In [1]:
```
import matplotlib.pylab as plt   # That gives plotting, and the next line makes plots a
%matplotlib inline
plt.rcParams['figure.dpi']=300

import numpy as np  # That gives numerical arrays and tools for manipulating them
```

Define tuning curves

In [2]:
```
def gaussian(mu,sigma, x):
    maxrate = 300 # max firing rate
    f = maxrate*np.exp(-0.5*((x-mu)/sigma)**2)
    return f

def cockroach_tuning(stim_dir, cell_num):

    if cell_num == 1:
        mu = 45
        sigma = 5
        f = gaussian(mu, sigma, stim_dir)
    elif cell_num == 2:
        mu = 45
        sigma = 10
        f = gaussian(mu, sigma, stim_dir)
    else:
        mu = 30
        sigma = 10
        f1 = gaussian(mu, sigma, stim_dir)
        mu = 60
        f2 = gaussian(mu, sigma, stim_dir)
        f = (f1 + f2)

    return f
```

Enter stimulus direction (in degrees) and cell number here

In [69]:
```
stimDir=45
cell_num=1
```

Enter number of trials (ntrials), and duration of each trial (nmsec) here

In [70]:
```
ntrials=100
nmsec = 300    # number of milliseconds to record for
```

Generate spike train

In [71]:

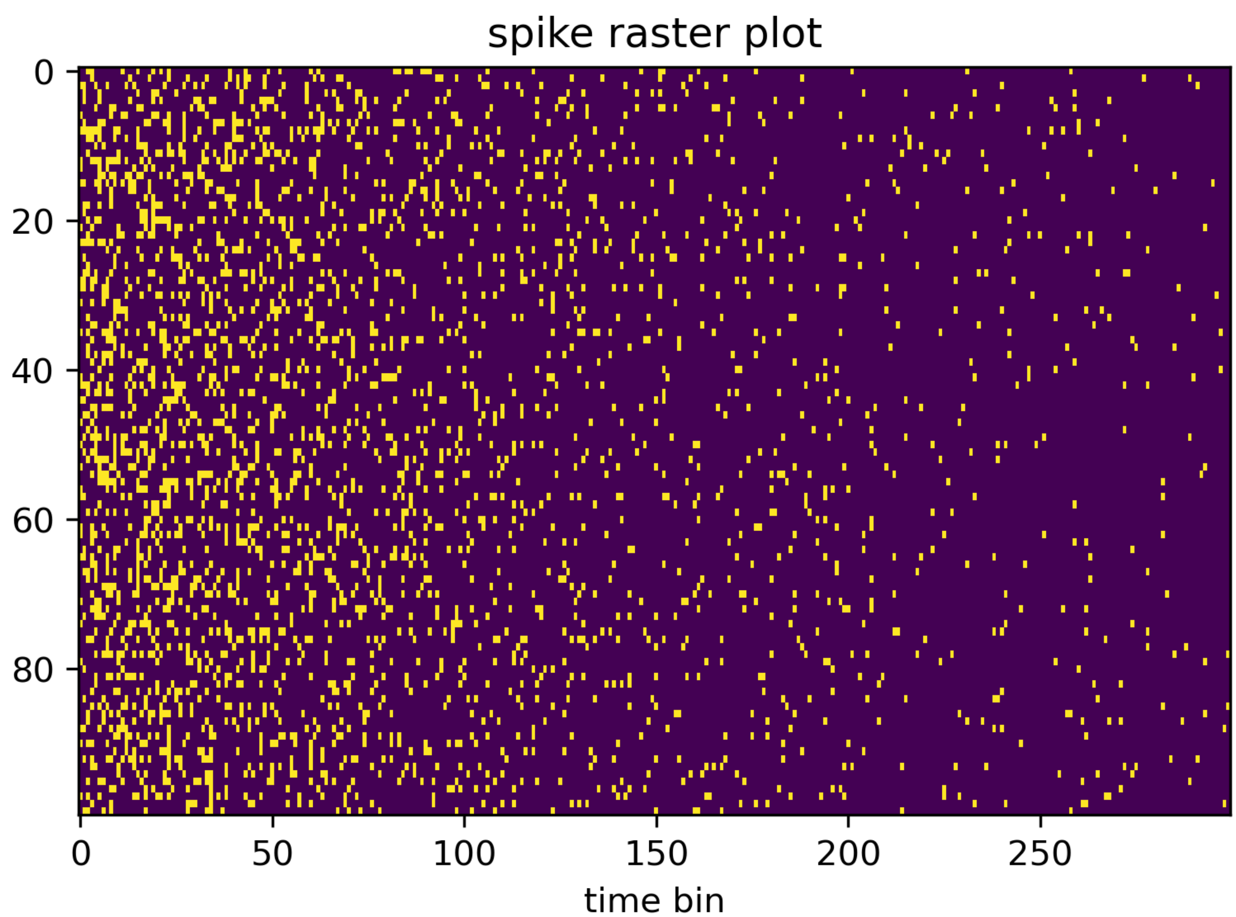
```
times=np.arange(nmsec) #array of time points (1 msec apart)
spiketrain = np.zeros((ntrials,nmsec))      # set up output data

rate = cockroach_tuning(stimDir, cell_num)  #returns rate, in Hz.
tau = 100      # adaptation time constant in msec
delta_t=0.001  #time bin, in seconds (1 msec)
ratelist = rate*np.exp(-times/tau)  # list of adapting rates


rng = np.random.default_rng()

for j in np.arange(ntrials):
    for i in np.arange(nmsec):
            spiketrain[j,i] = np.round(rng.uniform(0,1,1) + ratelist[i]*delta_t -1/2
```

In [72]:

```
plt.imshow(spiketrain,aspect='auto')
plt.title('spike raster plot')
plt.xlabel('time bin')
```

Out[72]:

```
Text(0.5, 0, 'time bin')
```



In [73]:

```
T = nmsec / 1000
rate_per_trial=1/T * np.sum(spiketrain,axis=1)

mean_rate_per_trial = np.mean(rate_per_trial)
print('mean_rate_per_trial=',mean_rate_per_trial)
```

```
std_dev_rate_per_trial = np.std(rate_per_trial)
print('std_dev_rate_per_trial=',std_dev_rate_per_trial)
```

```
mean_rate_per_trial= 95.83333333333334
std_dev_rate_per_trial= 16.602376001310443
```

For cell 1 at 45°, each row represents a single trial, so there are 100 rows in total corresponding to the 100 trials we run here, and every yellow bar shows the spikes during the 300 milliseconds timespan. The spiketrain is generally denser when we just begin the trials. The general pattern across 100 trials are pretty consistent and similar, which indicates that we probably have enough trials.

The mean rate per trial of 95.833 and the standard deviation of 16.602. Considering the standard deviation is much smaller than the mean, we believe that we have enough trials here.

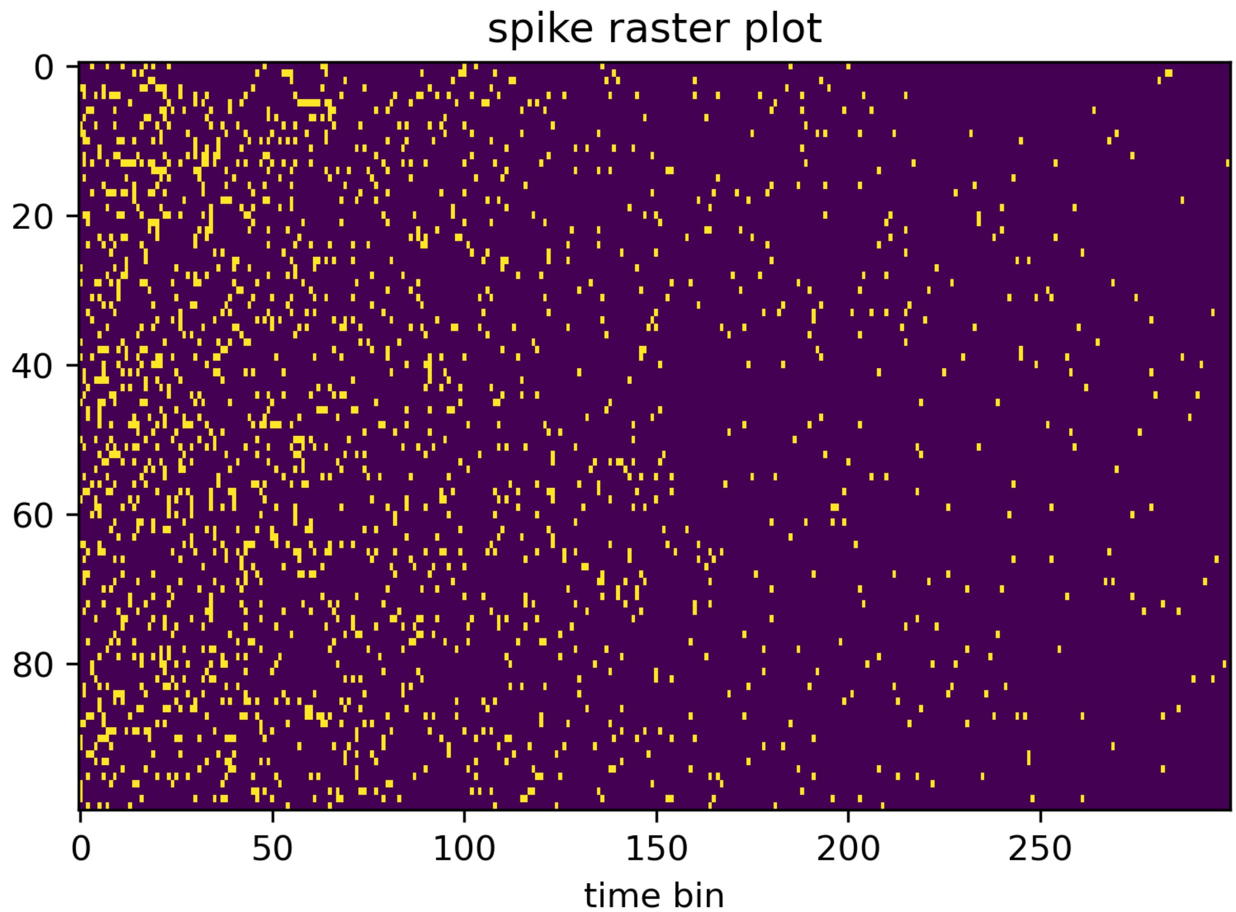
Cell 1 50°

In [75]:

```
stimDir=50
times=np.arange(nmsec) #array of time points (1 msec apart)
spiketrain = np.zeros((ntrials,nmsec))      # set up output data

rate = cockroach_tuning(stimDir, cell_num)  #returns rate, in Hz.
tau = 100      # adaptation time constant in msec
delta_t=0.001  #time bin, in seconds (1 msec)
ratelist = rate*np.exp(-times/tau)  # list of adapting rates


rng = np.random.default_rng()

for j in np.arange(ntrials):
    for i in np.arange(nmsec):
            spiketrain[j,i] = np.round(rng.uniform(0,1,1) + ratelist[i]*delta_t -1/2

plt.imshow(spiketrain,aspect='auto')
plt.title('spike raster plot')
plt.xlabel('time bin')
```

Out[75]:

```
Text(0.5, 0, 'time bin')
```



In [76]:

```
T = nmsec / 1000
rate_per_trial=1/T * np.sum(spiketrain,axis=1)

mean_rate_per_trial = np.mean(rate_per_trial)
print('mean_rate_per_trial=',mean_rate_per_trial)

std_dev_rate_per_trial = np.std(rate_per_trial)
print('std_dev_rate_per_trial=',std_dev_rate_per_trial)
```

```
mean_rate_per_trial= 59.833333333333333
std_dev_rate_per_trial= 14.70732999410687
```

For cell 1 at 50°, again each row represents a single trial, so there are 100 rows in total corresponding to the 100 trials we run here, and every yellow bar shows the spikes during the 300 milliseconds timespan. The plot is overall similar to the previous one.

The mean rate per trial of 59.833 and the standard deviation of 16.707. Considering the standard deviation is much smaller than the mean, we believe that we have enough trials here.

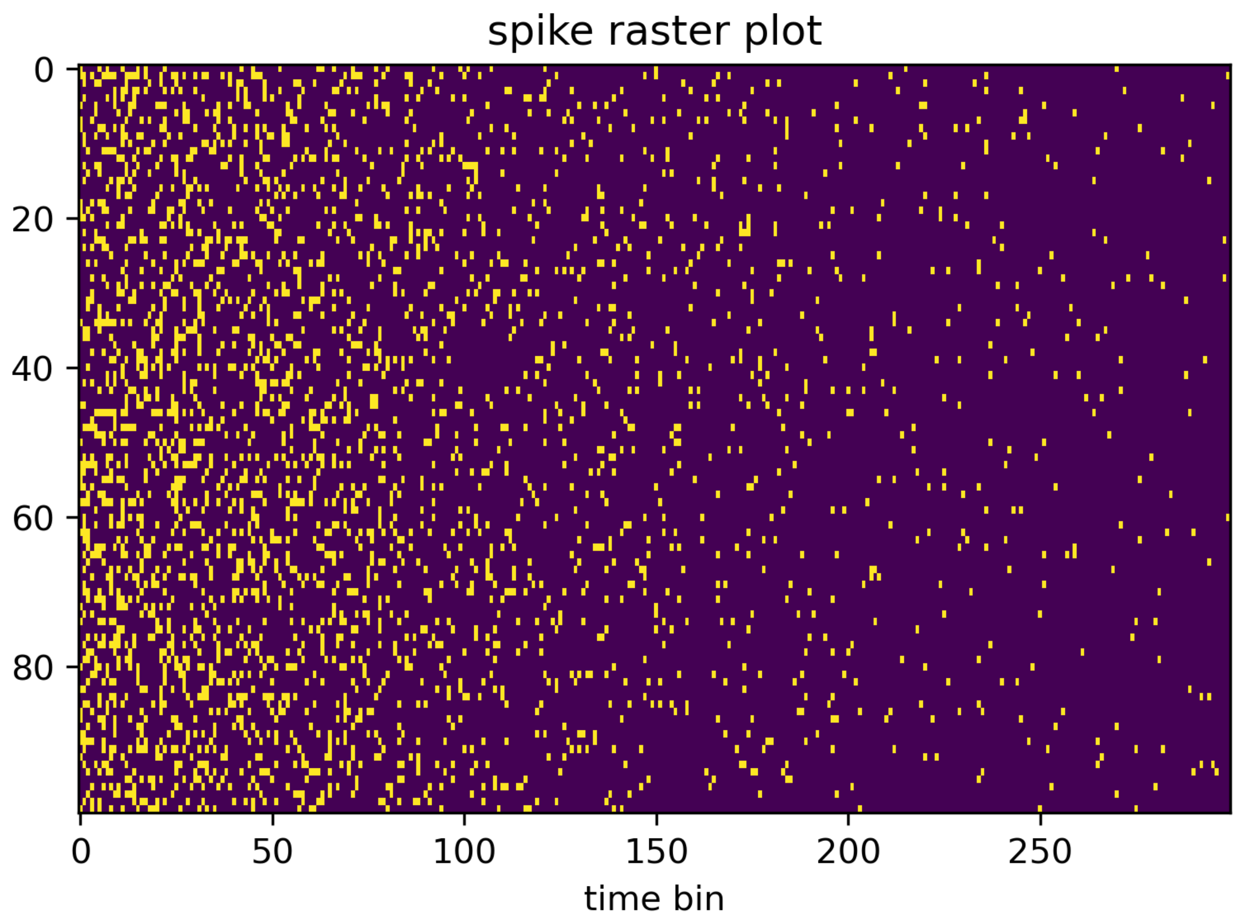
Cell 2 45°

In [77]:

```
stimDir=45
cell_num=2
times=np.arange(nmsec) #array of time points (1 msec apart)
spiketrain = np.zeros((ntrials,nmsec))      # set up output data
```

```
rate = cockroach_tuning(stimDir, cell_num)  #returns rate, in Hz.
tau = 100      # adaptation time constant in msec
delta_t=0.001  #time bin, in seconds (1 msec)
ratelist = rate*np.exp(-times/tau)  # list of adapting rates


rng = np.random.default_rng()

for j in np.arange(ntrials):
    for i in np.arange(nmsec):
            spiketrain[j,i] = np.round(rng.uniform(0,1,1) + ratelist[i]*delta_t -1/2

plt.imshow(spiketrain,aspect='auto')
plt.title('spike raster plot')
plt.xlabel('time bin')
```

Out[77]: Text(0.5, 0, 'time bin')



In [78]:
```
T = nmsec / 1000
rate_per_trial=1/T * np.sum(spiketrain,axis=1)

mean_rate_per_trial = np.mean(rate_per_trial)
print('mean_rate_per_trial=',mean_rate_per_trial)

std_dev_rate_per_trial = np.std(rate_per_trial)
print('std_dev_rate_per_trial=',std_dev_rate_per_trial)
```

mean_rate_per_trial= 94.8

```
std_dev_rate_per_trial= 16.682126163452107
```

For cell 2 at 45°, again each row represents a single trial, so there are 100 rows in total corresponding to the 100 trials we run here, and every yellow bar shows the spikes during the 300 milliseconds timespan. The plot is overall similar to the previous one.

The mean rate per trial of 94.8 and the standard deviation of 16.682. Considering the standard deviation is much smaller than the mean, we believe that we have enough trials here.

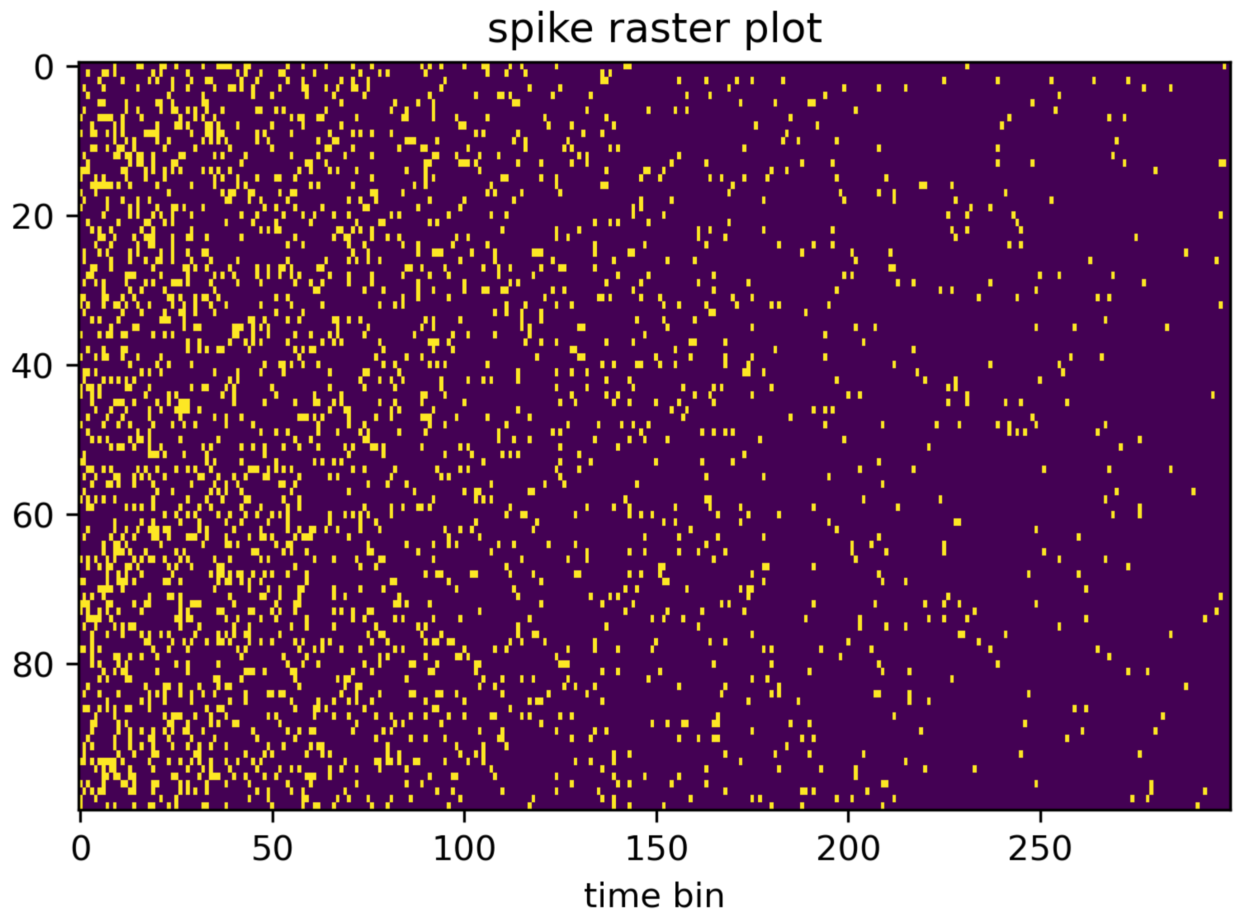
Cell 2 50°

In [79]:

```
stimDir=50
cell_num=2
times=np.arange(nmsec) #array of time points (1 msec apart)
spiketrain = np.zeros((ntrials,nmsec))      # set up output data

rate = cockroach_tuning(stimDir, cell_num)  #returns rate, in Hz.
tau = 100      # adaptation time constant in msec
delta_t=0.001  #time bin, in seconds (1 msec)
ratelist = rate*np.exp(-times/tau)  # list of adapting rates


rng = np.random.default_rng()

for j in np.arange(ntrials):
    for i in np.arange(nmsec):
            spiketrain[j,i] = np.round(rng.uniform(0,1,1) + ratelist[i]*delta_t -1/2

plt.imshow(spiketrain,aspect='auto')
plt.title('spike raster plot')
plt.xlabel('time bin')
```

Out[79]:

```
Text(0.5, 0, 'time bin')
```



In [80]:

```
T = nmsec / 1000
rate_per_trial=1/T * np.sum(spiketrain,axis=1)

mean_rate_per_trial = np.mean(rate_per_trial)
print('mean_rate_per_trial=',mean_rate_per_trial)

std_dev_rate_per_trial = np.std(rate_per_trial)
print('std_dev_rate_per_trial=',std_dev_rate_per_trial)
```

```
mean_rate_per_trial= 85.03333333333335
std_dev_rate_per_trial= 12.974290817698753
```

For cell 2 at 50°, again each row represents a single trial, so there are 100 rows in total corresponding to the 100 trials we run here, and every yellow bar shows the spikes during the 300 milliseconds timespan. The plot is overall similar to the previous one.

The mean rate per trial of 85.03 and the standard deviation of 12.974. Considering the standard deviation is much smaller than the mean, we believe that we have enough trials here.

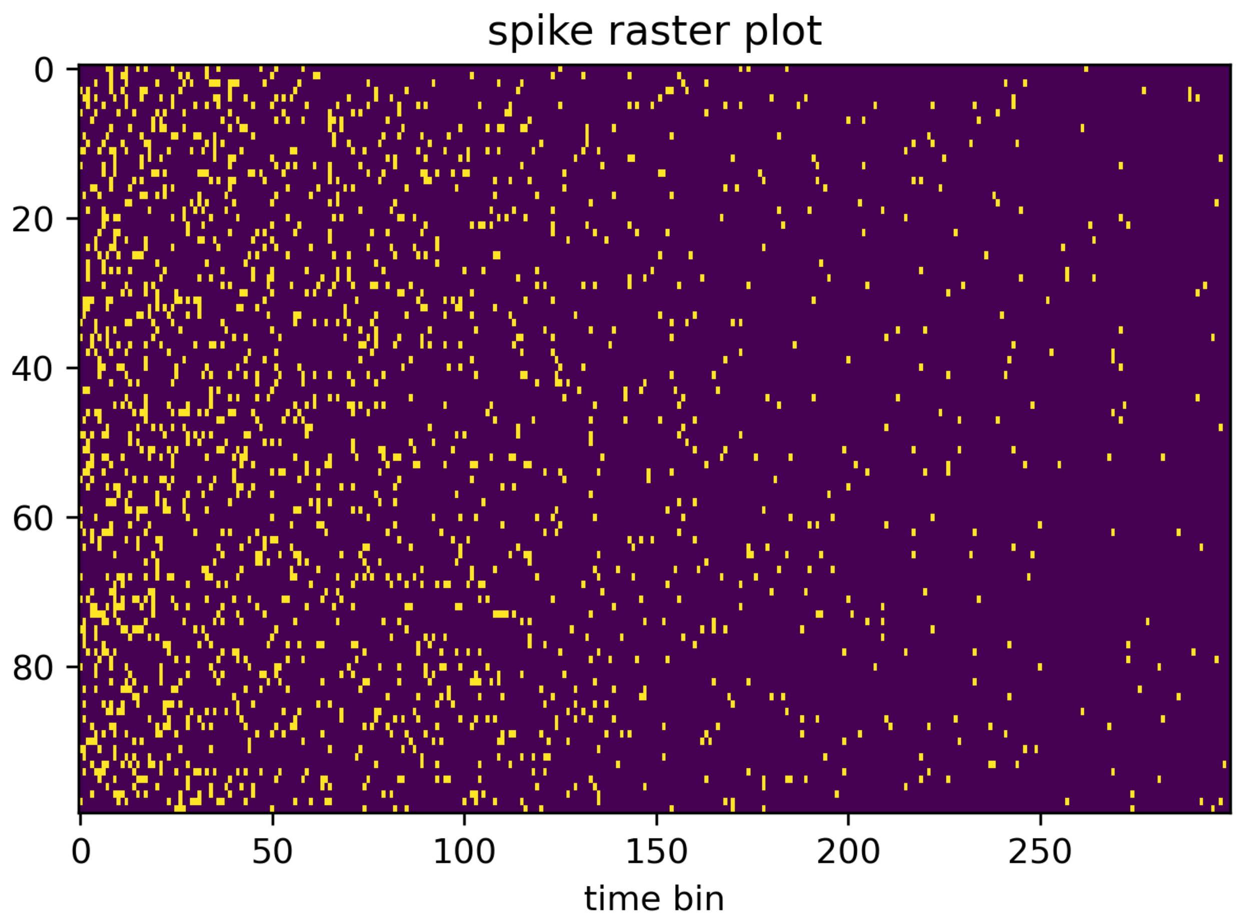
Cell 3 45°

In [81]:

```
stimDir=45
cell_num=3
times=np.arange(nmsec) #array of time points (1 msec apart)
spiketrain = np.zeros((ntrials,nmsec))      # set up output data
```

```
rate = cockroach_tuning(stimDir, cell_num)  #returns rate, in Hz.
tau = 100      # adaptation time constant in msec
delta_t=0.001  #time bin, in seconds (1 msec)
ratelist = rate*np.exp(-times/tau)  # list of adapting rates


rng = np.random.default_rng()

for j in np.arange(ntrials):
    for i in np.arange(nmsec):
            spiketrain[j,i] = np.round(rng.uniform(0,1,1) + ratelist[i]*delta_t -1/2

plt.imshow(spiketrain,aspect='auto')
plt.title('spike raster plot')
plt.xlabel('time bin')
```

Out[81]: Text(0.5, 0, 'time bin')



In [82]:

```
T = nmsec / 1000
rate_per_trial=1/T * np.sum(spiketrain,axis=1)

mean_rate_per_trial = np.mean(rate_per_trial)
print('mean_rate_per_trial=',mean_rate_per_trial)

std_dev_rate_per_trial = np.std(rate_per_trial)
print('std_dev_rate_per_trial=',std_dev_rate_per_trial)
```

mean_rate_per_trial= 62.06666666666667

```
std_dev_rate_per_trial= 13.002392942155772
```

For cell 3 at 45°, again each row represents a single trial, so there are 100 rows in total corresponding to the 100 trials we run here, and every yellow bar shows the spikes during the 300 milliseconds timespan. The plot is overall similar to the previous one.

The mean rate per trial of 62.067 and the standard deviation of 13.002. Considering the standard deviation is much smaller than the mean, we believe that we have enough trials here.

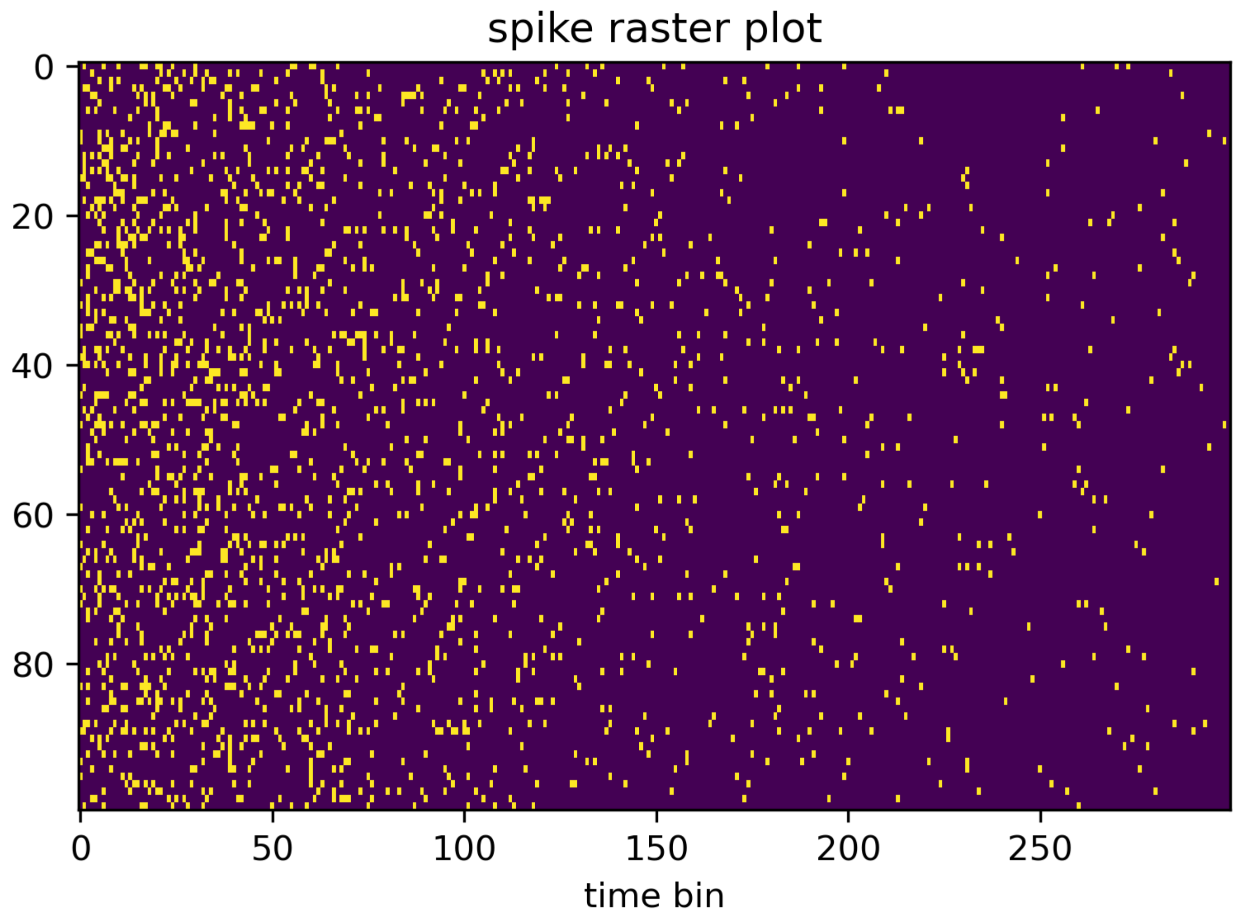
Cell 3 50°

In [83]:

```
stimDir=50
cell_num=3
times=np.arange(nmsec) #array of time points (1 msec apart)
spiketrain = np.zeros((ntrials,nmsec))      # set up output data

rate = cockroach_tuning(stimDir, cell_num)  #returns rate, in Hz.
tau = 100      # adaptation time constant in msec
delta_t=0.001  #time bin, in seconds (1 msec)
ratelist = rate*np.exp(-times/tau)  # list of adapting rates


rng = np.random.default_rng()

for j in np.arange(ntrials):
    for i in np.arange(nmsec):
            spiketrain[j,i] = np.round(rng.uniform(0,1,1) + ratelist[i]*delta_t -1/2

plt.imshow(spiketrain,aspect='auto')
plt.title('spike raster plot')
plt.xlabel('time bin')
```

Out[83]:

```
Text(0.5, 0, 'time bin')
```



In [84]:

```
T = nmsec / 1000
rate_per_trial=1/T * np.sum(spiketrain,axis=1)

mean_rate_per_trial = np.mean(rate_per_trial)
print('mean_rate_per_trial=',mean_rate_per_trial)

std_dev_rate_per_trial = np.std(rate_per_trial)
print('std_dev_rate_per_trial=',std_dev_rate_per_trial)
```

```
mean_rate_per_trial= 68.4
std_dev_rate_per_trial= 15.452292170850686
```

For cell 3 at 50°, again each row represents a single trial, so there are 100 rows in total corresponding to the 100 trials we run here, and every yellow bar shows the spikes during the 300 milliseconds timespan. The plot is overall similar to the previous one.

The mean rate per trial of 68.4 and the standard deviation of 15.452. Considering the standard deviation is much smaller than the mean, we believe that we have enough trials here.

In [ ]:

```
```