

DETYRE KURSI

PUNOI: Anjeza Ndreca

Permbajtje	
Hyrje	3
1 Vektoret dhe Matricat	4
1.1 Vektoret	4
1.2 Veprime me Vektoret	5
1.3 Matricat	7
1.4 Detyra	9
2.SHEMBUJ RAND()	15
3 Shperndarja dhe llojet e tyre.....	16
3.1 Shperndarja Normale	16
3.2 Shperndarja Binomiale	18
3.3 Shperndarja Puasonit	20
3.4 Shperndarja Eksponenciale	22
3.5 Shperndarja Gjeometrike	23
3.6 Shperndarja Uniforme	24
3.7 Gjenerimi i shpërndarjeve diskrete duke përdorur Shpërndarjen Uniforme	26
3.8 Metoda Box-Muller	28
4 MATEMATIKA NE MATLAB	30
4.1 Funksionet ne Matematike	30
4.2 Statistika	31
4.3 Funksionet Trigonometrike	34
4.4 Funksionet Trigonometrik (Grafiku)	37
4.5 Regulla e kosinusit	41
4.6 Rrenjet Komplekse	42
4.7 Polinomet	43
5 TEOREMA QENDRORE LIMITE.....	45
6 CIKLET DHE KUSHTET E KONTROLLIT	47
6.1 Cikli For	47
6.2 Cikli While	48
6.3 Kushti if-elseif-else	49
6.4 Kushti switch-case-otherwise	49
7 Permbledhje	50

Ky projekt synon të demonstrojë përdorimin e MATLAB-it për të kryer operacione të ndryshme matematike dhe statistikore duke përdorur vektorë dhe matrica. Projekti përfshin:

Vektorë dhe Matrica: Përdorimi i operacioneve bazë me vektorë dhe matrica, përfshirë krijimin, manipulimin dhe operacionet aritmetike.

Statistika: Zbatimi i metodave statistikore për të analizuar të dhëna, duke përfshirë shpërndarje të ndryshme.

Funksionet Trigonometrike: Zbatimi i funksioneve trigonometrike për të kryer llogaritje të ndryshme.

Rrënjët Kompleks: Përdorimi i MATLAB-it për të gjetur dhe analizuar rrënjët komplekse të ekuacioneve.

Shpërndarjet Statistike: Analiza e shpërndarjeve të ndryshme si normale, eksponenciale, Poisson, binomiale, gama, gjeometrike dhe uniforme.

Ciklet dhe Kushtet e Funksionit: Demonstrimi i përdorimit të cikleve **for** dhe **while**, si dhe strukturave kondicionale **if**, **else** për zgjidhjen e problemeve.

Teorema Qendrore Limite: Ilustrimi i teoremës qendrore limite dhe zbatimi i saj në MATLAB.

Ky projekt ofron një pasqyrë të gjerë mbi përdorimin e MATLAB-it për zgjidhjen e problemeve matematike dhe statistikore, duke përfshirë vektorët, matricat, funksionet trigonometrike, rrënjët komplekse dhe shpërndarjet statistikore, si dhe strukturat kondicionale dhe ciklet.

1 VEKTORET DHE MATRICAT

1.1 Vektoret

Ushtrimi 1:

```
% Definimi i vektorit rresht x me numrat e rinj
x = [3, 7, 9, 10, 12]; % Numra shembullorë

% Llogaritja e sinusit për çdo element të vektorit rresht
y_rresht = sin(x);

% Shfaqja e rezultateve për vektorin rresht
disp('Vlerat e sinusit për vektorin rresht:');
disp(y_rresht);

% Definimi i vektorit shtyllë x me numrat e rinj
x = [3; 7; 9; 10; 12]; % Numra shembullorë, tani si vektor shtyll
% Llogaritja e sinusit për çdo element të vektorit shtyllë
y_shtylle = sin(x);

% Shfaqja e rezultateve për vektorin shtyllë
disp('Vlerat e sinusit për vektorin shtyllë:');
disp(y_shtylle);
```

```
>> untitled
Vlerat e sinusit për vektorin rresht:
    0.1411    0.6570    0.4121   -0.5440   -0.5366

Vlerat e sinusit për vektorin shtyllë:
    0.1411
    0.6570
    0.4121
   -0.5440
   -0.5366

>>
```

1.2 Veprimet me Vektoret

Ushtrimi 1:

```
% Definimi i vektorit rresht x me numrat e rinj
```

```

x = [3, 7, 9, 10, 12]; % Numra shembullorë

% Llogaritja e sinusit për çdo element të vektorit rresht
y_rresht = sin(x);

% Shfaqja e rezultateve për vektorin rresht
disp('Vlerat e sinusit për vektorin rresht:');
disp(y_rresht);

% Definimi i vektorit shtyllë x me numrat e rinj
x = [3; 7; 9; 10; 12]; % Numra shembullorë, tani si vektor shtyllë

% Llogaritja e sinusit për çdo element të vektorit shtyllë
y_shtyllë = sin(x);

% Shfaqja e rezultateve për vektorin shtyllë
disp('Vlerat e sinusit për vektorin shtyllë:');
disp(y_shtyllë);

```

```

>> untitled
c_squared =
     9     16     25     36     49

three_powers_c =
      27      81     243     729    2187

c_transposed =
     3
     4
     5
     6
     7

>>

```

- `c_squared` do të jetë vektori i rezultateve të ngritjes në katror të secilit element të `c`.
- `three_powers_c` do të tregojë rezultatet e ngritjes së numrit 3 në fuqi të secilit element të `c`.
- `c_transposed` do të jetë transpozimi i vektorit `c`, duke e shndërruar atë nga një vektor rresht në një vektor shtyllë.

Ky shembull pasqyron përdorimin e operatorit element-me-element (pikës) për ngritje në fuqi, si dhe funksionin e transpozimi

Ushtrimi 2

a) Të ndërtohet një vektor rresht me 10 elementë nga 1-20 në dy mënyra.

b) Vektori i krijuar të transpozohet në vektor shtyllë.

Menyra 1: Duke përdorur funksionin *randi*:

```
% Mënyra 1: Përdorimi i funksionit randi për të gjeneruar vektorin
```

```
vec1 = randi([1, 20], 1, 10);
```

```
disp('Vektori i krijuar me randi:');
```

```
disp(vec1);
```

```
>> untitled
Probabiliteti P(X=2) është: 0.372200
>> untitled
Vektori i krijuar me randi:
     2    16    17    19     3     3     2    18     5     5
>>
```

Menyra 2: Duke përdorur funksionin *randperm*

```
% Mënyra 2: Përdorimi i funksionit randperm për të gjeneruar vektorin
```

```
vec2 = randperm(20, 10);
```

```
disp('Vektori i krijuar me randperm:');
```

```
disp(vec2);
```

b) % Transpozimi i vec1 në vektor shtyllë

```
vec1_column = vec1';
```

```
disp('Vektori shtyllë i krijuar nga vec1:');
```

```
disp(vec1_column);
```

```
% Transpozimi i vec2 në vektor shtyllë
```

```
vec2_column = vec2';
```

```
disp('Vektori shtyllë i krijuar nga vec2:');
```

```
disp(vec2_column);
```

```
>> untitled
Vektori shtyllë i krijuar nga vec1:
    2
   16
   17
   19
    3
    3
    2
   18
    5
    5
```

1.3 Matricat

MATLAB, i njohur si "Matrix Laboratory", është i specializuar në manipulimin dhe përpunimin e matricave. Si një nga elementet kryesore të këtij mjedisi programimi, matrica mundëson llogaritje të ndërlikuara matematikore, analizë të dhënave dhe modelim numerik. Ky mjedis ofron një gamë të gjerë funksionesh të integruara për manipulimin e matricave, bërë kështu një mjet jetik për profesionistë në shumë fusha që merren me të dhëna komplekse.

Ushtrimi 1:

```
% Deklarimi i matricës origjinale
D = [2 4 6; 8 10 12; 14 16 18];

% Ndryshimi i elementeve të matricës
D(3,2) = 0; % Ndërrimi i elementit në rreshtin e 3, kolonën e 2 në zero

D(1,7) = 20; % Zgjerimi i matricës me një kolonë të re, vendoset vlere 20 në rreshtin e parë, kolona e shtatë

% Shfaqja e matricës D pas ndryshimeve
disp('Matrica D pas ndryshimeve:');

disp(D);

% Krijimi i matricës E nga rreshtat e përmbysur të D
E = D(3:-1:1, :); % Merr rreshtat në mënyrë të përmbysur

% Bashkimi i matricës D me disa kolona të E për të formuar matricën F
F = [D E(:, [1 3])];

% Shfaqja e matricës F
```

```

disp('Matrica F:');

disp(F);

% Seleksionimi dhe formimi i vektorit G
G = D([1 3], [2 2 2 2]);

% Shfaqja e vektorit G
disp('Vektori G:');

disp(G);

% Kthimi i matricës D në një vektor kolonë dhe transpozimi i tij
H = D(:);
H = H';

% Shfaqja e vektorit H pas transpozimit
disp('Vektori H pas transpozimit:');

disp(H);

% Përdorimi i operatorëve relacional për të selektuar elemente mbi 10
dhe formimi i vektorit I
I = D(D > 10);

% Shfaqja e vektorit I
disp('Elementet e matricës D që janë më të mëdha se 10:');

disp(I);

% Kontrolli dhe shfaqja e dimensioneve të matricës D
[rows, cols] = size(D);

disp(['Numri i rreshtave: ', num2str(rows), ', Numri i kolonave: ',
num2str(cols)]);

Vektori G:
    4    4    4    4
    0    0    0    0

Vektori H pas transpozimit:
    2    8   14    4   10    0    6   12   18    0    0    0    0    0    0    0    0    0    20    0    0

Elementet e matricës D që janë më të mëdha se 10:
    14
    12
    18
    20

Numri i rreshtave: 3, Numri i kolonave: 7
>>

```

1.4 Detyra

Ushtrimi 1

Ndërttoni një matricë A me 3 rreshta dhe 4 kolona.

- a) Zbrisni numrin 4 nga çdo element i matricës A.

```
% Krijo një matricë A me 3 rreshta dhe 4 kolona
A = randi([1, 10], 3, 4);

disp('Matrica A:');

disp(A); % Zbrisni numrin 4 nga çdo element i matricës A
A_minus_4 = A - 4;

disp('Matrica A pasi i është zbritur numri 4:');

disp(A_minus_4);

>> untitled
Matrica A:
     6     1     9     4
     2     6     7     7
     2     2     3     5

Matrica A pasi i është zbritur numri 4:
     2    -3     5     0
    -2     2     3     3
    -2    -2    -1     1

>>
```

- b) Shumëzoni çdo element të matricës A me numrin 3 dhe pastaj zbritini numrin 2 dhe emërtojeni matricën e krijuar me shkronjën B.

```
% Shumëzoni çdo element të matricës A me numrin 3 dhe pastaj
zbritini numrin 2
B = (A * 3) - 2;

disp('Matrica B (A * 3 - 2):');

disp(B);

>> untitled
Matrica B (A * 3 - 2):
    16     1    25    10
     4    16    19    19
     4     4     7    13

>>
```

- c) Shumëzoni elementët e matricës A me elementët e matricës B dhe ngrijini në katror çdo element të matricës së krijuar

```
% Shumëzoni elementët e matricës A me elementët e matricës B
```

```

C = A .* B;
disp('Matrica C (A .* B):');
disp(C);

% Ngrijeni në katror çdo element të matricës C
C_squared = C.^ 2;
disp('Matrica C pasi është ngritur në katror çdo element:');
disp(C_squared);

>> untitled
Matrica C (A .* B):
    96     1   225    40
     8    96   133   133
     8     8    21    65

Matrica C pasi është ngritur në katror çdo element:
   9216         1   50625   1600
    64    9216   17689   17689
    64     64    441    4225

>>

```

Ushtrimi 2

- a) Ndërttoni një matricë 4x5, elementët e së cilës janë 1.

```

% Krijë një matricë 4x5 me elementë 1
A = ones(4, 5);
disp('Matrica A (4x5 me elementë 1):');
disp(A);

>> untitled
Matrica A (4x5 me elementë 1):
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1

>>

```

- b) Ndërttoni një matricë tjetër 4x4 elementët e së cilës janë 0. Ndërttoni një matricë G me katër rreshta dhe 4 shtylla dhe një matricë C me njësha, dimensionin e së cilës të jetë i barabartë me dimensionin e matricës G.

```

% Krijë një matricë 4x4 me elementë 0
B = zeros(4, 4);
disp('Matrica B (4x4 me elementë 0):');
disp(B);

% Krijë një matricë G me katër rreshta dhe katër shtylla
(dimensionin e B)
G = randi([1, 10], 4, 4); % Krijë matricën G me elementë të
rastësishëm nga 1 deri në 10

```

```

disp('Matrica G (4x4 me elementë të rastësishëm):');
disp(G);

% Krijo një matricë C me njësha, dimensionin e cilës është i
barabartë me dimensionin e matricës G

C = ones(size(G));
disp('Matrica C (dimensionin 4x4 me elementë 1):');
disp(C);

>> untitled
Matrica B (4x4 me elementë 0):
    0    0    0    0
    0    0    0    0
    0    0    0    0
    0    0    0    0

Matrica G (4x4 me elementë të rastësishëm):
    6    1    2    3
    5    2    7    7
    2    6    5    2
    1    6    9    4

Matrica C (dimensionin 4x4 me elementë 1):
    1    1    1    1
    1    1    1    1
    1    1    1    1
    1    1    1    1

>> |

```

Ushtrimi 3

Ndërttoni një matricë G me përmasa 3x4.

% Krijo një matricë G me përmasa 3x4 me elementë të rastësishëm nga 1 deri në 10

```
G = randi([1, 10], 3, 4);
```

```
disp('Matrica G (3x4 me elementë të rastësishëm):');
```

```
disp(G);
```

```
>> untitled
```

```
Matrica G (3x4 me elementë të rastësishëm):
```

```

    9    1   10    1
    1    8    4    8
    8    9    6    9

```

```
>>
```

a) Zëvendësoni elementin në rreshtin e dytë dhe në shtyllën e katërt me 0.

```
% Zëvendësoni elementin në rreshtin e dytë dhe në shtyllën e katërt
me 0
```

```
G(2, 4) = 0;
```

```
disp('Matrica G pas zëvendësimit të elementit në rreshtin e dytë dhe  
në shtyllën e katërt me 0:');
```

```
disp(G);
```

```
>> untitled
```

```
Matrica G pas zëvendësimit të elementit në rreshtin e dytë dhe në shtyllën e katërt me 0:
```

9	1	10	1
1	8	4	0
8	9	6	9

```
>>
```

b) Matrica G të transformohet nga 3x4 në 4x6 dhe elementi në rreshtin e katërt dhe shtyllën e gjashtë të jetë 5.

```
% Transformoni matricën G nga 3x4 në 4x6
```

```
G = [G, zeros(3, 2); zeros(1, 6)];
```

```
G(4, 6) = 5;
```

```
disp('Matrica G pasi është transformuar në 4x6 dhe elementi në rreshtin  
e katërt dhe në shtyllën e gjashtë është 5:');
```

```
disp(G);
```

```
>> untitled
```

```
Matrica G pasi është transformuar në 4x6 dhe elementi në rreshtin e katërt dhe në shtyllën e gjashtë është 5:
```

9	1	10	1	0	0
1	8	4	0	0	0
8	9	6	9	0	0
0	0	0	0	0	5

```
>>
```

c) Ndryshoni vlerat e rreshtit të tretë me vlerën 7.

```
% Ndryshoni vlerat e rreshtit të tretë me vlerën 7
```

```
G(3, :) = 7;
```

```
disp('Matrica G pasi vlerat e rreshtit të tretë janë ndryshuar në  
7:');
```

```
disp(G);
```

```
>> untitled
```

```
Matrica G pasi vlerat e rreshtit të tretë janë ndryshuar në 7:
```

9	1	10	1	0	0
1	8	4	0	0	0
7	7	7	7	7	7
0	0	0	0	0	5

```
>>
```

d) Ndërtoni një matricë me shtyllat e matricës G në renditje të anasjelltë.

```
% Ndërtoni një matricë me shtyllat e matricës G në renditje të
anasjelltë
```

```
G_reversed = G(:, end:-1:1);
```

```
disp('Matrica G me shtyllat në renditje të anasjelltë:');
```

```
disp(G_reversed);
```

```
>> untitled
```

```
Matrica G me shtyllat në renditje të anasjelltë:
```

0	0	1	10	1	9
0	0	0	4	8	1
7	7	7	7	7	7
5	0	0	0	0	0

```
>>
```

e) Ndërtoni një matricë C me elementët e matricës G dhe me shtyllën e dytë dhe të tretë të matricës B.

```
% Krijo një matricë B me përmasa 4x4 me elementë të rastësishëm nga 1
deri në 10
```

```
B = randi([1, 10], 4, 4);
```

```
disp('Matrica B (4x4 me elementë të rastësishëm):');
```

```
disp(B);
```

```
% Ndërtoni një matricë C me elementët e matricës G dhe me shtyllën e
dytë dhe të tretë të matricës B
```

```
C = [G, B(:, 2:3)];
```

```
disp('Matrica C me elementët e matricës G dhe me shtyllën e dytë dhe
të tretë të matricës B:');
```

```
disp(C);
```

```
>> untitled
```

```
Matrica B (4x4 me elementë të rastësishëm):
```

8	7	4	8
7	3	6	7
7	7	2	5
10	5	4	1

```
Matrica C me elementët e matricës G dhe me shtyllën e dytë dhe të tretë të matricës B:
```

9	1	10	1	0	0	7	4
1	8	4	0	0	0	3	6
7	7	7	7	7	7	7	2
0	0	0	0	0	5	5	4

```
>>
```

f) Ndërtoni një matricë D me elementë nga matrica C që gjenden në dy rreshtat e fundit dhe në shtyllën e parë dhe të pestë.

```
% Ndërtoni një matricë D me elementë nga matrica C që gjenden në dy
rreshtat e fundit dhe në shtyllën e parë dhe të pestë
```

```

D = C(end-1:end, [1, 5]);

disp('Matrica D me elementë nga matrica C që gjenden në dy rreshtat e fundit dhe në shtyllën e parë dhe të pestë:');

disp(D);

>> untitled
Matrica D me elementë nga matrica C që gjenden në dy rreshtat e fundit dhe në shtyllën e parë dhe të pestë:
    7    7
    0    0

>>

```

g) Transformoni matricën G ne vektor shtyllë.

```

% Transformoni matricën G në vektor shtyllë

G_column_vector = G(:);

disp('Matrica G e transformuar në vektor shtyllë:');

disp(G_column_vector);

```

Ushtrimi 4

- a) Gjeneroni 5 zgjedhje me përmasë 4.
- b) Veçoni 4 ndryshoret e rastit të gjeneruara në zgjedhjen e katërt.

```

% a) Gjeneroj 5 zgjedhje me përmasë 4
y = zeros(5, 4); % rezervoj memorjen
for i = 1:5 % përdor ciklin for për ekzekutimin e komandave i-herë
    rand('state', i) % vendos gjendjen e gjeneratorit
    y(i, :) = rand(1, 4); % gjeneroj ndryshoret me anë të komandës 'rand'
end

% Shfaq zgjedhjet e gjeneruara

disp('5 zgjedhje me përmasë 4:')
disp(y)

% b) Veçoj ndryshoret e rastit të gjeneruara në zgjedhjen e katërt
% Fillimisht specifikoj gjendjen e gjeneratorit
rand('state', 4)
z = rand(1, 4); % gjeneroj ndryshoret e rastit për zgjedhjen e katërt

% Shfaq ndryshoret e veçuara
disp('Ndryshoret e gjeneruara në zgjedhjen e katërt:')
disp(z)

```

```
>> untitled
5 zgjedhje me përmasë 4:
    0.9528    0.7041    0.9539    0.5982
    0.8752    0.3179    0.2732    0.6765
    0.5162    0.2252    0.1837    0.2163
    0.4360    0.8212    0.6647    0.6455
    0.8639    0.9062    0.2001    0.8594

Ndryshoret e gjeneruara në zgjedhjen e katërt:
    0.4360    0.8212    0.6647    0.6455

>>
```

2.Shembuj rand()

```
% Inicializoj matricën për të mbajtur 4 zgjedhjet me përmasë 4 secila
x = zeros(4, 4);

% Përdor ciklin for për të gjeneruar numra të rastësishëm
for i = 1:4

    rng(i+10, 'twister'); % Vendos gjendjen e gjeneratorit të numrave
    të rastësishëm duke shtuar 10 për diversitet

    x(i, :) = rand(1, 4); % Gjeneroj 4 ndryshore të rastit për çdo
    rresht

end

% Shfaq matricën x
disp('Zgjedhjet e gjeneruara me 4 elementë secila:');

disp(x);

% Veço ndryshoret e rastit të gjeneruara në zgjedhjen e dytë kësaj
radhe

rng(12, 'twister'); % Rikthehem në gjendjen për zgjedhjen e dytë (i+10
ku i=2)

y = rand(1, 4); % Gjeneroj ndryshoret për zgjedhjen e dytë

% Shfaq ndryshoret e zgjedhjes së dytë
disp('Ndryshoret e rastit të zgjedhjes së dytë:');

disp(y);
```

Zgjedhjet e gjeneruara me 4 elementë secila:

0.1803	0.0195	0.4632	0.7249
0.1542	0.7400	0.2633	0.5337
0.7777	0.2375	0.8243	0.9657
0.5139	0.7732	0.8704	0.0080

Ndryshoret e rastit të zgjedhjes së dytë:

0.1542	0.7400	0.2633	0.5337
--------	--------	--------	--------

- **Inicializimi:** `x = zeros(4, 4)` tani krijojmë një matricë ku do të mbajmë katër zgjedhjet me katër elementë secila.
- **Cikli for:** Përdoret për të gjeneruar katër rreshta të ndryshoreve të rastësishëm, me ndryshimin që tani përdorim `rng(i+10, 'twister')` për të diversifikuar gjendjet e gjeneratorit.
- **Veçimi i Ndryshoreve:** Kësaj radhe, zgjedhja e dytë është veçuar duke përdorur `rng(12, 'twister')`, e cila korrespondon me gjendjen e gjeneratorit për iteracionin e dytë të ciklit kur `i=2`.

3 SHPERNDARJET DHE LLOJET E TYRE

3.1 Shperndarja Normale

Ushtrimi 1:

```
% Përcaktoj numrin e elementeve
n = 300; % Tjetër numër i elementeve

% Gjeneroj vektor shtyllë me n elemente nga shpërndarja normale
standard

p = randn(n, 1);

% Caktoj mesataren matematike të re
m = 10; % Një mesatare matematike tjetër

% Caktoj devijimin standard të ri
s = 3; % Një devijim standard tjetër

% Rregulloj vlerat e gjeneruara për të pasur mesataren e dëshiruar dhe
devijimin standard
```

```
p = m + s * p;
```

```
% Shfaq vlerat e rregulluara  
disp('Vlerat e rregulluara:');  
disp(p);
```

Ushtrimi 2:

```
% Përcakto vlerat e intervalit  
a1 = 50; % vlera minimale  
a2 = 100; % vlera maksimale  
  
% Numri i vlerave rasti që do të gjenerohen  
n = 1000;  
  
% Gjenero numra rasti në intervalin [a1, a2]  
x = a1 + (a2 - a1) * rand(n, 1);  
  
% Ndërto histogramin e numrave të gjeneruar  
histogram(x, 50); % përdor 50 shtylla në histogram për më shumë detaje  
title('Histogrami i Numrave të Rastit të Gjeneruar');  
xlabel('Vlerat e Numrave');  
ylabel('Frekuenca');
```

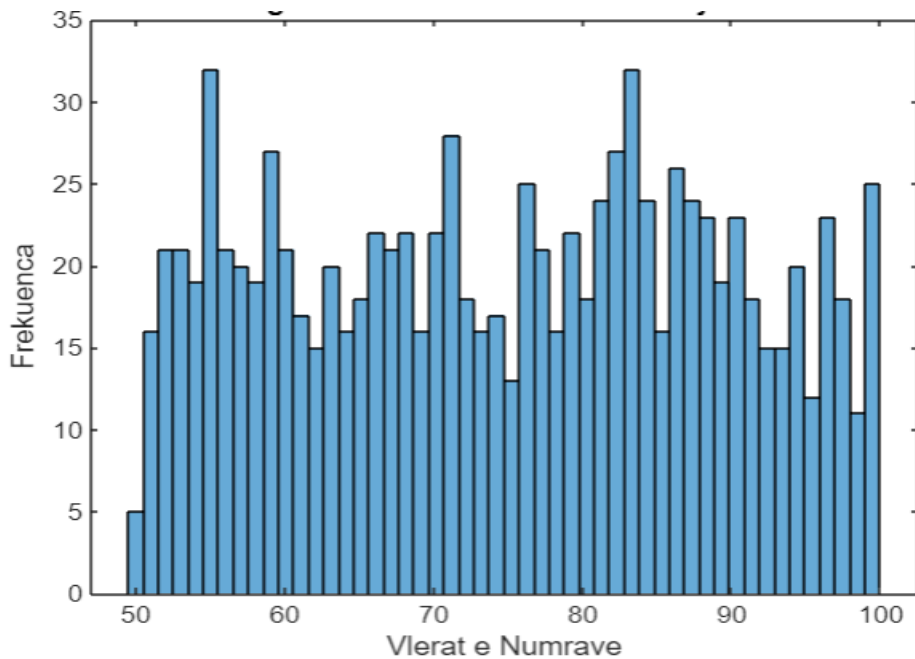


Fig 3.1 Histogrami I numrave te rastit te gjeneruar

Shembull për simulime dhe analiza statistikore ku është e nevojshme të gjenerohen të dhëna rasti në një interval të caktuar, dhe histogrami ndihmon në vlerësimin e uniformitetit të shpërndarjes së numrave të gjeneruar.

3.2 Shpërndarja Binomiale

- **Funksioni:** `binornd(n, p, M, N)`
- **Përdorimi:** Gjeneron numra nga një shpërndarje binomiale, ku `n` është numri i përpjekjeve, `p` është probabiliteti i suksesit për çdo përpjekje. `M` dhe `N` përcaktojnë përmasat e matricës së numrave të rastit që do të gjenerohen.

Ushtrimi 1:

Analizojmë probabilitetin që një person të zgjedhë të marrë pjesë në një sondazh. Supozojmë se probabiliteti që një person të marrë pjesë është 0.45. Do të përcaktojmë probabilitetin që nga një grup prej 15 personash të zgjedhur rastësisht:

- Të jenë saktësisht 7 persona që zgjedhin të marrin pjesë.
- Të jenë më pak se 5 persona që zgjedhin të marrin pjesë.
- Do të ndërtojmë grafikun e funksionit të densitetit të probabilitetit për këtë shpërndarje.

% X-numri i personave që zgjedhin të marrin pjesë. $X \sim B(15, 0.45)$

```

%a)  $P(X=7)=?$ 

%b)  $P(X<5)=?$ 

% Gjetja e probabilitetit për saktësisht 7 pjesëmarrës
prob_a = binopdf(7, 15, 0.45);
fprintf('P(X=7)=%f \n', prob_a);

% Gjetja e probabilitetit për më pak se 5 pjesëmarrës
prob_b = binocdf(4, 15, 0.45); % binocdf kalkulon probabilitetin
kumulativ deri në 4
fprintf('P(X<5)=%f \n', prob_b);

% Ndërtimi i vektorit me vlerat e ndryshores së rastit X
x = 0:15;

% Gjetja e densitetit të funksionit të shpërndarjes binomiale
y = binopdf(x, 15, 0.45);

% Ndërtimi i grafikut me shtylla të densiteteve të probabiliteteve
bar(x, y, 1, 'g') % Ngjyra e gjelbër për shtyllat
title('Histogrami i Shpërndarjes Binomiale për n=15, p=0.45')
xlabel('Numri i Pjesëmarrësve')
ylabel('Probabiliteti')

>> untitled
>> untitled
P(X=7)=0.201344
P(X<5)=0.120399
>>

```

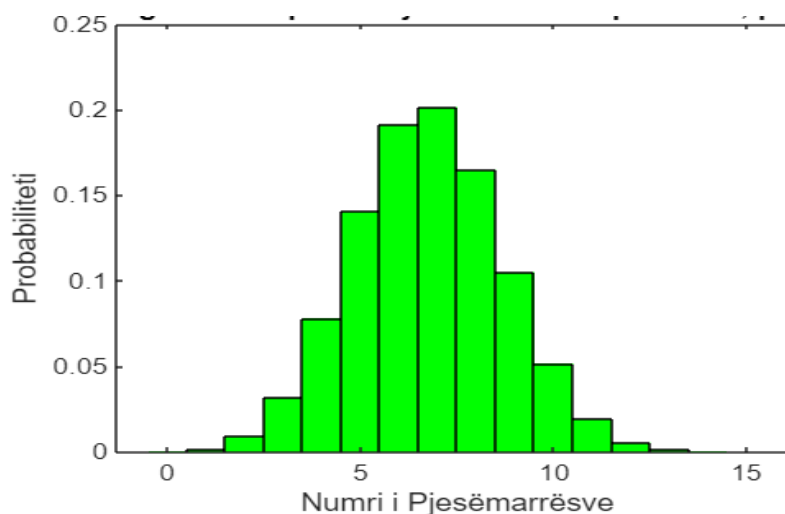


Fig 3.2 Histogrami i Shpërndarjes Binomiale për $n=15$, $p=0.45$

Ushtrimi 2

Ndërtoni një kod në Matlab i cili gjeneron disa numra të rastit nga shpërndarje Binomiale me parametër $n=100$, $p=0.5$. A mund të bëni një paraqitje grafike?

```
% Parametrat e shpërndarjes binomiale  
  
n = 100; % numri i eksperimenteve  
  
p = 0.5; % probabiliteti i suksesit në çdo eksperiment  
  
% Gjenerimi i numrave të rastit nga shpërndarja binomiale  
  
num_samples = 1000; % numri i mostrave  
  
binom_samples = binornd(n, p, [num_samples, 1]);  
  
% Shfaqja e histogramit të numrave të gjeneruar  
  
figure;  
  
histogram(binom_samples, 'Normalization', 'probability');  
  
xlabel('Numri i suksesit në 100 eksperimente');  
  
ylabel('Frekuenca relative');  
  
title('Shpërndarja e numrave të rastit nga shpërndarja binomiale');  
  
grid on;
```

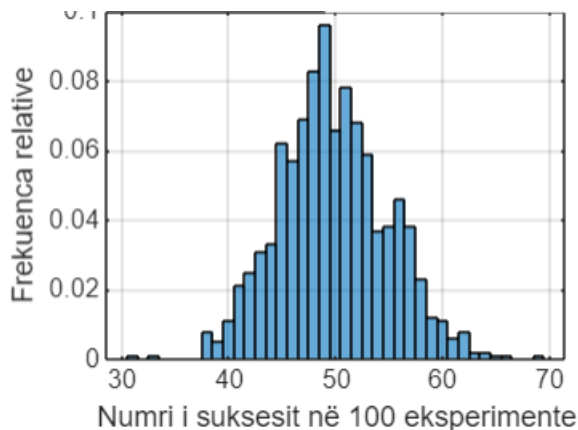


Fig 3.2 Shpërndarja e numrave të rastit nga shpërndarja binomiale

3.3 Shpërndarja Puasonit

- Funksioni: `poissrnd(lamda, M, N)`
- Përdorimi: Gjeneron numra nga një shpërndarje Poasoni, ku `lamda` është parametri i shkallës (intensiteti i ngjarjeve). `M` dhe `N` janë përsëri përmasat e matricës së numrave të rastit.

Ushtrimi 1:

Do të përdorim një ndryshore rasti me shpërndarje Puasoni dhe do të gjejmë probabilitetin që kjo ndryshore të marrë vlera më të mëdha se një pikë e caktuar, dhe më pas do të vizualizojmë shpërndarjen me një grafik.

Në këtë rast, do të përdorim një parametër lambda të ndryshëm, le të themi $\lambda = 5$, dhe do të gjejmë probabilitetin që ndryshorja rasti të marrë vlera më shumë se $x = 4$.

```
% Llogaritja e probabilitetit që ndryshorja rasti të marrë vlera më të  
mëdha se 4
```

```
prob_c = 1 - poisscdf(4, 5); % Përdorimi i funksionit poisscdf për të  
gjetur  $P(X \leq 4)$ 
```

```
fprintf('P(X>4)=1-P(X<=4)=%f \n', prob_c) % Komandë për të printuar  
dhe formatuar vlerën e probabilitetit
```

```
% Ndërtimi i grafikut të densitetit të probabiliteteve
```

```
x = 0:15; % Ndërtimi i një vektori
```

```
pdf_2 = poisspdf(x, 5); % Kthen vlerat e funksionit të densitetit  
Puasonian me  $\lambda=5$ 
```

```
bar(x, pdf_2, 1, 'r') % Ndërtimi i grafikut me shtylla të shpërndarjes  
Puasoniane me ngjyrë të kuqe
```

```
title('Grafiku i shpërndarjes së Puasonit me parametër  $\lambda=5$ ') %  
Vendosja e një titulli për grafikun
```

```
>> untitled  
>> untitled  
P(X=7)=0.201344  
P(X<5)=0.120399  
>> untitled  
P(X>4)=1-P(X<=4)=0.559507  
>>
```

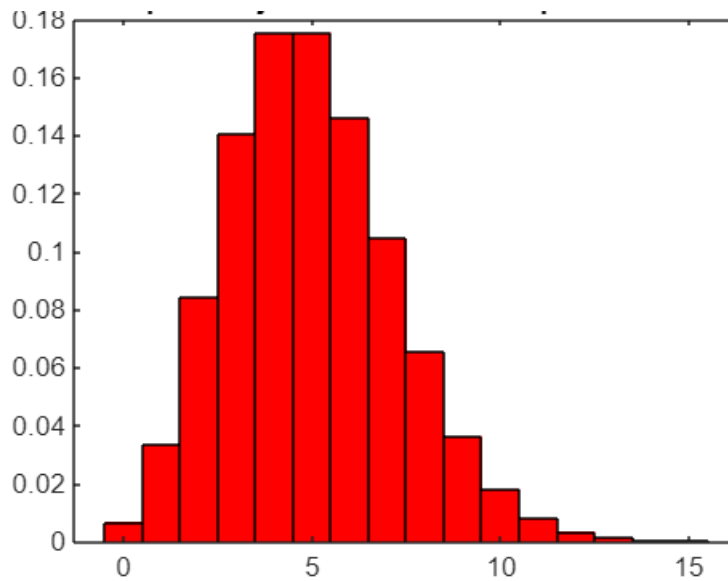


Fig3.3 Grafiku I shperndarjes se Puasonit me parameter lambda

3.4 Shperndarja Eksponenciale

- Funksioni: `exprnd(lamda)`
- Përdorimi: Gjeneron numra nga një shpërndarje eksponenciale, ku `lamda` është parametri i shkallës. Funksioni kthen një numër të vetëm të rastit për çdo thirrje, por mund të thirret brenda një cikli ose të përdoret me funksione si `arrayfun` për të gjeneruar disa vlera.

Ushtrimi 1:

Një bateri zgjat mesatarisht pesë vjet. Kohëzgjatja e baterisë ka shpërndarje eksponenciale. a) Cili është probabiliteti që një bateri zgjat më shumë se 3 vjet? b) Ndërtoni grafikun e densitetit të funksionit

```
% Shënoj me x = koha (në vite) që një bateri zgjat.
% X~E(lambda) ku lambda = 1/5, sepse mesatarisht zgjat 5 vjet
% a) P(X>3) = ?

lambda = 1/5; % Përcakto vlerën e lambdas për shpërndarje eksponenciale
prob_a = expcdf(3, 1/5); % Përdor komandën e funksionit të shpërndarjes
prob_a = 1 - prob_a; % Llogaritja e probabilitetit që X më shumë se 3 vjet

fprintf('P(X>3)=%f \n', prob_a) % Përdor funksionin 'fprintf' për të printuar dhe formatuar vlerën e probabilitetit

% b) Ndërtoni grafikun e densitetit të funksionit
```

```

x = 0:0.01:20; % Ndërtoj një vektor më të gjatë për të parë shtrirjen
e funksionit

pdf_1 = exppdf(x, 1/5); % Gjej densitetin e funksionit

plot(x, pdf_1) % Ndërtoj grafikun e densitetit të probabiliteteve

title('Shpërndarja Eksponenciale për Baterinë') % I vendos një titull
grafikut

xlabel('Vite'), ylabel('Densiteti i Probabilitetit') % Emërtoj boshtet
e grafikut

gtext('E(1/5)') % Përdor këtë komandë për të vendosur tekst brenda
grafikut

>> untitled
>> untitled
P(X=7)=0.201344
P(X<5)=0.120399
>> untitled
P(X>4)=1-P(X<=4)=0.559507
>> untitled
P(X>3)=0.000000

```

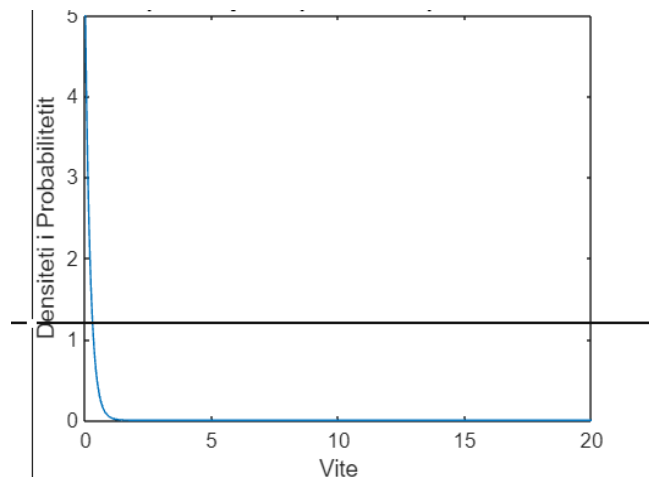


Fig 3.4 Shpërndarja Eksponenciale për Baterinë

3.5 Shpërndarja Gjeometrike:

Një lojtar hedh një zarrë të drejtë dhe ka një probabilitet prej $1/6$ që të hedhë një 6. Kërkesat do të jenë si më poshtë:

- Cili është probabiliteti që 6 të jetë hedhur për herë të parë në hedhjen e katërt?
- Cili është probabiliteti që 6 të jetë hedhur të paktën një herë në 4 hedhje?

% Përcakto parametrin e shpërndarjes gjeometrike

```
p = 1/6;
```

```

% a) Probabiliteti që 6 të jetë hedhur për herë të parë në hedhjen e
katërt

k = 4;

prob_a = (1 - p)^(k - 1) * p;

fprintf('\n P(X=4) = %f \n', prob_a);

% b) Probabiliteti që 6 të jetë hedhur të paktën një herë në 4 hedhje

n = 4;

prob_b = 1 - (1 - p)^n;

fprintf('\n P(X<=4) = %f \n', prob_b);

>> untitled

P(X=4) = 0.096451

P(X<=4) = 0.517747
>>

```

3.6 Shperndarja Uniforme

Ushtrimi 1

Midis numrave 1,2.....200 zgjidhe rastësisht një numër .

a)Gjeni probabilitetin që zgjidhet numri 50.

b)Gjeni probabilitetin që numri të jetë më i vogël se 60

```

% Definimi i parametrave

```

```

N = 200; % Numri total i numrave

```

```

X_a = 50; % Numri për të cilin do të llogaritim probabilitetin që
zgjidhet

```

```

X_b = 60; % Numri për të cilin do të llogaritim probabilitetin që
zgjidhet më i vogël se ky numër

```

```

% a) Gjeni probabilitetin që zgjidhet numri 50

```

```

prob_a = unidpdf(X_a, N); % Gjej densitetin e funksionit

```

```
fprintf('\n P(X=50)=%f \n', prob_a); % Përdor funksionin 'fprintf' për
të printuar dhe formatuar vlerën e probabilitetit
```

% b) Gjeni probabilitetin që numri të jetë më i vogël se 60

```
prob_b = unidcdf(X_b, N); % Gjej funksionin e shpërndarjes
```

```
fprintf('\n P(X<=60)=%f \n', prob_b); % Përdor funksionin 'fprintf'
për të printuar dhe formatuar vlerën e probabilitetit
```

```
>> untitled
```

```
P(X=50)=0.005000
```

```
P(X<=60)=0.300000
```

```
>>
```

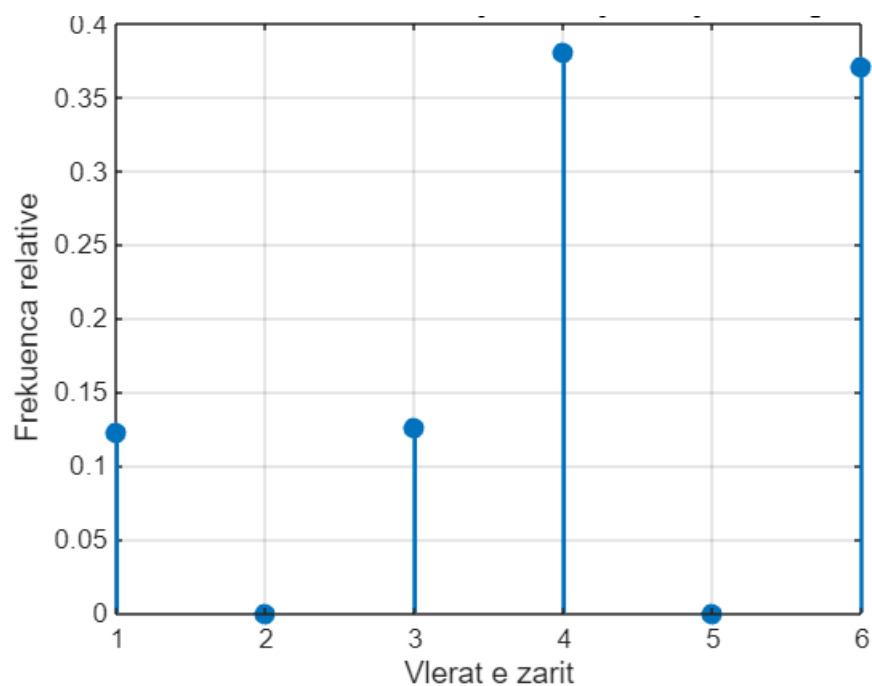


Fig 3.4 Frekuenca relative e hedhjeve të zarit të rregullt

Ushtrimi 2

Njihsoni funksionin densitet të probabiliteteve për $X = U_1 U_2$, ku U_1 dhe U_2 janë ndryshoret nga shpërndarja uniforme. Në cilën zonë shtrihen me se shumti të dhënat

```
rand('state',0)
```

```
x=zeros(10000,1);
```

```

for i=1:10000

x(i,1)=rand(1,1)*rand(1,1);

end

[yy,xx]=hist(x(1:10000),50);

amplitudax=xx(2)-xx(1);

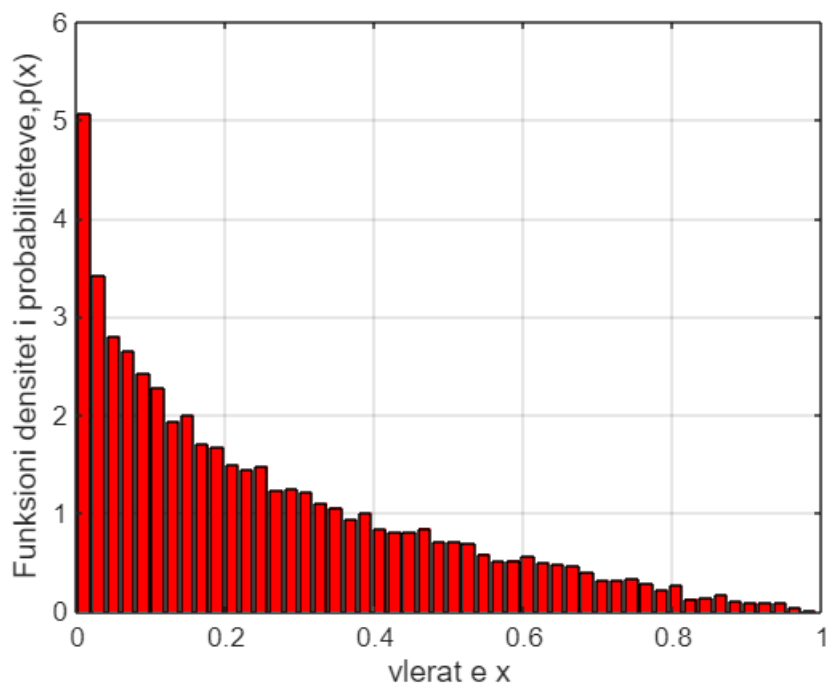
bar(xx,yy/(10000*amplitudax),'r')

grid

axis([0 1 0 6]); xlabel('vlerat e x')

ylabel('Funksioni densitet i probabiliteteve,p(x)')

```



3.7 Gjenerimi i shpërndarjeve diskrete duke përdorur Shpërndarjen Uniforme

Ushtrim 1

Shkruani një kod për të simuluar hedhjen e një monedhe të rregullt dhe të shikojmë nëse ligji i numrave të mëdhenj funksionon për një vëllim të caktuar 10.

```

% Gjenerojmë numra të rastësishëm nga shpërndarja uniforme

U = rand(1, 10); % gjenerojmë numra rasti matricë 1x10

koke = (U < 0.5); % nëse U < 0.5 atëherë = 1; përndryshe = 0

% Inicializimi i vektorëve për numrin e herëve kokë dhe mesataren

a = zeros(1, 11); % vektor për numrin e herëve kokë

mes = zeros(1, 10); % vektor për mesataren

% Llogaritja e numrit të herëve kokë dhe mesatares

for i = 2:11 % nisim nga 2 deri në 11 në rastin tonë

    a(i) = a(i-1) + koke(i-1); % llogarit numrin e herëve kokë

    mes(i-1) = a(i) / (i-1); % llogarit mesataren, shiko ligjin e
numrave të mëdhenj

end

% Shfaqja e grafikut të mesatares

figure;

plot(mes, 'b-o', 'LineWidth', 2);

xlabel('Numri i hedhjeve');

ylabel('Mesatarja e herëve kokë');

title('Ligji i Numrave të Mëdhenj për Hedhjen e Monedhës së Rregullt');

grid on;

```

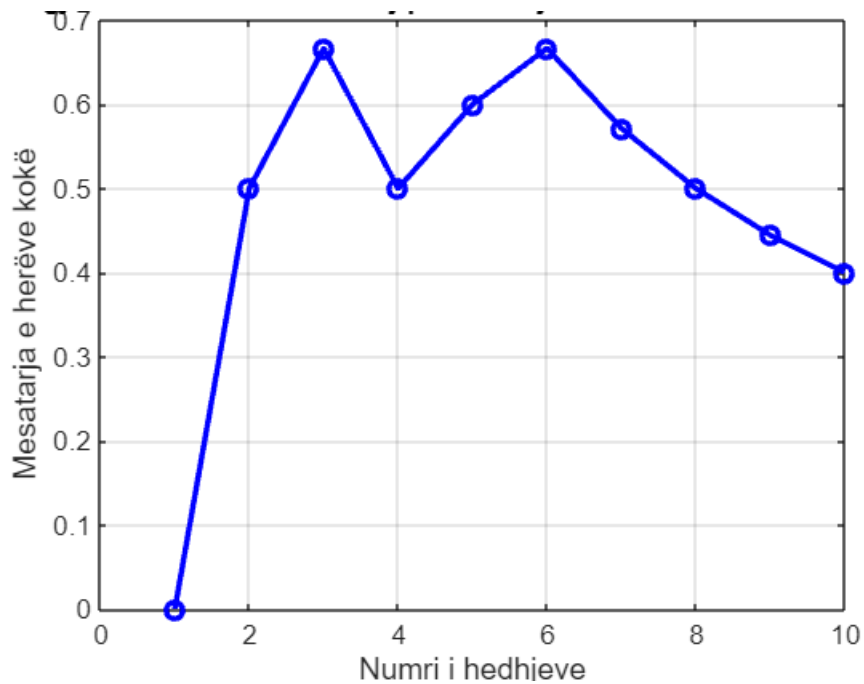


Fig 3.5 Ligji I numrave te medhenj per hedhjen e monedhes se rregullt

3.8 Metoda Box-Muller

Teoreme: (Transformimi Box-Muller)

Ndryshoret e rastit (Z_1, Z_2) te pavarura me shperndarje normale nese zbatojme transformimin

$$Z_1 = \sqrt{-2\ln U_1} \cos(2\pi U_2) \quad Z_2 = \sqrt{-2\ln U_1} \sin(2\pi U_2)$$

ku (U_1, U_2) ndryshore te pavarura me shperndarje Uniforme $(0,1)$.

Ushtrimi 1

Gjenerojmë një ndryshore rasti me funksion densitet $p(x)=3x^2$ për $0 < x < 1$ duke përdorur një gjenerator të numrave të rastit uniforme standarte U .

```
% Numri i ndryshoreve të cilat dëshirojmë t'i gjenerojmë
```

```
N = 1000;
```

```
% Gjenerimi i ndryshoreve të rastit uniforme standarte U
```

```
U = rand(N, 1);
```

```
% Gjetja e ndryshoreve rasti X me funksion densitet  $p(x) = 3x^2$  duke përdorur ndryshore rasti uniforme standarte U
```

```

X = U.^(1/3);

% Vizualizimi i histogramës së ndryshores rasti X për të vërtetuar që
ka p(x) si funksion densitet

histogram(X, 'Normalization', 'pdf'); % Normalizimi sipas densitetit
për të krahasuar me p(x)

hold on;

% Plotimi i grafikut të funksionit densitet p(x)

x_values = linspace(0, 1, 100); % Vlerat e x për grafikun

p_values = 3 * x_values.^2; % Vlerat e p(x) për këto vlera të x

plot(x_values, p_values, 'r', 'LineWidth', 2); % Plotimi i grafikut të
p(x)

xlabel('x'); % Etiketa e boshtit x

ylabel('p(x)'); % Etiketa e boshtit p(x)

title('Histograma e ndryshores rastit X dhe funksioni densitet p(x)');
% Titulli i grafikut

hold off;

```

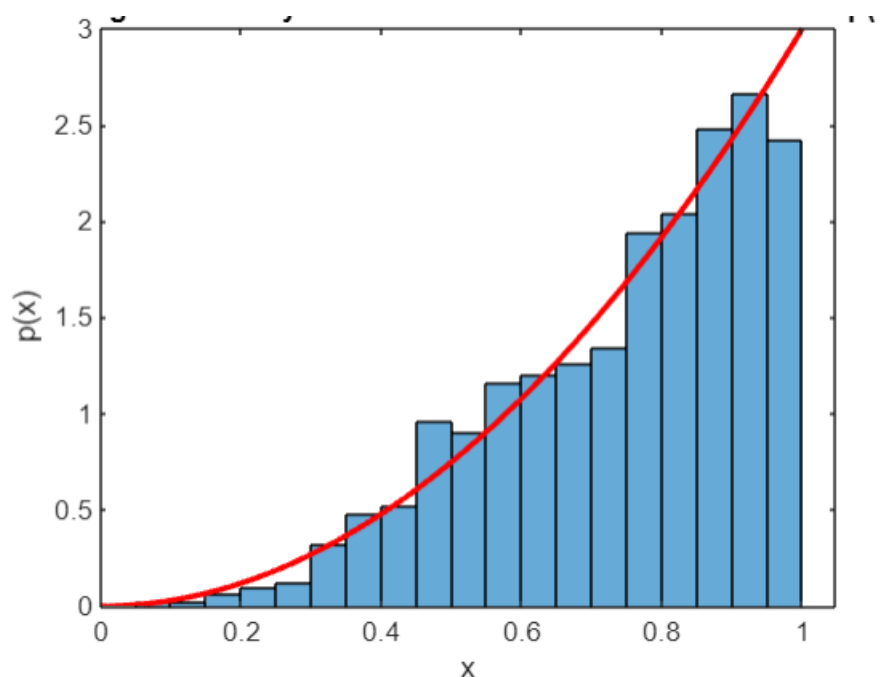


Fig 3.6 Histograma e ndryshores rastit x dhe funksioni densitet $p(x)$

4 MATEMATIKA NE MATLAB

4.1 Funksionet ne Matematike

Le të krijojmë një funksion të ri Matlab që llogarit një shprehje të ndryshme dhe ta testojmë për vlera të ndryshme të x dhe y . Këtu është një shembull me shprehjen

$$z=2x^3+x^2+y^2+e^{\log(y)}$$

Modeli i funksionit :

```
function z = llogarit_v2(x, y)
    z = 2 * x^3 + sqrt(x^2 + y^2) + exp(log(y));
end
```

Testimi I funksionit:

```
% Testimi i funksionit me vlera të ndryshme të x dhe y
x1 = 3;
y1 = 4;
z1 = llogarit_v2(x1, y1);
fprintf('Për x = %d dhe y = %d, z = %.4f\n', x1, y1, z1);

x2 = 5;
y2 = 6;
z2 = llogarit_v2(x2, y2);
fprintf('Për x = %d dhe y = %d, z = %.4f\n', x2, y2, z2);

x3 = 7;
y3 = 8;
z3 = llogarit_v2(x3, y3);
fprintf('Për x = %d dhe y = %d, z = %.4f\n', x3, y3, z3);
```

4.2 Statistika

Ushtrimi 1

Ndërtojmë një vektor me numra të rastit nga -50 në 50 dhe do të gjejmë vlerat e mesatares, mesores, devijimit standard, vlerat (min, max), dhe variancën empirik.

```

% Ndërtimi një vektor me numra të rastit nga -50 në 50

x = rand(100, 1) * 100 - 50;

disp('Vektori me numra të rastit nga -50 në 50:');

disp(x);

% Gjeni mesataren e vektorit

mean_value = mean(x);

fprintf('Mesatarja e vektorit: %.4f\n', mean_value);

% Gjeni mesoren e vektorit

median_value = median(x);

fprintf('Mesorja e vektorit: %.4f\n', median_value);

% Gjeni devijimin standard të vektorit

std_value = std(x);

fprintf('Devijimi standard i vektorit: %.4f\n', std_value);

% Gjeni minimumin e vektorit

min_value = min(x);

fprintf('Minimumi i vektorit: %.4f\n', min_value);

% Gjeni maksimumin e vektorit

max_value = max(x);

fprintf('Maksimumi i vektorit: %.4f\n', max_value);

% Gjeni variancën e vektorit

var_value = var(x);

fprintf('Varianca e vektorit: %.4f\n', var_value);

Mesatarja e vektorit: 1.6930
Mesorja e vektorit: 2.0938
Devijimi standard i vektorit: 28.7594
Minimumi i vektorit: -49.7910
Maksimumi i vektorit: 47.0660
Varianca e vektorit: 827.1034
>>

```

Ushtrimi 2

```

% Ndërtimi një vektor me 100 elementë nga shpërndarja normale me
parametra zero dhe një

```

```

y = randn(1, 100);

% Gjeni vlerën minimale
a = min(y);

fprintf('Vlera minimale e vektorit = %f\n', a);

% Gjeni vlerën maksimale
b = max(y);

fprintf('Vlera maksimale e vektorit = %f\n', b);

% Gjeni mesataren
c = mean(y);

fprintf('Mesatarja e vektorit = %f\n', c);

% Gjeni mesoren
d = median(y);

fprintf('Mesorja e vektorit = %f\n', d);

% Gjeni shmangien mesatare katrore
e = std(y);

fprintf('Shmangia mesatare katrore e vektorit = %f\n', e);

% Ndërtoni një histogram
hist(y, 10);

title('Histogrami i vektorit y');

xlabel('Vlerat');

ylabel('Frekuenca');

>> untitled
Vlera minimale e vektorit = -2.944284
Vlera maksimale e vektorit = 3.578397
Mesatarja e vektorit = 0.123085
Mesorja e vektorit = 0.095403
Shmangia mesatare katrore e vektorit = 1.162399
>>

```

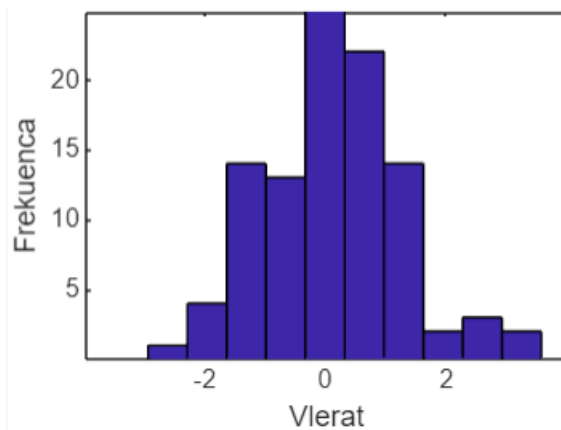


Fig 4.1 Histogrami i vektorit y

Ushtrimi 3:

Gjeneroni shpërndarjen e hedhjes së monedhës së rregullt, duke përsëritur eksperimentin 10000, ku në çdo prove monedha hidhet 4 herë. Te gjendet probabiliteti $P(X=2)$ ku X tregon numrin e rënieve kokë.

```
% Përcakto numrin e eksperimenteve dhe hedhjeve
num_experiments = 10000;
num_throws = 4;
% Rezervo memorjen për rezultatet e eksperimenteve
results = zeros(num_experiments, 1);
% Përsërit eksperimentin 10000 herë
for i = 1:num_experiment
    % Gjenero hedhjet e monedhës (1 - kokë, 0 - bisht)
    throws = rand(1, num_throws) > 0.5;
    % Numëro numrin e koka në 4 hedhje
    num_heads = sum(throws);
    % Ruaj rezultatin në listën e rezultateve
    results(i) = num_heads;
end
% Gjej probabilitetin P(X=2)
prob_X_equals_2 = sum(results == 2) / num_experiments;
```

```
% Shfaq rezultati

fprintf('Probabiliteti P(X=2) është: %f\n', prob_X_equals_2);

5 zgjedhje me përmasë 4:
    0.9528    0.7041    0.9539    0.5982
    0.8752    0.3179    0.2732    0.6765
    0.5162    0.2252    0.1837    0.2163
    0.4360    0.8212    0.6647    0.6455
    0.8639    0.9062    0.2001    0.8594

Ndryshoret e gjeneruara në zgjedhjen e dytë:
    0.8752    0.3179    0.2732    0.6765

>> untitled
Probabiliteti P(X=2) është: 0.372200
>>
```

4.3 Funksionet Trigonometrike

Ushtrimi 1

Konvertimin midis Fahrenheit dhe Celsius dhe anasjelltas. Këtu janë dy funksionet Matlab që kryejnë këto konvertime dhe një shembull për testimin e tyre.

Funksioni për konvertimin nga Fahrenheit në Celsius:

```
function C = fahrenheitToCelsius(F)

    C = (F - 32) * 5 / 9;

End
```

Funksioni për konvertimin nga Celsius në Fahrenheit:

```
function F = celsiusToFahrenheit(C)

    F = (C * 9 / 5) + 32;

End
```

Testimi i funksionaliteteve:

```
% Testimi i funksionit fahrenheitToCelsius

F1 = 98.6;

C1 = fahrenheitToCelsius(F1);

fprintf('Temperatura %.2f°F në Celsius është %.2f°C\n', F1, C1);

% Testimi i funksionit celsiusToFahrenheit
```

```

C2 = 37;

F2 = celsiusToFahrenheit(C2);

fprintf('Temperatura %.2f°C në Fahrenheit është %.2f°F\n', C2, F2);

% Testimi i saktësisë së funksioneve

% Konvertimi dyfishtë

F3 = 212;

C3 = fahrenheitToCelsius(F3);

F3_converted = celsiusToFahrenheit(C3);

fprintf('Konvertimi dyfishtë: %.2f°F -> %.2f°C -> %.2f°F\n', F3, C3,
F3_converted);

C4 = 100;

F4 = celsiusToFahrenheit(C4);

C4_converted = fahrenheitToCelsius(F4);

fprintf('Konvertimi dyfishtë: %.2f°C -> %.2f°F -> %.2f°C\n', C4, F4,
C4_converted);

```

Ushtrimi 2

Marrim një trekëndësh kënd-drejtë dhe do të llogarisim këndin α duke përdorur funksionet `asin`, `acos` dhe `atan`. Funksioni i ndërtuar do të kthejë rezultatin në gradë.

Funksioni ne Matlab:

```

function kendiA = kend_drejt(x, y, type)

    % Funksioni që llogarit këndin A bazuar në vlerat x dhe y dhe tipin
    e funksionit trigonometrik

    switch type
        case 'sin'
            kendiA = asin(x / y); % Llogarit këndin A duke përdorur
            arcsin
        case 'cos'
            kendiA = acos(x / y); % Llogarit këndin A duke përdorur
            arccos
        case 'tan'

```

```

        kendiA = atan(x / y); % Llogarit këndin A duke përdorur
arctan

        otherwise

            error('Tip i panjohur. Përdorni ''sin'', ''cos'' ose
''tan''.'');

        end

        % Konvertoni nga radian në gradë

        kendiA = rad2deg(kendiA);

    end
end

```

Testimi I Funksionit:

```

% Definimi i vlerave për trekëndëshin kënd-drejtë

a = 6;

b = 10;

c = sqrt(a^2 + b^2); % Teorema e Pitagorës

% Testimi i funksionit kend_drejt me vlera të ndryshme

kendiA_sin = kend_drejt(a, c, 'sin');

fprintf('Këndi A duke përdorur sin është: %.2f gradë\n', kendiA_sin);

kendiA_cos = kend_drejt(b, c, 'cos');

fprintf('Këndi A duke përdorur cos është: %.2f gradë\n', kendiA_cos);

kendiA_tan = kend_drejt(a, b, 'tan');

fprintf('Këndi A duke përdorur tan është: %.2f gradë\n', kendiA_tan);

>> untitled
Këndi A duke përdorur sin është: 30.96 gradë
Këndi A duke përdorur cos është: 30.96 gradë
Këndi A duke përdorur tan është: 30.96 gradë
>>

```

4.4 Funksionet Trigonometrik (Grafiku)

Ushtrimi 1

Le të krijojmë një grafik për funksionet $\tan(\theta)$ dhe $\cot(\theta)$ për $0 \leq \theta \leq \pi$ në të njëjtin plan. Grafiku do të përmbajë emërtimet e boshteve, emërtimet e vijave, dhe vijat e grafikëve do të jenë të ndryshme në stil dhe ngjyrë.

```

% Definimi i intervalit për  $\theta$ 

theta = 0:0.01:pi;

```

```

% Llogaritja e funksioneve tan(θ) dhe cot(θ)

tan_theta = tan(theta);
cot_theta = 1 ./ tan(theta);

% Ndërto grafikun e tan(θ)

figure;

plot(theta, tan_theta, 'b--', 'LineWidth', 1.5);

hold on;

% Ndërto grafikun e cot(θ)

plot(theta, cot_theta, 'm-.', 'LineWidth', 1.5);

% Emërtimi i boshteve

xlabel('θ (radian)', 'FontSize', 12);

ylabel('f(θ)', 'FontSize', 12);

% Titulli i grafikut

title('Grafiku i tan(θ) dhe cot(θ)', 'FontSize', 14);

% Legjenda

legend('tan(θ)', 'cot(θ)', 'Location', 'best');

% Vendosje e gridit

grid on;

% Kufizimi i boshteve për një pamje më të mirë

xlim([0 pi]);

ylim([-10 10]);

% Mbajtja e grafikut në të njëjtin plan

hold off;

```

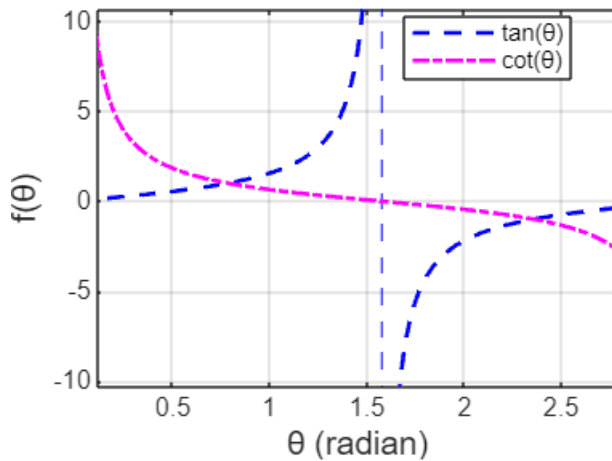


Fig 4.1 Grafiku i $\tan(\theta)$ dhe $\cot(\theta)$

Ushtrimi 2

Do të përdorim funksionet $y = \sin(3x)$ dhe $z = \cos(3x)$ dhe do të ndërtojmë grafikun e tyre në të njëjtin plan.

```
% Definimi i intervalit për x
x = 0:0.01:2*pi;

% Definimi i funksioneve y = sin(3x) dhe z = cos(3x)
y = sin(3 * x);
z = cos(3 * x);

% Ndërtimi i grafikut të funksioneve në të njëjtin plan
figure;

plot(x, y, 'b--o', 'LineWidth', 1.5); % Grafiku i y, me vijë blu të
ndërprerë dhe pika rrathë

hold on;

plot(x, z, 'r:+', 'LineWidth', 1.5); % Grafiku i z, me vijë të kuqe me
pika dhe pika plus

hold off;

% Emërtimi i boshtet dhe grafikut
xlabel('x', 'FontSize', 12);
ylabel('Vlerat e funksionit', 'FontSize', 12);

title('Grafiku i funksioneve y = sin(3x) dhe z = cos(3x)', 'FontSize',
14);

% Vendosja e legjendës
```

```

legend('y = sin(3x)', 'z = cos(3x)', 'Location', 'best');

% Vendosje e gridit

grid on;

```

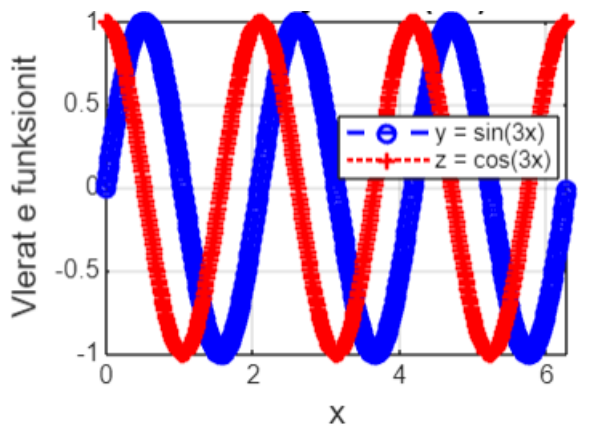


Fig 4.2 grafiku i funksioneve $y = \sin(3x)$ dhe $z = \cos(3x)$

Ushtrimi 3

Ndërtimi i grafikut të funksioneve në koordinata polare

a) $r_1 = 2 + \cos(\theta)$ dhe b) $r_2 = 3 - \sin(\theta)$

```

% Definimi i vektorit theta

theta = linspace(0, 3*pi, 1000); % përdorim komandën "linspace" për
ndërtimin e vektorit theta nga 0 deri në 3*pi me 1000 pika

% Ndërtimi i funksioneve r1 dhe r2

r1 = 2 + cos(theta); % ndërtojmë funksionin r1

r2 = 3 - sin(theta); % ndërtojmë funksionin r2

% Krijimi i grafikut në koordinata polare

figure;

% Ndani zonën në 2 pjesë dhe pozicionohemi në pjesën e parë

subplot(1, 2, 1);

polarplot(theta, r1, 'b.-'); % përdorim komandën "polarplot" për të
ndërtuar grafikun në koordinata polare

title('r1 = 2 + cos(\theta)');

% Ndani zonën në 2 pjesë dhe pozicionohemi në pjesën e dytë

subplot(1, 2, 2);

polarplot(theta, r2, 'r.-'); % përdorim komandën "polarplot" për të
ndërtuar grafikun në koordinata polare

title('r2 = 3 - sin(\theta)');

% Vendosja e tekstit mbi grafik

```

```
sgtitle('Grafikët e Funksioneve në Koordinata Polare'); % komanda
'sgtitle' vendos tekst mbi grafik
```

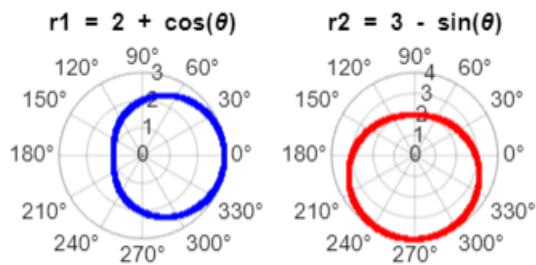


Fig 4.4 grafikete e funksionit me kordinata polare

Ushtrimi 4

Le të krijojmë një shembull duke ndërtuar një vijë spirale cilindrike. Në këtë rast, do të përdorim funksionet vektoriale të mëposhtme:

$$x = \cos(t) \quad y = \sin(t) \quad z = t$$

```
% Definimi i vektorit t
```

```
t = linspace(0, 6*pi, 1000); % përdorim komandën "linspace" për
ndërtimin e vektorit
```

```
% Ndërtimi i funksioneve x, y dhe z
```

```
x = cos(t); % ndërtojmë funksionin x
```

```
y = sin(t); % ndërtojmë funksionin y
```

```
z = t; % ndërtojmë funksionin z
```

```
% Krijimi i grafikut në 3D
```

```
figure;
```

```
hold on; % përdor komandën 'hold on' për të mbajtur të fiksuar
grafikun
```

```
plot3(x, y, z); % komanda 'plot3' për ndërtimin në 3D
```

```
box on; % shton një kuti në boshtet e grafikut
```

```
title('Vija Spirale Cilindrike'); % i vendosim një titull grafikut
```

```
xlabel('x');
```

```
ylabel('y');
```

```
zlabel('z');
```

```
grid on;
```

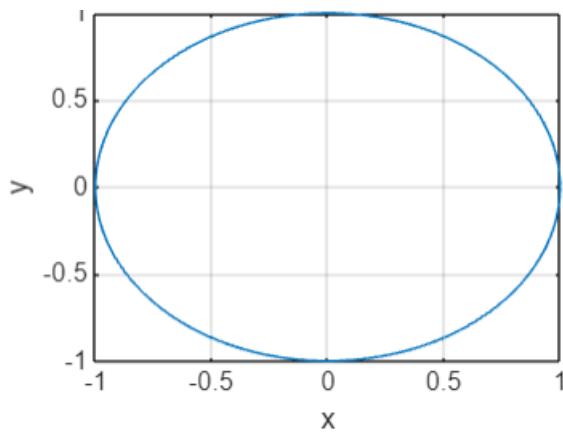


Fig 4.4 vlera spirale cilindrike

4.5 Rregalla Kosinusit

Llogarisim anën c të një trekëndëshi për vlera të ndryshme të a , b , dhe C

```
function c = kosinus(a, b, C)
```

```
% Funkcioni që llogarit anën c të trekëndëshit duke përdorur  
Rregullën e Kosinusit
```

```
c = sqrt(a^2 + b^2 - 2*a*b*cos(C));
```

```
end
```

Testimi I funksionit:

```
% Definimi i vlerave për trekëndëshin
```

```
a = 5;
```

```
b = 7;
```

```
C = pi/4; % Këndi C në radian (45 gradë)
```

```
% Testimi i funksionit kosinus me vlerat e mësipërme
```

```
c_value = kosinus(a, b, C);
```

```
fprintf('Vlera e c-së për a = %.2f, b = %.2f dhe C = %.2f radian është:  
%.4f\n', a, b, C, c_value);
```

```
% Testimi me vlera të tjera
```

```
a = 10;
```

```
b = 15;
```

```
C = pi/3; % Këndi C në radian (60 gradë)
```

```
c_value = kosinus(a, b, C);
```

```
fprintf('Vlera e c-së për a = %.2f, b = %.2f dhe C = %.2f radian është:
%.4f\n', a, b, C, c_value);
```

4.6 Rrenjet Komplekse

Ne ekuacionin $ax^2 + bx + c = 0$ llogarisim rrënjët e ekuacionit, shumën dhe ndryshesën e rrënjëve.

```
function x = zgjidhja(a, b, c)
```

```
    if a ~= 0
        x = zeros(2, 1);
        discriminant = b^2 - 4 * a * c;
        if discriminant >= 0
            x(1) = (-b + sqrt(discriminant)) / (2 * a);
            x(2) = (-b - sqrt(discriminant)) / (2 * a);
        else
            x(1) = (-b + sqrt(discriminant + 0i)) / (2 * a);
            x(2) = (-b - sqrt(discriminant + 0i)) / (2 * a);
        end
    elseif b ~= 0
        x = -c / b;
    else
        if c ~= 0
            disp('Nuk ka zgjidhje')
        else
            disp('Çdo numër kompleks është zgjidhje')
        end
    end
end
```

Testimi I funksionit:

```
% Definimi i koeficientëve për ekuacionin kuadrat
a = 2;
b = 6;
c = 10;
```

```

% Gjetja e rrënjëve të ekuacionit kuadrat
x = zgjidhja(a, b, c);

disp('Rrënjët e ekuacionit kuadrat janë:');

disp(x);

% Gjetja e shumës dhe ndryshesës së rrënjëve
sum_roots = sum(x);

diff_roots = diff(x);

fprintf('Shuma e rrënjëve është: %.2f\n', sum_roots);

fprintf('Ndryshesa e rrënjëve është: %.2f + %.2fi\n',
real(diff_roots), imag(diff_roots));

>> untitled
Rrënjët e ekuacionit kuadrat janë:
-1.5000 + 1.6583i
-1.5000 - 1.6583i

Shuma e rrënjëve është: -3.00
Ndryshesa e rrënjëve është: 0.00 + -3.32i
>>

```

4.7 Polinomet

Ushtrimi 1

Kemi polinomin $p(x)=3.5x^4-4.2x^3+6.3x^2-2x+8$ do të gjejmë rrënjët e tij dhe do të llogarisim vlerën e polinomit për një vlerë të caktuar të x .

Përcaktimi i polinomit dhe gjetja e rrënjëve:

```

% Definimi i polinomit p(x) = 3.5x^4 - 4.2x^3 + 6.3x^2 - 2x + 8
p = [3.5 -4.2 6.3 -2 8];

% Gjetja e rrënjëve të polinomit
roots_p = roots(p);

disp('Rrënjët e polinomit p(x):');

disp(roots_p);

```

Njehsimi i vlerës së polinomit për $x=3$:

```

% Njehsimi i vlerës së polinomit për x = 3

```

```

x_value = 3;

poly_value = polyval(p, x_value);

fprintf('Vlera e polinomit p(x) për x = %.2f është: %.4f\n', x_value,
poly_value);

```

Ushtrimi 2

Jane dhene dy polinome: $p_1(x)=1+x+x^2$, $p_2(x)=2+x^3$ Gjeni polinomin $p(x)=p_1(x)*p_2(x)$ duke perdorur MATLAB dhe gjeni rrenjet. Njehsoni vleren e polinomit $p(x)$ per $x=2$. Gjeni derivation e $p_2(x)$ dhe $p(x)$.

```

% Definimi i polinomeve p1(x) dhe p2(x)

p1 = [1 -2 1]; % Polinomi p1(x) = x^2 - 2x + 1
p2 = [1 0 0 3]; % Polinomi p2(x) = x^3 + 3

% Shumëzimi i polinomeve p1(x) dhe p2(x)

p = conv(p1, p2);

disp('Polinomi i prodhuar:');

disp(p);

% Gjetja e rrënjëve të polinomit p(x)

rrenjet = roots(p);

disp('Rrënjët e polinomit të prodhuar:');

disp(rrenjet);

% Njehsimi i vlerës së polinomit p(x) për x=2

z = polyval(p, 2);

disp('Vlera e polinomit të prodhuar për x=2:');

disp(z);

% Gjetja e derivatit të polinomit p2(x)

der_p2 = polyder(p2);

disp('Derivati i polinomit të dytë:');

disp(der_p2);

% Gjetja e derivatit të polinomit p(x)

der_p = polyder(p);

disp('Derivati i polinomit të prodhuar:');

```

```
disp(der_p);
```

5 TEOREMA QENDRORE LIMITE

Ushtrimi 1

Modifiko kodin duke marrë në konsideratë hedhjen zarit për të gjetur shpërndarjen e shumës për zaret e rregullt [3, 7, 10, 40, 60, 100]. Tregoni rezultatin edhe me anë të figurës.

```
% Definimi i numrit të zarave për secilin rast
num_zare = [3, 7, 10, 40, 60, 100];

% Numri i hedhjeve për secilin eksperiment
num_throws = 10000;

% Rezervimi i memorjes për rezultatet
results = cell(length(num_zare), 1);

% Kryerja e eksperimenteve
for i = 1:length(num_zare)
    n = num_zare(i);
    sums = zeros(num_throws, 1);
    for j = 1:num_throws
        % Gjenerimi i hedhjeve për n zare dhe llogaritja e shumës
        throws = randi(6, [n, 1]);
        sums(j) = sum(throws);
    end
    results{i} = sums;
end

% Shfaqja e rezultateve me anë të figurës
figure;
hold on;
colors = ['b', 'r', 'g', 'c', 'm', 'k'];
for i = 1:length(num_zare)
    histogram(results{i}, 'Normalization', 'probability',
'DisplayStyle', 'stairs', 'EdgeColor', colors(i));
end
```

```
hold off;

% Emërtimi i boshteve dhe titulli
xlabel('Shuma e hedhjeve të zarave');
ylabel('Frekuenca relative');

title('Shpërndarja e shumës për zarët e rregullt');

legend(arrayfun(@(x) sprintf('%d zare', x), num_zare, 'UniformOutput',
false));

grid on;
```

Ushtrimi 2

Marrim një zar jo të rregullt, ku $P(1)=P(3)=q$; $P(2)=P(5)=0$ dhe $P(4)=P(6)=3q$. Simuloni 2000 hedhjet e këtij zari dhe ndërtoni një grafik të frekuencës relative duke përdorur komandën stem

Probabilitetet e dhëna:

- $P(1)=P(3)=q$
- $P(2)=P(5)=0$
- $P(4)=P(6)=3q$

Kushti i normalizimit të probabiliteteve:

$$P(1)+P(2)+P(3)+P(4)+P(5)+P(6)=1$$

$$q+0+q+3q+0+3q=1$$

$$8q=1$$

$$q = 1/8$$

Probabilitetet e plota:

- $P(1)=1/8$
- $P(2)=0$
- $P(3)=1/8$
- $P(4)=3/8$
- $P(5)=0$
- $P(6)=3/8$

```
% Probabilitetet e vlerave të zarit
probs = [1/8, 0, 1/8, 3/8, 0, 3/8];
```

```
% Numri i hedhjeve
num_throws = 2000;
```

```
% Gjenerimi i hedhjeve të zarit
throws = randsample(1:6, num_throws, true, probs);
```

```
% Llogaritja e frekuencës relative
values = 1:6;
freq_relative = histcounts(throws, 'Normalization', 'probability',
'BinLimits', [0.5 6.5]);

% Shfaqja e grafikut të frekuencës relative duke përdorur stem
figure;
stem(values, freq_relative, 'filled', 'LineWidth', 1.5);
xlabel('Vlerat e zarit');
ylabel('Frekuenca relative');
title('Frekuenca relative e hedhjeve të një zari jo të rregullt');
grid on;
```

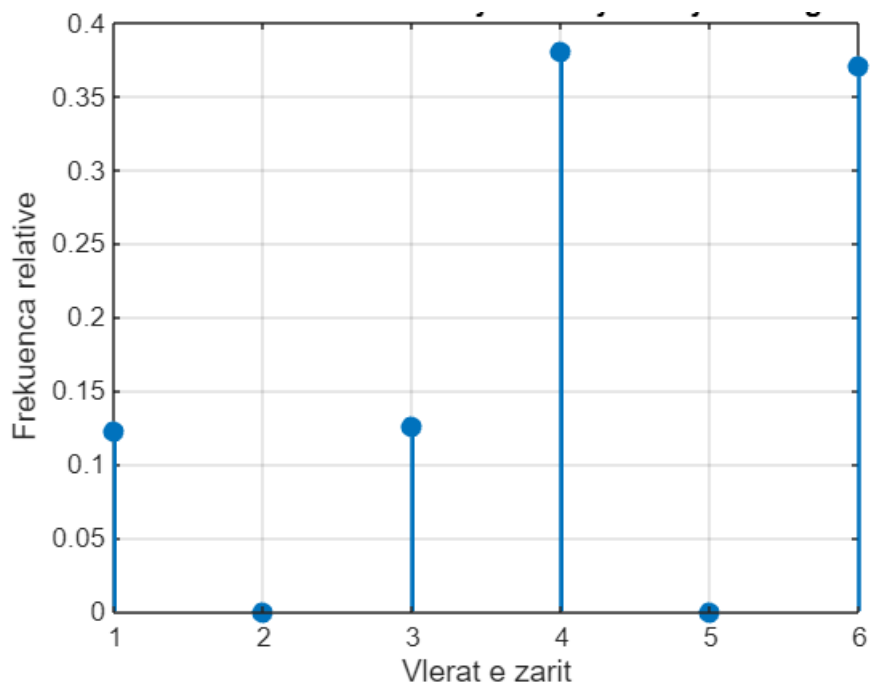


Fig 5.1 Frekuenca relative e hedhjeve të një zari jo të rregullt

6 CIKLET DHE KUSHTET E KONTROLLIT

6.1 Cikli For

Ushtrimi 1

```
% Vektori me vlera të ndryshme
y = [10, 20, 30, 40, 50];

% Inicializimi i shumës
shuma = 0;

% Numri i elementeve në vektor
N = length(y);
```

```

% Përshkruajmë çdo element të vektorit dhe llogaritim shumën e tyre
for k = 1:N
    shuma = shuma + y(k);
end
% Llogaritja e mesatares
mes = shuma / N;
% Shfaqja e rezultatit
disp(mes);

```

6.2 Cikli While

Do të zgjedhim një ndryshore rasti Y me vlerat e mëposhtme:

$$P(Y=1)=0.2$$

$$P(Y=2)=0.5$$

$$P(Y=3)=0.2$$

$$P(Y=4)=0.1$$

Të përcaktojmë intervalet përkatëse për P :

$$P=[0.2,0.7,0.9,1]$$

Ndërsa vlerat e Y janë:

$$Y=[1,2,3,4]$$

```
P = [0.2, 0.7, 0.9, 1]; % vlerat sipas intervaleve
```

```
Y = [1, 2, 3, 4]; % nje nga numrat e perzgjedhur do perfaqesoje Y-in
```

```
numerim = 1; % fillojme nga pozicioni pare
```

```
U = rand; % zgjedhim nje numer U nga shperndarja Uniforme
```

```
while (U > P(numerim))
```

```
    numerim = numerim + 1;
```

```
end
```

```
Y(numerim)
```

6.3 Kushti if-elseif-else

```

numri = -5;

if numri > 0
    disp('Numri është pozitiv.');
```

```
elseif numri < 0
    disp('Numri është negativ.');
```

```
else
    disp('Numri është zero.');
```

```
end
```

6.4 Kushti switch-case-otherwise

```

option = 'Z';

switch option

    case 'X'

        disp('Ju zgjodhët opsionin X.');
```

```
    case 'Y'

        disp('Ju zgjodhët opsionin Y.');
```

```
    case 'Z'

        disp('Ju zgjodhët opsionin Z.');
```

```
    otherwise

        disp('Opsioni nuk është në listë.');
```

```
end
```

Nëpërmjet këtij projekti, kemi arritur të:

1. Përdorim Vektorë dhe Matrica:

- Të krijojmë dhe manipulojmë vektorë dhe matrica, si dhe të kryejmë operacione aritmetike mbi to.
- Të zgjidhim sisteme ekuacionesh dhe të analizojmë vektorët dhe matricat për aplikacione të ndryshme.

2. Zbatojmë Metodën Statistikore:

- Të analizojmë të dhënat statistikore duke përdorur shpërndarje të ndryshme.
- Të llogarisim masat e përqendrimit dhe shpërndarjes, si mesatarja, varianca dhe devijimi standard.

3. Aplikojmë Funksionet Trigonometrike:

- Të përdorim funksionet trigonometrike për të kryer llogaritje të ndryshme që lidhen me këndet dhe distancat.

4. Analizojmë Rrënjët Komplekse:

- Të gjejmë dhe analizojmë rrënjët komplekse të ekuacioneve duke përdorur funksionet e MATLAB-it.

5. Studjojmë Shpërndarjet Statistike:

- Të analizojmë dhe vizualizojmë shpërndarje të ndryshme si normale, eksponenciale, Poisson, binomiale, gama, gjeometrike dhe uniforme.
- Të kuptojmë karakteristikat dhe aplikacionet e këtyre shpërndarjeve në kontekste reale.

6. Përdorim Ciklet dhe Strukturat Kondicionale:

- Të zgjidhim probleme komplekse duke përdorur ciklet **for** dhe **while** dhe strukturat kondicionale **if**, **else**.
- Të shkruajmë skripte të optimizuara dhe të strukturuar mirë për zgjidhjen e problemeve të ndryshme.

7. Ilustrojmë Teoremën Qendrore Limite:

- Të kuptojmë dhe demonstrojmë teoremën qendrore limite dhe rëndësinë e saj në statistikë.
- Të shohim sesi shpërndarja e mostrave konvergon në një shpërndarje normale me rritjen e madhësisë së mostrës.

Me këtë projekt, kemi arritur të krijojmë një themel të fortë për përdorimin e MATLAB-it në analizën dhe zgjidhjen e problemeve matematike dhe statistikore, duke përgatitur terrenin për hulumtime dhe aplikacione më të avancuara në të ardhmen.