Quantum Computing
First steps with Python & Qiskit

(Dated: April 2023)

Nowadays, it is possible to address *real* quantum computers over the cloud. For this, Python is a very prominent programming language, in which several companies such as Google, IBM, Microsoft and others have developed interfaces to write quantum circuits in order to run them on quantum computers.

I. GETTING STARTED WITH PYTHON

Should you be new to Python and standard scientific packages like NumPy, SciPy, or Matplotlib, this is an opportunity for you to gain some familiarity with these tools. You can find plenty of very nice tutorials on the internet. Here is a small selection [1–3].

First, you need to have a working Python environment, for which there are two main alternatives: (i) A web-based solution accessible from your browser, or (ii) a local Python installation. As the former requires little to no configuration on your part, it may be especially appealing to newcomers to Python or programming in general. Here, however, we focus on the second solution.

For a local installation of Python, we recommend using *Anaconda* (or *Miniconda*) [4, 5]. After successful installation of a python distribution, it is best practice to work with so-called *virtual environments* that do not interfere with your global Python installation [6]. Here is what you can type into your Anaconda prompt (or terminal) to create a conda environment and install relevant modules:

```
conda create -n quantum_computing python=3  # create a minimal environment with only Python installed in it
conda activate quantum_computing  # activate the environment
conda install numpy scipy matplotlib  # install (exemplary) needed packages
conda deactivate  # deactivate the environment
```

Note how `conda activate` and `conda deactivate` is used to switch between the local and global environment.
From this point onward, you can write and run python code in three main ways: A text editor of your choosing and command line access, your favourite integrated development environment (IDE) like *PyCharm* [7], or, if you find the notebook style appealing but prefer a more local usage, *JupyterLab* [8].

II. QISKIT AND CO.

We will perform some small exercises using qiskit [9], which is a python framework for programming quantum computers and is being developed by IBM Quantum (one possible alternative is cirq [10, 11], which is being developed by Google Quantum AI; other quantum computing companies, such as Alpine Quantum Technologies, Innsbruck [12], also use Cirq and QISKit as interfaces). Qiskit comes with an extensive user guide, where you can familiarise yourself with basic and even advanced concepts of quantum computing [13].

III. INSTALL QISKIT ON YOUR LAPTOP

Here, we detail how to install qiskit on your laptop. For further information, you can follow the link to the installation page:

```
https://qiskit.org/documentation/getting_started.html
```

Requirements: Python 3.7 or more, having Anaconda is better, Jupyter works well.

Once you have installed Anaconda, you can employ the following commands to install qiskit:

```
conda activate quantum_computing  # activate your new environment
pip install qiskit # install the Qiskit package
```

If the packages were installed correctly, you can run "conda list" to see the active packages in your virtual environment. If you intend to use visualization functionality or Jupyter notebooks it is recommended to install Qiskit with the extra visualization support:

```
conda activate quantum_computing  # activate your new environment
pip install qiskit[visualization] # install the Qiskit package with visualization support
```

It is worth pointing out that if you're a zsh user (which is the default shell on newer versions of macOS), you'll need to put qiskit[visualization] in quotes:

```
pip install 'qiskit[visualization]' # install the Qiskit package with visualization support
```

Once done with your computations, you can deactivate the environment:

```
conda deactivate  # deactivate the environment
```

---

[1] *Python documentation*, URL https://www.python.org/doc/.
[2] *The python tutorial*, URL https://docs.python.org/3/tutorial/index.html.
[3] *Real python*, URL https://realpython.com/.
[4] *Anaconda individual edition*, https://www.anaconda.com/products/individual, URL https://docs.conda.io/projects/conda/en/latest/user-guide/index.html.
[5] *Miniconda*, URL https://docs.conda.io/en/latest/miniconda.html.
[6] *Conda environments*, URL https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html#managing-environments.
[7] *Pycharm*, URL https://www.jetbrains.com/pycharm/.
[8] *Jupyterlab documentation*, URL https://jupyterlab.readthedocs.io/en/stable/.
[9] *Qiskit documentation*, URL https://qiskit.org/documentation/getting_started.html.
[10] *Cirq homepage*, URL https://quantumai.google/cirq.
[11] *Cirq guide*, URL https://quantumai.google/cirq/start.
[12] *Aqt homepage*, URL https://www.aqt.eu/.
[13] *Ibm q experience: User guide*, URL https://quantum-computing.ibm.com/docs/.