Συστήματα Αναμονής

3η Ομάδα Ασκήσεων

Άννα Κουτσώνη 03120019

Προσομοίωση συστήματος M/M/1/10

1. Για την μελέτη ενός συστήματος M/M/1/10, για το οποίο επιλέγουμε λ=5 και μ=5, όπως δίνεται, χρησιμοποιούμε τον παρακάτω κώδικα σε octave:

```
%1

clc;
clear all;
close all;

P = [0,0,0,0,0,0,0,0,0,0,0];
arrivals = [0,0,0,0,0,0,0,0,0,0,0];
total_arrivals = 0; % to measure the total number of arrivals
current_state = 0;  % holds the current state of the system
previous_mean_clients = 0; % will help in the convergence test
index = 0;

lambda = 5;
mu = 5;
threshold = lambda/(lambda + mu); % the threshold used to calculate probabilities

transitions = 0; % holds the transitions of the simulation in transitions steps

while transitions >= 0
  transitions = transitions + 1; % one more transitions step

  if mod(transitions,1000) == 0 % check for convergence every 1000 transitions steps
    index = index + 1;
    for i=1:1:length(arrivals)
        P(i) = arrivals(i)/total_arrivals; % calcuate the probability of every state ir
    endfor

    mean_clients = 0; % calculate the mean number of clients in the system
    for i=1:1:length(arrivals)
       mean_clients = mean_clients + (i-1).*P(i);
    endfor

    to_plot(index) = mean_clients;

    if abs(mean_clients - previous_mean_clients) < 0.00001 || transitions > 200000 % cc
      break;
    endif

    previous_mean_clients = mean_clients;

  endif

  random_number = rand(1); % generate a random number (Uniform distribution)
  if current_state == 0 || random_number < threshold % arrival
    if current_state < 11
      total_arrivals = total_arrivals + 1;
      if transitions < 31
        display("Current state = ");
        disp(current_state);
        display("Next transition");
        display("arrival");
        display("Total arrivals in current state = ");
        disp(arrivals(current_state+1));
      endif
      arrivals(current_state + 1) = arrivals(current_state + 1) + 1; % increase the num
      if current_state < 10
        current_state = current_state + 1;
      endif
    endif
  else % departure
    if current_state != 0 % no departure from an empty system
      if transitions < 31
        display("Current state = ");
        disp(current_state);
        display("Next transition");
        display("departure");
        display("Total arrivals in current state = ");
```

```
        disp(arrivals(current_state+1));
      endif
      current_state = current_state - 1;
    endif
  endif
endwhile



display("State propabilities:");
for i=1:1:length(arrivals)
  display(P(i));
endfor

g = lambda*(1-P(11));
average_delay_time = mean_clients / g;
display("Average delay time =");
disp(average_delay_time);
display("Blocking propability =");
disp(P(11));

figure(1);
plot(to_plot,"c","linewidth",1.3);
title("Average number of clients in the M/M/1 queue: Convergence");
xlabel("transitions in thousands");
ylabel("Average number of clients");

x=[0,1,2,3,4,5,6,7,8,9,10];
figure(2);
bar(x,P,0.4);
title("Probabilities")
```

Για το debugging της προσομοίωσης:

- Οι 30 πρώτες μεταβάσεις:

```
Current state =
0
Next transition
arrival
Total arrivals in current state =
0
Current state =
1
Next transition
departure
Total arrivals in current state =
0
Current state =
0
Next transition
arrival
Total arrivals in current state =
1
Current state =
1
Next transition
departure
Total arrivals in current state =
0
Current state =
0
Next transition
arrival
Total arrivals in current state =
2
Current state =
1
Next transition
departure
Total arrivals in current state =
0
Current state =
0
Next transition
arrival
```

```
Total arrivals in current state =
3
Current state =
1
Next transition
arrival
Total arrivals in current state =
0
Current state =
2
Next transition
arrival
Total arrivals in current state =
0
Current state =
3
Next transition
departure
Total arrivals in current state =
0
Current state =
2
Next transition
departure
Total arrivals in current state =
1
Current state =
1
Next transition
departure
Total arrivals in current state =
1
Current state =
0
Next transition
arrival
Total arrivals in current state =
4
Current state =
1
Next transition
departure
Total arrivals in current state =
1
Current state =
0
Next transition
arrival
Total arrivals in current state =
5
Current state =
1
Next transition
departure
Total arrivals in current state =
1
Current state =
0
Next transition
arrival
Total arrivals in current state =
6
Current state =
1
Next transition
arrival
Total arrivals in current state =
1
Current state =
2
Next transition
arrival
Total arrivals in current state =
1
Current state =
3
Next transition
arrival
Total arrivals in current state =
0
```

```
Current state =
4
Next transition
arrival
Total arrivals in current state =
0
Current state =
5
Next transition
departure
Total arrivals in current state =
0
Current state =
4
Next transition
arrival
Total arrivals in current state =
1
Current state =
5
Next transition
arrival
Total arrivals in current state =
0
Current state =
6
Next transition
departure
Total arrivals in current state =
0
Current state =
5
Next transition
departure
Total arrivals in current state =
1
Current state =
4
Next transition
arrival
```

```
Total arrivals in current state =
2
Current state =
5
Next transition
arrival
Total arrivals in current state =
1
Current state =
6
Next transition
departure
Total arrivals in current state =
0
Current state =
5
Next transition
departure
Total arrivals in current state =
2
```

- Οι πιθανότητες καταστάσεως του συστήματος

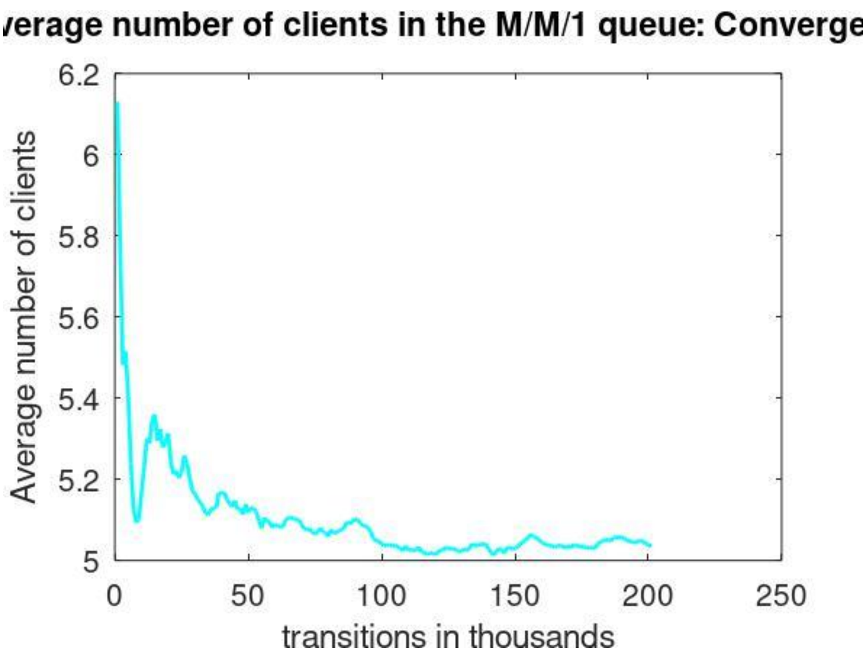
```
State propabilities:
0.091377
0.089441
0.089877
0.088985
0.091348
0.088900
0.090096
0.091149
0.093588
0.092477
0.092762
```

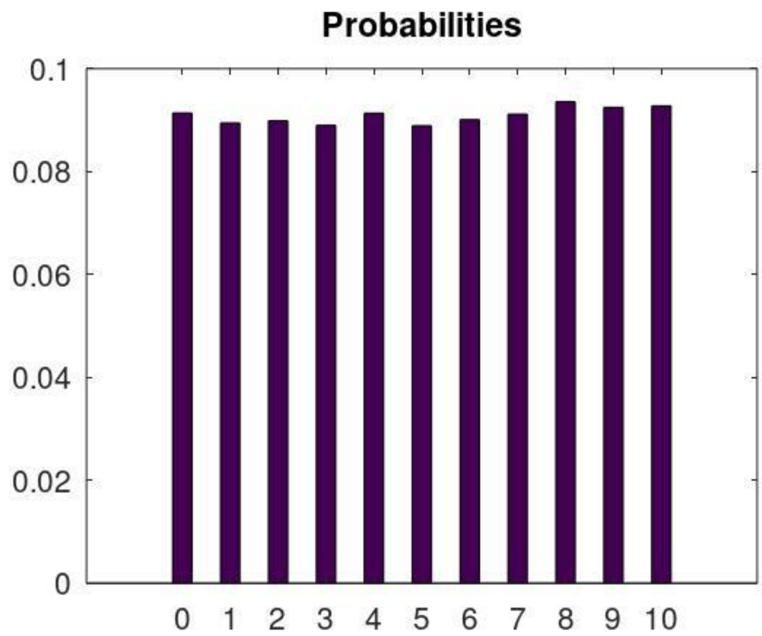
- Η πιθανότητα απόρριψης πελάτη από το σύστημα (Blocking Probability)

```
Blocking propability =
0.092762
>> |
```

- Ο μέσος χρόνος καθυστέρησης ενός πελάτη στο σύστημα

```
Average delay time =
1.1096
```

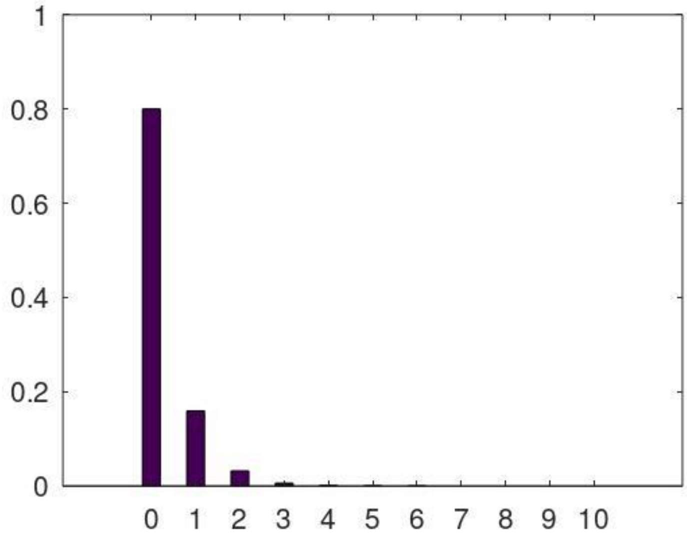
- 

- 

2.

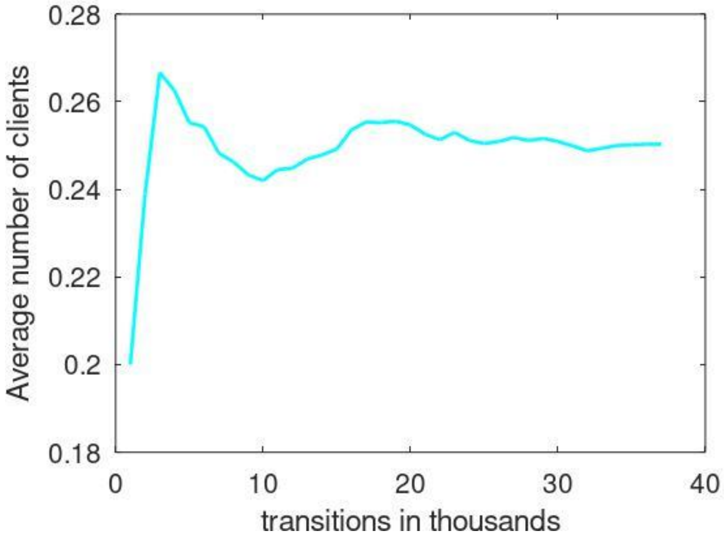
- Προσομοίωση για λ=1
  Γραφική παράσταση εργοδικών πιθανοτήτων:

Probabilities, lambda=1



Γραφική παράσταση εξέλιξης του μέσου αριθμού πελατών στο σύστημα:

e number of clients in the M/M/1 queue: Convergence,

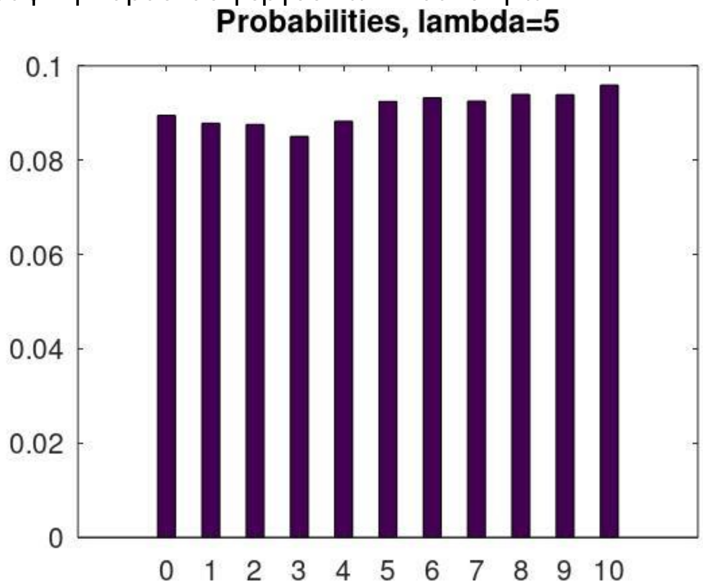


Κώδικας octave:

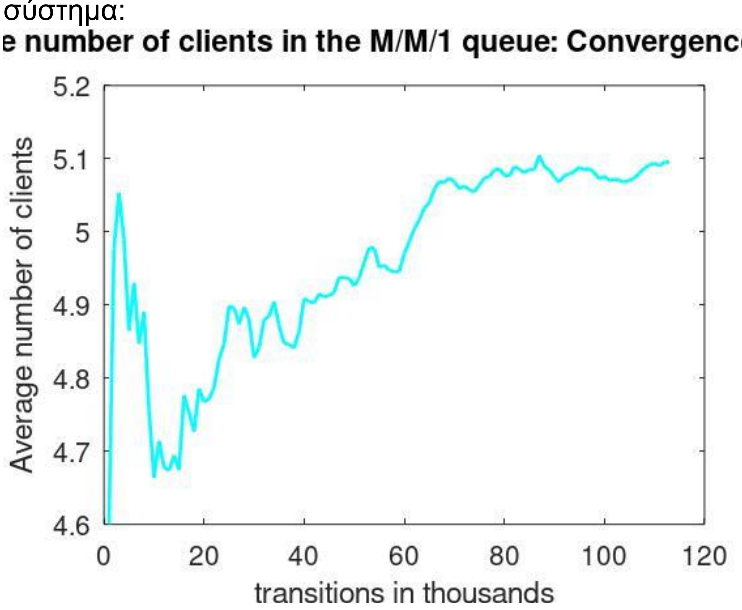
```
%2
%λ=1

clc;
clear all;
close all;

rand("seed",1);

P = [0,0,0,0,0,0,0,0,0,0,0];
arrivals = [0,0,0,0,0,0,0,0,0,0,0];
total_arrivals = 0; % to measure the total number of arrivals
current_state = 0;  % holds the current state of the system
previous_mean_clients = 0; % will help in the convergence test
index = 0;

lambda = 1;
mu = 5;
threshold = lambda/(lambda + mu); % the threshold used to calculate probabilities

transitions = 0; % holds the transitions of the simulation in transitions steps

while transitions >= 0
  transitions = transitions + 1; % one more transitions step

  if mod(transitions,1000) == 0 % check for convergence every 1000 transitions steps
    index = index + 1;
    for i=1:1:length(arrivals)
      P(i) = arrivals(i)/total_arrivals; % calcuate the probability of every state
    endfor

    mean_clients = 0; % calculate the mean number of clients in the system
    for i=1:1:length(arrivals)
       mean_clients = mean_clients + (i-1).*P(i);
    endfor

    to_plot(index) = mean_clients;

    if abs(mean_clients - previous_mean_clients) < 0.00001 || transitions > 1000000
      break;
    endif

    previous_mean_clients = mean_clients;

  endif

  random_number = rand(1); % generate a random number (Uniform distribution)
  if current_state == 0 || random_number < threshold % arrival
    if current_state < 11
      total_arrivals = total_arrivals + 1;
      arrivals(current_state + 1) = arrivals(current_state + 1) + 1; % increase the
      if current_state < 10
        current_state = current_state + 1;
      endif
    endif
  else % departure
    if current_state != 0 % no departure from an empty system
      current_state = current_state - 1;
    endif
  endif
endwhile



display("State propabilities:");
for i=1:1:length(arrivals)
  display(P(i));
endfor

g = lambda*(1-P(11));
average_delay_time = mean_clients / g;
display("Average delay time =");
disp(average_delay_time);
display("Blocking propability =");
disp(P(11));

figure(1);
plot(to_plot,"c","linewidth",1.3);
title("Average number of clients in the M/M/1 queue: Convergence, lambda=1");
xlabel("transitions in thousands");
ylabel("Average number of clients");

x=[0,1,2,3,4,5,6,7,8,9,10];
figure(2);
bar(x,P,0.4);
title("Probabilities, lambda=1")
display(transitions);

```

- Προσομοίωση για λ=5

  Γραφική παράσταση εργοδικών πιθανοτήτων:



  Γραφική παράσταση εξέλιξης του μέσου αριθμού πελατών στο σύστημα:

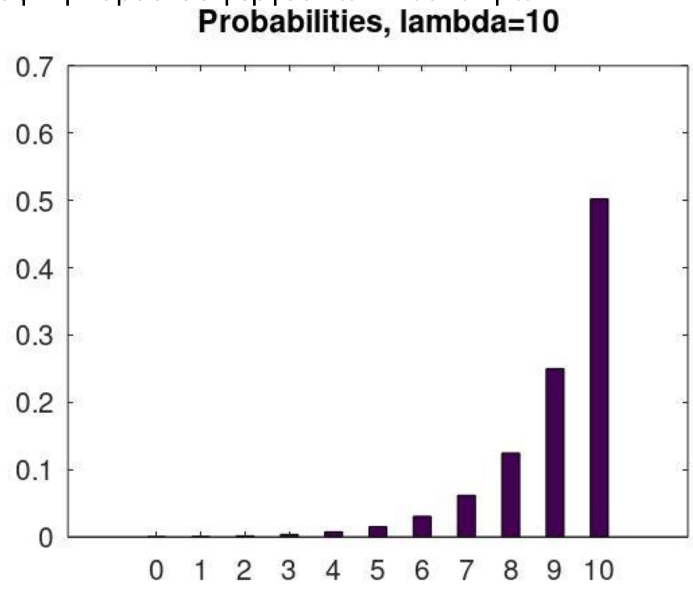


  Κώδικας octave:

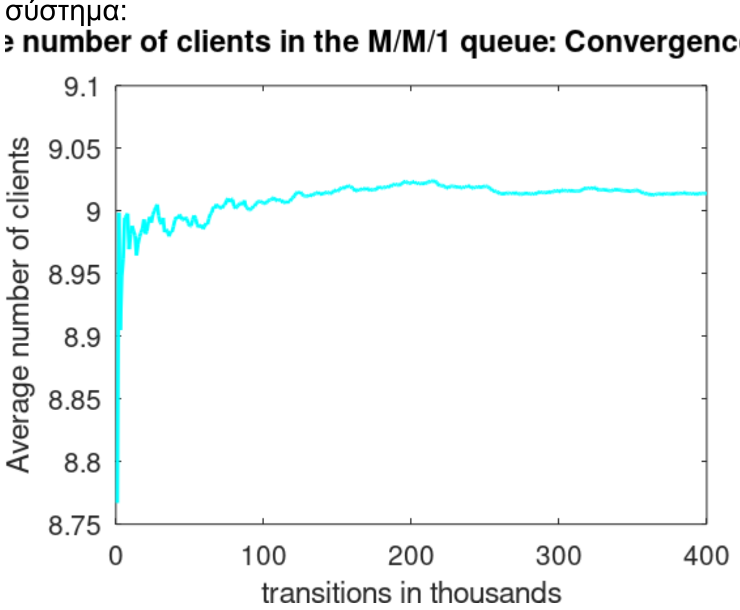
```
%λ=5
clc;
clear all;
close all;

rand("seed",1);

P = [0,0,0,0,0,0,0,0,0,0,0];
arrivals = [0,0,0,0,0,0,0,0,0,0,0];
total_arrivals = 0; % to measure the total number of arrivals
current_state = 0;  % holds the current state of the system
previous_mean_clients = 0; % will help in the convergence test
index = 0;

lambda = 5;
mu = 5;
threshold = lambda/(lambda + mu); % the threshold used to calculate probabilities

transitions = 0; % holds the transitions of the simulation in transitions steps

while transitions >= 0
  transitions = transitions + 1; % one more transitions step

  if mod(transitions,1000) == 0 % check for convergence every 1000 transitions steps
    index = index + 1;
    for i=1:1:length(arrivals)
        P(i) = arrivals(i)/total_arrivals; % calcuate the probability of every state i
    endfor

    mean_clients = 0; % calculate the mean number of clients in the system
    for i=1:1:length(arrivals)
       mean_clients = mean_clients + (i-1).*P(i);
    endfor

    to_plot(index) = mean_clients;

    if abs(mean_clients - previous_mean_clients) < 0.00001 || transitions > 1000000 %
      break;
    endif

    previous_mean_clients = mean_clients;

  endif

  random_number = rand(1); % generate a random number (Uniform distribution)
  if current_state == 0 || random_number < threshold % arrival
    if current_state < 11
      total_arrivals = total_arrivals + 1;
      arrivals(current_state + 1) = arrivals(current_state + 1) + 1; % increase the nu
      if current_state < 10
        current_state = current_state + 1;
      endif
    endif
  else % departure
    if current_state != 0 % no departure from an empty system
      current_state = current_state - 1;
    endif
  endif
endwhile



display("State propabilities:");
for i=1:1:length(arrivals)
  display(P(i));
endfor

g = lambda*(1-P(11));
average_delay_time = mean_clients / g;
display("Average delay time =");
disp(average_delay_time);
display("Blocking propability =");
disp(P(11));

figure(1);
plot(to_plot,"c","linewidth",1.3);
title("Average number of clients in the M/M/1 queue: Convergence, lambda=5");
xlabel("transitions in thousands");
ylabel("Average number of clients");

x=[0,1,2,3,4,5,6,7,8,9,10];
figure(2);
bar(x,P,0.4);
title("Probabilities, lambda=5")
display(transitions);
```

- Προσομοίωση για λ=10

  Γραφική παράσταση εργοδικών πιθανοτήτων:



  Γραφική παράσταση εξέλιξης του μέσου αριθμού πελατών στο σύστημα:



  Κώδικας octave:

```
%λ=10
clc;
clear all;
close all;

rand("seed",1);

P = [0,0,0,0,0,0,0,0,0,0,0];
arrivals = [0,0,0,0,0,0,0,0,0,0,0];
total_arrivals = 0; % to measure the total number of arrivals
current_state = 0;  % holds the current state of the system
previous_mean_clients = 0; % will help in the convergence test
index = 0;

lambda = 10;
mu = 5;
threshold = lambda/(lambda + mu); % the threshold used to calculate probabilities

transitions = 0; % holds the transitions of the simulation in transitions steps

while transitions >= 0
  transitions = transitions + 1; % one more transitions step

  if mod(transitions,1000) == 0 % check for convergence every 1000 transitions steps
    index = index + 1;
    for i=1:1:length(arrivals)
        P(i) = arrivals(i)/total_arrivals; % calcuate the probability of every state i
    endfor

    mean_clients = 0; % calculate the mean number of clients in the system
    for i=1:1:length(arrivals)
       mean_clients = mean_clients + (i-1).*P(i);
    endfor

    to_plot(index) = mean_clients;

    if abs(mean_clients - previous_mean_clients) < 0.00001 || transitions > 1000000 %
      break;
    endif

    previous_mean_clients = mean_clients;

  endif

  random_number = rand(1); % generate a random number (Uniform distribution)
  if current_state == 0 || random_number < threshold % arrival
    if current_state < 11
      total_arrivals = total_arrivals + 1;
      arrivals(current_state + 1) = arrivals(current_state + 1) + 1; % increase the nu
      if current_state < 10
        current_state = current_state + 1;
      endif
    endif
  else % departure
    if current_state != 0 % no departure from an empty system
      current_state = current_state - 1;
    endif
  endif
endwhile



display("State propabilities:");
for i=1:1:length(arrivals)
  display(P(i));
endfor

g = lambda*(1-P(11));
average_delay_time = mean_clients / g;
display("Average delay time =");
disp(average_delay_time);
display("Blocking propability =");
disp(P(11));

figure(1);
plot(to_plot,"r","linewidth",1.3);
title("Average number of clients in the M/M/1 queue: Convergence, lambda=10");
xlabel("transitions in thousands");
ylabel("Average number of clients");

x=[0,1,2,3,4,5,6,7,8,9,10];
figure(2);
bar(x,P,0.4);
title("Probabilities, lambda=10")
display(transitions);
```

3. Χρησιμοποιούμε την τελευταία εντολή του κάθε κώδικα για κάθε τιμή του λ για να υπολογίσουμε και να εμφανίσουμε το αποτέλεσμα των μεταβάσεων που χρειάζεται το σύστημα για να επέλθει σε ισορροπία. Προκύπτουν:

Για λ=1

```
transitions = 37000
>>
```

Για λ=5

```
transitions = 113000
>>
```

Για λ=10

```
transitions = 400000
>>
```

Παρατηρούμε ότι με την αύξηση του λ, η ταχύτητα σύγκλισης της προσομοίωσης μειώνεται, και άρα απαιτείται μεγαλύτερος αριθμός μεταβάσεων για την ικανοποίηση του κριτηρίου σύγκλισης. Αυτό είναι αναμενόμενο λόγω του τύπου $\rho = \frac{\lambda}{\mu}$, καθώς όσο αυξάνεται το λ και το μ παραμένει σταθερό τότε η ένταση του φορτίου ρ αυξάνεται. Αυτό σημαίνει ότι οι πελάτες φτάνουν με αυξημένο ρυθμό, ενώ το σύστημα διατηρεί τον ίδιο ρυθμό εξυπηρέτησης και άρα το σύστημα υπερφορτώνεται, κάτι το οποίο συνεπάγεται επίμηκυνση της μεταβατικής κατάστασης(της κατάστασης του συστήματος πριν επέλθει η εργοδική ισορροπία).
Παρατηρώντας τα παραπάνω διαγράμματα καταλήγουμε στο ότι οι αρχικές μεταβάσεις που μπορούν να αγνοηθούν ώστε να επιταχυνθεί η σύγκλιση της προσομοίωσης είναι:

- Για λ=1 πελάτες/λεπτό, 25000 μεταβάσεις
- Για λ=5 πελάτες/λεπτό, 90000 μεταβάσεις
- Για λ=10 πελάτες/λεπτό, 200000 μεταβάσεις

4. Εάν ο αριθμός εξυπηρέτησης μ ήταν μεταβλητός, εξαρτημένος από την κατάσταση στην οποία βρίσκεται το σύστημα i = {1, 2, ..., 10} και ίσος με $\mu_i = \mu * (i + 1)$ τότε θα επιλέξουμε και μεταβλητό κατώφλι ίσο με $threshold_i = i = \frac{\lambda}{\lambda+\mu_i} = \frac{\lambda}{\lambda+\mu(i+1)}$ όπου i είναι η κατάσταση στην οποία βρισκόμαστε, όπου το i ανήκει στην μεταβλητή current state. Στον κώδικα μετά από κάθε αλλαγή της μεταβλητής current state, θα πρέπει να υπολογίζεται ξανά το μ ως $mu = 5 * (current\ state + 1)$ καθώς και το threshold.