Preparation and Installation Instructions

Francisco Periago

Universidad Politécnica de Cartagena
https://multisimo.com/fpe/
https://github.com/fperiago/pinn_deeponet_for_beginners

XXI Jacques-Louis Lions Hispano-French School on Numerical Simulation in Physics and Engineering

Ciudad Real, july, 7-11, 2025

A straightforward installation

There are several possibilities to install what we need in this course. First, we present a simple way. A more professional installation that makes use of Visual Studio Code and Jupyter Notebook is described later.

Follow these instructions:

1. Download and install `miniconda` (select default options)
   https://www.anaconda.com/docs/getting-started/miniconda/main
2. Look for Anaconda Prompt in your computer and open `Anaconda Prompt`
3. In the terminal that is open run the following command:

```
conda create -n deep python numpy scipy==1.13 pandas scikit-learn matplotlib ipykernel autopep8 jupyter
```

4. Execute the following commands in the terminal:
   - `python -m pip install --upgrade pip`
   - `conda activate deep`
   - `pip install tensorflow`
   - `pip install deepxde`
   - `python -m ipykernel install --name=deep`
5. To run a Jupyter file (`name_file.ipynb`) you should write in your terminal `jupyter notebook` and then open your file. It is absolutely mandatory to select the kernel `deep` that we have just created.

For a new use, you just have to open `Anaconda Prompt` and execute `conda activate deep`. Then, go to step 5 above.

Installation using Visual Studio Code and Jupyter Notebook

Installation using Visual Studio Code and Jupyter Notebook

1. I strongly recommend install **Python** from **Anaconda**

   `https://www.anaconda.com/products/individual`

   Just click **Skip registration, Download**

Installation using Visual Studio Code and Jupyter Notebook

1. I strongly recommend install **Python** from **Anaconda**

   `https://www.anaconda.com/products/individual`

   Just click **Skip registration, Download**

2. We shall use **Visual Studio Code.** You can download it by clicking on the link `https://code.visualstudio.com/docs/?dv=win`

Installation using Visual Studio Code and Jupyter Notebook

1. I strongly recommend install **Python** from **Anaconda**

   `https://www.anaconda.com/products/individual`

   Just click **Skip registration, Download**

2. We shall use **Visual Studio Code.** You can download it by clicking on the link `https://code.visualstudio.com/docs/?dv=win`

3. We should link Visual Studio code with Python and Jupyter Notebook. You can do it from the left pannel of Visual Studio Code. Go to Extensions and select

   "Jupyter" notebook support, interactive and computing ?. Microsoft

   "Python" IntelliSense Pylance, linting, debugging?. Microsoft

To put it in a nutshell

Instalación, descarga y extensiones de Python

1.- Descarga de software:

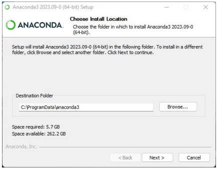
https://www.anaconda.com/products/individual (Pulsar en Skip registration, Download)

https://code.visualstudio.com/docs/?dv=win (se descarga directamente al pulsar el enlace)

2.- Instalación de Python:

1.- Primero se instala Anaconda individual edition

2.- Segundo instalar Visual Studio Code



Anaconda3 2023.09-0 (64-bit) Setup

Choose Install Location

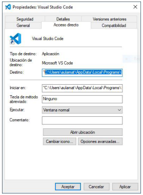
Choose the folder in which to install Anaconda3 2023.09-0 (64-bit).

Setup will install Anaconda3 2023.09-0 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\ProgramData\anaconda3

Space required: 5.7 GB
Space available: 262.2 GB

Anaconda, Inc.



Propiedades: Visual Studio Code

Visual Studio Code

Tipo de destino: Aplicación
Ubicación de destino: Microsoft VS Code
Destino: "C:\Users\aulamat\AppData\Local\Programs\
Iniciar en: "C:\Users\aulamat\AppData\Local\Programs\
Tecla de método abreviado: Ninguno
Ejecutar: Ventana normal
Comentario:

3.- Instalar extensiones:

Se hace desde Visual Studio Code / Extensiones:

- "Jupyter" notebook support, interactive and computing …. Microsoft.
- "Python" IntelliSense Pylance, linting, debugging…. Microsoft

Important: Add Anaconda to the system environment variables

If you are using Windows you must add Anaconda to the environment variables. Just follow these steps:

Añadir Anaconda a las variables de entorno

Para que el "Terminal" de Visual Studio funcione y esté activo hay que añadir antes la ruta de anaconda3 en las variables de entorno de Windows, ir a:

Acerca de / Configuración avanzada del sistema / Propiedades del Sistema / Pestaña opciones avanzadas "**variables de entorno" / "variables del Sistema"**

seleccionar "**Path**" pulsar en editar, nuevo y copiar ruta de instalación de anaconda: **C:\Users\Dmae\anaconda3** pulsar aceptar.

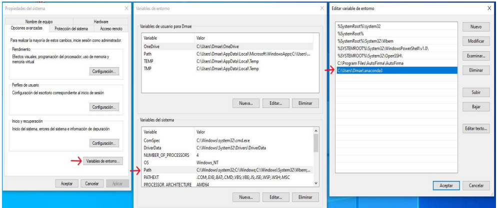
Reiniciar el ordenador.

Para probar: Desde VS Code se accede a New Terminal

Para comprobar si tenemos instalado conda escribimos en el nuevo terminal conda --version

si funciona nos dará la versión, ejemplo: conda 24.9.2

Con el comando **cmd** nos vamos a un terminal base.



Para realizar la mayoría de estos cambios, inicie sesión como administrador.

Rendimiento

Efectos visuales, programación del procesador, uso de memoria y memoria virtual

Perfiles de usuario

Configuración del escritorio correspondiente al inicio de sesión

Inicio y recuperación

Inicio del sistema, errores del sistema e información de depuración

Variables de usuario para Dmae

Variable	Valor
OneDrive	C:\Users\Dmae\OneDrive
Path	C:\Users\Dmae\AppData\Local\Microsoft\WindowsApps;C:\Users\...
TEMP	C:\Users\Dmae\AppData\Local\Temp
TMP	C:\Users\Dmae\AppData\Local\Temp

Variables del sistema

Variable	Valor
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

```
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\AutoFirma\AutoFirma
C:\Users\Dmae\anaconda3
```

Creation of a virtual environment

A **virtual environment** is an isolated workspace on your computer where you can install and manage software packages independently of the system-wide Python installation or other projects.

Creation of a virtual environment

A **virtual environment** is an isolated workspace on your computer where you can install and manage software packages independently of the system-wide Python installation or other projects.
There are different ways of creating a virtual environment:

Creation of a virtual environment

A **virtual environment** is an isolated workspace on your computer where you can install and manage software packages independently of the system-wide Python installation or other projects.

There are different ways of creating a virtual environment:

- By using the file deeponetcontrol.yml that is provided. Then, you should:
    1. Open a terminal (or Anaconda Prompt if you are in Windows and uses Anaconda). Then, run these instructions in the terminal:
    2. `conda env create -f deeponetcontrol.yml`
    3. `conda activate deeponetcontrol`

Creation of a virtual environment

A **virtual environment** is an isolated workspace on your computer where you can install and manage software packages independently of the system-wide Python installation or other projects.

There are different ways of creating a virtual environment:

- By using the file deeponetcontrol.yml that is provided. Then, you should:
  1. Open a terminal (or Anaconda Prompt if you are in Windows and uses Anaconda). Then, run these instructions in the terminal:
  2. `conda env create -f deeponetcontrol.yml`
  3. `conda activate deeponetcontrol`
- Alternatively, from Visual Studio code go to a New Terminal and write:
  1. `conda create -n deeponetcontrol python=3.9.12`
  2. `conda activate deeponetcontrol`
  3. Install the following packages: `numpy`, `matplotlib`, `pandas`, `scikit-learn`, `autopep8`, `ipykernel`, `deepxde`, `tensorflow`, `scikit-optimize` . Precisely, for each one of the packages write in the terminal:
  4. `python -m pip install name_of_package` or just `pip install name_of_package`

Creation of a virtual environment

A **virtual environment** is an isolated workspace on your computer where you can install and manage software packages independently of the system-wide Python installation or other projects.
There are different ways of creating a virtual environment:

- By using the file deeponetcontrol.yml that is provided. Then, you should:
    1. Open a terminal (or Anaconda Prompt if you are in Windows and uses Anaconda). Then, run these instructions in the terminal:
    2. `conda env create -f deeponetcontrol.yml`
    3. `conda activate deeponetcontrol`
- Alternatively, from Visual Studio code go to a New Terminal and write:
    1. `conda create -n deeponetcontrol python=3.9.12`
    2. `conda activate deeponetcontrol`
    3. Install the following packages: `numpy`, `matplotlib`, `pandas`, `scikit-learn`, `autopep8`, `ipykernel`, `deepxde`, `tensorflow`, `scikit-optimize` . Precisely, for each one of the packages write in the terminal:
    4. `python -m pip install name_of_package` or just `pip install name_of_package`

  If you still have problems, then you may follow the instructions in `https://github.com/lululxvi/deepxde`

Creation of a workspace

A **workspace** is a collection of files, environments, and configurations that you use to work on a specific project. Let's create our own workspace.

Creation of a workspace

A **workspace** is a collection of files, environments, and configurations that you use to work on a specific project. Let's create our own workspace.
At some place in your computer or in the cloud, create the folder

`pinn_course`

Inside this folder, create the subfolders:

- `data`
- `figures`
- `src`

Creation of a workspace

A **workspace** is a collection of files, environments, and configurations that you use to work on a specific project. Let's create our own workspace.
At some place in your computer or in the cloud, create the folder

`pinn_course`

Inside this folder, create the subfolders:

- `data`
- `figures`
- `src`

From Visual Studio Code (`File`) open `pinn_course`

Next, we associate the virtual environment `deeponetcontrol` to our workspace. First, from the wheel in the left-button corner of Visual Studio Code, select `Command Palette` . Then, `Python:  interpreter` and click on `deeponetcontrol`.

Creation of a workspace

A **workspace** is a collection of files, environments, and configurations that you use to work on a specific project. Let's create our own workspace.
At some place in your computer or in the cloud, create the folder

`pinn_course`

Inside this folder, create the subfolders:

- `data`
- `figures`
- `src`

From Visual Studio Code (`File`) open `pinn_course`

Next, we associate the virtual environment `deeponetcontrol` to our workspace. First, from the wheel in the left-button corner of Visual Studio Code, select `Command Palette` . Then, `Python: interpreter` and click on `deeponetcontrol`.

To conclude, go to `File` and select `Save Workspace As` and save it.