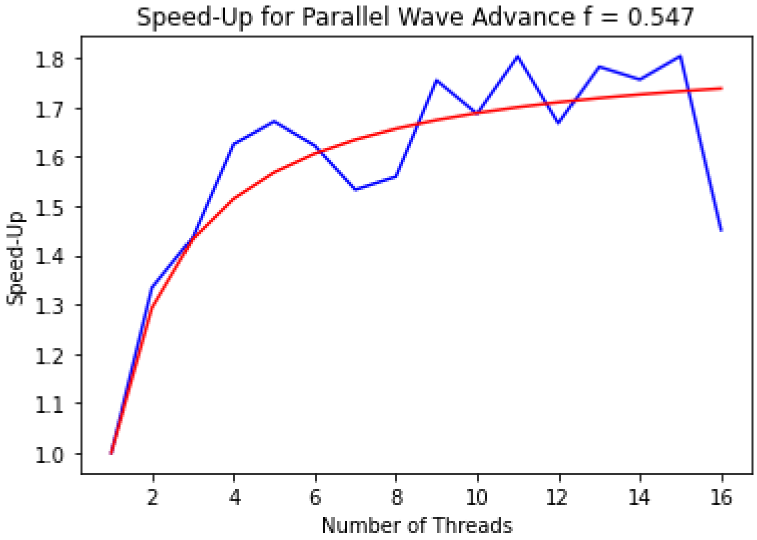
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit


def func(x, f):
  return 1/(f+(1-f)/x)


p_data = np.arange(1,17)
t_adv_data = np.array([
18.671,
13.996,
13,
11.486,
11.172,
11.506,
12.179,
11.971,
10.64,
11.068,
10.354,
11.195,
10.476,
10.631,
10.35,
12.865,
])
t_out_data = np.array([
19.724,
18.021,
17.500,
17.417,
17.218,
17.331,
17.454,
17.086,
17.239,
16.768,
17.046,
17.183,
18.099,
17.593,
18.048,
17.844,
])


s_adv_data = [t_adv_data[0]/a for a in t_adv_data]
s_out_data = [t_out_data[0]/a for a in t_out_data]
```

```
popt, _ = curve_fit(func, p_data, s_adv_data)
plt.figure()
plt.plot(p_data, s_adv_data, 'b-', label='data')
plt.plot(p_data, func(p_data, *popt), 'r-')
plt.title("Speed-Up for Parallel Wave Advance f = %5.3f" %(popt))
plt.xlabel('Number of Threads')
plt.ylabel('Speed-Up')
plt.show()
```



```
popt2, _ = curve_fit(func, p_data, s_out_data)
plt.figure()
plt.plot(p_data, s_out_data, 'b-', label='data')
plt.plot(p_data, func(p_data, *popt2), 'r-')
plt.title("Speed-Up for Parallel Wave Output f = %5.3f" %(popt2))
plt.xlabel('Number of Threads')
plt.ylabel('Speed-Up')
plt.show()
```

✓ 0s completed at 2:35 AM ● ×