

Report III

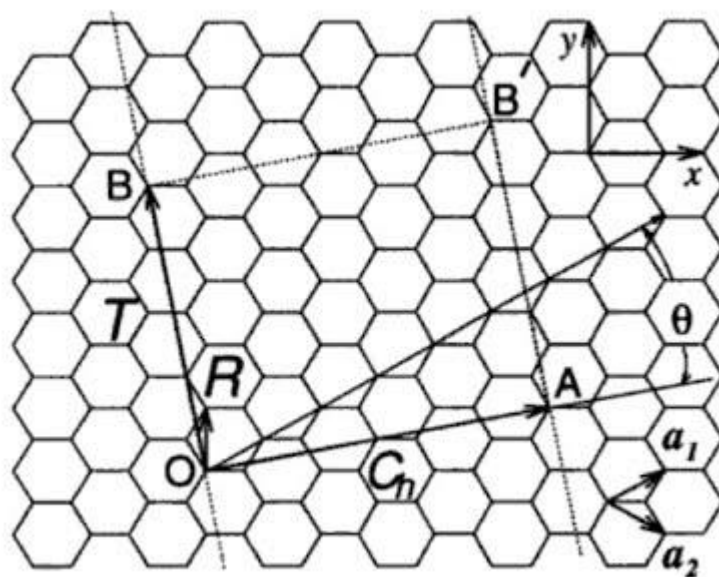
王嘉卿

1. 利用 ASE 生成包含碳纳米管 (GNT) 中各碳原子位置的文件。在 ase.structure 中有 nanotube 函数，直接调用即可创建 GNT。调用格式：

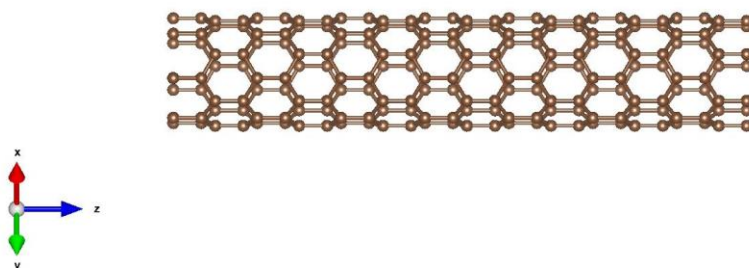
```
nanotube(n, m, length=1, bond=1.42, symbol='C', verbose=False)
```

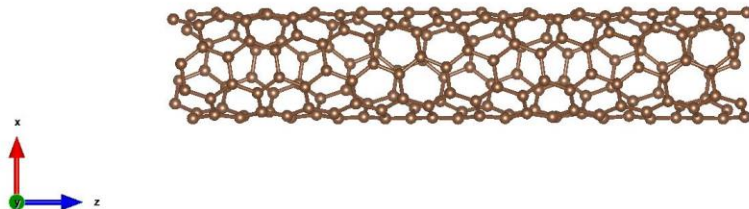
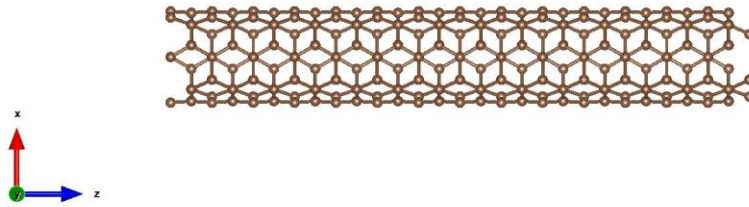
根据 GNT 的标准定义，n,m 分别为 Chiral Vector 中各基矢前的系数，即

$C_h = na_1 + ma_2$ ，Chiral Vector 的模决定了 GNT 的直径；垂直于 C_h 的为 Translation Vector，其长度代表 GNT 原胞长度；Chiral Vector 与 Translation Vector 就决定了 GNT 原胞。因此只需要确定 n,m 的值即可确定 GNT。（函数中 length 沿 GNT 方向上的原胞数。）



下面利用 nanotube 函数分别生成了 zigzag 型 GNT, armchair 型 GNT, 及任意 chiral angle 的 GNT，并利用免费可视化软件 VESTA 作图：





Python 代码:

```
from ase.structure import nanotube
cnt1 = nanotube(7,0,length=7,verbose=True)
cnt2 = nanotube(4,4,length=14,verbose=True)
cnt3 = nanotube(3,5,verbose=True)
write('cnt1.xyz',cnt1)
write('cnt2.xyz',cnt2)
write('cnt3.xyz',cnt3)
```

但在实际运行过程时出现:

TypeError: 'float' object cannot be interpreted as an integer

这是因为函数中 `absn` 变量未转换成整型, 修改源文件使其为整型即可运行。

2.利用 python 优化计算热导率 project。在 python 中创建了一个类 `Run`, 其功能主要是自动运行多个 `lammps script` 文件; 打印各个文件开头的注释信息; 并将每次运行 `lammps` 的日志文件重命名为与输入文件相关的文件名, 而不再是 `log.lammps`, 便于储存和查找。以下是实现上述功能的代码:

(因为初次接触 python, 很多功能和命令都不了解, 所以只是写了一个比较简单的程序)

```
import subprocess
```

```

class Run(object):

    def __init__(self, file):
        self.file = file

    def pre(self):
        file_handle = open(self.file)
        print(file_handle.readline())

    def run(self):
        cmd_line = 'lmp_serial < %s' % self.file
        p = subprocess.Popen(cmd_line, shell=True)
        print(cmd_line)
        flag = p.wait()
        if flag == 0: # 成功运行 lammmps 程序后打印
            print('Sucessfully Run %s' % self.file)

    def post(self):
        log_handle = open('log.lammmps')
        log_content = log_handle.read()
        out_name = 'log_%s' % self.file # 更改后日志文件格式为 log_文件名
        out_handle = open(out_name, 'w')
        out_handle.write(log_content)
        out_handle.close()

# 所需要运行的文件为 pytest 与 in.thermostat
file_name = ['pytest1', 'in.thermostat']
for i in file_name:
    runtest = Run(i)
    runtest.pre()
    runtest.run()
    runtest.post()

```