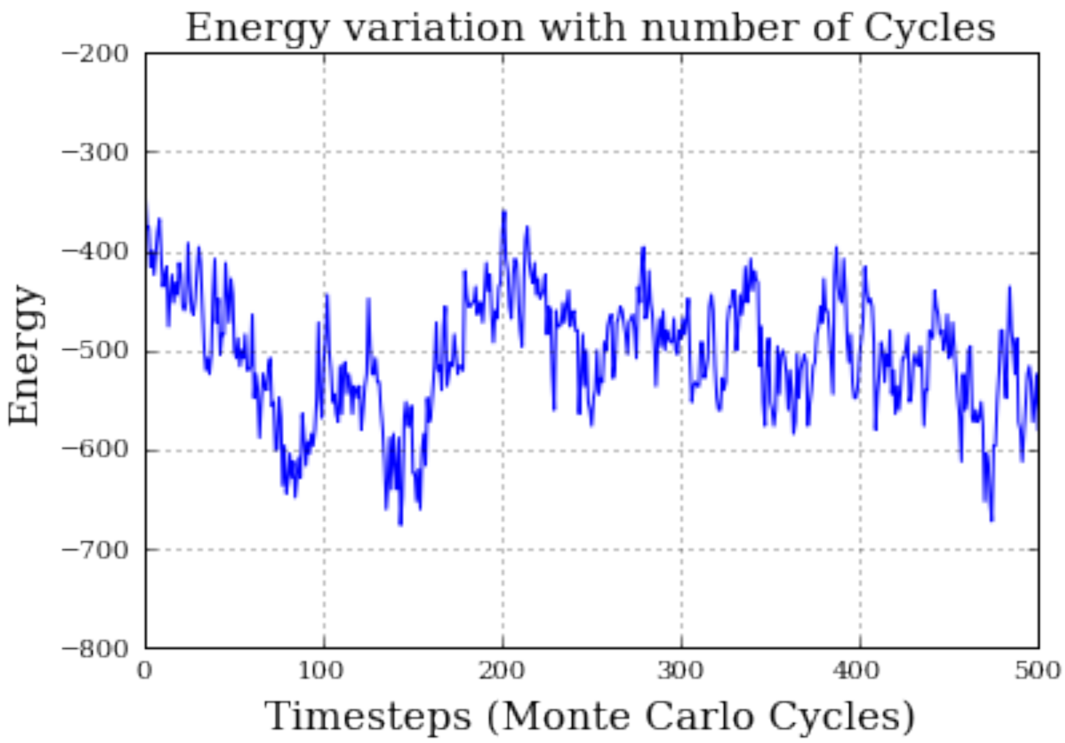
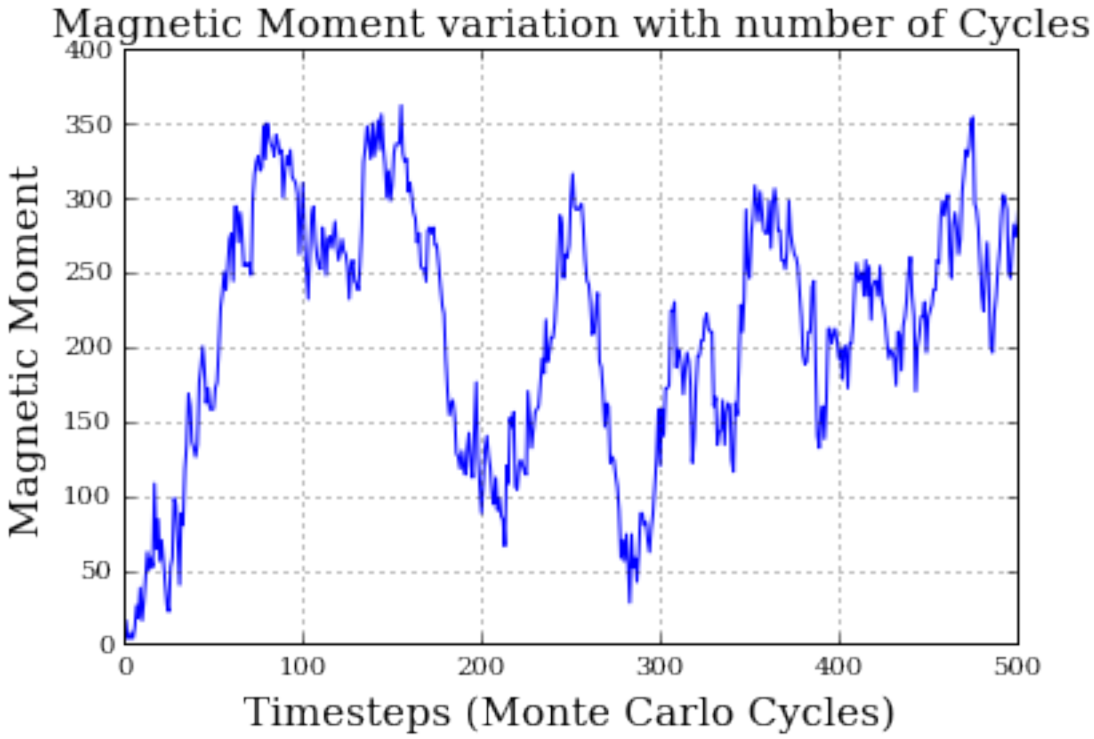
In [119]:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import rc




from pylab import *
from scipy.interpolate import UnivariateSpline
from matplotlib import interactive
#interactive(True)


rc('font',
**
{'family':'serif'})
import pylab
from numpy import *
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import scipy.interpolate
from scipy import stats
get_ipython().magic(u'matplotlib inline')



#T=2.4 1e4 timesteps
E, M, t = np.loadtxt('N20T24steps1e4.txt', usecols=(0,1,2), unpack=True) #N2T1e4#N2

plt.figure()
plt.plot(t, E)
plt.title("Energy variation with number of Cycles", size=15)
plt.ylabel("Energy", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
plt.xlim(0, 500)
plt.grid()

plt.figure()
plt.plot(t, M)
plt.title("Magnetic Moment variation with number of Cycles", size=15)
plt.ylabel("Magnetic Moment", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
plt.xlim(0, 500)
plt.grid()

#finding the probability

Prob_E8= np.where(E == -500.)
print(np.size(Prob_E8))
```

```
243
```

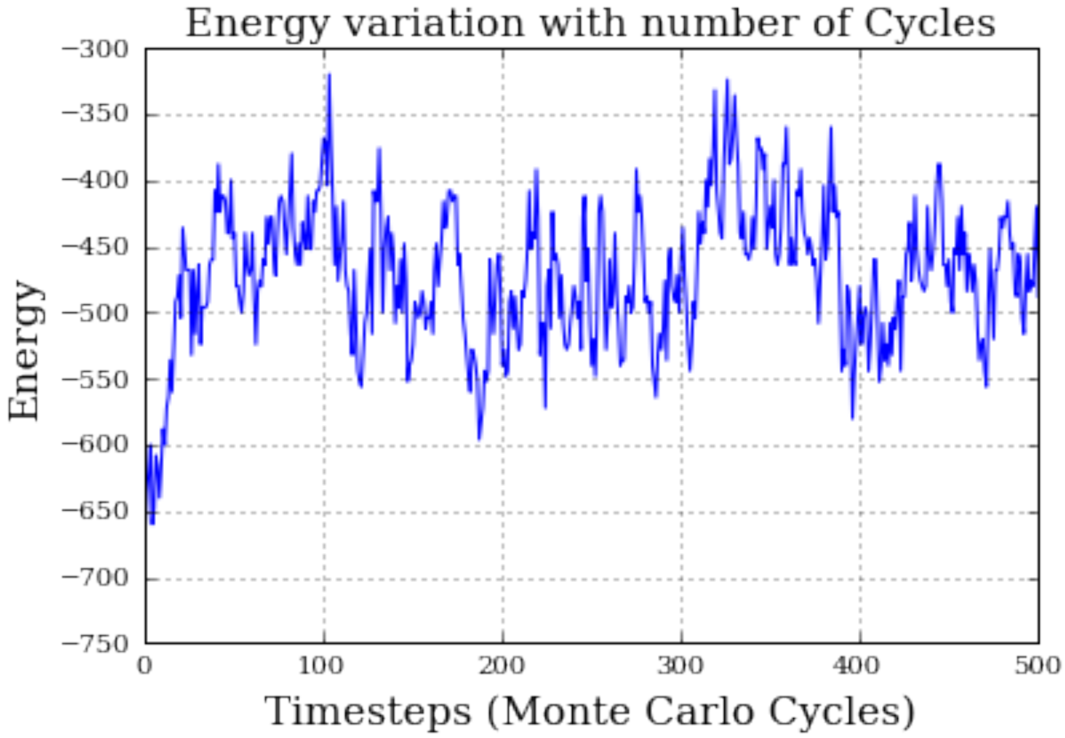


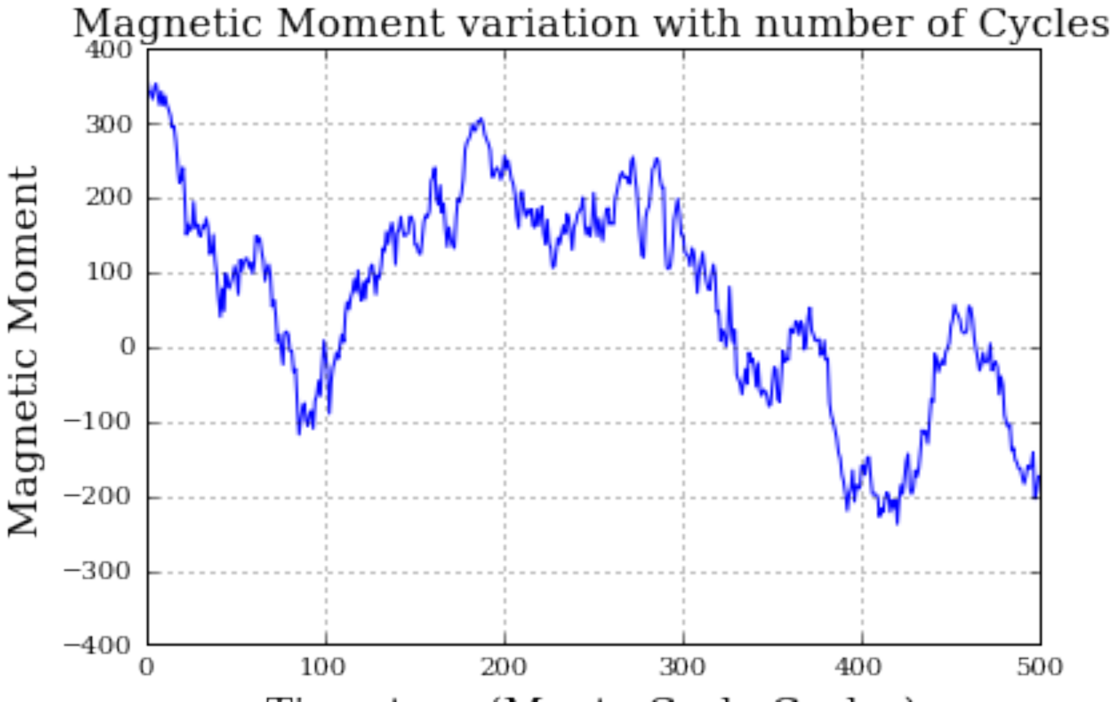


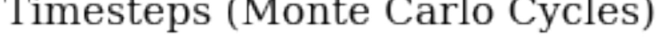
In [4]:

```
#2.4 ordered 1e4




#T=2.4 1e4 timesteps
E, M, t = np.loadtxt('N20T24steps1e4ORDERED.txt', usecols=(0,1,2), unpack=True) #N2

plt.figure()
plt.plot(t, E)
plt.title("Energy variation with number of Cycles", size=15)
plt.ylabel("Energy", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
plt.xlim(0, 500)
plt.xlim(0, 500)

plt.grid()

plt.figure()
plt.plot(t, M)
plt.title("Magnetic Moment variation with number of Cycles", size=15)
plt.ylabel("Magnetic Moment", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
plt.xlim(0, 500)

plt.grid()

#finding the probability

Prob_E8= np.where(E == -500.)
print(np.size(Prob_E8))
```

```
235
```



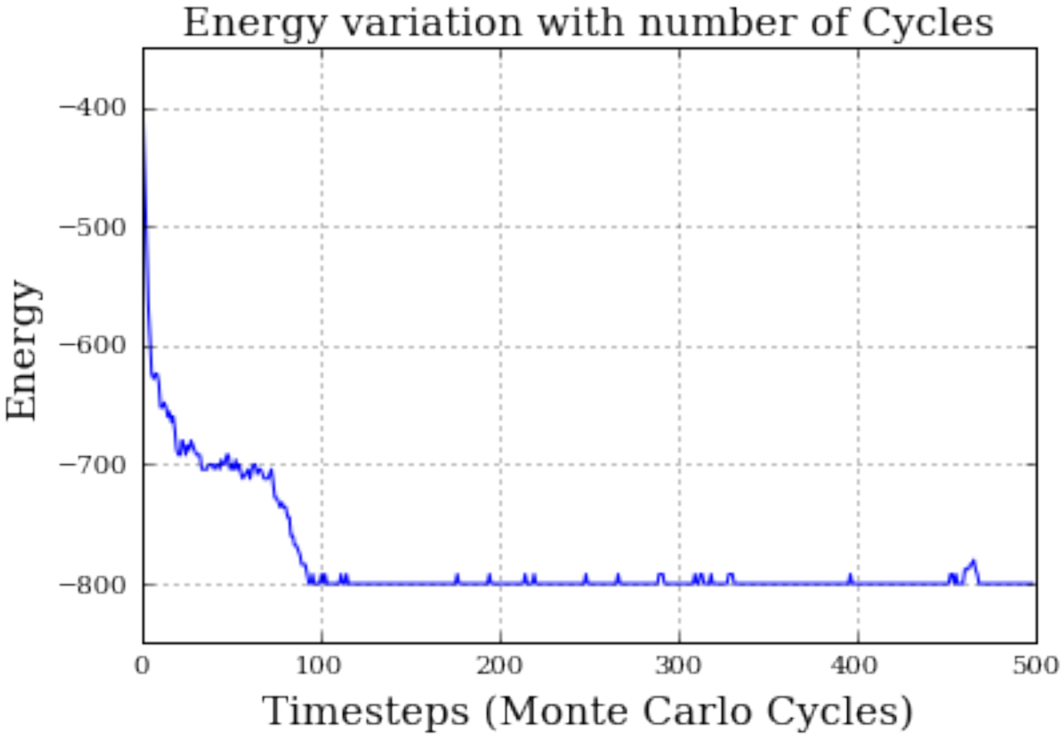


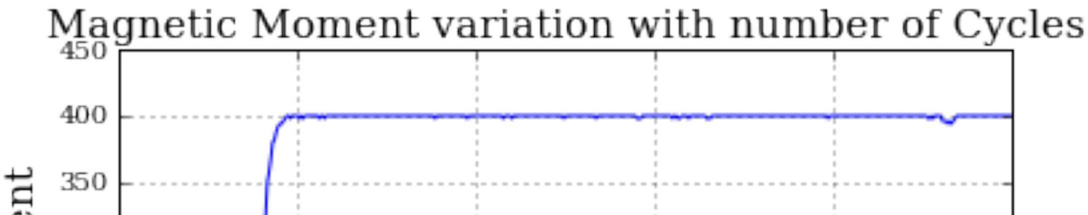


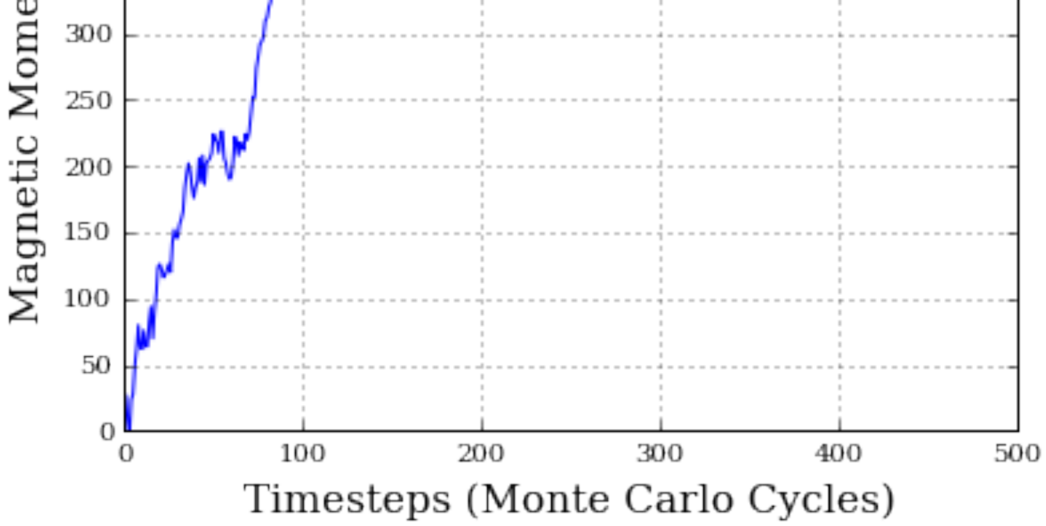
In [11]:

```
# In[36]: Temp = 1 500 timesteps

E, M, t = np.loadtxt('N20T1steps500.txt', usecols=(0,1,2), unpack=True) #N2T1e4

plt.figure()
plt.plot(t, E)
plt.title("Energy variation with number of Cycles", size=15)
plt.ylabel("Energy", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
plt.ylim(-850, -350)
plt.grid()

plt.figure()
plt.plot(t, M)
plt.title("Magnetic Moment variation with number of Cycles", size=15)
plt.ylabel("Magnetic Moment", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
plt.ylim(0, 450)
plt.grid()



pp=np.where(t > 100.)

Prob_E8= np.where(E[pp] == -800.)
print(np.size(Prob_E8))
```

```
367
```

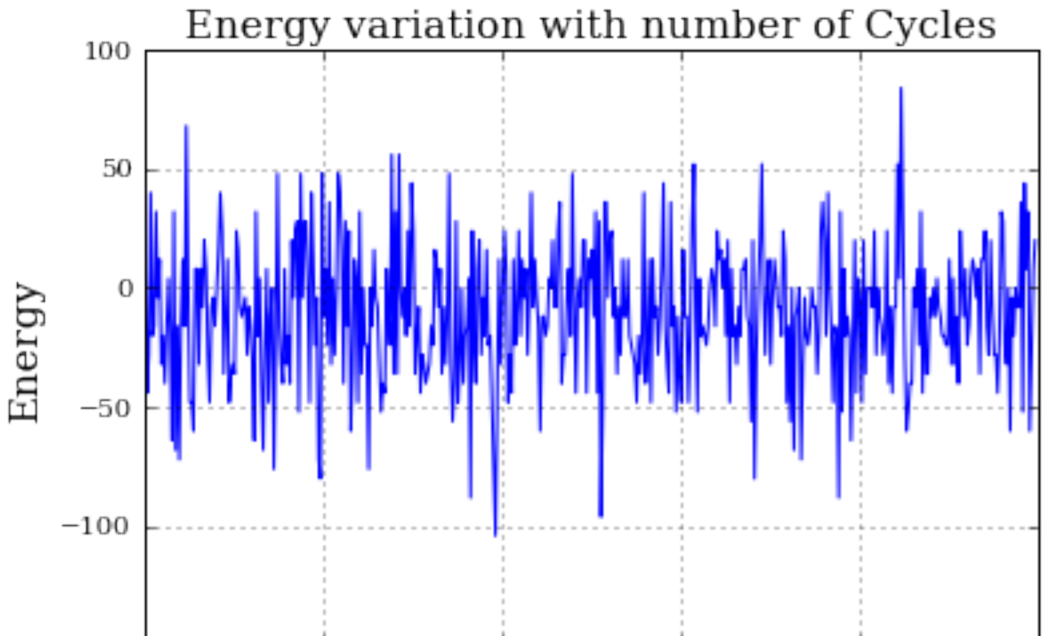


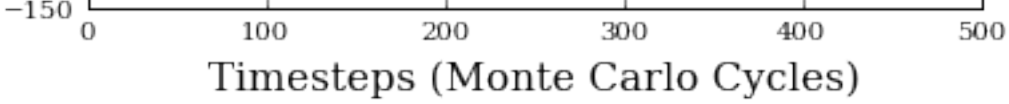


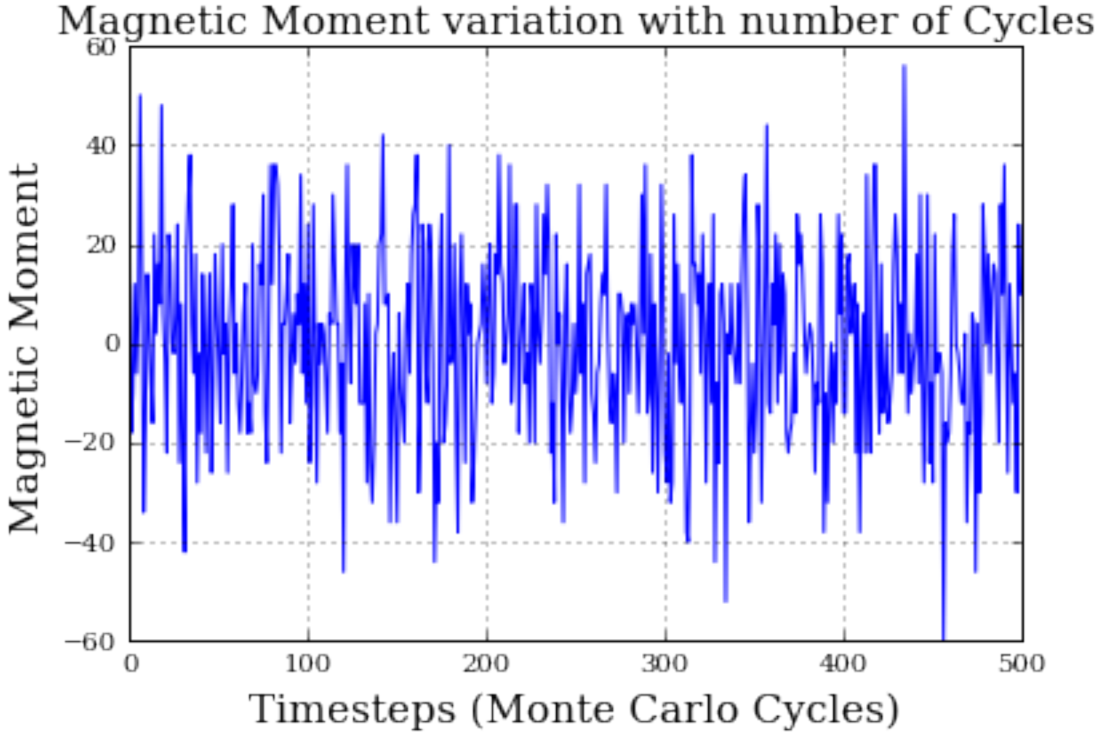


In [12]:

```
# In[38]:
#T=100 500 timesteps



E, M, t = np.loadtxt('N20T100steps500.txt', usecols=(0,1,2), unpack=True) #N2T1e4

plt.figure()
plt.plot(t, E)
plt.title("Energy variation with number of Cycles", size=15)
plt.ylabel("Energy", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
#plt.ylim(-850, -350)
plt.grid()

plt.figure()
plt.plot(t, M)
plt.title("Magnetic Moment variation with number of Cycles", size=15)
plt.ylabel("Magnetic Moment", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
#plt.ylim(0, 450)
plt.grid()
```

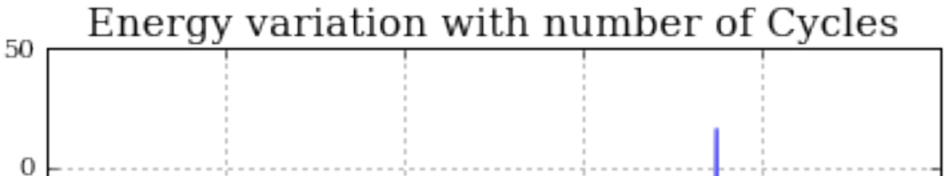

Energy variation with number of Cycles

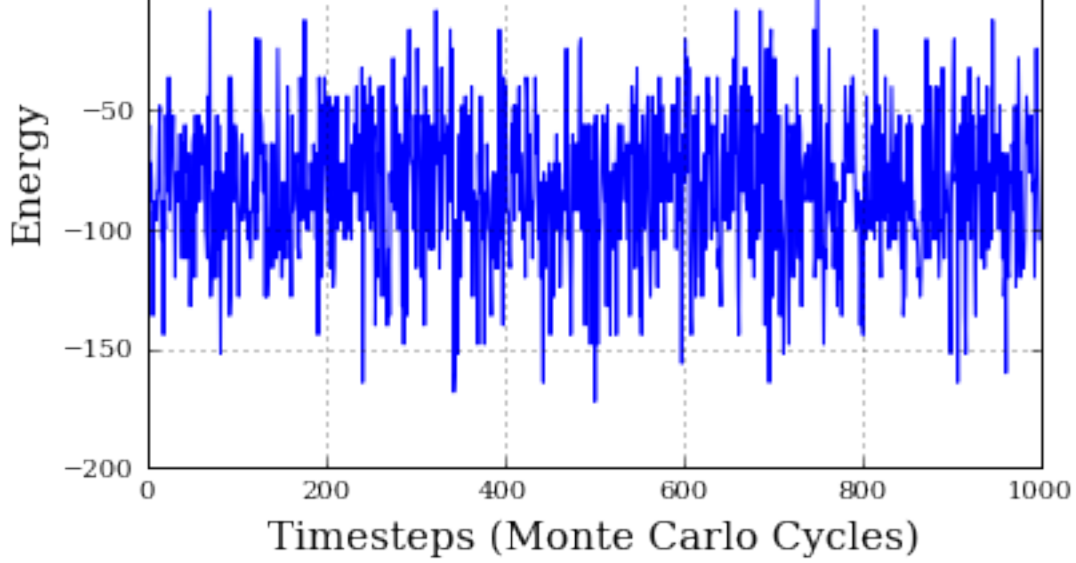


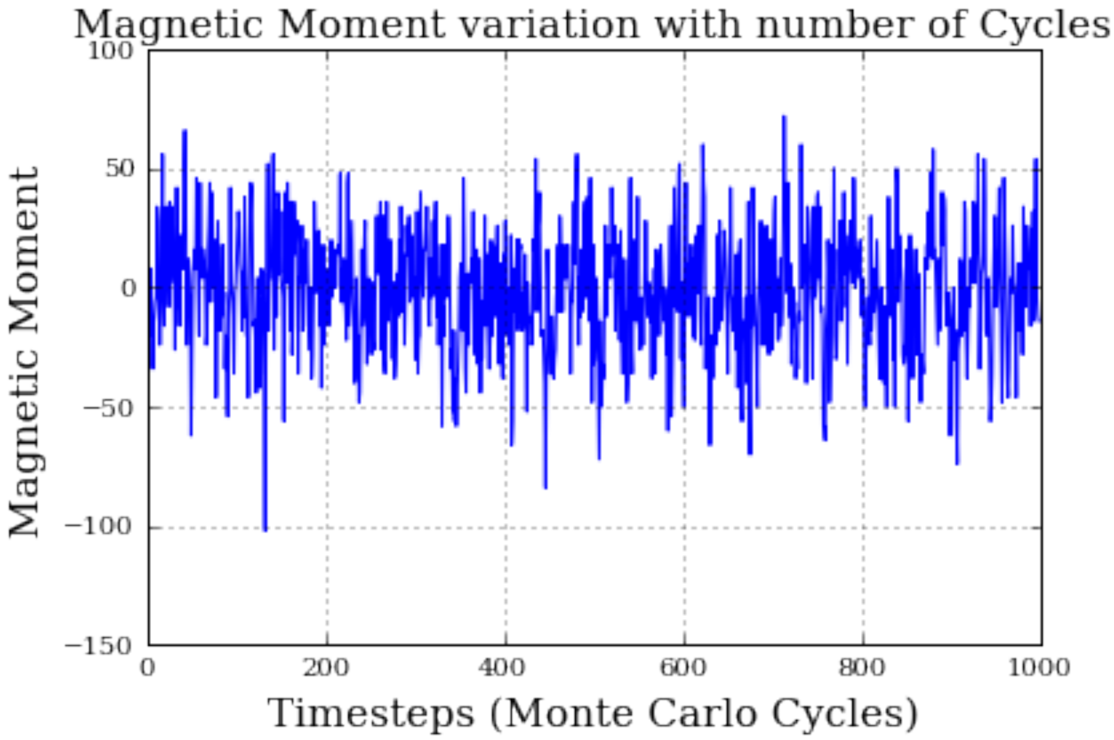


In [13]:

```
# In[41]:

#T=10 1e3 timesteps




E, M, t = np.loadtxt('N20T10steps1e3.txt', usecols=(0,1,2), unpack=True) #N2T1e4

plt.figure()
plt.plot(t, E)
plt.title("Energy variation with number of Cycles", size=15)
plt.ylabel("Energy", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
#plt.ylim(-850, -350)
plt.grid()

plt.figure()
plt.plot(t, M)
plt.title("Magnetic Moment variation with number of Cycles", size=15)
plt.ylabel("Magnetic Moment", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
#plt.ylim(0, 450)
plt.grid()
```

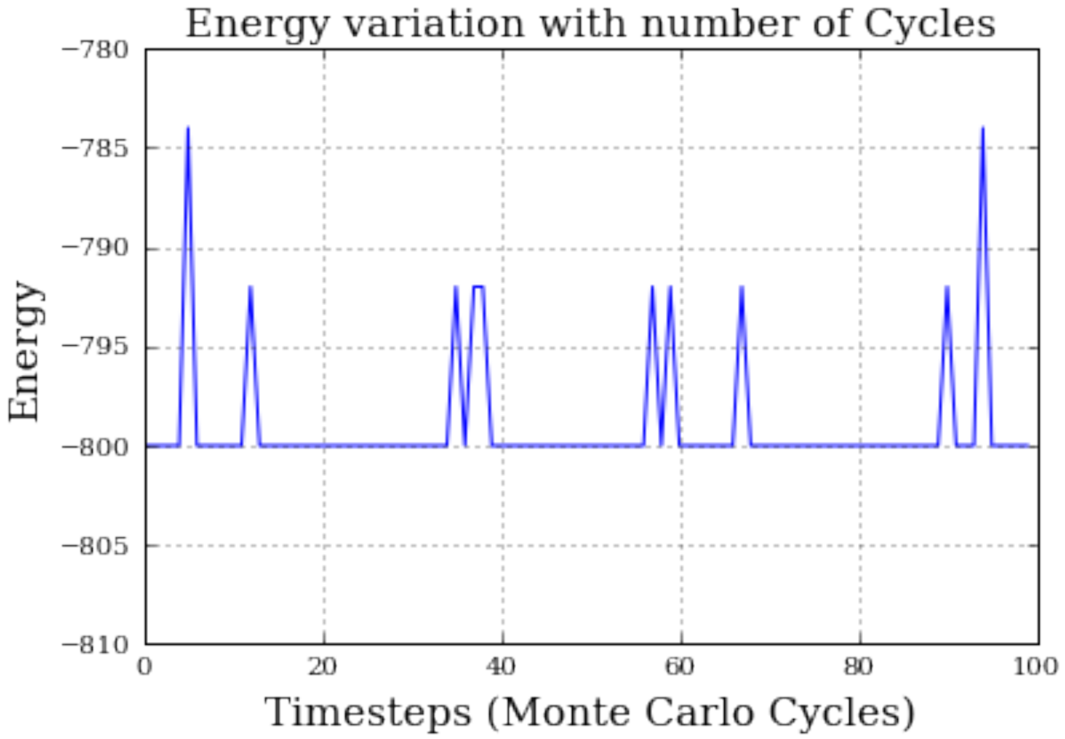


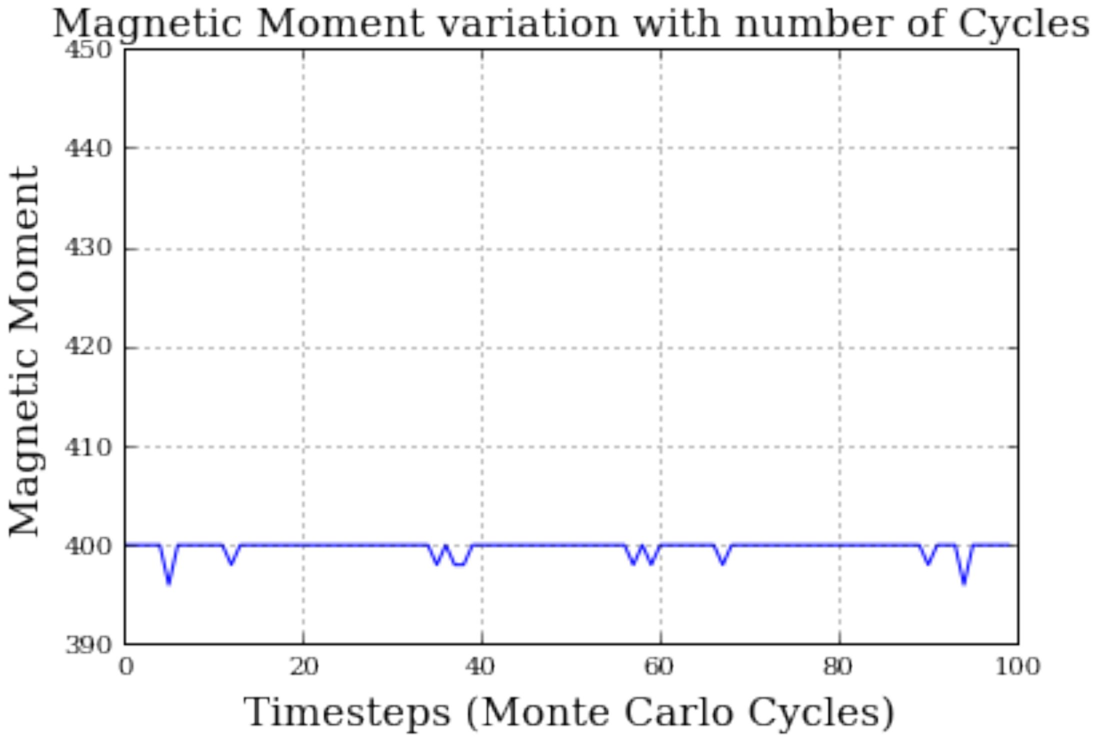



Magnetic Moment variation with number of Cycles

In [21]:

```
# In[10]:

E, M, t = np.loadtxt('N20T1steps100Order.txt', usecols=(0,1,2), unpack=True) #N2T1e


#N20T1steps100Order
plt.figure()
plt.plot(t, E)
plt.title("Energy variation with number of Cycles", size=15)
plt.ylabel("Energy", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
plt.ylim(-810, -780)
plt.grid()

plt.figure()
plt.plot(t, M)
plt.title("Magnetic Moment variation with number of Cycles", size=15)
plt.ylabel("Magnetic Moment", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
plt.ylim(390, 450)
plt.grid()
plt.show()


# In[32]:
```

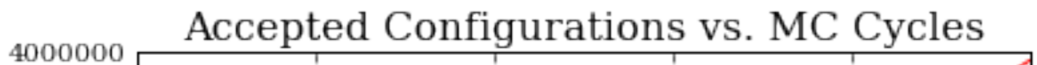


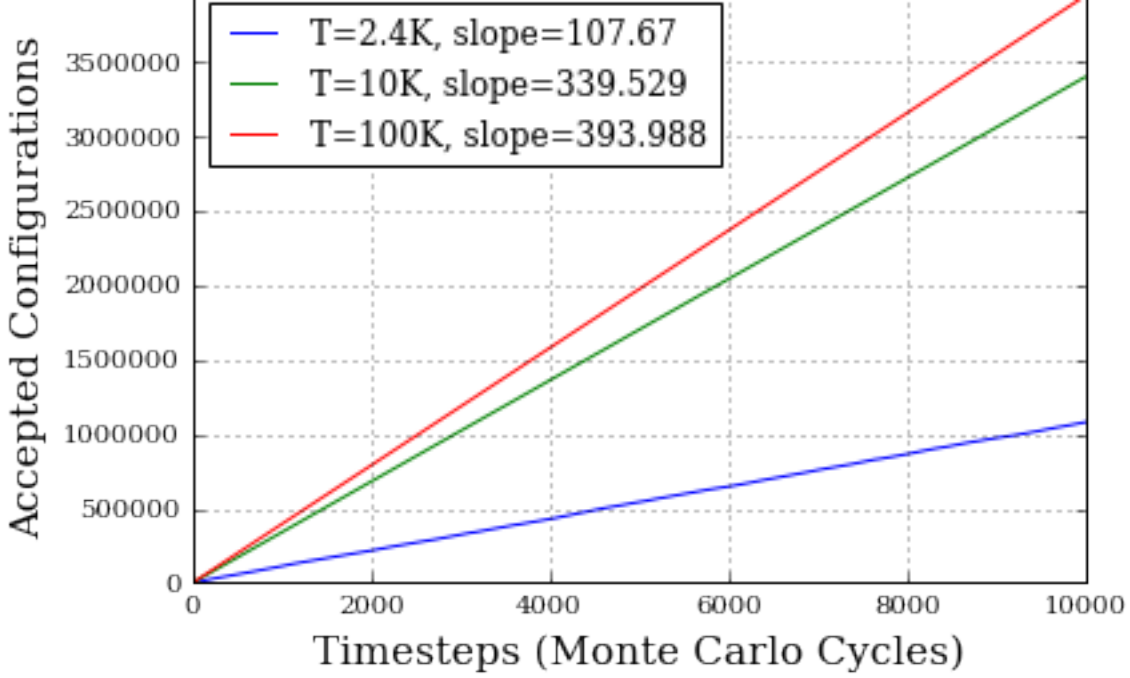


In [20]:

```
#accepted MCCycles

Accepted1, MCCycles = np.loadtxt('AcceptMCT24.txt', usecols=(0,1), unpack=True)

slope1, a,b,c,d = stats.linregress(MCCycles,Accepted1)



plt.figure(1)
plt.figure(1)
plt.plot(MCCycles, Accepted1, label=r'T=2.4K, slope=%g' %slope1)
plt.title("Accepted Configurations vs. MC Cycles", size=15)
plt.ylabel("Accepted Configurations", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
#plt.ylim(-850, -350)
plt.grid()




# In[33]:

Accepted2, MCCycles = np.loadtxt('AcceptMCT10.txt', usecols=(0,1), unpack=True)

slope2, a,b,c,d = stats.linregress(MCCycles,Accepted2)



plt.figure(1)
plt.plot(MCCycles, Accepted2, label=r'T=10K, slope=%g' %slope2)
plt.title("Accepted Configurations vs. MC Cycles", size=15)
plt.ylabel("Accepted Configurations", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
#plt.ylim(-850, -350)
plt.grid()


# In[34]:

Accepted3, MCCycles = np.loadtxt('AcceptMCT100.txt', usecols=(0,1), unpack=True)

slope3, a,b,c,d = stats.linregress(MCCycles,Accepted3)



plt.figure(1)
plt.plot(MCCycles, Accepted3, label=r'T=100K, slope=%g' %slope3)
plt.title("Accepted Configurations vs. MC Cycles", size=15)
plt.ylabel("Accepted Configurations", size=15)
plt.xlabel("Timesteps (Monte Carlo Cycles)", size=15)
#plt.ylim(-850, -350)
plt.grid()
plt.legend(loc='best')
```

```
Out[20]:

<matplotlib.legend.Legend at 0x110fdc208>
```

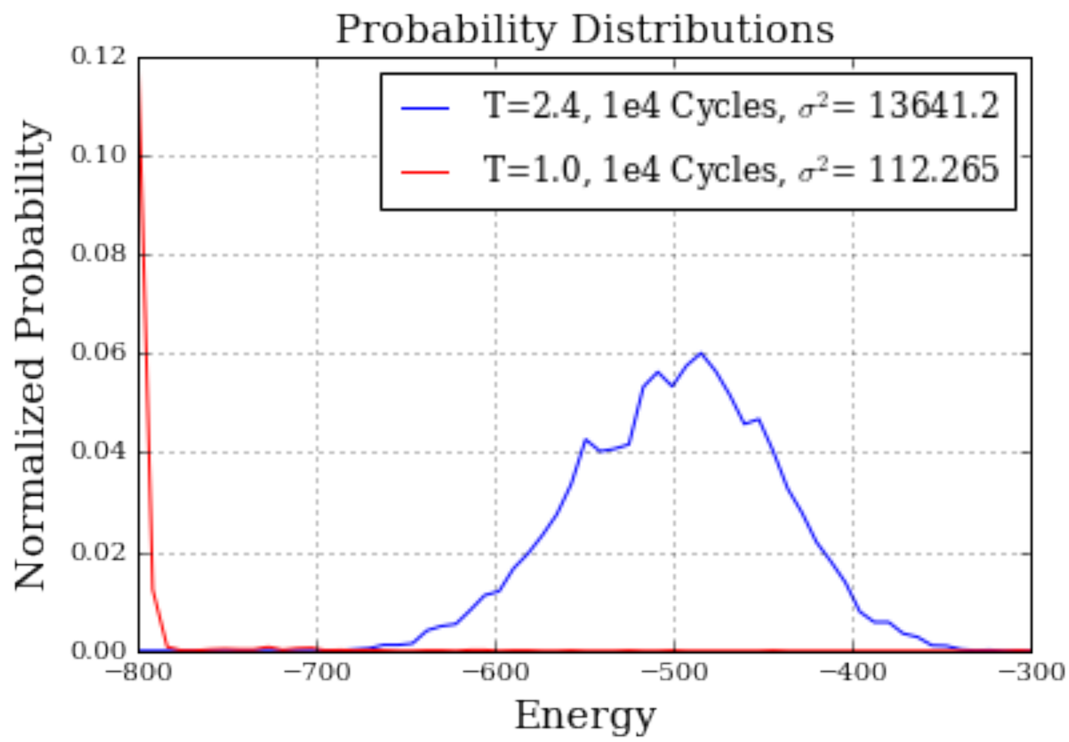




In [62]:

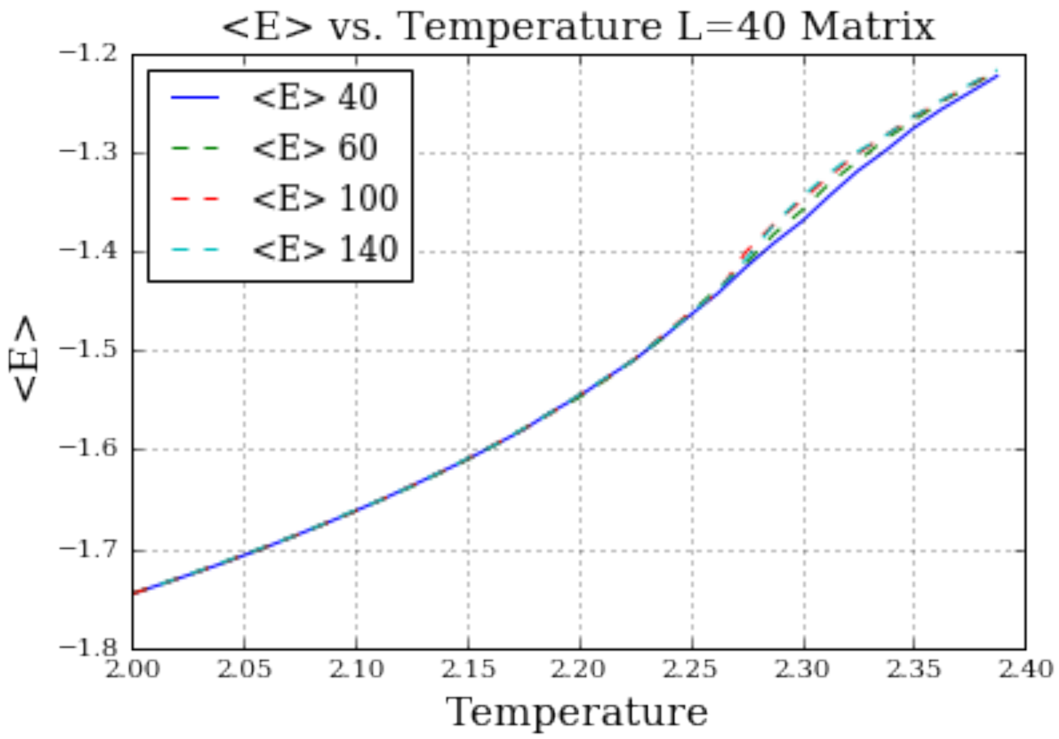
```
histval = np.loadtxt('hist2.dat', unpack=True)
histval1 = np.loadtxt('histT1.dat', unpack=True)


histval = histval[-1:0:-1]
histval1 = histval1[-1:0:-1]
N = 20
EE = np.linspace(-N*N*2,N*N*2,N*N*4/8-1)

xwhere=EE[where(histval > 0.0)]
xwhere2=EE[where(histval1 > 0.0)]

#var1=sqrt(mean(abs(xwhere - xwhere.mean())**2))
#var2=sqrt(mean(abs(xwhere2 - xwhere2.mean())**2))
var1= np.var(xwhere)
var12= np.var(xwhere2)




plt.xlim(-800,-300)
plt.figure(1)
plt.plot(EE,histval, label=r'T=2.4, 1e4 Cycles, $\sigma^2$= %g' %var1)
plt.title('Probability Distributions',size=15)
plt.ylabel('Normalized Probability',size=15)
plt.xlabel('Energy',size=15)

plt.figure(1)
plt.plot(EE,histval1, 'r',label='T=1.0, 1e4 Cycles, $\sigma^2$= %g' %var2)
plt.legend()
plt.grid()
plt.show()
```

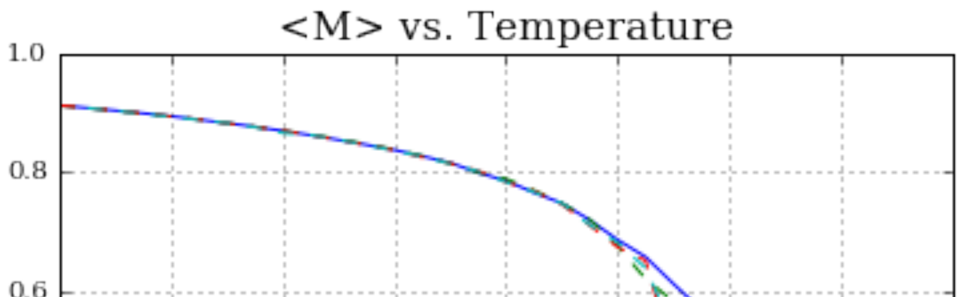
```
# need to plot the varience  ****
```

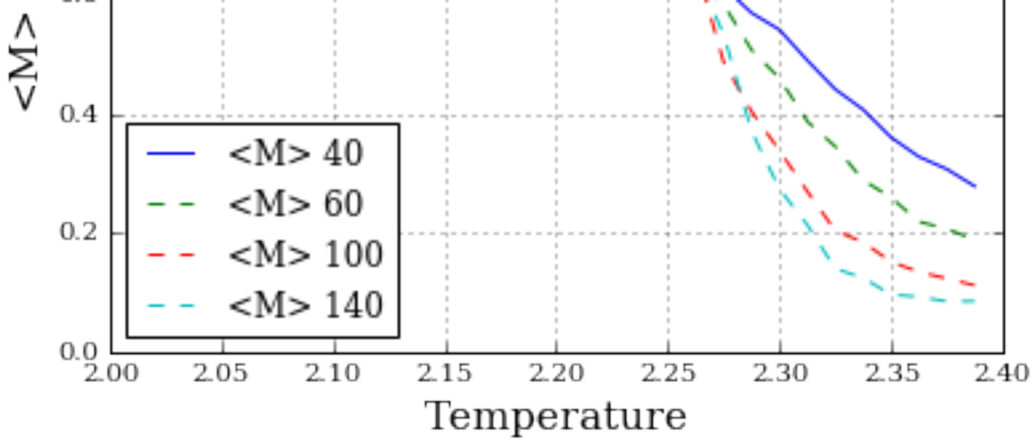


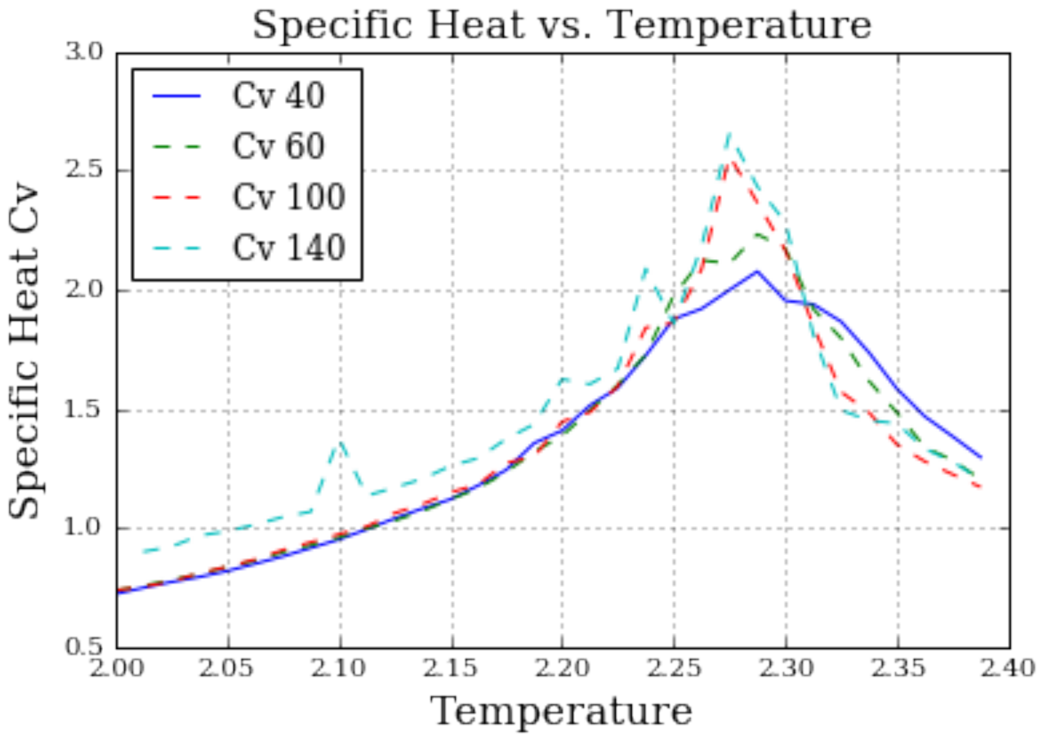
In [104]:

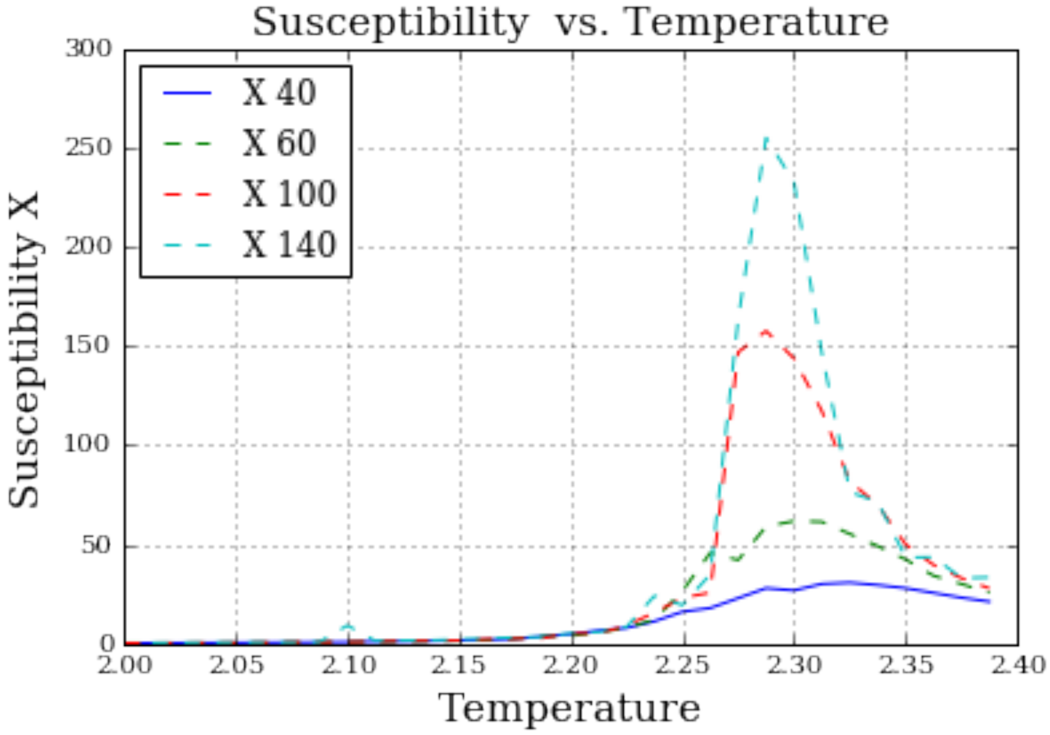
```
Temp40, Eex40, Mex40, Cv40, X40 = np.loadtxt('ALLVARS_100t2.txt', unpack=True) #32 
Temp60, Eex60, Mex60, Cv60, X60 = np.loadtxt('ALLVARS_100t60.txt', unpack=True) #32
Temp100, Eex100, Mex100, Cv100, X100 = np.loadtxt('ALLVARS_parallel.txt', unpack=Tru
Temp140, Eex140, Mex140, Cv140, X140 = np.loadtxt('ALLVARS_parallel_new_all.txt', un


N40= 40**2
N60=60**2
N100=100**2
N140=140**2

plt.figure(1)
plt.plot(Temp40,Eex40/N40, label='<E> 40')
plt.plot(Temp60,Eex60/N60, '--', label='<E> 60')
plt.plot(Temp100,Eex100/N100, '--', label='<E> 100')
plt.plot(Temp140,Eex140/N140, '--', label='<E> 140')
plt.grid()
plt.legend(loc='best')
plt.title('<E> vs. Temperature L=40 Matrix',size=15)
plt.ylabel('<E>',size=15)
plt.xlabel('Temperature',size=15)
plt.show()

plt.figure(2)
plt.plot(Temp40,Mex40/N40, label='<M> 40')
plt.plot(Temp60,Mex60/N60, '--', label='<M> 60')
plt.plot(Temp100,Mex100/N100, '--', label='<M> 100')
plt.plot(Temp140,Mex140/N140, '--', label='<M> 140')
plt.grid()
plt.title('<M> vs. Temperature',size=15)
plt.ylabel('<M>',size=15)
plt.xlabel('Temperature',size=15)
plt.legend(loc='best')
plt.show()

plt.figure(3)
plt.plot(Temp40,Cv40/N40, label='Cv 40')
plt.plot(Temp60,Cv60/N60, '--', label='Cv 60')
plt.plot(Temp100,Cv100/N100, '--', label='Cv 100')
plt.plot(Temp140,Cv140/N140, '--', label='Cv 140')
plt.title('Specific Heat vs. Temperature',size=15)
plt.ylabel('Specific Heat Cv',size=15)
plt.xlabel('Temperature',size=15)
plt.grid()
plt.legend(loc='best')
plt.show()

plt.figure(3)
plt.plot(Temp40,X40/N40,  label='X 40')
plt.plot(Temp60,X60/N60, '--', label='X 60')
plt.plot(Temp100,X100/N100, '--', label='X 100')
plt.plot(Temp140,X140/N140, '--', label='X 140')
plt.title('Susceptibility  vs. Temperature',size=15)
plt.ylabel('Susceptibility X',size=15)
plt.xlabel('Temperature',size=15)
plt.grid()
plt.legend(loc='best')
plt.show()
```






<M>
0.4
0.2
0.0
2.00
2.05
2.10
2.15
2.20
2.25
2.30
2.35
2.40
Temperature
<M> 40
<M> 60
<M> 100
<M> 140


Specific Heat vs. Temperature
Specific Heat Cv
3.0
2.5
2.0
1.5
1.0
0.5
2.00
2.05
2.10
2.15
2.20
2.25
2.30
2.35
2.40
Temperature
Cv 40
Cv 60
Cv 100
Cv 140


Susceptibility vs. Temperature
Susceptibility X
300
250
200
150
100
50
0
2.00
2.05
2.10
2.15
2.20
2.25
2.30
2.35
2.40
Temperature
X 40
X 60
X 100
X 140

```
In [ ]:
```

In [128]:

```
#\begin{equation}
 #   T_C(L)-T_C(L=\infty)=aL^{-1/\nu}
#\end{equation}

Tc140Cv=Temp[np.where(Cv140==max(Cv140))]
Tc100Cv=Temp[np.where(Cv100==max(Cv100))]
Tc60Cv=Temp[np.where(Cv60==max(Cv60))]
Tc40Cv=Temp[np.where(Cv40==max(Cv40))]


Tc100X=Temp[np.where(X140==max(X140))]
Tc100X=Temp[np.where(X100==max(X100))]
Tc60X=Temp[np.where(X60==max(X60))]
Tc40X=Temp[np.where(X40==max(X40))]

avtemp1=(Tc40Cv, Tc60Cv, Tc100Cv,Tc140Cv)

avtemp2=(Tc40X,Tc60X,  Tc100X,Tc100X)
print('Specific Heat Critical Temp =' + str(avtemp1))


avtempCV=sum(avtemp1)/len(avtemp1)
avtempX=sum(avtemp2)/len(avtemp2)


print(avtempCV)
print(avtempX)
```

```
Specific Heat Critical Temp =(array([ 2.2875]), array([ 2.2875]), arra
y([ 2.275]), array([ 2.2625]))
2.278125
2.3
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: