Installation guide for REBOUND on win10

The open source software package REBOUND is officially only supported for macOS and Linux.
In the following, we present the necessary steps to install rebound on win10. We do not guarantee 100% success. These steps have been tested using a VM on macOS.

Installation on Linux and macOS

The installation on macOS and Linux is straight forward and simple, especially for `python`. I recommend to use a virtual environment via the venv module and install the rebound module via pip.

```
1  python3 -m venv ${HOME}/local/python3_venv
```

Activate the virtual environment

```
1  source ${HOME}/local/python3_venv/bin/activate
```

and install all important modules via pip

```
1  pip install matplotlib rebound numpy scipy
```

The installation of the C version is also straight forward

```
1  git clone http://github.com/hannorein/rebound && cd rebound/examples/shearing_sheet && make && ./rebound
```

For more information about the installation on macOS and Linux, please see the offical docs.

Installation on win10

Windows Subsystem for Linux

First, we have to install the `Windows Subsystem for Linux`. The official guide can be found here. I recommend to update to Windows 10 version 2004 Build 19041 first (check your Windows version by selecting the Windows logo key + R, type winver, select OK), since you can then install WSL2 (quoting from microsoft „WSL 2 uses the latest and greatest in virtualization technology to run a Linux kernel inside of a lightweight utility virtual machine (VM)"). Before installing the subsystem, you have to enable the „Virtual Machine Platform" feature of Win10. Open a PowerShell as Administrator(!) and run

```
1  dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

Restart your machine to complete the WSL installation and update to WSL 2 via the following command (again PowerShell with Administrator permissions)

```
1  wsl --set-default-version 2
```

Linux distribution

Now, open the microsoft store and select your favourite Linux distribution (debian!). From the distribution's page, click „Get".
The first time you launch a newly installed Linux distribution, you will be prompted for a user name and a password. This initial user will have `sudo` rights. The name can be different to your windows username.

Installation of packages

Start up your Linux distribution. The following commands assume that you have chosen a debian-based distribution (debian, ubuntu...). You can also access the file system structure from Windows, see figure 1.
We will now install the C-compiler, the python packages, the openssh server and some other packages via

```
1  sudo apt-get install gcc python3 python3-venv vim screen openssh-server gnuplot
```

The openssh-server is not mandatory. We need this to be able to open programs with a graphical user interface on Linux (such as gnuplot). If you want to visualise your data with your Windows tools, then omit this package and access the files via the network share.
Now, create your private virtual python environment

```
1  python3 -m venv ${HOME}/local/python3_venv
```

Activate the virtual environment

```
1  source ${HOME}/local/python3_venv/bin/activate
```
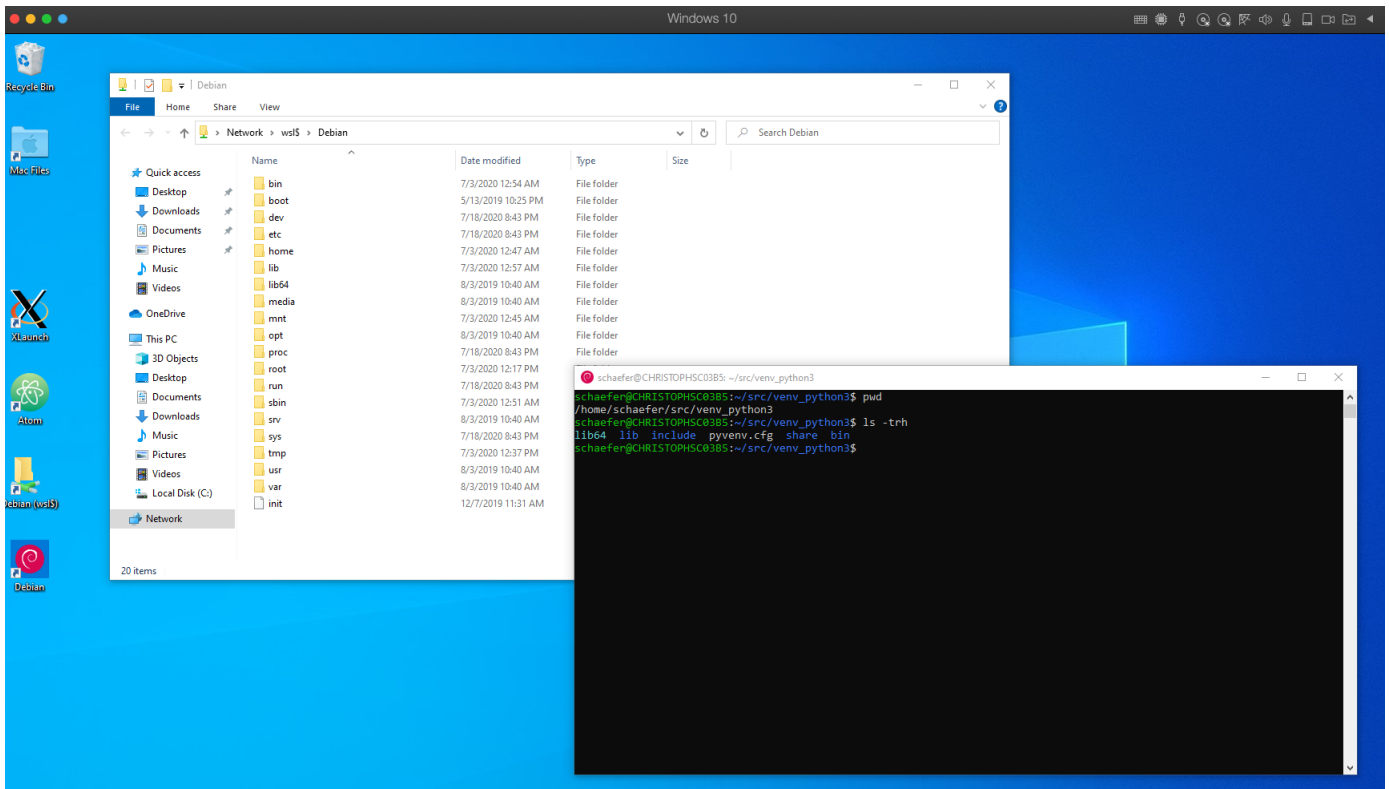
and install rebound via pip

Figure 1: Windows Subsystem for Linux running debian. The directory structure of the debian installation can be accessed from windows via the network share \\wsl$\Debian

```
1  pip install rebound
```

Additionally, install also `matplotlib`, `scipy`, and `numpy` via

```
1  pip install matplotlib scipy numpy
```

You can now test the rebound installation via

```
1  source ${HOME}/local/python3_venv/bin/activate
2  python3
3  Python 3.7.3 (dfault, Dec 20 2019, 18:57:59)
4  [GCC 8.3.0] on linux
5  Type "help", "copyright", "credits" or "license" for some information.
6  >>> import rebound
7  >>>
```

If you can import the module without any error message, kudos! Enjoy a cup of ☕ and be happy. We will continue on the windows side and install a xserver there to get a graphical connection to your linux installation.

Installation of Xserver on win10

In order to start GUI programs on your Linux, a Xserver running on win10 is required. I recommend the `VcXrv` package available via https://sourceforge.net/projects/vcxsrv/, but other Xservers or similar tools (xming, mobaXterm, ...) might do the job, too. Download the installer and execute it. at the prompt. This will tell your graphical applications to use the (default) display running on your localhost.

Test of Xserver with ipython

We will now try to plot interactively with ipython and matplotlib. Install the following package first to get a python gui

```
1  sudo apt-get install python3-tk
2  pip install ipython
```

Now start the Vc-Xsrv on windows via XLaunch. Choose „Multiple windows" in „Select display settings", choose „Start no client" in „Select how to start clients", check „Disable access control"in „Extra settings". Then switch to your Linux installation, and start ipython after setting the DISPLAY environment variable to
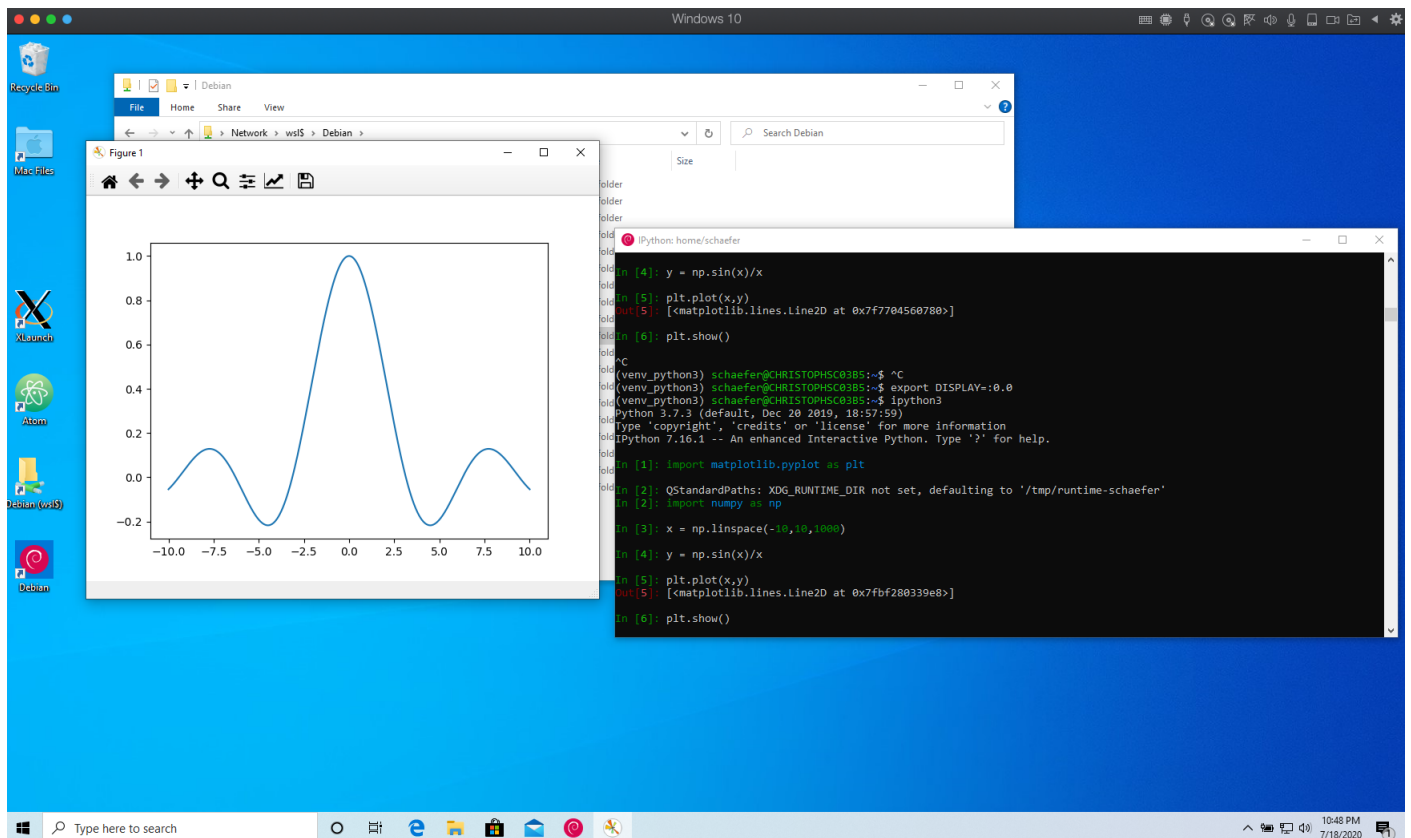
Figure 2: Interactive plotting with ipython on Linux using VcXsrv.

```
1  export DISPLAY=:0.0
```

Start ipython

```
1  ipython3
2  (...)
3  In [1]  import matplotlib.pyplot as plt
4  In [2]  import numpy as np
5  In [3]  x = np.linspace(-10,10,1000)
6  In [4]  y = np.sin(x)/x
7  In [5]  plt.plot(x,y)
8  (...)
9  In [6]  plot.show()
```

and you should get a nice plot of a sinc-function as shown in figure 2.

18 July 2020, Christoph M. Schäfer, ☒ ch.schaefer@uni-tuebingen.de, ✖ stoffelEE