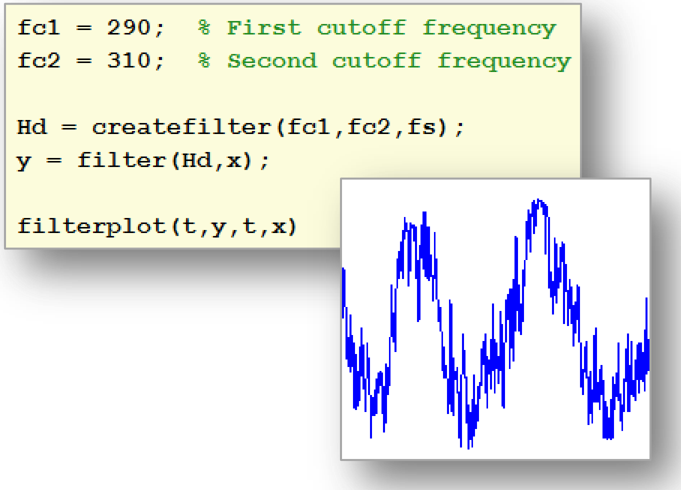
Programming with MATLAB

Sean de Wolski
Application Engineer

Today’s Objectives

- Introduce you to programming with MATLAB
  - The MATLAB language
  - Development tools
- Demonstrate the range of programming styles supported
  - Interactive command line, scripts and functions, object-oriented programming
- Show you how to program effectively in MATLAB

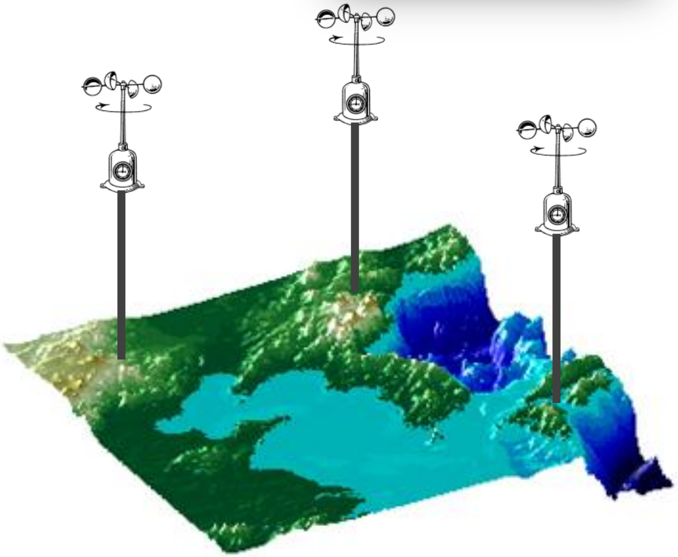


```
fc1 = 290;   % First cutoff frequency
fc2 = 310;   % Second cutoff frequency

Hd = createfilter(fc1,fc2,fs);
y = filter(Hd,x);

filterplot(t,y,t,x)
```

Demo: Assessment of Wind Turbine Locations

- **Goal**: Create analysis to choose best wind turbine location by estimating power generation at multiple locations

- **Approach:**
  - Interactively explore data and develop analysis approach
    - Load and preprocess data from observation towers
    - Fit wind speed probability distribution
    - Estimate average power
  - Automate to run on data from other locations

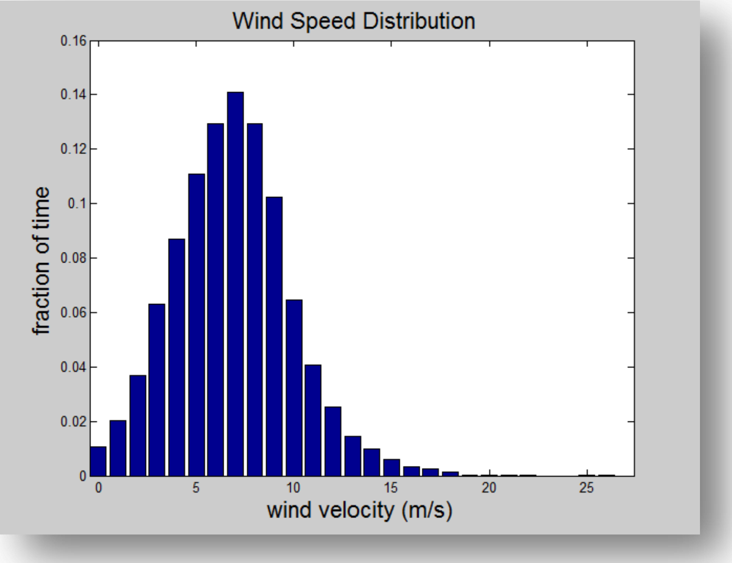


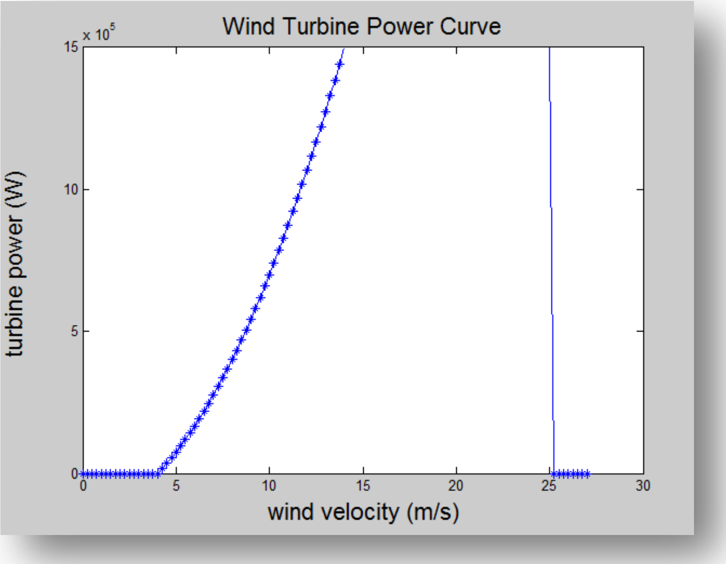
Calculating Average Turbine Power

$$Average\ Power = \int_{v_{min}}^{v_{max}} f(v) * W(v)\, dv$$

$f(v)$ *is the wind speed probablity distribution function*

$W(v)$ *is the turbine power curve (power as a function of wind speed)*





Wind Turbine Power Curve

Region I: $v < v_{in}$

$$W(v) = 0$$

Region II: $v_{in} < v < v_{rated}$

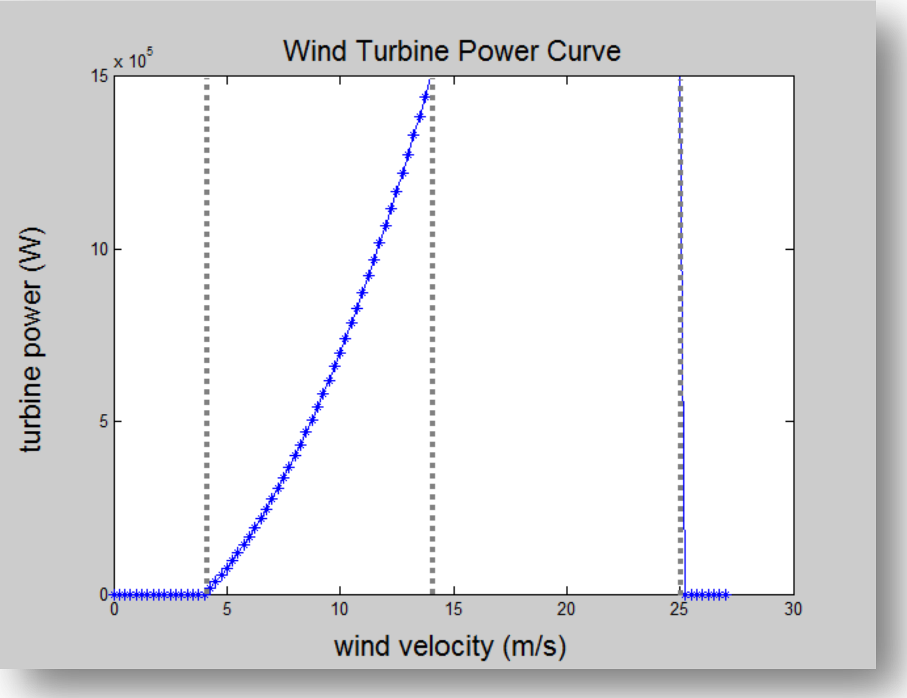
$$W(v) = p_{rated}\left(\frac{v^2 - {v_{in}}^2}{{v_{rated}}^2 - {v_{in}}^2}\right)$$

Region III: $v_{rated} < v < v_{out}$

$$W(v) = p_{rated}$$

Region IV: $v > v_{out}$

$$W(v) = 0$$

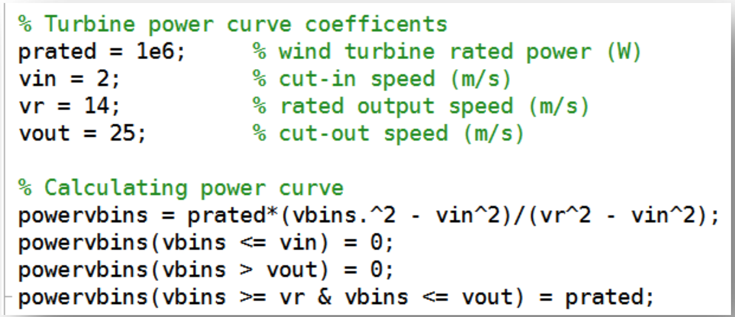


Demo Summary

Products Used

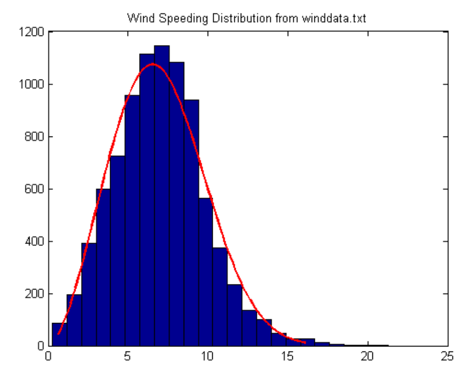
- **MATLAB**

- MATLAB language
  - High-level
  - Matrix-based
  - No need for low-level administrative tasks



```
% Turbine power curve coefficents
prated = 1e6;       % wind turbine rated power (W)
vin = 2;            % cut-in speed (m/s)
vr = 14;            % rated output speed (m/s)
vout = 25;          % cut-out speed (m/s)

% Calculating power curve
powervbins = prated*(vbins.^2 - vin^2)/(vr^2 - vin^2);
powervbins(vbins <= vin) = 0;
powervbins(vbins > vout) = 0;
powervbins(vbins >= vr & vbins <= vout) = prated;
```

- Math and graphics functions for engineering and science



- Supports a range of programming styles
  - Started working interactively (Command line, plotting)
  - Automated with a script and functions (Editor, command history, comments)

Demo: Virus Dynamics Modeling

- **Goal**: Modify the code for a working model of virus dynamics to make it more maintainable, reusable, and robust

- **Approach:**
  - Organize code by using different function types
  - Add error checking to validate inputs
  - Allow different calling syntaxes to support different use cases

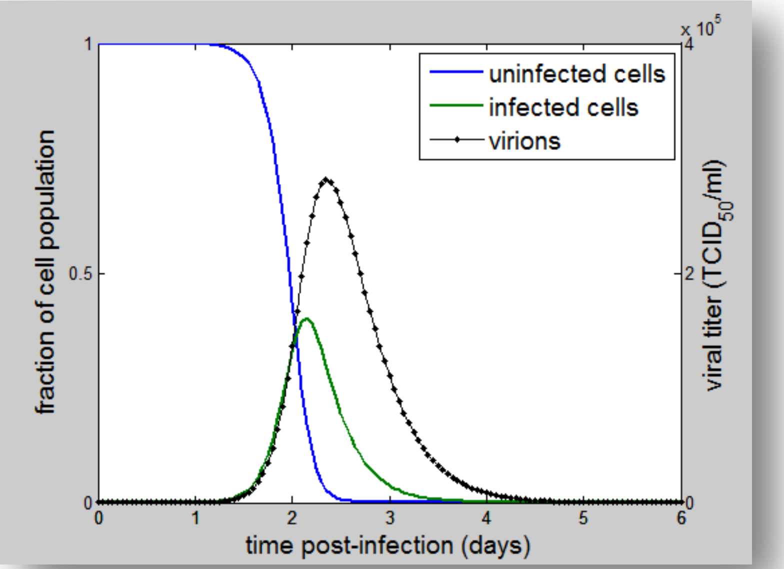
Mathematical Model of Virus Dynamics

$$\frac{dT}{dt} = -\beta\, T\, V$$

$$\frac{dI}{dt} = \beta\, T\, V - \delta\, I$$

$$\frac{dV}{dt} = p\, I - c\, V$$

T – target (uninfected cells)
I – infected cells
V – free virions (virus particles)



Model parameters:

β - infection rate of uninfected cells

δ - death rate for infected cells

p - production rate of virus particles

c - clearance rate of virus particles

$$R = \frac{p\,\beta\,T_o}{c\delta}$$

If R>1, infection can be established.

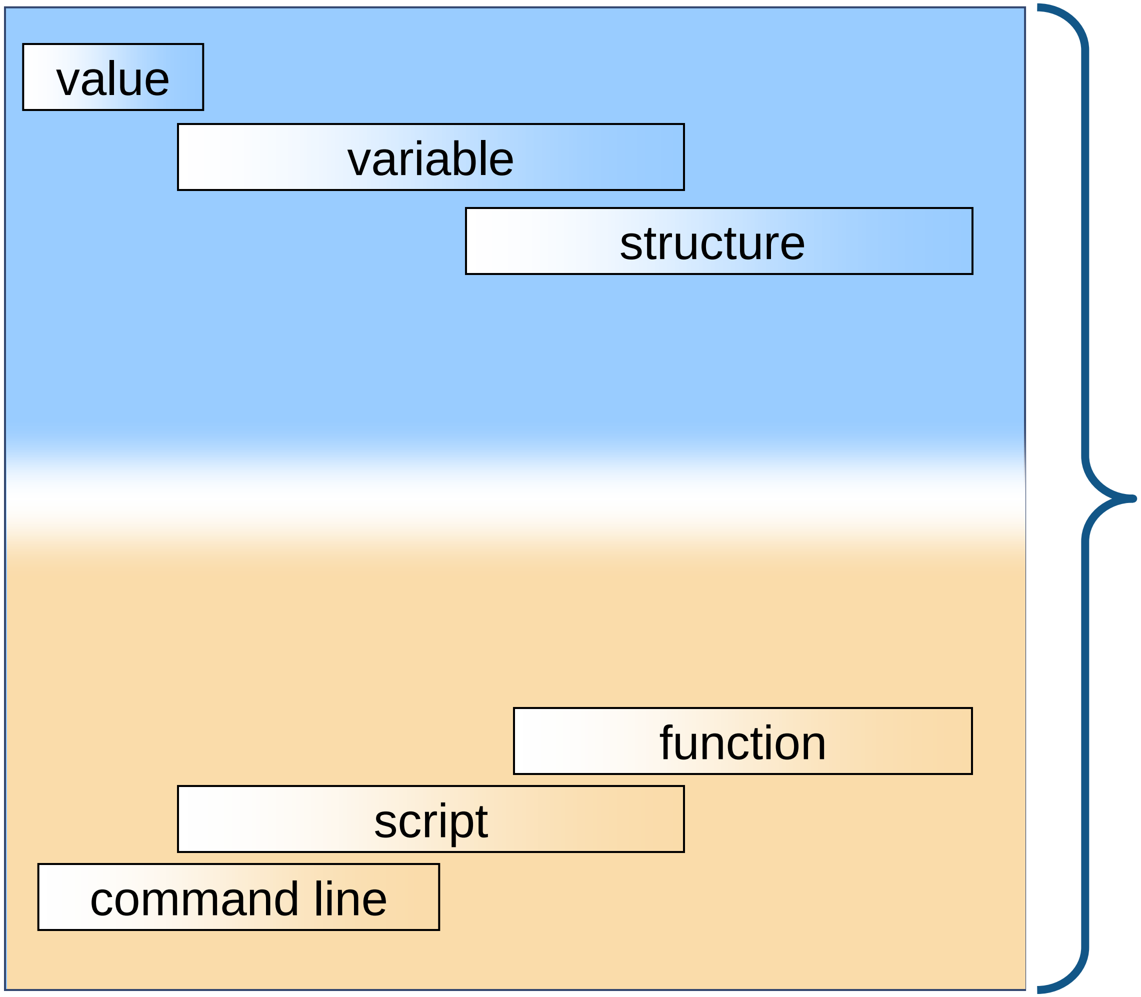
Demo Summary

Products Used
- MATLAB
- Curve Fitting Toolbox

- Started with working code
- Refined and improved code
  - Maintainable
    - Subfunctions, nested functions
  - Reusable / more general
    - Function with flexible input arguments
  - Robust
    - Error checking and validating inputs
    - Profiler to assess performance

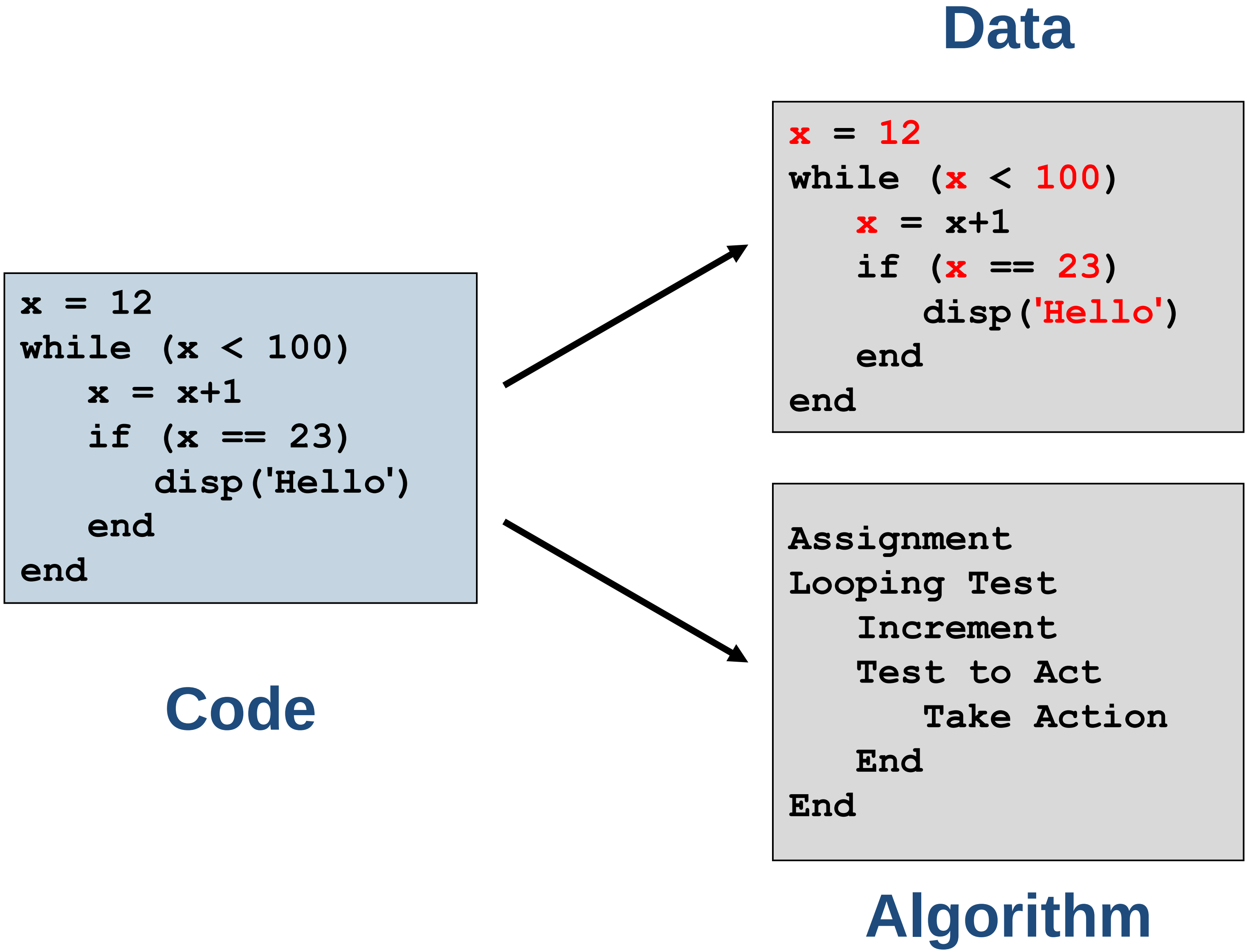
- Used development tools
  - Code Analyzer, Debugger

Range of Programming Techniques



- Manage larger, complex applications
- Organize data and algorithms

What is a program?



```
x = 12
while (x < 100)
    x = x+1
    if (x == 23)
        disp('Hello')
    end
end
```

Code

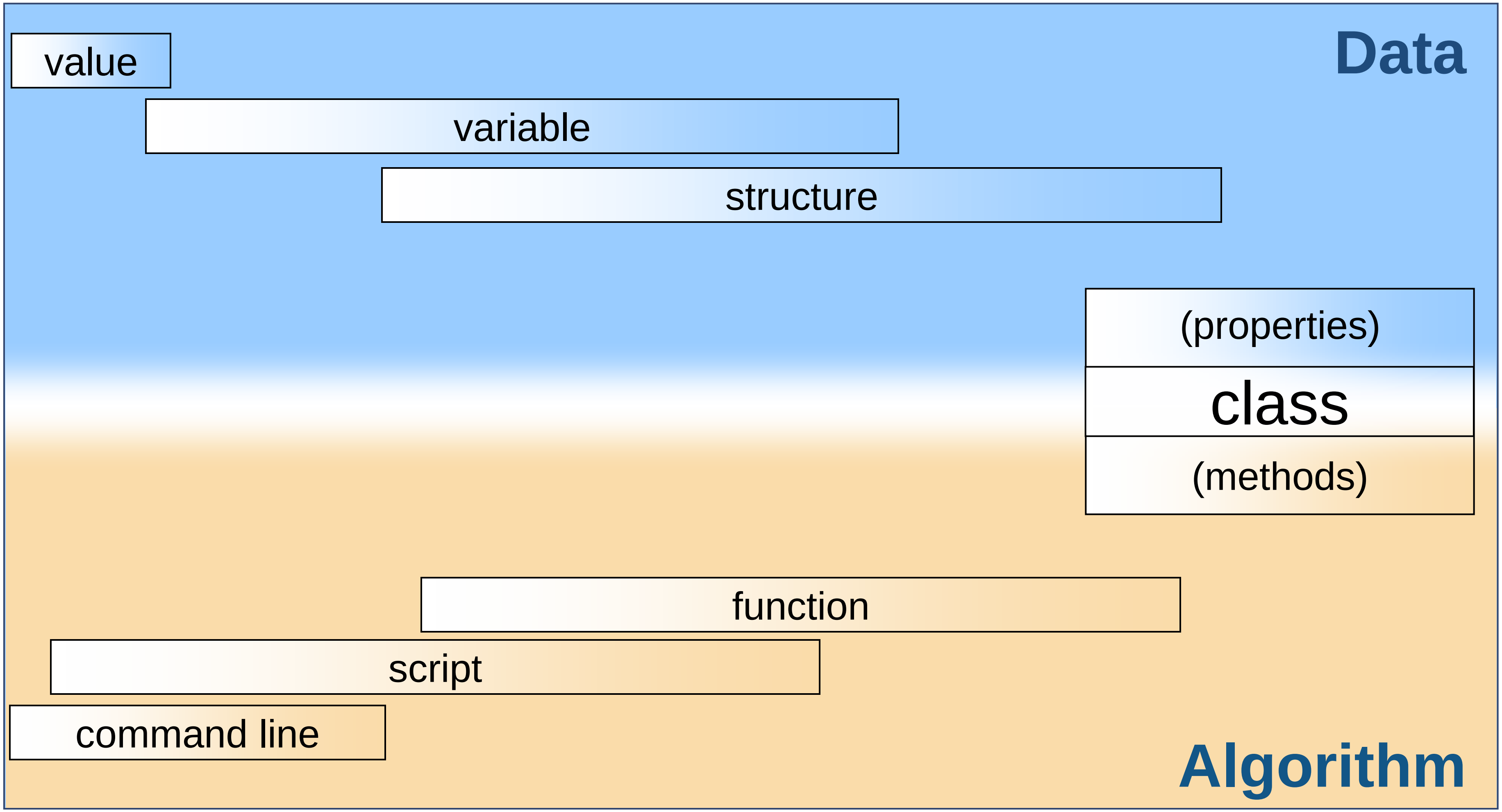
Data

```
x = 12
while (x < 100)
    x = x+1
    if (x == 23)
        disp('Hello')
    end
end
```

```
Assignment
Looping Test
    Increment
    Test to Act
        Take Action
    End
End
```

Algorithm

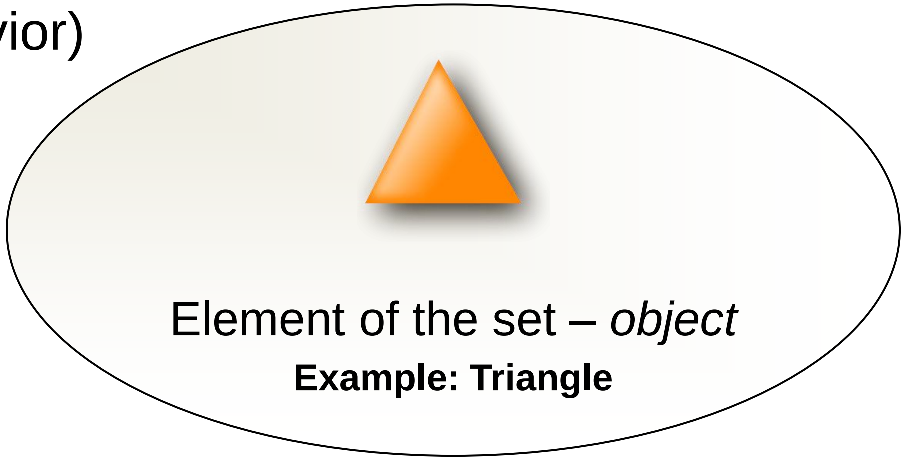
Range of Programming Techniques



Object-Oriented Terminology

- Class
  - Outline of an idea
  - *Properties* (data, states)
  - *Methods* (algorithms, behavior)

- Object
  - Specific example of a *class*
  - *Instance*



Element of the set – *object*

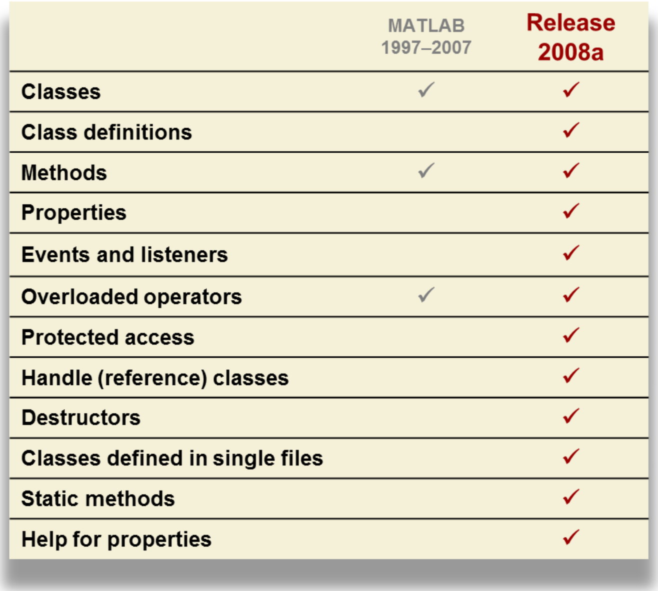
Example: Triangle

Defined set – *class*

Example: Polygons

Object-Oriented Programming with MATLAB

- Combines related data and algorithms
- Class definition files – describe object behavior
  - Build on existing classes with **inheritance**
  - Control access to properties and methods with **attributes**
  - Monitor object property changes and actions with **events and listeners**
- Use matrix-based aspects of MATLAB with objects
- Packages – define scope (namespace) of functions and classes

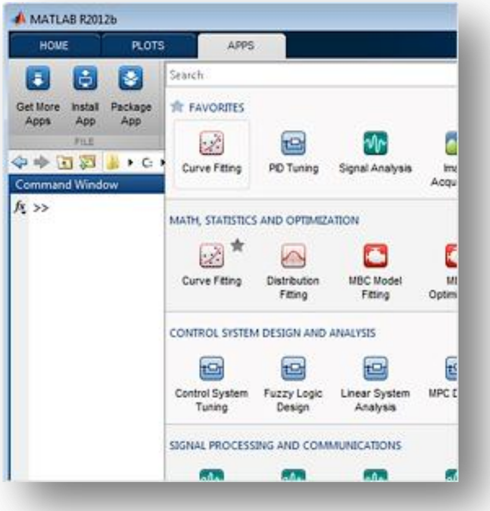


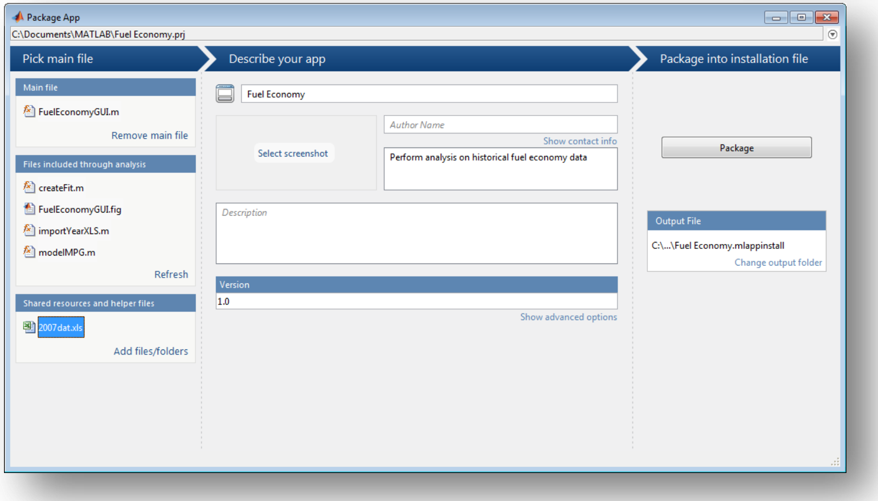
	MATLAB	Release
	1997–2007	2008a
Classes	✓	✓
Class definitions		✓
Methods	✓	✓
Properties		✓
Events and listeners		✓
Overloaded operators	✓	✓
Protected access		✓
Handle (reference) classes		✓
Destructors		✓
Classes defined in single files		✓
Static methods		✓
Help for properties		✓

Packaging and Sharing MATLAB Apps

- Create single file for distribution and installation into gallery

- Packaging tool:
  - Automatically includes all necessary files
  - Documents required products





C:\Documents\MATLAB\Fuel Economy.prj

Main file: FuelEconomyGUI.m

Files included through analysis: createFit.m, FuelEconomyGUI.fig, importYearXLS.m, modelMPG.m

Shared resources and helper files: 2007dat.xls

Fuel Economy

Author Name

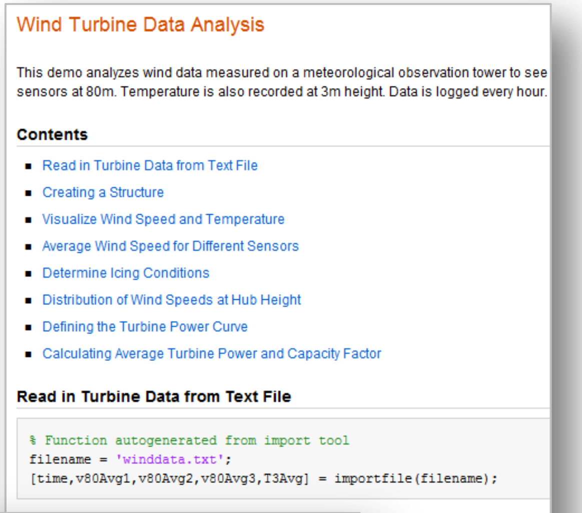
Perform analysis on historical fuel economy data

Version: 1.0

Output File: C:\...\Fuel Economy.mlappinstall

Sharing Results from MATLAB

- Automatically generate reports
  - Publish MATLAB files
- Create graphical user interfaces
  - Programmatically
  - GUIDE: GUI Design Environment (includes a layout editor)
- Package as an app



Wind Turbine Data Analysis

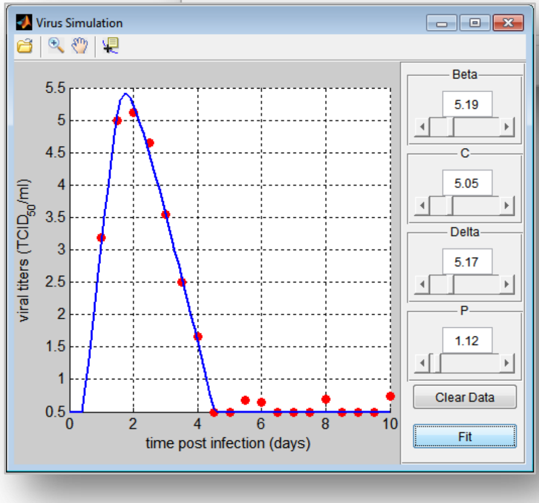
This demo analyzes wind data measured on a meteorological observation tower to see sensors at 80m. Temperature is also recorded at 3m height. Data is logged every hour.

Contents

- Read in Turbine Data from Text File
- Creating a Structure
- Visualize Wind Speed and Temperature
- Average Wind Speed for Different Sensors
- Determine Icing Conditions
- Distribution of Wind Speeds at Hub Height
- Defining the Turbine Power Curve
- Calculating Average Turbine Power and Capacity Factor

Read in Turbine Data from Text File

```
% Function autogenerated from import tool
filename = 'winddata.txt';
[time,v80Avg1,v80Avg2,v80Avg3,T3Avg] = importfile(filename);
```



Beta 5.19

C 5.05

Delta 5.17

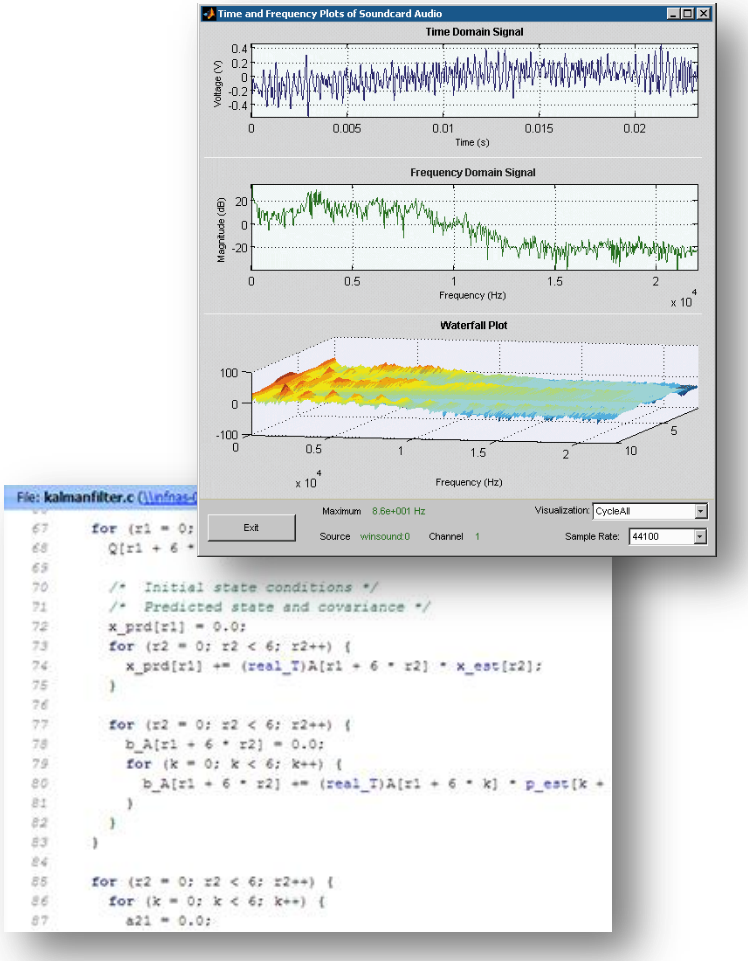
P 1.12

Deploying to Other Environments

- Share individual algorithms or complete applications

- Create stand-alone applications and software components

- Generate portable C code for desktop or embedded applications



Maximum 8.6e+001 Hz

Source winsound:0 Channel 1

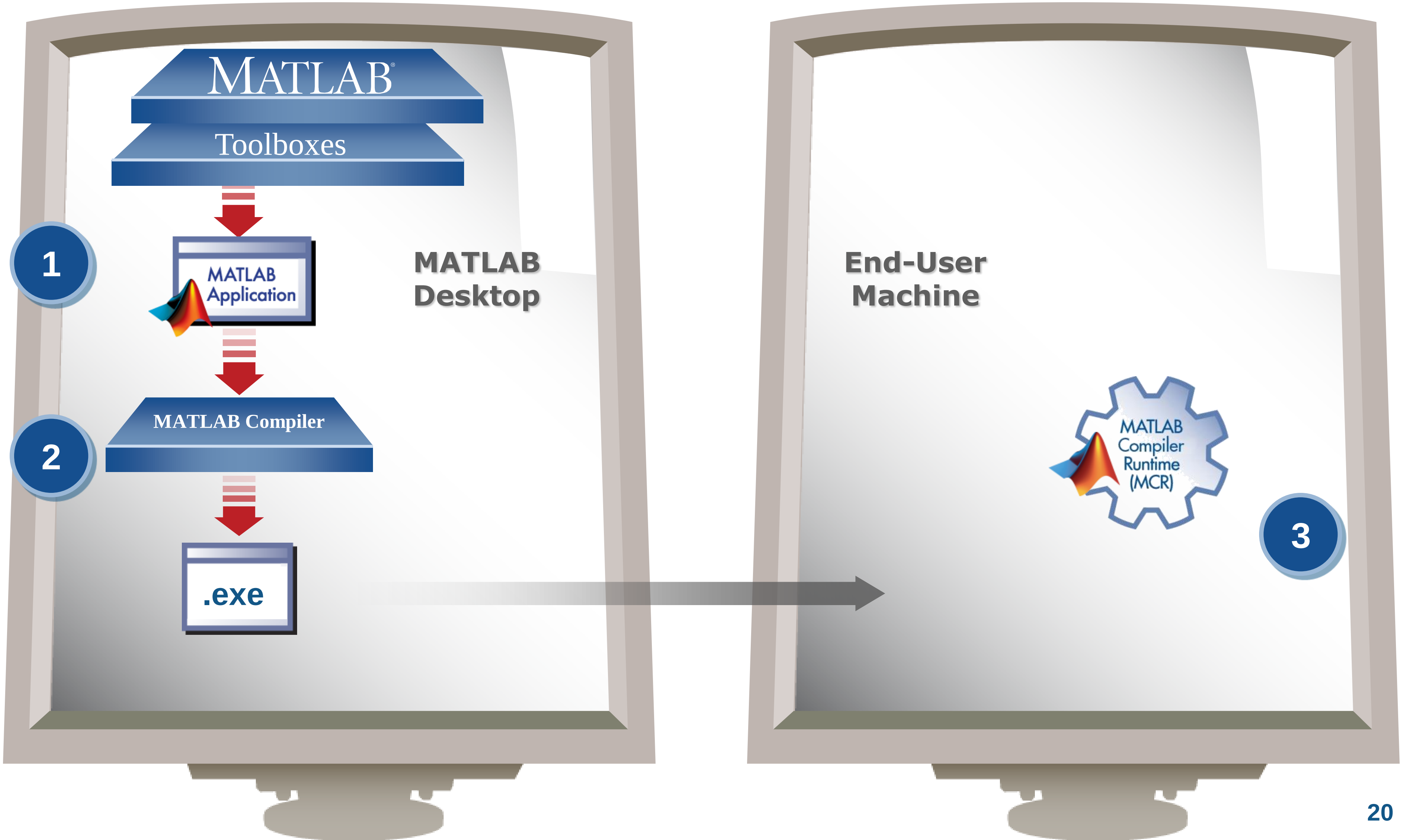
Visualization: CycleAll

Sample Rate: 44100

File: kalmanfilter.c (\\infnas-0

```
    for (r1 = 0;
      Q[r1 + 6 *

      /*  Initial state conditions */
      /*  Predicted state and covariance */
      x_prd[r1] = 0.0;
      for (r2 = 0; r2 < 6; r2++) {
        x_prd[r1] += (real_T)A[r1 + 6 * r2] * x_est[r2];
      }

      for (r2 = 0; r2 < 6; r2++) {
        b_A[r1 + 6 * r2] = 0.0;
        for (k = 0; k < 6; k++) {
          b_A[r1 + 6 * r2] += (real_T)A[r1 + 6 * k] * p_est[k +
        }
      }
    }

    for (r2 = 0; r2 < 6; r2++) {
      for (k = 0; k < 6; k++) {
        a21 = 0.0;
```

Deploying Applications with MATLAB

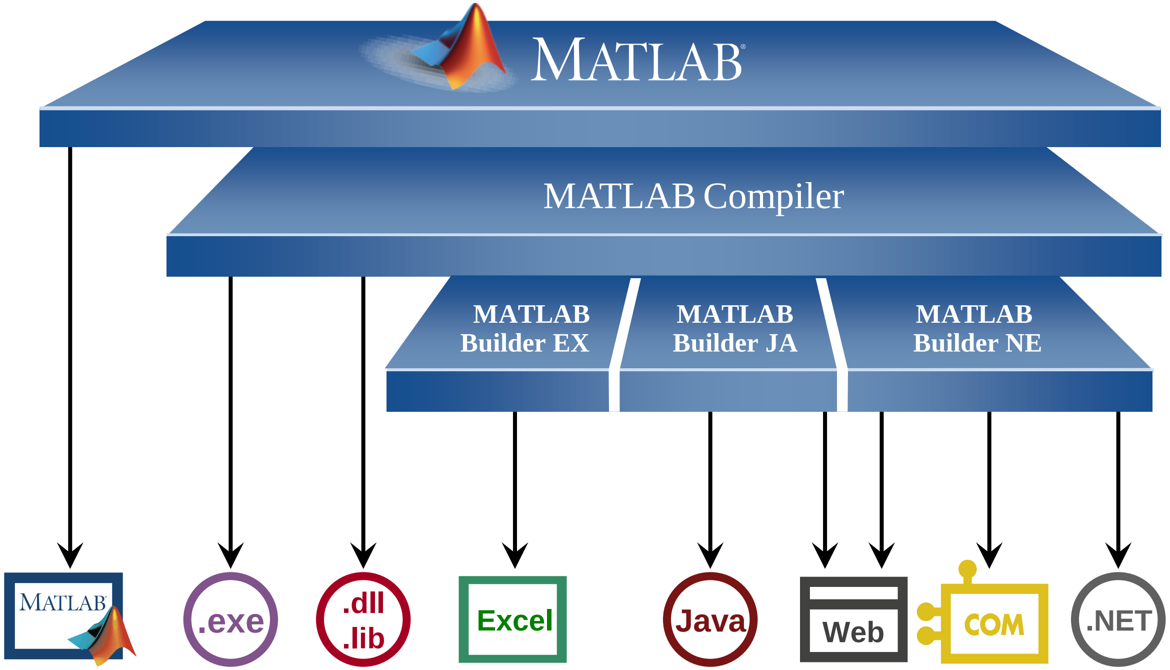


Deploying Applications with MATLAB

- Give MATLAB code to other users
  - MATLAB apps
  - MATLAB files

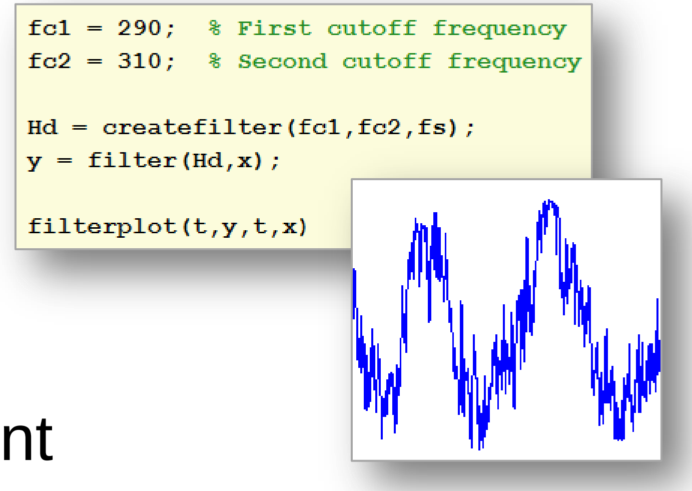
- Share applications with end users who do not need MATLAB
  - Stand-alone executables
  - Shared libraries
  - Software components

- Royalty-free distribution



Summary – MATLAB for Programming

- High-level language
  - Matrix-based
  - Math and graphics functions
  - Traditional programming language features
- Interactive development environment
  - Tools, visualizations, and help
- Supports a range of programming styles
  - Interactive command line, scripts and functions, object-oriented programming

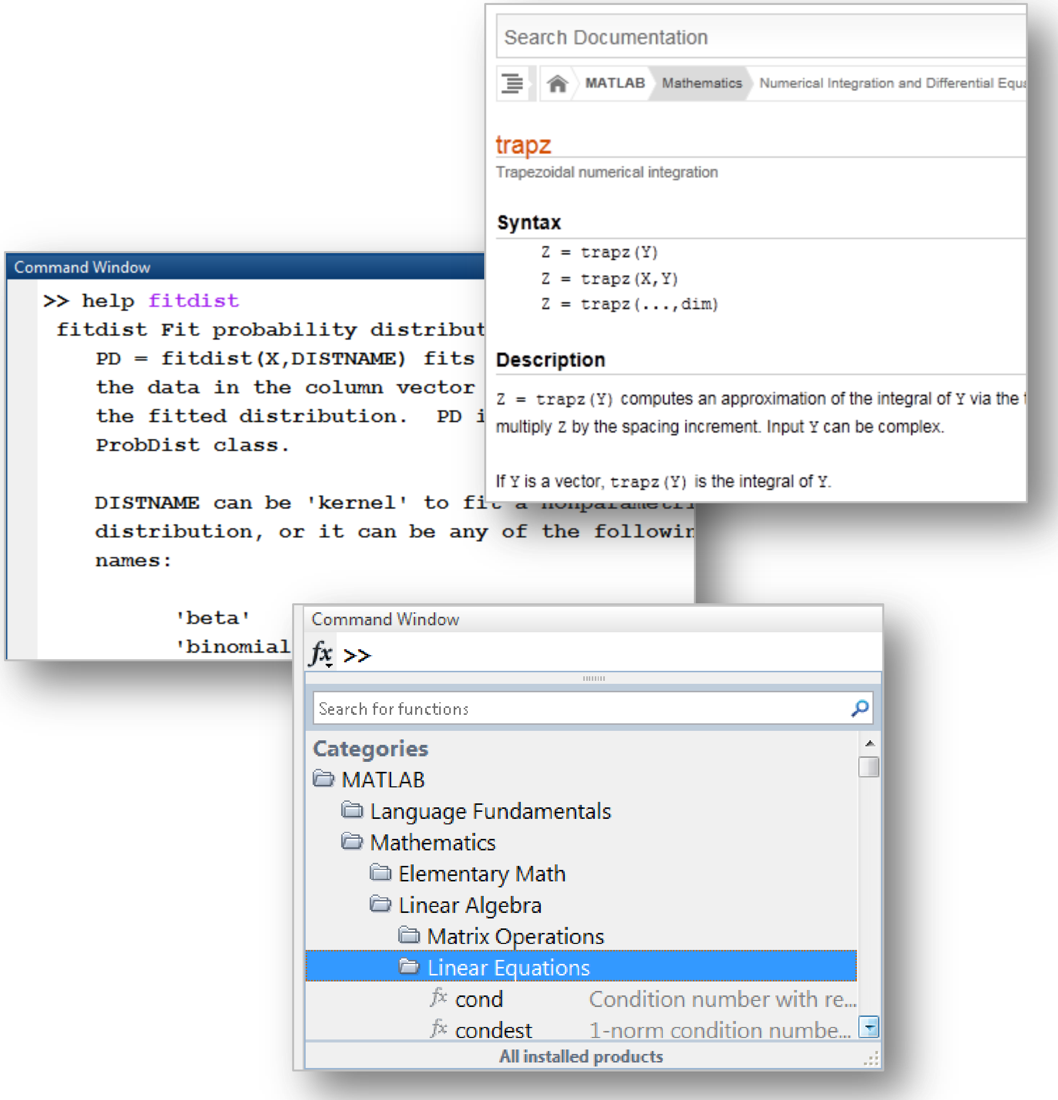


```
fc1 = 290;  % First cutoff frequency
fc2 = 310;  % Second cutoff frequency

Hd = createfilter(fc1,fc2,fs);
y = filter(Hd,x);

filterplot(t,y,t,x)
```

Summary

Multiple Ways to Get Help

- `doc`
- `help <name>`
- Function Browser,
  function hints,
  tab completion



Search Documentation

trapz

Trapezoidal numerical integration

Syntax

```
Z = trapz(Y)
Z = trapz(X,Y)
Z = trapz(...,dim)
```

Description

`Z = trapz(Y)` computes an approximation of the integral of `Y` via the t
multiply `Z` by the spacing increment. Input `Y` can be complex.

If `Y` is a vector, `trapz(Y)` is the integral of `Y`.

Command Window

```
>> help fitdist
 fitdist Fit probability distribut
    PD = fitdist(X,DISTNAME) fits
    the data in the column vector
    the fitted distribution.  PD i
    ProbDist class.

    DISTNAME can be 'kernel' to fit a nonparametri
    distribution, or it can be any of the followin
    names:

          'beta'
          'binomial
```

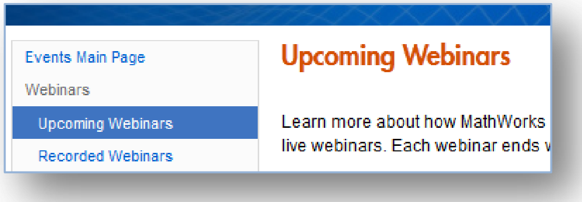
Command Window

```
fx >>
```

Search for functions

Resources

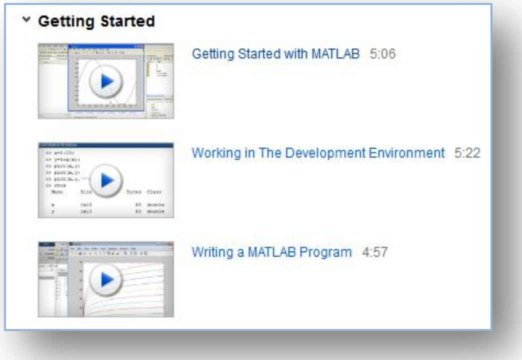
- Webinars
  - Object-Oriented Programming in MATLAB
  - MATLAB for C/C++ Programmers
  - MATLAB to C Made Easy



Upcoming Webinars

Learn more about how MathWorks live webinars. Each webinar ends w

- Videos and code examples
  - MATLAB product page
  - Documentation


Getting Started

Getting Started with MATLAB 5:06

Working in The Development Environment 5:22

Writing a MATLAB Program 4:57

- MATLAB Central
  - Open exchange for the MATLAB and Simulink user community

MATLAB CENTRAL

MATLAB Central

- **File Exchange**
  - Download free files
  - Over 19000K available – including functions, apps, examples, and models
- **MATLAB Answers**
  - Ask programming questions
  - Search 70000K+ existing answers
- **Cody**
  - Challenge and expand your knowledge of MATLAB
- **Blogs**
  - Read commentary from MathWorks engineers who design, build, and support MATLAB and Simulink

MATLAB CENTRAL

File Exchange Answers Newsgroup Link Exchange Blogs

Trendy
Connect the dots

Cody
Let the games begin

MathWorks Support for Student Competitions

Student teams use MATLAB and Simulink in math and robotics competitions.

File Exchange

Recent Files

- Convert SimBiology(R) model to COBRA Toolbox model *Sven Mesecke*
- A MATLAB Script for Terrestrial and Celestial Coordinate Conversion *David E*
- Aquiles *Luis Alberto Cañizares*
- International Reference Ionosphere (IRI) Model *Drew Compston*
- GetGoogleSpreadsheet *Daniel*
- An Approximation for the Prize-Collecting Steiner Tree Problem *Fangzhou Chen*