plots

October 21, 2024

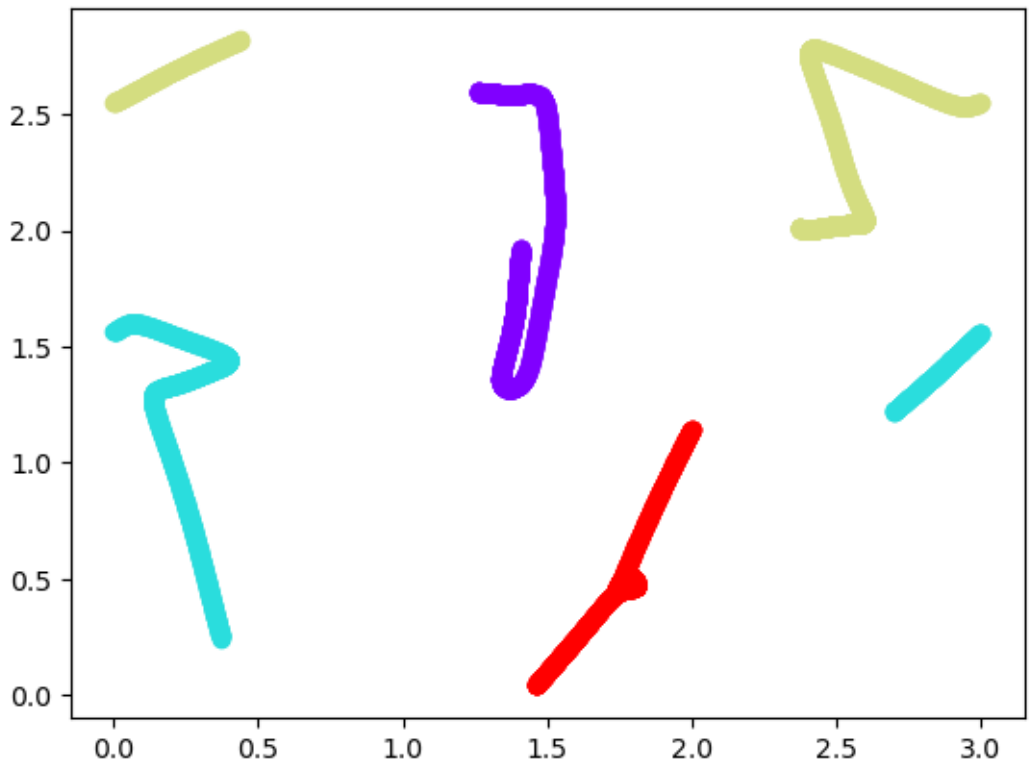
[37]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm
import seaborn as sns
```

1 Measurements

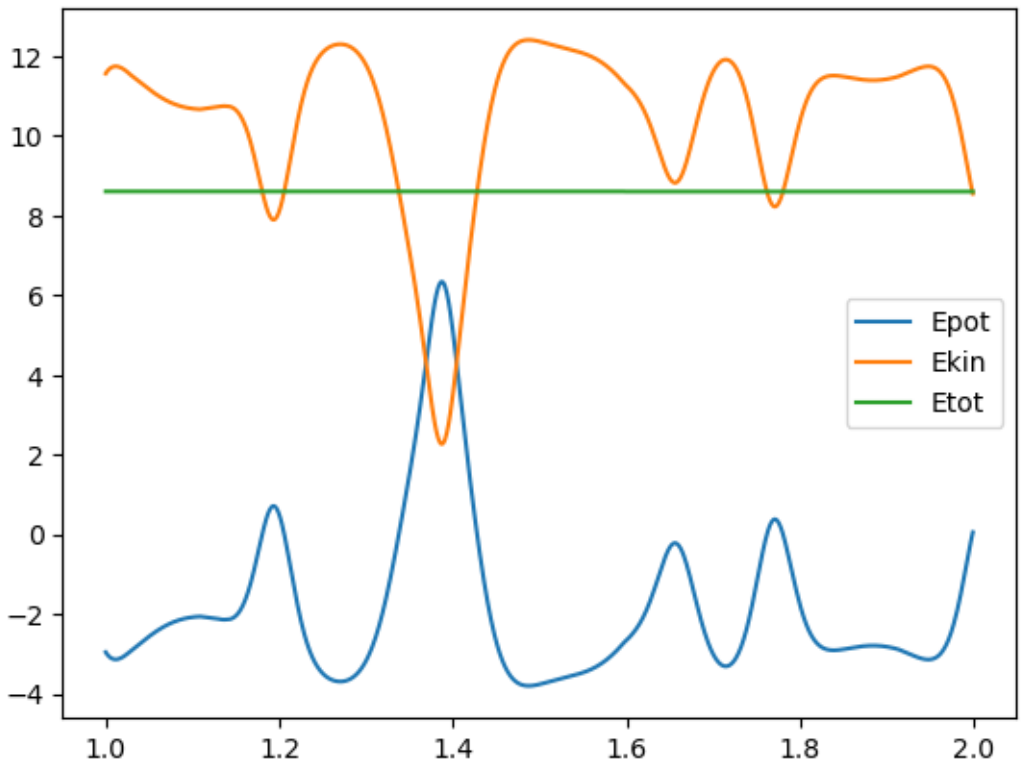
1.1 Liquid Regime

Below we see the results of a simulation of a 2x2 grid of 4 particles, given one second of thermalization time and one second of simulation time. The particles behave like a fluid in this regime, and the total energy is roughly conserved.

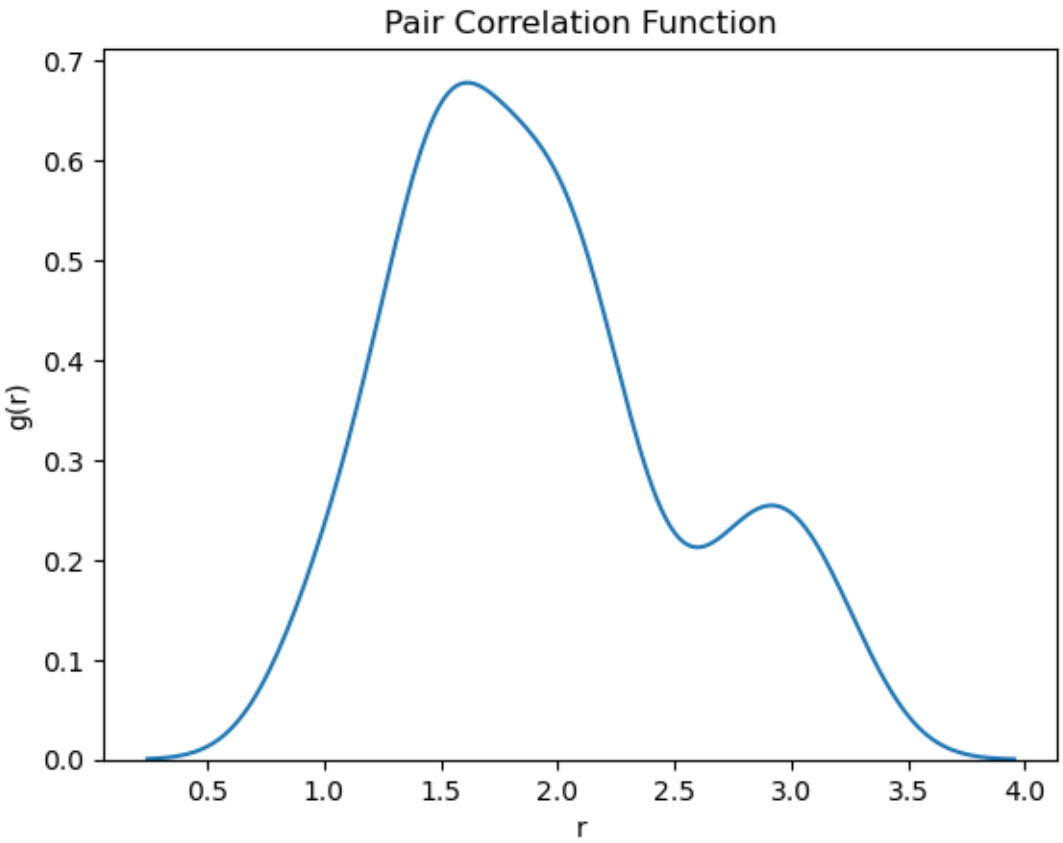
[5]:
```
df_pos_small = pd.read_csv("data/positions_small.csv")

N = len(df_pos_small.columns) // 2  # assuming x and y coordinates for each␣
 ↪particle
colors = cm.rainbow(np.linspace(0, 1, N))

for i in range(N):
    plt.scatter(df_pos_small[f'x{i}'][1000:2000], df_pos_small[f'y{i}'][1000:
 ↪2000], c=[colors[i]])

plt.show()
```



[6]:
```
df_energy_small = pd.read_csv("data/verification_small.csv")

plt.plot(df_energy_small['step'][1000:2000]*0.001, df_energy_small['Epot'][1000:
  ↪2000], label='Epot')
plt.plot(df_energy_small['step'][1000:2000]*0.001, df_energy_small['Ekin'][1000:
  ↪2000], label='Ekin')
plt.plot(df_energy_small['step'][1000:2000]*0.001, df_energy_small['Etot'][1000:
  ↪2000], label='Etot')
plt.legend()
plt.show()
```



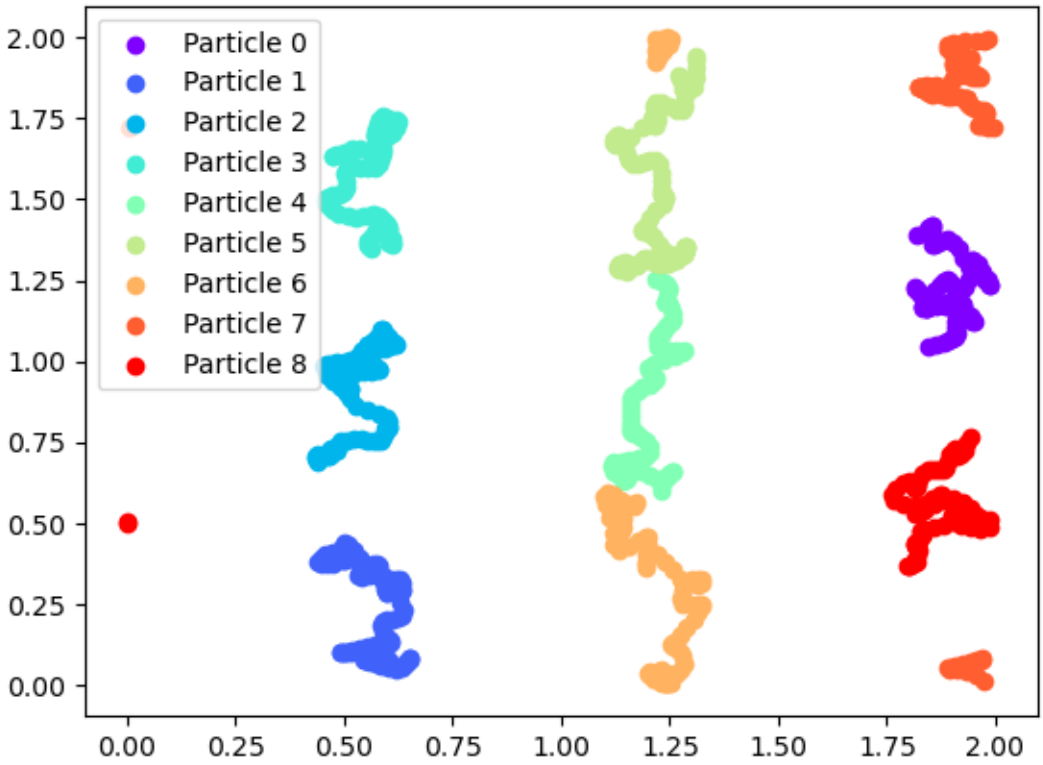
[41]:
```
dists = []
for time in range(20):
    locs = np.array(df_pos_small.iloc[50*time])
    for i in range(1,N):
        for j in range(i):
            dists.append(np.sqrt((locs[2*i] - locs[2*j])**2 + (locs[2*i+1] -␣
 ↪locs[2*j+1])**2))
sns.kdeplot(dists)
plt.xlabel("r")
plt.ylabel("g(r)")
plt.title("Pair Correlation Function")
plt.show()
```



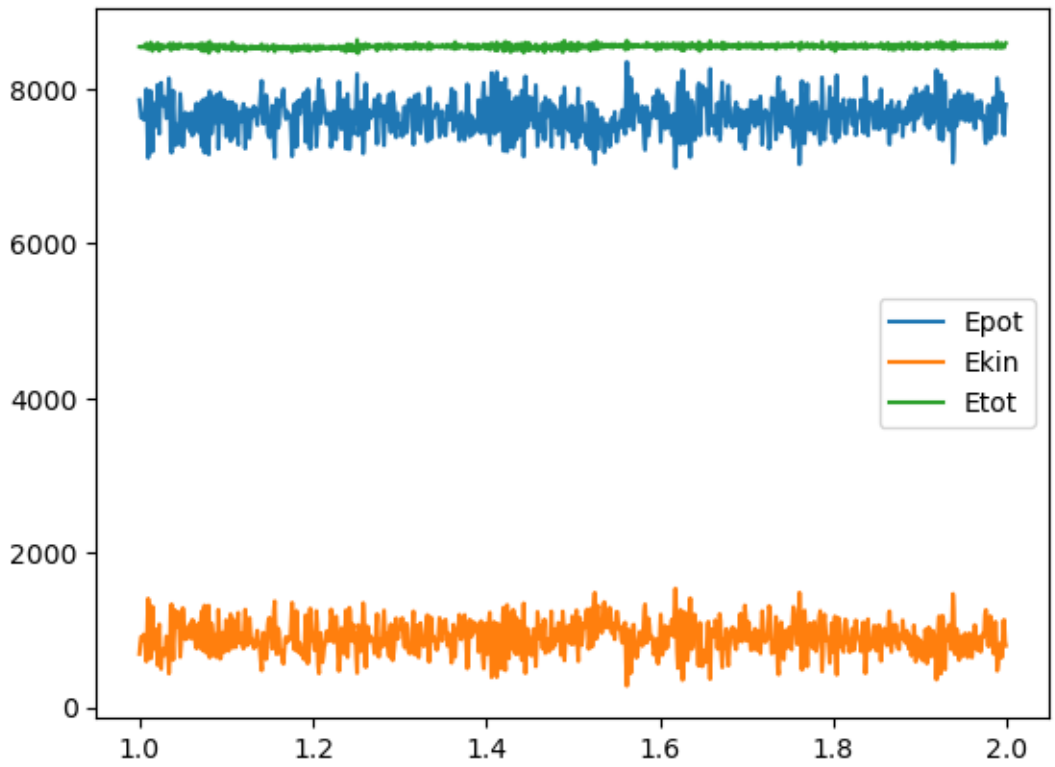
1.2 Solid Regime

Below we see the results of a simulation of a 2x2 grid of 9 particles, given one second of thermalization time and one second of simulation time. The particles behave like a solid in this regime, and once again the total energy is roughly conserved.

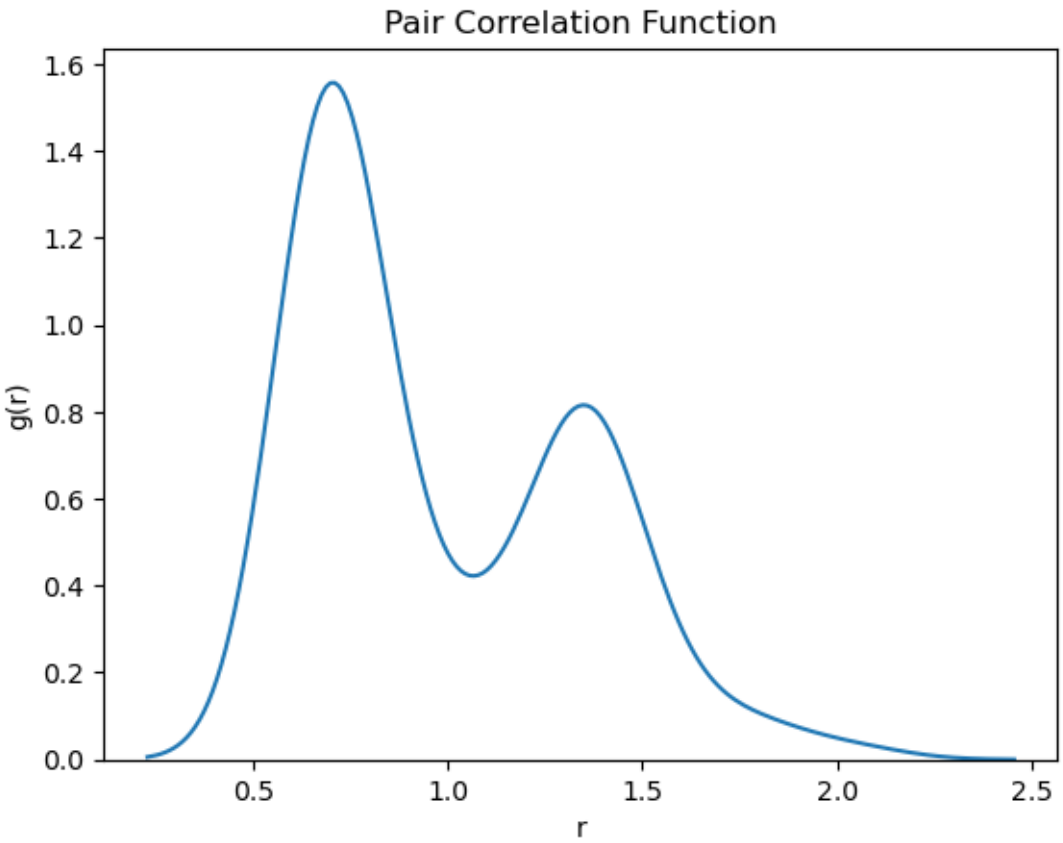
[3]:
```
df_pos_solid = pd.read_csv("data/positions_solid.csv")

N = len(df_pos_solid.columns) // 2  # assuming x and y coordinates for each␣
↪particle
colors = cm.rainbow(np.linspace(0, 1, N))

for i in range(N):
    plt.scatter(df_pos_solid[f'x{i}'][1000:1100], df_pos_solid[f'y{i}'][1000:
↪1100], c=[colors[i]], label=f'Particle {i}')

plt.legend(loc="best")
plt.show()
```



[4]:
```
df_energy_solid = pd.read_csv("data/verification_solid.csv")

plt.plot(df_energy_solid['step'][1000:2000]*0.001, df_energy_solid['Epot'][1000:
  ↪2000], label='Epot')
plt.plot(df_energy_solid['step'][1000:2000]*0.001, df_energy_solid['Ekin'][1000:
  ↪2000], label='Ekin')
plt.plot(df_energy_solid['step'][1000:2000]*0.001, df_energy_solid['Etot'][1000:
  ↪2000], label='Etot')
plt.legend()
plt.show()
```



[42]:
```
dists = []
for time in range(20):
    locs = np.array(df_pos_solid.iloc[50*time])
    for i in range(1,N):
        for j in range(i):
            dists.append(np.sqrt((locs[2*i] - locs[2*j])**2 + (locs[2*i+1] -␣
 ↪locs[2*j+1])**2))
sns.kdeplot(dists)
plt.xlabel("r")
plt.ylabel("g(r)")
plt.title("Pair Correlation Function")
plt.show()
```


Pair Correlation Function
g(r)
r
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0.5
1.0
1.5
2.0
2.5