ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΣΙΤΩΝ
ΡΟΗ Δ - ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ (QUEUING SYSTEMS)

ΓΕΩΡΓΑΚΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ 03120827

ΑΝΑΦΟΡΑ 3ΗΣ ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ

Προσομοίωση συστήματος M/M/1/10

Το σύστημα που θα προσωμοιώσουμε έχει έναν εξυπηρετητή με μέγιστη χωρητικότητα 10 πελάτες. Οι αφίξεις ακολουθούν κατανομή Poisson με ομοιόμορφο μέσο ρυθμό λ πελάτες/min και οι εξυπηρετήσεις ακολοθούν εκθετική κατανομή με ομοιόμορφο ρυθμό εξυπηρέτησης μ=5 πελάτες/min. Για την προσωμοίηση θα χρησιμοποιήσουμε την Octave

ΕΡΩΤΗΜΑ 1

Επιλέγουμε για αρχή τις τιμές λ = 5 πελάτες/min και μ=5 πελάτες/min.
Για την υλοποίηση της προσωμοίωσης δημιουργούμε ένα while loop το οποίο θα τερματίσει όταν φτάσουμε τις 1.000.000 μεταβάσεις ή εάν η διαφορά 2 διαδοχικών μέσων αριθμών πελατών είναι μικρότερη από 0.001% , όπου κάθε μέσος αριθμός πελατών υπολογίζεται ανά 1000 μεταβάσεις.
Για τις αφίξεις ή τις αναχωρήσεις χρησιμοποιούμε έναν τυχαίο αριθμό μέσω της rand() όπου εάν είναι μικρότερος ή μεγαλύτερος από το threshold το οποίο έχουμε ορίσει ως λ/λ+μ τότε πρόκειται για άφιξη ή αναχώρηση αντίστοιχα.
Το threshold δηλώνει το διάστημα όπου έχουμε άφιξη (λ) προς το διάστημα πιθανών μεταβάσεων(λ+μ).
Επίσης υλοποιούμε κάποιους ελέγχους για τις περιπτώσεις όπου έχουμε αναχώηρηση από την κατάσταση 0 ή άφιξη στην κατάσταση 10.

Για το debugging της προσωμοίωσης παράγουμε λεπτομερές trace των μεταβάσεων της κατάστασης του συστήματος για τις πρώτες 30 μεταβάσεις:
Εμφανίζουμε: α) την κατάσταση β)την επόμενη μετάβαση γ)τον συνολικό αριθμό αφίξεων στην παρούσα κατάσταση

```
Transition Number = 1
Current state = 0
Next transition = Arrival
Total arrivals in current state =  0
Transition Number = 2
Current state = 1
Next transition = Arrival
Total arrivals in current state =  0
Transition Number = 3
Current state = 2
Next transition = Arrival
Total arrivals in current state =  0
Transition Number = 4
Current state = 3
Next transition = Arrival
Total arrivals in current state =  0
```

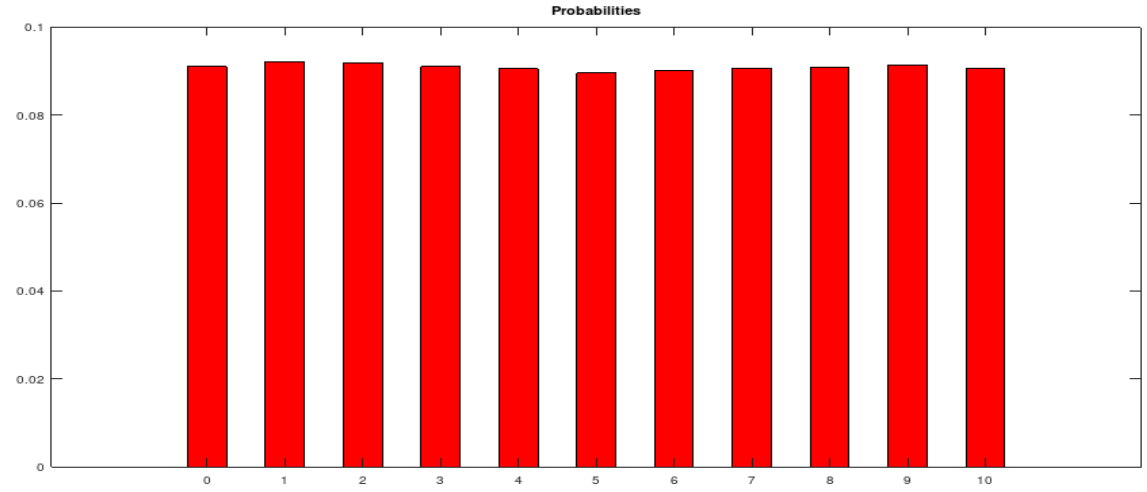
```
Transition Number = 5
Current state = 4
Next transition = Departure
Total arrivals in current state =  0
Transition Number = 6
Current state = 3
Next transition = Departure
Total arrivals in current state =  1
Transition Number = 7
Current state = 2
Next transition = Departure
Total arrivals in current state =  1
Transition Number = 8
Current state = 1
Next transition = Departure
Total arrivals in current state =  1
Transition Number = 9
Current state = 0
Next transition = Arrival
Total arrivals in current state =  1
Transition Number = 10
Current state = 1
Next transition = Arrival
Total arrivals in current state =  1
Transition Number = 11
Current state = 2
Next transition = Departure
Total arrivals in current state =  1
Transition Number = 12
Current state = 1
Next transition = Arrival
Total arrivals in current state =  2
Transition Number = 13
Current state = 2
Next transition = Arrival
Total arrivals in current state =  1
Transition Number = 14
Current state = 3
Next transition = Arrival
Total arrivals in current state =  1
Transition Number = 15
Current state = 4
Next transition = Departure
Total arrivals in current state =  0
Transition Number = 16
Current state = 3
Next transition = Departure
Total arrivals in current state =  2
Transition Number = 17
Current state = 2
Next transition = Arrival
Total arrivals in current state =  2
Transition Number = 18
Current state = 3
Next transition = Arrival
Total arrivals in current state =  2
Transition Number = 19
```

```
Current state = 4
Next transition = Arrival
Total arrivals in current state =  0
Transition Number = 20
Current state = 5
Next transition = Departure
Total arrivals in current state =  0
Transition Number = 21
Current state = 4
Next transition = Departure
Total arrivals in current state =  1
Transition Number = 22
Current state = 3
Next transition = Arrival
Total arrivals in current state =  3
Transition Number = 23
Current state = 4
Next transition = Arrival
Total arrivals in current state =  1
Transition Number = 24
Current state = 5
Next transition = Departure
Total arrivals in current state =  0
Transition Number = 25
Current state = 4
Next transition = Departure
Total arrivals in current state =  2
Transition Number = 26
Current state = 3
Next transition = Arrival
Total arrivals in current state =  4
Transition Number = 27
Current state = 4
Next transition = Arrival
Total arrivals in current state =  2
Transition Number = 28
Current state = 5
Next transition = Arrival
Total arrivals in current state =  0
Transition Number = 29
Current state = 6
Next transition = Arrival
Total arrivals in current state =  0
Transition Number = 30
Current state = 7
Next transition = Departure
Total arrivals in current state =  0
```

Επίσης, υπολογίζουμε τα ερωτήματα της εκφώνησης:

- Πιθανότητες των καταστάσεων του συστήματος

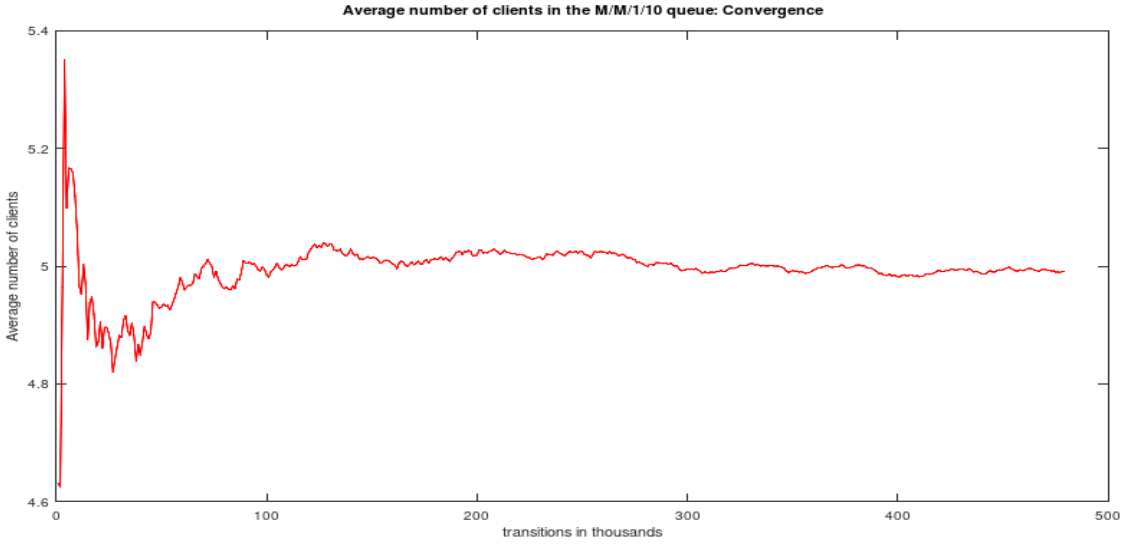
```
State propabilities:
state 0 -> 0.09105
state 1 -> 0.0922219
state 2 -> 0.0918393
state 3 -> 0.0910779
state 4 -> 0.0905558
state 5 -> 0.0895434
state 6 -> 0.0900855
state 7 -> 0.0907471
state 8 -> 0.090779
state 9 -> 0.091349
state 10 -> 0.0907511
```



- Πιθανότητα απόρριψης πελάτη από το σύστημα

```
Blocking propability = 0.0907511
```

- Μέσος αριθμός πελατών στο σύστημα



- Μέσος χρόνος καθυστέρησης ενός πελάτη στο σύστημα

```
Average delay time = 1.09776
```

Ο κώδικας που χρησιμοποιήθηκε για αυτό το ερώτημα είναι ο ακόλουθος:

```
clc;
clear all;
close all;
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

% initialization of probabilities and arrivals for each state (11 states because max customers=10)
P = zeros(1,11);
arrivals = zeros(1,11);

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 in the system
    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 % convergence test
      break;
    endif

    previous_mean_clients = mean_clients;

  endif
  %Εκτυπώνω τις πρώτες 30 μεταβάσεις
  random_number = rand(1); % generate a random number (Uniform distribution)
  if current_state == 0 || random_number < threshold % arrival
    %code for arrival
    % we dont have infinite capacity
    if current_state < 11
      total_arrivals = total_arrivals + 1;
      if transitions < 31
        fprintf('Transition Number = %d\n',transitions);
        fprintf('Current state = %d\n',current_state);
        display("Next transition = Arrival");
        fprintf('Total arrivals in current state =  %d\n',arrivals(current_state+1));
      endif
      arrivals(current_state + 1) = arrivals(current_state + 1) + 1; % increase the number of arrivals i
      if current_state < 10
        current_state = current_state + 1;
      endif
    endif
    %end code for arrival
  else % departure
    %code for departure
    if current_state != 0 % no departure from an empty system
      if transitions < 31
        fprintf('Transition Number = %d\n',transitions);
        fprintf('Current state = %d\n',current_state);
        display("Next transition = Departure");
        fprintf('Total arrivals in current state =  %d\n',arrivals(current_state+1));
      endif
      current_state = current_state - 1;
    endif
    %end code for departure
  endif
endwhile

%Average Time Delay and Blocking Probability
est = lambda*(1-P(11)); %estimate of the arrival rate during congestion
average_delay_time = mean_clients / est;
fprintf('Average delay time = %d\n',average_delay_time);
fprintf('Blocking propability = %d\n',P(11));
```

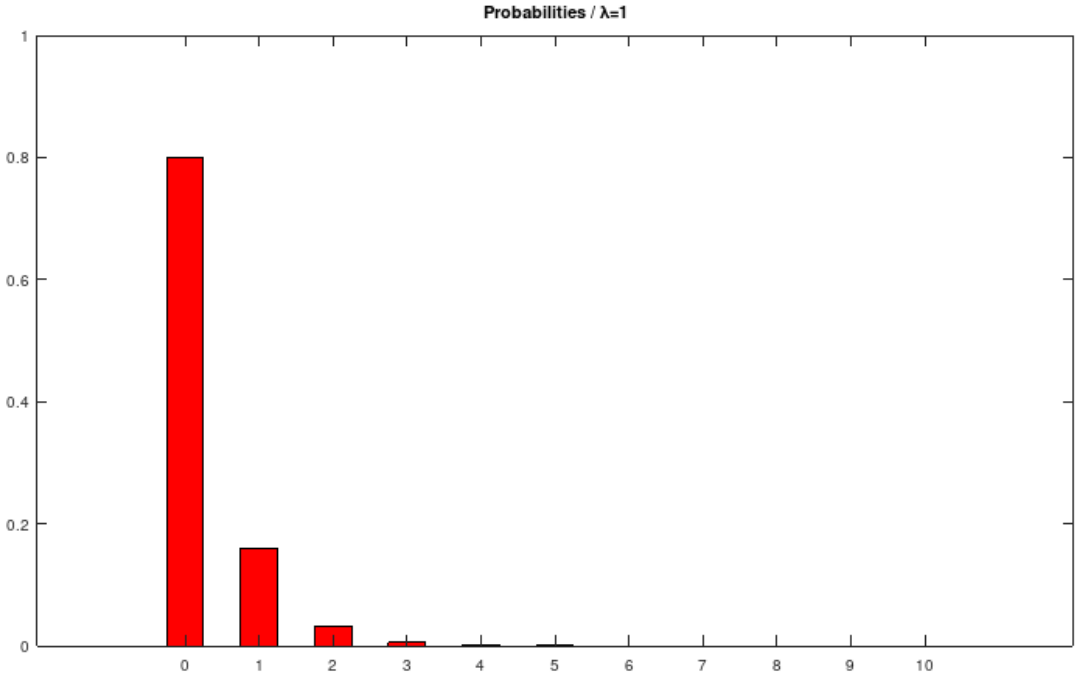
```
%Average number of clients in queue
figure(1);
plot(to_plot,"r","linewidth",1.3);
title("Average number of clients in the M/M/1/10 queue: Convergence");
xlabel("transitions in thousands");
ylabel("Average number of clients");

%Probabilities for every state
display("State propabilities:");
for i=1:1:length(arrivals)
  %display(P(i));
  fprintf('state %d -> %d\n',i-1,P(i)); %Το P ξεκινάει απο 1
endfor
%Plotting probabilities for every state
x = 0:10;
figure(2);
bar(x,P,'r',0.5);
title("Probabilities")
```

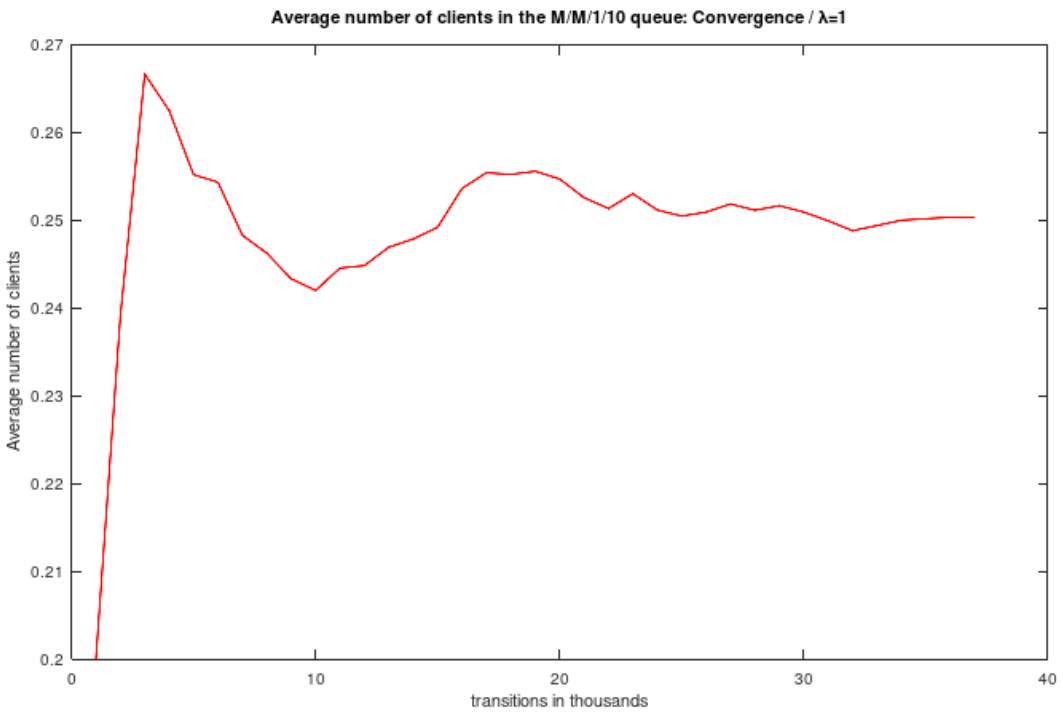
Σχόλιο: Χρησιμοποιήθηκε το αρχείο demo3.m

ΕΡΩΤΗΜΑ 2

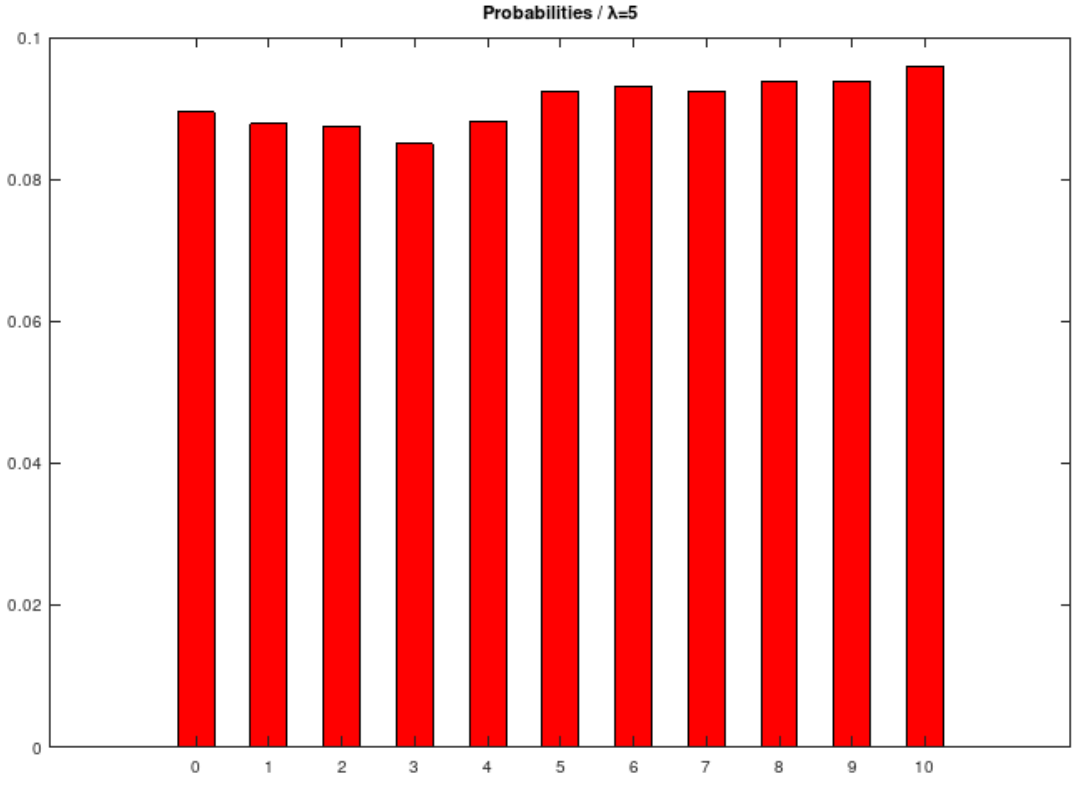
Εκτελούμε την ίδια προσομοίωση για τιμές λ=1 και λ=10:
Σε κάθε περίπτωση το μ παραμένει 5 πελάτες/min
Για κάθε τιμή του λ, εμφανίζουμε τα διαγράμματα πιθανοτήτων και μέσου αριθμού πελάτων στο σύστημα.

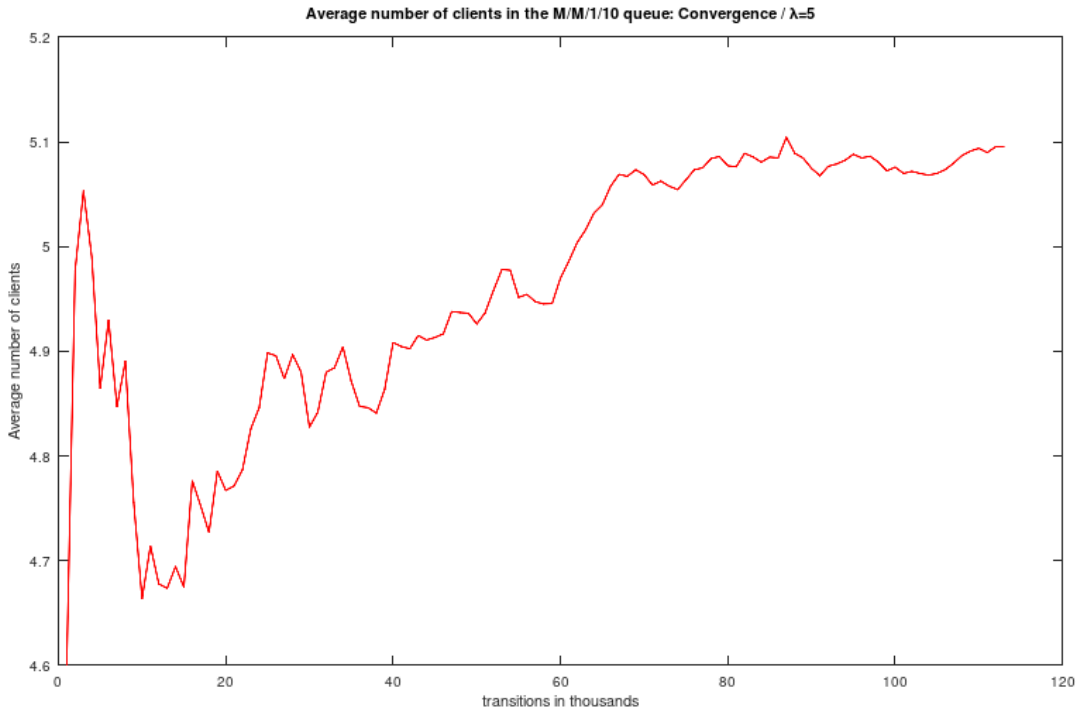
- λ = 1 πελάτες/min



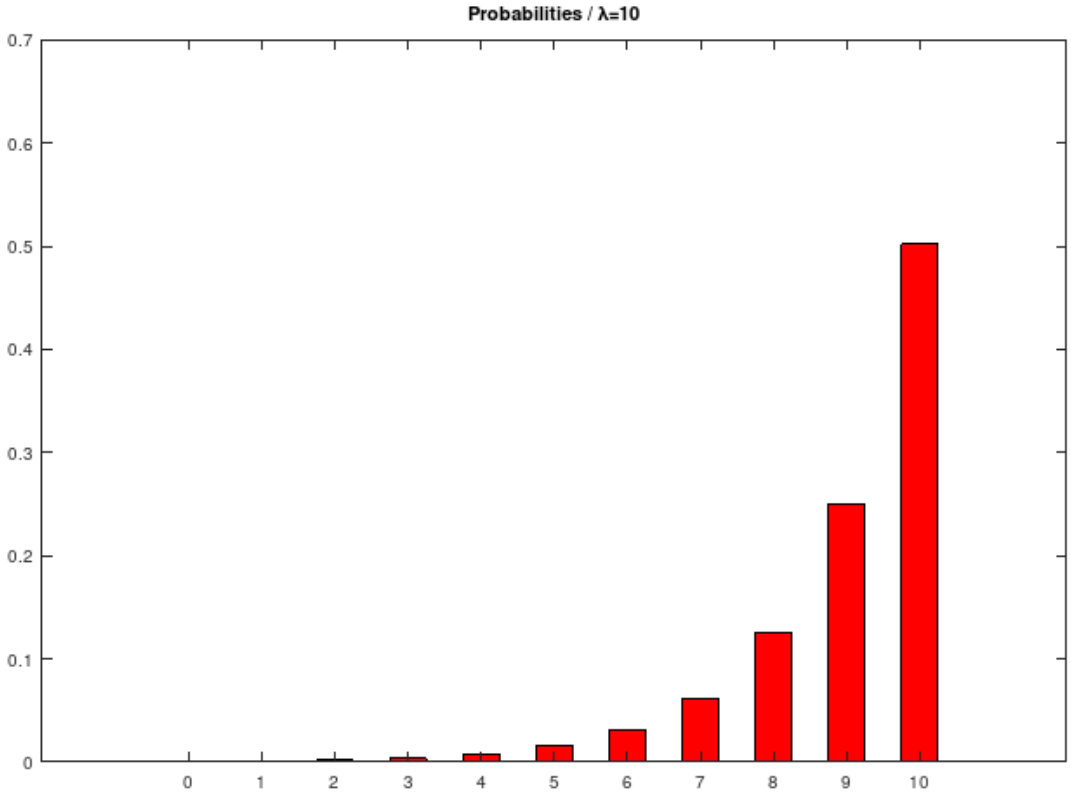


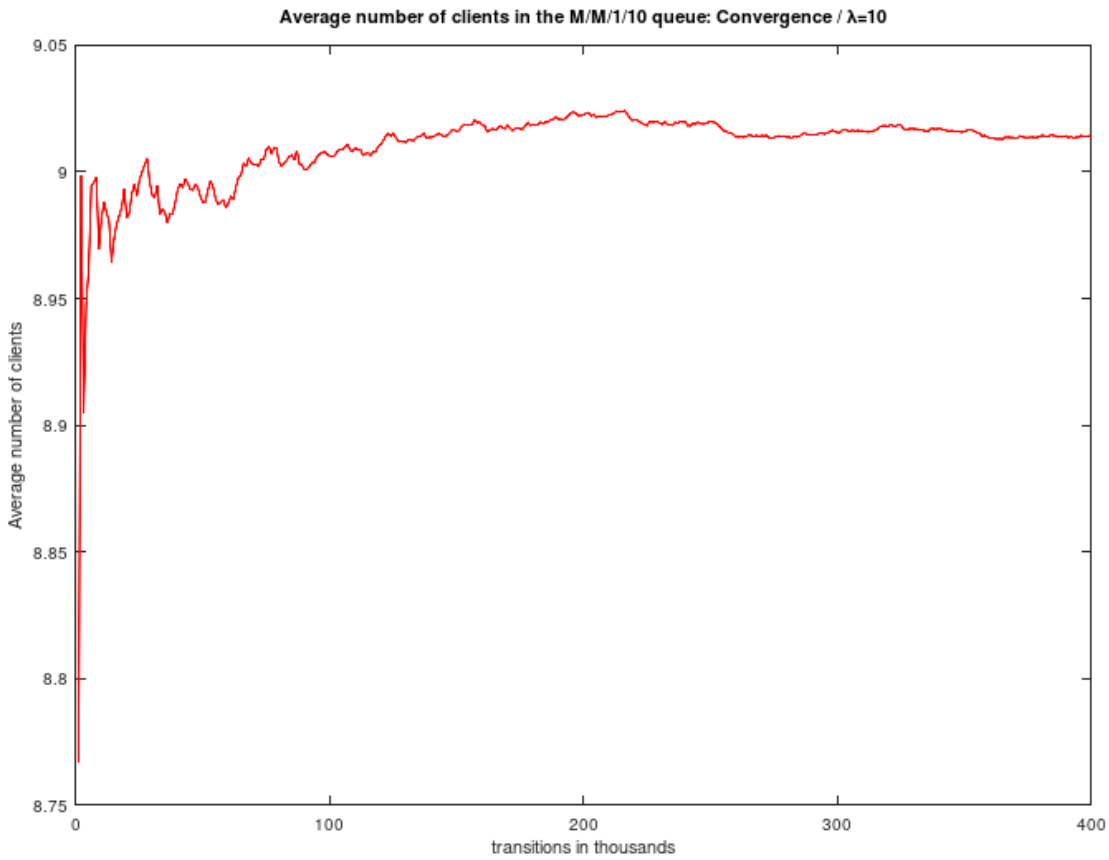
- λ= 5 πελάτες/min





- λ= 10 πελάτες/min





Ο κώδικας που χρησιμοποιήθηκε για αυτό το ερώτημα είναι ο ακόλουθος:

```
clc;
clear all;
close all;
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;
rand("seed",1);

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

% initialization of probabilities and arrivals for each state (11 states because max customers=10)
P = zeros(1,11);
arrivals = zeros(1,11);
```

```
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 in the system
    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 % convergence test
      break;
    endif

    previous_mean_clients = mean_clients;

  endif
```

```
  %Εκτυπώνω τις πρώτες 30 μεταβάσεις
  random_number = rand(1); % generate a random number (Uniform distribution)
  if current_state == 0 || random_number < threshold % arrival
    %code for arrival
    % we dont have infinite capacity
    if current_state < 11
      total_arrivals = total_arrivals + 1;
      arrivals(current_state + 1) = arrivals(current_state + 1) + 1; % increase the number of arrivals i
      if current_state < 10
        current_state = current_state + 1;
      endif
    endif
    %end code for arrival
  else % departure
    %code for departure
    if current_state != 0 % no departure from an empty system
      current_state = current_state - 1;
    endif
    %end code for departure
  endif
endwhile
% Tranzitions
fprintf('Transitions for λ= %d  are %d\n',lambda,transitions)

%Average Time Delay and Blocking Probability
est = lambda*(1-P(11)); %estimate of the arrival rate during congestion
average_delay_time = mean_clients / est;
fprintf('Average delay time = %d\n',average_delay_time);
fprintf('Blocking propability = %d\n',P(11));

%Average number of clients in queue
figure(1);
plot(to_plot,"r","linewidth",1.3);
title("Average number of clients in the M/M/1/10 queue: Convergence / λ=10");
xlabel("transitions in thousands");
ylabel("Average number of clients");
%Probabilities for every state
display("State propabilities:");
for i=1:1:length(arrivals)
  %display(P(i));
  fprintf('state %d -> %d\n',i-1,P(i)); %Το P ξεκινάει απο 1
endfor
%Plotting probabilities for every state
x = 0:10;
figure(2);
bar(x,P,'r',0.5);
title("Probabilities / λ=10")
```

Σχόλιο: Ο κώδικας είναι ίδιος για κάθε υποερώτημα, δηλαδή για κάθε διαφορετική τιμή του λ, με την μοναδική διαφορά στην γραμμή όπου ορίζουμε το λ.

ΕΡΩΤΗΜΑ 3

Καθώς μεγαλώνει η τιμή του λ, η ταχύτητα σύγκλισης της προσομοίωσης, δηλαδή ο απαιτούμενος αριθμός μεταβάσεων μέχρι να ικανοποιηθεί το κριτήριο σύγκλισης, μειώνεται Αυτό προκύπτει αν αναλογιστούμε τον τύπο της έντασης φορτίου ρ = λ/μ. Διαισθητικά, για μεγάλο λ, οι πελάτες φτάνουν στο σύστημα πολύ γρηγορότερα από ότι εξυπηρετούνται, δημιουργώντας έτσι συμφόρηση. Ετσι το σύστημα υπερφορτώνεται και χρειάζεται περισσότερος χρόνος να φτάσει σε εργοδική ισορροπία.
Μπορούμε να οδηγηθούμε σε αυτό το συμπέρασμα και μέσω του Octave. Στο τέλος κάθε κώδικα με την εντολή fprintf('Transitions for λ= %d are %d',lambda,transitions) μπορούμε να δούμε σε κάθε περίπτωση τις μεταβάσεις που χρειάστηκε για ικανοποιηθεί το κριτήριο σύγκλισης. Παίρνουμε τα εξής αποτελέσματα:

```
Transitions for λ= 1  are 37000
Transitions for λ= 5  are 113000
Transitions for λ= 10  are 400000
```

Σύμφωνα με τα παραπάνω αποτελέσματα ισχύει ότι όσο μεγαλώνει το λ τόσες περισσότερες μεταβάσεις χρειάζονται για να ικανοποιηθεί το κριτήριο σύγκλισης. Επόμενως η ταχύτητα σύγκλισης μειώνεται.

Επιπλέον, για να απαντήσουμε το ερώτημα πόσες είναι οι αρχικές μεταβάσεις που μπορούν να αγνοηθούν ώστε να επιταχυνθεί η σύγκλιση της προσομοίωσης, πρέπει να μελετήσουμε τα διαγράμματα του προηγούμενου ερωτήματος. Θέλουμε να αγνοήσουμε τις αρχικές μεταβάσεις όπου παρατηρούμε ανωμαλίες. Επομένως θα μπορούσαμε να αγνοήσουμε:

- Για λ=1 τις πρώτες 20.000 μεταβάσεις
- Για λ=5 τις πρώτες 90.000 μεταβάσεις
- Για λ=10 τις πρώτες 130.000 μεταβάσεις

Σχόλιο: Για να συγκρίνουμε σωστά τις επιμέρους προσομοιώσεις χρησιμοποιήσαμε την εντολή rand("seed",1) στην αρχή του προγράμματος.

ΕΡΩΤΗΜΑ 4

Εάν το σύστημα μάς είχε εκθετικές εξυπηρετήσεις με μεταβλητό μέσο ρυθμό εξυπηρέτησης $\mu_i = \mu \cdot (i+1)$ όπου i ={1,2...10}, τότε το threshold θα είναι επίσης μεταβλητο:

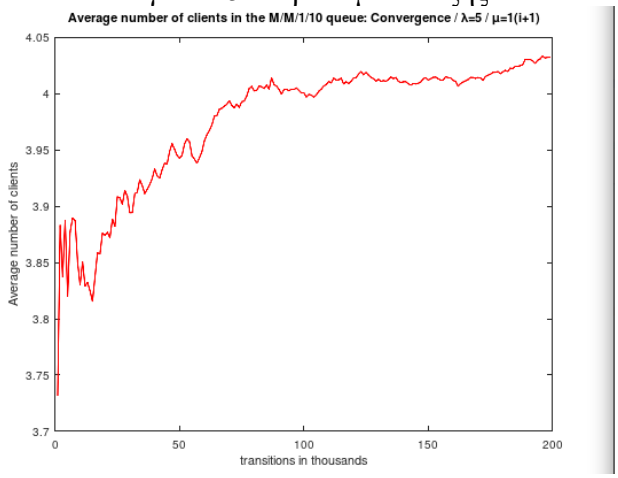
$$threshold = \frac{\lambda}{\lambda + \mu_i} = \frac{\lambda}{\lambda + \mu(i+1)}$$

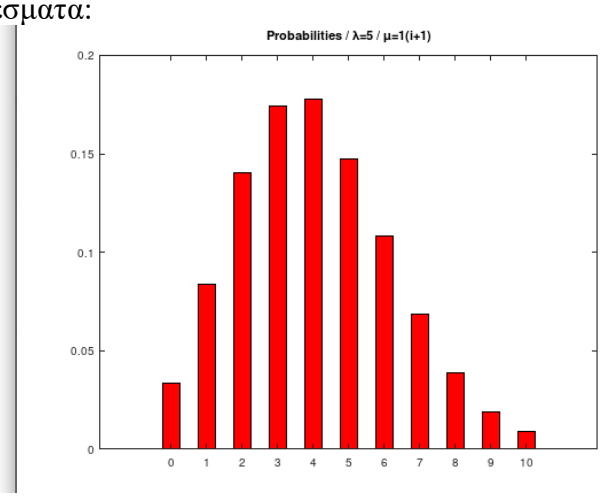
Στο πρόγραμμα θα μπορούσαμε να ορίσουμε το mu ως:

$$mu = mu_{stat} \cdot (current\ state + 1)\ ,\ mu_{stat} = 1$$

Ετσι σε κάθε νέα μετάβαση (δηλαδή σε κάθε νέο loop) θα υπολογίζουμε το καινούργιο mu και το καινουργιο threshold.

Ενδεικτικά για λ=5 παίρνουμε τα εξής αποτελέσματα:





Ο κώδικας που χρησιμοποιήθηκε για αυτό το ερώτημα είναι ο ακόλουθος:

```
clc;
clear all;
close all;
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;
rand("seed",1);

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

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

% initialization of probabilities and arrivals for each state (11 states because max customers=10)
P = zeros(1,11);
arrivals = zeros(1,11);
while transitions >= 0
  transitions = transitions + 1; % one more transitions step
  mu = mustat*(current_state+1);
  threshold = lambda/(lambda + mu); % the threshold used to calculate probabilities

  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 in the system
    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 % convergence test
      break;
    endif
```

```
    previous_mean_clients = mean_clients;

  endif

  %Εκτυπώνω τις πρώτες 30 μεταβάσεις
  random_number = rand(1); % generate a random number (Uniform distribution)
  if current_state == 0 || random_number < threshold % arrival
    %code for arrival
    % we dont have infinite capacity
    if current_state < 11
      total_arrivals = total_arrivals + 1;
      arrivals(current_state + 1) = arrivals(current_state + 1) + 1; % increase the number of arrivals in the current state
      if current_state < 10
        current_state = current_state + 1;
      endif
    endif
    %end code for arrival
  else % departure
    %code for departure
    if current_state != 0 % no departure from an empty system
      current_state = current_state - 1;
    endif
    %end code for departure
  endif
endwhile
% Tranzitions
fprintf('Transitions for λ= %d  are %d\n',lambda,transitions)

%Average Time Delay and Blocking Probability
est = lambda*(1-P(11)); %estimate of the arrival rate during congestion
average_delay_time = mean_clients / est;
fprintf('Average delay time = %d\n',average_delay_time);
fprintf('Blocking propability = %d\n',P(11));

%Average number of clients in queue
figure(1);
plot(to_plot,"r","linewidth",1.3);
title("Average number of clients in the M/M/1/10 queue: Convergence / λ=5 / μ=1(i+1)");
xlabel("transitions in thousands");
ylabel("Average number of clients");
```

```
%Probabilities for every state
display("State propabilities:");
for i=1:1:length(arrivals)
  %display(P(i));
  fprintf('state %d -> %d\n',i-1,P(i)); %Το P ξεκινάει απο 1
endfor
%Plotting probabilities for every state
x = 0:10;
figure(2);
bar(x,P,'r',0.5);
title("Probabilities / λ=5 / μ=1(i+1)")
```