

Introducere in Matlab

- Un limbaj interpretat, de nivel înalt
- Folosit în mod special pentru calcul numeric
- O unealtă puternică pentru manipularea și vizualizarea datelor

Operații simple

In [1]:	5+7
	ans = 12
In [2]:	6/4
	ans = 1.5000
In [3]:	5-6
	ans = -1
In [4]:	2^4
	ans = 16
In [5]:	5*5
	ans = 25
In [6]:	sin(5)
	ans = -0.95892
In [7]:	cos(5)
	ans = 0.28366
In [8]:	tan(pi)
	ans = -1.2246e-016
In [9]:	pi
	ans = 3.1416
In [10]:	format long
In [11]:	pi
	ans = 3.14159265358979
In [12]:	format short
In [13]:	exp(1)
	ans = 2.7183
In [14]:	log(exp(1))
	ans = 1
In [15]:	log10(10)
	ans = 1
In [16]:	sinh(4)
	ans = 27.290
In [17]:	tanh(3)
	ans = 0.99505
In [18]:	x1 = sqrt(100^2 - 60^2)
	x1*15*(60^2 + x1^2)^(-1/2)
	x1 = 80
	ans = 12

Matrici și vectori

Vector linie

```
In [19]: x=[1 2]
x =
     1     2
```

Vector coloană

```
In [20]: y=[3;4]
y =
     3
     4
```

Transpusa

```
In [21]: y'
ans =
     3     4
```

```
In [22]: A=[1 2; 3 4]
A =
     1     2
     3     4
```

A*A și A^2 sunt la fel dar A.^2 este diferită! A*A sau A^2 reprezintă produse de matrici iar A.^2 reprezintă un produs element cu element.

```
In [23]: A*A
A^2
A.^2
ans =
     7    10
    15    22

ans =
     7    10
    15    22

ans =
     1     4
     9    16
```

```
In [24]: B = [5 6; 7 8]
B =
     5     6
     7     8
```

Inversa unei matrici

```
In [25]: C=B^(-1)
C =
   -4.0000    3.0000
    3.5000   -2.5000
```

```
In [26]: A*C
ans =
    3.00000   -2.00000
    2.00000   -1.00000
```

```
In [27]: B*C
ans =
    1.00000    0.00000
    0.00000    1.00000
```

Înmulțire element cu element

```
In [28]: B.*C
```

```
ans =  
  
   -20.000   18.000  
   24.500  -20.000
```

```
In [29]: inv(B)
```

```
ans =  
  
   -4.0000   3.0000  
   3.5000  -2.5000
```

Determinantul unei matrici

```
In [30]: det(B)
```

```
ans = -2.0000
```

```
In [31]: help det
```

```
'det' is a built-in function from the file libinterp/corefcn/det.cc  
  
-- det (A)  
-- [D, RCOND] = det (A)  
   Compute the determinant of A.  
  
   Return an estimate of the reciprocal condition number if requested.  
  
   Programming Notes: Routines from LAPACK are used for full matrices  
   and code from UMFPACK is used for sparse matrices.  
  
   The determinant should not be used to check a matrix for  
   singularity. For that, use any of the condition number functions:  
   'cond', 'condest', 'rcond'.  
  
   See also: cond, condest, rcond.  
  
Additional help for built-in functions and operators is  
available in the online version of the manual. Use the command  
'doc <topic>' to search the manual index.  
  
Help and information about Octave is also available on the WWW  
at http://www.octave.org and via the help@octave.org  
mailing list.
```

Funcții predefinite utilizate des

```
In [32]: size(A)
```

```
ans =  
  
    2    2
```

```
In [33]: A=zeros(3)
```

```
A =  
  
    0    0    0  
    0    0    0  
    0    0    0
```

```
In [34]: A = eye(4)
```

```
A =  
  
Diagonal Matrix  
  
    1    0    0    0  
    0    1    0    0  
    0    0    1    0  
    0    0    0    1
```

```
In [35]: A=rand(4)
```

```
A =  
  
    0.366980    0.099671    0.588264    0.678752  
    0.336123    0.459991    0.750595    0.028883  
    0.249130    0.654571    0.073662    0.025556  
    0.987796    0.300037    0.881864    0.188891
```

Exemplu: Rezolvarea sistemului de ecuații $Ax = b$.

```
In [36]: A=[1 2 3; 3 3 4; 2 3 3];
b=[1; 1; 2];
x=A\b

x =

-0.50000
 1.50000
-0.50000
```

Vectori

```
In [37]: x = 1:5

x =

     1     2     3     4     5
```

```
In [38]: x = 1:0.5:4

x =

    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000
```

```
In [39]: length(x)

ans = 7
```

```
In [40]: x = linspace(0,5,8)

x =

    0.00000    0.71429    1.42857    2.14286    2.85714    3.57143    4.28571    5.00000
```

```
In [41]: index = find(x>2)

index =

     4     5     6     7     8
```

```
In [42]: x(index)

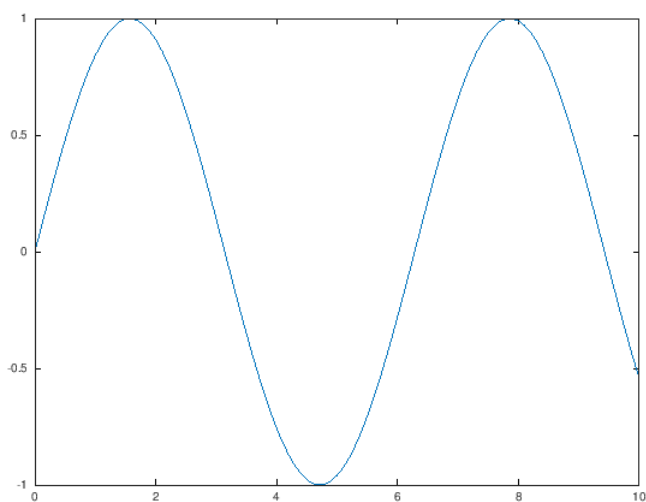
ans =

    2.1429    2.8571    3.5714    4.2857    5.0000
```

Execuția unui program matlab

De exemplu să presupunem că fișierul *plottest.m* conține un program matlab, execuția acestuia se face prin scrierea numelui fișierului fără extensia *.m* la linia de comandă

```
In [43]: plottest
```



Structura condițională *if-else*

```
In [44]: a = 3
b = 2

a = 3
b = 2
```

```
In [45]: if a < b
        disp('a mai mic ca b')
        else
            disp('a mai mare ca b')
        end

a mai mare ca b
```

Structuri repetitive

```
In [46]: for i = 1:5
        disp(i)
    end

1
2
3
4
5
```

```
In [47]: a = 0;
        while a <= 5
            disp(a)
            a = a + 1;
        end

0
1
2
3
4
5
```

Exemplu: Șirul Fibonacci

```
In [48]: n = 20;
        F(1) = 1;
        F(2) = 1;
        for i = 3:n
            F(i) = F(i-1) + F(i-2);
        end
        disp(F)

Columns 1 through 11:

         1         1         2         3         5         8        13        21        34        55        89

Columns 12 through 20:

        144        233        377        610        987       1597       2584       4181       6765
```

Funcții

Definirea unei funcții. Exemplu: $f(x) = 3x^2 + 2x + 1$

```
In [49]: function [f]=func(x)
        f = 3*x^2 + 2*x + 1;
    end
```

Apelarea functiei

```
In [50]: func(1.23)

ans = 7.9987
```

Funcții de mai multe variabile. Exemplu: $f_2(x, y, z) = x^2 + 2xy + z$

```
In [51]: function [f]=func2(x,y,z)
        f = x^2 + 2*x*y + z;
    end
```

```
In [52]: func2 = func2(1,2,3)

func2 = 8
```

Funcțiile pot returna mai multe valori:

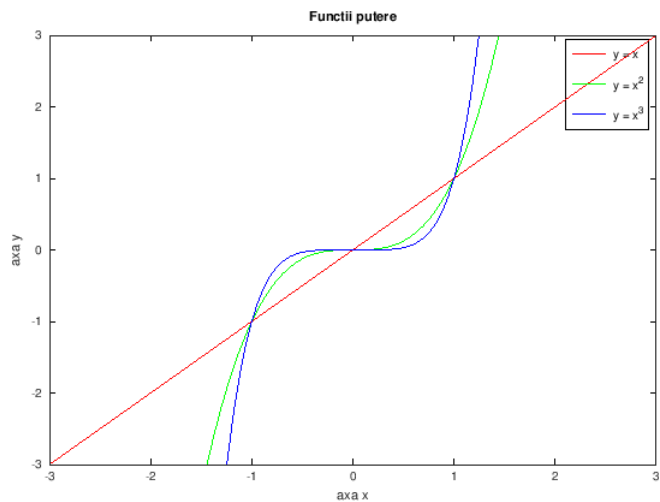
```
In [53]: function [x,y] = func3(a)
        x = 2*a;
        y = 3*a;
    end
```

```
In [54]: [x y] = func3(2);
x
y
x = 4
y = 6
```

Grafice

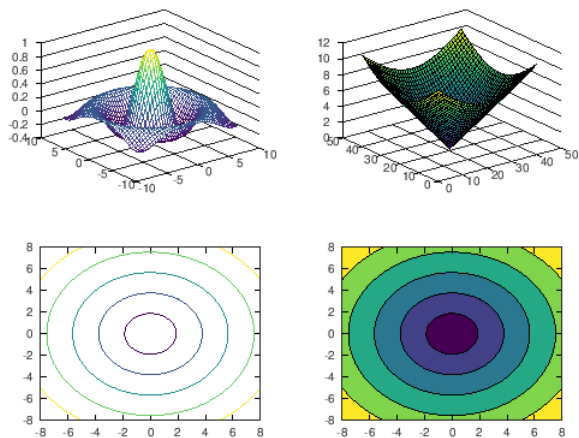
Grafice 2-dimensionale

```
In [55]: x = linspace(-3, 3, 100);
y1 = x;
y2 = x.^3;
y3 = x.^5;
plot(x, y1, 'r')
hold on
plot(x, y2, 'g')
hold on
plot(x, y3, 'b')
xlabel('axa x')
ylabel('axa y')
title('Funcții putere')
axis([-3, 3, -3, 3])
legend('y = x', 'y = x^2', 'y = x^3')
```



Grafice 3-dimensionale

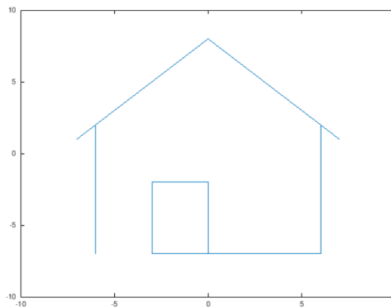
```
In [56]: tx = ty = linspace (-8, 8, 41);
[xx, yy] = meshgrid (tx, ty);
r = sqrt (xx.^2 + yy.^2);
tz = sin (r) ./ r;
subplot(2,2,1)
mesh (tx, ty, tz);
subplot(2,2,2)
surf(r)
subplot(2,2,3)
contour(xx,yy,r,5)
subplot(2,2,4)
contourf(xx,yy,r,5)
```



Metode numerice - Laborator 1

Operații cu matrici și vectori în Matlab

1. Se dă fișierul *lab1_data.txt* ce conține un tablou cu 2×11 elemente ce reprezintă coordonatele a 11 puncte 2D (prima linie conține componentele x ale punctelor iar a 2-a linie conține componentele y). Încărcați datele din fișier în Matlab folosind comanda `load p.txt`.
2. Afișați punctele pe un grafic folosind funcția `plot(p(1,:), p(2,:))`. Ar trebui să se afișeze următoarea figură:



3. Declarați și inițializați matricea A cu dimensiunea 2×2 și următoarele elemente:

$$A = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

4. Parcurgeți cu o buclă `for` fiecare coloană din tabloul p și înmulțiți-o la dreapta cu matricea A . Înregistrați rezultatele într-un alt tablou cu dimensiunea 2×11 numit p_new .
5. Afișați pe grafic punctele din tabloul p_new folosind funcția `plot` la fel ca la punctul 2.
6. Efectuați operația de la punctul 4 cu o singură linie de cod și fără a folosi o buclă `for`. Afișați din nou punctele pe grafic, punctele ar trebui să aibă aceeași poziție ca cele de la punctul 5.
7. Scrieți o funcție Matlab numită *rotation_matrix* ce primește ca parametru un scalar numit θ și returnează următoarea matrice:

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

8. Declarați și inițializați vectorul $\theta = [0, \frac{\pi}{4}, \frac{\pi}{3}, \frac{\pi}{2}, \pi]$. Pentru fiecare element din θ :
 - calculați matricea R folosind funcția de la punctul 7
 - calculați p_new folosind matricea R obținută mai sus
 - afișați pe grafic p_new
9. Scrieți o funcție Matlab numită *test_inv* care ia ca parametru un număr întreg n și face următoarele:

- Crează o matrice A de $n \times n$ elemente aleatoare
- Calculează matricea inversă $B = A^{-1}$
- Calculează matricea $C = AB$
- Calculează eroarea operației de inversiune în felul următor $e = \sum_{i,j=0}^n |C_{i,j} - I_{i,j}|$, reprezentând suma modulelor diferențelor element cu element între matricea $C = AB$ și matricea unitate I .
- Funcția va returna eroarea e

10. Apelați funcția `test_inv` pentru $n = 10, 11, 12, \dots, 300$ și înregistrați erorile e într-un vector.

11. Afișați erorile e pe un grafic unde pe axa x vor fi reprezentate valorile n pentru care a fost apelată funcția `test_inv` iar pe axa y vor fi erorile e . Ambele axe ale graficului vor fi reprezentate la scară logaritmică: folosiți funcția `loglog()` în loc de `plot()`.

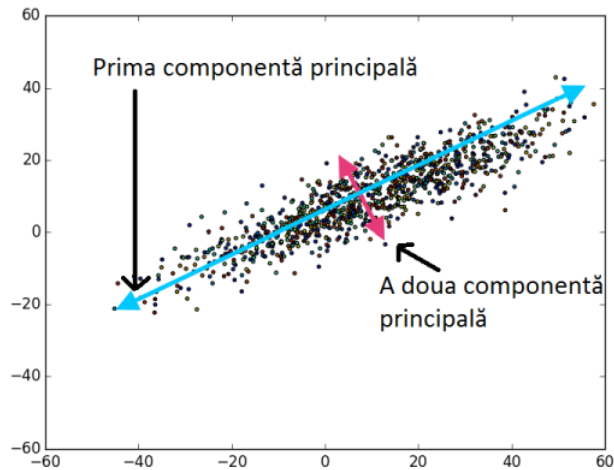
Analiza Componentelor Principale

Analiza componentelor principale (ACP) este o metodă des folosită și simplă de transformare liniară folosită în numeroase aplicații cum ar fi tranzacționarea de acțiuni, predicția evoluției piețelor și multe altele. Principalul scop al ACP este reducerea numărului de dimensiuni al unui set de date multidimensional fără a pierde prea multe informații. Acest lucru este realizat prin găsirea direcțiilor în care datele au varianța maximă și proiectarea datelor pe aceste direcții.

Scurt rezumat al metodei ACP

1. Calculul matricei de covarianță a datelor
2. Calculul vectorilor și valorilor proprii ai matricii de covarianță
3. Sortarea vectorilor proprii după valoarea valorilor proprii corespunzătoare și alegerea celor mai relevanți k vectori proprii
4. Construcția matricii de proiecție \mathbf{W} din cei k vectori proprii selectați
5. Transformarea datelor originale prin matricea de proiecție într-un spațiu cu k dimensiuni

Exemplu 2-Dimensional



Calculul matricei de covarianță

Fie \mathbf{X} o matrice cu dimensiunea $n \times d$ ce reprezintă n puncte d -dimensionale. Scopul principal al metodei este reprezentarea punctelor într-un spațiu k -dimensional unde $k \leq d$, spațiu în care se capturează cât mai multe informații din setul de date original. Abordarea clasică pentru ACP constă în calculul vectorilor și valorilor proprii pentru matricea de covarianță S , aceasta este o matrice $d \times d$ în care fiecare element reprezintă covarianța între două variabile. Covarianța între două variabile se calculează în felul următor:

$\sigma_{jk} = \frac{1}{n-1} \sum_{i=1}^N (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$, Unde \bar{x} reprezintă valoarea medie a variabile x . Calculul matricei de covarianță pentru întregul set de date \mathbf{X} se face în felul următor:

$$S = \frac{1}{n-1} ((\mathbf{X} - \bar{\mathbf{x}})^T (\mathbf{X} - \bar{\mathbf{x}}))$$

Matricea $\bar{\mathbf{X}}$ are dimensiunea $1 \times d$ și conține valorile medii \bar{x} pentru fiecare variabilă x_i : $\bar{\mathbf{X}} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_d]$

Calculul vectorilor și valorilor proprii pentru S și alegerea celor mai relevanți k

Odată ce se cunoaște matricea de covarianță S este nevoie de calculul vectorilor și valorilor proprii ai acestei matrici: \mathbf{v}_i și λ_i unde $i = 1, 2, \dots, d$. Mai departe este nevoie să se aleagă cei mai relevanți k vectori proprii, aceștia se vor alege în funcție de valorile proprii λ_i , vectori proprii cu valori proprii mari sunt cei mai relevanți iar cei cu valori proprii mici sunt cei mai puțin relevanți. Pentru a alege cei mai relevanți k vectori aceștia se sortează după valoarea proprie ce-i corespunde fiecăruia și se vor selecta primii k vectori.

Construcția matricei de proiecție \mathbf{V} din cei k vectori proprii selectați

Presupunem că au fost aleși k vectori proprii \mathbf{v}_i , din aceștia se poate construi matricea de proiecție \mathbf{V} în care fiecare coloană reprezintă un vector propriu: $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$. Dimensiunea matricii \mathbf{V} va fi $n \times k$.

Transformarea datelor originale prin matricea de proiecție într-un spațiu cu k dimensiuni

Pornind de la matricea inițială \mathbf{X} și matricea de proiecție \mathbf{V} este nevoie să se calculeze matricea \mathbf{X}' , ce reprezintă punctele reprezentate în spațiul dimensional redus. Cu alte cuvinte, matricea \mathbf{V} poate fi văzută ca o bază k -dimensională iar scopul acestui pas este de a reprezenta fiecare punct \mathbf{x}_i în baza \mathbf{V} . fiecare punct \mathbf{x}_i va fi transformat în \mathbf{x}'_i felul următor:

$\mathbf{x}'_i = [\mathbf{v}_1 \mathbf{x}_i, \mathbf{v}_2 \mathbf{x}_i, \dots, \mathbf{v}_k \mathbf{x}_i]$ Unde $\mathbf{v} \mathbf{x}$ reprezintă produsul scalar între vectorii \mathbf{v} și \mathbf{x} . Prin urmare Matricea \mathbf{X}' ce conține toți vectorii \mathbf{x}'_i se calculează în felul următor:

$$\mathbf{X}' = \mathbf{X} \times \mathbf{V}$$

Cum matricei \mathbf{X} este $n \times d$ iar dimensiunea matricii \mathbf{V} este $n \times k$ rezultă că dimensiunea matricii \mathbf{X}' va fi $n \times k$, prin urmare s-a redus dimensiunea datelor de la d la k .

Exemplu pe un set de date real

Descrierea datelor:

Setul de date conține măsurători pentru 4898 de tipuri de vin împreună cu calitatea vinului determinată prin degustare. Mai exact setul de date este dat ca o matrice cu 4898 linii și 12 coloane, fiecare linie corespunde unui tip de vin iar fiecare coloană este o proprietate a vinului:

- 1. Aciditatea fixă
- 2. Aciditatea volatilă
- 3. Cantitatea de acid citric
- 4. Zahăr rezidual
- 5. Cloruri
- 6. Dioxid de sulf liber
- 7. Dioxid de sulf total
- 8. Densitate
- 9. pH
- 10. Sulfai
- 11. Alcool
- 12. Calitate (scor între 1 și 10)

Încărcarea setului de date

```
In [1]: X = dlmread('winequality-white.csv', ';');

In [2]: X(1:6,:)

ans =

Columns 1 through 6:

7.0000e+000 2.7000e-001 3.6000e-001 2.0700e+001 4.5000e-002 4.5000e+001
6.3000e+000 3.0000e-001 3.4000e-001 1.6000e+000 4.9000e-002 1.4000e+001
8.1000e+000 2.8000e-001 4.0000e-001 6.9000e+000 5.0000e-002 3.0000e+001
7.2000e+000 2.3000e-001 3.2000e-001 8.5000e+000 5.8000e-002 4.7000e+001
7.2000e+000 2.3000e-001 3.2000e-001 8.5000e+000 5.8000e-002 4.7000e+001
8.1000e+000 2.8000e-001 4.0000e-001 6.9000e+000 5.0000e-002 3.0000e+001

Columns 7 through 12:

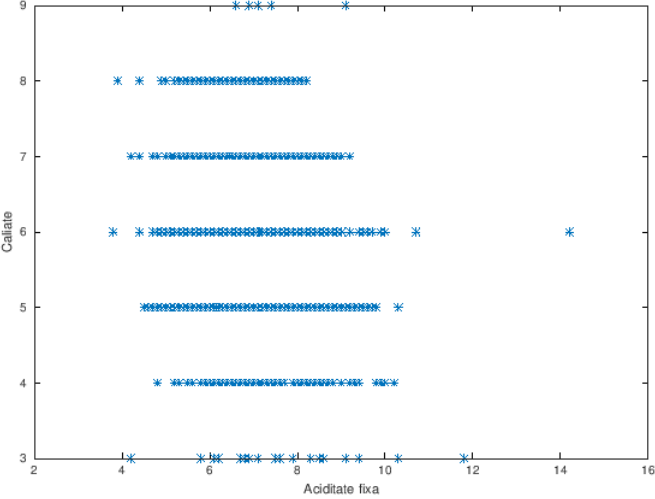
1.7000e+002 1.0010e+000 3.0000e+000 4.5000e-001 8.8000e+000 6.0000e+000
1.3200e+002 9.9400e-001 3.3000e+000 4.9000e-001 9.5000e+000 6.0000e+000
9.7000e+001 9.9510e-001 3.2600e+000 4.4000e-001 1.0100e+001 6.0000e+000
1.8600e+002 9.9560e-001 3.1900e+000 4.0000e-001 9.9000e+000 6.0000e+000
1.8600e+002 9.9560e-001 3.1900e+000 4.0000e-001 9.9000e+000 6.0000e+000
9.7000e+001 9.9510e-001 3.2600e+000 4.4000e-001 1.0100e+001 6.0000e+000
```

Vizualizarea datelor

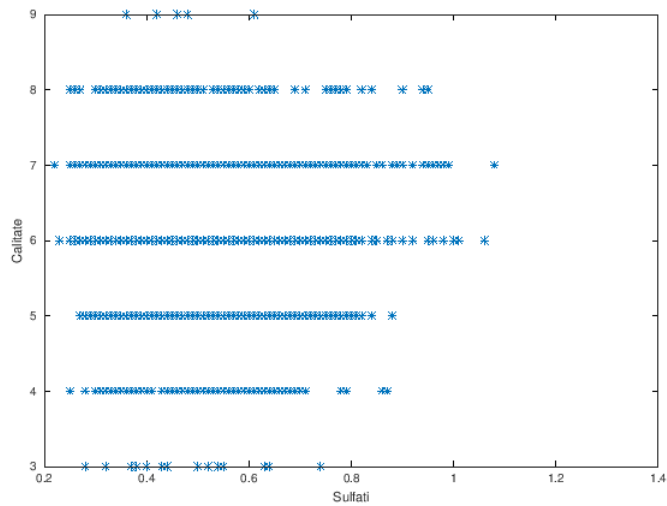
Datele pot fi separate în două tablouri X și Y cu dimensiunile 4898x11 respectiv 4898x1. Fiecare linie din X reprezintă un tip de vin iar fiecare coloană reprezintă o proprietate. Matricea coloană Y reprezintă calitatea vinului pentru fiecare tip. Mărimea de ieșire **Y** poate fi vizualizată pe grafic în funcție de fiecare proprietate din tabloul mărimilor de intrare **X**. Se observă că nu există o corelație clară între fiecare proprietate **X_i** și **Y**.

```
In [3]: X_ = X(:,1:11);
        Y_ = X(:,12);

In [4]: plot(X_(:,1), Y_, '*')
        xlabel('Aciditate fixa')
        ylabel('Calitate')
```



```
In [5]: plot(X(:,10), Y_, '*')
xlabel('Sulfati')
ylabel('Calitate')
```



Calculul matricei de covarianță S

```
In [6]: n = size(X)(1);
S = (X - mean(X))'*(X - mean(X))/(n-1);
```

Sau folosind funcția predefinită `cov`:

```
In [7]: S = cov(X);
```

Calculul vectorilor și valorilor proprii pentru S

In [8]: [V L] = eig(S)

V =

Columns 1 through 6:

-7.6660e-004	2.2474e-003	8.5782e-004	4.9697e-003	-4.6856e-002	7.4339e-002
-4.6970e-004	-2.4841e-002	9.4122e-001	-1.6243e-001	-2.8322e-001	-7.1912e-002
-3.3712e-004	-2.2003e-002	3.2148e-001	3.5207e-001	8.7680e-001	-4.0301e-002
-3.7369e-004	5.9405e-004	-2.4097e-003	1.3986e-004	3.2294e-004	6.2949e-003
-4.6169e-003	9.9934e-001	2.9476e-002	-1.5098e-003	1.4276e-002	-1.3743e-002
6.4853e-006	-1.0368e-005	1.1675e-003	-4.9434e-004	-7.8344e-004	6.3092e-004
-3.8128e-006	-2.8197e-005	-6.9332e-004	3.5524e-004	-5.9599e-005	-7.4732e-004
9.9998e-001	4.6294e-003	1.0105e-003	-1.3031e-004	3.2734e-004	3.6915e-003
-3.4818e-003	1.2204e-002	9.1602e-002	3.1221e-001	-1.1199e-001	9.3563e-001
-1.4465e-003	-1.5988e-003	-1.2895e-002	-8.6727e-001	3.6840e-001	3.3438e-001
1.0957e-003	6.8114e-003	-2.3868e-002	1.9285e-003	8.8510e-003	-2.7773e-004
8.3474e-005	6.2459e-004	2.7437e-002	4.7623e-005	-1.6677e-002	-1.4862e-002

Columns 7 through 12:

-8.4924e-002	9.8453e-001	1.2442e-001	-1.2924e-002	9.1667e-003	-1.5445e-003
4.0954e-002	-3.8497e-003	-5.0464e-003	-9.3440e-004	1.5462e-003	-1.6903e-004
-8.2749e-003	4.1664e-002	2.9386e-003	-1.2579e-003	-1.4037e-004	-3.3865e-004
3.7823e-002	-5.9492e-004	-7.5976e-002	-9.9513e-001	-1.4931e-002	-4.7328e-002
-2.1355e-003	-1.5129e-003	5.8641e-003	-7.9998e-005	7.2039e-005	-9.7579e-005
4.9068e-003	7.8863e-003	1.0835e-002	2.6284e-002	-9.6464e-001	-2.6187e-001
1.0440e-003	-1.7392e-003	-1.1977e-002	4.2851e-002	2.6268e-001	-9.6385e-001
-5.7720e-004	3.2526e-004	9.7756e-004	-4.4709e-004	1.8398e-005	-3.5971e-005
-1.7499e-004	-7.5481e-002	-1.6657e-002	7.0225e-003	4.0806e-005	-3.3620e-006
-1.0426e-002	-3.5613e-003	-5.0452e-003	2.1455e-003	3.6053e-004	-3.4089e-004
5.3579e-001	1.5219e-001	-8.2583e-001	8.2889e-002	-6.4797e-003	1.2504e-002
-8.3809e-001	-2.9863e-003	-5.4417e-001	9.5370e-003	-1.0993e-002	3.2804e-003

L =

Diagonal Matrix

Columns 1 through 6:

3.1286e-007	0	0	0	0	0
0	3.9584e-004	0	0	0	0
0	0	8.2065e-003	0	0	0
0	0	0	1.1446e-002	0	0
0	0	0	0	1.4116e-002	0
0	0	0	0	0	1.8389e-002
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Columns 7 through 12:

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
5.0041e-001	0	0	0	0	0
0	6.8671e-001	0	0	0	0
0	0	1.3162e+000	0	0	0
0	0	0	2.1563e+001	0	0
0	0	0	0	1.6847e+002	0
0	0	0	0	0	1.9315e+003

Selectarea celor mai relevanti *k* vectori

Presupunem că dorim să reducem dimensionalitatea datelor la 3, prin urmare selectăm trei doi vectori proprii ce au valorile proprii maxime, în acest caz λ_{12} , λ_{11} și λ_{10}

In [9]: v1 = V(:,12);
v2 = V(:,11);
v3 = V(:,10);

Matricea de proiecție *V*

In [10]: V_p = V(:,[12 11 10]);
size(V_p)

ans =

12 3

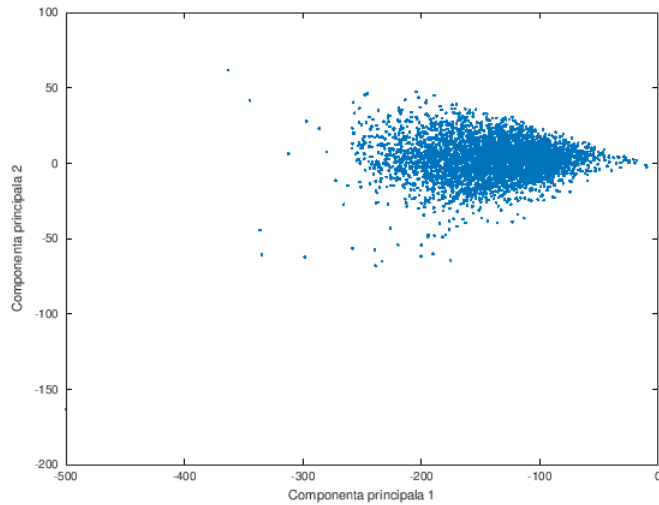
Reducerea dimensiunii datelor de la 12 la 2 folosind matricea *V*

In [11]: X_reduced = X*V_p;

```
In [12]: size(X_reduced)
```

```
ans =  
  
    4898      3
```

```
In [13]: plot(X_reduced(:,1), X_reduced(:,2),'.')  
xlabel('Componenta principala 1')  
ylabel('Componenta principala 2')
```



Reconstuctia datelor originale \mathbf{X} din datele reduse \mathbf{X}'

Știind ca $\mathbf{X}' = \mathbf{X}\mathbf{V}$, înmulțind la dreapta cu \mathbf{V}^{-1} se obține:

$$\mathbf{X} = \mathbf{X}'\mathbf{V}^{-1}$$

Cum \mathbf{V} este matricea ce conține vectori proprii, iar aceștia sunt ortonormali, rezultă că $\mathbf{V}^{-1} = \mathbf{V}^T$, prin urmare:

$$\mathbf{X}_r = \mathbf{X}'\mathbf{V}^T$$

```
In [14]: X_reconstructed = X_reduced*V_p';
```

Calculul erorii de reconstrucție

Eroarea de reconstrucție este o cantitate ce poate fi folosită pentru a determina cantitatea de informații pierdută prin reducerea dimensionalității, această eroare poate fi exprimată în procente folosind diferența relativă între datele originale \mathbf{X} și cele reconstruite \mathbf{X}_r ,

$$e = \frac{\sum_{i,j} |\mathbf{X}_r(i,j) - \mathbf{X}(i,j)|}{\sum_{i,j} |\mathbf{X}(i,j)|} \cdot 100\%$$

Cu cât numărul de componente principale ce au fost luate în considerare este mai mare, cu atât eroarea de reconstrucție este mai mică. Dacă toate componentele principale au fost luate în considerare atunci eroarea de reconstrucție este zero.

```
In [15]: e = sum(sum(abs(X_reconstructed - X)))/sum(sum(abs(X)))*100  
e = 15.142
```

Pentru trei componente principale selectate au fost pierdute 15% din informații.

Metode numerice - Laborator 2

Vectori și valori proprii, analiza componentelor principale

1. Se dă fișierul *lab2_data.txt* ce conține un tablou cu dimensiunea 500×15 reprezentând coordonatele a 500 de puncte 15-dimensionale. Încărcați fișierul folosind funcția *dlmread*:

```
X = dlmread('lab2_data.txt',';');
```

2. Calculați matricea de covarianță *S* a matricei *X* folosind definiția și apoi folosind funcția predefinită *cov*. Cele două rezultate trebuie să fie identice.
3. Calculați vectorii și valorile proprii ale matricei *S* utilizând funcția predefinită *eig*:

```
[V L] = eig(S);
```

4. Alegeți cei mai importanți *k* vectori proprii ce corespund valorilor proprii maxime și construiți matricea de proiecție *V*.
5. Utilizați matricea de proiecție *V* pentru a calcula matricea punctelor cu dimensiunea redusă *X'*.
6. Reconstruiți punctele originale *X*, utilizând matricea de proiecție *V* și matricea punctelor cu dimensiune redusă *X'*.
7. Calculați eroarea de reconstrucție *e*.
8. Repetați procesul pentru diferite valori ale lui *k*, deduceți care este numărul optim de vectori proprii necesari pentru a reduce dimensiunea punctelor.

3. REZOLVAREA ECUAȚIILOR NELINIARE

Enunțul problemei

Fie ecuația neliniară:

$$f(x) = 0;$$

pentru care se cunoaște că admite o rădăcină în intervalul $[a, b]$. Se cere determinarea valorii rădăcinii $\alpha \in [a, b]$ astfel încât $f(\alpha) < \epsilon$, unde ϵ reprezintă precizia cu care se dorește determinarea soluției.

Determinarea intervalului în care se găsește soluția se poate face fie analitic, dacă se cunoaște derivata funcției $f(x)$, utilizând șirul lui Rolle, fie pe baza unor considerente de natura tehnică.

Toate metodele de rezolvare constau în restrângerea intervalului inițial $[a, b]$ în jurul rădăcinii căutate.

În cazul tuturor metodelor se verifică în prealabil dacă în intervalul dat se afla o rădăcină a ecuației: Dacă $f(a) \cdot f(b) \leq 0$ în interval se află o rădăcină, în caz contrar nu și ca urmare procesul de căutare a soluției nu mai trebuie efectuat.

3.1. METODA ÎNJUMĂTĂȚIRII INTERVALULUI

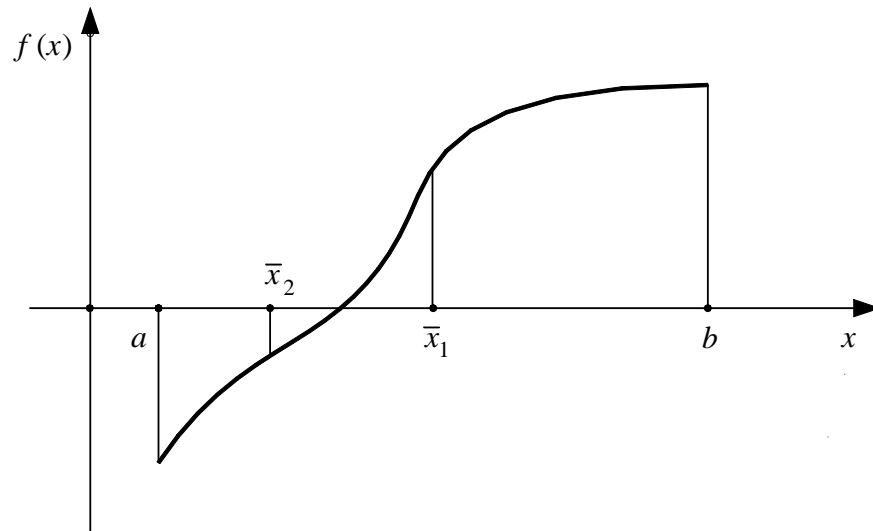


Fig. 3.1. Explicativă la metoda înjumătățirii intervalului.

Metoda constă în restrângerea intervalului inițial $[a, b]$ prin înjumătățiri succesive, de fiecare dată stabilind în care subinterval se găsește rădăcina.

Etapa 1

- Se determină mijlocul intervalului inițial $[a, b]$

$$\bar{x}_1 = \frac{a+b}{2}$$

- Se verifică în ce subinterval se găsește rădăcina, testând valoarea produsului $f(a) \times f(\overline{x_1})$

$$f(a) \times f(\overline{x_1}) \begin{cases} \leq 0 & \text{P } [a, \overline{x_1}] \\ > 0 & \text{P } [\overline{x_1}, b] \end{cases}$$

- Se verifică dacă s-a obținut rădăcina.

$$f(\overline{x_1}) \begin{cases} \text{e } 0 & \text{P } \text{rădăcina este } a = \overline{x_1} \text{ și algoritmul de căutare se oprește;} \\ \text{nu e } 0 & \text{P } \text{rădăcina nu a fost determinată și ca urmare se va mai parcurge o etapă.} \end{cases}$$

La etapele următoare se efectuează același tip de calcule dar pentru intervalul restrâns la etapa precedentă.

3.2. METODA SECANTEI FIXE

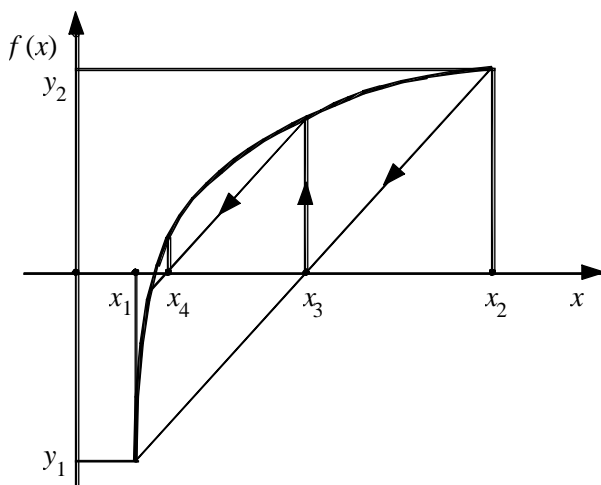


Fig. 3.2. Explicativă la metoda secantei fixe.

Această metodă utilizează, pentru apropierea de soluție, intersecția dintre axa absciselor și secanta la graficul funcției $f(x)$ care unește punctele de pe grafic determinate de extremitățile intervalului de căutare specificat inițial (vezi figura 3.2).

Panta secantei se va calcula cu relația:

$$m = \frac{y_2 - y_1}{x_2 - x_1};$$

iar ecuația secantei se poate scrie sub forma:

$$y - y_2 = m(x - x_2).$$

Abscisa punctului de intersecție al secantei cu axa orizontală este:

$$x_3 = x_2 - \frac{y_2}{m},$$

pentru această valoare se recalculează funcția obținându-se valoarea $y_3 = f(x_3)$.

În punctul de coordonate (x_3, y_3) se duce o paralelă la secanta inițială, o dreaptă de pantă m , care va fi utilizată pentru continuarea procesului de căutare. Ca urmare, valorile succesive care se vor obține pentru determinarea soluției se pot scrie prin formula de recurență:

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{m}. \quad (3.1)$$

Drept criteriu de oprire al algoritmului de căutare se poate utiliza relația:

$$|x_k - x_{k-1}| < \epsilon; \quad (3.2)$$

prin care se verifică dacă diferența dintre două determinări ale soluției este mai mică decât o valoare ϵ impusă inițial, sau relația:

$$|f(x_k)| < \epsilon; \quad (3.3)$$

prin care se verifică dacă valoarea funcției la determinarea în curs a devenit mai mică decât o valoare ϵ impusă inițial.

Pentru a face o apreciere asupra convergenței metodei se va utiliza dezvoltarea în serie Taylor a funcției. Presupunând că valoarea exactă a soluției este a și prin urmare $f(a)=0$, atunci două determinări succesive ale soluției se vor putea scrie sub forma:

$$x_k = a + e_k; \quad (3.4)$$

$$x_{k+1} = a + e_{k+1}; \quad (3.5)$$

în care e_k și e_{k+1} vor reprezenta erorile produse la cele două etape succesive.

Înlocuind aceste valori în relația (3.1) de recurență a metodei se obține:

$$a + e_{k+1} = a + e_k - \frac{f(a + e_k)}{m}. \quad (3.6)$$

Prin dezvoltarea în serie Taylor a funcției $f(x)$ în jurul punctului a și păstrând numai primii doi termeni ai dezvoltării, se obține egalitatea:

$$e_{k+1} = e_k - \frac{f(a) + e_k \times f'(a)}{m}; \quad (3.7)$$

sau:

$$e_{k+1} = e_k \times \left(1 - \frac{f'(a)}{m} \right) \quad (3.8)$$

Prin această relație poate fi exprimată eroarea care se produce la o etapă oarecare față de eroarea produsă la etapa precedentă. Procesul va fi convergent dacă paranteza are o valoare subunitară, deci dacă:

$$\left| 1 - \frac{f'(a)}{m} \right| < 1. \quad (3.9)$$

Această condiție poate fi îndeplinită pentru valori mari ale pantei inițiale ale secantei.

3.3. METODA SECANTEI VARIABLE

La această metodă, la fiecare etapă, se recalculează panta secantei. În acest scop punctele care reprezintă extremitățile secantei se aleg astfel: unul dintre ele, de exemplu cel din stânga, este fix alegându-se limita din stânga a intervalului de căutare specificat inițial; iar al doilea variabil determinat la fiecare etapă (vezi figura 3.3).

Etapa 1.

Se calculează panta secantei determinate de intervalul inițial, respectiv punctele având coordonatele (x_0, y_0) și (x_1, y_1) :

$$m_1 = \frac{y_1 - y_0}{x_1 - x_0}; \quad (3.10)$$

ca urmare ecuația secantei va fi:

$$y - y_1 = m_1(x - x_1). \quad (3.11)$$

Abscisa punctului de intersecție a secantei cu axa orizontală se va obține cu relația:

$$x_2 = x_1 - \frac{y_1}{m} = x_1 - \frac{f(x_1)}{m}; \quad (3.12)$$

pentru această abscisă valoarea funcției va fi $y_2 = f(x_2)$.

Etapa 2.

Se recalculează panta secantei determinate de punctele având coordonatele (x_0, y_0) și (x_2, y_2) :

$$m_2 = \frac{y_2 - y_0}{x_2 - x_0};$$

ca urmare ecuația secantei va fi:

$$y - y_2 = m_2 (x - x_2).$$

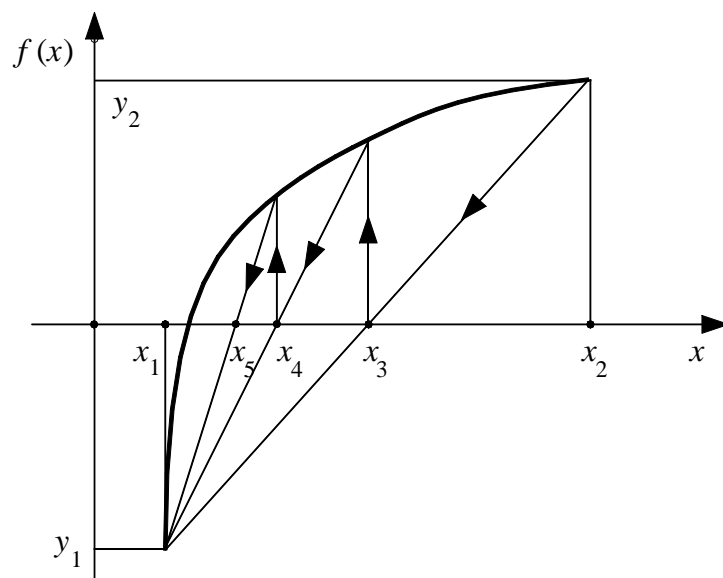


Fig. 3.3. Explicativă la metoda secantei variabile.

Abscisa punctului de intersecție a secantei cu axa orizontală se va obține cu relația:

$$x_3 = x_2 - \frac{y_2}{m} = x_2 - \frac{f(x_2)}{m},$$

pentru această abscisă valoarea funcției va fi $y_3 = f(x_3)$.

Ca urmare se pot stabili următoarele formule de recurență valabile pentru *etapa k*:

- Panta secantei la *etapa k*:

$$m_k = \frac{y_k - y_0}{x_k - x_0}. \quad (3.13)$$

- Coordonatele punctului care vor determina secanta la etapa următoare:

$$x_{k+1} = x_k - \frac{f(x_k)}{m}, \quad (3.14)$$

$$y_{k+1} = f(x_{k+1}). \quad (3.15)$$

Drept criteriu de oprire al algoritmului de căutare se poate utiliza relația:

$$|x_{k+1} - x_k| < \epsilon,$$

prin care se verifică dacă diferența dintre două determinări succesive ale soluției este mai mică decât o valoare ϵ impusă inițial, sau relația:

$$|f(x_{k+1})| < \epsilon,$$

prin care se verifică dacă valoarea funcției la determinarea în curs a devenit mai mică decât o valoare ϵ impusă inițial.

3.4. METODA TANGENTEI

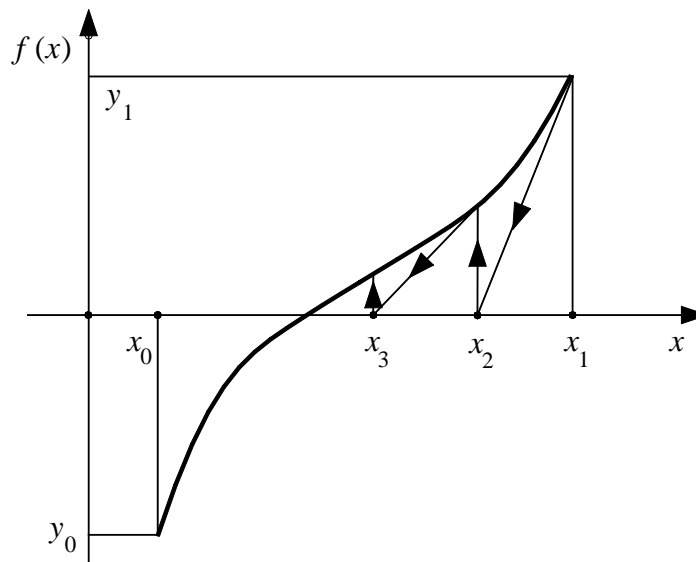


Fig. 3.4. Explicativă la metoda tangentei.

Această metodă utilizează, pentru apropierea de soluție, intersecția dintre axa absciselor și tangenta la graficul funcției $f(x)$ dusă în una dintre extremitățile intervalului de căutare, care la prima etapă, este cel specificat inițial (vezi figura 3.4). Ca urmare aplicarea metodei necesită cunoașterea atât a expresiei funcției $f(x)$ cât și a expresiei derivatei acesteia $f'(x)$.

Etapă 1.

Ecuția tangentei în punctul de coordonate (x_1, y_1) va avea expresia:

$$y - y_1 = f'(x_1)(x - x_1). \quad (3.16)$$

Abscisa punctului de intersecție al tangentei cu axa orizontală se va obține cu relația:

$$x_2 = x_1 - \frac{y_1}{f'(x_1)} = x_1 - \frac{f(x_1)}{f'(x_1)}; \quad (3.17)$$

pentru această abscisă valoarea funcției va fi $f(x_2)$ iar a derivatei $f'(x_2)$.

Etapa 2.

Ecuția tangentei în punctul de coordonate (x_2, y_2) va avea expresia:

$$y - y_2 = f'(x_2)(x - x_2).$$

Abscisa punctului de intersecție al tangentei cu axa orizontală se va obține cu relația:

$$x_3 = x_2 - \frac{y_2}{f'(x_2)} = x_2 - \frac{f(x_2)}{f'(x_2)};$$

pentru această abscisă valoarea funcției va fi $f(x_3)$ iar a derivatei $f'(x_3)$.

Ca urmare, se pot stabili următoarele formule de recurență valabile pentru *etapa k*:

Ecuția tangentei în punctul de coordonate (x_k, y_k) va avea expresia:

$$y - y_k = f'(x_k)(x - x_k). \quad (3.18)$$

Abscisa punctului de intersecție al tangentei cu axa orizontală se va obține cu relația:

$$x_{k+1} = x_k - \frac{y_k}{f'(x_k)} = x_k - \frac{f(x_k)}{f'(x_k)}; \quad (3.19)$$

pentru această abscisă valoarea funcției va fi $f(x_{k+1})$ iar a derivatei $f'(x_{k+1})$.

Drept criteriu de oprire al algoritmului de căutare se poate utiliza relația:

$$|x_{k+1} - x_k| < \epsilon$$

prin care se verifică dacă diferența dintre două determinări succesive ale soluției este mai mică decât o valoare ϵ impusă inițial, sau relația:

$$|f(x_{k+1})| < \epsilon$$

prin care se verifică dacă valoarea funcției la determinarea în curs a devenit mai mică decât o valoare ϵ impusă inițial.

Pentru a face o apreciere asupra convergenței metodei se va utiliza dezvoltarea în serie Taylor a funcției. Presupunând că valoarea exactă a soluției este a și prin urmare $f(a)=0$, atunci două determinări succesive ale soluției se vor putea scrie sub forma:

$$x_k = a + e_k ;$$

$$x_{k+1} = a + e_{k+1}$$

în care e_k și e_{k+1} vor reprezenta erorile produse la cele două etape succesive.

Înlocuind aceste valori în relația (3.19) de recurență a metodei se obține:

$$a + e_{k+1} = a + e_k - \frac{f(a + e_k)}{f'(a + e_k)}. \quad (3.20)$$

Prin dezvoltarea în serie Taylor a funcției $f(x)$ în jurul punctului a și păstrând numai primii doi termeni ai dezvoltării, se obține egalitatea:

$$e_{k+1} = e_k - \frac{f(a) + e_k \times f'(a)}{f'(a + e_k)}; \quad (3.21)$$

sau:

$$e_{k+1} = e_k \times \frac{f'(a)}{f'(a + e_k)} - \frac{f'(a)}{f'(a + e_k)} \times \frac{f'(a + e_k) - f'(a)}{f'(a + e_k)} = e_k \times \frac{f'(a) + e_k \times f''(a) - f'(a)}{f'(a + e_k)}$$

Prin această relație poate fi exprimată eroarea care se produce la o etapă oarecare față de eroarea produsă la etapa precedentă.

$$e_{k+1} = e_k^2 \times \frac{f''(a)}{f'(a + e_k)}. \quad (3.22)$$

Deoarece $e_{k+1} \propto e_k^2$ rezultă o convergență mult mai bună față de metoda secantei.

Rezolvarea sistemelor de ecuații liniare

1. Metoda de eliminare Gauss

Se dă sistemul de n ecuatii liniare cu n necunoscute scris sub formă matriceală $A\mathbf{x} = \mathbf{b}$ unde:

- A este matricea coeficienților și are dimensiunea $n \times n$
- \mathbf{x} este matricea coloană a necunoscutelor și are dimensiunea $n \times 1$
- \mathbf{b} este matricea coloană a vectorilor liberi are dimensiunea $n \times 1$

Metoda de eliminare Gauss constă în modificarea formei sistemului de ecuații astfel încât matricea coeficienților \mathbf{A} să fie o matrice triunghiulară (toate elementele de sub diagonală principală să fie zero). Aducerea sistemului de ecuații la o astfel de formă se face prin aplicarea succesivă unor operații elementare, mai exact:

- Permutarea liniei i cu linia j a matricei
- Înmulțirea liniei i cu o constantă c diferită de zero
- Adunarea liniei i cu linia j amplificată cu o constantă c

Efectuarea unei astfel de operații asupra unei matrici constă în înmulțirea la stânga cu o matrice elementară $E_{ij}(c)$.

Algoritmul de eliminare Gauss constă în doi pași:

- Aducerea sistemului de ecuații $A\mathbf{x} = \mathbf{b}$ la forma triunghiulară.
- Calculul soluțiilor \mathbf{x} prin înlocuire

Aducerea sistemului de ecuații la forma triunghiulară

Aducerea sistemului de ecuații la forma triunghiulară se face prin înmulțirea la stânga a sistemului cu o secvență de matrici elementare $E_1 E_2 \dots E_n$. Presupunem că dorim să aplicăm o operație elementară matricii sistemului A astfel încât elementul de pe linia i și coloana j să devină zero, acest lucru se poate face adunând la linia i a matricei o altă linie $i' \neq i$ amplificată cu o constantă $c = -\frac{A_{ij}}{A_{i'j}}$. Cu alte cuvinte acest lucru se poate face înmulțind la stânga ambii membri ai sistemului cu matricea elementară $E_{i'j}(-\frac{A_{ij}}{A_{i'j}})$. De obicei $i' = j$ dar acesta se poate alege diferit astfel încât $A_{i'j} \neq 0$ și să aibă o valoare absolută cât mai mare.

Fie sistemul de 2 ecuații și două necunoscute:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

Înmulțind ambii membri la stânga cu matricea $E_{1,2}(-\frac{a_{21}}{a_{11}})$ se obține:

$$\begin{bmatrix} 1 & 0 \\ -\frac{a_{21}}{a_{11}} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{a_{21}}{a_{11}} & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

De unde rezultă:

$$\begin{bmatrix} a_{11} & a_{12} \\ 0 & -a_{22} + \frac{a_{12}a_{21}}{a_{11}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - b_1 \frac{a_{21}}{a_{11}} \end{bmatrix}$$

Se observă că astfel s-a obținut zero sub diagonală principală, prin urmare din ecuația a 2-a se poate calcula necunoscuta x_2 . Apoi necunoscuta x_1 se poate calcula înlocuind x_2 în prima ecuație.

Pentru exemplul cu două ecuații și două necunoscute a fost suficientă premultiplicarea cu o singură matrice elementară, adică efectuarea unei singure operații elementare asupra sistemului. Pentru un sistem de n ecuații și necunoscute este nevoie să se efectueze mai multe astfel de operații până să se ajungă la forma triunghiulară. Mai exact, pentru fiecare element de sub diagonală principală este nevoie de premultiplicarea cu o matrice elementară.

Exemplu

```
In [1]: A = [1.3182606    0.8903206    0.5227336;
            0.6623476    1.5465000    0.0021365;
            0.8763162    0.5628361    1.6799823]

b = [0.41201;
     0.46483;
     0.67660]

A0 = A;
b0 = b;

A =

    1.3182606    0.8903206    0.5227336
    0.6623476    1.5465000    0.0021365
    0.8763162    0.5628361    1.6799823

b =

    0.41201
    0.46483
    0.67660
```

Se înmulțesc la stânga ambii membri pentru a elimina cei trei termeni diferiți de zero de sub diagonală principală. Pentru fiecare termen este nevoie de o multiplicare.

- Matricea elementară de eliminare pentru elementul de pe linia 2 și coloana 1

```
In [2]: n = 3;
E1 = eye(n,n);
```

Se aduna la linia 2, linia 1 multiplicată cu constanta $c = -\frac{A_{21}}{A_{11}}$

```
In [3]: E1(2,:) = E1(2,:) + E1(1,)*(-A(2,1)/A(1,1))

E1 =

    1.00000    0.00000    0.00000
   -0.50244    1.00000    0.00000
    0.00000    0.00000    1.00000
```

Se înmulțesc la stânga ambii membri ai sistemului de ecuației cu matricea E_1

```
In [4]: A = E1*A
b = E1*b

A =

    1.31826    0.89032    0.52273
    0.00000    1.09917   -0.26051
    0.87632    0.56284    1.67998

b =

    0.41201
    0.25782
    0.67660
```

Se observă că după multiplicare, elementul de linia 2 și coloana 1 a devenit zero. Mai departe se repetă procesul și pentru celelalte două elemente de sub diagonala principală.

Pentru linia 3, coloana 1:

```
In [5]: E2 = eye(n,n);
E2(3,:) = E2(3,:) + E2(1,)*(-A(3,1)/A(1,1))

E2 =

    1.00000    0.00000    0.00000
    0.00000    1.00000    0.00000
   -0.66475    0.00000    1.00000
```

```
In [6]: A = E2*A
b = E2*b

A =

    1.31826    0.89032    0.52273
    0.00000    1.09917   -0.26051
    0.00000   -0.02901    1.33249

b =

    0.41201
    0.25782
    0.40272
```

Și pentru linia 3, coloana 2:

```
In [7]: E3 = eye(n,n);
E3(3,:) = E3(3,:) + E3(2,)*(-A(3,2)/A(2,2))

E3 =

    1.00000    0.00000    0.00000
    0.00000    1.00000    0.00000
    0.00000    0.02639    1.00000
```

```
In [8]: A = E3*A
b = E3*b

A =

    1.31826    0.89032    0.52273
    0.00000    1.09917   -0.26051
    0.00000    0.00000    1.32562

b =

    0.41201
    0.25782
    0.40952
```

Se observă că matricea A a noului sistem de ecuatii obtinut are zero sub diagonala principală iar noul sistem are aceleași soluții ca vechiul sistem:


```
In [9]: x_vechi = inv(A0)*b0
x_nou = inv(A)*b
```

```
x_vechi =

-0.017823
0.307776
0.308927

x_nou =

-0.017823
0.307776
0.308927
```

Calculul soluțiilor prin înlocuire

Odată ce sistemul de ecuații a fost adus la forma triunghiulară, soluțiile pot fi calculate direct printr-un proces simplu de înlocuire. Mai exact, cum ultima linie maticiei A are un singur element diferit de zero, atunci această linie corespunde unei ecuații cu o singură necunoscută, deci se poate calcula direct soluția x_n a sistemului ca $x_n = b_n / A_{nn}$. Odată ce se cunoaște soluția x_n , aceasta poate fi înlocuită în ecuația $n - 1$, care este o ecuație cu două necunoscute: x_{n-1} și x_n . Înlocuind x_n în această ecuație se poate calcula x_{n-1} . Acest proces se repetă până când au fost calculate toate soluțiile.

Pentru exemplul de mai sus:

```
In [10]: x_3 = b(3)/A(3,3)
x_2 = (b(2) - A(2,3)*x_3)/A(2,2)
x_1 = (b(1) - A(1,2)*x_2 - A(1,3)*x_3)/A(1,1)

x_3 = 0.30893
x_2 = 0.30778
x_1 = -0.017823
```

Se observă că soluțiile obținute astfel sunt identice cu cele calculate mai sus prin inversiune.

Generalizarea pentru un sistem de n ecuații cu n necunoscute

```
In [11]: function [A_, b_] = gauss_elim(A,b)
# Variabilele de intrare sunt matricea coeficientilor A și matricea coloană a termenilor liberi b
# Variabilele de ieșire sunt noile matrici A_ și b_ unde A_ a fost adusă la forma triunghiulară
n = size(A)(1);
A_ = A;
b_ = b;
E = eye(n,n);
# Se parcurg toate elementele de sub diagonala principală
for col = 1:n
    for row = col+1:n
        # Se calculează matricea E pentru fiecare element de sub diagonala principală
        E = eye(n,n);
        factor = A_(row,col)/A_(col,col);
        E(row,:) = E(row,:) - E(col,:)*factor;

        # Se multiplică la stânga ambii membri ai sistemului cu matricea E calculată mai sus
        A_ = E*A_;
        b_ = E*b_;
    end
end
end
```

```
In [12]: function [x] = back_subst(A,b)
n = size(A)(1);
x = zeros(n,1);
# soluția x_n
x(n) = b(n)/A(n,n);
# soluțiile rămase în ordine inversă
for i = n-1:-1:1
    x(i) = (b(i) - A(i,i+1:n)*x(i+1:n))/A(i,i);
end
end
```

```
In [13]: [A_, b_] = gauss_elim(A0,b0)

A_ =

1.31826    0.89032    0.52273
0.00000    1.09917   -0.26051
0.00000    0.00000    1.32562

b_ =

0.41201
0.25782
0.40952
```

```
In [14]: x_ = back_subst(A_,b_)

x_ =

-0.017823
0.307776
0.308927
```

2. Descompunerea LU

Ideea de bază a descompunerii LU este descompunerea matricii A a sistemului într-un produs de două matrici triunghiulare L și U , $A = LU$ unde L are toate elementele de deasupra diagonalei principale zero iar U are toate elementele de sub diagonala principală zero.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Înlocuind matricea descompusă în sistemul de ecuații, se obține:

$$\left(\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Care, mutând parantezele, poate fi scris ca:

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \left(\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right) = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Notând produsul din interiorul parantezei $U\mathbf{x}$ cu \mathbf{d} se obține un nou sistem de ecuații unde vectorul necunoscutelor este \mathbf{d} :

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Cum matricea L este în forma triunghiulară, acest sistem de ecuații poate fi rezolvat direct prin înlocuire folosind metoda descrisă mai sus. După ce au fost obținute soluțiile \mathbf{d} , acestea se înlocuiesc în sistemul $U\mathbf{x} = \mathbf{d}$:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

Cum și acest sistem de ecuații are matricea coeficienților în forma triunghiulară, soluțiile \mathbf{x} pot fi obținute prin înlocuire.

Principalul avantaj al metodei descompunerii LU față de metoda Gauss este că operațiile se fac doar asupra matricii A a sistemului, nu și asupra vectorului \mathbf{b} , prin urmare această metodă este eficientă atunci când este nevoie să se rezolve mai multe sisteme de ecuații liniare în care matricea coeficienților A este aceeași și doar vectorul termenilor liberi \mathbf{b} se modifică. Practic se face descompunerea LU pentru matricea A o singură dată iar această descompunere este folosită pentru fiecare vector \mathbf{b} .

Calculul matricelor L și U

La baza descompunerii LU se află tot metoda eliminării Gauss. Fie $E_1 E_2 \dots E_n$ secvența de matrici elementare care a fost necesară pentru aducerea sistemului $A\mathbf{x} = \mathbf{b}$ la forma triunghiulară, deci matricea $E_1 E_2 \dots E_n A$ este o matrice triunghiulară superioară (toate elementele de sub diagonala principală sunt zero). Practic matricea U din descompunerea LU este matricea $E_1 E_2 \dots E_n A$. Pentru calculul matricii L se pornește de la afirmația că $A = LU$. Deci:

$$A = L E_1 E_2 \dots E_n A$$

De unde rezultă că:

$$L E_1 E_2 \dots E_n = I$$

Deci:

$$L = E_n^{-1} E_{n-1}^{-1} \dots E_1^{-1}$$

Cum matricile E sunt cunoscute, mai rămâne problema calculul inverselor acestora. Pentru aceasta ne folosim de proprietatea specifică matricelor elementare:

$$E_{ij}(c)^{-1} = E_{ij}(-c)$$

Adică:

$$E_{ij}(c)^{-1} = -E_{ij}(c) + 2I$$

```
In [15]: function [L,U] = LU_decomp(A)
    n = size(A)(1);
    L = eye(n,n);
    U = A;
    # Se parcurge fiecare element de sub diagonala principală
    for col = 1:n
        for row = col+1:n
            # Se calculează matricea elementară pentru reducerea elementului curent de sub diagonala principală
            E_tmp = eye(n,n);
            factor = U(row,col)/U(col,col);
            E_tmp(row,:) = E_tmp(row,:) - E_tmp(col,:)*factor;
            # Se calculează matricile U și L
            U = E_tmp*U;
            L = L*(-E_tmp + 2*eye(n,n)); #(-E_tmp + 2*eye(n,n)) reprezintă inversa matricii E_tmp
        end
    end
end
```

Exemplu

Se efectuează descompunerea LU a matricii A :

```
In [16]: [L U] = LU_decomp(A0)

L =

    1.00000    0.00000    0.00000
    0.50244    1.00000    0.00000
    0.66475   -0.02639    1.00000

U =

    1.31826    0.89032    0.52273
    0.00000    1.09917   -0.26051
    0.00000    0.00000    1.32562
```

Se observă că $LU = A$:

```
In [17]: L*U-A

ans =

    0.00000    0.00000    0.00000
    0.66235    0.44733    0.26264
    0.87632    0.56284    0.35436
```

Se rezolvă sistemul de ecuații $L\mathbf{d} = \mathbf{b}$ prin înlocuire:

```
In [18]: d_1 = b0(1)/L(1,1)
d_2 =(b0(2) - L(2,1)*d_1)/L(2,2)
d_3 =(b0(3) - L(3,1)*d_1 - L(3,2)*d_2)/L(3,3)

d_1 =    0.41201
d_2 =    0.25782
d_3 =    0.40952
```

Se rezolvă sistemul de ecuații $U\mathbf{x} = \mathbf{d}$ prin înlocuire:

```
In [19]: x_3 = d_3/U(3,3)
x_2 =(d_2 - U(2,3)*x_3)/U(2,2)
x_1 =(d_1 - U(1,2)*x_2 - U(1,3)*x_3)/U(1,1)

x_3 =    0.30893
x_2 =    0.30778
x_1 =   -0.017823
```

3. Calculul matricei inverse folosind descompunerea LU

Descompunerea LU este o metodă potrivită pentru a rezolva mai multe sisteme de ecuații liniare $A\mathbf{x} = \mathbf{b}$ ce au aceeași matrice A și diferite matrici \mathbf{b} . De altfel această metodă este avantajoasă pentru calculul inversei unei matrici deoarece este necesară o singură descompunere LU. Definiția unei matrici inverse A^{-1} este o matrice pentru care $AA^{-1} = I$:

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} a'_{11} & a'_{12} & a'_{13} \\ a'_{21} & a'_{22} & a'_{23} \\ a'_{31} & a'_{32} & a'_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Calculul A^{-1} pentru o matrice A cu dimensiunea $n \times n$ poate fi scrisă ca rezolvarea a n sisteme de ecuații liniare. De exemplu pentru cazul $n = 3$, prima coloană a matricei inverse se poate calcula rezolvând următorul sistem de ecuații:

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} a'_{11} \\ a'_{21} \\ a'_{31} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

A 2-a coloană se calculează schimbând matricea \mathbf{b} în $[0, 1, 0]^T$ iar a 3-a coloană cu $\mathbf{b} = [0, 0, 1]^T$. Se observă că doar matricea \mathbf{b} se modifică iar matricea A rămâne neschimbată. Prin urmare, pentru a rezolva aceste sisteme de ecuații este nevoie de o singură descompunere LU.

Exemplu:

Calculul inversei matricei A de mai sus:

```
In [20]: A = A0

A =

    1.3182606    0.8903206    0.5227336
    0.6623476    1.5465000    0.0021365
    0.8763162    0.5628361    1.6799823
```

Se calculează descompunerea LU pentru matricea A

```
In [21]: [L U] = LU_decomp(A)
```

L =

```
1.00000 0.00000 0.00000
0.50244 1.00000 0.00000
0.66475 -0.02639 1.00000
```

U =

```
1.31826 0.89032 0.52273
0.00000 1.09917 -0.26051
0.00000 0.00000 1.32562
```

Se rezolvă sistemele de ecuații:

$$A\mathbf{a}_1 = [1, 0, 0]^T$$

$$A\mathbf{a}_2 = [0, 1, 0]^T$$

$$A\mathbf{a}_3 = [0, 0, 1]^T$$

Unde \mathbf{a}_1 , \mathbf{a}_2 și \mathbf{a}_3 sunt coloanele matricei inverse.

```
In [22]: A_inv = zeros(3,3);
```

Prima coloană:

```
In [23]: d_1 = 1/L(1,1);
d_2 = (0 - L(2,1)*d_1)/L(2,2);
d_3 = (0 - L(3,1)*d_1 - L(3,2)*d_2)/L(3,3);
```

```
In [24]: x_3 = d_3/U(3,3);
x_2 = (d_2 - U(2,3)*x_3)/U(2,2);
x_1 = (d_1 - U(1,2)*x_2 - U(1,3)*x_3)/U(1,1);
A_inv(:,1) = [x_1,x_2,x_3]';
```

A 2-a coloană:

```
In [25]: d_1 = 0/L(1,1);
d_2 = (1 - L(2,1)*d_1)/L(2,2);
d_3 = (0 - L(3,1)*d_1 - L(3,2)*d_2)/L(3,3);
```

```
In [26]: x_3 = d_3/U(3,3);
x_2 = (d_2 - U(2,3)*x_3)/U(2,2);
x_1 = (d_1 - U(1,2)*x_2 - U(1,3)*x_3)/U(1,1);
A_inv(:,2) = [x_1,x_2,x_3]';
```

A 3-a coloană:

```
In [27]: d_1 = 0/L(1,1);
d_2 = (0 - L(2,1)*d_1)/L(2,2);
d_3 = (1 - L(3,1)*d_1 - L(3,2)*d_2)/L(3,3);
```

```
In [28]: x_3 = d_3/U(3,3);
x_2 = (d_2 - U(2,3)*x_3)/U(2,2);
x_1 = (d_1 - U(1,2)*x_2 - U(1,3)*x_3)/U(1,1);
A_inv(:,3) = [x_1,x_2,x_3]';
```

Verificare:

```
In [29]: A*A_inv
```

ans =

```
1.0000e+000 -6.0491e-017 -7.1886e-017
9.3203e-017 1.0000e+000 -3.6117e-017
7.2935e-017 -1.2433e-016 1.0000e+000
```

Metode numerice - Laborator 4

Rezolvarea ecuațiilor neliniare

Se dă funcția $f: [0.05, 2] \rightarrow \mathbb{R}$, $f(x) = \ln x + x$.

1. Implementați funcția f ca o funcție anonimă Matlab:

$$f = @(x) \log(x) + x$$

2. Rezolvați ecuația $f(x) = 0$ folosind, pe rând, fiecare din metodele enumerate mai jos. Intervalul în care se va căuta soluția ecuației este domeniul de definiție a funcției f . Salvați într-un vector eroarea obținută la fiecare iterație a algoritmului. Pentru fiecare metodă, soluția inițială x_0 de la care va porni căutarea este $x_0 = 1$. Condiția de oprire pentru metodele de rezolvare este ca eroarea e să fie mai mică decât 10^{-14}

- Metoda înjumătățirii intervalului
- Metoda secantei fixe
- Metoda secantei variabile
- Metoda Newton (Metoda tangentei)

Indicație: Eroarea e ce corespunde soluției x_i a ecuației $f(x) = 0$ se calculează astfel:
 $e = |f(x_i)|$

3. Reprezentați pe un grafic erorile obținute pentru fiecare din cele patru metode. Pe axa x a graficului va fi reprezentat numărul de iterații iar pe axa y vor fi reprezentate erorile pentru fiecare metodă. Axa y va fi reprezentată la scară logaritmică iar axa x la scară naturală (utilizați funcția *semilogy* în loc de *plot*).

4. INTERPOLARE NUMERICĂ

În mod curent există posibilitatea ca modul de variație al unei funcții să fie specificat prin intermediul unor puncte distincte obținute pe baza unor măsurători experimentale, fără a cunoaște expresia analitică a funcției. În această situație, determinarea valorii funcției într-un punct oarecare se va putea face prin interpolare sau extrapolare numerică.

4.1. INTERPOLARE LINIARĂ

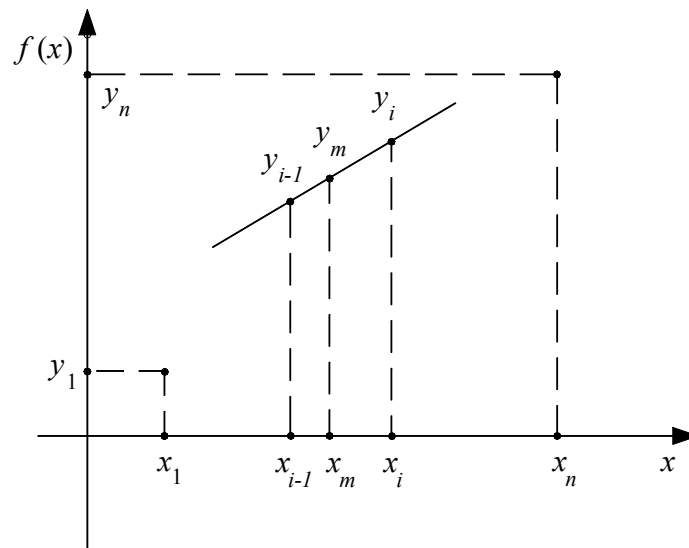


Fig. 4.1. Explicativă la interpolarea liniară.

Aproximarea valorii funcției se efectuează prin segmente de dreaptă.

Procedeeul este următorul (figura 4.1.):

- Se încadrează abscisa, pentru care se dorește determinarea funcției, între două abscise consecutive aparținând intervalului pentru care s-au efectuat măsurători, $x \in [x_{i-1}, x_i]$.
- Se aproximează liniar funcția prin dreapta care unește punctele de coordonate (x_{i-1}, y_{i-1}) și (x_i, y_i) adică dreapta având ecuația:

$$\frac{x - x_{i-1}}{x_i - x_{i-1}} = \frac{y - y_{i-1}}{y_i - y_{i-1}}.$$

- Valoarea funcției se obține efectuând intersecția dintre dreapta determinată anterior și verticala în punctul de abscisă x_m , se obține valoarea necunoscută a funcției:

$$y_m = y_{i-1} + \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \cdot (x_m - x_{i-1}). \quad (4.1)$$

Existența unui număr mare de determinări experimentale va asigura o bună precizie de determinare a valorilor funcției.

4.2. INTERPOLARE LAGRANGE

Aproximare prin polinom de gradul II.

Acest tip de aproximare necesită cunoașterea a 3 puncte ale graficului funcției. Notăm coordonatele acestor puncte astfel (x_{k-1}, y_{k-1}) , (x_k, y_k) , (x_{k+1}, y_{k+1}) .

Procedeul este următorul:

- Se încadrează abscisa pentru care se dorește determinarea funcției, între două abscise consecutive aparținând intervalului pentru care s-au efectuat măsurători, $x \in [x_{k-1}, x_k]$.
- Se aproximează funcția prin polinomul de gradul II a cărui grafic (parabola) conține punctele de coordonate (x_{k-1}, y_{k-1}) , (x_k, y_k) , (x_{k+1}, y_{k+1}) .

Considerăm polinomul de gradul II:

$$P(x) = a_2 \cdot x^2 + a_1 \cdot x + a_0$$

și vom determina coeficienții punând condiția ca graficul polinomului să conțină punctele specificate:

$$\begin{cases} a_2 \cdot x_{k-1}^2 + a_1 \cdot x_{k-1} + a_0 = y_{k-1} \\ a_2 \cdot x_k^2 + a_1 \cdot x_k + a_0 = y_k \\ a_2 \cdot x_{k+1}^2 + a_1 \cdot x_{k+1} + a_0 = y_{k+1} \end{cases}$$

Pentru a verifica dacă sistemul are soluție, presupunem cazul unui sistem omogen ($y_{k-1} = y_k = y_{k+1} = 0$). În această situație pot interveni două cazuri:

- Dacă determinantul sistemului este diferit de zero sistemul admite numai soluția banală ($a_0 = a_1 = a_2 = 0$)
- Dacă determinantul sistemului este zero sistemul admite soluție nebanală ($a_0 \neq 0, a_1 \neq 0, a_2 \neq 0$). Acest caz este însă imposibil deoarece ar însemna ca polinomul de gradul doi să aibă trei soluții distincte. Ca urmare, determinantul sistemului nu poate fi decât diferit de zero și prin urmare sistemul neomogen va fi întotdeauna compatibil determinat.

Pentru a obține, în mod convenabil, expresia polinomului $P(x)$, se va proceda după cum urmează. Se consideră polinoamele de gradul II:

$$Q_1(x) = (x - x_2) \cdot (x - x_1);$$

$$Q_2(x) = (x - x_1) \cdot (x - x_3);$$

$$Q_3(x) = (x - x_3) \cdot (x - x_2);$$

și se exprima polinomul $P(x)$ ca o combinație liniară a celor trei polinoame:

$$P(x) = b_1 \cdot Q_1(x) + b_2 \cdot Q_2(x) + b_3 \cdot Q_3(x). \quad (4.2)$$

Coeficienții reali b_1, b_2, b_3 se determină punând condițiile:

$$\begin{cases} P(x_1) = y_1 \\ P(x_2) = y_2 \\ P(x_3) = y_3 \end{cases} \Rightarrow \begin{cases} b_1 \cdot Q_1(x_1) = y_1 \\ b_2 \cdot Q_2(x_2) = y_2 \\ b_3 \cdot Q_3(x_3) = y_3 \end{cases} \Rightarrow \begin{cases} b_1 = \frac{y_1}{Q_1(x_1)} \\ b_2 = \frac{y_2}{Q_2(x_2)} \\ b_3 = \frac{y_3}{Q_3(x_3)} \end{cases}.$$

Ca urmare rezultă:

$$P(x) = y_1 \cdot \frac{Q_1(x)}{Q_1(x_1)} + y_2 \cdot \frac{Q_2(x)}{Q_2(x_2)} + y_3 \cdot \frac{Q_3(x)}{Q_3(x_3)} = \sum_{i=1}^3 y_i \cdot \frac{Q_i(x)}{Q_i(x_i)},$$

sau

$$P(x) = \sum_{i=1}^3 y_i \cdot \frac{\prod_{\substack{j=1 \\ j \neq i}}^3 (x - x_j)}{\prod_{\substack{j=1 \\ j \neq i}}^3 (x_i - x_j)}.$$

sau

$$P(x) = \sum_{i=1}^3 y_i \cdot \prod_{\substack{j=1 \\ j \neq i}}^3 \frac{x - x_j}{x_i - x_j}; \quad (4.3)$$

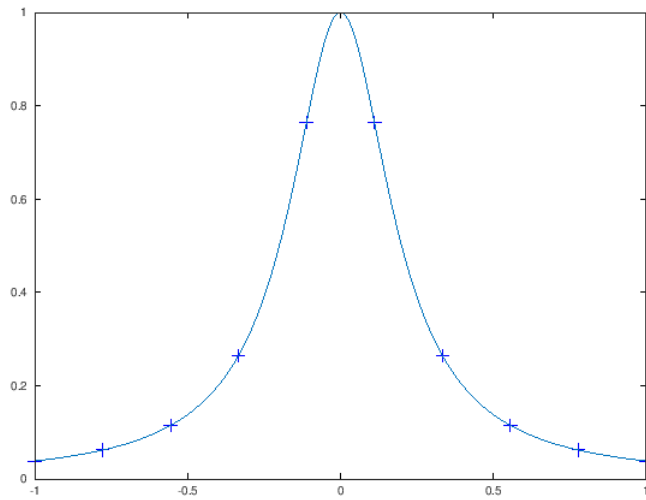
care reprezintă formula de interpolare a lui Lagrange pentru polinoame de gradul II.

În mod analog se obține formula de interpolare Lagrange pentru polinoame de gradul n :

$$P(x) = \sum_{i=1}^{n+1} y_i \cdot \prod_{\substack{j=1 \\ j \neq i}}^{n+1} \frac{x - x_j}{x_i - x_j}.$$

Interpolare numerică

```
In [1]: f = @(x) 1./(1+25*x.^2);  
x = linspace(-1,1,10);  
y = f(x);  
x_exact = linspace(-1,1,500);  
y_exact = f(x_exact);  
plot(x,y,'+b','MarkerSize',10,x_exact,y_exact)
```

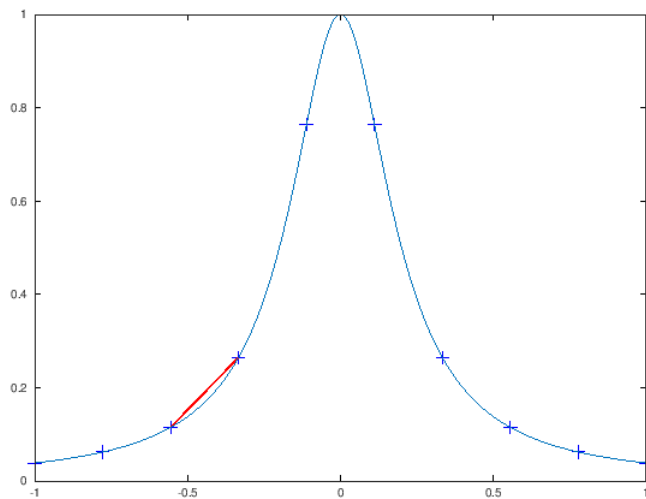


1. Interpolare liniară

```
In [2]: x1 = x(3)  
x2 = x(4)  
y1 = y(3)  
y2 = y(4)  
  
x1 = -0.55556  
x2 = -0.33333  
y1 = 0.11473  
y2 = 0.26471
```

```
In [3]: xm = linspace(x1,x2,100);  
for i = 1:100  
    ym(i) = y1 + (y2 - y1)/(x2 - x1)*(xm(i) - x1);  
end
```

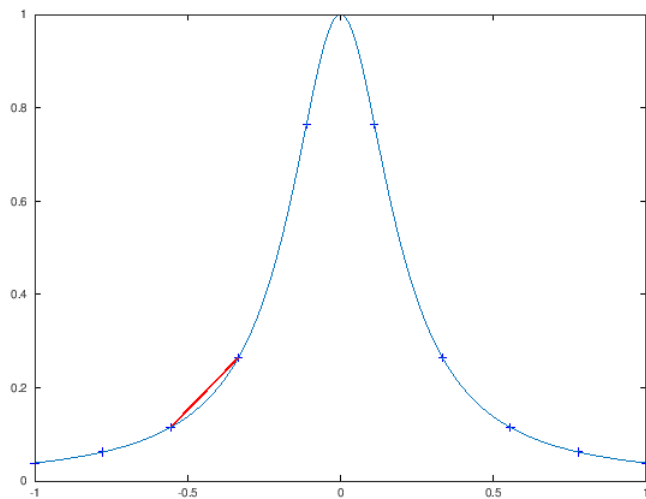
```
In [4]: plot(x,y,'+b','MarkerSize',10,xm,ym,'r','LineWidth',4,x_exact,y_exact)
```



Funcția predefinită Matlab `interp1()`

```
In [5]: ym = interp1(x,y,xm);
```

```
In [6]: plot(x,y,'+b',xm,ym,'r','LineWidth',4,x_exact, y_exact)
```



2. Interpolare polinomială

```
In [8]: px = x(2:4)';  
V = [px.^2 px.^1 px.^0]
```

V =

```
0.60494 -0.77778 1.00000  
0.30864 -0.55556 1.00000  
0.11111 -0.33333 1.00000
```

```
In [9]: det(V)
```

ans = -0.021948

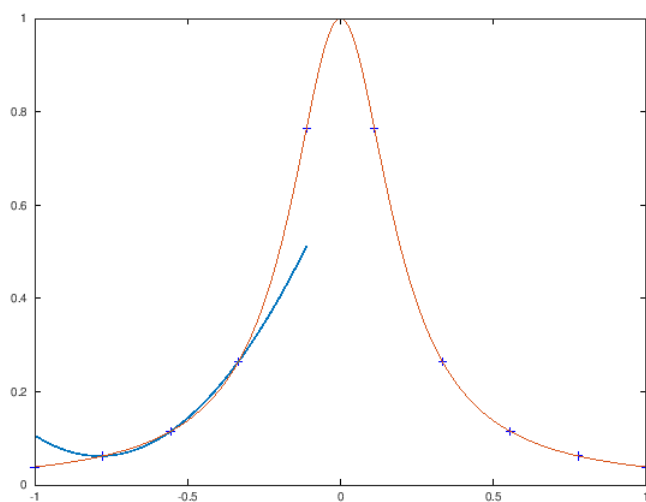
```
In [10]: py = y(2:4)';
```

```
In [11]: coefs = inv(V)*py
```

coefs =

```
0.98481  
1.55028  
0.67204
```

```
In [13]: xm = linspace(x(1), x(5),100);  
ym = polyval(coefs,xm);  
plot(x,y,'+b',xm, ym,'LineWidth',4,x_exact,y_exact)
```



```
In [14]: x1 = x(2)
x2 = x(3)
x3 = x(4)
y1 = y(2)
y2 = y(3)
y3 = y(4)
```

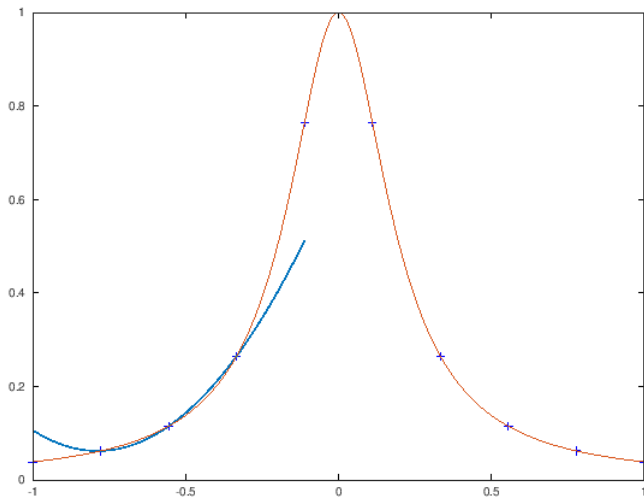
```
x1 = -0.77778
x2 = -0.55556
x3 = -0.33333
y1 = 0.062021
y2 = 0.11473
y3 = 0.26471
```

```
In [15]: l1 = @(x) (x - x2)./(x1 - x2) .* (x - x3)./(x1 - x3);
l2 = @(x) (x - x1)./(x2 - x1) .* (x - x3)./(x2 - x3);
l3 = @(x) (x - x2)./(x3 - x2) .* (x - x1)./(x3 - x1);
P = @(x) y1*l1(x) + y2*l2(x) + y3*l3(x)
```

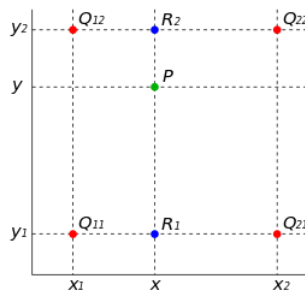
```
P =
```

```
@(x) y1 * l1 (x) + y2 * l2 (x) + y3 * l3 (x)
```

```
In [16]: plot(x,y,'+b',xm, P(xm),'LineWidth',4,x_exact,y_exact)
```



3. Interpolare biliniară



Presupunem că vrem să calculăm valoarea unei funcții necunoscute $f(x, y)$ cunoscându-se doar valoarea funcției în patru puncte: $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$ și $Q_{22} = (x_2, y_2)$.

Se pornește de la efectuarea unei interpolări liniare pe direcția x :

$$f(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

$$f(x, y_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

Folosind funcțiile interpolate anterior, se efectuează interpolarea și pe direcția y folosind expresiile pentru $f(x, y_1)$ și $f(x, y_2)$ calculate anter:

$$f(x, y) \approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2)$$

Înlocuind în ecuația de mai sus expresiile pentru $f(x, y_1)$ și $f(x, y_2)$ și efectuând calculele se obține:

$$f(x, y) = w_{11}f(Q_{11}) + w_{12}f(Q_{12}) + w_{21}f(Q_{21}) + w_{22}f(Q_{22}),$$

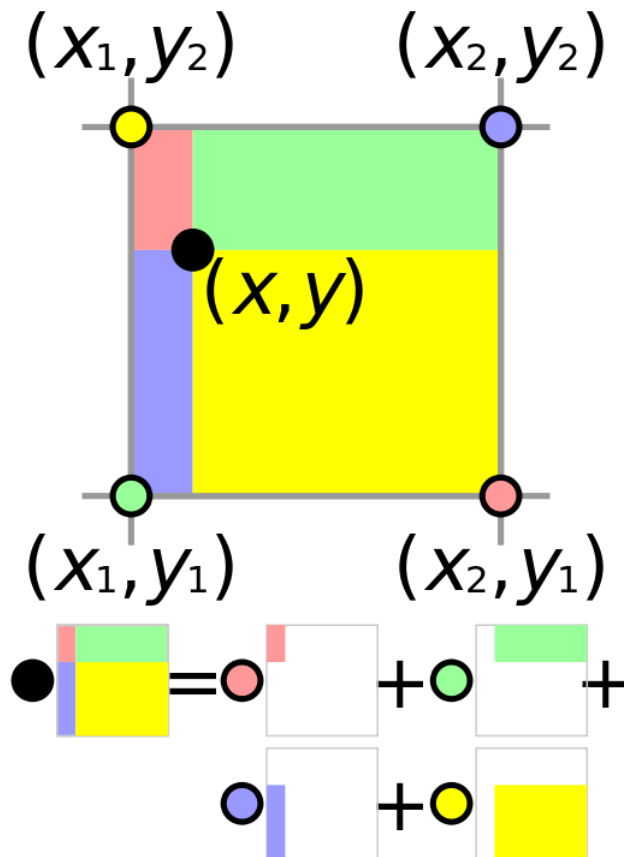
Unde:

$$w_{11} = \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)}$$

$$w_{12} = \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)}$$

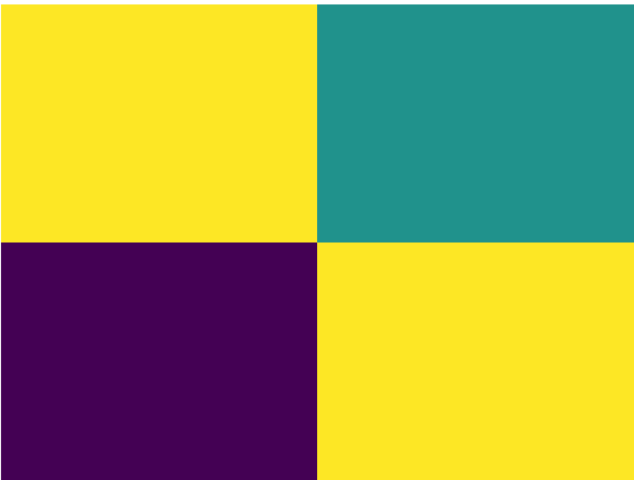
$$w_{21} = \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)}$$

$$w_{22} = \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)}$$



Exemplu

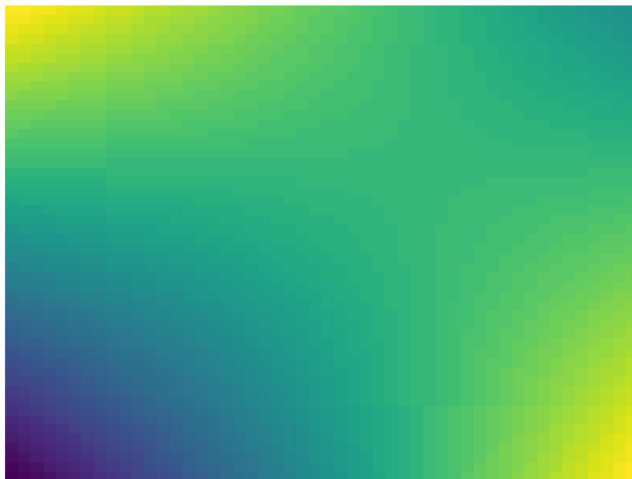
```
In [17]: x1 = 0;
y1 = 0;
x2 = 1;
y2 = 1;
f = [1, 0.5; 0, 1];
imagesc(f)
axis off;
```



```
In [18]: xp = linspace(0,1,50);
yp = linspace(0,1,50);
fp = zeros(50,50);

w00 = 1;
for i = 1:50
    for j = 1:50
        w11 = (x2 - xp(i))*(y2 - yp(j))/w00;
        w12 = (x2 - xp(i))*(yp(j) - y1)/w00;
        w21 = (xp(i) - x1)*(y2 - yp(j))/w00;
        w22 = (xp(i) - x1)*(yp(j) - y1)/w00;
        fp(i,j) = w11*f(1,1) + w12*f(1,2) + w21*f(2,1) + w22*f(2,2);
    end
end
```

```
In [19]: imagesc(fp)
axis off;
```



Metodă alternativă

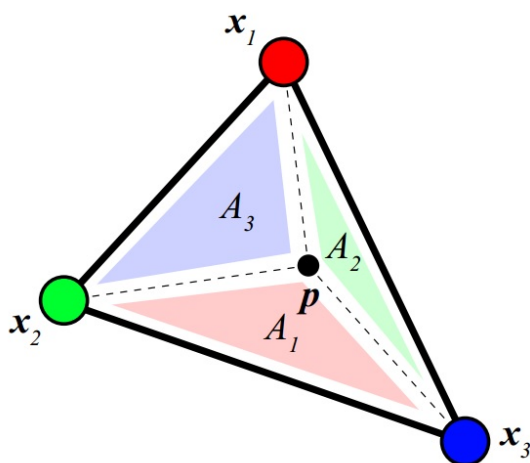
Se caută expresia funcției necunoscute $f(x, y)$ necunoscute ca o funcție de forma:

$$f(x, y) = a_0 + a_1x + a_2y + a_3xy,$$

Unde coeficienții a_0, a_1, a_2, a_3 pot fi calculați prin rezolvarea sistemului de ecuații liniare:

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_1 & y_2 & x_1y_2 \\ 1 & x_2 & y_1 & x_2y_1 \\ 1 & x_2 & y_2 & x_2y_2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f(Q_{11}) \\ f(Q_{12}) \\ f(Q_{21}) \\ f(Q_{22}) \end{bmatrix}$$

4. Interpolarea pe o rețea triunghiulară nestructurată



$$\left. \begin{aligned} \frac{A_1}{A} &= \lambda_1 \\ \frac{A_2}{A} &= \lambda_2 \\ \frac{A_3}{A} &= \lambda_3 \end{aligned} \right\} \begin{array}{l} \text{Coordonate} \\ \text{baricentrice} \end{array}$$

f în punctul p :

$$\sum_i \lambda_i = 1 \quad (A_1x_1 + A_2x_2 + A_3x_3)/A$$

Exemplu:

Fie triunghiul dat de următoarele trei puncte:

```
In [20]: p1 = [0 1];
p2 = [1,0];
p3 = [0,0];
```

Si funcția necunoscută f la care se cunoaște valoarea doar în cele trei puncte ale triunghiului: $f(\mathbf{p}_1), f(\mathbf{p}_2), f(\mathbf{p}_3)$

In [21]: `f = [1 1 0];`

Se cere calculul aproximativ pentru $f(\mathbf{p})$ unde \mathbf{p} este un punct oarecare ce se află în interiorul triunghiului.

In [22]: `p = [0.2 0.2];`

Se calculează coordonatele baricentrice:

In [23]: `A = 0.5*abs(det([p1 1; p2 1; p3 1]))`

`A = 0.50000`

In [24]: `A1 = 0.5*abs(det([p 1; p2 1; p3 1]))
A2 = 0.5*abs(det([p1 1; p 1; p3 1]))
A3 = 0.5*abs(det([p1 1; p2 1; p 1]))`

`A1 = 0.10000`

`A2 = 0.10000`

`A3 = 0.30000`

In [25]: `w1 = A1/A;
w2 = A2/A;
w3 = A3/A;`

In [26]: `w1*f(1)+w2*f(2)+w3*f(3)`

`ans = 0.40000`

In []:

In []:

Metode numerice - Laborator 5

Interpolare numerică

Fie funcția $f: \mathbb{R} \rightarrow \mathbb{R}$, necunoscută. Se cunoaște valoarea funcției în următoarele puncte $(x_i, f(x_i))$: (0,3), (1,2), (2,7), (3,24), (4,59), (5,118).

Se cere:

1. Să se calculeze valoarea funcției prin interpolare liniară în 100 de puncte distribuite uniform în intervalul $[0,5]$ și reprezentați pe grafic funcția interpolată.
2. Să se calculeze coeficienții a, b, c și d ai polinomului de gradul 3, $P(x) = ax^3 + bx^2 + cx + d$ ce trece prin primele patru puncte date.
3. Calculați valoarea funcției f în 100 de puncte distribuite uniform în intervalul $[0,3]$ utilizând polinomul $P(x)$ calculat la punctul 2 și reprezentați pe grafic variația funcției în acest interval.
4. **(Opțional)** Calculați $P(x)$ de la punctul 2 utilizând polinoame Lagrange și repetați punctul 3 cu noul polinom $P(x)$

Fie funcția de două variabile $f: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, necunoscută. Se cunoaște valoarea funcției în următoarele puncte:

	x			
		0	1	2
y	0	3	4	5
	1	5	7	9
	2	9	12	15

Se cere:

1. Să se calculeze valoarea funcției prin interpolare biliniară pentru 250 de puncte distribuite uniform în intervalul $[0,2] \times [0,2]$. Reprezentați pe grafic funcția interpolată (utilizați funcția predefinită *imagesc* pentru a reprezenta funcția ca o imagine 2D).
2. **(Opțional)** Să se calculeze valoarea funcției prin interpolare **bicubică** pentru 250 de puncte distribuite uniform în intervalul $[0,2] \times [0,2]$.

Metode numerice - Laborator 6

Metoda celor mai mici pătrate

1. Se dau fișierele `x1.csv` și `y1.csv` ce conțin coordonatele a 200 de puncte 2-dimensionale. Găsiți valorile a și b optime astfel încât dreapta de ecuație $y = ax + b$ să aproximeze cel mai bine punctele date. Afișați pe grafic punctele și dreapta găsită.
2. Se dau fișierele `x2.csv` și `y2.csv` ce conțin coordonatele a 200 de puncte 2-dimensionale. Găsiți valorile a, b și c optime astfel încât curba de ecuație $y = ax^2 + bx + c$ să aproximeze cel mai bine punctele date. Afișați pe grafic punctele și curba găsită.
3. Se dau fișierele `x3.csv` și `y3.csv` ce conțin coordonatele a 200 de puncte 3-dimensionale. Găsiți valorile parametrilor $a_i, i \in [1, 2, \dots, 7]$ optime astfel încât suprafața de ecuație $y = a_1 + a_2x_1 + a_3x_2 + a_4x_1x_2 + a_5x_1^2x_2 + a_6x_1x_2^2 + a_7x_1^2x_2^2$ să aproximeze cel mai bine punctele date. Reprezentați pe un grafic 3-dimensional punctele date și suprafața estimată. Folosiți codul de mai jos pentru crearea graficului:

```
f = @(x,y) a1+a2.*x+a3.*y+a4.*x.*y+a5.*x.^2.*y+a6.*x.*y.^2+a7.*x.^2.*y.^2;  
  
[X Y] = meshgrid(sort(x3(:,1)), sort(x3(:,2)));  
  
scatter3(x3(:,1),x3(:,2),y3, '+')  
  
hold on;  
  
surf(X,Y,f(X,Y), 'EdgeColor', 'none', 'FaceAlpha', 0.5)
```


5. DERIVARE ȘI INTEGRARE NUMERICĂ

5.1. DERIVARE NUMERICĂ

Acest calcul are drept scop rezolvarea numerică a limitei:

$$f'(x_0) = \lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0}. \quad (5.1)$$

- Aproximarea numerică a derivatei utilizând două puncte ale graficului funcției

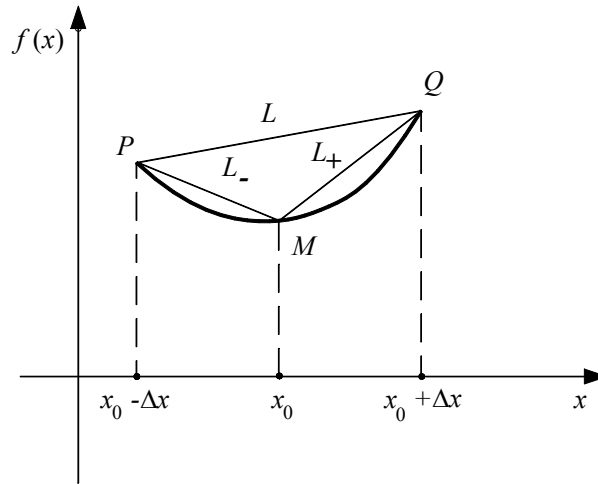


Fig. 5.1. Explicativă la derivarea numerică.

Pentru valori mici ale diferenței $\Delta x = x - x_0$ se poate aproxima valoarea derivatei funcției în punctul de abscisă x_0 fie cu panta dreptei MQ (L_+) fie cu panta dreptei MP (L_-) când $\Delta x < 0$.

$$\text{Dacă } \Delta x > 0 \Rightarrow f'(x_0) = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \text{ panta dreptei } (L_+).$$

$$\text{Dacă } \Delta x < 0 \Rightarrow f'(x_0) = \frac{f(x_0 - \Delta x) - f(x_0)}{-\Delta x} \text{ panta dreptei } (L_-).$$

Se poate aproxima derivata și prin panta dreptei PQ a cărei valoare este egală cu media aritmetică a pantelor dreptelor (L_+) și (L_-), deci

$$f'(x_0) = \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2 \cdot \Delta x};$$

notând $h = 2 \cdot \Delta x$ se obține în final relația:

$$f'(x_0) = \frac{f\left(x_0 + \frac{h}{2}\right) - f\left(x_0 - \frac{h}{2}\right)}{h}; \quad (5.2)$$

care reprezintă formula de aproximare numerică a derivatei unei funcții utilizând două puncte ale graficului.

Această expresie poate fi utilizată și pentru obținerea unei relații de calcul pentru derivata de ordinul 2 a unei funcții:

$$f''(x_0) = \frac{f'\left(x_0 + \frac{h}{2}\right) - f'\left(x_0 - \frac{h}{2}\right)}{h} = \frac{f(x_0 + h) - f(x_0) + f(x_0 - h) - f(x_0)}{h^2}.$$

Rezultă deci:

$$f''(x_0) = \frac{f(x_0 + h) - 2 \cdot f(x_0) + f(x_0 - h)}{h^2}. \quad (5.3)$$

• *Aproximarea numerică a derivatei utilizând trei puncte ale graficului funcției*

Notăm abscisele celor trei puncte astfel: $x_{-1} = x_0 - h_1$; x_0 ; $x_{+1} = x_0 + h_2$ și exprimăm derivata funcției în punctul de abscisă x_0 ca o combinație liniară a valorilor funcției calculate pentru cele trei abscise:

$$f'(x_0) = p_{-1} \cdot f(x_{-1}) + p_0 \cdot f(x_0) + p_{+1} \cdot f(x_{+1}). \quad (5.4)$$

Coefficienții reali p_{-1} , p_0 și p_{+1} se determină punând condiția ca relația de calcul care se obține să fie exactă pentru funcții de gradul 0, 1 și 2 adică să fie îndeplinite relațiile:

$$\text{Gradul 0} \quad f(x) = 1 \quad \rightarrow \quad f'(x_0) = 0.$$

$$\text{Gradul 1} \quad f(x) = x - x_0 \quad \rightarrow \quad f'(x_0) = 1.$$

$$\text{Gradul 2} \quad f(x) = (x - x_0)^2 \quad \rightarrow \quad f'(x_0) = 0.$$

Prin înlocuire se obține sistemul:

$$\begin{cases} p_{-1} + p_0 + p_{+1} = 0 \\ -h_1 \cdot p_{-1} + h_2 \cdot p_{+1} = 1 \\ h_1^2 \cdot p_{-1} + h_2^2 \cdot p_{+1} = 0 \end{cases};$$

care are soluțiile:

$$p_{-1} = \frac{-h_2^2}{h_1 \cdot h_2 \cdot (h_1 + h_2)}; \quad p_0 = \frac{h_2^2 - h_1^2}{h_1 \cdot h_2 \cdot (h_1 + h_2)}; \quad p_{+1} = \frac{h_1^2}{h_1 \cdot h_2 \cdot (h_1 + h_2)}.$$

Ca urmare se obține următoarea relație de calcul a derivatei:

$$f'(x_0) = \frac{-h_2^2 \cdot f(x_0 - h_1) + (h_2^2 - h_1^2) \cdot f(x_0) + h_1^2 \cdot f(x_0 + h_2)}{h_1 \cdot h_2 \cdot (h_1 + h_2)}. \quad (5.5)$$

Observație: Relații similare se pot deduce, utilizând același procedeu, pentru calculul derivatei utilizând 4 sau mai multe puncte ale graficului funcției.

5.2. INTEGRARE NUMERICĂ

Calculul are drept scop determinarea numerică a valorii integralei definite:

$$I = \int_a^b f(x) \cdot dx. \quad (5.6)$$

Sunt prezentate următoarele metode de calcul:

5.2.1. Metoda trapezelor

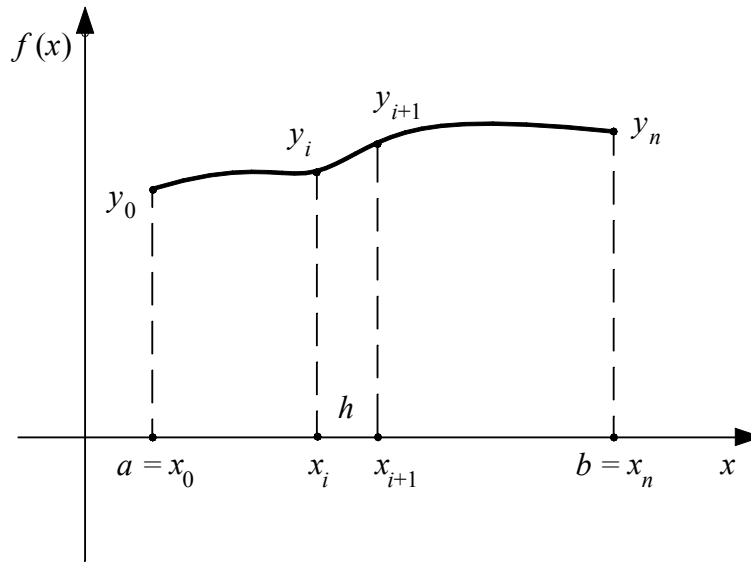


Fig. 5.2. Explicativă la metoda trapezelor.

Calculul constă în aproximarea suprafeței cuprinsă între graficul funcției, axa orizontală și verticalele duse în dreptul absciselor a și b , prin suma unor suprafețe elementare în formă de trapez, care rezultă prin împărțirea intervalului $[a, b]$ într-un număr n de subintervale. Rezultă trapeze de înălțime $h = \frac{b-a}{n}$. Astfel se aproximează liniar funcția pe fiecare subinterval.

Ca urmare se va face aproximarea:

$$I_i = \int_{x_i}^{x_{i+1}} f(x) \cdot dx \cong \frac{h}{2} \cdot (y_i + y_{i+1}). \quad (5.7)$$

Această aproximare permite determinarea valorii integralei prin relația:

$$I = \sum_{i=0}^n I_i = \frac{h}{2} \cdot (y_0 + y_1 + y_1 + y_2 + y_2 + \cdots + y_{n-1} + y_{n-1} + y_n).$$

Prin urmare relația de calcul a integralei definite, prin metoda trapezelor, va fi următoarea:

$$I = \frac{h}{2} \cdot (y_0 + 2 \cdot y_1 + 2 \cdot y_2 + 2 \cdot y_3 + \dots + 2 \cdot y_{n-2} + 2 \cdot y_{n-1} + y_n). \quad (5.8)$$

Pentru a putea evalua eroarea de trunchiere se va utiliza relația de dezvoltare în serie Taylor a funcției $f(x)$. Din relația de dezvoltare vom păstra termenii până la derivata de ordinul 2, adică primii trei termeni.

Pentru punctul de abscisă x_i (figura 5.2) se obține:

$$f_1(x) = f(x_i) + (x - x_i) \cdot f'(x_i) + \frac{(x - x_i)^2}{2} \cdot f''(x_i).$$

Iar pentru punctul de abscisă x_{i+1} se obține:

$$f_2(x) = f(x_{i+1}) + (x - x_{i+1}) \cdot f'(x_{i+1}) + \frac{(x - x_{i+1})^2}{2} \cdot f''(x_{i+1}).$$

Pentru simplificare se vor introduce notațiile:

$$f_1(x) = y_i + (x - x_i) \cdot y'_i + \frac{(x - x_i)^2}{2} \cdot y''_i ;$$

$$f_2(x) = y_{i+1} + (x - x_i - h) \cdot y'_{i+1} + \frac{(x - x_i - h)^2}{2} \cdot y''_{i+1} .$$

Pe intervalul $[x_i, x_{i+1}]$ aproximăm funcția prin expresia:

$$f(x) = \frac{f_1(x) + f_2(x)}{2},$$

ca urmare rezultă:

$$f(x) = \frac{y_i + y_{i+1}}{2} + (x - x_i) \cdot \frac{y'_i + y'_{i+1}}{2} - \frac{h}{2} \cdot y'_{i+1} + \frac{(x - x_i)^2}{4} \cdot (y''_{i+1} + y''_i) - \frac{(x - x_i) \cdot h}{2} \cdot y''_{i+1} + \frac{h^2}{4} \cdot y''_{i+1}$$

Utilizând această expresie se recalculează valoarea integralei pe intervalul $[x_i, x_{i+1}]$:

$$I_i = \int_{x_i}^{x_{i+1}} f(x) \cdot dx$$

După efectuarea calculelor și reducerea termenilor asemenea se obține:

$$I_i = \frac{h}{2} \cdot (y_{i+1} + y_i) - \frac{h^2}{4} \cdot (y'_{i+1} - y'_i) - \frac{h^3}{6} \cdot (y''_{i+1} - 2 \cdot y''_i). \quad (5.9)$$

În relația obținută (5.9), primul termen corespunde valorii calculate prin metoda trapezelor (5.7). Ca urmare eroarea de trunchiere produsă la această metodă poate fi apreciată ca fiind egală cu

$$e_{Ti} = -\frac{h^2}{2} \cdot (y'_{i+1} - y'_i) - \frac{h^3}{6} \cdot (y''_{i+1} - 2 \cdot y''_i). \quad (5.10)$$

Pentru valori mici ale lui h primul termen din relația (5.10) are valoarea dominantă. Vom presupune că eroarea de trunchiere are expresia:

$$e_{Ti} = K \cdot h^2 \cdot (y'_{i+1} - y'_i) \quad (5.11)$$

Deoarece relația (5.9) este valabilă pentru orice funcție alegem cazul particular $y = x^2$, ca urmare

$$I_i = \int_{x_i}^{x_{i+1}} x^2 \cdot dx = x_i^2 \cdot h + x_i \cdot h^2 + \frac{h^3}{3}. \quad (5.12)$$

pe de altă parte înlocuind în relația (5.9) $y_i = x_i^2$ și $y_{i+1} = (x_i + h)^2$ obținem

$$I_i = x_i^2 \cdot h + x_i \cdot h^2 + \frac{h^3}{2} + e_{Ti}. \quad (5.13)$$

Relațiile (5.12) și (5.13) conduc la egalitatea

$$e_{Ti} = -\frac{h^3}{6},$$

pe de altă parte, deoarece $y' = 2 \cdot x$ relația (5.11) devine

$$e_{Ti} = 2 \cdot K \cdot h^2 \cdot (x_i + h - x_i) = 2 \cdot K \cdot h^3,$$

rezultă

$$K = -\frac{1}{12},$$

și

$$e_{Ti} = -\frac{h^2}{12} \cdot (y'_{i+1} - y'_i). \quad (5.14)$$

Eroarea totală de trunchiere va fi

$$e_T = \sum_{i=0}^{i=n-1} e_{Ti} = -\frac{h^2}{12} \cdot (y'_b - y'_a); \quad (5.15)$$

în care a și b reprezintă capetele domeniului de integrare, ca urmare determinarea valorii erorii de trunchiere implică numai cunoașterea valorii primei derivate a funcției în două puncte și anume capetele domeniului de integrare.

O altă expresie a valorii erorii de trunchiere se poate obține cu ajutorul teoremei lui Lagrange:

$$y'_b - y'_a = (b-a) \cdot y''(\xi),$$

unde $a \leq \xi \leq b$, ca urmare noua expresie a erorii de trunchiere va fi:

$$e_T = -\frac{h^2}{12} \cdot (b-a) \cdot y''(\xi), \quad (5.16)$$

ca urmare determinarea valorii erorii de trunchiere implică numai cunoașterea valorii celei de a doua derivate a funcției într-un singur punct al domeniului de integrare.

Presupunând derivata de ordinul doi ca fiind aproximativ constantă pe intervalul de integrare relația (5.16) devine

$$e_T = c \cdot h^2, \quad (5.17)$$

unde c reprezintă o constantă.

Evaluarea erorii de trunchiere se poate face pe baza următorului raționament: presupunem I valoarea exactă a integralei, iar I_h valoarea integralei calculate prin metoda trapezelor utilizând

pasul $h = \frac{b-a}{n}$ și I_k valoarea integralei calculate prin metoda trapezelor utilizând pasul

$k = \frac{b-a}{m}$. Utilizând relația (5.17) pot fi scrise următoarele relații:

$$I = I_h + c \cdot h^2;$$

$$I = I_k + c \cdot k^2;$$

iar prin scădere se obține:

$$c = \frac{I_h - I_k}{k^2 - h^2};$$

și prin urmare valoarea integralei se poate scrie sub forma:

$$I = I_h + \frac{I_h - I_k}{k^2 - h^2} \cdot h^2;$$

sau:

$$I = I_h + \frac{I_h - I_k}{\frac{k^2}{h^2} - 1}, \quad (5.18)$$

rezultat care dă o mai bună aproximare a valorii integralei I decât I_h sau I_k .

5.2.2. Metoda Simpson

Metoda este similară metodei trapezelor deoarece presupune divizarea intervalului de integrare în subintervale iar funcția de integrat trebuie evaluată la capetele acestor subintervale.

Permite o aproximare mai bună a valorii integralei deoarece pe când în metoda trapezelor s-a folosit o dreaptă (polinom de gradul 1) pentru aproximarea ariei unui interval mic (rezultând un trapez) în metoda Simpson este utilizată o parabolă (polinom de gradul 2) pentru aproximarea ariei corespunzătoare la două intervale adiacente.

Formula de calcul a integralei se obține cu ajutorul relației obținute la metoda trapezelor. Presupunem numărul de subintervale n , din relația $h = \frac{b-a}{n}$, ca fiind par și alegem $k = 2 \cdot h$. Scriind relația dată de metoda trapezelor respectiv pentru subintervalele h și k obținem

$$I_h = \frac{h}{2} \cdot (y_0 + 2 \cdot y_1 + 2 \cdot y_2 + 2 \cdot y_3 + 2 \cdot y_4 + 2 \cdot y_5 + \dots), \quad (5.19)$$

respectiv

$$I_k = h \cdot (y_0 + 2 \cdot y_2 + 2 \cdot y_4 + 2 \cdot y_6 + \dots). \quad (5.20)$$

Pentru $k = 2 \cdot h$ relația (5.18) devine:

$$I = I_h + \frac{I_h - I_k}{4 - 1} = \frac{1}{3} \cdot (4 \cdot I_h - I_k). \quad (5.21)$$

Înlocuind relațiile (5.19) și (5.20) în relația (5.21) se obține:

$$I = \frac{h}{3} \cdot (y_0 + 4 \cdot y_1 + 2 \cdot y_2 + 4 \cdot y_3 + 2 \cdot y_4 + 4 \cdot y_5 + 2 \cdot y_6 + \dots); \quad (5.22)$$

relație care reprezintă formula de calcul a lui Simpson.

Observație: Pentru ambele metode prezentate cu cât numărul de puncte în care se evaluează funcția este mai mare (pasul de integrare este mai mic) cu atât rezultatul este mai precis.

5.2.3. Metoda Gauss

Metoda permite reducerea numărului de puncte în care se evaluează funcția la două.

Aplicarea metodei presupune efectuarea unei schimbări de variabilă astfel încât intervalul $[a, b]$ să fie reprezentat pe intervalul $[-1, 1]$. Schimbarea de variabilă va fi următoarea:

$$\frac{u - (-1)}{1 - (-1)} = \frac{x - a}{b - a}. \quad (5.23)$$

Rezultă:

$$u = \frac{2}{b - a} \cdot x - \frac{b + a}{b - a},$$

respectiv:

$$du = \frac{2}{b-a} \cdot dx.$$

Prin urmare va fi îndeplinită o egalitate de forma:

$$I = \int_a^b f(x) \cdot dx = \int_{-1}^1 \psi(u) \cdot du. \quad (5.24)$$

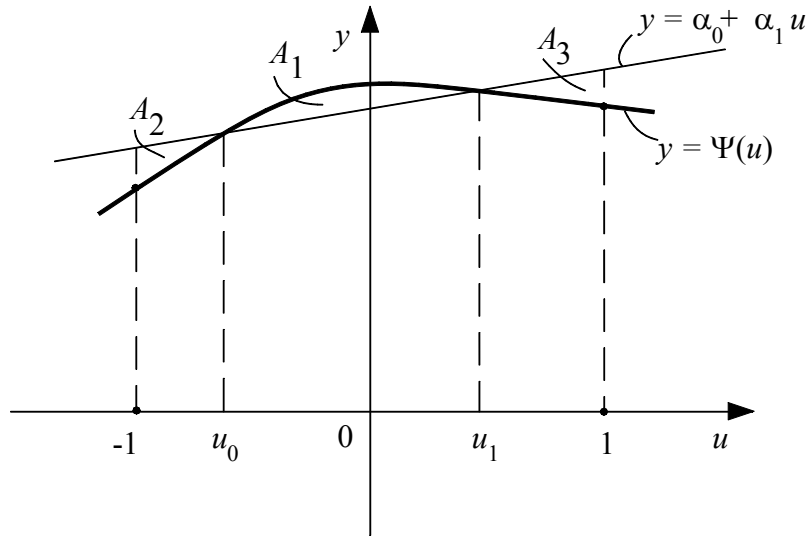


Fig. 5.3. Explicativă la metoda Gauss.

Metoda constă în determinarea unei drepte $y = \alpha_0 + \alpha_1 \cdot u$ pentru care să se obțină aceeași valoare a integralei pentru intervalul $[-1, 1]$, adică să fie îndeplinită egalitatea:

$$I = \int_{-1}^1 \psi(u) \cdot du = \int_{-1}^1 (\alpha_0 + \alpha_1 \cdot u) \cdot du. \quad (5.25)$$

Grafic această condiție revine la egalitatea dintre aria S_1 cuprinsă între graficul funcției și dreaptă, aflată deasupra dreptei și aria S_2 cuprinsă între graficul funcției și dreaptă, aflată sub dreaptă, figura 5.3.

Pentru a calcula integrala utilizând numai două evaluări ale funcției se mai pune condiția:

$$I = \int_{-1}^1 \psi(u) \cdot du = \int_{-1}^1 (\alpha_0 + \alpha_1 \cdot u) \cdot du = A_1 \cdot \psi(u_0) + A_2 \cdot \psi(u_1). \quad (5.26)$$

Valorile reale ale mărimilor α_0 , α_1 , A_1 , A_2 , u_0 , u_1 se obțin punând condiția de a se obține un rezultat exact în cazul unui polinom de gradul 3.

Se alege un polinom de gradul 3 având următoarea formă particulară:

$$\psi(u) = \alpha_0 + \alpha_1 \cdot u + (u - u_0) \cdot (u - u_1) \cdot (\beta_0 + \beta_1 \cdot u). \quad (5.27)$$

Valorile u_0 și u_1 se obțin punând condiția de a fi îndeplinită egalitatea:

$$I = \int_{-1}^1 (\alpha_0 + \alpha_1 \cdot u + (u - u_0) \cdot (u - u_1) \cdot (\beta_0 + \beta_1 \cdot u)) \cdot du = \int_{-1}^1 (\alpha_0 + \alpha_1 \cdot u) \cdot du,$$

care, după reducerea termenilor asemenea, devine:

$$\int_{-1}^1 (u - u_0) \cdot (u - u_1) \cdot (\beta_0 + \beta_1 \cdot u) \cdot du = 0. \quad (5.28)$$

În continuare se pune condiția ca această egalitate să fie îndeplinită oricare ar fi valorile coeficienților β_0 și β_1 . Ca urmare:

- Pentru perechea de valori $\beta_0 = 1$ și $\beta_1 = 0$ se obține egalitatea:

$$\int_{-1}^1 (u - u_0) \cdot (u - u_1) \cdot du = 0,$$

care devine:

$$\int_{-1}^1 u^2 \cdot du + (u_0 + u_1) \cdot \int_{-1}^1 u \cdot du + u_0 \cdot u_1 \cdot \int_{-1}^1 du = 0;$$

iar după rezolvarea integralelor:

$$\frac{2}{3} + 2 \cdot u_0 \cdot u_1 = 0;$$

adică:

$$u_0 \cdot u_1 = -\frac{1}{3}. \quad (5.29)$$

- Pentru perechea de valori $\beta_0 = 0$ și $\beta_1 = 1$ se obține egalitatea:

$$\int_{-1}^1 (u - u_0) \cdot (u - u_1) \cdot u \cdot du = 0, \text{ care devine:}$$

$$\int_{-1}^1 u^3 \cdot du + (u_0 + u_1) \cdot \int_{-1}^1 u^2 \cdot du + u_0 \cdot u_1 \cdot \int_{-1}^1 u \cdot du = 0;$$

iar după rezolvarea integralelor:

$$-\frac{2}{3} \cdot (u_0 + u_1) = 0;$$

adică:

$$u_0 + u_1 = 0. \quad (5.30)$$

Prin urmare se obține sistemul de ecuații:

$$\begin{cases} u_0 + u_1 = 0 \\ u_0 \cdot u_1 = -\frac{1}{3} \end{cases}$$

care are soluțiile:

$$u_0 = -\frac{\sqrt{3}}{3} \text{ și } u_1 = \frac{\sqrt{3}}{3}. \quad (5.31)$$

Coeficienții A_1 și A_2 se obțin punând condiția de a fi îndeplinită egalitatea:

$$I = \int_{-1}^1 (\alpha_0 + \alpha_1 \cdot u) \cdot du = A_1 \cdot \psi(u_0) + A_2 \cdot \psi(u_1);$$

care se poate scrie sub forma:

$$\alpha_0 \int_{-1}^1 du + \alpha_1 \int_{-1}^1 u \cdot du = A_1 \cdot \psi(u_0) + A_2 \cdot \psi(u_1). \quad (5.32)$$

Deoarece:

$$\psi(u_0) = \alpha_0 + \alpha_1 \cdot u_0;$$

și

$$\psi(u_1) = \alpha_0 + \alpha_1 \cdot u_1;$$

după rezolvarea integralelor, egalitatea devine:

$$2 \cdot \alpha_0 = A_1 \cdot (\alpha_0 + \alpha_1 \cdot u_0) + A_2 \cdot (\alpha_0 + \alpha_1 \cdot u_1). \quad (5.33)$$

După înlocuirea valorilor $u_0 = -\frac{\sqrt{3}}{3}$ și $u_1 = \frac{\sqrt{3}}{3}$ și gruparea termenilor se obține egalitatea:

$$\alpha_0 \cdot (A_1 + A_2) + \frac{\sqrt{3}}{3} \cdot \alpha_1 \cdot (-A_1 + A_2) = 2 \cdot \alpha_0. \quad (5.34)$$

În continuare se pune condiția ca această egalitate să fie îndeplinită oricare ar fi valorile coeficienților α_0 și α_1 . Ca urmare:

- Pentru perechea de valori $\alpha_0 = 1$ și $\alpha_1 = 0$ se obține egalitatea:

$$A_1 + A_2 = 2.$$

- Pentru perechea de valori $\alpha_0 = 0$ și $\alpha_1 = 1$ se obține egalitatea:

$$-A_1 + A_2 = 0.$$

Deci se obține sistemul:

$$\begin{cases} A_2 + A_1 = 2 \\ A_2 - A_1 = 0 \end{cases}$$

care are soluțiile:

$$A_1 = 1 \text{ și } A_2 = 1$$

În consecința formula de integrare prin metoda Gauss va avea expresia:

$$I = \psi\left(-\frac{\sqrt{3}}{3}\right) + \psi\left(\frac{\sqrt{3}}{3}\right). \quad (5.35)$$

Derivarea și integrarea numerică

```
In [5]: f = @(x) x.^3 + 3*x.^2 + 2*x + 5;  
f_der = @(x) 3*x.^2 + 6*x + 2;  
f_int = @(x) 1/4*x.^4 + x.^3 + x.^2 + 5*x;
```

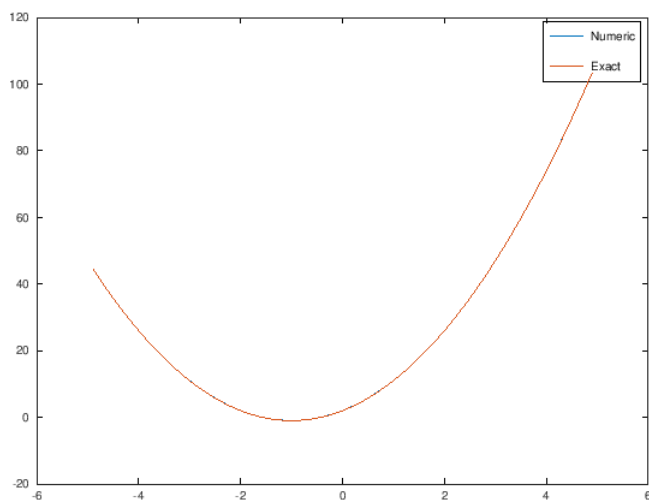
Derivarea numerică

```
In [6]: N = 50;  
x = linspace(-5,5,N);  
y = f(x);  
h = 10/(N-1);
```

```
In [7]: dydx = diff(y)./diff(x);
```

```
In [8]: dydx = diff(y)./h;
```

```
In [9]: x_ = x(1:N-1)+h/2;  
plot(x_,dydx,x_,f_der(x_))  
legend('Numeric','Exact')
```



Integrarea numerică

1. Metoda trapezelor

```
In [97]: N = 60;  
x = linspace(-5,5,N);  
y = f(x);
```

```
In [98]: h = 10/N
```

```
h = 0.16667
```

```
In [99]: w = ones(1,N)*2;  
w(1) = 1;  
w(N) = 1;
```

```
In [100]: I = h/2*sum(w.*y)
```

```
I = 295.14
```

```
In [101]: I_exact = f_int(5) - f_int(-5)
```

```
I_exact = 300
```

```
In [102]: e = abs(I-I_exact)
```

```
e = 4.8588
```

Îmbunătățirea preciziei prin calculul integralei cu doi pași k și h

- I_{exact} - Valoarea exactă a integralei
- I_h - Valoarea integralei calculată prin metoda trapezelor utilizând pasul h
- I_k - Valoarea integralei calculată prin metoda trapezelor utilizând pasul k

```
In [103]: sqrt(9.42)
```

```
ans = 3.0692
```

```

In [106]: Nh = 40;
          Nk = 20;
          xh = linspace(-5,5,Nh);
          xk = linspace(-5,5,Nk);
          yh = f(xh);
          yk = f(xk);
          h = 10/Nh
          k = 10/Nk

          h = 0.25000
          k = 0.50000

In [107]: wk = ones(1,Nk)*2;
          wh = ones(1,Nh)*2;
          wh(1) = 1;
          wh(Nh) = 1;
          wk(1) = 1;
          wk(Nk) = 1;

In [108]: Ih = h/2*wh*yh'
          Ik = k/2*wk*yk'

          Ih = 292.82
          Ik = 286.32

In [13]: I = Ih + (Ih - Ik)/(k^2/(h^2)-1)

          I = 294.99

```

3. Metoda Gauss

```

In [14]: a = -5;
          b = 5;

In [15]: u = 2/(b-a)*x-(b+a)/(b-a);

In [16]: Xi1 = interp1(u,y,-sqrt(3)/3)
          Xi2 = interp1(u,y,sqrt(3)/3)

          Xi1 = 0.098314
          Xi2 = 59.982

In [17]: I = (Xi1 + Xi2) * (b-a)/2

          I = 300.40

```

Metode numerice - Laborator 7

Derivare și integrare numerică

1. Se dă funcția $f: \mathbb{R} \rightarrow \mathbb{R}, f(x) = 2x^3 + 7x^2 + x + 1$.
 - a. Să se implementeze funcția f ca o funcție anonimă Matlab.
 - b. Să se calculeze derivata exactă $f'(x)$ și să se implementeze ca o funcție anonimă.
 - c. Să se calculeze derivata funcției $f(x)$ pe intervalul $[-10,10]$ printr-o metodă numerică.
 - d. Reprezentați pe grafic derivata exactă și cea numerică.
 - e. Calculați eroarea $e = \frac{1}{N} \sum_{i=1}^N |f'(x_i) - f'_{numeric}(x_i)|$ și observați efectul modificării pasului h asupra ei.
2. Se dă funcția $f: \mathbb{R} \rightarrow \mathbb{R}, f(x) = \sin(x)+1$
 - a. Să se implementeze funcția f ca o funcție anonimă Matlab.
 - b. Să se calculeze valoarea exactă a integralei $\int_0^{2\pi} f(x) dx$
 - c. Să se calculeze integrala de la punctul b prin metoda trapezelor cu pasul $h = 0.06$.
 - d. Să se calculeze integrala de la punctul b prin metoda trapezelor cu pasul $h = 0.03$.
 - e. Să se calculeze integrala de la punctul b prin metoda Simpson.
 - f. Să se calculeze integrala de la punctul b prin metoda cuadraturii Gauss în trei puncte.

Indicatie: Metoda cuadraturii Gauss în trei puncte are următorii parametri:

$$x_1 = -\sqrt{\frac{3}{5}}, w_1 = \frac{5}{9},$$

$$x_2 = 0, w_2 = \frac{8}{9},$$

$$x_3 = \sqrt{\frac{3}{5}}, w_3 = \frac{5}{9}$$

METODE DE INTEGRARE NUMERICĂ A ECUAȚIILOR DIFERENȚIALE FĂRĂ DERIVATE PARȚIALE

Aceste metode au drept scop determinarea graficului funcției pentru care se cunoaște expresia derivatei:

$$y' = f(x, y) ; \quad (6.1 \text{ a})$$

precum și valoarea funcției într-un punct:

$$y(x_0) = y_0 ; \quad (6.1 \text{ b})$$

această valoare purtând numele de *valoare inițială*.

Metodele utilizează formula de dezvoltare în serie Taylor, relație care permite determinarea valorii funcției într-un punct, dacă se cunoaște valoarea funcției într-un punct alăturat. Pentru o funcție de o singură variabilă, formula lui Taylor este:

$$y(x + Dx) = y(x) + Dx \times y' + \frac{(Dx)^2}{2!} \times y'' + \frac{(Dx)^3}{3!} \times y''' + \dots \quad (6.2)$$

Deoarece formula de dezvoltare în serie Taylor are o infinitate de termeni, se poate obține o relație de recurență prin păstrarea unui număr finit de termeni și alegerea unui pas de integrare h .

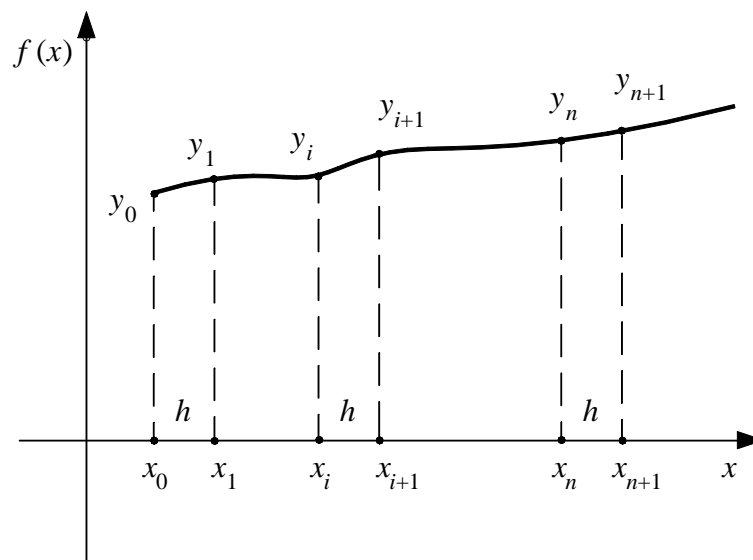


Fig. 6.1. Explicativă la integrarea numerică utilizând formula lui Taylor.

Ca urmare calculul valorilor funcției în puncte succesive se poate face utilizând o relație de recurență derivată din formula de dezvoltare în serie Taylor, astfel:

$$y_{m+1} = y_m + h \times y'_m + \frac{h^2}{2!} \times y''_m + \frac{h^3}{3!} \times y'''_m . \quad (6.3)$$

O astfel de relație nu poate fi utilizată practic deoarece expresia conține derivatele de ordin superior ale funcției, care sunt necunoscute. Ca urmare metodele de integrare numerică stabilesc formule de recurență care nu conțin derivatele funcției fiind totodată echivalente formulei de dezvoltare în serie Taylor.

Metodele de integrare pot fi de două categorii, în funcție de cantitatea de informație utilizată la deducerea valorii funcției într-un punct, precum și de modul de efectuare al calculelor:

- Ø *Metode directe* sau metode de tip Runge-Kutta, acest tip de metode utilizează numai informațiile obținute la punctul precedent iar calculul se efectuează prin aplicarea relației de recurență.
- Ø *Metode indirecte* sau de tip predictor-corector, acest tip de metode utilizează informațiile obținute la două puncte determinate anterior iar calculul se efectuează printr-un proces iterativ în care relația de recurență este aplicată până la obținerea valorii funcției cu o precizie impusă inițial.

În continuare vor fi prezentate cele două categorii de metode.

6.1. METODE DIRECTE SAU METODE DE TIP RUNGE-KUTTA

Acest tip de metode utilizează o formulă de recurență care simulează relația de dezvoltare în serie Taylor până la termenul care conține factorul h^p , valoarea lui p reprezentând *ordinul* metodei. De exemplu, o metodă care utilizează o formulă de recurență care simulează relația de dezvoltare în serie Taylor până la termenul care conține factorul h^4 va fi o metodă Runge-Kutta de ordinul 4. Modalitatea de deducere a formulelor de recurență este prezentată în continuare.

a) Metoda Runge-Kutta de ordinul 1 – metoda lui Euler

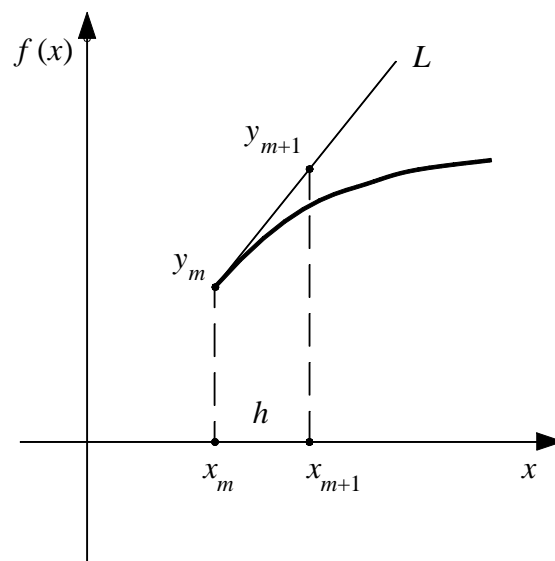


Fig. 6.2. Explicativă la metoda Runge-Kutta de ordinul 1 – metoda lui Euler.

Metoda constă în aproximarea graficului funcției într-un punct prin tangenta la grafic dusă în acel punct. Ca urmare formula de recurență se deduce astfel:

Presupunem cunoscute coordonatele (x_m, y_m) ale unui punct aparținând graficului funcției. Calculând valoarea derivatei funcției în acel punct, cu ajutorul relației (6.1 a), se obține valoarea $y'_m = f(x_m, y_m)$ care reprezintă panta tangentei la grafic în acel punct și a cărei ecuație este:

$$y - y_m = y'_m (x - x_m). \quad (6.4)$$

Dacă aproximăm graficul funcției cu tangenta la grafic rezultă că obținem următorul punct al graficului intersectând tangenta cu verticala dusă în dreptul abscisei $x_{m+1} = x_m + h$, ordonata acestui punct va fi:

$$y_{m+1} = y_m + h \times y'_m. \quad (6.5)$$

Relația (6.5) reprezintă formula de recurență în metoda Euler. După cum se poate constata relația (6.5) reproduce primii doi termeni ai dezvoltării în serie Taylor, relația (6.3), deci este echivalentă acesteia până la termenul care conține factorul h . Ca urmare, metoda Euler este o metodă de tip Runge-Kutta de ordinul 1.

Deoarece orice relație de recurență are un număr finit de termeni rezultă că va determina o eroare de trunchiere, notată cu e_T . Valoarea acesteia poate fi apreciată în funcție de pasul de integrare h . Deoarece valoarea pasului de integrare este subunitară rezultă că dintre toți termenii neglijați, din formula de dezvoltare în serie Taylor, cea mai mare valoare o va avea cel care conține pe h la puterea cea mai mică adică primul termen neglijat. Ca urmare la metoda Euler eroarea de trunchiere poate fi apreciată ca fiind $e_T = k \times h^2$. Aceasta determină o valoare semnificativă a erorii de trunchiere și ca urmare gradul de precizie al metodei Euler este redus.

b) Metode Runge-Kutta de ordinul 2

Aceste metode stabilesc formule de recurență care simulează formule de dezvoltare în serie Taylor până la termenul care conține factorul h^2 ca urmare eroarea de trunchiere produsă va fi apreciată ca fiind $e_T = k \times h^3$, deci metodele sunt mai precise decât cele de ordinul 1. După cum se va arăta în continuare, se pot determina o infinitate de formule de recurență de tipul menționat. Se vor prezenta două dintre ele precum și o generalizare.

- *Metoda lui Euler îmbunătățită.*

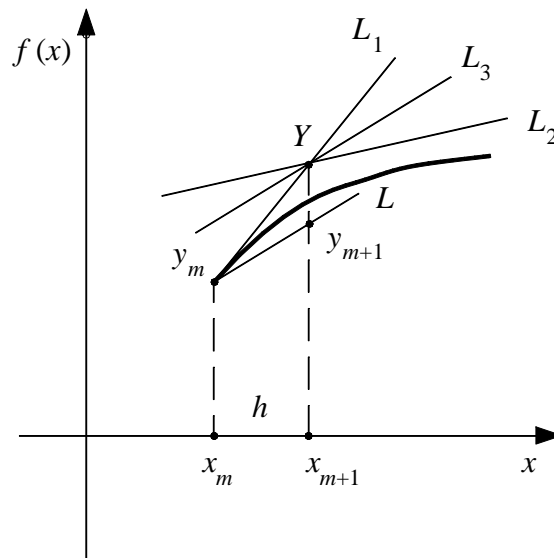


Fig. 6.3. Explicativă la metoda Runge-Kutta de ordinul 2 – metoda Euler îmbunătățită.

Presupunem cunoscute coordonatele (x_m, y_m) ale unui punct aparținând graficului funcției. Calculând valoarea derivatei funcției în acel punct, cu ajutorul relației (6.1 a), se obține valoarea $y'_m = f(x_m, y_m)$ care reprezintă panta tangentei la grafic în acel punct, dreapta L_1 din figura 6.2, și a cărei ecuație este:

$$y - y_m = y'_m (x - x_m) . \quad (6.6)$$

Intersecția acesteia cu verticala dusă în dreptul abscisei $x_{m+1} = x_m + h$ determină valoarea:

$$Y = y_m + h \times y'_m .$$

Pentru punctul de coordonate $[x_{m+1}, Y]$ se recalculează derivata funcției cu relația (6.1 a) obținându-se o valoare, $f(x_m + h, y_m + h \times y'_m)$, care reprezintă panta dreptei L_2 din figura 6.2.

Se determină panta dreptei L_3 ca medie aritmetică a pantelor dreptelor L_1 și L_2 :

$$Y = \frac{1}{2} [f(x_m, y_m) + f(x_m + h, y_m + h \times y'_m)] .$$

Dreapta L de pantă Y și care trece prin punctul de coordonate (x_m, y_m) va avea ecuația:

$$y - y_m = Y (x - x_m) .$$

Obținem următorul punct al graficului intersectând dreapta L cu verticala dusă în dreptul abscisei

$x_{m+1} = x_m + h$, ordonata acestui punct va fi:

$$y_{m+1} = y_m + \frac{h}{2} \left[f(x_m, y_m) + f(x_m + h, y_m + h \times y'_m) \right]; \quad (6.7)$$

relație care reprezintă formula de recurență pentru metoda Euler îmbunătățită.

Pentru a demonstra că relația (6.7) este echivalentă relației (6.3) până la termenul care conține factorul h^2 se va proceda după cum urmează. Se utilizează formula de dezvoltare în serie Taylor pentru o funcție de două variabile:

$$f(x + \Delta x, y + \Delta y) = f(x, y) + \Delta x \times \frac{\partial f}{\partial x} + \Delta y \times \frac{\partial f}{\partial y} + \dots$$

cu ajutorul căreia se poate scrie:

$$f(x_m + h, y_m + h \times y'_m) = f(x_m, y_m) + h \times \frac{\partial f}{\partial x} + h \times y'_m \times \frac{\partial f}{\partial y}; \quad (6.8)$$

se mai notează:

$$y'_m = f(x_m, y_m). \quad (6.9)$$

Se înlocuiește relația (6.8) în relația (6.7) și utilizând notația (6.9) se obține, după reducerea termenilor asemenea:

$$y_{m+1} = y_m + h \times y'_m + \frac{h^2}{2} \times \frac{\partial^2 f}{\partial x^2} + y'_m \times \frac{\partial^2 f}{\partial x \partial y}. \quad (6.10)$$

Derivând relația (6.1 a) în raport cu x se obține egalitatea:

$$y'' = \frac{\partial^2 f}{\partial x^2} + y' \times \frac{\partial^2 f}{\partial x \partial y}; \quad (6.11)$$

ca urmare relația (6.10) devine:

$$y_{m+1} = y_m + h \times y'_m + \frac{h^2}{2} \times y''_m;$$

adică este echivalentă relației (6.3) până la termenul care conține factorul h^2 prin urmare metoda Euler îmbunătățită este o metodă Runge-Kutta de ordinul 2.

- Metoda lui Euler modificată.

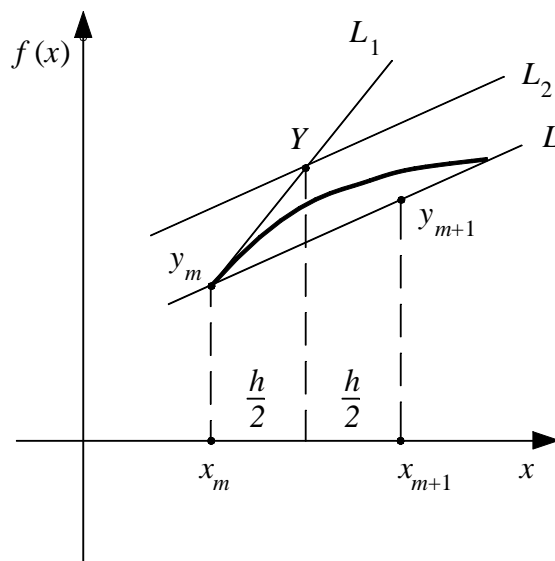


Fig. 6.4. Explicativă la metoda Runge-Kutta de ordinul II – metoda Euler modificată.

Presupunem cunoscute coordonatele (x_m, y_m) ale unui punct aparținând graficului funcției. Calculând valoarea derivatei funcției în acel punct, cu ajutorul relației (6.1 a), se obține valoarea $y'_m = f(x_m, y_m)$ care reprezintă panta tangentei la grafic în acel punct, dreapta L_1 din figura 6.3, și a cărei ecuație este:

$$y - y_m = y'_m (x - x_m) \quad (6.12)$$

Intersecția acesteia cu verticala dusă în dreptul abscisei $x_m + \frac{h}{2}$ determină valoarea:

$$Y = y_m + \frac{h}{2} \times y'_m.$$

Pentru punctul de coordonate $\left(x_m + \frac{h}{2}, Y\right)$ se recalculează valoarea derivatei funcției cu relația (6.1 a)

obținându-se o valoare, $f\left(x_m + \frac{h}{2}, y_m + \frac{h}{2} \times y'_m\right)$, care reprezintă panta dreptei L_2 din figura 6.3.

Dreapta L paralelă cu L_2 și care trece prin punctul de coordonate (x_m, y_m) va avea ecuația:

$$y - y_m = f\left(x_m + \frac{h}{2}, y_m + \frac{h}{2} \times y'_m\right) (x - x_m).$$

Obținem următorul punct al graficului intersectând dreapta L cu verticala dusă în dreptul abscisei

$x_{m+1} = x_m + h$, ordonata acestui punct va fi:

$$y_{m+1} = y_m + h \times f\left(x_m + \frac{h}{2}, y_m + \frac{h}{2} \times y'_m\right); \quad (6.13)$$

relație care reprezintă formula de recurență pentru metoda Euler modificată.

Într-un mod similar celui utilizat la metoda Euler îmbunătățită se poate arăta că și formula de recurență (6.13) este echivalentă relației (6.3) până la termenul care conține factorul h^2 , deci reprezintă o metodă Runge-Kutta de ordinul 2.

Generalizare

După cum se poate constata din relațiile (6.7) și (6.13) pentru determinarea relațiilor de recurență pentru metodele Runge-Kutta de ordinul 2, este utilizată o dreaptă de pantă Y care trece prin punctul de coordonate (x_m, y_m) . Vom determina panta acestei drepte astfel încât utilizarea ei să conducă la o formulă de recurență valabilă pentru o metodă Runge-Kutta de ordinul 2. În acest scop vom exprima panta Y prin expresia:

$$Y = a_1 \times f(x_m, y_m) + a_2 \times f(x_m + b_1 \times h, y_m + b_2 \times h \times y'_m); \quad (6.14)$$

în care au fost introduși coeficienții reali a_1, a_2, b_1, b_2 a căror valori urmează a fi determinate corespunzător scopului propus.

Se poate constata că pentru valorile:

$$a_1 = \frac{1}{2}, a_2 = \frac{1}{2}, b_1 = 1, b_2 = 1; \text{ se obține relația de recurență (6.7);}$$

iar pentru valorile:

$$a_1 = 0, a_2 = 0, b_1 = \frac{1}{2}, b_2 = \frac{1}{2}; \text{ se obține relația de recurență (6.13).}$$

Așa după cum am arătat determinarea valorilor posibile ale coeficienților reali a_1, a_2, b_1, b_2 se va efectua punând condiția ca relațiile:

$$y_{m+1} = y_m + h \times Y; \quad (6.15)$$

și

$$y_{n+1} = y_n + h \times y'_n + \frac{h^2}{2!} \times y''_n; \quad (6.16)$$

să fie echivalente.

Utilizând formula de dezvoltare în serie Taylor pentru o funcție de două variabile, se poate exprima:

$$f(x_m + b_1 \times h, y_m + b_2 \times h \times y'_m) = f(x_m, y_m) + b_1 \times h \times \frac{\partial f}{\partial x} + b_2 \times y'_m \times h \times \frac{\partial f}{\partial y}. \quad (6.17)$$

După înlocuirea relației (6.17) în relația (6.14) și după reducerea termenilor asemenea se obține:

$$Y = (a_1 + a_2) \times f(x_m, y_m) + a_2 \times b_1 \times h \times \frac{f}{x} + a_2 \times b_2 \times y'_m \times h \times \frac{f}{y};$$

utilizând notația (6.9) rezultă expresia:

$$Y = (a_1 + a_2) \times y'_m + a_2 \times b_1 \times h \times \frac{f}{x} + a_2 \times b_2 \times y'_m \times h \times \frac{f}{y};$$

și ca urmare relația (6.15) devine:

$$y_{m+1} = y_m + h \times (a_1 + a_2) \times y'_m + a_2 \times b_1 \times h \times \frac{f}{x} + a_2 \times b_2 \times y'_m \times h \times \frac{f}{y};$$

sau

$$y_{m+1} = y_m + h \times (a_1 + a_2) \times y'_m + h^2 \times a_2 \times b_1 \times \frac{f}{x} + a_2 \times b_2 \times y'_m \times h \times \frac{f}{y}. \quad (6.18)$$

Ținând seama și de relația (6.11) condițiile de echivalență dintre relațiile (6.18) și (6.16) vor fi:

$$\begin{aligned} a_1 + a_2 &= 1 \\ a_2 \times b_1 &= \frac{1}{2}; \\ a_2 \times b_2 &= \frac{1}{2} \end{aligned}$$

Rezultă prin urmare un sistem de 3 ecuații cu 4 necunoscute, sistem compatibil nedeterminat. Pentru rezolvarea acestuia se alege una dintre necunoscute ca variabilă și se obține:

$$\begin{aligned} a_1 &= 1 - l \\ a_2 &= l \\ b_1 &= \frac{1}{2l} \quad l \in R - \{0\}. \\ b_2 &= \frac{1}{2l} \end{aligned}$$

Pentru $l = \frac{1}{2}$ se obține relația de recurență (6.7) iar pentru $l = 1$ se obține relația de recurență (6.13).

În mod analog se pot determina metode Runge-Kutta de ordinul 3 și 4. Cea mai utilizată metodă este metoda Runge-Kutta de ordinul 4 care este definită prin următoarea relație de recurență:

$$y_{m+1} = y_m + \frac{h}{6} \times (k_1 + 2 \times k_2 + 2 \times k_3 + k_4), \quad (6.19)$$

unde

$$k_1 = f(x_m, y_m), \quad (6.20)$$

$$k_2 = f\left(x_m + \frac{h}{2}, y_m + \frac{h \times k_1}{2}\right), \quad (6.21)$$

$$k_3 = f\left(x_m + \frac{h}{2}, y_m + \frac{h \times k_2}{2}\right), \quad (6.22)$$

$$k_4 = f(x_m + h, y_m + h \times k_3). \quad (6.23)$$

Eroarea de trunchiere în acest caz este $e_T \approx K h^5$. Aceasta valoare poate fi evaluată pe baza următorului raționament. Fie Y_m valoarea reală a soluției în punctul $x = x_0 + m \times h$, ca urmare:

$$Y_m = y_m^{(h)} + K h^5, \quad (6.24)$$

unde indicele superior (h) indică faptul că y_m a fost calculat pentru dimensiunea h a intervalului.

Recalculăm soluția pentru o dimensiune de $\frac{h}{2}$, astfel încât

$$Y_m = y_m^{\frac{h}{2}} + K \left(\frac{h}{2}\right)^5. \quad (6.25)$$

Scăzând (6.24) din (6.25), obținem

$$y_m^h - y_m^{\frac{h}{2}} = -\frac{31}{32} K h^5,$$

iar eroarea de trunchiere este

$$e_T \approx K h^5 = \frac{32}{31} \left(y_m^{\frac{h}{2}} - y_m^{(h)} \right). \quad (6.26)$$

Evaluarea este suficient de bună, dificultatea constă în faptul că soluția trebuie calculată de două ori. Pentru calculul valorii $y_m^{(h)}$ sunt necesare $4m$ evaluări ale funcției (patru pentru fiecare interval), iar calculul valorii $y_m^{\frac{h}{2}}$ necesită $8m$ evaluări ale funcției. Ca urmare utilizarea relației (6.26) necesită un număr mare de calcule.

Caz particular: $f(x, y) = F(x)$

În situația în care funcția de integrat depinde numai de x relația (6.1 a) devine

$$y' = F(x);$$

și ca urmare

$$\int_{x_0}^x F(x) dx = y(x) - y(x_0) = y(x) - y_0.$$

În continuare vom utiliza mărimea:

$$p = \frac{h}{2};$$

precum și următoarele notații:

$$F_{2m} = F(x_0 + m \cdot h) = F(x_0 + 2m \cdot p),$$

$$Y_{2m} = y(x_0 + m \cdot h) = y(x_0 + 2m \cdot p).$$

Utilizând aceste notații relațiile (6.20), (6.21), (6.22) și (6.23) devin, respectiv:

$$k_1 = f(x_m) = F_{2m},$$

$$k_2 = f'(x_m) + \frac{h^2}{2} F_{2m+1},$$

$$k_3 = f''(x_m) + \frac{h^2}{2} F_{2m+1},$$

$$k_4 = f(x_m + h) = F_{2m+2}$$

și ca urmare relația de recurență (6.19) devine

$$Y_{2m+2} - Y_{2m} = \frac{p}{3} (F_{2m} + 4F_{2m+1} + F_{2m+2}),$$

unde $m = 0, 1, 2, \dots, n-1$. Astfel:

$$Y_2 - Y_0 = \frac{p}{3} (F_0 + 4F_1 + F_2),$$

$$Y_4 - Y_2 = \frac{p}{3} (F_2 + 4F_3 + F_4),$$

$$Y_6 - Y_4 = \frac{p}{3} (F_4 + 4F_5 + F_6),$$

.....

$$Y_{2n} - Y_{2n-2} = \frac{p}{3} (F_{2n-2} + 4F_{2n-1} + F_{2n}).$$

Adunând aceste ecuații, obținem

$$Y_{2n} - Y_0 = \frac{P}{3} (F_0 + 4F_1 + 2F_2 + 4F_3 + 2F_4 + \dots + 2F_{2n-2} + 4F_{2n-1} + F_{2n}).$$

Relația obținută reprezintă formula lui Simpson (5.12). Ca urmare metoda Runge-Kutta de ordinul 4 dată de relațiile (6.19) – (6.23) este o generalizare a metodei Simpson pentru funcții de mai multe variabile. Din acest motiv este numită și metoda Runge-Kutta-Simpson.

Metode de integrare numerică a ecuațiilor diferențiale

1. Metoda Euler

Fie ecuația diferențială $y' = y$ unde $y = y(t)$ și condiția inițială $y(0) = 1$. Soluția analitică a acestei ecuații este de forma $y(t) = ce^t$ iar punând condiția $y(0) = 1$ se obține $c = 1$, deci soluția ecuației este $y(t) = e^t$. În continuare se va aplica metoda Euler pentru rezolvarea numerică a acestei ecuații.

```
In [33]: y_exact = @(x) exp(x); # Soluția exactă
```

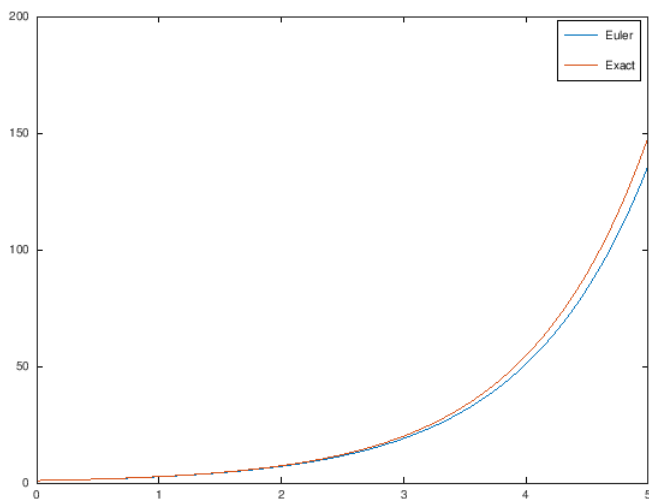
```
In [46]: a = 0; b = 5; # intervalul pe care dorim să rezolvăm ecuația
N = 200; # Numărul de puncte în care este împărțit intervalul

t = linspace(a,b,N);
h = (b-a)/N;

y = zeros(1,N);
y(1) = 1; # condiția inițială
```

```
In [47]: for i = 1:N-1
          y(i+1) = y(i) + h*y(i);
        end
y_euler = y;
```

```
In [48]: plot(t,y, t, y_exact(t))
legend('Euler', 'Exact')
```

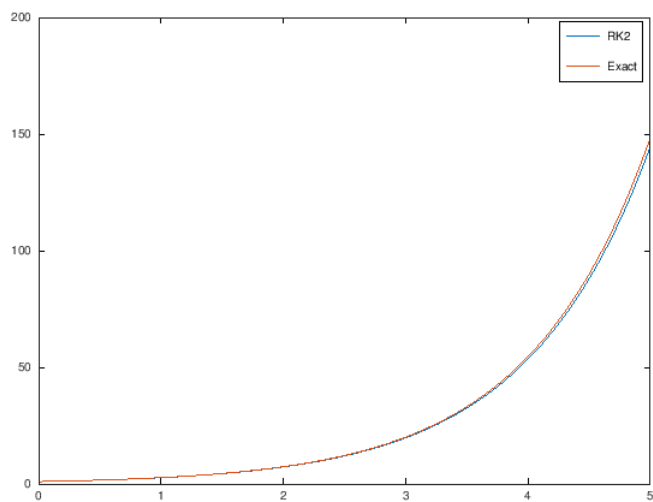


2. Metoda Runge-Kutta de ordinul 2

Varianta 1:

```
In [49]: for i = 1:N-1
          y(i+1) = y(i) + h/2*(y(i) + (y(i) + h*y(i)));
        end
y_rk2_v1 = y;
```

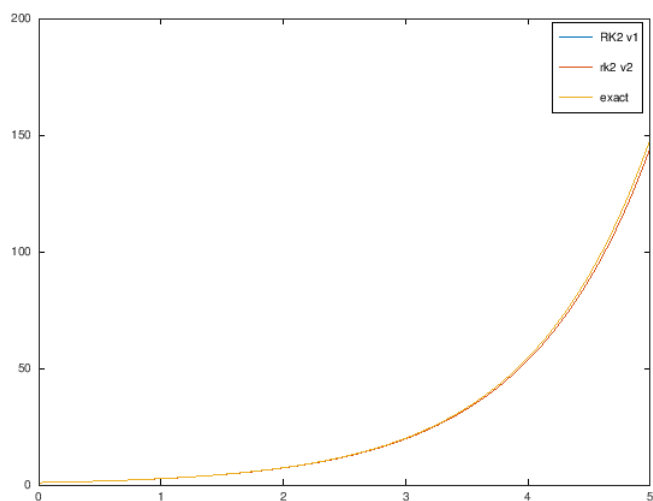
```
In [50]: plot(t,y, t, y_exact(t))
legend('RK2', 'Exact')
```



Varianta 2:

```
In [55]: for i = 1:N-1
          y(i+1) = y(i) + h*(y(i) + h/2*y(i));
        end
        y_rk2 = y;
```

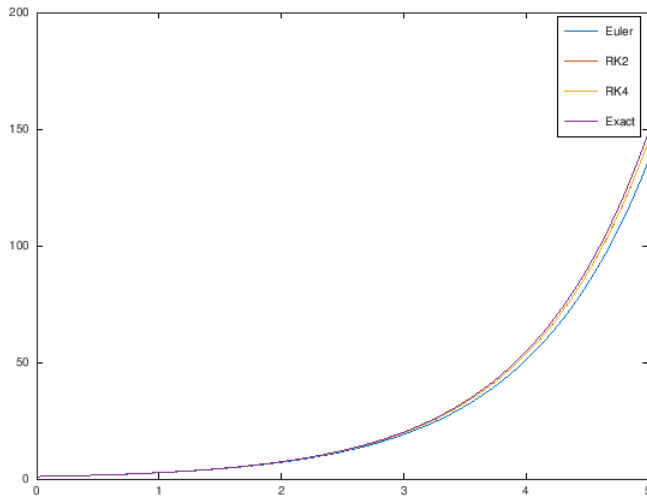
```
In [56]: plot(t,y_rk2_v1,t,y_rk2_v2, t, y_exact(t))
legend('RK2 v1', 'rk2 v2', 'exact')
```



Metoda Runge-Kutta de ordinul 4

```
In [57]: for i = 1:N-1
          k1 = y(i);
          k2 = y(i) + h*k1/2;
          k3 = y(i) + h*k2/2;
          k4 = y(i) + h*k3;
          y(i+1) = y(i) + h/6*(k1 + 2*k2 + 2*k3 + k4);
        end
        y_rk4 = y;
```

```
In [58]: plot(t,y_euler,t, y_rk2, t, y_rk4, t, y_exact(t))
legend('Euler','RK2','RK4','Exact')
```



Integrarea numerică a sistemelor de ecuații diferențiale de ordinul 1

1. Aproximarea derivatei cu diferențe finite (Metoda Euler)

Fie sistemul de ecuații diferențiale liniare $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B(t)$, unde $\mathbf{x} = [x_1(t), x_2(t), \dots, x_n(t)]^T$ este matricea coloană a necunoscutelor și are dimensiunea $n \times 1$, A este matricea coeficienților și are dimensiunea $n \times n$, $\dot{\mathbf{x}} = [\dot{x}_1(t), \dot{x}_2(t), \dots, \dot{x}_n(t)] = \left[\frac{dx_1(t)}{dt}, \frac{dx_2(t)}{dt}, \dots, \frac{dx_n(t)}{dt} \right]$ este derivata lui \mathbf{x} în raport cu variabila t iar B este matricea coloană a termenilor liberi.

Metoda constă în aproximarea derivatei cu diferențe finite de ordinul 1 cu scopul de a obține o relație de recurență din care se poate calcula $\mathbf{x}(t + \Delta t)$ în funcție de \mathbf{x} la pasul precedent adică $\mathbf{x}(t)$.

$$\dot{\mathbf{x}} \approx \frac{\mathbf{x}(t + \Delta t) - \mathbf{x}(t)}{\Delta t}$$

Înlocuind aproximarea în sistemul de ecuații rezultă:

$$\frac{\mathbf{x}(t + \Delta t) - \mathbf{x}(t)}{\Delta t} = A\mathbf{x}(t) + B(t)$$

De unde se poate formula relația de recurență:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t (A\mathbf{x}(t) + B(t))$$

Relația de mai sus poate fi văzută ca echivalentul metodei Euler pentru sisteme de ecuații diferențiale.

Exemplu

Fie ecuația diferențială $\ddot{y}(t) + 2\zeta\omega_n\dot{y}(t) + \omega_n^2 y(t) = F(t)$ ce reprezintă ecuația de mișcare a unui oscilator armonic.

- $y(t)$ este deplasarea în funcție de timp
- ζ este factorul de amortizare și este dat de coeficientul de frecare
- ω_n este frecvența proprie a sistemului

Orice ecuație diferențială liniară de ordinul n poate fi scrisă ca un sistem de n ecuații liniare de ordinul 1. Prin urmare ecuația de mai sus poate fi scrisă ca un sistem de 2 ecuații diferențiale de ordinul 1 prin următoarele schimbări de variabilă:

$$x_1 = y \quad x_2 = \dot{y}$$

De unde rezultă:

$$\dot{x}_1 = x_2 \quad \dot{x}_2 = -2\zeta\omega_n x_1 - \omega_n^2 x_1 + F(t) = -2\zeta\omega_n x_2 - \omega_n^2 x_1 + F(t)$$

Sau:

$$\begin{bmatrix} \dot{x}_1 & \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x_1 & x_2 \end{bmatrix} + \begin{bmatrix} 0 & F(t) \end{bmatrix}$$

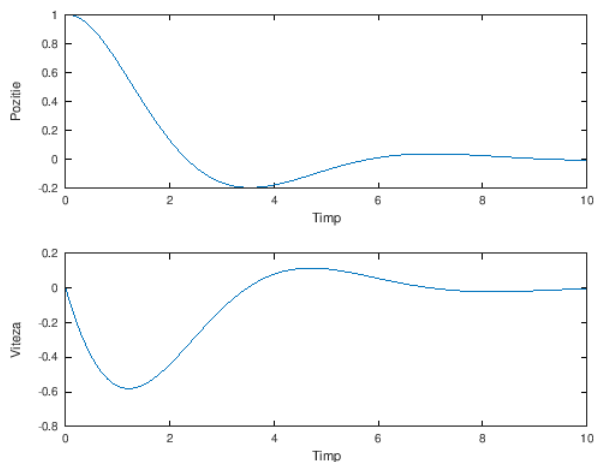
Fie $a = 2\zeta\omega_n$ și $b = \omega_n^2$.

```
In [82]: a = 1; # 2*zeta*omega_n
b = 1; # omega_n^2
F = 0; # forta aplicată corpului
N = 100; # numarul de puncte in care se rezolvă ecuatia
t = linspace(0,10,N); # intervalul de timp pe care se rezolva ecuatia
x = zeros(2,N); # soluția sistemului
x0 = [1;0]; # soluția inițială
```

```
In [83]: A = [0,1;-b,-a];
B = [0;F];
```

```
In [84]: delta_t = 10/N;
x(:,1) = x0;
for i=2:N
    x(:,i) = x(:,i-1) + delta_t.*(A*x(:,i-1) + B);
end
x_euler = x;
```

```
In [85]: subplot(2,1,1)
plot(t,x(1,:))
xlabel('Timp')
ylabel('Pozitie')
subplot(2,1,2)
plot(t,x(2,:))
xlabel('Timp')
ylabel('Viteza')
```



2. Utilizarea transformatei Laplace și a exponențialei matriceale

Sisteme omogene

Fie sistemul de ecuații diferențiale liniare $\dot{\mathbf{x}} = A\mathbf{x}$. Aplicând transformata Laplace ecuației se obține:

$$\mathcal{L}(\dot{\mathbf{x}}) = \mathcal{L}(A\mathbf{x}) = s\mathbf{x}(s) - \mathbf{x}(0) = A\mathbf{x}(s)$$

De unde rezultă:

$$(sI - A)\mathbf{x}(s) = \mathbf{x}(0)$$

Sau:

$$\mathbf{x}(s) = (sI - A)^{-1}\mathbf{x}(0)$$

Aplicând transformata Laplace inversă \mathcal{L}^{-1} rezultă:

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0)$$

Unde e^{At} este exponențiala matriceală și se calculează pornind de la exprimarea funcției exponențiale ca o serie infinită:

$$e^{At} = I + At + \frac{(At)^2}{2} + \dots = \sum_{k=0}^{\infty} \frac{(At)^k}{k!}$$

Numeric, această sumă se calculează păstrând doar un număr finit de termeni. De exemplu dacă s-ar păstra doar primii doi termeni ai seriei atunci metoda de integrare s-ar reduce la metoda Euler.

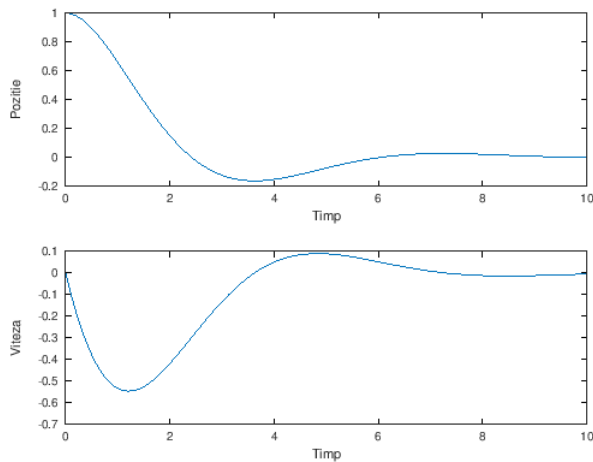
Exemplu:

```
In [15]: a = 1;
b = 1;
N = 100;
t = linspace(0,10,N);
x = zeros(2,N);
x0 = [1;0];
```

```
In [16]: A = [0,1;-b,-a];
```

```
In [17]: for i=1:N
exp_A_t = expm(A*t(i));
x(:,i) = exp_A_t*x0;
end
x_exp = x;
```

```
In [18]: subplot(2,1,1)
plot(t,x(1,:))
xlabel('Timp')
ylabel('Pozitie')
subplot(2,1,2)
plot(t,x(2,:))
xlabel('Timp')
ylabel('Viteza')
```



Exprimarea soluției $\mathbf{x}(t) = e^{At}\mathbf{x}(0)$ ca o relație de recurență

Rezolvarea sistemului de ecuații prin aplicarea directă a soluției $\mathbf{x}(t) = e^{At}\mathbf{x}(0)$ poate fi o operație dificilă datorită necesității de a calcula exponențiala matriceală e^{At_i} pentru fiecare t_i la care se dorește soluția. Această problemă se poate rezolva exprimând soluția sistemului de ecuații ca o relație de recurență. Această relație se poate obține pornind de la soluția obținută mai sus $\mathbf{x}(t) = e^{At}\mathbf{x}(0)$. Această expresie corespunde cazului când momentul inițial de la care este pornit procesul de integrare este zero $t_0 = 0$. Pentru cazul general, adică orice t_0 , soluția este:

$$\mathbf{x}(t) = e^{A(t-t_0)}\mathbf{x}(t_0)$$

Presupunând că intervalul $t - t_0$ pe care dorim să efectuăm procesul de integrare a sistemului de ecuații este Δt rezultă:

$$\begin{aligned}\Delta t &= t - t_0 \\ \mathbf{x}(t) &= e^{A\Delta t}\mathbf{x}(t - \Delta t)\end{aligned}$$

Sau:

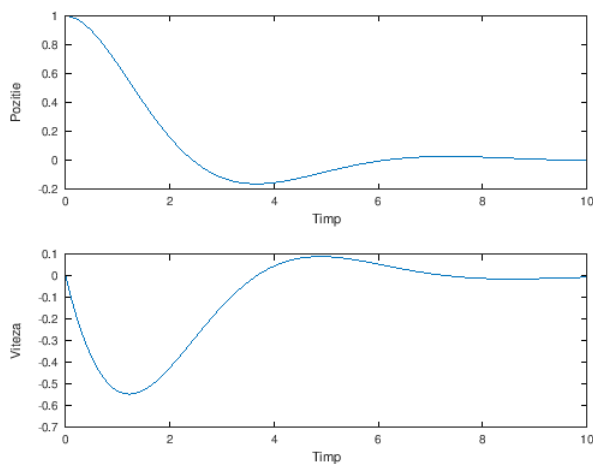
$$\mathbf{x}(t + \Delta t) = e^{A\Delta t}\mathbf{x}(t)$$

Se observă că exponențiala $e^{A\Delta t}$ nu mai depinde de variabila t , prin urmare aceasta poate fi calculată o singură dată la început.

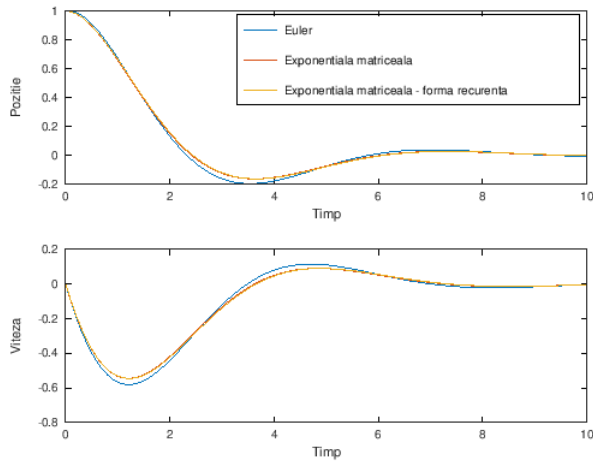
Exemplu:

```
In [19]: delta_t = 10/N;
exp_A_delta_t = expm(A*delta_t);
x(:,1) = x0;
for i=2:N
    x(:,i) = exp_A_delta_t*x(:,i-1);
end
x_exp_recurent = x;
```

```
In [20]: subplot(2,1,1)
plot(t,x(1,:))
xlabel('Timp')
ylabel('Pozitie')
subplot(2,1,2)
plot(t,x(2,:))
xlabel('Timp')
ylabel('Viteza')
```



```
In [21]: subplot(2,1,1)
plot(t,x_euler(1,:), t, x_exp(1,:), t, x_exp_recurent(1,:))
xlabel('Timp')
ylabel('Pozitie')
legend('Euler','Exponentiala matriceala', 'Exponentiala matriceala - forma recurenta')
subplot(2,1,2)
plot(t,x_euler(2,:), t, x_exp(2,:), t, x_exp_recurent(2,:))
xlabel('Timp')
ylabel('Viteza')
```



Sisteme neomogene

Fie sistemul de ecuații diferențiale liniare $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B(t)$. Aplicând transformata Laplace se obține:

$$\mathcal{L}(\dot{\mathbf{x}}) = \mathcal{L}(A\mathbf{x} + B(t)) = s\mathbf{x}(s) - \mathbf{x}(0) = A\mathbf{x}(s) + B(s)$$

De unde rezultă:

$$(sI - A)\mathbf{x}(s) = \mathbf{x}(0) + B(s)$$

$$\mathbf{x}(s) = (sI - A)^{-1}\mathbf{x}(0) + (sI - A)^{-1}B(s)$$

Aplicând transformata Laplace inversă \mathcal{L}^{-1} rezultă:

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0) + \int_0^t e^{A(t-\tau)}B(\tau)d\tau$$

Se poate obține o expresie recurentă a soluției prin aproximarea integralei din membrul drept cu o sumă:

$$\mathbf{x}(n\Delta t) = e^{An\Delta t}\mathbf{x}(0) + \sum_{k=0}^n e^{A(n\Delta t - k\Delta t)}B(k\Delta t)\Delta t$$

Separând ultimul termen al sumei rezultă:

$$\mathbf{x}(n\Delta t) = (e^{An\Delta t}\mathbf{x}(0) + \sum_{k=0}^{n-1} (e^{A(n-k)\Delta t}B(k\Delta t)\Delta t) + B(n\Delta t)\Delta t$$

Mărimea $e^{A\Delta t}$ se dă factor comun forțat între primii doi termeni, ca urmare se obține:

$$\mathbf{x}(n\Delta t) = e^{A\Delta t} \left(e^{A(n-1)\Delta t}\mathbf{x}(0) + \sum_{k=0}^{n-1} (e^{A(n-k-1)\Delta t}B(k\Delta t)\Delta t) \right) + B(n\Delta t)\Delta t$$

De unde se poate extrage formula de recurență:

$$\mathbf{x}(n\Delta t) = e^{A\Delta t}\mathbf{x}((n-1)\Delta t) + B(n\Delta t)\Delta t$$

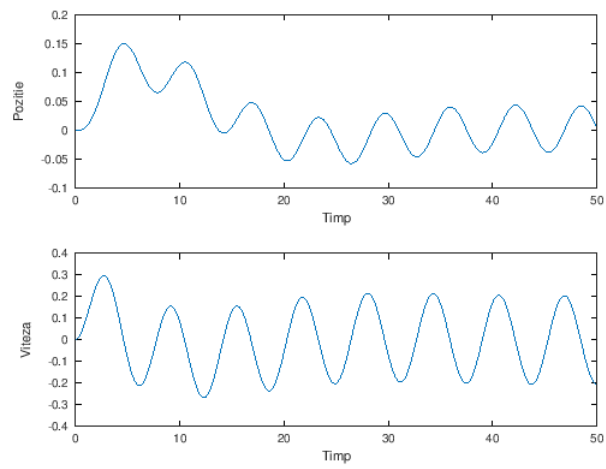
Exemplu:

```
In [93]: a = 1;
b = 1;
N = 500;
t = linspace(0,50,N);
x = zeros(2,N);
x0 = [0;0];

In [94]: A = [0,1;-b,-a];

In [95]: delta_t = 10/N;
exp_A_delta_t = expm(A*delta_t);
x(:,1) = x0;
for i=2:N
    B = [0; sin(t(i))];
    x(:,i) = exp_A_delta_t*x(:,i-1) + B*delta_t;
end
```

```
In [96]: subplot(2,1,1)
plot(t,x(1,:))
xlabel('Timp')
ylabel('Pozitie')
subplot(2,1,2)
plot(t,x(2,:))
xlabel('Timp')
ylabel('Viteza')
```

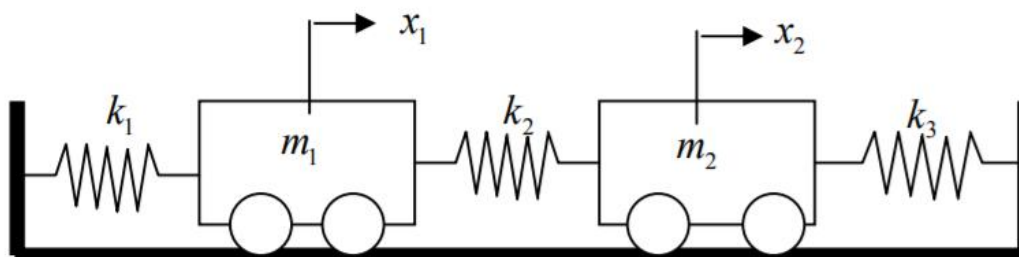


In []:

Metode numerice - Laborator 8

Rezolvarea ecuațiilor și a sistemelor de ecuații diferențiale

Se dă sistemul fizic cu două grade de libertate din figura de mai jos:



Si diferențiale ce descriu dinamica acestui sistem:

$$m_1 \ddot{x}_1 + (k_1 + k_2)x_1 - k_2 x_2 = F_1$$

$$m_2 \ddot{x}_2 - k_2 x_1 + (k_2 + k_3)x_2 = F_2$$

Unde: m_1 și m_2 reprezintă masele celor două corpuri, x_1 și x_2 reprezintă deplasările, k_1, k_2, k_3 reprezintă constantele cuplajelor elastice iar F_1 și F_2 reprezintă forțele aplicate celor două corpuri.

Valorile numerice pentru parametrii sistemului sunt:

$$m_1 = 1Kg, m_2 = 2Kg, k_1 = k_2 = k_3 = 1 \frac{N}{m}$$

Se cere:

1. Formulați sistemul de ecuații de mai sus ca un sistem de ecuații de ordinul 1 de forma:
 $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}$
2. Simulați sistemul pentru 10 secunde pentru condițiile enumerate mai jos și reprezentați pe grafic deplasările x_1 și x_2 în raport cu timpul. Pentru rezolvarea numerică a sistemului de ecuații puteți folosi orice metodă.

$$F_1 = F_2 = 0 N$$

$$x_1(0) = 0 m$$

$$\dot{x}_1(0) = 0 \frac{m}{s}$$

$$x_2(0) = 1 \text{ m}$$

$$\dot{x}_2(0) = 0 \frac{\text{m}}{\text{s}}$$

3. Simulați sistemul pentru 10 secunde pentru condițiile enumerate mai jos și reprezentați pe grafic deplasările x_1 și x_2 în raport cu timpul. Pentru rezolvarea numerică a sistemului de ecuații puteți folosi orice metodă.

$$F_1 = 1 \text{ N}$$

$$F_2 = 0$$

$$x_1(0) = x_2(0) = \dot{x}_1(0) = \dot{x}_2(0) = 0$$