MEEP Installation

This documents my method for installing MEEP
(**M**IT **E**lectromagnetic **E**quation **P**ropagation) via WSL.

As I have not tested this thoroughly on other machines than my own, methods for your individual installation may differ and problems may occur.

As such, I have linked the individual installation guide for each step.

Main MEEP Installation Page

This document will focus on installation of the MEEP python package via https://meep.readthedocs.io/en/latest/Installation/ but notes a few key differences in installation.

MEEP works solely with Unix systems, and so native Windows installation is unsupported.

Installation then either involves installing Windows Subsystem for Linux (WSL), or dual-booting Linux onto your device along with your Windows installation.

The method I have used to install MEEP was through WSL, so that is used here, though the steps forward are identical for Linux and WSL – just the method of dual booting Linux is not discussed, and can be investigated elsewhere.

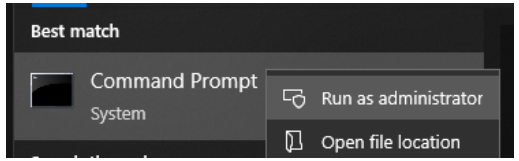
Also, for dual-boot, jupyter is not necessarily required, as discussed in the MEEP installation page linked above. Similarly any mention of anything specific to Windows is not required.

WSL Installation

Main Documentation: https://learn.microsoft.com/en-us/windows/wsl/install

WSL may only be installed via this method on Windows 10 version 2004 or higher, or Windows 11, the steps in the main documentation should be followed if you possess an earlier version.

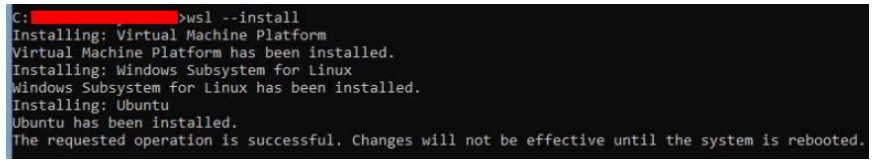
To install WSL, Command Prompt (The Windows CLI) must be opened in administrator mode.



WSL is installed in the Windows Command Line Interface (CLI) via running the command:

wsl --install

Which should return the following lines:



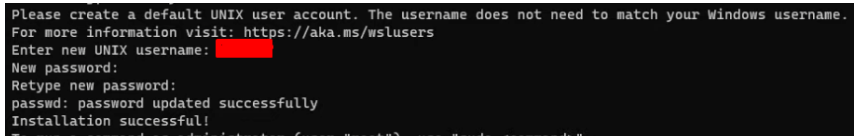
The default Linux Distribution this will install onto your device is Ubuntu, and may be changed at will if desired.

Once the installation has completed, the computer like stated must be rebooted to observe the change.

After reboot, Ubuntu may be started via opening the same CLI in administrator mode and typing ‘wsl’. Alternatively, it may be started by finding and double clicking the ‘Ubuntu’ application in the file list of the start menu.

On initial start, the CLI will prompt for a username for the system – this can be anything, and is only needed for logging in (the system typically automatically logs in on start anyway).

The CLI will then prompt for a password. Whilst typing your password nothing will appear on screen (blind typing), so ensure all keypresses are correct. The password is confirmed by pressing enter. (The password must also be typed again when prompted to retype new password).



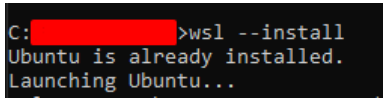
The installation should then be complete, and Ubuntu should open, indicated by the green prefix in the CLI.

It is advisable to run the commands ‘sudo apt-get update’ and ‘sudo apt-get upgrade’ to ensure everything is up to date in the installation.

To then exit Ubuntu, if desired, type ‘exit’. This will return you back to the regular windows grey CLI.

WSL may then be activated again by typing ‘wsl’ in the CLI.

Windows 11 may come with WSL pre-installed, if so running wsl –install should return:



This will then launch the Ubuntu distribution, in which the setup may continue as described.

Miniconda installation

Now WSL is installed, installation follows exactly as if using a Linux machine.

MEEP is recommended to be installed using conda, the python package manager.

Miniconda, the barebones installation of conda and python that is required, may be installed as follows:

Open the CLI, run ‘wsl’ to start Ubuntu. Then go to https://docs.anaconda.com/miniconda/, scroll to the bottom under ‘Quick Command line install’ and select Linux. The instructions are then described there for the installation, but are also as follows:

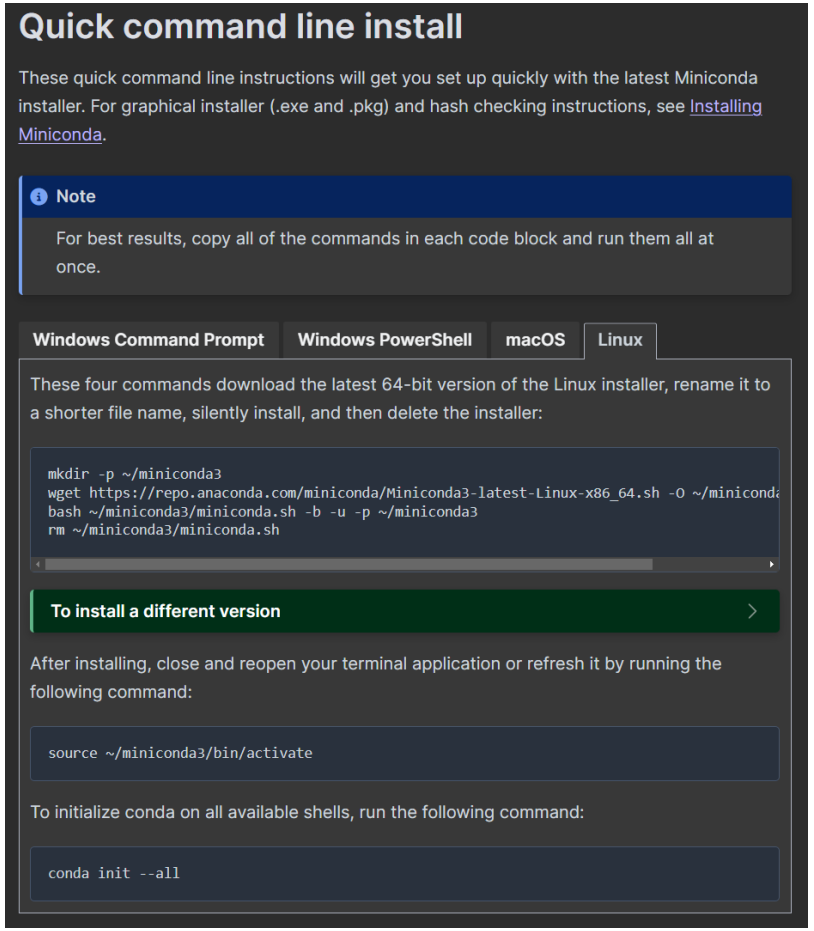
Run the paragraph:

‘mkdir -p ~/miniconda3 wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda3/miniconda.sh bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3 rm ~/miniconda3/miniconda.sh’

As described on the page linked above (the page link will always be the most up to date resource).

Once this has been run, as recommended run ‘source ~/miniconda3/bin/activate’ to refresh the terminal, and then run ‘conda init –all’ to initialize conda on all available shells.

I have chosen to install conda this way, differing from that recommended by MEEP, as I had problems with MEEP’s method of installation.



MEEP Installation

Main Documentation: https://meep.readthedocs.io/en/latest/Installation/

Now that miniconda is installed, a conda environment can be created to include the required packages of pymeep (MEEP python packages), pymeep-extras (extra packages), and jupyter (Adds jupyter notebooks, to use as a python IDE).

This is done via running in Ubuntu:

conda create -n mp -c conda-forge pymeep pymeep-extras jupyter

Which creates an environment called ‘mp’ using channel (-c) conda-forge to find and install the packages pymeep, pymeep-extras, and jupyter. Installing meep automatically installs numpy and matplotlib.

After following the given instruction in the CLI, the environment may be ‘activated’ using:

conda activate mp

This will activate your ‘mp’ environment (indicated by (mp) before the prefix), in which you may then run ‘jupyter notebook’ to start a jupyter notebook server. Placing the link provided by the CLI in your browser should show the jupyter interface, in which new python files may then be created.

Finally, a new python file may be created, and you may follow the basic tutorials provided by MEEP in https://meep.readthedocs.io/en/latest/Python_Tutorials/Basics/ to begin learning how to use the package.

If you need any extra packages added to the conda environment, first ensure the environment is open ((mp) before prefix), and you may run:

conda install package-name

Where package-name is the name of the package to be installed.

The mp conda environment may be deactivated by running **conda deactivate mp**

Further

All documentation used may be found at the following links:

WSL: https://learn.microsoft.com/en-us/windows/wsl/install

MEEP: https://meep.readthedocs.io/en/latest/Installation/

Miniconda: https://docs.anaconda.com/miniconda/

Links here are to the specific pages, but the documentations are thorough and may be explored for other problems.

MEEP also allows installation of parallel pymeep, by in the **conda create** command replacing 'pymeep' with 'pymeep=*=mpi_mpich_*', or may be added to a pre-existing environment by alternatively running the command in your environment:

conda install –c conda-forge pymeep=*=mpi_mpich_*

MEEP may require you to also install nomkl, which prevents the use of MKL (Math Kernal Library) of which numpy typically functions with, instead using OpenBLAS. This may be completed by running within your environment:

conda install conda-forge::nomkl

There is discussion on the MEEP installation page as to when this is required.

Also, it is recommended to install the 'terminal' application from the Microsoft Store on Windows if it is not installed already, as this application allows the easy management of multiple terminals via tabs.

This is useful, as converting h5 files to png requires use of the base Linux terminal – which I have found is not typically accessible while a jupyter notebook is running. As such, terminal allows for a second Linux terminal to be opened, to perform these operations in.

Finally, if it is desired to turn the output pngs to a gif format, the MEEP tutorials use ImageMagick convert, which can also be installed via Miniconda:

conda install conda-forge::imagemagick

The developers of MEEP have kindly requested you cite the following paper in any published works featuring MEEP:

A. Oskooi, D. Roundy, M. Ibanescu, P. Bermel, J.D. Joannopoulos, and S.G. Johnson, MEEP: A flexible free-software package for electromagnetic simulations by the FDTD method, Computer Physics Communications, Vol. 181, pp. 687-702 (2010) (pdf).