

# 纳米液滴在不同表面的浸润行为

姜壬雯 石金泽 吴汶津

2021 年 1 月 3 日

## 1 分工情况

大部分工作由小组成员共同完成，工作量最大的MD建模前期，我们先各自尝试了不同的方法；在找到合适的方法后，又一起完善了可行代码的相关计算命令。在这个过程中每个人都完整地走过了一遍文献阅读、力场确定、in文件编写、初始原子数据设置等流程。后期分析过程中我们分工完成了平衡态分析、结构分析、动力学分析和接触角测量的工作，吴汶津同学完成了平衡态分析以及结构分析、动力学分析中的部分内容，姜壬雯同学完成了结构分析、动力学分析的部分内容，石金泽同学完成了动力学分析和接触角测量的部分工作。报告最后附有三位成员的感想。

## 2 简介

### 2.1 接触角

接触角是指当系统达到平衡时，在气、液、固三相交界处，气液界面与固液界面之间的夹角。接触角的大小是由在气液固三相交界处，三种界面张力的相对大小所决定的。平衡时这三种力与接触角之间的关系由杨式方程(1)描述.

$$\gamma_{SG} = \gamma_{SL} + \gamma_{LG} \cos \theta \quad (1)$$

式中 $\gamma_{SG}$ 是固气界面张力， $\gamma_{SL}$ 是固液界面张力， $\gamma_{LG}$ 是液气界面张力.

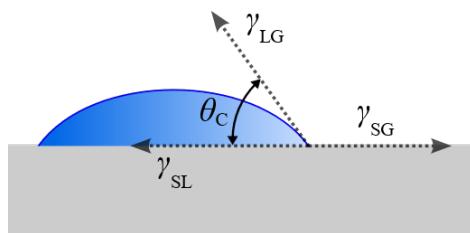


Figure.1 Contact angle of a liquid droplet wetted to a rigid solid surface

### 2.2 浸润

润湿过程是指，在一块固体表面滴上少许液体，在未滴液体之前固体是和气体接触的，滴上液滴后取代了部分固气界面，产生了新的液固界面。润湿过程可以分为三类：粘湿、浸湿（浸润）和铺展。当液体在固体上的接触角 $\leq 180^\circ$ 时，发生粘湿；若接触角 $\leq 90^\circ$ ，则发生浸润； $\approx 0^\circ$ 时，发生铺展。

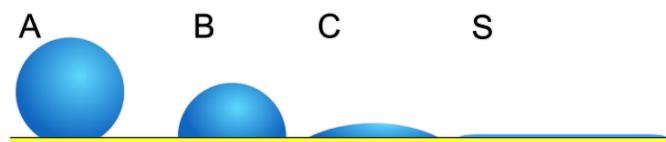


Figure.2 Wetting of different fluids: A shows a fluid with very little wetting, while C shows a fluid with more wetting. A has a large contact angle, and C has a small contact angle.

### 3 MD模拟细节

#### 3.1 模型建立

水:  $30 * 20 * 30 \text{ \AA}$  vmd建模 540<sup>1</sup> 个水分子

石墨烯:  $100 * 20 \text{ \AA}$  vmd 建模 (0、1、2、3、4、5、6)层

金属铜: fcc结构  $100 * 20 * 36.1^2 \text{ \AA}$  lammps生成

方形水滴与最顶层石墨烯间距  $0.25 \text{ \AA}$

最底层石墨烯与金属铜间距  $2.8 \text{ \AA}$

最终体系在x方向为  $(0, 100) \text{ \AA}$  , z方向为  $(-36.1, 300) \text{ \AA}$  , 水滴上方有足够大的真空环境, 确保打开周期性边界条件之后, 该体系在x, z方向不会受镜像体系的影响, 而y方向会形成无限长的液柱(采用这种模型的具体原因将在接触角的测量部分进行解释).

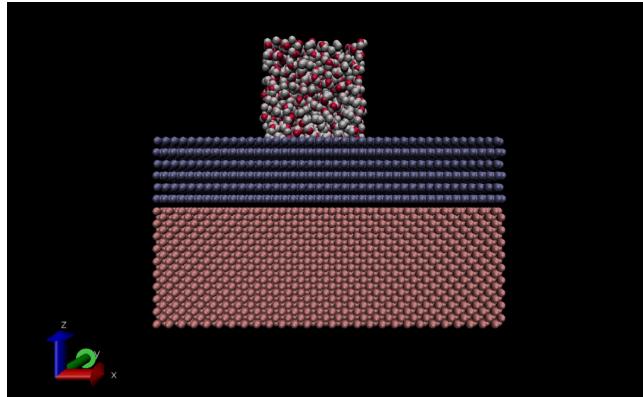


Figure.3 The initial structure.

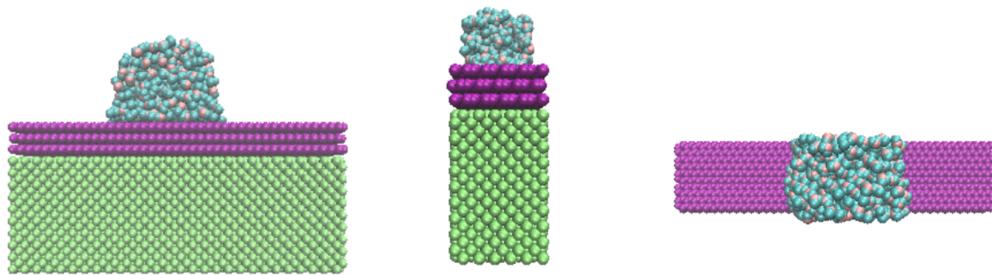


Figure.4 Elevation view, top view and side view of the model after a period of simulation. Blue for oxygen, pink for hydrogen, purple for carbon and green for copper.

#### 3.2 力场

水之间的相互作用采用TIP3P模型, 这当中包含O(-0.834e)与H(+0.417e)上的局部电荷之间的库仑

1.关于水分子个数的讨论: 我们也尝试过将水分子体系扩大4倍 (水  $60*20*60 \text{ \AA}$ , 石墨烯 $200*20 \text{ \AA}$ ), 相应为2160个水分子; 但是机时和内存有限, 运行完55000步后水滴甚至还具有比较明显的类长方体形, 而且dump文件已经达到G的量级, 不便于后续的分析. 后续可以做的尝试是, restart延长模拟时间、优化分析程序的数据结构等等.

2. z 方向有10个晶格常数尺寸的Cu, 保证能够模拟出体态铜.

相互作用，以及各原子之间的LJ相互作用；各相互作用系数在Table.1中列出，截断取为10 Å<sup>3</sup> .

Pair	$\epsilon$ (kcal/mol)	$\sigma$ (Å)
O-O	0.1020	3.1880
O-H	0.0836	1.7753
H-H	0.0460	0.4000
O-C/H-C	0.1143	3.2751
O-Cu	0.1700	3.1900
C-C	0.0860	3.400

Table.1 LJ parameters of different pairs. <https://youtu.be/VGBzGGgnTI>

### 3.3 模拟

用郎之万恒温方法<sup>4</sup>为水滴控温，控制在300K；先在nve系综中以limit<sup>5</sup>的模式运行5000步，再在一般模式下的nve系综下行50000步，步长为2fs。我们模拟的总体思路是通过改变金属铜表面石墨烯的层数，由0层石墨烯到6层石墨烯，来对比观察纳米水滴在不同表面上的浸润行为。

## 4 分析

### 4.1 平衡分析

由Figure.5可知，具有不同石墨烯层数的结构均在20ps内达到了动态平衡。且最后平衡的温度都在320K附近的区间，与我们设定的温度基本相符。

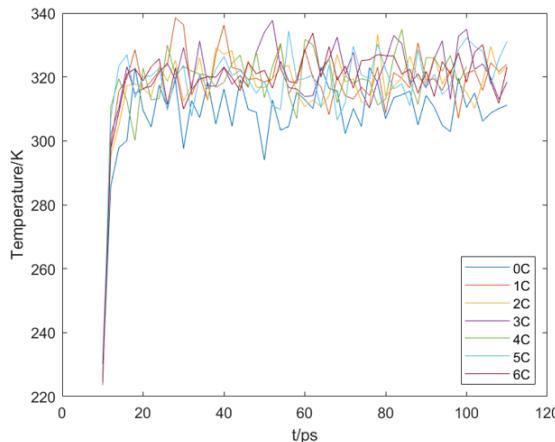


Figure.5 The structures with different number of graphene layers reached dynamic equilibrium within 20ps.

### 4.2 结构分析

#### 4.2.1 水分子径向分布函数g(r)

计算RDF时，打开y方向的周期性边界条件，关闭x、z方向的周期性边界条件；由Figure.6可知，

3.关于截断取为10 Å 的讨论：在doi:10.1038/nmat3228这篇文献中，对浸润透明性进行了最初的分子动力学模拟和讨论，文献中的截断取为20 Å，其理由是避免截断过小而人为导致非浸润效应；然而在我们的模拟中，采用20 Å的截断尚未能解决overflow的error，因此我们查阅了相关类似的文献，发现后继的大部分文献都以10 Å 为截断，所以在我们的模拟中也采用了10 Å 的截断。

4.这种温度调节方法是通过重新标定速度场，使系统的动能保持恒定来控制温度的。郎之万恒温方法用到了郎之万运动方程， $ma = -\xi v + f(v) + f'$ ，其中 $\xi$ 为摩擦系数， $f'$ 为一个附加摩擦力，计算中通过调节速度来保持温度，使动能与LAMMPS设定的温度相匹配。

5.nve/limit 在每个步长用恒定的NVE更新原子的位置和速度，一个原子在一个时间步内所能移动的最大距离会有一个限制，在我们这种可能导致原子高度重叠的结构中，以这种模式开始模拟是一个比较有效的方式。

gOO只有第一个峰相对显著，这说明水滴的结构短程有序、长程无序，所以模拟中的水应为液态。

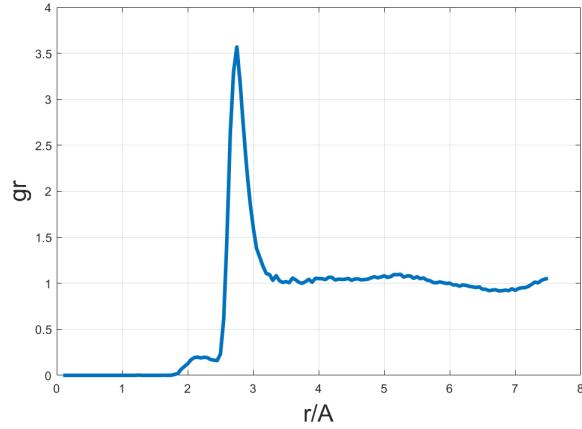


Figure.6 Radial distribution function of oxygen atoms.

#### 4.2.2 Q6

对于不同的石墨烯层数，Q6均随时间波动上升<sup>6</sup>，最终落在0.27到0.28的范围内，这说明水滴的结构始终具有随机性。

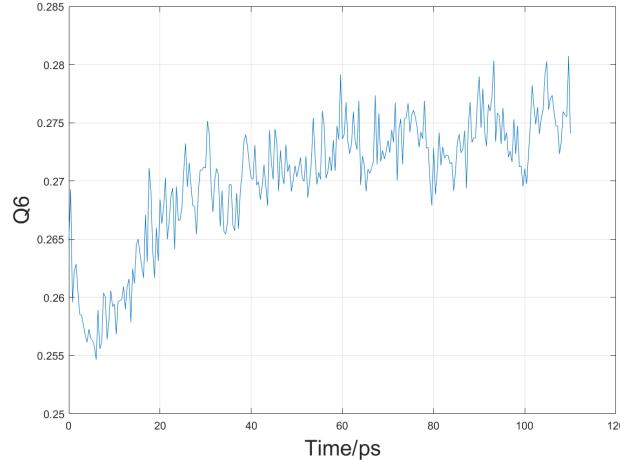


Figure.7  $Q_6$  of the water droplets when there are 6 layers of graphene.

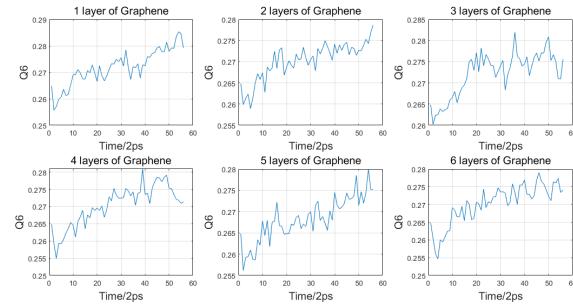


Figure.8:  $Q_6$  of water droplets when there are different number of graphene's layers.  $Q_6$  ascends with fluctuation, but eventually falls within the range of 0.27到0.28, indicating that the structure of water droplets is random all the time.

	fcc	hcp	random
$Q_6$	0.574	0.485	0.289

Table.2  $Q_6$  in different lattice structure.

### 4.2.3 水分子z方向密度分布

Saleh Bagheri等人在2017年研究了不同石墨烯表面水滴在z方向的密度分布<sup>[6]</sup>，得到了如下的主要结果。随着石墨烯结构逐渐稀疏（如下左图），呈现出类似径向密度分布函数的z向密度分布的峰值逐渐降低（如下右图），表明有序程度的降低。

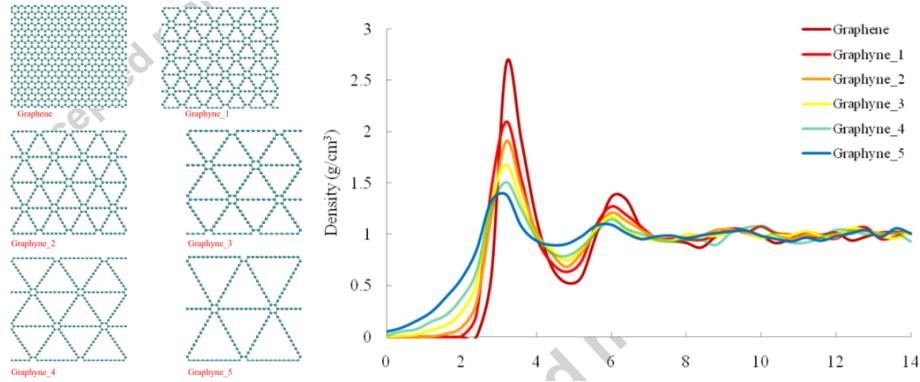


Figure.9 Saleh Bagheri,2017

受到这篇论文的启发，我们尝试探究了不同石墨烯层数对于z向密度分布的影响，在MD模拟体系平衡后计算z向密度的时间平均得到：

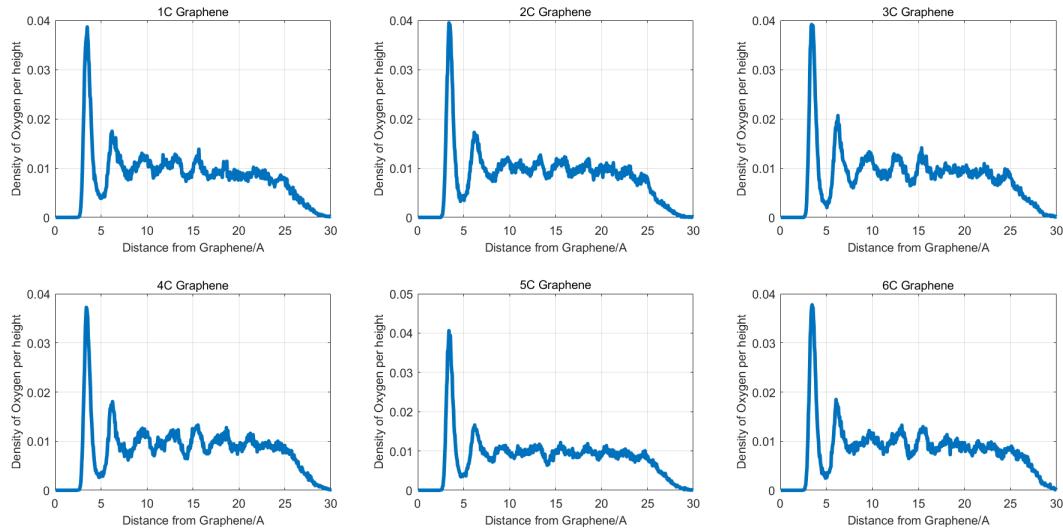


Figure.10 Density in z direction basically invariant with the layer's number of Graphene.

我们发现改变石墨烯层数对水滴z向密度分布的影响非常小，第一个峰值基本一致，所以可以得到一个粗糙的结论：改变石墨烯层数不会影响纳米水水滴的结构。

6.图中Q6均先短暂地下降而后波动上升，最终达到动态平衡。猜测相应的物理过程可能为：用vmd建模生成的初始长方体水滴中，水分子的坐标是随机的，此时水滴尚未由于与基底（Cu或者C）的相互作用引起其宏观、微观结构的改变；随时间推移，水分子与基底的相互作用让水滴的形状和水滴的表面积发生变化，水滴由长方体趋近类圆柱体。在计算Q6时仅打开y方向的周期性边界条件而关闭x, z方向的边界条件；也易推断处在水滴表面的水分子，其Q6会明显不同于处在水滴内部的水分子的Q6（理应偏小）；而模拟中由于条件所限，水滴较小，处在表面的水分子占有不可忽视的比重；在水滴表面积发生变化时，由表面水分子个数变化引起的整体平均Q6的改变；后期水滴达到平衡状态，表面积相对稳定，因而Q6的值也相对稳定。但总体的Q6不会超过0.29，说明水滴始终具有随机结构。

## 4.3 动力学过程分析

### 4.3.1 MSD分析

对比不同相的MSD，可知水滴的确处于液态。

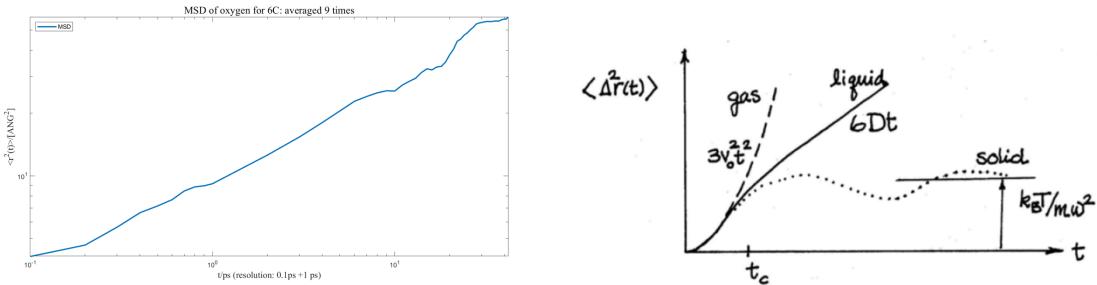


Figure.11、12 MSD curve<sup>7</sup> in this simulation when there are 6 layers of graphene; MSD curve in different phase.

### 4.3.2 扩散情况z向分布

在这一部分里，我们记录了水分子中氧原子的初始高度，并计算其后续平衡后的MSD，利用如下表达式计算了氧原子的扩散情况，定义扩散系数 $D_i$ ：

$$D_i = \frac{1}{t_N - t_0} \sum_{t_0}^{t_N} \langle r_i(t_N) - r_i(t_0) \rangle^2 \quad (2)$$

$r_i(t)$ 是氧原子*i*在*t*时刻的位置。

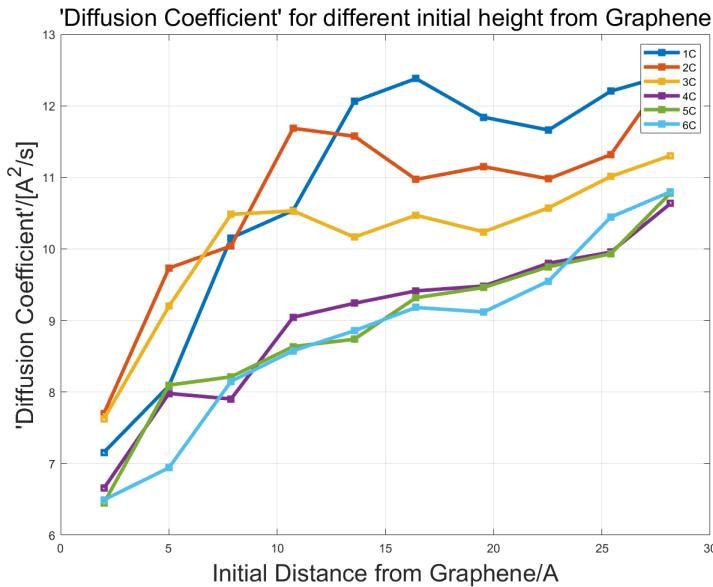


Figure.13 “Diffusion Coefficient” for different initial height from graphene

通过得到的计算结果我们得到上图。在石墨烯层数比较少时(eg,1,2,3)，水分子的扩散情况随着高度增加先单调递增，然后曲线出现了凹陷，且随着层数增加凹陷开始出现的高度逐渐提前。当石墨烯层数增加到4至6层时，凹陷消失，曲线转成单调递增。计算的结果显示了在石墨烯层数较少时，水滴中部的水分子扩散较两边更加受限，也就意味着该处的可能有一个小势阱限制了水分子的扩散，随着石墨烯层数的增加，铜对体系的影响越来越弱，势阱消失，扩散程度在z+方向单调递增。

7.对于不同平均次数以及时间分辨率对MSD曲线的影响的详细讨论见附录。

### 4.3.3 动能Ek的z向分布

我们进而分析了动能在z方向上的分布，与扩散情况相似，在石墨烯层数较少时，中部位置水分子动能出现了比较明显的凹陷。我们知道lammps中的温度计算是基于各个原子动能的平均，所以这也能够体现出水滴中温度分布的不均匀性。

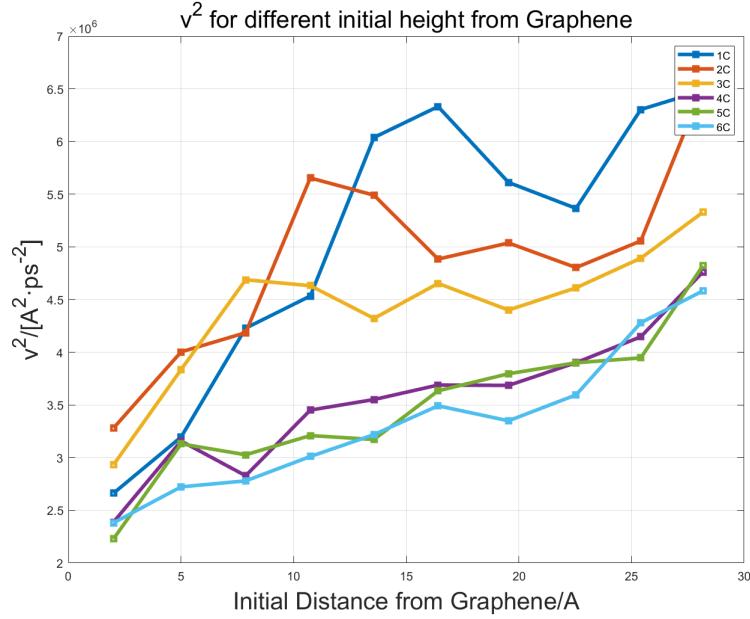


Figure.14  $v^2$  for different initial height from graphene.

### 4.4 接触角分析

使用分子动力学模拟水滴在石墨烯表面的接触角时，通常水分子的数目很小，约在2000—10000 的范围内，对应水滴的尺寸在10nm 以内，此时水滴与石墨烯接触界面的线张力无法忽略，杨氏方程需要进行修正，修正后的形式和原始的形式 $\theta$ 为微观尺度下的接触角； $\theta_\infty$ 为宏观尺度下的接触角； $\tau$ 为水滴与石墨烯接触界面的线张力； $r_B$ 为水滴在基底的半径； $\gamma_{SV}$ ,  $\gamma_{SL}$ ,  $\gamma_{LV}$ 分别为固-气、固-液、液-气表面张力。

$$\cos\theta = \cos\theta_\infty - \frac{\tau}{\gamma_{LV}} \frac{1}{r_B} \quad (3)$$

$$\cos\theta_\infty = (\gamma_{SV} - \gamma_{SL})/\gamma_{LV} \quad (4)$$

线张力的影响会使得微纳尺度下的水滴接触角与宏观尺度下的水滴接触角有较大的差别，因此需要对分子模拟的结果进行合理的外推才能得到宏观的接触角。为了得到正确的外推结果，需要对线张力有明确的认识，但是目前对于线张力的研究还停留在初步探索阶段，对于其性质和产生机理并没有成熟一致的认识。因此本文采用无限长液柱的模拟体系来避免线性张力对模拟结果的影响<sup>[2]</sup>。

我们将数据导入tracker，测量如图所示的A、B、C三个点的坐标，利用余弦公式计算了水滴圆柱四个角在平衡后的时间平均：

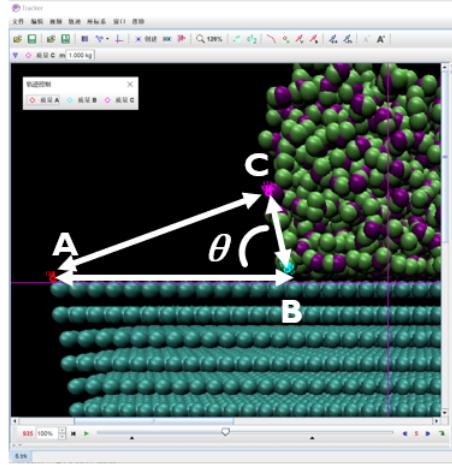


Figure.15 Contact angle measurement in Tracker.

$$\|AB\| = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \quad (5)$$

$$\|BC\| = \sqrt{(x_C - x_B)^2 + (y_C - y_B)^2} \quad (6)$$

$$\|CD\| = \sqrt{(x_C - x_D)^2 + (y_C - y_D)^2} \quad (7)$$

$$\theta = \sum_{t_1}^{t_N} \arccos\left(\frac{\|AB\|^2 + \|BC\|^2 - \|AC\|^2}{2\|AB\|\|BC\|}\right)/(t_N - t_1) \quad (8)$$

查阅文献易知，水滴在金属铜表面表现出浸润性，在石墨表面表现出非浸润性。通过模拟，得到如Figure.16的结果，可以发现随石墨烯层数增加，固体表面的水滴逐渐表现出非浸润性，这与相关论文中的结果是相符的，在一定程度上说明了模拟结果的正确性。Rafee J等人首次通过分子动力学模拟对纳米水滴在金属铜表面的浸润透明性<sup>8</sup>进行了探讨，其模拟结果如Figure.17. 由图中数据知，其在添加第一层石墨烯时，接触角<sup>9</sup>增大了约3度。而在此次模拟中，添加第一层石墨烯时接触角变化达到了8.1度，与文献中的结果差异较大。其主要原因可能为以下三点：1. 纳米水滴偏小，查阅所得的各种文献中都至少用了1000个水分子，而此次模拟中仅用了540个<sup>10</sup>水分子；2. 由于该篇文献给出的某些参数有误，此次模拟参考了其他文献的力场参数，所用力场及力场参数不同，会对模拟结果造成一定影响；3. Tracker手动标记测量存在误差。但也应注意到，由Figure.17，当所添加的石墨烯层数为1、2、3时，接触角均小于90度，水滴均表现出浸润性；当石墨烯层数为4、5、6时，接触角大于90度，水滴表现出了非浸润性。在此次模拟中，添加石墨烯层数为1、2、3时，接触角均大于1.57(也即 $\pi/2$ )，水滴呈浸润性；当石墨烯层数为4、6时，接触角均小于1.57，水滴呈现出非浸润性。两者的结果一致。因而我们更倾向于增大水滴体系，以多次重复模拟的平均测量结果作为最终结果来进一步验证浸润透明性。

8. 浸润透明性是指，石墨烯涂层不会明显地破坏那些处于基底表面、与基底之间的相互作用以范德华力为主的水滴的润湿行为。关于浸润透明性物理本质的讨论详见doi:10.1038/nmat3228.

9. 该篇论文中的接触角是指那个最通用的被定义为接触角的角，详见Figure.1；而我们这里的接触角其实是它的补角，见Figure.15，定义方式不同但不会引起本质影响。

10. 关于所用水分子个数的讨论见脚注1.

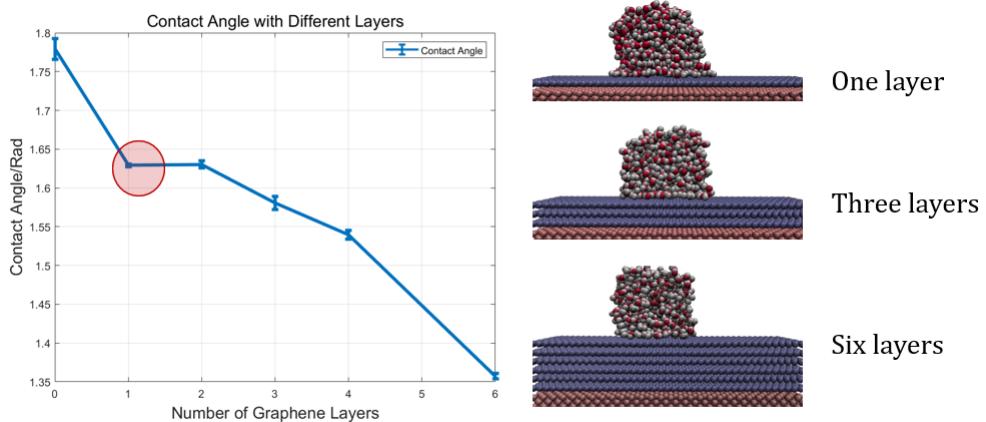


Figure.16 Contact angle with different layers.

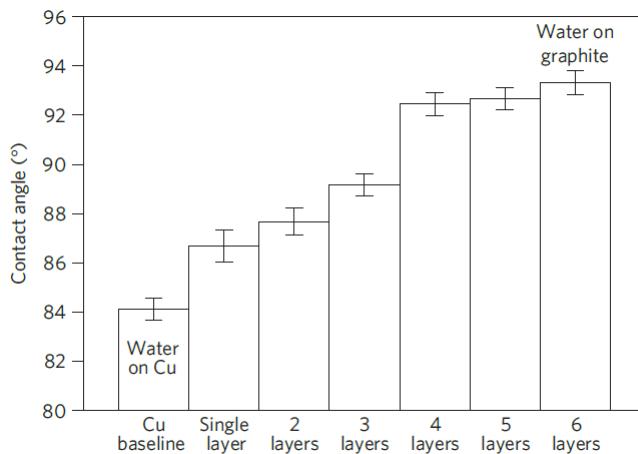


Figure.17 Predicted contact-angle variation with an increasing number of graphene layers on the Cu substrate. doi:10.1038/nmat3228

## 5 结论及展望

浸润透明性能在不破坏表面润湿效应的情况下独立调节表面的其他特性，对导电、保形和不渗透表面涂层的设计具有重要意义。但在我们的模拟中并没有得到足够理想的模拟结果，未来可以着手以下三点进行改进：1. 寻找类似体系的相关参数做进一步模拟；2. 改进数据读取的代码结构，用C语言代替Matlab和python，以适应加大体系的水分子数后文件读取、运算效率低下等问题；3. 利用程序计算接触角的方法代替手动标记的方法。

## 6 参考文献

- [1]Rafiee J, Mi X, Gullapalli H, et al. Wetting transparency of graphene. *Nat Mater.* 2012;11(3):217-222. Published 2012 Jan 22. doi:10.1038/nmat3228
- [2]Shi Chao, Lin Chen-Sen, Chen Shuo, Zhu Jun. Molecular dynamics simulation of characteristic water molecular arrangement on graphene surface and wetting transparency of graphene. *Acta Physica Sinica*, 2019, 68(8): 086801. doi: 10.7498/aps.68.20182307
- [3]Yanbin Wu and N. R. Aluru. Graphitic Carbon–Water Nonbonded Interaction Parameters. *The Journal of Physical Chemistry B*, 2013, 117 (29): 8802-8813.doi: 10.1021/jp402051t

[4]Joseph Eugene Andrews, Shayandev Sinha, Peter W. Chunga, Siddhartha Das. Wetting dynamics of a water nanodrop on graphene. Phys. Chem. Chem. Phys., 2016,18: 23482-23493.doi:10.1039 C6CP01936F

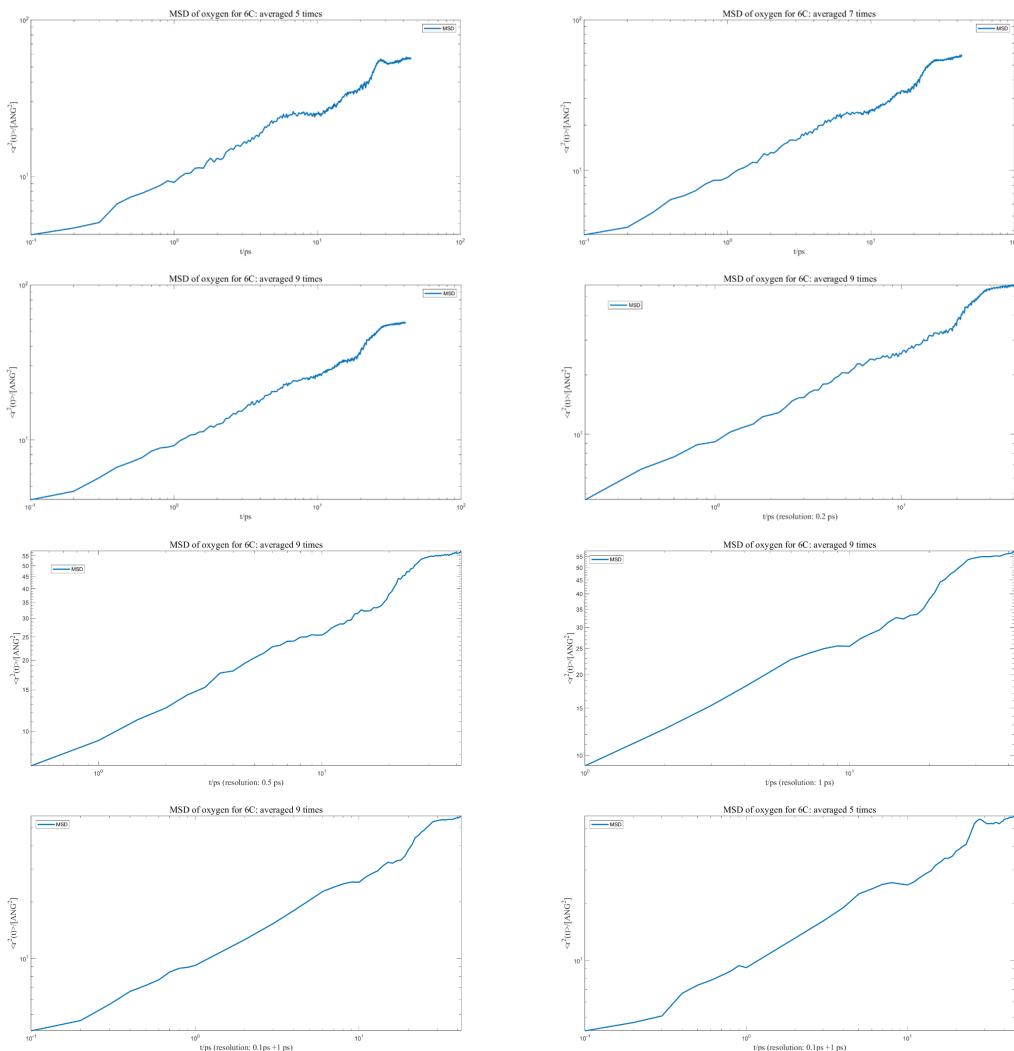
[5]Romain Perriot, Xiang Gu, You Lin, Vasily V. Zhakhovsky, and Ivan I. Oleynik. Screened environment-dependent reactive empirical bond-order potential for atomistic simulations of carbon materials. Phys. Rev. B., 2013, 88:12. doi:10.1103/PhysRevB.88.064101

[6]Saleh Bagheri, Abolghasem Shameli, Mehdi Darvishi, Ghasem Fakhrpour, Molecular investigation of water adsorption on graphene and graphyne surfaces, Physica E: Low-dimensional Systems and Nanostructures, Volume 90,2017, Pages 123-130, ISSN 1386-9477.

## 7 附件

### 7.1 MSD

关于不同平均次数以及时间分辨率对MSD曲线的影响的讨论:



Sup.Fig.1: MSD curve of oxygen for 6 layers of graphene: (averaged times, time resolution/ps): (5,0.1), (7,0.1), (9,0.1), (9,0.2), (9,0.5), (9,1.0); The last two: averaged 9 times with 0.1ps time resolution within 1ps, and 1ps time resolution after 1ps, averaged 5 times with 0.1ps time resolution within 1ps, and 1ps time.

## 7.2 以6层碳为例的in.file

```
1 units real
2 dimension 3
3 boundary p p p
4 atom_style full
5
6 lattice fcc 3.61
7 region box block 0 28 0 5.75 -10 -0.8
8 create_box 4 box bond/types 2 angle/types 2 dihedral/types 1 extra/bond/per/atom 2 extra/
9     angle/per/atom 3 extra/dihedral/per/atom 8
10 create_atoms 4 box
11 mass 4 64
12
13 # read topology
14 read_data 6C.data add append
15
16 # Force-field parameters
17 # C=1, H=2, O=3
18
19 pair_style lj/cut/coul/cut 10.0
20 bond_style harmonic
21 angle_style hybrid charmm harmonic
22 dihedral_style charmm
23
24 special_bonds lj/coul 0.0 0.0 0.0
25
26 pair_coeff 1 1 0.086 3.4
27 bond_coeff 2 305.0 1.4
28 angle_coeff 1 charmm 40.000 120.00 35.00 2.41620
29 dihedral_coeff 1 3.100 2 180 0.0
30
31 # Cu
32 pair_coeff 3 4 0.17 3.19
33 pair_coeff 4 4 0 0
34 pair_coeff 1*2 4 0 0
35
36 # TIP3 water H=2, O=3
37 pair_coeff 3 3 0.1020 3.1880 # modified 0.1521 3.1507
38 pair_coeff 2 2 0.0460 0.4000
39 pair_coeff 2 3 0.0836 1.7753
40 bond_coeff 1 450 0.9572
41 angle_coeff 2 harmonic 55 104.52
42
43 pair_coeff 1 2*3 0.1143 3.2751
44
45 neighbor 2.0 bin
46 neigh_modify delay 0 every 1 check yes exclude type 1 3
47
48 # Output
49 thermo 1000
50 thermo_style multi
51 #dump 22 all dcd 1000 project.dcd
52
53 timestep 2.0
54
55 group water type 2 3
56 group oxygen type 3
57 group cnt type 1
58 group copper type 4
```

```

59 fix 2 copper setforce 0.0 0.0 0.0
61
63 # Equilibration
65 velocity water create 300.0 4928459 rot yes dist gaussian
67
69 #dump trj all atom 50 all.lammpstrj
#dump trj2 water atom 50 water.lammpstrj
71
73 compute msd oxygen msd com yes average yes
75 fix msd oxygen ave/time 5 10 50 c_msd[4] file wat.msd
77
79 #compute rdf water rdf 500 2 2 2 3 3 3 #gOO gOH gHH
#fix rdf water ave/time 10 3000 55000 c_rdf[*] file wat.rdf mode vector
81
83 fix 11 water langevin 300 300 1000 212894
85 fix 12 water nve/limit 0.1
#fix 13 water deform 1 x final -17.5 17.5 final -17.5 17.5 z final -15 35 units box
87
89 run 5000
91 unfix 12
#unfix 13
93
95 # Simulation
97 fix 21 water nve
99 run 50000
101
103 write-restart restart.all

```

### 7.3 6层碳data file

```

1 LAMMPS data file. CGCMM style. atom_style full generated by VMD/TopoTools v1.7 on Sun Dec
 13 11:34:36 +0800 2020
2 6540 atoms
3 8154 bonds
4 14088 angles
5 25962 dihedrals
6 0 impropers
7 3 atom types
8 2 bond types
9 2 angle types
10 1 dihedral types
11 0 improper types
12 0 100 xlo xhi
13 0 20 ylo yhi
14 0 300 zlo zhi
15
16 # Pair Coeffs
17 #
18 # 1 CA
19 # 2 HT
20 # 3 OT
21
22 # Bond Coeffs
23 #
24 # 1
25 # 2 CA-CA

```

```

27 # Angle Coeffs
#
29 # 1 CA-CA-CA
# 2 unknown
31
# Dihedral Coeffs
33 #
# 1 CA-CA-CA-CA
35
# Improper Coeffs
37 #
# 1 CA-CA-CA-CA
39
Masses
41
1 12.010700 # CA
43 2 1.007940 # HT
45 3 15.999400 # OT
47
Atoms # full
49
1 1 1 0.000000 0.000000 0.000000 # CA GRA
51 2 2 1 0.000000 -1.228024 0.709000 0.000000 # CA GRA
53 3 3 1 0.000000 -1.228024 2.127000 0.000000 # CA GRA
55 .....
57
Bonds
59
1 2 1 2
2 2 1 6
61 3 2 2 3
63 .....
65
Angles
67
1 1 2 1 6
69 2 1 1 2 3
71 3 1 2 3 4
73 .....
75
Dihedrals
77
1 1 6 1 2 3
2 1 2 1 6 5
79 3 1 2 1 6 7
81 .....

```

## 7.4 Matlab and Python 分析代码

### 7.4.1 gr

```

1 clc;clear all
2 fpn = fopen('6.lammpstrj', 'rt');
3 l = [0,1,2,3,4,5,6,7,8,9];

```

```

4 h = waitbar(0, 'Reading Data ...');
5 datasets = [];
6 i = 1;
7 n = 1;
8 while feof(fpn) ~= 1
9     waitbar(n/(56*1629),h)
10    file = fgetl(fpn);
11    if i ~= 1 && i ~= 2 && i ~= 3 && i ~= 4 && i ~= 5 && i ~= 6 && i ~= 7 && i ~= 8 && i ~= 9
12        tt = str2num(file);
13        if tt(2) == 3
14            datasets = [datasets;str2num(file)];
15        end
16    end
17    if mod(i,1629)==0
18        i = 0;
19    end
20    i = i+1;
21    n = n +1;
22 end

24 len = size(datasets);
25 for i = 1:round(len(1)/500)
26     plot3(datasets((i-1)*500+1:i*500,3),datasets((i-1)*500+1:i*500,4),datasets((i-1)*500+1:i*500,5),'r','MarkerSize',50)
27     axis([0 20 0 20 0 20]);
28     grid on
29     pause(0.1)
30 end

32 % FDR Calculation
33 dr = 0.05;
34 gr = zeros(56,10000);
35 L = 1;
36 for t = 1:56
37     for i = 540*(t-1)+1:540*t
38         x0 = datasets(i,3);
39         y0 = datasets(i,4);
40         z0 = datasets(i,5);
41         for j = 540*(t-1)+1:540*t
42             x1 = datasets(j,3);
43             y1 = coordinate_changeY(datasets(j,4),y0);
44             z1 = datasets(j,5);
45             r = ((x1-x0)^2+(y1-y0)^2+(z1-z0)^2)^0.5;
46             gr(t,floor(r/dr)+1) = gr(t,floor(r/dr)+1) + 1/(floor(r/dr)*dr)^2;
47         end
48     end
49     time_step = size(datasets);
50     time_step = time_step(1)/540;
51     gr(t,:) = (gr(t,:)/540/dr/4/pi)/(540/18000);
52 end

54 % g(r) Modulation
55 l = dr:dr:150*dr;
56 for i = 1:1:56
57     plot(l,gr(i,1:length(l)), 'Color',[0 0 i/56], 'linewidth',1)
58     axis([0 8 0 4]);
59     pause(1)
60 end
61 function x1 = coordinate_changeY(r1,r0)

```

```

64 L = 20;
65 if abs(r1-r0)>=L-abs(r1-r0) && r1 >= r0
66 x1 = r1-L;
67 elseif abs(r1-r0)>=L-abs(r1-r0) && r1 < r0
68 x1 = L+r1;
69 else
70 x1 = r1;
71 end
72 end

```

### 7.4.2 Q6

```

clc; clear all
1
2 h = waitbar(0, 'Reading Data ...');
3 fpn = fopen('water.lammpstrj', 'rt');
4 l = [0,1,2,3,4,5,6,7,8,9];
5 datasets = [];
6 i = 1;
7 n = 1;
8 while feof(fpn) ~= 1
9     waitbar(n/(276*1629),h)
10    file = fgetl(fpn);
11    if i ~= 1 && i ~=2 && i~=3 && i ~=4 && i~=5 && i ~=6 && i ~=7 && i ~=8 && i ~=9
12        tt = str2num(file);
13        if tt(2) == 3
14            datasets = [datasets,str2num(file)];
15        end
16    end
17    if mod(i,1629)==0
18        i = 0;
19    end
20    i = i+1;
21    n = n +1;
22 end
23
24
25 l = 6; %0.1912
26 %l = 6; 0.4883
27 %l = 8; 0.3515
28 Q1_time= [];
29 for t = 1:276
30     if t~=276
31         waitbar(t/209,h, 'Calculating ...')
32     else
33         close(h)
34         msgbox('Finished');
35     end
36
37
38 Q1_atoms =0;
39 for i = 540*(t-1)+1:540*t
40     min_atoms = 1000*ones(1,12);
41     cos_theta_list = ones(1,12);
42     phi_list = 1000*ones(1,12);
43     x0 = datasets(i,3);
44     y0 = datasets(i,4);
45     z0 = datasets(i,5);
46     for j = 540*(t-1)+1:540*t

```

```

48     if j ~= i
49         x1 = datasets(j,3);
50         y1 = coordinate_changeY(datasets(j,4),y0);
51         z1 = datasets(j,5);
52         r = (x1-x0)^2+(y1-y0)^2+(z1-z0)^2;
53         for k =1:12
54             if r <= min_atoms(1,k)
55                 if k ==1
56                     min_atoms = [r,min_atoms];
57                     min_atoms = min_atoms(1:12);

58                     cos_theta_item = (z1-z0)/r^0.5;
59                     cos_theta_list = [cos_theta_item, cos_theta_list];
60                     cos_theta_list =cos_theta_list(1:12);

62                     phi_item = acos((x1-x0)/((x1-x0)^2+(y1-y0)^2)^0.5);
63                     phi_list = [phi_item, phi_list];
64                     phi_list = phi_list(1:12);
65                     break
66                 else
67                     min_atoms = [min_atoms(1:k-1),r,min_atoms(k:end)];
68                     min_atoms = min_atoms(1:12);

70                     cos_theta_item = (z1-z0)/r^0.5;
71                     cos_theta_list = [cos_theta_list(1:k-1),cos_theta_item,
72                                     cos_theta_list(k:end)];
73                     cos_theta_list =cos_theta_list(1:12);

74                     phi_item = acos((x1-x0)/((x1-x0)^2+(y1-y0)^2)^0.5);
75                     phi_list = [phi_list(1:k-1),phi_item, phi_list(k:end)];
76                     phi_list = phi_list(1:12);
77                     break
78                 end
79             end
80         end
81     end
82
83     Ql = 0;
84     for m = -l:1:l
85         Ylm_average = 0;
86         for k=1:12
87             Ylm_average = Ylm(l,m,cos_theta_list(k),phi_list(k));
88         end
89         Ql = Ql + (abs(Ylm_average/12))^2;
90     end
91     Ql_atoms =Ql_atoms +(4*pi/(2*l+1)*Ql)^0.5;
92
93     Ql_time = [Ql_time Ql_atoms/540];
94 end
95
96
97
98 function SP_value =Ylm(l,m,cos_theta,phi)
99 PLM = legendre(l,cos_theta);
100 SP_value =(-1)^m*((2*l+1)/4/pi*factorial(1-abs(m))/factorial(1+abs(m)))^0.5*PLM(abs(m)+1)*(
101     cos(m*phi)+1i*sin(m*phi));
102 end
103
104 function x1 = coordinate_changeY(r1,r0)
105 L = 20;

```

```

106 if abs(r1-r0)>=L-abs(r1-r0) && r1 >= r0
107 x1 = r1-L;
108 elseif abs(r1-r0)>=L-abs(r1-r0) && r1 < r0
109 x1 = L+r1;
110 else
111 x1 = r1;
112 end
113 end

```

### 7.4.3 MSD

```

1 import numpy as np
2
3
4 f = open('water_new.lammpstrj')
5 f_line = f.readlines()
6 f_num = len(f_line)
7 minID = 11305
8 maxID = 12924
9 atomnum = 1620
10 molnum = int(atomnum/3)
11 s1 = [i for i in range(f_num) if 'ITEM: ATOMS id type x y z' in f_line[i]]
12 s2 = [(s1[x]+1, s1[x+1]-8) for x in range(len(s1)-1)]
13 nframe = [f_line[i[0]:i[1]] for i in s2]
14 nframe.append(f_line[(s1[-1]+1):])
15 nframe_1 = [[nframe[i][j].split() for j in range(atomnum)] for i in range(len(nframe))]
16 nframe_2 = [sorted([[int(nframe_1[t][id][0])-minID//3+1,int(nframe_1[t][id][1]),float(
17     nframe_1[t][id][2]),float(nframe_1[t][id][3]),float(nframe_1[t][id][4])]
18         for id in range(atomnum) if int(nframe_1[t][id][1]) == 3]) for t in
19             range(len(nframe_1))]
20 # nframe_2 = [sorted([[int(nframe_1[t][id][0]),int(nframe_1[t][id][1]),float(nframe_1[t][id][2]),
21     float(nframe_1[t][id][3]),float(nframe_1[t][id][4])]
22         for id in range(atomnum) if int(nframe_1[t][id][1]) == 3]) for t in
23             range(len(nframe_1))]
24 nframe_3 = np.array(nframe_2)
25 # print(len(nframe_3),nframe_3[-1])
26
27 startpoint = 0
28 endpoint = 920
29 # endpoint = 200
30 frameIntveral = 20
31 num_frame = (endpoint - startpoint)//frameIntveral
32 avertime = 5
33 averInter = 20
34 statlst = [startpoint+averInter*x for x in range(avertime)]
35 endlst = [endpoint+averInter*x for x in range(avertime)]
36 # print(statlst, endlst)
37
38 # MSD = np.zeros(num_frame+1)
39
40 # for x in range(avertime):
41 #     msd = []
42 #     for t in range(statlst[x], endlst[x]+1, frameIntveral):
43 #         msd_t = 0
44 #         for ID in range(1, molnum+1):
45 #             # print(ID)

```

```

43 #           displacement_t = np.linalg.norm(nframe_3[t, ID-1, 2:]-nframe_3[statlst[x], ID
44 #               -1, 2:])
45 #           # print('disp', displacement_t)
46 #           msd_t += displacement_t**2
47 #           # print('before',msd_t)
48 #           msd_t /= molnum
49 #           msd.append(msd_t)
50 #           # print('after',msd_t)
51 #           MSD += np.array(msd)
52 #           if x == avertime-1:
53 #               MSD /= avertime
54 #
55 frameIntveral1 = 2 # 1ps/0.1ps = 10
56 frameIntveral2 = 20
57 num_frame_new = 11 + (endpoint - (startpoint+20))//frameIntveral2
58 MSD = np.zeros(num_frame_new)
59
60
61 for x in range(avertime):
62     msd = []
63     for t in range(statlst[x], statlst[x]+20, frameIntveral1):
64         msd_t = 0
65         for ID in range(1, molnum+1):
66             # print(ID)
67             displacement_t = np.linalg.norm(nframe_3[t, ID-1, 2:]-nframe_3[statlst[x], ID
68                 -1, 2:])
69             # print('disp', displacement_t)
70             msd_t += displacement_t**2
71             # print('before',msd_t)
72             msd_t /= molnum
73             msd.append(msd_t)
74         for t in range(statlst[x]+20, endlst[x]+1, frameIntveral2):
75             msd_t = 0
76             for ID in range(1, molnum+1):
77                 # print(ID)
78                 displacement_t = np.linalg.norm(nframe_3[t, ID-1, 2:]-nframe_3[statlst[x], ID
79                     -1, 2:])
80                 # print('disp', displacement_t)
81                 msd_t += displacement_t**2
82                 # print('before',msd_t)
83                 msd_t /= molnum
84                 msd.append(msd_t)
85                 # print('after',msd_t)
86             MSD += np.array(msd)
87             if x == avertime-1:
88                 MSD /= avertime
89 # print("MSD", len(MSD),MSD)
90 # target_list = np.array([(1,2),(2,3),(3,4)])
91 with open ("MSD.txt", 'w') as file:
92     for i, line in enumerate (MSD):
93         file.write(str(line).lstrip('[').rstrip(']')+ '\n')
94 file.close()

```

#### 7.4.4 z方向扩散系数关系

```

1 clc;clear all
2 h = waitbar(0, 'Reading Data ...');

```

```

3 fpn = fopen('altitudeMSD\4.lammpstrj', 'rt');
4 l = [0,1,2,3,4,5,6,7,8,9];
5 datasets = [];
6 i = 1;
7 n = 1;
8 while feof(fpn) ~= 1
9     waitbar(n/(276*1629),h)
10    file = fgetl(fpn);
11    if i ~= 1 && i ~= 2 && i ~= 3 && i ~= 4 && i ~= 5 && i ~= 6 && i ~= 7 && i ~= 8 && i ~= 9
12        tt = str2num(file);
13        if tt(2) == 3
14            datasets = [datasets;str2num(file)];
15        end
16    end
17    if mod(i,1629)==0
18        i = 0;
19    end
20    i = i+1;
21    n = n +1;
22 end
23
24 for n = 1:276
25     datasets((n-1)*540+1:n*540) = sortrows(datasets((n-1)*540+1:n*540),1);
26 end
27
28
29 z_matrix = zeros(276,540);
30 r_matrix = zeros(276,540);
31 for id = 1:540
32     if id~=540
33         waitbar(id/540,h,'Calculating... ')
34     else
35         close(h)
36         msgbox('Finished');
37     end
38     x0 = datasets(id,3);
39     y0 = datasets(id,4);
40     z0 = datasets(id,5);
41     z_matrix(1,id) = z0;
42     for t = 2:276
43         x1 = datasets(540*(t-1)+id,3);
44         y1 = coordinate_changeY(datasets(540*(t-1)+id,4),y0);
45         z1 = datasets(540*(t-1)+id,5);
46         r = ((x1-x0)^2+(y1-y0)^2+(z1-z0)^2)^0.5;
47         z_matrix(t,id) = z1;
48         r_matrix(t,id) = r + r_matrix(t-1,id);
49         x0 = x1;
50         y0 = y1;
51         z0 = z1;
52     end
53 end
54
55 function SP_value = Ylm(l,m,cos_theta,phi)
56 PLM = legendre(l,cos_theta);
57 SP_value = (-1)^m*((2*l+1)/4/pi*factorial(l-abs(m))/factorial(l+abs(m)))^0.5*PLM(abs(m)+1)*(
58     cos(m*phi)+1i*sin(m*phi));
59 end
60
61 function x1 = coordinate_changeY(r1,r0)
62 L = 20;

```

```

63 if abs(r1-r0)>=L-abs(r1-r0) && r1 >= r0
64 x1 = r1-L;
65 elseif abs(r1-r0)>=L-abs(r1-r0) && r1 < r0
66 x1 = L+r1;
67 else
68 x1 = r1;
69 end
end

```

#### 7.4.5 z向密度分布函数

```

1 clc;clear all
2 h = waitbar(0,'Reading Data ...');
3 fpn = fopen('altitudeMSD\4.lammpstrj','rt');
4 l = [0,1,2,3,4,5,6,7,8,9];
5 datasets = [];
6 i = 1;
7 n = 1;
8 while feof(fpn) ~= 1
9     waitbar(n/(276*1629),h)
10    file = fgetl(fpn);
11    if i ~= 1 && i ~=2 && i ~=3 && i ~=4 && i ~=5 && i ~=6 && i ~=7 && i ~=8 && i ~=9
12        tt = str2num(file);
13        if tt(2) == 3
14            datasets = [datasets;str2num(file)];
15        end
16    end
17    if mod(i,1629)==0
18        i = 0;
19    end
20    i = i+1;
21    n = n +1;
22 end
23
24 for n = 1:276
25     datasets((n-1)*540+1:n*540) = sortrows(datasets((n-1)*540+1:n*540),5);
26 end
27
28 density_z = zeros(1,length(0:0.05:300));
29 dz = 0.05;
30 for t = 5:276
31     if t~=276
32         waitbar(t/271,h,'Calculating ...')
33     else
34         close(h)
35         msgbox('Finished');
36     end
37     for i = 540*(t-1)+1:540*t
38         z = datasets(i,5);
39         density_z(floor(z/dz)+1) = density_z(floor(z/dz)+1) + 1;
40     end
41 end

```

## 7.5 成员感想

### 7.5.1 姜壬雯

我们组的三位同学一直是齐头并进的，从目标是写出一个不报错能运行的in文件开始，每周大家都会一起写bug，一起debug，一起爆肝，一起一次次仿佛看到了进展，又一次次一起陷入绝望，直到最后一起看到了这个课题能够做出有点像样子的曙光。从期中考试结束的那一阵子其实我们就开始着手这个课题了，但是由于能力所限，我们还是在in文件的尝试上花费了过多的时间，以至于在前两周我们才第一次有了能够成功运行的不离谱的模拟。虽然最后报告里仅能展现这么不完美也很真实的一点点，但是我感觉我个人在这个过程中还是学到了很多。不仅仅是关于MD本身关于Lammps的使用，我认为更多更重要的是我对于我的两位伙伴有了新的认识。我惊讶于汶津的细腻、脚踏实地和认真。她总是愿意去查阅各种各样的文献，坚持做每一步都要有理有据；在我时常习得性无助打算放弃的时候，她还是坚持，这个问题我要弄明白，后来也往往是汶津给我们带来了突破性的进展；我印象最深的一句话是，“为什么会想到要放弃呢，我觉得也就是时间久一点，没有什么关系”。我想，如果没有汶津，这个课题是不会有什么结果的。我好喜欢这个积极主动又脚踏实地的女孩子。金泽是我的老搭档了，然而我还是被金泽优秀的物理直觉、强大的编程和数据分析能力、还有pre可以信手拈来各种操作震撼到了。最后这篇报告中很多的图都是我们在最后两周赶进度的时候，金泽画出来的，我想这种速度和强度是我在长期内都难以望其项背的。最主要的是，我一次次从金泽凌晨两三点发在群文件的图中，感受到了这个男孩子对于一个团队的担当和对卓越的追求。

这大概是我上大学以来最充实的也最快乐的一次teamwork了。

——相比二位好像也就打了打杂，提了提意见，没啥特殊战斗力的姜壬雯

### 7.5.2 石金泽

妈呀真的是年纪大了，泪点低到不行，下午打开report，大哥一封小作文直接戳我心窝。没心没肺的我现在才写下对二位的感谢。我觉得不说结果怎么样，跟你们一起共事真的超级幸运！我自己也学到了很多你们身上珍贵的品质。汶津的踏踏实实、认真和耐心，常常让我自愧不如，大哥的不甘平庸、敢于挑战、对问题追本溯源的态度，更让我时而羞愧不已。我从一开始有点功利的态度，慢慢被大家的热情和坚持打动，我常常为了最后的目的想退路，现在想想这种态度实在是配不上你们的付出和努力。虽然我常常深夜发图，但那只是我睡得比较晚啦！我起的也比你们晚呀。大家都是很好很好很好，很善良很善良的人，每次project我都觉得很快乐很开心，谢谢你们用纯净洗涤、治愈了我一些脏脏的心态，也让我更深刻地认识到了对待科研工作的态度。有机会一起粗去玩!!! 谢谢大噶！

### 7.5.3 吴汶津

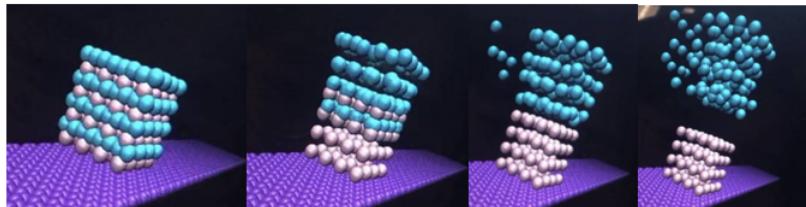
能和大家一起完成这个project，我真是太幸运了！当看到姜姜写的话，我好震惊！因为我一直觉得自己没有做很多，像是一个很蠢的吉祥物，偶尔运气比较好才能给大家提供一点点帮助。大家在讨论和计算一些物理量的时候，我感觉我的脑子像是刚刚进化出来，我真是好大一个废物。我会说出那句话，是因为我觉得有两位大佬在，怎么会做不出来呢嘿嘿。姜姜一直在说自己没有做什么，但我觉得！是因为姜姜什么都在做！建模型写in文件算参数！都在做！根本不是打杂好吗！而且每次我脑袋糊成一团的时候，都是姜姜告诉我应该做什么该怎么做。谢谢姜姜的陪伴和鼓励，才让我有勇气能够做下去。石老师真的好强好强好强好强好强！比我聪明的人还比我努力，凌晨三点我睡得像猪石老师还在努力学习，我只想跪下来膜拜。谢谢石老师能让我抱大腿，还请我们吃泡芙喝奶茶，石老师物院第一帅!!!! 然后我想说的是，两位都是强大且温柔的人，我很少能在同龄人的身上感受到。谢谢你们每次在我干了蠢事之后，都只会给我说没事没事；谢谢你们对我的鼓励，让我有勇气表达自己的观点；谢谢你们愿意倾听我的想法，无论我的想法有多愚蠢。我原来以为我可能始终都只会是个局外人，但我发现我好像有在逐渐融入大家。我真的好开心。谢谢你们，我想成为和你们一样的人。

## 7.6 失败集锦

Titbits: What have we met in our try...?

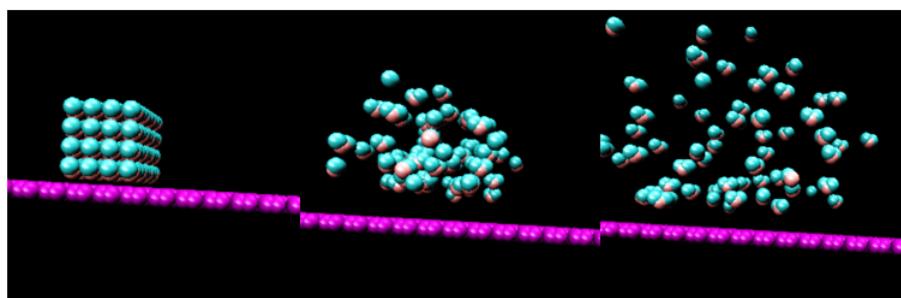
### 7.6.1 “裂解”

It turns out that the graphene can split water molecules ...?



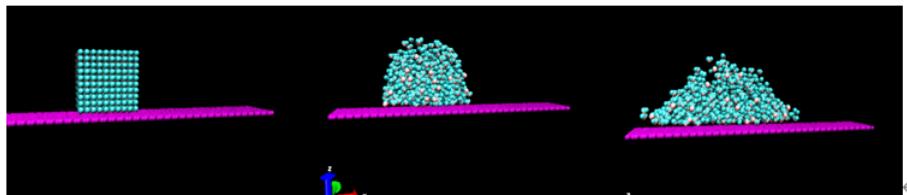
### 7.6.2 “炸开”

Something seems like anti-gravity...



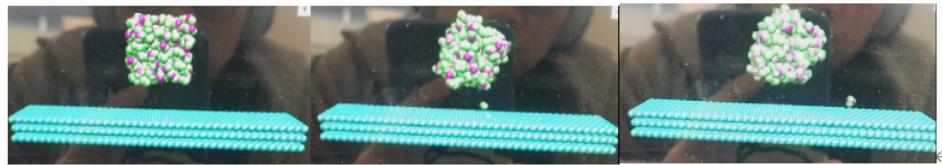
### 7.6.3 “烂泥”

Something behaves like a pool of mud... Oh, and the waterfall ALL THE TIME, which I don't want to show here...because it always makes me feel like I'm doing a hydrodynamics simulation instead of a molecular dynamics simulation!



### 7.6.4 “悬浮”

Always suspended ???



### 7.6.5 “起飞”

Now, Go Ahead, Rocket O!!!

