Aksimentiev Group
Department of Physics and
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign

Introduction to MD simulations of Plasmonic Nanotweezers

Authors:
Chen-Yu Li
Aleksei Aksimentiev

Contents

1 Introduction

Proteins are one of the most important macromolecules in living cells. Most of the vital activities inside living organisms are carried out by proteins, including catalyzing metabolic reactions, DNA replication, responding to stimuli, and transporting molecules from one location to another. Proteins usually consist of one or more polypeptides. A polypeptide is a long linear chain of amino acid residues. The amino acid sequences of each polypeptide determine the structure and function of the protein. Therefore, determining the sequence of proteins in living organisms is an important tool for understanding cellular processes, and allows drugs that target specific metabolic pathways to be invented more easily.

Current protein sequencing methods suffer from limitations associated with the size of proteins that can be sequenced, the time, and the cost of the sequencing procedures. Nanometer-scale holes in synthetic membranes, or simply solid-state nanopores, have emerged as a promising platform for inexpensive, reliable protein sequencing. In a typical setup, a membrane with a nanopore is placed in a solution and an applied electric bias drives charged molecules such as polypeptide and ions through the hole. The idea of the nanopore-based protein sequencing method is to read the sequence of polypeptides in a molecule as it moves through the pore. Typically employed ionic current measurements are, however, not sensitive enough to detect the minute differences in the organization of atoms in the polypeptides. Moreover, fast polypeptide motion through synthetic nanopores makes it difficult to obtain clear signals needed to extract the information about polypeptide sequence. Previously, our group has demonstrated that plasmonic nanopores engage nanoplasmonics in controlling the electrophoretic motion of DNA through the pore using Molecular Dynamics (MD) simulation [1].

In this tutorial, we will demonstrate how to manipulate the translocation of a polypeptide through a solid-state nanopore with a plasmonic field in MD simulation. We will start from obtaining the 3D plasmonic field distribution using nanoDDSCAT+ (`https://nanohub.org/tools/ddaplus`) on nanoHUB. Then, apply the 3D plasmonic field distribution in MD simulation to trap the translocation of a charged polypeptide in an electric field using NAMD (`http://www.ks.uiuc.edu/Research/namd/`), a parallel molecular dynamics simulation program. Finally, we will use a molecular visualization software VMD (`http://www.ks.uiuc.edu/Research/vmd/`) to see the simulation trajectory and analyze the result. It is assumed that the reader of this tutorial has some basic knowledge about interacting with the UNIX command-line interface, like `cd, ls, mkdir, cp` and so on. If you encountered any problems or bugs while using this tutorial, please don't hesitate to contact Aksimentiev group (`http://bionano.physics.illinois.edu/people/aleksei-aksimentiev`).

Here is a complete list of prerequisites:

- A Java-enabled web browser (section 2)
- A command-line interface terminal (section 3)

- Python 2.7 (with following modules installed through conda or pip) (section 3)
  - NumPy 1.11.1 (conda or pip)
  - vtk 6.3.0 (conda)
  - MDAnalysis 0.15.0 (pip)
- VMD program (section 3)
- NAMD program (section 3.3, 3.4 and 3.5)

Do not worry if you couldn't successfully install all of them! We provided step-by-step instruction as well as outputs in the corresponding "example_output" folder. So, if you have issues with any of the steps, you can still continue the tutorial by using the example files in the "example_output" folder. **However, it is recommended to have VMD installed in order to see the simulation results.** We will briefly describe how to install Python 2.7, VMD and NAMD in section 1.2, 1.3 and 1.4, respectively.

1.1 Content of the tutorial files

The hierarchy of subdirectories extracted from "Tutorial.tar.gz":

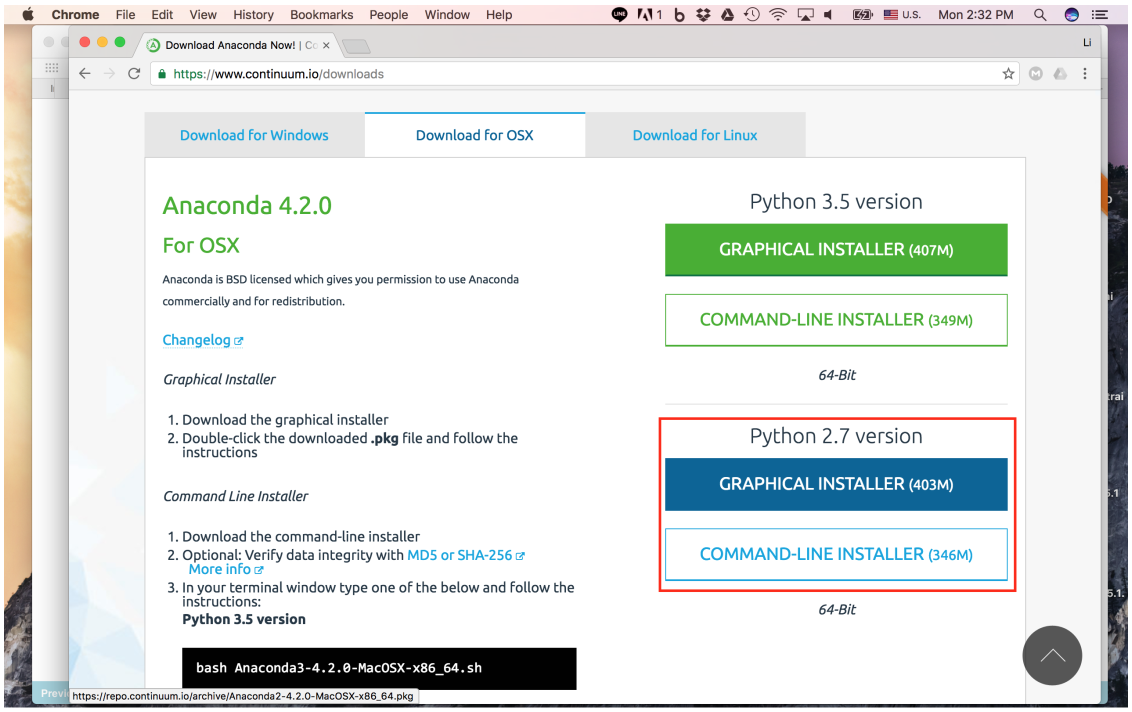
- Tutorial
  - ch2
    - example_output
      - DipolesPerNM0.2.vtk
      - DipolesPerNM0.5.vtk
      - DipolesPerNM1.vtk
      - DipolesPerNM1_wall.dx
      - bowTie.blend
  - ch3
    - Efield_500mV_double.dx
    - FDFD.pdb
    - FDFD.psf
    - FDFD_Efield.pdb
    - FDFD_plasmonic.pdb
    - FDFD_wall.pdb
    - example_output
      - DipolesPerNM1_plasmonic_double.dx

    - DipolesPerNM1_wall_clean_shift_double.dx
    - plasmonicOff.0.COM.dat
    - plasmonicOff.0.dcd
    - plasmonicOff.0.xst
    - plasmonicOn.0.COM.dat
    - plasmonicOn.0.dcd
    - plasmonicOn.0.xst
    - plasmonicSwitch.0.COM.dat
    - plasmonicSwitch.0.dcd
    - plasmonicSwitch.0.xst
    - plasmonicSwitch.1.COM.dat
    - plasmonicSwitch.1.dcd
    - plasmonicSwitch.1.xst
    - plasmonicSwitch.2.COM.dat
    - plasmonicSwitch.2.dcd
    - plasmonicSwitch.2.xst
    - plasmonicSwitch.3.COM.dat
    - plasmonicSwitch.3.dcd
    - plasmonicSwitch.3.xst
  - measureCOM.tcl
  - par_all36_prot.prm
  - plasmonicOff.0.namd
  - plasmonicOff_COM.sh
  - plasmonicOff_v.tcl
  - plasmonicOn.0.namd
  - plasmonicOn_COM.sh
  - plasmonicOn_v.tcl
  - plasmonicSwitch.0.namd
  - plasmonicSwitch.1.namd
  - plasmonicSwitch.2.namd
  - plasmonicSwitch.3.namd
  - plasmonicSwitch_COM.sh
  - plasmonicSwitch_v.tcl
  - shiftDX.py
  - vtkToDx.py

Each subdirectory (ch2 and ch3) contains the associated files for each section. If your archive does not contain all the files in the list, please download the latest version from the Aksimentiev group website (`http://bionano.physics.illinois.edu/`).

1.2 Install Python

In this section, we will briefly describe how to install Python 2.7 from Anaconda and other essential modules for this tutorial.


Chrome File Edit View History Bookmarks People Window Help
Mon 2:32 PM
Download Anaconda Now! | Co ×
https://www.continuum.io/downloads
Download for Windows
Download for OSX
Download for Linux
Python 3.5 version
GRAPHICAL INSTALLER (407M)
COMMAND-LINE INSTALLER (349M)
64-Bit
Python 2.7 version
GRAPHICAL INSTALLER (403M)
COMMAND-LINE INSTALLER (346M)
64-Bit
Changelog
MD5 or SHA-256
More info
https://repo.continuum.io/archive/Anaconda2-4.2.0-MacOSX-x86_64.pkg

Anaconda 4.2.0

For OSX

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

Graphical Installer

1. Download the graphical installer
2. Double-click the downloaded **.pkg** file and follow the instructions

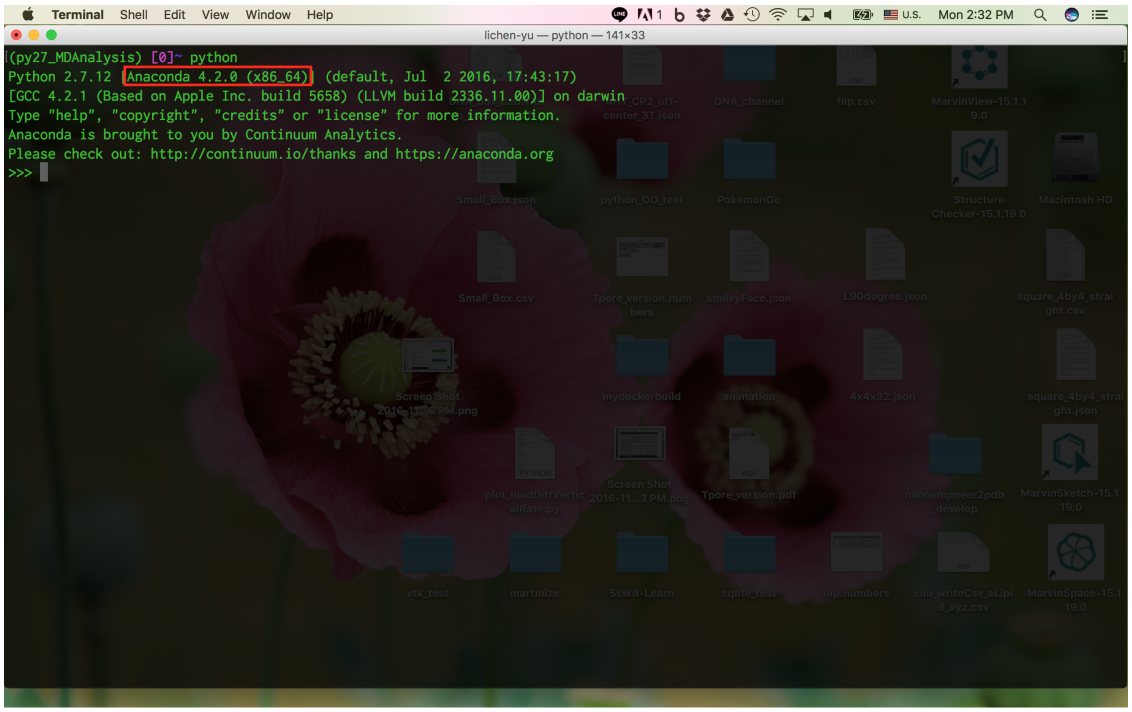
Command Line Installer

1. Download the command-line installer
2. Optional: Verify data integrity with MD5 or SHA-256 More info
3. In your terminal window type one of the below and follow the instructions:
   Python 3.5 version

```
bash Anaconda3-4.2.0-MacOSX-x86_64.sh
```

Figure 1: Install Anaconda Python 2.7 through the graphical installer or the command-line installer.

First, go to the Anaconda website. Choose your operating system and Python 2.7, Figure 1. You can use the graphical installer or the command-line installer. For users who are not familiar with the command-line interface, it is recommended to use the graphical installer. After the installation is finished, you will be asked whether to put the Anaconda Python 2.7 path to your .bash_rc file or not. Choose "yes" will allow you to launch Anaconda Python 2.7 without specifying the complete path.


Terminal Shell Edit View Window Help
Mon 2:32 PM
lichen-yu — python — 141×33

```
(py27_MDAnalysis) [0]~ python
Python 2.7.12 |Anaconda 4.2.0 (x86_64)| (default, Jul  2 2016, 17:43:17)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2336.11.00)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> 
```

Figure 2: Make sure you are using Python 2.7 from Anaconda.

Next, open a command-line terminal. Type “python” in your terminal to initiate the python interpreter, Figure 2. If you see something like Anaconda 4.2.0, red box in Figure 2, you are using Anaconda Python 2.7 and the installation completed successfully. You can exit the python interpreter by pressing “ctrl+D”.

Finally, we will install the modules needed. The NumPy module comes with Anaconda Python 2.7, so we don’t need to install it manually. But, we do need to install vtk and MDAnalysis. Type the following command in the terminal to install vtk:

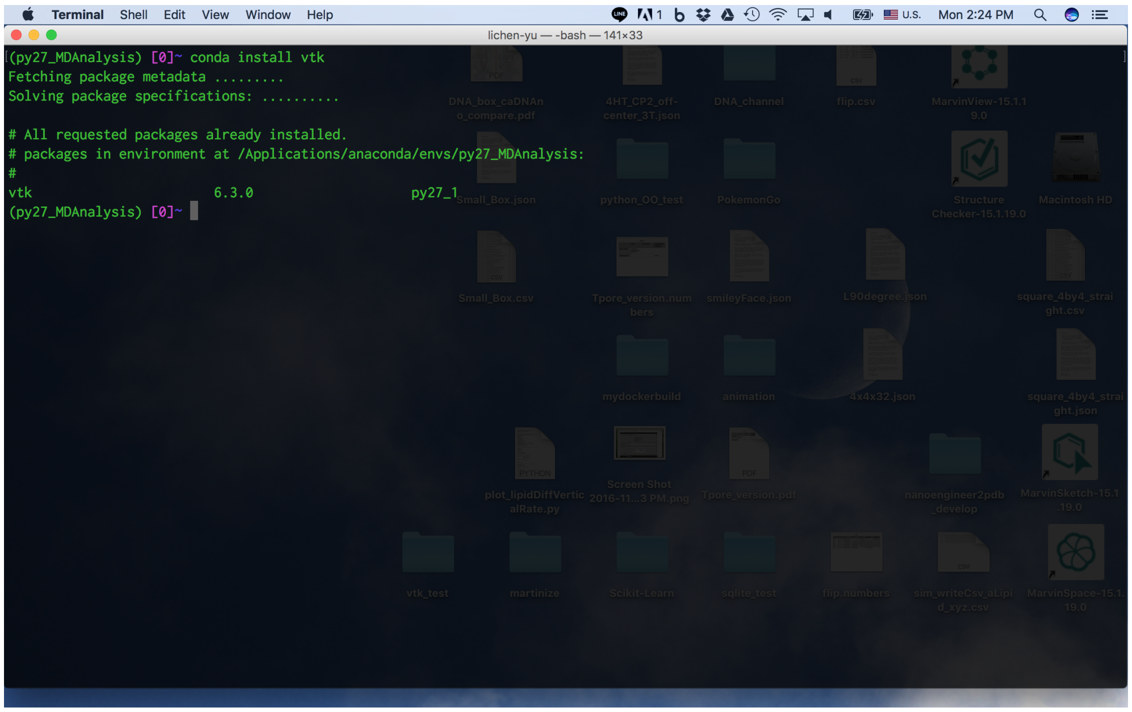
```
conda install vtk
```

If you do not have vtk installed, the above command will initiate the installation process. After the installation is complete, if you type the command again, you should see a result like Figure 3.

Similarly, use the following command in the terminal to install MDAnalysis:

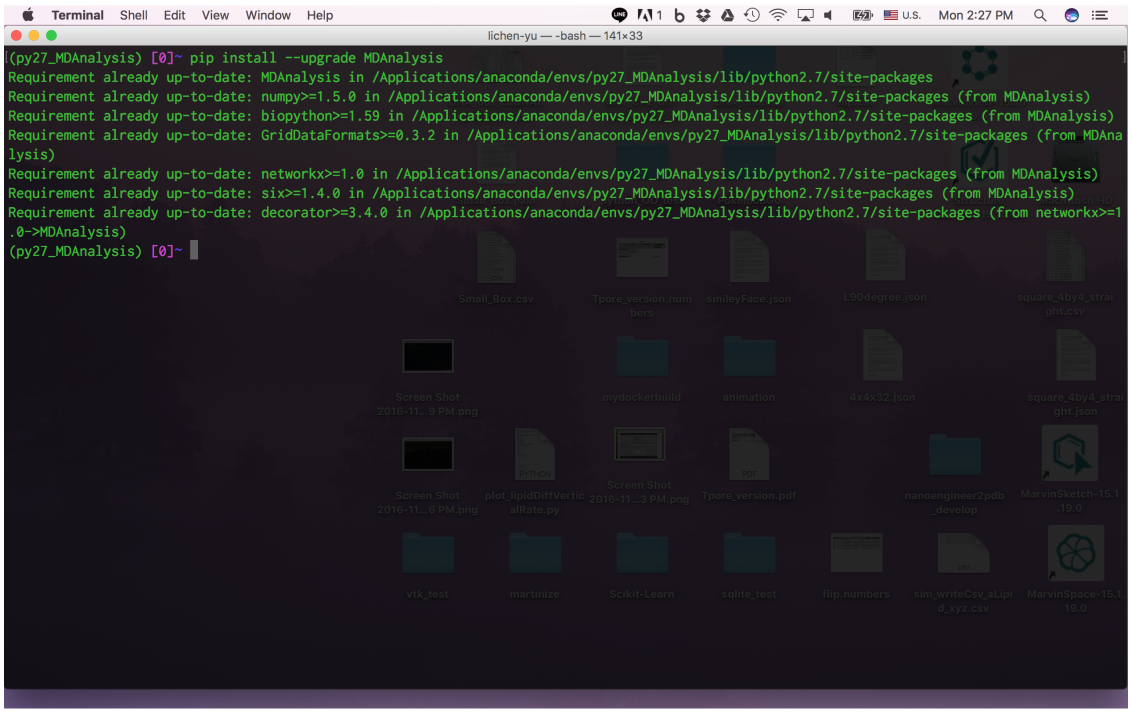
```
pip install --upgrade MDAnalysis
```

After the installation is complete, if you type the command again, you should see a result like Figure 4.



```
(py27_MDAnalysis) [0]~ conda install vtk
Fetching package metadata .........
Solving package specifications: ..........

# All requested packages already installed.
# packages in environment at /Applications/anaconda/envs/py27_MDAnalysis:
#
vtk                       6.3.0                    py27_1
(py27_MDAnalysis) [0]~
```

Figure 3: Install vtk.



```
(py27_MDAnalysis) [0]~ pip install --upgrade MDAnalysis
Requirement already up-to-date: MDAnalysis in /Applications/anaconda/envs/py27_MDAnalysis/lib/python2.7/site-packages
Requirement already up-to-date: numpy>=1.5.0 in /Applications/anaconda/envs/py27_MDAnalysis/lib/python2.7/site-packages (from MDAnalysis)
Requirement already up-to-date: biopython>=1.59 in /Applications/anaconda/envs/py27_MDAnalysis/lib/python2.7/site-packages (from MDAnalysis)
Requirement already up-to-date: GridDataFormats>=0.3.2 in /Applications/anaconda/envs/py27_MDAnalysis/lib/python2.7/site-packages (from MDAna
lysis)
Requirement already up-to-date: networkx>=1.0 in /Applications/anaconda/envs/py27_MDAnalysis/lib/python2.7/site-packages (from MDAnalysis)
Requirement already up-to-date: six>=1.4.0 in /Applications/anaconda/envs/py27_MDAnalysis/lib/python2.7/site-packages (from MDAnalysis)
Requirement already up-to-date: decorator>=3.4.0 in /Applications/anaconda/envs/py27_MDAnalysis/lib/python2.7/site-packages (from networkx>=1
.0->MDAnalysis)
(py27_MDAnalysis) [0]~
```

Figure 4: Install MDAnalysis.

1.3 Install VMD

In this section, we will briefly describe how to install VMD. VMD is a molecular visualization program for displaying, animating, and analyzing large biomolecular systems using 3-D graphics and built-in scripting. We will use VMD in the last section to visualize the simulation trajectory.

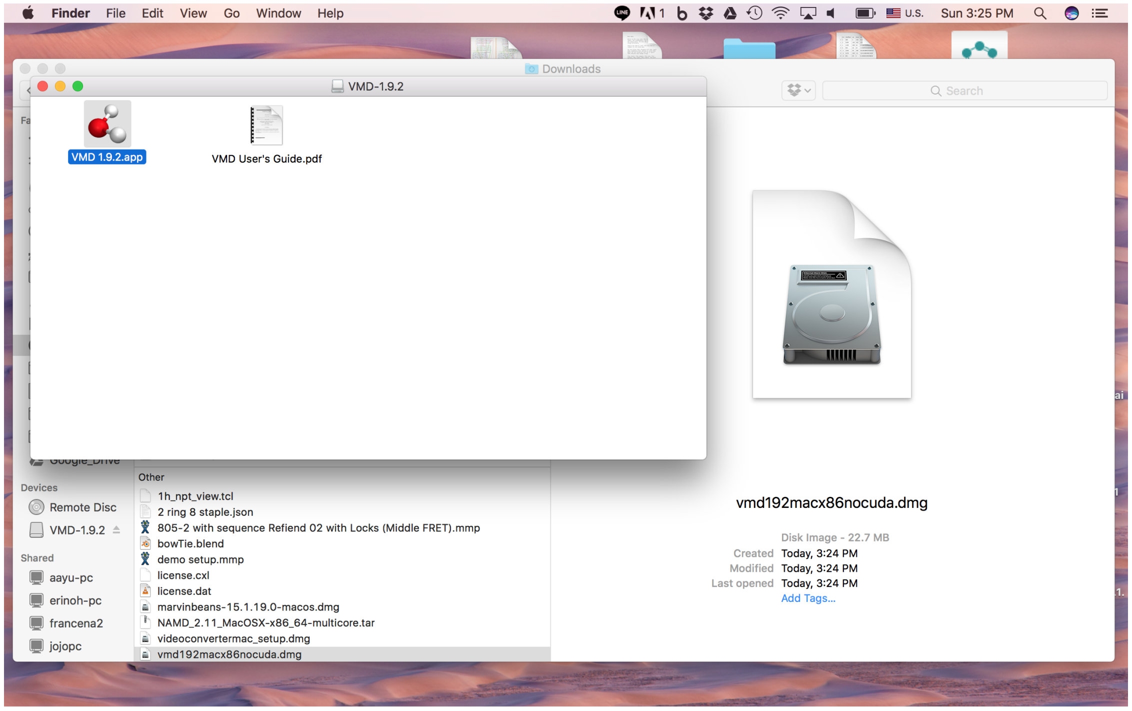


Figure 5: After extracting the archive, you should see the executable.

First, go to VMD's download page: http://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=VMD. Choose the latest stable version (without "beta") for your operating system. After creating an account, you will be able to download VMD. Extract the archive, and you should see the executable. Figure 5 is the example in MacOS. You should be able to launch VMD by double-clicking the executable or running it from the command-line.

To proceed with this tutorial smoothly, it is recommended to either put VMD in your search PATH or create an alias to the VMD binary. First, find where your VMD is installed. Then, you can create an alias by typing the following command with the correct path in the terminal.

Example for MacOS:

```
alias vmd=/Applications/VMD\ 1.9.2.app/Contents/MacOS/startup.command
```

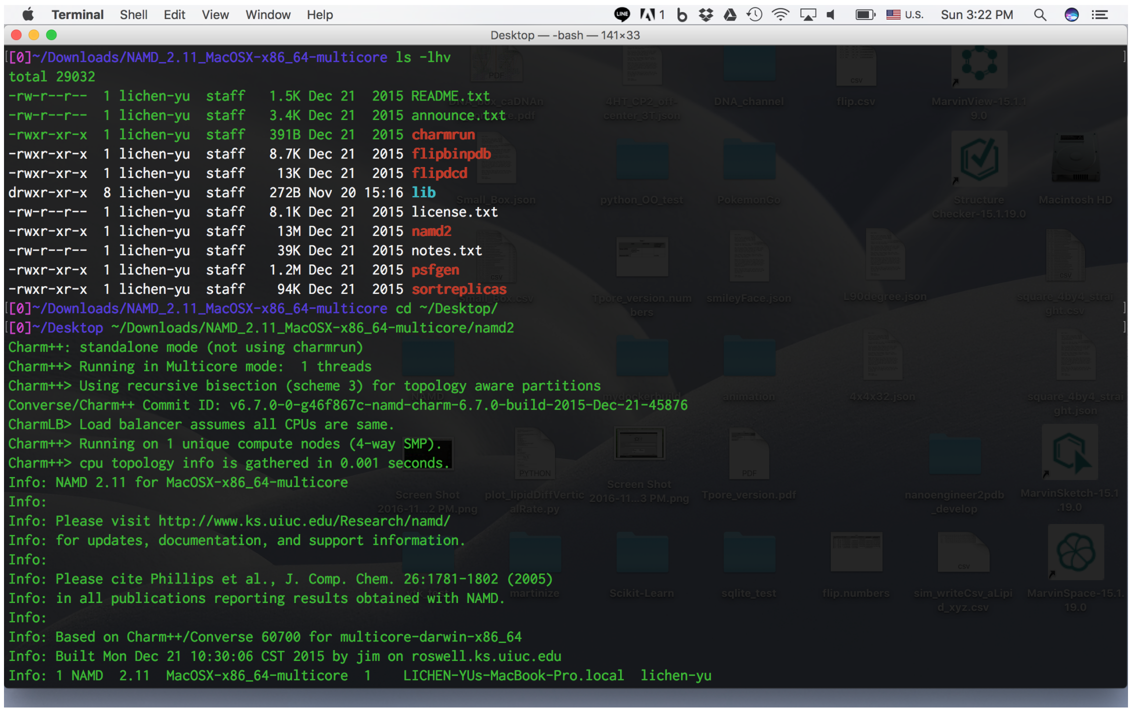
Example for Linux:

```
export PATH="$install_bin_dir:$PATH"
```

The $install_bin_dir is the binary directory you set during configuration.

For more details about VMD, please visit VMD User's Guide or VMD Tutorial.

1.4 Install NAMD



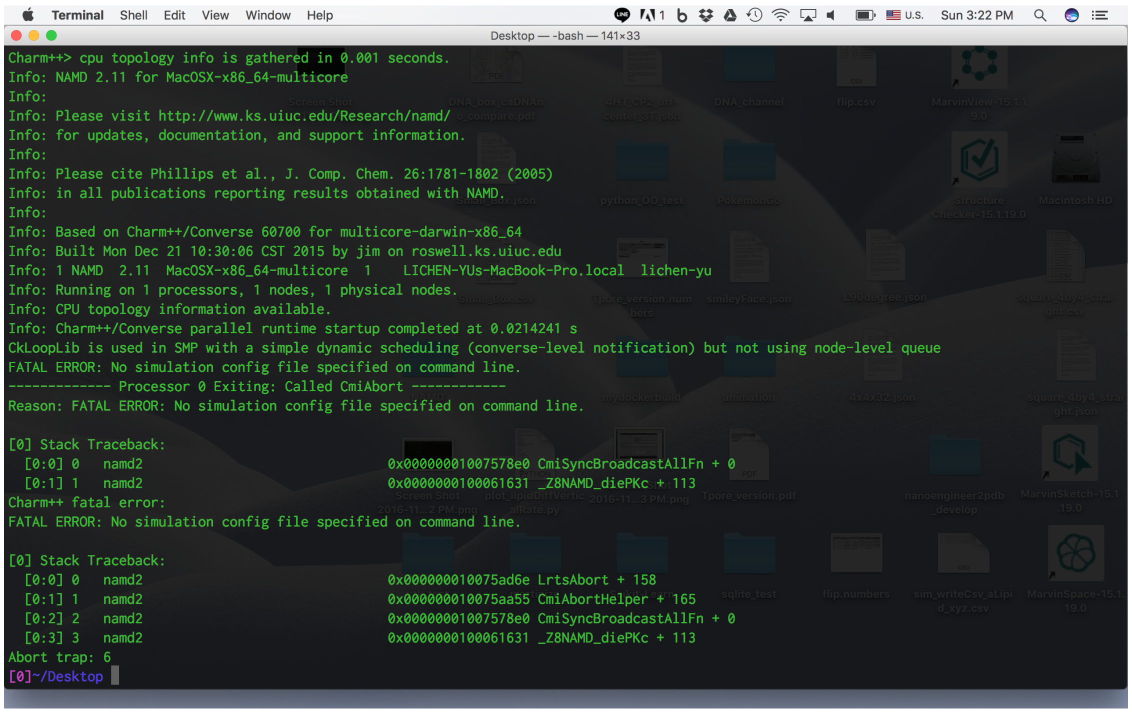
```
[0]~/Downloads/NAMD_2.11_MacOSX-x86_64-multicore ls -lhv
total 29032
-rw-r--r--  1 lichen-yu  staff   1.5K Dec 21  2015 README.txt
-rw-r--r--  1 lichen-yu  staff   3.4K Dec 21  2015 announce.txt
-rwxr-xr-x  1 lichen-yu  staff   391B Dec 21  2015 charmrun
-rwxr-xr-x  1 lichen-yu  staff   8.7K Dec 21  2015 flipbinpdb
-rwxr-xr-x  1 lichen-yu  staff    13K Dec 21  2015 flipdcd
drwxr-xr-x  8 lichen-yu  staff   272B Nov 20 15:16 lib
-rw-r--r--  1 lichen-yu  staff   8.1K Dec 21  2015 license.txt
-rwxr-xr-x  1 lichen-yu  staff    13M Dec 21  2015 namd2
-rw-r--r--  1 lichen-yu  staff    39K Dec 21  2015 notes.txt
-rwxr-xr-x  1 lichen-yu  staff   1.2M Dec 21  2015 psfgen
-rwxr-xr-x  1 lichen-yu  staff    94K Dec 21  2015 sortreplicas
[0]~/Downloads/NAMD_2.11_MacOSX-x86_64-multicore cd ~/Desktop/
[0]~/Desktop ~/Downloads/NAMD_2.11_MacOSX-x86_64-multicore/namd2
Charm++: standalone mode (not using charmrun)
Charm++> Running in Multicore mode:  1 threads
Charm++> Using recursive bisection (scheme 3) for topology aware partitions
Converse/Charm++ Commit ID: v6.7.0-0-g46f867c-namd-charm-6.7.0-build-2015-Dec-21-45876
CharmLB> Load balancer assumes all CPUs are same.
Charm++> Running on 1 unique compute nodes (4-way SMP).
Charm++> cpu topology info is gathered in 0.001 seconds.
Info: NAMD 2.11 for MacOSX-x86_64-multicore
Info:
Info: Please visit http://www.ks.uiuc.edu/Research/namd/
Info: for updates, documentation, and support information.
Info:
Info: Please cite Phillips et al., J. Comp. Chem. 26:1781-1802 (2005)
Info: in all publications reporting results obtained with NAMD.
Info:
Info: Based on Charm++/Converse 60700 for multicore-darwin-x86_64
Info: Built Mon Dec 21 10:30:06 CST 2015 by jim on roswell.ks.uiuc.edu
Info: 1 NAMD  2.11  MacOSX-x86_64-multicore  1    LICHEN-YUs-MacBook-Pro.local  lichen-yu
```

Figure 6: Here is the list of files inside the archive. You can launch NAMD from anywhere as long as you provide the absolute path of your NAMD binary.

In this section, we will briefly describe how to install NAMD. NAMD is a parallel, object-oriented molecular dynamics code designed for high-performance simulation of large biomolecular systems. We will use NAMD in section 3.3 and 3.4 to perform the MD simulation. However, if you encountered issues installing or using NAMD, we provided the simulation trajectory in "ch3/example_output". You can still visualize the simulation output and analyze the simulation results.

First, go to NAMD's download page: `http://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=NAMD`. Choose the latest stable version (without "beta"or "b") for your operating system. Using the same account when you install VMD, you will be able to download NAMD. Extract the archive, and you should see the executable, Figure 6.

There is no GUI for NAMD, so you will have to launch it from the command-line. You can launch NAMD from anywhere as long as you provide the absolute path of your NAMD binary. If you do not provide the NAMD configuration file, you would see the error message as in Figure 7.

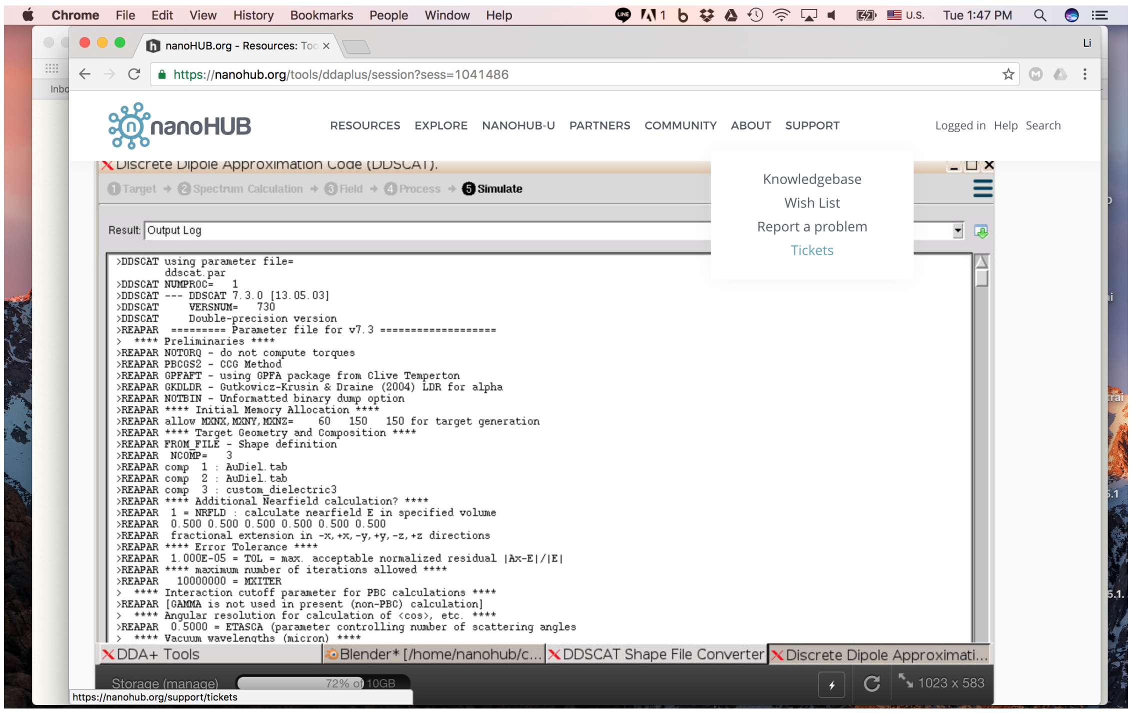


```
Charm++> cpu topology info is gathered in 0.001 seconds.
Info: NAMD 2.11 for MacOSX-x86_64-multicore
Info:
Info: Please visit http://www.ks.uiuc.edu/Research/namd/
Info: for updates, documentation, and support information.
Info:
Info: Please cite Phillips et al., J. Comp. Chem. 26:1781-1802 (2005)
Info: in all publications reporting results obtained with NAMD.
Info:
Info: Based on Charm++/Converse 60700 for multicore-darwin-x86_64
Info: Built Mon Dec 21 10:30:06 CST 2015 by jim on roswell.ks.uiuc.edu
Info: 1 NAMD  2.11  MacOSX-x86_64-multicore  1    LICHEN-YUs-MacBook-Pro.local  lichen-yu
Info: Running on 1 processors, 1 nodes, 1 physical nodes.
Info: CPU topology information available.
Info: Charm++/Converse parallel runtime startup completed at 0.0214241 s
CkLoopLib is used in SMP with a simple dynamic scheduling (converse-level notification) but not using node-level queue
FATAL ERROR: No simulation config file specified on command line.
------------- Processor 0 Exiting: Called CmiAbort ------------
Reason: FATAL ERROR: No simulation config file specified on command line.

[0] Stack Traceback:
  [0:0] 0   namd2                               0x00000001007578e0 CmiSyncBroadcastAllFn + 0
  [0:1] 1   namd2                               0x0000000100061631 _Z8NAMD_diePKc + 113
Charm++ fatal error:
FATAL ERROR: No simulation config file specified on command line.

[0] Stack Traceback:
  [0:0] 0   namd2                               0x000000010075ad6e LrtsAbort + 158
  [0:1] 1   namd2                               0x000000010075aa55 CmiAbortHelper + 165
  [0:2] 2   namd2                               0x00000001007578e0 CmiSyncBroadcastAllFn + 0
  [0:3] 3   namd2                               0x0000000100061631 _Z8NAMD_diePKc + 113
Abort trap: 6
[0]~/Desktop
```

Figure 7: The error message showing that no config file is provided.

This confirms that you have successfully launched NAMD from the terminal.

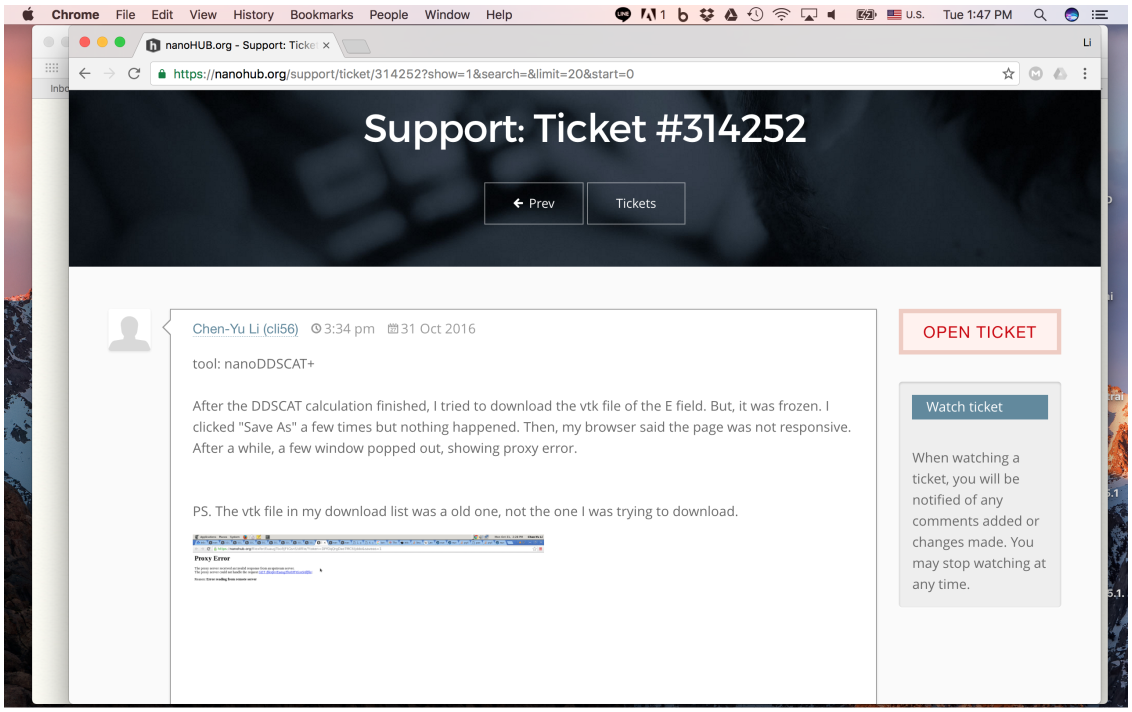
1.5 Found a bug in nanoDDSCAT+



```
>DDSCAT using parameter file=
        ddscat.par
>DDSCAT NUMPROC=   1
>DDSCAT --- DDSCAT 7.3.0 [13.05.03]
>DDSCAT     VERSNUM=   730
>DDSCAT     Double-precision version
>REAPAR  ========= Parameter file for v7.3 ===================
>  **** Preliminaries ****
>REAPAR NOTORQ - do not compute torques
>REAPAR PBCGS2 - CCG Method
>REAPAR GPFAFT - using GPFA package from Clive Temperton
>REAPAR GKDLDR - Gutkowicz-Krusin & Draine (2004) LDR for alpha
>REAPAR NOTBIN - Unformatted binary dump option
>REAPAR **** Initial Memory Allocation ****
>REAPAR allow MXNX,MXNY,MXNZ=    60   150   150 for target generation
>REAPAR **** Target Geometry and Composition ****
>REAPAR FROM_FILE - Shape definition
>REAPAR  NCOMP=   3
>REAPAR comp  1 : AuDiel.tab
>REAPAR comp  2 : AuDiel.tab
>REAPAR comp  3 : custom_dielectric3
>REAPAR **** Additional Nearfield calculation? ****
>REAPAR  1 = NRFLD : calculate nearfield E in specified volume
>REAPAR  0.500 0.500 0.500 0.500 0.500 0.500
>REAPAR  fractional extension in -x,+x,-y,+y,-z,+z directions
>REAPAR **** Error Tolerance ****
>REAPAR  1.000E-05 = TOL = max. acceptable normalized residual |Ax-E|/|E|
>REAPAR **** maximum number of iterations allowed ****
>REAPAR   10000000 = MXITER
>  **** Interaction cutoff parameter for PBC calculations ****
>REAPAR [GAMMA is not used in present (non-PBC) calculation]
>  **** Angular resolution for calculation of <cos>, etc. ****
>REAPAR  0.5000 = ETASCA (parameter controlling number of scattering angles
>  **** Vacuum wavelengths (micron) ****
```

Figure 8: Go to "SUPPORT" > "Tickets" to submit a ticket.

If nanoDDSCAT+ crashes for unknown reasons, please submit a ticket directly through nanoHUB. You can submit a ticket by going to "SUPPORT" > "Tickets" on the nanoHUB web page (Figure 8).


Chrome File Edit View History Bookmarks People Window Help
Tue 1:47 PM
nanoHUB.org - Support: Ticke
https://nanohub.org/support/ticket/314252?show=1&search=&limit=20&start=0
Prev
Tickets
OPEN TICKET
Watch ticket

Support: Ticket #314252

Chen-Yu Li (cli56) 3:34 pm 31 Oct 2016

tool: nanoDDSCAT+

After the DDSCAT calculation finished, I tried to download the vtk file of the E field. But, it was frozen. I clicked "Save As" a few times but nothing happened. Then, my browser said the page was not responsive. After a while, a few window popped out, showing proxy error.

PS. The vtk file in my download list was a old one, not the one I was trying to download.

Proxy Error

When watching a ticket, you will be notified of any comments added or changes made. You may stop watching at any time.

Figure 9: An example ticket. It is best to provide the name of the tool, steps needed to reproduce the bug (if applicable) and a snapshot of the error or error message.

Figure 9 is an example of a ticket. To make the communication (ticket) more effective, please specify the name of the tool, the detailed steps needed to reproduce the bug (if applicable) and a snapshot image of the bug.

We apologize for any inconvenience caused by bugs in the beta versions of the tools, and we appreciate your report to help us improve them!

2 Obtain the plasmonic field using nanoDDSCAT+

In this section, we will use the nanoDDSCAT+ on nanoHUB to calculate the 3D plasmonic field distribution. Then, we will download the plasmonic field distribution and the 3D grid file representing the shape of our system from nanoHUB to our desktop.

We will use them for the MD simulation in section 3.

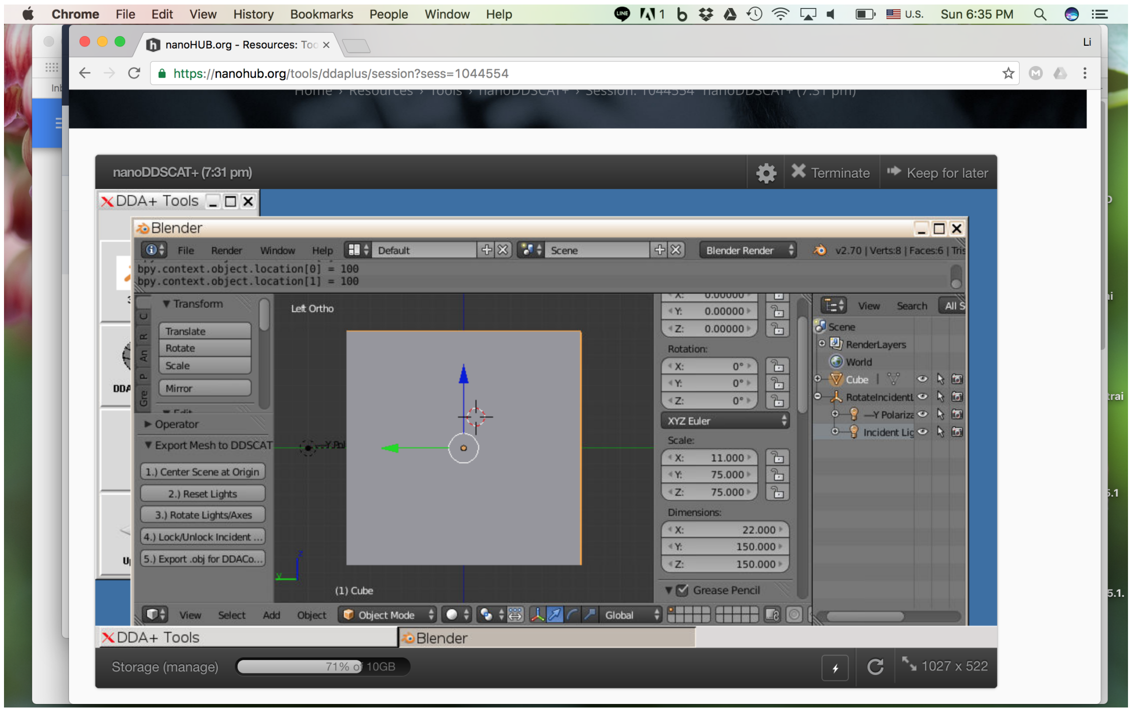
First, let's open a terminal and navigate to the "ch2" folder.

2.1 Create the shape of the nanopore system in Blender

Go to the nanoDDSCAT+ tool page on nanoHUB. After launching the nanoDDSCAT+, you should see a workspace with a toolbar on the left. On the toolbar, from top to bottom are “Blender”, “DDA convert”, “DDSCAT” and “Upload/Download”. The pipeline of nanoDDSCAT+ is “Blender” ⇒ “DDA convert” ⇒ “DDSCAT”. So, we need to create the shape of the nanopore system in Blender.

We will use the same nanopore system as in the paper by Belkin *et al* [1]. The system contains two 34 nm thick, 60 nm on a side equilateral gold triangles (bow tie structure), separated by a 10 nm gap on a 22 nm thick Si_3N_4 membrane. A 10 nm in diameter nanopore parallel to the Z-axis was made at the gap center of the bow tie antenna. The incident light will be coming from top (+Z direction) with a polarization along the Y-axis. However, the incident light in DDSCAT is coming from the X-axis by default. So, to make the process easier, we will rotate our system 90 degrees along the Y-axis in Blender.

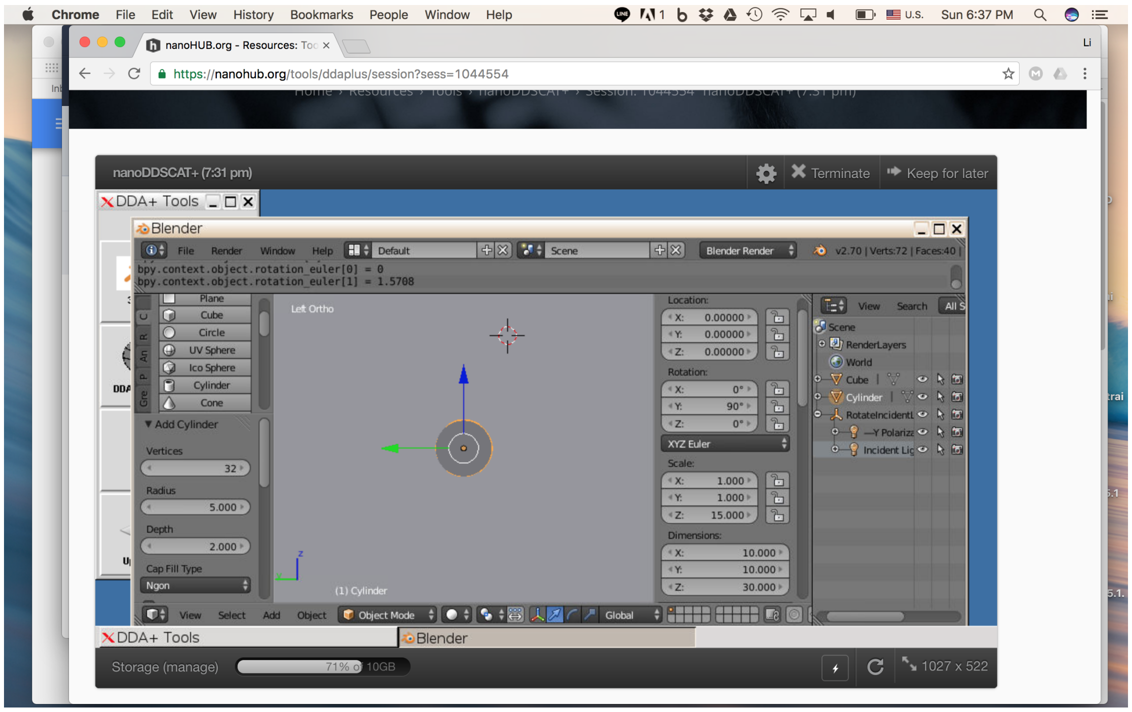
Below, we provide one of the many methods to make the shape of our system in Blender. **If you are familiar with Blender, you are free to choose any method you like, as long as the final outcome is the same.**



```
bpy.context.object.location[0] = 100
bpy.context.object.location[1] = 100
```
Rotation: X: 0° Y: 0° Z: 0°
XYZ Euler
Scale: X: 11.000 Y: 75.000 Z: 75.000
Dimensions: X: 22.000 Y: 150.000 Z: 150.000

Figure 10: Change the initial cube into a plate with X = 22, Y = Z = 150. Put it at the origin (0,0,0).

First, change the initial cube into a plate with X = 22, Y = Z = 150. Keep its position at the origin (0,0,0), Figure 10. This is our 22 nm thick Si_3N_4 membrane.



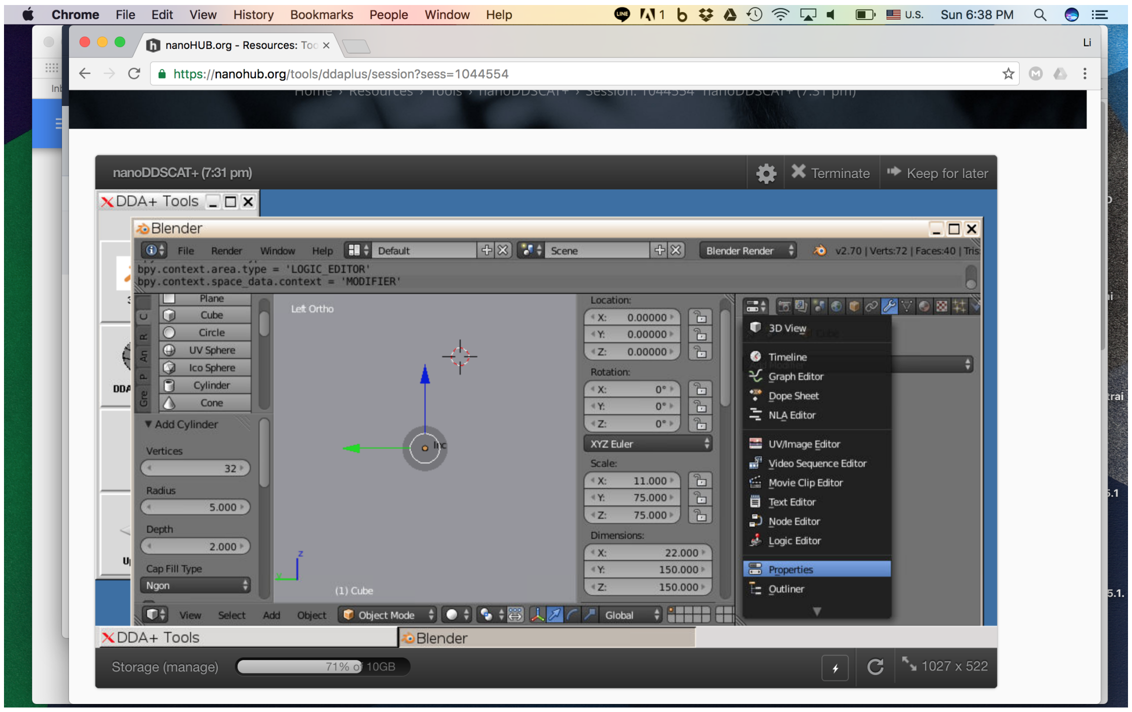
```
bpy.context.object.rotation_euler[0] = 0
bpy.context.object.rotation_euler[1] = 1.5708
```

Add Cylinder
Vertices: 32
Radius: 5.000
Depth: 2.000
Cap Fill Type: Ngon

Location: X: 0.00000, Y: 0.00000, Z: 0.00000
Rotation: X: 0°, Y: 90°, Z: 0°
XYZ Euler
Scale: X: 1.000, Y: 1.000, Z: 15.000
Dimensions: X: 10.000, Y: 10.000, Z: 30.000

Figure 11: Create a cylinder with R = 5 and Z = 30. Put it at the origin (0,0,0). Rotate along Y-axis by 90 degrees.

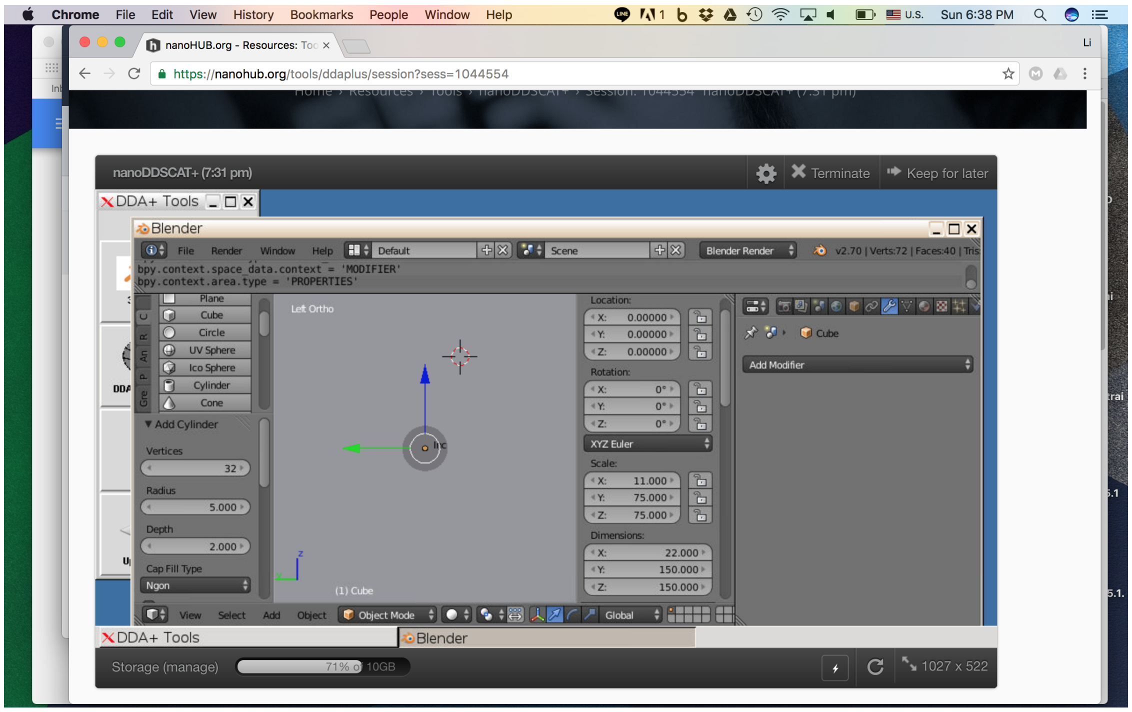
Next, we need to make a hole with a diameter of 10 as the nanopore on the Si_3N_4 membrane. We will achieve this by subtracting a cylinder with a diameter of 10 from the Si_3N_4 membrane. First, create a cylinder with R = 5 and Z = 30. Put it at the origin (0,0,0). Rotate along Y-axis by 90 degrees, Figure 11.



```
bpy.context.area.type = 'LOGIC_EDITOR'
bpy.context.space_data.context = 'MODIFIER'
```

Figure 12: Select the Si_3N_4 membrane. Choose the “Properties” from the editor type menu.

Select the Si_3N_4 membrane. Choose the “Properties” from the editor type menu, Figure 12.

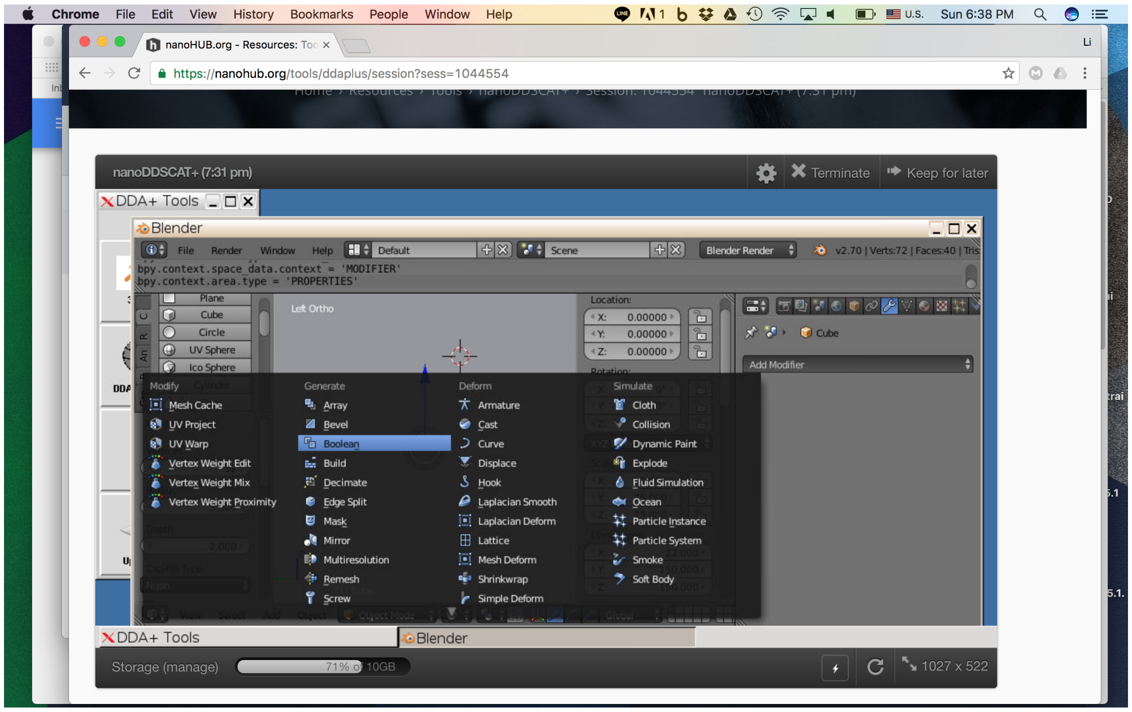


```
bpy.context.space_data.context = 'MODIFIER'
bpy.context.area.type = 'PROPERTIES'
```

https://nanohub.org/tools/ddaplus/session?sess=1044554
nanoDDSCAT+ (7:31 pm)
Terminate
Keep for later
DDA+ Tools
Blender
Add Cylinder
Vertices 32
Radius 5.000
Depth 2.000
Cap Fill Type Ngon
Location: X: 0.00000 Y: 0.00000 Z: 0.00000
Rotation: X: 0° Y: 0° Z: 0°
XYZ Euler
Scale: X: 11.000 Y: 75.000 Z: 75.000
Dimensions: X: 22.000 Y: 150.000 Z: 150.000
Cube
Add Modifier
Storage (manage)
1027 x 522

Figure 13: Choose “Modifiers”.

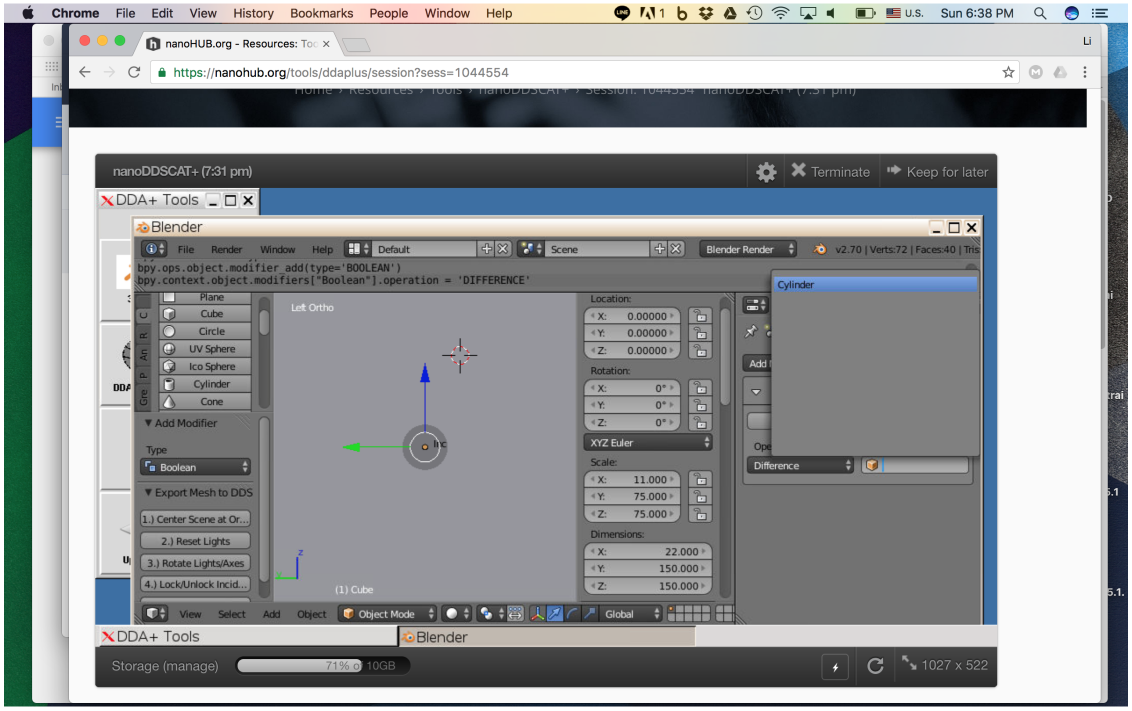
Choose “Modifiers”, Figure 13.



```
bpy.context.space_data.context = 'MODIFIER'
bpy.context.area.type = 'PROPERTIES'
```

Figure 14: “Add Modifier” → “Boolean”.

Click "Add Modifier". Then, select "Boolean", Figure 14.

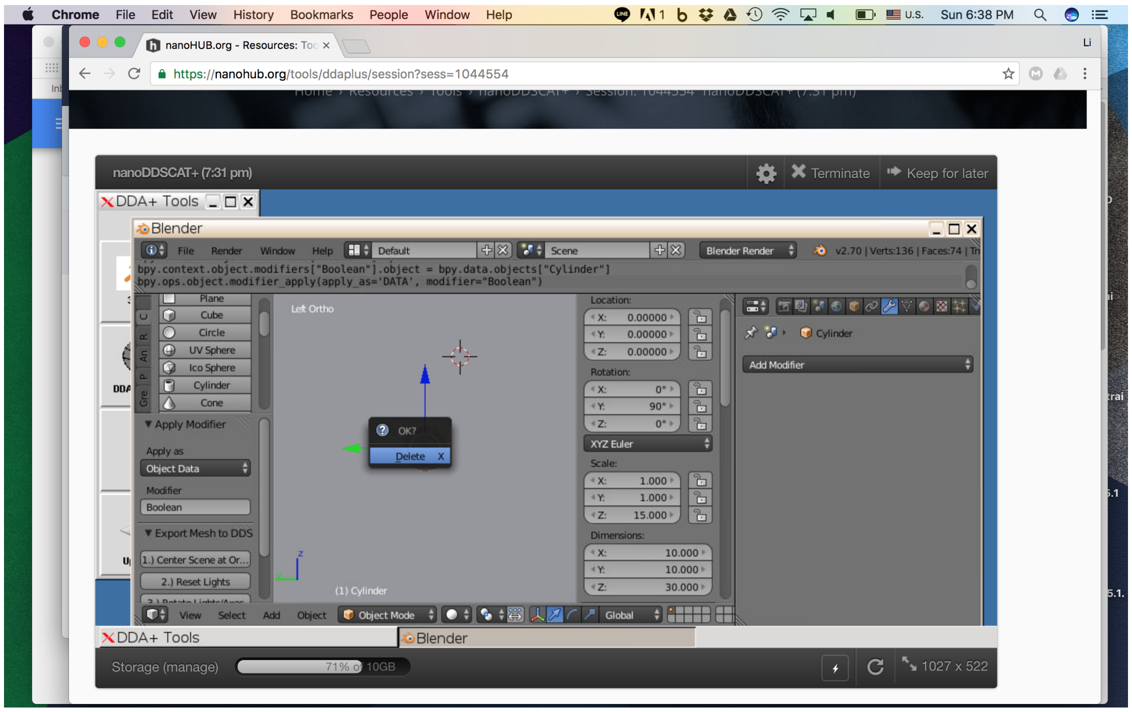


```
bpy.ops.object.modifier_add(type='BOOLEAN')
bpy.context.object.modifiers["Boolean"].operation = 'DIFFERENCE'
```

Type: Boolean
Difference
Cylinder

Figure 15: Choose "Difference" with "Cylinder".

Choose "Difference" and select the "Cylinder" object we just added, Figure 15.



https://nanohub.org/tools/ddaplus/session?sess=1044554

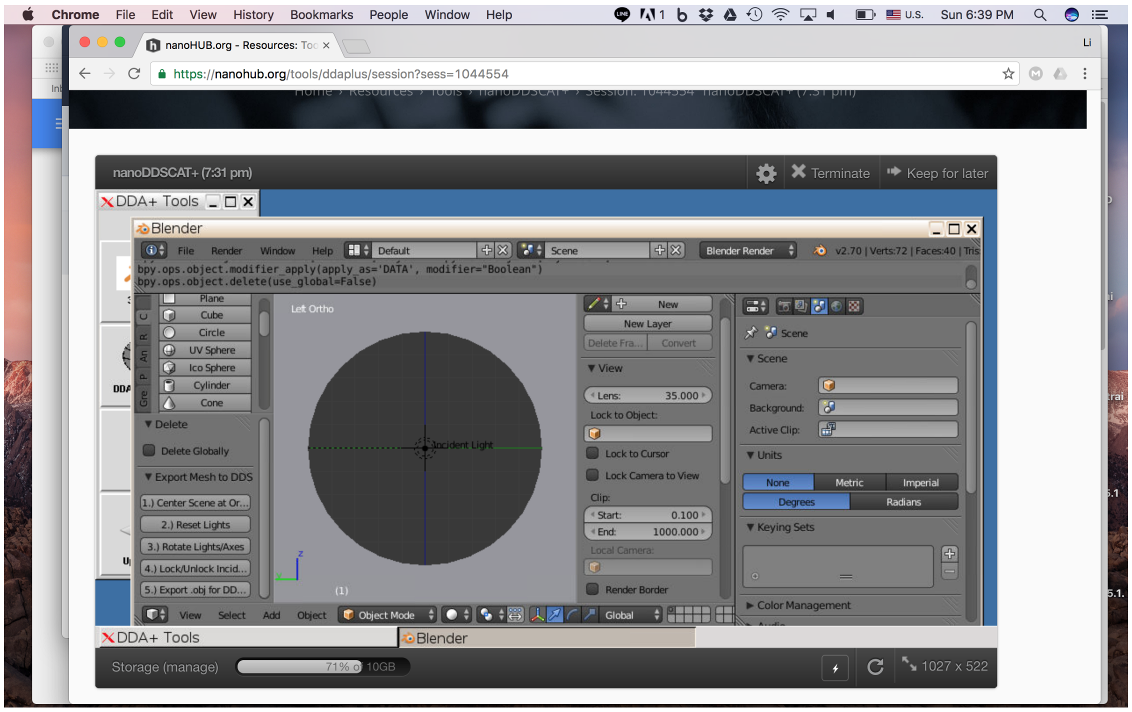
```
bpy.context.object.modifiers["Boolean"].object = bpy.data.objects["Cylinder"]
bpy.ops.object.modifier_apply(apply_as='DATA', modifier="Boolean")
```

Apply as: Object Data
Modifier: Boolean

Location: X: 0.00000, Y: 0.00000, Z: 0.00000
Rotation: X: 0°, Y: 90°, Z: 0°
XYZ Euler
Scale: X: 1.000, Y: 1.000, Z: 15.000
Dimensions: X: 10.000, Y: 10.000, Z: 30.000

Figure 16: Select “Cylinder”. Click hot key “x” to remove the “Cylinder”.

Now, let’s remove the cylinder to see whether we really made a hole in the membrane. Select “Cylinder” object. Click hot key “x” to remove the “Cylinder”, Figure 16.

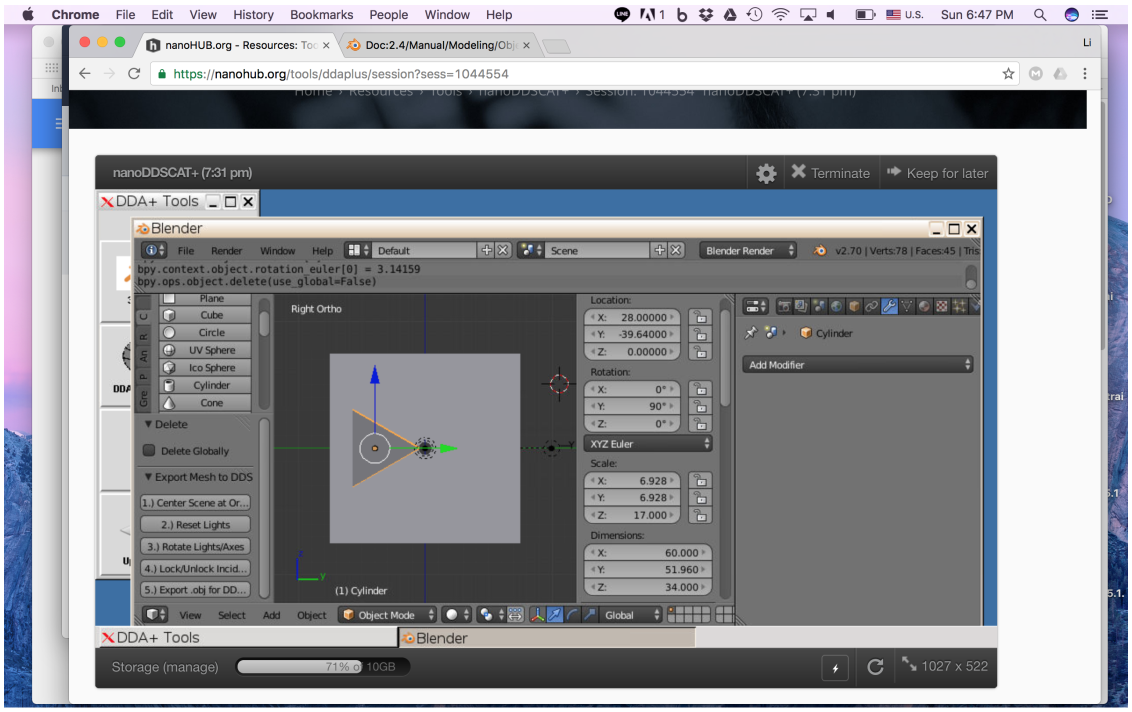


```
bpy.ops.object.modifier_apply(apply_as='DATA', modifier="Boolean")
bpy.ops.object.delete(use_global=False)
```

Lens: 35.000
Clip: Start: 0.100 End: 1000.000

Figure 17: A hole with R = 5.

Now, you should see a hole with R = 5 on the membrane, Figure 17.

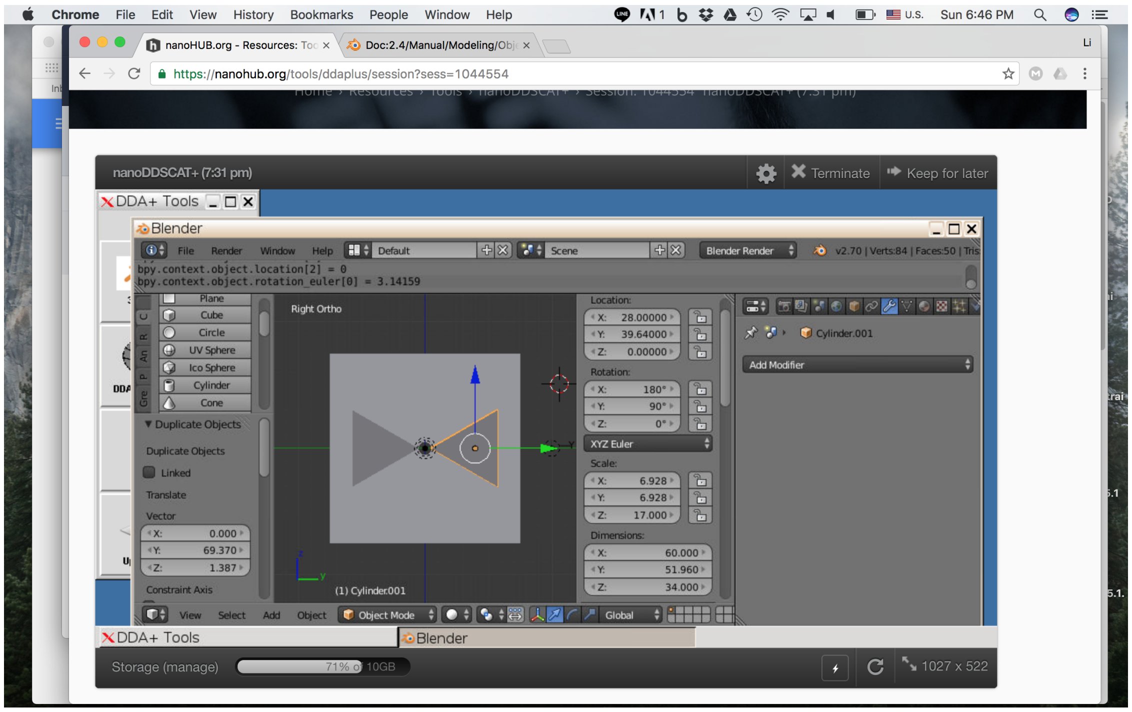


```
bpy.context.object.rotation_euler[0] = 3.14159
bpy.ops.object.delete(use_global=False)
```

Location:
X: 28.00000
Y: -39.64000
Z: 0.00000
Rotation:
X: 0°
Y: 90°
Z: 0°
XYZ Euler
Scale:
X: 6.928
Y: 6.928
Z: 17.000
Dimensions:
X: 60.000
Y: 51.960
Z: 34.000

Figure 18: Create a cylinder with vertices = 3. Change the size to X = 60, Y = 51.96 and Z = 34. Put it at (28, -39.64, 0). Rotate along Y-axis by 90 degrees.

Next, we will add the two bow tie structures. Create a cylinder with vertices = 3. Change the size to X = 60, Y = 60 $\times\sqrt{3}$ / 2 = 51.96 and Z = 34. Change the location to X = 34 (thickness of the bow tie structure) / 2 + 22 (thickness of the membrane) / 2 = 28, Y = -1 * ((60 $\times\sqrt{3}$ / 2) $\times$ (2 / 3) + 5 (radius of the pore)) = -39.64 and Z = 0. Rotate along Y-axis by 90 degrees, Figure 18.



```
bpy.context.object.location[2] = 0
bpy.context.object.rotation_euler[0] = 3.14159
```

Location: X: 28.00000 Y: 39.64000 Z: 0.00000
Rotation: X: 180° Y: 90° Z: 0°
XYZ Euler
Scale: X: 6.928 Y: 6.928 Z: 17.000
Dimensions: X: 60.000 Y: 51.960 Z: 34.000

Duplicate Objects
Linked
Translate
Vector: X: 0.000 Y: 69.370 Z: 1.387

(1) Cylinder.001

Chrome File Edit View History Bookmarks People Window Help
https://nanohub.org/tools/ddaplus/session?sess=1044554
nanoDDSCAT+ (7:31 pm)
Terminate
Keep for later
DDA+ Tools
Blender
File Render Window Help Default Scene Blender Render
Plane
Cube
Circle
UV Sphere
Ico Sphere
Cylinder
Cone
Right Ortho
Add Modifier
View Select Add Object Object Mode Global
Storage (manage) 71% of 10GB
1027 x 522

Figure 19: "Shift" + "D" to duplicate the bow tie structure. Put it at (28, 39.64, 0). Rotate 180 degrees along X-axis.

Then, duplicate the bow tie structure by "Shift" + "D". Put it at (28, 39.64, 0). Rotate 180 degrees along X-axis, Figure 19.

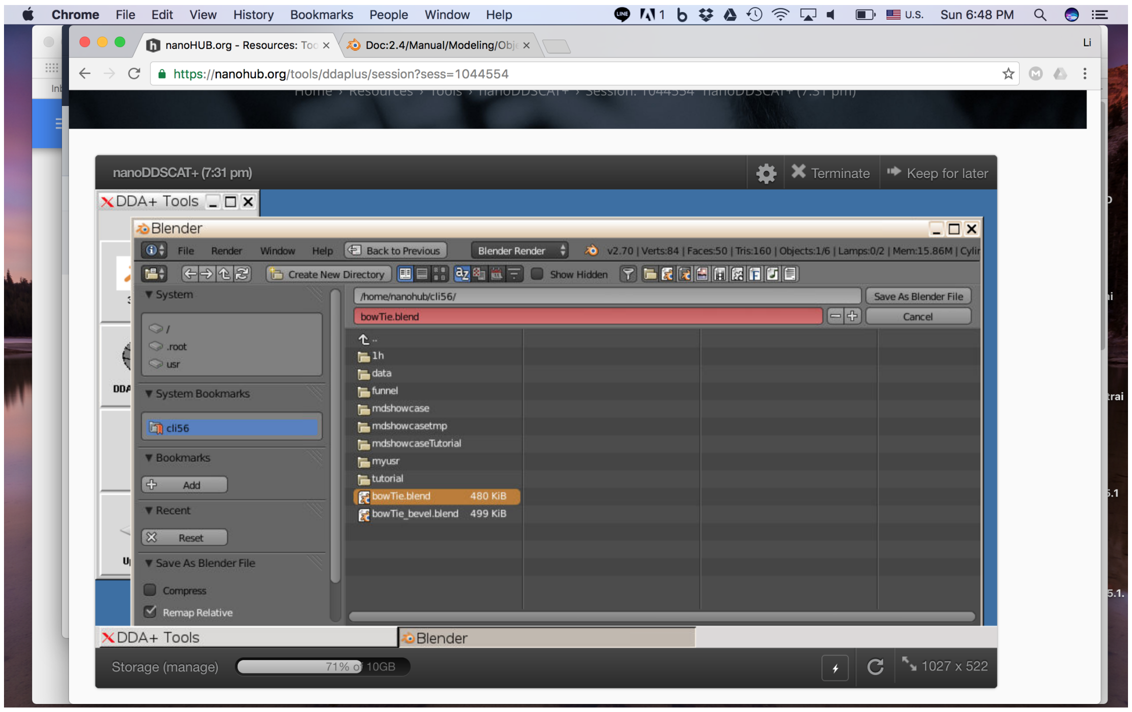


Figure 20: Save the design as “bowTie.blend” in the home directory in the nanoHUB workspace.

Let’s save the file in case anything happens. Save the design as “bowTie.blend” in the home directory in the nanoHUB workspace, Figure 20.

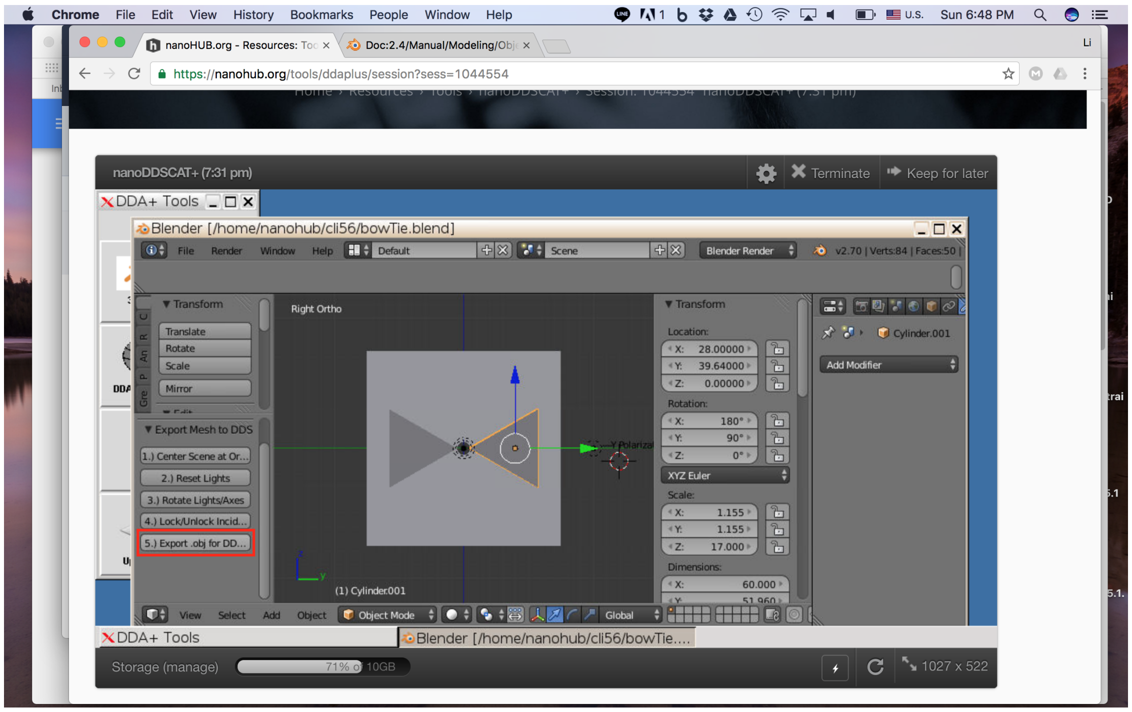


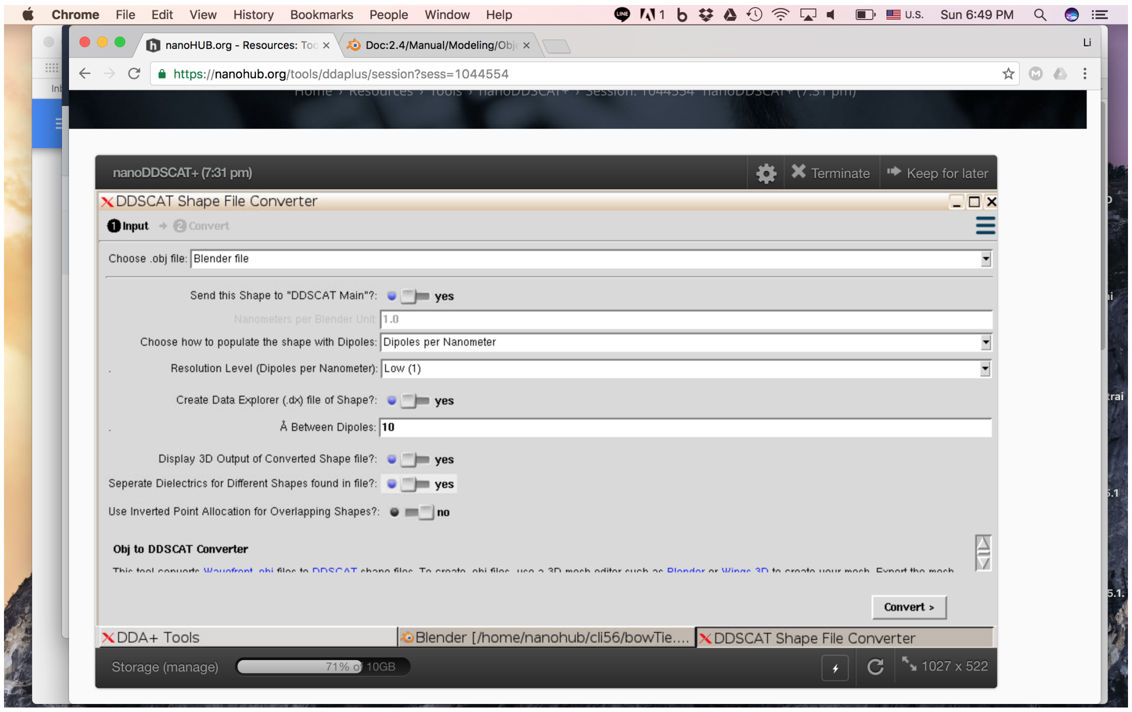
Figure 21: Click “Export .obj for DDAConvert”.

Finally, we need to export the design as an object (.obj) file for DDA convert, which is the next component in the pipeline, Figure 21.

If you encountered any issue creating the shape in Blender, please use the example file “bowTie.blend” in “example_output”. You will need to upload the file to your home directory in the nanoHUB workspace. Open the file in Blender. Then, export the .obj as in Figure 21.

2.2 Discretize the shape into dipole points

Next, we will use DDA Convert to convert the object file to discrete dipoles for DDSCAT calculation. Click the “DDA Convert” in the toolbar and you should see the window like Figure 22.


https://nanohub.org/tools/ddaplus/session?sess=1044554
nanoDDSCAT+ (7:31 pm)
Terminate
Keep for later
DDSCAT Shape File Converter
Input
Convert
Convert >
DDA+ Tools
Blender [/home/nanohub/cli56/bowTie....
Storage (manage)
71% of 10GB
1027 x 522

Choose .obj file: Blender file

Send this Shape to "DDSCAT Main"?: yes

Nanometers per Blender Unit: 1.0

Choose how to populate the shape with Dipoles: Dipoles per Nanometer

Resolution Level (Dipoles per Nanometer): Low (1)

Create Data Explorer (.dx) file of Shape?: yes

Å Between Dipoles: 10

Display 3D Output of Converted Shape file?: yes

Seperate Dielectrics for Different Shapes found in file?: yes

Use Inverted Point Allocation for Overlapping Shapes?: no

Obj to DDSCAT Converter

Figure 22: Choose the "Blender file" as the input .obj file. Set resolution to low(1). Check "yes" on "create data explorer file". Set "10" Å between dipoles. Check "yes" on display 3D output. Check "yes" on separate dielectric between different shape.

Since we have exported the .obj file in Blender, we will select the "Blender file" as the input .obj file. The size of our shape is relatively large. By having 1 dipole per nanometer, we will have ∼1,350,000 dipoles. That will take a few hours to calculate on nanoHUB. Decreasing the number of dipoles per nanometer will reduce the calculation time. But, it will also reduce the resolution of the resulting plasmonic field, which may affect the trapping efficiency. You are free to tune the resolution to find the optimum number for your purpose, but we will continue with 1 dipole per nanometer for this tutorial.

Next, check "yes" on create data explorer file. We will use it to represent the bow tie structure and the membrane, which the polypeptide should not overlap with. The details will be explained in section 3.1. Set the distance between dipoles as "10" Å, the same resolution as before. Check "yes" on display 3D output, so that we can visualize the output from discretization. Check "yes" on separate dielectric between different shape because we have different materials (gold and Si_3N_4).

Finally, click "Convert" on the bottom right. The conversion will start.

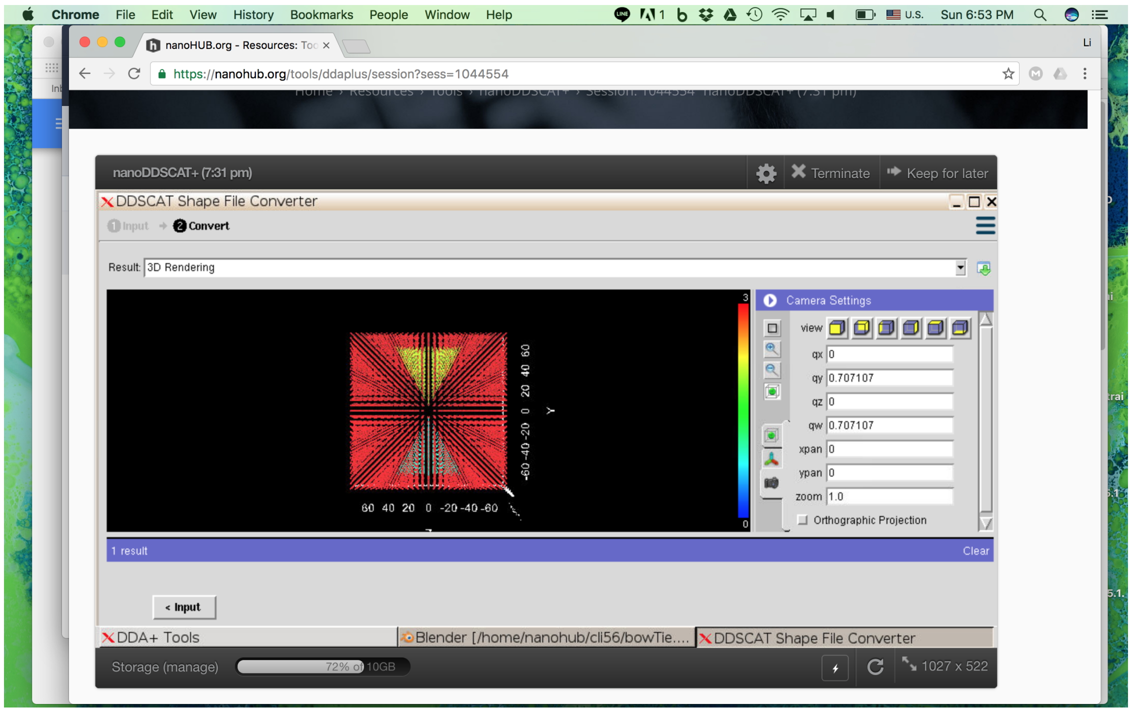
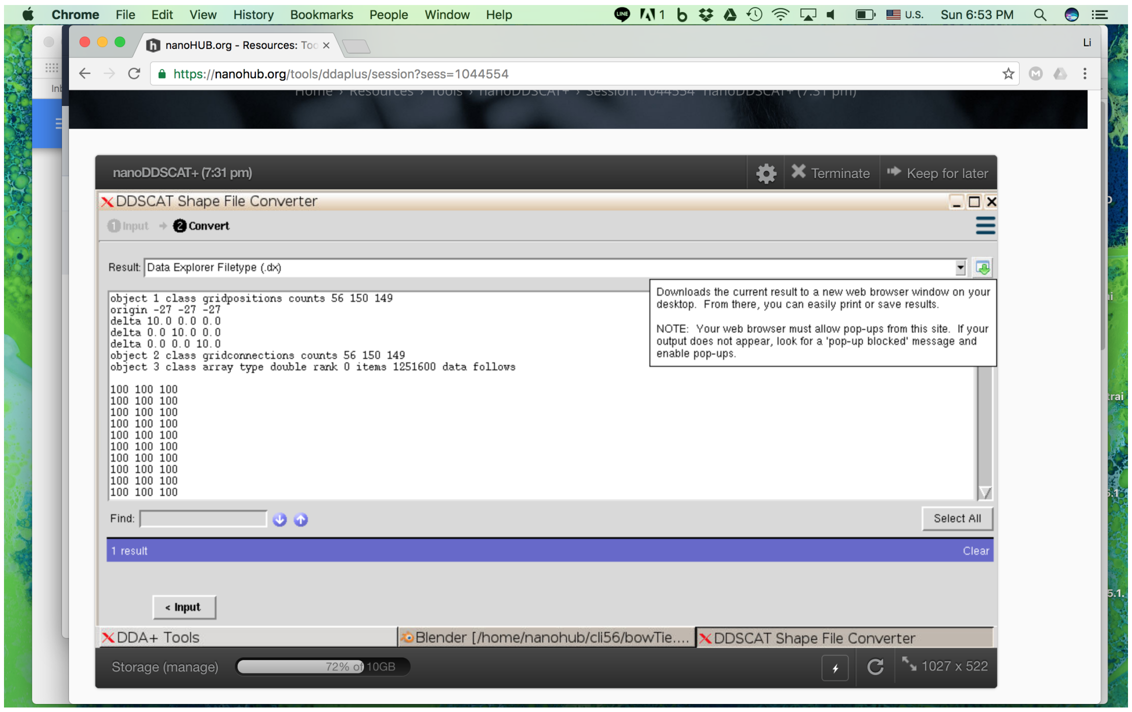


Figure 23: The 3D rendering result of the discrete dipoles. The 3 different objects (2 bow tie structures and 1 membrane) are colored by their indices.

Once the conversion finished, you should see the output log. You can choose to see different output using the result drop-down menu. Choose the "3D Rendering" result and you should see the 3D rendering result of the discrete dipoles, Figure 23. The 3 different objects (2 bow tie structures and 1 membrane) are colored by their indices. We will use those indices to assign different dielectric constants in DDSCAT.



```
object 1 class gridpositions counts 56 150 149
origin -27 -27 -27
delta 10.0 0.0 0.0
delta 0.0 10.0 0.0
delta 0.0 0.0 10.0
object 2 class gridconnections counts 56 150 149
object 3 class array type double rank 0 items 1251600 data follows

100 100 100
100 100 100
100 100 100
100 100 100
100 100 100
100 100 100
100 100 100
100 100 100
100 100 100
100 100 100
```

Downloads the current result to a new web browser window on your desktop. From there, you can easily print or save results.

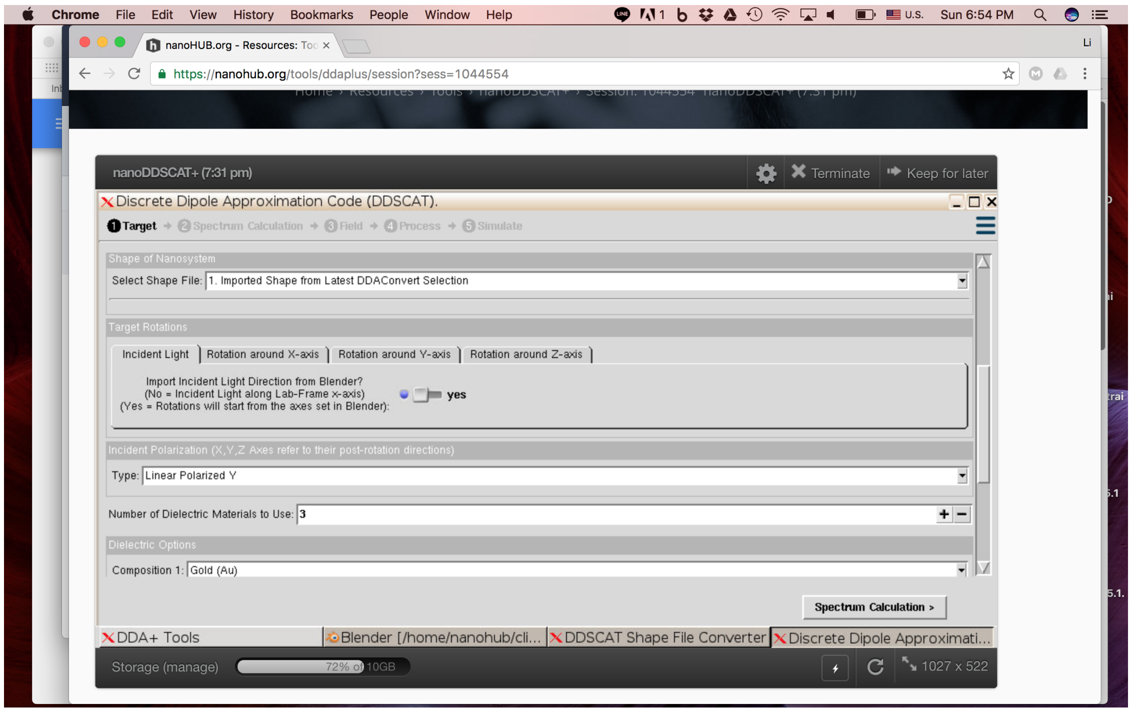
NOTE: Your web browser must allow pop-ups from this site. If your output does not appear, look for a 'pop-up blocked' message and enable pop-ups.

Figure 24: Choose the "Data Explorer Filetype (.dx)" result. Click the "Download" button on the right and save it as "DipolesPerNM1_wall.dx".

Finally, go to the "Data Explorer Filetype (.dx)" result, Figure 24. Click the "Download" button on the right and save it as "DipolesPerNM1_wall.dx". Again, we will use it to represent the bow tie structure and the membrane, which the polypeptide should not overlap with. The details will be explained in section 3.1.

2.3 Calculate the plasmonic field using DDSCAT

The last step in Section 2 is the calculation of the plasmonic field using DDSCAT. Click the "DDSCAT" in the toolbar and you should see the window like Figure 25.


Chrome File Edit View History Bookmarks People Window Help
nanoHUB.org - Resources: Too
https://nanohub.org/tools/ddaplus/session?sess=1044554
nanoDDSCAT+ (7:31 pm)
Terminate
Keep for later
Discrete Dipole Approximation Code (DDSCAT).
1 Target
2 Spectrum Calculation
3 Field
4 Process
5 Simulate
Incident Light
Rotation around X-axis
Rotation around Y-axis
Rotation around Z-axis
Spectrum Calculation >
DDA+ Tools
Blender [/home/nanohub/cli...
DDSCAT Shape File Converter
Discrete Dipole Approximati...
Storage (manage)
72% of 10GB
1027 x 522

Shape of Nanosystem
Select Shape File: 1. Imported Shape from Latest DDAConvert Selection

Target Rotations
Import Incident Light Direction from Blender?
(No = Incident Light along Lab-Frame x-axis)
(Yes = Rotations will start from the axes set in Blender): yes

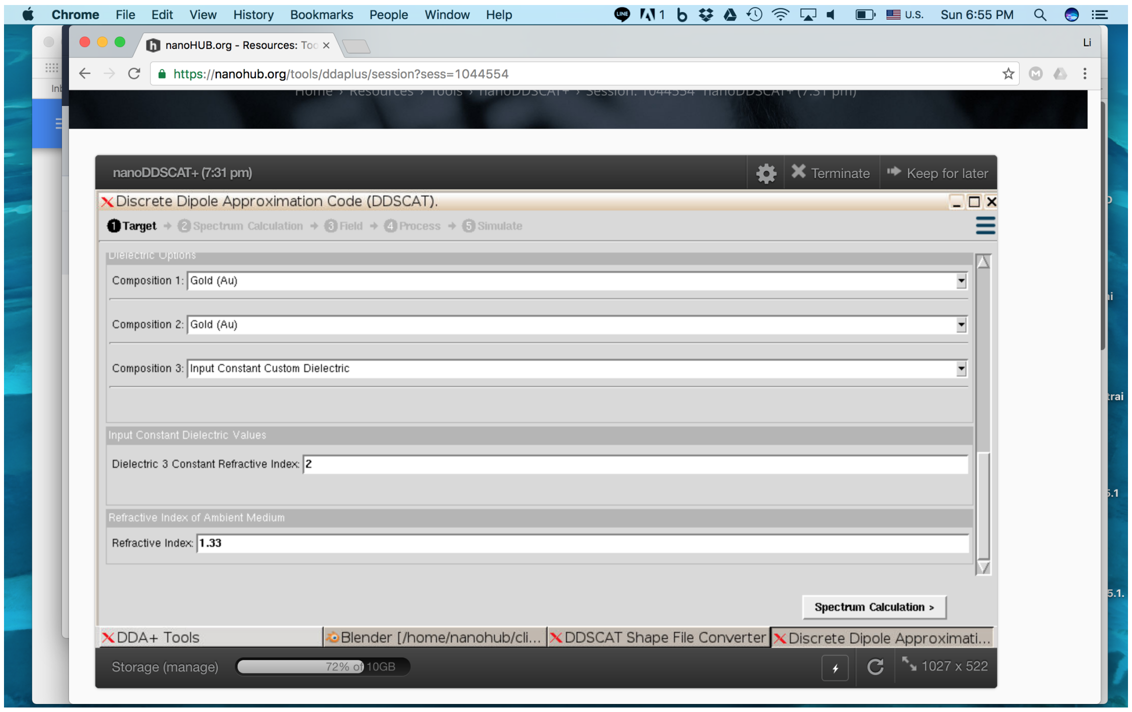
Incident Polarization (X,Y,Z Axes refer to their post-rotation directions)
Type: Linear Polarized Y

Number of Dielectric Materials to Use: 3

Dielectric Options
Composition 1: Gold (Au)

Figure 25: Select "Imported Shape from Latest DDA Convert" as the shape file. Increase number of dielectric material to "3".

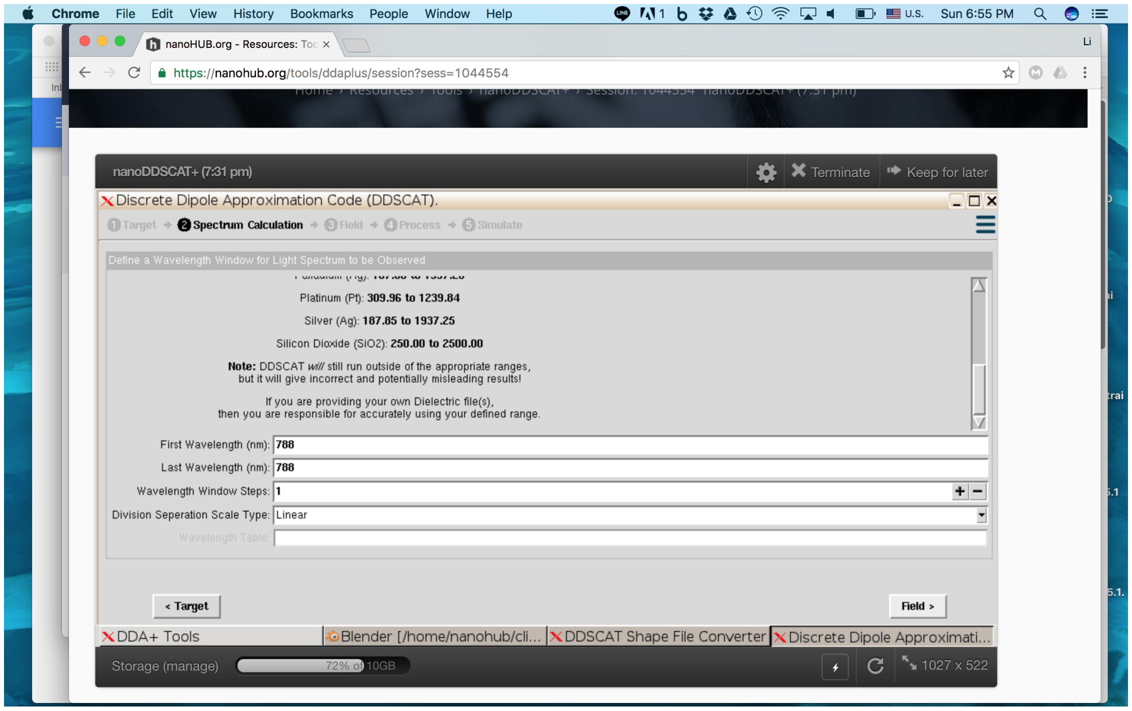
First, we will select "Imported Shape from Latest DDA Convert" as the shape file. Since we have 3 different objects in the system, we need to assign dielectric constant individually. Increase number of dielectric material to "3".


nanoDDSCAT+ (7:31 pm)
Terminate
Keep for later
Discrete Dipole Approximation Code (DDSCAT).
1 Target
2 Spectrum Calculation
3 Field
4 Process
5 Simulate
Dielectric Options
Input Constant Dielectric Values
Refractive Index of Ambient Medium
Spectrum Calculation >
DDA+ Tools
Blender [/home/nanohub/cli...
DDSCAT Shape File Converter
Discrete Dipole Approximati...
Storage (manage)
72% of 10GB
1027 x 522

Composition 1: Gold (Au)
Composition 2: Gold (Au)
Composition 3: Input Constant Custom Dielectric
Dielectric 3 Constant Refractive Index: 2
Refractive Index: 1.33

Figure 26: Set the first two compositions as "Gold (Au)". Set the third composition "Input Custom dielectric" and set the custom dielectric to "2". Set the refractive index of the medium to "1.33".

Then, we can assign the dielectric constant for each object, Figure 26. To identify which compositions are the two bow tie structures and the membrane, we need to recall the 3D rendering result in Figure 23. In Figure 23, the two bow tie structures are colored in cyan and yellow, while the membrane is colored in red. From the color bar on the right, we know that the red represent the highest value (3). So, the index of the membrane is 3 and the indices of the two gold bow tie structure are 1 and 2. Let's set the first two compositions as "Gold (Au)". Set the third composition "Input Custom dielectric" and set the custom dielectric to "2". Set the refractive index of the medium to "1.33". Then, click "Spectrum Calculation".


https://nanohub.org/tools/ddaplus/session?sess=1044554
nanoDDSCAT+ (7:31 pm)
Terminate
Keep for later
Discrete Dipole Approximation Code (DDSCAT).
1 Target
2 Spectrum Calculation
3 Field
4 Process
5 Simulate
< Target
Field >
DDA+ Tools
Blender [/home/nanohub/cli...
DDSCAT Shape File Converter
Discrete Dipole Approximati...
Storage (manage)
72% of 10GB
1027 x 522

Define a Wavelength Window for Light Spectrum to be Observed

Platinum (Pt): **309.96 to 1239.84**

Silver (Ag): **187.85 to 1937.25**

Silicon Dioxide (SiO2): **250.00 to 2500.00**

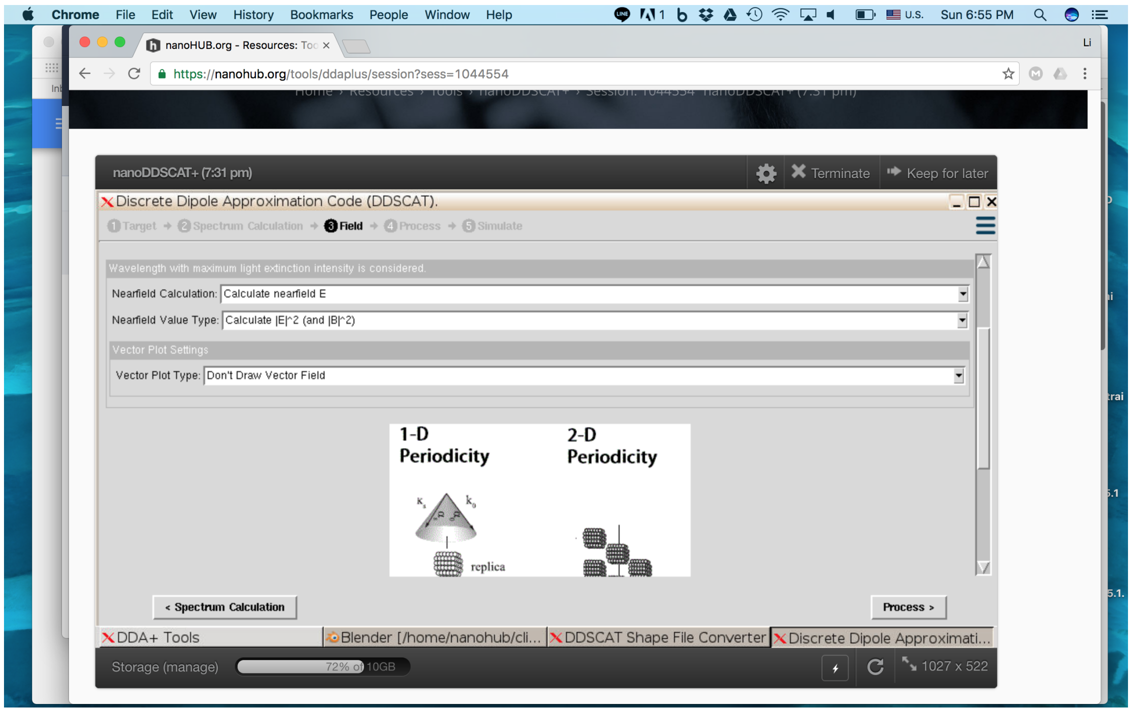
Note: DDSCAT *will* still run outside of the appropriate ranges, but it will give incorrect and potentially misleading results!

If you are providing your own Dielectric file(s), then you are responsible for accurately using your defined range.

First Wavelength (nm):	788
Last Wavelength (nm):	788
Wavelength Window Steps:	1
Division Seperation Scale Type:	Linear
Wavelength Table:	

Figure 27: Set the first and last wavelength to “788” nm.

The incident light with a wavelength of “788” nm was used in the paper by Belkin *et al* [1]. So, let’s Set the first and last wavelength to “788” nm, Figure 27. Then, click “Field”.



https://nanohub.org/tools/ddaplus/session?sess=1044554

Wavelength with maximum light extinction intensity is considered.

Nearfield Calculation: Calculate nearfield E

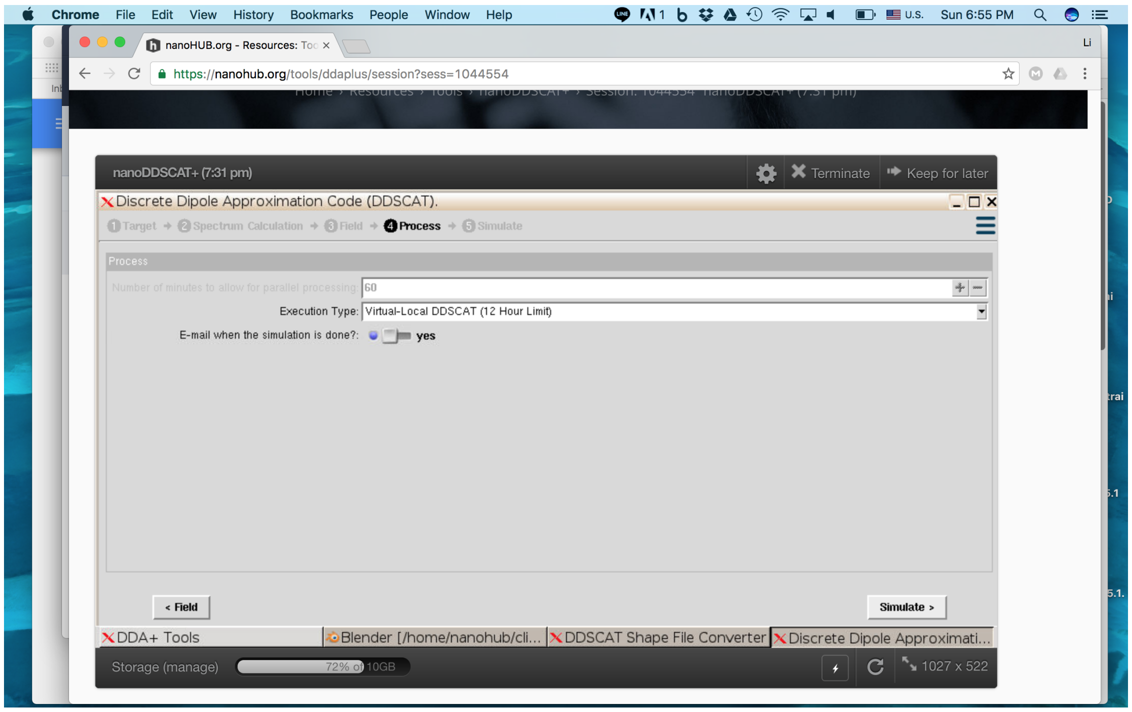
Nearfield Value Type: Calculate |E|^2 (and |B|^2)

Vector Plot Settings

Vector Plot Type: Don't Draw Vector Field

Figure 28: Set "Calculate nearfield E" for nearfield calculation. Set "Calculate |E|ˆ2 (and |B|ˆ2)" for the nearfield value type.

Set "Calculate nearfield E" for nearfield calculation, Figure 28. Set "Calculate |E|ˆ2 (and |B|ˆ2)" for the nearfield value type. Click "Process". Then, if you would like to receive an e-mail when the simulation is done, check "yes", Figure 29. Finally, click "Simulate".


nanoDDSCAT+ (7:31 pm)
Terminate
Keep for later
Discrete Dipole Approximation Code (DDSCAT).
1 Target → 2 Spectrum Calculation → 3 Field → 4 Process → 5 Simulate
Process
< Field
Simulate >
DDA+ Tools
Storage (manage)
1027 x 522

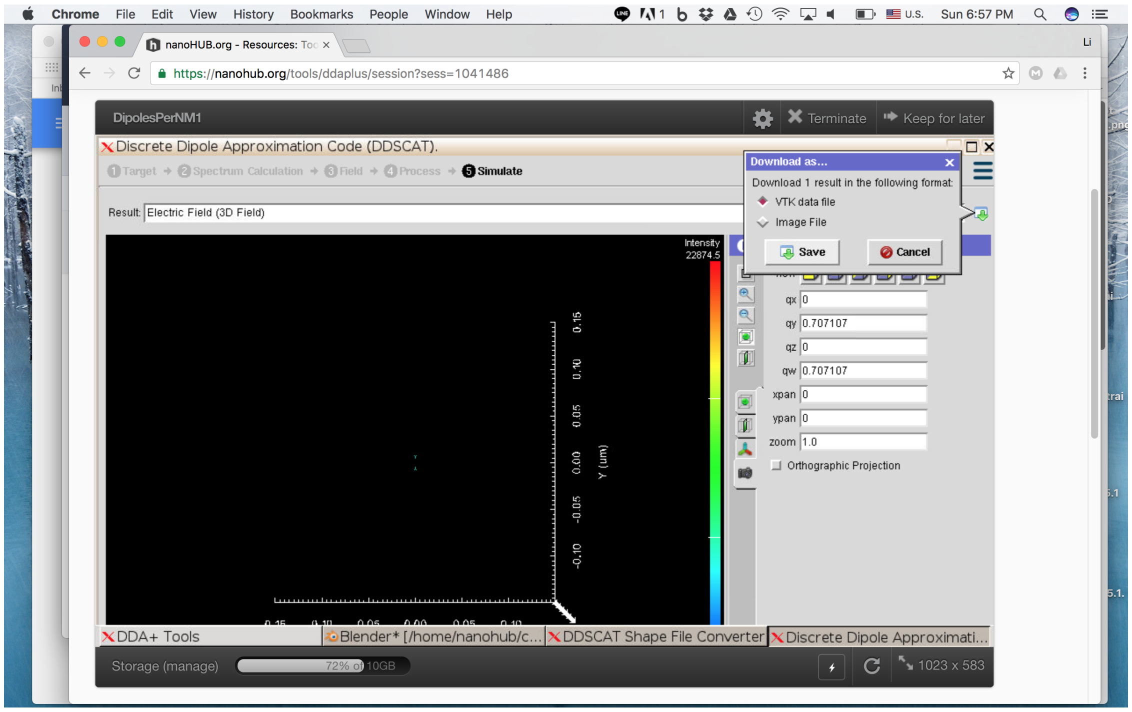
https://nanohub.org/tools/ddaplus/session?sess=1044554

Number of minutes to allow for parallel processing: 60

Execution Type: Virtual-Local DDSCAT (12 Hour Limit)

E-mail when the simulation is done?: yes

Figure 29: Check “yes” for e-mail when the simulation is done.



https://nanohub.org/tools/ddaplus/session?sess=1041486

Result: Electric Field (3D Field)

Download 1 result in the following format:
- VTK data file
- Image File

qx 0
qy 0.707107
qz 0
qw 0.707107
xpan 0
ypan 0
zoom 1.0
Orthographic Projection

Figure 30: The 3D rendering of the Electric field. Click “Download” button on the right and download it as VTK data file.

Once the calculation finished, you should see the output log. Choose the "Electric Field (3D field)" result and you should see the 3D rendering result of the electric field, Figure 30. Click "Download" button on the right and download it as VTK data file. Save the downloaded vtk file as "DipolesPerNM1.vtk".

If you encountered any issue calculating the plasmonic field, or you don't have time to wait, please use the example file "DipolesPerNM1.vtk" in "example_output" to continue the tutorial.

3 Simulate the trapping of a polypeptide by the plasmonic field

In this section, we will simulate the translocation of a polypeptide by in an electric field. Then, we will apply the plasmonic field obtained from nanoDDSCAT+ to trap the polypeptide. We will also demonstrate how to control the translocation by a laser pulse. Finally, we will visualize the simulation result and analyze the displacement of the polypeptide.

The simulation method we use here is called Molecular Dynamics (MD). In short, the trajectories of atoms and molecules are determined by numerically solving Newton's equations of motion for a system of interacting particles, where forces between the particles and their potential energies are calculated using interatomic potentials or molecular mechanics force fields. For more details, please visit wikipedia: Molecular dynamics. We will use one of the most popular MD software NAMD to perform the MD simulation. The simulation output (trajectories) of NAMD is DCD file. You are free to use any molecular visualization software to see the trajectories. But for this tutorial, we will use VMD.

To reduce the computational cost so that people can finish this tutorial using their laptop, we will not perform all-atom MD simulation. Instead, we will simulate the polypeptide with implicit solvent. In addition, the gold bow tie structures and the Si_3N_4 membrane will be modeled using a set of grid potential to prevent the overlap with the polypeptide. The electric field and the plasmonic field will be modeled as grid potential as well. The polypeptide ("FDFD.pdb") we used for this tutorial contains simple repetitive sequence: $(FD)_{24}$.

Please navigate to the "ch3" folder.

3.1 Prepare the grid potential representing the nanopore system

First, please copy the "DipolesPerNM1_wall.dx" downloaded in section 2.2 to the current "ch3" folder.

If you did not complete section 2.2, you can use the example file "DipolesPerNM1_wall.dx" in "ch2/example_output".

The "DipolesPerNM1_wall.dx" is the 3D grid file representing the bow tie structure and the membrane. It contains one data value for each grid point in a 3-dimensional space. It also contains the number of grid points in each (X, Y and Z) direction, the distance between two grid points in each (X, Y and Z) direct ion and one of the vertexes (minX, minY and minZ). In VMD and NAMD, the grid has to be an orthogonal unit cell, and the distance will be treated as Å. In "DipolesPerNM1_wall.dx", the grid points within the bow tie structures and the membrane will have a value of 100. The rest will have a value of -100. However, there is an additional line in "DipolesPerNM1_wall.dx" which has to be removed.

Please type the following command in the terminal:

```
sed "8d" DipolesPerNM1_wall.dx > DipolesPerNM1_wall_clean.dx
```

By default, this grid does not align well with the grid for the plasmonic field. So, we need to align it by using a python script:

```
python shiftDX.py DipolesPerNM1_wall_clean.dx \
DipolesPerNM1_wall_clean_shift.dx
```

The script will also change the data value to 1 (within gold bow tie structures and membrane) and 0 (the rest) to reduce the force. Finally, to make the grid file (.dx) readable for NAMD, we need to change the data type from "float" to "double":

```
sed "s/float/double/g" DipolesPerNM1_wall_clean_shift.dx \
> DipolesPerNM1_wall_clean_shift_double.dx
```

Now, you have the final grid file for the structure.

If you did not complete the previous steps, you could use the example file "DipolesPerNM1_wall_clean_shift_double.dx" in "example_output". You can simply copy it to the current "ch3" folder.

3.2 Convert the DDSCAT output to grid potential

Next, we need to convert the output from DDSCAT (.vtk) to the grid file (.dx). Please copy the "DipolesPerNM1.vtk" downloaded in section 2.3 to the current "ch3" folder.

If you did not complete section 2.3, you could use the example file "DipolesPerNM1.vtk" in "ch2/example_output".

Please type the following command in the terminal:

```
python vtkToDx.py DipolesPerNM1.vtk DipolesPerNM1_plasmonic.dx
```

Again, we need to change the data type from “float” to “double”:

```
sed "s/float/double/g" DipolesPerNM1_plasmonic.dx \
> DipolesPerNM1_plasmonic_double.dx
```

Now, you have the final grid file for the plasmonic field.

If you did not complete the previous steps, you could use the example file “DipolesPerNM1_plasmonic_double.dx” in “example_output”. You can simply copy it to the current “ch3” folder.

3.3 Simulate the translocation of a polypeptide without the plasmonic field

Now, we have all the files needed and are ready for MD simulation. First, let’s create a folder called “output” to collect all of our simulation output. Let’s simulate the translocation of a polypeptide in an electric field, but without the plasmonic field. To speed up the translocation and reduce the computational cost, we will apply a bias of 50V, which is usually not the experimental condition. This is just for demonstration purpose. The bias used in the paper by Belkin *et al* [1] is 50 mV.

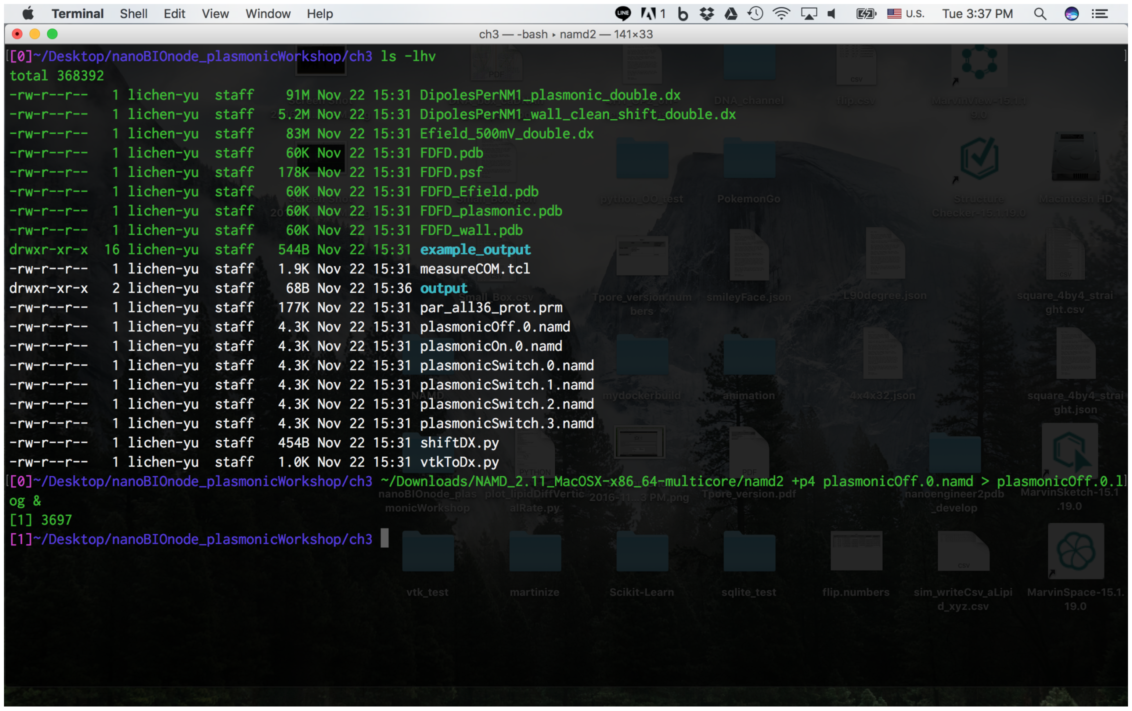
Please type the following command in the terminal:

```
mkdir output
```

Next, use the following command to perform the simulation:

```
namd2 +p<number of processors> plasmonicOff.0.namd > plasmonicOff.0.log &
```

If you did not put the executable `namd2` in your PATH, you will have to execute NAMD from the complete path, Figure 31. NAMD is a parallel program. You can use multiple cores on your laptop to run the simulation by simply replacing “number of processors” with the number you want to use, Figure 31.

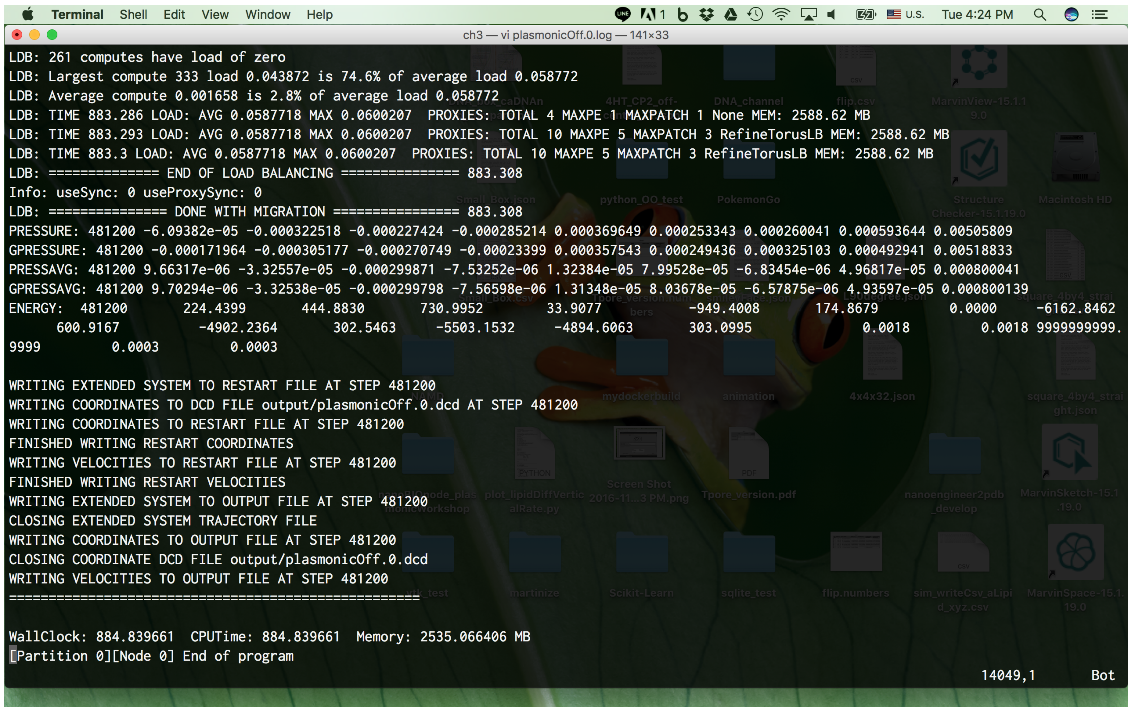


```
[0]~/Desktop/nanoBIOnode_plasmonicWorkshop/ch3 ls -lhv
total 368392
-rw-r--r--   1 lichen-yu  staff    91M Nov 22 15:31 DipolesPerNM1_plasmonic_double.dx
-rw-r--r--   1 lichen-yu  staff   5.2M Nov 22 15:31 DipolesPerNM1_wall_clean_shift_double.dx
-rw-r--r--   1 lichen-yu  staff    83M Nov 22 15:31 Efield_500mV_double.dx
-rw-r--r--   1 lichen-yu  staff    60K Nov 22 15:31 FDFD.pdb
-rw-r--r--   1 lichen-yu  staff   178K Nov 22 15:31 FDFD.psf
-rw-r--r--   1 lichen-yu  staff    60K Nov 22 15:31 FDFD_Efield.pdb
-rw-r--r--   1 lichen-yu  staff    60K Nov 22 15:31 FDFD_plasmonic.pdb
-rw-r--r--   1 lichen-yu  staff    60K Nov 22 15:31 FDFD_wall.pdb
drwxr-xr-x  16 lichen-yu  staff   544B Nov 22 15:31 example_output
-rw-r--r--   1 lichen-yu  staff   1.9K Nov 22 15:31 measureCOM.tcl
drwxr-xr-x   2 lichen-yu  staff    68B Nov 22 15:36 output
-rw-r--r--   1 lichen-yu  staff   177K Nov 22 15:31 par_all36_prot.prm
-rw-r--r--   1 lichen-yu  staff   4.3K Nov 22 15:31 plasmonicOff.0.namd
-rw-r--r--   1 lichen-yu  staff   4.3K Nov 22 15:31 plasmonicOn.0.namd
-rw-r--r--   1 lichen-yu  staff   4.3K Nov 22 15:31 plasmonicSwitch.0.namd
-rw-r--r--   1 lichen-yu  staff   4.3K Nov 22 15:31 plasmonicSwitch.1.namd
-rw-r--r--   1 lichen-yu  staff   4.3K Nov 22 15:31 plasmonicSwitch.2.namd
-rw-r--r--   1 lichen-yu  staff   4.3K Nov 22 15:31 plasmonicSwitch.3.namd
-rw-r--r--   1 lichen-yu  staff   454B Nov 22 15:31 shiftDX.py
-rw-r--r--   1 lichen-yu  staff   1.0K Nov 22 15:31 vtkToDx.py
[0]~/Desktop/nanoBIOnode_plasmonicWorkshop/ch3 ~/Downloads/NAMD_2.11_MacOSX-x86_64-multicore/namd2 +p4 plasmonicOff.0.namd > plasmonicOff.0.l
og &
[1] 3697
[1]~/Desktop/nanoBIOnode_plasmonicWorkshop/ch3 
```

Figure 31: Running NAMD.

The “plasmonicOff.0.namd” is the configuration file for NAMD. It tells NAMD how we want to run the simulation, which molecule we want to use, what parameter to use and so on. You can use your favorite text editor to see the content in “plasmonicOff.0.namd”. For details about NAMD configuration, please visit NAMD User’s Guide.

The “plasmonicOff.0.log” is the output log file. Once the simulation is finished, the “plasmonicOff.0.log” should look like Figure 32, with “End of program” in the last line.


Terminal Shell Edit View Window Help
ch3 — vi plasmonicOff.0.log — 141×33

```
LDB: 261 computes have load of zero
LDB: Largest compute 333 load 0.043872 is 74.6% of average load 0.058772
LDB: Average compute 0.001658 is 2.8% of average load 0.058772
LDB: TIME 883.286 LOAD: AVG 0.0587718 MAX 0.0600207  PROXIES: TOTAL 4 MAXPE 1 MAXPATCH 1 None MEM: 2588.62 MB
LDB: TIME 883.293 LOAD: AVG 0.0587718 MAX 0.0600207  PROXIES: TOTAL 10 MAXPE 5 MAXPATCH 3 RefineTorusLB MEM: 2588.62 MB
LDB: TIME 883.3 LOAD: AVG 0.0587718 MAX 0.0600207  PROXIES: TOTAL 10 MAXPE 5 MAXPATCH 3 RefineTorusLB MEM: 2588.62 MB
LDB: ============== END OF LOAD BALANCING =============== 883.308
Info: useSync: 0 useProxySync: 0
LDB: =============== DONE WITH MIGRATION ================ 883.308
PRESSURE: 481200 -6.09382e-05 -0.000322518 -0.000227424 -0.000285214 0.000369649 0.000253343 0.000260041 0.000593644 0.00505809
GPRESSURE: 481200 -0.000171964 -0.000305177 -0.000270749 -0.00023399 0.000357543 0.000249436 0.000325103 0.000492941 0.00518833
PRESSAVG: 481200 9.66317e-06 -3.32557e-05 -0.000299871 -7.53252e-06 1.32384e-05 7.99528e-05 -6.83454e-06 4.96817e-05 0.000800041
GPRESSAVG: 481200 9.70294e-06 -3.32538e-05 -0.000299798 -7.56598e-06 1.31348e-05 8.03678e-05 -6.57875e-06 4.93597e-05 0.000800139
ENERGY:  481200       224.4399       444.8830       730.9952       33.9077       -949.4008       174.8679         0.0000     -6162.8462
      600.9167         -4902.2364       302.5463     -5503.1532     -4894.6063       303.0995           0.0018          0.0018 9999999999.
9999         0.0003         0.0003

WRITING EXTENDED SYSTEM TO RESTART FILE AT STEP 481200
WRITING COORDINATES TO DCD FILE output/plasmonicOff.0.dcd AT STEP 481200
WRITING COORDINATES TO RESTART FILE AT STEP 481200
FINISHED WRITING RESTART COORDINATES
WRITING VELOCITIES TO RESTART FILE AT STEP 481200
FINISHED WRITING RESTART VELOCITIES
WRITING EXTENDED SYSTEM TO OUTPUT FILE AT STEP 481200
CLOSING EXTENDED SYSTEM TRAJECTORY FILE
WRITING COORDINATES TO OUTPUT FILE AT STEP 481200
CLOSING COORDINATE DCD FILE output/plasmonicOff.0.dcd
WRITING VELOCITIES TO OUTPUT FILE AT STEP 481200
====================================================

WallClock: 884.839661  CPUTime: 884.839661  Memory: 2535.066406 MB
[Partition 0][Node 0] End of program
                                                                                                                     14049,1        Bot
```

Figure 32: The last few lines in "plasmonicOff.0.log" when the simulation is finished. The last line shows "End of program".

We will analyze the result in the last section.

3.4 Simulate the translocation of a polypeptide with the plasmonic field

Next, we will simulate it with the plasmonic field to see if we can trap the polypeptide. The magnitude of the applied plasmonic field corresponds to the field excited by a 166.5 mW incident laser light, which is also higher than in the paper by Belkin *et al* [1].

The process is the same as before, but we will use a different configuration file. In this configuration file, the plasmonic field is applied. You can open the two configuration files to see the difference.

Please use the following command to perform the simulation:

```
namd2 +p<number of processors> plasmonicOn.0.namd > plasmonicOn.0.log &
```

We will analyze the result in the last section.

3.5 Manipulate the translocation of a polypeptide with a laser pulse

In the last set of the simulation, we will simulate manipulating the translocation of a polypeptide with a laser pulse. Essentially, the simulation is done by switching the plasmonic field on and off.

Please use the following command to perform the simulations sequentially (the previous one has to finish before running the next one):

```
namd2 +p<number of processors> plasmonicSwitch.0.namd \
> plasmonicSwitch.0.log &
namd2 +p<number of processors> plasmonicSwitch.1.namd \
> plasmonicSwitch.1.log &
namd2 +p<number of processors> plasmonicSwitch.2.namd \
> plasmonicSwitch.2.log &
namd2 +p<number of processors> plasmonicSwitch.3.namd \
> plasmonicSwitch.3.log &
```

Once the first simulation is finished, the second simulation using "plasmonicSwitch.1.namd" will continue from the last frame of the previous simulation. In this way, we can continue to extend our simulation. In "plasmonicSwitch.0.namd" and "plasmonicSwitch.2.namd", the plasmonic field is turned on. In others, the plasmonic field is turned off.

We will analyze the result in the last section.

3.6 Visualization and Analysis

In the last section, we will use VMD to visualize and analyze the simulation result from previous sections. Please make sure you have VMD installed.

We provided three scripts for visualization: "plasmonicOff_v.tcl", "plasmonicOn_v.tcl" and "plasmonicSwitch_v.tcl" for simulation results in section 3.3, section 3.4 and section 3.5 .

First, make sure you are still in the "ch3" folder. Second, make sure you have the grid files ("DipolesPerNM1_wall_clean_shift_double.dx" and "DipolesPerNM1_plasmonic_double.dx") in the current folder. If you did not create them in the previous sections, please copy them to the current "ch3" folder from the "example_output" folder. Then, make sure you have "plasmonicOff.0.dcd", "plasmonicOn.0.dcd", "plasmonicSwitch.0.dcd", "plasmonicSwitch.1.dcd", "plasmonicSwitch.2.dcd" and "plasmonicSwitch.3.dcd" in the "output" folder. If you did not perform the simulation in the previous sections, please make an "output" folder and copy them from the "example_output" folder to the "output" folder.

Then, you can use the following commands one at a time in the terminal to visualize the simulation trajectory:

```
vmd -e plasmonicOff_v.tcl
vmd -e plasmonicOn_v.tcl
vmd -e plasmonicSwitch_v.tcl
```

But, if your VMD is not in your PATH, you will have to replace `vmd` with the complete PATH, for example, `/Applications/VMD\ 1.9.2.app/Contents/MacOS/startup.command` in MacOS.

Alternatively, you can launch VMD by clicking the VMD icon. Then, in the "VMD Main" window, go to "Extensions" → "Tk Console", Figure 33.

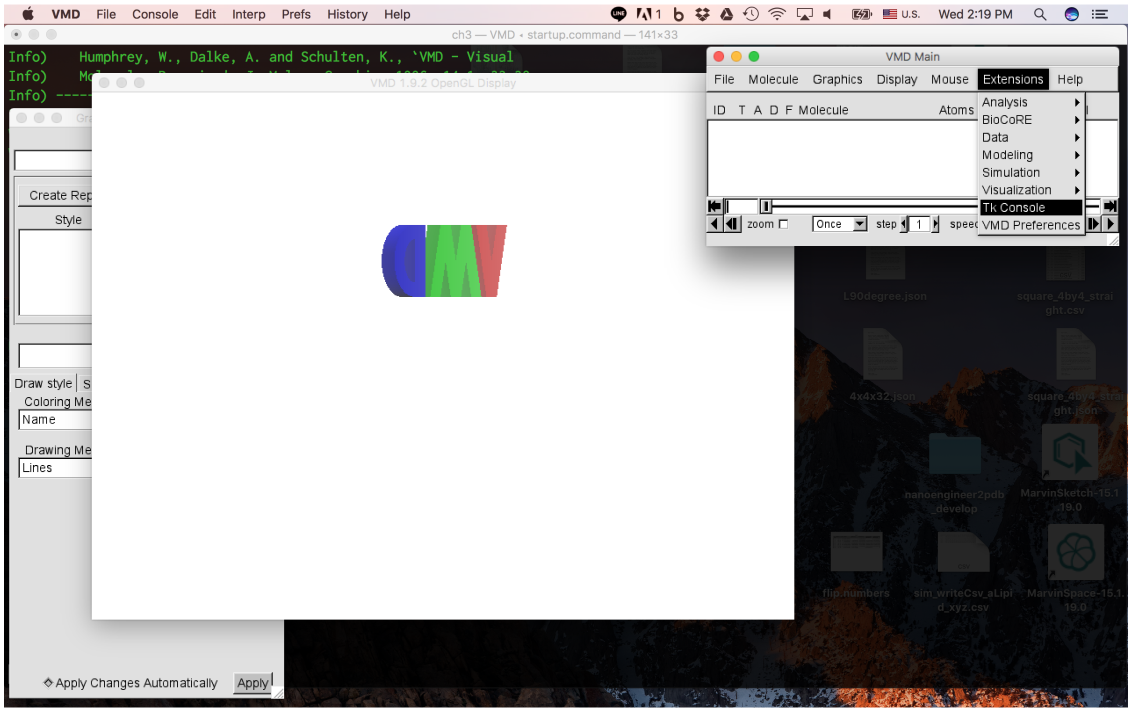
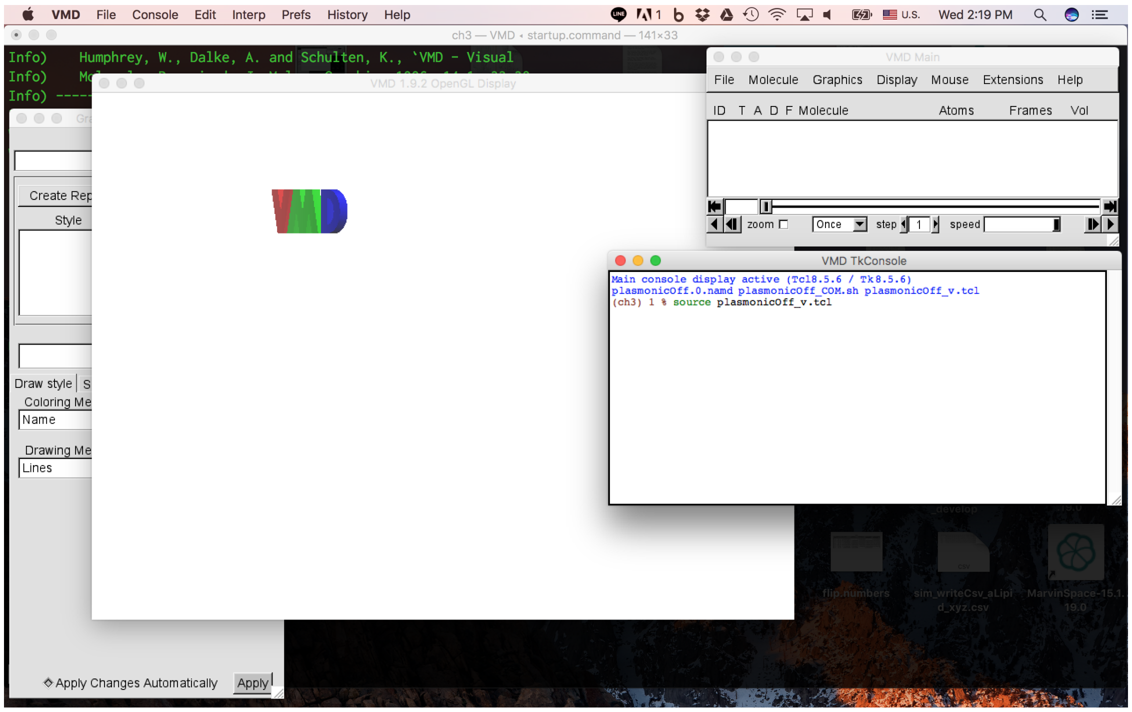

```
Info)    Humphrey, W., Dalke, A. and Schulten, K., `VMD - Visual
Info)    Mo
Info) -----
```

Figure 33: Go to "Extensions" → "Tk Console" in the "VMD Main" window.

You can use `cd`, `ls` and `pwd` commands in the "Tk Console". So, please use them in the "Tk Console" to navigate to the "ch3" folder, Figure 34. Then, you can use the following commands one at a time in the "Tk Console" to visualize the simulation trajectory:

```
source plasmonicOff_v.tcl
source plasmonicOn_v.tcl
source plasmonicSwitch_v.tcl
```


VMD File Console Edit Interp Prefs History Help
ch3 — VMD ◂ startup.command — 141×33
VMD 1.9.2 OpenGL Display
VMD Main
File Molecule Graphics Display Mouse Extensions Help
ID T A D F Molecule Atoms Frames Vol
zoom
Once
step 1
speed
VMD TkConsole
Create Rep
Style
Draw style
Coloring Me
Name
Drawing Me
Lines
Apply Changes Automatically
Apply

```
Info)    Humphrey, W., Dalke, A. and Schulten, K., `VMD - Visual
Info)    Mo
Info) -----
```

```
Main console display active (Tcl8.5.6 / Tk8.5.6)
plasmonicOff.0.namd plasmonicOff_COM.sh plasmonicOff_v.tcl
(ch3) 1 % source plasmonicOff_v.tcl
```

Figure 34: Navigate to the "ch3" folder and type "source plasmonicOff_v.tcl".

The "plasmonicOff_v.tcl" will load the simulation trajectory from section 3.3. The trajectory is 2.4 ps per frame, ∼400 frames in total. The simulation was performed without the plasmonic field, so nothing would prevent the polypeptide from transporting through the nanopore, Figure 35.

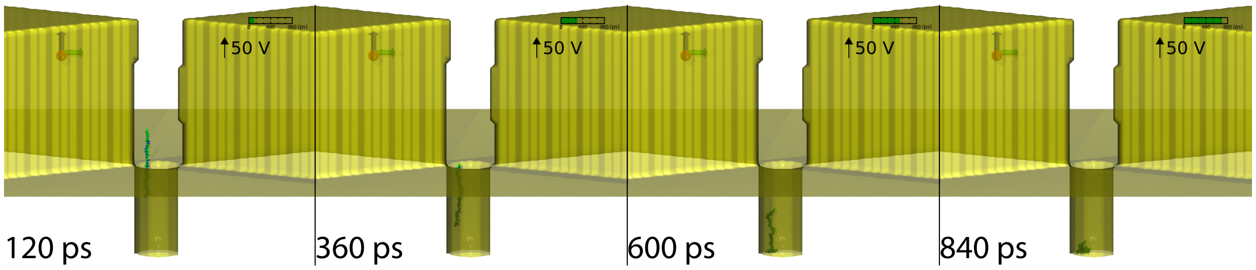


Figure 35: The snapshots of the simulation trajectory without the plasmonic field. Both the gold bow tie structures and the Si_3N_4 membrane are shown as a yellow transparent surface. The polypeptide is colored by green (F) and blue (D).

The "plasmonicOn_v.tcl" will load the simulation trajectory from section 3.4. It should look like Figure 36. The polypeptide will be trapped at the hot spot of the plasmonic field.

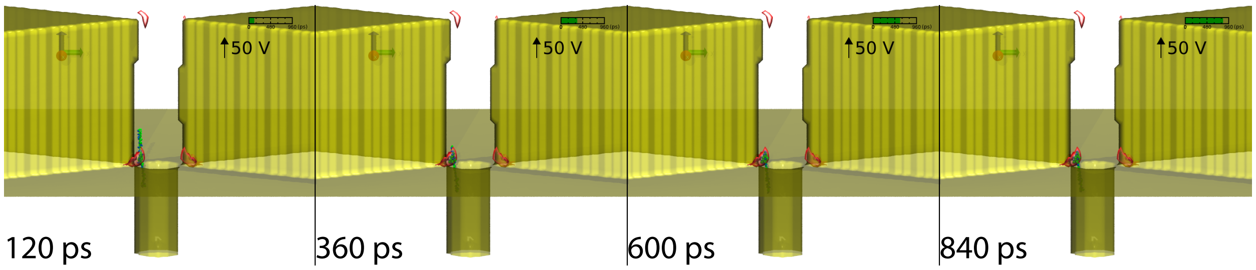


Figure 36: The snapshots of the simulation trajectory with the plasmonic field. Both the gold bow tie structures and the Si_3N_4 membrane are shown as a yellow transparent surface. The polypeptide is colored by green (F) and blue (D). The isosurface of the plasmonic field with value = 4000 is shown as a red transparent surface.

Finally, the "plasmonicSwitch_v.tcl" will load the simulation trajectory from section 3.5. The result should look like Figure 37. By switching the plasmonic field on/off, we can control the translocation of the polypeptide.

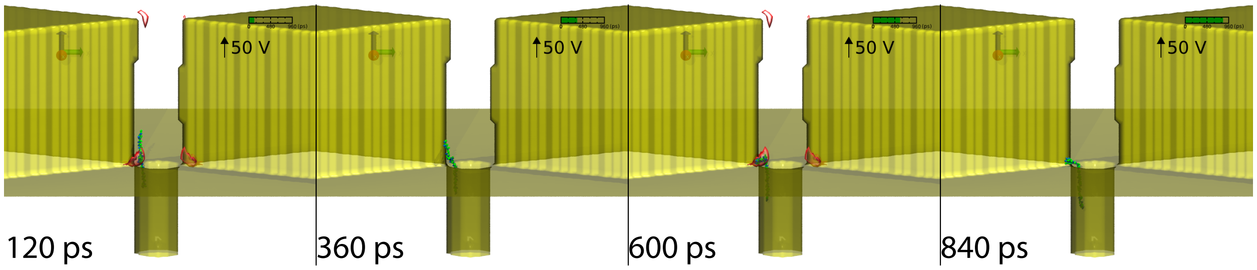


Figure 37: The snapshots of the simulation trajectory with a alternating plasmonic field. Both the gold bow tie structures and the Si_3N_4 membrane are shown as a yellow transparent surface. The polypeptide is colored by green (F) and blue (D). The isosurface of the plasmonic field with value = 4000 is shown as a red transparent surface.

Next, we will demonstrate how to quantitatively compare the translocation from the three simulations. First, let's create a "data" folder inside the "ch3" folder using the following command:

```
mkdir data
```

We will put all the data files in this "data" folder. Then, make sure you have "plasmonicOff.0.xst", "plasmonicOn.0.xst", "plasmonicSwitch.0.xst", "plasmonicSwitch.1.xst", "plasmonicSwitch.2.xst" and "plasmonicSwitch.3.xst" in the "output" folder. If you did not perform the simulation in the previous sections, please make an "output" folder and copy them from the "example_output" folder to the "output" folder.

We provided a tcl script "measureCOM.tcl" to measure the center of the mass of the polypeptide. We also provided three shell script "plasmonicOff_COM.sh", "plasmonicOn_COM.sh" and "plasmonicSwitch_COM.sh", which will call "measureCOM.tcl" to measure the center of the mass of the polypeptide from the simulation trajectory from section 3.3, section 3.4 and section 3.5, respectively.

Here is the content from "plasmonicOff_COM.sh":

```
psfPrefix=FDFD
selText=all
timestep=2

base=output/plasmonicOff
outBase=data/plasmonicOff

dcdFiles=(`ls -v ${base}*dcd | egrep "${base}.[0-9]*.dcd"`)

for dcd in ${dcdFiles[@]}
do

    stringL=`echo "${#dcd} - 4" | bc -l`
    dcdPrefix=${dcd:0:${stringL}}
    outPrefix=`echo "$dcdPrefix" | sed "s:${base}:${outBase}:g"`

    #echo "$psfPrefix $dcd $dcdPrefix $selText $timestep \
    ${outPrefix}.COM.dat"

    vmd -dispdev text -args $psfPrefix $dcd $dcdPrefix $selText \
    $timestep ${outPrefix}.COM.dat < measureCOM.tcl

done
```

Again, you will have to replace `vmd` with the complete PATH for all of the three shell script (.sh), if it is not in your PATH.

Type the following commands one by one to calculate the center of mass of the polypeptide from all three simulations:

```
./plasmonicOff_COM.sh
./plasmonicOn_COM.sh
./plasmonicSwitch_COM.sh
```

The resulting data files "plasmonicOff.0.COM.dat", "plasmonicOn.0.COM.dat", "plasmonicSwitch.0.COM.dat", "plasmonicSwitch.1.COM.dat", "plasmonicSwitch.2.COM.dat" and "plasmonicSwitch.3.COM.dat" will be put in the "data" folder.

If you encountered any issue calculating the center of mass of the polypeptide, you can use the data files in the "example_output" folder to make the figures.

Those data files contain 4 columns. The first is the time in nanosecond and the rest three are X, Y and Z coordinate of the center of mass of the polypeptide at each frame. If you plot the displacement along the Z-axis (Z(t) - Z(0)) for "plasmonicOff.0.COM.dat" and "plasmonicOn.0.COM.dat", the result should look like Figure 38.

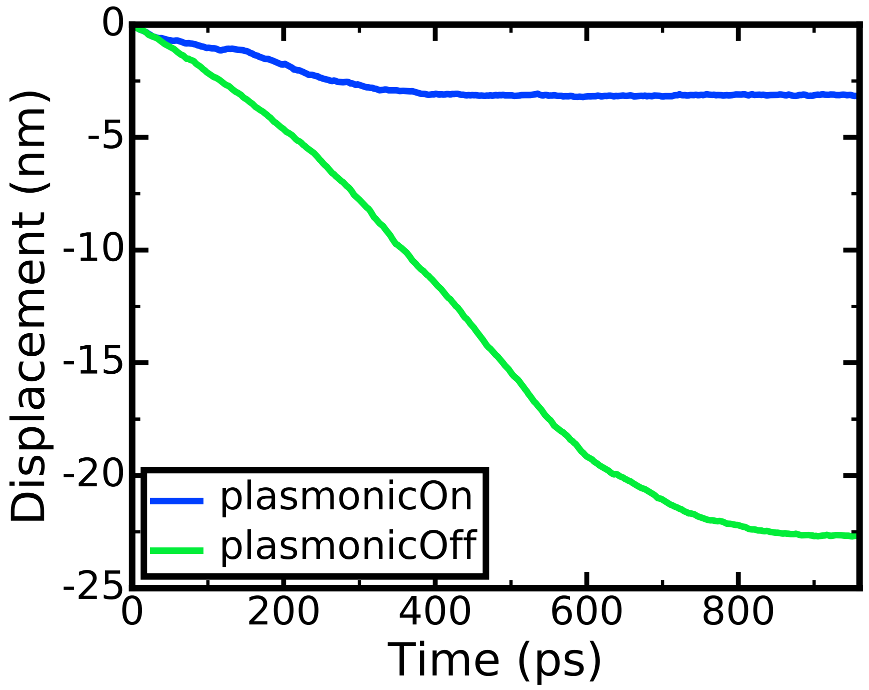


Figure 38: The displacement along the Z-axis versus time without the plasmonic field (green) and with the plasmonic field (blue).

Then, if you plot the displacement versus time for all of the "plasmonicSwitch.*.COM.dat", the result should look like Figure 39.

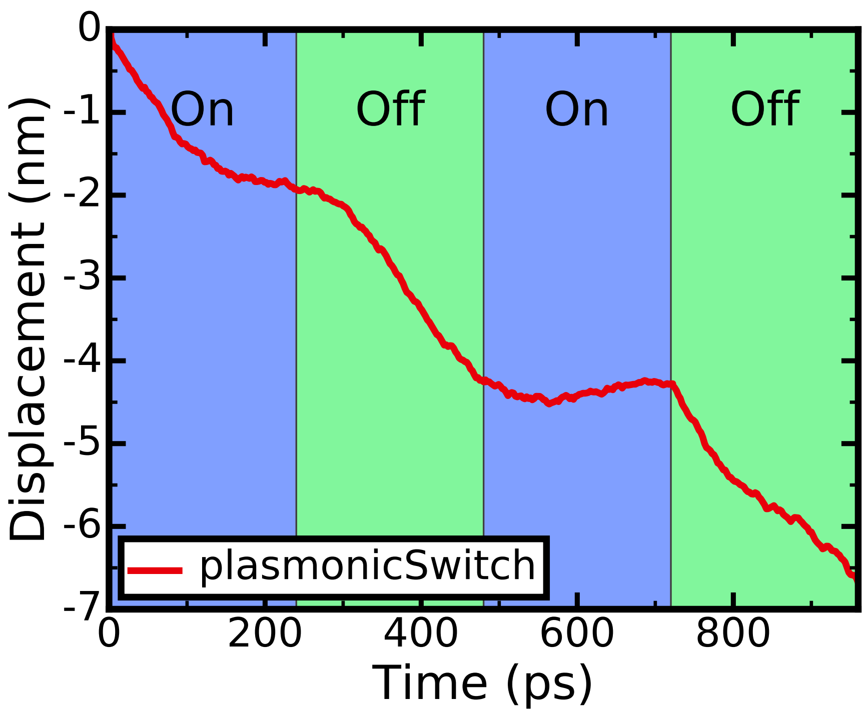


Figure 39: The displacement along the Z-axis versus time with a alternating plasmonic field.

Now, you have completed this tutorial. Congratulations!!

References

[1] Maxim Belkin, Shu-Han Chao, Magnus P. Jonsson, Cees Dekker, and Aleksei Aksimentiev. Plasmonic nanopores for trapping, controlling displacement, and sequencing of DNA. *ACS Nano*, 9(11):10598–10611, 2015.