5.5. Calcule cu funcții matematice

5.5.1 Evaluări matematice uzuale

În practica inginerească, în contextul unor modele matematice mai complexe este necesar de multe ori să se efectueze anumite evaluări numerice indispensabile. Astfel de evaluări sunt: determinarea semnului, determinarea restului împărțirii, aflarea divizorilor şi multiplilor comuni. Pentru acestea, Matlab pune la dispoziție funcții uşor de aplicat.

1. Funcția semn

Funcția semn, cu sintaxa `sign(x)` primeşte ca argument o variabilă vector şi returneaza un vector cu elementele -1, 0 sau 1, după următoarea regulă:

$$\mathrm{sign}(x) = \begin{cases} 1, & \text{daca} \quad x > 0 \\ 0, & \text{daca} \quad x = 0 \\ -1, & \text{daca} \quad x < 0 \end{cases}$$

De exemplu, determinarea semnului pentru elementele unui vector numeric:

```
>> x=[1 0 -2 -5 0.5 -2.6]
x =
    1.0000    0    -2.0000    -5.0000     0.5000    -2.6000

>> sign(x)
ans =
     1     0     -1     -1      1     -1
```

2. Funcția rest

Funcția rest, cu sintaxa `Z=rem(x,y)` returnează restul împărțirii lui x la y. În cazul vectorilor sau matricelor, operația se efectuează între elementele omoloage din fiecare poziție.

```
>> x=428
x =
   428

>> y=255
y =
  255

>> z=rem(x,y)
z =
  173
```

```
>> x=[1 10 -2]
x =
     1    10    -2

>> y=[0.5 2 3]
y =
    0.5000    2.0000    3.0000

>> z=rem(x,y)
z =
     0     0    -2
```

3) Funcții pentru calculul divizorilor şi multiplilor comuni

- Funcția `gcd` (*greatest common divizor*) calculează *cel mai mare divizor comun* a două numere întregi cu sintaxa `a=gcd(x,z)`.
- Funcția `lcm` (*least common multiple*) calculează *cel mai mic multiplu comun* a două numere întregi cu sintaxa `a=lcm(x,z)`.

```
>> x=[11 23 45 12 30]

x =
    11    23    45    12    30

>> y=[2 3 10 9 24]

y =
     2     3    10     9    24

>> a=gcd(x,y)

a =
     1     1     5     3     6
```

```
>> b=lcm(x,y)

b =
    22    69    90    36   120
```

Observație. În cazul variabilelor vector, operațiile se aplică între două elemente omoloage.

5.5.2 Funcții matematice de bază

Funcții elementare

În Matlab există o suită de funcții pentru operații analitice elementare, aşa cum sunt descrise în tabelul 5.4.

Tab. 5.4. Funcții matematice uzuale în Matlab

Operația	Simbol	Funcția/sintaxa Matlab
Exponențiere	e^x	`exp(x)`
Logaritmare cu diferite baze	$\ln(x)$	`log(x)`
	$\log_2(x)$	`log2(x)`
	$\lg(x)$	`Log10(x)`
Puterile lui 2	2^n	`pow2(n)`
	$m \cdot 2^n$	`pow2(m,n)`
Puterea N a lui 2 care majorează modulul argumentului	$\lvert arg \rvert \le 2^N$	`nextpow2(arg)`
Radicalul de ordinul doi	$\sqrt{x}$	`sqrt(x)`
Ridicare la putere	x^y	`x^y`

Funcții trigonometrice şi hiperbolice

Funcțiile *trigonometrice* (cunoscute şi sub numele de funcții circulare, armonice sau periodice) directe şi inverse, cât şi cele *hiperbolice* directe şi inverse sunt utilizate în mod curent în problemele de calcul ingineresc. Matlab pune la dispoziție întreaga colecție de funcții necesare acestor calcule care pot fi apelate cu sintaxa generală:

```
x=nume_functie(argument)
```

unde:

- `argument` este o variabilă sau constantă (scalar sau vector) reprezentând valoarea sau valorile pentru care se evaluează funcția;
- `nume_functie` este numele uneia dintre funcțiile descrise în tabelul 5.5;
- x este variabila în care se returnează rezultatul (valoarea sau valorile funcției).

Tab. 5.5 Funcții trigonometrice şi hiperbolice

Funcții		Formula/Definiție	Funcția Matlab
Trigonometrice directe	Sinus	$R \to [-1,1]$	`sin`
	Cosinus	$R \to [-1,1]$	`cos`
	Tangenta	$\frac{\sin(u)}{\cos(u)}$, $R \to R$	`tan`
	Cotangenta	$\frac{\cos(u)}{\sin(u)}$, $R \to R$	`cot`
	Secanta	$\frac{1}{\cos(u)}$, $R \to R$	`sec`
	Cosecanta	$\frac{1}{\sin(u)}$, $R \to R$	`csc`
Trigonometrice inverse	Arcsinus	$\sin(\arcsin(u)) = u$, $[-1,1] \to R$	`asin`
	Arccosinus	$\sin(\arcsin(u)) = u$, $[-1,1] \to R$	`acos`
	Arctangenta	$\sin(\arcsin(u)) = u$, $R \to R$	`atan` `atan2`
	Arccotangenta	$\sin(\arcsin(u)) = u$, $R \to R$	`acot`
	Arcsecanta	$\arccos(1/u)$, $(-\infty, -1]$ sau $[1, \infty) \to$ $\to [0, \pi/2)$ sau $(\pi/2, \pi]$	`asec`
	Arccosecanta	$\arcsin(1/u)$, $(-\infty, -1]$ sau $[1, \infty) \to$ $\to [-\pi/2, 0)$ sau $(0, \pi/2]$	`acsc`
Hiperbolice directe	Sinus hiperbolic	$sh(u) = (e^u - e^{-u})/2$	`sinh`
	Cosinus hiperbolic	$ch(u) = (e^u + e^{-u})/2$	`cosh`
	Tangenta hiperbolică	$th(u) = \frac{sh(u)}{ch(u)} = \frac{e^u - e^{-u}}{e^u + e^{-u}}$	`tanh`
	Cotangenta hiperbolică	$cth(u) = \frac{ch(u)}{sh(u)} = \frac{e^u + e^{-u}}{e^u - e^{-u}}$	`coth`
	Secanta hiperbolică	$\sec h(u) = \frac{1}{ch(u)} = \frac{2}{e^u + e^{-u}}$	`sech`
	Cosecanta hiperbolică	$\cos ech(u) = \frac{1}{sh(u)} = \frac{2}{e^u - e^{-u}}$	`csch`

Tab. 5.5 (*Continuare*)

Hiperbolice inverse	Argsinus hiperbolic	$\arg\mathrm{sh}(u) = \ln\left(u + \sqrt{u^2+1}\right)$	`asinh`
	Argcosinus hiperbolic	$\arg\mathrm{ch}(u) = \ln\left(u \pm \sqrt{u^2+1}\right)$	`acosh`
	Argtangenta hiperbolică	$\arg\mathrm{th}(u) = \frac{1}{2}\ln\left(\frac{1+u}{1-u}\right)$	`atanh`
	Argcotangenta hiperbolică	$\arg\mathrm{coth}(u) = \frac{1}{2}\ln\left(\frac{u+1}{u-1}\right)$	`acoth`
	Argsecanta hiperbolică	argch(1/u)	`asech`
	Argcosecanta hiperbolică	argsh(1/u)	`acsch`

Observații:

- Funcțiile trigonometrice directe primesc argumentul în *radiani*.
- Funcțiile trigonometrice inverse returnează rezultatul, de asemenea, în *radiani*.
- Dacă nu se respectă domeniul de definiție al unei funcții trigonometrice inverse, aceasta returnează un număr complex a cărui parte reală are valoarea corespunzătoare limitei intervalului de definiție.

Exemplificare.

```
>> x=sin(pi/6)
x =
    0.5000
```

`>> y=asin(2)`
`y =`
`   1.5708 - 1.3170i`
	`>> y=asin(1)`
	`y =`
	`    1.5708`

5.6. Prelucrarea datelor şi calcule statistice

5.6.1 Funcţii pentru prelucrarea datelor

Funcţiile pentru prelucrarea datelor implementate în Matlab sunt următoarele:

`max`	Determină cea mai mare componentă într-un tablou de date;
`min`	Determină cea mai mică componentă într-un tablou de date;
`sort`	Sortează elementele unui tablou de date în ordine crescătoare;
`sum`	Calculează suma;
`prod`	Calculează produsul;
`diff`	Calculează diferenţele dintre numere succesive;
`cumsum`	Calculează suma cumulată;
`cumprod`	Calculează produsul cumulat;
`find`	Găsirea şi eliminarea datelor eronate/valorilor nedefinite.

- *Cea mai mare componentă a unui vector sau matrice*:

```
M=max(X)
```

- *Cea mai mică componentă a unui vector sau matrice*:

```
m=min(X),
```

unde:

X – vector sau matrice,

M, m – scalari sau vectori linie conţinând elementele maxime/minime din fiecare coloană.

Cu sintaxa `[M,I]=max(X)` respectiv `[m,e]=min(X)` se returnează atât elementele scalare/vector, cât şi poziţia/poziţiile acestor elemente în cadrul vectorului/matricei de date. În cazul matricelor, indicii specifică numărul liniei în care se găseşte elementul minim sau maxim. Când există mai multe elemente egale, care îndeplinesc condiţia *min* sau *max*, atunci este returnat doar indicele de poziţie al primului element.

- *Sortarea elementelor* unui vector sau matrice în ordine crescătoare se poate face cu următoarele sintaxe de utilizare a funcţiei `sort`:

```
Y=sort(X)
[Y,I]=sort(X)
```

În cazul matricelor, sortarea se face pe *coloane*, iar indicii reprezintă numărul de ordine în cadrul coloanei respective (numărul liniei).

Exemplu. Să se aplice funcțiile de prelucrare a datelor menționate mai sus pentru datele conținute în vectorul `x=[-1 0 -5 1 5 6 3 4 9 10]`.

```
>> x=[-1 0 -5 1 5 6 3 4 9 10]
x =
    -1     0    -5     1     5     6     3     4     9    10

>> M=max(x)
M =
    10

>> m=min(x)
m =
    -5

>> [M,I]=max(x)
M =
    10
I =
    10

>> [m,e]=min(x)
m =
    -5
e =
     3

>> y=sort(x)
y =
    -5    -1     0     1     3     4     5     6     9    10


>> [y,J]=sort(x)

y =
    -5    -1     0     1     3     4     5     6     9    10

J =
     3     1     2     4     7     8     5     6     9    10
```

Exemplu. Să se aplice funcțiile de mai sus pentru prelucrarea datelor din matricea:

$$X=\begin{bmatrix} 3 & -4 & 2 \\ -10 & 0 & 8 \\ 0 & 12 & 5 \end{bmatrix}$$

```
>> M=max(X)

M =
      3    12     8


>> [M,I]=max(X)

M =
      3    12     8
I =
      1     3     2

>> m=min(X)

m =
   -10    -4     2


>> [m,e]=min(X)

m =
   -10    -4     2
e =
      2     1     1
```

```
>> Y=sort(X)

Y =
   -10    -4     2
     0     0     5
     3    12     8


>> [Y,I]=sort(X)

Y =
   -10    -4     2
     0     0     5
     3    12     8

I =

     2     1     1
     3     2     3
     1     3     2
```

- <u>*Calculul sumei elementelor*</u> unui vector: $s=\sum_{k=1}^{N} x_k$ se face utilizând funcția `sum` cu sintaxa:

`Y=sum(x)`

Dacă argumentul X este un vector, funcția `sum` returnează un scalar egal cu suma elementelor vectorului, iar dacă argumentul este o matrice, funcția returnează un vector linie care conține suma elementelor pe fiecare coloană.

- <u>*Calculul produsului elementelor*</u> unui vector: $p=\prod_{k=1}^{N} x_k$ se face utilizând funcția `prod` cu sintaxa:

`Y=prod(x)`

Dacă argumentul X este un vector, funcția `prod` returnează un scalar egal cu produsul elementelor vectorului, iar dacă argumentul este o matrice, funcția returnează un vector linie care conține produsul elementelor de pe fiecare coloană.

- <u>*Calculul sumei cumulate*</u>: $S_j=\sum_{k=1}^{j} x_k$ j=1,2,…,N, în sensul $S_j=x_1+(x_1+x_2)+(x_1+x_2+x_3)+\ldots+(x_1+x_2+\ldots+x_N)$ se obține aplicând funcția:

`Y=cumsum(X)`

Dacă argumentul X este un vector, funcția returnează un vector care conține pe fiecare poziție suma cumulată a elementelor vectorului, de forma:

$$Y=x_1, (x_1+x_2), (x_1+x_2+x_3), \ldots,(x_1+x_2+\ldots+x_N),$$

iar dacă argumentul este o matrice, funcția returnează o matrice care pe fiecare coloană conține suma cumulată a elementelor din coloana respectivă.

- <u>*Calculul produsului cumulat*</u>: $p_j=\prod_{k=1}^{j} x_k$ j=1,2,…,N, în sensul

$p_j= x_1(x_1x_2)(x_1x_2x_3)\ldots(x_1x_2\ldots x_N)$ se obține prin aplicarea funcției:

`Y=cumprod(x)`

Dacă argumentul X este un vector, funcția returnează un vector care conține pe fiecare poziție produsul cumulat al elementelor vectorului, de forma:

$$Y= x_1, (x_1x_2), (x_1x_2x_3), \ldots,(x_1x_2\ldots x_N),$$

iar dacă argumentul este o matrice, funcția returnează o matrice care pe fiecare coloană conține produsul cumulat al elementelor din coloana respectivă.

Exemplu. Aplicarea funcțiilor pentru sume şi produse simple şi cumulate pentru date conținute într-un vector, respectiv într-o matrice.

```
% vector de date
>> x=[-1 0 -5 1 5 6 3 4 9 10]

x =
    -1     0    -5     1     5     6     3     4     9    10

>> s=sum(x)

s =
    32

>> p=prod(x)

p =
     0

>> sc=cumsum(x)
sc =
    -1    -1    -6    -5     0     6     9    13     22    32

>> pc=cumprod(x)
pc =
    -1     0     0     0     0     0     0     0     0     0
```

```
%matrice de date
>>X=[3 -4 2;-10 0 8;0 12
5]
X =
     3    -4     2
   -10     0     8
     0    12     5

>> S=sum(X)
S =
    -7     8    15

>> P=prod(X)
P =
     0     0    80
```

```
>> SC=cumsum(X)
SC =
     3    -4     2
    -7    -4    10
    -7     8    15


>> PC=cumprod(X)
PC =
     3    -4     2
   -30     0    16
     0     0    80
```

- *Identificarea pozițiilor valorilor nedefinite*, adică a valorii NaN (Not-a-Number) într-un vector:

```
I=find(isnan(x)),
```

unde I – vectorul indicilor (poziției) elementelor NaN.

- *Eliminarea datelor eronate* se poate face cu ajutorul funcției `find`.

Efectul acestei funcții este de a găsi datele care nu îndeplinesc o condiție impusă de utilizator. Eliminarea propriu zisă a acestora din cadrul structurii de date (vector sau matrice) se face prin operații de manipulare a matricelor.

Exemplu. Se crează un vector care conține şi elemente ce rezultă în urma unor calcule nepermise (nedeterminări de tipul $\frac{0}{0}$). Să se identifice aceste elemente şi apoi să se elimine.

```
>> a=[0 1 3 0 0 5 10 0 6 1]

a =
     0     1     3     0     0     5    10     0     6     1

>> b=[0 1 0 10 4 9 5 0 0 0]

b =
     0     1     0    10     4     9     5     0     0     0

>> x=a./b

x =
   NaN  1.00  Inf    0    0   0.55  2.00   NaN  Inf   Inf

>> I=find(isnan(x))

I =
     1     8

% se elimina elementele gasite
>> y=x(:,[2:7,9:10])

y =
   1.00  Inf   0   0   0.55   2.00   Inf   Inf
```

5.6.2 Funcții pentru calcule statistice în Matlab

Funcțiile Matlab pentru calcule statistice asupra seturilor de date sunt următoarele:

`hist`	Reprezintă histograma
`mean`	Calculează valoarea medie
`std`	Calculează deviația standard
`median`	Calculează valoarea mediană
`corcoef`	Calculează coeficienții de corelație
`cov`	Calculează matricea de covarianță

a) Conceptul de variabilă aleatoare

Din punct de vedere formal, prin variabilă aleatoare se înțelege o aplicație a unui câmp de evenimente arbitrare pentru care nu cunoaştem un sistem complet generator, pe mulțimea numerelor reale. Practic, aplicația menționată atribuie unei caracteristici oarecare a unui sistem valori numerice arbitrare.

În sistemele tehnice şi în natură în general, variabilele aleatoare sunt caracterizate de anumite *funcții (legi) de repartiție*. Acestea se referă la *distribuția frecvențelor de apariție* a valorilor numerice arbitrare (ca rezultat al unor observații sau măsurători) pentru o caracteristică dată, ceea ce echivalează cu *probabilitatea de apariție*. De exemplu, într-un proces de fabricație în serie a rezistoarelor, valoarea rezistenței electrice nu este aceeaşi la toate componentele ci are o anumită distribuție pe un interval numeric în vecinătatea valorii nominale (prescrise).

În statistica matematică există diverse legi de probabilitate care se definesc prin *densitatea de repartiție* pe intervale date. Două tipuri de legi de probabilitate remarcabile sunt:

- *legea (repartiția) normală standard* descrisă prin curba (clopotul) lui Gauss,
- *legea (repartiția) uniformă* care nu se poate descrie printr-un anumit tip de curbă regulată.

În inginerie se utilizează şi alte legi de repartiție statistică, care în timp au dovedit că reflectă bine anumite fenomene şi procese, cum sunt: repatiția Poision, repartiția Weibull, repartiția χ^2, repartiția Student, repartiția exponențială, repartiția log-normală, repartiția binomială. Pentru detalii complete asupra acestor repartiții de probabilitate importante recomandăm consultarea lucrării [20] din lista de referințe bibliografice.

În aplicațiile inginerești se folosesc diverse caracteristici numerice și indicatori grafici referitor la variabilele aleatoare, care permit să ne facem o idee asupra repartiției valorilor.

- *Histograma*

Histograma este un tip de reprezentare grafică utilizată preponderent în statistică pentru ilustrarea legilor de repartiție probabiliste. În Matlab se utilizează funcția `hist` cu următoarele sintaxe:

```
hist(x),
hist(x,N),
```

unde: x – setul de date, N – număr de intervale de reprezentare (implicit 10).

Cu sintaxele:

```
[N,Y]=hist(x),
[N,Y]=hist(x,N),
```

se dă posibilitatea de a obține numărul de elemente N din fiecare interval de reprezentare și valorile variabilei aleatoare Y corespunzătoare fiecărui interval de reprezentare.

De exemplu, histograma valorilor nominale a rezistenței electrice pentru o mie de bucăți de rezistoare obținute în condiții normale de fabricație este reprezentată în figura 5.2 pe N=20 intervale. În figura 5.3 este reprezentată densitatea de repartiție corespunzătoare legii normale (Gausiene) care aproximează distribuția de valori obținute și care se exprimă analitic prin relația:

$$\rho(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]}$$

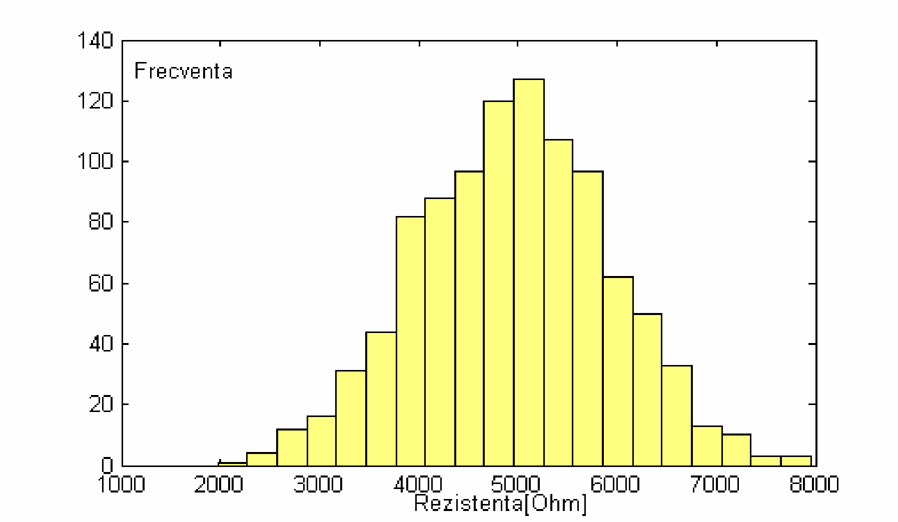


Fig. 5.2. Histograma repartiției normale pentru N=20.

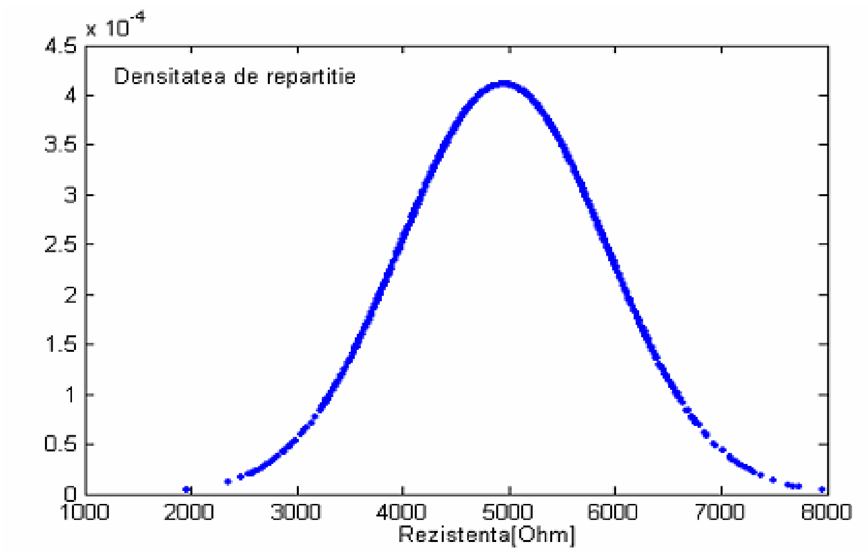


Fig. 5.3. Curba de repartiție Gauss

b) Conceptul de valoare medie şi dispersie

Într-o serie de date obținute din *N* determinări (care pot fi rezultatul unor măsurători, determinări experimentale, generare prin calcul, etc.), caracteristica înregistrată X este o variabilă aleatoare cu valori numerice care apar într-o ordine oarecare: $x_1, x_2, ..., x_{N-1}, x_N$.

- *Calculul mediei aritmetice*

$$\mu = \frac{\sum_{k=1}^{N} x_k}{N} = \frac{x_1 + x_2 + ... + x_N}{N}$$

Se aplică cu sintaxa `m=mean(X)`, în care:

- X poate să fie vector de date şi rezultă *media* elementelor, un scalar *m*;
- X poate fi matrice şi rezultă *m*, un vector linie ale cărui elemente sunt *mediile* efectuate pe fiecare coloană.

Notă. Media este un indicator statistic care ne dă o informaţie cu privire la *centrarea datelor*. În diferite probleme valoarea medie este numită şi *aşteptare* matematică sau *speranţă* matematică.

- *Varianţa şi deviaţia standard*

Varianţa unui set de date reprezintă media aritmetică a pătratelor diferenţelor dintre fiecare valoare şi media acestora şi se calculează cu expresia:

$$\sigma^2 = \frac{\sum_{k=1}^{N} (x_k - \mu)^2}{N-1},$$

unde termenul $(x_k - \mu)$ reprezintă diferenţa sau abaterea lui x_k faţă de valoarea medie μ. Prin urmare, varianţa reprezintă *pătratul abaterii medii pătratice* a datelor faţă de medie.

Observaţie. Varianţa unui şir de date numerice nu are aceeaşi dimensiune ca a datelor.

Deviația standard este definită ca rădăcina pătrată a varianței, $\sigma = \sqrt{\sigma^2}$ şi are semnificația unei *abateri medii pătratice* ce se calculează cu expresia:

$$\sigma = \sqrt{\frac{\sum_{k=1}^{N}(x_k - \mu)^2}{N-1}} = \sqrt{\sigma^2}\ .$$

Acest indicator statistic dă o măsură asupra *dispersiei* (împrăţtierii) datelor în raport cu valoarea lor medie. Funcția Matlab pentru calculul deviației standard datelor se apelează cu sintaxa:

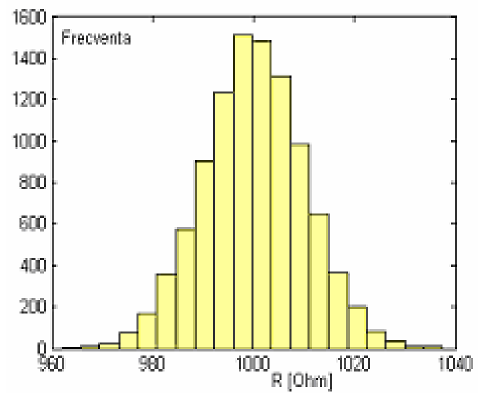
```
s=std(X)
```

Argumentul `X` este vector sau matrice, iar `s` scalar sau vector ce reprezintă valoarea/valorile abaterii medie pătratice. Dacă X este un vector, funcția `std` returnează un scalar care reprezintă abaterea medie pătratică a elementelor vectorului. Pentru matrice, funcția returnează un vector linie conținând abaterea medie pătratică a fiecărei coloane.

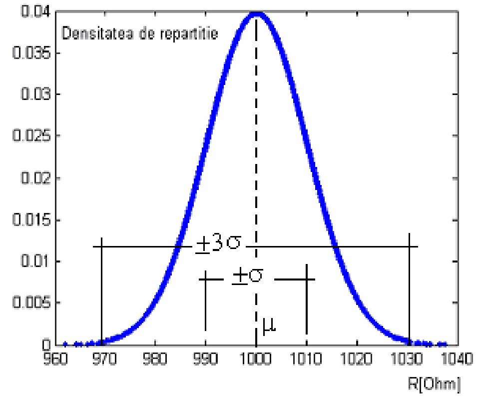
Observații.

- Varianța şi deviația standard reprezintă măsuri ale *dispersiei* sau împrăştierii datelor.
- Media reprezintă *indicele de centrare* în timp ce deviația standard constituie *indicele de împrăştiere* a datelor față de valoarea lor medie.
- Valoarea deviației standard depinde de *întinderea histogramei* şi de modul în care se *repartizează* datele pe întinderea respectivă.
- Deviația standard se exprimă în aceleaşi unități de măsură ca ale datelor.
- Un alt indice al dispersiei (împrăştierii) datelor este *amplitudinea* (domeniul de întindere) care reprezintă diferența între valoarea maximă X_M şi valoarea minimă X_m a caracteristicii considerate în cadrul unui lot sau a unei selecții.

> Aplicația 1. Să se calculeze indicatorii statistici *valoare medie* şi *deviația standard* precum şi amplitudinea dispersiei pentru o repartiție normală a valorilor rezistenței în procesul de fabricație a unui lot de zece mii de rezistoare. Să se evidențieze grafic semnificația parametrilor calculați.

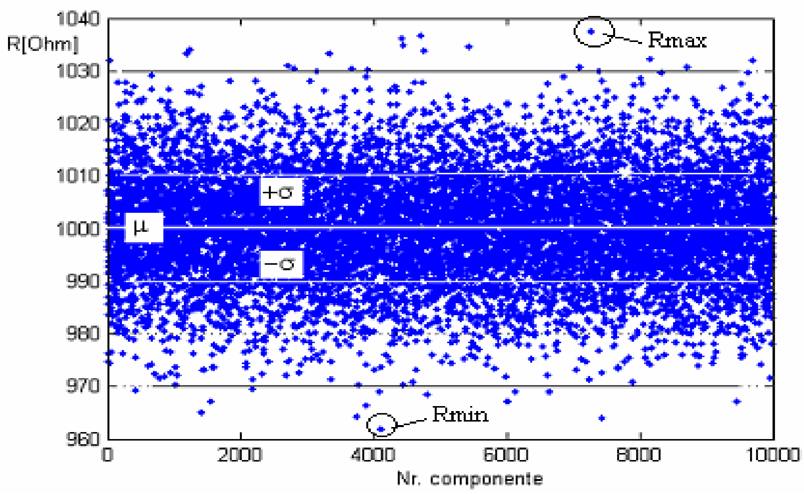


Histograma frecvențelor relative



Repartiția normală a valorilor

Distribuția valorilor aleatoare ale rezistenței în procesul de fabricație a lotului de 10.000 de rezistoare.

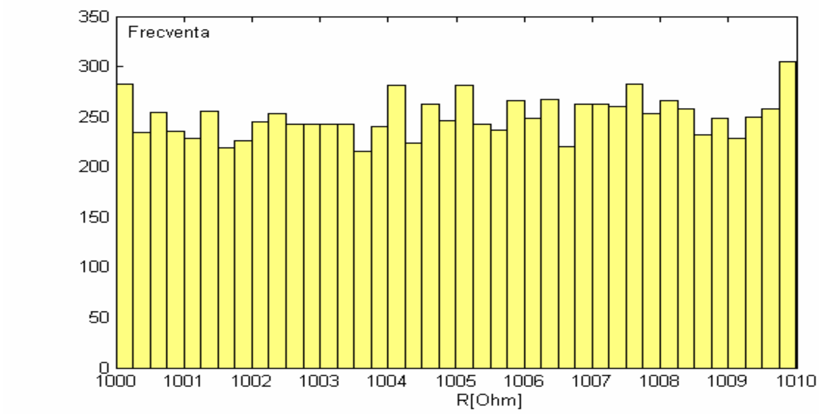


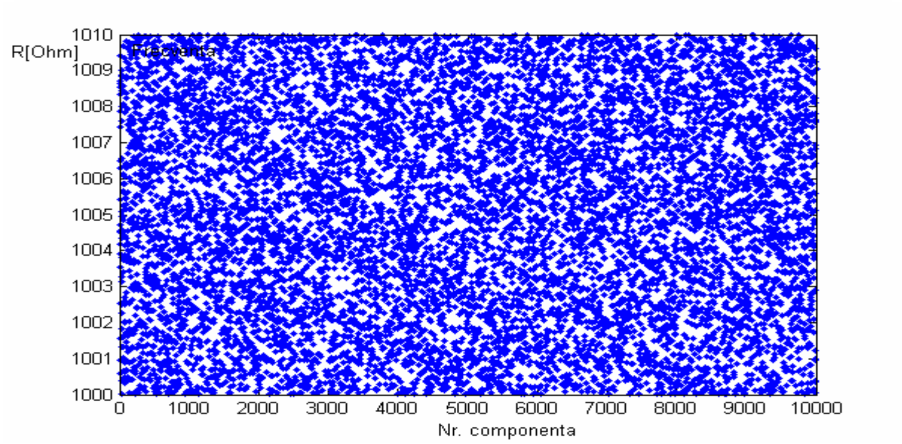
Reprezentarea grafică a valorilor rezistenței pentru întregul lot

```
>> media=mean(R)

media =
       1.0001e+003


>> sigma=std(R)

sigma =
       10.0518
```

```
>> [Rmax,I]=max(R)
Rmax =
      1.0375e+003
I =
    7255
>> [Rmin,J]=min(R)
Rmin =
      962.0114
J =
    4108
>> Amplitudinea=Rmax-Rmin
Amplitudinea =
              75.4647
```

Aplicația 2. Să se rezolve cerințele aplicației precedente presupunând legea de repartiție uniformă a valorilor rezistenței în procesul de fabricație.






```
>> media=mean(R)

media =
    1.0051e+003

>> sigma=std(R)

sigma =
    2.8966
```

```
>> [Rmin,J]=min(R)
Rmin =
     1.0000e+003
J =
      5804
>> [Rmax,I]=max(R)
Rmax =
   1.0100e+003

I =
      3411
>> Amplitudinea=Rmax-Rmin
Amplitudinea =
     9.9979
```

Considerații de ordin practic

Exemplele aplicative de mai sus se referă la un proces de producție din industria electronică pentru care facem următoarele mențiuni:

Cazul repartiției normale (Gauss)	Cazul repartiției uniforme
Corespunde proceselor de fabricație unde fenomenele sunt normale, în sensul că: - în general, nu intervin influențe sistematice (ci doar întâmplătoare), - utilajele au parametrii de precizie prevăzuți, - materialele şi materiile prime sunt omogene ca structură şi caracteristici, - muncitorul executant este pregătit corespunzător.	Corespunde proceselor de fabricație unde nu există normalitate, iar rezultatele obținute sunt haotice, neregulate datorată următoarelor aspecte: - utilajele nu funcționează în parametrii de precizie corespunzători, - materialele nu sunt omogene ca proprietăți, - muncitorul executant are nivelul profesional scăzut.

- *Calculul medianei*

Mediana este elementul de la mijlocul unui set de date ordonat crescător. Dacă numărul N al valorilor setului este impar, valoarea mediană este cea din poziția (N+1)/2, iar dacă acesta este par, valoarea mediană este media elementelor din pozițiile (N/2) şi (N/2)+1, adică:

$$\text{median}\,(x) = \begin{cases} x\left(\frac{N+1}{2}\right) & \text{daca} \quad N = 2k+1 \\ \frac{x\left(\frac{N}{2}\right)+x\left(\frac{N}{2}+1\right)}{2} & \text{daca} \quad N = 2k \end{cases}$$

Se aplică sintaxa `M=median(x)`, în care:

x este un vector → M scalar = valoarea *mediană* a datelor, sau x este o matrice → m vector linie ce conține valoarea *mediană* a fiecărei coloane.

Observație. Funcția `median` operează chiar dacă şirul numeric nu este ordonat crescător în prealabil (cu funcția `sort`, de pildă), deoarece implicit ea efectuează operația de ordonare.

Exemplu. Să se calculeze valoarea mediană pentru un şir de 10 numere generate aleator cu distribuţie normală (media≅0 şi deviaţia standard≅1). Să se adauge al 11-lea număr şi să se recalculeze valoarea mediană al şirului astfel obţinut.

```
>> x=randn(1,10)

x =
   -0.4326   -1.6656    0.1253    0.2877   -1.1465
1.1909    1.1892   -0.0376    0.3273    0.1746


>> y=sort(x)

y =
   -1.6656   -1.1465   -0.4326   -0.0376    0.1253
0.1746    0.2877    0.3273    1.1892    1.1909


>> M=median(x)

M =
    0.1500


 x1 =
    -0.4326   -1.6656    0.1253    0.2877   -1.1465
1.1909    1.1892   -0.0376    0.3273    0.1746  1.0000


>> y1=sort(x1)

y1=
   -1.6656   -1.1465   -0.4326   -0.0376    0.1253
0.1746    0.2877    0.3273    1.0000    1.1892  1.1909


>> M1=median(x1)

M1 =
    0.1746
```

c) Conceptul de corelaţie şi covarianţă

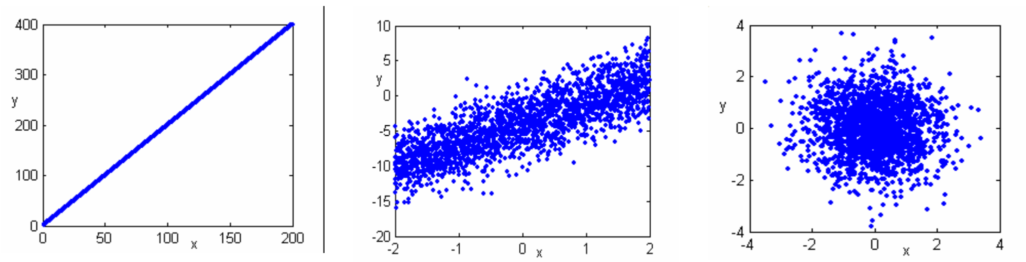
În practică pot fi luate în considerare procese în care se manifestă două caracteristici ale unui proces probabilist, iar în acest caz problema se extinde la două variabile aleatoare. Două caracteristici ale unei colectivități statistice se pot prezenta sub următoarele aspecte:

- cele două caracteristici sunt *independente* una față de alta;
- între cele două caracteristici există un *grad de dependență* probabilistică; care poate da naştere la o *relație funcțională* certă;
- între cele două caracteristici poate exista o *corelație* atât de puternică încât se poate transforma într-o *relație funcțională* certă.

În domeniul ingineriei se lucrează în general cu *relații funcționale deterministe*, dar există numeroase cazuri de corelație statistică între diferite caracteristici ale sistemelor tehnice. Astfel, se poate presupune că urmărind două caracteristici, sau doi parametri de stare X şi Y ai unui sistem şi efectuând o serie de determinări experimentale (*n* observații) se obțin vectorii datelor de observație:

		Observații succesive
Caracteristica	X	$x_1, x_2, x_3, \dots, x_n$
	Y	$y_1, y_2, y_3, \dots, y_n$

Repartiția empirică a celor două variabile X şi Y se reprezintă grafic într-un sistem de coordonate xOy prin punctele de coordonate (x_i, y_i), care formează un *câmp de corelație* sau *nor statistic* ce poate avea diferite forme şi distribuții aşa cum se exemplifică în figura 4.7.



a) Dependență funcțională b) Dependență corelațională c) Lipsa totală a corelației

Fig. 5.4. Aspecte posibile ale câmpului de corelație

- *Coeficientul de corelație a datelor*

Definiție. Dacă u şi v sunt două variabile aleatoare şi $\bar{u}$ respectiv $\bar{v}$ sunt valorile lor medii, atunci valoarea medie a produsului celor două abateri $(u-\bar{u})$ şi $(v-\bar{v})$, adică $\mu=\frac{1}{n}\sum_{k=1}^{n}(u_k-\bar{u})\cdot(v_k-\bar{v})$ se numeşte *corelaţie* sau *covarianţă* a variabilelor aleatoare u şi v. Indicatorul cantitativ al corelaţiei este *coeficientul de corelaţie*, care se calculează cu expresia:

$$r=\frac{\sum_{k=1}^{n}(u_k-\bar{u})\cdot(v_k-\bar{v})}{n\cdot\sigma_u\cdot\sigma_v}$$

Funcţia Matlab `corrcoef` stabileşte dacă între două seturi (observaţii) de date (variabile), înregistrate în vectorii notaţi cu u şi v, există *o dependenţă liniară*, calculând *coeficientul de corelaţie* cu relaţia următoare:

$$R_{uv}=\frac{N\sum_{i=1}^{N}u_iv_i-\sum_{i=1}^{N}u_i\sum_{i=1}^{N}v_i}{\sqrt{\left[N\sum_{i=1}^{N}u_i^2-\left(\sum_{i=1}^{N}u_i\right)^2\right]\left[N\sum_{i=1}^{N}v_i^2-\left(\sum_{i=1}^{N}v_i\right)^2\right]}}$$

Apelată cu sintaxa:

```
Rxy=corrcoef(X,Y),
```

funcţia primeşte ca argumente vectorii datelor observate (în linie) şi returnează coeficientul de corelaţie dintre liniile de date observate sub forma unei matrice de ordinul doi simetrice, ale cărei elemente reprezintă coeficientul de corelaţie în formatul următor:

Coeficientul de corelaţie între X şi X	*Coeficientul de corelaţie între* X şi Y
Coeficientul de corelaţie între Y şi X	*Coeficientul de corelaţie între* Y şi Y

Funcţia poate fi apelată şi cu sintaxa:

```
Rx=corrcoef(X),
```

care primeşte ca argument o matrice X ale cărei linii conţin *datele observate* (variabilele), iar coloanele corespund *caracteristicilor urmărite (monitorizate)* şi returnează matricea coeficienţilor de corelaţie în formatul prezentat.

	Date/Observații succesive
Caracteristica urmărită	$x_1, x_2, x_3, \dots, x_n$
	$y_1, y_2, y_3, \dots, y_n$
	:
	$w_1, w_2, \dots, w_n$

Matricea coeficienţilor de corelaţie având proprietatea de simetrie R(i,j)=R(j,i), se poate găsi în una dintre situaţiile următoare:

$$R = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \text{ sau } R = \begin{bmatrix} 1 & 0\cdots1 \\ 0\cdots1 & 1 \end{bmatrix},$$

în care 0…1 reprezintă intervalul pentru o valoare posibilă a coeficientului de corelaţie între X şi Y respectiv între Y şi X.

Interpretarea coeficientului de corelaţie

Coeficientul de corelaţie este o mărime abstractă care nu depinde de unităţile în care sunt exprimate variabilele x şi z. El exprimă *o măsură a dependenţei liniare* dintre cele două caracteristici ale sistemului, iar valorile sale se interpretează astfel:

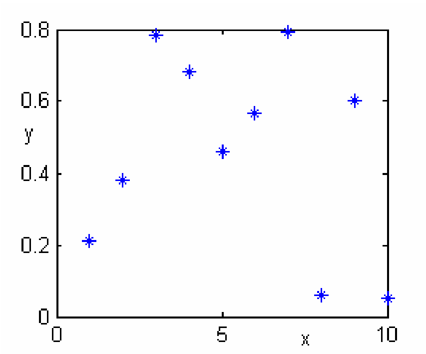
- dacă **R=0**, între cele două caracteristici ale sistemului nu poate exista vreo corelaţie definită printr-o dependenţă funcţională liniară, acest caz neexcluzând existenţa unei eventuale dependenţe neliniare;
- dacă **R=±1**, între caracteristicile considerate există o corelaţie de forma unei dependenţe funcţionale perfect liniare, în care odată cu creşterea lui x creşte şi y dacă valoarea este +1, sau cu creşterea lui x descreşte y dacă valoarea este −1.
- Dacă **R=0...1** rezultă un anumit grad de dependenţă liniară, care se evidenţiază în tendinţa globală de ordonare a datelor pe o anumită direcţie.

Exemplu. Să se determine corelația dintre două caracteristici numerice x şi y ale unui sistem obținute prin 10 observații succesive.

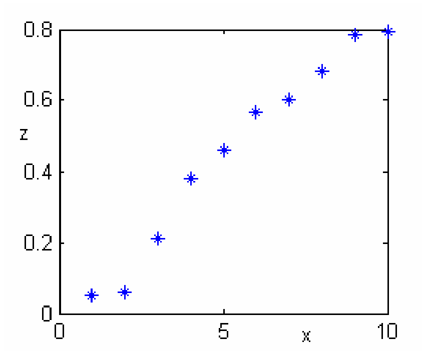
Rezolvare. S-au ales doi vectori numerici cu dimensiunea 10, primul generat cu pas liniar şi al doilea generat aleator cu distribuție uniformă. Se aplică funcția `corrcoef` şi se constată lipsa corelării între cele două seturi de date, fapt ilustrat în figura 5.5a. Apoi, se face ordonarea crescătoare a vectorului de date aleatoare (aplicând funcția `sort`) obținându-se vectorul z şi se testează din nou corelația celor două seturi de valori (x şi z). În acest caz se constată un coeficient de corelație apropiat de unu, fapt ilustrat şi prin dependența funcțională aproape liniară din figura 5.5b.

```
x =
   1    2    3     4     5     6     7     8     9    10
>> y=rand(1,10)

y =
   0.2091  0.3798  0.7833  0.6808  0.4611  0.5678
0.7942   0.0592   0.6029   0.0503


>> r=corrcoef(x,y)

r =
    1.0000   -0.2008
   -0.2008    1.0000
```

```
x =
   1    2    3     4     5     6     7     8     9    10
>> z=sort(y)

z =
   0.0503   0.0592   0.2091   0.3798   0.4611   0.5678
0.6029   0.6808    0.7833    0.7942


>> r=corrcoef(x,z)

r =
    1.0000    0.9838
    0.9838    1.0000
```



a) Lipsa corelației



b) Corelație aproximativ liniară

Fig. 5.5. Efecte de corelare a două caracteristici x şi y