How to call Python function in MatLab step-by-step

Updated Jan 25, 2025

# Introduction

The following steps will help prepare the user to call Python functions (and entire scripts) in MATLAB version R2019b. This guide covers the installation of a compatible Python version, configuring MATLAB to recognize Python, and executing Python functions effectively. By following these steps, users can integrate Python's versatile programming capabilities with MATLAB's robust computational environment.

**Step 1: Install a Compatible Python Version**

MATLAB requires a compatible version of Python to ensure smooth integration. The version of Python must:

1. Match the architecture of MATLAB (e.g., both must be 64-bit).
2. Be supported by your MATLAB version. As of recent MATLAB releases, Python versions from 3.7 to 3.10 are commonly supported. Always refer to the official MATLAB documentation for specific compatibility details.

**Installation Steps**

1. Download a compatible Python installer from the official Python website: [python.org](https://www.python.org/downloads/).
   * Ensure you select the **64-bit version**.
2. Run the installer and:
   * Select **"Add Python to PATH"** during installation.
   * Choose **"Customize Installation"** to include optional features like pip.
3. Verify the installation by opening a Command Prompt (CMD) and running:

python --version

python -c "import platform; print(platform.architecture()[0])"

Ensure the output confirms a 64-bit Python installation.

**Step 2: Configure MATLAB to Recognize Python**

**Set the Python Environment in MATLAB**

1. Launch MATLAB.
2. Configure MATLAB to use the installed Python version by running the following command:

pyenv('Version', 'C:\Path\To\Python\python.exe');

Replace C:\Path\To\Python\python.exe with the full path to your Python executable.

1. Verify the Python version configured in MATLAB:

pyversion

The output should display the Python version, executable path, and other details. If these fields are empty, troubleshoot by verifying the Python installation and path.

**Step 3: Prepare the Python Script**

**Write Your Python Script**

1. Create a Python file (e.g., myscript.py) that contains the functions you want to call.
2. Save the file in a directory accessible to MATLAB.

**Add the Script Path in MATLAB**

In MATLAB, ensure the folder containing the Python script is in MATLAB’s path:

addpath('C:\Path\To\Python\Script\Folder');

Replace C:\Path\To\Python\Script\Folder with the actual directory path.

**Step 4: Call Python Functions in MATLAB**

**Basic Function Calls**

Use the py prefix to call Python functions directly from MATLAB. For example:

% Call the add\_numbers function from myscript.py

result\_add = py.myscript.add\_numbers(10, 20);

disp(result\_add); % Output: 30

% Call the multiply\_numbers function

result\_multiply = py.myscript.multiply\_numbers(5, 4);

disp(result\_multiply); % Output: 20

**Passing and Receiving Data**

MATLAB can automatically convert basic data types (numbers, strings) to Python-compatible types. For more complex types:

* Convert MATLAB cell arrays to Python lists:

py\_list = py.list({1, 2, 3});

* Convert Python lists back to MATLAB arrays:

matlab\_array = double(py.list({1, 2, 3}));