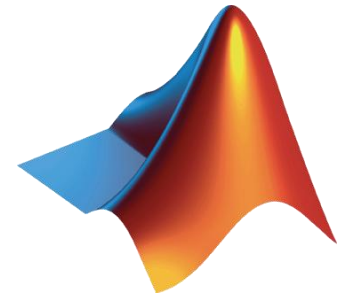


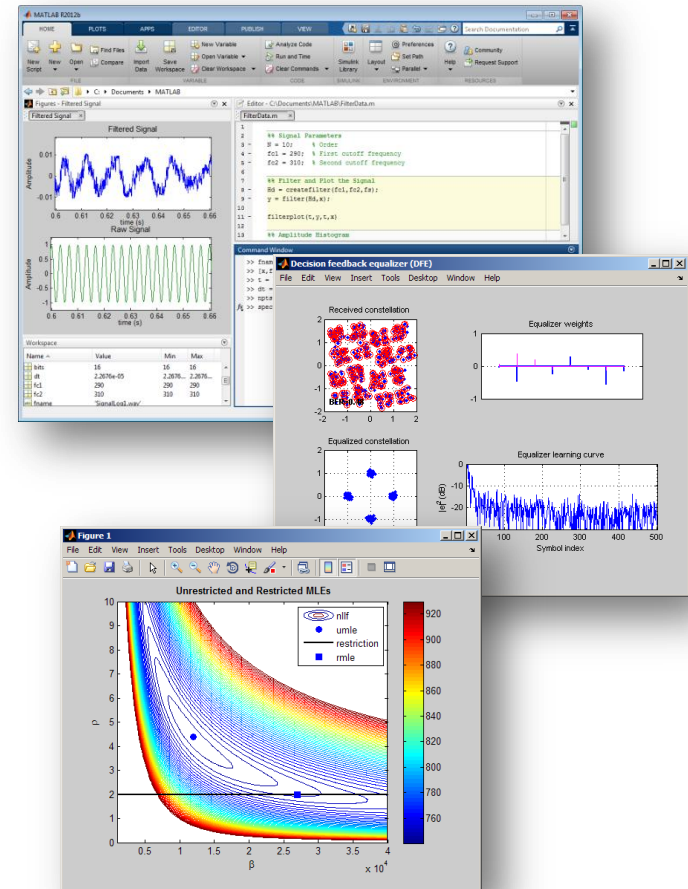
Introduction to MATLAB

Gergely Somlay
Application Engineer
gergely.somlay@gamax.hu



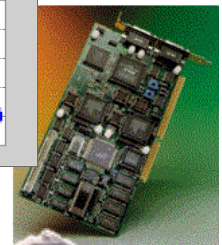
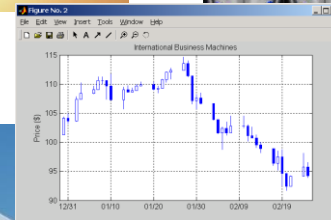
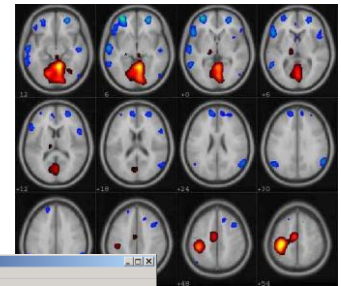
What is MATLAB?

- High-level language
- Interactive development environment
- Used for:
 - Numerical computation
 - Data analysis and visualization
 - Algorithm development and programming
 - Application development and deployment

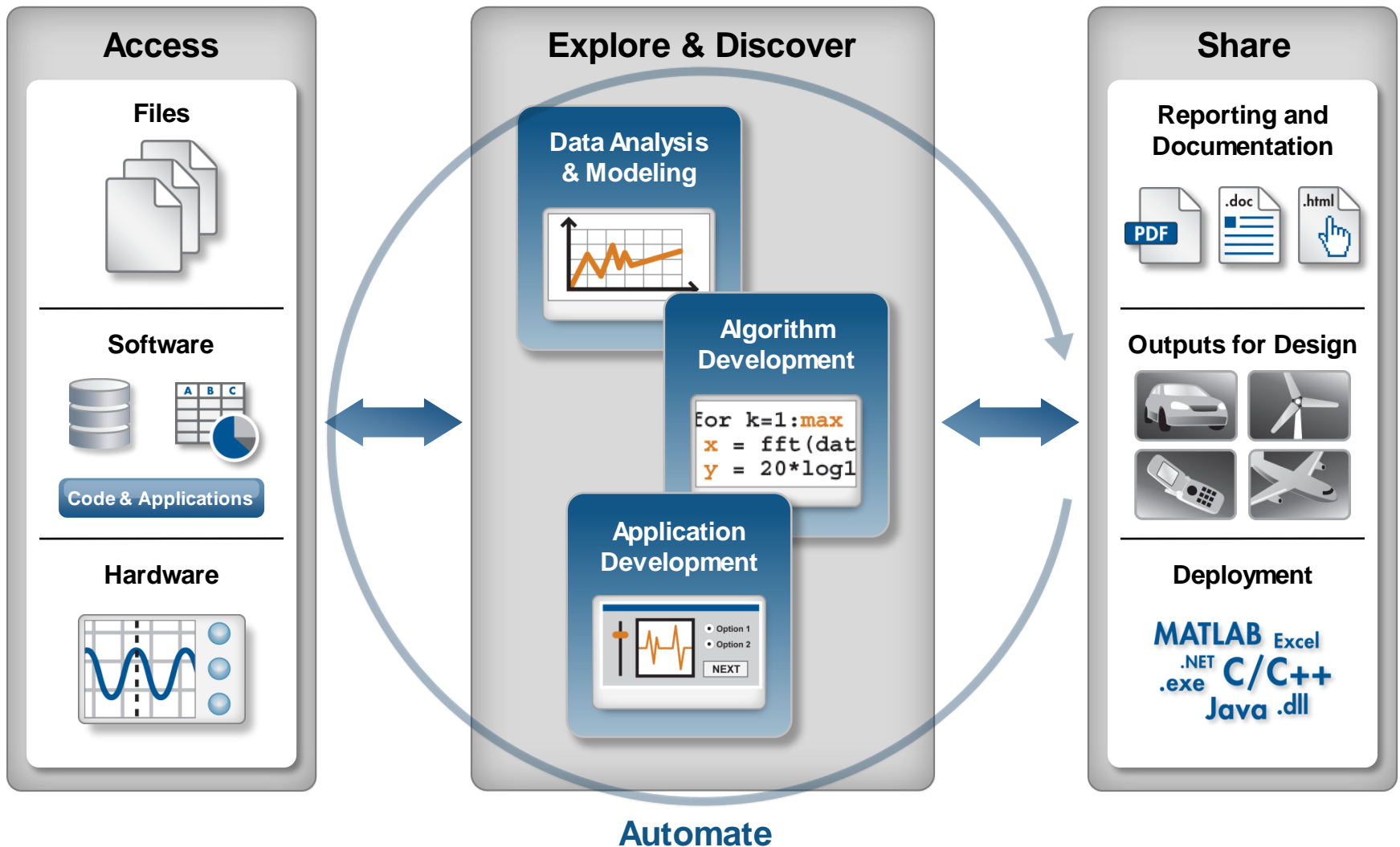


Key Industries

- Aerospace and Defense
- Automotive
- Biotech and Pharmaceutical
- Communications
- Education
- Electronics
- Energy and Power Production
- Financial Services
- Industrial Automation and Machinery
- Semiconductor



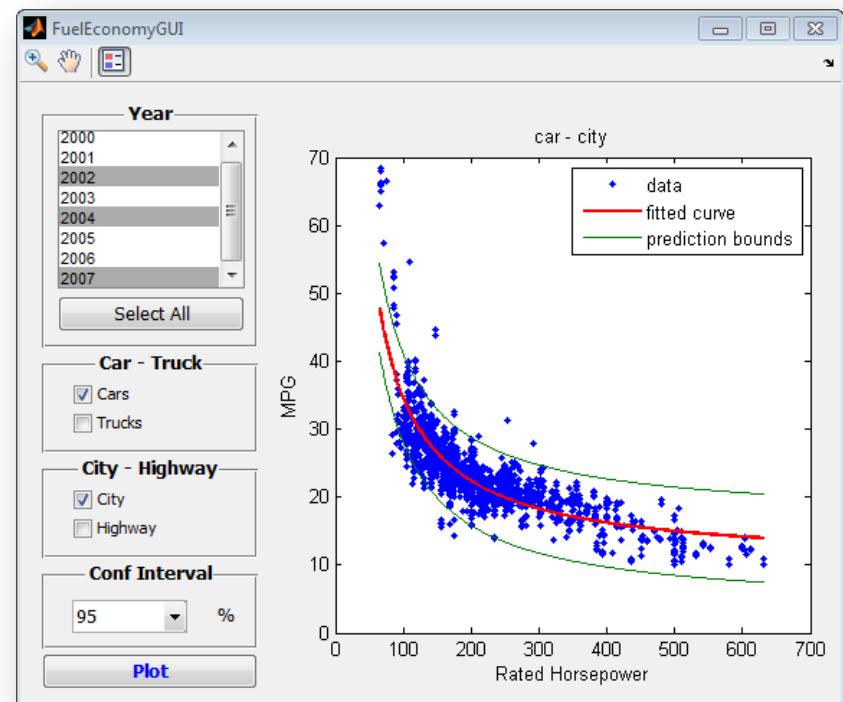
Technical Computing Workflow



Demo: Fuel Economy Analysis

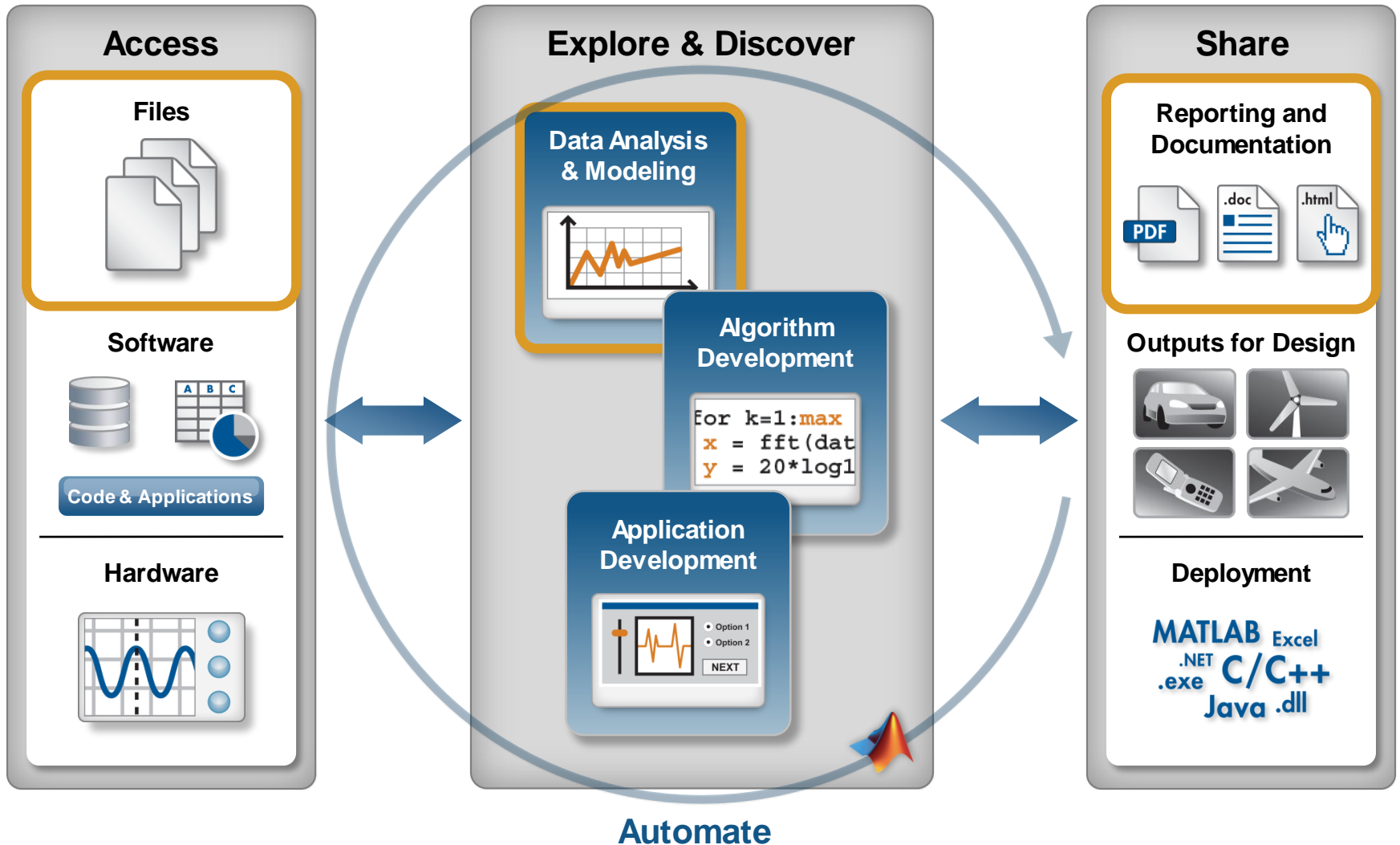
- Goal:
 - Study the relationships between fuel economy, horsepower, and type of vehicle

- Approach:
 - Access data from Excel
 - Interactively visualize and explore trends
 - Create a model
 - Document results



Demo: Fuel Economy Analysis

- Products Used
- MATLAB
 - Statistics Toolbox
 - Curve Fitting Toolbox



Accessing Data from MATLAB

Access

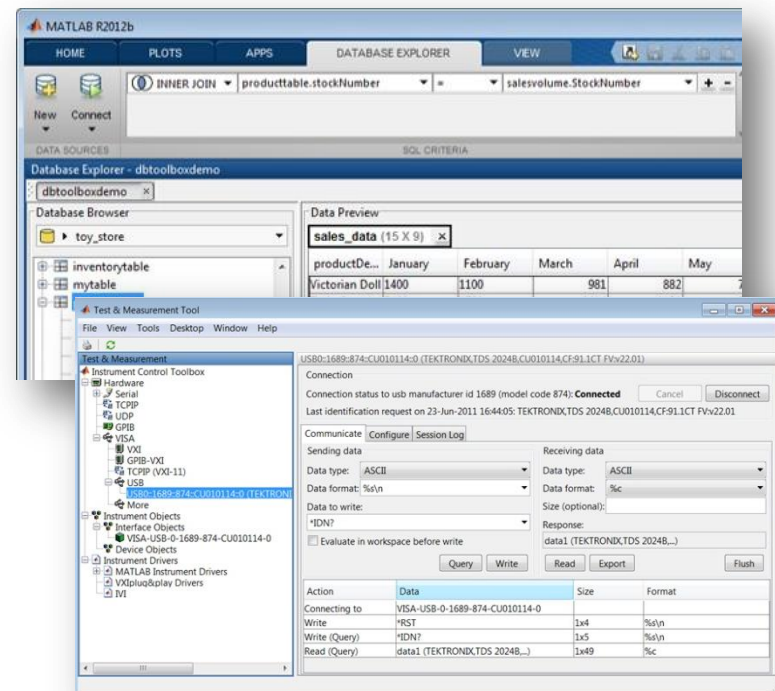
Explore & Discover

Share

- Files
 - Excel, text, or binary
 - Audio and video, image
 - Scientific formats and XML

- Applications and languages
 - C/C++, Java, FORTRAN
 - COM, .NET, shared libraries
 - Databases
(*Database Toolbox*)

- Measurement hardware
 - Data acquisition hardware
(*Data Acquisition Toolbox*)
 - Stand-alone instruments and devices
(*Instrument Control Toolbox*)



Data Analysis and Visualization in MATLAB

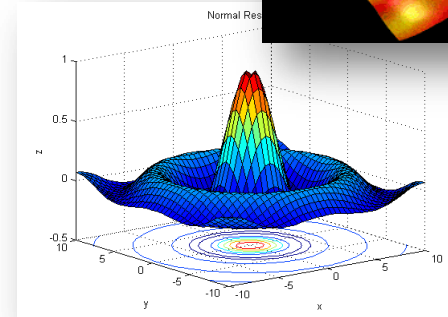
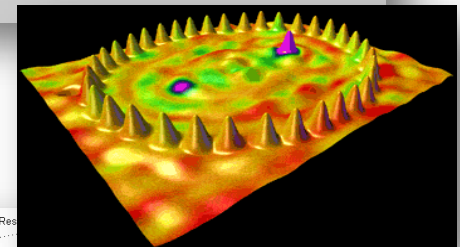
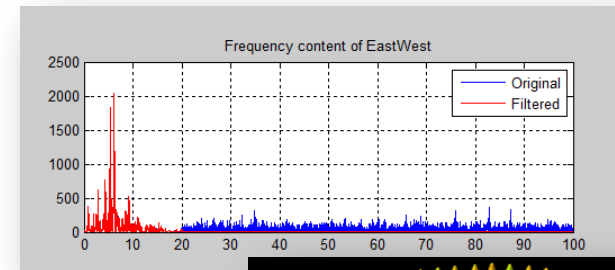
Access

Explore & Discover

Share

- Built-in engineering and mathematical functions
 - Interpolation, filtering, smoothing, Fourier analysis

- Extensive plotting capabilities
 - 2-D, 3-D, and volume visualization
 - Tools for creating custom plots



Expanding the Capabilities of MATLAB

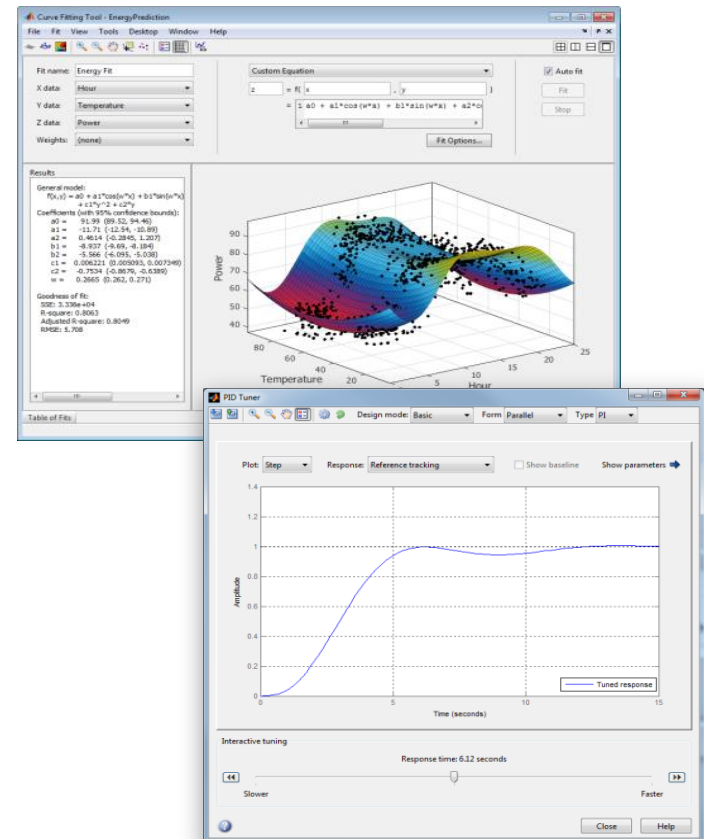
Access

Explore & Discover

Share

- MathWorks add-on tools for:
 - Math, statistics, and optimization
 - Control system design and analysis
 - Signal processing and communications
 - Image processing and computer vision
 - Parallel computing and more...

- Partner products provide:
 - Additional interfaces
 - Domain-specific analysis
 - Support for niche applications



Sharing Results from MATLAB

Access

Explore & Discover

Share

- Automatically generate reports
 - Publish MATLAB files
 - Customize reports using MATLAB Report Generator

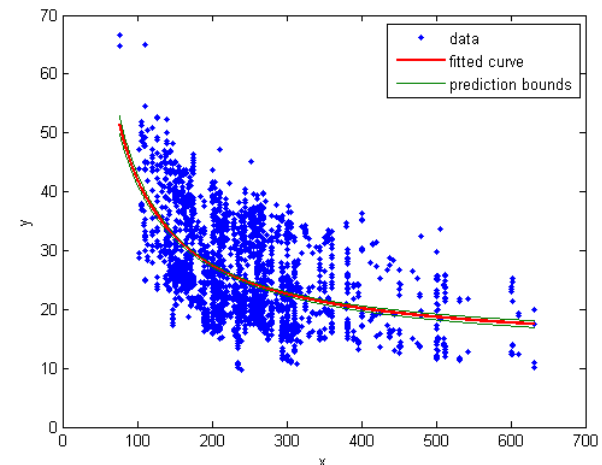
- Package as an app

- Deploy applications to other environments

Plot Data and Model

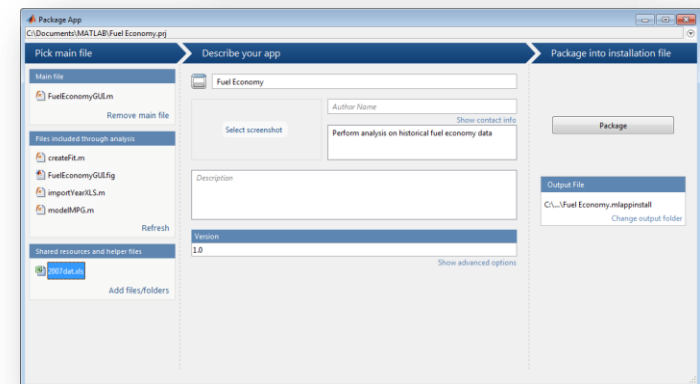
The result from the Curve Fitting Toolbox has a `plot` method for displaying the result graphically. We can choose to display the prediction bounds for the fit.

```
figure;
hh = plot(cf, 'r', carDataDS.RatedHP, carDataDS.MPG, 'predfunc', 0.95);
set(hh(2) , 'LineWidth', 2);
set(hh(3:4), 'LineStyle', '--', 'Color', [0 .5 0]);
```



Packaging and Sharing MATLAB Apps

- MATLAB apps
 - Interactive applications to perform technical computing tasks
 - Displayed in apps gallery
- Included in many MATLAB products
- Package your own app
 - Create single file for distribution and installation into gallery
 - Packaging tool:
 - Automatically includes all necessary files
 - Documents required products



Deploying Applications with MATLAB

Access

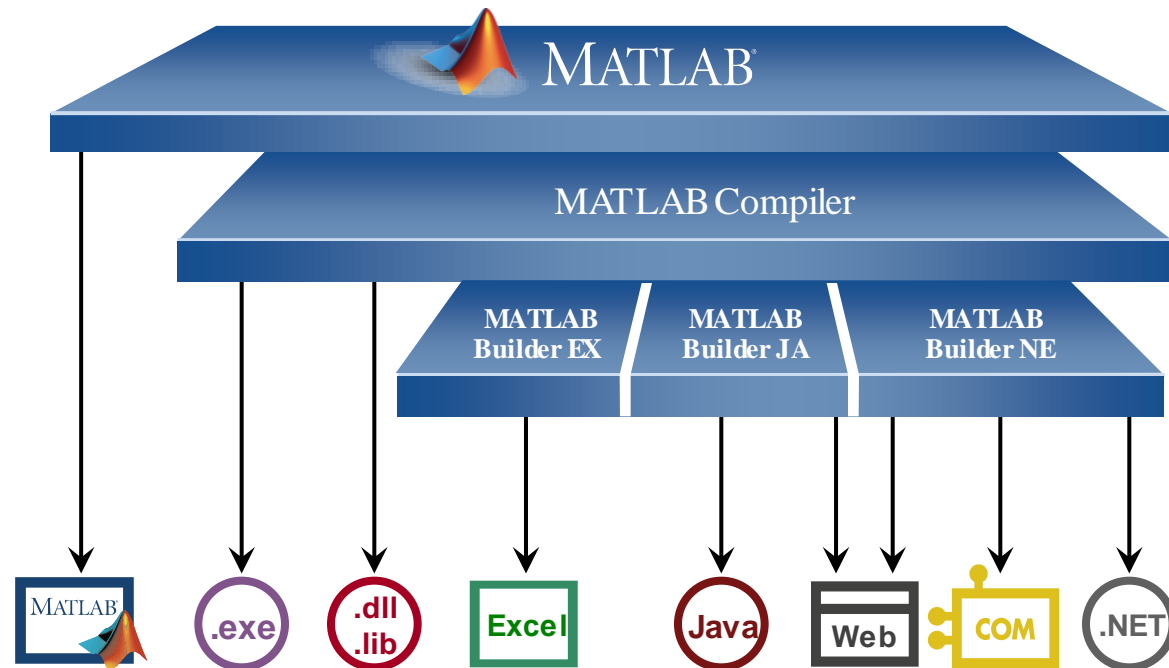
Explore & Discover

Share

- 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

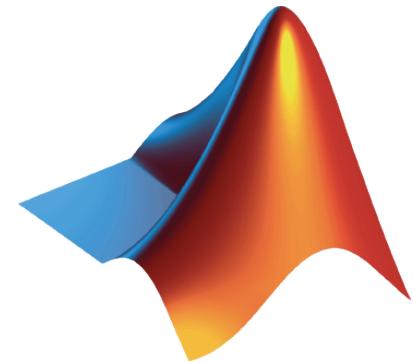


Using MATLAB

- High-level language
 - Native support for vector and matrix operations
 - Built-in math and visualization functions

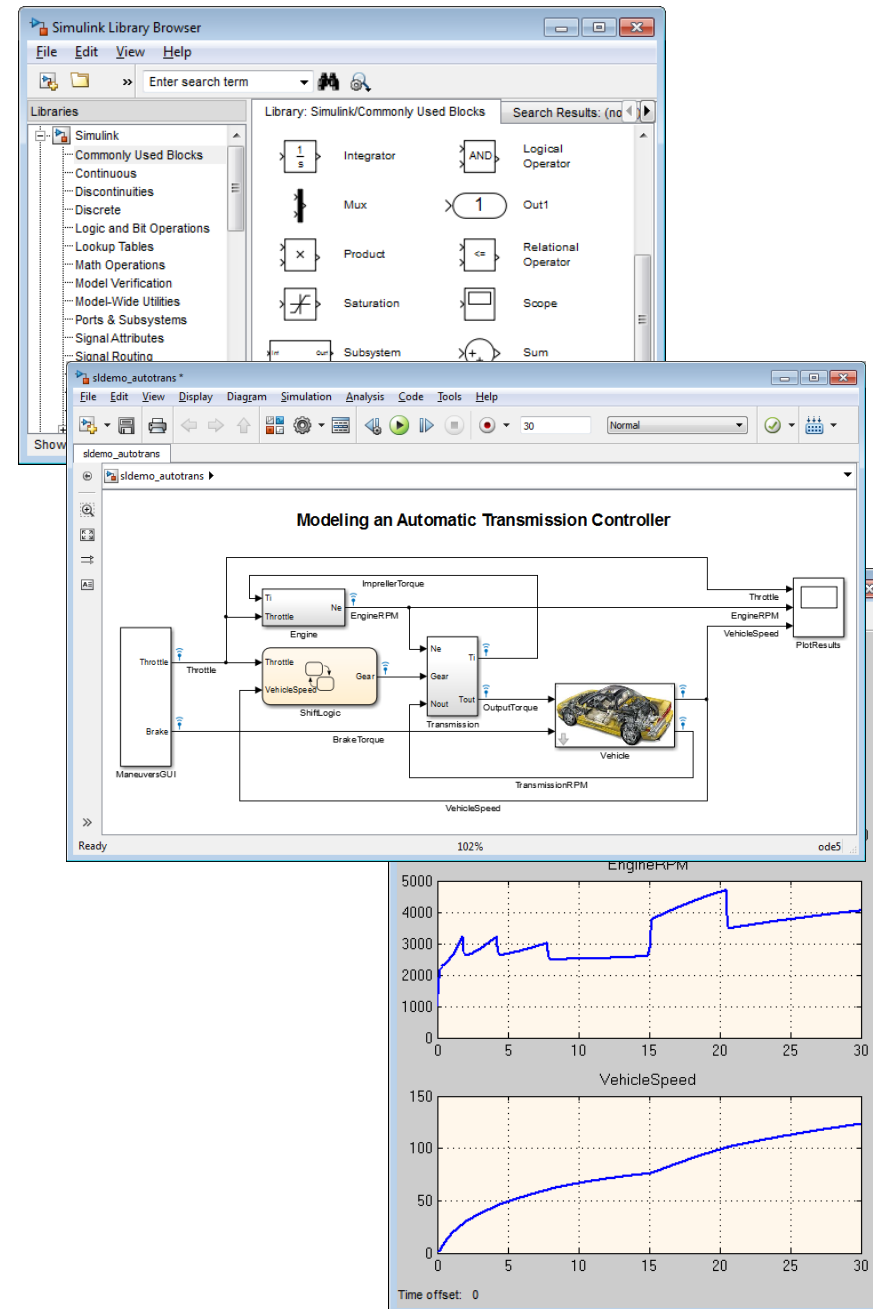
- Development environment
 - Interactive and easy to get started
 - Ideal for iterative exploration and design

- Technical computing platform
 - Add-on products for a range of application areas
(e.g., signal processing and communications, image and video processing, control systems, test and measurement)



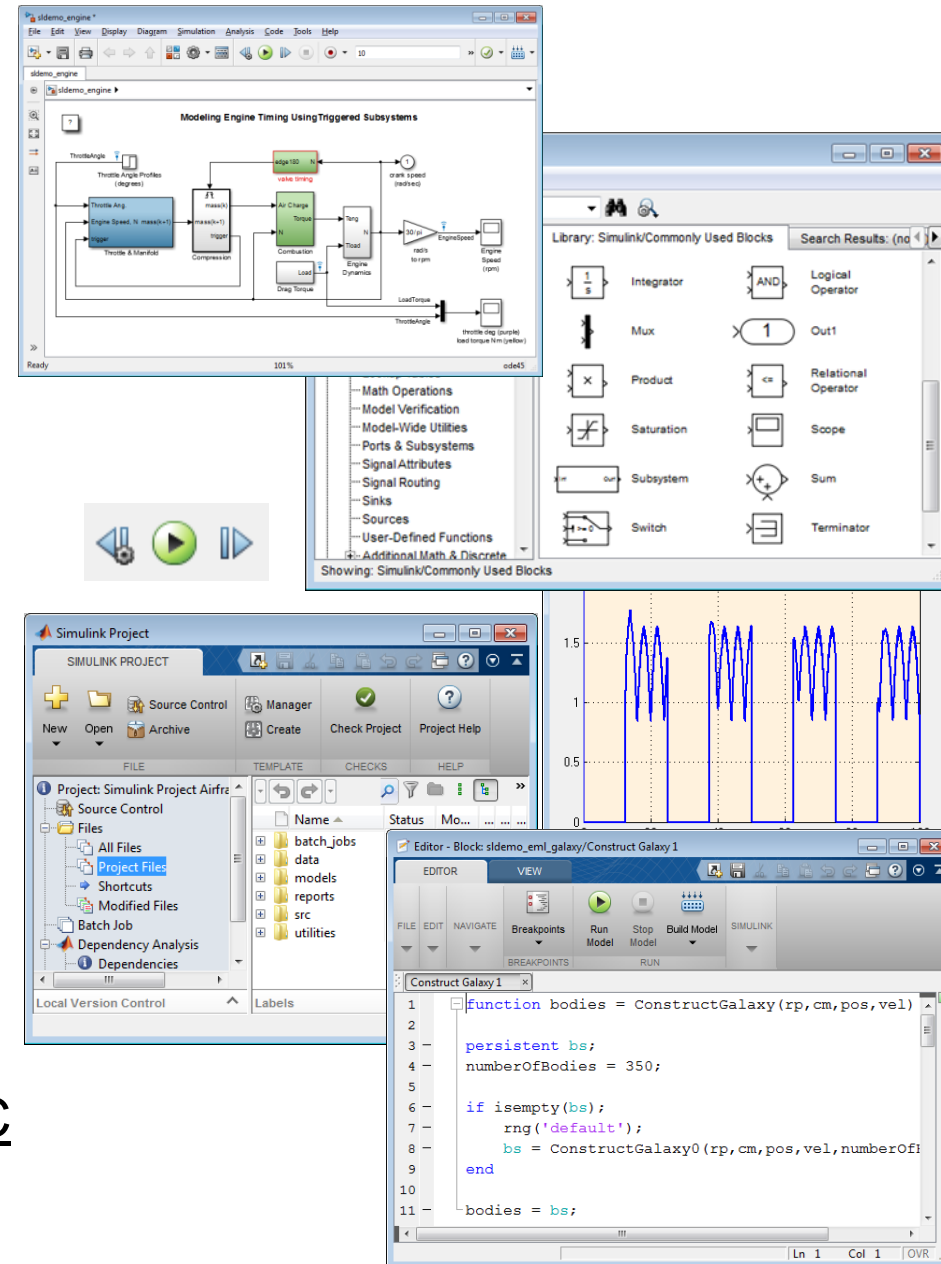
What is Simulink?

- Block-diagram environment
- Model, simulate, and analyze multidomain systems
- Design, implement, and test:
 - Control systems
 - Signal processing systems
 - Communications systems
 - Other dynamic systems
- Platform for Model-Based Design

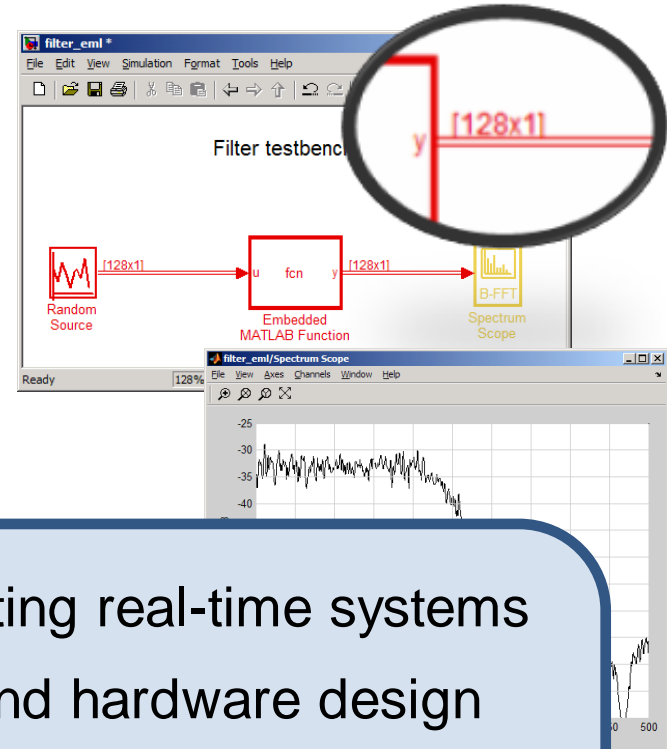
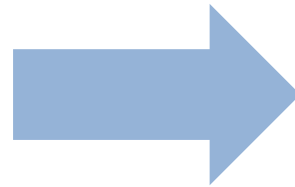
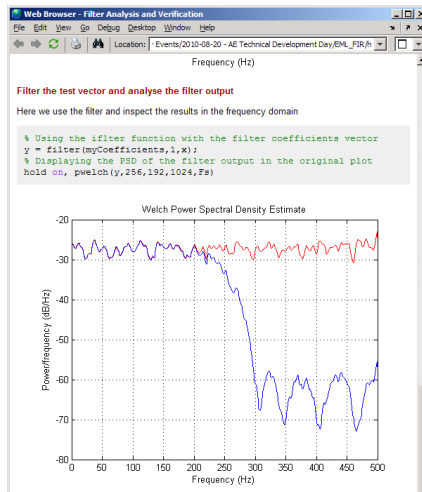


Simulink Key Features

- Graphical editor for building hierarchical block diagrams
- Libraries of continuous-time and discrete-time blocks
- Simulation engine with fixed-step and variable-step ODE solvers
- Scopes and data displays for viewing simulation results
- Project and data management tools
- MATLAB Function block for importing MATLAB algorithms
- Legacy Code Tool for importing C and C++ code into models

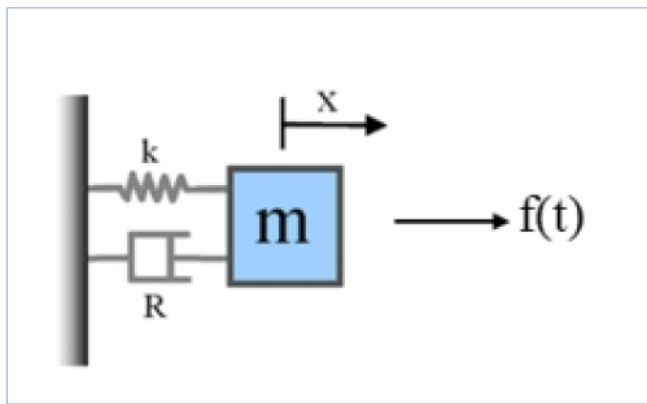


From MATLAB to Simulink – why?



- Intrinsic time management – simulating real-time systems
- Greater capabilities for fixed-point and hardware design
- Solid embedded code generation (C / HDL) infrastructure
- Dataflow-style diagram – easily document and reuse
- Integration with Analogue / Mixed Signal models

Case study: damped oscillation



$$m\ddot{x} + R\dot{x} + kx = f(t)$$

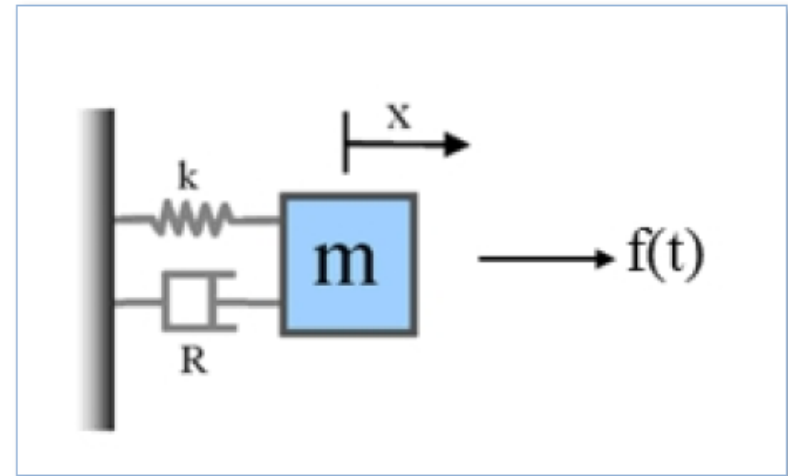
$$x(0) = x_0; \dot{x}(0) = \dot{x}_0$$

$$\Rightarrow \ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = \frac{1}{m}f(t)$$

$$\omega_n = \sqrt{\frac{k}{m}} = \textit{natural frequency}$$

$$\zeta = \frac{R}{2\sqrt{km}} = \textit{damping ratio}$$

Methods



- Handling a system with various methods
 - Numeric – **MATLAB**
 - Symbolic – **Symbolic Math Toolbox**
 - Dynamic system – **Simulink**
 - Physical Modeling – **Simscape**

MATLAB Environment

```

Editor - C:\01-Documents\Szilard\MATLAB\m-files\EDU\spring_mass_damper.m
EDITOR PUBLISH VIEW
+ New Open Save Compare Print FILE
Insert Comment Indent EDIT
Go To Breakpoints Run Run and Time Run and Advance RUN
spring_mass_damper.m spring.ssc
13 % We would like to examine the behavior of the free response
14 % system as the damping ratio changes. This will be done f
15 % the initial value problem for various damping ratios and
16 % results. Then, we will explore changes in the damping ra
17 % characteristic roots. (Recall that the characteristic ro
18 % free-response behavior.)
19 %
20
21
22 %% Set physical and derived parameters.
23 %
24 % To enable numerical solution of the differential equation
25 % specific values for the mass and (linear) spring rate. W
26 % these to derive the (circular) natural frequency.
27 %
28
29 close all
30 clc
31
32 param.m = 1; % Mass [kg]
33 param.k = 1; % Spring rate [N/m]
34 param.wn = sqrt(param.k/param.m); % (Circular) natura
35 param.OM = 2; % Exciting frequenc
36 param.isexcited = 1;
37
38
39 %% Set initial conditions.
40 %
41 % To solve our 2nd order differential equation, we need two
42 % conditions: one on position, the other on velocity. More
43 % from the accompanying PowerPoint slides that we have repr

```

spring_mass_damper.pdf - Adobe Reader

File Edit View Window Help

1 (1 of 4) 64.1%

Tools Sign Comment

Bookmarks

- Table of Contents
- Initial Value Problem
- Set physical and derived parameters.
- Set initial conditions.
- Set simulation parameters.
- Undamped system.
- Consider several damping ratios.

Transient Behavior and Location of Characteristic Roots in the Complex Plane

Table of Contents

Initial Value Problem	1
Set physical and derived parameters	1
Set initial conditions	1
Set simulation parameters	2
Undamped system	2
Consider several damping ratios	3

Initial Value Problem

Consider the system described by the following second-order, ordinary differential equation and associated initial conditions:

$$\frac{d^2 x}{dt^2} + 2\zeta\omega_n \frac{dx}{dt} + \omega_n^2 x = \frac{1}{m} f(t); x(0) = x_0, \frac{dx}{dt}(0) = 0$$

We would like to examine the behavior of the free response of the system as the damping ratio changes. This will be done first by solving the initial value problem for various damping ratios and plotting the results. Then, we will explore changes in the damping ratio affect the characteristic roots. (Recall that the characteristic roots determine the free-response behavior.)

Set physical and derived parameters.

To enable numerical solution of the differential equation, we will set specific values for the mass and (linear) spring rate. We will then use these to derive the (circular) natural frequency.

```

close all
clc

param.m = 1; % Mass [kg]
param.k = 1; % Spring rate [N/m]
param.wn = sqrt(param.k/param.m); % (Circular) natural frequency [rad/s]
param.OM = 2; % Exciting frequency
param.isexcited = 1;

```

Set initial conditions.

To solve our 2nd order differential equation, we need two initial conditions: one on position, the other on velocity. Moreover, recall from the accompanying PowerPoint slides that we have represented our 2nd

1

Transient Behavior and Location of Characteristic Roots in the Complex Plane

order linear ODE into a system of 1st order ODE's amenable to solution using MATLAB. Having defined position and velocity as our system states, we must define an initial state vector.

```

x0 = 0; % Position [m]
x0_dot = 0; % Velocity [m/s]
x0 = [x0; x0_dot]; % Initial condition on state vector

```

Set simulation parameters.

Symbolic Math - MuPAD

ode_simple* - MuPAD

File Edit View Navigation Insert Format Notebook Window Help




Generic Monospace 16 B I U x₂ x²

```

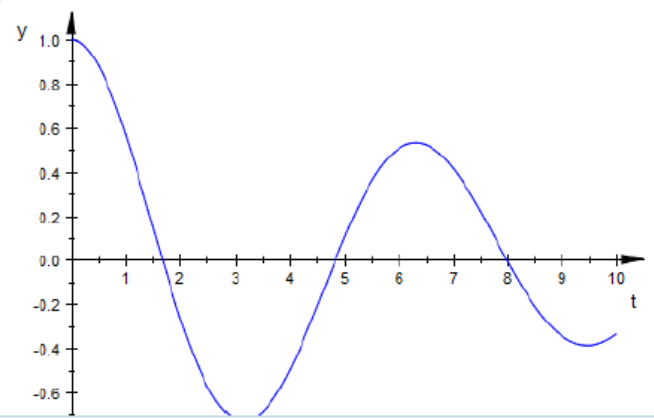
x''(t) + 2 ωn ζ x'(t) + ωn x(t)
o := ode({ODE, x(0)=1, x'(0)=0}, x(t))
ode({x'(0) = 0, x(0) = 1, x''(t) + 2 ωn ζ x'(t) + ωn x(t)}, x(t))
Zeta := 0.1:
omega_n := 1:
OM := 2:
o2 := subsex(o, wn = omega_n, z = Zeta)
ode({x'(0) = 0, x(0) = 1, x''(t) + 0.2 x'(t) + x(t)}, x(t))
s := solve(o2)
{1.0 e-0.1 t cos(0.9949874371 t) + 0.1005037815 e-0.1 t sin(0.9949874371 t)}
plot(s, t = 0..10)

```

Command Bar

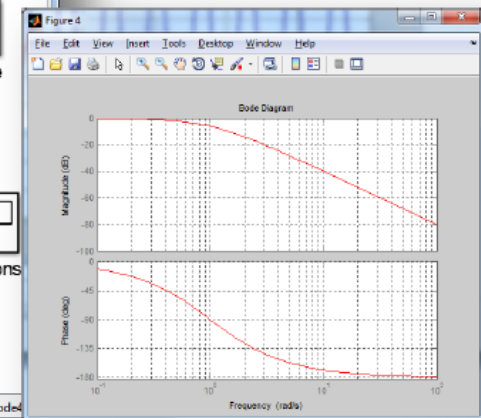
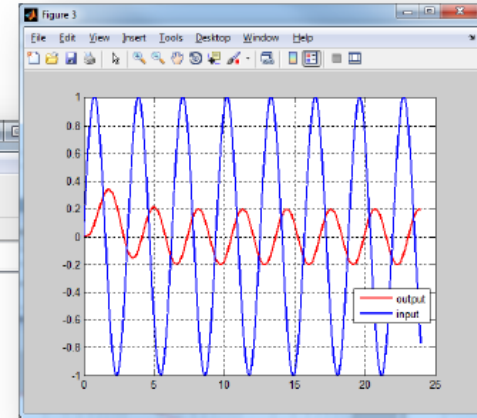
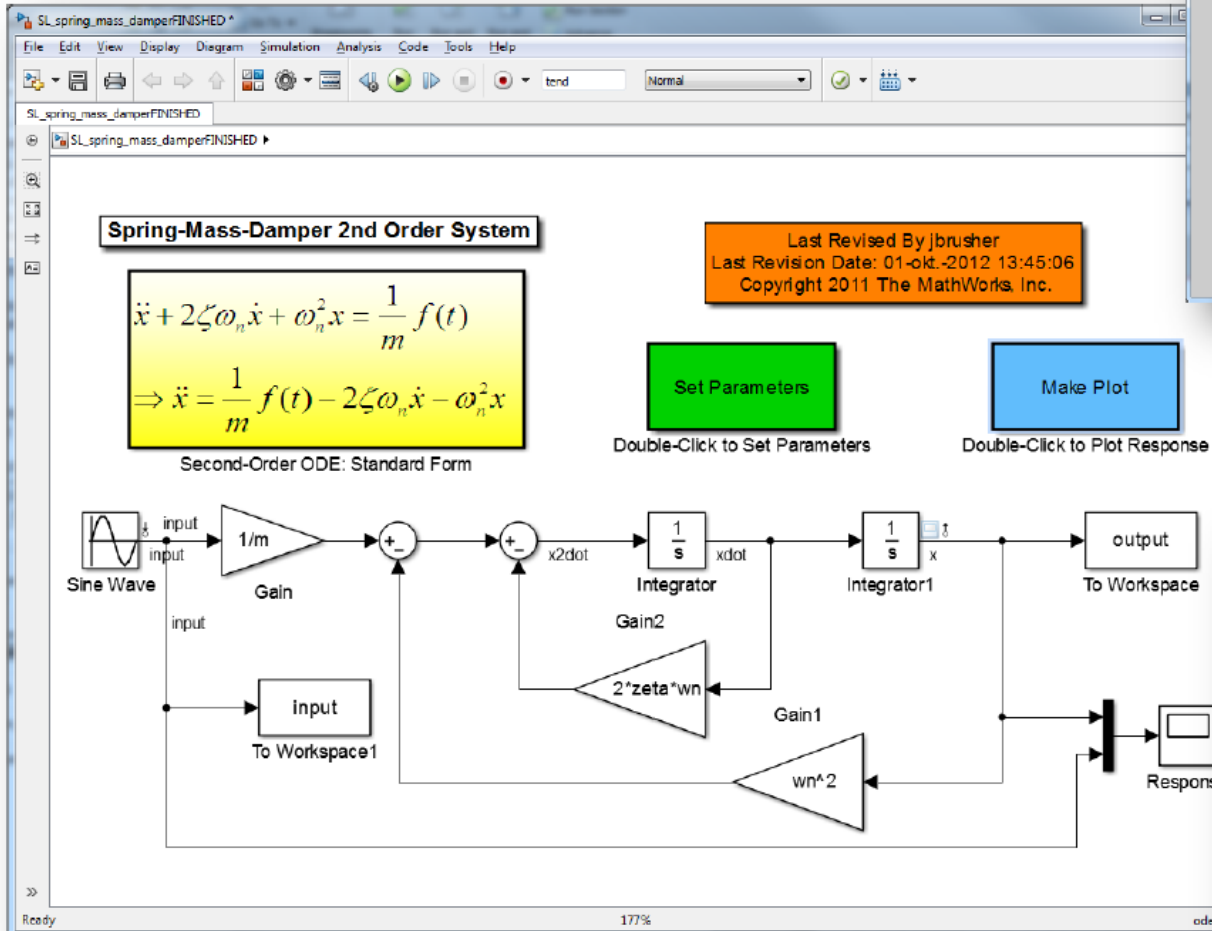
$\frac{\partial}{\partial x} f$ $\lim_{x \rightarrow a} f$ $\sum_n f$
 $\int f dx$ $f \Rightarrow \hat{f}$ $\prod_n f$
 $\{x\}_{f=0}$ $f \Rightarrow f$ $f|_{x=a}$
 $\pi \approx \dots$ $a = b$ $a := b$
 $a + b$ $n!$ $x \rightarrow f(x)$
 $\sin a$ e^a $\{v_i \text{ if } v_i\}$
 $e \dots \infty$ $\alpha \dots \Omega$ mks
  

▼ General Math
▼ Plot Commands

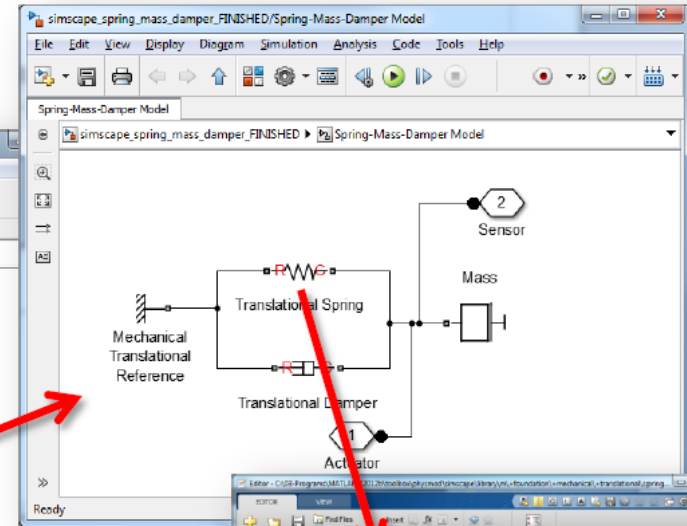
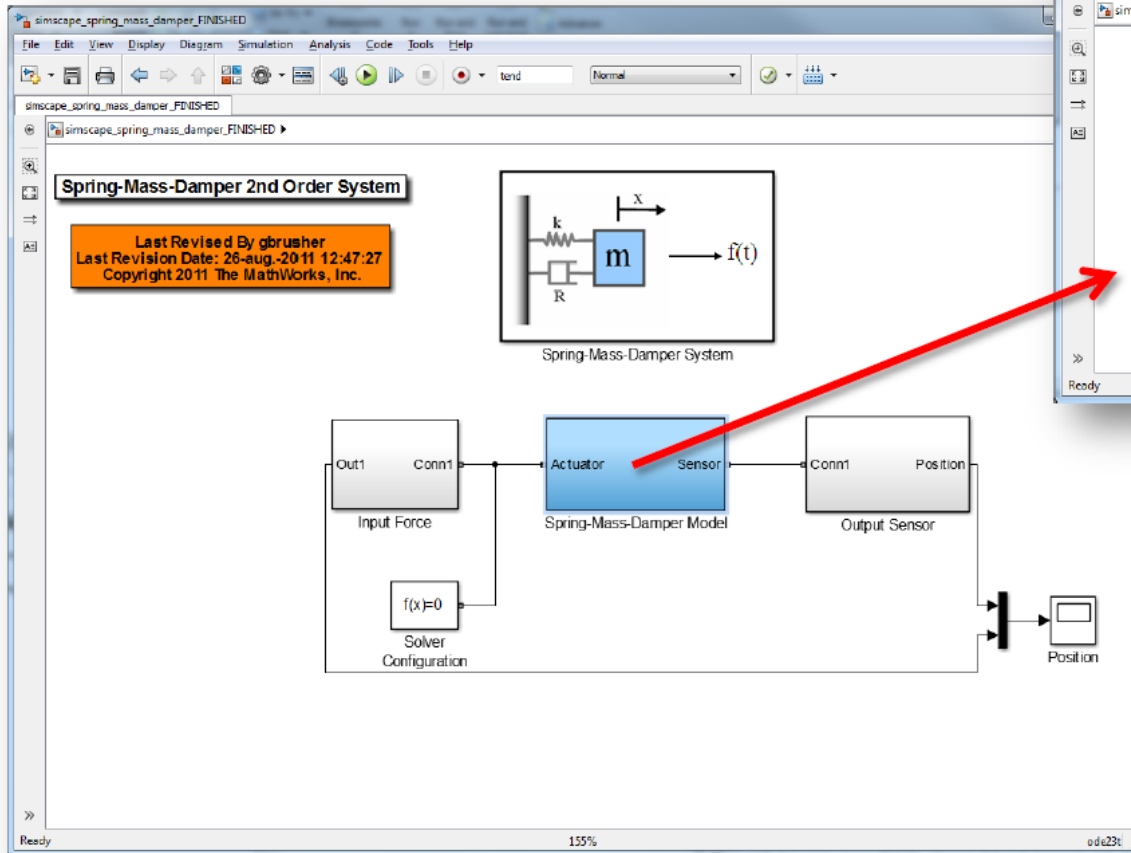


Mem 25 MB, T 0 s Outp INS

Simulink



Simscape

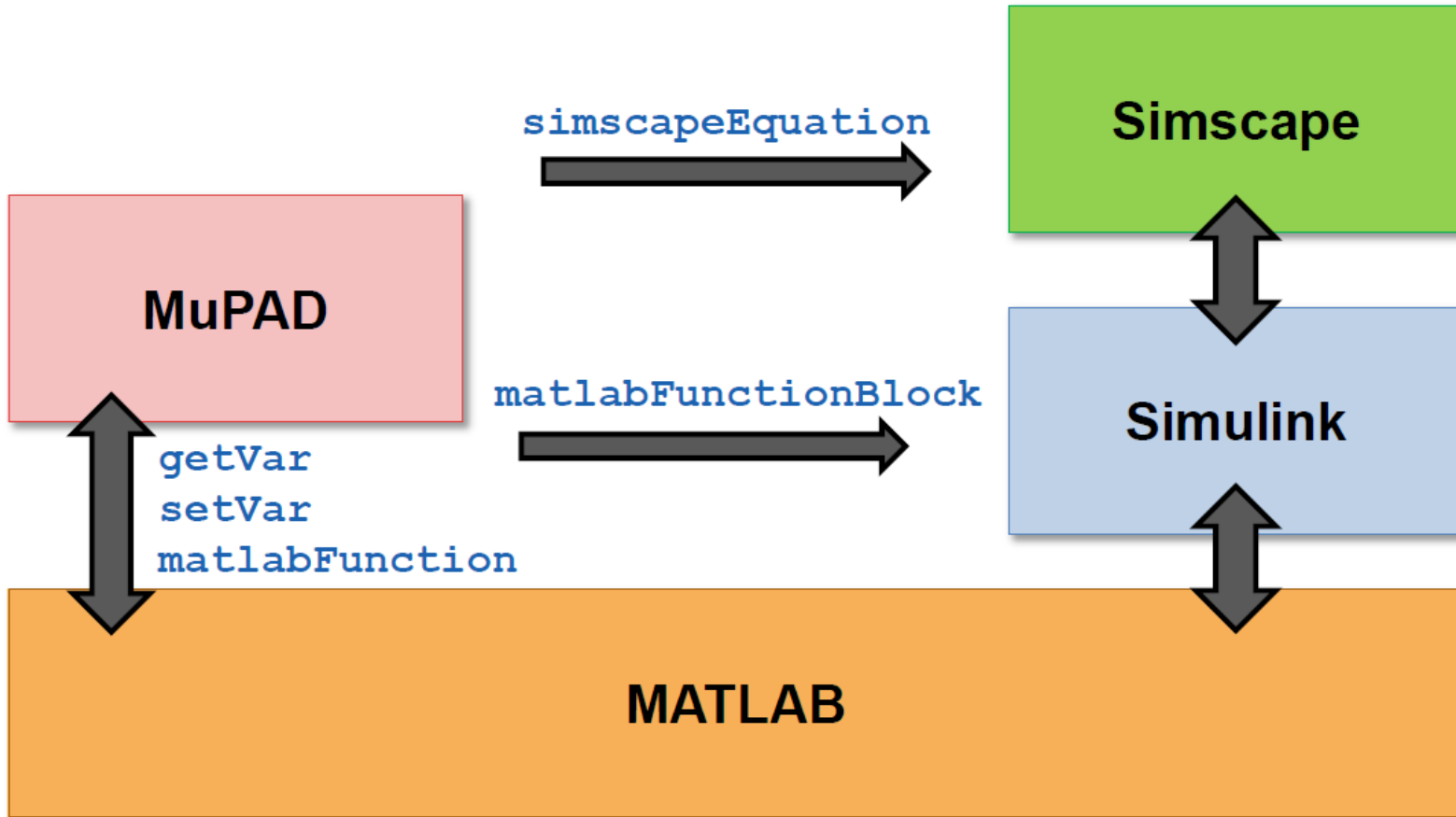


```

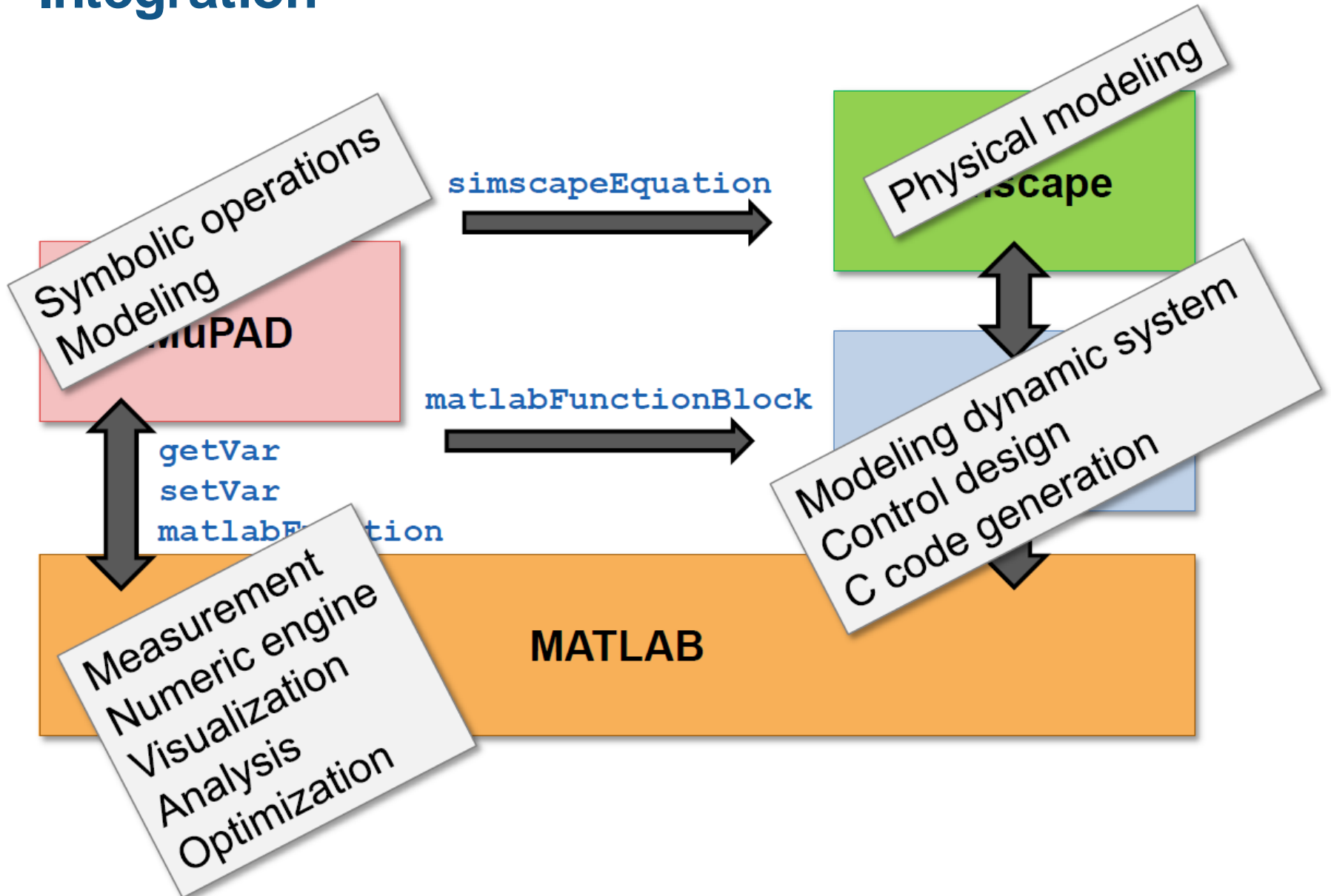
1 component spring < foundation.mechanical.translational.branch
2   % Translational Spring
3   % The block represents an ideal mechanical linear spring.
4   %
5   % Connections R and C are mechanical translational conserving p
6   % The block positive direction is from port R to port C. This m
7   % the force is positive if it acts in the direction from R to C
8   %
9   % Copyright 2005-2008 The MathWorks, Inc.
10
11 parameters
12   spr_rate = ( 1000, 'N/m' ); % Spring rate
13   init_def = ( 0, 'm' ); % Initial deformation
14 end
15
16 variables
17   x = ( 0, 'm' );
18 end
19
20 function setup
21   if spr_rate <= 0
22     pm_error( 'simscpe:GreaterThanZero', 'Spring rate' )
23   end
24   x = init_def;
25 end

```

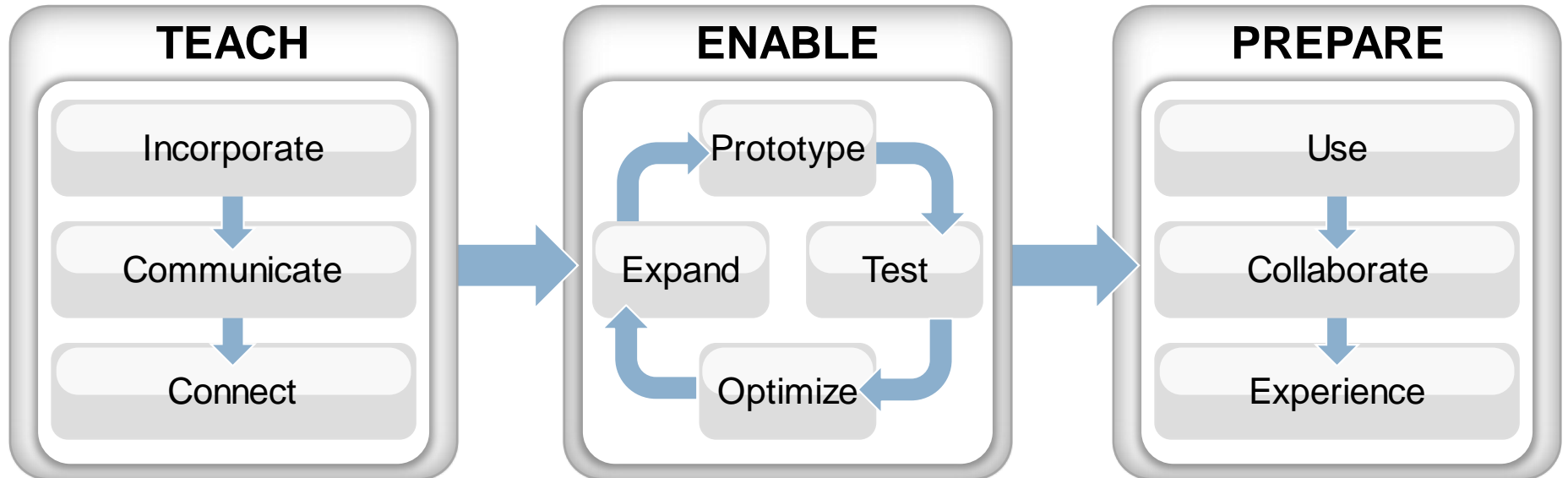
Integration



Integration



Engage Students with Modeling and Simulation



Physical Modeling Key Messages For Classrooms and Labs

Teach

1. Incorporate realistic and engaging examples
2. Communicate concepts using best available method
3. Connect theory to real systems

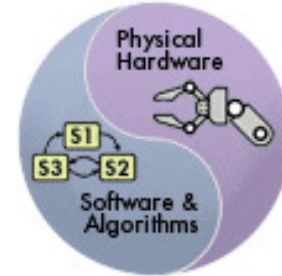
Enable

4. Expand learning to adjacent disciplines
5. Prototype new designs rapidly
6. Test designs completely
7. Optimize designs

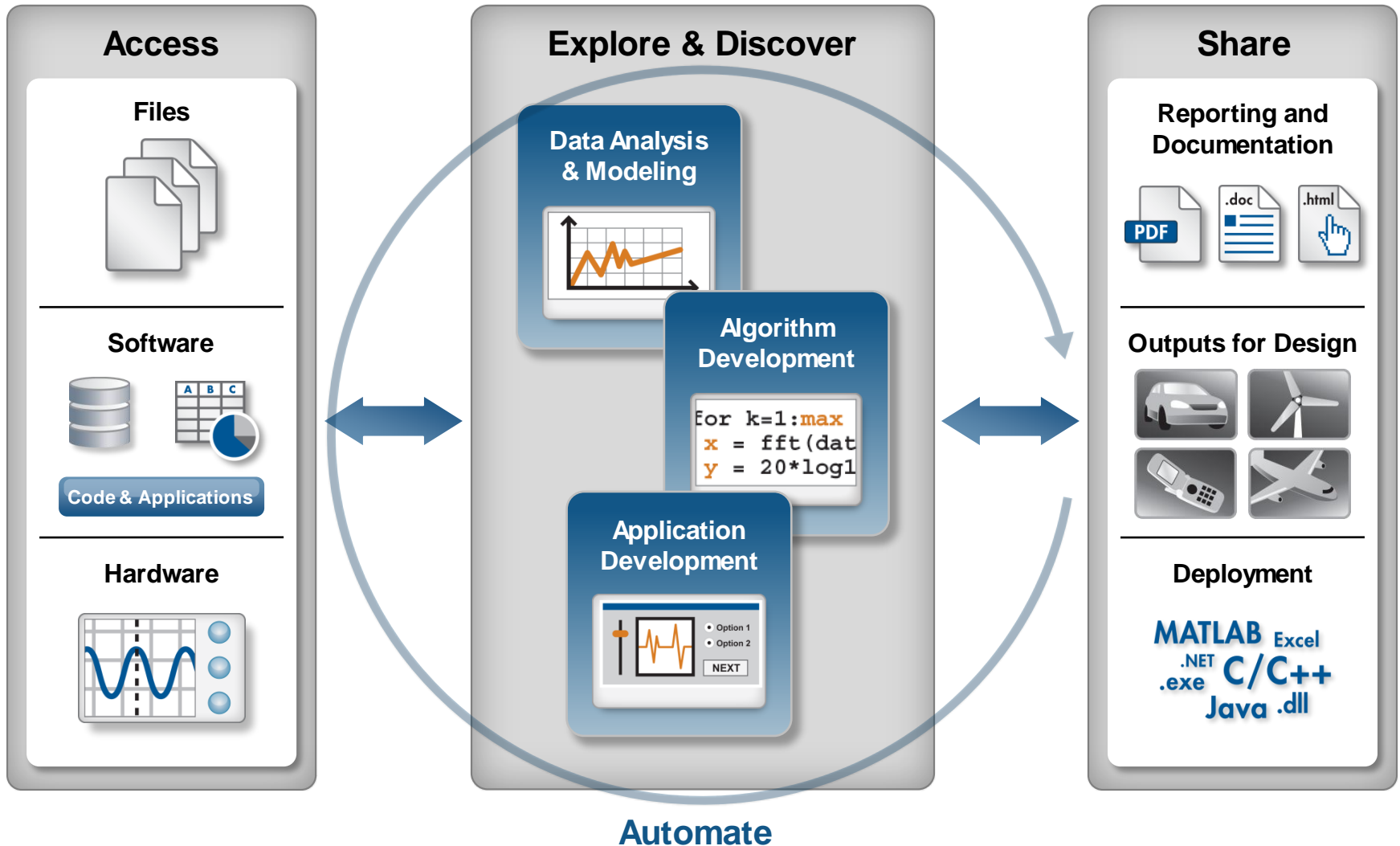
Prepare

8. Use industry-standard tools
9. Collaborate with other floors
10. Experience Model-Based Design

Engineering Systems



Technical Computing Workflow



Further Information



Accelerating the pace of engineering and science

United States | Contact Us | Store

Gergely Somlay | My Account | Log Out

Products & Services Solutions Academia Support User Community Events Company



The New MATLAB Desktop
Find what you need.

Learn more and watch videos

Latest News Optimization of Hydroelectric Flow with MATLAB 24 Oct 2012

Products

Events

Training

Explore products for MATLAB, the language of technical computing, and Simulink, for simulation and Model-Based Design.

Featured Products

Parallel Computing Toolbox
Data Acquisition Toolbox
Instrument Control Toolbox
Bioinformatics Toolbox

Simscape
HDL Coder
xPC Target
Polyspace code verifiers

[View all products or get trial software](#)

Latest Release



R2012b
Updates to MATLAB, Simulink, and 82 Other Products

[» Watch the video](#)

Featured Resource

Take the Cody Challenge!

[» Get started](#)

Cody

Recent Problems

- Fix y_correct: Haok
- AVIRIS Inscribed Rectangle Speed Test
- Functionality 1
- Now!
- Become the leader
- AVIRIS Hyperspectral

<http://www.mathworks.com>

Next steps

- Trials, license and price informations:
Attila Fekete attila.fekete@gamax.hu

Questions and Answers

Gergely Somlay

gergely.somlay@gamax.hu

Like us on Facebook

www.facebook.com/matlab.gamax