

Au Nanowires

Brian Sarracino-Aguilera

Define Project

Project option 1



Create a bulk gold model, carve out nanowires from the bulk along different crystal directions (e.g. $\langle 111 \rangle$, $\langle 100 \rangle$, $\langle 110 \rangle$ etc.

1. Compare the energetics of these nanowires after energy minimization.
2. Test the elastic properties of these nanowires and try to understand the phenomenon.

Creating the Bulk

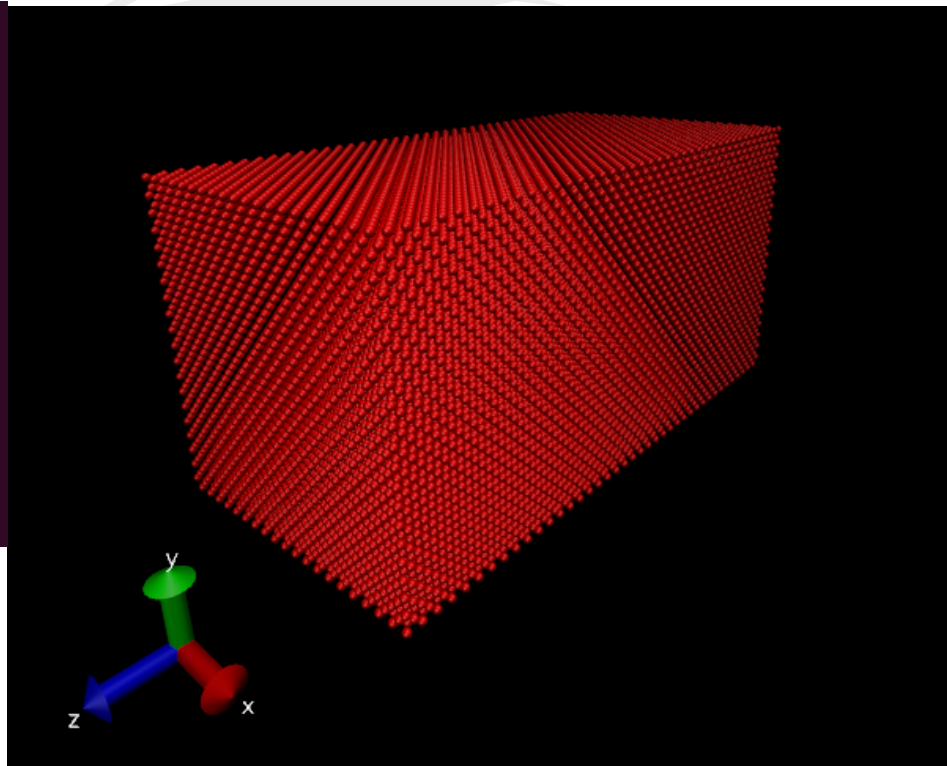
```
# from slides, change later to correct values for Au
#lattice custom 3.2 &
#      a1 -0.5 0.5 0.5 &
#      a2 0.5 -0.5 0.5 &
#      a3 0.5 0.5 -0.5 &
#      basis 0.0 0.0 0.0

lattice fcc 4.080

region myRegion block 0 20 0 20 0 40 units lattice

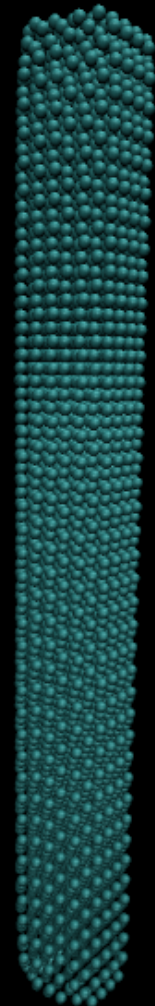
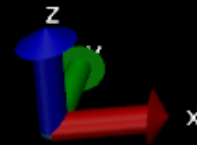
# create a box object from the above defined region
create_box 1 myRegion
create_atoms 1 box basis 1 1
mass 1 196.967
```

- Created brick of 64,000 atoms
- Energy -251,520eV



Creating 001 NW

```
pr = np.loadtxt('pydump.xyz')
Alength = len(A)
x = A[0:Alength,0]    # first column of data
y = A[0:Alength,1]    # Second column of data
z = A[0:Alength,2]    # third column of data
#Reshape with x=x.reshape(length,1)
x=x.reshape(Alength,1)
y=y.reshape(Alength,1)
z=z.reshape(Alength,1)
#Set Radius of NW r
r = 10    #style metal is in Angostroms
##### Nanowire <001> #####
indices2remove=[]
for k in range(Alength):
    if (x[k]-30)**2 + (y[k]-30)**2 > r**2:
        #remove those elements
        #storing indices to an array
        lind2rm=len(indices2remove)
        indices2remove[(1+lind2rm):]=[k]
```



```
##### Nanowire <111> #####
# define function to move increment accurately
def drange2(start, stop, step):
    numelements = int((stop-start)/float(step))
    for i in range(numelements+1):
        yield start + i*step
#arrange in order of Z
for k in range(len(xout)):
    for j in range(k+1,len(xout)):
        if A[k,2] > A[j,2]:
            # method 1: Basic slicing
            temp=np.copy(A[j,:])
            A[j,:]=A[k,:]
            A[k,:]=temp

print "track z-values with NO REPEATS"
indices2remove=[]
zval=[r for r in A[:,2] ] # makes 1D list of zvalues from A
#print zval
c=0
for k in range(c,len(zval)):
    for j in range(k+1,len(zval)):
        if zval[k] == zval[j]:
            lindex=len(indices2remove)
            indices2remove[(1+lindex):]=[j]
            zval=np.delete(zval,indices2remove)
            c+=1
            indices2remove=[]
print zval
# end of tracking zvalues
#for t in range():
for lava in zval:
    for k in range(Alength):
        for t in drange2(11,57,161.0/57):
            if z[k] == lava:
                if (x[k]-t)**2 + (y[k]-t)**2 > r**2:
                    #remove those elements
```

Creating 011 & 111 NWs

#NW 2:

```
#region LLG cylinder y 85 85 10 INF INF units box
#lattice fcc 4.08 orient x 1 0 0 orient y 0 1 -1 orient z 0 1 1
#create_atoms 1 region LLG
```

Energy of 001 & 011

~14eV difference

Setting up minimization ...

Memory usage per processor = 5.54021 Mbytes

Step Press Temp PotEng KinEng TotEng Volume

0	2532.7889	0	-4086.4955
---	-----------	---	------------

4096000

200	-24.149522	0	-11008.795
-----	------------	---	------------

4096000

362	-25.95227	0	-11010.554
-----	-----------	---	------------

4096000

Loop time of 2.93481 on 1 procs for 362 steps with 2964 atoms

Minimization stats:

Stopping criterion = energy tolerance

Energy initial, next-to-last, final =

-4086.49553199	-11010.5542193	-11010.5543124
----------------	----------------	----------------

Setting up minimization ...

Memory usage per processor = 5.53819 Mbytes

Step Press Temp PotEng KinEng TotEng Volume

0	0	24.513586	0	-10221.683
---	---	-----------	---	------------

4096000

0	200	-60.357065	0	-10987.878
---	-----	------------	---	------------

4096000

0	389	-57.333083	0	-10997.364
---	-----	------------	---	------------

4096000

Loop time of 2.95262 on 1 procs for 389 steps with 2964 atoms

Minimization stats:

Stopping criterion = energy tolerance

Energy initial, next-to-last, final =

-10221.6825458	-10997.3640976	-10997.3640976
----------------	----------------	----------------

Compression

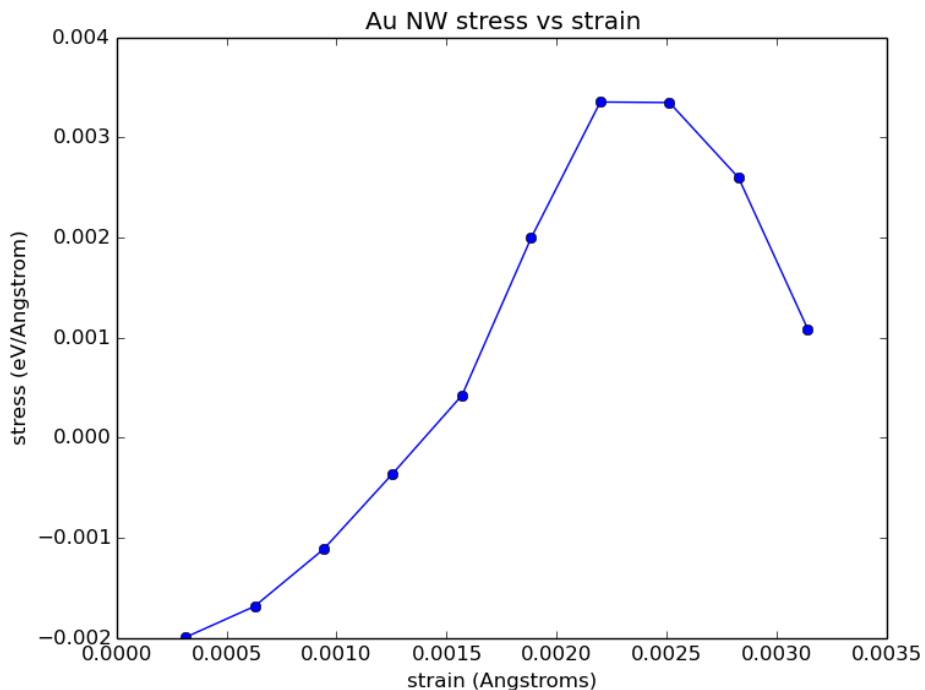
```
variable Nstep loop 10    # repeat 5 times of compression.  
label looplable  
# -0.05  
displace_atoms top move 0 0 -0.09 units box    # compress 0.05 Ang  
mpression step  
#displace_atoms bottom move 0 0 0.09 units box    # compress 0.05  
compression step  
fix 1 3 nvt temp 300.0 300.0 1.0  
run 2000  
next Nstep  
jump in.nwcompress001.txt looplable
```

- 1) <https://www.youtube.com/watch?v=jE8En6FZjQU&feature=youtu.be>
- 2) <https://www.youtube.com/watch?v=tzJn2lCvGKQ&feature=youtu.be>
- 3) <https://www.youtube.com/watch?v=7jjGF-hXtms&feature=youtu.be>
- 4) <https://www.youtube.com/watch?v=liWQjjuVCx4&feature=youtu.be>

Video Links

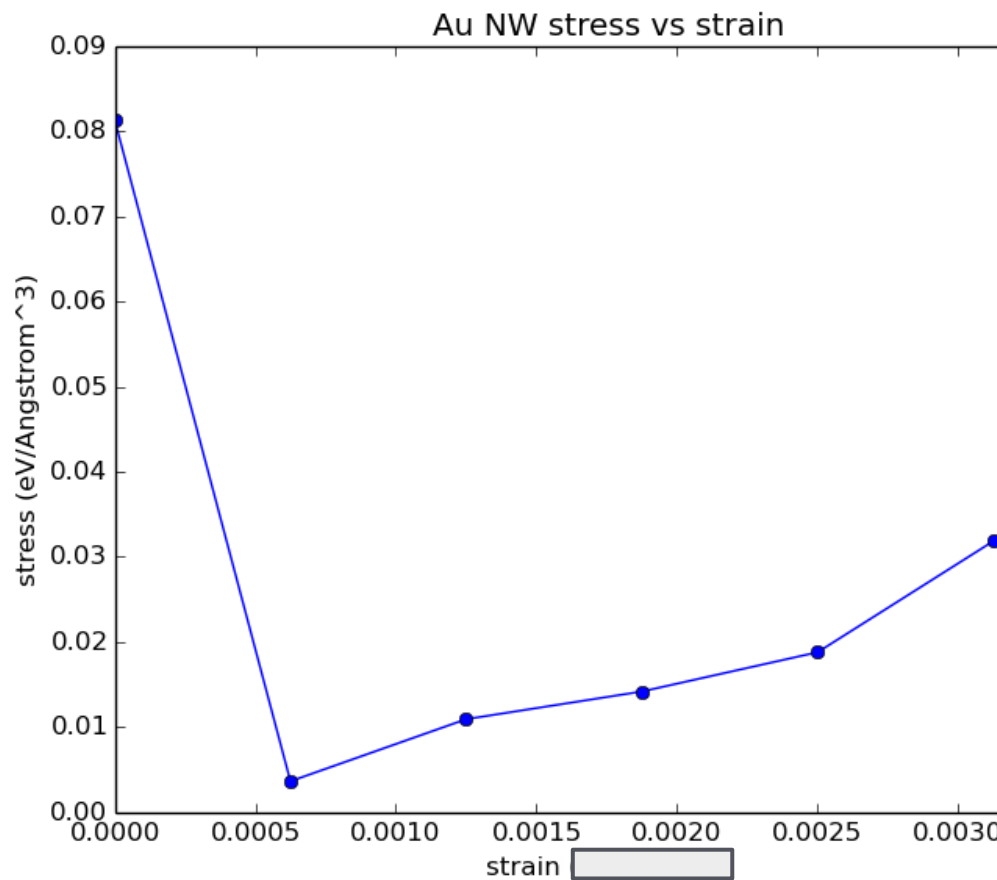
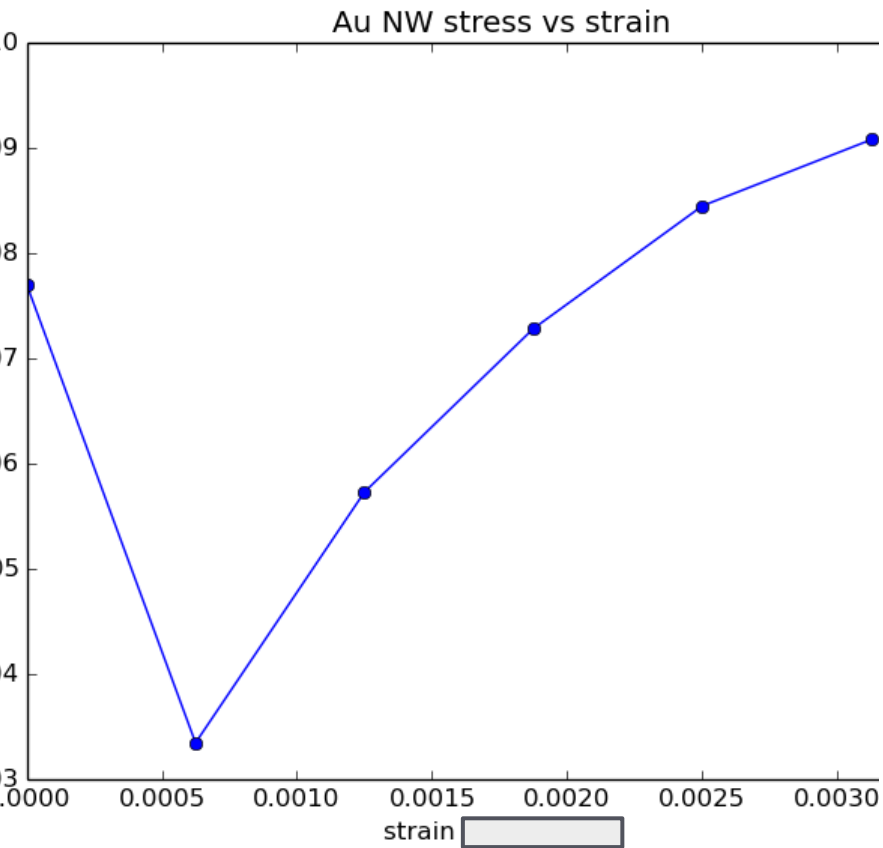


Young's Modulus



```
time_of_loop_in_inputfile=4000
# 1st NW
a=np.loadtxt('outputforce.txt')
b=[]
for k in range(len(a)):
    #if the time is a multiple of 2000 grab it
    if a[k,0]%time_of_loop_in_inputfile == 0 :
        b.append(a[k,:])
B=np.array(b)
print B
del b
#same radius set in py2NW.py
r=10.0
area=np.pi*r**2
#calculate stress
stress=B[:,1]/(np.pi*r**2)
#calculate strain, first declare eprate as in
0 steps or however many Angstroms per #picosecond
eprate=abs(-0.05)
deltaL=eprate/time_of_loop_in_inputfile
L=159.273
strain=B[:,0]*deltaL/L
#plot1
```


Moduli $r=10, ed=0.1$, 001(left) & 011(right)



Initial Results in eV/Angstrom³

$$e4 = (\text{stress}[4] - \text{stress}[1]) / (\text{strain}[4] - \text{strain}[1]) = 2.2388042$$

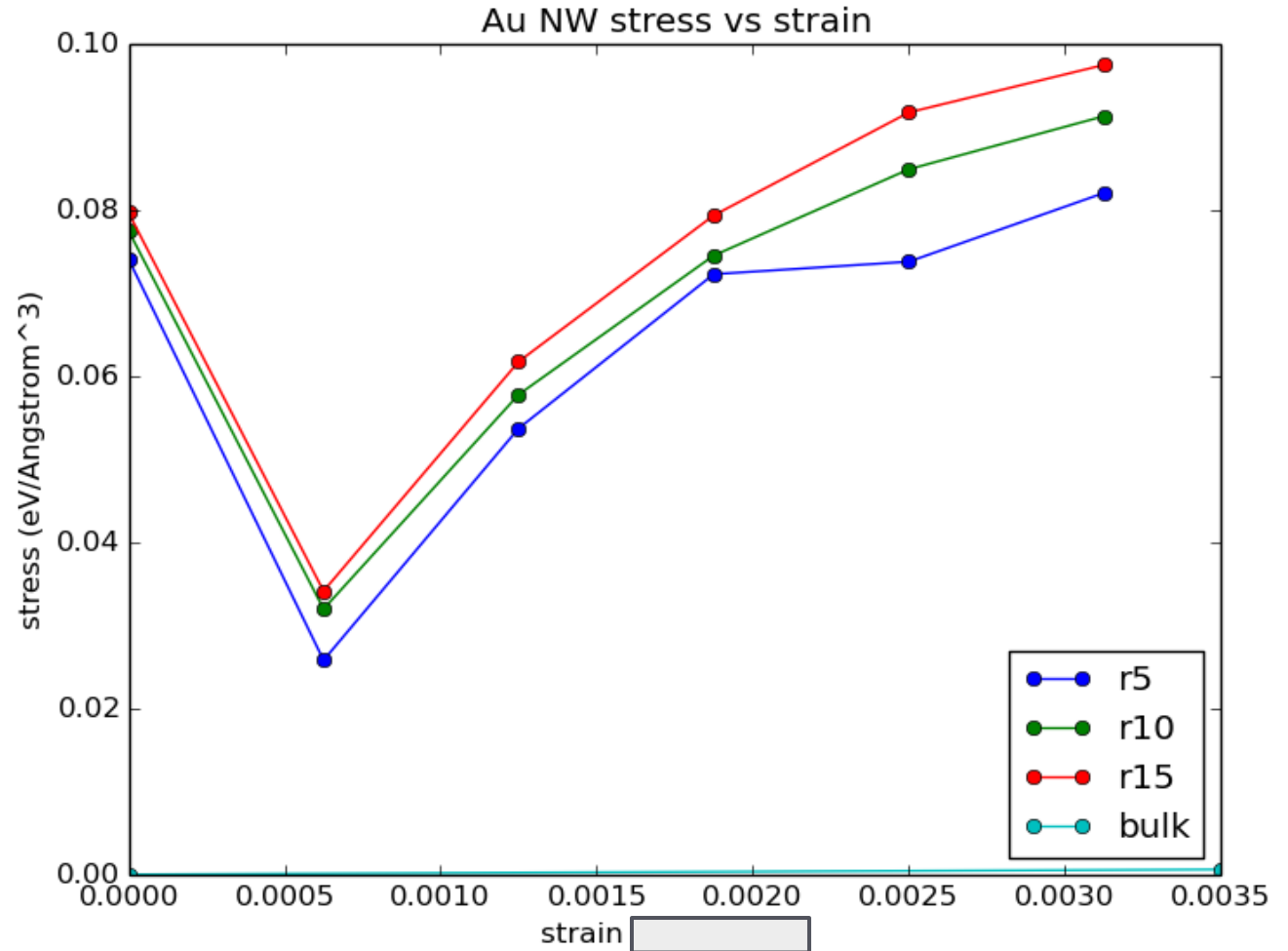
$$e6 = (\text{stress}[6] - \text{stress}[4]) / (\text{strain}[6] - \text{strain}[4]) = 4.6673040$$

$$\text{enwavg} = 3.4530541$$

$$\text{ebulk} = 0.1715986331$$

- Bulk Au has $E = 29 \text{ GPa}$ or 0.172 eV/\AA^3
- NW Au has $E = 583.5 \text{ GPa}$
- Ratio $\text{Enwavg}/\text{Ebulk} = 20.12850$
- Next steps
 - Compare with a different radii NWs
 - Test orientations $\langle 011 \rangle$, $\langle 111 \rangle$

Comparing Young's Modulus for different size radii



Extras

- 1) <https://www.youtube.com/watch?v=OSw3CaI2u6U&feature=youtu.be>
- 2) <https://www.youtube.com/watch?v=dpQ3-waEUT4&feature=youtu.be>