First of all, let's import all the needed libraries.

In [45]:
```
import numpy as np
import matplotlib.pyplot as plt
```

Let's set some useful values for our experiment and let's define also the variables where to store our drawings.

We are going to simulate both methods:

- method 1: draw N arrival times uniformly distributed in [0, T]
- method 2: draw a set of N inter arrival times exponentially distributed in [0, T] (exponential RV of parameter λ).

In [46]:
```
BINS = 100 #number of bins for histogram

arrival_rate = 5 #events per unit of time
T = 5000 #time period of interest

N = int(arrival_rate * T) #number of events in time period

repetitions = 50
#we repeat our experiment 50 times to smooth the histogram

events_uniform = []
exponential_arrival_time = []
```

Let's define the exponential pdf that our inter-arrival times obtained by method 1 (by computing the distance between pairs of events) should fit.

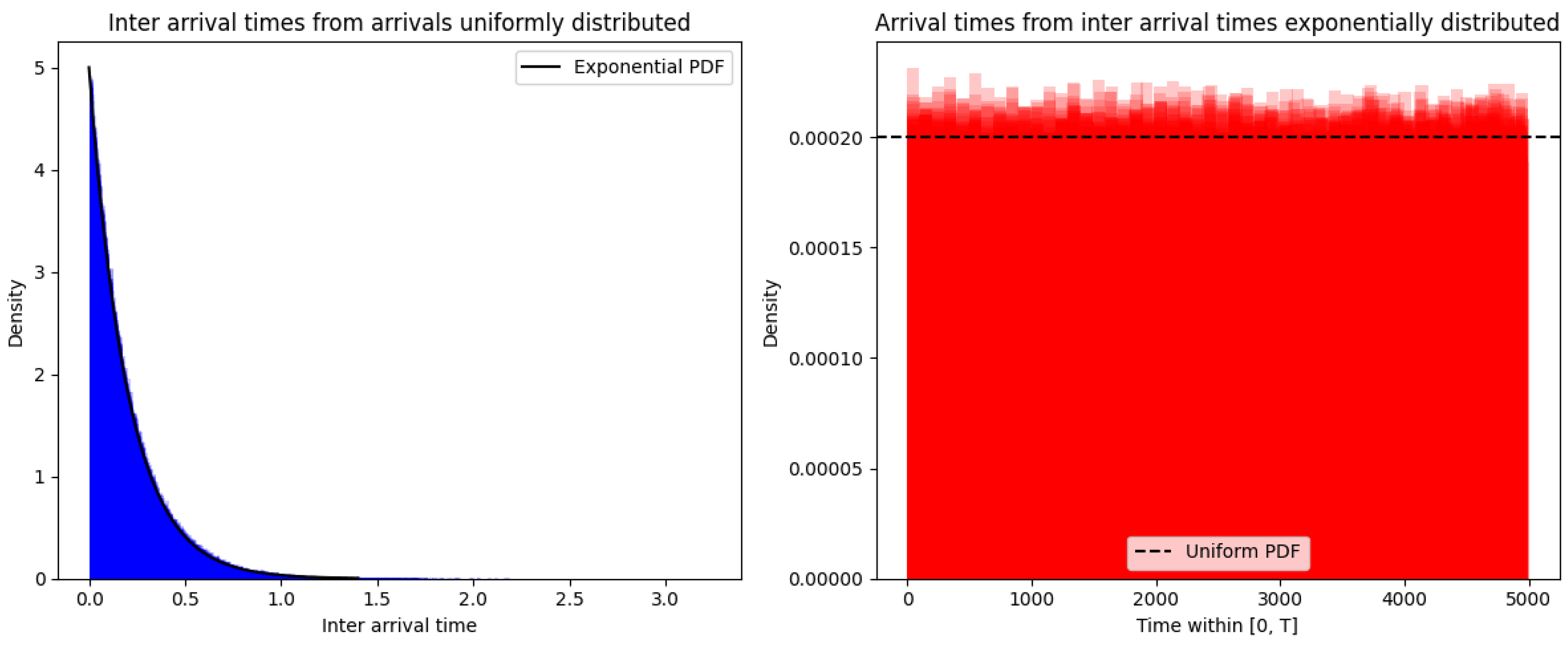
Let's define also the uniform distribution that our arrival times obtained by method 2 (by cumulating summing the arrival times) should fit.

In [47]:
```
def exp_pdf(x, param=(arrival_rate)):
    return param * np.exp(-param * x)

#let's take some points for plotting
x_points = np.linspace(0, 7/arrival_rate, 200)
y_points = exp_pdf(x_points)

def uniform_pdf(x, params=(0, T)):
    return 1 / (params[1] - params[0])
```

Now let's draw our points and do our simulation.

We can see with the black lines the expected behaviour of our distribution.

In [48]:
```
N = int(arrival_rate * T)

#define plot
fig = plt.figure(figsize=(12, 5))
ax1 = fig.add_subplot(1, 2, 1)
ax2 = fig.add_subplot(1, 2, 2)

ax1.plot(x_points, y_points, color='black',label='Exponential PDF')
ax2.axhline(uniform_pdf(0), color='black', linestyle='--', label='Uniform PDF')

ax1.set_title('Inter arrival times from arrivals uniformly distributed')
ax2.set_title('Arrival times from inter arrival times exponentially distributed')

ax1.set_xlabel('Inter arrival time')
ax1.set_ylabel('Density')

ax2.set_xlabel('Time within [0, T]')
ax2.set_ylabel('Density')

ax1.legend()
ax2.legend()

#draw the histograms
for _ in range(repetitions):
    # Method 1:
    events_uniform = np.random.uniform(0, T, N)
    sorted_events_uni = np.sort(events_uniform)
    inter_arrival_times = np.diff(sorted_events_uni)

    # Method 2:
    exponentials = np.random.exponential(1/arrival_rate, N) #first parameter is the mean of the distribution
    events_exp = np.cumsum(exponentials)
    while events_exp[-1] > T:
        exponentials = np.random.exponential(1/arrival_rate, N)
        events_exp = np.cumsum(exponentials)

    ax1.hist(inter_arrival_times, bins=BINS, density=True, alpha=0.2, color='blue', histtype='stepfilled')
    ax2.hist(events_exp, bins=BINS//2, density=True, alpha=0.2, color='red', histtype='stepfilled')

plt.tight_layout()
plt.show()
```



It is important to underline that for method 2 we need to put in practice additional checks for the drawing.

In order to be sure to have events in [0, T] we discard the sampling of inter arrival times when the last arrival exceed T!

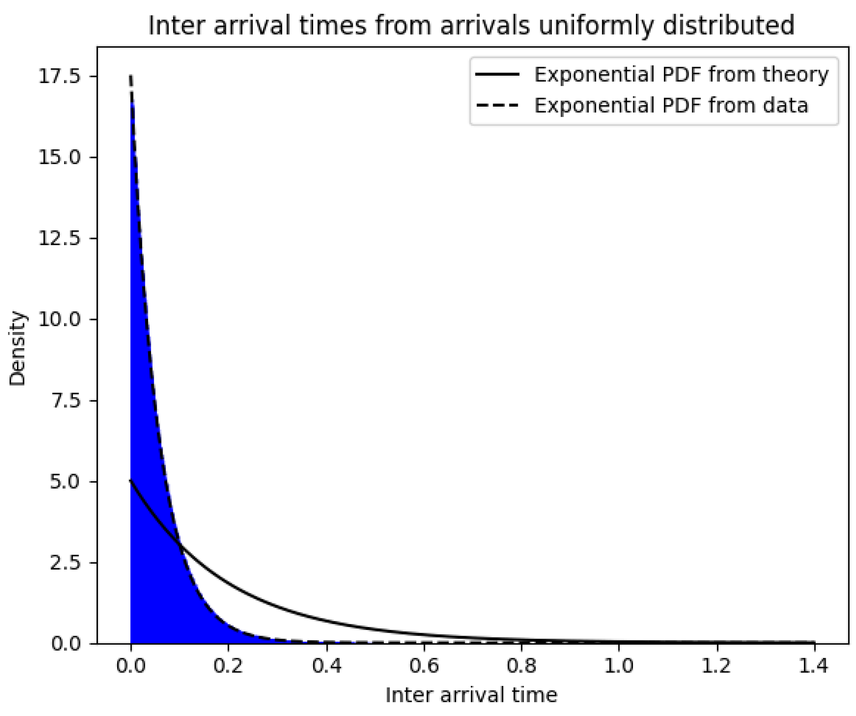
We could have done also a rescaling of the values, but then our distribution would have became distorted!

What if N >> λ * T

In that case we are drawing *more* examples than expected. The difference is shown in the inter-arrival times, computed from the uniform distribution: our drawings are no more following the theoretical pdf.

Actually we are having an *higher* arrival rate. So the samplings will follow the distribution as it would have an higher parameter!

In [49]:
```
BIGGER_N = int(arrival_rate * T*3.5) #increase number of events

#define plot
fig = plt.figure(figsize=(6, 5))
ax1 = fig.add_subplot(1, 1, 1)

#let's take some points for plotting for the new exp pdf
ax1.plot(x_points, y_points, color='black',label='Exponential PDF from theory')
new_y_points = exp_pdf(x_points, param=(BIGGER_N/T))
ax1.plot(x_points, new_y_points, color='black', linestyle='--', label='Exponential PDF from data')

ax1.set_title('Inter arrival times from arrivals uniformly distributed')

ax1.set_xlabel('Inter arrival time')
ax1.set_ylabel('Density')

ax1.legend()

#draw the histogram
for _ in range(repetitions):
    events_uniform = np.random.uniform(0, T, BIGGER_N)
    sorted_events_uni = np.sort(events_uniform)
    inter_arrival_times = np.diff(sorted_events_uni)

    ax1.hist(inter_arrival_times, bins=BINS, density=True, alpha=0.2, color='blue', histtype='stepfilled')

plt.tight_layout()
plt.show()
```

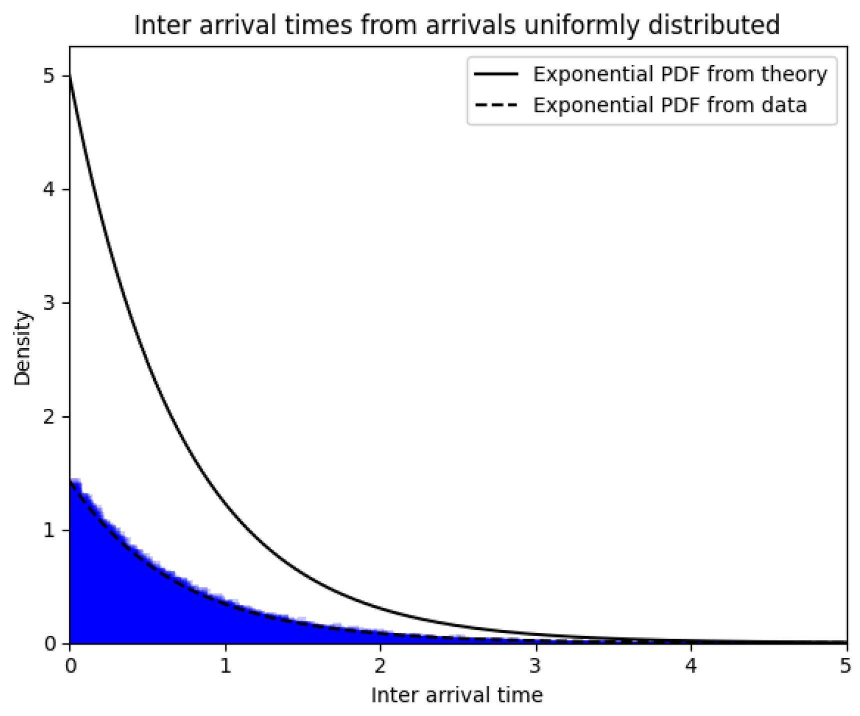


What if N << λ * T

Similar as above. In that case we are drawing *less* examples than expected. The difference is shown in the inter-arrival times, computed from the uniform distribution: our drawings are no more following the theoretical pdf.

Actually we are having a *lower* arrival rate. So the samplings will follow the distribution as it would have a lower parameter!

In [50]:
```
SMALLER_N = int(arrival_rate * T / 3.5) #decrease number of events

#define
fig = plt.figure(figsize=(6, 5))
ax1 = fig.add_subplot(1, 1, 1)

new_x_points = np.linspace(0, 5, 200) #it converges faster
#let's take some points for plotting for the new exp pdf
new_y_points = exp_pdf(new_x_points, param=(SMALLER_N/T))
ax1.plot(new_x_points, y_points, color='black',label='Exponential PDF from theory')
ax1.plot(new_x_points, new_y_points, color='black', linestyle='--', label='Exponential PDF from data')

ax1.set_title('Inter arrival times from arrivals uniformly distributed')

ax1.set_xlabel('Inter arrival time')
ax1.set_ylabel('Density')
ax1.set_xlim(0, 5)

ax1.legend()

#draw the histogram
for _ in range(repetitions):
    events_uniform = np.random.uniform(0, T, SMALLER_N)
    sorted_events_uni = np.sort(events_uniform)
    inter_arrival_times = np.diff(sorted_events_uni)

    ax1.hist(inter_arrival_times, bins=BINS, density=True, alpha=0.2, color='blue', histtype='stepfilled')

plt.tight_layout()
plt.show()
```


Inter arrival times from arrivals uniformly distributed
Exponential PDF from theory
Exponential PDF from data
Density
Inter arrival time
0
1
2
3
4
5
0
1
2
3
4
5