

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ  
Государственное образовательное учреждение  
высшего профессионального образования  
«Оренбургский государственный университет»

Кафедра металлообрабатывающих станков и комплексов

А.Н.ПОЛЯКОВ

**ИСПОЛЬЗОВАНИЕ СИСТЕМЫ  
МАТЛАВ В МАТЕМАТИЧЕСКОМ  
МОДЕЛИРОВАНИИ СТАНКОВ.  
БАЗОВЫЕ ПОЛОЖЕНИЯ  
СИСТЕМЫ**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Рекомендовано к изданию Редакционно – издательским советом  
государственного образовательного учреждения высшего профессионального  
образования «Оренбургский государственный университет»

Оренбург 2005

УДК 681.7. 053.2 (075.8)

ББК 34.7 я73

П 78

Рецензент

кандидат технических наук, доцент И.П. Никитина

П 78

**Поляков А.Н.**

**Использование системы MATLAB в математическом моделировании станков. Базовые положения системы: методические указания/А.Н. Поляков - Оренбург: ГОУ ОГУ, 2005. – 76с.**

Методическое указание содержит ряд начальных сведений по использованию автоматизированной системы математических вычислений MATLAB в инженерной практике. Приводятся общие, наиболее часто встречающиеся частные математические задачи, на основе решений которых строится решение конкретных инженерных задач.

Данное учебное пособие выполнено в рамках программ по дисциплинам кафедры металлообрабатывающих станков и комплексов: технология создания программ и информационные среды, математическое моделирование станков. Учебное пособие рекомендуется использовать при выполнении курсовых и дипломных проектов для специальностей 151002, а также в инженерных расчетах при решении практических задач.

ББК 34.7я73

© Поляков А.Н., 2005

© ГОУ ОГУ, 2005

## Введение

Современная компьютерная математика предлагает целый набор интегрированных программных систем и пакетов программ для автоматизации математических расчетов: Eureka, Gauss, TK Solver!, Derive, Mathcad, Mathematica, Maple V, Matlab и др.

MATLAB — одна из старейших, тщательно проработанных и проверенных временем систем автоматизации математических расчетов, построенная на расширенном представлении и применении матричных операций. Это нашло отражение в названии системы — MATrix LABoratory — матричная лаборатория. 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.

Однако в настоящее время MATLAB далеко вышла за пределы специализированной матричной системы и стала одной из наиболее мощных универсальных интегрированных СКМ. 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 и 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. И вот теперь появилась новейшая версия этой системы — MATLAB 6. Одной из основных задач системы было предоставление 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, 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.

В обширном и постоянно пополняемом комплексе команд, функций и прикладных программ (пакетов расширения, пакетов инструментов, «toolbox» - пакет инструментов, пакет расширения, прикладная программа — почти синонимы при переводе термина toolbox, но пакет инструментов собственно MATLAB 6 рассматривается как один из toolbox всей системы, включающей MATLAB 6, Simulink и другие пакеты.

Система MATLAB включает специальные средства для электротехнических и радиотехнических расчетов (операции с комплексными числами, матрицами, векторами и полиномами, обработка данных, анализ сигналов и цифровая фильтрация), обработки изображений, реализации нейронных сетей, а также средства, относящиеся к другим новым направлениям науки и техники. Они иллюстрируются множеством практически полезных примеров. К разработкам расширений для системы MATLAB привлечены многие научные школы мира и руководящие ими крупные ученые и педагоги университетов.

Важными достоинствами системы являются ее открытость и расширяемость. Большинство команд и функций системы реализованы в виде текстовых m-файлов (с расширением .m) и файлов на языке Си, причем все файлы доступны для модификации. Пользователю дана возможность создавать не только отдельные файлы, но и библиотеки файлов для реализации специфических задач.

Поразительная легкость модификации системы и возможность ее адаптации к решению специфических задач науки и техники привели к созданию десятков пакетов прикладных программ (toolbox), намного расширивших сферы применения системы. Некоторые из них, например Notebook (интеграция с текстовым процессором Word и подготовка «живых» электронных книг), Symbolic Math и Extended Symbolic Math (символьные вычисления с применением ядра системы Maple V R5) и Simulink (моделирование динамических систем и устройств, заданных в виде системы блоков), настолько органично интегрировались с системой MATLAB, что стали ее составными частями.

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 — увы, как математики так 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 последних версий MATLAB — Notebook — позволяет готовить документы в текстовом процессоре Word 95/97/2000 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 версии MATLAB 6 предусмотрены также улучшенные средства для экспорта графики в слайды презентационной программы Microsoft PowerPoint.

В состав системы MATLAB входит ядро одной из самых мощных, популярных и хорошо апробированных систем символьной математики Maple V Release 5. 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.

# 1 Введение в систему MATLAB

## 1.1 MATLAB в роли суперкалькулятора

Система MATLAB создана таким образом, что любые (подчас весьма сложные) вычисления можно выполнять в режиме прямых вычислений, то есть без подготовки программы. Это превращает MATLAB в необычайно мощный калькулятор, который способен производить не только обычные для калькуляторов вычисления (например, выполнять арифметические операции и вычислять элементарные функции), но и операции с векторами и матрицами, комплексными числами, рядами и полиномами. Можно почти мгновенно задать и вывести графики различных функций — от простой синусоиды до сложной трехмерной фигуры.

Работа с системой в режиме прямых вычислений носит диалоговый характер и происходит по правилу «задал вопрос, получил ответ». Пользователь набирает на клавиатуре вычисляемое выражение, редактирует его (если нужно) в командной строке и завершает ввод нажатием клавиши ENTER:

- 1) для блокировки вывода результата вычислений некоторого выражения после него надо установить знак ; (точка с запятой);
- 2) если не указана переменная для значения результата вычислений, то MATLAB назначает такую переменную с именем ans;
- 3) знаком присваивания является знак равенства «=», а не комбинированный знак :=, как во многих других языках программирования и математических системах;
- 4) результат вычислений выводится в строках вывода;
- 5) встроенные функции (например, sin) записываются строчными буквами, и их аргументы указываются в круглых скобках;
- 6) диалог происходит в стиле «задал вопрос — получил ответ».

## 1.2 Перенос строки в сессии

В некоторых случаях вводимое математическое выражение может оказаться настолько длинным, что для него не хватит одной строки. В этом случае часть выражения можно перенести на новую строку с помощью знака многоточия «...» (3 или более точек), например:

$$s = 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 ... \\ 1/8 + 1/9 - 1/10 + 1/11 - 1/12;$$

Максимальное число символов в одной строке командного режима — 4096, а в m-файле — не ограничено, но со столь длинными строками работать неудобно. В ранних версиях в одной строке было не более 256 символов.

## 1.3 Основные объекты MATLAB

### 1.3.1 Математическое выражение

Центральным понятием всех математических систем является математическое выражение. Оно задает то, что должно быть вычислено в численном (реже символьном) виде:

```
2+3
2.301*sin(x)
4+exp(3)/5
sqrt(y)/2
sin(pi/2)
```

### 1.3.2 Действительные и комплексные числа

Число — простейший объект языка MATLAB, представляющий количественные данные. Числа можно считать константами, имена которых совпадают с их значениями. Числа используются в общепринятом представлении о них. Они могут быть целыми, дробными, с фиксированной и плавающей точкой. Возможно представление чисел в хорошо известном научном формате с указанием мантиссы и порядка числа. Пробелы между символами в числах не допускаются. Числа могут быть комплексными:  $z = \text{Re}(z) + \text{Im}(z) \cdot i$ . Такие числа содержат действительную  $\text{Re}(z)$  и мнимую  $\text{Im}(z)$  части. Мнимая часть имеет множитель  $i$  или  $j$ , означающий корень квадратный из  $-1$ . Функция `real(z)` возвращает действительную часть комплексного числа,  $\text{Re}(z)$ , а функция `imag(z)` — мнимую,  $\text{Im}(z)$ . Для получения модуля комплексного числа используется функция `abs(z)`, а для вычисления фазы — `angle(Z)`.

В MATLAB не принято делить числа на целые и дробные, короткие и длинные и т. д. Операции над числами выполняются в формате, который принято считать форматом с двойной точностью. Такой формат удовлетворяет подавляющему большинству требований к численным расчетам, но совершенно не подходит для символьных вычислений с произвольной (абсолютной) точностью. Символьные вычисления MATLAB может выполнять с помощью специального пакета расширения `Symbolic Math Toolbox`.

### 1.3.3 Константы и системные переменные

Константа — это предварительно определенное числовое или символьное значение, представленное уникальным именем. Числа (например 1, -2 и 1.23) являются безымянными числовыми константами. Другие виды констант в MATLAB принято называть системными переменными, поскольку,



с одной стороны, они задаются системой при ее загрузке, а с другой — могут переопределяться. Основные системные переменные, применяемые в системе MATLAB:

- 1)  $i$  или  $j$  — мнимая единица (корень квадратный из  $-1$ );
- 2)  $\pi$  - число  $\pi$  - 3.1415926...;
- 3)  $\text{eps}$  — погрешность операций над числами с плавающей точкой;
- 4)  $\text{realmin}$  — наименьшее число с плавающей точкой;
- 5)  $\text{realmax}$  — наибольшее число с плавающей точкой;
- 6)  $\text{inf}$  — значение машинной бесконечности;
- 7)  $\text{ans}$  — переменная, хранящая результат последней операции и обычно вызывающая его отображение на экране дисплея;
- 7)  $\text{NaN}$  — указание на нечисловой характер данных (Not-a-Number).

Символьная константа — это цепочка символов, заключенных в апострофы, например:

```
'Hello my friend!'  
'2+3'
```

Если в апострофы помещено математическое выражение, то оно не вычисляется и рассматривается просто как цепочка символов. С помощью специальных функций преобразования символьные выражения могут быть преобразованы в вычисляемые.

### 1.3.4 Переменные и присваивание им значений

Переменные — это имеющие имена объекты, способные хранить некоторые, обычно разные по значению, данные. В зависимости от этих данных переменные могут быть числовыми или символьными, векторными или матричными. В системе MATLAB можно задавать переменным определенные значения. Для этого используется операция присваивания, вводимая знаком равенства  $=$ .

Типы переменных заранее не декларируются. Они определяются выражением, значение которого присваивается переменной. Так, если это выражение — вектор или матрица, то переменная будет векторной или матричной.

Имя переменной (ее идентификатор) может содержать сколько угодно символов, но запоминается и идентифицируется только 31 начальный символ. Имя любой переменной не должно совпадать с именами других переменных, функций и процедур системы, т. е. оно должно быть уникальным. Имя должно начинаться с буквы, может содержать буквы, цифры и символ подчеркивания  $\_$ . Недопустимо включать в имена переменных пробелы и специальные знаки, например «+, -, \*, / и т. д.».

### 1.3.5 Уничтожение определений переменных

В памяти компьютера переменные занимают определенное место, называемое рабочей областью (workspace). Для очистки рабочей области используется функция `clear` в разных формах, например:

- 1) `clear` — уничтожение определений всех переменных;
- 2) `clear x` — уничтожение определения переменной `x`;
- 3) `clear a, b, c` — уничтожение определений нескольких переменных.

Уничтоженная переменная становится неопределенной. Использовать неопределенные переменные нельзя, и такие попытки будут сопровождаться выдачей сообщений об ошибке.

### 1.3.6 Операторы и функции

Оператор — это специальное обозначение для определенной операции над данными — операндами. Например, простейшими арифметическими операторами являются знаки суммы `+`, вычитания `-`, умножения `*` и деления `/`. Операторы используются совместно с операндами. Например, в выражении `2+3` знак `+` является оператором сложения, а числа `2` и `3` — операндами.

Большинство операторов в Matlab относится к матричным операциям, что может служить причиной ошибок. Например, операторы умножения `*` и деления `/` вычисляют произведение и частное от деления двух многомерных массивов, векторов или матриц. Есть ряд специальных операторов, например, оператор «`\`» означает деление справа налево, а операторы «`.*`» и «`./`» означают соответственно поэлементное умножение и поэлементное деление массивов.

Полный список операторов можно получить, используя команду `» help ops`.

Функции - это объекты, имеющие уникальные имена, и выполняющие определенные преобразования своих аргументов и при этом возвращающие результаты этих преобразований. Возврат результата - отличительная черта функций. При этом результат вычисления функции с одним выходным параметром подставляется на место ее вызова, что позволяет использовать функции в математических выражениях, например функцию `sin` в `2*sin(pi/2)`.

Функции в общем случае имеют список аргументов (параметров), заключенный в круглые скобки. Например, функция Бесселя записывается как `bessel(NU,X)`. В данном случае список параметров содержит два аргумента — `NU` в виде скаляра и `X` в виде вектора. Многие функции допускают ряд форм записи, отличающихся списком параметров. Если функция возвращает несколько значений, то она записывается в виде `[Y1, Y2,...]=func(X1, X2,...)`, где `Y1, Y2,...` - список выходных параметров и `X1, X2....` — список входных аргументов (параметров).

Со списком элементарных функций можно ознакомиться, выполнив команду «`help elfun`», а со списком специальных функций - с помощью команды «`help specfun`». Функции могут быть встроенными (внутренними) и внешними, или `m`-функциями. Так, встроенными являются наиболее распространенные элементарные функции например, `sin(x)` и `exp(y)`, тогда

как функция  $\sinh(x)$  является внешней функцией. Внешние функции содержат свои определения в m-файлах. Встроенные функции хранятся в откомпилированном ядре системы MATLAB, в силу чего они выполняются предельно быстро.

Очень часто необходимо произвести формирование упорядоченных числовых последовательностей. Такие последовательности нужны для создания векторов или значений абсциссы при построении графиков. Для этого в MATLAB используется оператор : (двоеточие):

**«Начальное\_значение:Шаг:Конечное\_значение»**

Данная конструкция порождает возрастающую последовательность чисел, которая начинается с начального значения, идет с заданным шагом и завершается конечным значением. Если Шаг не задан, то он принимает значение 1. Если конечное значение указано меньшим, чем начальное значение, - выдается сообщение об ошибке.

В системе MATLAB внешние определения используются точно так же, как и встроенные функции и операторы. Никаких дополнительных указаний на их применение делать не надо. Достаточно лишь позаботиться о том, чтобы используемые определения действительно существовали в виде файлов с расширением «.m».

### **1.3.7 Формирование векторов и матриц**

Так как MATLAB - система, специально предназначенная для проведения сложных вычислений с векторами, матрицами и массивами, то в ней по умолчанию предполагается, что каждая заданная переменная - это вектор, матрица или массив. Все определяется конкретным значением переменной. Например, если задано  $X=1$ , то это значит, что X -это вектор с единственным элементом, имеющим значение 1. Если надо задать вектор из трех элементов, то их значения следует перечислить в квадратных скобках, разделяя пробелами. Задание матрицы требует указания нескольких строк. Для разграничения строк используется знак «;» (точка с запятой). Этот же знак в конце ввода предотвращает вывод матрицы или вектора (и вообще результата любой операции) на экран дисплея.

Возможен ввод элементов матриц и векторов в виде арифметических выражений, содержащих любые доступные системе функции.

Для указания отдельного элемента вектора или матрицы используются выражения вида  $V(1)$  или  $M(i, j)$ .

Если нужно присвоить элементу  $M(i,j)$  /В тексте программ MATLAB лучше не использовать  $i$  и  $j$  как индексы, так как  $i$  и  $j$  - обозначение квадратного корня из -1./ новое значение  $x$ , следует использовать выражение:

**$M(i, j)=x;$**

Выражение  $M(i)$  с одним индексом дает доступ к элементам матрицы, развернутым в один столбец. Такая матрица образуется из исходной, если подряд выписать ее столбцы, например, для матрицы «M»:

```
» M=[1 2 3; 4 5 6; 7 8 9]
```

```
M =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
» M(2)
```

```
ans =
```

```
4
```

```
» M(8)
```

```
ans =
```

```
6
```

```
» M(9)
```

```
ans =
```

```
9
```

```
» M(5)=100;
```

```
» M
```

```
M =
```

```
1 2 3
```

```
4 100 6
```

```
7 8 9
```

Возможно задание векторов и матриц с комплексными элементами, например:

```
» i=sqrt(-1);
```

```
» CM=[1 2; 3 4] + i*[5 6; 7 8]
```

Это создает матрицу:

```
CM =
```

```
1.0000 + 5.0000i 2.0000 + 6.0000i
```

```
3.0000 + 7.0000i 4.0000 + 8.0000i
```

Наряду с операциями над отдельными элементами матриц и векторов система позволяет производить операции умножения, деления и возведения в степень сразу над всеми элементами, т. е. над массивами. Для этого перед знаком операции ставится точка. Например, оператор «\*» означает умножение для векторов или матриц, а оператор «.\*» - поэлементное умножение всех элементов массива. Так, если  $M$  - матрица, то « $M.*2$ » даст матрицу, все элементы которой умножены на скаляр - число 2.

Описанный способ задания матриц позволяет выполнить операцию конкатенации - объединения малых матриц в большую. Например, создадим вначале матрицу  $A$  размера  $3 \times 3$ , а затем построим матрицу, содержащую четыре матрицы:

```
B=[A A+16; A+32 A+16]
```

Полученная матрица имеет уже размер 6х6. Запись «В'» означает транспонирование матрицы В, т. е. замену строк столбцами.

Для формирования матриц и выполнения ряда матричных операций возникает необходимость удаления отдельных столбцов и строк матрицы. Для этого используются пустые квадратные скобки «[ ]». Проведем это с матрицей М:

```
>> M=[1 2 3; 4 5 6; 7 8 9]
M =
     1     2     3
     4     5     6
     7     8     9
>> M(:,2)=[];
>> M
M =
     1     3
     4     6
     7     9
```

#### 1.4 Сохранение рабочей области сессии

Переменные и определения новых функций в системе MATLAB хранятся в особой области памяти, именуемой рабочей областью. MATLAB позволяет сохранять значения переменных в виде бинарных файлов с расширением .mat Для этого служит команда save, которая может использоваться в ряде форм:

- 1) save fname - записывается рабочая область всех переменных в файле бинарного формата с именем fname.mat;
- 2) save fname X - записывает только значение переменной X;
- 3) save fname X Y Z - записывает значения переменных X, Y и Z.

#### 1.5 Ведение дневника

Сессии не записываются на диск стандартной командой save. Однако если такая необходимость есть, можно воспользоваться специальной командой для ведения так называемого дневника сессии:

- 1) diary file name — ведет запись на диск всех команд в строках ввода и полученных результатов в виде текстового файла с указанным именем;
- 2) diary off - приостанавливает запись в файл;
- 3) diary on - вновь начинает запись в файл.

Таким образом, чередуя команды diary off и diary on, можно сохранять нужные фрагменты сессии в их формальном виде. Команду diary можно задать и в виде функции diary('file'), где строка 'file' задает имя файла.

Команда «type» позволяет просмотреть текст такого файла со всеми записанными действиями. Рекомендуется записывать файл с расширением,

отличным от «.m», например «.txt». Это позволит встраивать подобные текстовые файлы дневника сессии в документы, содержащие ее описание.

## **1.6 Загрузка рабочей области сессии**

Для загрузки рабочей области ранее проведенной сессии (если она была сохранена) можно использовать команду «load»: «load fname» - загрузка ранее сохраненных в файле «fname.mat» определений со спецификациями на месте многоточия, подобными описанным для команды save (включая ключ -mat для загрузки файлов с расширением .mat обычного бинарного формата, используемого по умолчанию). Загрузка файла fname.mat в форме функции осуществляется в виде load( 'fname'...). Если команда (или функция) «load» используется в ходе проведения сессии, то произойдет замена текущих значений переменных теми значениями, которые были сохранены в считываемом MAT-файле.

Для задания имен загружаемых файлов может использоваться знак «\*», означающий загрузку всех файлов с определенными признаками. Например, load demo\*.mat означает загрузку всех файлов с началом имени demo, например demo1, demo2, demoa, demob и т. д. Имена загружаемых файлов можно формировать с помощью операций над строковыми выражениями.

## **1.7 Завершение вычислений и работы с системой**

### **1.7.1 Завершение вычислений**

Иногда из-за ошибок в программе или из-за сложности решаемой задачи MATLAB «зацикливается» и перестает выдавать результаты либо непрерывно выдает их, хотя в этом уже нет необходимости. Для прерывания вычислений в этом случае достаточно нажать одновременно клавиши Ctrl и C (латинское).

### **1.7.2 Завершение работы с системой**

Для завершения работы с системой можно использовать команды exit, quit (которые сохраняют содержимое рабочей области и выполняет другие действия в соответствии с файлом сценария finish.m) или комбинацию клавиш Ctrl+Q. Если необходимо сохранить значения всех переменных (векторов, матриц) системы, то перед вводом команды exit следует дать команду save нужной формы. Команда load после загрузки системы считывает значения этих переменных и позволяет начать работу с системой с того момента, когда она была прервана.

## 2 Основы графической визуализации вычислений

### 2.1 Построение графика функций одной переменной

В режиме непосредственных вычислений доступны практически все возможности системы. Широко используется, например, построение графиков различных функций, дающих наглядное представление об их поведении в широком диапазоне изменения аргумента. При этом графики строятся в отдельных масштабируемых и перемещаемых окнах.

Возьмем простейший пример - построение графика синусоиды. MATLAB строит графики функций по ряду точек, соединяя их отрезками прямых, т. е. осуществляя линейную интерполяцию функции в интервале между смежными точками. Зададим интервал изменения аргумента  $x$  от 0 до 10 с шагом 0.1. Для построения графика достаточно вначале задать вектор « $x=0:0.1:10$ », а затем использовать команду построения графиков `plot(sin(x))`.

Графики MATLAB строит в отдельных окнах, называемых графическими окнами. В главном меню окна появилась позиция Tools (Инструменты), которая позволяет вывести или скрыть инструментальную панель, видимую в верхней части окна графики. Средства этой панели позволяют легко управлять параметрами графиков и наносить на них текстовые комментарии в любом месте.

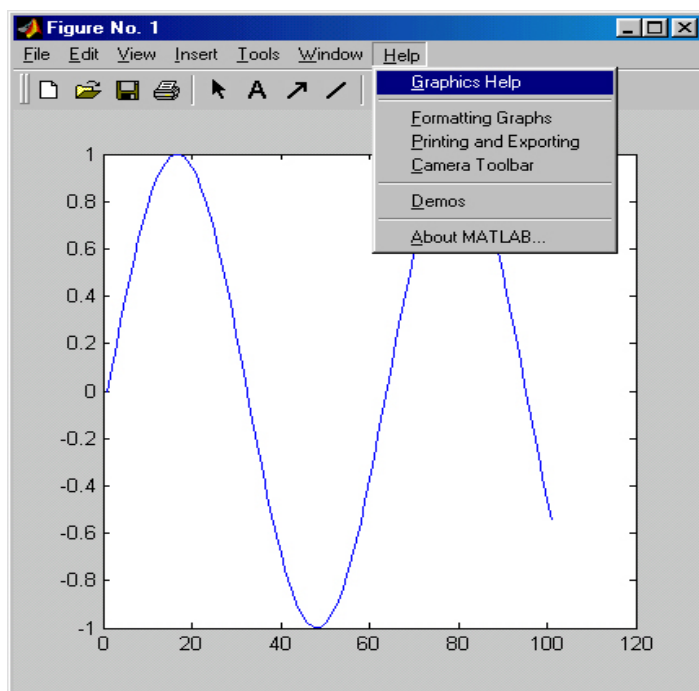


Рисунок 2.1 - Построение синусоиды

### 2.2 Построение в одном окне графиков нескольких функций

Построим графики трех функций:  $\sin(x)$ ,  $\cos(x)$  и  $\sin(x)/x$ . Прежде всего отметим, что эти функции могут быть обозначены переменными, не имеющими явного указания аргумента в виде  $y(x)$ :

```
y1=sin(x); y2=cos(x); y3=sin(x)./x;
```

Такая возможность обусловлена тем, что эти переменные являются векторами — как и переменная  $x$ . Теперь можно использовать одну из ряда форм команды:

```
plot: plot(a1,f1,a2,f2,a3,f3,...),
```

где  $a1$ ,  $a2$ ,  $a3$  - векторы аргументов функций (в данном случае все они -  $x$ );  
 $f1$ ,  $f2$ ,  $f3$  - векторы значений функций, графики которых строятся в одном окне.

В данном примере для построения графиков указанных функций мы должны записать следующее:

```
plot(x,y1,x,y2,x,y3)
```

### 2.3 Графическая функция fplot

Эта функция позволяет строить функцию, заданную в символьном виде, в интервале изменения аргумента « $x$ » от  $x_{\min}$  до  $x_{\max}$  без фиксированного шага изменения  $x$ , например:

```
fplot('tanh',[-2 2])
```

Результат представлен на рисунке 2.2

### 2.4 Столбцовые диаграммы

В прикладных расчетах часто встречаются графики, именуемые столбцовыми диаграммами, отражающие содержание некоторого вектора. При этом каждый элемент вектора представляется столбцом, высота которого пропорциональна значению элемента. Столбцы нумеруются и масштабируются по отношению к максимальному значению наиболее высокого столбца. Выполняет построение такого графика команда `bar(V)`.

Столбцовые диаграммы лишь один из многих типов графиков, которые может строить система MATLAB.



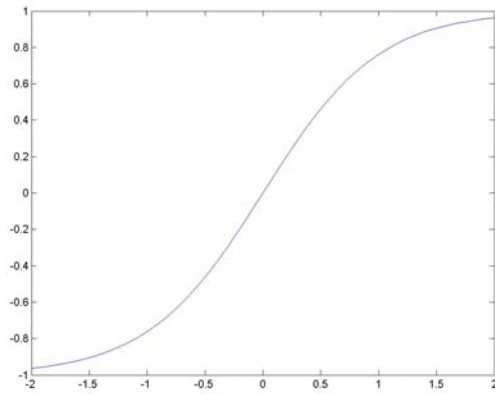


Рисунок 2.2 - Результат работы функции `fplot`

## 2.5 Построение трехмерных графиков

Для построения графика поверхности и ее проекции в виде контурного графика на плоскость под поверхностью достаточно использовать следующие команды:

```
[X,Y] = meshgrid(-3:.125:3);
Z = peaks(X,Y);
meshc(X,Y,Z);
axis([-3 3 -3 3 -10 5])
```

Здесь первая функция устанавливает диапазон изменения для аргументов X и Y. Вторая функция генерирует условный пик (по закону нормального распределения). Третья функция собственно строит поверхность, а четвертая функция показывает оси координат в заданном диапазоне (рисунок 2.3).

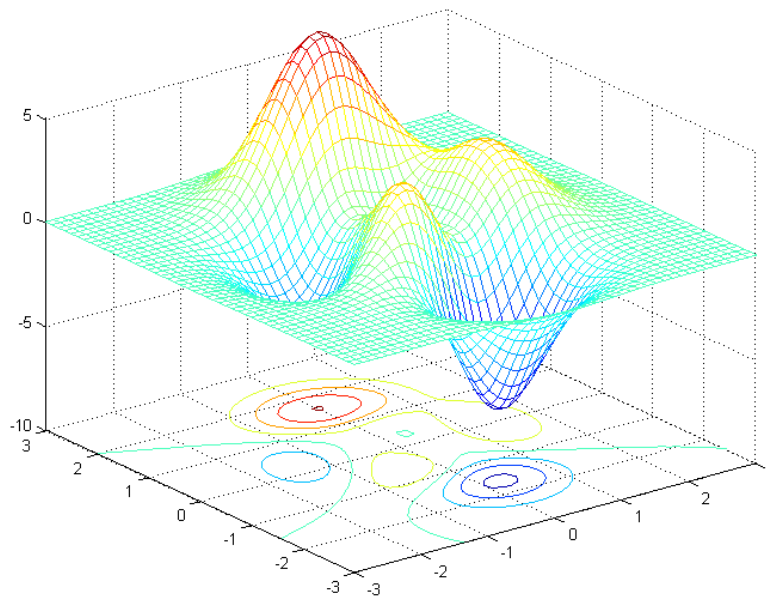


Рисунок 2.3 – Трехмерное представление поверхности

## **2.6 Перемещение графика в графическом окне**

Обычно график занимает фиксированное положение в центре графического окна. Однако в режиме редактирования графиков, когда курсор мыши находится в области графика, в контекстном меню правой клавиши мыши есть команда `Unlock Axes Position` (Отключить позиционирование осей). Она снимает фиксацию положения координатных осей графика и позволяет двигать его мышью вместе с осями.

## 3 Базовые вычисления

### 3.1 Вычисление собственных значений

Во многих областях математики и прикладных наук большое значение имеют средства для вычисления собственных значений матриц и принадлежащих им векторов:

1)  $\text{eig}(A)$  - возвращает вектор собственных значений квадратной полной или симметрической разреженной матрицы  $A$  обычно после автоматического масштабирования, но для больших разреженных матриц (в терминологии MATLAB - это просто полные матрицы со сравнительно большим числом нулей), а также во всех случаях, где помимо собственных значений необходимо получать и собственные вектора разреженной матрицы, вместо нее рекомендовано использовать  $\text{eigs}(A)$ ;

2)  $\text{eig}(A,B)$  - возвращает вектор обобщенных собственных значений квадратных матриц  $A$  и  $B$  для типовой задачи вида:

$$Ax = \lambda Bx,$$

где  $x$  – собственные векторы;  $\lambda$  - собственные значения.

### 3.2 Решения СЛУ

#### 3.2.1 Элементарные средства

Решение систем линейных уравнений (СЛУ) относится к самой массовой области применения матричных методов.

Для случая нескаллярных  $A$  и  $B$  число столбцов матрицы  $A$  должно равняться числу строк матрицы  $B$ . Скаляр может умножаться на матрицу любого размера.

1) «/» - правое деление. Выражение « $X=B/A$ » дает решение ряда систем линейных уравнений « $AX=B$ », где  $A$  - матрица размера  $m \times n$  и  $B$  — матрица размера  $n \times k$ ;

2) «\» — левое деление. Выражение  $X=B \setminus A$  дает решение ряда систем линейных уравнений  $XA=B$ , где  $A$  — матрица размера  $m \times n$  и  $B$  — матрица размера  $n \times k$ . Если  $A$  — квадратная матрица, то  $A \setminus B$  - примерно то же самое, что и  $\text{inv}(A)*B$ , в остальных случаях возможны варианты, отмеченные ниже.

Если  $A$  - матрица размера  $n \times n$ , а  $B$  — вектор-столбец с  $n$  компонентами или матрица с несколькими подобными столбцами, тогда  $X=A \setminus B$  — решение уравнения  $AX=B$ , которое находится хорошо известным методом исключения Гаусса.

### 3.2.2 Функции для решения систем линейных уравнений с ограничениями

Для решения систем линейных уравнений с ограничениями методом наименьших квадратов введены функции:

1)  $X = \text{lscov}(A,B,V)$  — возвращает вектор  $X$  решения СЛУ вида  $A \cdot X = B + e$ , где « $e$ » — вектор шумов, который имеет ковариационную матрицу  $V$ . Реализуется метод наименьших квадратов в присутствии шумов с известной ковариацией. Прямоугольная матрица  $A$  должна быть размера  $m \times n$ , где  $m > n$ . При решении минимизируется следующее выражение: « $(AX - b)' \cdot \text{inv}(V) \cdot (AX - b)$ ». Решение имеет вид  $X = \text{inv}(A' \cdot \text{inv}(V) \cdot A) \cdot A' \cdot \text{inv}(V) \cdot B$ ;

2)  $[X, dX] = \text{lscov}(A,B,V)$  — возвращает также стандартную погрешность  $X$ , помещая ее в переменную  $dX$ ;

3)  $X = \text{lsqnonneg}(A,B)$  — решение СЛУ  $AX = B$  методом наименьших квадратов с неотрицательными ограничениями.  $A$  — действительная прямоугольная матрица,  $B$  — действительный вектор. Вектор  $X$  содержит неотрицательные элементы  $X_j \geq 0$ , где  $j = 1, 2, \dots, n$ ;

4)  $X = \text{lsqnonneg}(A,B,X_0)$  — решение СЛУ с явно заданным неотрицательным начальным значением  $X$  для итераций.

Применение ограничений позволяет избежать получения отрицательных корней, но ведет к появлению больших погрешностей решения, чем в случае решений без ограничений:

```
C = [0.0372 0.2869; 0.6861 0.7071; 0.6233 0.6245; 0.6344... 0.6170];
```

```
d = [0.8587 0.1781 0.0747 0.8405];
```

```
% solving two variant
```

```
[C\d lsqnonneg(C,d)] =
```

```
-2.5627 0
```

```
3.1108 0.6929
```

```
% check accuracy
```

```
[norm(C*(C\d)-d) norm(C*lsqnonneg(C,d)-d)] =
```

```
0.6674 0.9118
```

### 3.2.3 Решение СЛУ с разреженными матрицами

Большинство методов решения СЛУ с разреженными матрицами относится к итерационным, т. е. к тем, решение которых получается в ходе ряда шагов - итераций, постепенно ведущих к получению результата с заданной погрешностью или с максимальным правдоподобием. Описанные ниже функции MATLAB могут применяться и при решении обычных СЛУ без разреженных матриц и реализуют различные методы (точное решение, метод наименьших квадратов и сопряженных градиентов):

1)  $\text{lsqr}(A, B)$  - возвращает точное решение  $X$  СЛУ  $A \cdot X = B$ , если матрица последовательная, в противном случае — возвращает решение, полученное итерационным методом наименьших квадратов. Матрица коэффициентов  $A$  должна быть прямоугольной размера  $m \times n$ , а вектор-столбец правых частей

уравнений  $B$  должен иметь размер  $m$ . Условие  $m \geq n$  может быть и необязательным. Функция начинает итерации от начальной оценки, по умолчанию представляющей собой вектор размером  $m$ , состоящий из нулей. Итерации производятся или до сходимости к решению, или до появления ошибки, или до достижения максимального числа итераций;

2) `lsqr(A,B,tol)` - возвращает решение с заданной погрешностью (порогом отбора) `tol`;

3) `lsqr(A,b,tol,maxit)` - возвращает решение при заданном максимальном числе итераций `maxit`.

Решение СЛУ с разреженной матрицей возможно также двунаправленным методом сопряженных градиентов. Он реализован указанной ниже функцией в нескольких представлениях:

1) `bicg(A, B)` — возвращает решение  $X$  СЛУ  $A \cdot X = B$ . Матрица коэффициентов  $A$  должна быть квадратной размера  $n \times n$ , а вектор-столбец правых частей уравнений  $B$  должен иметь длину  $n$ . Функция `bicg` начинает итерации от начальной оценки, по умолчанию представляющей собой вектор размером  $n$ , состоящий из нулей. Итерации производятся или до сходимости к решению, или до появления ошибки, или до достижения максимального числа итераций. Благодаря использованию двунаправленного метода сопряженных градиентов `bicg` сходится за меньшее число итераций, чем `lsqr`, но требует квадратную матрицу  $A$ , отбрасывая информацию, содержащуюся в дополнительных уравнениях, в то время как `lsqr` работает и с прямоугольной матрицей;

2) `bicg(A,B,tol)` - выполняет и возвращает решение с погрешностью (порогом отбора) `tol`;

3) `bicg(A,b,tol,maxit)` - выполняет и возвращает решение при заданном максимальном числе итераций `maxit`;

4) `bicg(A,b,tol,maxit,M)` и `bicg(A,b,tol,maxit,M1,M2)` — при решении используются матрица предусловий  $M$  или  $M = M1 \cdot M2$ , так что производится решение системы  $\text{inv}(M) \cdot A \cdot x = \text{inv}(M) \cdot b$  относительно  $x$ . Если  $M1$  или  $M2$  — пустые матрицы, то они рассматриваются как единичные матрицы, что эквивалентно отсутствию входных условий вообще;

5) `bicg(A,B,tol,maxit,M1,M2,X0)` - точно задается начальное приближение  $X0$ . Если  $X0$  - пустая матрица, то по умолчанию используется вектор, состоящий из нулей, например:

```
n = 100;
on = ones(n,1);
A = spdiags([-2*on 4*on -on],[-1:1,n,n]);
b = sum(A,2);
tol = 1e-8;
maxit = 15;
M1 = spdiags([on/(-2) on],[-1:0,n,n]);
M2 = spdiags([4*on -on],0:1,n,n);
x = bicg(A,b,tol,maxit,M1,M2,[]);
```

Итерационный метод сопряженных градиентов реализован функцией `pcg`. Функция `pcg(A,B)` - возвращает решение  $X$  СЛУ  $A*X=B$ . Матрица  $A$  должна быть квадратной, симметрической и положительно определенной. Функция `pcg` начинает итерации от начальной оценки, представляющей собой вектор размером  $n$ , состоящий из нулей. Итерации производятся либо до сходимости решения, либо до появления ошибки, либо до достижения максимального числа итераций. Функция `pcg(...)` имеет и ряд других форм записи, описанных в руководстве по Matlab.

Квадратичный метод сопряженных градиентов реализуется в системе MATLAB с помощью функции `cg`. Функция `cg(A,B)` - возвращает решение  $X$  СЛУ  $A*X=B$ .  $A$  - квадратная матрица. Функция `cg` начинает итерации от начальной оценки, по умолчанию представляющей собой вектор размера  $n$ , состоящий из нулей. Итерации производятся либо до сходимости метода, либо до появления ошибки, либо до достижения максимального числа итераций. Функция `cg(...)` также имеет и ряд других форм записи, подобных выше описанным и приведенным в руководстве по Matlab.

Итерационный метод минимизации обобщенной невязки также реализован в системе MATLAB. Для этого используется функция `gmres`. Функция `gmres(A, B, restart)` — возвращает решение  $X$  СЛУ  $A*X=B$ .  $A$  - квадратная матрица. Функция `gmres` начинает итерации от начальной оценки, представляющей собой вектор размера  $n$ , состоящий из нулей. Максимальное число итераций - минимум из  $n/restart$  и 10.

Метод решения СЛУ с квазиминимизацией невязки реализует функция `qmr`. Функция `qmr(A, B)` - возвращает решение  $X$  СЛУ  $A*X=B$ . Матрица  $A$  должна быть квадратной.

### 3.3 Вычисление нулей функции одной переменной

Ряд функций системы MATLAB предназначен для работы с функциями. По аналогии с дескрипторами графических объектов могут использоваться объекты класса дескрипторов функций, задаваемых с помощью символа `@`, например: `fe=@exp`.

Численные значения таких функций, заданных дескрипторами, вычисляются с помощью функции `feval`, например:

```
[V,D] = eig(A);  
[V,D] = feval(@eig,A);
```

Довольно часто возникает задача решения нелинейного уравнения вида:

$$f(x) = 0 \text{ или } f_1(x) = f_2(x).$$

Последнее можно свести к виду:

$$f(x) = f_1(x) - f_2(x) = 0.$$

Таким образом, задача сводится к нахождению значений аргумента  $x$  функции  $f(x)$  одной переменной, при котором значение функции равно нулю:

1) `fzero(@fun,x)` - возвращает уточненное значение  $x$ , при котором достигается нуль функции `fun`, представленной в символьном виде, при начальном значении аргумента  $x$ . Возвращенное значение близко к точке, где функция меняет знак, или равно NaN, если такая точка не найдена;

2) `fzero(@fun,[x1 x2])` - возвращает значение  $x$ , при котором  $fun(x)=0$  с заданием интервала поиска с помощью вектора  $x=[x1\ x2]$ . Если это не так, выдается сообщение об ошибке. Вызов функции `fzero` с интервалом гарантирует, что `fzero` возвратит значение, близкое к точке, где `fun` изменяет знак;

3) `fzero(@fun,x,tol)` - возвращает результат с заданной погрешностью `tol`;

4) `fzero(@fun,x,tol,trace)` - выдает на экран информацию о каждой итерации;

5) `fzero(@fun,x,tol,trace,P1,P2,...)` - предусматривает дополнительные аргументы, передаваемые в функцию `fun(x,P1,P2,...)`. При задании пустой матрицы для `tol` или `trace` используются значения по умолчанию, например:

```
% first modul
function y = f(x)
y = x.^3-2*x-5;

% second modul
z = fzero(@f,2)
z =
    2.0946
```

Решение подобных задач возможно при помощи функции «`fsolve`» из пакета Optimization Toolbox, которая решает систему нелинейных уравнений вида  $f(x)=0$  методом наименьших квадратов, и ищет не только точки пересечения, но и точки касания. Функция `fsolve` имеет почти те же параметры (дополнительный параметр - задание якобиана) и почти ту же запись, что и функция `lsqnonneg`.

### 3.4 Минимизация функции одной переменной

Важная задача в математическом моделировании сложных технических систем - поиск минимума функции  $f(x)$  в некотором интервале изменения  $x$  - от  $x1$  до  $x2$ . Если нужно найти максимум такой функции, то достаточно поставить знак «минус» перед функцией. Для решения этой задачи используется функция:

```
[X,fval,exitflag,output] = fminbnd(@fun,x1,x2,options,pl,p2,...)
```

Функция `fminbnd` имеет следующие представления:

1) `fminbnd(@fun,x1,x2)` - возвращает значение  $x$ , которое является локальным минимумом функции  $\text{fun}(x)$  на интервале  $x_1 < x < x_2$ ;

2) `fminbnd(@fun,x1,x2,options)` - сходна с описанной выше формой функции, но использует параметры из вектора `options`, предварительно установленные при помощи команды `optimset`;

3) `fminbnd(@fun,x1,x2,options,P1,P2...)` - сходна с описанной выше, но передает в целевую функцию дополнительные аргументы:  $P_1, P_2, \dots$ . Если требуется использовать параметры вычислений по умолчанию, то вместо `options` перед  $P_1, P_2$  необходимо ввести `[]` (пустой массив);

4) `[x,fval] = fminbnd(...)` - дополнительно возвращает значение целевой функции `fval` в точке минимума;

5) `[x,fval,exitflag] = fminbnd(...)` - дополнительно возвращает параметр `exitflag`, равный 1, если функция сошлась с использованием `options`, и 0, если достигнуто максимальное число итераций `options.maxiter`.

Пример использования функции:

```
options = optimset('Display','iter','TolFun',1e-8, 'tolX',1.e-10);  
[x]=fminbnd(@cos,3,4,options)  
x =  
3.1416
```

### 3.5 Минимизация функции нескольких переменных

Значительно сложнее задача минимизации функций нескольких переменных  $f(x_1, \dots)$ . При этом значения переменных представляются вектором  $x$ , причем начальные значения задаются вектором  $x_0$ . В качестве простейших примеров решения такого класса задач в MATLAB обычно используют функции, реализующие разновидности симплекс-метода Нелдера-Мида.

Этот метод сводится к построению симплекса в  $n$ -мерном пространстве, заданного  $n+1$  вершиной. В двумерном пространстве симплекс является треугольником, а в трехмерном — пирамидой. На каждом шаге итераций выбирается новая точка решения внутри или вблизи симплекса. Она сравнивается с одной из вершин симплекса. Ближайшая к этой точке вершина симплекса обычно заменяется этой точкой. Таким образом, симплекс перестраивается и обычно позволяет найти новое, более точное положение точки решения. Решение повторяется, пока размеры симплекса по всем переменным не станут меньше заданной погрешности решения.

Реализующая симплекс-методы Нелдера-Мида функция записывается в виде:

1) `fminsearch(@fun,x0)` - возвращает вектор  $x$ , который является локальным минимумом функции  $\text{fun}(x)$  вблизи  $x_0$ .  $x_0$  может быть скаляром, вектором (отрезком) при минимизации функции одной переменной или матрицей (для функции нескольких переменных);

2) `fminsearch(@fun,x0,options)` - аналогична описанной выше функции, но использует вектор параметров `options` точно так же, как функция `fminbnd`;



3) `fminsearch(@fun,x0,options,P1,P2,...)` - сходна с описанной выше функцией, но передает в минимизируемую функцию нескольких переменных `fun(x,P1,P2,...)` ее дополнительные аргументы `P1, P2,...`. Если требуется использовать параметры вычислений по умолчанию, то вместо `options` перед `P1, P2` необходимо ввести `[]`;

4) `[x,fval] = fminsearch(...)` - дополнительно возвращает значение целевой функции `fval` в точке минимума;

5) `[x,fval,exitflag] = fminsearch(...)` - дополнительно возвращает параметр `exitflag` положительным, если процесс итераций сходится с использованием `options` и отрицательным, если итерационный процесс не сходится к полученному решению `x`, а также 0, если превышено максимальное число итераций.

6) `[x, fval, exitflag, output]` - `fminsearch(...)` возвращает структуру `output`.

Структура `output` включает:

1) `output.algorithm` - использованный алгоритм;

2) `output.funcCount` - число оценок целевой функции;

3) `output.Iterations` - число проведенных итераций.

Для минимизации функций нескольких переменных можно использовать также функцию MATLAB `fminunc` и функцию `lsqnonlin` из пакета `Optimization Toolbox`. Первая из них позволяет использовать предварительно заданные командой `optimset` порог сходимости для значения целевой функции, вектор градиентов `gradobj`, матрицу Гесса, функцию умножения матрицы Гесса или график разреженности матрицы Гесса целевой функции. `lsqnonlin` реализует метод наименьших квадратов и, как правило, дает наименьшее число итераций при минимизации.

Классическим примером применения функции `fminsearch` является поиск минимума тестовой функции Розенброка, точка минимума которой находится в «овраге» с «плоским дном»:

```
function f = banana(x)  
f = 100*(x(2)-x(1)^2)^2+(1-x(1))^2;  
% main function  
[x,fval] = fminsearch(@banana,[-1.2, 1]);  
% results  
x =  
    1.0000    1.0000  
fval =  
    8.1777e-010
```

### 3.6 Численное интегрирование

Численное интегрирование традиционно является одной из важнейших сфер применения математического аппарата. Численное интегрирование заключается в приближенном вычислении определенного интеграла.

### 3.6.1 Метод трапеций

Приведенные ниже функции выполняют численное интегрирование методом трапеций и методом трапеций с накоплением:

1) `trapz(Y)` - возвращает определенный интеграл, используя интегрирование методом трапеций с единичным шагом между отсчетами. Если  $Y$  - вектор, то `trapz(Y)` возвращает интеграл элементов вектора  $Y$ , если  $Y$  - матрица, то `trapz(Y)` возвращает вектор-строку, содержащую интегралы каждого столбца этой матрицы.

2) `trapz(X,Y)` - возвращает интеграл от функции  $Y$  по переменной  $X$ , используя метод трапеций (пределы интегрирования в этом случае задаются начальным и конечным элементами вектора  $X$ ):

```
X = 0:pi/100:pi; Y=sin(X); Z = trapz(X,Y)
Z =
    1.9998
```

### 3.6.2 Численное интегрирование методом квадратур

Приведенные ниже функции осуществляют интегрирование и двойное интегрирование, используя квадратурную формулу Симпсона или метод Гаусса-Лобатто. Квадратура - численный метод нахождения площади под графиком функции  $f(x)$ .

Функции `quad` и `quadl` используют два различных алгоритма квадратуры для вычисления определенного интеграла. Функция `quad` выполняет интегрирование по методу низкого порядка, используя рекурсивное правило Симпсона. Но она может быть более эффективной при негладких подынтегральных функциях или при низкой требуемой точности вычислений. Функция `quadl` (квадратура Лобатто) использует адаптивное правило квадратуры Гаусса— Лобатто очень высокого порядка:

1) `quad(@fun,a,b)` - возвращает численное значение определенного интеграла от заданной функции `@fun` на отрезке  $[a, b]$ .

2) `quad(@fun,a,b,tol)` - возвращает численное значение определенного интеграла с заданной относительной погрешностью `tol`. По умолчанию `tol=1.e-6`.

3) `quad(@fun,a,b,tol,trace)` - возвращает численное значение определенного интеграла, а при значении `trace`, не равном нулю, строит график, показывающий ход вычисления интеграла.

4) `quad(@fun,a,b,tol,trace,P1,P2,...)` - возвращает численное значение определенного интеграла. Для подынтегральной функции `fun`, использует дополнительные аргументы  $P1, P2, \dots$ , которые напрямую передаются в подынтегральную функцию:  $G=fun(X,P1,P2,\dots)$ , например:

```
% first variant
Q = quad('1./(x.^3-2*x-5)',0,2);
% second variant - An inline object
```

```

F = inline('1./(x.^3-2*x-5)');
Q = quad(F,0,2);
% third variant
Q = quad(@myfun,0,2);
% where myfun.m is an M-file.
function y = myfun(x)
y = 1./(x.^3-2*x-5);

```

5) `dblquad(@fun,inmin,inmax,outmin,outmax)` - вычисляет и возвращает значение двойного интеграла для подынтегральной функции `fun`, по умолчанию используя квадратурную функцию `quad`.

6) `dblquad(@fun,inmin,inmax,outmin,outmax,tol,trace)` - передает в функцию `dblquad` параметры `tol` и `trace`, например:

```

% first variant
Q = dblquad(inline('y*sin(x)+x*cos(y)'), pi, 2*pi, 0, pi)
% second variant
Q = dblquad(@integrnd, pi, 2*pi, 0, pi)
% where
function z = integrnd(x, y)
z = y*sin(x)+x*cos(y);

```

Первый аргумент в перечисленных функциях `@fun` - строка, описывающая подынтегральную функцию. Это может быть либо дескриптор функции, либо объект `inline` (в последнем случае символ «@» в ее записи отсутствует). Функция должна быть функцией двух переменных:

**fout=fun(inner,outer)**

Аргументами функции являются вектор `inner` и скаляр `outer`. Функция возвращает вектор `fout`, который является функцией, вычисленной в `outer` и каждом значении `inner`.

## 4 Интерполяция и аппроксимация данных

Под аппроксимацией подразумевается описание некоторой, зависимости или совокупности представляющих ее данных с помощью другой, обычно более простой или более единообразной зависимости. Часто данные находятся в виде отдельных узловых точек, координаты которых задаются таблицей данных. Результат аппроксимации может не проходить через узловые точки. Напротив, задача интерполяции - найти данные в окрестности узловых точек. Для этого используются подходящие функции, значения которых в узловых точках совпадают с координатами этих точек. Например, при линейной интерполяции зависимости  $y(x)$  узловые точки соединяются друг с другом отрезками прямых и считается, что искомые промежуточные точки расположены на этих отрезках.

Для повышения точности интерполяции применяют параболы (квадратичная интерполяция) или полиномы более высокой степени (полиномиальная интерполяция). Для обработки данных MATLAB использует различные функции интерполяции и аппроксимации данных.

### 4.1 Полиномиальная регрессия

Одна из наиболее известных аппроксимаций - полиномиальная. В системе MATLAB определены функции аппроксимации данных полиномами по методу наименьших квадратов - полиномиальной регрессии:

1) `polyfit(x,y,n)` - возвращает вектор коэффициентов полинома  $p(x)$  степени  $n$ , который с наименьшей среднеквадратичной погрешностью аппроксимирует функцию  $y(x)$ . График полинома точно проходит через узловые точки с координатами  $(x,y)$ , хранящиеся в векторах  $x$  и  $y$ . В противном случае точного совпадения графика с узловыми точками не наблюдается;

2) `[p,S] = polyfit(x,y,n)` - возвращает коэффициенты полинома  $p$  и структуру  $S$  для использования вместе с функцией `polyval` с целью оценивания или предсказания погрешности, например:

```
x = (0: 0.1: 2.5)';
y = erf(x);
p = polyfit(x,y,6)
p =
0.0084 -0.0983 0.4217 -0.7435 0.1471 1.1064 0.0004
f = polyval(p,x);
y-f
ans =
1.0e-003 *
-0.4412
0.6084...
```

## 4.2 Интерполяция периодических функций рядом Фурье

Под интерполяцией обычно подразумевают вычисление значений функции  $f(x)$  в промежутках между узловыми точками. Линейная, квадратичная и полиномиальная интерполяция реализуются при полиномиальной аппроксимации. А вот для периодических (и особенно для гладких периодических) функций хорошие результаты может дать их интерполяция тригонометрическим рядом Фурье. Для этого используется следующая функция:

`interpft(x,n)` - возвращает вектор  $y$ , содержащий значения периодической функции, определенные в  $n$  равномерно расположенных точках.

## 4.3 Интерполяция на неравномерной сетке

Для интерполяции на неравномерной сетке используется функция `griddata` в следующих представлениях:

1) `ZI = griddata(x,y,z,XI,YI)` - преобразует поверхность вида  $z = f(x, y)$ , которая определяется векторами  $(x,y,z)$  с неравномерно распределенными элементами. Функция `griddata` аппроксимирует эту поверхность в точках, определенных векторами  $(XI,YI)$  в виде значений  $ZI$ . Поверхность всегда проходит через заданные точки.  $XI$  и  $YI$  обычно формируют однородную сетку, созданную с помощью функции `meshgrid`.  $XI$  может быть вектором-строкой, в этом случае он определяет матрицу с постоянными столбцами. Точно так же  $YI$  может быть вектором-столбцом, тогда он определяет матрицу с постоянными строками;

2) `[XI,YI,ZI] = griddata(x,y,z,xi,yi)` - возвращает аппроксимирующую матрицу  $ZI$ , как описано выше, а также возвращает матрицы  $XI$  и  $YI$ , сформированные из вектора-столбца  $xi$  и вектора-строки  $yi$ ;

3) `[...]=griddata(...method)` - использует различные методы интерполяции:

а) 'nearest' - ступенчатая интерполяция;

б) 'linear' - линейная интерполяция (принята по умолчанию);

в) 'cubic' - кубическая интерполяция.

Метод определяет тип аппроксимирующей поверхности. Метод 'cubic' формирует гладкие поверхности, в то время как 'linear' и 'nearest' имеют разрывы первых и нулевых производных соответственно. Все методы основаны на триангуляции Делоне.

Пример `m`-файла и результат работы программы (рисунок 4.1):

```
rand('seed',0);
x = rand(100,1)*4-2; y = rand(100,1)*4-2;
z = x.*exp(-x.^2-y.^2);
ti = -2:.25:2;
[XI,YI] = meshgrid(ti,ti);
ZI = griddata(x,y,z,XI,YI);
mesh(XI,YI,ZI), hold
```

`plot3(x,y,z,'o'), hold off`

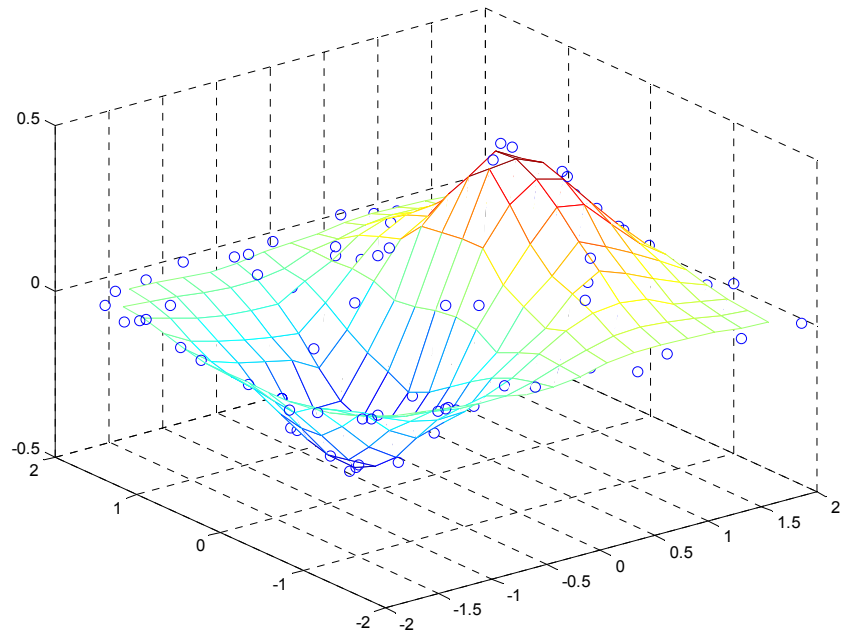


Рисунок 4.1 – Пример интерполяции

#### 4.4 Двумерная табличная интерполяция

Двумерная интерполяция существенно сложнее, чем одномерная, рассмотренная выше, хотя смысл ее тот же — найти промежуточные точки некоторой зависимости  $z(x, y)$  вблизи расположенных в пространстве узловых точек. Для двумерной табличной интерполяции используется функция `interp2`:

1)  $ZI = \text{interp2}(X, Y, Z, XI, YI)$  — возвращает матрицу  $ZI$ , содержащую значения функции в точках, заданных аргументами  $XI$  и  $YI$ , полученные путем интерполяции двумерной зависимости, заданной матрицами  $X$ ,  $Y$  и  $Z$ . При этом  $X$  и  $Y$  должны быть монотонными и иметь тот же формат, как если бы они были получены с помощью функции `meshgrid` (строки матрицы  $X$  являются идентичными; то же можно сказать о столбцах массива  $Y$ ). Матрицы  $X$  и  $Y$  определяют точки, в которых задано значение  $Z$ . Параметры  $XI$  и  $YI$  могут быть матрицами, в этом случае `interp2` возвращает значения  $Z$ , соответствующие точкам  $(XI(i,j), YI(i,j))$ . В качестве альтернативы можно передать в качестве параметров вектор-строку  $xi$  и вектор-столбец  $yi$ . В этом случае `interp2` представляет эти векторы так, как если бы использовалась команда `meshgrid(xi, yi)`;

2)  $ZI = \text{interp2}(Z, XI, YI)$  - подразумевает, что  $X=1:n$  и  $Y=1:m$ , где  $[m,n]=\text{size}(Z)$ ;

3)  $ZI = \text{interp2}(Z, ntimes)$  - осуществляет интерполяцию рекурсивным методом с числом шагов  $ntimes$ ;

4)  $ZI = \text{interp2}(X, Y, Z, XI, YI, method)$  — позволяет с помощью опции `method` задать метод интерполяции:

- а) 'nearest' — интерполяция по соседним точкам;
- б) 'linear' — линейная интерполяция;
- в) 'cubic' — кубическая интерполяция (полиномами Эрмита);
- г) 'spline' — интерполяция сплайнами.

Все методы интерполяции требуют, чтобы X и Y изменялись монотонно и имели такой же формат, как если бы они были получены с помощью функции meshgrid. Когда X и Y — векторы равномерно распределенных точек, для более быстрой интерполяции лучше использовать методы '\*linear', '\*cubic', или '\*nearest'.

Пример совместного использования графических функций и функции interp2 (рисунок 4.2):

```
[X,Y] = meshgrid(-3:.25:3);  
Z = peaks(X,Y);  
[XI,YI] = meshgrid(-3:.125:3);  
ZI = interp2(X,Y,Z,XI,YI);  
mesh(X,Y,Z), hold, mesh(XI,YI,ZI+15)  
hold off  
axis([-3 3 -3 3 -5 20])
```

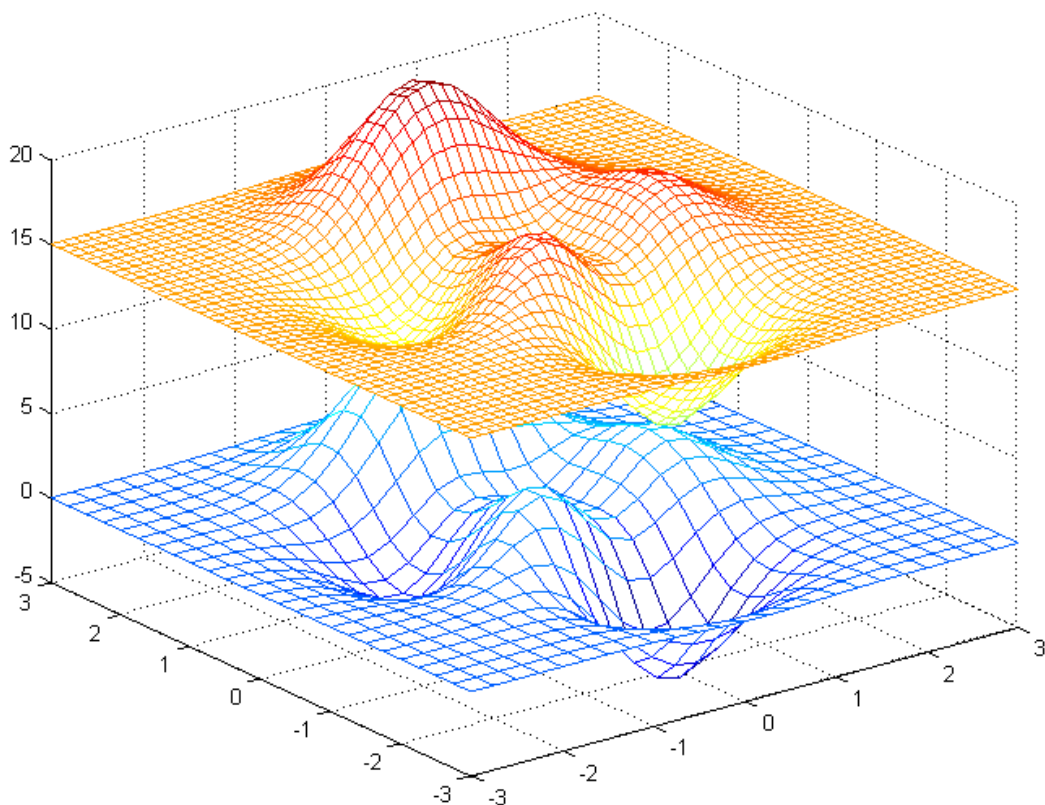


Рисунок 4.2 - Графическое представление результата

## 4.5 Трехмерная табличная интерполяция

Для трехмерной табличной интерполяции используется функция `interp3`:

1)  $VI = \text{interp3}(X, Y, Z, V, XI, YI, ZI)$  - интерполирует, чтобы найти  $VI$ , значение основной трехмерной функции  $V$  в точках матриц  $XI$ ,  $YI$  и  $ZI$ . Матрицы  $X$ ,  $Y$  и  $Z$  определяют точки, в которых задано значение  $V$ ,  $XI$ ,  $YI$  и  $ZI$  могут быть матрицами, в этом случае `interp3` возвращает значения  $Z$ , соответствующие точкам  $(XI(i, j), YI(i, j), ZI(i, j))$ . В качестве альтернативы можно передать векторы  $x_i$ ,  $y_i$  и  $z_i$ . Векторы аргументы, имеющие неодинаковый размер, представляются, как если бы использовалась команда `meshgrid`;

2)  $VI = \text{interp3}(V, XI, YI, ZI)$  - подразумевает  $X=1:N, Y=1:M, Z=1:P$ , при этом  $[M, N, P] = \text{size}(V)$ ;

3)  $VI = \text{interp3}(V, \text{ntimes})$  - осуществляет интерполяцию рекурсивным методом с числом шагов  $\text{ntimes}$ ;

4)  $VI = \text{interp3}(\dots, \text{method})$  - позволяет задать метод интерполяции:

а) 'nearest' - ступенчатая интерполяция;

б) 'linear' — линейная интерполяция;

в) 'cubic' — кубическая интерполяция (полиномами Эрмита);

г) 'spline' — интерполяция сплайнами.

Все методы интерполяции требуют, чтобы  $X$ ,  $Y$  и  $Z$  изменялись монотонно и имели такой же формат, как если бы они были получены с помощью функции `meshgrid`. Когда  $X$  и  $Y$  и  $Z$  - векторы равномерно распределенных в пространстве узловых точек, для более быстрой интерполяции лучше использовать методы '\*li'near', '\*cubic' или '\*nearest'.

#### 4.6 N-мерная табличная интерполяция

MATLAB позволяет выполнить даже  $n$ -мерную табличную интерполяцию. Для этого используется функция `interp $n$` :

1)  $VI = \text{interp $n$ }(X1, X2, X3, \dots, V, Y1, Y2, Y3, \dots)$  - интерполирует, чтобы найти  $VI$ , значение основной  $n$ -мерной функции  $V$  в точках массивов  $Y1$ ,  $Y2$ ,  $Y3, \dots$ . Функции `interp $n$`  должно передаваться  $2N+1$  аргументов, где  $N$  - размерность интерполируемой функции. Массивы  $X1$ ,  $X2$ ,  $X3, \dots$  определяют точки, в которых задано значение  $V$ . Параметры  $Y1$ ,  $Y2$ ,  $Y3, \dots$  могут быть матрицами, в этом случае `interp $n$`  возвращает значения  $VI$ , соответствующие точкам  $(Y1(i, j), Y2(i, j), Y3(i, j), \dots)$ ;

2)  $VI = \text{interp $n$ }(V, Y1, Y2, Y3, \dots)$  - подразумевает  $X1=1:\text{size}(V, 1)$ ,  $X2=1:\text{size}(V, 2)$ ,  $X3=1:\text{size}(V, 3)$  и т. д.;

3)  $VI = \text{interp $n$ }(V, \text{ntimes})$  - осуществляет интерполяцию рекурсивным методом с числом шагов  $\text{ntimes}$ ;

4)  $VI = \text{interp $n$ }(... \text{method})$  - позволяет указать метод интерполяции:

а) 'nearest' - ступенчатая интерполяция;

б) 'linear' - линейная интерполяция;

в) 'cubic' — кубическая интерполяция.



## 4.7 Интерполяция кубическим сплайном

Сплайн-интерполяция используется для представления данных отрезками полиномов невысокой степени - чаще всего третьей. При этом кубическая интерполяция обеспечивает непрерывность первой и второй производных результата интерполяции в узловых точках. Свойства кубической сплайн-интерполяции:

- график кусочно-полиномиальной аппроксимирующей функции проходит точно через узловые точки;
- в узловых точках нет разрывов и резких перегибов функции;
- связь между числом узловых точек и степенью полинома отсутствует;
- возможность аппроксимации функций с множеством пиков и впадин.

Сплайн-интерполяция реализуется различными представлениями функцией:

1)  $y_i = \text{spline}(x,y,x_i)$  - использует векторы  $x$  и  $y$ , содержащие аргументы функции и ее значения, и вектор  $x_i$ , задающий новые точки; для нахождения элементов вектора  $y_i$  используется кубическая сплайн-интерполяция;

2)  $pp = \text{spline}(x,y)$  - возвращает  $pp$ -форму сплайна, используемую в функции  $ppval$  и других сплайн-функциях.

Ниже приведен пример использования данной функции в  $m$ -файле и графическое представление результатов (рисунок 4.3):

```
x = 0:10;  
y = sin(x);  
xx = 0:.25:10;  
yy = spline(x,y,xx);  
plot(x,y,'o',xx,yy)
```

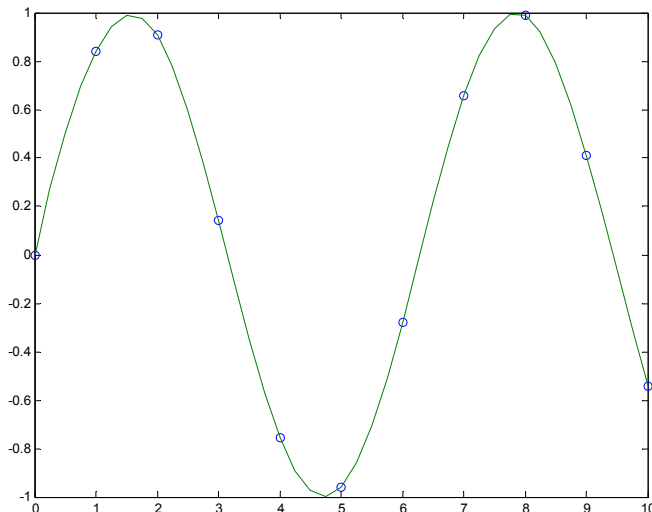


Рисунок 4.3 - Пример использования функции spline

Пример совместного использования функций spline и ppval в  $m$ -файле и графическое представление результатов (рисунок 4.4):

```

x = pi*[0:.5:2];
y = [0 1 0 -1 0 1 0; 1 0 1 0 -1 0 1];
pp = spline(x,y);
yy = ppval(pp, linspace(0,2*pi,101));
plot(yy(1,:),yy(2,:),'-b',y(1,2:5),y(2,2:5),'or'), axis equal

```

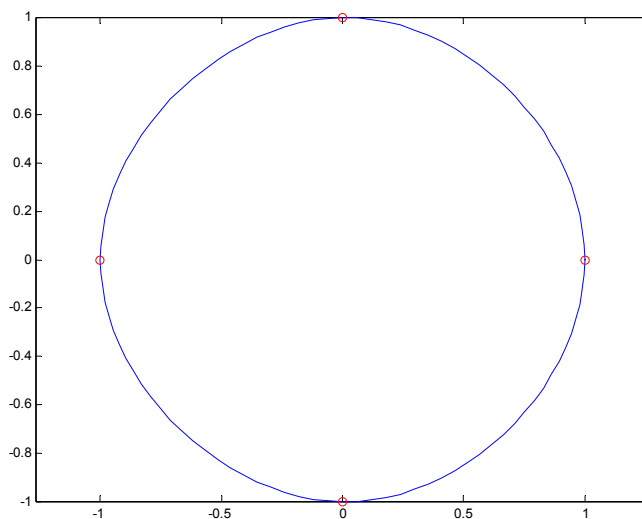


Рисунок 4.4 – Пример использования функций spline и ppval

## 4.8 Обработка данных в графическом окне

### 4.8.1 Средства обработки данных в графическом окне

Решение большинства задач интерполяции и аппроксимации функций и табличных данных должно сопровождаться их визуализацией. Она заключается в построении узловых точек функции (или табличных данных) и в построении функции аппроксимации или интерполяции. Для простых видов аппроксимации, например полиномиальной, желательно нанесение на график формулы, полученной для аппроксимации.

В MATLAB 6 совмещение функций аппроксимации с графической визуализацией предусмотрена аппроксимация рядом методов точек функции, график которой построен. Это выполняется в окне редактора графики Property Editor. Для этого в позиции Tools графического окна имеются две команды:

- 1) Basic Fitting - основные виды аппроксимации (регрессии);
- 2) Data Statistics - статистические параметры данных.

Команда Basic Fitting открывает окно, дающее доступ к ряду видов аппроксимации и регрессии: сплайновой, эрмитовой и полиномиальной со степенями от 1 (линейная аппроксимация) до 10. В том числе со степенью 2 (квадратичная аппроксимация) и 3 (кубическая аппроксимация). Команда Data Statistics открывает окно с результатами простейшей статистической обработки данных.

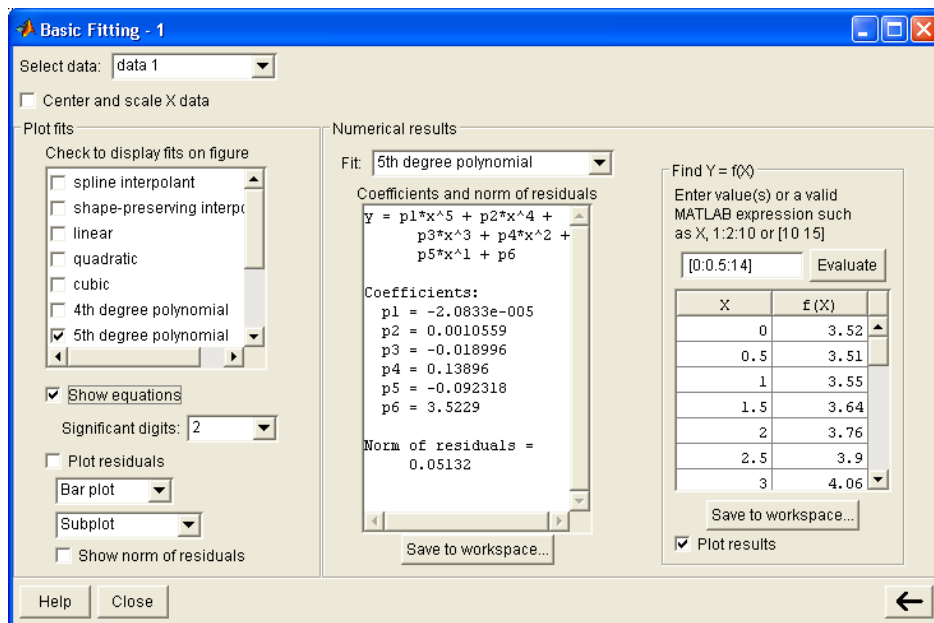
## 4.8.2 Полиномиальная регрессия для табличных данных

Рассмотрим самый характерный пример обработки данных, представляющих некоторую зависимость вида  $y(x)$ . Пусть она задана в табличной форме, причем колонки таблицы соответствуют элементам векторов  $X$  и  $Y$  одинакового размера в следующем примере:

- »  $X=[2,4,6,8,10,12,14];$
- »  $Y=[3.76, 4.4, 5.1, 5.5, 6, 6.63, 6.7];$
- »  $\text{plot}(X,Y,'o');$

Исполнив команду `Tools > Basic Fitting`, можно получить окно регрессии. Оно показано на рисунке 4.5 слева прямо под записью исходных команд в командной строке. В этом окне птичкой отмечен вид полиномиальной регрессии. При проведении полиномиальной аппроксимации надо помнить, что максимальная степень полинома на 1 меньше числа точек, т. е. числа элементов в векторах  $X$  и  $Y$ .

Установив птичку у параметра `Show equations` (Показать уравнения), можно получить в графическом окне запись уравнений регрессии (аппроксимации). Можно сместить выводимую по умолчанию легенду в место, где она не закрывала бы другие детали графика.



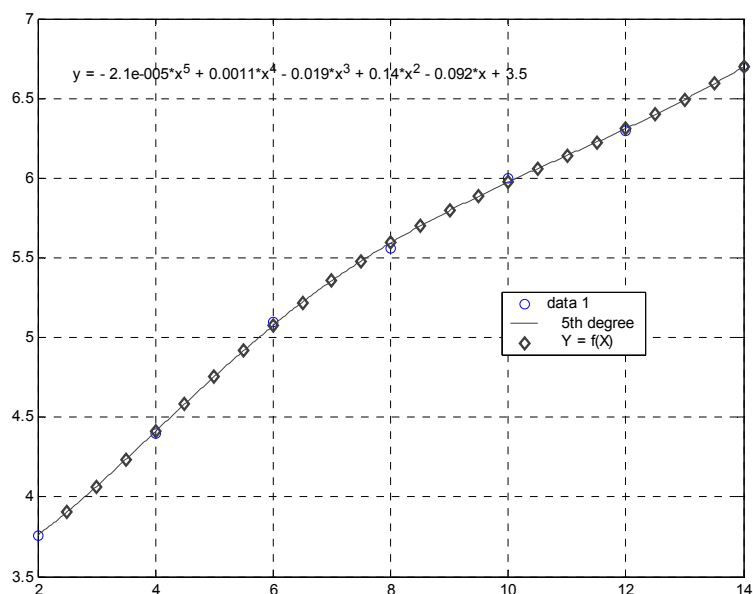


Рисунок 4.5 - Пример выполнения полиномиальной регрессии

Исполнив команду Tools > Data Statistics, можно получить окно с рядом статистических параметров данных, представленных векторами X и Y. Отметив птичкой тот или иной параметр в этом окне, можно наблюдать соответствующие построения на графике, например вертикалей с минимальным, средним и максимальных значений y и горизонталей с минимальным, средним и максимальным значением x. Средства обработки данных из графического окна позволяют строить столбцовый или линейчатый график погрешностей в узловых точках и наносить на эти графики норму погрешности. Норма дает статистическую оценку среднеквадратической погрешности. Чем она меньше, тем точнее аппроксимация. Для вывода графика погрешности надо установить птичку у параметра Plot residuals (График погрешностей) и в меню ниже этого параметра выбрать тип графика.

## 5 Работа с файлами

### 5.1 Открытие и закрытие файлов

Файл является некоторой совокупностью данных, объединенных одним именем. Тип файла, как правило, определяется его расширением. Перед использованием любого файла он должен быть открыт, а по окончании использования - закрыт. Много файлов может быть открыто и доступно для чтения одновременно.

Команда `open имя`, где имя должно содержать массив символов или символьную переменную, открывает файлы в зависимости от анализа параметра имя и расширения:

1) `open name.mat` - открывает файл с сохраненными переменными рабочей области;

2) `open name.fig` - открывает файл в редакторе дескрипторной графики Property Editor;

3) `open name.m` - открывает m-файл в редакторе-отладчике;

4) `open name.mdl` - открывает модель в Simulink;

5) `open name.p` - открывает, если он есть, m-файл с тем же именем;

6) `open name.html` - открывает HTML документ в браузере помощи.

Функция `[FILENAME, PATHNAME] = uigetfile(FILTERSPEC, Title)` открывает диалог с именем Title и фильтром FILTERSPEC (например, массивом ячеек, содержащим расширения файлов) и возвращает файл, выбранный пользователем, и путь к нему. Возвращает значение переменной FILENAME, равное нулю, если файл не существует или если пользователь нажал на Cancel.

Функция вида:

`[FILENAME, PATHNAME] = uigetfile (FILTERSPEC, Title, X, Y)`

размещает окно диалога в точке X, Y (координаты в пикселях), например:

`[fname,pname] = uigetfile('*.*','Select the M-file');`

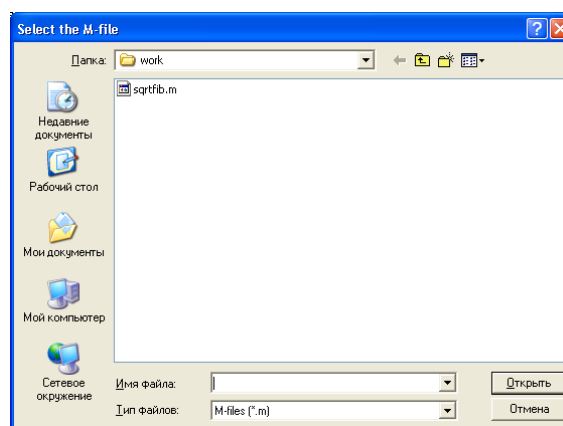


Рисунок 5.1 – Открытие файла

Функция вида:

```
[FILENAME, PATHNAME] = uinputfile(FILTERSPEC, TITLE)
```

сохраняет файл в диалоге, управляемом пользователем. Параметры аналогичны таковым в функции `uigetfile`, например:

```
[newfile,newpath] = uinputfile('animinit.m','Save file name');
```

Команда `uiopen` открывает диалог, и если пользователь выбрал файл с известным расширением, вызывает его, используя `open`, или если имя файла имеет неизвестное расширение, то вызывается `uigetfile`. Входными аргументами `uiopen` могут быть `matlab`, `load`, `figure`, `simulink`, `editor`. Без входных аргументов или с входным аргументом `matlab` в окне диалога предлагается выбрать `*.m`, `*.fig`, `*.mat`, `*.mdl` (если Simulink установлен), `*.cdr` (если state flow установлен), `*.rtw`, `*.tmf`, `*.tlc`, `*.c`, `*.h`, `*.ads`, `*.adb` (если установлен Real-Time Workshop). С аргументом `load` - `*.mat`. С аргументом `figure` предлагаются файлы с расширениями: `*.fig`, `*.mdl`, `*.m`, `*.mdt`, `*.cdr`, `*.rtw`, `*.tmf`, `*.tlc`, `*.c`, `*.h`, `*.ads`, `*.adb`, например:

#### **`uiopen figure`**

Команда `uiloader` открывает файл в диалоге, управляемом пользователем, с использованием команды `load`.

Функция `uiimport` запускает Мастер импорта (Import Wizard), импортирующий из файла в текущей папке или буфера обмена Windows. Она соответствует выбору Import Data из меню File или выбору Paste Special из меню Edit MATLAB. `Uiimport (FILENAME)` - запускает Мастер Импорта, открывая файл `FILENAME`. Мастер импорта показывает данные для предварительного просмотра. В окне предварительного просмотра появляются данные и их представление в виде переменных MATLAB:

- 1) `uiimport ('-file')` - вначале выводит диалог выбора файла;
- 2) `uiimport ('-pastespecial')` - вначале выводит для предварительного просмотра содержимое буфера обмена Windows;
- 3) `S = uiimport (...)` хранит результирующие переменные как поля структуры `S`.

Команда `uisave` - управляемое пользователем сохранение с Windows диалогом.

Функция `saveas` - сохраняет рисунок или модель Simulink в желаемом формате на носителе информации или на устройстве, разрешенном print.

Функция `saveas (H, 'FILENAME')` - сохраняет данные в соответствии с командой дескрипторной графики `H` в файле `FILENAME`. Формат файла определяется расширением имени `FILENAME`.

Функция `saveas (H, 'FILENAME', 'FORMAT')` - выполняет то же, но с параметром `FORMAT` (формат задается тем же способом, что и расширение

имени файла и может от него отличаться). FORMAT имеет приоритет перед расширением имени файла.

Команда или функция delete удаляет файл или объект графики. «delete имя файла» удаляет файл текущей папки. Может быть использована \*.

Функция delete(H) - удаляет графический объект с дескриптором H. Если этот объект - окно, то оно предварительно закрывается.

Функция close(H) закрывает только графические окна. Для закрытия файлов необходимо использовать команду fclose.

Для записи файлов на диск служит команда save, используемая в довольно очевидных формах: save(h, filename), например:

```
h = actxcontrol('MwSamp.mwsampctrl.1');  
save(h, 'c:\temp\mycontrol.acx');
```

Для считывания файлов с диска служит команда load: load(h, filename). В этих командах имя файла указывается по правилам, принятым в операционных системах класса MS-DOS. Эти команды обычно дублируются кнопками панелей инструментов и браузером файлов.

## 5.2 Операции с двоичными файлами

Двоичными, или бинарными, называют файлы, данные которых представляют собой машинные коды. Основные операции с такими кодами перечислены ниже:

1) fid = fopen(filename) - открывает файл с именем filename и возвращает идентификатор fid со значением: 0 - чтение с клавиатуры; 1 - вывод на дисплей; 2 - вывод сообщения об ошибке; -1 - неудача в открытии файла с выводом сообщения message о типе ошибки. Идентификатор fid часто используется в качестве аргумента другими функциями и программами ввода-вывода. Имя файла filename может содержать путь к файлу;

2) fopen(filename, permission) - открывает файл с именем filename и параметром, определенным в permission, и возвращает идентификатор fid с перечисленными выше значениями: 0 - чтение с клавиатуры (permission установлено в 'r'); 1 - вывод на дисплей (permission установлено в 'a'); 2 - вывод сообщения об ошибке (permission установлен в 'a'); -1 - неудача в открытии файла с выводом сообщения message о типе ошибки. Идентификатор fid часто используется в качестве аргумента другими функциями и программами ввода-вывода. Параметр permission может принимать одно из следующих основных значений (другие см. в справочной системе):

- а) 'r' - открытие файла для чтения (по умолчанию);
- б) 'r+' - открытие файла для чтения и записи;
- в) 'w' - удаление содержимого существующего файла или создание нового и открытие его для записи;

г) 'a' - создание и открытие нового файла или открытие существующего для записи с добавлением в конец файла.

Добавление к этой строке 'b' предписывает системе открыть файл в двоичном режиме. Добавление вместо b к этой строке 't', например 'rt', в операционных системах, которые имеют различие между текстовыми и двоичными файлами, предписывает системе открыть файл в текстовом режиме.

Определенные вызовы функций fread или fwrite могут отменить числовой формат, заданный при вызове функции fopen:

1) fids = fopen ('all') - возвращает вектор-строку, содержащую идентификаторы всех открытых файлов, не включая стандартные потоки 0, 1 и 2. Число элементов вектора равно числу открытых пользователем файлов;

2) [filename, permission, format] = fopen(fid) - возвращает полное имя файла, строку permission и строку format. При использовании недопустимых значений fid возвращаются пустые строки для всех выходных аргументов.

Команда fclose закрывает файл:

1) status = fclose(fid) - закрывает файл, если он открыт. Возвращает статус файла status, равный 0, если закрытие завершилось успешно, и -1 в противном случае. Аргумент fid - это идентификатор, связанный с открытым;

2) status = fclose('all') закрывает все открытые файлы. Возвращает 0 в случае успешного завершения и -1 — в противном случае.

Функция чтения данных fread имеет следующие представления:

1) [A,count] = fread(fid, size, precision) - считывает двоичные данные из заданного файла и помещает их в матрицу A. Выходной аргумент count содержит число удачно считанных элементов. Значение идентификатора fid - это целое число, возвращенное функцией fopen; size - аргумент, определяющий количество считываемых данных. Если аргумент size не определен, функция fread считывает данные до конца файла;

2) [A,count] = fread(fid,size,precision,skip) - включает произвольный аргумент skip, который определяет число байтов, которые необходимо пропустить после каждого считывания.

Функция записи данных fwrite имеет следующие представления:

1) count=fwrite(fid,A,precision) - записывает элементы матрицы A в файл, представляя их с заданной точностью. Данные записываются в файл по столбцам, выходной аргумент count содержит число удачно записанных элементов. Значение идентификатора fid - это целое число, полученное при использовании функции fopen. Добавляет символы «возврат каретки» перед началом новой строки;

2) count=fwrite(fid, A, precision, skip) - делает то же, но включает произвольный аргумент skip, который определяет число байтов, которые надо пропустить перед каждой записью. Это полезно при вставке данных в несмежные области в записях фиксированной длины. Если precision имеет битовый формат, то значение skip определяется в битах.



### 5.3 Операции над форматированными файлами

Файлы, содержащие форматированные данные, называют форматированными файлами. Для них характерно использование следующих функций:

1) `line = fgetl(fid)` - возвращает строку из файла с идентификатором `fid` с удалением символа конца строки. Если функция `fgetl` обнаруживает конец файла, то она возвращает значение `-1`;

2) `line = fgets(fid)` - возвращает строку из файла с идентификатором `fid`, не удаляя символ конца строки. Если функция `fgets` обнаруживает конец файла, то она возвращает значение `-1`;

3) `line = fgets(fid, nchar)` - возвращает не больше чем `nchar` первых символов строки. После признака конца строки или конца файла никакие дополнительные символы не считываются;

4) `count = fprintf(fid, format, A, ...)` - форматирует данные, содержащиеся в действительной части матрицы `A`, под контролем строки `format` и записывает их в файл с идентификатором `fid`. Функция `fprintf` возвращает число записанных байтов. Значение идентификатора `fid` - целое число, возвращаемое функцией `fopen`. Если опустить идентификатор `fid` в списке аргументов функции `fprintf`, то вывод будет осуществляться на экран, так же как при использовании стандартного вывода (`fid=1`);

5) `fprintf(format, A, ...)` - запись осуществляется на стандартное устройство - экран (но не в файл). Строка `format` определяет систему счисления, выравнивание, значащие цифры, ширину поля и другие атрибуты выходного формата. Она может содержать обычные буквы алфавита наряду со спецификаторами, знаками выравнивания и т. д. Функция `fprintf` ведет себя, как аналогичная функция `fprintf` языка ANSI C и с некоторыми исключениями и расширениями;

6) `A = fscanf(fid, format)` - читает все данные из файла с идентификатором, `fid`, преобразует их согласно значению параметра `format` и возвращает в виде матрицы `A`. Значение идентификатора `fid` - целое число, возвращаемое функцией `fopen`. Параметр `format` представляет собой строку, определяющую формат данных, которые необходимо прочитать;

7) `[A, count] = fscanf(fid, format, size)` - считывает количество данных, определенное параметром `size`, преобразует их в соответствии с параметром `format` и возвращает вместе с количеством успешно считанных элементов `count`. Параметр `size` - это произвольный аргумент, определяющий количество считываемых данных. Допустимы следующие значения:

а) `n` - чтение `n` элементов в вектор-столбец;

б) `inf` - чтение элементов до конца файла и помещение их в вектор-столбец, содержащий такое же количество элементов, что и в файле;

в) `[m, n]` - считывает столько элементов, сколько требуется для заполнения матрицы размера `mхn`. Заполнение происходит по столбцам.

Строка `format` состоит из обычных символов и (или) спецификаторов. Спецификаторы указывают тип считываемых данных и включают символ `%`, опцию ширины поля и символы формата.

## 5.4 Позиционирование файла

При считывании и записи файлов они условно представляются в виде линейно расположенных данных. Место, с которого идет считывание в данный момент (или позиция, начиная с которой идет запись), определяется специальным указателем. Файлы последовательного доступа просматриваются строго от начала до конца, а в файлах произвольного доступа указатель может быть размещен в любом месте, начиная с которого ведется запись или считывание данных файла. Указатель обеспечивает позиционирование файлов. Имеется ряд функций позиционирования:

1) `eofstat = feof(fid)` - проверяет, достигнут ли конец файла с идентификатором `fid`. Возвращает 1, если указатель установлен на конец файла, и 0 - в противном случае;

2) `message = ferror(fid)` - возвращает сведения об ошибке в виде строки `message`. Аргумент `fid` - идентификатор открытого файла;

3) `message = ferror(fid,'clear')` - очищает индикатор ошибки для заданного файла;

4) `[message,errnum] = ferror(...)` - возвращает номер ошибки `errnum` последней операции ввода-вывода для заданного файла. Если последняя операция ввода-вывода, выполненная для определенного значением `fid` файла, была успешной, значение `message` - это пустая строка, а `errnum` принимает значение 0. Значение `errnum`, отличное от нуля, говорит о том, что при последней операции ввода-вывода произошла ошибка. Параметр `message` содержит строку, содержащую информацию о характере возникшей ошибки.

5) `frewind(fid)` - устанавливает указатель позиции в начало файла с идентификатором `fid`;

6) `status = fseek(fid,offset,origin)` - устанавливает указатель в файле с идентификатором `fid` в заданную позицию на байт, указанный параметром `offset` относительно `origin`.

7) `position=ftell(fid)` - возвращает позицию указателя для файла с идентификатором `fid`, полученным с помощью функции `fopen`. Выходной аргумент `position` - неотрицательное целое число, определяющее позицию указателя в байтах относительно начала файла. Если запрос был неудачным, `position` принимает значение `-1`;

8) `s=sprintf(format,A,...)` - форматирует данные в матрице `A` в формате, заданном параметром `format`, и создает из них строковую переменную `s`;

9) `[s, errmsg] = sprintf(format,A,...)` - аналогична ранее описанной функции, но дополнительно возвращает строку ошибки `errmsg`, если ошибка

имела место, или пустую строку в противном случае. Строка `format` определяет систему счисления, выравнивание, значащие цифры, ширину поля и другие атрибуты выходного формата. Она может содержать обычные символы наряду со спецификаторами, знаками выравнивания и т. д. Функция `sprintf` ведет себя, как и аналогичная функция `sprintf()` языка ANSI C, с некоторыми исключениями и расширениями.

Аргументы функции `fseek`:

- 1) `fid` - идентификатор файла, возвращенный функцией  `fopen`;
- 2) `offset` - значение, которое интерпретируется следующим образом:
  - а) `offset>0` - изменяет позицию указателя на `offset` байт в направлении к концу файла;
  - б) `offset=0` - не меняет позицию указателя;
  - в) `offset<0` - изменяет позицию указателя на `offset` байт в направлении к началу файла;
- 3) `origin` - аргумент, принимающий следующие значения:
  - а) `'bof'` или `-1` - начало файла;
  - б) `'cof'` или `0` - текущая позиция указателя в файле;
  - в) `'eof'` или `1` - конец файла;
- 4) `status` - выходной аргумент. Принимает значение `0`, если операция `fseek` произошла успешно, и `-1` в противном случае. Если произошла ошибка, используйте функцию `ferror` для получения более подробной информации.

Функция `sscanf` аналогична функции `fscanf` за исключением того, что она считывает данные из символьной переменной системы MATLAB, а не из файла:

1) `A = sscanf(s,format)` - считывает данные из символьной переменной `s`, преобразует их согласно значению `format` и создает на основе этих данных матрицу `A`. Параметр `format` определяет формат данных, которые нужно считать;

2) `A = sscanf(s,format,size)` - считывает количество данных, определенное параметром `size`, и преобразует их согласно строке `format`. Параметр `size` представляет собой аргумент, определяющий количество данных для чтения. Допустимы следующие значения параметра:

- а) `n` - чтение `n` элементов в вектор-столбец;
- б) `inf` - чтение элементов до конца символьной переменной и помещение их в вектор-столбец, содержащий такое же количество элементов, как и в строковой переменной;
- в) `[m, n]` - считывает столько элементов, сколько требуется для заполнения матрицы размера `mхn`. Заполнение происходит по столбцам. Величина `n` может принимать значение `Inf`.

3) `[A,count,errmsg,nextindex] = sscanf(...)` - считывает данные из символьной переменной `s`, преобразует их согласно значению `format` и возвращает в матрицу `A`. Параметр `count` - выходной аргумент, который возвращает число успешно считанных элементов; `errmsg` — выходной аргумент, который возвращает строку ошибки, если ошибка произошла, и

пустую строку в противном случае; `nextindex` — выходной аргумент, который содержит число, на единицу большее, чем количество символов в `s`. Строка `format` состоит из обычных символов и спецификаторов. Спецификаторы указывают тип данных и включают в себя символ `%`, опцию ширины поля и символы формата. Пояснения можно найти в описании функции `fscanf`.

## 6 Общие сведения по программированию в MATLAB

### 6.1 Переменные `varargin` и `varargout`

Для упрощения записи аргументов функций их можно представить списком, который определяет специальная переменная `varargin`, являющаяся массивом ячеек. Она должна записываться строчными буквами и может включать в себя как аргументы, так и опции функций, например:

```
% first function
function myplot(x,varargin)
plot(x,varargin{:})
% main file
myplot(sin(0:.1:1),'color',[.5 .7 .3],'linestyle',':')
```

Переменная `varargin` включает все входные параметры и опции начиная со второго аргумента. Здесь `varargin` представляет массив ячеек размера  $1 \times 4$ , включающий в себя значения : «'color', [.5 .7 .3], 'linestyle' и ':'»:

```
% The function
function [s,varargout] = mysize(x)
nout = max(nargout,1)-1;
s = size(x);
for k=1:nout, varargout(k) = {s(k)}; end
% main function
[s,rows,cols] = mysize(rand(4,5));
```

Аналогично `varargin` переменная `varargout` объединяет любое число выходных параметров в массив ячеек. Эта переменная, как и `varargin`, должна быть последней в списке аргументов. Обычно эта переменная не создается при вызове функций.

### 6.2 Управляющие структуры

Помимо программ с линейной структурой, инструкции которых исполняются строго по порядку, существует множество программ, структура которых нелинейна. При этом ветви программ могут выполняться в зависимости от определенных условий, иногда с конечным числом повторений - циклов, иногда в виде циклов, завершаемых при выполнении заданного условия. Для создания программ необходимы специальные управляющие структуры.

#### 6.2.1 Диалоговый ввод

Для диалогового ввода данных используется функция `input`:

`user_entry = input('string')` – на экране появляется выражение `string`, которое ожидает ввод данных в `user_entry`; функция возвращает введенное значение `user_entry`.

Функция `input` может использоваться для ввода произвольных строковых выражений в следующем виде: `input('Комментарий', V)`, где `V` – символьная переменная. При выполнении этой функции ожидается ввод строкового комментария. После ввода возвращается набранная строка, например:

```
reply = input('Do you want more? Y/N [Y]: ','s');
if isempty(reply)
    reply = 'Y';
end
```

### 6.2.2 Условный оператор

Условный оператор **if** в общем виде записывается следующим образом:

```
if Условие
    Инструкции_1
elseif Условие
    Инструкции_2
else
    Инструкции_3
end
```

Эта конструкция допускает несколько частных вариантов. В простейшем случае, типа `if...end`: `if Условие Инструкции end`.

Пока «Условие» возвращает логическое значение один (то есть «истина»), выполняются «Инструкции», составляющие тело структуры `if...end`. При этом оператор `end` указывает на конец перечня инструкций. Инструкции в списке разделяются оператором «`,`» (запятая) или «`;`» (точка с запятой). Если Условие не выполняется (дает логическое значение ноль, «ложь»), то «Инструкции» не выполняются.

Ниже приведенная конструкция выполняет «Инструкции\_1», если выполняется «Условие», или «Инструкции\_2» в противном случае:

```
if Условие
    Инструкции_1
else
    Инструкции_2
end
```

### 6.2.3 Циклы типа `for...end`

Циклы типа `for...end` обычно используются для организации вычислений с заданным числом повторяющихся циклов. Конструкция такого цикла имеет два альтернативных представления вида:

```
for var=Выражение  
    Инструкции  
end  
или
```

```
for var=Выражение, Инструкции, end
```

«Выражение» чаще всего записывается в виде `s:d:e`, где `s` - начальное значение переменной цикла `var`, `d` - приращение этой переменной и `e` — конечное значение управляющей переменной, при достижении которого цикл завершается. Возможна и запись в виде `s:e` (в этом случае `d=1`). Список выполняемых в цикле инструкций завершается оператором `end`.

Возможны вложенные циклы, например:

```
for i=1:3  
    for j=1:3  
        A(i,j)=i+j;  
    end  
end
```

Оператор `continue` передает управление в следующую итерацию цикла, пропуская операторы, которые записаны за ним, причем во вложенном цикле он передает управление на следующую итерацию основного цикла. Оператор `break` может использоваться для досрочного прерывания выполнения цикла. Как только он встречается в программе, цикл прерывается.

#### 6.2.4 Циклы типа `while...end`

Цикл типа `while` выполняется до тех пор, пока выполняется «Условие»:

```
while Условие Инструкции end
```

Досрочное завершение циклов реализуется с помощью операторов `break` или `continue`:

```
eps = 1;  
while (1+eps) > 1  
    eps = eps/2;  
end  
eps = eps*2
```

#### 6.2.5 Конструкция переключателя

Для осуществления множественного выбора (или ветвления) используется конструкция с переключателем типа switch:

```
switch switch_expr
  case case_expr
    statement,....,statement
  case {case_expr1,case_expr2,case_expr3,...}
    statement,....,statement
  ...
  otherwise
    statement,....,statement
end
```

Если выражение после заголовка switch имеет значение одного из выражений case\_expr, то выполняется блок операторов case, в противном случае - список инструкций после оператора otherwise. При выполнении блока case исполняются те списки инструкций, для которых case\_expr совпадает со switch\_expr. Обратите внимание на то, что case\_expr может быть числом, константой, переменной, вектором ячеек или даже строчной переменной. Пример:

```
method = 'Bilinear';
switch lower(method)
  case {'linear','bilinear'}
    disp('Method is linear')
  case 'cubic'
    disp('Method is cubic')
  case 'nearest'
    disp('Method is nearest')
  otherwise
    disp('Unknown method.')
end
% result
Method is linearend
```

### 6.2.6 Конструкция try...catch...end

Конструкция блока вывода ошибок: try...catch...end. Эта конструкция выполняет все списки инструкций. Если в каком-то списке до оператора catch появляется ошибка, то выводится сообщение об ошибке, но системная переменная последней ошибки lasterr не меняется. В сообщениях после catch сообщения об ошибке не выводятся:

```
try,
statement,
...,
statement,
catch,
statement,
```



```
...,
statement,
end
```

Пример реализации в системе MATLAB:

```
try
2+3
3+4
2/0
catch
4+5
end
% result
5
7
```

**Warning: Divide by zero.**

ans =

**Inf**

### 6.2.7 Создание паузы в вычислениях

Для остановки программы используется оператор `pause`. Он используется в следующих формах:

- 1) `pause` - останавливает вычисления до нажатия любой клавиши;
- 2) `pause(N)` - останавливает вычисления на N секунд;
- 3) `pause on` - включает режим отработки пауз;
- 4) `pause off` - выключает режим отработки пауз.

## 7 Разработка графического интерфейса пользователя

При разработке прикладных программ удобно создание графического интерфейса пользователя. Обязательными элементами графического интерфейса при решении задач:

а) одно или несколько окон для вывода графических и текстовых результатов расчета;

б) несколько редактируемых окон, с помощью которых задаются и/или изменяются значения параметров задачи;

в) управляющие кнопки, которые позволяют запускать и останавливать процесс расчета, перерисовывать результаты, выходить из задачи;

г) поясняющие надписи (статический текст);

д) прокручиваемые списки;

е) радио-кнопки для выбора одного из многих вариантов.

На рисунке 7.1 показан интерфейс, созданный для экспериментального модального анализа термдеформационного состояния станков.

Как видно из рисунка, в данном случае используются:

а) несколько окон для вывода графических и текстовых результатов расчета;

б) управляющие кнопки, которые позволяют запускать и останавливать процесс расчета, перерисовывать результаты, выходить из задачи;

в) поясняющие надписи;

г) прокручиваемые списки.

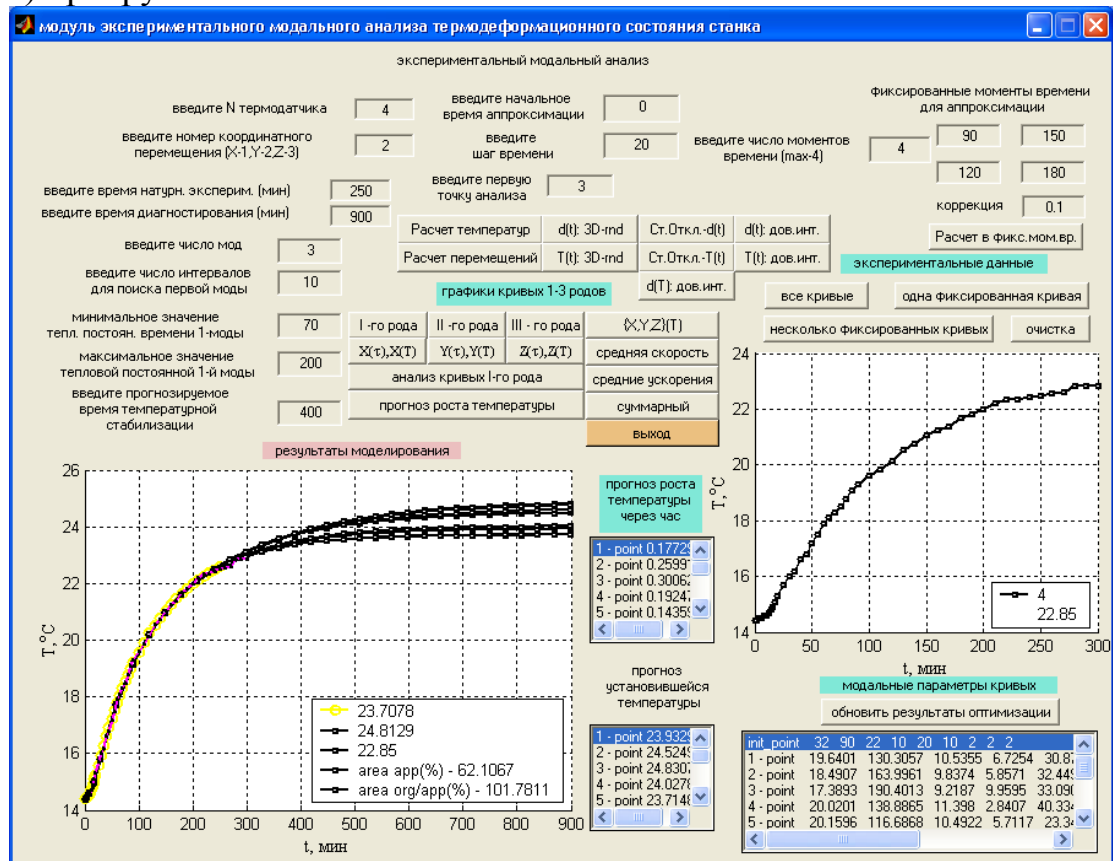


Рисунок 7.1- Графический интерфейс программы

Для создания такого интерфейса можно воспользоваться функциями графического вывода, а также специальной функцией, разработанной для интерактивного взаимодействия пользователя с рисунком. Эта функция называется `icontrol`. Но для упрощения работы и создания однотипных элементов интерфейса в системе MATLAB имеется специальная программа, которая позволяет на уровне визуального программирования создать требуемые элементы.

## 7.1 Создание внешнего вида интерфейса

В этом параграфе мы рассмотрим использование MATLAB для разработки внешнего вида графического интерфейса (GUI-GraphicsUserInterface) с использованием средств графического (визуального) программирования.

Для вызова визуального редактора необходимо в командном окне MATLAB набрать команду `guide`. По истечении определенного времени появятся два новых окна, показанные на рисунке 7.2

Одно из них – панель управления (Control Panel, на рисунке слева) и форма или область рисования (Figure, на рисунке справа). Показанная выше картинка появится на экране в том случае, если перед вызовом `guide` отсутствует какой-либо открытый рисунок. В случае же если функция `guide` вызывается после отрисовки какого-либо рисунка, то он открывается вместо пустого.

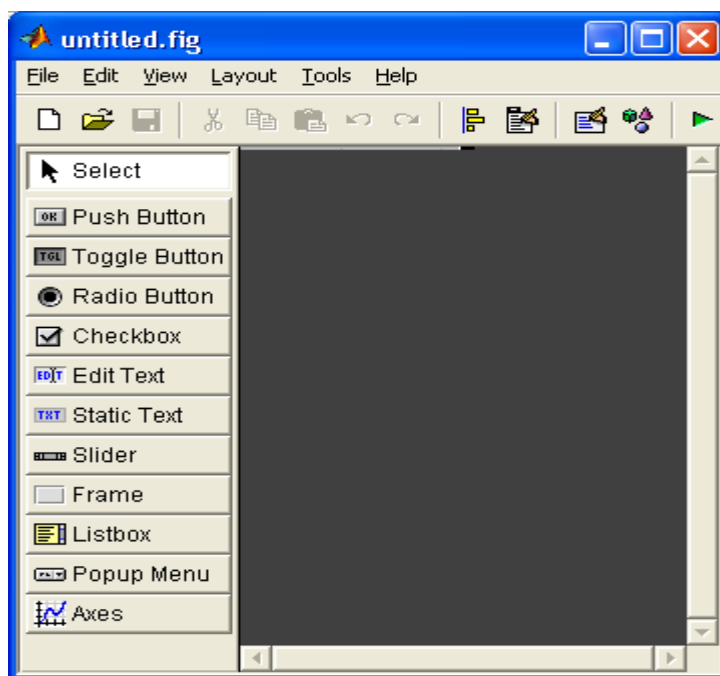


Рисунок 7.2 - Общий вид визуального графического редактора и окна редактирования

Перед созданием графического интерфейса желательно “разработать проект” того, что вы хотите иметь в качестве интерфейса.

Так как данный интерфейс использует одно рабочее окно, то первоначально необходимо выбрать размер этого окна. Для этого необходимо растянуть рабочее окно за нижний правый угол (рисунок 7.3)

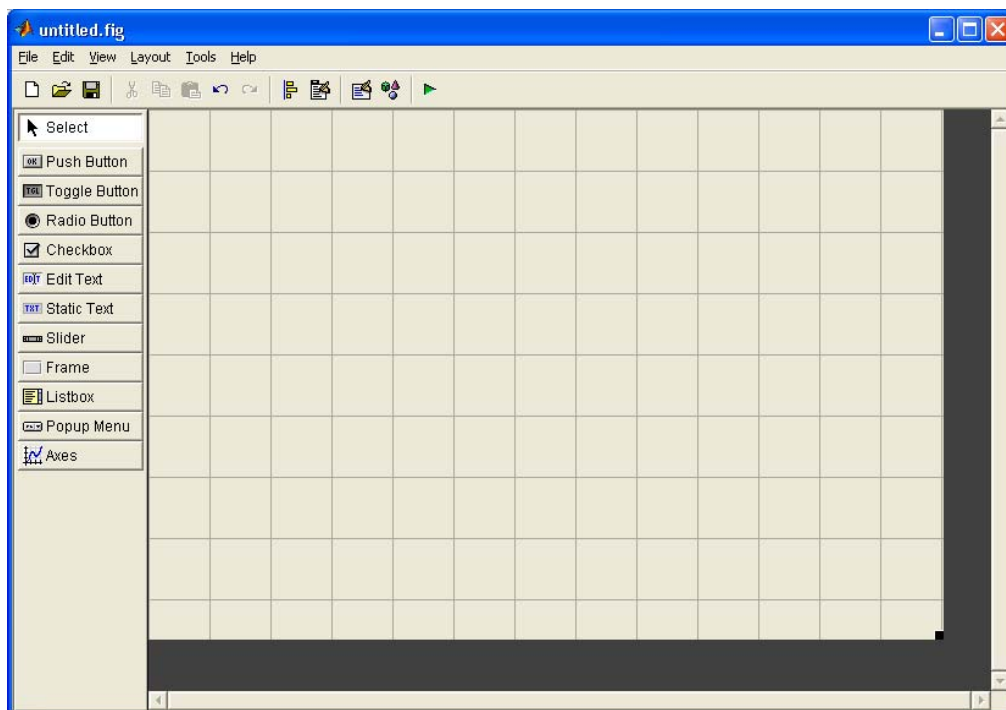


Рисунок 7.3 – Выбор предварительного размера рабочего окна интерфейса

Для создания статических надписей используются окна Static Text, а для ввода/редактирования данных Edit Text. Их размеры предварительно устанавливаются произвольно, а затем корректируются с учетом объема вводимой информации.

Щелкнув по нужному элементу на панели управления, и переведя мышь на панель рисунка, необходимо поместить крест, который будет на кончике мыши, в то место, где должен находиться левый верхний угол текста. Нажав и удерживая левую кнопку мыши, необходимо вытянуть получающийся прямоугольник до нужных размеров. Аналогично размещаются все элементы управления.

Для того чтобы внутри области статического текста появилась какая-либо надпись, необходима работа с редактором свойств, который вызывается либо при помощи кнопки Property inspector на панели, возникающей по нажатию правой кнопки мыши, либо с помощью двойного нажатия левой кнопкой мыши на соответствующем объекте на панели рисунка (рисунок 7.4). После чего откроется панель свойств (рисунок 7.5).

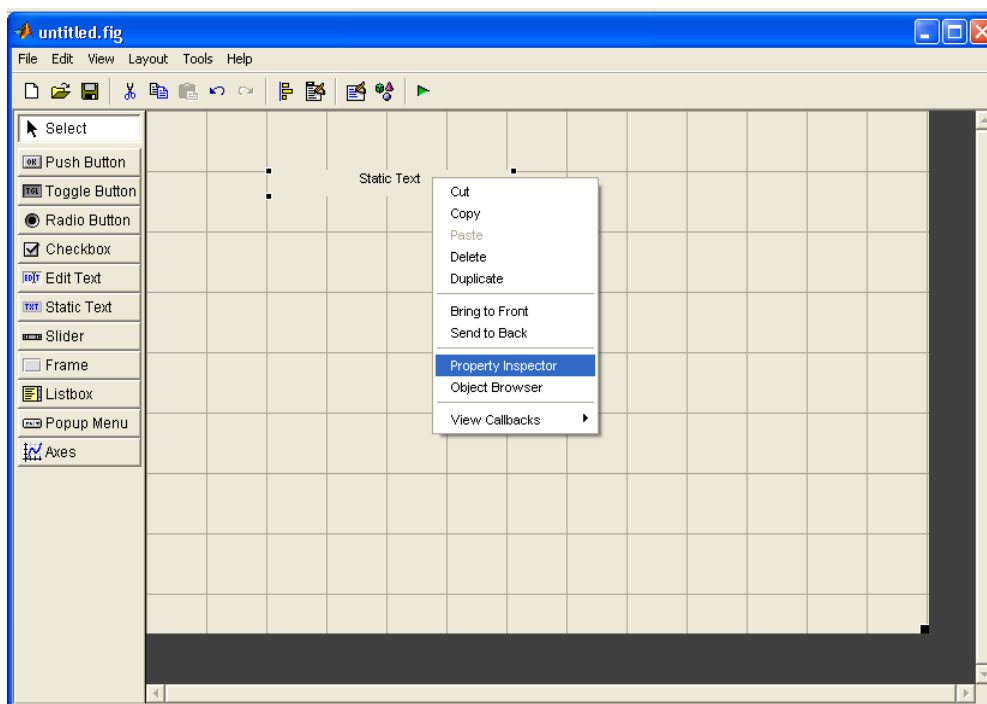


Рисунок 7.4 - Вызов панели свойств

Поскольку по умолчанию размеры всех объектов задаются в пикселях, это может привести к тому, что при изменении рабочего размера окна отдельные элементы управления могут наехать друг на друга. Для избежания такого нежелательного явления необходимо задать единицы размера всех объектов в безразмерных переменных – долях размера окна. Этот размер называется *normalized*. Для этого необходимо в редакторе свойств находим свойство *Units* (единицы измерения) и выбираем его щелчком левой кнопки мыши. После этого в средней части редактора в специальном окне слева появится свойство *Units*, а справа - окно выбора значений, в котором имеется раскрывающийся список с допустимыми значениями свойства. Для свойства *Units* необходимо выбрать значение *normalized*. Аналогично надо задать значение свойства *FontUnits* – единицы измерения размера шрифтов. Это обеспечивает изменение размера шрифта при изменении размера окна.

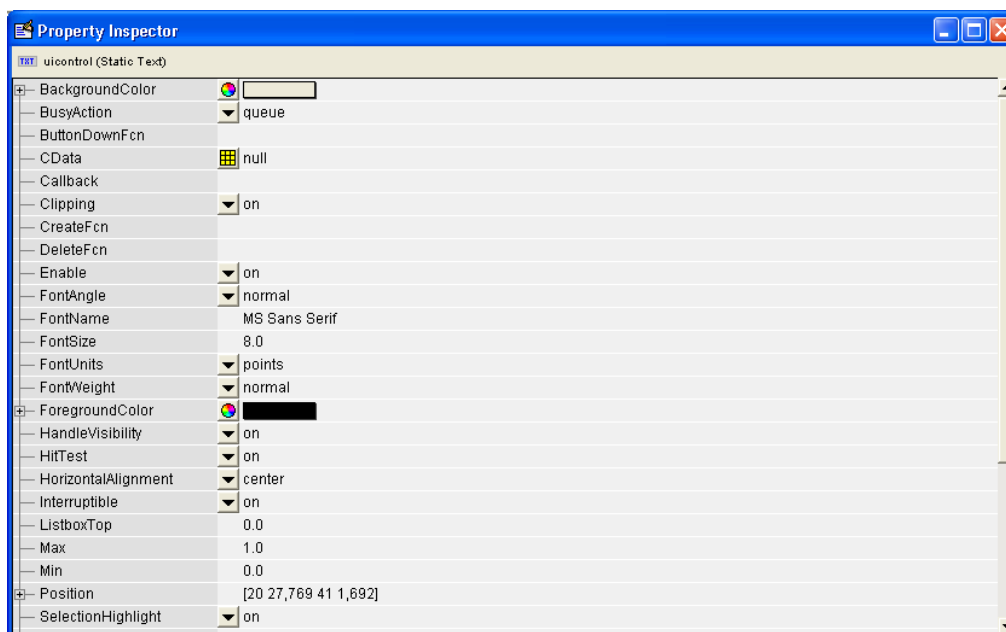


Рисунок 7.5 – Панель свойств

Для создания редактируемых окон ввода используется кнопка Edit Text, показанная справа. Используется она так же, как в предыдущем случае. Сначала появляется курсор в виде крестика, с помощью которого устанавливается требуемый размер прямоугольника ввода.

Для создания и размещения кнопок используется панель с надписью Push Button. Способ размещения кнопки и выбора ее размера полностью совпадает с методом, описанным выше.

Для создания подокон, в которые будут выводиться графики, используется кнопка Axes (окно и оси). Щелкнув по этому элементу на панели управления и переведя мышь на панель рисунка, необходимо поместить курсор-крестик, в то место, где должен находиться левый верхний угол первого подокна. Нажав и удерживая левую кнопку мыши, необходимо вытянуть получающийся прямоугольник до нужных размеров.

Для размещения надписей на кнопках и в области статического текста необходимо выделить соответствующий объект (либо двойным щелчком прямо в области рисунка, либо в верхнем окне редактора свойств) и в нижнем окне редактора свойств найти свойство String, и после его выделения вписать требуемый текст.

Построенные таким образом окна вывода и редактирования, окна статического текста и кнопки, а также другие объекты можно выровнять и установить определенные промежутки между ними с помощью панели выравнивания (Align Objects) (рисунок 7.6). Для этого необходимо на панели управления щелкнуть по соответствующей кнопке, и появится панель выравнивания. Для задания ряда объектов, с которыми будут выполняться какие-либо действия, необходимо их выделить, щелкая по каждому из них при нажатой клавише Shift. Выделенные объекты отмечаются черными точками вокруг соответствующих объектов. При необходимости изменить

размер какого-либо объекта (кнопки, окна и т.д.) необходимо щелкнуть по этому объекту с помощью левой кнопки мыши и с помощью мыши изменить требуемый размер так же, как и размер любого окна Windows.

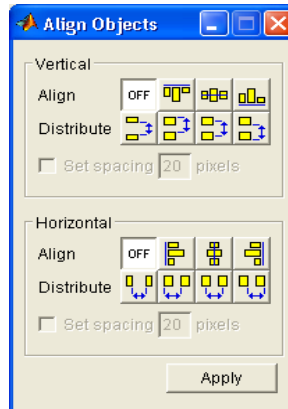


Рисунок 7.6 – Панель выравнивания

Для этого необходимо на панели управления щелкнуть по соответствующей кнопке, и появится панель выравнивания. Для задания ряда объектов, с которыми будут выполняться какие-либо действия, необходимо их выделить, щелкая по каждому из них при нажатой клавише Shift. Выделенные объекты отмечаются черными точками вокруг соответствующих объектов. При необходимости изменить размер какого-либо объекта (кнопки, окна и т.д.) необходимо щелкнуть по этому объекту с помощью левой кнопки мыши и с помощью мыши изменить требуемый размер так же, как и размер любого окна Windows.

Для задания надписей над каждым из окон вывода необходимо выделить соответствующее окно и вызвать редактор свойств, в нижнем окне которого надо найти свойство Name.

Следует еще отметить свойство Tag, которое имеют все объекты. Особенно важно знать и/или задать это свойство для тех объектов, к которым потом в программе придется обращаться. Например, если на рисунке задано несколько окон вывода, то для вывода графика в требуемое окно проще всего его будет идентифицировать окно с помощью свойства Tag. Это свойство, как правило, имеет по умолчанию вполне осмысленное значение (например, axes1), но при желании его можно изменить.

После разработки внешнего вида графического интерфейса с помощью визуального редактора необходимо сохранить это в виде m-файла. При нажатии на панели управления пункта меню File/Close control panel появится запрос на сохранение созданного интерфейса, и в случае положительного ответа будут созданы два файла – NAME.M и NAME.MAT (где NAME – заданное вами имя). Первый файл – это соответствующий текст программы, реализующий разработанный интерфейс, а второй – набор данных для него. При необходимости впоследствии внести изменения во внешний вид графического интерфейса нужно в командном окне MATLAB запустить файл

NAME.M, а потом с помощью команды `guide(gcf)` вызвать визуальный редактор с этим же файлом. Следует полностью выполнить этап разработки внешнего вида графического интерфейса, поскольку дальнейшая модификация текста программы NAME.M будет производиться вручную, путем изменения и/или дополнения текста программы определенными операторами. Если Вы после этого решите снова вызвать GUIDE – визуальный редактор формы, то все ваши изменения, внесенные вручную, при попытке сохранить доработанный интерфейс, пропадут. Поэтому сначала разрабатывайте внешний вид своего графического интерфейса, а потом его дорабатывайте.

## 7.2 Способы взаимодействия графического интерфейса с функциями пользователя

Созданный по описанной выше схеме GUI не обладает никакой функциональностью – он пока ничего не может делать за исключением стандартных функций Windows -менять размеры окна и закрывать окно. Основным средством взаимодействия графического интерфейса с функциями, выполняющими требуемые действия, является задание свойств кнопок и редактируемых окон, которое называется `Callback`. Значение (типа строка), которое присваивается этому свойству, и есть имя функции, которая вызывается при активации соответствующего объекта. Эта активация происходит при нажатии мышкой на соответствующую кнопку (если это свойство кнопки) и при нажатии клавиши `Enter` либо любой другой кнопки (если это свойство редактируемого окна).

Для задания этих свойств можно использовать визуальный редактор GUIDE (при вызванном рисунке) или с помощью стандартного редактора `m`-файлов дописать соответствующие параметры функции `icontrol` при обращении к ней в функции NAME.M, которую создал GUIDE. Например, для того, чтобы задать имя функции, вызываемой при нажатии фиксированной кнопки, необходимо вызвать соответствующий рисунок-интерфейс (запустить NAME.M) и после этого вызвать GUIDE.

Выделив соответствующий элемент при редактировании, и вызвав нажатием правой кнопки мыши дополнительную панель, следует сформировать соответствующий (по умолчанию) текст в `m`-файле. Для этого на возникшей панели следует перейти к строке `View Callbacks` и на новой панели активизировать строку `Callback`. Например, для элемента `PushButton1` и файла - `ex_mod_t_d10` в `m`-файле сформируется текст:

```
function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)
```

Если вызвать редактор свойств данной кнопки, то в списке свойств `Callback` появится текст вида:

```
ex_mod_t_d10('pushbutton1_Callback',gcbo,[],guidata(gcbo)))
```



Аналогично задаются функции и для других кнопок и редактируемых окон. При ручной правке функции NAME.M необходимо добавить в аргументы функции uicontrol строку вида: CallBack', 'Func\_run', ....

### 7.3 Примеры написания функций для реализации элементов управления

#### 7.3.1 Элемент управления Push button

Обязательным элементом функции для данного элемента является задание глобальных переменных (global) и дополнительных функций, например, устанавливающие конкретный вид расчета. В данном примере это функции data\_deck10, appr\_deck\_lt, appr\_deck\_st\_lt (рисунок 7.7).

```
function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)
global num_di x_init x_init_max x_init_n col_interval time_end time_end_default
time_diagn ze_rez col_mod num_d_default sti X_m init_curve_t sti_lt sti_sm;
x_init(2)=x_init_n;
[X,Y] = data_deck10(num_d_default);
sti=0; sti_sm=0; ze_rez=0; sti_lt=0;
sti_lt = appr_deck_lt(h, handles, num_di, x_init, x_init_max, col_interval, time_end,
time_end_default, time_diagn, col_mod, init_curve_t, X, Y);
[ze_rez, X_m, sti, sti_sm] = appr_deck_st_lt(h, handles, num_di, x_init, x_init_max,
col_interval, time_end, time_end_default, time_diagn, col_mod, init_curve_t, X, sti_lt);
```

Рисунок 7.7 – Текст фрагмента m-файла для Push Button1

#### 7.3.2 Элемент управления Edit Text

Кроме глобальных переменных данная функция включает обработку и преобразование символьных переменных. Для этого обязательно использование функций str2num, num2str и значений переменных Value и String. При необходимости можно использовать переприсвоение переменных. В данном случае элементы массива x\_init.

```
function varargout = edit9_Callback(h, eventdata, handles, varargin)
global x_init_n x_init;
x_init_n= str2num(get(handles.edit9,'String'));
set(handles.edit9,'Value',x_init_n);
set(handles.edit9,'String', num2str(get(handles.edit9,'Value')));
x_init(2)=x_init_n;
```

Рисунок 7.8 - Текст фрагмента m-файла для Edit9

#### 7.3.3 Элементы управления Axes и Push Button

Чтобы графическое представление результатов расчета появлялось по фиксированному событию здесь рассмотрен вариант события: нажатие

кнопки №12, считывание данных (функция data\_deck10), построение осей в отдельном окне (figure, axes), дополнительная функция обработки и вывода графиков.

```
function varargout = pushbutton12_Callback(h, eventdata, handles, varargin)
global num_di x_init x_init_max x_init_n col_interval time_end time_end_default
time_diagn ze_rez col_mod num_d_default X_m...
    sti rod2_Y sti_lt sti_sm;
x_init(2)=x_init_n;
[X,Y] = data_deck10(num_d_default);
h2=figure;
axes;
if sti(1,2)
    graf_curve1_fig(handles, time_diagn, time_end_default, time_end, col_interval, num_di,
X_m, sti, sti_lt, sti_sm, X, Y);
end
```

Рисунок 7.9 - Текст m-файла обработки Push Button12

```
function graf_curve1_fig(handles, time_diagn, time_end_default, time_end, col_interval,
num_di,...
    X_m, sti, sti_lt, sti_sm, X, Y)
    grid on; hold on
if time_diagn < 0 | time_diagn > 1400, n=max(find(X_m <=1440));
else n=max(find(X_m <=time_diagn));
end
if time_end < 0 | time_end > 300, n_end=max(find(X <=300));
else n_end=max(find(X <=time_end));
end
if sti_lt
plot(X(1:n_end),sti_lt(1:n_end),'-oy','LineWidth',2);
end
for i=1:col_interval
    plot(X_m(1:n),sti(i,:),'-ok','LineWidth',2,'MarkerEdgeColor','k',...
        'MarkerFaceColor','w','MarkerSize',4);
max_sti(i) = max(sti(i,:));
end
    plot(X,Y(:,num_di),'-xm','LineWidth',2,'MarkerEdgeColor','k',...
        'MarkerFaceColor','w','MarkerSize',4);
xlabel('t, мин','FontName','Times New Roman'); ylabel('T, ^oC','FontName','Times New
Roman');
max_Y=max(max_sti); min_Y=min(max_sti);
legend(num2str(min_Y),num2str(max_Y),num2str(max(Y(:,num_di))),4);
```

Рисунок 7.10 – Текст функции graf\_curve1\_fig

### 7.3.4 Пример использования элемента управления listbox

Фактически использование функции listbox сводится к коду вида:

```
set(handles.listbox1,'String',ResultsStr);
```

Однако, учитывая необходимость подготовки выводимой информации используется комбинация двух функций: Push Button и listbox. В этом случае по нажатию зарезервированной кнопки (в данном случае Button11) активизируется функция updat\_listbox (рисунок 7.11), в которой выполняется подготовка необходимой к выводу в прокручиваемом окне (listbox1) информации (рисунок 7.12).

```
function varargout = pushbutton11_Callback(h, eventdata, handles, varargin)
    updat_listbox(handles)
```

Рисунок 7.11 – Пример функции, вызывающей функцию listbox1

```
function updat_listbox(handles)
    global x_init ze_rez col_interval col_mod;
    if col_mod == 1
        ResultsStr(1) = {'init_point ', num2str(x_init(1,1)), ' ', num2str(x_init(1,2)),
            ', num2str(x_init(1,3))};
        for i=1:col_interval
            ResultsStr(i+1) = {[num2str(i), ' - point ', num2str(ze_rez(i,1)),
            ', num2str(ze_rez(i,2)), ' ', num2str(ze_rez(i,3))]};
        end
    elseif col_mod == 2
        ResultsStr(1) = {'init_point ', num2str(x_init(1,1)), ' ', num2str(x_init(1,2)),
            ', num2str(x_init(1,3)), ' ', ...
            num2str(x_init(1,4)), ' ', num2str(x_init(1,5)), ' ', num2str(x_init(1,6))};
        for i=1:col_interval
            ResultsStr(i+1) = {[num2str(i), ' - point ', num2str(ze_rez(i,1)),
            ', num2str(ze_rez(i,2)), ' ', num2str(ze_rez(i,3)), ' ', ...
            num2str(ze_rez(i,4)), ' ', num2str(ze_rez(i,5)), ' ', num2str(ze_rez(i,6))]};
        end
    elseif col_mod > 2
        ResultsStr(1) = {'init_point ', num2str(x_init(1,1)), ' ', num2str(x_init(1,2)),
            ', num2str(x_init(1,3)), ' ', num2str(x_init(1,4)), ' ', num2str(x_init(1,5)),
            ', num2str(x_init(1,6)), ' ', ...
            num2str(x_init(1,7)), ' ', num2str(x_init(1,8)), ' ', num2str(x_init(1,9))};
        for i=1:col_interval
            ResultsStr(i+1) = {[num2str(i), ' - point ', num2str(ze_rez(i,1)),
            ', num2str(ze_rez(i,2)), ' ', num2str(ze_rez(i,3)), ' ', num2str(ze_rez(i,4)),
            ', num2str(ze_rez(i,5)), ' ', num2str(ze_rez(i,6)), ...
            ' ', num2str(ze_rez(i,7)), ' ', num2str(ze_rez(i,8)), ' ', num2str(ze_rez(i,9))]};
        end
    end
    set(handles.listbox1, 'String', ResultsStr);
```

Рисунок 7.12 – Текст m-файла

### 7.3.5 Пример использования элемента управления Radio Button

Применение элемента управления Radio Button покажем на примере выбора ряда геометрической прогрессии в кинематическом расчете коробок передач (рисунок 7.13). В соответствии с теорией кинематического расчета

все многообразие знаменателей ряда сводится к небольшому набору фиксированных значений - в данном примере это значения 1,12; 1,26; 1,41; 1,58. Поэтому, одним из наиболее удобных вариантов реализации выбора одного значения является использование элемента управления Radio Button.

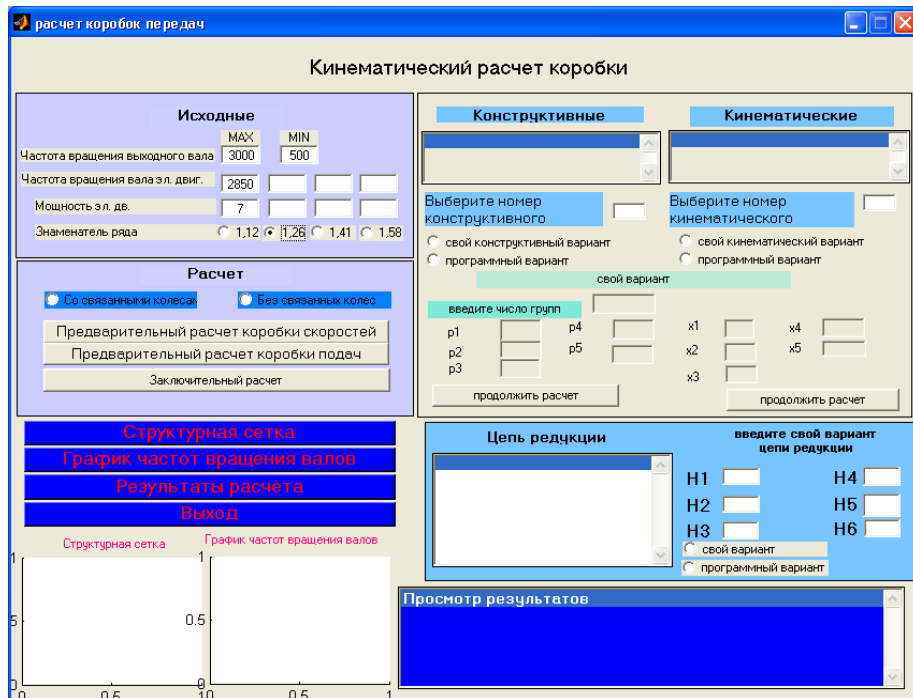


Рисунок 7.13

Для исключения совместного включения кнопок Radio Button в соответствующих функциях необходимо использовать функцию отключения n-1- кнопок, при условии активности текущей кнопки. Это реализуется функцией `mutual_exclude` (рисунок 7.14)

```
function varargout = radiobutton1_Callback(h, eventdata, handles, varargin)
global f;
off = [handles.radiobutton2, handles.radiobutton3, handles.radiobutton4];
mutual_exclude(off)
f=1.12;
function varargout = radiobutton2_Callback(h, eventdata, handles, varargin)
global f;
off = [handles.radiobutton1, handles.radiobutton3, handles.radiobutton4];
mutual_exclude(off)
f=1.26;
```

Рисунок 7.14 – Фрагмент m-файла, реализующего элемент управления Radio Button, лист 1

```
% -----
function varargout = radiobutton3_Callback(h, eventdata, handles, varargin)
global f;
off = [handles.radiobutton1, handles.radiobutton2, handles.radiobutton4];
mutual_exclude(off)
```

```

f=1.41;

% -----
function varargout = radiobutton4_Callback(h, eventdata, handles, varargin)
global f;
off = [handles.radiobutton1, handles.radiobutton2, handles.radiobutton3];
mutual_exclude(off)
f=1.58;

function mutual_exclude(off)
set(off,'Value',0)

```

Рисунок 7.14, лист 2

### 7.3.6 Пример управляющей программы

Чтобы представить сложность и простоту создания интерфейса в системе MATLAB, ниже приведен m-файл управляющей программы (рисунок 7.15), реализующий интерфейс, изображенный на рисунке 7.1. Конечно, данный код не позволит осуществлять управление всеми элементами управления. Для этого необходимо дописывать соответствующий код программы в соответствии с инструкциями, приведенными выше.

```

function varargout = ex_mod_t_dl10(varargin)
global num_di time_end time_end_default num_d_default dlit_default col_mod;
global x_init col_interval x_init_max time_diagn ze_rez rod2_Y rod3_Y rod4_Y
init_rod_Y init_curve_t ocen_stat sti_lt sti_sm;
global time_step rez_y time_1 time_2 time_3 time_4 col_fix_time korrec_y;
warning off;
num_d_default=11; dlit_default=300; time_end_default=300;
x_init=[32., 90., 22., 10., 20, 10., 2., 2., 2.]; % - initial point of optimization
if nargin == 0 % LAUNCH GUI
    fig = openfig(mfilename,'reuse');
    % Use system color scheme for figure:
    set(fig,'Color',get(0,'defaultUicontrolBackgroundColor'));
    % Generate a structure of handles to pass to callbacks, and store it.
    handles = guihandles(fig);
    ResultsStr = {'init_point ',num2str(x_init(1)), ' ',num2str(x_init(2)), '...
',num2str(x_init(3)), ' ',...
                num2str(x_init(4)), ' ',num2str(x_init(5)), ' ',num2str(x_init(6)), '...
',num2str(x_init(7)),...
                ' ',num2str(x_init(8)), ' ',num2str(x_init(9))]];
    set(handles.listbox1,'String',ResultsStr);

    Рисунок 7.15 – Текст m-файла управляющей программы, лист 1
    set(handles.listbox2,'String','max -0.1');
    guidata(fig, handles);
    if nargin > 0
        varargout{1} = fig;

```

```
    end
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
    try
        if (nargout)
            [varargout{1:nargout}] = feval(varargin{:}); % FEVAL
switchyard
        else
            feval(varargin{:}); % FEVAL switchyard
        end
    catch
        disp(lasterr);
    end
end
end
```

Рисунок 7.15, лист 2

## Приложение А (обязательное)

### Лабораторная работа 1 Решение систем линейных алгебраических уравнений

Задание 1. Найдите решения систем уравнений с использованием функций MATLAB, сравните полученные результаты с построением диаграмм.

Задание 2. Найдите собственные значения и собственные векторы. Приведите графическое представление результатов.

Таблица А.1 - Варианты заданий

	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 50%; text-align: center;">А</th> <th style="width: 50%; text-align: center;">В</th> </tr> <tr> <td style="text-align: center;">1   0.47   -0.11   0.55</td> <td style="text-align: center;">.1.33</td> </tr> <tr> <td style="text-align: center;">0.42   1   0.35   0.17</td> <td style="text-align: center;">1.29</td> </tr> <tr> <td style="text-align: center;">-0.25   0.67   1   0.36</td> <td style="text-align: center;">2.11</td> </tr> <tr> <td style="text-align: center;">0.54   -0.32   -0.74   1</td> <td style="text-align: center;">0.10</td> </tr> </table>	А	В	1   0.47   -0.11   0.55	.1.33	0.42   1   0.35   0.17	1.29	-0.25   0.67   1   0.36	2.11	0.54   -0.32   -0.74   1	0.10	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 75%; text-align: center;">С</th> <th style="width: 25%; text-align: center;">D</th> </tr> <tr> <td style="text-align: center;">1   2   3</td> <td style="text-align: center;">13</td> </tr> <tr> <td style="text-align: center;">2   3   5</td> <td style="text-align: center;">4</td> </tr> <tr> <td style="text-align: center;">3   5   9</td> <td style="text-align: center;">17</td> </tr> </table>	С	D	1   2   3	13	2   3   5	4	3   5   9	17
А	В																			
1   0.47   -0.11   0.55	.1.33																			
0.42   1   0.35   0.17	1.29																			
-0.25   0.67   1   0.36	2.11																			
0.54   -0.32   -0.74   1	0.10																			
С	D																			
1   2   3	13																			
2   3   5	4																			
3   5   9	17																			
№1	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0.63   1   0.11   0.34</td> <td style="text-align: center;">2.08</td> </tr> <tr> <td style="text-align: center;">0.17   1.18   -0.45   0.11</td> <td style="text-align: center;">0.17</td> </tr> <tr> <td style="text-align: center;">0.31   -0.15   1.17   -2.35</td> <td style="text-align: center;">1.28</td> </tr> <tr> <td style="text-align: center;">0.58   0.21   -3.45   -1.18</td> <td style="text-align: center;">0.05</td> </tr> </table>	0.63   1   0.11   0.34	2.08	0.17   1.18   -0.45   0.11	0.17	0.31   -0.15   1.17   -2.35	1.28	0.58   0.21   -3.45   -1.18	0.05	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">1   2   3</td> <td style="text-align: center;">0.55</td> </tr> <tr> <td style="text-align: center;">1   4   9</td> <td style="text-align: center;">1.35</td> </tr> <tr> <td style="text-align: center;">1   8   27</td> <td style="text-align: center;">3.55</td> </tr> </table>	1   2   3	0.55	1   4   9	1.35	1   8   27	3.55				
0.63   1   0.11   0.34	2.08																			
0.17   1.18   -0.45   0.11	0.17																			
0.31   -0.15   1.17   -2.35	1.28																			
0.58   0.21   -3.45   -1.18	0.05																			
1   2   3	0.55																			
1   4   9	1.35																			
1   8   27	3.55																			
№2	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0.77   0.04   -0.21   0.18</td> <td style="text-align: center;">1.24</td> </tr> <tr> <td style="text-align: center;">-0.45   1.23   -0.06   0</td> <td style="text-align: center;">-0.88</td> </tr> <tr> <td style="text-align: center;">-0.26   -0.34   1.11   0</td> <td style="text-align: center;">0.62</td> </tr> <tr> <td style="text-align: center;">-0.05   0.26   -0.34   1.12</td> <td style="text-align: center;">-1.17</td> </tr> </table>	0.77   0.04   -0.21   0.18	1.24	-0.45   1.23   -0.06   0	-0.88	-0.26   -0.34   1.11   0	0.62	-0.05   0.26   -0.34   1.12	-1.17	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0.42   1.43   0.27</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">1.43   -0.84   0.93</td> <td style="text-align: center;">2</td> </tr> <tr> <td style="text-align: center;">0.27   0.93   -0.48</td> <td style="text-align: center;">3</td> </tr> </table>	0.42   1.43   0.27	1	1.43   -0.84   0.93	2	0.27   0.93   -0.48	3				
0.77   0.04   -0.21   0.18	1.24																			
-0.45   1.23   -0.06   0	-0.88																			
-0.26   -0.34   1.11   0	0.62																			
-0.05   0.26   -0.34   1.12	-1.17																			
0.42   1.43   0.27	1																			
1.43   -0.84   0.93	2																			
0.27   0.93   -0.48	3																			
№3	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0.79   -0.12   0.34   0.16</td> <td style="text-align: center;">-0.64</td> </tr> <tr> <td style="text-align: center;">-0.34   1.18   -0.17   0.18</td> <td style="text-align: center;">1.42</td> </tr> <tr> <td style="text-align: center;">-0.16   -0.34   0.85   0.31</td> <td style="text-align: center;">-0.42</td> </tr> <tr> <td style="text-align: center;">-0.12   0.26   0.08   0.75</td> <td style="text-align: center;">0.83</td> </tr> </table>	0.79   -0.12   0.34   0.16	-0.64	-0.34   1.18   -0.17   0.18	1.42	-0.16   -0.34   0.85   0.31	-0.42	-0.12   0.26   0.08   0.75	0.83	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0.64   0.54   -0.33</td> <td style="text-align: center;">3</td> </tr> <tr> <td style="text-align: center;">0.54   -0.92   0.24</td> <td style="text-align: center;">2</td> </tr> <tr> <td style="text-align: center;">-0.33   0.24   0.78</td> <td style="text-align: center;">1</td> </tr> </table>	0.64   0.54   -0.33	3	0.54   -0.92   0.24	2	-0.33   0.24   0.78	1				
0.79   -0.12   0.34   0.16	-0.64																			
-0.34   1.18   -0.17   0.18	1.42																			
-0.16   -0.34   0.85   0.31	-0.42																			
-0.12   0.26   0.08   0.75	0.83																			
0.64   0.54   -0.33	3																			
0.54   -0.92   0.24	2																			
-0.33   0.24   0.78	1																			
№4	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">-0.68   -0.18   0.02   0.21</td> <td style="text-align: center;">-1.83</td> </tr> <tr> <td style="text-align: center;">0.16   -0.88   -0.14   0.27</td> <td style="text-align: center;">0.65</td> </tr> <tr> <td style="text-align: center;">0.37   0.27   -1.02   -0.24</td> <td style="text-align: center;">-2.23</td> </tr> <tr> <td style="text-align: center;">0.12   0.21   -0.18   -0.75</td> <td style="text-align: center;">1.13</td> </tr> </table>	-0.68   -0.18   0.02   0.21	-1.83	0.16   -0.88   -0.14   0.27	0.65	0.37   0.27   -1.02   -0.24	-2.23	0.12   0.21   -0.18   -0.75	1.13	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">0.5   1.77   0.39</td> <td style="text-align: center;">1.5</td> </tr> <tr> <td style="text-align: center;">0.84   1.79   0.95</td> <td style="text-align: center;">2.5</td> </tr> <tr> <td style="text-align: center;">0.24   1.03   -0.41</td> <td style="text-align: center;">3</td> </tr> </table>	0.5   1.77   0.39	1.5	0.84   1.79   0.95	2.5	0.24   1.03   -0.41	3				
-0.68   -0.18   0.02   0.21	-1.83																			
0.16   -0.88   -0.14   0.27	0.65																			
0.37   0.27   -1.02   -0.24	-2.23																			
0.12   0.21   -0.18   -0.75	1.13																			
0.5   1.77   0.39	1.5																			
0.84   1.79   0.95	2.5																			
0.24   1.03   -0.41	3																			
№5																				

Продолжение таблица А.1

№6	<table border="1"> <tr><td>-</td><td>-</td><td>0.03</td><td>0</td><td>-</td></tr> <tr><td>0.58</td><td>0.32</td><td></td><td></td><td>0.44</td></tr> <tr><td>0.11</td><td>-</td><td>-</td><td>0</td><td>-</td></tr> <tr><td></td><td>1.26</td><td>0.36</td><td></td><td>1.42</td></tr> <tr><td>0.12</td><td>0.08</td><td>-</td><td>-</td><td>0.83</td></tr> <tr><td></td><td></td><td>1.14</td><td>0.24</td><td></td></tr> <tr><td>0.15</td><td>-</td><td>-</td><td>0</td><td>-</td></tr> <tr><td></td><td>0.35</td><td>0.18</td><td></td><td>1.42</td></tr> </table>	-	-	0.03	0	-	0.58	0.32			0.44	0.11	-	-	0	-		1.26	0.36		1.42	0.12	0.08	-	-	0.83			1.14	0.24		0.15	-	-	0	-		0.35	0.18		1.42	<table border="1"> <tr><td>0.19</td><td>0.51</td><td>0.86</td><td>-</td><td>0.35</td></tr> <tr><td>0.51</td><td>0.32</td><td>0.95</td><td>-</td><td>0.42</td></tr> <tr><td>0.86</td><td>0.95</td><td>-</td><td>0.12</td><td>-</td></tr> <tr><td></td><td></td><td></td><td></td><td>0.45</td></tr> </table>	0.19	0.51	0.86	-	0.35	0.51	0.32	0.95	-	0.42	0.86	0.95	-	0.12	-					0.45										
-	-	0.03	0	-																																																																				
0.58	0.32			0.44																																																																				
0.11	-	-	0	-																																																																				
	1.26	0.36		1.42																																																																				
0.12	0.08	-	-	0.83																																																																				
		1.14	0.24																																																																					
0.15	-	-	0	-																																																																				
	0.35	0.18		1.42																																																																				
0.19	0.51	0.86	-	0.35																																																																				
0.51	0.32	0.95	-	0.42																																																																				
0.86	0.95	-	0.12	-																																																																				
				0.45																																																																				
№7	<table border="1"> <tr><td>-</td><td>0.31</td><td>-</td><td>0.22</td><td>-</td></tr> <tr><td>0.83</td><td></td><td>0.18</td><td></td><td>1.71</td></tr> <tr><td>-</td><td>-</td><td>0</td><td>0.22</td><td>-</td></tr> <tr><td>0.21</td><td>0.67</td><td></td><td></td><td>0.62</td></tr> <tr><td>0.32</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td></td><td>0.18</td><td>0.95</td><td>0.19</td><td>0.89</td></tr> <tr><td>0.12</td><td>0.28</td><td>-</td><td>-1</td><td>-</td></tr> <tr><td></td><td></td><td>0.14</td><td></td><td>0.94</td></tr> </table>	-	0.31	-	0.22	-	0.83		0.18		1.71	-	-	0	0.22	-	0.21	0.67			0.62	0.32	-	-	-	-		0.18	0.95	0.19	0.89	0.12	0.28	-	-1	-			0.14		0.94	<table border="1"> <tr><td>0.64</td><td>1.54</td><td>-</td><td>-</td><td>0.3</td></tr> <tr><td></td><td></td><td>0.33</td><td></td><td></td></tr> <tr><td>1.54</td><td>-</td><td>0.24</td><td>-</td><td>0.2</td></tr> <tr><td></td><td>0.92</td><td></td><td></td><td></td></tr> <tr><td>-</td><td>0.24</td><td>0.78</td><td>-</td><td>0.1</td></tr> <tr><td>0.33</td><td></td><td></td><td></td><td></td></tr> </table>	0.64	1.54	-	-	0.3			0.33			1.54	-	0.24	-	0.2		0.92				-	0.24	0.78	-	0.1	0.33				
-	0.31	-	0.22	-																																																																				
0.83		0.18		1.71																																																																				
-	-	0	0.22	-																																																																				
0.21	0.67			0.62																																																																				
0.32	-	-	-	-																																																																				
	0.18	0.95	0.19	0.89																																																																				
0.12	0.28	-	-1	-																																																																				
		0.14		0.94																																																																				
0.64	1.54	-	-	0.3																																																																				
		0.33																																																																						
1.54	-	0.24	-	0.2																																																																				
	0.92																																																																							
-	0.24	0.78	-	0.1																																																																				
0.33																																																																								
№8	<table border="1"> <tr><td>-</td><td>0.27</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>0.87</td><td></td><td>0.22</td><td>0.18</td><td>1.21</td></tr> <tr><td>-</td><td>-1</td><td>-</td><td>0.18</td><td>-</td></tr> <tr><td>0.21</td><td></td><td>0.45</td><td></td><td>0.33</td></tr> <tr><td>0.12</td><td>0.13</td><td>-</td><td>0.18</td><td>-</td></tr> <tr><td></td><td></td><td>0.33</td><td></td><td>0.48</td></tr> <tr><td>0.33</td><td>-</td><td>0</td><td>-1</td><td>-</td></tr> <tr><td></td><td>0.41</td><td></td><td></td><td>1.21</td></tr> </table>	-	0.27	-	-	-	0.87		0.22	0.18	1.21	-	-1	-	0.18	-	0.21		0.45		0.33	0.12	0.13	-	0.18	-			0.33		0.48	0.33	-	0	-1	-		0.41			1.21	<table border="1"> <tr><td>0.55</td><td>1.77</td><td>0.39</td><td>-</td><td>1.5</td></tr> <tr><td>0.84</td><td>1.79</td><td>0.95</td><td>-</td><td>2.5</td></tr> <tr><td>0.24</td><td>1.03</td><td>-0.41</td><td>-</td><td>3</td></tr> </table>	0.55	1.77	0.39	-	1.5	0.84	1.79	0.95	-	2.5	0.24	1.03	-0.41	-	3															
-	0.27	-	-	-																																																																				
0.87		0.22	0.18	1.21																																																																				
-	-1	-	0.18	-																																																																				
0.21		0.45		0.33																																																																				
0.12	0.13	-	0.18	-																																																																				
		0.33		0.48																																																																				
0.33	-	0	-1	-																																																																				
	0.41			1.21																																																																				
0.55	1.77	0.39	-	1.5																																																																				
0.84	1.79	0.95	-	2.5																																																																				
0.24	1.03	-0.41	-	3																																																																				
№9	<table border="1"> <tr><td>-</td><td>-</td><td>0.38</td><td>-</td><td>-</td></tr> <tr><td>0.81</td><td>0.07</td><td></td><td>0.21</td><td>0.81</td></tr> <tr><td>-</td><td>-</td><td>0.11</td><td>0.33</td><td>-</td></tr> <tr><td>0.22</td><td>0.92</td><td></td><td></td><td>0.64</td></tr> <tr><td>0.51</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td></td><td>0.07</td><td>0.81</td><td>0.11</td><td>1.71</td></tr> <tr><td>0.33</td><td>-</td><td>0</td><td>-1</td><td>-</td></tr> <tr><td></td><td>0.41</td><td></td><td></td><td>1.21</td></tr> </table>	-	-	0.38	-	-	0.81	0.07		0.21	0.81	-	-	0.11	0.33	-	0.22	0.92			0.64	0.51	-	-	-	-		0.07	0.81	0.11	1.71	0.33	-	0	-1	-		0.41			1.21	<table border="1"> <tr><td>0.59</td><td>1.77</td><td>1.39</td><td>-</td><td>1.5</td></tr> <tr><td>0.84</td><td>1.79</td><td>0.95</td><td>-</td><td>2.5</td></tr> <tr><td>1.24</td><td>1.03</td><td>-0.41</td><td>-</td><td>3</td></tr> </table>	0.59	1.77	1.39	-	1.5	0.84	1.79	0.95	-	2.5	1.24	1.03	-0.41	-	3															
-	-	0.38	-	-																																																																				
0.81	0.07		0.21	0.81																																																																				
-	-	0.11	0.33	-																																																																				
0.22	0.92			0.64																																																																				
0.51	-	-	-	-																																																																				
	0.07	0.81	0.11	1.71																																																																				
0.33	-	0	-1	-																																																																				
	0.41			1.21																																																																				
0.59	1.77	1.39	-	1.5																																																																				
0.84	1.79	0.95	-	2.5																																																																				
1.24	1.03	-0.41	-	3																																																																				
№10	<table border="1"> <tr><td>-1</td><td>0.22</td><td>-</td><td>0.31</td><td>-</td></tr> <tr><td></td><td></td><td>0.11</td><td></td><td>-2.7</td></tr> <tr><td>0.38</td><td>-1</td><td>-</td><td>0.22</td><td>-</td></tr> <tr><td></td><td></td><td>0.12</td><td></td><td>1.5</td></tr> <tr><td>0.11</td><td>0.23</td><td>1</td><td>-</td><td>-</td></tr> <tr><td></td><td></td><td></td><td>0.51</td><td>1.2</td></tr> <tr><td>0.17</td><td>-</td><td>0.31</td><td>-1</td><td>-</td></tr> <tr><td></td><td>0.21</td><td></td><td></td><td>0.17</td></tr> </table>	-1	0.22	-	0.31	-			0.11		-2.7	0.38	-1	-	0.22	-			0.12		1.5	0.11	0.23	1	-	-				0.51	1.2	0.17	-	0.31	-1	-		0.21			0.17	<table border="1"> <tr><td>1.42</td><td>1.43</td><td>0.27</td><td>-</td><td>0.1</td></tr> <tr><td>1.43</td><td>-</td><td>0.93</td><td>-</td><td>0.2</td></tr> <tr><td></td><td>0.84</td><td></td><td></td><td></td></tr> <tr><td>0.27</td><td>0.93</td><td>-</td><td>0.48</td><td>-</td></tr> <tr><td></td><td></td><td></td><td></td><td>0.3</td></tr> </table>	1.42	1.43	0.27	-	0.1	1.43	-	0.93	-	0.2		0.84				0.27	0.93	-	0.48	-					0.3					
-1	0.22	-	0.31	-																																																																				
		0.11		-2.7																																																																				
0.38	-1	-	0.22	-																																																																				
		0.12		1.5																																																																				
0.11	0.23	1	-	-																																																																				
			0.51	1.2																																																																				
0.17	-	0.31	-1	-																																																																				
	0.21			0.17																																																																				
1.42	1.43	0.27	-	0.1																																																																				
1.43	-	0.93	-	0.2																																																																				
	0.84																																																																							
0.27	0.93	-	0.48	-																																																																				
				0.3																																																																				



Продолжение таблицы А.1

№11	<table border="1"> <tbody> <tr> <td>-</td><td>-</td><td>0.11</td><td>-</td><td>0.51</td> </tr> <tr> <td>0.93</td><td>0.08</td><td></td><td>1.18</td><td></td> </tr> <tr> <td>0.18</td><td>-</td><td>0</td><td>0.21</td><td>-</td> </tr> <tr> <td></td><td>0.48</td><td></td><td></td><td>1.17</td> </tr> <tr> <td>0.13</td><td>0.31</td><td>-1</td><td>-</td><td>1.02</td> </tr> <tr> <td></td><td></td><td></td><td>0.21</td><td></td> </tr> <tr> <td>0.08</td><td>0</td><td>-</td><td>-</td><td>0.28</td> </tr> <tr> <td></td><td></td><td>0.33</td><td>0.72</td><td></td> </tr> </tbody> </table>	-	-	0.11	-	0.51	0.93	0.08		1.18		0.18	-	0	0.21	-		0.48			1.17	0.13	0.31	-1	-	1.02				0.21		0.08	0	-	-	0.28			0.33	0.72		<table border="1"> <tbody> <tr> <td>-</td><td>-</td><td>0.11</td><td>1.18</td> </tr> <tr> <td>0.93</td><td>0.08</td><td></td><td></td> </tr> <tr> <td>0.18</td><td>-</td><td>0</td><td>0.21</td> </tr> <tr> <td></td><td>0.48</td><td></td><td></td> </tr> <tr> <td>0.13</td><td>0.31</td><td>-1</td><td>0.210</td> </tr> </tbody> </table>	-	-	0.11	1.18	0.93	0.08			0.18	-	0	0.21		0.48			0.13	0.31	-1	0.210				
-	-	0.11	-	0.51																																																														
0.93	0.08		1.18																																																															
0.18	-	0	0.21	-																																																														
	0.48			1.17																																																														
0.13	0.31	-1	-	1.02																																																														
			0.21																																																															
0.08	0	-	-	0.28																																																														
		0.33	0.72																																																															
-	-	0.11	1.18																																																															
0.93	0.08																																																																	
0.18	-	0	0.21																																																															
	0.48																																																																	
0.13	0.31	-1	0.210																																																															
№12	<table border="1"> <tbody> <tr> <td>-</td><td>-</td><td>-</td><td>0.14</td><td>2.17</td> </tr> <tr> <td>0.95</td><td>0.06</td><td>0.12</td><td></td><td></td> </tr> <tr> <td>0.04</td><td>-</td><td>0.08</td><td>0.11</td><td>1.4</td> </tr> <tr> <td></td><td>1.12</td><td></td><td></td><td></td> </tr> <tr> <td>0.11</td><td>0.12</td><td>0</td><td>1.03</td><td>0.8</td> </tr> <tr> <td>0.34</td><td>0.08</td><td>-</td><td>0.14</td><td>2.1</td> </tr> <tr> <td></td><td></td><td>1.06</td><td></td><td></td> </tr> </tbody> </table>	-	-	-	0.14	2.17	0.95	0.06	0.12			0.04	-	0.08	0.11	1.4		1.12				0.11	0.12	0	1.03	0.8	0.34	0.08	-	0.14	2.1			1.06			<table border="1"> <tbody> <tr> <td>-1</td><td>-</td><td>0.21</td><td>0.92</td> </tr> <tr> <td></td><td>0.07</td><td></td><td></td> </tr> <tr> <td>1</td><td>0.03</td><td>-</td><td>0.92</td> </tr> <tr> <td></td><td></td><td>0.42</td><td></td> </tr> <tr> <td>-</td><td>1</td><td>-</td><td>1.2</td> </tr> <tr> <td>0.03</td><td></td><td>0.04</td><td></td> </tr> </tbody> </table>	-1	-	0.21	0.92		0.07			1	0.03	-	0.92			0.42		-	1	-	1.2	0.03		0.04						
-	-	-	0.14	2.17																																																														
0.95	0.06	0.12																																																																
0.04	-	0.08	0.11	1.4																																																														
	1.12																																																																	
0.11	0.12	0	1.03	0.8																																																														
0.34	0.08	-	0.14	2.1																																																														
		1.06																																																																
-1	-	0.21	0.92																																																															
	0.07																																																																	
1	0.03	-	0.92																																																															
		0.42																																																																
-	1	-	1.2																																																															
0.03		0.04																																																																
№13	<table border="1"> <tbody> <tr> <td>0</td><td>-</td><td>0.27</td><td>-</td><td>1.2</td> </tr> <tr> <td></td><td>0.19</td><td></td><td>0.88</td><td></td> </tr> <tr> <td>-</td><td>-1</td><td>-</td><td>0.21</td><td>0.92</td> </tr> <tr> <td>0.33</td><td></td><td>0.07</td><td></td><td></td> </tr> <tr> <td>0.11</td><td>0</td><td>1.03</td><td>-</td><td>0.92</td> </tr> <tr> <td></td><td></td><td></td><td>0.42</td><td></td> </tr> <tr> <td>-</td><td>-</td><td>0</td><td>-</td><td>1.2</td> </tr> <tr> <td>0.92</td><td>0.03</td><td></td><td>0.04</td><td></td> </tr> </tbody> </table>	0	-	0.27	-	1.2		0.19		0.88		-	-1	-	0.21	0.92	0.33		0.07			0.11	0	1.03	-	0.92				0.42		-	-	0	-	1.2	0.92	0.03		0.04		<table border="1"> <tbody> <tr> <td>-1</td><td>-</td><td>0.21</td><td>0.92</td> </tr> <tr> <td></td><td>0.07</td><td></td><td></td> </tr> <tr> <td>0.07</td><td>0.03</td><td>-</td><td>0.92</td> </tr> <tr> <td></td><td></td><td>0.42</td><td></td> </tr> <tr> <td>0.21</td><td>0.42</td><td>-</td><td>1.2</td> </tr> <tr> <td></td><td></td><td>0.04</td><td></td> </tr> </tbody> </table>	-1	-	0.21	0.92		0.07			0.07	0.03	-	0.92			0.42		0.21	0.42	-	1.2			0.04	
0	-	0.27	-	1.2																																																														
	0.19		0.88																																																															
-	-1	-	0.21	0.92																																																														
0.33		0.07																																																																
0.11	0	1.03	-	0.92																																																														
			0.42																																																															
-	-	0	-	1.2																																																														
0.92	0.03		0.04																																																															
-1	-	0.21	0.92																																																															
	0.07																																																																	
0.07	0.03	-	0.92																																																															
		0.42																																																																
0.21	0.42	-	1.2																																																															
		0.04																																																																
№14	<table border="1"> <tbody> <tr> <td>-</td><td>-</td><td>0.25</td><td>-</td><td>1.24</td> </tr> <tr> <td>0.88</td><td>0.23</td><td></td><td>0.16</td><td></td> </tr> <tr> <td>0.33</td><td>0.03</td><td>-</td><td>-</td><td>-</td> </tr> <tr> <td></td><td></td><td>0.84</td><td>0.32</td><td>1.15</td> </tr> <tr> <td>0.14</td><td>-</td><td>-</td><td>0.24</td><td>0.89</td> </tr> <tr> <td></td><td>0.66</td><td>0.18</td><td></td><td></td> </tr> <tr> <td>0.12</td><td>-</td><td>0</td><td>-</td><td>0.57</td> </tr> <tr> <td></td><td>0.05</td><td></td><td>0.85</td><td></td> </tr> </tbody> </table>	-	-	0.25	-	1.24	0.88	0.23		0.16		0.33	0.03	-	-	-			0.84	0.32	1.15	0.14	-	-	0.24	0.89		0.66	0.18			0.12	-	0	-	0.57		0.05		0.85		<table border="1"> <tbody> <tr> <td>0.08</td><td>-</td><td>-</td><td>0.32</td> </tr> <tr> <td></td><td>0.12</td><td>0.77</td><td></td> </tr> <tr> <td>0.25</td><td>0.22</td><td>0.14</td><td>-1</td> </tr> <tr> <td>-</td><td>-</td><td>0.06</td><td>-</td> </tr> <tr> <td>0.77</td><td>0.14</td><td></td><td>0.12</td> </tr> </tbody> </table>	0.08	-	-	0.32		0.12	0.77		0.25	0.22	0.14	-1	-	-	0.06	-	0.77	0.14		0.12				
-	-	0.25	-	1.24																																																														
0.88	0.23		0.16																																																															
0.33	0.03	-	-	-																																																														
		0.84	0.32	1.15																																																														
0.14	-	-	0.24	0.89																																																														
	0.66	0.18																																																																
0.12	-	0	-	0.57																																																														
	0.05		0.85																																																															
0.08	-	-	0.32																																																															
	0.12	0.77																																																																
0.25	0.22	0.14	-1																																																															
-	-	0.06	-																																																															
0.77	0.14		0.12																																																															
№15	<table border="1"> <tbody> <tr> <td>0.12</td><td>-1</td><td>0.32</td><td>-</td><td>0.72</td> </tr> <tr> <td></td><td></td><td></td><td>0.18</td><td></td> </tr> <tr> <td>0.08</td><td>-</td><td>-</td><td>0.32</td><td>0.58</td> </tr> <tr> <td></td><td>0.12</td><td>0.77</td><td></td><td></td> </tr> <tr> <td>0.25</td><td>0.22</td><td>0.14</td><td>-1</td><td>-</td> </tr> <tr> <td></td><td></td><td></td><td></td><td>1.56</td> </tr> <tr> <td>-</td><td>-</td><td>0.06</td><td>-</td><td>-</td> </tr> <tr> <td>0.77</td><td>0.14</td><td></td><td>0.12</td><td>1.21</td> </tr> </tbody> </table>	0.12	-1	0.32	-	0.72				0.18		0.08	-	-	0.32	0.58		0.12	0.77			0.25	0.22	0.14	-1	-					1.56	-	-	0.06	-	-	0.77	0.14		0.12	1.21	<table border="1"> <tbody> <tr> <td>0.08</td><td>0.25</td><td>-</td><td>0.32</td> </tr> <tr> <td></td><td></td><td>0.77</td><td></td> </tr> <tr> <td>0.25</td><td>0.22</td><td>0.14</td><td>-1</td> </tr> <tr> <td>-</td><td>0.14</td><td>0.06</td><td>-</td> </tr> <tr> <td>0.77</td><td></td><td></td><td>0.12</td> </tr> </tbody> </table>	0.08	0.25	-	0.32			0.77		0.25	0.22	0.14	-1	-	0.14	0.06	-	0.77			0.12				
0.12	-1	0.32	-	0.72																																																														
			0.18																																																															
0.08	-	-	0.32	0.58																																																														
	0.12	0.77																																																																
0.25	0.22	0.14	-1	-																																																														
				1.56																																																														
-	-	0.06	-	-																																																														
0.77	0.14		0.12	1.21																																																														
0.08	0.25	-	0.32																																																															
		0.77																																																																
0.25	0.22	0.14	-1																																																															
-	0.14	0.06	-																																																															
0.77			0.12																																																															

Продолжение таблицы А.1

№16	<table border="1"> <tbody> <tr> <td>-</td><td>0.23</td><td>0.18</td><td>0.17</td><td>1.42</td></tr> <tr> <td>0.86</td><td></td><td></td><td></td><td></td></tr> <tr> <td>0.12</td><td>-</td><td>0.08</td><td>0.09</td><td>0.83</td></tr> <tr> <td></td><td>1.14</td><td></td><td></td><td></td></tr> <tr> <td>0.16</td><td>0.24</td><td>-1</td><td>-</td><td>-</td></tr> <tr> <td></td><td></td><td></td><td>0.35</td><td>1.21</td></tr> <tr> <td>0.23</td><td>-</td><td>0.05</td><td>-</td><td>-</td></tr> <tr> <td></td><td>0.08</td><td></td><td>0.75</td><td>0.65</td></tr> </tbody> </table>	-	0.23	0.18	0.17	1.42	0.86					0.12	-	0.08	0.09	0.83		1.14				0.16	0.24	-1	-	-				0.35	1.21	0.23	-	0.05	-	-		0.08		0.75	0.65	<table border="1"> <tbody> <tr> <td>1</td><td>2</td><td>3</td><td>0.55</td></tr> <tr> <td>2</td><td>4</td><td>9</td><td>0.35</td></tr> <tr> <td>3</td><td>9</td><td>5</td><td>0.55</td></tr> </tbody> </table>	1	2	3	0.55	2	4	9	0.35	3	9	5	0.55
-	0.23	0.18	0.17	1.42																																																		
0.86																																																						
0.12	-	0.08	0.09	0.83																																																		
	1.14																																																					
0.16	0.24	-1	-	-																																																		
			0.35	1.21																																																		
0.23	-	0.05	-	-																																																		
	0.08		0.75	0.65																																																		
1	2	3	0.55																																																			
2	4	9	0.35																																																			
3	9	5	0.55																																																			
№17	<table border="1"> <tbody> <tr> <td>76</td><td>21</td><td>6</td><td>-34</td><td>-142</td></tr> <tr> <td>12</td><td>-114</td><td>8</td><td>9</td><td>83</td></tr> <tr> <td>16</td><td>24</td><td>-100</td><td>-35</td><td>-121</td></tr> <tr> <td>23</td><td>-8</td><td>5</td><td>-75</td><td>85</td></tr> </tbody> </table>	76	21	6	-34	-142	12	-114	8	9	83	16	24	-100	-35	-121	23	-8	5	-75	85	<table border="1"> <tbody> <tr> <td>1</td><td>2</td><td>4</td><td>0.55</td></tr> <tr> <td>1</td><td>4</td><td>16</td><td>1.35</td></tr> <tr> <td>1</td><td>8</td><td>64</td><td>3.55</td></tr> </tbody> </table>	1	2	4	0.55	1	4	16	1.35	1	8	64	3.55																				
76	21	6	-34	-142																																																		
12	-114	8	9	83																																																		
16	24	-100	-35	-121																																																		
23	-8	5	-75	85																																																		
1	2	4	0.55																																																			
1	4	16	1.35																																																			
1	8	64	3.55																																																			
№18	<table border="1"> <tbody> <tr> <td>-83</td><td>27</td><td>-13</td><td>-11</td><td>142</td></tr> <tr> <td>5</td><td>-68</td><td>13</td><td>24</td><td>26</td></tr> <tr> <td>9</td><td>54</td><td>127</td><td>36</td><td>23</td></tr> <tr> <td>13</td><td>27</td><td>34</td><td>156</td><td>49</td></tr> </tbody> </table>	-83	27	-13	-11	142	5	-68	13	24	26	9	54	127	36	23	13	27	34	156	49	<table border="1"> <tbody> <tr> <td>0.64</td><td>0.53</td><td>-</td><td>3</td></tr> <tr> <td></td><td></td><td>0.33</td><td></td></tr> <tr> <td>0.53</td><td>-</td><td>0.23</td><td>2</td></tr> <tr> <td></td><td>0.92</td><td></td><td></td></tr> <tr> <td>-</td><td>0.23</td><td>0.78</td><td>1</td></tr> <tr> <td>0.33</td><td></td><td></td><td></td></tr> </tbody> </table>	0.64	0.53	-	3			0.33		0.53	-	0.23	2		0.92			-	0.23	0.78	1	0.33											
-83	27	-13	-11	142																																																		
5	-68	13	24	26																																																		
9	54	127	36	23																																																		
13	27	34	156	49																																																		
0.64	0.53	-	3																																																			
		0.33																																																				
0.53	-	0.23	2																																																			
	0.92																																																					
-	0.23	0.78	1																																																			
0.33																																																						
№19	<table border="1"> <tbody> <tr> <td>1</td><td>2</td><td>3</td><td>9</td><td>1.11</td></tr> <tr> <td>2</td><td>1</td><td>9</td><td>4</td><td>1.16</td></tr> <tr> <td>3</td><td>9</td><td>1</td><td>4</td><td>1.24</td></tr> <tr> <td>9</td><td>1</td><td>3</td><td>4</td><td>1.55</td></tr> </tbody> </table>	1	2	3	9	1.11	2	1	9	4	1.16	3	9	1	4	1.24	9	1	3	4	1.55	<table border="1"> <tbody> <tr> <td>5</td><td>3</td><td>1</td><td>11</td></tr> <tr> <td>3</td><td>5</td><td>3</td><td>17</td></tr> <tr> <td>1</td><td>3</td><td>5</td><td>19</td></tr> </tbody> </table>	5	3	1	11	3	5	3	17	1	3	5	19																				
1	2	3	9	1.11																																																		
2	1	9	4	1.16																																																		
3	9	1	4	1.24																																																		
9	1	3	4	1.55																																																		
5	3	1	11																																																			
3	5	3	17																																																			
1	3	5	19																																																			
№20	<table border="1"> <tbody> <tr> <td>-1</td><td>0.28</td><td>-</td><td>0.06</td><td>-21</td></tr> <tr> <td></td><td></td><td>0.17</td><td></td><td></td></tr> <tr> <td>0.52</td><td>-1</td><td>0.12</td><td>0.17</td><td>117</td></tr> <tr> <td>0.17</td><td>-</td><td>-</td><td>0</td><td>0.81</td></tr> <tr> <td></td><td>0.18</td><td>0.79</td><td></td><td></td></tr> <tr> <td>0.11</td><td>0.22</td><td>0.03</td><td>-</td><td>-</td></tr> <tr> <td></td><td></td><td></td><td>0.95</td><td>0.72</td></tr> </tbody> </table>	-1	0.28	-	0.06	-21			0.17			0.52	-1	0.12	0.17	117	0.17	-	-	0	0.81		0.18	0.79			0.11	0.22	0.03	-	-				0.95	0.72	<table border="1"> <tbody> <tr> <td>11</td><td>12</td><td>13</td><td>13</td></tr> <tr> <td>12</td><td>13</td><td>15</td><td>4</td></tr> <tr> <td>13</td><td>15</td><td>19</td><td>17</td></tr> </tbody> </table>	11	12	13	13	12	13	15	4	13	15	19	17					
-1	0.28	-	0.06	-21																																																		
		0.17																																																				
0.52	-1	0.12	0.17	117																																																		
0.17	-	-	0	0.81																																																		
	0.18	0.79																																																				
0.11	0.22	0.03	-	-																																																		
			0.95	0.72																																																		
11	12	13	13																																																			
12	13	15	4																																																			
13	15	19	17																																																			
№21	<table border="1"> <tbody> <tr> <td>76</td><td>21</td><td>6</td><td>-34</td><td>142</td></tr> <tr> <td>12</td><td>-114</td><td>8</td><td>9</td><td>83</td></tr> <tr> <td>16</td><td>24</td><td>-100</td><td>35</td><td>121</td></tr> <tr> <td>23</td><td>-8</td><td>5</td><td>-75</td><td>85</td></tr> </tbody> </table>	76	21	6	-34	142	12	-114	8	9	83	16	24	-100	35	121	23	-8	5	-75	85	<table border="1"> <tbody> <tr> <td>1</td><td>2</td><td>4</td><td>0.55</td></tr> <tr> <td>2</td><td>4</td><td>7</td><td>1.35</td></tr> <tr> <td>3</td><td>6</td><td>14</td><td>3.55</td></tr> </tbody> </table>	1	2	4	0.55	2	4	7	1.35	3	6	14	3.55																				
76	21	6	-34	142																																																		
12	-114	8	9	83																																																		
16	24	-100	35	121																																																		
23	-8	5	-75	85																																																		
1	2	4	0.55																																																			
2	4	7	1.35																																																			
3	6	14	3.55																																																			

Продолжение таблицы А.1

№22	<table border="1"> <tbody> <tr> <td>-83</td> <td>27</td> <td>-13</td> <td>-11</td> <td>142</td> </tr> <tr> <td>5</td> <td>-68</td> <td>13</td> <td>24</td> <td>26</td> </tr> <tr> <td>9</td> <td>54</td> <td>127</td> <td>36</td> <td>23</td> </tr> <tr> <td>13</td> <td>27</td> <td>34</td> <td>156</td> <td>49</td> </tr> </tbody> </table>	-83	27	-13	-11	142	5	-68	13	24	26	9	54	127	36	23	13	27	34	156	49	<table border="1"> <tbody> <tr> <td>1.64</td> <td>0.53</td> <td>-</td> <td>3</td> </tr> <tr> <td>0.53</td> <td>-</td> <td>0.33</td> <td>2</td> </tr> <tr> <td>-</td> <td>0.23</td> <td>1.78</td> <td>1</td> </tr> </tbody> </table>	1.64	0.53	-	3	0.53	-	0.33	2	-	0.23	1.78	1										
-83	27	-13	-11	142																																								
5	-68	13	24	26																																								
9	54	127	36	23																																								
13	27	34	156	49																																								
1.64	0.53	-	3																																									
0.53	-	0.33	2																																									
-	0.23	1.78	1																																									
№23	<table border="1"> <tbody> <tr> <td>25</td> <td>3</td> <td>5</td> <td>4</td> <td>1.11</td> </tr> <tr> <td>5</td> <td>4</td> <td>3</td> <td>25</td> <td>1.16</td> </tr> <tr> <td>3</td> <td>25</td> <td>4</td> <td>5</td> <td>1.24</td> </tr> <tr> <td>4</td> <td>5</td> <td>25</td> <td>3</td> <td>1.55</td> </tr> </tbody> </table>	25	3	5	4	1.11	5	4	3	25	1.16	3	25	4	5	1.24	4	5	25	3	1.55	<table border="1"> <tbody> <tr> <td>15</td> <td>13</td> <td>11</td> <td>11</td> </tr> <tr> <td>13</td> <td>15</td> <td>13</td> <td>17</td> </tr> <tr> <td>11</td> <td>13</td> <td>15</td> <td>19</td> </tr> </tbody> </table>	15	13	11	11	13	15	13	17	11	13	15	19										
25	3	5	4	1.11																																								
5	4	3	25	1.16																																								
3	25	4	5	1.24																																								
4	5	25	3	1.55																																								
15	13	11	11																																									
13	15	13	17																																									
11	13	15	19																																									
№24	<table border="1"> <tbody> <tr> <td>0.12</td> <td>-1</td> <td>0.32</td> <td>-</td> <td>0.72</td> </tr> <tr> <td>0.08</td> <td>-</td> <td>-</td> <td>0.32</td> <td>0.58</td> </tr> <tr> <td>0.25</td> <td>0.02</td> <td>0.14</td> <td>-1</td> <td>-</td> </tr> <tr> <td>-</td> <td>-</td> <td>0.06</td> <td>-</td> <td>-</td> </tr> <tr> <td>0.77</td> <td>0.14</td> <td>0.12</td> <td></td> <td>1.21</td> </tr> </tbody> </table>	0.12	-1	0.32	-	0.72	0.08	-	-	0.32	0.58	0.25	0.02	0.14	-1	-	-	-	0.06	-	-	0.77	0.14	0.12		1.21	<table border="1"> <tbody> <tr> <td>0.08</td> <td>0.25</td> <td>-</td> <td>0.32</td> </tr> <tr> <td>0.25</td> <td>0.22</td> <td>0.14</td> <td>-1</td> </tr> <tr> <td>-</td> <td>0.14</td> <td>0.06</td> <td>-</td> </tr> <tr> <td>1.77</td> <td></td> <td></td> <td>0.12</td> </tr> </tbody> </table>	0.08	0.25	-	0.32	0.25	0.22	0.14	-1	-	0.14	0.06	-	1.77			0.12	
0.12	-1	0.32	-	0.72																																								
0.08	-	-	0.32	0.58																																								
0.25	0.02	0.14	-1	-																																								
-	-	0.06	-	-																																								
0.77	0.14	0.12		1.21																																								
0.08	0.25	-	0.32																																									
0.25	0.22	0.14	-1																																									
-	0.14	0.06	-																																									
1.77			0.12																																									
№25	<table border="1"> <tbody> <tr> <td>-</td> <td>0.23</td> <td>0.18</td> <td>0.17</td> <td>1.42</td> </tr> <tr> <td>0.86</td> <td>-</td> <td>0.08</td> <td>0.09</td> <td>0.83</td> </tr> <tr> <td>0.12</td> <td>1.14</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>0.16</td> <td>0.24</td> <td>-1</td> <td>0.35</td> <td>1.21</td> </tr> <tr> <td>0.23</td> <td>-</td> <td>0.05</td> <td>-</td> <td>-</td> </tr> <tr> <td>0.08</td> <td>0.75</td> <td></td> <td></td> <td>0.65</td> </tr> </tbody> </table>	-	0.23	0.18	0.17	1.42	0.86	-	0.08	0.09	0.83	0.12	1.14	-	-	-	0.16	0.24	-1	0.35	1.21	0.23	-	0.05	-	-	0.08	0.75			0.65	<table border="1"> <tbody> <tr> <td>11</td> <td>21</td> <td>31</td> <td>0.55</td> </tr> <tr> <td>21</td> <td>41</td> <td>91</td> <td>0.35</td> </tr> <tr> <td>31</td> <td>91</td> <td>51</td> <td>0.55</td> </tr> </tbody> </table>	11	21	31	0.55	21	41	91	0.35	31	91	51	0.55
-	0.23	0.18	0.17	1.42																																								
0.86	-	0.08	0.09	0.83																																								
0.12	1.14	-	-	-																																								
0.16	0.24	-1	0.35	1.21																																								
0.23	-	0.05	-	-																																								
0.08	0.75			0.65																																								
11	21	31	0.55																																									
21	41	91	0.35																																									
31	91	51	0.55																																									

## Приложение Б (обязательное)

### Лабораторная работа 2 Решение нелинейных уравнений

Задание 1. Построить графики функций. Выполните отделение корней. Сравните объем вычислений при использовании различных методов.

Задание 2. Выполните численное интегрирование, используя в качестве подинтегральных функций, функции из вариантов заданий.

Таблица Б.1 - Варианты заданий

№	1	2	3
1	$3x^4+4x^3-12x^2-5=0$	$\ln(x)+(x+1)^3=0$	$\sin(x+1)-y=1.2$ $2x+\cos(x)=2$
2	$2x^3-9x^2-60x+1=0$	$x \cdot 2^x=1$	$\operatorname{tg}(xy+0.4)=x^2$ $0.6x^2+2y^2=1$
3	$x^4-x-1=0$	$x+\cos(x)=1$	$\cos(x-1)+y=0.5$ $x-\cos(x)=3$
4	$2x^4-x^2-10=0$	$x+\lg(1+x)=1.5$	$\sin(x)+2y=2$ $\cos(y-1)+x=0.7$
5	$3x^4+8x^3+6x^2-10=0$	$\lg(2+x)+2x=3$	$\cos(x-1)+y=1$ $\sin(y)+2x=1.6$
6	$x^4-18x^2+5x-8=0$	$2^x+5x-3=0$	$\sin(x+1)-y=1$ $2x+\cos(y)=2$
7	$x^4+4x^3-12x^2+1=0$	$5^x+3x=0$	$\sin(x-y)-xy=0$ $x^2-y^2=0.75$
8	$x^4-x^3-2x^2+3x-3=0$	$3e^x=5x+2$	$\sin(x+y)-1.5xy=0$ $x^2+y^2=1$
9	$3x^4+4x^3-12x^2+1=0$	$5^x=6x+3$	$\sin(x-y)-xy+1=0$ $x^2-y^2=0.75$
10	$3x^4-8x^3-18x^2+2=0$	$2e^x+5x-6=0$	$y=1/(x^{3/2}+1)$ $x^2+y^2=9$
11	$2x^4-8x^3+8x^2-1=0$	$2\operatorname{arctg}(x)-x+3=0$	$x^2+y^2=9$ $y=1+e^{-x}$
12	$2x^4+8x^3+8x^2-1=0$	$(x-3) \cdot \cos(x)=1$	$x^2+y^2=5$ $y=1-2e^{-xy}$
13	$x^4-4x^3-8x^2+1=0$	$x^x=20-9x$	$x^2+y^2=5$ $y=e^{-xy}$
14	$2x^4-9x^3-60x^2+1=0$	$x \cdot \lg(x)=1$	$\sin(x-0.6)-y=1.6$ $3x-\cos(y)=0.9$

Продолжение таблицы Б.1

№	1	2	3
15	$x^5 + x^2 - 5 = 0$	$\operatorname{tg}^3 x = x - 1$	$x^2 + y^2 = 6$ $y = e^{-x}$
16	$3x^4 + 4x^3 - 12x^2 - 7 = 0$	$5^x = 1 + e^{-x}$	$x^3 + y^3 = 6$ $y = e^{-x}$
17	$3x^4 + 8x^3 + 6x^2 - 11 = 0$	$5^x = 3 - e^x$	$x^4 + y^4 = 5$ $y = e^{-x}$
18	$x^4 - 18x^3 - 10 = 0$	$\operatorname{arctg}(x^2 + 1/x) = x$	$x^2 + y^2 = 1$ $\sin(x+y) = 1.2x$
19	$3x^4 - 8x^3 - 18x^2 + 2 = 0$	$\operatorname{tg}(0.55x + 0.1) = x^2$	$x^2 + y^2 = 1$ $\sin(x+y) = 0.2 + x$
20	$x^4 - 18x - 10 = 0$	$5^x - 6x = 7$	$x + \cos(y-1) = 0.8$ $y - \cos(x) = 2$
21	$x^4 + 18x - 10 = 0$	$5^x - 6x = 3$	$x^2 + y^2 = 1$ $x^3 + y^3 = 2$
22	$x^4 + 18x^3 - 6x^2 + x - 10 = 0$	$5^x = 1 + e^{-2x}$	$x^2 + y^2 = 1$ $x - y^3 = 0.5$
23	$x^5 + 12x^3 - 6x^2 + x - 10 = 0$	$7^x - 6x = 2$	$x^3 + y^3 = 8$ $y = x^{3/2}$
24	$3x^5 - 8x^3 - 18x^2 + 2 = 0$	$5^x = 2 + e^{-2x}$	$x^3 + y^3 = 8$ $y = 1 + x^{3/2}$
25	$x^3 - 18x - 10 = 0$	$x \cdot 2^x = 3$	$x^3 + y^3 = 8$ $y = 1 - x^{3/2}$

## Приложение В (обязательное)

### Лабораторная работа 3 Аппроксимация функций

Задание 1. Выберите из таблицы вариантов 11 значений функции  $f(x)$ , начиная с узла, равного номеру вашего варианта. Постройте таблицы конечных разностей. Выполните экстраполяцию на два узла от начала и от конца таблицы.

Задание 2. Для выбранной таблицы выполните квадратичную сплайн-интерполяцию (по шести узлам). Проконтролируйте полученные оценки для промежуточных узлов.

Задание 3. Считая выбранную таблицу заданной для диапазона от 0 до  $2\pi$ , выполните среднеквадратическую аппроксимацию тригонометрическим многочленом (отрезком ряда Фурье) третьей степени.

Задание 4. Для выбранной таблицы выполните аппроксимации алгебраическими многочленами различной степени и оцените их качество по отношению остаточного и исходного среднеквадратического отклонений.

Таблица В.1 - Варианты заданий

x	y(x)	x	y(x)	x	y(x)	x	y(x)
1,0	0	4,0	0,000196	7,0	0,00018	10,0	0,000108
1,1	0,324097	4,1	-5,19505	7,1	0,856485	10,1	-1,02845
1,2	0,643881	4,2	-10,3689	7,2	1,640842	10,2	-1,96638
1,3	0,922415	4,3	-14,959	7,3	2,27459	10,3	-2,72032
1,4	1,1253	4,4	-18,4126	7,4	2,692863	10,4	-3,21408
1,5	1,224745	4,5	-20,25	7,5	2,851227	10,5	-3,3964
1,6	1,20301	4,6	-20,1243	7,6	2,730379	10,6	-3,24618
1,7	1,054847	4,7	-17,8711	7,7	2,338403	10,7	-2,77495
1,8	0,788625	4,8	-13,5425	7,8	1,710348	10,8	-2,02598
1,9	0,425989	4,9	-7,41942	7,9	0,905108	10,9	-1,07035
2,0	$4,62 \cdot 10^{-5}$	5,0	0	8,0	$-9,6 \cdot 10^{-5}$	11,0	-0,00023
2,1	-0,44776	5,1	8,037451	8,1	-0,91714	11,1	1,080087
2,2	-0,87178	5,2	15,89357	8,2	-1,75557	11,2	2,064282
2,3	-1,2269	5,3	22,72513	8,3	-2,43156	11,3	2,854531
2,4	-1,47335	5,4	27,73269	8,4	-2,8763	11,4	3,37121
2,5	-1,58114	5,5	30,25	8,5	-3,04297	11,5	3,560925
2,6	-1,53356	5,6	29,82532	8,6	-2,91168	11,6	3,402017
2,7	-1,3294	5,7	26,2854	8,7	-2,49175	11,7	2,90698
2,8	-0,98363	5,8	19,77381	8,8	-1,82115	11,8	2,121544

Продолжение таблицы В.1

x	y(x)	x	y(x)	x	y(x)	x	y(x)
2,9	-0,52634	5,9	10,75785	8,9	-0,96308	11,9	1,120452
3,0	-0,00011	6,0	0,001176	9,0	-10 <sup>-13</sup>	12,0	0,000357
3,1	0,543966	6,1	-11,4973	9,1	0,97427	12,1	-1,12952
3,2	1,051358	6,2	-22,5932	9,2	1,863736	12,2	-2,15806
3,3	1,469572	6,3	-32,1089	9,3	2,579679	12,3	-2,98314
3,4	1,753617	6,4	-38,9547	9,4	3,049516	12,4	-3,52184
3,5	1,870829	6,5	-42,25	9,5	3,224158	12,5	-3,71872
3,6	1,804553	6,6	-41,4287	9,6	3,083118	12,6	-3,55153
3,7	1,556275	6,7	-36,3182	9,7	2,636854	12,7	-3,03371
3,8	1,145949	6,8	-27,1814	9,8	1,926069	12,8	-2,21331
3,9	0,610438	6,9	-14,7151	9,9	1,01801	12,9	-1,16858

Таблица В.2

x	y(x)	x	y(x)	x	y(x)	x	y(x)
13,0	-0,00055	17,0	-0,00128	21,0	-0,00133	25,0	-0,00055
13,1	0,447264	17,1	0,761981	21,1	0,643412	25,1	0,328549
13,2	0,871348	17,2	1,462508	21,2	1,250753	25,2	0,627288
13,3	1,226577	17,3	2,029831	21,3	1,757043	25,3	0,866148
13,4	1,473176	17,4	2,405585	21,4	2,106558	25,4	1,021332
13,5	1,581139	17,5	2,549509	21,5	2,257585	25,5	1,077122
13,6	1,533737	17,6	2,443738	21,6	2,187247	25,6	1,027476
13,7	1,329751	17,7	2,094918	21,7	1,894532	25,7	0,876673
13,8	0,984119	17,8	1,533913	21,8	1,401147	25,8	0,638952
13,9	0,526919	17,9	0,8131	21,9	0,750042	25,9	0,337172
14,0	0,000735	18,0	0	22,0	0,001688	26,0	-0,00055
14,1	-0,54336	18,1	-0,06906	22,1	-0,77156	26,1	0,328549
14,2	-1,05084	18,2	-0,20633	22,2	-1,49251	26,2	0,627288
14,3	-1,46919	18,3	-0,36975	22,3	-2,08686	26,3	0,866148
14,4	-1,75341	18,4	-0,52416	22,4	-2,49087	26,4	1,021332
14,5	-1,87083	18,5	-0,63728	22,5	-2,6582	26,5	1,077122
14,6	-1,80476	18,6	-0,68247	22,6	-2,56506	26,6	1,027476
14,7	-1,55668	18,7	-0,64188	22,7	-2,21332	26,7	0,876673
14,8	-1,14651	18,8	-0,50883	22,8	-1,63099	26,8	0,638952
14,9	-0,61111	18,9	-0,28908	22,9	-0,87009	26,9	0,337172

Продолжение таблицы В.2

x	y(x)	x	y(x)	x	y(x)	x	y(x)
15,0	-0,00091	19,0	-0,00059	23,0	-0,00046	27,0	0,000606
15,1	0,624825	19,1	0,328168	23,1	0,309544	27,1	-0,33789
15,2	1,203832	19,2	0,661197	23,2	0,591083	27,2	-0,64508
15,3	1,677044	19,3	0,95903	23,3	0,816324	27,3	-0,89064
15,4	1,994648	19,4	1,183327	23,4	0,962789	27,4	-1,05011
15,5	2,12132	19,5	1,301545	23,5	1,015605	27,5	-1,10737
15,6	2,040105	19,6	1,29113	23,6	0,969005	27,6	-1,05623
15,7	1,754519	19,7	1,142726	23,7	0,826959	27,7	-0,90112
15,8	1,288629	19,8	0,86201	23,8	0,602835	27,8	-0,65672
15,9	0,685062	19,9	0,46989	23,9	0,318152	27,9	-0,34653
16,0	0,001095	20,0	0,000974	24,0	0,000505	28,0	-0,00066
16,1	-0,6968	20,1	-0,49956	24,1	-0,3191	28,1	0,347125
16,2	-1,33944	20,2	-0,98043	24,2	-0,60929	28,2	0,662688
16,3	-1,86182	20,3	-1,38957	24,3	-0,84138	28,3	0,914876
16,4	-2,20969	20,4	-1,67979	24,4	-0,99223	28,4	1,078594
16,5	-2,34521	20,5	-1,8141	24,5	-1,04654	28,5	1,137301
16,6	-2,25098	20,6	-1,77023	24,6	-0,99841	28,6	1,084682
16,7	-1,93222	20,7	-1,54364	24,7	-0,85196	28,7	0,925319
16,8	-1,41658	20,8	-1,14883	24,8	-0,62099	28,8	0,674301
16,9	-0,7518	20,9	-0,6186	24,9	-0,32771	28,9	0,355793



# Приложение Г (обязательное)

## Лабораторная работа 4 Численное интегрирование

---


Задание:

1 Найдите оценки по формулам:


- прямоугольников (с центральным узлом)

 Формула прямоугольников

- трапеций

 Формула трапеций

























- Симпсона

 Формула Симпсона

























2 Найти точное его значение по формуле Ньютона-Лейбница

 Формула Ньютона-Лейбница













Таблица Г.1 - Варианты заданий

1	 Интеграл	 Интеграл
2	 Интеграл	 Интеграл
3	 Интеграл	 Интеграл
4	 Интеграл	 Интеграл
5	 Интеграл	 Интеграл
6	 Интеграл	 Интеграл
7	 Интеграл	 Интеграл
8	 Интеграл	 Интеграл
9	 Интеграл	 Интеграл
10	 Интеграл	 Интеграл
11	 Интеграл	 Интеграл
12	 Интеграл	 Интеграл

Продолжение таблицы Г.1

13	 Инт...	 Интеграл
14	 Интеграл	 Интеграл
15	 Интеграл	 Интеграл
16	 Интеграл	 Интеграл
17	 Интеграл	 Интеграл
18	 Интеграл	 Интеграл
19	 Интеграл	 Интеграл
20	 Интеграл	 Интеграл
21	 Интеграл	 Интеграл
22	 Интеграл	 Интеграл
23	 Интеграл	 Интеграл
24	 Интеграл	 Интеграл

Продолжение таблицы Г.1

25	 Интеграл	 Интеграл
26	 Интеграл	 Интеграл
27	 Интеграл	 Интеграл
28	 Интеграл	 Интеграл
29	 Интеграл	 Интеграл
30	 Интеграл	 Интеграл

## **Список использованных источников**

- 1 **Дьяконов, В.П.** MATLAB: учебный курс/ В.П. Дьяконов.- СПб.: Питер, 2001.-592с.
- 2 **Дьяконов, В.П.** Математические пакеты расширения MATLAB: спец. справочник / В.П. Дьяконов, В. Круглов. – СПб.: Питер, 2001.480с.
- 3 **Потемкин, В.Г.** MATLAB 6: среда проектирования инженерных приложений / В.Г. Потемкин. – М.: ДИАЛОГ-МИФИ, 2003.-448с.
- 4 **Гультяев, А.** Визуальное моделирование в среде MATLAB: учебный курс / А. Гультяев. - СПб.: Питер, 2000.-448с.
- 5 **Мартынов, Н.Н.** MATLAB 5.x Вычисления, визуализация, программирование / Н.Н. Мартынов. - М.: КУДИЦ-ОБРАЗ, 2000.-336с.