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
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

\[ \text{Average Power} = \int_{v_{\text{min}}}^{v_{\text{max}}} f(v) \times W(v) \, dv \]

- \( f(v) \) is the wind speed probability 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

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

- 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

\[
\frac{dT}{dt} = -\beta TV \\
\frac{dI}{dt} = \beta TV - \delta I \\
\frac{dV}{dt} = pI - cV
\]

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

Model parameters:

\( \beta \) - infection rate of uninfected cells
\( \delta \) - death rate for infected cells
\( p \) - production rate of virus particles
\( c \) - clearance rate of virus particles

\[
R = \frac{p \beta T_o}{c\delta} \quad \text{If } R>1, \text{ infection can be established.}
\]
Demo Summary

- 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

Products Used
- MATLAB
- Curve Fitting Toolbox
Range of Programming Techniques

- Manage larger, complex applications
- Organize data and algorithms
What is a program?

**Code**

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

**Algorithm**

Assignment
Looping Test
  Increment
  Test to Act
    Take Action
  End
End

**Data**

```plaintext
x = 12
while (x < 100)
    x = x+1
    if (x == 23)
        disp('Hello')
    end
end
```
Range of Programming Techniques

Value
- variable
- structure

Data
- (properties)
- class
- (methods)

Algorithm
- function
- script
- command line

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
Packaging and Sharing MATLAB Apps

- Create single file for distribution and installation into gallery

- Packaging tool:
  - Automatically includes all necessary files
  - Documents required products
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
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
Deploying Applications with MATLAB

1. **MATLAB Desktop**
   - MATLAB Application
   - MATLAB Compiler
   - .exe

2. **Toolboxes**

3. **End-User Machine**
   - MATLAB Compiler Runtime (MCR)
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

```matlab
fc1 = 290; % First cutoff frequency
c2 = 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
Resources

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

- **Videos and code examples**
  - MATLAB product page
  - Documentation

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