

## 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 și multiplilor comuni. Pentru acestea, Matlab pune la dispoziție funcții ușor de aplicat.

#### 1. Funcția semn

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

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

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 și multiplilor comuni

- Funcția `gcd` (*greatest common divisor*) 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, așa cum sunt descrise în tabelul 5.4.

Tab. 5.4. Funcții matematice uzuale în Matlab

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

#### Funcții trigonometrice și hiperbolice

Funcțiile *trigonometrice* (cunoscute și sub numele de funcții circulare, armonice sau periodice) directe și inverse, cât și cele *hiperbolice* directe și 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 și hiperbolice

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

Tab. 5.5 (Continuare)

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

<pre>&gt;&gt; x=sin(pi/6)  x =      0.5000</pre>	
<pre>&gt;&gt; y=asin(2)  y =      1.5708 - 1.3170i</pre>	<pre>&gt;&gt; y=asin(1)  y =      1.5708</pre>

## 5.6. Prelucrarea datelor și calcule statistice

### 5.6.1 Funcții pentru prelucrarea datelor

Funcțiile 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 diferențele dintre numere succesive;
cumsum	Calculează suma cumulată;
cumprod	Calculează produsul cumulativ;
find	Găsirea și 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 conținând elementele maxime/minime din fiecare coloană.

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

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

$$Y = \text{sort}(X)$$

$$[Y, I] = \text{sort}(X)$$

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
```

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

$$Y=\text{sum}(x)$$

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.



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

$$Y = \text{prod}(x)$$

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.

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

$$Y = \text{cumsum}(X)$$

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

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

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

- Calculul produsului cumulat:  $p_j = \prod_{k=1}^j x_k$   $j=1,2,\dots,N$ , în sensul  $p_j = x_1(x_1 x_2)(x_1 x_2 x_3) \dots (x_1 x_2 \dots x_N)$  se obține prin aplicarea funcției:

$$Y = \text{cumprod}(x)$$

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

$$Y = x_1, (x_1 x_2), (x_1 x_2 x_3), \dots, (x_1 x_2 \dots x_N),$$

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 și produse simple și 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 și elemente ce rezultă în urma unor calcule nepermise (nedeterminări de tipul  $\frac{0}{0}$ ). Să se identifice aceste elemente și 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 cunoaștem un sistem complet generator, pe mulțimea numerelor reale. Practic, aplicația menționată atribuie unei caracteristici oarecare a unui sistem valori numerice arbitrare.

În sistemele tehnice și 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 aceeași 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 și alte legi de repartiție statistică, care în timp au dovedit că reflectă bine anumite fenomene și procese, cum sunt: repartiția Poisson, *repartiția Weibull*, repartiția  $\chi^2$ , 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 ingineresti 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:

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

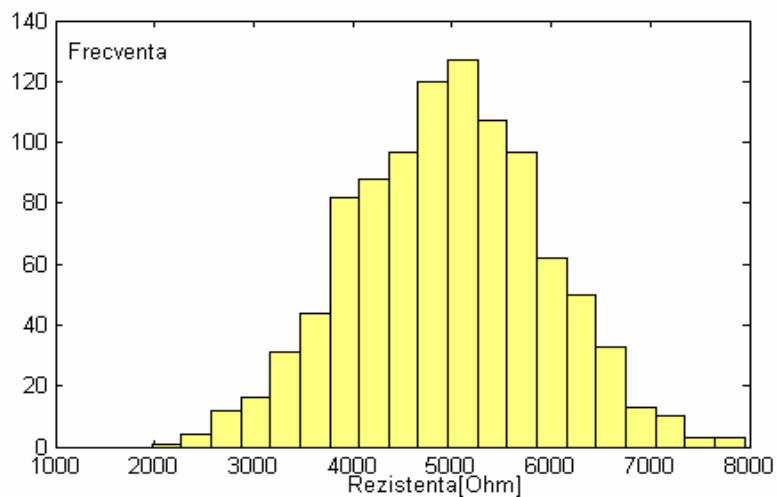


Fig. 5.2. Histograma repartiției normale pentru  $N=20$ .

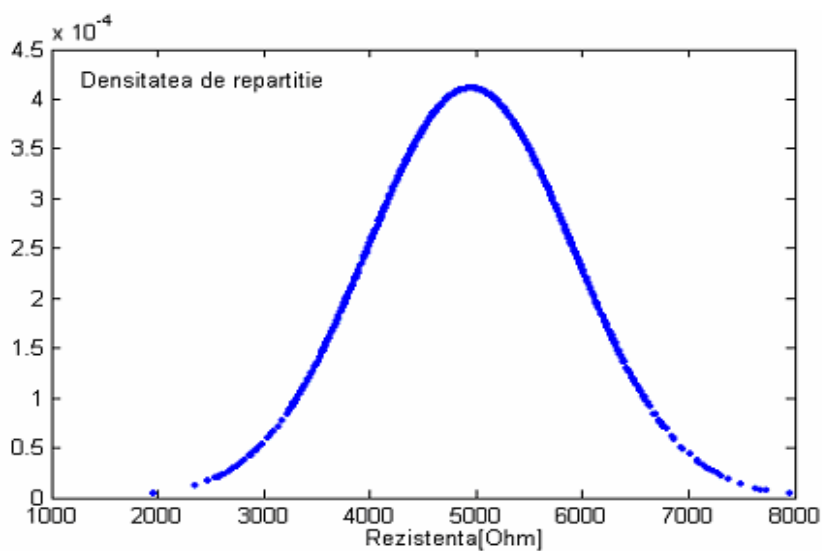


Fig. 5.3. Curba de repartiție Gauss

## b) Conceptul de valoare medie și 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:  $x_1, x_2, \dots, x_{N-1}, x_N$ .

▪ *Calculul mediei aritmetice*

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

Se aplică cu sintaxa  $m = \text{mean}(X)$ , în care:

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

*Notă.* Media este un indicator statistic care ne dă o informație cu privire la *centrarea datelor*. În diferite probleme valoarea medie este numită și *așteptare* matematică sau *speranță* matematică.

▪ *Varianța și deviația standard*

*Varianța* unui set de date reprezintă media aritmetică a pătratelor diferențelor dintre fiecare valoare și media acestora și se calculează cu expresia:

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

unde termenul  $(x_k - \mu)$  reprezintă diferența sau abaterea lui  $x_k$  față de valoarea medie  $\mu$ . Prin urmare, varianța reprezintă *pătratul abaterii medii pătratice* a datelor față de medie.

*Observație.* Varianța unui șir de date numerice nu are aceeași dimensiune ca a datelor.

*Deviația standard* este definită ca rădăcina pătrată a varianței,  $\sigma = \sqrt{\sigma^2}$  și 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* (împrăstierii) datelor în raport cu valoarea lor medie. Funcția Matlab pentru calculul deviației standard datelor se apelează cu sintaxa:

$$s = \text{std}(X)$$

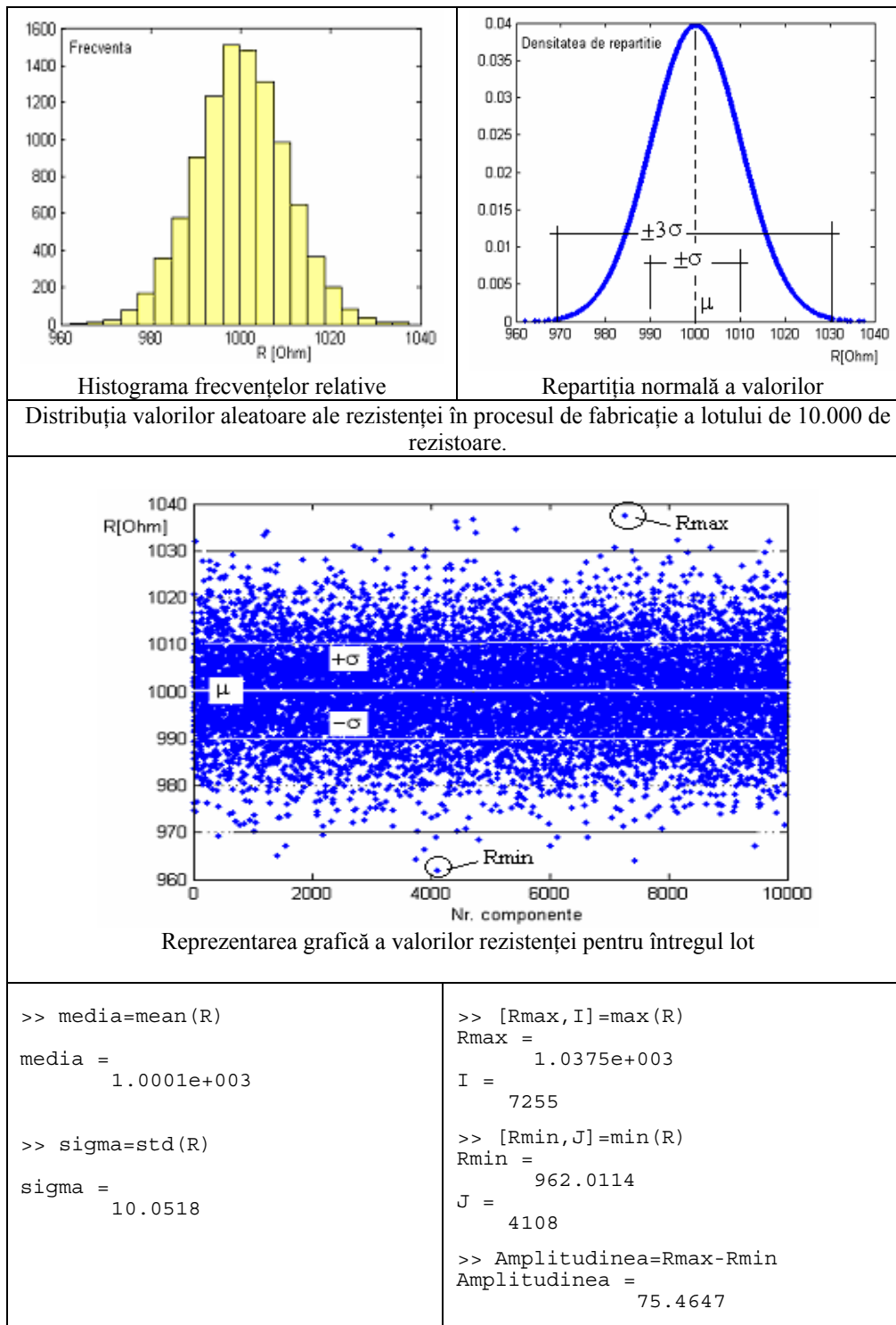
Argumentul  $X$  este vector sau matrice, iar  $s$  scalar sau vector ce reprezintă valoarea/valorile abaterii medii pătratice. Dacă  $X$  este un vector, funcția `std` returnează un scalar care reprezintă abaterea medie pătratică a elementelor vectorului. Pentru matrice, funcția returnează un vector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

*Observații.*

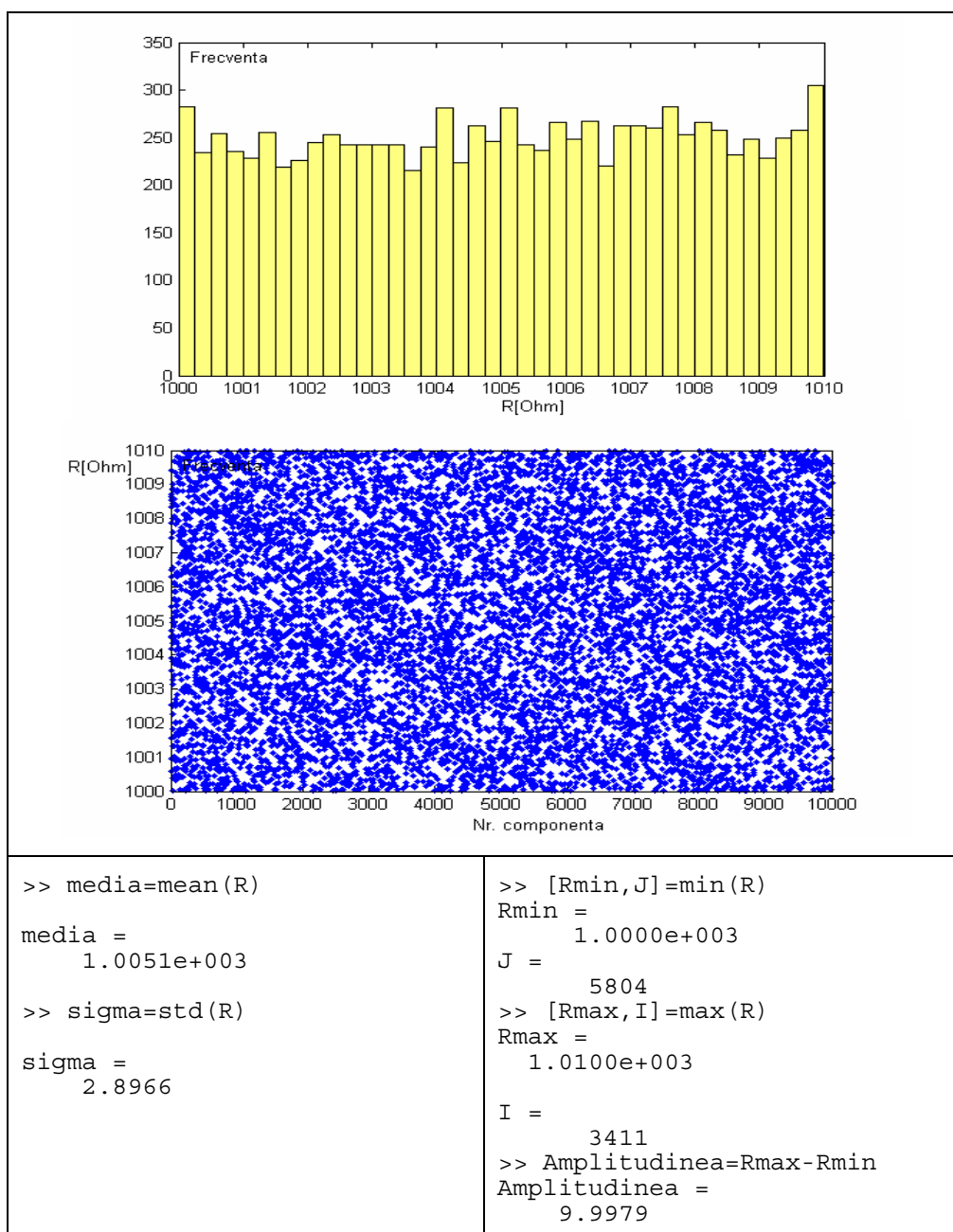
- Varianța și deviația standard reprezintă măsuri ale *dispersiei* sau *împrăstierii* datelor.
- Media reprezintă *indicele de centrare* în timp ce deviația standard constituie *indicele de împrăștiere* a datelor față de valoarea lor medie.
- Valoarea deviației standard depinde de *întinderea histogramei*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

Aplicația 1. Să se calculeze indicatorii statistici *valoare medie* și *deviația standard* precum și 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.





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



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
<p>Corespunde proceselor de fabricație unde fenomenele sunt normale, în sensul că:</p> <ul style="list-style-type: none"> <li>- în general, nu intervin influențe sistematice (ci doar întâmplătoare),</li> <li>- utilajele au parametri de precizie prevăzuți,</li> <li>- materialele și materiile prime sunt omogene ca structură și caracteristici,</li> <li>- muncitorul executant este pregătit corespunzător.</li> </ul>	<p>Corespunde proceselor de fabricație unde nu există normalitate, iar rezultatele obținute sunt haotice, neregulate datorată următoarelor aspecte:</p> <ul style="list-style-type: none"> <li>- utilajele nu funcționează în parametri de precizie corespunzători,</li> <li>- materialele nu sunt omogene ca proprietăți,</li> <li>- muncitorul executant are nivelul profesional scăzut.</li> </ul>

▪ *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) și (N/2)+1, adică:

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

Se aplică sintaxa  $M = \text{median}(x)$ , 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 șirul 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 șir de 10 numere generate aleator cu distribuție normală (media $\approx$ 0 și deviația standard $\approx$ 1). Să se adauge al 11-lea număr și să se recalculeze valoarea mediană al șirului astfel obținut.

```
>> 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 corelație și covarianță

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 naștere 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 și Y ai unui sistem și efectuând o serie de determinări experimentale ( $n$  observații) se obțin vectorii datelor de observație:

$$\begin{array}{c|c|c} & \text{Observații succesive} & \\ \text{Caracteristica} & \begin{array}{c} X \\ Y \end{array} & \begin{array}{c} x_1, x_2, x_3, \dots, x_n \\ y_1, y_2, y_3, \dots, y_n \end{array} \end{array}$$

Repartiția empirică a celor două variabile X și 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 și distribuții așa cum se exemplifică în figura 4.7.

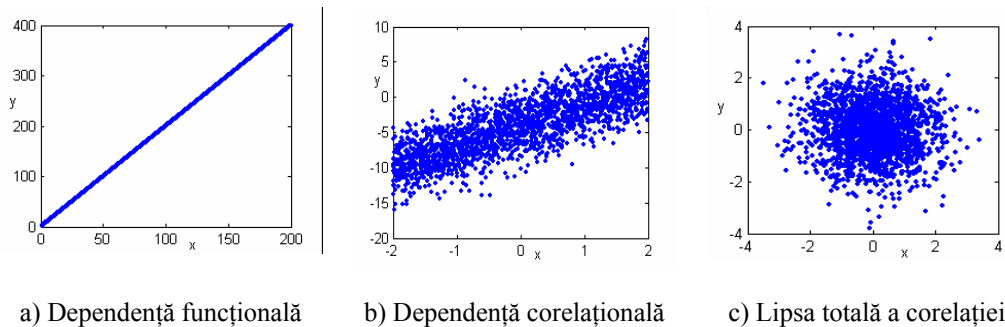


Fig. 5.4. Aspecte posibile ale câmpului de corelație

▪ *Coeficientul de corelație a datelor*

*Definiție.* Dacă  $u$  și  $v$  sunt două variabile aleatoare și  $\bar{u}$  respectiv  $\bar{v}$  sunt valorile lor medii, atunci valoarea medie a produsului celor două

abateri  $(u - \bar{u})$  și  $(v - \bar{v})$ , adică  $\mu = \frac{1}{n} \sum_{k=1}^n (u_k - \bar{u}) \cdot (v_k - \bar{v})$  se numește *corelație* sau *covarianță* a variabilelor aleatoare  $u$  și  $v$ . Indicatorul cantitativ al corelației este *coeficientul de corelație*, care se calculează cu expresia:

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

Funcția Matlab `corrcoef` stabilește dacă între două seturi (observații) de date (variabile), înregistrate în vectorii notați cu  $u$  și  $v$ , există o *dependență liniară*, calculând *coeficientul de corelație* cu relația următoare:

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

Apelată cu sintaxa:

$$R_{xy} = \text{corrcoef}(X, Y),$$

funcția primește ca argumente vectorii datelor observate (în linie) și returnează coeficientul de corelație dintre liniile de date observate sub forma unei matrice de ordinul doi simetrice, ale cărei elemente reprezintă coeficientul de corelație în formatul următor:

Coeficientul de corelație între X și X	Coeficientul de corelație între X și Y
Coeficientul de corelație între Y și X	Coeficientul de corelație între Y și Y

Funcția poate fi apelată și cu sintaxa:

$$R_x = \text{corrcoef}(X),$$

care primește ca argument o matrice X ale cărei linii conțin *datele observate* (variabilele), iar coloanele corespund *caracteristicilor urmărite (monitorizate)* și returnează matricea coeficienților de corelație în formatul prezentat.

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

Matricea coeficienților de corelație având proprietatea de simetrie  $R(i,j)=R(j,i)$ , se poate găsi în una dintre situațiile următoare:

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

în care  $0 \dots 1$  reprezintă intervalul pentru o valoare posibilă a coeficientului de corelație între X și Y respectiv între Y și X.

#### *Interpretarea coeficientului de corelație*

Coeficientul de corelație este o mărime abstractă care nu depinde de unitățile în care sunt exprimate variabilele x și z. El exprimă *o măsură a dependenței 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 corelație definită printr-o dependență funcțională liniară, acest caz neexcluzând existența unei eventuale dependențe neliniare;
- dacă  $R=\pm 1$ , între caracteristicile considerate există o corelație de forma unei dependențe funcționale perfect liniare, în care odată cu creșterea lui x crește și y dacă valoarea este +1, sau cu creșterea lui x descrește y dacă valoarea este -1.
- Dacă  $R=0 \dots 1$  rezultă un anumit grad de dependență liniară, care se evidențiază în tendința globală de ordonare a datelor pe o anumită direcție.

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

*Rezolvare.* S-au ales doi vectori numerici cu dimensiunea 10, primul generat cu pas liniar și al doilea generat aleator cu distribuție uniformă. Se aplică funcția `corrcoef` și 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  $z$  și se testează din nou corelația celor două seturi de valori ( $x$  și  $z$ ). În acest caz se constată un coeficient de corelație apropiat de unu, fapt ilustrat și 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
```



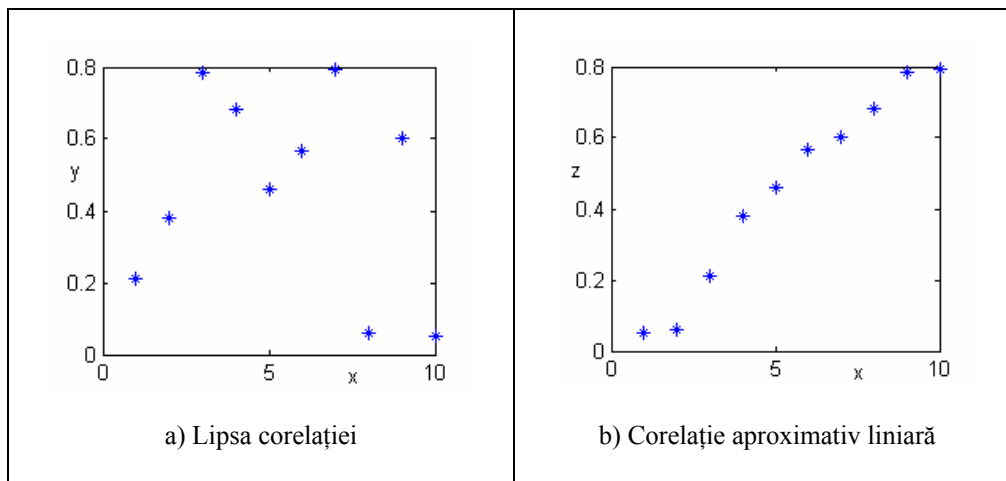


Fig. 5.5. Efecte de corelare a două caracteristici x și y