Tutorials

T1: Framed membranes

The simplest simulation that can be performed by FreeDTS is a flat membrane (a fluid elastic surface) in a PBC box. To do this, we have to first generate a flat TS file. This can be done by using **GEN** binary. Using this you can generate a TS file in a q file format using the below command line.

```
$path/GEN -box 30 30 100 -type flat -o topol.q
```

This command creates a TS file in *q* file format with a box size of 50*50*100. Next, this file name should be added to a topology file name with top extension (here it is named top.top), and we should add the below line in the top file.

```
topol.q 10
```

The number in front of the file gives an *id* to the entire mesh (the value is not important but needs to be unique if multiple files are included). To run a simulation, an input file (with dts extension) is required to define the simulation parameters (see below box and the input file format section).

```
Integrator                      = MC
MC_Moves                        = 1 1 1
Initial_Step                    = 1
Final_Step                      = 100000
Display_periodic                = 1000
OutPutEnergy_periodic           = 100
Restart_periodic                = 10000
Kappa                           = 20 0
OutPutTRJ_TSI                   = 1000  10  TrjTSI
GeneralOutputFilename           = output
```

Using these two files we are now able to run DTS simulations as:

```
$PATH/DTS -in input.dts -top top.top -seed 76532
```

Everything above could be done by running the *./run.sh* script in the T1/dts_tutorials/T1-FramedMembrane_a folder in the source code.

To make a membrane tensionless, we need to add below command to the input file (such the example in the input file section).

```
Frame_Tension                   = on Position_Rescale 0 5
```

In the dts_tutorials/T1/T1-FramedMembrane_b folder run the *./run.sh* script to perform this section.

Each run will give multiple outputs. Folder VTU_Frames contains *paraview* readable files to visualize the evolution of the system. TrjTSI folder provides files in *tsi* format for analysis of the simulation. *output-en.xvg* file contains information about system energy, box size (for constant tensions simulations) membrane volume for constant volume simulations (see the vesicle tutorials) …. as a function of the simulation steps.

T2: Vesicle simulations

Perpetrating a Vesicle

For a simulation of a vesicle, we first need to create a vesicle structure. For this, we use **GEN** script to generate a tetrahedron and run a short simulation with *EllipsoidalShell* confinement to shape the tetrahedron into a vesicle. The below command creates a *TS* file in the shape of tetrahedron. The size of the tetrahedron can be changed by -N option

```
$PATH/GEN -box 50 50 50 -type tetrahedron -N 20 -o topol.q
```

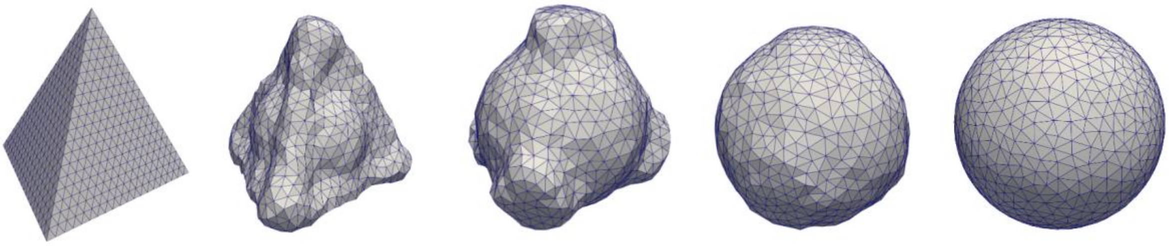
Next, we add the generated TS file to the topology file and run a normal simulation while having below line in the input file (for more see “Membranes in confined spaces” section in the manual).

```
CoupleToRigidWalls              = on EllipsoidalShell  10000  10 10 10  0.01
```

After about 20K steps, you should get a spherical TS file (the run steps may differ for different system)

In the dts_tutorials/*T2-Vesicle* folder run the *./run.sh* script to perform this part. Also see the video in the folder.

Note*: one can easily create a TS file with a spherical shape. However, we prefer this method because it allow us to create a TS file with regular edge size (between range required for DTS simulations) on many different shapes.*



Constant Volume Vesicle

Using the obtain spherical TS file, we can obtain a shape of a vesicle with a constant volume. For this use the **CNV** script and convert last frame of the previous simulation into a q file for a new simulation.

```
$PATH/CNV -in output20.tsi  -o vesicle.q
```

Remove below line from your input file

```
CoupleToRigidWalls              = on EllipsoidalShell  10000  10 10 10  0.01
```

And add below line to the input file (see “Osmotic pressure or fixed volume simulation” section)

```
Volume_Constraint               = on 1000000     0  0.1     0.4
```

In the dts_tutorials/*T2-Vesicle* folder run the *./run_2.sh* script to perform this part.

T3: Proteins

In this tutorial, we want to make a flat membrane covered by 20% curvature inducing proteins.

Inclusions: Proteins are modeled as inclusions. The type and the model parameters of the corresponding inclusion must be defined in the input file (*.dts file). Below is an example of defining two inclusion types in the input files.

```
INCLUSION
Define 2 Inclusions
SRotation Type   K   KG  KP  KL  C0  C0P C0L
0         Pro1   10  0   0   0   0.4       0   0
0         Pro2   20  0   0   0   -0.4      0   0
```

Interactions: Inclusion-inclusion interactions are also defined in the input file. For interaction type 1 (the tested version) $E_{ij} = -A + B\cos N\Theta$ it will be as:

```
Inclusion-Inclusion-Int
```

```
i     j     ftype    N      A      B
```

In this dts file, this will be defined as below. Please note: this section MUST always be the last section of the input file (*.dts file).

```
Inclusion-Inclusion-Int
1     1     1    2      2      0.0
1     2     1    2      0      0.0
2     2     1    2      2      0.0
```

Defining inclusions: The number of the inclusions in a simulation can be defined in two different ways. If the simulation starts from a q file, a random distribution of inclusions must be defined in the input file.

```
GenerateInclusions
Selection_Type Random
TypeID      1      2      3
Density     0.3    0.1      0

```

However, if the simulation started from a tsi file, inclusions will be read from the inclusion section of this file.

In this first step of the tutorial, we will generate the inclusions from an input file and force them to cluster to enhance the aggregation of the inclusions (to accelerate the process). For this, we will turn off the vertex, edge flip and box change moves.

```
MC_Moves                                = 0  0  1
```

In the dts_tutorials/*T3-Membranes_Inclusions* folder run the *./run.sh* script to perform this.

Next, we use the last frame (tsi file) of the previous simulation to start a new simulation with all the required moves.

In the dts_tutorials/*T3-Membranes_Inclusions* folder run the *./run_2.sh* script to perform this.

T4: Protein Sorting

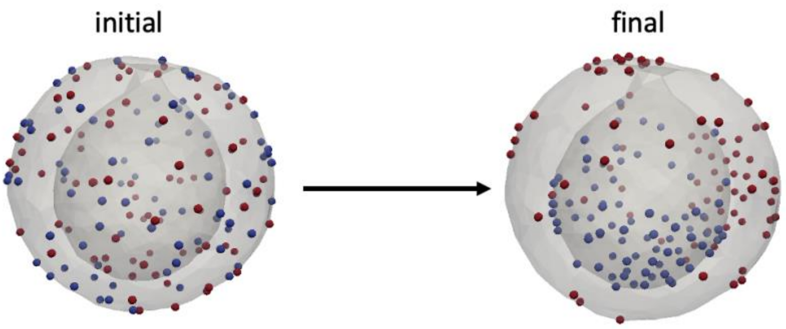
With FreeDTS, we can look at the protein sorting without allowing for membrane shape change. In this way, we can talk for example the shape of membrane structure from Cryo-ET and obtain protein organizations on the membrane. For this, we need to turn off the vertex, edge flip moves and only allow for inclusion moves.

```
MC_Moves                                = 0  0  1
```

We will take the stomotosyte shape in the tutorial 2 and add two different protein type.

```
INCLUSION
Define 2 Inclusions
SRotation    Type    K    KG   KP   KL   C0      C0P   C0L
1          Pro1    20   0    0    0    0.3       0    0
1          Pro2    20   0    0    0   -0.3       0    0
1          Pro2    20   0   10    0    0    1    0
GenerateInclusions
Selection_Type Random
TypeID      1      2      3
Density     0.1    0.1   0.0
```

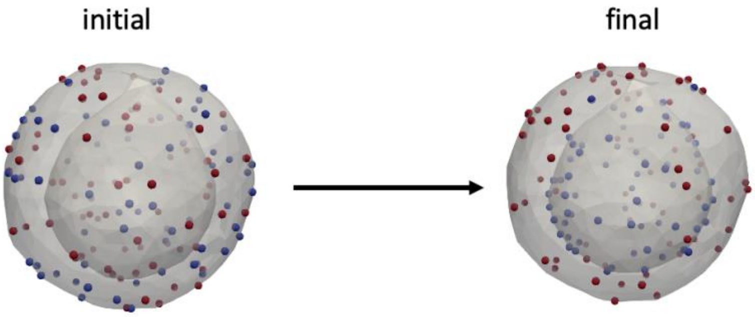
```
Inclusion-Inclusion-Int
1     1     1    0      1      0.0
2     2     1    0      1      0.0
1     2     1    0      1      0.0
```



In the dts_tutorials/ *T4-protein_sorting* folder run the *./run.sh* script to perform this.

The same simulation can be done a system that inclusion one prefers inclusion 2 but still the get sorted by the membrane curvature. For this

```
Inclusion-Inclusion-Int
1     1     1    0      0      0.0
2     2     1    0      0      0.0
1     2     1    0      1      0.0
```



In the dts_tutorials/ *T4-protein_sorting* folder run the *./run_2.sh* script to perform this.

T5: Pulling a membrane nanotube

To pull a nanotube from a membrane, we need to have a membrane under tension (see previous tutorials for such a simulations). Then we can apply a harmonic potential between two groups, one group can be the single vertex while the other group can be all the other vertices. Using a corresponding index file (See the index file section) and adding below command to the input file we have everything we need to pull a nanotube from a flat membrane.

```
HarmonicPotentialBetweenTwoGroups = on 10 100 1000000 Group1 Group2 0 0 1
```

Next, just run below command and wait for the nanotube to form.

```
$PATH/DTS -in input.dts -top top.top -seed 76532 -ndx index.inx
```

In the dts_tutorials/*T4-tether_pulling* folder run the *./run.sh* script to perform this tutorial.

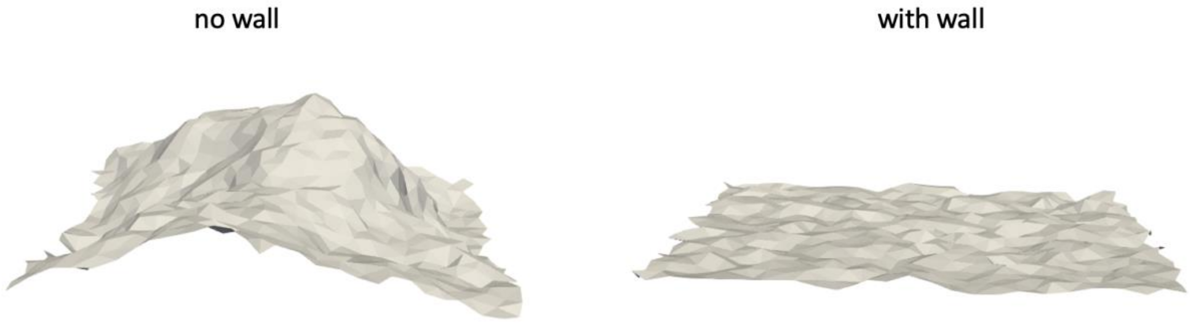
Note: this tutorial might take long.

T6: Confirmed membranes

How membranes behave in a confined space. This can be studied using FreeDTS. In this tutorial we will look how two sandwiching hard walls. For this, we need add below command to the inpot file

```
CoupleToRigidWalls                = on TwoFlatParallelWall  10000  0.5
```

We also set the kappa to 4 to make the


no wall

with wall

In the dts_tutorials/*T6_confined_membranes* folder run the *./run.sh* script to perform this tutorial.