Chapter 2

Software Installation

This chapter will guide the readers to install all necessary software for running Quantum ESPRESSO. We expect that the readers already have a computer with an operating system (OS) such as Linux, Windows, and macOS. Firstly, we will show you the minimum requirements for each OS to avoid start-up problems in installing Quantum ESPRESSO. Next, we will show the main method to install Quantum ESPRESSO and its supporting software. Finally, we will show you how to run the simplest running instance of Quantum ESPRESSO.

2.1 Preparing the operating systems

Installing Quantum ESPRESSO could be challenging for beginners, but we are confident that by following all the steps given in this chapter, the readers will have a smooth, working environment of Quantum ESPRESSO. The main reason for the challenge is that Quantum ESPRESSO is an open-source software that consists of various packages originally targeted for Linux (or Unix-like OSs), which might sound unfamiliar for most Windows users in the world. However, current technology allows us to install Linux (or any other

Quantum ESPRESSO Course for Solid-State Physics
Nguyen T. Hung, Ahmad R. T. Nugraha, and Riichiro Saito

ISBN 978-981-4968-37-9 (Hardcover), 978-1-003-29096-4 (eBook)
www.jennystanford.com

OS) as a "guest" OS on Windows using either Windows subsystem for Linux (WSL) or a virtual machine (such as VirtualBox). Therefore, it is currently possible for Windows users to "run Quantum ESPRESSO in Windows" by using the WSL or VirtualBox.

Since we realize that the readers of this book use various operating systems in their daily life, we will show you how to install Quantum ESPRESSO in Linux, Windows, and macOS. We chose these OSs simply due to their popularity although we will not explain how to install the OSs. We assume the readers have such a specific OS and **internet access** in their own computers.

It should be noted that the installation guide for Windows users is mostly arranged in a way to show how we can utilize the virtualization feature. It means that after installing the "virtual" Linux on Windows, the remaining Quantum ESPRESSO installation procedure should be the same as that on Linux. Before proceeding to the Quantum ESPRESSO installation guide, we recommend you ensure that your OS is updated to the latest version. We will show you how to prepare three specific OSs, i.e., Ubuntu Linux, Windows, and macOS Catalina.

2.1.1 Ubuntu Linux

In Ubuntu, please check that you can open a `Terminal` window from the `Activities` panel. `Terminal` is a place in the OS where we can interact with the "shell". Simply saying, the shell is a program that takes commands from the keyboard and returns them to the operating system to perform.

In the old days, the shell was the only user interface available on a Unix-like system. Nowadays, we have graphical user interfaces (GUIs) in addition to the shell, which is a kind of command-line interfaces (CLIs). To interact with the shell in the terminal, we can type a line of "command" or (lines of commands) corresponding to some computer operations.

The basic structure of Ubuntu `Terminal` is shown in Fig. 2.1. We can see there is a prompt showing the login username, machine hostname, and a current working directory ("∼" sign, indicating the so-called "home" directory of the user). The prompt ends with a dollar ($) sign. The cursor next to the $ sign is the starting point

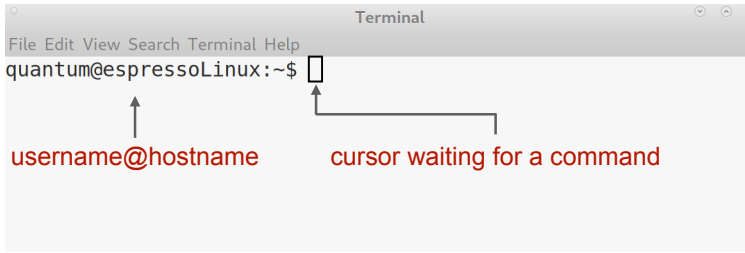

Figure 2.1 Terminal in Ubuntu.

where we can type the actual Linux commands. There are so many commands for various purposes, but we will only use a few of them to install software and manage our Quantum ESPRESSO jobs and files.

To update all software in Ubuntu, on the terminal, type the following commands (next to the $ sign) and press enter after each line:

```
$ sudo apt update
$ sudo apt upgrade
```

You will be asked for "your password" (not root password) when executing the above commands because the administrator (often called "root") privileges are required when installing or updating the software in Ubuntu. The root privileges are represented by the `sudo` command.

The `apt` command corresponds to the "Advanced Packaging Tool", a command-line tool that helps in handling packages in Ubuntu. Its main task is to retrieve the information and packages from the authenticated sources (mostly from the internet) for installation, upgrade, and removal of packages along with their dependencies. In the above example, the `update` and `upgrade` texts are a set of commands to ensure your Ubuntu software is all updated.

We also need some development tools and libraries, such as Git (version control system), GNU Wget (file retrieval software), GCC/G++(C/C++ compiler), GFortran (Fortran compiler), LAPACK (linear algebra package), FFTW (Fourier transform library), and Open MPI (parallel computing implementation). All of them are

necessary to compile Quantum ESPRESSO. On the terminal, execute the following commands one by one:

```
$ sudo apt install git wget build-essential
$ sudo apt install g++ gfortran liblapack-dev
$ sudo apt install libfftw3-dev libopenmpi-dev
```

If you are prompted by a "Yes/No" question, just type "y" (or "Yes") and press enter to agree with the installation.

2.1.2 Windows

For Windows users, the OS should be updated to at least Windows 10 version 1903. If you want a native support for Linux GUI applications, we recommend to use or update Windows 11 with build number at least 22000. To perform the update, we should open `Windows Update` from the `Start` menu in Windows, and we will either see the option for `Feature update`, or we will have to click `Check for updates` (see Fig. 2.2).

If you still do not see the option to update Windows to the latest version, click `Check for updates`, reboot your computer, and try the process again. There is also a possibility that you already have this update without realizing it. You can find the OS information in Windows by typing → `About your PC` in the Windows `Start` menu. Under Windows specifications, check which edition and version of Windows your computer is running. If you see version 1903 or above (see Fig. 2.3), you can continue to the next step.

Even if Windows Update does not offer the update, we can download Microsoft's Update Assistant tool to install it manually. This tool will give you the update even if Microsoft is not confident it is ready for your computer yet. With a version of more than 1903, it is possible for us to install Windows subsystem for Linux (WSL) with the ability to access Linux files from File Explorer or other file manager applications. Unlike older ways of accessing virtual Linux files, this version of Windows offers full read-write access without the worry of breaking anything.

Next, we have to check whether or not our computer supports virtualization. In most computers, the virtualization can be enabled

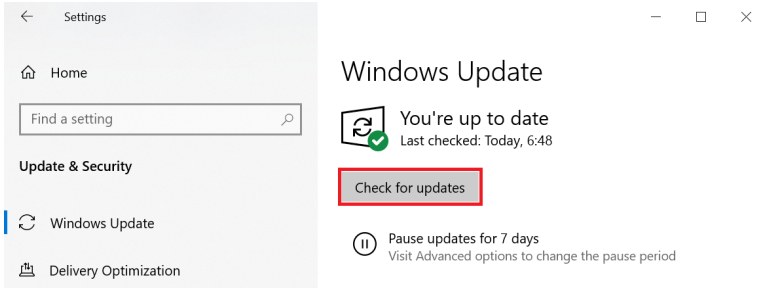

Figure 2.2 Windows Update. You can click `Check for updates` and `Download and install now` to guarantee your Windows is in the latest version.

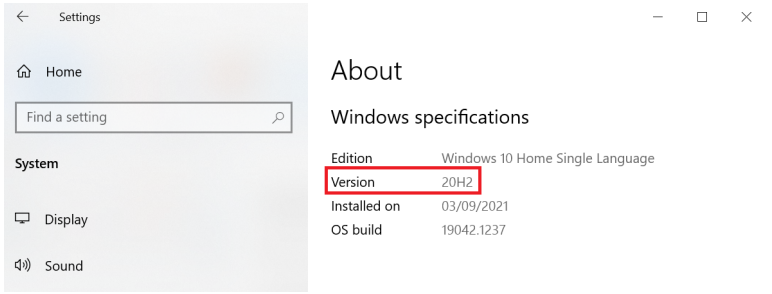

Figure 2.3 Windows specifications in `About your PC`. Make sure that you are running at least Windows 10 version 1903. Again, for a native support of Linux graphical applications in WSL, we recommend at least Windows 11 build 22000.

from BIOS, which can be accessed before booting your OS. The key you should press on the keyboard to access BIOS will depend on the manufacturer of the computer. You can check from the computer screen when you turn it on. Assuming the key is `F2` (or `DEL`), below are the steps to enable the virtualization from the BIOS settings:

- Turn on or restart the computer.
- Press `F2` (or `DEL`) key at startup BIOS Setup.
- Press the right arrow key to `System Configuration` tab, select `Virtualization Technology` or a similar option and then press the `Enter` key.
- Select `Enabled` and press the `Enter` key.

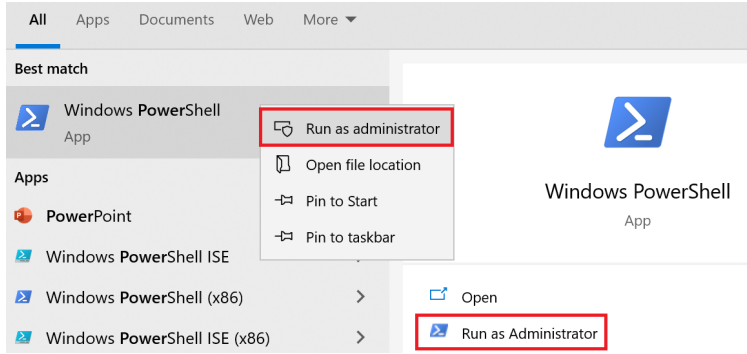

Figure 2.4 Opening PowerShell as the administrator.

- Press the F10 key and select `Yes` and press the `Enter` key to save changes and Reboot into Windows.

After performing the above steps, we are ready to install Windows Subsystem for Linux (WSL).

WSL is a Windows feature that enables us to run native Linux command-line tools directly on Windows. The merit of using WSL over other virtual machines is that WSL requires fewer resources (CPU, memory, and storage). WSL also allows us to run Linux command-line tools and applications (apps) alongside Windows command-line, desktop, and store apps and access Windows files from Linux. This technique allows us to use Windows apps and Linux command-line tools on the same set of files.

To install WSL and Ubuntu Linux automatically on Windows, first open PowerShell as an administrator by searching "PowerShell" in the Windows menu and right-click PowerShell to get access as an administrator (see Fig. 2.4). Inside PowerShell, type the following command and press enter:

```
wsl --install
```

Follow the instructions on the screen and restart the computer when prompted. This reboot is required in order to ensure that WSL can initiate a trusted execution environment.

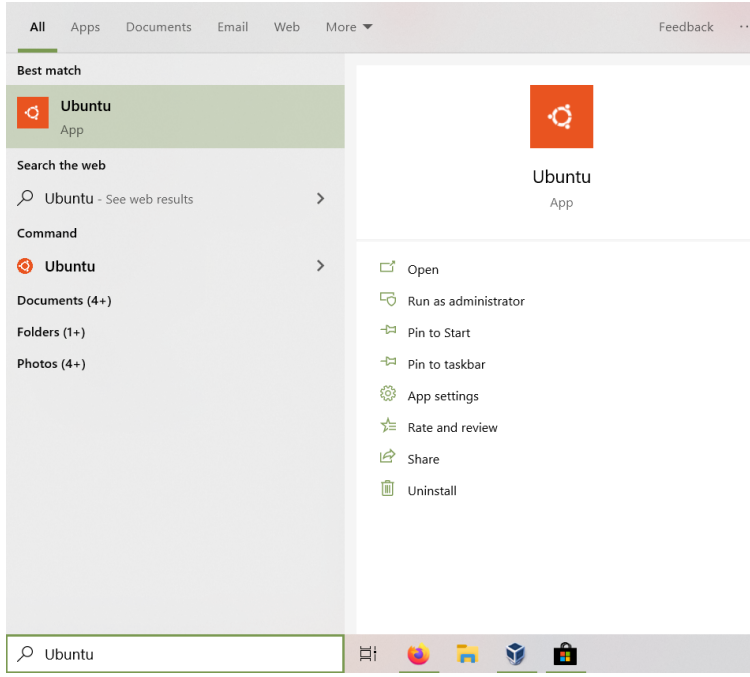

Figure 2.5 Launching Ubuntu from the Start menu.

After reboot, WSL should initialize the Ubuntu instance once before the system can be used properly. If everything goes normally, Ubuntu initialization should launch automatically when we re-login to Windows. *If not*, we should launch a new instance of Ubuntu by clicking the "launch" button in the Microsoft Store app for Ubuntu or launching Ubuntu from the Start menu (Fig. 2.5). Note that if you cannot perform this kind of WSL installation, please follow the detailed manual instructions in Microsoft Docs: `https://docs.microsoft.com/en-us/windows/wsl/install-manual`

The first time a new Ubuntu instance runs, a console window will open, and we will be asked to wait for several minutes for the installation to complete. During this final stage of installation, the Ubuntu files are extracted and stored on your PC. It may take a couple of minutes, depending on the performance of your computer's storage devices. This initial installation phase is only required when a Linux distribution in Windows is clean-installed. All future launches should take less than a second.

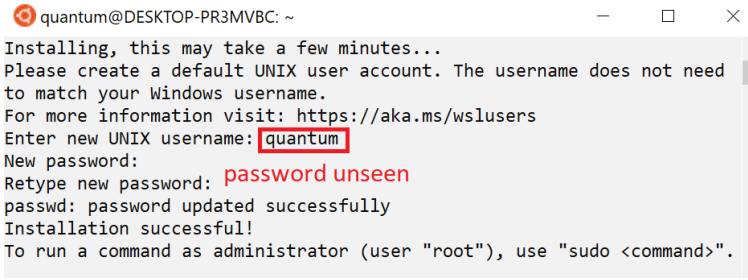

Figure 2.6 First Ubuntu instances through WSL and creation of a user account.

Once installation is complete, we will be prompted to create a new user account (and its password), as shown in Fig. 2.6. This user account is for the normal non-admin user that will log in as default when launching Ubuntu. We can choose any username and password. There is no necessity to match the Windows username.

In our example, we set "quantum" as the username. In Fig. 2.6, you should notice that the password is not displayed by the system. Therefore, be careful when typing it for the first time. Next time we open a new Ubuntu instance, we will not be prompted for the password. Only when we want to elevate a process using `sudo` to access root privilege, we need to enter the user password, so make sure to choose a password that is difficult to forget as well as difficult to break.

What you see in Fig. 2.6 is essentially the same shell as in Fig. 2.1 that shows the Ubuntu terminal. Therefore, hereafter, we loosely refer to the shell as the Ubuntu terminal. You can open a File Explorer window directly from the terminal in the current directory, which is the home (also denoted `$HOME`) directory of Linux. The corresponding command is:

```
$ explorer.exe .
```

You can work with files normally from here (see Fig. 2.7). Use drag and drop, copy and paste them, or even open them directly in Windows applications.

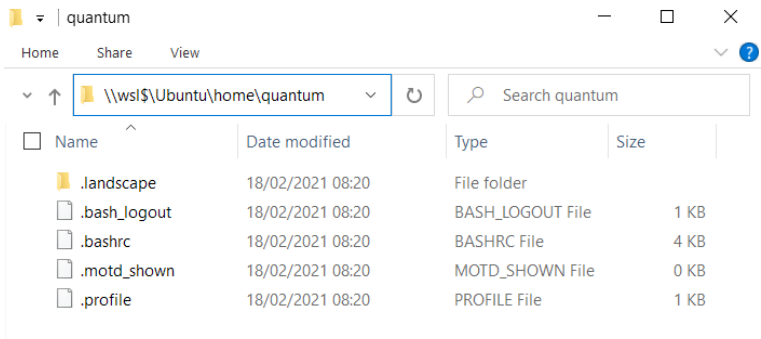

Figure 2.7 WSL `$HOME` directory opened in Windows.

Besides typing the above command, you can also open the `$HOME` directory from Windows Explorer by directing the explorer to the following address:

```
\\wsl$\Ubuntu\home\quantum
```

(Change "quantum" with your username of Ubuntu in WSL.)

As in Ubuntu Linux, we should update all the default software and install development tools with libraries necessary to compile Quantum ESPRESSO. We type the following commands one by one in the Ubuntu terminal on WSL:

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt install git wget build-essential
$ sudo apt install g++ gfortran liblapack-dev
$ sudo apt install libfftw3-dev libopenmpi-dev
```

Note again, the $ sign should not be typed/copied to the terminal. You will also be asked for your password when executing the above commands because administrator privileges are required. The detailed explanation of the meaning of each line above can be read in the previous section (Sec. 2.1.1).

By completing all the above steps, we consider that you are already "having a Linux OS" on Windows. Therefore, the remaining procedure to install Quantum ESPRESSO and its supporting software

is the same as that in the "real" Linux OS, which will be explained in Sec. 2.2.

2.1.3 macOS Catalina

macOS is one of Unix-based OS, so that it already has the terminal tool. However, some additional packages are needed before we can install Quantum ESPRESSO. We recommend the Homebrew package manager to install the missing development tools and libraries. For the tutorial in this book, we tested the package installations on macOS Catalina. On the terminal, we first have to execute the following command to get Homebrew:

```
$ /bin/bash -c "$(curl -fsSL https://raw.
    githubusercontent.com/Homebrew/install/HEAD/
    install.sh)"
```

The command is quite long, and it may look broken into several lines in this book or your terminal, but it is still a single line of command. You should ensure typing the command properly to the terminal as a unified entity, without pushing "Enter" key.

By having Homebrew in macOS, the prerequisites for the Quantum ESPRESSO installation can be obtained by another single line of command:

```
$ brew install git wget gcc gfortran fftw lapack
    openblas open-mpi eigen
```

Again, no matter how the above (long) command looks broken into separate lines, it is a single line of command.

2.2 Installation of Quantum ESPRESSO and its supporting software

With the various OSs that the readers have, we are ready to install Quantum ESPRESSO using the same approach on the terminal.

We will install Quantum ESPRESSO and an additional Wannier90 package, either in "real" Ubuntu Linux, WSL Ubuntu, or macOS.

For beginner users of Ubuntu Linux and WSL Ubuntu, the following commands are sufficient to install Quantum ESPRESSO and Wannier90:

```
$ sudo apt install quantum-espresso
$ sudo apt install wannier90
```

The above commands will install slightly earlier versions of Quantum ESPRESSO and Wannier90 which are already sufficient for running the tutorials given in Chapter 3. On the other hand, for readers who want to install the latest version of Quantum ESPRESSO and Wannier90 from source code, please follow the rest of this section.

We recommend the readers to create a new folder `opt` in the `$HOME` directory. Type this command on the terminal:

```
$ mkdir opt
```

Then, go into the `opt` folder:

```
$ cd opt
```

In the `opt` folder, we can download the latest source files of Quantum ESPRESSO using the Git utility as follows:

```
$ git clone https://github.com/QEF/q-e.git
```

After the source files are downloaded, we can enter the `q-e` folder:

```
$ cd ~/opt/q-e/
```

In this folder, we execute the configuration command:

```
$ ./configure
```

Quantum ESPRESSO will automatically determine the best setting for our OS. If no error occurs, we can continue with making the executable files:

```
$ make all
$ make w90
```

The purpose of the first line above is to make executable files for all standard Quantum ESPRESSO capabilities, such as electronic structure, phonon dispersion, and optical properties calculations. The second line is to make the additional Wannier90 package available in the binary folder. In addition to this compilation process, our OS must be "taught" how to find the executable files and include them in its variable path. For this purpose, we can execute two more commands to update the path:

```
$ echo 'export PATH=$PATH:$HOME/opt/q-e/bin' >> ~/.
    bashrc
$ source ~/.bashrc
```

The above command lines enable us to access all binary files of Quantum ESPRESSO without typing the complete path up to its `bin` folder.

To check that our Quantum ESPRESSO installation is done successfully, we can type the following command, which is the most basic command of Quantum ESPRESSO:

```
$ pw.x
```

We will learn how to use this command briefly in Sec. 2.4 and more extensively in Chapter 3. To exit the Quantum ESPRESSO environment generated from the above command line, push `CTRL+c` keystroke.

We can install some additional software to support our workflow when performing first-principles calculations with Quantum ESPRESSO. Executing the following commands line by line on the terminal will give us all necessary software:

```
$ sudo apt install xcrysden gnuplot
$ sudo apt install python3-dev python3-pip
$ pip3 install numpy scipy sympy
$ pip3 install matplotlib jupyterlab
$ echo 'export PATH="$HOME/.local/bin:$PATH"' >> ~/.
```

Some notes:

- XCrysden is useful for visualization of molecule and crystal structures.
- Gnuplot is a plotting software. We may also use other plotting software. In the above commands, we include Python numerical libraries and graphical tools such as Matplotlib and JupyterLab that will be mainly used in this book.
- GUI support in WSL is required to run XCrysden, Gnuplot, and Jupyter. Windows 11 with build number at least 22000 already natively support GUIs. If you have a lower Windows version, you need to install an additional "X server", e.g., VcXsrv, X410 App, or Kali App. See the following Microsoft Tech Community for the X server installation: `https://techcommunity.microsoft.com/t5/windows-dev-appconsult/running-wsl-gui-apps-on-windows-10/ba-p/1493242`
- Depending on the system, it may take around 15-30 minutes to complete all the installation in this section.
- We have created a simple Bash script (see Sec. 6.3.2 to learn more about Bash), which collects all the above commands to install Quantum ESPRESSO and its supporting software for Ubuntu Linux and WSL Ubuntu that can be downloaded from `https://github.com/nguyen-group/QE-SSP/blob/master/QEinstall.sh`. After downloading the script, the readers can execute the following command line in the terminal:

```
$ bash QEinstall.sh
```

2.3 VirtualBox approach

If you have already succeeded to obtain a working Quantum ESPRESSO environment from the previous sections, you do not need to follow this VirtualBox approach. VirtualBox is one of the virtual machines that can be used to run other guest OSs from a specific OS. In Windows, for example, we can install VirtualBox to run a Linux distribution that has already been configured to include Quantum ESPRESSO and other necessary software inside it. There are three recommended virtual machines for Quantum ESPRESSO:

1. Quantum Mobile:
   `https://quantum-mobile.readthedocs.io`
2. MateriApps LIVE!:
   `https://cmsi.github.io/MateriAppsLive/`
3. QE-2021: `http://qe2021.ijs.si`

Each virtual machine has its username and password, which may change over time. Therefore, it is better to check the documentation on one of the websites above, depending on which virtual machine you like. However, all of them use the same capability of VirtualBox to host a virtual machine. Here we just explain how to install Virtual Box and add the virtual machine. The steps to install and use the virtual machine is as follows:

1. Download VirtualBox from: `https://www.virtualbox.org/wiki/Downloads`. Choose the Windows host (or macOS if the macOS users decide to use VirtualBox, too). Then, install and open VirtualBox.
2. Download the extension pack from the same webpage and install it while opening VirtualBox.
3. Download a virtual machine, either Quantum Mobile, MateriApps LIVE!, or QE-2021 from their websites listed previously. It is huge, around 3–4 GB, so if your internet connection is slow, better download the file when you are going to sleep.
4. From the VirtualBox software, press `Machine` and then `Add`. Select one of the virtual machines you have downloaded.

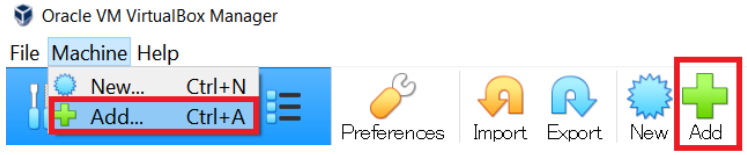

Figure 2.8 Adding a virtual machine in VirtualBox. Click the `Add` button, either from the `Machine` menu or from the toolbar.

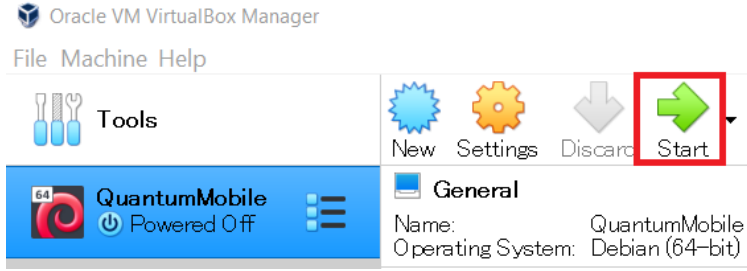

Figure 2.9 Starting the virtual machine.

Figure 2.10 Example desktop appearance of a virtual machine.

5. The virtual machine will be ready in a few minutes after extracting. Select the virtual machine from the left panel and press the "Start" button as shown in Fig. 2.9.
6. After starting up, we can see the desktop of the virtual machine as shown in Fig. 2.10.

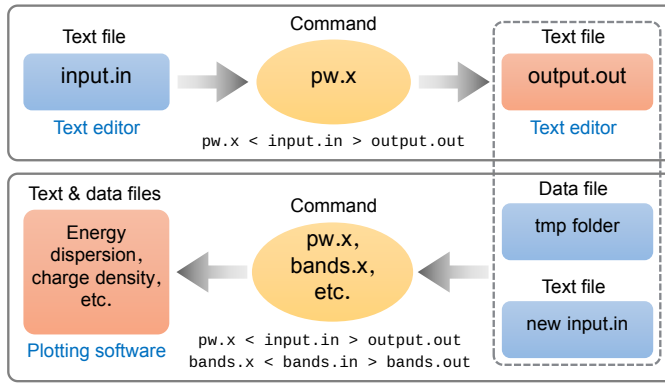

Figure 2.11 A typical Quantum ESPRESSO workflow.

2.4 Processing input and output files

With the Quantum ESPRESSO installed on our computer, we are now able to execute the simplest running instance of Quantum ESPRESSO, as shown in the diagram of a typical workflow of Quantum ESPRESSO in Fig. 2.11.

2.4.1 Basic execution of Quantum ESPRESSO commands

There are two main calculations in Quantum ESPRESSO. The first one is the self-consistent-field (SCF) calculation using the `pw.x` command. We will learn the details of the SCF calculation in Chapter 3, but we will need to prepare an input file that will be "read" by this command (for example: `input.in`), and then the calculation logs will be written into an output file (for example: `output.out`). The input file has some specifications for variables and parameters of the material simulation that should be prepared in advance using a text editor. We do not restrict the text editor that we use to edit the input file or check the output contents. Any text editor should be fine, for example, Notepad (or Notepad++) in Windows and `nano`, `vi`, or `emacs` in Linux. Suppose we already have the `input.in` file, the execution of the `pw.x` command is as follows:

```
$ pw.x < input.in > output.out
```

After the mandatory SCF calculation, the second calculation of a typical Quantum ESPRESSO simulation is the "real" stage for calculating the physical properties of the material under consideration. For example, in Fig. 2.11 we show another possibility of running `pw.x`, which is for the non-SCF (NSCF) calculation that includes the band structure and density of states (DOS). There will already be some data files produced from the NSCF calculations, but in most cases, we need to do "post-processing" before plotting and interpreting the data. Examples of the post-processing commands in Quantum ESPRESSO are `bands.x`, `dos.x`, `epsilon.x`, as shown in Table 2.1, which all will be learned in more detail in Chapter 3. The basic structure of the command execution is the same as the `pw.x` command, i.e.,

```
$ QEcommand < input.in > output.out
```

where `QEcommand` is any Quantum ESPRESSO command. For example, we may run the following in the case of `bands.x` command:

```
$ bands.x < band.in > band.out
```

We list some frequently-used Quantum ESPRESSO commands in Table 2.1. Note that besides the logs in the `output.out` file, the Quantum ESPRESSO commands also give other output (or data) files in a specified directory depending on the calculation type. The extension of the files is specified by the input file.

2.4.2 Choice of plotting software

In most cases, only after performing the post-processing commands we can obtain some meaningful data files related to the properties of the material and make some plots using our favorite plotting software. Although there are plenty of such softwares in the market, we decide to specifically suggest the readers of this book to

Table 2.1 Some frequently-used Quantum ESPRESSO commands.

Command	Purpose
`pw.x`	SCF and NSCF calculations
`bands.x`	band structure post-processing
`dos.x`	DOS postprocessing
`epsilon.x`	optical properties calculation
`ph.x`	phonon calculation

use Matplotlib libraries from the Python programming language. Therefore, for plotting the Quantum ESPRESSO calculation data, we will give the hands-on plotting examples either in `.py` or `.ipynb` extension, which is particularly comfortable to be edited and interacted with using the JupyterLab interface (`https://jupyter.org/`). The installation of Python, Matplotlib, JupyterLab, along with other important Python libraries, is already explained in Sec. 2.2. To open JupyterLab, we can type the following command in the terminal

```
$ jupyter-lab
```

which results in opening a web browser (e.g., Chrome/Firefox/Safari) with some menus shown in Fig. 2.12.

Following Fig. 2.12, there are three most important parts of JupyterLab we should understand:

(1) The address of JupyterLab instance: This address is shown in the web browser address locator. It will almost certainly start with `localhost`, followed by 4-digit number of the port used by JupyterLab. You need not worry about the number because JupyterLab will automatically determine it.

(2) List of files and folders at which we open JupyterLab: You will be able to see the files/folders in that panel if the working directory where we execute the `jupyter lab` command already consists of some files/folders. We can also browse through the files/folders by clicking the filename/folder name there.

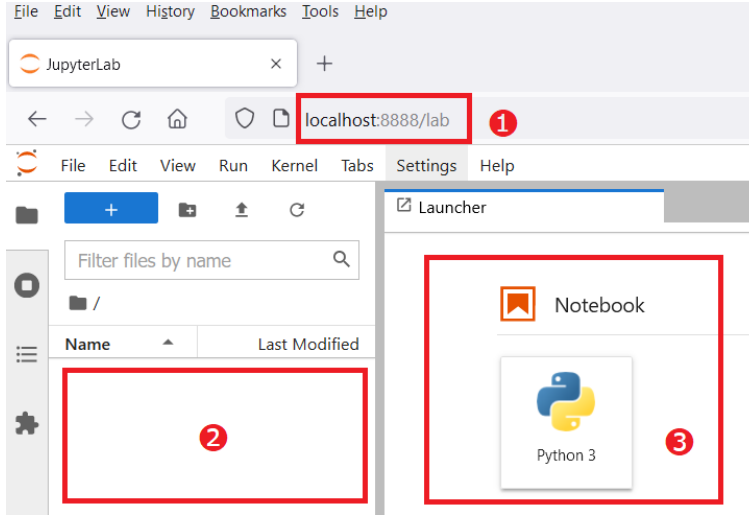

Figure 2.12 Jupyter lab interface.

(3) Jupyter Notebook: We should click the icon in order to open a Python 3 developing environment in the form of Jupyter Notebook. Then, we can save the file with the `.ipynb` or `.py` extension for the plotting purpose.

Note that some traditional Python users may prefer to directly access the Jupyter notebook from the command line by typing:

```
$ jupyter-notebook
```

2.4.3 Obtaining example files for hands-on tutorials

We provide example input/output files of Quantum ESPRESSO calculations, additional tips, and also errata of the book in the following public repository: `https://github.com/nguyen-group/QE-SSP`. The readers can use the input files from the repository to practice the Quantum ESPRESSO calculations directly on their computers. A step-by-step guide to run these files is given in

Chapter 3. To download the files from the terminal, we can use the one-line `git clone` utility:

```
$ git clone https://github.com/nguyen-group/QE-SSP.
    git
```

With all the information in this chapter, we are now ready to dive into hands-on calculations of Quantum ESPRESSO. It would be nice to read the remaining chapters and practice all the calculations accompanied by a cup of ESPRESSO coffee.