Introduction to C/C++ code generation from MATLAB code with MATLAB Coder

Generating **readable** and **portable** C/C++ code from your MATLAB algorithms

Ryan Livingston

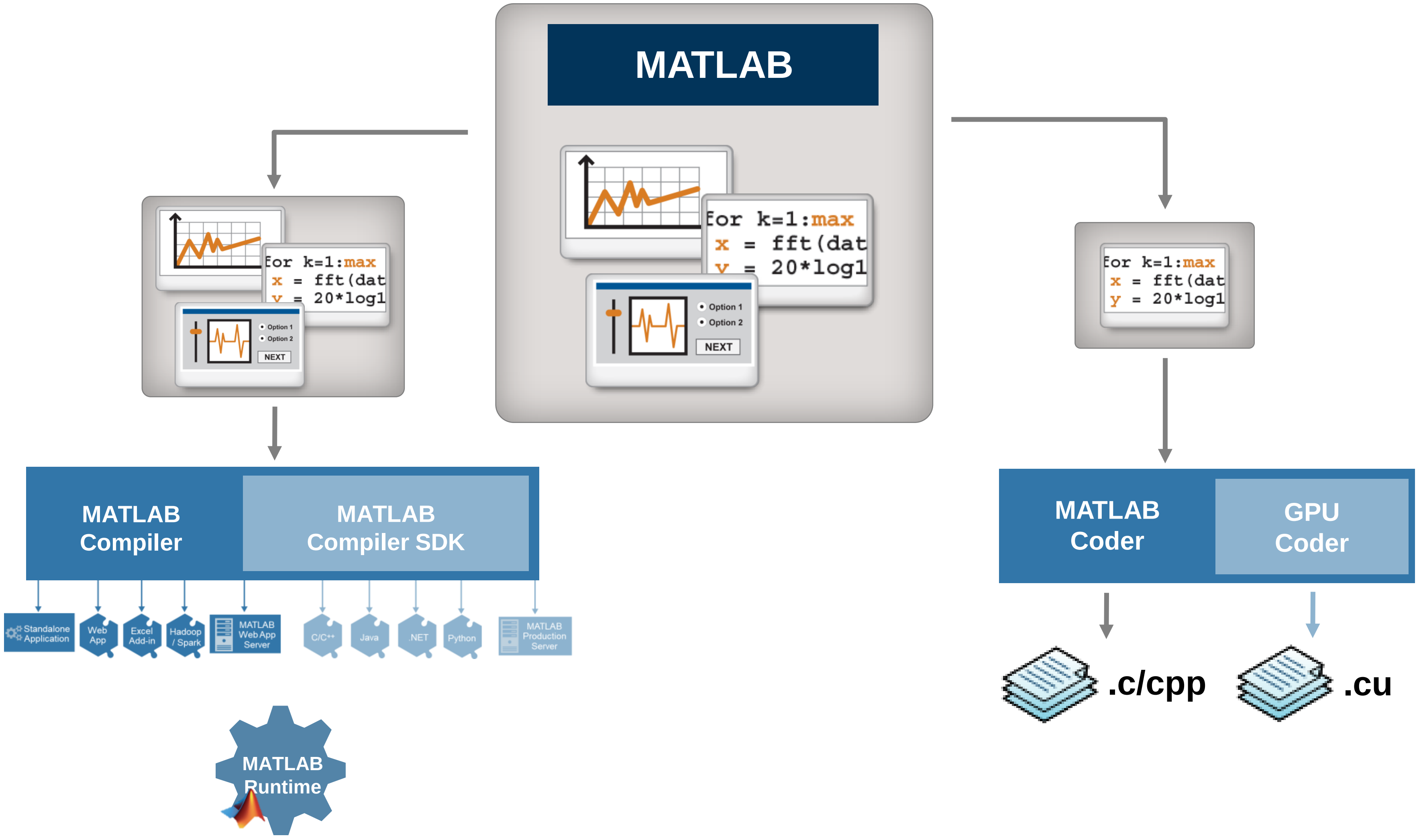
Agenda

- Motivation
  - Why translate MATLAB to C/C++?
  - Challenges of manual translation
- Using MATLAB Coder
  - Three-step workflow for generating code
- Use cases
  - Integrate algorithms using source code/libraries
  - Accelerate through MEX
  - Prototype by generating EXE
- Conclusion
  - Integration with Simulink, Embedded Coder, and GPU Coder
  - Other deployment solutions

Why Engineers Translate MATLAB to C/C++ Today

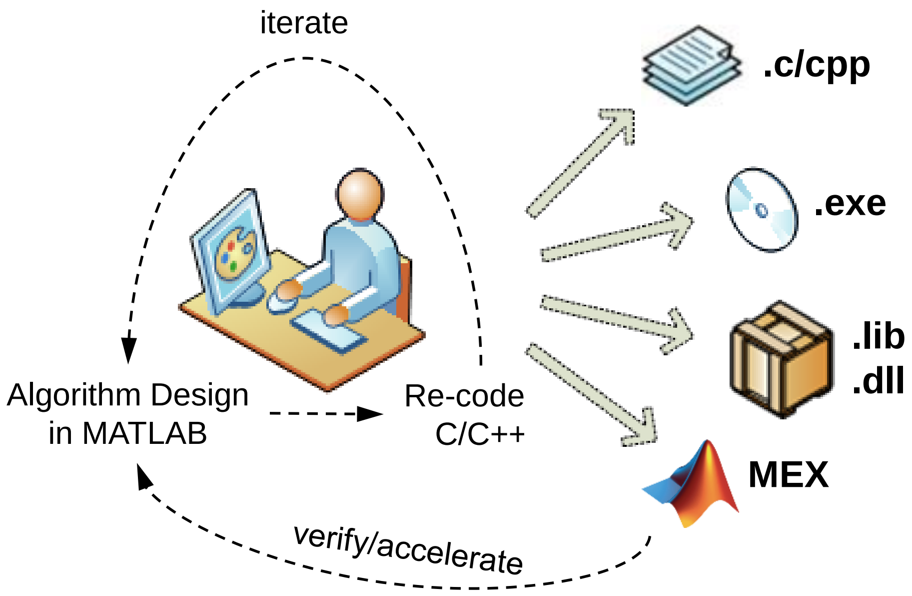
.c/cpp	**Implement** C/C++ code on processors or hand off to software engineers
.lib .dll	**Integrate** MATLAB algorithms with existing C/C++ environment using source code and static/dynamic libraries
.exe	**Prototype** MATLAB algorithms on desktops as standalone executables
MEX	**Accelerate** user-written MATLAB algorithms

Deploying MATLAB Algorithms



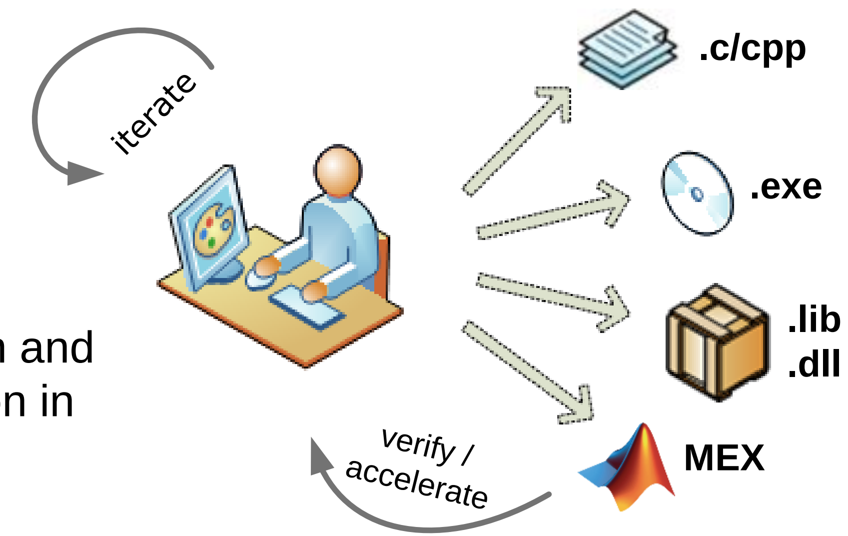
Challenges with Manual Translation

from MATLAB to C/C++



- Separate functional and implementation specification
  - Leads to multiple implementations that are inconsistent
  - Hard to modify requirements during development
  - Difficult to keep reference MATLAB code and C/C++ code in sync
- Manual coding errors
- Time-consuming and expensive process

Automatic Translation of MATLAB to C/C++



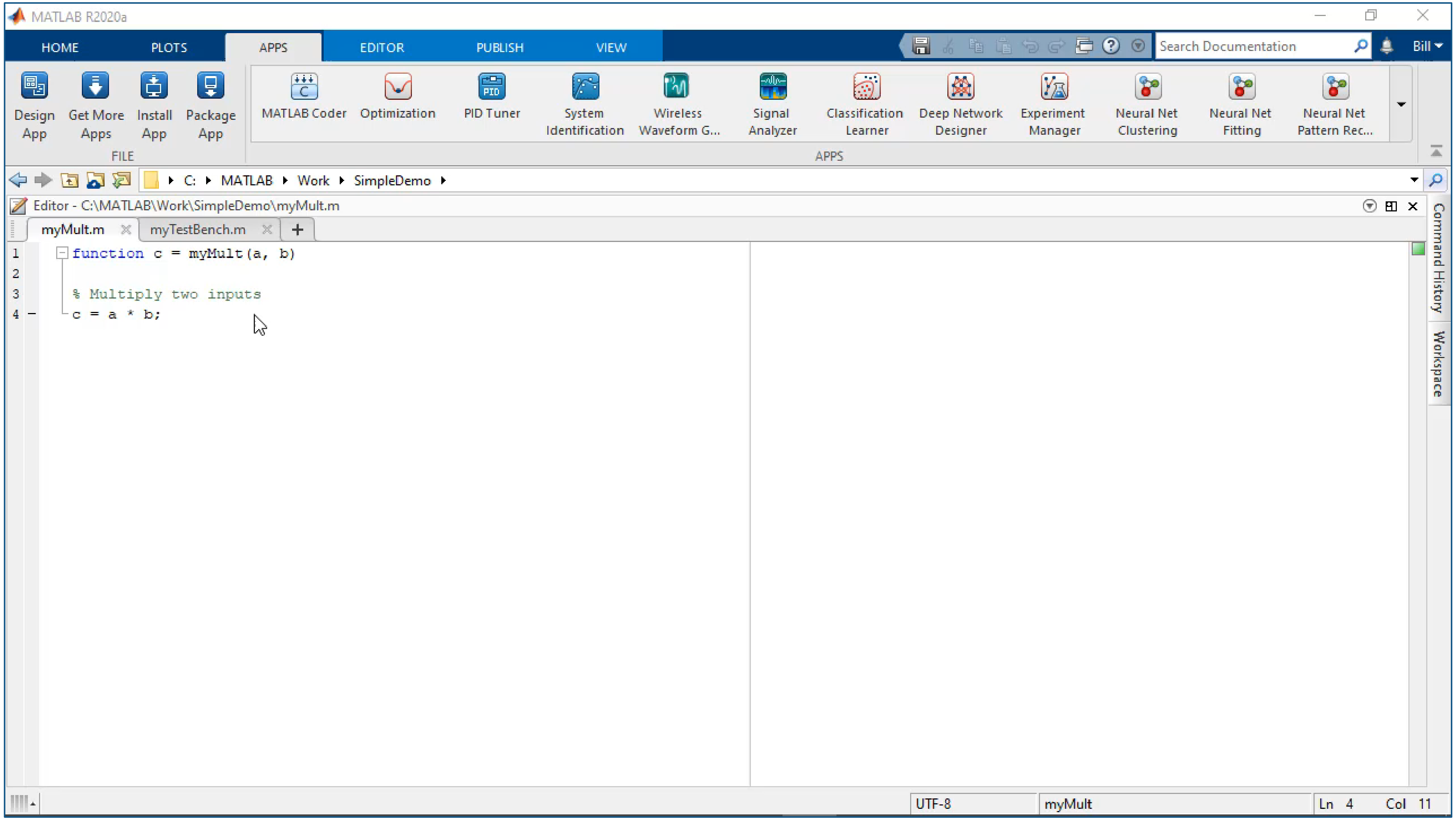
Algorithm Design and Code Generation in MATLAB

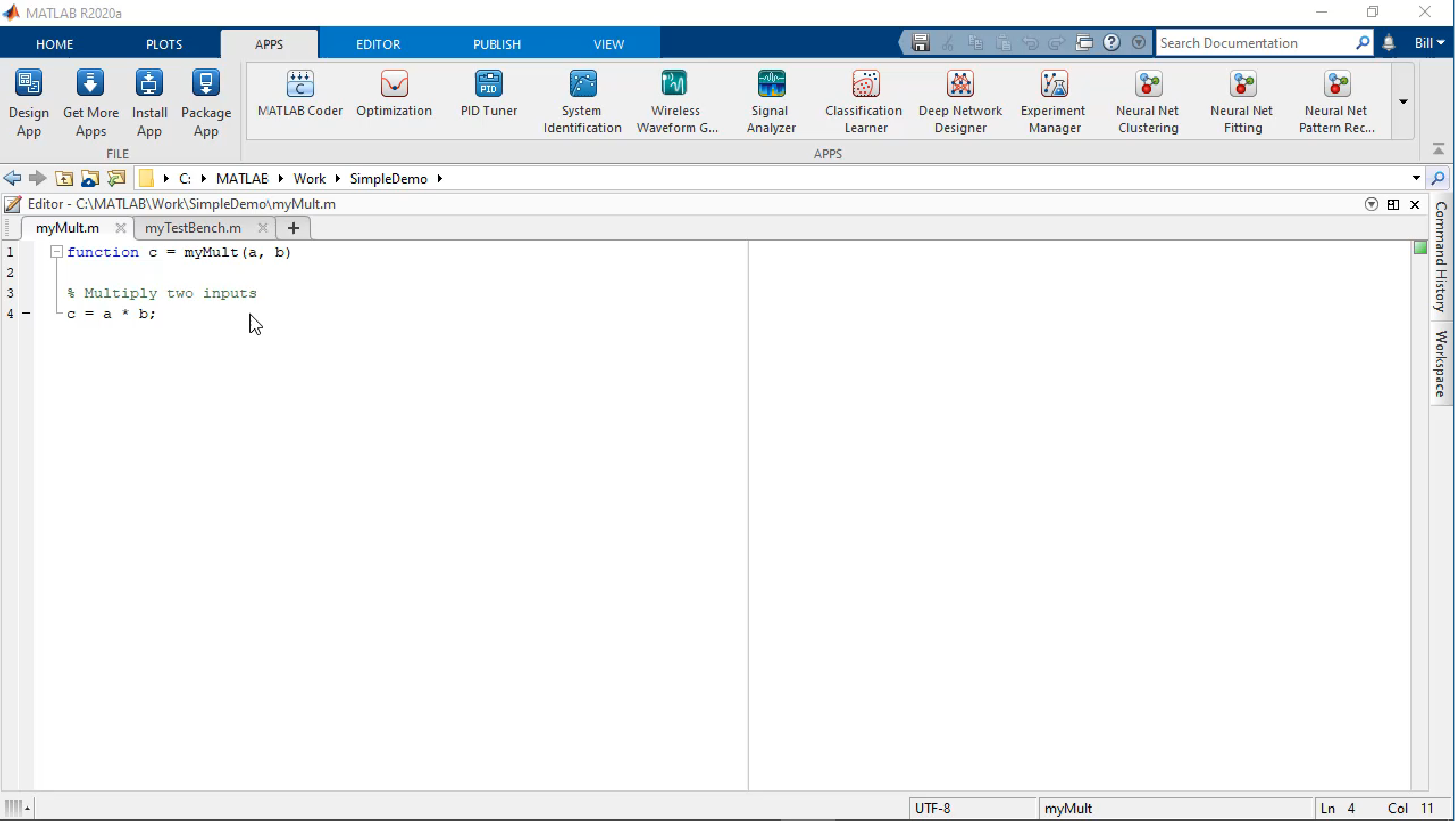
With MATLAB Coder, design engineers can:

- Maintain one design in MATLAB
- Design faster and get to C/C++ quickly
- Test more systematically and frequently
- Spend more time improving algorithms in MATLAB

Simple Example

c = a*b




MATLAB R2020a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Bill
Design App
Get More Apps
Install App
Package App
FILE
MATLAB Coder
Optimization
PID Tuner
System Identification
Wireless Waveform G...
Signal Analyzer
Classification Learner
Deep Network Designer
Experiment Manager
Neural Net Clustering
Neural Net Fitting
Neural Net Pattern Rec...
APPS
C: ▸ MATLAB ▸ Work ▸ SimpleDemo ▸
Editor - C:\MATLAB\Work\SimpleDemo\myMult.m
myMult.m
myTestBench.m
function c = myMult(a, b)
% Multiply two inputs
c = a * b;
Command History
Workspace
UTF-8
myMult
Ln 4
Col 11

Command-line Code Generation

Editor - /mathworks/devel/sandbox/rlivings/temp/mlcoder-mab-2021/myMult.m

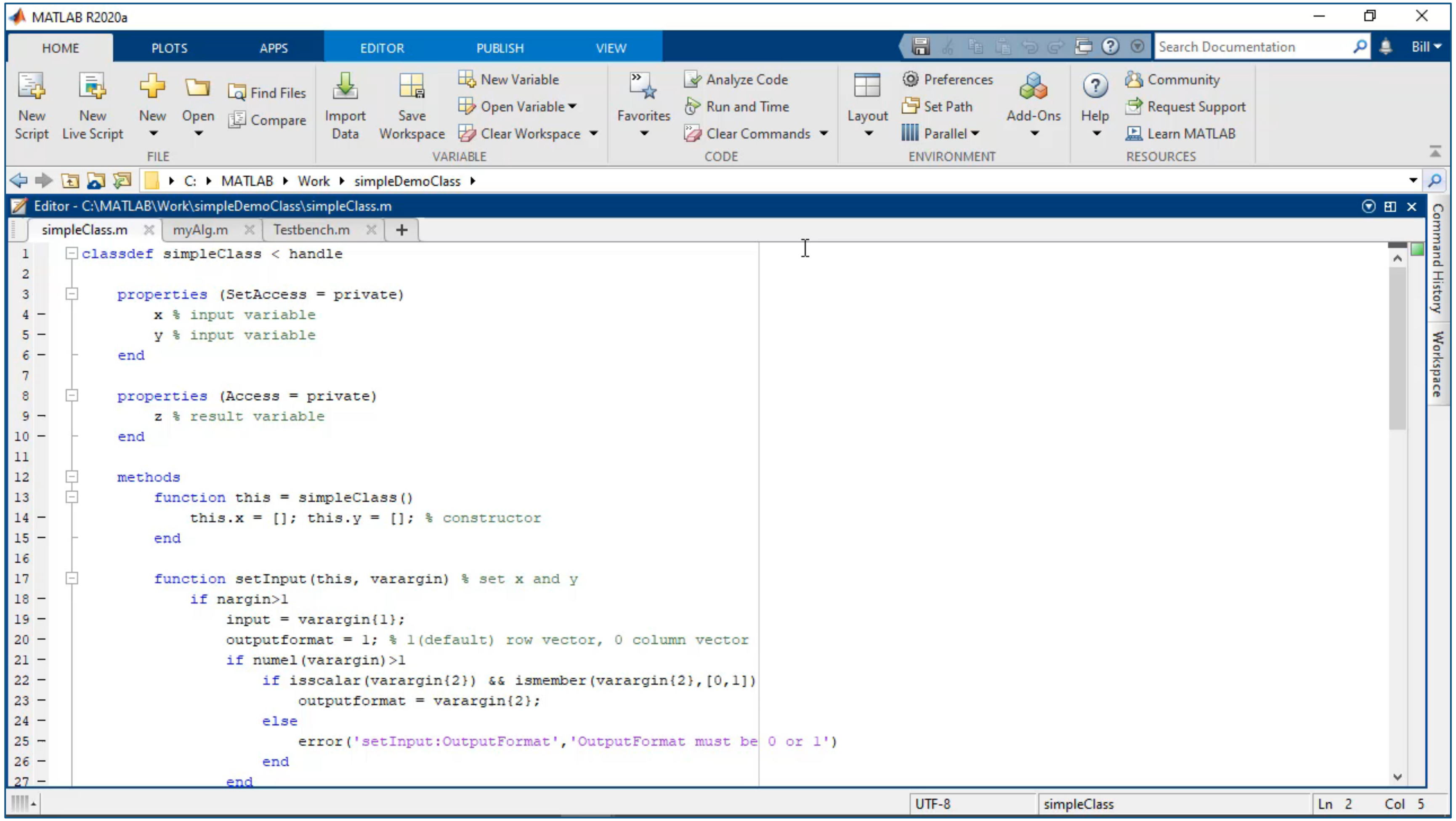
featureExtractionWithFft.m | doFeatureExtractionWithFft.m | untitled * | myMult.m

```
function c = myMult(a,b)

% Multiply two inputs
c = a*b;
```

Command Window

```
>> codegen myMult -args {zeros(3,4), zeros(4,5)} -report -config:lib
Code generation successful: View report

>>
```

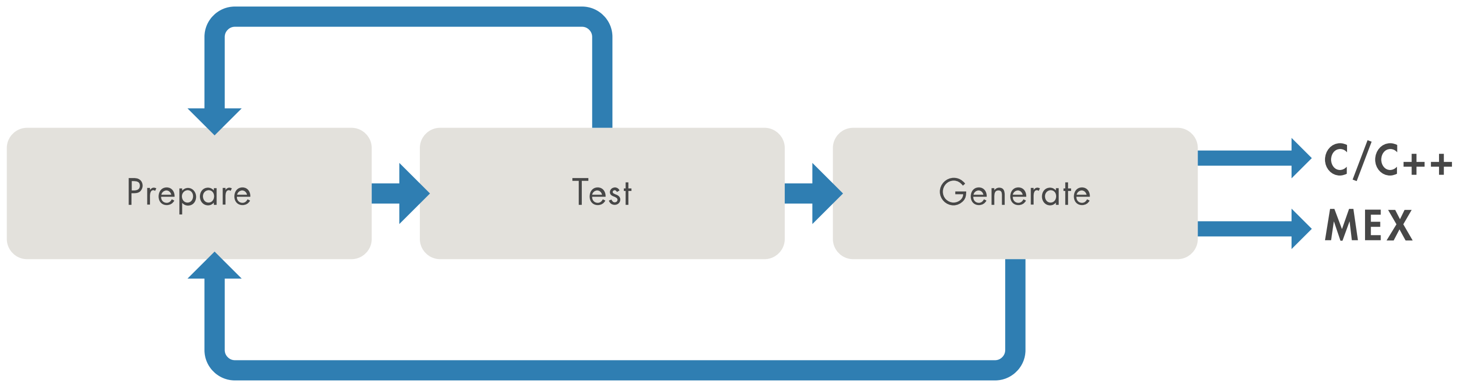
MATLAB Class to C++ Class Example



Agenda

- Motivation
  - Why translate MATLAB to C/C++?
  - Challenges of manual translation
- Using MATLAB Coder
  - Three-step workflow for generating code
- Use cases
  - Integrate algorithms using source code/libraries
  - Accelerate through MEX
  - Prototype by generating EXE
- Conclusion
  - Integration with Simulink, Embedded Coder, and GPU Coder
  - Other deployment solutions

Using MATLAB Coder: Three-Step Workflow



Prepare your MATLAB algorithm for code generation

- Make implementation choices
- Use supported language features

Test if your MATLAB code is ready for code generation

- Validate that MATLAB program generates code
- Accelerate execution of user-written algorithm

Generate source code or MEX for final use

- Iterate your MATLAB code to optimize
- Implement as source, executable, or library

Implementation Considerations

```
function a= foo(b,c)
a = b * c;
```

- Scalar multiply
- Dot product
- Matrix multiply

logical
integer
real
complex
…

C

```
double foo(double b, double c)
{
   return b*c;
}
```

```
void foo(const double b[15],
              const double c[30], double a[18])
{
  int i0, i1, i2;
  for (i0 = 0; i0 < 3; i0++) {
    for (i1 = 0; i1 < 6; i1++) {
      a[i0 + 3 * i1] = 0.0;
      for (i2 = 0; i2 < 5; i2++) {
        a[i0 + 3 * i1] += b[i0 + 3 * i2] * c[i2 + 5 * i1];
      }
    }
  }
}
```

Implementation Considerations

- Polymorphism
- Memory allocation
- Processing matrices and arrays
- Fixed-point data types

7	Lines of MATLAB
105	**Lines of C**

```matlab
function [x_est, p_est] = kalman_estimate(R,H,x_prd,p_prd,z)
S = H * p_prd' * H' + R;
B = H * p_prd';
klm_gain =  (S \ B)';
x_est = x_prd + klm_gain * (z - H * x_prd);
p_est = p_prd - klm_gain * H * p_prd;
```

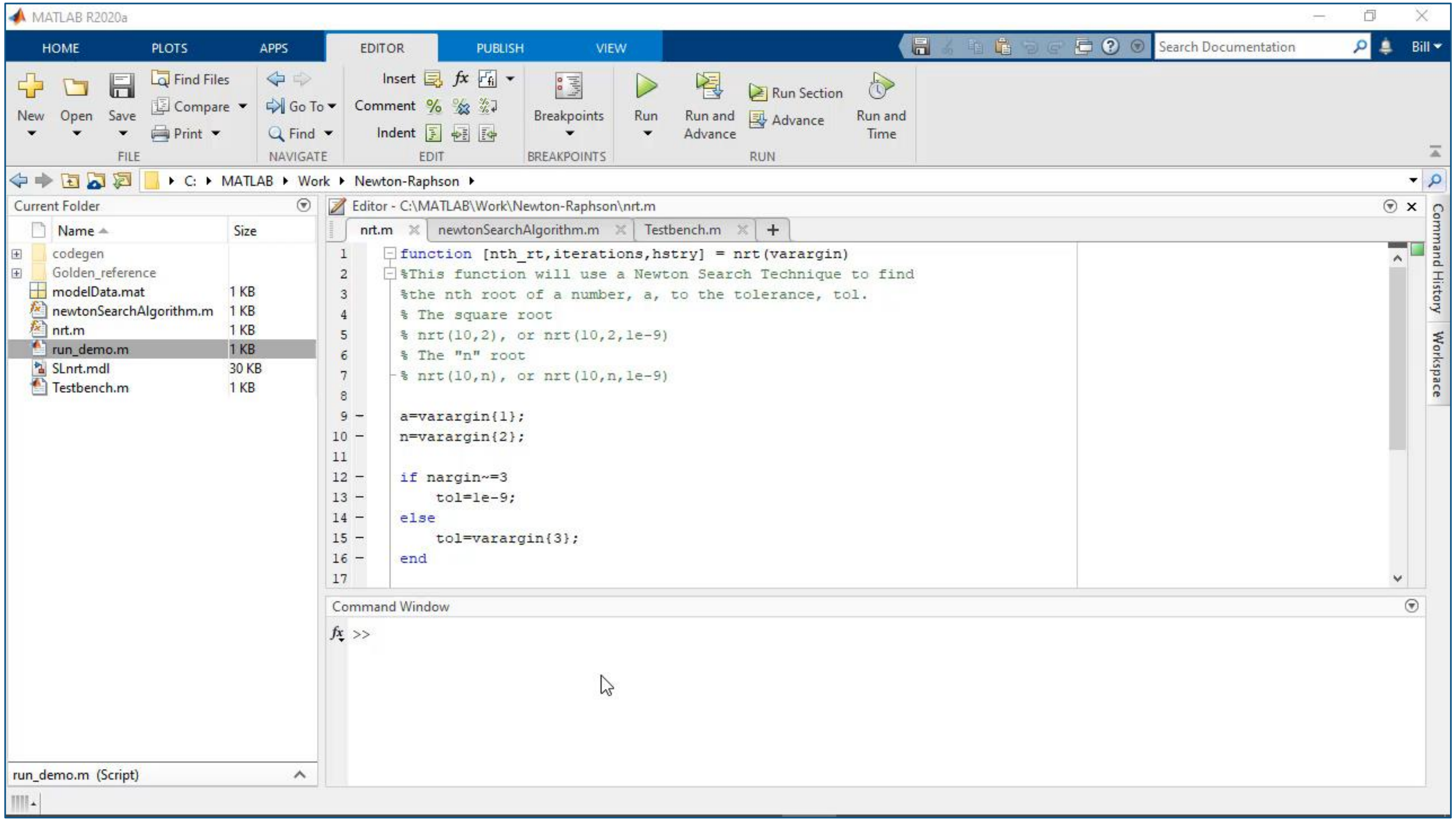
```c
#include "kalman_estimate.h"

void kalman_estimate(const do
                     const do
                     double p
{
  double klm_gain[12];
  int r1;
  int r2;
  int k;
  double S[4];
  double a21;
  double B[12];
  double a22;
  double Y[12];
  double b_z[2];
  double b_klm_gain[36];
  for (r1 = 0; r1 < 2; r1++) {
    for (r2 = 0; r2 < 6; r2++) {
      klm_gain[r1 + (r2 << 1)] = 0.0;
```

```c
      for (k = 0; k < 6; k++) {
        klm_gain[r1 + (r2 << 1)] += H[r1 + (k << 1)] *
      }
    }
  }

  for (r1 = 0; r1 < 2; r1++) {
    for (r2 = 0; r2 <
      a21 = 0.0;
      for (k = 0; k <
        a21 += klm_ga
      }

      S[r1 + (r2 << 1
    }
  }

  for (r1 = 0; r1 < 2
    for (r2 = 0; r2 <
```

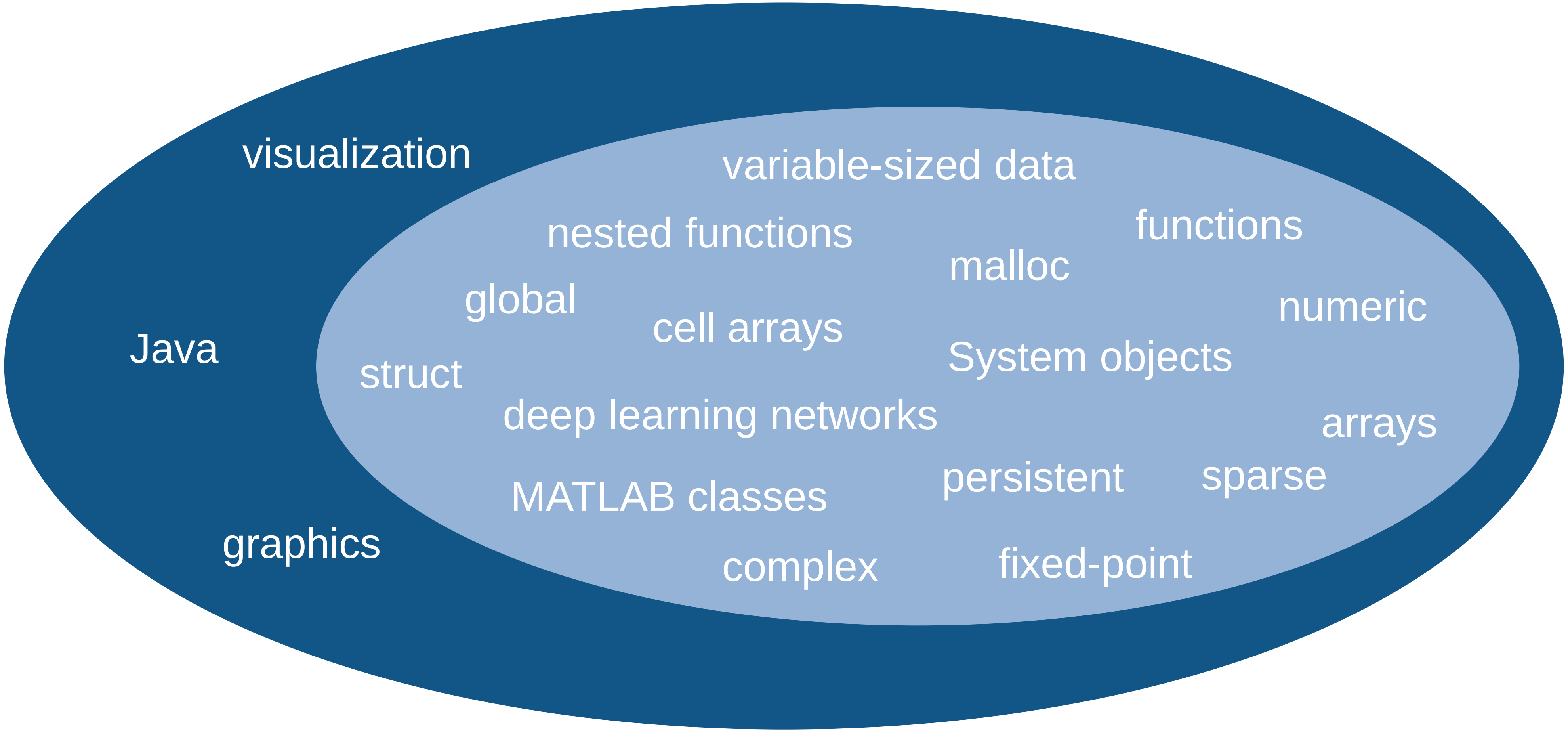
```c
  if (fabs(S[1]) > fabs(S[0])) {
    r1 = 1;
    r2 = 0;
  } else {
    r1 = 0;
    r2 = 1;
  }

  a21 = S[r2] / S[r1];
  a22 = S[2 + r2] - a21 * S[2 + r1
  for (k = 0; k < 6; k++) {
    Y[1 + (k << 1)] = (B[r2 + (k << 1)] - B[r1 + (k << 1)] * a21) / a22;
    Y[k << 1] = (B[r1 + (k << 1)] - Y[1 + (k << 1)] * S[2 + r1]) / S[r1];
  }

  for (r1 = 0; r1 < 2; r1++) {
    for (r2 = 0; r2 < 6; r2++) {
      klm_gain[r2 + 6 * r1] = Y[r1 + (r2 << 1)];
    }
  }
```

```c
  for (r1 = 0; r1 < 6; r1++) {
    for (r2 = 0; r2 < 6; r2++) {
      b_klm_gain[r1 + 6 * r2] = 0.0;
      for (k = 0; k < 2; k++) {
        b_klm_gain[r1 + 6 * r2] += klm_gain[r1 + 6 * k] * H[k + (r2 << 1)];
      }
    }
  }

  for (r1 = 0; r1 < 6; r1++) {
    for (r2 = 0; r2 < 6; r2++) {
      a21 = 0.0;
      for (k = 0; k < 6; k++) {
        a21 += b_klm_gain[r1 + 6 * k] * p_prd[k + 6 * r2];
      }

      p_est[r1 + 6 * r2] = p_prd[r1 + 6 * r2] - a21;
    }
  }
}
```

Newton/Raphson Example



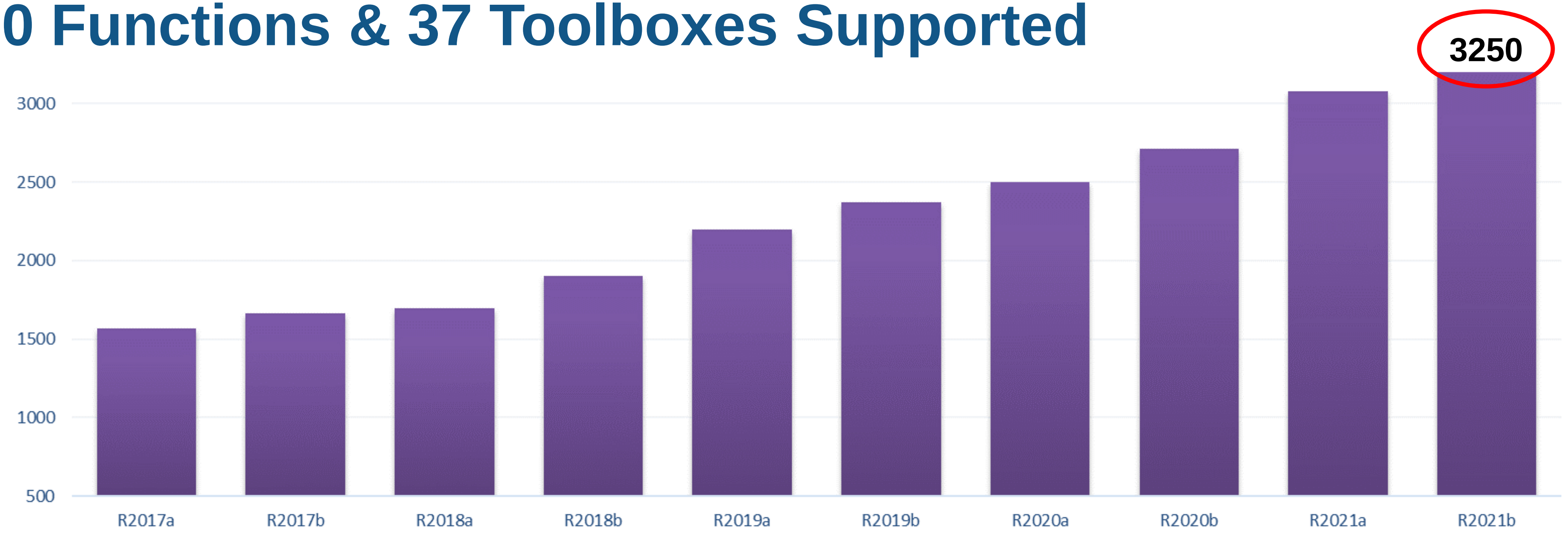
Growing MATLAB Language Support for Code Generation



MathWorks

R2021b

3250 Functions & 37 Toolboxes Supported



- 5G Toolbox
- Aerospace Toolbox
- Antenna Toolbox
- Audio System Toolbox
- Automated Driving Toolbox
- Communications Toolbox
- Computer Vision Toolbox
- Control System Toolbox
- Deep Learning Toolbox
- DSP System Toolbox
- Fixed-Point Designer
- Fuzzy Logic Toolbox
- Image Acquisition Toolbox
- Image Processing Toolbox
- Instrumental Control Toolbox
- Lidar Toolbox R2021a
- Mapping Toolbox R2021a
- Mixed-Signal Blockset
- Model Predictive Control Toolbox
- Navigation Toolbox
- Optimization Toolbox
- Phased Array System Toolbox
- Predictive Maintenance Toolbox R2021a
- Radar Toolbox R2021a
- Reinforcement Learning Toolbox R2021b
- Robotics System Toolbox
- ROS Toolbox
- Satellite Communications Toolbox
- Sensor Fusion and Tracking Toolbox
- SerDes Toolbox
- Signal Processing Toolbox
- Stats & Machine Learning Toolbox
- System Identification Toolbox
- UAV Toolbox
- Vision HDL Toolbox R2021b
- Wavelet Toolbox
- WLAN System Toolbox

Agenda

- Motivation
  - Why translate MATLAB to C/C++?
  - Challenges of manual translation
- Using MATLAB Coder
  - Three-step workflow for generating code
- Use cases
  - Integrate algorithms using source code/libraries
  - Accelerate through MEX
  - Prototype by generating EXE
- Conclusion
  - Integration with Simulink, Embedded Coder, and GPU Coder
  - Other deployment solutions

MATLAB Coder Use Cases

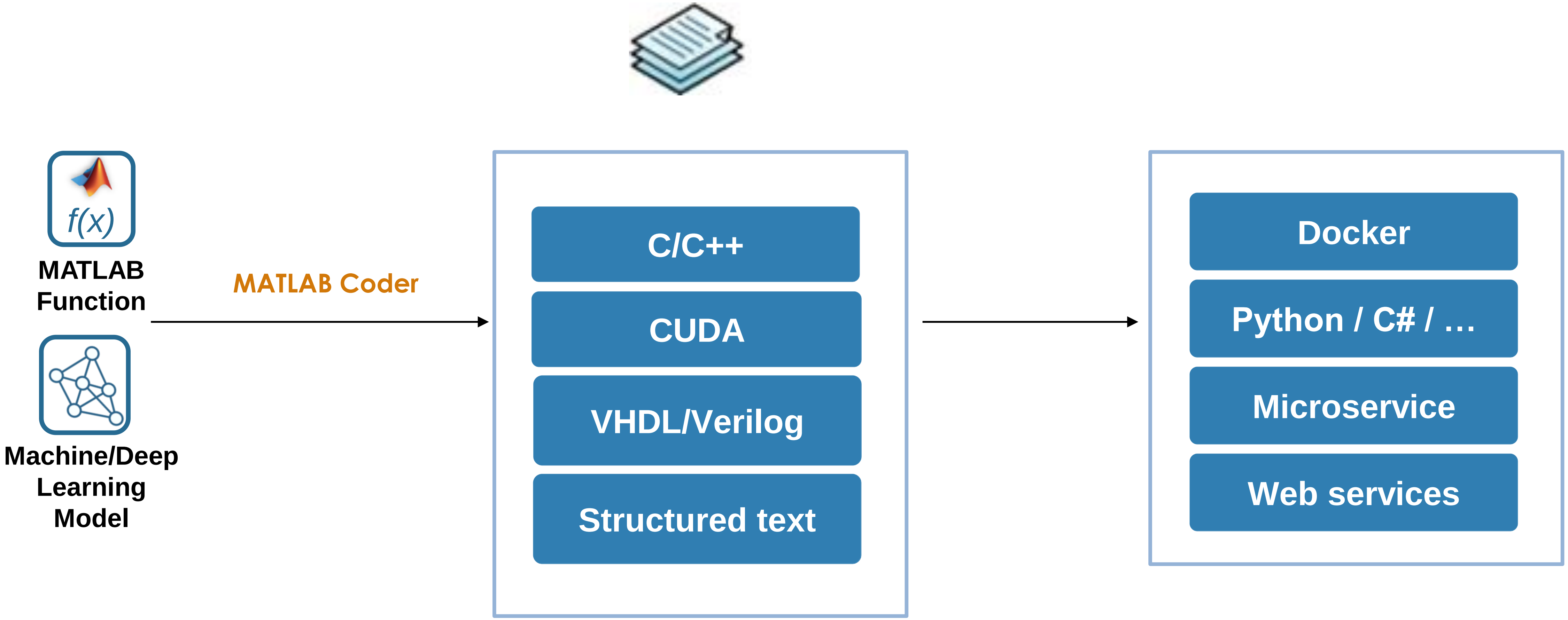
.lib
.dll

.exe

MEX

.c/cpp

Integrate Generated Code with Other Systems



Building a Container from Generated Code

Dockerfile (handwritten)

```
FROM ubuntu:18.04
LABEL Name=mlcdockerdemo Version=0.0.1
RUN apt-get -y update
WORKDIR /workdir
COPY ./generatedApplication .
CMD ["sh", "-c", "./generatedApplication"]
```

.cpp

.lib
.dll

MATLAB Coder

f(x)

MATLAB Function

main.cpp (handwritten)

Communication

Driver

Example Python Bindings

```matlab
function y = timestwo(x)
y = 2*x;
```

coder-swig on github:

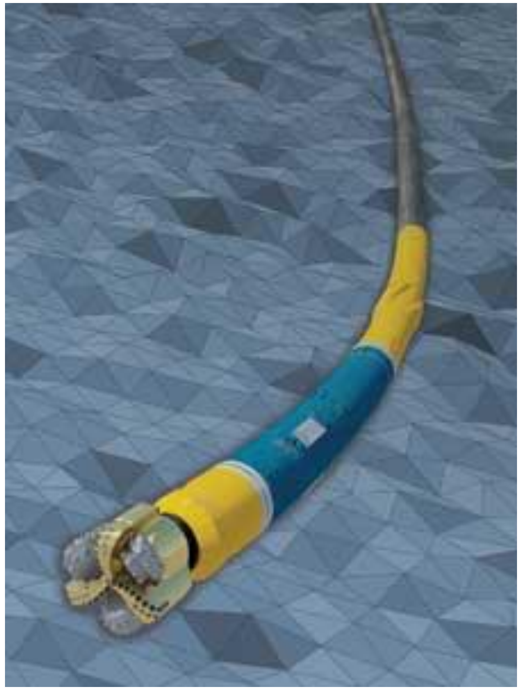
https://github.com/mathworks/coder-swig

```python
def main():
    "Main function to test timestwo generated code"
    from timestwoPython import timestwo

    # Call initialize function to set up state
    print "Calling initialize"
    timestwo.timestwo_initialize()
    input = 3.0;
    print "Input = {0:g}".format(input)

    #Call entry-point
    result = timestwo.timestwo(input)

    print "Result = {0:g}".format(result)

    # Call terminate function to perform clean up
    print "Calling terminate"
    timestwo.timestwo_terminate()

if __name__ == "__main__":
    main()
```

Examples of MATLAB Coder Usage

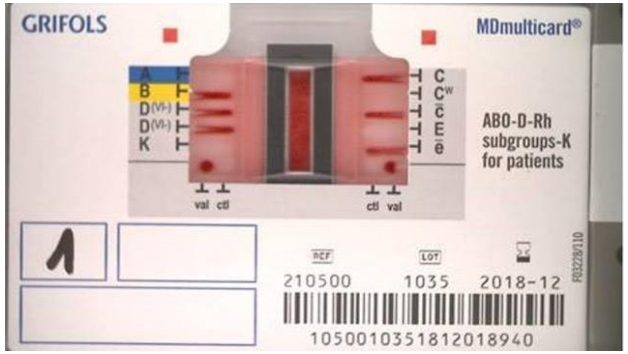
Integrate
algorithms with custom software

Implement
algorithms on embedded processors



Baker Hughes

Qualcomm



Idneo

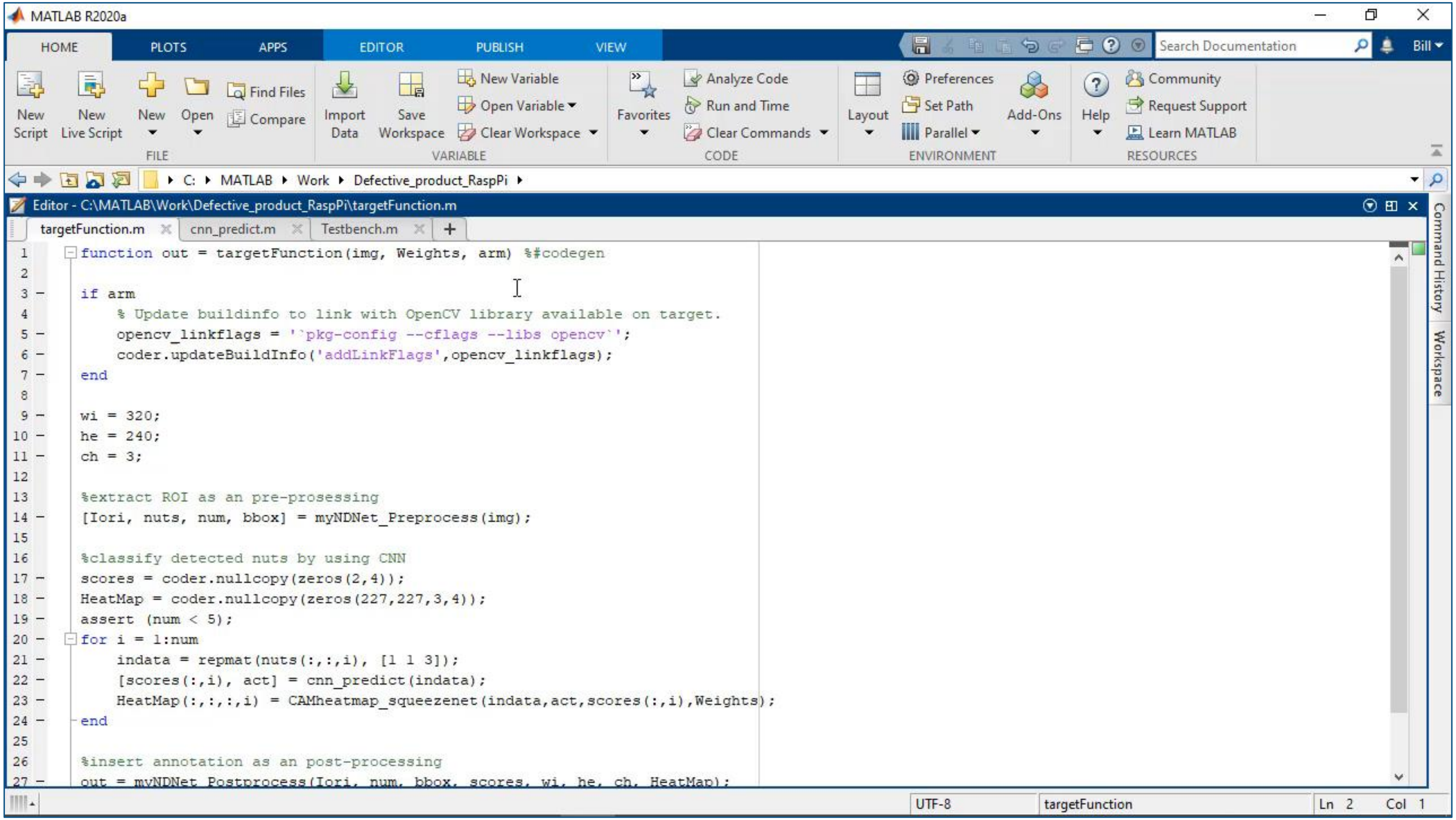
Delphi

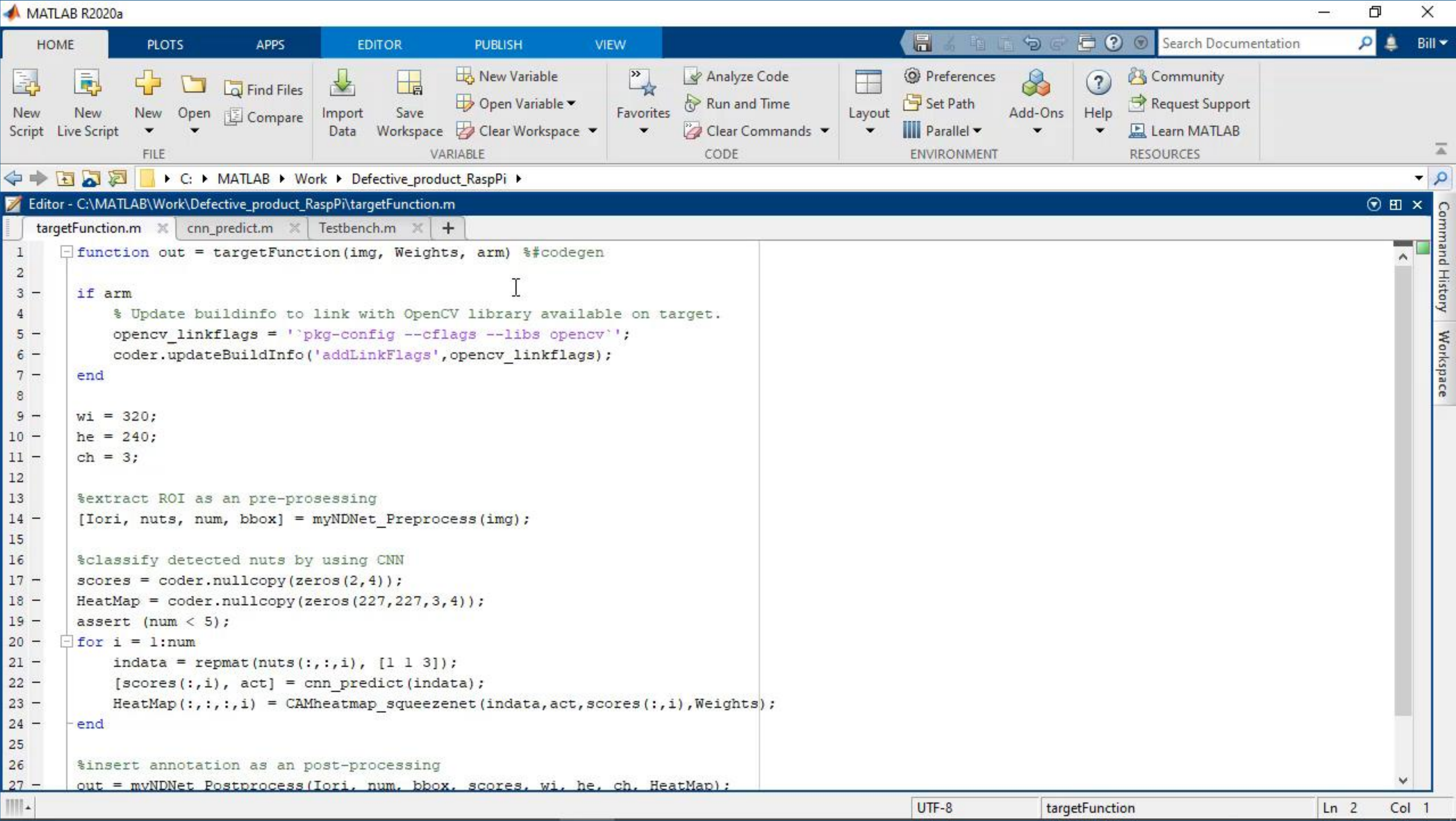
dorsaVi

Respiri

Deep Learning on Raspberry Pi Example

Implement
algorithms on embedded processors




MATLAB R2020a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Bill
New Script
New Live Script
New
Open
Find Files
Compare
FILE
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
VARIABLE
Favorites
Analyze Code
Run and Time
Clear Commands
CODE
Layout
Preferences
Set Path
Parallel
Add-Ons
ENVIRONMENT
Help
Community
Request Support
Learn MATLAB
RESOURCES
C: ▸ MATLAB ▸ Work ▸ Defective_product_RaspPi ▸
Editor - C:\MATLAB\Work\Defective_product_RaspPi\targetFunction.m
targetFunction.m
cnn_predict.m
Testbench.m
function out = targetFunction(img, Weights, arm) %#codegen

if arm
    % Update buildinfo to link with OpenCV library available on target.
    opencv_linkflags = '`pkg-config --cflags --libs opencv`';
    coder.updateBuildInfo('addLinkFlags',opencv_linkflags);
end

wi = 320;
he = 240;
ch = 3;

%extract ROI as an pre-prosessing
[Iori, nuts, num, bbox] = myNDNet_Preprocess(img);

%classify detected nuts by using CNN
scores = coder.nullcopy(zeros(2,4));
HeatMap = coder.nullcopy(zeros(227,227,3,4));
assert (num < 5);
for i = 1:num
    indata = repmat(nuts(:,:,i), [1 1 3]);
    [scores(:,i), act] = cnn_predict(indata);
    HeatMap(:,:,:,i) = CAMheatmap_squeezenet(indata,act,scores(:,i),Weights);
end

%insert annotation as an post-processing
out = myNDNet_Postprocess(Iori, num, bbox, scores, wi, he, ch, HeatMap);
Command History
Workspace
UTF-8
targetFunction
Ln 2
Col 1

Acceleration Strategies

Accelerate
algorithm execution

- Better algorithms

  Matrix inversion vs. QR or SVD

  – Different approaches to solving the same problem

- More efficient implementation

  Hand-coded vs. optimized library (BLAS and LAPACK)

  – Different optimization of the same algorithm

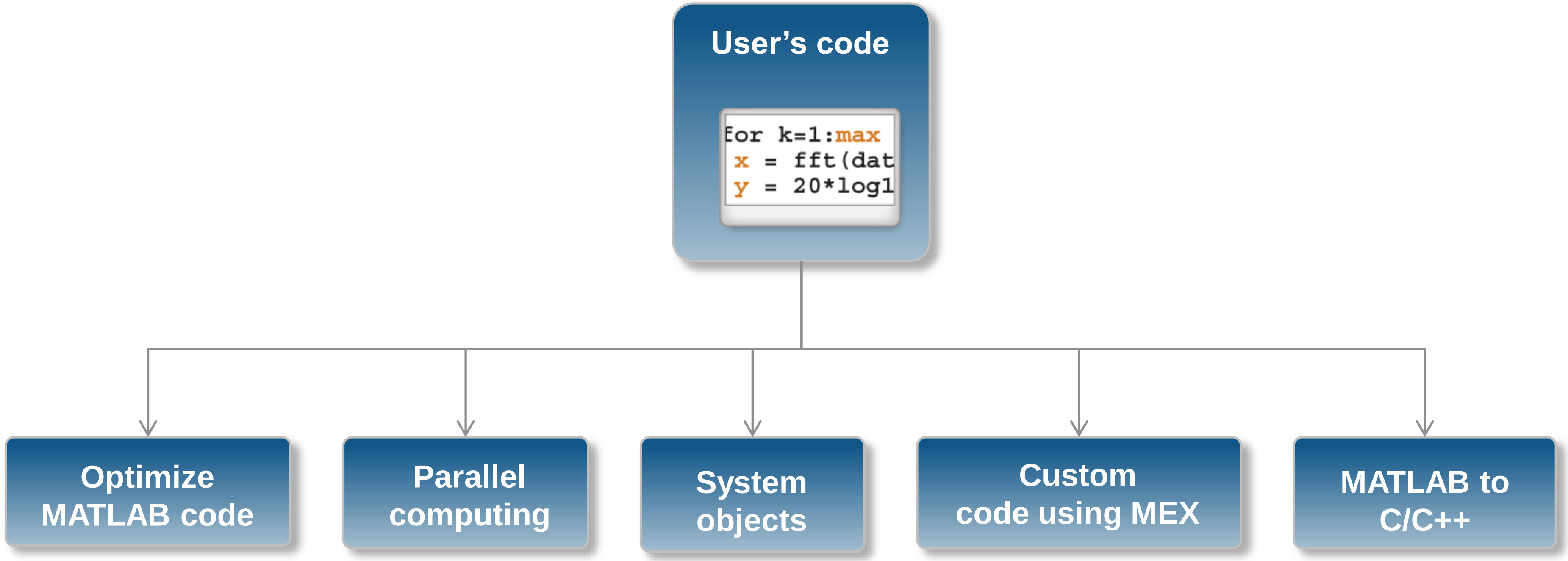
- More computational resources

  Single-threaded vs. multithreaded (multithreaded BLAS)

  – Leveraging additional processors, cores, GPUs, FPGAs, etc.

Accelerating Algorithm Execution

Accelerate
algorithm execution



Acceleration Using MEX

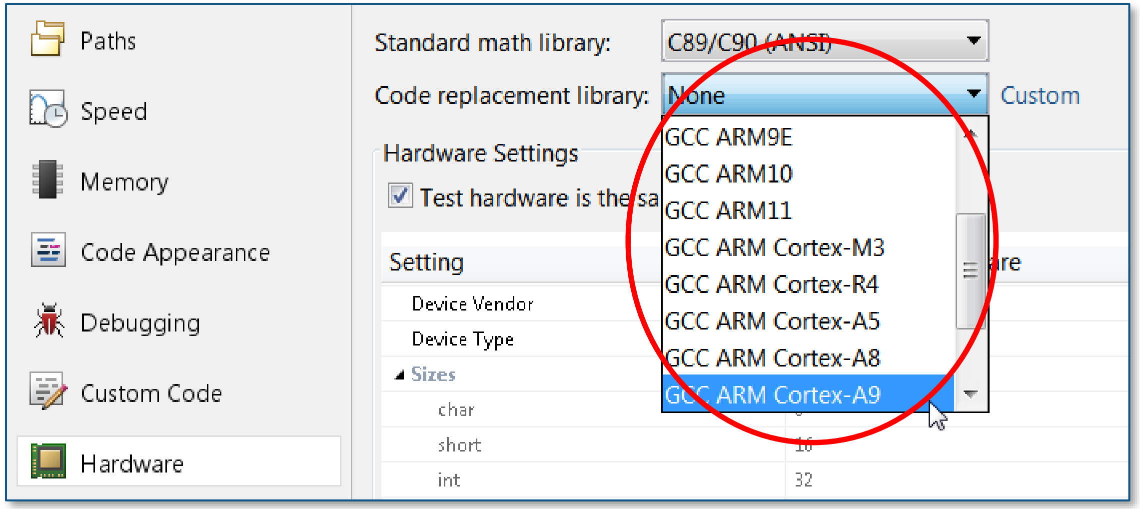
Accelerate
algorithm execution

- Speed-up factor will vary
- When you **may** see a speedup:
  - Often for communications and signal processing
  - Always for fixed point
  - Likely for loops with states or when vectorization isn't possible
- When you **may not** see a speedup:
  - MATLAB implicitly multithreads computation.
  - Built-functions call IPP or BLAS libraries.

Agenda

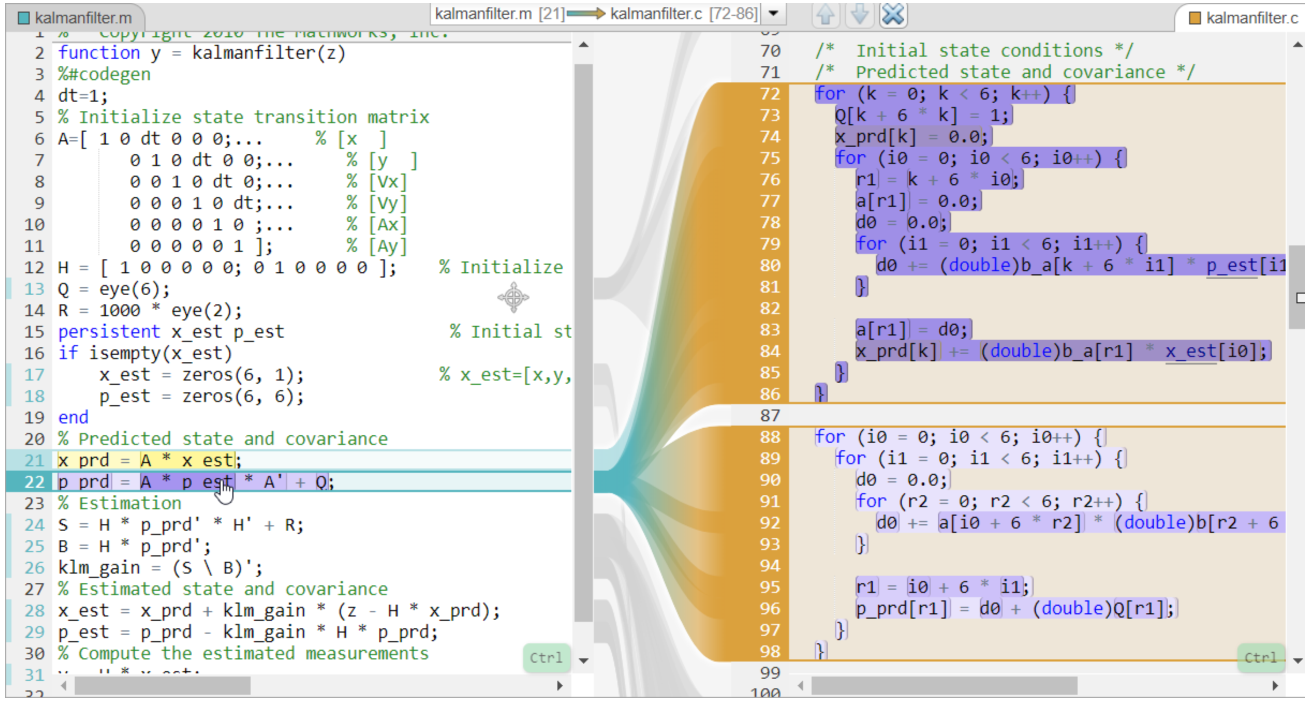
- Motivation
  - Why translate MATLAB to C/C++?
  - Challenges of manual translation
- Using MATLAB Coder
  - Three-step workflow for generating code
- Use cases
  - Integrate algorithms using source code/libraries
  - Accelerate through MEX
  - Prototype by generating EXE
- Conclusion
  - Integration with Simulink, Embedded Coder, and GPU Coder
  - Other deployment solutions

Working with Embedded Coder



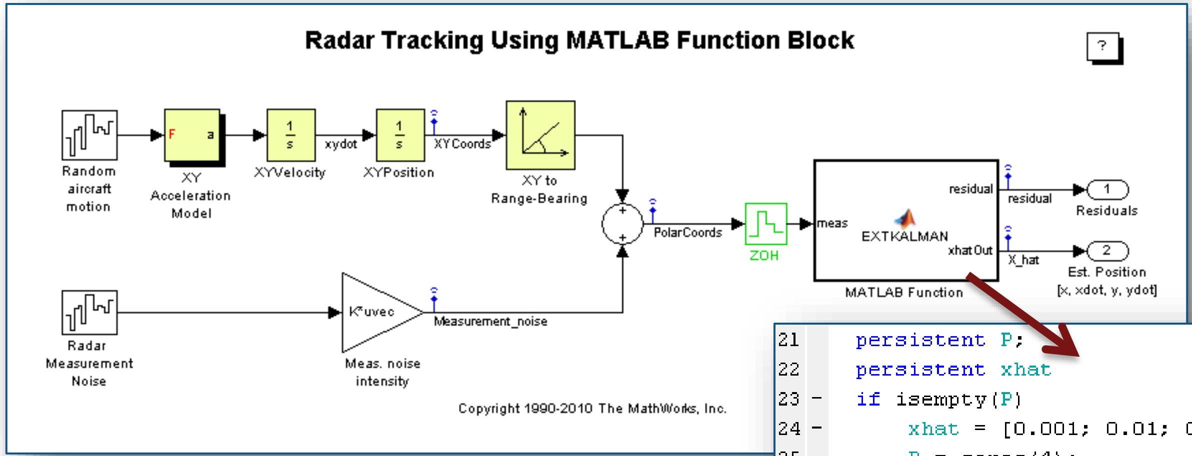
Advanced support for MATLAB Coder, including:

- Speed & Memory
- Hardware-specific optimization
- Code appearance
- Bidirectional traceability
- Software/Processor-in-the-loop verification
- Execution profiling



Working with Simulink and Embedded Coder

MATLAB Function block in Simulink

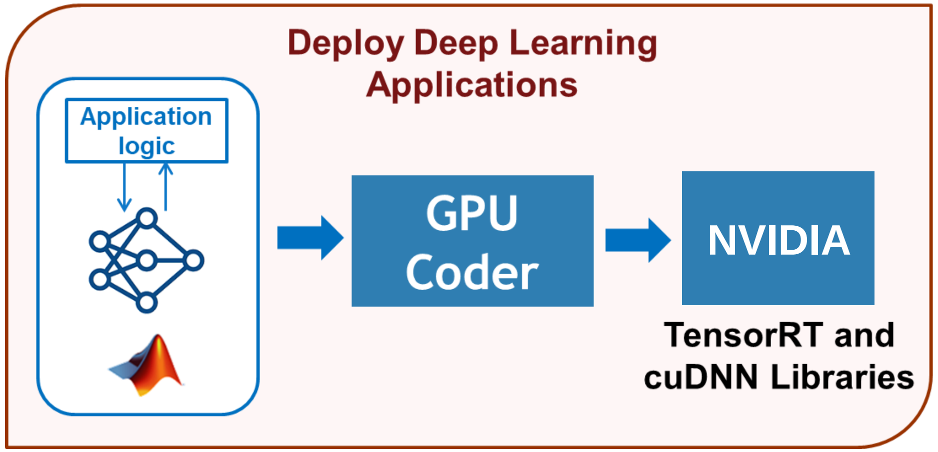



```matlab
persistent P;
persistent xhat
if isempty(P)
    xhat = [0.001; 0.01; 0.001; 400];
    P = zeros(4);
end

%  Radar update time deltat is inherited from mo

% 1. Compute Phi, Q, and R
Phi = [1 deltat 0 0; 0 1 0 0 ; 0 0 1 deltat; 0 0
Q =  diag([0 .005 0 .005]);
R =  diag([300^2 0.001^2]);

% 2. Propagate the covariance matrix:
P = Phi*P*Phi' + Q;

% 3. Propagate the track estimate::
xhat = Phi*xhat;

% 4 a). Compute observation estimates:
```

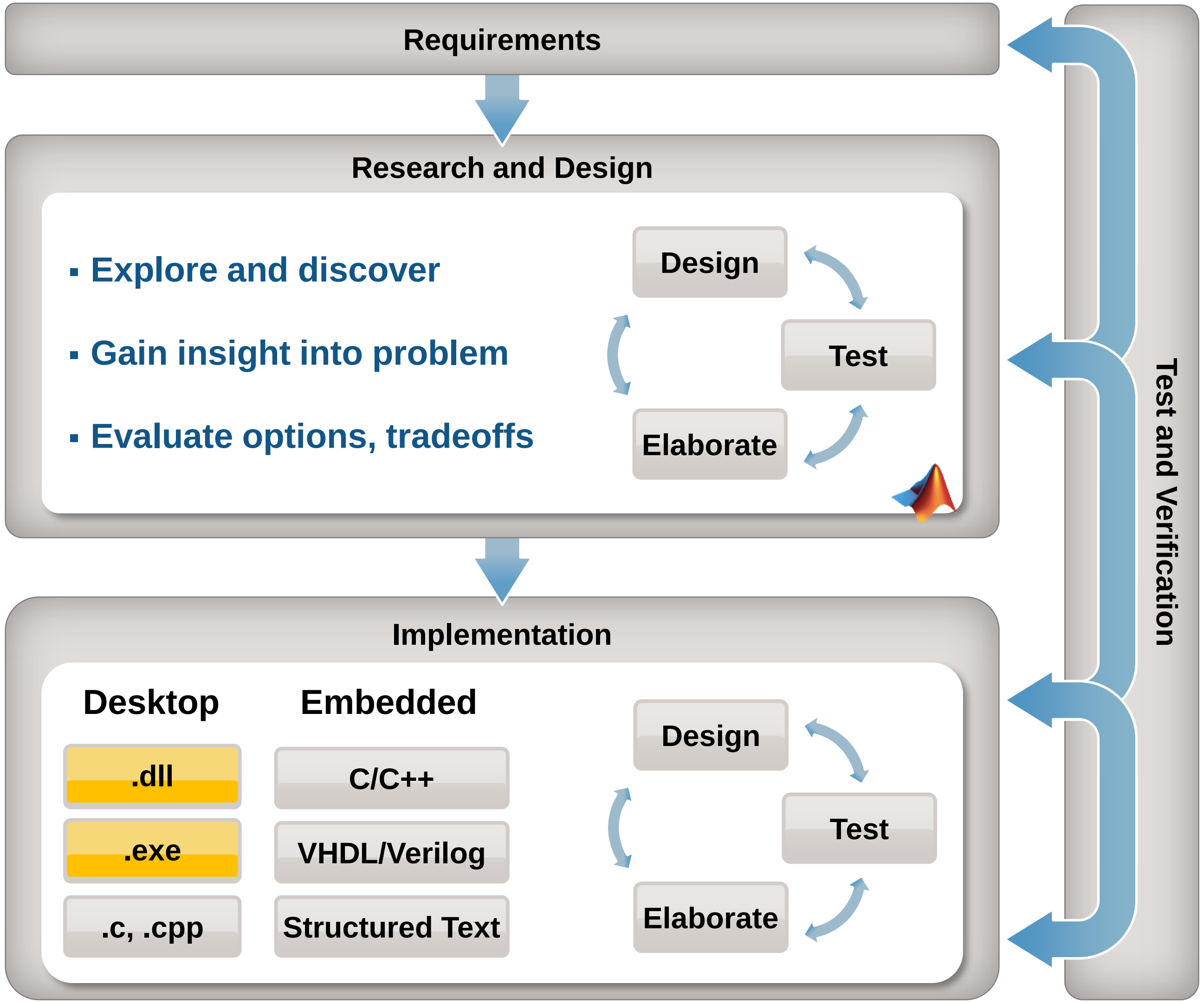
Working with GPU Coder

Generate CUDA for NVIDIA GPUs

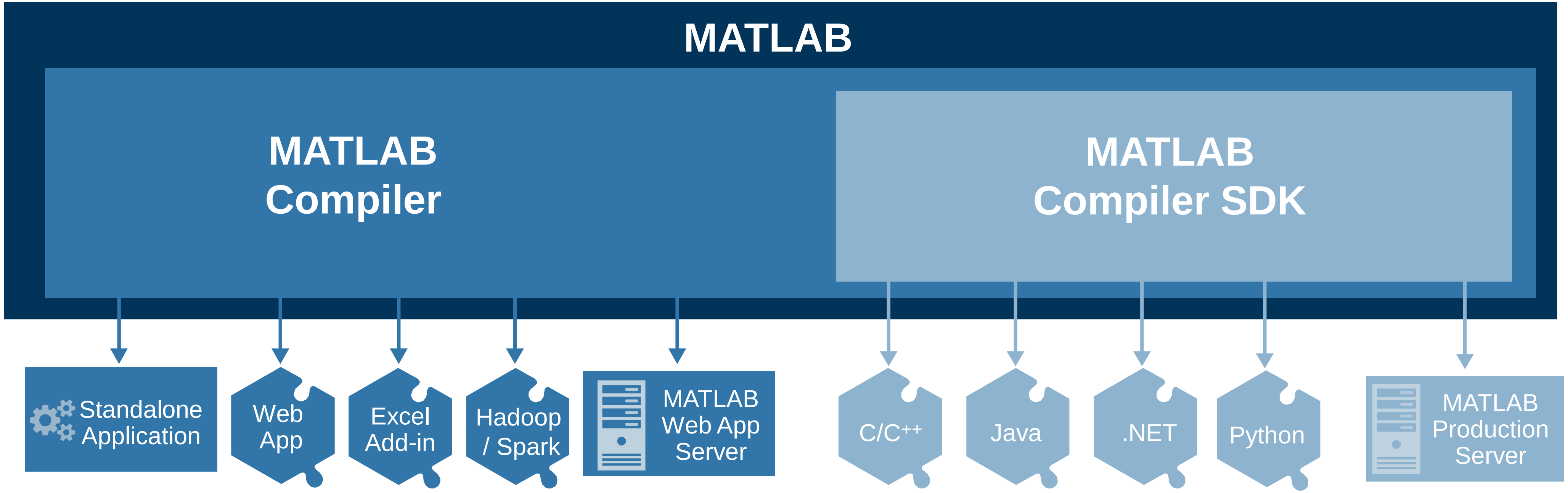
- Deploy deep learning applications, include pre- and post-processing
- Create CUDA kernels from MATLAB algorithms for acceleration on GPUs
- Automated deployment to NVIDIA GPUs, including Jetson/DRIVE



Other Desktop Deployment Options



Other Deployment Options



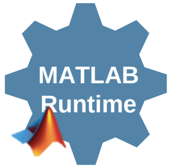
MATLAB Compiler for sharing MATLAB programs without integration programming

MATLAB Web App Server provides feature set to publish MATLAB apps and Simulink simulations created using App Designer as interactive web apps

MATLAB Compiler SDK provides implementation and platform flexibility for software developers

MATLAB Production Server provides the most efficient development path for secure and scalable web and enterprise applications

Choosing the Right Deployment Solution



	MATLAB Coder	MATLAB Compiler MATLAB Compiler SDK
Output	Portable and readable C/C++ source code	Executable or software component/shared library
Main Use Case	Deploy MATLAB code as portable C/C++ code on embedded platforms or desktop	Deploy MATLAB programs as standalone applications on desktop or production servers
MATLAB language support	Subset	Full
Supported toolboxes	Some toolboxes	Most toolboxes
Production	Embedded Coder	MATLAB Production Server
Graphics Support	None	Full
Library Dependency	None	MATLAB Runtime

More Information

- To learn more, visit the product page:
  mathworks.com/products/matlab-coder

- To request a trial license:
  - Talk to your MathWorks account manager to request a trial license and set up a guided evaluation with an application engineer