A Guide for LAMMPS Installation

Molecular and Business Modelling

UCL NatSci Innovation Lab 2020

Lau Pak To (Ryan)
15/7/2020

Contents

To preface: Stack Overflow, Stack Exchange, Reddit, Quora, YouTube, as well as Google, are our good friends.

1 Setting up GitHub

It would be good practise to setup a GitHub account for sharing and uploading codes or scripts in the future.
GitHub is basically a Google Drive for programmers. It allows them to share or publish codes, as well as networking. There are some files in the official LAMMPS repository that are needed for installation. Moreover, there are good reference materials we can learn from some of the files, such as sample data or scripts.
It will be useful to create our own repository, and upload our codes or scripts into GitHub for sharing.

1.1 Setting GitHub account

1. Setup a GitHub account if you do not have one by clicking this link: https://github.com/

2. Follow the instructions given while signing up. You will need an email for it. Process will not take too long
   Important note: **Please remember your username and email used. You will need these information in the next section.**

1.2 Create a new repository

1. Select the green button saying "Create repository" on the left hand side of the screen
2. Enter your repository name and select the option: "Initialize this repository with a README". Finish off by clicking the green button "Create repository".
3. Click"Add File" and select "Create new file".
4. Name your file as *.txt (e.g. Code.txt)
   We we will be using text files to create our LAMMPS input code.
5. Enter some random text under "Edit new file".
6. Finish off with clicking the green button "Commit new file" at the bottom of the page.

2 Git Bash

Bash is a terminal application used to interface with an operating system. Git Bash can directly link our computers to our GitHub accounts.
We can use it to download, or in Git terms, "clone", files or repositories into our computers. We can upload, or "push" files into our GitHub accounts, as well as downloading, or "pull" existing peer edited files. Moreover, we can keep track of changes made by different users and previous versions of our codes.
For more information on commands: https://git-scm.com/docs/user-manual

2.1 Installation

2.1.1 Windows

Click on this link to get to the download page: https://git-scm.com/downloads

1. Click on "Windows" in the downloads section. The link will direct you to the downloading page.
2. The page will ask you to choose the location you want the application to be downloaded. The downloading process for your computer will be start automatically.
3. In your download location, click on the *.exe application. It will direct you to the Git Setup for installation.
   (The application name has the form similar to/same as: e.g. Git-2.27.0-64-bit)
4. Click "Next" for the GNU General Public Licence page.
5. Choose the destination folder you want to install Git in. Click "Next" when finished.
6. Keep selected components unchanged and click "Next".

7. Do not change anything on the “Select Start Menu Folder” page. Click ”Next”.

8. Keep the following as default and click “Next” for all:

    - Editor used by Git (selection: Vim)
    - Adjusting PATH environment (selection: Git from the command line and also from 3rd-party software)
    - Choosing HTTPS transport backend (Selection: Use the OpenSSL library)
    - Configuring the line ending conversions (Selection: Checkout Windows-style, commit Unix-style line endings)
    - Configuring the terminal emulator to use with GitBash (Selection: Use MinTTY)
    - Choose the default behaviour of ’git pull’ (Selection: Default)
    - Configuring extra options (Selections: Tick Enable file system caching and Enable Git Credential Manager)

9. (Optional) In the Configuring experimental options, you may choose to tick the box.

10. Click “Install” to start installation.

11. Launch Git Bash to complete the Git Setup Wizard.

Fig 1: Sample of a Bash console.

2.1.2 MacOS

Installation process will be easier than that of Windows. You will be using Terminal in Mac. Download link: https://brew.sh/
For more information on Homebrew: https://docs.brew.sh/

1. Click on the download link and run the command shown: (/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)")

2. Enter your password as asked.(Your password will not be visible in the console.)The following will be shown:

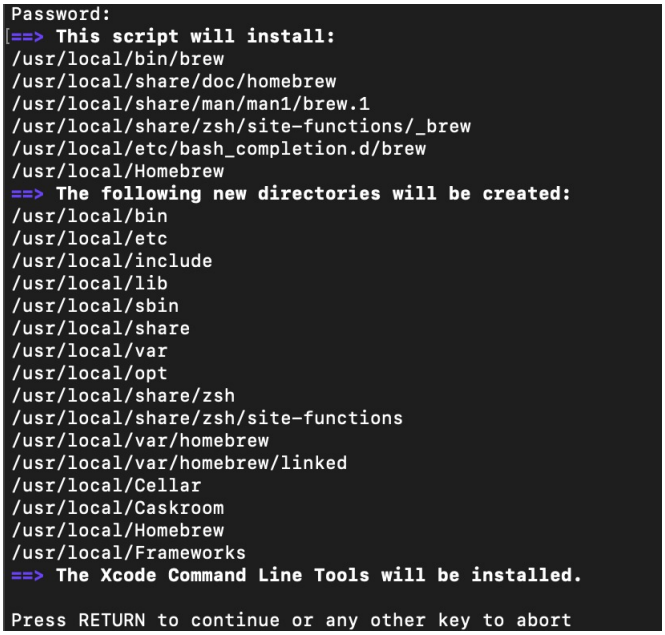

Fig 2
Press RETURN to continue. The required files will be downloaded.

3. When completed, the Terminal will show a list of commands as below:

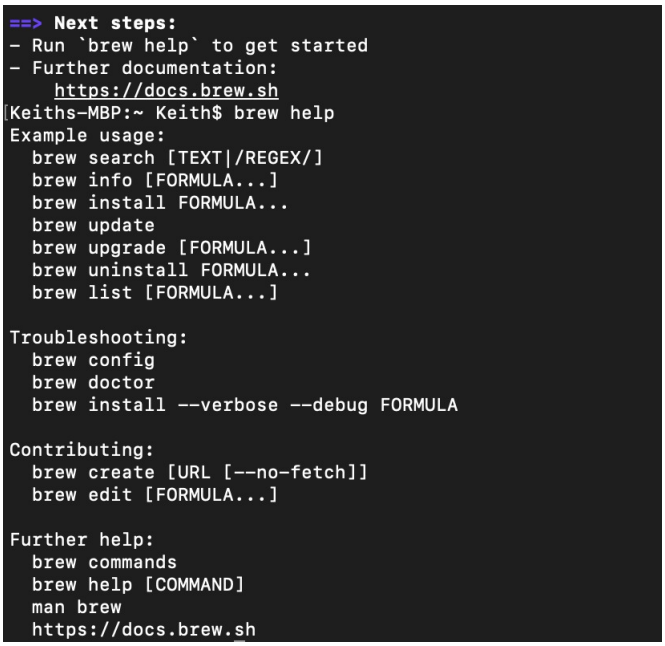

Fig 3

4. Enter: brew install git. You will be using Git Bash in the Terminal.

5. Installation is finished. Commands will be similar to that of Windows. I will refer Terminal as the Bash console, as they both do the same job.

2.2 Linking to GitHub (Windows and MacOS)

Here are some useful tips for this section.
*If you wish to navigate in the Bash console, use your arrow keys on your keyboard.
*Do not type in the dollar sign($) or percentage sign(%) at the start of the command. It has already been done for you in the Bash console.
*To execute/run a command, press ENTER.
*To paste something you copied on the console, use Shift+Ins
*To terminate a running process, use Ctrl + C
*For extra information on commands and more go to:
https://git-scm.com/docs

2.2.1 Cloning

1. Enter your username and email (same as the username and email used in signing up for GitHub) by typing in the following commands:

   - $ git config --global user.name "username"
   - $ git config --global user.email "email"

2. From your created repository page created in GitHub during section 1, click the green button which indicates to download the code. A mini-window will appear as Fig 2 shows:

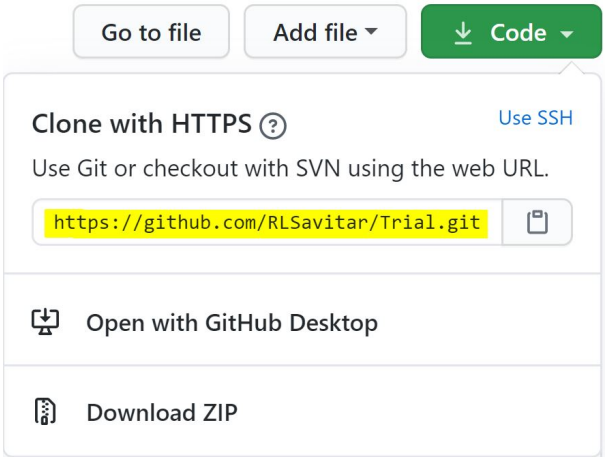

Fig 4

3. Copy the web URL (As highlighted in yellow)

4. In Bash console, run the following command: - $ git clone [URL]
   Replace [URL] with URL copied from GitHub.

5. Run the following command in Bash: - $ cd [Name] (e.g. cd Trial)
   Replace [Name] with your repository name.

6. Execute the following: - $ ls
   The console will show the files under the repository name. E.g.:

```
Ryan@LAPTOP-6RV84KT3 MINGW64 ~/Trial (master)
$ ls
code.txt  README.md
```

Fig 5

7. Run the command:
   - $ start *.txt (Windows)
   - $ open *.txt (MacOS)
   Replace the " * " with the text file's name. (e.g. code.txt)
   The file should open in Notepad(Windows) or Textmaker(MacOS).
   For Textmaker, remember to select the preference of editting text files rather than rich text files.

2.2.2 Adding, Committing & Pushing

1. Edit the contents of the text file in Notepad. Save it afterwards.

2. In Bash, enter the command: - $ git status
   It will show something similar to the following example:

```
Ryan@LAPTOP-6RV84KT3 MINGW64 ~/Trial (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   code.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Fig 6

3. In Bash, enter the command: - $ git add .
   It will "add" all the changed file contents into an index, or a "staging area".
   *The " . " represents "all".

4. Enter the command: - $ git commit -m"[message]"
   Replace [message] with a log message which you want to enter.
   The command will save changes to the local repositories.

5. Enter the command again: - $ git status
   An example of the output message from Bash is shown below in Fig 5:

```
Ryan@LAPTOP-6RV84KT3 MINGW64 ~/Trial (master)
$ git add .

Ryan@LAPTOP-6RV84KT3 MINGW64 ~/Trial (master)
$ git commit -m"Some random message"
[master 8fd0723] Some random message
 1 file changed, 2 insertions(+), 1 deletion(-)

Ryan@LAPTOP-6RV84KT3 MINGW64 ~/Trial (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

   Fig 7

6. Enter the command: $ git push
   This process will "push", or upload the file into your GitHub repository.
   For Windows: A GitHub window will pop-up as Fig 6a shows. Enter your email and password of your GitHub account.
   For MacOS: You will just need to enter your email and passwords in the Terminal as in Fig 6b.

Fig 8a

```
Keiths-MBP:testing Keith$ git push
Username for 'https://github.com': abc185
Password for 'https://abc185@github.com':
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 588 bytes | 588.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/abc185/testing.git
   1e7488e..dcb1dab  master -> master
```

Fig 8b: Including Push message

7. The console will display a similar message as shown below:

```
Ryan@LAPTOP-6RV84KT3 MINGW64 ~/Trial (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 316 bytes | 79.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/RLSavitar/Trial.git
   c68dda7..8fd0723  master -> master
```

Fig 9

8. Check your GitHub repository. You will see the message you wrote during committing in this section's step 4 beside the text file's name.

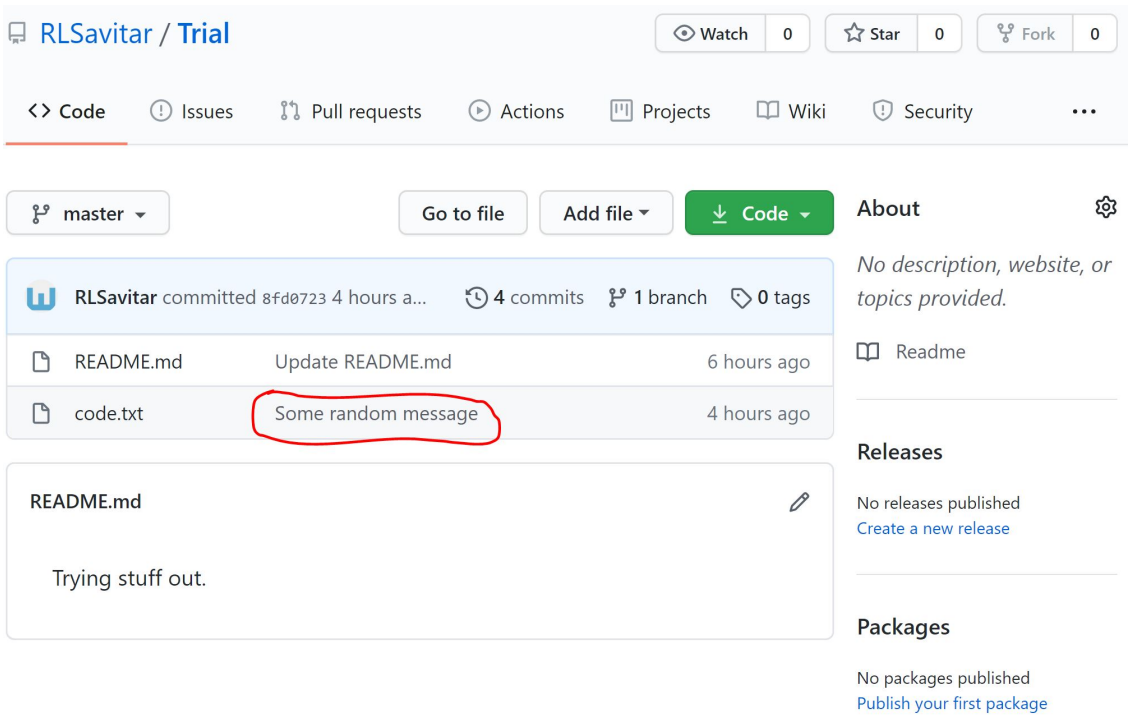

Fig 10: Example of the commit message as circled in red.

9. Click into the *.txt file. You will see the updated text.

Your computer will save your account details by default. Next time, if you are "pushing" scripts (Step 6), you will not need to enter your email and password again.

(If you would like to change or remove such information, go to: Control Panel → User Accounts → Manage your credentials → Windows Credentials → under Generic Credentials, there are credentials related to Github.)

We will use the command: - $ git pull, when we need to download peer-editted scripts from a shared repository.

3 Installing LAMMPS

The following link will be useful: https://lammps.sandia.gov/doc/Install.html. You may wish to explore the website.

3.1 Windows

Download link: http://packages.lammps.org/windows.html

1. Click on the "64-bit Windows download area" under "Installing LAMMPS" on Windows.

2. Select LAMMPS-64bit-latest-MPI.exe. Choose the location you wish the file to be downloaded. The *.exe file will be downloaded. Click on the *.exe file afterwards.

3. Choose the location you want the files to be installed in, then click "Install". After the files are installed, select "close". The installation should be complete.

3.2 MacOS

The following link explains clearl on how to download the LAMMPS using the MacOS Terminal. https://lammps.sandia.gov/doc/Install_mac.html

1. Enter: % brew install lammps
   The software will be downloaded.

2. Enter:% brew test lammps -v to test if LAMMPS is successfully installed. The file used will be a Lennard-Jones benchmark file. Simulation of the file will be run and the following will be shown in the Terminal:

```
Keiths-MBP:testing Keith$ brew test lammps -v
==> Testing lammps
/usr/bin/sandbox-exec -f /private/tmp/homebrew20200713-68583-vhds57.sb ruby -W0 -I $LOAD_PATH -- /usr/local/Homebrew/Library/Homebrew/test.rb /usr/local/Homebrew/Library/Taps/homebrew/homebrew-core/Formula/lammps.rb --verbose
==> /usr/local/Cellar/lammps/2020-03-03/bin/lmp_serial -in /usr/local/Cellar/lammps/2020-03-03/share/lammps/bench/in.lj
LAMMPS (3 Mar 2020)
Lattice spacing in x,y,z = 1.6796 1.6796 1.6796
Created orthogonal box = (0 0 0) to (33.5919 33.5919 33.5919)
  1 by 1 by 1 MPI processor grid
Created 32000 atoms
  create_atoms CPU = 0.00381613 secs
Neighbor list info ...
  update every 20 steps, delay 0 steps, check no
  max neighbors/atom: 2000, page size: 100000
  master list distance cutoff = 2.8
  ghost atom cutoff = 2.8
  binsize = 1.4, bins = 24 24 24
  1 neighbor lists, perpetual/occasional/extra = 1 0 0
  (1) pair lj/cut, perpetual
      attributes: half, newton on
      pair build: half/bin/atomonly/newton
      stencil: half/bin/3d/newton
      bin: standard
Setting up Verlet run ...
  Unit style    : lj
  Current step  : 0
  Time step     : 0.005
Per MPI rank memory allocation (min/avg/max) = 13.82 | 13.82 | 13.82 Mbytes
Step Temp E_pair E_mol TotEng Press
       0          1.44   -6.7733681            0   -4.6134356   -5.0197073
     100    0.7574531   -5.7585055            0   -4.6223613    0.20726105
Loop time of 2.01734 on 1 procs for 100 steps with 32000 atoms

Performance: 21414.326 tau/day, 49.570 timesteps/s
98.1% CPU use with 1 MPI tasks x no OpenMP threads

MPI task timing breakdown:
Section |  min time  |  avg time  |  max time  |%varavg| %total
---------------------------------------------------------------
Pair    | 1.7059     | 1.7059     | 1.7059     |   0.0 | 84.56
Neigh   | 0.21773    | 0.21773    | 0.21773    |   0.0 | 10.79
Comm    | 0.031848   | 0.031848   | 0.031848   |   0.0 |  1.58
Output  | 0.00010395 | 0.00010395 | 0.00010395 |   0.0 |  0.01
Modify  | 0.054302   | 0.054302   | 0.054302   |   0.0 |  2.69
Other   |            | 0.007437   |            |       |  0.37

Nlocal:    32000 ave 32000 max 32000 min
Histogram: 1 0 0 0 0 0 0 0 0 0
Nghost:    19657 ave 19657 max 19657 min
Histogram: 1 0 0 0 0 0 0 0 0 0
Neighs:    1.20283e+06 ave 1.20283e+06 max 1.20283e+06 min
Histogram: 1 0 0 0 0 0 0 0 0 0

Total # of neighbors = 1202833
Ave neighs/atom = 37.5885
Neighbor list builds = 5
Dangerous builds not checked
Total wall time: 0:00:02
```

Fig 11: Output of the testing file

3.3 More LAMMPS resources

For more resources and files in LAMMPS, you may wish to download, or "clone" them in the official public LAMMPS repository which can be found here:
https://github.com/lammps/lammps
Follow the steps of "Cloning" in the previous section.

4 Installing OVITO

(This software is for Windows and macOS users only, as the software is already in the program archive for Linux users.)

OVITO is a scientific software used for visualisation and analysis for atomistic and particle simulation data. It is a recommended tool for LAMMPS visulisation. Steps are as follows:

1. Go to https://www.ovito.org/
2. Select your computer's operating system, namely Windows or macOS.
3. For all operating systems, you will only need OVITO Basic
   If your operating system is:
   - Windows
     (a) Click on the ovito-basic-[version number]-win64.exe file. Software will be downloaded. (Example of version number: 3.1.1)
     (b) Go to your directory of which the file has been downloaded, click the "ovito-basic-[version number]-win64", which will bring you to the OVITO Basic Setup.
     (c) Click "I Agree" to agree on the Licence Agreement.
     (d) Select the folder you want the software to be installed in. Click "Next" to proceed.
     (e) You will be directed to a Choose Start Menu Folder. Do not change anything and click "Next".
     (f) Select Ovito. A green tick in the box will be indicated. Click "Install". The program will be installed.
     (g) Click "Finish" to exit the setup.
   - macOS
     – Click on the ovito-basic-[version number]-macos.dmg file. It will be downloaded.
     – Click "I Agree" on the Licence Agreement.
     – When download is finished, drag the application to the "Applications" folder. An example can be seen below in Fig 14.

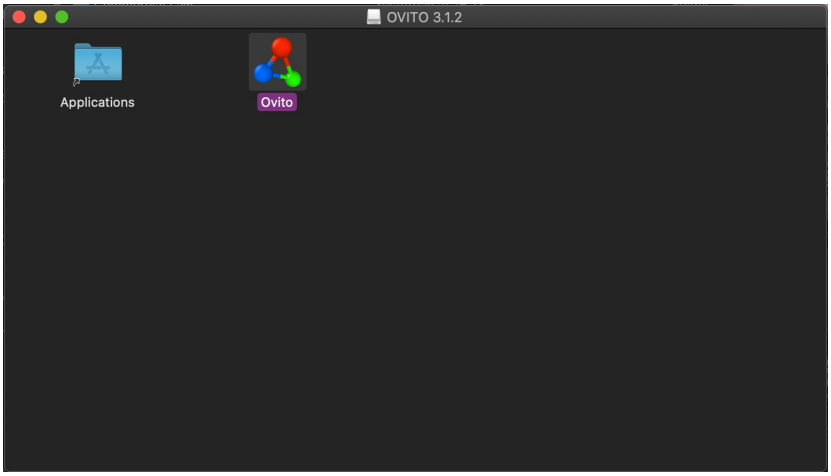

Fig 14: Example of Mac's application folder.

4. To open OVITO:
    - Go to the folder in which you installed the software.
    - Click on the application file named “Ovito”, which has a 3-coloured triangle by its name.
    - The software should be opened successfully.

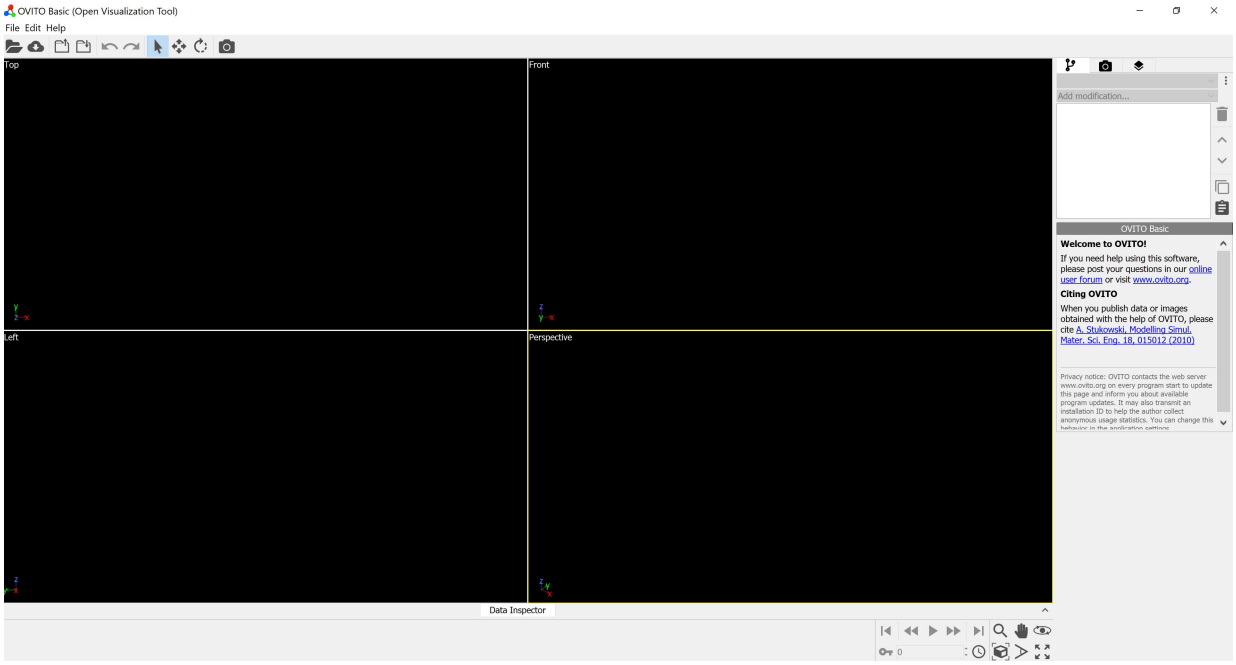

Fig 15: Example of an OVITO interface

5 Installing Avogadro

Avogadro is an open-source molecular editor and visualization tool. It is compatible for Windows, MacOS, and Linux. It is extremely helpful in molecular modelling, computational chemistry, materials science and vice versa.
Avogadro can help us write input or data files for our LAMMPS simulations.
For more information and documentations: https://avogadro.cc/docs/
https://avogadro.cc/ Here are the steps to download the application:

1. Click on this link:
   https://sourceforge.net/projects/avogadro/files/latest/download
   The download will start automatically.
2. Click on the downloaded file. A setup wizard will pop up.
3. Select “Next”, then “I Agree”
4. For Install Options section, choose the default option(i.e. Do not add Avogadro to the system PATH), and then tick whether you want to create an Avogadro desktop icon. Clcik “Next” once you have finished.
5. For Choose Install Location section, choose the directory you want the application to be installed. Click “Next” to continue.
6. Click Install for the next section. The program will be installed.