Installing LAMMPS, PYTHON and OVITO

Installing Ubuntu environment on Windows:

This guide explains how to set up an Ubuntu environment on a Windows machine using Windows Subsystem for Linux (WSL). Once the environment is set up, you can use it to run installation scripts like the one provided for Python and other tools.

Note:

- **In the case of failure please follow CHATGPT, but never use chat gpt to solve your Exercises.**
- **If you are stuck you can send an email to id vikkivarma16@gmail.com. But before, please first ask CHATGPT.**

Some trick of the trade:

Please remove any extra space, while copying these commands. The best way is to copy and paste these commands to CHATGPT and ask to rectify the commands. Copy those commands from the chatgpt output and paste in your terminal, and then press enter.

Step 1: Enable Windows Subsystem for Linux (WSL)

Follow these steps to enable WSL on your Windows machine:

1. Open PowerShell as administrator.
2. Run the following command to enable the WSL feature:

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-
    Subsystem-Linux /all /norestart
```

3. Enable the Virtual Machine Platform feature:

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

4. Restart your computer to apply the changes.

Step 2: Install Ubuntu Using PowerShell

1. Open PowerShell (please this time, do not open it as Administrator).
2. Run the following command to set WSL 2 as the default version:

```
wsl --update
wsl --set-default-version 2
```

3. Download and install the Ubuntu distribution directly from PowerShell. For example, to install Ubuntu 20.04 LTS, run and wait:

   ```
   wsl --install -d Ubuntu-20.04
   ```

   If it does not work please type,

   ```
   wsl --install -d Ubuntu
   ```

4. Restart your computer.

5. Wait for the installation process to complete. Once finished, Ubuntu will be installed on your system.

Step 3: Set Up Ubuntu

1. Launch Ubuntu by typing the following command in PowerShell:

   ```
   wsl
   ```

2. Follow the prompts to create a new UNIX username and password.

3. Update the package list and upgrade existing packages by running:

   ```
   sudo apt update
   sudo apt upgrade -y
   ```

Alternate method

This document provides a step-by-step guide on how to install and set up Ubuntu from the Microsoft Store. By following these instructions, you can run a Linux-based environment directly on your Windows machine using the Windows Subsystem for Linux (WSL).

Prerequisites

Before installing Ubuntu, ensure the following:

- You are using Windows 10 version 2004 or later, or Windows 11.

- Windows Subsystem for Linux (WSL) is enabled.

Step 1: Enable Windows Subsystem for Linux (WSL)

To enable WSL, follow these steps:

1. Open PowerShell as an administrator. You can do this by searching for **PowerShell** in the Start menu, right-clicking on it, and selecting **Run as administrator**.

2. Run the following command to enable WSL and set WSL 2 as the default version:

   ```
   wsl --install
   ```

3. Restart your computer if prompted.

4. If the above command does not work, you can manually enable WSL and the Virtual Machine Platform feature using the following commands:

   (a) Run the following command to enable the WSL feature:

```
dism.exe /online /enable-feature /featurename:Microsoft-
    Windows-Subsystem-Linux /all /norestart
```

(b) Enable the Virtual Machine Platform feature:

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

(c) Restart your computer to apply the changes.

Step 2: Install Ubuntu from the Microsoft Store

1. Open the Microsoft Store from the Start menu.
2. Search for **Ubuntu** in the Microsoft Store.
3. Select the desired version of Ubuntu (e.g., Ubuntu 20.04 LTS or Ubuntu 22.04 LTS).
4. Click **Get** or **Install** to download and install Ubuntu.

Step 3: Launch and Set Up Ubuntu

1. Once the installation is complete, click **Launch** from the Microsoft Store, or open Ubuntu from the Start menu.
2. Wait for the installation process to complete. This may take a few minutes.
3. Create a new UNIX username and password when prompted. This will be your default user for the Ubuntu environment.
4. You can now use the Ubuntu terminal for your development tasks.

Step 4: Update and Upgrade Packages

It is recommended to update and upgrade the installed packages to ensure you have the latest updates. Run the following commands in the Ubuntu terminal:

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

Conclusion

You have successfully installed Ubuntu on your Windows machine using the Microsoft Store. You can now enjoy the power of Linux directly within your Windows environment.

For more information, refer to the official WSL documentation: `https://docs.microsoft.com/en-us/windows/wsl/`.

Installing LAMMPS and related packages on Ubuntu on-Windows:

You can download all the tutorial stuffs for both the tutorial I and tutorial II by using the git clone command, in whatever the folder you prefer.

- Open the Ubuntu terminal.

  Note: Students are supposed to know some Linux commands, such as:

  1. To move your terminal directory to the home (Important):

     ```
     cd
     ```

  2. To move into a single directory (Important):

     ```
     cd <directory_name>
     ```

  3. To move up one directory (to the parent directory or previous directory) (Important):

     ```
     cd ../
     ```

  4. To list all files and directories in the current directory (Important):

     ```
     ls
     ```

  5. To list all files, including hidden ones, with detailed information:

     ```
     ls -la
     ```

  6. To create a new directory:

     ```
     mkdir <directory_name>
     ```

  7. To delete a file:

     ```
     rm <file_name>
     ```

  8. To delete a directory and its contents recursively (most dangerous command, Thanos's finger snap):

     ```
     rm -r <directory_name>
     ```

  9. To copy a file or directory:

     ```
     cp <source> <destination>
     ```

  10. To move or rename a file or directory:

      ```
      mv <source> <destination>
      ```

  11. To display the contents of a file:

      ```
      cat <file_name>
      ```

  12. To check your current working directory:

      ```
      pwd
      ```

- Navigate to the desired folder using `cd <directory name>` and then type and enter `ls` to show all the content from where you can see the directory files and these things and then again type and enter `cd <directory name>`. And keep on doing until you reach the desired folder stored within folder. If you are moving in the wrong directory use `cd ../` to move in the previous folder. If you want to switch direct to the home directory please type and enter `cd`.

- First install git typing the following command in your Ubuntu terminal.

```
sudo apt update
sudo apt install git -y
```

- Then use the following command to clone all your tutorial and exercise stuffs.

```
git clone https://github.com/vikkivarma16/lammps_tutorial.git
```

  If it does not work please click on the following link and download the zip file,

  https://github.com/vikkivarma16/lammps_tutorial

Ask CHATGPT how to navigate to a folder in Ubuntu using the command line. Also learn other commands like, how to create, copy, paste a file or folder etc.

Students are free to follow their own way of installation of **PYTHON3** packages like **SCIPY**, **NUMPY**, **MATPLOTLIB**, **TDQM** along with **LAMMPS**.

However, it can be done in the following way also:

1. You have the installation bash script stored and downloaded from the following Github link
   https://www.github.com/vikkivarma16/lammps_tutorial.

2. You can also clone it by using the following command,

```
sudo apt update
sudo apt install git -y
git clone https://github.com/vikkivarma16/lammps_tutorial.git
```

3. After 'cloning', or lets say 'downloading' the directory, you must have a file named `install_tools.sh`.

4. Open the Ubuntu terminal navigate to the folder which you have cloned (which might be named as `lammps_tutorial-main`).

5. Type the command `ls`, and press enter to see if you have the file named `install_tools.sh` along with the other folder and files.

6. Make the script executable by typing and entering the following command:

```
chmod +x install_tools.sh
```

7. Run the script by typing and entering the following command and then type the password whenever asked:

```
./install_tools.sh
```

Under the network section, in the left påne of the folders UI (where you get after clicking **This PC** button), you will have the access to the ubuntu folder. Within that folder you will get the home folder, which you can enable for the quick access and later open using **UI** on windows. So you are supposed to clone all your stuffs within that folder, as it will be easier to open things and read using **UI**. To run the codes, you can specifically navigate to those folders from the terminal.

Installing LAMMPS and related packages on Ubuntu:

Please follow the commands as given below to install LAMMPS, MATPLOTLIB, PYTHON3 and many more packages.

1. You have the installation bash script stored from the following Github link **https://www.github.com/vikkivarma16/lammps_tutorial**.
2. After cloning the directory, you must have a file named `install_tools.sh`.
3. Open the Ubuntu terminal.
4. Make the script executable:

   ```
       chmod +x install_tools.sh
   ```

5. Run the script and type the password whenever asked:

   ```
       ./install_tools.sh
   ```

Conclusion

You have successfully set up an Ubuntu environment on your Windows machine and installed the required tools and libraries. You can now use this environment for development, simulation, or any other tasks.

Run the LAMMPS code:

Now you can run your code using the following command,

```
lmp -in input_script.in
```

Run the PYTHON code:

Now you can run your code using the following command,

```
python3 input_code.py
```

Installing OVITO on Window:

OVITO is a scientific data visualisation and analysis software for molecular and other particle-based simulation models, which are commonly used in computational materials science and engineering, physics, and chemistry disciplines.

Steps to download and install OVITO:

1. **Go to:** https://www.ovito.org/#download
2. **Click:** Download.
3. **Choose:** your operating system.
4. **Click:** ”Download Basic”.
5. **When the download is complete, execute the file and follow the instructions.**

Installing OVITO on Ubuntu:

In Ubuntu, you don’t need to install **OVITO**. However, these steps can be followed to use it in Ubuntu.

Steps to download and install OVITO:

1. **Go to:** https://www.ovito.org/#download and download for the linux version.
2. **Unzip the file:** ”ovito-basic-3.11.2-x86_64” or whatever the version you have.
3. **Navigate to the directory**

```
ovito-basic-3.11.2-x86_64/bin
```

4. **Open the terminal and use the command to run**

```
./ovito
```

Instructions for the programming geeks:

If you are a hard-core UBUNTU user then thank you so much. You can follow the following steps to built the **LAMMPS** package using the **CMAKE**.

Caution: **Please remove any extra space, while copying these commands. The best way is to copy and paste these commands to CHATGPT and asked to rectify the commands. Copy those commands from the chatgpt output and paste in your terminal, and then press enter.**

1. Prerequisites

Dear students, before starting the **LAMMPS** installation process using **CMAKE**, please ensure that the following packages are installed on your system. Use the below commands to install them:

```
sudo apt update
sudo apt install -y build-essential cmake g++ gfortran git \
  libfftw3-dev libjpeg-dev libpng-dev libhdf5-dev libopenmpi-dev \
  liblapack-dev libblas-dev libpython3-dev libcurl4-openssl-dev \
  libzstd-dev gedit
```

Common Issues During Configuration

- Missing libraries such as `libkim-api.so` or `libhdf5`. Make sure these libraries are also installed. - Conflicts with Conda libraries. Use a non-Conda shell during configuration if needed.

Note: This list covers common dependencies needed for LAMMPS. If you enable additional packages (e.g., KIM or Voronoi), ensure their dependencies are installed. You can install these easily by using `sudo apt install package_name` command.

2. Download and Setup LAMMPS

Clone the LAMMPS repository and prepare the build directory:

```
# Clone the LAMMPS repository
git clone https://github.com/lammps/lammps.git
cd lammps

# Create a build directory
mkdir build
cd build
```

3. Configure LAMMPS with CMake

While being within the build directory you would need to use cmake

```
ccmake ../cmake
```

this command will lead to run cmake while copying the cmake.txt file from the cmake (”.. cmake” part of the last command) directory already available in the downloaded lammps folder.

Now please press the **”c”** button it will configure and then press **”e”** button to remake the configuration and click enter on the packages which you want to install and again press **”c”** to configure once all the necessary packages listed below are made **”on”**.

Note: Please avoid every module package enabling for the installation. You would just need to make these packages ”on” by clicking the enter button, where the list is given below...

Required LAMMPS Packages

ASPHERE ATC AWPMD BOCS BODY CG-DNA CG-SPICA CLASS2 COLLOID COMPRESS CORESHELL DIELECTRIC DIFFRACTION DIPOLE DPD-BASIC DPD-MESO DPD-REACT DPD-SMOOTH DRUDE EFF ELECTRODE EXTRA-PAIR FEP GRANULAR H5MD KIM KSPACE LATBOLTZ MANIFOLD MANYBODY MC MEAM MGPT MISC MOFFF MOLECULE MOLFILE NETCDF OPENMP OPT ORIENT PERI PHONON POEMS PYTHON QEQ QTB REPLICA RIGID SHOCK SMTBQ SPH SRD TALLY UEF VORONOI

The ‘CMAKE_INSTALL_PREFIX‘ specifies where LAMMPS will be installed. If the configuration fails, ensure that all required dependencies (see Section 1) are installed.

4. Build and Install LAMMPS

Compile and install LAMMPS:

```
make -j$(nproc)     # Compile using all available cores
make install        # Install to the specified directory
```

5. Make LAMMPS Executable Globally Available

To make the LAMMPS executable accessible from any directory:

1. Add the installation directory to your `PATH` variable. Edit your `~/.bashrc` file:

   ```
       nano ~/.bashrc
   ```

   You can also use **gedit** command to open which would be easier to edit and save due to the user interface.

   ```
       gedit ~/.bashrc
   ```

2. Add the following line at the end of the file:

   ```
       export PATH=$PATH:/home/yourcomputername/user/local/bin
       export PATH=$PATH:~/user/local/bin
       export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/yourcomputername/lammps/
          build/kim_build-prefix/lib
   ```

   Caution: Please replace ”yourcomputername” with your own computer name.

3. Save and exit the editor. Then reload the `.bashrc` file:

   ```
       source ~/.bashrc
   ```

Test the Setup: Run the following command to verify:

```
lmp -h
```

This should display the help information for the LAMMPS executable.

6. Notes on Using LAMMPS with Conda

If you are using Conda, you may encounter library conflicts. To avoid this:

1. Either deactivate Conda before running LAMMPS:

```
    conda deactivate
```

2. Or ensure that the correct library paths are loaded by setting:

```
    export LD_LIBRARY_PATH=/home/yourcomputername/user/local/lib:
       $LD_LIBRARY_PATH
```

7. Summary

This guide provides a step-by-step process to install LAMMPS using CMake, resolve common issues, and configure the environment for ease of use. If you encounter additional issues, refer to the LAMMPS documentation or community forums for support.

8. In case of failure

Take help from the **CHATGPT** (free version can make it through). Follow step by step and carefully read the error message. Do not give up until installed and executable globally (from any directory) within the home environment.