MATLAB® in Bioscience and Biotechnology
Volumes in the Series

22. *Process Validation for the Production of Biopharmaceuticals: Principles and Best Practice. Applied Strategies for Bioprocess Development and
Biohealthcare Publishing Series on Medical and Health Science, Technology and Policy

Volumes in the Series


Biohealthcare Publishing Series on Bioinformatics

Volumes in the Series


In memory of my father Matvey.  
To my mother Leda, my wife Inna, and my son Dmitri
# Contents

Preface xi
List of figures and tables xiii
About the author xvii

1 Introduction 1

2 MATLAB® basics 3
   2.1 Starting with MATLAB® 3
   2.2 Vectors, matrices and arrays 15
   2.3 Flow control 33
   2.4 Questions for self-checking and exercises 44
   2.5 Answers to selected exercises 48

3 MATLAB® graphics 49
   3.1 Generation of XY plots 49
   3.2 Generation of XYZ plots 62
   3.3 Specialized 2D and 3D plots 73
   3.4 Application examples 79
   3.5 Questions for self-checking and exercises 88
   3.6 Answers to selected exercises 93

4 Script, function files and some useful MATLAB® functions 95
   4.1 Script file 95
   4.2 Functions and function files 100
   4.3 Some useful MATLAB® functions 104
   4.4 Application examples 116
   4.5 Questions for self-checking and exercises 127
   4.6 Answers to selected exercises 129
## 5 Ordinary and partial differential equation solvers

- 5.1 Solving ordinary differential equations with ODE solvers 133
- 5.2 Solving partial differential equations with the PDE solver 151
- 5.3 Questions for self-checking and exercises 162
- 5.4 Answers to selected exercises 167

## 6 Bioinformatics tool for sequence analysis

- 6.1 About toolboxes 171
- 6.2 The functions of the Bioinformatics toolbox™ 172
- 6.3 Public databases, data formats and commands for their management 173
- 6.4 Sequence analysis 182
- 6.5 Sequence analysis examples 197
- 6.6 Questions for self-checking and exercises 209
- 6.7 Answers to selected exercises 211

Appendix: MATLAB® characters, operators and commands 217

Index 223
In the last few decades, two seemingly disparate sciences – computer science and biology – have interpenetrated and affected one another. We see students enrolled in computer science beginning their careers in biotechnological laboratories, and biotechnologists creating bio-computers and being active in computer science. There appears to be an urgent need to familiarize biotechnologists with the same computing tools as are imparted to technicians.

This book represents a short introduction to MATLAB® oriented towards various collaborative areas of biotechnology and bioscience. My hope is that it will be equally useful to undergraduate and graduate students and to practising engineers. It concentrates on the fundamentals of MATLAB® and gives examples of its application to a wide range of current bioengineering problems in computational biology, molecular biology, biokinetics, biomedicine, bioinformatics and biotechnology. In the last decade MATLAB® has been presented to students as a basic computational tool that they need to learn. Consequently, many students unfamiliar with programming, engineers and scientists have come to regard it as user-friendly and highly convenient in solving their specific problems. Numerous books are available on programming in MATLAB® for engineers in general, irrespective of their specialization, or for those specializing in some specific area, but none has been designed specifically for a wide, interdisciplinary, topical area such as bioengineering. Thus, MATLAB® is presented here with examples and applications to various school- and advanced bioengineering problems – from growing populations of microorganisms and population dynamics, to reaction kinetics and reagent concentrations, predator–prey models, mass-transfer problems, and to sequence analysis and sequence statistics.

The book distills my experience of many years of MATLAB® teaching in introductory and advanced courses for students, engineers and scientists specializing in bioscience and engineering.

I would like to thank the people who attracted me to the subject and thereby played key roles in the inception and appearance of this book: my
colleague Professor Rosa Azhari (Biotechnology Department, ORT Braude College), and software support team head Moshe Barak (Computer Center, Technion – Israel Institute of Technology). I also thank MathWorks Inc. (3 Apple Hill Drive, Natick, MA 01760-2098, USA, Tel: 508-647-7000, Fax: 508-647-7001, E-mail: info@mathworks.com, Web: www.mathworks.com) who graciously granted permission to reproduce material appearing in this book.

I thank Ing. Eliezer Goldberg, former resident scientific editor at Technion, for patience and invaluable editorial assistance, and would also like to thank Dr Glyn Jones, head of Biohealthcare Publishing (Oxford) Ltd, for invaluable support throughout all stages of publication of this book.

I hope this book will prove useful to students and engineers in both natural and life sciences and provide them with an opportunity to work with one of the finest software tools.

Any reports of errata or bugs, comments and suggestions on the book’s contents will be accepted gratefully by the author.

Leonid Burstein
Nesher, Haifa, Karmiel, Israel
September 2010
List of figures and tables

Figures

2.1  MATLAB® logo. The image can be produced with the logo command; the background color has been changed 4
2.2  MATLAB® desktop 4
2.3  Command Window; the view after separation from the desktop 6
2.4  Help window with information about the aminolookup command 9
3.1  Biomass data plotted in the Figure Window with default settings 50
3.2  Biomass data generated with specifiers and property settings in the plot command 52
3.3  Two curves (sine and cosine) in a single plot 54
3.4  sin x, cos x and sin x² in a single plot 54
3.5  Possible arrangements of one page in four panes 55
3.6  Four plots on the same page 56
3.7  Sine and cosine plot constructed with the axis tight command 58
3.8  Biomass data plot formatted with the xlabel, ylabel, title, text and grid commands 60
3.9  Plot of the sine and cosine functions with legend 61
3.10  Plot Editor buttons in the Figure Window 62
3.11  Line in 3D coordinates 64
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.12</td>
<td>Points in 3D interpretation and their x,y-plane projection</td>
<td>65</td>
</tr>
<tr>
<td>3.13</td>
<td>Mesh plot</td>
<td>67</td>
</tr>
<tr>
<td>3.14</td>
<td>Surface plot</td>
<td>68</td>
</tr>
<tr>
<td>3.15</td>
<td>Boxed surface plot</td>
<td>69</td>
</tr>
<tr>
<td>3.16</td>
<td>Viewpoint, azimuth and elevation</td>
<td>70</td>
</tr>
<tr>
<td>3.17</td>
<td>The function $z = e^{-x^2-y^2}$ with viewing angles</td>
<td>71</td>
</tr>
<tr>
<td>3.18</td>
<td>Plot in rotation regime with the rotate cursor and values of azimuth and elevation angles</td>
<td>72</td>
</tr>
<tr>
<td>3.19</td>
<td>Plot of biomass–time data with error bars</td>
<td>74</td>
</tr>
<tr>
<td>3.20</td>
<td>Histogram of the weight data</td>
<td>75</td>
</tr>
<tr>
<td>3.21</td>
<td>Semi-logarithmic graph for the relationship between time elapsed and residual radioactivity</td>
<td>76</td>
</tr>
<tr>
<td>4.1</td>
<td>The Editor Window</td>
<td>96</td>
</tr>
<tr>
<td>4.2</td>
<td>Script file in the Editor Window</td>
<td>97</td>
</tr>
<tr>
<td>4.3</td>
<td>The ‘Current Folder’ field and ‘Browse for Folder’ Window</td>
<td>98</td>
</tr>
<tr>
<td>4.4</td>
<td>Typical function file in the Editor Window</td>
<td>101</td>
</tr>
<tr>
<td>4.5</td>
<td>Original data, interpolation and extrapolation points</td>
<td>105</td>
</tr>
<tr>
<td>4.6</td>
<td>Definite integral of the function $f(x)$ given analytically and by the data points</td>
<td>108</td>
</tr>
<tr>
<td>4.7</td>
<td>Geometrical representation of the derivative</td>
<td>111</td>
</tr>
<tr>
<td>4.8</td>
<td>First-degree polynomial fit</td>
<td>115</td>
</tr>
<tr>
<td>5.1</td>
<td>Concentration–time dependence for second-order reaction; the solution to $-\frac{d[A]}{dt} = k[A]$, with $A_0 = 0.25$</td>
<td>138</td>
</tr>
<tr>
<td>5.2</td>
<td>Concentration–coordinate–time dependences; the solution to $\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}$. The level values were replaced with the Plot Editor</td>
<td>156</td>
</tr>
<tr>
<td>6.1</td>
<td>Matlab® Web Browser with NCBI home page</td>
<td>176</td>
</tr>
<tr>
<td>6.2</td>
<td>The page with accession number for rat hexosaminidase A</td>
<td>177</td>
</tr>
</tbody>
</table>
List of figures and tables

6.3 Bar chart of amino acid amounts produced by the aacount command for a randomly generated 25-letter sequence 186
6.4 A, C, G, T and A, T, C, G nucleotide density plots generated by the ntdensity command for the rhesus macaque semen sequence 187
6.5 Sequence score example 190
6.6 Scoring space heat map and winning path generated with the nwalign command 192
6.7 Scoring space heat map and winning path generated with the swalign command 193
6.8 Dot plot for mouse and rat hexoaminidase A sequences, generated by the seqdotplot command 195
6.9 Aligned Sequence window generated by the showalignment command; pairwise alignment 196
6.10 Aligned Sequence window generated by the showalignment command; multiple alignment 196
6.11 Multiple Sequence Alignment Viewer with four aligned sequences generated with the multialignviewer command 198

Tables

2.1 Elementary and trigonometric mathematical functions 7
2.2 Biomass data 15
2.3 Enzyme activity (mg⁻¹) 17
2.4 Command for matrix manipulations, generation and analysis 26
2.5 If statements 37
2.6 Loops 38
3.1 Line style, color and marker type specifiers 51
3.2 Property names and property values 51
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>Additional commands and plots for 2D and 3D graphics</td>
<td>77</td>
</tr>
<tr>
<td>3.4</td>
<td>Air temperature–density data</td>
<td>84</td>
</tr>
<tr>
<td>5.1</td>
<td>MATLAB® ODE solvers</td>
<td>135</td>
</tr>
<tr>
<td>6.1</td>
<td>Additional commands for database management</td>
<td>179</td>
</tr>
<tr>
<td>6.2</td>
<td>Additional utility and statistical commands</td>
<td>188</td>
</tr>
</tbody>
</table>
About the author

Leonid Burstein is Senior Lecturer at large at Technion – Israel Institute of Technology, at the ORT Braude College, in the Biotechnology and Software Engineering Departments, and at a number of other universities and highschools in Western and Lower Galilee.

Following an MA in thermophysics at Lomonosov Technological Institute at Odessa, Ukraine, and a doctorate at the National Research Institute for Physical and Radio Engineering Measurements at Moscow, he obtained his PhD in physical properties of materials from the Heat/Mass Transfer Institute of the Belarus Academy of Science, Minsk, in 1974. After a short period of work in Russia and Belarus, Dr Burstein started his carrier at the Piston Ring Institute in Odessa, where he served from 1974 to 1990 as Head of Projects and Head of the CAD/CAM group. In 1991, he began work at the Technion – IIT, Israel, at the Faculty of Mechanical Engineering, in the Quality Assurance and Reliability Program at the Faculty of Industrial Engineering and Management, and at the Taub Computer Centre as an advisor on MATLAB® and other scientific software. He also worked at the Technion Research and Development Foundation as principal researcher in funded projects in various areas such as diesel tribology and environment control. He also taught various courses at Haifa University, at the Technion, at the Kinnereth Academic College and elsewhere. He currently teaches a MATLAB® course for biotechnologists at ORT Braude College.

He is an Editorial Board Member and reviewer for a number of international journals and a Committee Member of numerous conferences. He is the author of several patents, has published four chapters in scientific books and authored/co-authored more than 60 publications in leading scientific journals.
He can be contacted at:

Technion – Israel Institute of Technology
Technion City
32000 Haifa
Israel
E-mail: leonidburstein@gmail.com (preferable) or leonidb@technion.ac.il
Introduction

Everything that can be counted – should be.
Anonymous

Biological engineering is defined as application of engineering principles to the widest spectrum of living systems – from molecular biology, biochemistry and microbiology, to bio-medicine, genetics and bioinformatics. And as in general engineering, computers and the ability to use them are vitally important. This is true also for other professionals of any bio-industry. Thus, bio-specialists and scientists working in these areas need to have the computational resources to be able to solve various problems. A widespread and powerful tool for such purposes is MATLAB® – the software for technical computing. It is designed to solve both general and specific problems; of these, the latter are treated with so-called toolboxes, which currently include means specialized for bio-problems. An obstacle to the effective understanding and implementation of MATLAB® in practice is the inadequate level of math reached by students and specialists in areas of bioscience, combined with a lack of textbooks tailored to such audiences. This book is intended as a remedy. It is organized as follows.

I begin by covering primary MATLAB® programming and then move to more complicated problems by means of this language; the material is illustrated throughout by examples from different areas of bioengineering and biological science. The topics were chosen on the basis of several years of teaching MATLAB® for biotechnologists and they are presented so that inexperienced users can progress gradually, with the previously presented material being the only prerequisite for each new chapter.

Chapter 2 introduces the MATLAB® environment, language design, help options, variables, matrix and array manipulations, elementary and special functions, flow chart control, conditional statements and other basic MATLAB® features.
In chapter 3 the plotting tool is described by using examples of graphic presentation in various calculations. Mastering the material in chapters 2 and 3 will allow readers to create their own MATLAB® programs.

Chapter 4 presents the MATLAB® script- and m-files; the commands for numerical integration, differentiation, inter-/extrapolation and curve fitting, together with their various applications, are given.

In chapter 5, particular solutions for ordinary and partial differential equations are briefly presented together with examples from bio-systems involving a single differential equation or a set. This chapter assumes a somewhat greater familiarity with mathematics.

In the final chapter the bioinformatics tool is introduced through applications employed in sequence analysis and statistics. Emphasis is placed on DNA and protein sequence database access and further pairwise or multiple alignments.

The Appendix details the studied MATLAB® commands and functions.

Application problems included at the end (and sometimes in the middle) of each chapter are solved with commands accessible to the reader; the solutions are not necessarily the shortest or most original, but should be easy to understand and follow up. Readers are invited to write their own solutions and check the results against those given herein. At the end of each chapter are questions and problems, and readers are encouraged to attempt them for better assimilation of the material. The contexts and values used in the problems are not factual and are intended for learning purposes only.

The MATLAB® used in the book is R2010a, version 7.10.0. Each subsequent version should incorporate all previous ones; hence, the fundamental commands given here should be valid in future versions. It is assumed that the user has a computer with MATLAB® installed on it and is able to perform basic computer operations.

Each command is explained here in its simplest form; additional information is available in the MATLAB®-help or original MATLAB® documentation.

Let us begin.
MATLAB® basics

MATLAB® came into being in the 1970s as a tool for mathematicians and educators, but was soon adopted by engineers as an effective means for technical computing. Its name is a composite of the words ‘Matrix’ and ‘Laboratory’, emphasizing that its main element is the matrix. Such an approach permitted unification of the processes of various calculations, graphics, modeling, simulation and algorithm development. This chapter introduces the main windows and starting procedure, describes the main commands for simple arithmetic, algebraic and matrix operations, and presents the basic loops and relational and logical operators.

2.1 Starting with MATLAB®

MATLAB® can be installed on computers running different operation systems, but I will assume here that the reader uses a personal computer running a Windows operating system. To start one has simply to click on the MATLAB® icon (Figure 2.1) provided with a MATLAB® subscription; the icon is placed on the Quick Lunch bar or on the Windows Desktop. Another way to start the program is to select MATLAB® 20010a in the MATLAB®-directory in the ‘All Programs’ option of the Windows ‘Start’ menu.

2.1.1 MATLAB® Desktop and its windows

The window that first opens is the MATLAB® Desktop (Figure 2.2), which comprises four windows: Command, Current Folder, Workspace and Command History.

These are the most intensively used windows and are briefly described further. There are also Help, Editor and Figure windows that do not appear
Figure 2.1 MATLAB® icon (enlarged). The image can be produced with the `logo` command; the background color has been changed.

Figure 2.2 MATLAB® desktop.
with the MATLAB® Desktop and are described in the chapters where they are used.

The Desktop also contains: the Menu, which can be changed depending on the tool being used; the MATLAB® Tools bar, which contains the more common functions; the Shortcuts bar, where one can place icons for quick running of MATLAB® programs or group commands; and the Start button, used to access various tools, demos, shortcuts and documentation.

The Command Window is the main outlet where commands are entered and results are displayed. Sometimes it is convenient to separate it from the desktop by clicking to the right of the title bar. Such separation is possible for all Desktop windows. To combine windows one has to click on or select the Default line in the Desktop Layout of the Desktop option at the Menu bar.

Workspace is the graphical interface that allows us to view and manage the variables and other objects of the MATLAB® workspace; it also displays and automatically updates the values of each variable.

Current Folder presents a browser that shows the full path to the current folder, and shows the contents of the current folder. When starting MATLAB®, we view a starting directory which is called the startup directory. After selecting the file, information about it appears in the Details panel.

Command History stores the commands most recently entered in the Command Window.

2.1.2 Elementary functions and interactive calculations

Two main working modes are available in MATLAB® – interactive and with m-files. I will explain the latter in later chapters. The interactive mode is discussed briefly here.

To enter and execute a command, it must be typed in the Command Window immediately after the command prompt >>. Figure 2.3 shows this window with some elementary commands.

The symbol , which appears in the most recent versions of MATLAB®, is called the Function Browser, and helps to find the function required and information about its syntax and usage.

Entering a command and manipulating with it require us to master the following operations:

- the command must be typed next to the prompt >>;
- the Enter key must be pressed for execution;
a command in a preceding line cannot be changed; to correct or repeat
an executed command the up-arrow key ↑ should be pressed;

• a long command can be continued in the next line by typing ... – three
periods; commands in the same line should be divided by semicolons (;)
or by commas (,); a semicolon at the end of a command prevents the
answer from being displayed;

• the symbol % (percentage symbol) designates those comments that should
be written after it in the line, and the comments are not executed after
entering;

• the command clc clears the Command Window.

The Command Window can be used as a calculator by using the following
symbols for arithmetical operations: + (addition), − (subtraction),
∗ (multiplication), / (right division), \ (left division, used mostly for
matrices), ^ (exponential function).

These operations are applicable to a wide variety of elementary and
trigonometric functions that should be written as the name with the
argument in parentheses, e.g. sin x should be written as sin(x); in
trigonometric functions the argument x should be given in radians. A short
list of such functions and variables is given in Table 2.1. Hereinafter the
operations executed in the Command Window are written after the
command line prompt (>>), and the user will need to press the Enter key
after entering one or more commands written in one command line.
Table 2.1 Elementary and trigonometric mathematical functions

<table>
<thead>
<tr>
<th>Functions and constants in Math</th>
<th>MATLAB® presentation</th>
<th>MATLAB® example (inputs and outputs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>x</td>
<td>$ – absolute value</td>
</tr>
<tr>
<td>$e^x$ – exponential function</td>
<td>$\text{exp}(x)$</td>
<td>$\gg \text{exp}(2.7)$ $\text{ans} = 14.8597$</td>
</tr>
<tr>
<td>$\ln x$ – natural (base $e$) logarithm</td>
<td>$\text{log}(x)$</td>
<td>$\gg \log(10)$ $\text{ans} = 2.3026$</td>
</tr>
<tr>
<td>$\log x$ – Napierian (base 10) logarithm</td>
<td>$\text{log10}(x)$</td>
<td>$\gg \log10(10)$ $\text{ans} = 1$</td>
</tr>
<tr>
<td>$\sqrt{x}$ – square root</td>
<td>$\text{sqrt}(x)$</td>
<td>$\gg \text{sqrt}(2/3)$ $\text{ans} = 0.8165$</td>
</tr>
<tr>
<td>$\pi$ – the number $\pi$</td>
<td>$\pi$</td>
<td>$\gg 2*\pi$ $\text{ans} = 6.2832$</td>
</tr>
<tr>
<td>Round towards minus infinity</td>
<td>$\text{floor}(x)$</td>
<td>$\gg \text{floor}(-12.1)$ $\text{ans} = -13$</td>
</tr>
<tr>
<td>Round to the nearest integer</td>
<td>$\text{round}(x)$</td>
<td>$\gg \text{round}(12.6)$ $\text{ans} = 13$</td>
</tr>
<tr>
<td>$\sin x$ – sine</td>
<td>$\sin(x)$</td>
<td>$\gg \sin(\pi/3)$ $\text{ans} = 0.8660$</td>
</tr>
<tr>
<td>$\cos x$ – cosine</td>
<td>$\cos(x)$</td>
<td>$\gg \cos(\pi/3)$ $\text{ans} = 0.5000$</td>
</tr>
<tr>
<td>$\tan x$ – tangent</td>
<td>$\tan(x)$</td>
<td>$\gg \tan(\pi/3)$ $\text{ans} = 1.7321$</td>
</tr>
<tr>
<td>$\cot x$ – cotangent</td>
<td>$\cot(x)$</td>
<td>$\gg \cot(\pi/3)$ $\text{ans} = 0.5774$</td>
</tr>
<tr>
<td>$\arcsin x$ – inverse sine</td>
<td>$\text{asin}(x)$</td>
<td>$\gg \text{asin}(1)$ $\text{ans} = 1.5708$</td>
</tr>
<tr>
<td>$\arccos x$ – inverse cosine</td>
<td>$\text{acos}(x)$</td>
<td>$\gg \text{acos}(1)$ $\text{ans} = 0$</td>
</tr>
<tr>
<td>$\arctan x$ – inverse tangent</td>
<td>$\text{atan}(x)$</td>
<td>$\gg \text{atan}(1)$ $\text{ans} = 0.7854$</td>
</tr>
<tr>
<td>$\arccot x$ – inverse cotangent</td>
<td>$\text{acot}(x)$</td>
<td>$\gg \text{acot}(1)$ $\text{ans} = 0.7854$</td>
</tr>
<tr>
<td>$n!$ – factorial</td>
<td>$\text{factorial}(n)$</td>
<td>$\gg \text{factorial}(5)$ $\text{ans} = 120$</td>
</tr>
</tbody>
</table>
The result of entering a command is a variable with name \texttt{ans}. The equal sign (\texttt{=}) is called the assignment operator and is used to specify a value to a variable, e.g. to the \texttt{ans}. An entered new value cancels its predecessor.

Arithmetic operations are performed in the following order: operations in parentheses (starting with the innermost), exponentiation, multiplication and division, addition and subtraction. If an expression contains operations of the same priority, they run from left to right.

Examples of arithmetic operations in the Command Window are given below:

\begin{verbatim}
>> 6/2/5
ans =
 6.4000
>> (6+2)/5
ans =
 1.6000
>> 6+2/5
ans =
 8.5000
>> 6.203*10^23
ans =
>> 2.3^1/3, 2.3^(1/3)
ans =
 0.7667
ans =
 1.3200
>> 2^1.72^1/4+log(15*1.005)/asin(pi/8)...-sqrt(8.3)
ans =
 1.9373
\end{verbatim}

The outputted numbers are displayed here in short format (default format) – a fixed point followed by four decimal points. The format can be changed to long, 14 digits after the point, by typing the command: \texttt{format long}. To return to the default format the user has to type \texttt{format}.

There are other formats that can be obtained by typing \texttt{help format}; the word after \texttt{help} appears in blue, for ease of viewing.

2.1.3 Help and Help Window

For information about use of some commands, type and enter \texttt{help} with the command name after a space next to this word, e.g. \texttt{help format} as above. The explanations appear immediately after this in the Command Window.
For a command concerning a particular topic of interest, the `lookfor` command may be used. For example, for the name of MATLAB® command(s) on the subject of codons one should enter `lookfor codon` and the commands will subsequently appear on the screen, as shown below:

```
>> lookfor codon
codonbias - reports codon usage per amino acid for a DNA sequence.
codoncount - report codon counts for a sequence.
aminolookup - displays AA codes, integers, abbreviations, names, and codons.
```

For further information the user has to click on the selected command or again use the `help` command. To interrupt the search process, the two abort keys `Ctrl` and `c` should be clicked together; these keys should also be used to interrupt any other process, e.g. that of program/command execution.

For more detailed information one can similarly use the `doc` command, e.g. `doc aminolookup`, in which case the Help window will be opened. The Help window can also be opened by selecting the Product Help line in the Help options on the MATLAB® Desktop menu line (Figure 2.4).

![Help window with information about the aminolookup command.](Figure 2.4)
The Help window comprises three panes: on the left are the Contents or Search Results and on the right is the page containing information on the topic. Information on any subject is obtainable by typing the word(s) into the search line in the upper left-hand corner. The Search Results pane shows a preview of where the search words were found within the page, and the concrete information is displayed on the right.

2.1.4 Variables and commands for management of variables

A variable is a symbolic term written as a letter(s) and associated with a concrete numerical value. MATLAB® allocates memory space for storage of variable names and their values. A variable can be a scalar – a single number – or an array – a table of numbers. The name can be as many as 63 characters long, and contain letters, digits and underscores, but the first character must be a letter. Existing commands (sin, cos, sqrt, etc.) cannot be used as names.

The assignment and usage of variables in algebraic calculations is demonstrated next.

```
>> a=3
a =
3
>> b=4
b =
4
>> c=sqrt(a^2+b^2)
c =
5
```

Predefined MATLAB® variables can be used without being assigned. Except for the previously mentioned pi and ans, these are inf (infinity), i or j (square root of -1), and NaN (not-a-number, used when a numerical value is moot, e.g. 0/0).

The following commands can be used for management of variables: clear, to remove from memory; clear x y, for removing named variables x and y only; who for displaying the names of variables; or whos for displaying variable names, matrix sizes, variable byte sizes and variable classes. This information can also be obtained in the Workspace Window, where each variable is presented by the icon with the same information as in the case of whos but with additional data; the popup menu for selection of desirable information appears by right-clicking with the cursor placed on the Workspace Window menu line.
2.1.5 Output commands

As previously noted, MATLAB® automatically displays the result after each command is entered, but does not display it if the command is followed by a semicolon. MATLAB® has additional display commands, the two most frequently used of which are \texttt{disp} and \texttt{fprintf}.

The \texttt{disp} command is used to display text or variable values without the name of the variable and the equal sign. Each new \texttt{disp} command yields its result in a new line. In general form the command reads

\begin{verbatim}
disp('Text string') or disp(Variable name)
\end{verbatim}

The text between quotes is displayed in blue.

For example:

\begin{verbatim}
>> Na=6.0221*10^-23
Na =
 6.0221e+023
>> disp('Avogadro Const'),disp(Na)
Avogadro Const
 6.0221e+023
\end{verbatim}

The \texttt{fprintf} command is used to display text and data or to save them to file. The command has various forms that present difficulties for beginners, and here I give the simplest of them for displaying the results of a calculation.

To display text and a number on the same line the following form is used:

\begin{verbatim}
fprintf('Text string %6.3f additional text', variable name)
\end{verbatim}

The \% marks the place where we intend to type the number

- \% marks the place where the number will be displayed in the place marked by \%
- 6 – field width – the number including sign
- 3 – number of digits after the point
- f – fixed point conversion character

To divide a text into two or more lines, or starting with a new line, \texttt{\textbackslash n (slash n)} must be written before the word or sign that we want to see on the new line.
The field width and the number of digits after the point (6.3 in the example presented) are optional; the sign % and the character f, called conversion character, are obligatory. The character f specifies the fixed point notation in which the number is displayed. Some additional notations that can be used are: i, integer, e, exponential (e.g. 2.309123e+001); and g, the more compact form of e or f, with no trailing zeros.

Addition of several %f units (or full formatting elements) permits inclusion of multiple variable values in the text. For example, using the fprintf command:

```
>> m_H2=2.01588; m_CL2=70.906;
>> fprintf('Mass of H_2 is %6.3f g/mol\nMass of Cl_2 is %6.3f g/mol\n',m_H2,m_CL2)
Mass of H_2 is 2.016 g/mol
Mass of Cl_2 is 70.906 g/mol
```

The color of the text in quotes is the same as in disp (blue).

The commands described can be used to output tables as will be shown later, after introduction of vectors and matrices.

### 2.1.6 Application examples

#### 2.1.6.1 DNA volume

As shown here

![Diagram of DNA molecule](image)

the idealized volume of the DNA molecule can be calculated using the expression for the volume of a cylinder:

\[
V = \pi (d/2)^2 h
\]

where \( r \), the radius of the DNA molecule, is about \( 1.58 \times 10^{-3} \) µm, and \( h \), its length, is \( 3.34 \times 10^{-3} \) µm.