Introduction to MATLAB

Enabling Your Computational Science and Engineering Solutions

👤 Dr. Daniel R. Herber

✉ daniel.herber@colostate.edu
🏛 Colorado State University – Department of Systems Engineering

➜ Outline

①

Getting Started at CSU

→ What is MATLAB?

- MATLAB is an abbreviation for "matrix laboratory"

  "MATLAB is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world. The heart of MATLAB is the MATLAB language, a matrix-based language allowing the most natural expression of computational mathematics." — *The MathWorks*[1]

- This includes tasks like analyzing data, developing algorithms, and creating models and applications for a wide range of areas, industries, and disciplines[2]

- Originally simple terminal applications made into a commercial product in the 1980's with a desktop interface coming in 2000[3]

---

[1] 🔗 www.mathworks.com/discovery/what-is-matlab.html [2] 🔗 www.mathworks.com/solutions.html [3] See a Brief History of MATLAB at 🔗 www.mathworks.com/company/technical-articles/a-brief-history-of-matlab.html

➜ Example MATLAB Script

```matlab
1 clear; % removes all variables from workspace
2
3 A = [8,1,6;3,5,7;4,9,2] % make matrix
4 A = A.^2 - 1; % matrix math
5 func = @(in) in.^2 - 1; % anonymous function
6 A = func(A); % call function
7 b = ones(3,1); % ones vector
8 b(2) = pi; % change 2nd entry
9 x = A\b; % solve linear system
10
11 figure('Units','inches','Position',[10 8 1 2]);
12 bar(x); % draw bar graph
13 ha = gca; ha.FontSize = 8; % axis font size
14 xlabel('Index','FontSize',10) % add label
15 exportgraphics(gcf,'myfigure.pdf') % save PDF
```
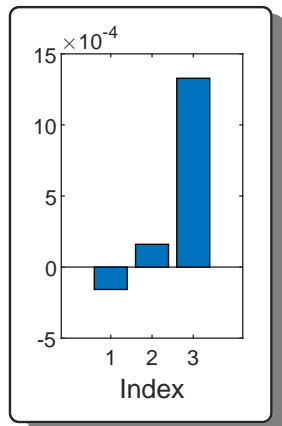
→ MATLAB Access at CSU — You Can Obtain It for Free!

"College of Engineering participates in a campus-wide MATLAB license that allows for virtually unlimited use of the MATLAB application and included toolboxes" — CSU ETS[1]

"Most faculty, staff and students with a current CSU eID are eligible to download and install MATLAB and toolboxes ON PERSONALLY-OWNED computers."

[1] 🔗 www.engr.colostate.edu/ets/matlab

➙ Obtaining MATLAB and Accompanying Products

1. *Local Installation* 🖥

    • You can install MATLAB, Simulink, and accompanying toolboxes on your personal computer for Windows, Linux, and Mac platforms

    • Engineering Technology Services (ETS) provides great CSU-specific instructions at the link below[1]

    • Also see the CSU MATLAB Access portal page[2]

    • See the link below for general installation guidance[3]

2. *Web-based Option* 🌐

    • MATLAB Online provides access to the latest version of MATLAB from a web browser using MathWorks computing resources[4]

    Remark 💬 | This option can be great for usage in a course or initial exploration but I strongly recommend the local installation for research.

---

[1] 🔗 www.engr.colostate.edu/ets/matlab
[2] 🔗 www.mathworks.com/academia/tah-portal/colorado-state-university-40638290.html
[3] 🔗 www.mathworks.com/help/install/ug/install-products-with-internet-connection.html
[4] 🔗 www.mathworks.com/help/matlab-online-server/ug/matlab-online.html

➜ [Read Later!] Versions

> Remark
> 💬
>
> Generally, you should install the most recent version available unless you have a specific reason for selecting an older version.

- Follows a twice-yearly general release schedule — one release around March and a second around September[1]
- Release name consists of the calendar year followed by "a" for the first release of the year, or "b" for the second
- For recent releases (2018b –), see the MATLAB Release Notes webpage[2]
- List of bugs can be found at the link below[3]

---

[1] 🔗 www.mathworks.com/products/new_products/release_model.html

[2] 🔗 www.mathworks.com/help/matlab/release-notes.html

[3] 🔗 www.mathworks.com/support/bugreports

➜ Recommended Starting Resources

- 2-hour MATLAB onramp course[1]
- 2-hour Simulink onramp course[2]
- Desktop interface basics[3]
- Language learning:
    - Webpage-based tutorial[4]
    - My *single file* MATLAB syntax overview[5]
    - Pick a book from the Introductory list on Slide 35
- MathWorks "Get Started with MATLAB" page and list of tutorials[6]

Upcoming
▶▶ See the Resources section for more online tutorials, books, and other resources.

---

[1] 🔗 matlabacademy.mathworks.com/details/matlab-onramp/gettingstarted
[2] 🔗 matlabacademy.mathworks.com/details/simulink-onramp/simulink
[3] 🔗 www.mathworks.com/help/matlab/learn_matlab/desktop.html
[4] 🔗 www.tutorialspoint.com/matlab
[5] 🔗 ex_matlab_basics.m on GitHub
[6] 🔗 www.mathworks.com/help/matlab/getting-started-with-matlab.html

②

MATLAB Language Highlights

➜ Language Fundamentals

- Language fundamentals[1] includes 1) entering commands, 2) matrices and arrays, 3) data types, 4) operators and elementary operations, and 5) loops and conditional statements
- Uses 1-based indexing and is case-sensitive
- A variable name must begin with a letter which may be followed by letters, digits, or underscores (no other special character allowed)
- Add comments with `%`, suppress output with `;`, and continue lines with `...`

```
1 a = [1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];
2 a(2,5) % reference the 2nd row and 5th column of a
3 v = a(:,4) % extract 4th column of a
4 q = a(2:3,:) % extract 2nd and 3rd rows of a
5 a(4,:) = [] % delete 4th row of a
6 a(:,[1 3]) = [] % delete 1st and 3rd columns of a
7 a(end,:) % last row of a
8 a(:,end-1:end) % last and 2nd to last columns of a
```

[1] 🔗 www.mathworks.com/help/matlab/language-fundamentals.html and 🔗 ex_matlab_basics.m on GitHub

➡ Programming (Scripts, Functions, Classes, Files, and Folders)

- We often need to go beyond only typing commands into the Command Window[1]
- Several paradigms for repeatable and callable code:
    - Scripts — contains a set of commands exactly as you would type them at the command line
    - Functions — accept inputs and return outputs with `function` keyword
    - Live scripts and live functions — useful for interacting with a sequence of commands (similar to Python, R, and Wolfram Notebooks)
    - Classes — use object-oriented programming techniques
- Programming utilities for indirectly evaluating expressions, setting timers for program execution, capturing data on errors, and handling exceptions[2]
- Various file operations and file path construction functions[3]
    - Need to consider where MATLAB looks for files, the "search path"[4]

---

[1] 🔗 www.mathworks.com/help/matlab/programming-and-data-types.html

[2] 🔗 www.mathworks.com/help/matlab/code-execution.html and 🔗 www.mathworks.com/help/matlab/exception-handling.html

[3] 🔗 www.mathworks.com/help/matlab/files-and-folders.html

[4] 🔗 www.mathworks.com/help/matlab/matlab_env/files-and-folders-that-matlab-accesses.html

→ Data Import and Analysis

- Variety of ways to access, explore, visualize, and analyze data[1]
  - Data import and export (text files, spreadsheets, custom formats, hardware, web access, large files, big data, and more)
  - Preprocessing data (data cleaning, smoothing, grouping)
  - Descriptive statistics (range, central tendency, standard deviation, variance, correlation)
  - Visual exploration
- This example reads a block of mixed text and numeric data from a text file:

```
1 type('bigfile.txt')
2 opts = detectImportOptions('bigfile.txt');
3 opts.DataLines = [3 8];
4 opts.VariableNames = {'Timestamp','Temp',...
5                       'Humidity','Wind','Weather'};
6 T_first = readtable('bigfile.txt',opts)
7 plot(T_first.Temp,T_first.Humidity,'.')
```

---

[1] 🔗 www.mathworks.com/help/matlab/data-import-and-analysis.html

➜ Mathematics

- Math functions provide a range of numerical computation methods for analyzing data, developing algorithms, and creating models[1]
- Key areas include 1) elementary math, 2) linear algebra, 3) random number generation, 4) interpolation, 5) optimization, 6) numerical integration and differential equations, 7) Fourier analysis and filtering, 8) sparse matrices, 9) graph and network algorithms, 10) computational geometry, and 11) quantum computing
- Find numerically when $\sin(x) = 0$ starting at $x_0 = 3$:

```
1 fun = @sin; % function
2 x0 = 3; % initial point
3 x = fzero(fun,x0)
```

---

[1] 🔗 www.mathworks.com/help/matlab/mathematics.html

➡ Graphics and Plotting

- Includes 2-D and 3-D plotting functions to visualize data and communicate results[1] with a great overview of the types below[2]

Line Plots	Scatter and Bubble Charts	Data Distribution Plots	Discrete Data Plots	Geographic Plots	Polar Plots	Contour Plots	Vector Fields	Surface and Mesh Plots	Volume Visualization	Animation	Images
plot	scatter	histogram	bar	geoplot	polarplot	contour	quiver	surf	streamline	animatedline	image
plot3	scatter3	histogram2	barh	geoscatter	polarhistogram	contourf	quiver3	surfc	streamslice	comet	imagesc
stairs	bubblechart	scatterhistogram	bar3	geobubble	polarscatter	contour3	feather	surfl	streamparticles	comet3	
errorbar	bubblechart3	boxchart	bar3h		polarbubblechart	contourslice		ribbon	streamribbon		
area	swarmchart	swarmchart	pareto		compass	fcontour		pcolor	streamtube		
stackedplot	swarmchart3	swarmchart3	stem		fpolarplot			fsurf	coneplot		

More in this table! See the link below

→ [Read Later!] Relative Location of the Current File

> Remark ● A common issue I see is using an absolute path rather than a relative path within your MATLAB code.

- Say I have the following script, and I want to save the figure always in the *same* location as the script
- Then consider using the following example:

```
1 fullPath = which(mfilename('fullpath')); % current location and
      file name
2 localPath = fullfile(fileparts(fullPath), filesep); % location only
3 figureName = "myfig.pdf"; % desired figure name
4 exportgraphics(gca,fullfile(localPath,figureName)); % save pdf
```

➜ [Read Later!] External Interfaces and Integrations

- Provides a flexible, two-way integration with other programming languages[1]
    - Allows you to reuse legacy (or current) code
- Call functions and using objects natively created in C++, C, Java, Python, .NET, and COM
- Call MATLAB from another programming language, including APIs for C++, C, Java, Python, and Fortan
- Communicate with web services (RESTful, WSDL, HTTP interfaces)
- Use the `system` command for programmatic access to your operating system, running of external programs, and more[2]
- Native integration with Magic Model Analyst (Cameo Simulation Toolkit) in Systems of Systems Architect (Cameo Systems Modeler)[3]

---

[1] 🔗 www.mathworks.com/help/matlab/external-language-interfaces.html and
🔗 www.mathworks.com/help/matlab/matlab_external/integrate-matlab-with-external-programming-languages-and-systems.html
[2] 🔗 www.mathworks.com/help/matlab/matlab-environment-control.html
[3] 🔗 docs.nomagic.com/display/MSI2022xR2/Integration+with+MATLAB

➜ [Read Later!] Software Development Tools

- Provides capabilities to support collaborative software development practices[1]
- Can integrate your MATLAB files with Git or Subversion source control systems[2]
- Features to test the functionality[3] and performance[4] of your code
- To share code with others, package projects[5] or other files as a toolbox[6]

---

[1] 🔗 www.mathworks.com/help/matlab/software-development.html
[2] 🔗 www.mathworks.com/help/matlab/source-control.html
[3] 🔗 www.mathworks.com/help/matlab/matlab-unit-test-framework.html
[4] 🔗 www.mathworks.com/help/matlab/performance-and-memory.html
[5] 🔗 www.mathworks.com/help/matlab/projects.html
[6] 🔗 www.mathworks.com/help/matlab/creating-help.html

→ [Read Later!] Environment, Settings, and Add-Ons

- Review general desktop preferences like appearance, accessibility, fonts, colors, and keyboard shortcuts[1]
- Consider add-ons, including optional features, apps, toolboxes, and support packages[2]
- Commands to understand your environment[3]

```
1 matlabRelease % current MATLAB release information
2 version % version number for MATLAB and libraries
3 matlab.addons.installedAddons % list of installed add-ons
4 matlabshared.supportpkg.getInstalled % installed support packages
5 license('inuse') % features checked out in the current session
6 computer % information about computer on which MATLAB is running
7 canUseGPU % verify supported GPU is available
8 canUseParallelPool % verify can use a parallel pool
```

---

[1] 🔗 www.mathworks.com/help/matlab/desktop.html
[2] 🔗 www.mathworks.com/help/matlab/add-ons.html
[3] 🔗 www.mathworks.com/help/matlab/matlab-version-and-license.html

➜ [Read Later!] Debugging and Profiling

- Diagnose problems and check syntax and release compatibility[1]
    - MATLAB automatically identifies potential coding problems
    - Can run-to-line, add breakpoints (conditionally), and display outputs to help debug[2]
- Profile code, improve performance, and reduce memory requirements[3]
    - Use the profiler[4] to create a *flame graph* showing visual representation of the time MATLAB spent running the profiled function
    - Avoid unnecessary copies of data[5]

---

[1] 𝒮 www.mathworks.com/help/matlab/debugging-code.html
[2] 𝒮 www.mathworks.com/help/matlab/matlab_prog/debugging-process-and-features.html
[3] 𝒮 www.mathworks.com/help/matlab/performance-and-memory.html
[4] 𝒮 www.mathworks.com/help/matlab/matlab_prog/profiling-for-improving-performance.html
[5] 𝒮 www.mathworks.com/help/matlab/matlab_prog/avoid-unnecessary-copies-of-data.html

③

Toolboxes Overview and Highlights

➜ MATLAB Toolboxes Provide More Functionality

- MATLAB *toolboxes*[1] are designed to help solve specific types of problems and are often used to streamline and simplify complex tasks within various fields of engineering, science, and mathematics
- Easiest way to obtain the toolboxes is to select them during installation[2]
- Can also use the Add-On Explorer[3]

---

[1] 🔗 www.mathworks.com/products.html
[2] 🔗 www.mathworks.com/matlabcentral/answers/98886
[3] 🔗 www.mathworks.com/videos/add-on-explorer-106745.html

➜ 109 Toolboxes Available at CSU

#	Toolbox		#	Toolbox		#	Toolbox		#	Toolbox
1	5G Toolbox		29	HDL Verifier		57	Powertrain Blockset		85	Simulink Coverage
2	Aerospace Blockset		30	Image Acquisition Toolbox		58	Predictive Maintenance Blockset		86	Simulink Design Optimization
3	Aerospace Toolbox		31	Image Processing Toolbox		59	Reinforcement Learning Toolbox		87	Simulink Design Verifier
4	Antenna Toolbox		32	Instrument Control Toolbox		60	RF Blockset		88	Simulink Desktop Real-Time
5	Audio Toolbox		33	Lidar Toolbox		61	RF Toolbox		89	Simulink PLC Coder
6	Automated Driving Toolbox		34	LTE Toolbox		62	Risk Management Toolbox		90	Simulink Real-Time
7	AUTOSAR Blockset		35	Mapping Toolbox		63	Roadrunner		91	Simulink Report Generator
8	Bioinformatics Toolbox		36	Online Training Suite		64	Roadrunner Asset Library		92	Simulink Requirements
9	Communications Toolbox		37	MATLAB Compiler SDK		65	Robotics System Toolbox		93	Simulink Test
10	Computer Vision Toolbox		38	MATLAB Coder		66	Robust Control Toolbox		94	SoC Blockset
11	Control System Toolbox		39	MATLAB Compiler		67	ROS Toolbox		95	Spreadsheet Link
12	Curve Fitting Toolbox		40	MATLAB Grader LMS Support		68	Sensor Fusion and Tracking Toolbox		96	Stateflow
13	Data Acquisition Toolbox		41	MATLAB Parallel Server		69	SerDes Toolbox		97	Statistics and Machine Learning Toolbox
14	Database Toolbox		42	MATLAB Production Server		70	Signal Processing Toolbox			
15	Datafeed Toolbox		43	MATLAB Report Generator		71	SimBiology		98	Symbolic Math Toolbox
16	Deep Learning Toolbox		44	MATLAB Web App Server		72	Simscape		99	System Composer
17	Deep Learning HDL Toolbox		45	Mixed-Signal Blockset		73	Simscape Driveline		100	System Identification Toolbox
18	DSP System Toolbox		46	Model Predictive Control Toolbox		74	Simscape Electrical		101	Text Analytics Toolbox
19	Econometrics Toolbox		47	Model-Based Calibration Toolbox		75	Simscape Fluids		102	Trading Toolbox
20	Embedded Coder		48	Motor Control Blockset		76	Simscape Multibody		103	UAV Toolbox
21	Filter Design HDL Coder		49	Navigation Toolbox		77	SimEvents		104	Vehicle Dynamics Blockset
22	Financial Instruments Toolbox		50	OPC Toolbox		78	Simulink		105	Vehicle Network Toolbox
23	Financial Toolbox		51	Optimization Toolbox		79	Simulink 3D Animation		106	Vision HDL Toolbox
24	Fixed-Point Designer		52	Parallel Computing Toolbox		80	Simulink Check		107	Wavelet Toolbox
25	Fuzzy Logic Toolbox		53	Partial Differential Equation Toolbox		81	Simulink Code Inspector		108	Wireless HDL Toolbox
26	Global Optimization Toolbox		54	Phased Array System Toolbox		82	Simulink Coder		109	WLAN Toolbox
27	GPU Coder		55	Polyspace Bug Finder		83	Simulink Compiler			
28	HDL Coder		56	Polyspace Code Prover		84	Simulink Control Design			

� ➙ Highlighted Toolboxes (1): General

- *Simulink* — simulation and model-based design with block diagrams[1]
  - ➤ Thermal Model of a House
        openExample('simulink_general/sldemo_househeatExample')
- *MATLAB Compiler* — build standalone executables and web apps from MATLAB programs[2]
  - ➤ Create Standalone Application from MATLAB Function

        openExample('compiler/
    CreateStandaloneApplicationFromMATLABFunctionExample')
- *Parallel Computing Toolbox* — perform parallel computations on multicore computers, GPUs, and computer clusters[3]
  - ➤ Run MATLAB Functions on Multiple GPUs

        openExample('parallel/MultiGPUExample')
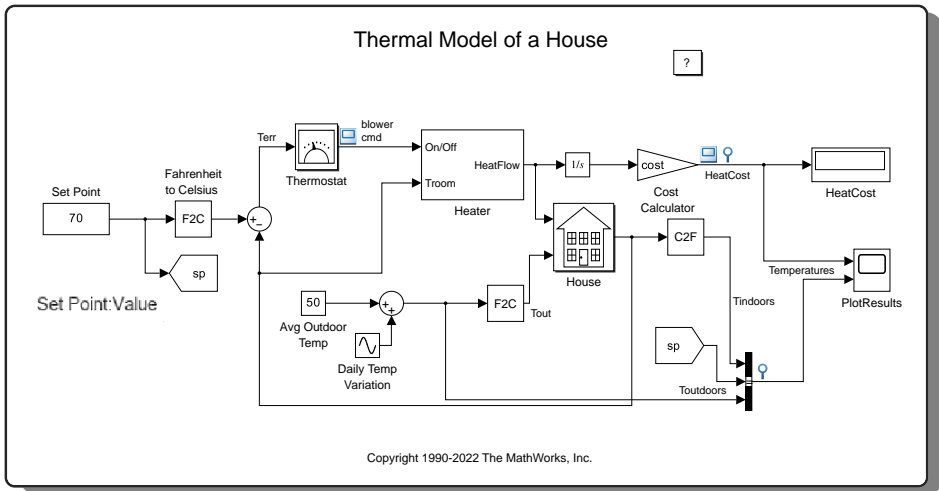
---

[1] 🔗 www.mathworks.com/help/simulink and 🔗 example link
[2] 🔗 www.mathworks.com/help/compiler and 🔗 example link
[3] 🔗 www.mathworks.com/help/parallel-computing and 🔗 example link

→ Highlighted Toolboxes (2): Simulink Example

Thermal Model of a House

Copyright 1990-2022 The MathWorks, Inc.

➜ Highlighted Toolboxes (3): AI, Data Science, and Statistics

- *Curve Fitting Toolbox* — fit curves and surfaces to data using regression, interpolation, and smoothing[1]

  ↪ Cubic Smoothing Splines

  ```
  openExample('curvefit/csapsdem')
  ```

- *Statistics and Machine Learning Toolbox* — analyze and model data using t-tests, neural networks, k-means classification, and more[2]

  ↪ Two-Way ANOVA

  ```
  openExample('stats/TwoWayANOVAForUnbalancedDesignExample')
  ```

- *Deep Learning Toolbox* — design, train, and analyze deep learning networks, including convolutional neural networks (CNNs), long short-term memory (LSTM) networks, generative adversarial networks (GANs), and more [3]

  ↪ Classify Image Using GoogLeNet

  ```
  openExample('nnet/ClassifyImageUsingGoogLeNetExample')
  ```

---

[1] 🔗 www.mathworks.com/help/curvefit and 🔗 example link

[2] 🔗 www.mathworks.com/help/stats/index.html and 🔗 example link

[3] 🔗 www.mathworks.com/help/deeplearning/index.html and 🔗 example link

➙ Highlighted Toolboxes (4): Systems Engineering

- *System Composer* — design and analyze system and software architectures as an MBSE tool (but not SysML compliant)[1]

  ↪ Modeling System Architecture of Small UAV

  ```
  openExample('shared_systemsengineering/
  ModelingSystemArchitectureOfSmallUAVExample')
  ```

- *Simulink Requirements* — author, link, and validate requirements for designs and tests[2]

  ↪ Import Requirements from a Microsoft Excel Document

  ```
  openExample('slrequirements/
  ProcessRequirementsImportedFromMicrosoftExcelExample')
  ```

- *Simulink Test* — develop, manage, and execute simulation-based tests[3]

  ↪ Test Downshift Points of a Transmission Controller

  ```
  openExample('simulinktest/
  TransmissionDownshiftTestSequenceExample')
  ```

[1] 𝒮 www.mathworks.com/help/systemcomposer and 𝒮 example link
[2] 𝒮 www.mathworks.com/help/slrequirements and 𝒮 example link
[3] 𝒮 www.mathworks.com/help/sltest and 𝒮 example link

➡ Highlighted Toolboxes (5): Mathematics and Optimization

- *Symbolic Math Toolbox* — perform symbolic math computations[1]
  - ➴ Evaluating the Average Power Delivered by a Wind Turbine

    `openExample('symbolic/AvgWindTurbinePower')`
- *Optimization Toolbox* — solve linear, quadratic, conic, integer, and nonlinear optimization problems[2]
  - ➴ Factory, Warehouse, Sales Allocation Model: Problem-Based

    `openExample('optim/FactoryExample')`
- *Global Optimization Toolbox* — solve multiple maxima, multiple minima, and nonsmooth optimization problems[3]
  - ➴ Constrained Minimization Using the Genetic Algorithm

    `openExample('globaloptim/gaconstrained')`

---

[1] 🔗 www.mathworks.com/help/symbolic and 🔗 example link
[2] 🔗 www.mathworks.com/help/optim and 🔗 example link
[3] 🔗 www.mathworks.com/help/gads and 🔗 example link

→ Highlighted Toolboxes (6): Physical Modeling

- *Simscape* — model and simulate multidomain physical systems[1]
  - ➡ Lithium Battery Cell - Two RC-Branch Equivalent Circuit

    ```
    openExample('simscape/
    LithiumBatteryCellTwoRCBranchEquivalentCircuitExample')
    ```

- *Simscape Electrical* — model and simulate electronic, mechatronic, and electrical power systems[2]
  - ➡ Asynchronous Machine Direct Torque Control

    ```
    openExample('simscapeelectrical/InductionMachineDTCExample')
    ```

- *Simscape Driveline* — model and simulate rotational and translational mechanical systems[3]
  - ➡ Single Cylinder Spark Ignition Engine

    ```
    openExample('sdl/SingleCylinderSparkIgnitionEngineExample')
    ```

---

[1] 🔗 www.mathworks.com/help/simscape and 🔗 example link
[2] 🔗 www.mathworks.com/help/sps and 🔗 example link
[3] 🔗 www.mathworks.com/help/sdl and 🔗 example link

➡ Highlighted Toolboxes (7): Control Systems

- *Control System Toolbox* — design and analyze control systems[1]
  - ↪ Design PID Controller for Disturbance Rejection Using PID Tuner

    `openExample('control/isapiddemo')`
- *Robust Control Toolbox* — frequency-domain MIMO controller design, controller design for uncertain systems[2]
  - ↪ Robust Control of Active Suspension

    `openExample('robust/ActiveSuspensionExample')`
- *System Identification Toolbox* — create linear and nonlinear dynamic system models from input-output data[3]
  - ↪ Estimating Simple Models from Real Laboratory Process Data

    `openExample('ident/iddemo1')`

---

[1] 🔗 www.mathworks.com/help/control and 🔗 example link
[2] 🔗 www.mathworks.com/help/robust and 🔗 example link
[3] 🔗 www.mathworks.com/help/ident and 🔗 example link

➙ Highlighted Toolboxes (8): Automotive

- *Vehicle Dynamics Blockset* — model and simulate vehicle dynamics in a virtual 3D environment[1]

  ➙ Three-Axle Tractor Towing a Three-Axle Trailer

  ```
  openExample('vdynblks/TrctrExample')
  ```

- *Automated Driving Toolbox* — design, simulate, and test Advanced Driver Assistance Systems (ADAS) and autonomous driving systems[2]

  ➙ Adaptive Cruise Control with Sensor Fusion

  ```
  openExample('autonomous_control/
  AdaptiveCruiseControlWithSensorFusionExample')
  ```

- *Vehicle Network Toolbox* — communicate with in-vehicle networks using CAN, J1939, and XCP protocols[3]

  ➙ Decode CAN Data from BLF-Files

  ```
  openExample('vnt/DecodingCANDataFromBLFFilesExample')
  ```

[1] 🔗 www.mathworks.com/help/vdynblks and 🔗 example link
[2] 🔗 www.mathworks.com/help/driving and 🔗 example link
[3] 🔗 www.mathworks.com/help/vnt and 🔗 example link

④

Discussion

➙ Advantages (1): Extensive Capabilities and Interoperability

1. *Extensive capabilities* — already capable of helping with many tasks; revisit the language and toolbox highlights

2. *Rapid development and prototyping* — easy to get started on simple and complex projects

   • Flexible native and toolbox capabilities; desktop platform independence; interpreted language; easy debugging and profiling

3. *Internal interoperability* — can readily combine advanced technical concepts and functions together without much need for adaptations or middleware

   • Combine together requirements, symbolic math computations, graph theory, Simscape physical models, control techniques, genetic algorithm for optimization, and data analysis and graphics tools[1]

4. *Interoperability with other languages* — can call functions and objects from several other languages (and vice versa)[2]

---

[1] One of my papers that includes many of these things 🔗 link   [2] See Slide 15

→ Advantages (2): Support and Availability

5. *Community support and resources* — great documentation, resources, and support as a commercial product[1]; also, good open-source community sharing new capabilities and toolboxes[2]

6. *Educational tool* — effective tool for teaching and learning various concepts; general intuitive syntax and extensive documentation

7. *Availability at CSU* — can be easily obtained at CSU[3]

---

[1] See Resources section   [2] See Slide 38   [3] See Slide 6

➜ Challenges (1): Closed and Paid Product

1. *Commercial product* — not free to obtain (equity issues) and potential license restrictions
   - If there are licenses available at a particular organization, they may be limited in both quantity and scope
2. *Not open source* — source code for many built-in functions is not available
   - Documentation is generally quite good, but no substitute for readable and editable source code
   - Sometimes, this is because the function is written in a lower-level language
3. *Proprietary language and data formats* — some choices within MATLAB/Simulink are decided by the company, not an open standards body, resulting in proprietary implementations
   - Often support major standards, but not always (e.g., SysML for system architecture modeling in System Composer in favor of a proprietary language)
   - Simulink `.slx` files adhere to Open Packaging Conventions (OPC) interoperability standard but can only be used within Simulink

➜ Challenges (2): Limitations

4. *Adoption* — limited and inconsistent adoption in salient areas and industries
   • Several metrics even indicate a slow decline[1]

5. *Slower evaluation time* — as an interpreted, higher-level language, code evaluation time can be slower than other languages (e.g., compiled languages)
   • Generally well positioned compared to similar alternatives though
   • Can be improved with a deeper understanding of language best practice[2] (as is the case with all languages) and compiling[3]

6. *Some (minor) graphics limitations* — certain features not supported, and still some issues with the now ten years old HG2 graphics engine
   • Cannot universally replace fonts without some potential issues
   • Only basic LaTeX support (no custom packages/commands)

---

[1] 🔗 asterisk.dynevor.org/popularity-of-python-and-matlab.html   [2] Great book on this topic: *Accelerating MATLAB Performance*   [3] You can generate C/C++ code from MATLAB code with `codegen` 🔗 www.mathworks.com/help/coder/ref/codegen.html

→ Alternatives to MATLAB/Simulink

- For a free, open-source, and mostly compatible with MATLAB language, consider GNU Octave[1]
- For a free, open-source, interpreted, and interactive language, consider Python[2], R[3], Julia[4], or Scilab[5]
- For block-diagram modeling, consider Modelica[6], Xcos[7], and Simcenter Amesim[8]
- For faster execution and memory advantages, consider compiled or lower-level languages

---

[1] 🔗 octave.org   [2] 🔗 www.python.org   [3] 🔗 www.r-project.org   [4] 🔗 julialang.org   [5] 🔗 www.scilab.org
[6] 🔗 modelica.org   [7] 🔗 www.scilab.org/software/xcos   [8] 🔗 plm.sw.siemens.com/en-US/simcenter/systems-simulation/amesim

⑤

Resources

→ Online Tutorials

1. Self-paced online courses provided by MathWorks[1]
    - For example, a MATLAB Fundamentals course[2]
    - Also, a machine learning onramp[3]
2. MathWorks Help Center videos[4]
3. MATLAB YouTube channel[5]
    - Check out the playlists, including "How To" guides
4. Webpage-based tutorial[6]
5. Free Vanderbilt University MATLAB Programming for Engineers and Scientists Specialization course[7]

[1] matlabacademy.mathworks.com
[2] matlabacademy.mathworks.com/details/matlab-fundamentals/mlbe
[3] matlabacademy.mathworks.com/details/machine-learning-onramp/machinelearning
[4] www.mathworks.com/support/search.html?fq%5B%5D=asset_type_name%3Avideo
[5] www.youtube.com/@MATLAB
[6] www.tutorialspoint.com/matlab/index.htm
[7] www.coursera.org/specializations/matlab-programming-engineers-scientists

→ Books (General)

- Introductory:
  - *MATLAB: An Introduction with Applications*
  - *MATLAB For Dummies*
  - *Beginning MATLAB and Simulink: From Beginner to Pro*
  - *R and MATLAB*
- Advanced:
  - *Accelerating MATLAB Performance*
  - *MATLAB Recipes: A Problem-Solution Approach*

Remark 💬

Links to the books with your CSU credentials are in the References on Slide 39. If you are having trouble with the provided links, search for the reference directly on 🔗 lib.colostate.edu.

→ Books (Specific Subjects)

- *Applied Numerical Methods Using Matlab*
- *Dynamic System Modeling and Analysis with MATLAB and Python: For Control Engineers*
- *Introduction to Nonlinear Optimization - Theory, Algorithms, and Applications with MATLAB*
- *Practical MATLAB Deep Learning: A Projects-Based Approach*
- *Machine and Deep Learning Using MATLAB: Algorithms and Tools for Scientists and Engineers*
- *Simulation of Power Electronics Circuits with MATLAB/Simulink: Design, Analyze, and Prototype Power Electronics*
- *Robotics, Vision and Control: Fundamental Algorithms In MATLAB*

Remark Links to the books with your CSU credentials are in the References on
💬   Slide 39. If you are having trouble with the provided links, search for
        the reference directly on 🔗 lib.colostate.edu.

➜ Answers to Your Questions

- Search the MATLAB documentation website[1]
  - If you know the function/method/class/toolbox/variable by `name`, you can use `doc name` to show the reference page in Help browser
  - Similarly, you can use `help name` for the help text in the Command Window
  - Also consider `docsearch`[2] and `lookfor`[3]
- Search and ask on MATLAB Answers[4]
- Search and ask on Stack Overflow[5] with the tag `[Matlab]`

?

---

[1] 🔗 www.mathworks.com/help/matlab
[2] 🔗 www.mathworks.com/help/matlab/ref/docsearch.html
[3] 🔗 www.mathworks.com/help/matlab/ref/lookfor.html
[4] 🔗 www.mathworks.com/matlabcentral/answers/index and 🔗 www.mathworks.com/matlabcentral/answers
[5] 🔗 stackoverflow.com/questions/tagged/matlab

�l Find Open-Source MATLAB Code

- Search on Mathworks File Exchange (FX)[1]
  - "File Exchange lets you find and share custom applications, classes, code examples, drivers, functions, Simulink models, scripts, and videos."[2]
- Search on GitHub with `MATLAB` language specified[3]
- Also consider searching on GitLab, BitBucket, and SourceForge

---

[1] 🔗 www.mathworks.com/matlabcentral/fileexchange
[2] 🔗 www.mathworks.com/matlabcentral/content/fx/about.html
[3] 🔗 github.com/search?q=language%3AMATLAB

→ References

📕 Y. M. Altman (2014). *Accelerating MATLAB Performance*. Chapman and Hall/CRC. URL: https://learning-oreilly-com.ezproxy2.library.colostate.edu/library/view/accelerating-matlab-performance/9781482211290/

📕 B. Amir (2014). *Introduction to Nonlinear Optimization - Theory, Algorithms, and Applications with MATLAB*. SIAM. URL: https://epubs-siam-org.ezproxy2.library.colostate.edu/doi/book/10.1137/1.9781611973655

📕 F. Asadi (2022). *Simulation of Power Electronics Circuits with MATLAB/Simulink: Design, Analyze, and Prototype Power Electronics*. Apress. URL: https://learning-oreilly-com.ezproxy2.library.colostate.edu/library/view/simulation-of-power/9781484282205

📕 P. Corke (2017). *Robotics, Vision and Control: Fundamental Algorithms In MATLAB*. 2nd ed. Springer. URL: https://link-springer-com.ezproxy2.library.colostate.edu/book/10.1007/978-3-319-54413-7

📕 S. Eshkabilov (2022). *Beginning MATLAB and Simulink: From Beginner to Pro*. Apress. URL: https://learning-oreilly-com.ezproxy2.library.colostate.edu/library/view/beginning-matlab-and/9781484287484/

📕 A. Gilat (2014). *MATLAB: An Introduction with Applications*. 5th ed. Wiley. URL: https://learning-oreilly-com.ezproxy2.library.colostate.edu/library/view/matlab-an-introduction/9781118629864/

📕 D. E. Hiebeler (2018). *R and MATLAB*. Chapman and Hall/CRC. URL: https://learning-oreilly-com.ezproxy2.library.colostate.edu/library/view/r-and-matlab/9781466568396/

➜ References (Continued)

∎ J. Kim (2022). *Dynamic System Modeling and Analysis with MATLAB and Python: For Control Engineers*. Wiley. URL: https://onlinelibrary-wiley-com.ezproxy2.library.colostate.edu/doi/book/10.1002/9781119801658

∎ K. Al-Malah (2023). *Machine and Deep Learning Using MATLAB: Algorithms and Tools for Scientists and Engineers*. Wiley. URL: https://onlinelibrary-wiley-com.ezproxy2.library.colostate.edu/doi/book/10.1002/9781394209118

∎ J. P. Mueller and J. Sizemore (2021). *MATLAB For Dummies*. 2nd ed. For Dummies. URL: https://learning-oreilly-com.ezproxy2.library.colostate.edu/library/view/matlab-for-dummies/9781119796886

∎ M. Paluszek and S. Thomas (2020). *MATLAB Recipes: A Problem-Solution Approach*. Apress. URL: https://link-springer-com.ezproxy2.library.colostate.edu/book/10.1007/978-1-4842-6124-8

∎ M. Paluszek, S. Thomas, and E. Ham (2022). *Practical MATLAB Deep Learning: A Projects-Based Approach*. 2nd ed. Apress. URL: https://learning-oreilly-com.ezproxy2.library.colostate.edu/library/view/practical-matlab-deep/9781484279120

∎ W. Y. Yang et al. (2020). *Applied Numerical Methods Using Matlab*. 2nd ed. Wiley. URL: https://onlinelibrary-wiley-com.ezproxy2.library.colostate.edu/doi/book/10.1002/9781119626879

Questions?

Introduction to MATLAB Enabling Your Computational Science and Engineering Solutions

Dr. Daniel R. Herber

→ How to Create the MATLAB Logo in MATLAB

```matlab
1  L = 160*membrane(1,100);
2  f = figure;
3  ax = axes;
4  s = surface(L);
5  s.EdgeColor = 'none';
6  view(3)
7  ax.XLim = [1 201];
8  ax.YLim = [1 201];
9  ax.ZLim = [-53.4 160];
10 ax.CameraPosition = [-145.5 -229.7 283.6];
11 ax.CameraTarget = [77.4 60.2 63.9];
12 ax.CameraUpVector = [0 0 1];
13 ax.CameraViewAngle = 36.7;
14 ax.Position = [0 0 1 1];
15 ax.DataAspectRatio = [1 1 .9];
16 l1 = light;
17 l1.Position = [160 400 80];
18 l1.Style = 'local';
19 l1.Color = [0 0.8 0.8];
20 l2 = light;
21 l2.Position = [.5 -1 .4];
22 l2.Color = [0.8 0.8 0];
23 s.FaceColor = [0.9 0.2 0.2];
24 axis off
25 exportgraphics(gcf,'matlab.png')
```