

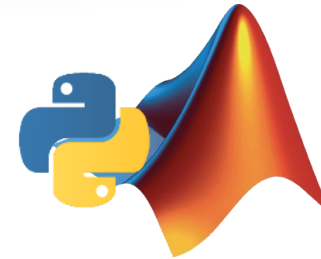
# Using MATLAB with Python



Kantika Wongkasem  
Application Engineer  
Ascendas Systems Co.,Ltd



[Kantika@ascendas-asia.com](mailto:Kantika@ascendas-asia.com)



Date : 27<sup>th</sup> Jan 2023

Time : 09:00 – 11:00

TECHSOURCE



MATLAB®  
& SIMULINK®

As the **Authorized Reseller** in Southeast Asia for MathWorks Inc, developer of the MATLAB® and Simulink® family of products since 1996, we provide organizations and businesses with a wide spectrum of the best tools, products and services to facilitate innovations.



CERT NO.: 2006-2-1559  
ISO 9001 : 2008





## Headquarters

Natick, MA USA

### North America

United States

### Europe

Finland  
France  
Germany  
Ireland  
Italy  
Netherlands  
Spain  
Sweden  
Switzerland  
UK

### Asia-Pacific

Australia  
China  
India  
Japan  
Korea



**4 million+**  
**users**

in more than 180  
countries



**5000+**  
**staff**

in 31 offices around  
the world



**\$1+**  
**billion**

in 2019 revenues with  
60% from outside the US



**Privately**  
**held**

and profitable every year

# DATA QUADRANT AWARDS 2022

## Machine Learning



### MathWorks Matlab

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming.

8.8

COMPOSITE  
SCORE

+98

EMOTIONAL  
FOOTPRINT

92%

LIKELINESS TO  
RECOMMEND

Review Software

Product Report 15+  
pages



**MATLAB® is the enterprise engineering platform for AI.**

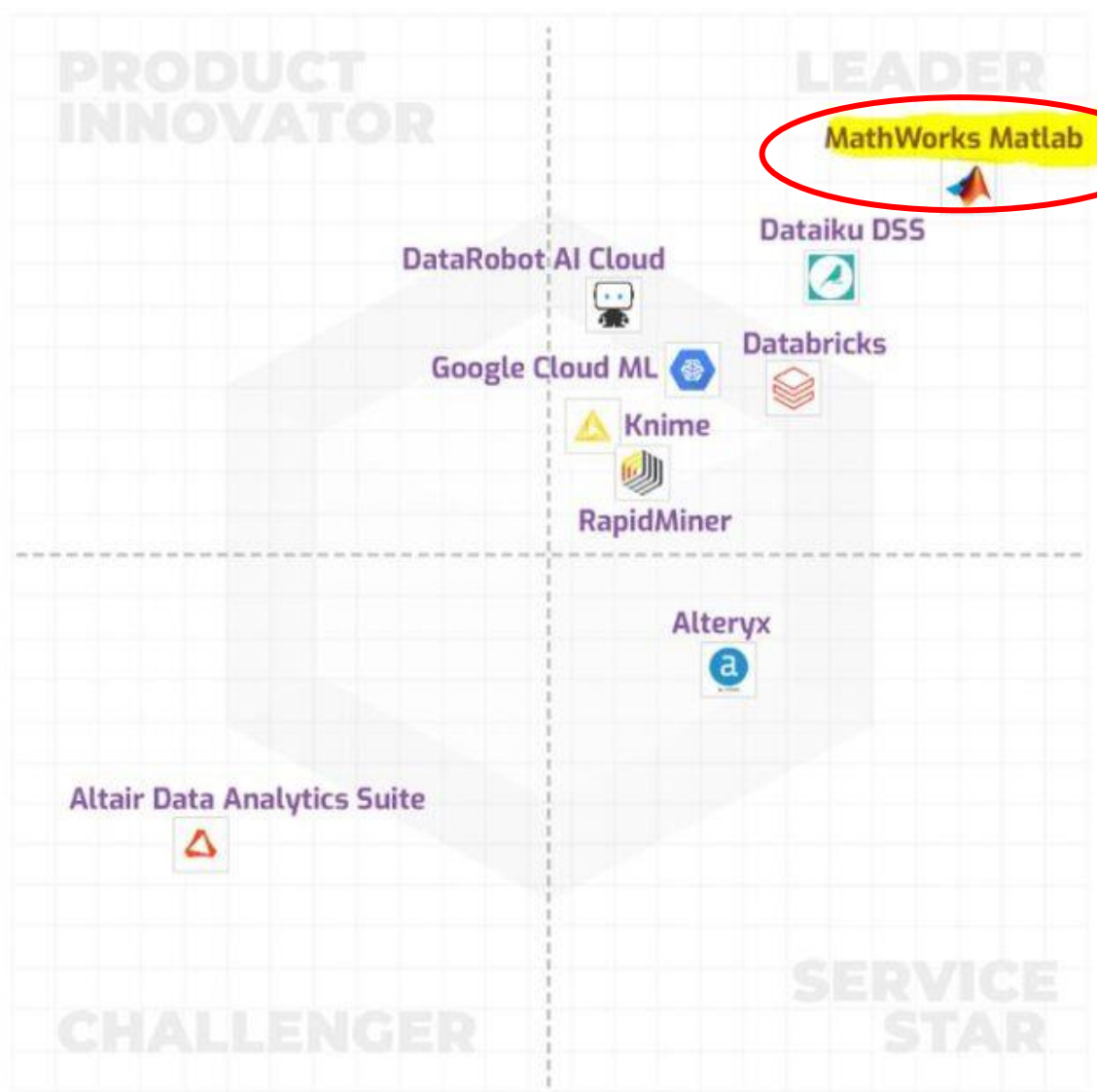
- Empower your team, including those with limited AI or data science experience
- Apply complete workflows for data preparation, AI modeling, system design, and production
- Deploy AI models on embedded devices, edge, enterprise systems, and the cloud
- Tackle integration challenges and reduce risk in designing AI-driven systems with Simulink®

\*[Top Machine Learning Software Awards 2022 | Software Reviews](#)

8.9

PRODUCT FEATURES AND SATISFACTION

7.2



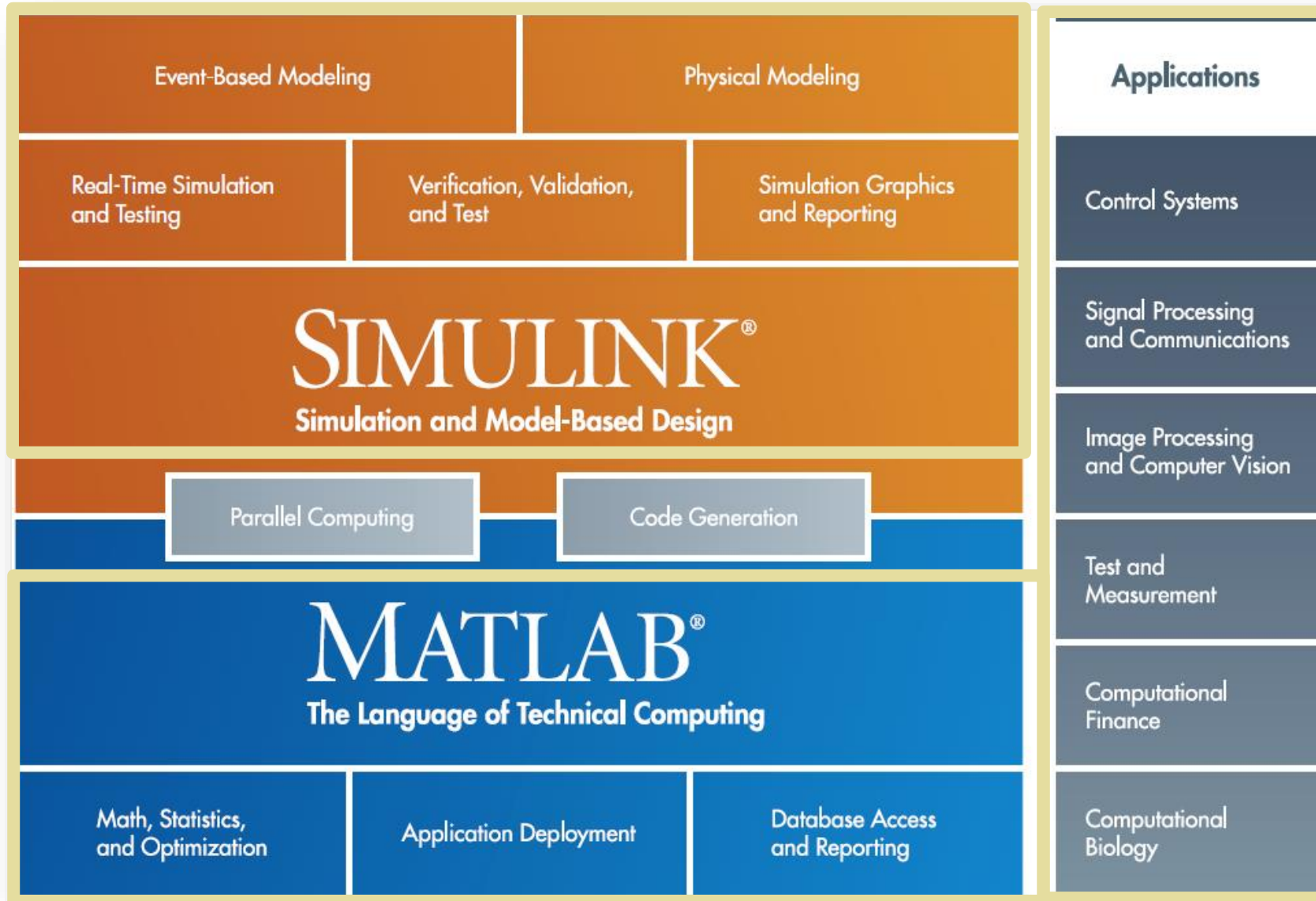
6.2

VENDOR EXPERIENCE AND CAPABILITIES

9.2



# MathWorks Product Overview



MathWorks offers nearly 119 products for technical computing and Model-Based Design. Widely used throughout industry, government and academia, these products are accelerating the pace of discovery, innovation, development, and learning in engineering and science

# Our Solutions



Artificial Intelligence



Wireless Systems



Computational Biology

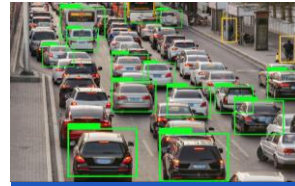


Image Processing & Computer Vision



Predictive Maintenance



Control Systems



Machine Learning



Robotics



Power Electronics Control Design



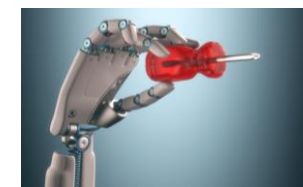
Internet of Things (IoT)



Power System Analysis & Design



Data Science



Mechatronics



Signal Processing



Automated Driving Systems



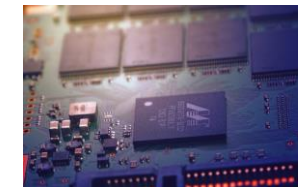
Deep Learning



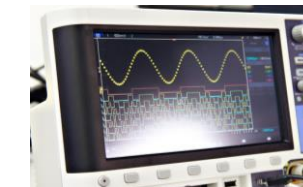
Test & Measurement



Embedded Systems



FPGA, ASIC & SoC Development



Mixed Signal System



Enterprise & IT Systems



## Our Locations

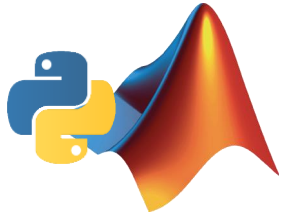


- **Singapore (Headquarter)**
- **Malaysia**
- **Thailand**
- **Philippines**
- **Vietnam**
- **Indonesia**
- **Brunei**

**Contact us at**

[www.techsource-asia.com/contact-us](http://www.techsource-asia.com/contact-us)





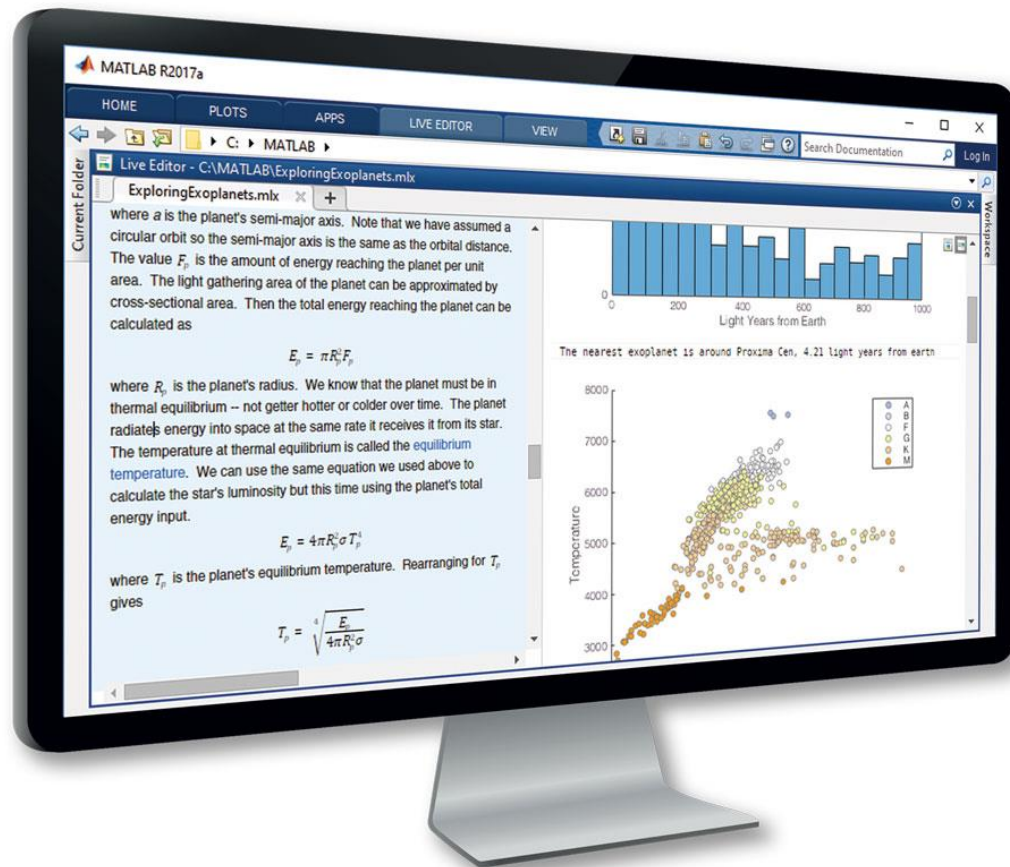
# “Using MATLAB with Python”

## Agenda (2hr)

- |                                   |                       |
|-----------------------------------|-----------------------|
| - Introduction MATLAB and Python  | 09:00 - 09:30 (15min) |
| - Co-execution multiple languages | 09:30 - 09:45 (15min) |
| - Call Python from MATLAB         | 09:45 - 10:00 (15min) |
| - Short Break                     | 10:00 - 10:15 (15min) |
| - Call MATLAB from Python         | 10:15 - 10:35 (30min) |
| - Resources for further learning  | 10:35 - 10:50 (15min) |
| - Q&A                             | 10:50 – 11:00 (10min) |

# What's new in MATLAB

# Teach with MATLAB Live Editor



## MATLAB in an Executable Notebook

Use live scripts to create **engaging lectures** that combine explanatory text, mathematical equations, code, and results

**Share** live scripts directly with colleagues or students

Work in a **single environment** to eliminate context switching



# Use data to tell a story – Live Scripts

```

1 % pulse of input DC current (stimulation parameters)
2 params.stm.pulseWidth = 0.8;
3 params.stm.pulseAmpl = 70;
4 params.mem.C = 140; % membrane capacitance
5 params.mem.vr = -60; % resting membrane potential
6 params.mem.vt = -52; % instantaneous threshold potential
7 params.mem.a = 0.01; % recovery time constant
8 params.mem.b = -0.9; % determines whether u is amplifying (b<0) or resonant (b>0)
9 params.mem.c = -40; % voltage reset value
10 params.mem.d = 170; % total outward-inward current during spike
11 params.mem.vpeak = 17; % spike cutoff
12
13 % initialize/calculate some parameters
14 T = 1000; tau = 1; % total time of simulation and time step
15 n = round(T/tau); % simulation time steps

```

- MATLAB code and related output together
- Text, pictures, graphics and equations inline
- UI controls for users to select options
- Export to html, latex, word, pdf

29 end

- Communicate your research (Lab meetings, collaborators)
- Allow others to interact with your code
- Upload code, documentation & results together

```

43 ylabel('membrane potential')
44

```

## Spiking model of cortical neurons

This live script is an attempt to reproduce the neuronal dynamics of most cortical neurons by using the Izhikevich model of spiking neurons. It is biologically as plausible as the Hodgkin-Huxley model, yet computationally viable like the integrate-and-fire models. For more information, please see the [original paper](#) of Eugene Izhikevich [*IEEE Trans on Neural Networks*, 14(6), 2003] as well as the [project report](#) of Ahmed Salem from the University of Edinburgh, School of Informatics.

### 1 The spiking model

#### 1.1 Spiking model equation of Izhikevich

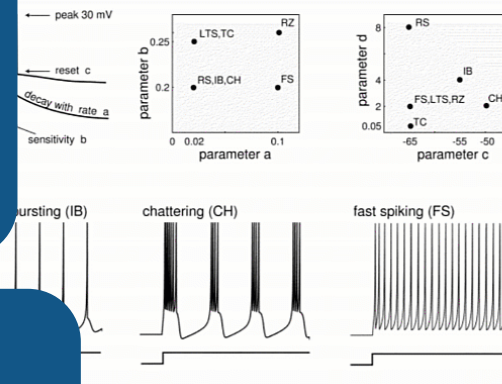
The spiking model of cortical neurons is based on the following equations

$$C\dot{v} = k(v - v_r)(v - v_t) - u + I \dots\dots\dots [1]$$

$$\dot{u} = a\{b(v - v_r) - u\} \dots\dots\dots [2]$$

$$\text{If } v \geq v_{\text{peak}}, \text{ then } v = c, u = u + d \dots\dots\dots [3]$$

where  $v$  is the membrane potential,  $C$  is the membrane capacitance and  $u$  is the membrane recovery variable.  $v_r$  is the resting membrane potential and  $v_t$  is the instantaneous spiking threshold. A large diversity of cortical neurons can be simulated by setting the [membrane parameters](#)  $a$ ,  $b$ ,  $c$  &  $d$  appropriately (see figure below).



#### Parameters of the model

##### Current parameters

that is injected

width = 0.7 ;  
ampl = 70;

##### Parameters

The four parameters of the model  $a$ ,  $b$ ,  $c$  &  $d$  control various membrane properties and affect spiking behavior. They are

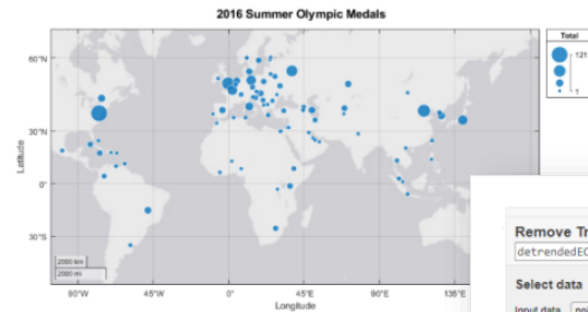
- $a$  = recovery time constant
- $b$  = determines whether recovery variable,  $u$ , is an amplifying ( $b < 0$ ) or a resonant ( $b > 0$ ) variable.
- $c$  = voltage reset value following spiking

The image displays the MATLAB R2020a Live Editor interface. The top ribbon includes tabs for HOME, PLOTS, APPS, LIVE EDITOR (selected), INSERT, and VIEW. The LIVE EDITOR tab contains icons for Code, Control, Task, Section Break, Text, Table of Contents, Code Example, Image, Hyperlink, and Equation. Below the ribbon, the breadcrumb path is C:\Shubo\_files\openScienceWebinar\LiveEditor. The interface is divided into two main panes: the Workspace on the left and the Editor/Live Editor on the right. The Workspace pane has a table with columns 'Name' and 'Value'. The Editor/Live Editor pane shows a file named 'createSineWave\_simple.mlx' with other files like 'sine\_code.m', 'callMATLABfromPython.mlx', and 'analyseOSattendees.mlx' open in the background.

Workspace

Editor/Live Editor

# The Live Script Gallery

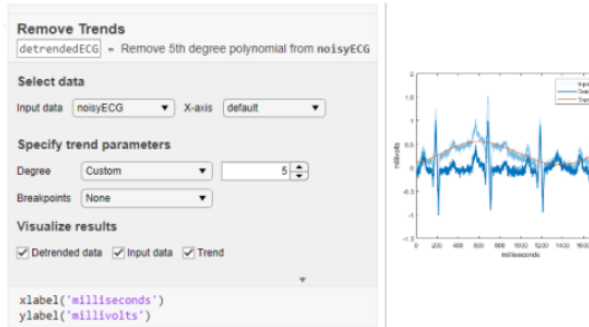


## GDP and Olympic Success

by David Garrison

[Run this example](#)

[View on File Exchange](#)

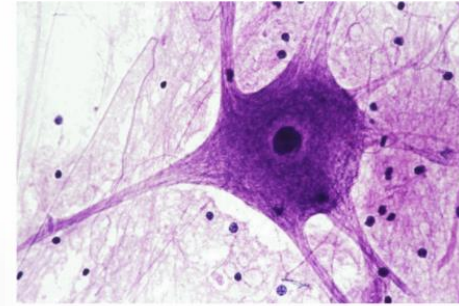


## Calculate Heart Rate from Electrocardiogram Data

by Heather Gorr

[Run this example](#)

[View on File Exchange](#)

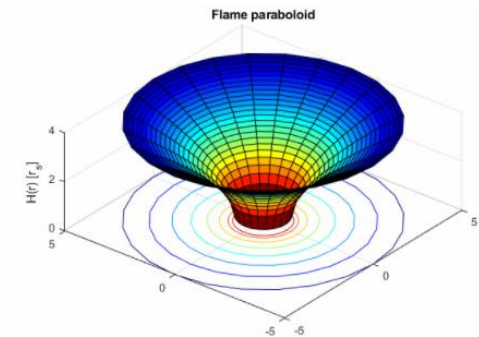


## Inside the Brain: Modelling the Neuron

by Davide Borra

[Run this example](#)

[View on File Exchange](#)



## Schwarzschild metric and black holes (Computational example)

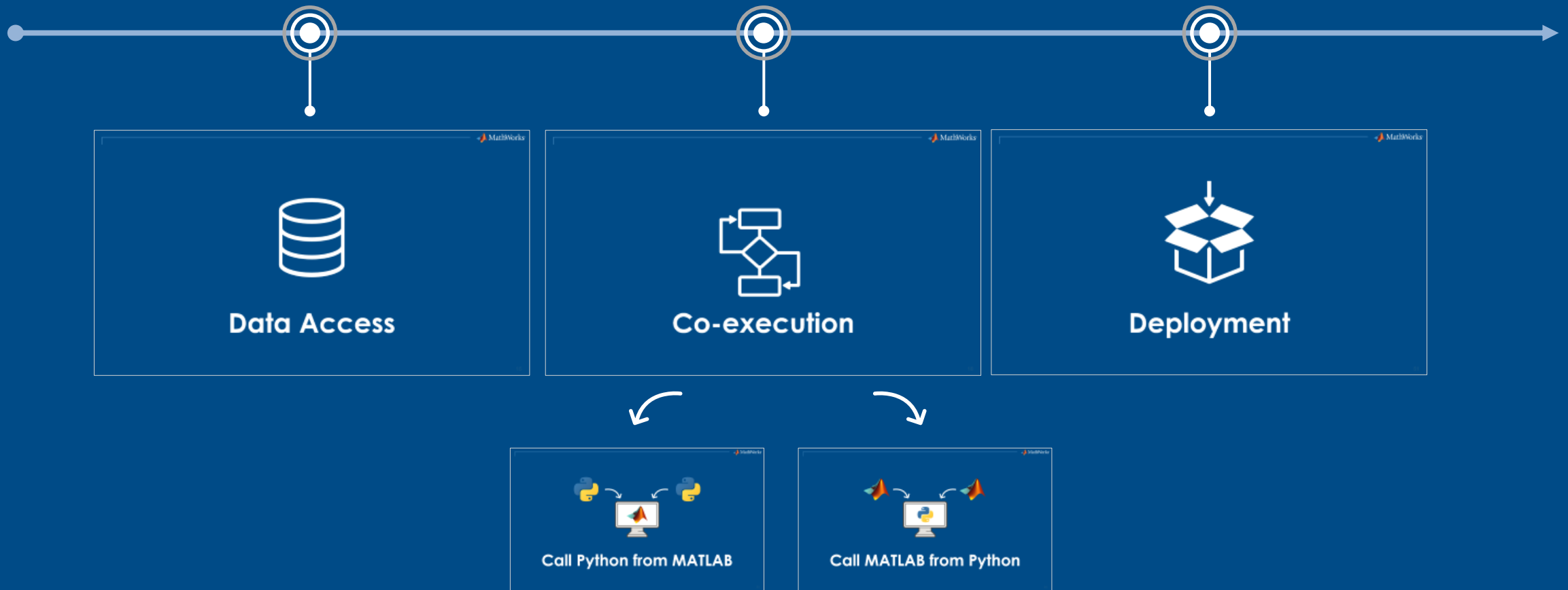
by Kurt Braeuer

[Run this example](#)

[View on File Exchange](#)



# Plan





# Data Access

# Setup tips



- Use **pyenv** to update settings (**pyversion** before R2019b)
- Ensure all code is on path
- Check environment settings, depending on how you set up python



- How to select the MATLAB engine version if I have multiple MATLAB releases to interface?

## Setup

Ensure Python is installed and confirm the version with `pyenv`.

We will provide the path to the correct executable for MATLAB Online.

```
pyenv
```

```
ans =
```

```
PythonEnvironment with properties:
```

```
Version: "3.9"  
Executable: "C:\Users\KantikaWongkasem\anaconda3\python.EXE"  
Library: "C:\Users\KantikaWongkasem\anaconda3\python39.dll"  
Home: "C:\Users\KantikaWongkasem\anaconda3"  
Status: Loaded  
ExecutionMode: OutOfProcess  
ProcessID: "4668"  
ProcessName: "MATLABPyHost"
```

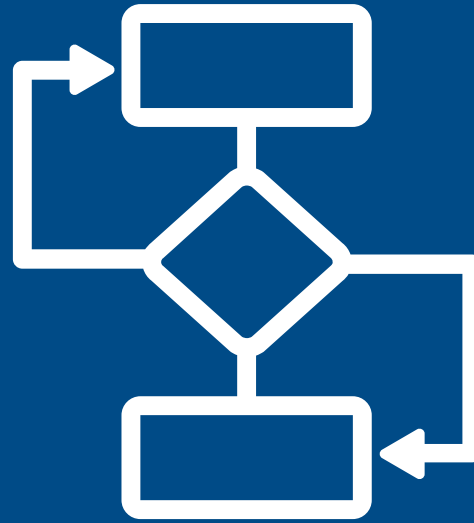
## Setup

## Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

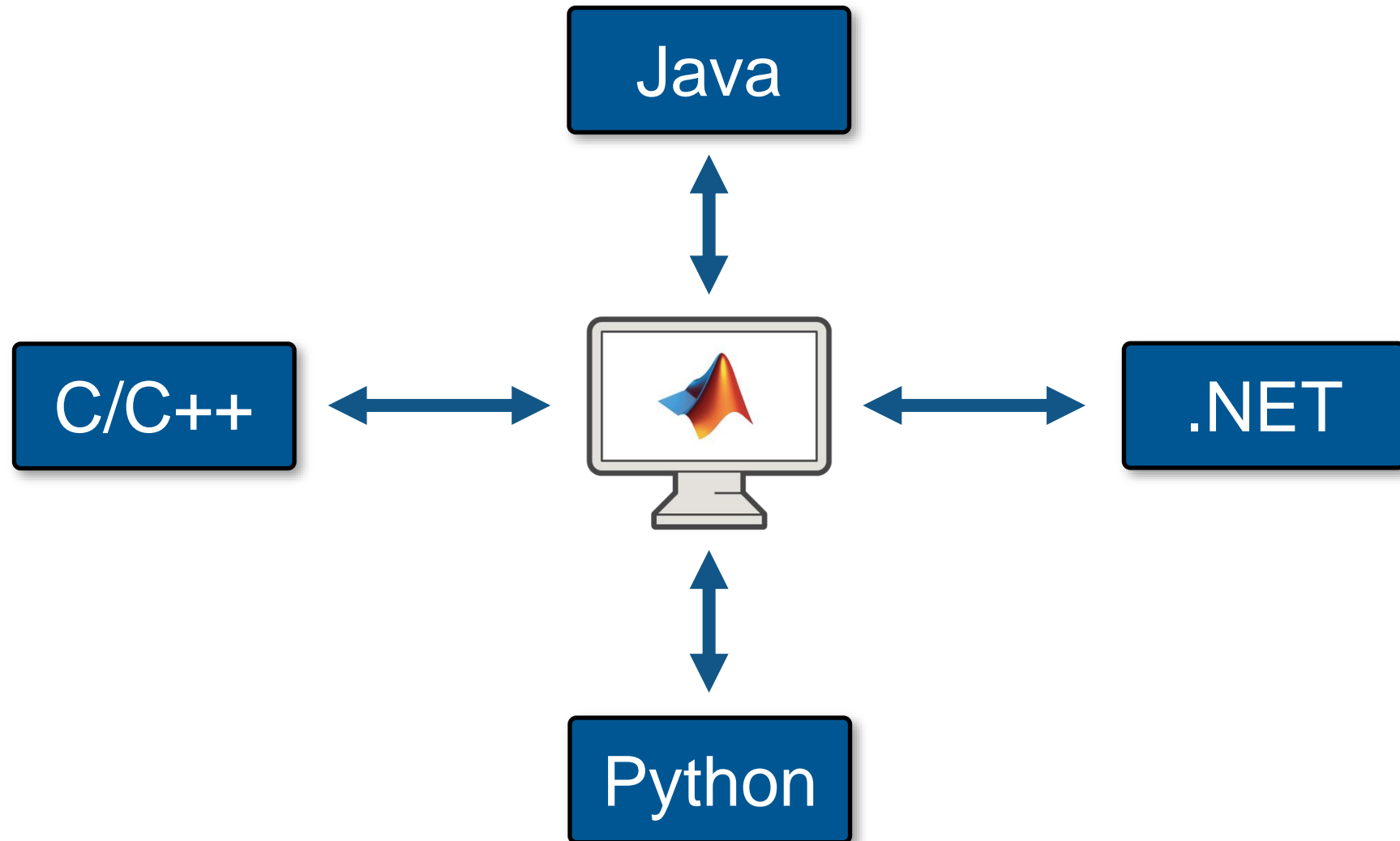
## Deployment





# Co-execution

MATLAB provides flexible integration with multiple languages



Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

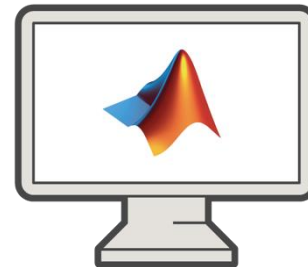
Deployment

# Given: Existing Python Code accessing & preparing weather data

Data preparation

Modeling

Deployment

**Weather Data**Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

# Call Python from MATLAB

Data preparation

Modeling

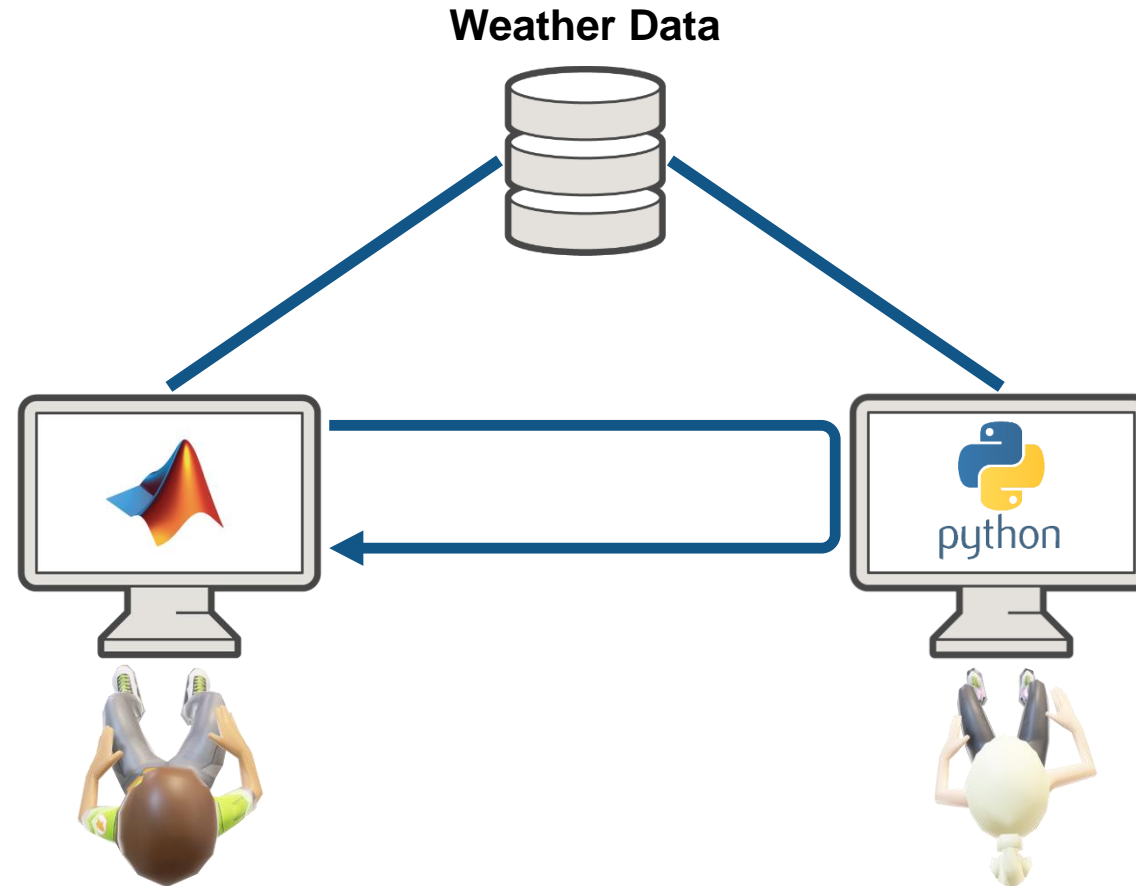
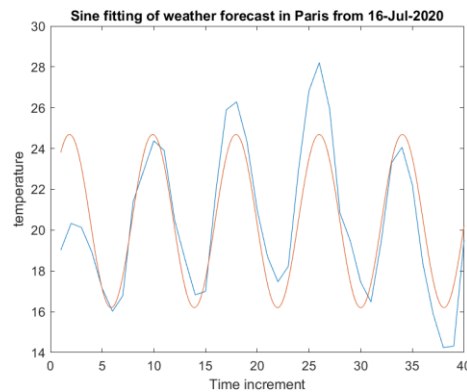
Deployment

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment





# Call MATLAB from Python

Data preparation

Modeling

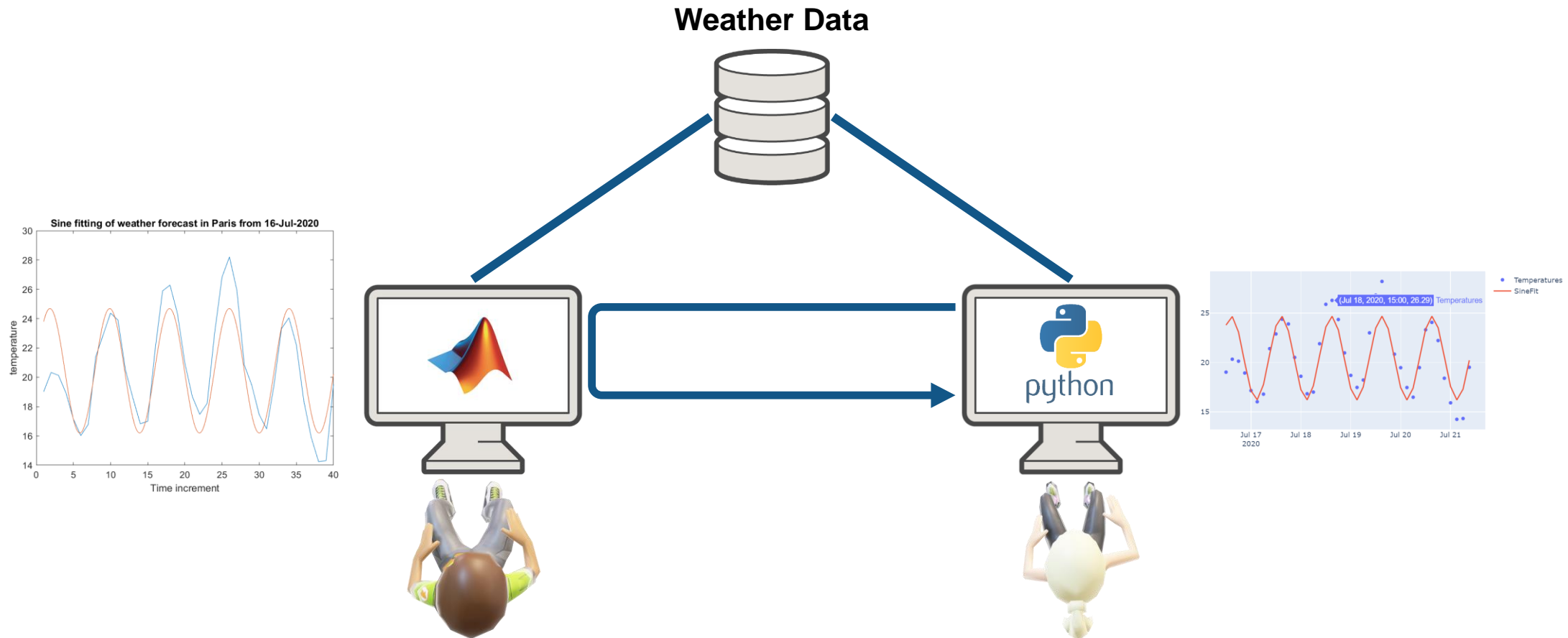
Deployment

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



# Deploy: MATLAB Analytics into Python

Data preparation

Modeling

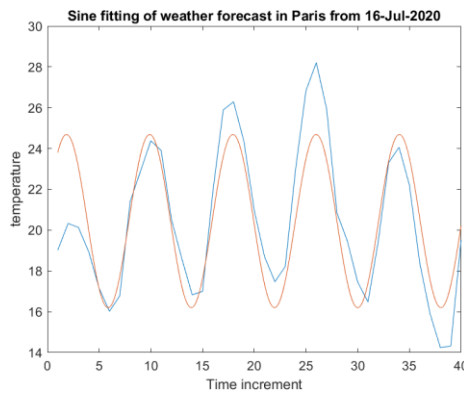
Deployment

Data  
Access

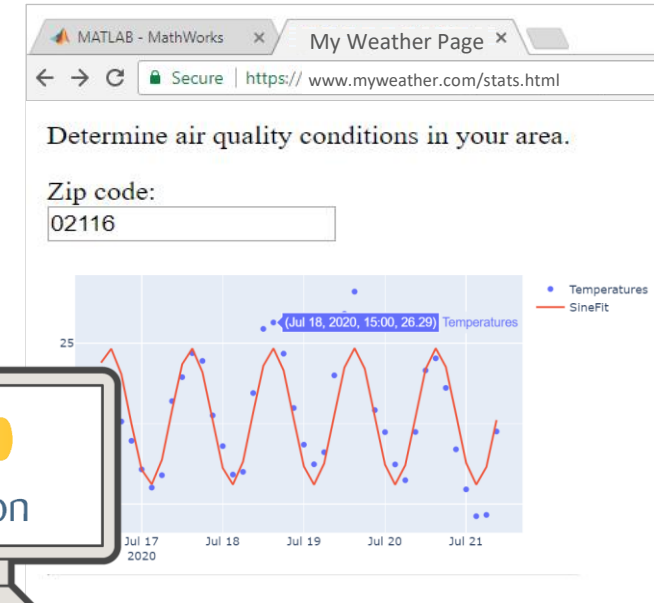
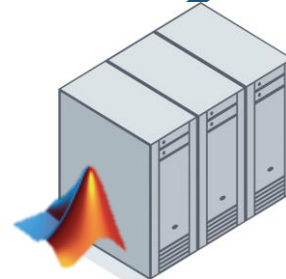
Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



Weather Data



# Deploy: MATLAB Analytics into Python

Data preparation

Modeling

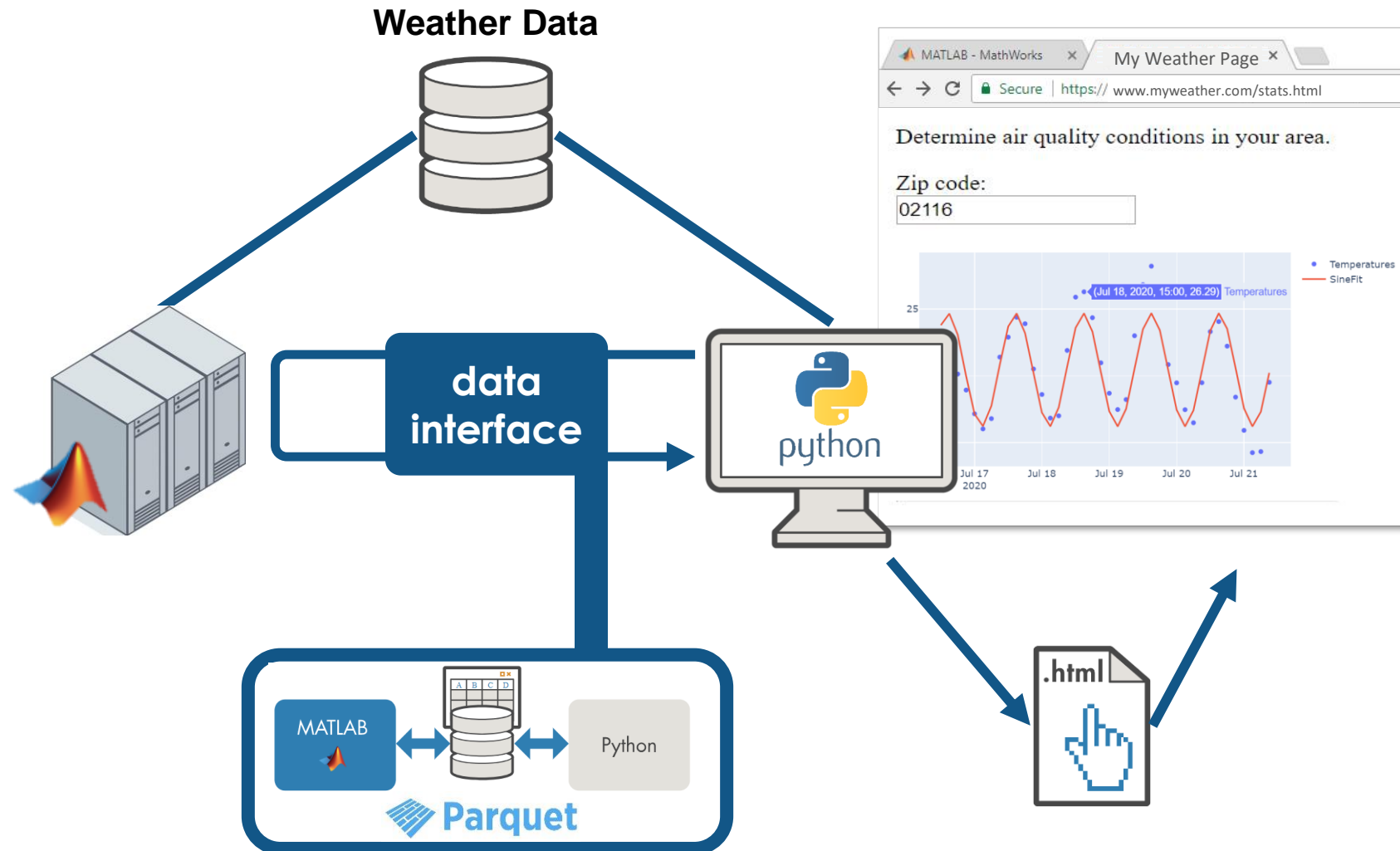
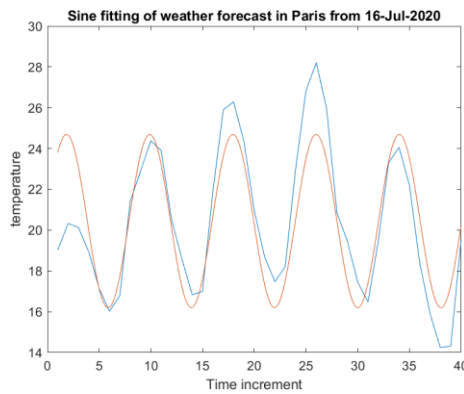
Deployment

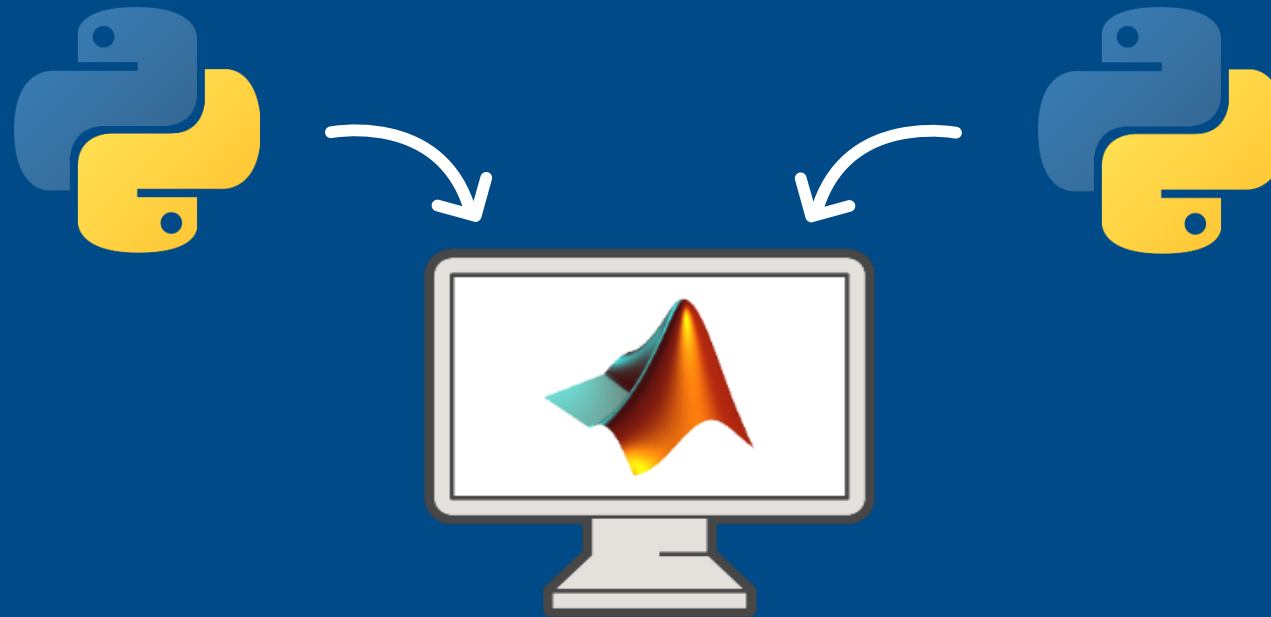
Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment





# Call Python from MATLAB

# Create List, Tuple, and Dictionary Types

- This table shows the commands for creating **list**, **tuple**, **dict** types. The commands on the left are run from the Python interpreter. The commands on the right are MATLAB commands*

Python list — []	MATLAB py.list
>>> ['Kantika', 'Wongkasem', 'Fah']	>> py.list({'Kantika','Wongkasem','Joseph'})
>>> [[1,2],[3,4]]	>> py.list({py.list([1,2]),py.list([3,4])})
Python tuple — ()	MATLAB py.tuple
>>> ('Kantika', 27, 'Engineer')	>> py.tuple({'Kantika',19,'Engineer'})
Python dict — {}	MATLAB py.dict
>>> {'Ink': 424 , 'Pink': 543 , 'Blue': 320}	>>py.dict(pyargs('Ink',424,'Pink',543,'Blue',320))
	For information about passing keyword arguments, see <a href="#">pyargs</a>

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



# Access Python Standard Library Modules in MATLAB

## Data Access

## Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

## Deployment

- Create a Python `list` datatype:
- MATLAB recognizes Python objects and automatically converts the MATLAB cell array to the appropriate Python type.
- To convert the list variable to a MATLAB variable, call `cell` on the list and `string` on the elements of the list

```
l = py.list({'Kantika','Wongka','Fah'})

l =
Python list with values:

['Kantika', 'Wongka', 'Fah']

Use string, double or cell function to convert to a MATLAB array.
```

```
l.append('Engineer')
l
```

```
l =
Python list with values:

['Kantika', 'Wongka', 'Fah', 'Engineer']

Use string, double or cell function to convert to a MATLAB array.
```

```
myl = cellfun(@string,cell(l))
```

```
myl = 1x4 string
    "Kantika"    "Wongka"    "Fah"    "Engineer"
```

```
cell(l)
```

```
ans = 1x4 cell
```

	1	2	3	4
1	1x7 str	1x6 str	1x3 str	1x8 str

# Call User-Defined Python Module(1)

*This example shows how to call methods from following Python module:*

```

Editor - D:\DriveD\Kantika W\Edu\Python Webinar\MATLABwithPythonWorkshopExpo\mymod.py
+6 CallPythonFromMATLAB_pre2022a.mlx imp.m py2mat.m mat2py.m ExploringHyperspectralDataInHypers
1 # mymod.py
2 """Python module demonstrates passing MATLAB types to Python functions"""
3 def search(words):
4     """Return list of words containing 'ka'"""
5     newlist = [w for w in words if 'ka' in w]
6     return newlist
7
8 def engineer(words):
9     """Append 'Engineer' to list of words"""
10    words.append('Engineer')
11    return words
  
```

Command Window

New to MATLAB? See resources for [Getting Started](#).

```

>> l = py.list({'Kantika','Kittika','Fah'})

l =

Python list with values:

    ['Kantika', 'Kittika', 'Fah']

Use string, double or cell function to convert to a MATLAB array.

>> py.mymod.search(l)

ans =

Python list with values:

    ['Kantika', 'Kittika']

Use string, double or cell function to convert to a MATLAB array.
  
```

- ✓ How to use call mymod.py and To convert the list variable to a MATLAB variable

Command Window

New to MATLAB? See resources for [Getting Started](#).

```

>> py.mymod.engineer(l)

ans =

Python list with values:

    ['Kantika', 'Kittika', 'Fah', 'Engineer']

Use string, double or cell function to convert to a MATLAB array.

>> cellfun(@string,cell(l))

ans =

    1x4 string array

    'Kantika'    'Kittika'    'Fah'    'Engineer'

>> cell(l)

ans =

    1x4 cell array

    {1x7 py.str}    {1x7 py.str}    {1x3 py.str}    {1x8 py.str}
  
```

# Display Python Documentation in MATLAB

*You can display help text for Python functions in MATLAB. For example*

```
>> py.help('list.append')  
Help on method_descriptor in list:  
  
list.append = append(self, object, /)  
    Append object to the end of the list.
```

- Tab completion when typing `py`. Does not display available Python functionality. For more information,
- see [Limitations to Python Support - MATLAB & Simulink \(mathworks.com\)](https://www.mathworks.com/help/matlab/matlab_external/limitations-to-python-support.html)

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

# Call User-Defined Python Module(2)

*If you create `mymod.py` in a Python editor, be sure that the module is on the Python search path*

- From the MATLAB command prompt, add the current folder to the Python search path: `>> py.sys.path()`

```
if count(py.sys.path,pwd()) == 0
    insert(py.sys.path,int32(0),pwd());
end|
```

- To learn how to call the function, read the function signature for the search function in the `mymod.py` source file. The function takes one input argument, `words`.

```
def search(words):
```

# Reload Modified User-Defined Python Module

## Data Access

## Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

## Deployment

- **Create Python Module**

Save the file `mymod.py`:

```
def myfunc():  
    return 'version 1'
```

- **Modify Module**

Modify the function:

```
    return 'version 2'
```

- **Unload Module**

```
>> clear classes
```

- **Import & Reload Modified Module**

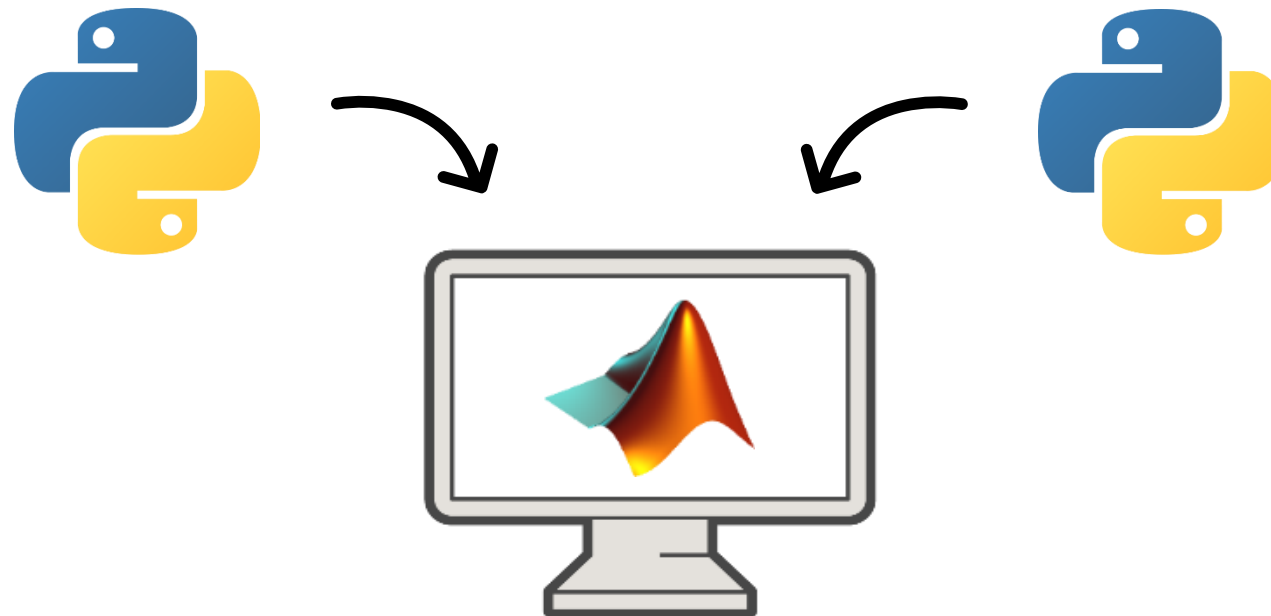
```
>> mod = py.importlib.import_module('mymod');  
>> py.importlib.reload(mod);
```



# Why Call Python from MATLAB?

Already working in MATLAB, and:

- Want to reuse existing Python code
- Need functionality that is only available in Python



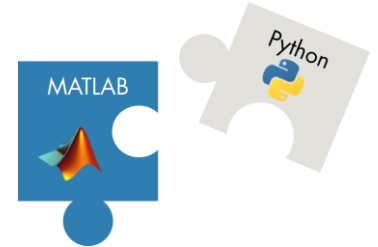
Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

# Calling Python libraries from MATLAB



Use the weather.py module to get the air quality for Paris. This is a user-defined Python module which includes functions to read and parse the current and forecasted weather data by location.

```
jsonData = py.weather.get_current_weather("Paris", "France", apikey.Key)
```

jsonData =  
Python **dict** with no properties.

```
{'coord': {'lon': 2.35, 'lat': 48.85}, 'weather': [{'id': 803, 'main': 'Cloudy'}
```

Parse the json data returned from the weather API.

The Python dictionary can be represented as a MATLAB struct.

```
weatherData = py.weather.parse_json(jsondata);  
struct(weatherData)
```

```
ans = struct with fields:  
    temp: 18.7100  
 feels_like: 17.3000  
    temp_min: 17.7800  
    temp_max: [1x1 py.int]
```

Use a function (prepData.m) to prepare data for machine learning (create a table with the expected variable names, preprocessing steps, etc).

```
currentData = prepData(weatherData)
```

currentData = 1x12 table

	DateLocal	city	StateName	T	P	DP	RH	WindDir	WindSpd		
1	01-Jul-2020 11:...	"Paris"	Ile de France	21.6200	20.2600	349.2200	1010	5.1000	73		

```
def get_current_weather(city, country, apikey):  
    # get current conditions in specified location  
    # get_current_weather('boston', 'us', key)  
    import urllib.request  
    import json  
    # read current conditions  
    try:  
        url = "https://api.openweathermap.org/data/2.5/weather?q="+city+", "+country+"&appid="+apikey  
        response = urllib.request.urlopen(url)  
        html = response.read()  
        json_data = json.loads(html)  
  
    except urllib.error.URLError:  
        # if weather API doesn't work, read the file  
        json_data = read_backup(city)  
  
    return json_data
```

Data  
Access

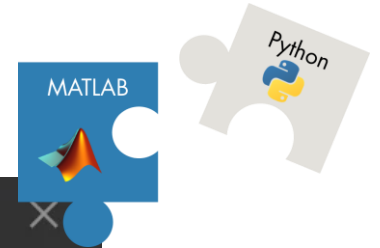
Co-Execution

- Call Python  
from MATLAB

- Call MATLAB  
from Python

Deployment

# Calling Python libraries from MATLAB



Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

A screenshot of the Visual Studio Code editor interface. The top menu bar includes File, Edit, Selection, View, Go, Run, and Terminal. The title bar shows 'weather.py - Visual Studio ...'. The editor window displays a Python script named 'weather.py' with the following code:

```
9
10
11 # weather.py
12 import csv
13 import datetime
14 import json
15 import urllib.request
16
17 BASE_URL = 'https://api.openweathermap.org/data/2.5/{?q={},{}&uni
18 FORECAST_KEYS = {'current_time': 'DateLocal', 'temp': 'T', 'deg': 'Wi
19 | | | | 'speed': 'WindSpd', 'humidity': 'RH', 'pressure': 'P
20
21 def read_backup(city):
22     '''Read example data from a backup file'''
23
24     with open('backupdata.csv', newline='') as csvfile:
25         reader = csv.DictReader(csvfile)
26         for s in [*reader]:
```

The status bar at the bottom indicates 'Ln 11, Col 13 (10 selected)', 'Spaces: 4', 'UTF-8', 'LF', 'Python', and icons for search, settings, and notifications.

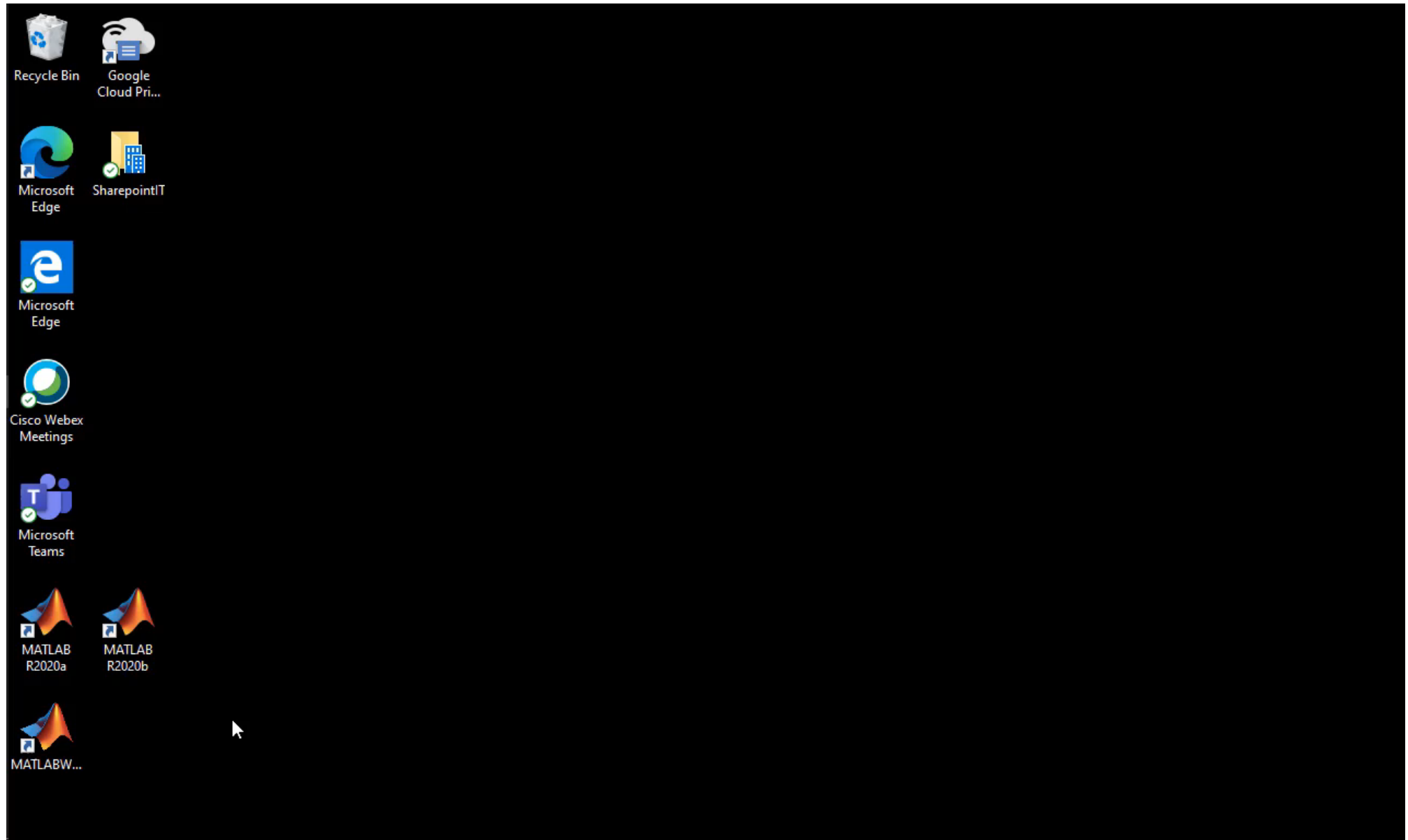
# Connect to Python

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment





MATLAB R2020b

HOME PLOTS APPS LIVE EDITOR INSERT VIEW

Code Control Task Section Break Text Table of Contents Code Example Image Hyperlink Equation

CODE SECTION TEXT IMAGE LINK EQUATION

Current Folder Workspace

Live Editor - C:\\_work\MATLABwithPython\weatherPrediction\1\_CallPythonFromMATLAB\CallPythonFromMATLAB.mlx \*

CallPythonFromMATLAB.mlx \* predictAirQual.m prepData.m BuildAirQualityModel.mlx +

Executable: "C:\Users\fperino\AppData\Local\Programs\Python\Python36\python.exe"  
Library: "C:\Users\fperino\AppData\Local\Programs\Python\Python36\python36.dll"  
Home: "C:\Users\fperino\AppData\Local\Programs\Python\Python36"  
Status: Loaded  
ExecutionMode: InProcess  
ProcessID: "256"  
ProcessName: "MATLAB"

Test a simple function.

2 py.math.sqrt(12)

ans = 3.4641

Details

Command Window

UTF-8 script

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

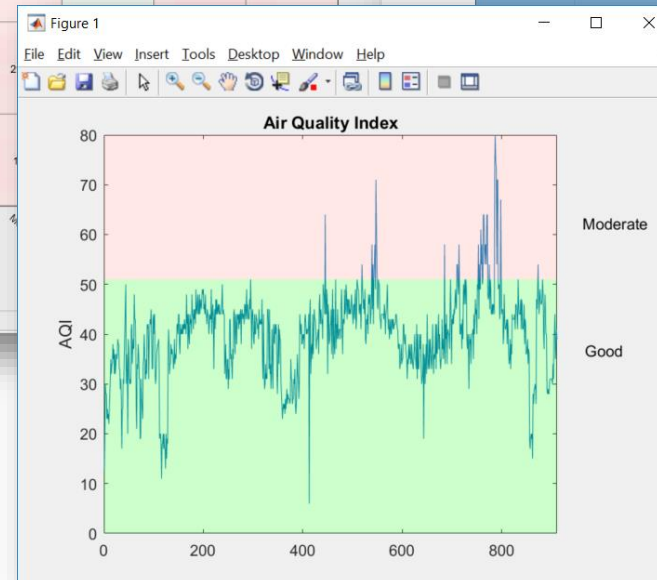
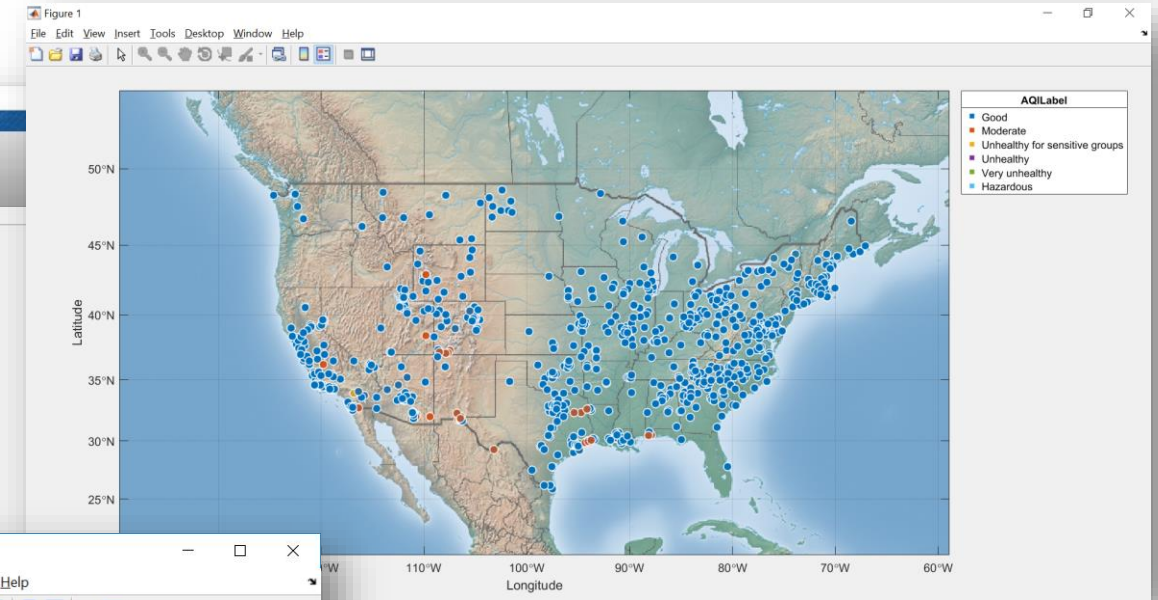
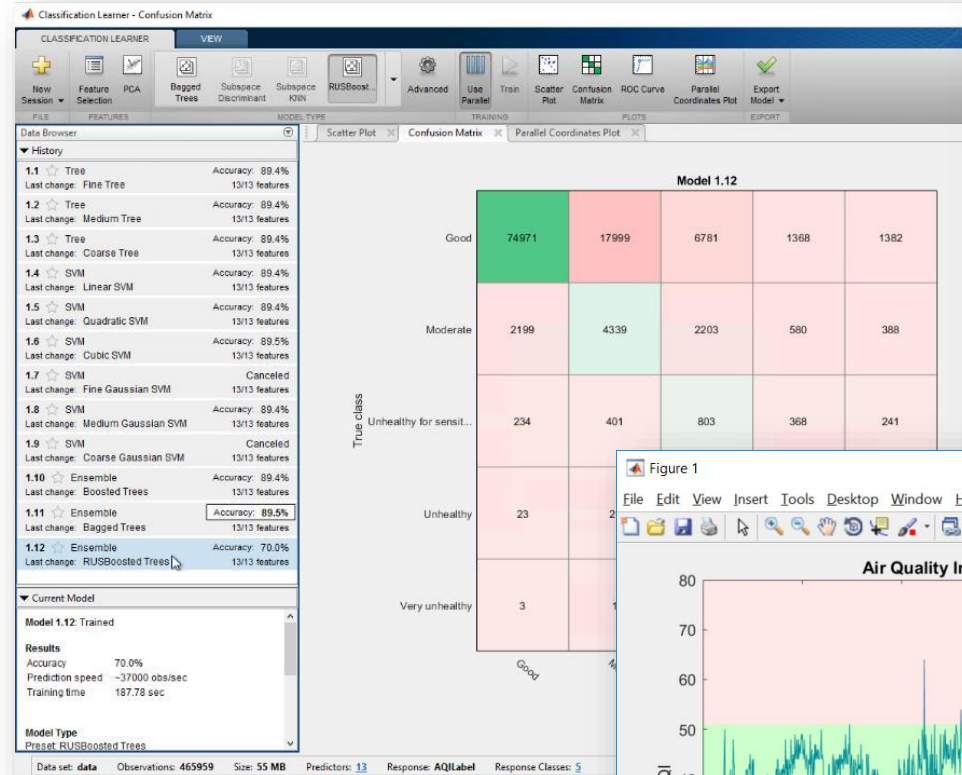
# Air Quality Prediction Model

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment





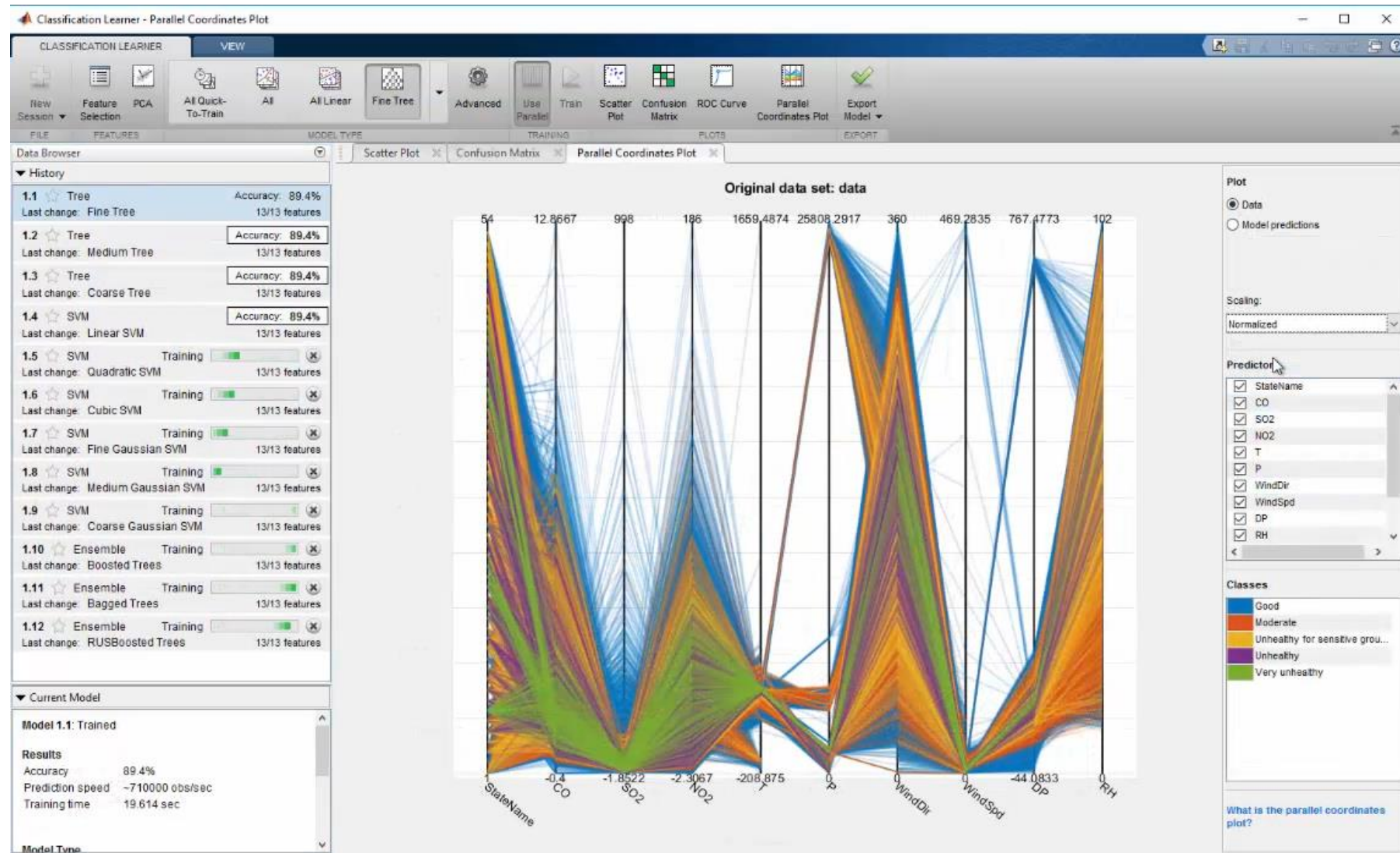
# Train the Air Quality Prediction Model

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



Model development is illustrated in Heather's webinar [MATLAB with Data Science](#)

# Call the Air Quality Prediction Model

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

The screenshot shows the MATLAB R2020b interface. The Live Editor displays a script named `predictAirQual.m` within a project folder `1_CallPythonFromMATLAB`. The script includes comments and code to load a pre-trained model and use a function to prepare data for machine learning.

Use the model to predict the air quality for the new weather data.

'*airQualModel.mat*' is a pre-trained Bagged Classification Tree/ "Random Forest" classification network. The model was saved as a \*.mat file for use in predicting air quality in this demonstration.

```
load airQualModel
```

Use a function (`prepData.m`) to convert and prepare data for machine learning (create a table with the expected variable names, preprocessing steps, etc).

```
apiKey      = readtable("accessKey.txt","TextType","string");
List = "Houston" ;
jsonData    = py.weather.get_current_weather(List,"US",apiKey.Key);
weatherData = py.weather.parse_current_json(jsonData);

currentData = prepData(weatherData);
airQual     = predict(model,currentData)
```

The interface also shows a file explorer on the left with files like `__pycache__`, `accessKey.txt`, `airQualModel.mat`, `airQualModelOld.mat`, `CallPythonFromMATLAB.mlx`, `cities.mat`, `predictAirQual.m`, `prepData.m`, and `weather.py`.

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

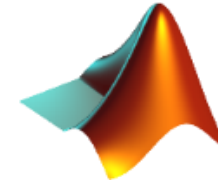
# Recap: Calling Python from MATLAB



# Syntax differences when calling Python from MATLAB



Python



MATLAB

```
>>> import math  
>>> math.sqrt(42)
```



```
>> py.math.sqrt(42)
```

```
>>> print('hello', 'world', sep=', ')
```



```
>> py.print('hello', 'world', ...  
           pyargs('sep', ', '))
```

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

# Data are automatically converted where possible

## *Otherwise convert explicitly*

### Data Access

### Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

### Deployment

#### MATLAB to Python Data Type Mapping

R2021a

When calling a Python® function, MATLAB® converts MATLAB data into types that best represent the data to the Python language.

##### Pass Scalar Values to Python

MATLAB Input Argument Type — Scalar Values Only	Resulting Python py. Type	Examples
double single	float	Use Python Numeric Variables in MATLAB
Complex single Complex double	complex	<pre>z = complex(1,2); py.cmath.polar(z)</pre> <p>ans = Python tuple with no properties.</p> <p>(2.23606797749979, 1.1071487177940904)</p>
int8 uint8 int16 uint16 int32	int	
uint32 int64 uint64	int long (version 2.7 only)	
NaN	float("nan")	
Inf	float("inf")	
string scalar	str	Use Python str Variables in MATLAB

[https://mathworks.com/help/matlab/matlab\\_external/passing-data-to-python.html](https://mathworks.com/help/matlab/matlab_external/passing-data-to-python.html)

# Learn more about

## Data Science with MATLAB

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

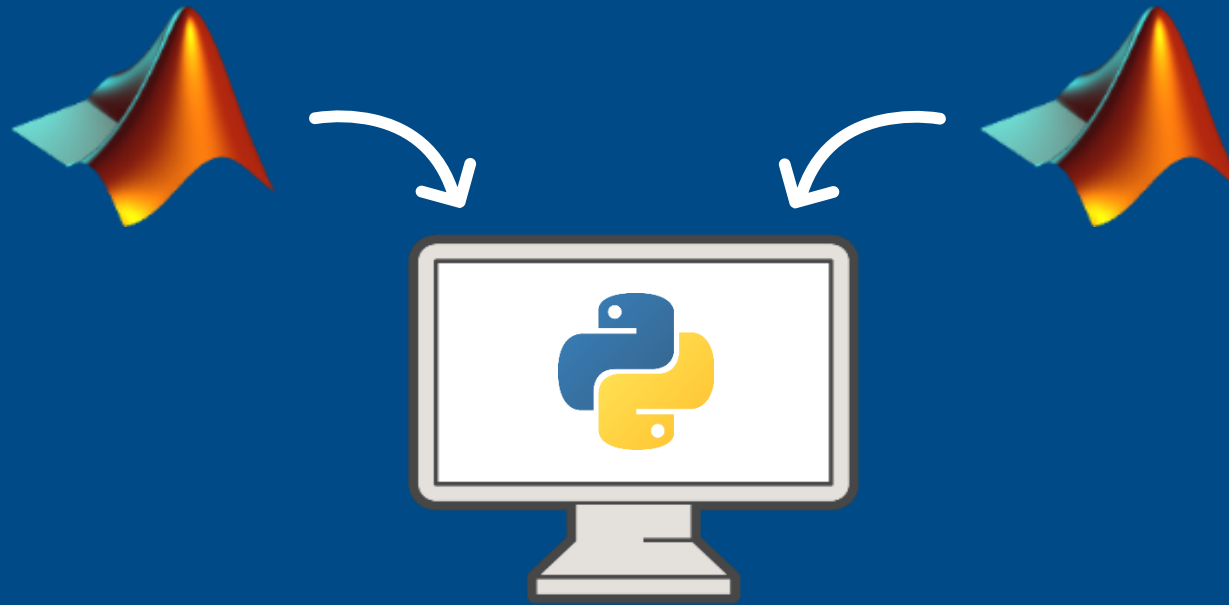
## Data Science With MATLAB

Heather Gorr, PhD  
Senior Product Manager, MATLAB  
MathWorks

Instagram: @heather.codes

Twitter: @HeatherGorr



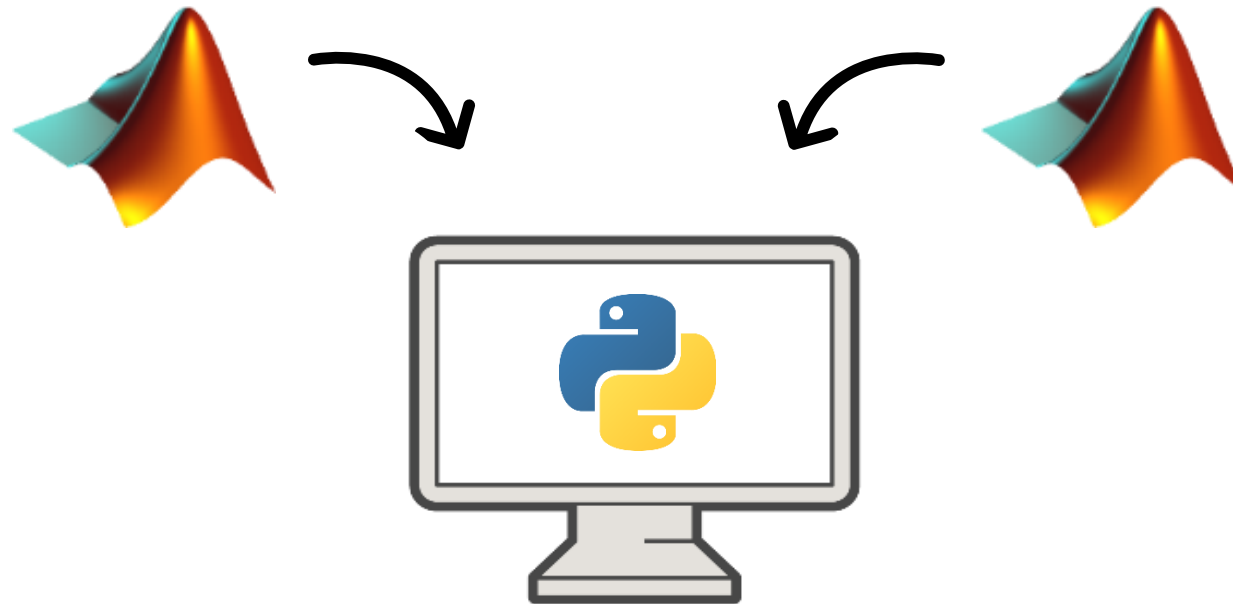


# Call MATLAB from Python

# Why call MATLAB from Python?

Already working in Python, and:

- Want to reuse existing MATLAB code
- Need functionality available in MATLAB
- Want to collaborate with MATLAB users

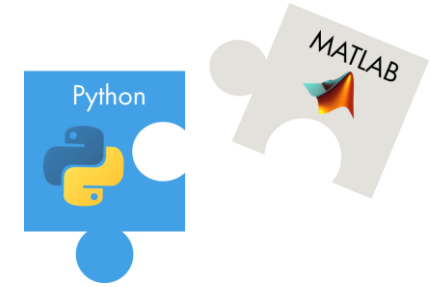


Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



# Call MATLAB from Python

## MATLAB Engine API

- Install MATLAB Engine API for Python

```
$ cd "C:\Program Files\MATLAB\R2021b\extern\engines\python"  
$ python setup.py install
```

- Start a MATLAB process

```
>>> import matlab.engine  
>>> m = matlab.engine.start_matlab()
```

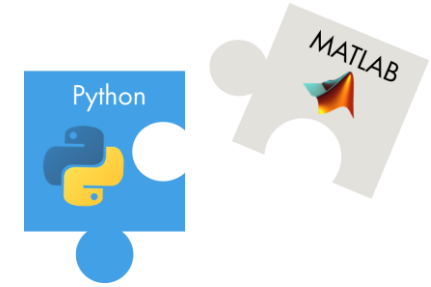
- Call MATLAB functions

```
>>> x = m.sqrt(float(42))
```



# Integrate the MATLAB model in a Python environment

[CallMATLABfromPython.ipynb](#)

A screenshot of a Jupyter Notebook interface. The left sidebar shows a file explorer with a tree view of files and folders. The main area displays a code cell with the title 'Predict Air Quality: Calling MATLAB from Python Using MATLAB Engine API'. The code cell contains text about importing weather data and using the MATLAB code to predict air quality, followed by a link to more information.

**Predict Air Quality: Calling MATLAB from Python Using MATLAB Engine API**

Import the weather data and use the MATLAB code to predict air quality.

This example uses the MATLAB Engine API, which runs a MATLAB session. More info and set up instructions are included here:  
<https://www.mathworks.com/help/matlab/matlab-engine-for-python.html>

Python 3.7.3 64-bit (conda) 0 0

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

Data  
Access

Co-Execution

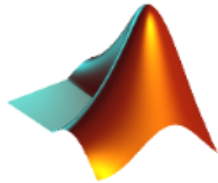
- Call Python  
from MATLAB
- **Call MATLAB  
from Python**

Deployment

# Recap: Calling MATLAB from Python

# Syntax differences when calling MATLAB from Python

Data  
Access



MATLAB



Python

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

```
>> [s,sidx] = sort(x)
```

```
>> foo(x)
```



```
>>> s = eng.sort(x,nargout=2)
```

```
>>> eng.foo(x,nargout=0)
```

Deployment

```
>> C = A + B
```



```
>>> C = eng.plus(A,B)
```

# Data are automatically converted where possible

## Data Access

## Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

## Deployment

### Pass Data to MATLAB from Python

R2021a

#### Python Type to MATLAB Scalar Type Mapping

When you pass Python® data as input arguments to MATLAB® functions, the MATLAB Engine for Python converts the data into equivalent MATLAB data types.

Python Input Argument Type — Scalar Values Only	Resulting MATLAB Data Type
float	double
complex	Complex double
int	int64
long (Python 2.7 only)	int64
float(nan)	NaN
float(inf)	Inf
bool	logical
str	char
unicode (Python 2.7 only)	char
dict	Structure if all keys are strings not supported otherwise

#### Python Container to MATLAB Array Type Mapping

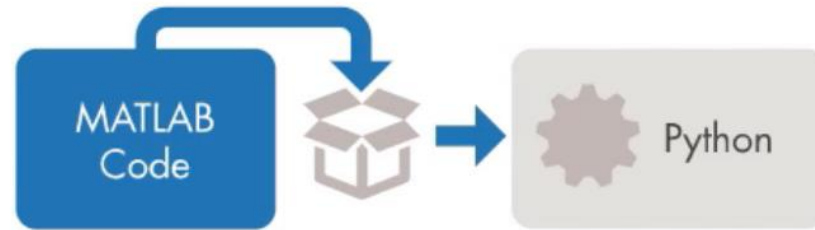
Python Input Argument Type — Container	Resulting MATLAB Data Type
matlab numeric array object (see <a href="#">MATLAB Arrays as Python Variables</a> )	Numeric array
bytearray	uint8 array
bytes (Python 3.x) bytes (Python 2.7)	uint8 array char array
list	Cell array
set	Cell array
tuple	Cell array

[https://mathworks.com/help/matlab/matlab\\_external/pass-data-to-matlab-from-python.html](https://mathworks.com/help/matlab/matlab_external/pass-data-to-matlab-from-python.html)



# Deployment

# Generate Python library from MATLAB functions

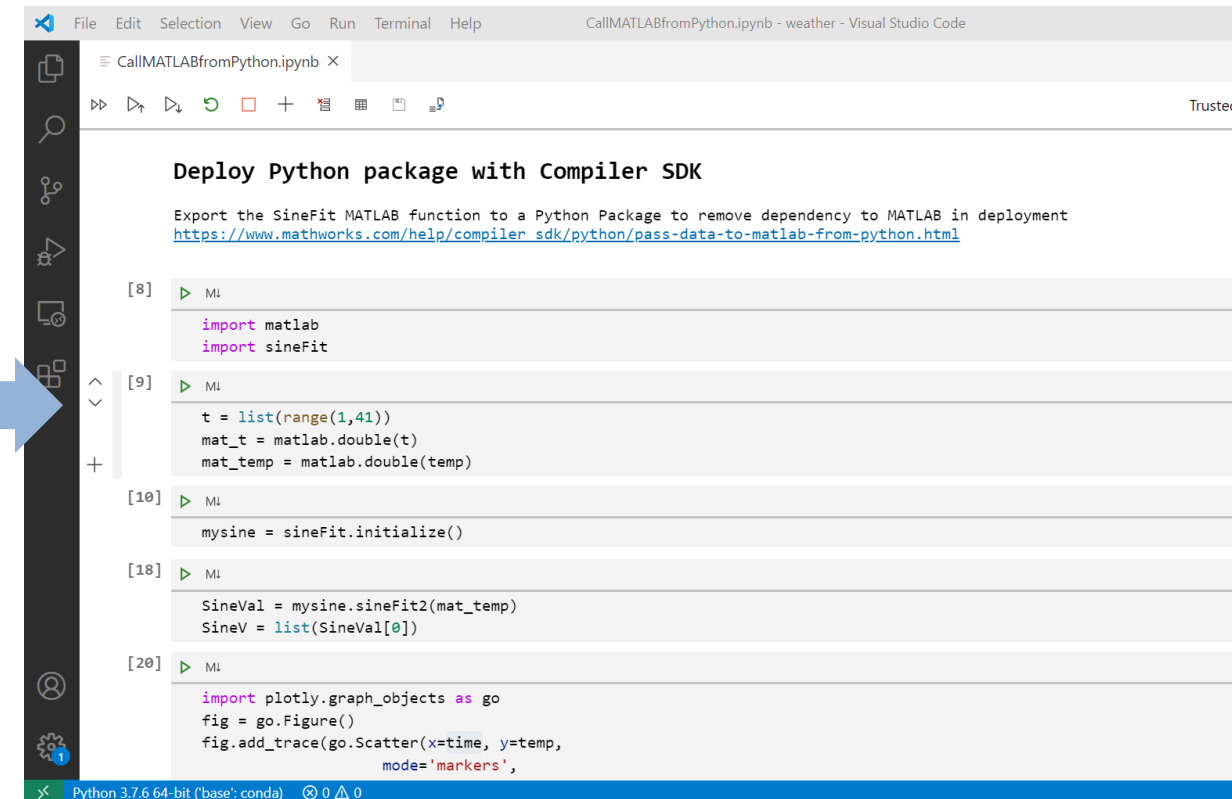
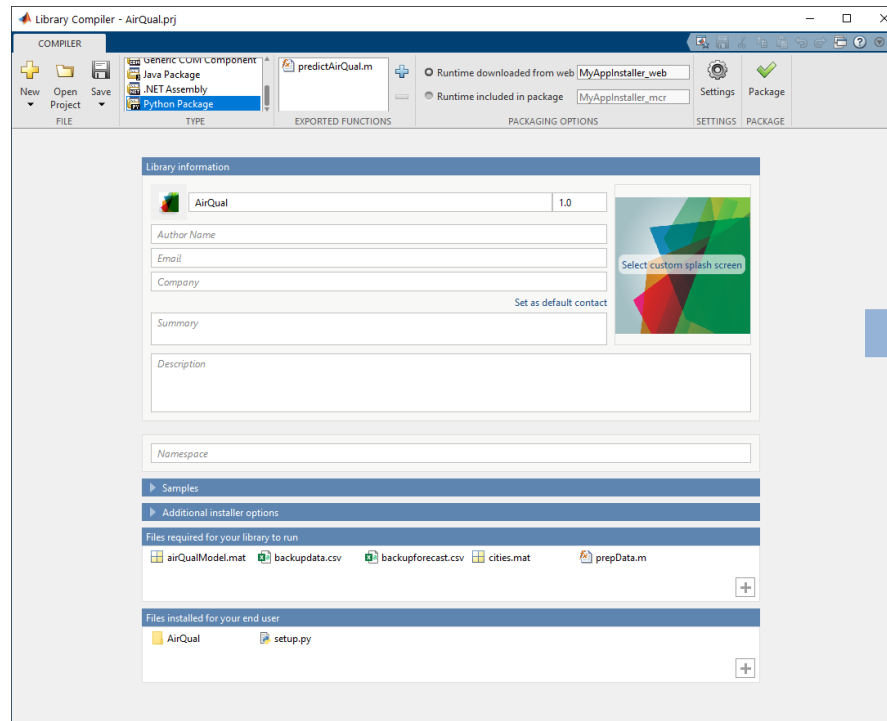


Data  
Access

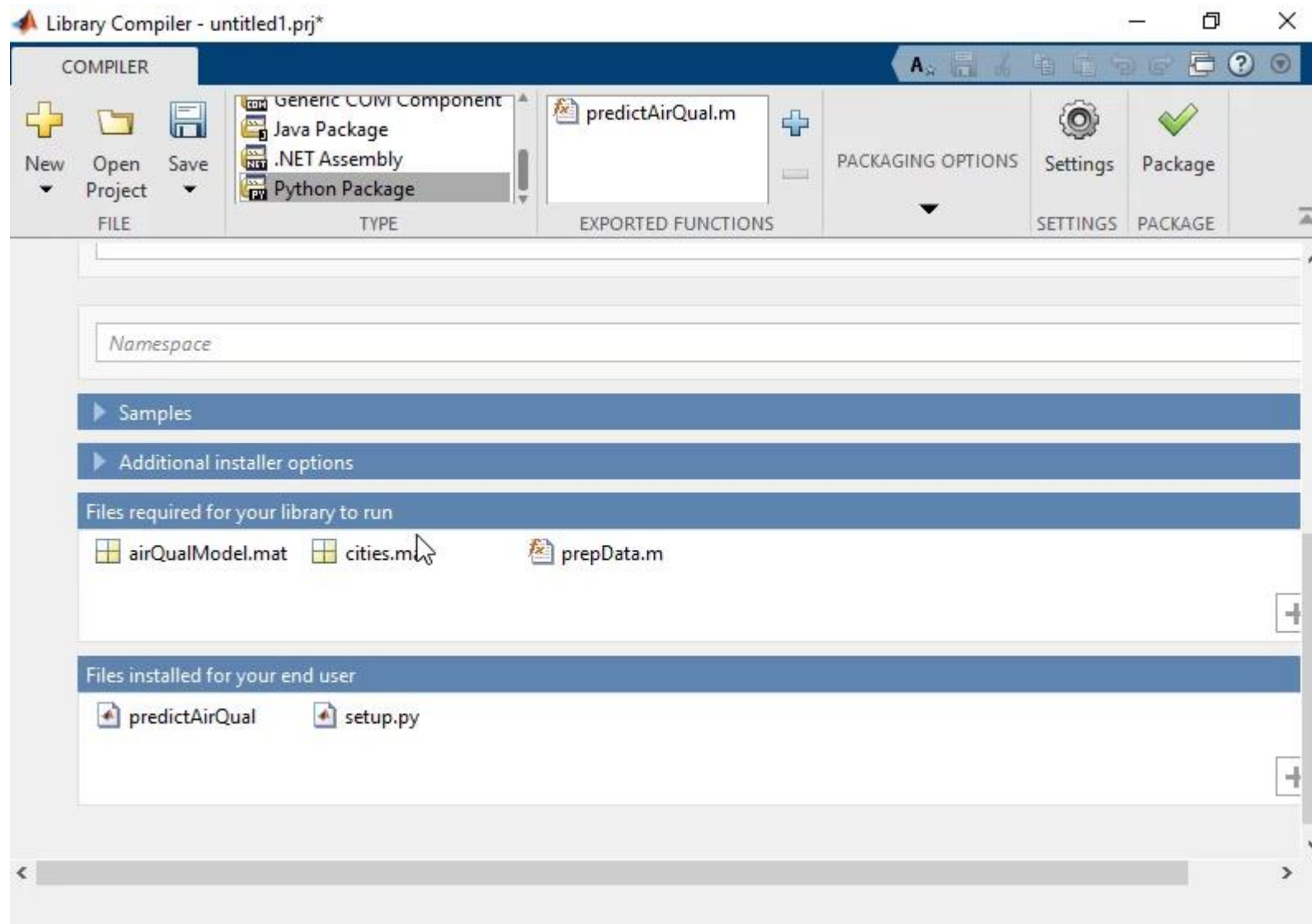
Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



# Generate Python library from MATLAB functions



Data  
Access

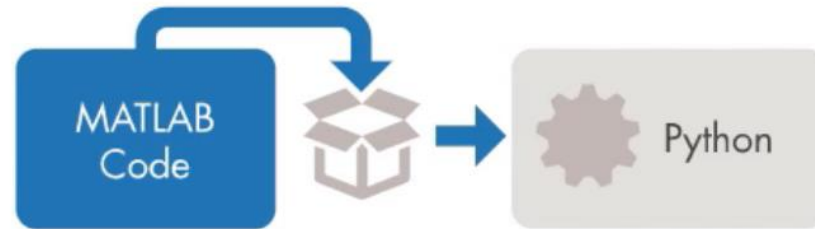
Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



# Generate Python library from MATLAB functions

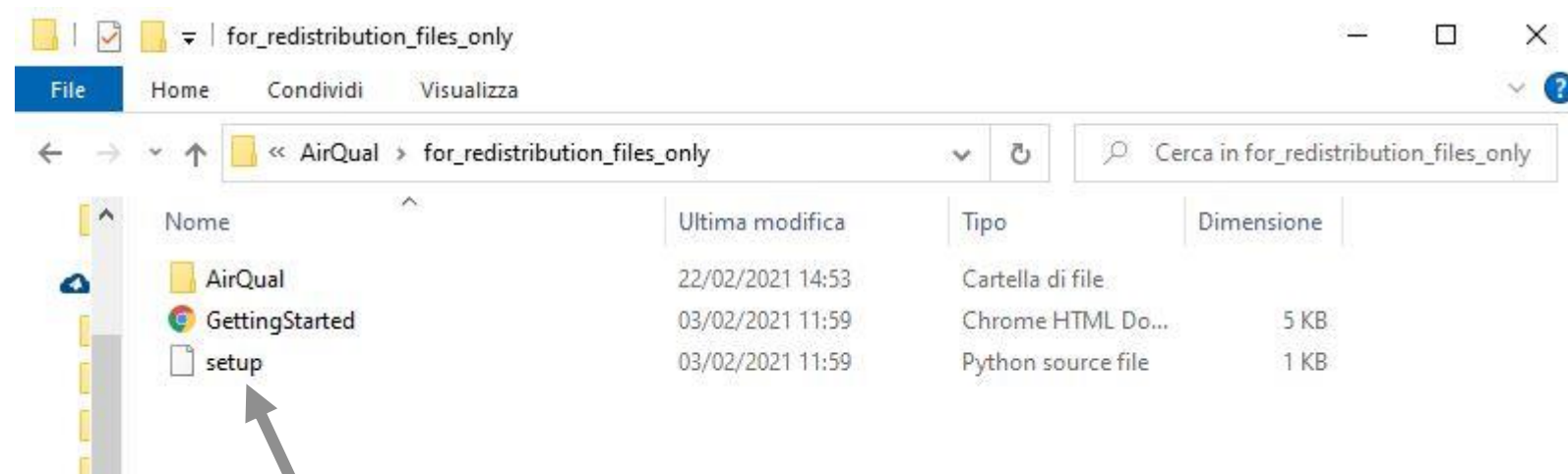


Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



Install library

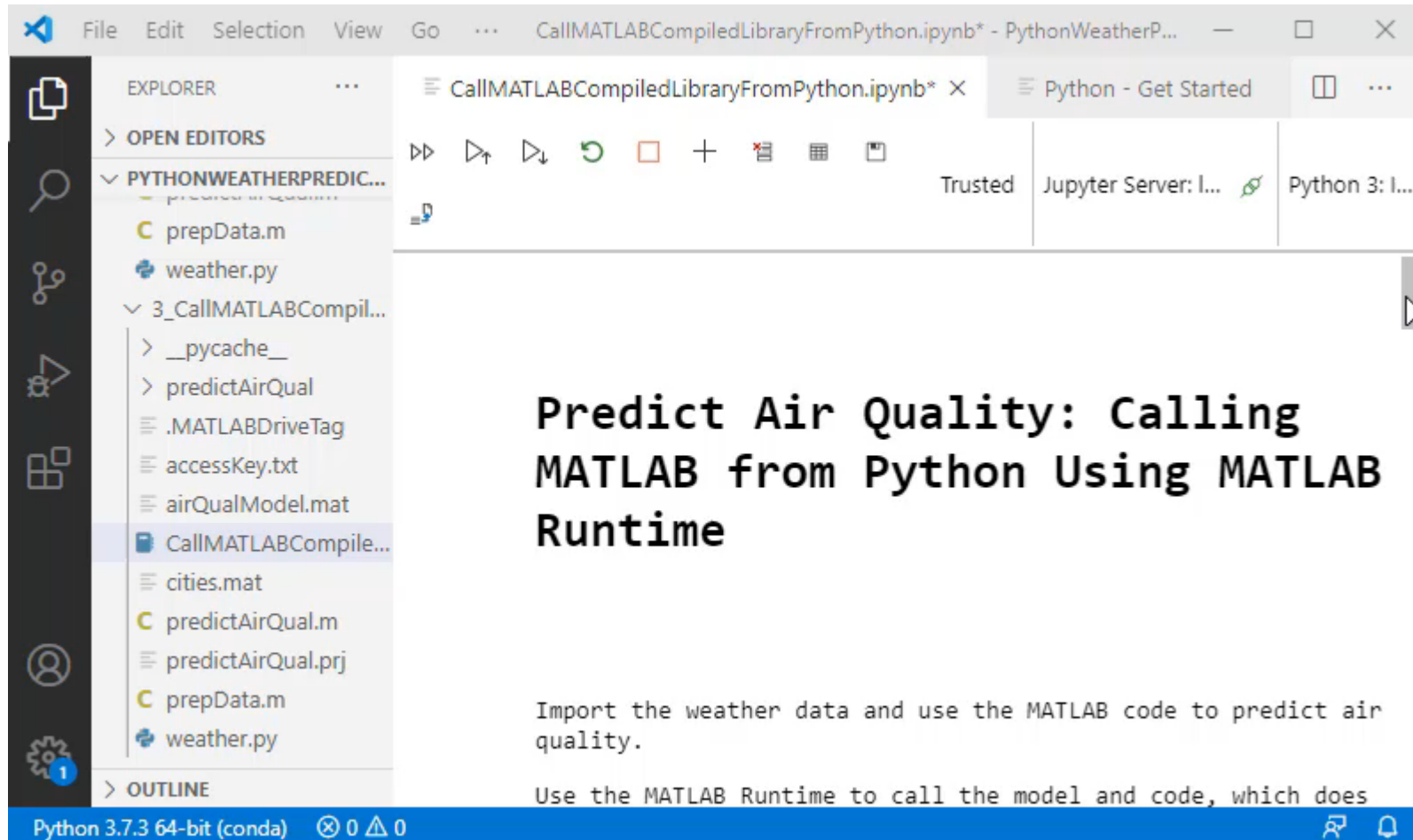
# Execute Python library from MATLAB functions

## Data Access

## Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

## Deployment



# MATLAB Production Server

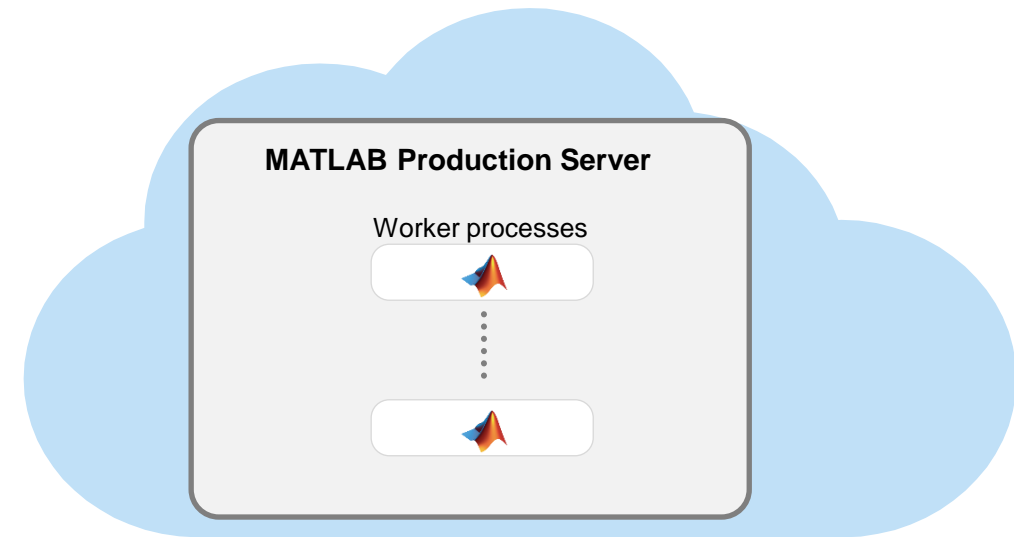
## Access functions as web services

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



Calling our function:

```
{"nargout":1,"rhs":["input"]}
```

Getting the result:

```
{"lhs":[{"mwdata":["output"],"mwsiz": [1,6],"mwtype":"char"}]}
```

# Execute Python library from MATLAB functions

## Data Access

## Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

## Deployment

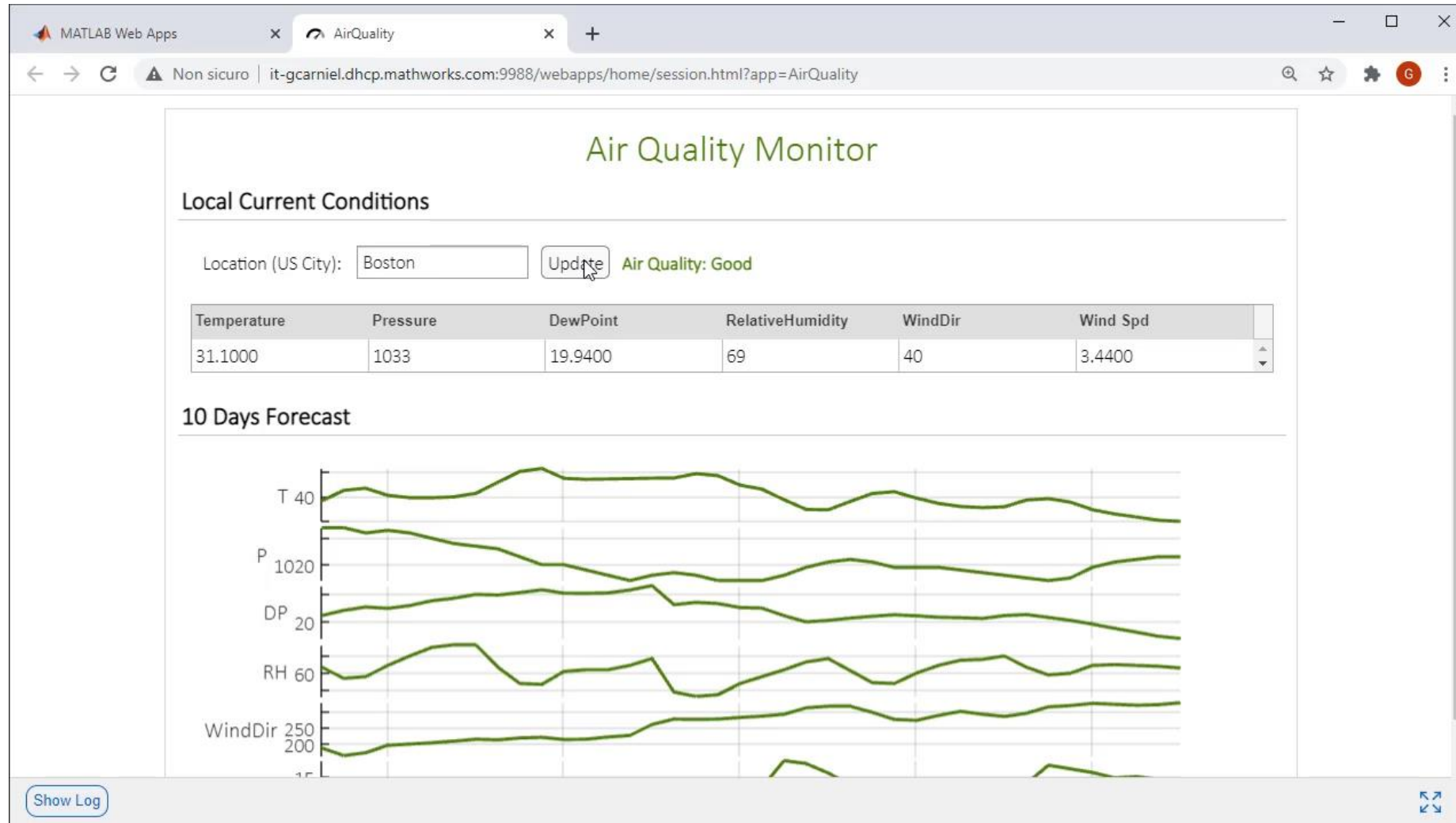
# Share MATLAB App in the Web – Central Deployment

## Data Access

## Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

## Deployment



# MATLAB App Designer

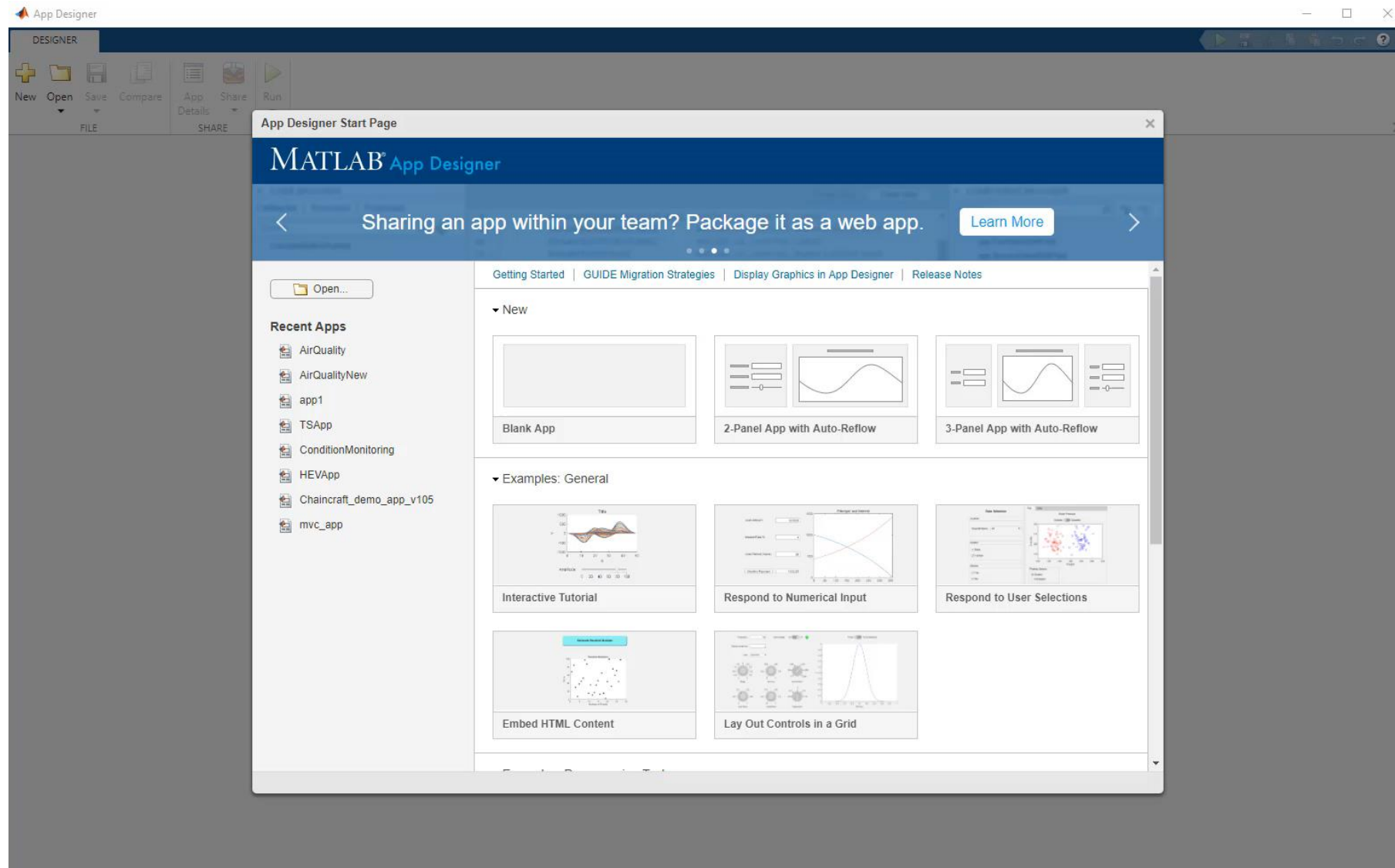
## App development for Non-Programmers

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



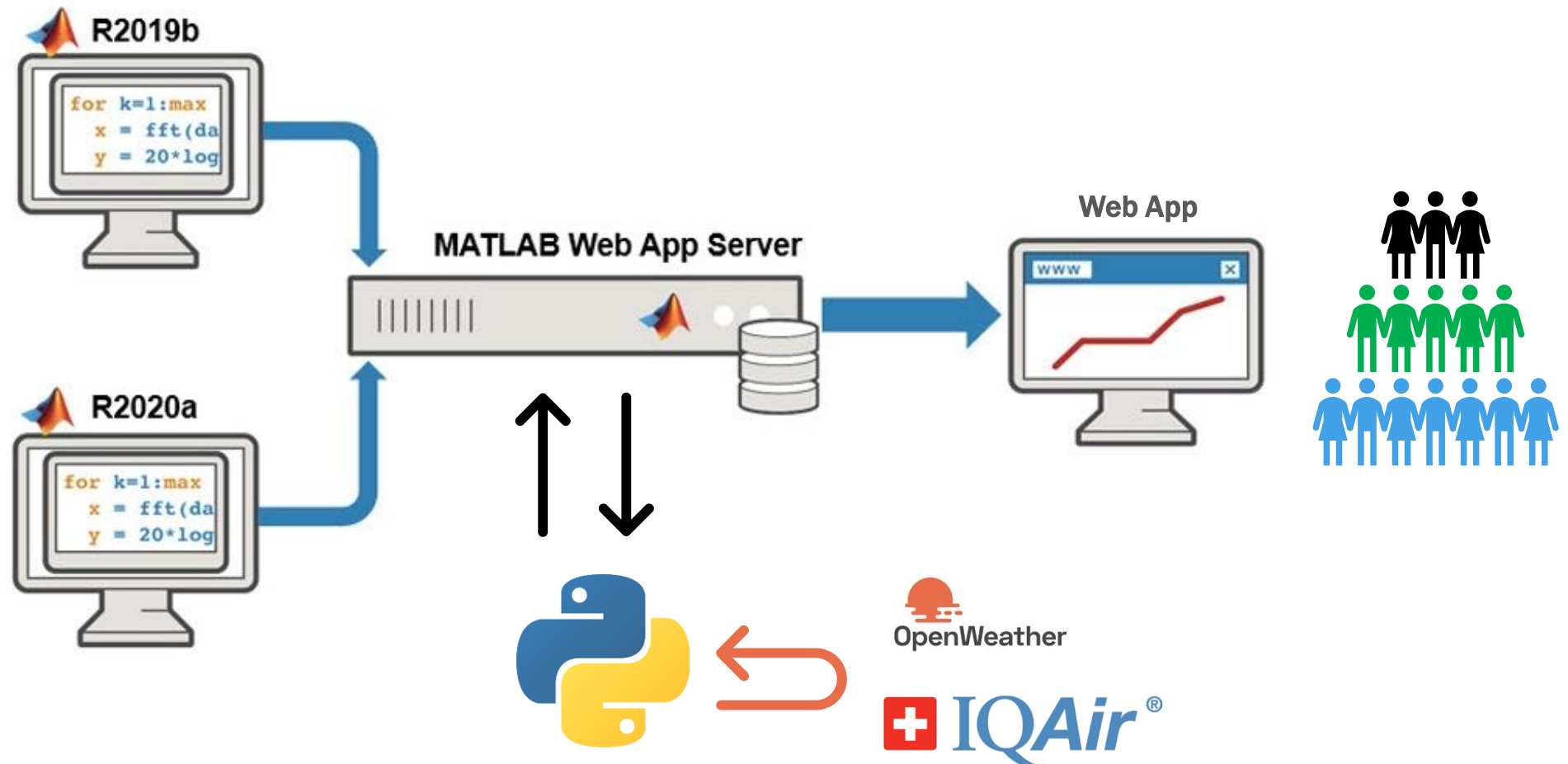
# MATLAB Web App Server – Central Deployment

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment





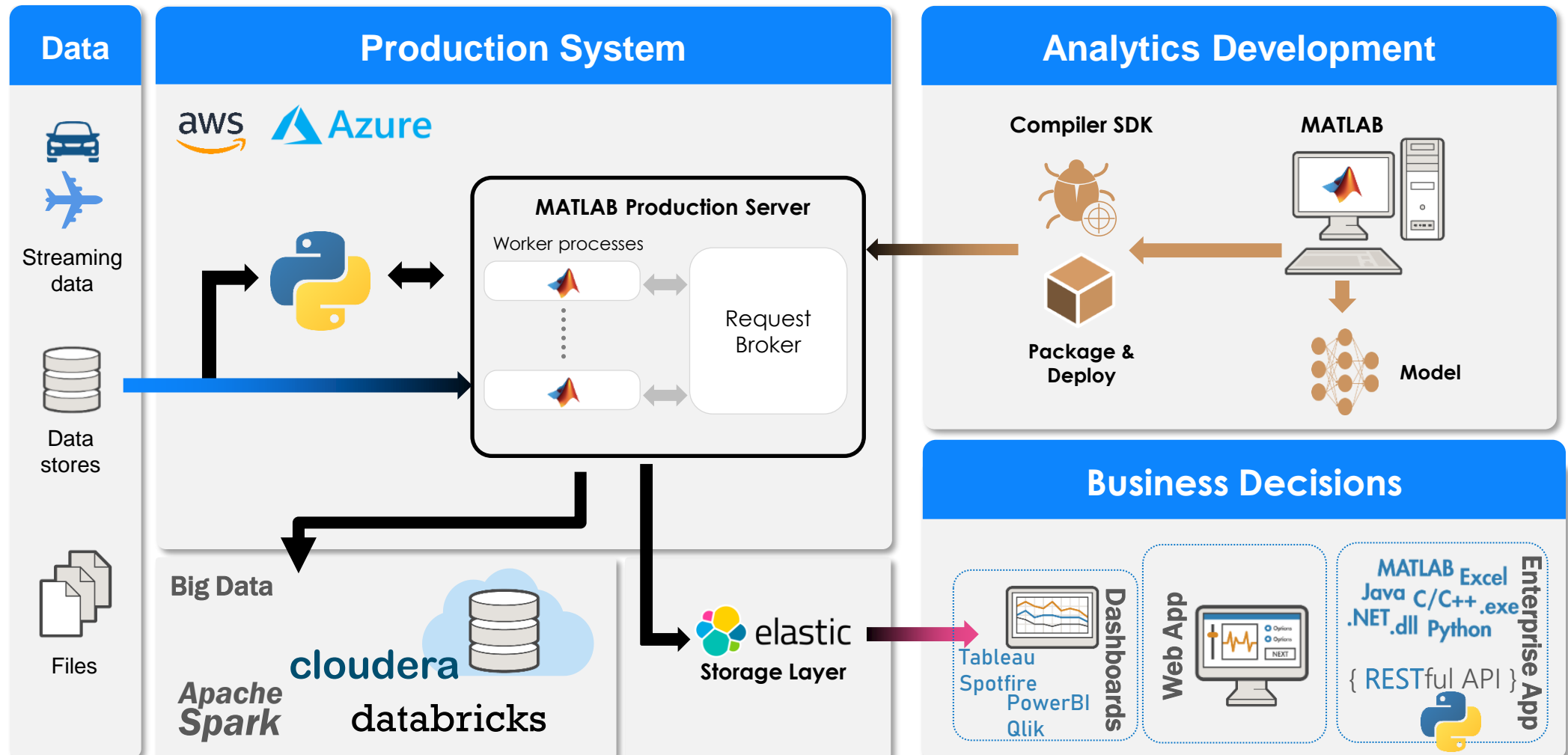
# Integrate your Production System in an IT ecosystem

Data Access

Co-Execution

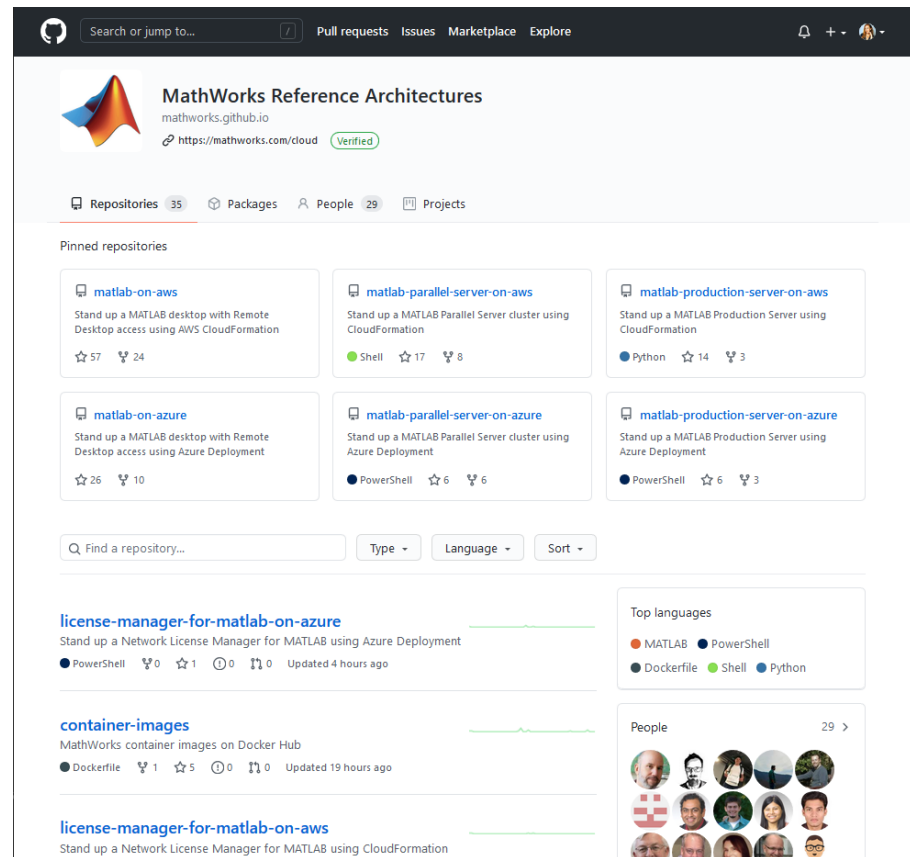
- Call Python from MATLAB
- Call MATLAB from Python

Deployment



# Use MATLAB Reference Architectures for easy cloud setup, Dockerfiles, and interfaces to OSS

- <https://github.com/mathworks-ref-arch/matlab-dockerfile>



<https://github.com/mathworks-ref-arch>

Data  
Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

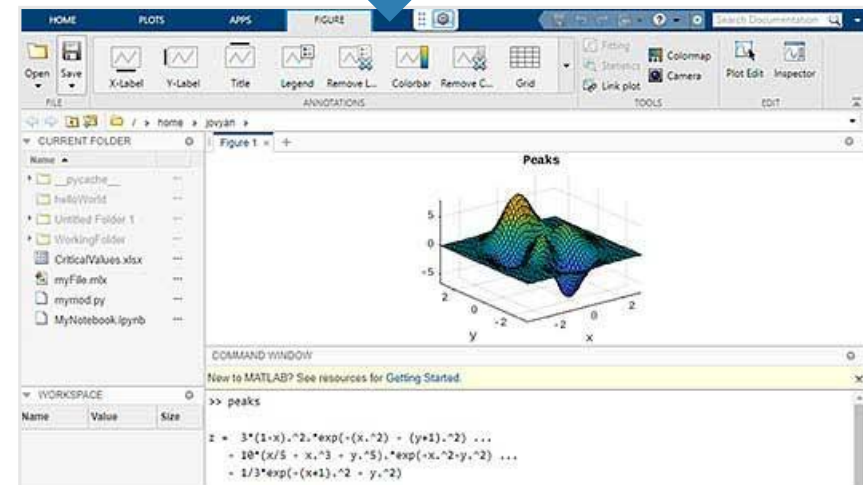
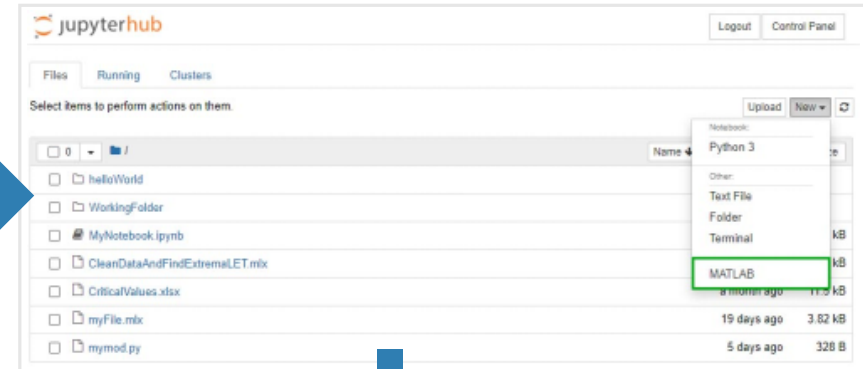
# MATLAB Integration for Jupyter

## Data Access

## Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

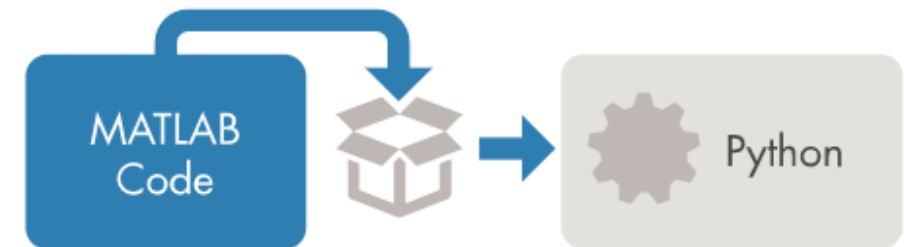
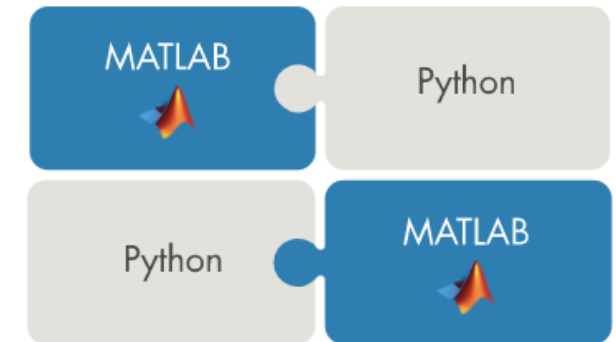
## Deployment



[MATLAB Integration for Jupyter \(mathworks.com\)](https://mathworks.com)

# Summary: Using MATLAB with Python

- Access Data
  - Weather App example
- Interoperability
  - Calling libraries written in Python from MATLAB
  - Calling MATLAB from Python
- Deploy Apps & Algos
  - Web App
  - Production API



# Python libraries in MATLAB

Documentation

# Python libraries in MATLAB (1)

## Directly call Python® functionality from MATLAB®

### Using Python Libraries

- **Access Python Modules from MATLAB - Getting Started**
  - How to create and use a Python object in MATLAB.
- **Configure Your System to Use Python**
  - How to verify you have installed a supported version of Python.
- **Call User-Defined Python Module**
  - Create a Python module used by examples in this documentation.
- **Understand Python Function Arguments**
  - Python method syntax which might be unfamiliar to MATLAB users.
- **Advanced Topics**
  - Code pattern differences you should be aware of.
- **Out-of-Process Execution of Python Functionality**
  - Execute Python scripts in processes that are separate from the MATLAB process.
- **Reload Out-of-Process Python Interpreter**
  - Reload out-of-process Python interpreter without restarting MATLAB.



```

tw = py.textwrap.TextWrapper(pyargs(...
    'initial_indent', '% ', ...
    'subsequent_indent', '% ', ...
    'width', int32(30)));
wrapped = wrap(tw,T);
wrapped = cellfun(@char,...
    cell(wrapped),...
    'UniformOutput',false);
fprintf('%s\n', wrapped{:})
% Customize the output of the
% paragraph using keyword
% arguments.
% >> |
  
```

**Call Python Function in MATLAB to Wrap Paragraph Text**

Use Python language functions and modules within MATLAB. The example calls a text-formatting module from the Python standard

[Open Live Script](#)



# Python libraries in MATLAB (2)

## Directly call Python® functionality from MATLAB®

### Passing Data

- **MATLAB to Python Data Type Mapping**
  - How MATLAB converts MATLAB data into compatible Python data types.
- **Access Elements in Python Container Types**
  - A Python container is typically a sequence type (list or tuple) or a mapping type (dict).
- **Pass Python Function to Python map Function**
  - This example shows how to display the length of each word in a list.

```
>> %Use Python array Types in MATLAB
P = py.array.array('d', 1:5)

P =

Python array:

    1     2     3     4     5

Use details function to view the properties of the Py
Use double function to convert to a MATLAB array.
```



#### Use Python Numeric Variables in MATLAB

Use Python numeric variables with MATLAB.

[Open Live Script](#)

```
F = py.os.listdir(folder)

F =

Python list with no properties.

    ['5g', 'aero', 'aeroblks_prod']
```



#### Use Python str Variables in MATLAB

Use Python str variables with MATLAB.

[Open Live Script](#)

```
listVar = py.list(...)
listVar =
Python list with no properties.

    ['Name 1', 'Name 2', 'Name 3']
```



#### Use Python list Variables in MATLAB

Use Python list variables with MATLAB.

[Open Live Script](#)

```
tupleVar = py.tuple(...)
tupleVar =
Python tuple with no properties.

    ('Name', 'Subject', 95.0)
```



#### Use Python tuple Variables in MATLAB

Use Python tuple variables with MATLAB.

[Open Live Script](#)

```
dict(pyargs('Name1', 357, 'Name2', 229.0))
dict =
Python dict with no properties.

    'Name2': 229.0, 'Name1': 357.0
```



#### Use Python dict Variables in MATLAB

Use Python dict variables with MATLAB.

[Open Live Script](#)

## Additional resources

# Resources

- General:
  - <https://www.mathworks.com/products/matlab/matlab-and-python.html>
- Python from MATLAB:
  - <https://www.mathworks.com/help/matlab/call-python-libraries.html>
- MATLAB from Python:
  - MATLAB Engine API:
    - <https://www.mathworks.com/help/matlab/matlab-engine-for-python.html>
  - MATLAB Compiler SDK:
    - [https://www.mathworks.com/help/compiler\\_sdk/python\\_packages.html](https://www.mathworks.com/help/compiler_sdk/python_packages.html)
  - Data type conversions:
    - <https://www.mathworks.com/help/matlab/python-data-types.html>
- Example:
  - <https://github.com/mathworks/matlab-with-python>

# Cheatsheet



## Using MATLAB® and Python® Together

The [≥](#) icon provides links to relevant sections of the MATLAB documentation to learn more.

### Call Python in MATLAB

Access settings and status of Python interpreter:

```
>> pe = pyenv
```

Specify version to use:

```
>> pe = pyenv("Version",3.7)
```

Call Python modules and functions:

```
py.module_name.function_name
```

```
>> py.math.sqrt(42)
```

#### Pass keyword arguments

Use pyargs to pass keyword arguments

```
>>> foo(5,bar=42)
```

```
>> py.foo(5,pyargs('bar',42))
```

#### Reload modules

Reload the module after making updates:

```
>> py.importlib.reload(module)
```

### Call MATLAB in Python

#### Install MATLAB Engine API for Python [≥](#)

Run `setup.py` from an OS command window

```
$ cd [matlabroot]/extern/engines/python
```

```
$ python setup.py install
```

#### Call MATLAB functions

Import the module and start the engine

```
>>> import matlab.engine
```

```
>>> eng =
    matlab.engine.start_matlab()
```

Call functions through the engine

```
>>> x = eng.sqrt(42.0)
```

Capture multiple outputs

```
>>> x = eng.gcd(42.0,8.0,nargout=3)
```

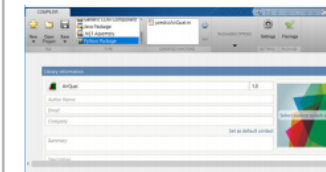
Stop the engine

```
>>> eng.exit()
```

### Create Python Package

#### Package MATLAB functions [≥](#)

Use the Library Compiler App to create a Python package for MATLAB functions



#### Invoke MATLAB functions from the Python package

```
>>> import PackageName
```

```
>>> pkg =
```

```
    PackageName.initialize()
```

```
>>> result = pkg.foo()
```

#### Close package

```
>>> pkg.terminate()
```

# NEW MATLAB for Python Users Cheat Sheet



## MATLAB® for Python® Users

The MATLAB language is designed primarily for math-intensive scientific computing. MATLAB combines a desktop environment tuned for iterative analysis with a programming language that expresses matrix and array mathematics directly. Understanding the philosophy and API design can help while learning MATLAB.

### » General Behavior

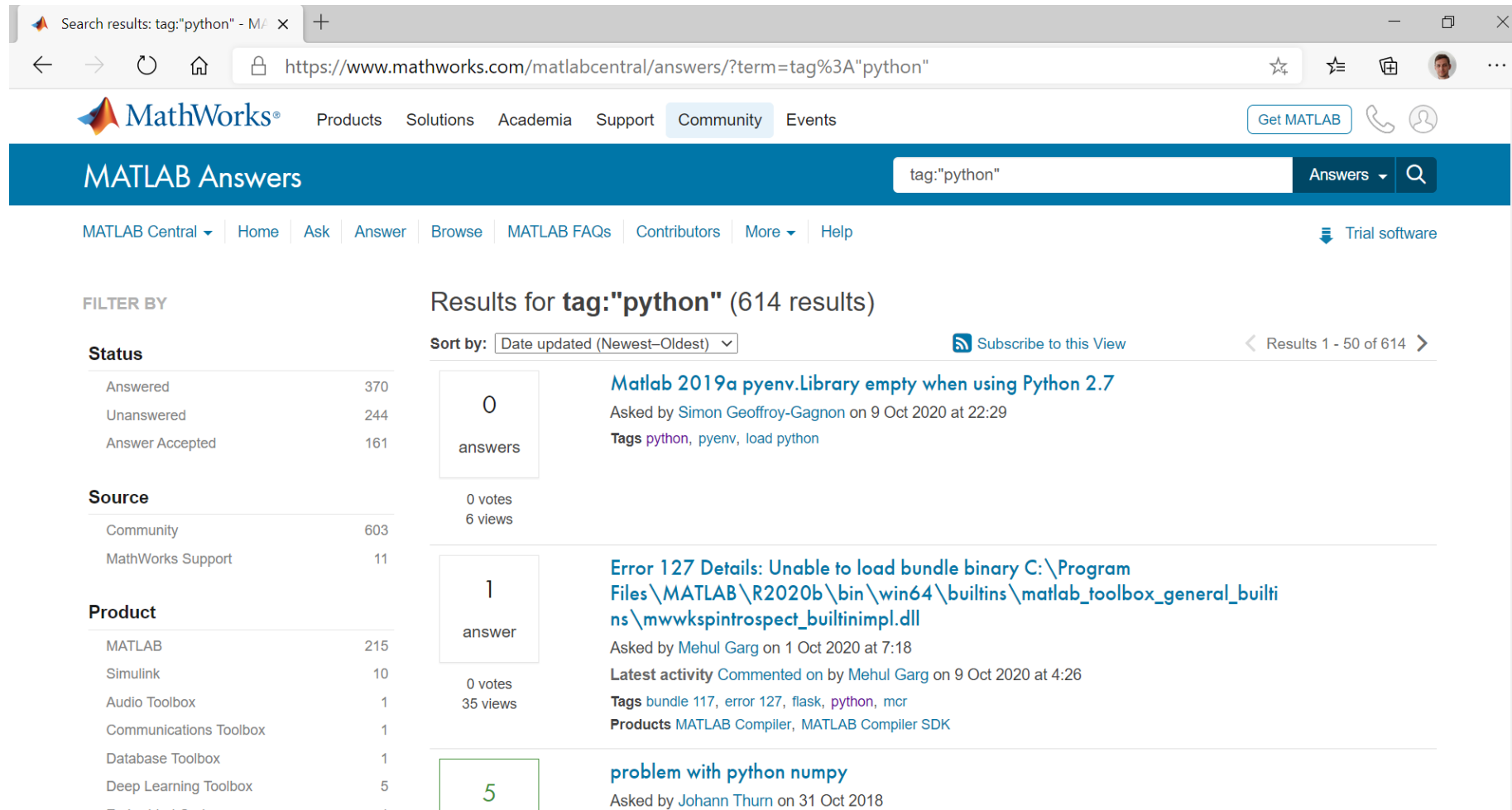
Python Syntax	MATLAB Syntax	Purpose	MATLAB Examples
#	%	Comment	%hello
print	Do not terminate with;	Print output	x
/	...	Continue to next line	x = 1+...2;
os	!	Operating system command	! echo hi
+ - * /	+ - * /	Mathematical operators	x = 1+2
**	^	Exponent	x = y^2
* / **	.* ./ .^	Element-wise operators	x = [1 2].* [3 4]
not, and, or	~ &	NOT, AND, OR logical operators	if x<2 & x>2
del	clear	Clear variable from memory	clear x y
clear	clc	Clear command window	clc

### » Referencing

MATLAB Syntax	Purpose	Example
()	Index (copy-on-write)	x(1,1)
[]	Create array	x = [1 2 3]
	Join arrays	z = [x ; y]
{ }	Create cell arrays	x = {42; "hello world"}
	Extract contents from a container	x{1,1}
.	Access class property or method	obj.Data
	Reference table or struct field	t.FieldName



- Beginning element has an index of 1.
- Indexing is left and right inclusive.
- Indexing options include N-D indexing (row,col), linear indexing (element number), and logical indexing (conditional statement).


# MATLAB Answers – tag:"python"



The screenshot shows the MATLAB Answers website interface. The browser address bar displays the URL: <https://www.mathworks.com/matlabcentral/answers/?term=tag%3Apython>. The page header includes the MathWorks logo and navigation links: Products, Solutions, Academia, Support, Community, and Events. A search bar in the header contains the text "tag:'python'". Below the header, a navigation bar shows "MATLAB Answers" and a search bar with "tag:'python'". The main content area displays "Results for tag:'python' (614 results)". On the left, there is a "FILTER BY" section with three categories: Status, Source, and Product. The Status section shows 370 Answered, 244 Unanswered, and 161 Answer Accepted. The Source section shows 603 Community and 11 MathWorks Support. The Product section shows 215 MATLAB, 10 Simulink, 1 Audio Toolbox, 1 Communications Toolbox, 1 Database Toolbox, 5 Deep Learning Toolbox, and 4 Embedded Coder. The main results area shows three questions. The first question is "Matlab 2019a pyenv.Library empty when using Python 2.7" asked by Simon Geoffroy-Gagnon on 9 Oct 2020 at 22:29, with 0 answers, 0 votes, and 6 views. The second question is "Error 127 Details: Unable to load bundle binary C:\Program Files\MATLAB\R2020b\bin\win64\builtins\matlab\_toolbox\_general\_builtins\mwwkspintrospect\_builtinimpl.dll" asked by Mehul Garg on 1 Oct 2020 at 7:18, with 1 answer, 0 votes, and 35 views. The third question is "problem with python numpy" asked by Johann Thurn on 31 Oct 2018, with 5 answers.

**Search results: tag:"python" - M/ x +**

**MathWorks®** Products Solutions Academia Support Community Events [Get MATLAB](#)  

**MATLAB Answers** tag:"python" Answers 

MATLAB Central ▾ Home Ask Answer Browse MATLAB FAQs Contributors More ▾ Help [Trial software](#)

**FILTER BY**

**Status**

Answered	370
Unanswered	244
Answer Accepted	161

**Source**

Community	603
MathWorks Support	11

**Product**

MATLAB	215
Simulink	10
Audio Toolbox	1
Communications Toolbox	1
Database Toolbox	1
Deep Learning Toolbox	5
Embedded Coder	4

**Results for tag:"python" (614 results)**

Sort by:  [Subscribe to this View](#) [Results 1 - 50 of 614](#)

**0 answers**

**Matlab 2019a pyenv.Library empty when using Python 2.7**

Asked by Simon Geoffroy-Gagnon on 9 Oct 2020 at 22:29

**Tags** python, pyenv, load python

0 votes  
6 views

**1 answer**

**Error 127 Details: Unable to load bundle binary C:\Program Files\MATLAB\R2020b\bin\win64\builtins\matlab\_toolbox\_general\_builtins\mwwkspintrospect\_builtinimpl.dll**

Asked by Mehul Garg on 1 Oct 2020 at 7:18

**Latest activity** Commented on by Mehul Garg on 9 Oct 2020 at 4:26

**Tags** bundle 117, error 127, flask, python, mcr

**Products** MATLAB Compiler, MATLAB Compiler SDK

0 votes  
35 views

**5 answers**

**problem with python numpy**

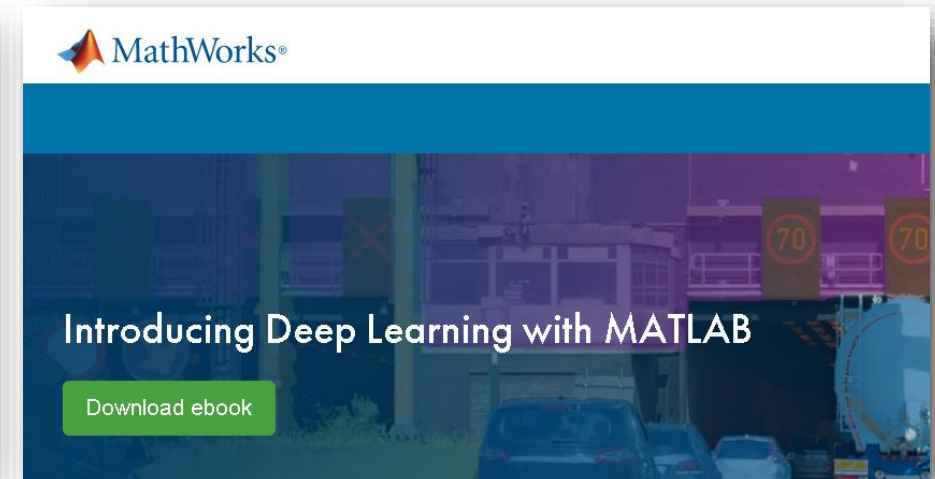
Asked by Johann Thurn on 31 Oct 2018

<https://www.mathworks.com/matlabcentral/answers/?term=tag%3Apython>



# Additional Resources

- Predictive Analytics
  - <https://www.mathworks.com/discovery/predictive-analytics.html>
- Deep Learning
  - <https://www.mathworks.com/discovery/deep-learning.html>
- Reinforcement Learning
  - <https://www.mathworks.com/products/reinforcement-learning.html>



# Learn MATLAB for Free

Hands-on practice sessions and demonstrations



FREE

## MATLAB Onramp

Get started quickly with the basics of MATLAB.



FREE

## Simulink Onramp

Get started quickly with the basics of Simulink.



FREE

## Machine Learning Onramp

Learn the basics of practical machine learning methods for classification problems.



FREE

## Deep Learning Onramp

Get started quickly using deep learning methods to perform image recognition.



FREE

## Reinforcement Learning Onramp

Master the basics of creating intelligent controllers that learn from experience.



FREE

## Image Processing Onramp

Learn the basics of practical image processing techniques in MATLAB.



FREE

## Signal Processing Onramp

An interactive introduction to signal processing methods for spectral analysis.



FREE

## Simscape Onramp

Learn the basics of simulating physical systems in Simscape.



FREE

## Stateflow Onramp

Learn the basics of creating, editing, and simulating state machines in Stateflow.



FREE

## Control Design Onramp with Simulink

Get started quickly with the basics of feedback control design in Simulink.



FREE

## Optimization Onramp

Learn the basics of solving optimization problems in MATLAB using the problem-based approach.



## Circuit Simulation Onramp

7 modules | 2 hours | Languages

Learn the basics of simulating electrical circuits in Simscape.

<https://matlabacademy.mathworks.com/#getting-started>

## Available Courses

<https://matlabacademy.mathworks.com/>



### MATLAB Fundamentals

Details



### Machine Learning with MATLAB

Details



### Deep Learning with MATLAB

Details



### MATLAB for Data Processing and Visualization

Details



### Solving Ordinary Differential Equations with MATLAB

Details



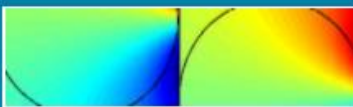
### Introduction to Linear Algebra

Details



### Introduction to Statistical Methods with MATLAB

Details



### Solving Nonlinear Equations with MATLAB

Details



### MATLAB Programming Techniques

Details



### MATLAB for Financial Applications

Details



### Introduction to Symbolic Math with MATLAB



### Image Processing with MATLAB



# MATLAB and Simulink Training

[Training Overview](#) | 
 [Find a Course ▾](#) | 
 [Get Certified](#) | 
 [Training at Your Facility](#) | 
 [More ▾](#)

[» My Courses](#) | 
 [? Contact Training](#)

## MATLAB and Simulink Course Schedule

Results 1 - 50 of 223 [»](#)

FILTERED BY English ✕ [Clear Filters](#)

Dates	Course	Location	Language	Price	Register
On Demand	 <b>MATLAB Programming Techniques</b> 180 days of full access from the day of purchase	Self-Paced	English	USD 350	
On Demand	 <b>Machine Learning with MATLAB</b> 180 days of full access from the day of purchase	Self-Paced	English	USD 350	
On Demand	 <b>MATLAB Fundamentals</b> 180 days of full access from the day of purchase	Self-Paced	English	USD 500	
On Demand	 <b>Machine Learning with MATLAB</b> 180 days of full access from the day of purchase	Self-Paced	English	USD 350	
On Demand	 <b>Deep Learning with MATLAB</b> 180 days of full access from the day of purchase	Self-Paced	English	USD 350	
On Demand	 <b>MATLAB for Data Processing and Visualization</b> 180 days of full access from the day of purchase	Self-Paced	English	USD 200	

**Campus Wide License Program Offer FREE !!**

# MATLAB Certification

Certification sets individuals apart in the job market and can help accelerate professional growth

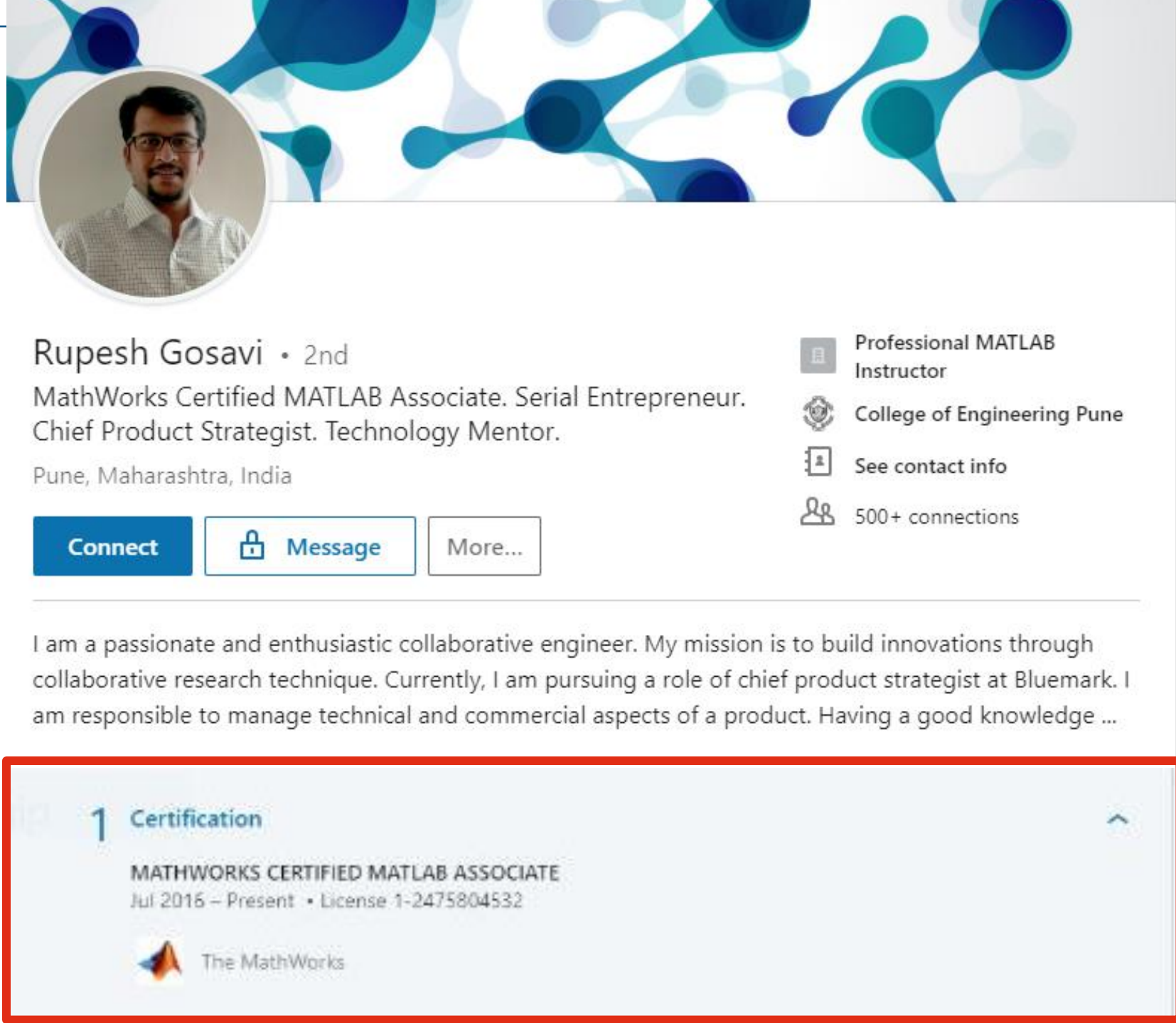
Two paid levels of certification are offered:

[Certified MATLAB Associate](#)

[Certified MATLAB Professional](#)

Share credentials on platforms such as

[LinkedIn](#) and [Facebook](#)



The image shows a LinkedIn profile for Rupesh Gosavi. The profile includes a circular profile picture of a man with glasses and a beard. The background of the header is a blue and white abstract pattern. The profile name is 'Rupesh Gosavi • 2nd'. The headline is 'MathWorks Certified MATLAB Associate. Serial Entrepreneur. Chief Product Strategist. Technology Mentor.' The location is 'Pune, Maharashtra, India'. The profile has three buttons: 'Connect', 'Message', and 'More...'. On the right side, there are four icons with text: 'Professional MATLAB Instructor', 'College of Engineering Pune', 'See contact info', and '500+ connections'. The bio states: 'I am a passionate and enthusiastic collaborative engineer. My mission is to build innovations through collaborative research technique. Currently, I am pursuing a role of chief product strategist at Bluemark. I am responsible to manage technical and commercial aspects of a product. Having a good knowledge ...'. At the bottom, there is a section titled '1 Certification' which lists 'MATHWORKS CERTIFIED MATLAB ASSOCIATE' with the dates 'Jul 2016 – Present' and the license number '1-2475804532'. The MathWorks logo is also present.

**Rupesh Gosavi** • 2nd

MathWorks Certified MATLAB Associate. Serial Entrepreneur. Chief Product Strategist. Technology Mentor.

Pune, Maharashtra, India

[Connect](#) [Message](#) [More...](#)

[Professional MATLAB Instructor](#)

[College of Engineering Pune](#)


[See contact info](#)

[500+ connections](#)

I am a passionate and enthusiastic collaborative engineer. My mission is to build innovations through collaborative research technique. Currently, I am pursuing a role of chief product strategist at Bluemark. I am responsible to manage technical and commercial aspects of a product. Having a good knowledge ...

**1 Certification**

**MATHWORKS CERTIFIED MATLAB ASSOCIATE**  
Jul 2016 – Present • License 1-2475804532

 The MathWorks

# Certificates Learning Course



## Course Completion Certificate

Kantika Wongkasem

has successfully completed 100% of the self-paced training course

Deep Learning Onramp

  
DIRECTOR, TRAINING SERVICES

06 April 2021



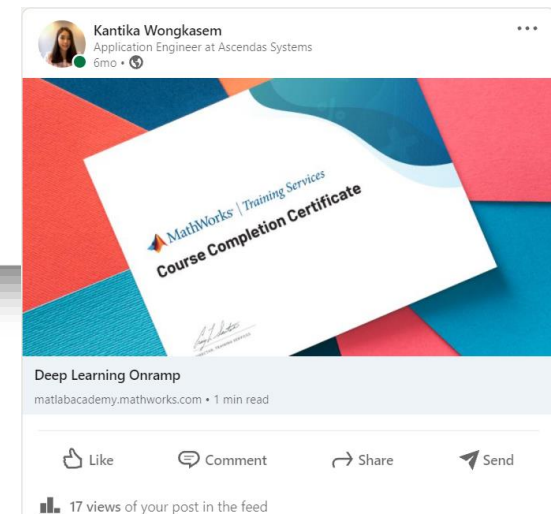
### Progress Report

**Name:** Kantika Wongkasem  
**Course:** Deep Learning Onramp  
**Progress:** 100% complete (as of 06 April 2021)

#### Chapters

1. Introduction 100%
2. Using Pretrained Networks 100%
3. Managing Collections of Image Data 100%
4. Performing Transfer Learning 100%
5. Conclusion 100%

Release: R2020b | Language: English



Share Certificates upgrade profile on



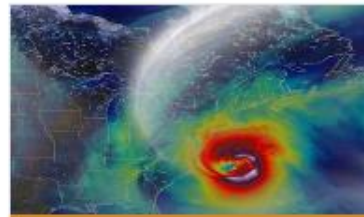
Free sets of course materials developed by faculty from top universities  
Curricula available for **all STEM disciplines** and at multiple levels

# MATLAB Courseware



## Teaching Quantitative Finance and Risk Management with MATLAB

- » Integrate MATLAB into your Quantitative Finance and Risk Management curriculum



## Teaching Computational Science Using MATLAB

- » Integrate MATLAB into your robust data analysis, data visualization and exploration curriculum



## Teaching Biology with MATLAB

- » Integrate MATLAB into your Biology curriculum



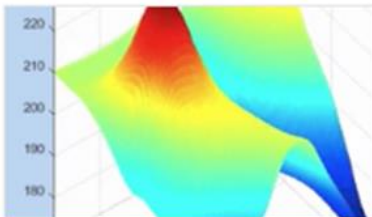
## Teaching Calculus with MATLAB

- » Integrate MATLAB into your Calculus curriculum



## Teaching Chemistry with MATLAB

- » Integrate MATLAB into your Chemistry curriculum



## Teaching Geoscience with MATLAB

- » Integrate MATLAB into your Geoscience curriculum



## Teaching Physics with MATLAB

- » Integrate MATLAB into your Physics curriculum



## Teaching Psychology and Neuroscience with MATLAB

- » Integrate MATLAB into your Psychology and Neuroscience curriculum



## Teaching Data Science with MATLAB

- » Integrate MATLAB into your Data Science curriculum



## Teaching Deep Learning with MATLAB

- » Integrate MATLAB into your Deep Learning curriculum



## Teaching Econometrics with MATLAB

- » Integrate MATLAB into your Econometrics curriculum



# MATLAB and Simulink Based Books

- More than 2000 titles in 26 languages for educational and professional use

- Subjects include:

- Biosciences and Biomedical
- Chemistry and Chemical Engineering
- Control Systems
- Digital Signal Processing
- Earth Sciences
- Economics and Computational Finance
- Image and Video Processing
- Mathematics
- Mechanical Engineering
- Neural Networks and Fuzzy Logic
- Physics
- Programming and Computer Science
- Robotics
- System Modeling and Simulation



# Ascendas Solutions: Tools & Support

## Technical Support

### Technical Support:

✉ : [support@techsource-asia.com](mailto:support@techsource-asia.com)

### Contact Us:

📄 : <https://www.ascendas-asia.com>

✉ : [th-events@ascendas-asia.com](mailto:th-events@ascendas-asia.com)

☎ : +66 2 234 6721



[TechSource Systems | Events \(techsource-asia.com\)](https://techsource-asia.com)