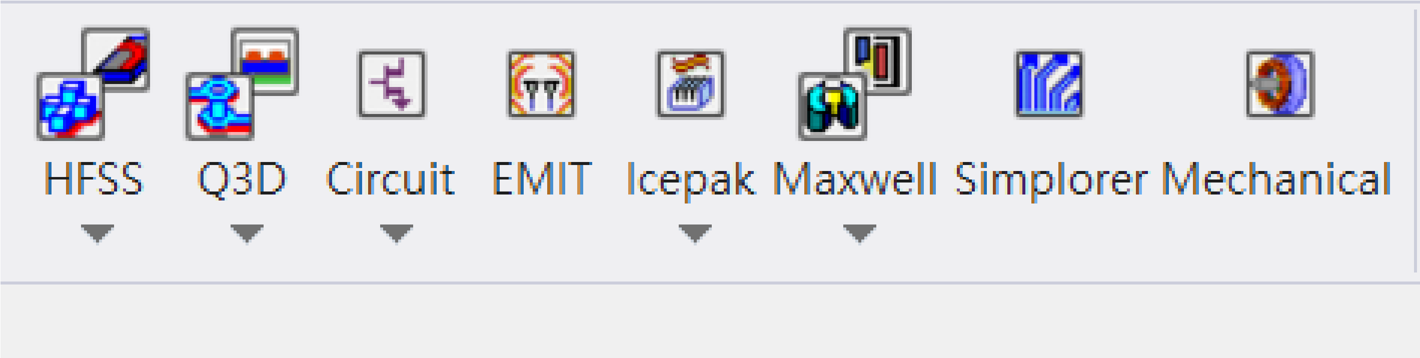
AEDT Automation Development with PyAEDT

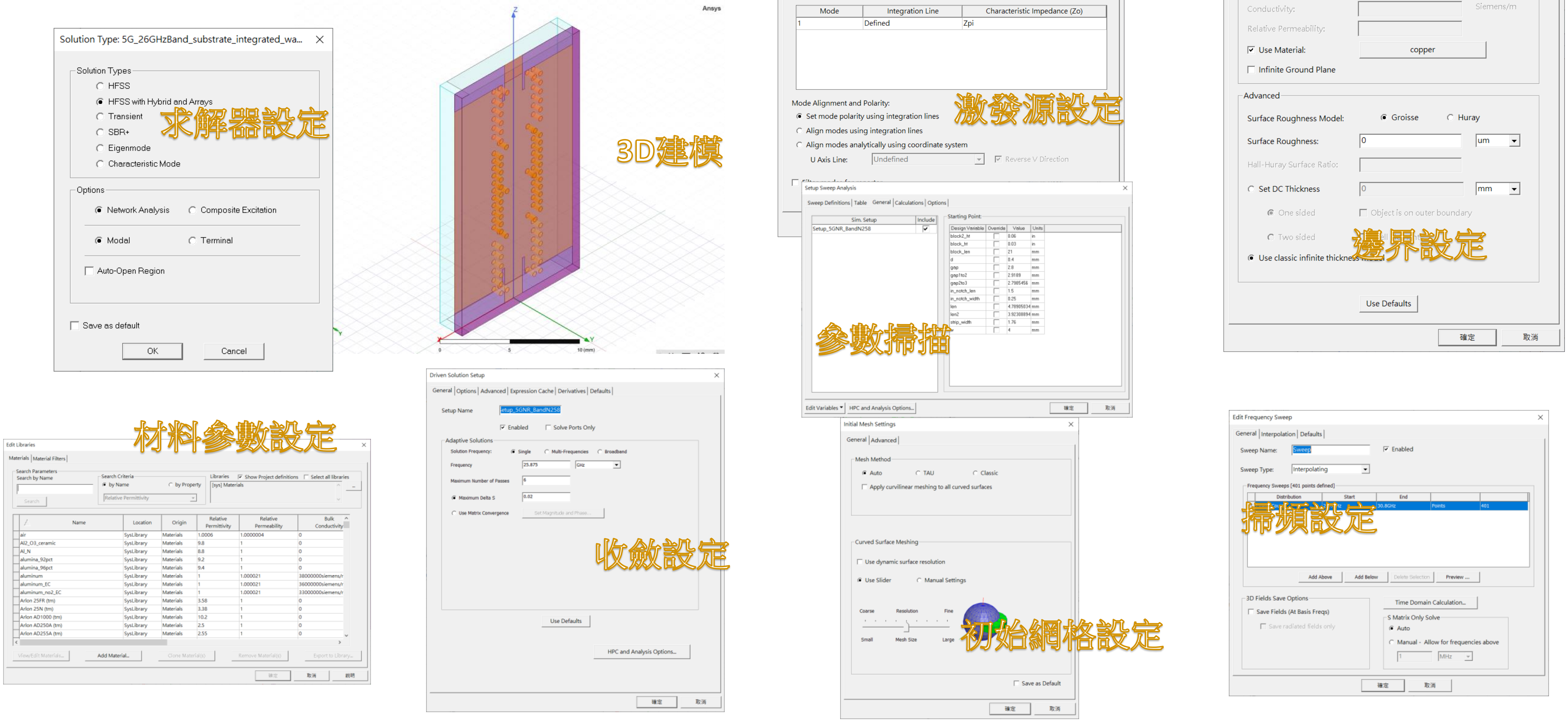
Lin, Ming Chih

Agenda

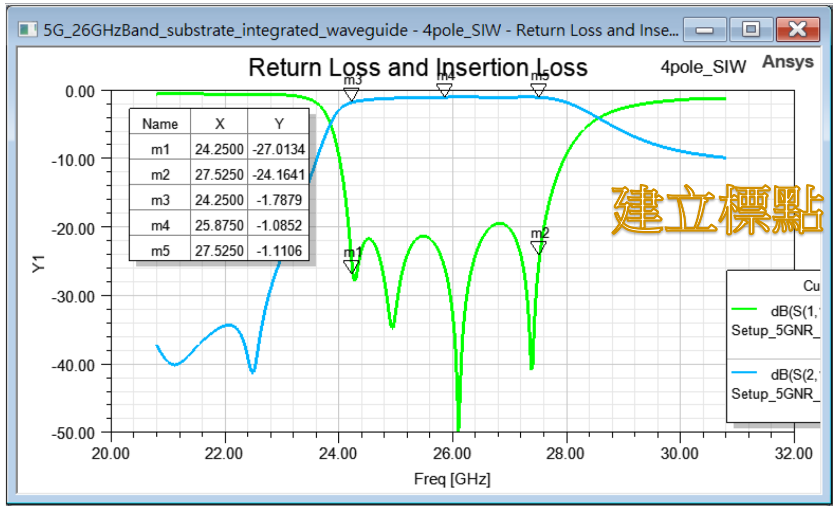
- PyAEDT Overview
- Examples Demo
- Installation
- IDE Introduction
- Scripting Demo

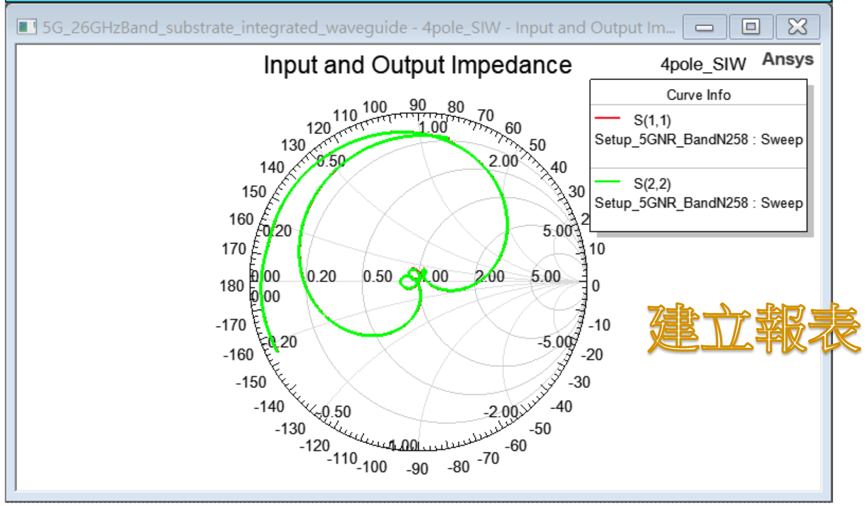


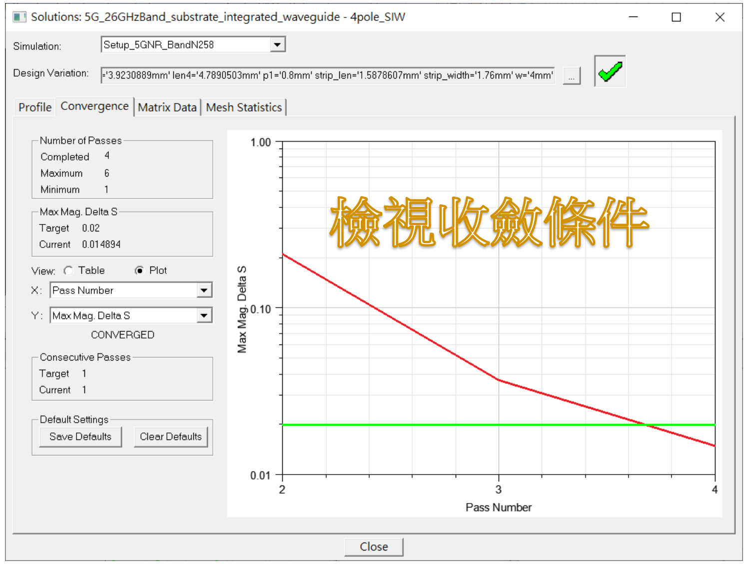
PreSim Setting in AEDT HFSS

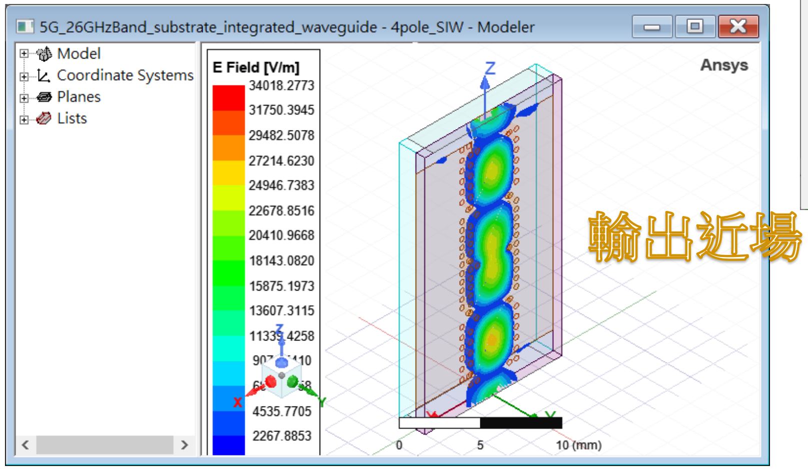


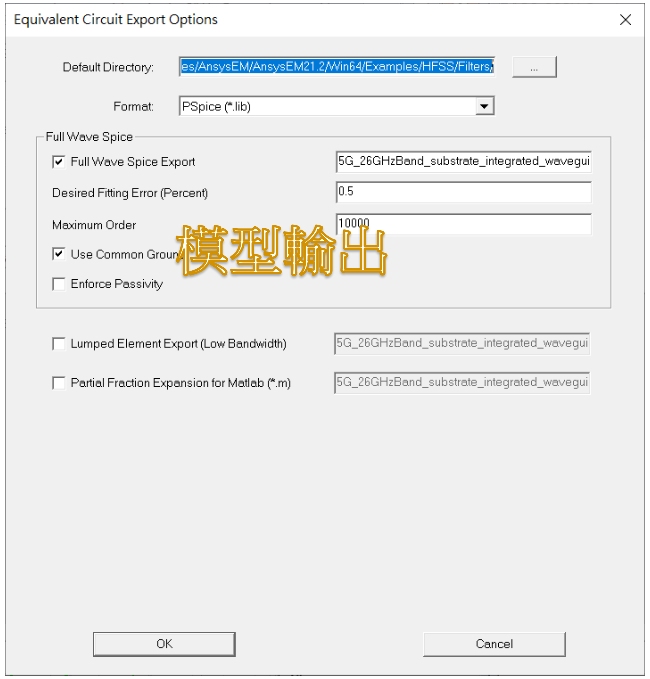
PostSim Setting in AEDT HFSS











Typical GUI-Based Setting Flow

GUI is the native operation mode but there exist drawbacks:

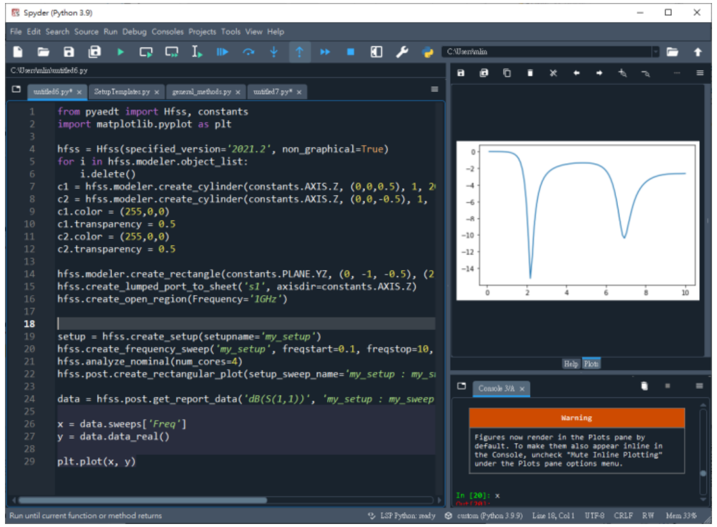
- GUI license is long time occupied.
- Setting mistakes is easy to happen.
- Settings are distributed in dozes of windows, difficult to inspect.
- Comments of the settings are not supported.
- Hard to compare setting and data between projects and designs.
- Takes time to organize report and simulation data
- Advanced data processing is limited(e.g. Machine Learning)

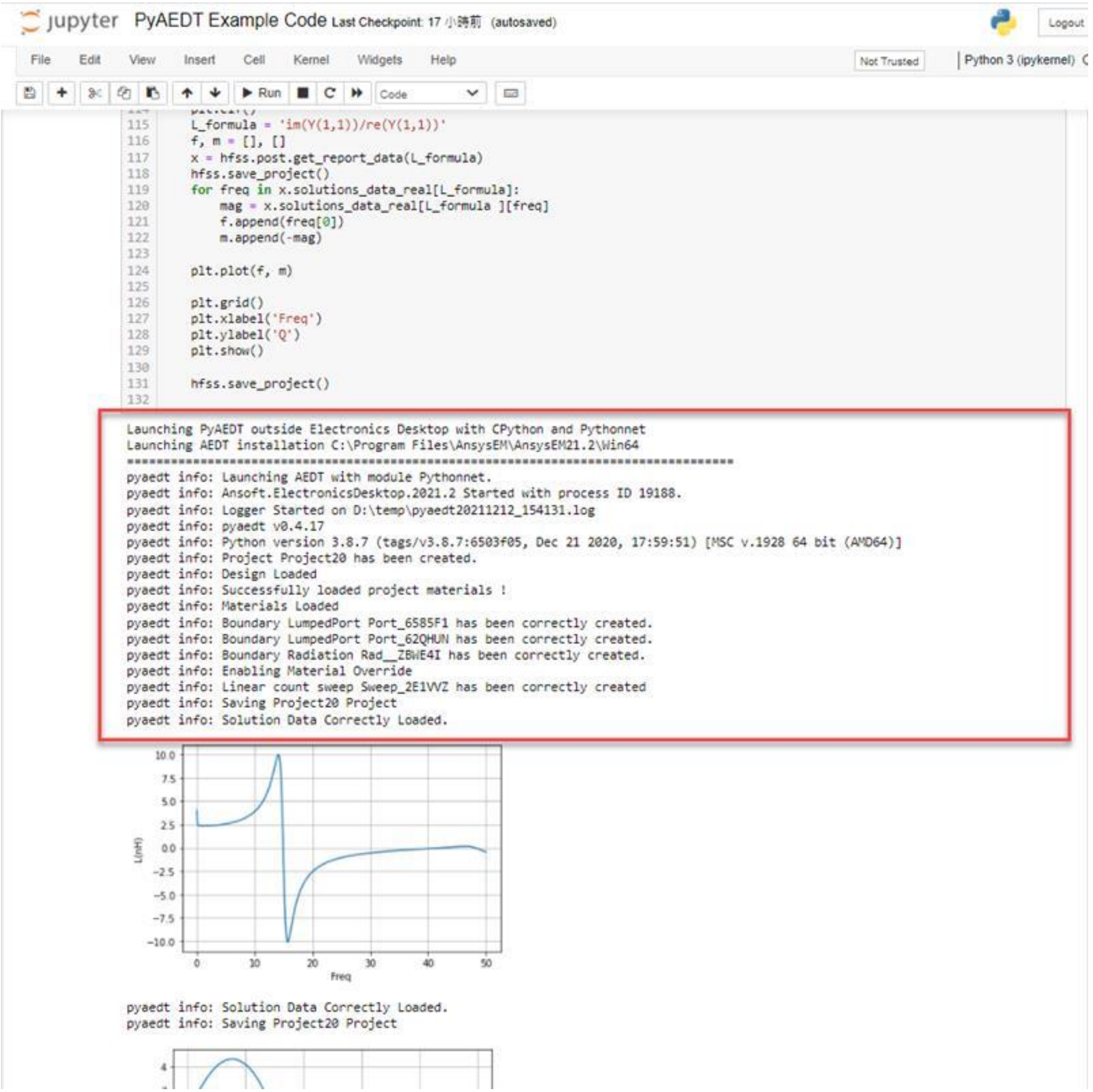
Python + PyAEDT + Jupyter Notebook/Streamlit

- Research and Design Optimization, like:
  - Design of Experiment
  - Variation and Yield Analysis
  - Data Processing
- Multi-Physics
  - Electrical-Thermal-Mechanical
  - Electrical-Optical
  - …
- Automate Routine Jobs, like:
  - Model Extraction
  - Signal Process
- Interactive Training Material, like:
  - Mode Theory
  - Array Antenna Mechanism

Examples

- Dipole Antenna Sweep
- Spiral Inductor Simulation
- PCB Model Extraction
- Package 3D Modeling





PyAEDT Overview

What is PyAEDT

- PyAEDT is a Python library that interacts directly with the AEDT API to make scripting simpler for the end user. It uses an architecture that can be reused for all AEDT 3D products (HFSS, Icepak, Maxwell 3D, Q3D and Mechanical) as well as 2D tools and circuit tools like Nexxim and Simplorer.
- Finally, it provides scripting capabilities in Ansys layout tools like HFSS 3D Layout and EDB. Its class and method structures simplify operation for the end user while reusing information as much as possible across the API.

Classical API

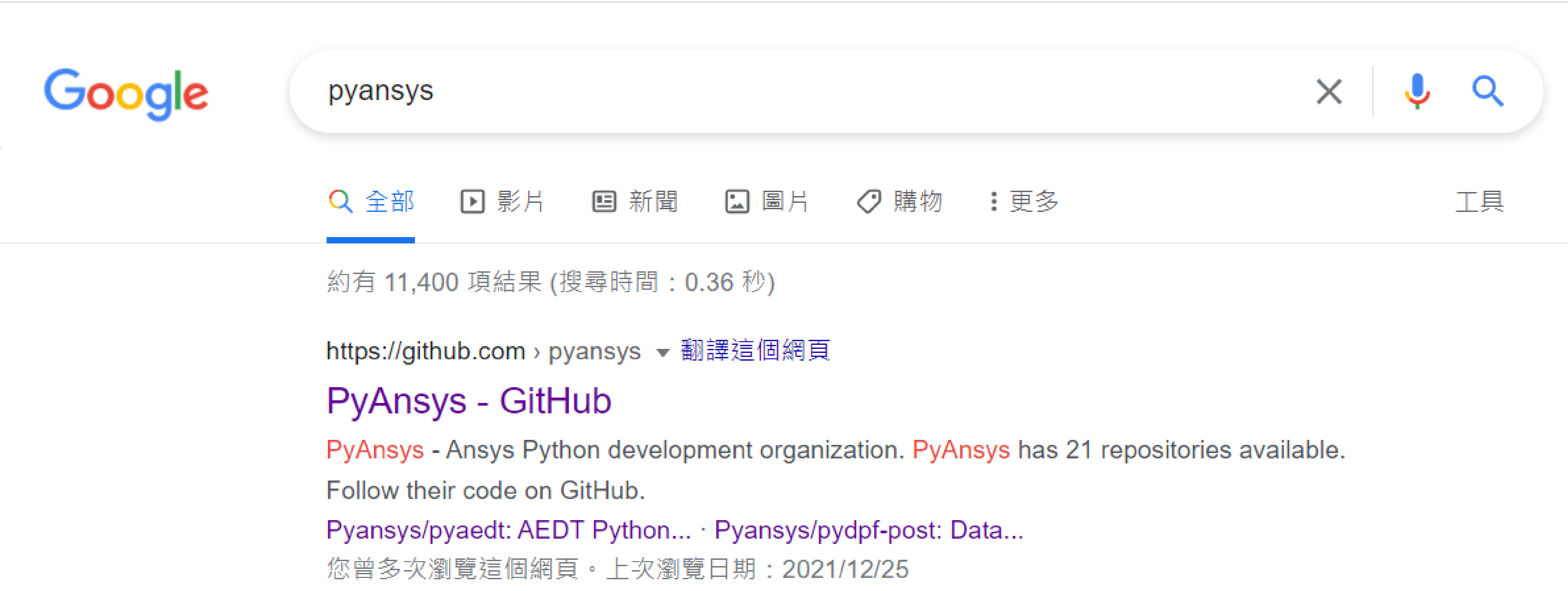
A quick and easy approach for automating a simple operation in the AEDT UI is to record and reuse a scripts. However, disadvantages of this approach are:

- Recorded code is dirty and difficult to read and understand.
- Recorded scripts are difficult to reuse and adapt.
- Complex coding is required by many global users of AEDT.
- No IDE can be used for code development.
- C-python modules can’t be integrated.

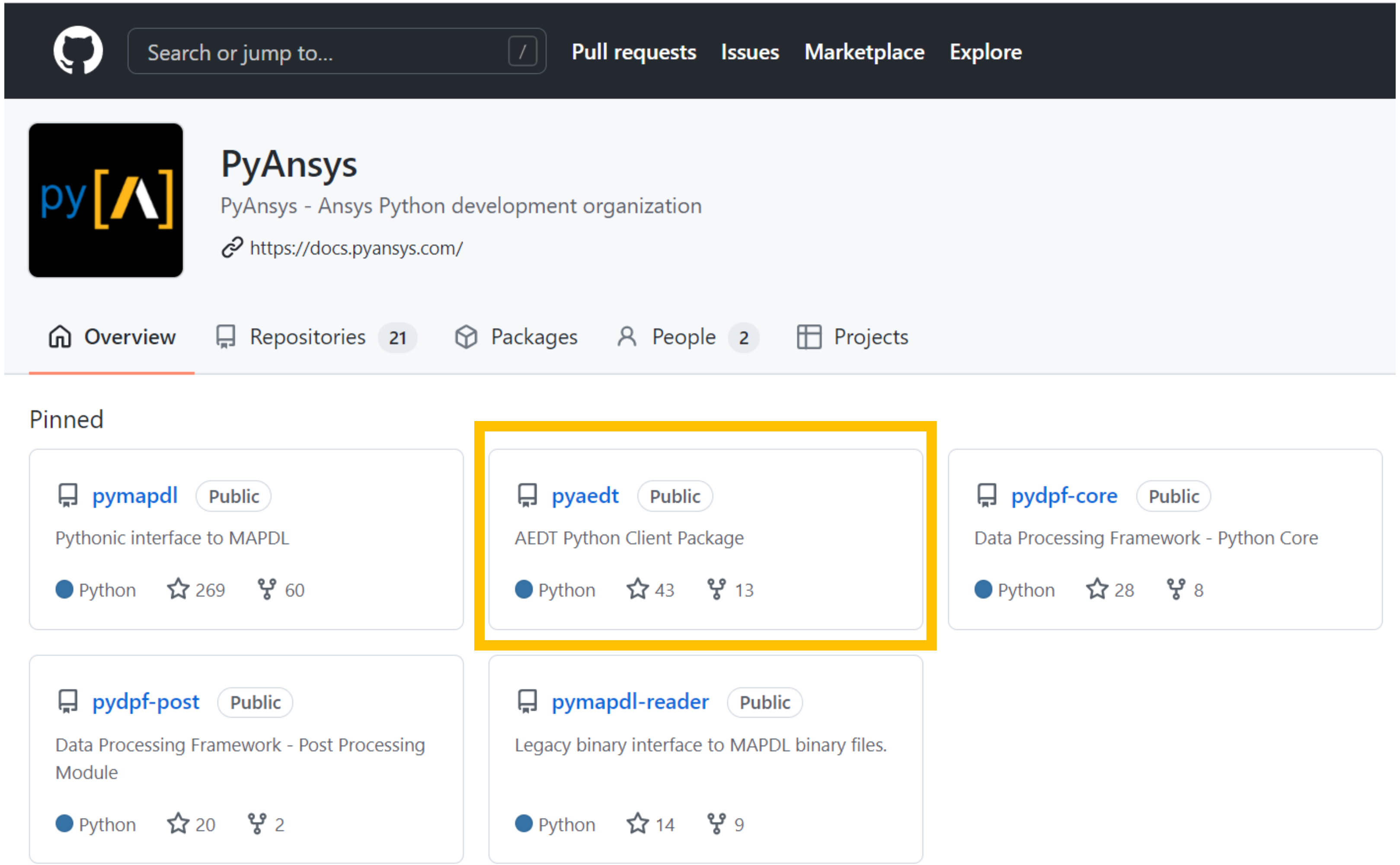
The Main Advantages of PyAEDT API

- Automatic initialization of all AEDT objects, such as desktop objects like the editor, boundaries, and so on
- Error management
- Log management
- Variable management
- Compatibility with IronPython and CPython
- Simplification of complex API syntax using data objects while maintaining PEP8 compliance.
- Code reusability across different solvers
- Clear documentation on functions and API
- Unit tests of code to increase quality across different AEDT versions

Google PyAnsys



Source Code in Github



https://aedtdocs.pyansys.com/

py[Λedt]

Installation User Guide API Documentation Examples Code Guidelines Contributing

Search the docs ...

PyAEDT Documentation 0.4.18

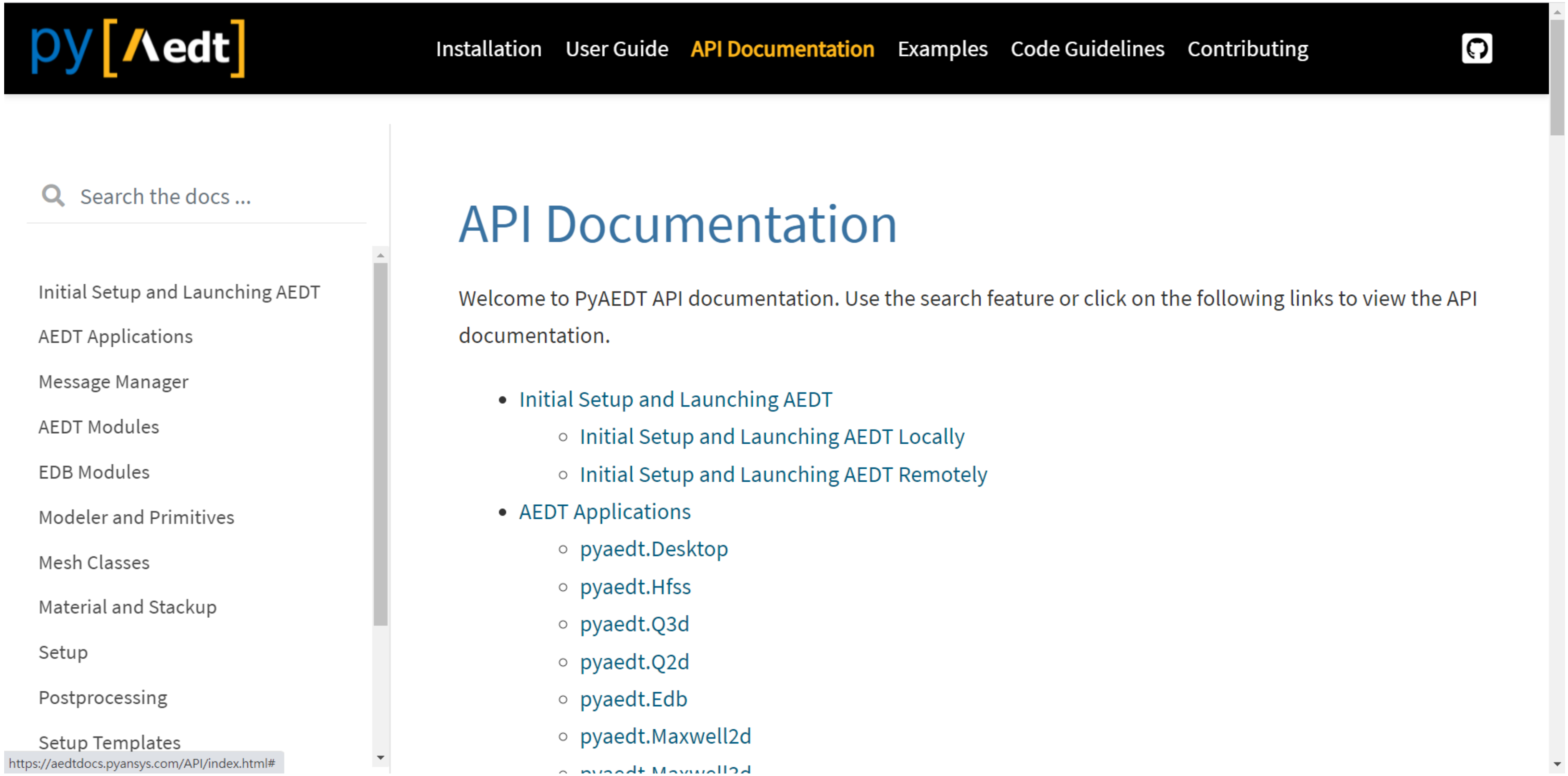
Introduction

PyAEDT is intended to consolidate and extend all existing functionalities around scripting for Ansys Electronics Desktop (AEDT) to allow reuse of existing code, sharing of best practices, and increased collaboration. PyAEDT is licensed under the MIT License.

PyAEDT includes functionality for interacting with the following AEDT tools and Ansys products:

- HFSS and HFSS 3D Layout
- Icepak
- Maxwell 2D/3D and RMxprt
- Q3D/2DExtractor
- Mechanical

API Documentation


Installation User Guide API Documentation Examples Code Guidelines Contributing

Search the docs ...

Initial Setup and Launching AEDT
AEDT Applications
Message Manager
AEDT Modules
EDB Modules
Modeler and Primitives
Mesh Classes
Material and Stackup
Setup
Postprocessing
Setup Templates

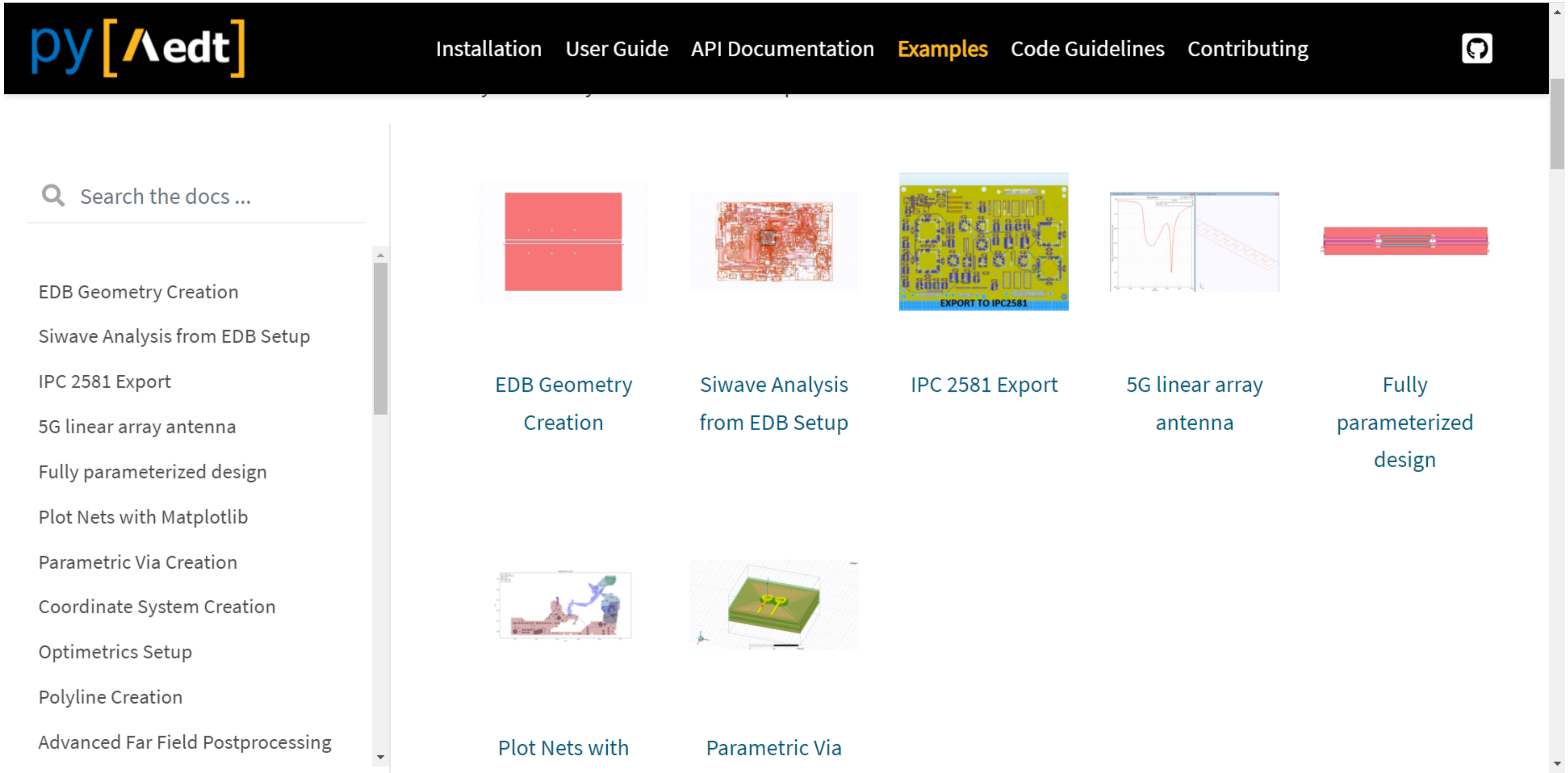
https://aedtdocs.pyansys.com/API/index.html#

API Documentation

Welcome to PyAEDT API documentation. Use the search feature or click on the following links to view the API documentation.

- Initial Setup and Launching AEDT
  - Initial Setup and Launching AEDT Locally
  - Initial Setup and Launching AEDT Remotely
- AEDT Applications
  - pyaedt.Desktop
  - pyaedt.Hfss
  - pyaedt.Q3d
  - pyaedt.Q2d
  - pyaedt.Edb
  - pyaedt.Maxwell2d

API Example


py[Aedt]
Installation
User Guide
API Documentation
Examples
Code Guidelines
Contributing
Search the docs ...
EDB Geometry Creation
Siwave Analysis from EDB Setup
IPC 2581 Export
5G linear array antenna
Fully parameterized design
Plot Nets with Matplotlib
Parametric Via Creation
Coordinate System Creation
Optimetrics Setup
Polyline Creation
Advanced Far Field Postprocessing
EXPORT TO IPC2581
EDB Geometry Creation
Siwave Analysis from EDB Setup
IPC 2581 Export
5G linear array antenna
Fully parameterized design
Plot Nets with
Parametric Via

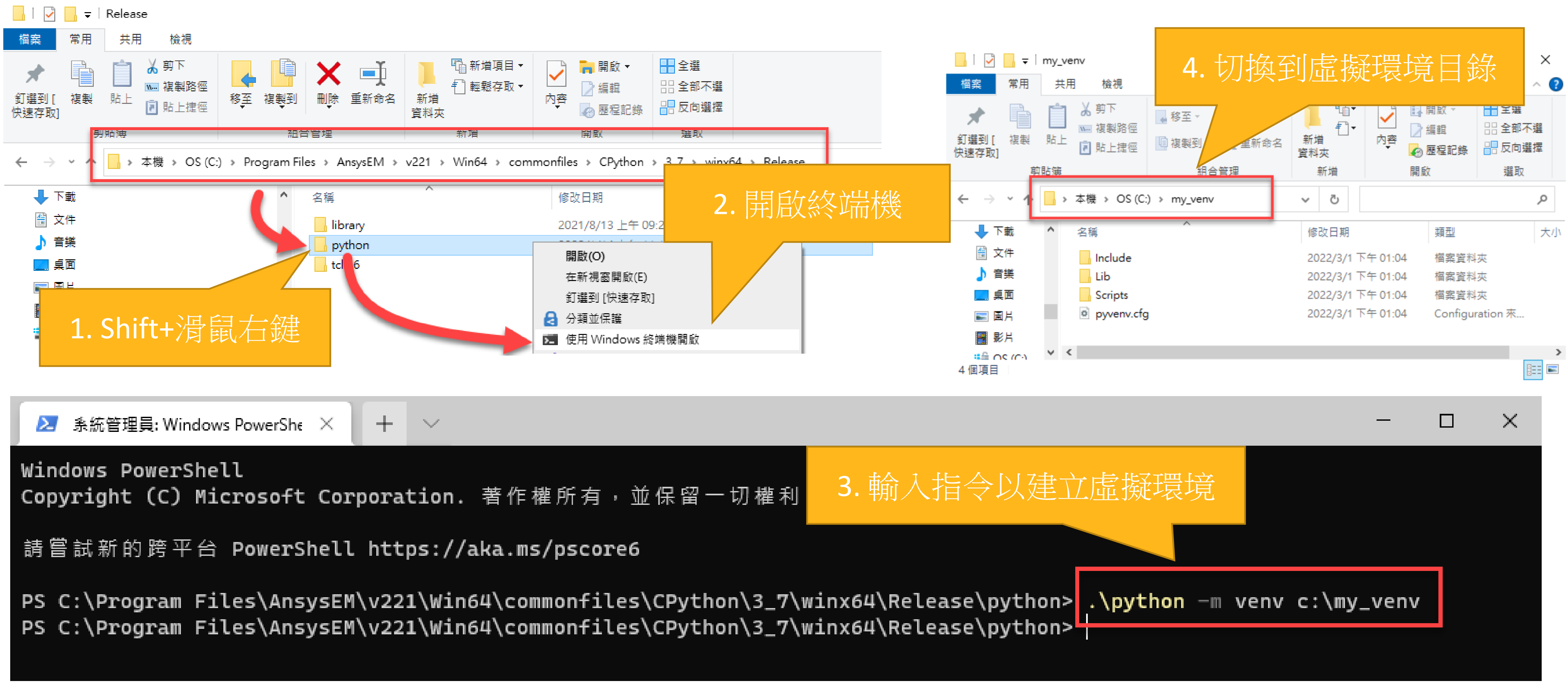
Installation of PyAEDT Module

Python虛擬環境

- Python 應用程式通常會用到不在標準函式庫的套件和模組。應用程式有時候會需要某個特定版本的函式庫，因為這個應用程式可能需要某個特殊的臭蟲修正，或是這個應用程式是根據該函式庫特定版本的介面所撰寫。
- 這意味著不太可能安裝一套 Python 就可以滿足所有應用程式的要求。如果應用程式 A 需要一個特定的模組的 1.0 版，但另外一個應用程式 B 需要 2.0 版，那麼這整個需求不管安裝 1.0 或是 2.0 都會衝突，以致於應用程式無法使用。
- 解決方案是創建一個虛擬環境 (virtual environment)，這是一個獨立的資料夾，並且裡面裝好了特定版本的 Python，以及一系列相關的套件。
- 不同的應用程式可以使用不同的虛擬環境。以前述中需要被解決的例子中，應用程式 A 能夠擁有它自己的虛擬環境，並且是裝好 1.0 版，然而應用程式 B 則可以是用另外一個有 2.0 版的虛擬環境。要是應用程式 B 需要某個函式庫被升級到 3.0 版，這並不會影響到應用程式 A 的環境。

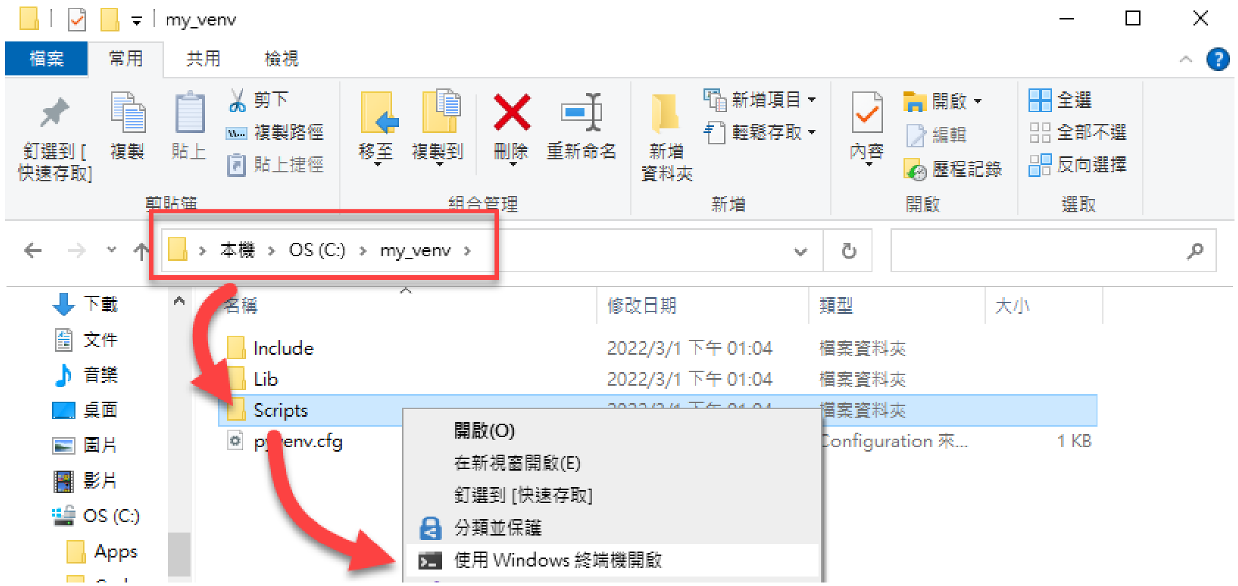
Ansys

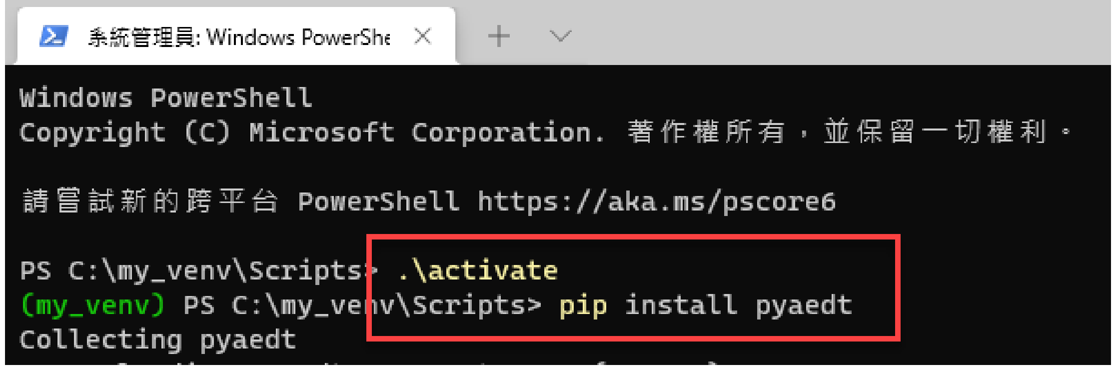
建立Python虛擬環境

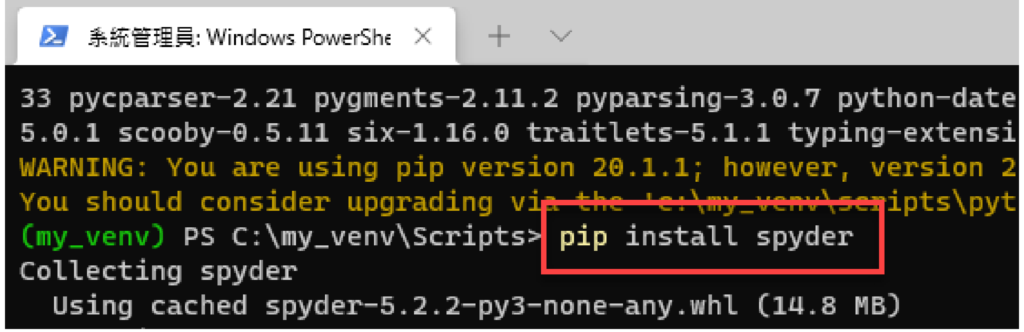


方法1. 透過網路連線安裝

- 選擇Scripts目錄並開啟終端機
- 啟動虛擬環境
  - .\activate
- 安裝pyaedt
  - pip install pyaedt
- 安裝Spyder IDE
  - pip install spyder



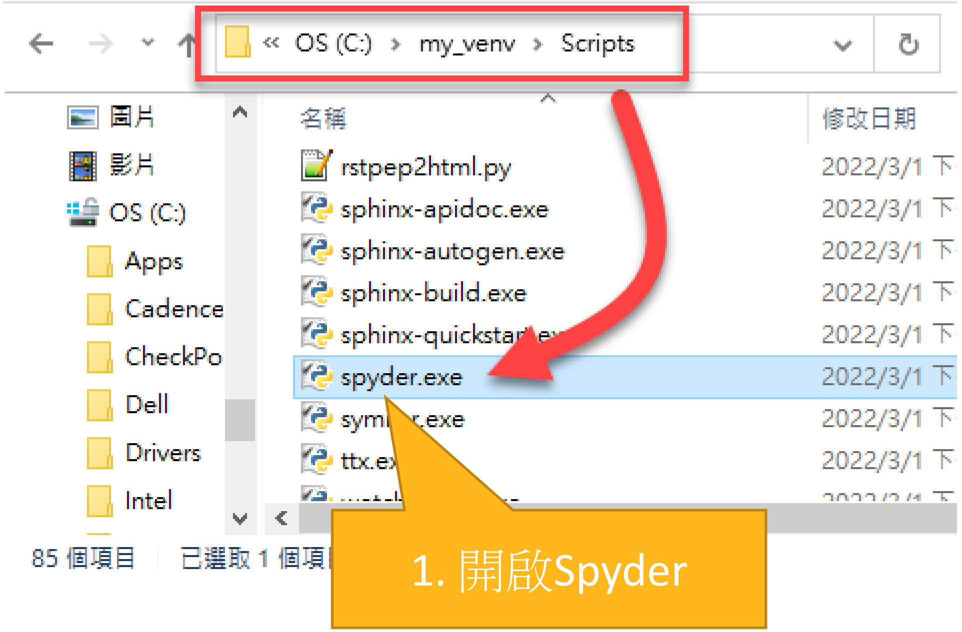


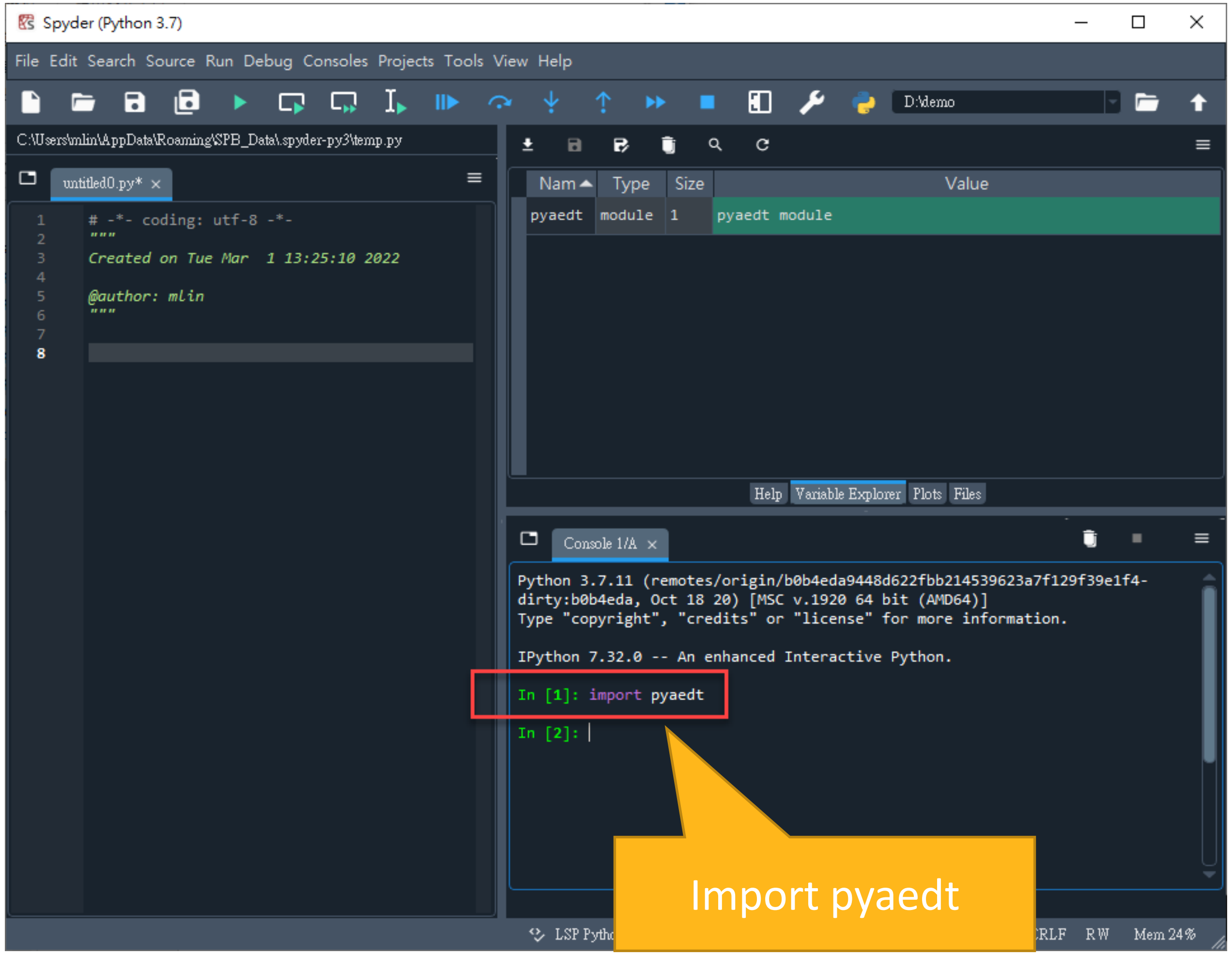


方法2. 離線安裝(洽ANSYS取得安裝包，或自行建立，見附註)

- 將取得的requirements.txt及wheelhouse目錄放到C碟當中
- 開啟終端機
- 啟動虛擬環境
  - .\activate
- pip install -r c:\requirements.txt --no-index --find-links c:\wheelhouse

安裝完成之後，開啟Spyder，測試pyaedt模組安裝



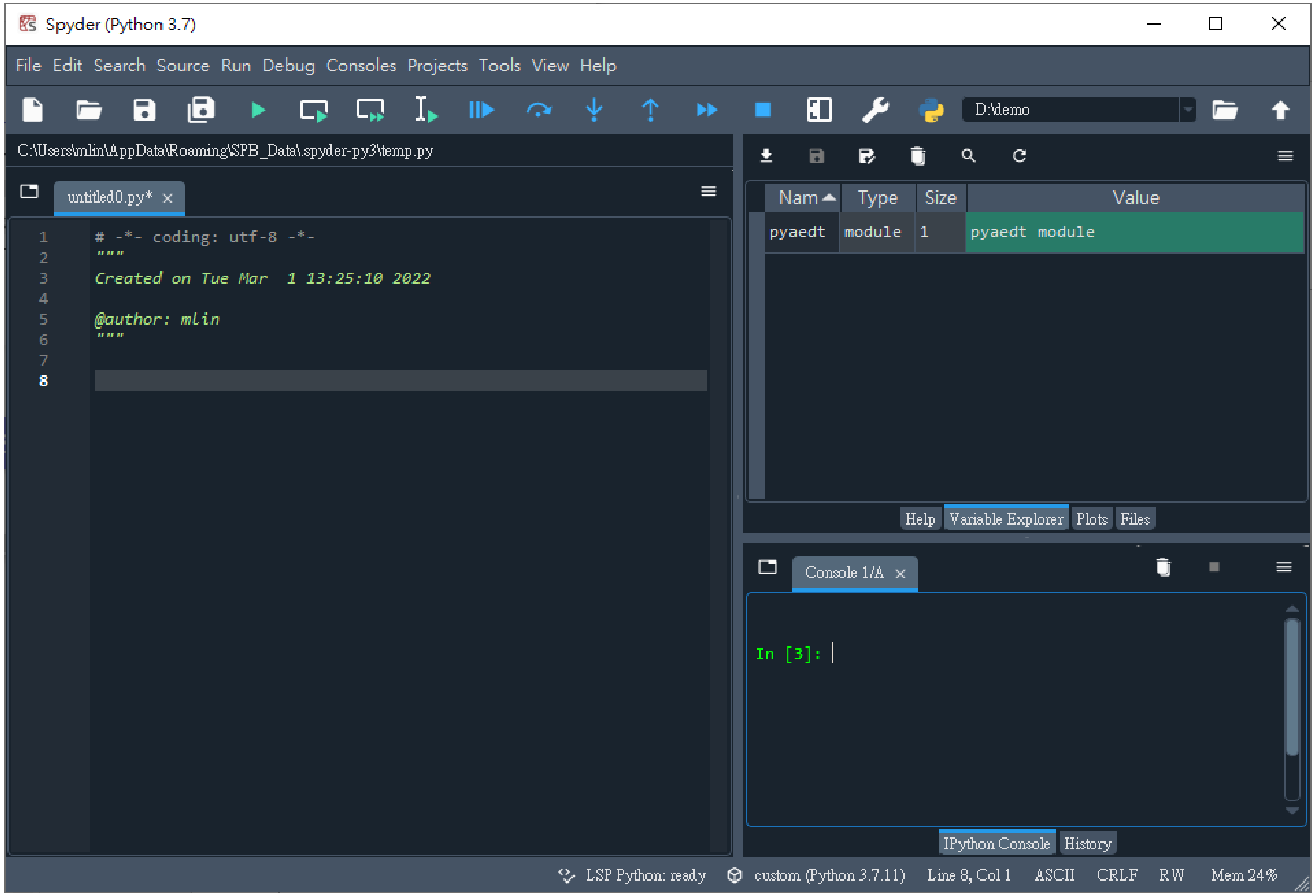


開發模式 v.s. 使用模式

Ansys

開發模式

- 漸進式開發(Spyder)
  - F5 (Run File)
  - Ctrl + F5 (Run Cell)
  - F9 (Run Single Line/Selection)
- Console檢視物件
- Variable Explorer
- Debug Mode
- 檢視物件
  - dir()
  - type()

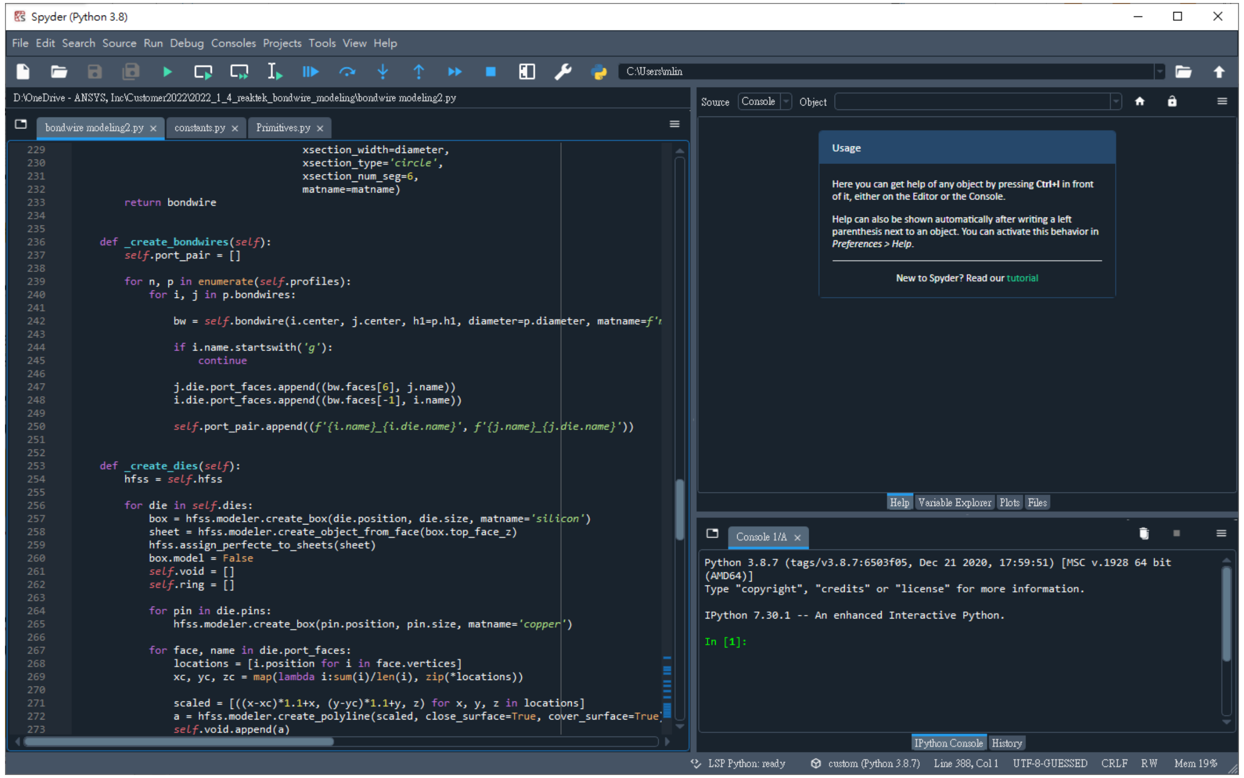


使用模式

- 透過Spyder IDE使用
  - 適合研究
  - 可搭配Numpy, Matplotlib, Scipy…
- 透過Jupyter使用
- 透過AEDT GUI使用
  - 需另外安裝pyaedt模組
  - 使用AEDT當中的圖表作後處理
  - 可加入Windows Form, ，整合到toolkit當中
- 透過Console外部使用
- 透過Windows Form/PyQT5外部使用

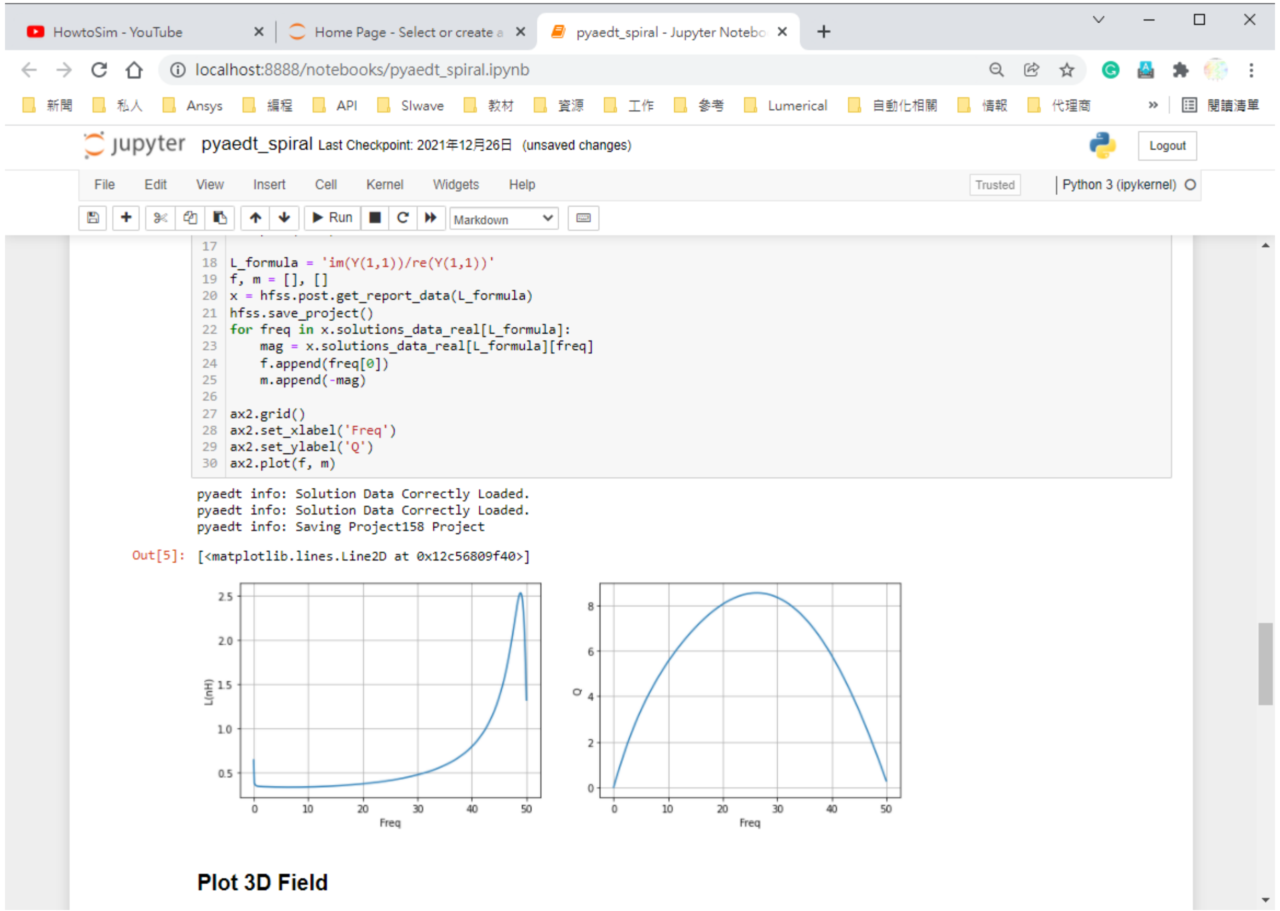
Spyder

- 編輯器：
  - 具有函式和類檢視器
  - 代碼分析特性
  - 代碼補全
  - 直接跳入定義
- 互動視窗：
  - Python或IPython埠都在工作區可以調整和使用。支援對編輯器里的代碼直接除錯。 。
- 文件瀏覽器：
  - 在編輯器或埠中顯示任意類或函式呼叫的文件。
- 變數瀏覽視窗
- Matplotlib的圖表顯示視窗
- 歷史記錄



Jupyter Notebook

- HTML格式混和支援多種資料型態
  - 程式碼
  - 文字，Markdown語法
  - 圖表輸出
- 以Cell為單位
- 可讀性高



PyAEDT Classes

Class v.s Object

- Class
  - Define
    - Attributes/Properties
    - methods
- Object
  - Access
    - Attributes
    - methods

類別關係: Is-a v.s. Has-a

- Is-a
  - Inheritance(繼承)
  - Family
  - Module
  - One-way
- Has-a
  - Composition(組成)
  - Team
  - Application
  - Two-way

Design Operation

Validate and Solve

- validate_simple
  - Hfss.validate_simple(*logfile=None*)
- validate_full_design
  - Hfss.validate_full_design(*dname=None*, *outputdir=None*, *ports=None*)
- solve_in_batch
  - Hfss.solve_in_batch(*filename=None*, *machine='local'*, *run_in_thread=False*)
- submit_job
  - Hfss.submit_job(*clustername*, *aedt_full_exe_path=None*, *numnodes=1*, *numcores=32*, *wait_for_license=True*, *setting_file=None*)
- analyze_nominal
  - Hfss.analyze_nominal(*num_cores=None*, *num_tasks=None*, *num_gpu=None*, *acf_file=None*)

初始化與退出

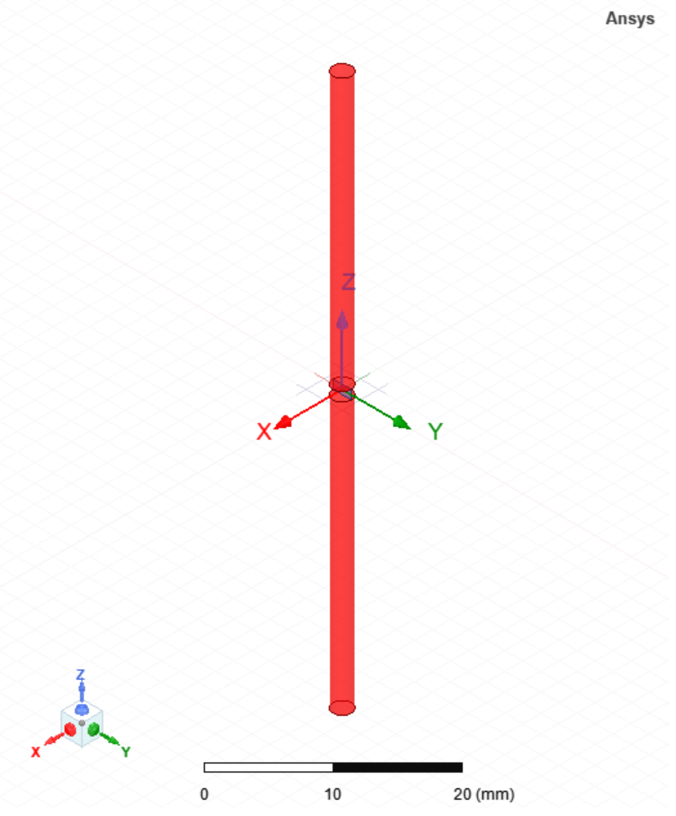
```
from pyaedt import Hfss, constants
hfss = Hfss(specified_version='2021.2’)
#%%
……………………….

#%%
hfss.save_project()
hfss.release_desktop()
```

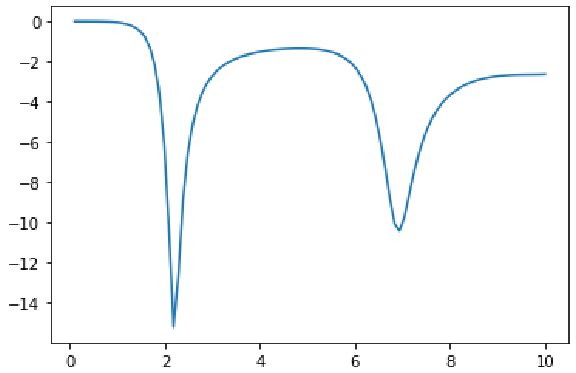
模擬流程常用函式

- 屬性
  - hfss['length'] = '3mm'
- 材料
  - hfss.materials.add_material()
- 物件
  - hfss.modeler.create_box()
- 激發源
  - hfss.create_lumped_port_to_sheet()
- 邊界條件
  - hfss.create_open_region()
- 網格
  - hfss.mesh.assign_initial_mesh_from_slider()
- 模擬設定
  - hfss.create_setup()
  - hfss.create_linear_count_sweep()
  - hfss.analyze_all()
- 生成報告
  - hfss.create_lumped_port_to_sheet()
- 輸出資料
  - hfss.post.get_report_data()
- 常數
  - constants.AXIS.X (Y, Z)
  - constants.PLANE.XY, (YZ, ZX)

Example Code

```python
from pyaedt import Hfss, constants
import matplotlib.pyplot as plt

hfss = Hfss(specified_version='2021.2', non_graphical=True)

c1 = hfss.modeler.create_cylinder(constants.AXIS.Z, (0,0,0.5), 1, 20, matname='copper')
c2 = hfss.modeler.create_cylinder(constants.AXIS.Z, (0,0,-0.5), 1, -20, matname='copper')

hfss.modeler.create_rectangle(constants.PLANE.YZ, (0, -1, -0.5), (2,1), 's1')
hfss.create_lumped_port_to_sheet('s1', axisdir=constants.AXIS.Z)
hfss.create_open_region(Frequency='1GHz')

setup = hfss.create_setup(setupname='my_setup')
hfss.create_frequency_sweep('my_setup', freqstart=0.1, freqstop=10, num_of_freq_points=101,sweepname='my_sweep')
hfss.analyze_nominal(num_cores=4)
hfss.post.create_rectangular_plot(setup_sweep_name='my_setup : my_sweep')

data = hfss.post.get_report_data('dB(S(1,1))', 'my_setup : my_sweep')

x = data.sweeps['Freq']
y = data.data_real()

plt.plot(x, y)
hfss.release_desktop()
```





DDR Simulation

```
from pyaedt import Hfss3dLayout, Edb

edb = Edb()
edb.import_cadence_file("d:/demo/Galileo_G87173_204.brd")
edb.core_hfss.create_coax_port_on_component(['U1B5', 'U2A5'], ['M_DQ<0>', 'M_DQ<1>'])

edb.core_components.set_solder_ball('U1B5')
edb.core_components.set_solder_ball('U2A5')

edb.create_cutout(['M_DQ<0>', 'M_DQ<1>'],
                  ['GND'],
                  output_aedb_path='d:/demo/ddr.aedb',
                  open_cutout_at_end=False)
edb.close_edb()

h3d = Hfss3dLayout('d:/demo/ddr.aedb/edb.def')
setup1 = h3d.create_setup('setup1')
setup1.props['Frequency'] = '1GHz'
h3d.create_frequency_sweep('setup1', 'GHz', 0, 1, 11)
setup1.update()

h3d.analyze_all()
h3d.save_project()
```

附註：如何包裹pyaedt套件以提供離線安裝

- 透過有網路的電腦安裝好虛擬環境及相關套件之後，可以將其輸出到
  - requirements.txt
  - wheelhouse
- 將兩者複製到離網的電腦，使用離線安裝

```
系統管理員: Windows PowerShe
Windows PowerShell
Copyright (C) Microsoft Corporation. 著作權所有，並保留一切權利。

請嘗試新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\my_venv\Scripts> .\activate
(my_venv) PS C:\my_venv\Scripts> pip freeze c:\requirements.txt
aiohttp==3.8.1
aiosignal==1.2.0
```

```
系統管理員: Windows PowerShe
(my_venv) PS C:\my_venv\Scripts> mkdir c:\wheelhouse


    目錄: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2022/3/1   下午 02:00                wheelhouse


(my_venv) PS C:\my_venv\Scripts> pip download -r c:\requirements.txt -d c:\wheelhouse
```

Ansys