MATLAB® 7

The Language of Technical Computing

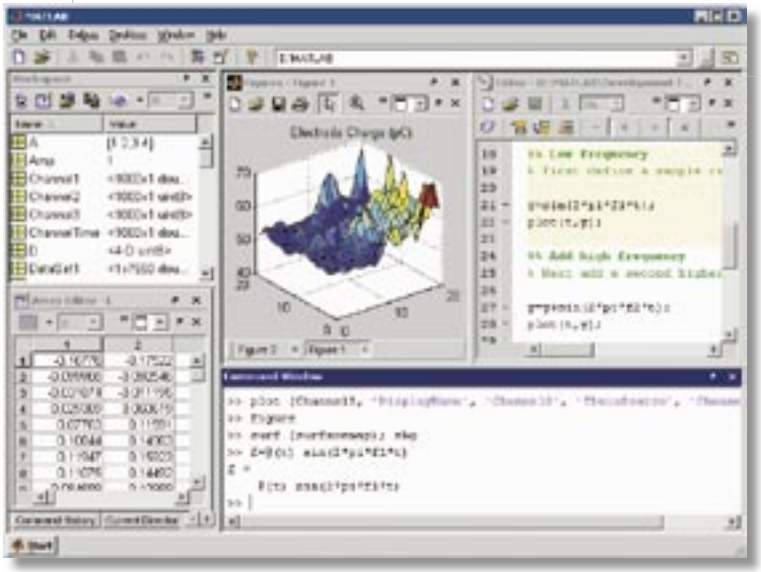
MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. Using MATLAB, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.

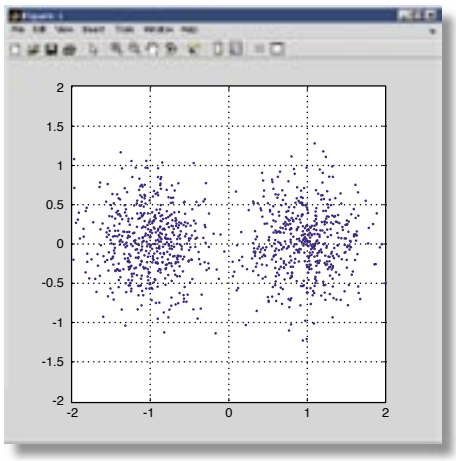
MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

KEY FEATURES

- High-level language for technical computing
- Development environment for managing code, files, and data
- Interactive tools for iterative exploration, design, and problem solving
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration
- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces
- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java, COM, and Microsoft Excel



The MATLAB environment lets you develop algorithms, interactively analyze data, view data files, and manage projects.



```
% Generate a
vector of N bits
N = 1024;
Bits = rand(N,1)>0.5;

% Convert to symbols
Tx = 1-2*Bits;

% Add white Gaussian noise
P = 0.4;
Nz = P*(randn(N,1)+i*randn(N,1));
Rx = Tx + Nz;

% Display constellation
plot(Rx,'.');
axis([-2 2 -2 2]);
axis square, grid;
```

A communications modulation algorithm that generates 1,024 random bits, performs modulation, adds complex Gaussian noise, and plots the result–all in just 9 lines of MATLAB code.

Developing Algorithms and Applications

MATLAB provides a high-level language and development tools that let you quickly develop and analyze your algorithms and applications.

The MATLAB Language

The MATLAB language supports the vector and matrix operations that are fundamental to engineering and scientific problems. It enables fast development and execution.

With the MATLAB language, you can program and develop algorithms faster than with traditional languages because you do not need to perform low-level administrative tasks, such as declaring variables, specifying data types, and allocating memory. In many cases, MATLAB eliminates the need for 'for' loops. As a result, one line of MATLAB code can often replace several lines of C or C++ code.

At the same time, MATLAB provides all the features of a traditional programming language, including arithmetic operators, flow control, data structures, data types, object-oriented programming (OOP), and debugging features.

MATLAB lets you execute commands or groups of commands one at a time, without compiling and linking, enabling you to quickly iterate to the optimal solution.

For fast execution of heavy matrix and vector computations, MATLAB uses processor-optimized libraries. For general-purpose scalar computations, MATLAB generates machine-code instructions using its JIT (Just-In-Time) compilation technology.

This technology, which is available on most platforms, provides execution speeds that rival those of traditional programming languages.

Development Tools

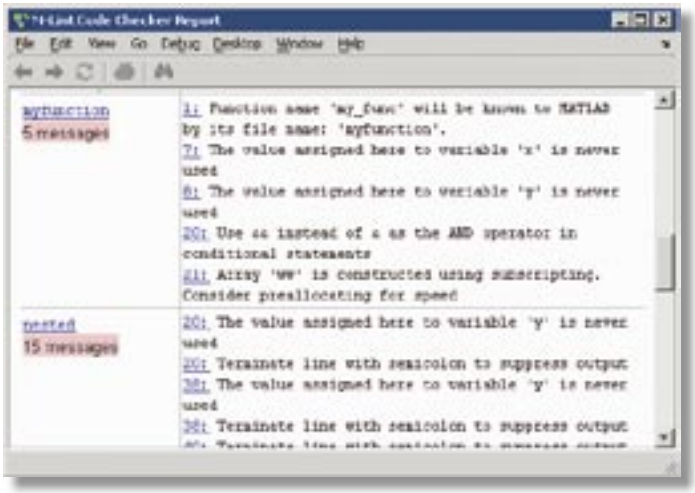
MATLAB includes development tools that help you implement your algorithm efficiently. These include the following:

MATLAB Editor—Provides standard editing and debugging features, such as setting breakpoints and single stepping

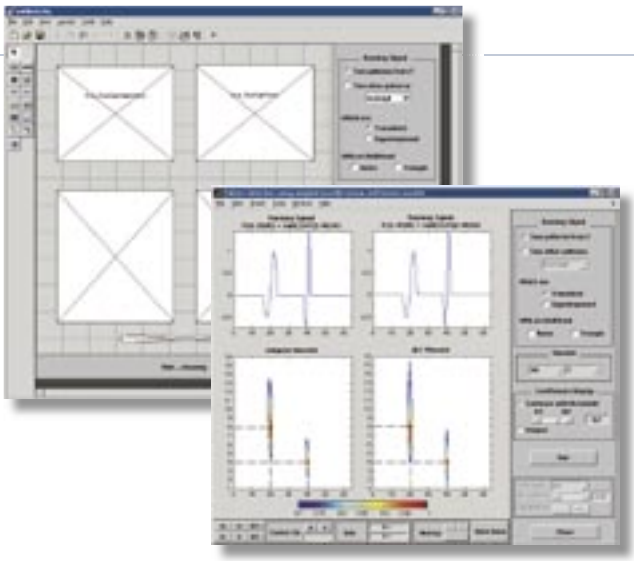
M-Lint Code Checker—Analyzes your code and recommends changes to improve its performance and maintainability

MATLAB Profiler—Records the time spent executing each line of code

Directory Reports—Scan all the files in a directory and report on code efficiency, file differences, file dependencies, and code coverage



An M-Lint Code Checker report that includes recommendations for making the code faster and easier to maintain.



GUIDE layout of a wavelet analysis GUI (top) together with the completed interface (bottom).

HDF data from a satellite, selected and imported into MATLAB using the MATLAB HDF Import Tool.

Designing Graphical User Interfaces

You can use the interactive tool GUIDE (Graphical User Interface Development Environment) to lay out, design, and edit user interfaces. GUIDE lets you include list boxes, pull-down menus, push buttons, radio buttons, and sliders, as well as MATLAB plots and ActiveX controls. Alternatively, you can create GUIs programmatically using MATLAB functions.

Analyzing and Accessing Data

MATLAB supports the entire data analysis process, from acquiring data from external devices and databases, through preprocessing, visualization, and numerical analysis, to producing presentation-quality output.

Data Analysis

MATLAB provides interactive tools and command-line functions for data analysis operations, including:

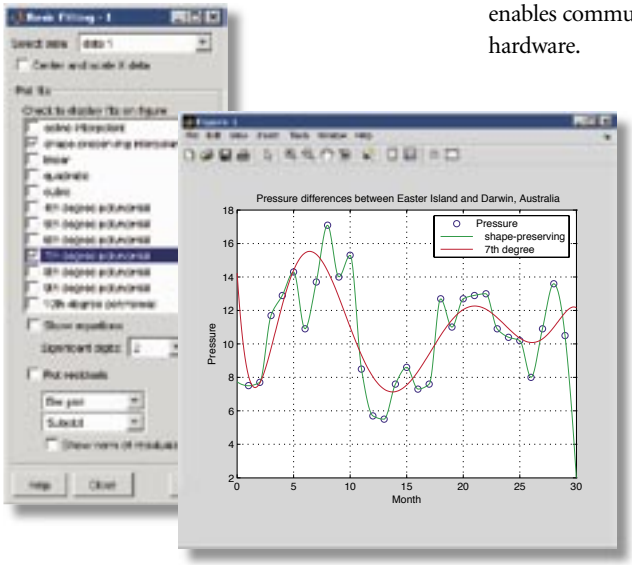
- Interpolating and decimating
- Extracting sections of data, scaling, and averaging
- Thresholding and smoothing
- Correlation, Fourier analysis, and filtering
- 1-D peak, valley, and zero finding
- Basic statistics and curve fitting
- Matrix analysis

Data Access

MATLAB is an efficient platform for accessing data from files, other applications, databases, and external devices. You can read data from popular file formats, such as Microsoft Excel; ASCII text or binary files; image, sound, and video files; and scientific files, such as HDF and HDF5. Low-level binary file I/O functions let you work with data files in any format. Additional functions let you read data from Web pages and XML.

You can call other applications and languages, such as C, C++, COM objects, DLLs, Java, Fortran, and Microsoft Excel, and access FTP sites and Web services. Using the Database Toolbox, you can also access data from ODBC/JDBC-compliant databases.

You can acquire data from hardware devices, such as your computer's serial port or sound card. Using the Data Acquisition Toolbox, you can stream live, measured data directly into MATLAB for analysis and visualization. The Instrument Control Toolbox enables communication with GPIB and VXI hardware.



Plot showing curve fitted to the monthly averaged atmospheric pressure differences between Easter Island and Darwin, Australia.

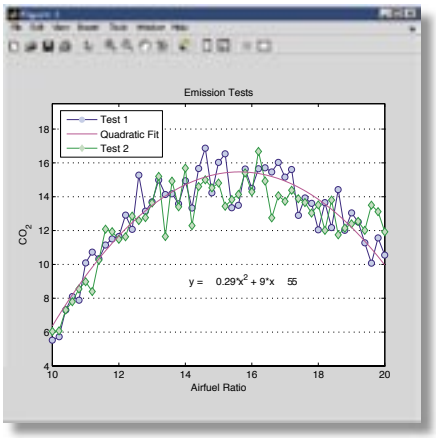
Visualizing Data

All the graphics features that are required to visualize engineering and scientific data are available in MATLAB. These include 2-D and 3-D plotting functions, 3-D volume visualization functions, tools for interactively creating plots, and the ability to export results to all popular graphics formats. You can customize your plots by adding multiple axes; changing line colors and markers; adding annotation, LaTEX equations, and legends; and drawing shapes.

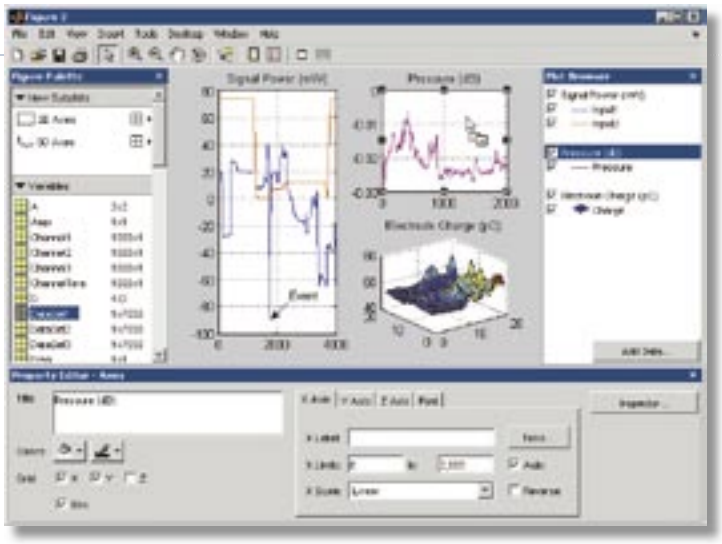
2-D Plotting

You can visualize vectors of data with 2-D plotting functions that create:

- Line, area, bar, and pie charts
- Direction and velocity plots
- Histograms
- Polygons and surfaces
- Scatter/bubble plots
- Animations



Line plots of multiple engine emission test results, with a curve fitted to the raw data.

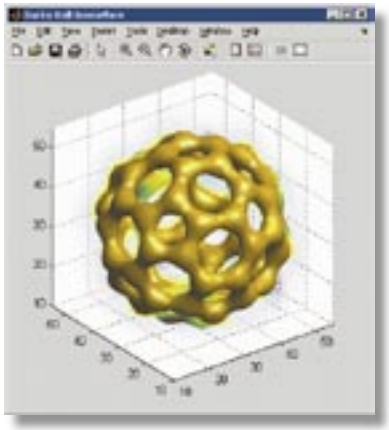


A collection of graphs, constructed interactively by dragging data sets onto the plot window, creating new subplots, changing properties such as colors and fonts, and adding annotation.

3-D Plotting and Volume Visualization

MATLAB provides functions for visualizing 2-D matrices, 3-D scalar, and 3-D vector data. You can use these functions to visualize and understand large, often complex, multidimensional data. You can specify plot characteristics, such as camera viewing angle, perspective, lighting effect, light source locations, and transparency. 3-D plotting functions include:

- Surface, contour, and mesh
- Image plots
- Cone, slice, stream, and isosurface



A 3-D isosurface plot revealing the geodesic dome structure of a carbon-60 fullerene molecule.

Creating and Editing Plots Interactively

MATLAB provides interactive tools for designing and modifying graphics. From a MATLAB figure window, you can perform the following tasks:

- Drag and drop new data sets onto the figure
- Change the properties of any object on the figure
- Zoom, rotate, pan, and change camera angle and lighting
- Add annotations and data tips
- Draw shapes
- Generate an M-code function that can be reused with different data

Importing and Exporting Graphic Files

MATLAB lets you read and write common graphical and data file formats, such as GIF, JPEG, BMP, EPS, TIFF, PNG, HDF, AVI, and PCX. As a result, you can export MATLAB plots to other applications, such as Microsoft Word and Microsoft PowerPoint, or to desktop publishing software. Before exporting, you can create and apply style templates, covering characteristics such as layout, font, and line thickness, to meet publication specifications.

Performing Numeric Computation

MATLAB contains mathematical, statistical, and engineering functions to support all common engineering and science operations. These functions, developed by experts in mathematics, are the foundation of the MATLAB language. The core math functions use the LAPACK and BLAS linear algebra subroutine libraries and the FFTW Discrete Fourier Transform library. Because these processor-dependent libraries are optimized to the different platforms that MATLAB supports, they execute faster than the equivalent C or C++ code.

MATLAB provides the following types of functions for performing mathematical operations and analyzing data:

- Matrix manipulation and linear algebra
- Polynomials and interpolation
- Fourier analysis and filtering
- Data analysis and statistics
- Optimization and numerical integration
- Ordinary differential equations
- Partial differential equations
- Sparse matrix operations

MATLAB can perform arithmetic on a wide range of data types, including doubles, singles, and integers.

Add-on toolboxes (available separately) provide specialized mathematical computing functions for areas including signal processing, optimization, statistics, symbolic math, partial differential equation solving, and curve fitting.

Publishing Results and Deploying Applications

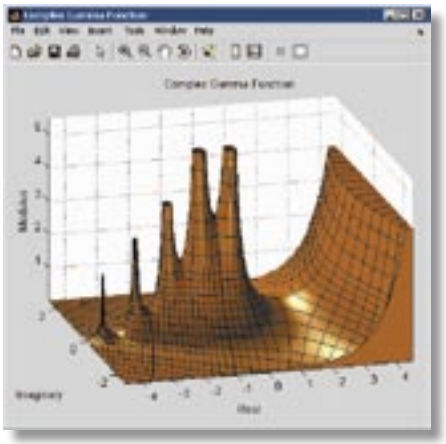
MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications and deploy your MATLAB algorithms and applications as stand-alone programs or software modules.

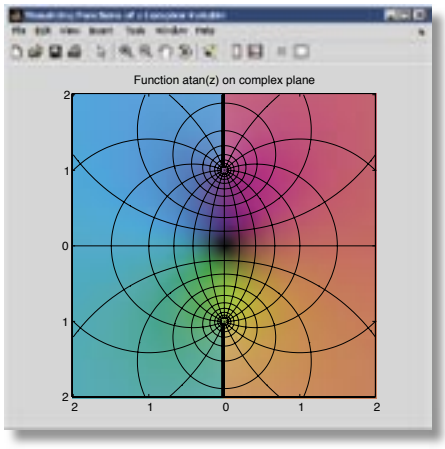
Publishing Results

MATLAB lets you export your results as plots or as complete reports. You can export plots to all popular graphics file formats and then import them into other packages, such as Microsoft Word or Microsoft PowerPoint. Using the MATLAB Editor, you can automatically publish your MATLAB code in HTML, Word, LaTeX, and other formats.

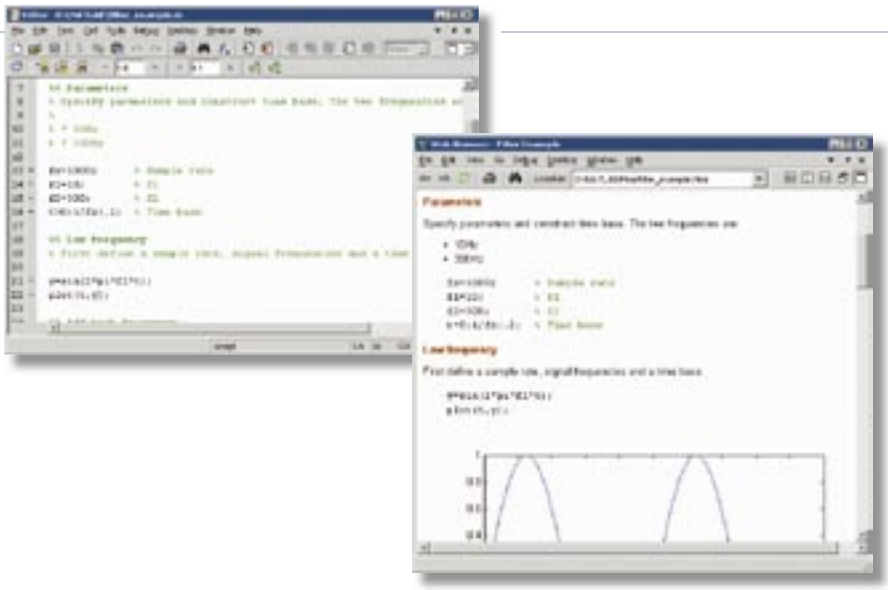
To create more complex reports, such as simulation runs and multiple parameter tests, you can use the MATLAB Report Generator (available separately).

Plot showing the complex valued gamma function on the complex plane, where the height of the surface is the modulus, or absolute value, and the contour lines are modulus and phase.





Plot of complex function `atan(z)`. Contour lines for the real and imaginary parts are superimposed on a color image showing magnitude and phase.



M-code program (left) published to HTML (right) using the MATLAB Editor. Results output to the command window or to plots are captured and included, and the comments are turned into section headings and body text in the HTML.

Integrating MATLAB Code with Other Languages and Applications

MATLAB provides functions for integrating C and C++ code, Fortran code, COM objects, and Java code with your applications. You can call DLLs, Java classes, and ActiveX controls. Using the MATLAB engine library, you can also call MATLAB from C, C++, or Fortran code.

Deploying Applications

You can create your algorithm in MATLAB and distribute it to other MATLAB users as M-code.

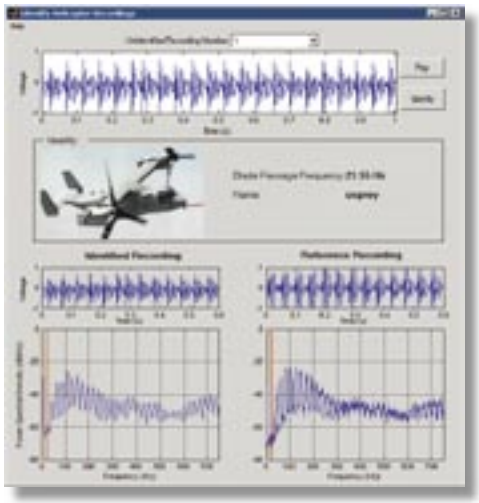
Using the MATLAB Compiler (available separately), you can deploy your algorithm, as a stand-alone application or as a software module that you include in your project, to users who do not have MATLAB.

Additional products let you convert your algorithm into a software module that is callable from COM, or Microsoft Excel.

Platform and System Requirements

For platform and system requirements, visit *www.mathworks.com/products/matlab*

Helicopter sound identification application, deployed and running outside MATLAB. The application, developed in MATLAB, directly acquires signals from measurement hardware, performs analysis and plotting, and includes GUI controls.



For demos, application examples, tutorials, user stories, and pricing:

- Visit **www.mathworks.com**
- Contact The MathWorks directly

US & Canada	508-647-7000
Benelux	+31 (0)182 53 76 44
France	+33 (0)1 41 14 67 14
Germany	+49 (0)241 470 750
Italy	+39 (011) 2274 700
Korea	+82 (0)2 6006 5114
Spain	+34 93 362 13 00
Sweden	+46 (8)505 317 00
Switzerland	+41 (0)31 950 60 20
UK	+44 (0)1223 423 200

Visit **www.mathworks.com** to obtain contact information for authorized MathWorks representatives in countries throughout Asia Pacific, Latin America, the Middle East, Africa, and the rest of Europe.

The MathWorks Tel: 508.647.7000 info@mathworks.com www.mathworks.com

91199v00 05/04