Table of Contents

[Some environment variables 1](#_Toc361648584)

[Experimental data 1](#_Toc361648585)

[Basis data; creating Nexus files for experimental runs 1](#_Toc361648586)

[CNCS data 1](#_Toc361648587)

[AMBER input from Stan Anderson 2](#_Toc361648588)

[Single molecule runs 2](#_Toc361648589)

[RMS calculations 2](#_Toc361648590)

[Sassena calculations 2](#_Toc361648591)

[Trajectories without COM translations and global rotations are too rigid 2](#_Toc361648592)

[Longer simulations are needed for precision in the µeV scale 3](#_Toc361648593)

[Combination of atomic diffusion and libration 4](#_Toc361648594)

[Meeting with Niina 5](#_Toc361648595)

[Is the Lorentzian HWHM is independent of Q? 5](#_Toc361648596)

[Torsional angles in the trajectory 5](#_Toc361648597)

[Modifying the Methyl torsional potential 6](#_Toc361648598)

[Inelastic domain 7](#_Toc361648599)

[Update slides 7](#_Toc361648600)

[Runs at Low temperatures 7](#_Toc361648601)

[Origin of spurious peaks 8](#_Toc361648602)

[Spectrum at low temperatures 11](#_Toc361648603)

[Preparing the crystal simulation 12](#_Toc361648604)

# Some environment variables

CODED=/home/jbq/code/projects/POSS/me8t8

PROJD=/projects/research/POSS/me8t8

# Experimental data

## Basis data; creating Nexus files for experimental runs

* Nina sent me Dave group files (BSS\_\*\_1run\_divided.dat) for the experimental runs, which I stored under *$PROJD/expdata/*. I made soft links of the sort *exp\*K.dat* to these files. The S(Q,E) in the Dave groups was normalized by a Vanadium run.
* Here’s the conversion between run number and temperature:

15026 13

15027 13

15031 100

15035 150

15039 200

15044 175

15047 225

15050 250

15053 300

15056 350

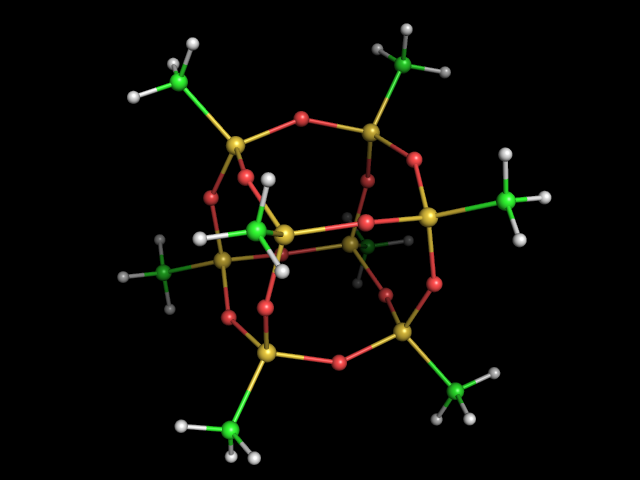
* Create Nexus files for each Dave group: me8t8.py mantidjobs --kwargs="job=create experimental Nexus files". Creates *exp\*K.nxs*, *elastic.nxs*, and *resolution.nxs*.

## CNCS data

* Niina sent me Nexus files with reduced data for each detector for different runs, which I stored under *$PROJD/expdata/CNCS/*. Files names are of the type CNCS\_run\_number\_\*.nxspe containing the reduced data and CNCS\_run\_number\_\*\_summary.txt containing a description of the reduction procedure.
* Excel spreadsheet *runlist.xlsx* contains the temperature of each run, among other things.

# AMBER input from Stan Anderson

* Stan Anderson, from Michaels T. Bowers’ group, sent me *me8t8.prmtop*, *me8t8.mdcrd*, and *me8t8.mol2* files. He also sent me ***siparms.pcy*** containing the force-field.
* I use ptraj to generate *me8t8.pdb*.



*$ PROJD/charmm/me8t8.pdb(pse,png)*

* The prmtop file contains a total residual charge, that is, the molecule is not electrically neutral. Thus, I redistributed this charge among all atoms equally.

met8t8.py single\_molecule --kwargs="job=remove residual total charge,pdb=$PROJD/me8t8.pdb,prmtop=$PROJD/me8t8.prmtop,outprmtop=junk.prmtop"

mv -f junk.prmtop me8t8.prmtop

# Single molecule runs

* As starter, we do single molecule runs and the idea is to compare to experiment. The next step (if comparison is too crappy) is to create a crystal, for which lattice cell parameters have been published in the literature.
* Directory *$PROJD/charmm/nchain\_1*.
* For every temperature, we do a short equilibration (min0) followed by 4ns equilibrium run (run0). We output every 0.1ps for a trajectory containing 40,000 frames. Notice there is no PME because the molecule is small. We can calculate the electrostatic energy term with a sum in real space.
* Simulations are carried out in the NVT ensemble, with a Langevin bath and no pressure bath. Pressure does have no sense for a single molecule in vacuum.

## RMS calculations

* rms first conformation. Output trajectory run0\_rms2first.dcd in order to calculate S(Q,E) without overall translations and rotations.

ptraj11 me8t8.prmtop < $PROJD/charmm/nmolec\_1/ptraj/rms2first.in generates *rms2first.dat* and *run0\_rms2first.dcd*.

* the rms calculation is done only with the Si atoms (the corners of the cube).

# Sassena calculations

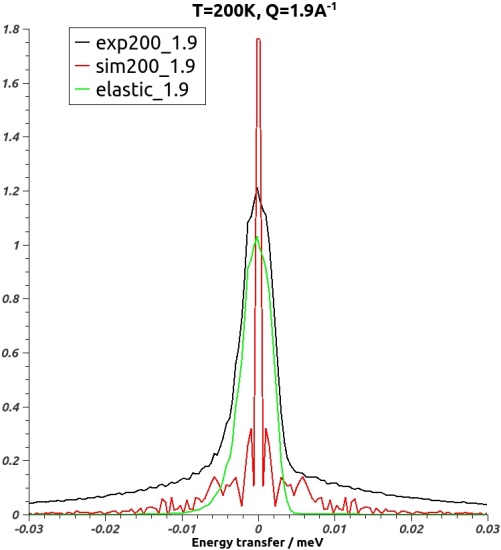
* Sassena calculations for the trajectory where we remove translations and rotations of the COM. Unfortunately, sassena cannot read ptraj output files *run0\_rms2first.dcd*. The fix consist in loading *run0\_rms2first.dcd* and then saving as DCD. Sassena can process a DCD trajectory output by VMD.

met8t8.py single\_molecule --kwargs="job= convert ptraj DCD to vmd DCD,indcd= run0\_rms2first.dcd ,outdcd=junk.dcd "

mv -f junk.dcd run0\_rms2first.dcd”

# Trajectories without COM translations and global rotations are too rigid

* Removal of center of mass (COM) translations and global rotation around the COM by doing rms2first yields S(Q,E) denoting a ‘too rigid’ system, when compared to experiments.
* Note: global rotation was obtain by minimizing RMS of the Silicium atoms. This leaves Methyl group rotations in the ‘vibrational’ part

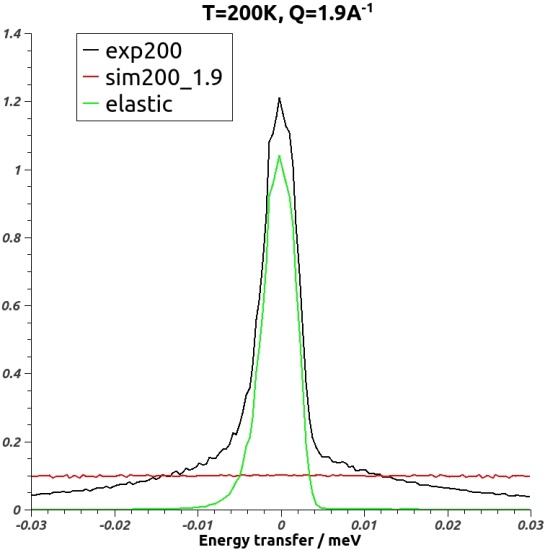


*charmm/nmolec\_1/T\_200/compare\_run0\_SQE\_rms2first\_to\_exp\_Q1.9.jpeg*

* In the above picture, if we remove the contribution from the elastic line (green) from the experiment (black) we would obtain a black-green curve with maxima of ~0.2 at E=0 and slowly decaying. On the other hand, the simulation (red) shows a narrow peak followed by a very fast decay. Convolution with resolution function will still yield a decay much more pronounced than that of the hypothetical black-green curve.

Trajectories with simulated COM translations and global rotations are too soft

* If we calculate S(Q,E) for the simulated trajectory, which included translations of the center of mass and rotations around the COM, then it decays way to fast.

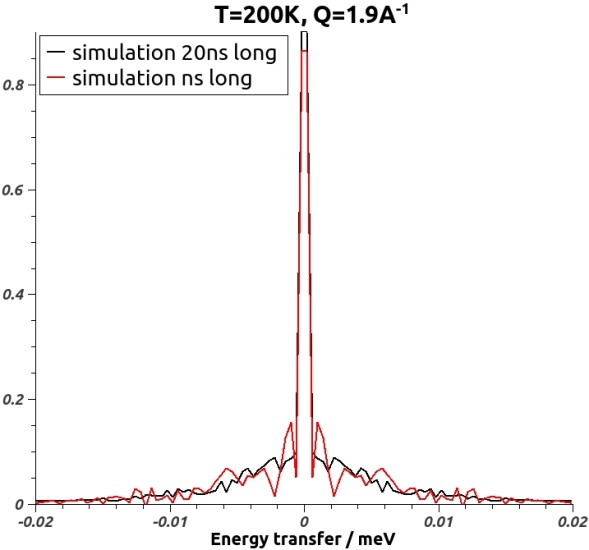


*charmm/nmolec\_1/T\_200/compare\_run0\_SQE\_to\_exp\_Q1.9.jpeg*

* The computed S(Q,E) (red line) is flat, meaning all the elastic intensity has gone into diffusive process.

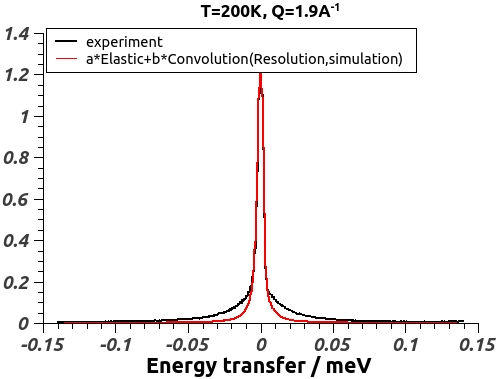
# Longer simulations are needed for precision in the µeV scale

* For simulations 4ns long (run0), S(Q,E) is wiggly in the µeV scale. The energy↔time conversion is 4.136ns↔1 µeV. We do **run1** with 20ns in order to get a minimum ΔE=0.2µeV. Energy points in the simulation are separated by 0.4µeV.
* The figure below shows how the increased simulation time produces a smoother S(Q,E) in the µeV scale. This curve will even be smoother once we do convolution with the resolution function.



*charmm/nmolec\_1/T\_200/compare\_4ns\_to\_20ns.jpeg*

* The increase in resolution with the 20ns has not resolved the issue regarding the rigidity of the simulations with respect to experiments. Below a comparison of simulation and experiment at 200K are shown. The fit for the simulations was



*charmm/nmolec\_1/T\_200/compare\_run1\_SQE\_to\_exp\_Q1.9.jpeg*

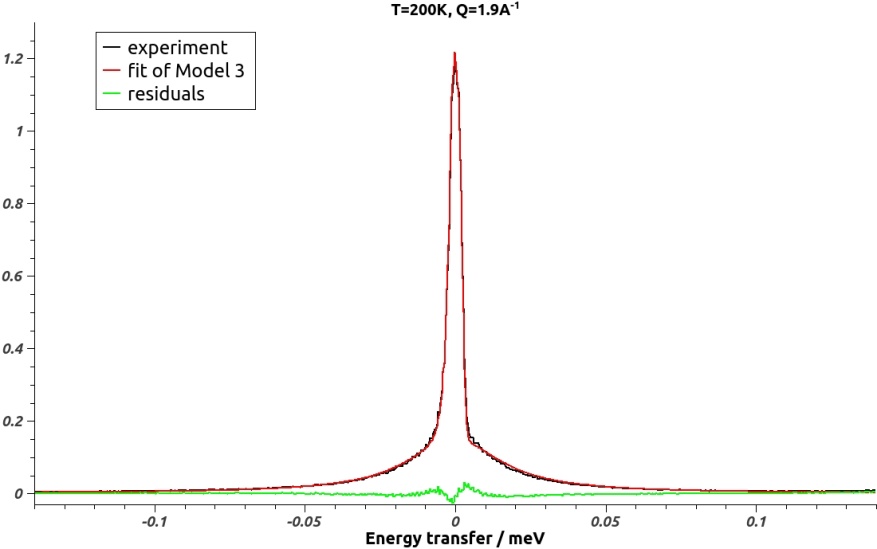
# Combination of atomic diffusion and libration

* The single molecule libration motions, S(Q,E) after rms to first conformation, decays too fast to reproduce the experimental S(Q,E). This could indicate that each molecule in the sample undergoes some degree of rotation around the COM plus diffusion of the COM.
* Most of the signal comes from incoherent scattering of the methyl hydrogens, which are located from the same average distance to the COM and undergo the same type of global rotational diffusion. Thus, we will merge into a single term representing the translational diffusion of each hydrogen (not of the COM)
* The translational diffusion is represented by a [Lorentzian](http://www.mantidproject.org/Lorentzian).
* We compare three increasingly complex models, although all of them have three degrees of freedom (a, b, and HWHM of the Lorentzian. All other parameters are fixed).

|  |  |  |
| --- | --- | --- |
| Model 1 |  | elastic line plus a Lorentzian (no vibration) |
| Model 2 |  | Lorentzian shaped by resolution |
| Model 3 |  | Translation+vibration |

* Mantid is used to fit the Models for the case T=200K, Q=1.9A-1 ( or L~3.3 Å ). We ran Sassena on trajectory run1\_rms2first.dcd and perform FFT to get Srmsd2first. Mantid has trouble doing the fit of Model 3 because of the double convolution. We take advantage of the commutative property of the convolution operator in order to pre-calculate , since this function does not contain any fitting parameter. Then we use it to do the remaining convolution with the Lorentzian. Below are the results of the fits:

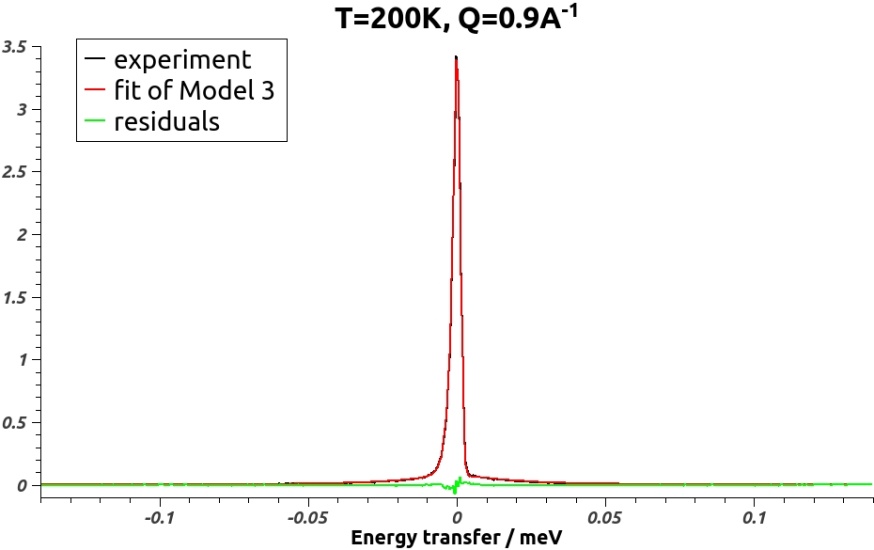
|  |  |
| --- | --- |
| Model # | χ2 of the fit to experimental S(Q=1.9A-1,T=200K,E) |
| 1 | 17.87 |
| 2 | 14.65 |
| 3 | 11.27 |



*charmm/nmolec\_1/T\_200/fit\_model3\_run1\_rmsd2first\_T200\_Q1.9.jpeg*

* For the fit shown, HWHM=15.708µeV in the Lorentzian representing atomic diffusion, or HWHM=15.708µeV / (4,136ps x 1 µeV ) = 1/(263ps). Since HWHM=D x Q2, we obtain (Q=1.9A-1 or L~3.3 Å ) a diffusion coefficient D=1.052Å2/ns. Thus, the hydrogens diffuse about 1Å every nanosecond. Compare with the diffusion coefficient of bulk water at room temperature and 1atm of 230Å2/ns. In these conditions, the water molecule diffuses about 15Å every nanosecond.
* The procedure is repeated for Q=0. 9Å-1, or L~7Å which precluded the possibility of Methyl rotations if there is any doubt.

|  |  |
| --- | --- |
| Model # | χ2 of the fit to experimental S(Q=0.9A-1,T=200K,E) |
| 1 | 6.6 |
| 2 | 5.8 |
| 3 | 5.5 |



* Difference in the fitting cost of the models are less pronounced for Q=0.9Å -1 compared to the case Q=1.9Å-1. This is to be expected because we are looking at a higher length scale, and the higher the length scale, the less relevance has adding the vibrational component, which is restricted to lengths scales <2Å.
* For the fit shown, HWHM=19.202µeV in the Lorentzian representing atomic diffusion, or HWHM=19.202µeV / (4,136ps x 1 µeV ) = 1/(215ps). Since HWHM=D x Q2, we obtain (for Q=0.9A-1 or L~3.3 Å ) a diffusion coefficient D=5.7Å2/ns. This results only tell us that a smaller fraction (most of the signal at this Q is elastic) of the POSS molecules diffuse faster than the fraction for the case Q=0.9Å-1. Again, this is to be expected, because we are looking at higher length scales but within the same time scale (the time-scale of BASIS)

# Meeting with Niina

* Niina showed me her model fitting of the experimental results. She used a linear background plus one Lorentzian:

Free parameters are *f, p1, p2, a, b* plus constrain *p1+p2=1*.

Parameters *f, p1, p2, a, b* are Q-dependent, that is, the fits are done independently for each Q.

* Niina showed me that Δ is independent of Q. This would be the case if the diffusion was confined as in the case of Methyl rotations. In the case of pure translational diffusion, one has .

# Is the Lorentzian HWHM is independent of Q?

The two fits I did at T=200K gave me different HWHM. Thus, I now try to reproduce Niina’s results that show that Δ is independent of Q. We use Mantid and model obtaining:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Q | 0.3 | 0.5 | 0.7 | 0.9 | 1.1 | 1.3 | 1.5 | 1.7 | 1.9 |
| Δ | 0.017117 | 0.0169534 | 0.017093 | 0.0017812 | 0.0165092 | 0.0162985 | 0.0162563 | 0.0162 | 0.0162498 |
| χ2 of the fit | 1.85813 | 2.44625 | 2.25578 | 2.54636 | 2.3874 | 2.49816 | 2.77241 | 3.65366 |  |

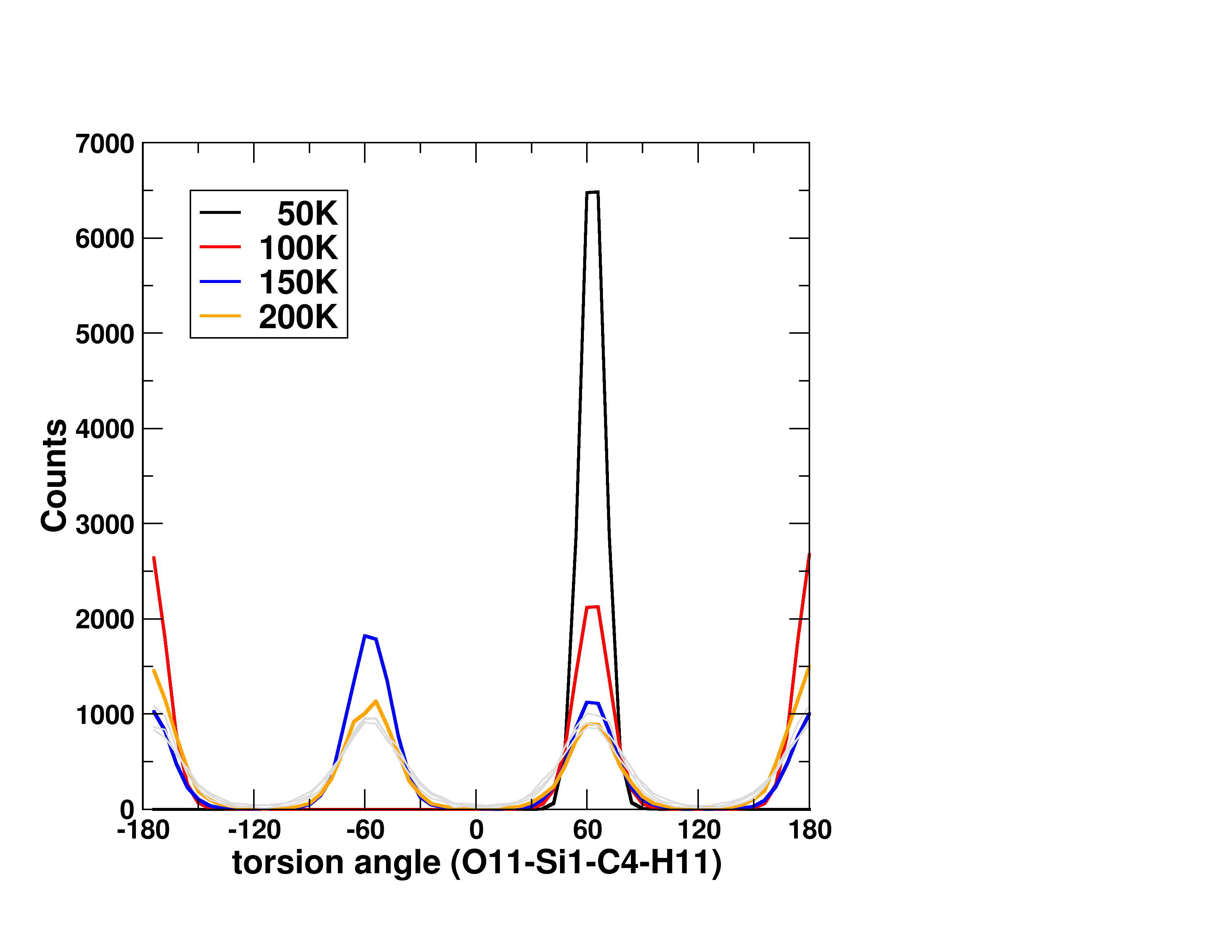
* Inclusion of the linear background in the model reduces χ2 about five times.
* Δ seems to be independent of Q. This indicates that the system is not undergoing pure translational diffusion because for translational diffusion and that would yield a Q-dependent diffusion constant. In other words, the diffusion constant would depend on the length scale we are observing. Furthermore Bee’s book indicates a Lorentzian with a Q-independent Δ for the model of discrete jumps in a circle, as is the case of the Methyl groups.
* Assuming the model of discrete jumps in a circle, we obtain an average residence time of (4,136ps x 1 µeV )/17µeV~250ps. Thus and according to experiments, the hydrogen would wait on average 250ps before jumping over the torsional potential barrier when T=200K.

# Torsional angles in the trajectory

* Use “dihedral” command from ptraj to find the different torsional angles for the hydrogens. There are 24 hydrogens in the Methyl groups but we only need 8 angles. If we know the torsion angle for one of the hydrogens in a Methyl group, then we automatically know the torsion angles for the other two.
* These are the list of atoms making up the 8 dihedrals of interest

|  |  |  |  |
| --- | --- | --- | --- |
| O11 | Si1 | C4 | H11 |
| O12 | Si2 | C2 | H4 |
| O5 | Si3 | C7 | H19 |
| O4 | Si4 | C6 | H16 |
| O9 | Si5 | C3 | H7 |
| O7 | Si6 | C1 | H1 |
| O12 | Si7 | C5 | H13 |
| O8 | Si8 | C8 | H22 |

* We run ptraj me8t8.prmtop ../ptraj/dihedral\_O11Si1C4H11\_run1\_rms2first.in to get the torsion angles for the first methyl group, obtaining file



* The distribution of torsion angles for the first Methyl group shows it is frozen a 50K, it populates only two states at 100K, and the three states (albeit with different weights) at 150K.
* A quick inspection of the torsion angle time evolution for the 8 angles at T=200K in the trajectory shows about 100 jumps in the 2ns run, which translates to 200ps resident time. This rough estimation corresponds quite well with the 250ps residence time obtain with the fitting of the Lorentzian to the experiment at 200K.

# Modifying the Methyl torsional potential

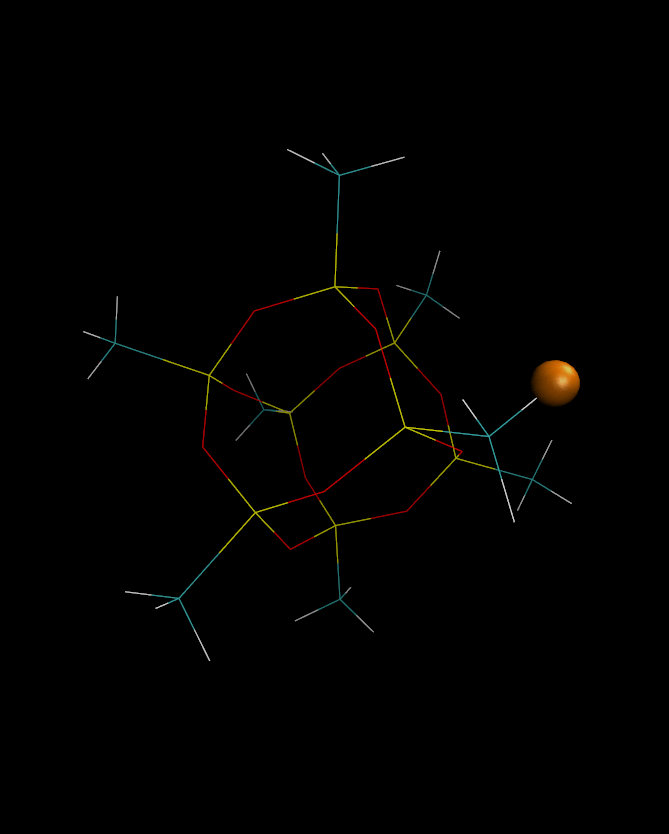
* The methyl torsional potential force constant is defined under entry FLAG DIHEDRAL\_FORCE\_CONSTANT in me8t8.prmtop, and it is the second number (1.11111111E-01). I found out this by looking at FLAG DIHEDRALS\_INC\_HYDROGEN. See document *$HOME/Papers/amber\_prmtop\_specs.pdf* for a detailed description.
* I set up subdirectories K$K/T\_200/ under charmm/nmolec\_1/ in order to study S(Q,E) under different torsion potentials. The force constant of the original potential is K=1.1111111E-01Kcal/mol. We vary it in the range 0.02Kcal/mol to 0.08Kcal/mol. We selected T=200K for now.
* Job run1 simulates for 20ns every 1ps for a total of 20,000 frames. We remove translations of the center of mass (COM) and rotations around the COM by RMS to Silica atoms, and save trajectory *run1\_rms2first.dcd*. Then we use Sassena to calculate *fqt\_int\_run1\_rms2first.h5*. These are the I(Q,t) we will use to compare to experiment.
* Below is a fit of S(T=200K,Q=1.9Å-1,E) with different force constants. The model we use is:

|  |  |  |  |
| --- | --- | --- | --- |
| *K****0.06****/T\_200/compare\_run1\_SQE\_to\_exp\_Q1.9.jpeg* | *K****0.07****/T\_200/compare\_run1\_SQE\_to\_exp\_Q1.9.jpeg* | *K****0.08****/T\_200/compare\_run1\_SQE\_to\_exp\_Q1.9.jpeg* | *T\_200/compare\_run1\_SQE\_to\_exp\_Q1.9.****v2****.jpeg* |

* It seems K~0.07Kcal/mo optimizes the fit, which is a ~37% reduction with respect to the original value of 0.1111111.

Methyl rotation movie

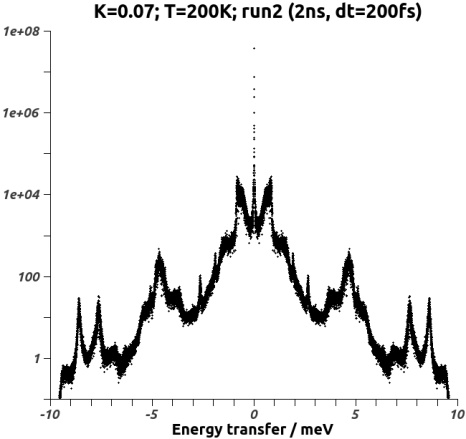
* I use trajectory *K0.07/T\_200/run1\_rms2first.dcd* for the movie.
* The movie maker plugin in vmd allowed us to make an animated GIF (VMD→Extensions→Visualization→Movie Maker. The final file is *charmm/nmolec\_1/K0.07/T\_200*/*me8t8.gif*. I created a VDW representation of the first hydrogen (index 1) in order to visualize its rotations as time goes on. There were several nuances with the plugin.
  + There were too many snaphsots in *run1\_rms2first.dcd*. We removed most of the frames (vmd> animate delete beg 500 end 19999)since there are enough methyl rotations in the first 0.5ns. Furthermore, we skipped 1 frame(vmd> animate delete skip 1). The final number of frames was 250.
  + Set working directory to /tmp
  + Set Name of movie to ‘me8t8
  + Renderer→snapshot
  + Movie Settings→trajectory
  + Format→Animated Gif
* Notice: the plugin dumps one image per snapshot to */tmp*, then calls the Linux command convert as convert –delay 4.7 –loop 4 /tmp/me8t8.\*.ppm /tmp/me8t8.gif. The delay and loop options are not desirable so I stopped the conversion process and retyped the command in a terminal but without the aforementioned options. You can tell when the plugin is doing the conversion because the command is printed to the vmd> console



*charmm/nmolec\_1/K0.07/T\_200*/*me8t8.gif*

# Inelastic domain

* Niina showed me peaks in the [4,9]meV energy scale at T=200K. Thus, we do a simulation to check if we observe the same peaks. Job “run2” in *charmm/nmolec\_1/K0.07/T\_200/* last 2ps and writes every 200ps for a total f 10,000frames. The time between consecutive frames of 200ps corresponds to ΔE~20meV so that the [4,9]meV scale is confortably covered.
* The incoherent structure factor *charmm/nmolec\_1/K0.07/T\_200/fqt\_inc\_run2\_rms2first.h5* (computed in Chadwick) shows cualitative agreement with the experiment.



*charmm/nmolec\_1/K0.07/T\_200/run2\_Q1.9.jpeg*

# Update slides

* In the laptop, *My Documents/projects/research/POSS/me8t8/* I started a powerpoint presentation in order to log meeting presentations and various updates.

# Runs at Low temperatures

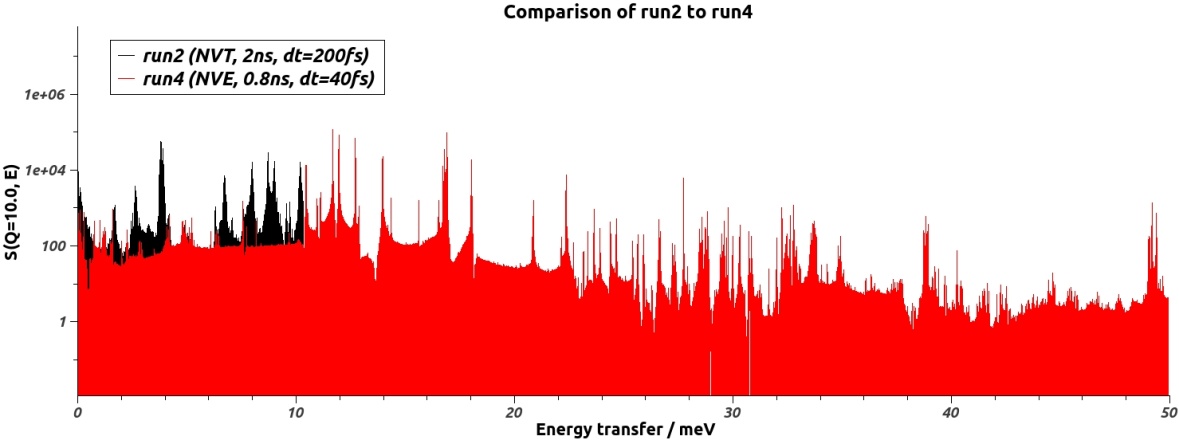
* We run simulations at T=5K in order to match data gave to me by Niina at that temperature:
* Job “**K0.07/T\_5/run1**” does 20ns in order to obtain the diffusivity regime
* Job “**K0.07/T\_5/run2**” does 2ns in order to obtain the inelastic regime in the [1,10]meV, with LangevingDamping=0.1(1/ps), corresponding to a frequency of 24µeV. This ensures that the friction coefficient of the heat bath has a small damping effect on all vibrations in the higher-frequency [1,10]meV regime. By contrast, job “**K0.07/T\_5/run3**” is the same as job “**K0.07/T\_5/run2**” except that LangevingDamping=5(1/ps), corresponding to a frequency of 1.2meV. Here the friction coefficient should have a dampening effect on the vibrations in the [1,10meV].
* A comparison of S(Q=10.0, E) from jobs run2 and run3 confirm the dampening effect of LangevingDamping=5(1/ps), except for one strong peak at ~4meV. The same spectra is observed for all other values of Q.

|  |  |
| --- | --- |
| *charmm/nmolec\_1/K0.07/T\_5/run2\_Q10.jpeg* | *charmm/nmolec\_1/K0.07/T\_5/run3\_Q10.jpeg* |

* At this low T, we should run in the NVE ensemble, not to destroy any peaks due to coupling with heat bath.

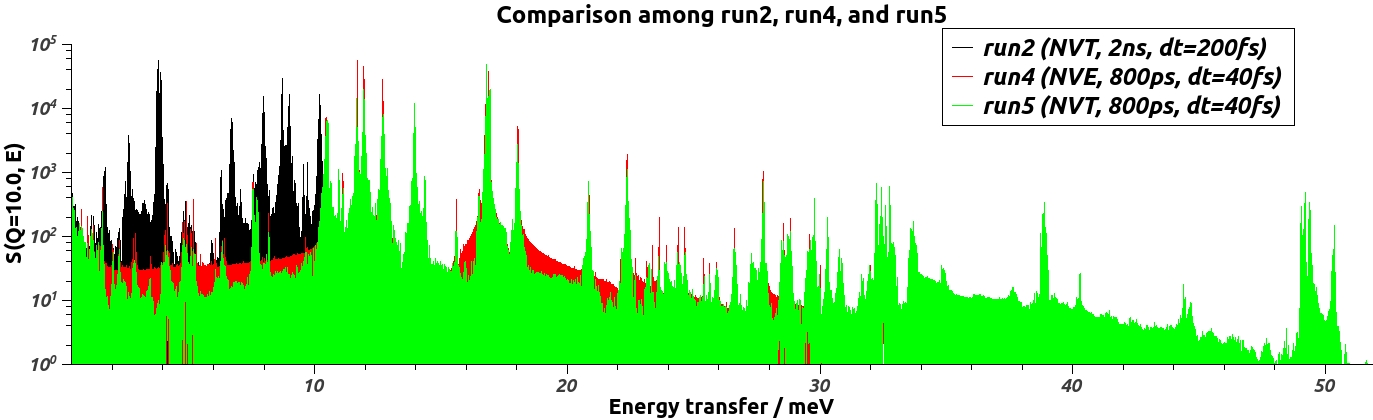
# Origin of spurious peaks

* Job “**K0.07/T\_5/run4**” simulates 800ps in the NVE ensemble, output every 40fs for a total of 20,000 frames inelastic regime in the [0.005,100]meV regime. Surprisingly, we observe that this jobs has suppressed all peaks below 10meV, as compared to jobs run2 and run3. Could this be because we ran in the NVE ensemble and the population of modes with E<10meV was low at the very beginning? Recall that the energy stored in each normal mode should be conserved in the NVE ensemble. This result is worrisome because experiments show three strong peaks in the [3,9]meV range.



*charmm/nmolec\_1/K0.07/T\_5/run4\_Q10.jpeg*

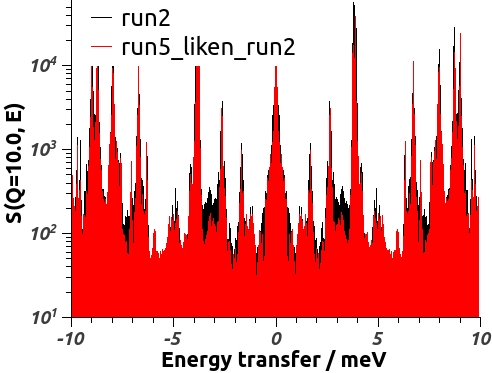
* We will simulate in the NVT ensemble in order to allow energy transfer between modes through the thermal bath. In addition, we use LangevingDamping=5(1/ps), which will dampen oscillations with ν<0.4meV, well below the [1,50]meV range we are interested in. Job “**K0.07/T\_5/run5**” simulates 800ps in the NVT ensemble, output every 40fs for a total of 20,000.
* Comparison between run4 and run5 show diffuse peaks in run5 as compared to run4. This is expected because the heat bath produces energy transfer between modes.



*charmm/nmolec\_1/K0.07/T\_5/run5\_Q10.jpeg*

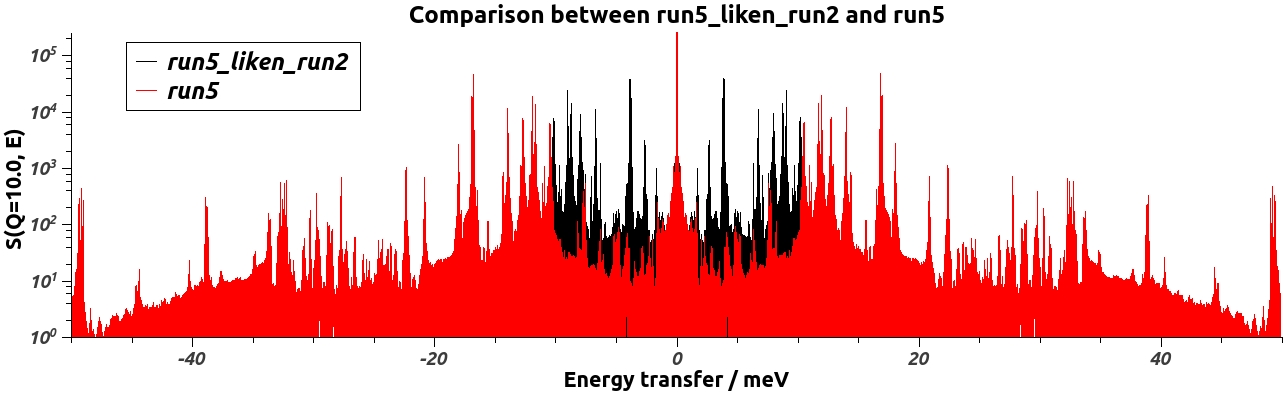
* As in run4, run5 does not have the peaks that run2 showed in the [1,10]meV range! I was expecting these peaks to show up when simulating in the NVT ensemble.
* We extract frames from run5, one frame every five, so that we end up with a trajectory of 800ps with frames every 200fs for a total of 4000 frames. This trajectory should be similar to the first 800ps of run2, since it was produced starting from the same conformation (albeit the heat bath will produce differences). The idea is to compare S(Q,E) from this trajectory to that of run2, see if we obtain the peaks in the [1,10]meV range.

ptraj me8t8.prmtop < **run5\_liken\_run2**.in produces *run5\_liken\_run2.dcd*, which is the aforementioned trajectory.



*charmm/nmolec\_1/K0.07/T\_5/run5\_liken\_run2\_Q10.jpeg*

* Both run2 and the run5\_liken\_run2 produce virtually identical S(Q,E), as shown in the figure above. However, run5 and run5\_liken\_run2 have very different spectrum in the [1,10]meV. Notice that run5 and run5\_liken\_run2 are *the same trajectory*, the differences being that the time between frames in run5 is 40fs for a total of 20,000 frames and the time between frames in run5\_liken\_run2 is 200fs for a total of 4000 frames.



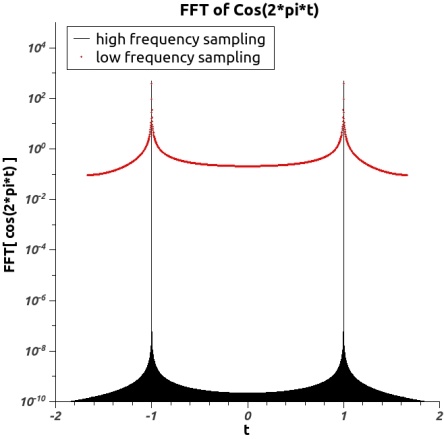
*charmm/nmolec\_1/K0.07/T\_5/run5\_liken\_run2\_Q10.II.jpeg*

* Are the dissimilarities in S(Q,E) between run5 and run5\_liken\_run2 due to the difference in frame-recording rate or due to the difference in the number of frames? Let’s extract the first 4000 frames of *run5\_rms2first.dcd* into a trajectory, ***run5\_160ps****.dcd*. This trajectory has the same frame-saving rate (40fs) than *run5\_rms2first.dcd*, but different number of frames (and therefore different span). Analogously, trajectory *run5\_160ps.dcd* has the same number of frames as *run5\_liken\_run2.dcd*, but different frame-recording rate.

ptraj me8t8.prmtop < run5\_160ps.in produces *run5\_160ps.dcd*.

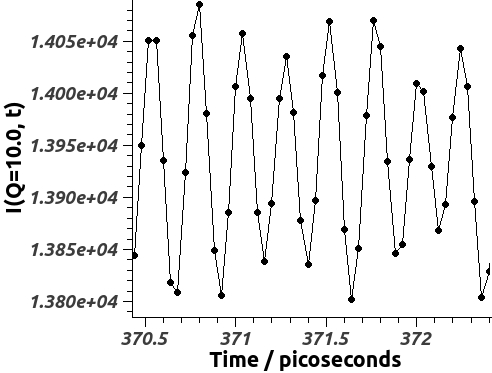
|  |  |
| --- | --- |
| *charmm/nmolec\_1/K0.07/T\_5/run5\_160ps\_Q10.II.jpeg* | *charmm/nmolec\_1/K0.07/T\_5/run5\_160ps\_Q10.jpeg* |

* We observe that run5\_160ps is very similar to the extended trajectory run5\_rms2first (800ps), except for the [-10,10]meV range where the extended trajectory shows a few low intensity peaks while 160ps trajectory. This may be expected since the 800ps trajectory has better resolution in the small energy scale. More important, run5\_160ps is very different to run5\_liken\_run2 in the [-10,10] range. The strong peaks of run5\_liken\_run2 are absent in run5\_160ps. Recall these trajectories have same number of frames but different frame-recording rate. Thus, it is this feature that results in the spectra being that different.
* I did a controlled benchmark by calculating the Fourier transform of *I(t)=Cos(2πt)* sampled at different rates.



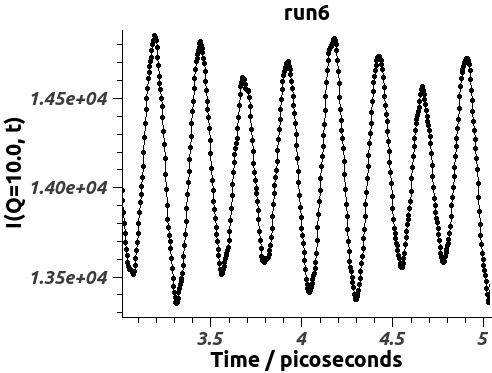
*charmm/nmolec\_1/K0.07/T\_5/fft\_cos.jpeg*

* When the sampling frequency of *I(t)=Cos(2πt)* is high, we obtain the expected result, that is, two peaks with an intensity much higher than the background (black baseline). However, when the sampling frequency is low, the intensity of the peaks as compared to the background diminishes considerably (red line). You can think as extra peaks arising in the [-1,1] range. This effect is precisely what I observed in the run5\_liken\_run2 trajectory, where extra peaks in the [-10,10]meV range are present when compared with run5\_rmsfirst trajectory, which has a frame-recording rate five times higher.
* The conclusion is that in order to obtain a reliable Fourier transform, I(t) has to be “smooth”, that is, one must have many points describing the fastest oscillations of I(t).



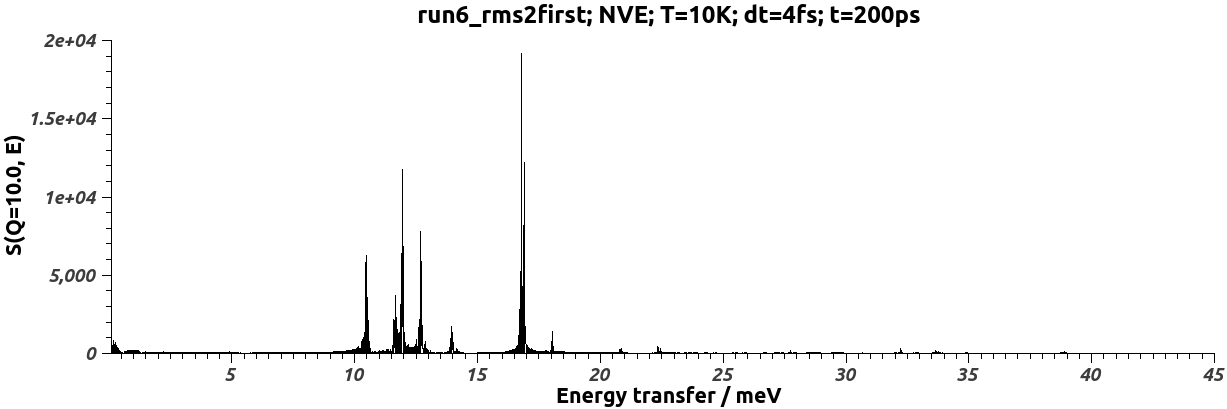
*charmm/nmolec\_1/K0.07/T\_5/run5\_Q10.III.jpeg*

* As we see from the picture above, we output in *run5.dcd* about 6 frames (points) covering the fastest oscillations of the system, which corresponds to 6\*40fs=240fs. It would be better if we would output many more points per fastest oscillation. In fact, 240fs may not be the fastest oscillation in the system, but only the fastest oscillation that can be observed with a 40fs frame-recording rate.
* Job “**K0.07/T\_5/run6**” in the NVT ensemble lasts 200ps with a frame-recording frequency of 4fs, thus yielding 50,000 frames.



*charmm/nmolec\_1/K0.07/T\_5/run6\_Q10.jpeg*

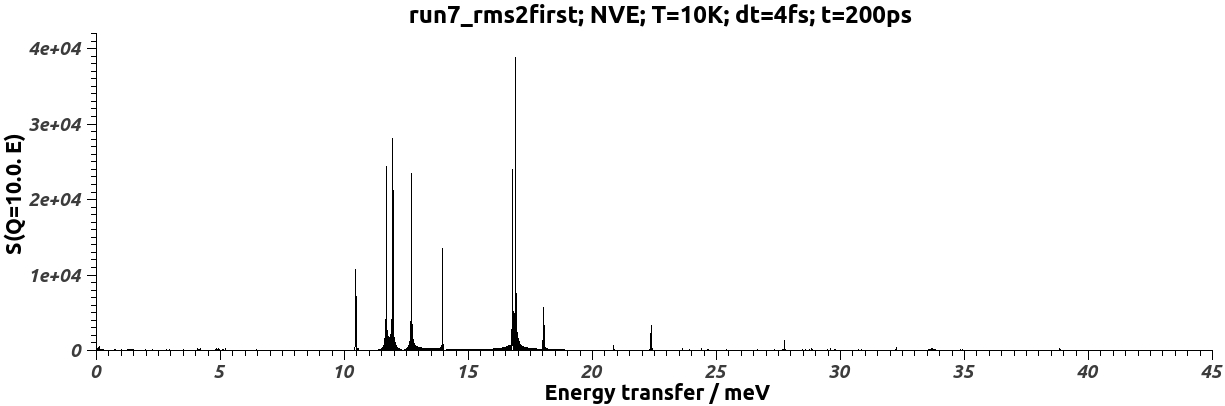
* The detailed in the plot of I(Q=10.0, t) shows a smooth function, with many points describing the fastest oscillation, which is still about 240fs. The Fourier transform should now be reliable.



*charmm/nmolec\_1/K0.07/T\_5/run6\_Q10.II.jpeg*

# Spectrum at low temperatures

* Since we had determined that runs in the NVE ensemble contain less background, we run job “**K0.07/T\_5/run7**” which is like “**K0.07/T\_5/run6**” but in the NVE ensemble. Thus, the jobs lasts 200ps with a frame-recording frequency of 4fs, thus yielding 50,000 frames. The starting coordinates and velocities are from run1.restart.cood(.vel), just like job “**K0.07/T\_5/run4**”. We expect an S(Q,E) with a smaller background than S(Q,E) for “**K0.07/T\_5/run6**”. However, some peaks may have very small intensity or even disappear if the corresponding mode had a low intensity in the starting conformation or even was not present.



*charmm/nmolec\_1/K0.07/T\_5/run7\_Q10.jpeg*

* Compared to figure *run6\_Q10.II.jpeg*, we observe that the peaks of *run7\_Q10.jpeg* are sharper and with higher maximum. This was the expected result of simulating in the NVE ensemble where no dispersion of the energy is present due to a heat bath.
* Convoluting this signal with a Gaussian of width 1meV, approximately the resolution of the experiment, we compare this spectrum to experimental data:

|  |  |
| --- | --- |
| *charmm/nmolec\_1/K0.07/T\_5/run7\_Q10.II.agr(png)* | *expdata/CNCS/s\_Q3-7\_E0-45\_T5.png* |

|  |  |
| --- | --- |
| *charmm/nmolec\_1/K0.07/T\_5/run7\_Q10.III.agr(png)* | *expdata/CNCS/s\_Q3-4\_E0-10\_T5.png* |

# Preparing the crystal simulation

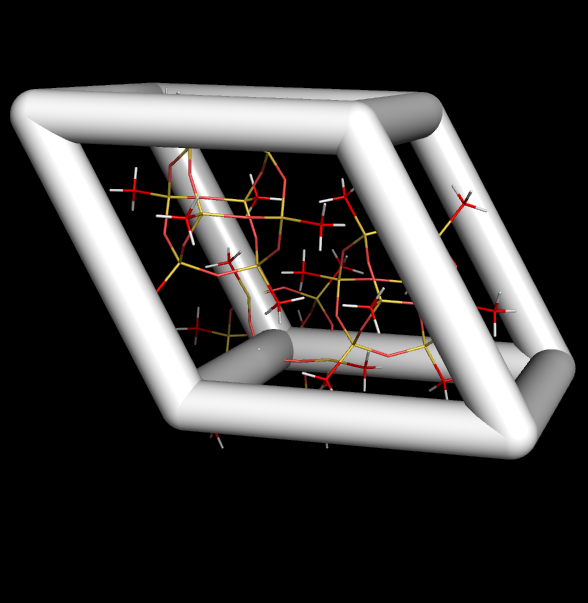
* Abe12.pdf contains MD simulations of crystals for Me-POSS (also fo ibu-POSS, H-POSS, and CP-POSS).
* <http://neutrons.ornl.gov/portal> has a link to the Cambridge Structural Database. I did a “text/numeric” search using “methylsilsesquioxane” as “exact word” text search, obtaining the entry: <http://webcsd.ccdc.cam.ac.uk/display_csd_search_results.php?xml_temp_file=/temp/text_numeric_query_063656400137278156051d2fbf89cb9c.xml>
* I downloaded *OCMSIO03.cif* file (to *$PROJD/preparing/*) which contains the crystallographic data in a format (CIF) that understood by pymol’s SuperSym plugin. Soft link *me8t8.cif* points to this file.
* The atom names in are not standard AMBER atom names. To circumvent this problem, we have to create a lattice unit starting with the asymmetric unit *me8t8.cif*, and then substitute each molecule in the lattice unit with a molecule containing the right atom names. This is achieved by loading *me8t8.pdb* and then aligning this molecule to each of the molecules in the lattice unit.
* A caveat to the above procedure is that the different pymol commands to align two structures fail to do the alignment. To circumvent this problem, we edit *me8t8.cif* and rename the Si atoms as those of *me8t8.pdb*, saving the modified file as preparing/*me8t8.v2.cif*. Below is the table to translate between the Si atom names. This table preserves the topology between the two sets of Si atoms.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PDB - Si | Si1 | Si2 | Si3 | Si4 | Si5 | Si6 | Si7 | Si8 |
| CIF - Si | Si1 | Si2 | Si2K | Si2A | Si2I | Si2B | Si2J | Si1J |

* Unit cell pymol session *me8t8\_unit\_cell.pse* was created by loading *me8t8.v2.cif* and creating the default lattice with the SuperSym plugin. As it turns out, there are redundant molecules created because SuperSym tries all the symmetry operations defined in *me8t8.v2.cif*. Thus, we have to remove the extra molecules until we end up with the two molecules inside the unit cell and another one in one corner of the cell. As it turns out, molecules that superimpose to each other have different topologies in the sense that the numbering of the Si atoms is different (who is to the right of who, who is to the left of who…). Thus, we have to leave only the three molecules that have the same topology of the Si atoms, so that the alignment using *me8t8.pdb* is successful.
* For the alignment, instead of align command we will use pair\_fit command which allows biyective relations between pair of atoms of the two structures, like this:

pair\_fit (first and name Si1),(second and name Si1), (first and name Si2),(second and name Si2), (first and name Si3),(second and name Si3), (first and name Si4),(second and name Si4), (first and name Si5),(second and name Si5), (first and name Si6),(second and name Si6), (first and name Si7),(second and name Si7), (first and name Si8),(second and name Si8)

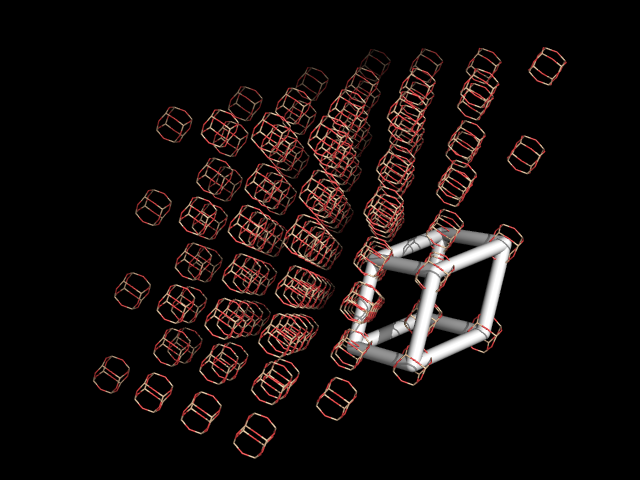
* We save the unit cell containing the original and the supplanted molecules in *me8t8\_unit\_cell.pse*. We remove the original molecules and save the unit cell containing the POSS molecules with the correct AMBER atom names in *me8t8\_unit\_cell.v3.pse* and *me8t8\_unit\_cell.pdb*.



*preparing/me8t8\_unit\_cell.pdb(.****v3****.pse,.v3.png)*

* There are problems doing translation of the molecules in order to generate the neighboring cells. Apparently the molecules have internal ‘states’ that are accessed while doing a translation. There’s more than what one sees in the display window.
* We desist on using pymol to recreate the crystal. Instead, we use **mercury**, the Cambridge Structural Database tool to visualize their entries.

Mercury -> Load *me8t8.v2.cif* -> Calculate -> Packing/Slicing -> Check the “Pack” box -> 3x3x3 -> Save as PDB -> ***mercury\_3.1x3.1x3.1.pdb***



*preparing/mercury\_3.1x3.1x3.1.pdb(png,pse)*

* Note that *mercury\_3.1x3.1x3.1.pdb* cannot be our system for simulation, because it contains the symmetry partners (a=±3, b=±3, c=±3) which cannot be present if periodic boundary conditions are employed. Thus, we rerun mercury with parameters 2.9 instead of 3.0.

Mercury -> Load *me8t8.v2.cif* -> Calculate -> Packing/Slicing -> Check the “Pack” box -> 2.9x2.9x2.9 -> Save as PDB -> ***mercury\_3x3x3.pdb***

|  |  |
| --- | --- |
| View along a-axis    *preparing/mercury\_3x3x3.png(pse)* | View along c-axis    *preparing/mercury\_3x3x3.v2.png(pse)* |

* In the pictures above, we see two views of the superposition of *mercury\_3x3x3.pdb* onto *mercury\_3.1x3.1x3.1.pdb*. The red molecules are the symmetry partners (a=±3, b=±3, c=±3).
* File mercury\_3x3x3.pdb contains 81 molecules. Each molecule has 8 redundant atoms (53 through 60) that must be removed. Furthermore, we have to translate the names of the atoms used in the Cambridge Structural Database to the names used by AMBER.

me8t8.py preparing --kwargs="job=translate PDB from CSD to AMBER,inpdb=mercury\_3x3x3.pdb,outpdb=me8t8\_3x3x3.pdb"

## Generate topology library file

* In order to create *.prmtop* and *.mdcrd* initial files, we first need to create a *.lib* file. We do it by loading *me8t8\_Qneutral.mol2* file and then saving it as a *.lib* file. In *$PROJD/preparing/*:

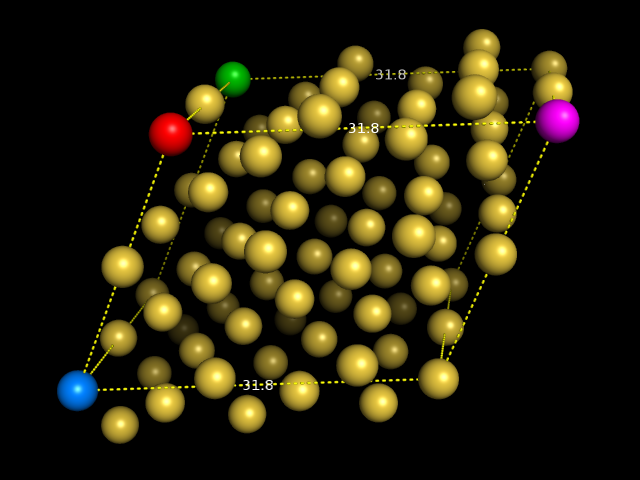
tleap -f me8t8.lib.tleap.in creates ***me8t8.lib***.

* The force-field that Stan Anderson sent me does not load in tleap, the reason being the incorrect syntax in the torsion potentials. I corrected these and saved them as file ***me8t8.dat***.
* We create a modified force-field file ***me8t8.frcmod*** that inserts a new data type for the silicon atom and loads the library *me8t8.lib* and parameter set *me8t8.dat*.
* Now we are ready to create the *.prmtop* and *.mdcrd* initial files

tleap -f me8t8\_3x3x3.tleap.in > me8t8\_3x3x3.tleap.out creates ***me8t8\_3x3x3.prmtop*** and ***me8t8\_3x3x3.mdcrd***.

# Equilibrate the 3x3x3 crystal

* Directory *$PROJD/3x3x3/*.



*3x3x3/me8t8\_3x3x3.png(pse)*

* In the figures, spheres represent atoms Si1. Red sphere is closest Si1 to the origin, red to magenta is the X-axis (1,0,0). Red to green is the Z-axis (0,0,1). Red to blue is (cos(240), sin(240), 0). Knowing this, we can set the cell basis vectors in the namd configuration file as

cellBasisVector1 37.54761 0.0 0.0

cellBasisVector2 -18.773805 -32.517184111 0.0 ;#cos(240)\*36.7 sin(240)\*\*36.7

cellBasisVector3 0.0 0.0 39.34200

## NAMD fails to read the topology file

* This is a known reading error of NAMD when accessing AMBER topology files. To correct this, we have to manually edit *me8t8\_3x3x3.prmtop* and remove sections %FLAG SCEE\_SCALE\_FACTOR and %FLAG SCNB\_SCALE\_FACTOR.
* After removal of these flags, NAMD stills reports “FATAL ERROR: Failed to read AMBER parm file!”. Thus, I will see if I can run a simulation in AMBER with this *me8t8\_3x3x3.prmtop* file.

## Creating topology for single molecule

* Since NAMD fails to read the topology file for the crystal, we generate a topology file for a single molecule and check whether NAMD can read this file.

tleap -f me8t8\_singlemolecule\_tleap.in > me8t8\_singlemolecule\_tleap.out

creates files *singlemolecule.mdcrd* and *singlemolecule.prmtop*.

* NAMD fails to read the topology file *singlemolecule.prmtop*. Thus, we compare *singlemolecule.prmtop* and *me8t8.prmtop*. The comparison shows the following differences:
  + The following extra flags in *singlemolecule.prmtop*: %FLAG ATOMIC\_NUMBER, %FLAG SCEE\_SCALE\_FACTOR, and %FLAG SCNB\_SCALE\_FACTOR
  + The following flags have no contents, and should contain “ 0.00000000E+00”: %FLAG HBOND\_ACOEF, %FLAG HBOND\_BCOEF, and %FLAG HBCUT
  + The following extra flag in me8t8.prmtop: %FLAG BOX\_DIMENSIONS
* Removal of the extra flags in *singlemolecule.prmtop* and inclusion of “ 0.00000000E+00” in the aforementioned flags fixes the reading errors of NAMD.
* We perform the same modifications in file *me8t8\_3x3x3.prmtop*.

## Copying force-field from me8t8.prmtop

* It turns out that the contents of the flags describing the force-field in *me8t8\_3x3x3.prmtop* (BOND\_FORCE\_CONSTANT, BOND\_EQUIL\_VALUE, ANGLE\_FORCE\_CONSTANT, ANGLE\_EQUIL\_VALUE, DIHEDRAL\_FORCE\_CONSTANT, LENNARD\_JONES\_ACOEF, and LENNARD\_JONES\_BCOEF) have very different values that the contents of the corresponding flags for *me8t8.prmtop*. This indicates that the *siparms.cpy* that Stan sent me is not the parameter set that was used to generate *me8t8.prmtop*.
* Z