

## Введение

При решении задач анализа электрических цепей дисциплин «Основы компьютерного анализа электрических цепей» и «Теоретические основы электротехники» требуются знания и умения, чтобы владеть современным математическим аппаратом и программными средствами.

Для численного и аналитического решения указанных выше задач имеется несколько универсальных программных средств. Это современные пакеты прикладных программ, такие как Maple [1], Mathematica [2], Scilab [3], Matlab [4–7] и некоторые другие, которые являются мощными программными системами, включающими в себя специализированные системные и языковые средства, рассчитанные на применение в различных областях науки и техники.

Каждый их перечисленных выше математических пакетов имеет особенности в своей архитектуре или структуре. Тем не менее для современных универсальных математических пакетов можно представить общую структуру, которая состоит из следующих элементов [8]:

- ядро системы — это программные коды множества заранее откомпилированных функций и процедур, обеспечивающих достаточно представительный набор встроенных функций и операторов системы;
- интерфейс, позволяющий пользователю обращаться к ядру со своими запросами и получать результат решения на экране дисплея;
- библиотеки функций и процедур, включенные в ядро, значительно ускоряющие вычислительный процесс;
- пакеты расширения, позволяющие осуществлять расширение возможностей систем и их адаптацию к решаемым конкретными пользователями задачам;
- справочная система, аккумулирующая сведения о системе и ее возможностях.

Математический пакет Matlab, некоторые возможности которого будут рассмотрены в данном пособии, предназначен как для численных вычислений, так и для применения в сфере символической математики и моделирования. Matlab — это одна из старейших, тщательно проработанных и проверенных временем систем ав-

томатизации математических расчетов, построенная на расширенном представлении и применении матричных операций. Синтаксис языка программирования системы продуман настолько тщательно, что данная ориентация почти не ощущается теми пользователями, которых не интересуют непосредственно матричные вычисления.

Несмотря на то что изначально Matlab предназначался исключительно для вычислений, в процессе эволюции (а сейчас выпущена уже версия 2020a) в дополнение к численным средствам в Matlab был разработан пакет символьных вычислений (Symbolic Math Toolbox [9]), а также появились библиотеки, которые обеспечивают в Matlab уникальные для математических пакетов символьные функции.

Все библиотеки Matlab отличаются высокой скоростью численных вычислений. Здесь матрицы широко применяются не только в таких математических расчетах, как решение задач линейной алгебры и математического моделирования, обчета статических и динамических систем и объектов. Они являются основой автоматического составления и решения уравнений состояния динамических объектов и систем. Именно универсальность аппарата матричного исчисления значительно повышает интерес к системе Matlab, вобравшей в себя лучшие достижения в области быстрого решения матричных задач. Поэтому Matlab давно уже вышла за рамки специализированной матричной системы, превратившись в одну из наиболее мощных универсальных интегрированных систем компьютерной математики.

Отличие символьных вычислений, называемых также аналитическими, от численных заключается в том, что исходные данные на вычисление задаются в виде аналитических выражений (т. е. представленных в виде математических выражений) и результаты вычислений также получаются в символьном (аналитическом) виде. Причем численные результаты являются частными случаями результатов символьных вычислений.

Средства пакета Matlab — Symbolic Math Toolbox (в дальнейшем просто Symbolic) дают Matlab качественно новые возможности по отношению численных вычислений, связанные с выполнением символьных вычислений и преобразований, и, в частности, упрощают процесс исследования объектов и понимание процессов, протекающих в них. Причем Symbolic при работе с числовыми значениями может использовать либо символьную, либо числовую арифметику, при этом числовая арифметика — это либо арифметика с переменной точностью, либо арифметика с плавающей точкой двойной точности.

Для визуализации результатов анализа решения различных задач система Matlab имеет широкий спектр функций, поддерживающих визуализацию проводимых вычислений непосредственно из среды Matlab, что улучшает возможности построения алгоритмов обработки изображений. Усовершенствованные методы графической библиотеки в совокупности с языком программирования Matlab обеспечивают открытую расширяемую систему, которая может быть использована для создания специальных приложений, пригодных для обработки графики.

Исходя из своей универсальности, система Matlab может использоваться практически во всех технических дисциплинах вузов как для проведения сложных математических расчетов, так и для решения задач математического моделирования процессов и систем, в частности система Matlab давно используется для решения проблем математического моделирования в электрических схемах.

Учебное пособие состоит из двух разделов. В первом разделе рассмотрены рабочая среда Matlab и ее компоненты, основные понятия, среда и программные средства символьных вычислений, возможности визуализации символьных вычислений и математические программные средства для решения задач теории электрических цепей. Второй раздел пособия посвящен решению задач анализа линейных цепей средствами Symbolic, основанных на законах и методах анализа цепей, режимов гармонических колебаний и негармонических воздействий и нахождении частотных характеристик цепей.

Материал данного учебного пособия соответствует стандарту подготовки бакалавров по направлению 11.02.03 «Инфокоммуникационные технологии и системы связи» и может быть использован для дневной, заочной и дистанционной форм обучения по дисциплине «Основы компьютерного анализа электрических цепей».

# I Средства аналитических вычислений пакета Symbolic в среде Matlab

---

---

## 1 Элементы рабочей среды Matlab

---

### 1.1. Рабочая среда Matlab и ее компоненты

Рабочая среда пакета Matlab [8] появляется на экране дисплея сразу после его запуска. В стандартной конфигурации графический интерфейс пользователя, аналогичен интерфейсу Windows-приложений с ленточной структурой (рис. 1.1).

Таким образом, система Matlab сразу после запуска готова к проведению вычислений в командном режиме. Если же рабочая среда системы Matlab находится не в стандартной конфигурации, то для перехода в стандартную конфигурацию необходимо щелкнуть мышкой по элементу **Layout** (Макет) и в открывшемся окне **SELECT LAYOUT** (рис. 1.2) выполнить команду **Default**, которая возвращает конфигурацию рабочей среды, принятую по умолчанию. Поскольку рабочая среда системы Matlab — это обычное окно приложений MS Windows, то его также можно перемещать, изменять в размерах и открывать на весь экран.

В рабочей среде стандартной конфигурации размещены различные компоненты Matlab. На рис. 1.1 все видимые компоненты снабжены соответствующими выносками и пронумерованы. Причем следует отметить, что в данном разделе описаны только компоненты с номерами от 1 до 10, а компоненты с номерами от 11 до 13 будут показаны и описаны ниже в соответствующих им разделах.

Рабочая среда Matlab имеет следующие основные компоненты [9].

Стандартное меню (рис. 1.3), позволяющее выполнить команды: **Установить путь**, **Открыть редактор**, **Сохранить**, **Вырезать**, **Скопировать**, **Вставить**, **Отменить**, **Вызвать справку** и многое другие. Кроме перечисленных команд, имеется команда, вызывающая меню, элементы которого позволяют выбирать активные

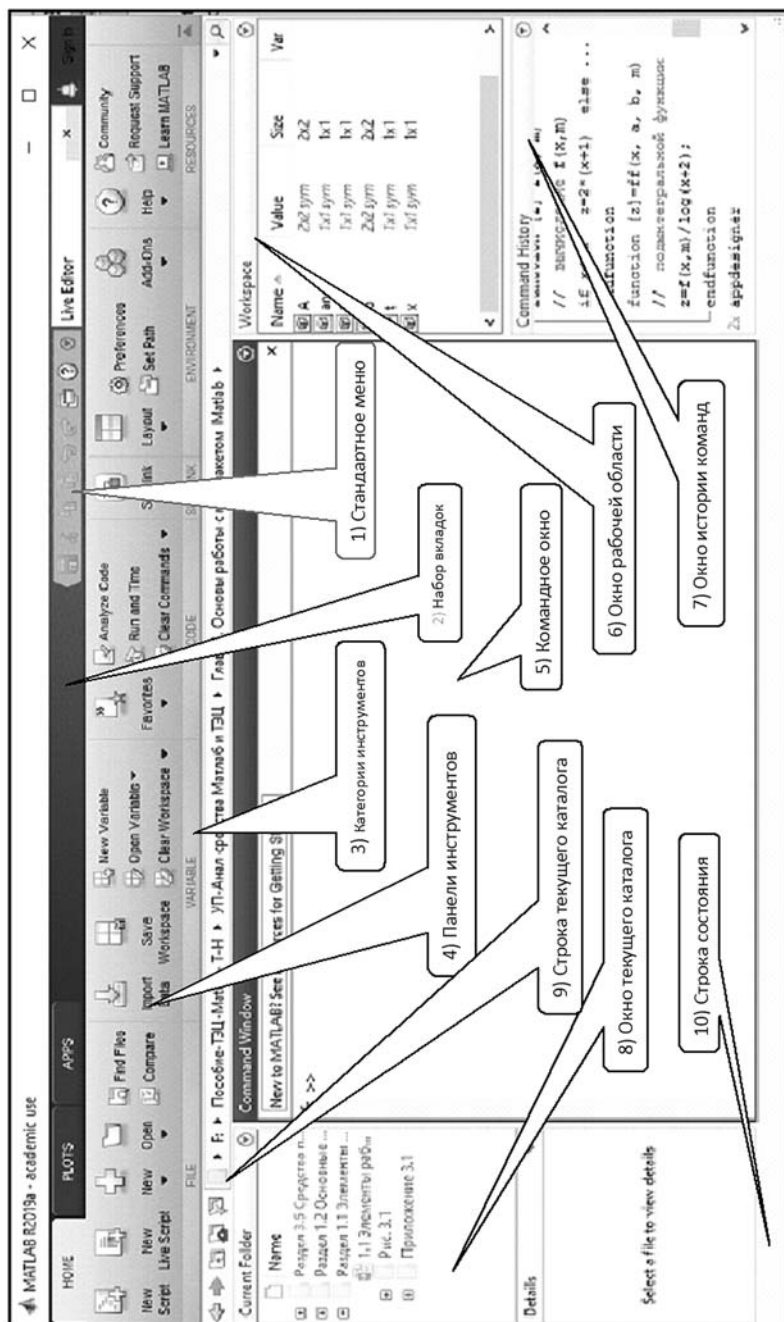


Рис. 1.1. Стандартная конфигурация графического интерфейса пользователя

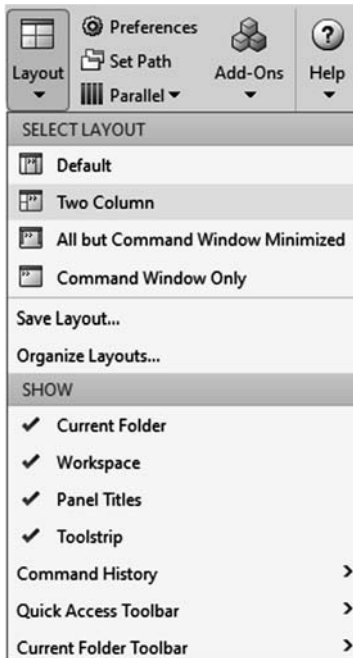


Рис. 1.2. Окно SELECT LAYOUT инструмента *Layout*

окна рабочей среды, и команда, с помощью которой можно создавать дополнительные кнопки пользователя, предназначенные для быстрого запуска на выполнение наиболее часто используемых наборов команд.

**Набор вкладок**, позволяющий активизировать соответствующие панели инструментов.

**Категории инструментов**, на которые разделен весь набор инструментов инструментальной панели вкладок.

**Панели инструментов**, содержащие наборы элементов, каждый из которых имеет соответствующее изображение и надпись, подсказывающее его назначение. Эти элементы позволяют выполнить наиболее часто встречающиеся команды, причем набор элементов инструментальной панели всегда соответствует текущей вкладке.

**Командное окно (Command Window)** отображает вводимые команды, результаты их выполнения и сообщения об ошибках.

**Окно рабочей области (Workspace)** отображает содержимое рабочего пространства объектов Matlab и позволяет выполнять определенные действия с объектами этого пространства (скалярными переменными, векторами, матрицами, функциями и др.).

**Окно истории команд (Command History)** осуществляет просмотр и повторный вызов ранее введенных команд.

**Окно текущего каталога (Current Folder)** предназначено для просмотра путей доступа к файловой системе. В этом окне перед работой в Matlab с конкретным файлом (чтение или запись файла на внешний носитель), расположенном в определенной папке, указывается путь доступа к файлу. В нижней части окна отображается детальная информация о типе выбранного файла.



Рис. 1.3. Стандартное меню



Рис. 1.4. Строка текущего каталога

Строка текущего каталога (рис. 1.4) предназначена для быстрого изменения текущего каталога.

Строка состояния отображает системные сообщения.

Редактор *m*-файлов (Matlab Editor) используется для создания и отладки *m*-файлов.

Редактор *mlx*-файлов (Live Editor) используется для создания и отладки *mlx*-файлов.

Редактор данных (Array Editor) используется для визуального просмотра и редактирования в основном одно- или двумерных массивов, которые расположены в рабочей области.

Профилировщик (Profiler) представляет собой графический интерфейс пользователя, помогающий увеличить скорость работы с *m*-файлами.

Окно для отображения графиков (Figure), в каждом из которых может быть отображен один или несколько графиков.

В стандартной конфигурации четыре основных окна вписаны в рабочую среду Matlab и закреплены (поставлены «на якорь»), т. е. они могут передвигаться вместе с основным окном и вместе с ним изменять свои размеры. В то время как компоненты с 1-го по 10-й почти всегда присутствуют в рабочей среде (см. рис. 1.1), компоненты с 11-го по 15-й и некоторые другие являются контекстно-зависимыми и появляются в рабочей среде по мере активизации соответствующих инструментов.

Границы между окнами можно изменять. Предварительно это окно нужно открепить (снять с якоря), и только тогда оно может занимать автономную позицию на экране. Расположение и границы всех окон можно настраивать командами окна **SELECT LAYOUT** (Выбор макета), которые открываются как нажатием на инструмент **Layout** рабочей среды (см. рис. 1.2), расположенный на инструментальной панели, так и командами контекстных меню соответствующих окон.

Контекстные меню любого окна можно отобразить на экране, щелкнув правой кнопкой мыши в область соответствующего окна при условии, что это окно активно.

## 1.2. Набор вкладок интерфейса Matlab и их инструменты

Рассмотрим основные компоненты рабочей среды (см. рис. 1.1) в стандартной конфигурации (Default) более подробно.

Набор вкладок интерфейса Matlab, расположенный в верхней части рабочей среды после загрузки Matlab состоит из четырех элементов: **HOME**, **PLOTS,APPS** и **SHORTCUTS**. Каждому из этих элементов соответствует своя панель инструментов.

При активизации элемента **HOME** (см. рис. 1.1) в меню отображается панель инструментов, в которой присутствуют инструменты, необходимые для текущей работы с Matlab. Инструменты разбиты на следующие категории.

**FILE** — категория, включающие инструменты, которые позволяют создавать новые наборы команд и программы и сохранять их в файлах; открывать существующие наборы команд и программы или загружать их из файлов; создавать различные объекты Matlab, осуществлять поиск файлов различных типов:

- **New Script** — открывает редактор для создания m-файлов (сценарий), при этом в рабочей среде добавляются еще три вкладки и соответствующие им панели инструментов (подробнее остановимся на этих инструментах в п. 1.2);
- **New** — создает новый документ Matlab, который может быть m-файлом, примером, классом, системным объектом, графиком и пр.;
- **Open** — открывает один из ранее созданных документов;
- **Find Files** и **Compare** — находит и сравнивает указанные файлы.

**VARIABLE** — категория, включающая инструменты, которые позволяют осуществлять различные действия с содержимым окна **Workspace** (рабочей области данных):

- **Import Data** — импортирует данные (файлы с различными расширениями) в систему Matlab;
- **Save Workspace** — сохраняет данные области **Workspace**;
- **New Variable** — открывает редактор для создания и редактирования переменных области **Workspace**, при этом в рабочей среде добавляются еще две вкладки и их панели инструментов (подробнее об этих инструментах в п. 1.2);
- **Open Variable** — открывает редактор для просмотра и редактирования переменных области **Workspace**;
- **Clear Workspace** — удаляет переменные из области **Workspace**.

**CODE** — категория, инструменты которой позволяют организовать анализ кода и профилирование, а также обнулять содержимое окон **Command Window**.

**SIMULINK** — категория, инструменты которой позволяют вызывать и работать с моделями **Simulink**;



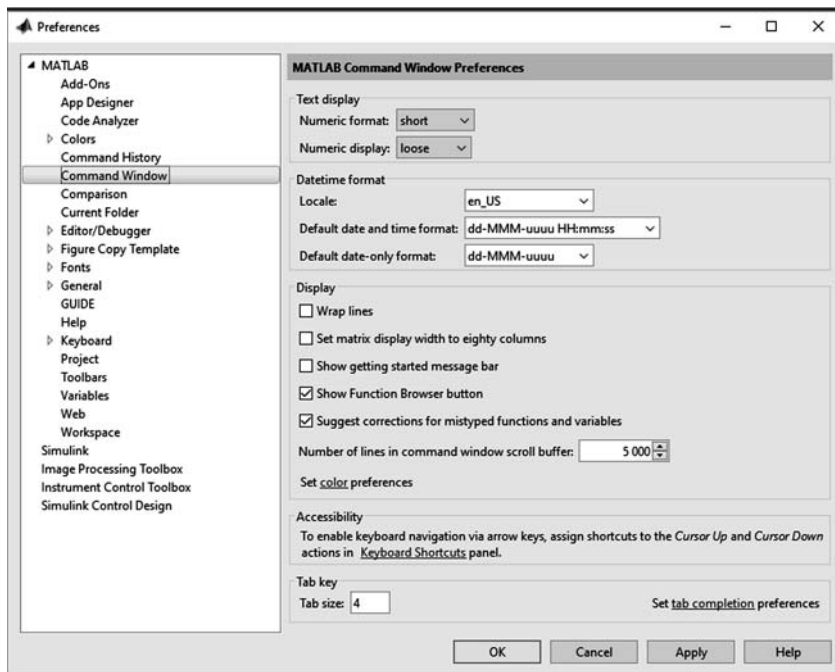


Рис. 1.5. Окно Preferences, в котором устанавливают свойства объектов среды системы Matlab

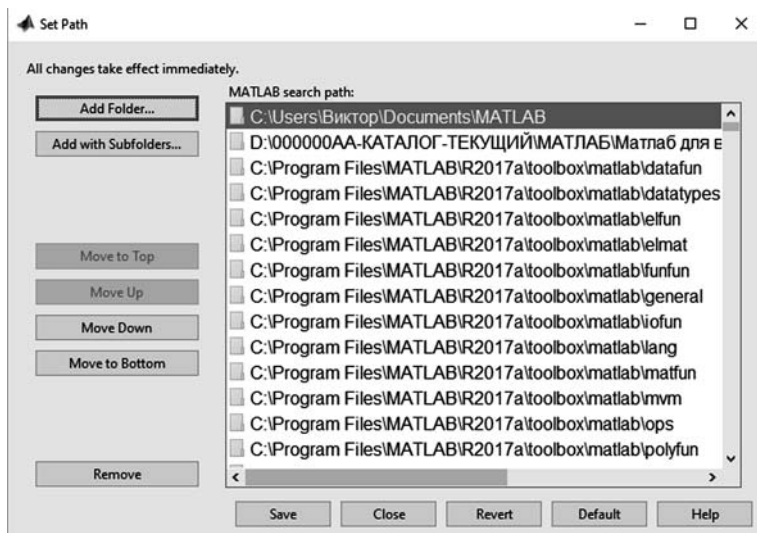


Рис. 1.6. Окно Set Path установки пути для текущих файлов

**ENVIRONMENT** — категория, инструменты которой позволяют устанавливать Preferences (Конфигурацию) и свойства объектов рабочей среды Matlab, а также пути доступа к файлам, где хранятся программы и данные:

- **Layout** — открывает окно **SELECT LAYOUT**, команды которого устанавливают конфигурацию окон рабочей среды (см. рис. 1.2);
- **Preferences** — открывает окно, команды которого устанавливают свойства объектов Matlab (рис. 1.5);
- **Set Path** — открывает окно установки путей для доступа к файлам (рис. 1.6), которое будут использоваться при работе Matlab. Для указания или добавления пути к файлу, расположенному в новой папке или на новом носителе, служит редактор доступа файловой системы. Его окно открывается кнопкой инструментальной панели **Set Path** (Установить путь). Окно показывает список папок с файлами Matlab. Имеется возможность переноса папок вверх или вниз по списку, их уничтожения и переименования. Установка путей доступа к системным файлам задается по умолчанию.
- **Parallel** — используется для организации параллельных вычислений.

**RESOURCES** — категория, инструменты которой позволяют осуществлять доступ к дополнительным пакетам и приложениям, к справочной системе Matlab и организовать поддержку этой системы компанией MathWorks.

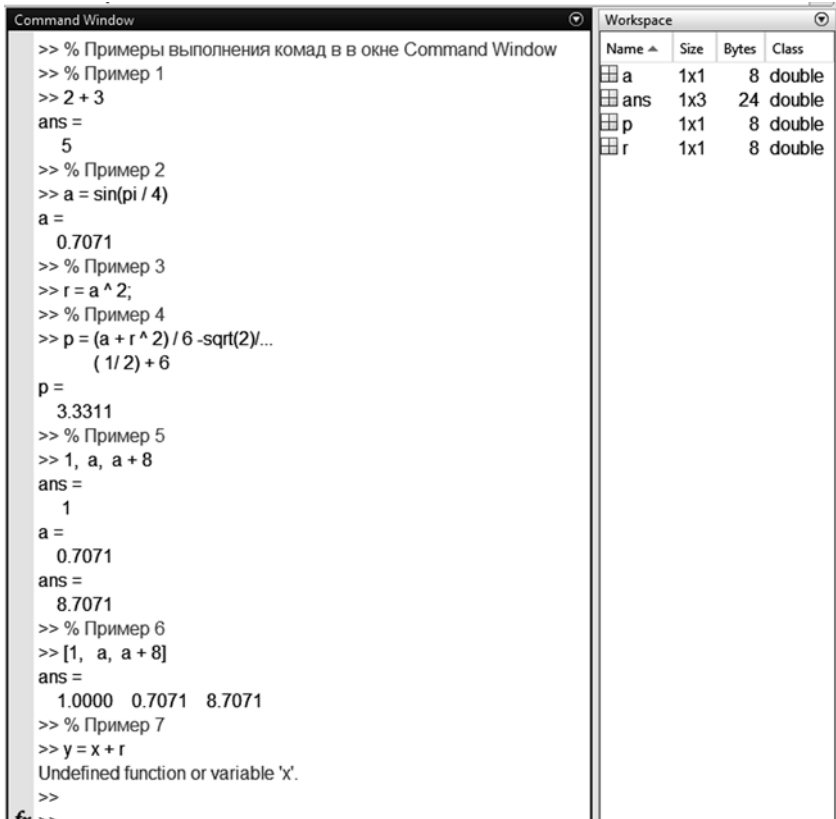
При активизации элемента **PLOTS** активизируется панель инструментов, которые используются для работы с различными типами графиков без программирования.

### 1.3. Основные окна рабочей среды

#### Работа в окне Command Window

Командное окно **Command Window** предназначено для ввода команд с соответствующими данными и вывода результатов их выполнения. Работа происходит в диалоговом режиме: пользователь вводит команду и передает ее ядру Matlab, ядро обрабатывает полученную команду и возвращает результат. Все команды вводятся в командные строки только после появления *приглашения* — `>>`. Заканчивается ввод каждой командной строки нажатием клавиши Enter.

Описанный выше сеанс работы с Matlab в окне **Command Window** принято называть *сессией*.



```
Command Window
>> % Примеры выполнения команд в окне Command Window
>> % Пример 1
>> 2 + 3
ans =
    5
>> % Пример 2
>> a = sin(pi / 4)
a =
    0.7071
>> % Пример 3
>> r = a ^ 2;
>> % Пример 4
>> p = (a + r ^ 2) / 6 - sqrt(2) / ...
    (1/2) + 6
p =
    3.3311
>> % Пример 5
>> 1, a, a + 8
ans =
    1
a =
    0.7071
ans =
    8.7071
>> % Пример 6
>> [1, a, a + 8]
ans =
    1.0000 0.7071 8.7071
>> % Пример 7
>> y = x + r
Undefined function or variable 'x'.
>>
~\~
```

Name	Size	Bytes	Class
a	1x1	8	double
ans	1x3	24	double
p	1x1	8	double
r	1x1	8	double

Рис. 1.7. Примеры простейших вычислений в окне Command Window

Рассмотрим несколько простых примеров, выполненных в диалоговом режиме в окне **Command Window** (рис. 1.7).

В примере 1 вычисляется результат вычисления  $2 + 3$ , по умолчанию создаётся переменная с именем **ans**, в которую записывается значение результата текущей операции, и это значение выводится на экран в следующей строке.

В примере 2 создаётся переменная **a**, вычисляется значение выражения  $\sin(\pi/4)$  и результат присваивается переменной **a**. Теперь эта переменная определена (атрибуты этой переменной отображаются в окне **Workspace**) и ее можно использовать для дальнейших вычислений.

В примере 3 выражение заканчивается точкой с запятой, которая «блокирует» вывод результата, но результат по-прежнему сохраняется в данном случае в переменной с именем **r**. При этом

атрибуты переменной `g` отображается в окне **Workspace**. Значение переменной `g` в любой момент можно вывести в строке окна **Command Window**, набрав имя переменной в командной строке и нажав клавишу `Enter`.

В примере 4 показана запись выражения в двух строках, где в качестве символа продолжения используется три точки.

В примере 5 показан вывод различных объектов Matlab в одной строке, отделенных друг от друга запятыми. Такая запись называется *выводом данных списком, разделенных запятыми*.

В примере 6 показан вывод одной строкой различных объектов Matlab, разделенных запятыми и заключенных в квадратные скобки. Такая запись позволяет отображать значения выводимых данных в одной строке.

В примере 7 выдается сообщение об ошибке, так как значение переменной `x` предварительно не определено.

*Обратите внимание, что предпочтительнее вычислять длинное выражение по частям с использованием промежуточных переменных.*

*Обратите внимание, что в любой момент значение переменной можно отобразить в командном окне, набрав имя переменной и нажав `Enter`, либо использовать функцию `disp` или другие средства вывода.*

Переменные системы Matlab размещаются в рабочей области оперативной памяти и отображаются в окне **Workspace** (рис. 1.8), информацию о переменных (имена, значения, размерности, класс и др.) можно в любой момент отобразить на экране с помощью команд Matlab `who` или `whos`.

Для очистки командного окна служит команда `clc`, а для очистки рабочей памяти — `clear all`. Очистить командное окно и рабочую память можно с помощью соответствующих команд инструментальной панели рабочей среды.

*Обратите внимание, что тип переменных объявлять не надо, так как по умолчанию все математические вычисления в Matlab выполняются с двойной точностью.*



Name	Value	Size	Class	Var
a	1.2246e-16	1x1	double	0
ans	5	1x1	double	0
p	3.1716	1x1	double	0
g	1.4998e-32	1x1	double	0

Рис. 1.8. Окно рабочей области **Workspace**

Формат вывода числовых значений на экран можно установить принудительно в окне **Command Window**, указав тип формата командой

`format тип формата`

или с помощью установки свойств в окне **Preferences**, которое можно вызвать инструментом панели инструментов вкладки **Home** (см. рис. 1.1).

Для установки формата в окне **Preferences** внутри группы **Text display** следует открыть список **Numeric format**, а затем из него выбрать один из предлагаемых форматов.

Например, тип формата **short** выводит короткое число с плавающей точкой, которое по умолчанию представляется с помощью четырех цифр после десятичной точки.

В стандартной конфигурации рабочей среды для выделения результатов вычисления или значений переменных Matlab вставляет перед выводимым значением пустую строку. Управлять появлением или отсутствием пустой строки можно в диалоговом окне **Preferences** внутри группы **Text display** с помощью форматов:

- **Compact** — строки с результатами выводятся подряд;
- **loose** — строки с результатами разделяются пустой строкой.

### Окно **Workspace**

Окно **Workspace** предназначено для быстрого просмотра атрибутов переменных, расположенных в рабочей области, для записи их в файл или для чтения из файла. В этом окне можно увидеть имя переменной (**Name**), значение (**Value**), ее размер (**Size**), число байтов, занимаемых переменной в памяти (**Bytes**), и ее тип (**Class**). Для идентификации класса переменной слева от имени используется соответствующая иконка.

Если окно **Workspace** не выведено на экран, то его можно отобразить либо с помощью соответствующей команды инструментальной панели, либо с помощью задания команды **Workspace** в командном окне.

Команды контекстного меню **Workspace** позволяют выполнять следующие действия: создать новую переменную (**New**); открыть редактор данных для просмотра или редактирования значений выделенной переменной (**Open Selection**); загружать данные из файла в рабочую область (**load**); сохранять рабочую область в файле (**Save**); печатать содержимое рабочей области (**Print**); удалить выделенную переменную (**Delete**); строить графики (**plot(d)**).

Полную информацию о содержании рабочей области можно получить с помощью команды **whos**, в результате выполнения которой

в окне **Command Window** выводится информация обо всех переменных и общий объем занимаемой ими памяти.

Приведем некоторые часто встречающиеся команды работы с данными, которые можно выполнить в окне **Command Window**:

- **clear имя переменной** — удаление переменной из рабочей области;
- **clear** без параметров — очистка всей рабочей области;
- **load имя файла** — загрузка всех данных из файла;
- **load имя файла + имя переменной** — выборочная загрузка данных;
- **save имя файла** — сохранение рабочей области на диске;
- **save имя файла + имя переменной** — выборочное сохранение переменных из рабочей области.

### Окно Command History

```

Command History
%-- 21.02.2015 9:43 --%
% Пример 1
2+3
% Пример 2
a=sin(pi)
% Пример 3
r=a^2;
a
%Пример 4
- y=x+r
% Пример
p=(a+r^2)/6-sqrt(2)...
r/(1/2)+6
  
```

Рис. 1.9. Окно истории команд **Command History**

На рис. 1.9 показано окно **Command History**. Это окно служит для просмотра команд, введенных ранее в командной строке **Command Window**.

Изменять параметры работы окна истории команд можно с помощью диалогового окна **Preferences** (см. рис. 1.1) при активном окне **Command History**. Окно **Command History** хранит все командные строки, набираемые пользователем. В отличие от содержимого **Command Window**, здесь не отображаются со-

общения системы и результаты вычислений.

Окно **Command History** целесообразно использовать при вводе команд, однотипных вычислений или при повторениях группы командных строк. Чтобы ввести в текущую строку содержимое ранее введенной командной строки, достаточно подобрать путем многократного нажатия клавиш  $\uparrow$  или  $\downarrow$ . Перенести конкретную строку в командное окно можно также двойным щелчком мыши по нужной строке в окне **Command History**.

Для очистки области истории команд используются команда **clear** или команда контекстного меню **Command History** — **clear**.