Installation Manual

The Fifth Winter School of Computational Chemistry
Sharif University of Technology, Tehran, Iran

February 2025

Contents

1 Introduction

Welcome to the **Installation Manual for the Fifth Winter School of Computational Chemistry (WSCC-2025)** at **Sharif University of Technology**. This guide is designed to help you set up the necessary software tools for a smooth and productive experience during the school.
Throughout WSCC-2025, we will be working with a range of computational chemistry software, programming environments, and parallelization tools. This document provides step-by-step instructions on how to download, install, and verify the installation of the following essential programs:

- **VSCode** – A lightweight yet powerful code editor
- **ORCA** – A powerful quantum chemistry package
- **Parallelization Packages** – Tools to optimize performance on multi-core systems
- **Python** – The scripting language essential for automation and data analysis
- **Avogadro** – A molecular visualization and editing tool

By carefully following the instructions in this manual, you will ensure that your computational setup is properly configured for hands-on exercises and tutorials.
We hope you find this guide helpful and that you enjoy the Fifth Winter School of Computational Chemistry. If you have any questions or encounter any issues during the installation process, feel free to reach out to us in the **WSCC-2025** Telegram group
Tips:

- To open terminal in Linux you can use Ctrl+Alt+T shortcur. For Windows use Window+R command and type `cmd` or just search for **cmd**.

2 System Requirements

For the successful installation of softwares used in the Winter School, the system should meet the following minimum requirements: at least 8 GB of RAM for optimal performance, though 16 GB is recommended for handling larger systems. A modern operating system, such as Windows 10/11, macOS 10.14 or later, or a Linux distribution (Ubuntu 20.04 or higher), is necessary. Ensure that sufficient disk space is available , with at least 20 GB free for installation and dependencies. Additionally, a stable internet connection is required for downloading software packages and updates.

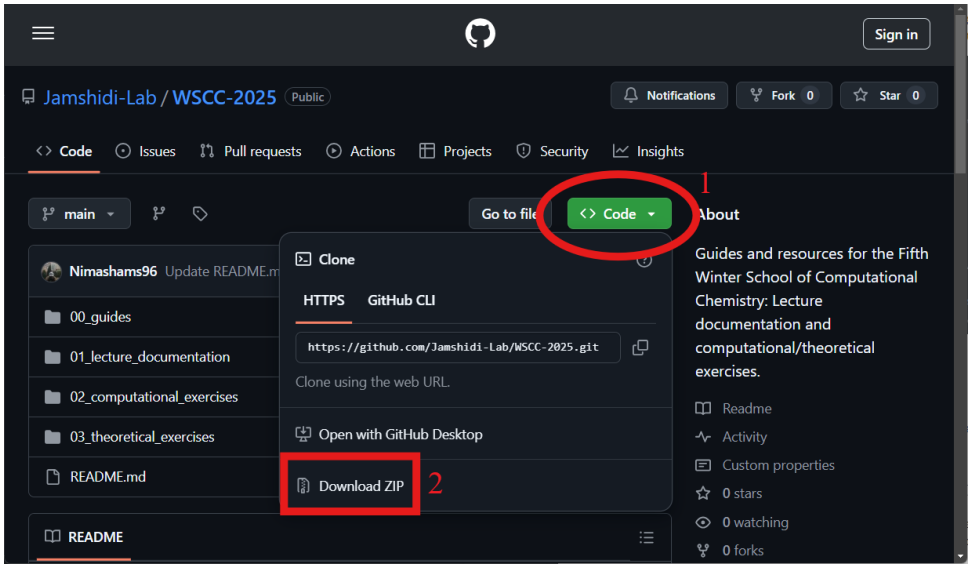
3 Github Repository

To access all materials for the Winter School, we have created a GitHub repository. You can download the materials in two ways:

1. **Using Git Bash**: Run the following command:

```
git clone https://github.com/Jamshidi-Lab/WSCC-2025
```

2. **Manually**: Visit github.com/Jamshidi-Lab/WSCC-2025, and download the ZIP file by following the instructions in the image below:



Important Note: The Github repository updates during the winter school, so will need to update it based on the announcements you recieve from us.

4 VSCode

Visual Studio Code (VSCode) is a powerful and versatile integrated development environment (IDE) that supports multiple programming languages, including Python, C++, and even LaTeX for document preparation. It is widely used for tasks ranging from software development to data analysis, making it an excellent choice for computational chemists. VSCode is free, open-source, and backed by a large community, offering extensive extensions and tools to enhance productivity. Its flexibility, lightweight design, and rich ecosystem make it a go-to editor for scientific computing and programming.

4.1 Downloading the Software

To download VSCode, visit the official download page: code.visualstudio.com/download Select the appropriate version for your operating system and proceed with the download.

4.2 Installating the Software

4.2.1 Linux

To install VSCode on Linux, specifically on Debian-based distributions, follow these steps:

1. Open a terminal.
2. Run the following command to install the VSCode `.deb` package:

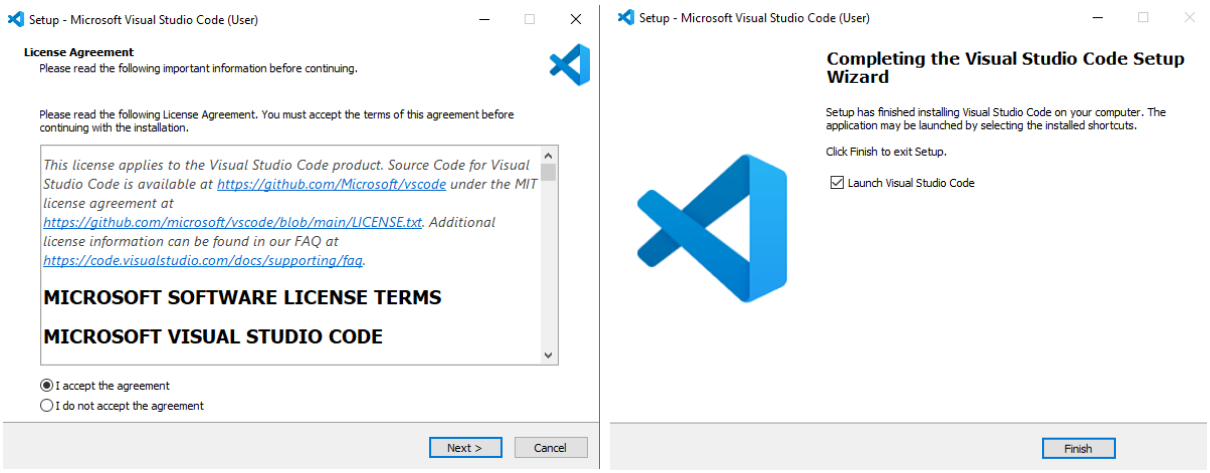
```
sudo dpkg -i <full path of the ".deb" file,
    e.g.: /home/ubuntu/Downloads/code_1.97.deb >
```

3. If you are using Ubuntu 20.04 or later, the installation should proceed without issues.
4. For older Ubuntu versions, you may need to install an older VSCode version. Install VSCode using Snap:

```
sudo snap install --classic code
```

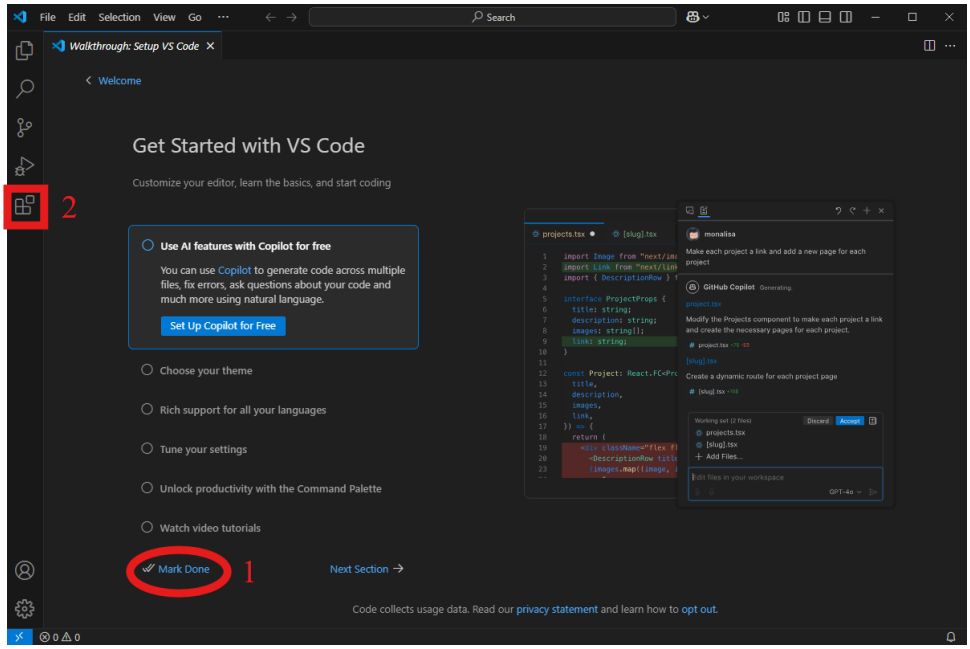
4.2.2 Windows

To install VSCode on Windows, run the setup file and follow the standard installation procedure.

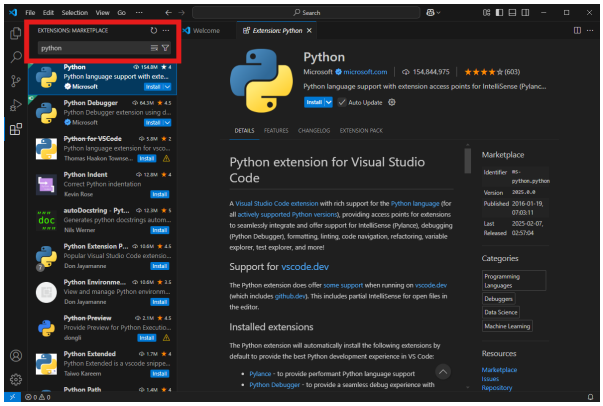


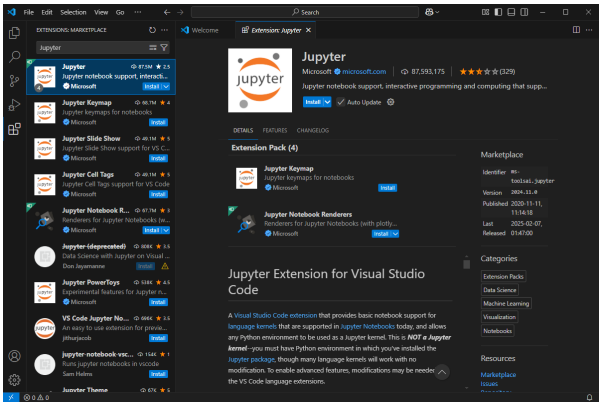
4.3 Installating the Extensions

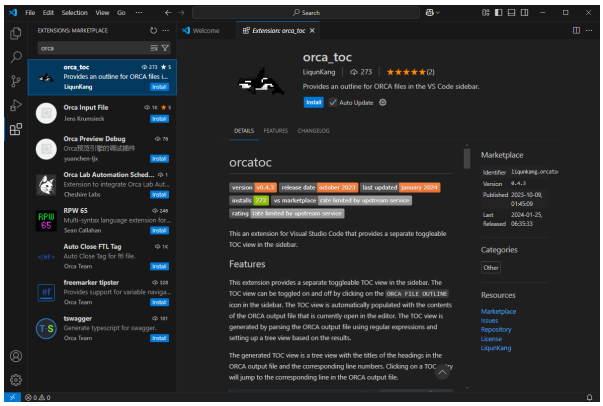
After installing the VSCode successfully open the software, and follow the setps like the picture below to open "Extensions" store.

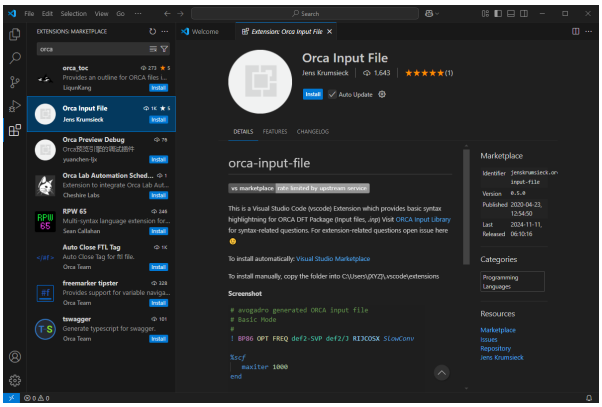


In the search box you can search for your desired extensions, for the sake of this winter school you need to search and install these four extensions: "Python", "Jupyter", "ORCA_toc", and "ORCA Input File".


Python
Python extension for Visual Studio Code
Installed extensions


Jupyter
Jupyter Extension for Visual Studio Code


orca_toc
orcatoc
Features

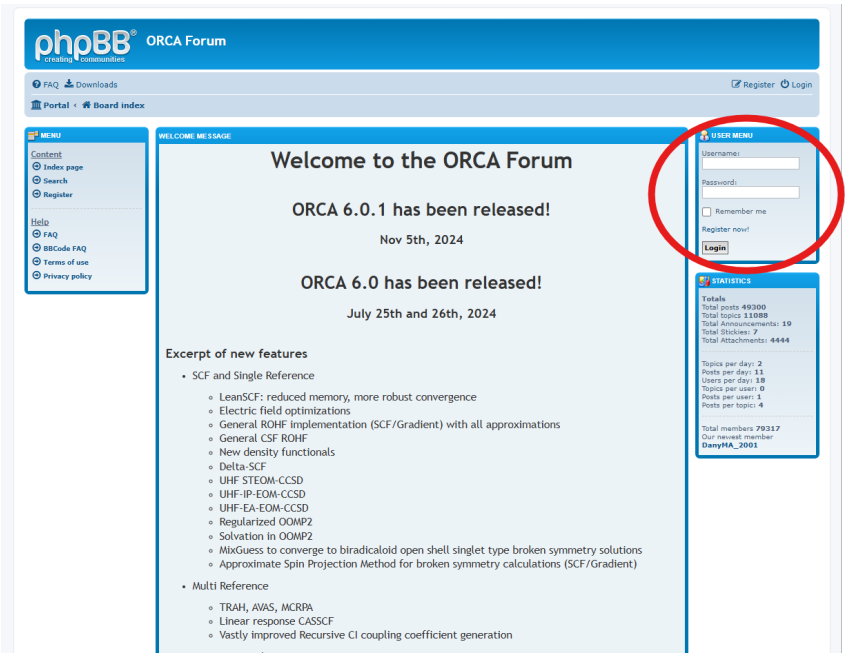

Orca Input File
orca-input-file

5 ORCA

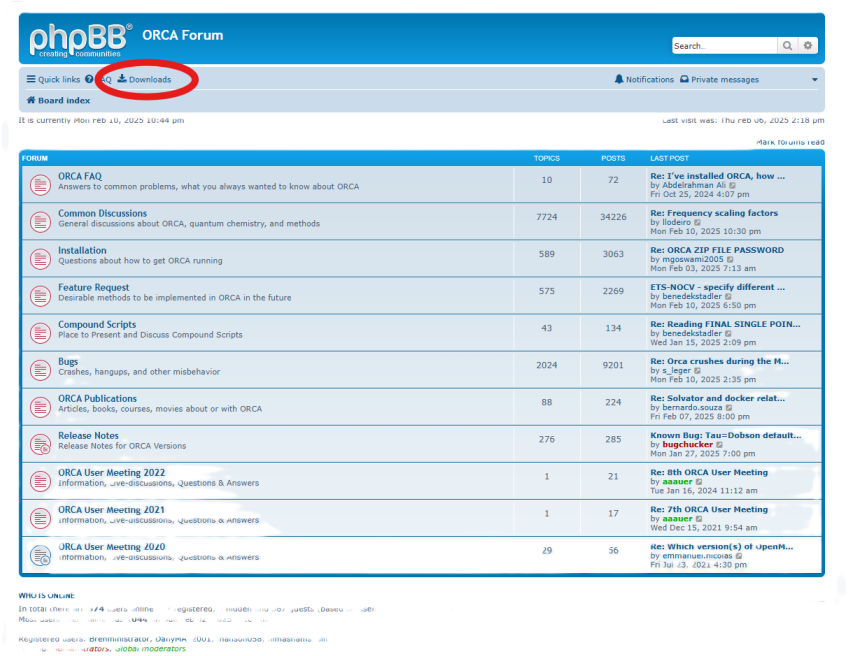
ORCA is an ab initio quantum chemistry program package developed by Prof. Frank Neese and his research group, offering a wide range of modern electronic structure methods, including density functional theory, many-body perturbation, coupled cluster, and semi-empirical quantum chemistry methods. It is designed for the study of larger molecules, transition metal complexes, and their spectroscopic properties, making it a valuable tool for computational chemists, as well as chemists, physicists, and biologists interested in the full information content of their systems. The program is available for various operating systems, including Linux, Microsoft Windows, and macOS, and the free version is accessible for academic use at academic institutions. ORCA is known for its user-friendly nature and is considered to be an efficient and flexible tool for quantum chemistry simulations, with a specific emphasis on the spectroscopic properties of open-shell systems.

5.1 Downloading the Software

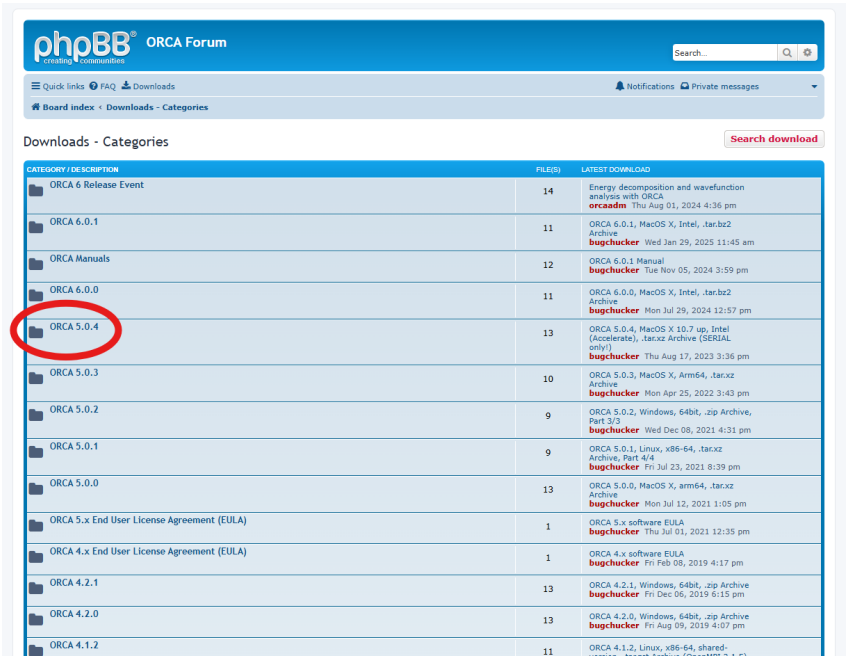
To download ORCA, follow these steps:
1. Visit the ORCA forum webpage: orcaforum.kofo.mpg.de.
2. Use the panel on the right side to log in to your account. If you have not registered before, create an account and then sign in.



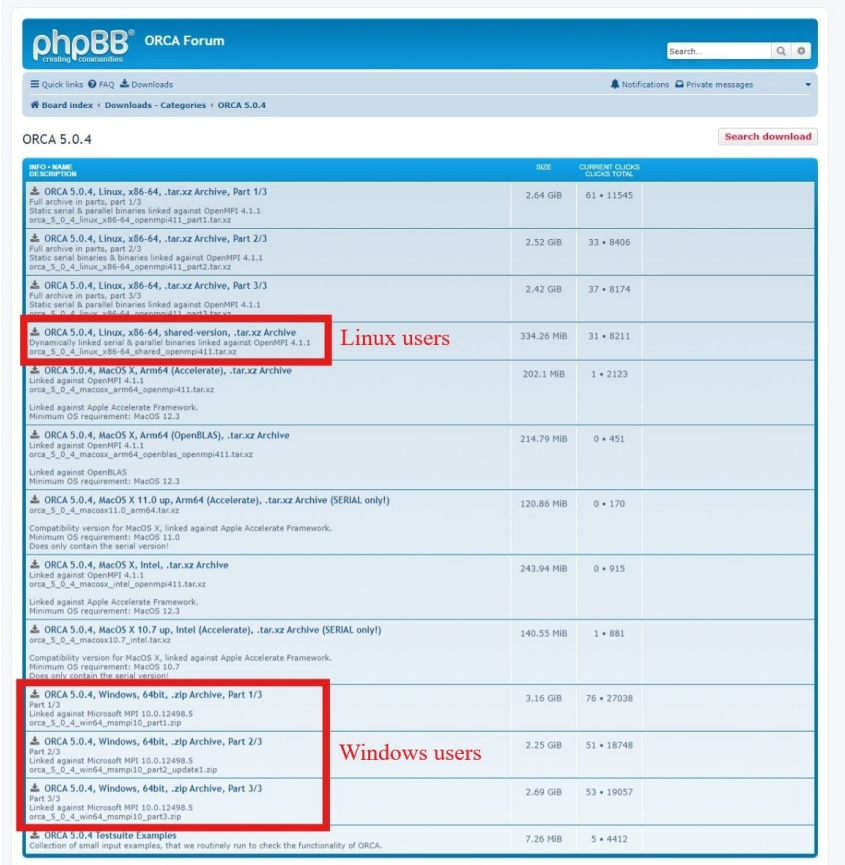
3. Navigate to the “Downloads” section.



4. Select ORCA version 5.0.4.



5. Download the appropriate version for your operating system.



5.2 Installating the Software

ORCA is a compiled program and does not require installation. Simply extract the downloaded archive files into a directory and set up the environment path for the program. Follow the instructions below to install ORCA on your system.

5.2.1 Linux

In order to install ORCA version 5.0.4 on a Linux machine, open a terminal and execute the following commands. To install the program, you first need to define a directory for it. Here, we set our directory to `$HOME/Programs`:

```
mkdir -p $HOME/Programs
```

Next, extract the files into the defined directory using the following command. We assume the downloaded file from orcaforum.kofo.mpg.de is located in the `$HOME/Downloads` directory:

```
sudo tar \
  -xvf $HOME/Downloads/orca_5_0_4_linux_x86-64_shared_openmpi411.tar.xz \
  -C $HOME/Programs
```

If you have downloaded the multi-part version of ORCA 5.0.4 for Linux, use the following command instead:

```
cat $HOME/Downloads/orca_5_0_4_linux_x86-64_openmpi411_part*.tar.xz | \
  tar -xf -C $HOME/Programs
```

Then, rename the extracted directory for convenience:

```
mv $HOME/Programs/orca_5_0_4_linux_x86-64_shared_openmpi411 \
  $HOME/Programs/orca_504
```

After completing these steps, you need to set the environment variables for ORCA. To do so, enter the following commands in your terminal. Make sure to adjust the ORCA directory based on your chosen installation path:

```
echo 'export PATH="$HOME/Programs/orca_504:$PATH"' >> $HOME/.bashrc
```

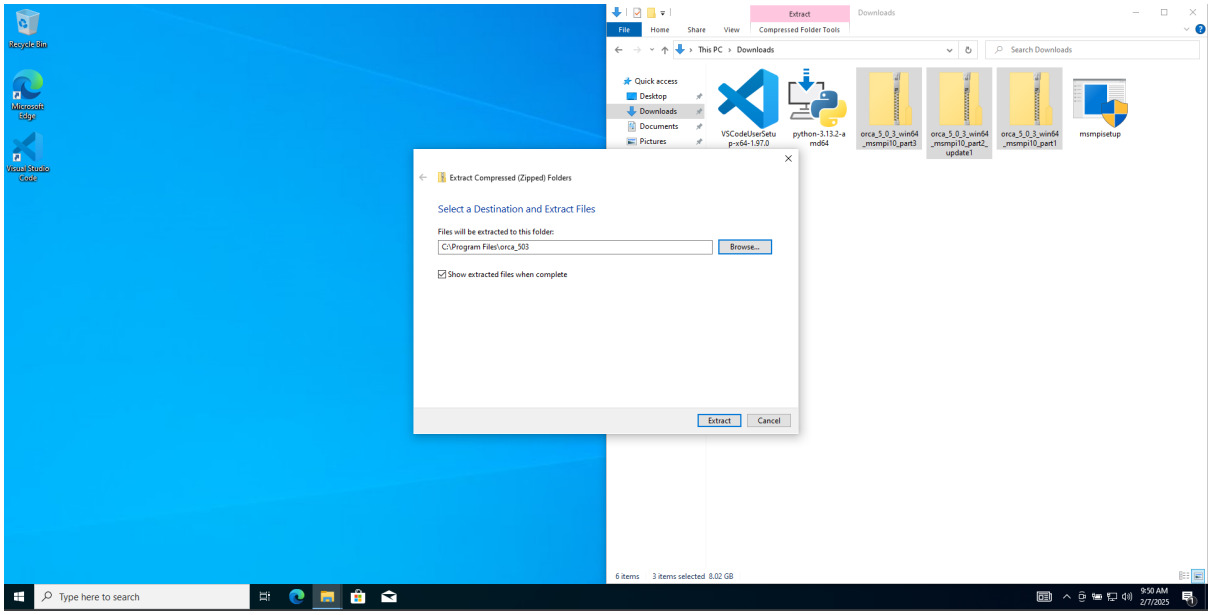
```
echo \
  'export LD_LIBRARY_PATH="$HOME/Programs/orca_504:$LD_LIBRARY_PATH"' \
  >> $HOME/.bashrc
```

Finally, apply the changes:

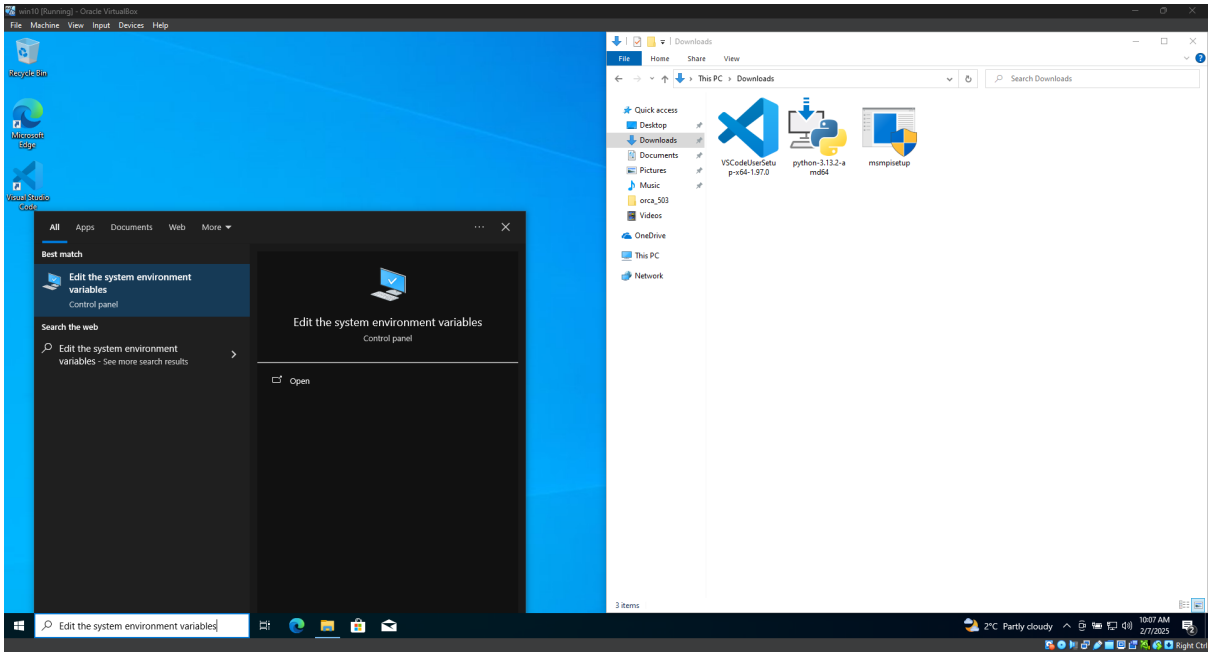
```
source $HOME/.bashrc
```

5.2.2 Windows

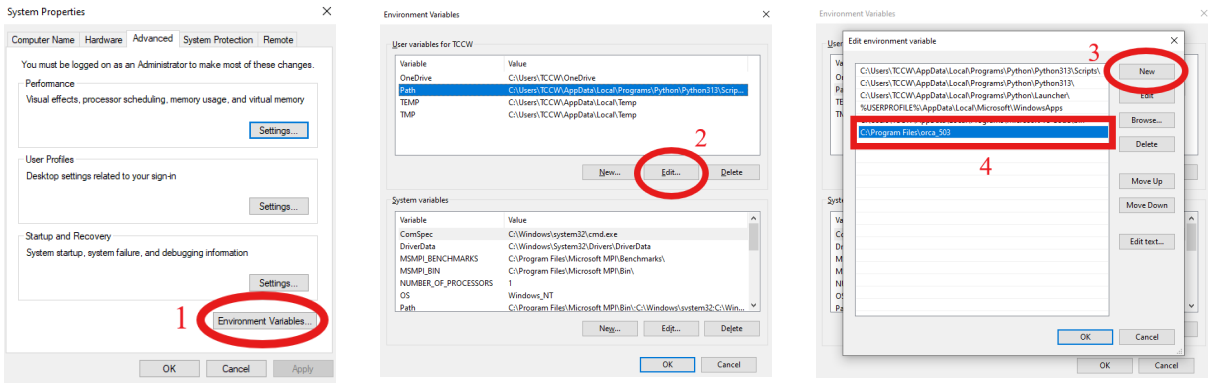
To install ORCA 5.X.X on Windows, navigate to the directory where you downloaded the files. Then, right-click on the ORCA archive and extract it to a predefined directory:



Press the Start button on your keyboard and search for **Edit the system environment variables**. Ensure that you extract all three parts of the program.



After opening the Environment Variables settings, follow the steps illustrated below to add ORCA to your system's PATH variables. Make sure to update the directory in step 4 to match the location where you extracted ORCA.



5.3 Verifying the Installation

To verify the ORCA installation, open VSCode. From the **File** tab, select **Open Folder** and navigate to the `WSCC-2025` directory (files you downloaded from our GitHub repository). Next, go to the **Terminal** tab and select **New Terminal**. In the terminal, use the following command to navigate to the `00_guides/02_test_scripts/01_orca_serial` folder:

- Linux:

```
cd 00_guides/02_test_scripts/01_orca_serial
```

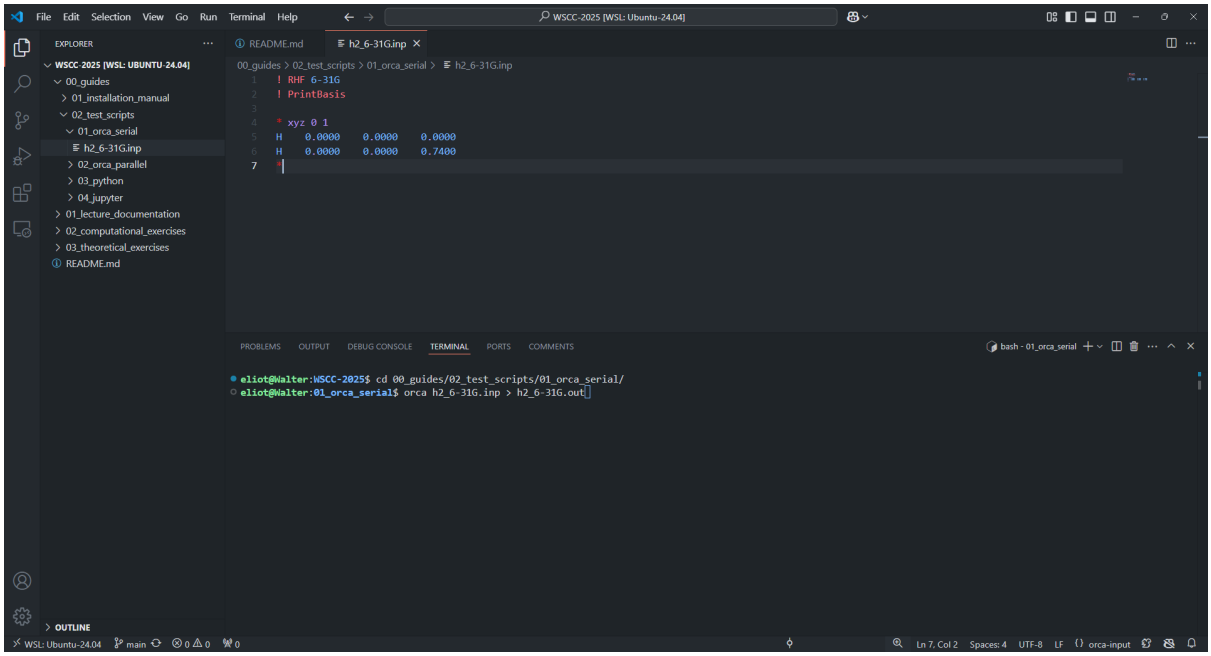
- Windows:

```
cd 00_guides\02_test_scripts\01_orca_serial
```

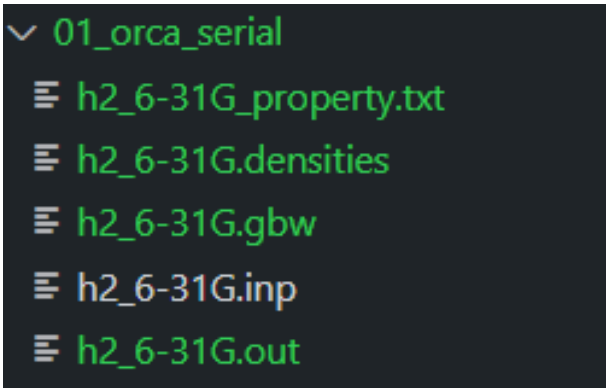
This directory contains a sample input file for an ORCA serial job. A serial job runs on a single core of your CPU. To execute the job, enter the following command:

```
orca h2_6-31G.inp > h2_6-31G.out
```

At this step your working environment should be like this:



The job should complete within seconds. After execution, you should see the following files in the directory, green files are the ones that created after job's compeltion:



Open the `h2_6-31G.out` file and scroll down to the final lines. You should see the following message, confirming that the job finished successfully:

```
****ORCA TERMINATED NORMALLY****
```

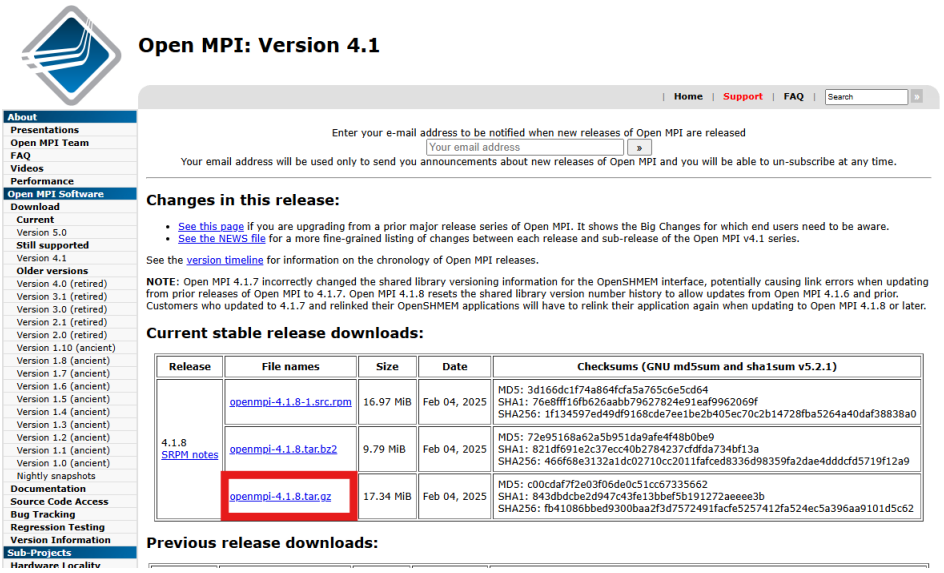
6 Parallelization Packages (Optional)

ORCA supports parallel execution, allowing calculations to run on multiple CPU cores simultaneously. This significantly accelerates computations, reducing the time required for complex simulations. By distributing tasks across multiple processors, parallelization enhances efficiency and enables the handling of larger systems. To run parallel jobs, you need to install Open MPI for Linux and Microsoft MPI for Windows. Proper installation and configuration of these libraries are essential for optimal parallel performance in ORCA.

6.1 Downloading the Software

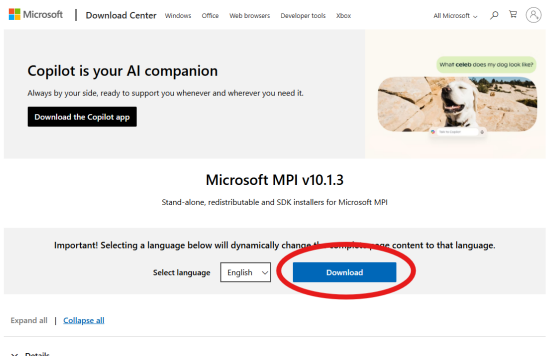
6.1.1 Linux

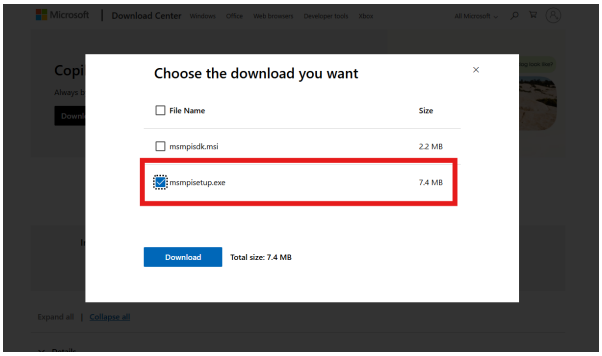
To download Open MPI, first visit their web page: open-mpi.org. Then download the 4.1.8 version in gz format:



6.1.2 Windows

To download Microsoft MPI, first visit their web page: microsoft.com. Then download the `.exe` version of the software:





6.2 Installating the Software

6.2.1 Linux

To Install the Open MPI package, first extract the archive file using the command below. We assumed that the downloaded file is in `$HOME/Downloads` directory:

```
sudo tar -xvzf $HOME/Downloads/openmpi-4.1.8.tar.gz -C $HOME/Downloads
```

Then run the following commands in order to install Open MPI. Here we assumed we want to install the package in `$HOME/Programs` directory:

```
cd $HOME/Downloads/openmpi-4.1.8
```

```
sudo ./configure --prefix=$HOME/Programs/openmpi_418
```

```
sudo make all -j$(nproc)
```

```
sudo make install
```

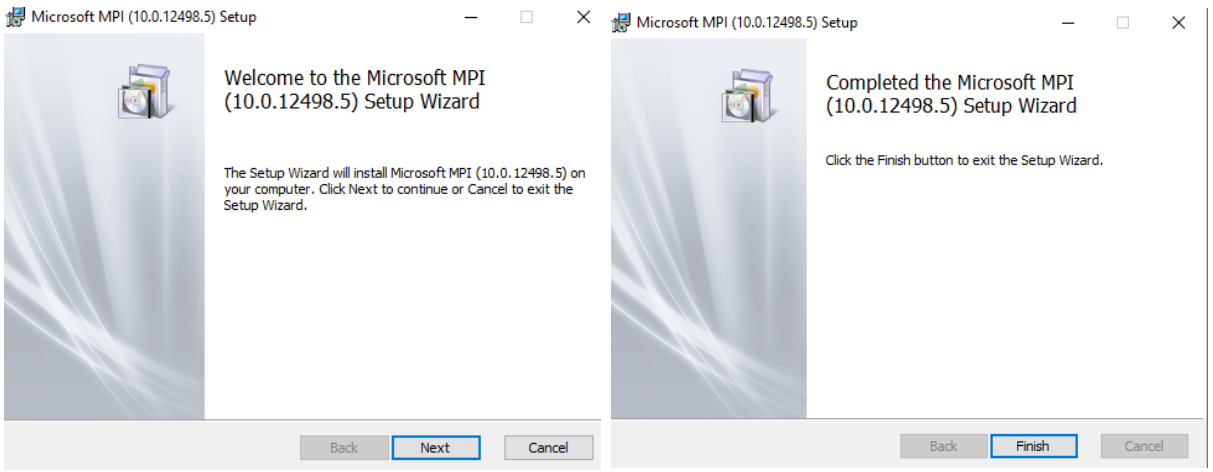
The final step is to setup the path variables for Open MPI package. In your terminal enter the following commands in order:

```
echo 'export PATH="$HOME/Programs/openmpi_418:$PATH"' >> $HOME/.bashrc
```

```
echo \
    'export LD_LIBRARY_PATH="$HOME/Programs/openmpi_418:$LD_LIBRARY_PATH"' \
    >> $HOME/.bashrc
```

6.2.2 Windows

To install Microsoft MPI on Windows, run the setup file and follow the standard installation procedure.



6.3 Verifying the Installation

In order to verify your installation of parallelization packages was successful, open VS-Code. From the **File** tab, select **Open Folder** and navigate to the `WSCC-2025` directory. Next, go to the **Terminal** tab and select **New Terminal**. In the terminal, use the following command to navigate to the `00_guides/02_test_scripts/02_orca_parallel` folder:

- Linux:

```
cd 00_guides/02_test_scripts/02_orca_parallel
```

- Windows:

```
cd 00_guides\02_test_scripts\02_orca_parallel
```

This directory contains a sample input file for an ORCA parallel job. A parallel job runs on multiple cores of your CPU. The important note on executing parallel jobs is that you should call ORCA using its full path:

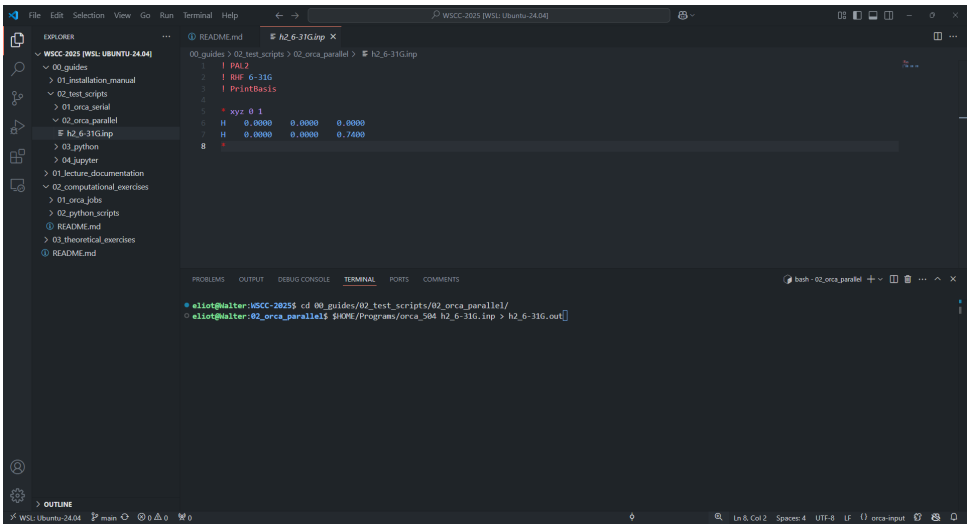
- Linux:

```
$HOME/Programs/orca h2_6-31G.inp > h2_6-31G.out
```

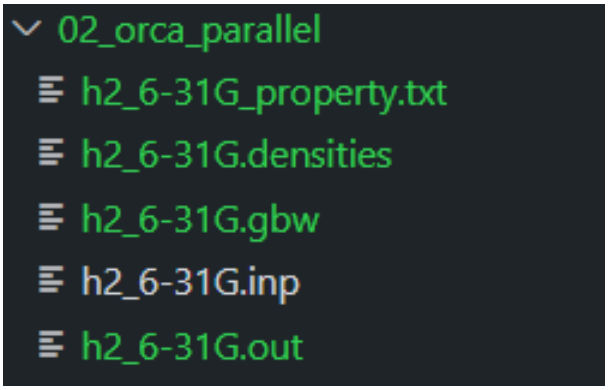
- Windows:

```
"C:\Program Files\orca_504\orca.exe" h2_6-31G.inp > h2_6-31G.out
```

At this step your working environment should be like this:



Note that here we assume ORCA is installed in `$HOME/orca_504` directory for Linux and `C:\Program Files\orca_504` directory for Windows. The job should complete within seconds. After execution, you should see the following files in the directory:



Open the `h2_6-31G.out` file and scroll down to the final lines. You should see the following message, confirming that the job finished successfully:

```
****ORCA TERMINATED NORMALLY****
```

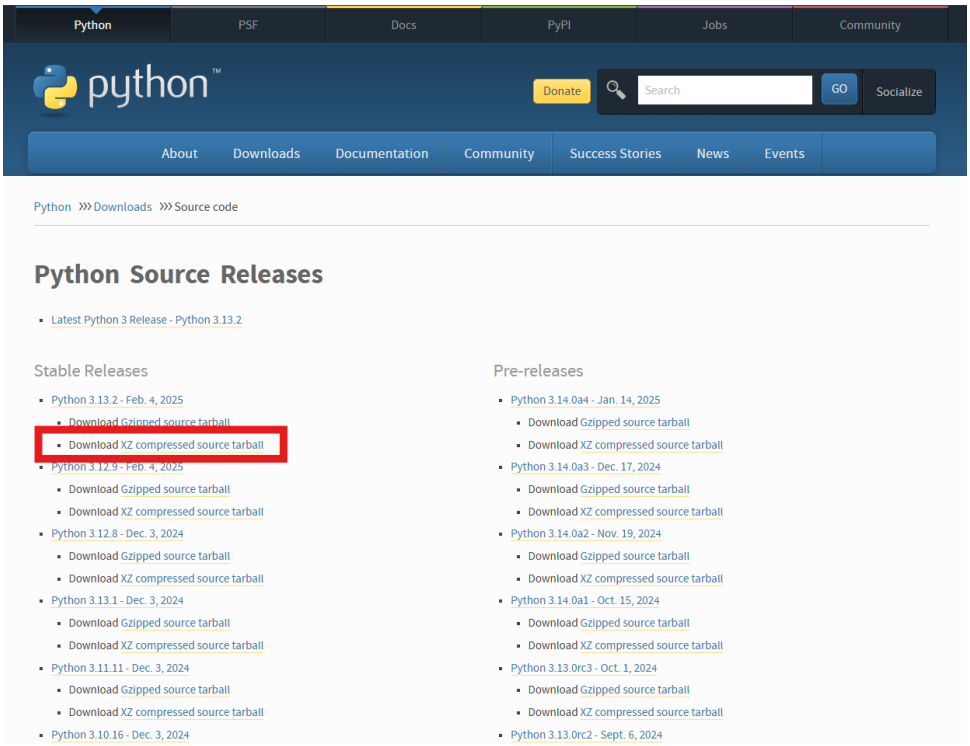
7 Python

Python is a widely used programming language in computational chemistry due to its ease of use, flexibility, and powerful libraries. It enables automation of workflows, data analysis, molecular modeling, and integration with quantum chemistry and molecular dynamics software. With libraries like NumPy, Pandas, and ASE, Python streamlines simulations, visualization, and computational tasks, making it an essential tool for chemists.

7.1 Downloading the Software

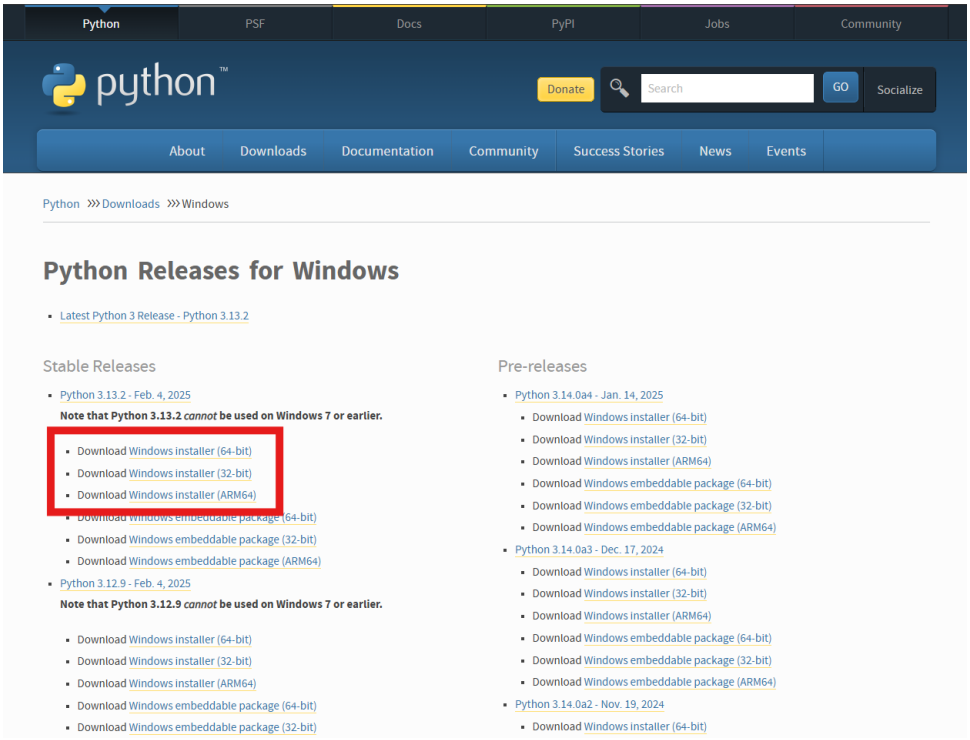
7.1.1 Linux

To download Python for Linux, first visit their web: python.org. Then download the version that showed in the picture below:



7.1.2 Windows

To download Python for Windows, first visit their web page: python.org.Then download the version that showed in the picture below:



7.2 Installating the Software

7.2.1 Linux

Linux has its own version of Python preinstalled, but you should not work with that becasue it might interrput your system files and crash your OS. So follow the precedure below to install it properly. Open a terminal and enter the command below to install the dependencies for Python:

```
sudo apt update && sudo apt upgrade
sudo apt install -y build-essential libssl-dev zlib1g-dev libbz2-dev \
                    libreadline-dev libsqlite3-dev wget curl llvm \
                    libncurses-dev xz-utils tk-dev libffi-dev liblzma-dev \
                    python3-openssl git libgdbm-dev libc6-dev libdb-dev \
                    libexpat1-dev libpcap-dev libmysqlclient-dev
```

Here we assumed the downloaded Python file is place at `$HOME/Downloads` folder. Extract the archive using the following command:

```
sudo tar -xf $HOME/Download/Python-3.13.2.tar.xz -C $HOME/Downloads
```

Next navigate through the extracted folder:

```
cd $HOME/Downloads/Python-3.13.2
```

Enter the following commands in order to install Python version 3.13:

```
sudo ./configure --enable-optimization
```

```
sudo make -j$(nproc)
```

```
sudo make altinstall
```

After these stepts, the Python 3.13 is installed at `/usr/local/bin/` directory. In the next step you will install the Virtualenv python package to have isolated workspace for the winter school:

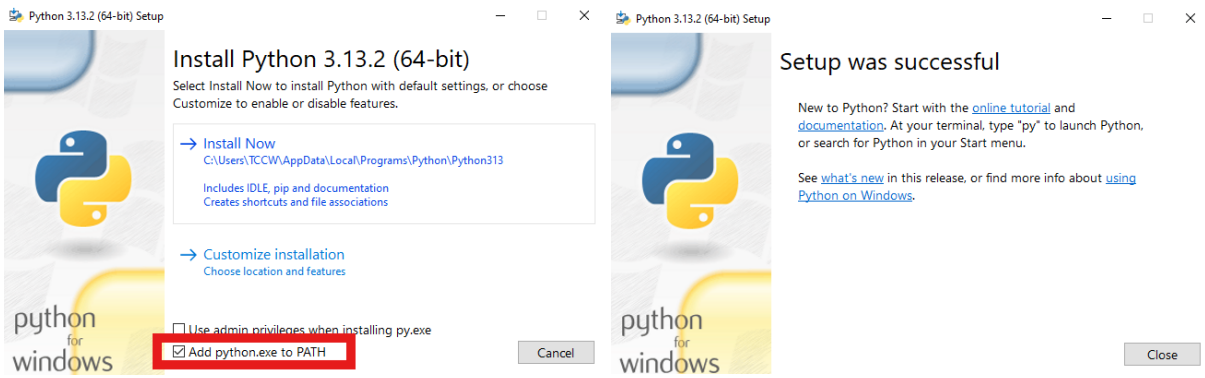
```
python3.13 -m pip install virtualenv
```

For the final step, to create a Python virtual environment for the winter school, enter the following command in the terminal. This commands create a virtualenv at `$HOME/Documents/wscc-env` directory:

```
python3.13 -m virtualenv $HOME/Documents/wscc_env
```

7.2.2 Windows

To install Python on Windows, run the setup file and follow the standard installation procedure. Just Make sure you checked the —textbfAdd python.exe to PATH:



After these stepts, the Python 3.13 is installed at `C:\Users\<User Name>\AppData\Local\Programs\Python\Python313` directory. In the next step you will install the Virtualenv python package to have isolated workspace for the winter school. Enter the following command in your terminal

```
pip install virtualenv
```

For the final step, to create a Python virtual environment for the winter school, use the following command. This commands create a virtualenv at `C:\Users\<User Name>\Documents\wscc_env` directory:

```
virtualenv C:\Users\<User Name>\Documents\wscc_env
```

Note that in the two previous steps you should replace **<User Name >** by your actuall windows username.

7.2.3 Python Liberaries

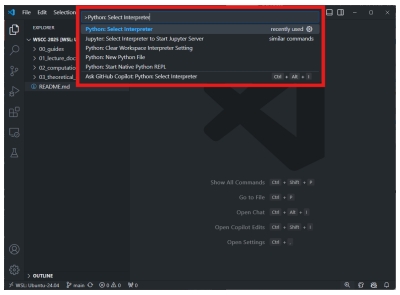
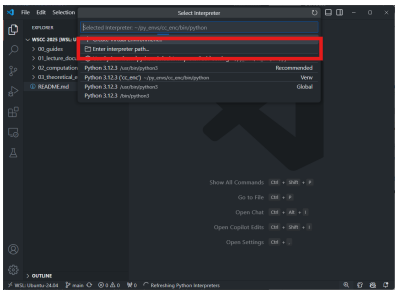
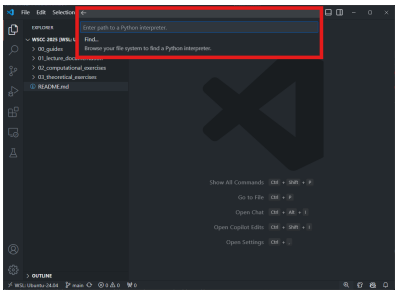
To install the Python libraries that needed in the winter school, you first need to activate the virtual environment you created in previous steps. Open VSCode. From the **File** tab, select **Open Folder** and navigate to the `WSCC-2025` directory. Next press Ctrl+Shift+P to open search bar for VSCode command pallet. in this box search for **Python: Select Interpreter**. Next click on **Enter interpreter path**, and then **Find**. Here go to the virtualenv directory, if you successfully finished the previous steps, the virtualenc python interpreter path is like the one we show below, find it and select it:

- Linux:

```
$HOME/Documents/wscc_env/bin/python3
```

- Windows:

```
C:\Users\<User Name>\Documnets\wscc_env\Scripts\python
```

In the next from the **Terminal Tab**, open a new terminal and enter the following command:

- Linux:

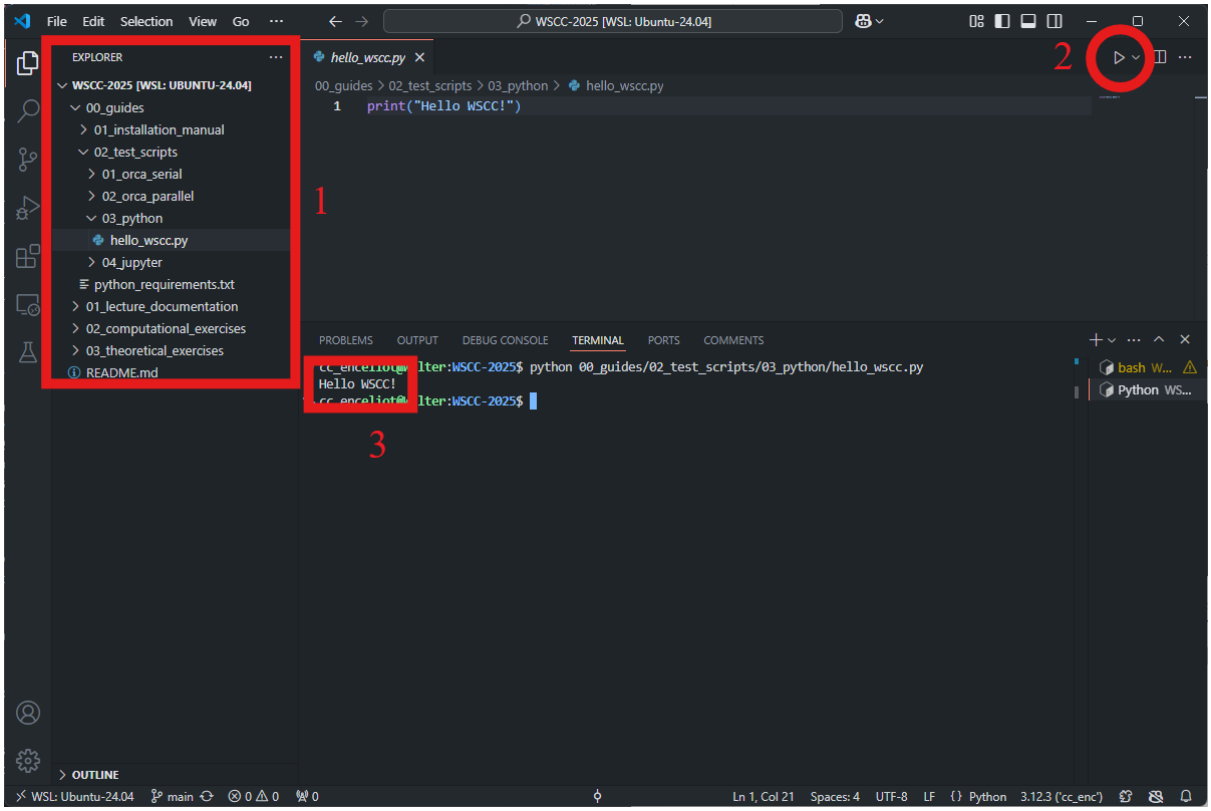
```
pip install -r 00_guides/python_requirements.txt
```

- Windows:

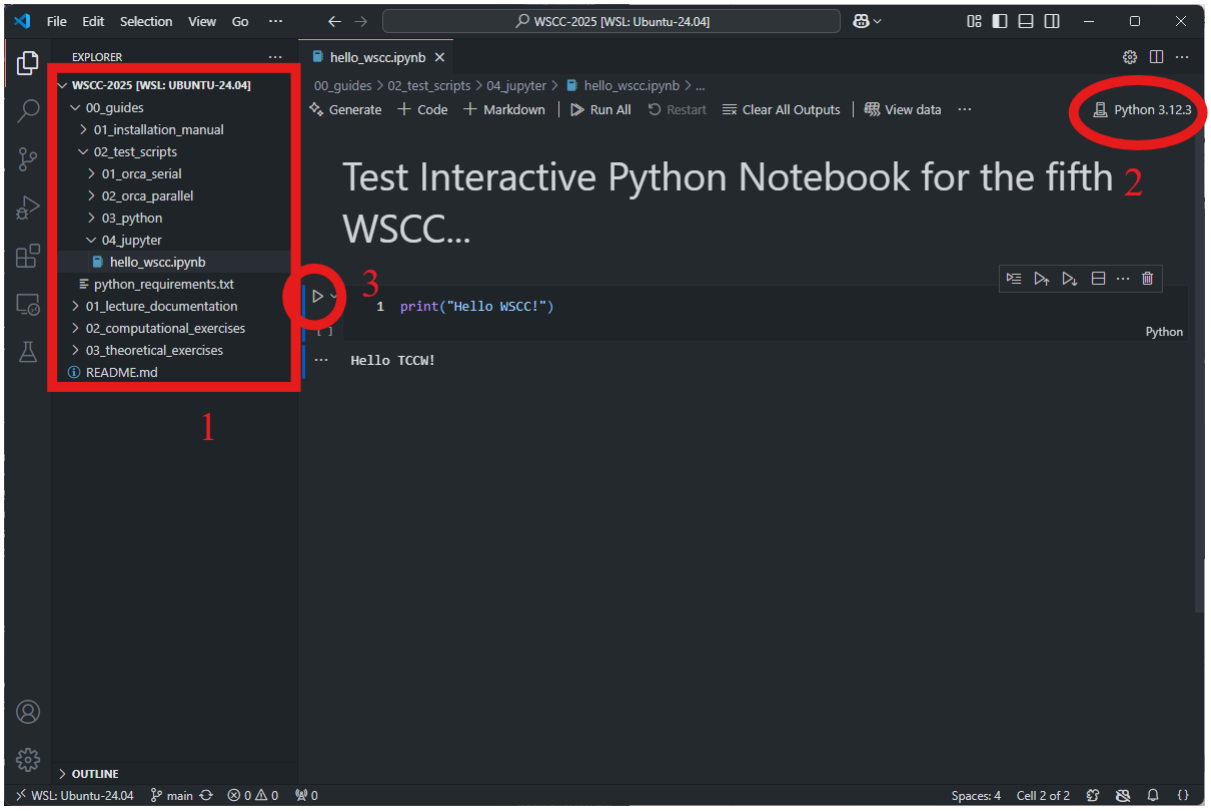
```
pip install -r 00_guides\python_requirements.txt
```

7.3 Verifying the Installation

To verify the Python installation, open VSCode. From the **File** tab, select **Open Folder** and navigate to the `WSCC-2025` directory. Next activate the virtualenv as we explained in **Section 7.2.3**. From the file explorer panel at the top left side of the VSCode go to the `00_guides/02_test_scripts/03_python`, open `hello_wscc.py` file. Press the play button from the top right corner. In the opened terminal you should see **Hello WSCC!** dialog.

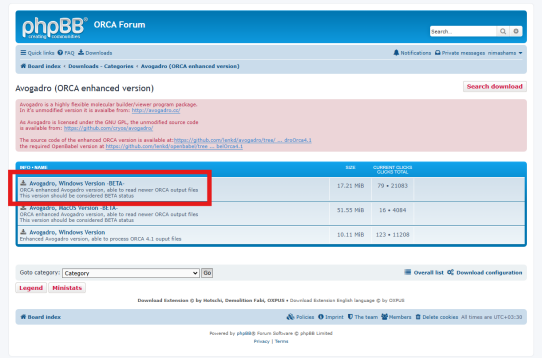


To verify Jupyetr installation, from the file explorer panel at the top left side of the VSCode go to the `00_guides/02_test_scripts/04_jupyter`, open `hello_wscc.ipynb` file. Press the play button at the left side of the cell. Below the cell you should see **Hello WSCC!** dialog.



8 Avogadro

Avogadro is an open-source molecular visualization and editing software widely used in computational chemistry. It provides an intuitive interface for building molecular structures, preparing input files, and visualizing results from quantum chemistry calculations. Avogadro is an excellent choice for working alongside ORCA, as it simplifies the creation of input files, enables easy geometry optimization, and helps visualize molecular orbitals, vibrational modes, and electron densities from ORCA output files. Its flexibility and plugin support make it a valuable tool for researchers in quantum chemistry.
ORCA has its own Avogadro version wich is the recommended version for this winter school. This version only is available for Windows users. In order to download it go to the downloads section of the orcaforum.kofo.mpg.de, find **Avogadro (ORCA Enhanced Version)**, and the download **Avogadro, Windows Version -BETA**. For installation just open the downloaded file and follow the standard installation procedure.



Linux users can install normal version of Avogadro using the apt command:

```
sudo apt install avogadro
```