Exceptional service in the national interest

Sandia National Laboratories

Basic Output Post-Processing

Aidan Thompson, Dept. 1444, Multiscale Science
Sandia National Laboratories

August 2015 LAMMPS Users' Workshop and Symposium

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

LDRD LABORATORY DIRECTED RESEARCH & DEVELOPMENT

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. **SAND NO. 2013-1055C, 2013-5546C, 2013-5547C**

Post-Processing Requirements

- Lightweight ➔ No fancy post-processing packages
- Extensible/Reusable ➔ No obscure one-line commands
- General ➔ No specialized LAMMPS commands
- Organized Workflow ➔ Generate named files along the way

Possible Solutions

Microsoft Excel

- Widely used for other purposes
- Fast learning curve
- Easy things are easy
- Not easy to automate or extend
- Plotting included

Python

- Requires some programming knowledge
- LAMMPS provides some good support via Pizza.py
- Not as easy as Excel
- Easy to automate/extend
- Plotting not included

Pizza.py

Powerful, but not easy

Weird Shell Miniscripts

```
awk /Step/,/step/ log.lammps | grep -vi step | xmgrace -block -bxy 1:3
```

Example Problem: Lennard-Jones Melt

```
# 3d Lennard-Jones melt
variable        t index 0.0
variable        d index 0.8442
log             melt_t$t.log

units           lj
atom_style      atomic

lattice         fcc $d
region          box block 0 10 0 10 0 10
create_box      1 box
create_atoms    1 box
mass            1 1.0
velocity        all create $t 87287

pair_style      lj/cut 2.5
pair_coeff      1 1 1.0 1.0 2.5
neighbor        0.3 bin
neigh_modify    every 20 delay 0 check no

fix             1 all nve
thermo          10
run             1000
```

Example Problem: Lennard-Jones Melt

```
# 3d Lennard-Jones melt
variable        t index 0.0
variable        d index 0.8442
log             melt_t$t.log

units           lj
atom_style      atomic

lattice         fcc $d
region          box block 0 10 0 10 0 10
create_box      1 box
create_atoms    1 box
mass            1 1.0
velocity        all create $t 87287

pair_style      lj/cut 2.5
pair_coeff      1 1 1.0 1.0 2.5
neighbor        0.3 bin
neigh_modify    every 20 delay 0 check no

fix             1 all nve
thermo          10
run             1000
```

Example Problem: Lennard-Jones Melt

```
[athomps@s974522 melt]$ $LAMMPS/src/lmp_mac_mpi -in in.melt -var t 3.0
:
Memory usage per processor = 2.19271 Mbytes
Step Temp          E_pair          E_mol      TotEng      Press
     0    3        -6.7733681          0   -2.2744931   -3.7033504
    10    2.1031859  -5.4317227          0   -2.2777326    2.4556252
    20    1.6681241  -4.7867954          0   -2.2852348    5.6910177
    30    1.6750435  -4.7955971          0   -2.2836601    5.6241836
:
   990    1.6444585  -4.7526932          0   -2.2866221    5.8339173
  1000    1.6547203  -4.7681016          0   -2.2866417    5.7734588
Loop time of 2.36568 on 1 procs for 1000 steps with 4000 atoms
:
```

Example Problem: Lennard-Jones Melt

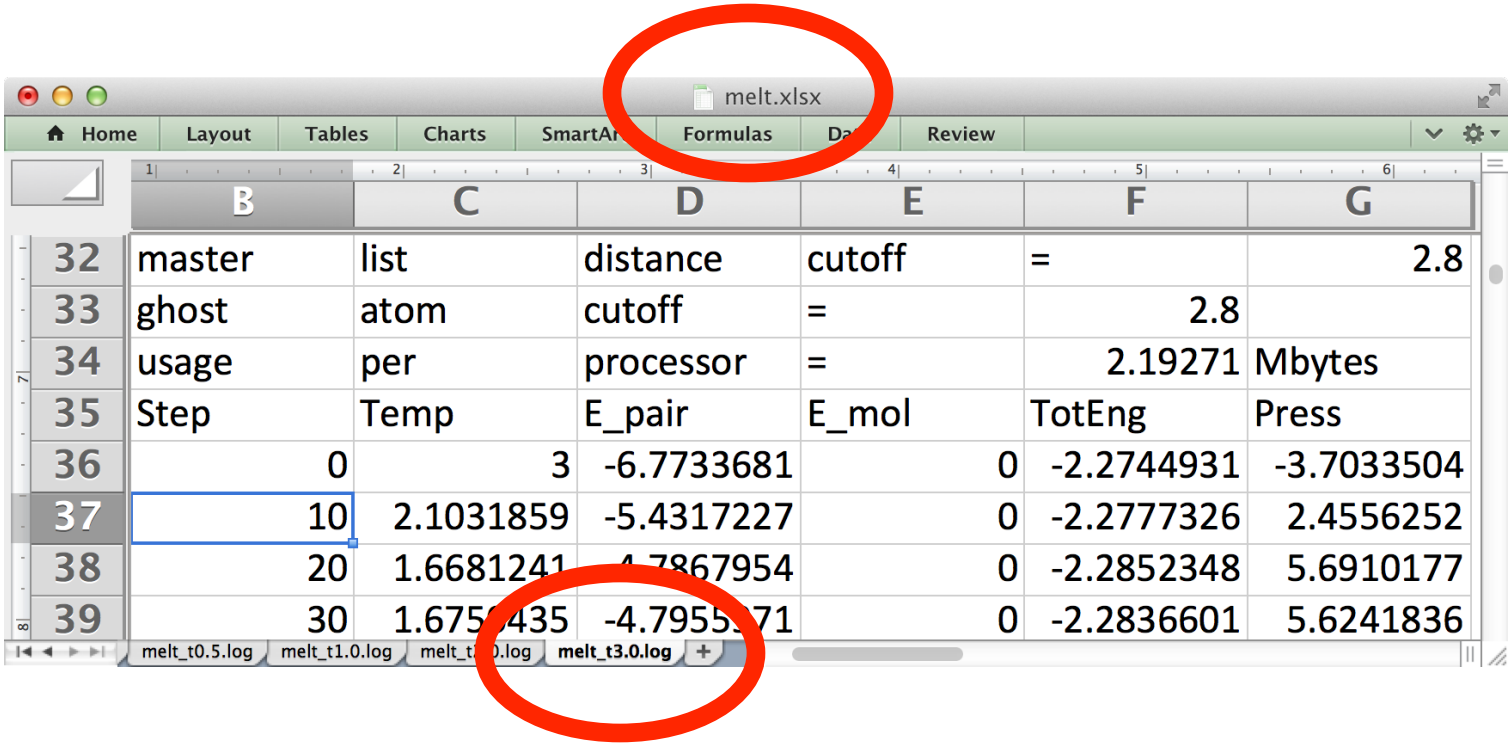
```
[athomps@s97 melt]$ lmp_mac_mpi -in in.melt -var t 0.5 -screen none
[athomps@s97 melt]$ lmp_mac_mpi -in in.melt -var t 1.0 -screen none
[athomps@s97 melt]$ lmp_mac_mpi -in in.melt -var t 2.0 -screen none
[athomps@s97 melt]$ lmp_mac_mpi -in in.melt -var t 3.0 -screen none
[athomps@s97 melt]$ ls -1 *.log
melt_t0.5.log
melt_t1.0.log
melt_t2.0.log
melt_t3.0.log
```

Objectives

1. Check that simulation is equilibrated
2. Estimate equilibrium temperature and potential energy
3. Estimate uncertainty in these estimates
4. Create nice graph of Epot versus Temp, with error bars

Microsoft Excel Spreadsheet

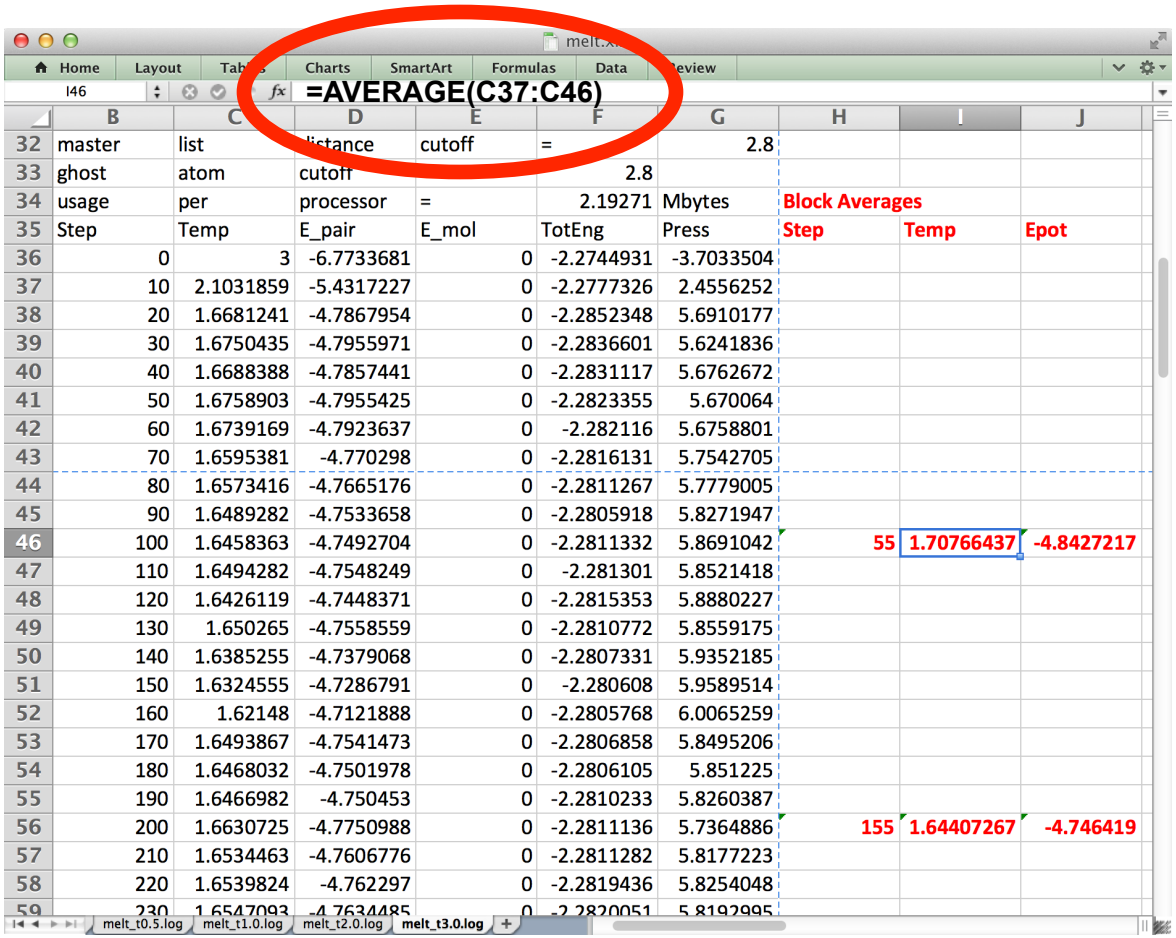
- Each log file can be opened using “File/Open” or “Command O”
- In Import Wizard, select Delimited, Tab and Space Delimiters
- Drag the sheet tab to the current workbook
- Sheet will have same name as log file



	B	C	D	E	F	G
32	master	list	distance	cutoff	=	2.8
33	ghost	atom	cutoff	=	2.8	
34	usage	per	processor	=	2.19271	Mbytes
35	Step	Temp	E_pair	E_mol	TotEng	Press
36	0	3	-6.7733681	0	-2.2744931	-3.7033504
37	10	2.1031859	-5.4317227	0	-2.2777326	2.4556252
38	20	1.6681241	-4.7867954	0	-2.2852348	5.6910177
39	30	1.6756435	-4.7955[illegible]71	0	-2.2836601	5.6241836

Microsoft Excel Spreadsheet

- Fairly easy to add formulas for block averages



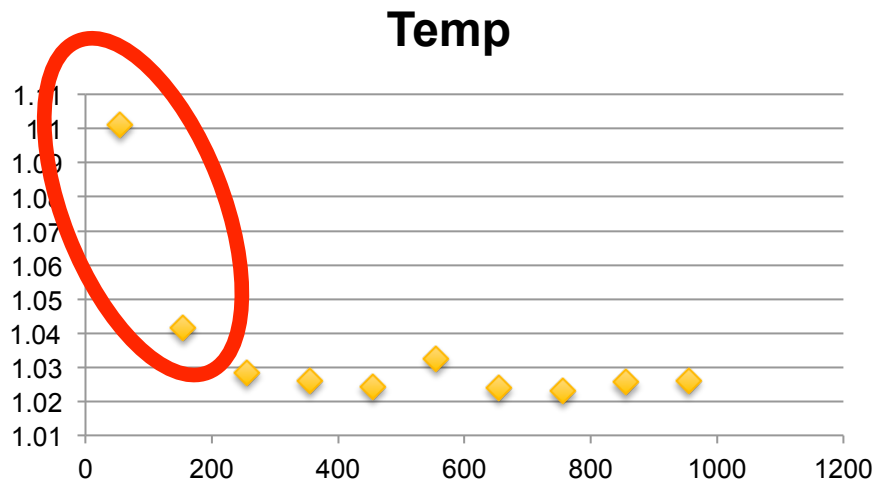
=AVERAGE(C37:C46)

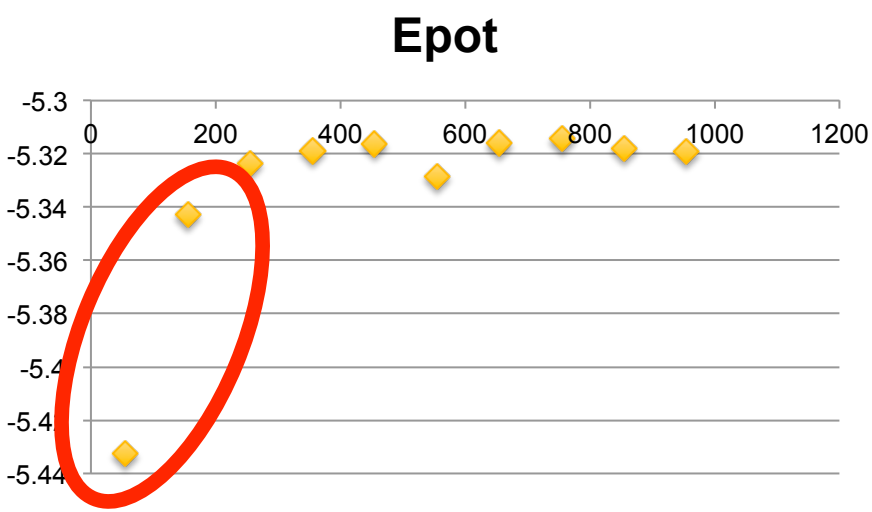
	B	C	D	E	F	G	H	I	J
32	master	list	distance	cutoff	=	2.8			
33	ghost	atom	cutoff		2.8				
34	usage	per	processor	=	2.19271	Mbytes	Block Averages		
35	Step	Temp	E_pair	E_mol	TotEng	Press	Step	Temp	Epot
36	0	3	-6.7733681	0	-2.2744931	-3.7033504			
37	10	2.1031859	-5.4317227	0	-2.2777326	2.4556252			
38	20	1.6681241	-4.7867954	0	-2.2852348	5.6910177			
39	30	1.6750435	-4.7955971	0	-2.2836601	5.6241836			
40	40	1.6688388	-4.7857441	0	-2.2831117	5.6762672			
41	50	1.6758903	-4.7955425	0	-2.2823355	5.670064			
42	60	1.6739169	-4.7923637	0	-2.282116	5.6758801			
43	70	1.6595381	-4.770298	0	-2.2816131	5.7542705			
44	80	1.6573416	-4.7665176	0	-2.2811267	5.7779005			
45	90	1.6489282	-4.7533658	0	-2.2805918	5.8271947			
46	100	1.6458363	-4.7492704	0	-2.2811332	5.8691042	55	1.70766437	-4.8427217
47	110	1.6494282	-4.7548249	0	-2.281301	5.8521418			
48	120	1.6426119	-4.7448371	0	-2.2815353	5.8880227			
49	130	1.650265	-4.7558559	0	-2.2810772	5.8559175			
50	140	1.6385255	-4.7379068	0	-2.2807331	5.9352185			
51	150	1.6324555	-4.7286791	0	-2.280608	5.9589514			
52	160	1.62148	-4.7121888	0	-2.2805768	6.0065259			
53	170	1.6493867	-4.7541473	0	-2.2806858	5.8495206			
54	180	1.6468032	-4.7501978	0	-2.2806105	5.851225			
55	190	1.6466982	-4.750453	0	-2.2810233	5.8260387			
56	200	1.6630725	-4.7750988	0	-2.2811136	5.7364886	155	1.64407267	-4.746419
57	210	1.6534463	-4.7606776	0	-2.2811282	5.8177223			
58	220	1.6539824	-4.762297	0	-2.2819436	5.8254048			
59	230	1.6547093	-4.7634485	0	-2.2820051	5.8192995			

melt_t0.5.log | melt_t1.0.log | melt_t2.0.log | melt_t3.0.log

Microsoft Excel Spreadsheet

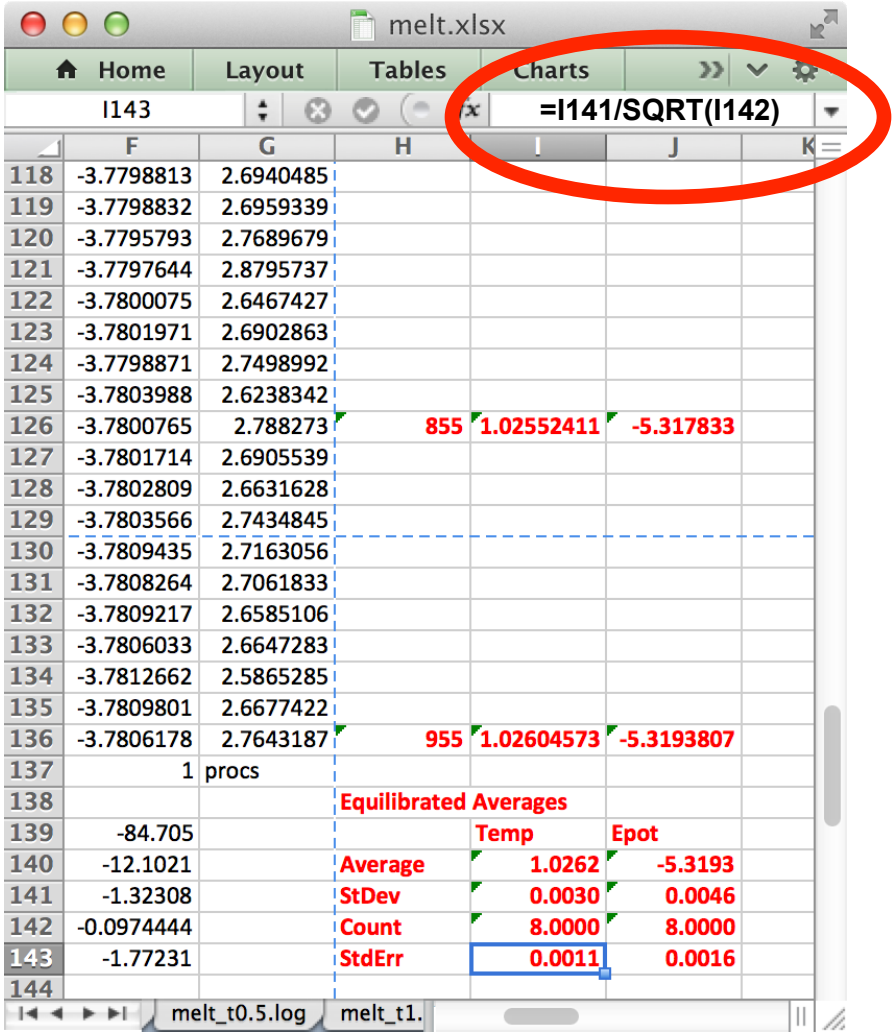
- Plot the block averages
- Decide how many blocks to throw away (2)





Microsoft Excel Spreadsheet

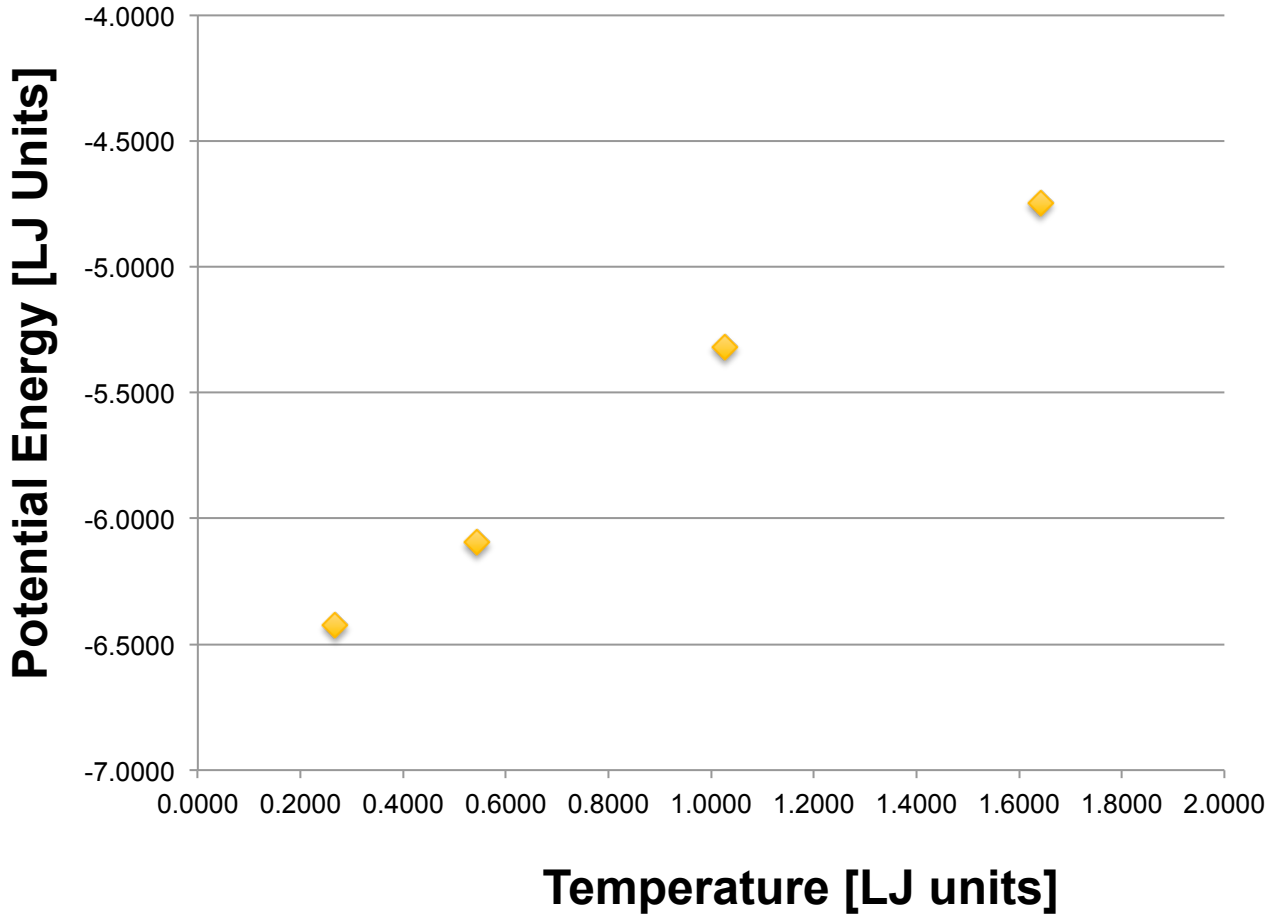
- Calculate average over equlibrated blocks
- Calculate standard error = StDev/Sqrt(N)



	F	G	H	I	J
118	-3.7798813	2.6940485			
119	-3.7798832	2.6959339			
120	-3.7795793	2.7689679			
121	-3.7797644	2.8795737			
122	-3.7800075	2.6467427			
123	-3.7801971	2.6902863			
124	-3.7798871	2.7498992			
125	-3.7803988	2.6238342			
126	-3.7800765	2.788273	855	1.02552411	-5.317833
127	-3.7801714	2.6905539			
128	-3.7802809	2.6631628			
129	-3.7803566	2.7434845			
130	-3.7809435	2.7163056			
131	-3.7808264	2.7061833			
132	-3.7809217	2.6585106			
133	-3.7806033	2.6647283			
134	-3.7812662	2.5865285			
135	-3.7809801	2.6677422			
136	-3.7806178	2.7643187	955	1.02604573	-5.3193807
137	1	procs			
138			Equilibrated Averages		
139	-84.705			Temp	Epot
140	-12.1021		Average	1.0262	-5.3193
141	-1.32308		StDev	0.0030	0.0046
142	-0.0974444		Count	8.0000	8.0000
143	-1.77231		StdErr	0.0011	0.0016
144					

Microsoft Excel Spreadsheet

- Plot T vs. E



Python Scripting

Location of LAMMPS pizza tools

```
[ath@s97 melt]$ setenv LAMMPS_PYTHON_TOOLS $LAMMPS/tools/python/pizza
[ath@s97 melt]$ more thermo.py
#!/usr/bin/env python
import sys,os
path = os.environ["LAMMPS_PYTHON_TOOLS"]
#path = '..../tools/python/pizza'
sys.path.append(path)
from log import log
:
[ath@s97 melt]$ chmod +x thermo.py
```

Determines which Python to use

Loads the log.py tool

Gives you permission to run thermo.py

Python Scripting

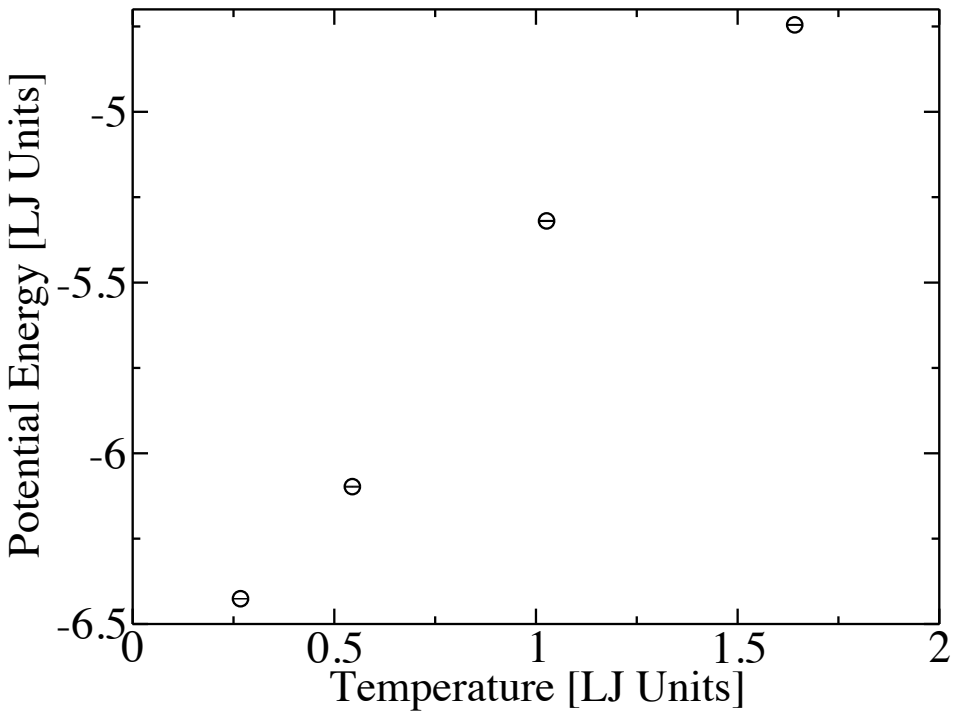
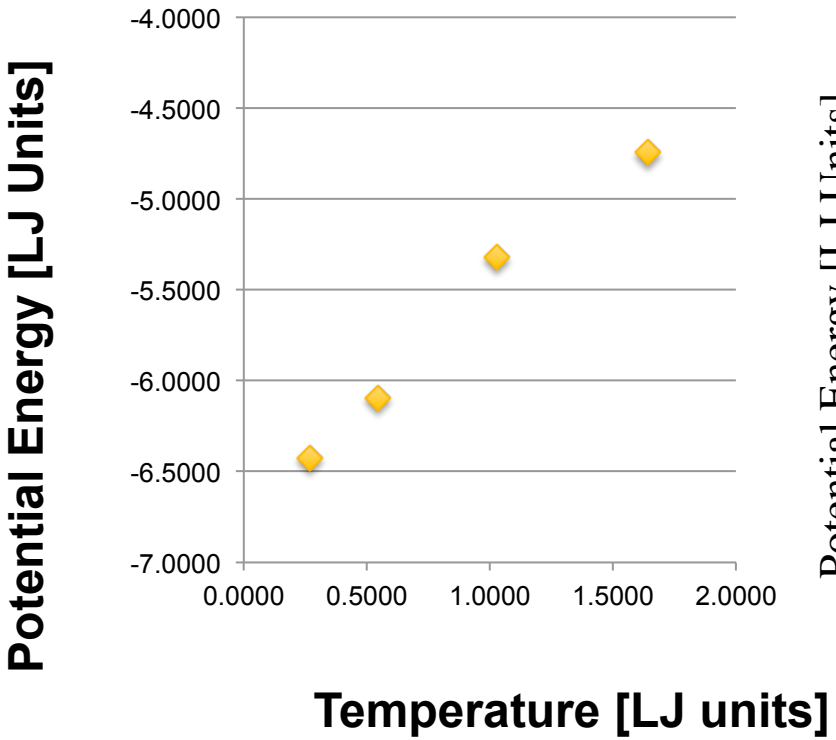
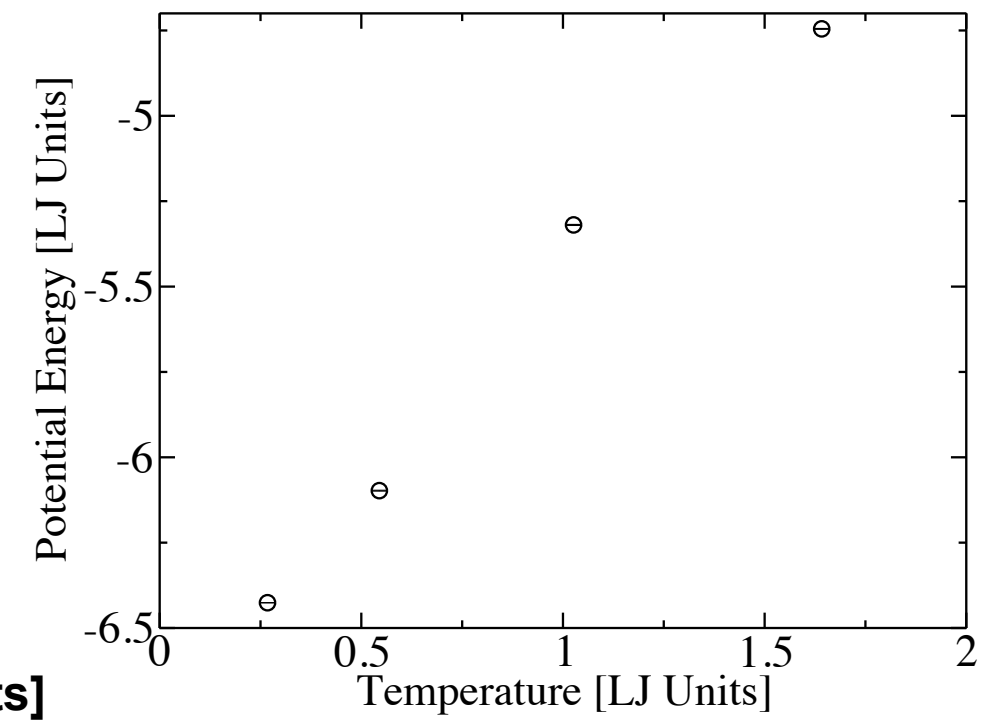
```
[ath@s97 melt]$ more thermo.py
#!/usr/bin/env python
from log import log
from glob import glob
from math import sqrt

def onelog(filename):
    l = log(filename)
    print filename
    list = l.get("E_pair")
    nequil = 2
    nblock = 10

    sumav = 0.0
    sumavsq = 0.0
    nsumav = 0
    n = 0
    sum = 0.0
    for val in list:
        sum += val
        n += 1
        if (n % nblock == 0):
            av = sum/nblock
            if (n > nequil):
                sumav += av
                sumavsq += av*av
                nsumav += 1
            print n,av
            sum = 0.0

    av = sumav/nsumav
    var = sumavsq/nsumav - av*av
    stdev = sqrt(var/nsumav)
    print "Average potential energy " \
        "(LJ Units) = %g +/- %g from " \
        "last %d blocks" \
        % (av,stdev,nsumav)

files = glob('melt_t*.log')
for file in files:
    onelog(file)
```

Python Script

```
[ath@s97 melt]$ thermo.py
1000
read 101 log entries
melt_t0.5.log
10 -6.44181068
20 -6.42596615
:
1000
read 101 log entries
melt_t3.0.log
10 -5.0451315
20 -4.74383611
30 -4.75156147
40 -4.74670976
50 -4.74370168
60 -4.74488213
70 -4.74370246
80 -4.74835012
90 -4.74458378
100 -4.73887062
Average potential energy (LJ Units) = -4.7453 +/- 0.00123508 from last 8 blocks
```

Python Scripting

```
[athomps@s974522 melt]$ thermo.py | grep Average
Average potential energy (LJ Units) = -6.4259 +/- 0.000220776 from last 8 blocks
Average potential energy (LJ Units) = -6.09779 +/- 0.00118196 from last 8 blocks
Average potential energy (LJ Units) = -5.31951 +/- 0.00146715 from last 8 blocks
Average potential energy (LJ Units) = -4.7453 +/- 0.00123508 from last 8 block
```



Comparison of Final Results