Appendix B

MacOS

B.1 Install "Command line utilities"

The Command Line Tool package gives Mac terminal users many commonly used tools, utilities, and compilers, including make, GCC, clang, perl, svn, git, size, strip, strings, libtool, cpp, what, and many other useful commands that are usually found in default linux installations.

Start the terminal app, found at `/Applications/Utilities/Terminal.app`. Type the following command string: `xcode-select --install`

For those interested in knowing the details of what is installed on their Mac and where it's going, the entire command line toolkit package gets placed in the following directory:

```
/Library/Developer/CommandLineTools/
```

B.2 Install "XQuartz"

The XQuartz project is an open-source effort to develop a version of the X.Org X Window System that runs on OS X.
`https://github.com/XQuartz/XQuartz/releases/download/XQuartz-2.8.1/XQuartz-2.8.1.dmg`

B.3 Install "Homebrew"

Homebrew is a package manager for macOS and installs the stuff you need that Apple did not. Install homebrew using (see `https://brew.sh/index`)

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

When you install Homebrew on an Intel Mac, it installs in the `/usr/local/Homebrew` directory. On Apple M1, the installation directory is `/opt/homebrew`.

B.4 Install gfortran via Homebrew

```
brew install gcc
```

(gcc install comes with gfortran)

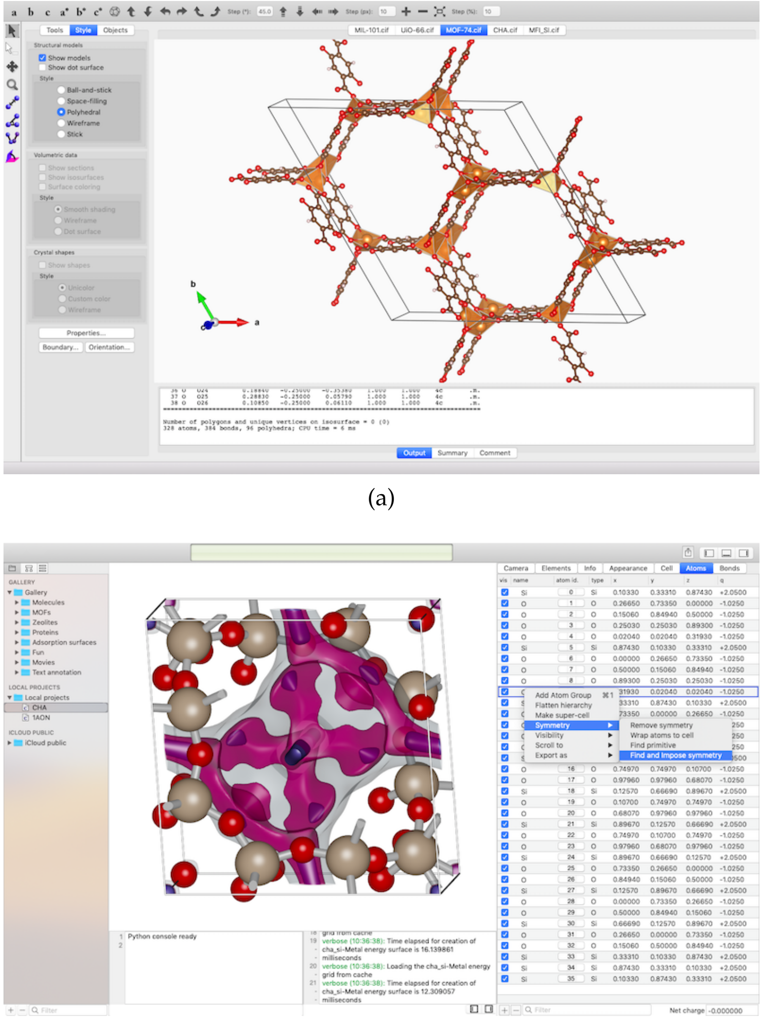
B.5 Install gnuplot

```
brew install gnuplot
```

B.6 Install a visualizer

To be able to view molecular systems, install a molecular visualizer. Common options are:

- Install VMD from: https://www.ks.uiuc.edu/Research/vmd/
- Version `VMD 1.9.4a55-x86_64-Rev11` works on Monterey.
- iRASPA is available from the MacOS AppStore.
- VESTA can be downloaded from: https://jp-minerals.org/vesta/en/



```
  36 O   O24        0.18840   -0.25000   -0.35380   1.000    1.000    4c       .m.
  37 O   O25        0.28830   -0.25000    0.05790   1.000    1.000    4c       .m.
  38 O   O26        0.10850   -0.25000    0.06110   1.000    1.000    4c       .m.
================================================================================

Number of polygons and unique vertices on isosurface = 0 (0)
328 atoms, 384 bonds, 96 polyhedra; CPU time = 6 ms
```

(a)

```
Python console ready
```

```
grid from cache
verbose (10:36:38): Time elapsed for creation of cha_si-Metal energy surface is 16.139861 milliseconds
verbose (10:36:38): Loading the cha_si-Metal energy grid from cache
verbose (10:36:38): Time elapsed for creation of cha_si-Metal energy surface is 12.309057 milliseconds
```

vis	name	atom id.	type	x	y	z	q
☑	Si	0	Si	0.10330	0.33310	0.87430	+2.0500
☑	O	1	O	0.26650	0.73350	0.00000	-1.0250
☑	O	2	O	0.15060	0.84940	0.50000	-1.0250
☑	O	3	O	0.25030	0.25030	0.89300	-1.0250
☑	O	4	O	0.02040	0.02040	0.31930	-1.0250
☑	Si	5	Si	0.87430	0.10330	0.33310	+2.0500
☑	O	6	O	0.00000	0.26650	0.73350	-1.0250
☑	O	7	O	0.50000	0.15060	0.84940	-1.0250
☑	O	8	O	0.89300	0.25030	0.25030	-1.0250
☑	O			31930	0.02040	0.02040	-1.0250
☑	S			33310	0.87430	0.10330	+2.0500
☑	O			73350	0.00000	0.26650	-1.0250
☑	O						250
☑	O						250
☑	O						250
☑	S						500
☑	O	16	O	0.74970	0.74970	0.10700	-1.0250
☑	O	17	O	0.97960	0.97960	0.68070	-1.0250
☑	Si	18	Si	0.12570	0.66690	0.89670	+2.0500
☑	O	19	O	0.10700	0.74970	0.74970	-1.0250
☑	O	20	O	0.68070	0.97960	0.97960	-1.0250
☑	Si	21	Si	0.89670	0.12570	0.66690	+2.0500
☑	O	22	O	0.74970	0.10700	0.74970	-1.0250
☑	O	23	O	0.97960	0.68070	0.97960	-1.0250
☑	Si	24	Si	0.89670	0.66690	0.12570	+2.0500
☑	O	25	O	0.73350	0.26650	0.00000	-1.0250
☑	O	26	O	0.84940	0.15060	0.50000	-1.0250
☑	Si	27	Si	0.12570	0.89670	0.66690	+2.0500
☑	O	28	O	0.00000	0.73350	0.26650	-1.0250
☑	O	29	O	0.50000	0.84940	0.15060	-1.0250
☑	Si	30	Si	0.66690	0.12570	0.89670	+2.0500
☑	O	31	O	0.26650	0.00000	0.73350	-1.0250
☑	O	32	O	0.15060	0.50000	0.84940	-1.0250
☑	Si	33	Si	0.33310	0.10330	0.87430	+2.0500
☑	Si	34	Si	0.87430	0.33310	0.10330	+2.0500
☑	Si	35	Si	0.10330	0.87430	0.33310	+2.0500

Net charge -0.000000

(b)

Figure B.1: Mac screenshots of (a) VESTA, (b) iRASPA.

Appendix C

Linux

C.1 Install necessary linux packages

For Ubuntu or distributions derived from Ubuntu, use the following commands

```
sudo apt-get install build-essential gfortran
sudo apt-get install gnuplot
sudo apt-get install csh vim
```

C.2 Install a visualizer

- Install VMD from: https://www.ks.uiuc.edu/Research/vmd/
- iRASPA is available from the Canonical Store: https://snapcraft.io/iraspa
- VESTA can be downloaded from: https://jp-minerals.org/vesta/en/

C.3 Useful optional linux packages

These packages are generally useful in scientific computing.

```
sudo apt-get install git bzr xfig emacs
sudo apt-get install texmaker grace gimp automake curl chemical-mime-data
sudo apt-get install openbabel avogadro jmol cp2k pymol
sudo apt-get install tcsh vim speedcrunch ftnchek cmake
```

Appendix D

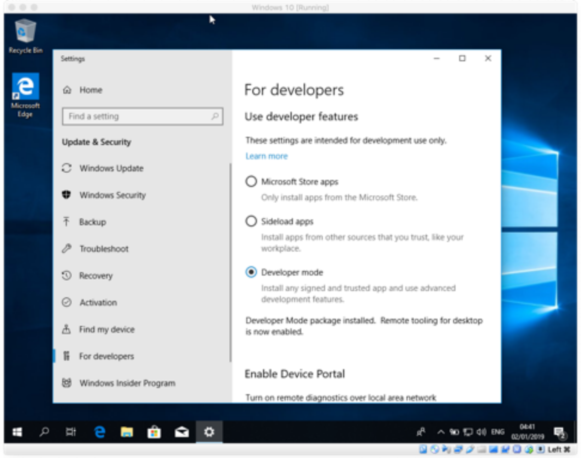
Windows 10/11

D.1 Requirements

- Minimum windows 10 version is the Anniversary Update (Version 1607),
- 64-bit version of Windows 10

D.2 Turn on Developer Mode (not needed on recent versions)

- Go to "Start menu"
- Goto "Settings"
- Search for "developer settings"
- Turn on "Developer Mode"



For developers

Use developer features

These settings are intended for development use only.

○ Microsoft Store apps
Only install apps from the Microsoft Store.

○ Sideload apps
Install apps from other sources that you trust, like your workplace.

● Developer mode
Install any signed and trusted app and use advanced development features.

Developer Mode package installed. Remote tooling for desktop is now enabled.

Enable Device Portal

Turn on remote diagnostics over local area network

Figure D.1: Turn on "Developer Mode"

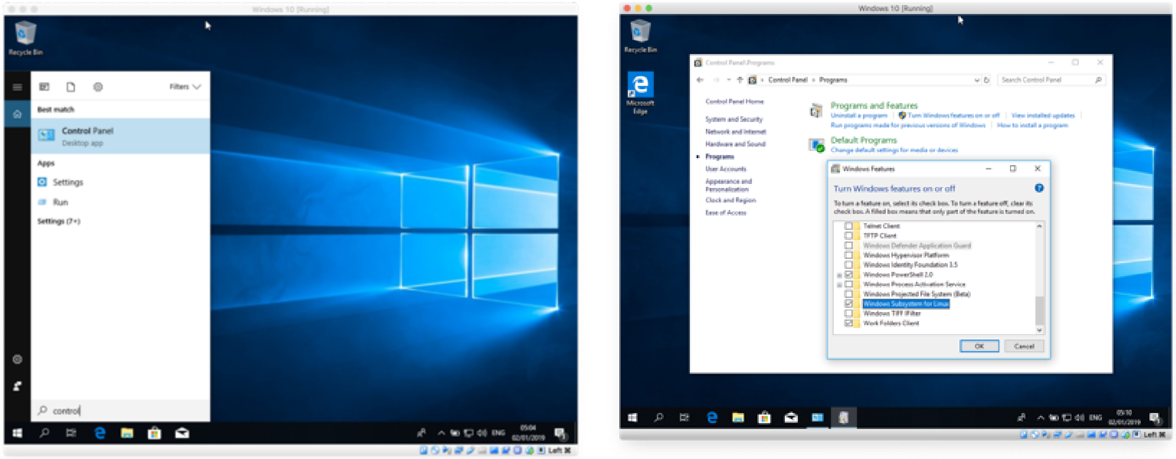
D.3 Install Windows Subsystem for Linux (WSL)

Computers that do *not* have a CPU with Hyper-V Virtualization support are only capable of running WSL1.

- Go to “Control Panel”
- Goto “Programs” and “Program and Features”
- Goto “Turn Windows features on or off”
- Select “Windows Subsystem for Linux”
- Reboot if required

Open a powershell and run

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



Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- □ Telnet Client
- □ TFTP Client
- □ Windows Defender Application Guard
- □ Windows Hypervisor Platform
- □ Windows Identity Foundation 3.5
- ☑ Windows PowerShell 2.0
- □ Windows Process Activation Service
- □ Windows Projected File System (Beta)
- ☑ Windows Subsystem for Linux
- □ Windows TIFF IFilter
- ☑ Work Folders Client

(a)

(b)

Figure D.2: Install Windows Subsystem for Linux: (a) start Control Panel, (b) select “Windows Subsystem for Linux”

Computers that have a CPU with Hyper-V Virtualization support are capable of running WSL2. WSL 2 is a major overhaul of the underlying architecture and uses virtualization technology and a Linux kernel to enable new features. WSL 2 is faster, more versatile, and uses a real Linux kernel. The primary goals of this update are to increase file system performance and add full system call compatibility. WSL 2 is only available in Windows 10, Version 1903, Build 18362 or higher.

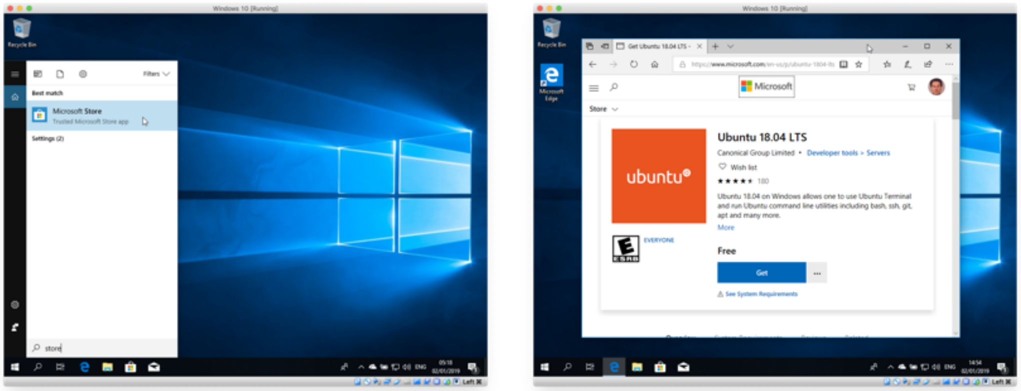
- Go to “Control Panel”
- Goto “Programs” and “Program and Features”

- Goto “Turn Windows features on or off”
- Select “Windows Subsystem for Linux”
- Select “Virtual Machine Platform”
  ”Enables platform support for virtual machines” and is required for WSL2.
- Select “Hyper-V”
  A computer with Hyper-V Virtualization support is required for WSL2.
- Reboot if required

D.4 Download Ubuntu 18 from the Windows Store

- Go to “Windows Store”
- Search “Ubuntu 20”
- Install “Ubuntu 20.04 LTS”

Other available linux distributions are: Ubuntu 18.04 LTS, OpenSUSE Leap 42, SUSE Linux Enterprise Server 15, Kali Linux, Debian GNU/Linux, Fedora Remix for WSL, Pengwin, Pengwin Enterprise, and Alpine WSL.



Ubuntu 18.04 on Windows allows one to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

(a)

(b)

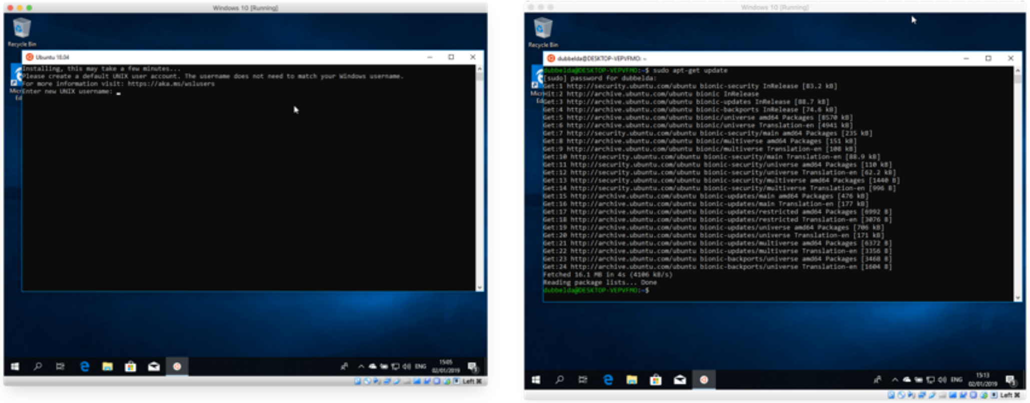
Figure D.3: Install Windows Subsystem for Linux: (a) start Control Panel, (b) install “Ubuntu 20.04 LTS”.

D.5 Create a linux account

This account does not have to be the same as your Windows account.

- Enter a username in the required field and press Enter (you can not use the username “admin”)

- Enter a password (twice)



```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: _
```

(a)

```
dubbelda@DESKTOP-VEPVFMO:~$ sudo apt-get update
[sudo] password for dubbelda:
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [235 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [88.9 kB]
Get:11 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [110 kB]
Get:12 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [62.2 kB]
Get:13 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [1440 B]
Get:14 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [996 B]
Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [476 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [177 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [6992 B]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [3076 B]
Get:19 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [706 kB]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [171 kB]
Get:21 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [6372 B]
Get:22 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [3356 B]
Get:23 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [3468 B]
Get:24 http://archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [1604 B]
Fetched 16.1 MB in 4s (4106 kB/s)
Reading package lists... Done
dubbelda@DESKTOP-VEPVFMO:~$
```

(b)

Figure D.4: Setup (a) the default linux user account (you can not use the username "admin"), (b) the linux system.

D.6 Install required and useful linux packages

- `sudo apt-get update`
- `sudo apt-get upgrade`
- `sudo apt-get install build-essential csh gfortran gnuplot qt5-default gedit`
- `sudo apt install ca-certificates findutils command-not-found nano curl git`
- `sudo apt install vim openssh-client less gimp apt-utils htop whois python3-pip`

On WSL1, you will have to run:

```
sudo strip --remove-section=.note.ABI-tag /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
```

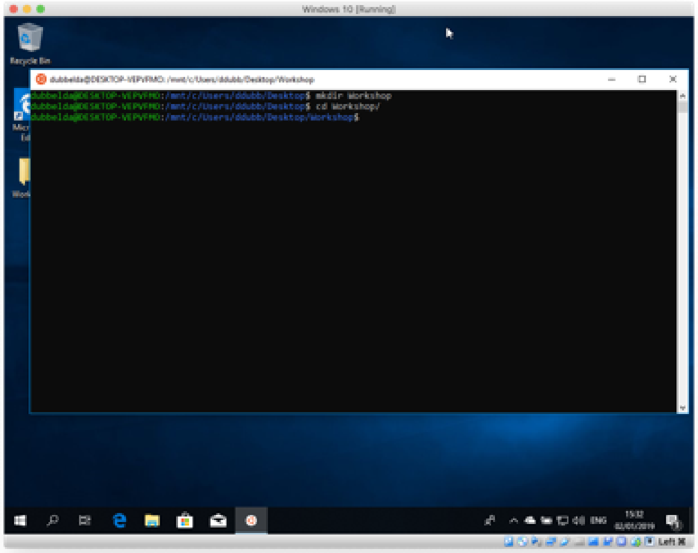
which solve the error error ″libQt5Core.so.5: cannot open shared object file: No such file or directory″ which you get when you try to run gnuplot.

D.7 Where to put your files?

To use your files from both windows and linux, you should put your files in `/mnt/c/Users` and then your windows username. Any change made from the linux side will immediately show up in windows. You can then use installed window software to read and visualize output files.

When you access files on your Windows file-system from within Bash, it will honor the NT file-system behaviors (e.g. case-insensitivity), permissions, etc. so you can easily access the same files using both Windows tools and Bash tools without having to copy files back and forth between file-systems.

If you want to access a system folder your user account does not have permission to access, you would need to right-click the Bash shell shortcut and select "Run as Administrator" to launch the Bash shell with Windows Administrator privileges.



dubbelda@DESKTOP-VEPVFMO: /mnt/c/Users/ddubb/Desktop/Workshop

```
dubbelda@DESKTOP-VEPVFMO:/mnt/c/Users/ddubb/Desktop$ mkdir Workshop
dubbelda@DESKTOP-VEPVFMO:/mnt/c/Users/ddubb/Desktop$ cd Workshop/
dubbelda@DESKTOP-VEPVFMO:/mnt/c/Users/ddubb/Desktop/Workshop$
```

Figure D.5: Your windows files are accessible from `/mnt/c`.

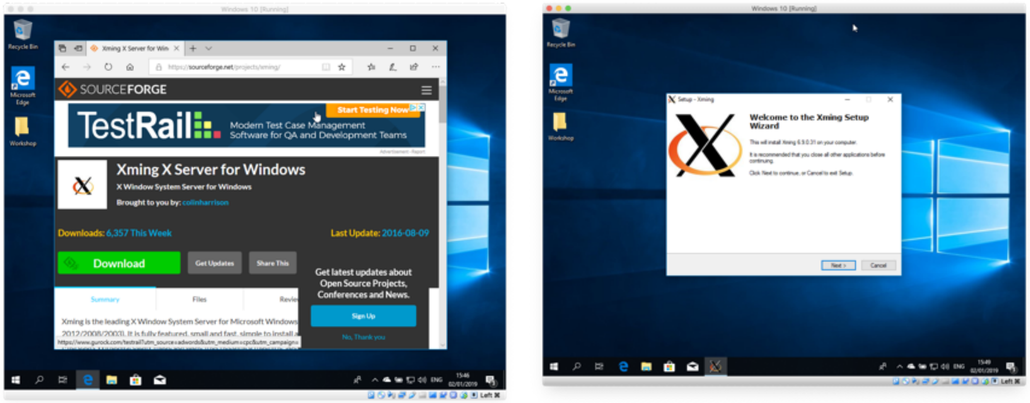
D.8 Run graphical programs in linux

Windows 11:
WSL 2 enables Linux GUI applications to feel native and natural to use on Windows. You will need to be on Windows 11 Build 22000 or higher to access this feature.

- Launch Linux apps from the Windows Start menu
- Pin Linux apps to the Windows task bar
- Use alt-tab to switch between Linux and Windows apps
- Cut + Paste across Windows and Linux apps

Note: the X-server does not automatically start when running `gnuplot` or `gedit`. If so, run `gimp` once to start the server.

For Windows 10, a windows X-server like Xming or VcXsrv needs to be installed.


Xming X Server for Windows

X Window System Server for Windows

Brought to you by: colinharrison

Downloads: 6,357 This Week

Last Update: 2016-08-09

Xming is the leading X Window System Server for Microsoft Windows 2012/2008/2003). It is fully featured, small and fast, simple to install a

Get latest updates about Open Source Projects, Conferences and News.

Welcome to the Xming Setup Wizard

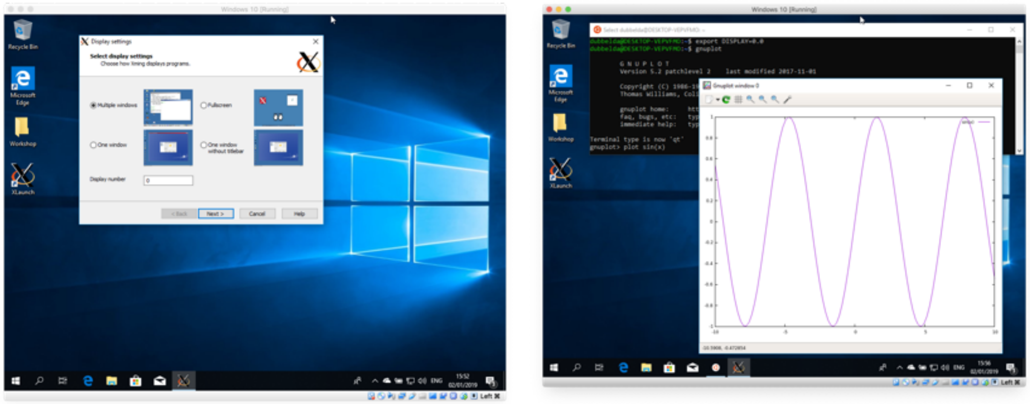
This will install Xming 6.9.0.31 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

(a) (b)

Figure D.6: Download the Xming X-server and install it.


Select display settings

Choose how Xming displays programs.

● Multiple windows ○ Fullscreen ○ One window ○ One window without titlebar

Display number: 0

```
dubbelda@DESKTOP-VEPVFMO:~$ export DISPLAY=0.0
dubbelda@DESKTOP-VEPVFMO:~$ gnuplot

        G N U P L O T
        Version 5.2 patchlevel 2    last modified 2017-11-01

        Copyright (C) 1986-19
        Thomas Williams, Coli

        gnuplot home:     htt
        faq, bugs, etc:   typ
        immediate help:   typ

Terminal type is now 'qt'
gnuplot> plot sin(x)
```

(a) (b)

Figure D.7: Using gnuplot in linux, (a) start Xming, (b) connect to the X-server display `export DISPLAY=:0` and start gnuplot.

In order to run Linux GUI applications using Windows 10, you must:

- Install a X server for Windows
- Configure bash to tell GUIs to use the local X server

Installing an X server will allow graphical linux applications to appear on your Windows desktop. The Xming X Server for Windows works well for this, although there are other X servers for Windows and they should also work. Download the Xming X-server from
`https://sourceforge.net/projects/xming`
and install Xming using the default settings. Launch Xming and it will appear in your system tray, running in the background and waiting for you to launch a graphical Linux program.

Xming is the leading X Window System Server for Microsoft Windows. It is fully featured, small and fast, simple to install and because it is standalone native Microsoft Windows, easily made portable (not needing a machine-specific installation). It features support of several languages and has Mesa 3D, OpenGL, and GLX 3D graphics extensions capabilities. Once you've installed the program, you can specify the X server and then launch the application. For example, to launch gnuplot, run the below commands.

```
export DISPLAY=:0
gnuplot
```

Configure bash to use the local X server automatically.

- In bash run:
  `echo "export DISPLAY=localhost:0.0" >> ~/.bashrc`

- To have the configuration changes take effect, restart bash, or run:
  `. ~/.bashrc`

D.9 Use molecular viewers in windows

VMD is designed for viewing and analyzing molecular dynamics data of biological systems such as proteins, nucleic acids, lipid bilayer assemblies, etc. It also includes tools for working with volumetric data and sequence data. The functionality can be easily extended using python and Tcl scripts as VMD includes embedded Tcl and Python interpreters. Figure D.9a shows a screen shot of VMD while visualizing an MD simulation of a protein embedded in a membrane. Download VMD from the following link to install it:
`https://www.ks.uiuc.edu/Research/vmd/`.

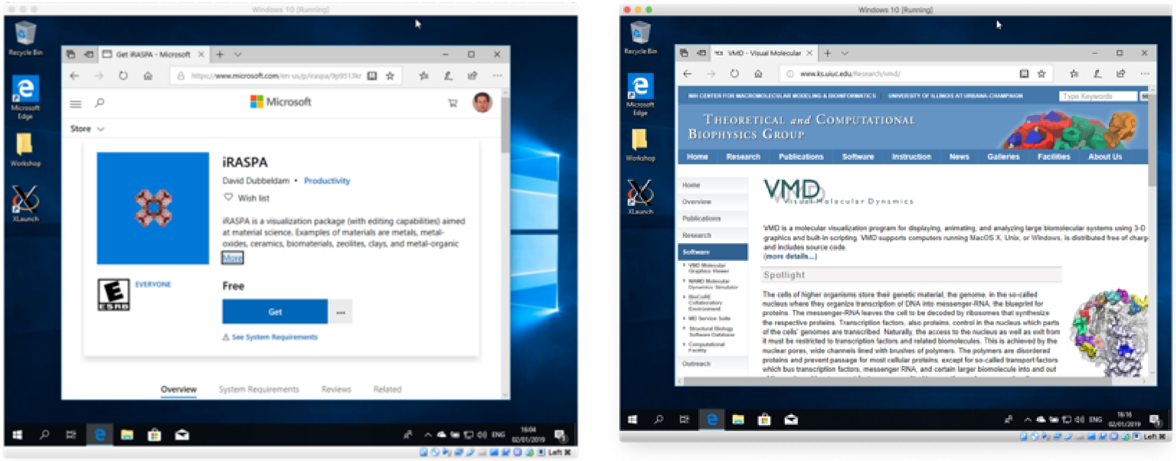
VESTA is a 3D visualization program for structural models, volumetric data such as electron/nuclear densities, and crystal morphologies. VESTA can be downloaded from: https://jp-minerals.org/vesta/en/

iRASPA is a visualization package (with editing capabilities) aimed at material science. Figure D.9b shows a screenshot of the program. iRASPA supports crystallographic operations like space group detection and finding the primitive cell, and extensively utilizes GPU computing. iRASPA is available from the Windows store. It requires OpenGL and OpenCL drivers to be installed for your video card.

D.10 Useful additional linux packages

These packages are generally useful in scientific computing.

```
sudo apt-get install git bzr xfig emacs
sudo apt-get install texmaker grace gimp automake curl chemical-mime-data
sudo apt-get install openbabel avogadro jmol cp2k pymol
sudo apt-get install tcsh speedcrunch ftnchek cmake
```



iRASPA is a visualization package (with editing capabilities) aimed at material science. Examples of materials are metals, metal-oxides, ceramics, biomaterials, zeolites, clays, and metal-organic

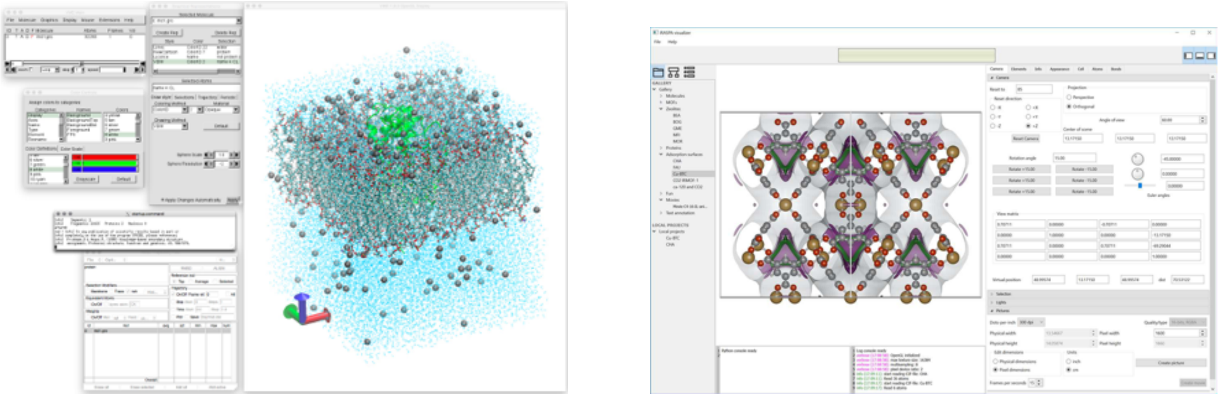
VMD is a molecular visualization program for displaying, animating, and analyzing large biomolecular systems using 3-D graphics and built-in scripting. VMD supports computers running MacOS X, Unix, or Windows, is distributed free of charg- and includes source code.

The cells of higher organisms store their genetic material, the genome, in the so-called nucleus where they organize transcription of DNA into messenger-RNA, the blueprint for proteins. The messenger-RNA leaves the cell to be decoded by ribosomes that synthesize the respective proteins. Transcription factors, also proteins, control in the nucleus which parts of the cells' genomes are transcribed. Naturally, the access to the nucleus as well as exit from it must be restricted to transcription factors and related biomolecules. This is achieved by the nuclear pores, wide channels lined with brushes of polymers. The polymers are disordered proteins and prevent passage for most cellular proteins, except for so-called transport factors which bus transcription factors, messenger RNA, and certain larger biomolecule into and out

(a)

(b)

Figure D.8: Download visualization packages (a) iRASPA from the Windows 10 Store, or (b) VMD from `https://www.ks.uiuc.edu/Research/vmd/`



(a)

(b)

Figure D.9: Windows 10 screenshots of (a) VMD showing a protein embedded in a membrane with water and ions at both sides, (b) iRASPA showing the primitive unit cell of a CHA-type zeolite with three adsorption surfaces showing the shape of the cavity, the diffusion paths, and the adsorption sites.