Συστήματα Αναμονής – 1η Σειρά Ασκήσεων

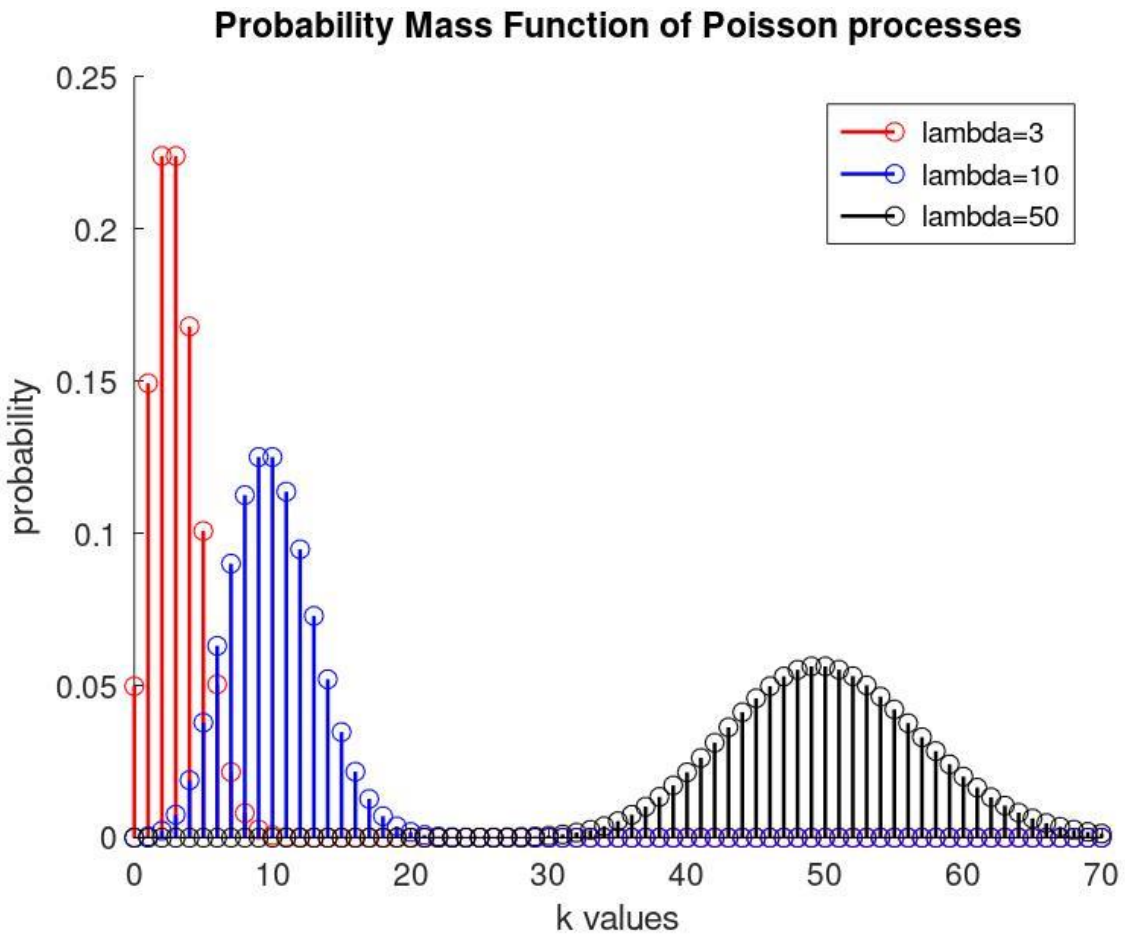
Γεώργιος Κυριακόπουλος – el18153

Κατανομή Poisson:

Α) Με βάση το δοσμένο κώδικα και τα γραφήματα που προκύπτουν από αυτόν, το πλάτος της συνάρτησης μειώνεται, καθώς μεγαλώνει η τιμή της παραμέτρου λ, ενώ η συνάρτηση μετατοπίζεται προς τα δεξιά. Αυτό συμβαίνει διότι η παράμετρος λ μίας κατανομής $Poisson$ ισούται με τη μέση τιμή της και με τη διακύμανση της. Χρειάζεται, επομένως μεγαλύτερη διακύμανση, η οποία μοιράζει το συνολικό άθροισμα των πλατών που πρέπει να ισούται με 1, σε περισσότερα σημεία με μικρότερα πλάτη το καθένα.

```
% TASK: In a common diagram, design the Probability Mass Function of Poiss
on processes
% with lambda parameters 3, 10, 50. In the horizontal axes, choose k param
eters
% between 0 and 70.

k = 0:1:70;
lambda = [3,10,50,30];

for i=1:columns(lambda)
  poisson(i,:) = poisspdf(k,lambda(i));
endfor

colors = "rbkm";
figure(1);
hold on;

% Don't draw for lambda = 30 (last column), since it's not asked.
for i=1:(columns(lambda)-1)
  stem(k,poisson(i,:),colors(i),"linewidth",1.2);
endfor
hold off;

title("Probability Mass Function of Poisson processes");
xlabel("k values");
```

```
ylabel("probability");
legend("lambda=3","lambda=10","lambda=50");
```



B) Σύμφωνα και με το δοσμένα κώδικα και τα αποτελέσματα που εμφανίζει αυτός, η μέση τιμή και η διακύμανση της κατανομής $Poisson$ είναι ίσες μεταξύ τους και ίσες με την παράμετρο $\lambda = 30$.

```
% TASK: Regarding the Poisson process with parameter lambda 30, compute it
s mean
% value and variance.

index = find(lambda == 30);
chosen = poisson(index,:);

mean_value = 0;
for i=0:(columns(poisson(index,:))-1)
  mean_value = mean_value + i.*poisson(index,i+1);
endfor
```

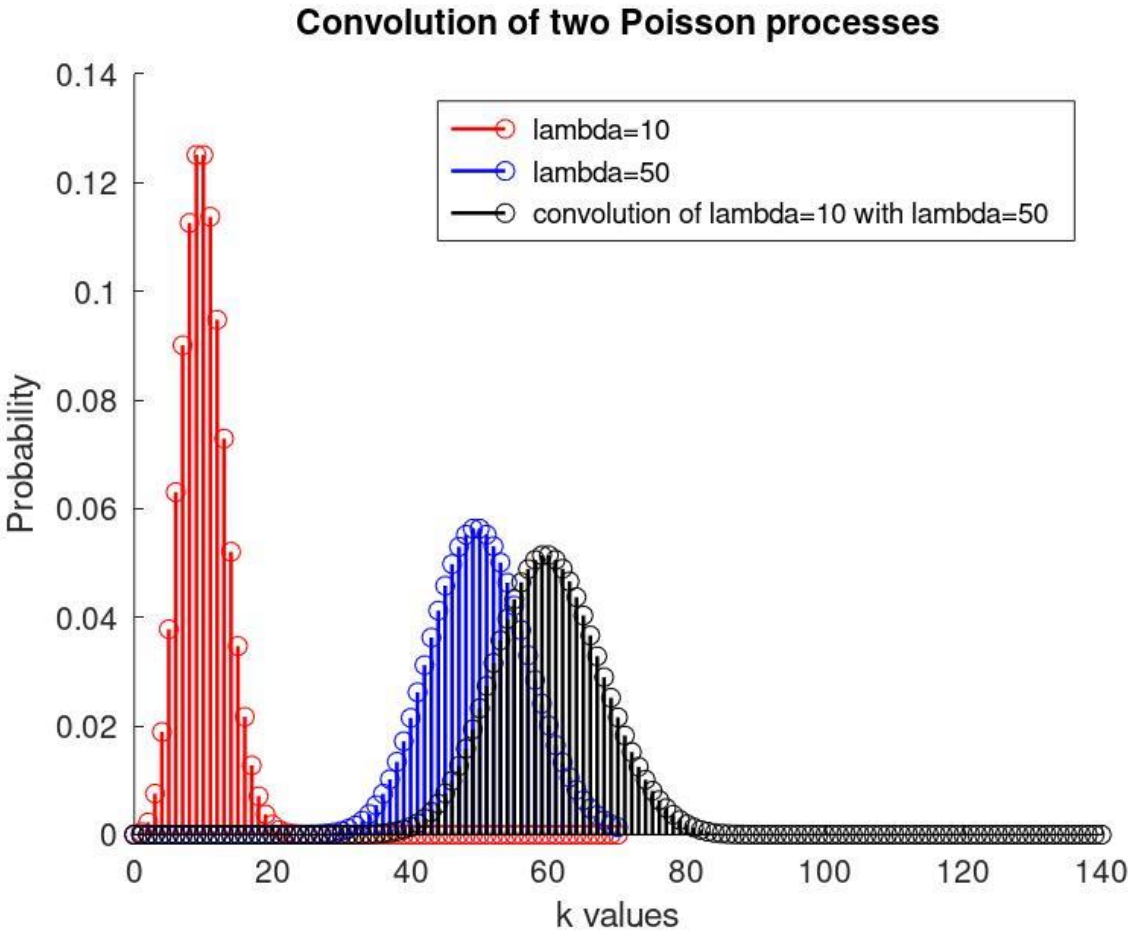
```
printf("Mean value of Poisson with lambda 30 is: %d.\n", mean_value);

second_moment = 0;
for i=0:(columns(poisson(index,:))-1)
  second_moment = second_moment + i.*i.*poisson(index,i+1);
endfor

variance = second_moment - mean_value.^2;
printf("Variance of Poisson with lambda 30 is: %d.\n", variance);
```

```
Mean value of Poisson with lambda 30 is: 30.
Variance of Poisson with lambda 30 is: 30.
```

Γ) Όπως είναι εμφανές και στα γραφήματα που παράγει ο κώδικας, η κατανομή που προκύπτει είναι και αυτή μία κατανομή $Poisson$, η οποία έχει μάλιστα παράμετρο $\lambda = 50 + 10 = 60$, κάτι που μπορούμε εύκολα να επιβεβαιώσουμε και γραφικά με τον τρόπο που φαίνεται και στα σχόλια του σχετικού μέρους στον κώδικα. Η προϋπόθεση για να συμβαίνει αυτό είναι οι τυχαίες μεταβλητές των δύο κατανομών που υπερθέτουμε να είναι ανεξάρτητες μεταξύ τους. Γενικά, με την προϋπόθεση αυτή, έχουμε ότι η υπέρθεση δύο ή περισσοτέρων κατανομών $Poisson$ μας δίνει επίσης μία κατανομή $Poisson$ με παράμετρο λ το άθροισμα των δύο η περισσοτέρων παραμέτρων λ των κατανομών που υπερθέτουμε.

```
% TASK: Consider the convolution of the Poisson distribution with lambda 1
0 with
% the Poisson distribution with lambda 50.

first = find(lambda==10);
second = find(lambda==50);
poisson_first = poisson(first,:);
poisson_second = poisson(second,:);

composed = conv(poisson_first,poisson_second);
new_k = 0:1:(2*70);

figure(2);
hold on;
stem(k,poisson_first(:),colors(1),"linewidth",1.2);
stem(k,poisson_second(:),colors(2),"linewidth",1.2);
stem(new_k,composed,colors(3),"linewidth",1.2);
```

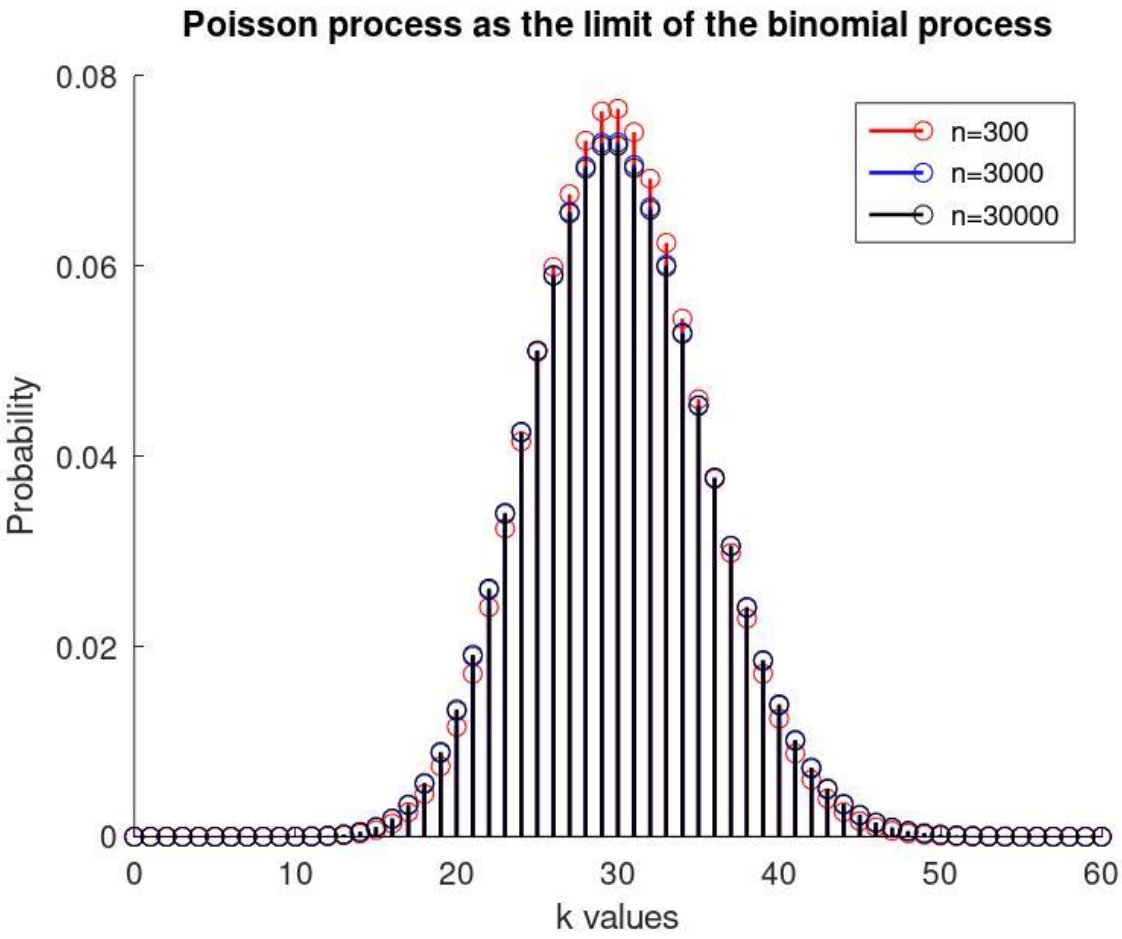
```
% The commented commands below can be used to graph a Poisson distribution
 with lambda = 60
% to observe that it is the same as the convolution of the 2 other distrib
utions.

% poisson_sum(1,:) = poisspdf(new_k,60);
% stem(new_k,poisson_sum(:),colors(4),"linewidht",1.2);

hold off;
title("Convolution of two Poisson processes");
xlabel("k values");
ylabel("Probability");
legend("lambda=10","lambda=50","convolution of lambda=10 with lambda=50");
```



Δ) Η κατανομή $Poisson$ προκύπτει ως μία οριακή περίπτωση της διωνυμικής κατανομής, όπου ο αριθμός των πειραμάτων n αυξάνει άπειρα, ενώ το γινόμενο $\mu = np$, που ισούται με την

αναμενόμενη τιμή του αριθμού των επιτυχημένων πειραμάτων, παραμένει περίπου σταθερό και πλησιάζει την παράμετρο λ της κατανομής $Poisson$. Έτσι, ακολουθώντας την ίδια διαδικασία, ορίζουμε $\lambda = 30\ σημεία/sec$ και ανάλογα τα n και p, ώστε να αρχίσουμε να πλησιάζουμε την επιθυμητή κατανομή $Poisson$.

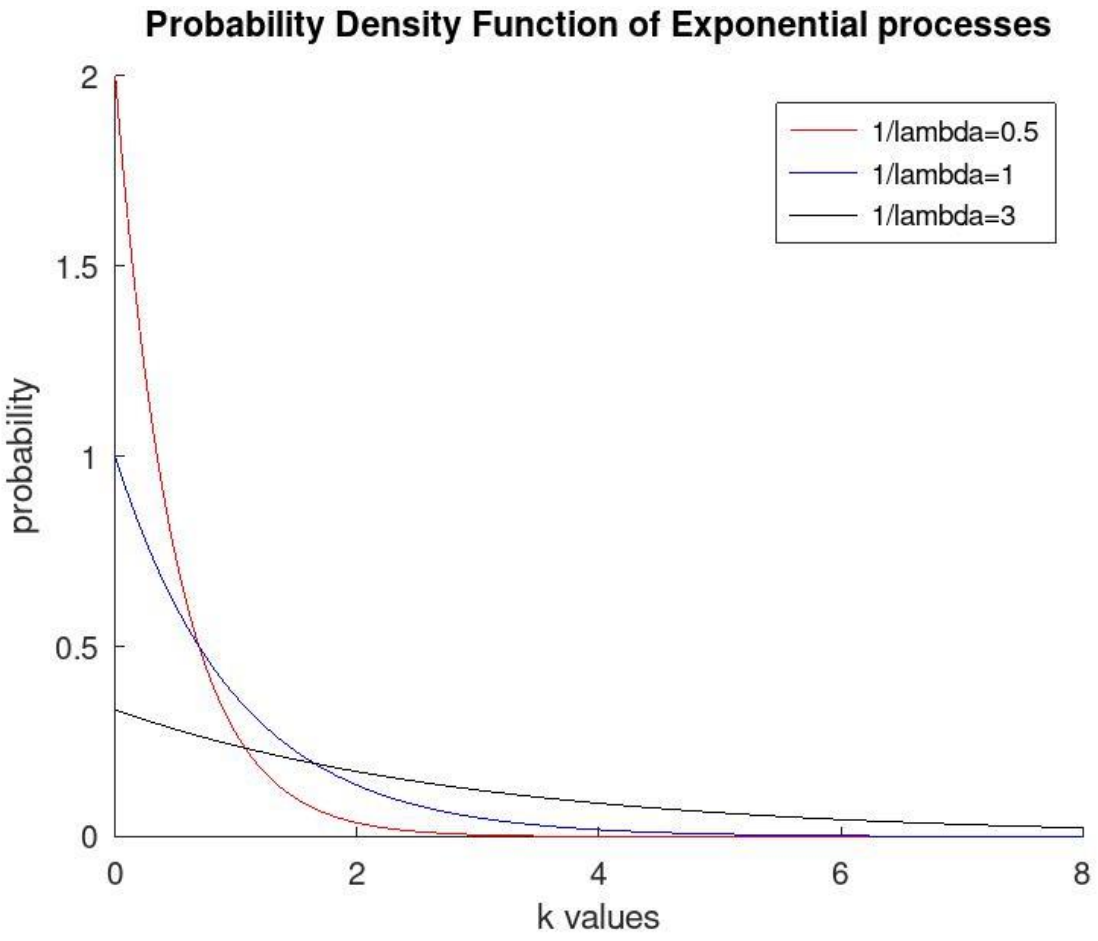
```
% TASK: Show that Poisson process is the limit of the binomial distributio
n.

k = 0:1:60;

% Define the desired Poisson Process
lambda = 30;
i = 1:1:3;
n = [300, 3000, 30000];
p = lambda./n;

figure(3);
title("Poisson process as the limit of the binomial process");
xlabel("k values");
ylabel("Probability");
hold on;

for i=1:3
  binomial = binopdf(k,n(i),p(i));
  stem(k,binomial,colors(i),"linewidth",1.2);
endfor
legend("n=300","n=3000","n=30000");
hold off;
```


Poisson process as the limit of the binomial process
0.08
0.06
0.04
0.02
0
Probability
0
10
20
30
40
50
60
k values
n=300
n=3000
n=30000

Εκθετική κατανομή:

Α) Χρησιμοποιώντας την εντολή $exppdf()$ και ορίζοντας ως μέσους όρους τα $1/\lambda = \{0.5, 1, 3\}$, σχεδιάζουμε τις τρεις συναρτήσεις πυκνότητας πιθανότητας των εκθετικών κατανομών σε κοινό διάγραμμα.

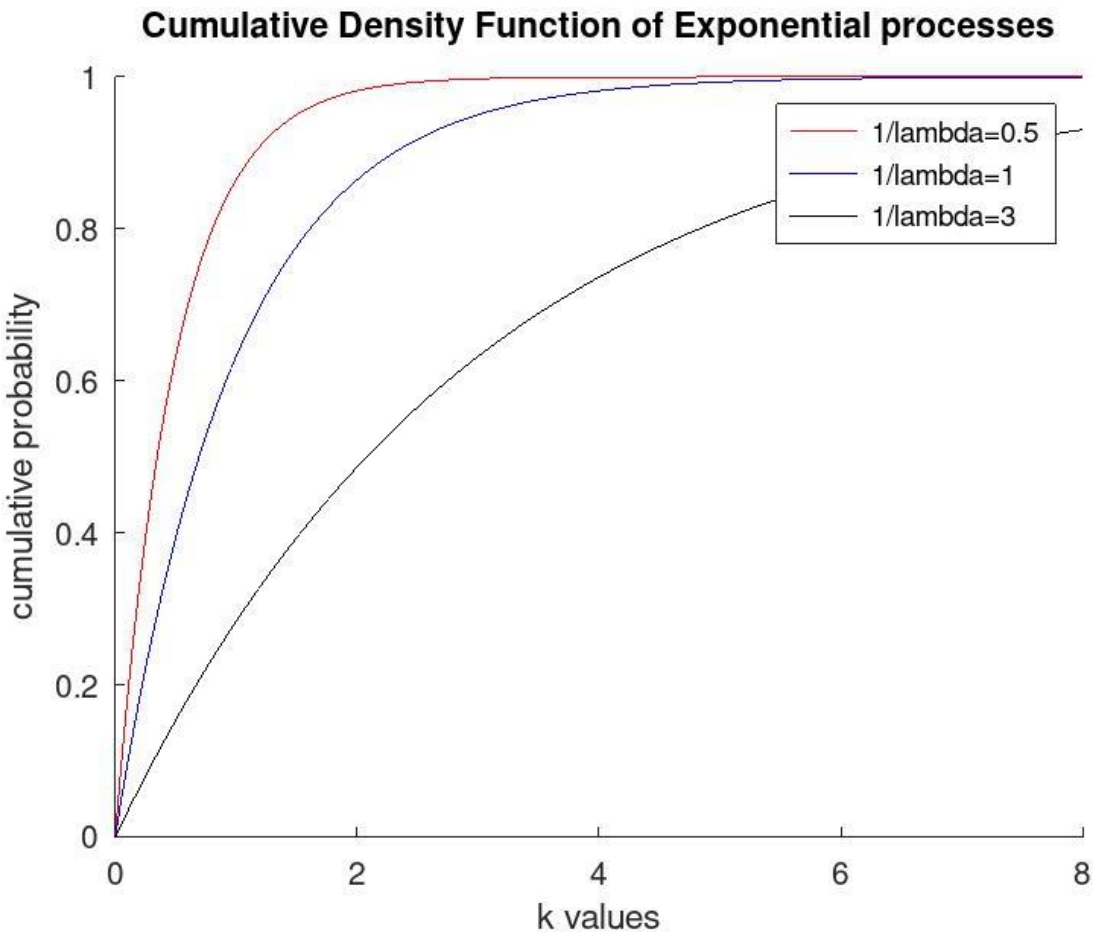
```
% TASK: In a common diagram, design the Probability Density Function of Ex
ponential processes
% with mean parameters (1/lambda) 0.5, 1, 3. In the horizontal axes, choos
e k parameters
% between 0 and 8.

k = 0:0.00001:8;
mean = [0.5,1,3];

for i=1:columns(mean)
  exponential(i,:) = exppdf(k,mean(i));
endfor

colors = "rbk";
figure(1,"graphicssmoothing","off");
hold on;

for i=1:columns(mean)
  plot(k,exponential(i,:),colors(i),"linewidth",0.1);
endfor

title("Probability Density Function of Exponential processes");
xlabel("k values");
ylabel("probability");
legend("1/lambda=0.5","1/lambda=1","1/lambda=3");
hold off;
```



B) Χρησιμοποιώντας, αυτή τη φορά, την εντολή $expcdf()$ και το ίδιο όρισμα για τους μέσους όρους, σχεδιάζουμε τις τρεις αθροιστικές συναρτήσεις κατανομής των εκθετικών κατανομών σε κοινό διάγραμμα.

```
% TASK: In a common diagram, design the Cumulative Density Function of Exp
onential processes
% with mean parameters (1/lambda) 0.5, 1, 3. In the horizontal axes, choos
e k parameters
% between 0 and 8.

for i=1:columns(mean)
  exponentialc(i,:) = expcdf(k,mean(i));
endfor

colors = "rbk";
figure(2,"graphicssmoothing","off");
hold on;
```

```
for i=1:columns(mean)
  plot(k,exponentialc(i,:),colors(i),"linewidth",0.1);
endfor

title("Cumulative Density Function of Exponential processes");
xlabel("k values");
ylabel("cumulative probability");
legend("1/lambda=0.5","1/lambda=1","1/lambda=3");
hold off;
```



Γ) Οι δύο αυτές πιθανότητες είναι ίσες, λόγω της ιδιότητας απώλειας μνήμης της εκθετικής κατανομής, το οποίο πρακτικά σημαίνει ότι το σύστημα ξεχνάει συνεχώς την κατάσταση στην οποία βρίσκεται. Αυτό είναι φανερό και από τον γενικό κανόνα: $Pr(X > m + n|X > m) = Pr(X > n)$, ο οποίος εφαρμόζεται και στο δικό μας παράδειγμα ως: $Pr(X > 50000|X > 20000) = Pr(X > 20000 + 30000|X > 20000) = Pr(X > 30000)$. Επίσης χρησιμοποιούμε τις ιδιότητες: $Pr(X > m) = 1 - Pr(X < m)$ και $Pr(X > m + n|X > m) = Pr(X > m + n)$.

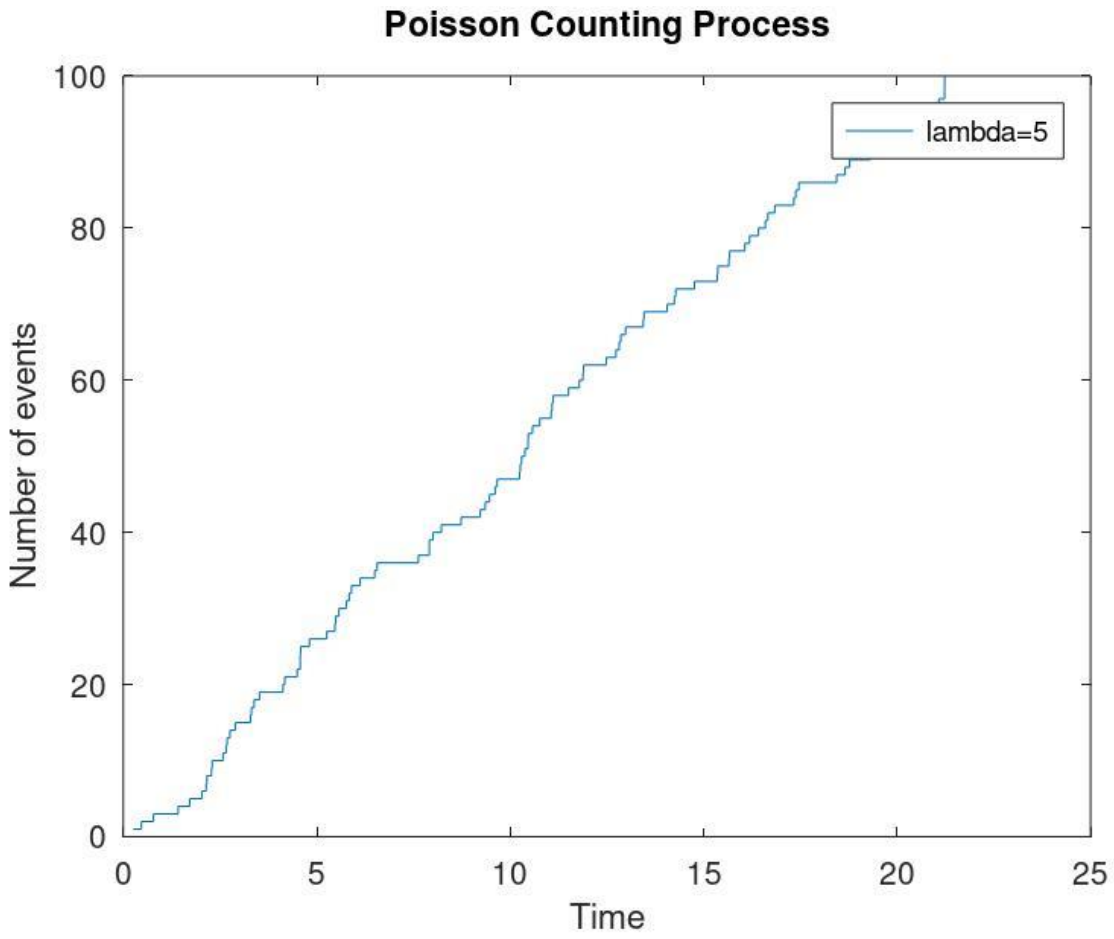
```
% TASK: Using the Cumulative Density Function of Exponential Processes cal
culate the following
% probabilities: P(X>30000) and Pr(X>50000|X>20000).

k = 0:0.00001:8;
expo(1,:) = expcdf(k,2.5);

p_30000 = 1 - expo(30000);
p_50000 = 1 - expo(50000);
p_20000 = 1 - expo(20000);

%P(A|B) = P(A^B)/P(B);
conditional = p_50000/p_20000;

printf("Pr(X>30000) is equal to: %d.\n", p_30000);

printf("Pr(X>50000|X>20000) is equal to: %d.\n", conditional);
```

```
Pr(X>30000) is equal to: 0.886924.
Pr(X>50000|X>20000) is equal to: 0.88692.
```

Διαδικασία Καταμέτρησης Poisson:

Α) Οι χρόνοι αναμονής γνωρίζουμε ότι ακολουθούν μία εκθετική κατανομή με παράμετρο $1/\lambda$. Γνωρίζουμε επίσης ότι ο αριθμός των συνολικών γεγονότων ισούται με το συνολικό χρόνο αναμονής επί την παράμετρο λ. Χρησιμοποιώντας την $exprnd()$ για 100 διαδοχικά τυχαία γεγονότα με $\lambda = 5\ γεγονότα/sec$, σχεδιάζουμε μία διαδικασία καταμέτρησης $Poisson$. Στη μεταβλητή $arrival\_time$ βάζουμε το $prefix\ sum$ του πίνακα των $samples$, φτιάχνοντας ένα πίνακα χρόνου με τιμές τον συνολικό χρόνο από την αρχή μέχρι την άφιξη κάποιου πακέτου, ώστε αργότερα με τη χρήση της $stairs(arrival\_time, 1{:}\,100)$ να ανεβαίνουμε ένα γεγονός στο μετρητή μας (άξονας y), όταν οι τιμές του άξονα x ισούνται με τους χρόνους άφιξης. Παρατηρούμε, επίσης, ότι ο συνολικός χρόνος είναι κοντά στα $20\ sec$, κάτι το οποίο επιβεβαιώνει το $counter = \lambda \cdot T$ $(100 = 5 \cdot 20)$.

```
% TASK: Create 100 consecutive Poisson processes wiith exprnd() and draw t
he stairs function
% of one Poisson counting process with lambda = 5.

lambda = 5;
samples = exprnd(1/lambda, 1, 100);
arrival_time = cumsum(samples);
stairs(arrival_time, 1:100);

title("Poisson Counting Process");
xlabel("Time");
ylabel("Number of events");
legend("lambda=5");
hold off;
```



Β) Ο αριθμός γεγονότων σε ένα χρονικό παράθυρο $\Delta T = t1 - t2$ γνωρίζουμε ότι ακολουθεί μία κατανομή $Poisson$ με μέσο όρο ίσο με την παράμετρο λ επί το χρονικό παράθυρο ΔT. Βρίσκοντας το μέσο αριθμό γεγονότων ανά μονάδα χρόνου για $100, 200, 300, 500, 1000, 10000$ διαδοχικά τυχαία γεγονότα διαιρώντας τον συνολικό αριθμό γεγονότων με τον συνολικό χρόνο που χρειάζεται μέχρι να συμβεί και το τελευταίο γεγονός, παρατηρούμε ότι αυτός πλησιάζει όλο και περισσότερο την παράμετρο λ (μέσω της απόκλισης του από αυτήν, την οποία εκτυπώνουμε), όσο αυξάνεται ο αριθμός των γεγονότων.

```
% TASK: Find mean number of events per second for a window DT = t1 - t2 of
 100 consecutive
% events. Repeat for 200, 300, 500, 1000, 10000 consecutive events.

end_time = arrival_time(end);
mean = 100/end_time;
deviation = abs(lambda-mean);
printf("For 100 samples mean is equal to %d and deviation is equal to %d.\
n", mean, deviation);
```

```
n_samples = [200, 300, 500, 1000, 10000];

for i=1:5
  samples = exprnd(1/lambda, 1, n_samples(i));
  arrival_time = cumsum(samples);
  end_time = arrival_time(end);
  mean = n_samples(i)/end_time;
  deviation = abs(lambda-mean);
  printf("For %d samples mean is equal to %d and deviation is equal to %d.
\n", n_samples(i), mean, deviation);
endfor
```

```
For 100 samples mean is equal to 4.71043 and deviation is equal to 0.289573.
For 200 samples mean is equal to 5.3588 and deviation is equal to 0.358801.
For 300 samples mean is equal to 5.58271 and deviation is equal to 0.582708.
For 500 samples mean is equal to 4.91755 and deviation is equal to 0.0824512.
For 1000 samples mean is equal to 4.94266 and deviation is equal to 0.0573386.
For 10000 samples mean is equal to 5.02974 and deviation is equal to 0.0297359.
```