```python
average_heat_capacity = calculate_derivative_thermodynamic_average(energy,Ns,Ts,2,equilibrating_sweeps)


# In[ ]:


# Plot thermodynamic variables
N_indices = range(len(Ns))
T_indices = range(len(Ts))
plot_temperature_dependence(average_magnetisation,Ns,N_indices,Ts,T_indices,'Temperature','Average Magnetisation')
plot_temperature_dependence(average_energy,Ns,N_indices,Ts,T_indices,'Temperature','Average Energy')
plot_temperature_dependence(average_susceptibility,Ns,N_indices,Ts,T_indices,'Temperature','Average Susceptibility')
plot_temperature_dependence(average_heat_capacity,Ns,N_indices,Ts,T_indices,'Temperature','Average Heat Capacity')


# In[ ]:


# Fit magnetisation
N_indices = range(len(Ns))
T_indices = range(len(Ts))
parameters, errors = data_fitting(average_magnetisation,Ns,N_indices,Ts,T_indices,[0.,2.269,0.125])
print(parameters, errors)


# In[ ]:


# Plot fitted vs measured data
for N_index in N_indices:
    plt.plot(Ts[:],average_magnetisation[N_index,:],'-o',label='measured data')

    plt.plot(Ts[T_indices],shape_function(Ts[T_indices],parameters[N_index,0],parameters[N_index,1],parameters[N_index,2]),label='fitted data, N = {0}, Tc = {1:.3f}, beta = {2:.3f}'.format(Ns[N_index],parameters[N_index,1],parameters[N_index,2]))
    plt.legend(loc='best')
    plt.xlabel('Temperature')
    plt.ylabel('Average Magnetisation')
    plt.savefig('plots/Fitted vs Measured Magnetisation N = {0}.pdf'.format(Ns[N_index]))
    plt.figure()
```