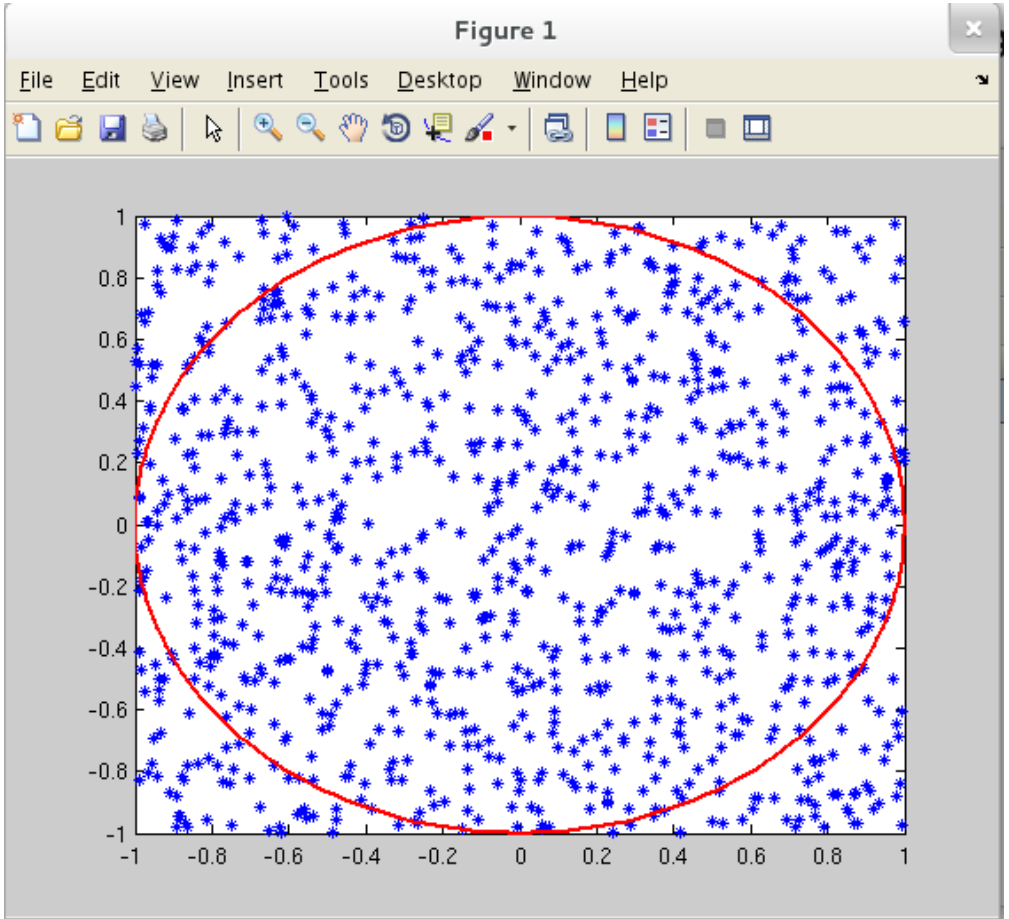
Προσομοίωση (simulation) στο Matlab

Monte Carlo simulation:

Μια γεννήτρια τυχαίων αριθμών μπορεί να χρησιμοποιηθεί για μια εκτίμηση του π ως εξής. Γράψτε ένα script που παράγει τυχαία σημεία σ'ένα τετράγωνο με πλευρές μήκους 2 και μετρά το ποσοστό των σημείων που πέφτουν μέσα σε ένα εγγεγραμμένο κύκλο. Το ποσοστό αυτό είναι ο λόγος του εμβαδού του κύκλου προς το εμβαδό του τετραγώνου. Από τον λόγο αυτό παίρνουμε μια προσέγγιση του π.



```
clear all; close all
circle = 0;
square = 10000;
for i = 1 : square
    x = 2 * rand - 1;
    y = 2 * rand - 1;
    if (x * x + y * y) < 1
        circle = circle + 1;
    end
    plot(x,y,'*')
    hold on
end
disp( circle / square * 4 );
th=0:pi/100:2*pi;
plot(sin(th),cos(th),'LineWidth',2,'Color','red')
```

Κυκλοφορία αυτοκινήτων σε φωτεινό σηματοδότη

Μια εφαρμογή της προσομοίωσης είναι στην κυκλοφοριακή ροή αυτοκινήτων σε μεγάλες πόλεις. Εδώ θα δούμε πως να προσομοιώσουμε μια μόνο ροή διαμέσου ενός φωτεινού σηματοδότη με τις εξής παραδοχές.

- Η ροή είναι ευθεία χωρίς στροφές.
- Η πιθανότητα ένα αυτοκίνητο να φθάνει στα φώτα σε ένα τυχαίο δευτερόλεπτο είναι ανεξάρτητη από το τι συνέβει το προηγούμενο δευτερόλεπτο. (Poisson διαδικασία). Η πιθανότητα αυτή p εκτιμάται παρατηρώντας τα αυτοκίνητα που φθάνουν στα φώτα και καταγράφοντας τον ρυθμό. Στην προσωμοίωση θα πάρουμε p=0.3
- Όταν τα φώτα είναι πράσινα, υποθέτουμε ότι τα αυτοκίνητα κινούνται σε συνεχή ροή οκτώ για κάθε 10 δευτερόλεπτα.
- Θα πάρουμε ως βασική χρονική περίοδο τα δέκα δευτερόλεπτα, οπότε θέλουμε να δούμε την ουρά που σχηματίζεται στα φώτα (αν υπάρχει) κάθε 10 δευτερόλεπτα.
- Θα ανάβουμε τα φώτα κόκκινα-πράσινα σε μεταβλητά πολλαπλάσια των 10 δευτερολέπτων.

- Τα φώτα είναι κόκκινα για 40 δευτερόλεπτα (red=4) και πράσινα (green=2) για 20 δευτερόλεπτα.
- Το μοντέλο μας θα έχει ένα βασικό script (traffic.m) που θα καλεί τρεις συναρτήσεις go.m , stop.m , prq.m. Επειδή οι συναρτήσεις χρειάζονται τις τιμές κάποιων μεταβλητών που ορίζονται στο traffic.m οι μεταβλητές αυτές δηλώνονται ως global στο traffic.m και αναλόγως στις τρεις συναρτήσεις.

```
clc
clear                                   % σβήσε όλα τα προηγούμενα!
global CARS GTIMER GREEN LIGHTS RED RTIMER T
CARS = 0;                               % αριθμός αυτοκινήτων στην ουρά
GTIMER = 0;                             % μετρητής για πράσινα φώτα
GREEN = 2;                              % περίοδος των φώτων όταν είναι πράσινα
LIGHTS = 'R';                           % χρώμα των φώτων
n = 48;                                 % αριθμός περιόδων των 10-sec
p = 0.3;                                % πιθανότητα να φτάνει ένα αυτοκίνητο
RED = 4;                                % περίοδος φώτων όταν είναι κόκκινα
RTIMER = 0;                             % μετρητής κόκκινων φώτων
for T = 1:n                             % για κάθε περίοδο των 10-sec
   r = rand(1,10);                      % 10 seconds means 10 random numbers
   CARS = CARS + sum(r < p);            % αυτοκίνητα που φτάνουν σε 10 sec
if LIGHTS == 'G'
   go                                   % συνάρτηση για τα πράσινα φώτα
else
   stop                                 % συνάρτηση για τα κόκκινα φώτα
end;
end;
```

```
function stop
global LIGHTS RED RTIMER
RTIMER = RTIMER + 1;     % προχώρα τον κόκκινο μετρητή κατά 1
prq;                     % απεικόνισε τα αυτοκίνητα στην ουρά
if RTIMER == RED         % έλεγξε αν τα φώτα πρέπει να αλλάξουν
   LIGHTS = 'G';
   RTIMER = 0;
end;
```

```
function go
global CARS GTIMER GREEN LIGHTS
GTIMER = GTIMER + 1;     % προχώρα τον πράσινο μετρητή κατά 1
CARS = CARS - 8;         % άφησε 8 αυτοκίνητα να περάσουν
if CARS < 0              % ... μπορεί να ήρθαν < 8 αυτοκίνητα
   CARS = 0;
end;
prq;                     % απεικόνισε την ουρά των αυτοκινήτων
if GTIMER == GREEN       % έλεγξε αν τα φώτα πρέπει να αλλάξουν
   LIGHTS = 'R';
   GTIMER = 0;
end;
```

```
function prq
global CARS LIGHTS T
fprintf( '%3.0f ', T );            % απεικόνισε τον αριθμό της περιόδου
if LIGHTS == 'R'
   fprintf( 'R' );                 % απεικόνισε το χρώμα των φώτων
else
   fprintf( 'G' );
end;
for i = 1:CARS
    fprintf( '*' );                % σημείωσε με * κάθε αυτοκίνητο
end;
fprintf( '\n' )                    % νέα γραμμή εντολών
```

Αποσύνθεση ραδιενεργού άνθρακα

Ο ραδιενεργός άνθρακας 11 έχει ρυθμό αποσύνθεσης k=0.0338 ανά λεπτό, δηλαδή ένα άτομο C^{11} έχει πιθανότητα 3.38% να αποσυντεθεί σε κάθε ένα λεπτό.

- Ας υποθέσουμε ότι αρχίζουμε με 100 τέτοια άτομα. Θέλουμε να προσομοιώσουμε την τύχη τους σε μια περίοδο 100 λεπτών. Θέλουμε να απεικονίσουμε τα αποτελέσματα σε ένα ιστόγραμμα που να δείχνει πόσα άτομα παραμένουν χωρίς να έχουν αποσυντεθεί μετά από 1,2,...100 λεπτά.

- Πρέπει να προσομοιώσουμε πότε κάθε άτομο από τα 100 αποσυντίθεται. Αυτό μπορεί να γίνει για κάθε άτομο, παράγοντας ένα τυχαίο αριθμό r για κάθε από τα 100 λεπτά, μέχρι ή r>k (το άτομο αποσυντίθεται) ή τα 100 λεπτά τελείωσαν. Αν το άτομο αποσυντίθεται σε χρόνο t<100, επαυξάνουμε την συχνότητα κατανομής f(t) κατά 1. f(t) θα είναι ο αριθμός των ατόμων που αποσυντίθεται σε χρόνο t λεπτών.

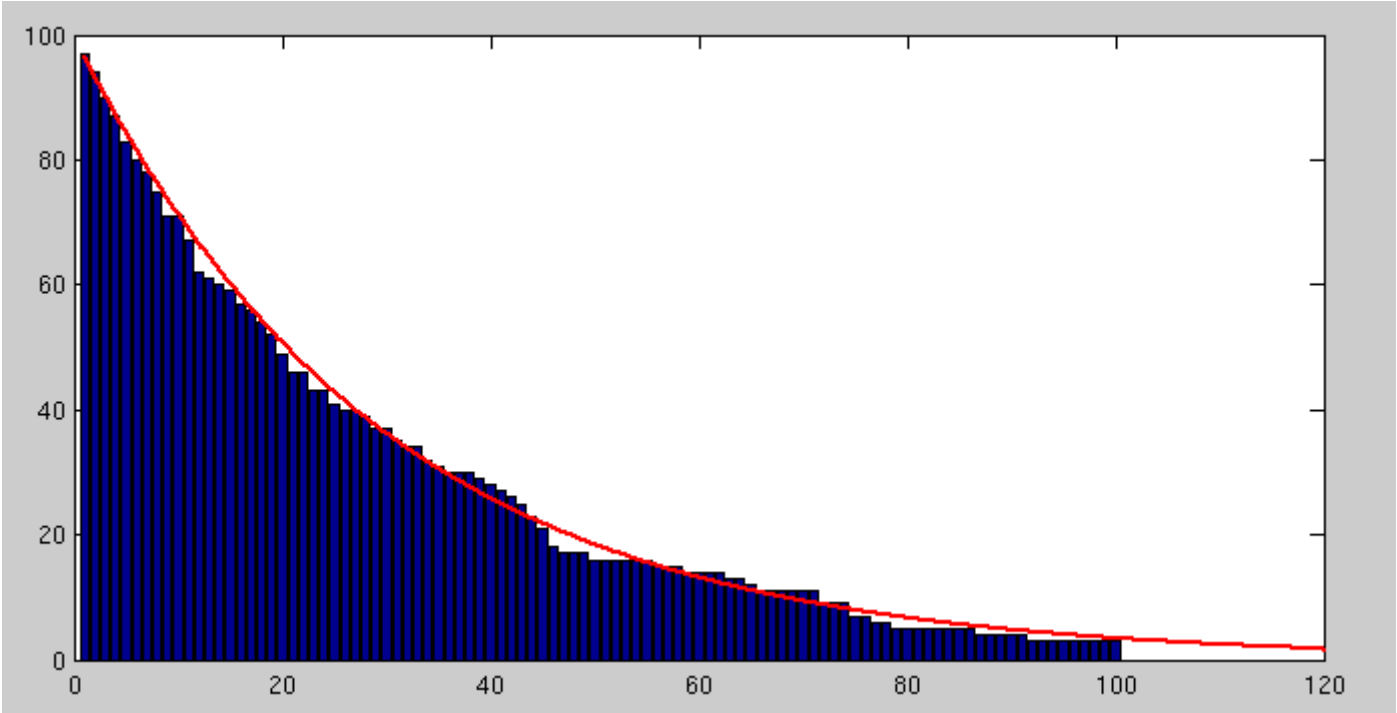
- Μετά θα μετατρέψουμε τον αριθμό των αποσυντιθεμένων ατόμων f(t) κάθε λεπτό στο αριθμό R(t) αυτών που παραμένουν χωρίς να έχουν αποσυντεθεί. Αν υπάρχουν n άτομα στην αρχή, μετά από ένα λεπτό ο αριθμός R(1) των ατόμων που παραμένουν είναι n-f(1), αφού f(1) είναι ο αριθμός των ατόμων που έχουν αποσυντεθεί στο πρώτο λεπτό. Ο αριθμός R(2) αυτών που παραμένουν μετά από 2 λεπτά θα είναι n-f(1)-f(2). Γενικά το R(t) είναι

  R(t) = n - sum(f(1:t))

Γράψτε ένα script που υπολογίζει το R(t) και απεικονίζει το ιστόγραμμά του. Στο ίδιο γραφικό σχεδιάστε την θεωρητική τιμή του R(t) η οποία είναι

$$R(t) = 100 \exp(- k t)$$

Τυπικά αποτελέσματα δίνονται από το παρακάτω σχήμα.



Script file

```
clear all;close all;
n=100;
f=zeros(1,100);
for A=1:n
    r=rand(1,100);
    t=find(r<=0.0338);
    if isempty(t)==0
        f(t(1))=f(t(1))+1;
    end
for T=1:100
    R(T)=n-sum(f(1:T));
end
end
bar(R)
hold on;
x=1:0.1:120;
y=100*exp(-0.0338.*x);
plot(x,y,'LineWidth',2,'Color','red')
hold off
```