

Handbook of Common Research Tools

实用科研工具指南

[Updated on 2026/02/19]

Jungeng Zhou (周俊耿)



Laboratory of Quantum Engineering
& Quantum Metrology

*Laboratory of Quantum Engineering and Quantum Metrology, School of Physics and
Astronomy, Sun Yat-Sen University (Zhuhai Campus), Zhuhai 519082, China
*Institute of Quantum Precision Measurement, State Key Laboratory of Radio Frequency
Heterogeneous Integration, Shenzhen University, Shenzhen 518060, China

*College of Physics and Optoelectronic Engineering, Shenzhen University, Shenzhen
518060, China

前言

本指南旨在为组内同学提供科研工具的上手指南及备忘查询，主要包括安装和简单的使用方法。对于大多数工具，我们将给出简单的入门使用例子，但具体应用内容及科研计算等需要自己根据课题探索。也欢迎组内同学提出宝贵意见和建议，或者一同参与并补充完善本指南。

由于本指南主要面向新手，因此会尽量使用简单易懂的步骤说明及演示截图，语句内容不够严谨之处还望谅解（亦可反馈给当前管理员更新）。

Abstract

Key words

命令行, SSH, Python, Matlab, Vscode, Mathematica

目录

| | |
|---|-----------|
| 1 引言 | 1 |
| 1.1 重要提示 | 2 |
| 2 服务器使用及管理 | 2 |
| 2.1 服务器的 SSH 连接及基本命令行操作 | 2 |
| 2.1.1 SSH 插件需求 | 2 |
| 2.1.2 PowerShell-SSH 连接及基本命令 | 3 |
| 2.1.3 设置 SSH 免密登录/密钥配对（根据个人需求决定是否需要） | 5 |
| 2.1.4 服务器与本地之间的文件传输 | 8 |
| 2.2 Windows 远程桌面连接可视化使用【最简便推荐】 | 9 |
| 2.3 服务器管理 *（非管理员可忽略） | 11 |
| 2.4 添加新用户 | 11 |
| 2.5 网卡设置 | 11 |
| 3 Python-Anaconda-Vscode 安装及使用 | 11 |
| 3.1 Anaconda 安装使用 (Windows) | 11 |
| 3.1.1 在 Windows 系统安装 Anaconda | 11 |
| 3.1.2 在 Linux 系统安装 Anaconda | 12 |
| 3.1.3 Anaconda PowerShell Prompt 使用与库函数安装 | 14 |
| 3.2 Vscode 安装与使用 | 19 |
| 3.3 Vscode 远程连接服务器使用 | 19 |
| 3.3.1 Reomte -SSH 安装及使用 | 19 |
| 3.3.2 利用 Vscode 完成密钥配对 | 20 |
| 3.3.3 Remote config 配置信息 | 21 |
| 4 Matlab 安装及使用 | 22 |
| 4.1 Matlab 本地安装及使用 | 22 |
| 4.2 Matlab 远程连接服务器使用 (基于远程桌面连接) 【推荐】 | 22 |
| 4.3 Matlab 远程连接服务器使用 (基于 PowerShell, 可后台运行) | 22 |
| 4.4 Matlab 远程连接服务器使用 (基于 MobaXterm, 窗口化) | 23 |
| 4.4.1 MobaXterm 安装 | 23 |
| 4.4.2 MobaXterm 新建 SSH 会话 | 24 |
| 4.4.3 MobaXterm 文件传输 | 26 |

| | |
|---|-----------|
| 4.4.4 MobaXterm 使用远程 Matlab 图形化界面 | 27 |
| 5 Zotero 文献管理 | 29 |
| 5.1 Zotero 安装使用 | 29 |
| 5.2 Zotero 常用功能 | 33 |
| 6 Latex 写作 (待施工) | 34 |
| 6.1 TexLive + Vscode 本地写作 | 34 |
| 6.2 Overleaf 在线协同写作 (推荐) | 34 |
| 7 ... | 34 |
| 附录 A 常用 DOS 命令 | 34 |
| 附录 B windows 端 | 34 |
| 附录 C Linux 端 | 34 |
| 附录 D 数值计算实践示例 | 34 |
| 附录 E Python 实践示例：单轴扭曲作用中压缩参数随时间的变化 | 34 |
| 附录 F Matlab 实践示例：布洛赫球的绘制 | 34 |
| 参考文献 | 34 |

表格

插图

| | |
|------------------------------|----|
| Fig. 1 安装 SSH 插件 | 3 |
| Fig. 2 windows 打开 PowerShell | 4 |
| Fig. 3 windows 设置 PowerShell | 4 |
| Fig. 4 终端 SSH 连接服务器 | 5 |
| Fig. 5 查看秘钥文件 | 6 |
| Fig. 6 生成秘钥文件 | 6 |
| Fig. 7 查看服务器上授权文件 | 7 |
| Fig. 8 复制密钥文件到服务器 | 8 |
| Fig. 9 Windows 自带远程桌面连接软件 | 10 |
| Fig. 10 Anaconda 应用列表 | 12 |

| | |
|--------------------------------------|----|
| Fig. 11 复制 Anaconda 下载链接 | 13 |
| Fig. 12 在 linux 上下载成功 | 13 |
| Fig. 13 安装 anaconda 应用 | 14 |
| Fig. 14 安装 anaconda 应用 | 14 |
| Fig. 15 安装 anaconda 应用 | 15 |
| Fig. 16 conda list & pip list | 16 |
| Fig. 17 conda clone environment | 16 |
| Fig. 18 conda env 相关命令 | 17 |
| Fig. 19 conda/pip install qutip | 18 |
| Fig. 20 Vscode remote SSH | 19 |
| Fig. 21 利用 Vscode 完成密钥配对 | 20 |
| Fig. 22 修改远程配置信息 | 21 |
| Fig. 23 安装 MobaXterm | 24 |
| Fig. 24 使用 MobaXterm 新建会话进行 SSH 协议连接 | 25 |
| Fig. 25 MobaXterm 会话列表 | 25 |
| Fig. 26 MobaXterm 文件监视器 | 26 |
| Fig. 27 MobaXterm 本地命令终端 | 26 |
| Fig. 28 打开远程 Matlab | 27 |
| Fig. 29 打开远程 Matlab | 28 |
| Fig. 30 Zotero 官网 | 29 |
| Fig. 31 Zotero 安装 | 30 |
| Fig. 32 Zotero 同步设置 | 31 |
| Fig. 33 Zotero 插件市场 | 32 |
| Fig. 34 Zotero 插件安装 | 32 |
| Fig. 35 通过 Zotero 插件市场安装其它插件 | 33 |

1. 引言

“工欲善其事，必先利其器”。

鉴于在管理服务器和探索科研工具实用中遇到的各种问题，我们总结了一些基本的笔记以提供给组内同学更高效规范的上手，希望能够帮助减少大家初次使用这些工具时的难度，从而提高效率。

另一方面，部分工具如 Zotero 文献管理和 Overleaf 的协同写作都是比较常用和方便的，因此我们也将逐渐加入本指南中方便大家查阅。但具体的使用方式，如 Latex 常用的各种命令我们不会过于详细介绍，这些有赖于个人后续实践的熟练。

本指南的初衷是介绍服务器上对各工具的使用，因此第二章会先从介绍服务器的命令行操作和基本的 SSH 连接方式开始介绍，如果暂不需要了解可以直接跳到对应工具的本地使用介绍部分。服务器管理的内容也同理，主要用于管理员自行记录备忘及后续新管理员查询。

服务器是一种高性能计算机，作为网络的节点，存储、处理网络上 80% 的数据、信息，因此也被称为网络的灵魂。也可以这样讲，服务器指一个管理资源并为用户提供服务的计算机软件，通常分为文件服务器、数据库服务器和应用程序服务器。运行以上软件的计算机或计算机系统也被称为服务器。相对于普通 PC 来说，服务器在稳定性、安全性、性能等方面都要求更高，因此 CPU、芯片组、内存、磁盘系统、网络等硬件和普通计算机有所不同，在质量与处理器数据性能上更出色。

不过服务器与普通电脑的区别也不仅仅是硬件性能指标不同，在系统方面也很不相同，一般我们电脑是使用 windows XP 或者 windows10 等系统，但服务器一般使用 windows 2000、windows 2003、windows 2008 以及 Linux 等服务器系统，内部界面与 windows xp 类似，只是里面多了一些服务器应用软件。其中 Linux 系统更复杂，多数采用 dos 命令操作。

(原文链接：<https://blog.csdn.net/hephsouidc/article/details/88566749>)

DOS 是 Disk Operating System 的缩写，即“磁盘操作系统”。DOS 主要是一种面向磁盘的系统软件，有了 DOS，我们就可以更容易理解怎么给机器下命令，不必去深入了解机器的硬件结构，也不必去死记硬背那些枯燥二进制数字的机器命令，只需通过一些接近于英语的 DOS 命令，我们就可以轻松地完成绝大多数的日常操作。

(原文链接：https://blog.csdn.net/qq_46700234/article/details/122651261

我们常用的电脑是 windos 系统，而服务器大多是 Linux 系统，因此我们需要知道一些这两种系统上的基本 DOS 命令，包括如何查看系统的型号,ip, 文件的打开等。

常用的 DOS 命令可查阅附录 [A](#)。

1.1 重要提示

此处提及一点重要的不同,Windows 系统默认的文件路径使用反斜杠 “\” 如 “C:\Users\Del” , 而 linux 系统默认的是左斜杠 “/” 如 “/home/user” . 在使用时 windows 系统对两种斜杠都能识别, 而 linux 系统只能识别默认的左斜杠方式, 在日常使用中建议路径一般都以左斜杠方式书写较好。

另一个重要的原因是在包括 Latex,python 等应用中, 反斜杠还作为某些命令 (转义字符) 使用, 在路径中使用时容易发生冲突, 比如 “\n” 表示的是换行书写, 如果路径中含有以 “n” 为首位字母的文件或者文件夹, 读取路径时就会失败。路径写为 “C:\Users\nothing” 是读取不了的, 需要用两个反斜杠取消识别冲突 “C:\Users\\nothing” 或该改为斜杠 “C:/Users/nothing” 才行.

因此, 文件在命名时最好都尽量只包括英文字母, 下划线, 阿拉伯数字这三种字符, 同时避开以下转义字符作为文件名首字母: ”\a”, ”\b”, ”\f”, ”\n”, ”\r”, ”\t”

2. 服务器使用及管理

2.1 服务器的 SSH 连接及基本命令行操作

传统的互联网通信使用明文传输数据, 内容一旦被截获就会完全暴露, 存在很多安全隐患。SSH 协议通过对网络数据进行加密和验证, 建立 SSH 客户端和 SSH 服务器之间的安全隧道, 在不安全的网络环境中为网络服务提供了安全的传输通道。简而言之, 利用 SSH 协议我们就可以将本地计算机和远程的计算机以某种方式连接起来.

2.1.1 SSH 插件需求

不是所有的系统都预先安装好了 SSH 协议插件, 因此我们需要一些满足 SSH 协议的基本插件, 根据系统版本进行如图 1所示操作进行安装:

其中 OpenSSH 客户端是连接远程是需要使用的,OpenSSH 服务端则是将本地计算机当做服务器供其他计算机远程连接使用, 如果不需要也可不安装 OpenSSH 服务端。

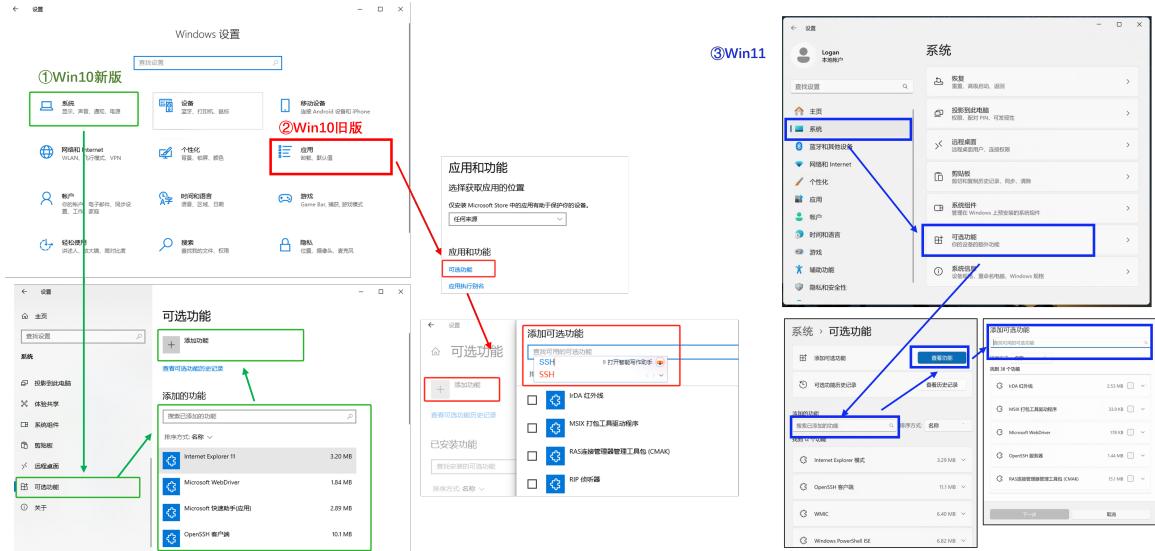


Fig. 1. 安装 SSH 插件

2.1.2 PowerShell-SSH 连接及基本命令

我们有多种方式使用 SSH 协议连接远程服务器，可以借助系统自带的 PowerShell,cmd 等终端使用 DOS 命令。Windows 系统可以直接在搜索栏直接搜索 PowerShell 或 cmd 即可，linux 和 Mac 等系统则是打开终端。

注意，windows 系统下 cmd 和 PowerShell 都是终端工具，cmd 只支持 windows 命令，而 PowerShell 部分支持 linux 命令，如 `ls path` 中的 ‘`ls`’ 命令可以查看 path 路径目录下内容。后续大部分操作基本都是基于 PowerShell 使用的，因此建议还是用 PowerShell 而非 cmd。

PowerShell 的复制粘贴等命令，一般是 `ctrl+c/v` 或者 `ctrl+shift+c/v`，可如图 3 进行设置，左图是 win10 版，右图为 win11 版。同时 PowerShell 终端有记录历史命令的功能，使用 “`↑`” 和 “`↓`” 即可查阅使用。

不同系统的使用的 DOS 会有一些区别，但在用 SSH 协议连接上是相同的命令。在终端内输入以下命令

```
> 登录 (如图 4)
ssh user@172.xx.xxx.xx
```

其中 `user` 为管理员提供的账户名，`172.xx.xxx.xx` 为服务器 ip 地址。如果连接成功则显示需要输入密码，输入即可。（注意，由于隐私保护机制，输入密码时并不会出



Fig. 2. windows 打开 PowerShell

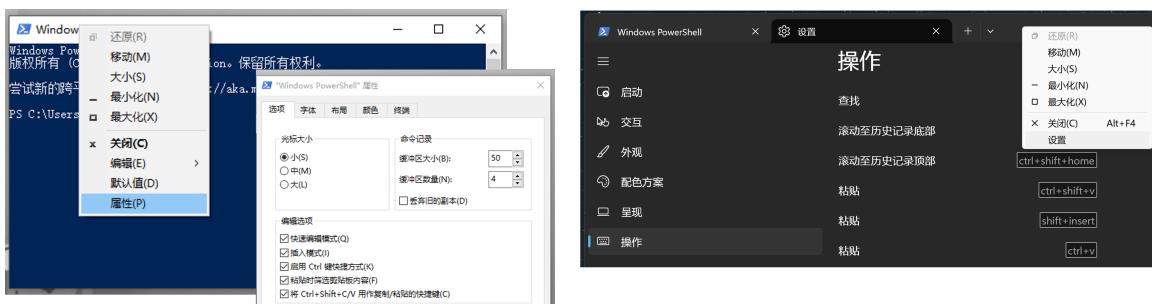


Fig. 3. windows 设置 PowerShell

The screenshot shows a Windows PowerShell window titled "选择 Windows PowerShell". The command PS C:\Users\DeII> ssh zhoujungeng@172.16. [REDACTED] is run. The output shows a password prompt, a successful login message, and a command to exit. A red box highlights the command PS C:\Users\DeII> [REDACTED] if it's PS C:\.....则表示在本地.

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\DeII> ssh zhoujungeng@172.16. [REDACTED]
zhoujungeng@172.16. [REDACTED]'s password:
Last login: Mon Sep 2 08:55:36 2024 from 172.17. [REDACTED]
(base) [zhoujungeng@localhost ~]$ 前面出现[用户名@localhost ~]表示当前在远程端,
(base) [zhoujungeng@localhost ~]$ exit 使用exit命令可退出远程, 或者打开新的powershell窗口
logout
Connection to 172.16.210.43 closed.
[PS C:\Users\DeII> ]如果是PS C:\.....则表示在本地
```

Fig. 4. 终端 SSH 连接服务器

现任何符号在屏幕上, 输入完成回车即可。)

一般首次连接时还会向你再次确认是否连接“Are you sure you want to continue connecting (yes/no)?”, 根据出现的提示输入“yes”即可。

如果出现“Connection timed out”, 则说明连接不成功, 检查账户名和 ip 地址是否有误, 或者当前是否在校园网内连接服务器。

> 修改密码: passwd

> 若出现错误 ‘WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED’ 需要重连

输入 ssh-keygen -R 172.20.10.2 进行删除操作, 之后重新登录即可

> 退出连接服务器: exit

备注: 当登录成功后

2.1.3 设置 SSH 免密登录/密钥配对 (根据个人需求决定是否需要)

如果每次登录都需要输入密码, 那无疑是很不方便的, 尤其是在需要频繁打开或者写入某些操作时, 每次都需要再次输入密码, 因此这里介绍如何配置免密登录供参考。步骤如下:



Fig. 5. 查看秘钥文件

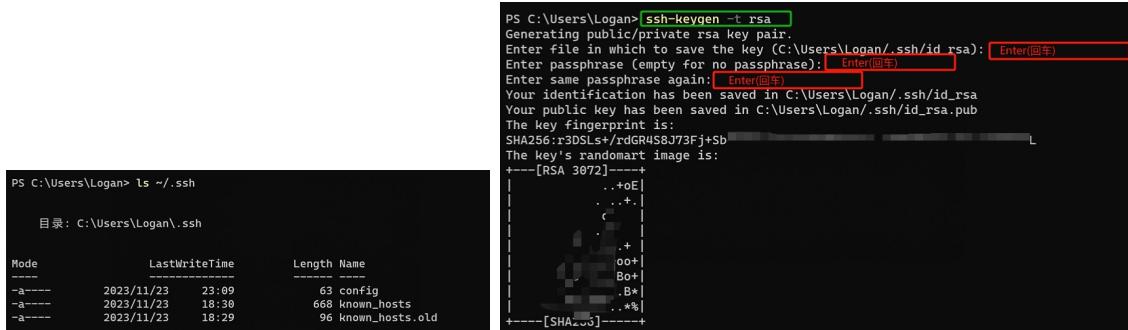


Fig. 6. 生成秘钥文件

1) 在本地，打开 Powershell 即可不用登录服务器（可重新打开一个新的 PowerShell 窗口）查看是否已经有下列两个密钥文件，可以在命令行输入 `ls ~/ssh` 或者在路径 `C:/Users/Dell/.ssh` 文件夹查看，如图 5可以看到下列两个文件

`id_rsa` 代表 SSH 密钥的私钥 (Private Key)

`id_rsa.pub` 代表 SSH 密钥的公钥 (Public Key)

2) 若无上述文件（如图 6）则输入 `ssh-keygen -t rsa` 命令生成本地密钥，此时会出现如下内容：

注意下面的步骤都不需要输入任何内容，一路直接回车（Enter），不输入内容才能实现免密，输入的任何内容则会变成新的备忘密码

'Enter file in which to save the key (C:\Users\Logan/.ssh/id_rsa):'

直接 ‘Enter/回车’

'Enter passphrase (empty for no passphrase): [Type a passphrase]'

不输入内容直接 ‘Enter/回车’

'Enter same passphrase again: [Type passphrase again]'

出现再次确认密码，再次回车确认

情况1:

```
(base) [zhoujungeng@localhost ~]$ ls ~/.ssh  
authorized_keys  
(base) [zhoujungeng@localhost ~]$ chmod 600 ~/.ssh/authorized_keys  
(base) [zhoujungeng@localhost ~]$
```

情况2:

```
[root@localhost ~]# ls ~/.ssh  
known_hosts  
[root@localhost ~]# touch ~/.ssh/authorized_keys  
[root@localhost ~]# chmod 600 ~/.ssh/authorized_keys  
[root@localhost ~]#
```

情况3:

```
[wu@localhost ~]$ ls ~/.ssh  
ls: cannot access /home/wu/.ssh: No such file or directory  
[wu@localhost ~]$ mkdir ~/.ssh  
[wu@localhost ~]$ chmod 700 ~/.ssh  
[wu@localhost ~]$ touch ~/.ssh/authorized_keys  
[wu@localhost ~]$ chmod 600 ~/.ssh/authorized_keys
```

Fig. 7. 查看服务器上授权文件

3) 在服务器, 比如用 Powershell 登录, 同样用命令 `ls ~/.ssh` 检查服务器是否有 ‘authorized_keys’ 文件, 如图 7

情况 1: 已经有 ‘authorized_keys’ 文件, 执行 ④

情况 2: 能打开路径/.ssh 但路径下无 ‘authorized_keys’ 文件, 执行 ③④

情况 3: 无路径/.ssh, 执行 ①②③④

① 创建路径文件夹 `mkdir ~/.ssh`

② 授予拥有者有读、写、执行权限 `chmod 700 ~/.ssh`

③ 创建/更新 authorized_keys 文件 `touch ~/.ssh/authorized_keys`

④ 授予拥有者有读、写权限 `chmod 600 ~/.ssh/authorized_keys`

4) 复制密钥到服务器 authorized_keys 文件中

由于需要写入密钥到服务器文件, 因此我们需要一点点 linux 系统下文件编辑的知识, 按照操作执行即可。linux 下可以用 vim 编辑器对文本进行编辑, 类似 windows 下的记事本。

(补充:如果已安装有 Vscode, 并了解如何用 Vscode 连接服务器 [见 3.3], 那么可以利用 Vscode 连接服务器后直接在“Explorer”中双击打开 authorized_keys 文件, 同样完成下列的粘贴并保存即可。)

① 如图 8(a) 用 vim 命令打开文件: `vim ~/.ssh/authorized_keys`

② 此时会看到如图 8(b) 的 vim 编辑器界面, 文件已有内容会显示在上方, 我

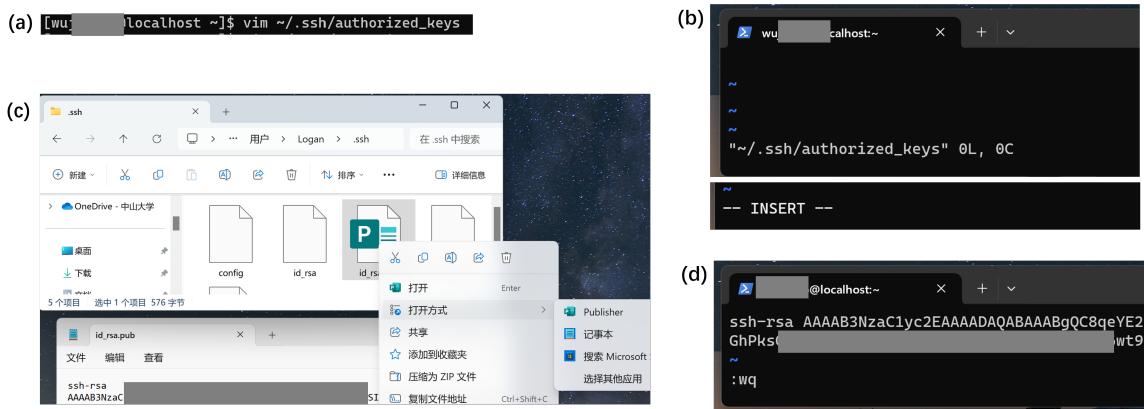


Fig. 8. 复制密钥文件到服务器

们创建的是新文件所以上方是空白显示的，此时在键盘上按下”Insert”键即可进入编辑状态输入内容。

③我们需要在这个文件进行密钥配对，即将本地公钥 id_rsa.pub 中的内容复制过来，先在本地用记事本打开 id_rsa.pub 并复制其中的内容，如图 8(c)

④回到 PowerShell 将公钥内容复制到服务器文件中（如果文件不是空白，已有其他公钥，则添加到末尾）

⑤ 复制完成后，按下”ESC”退出编辑状态，再输入 :wq 保存并退出即可，如图 8(d)

2.1.4 服务器与本地之间的文件传输

服务器与本地之间的文件传输可以在自带的 PowerShell 中用 scp 命令完成，也可以利用其他远程工具（需另外下载，如 xshell,putty,MobaXterm 等），这里作为入门我们先介绍基础的 scp 命令。

首先，scp 命令是在本地使用的，不需要登录到服务器，直接打开 PowerShell 终端即可，但在传输过程中需要输入密码确认（如果设置了免密登录就不用）。

scp 命令由三部分组成：‘命令 + 文件地址 + 目标地址’。对于任何类型的单个文件，我们都可以直接用 scp 即可，**当传输的是整个文件夹时则只需要将开头的命令改为 scp -r 即可。**

示例 (1) 从服务器复制文件到本地：

```
scp user@172.xx.xxx.xx:~/folder1/data/test.txt D:/localfolder
```

这表示用 scp 命令将 user@172.xx.xxx.xx 服务器上路径为 ~/folder1/data/test.txt 的文件复制到本地的 D:/localfolder 目录中

示例 (2) 从本地复制文件到服务器：

```
scp D:/localfolder/test.txt user@172.xx.xxx.xx:~/folder1/data
```

此时文件地址为本地，目标地址为远程服务器目录。

如果并不喜欢这种命令行的方式，我们推荐另外两种所见即所得的可视化方式传输文件：常用 Python 的可以借助 Vscode 完成连接和文件操作，参见第 3.3 节；常用 MATLAB 的可以借助 MobaXterm 完成连接和文件操作，参见第 4.4.3 节。

2.2 Windows 远程桌面连接可视化使用【最简便推荐】

【重磅更新！服务器安装桌面图形后已支持桌面化远程操作】

如图 9，使用 windows 自带的远程桌面连接，输入对应的账户密码即可直接远程操作服务器，类似于向日葵/Teamviewer/TodoDesk 等远程控制，等同于另一台电脑，并在关闭远程桌面连接软件后（注意不是在服务器上执行休眠关机等操作）不影响服务器后台运行。

其它注意事项：

- ① 若出现连接失败，进行以下更改后重试：
“显示” - “选择远程会话的颜色深度” - “增强色（16 位）”
- ② matlab 已安装全用户可用，若桌面无图标在左上角 “Applications” - “Others” - “Matlab” 可找到，可直接将图标拖动到桌面生成快捷方式
- ③ 文件（包括程序代码）必须在自己的个人文件夹路径下使用和保存，否则没有读写权限
- ④ 需要传输文件请参考[2.1.4](#) 或者 [4.4.3](#)

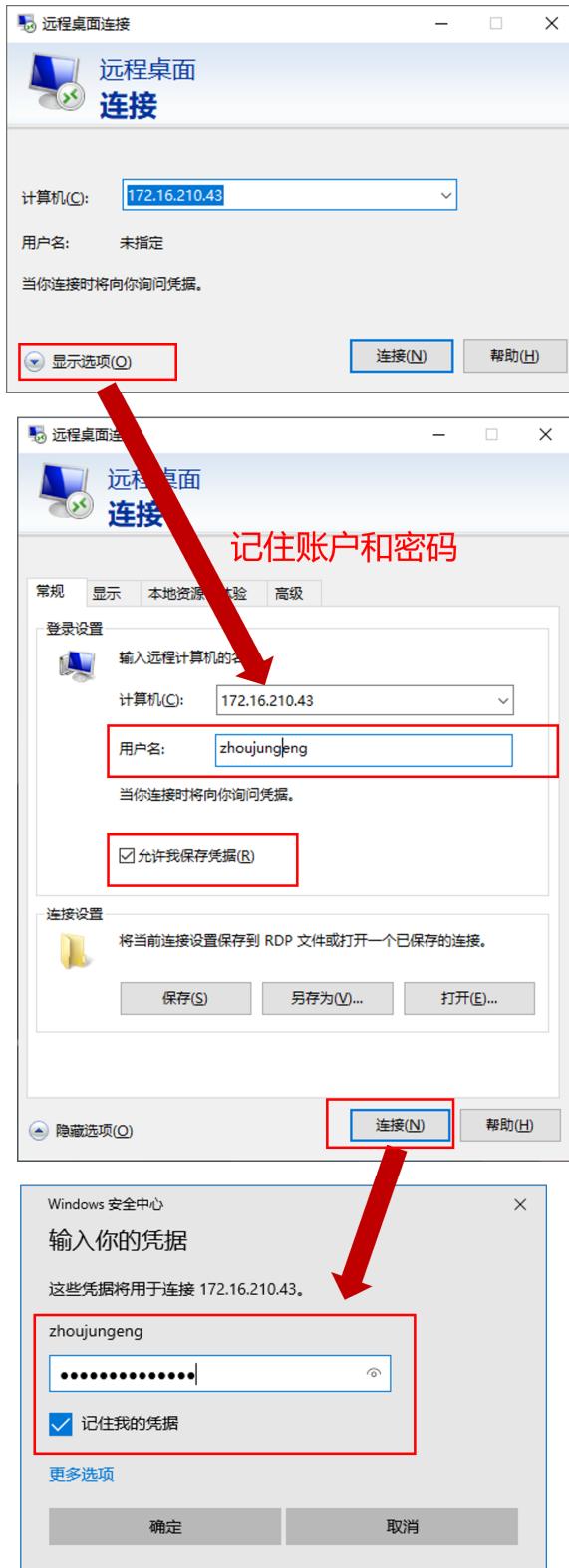


Fig. 9. Windows 自带远程桌面连接软件

2.3 服务器管理 * (非管理员可忽略)

2.4 添加新用户

添加名为“NewUserName”的新账号

```
sudo adduser NewUserName
```

```
chown NewUserName:NewUserName -R /home/NewUserName
```

```
sudo passwd NewUserName
```

2.5 网卡设置

查看网卡状态: `ifconfig`

```
关闭指定网卡”docker0”: ifconfig docker0 down
```

3. Python-Anaconda-Vscode 安装及使用

3.1 Anaconda 安装使用 (Windows)

Anaconda 指的是一个开源的 Python 发行版本, 其包含了 conda、Python 等 180 多个科学包及其依赖项。在使用体验上比直接单独用 Python 自身的 IDLE 编译器有更好的体验, 因此建议直接下载 Anaconda 即可。

下载地址: <https://www.anaconda.com/download>

3.1.1 在 Windows 系统安装 Anaconda

安装过程可以参考: <https://zhuanlan.zhihu.com/p/123188004>

安装完成后可以在应用中看到其还包含了 Anaconda Navigator, Spyder, Jupyter Notebook, Anaconda PowerShell Prompt 等应用。这些应用可以在 Anaconda Navigator 中打开, 也可以直接打开。Spyder 是一个本地的编译器, 如果我们对 python 的编译要求很简单, 那么直接用它就可以了。Jupyter Notebook 则是将 markdown 和 python 代码行以笔记模块形式结合的应用, 使得我们在写代码时能够分块分行并注释, 能够极大提高编程体验, 强烈建议使用。

当然,Anaconda 内置的 Jupyter Notebook 的功能相对来说没有那么齐全, 如果要有更好的体验还需要自己安装一些插件, 而如果我们安装另一款软件 Vscode 则可

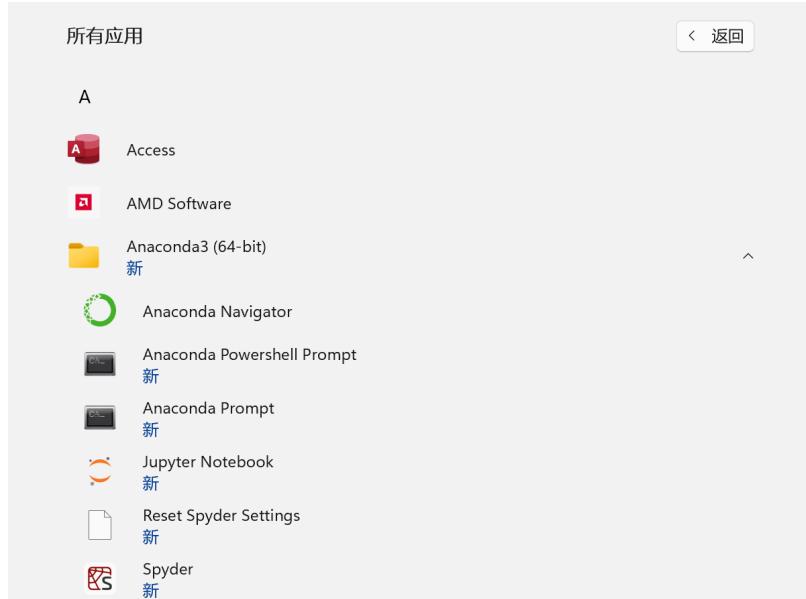


Fig. 10. Anaconda 应用列表

以更方便的使用 JupyterNotebook, 包括直接查看变量的格式和内容等。同时, Vscode 还包括了 Latex, Julia, 远程连接服务器等功能, 因此也十分建议安装。

Anaconda PowerShell Prompt 与 windows 的 PowerShell 是相同的, 只不过是激活了 Anaconda 环境。如果要额外安装一些 Python 的库函数, 如 qutip 和 tensorflow 等, 就需要在 Anaconda PowerShell Prompt 中使用命令行进行安装。(部分通用性更高的库函数也可以在 Anaconda Navigator 中进行安装。)

3.1.2 在 Linux 系统安装 Anaconda

在 Linux 系统如远程服务器上使用 python, 也可以通过安装 Anaconda 的方式, 安装方式与 Windows 系统稍有不同。对于需要在服务器上远程使用 python 的老师和同学, 这一步也是必需的。

在 Linux 系统中打开终端 (在服务器上使用则通过 PowerShell 命令行登录服务器, 登录过程详见 2.1 节)。

安装过程参考https://blog.csdn.net/G_C_H/article/details/133553961

第一步, 下载安装文件

```
wget -c https://repo.anaconda.com/archive/Anaconda3-2023.09-0-Linux-x86_64.sh
```

其中, wget 是下载工具命令, -c 是工具参数, 最后的是下载链接。下载链接可

以在官网得到，方法如图 11 所示。

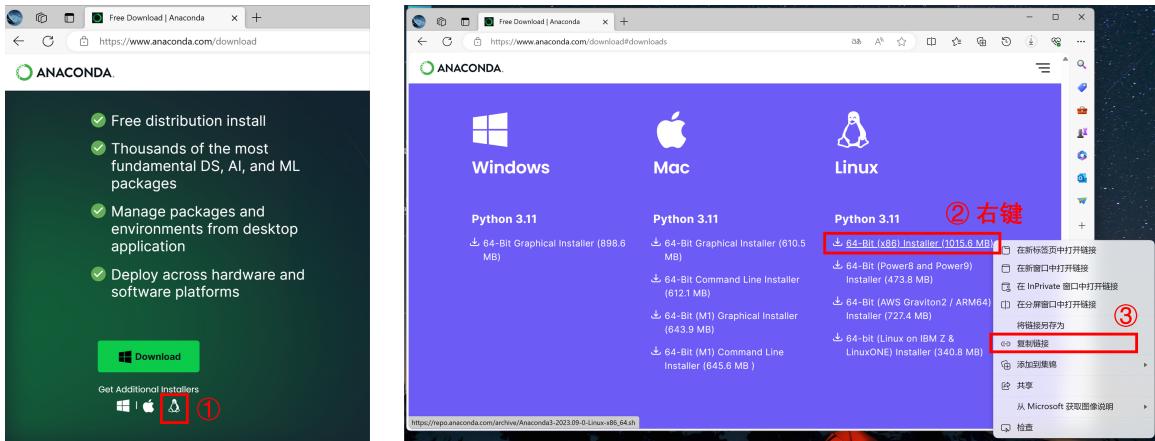


Fig. 11. 复制 Anaconda 下载链接

复制该链接并执行下载命令，下载成功如图 12 所示。

```
[root@localhost ~]# wget -c https://repo.anaconda.com/archive/Anaconda3-2023.09-0-Linux-x86_64.sh
--2024-01-12 21:01:44-- https://repo.anaconda.com/archive/Anaconda3-2023.09-0-Linux-x86_64.sh
Resolving repo.anaconda.com (repo.anaconda.com)... 2606:4700::6810:8203, 2606:4700::6810:8303, 104.16.131.3, ...
Connecting to repo.anaconda.com (repo.anaconda.com)|2606:4700::6810:8203|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1153404010 (1.1G) [application/x-sh]
Saving to: ‘Anaconda3-2023.09-0-Linux-x86_64.sh’

100%[=====>] 1,153,404,010 1.68MB/s   in 11m 8s

2024-01-12 21:12:52 (1.65 MB/s) - ‘Anaconda3-2023.09-0-Linux-x86_64.sh’ saved [1153404010/1153404010]
```

Fig. 12. 在 linux 上下载成功

第二步，解压安装文件，注意文件名（如 Anaconda3-2023.09-0-Linux-x86_64.sh）要与下载的相同

```
bash Anaconda3-2023.09-0-Linux-x86_64.sh
```

此时会出现安装协议，**只按两三下回车**简单阅读即可。后续安装协议内容较多，可输入 q（在英文输入法下）跳过后续文件内容，进入安装。**注意此时不要盲目回车，要看出现内容的提示：因为回车是默认选择“no”，需要输入“yes”再回车才是同意协议，并会出现安装路径的确认，如图 14。**

这里一般使用默认路径，继续**只按一下回车**确认即可开始安装。

安装完成后如图 15 出现 You can undo this by running conda init --reverse \$SHELL? [yes|no]: 的提示后同样需要输入 yes，**最好不要按回车使用默认的 no**。

如果不选 No，执行以下两步命令手动设置环境变量即可

```
export PATH=/home/(your_user_name)/anaconda3/bin:$PATH
```

```
[liuwenjie@localhost ~]$ bash Anaconda3-2023.09-0-Linux-x86_64.sh

Welcome to Anaconda3 2023.09-0

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
=====
End User License Agreement - Anaconda Distribution
=====

Copyright 2015-2023, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Distribution (which was formerly known as Anaconda Individual Edition).

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusive, non-transferable license to:

* Install and use the Anaconda Distribution (which was formerly known as Anaconda Individual Edition),
* Modify and create derivative works of sample source code delivered in Anaconda Distribution from Anaconda's repository, and;
* Redistribute code files in source (if provided to you by Anaconda as source) and binary forms, with or without modification subject to the requirements set forth below, and;

Anaconda may, at its option, make available patches, workarounds or other updates to Anaconda Distribution. Unless the updates are provided with their separate governing terms, they are deemed part of Anaconda Distribution licensed to you as provided in this Agreement. This Agreement does not entitle you to any support for Anaconda Distribution.

Anaconda reserves all rights not expressly granted to you in this Agreement.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following dis
--More--
```

Fig. 13. 安装 anaconda 应用

```
Do you accept the license terms? [yes|no]
[no] >>> yes

Anaconda3 will now be installed into this location:
/home/[REDACTED]/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/[REDACTED]/anaconda3] >>> [REDACTED] 回车(Enter)
```

Fig. 14. 安装 anaconda 应用

source .bashrc

3.1.3 Anaconda PowerShell Prompt 使用与库函数安装

在 Anaconda PowerShell Prompt 我们可以用命令行额外安装一些库函数，主要的命令行有 conda 类与 pip 类。

在这之前，我们先介绍环境配置，以及如何查看该环境下已有的库函数。

(1) 默认环境

在应用中打开 Anaconda PowerShell Prompt（之后简称为 Prompt）会显示环

```

[~/home/anaconda3] >>>
PREFIX=/home/anaconda3
Unpacking payload ...

Installing base environment...

Downloading and Extracting Packages

Downloading and Extracting Packages

Preparing transaction: done
Executing transaction: \

    Installed package of scikit-learn can be accelerated using scikit-learn-intelex.
    More details are available here: https://intel.github.io/scikit-learn-intelex

For example:

$ conda install scikit-learn-intelex
$ python -m sklearnex my_application.py


done
installation finished.
Do you wish to update your shell profile to automatically initialize conda?
This will activate conda on startup and change the command prompt when activated.
If you'd prefer that conda's base environment not be activated on startup,
run the following command when conda is activated:

conda config --set auto_activate_base false

You can undo this by running `conda init --reverse $SHELL`? [yes|no]
[no] >>>

```

Fig. 15. 安装 anaconda 应用

境与路径，界面与 windows PowerShell 一致，区别在于路径前有标注当前的 conda 环境，默认环境名称为 base。

在 power shell 中显示的是 PS C:\Users\Logan>

在 Prompt 中显示的是 (base) PS C:\Users\Logan>

(2) 切换路径

切换同盘工作路径 (change directory): `cd path`, 如 `cd \myfolder\mysubfolder`

如果是切换不同盘 (如 D 盘) 工作路径，可以输入 `d:` 先切换盘再进行同盘路径切换即可。(注意 `d:` 命令切换盘符时一般会直接切换到上一次在该盘符的最后使用路径，而不是该盘的根目录。)

返回上一级目录 `cd..`

返回根目录 `cd\`

(3) 在 Prompt 中查看 conda 库的版本

方法一: `conda -version`

方法二: `conda -V`

| (base) PS C:\Users\Logan> conda list | | | |
|--------------------------------------|------------|------------------|---------|
| Package | Version | Build | Channel |
| _anaconda_depends | 2023.09 | py311_mkl_1 | |
| abseil-cpp | 20211102.0 | hd77b12b_0 | |
| aibotocore | 2.5.0 | py311haa95532_0 | |
| aiofiles | 22.1.0 | py311haa95532_0 | |
| aiohttp | 3.8.5 | py311hb2bbff1b_0 | |

| (base) PS C:\Users\Logan> pip list | |
|------------------------------------|---------|
| Package | Version |
| aiobotocore | 2.5.0 |
| aiofiles | 22.1.0 |
| aiohttp | 3.8.5 |
| aiohttp | 3.8.5 |
| aioitertools | 0.7.1 |
| aiosignal | 1.2.0 |
| iosqlite | 0.18.0 |

Fig. 16. conda list & pip list

| | |
|--|--|
| (base) PS C:\Users\Logan> conda create -n env1 --clone base | webencodings 0.5.1 |
| Source: C:\ProgramData\anaconda3 | websocket-client 0.58.0 |
| Destination: C:\Users\Logan\.conda\envs\env1 | Werkzeug 2.2.3 |
| The following packages cannot be cloned out of the root environment: | whatthepatch 1.0.2 |
| - defaults/win-64::conda-23.7.4-py311haa95532_0 | wheel 0.38.4 |
| - defaults/win-64::anaconda-anon-usage-0.4.2-py311hfc23b7f_0 | widgetsnbextension 4.0.5 |
| - defaults/win-64::anaconda-client-1.12.1-py311haa95532_0 | win-inet-pton 1.1.0 |
| - defaults/win-64::anaconda-navigator-2.5.0-py311haa95532_0 | wrapt 1.14.1 |
| - defaults/win-64::anaconda-project-0.11.1-py311haa95532_0 | xarray 2023.6.0 |
| - defaults/win-64::conda-build-3.26.1-py311haa95532_0 | xlwings 0.29.1 |
| - defaults/win-64::conda-index-0.3.0-py311haa95532_0 | xxhash 2.0.2 |
| - defaults/win-64::conda-libmamba-solver-23.7.0-py311haa95532_0 | xyzservices 2022.9.0 |
| - defaults/win-64::conda-repo-cli-1.0.75-py311haa95532_0 | y-py 0.5.9 |
| - defaults/noarch::conda-token-0.4.0-pyhd3eb1b0_0 | yapf 0.31.0 |
| - defaults/win-64::console_shortcut-0.1.1-4 | yarl 1.8.1 |
| - defaults/win-64::navigator-updater-0.4.0-py311haa95532_1 | ypy-websocket 0.8.2 |
| - defaults/win-64::powershell_shortcut-0.0.1-3 | zict 2.2.0 |
| Packages: 476 | zip 3.11.0 |
| Files: 5 | done |
| Downloading and Extracting Packages | # |
| Downloading and Extracting Packages | # To activate this environment, use |
| Preparing transaction: done | # |
| Verifying transaction: done | # \$ conda activate env1 |
| Executing transaction: / | # To deactivate an active environment, use |
| | # \$ conda deactivate |

(a)

(b)

Fig. 17. conda clone environment

返回的是版本号 conda 23.7.4

(4) 在 Prompt 中查看该环境下的库函数

方法一: `conda list`

方法二: `pip list`

如图 16 通过上述两种方式我们都可以看到该环境下的库函数名称及版本，包括 python 的版本。方法一还会额外显示通道信息 (Build Channel)。

(5) conda 与环境设置

在实际应用中，我们需要额外安装一些库函数，但某些库函数之间可能会相互冲突，因此可以通过创建虚拟环境的方式避免或者备份。如果对于 Python 的版本没有要求，这里推荐直接复制 base 环境作为新的工作环境。

复制环境的命令为: `conda create -n name -clone base`

```
(base) PS C:\Users\Logan> conda env list
# conda environments:
#
#                                     *
base          C:\ProgramData\anaconda3
env1          C:\Users\Logan\.conda\envs\env1

(base) PS C:\Users\Logan> conda activate env1
(env1) PS C:\Users\Logan> conda deactivate
(base) PS C:\Users\Logan> conda deactivate
PS C:\Users\Logan> conda activate base
(base) PS C:\Users\Logan>
```

Fig. 18. conda env 相关命令

这个命令意思是复制 ‘base’ 环境并将新环境命名为 ‘name’，可以根据需求修改为自己需要的环境名。如图 17 我们将新环境命名为 ‘env1’，可以看到其安装执行过程。环境复制或者创建完会都提示两个基本的激活和退出环境命令，如图 17(b).

激活（切换）到 ‘name’ 环境: `conda activate name`

退出当前环境 `conda deactivate`

当然我们也可以查看有哪些环境可供使用: `conda env list`

如图 18，我们可以看到由于前面已经复制了一个环境，在使用 `conda env list` 时会看到 ‘base’ 和 ‘env1’ 两个环境。此时我们再通过 `conda activate env1` 可以切换到 ‘env1’ 环境中，表现为括号中的环境名发生改变。再通过 `conda deactivate` 可以退出回到默认环境 ‘base’。如果是再默认环境 ‘base’ 下执行退出环境，可以看到前面的括号消失了，但 `conda` 命令仍然可以执行。

若不通过复制，而是直接创建新环境: `conda create -n name python=x.x`

(6) 安装额外的库函数

安装库函数命令 `conda install package`，其中 `package` 是对应的库函数名，如 `qutip`.

注意，先确认当前的环境是否是需要安装新函数的环境，否则的话先进行切换环境。

由于网络通道设置，可能存在无法安装的情况 [如图 19(a)]，这时候可以尝试另一种方法 `pip install package`[如图 19(b)]，如果还是安装失败我们还可以设置国内镜像通道。

(a)

```
(base) PS C:\Users\Logan> conda activate env1
(env1) PS C:\Users\Logan> conda install qutip
Retrieving notices: ...working... done
Collecting package metadata (current_repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.

PackagesNotFoundError: The following packages are not available from current channels:

- qutip

Current channels:

- https://repo.anaconda.com/pkgs/main/win-64
- https://repo.anaconda.com/pkgs/main/noarch
- https://repo.anaconda.com/pkgs/r/win-64
- https://repo.anaconda.com/pkgs/r/noarch
- https://repo.anaconda.com/pkgs/msys2/win-64
- https://repo.anaconda.com/pkgs/msys2/noarch

To search for alternate channels that may provide the conda package you're
looking for, navigate to

    https://anaconda.org

and use the search bar at the top of the page.
```

(b)

```
(env1) PS C:\Users\Logan> pip install qutip
Collecting qutip
  Obtaining dependency information for qutip from https://files.pythonhosted.org/packages/10/cd/837647e62ab82a6ff4b2bb2bf22ef4b9c70ada215250c17fb18802bc5bfe/qutip-4.7.3-cp311-cp311-win_amd64.whl.metadata
    Downloading qutip-4.7.3-cp311-cp311-win_amd64.whl.metadata (7.8 kB)
Requirement already satisfied: numpy>=1.16.6 in c:\users\logan\.conda\envs\env1\lib\site-packages (from qutip) (1.24.3)
Requirement already satisfied: scipy>=1.0 in c:\users\logan\.conda\envs\env1\lib\site-packages (from qutip) (1.11.1)
Requirement already satisfied: packaging in c:\users\logan\.conda\envs\env1\lib\site-packages (from qutip) (23.1)
  Downloading qutip-4.7.3-cp311-cp311-win_amd64.whl (5.4 MB)
[██████████] 0.7/5.4 MB 23.4 kB/s eta 0:03:21
[██████████] 0.8/5.4 MB 23.1 kB/s eta 0:03:21
[██████████] 0.8/5.4 MB 24.3 kB/s eta 0:03:09
[██████████] 5.4/5.4 MB 23.0 kB/s eta 0:00:00
Installing collected packages: qutip
Successfully installed qutip-4.7.3
```

Fig. 19. conda/pip install qutip

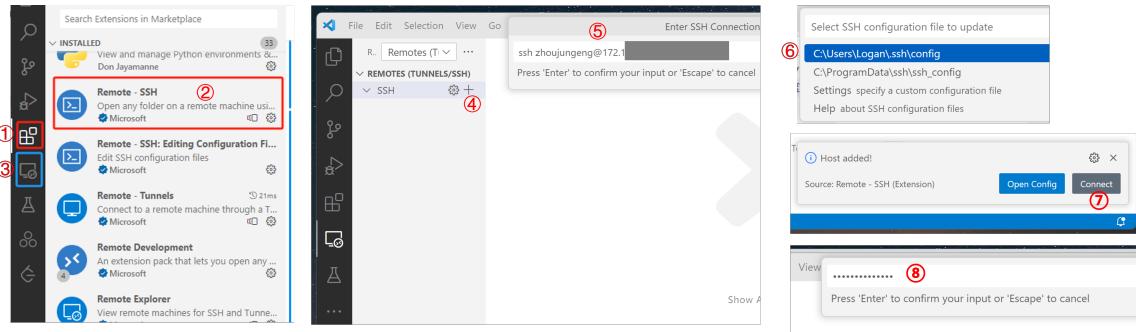


Fig. 20. Vscode remote SSH

3.2 Vscode 安装与使用

Vscode 是一款非常强大的软件

Vscode 安装步骤教程: <https://blog.csdn.net/msdcp/article/details/127033151>

3.3 Vscode 远程连接服务器使用

3.3.1 Reomte -SSH 安装及使用

像使用 jupyterNotebook 一样, 远程连接也需要特殊的插件, 在拓展中找到“Reomte -SSH”并安装即可, 安装完成后左侧菜单栏会出现一个 Remote 功能模块。安装和连接过程如图 20 步骤①至⑦所示

步骤① 打开插件商店

步骤② 搜索“Reomte -SSH”并安装, 其余附加插件一般会自动一同安装

步骤③ 安装完成后左侧菜单栏会出现一个 Remote 功能模块, 点击即可开始连接

步骤④ 点击 SSH 后的 + 号开始进行连接

步骤⑤和在 PowerShell 中进行连接相同, 输入 user@ip 的用户名和 ip 地址,

(注意, 如果出现选择服务器系统类型, 根据远程服务器进行选择即可, 如组内的服务器是 linux 系统我们便选择 linux)

步骤⑥是将信息更新到配置文件 ‘config’ 中, 后续可以在里面自己定义别名或者更改其他配置信息。

步骤⑦会在右下角出现, 选择“Connect”进行连接即可, 配置信息可以后续再修改

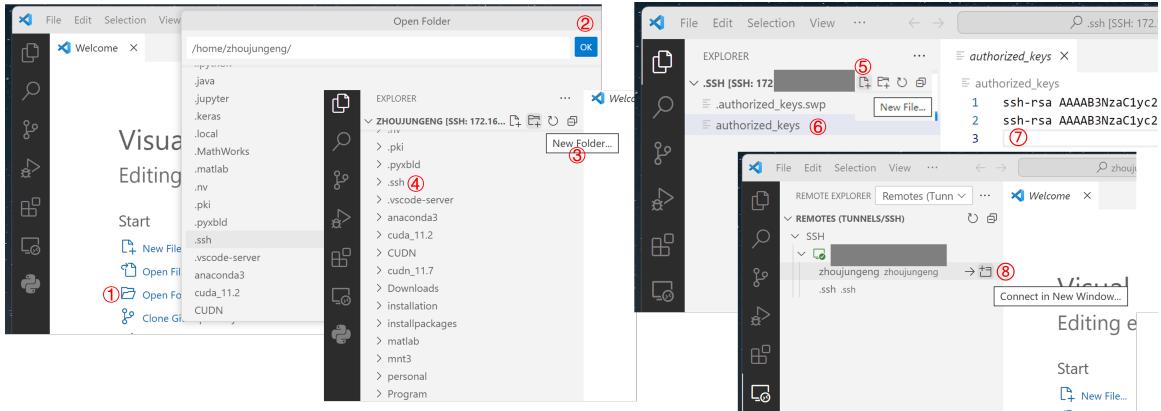


Fig. 21. 利用 Vscode 完成密钥配对

步骤⑧与 PowerShell 一样输入密码即可，不同的是这里可以看到密码输入过程而 PowerShell 不会显示任何符号。同样在输入完成后进行回车确认即可。完成上述操作后会出现欢迎界面，代表连接成功。

3.3.2 利用 Vscode 完成密钥配对

下面我们补充如何利用 Vscode 快速配置免密登录，原理同2.1.3，不同之处在于我们可以用 Vscode 直观查看/创建/编辑/保存文件。

同样查看是否有.ssh 文件夹和 authorized_keys 文件，如图 21可以直接打开目录，浏览列表选择文件夹/文件。(此时通常需要再输入密码才能打开文件夹，这也是为什么我们需要免密配对，不然每打开或编辑一次都需要输入密码。)

若没有看到.ssh 文件夹，则创建文件夹，如果没有 authorized_keys 文件则创建该文件。注意在创建后需要赋予权限（注意是在远程终端，如用 Powershell 登录服务器）。

- ② 授予拥有者有读、写、执行权限 `chmod 700 ~/.ssh`
- ③ 创建/更新 authorized_keys 文件 `touch ~/.ssh/authorized_keys`
- ④ 授予拥有者有读、写权限 `chmod 600 ~/.ssh/authorized_keys`

接下来，只需要同图 8(c) 一样从本地复制密钥信息到 authorized_keys 文件的末尾并保存即可。【关于如何创建本地密钥，请详细查看2.1.3步骤 1) 2)；需要复制的信息见图 8(c)；复制后粘贴到 authorized_keys 文件中，见图 21⑦】

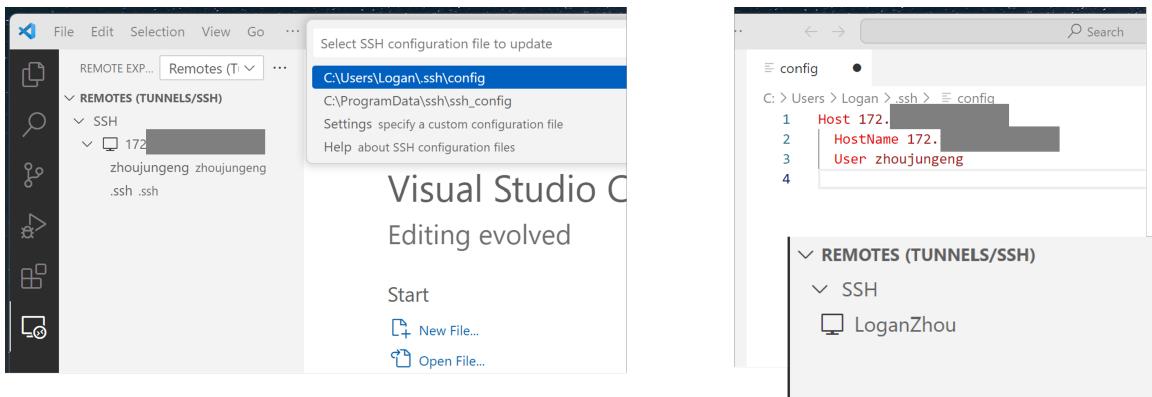


Fig. 22. 修改远程配置信息

完成图 21 步骤①至⑦后，可以关掉所有窗口再次打开尝试登录，点击⑧连接此时会发现不再需要输入密码了。

3.3.3 Remote config 配置信息

在前面我们提到了 config 配置文件，这里稍微补充一下。如图 22，我们可以打开配置文件查看，其中 Host 是配置名称，Hostname 是服务器 ip 地址，User 是账户名。后两者对于连接是确定的了，而 Host 的配置名称是可以自定义起名的，比如将其改为 ‘LoganZhou’ 则以后打开 remote 时该配置显示的都是此名称而非原来默认的 ‘172.xx.xxx.xx’ 的。

除了这三个信息外，有时候还可能还会有一行记录端口信息，如 ‘Port 22’，是否添加取决于服务器开放的端口，一般已经能够连接成功的话则可先忽略。

至此，Vscode 的远程连接已完成。利用 Vscode 远程连接的好处是其图形化界面与本地使用没有区别，只要在对应路径下创建 Python 文件或者 JupyterNotebook 文件即可运行。同时对于文本类文件，也可以直接打开进行编辑，这点在复制密钥时已体现其方便。

4. Matlab 安裝及使用

4.1 Matlab 本地安装及使用

步骤详见信息技术服务中心提供的指南:<https://software.sysu.edu.cn/matlabhome>

4.2 Matlab 远程连接服务器使用 (基于远程桌面连接) 【推荐】

参考 [2.2](#)

4.3 Matlab 远程连接服务器使用 (基于 PowerShell, 可后台运行)

利用 PowerShell 登录到服务器, 参见第 [2.1.2](#)节

```
> 登录 (如图 4) ssh user@172.xx.xxx.xx
```

切换到需要运行的文件目录 /home/username/foldername,

```
cd /home/username/foldername
```

确认一下文件是否在该路径, 可以用 `ls -la` 查看

若无需要的文件, 可参考第 [2.1.4](#)节将文件传输到服务器。若有文件则按照下一步操作运行 Matlab 文件 filename.m。

```
> 后台运行 Matlab
```

```
nohup matlab -nosplash -nodesktop < filename.m 1>running.log 2>running.err
```

```
&
```

符号 `<` 表示标准输入, 及如果有 ‘`-nosplash -nodesktop`’ 这些就需要这个符号。`‘1>running.log’` 就是把 matlab 标准屏幕输出到文件 ‘running.log’ 中, 同理 ‘`2>running.err`’ 就是把 matlab 运行时的错误写入 ‘running.err’ 文件中。‘`&`’ 表示后台运行, 这样就可以关闭终端, Matlab 还会继续运行。

运行上述命令之后会显示一个进程号，这个进程号可以用来查看 Matlab 是否在运行，或者用于中止该进程。中止某个进程 pidnumber 的命令为：`kill -9 pidnumber`，进程号也可以通过 `ps -ef | grep matlab` 查看。

中止 MATLAB 所有进程的命令为：`killall MATLAB`。

运行过程中原本打印的输出信息会被重定向到 `running.log` 文件中，如果需要查看 Matlab 运行的输出信息，可以通过 `cat` 命令查看 `running.log` 文件：`cat running.log`。

4.4 Matlab 远程连接服务器使用 (基于 MobaXterm, 窗口化)

使用 PowerShell 等命令行方式可以远程使用 Matlab，但使用体验较差，没有利用到 Matlab 可视化窗口的便捷性。因此，我们在这里介绍如何远程使用 MatLab 的。

4.4.1 MobaXterm 安装

与 Vscode 之于 Python 一样，我们也可以使用图形化窗口实现对 MATLAB 的远程操作。

在这里我们借助的工具是 **MobaXterm**，MobaXterm 是一个综合性的远程连接工具，支持 SSH、X11、RDP、VNC、FTP、SFTP 等多种协议，同时还集成了 X server，可以在 windows 下运行 Linux 程序。简而言之，MobaXterm 可以视为一个综合性的终端工具，相当于是具有更多功能的 PowerShell。因此在 Powershell 中可以实现的功能，MobaXterm 也可以实现，包括之前的 SSH 连接，文件传输等，只需要在 MobaXterm 打开的终端进行相同的输入即可。

MobaXterm 的安装使用见图 23，如遇到问题，可参考其它安装指南 https://blog.csdn.net/qq_44074697/article/details/118544904

(除 MobaXterm 之外，还可以借助于 Putty 和 Xshell 等，但步骤相对更复杂一些。由于 MobaXterm 没有 Mac 版，因此对于 Mac 系统来说只能使用这些方法。)



Fig. 23. 安装 MobaXterm

4.4.2 MobaXterm 新建 SSH 会话

MobaXterm 同样支持 SSH 协议对远程服务器进行连接，连接方式与 PowerShell 类似。

通过新建会话的方式，我们可以将 ip 地址和用户名记录下来，之后使用的时候

只需要点击对应账户即可，不需要重复输入这些信息。

新建会话的方式如图 24 所示，连接后的界面与前文 [2.1.2] 使用 PowerShell 类似。

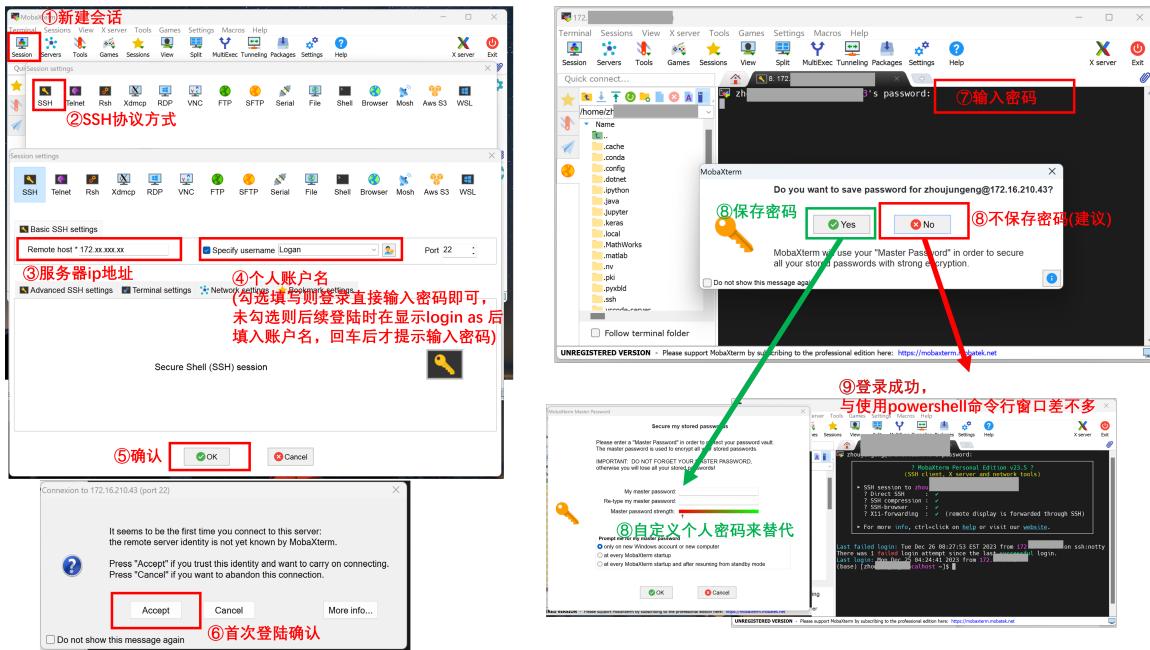


Fig. 24. 使用 MobaXterm 新建会话进行 SSH 协议连接

以上是新建会话进行连接，如果已经新建过了，之后只需要双击会话列表中对应的会话或右键选择‘Execute’即可进行连接，如图 25

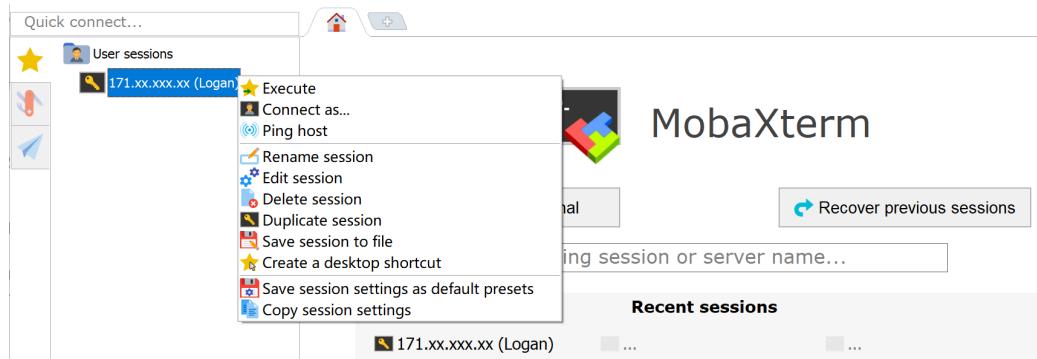


Fig. 25. MobaXterm 会话列表

4.4.3 MobaXterm 文件传输

这里补充利用 MobaXterm 的可视化文件传输方法，在我们每次登录远程成功后，在左侧都会出现如图 26 的文件监视器栏，可以使用红色小方框内的这些工具栏图标直接上传和下载文件，较为方便。

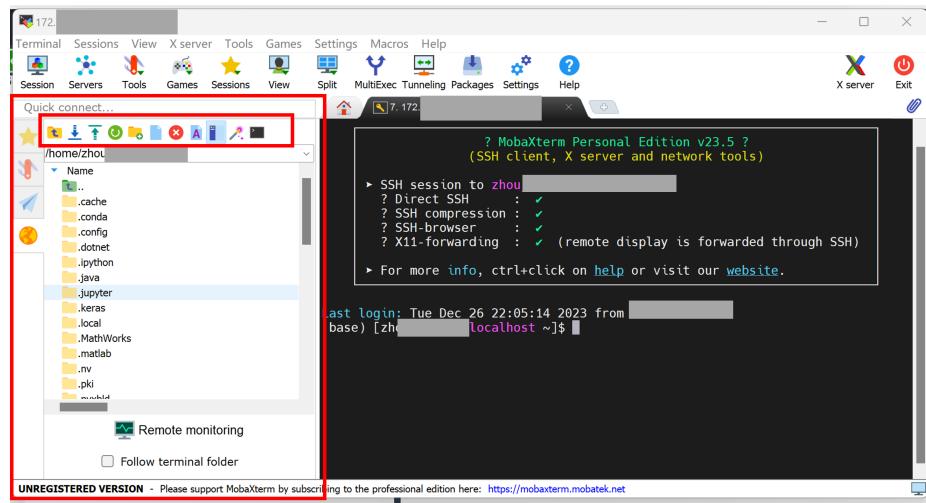


Fig. 26. MobaXterm 文件监视器

不过该方法存在一个缺陷，就是不能在远程端将一个文件（夹）复制到另一个文件（夹）中。

命令行方式进行文件传输可以参照第 2.1.4 节中给出的方法，这些命令也可以在 MobaXterm 中执行。命令行的方法需要在本地终端使用，不需要登录到服务器，MobaXterm 的本地终端打开方法如图 27 所示。(直接使用 PowerShell 亦可，MobaXterm 算是一种高级终端，使用也与 PowerShell 一致，这里不再赘述。)

如果熟悉 Vscode，亦可以参照第 3.3 节使用 vscode-remote。

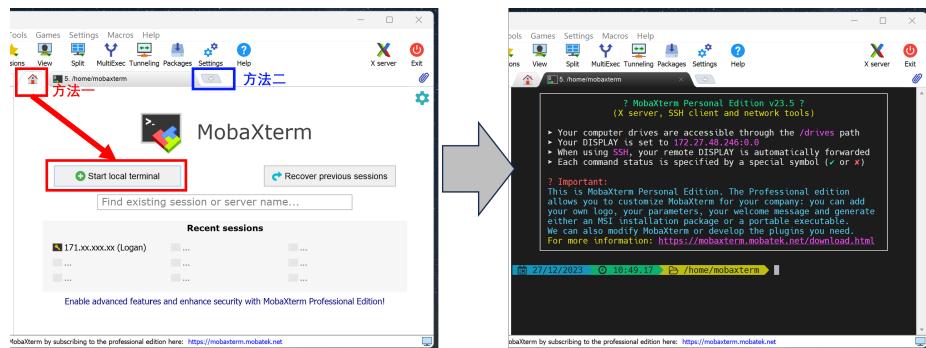


Fig. 27. MobaXterm 本地命令终端

4.4.4 MobaXterm 使用远程 Matlab 图形化界面

若已经如图 24⑨所示连接到服务器，继续进行如图 28所示的下列两步操作即可

步骤 1：切换到 MATLAB 的安装目录 `cd /opt/MATLAB/R2023a/bin`

步骤 2：启动 MATLAB `./matlab`

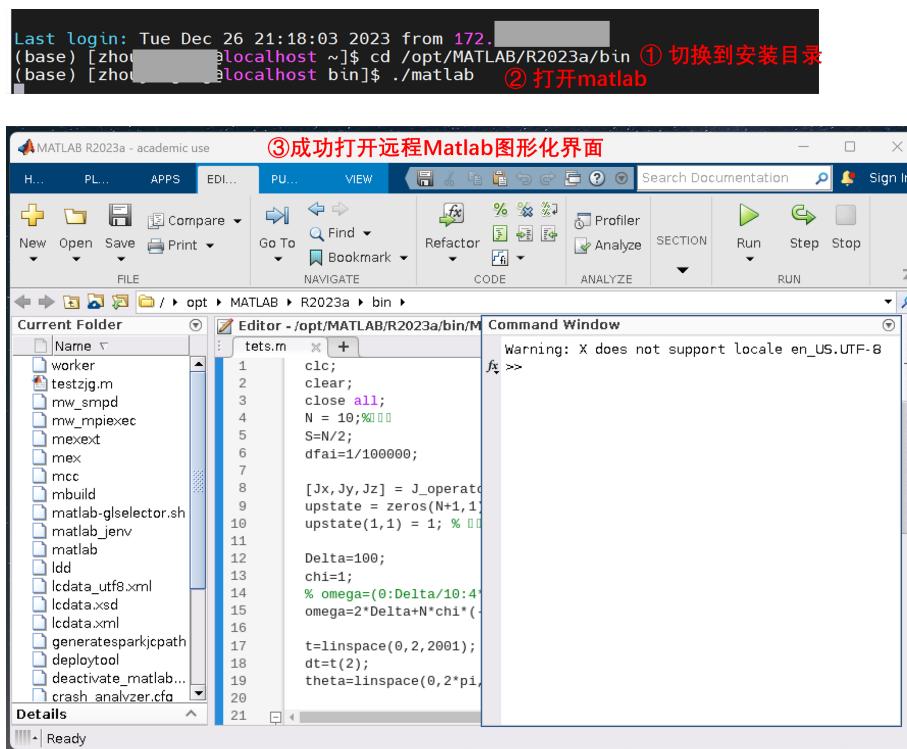


Fig. 28. 打开远程 Matlab

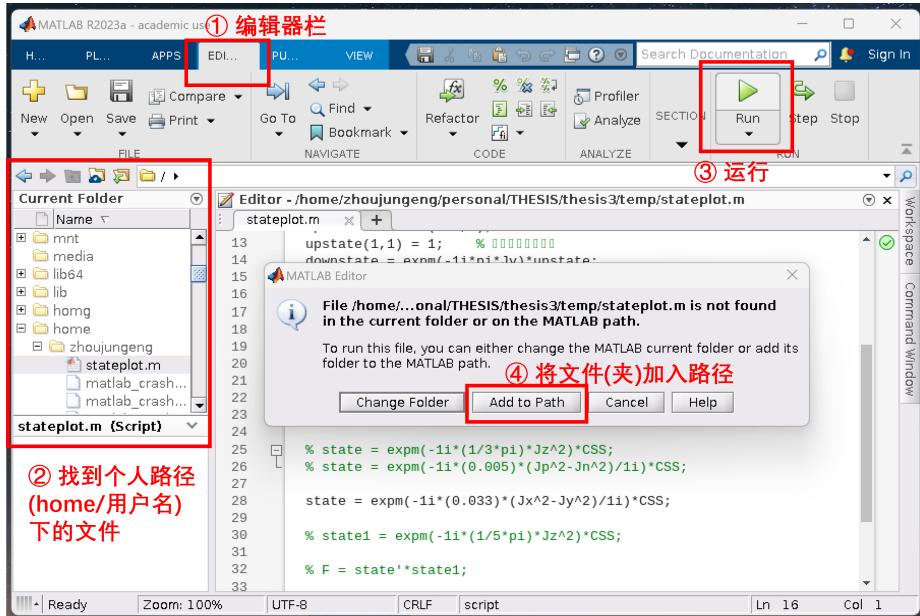
(tips: MobaXterm 单击右键即可粘贴，同时可以直接通过方向键的“↑,↓”调用历史命令。)

到此为止，我们已基本可以与本地一样使用远程的 Matlab 开展后续工作了。不过还有一点内容需要注意，为了让所有用户都能使用 Matlab，我们将其安装在公共路径，但即便如此，该路径也只有管理员有写入权限。因此，我们需要将自己的程序文件上传或者新建在个人账户文件夹中（方法见上一小节4.4.3），同时保存文件和运行也需要在个人路径下（尤其是代码中保存文件时需要写出完整的路径，而不能只写相对路径）。

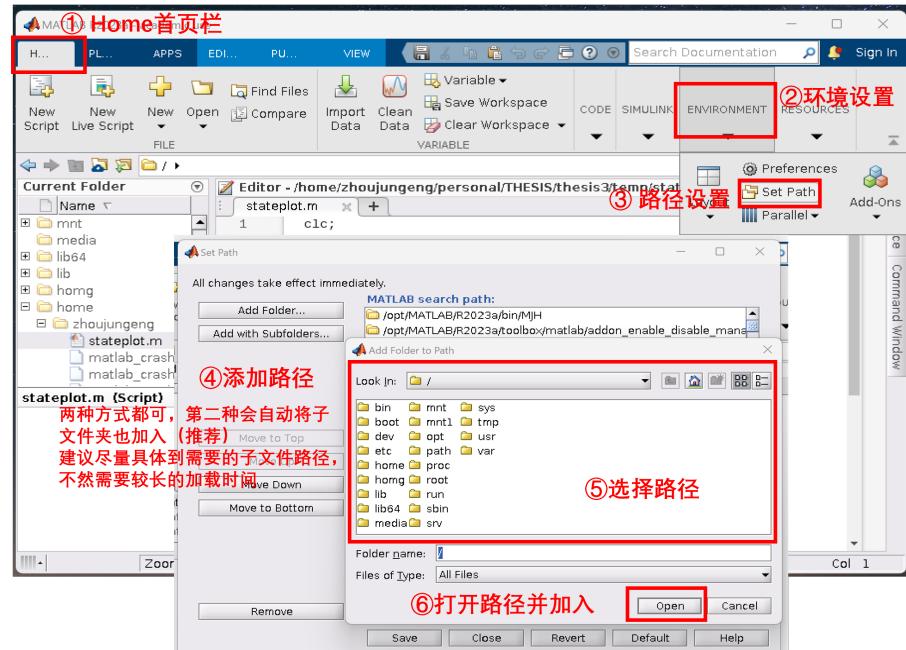
对于路径设置，我们给出如图 29的两种路径设置方法：

(a) 方法一：打开程序文件后自动加入路径；

(b) 方法二：手动设置 Matlab 路径；



(a) 方法一：打开程序后选择自动加入路径



(b) 方法二：在环境设置中手动加入路径

Fig. 29. 打开远程 Matlab

完成上述设置后即可，后续与本地使用 Matlab 基本一致，不再赘述。

5. Zotero 文献管理

5.1 Zotero 安装使用

官方下载地址 <https://www.zotero.org/download/> (截止 20260218 已更新至 Zotero 8, 包含程序安装包与谷歌浏览器插件)

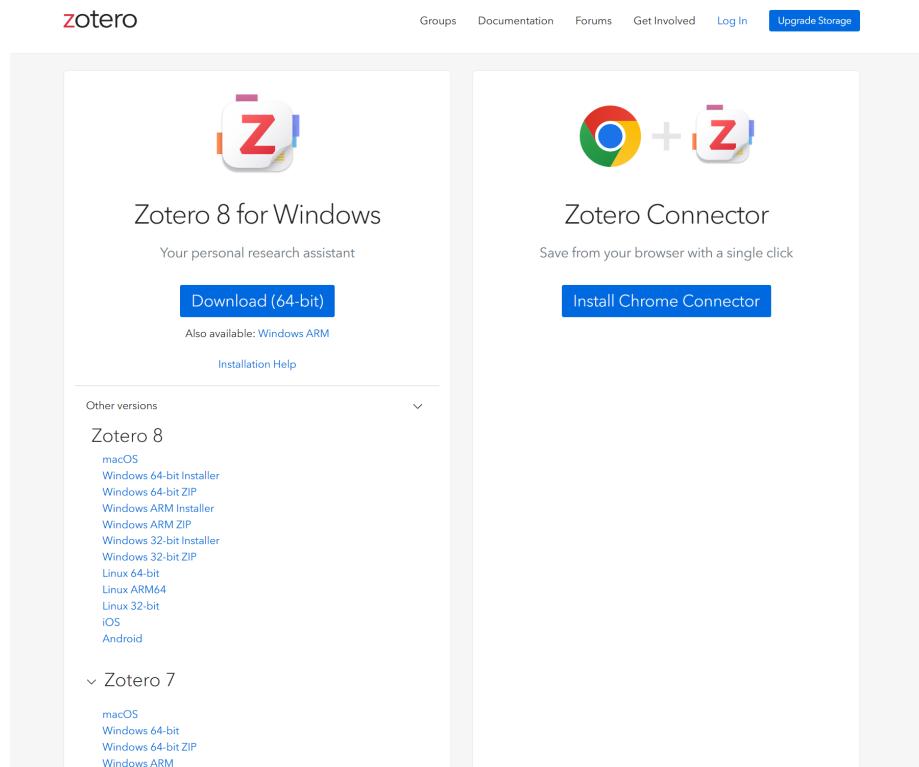


Fig. 30. Zotero 官网

1) 安装

常规安装步骤, 参考图 31

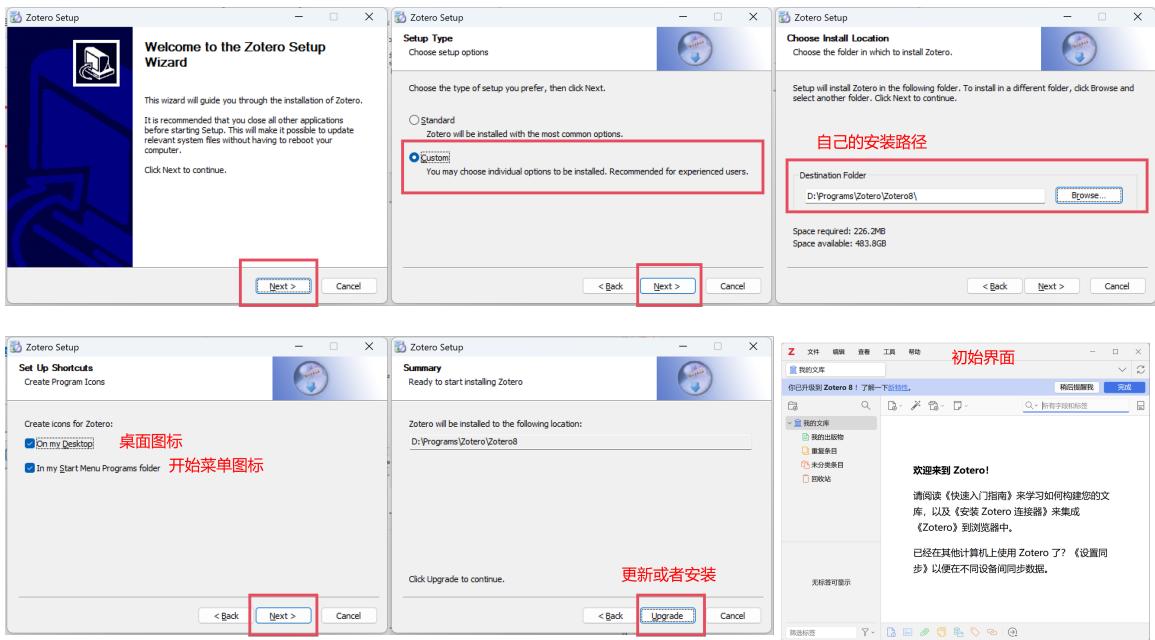


Fig. 31. Zotero 安装

2) 设置数据同步文件夹目录

- ① 注册/登录官方账号，具有一定的同步空间，无需进行其它设置
- ② 使用第三方云盘同步：将两个文件夹路径设置在具有同步功能的路径即可，最好是在同一个大文件夹中。

如果已经在其它设备安装使用过，需要将该设备所设置的两个同步文件夹通过复制或者云同步 (Onedrive/WPSCloud/坚果云) 从旧电脑转移到新设备（建议将这两个数据文件夹单独放在同一个大文件夹），设置到该路径即可。

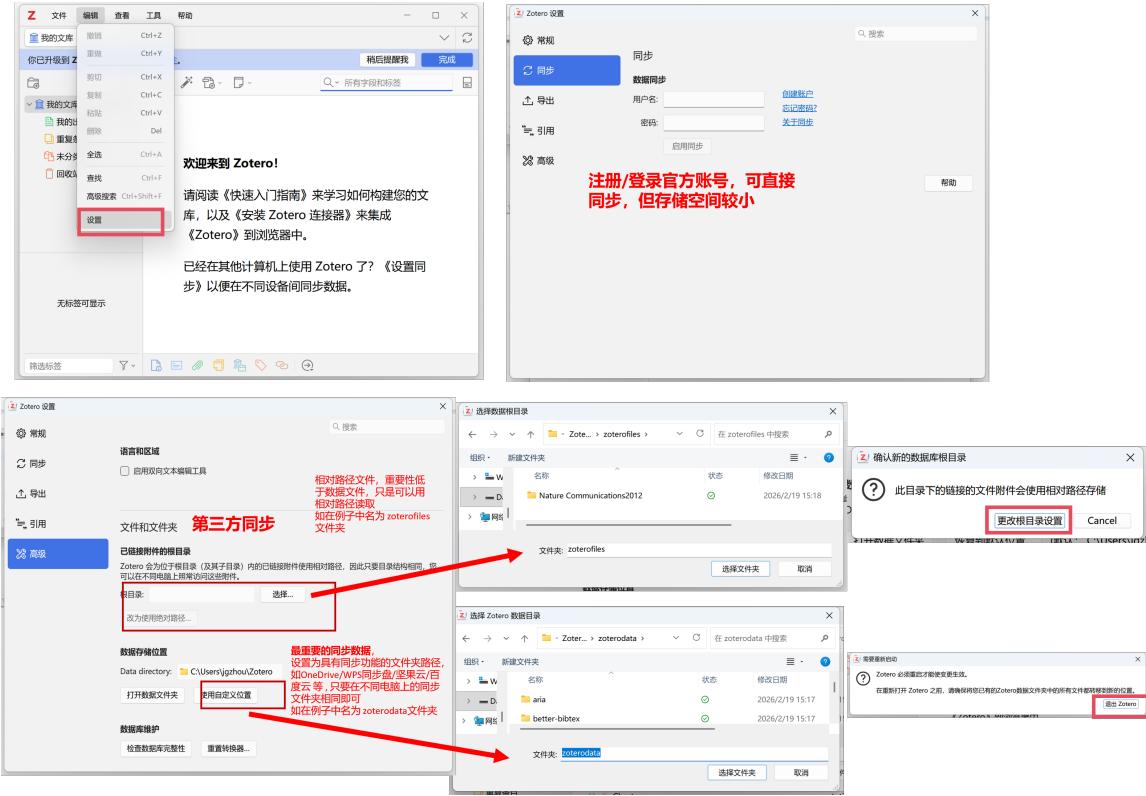


Fig. 32. Zotero 同步设置

参考图 32:

3) 安装插件

插件市场 (推荐): <https://zotero-chinese.com/plugins/>

官方目录: <https://www.zotero.org/support/plugins>

最简便的方式: 先安装名为“Zotero Add-on Market (Zotero 插件市场)”的这个插件, 其余插件便可以通过该插件提供的插件市场直接安装。

① 通过 github 库 <https://github.com/syt2/zotero-addons/releases> 或者上方推荐的插件市场, 如图 33 进行下载得到 zotero-addons.xpi 这个安装文件。

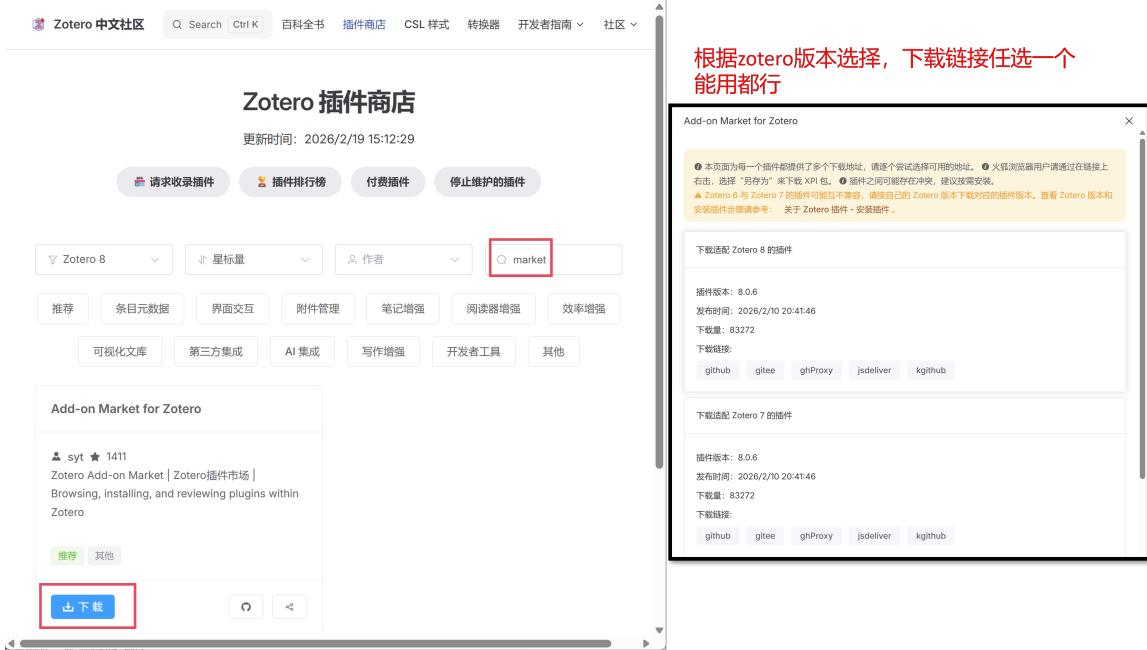


Fig. 33. Zotero 插件市场

②导入插件进行安装：图 34

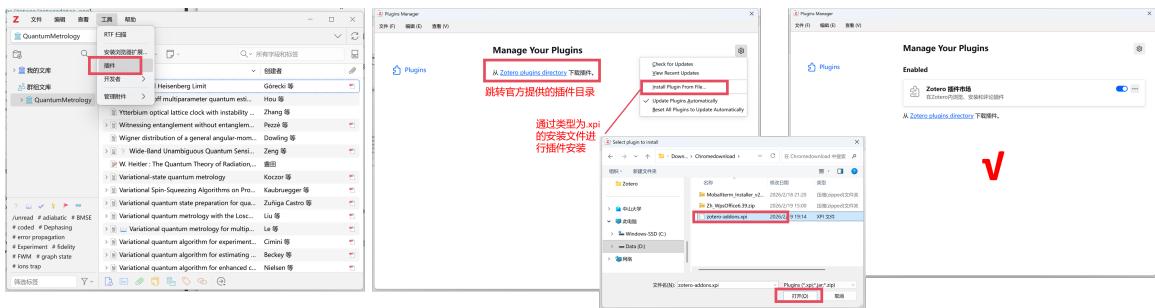


Fig. 34. Zotero 插件安装

③成功后即可如图 35 通过插件市场安装其它插件。

插件根据需求查看其简介自行选择即可，得星较高的基本都较为实用（双击还可进行评论向插件作者反馈）。

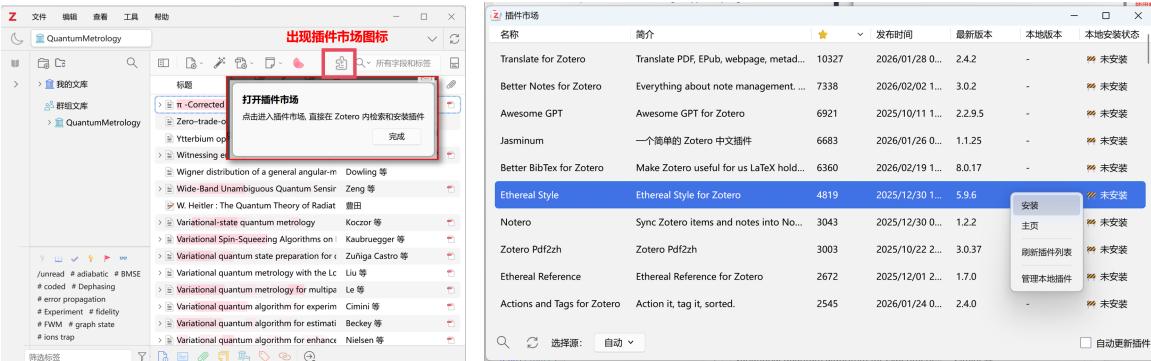


Fig. 35. 通过 Zotero 插件市场安装其它插件

5.2 Zotero 常用功能

文献分类管理:

文献引用导出:

文献笔记:

6. Latex 写作 (待施工)

6.1 TexLive + Vscode 本地写作

6.2 Overleaf 在线协同写作 (推荐)

7. ...

A. 常用 DOS 命令

B. windows 端

C. Linux 端

D. 数值计算实践示例

E. Python 实践示例：单轴扭曲作用中压缩参数随时间的变化

F. Matlab 实践示例：布洛赫球的绘制

见 <https://github.com/jungengzhou/CollectiveSpin.git>

参考文献