

0.1 Simulation error of Wang-Landau results for black Statistical Images

```
1 import numpy as np
2 from scipy import interpolate, special
3 import os, h5py, hickle
4 import matplotlib.pyplot as plt
5 import pprint

1 import sys
2 if 'src' not in sys.path: sys.path.append('src')
3 import wanglandau as wl
4 from statistical_image import exact_bw_gs
5 import canonical_ensemble as canonical
```

0.1.1 The setup

```
1 datadir = 'data/black-images'
2 paths = [os.path.join(datadir, f) for f in os.listdir(datadir)]
3 len(paths)
```

1024

```
1 with h5py.File(paths[0], 'r') as f:
2     result = hickle.load(f)
3     imp = result['parameters']['system']['StatisticalImage']
4     N = len(imp['I0'])
5     M = imp['M']
6     Es = result['results']['Es'][:-1]

1 pprint.pprint(result['parameters'])

{'log': True,
 'simulation': {'eps': 1e-08,
               'flat_sweeps': 10000,
               'flatness': 0.2,
               'logf0': 1,
               'max_sweeps': 100000000},
 'system': {'StatisticalImage': {'I': array([10, 7, 1, 10, 6, 0, 0, 0, 28, 0, 7, 2, 1, 1, 1, 11]),
                                'I0': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]),
                                'M': 31}}})

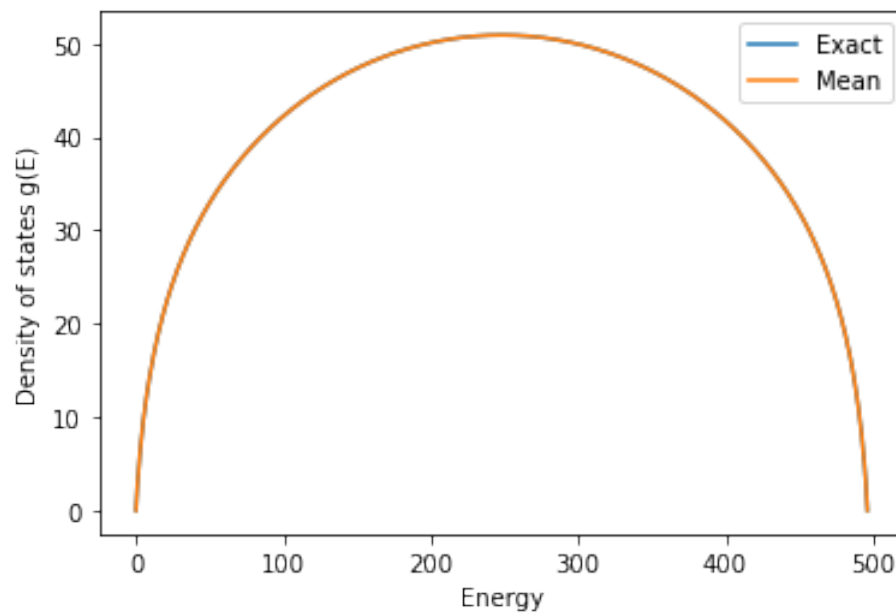
1 def file_lngs(path):
2     with h5py.File(path, 'r') as f:
3         result = hickle.load(f)
4         S = result['results']['S']
5         # Shift for computing exponentials
6         S -= min(S)
7         # Set according to the correct total number of states ((M+1)**N)
8         S += N*np.log(M+1) - np.log(np.sum(np.exp(S)))
9         # Set according to leftmost value
10        # S -= S[0]
11        return S
```

0.1.2 Error in the log density of states

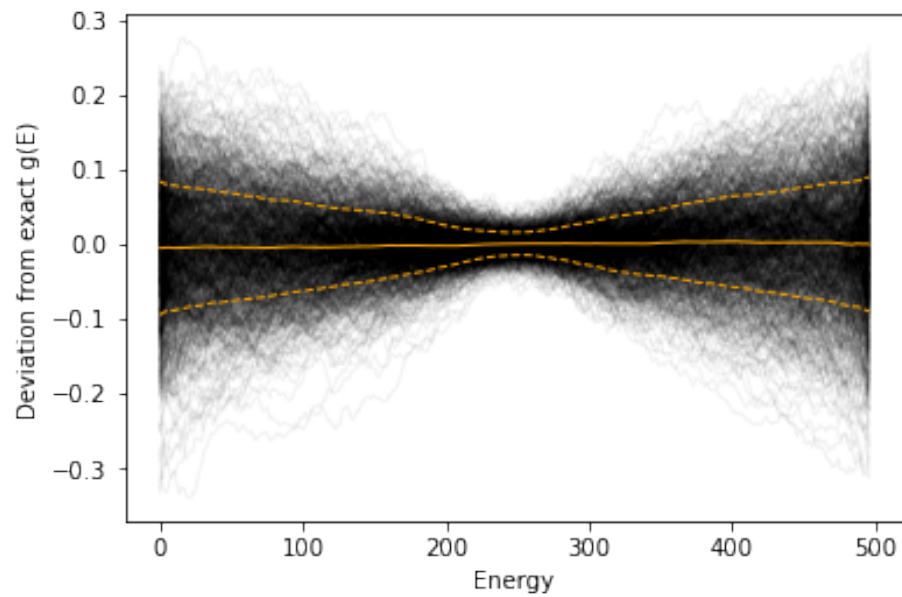
```
1 xEs, xgs = exact_bw_gs(N, M)
2 xlng = np.log(xgs)

1 mean_lng = np.zeros(len(Es))
2 std_lng = np.zeros(len(Es))
3 for lng in map(file_lngs, paths):
4     mean_lng += lng
5 mean_lng /= len(paths)
6 for lng in map(file_lngs, paths):
7     std_lng += (mean_lng - lng)**2
8 std_lng = np.sqrt(std_lng / (len(paths) - 1))

1 plt.plot(xEs, np.log(xgs), label='Exact')
2 plt.plot(Es, mean_lng, label='Mean')
3 plt.xlabel('Energy')
4 plt.ylabel('Density of states g(E)')
5 plt.legend();
```



```
1 for lng in map(file_lngs, paths):
2     plt.plot(Es, lng - xlng, 'black', alpha=0.05, linewidth=1)
3 plt.plot(Es, mean_lng - xlng, 'orange', linewidth=1)
4 plt.plot(Es, (mean_lng - std_lng) - xlng, 'orange', linestyle='dashed', linewidth=1)
5 plt.plot(Es, (mean_lng + std_lng) - xlng, 'orange', linestyle='dashed', linewidth=1)
6 plt.xlabel('Energy')
7 plt.ylabel('Deviation from exact g(E)');
```

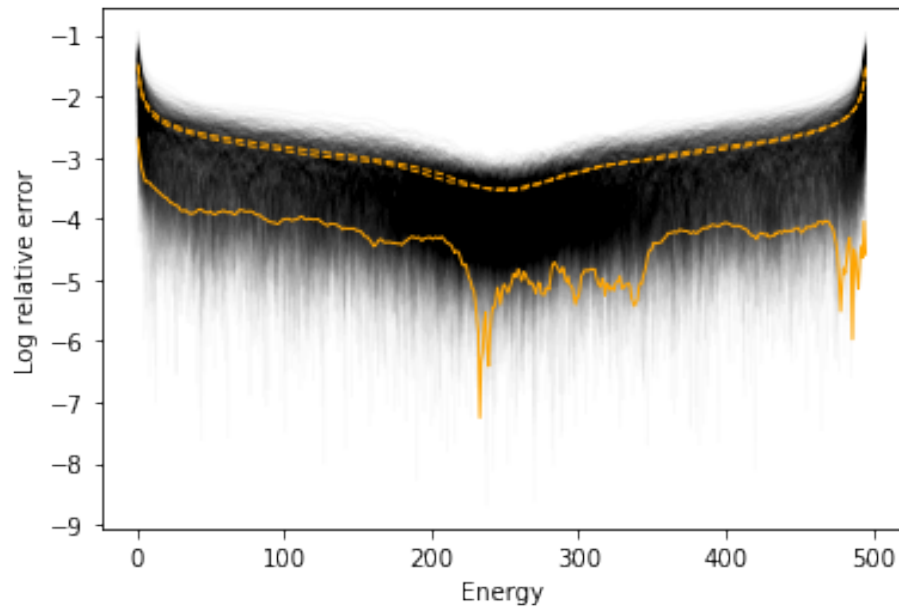


```

1 def relative_error(sim, exact):
2     if exact == 0.0:
3         return np.inf
4     else:
5         return np.abs(sim - exact) / exact
6 def log_relerror(sim, exact = xlng):
7     return np.log10(np.vectorize(relative_error)(sim, exact))

1 for lng in map(file_lngs, paths):
2     plt.plot(Es, log_relerror(lng), 'black', alpha=0.02, linewidth=1)
3     plt.plot(Es, log_relerror(mean_lng), 'orange', linewidth=1)
4     plt.plot(Es, log_relerror(mean_lng - std_lng), 'orange', linestyle='dashed', linewidth=1)
5     plt.plot(Es, log_relerror(mean_lng + std_lng), 'orange', linestyle='dashed', linewidth=1)
6     plt.xlabel('Energy')
7     plt.ylabel('Log relative error');

```



o.1.3 Error in canonical ensemble variables

```

1   $\beta$ s = [np.exp(k) for k in np.linspace(-7, 4, 500)]
2  exact_ens = canonical.Ensemble(Es, xlng, 'Exact')
3  mean_ens = canonical.Ensemble(Es, mean_lng, 'Mean WL')
4  mo_ens = canonical.Ensemble(Es, mean_lng, 'Mean -  $\sigma$  WL')
5  po_ens = canonical.Ensemble(Es, mean_lng, 'Mean +  $\sigma$  WL')

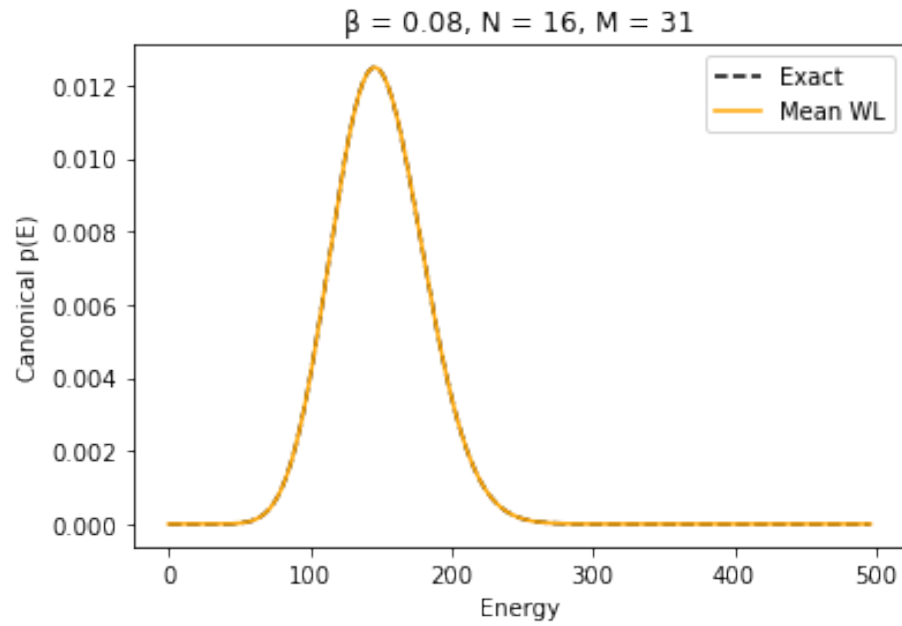
```

The canonical distribution for fixed β .

```

1   $\beta$ c = 8e-2
2  plt.plot(Es, exact_ens.p( $\beta$ c), 'black', label=exact_ens.name, linestyle='dashed')
3  plt.plot(Es, mean_ens.p( $\beta$ c), 'orange', label=mean_ens.name)
4  plt.title('  $\beta$  = {}, N = {}, M = {}'.format( $\beta$ c, N, M))
5  plt.xlabel("Energy")
6  plt.ylabel("Canonical p(E)")
7  plt.legend();

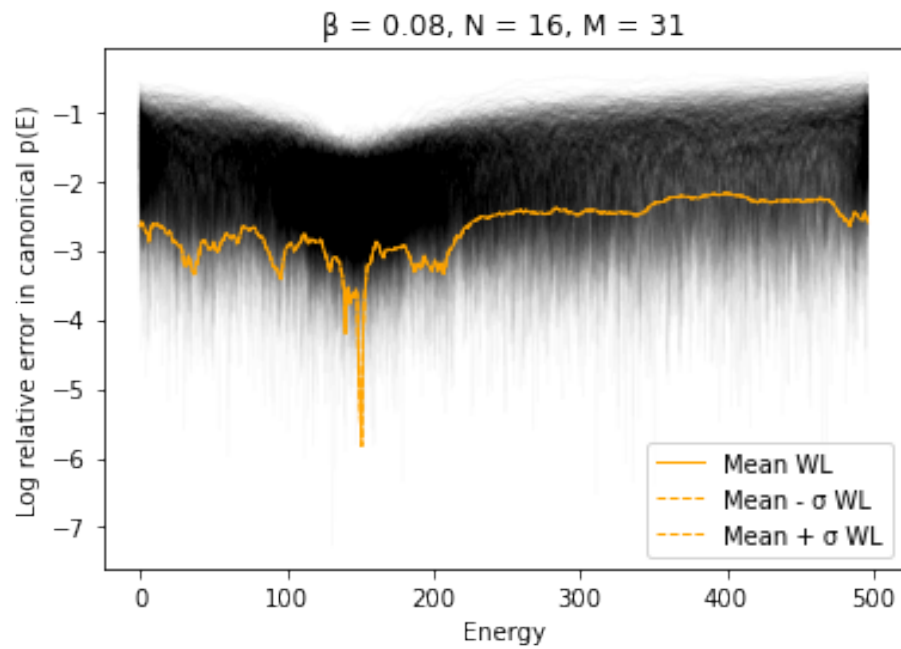
```



```

1  for lng in map(file_lngs, paths):
2      ens = canonical.Ensemble(Es, lng)
3      plt.plot(Es, log_relderror(ens.p( $\beta$ c), exact_ens.p( $\beta$ c)),
4               'black', alpha=0.02, linewidth=1)
5      plt.plot(Es, log_relderror(mean_ens.p( $\beta$ c), exact_ens.p( $\beta$ c)),
6               'orange', linewidth=1, label=mean_ens.name)
7      plt.plot(Es, log_relderror(mo_ens.p( $\beta$ c), exact_ens.p( $\beta$ c)),
8               'orange', linewidth=1, linestyle='dashed', label=mo_ens.name)
9      plt.plot(Es, log_relderror(po_ens.p( $\beta$ c), exact_ens.p( $\beta$ c)),
10             'orange', linewidth=1, linestyle='dashed', label=po_ens.name)
11  plt.title('  $\beta = \{ \}$ ,  $N = \{ \}$ ,  $M = \{ \}$ '.format( $\beta$ c, N, M))
12  plt.xlabel("Energy")
13  plt.ylabel("Log relative error in canonical  $p(E)$ ")
14  plt.legend();

```

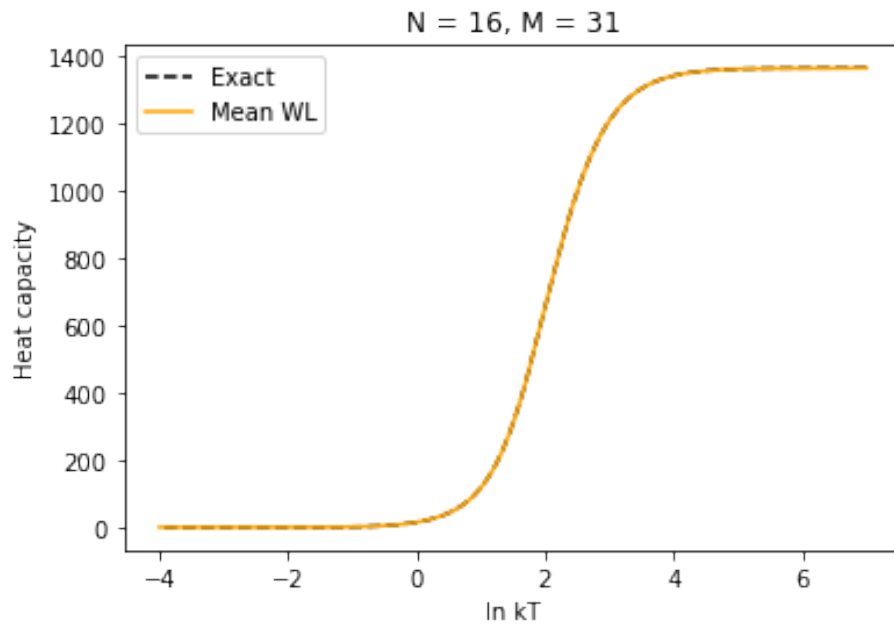


The relative error in the heat capacity provides a stringent test of the results.

```

1 plt.plot(-np.log(βs), exact_ens.heat_capacity(βs), 'black', label=exact_ens.name, linestyle='dashed')
2 plt.plot(-np.log(βs), mean_ens.heat_capacity(βs), 'orange', label=mean_ens.name)
3 plt.xlabel("ln kT")
4 plt.ylabel("Heat capacity")
5 plt.title('N = {}, M = {}'.format(N, M))
6 plt.legend();

```



```

1  for lng in map(file_lngs, paths):
2      ens = canonical.Ensemble(Es, lng)
3      plt.plot(-np.log( $\beta$ s), log_relererror(ens.heat_capacity( $\beta$ s), exact_ens.heat_capacity( $\beta$ s)),
4              'black', alpha=0.02, linewidth=1)
5      plt.plot(-np.log( $\beta$ s), log_relererror(mean_ens.heat_capacity( $\beta$ s), exact_ens.heat_capacity( $\beta$ s)),
6              'orange', label=mean_ens.name)
7      plt.plot(-np.log( $\beta$ s), log_relererror(mo_ens.heat_capacity( $\beta$ s), exact_ens.heat_capacity( $\beta$ s)),
8              'orange', linestyle='dashed', label=mo_ens.name)
9      plt.plot(-np.log( $\beta$ s), log_relererror(po_ens.heat_capacity( $\beta$ s), exact_ens.heat_capacity( $\beta$ s)),
10             'orange', linestyle='dashed', label=po_ens.name)
11  plt.xlabel("ln kT")
12  plt.ylabel("Log relative error in heat capacity")
13  plt.title('N = {}, M = {}'.format(N, M))
14  plt.legend();

```

