*
Издательство "ИТ Пресс", 2006г. 464 с.

Неделя 8 : Лекция. Simulink – среда визуального имитационного и событийного моделирования. Моделирование замкнутой системы регулирования. Моделирование нелинейных систем и систем с распределенными параметрами. Организация экспериментов с Simulink.

Литература:

Обязательная:

1. *В.Дьяконов. Simulink 4. Специальный справочник. СПб: Питер, 2002.*
528 с.

Дополнительная:

1. *Дж.Дэбни, Т.Харман. Simulink 4. Секреты мастерства.*
Издательство: Бином. Лаборатория знаний, 2003 г. 404 с.

Неделя 9 : Лекция. Средства работы с данными в средах MatLab и Maple и статистическая обработка данных в численно-аналитических пакетах и в пакете Statistica.

Коллекции файлов для обработки данных в MatLab (Data Acquisition Toolboxes). Методы обработки информации, получаемой с подключенного РС-совместимого оборудования для сбора данных. Поддержка аналоговых входа и выхода и цифровых подсистем. Управляемый событиями сбор данных. Средства работы с данными в Maple. Коллекции Curve Fitting Toolbox и Statistics Toolbox в MatLab. Средства статистической обработки данных в Maple. Пакет Statistica.

Литература:

Обязательная:

1. Виктор Аладьев. Система компьютерной алгебры Maple: Библиотека новых программных средств.
2. В.Аладьев, М.Богдьявичюс. Maple 6: Решение математических, статистических и инженерно-физических задач. Лаборатория Базовых Знаний. 2001.
www.lib.grsu.by/library/data/links/aladiev/DownLoad/Maple_Lib.htm
3. В.З. Аладьев. Системы компьютерной алгебры. MAPLE: Искусство программирования. Бином. Лаборатория знаний, 2006.
4. MathWorks: Documentation (Release 14) \ Data Acquisition Toolbox // www.exponenta.ru
5. Дж.Смит. Сопряжение компьютеров с внешними устройствами. Уроки реализации. М.: Мир, 2000. 266 с.
6. MathWorks: Documentation (Release 14) \ Curve Fitting Toolbox // www.exponenta.ru

Дополнительная:

1. Алексеев Е.Р., Чеснокова О.В. MATLAB 7. Самоучитель. Издательство "НТ Пресс", 2006г. 464 с.
2. Иглин С.П. Теория вероятностей и математическая статистика на базе MATLAB. Издательство НТУ "ХПИ", 2006, Харьков, Украина, 612 с.

ПРАКТИЧЕСКИЕ РАБОТЫ ПО КУРСУ:

I. Изучение физических и технических систем на простых примерах

Практическая работа №1 «Изучение режимов колебания маятника с вибрирующим подвесом в Maple»

Практическая работа №2 «Решение дифференциальных уравнений механических систем в MatLab»

Практическая работа №3 «Изучение модели типа “хищник-жертва” в Maple, MatLab и MathCad»

Аннотация:

На сравнительно простых примерах в работах 1-3 приобретаются навыки изучения физических и технических систем с помощью численно-аналитических пакетов. Составляются дифференциальные уравнения, описывающие движение маятника Капицы и ряда простых механических систем и модели “хищник-жертва”. Полученные уравнения решаются точно и/или численно. Выясняются условия равновесия систем (в частности, условия стабилизации маятника Капицы в верхнем положении). Во всех случаях проводится визуализация полученных результатов (в том числе анимационная). Проводится сравнительный анализ решения одной и той же задачи с помощью трех пакетов; результаты сравнения сопоставляются с данными S.Steinhaus.

Литература:

Обязательная:

1. В.П. Дьяконов. *Maple 9 в математике, физике и образовании*. М.: СОЛОН-Пресс, 2004.
2. Алексеев Е.Р., Чеснокова О.В. *MATLAB 7. Самоучитель*. Издательство "ИТ Пресс", 2006г. 464 с.
3. Stefan Steinhaus. *Comparison of mathematical programs for data analysis*. 2007. // <http://www.scientificweb.de/ncrunch/>

4. В.Говорухин, В.Цибулин. *Компьютер в математическом исследовании: Maple, MATLAB, LaTeX. Питер. 2001.*

Дополнительная:

1. А. Н. Васильев *Maple 8. Самоучитель, М.: Диалектика, Вильямс, 2003.*
2. О.А. Сдвижков *Математика на компьютере: Maple 8, Солон-пресс, 2003.*
3. Н.Н. Мартынов. *Matlab 7. Элементарное введение. М: "Кудлиц-Образ", 2005г, 416 с.*
4. Поршнев С.В. *MATLAB 7. Основы работы и программирования. Учебник. Издательство "Бином. Лаборатория знаний" 2006г. 320 с.*

II. Основы моделирования в точных науках:

Практическая работа №4 «Simulink-модели наиболее распространенных передаточных функций и моделирование нелинейных моделей регулирования»

Практическая работа №5 «Синтез фильтра Калмана и его использование для стабилизации движения объекта»

Практическая работа №6 «Симулятор GPRS-ГЛОНАСС передатчика»

Практическая работа №7 «Расчет сигналов и модель диодного выпрямителя»

Практическая работа №8 «Моделирование систем с помощью клеточных автоматов»

Аннотация:

В работах 4-8 изучаются основные инструменты теории управления и их реализация в среде MatLab. Изучаются навыки работы с сигналами, расчета цепей, составления статистических моделей электронных устройств (диода), решения нелинейных уравнений. На примере имитации данных, поступающих от спутников систем глобального

позиционирования, изучаются приемы симуляции потоков данных, применяемых для тестирования и отладки моделей. Отрабатываются навыки составления моделей типа клеточных автоматов.

Литература:

Обязательная:

1. Андриевский Б.Р., Фрадков А.Л. *Избранные главы теории автоматического управления с примерами в системе MATLAB.* – СПб.: Наука, 1999.
2. Е.В.Никульчев. *Пособие "Control System Toolbox" MathWorks: Documentation (Release 14) \ Control System Toolbox //*
www.exponenta.ru
3. Е.В.Никульчев. *Пособие "Control System Toolbox" MathWorks: Documentation (Release 14) \ Model Based Calibration Toolbox //*
www.exponenta.ru
4. В.Дьяконов. *Simulink 4. Специальный справочник.* СПб: Питер, 2002. 528 с.

Дополнительная:

1. Г.А.Доррер. *Основы теории управления.* Красноярск: СибГТУ, 2003. 228 с.

III. Основы обработки данных в численно-аналитических пакетах:

Практическая работа №9 «Снятие частотных характеристик объекта с использованием виртуальных фазометра и измерителя затуханий»

Практическая работа №10 «Преобразование цифрового сигнала в аналоговый»

Аннотация:

В работах 9-10 изучаются основы моделирования систем обработки и преобразования данных. Работы базируются на инструментарии Simulink пакета MATLAB. На примере ЦАП-преобразователя изучаются

простейшие навыки сопряжения компьютера с внешними устройствами и организации обработки поступающих данных. Демонстрируются возможности системы MatLab создания драйверов, не уступающие C++.

Литература:

Обязательная:

1. В.Дьяконов. *Simulink 4. Специальный справочник*. СПб: Питер, 2002. 528 с.
2. Дж.Смит. *Сопряжение компьютеров с внешними устройствами. Уроки реализации*. М.: Мир, 2000. 266 с.
3. Алексеев Е.Р., Чеснокова О.В. *MATLAB 7. Самоучитель*. Издательство "НТ Пресс", 2006г. 464 с.

Дополнительная:

1. Р. Гонсалес, Р. Вудс, С. Эддинс *Цифровая обработка изображений в среде MATLAB*. Москва: Техносфера, 2006. - 616 с.

IV. Моделирование сложных физических и технических систем:

Практическая работа №11 «Модель Изинга: применение метода Монте-Карло и непосредственный расчет в среде Maple»

Практическая работа №12 «Расчет движения электрона в атоме водорода в среде Maple»

Практическая работа №13 «Расчет и моделирование движения волновых пакетов в среде с дисперсией в среде MatLab»

Практическая работа №14 «Самофокусировка света в нелинейной среде: расчет и моделирование в среде MatLab»

Практическая работа №15 «Система управления шаговым двигателем (с помощью MatLab-Simulink)»

Практическая работа №16 «Моделирование канонического ансамбля Гиббса методом Монте-Карло в пакете MATLAB»

Практическая работа №17 «Моделирование системы автоматического управления двигателем переменного тока»

Практическая работа №18 «Движение пылинки под действием стоячей звуковой волны в резонаторе – расчет и моделирование в среде Maple»

Аннотация:

В работах 11-18 студенты осваивают новые и закрепляют ранее изученные методы и навыки путем моделирования сложных систем. Исследования производятся путем решения (точного или численного) дифференциальных уравнений и в ряде случаев с помощью имитационного моделирования. Применяются изученные способы обработки и визуализации данных и управления сложными системами.

Литература:

Обязательная:

1. Д.П. Голоскоков *Уравнения математической физики. Решение задач в системе Maple. С-Пб: Питер, 2004.*
2. А. Н. Васильев *Maple 8. Самоучитель, М.: Диалектика, Вильямс, 2003.*

Дополнительная:

1. В.Аладьев, М.Богдьявичюс. *Maple 6: Решение математических, статистических и инженерно-физических задач. Лаборатория Базовых Знаний. 2001*

Календарный план курса

Виды и содержание учебных занятий				
Неде- ля	Лекции	Число часов	Практические занятия	Число часов
1	Основы аналитического программирования и обработки данных. Пакет Maple	2	Практическая работа из раздела I. Изучение физических и технических систем на простых примерах*	4
2	Пакет MatLab – начальные сведения	2		
3	Пакет MathCad и сравнительный анализ трех численно-аналитических пакетов Maple, MatLab и MathCad.	2	Практическая работа из раздела II. Основы моделирования в точных науках*	4
4	Maple, MatLab и MathCad	2		
5	Основы теории управления - 1.	2	Практическая работа из раздела II. Основы моделирования в точных науках*	4
6	Основы теории управления - 2.	2		
7	Проектирование систем управления и реализация в численно-аналитических пакетах	2	Практическая работа из раздела II. Основы моделирования в точных науках*	4
8	Simulink – среда визуального имитационного и событийного моделирования	2		
9	Промежуточный контроль знаний			8
9	Средства работы с данными в средах MatLab и Maple и статистическая обработка данных в численно-аналитических пакетах и в пакете Statistica	2	Практическая работа из раздела III. Основы обработки данных в численно-аналитических пакетах*	6
10				

11			Практическая работа из раздела III. Основы обработки данных в численно-аналитических пакетах*	8
12				
13			Практическая работа из раздела IV. Моделирование сложных физических и технических систем	8
14				
15			Практическая работа из раздела IV. Моделирование сложных физических и технических систем*	8
16				
17			Практическая работа из раздела IV. Моделирование сложных физических и технических систем*	8
18				
19	Сдача контрольного практического задания (курсовой)			8
20	Итоговый контроль знаний			8

*Примечание: Преподаватель выбирает одну из работ соответствующего раздела.

Примерный перечень тем курсовых работ и контрольных практических заданий

Темы курсовых работ:

1. Исследование движения маятника с вибрирующим подвесом с горизонтальной составляющей: возможность стабилизации в любом положении.
2. Перевернутый маятник со смещаемой точкой подвеса: составление математического обеспечения для реальной дискретно-позиционной управляющей системы и имитация моделирования в MatLab.
3. Система управления асинхронным двигателем.
4. Моделирование системы связанных осцилляторов (на примере маятников и колебательных контуров).
5. Моделирование маятника Дубочинского.
6. Имитационное моделирование квантовых процессов в Simulink.
7. Квантовая механика в MatLab.
8. Квантовая механика в MathCad.
9. Задачи физической электроники в Maple.
10. Задачи физики плазмы в Maple.
11. Система стабилизации курса движения самолета (имитация в Simulink).

Темы контрольных практических заданий:

1. Преобразование аналогового сигнала в цифровой.
2. Расчет маятника Челомея.
3. Моделирование большого канонического ансамбля методом Монте-Карло в пакете MATLAB.
4. Движение пылинки под действием стоячей звуковой волны в резонаторе – расчет и моделирование в среде MatLab.

5. Движение частицы в поле заданного потенциала: исследование с применением Maple, MatLab и MathCad.
6. Составление программы, имитирующей случайные блуждания.
7. Интегрирование уравнений движения заряда в скрещенных электрическом и магнитном полях (с помощью Maple).
8. Интегрирование уравнений Лагранжа для заряженных релятивистских частиц в Maple.
9. Решение основной задачи динамики при криволинейном движении материальной точки для двух случаев: а) без учета силы сопротивления воздуха; б) с учетом силы сопротивления воздуха, пропорциональной скорости движения точки (в среде Maple и MatLab).
10. Расчеты процессов квантового туннелирования в Maple.
11. Моделирование колебаний двухатомной молекулы (квазиклассическое приближение) в Maple.
12. Анализ и визуализация состояний трехмерного изотропного квантового осциллятора с помощью пакетов Maple, Mathcad.
13. Расчет максимальной энергии электрона в условиях электронного циклотронного резонанса в однородном магнитном поле.
14. Одномерное отражение частиц от прямоугольной потенциальной ступеньки. Вычисления в среде Maple.

Использованные литературные источники.

1. *Межгосударственного стандарта -ГОСТ 7.32-2001 «Отчет о научно-исследовательской работе»*
2. *ГОСТ 8.417-81 "ГСИ. Единицы физических величин".*