

**Nicoleta Breaz, Marian Crăciun, Păstorel Gașpar,
Maria Miroiu, Iuliana Paraschiv-Munteanu**

**MODELAREA
MATEMATICĂ
PRIN
MATLAB**

Editura StudIS

adicenter@yahoo.com

Iasi, Sos. Stefan cel Mare, nr.5

Tel./fax: 0232 – 217.754

Descrierea CIP a Bibliotecii Naționale a României

COORDONATOR: MARIA MIROIU

Nicoleta Breaz, Marian Crăciun, Păstorel Gașpar, Iuliana Paraschiv-Munteanu.

Modelarea matematică prin Matlab / Maria Miroiu, Nicoleta Breaz, Marian Crăciun, Păstorel Gașpar, Iuliana Paraschiv- Munteanu - Vatra Dornei : StudIS, 2013

Bibliogr.

ISBN: 978-606-624-303-2

- I. Maria Miroiu
- II. Nicoleta Breaz
- III. Marian Crăciun
- IV. Păstorel Gașpar
- V. Iuliana Paraschiv- Munteanu

Consilier editorial: Dranca Adrian

Secretar editorial: Moroșanu Paul

Pre-press, tipar digital și finisare:

S.C. ADI CENTER SRL

Șos. Ștefan ce Mare, nr. 5

Tel.: 217 754



Copyright © 2013

Toate drepturile asupra acestei ediții sunt rezervate autorului

Cuprins

Prefață	5
Capitolul I - Calcul simbolic în Matlab	8
1.1 Definierea variabilelor și funcțiilor simbolice	8
1.2 Substituții	12
1.3 Reprezentarea simbolică a numerelor	15
1.4 Matrice simbolice	16
1.5 Reprezentarea grafică a funcțiilor	19
1.6 Derivarea funcțiilor simbolice	23
1.7 Limite de funcții simbolice	25
1.8 Integrarea funcțiilor simbolice	27
1.8.1 Calculul integralelor nedefinite	27
1.8.2 Calculul integralelor definite	27
1.8.3 Calculul integralelor ce depind de un parametru	29
1.9 Calculul sumelor simbolice	30
1.10 Dezvoltare în serie Taylor	31
1.11 Rezolvarea simbolică a ecuațiilor și sistemelor algebrice	32
1.12 Rezolvarea simbolică a ecuațiilor și sistemelor diferențiale ordinare	36
1.13 Exerciții propuse spre rezolvare	38
Capitolul II - Reprezentări grafice în Matlab	42
2.1 Reprezentarea grafică 2D	42
2.1.1 Reprezentarea grafică a funcțiilor reale de o variabilă reală	45
2.1.2 Reprezentarea în plan a curbelor date prin ecuații parametrice	49
2.1.3 Reprezentarea în plan a curbelor date prin ecuații polare	50
2.1.4 Reprezentarea în plan a curbelor date prin ecuații implicite	52
2.2 Reprezentarea grafică 3D	53
2.2.1 Reprezentarea grafică a funcțiilor reale de două variabile reale	53
2.2.2 Reprezentarea în spațiu a curbelor date prin ecuații parametrice	56
2.2.3 Primitive grafice pentru suprafețe	58
2.2.4 Reprezentarea în spațiu a curbelor date prin ecuații parametrice	59
2.2.5 Reprezentarea în spațiu a curbelor și suprafețelor date implicit	61
2.2.6 Reprezentarea în sistemele de coordonate sferice și cilindrice	62
2.3 Alte tipuri de reprezentări grafice	63
2.4 Utilizarea instrumentelor de desenare pentru editarea graficelor	66
2.5 Manipularea obiectelor grafice în Matlab	68
2.6 Exerciții propuse spre rezolvare	71

Capitolul III - Modele statistice	73
3.1 Problema regresiei	74
3.2 Modelul de regresie liniară simplă	75
3.2.1 Definirea modelului liniar simplu	75
3.2.2 Verificarea existenței unei corelații și identificarea modelului	77
3.2.3 Ajustarea modelului prin determinarea unor estimatori ai parametrilor săi	78
3.2.4 Ipotezele fundamentale-Gauss-Markov. Validarea și estimarea modelului	79
3.2.5 Utilizarea modelului pentru calcule de prognoză	80
3.2.6 Funcții specifice și aplicații privind modelul liniar simplu în Matlab	81
3.2.7 Teme de laborator în Matlab	87
3.3 Alte modele de regresie liniară	87
3.3.1 Modelul polinomial	87
3.3.2 Alte modele de regresie simplă neliniară	89
3.3.3 Analiza comparativă a modelelor de regresie	93
3.3.4 Funcții și aplicații privind alte modele de regresie simplă în Matlab	95
3.3.5 Teme de laborator în Matlab	103
3.4 Modele de regresie multiplă	104
3.4.1 Fundamentarea modelului de regresie multiplă	104
3.4.2 Funcții și aplicații privind modelul de regresie multiplă în Matlab	108
3.4.3 Teme de laborator în Matlab	114
Capitolul IV - Procese stochastice staționare aplicate în procesarea semnalelor	115
4.1 Semnale aleatoare și procese stochastice	115
4.1.1 Tipuri de procese stochastice	116
4.1.2 Caracteristici ale proceselor aleatoare	116
4.2 Procese staționare cu parametru discret de timp	117
4.3 Procese aleatoare la ieșirea dintr-un filtru liniar	120
4.4 Interpretarea desității spectrale	126
4.5 Filtrare Wiener	127
4.5.1 Netezire Wiener	129
4.5.2 Predicție	132
4.6 Exemplu concret – sintetizarea vocii	137
4.7 Exerciții și probleme propuse spre rezolvare	139
Capitolul V - Probleme de mecanică rezolvate în Matlab	142
5.1 Calcul vectorial	142
5.2 Probleme de cinematică	145
5.3 Mișcarea în câmp central	161
5.4 Mișcarea pe o curbă	178
Bibliografie	191

Prefață

Cartea de față a fost elaborată în cadrul proiectului POSDRU/56/1.2/S/32768, “Formarea cadrelor didactice universitare și a studenților în domeniul utilizării unor instrumente moderne de predare-învățare-evaluare pentru disciplinele matematice, în vederea creării de competențe performante și practice pentru piața muncii”.

Finanțat din Fondul Social European și implementat de către Ministerul Educației, Cercetării, Tineretului și Sportului, în colaborare cu The Red Point, Oameni și Companii, Universitatea din București, Universitatea Tehnică de Construcții din București, Universitatea „Politehnica” din București, Universitatea din Pitești, Universitatea Tehnică „Gheorghe Asachi” din Iași, Universitatea de Vest din Timișoara, Universitatea „Dunărea de Jos” din Galați, Universitatea Tehnică din Cluj-Napoca, Universitatea “1 Decembrie 1918” din Alba-Iulia, proiectul contribuie în mod direct la realizarea obiectivului general al Programului Operațional Sectorial de Dezvoltare a Resurselor Umane – POSDRU și se înscrie în domeniul major de intervenție 1.2 Calitate în învățământul superior.

Proiectul are ca obiectiv adaptarea programelor de studii ale disciplinelor matematice la cerințele pieței muncii și crearea de mecanisme și instrumente de extindere a oportunităților de învățare.

Evaluarea nevoilor educaționale obiective ale cadrelor didactice și studenților legate de utilizarea matematicii în învățământul superior, masterate și doctorate precum și analizarea eficacității și relevanței curriculelor actuale la nivel de performanță și eficiență, în vederea dezvoltării de cunoștințe și competențe pentru studenții care învață discipline matematice în universități, reprezintă obiective specifice de interes în cadrul proiectului. Dezvoltarea și armonizarea curriculelor universitare ale disciplinelor matematice, conform exigențelor de pe piața muncii, elaborarea și implementarea unui program de formare a cadrelor didactice și a studenților interesați din universitățile partenere, bazat pe dezvoltarea și armonizarea de curriculum, crearea unei baze de resurse inovative, moderne și funcționale pentru predarea-învățarea-evaluarea în disciplinele matematice pentru învățământul universitar sunt obiectivele specifice care au ca raspuns materialul de față.

Formarea de competențe cheie de matematică și informatică presupune crearea de abilități de care fiecare individ are nevoie pentru dezvoltarea personală, incluziune socială și inserție pe piața muncii. Se poate constata însă că programele disciplinelor de matematică nu au întotdeauna în vedere identificarea și sprijinirea elevilor și studenților potențial talentați la matematică. Totuși, studiul matematicii a evoluat în exigențe până a ajunge să accepte provocarea de a folosi noile tehnologii în procesul de predare-învățare-evaluare pentru a face matematica mai atractivă.

În acest context, analiza flexibilității curriculei, însoțită de analiza metodelor și instrumentelor folosite pentru identificarea și motivarea studenților talentați la matematică ar putea răspunde deopotrivă cerințelor de masă, cât și celor de elită.

Viziunea pe termen lung a acestui proiect preconizează determinarea unor schimbări în abordarea fenomenului matematic pe mai multe planuri: informarea unui număr cât mai mare de membri ai societății în legătură cu rolul și locul matematicii în educația de bază în instrucție și în descoperirile științifice menite să îmbunătățească calitatea vieții, inclusiv popularizarea unor mari descoperiri tehnice, și nu numai, în care matematica cea mai avansată a jucat un rol hotărâtor. De asemenea, se urmărește evidențierea a noi motivații solide pentru învățarea și studiul matematicii la nivelele de bază și la nivel de performanță; stimularea creativității și formarea la viitorii cercetători matematicieni a unei atitudini deschise față de însușirea aspectelor specifice din alte științe, în scopul participării cu succes în echipe mixte de cercetare sau a abordării unei cercetări inter și multi disciplinare; identificarea unor forme de pregătire adecvată de matematică pentru viitorii studenți ai disciplinelor matematice, în scopul utilizării la nivel de performanță a aparatului matematic în construirea unei cariere profesionale.

Lucrarea de față se adresează studenților în special din învățământul superior tehnic, cât și tuturor celor interesați de studiul diverselor modele matematice și implementări ale acestora în Matlab.

Conceptul de „model”, atât de des folosit în știința modernă, este relativ nou, dar metoda modelării este tot atât de veche pe cât sunt preocupările oamenilor pentru cunoașterea științifică. Putem considera că modelul este o reprezentare izomorfă a realității, care, oferind o imagine intuitivă și totuși riguroasă, în sensul structurii logice, a fenomenului studiat, facilitează descoperirea unor legături imposibil sau foarte greu de găsit pe alte căi. Preocupările legate de modelarea matematică sunt justificate de utilizarea acestora în foarte multe domenii de activitate.

Capitolul I intitulat „Calcul simbolic în Matlab” a fost elaborat de Maria Miroiu de la Universitatea din Pitești și prezintă diverse funcții Matlab din Symbolic Math Toolbox și legătura lor cu derivabilitatea, integrarea, calculul limitelor, dezvoltările în serie Taylor, rezolvările de ecuații și sisteme algebrice, respectiv diferențiale, etc. În calculele ingineresti se utilizează frecvent expresii simbolice, cel puțin în calculele numerice cu matrice și analiză numerică.

Titlul celui de-al doilea capitol este „Reprezentări grafice în Matlab”, acesta fiind realizat de Marian Viorel Crăciun de la Universitatea „Dunărea de jos” din Galați. Aici sunt prezentate principalele facilități puse la dispoziție de mediul Matlab pentru reprezentarea grafică 2D și 3D.

Al treilea capitol este intitulat „Modele statistice” și a fost elaborat de Marcela Nicoleta Breaz de la Universitatea „1 Decembrie 1918” din Alba Iulia. Aici sunt tratate atât elementele teoretice pentru fundamentarea unor modele statistice precum regresia, cât și suportul computațional asigurat de pachete specializate din Matlab. De asemenea, pentru o înțelegere mai completă a realității sunt tratate și probleme din lumea reală la care sunt date și implementările acestora în Matlab.

Capitolul al patrulea, cu titlul „Procese stochastice staționare aplicate în procesarea semnalelor” a fost elaborat de Octavian Păstorel Gașpar de la Universitatea „Aurel Vlaicu” din Arad și prezintă diverse modele probabiliste precum procese aleatoare la ieșirea dintr-un filtru liniar și filtrarea Wiener. De asemenea, se tratează și un exemplu concret și anume sintetizarea vocii.

Ultimul capitol intitulat „Probleme de mecanică rezolvate în Matlab” a fost elaborat de Iuliana Paraschiv-Munteanu de la Universitatea din București și cuprinde diverse probleme de mecanică tratate întâi teoretic cu legile mecanicii, apoi prin programe Matlab. Prin aceste probleme s-a încercat ilustrarea unor procese de modelare matematică a unor fenomene fizice, prezum și simularea și vizualizarea pe calculator a desfășurării și evoluției acestor fenomene.

Parcurgând cartea, se poate observa că modelele matematice sunt des utilizate în diverse domenii de activitate, aceasta datorită în special a capacității acestora de a condensa riguros esențialul, cât și posibilității lor de a fi programate cu ajutorul calculatoarelor electronice, alcătuind împreună un instrument de investigație științifică de o putere necunoscută până în prezent, o prodigioasă „prelungire” a inteligenței umane.

Autorii

Capitolul 1

Calcul simbolic în MATLAB

Acest capitol are ca scop prezentarea noțiunilor despre calcul simbolic în MATLAB.

În calcule ingineresti se utilizează frecvent expresii simbolice cel puțin în calculele numerice cu matrice și de analiză numerică.

MATLAB folosește **Symbolic Math Toolbox** care este un grup de instrumente matematice simbolice folosit pentru crearea și operarea cu expresii matematice simbolice. Acesta conține sute de funcții simbolice MATLAB utilizate pentru derivabilitate, integrare, simplificări, transformări, rezolvări de ecuații, etc.

1.1 Definirea variabilelor și funcțiilor simbolice

Calculul simbolic operează cu obiecte simbolice: variabile, matrice și expresii simbolice. *Variabilele simbolice se definesc* în două feluri:

- cu instrucțiunea `sym` se poate defini o variabilă simbolică, respectiv cu instrucțiunea `syms` se pot defini mai multe variabile simbolice;
- orice variabilă ce primește ca valoare o expresie simbolică (prin folosirea operatorului `=`) devine o variabilă simbolică.

Instrucțiunea `sym` are forma

```
sym var
```

unde `var` este un nume de variabilă simbolică.

Instrucțiunea `syms` are forma

```
syms var1 var2 var3 ...
```

unde `var1`, `var2`, `var3`, ... sunt nume de variabile simbolice.

Exemplul 1.1. Să se definească variabilele simbolice a , b și c cu instrucțiunea `syms` și să se inițializeze variabila a cu expresia simbolică $b + c$.

Soluție:

```
>> syms a b c %a,b,c sunt variabile simbolice  
>> a = b + c  
a =  
b + c
```

Să considerăm în continuare instrucțiunea

```
>> d = b + c %b,c = var.simbolice => m var.simbolica  
d =  
b + c
```

Astfel d este de asemenea o variabilă simbolică, deoarece a primit ca valoare o expresie simbolică.

Expresiile simbolice se definesc la fel ca expresiile aritmetice din Matlab, în care variabilele numerice sunt înlocuite cu variabile simbolice. Se includ aici și vectorii și matricele cu elemente expresii simbolice, asupra cărora se pot efectua calcule simbolice similare calculelor numerice din Matlab.

Expresiile simbolice au ca termeni variabile simbolice și funcții simbolice, iar ca operatori aritmetici: +, -, *, /, \, ^. Operatorii +, -, *, /, sunt cei patru operatori aritmetici cunoscuți. Operația $a \setminus b$ se definește ca $a^{-1}b$. Operatorul ^ este ridicarea la putere (expresia a^b este a^b).

Prioritățile operatorilor aritmetici sunt cele cunoscute. Operatorii + și - unari au cea mai mare prioritate, urmați de operatorii *, /, \, ^, iar operatorii + și - binari au cea mai mică prioritate. Operatorii unari sunt asociativi la dreapta, operatorii binari sunt asociativi este la stânga. În cadrul expresiilor aritmetice se pot utiliza paranteze rotunde, (și), pentru a modifica prioritățile operatorilor.

Funcțiile matematice uzuale ce se pot utiliza în expresiile simbolice sunt cele de mai jos:

```
sin    asin  sqrt  exp
cos    acos  abs   log
tan    atan           log10
```

Există încă un mod de a crea expresii simbolice folosind funcția sym. Această funcție poate să mai aibă forma

```
variabila_simbolica = sym('expresie_simbolica')
```

și are ca rezultat o expresie simbolică ce poate fi atribuită unei variabile simbolice. Reamintim că o expresie simbolică are ca termeni variabile simbolice, definite anterior.

Exemplul 1.2. Să se definească funcția simbolică liniară $f(x) = ax^2 + bx + c$.

Soluție:

O primă modalitate de a defini funcția este de a defini variabilele simbolice a , x , b și f cu instrucțiunea syms și apoi atribuim lui f ca valoare expresia simbolică:

```
>> syms a b c x
>> f = a * x ^ 2 + b * x + c
f =
a*x^2 + b*x + c
```

Al doilea mod definește expresia simbolică cu funcția sym:

```
>> syms a b c x
>> f = sym('a * x ^ 2 + b * x + c');
```

În unele cazuri trebuie să creăm expresii simbolice în care intervin constante. Acest lucru se face cu funcția sym. De exemplu, pentru a crea o funcție simbolică constantă ce are valoarea 3 vom scrie

```
>> f = sym('3');
```

sau

```
>> f = sym(3);
```

Reamintim că instrucțiunea

```
>> f = 3;
```

crează o variabilă numerică ce are valoarea 3.

Exemplul 1.3. Să se definească variabilele simbolice a , b și x și polinoamele simbolice $f(x) = ax + 3$ și $g(x) = x + b$ și să se calculeze produsul lor, $f(x) \cdot g(x)$.

Soluție:

Secvența de cod este următoarea

```
>> syms a b x
>> f = a * x + 3
f =
a*x+3

>> g = x + b
g =
x+b

>> f * g
ans =
(a*x+3)*(x+b)
```

Exemplul 1.4. Să se definească variabila simbolică x și funcția simbolică $f(x) = \sqrt{x^2 + x + 1}$.

Soluție:

```
>> syms x
>> f = sqrt(x^2 + x + 1)
f =
(x^2 + x + 1)^(1/2)
```

Definirea de variabile simbolice reale și complexe

O variabilă complexă este o pereche ordonată de două variabile reale. Fie x și y două variabile reale. O variabilă complexă este

$$z = x + i \cdot y,$$

unde x este partea reală, y este partea imaginară a variabilei z , iar $i = \sqrt{-1}$. Pentru a defini o variabilă complexă trebuie să definim mai întâi două variabile simbolice reale, corespunzând părții reale și celei imaginare a variabilei complexe, apoi se utilizează formula de mai sus în care variabila i este predefinită. Pentru aceasta se utilizează instrucțiunea `syms` cu opțiunea `real` sau funcția `sym` cu aceeași opțiune:

§ cu instrucțiunea `syms`

```
>> syms x y real
>> z = x + i * y
z =
x + i * y
```

cu funcția `sym`

```
>> x = sym('x','real')
x =
x
>> y = sym('y','real')
y =
y
>> z = x + i * y
z =
x + i*y
```

În expresiile cu numere complexe se pot utiliza funcțiile matematice standard: `real`, `imag`, `conj`, `abs`, etc.

Exemplul 1.5. Să se definească o variabilă simbolică complexă și să se calculeze conjugata, partea reală și modulul.

Soluție:

```
>> syms x y
>> z = x + i * y;
>> conj(z)

ans =
x-i*y

>> real(z)

ans =
x

>> abs(z)

ans =
(x^2+y^2)^(1/2)
```

Dacă se dorește ștergerea proprietății real a unei variabile declarată anterior cu instrucțiunea `syms` sau funcția `sym`, se poate instrucțiunea `syms` cu opțiunea `unreal`.

Exemplul 1.6. Să se șteargă proprietatea real a unei variabilei `x` declarată ca fiind reală.

Soluție:

```
>> sym x real;
...
>> sym x unreal;
```

Crearea de funcții abstracte

În multe cazuri este avantajos să se creeze o funcție nedeterminată (abstractă) care acționează asupra obiectelor simbolice și apoi poate fi utilizată în expresii. Acest lucru se poate face cu instrucțiunea

```
>> f = sym('f(x)')
f =
f(x)
```

Variabila simbolică f poate fi utilizată apoi în expresii simbolice. De exemplu, pentru a crea expresia simbolică

$$\frac{f(x+h) - f(x)}{h}$$

se poate scrie

```
>> syms x h
>> f = sym('f(x)');
>> df = (subs(f, x, x+h) - f)/h
df =
(f(h + x) - f(x))/h
```

1.2 Substituții

Este posibil să atribuim valori numerice variabilelor simbolice și, în acest caz, expresiile simbolice sunt evaluate la valoarea numerică corespunzătoare. Aceste substituții ale variabilelor simbolice cu valori numerice se fac cu funcția `subs` cu formele

```
subs(expresie_simbolica, variabilă_simbolica, valoare_numerică)
```

în cazul când substituim o singură variabilă, și

```
subs(expresie_simbolica, {variabile_simbolice}, {valori_numerice})
```

în cazul când substituim mai multe variabile.

Exemplul 1.7. Fie funcția simbolică $f(x) = x^2 - 5x + 4$. Să se substituie variabila simbolică x întâi cu valoarea 4, apoi cu valoarea -2 .

Soluție:

```
>> syms f x
>> f = x^2 - 5 * x + 4
f =
x^2 - 5*x + 4

>> subs(f, x, 4)
ans =
    0

>> subs(f, x, -2)
ans =
    18
```

Exemplul 1.8. Fie funcția de două variabile $f(x, y) = x^2 + xy$. Să se substituie întâi variabila simbolică x cu valoarea 3, apoi variabila simbolică y cu valoarea 2 și în final variabila x cu valoarea -2 și variabila y cu 1.

Soluție:

```
>> syms x y
>> f = x^2 + x * y
f =
x^2 + y*x
>> subs(f, x, 3)

ans =
3*y + 9

>> subs(f, y, 2)

ans =
x^2 + 2*x

>> subs(f, {x, y}, {-2, 1})

ans =
2
```

Reamintim că, lista cu variabilele simbolice ce vor fi substituite și lista cu valori sunt incluse între acolade, { }.

În cazul în care se substituie o singură variabilă simbolică cu o valoare, funcția `subs` are și următoarea formă simplă

```
subs(expresie_simbolica, valoare_numerica)
```

Dacă expresia simbolică depinde de *mai mult de o variabilă și variabila pentru care se face substituția nu este specificată*, substituția se face pentru variabila simbolică implicită, care se alege după următoarea **regulă**: *se alege litera cea mai apropiată de x din alfabet; dacă există două litere egal depărtate de x , se alege ultima din alfabet din cele două*.

Exemplul 1.9. Fie expresia simbolică $f(x, t) = t^2 + \sin(x + t)$. Să se substituie variabila simbolică x cu valoarea 3.

Soluție:

```
>> syms x t
>> f = t^2 + sin(x + t);
f =
t^2 + sin(t + x)
>> subs(f, 3)

ans =
t^2 + sin(3+t)
```

După cum se observă, se substituie variabila simbolică x cu valoarea 3, aleasă după regula anterioară.

Variabilele simbolice dintr-o expresie se pot determina cu funcția `findsym`.

Exemplul 1.10. Să se determine variabilele simbolice din expresiile simbolice:
 $f(x) = \cos(\omega t + \psi)$, respectiv $g(x) = x^2 - 3x + 5$.

Soluție:

```
>> syms t omega psi x
>> f = cos(omega * t + psi)
f =
cos(psi + omega*t)

>> g = x^2 - 3*x +5
g =
x^2 - 3*x + 5

>> findsym(f)
ans =
omega, psi, t

>> findsym(g)
ans =
x
```

Variabila simbolică implicită dintr-o expresie f se poate determina cu funcția

findsym(f, 1)

Exemplul 1.11. Să se determine variabila simbolică implicită din expresia simbolică $t^2 \cdot \cos(x+t)$.

Soluție:

```
>> syms x t
>> f = t^2 * cos(x + t);
>> findsym(f, 1)
ans =
x
```

Funcțiile de mai jos aplică diverse identități asupra expresiilor simbolice.

Funcția `expand` cu forma

expand(f)

aplică identitățile din tabelul 1.1 asupra polinoamelor, funcțiilor exponențiale și logaritmice, funcțiilor trigonometrice directe și inverse.

f	expand(f)
$a*(x + y)$	$a * x + a * y$
$a*(x - y)$	$a * x - a * y$
$a^{(x+y)}$	$a^x * a^y$
$a^{(x-y)}$	a^x / a^y
$\exp(a + b)$	$\exp(a) * \exp(b)$
$\exp(a-b)$	$\exp(a)/\exp(b)$
$\log(x*y)$	$\log(x) + \log(y)$
$\log(x/y)$	$\log(x) - \log(y)$
$\log(a^y)$	$y*\log(a)$
$\sin(a + b)$	$\sin(a)*\cos(b) + \cos(a)*\sin(b)$
$\sin(2 * \text{asin}(a))$	$2*a*(1-a^2)^{(1/2)}$

Tabelul 1.1. *Expandarea expresiilor simbolice.*

- la expandarea logaritmilor argumentele trebuie să fie pozitive;
- funcția `expand` poate expanda toate funcțiile trigonometrice directe, $\sin(a+b)$, $\cos(a+b)$, $\tan(a+b)$;
- funcția `expand` poate expanda toate funcțiile trigonometrice inverse, $\sin(k*\text{asin}(x))$, $\cos(k*\text{acos}(x))$, $\tan(k*\text{atan}(x))$, etc., unde k este un număr natural.

Funcția `simplify` cu forma

`simplify(f)`

aplică identitățile din tabelul 1.2 asupra expresiilor simbolice.

f	<code>simplify(f)</code>
$\sin(x)^2 + \cos(x)^2$	1
$a^x * a^y$	$a^{(x+y)}$
a^x / a^y	$a^{(x-y)}$
$\exp(a) * \exp(b)$	$\exp(a + b)$
$\exp(a) / \exp(b)$	$\exp(a - b)$
$\exp(\log(x))$	x
$\log(\exp(x))$	x

Tabelul 1.2. *Simplificarea expresiilor simbolice.*

1.3 Reprezentarea simbolică a numerelor

Numerele se reprezintă simbolic în următoarele feluri:

- § în virgulă mobilă, sub forma unei mantise înmulțită cu 2 la un anumit exponent; mantisa conține 13 cifre hexazecimale,
- § ca număr rațional, ca raport a două numere întregi,
- § ca număr rațional plus precizia relativă; precizia relativă este diferența între valoarea numărului rațional și valoarea lui reprezentată în virgulă mobilă, și este exprimată în funcție de `eps`, precizia reprezentării numerelor în virgulă mobilă,
- § ca număr zecimal cu un număr specificat de cifre semnificative; implicit, numărul de cifre este 32, dar el poate fi specificat cu funcția `digits` după cum se arată mai jos.

Conversia între număr și reprezentarea sa simbolică se face cu funcția `sym` cu un parametru ce specifică tipul reprezentării simbolice după cum se arată în tabelul 1.3. În tabelul 1.3, în expresia funcției `sym`, x este o variabilă Matlab.

Reprezentarea simbolică	Funcția <code>sym()</code>
în virgulă mobilă	<code>sym(x, 'f')</code>
ca număr rațional	<code>sym(x, 'r')</code>
ca număr rațional plus precizia relativă	<code>sym(x, 'e')</code>
ca număr zecimal cu un număr specificat de cifre semnificative	<code>sym(x, 'd')</code>

Tabelul 1.3. *Reprezentarea simbolică a numerelor.*

Operațiile cu numere raționale se fac după regulile din aritmetică.

Exemplul 1.12. Fie expresia $\frac{1}{2} + \frac{3}{8}$. Să se aducă la același numitor și să se facă simplificările necesare folosind expresii simbolice.

Soluție:

```
sym(1)/sym(2) + sym(3)/sym(8)

ans =
7/8
```

Exemplul 1.13. Să se convertească numărul 0.2 într-un număr simbolic în toate reprezentările posibile.

Soluție:

```
>> sym(0.2, 'f')
ans =
(2^-3+2702159776422298*2^-55)

>> sym(0.2, 'r')
ans =
1/5

>> sym(0.2, 'e')
ans =
1/5 + eps/20

>> sym(0.2, 'd')
ans =
.200000000000000001110223024625157

>> digits(10)           %10 cifre dupa virgula
>> sym(0.2, 'd')
ans =
.2
```

1.4 Matrice simbolice

Matricele simbolice au ca elemente obiecte simbolice, constante, variabile sau expresii simbolice. Operațiile cu matrice se efectuează cu operatorii +, -, *, /, \, ^, .*, ./, \ și .^. Semnificația operatorilor +, -, * este cea de adunare, scădere și înmulțire a matricelor cu dimensiuni corespunzătoare. Semnificația celorlalți operatori este cea tabelul 1.4.

Operator	Exemplu	Semnificație
/	a/b	$a * \text{inv}(b)$
\	a\b	$\text{inv}(a) * b$
^	a^b	a^b
.*	c = a .* b	$c_{ij} = a_{ij} * b_{ij}$
./	c = a ./ b	$c_{ij} = a_{ij} / b_{ij}$
.\	c = a .\ b	$c_{ij} = b_{ij} * a_{ij}$
.^	c = a .^ b	$c_{ij} = a_{ij} ^ b_{ij}$

Tabelul 1.4. Operații cu matrice simbolice.

Elementele matricelor simbolice se adresează cu indici în același fel ca elementele matricelor numerice. Putem defini matrice simbolice în două feluri:

- matrice cu elemente simbolice;
- să convertim matrice numerice în matrice simbolice cu funcția `sym`.

Exemplul 1.14. Să se definească matricea cu elemente simbolice $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ și să

se calculeze determinantul și inversa acesteia.

Soluție:

```
>> syms a b c d
>> M = [a b; c d]
M =
[ a, b]
[ c, d]

>> det(M)
ans =
a*d-b*c

>> inv(M)
ans =
[ d/(a*d-b*c), -b/(a*d-b*c)]
[ -c/(a*d-b*c), a/(a*d-b*c)]

>> R = M * inv(M)
R =
[ a*d/(a*d-b*c)-b*c/(a*d-b*c), 0]
[ 0, a*d/(a*d-b*c)-b*c/(a*d-b*c)]

>> R = inv(M) * M
R =
[ a*d/(a*d-b*c)-b*c/(a*d-b*c), 0]
[ 0, a*d/(a*d-b*c)-b*c/(a*d-b*c)]
```

Pentru a afișa o expresie simbolică într-un format apropiat de cel din matematică se poate utiliza funcția `pretty` al cărei parametru este chiar expresia simbolică.

```
>> pretty(R)

[  a d      b c      ]
[----- - -----]
[ a d - b c  a d - b c ]
[                ]
[                a d      b c      ]
[ 0                ----- - -----]
[                a d - b c  a d - b c ]
```

Se pot simplifica expresiile de mai sus cu funcția `simplify` și se obține matricea unitate.

```
>> simplify(M*inv(M))
ans =
[ 1, 0]
[ 0, 1]
```

Exemplul 1.15. Să se definească matricele simbolice $\begin{bmatrix} a & 0 \\ b & c \end{bmatrix}$ și $\begin{bmatrix} -a & b \\ -c & d \end{bmatrix}$ și

să se calculeze suma și produsul lor.

Soluție:

```
>> syms a b c d
>> A = [a 0; b c]
A =
[ a, 0]
[ b, c]

>> B = [-a b; -c d]
B =
[ -a,  b]
[ -c,  d]

>> R = A+B
R =
[ 0,  b]
[ b-c, c+d]

>> R = A*B
R =
[ -a^2,  a*b]
[ -a*b-c^2,  b^2+c*d]

>> pretty(R) %afisarea intr-o forma matematica

[      2          ]
[  -a          a b  ]
[                ]
[      2      2    ]
[ -a b - c    b  + c d  ]
```

Exemplul 1.16. Să se definească matricea simbolică cu elemente complexe $\begin{bmatrix} a+i*b & c \\ -c & a-i*b \end{bmatrix}$ și să se calculeze inversa și conjugata acesteia.

Soluție:

```
>> syms a b c real
>> A = [a+i*b c; -c a-i*b]
A =
[ a+i*b,  c]
[ -c,  a-i*b]

>> det(A)
ans =
a^2+b^2+c^2

>> conj(A)
ans =
[ a-i*b,  c]
[ -c,  a+i*b]
```

Exemplul 1.17. Să se considere matricea Hilbert de dimensiune trei și să se convertească această matrice într-o matrice simbolică (calculul cu numere simbolice raționale se face cu precizie infinită) și să se calculeze determinantul și inversa acesteia.

Soluție:

```
>> H = hilb(3)
H =
    1.0000    0.5000    0.3333
    0.5000    0.3333    0.2500
    0.3333    0.2500    0.2000

>> H = sym(H)
H =
[ 1, 1/2, 1/3]
[ 1/2, 1/3, 1/4]
[ 1/3, 1/4, 1/5]

>> det(H)
ans =
1/2160

>> inv(H)
ans =
[ 9, -36, 30]
[ -36, 192, -180]
[ 30, -180, 180]
```

1.5 Reprezentarea grafică a funcțiilor

Reprezentarea grafică a funcțiilor se face cu funcția `ezplot`.

Reprezentarea grafică a unei funcții de o variabilă $f(x)$ se face cu funcția

```
ezplot(f, [min, max])
```

unde `min` și `max` sunt limitele domeniului. Dacă limitele sunt omise, domeniul de reprezentare este $-2\pi < x < 2\pi$.

Exemplul 1.18. Să se reprezinte grafic funcția $f(x) = \frac{x^2 - 2}{x + 1}$

Soluție:

```
>> syms x
>> f = (x^2-2)/(x+1);
>> ezplot(f), grid
```

Reprezentarea grafică a funcției este cea din Figura 1.1.

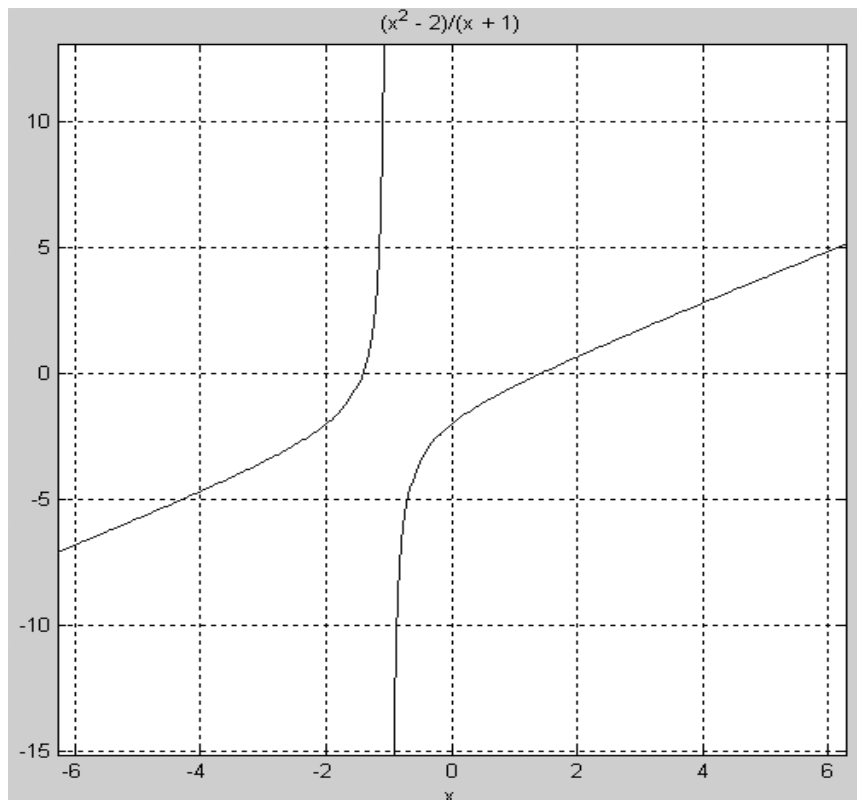


Figura 1.1. Reprezentarea grafică a unei funcții.

Curbele plane definite parametric prin formulele simbolice $x = x(t)$ și $y = y(t)$ se reprezintă grafic cu funcția

`ezplot(x, y, [tmin, tmax])`

unde t_{\min} , t_{\max} sunt limitele domeniului de reprezentare. Dacă limitele sunt omise, domeniul de reprezentare este $0 < t < 2\pi$.

Exemplul 1.19. Să se reprezinte grafic curba definită parametric prin ecuațiile parametriche:

$$\begin{aligned}x &= 5 \sin(3t) \\ y &= 2 \cos(3t)\end{aligned}$$

Soluție:

```
>> syms t
>> x = 5 * sin(3 * t);
>> y = 2 * cos(3 * t);
>> ezplot(x, y), grid
```

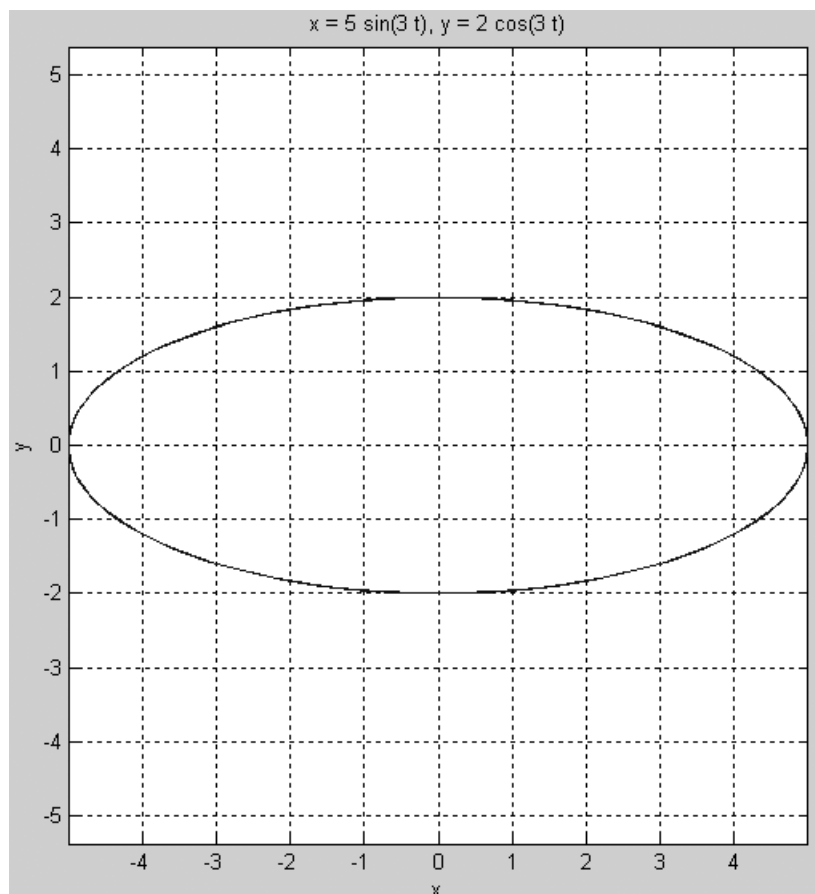


Figura 1.2. Graficul unei curbe definite parametric.

Curba este reprezentată în Figura 1.2. Variabila independentă t ia valori în intervalul $0 < t < 2\pi$.

Curbele plane definite implicit de relația $f(x, y) = 0$ se reprezintă grafic cu funcția

`ezplot(f, [xmin, xmax, ymin, ymax])`

unde $xmin, xmax, ymin, ymax$ sunt limitele domeniului de reprezentare. Funcția

`ezplot(f, [min, max])`

reprezintă grafic funcția $f(x, y) = 0$ pe domeniul $min < x < max, min < y < max$. În cazul în care domeniul nu este precizat, funcția

`ezplot(f)`

reprezintă grafic funcția $f(x, y) = 0$ pe domeniul $-2\pi < x < 2\pi, -2\pi < y < 2\pi$.

Exemplul 1.20. Să se reprezinte grafic funcția $2x^3 + 3y^4 - x + y = 0$.

Soluție:

Secvența de cod Matlab este următoarea

```
>> syms x y
>> f = 2*x^3 + 3*y^4 - x + y

f =
2*x^3 - x + 3*y^4 + y

>> ezplot(f), grid
```

Funcția este reprezentată în figura 1.3.

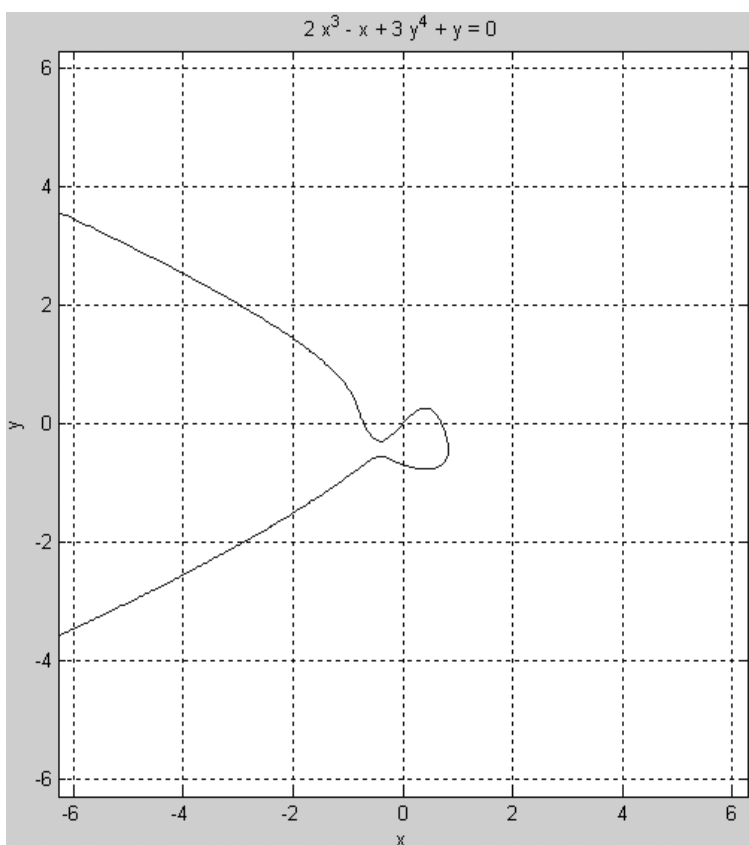


Figura 1.3. *Reprezentarea unei curbei implicite.*

Alte exemple de reprezentări grafice vor fi prezentate în paragrafele următoare.

1.6 Derivarea funcțiilor simbolice

Calculul derivatei simbolice se face cu funcția `diff`, care poate calcula derivatele funcțiilor de una sau mai multe variabile, derivate de ordin 1 sau de ordin superior, etc. Funcția are următoarele forme:

§ pentru a calcula derivata de ordin întâi a unei funcții în raport cu variabila x

`diff(f, x)`

§ pentru a calcula derivata de ordin n a unei funcții în raport cu variabila x

`diff(f, x, n)`

§ pentru a calcula derivata unei funcții în raport cu variabila implicită

`diff(f)`

variabila implicită se alege după regula enunțată la funcția `subs` în paragraful de mai sus referitor la substituții.

Exemplul 1.21. Să se determine derivatele de ordinul întâi ale funcțiilor:

$$\begin{aligned}f(x) &= \cos(2x) \\f(x, y) &= x \cdot \cos(y) - y \cdot \cos(x) \\f(x) &= e^{ix}.\end{aligned}$$

Soluție:

```
>> syms x
>> f = cos(2 * x);
>> diff(f)
ans =
(-2)*sin(2*x)
```

Așadar, $(\cos 2x)' = -2\sin(2x)$.

```
>> syms x y
>> f = x * cos(y) - y * cos(x);
>> diff(f, x), diff(f, y)
ans =
cos(y) + y*sin(x)
ans =
-x*sin(y) - cos(x)
```

Se poate verifica matematic că derivatele parțiale ale funcției $f(x, y) = x\cos(y) - y\cos(x)$ sunt:

$$\frac{\partial(x \cos y - y \cos x)}{\partial x} = \cos y + y \sin x$$

și

$$\frac{\partial(x \cos y - y \cos x)}{\partial y} = -x \sin y - \cos x$$

```
>> syms x
>> f = exp(i * x);
>> diff(f, x)
ans =
i*exp(i*x)
```

Derivata unei matrice simbolice este matricea obținută derivând fiecare element.

Exemplul 1.22. Să se calculeze derivata matricei simbolice

$$A = \begin{pmatrix} x^2 & xy \\ 2x & y^3 \end{pmatrix}$$

Soluție:

```
>> syms x y
>> A = [ x^2 x*y; 2*x y^3]

A =
[ x^2, x*y]
[ 2*x, y^3]

>> diff(A)

ans =
[ 2*x, y]
[ 2, 0]
```

Calculul iacobianului, $\frac{D(f,g,h,\mathbf{K})}{D(x,y,z,\mathbf{K})}$, se face cu funcția jacobian cu forma

jacobian([f; g; h; ...], [x, y, z, ...])

Primul argument al funcției jacobian este un vector coloană cu funcțiile f, g, h, \dots , iar al doilea argument este un vector linie cu variabilele x, y, z, \dots

Exemplul 1.23. Fie transformarea dată de funcțiile:

$$x = r * \cos t$$

$$y = r * \sin t$$

Să se determine în Matlab iacobianul transformării.

Soluție:

```
>> syms r t
>> x = r * cos(t);
>> y = r * sin(t);
>> D = jacobian([x; y], [r, t])

D =
[ cos(t), -r*sin(t)]
[ sin(t), r*cos(t)]
```

Determinantul transformării se calculează și se simplifică astfel

```
>> det(D)
ans =
cos(t)^2*r+r*sin(t)^2

>> simplify(det(D))
ans =
r
```


1.7 Limite de funcții simbolice

Pentru a calcula

$$\lim_{x \rightarrow a} f(x)$$

se poate folosi funcția `limit` cu forma

`limit(f, x, a)`

unde a este un număr, sau `inf` pentru ∞ (infini).

Funcția `limit` are și forma

`limit(f, a)`

caz în care variabila după care se calculează limita este variabila implicită care se alege după regula enunțată la funcția `subs` în paragraful referitor la substituții.

Exemplul 1.24. Să se calculeze în Matlab limita $\lim_{n \rightarrow \infty} \frac{2n^3 + 5n^2 - 1}{8n^3 - 2n + 3}$.

Soluție:

```
>> syms n
>> f = 2*n^3 + 5*n^2 - 1;
>> g = 8*n^3 - 2*n + 3;
>> limit(f / g, n, inf)
```

```
ans =
1/4
```

Exemplul 1.25. Să se calculeze în Matlab limita $\lim_{x \rightarrow 0} (1+x)^x$ (după cum se știe, această limită este e).

Soluție:

```
>> syms x
>> limit((1 + x)^(1 / x), x, 0)
```

```
ans =
exp(1)
```

Exemplul 1.26. Să se calculeze în Matlab limita $\lim_{x \rightarrow 0} \frac{1+e^x}{x^2}$.

Soluție:

```
>> syms x
>> f = (1+exp(x))/x^2
f =
(exp(x) + 1)/x^2
>> limit(f, x, 0)
ans =
Inf
```

Exemplul 1.27. Să se calculeze în Matlab limita $\lim_{x \rightarrow 0} \cos \frac{1}{x}$ (știm de la analiză matematică că această limită nu există).

Soluție:

```
>> syms x
>> limit(cos(1/x), x, 0)
ans =
limit(cos(1/x), x = 0)
```

Exemplul 1.28. Să se calculeze limita $\lim_{h \rightarrow 0} \frac{\cos(x) - \cos(x+h)}{h}$

Soluție:

```
>> syms x h
>> g = (cos(x)-cos(x+h))/h;
>> limit(g, h, 0)

ans =
sin(x)
```

Calculul limitelor laterale

Pentru a calcula limitele laterale

$$\lim_{\substack{x \rightarrow a \\ x > a}} f(x)$$

și

$$\lim_{\substack{x \rightarrow a \\ x < a}} f(x)$$

în Matlab, se poate folosi funcția `limit` cu formele

```
limit(f, x, a, 'right')
```

pentru limita la dreapta, și

```
limit(f, x, a, 'left')
```

pentru limita la stânga.

Exemplul 1.29. Să se calculeze limitele laterale ale funcției $f(x) = \frac{1}{x-1}$ în punctul $x = 1$.

Soluție:

```
>> syms x
>> f = 1 / (x - 1);
>> limit(f, x, 1, 'right')
ans =
Inf
>> limit(f, x, 1, 'left')
ans =
-Inf
```

1.8 Integrarea funcțiilor simbolice

1.8.1 Calculul integralelor nedefinite

Considerăm acum problema determinării integralei nedefinite a funcției f , $\int f(x)dx$. În Matlab, integrarea simbolică se efectuează cu funcția `int` cu formele:

`int(f, x)`

în care specificăm variabila de integrare, sau

`int(f)`

când variabila de integrare se alege după regula enunțată în paragraful referitor la substituții.

Exemplul 1.30. Să se calculeze în Matlab integrala nedefinită a funcției

$$f(x) = a \cdot \sin x + b \cdot \cos x.$$

Soluție:

```
>> syms a b x
>> f = a * sin(x) + b * cos(x)
f =
b*cos(x) + a*sin(x)

>> int(f, x)
ans =
b*sin(x) - a*cos(x)
```

1.8.2 Calculul integralelor definite

Calculul integralei definite a funcției f

$$\int_a^b f(x)dx$$

se poate face în Matlab cu funcția

`int(f, x, a, b)`

sau cu funcția

`int(f, a, b)`

În ultimul caz, variabila de integrare se alege după regula prezentată în paragraful referitor la substituții.

Exemplul 1.31. Să se calculeze integrala definită $\int_1^3 (x^3 + x + 2)dx$

Soluție:

```
>> syms x
>> f = x^3 + x + 2;
>> int(f, 1, 3)
```

```
ans =
28
```

Exemplul 1.32. Să se calculeze integrala improprie $\int_{-\infty}^{\infty} e^{-\frac{x^2}{2}} dx$ (a cărei valoare

știm că este $\sqrt{2p}$) și să se reprezinte grafic funcția din interiorul integralei.

Soluție:

```
>> syms x
>> f = exp(- x^2 / 2)
f =
1/exp(x^2/2)

>> ezplot(f), grid
>> int(f, x, -inf, inf)
```

```
ans =
2^(1/2)*pi^(1/2)
```

Reprezentarea grafică a funcției este cea din Figura 1.5.

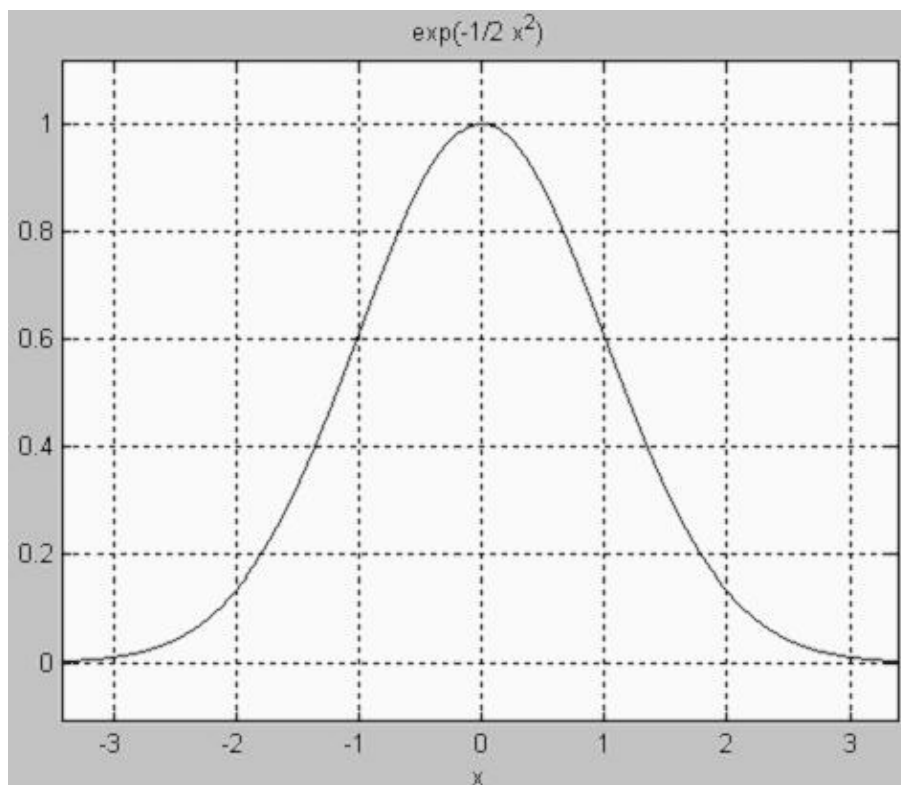


Figura 1.5. Graficul funcției $f(x) = \exp(-x^2/2)$.

1.8.3 Calculul integralelor ce depind de un parametru

Fie o funcție f ce depinde de un parametru. Dacă parametrul nu are o valoare specificată, el este presupus ca fiind un număr complex. În cazul în care parametrul este un număr real sau un număr pozitiv, el trebuie declarat ca atare în instrucțiunea `syms`.

Exemplul 1.33. Să se calculeze integrala improprie $\int_{-\infty}^{\infty} e^{-ax^2} dx$, unde $a > 0$.

Soluție:

Cum $a > 0$ se poate demonstra matematic că integrala este convergentă. Se poate declara că parametrul a este pozitiv folosind instrucțiunea `syms`:

```
>> syms a positive
>> syms x
>> f = exp(-a * x^2)

f =
1/exp(a*x^2)

>> R = int(f, x, -inf, inf)

R =
1/a^(1/2)*pi^(1/2)

>> pretty(R)

      1/2
     pi
    -----
      1/2
     a
```

În cazul în care parametrul a este complex vom scrie

```
>> syms a x
>> f = exp(-a * x^2);
>> R = int(f, x, -inf, inf)

R =
piecewise([abs(Im(a^(1/2))) <= abs(Re(a^(1/2)))] and
Re(a^(1/2)) <> 0,
(pi^(1/2)*sign(Re(a^(1/2))))/a^(1/2)],
[abs(Re(a^(1/2))) < abs(Im(a^(1/2)))] and
sign(a)^(1/2) in {i, -i}, Inf], [abs(arg(a)) <=
1/2*pi and a <> 0, pi^(1/2)/a^(1/2)])
```

1.9 Calculul sumelor simbolice

Pentru calculul simbolic al sumei

$$\sum_{k=m}^{k=n} x(k)$$

se poate folosi instrucțiunea Matlab `symsum` cu formele

`symsum(x, v, m, n)`

sau

`symsum(x, m, n)`

unde x este termenul general al sumei, iar m și n sunt limitele inferioară și superioară, care pot fi numere întregi sau ∞ , simbolizat prin `inf`. În prima formă variabila după care se face însumarea este v , în a doua formă se utilizează variabila simbolică implicită, determinată de funcția `findsym`.

Exemplul 1.34. Să se calculeze în Matlab suma $\sum_{k=1}^n k^2$.

Soluție:

```
>> syms k n
>> s = symsum(k^2, 1, n)
s =
(n*(2*n + 1)*(n + 1))/6

>> pretty(s)

n (2 n + 1) (n + 1)
-----
6
```

Exemplul 1.35. Să se calculeze în Matlab sumele $\sum_{k=1}^n \frac{1}{k}$, $\sum_{k=1}^{\infty} \frac{1}{k}$, $\sum_{k=1}^{\infty} \frac{1}{k^2}$.

Soluție:

```
>> syms k n
>> r = symsum(1 / k, 1, n)

r =
eulergamma + psi(n + 1)

>> r = symsum(1 / k, 1, Inf)

r =
Inf

>> r = symsum(1/k^2, 1, Inf)

r =
pi^2/6
```

1.10 Dezvoltare în serie Taylor

Fie o funcție f care are derivate până la ordinul n în raport cu x . Funcția Matlab

```
taylor(f, n, x, a)
```

conduce la obținerea dezvoltării funcției f în serie Taylor până la ordinul $n - 1$ în jurul punctului a .

$$\sum_{k=0}^{n-1} (x-a)^k \frac{f^{(k)}(a)}{k!}$$

Dacă argumentul a lipsește, funcția

```
taylor(f, n, x)
```

conduce la dezvoltarea în serie MacLaurin a funcției f până la ordinul $n - 1$ (dezvoltarea în serie în jurul punctului $a = 0$). Dacă argumentul x lipsește, variabila independentă este determinată de funcția `fyndsym`. Dacă argumentul n lipsește, $n = 6$.

Exemplul 1.36. Să se calculeze în Matlab dezvoltarea în serie Taylor a funcției sinus în jurul originii.

Soluție:

```
>> syms x
>> f = sin(x);
>> taylor(f, 5)

ans =
x-x^3/6
```

Exemplul 1.37. Să se calculeze dezvoltarea în serie Taylor a funcției e^x în jurul originii.

Soluție:

```
>> syms x
>> f = exp(x);
>> T = taylor(f, 5, 0)

T =
x^4/24 + x^3/6 + x^2/2 + x + 1

>> pretty(T)

      4      3      2
     x      x      x
     -- + -- + -- + x + 1
     24     6     2
```

Exemplul 1.38. Să se calculeze dezvoltarea în serie Taylor a funcției e^x în jurul punctului $x = 1$.

Soluție:

```
>> syms x
>> f = exp(x);
>> T = taylor(f, 3, 1)
```

```
T =
exp(1) + exp(1)*(x - 1) + (exp(1)*(x - 1)^2)/2

>> pretty(T)

exp(1) + exp(1) (x - 1) +  $\frac{\exp(1) (x - 1)^2}{2}$ 
```

1.11 Rezolvarea simbolică a ecuațiilor algebrice

Fie e o variabilă simbolică ce are ca valoare o expresie algebrică simbolică. Rezolvarea ecuației algebrice

$$e = 0$$

se face cu funcția `solve` ce poate lua următoarele forme

```
solve (e, x)
```

când se rezolvă ecuația după variabila simbolică x și

```
solve(e)
```

când expresia simbolică e are o singură variabilă, sau ecuația se rezolvă după variabila simbolică implicită aleasă după regula enunțată în paragraful relativ la substituții. Funcția `solve` poate avea ca argument și o expresie simbolică între paranteze:

```
solve ('expresie imbolică', x)
```

sau

```
solve('expresie simbolică')
```

Rezultatul funcției `solve` este:

- un vector cu soluții, dacă rezultatul funcției `solve` este un vector cu un număr de componente egal cu numărul de variabile;
- o structură în care numele câmpurilor sunt numele variabilelor ecuațiilor, dacă rezultatul funcției `solve` este o variabilă.

Exemplul 1.39. Să se determine soluțiile ecuației de gradul doi $ax^2+bx+c = 0$.

Soluție:

```
>> syms a b c x
>> f = a*x^2+b*x+c;
>> solve(f)

ans =
-(b + (b^2 - 4*a*c)^(1/2))/(2*a)
-(b - (b^2 - 4*a*c)^(1/2))/(2*a)
```


Exemplul 1.40. Să rezolvăm ecuația de gradul doi de mai sus în raport cu variabila b .

Soluție:

```
>> syms a b c x
>> f = a*x^2 + b*x + c;
>> solve(f, b)

ans =
-(a*x^2 + c)/x
```

În cazul în care se dorește rezolvarea unei *ecuații de forma* $f(x) = g(x)$, se poate utiliza funcția Matlab `solve` cu următoarele două forme

```
solve('f(x) = g(x)', x)
```

sau

```
solve('f(x) = g(x)')
```

Exemplul 1.41. Să se rezolve în Matlab ecuația transcendențială $e^x = x+1$, $x \in \mathbf{R}$.

Soluția:

Rezolvarea ecuației se poate face prin următoarele comenzi Matlab:

```
>> sym(x, 'real')
>> solve('exp(x) = x + 1')

ans =
0
```

Funcția `solve` poate rezolva și *sisteme de ecuații algebrice*. Fie sistemul de două ecuații algebrice

$$\begin{aligned} f(x, y) &= 0 \\ g(x, y) &= 0 \end{aligned}$$

Dacă dorim ca rezultatul funcției `solve` să fie un vector cu soluții, funcția `solve` are forma

```
[x, y] = solve(f, g)
```

sau forma

```
[x, y] = solve('f(x, y)', 'g(x, y)')
```

Dacă dorim ca rezultatul să fie o structură, în stânga semnului egal vom scrie o variabilă.

Exemplul 1.42. Să se rezolve sistemul de ecuații:

$$\begin{cases} x^2 + ax + (a+1)y + a = 0 \\ y^2 + ay + (a+1)x + a = 0 \end{cases}, a \in \mathbf{R}.$$

Soluție:

```
>> syms x y a
>> f = x^2 + a*x + (a+1)*y + a
f =
x^2 + a*x + a + y*(a + 1)

>> g = y^2 + a*y + (a+1)*x + a
g =
y^2 + a*y + a + x*(a + 1)

>> s = solve(f,g)
s =
      x: [4x1 sym]
      y: [4x1 sym]
```

Rezultatul este o structură cu două câmpuri, x și y .

```
>> s.x
ans =
      (- 8*a - 3)^(1/2)/2 + 1/2
      1/2 - (- 8*a - 3)^(1/2)/2
      (4*a^2 + 1)^(1/2)/2 - a - 1/2
      - a - (4*a^2 + 1)^(1/2)/2 - 1/2

>> s.y
ans =
      1/2 - (- 8*a - 3)^(1/2)/2
      (- 8*a - 3)^(1/2)/2 + 1/2
      (4*a^2 + 1)^(1/2)/2 - a - 1/2
      - a - (4*a^2 + 1)^(1/2)/2 - 1/2

>> S = [s.x, s.y];
>> pretty(S)
```

$$\begin{array}{cc}
 \frac{(-8a - 3)^{1/2}}{2} + \frac{1}{2}, & \frac{1}{2} - \frac{(-8a - 3)^{1/2}}{2} \\
 \frac{1}{2} - \frac{(-8a - 3)^{1/2}}{2}, & \frac{(-8a - 3)^{1/2}}{2} + \frac{1}{2} \\
 \frac{(4a^2 + 1)^{1/2}}{2} - a - \frac{1}{2}, & \frac{(4a^2 + 1)^{1/2}}{2} - a - \frac{1}{2} \\
 -a - \frac{(4a^2 + 1)^{1/2}}{2} - \frac{1}{2}, & -a - \frac{(4a^2 + 1)^{1/2}}{2} - \frac{1}{2}
 \end{array}$$

Exemplul 1.43. Să se rezolve sistemul de trei ecuații:

$$x = \log\left(\frac{y}{z}\right) + 1$$

$$y = 0.4 + z^2 - 2x^2$$

$$z = 2 + \frac{xy}{20}$$

Soluție:

```
>> syms x y z
>> f = x - log(y/z) - 1

f =
x - log(y/z) - 1

>> g = y - 0.4 - z^2 + 2*x^2

g =
2*x^2 - z^2 + y - 2/5

>> h = z - 2 - x*y/20

h =
z - (x*y)/20 - 2

>> s = solve(f,g,h)

s =
    x: [1x1 sym]
    y: [1x1 sym]
    z: [1x1 sym]
```

În acest caz soluția s este o structură cu componentele x , y și z . Rezultatele sunt:

```
>> digits(4)
>> s.x
ans =
3.509

>> s.y
ans =
-21.25

>> s.z
ans =
-1.728
```

1.12 Rezolvarea simbolică a ecuațiilor diferențiale ordinare

Fie ecuația diferențială ordinară de ordinul întâi

$$y' = f(y, t)$$

Soluția generală a acestei ecuații diferențiale depinde de o constantă arbitrară C . Dacă ne interesează o soluție care la momentul inițial t_0 trece prin punctul y_0 , constanta C ia valoarea y_0 . În continuare vom presupune că soluția ecuației diferențiale există. În funcție de proprietățile funcției f , soluția poate să fie unică sau nu.

Rezolvarea ecuațiilor diferențiale ordinare se face cu funcția `dsolve` care poate rezolva ecuații diferențiale de ordin superior și sisteme de ecuații diferențiale ordinare. Funcția `dsolve` are ca parametri:

- expresia simbolică a ecuației diferențiale sau, în cazul unui sistem, expresiile simbolice ale ecuațiilor sistemului; în aceste expresii, derivata de ordin întâi este notată cu D , iar derivatele de ordin superior cu $D2, D3, \dots, Dn$. În notația de mai sus dy/dt se notează cu Dy , d^2y/dt^2 se notează cu $D2y$, etc.
- condițiile inițiale ale ecuației diferențiale, dacă există, sunt argumentele următoare ale funcției `dsolve`; ele au forma $y(t_0) = y_0, Dy(t_0) = y_0'$, etc.,
- variabila independentă implicită este t ; variabila independentă poate fi și altă variabilă simbolică, care va fi ultimul argument al funcției `dsolve`.

Toți parametrii funcției `dsolve` sunt incluși între apostrofuri.

Soluția ecuației diferențiale poate fi:

- un vector cu un număr de componente egal cu cel al variabilelor dependente, dacă rezultatul funcției `dsolve` este un vector,
- o structură în care numele câmpurilor sunt numele variabilelor dependente, dacă rezultatul funcției `dsolve` este o variabilă.

Exemplul 1.44. Fie ecuația diferențială:

$$y'' + y = 0$$

Să se calculeze soluția generală a ecuației și apoi soluția ce trece prin punctele

$$y(0) = 1, y'(0) = -1.$$

Soluție:

Rezolvarea problemei se poate face cu instrucțiunile Matlab următoare:

```
>> syms y
>> y = dsolve('D2y + y = 0')
y =
C5*cos(t) + C6*sin(t)

>> syms y
>> y = dsolve('D2y + y = 0', 'y(0) = 1', 'Dy(0) = -1')
y =
cos(t) - sin(t)
```

Exemplul 1.45. Fie sistemul de ecuații diferențiale:

$$\begin{aligned}x' &= x - y \\y' &= x + y\end{aligned}$$

Să se calculeze soluția într-un vector, respectiv într-o structură.

Soluție:

```
>> syms x y
>> [x, y] = dsolve('Dx = x - y', 'Dy = x + y')
x =
C8*i*exp(t*(i + 1)) - (C9*i)/exp(t*(i - 1))
y =
C8*exp(t*(i + 1)) + C9/exp(t*(i - 1))

>> s = dsolve('Dx = x - y', 'Dy = x + y')
s =
    x: [1x1 sym]
    y: [1x1 sym]

>> s.x
ans =
C8*i*exp(t*(i + 1)) - (C9*i)/exp(t*(i - 1))

>> s.y
ans =
C8*exp(t*(i + 1)) + C9/exp(t*(i - 1))

>> [x,y]=dsolve('Dx=x-y','Dy=x+y','x(0)=1','y(0)=2')
x =
exp(t*(i + 1))*(i + 1/2) - (i - 1/2)/exp(t*(i - 1))
y =
(i/2 + 1)/exp(t*(i - 1)) - exp(t*(i + 1))*(i/2 - 1)
```

Exemplul 1.46. Să se rezolve ecuația diferențială:

$$y'' + 3y' + 2y = e^{3t} + 2t + 1$$

Soluție:

Problema se poate rezolva folosind următoarele instrucțiuni Matlab:

```
>> y = dsolve('D2y + 3* Dy + 2*y = exp(3*t) +2*t + 1')
y =
(exp(4*t)/4 - exp(t) + 2*t*exp(t))/exp(t) - (exp(5*t)/5 +
t*exp(2*t))/exp(2*t) + C2/exp(t) + C3/exp(2*t)

>> y = dsolve('D2y + 3* Dy + 2*y = exp(3*t) +2*t + 1', 'y(0) =
1', 'Dy(0) = 0')

y =
11/(4*exp(t)) - 4/(5*exp(2*t)) + (exp(4*t)/4 - exp(t) +
2*t*exp(t))/exp(t) - (exp(5*t)/5 + t*exp(2*t))/exp(2*t)
```

1.13 Exerciții propuse spre rezolvare

Exercițiul 1.1: Să se substituie variabila x cu $x+1$ în expresia

$$\frac{x^3}{3} - \frac{x^2}{2} + x.$$

Exercițiul 1.2: Să se simplifice în Matlab expresiile:

$$\text{a) } f(x) = \frac{x^3 - 1}{x^2 - 3x + 2}$$

$$\text{b) } f(x) = (3a-b)^2 - 3(a-3b) + 2(a+2b)^2$$

Exercițiul 1.3: Să se determine în Matlab inversele, transpusele și determinanții următoarelor matrice:

$$\text{a) } A = \begin{pmatrix} 1 & x \\ x & 0 \end{pmatrix}$$

$$\text{b) } B = \begin{pmatrix} x & 1/x \\ -1/x & x \end{pmatrix}$$

Exercițiul 1.4: Să se reprezinte grafic

$$\text{a) } f(x) = \operatorname{tg} x.$$

$$\text{b) } f(x) = \sin x - \frac{1}{2}, \text{ pentru } x \in [-10, 10]$$

$$\text{c) } \begin{cases} x = \cos^3 t \\ y = \sin^3 t \end{cases}, t \in [0, 2\pi]$$

$$\text{d) } \begin{cases} x = \cos t \\ y = \sin t \\ z = t \end{cases}, t \in [0, 10\pi]$$

Exercițiul 1.5: Să se calculeze în Matlab următoarele derivate

$$\text{a) } \frac{\partial}{\partial x}(x^{n+1}), n \in \mathbf{N}.$$

$$\text{b) } \frac{\partial}{\partial q}(e^{iq})$$

$$\text{c) } \frac{\partial}{\partial t} \sin(at + b), a, b \in \mathbf{R}.$$

$$\text{d) } \frac{\partial^2}{\partial q^2}(e^{iq})$$

$$\text{e) } \frac{\partial}{\partial x}(2x + 3 + \sin^2 x)$$

$$\text{f) } \frac{\partial}{\partial x} \left(\frac{\ln x}{\sqrt{x+1}} \right)$$

Exercițiul 1.6: Să se determine în Matlab iacobianul următoarelor transformări

$$\begin{aligned} \text{a)} & \begin{cases} u = 2x^4 - \ln(x^2 + 4) \\ v = e^{x^2 - 8} + 4 \ln(3x^3 - 2) \end{cases} \\ \text{b)} & \begin{cases} u = \cos(ax + by) \\ v = \sin(cx - dy) \end{cases}, a, b, c, d \in \mathbf{R}. \end{aligned}$$

Exercițiul 1.7: Să se calculeze în Matlab limitele

$$\begin{aligned} \text{a)} & \lim_{n \rightarrow \infty} \left(1 + \frac{x}{n} \right)^n \\ \text{b)} & \lim_{n \rightarrow \infty} \left(1 - \frac{x}{n} \right)^n \\ \text{c)} & \lim_{x \rightarrow a} \frac{x^x - a^x}{a^x - a^a}, a \in \mathbf{R}. \\ \text{d)} & \lim_{\substack{x \rightarrow a \\ x < a}} \exp \left\{ \frac{1}{x - a} \right\} \\ \text{e)} & \lim_{\substack{x \rightarrow a \\ x > a}} \exp \left\{ \frac{1}{x - a} \right\}, a \in \mathbf{R}. \end{aligned}$$

Exercițiul 1.8: Să se calculeze în Matlab următoarele primitive

$$\begin{aligned} \text{a)} & \int e^x \sin x dx \\ \text{b)} & \int \left(x + \frac{1}{x} \right) dx \\ \text{c)} & \int (3x^2 - 4xy) dy. \\ \text{d)} & \int (3x^2 - 4xy) dx \\ \text{e)} & \int \frac{x}{z^2 + 1} dz \\ \text{f)} & \int \frac{1}{\sin^2 x \cdot \cos^2 x} dx \end{aligned}$$

Exercițiul 1.9: Să se calculeze în Matlab următoarele integrale definite

$$\begin{aligned} \text{a)} & \int_0^1 x \ln(x + 1) dx \\ \text{b)} & \int_0^{2p} \sqrt{\sin x} dx \end{aligned}$$

$$\text{c) } \int_{\sin(t)}^1 2x dx$$

$$\text{d) } \int_1^2 (3x^2 - 4x^3 y) dx$$

Exercițiul 1.10: Să se calculeze în Matlab următoarele sume

$$\text{a) } \sum_{k=1}^n \frac{1}{k^2 + k}$$

$$\text{b) } \sum_{k=0}^{\infty} (k^2 + 1)x^k, |x| < 1.$$

$$\text{c) } \sum_{k=1}^{\infty} \frac{1}{k^2}$$

$$\text{d) } \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+2)!}, x \in \mathbf{R}.$$

Exercițiul 1.11: Să se dezvolte în Matlab în serii de puteri următoarele funcții

$$\text{a) } e^x$$

$$\text{b) } \ln(x+1)$$

$$\text{c) } \sin x$$

Exercițiul 1.12: Să se determine în Matlab primii 4 termeni ai dezvoltării în serie Taylor în jurul punctului $x=3$ a funcției

$$f(x) = e^{x^2}.$$

Exercițiul 1.13: Să se rezolve în Matlab ecuațiile

$$\text{a) } \sin(x) + \cos(x) = 1$$

$$\text{b) } \sin x + \sin 2x + \sin 3x = 0$$

$$\text{c) } \sin x = \frac{\operatorname{tg} x}{3}$$

$$\text{d) } e^x = -1$$

$$\text{e) } ax + b = 0, a, b \in \mathbf{R}.$$

$$\text{f) } \frac{ax+1}{x-b} = e^{-a}, a, b \in \mathbf{R}.$$

$$\text{g) } x^2 - 2ax + 7 = 0, a \in \mathbf{R}.$$

$$\text{h) } x^3 + a = 0, a \in \mathbf{R}.$$

Exercițiul 1.14: Să se rezolve în Matlab următoarele sisteme

$$\text{a) } \begin{cases} x = 2^{y-2} - 2 \\ 2 \ln(x+7) = y \ln 3 \end{cases}$$

$$b) \begin{cases} (x+y)^2 = 3z-1 \\ (y+z)^2 = 3u-1 \\ (z+u)^2 = 3x-1 \\ (u+x)^2 = 3y-1 \end{cases}$$

Exercițiul 1.15: Să se rezolve în Matlab următoarea ecuație diferențială

a) $x' = x^2 + 1$

b) $y' = e^{-y}$

c) $x' = kx, x(0) = 3$

d) $y' = \frac{-0.9}{1+2t} y, y(0)=1.$

e) $x''+2x'+x = 0, x(0)=0, x'(0) = 1$

f) $x''+3x'+2x = \frac{1}{e^t + 1}$

g) $\begin{cases} x' = -x + 8y \\ y' = x + y \end{cases}$

Capitolul 2

Reprezentări grafice în MATLAB

MATLAB oferă numeroase facilități pentru prezentarea vizuală a datelor atât în mod interactiv, apelând la instrumente de editare dedicate, cu interfață grafică pentru utilizator, cât și cu ajutorul funcțiilor specializate, apelate direct în fereastra de comenzi sau introduse în fișiere script (*M-file*).

Acest capitol prezintă principalele facilități puse la dispoziție de mediul MATLAB pentru reprezentarea grafică bidimensională și tridimensională a curbelor și suprafețelor, în diverse formate: explicit, implicit, parametric, polar, cilindric, sferic.

Pe lângă problemele legate de reprezentarea grafică propriu-zisă, se prezintă și o serie de moduri în care se pot edita stilurile de desenare a liniilor, culorile, marcăjele, modalitățile prin care se pot adauga mai multe reprezentări pe același grafic sau se pot salva imaginile rezultate în diverse formate.

2.1 Reprezentarea grafică 2D

Pentru cei care au utilizat deja sistemul MATLAB este bine-cunoscut faptul că acesta este *array smart*. Adică, întregul sistem MATLAB se bazează pe operarea cu ușurință a datelor în format vectorial, la fel de simplu ca și în cazul scalarilor.

Exemplul 2.1 Să presupunem că se dorește calcularea valorilor funcției radical pentru mai multe valori de intrare și apoi afișarea grafică a rezultatelor, intrare vs. ieșire.

Soluție: Funcția radical în MATLAB este **sqrt** și aceasta are următoarea sintaxă, iar rezultatul returnat reprezintă rădăcina pătrată a elementelor vectorului *x*:

```
>> x = [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
0     1     4     9    16    25    36    49    64    81
>> y = sqrt(x)
0     1     2     3     4     5     6     7     8     9
```

Pentru reprezentarea grafică a acestor valori discrete se poate face apel la facilitățile puse la dispoziție de interfața grafică: se selectează cele două variabile din spațiul de lucru (*workspace*), mai întâi *x* și apoi *y*, după care se dă click pe butonul marcat în figura următoare.

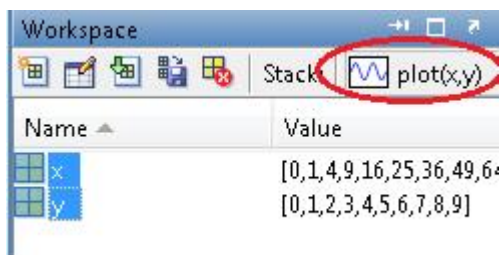


Figura 2.1. Reprezentarea grafică a variabilelor direct din spațiul de lucru

Rezultatul reprezentării grafice poate fi văzut în următoarea figură:

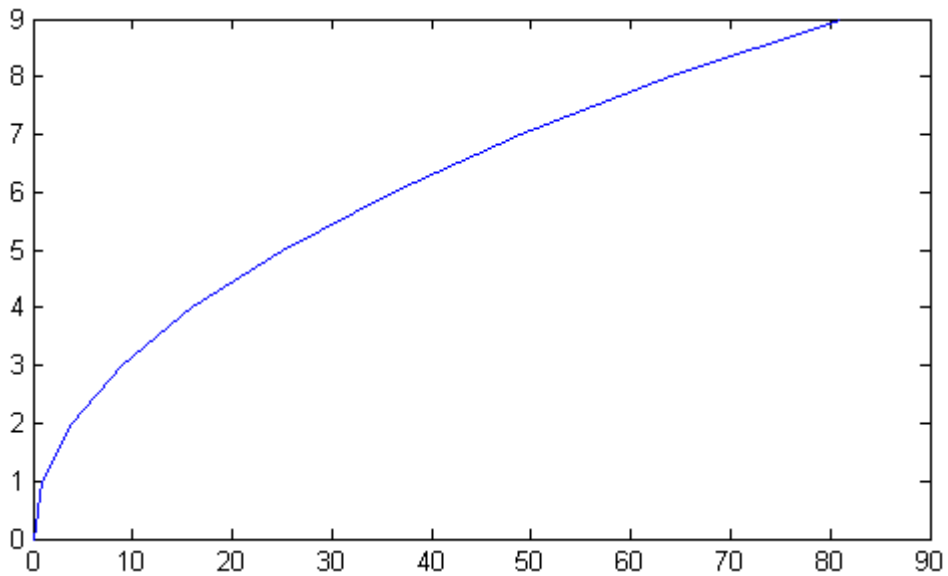


Figura 2.2. Reprezentarea grafică a funcției radical

Dacă se dorește un alt tip de reprezentare grafică, acesta se poate selecta din lista ce apare apăsând cu *mouse*-ul butonul marcat în figura următoare.

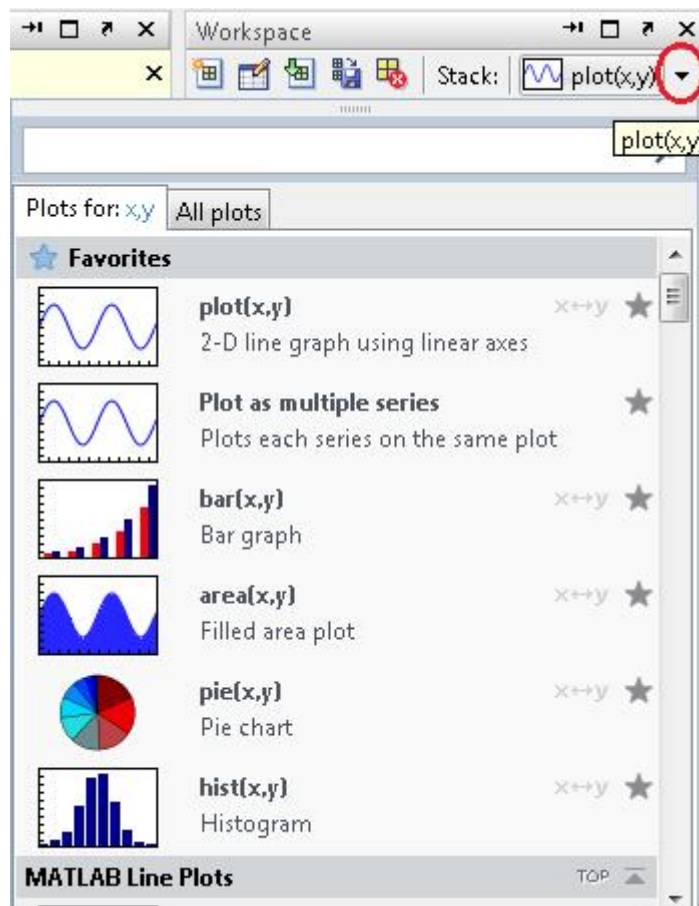


Figura 2.3. Selectarea tipului de reprezentare grafică

Aceleași facilități sunt accesibile și dacă, după selectarea celor două variabile, se apasă pe butonul drept al *mouse*-ului:

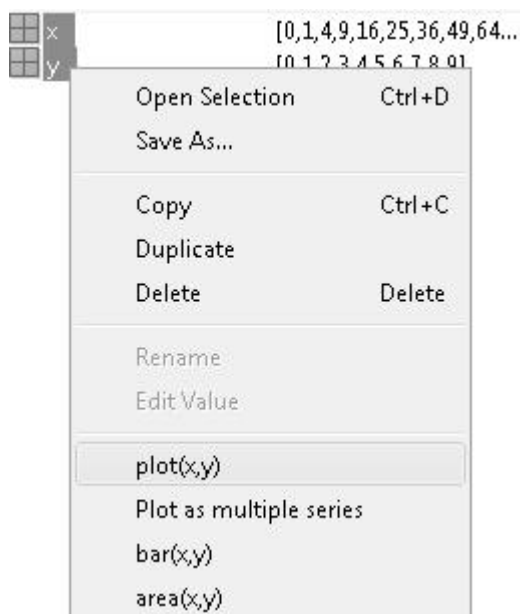


Figura 2.4. Altă modalitate pentru reprezentare grafică a variabilelor din spațiul de lucru

O reprezentare grafică asemănătoare celei din fig. 2.2 se poate obține direct în linie de comandă, apelând la funcția **plot**:

```
>> plot(x,y)
```

Funcția `plot` are diverse forme, în funcție de argumentele care se furnizează. Sintaxa este următoarea:

```
plot(x,y,s)
```

unde x și y sunt tablouri de numere, iar s este un șir de caractere.

Dacă se utilizează doar un singur argument,

```
plot(y)
```

se vor reprezenta grafic elementele vectorului y în raport cu indicii acestora în vector.

Dacă se utilizează două argumente,

```
plot(x,y)
```

se reprezintă grafic elementele vectorului y în raport cu cele ale vectorului x .

În cazul în care se folosesc toate cele trei argumente, ultimul dintre acestea este un șir de caractere de lungime 1, 2 sau 3 folosit special pentru facilitățile de formatare a graficului: selectarea culorii (roșu, galben, albastru, negru etc.), selectarea tipului marcajului (punct, cerc, stea, pătrat etc.), selectarea tipului de linie (solidă, punctată etc.). Tabelul următor listează valorile posibile pentru caracterele utilizate în cadrul șirului s din lista parametrilor funcției **plot**.

caracter	culoare	caracter	marcaj	caracter	tip linie
b	albastru (<u>b</u> lue)	.	punct	-	continuă
g	verde (<u>g</u> reen)	o	cerc	:	punctată
r	roșu (<u>r</u> ed)	x	semnul x	-.	întreruptă și punctată
c	turcoaz (<u>c</u> yan)	+	plus	--	întreruptă
m	purpuriu (<u>m</u> agenta)	*	stea		
y	galben (<u>y</u> ellow)	s	pătrat (<u>s</u> quare)		
k	negru (<u>k</u> black)	d	romb (<u>d</u> iamond)		
w	alb (<u>w</u> hite)	v	triunghi (jos)		
		^	triunghi (sus)		
		<	triunghi (stânga)		
		>	triunghi (dreapta)		
		p	pentagon		
		h	hexagon		

Tabelul 2.1 Opțiunile de formatare ale funcției **plot**

Pentru mai multe detalii pot fi studiate paginile de manual ale funcției **plot**:

```
>> help plot
```

2.1.1 Reprezentarea grafică a funcțiilor reale de o variabilă reală

Această secțiune discută despre reprezentarea grafică a funcțiilor reale de o variabilă reală, punând accent pe diferitele modalități de generare și de configurare a graficelor rezultate (axe, legendă, etichete, adnotări etc.).

Exemplul 2.2. Să se reprezinte grafic funcția pătratică $y = x^2 + 3x + 2$, pe intervalul de valori $[-5, 5]$.

Soluție: La fel ca și în abordarea clasică, utilizată de elevi și studenți pentru a reprezenta grafic o funcție, MATLAB oferă următoarea soluție bazată pe tabelarea unor valori din intervalul de definiție, urmată de apelarea funcției **plot**:

```
>> x = -5:5;
>> y = x.^2+3*x+2;
>> plot (x,y, '*');
```

Cu cât numărul de valori în care se calculează valoarea funcției este mai mare, cu atât graficul rezultat este mai „neted”. Pentru a crește numărul de puncte în care se calculează valorile funcției, se stabilește un alt pas de divizare:

```
>> x = -5:0.5:5;
```

O altă modalitate prin care se pot defini valorile de intrare, cele stocate în vectorul x , se bazează pe utilizarea funcției **linspace**. Spre deosebire de metoda bine-cunoscută, utilizată și în exemplele anterioare, în care se specifică pasul de divizare,

de data aceasta se specifică numărul de puncte intermediare între cele două capete ale intervalului de definiție:

```
>> x = linspace(-5, 5, 21);
```

Modificarea intrării presupune recalcularea valorilor de ieșire, înainte de o nouă reprezentare grafică:

```
>> y = x.^2+3*x+2;  
>> plot(x,y,'*');
```

Cele două reprezentări grafice pot fi analizate comparativ în următoarea figură:

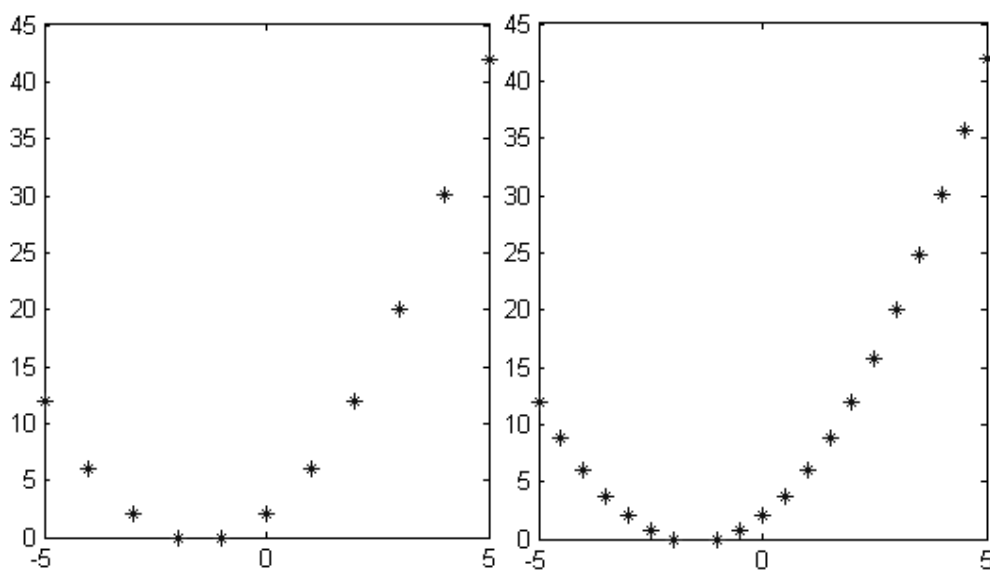


Figura 2.5. Reprezentarea grafică a funcției $y = x^2 + 3x + 2$

În multe situații poate apărea necesitatea reprezentării, în aceeași fereastră, a mai multor curbe, pentru o analiză comparativă. În MATLAB există mai multe modalități prin care se poate realiza acest lucru, fie suprapus, în același sistem de axe, fie alăturat.

O *primă variantă* presupune utilizarea comenzii **hold**. Aceasta are ca efect păstrarea proprietăților ferestrei grafice și a tuturor axelor graficului curent pentru următoarea reprezentare grafică (**hold on**) sau revenirea la modul implicit în care are loc ștergerea graficului curent înainte de a face o nouă reprezentare grafică (**hold off**).

Cea de-a *doua variantă* presupune utilizarea funcției **plot** cu mai multe argumente. Lista celor trei argumente prezentată anterior poate fi multiplicată pentru a include mai multe reprezentări grafice simultan:

```
>> plot(x, y, 'b-', x, z, 'r-')
```

A *treia variantă* presupune utilizarea funcției **subplot**. Aceasta împarte fereastra de reprezentare grafică într-o matrice de axe în care se pot face mai multe reprezentări simultan.

Exemplul 2.3. Să se reprezinte, în același grafic, funcțiile *sin* și *cos* pe intervalul $[0, 4\pi]$.

Soluție:

Varianta I.

```
>> x = linspace(0, 4*pi, 200);  
>> y = sin(x);  
>> plot(x,y,'b-');  
>> hold on;  
>> y = cos(x);  
>> plot(x,y,'r-');
```

Extrem de utilă, mai ales atunci când se reprezintă grafic mai multe funcții, se poate dovedi funcția **legend**. Aceasta permite introducerea unei descrieri a reprezentărilor grafice din fereastra respectivă:

```
>> legend('y = sin(x)', 'y = cos(x)');
```

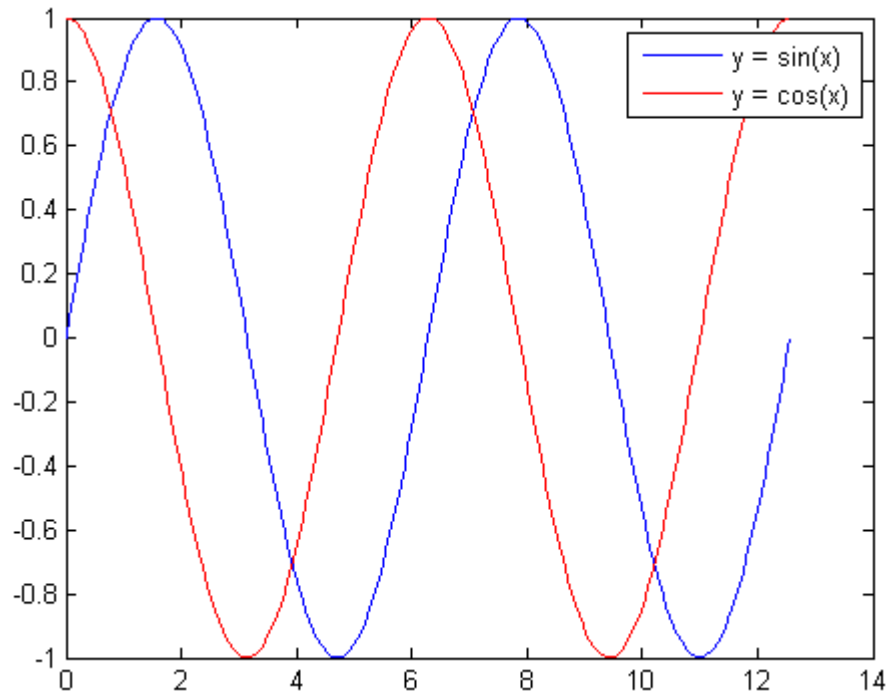


Figura 2.6. Reprezentarea grafică simultană a funcțiilor *sin* și *cos*

Varianta II.

```
>> x = linspace(0, 4*pi, 200)  
>> y = sin(x);  
>> z = cos(x);  
>> plot(x,y,'b-',x,z,'r-');
```

Rezultatul acestor comenzi este identic celui prezentat în fig. 2.6.

Varianta III.

```
>> x = linspace(0, 4*pi, 200);  
>> y = sin(x);  
>> subplot(2,1,1);  
>> plot(x,y,'b-');  
>> legend('y = sin(x)');  
>> subplot(2,1,2);  
>> y = cos(x);  
>> plot(x,y,'r-');  
>> legend('y = cos(x)');
```

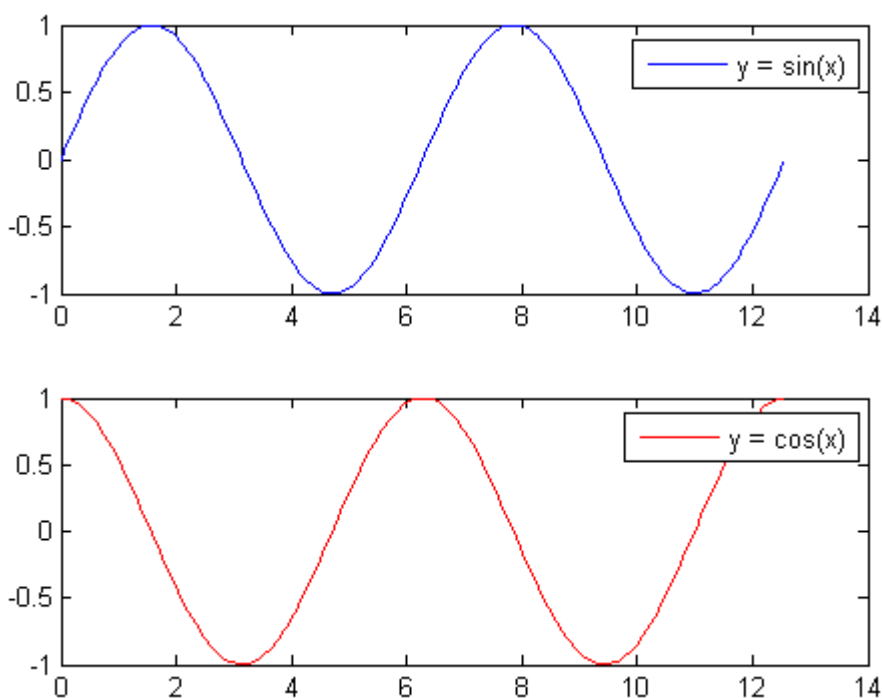


Figura 2.7. Reprezentarea alăturată a graficelor funcțiilor \sin și \cos

Pentru a adăuga mai multe explicații sau detalii unei reprezentări grafice pot fi folosite, pe lângă funcția `legend`, următoarele funcții:

Funcții	Utilizare
xlabel, ylabel	permit adăugarea etichetelor text de-a lungul axelor de coordonate
title	permite adăugarea unui titlu (text) deasupra graficului
text	permite adăugarea unui text oriunde în grafic (trebuie specificate coordonatele)
line	primitivă care permite trasarea unei linii (utilă pentru a desena axele sistemului de coordonate)
grid	permite adăugarea unei grile pe grafic

Tabelul 2.2 Funcții suplimentare pentru formatarea reprezentărilor grafice

Efectele utilizării acestor funcții asupra unei reprezentări grafice pot fi văzute în fig. 2.8. Această reprezentare este rezultatul următoarei secvențe de comenzi.

```
>> title('Graficul functiilor sin si cos');
>> xlabel('Axa Ox');
>> ylabel('Axa Oy');
>> x = [0, 4*pi]; y = [0, 0]; line(x,y);
>> x = [-1, 1]; y = [0, 0]; line(x,y);
>> grid on
```

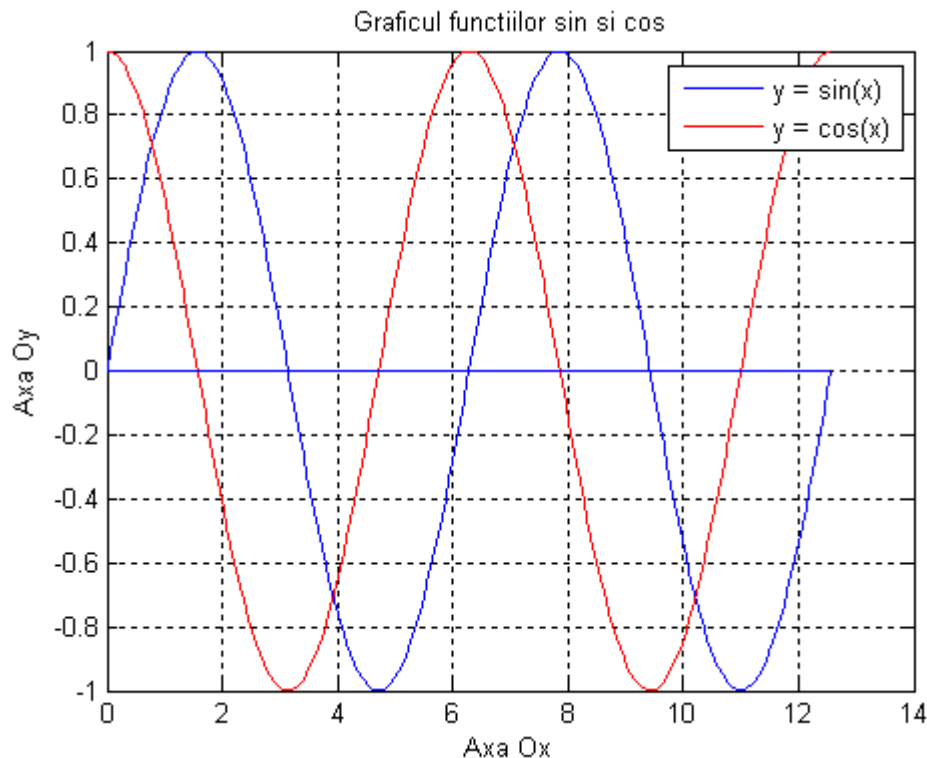


Figura 2.8. *Reprezentare grafică însoțită de explicații*

2.1.2 Reprezentarea în plan a curbelor date prin ecuații parametrice

Această secțiune pune accent pe reprezentarea grafică a curbelor plane date prin ecuații parametrice. Nu există diferențe majore din punctul de vedere al utilizării comenzilor și funcțiilor MATLAB. Abordarea se bazează, ca de obicei, pe identificarea coordonatelor punctelor din plan care satisfac respectivele ecuații parametrice. Așadar, singura problemă care trebuie rezolvată mai întâi este dată de calculul valorilor coordonatelor x și y în funcție de cele ale parametrului t .

Exemplul 2.4. Să se reprezinte grafic curba dată prin ecuația:

$$\begin{aligned} x &= 1 + |t| \\ y &= |1 - t^2|, \text{ pentru } t \in [-1, 1]. \end{aligned}$$

Soluție:

Comenzile MATLAB sunt următoarele:

```
>> t = linspace(-1,1,500);
```

```

>> x = 1+abs(t); y = abs(1-t.^2);
>> subplot(1,3,1); plot(t,x); title('x(t)');
>> subplot(1,3,2); plot(t,y); title('y(t)');
>> subplot(1,3,3); plot(x,y); title('y(x)');

```

Această secvență de comenzi realizează atât reprezentarea grafică a coordonatelor în funcție de parametrul t , $x = x(t)$, $y = y(t)$, cât și a legăturii între coordonate $y = y(x)$.

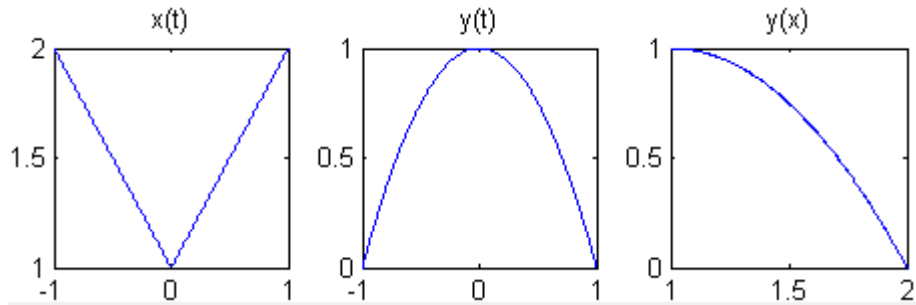


Figura 2.9. *Reprezentare grafică a curbelor date prin ecuații parametrice*

După cum se poate observa în reprezentările grafice din fig. 2.8, valorile coordonatei x , în funcție de valorile parametrului, mai întâi scad și apoi cresc, cele ale coordonatei y mai întâi cresc și apoi scad în timp ce, în aparență, valorile coordonatei y scad odată cu creșterea valorilor coordonatei x . În realitate lucrurile nu stau chiar așa. Pentru a obține o informație corectă, se poate utiliza funcția **comet**. Această funcție realizează o animație în care se poate vedea foarte clar sensul în care se modifică valorile unei coordonate în funcție de cealaltă (fig 2.9.).

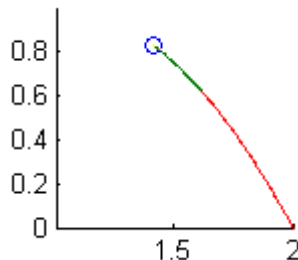


Figura 2.10. *Animație creată cu ajutorul funcției comet*

2.1.3 Reprezentarea în plan a curbelor date prin ecuații polare

Este cunoscut faptul că sistemul de coordonate carteziene nu reprezintă singura modalitate de a descrie poziția punctelor în plan. În loc de a furniza coordonatele pe cele două axe (Ox și Oy), se poate identifica locația punctului în funcție de distanța acestuia față de origine și de unghiul pe care dreapta ce unește punctul și originea îl face cu una din axele de coordonate. Acest sistem de reprezentare poartă numele de sistem de *coordonate polare* (fig. 2.10).

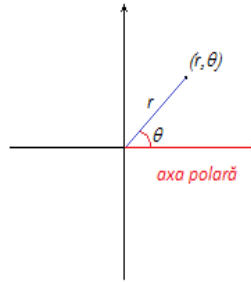


Figura 2.11. Sistemul de coordonate polare

Dificultatea în cazul sistemului de coordonate polare, ținând cont de periodicitatea funcțiilor trigonometrice, constă în faptul că punctele pot avea reprezentări multiple. De aceea trebuie tratate cu atenție aceste tipuri de reprezentări grafice.

În MATLAB, reprezentarea în sistemul de coordonate polar se face cu ajutorul funcției **polar**.

Exemplul 2.5. Să se reprezinte grafic, în coordonate polare, curba dată prin ecuația $r = 2 \sin 2q$, $q \in [0, p]$.

Soluție:

Comenzile MATLAB necesare acestei reprezentări grafice sunt următoarele:

```
>> teta = linspace(0,pi);
>> r = 2 * sin(2*teta);
>> polar(teta,r);
```

Rezultatul poate fi observat în figura următoare.

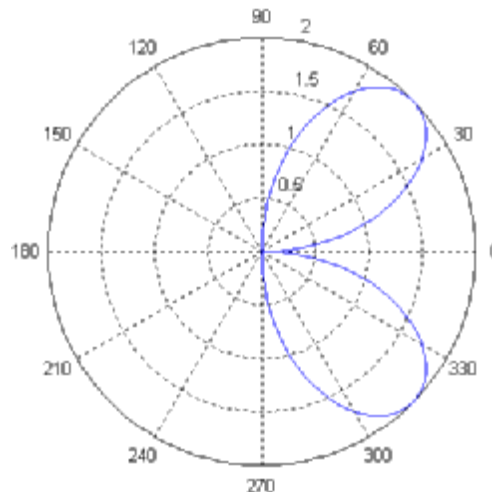


Figura 2.12. Reprezentarea grafică, în coordonate polare, a funcției $r = 2 \sin 2q$

Ținând cont de legăturile existente între sistemele de coordonate cartezian și polar:

$$x = r \cos q$$

$$y = r \sin q$$

MATLAB pune la dispoziția utilizatorilor două funcții de trecere de la un sistem de reprezentare la celălalt: **cart2pol** și **pol2cart**. Ambele primesc drept argumente coordonatele punctelor într-un sistem de coordonate și le returnează pe cele din celălalt sistem. Ordinea în care trebuie furnizate coordonatele este următoarea: pentru sistemul de coordonate cartezian, ordinea naturală, mai întâi x și apoi y , iar pentru cel polar, mai întâi unghiul q și apoi raza r .

Așadar, reprezentarea curbei din figura 2.11 se poate obține utilizând funcțiile **plot** sau **comet**, după ce, în prealabil, coordonatele carteziene sunt calculate în funcțiile de cele polare:

```
>> [x,y] = pol2cart(teta,r);
>> plot(x,y); axis equal;
```

Ultima comandă este utilizată pentru a asigura utilizarea aceleiași scări de reprezentare pe cele două axe de coordonate. În caz contrar, este posibil ca graficul să fie turtit (alungit), în funcție de forma și dimensiunea ferestrei în care se efectuează reprezentarea grafică.

2.1.4 Reprezentarea în plan a curbelor date prin ecuații implicite

În practică există și situații în care nu există o relație explicită sau parametrică între coordonatele punctelor curbei. În această situație, funcțiile *plot*, *comet* sau *polar* nu sunt de niciun ajutor. Sistemul MATLAB pune la dispoziție însă un set de funcții simple (*easy*) care să ajute utilizatorul atunci când legătura între coordonate este implicită. Dintre acestea, în secțiunea curentă se va exemplifica doar utilizarea funcției **ezplot**. În mod similar se poate utiliza și funcția **ezpolar**. Pentru mai multe detalii se pot consulta paginile de manual.

Exemplul 2.6. Să se reprezinte grafic funcția dată prin ecuația:

$$(x^2 + y^2)^2 = x^2 - y^2,$$

în intervalul $[-2, 2]$.

Soluție:

Comanda MATLAB care generează acest grafic este următoarea:

```
>> ezplot('(x.^2+y.^2).^2-(x.^2-y.^2)', [-2,2,-1,1])
```

Rezultatul poate fi observat în figura următoare:

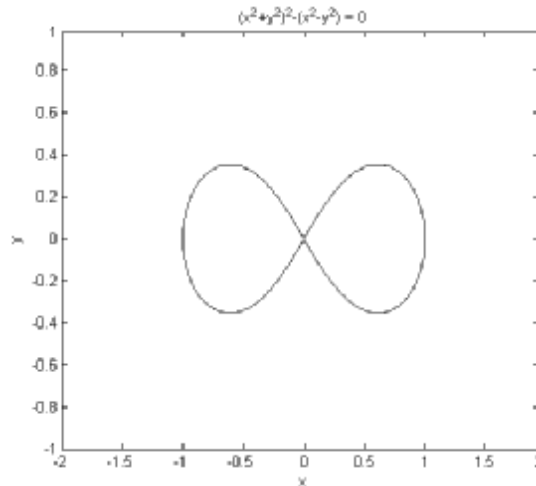


Figura 2.13. Reprezentarea grafică a funcției $(x^2 + y^2)^2 = x^2 - y^2$

2.2 Reprezentarea grafică 3D

Această secțiune prezintă principalele facilități puse la dispoziție de mediul MATLAB pentru reprezentarea grafică tridimensională a curbelor și suprafețelor, în diverse formate (explicit, parametric, polar, etc.). Spre deosebire de spațiul bidimensional, reprezentat în mod natural pe ecran, în cazul reprezentărilor grafice 3D, mai dificil de prezentat pe ecran, trebuie „simulată” cea de-a treia axă pentru a crea o imagine cât mai apropiată de cea reală. Acest lucru este realizat însă de către mediul MATLAB, degrevând utilizatorul de eventualele dificultăți.

2.2.1 Reprezentarea grafică a funcțiilor reale de două variabile reale

La fel cum funcțiile reale de o variabilă reală, $y = f(x)$ conduc la perechi ordonate (x, y) reprezentând coordonatele unor puncte în planul Cartezian, funcțiile reale de două variabile reale, $z = f(x, y)$ conduc tripletul (x, y, z) , triplet ce reprezintă coordonatele unor puncte în sistemul de coordonate Cartezian tridimensional.

Exemplul 2.7. Se consideră funcția $f(x, y) = x + 2y - 4$. Pentru $x = 2$, $y = 2$ se obține $z = f(2, 2) = 2 + 4 - 4 = 2$, ceea ce conduce la punctul M de coordonate $(2, 2, 2)$, punct reprezentat în figura următoare:

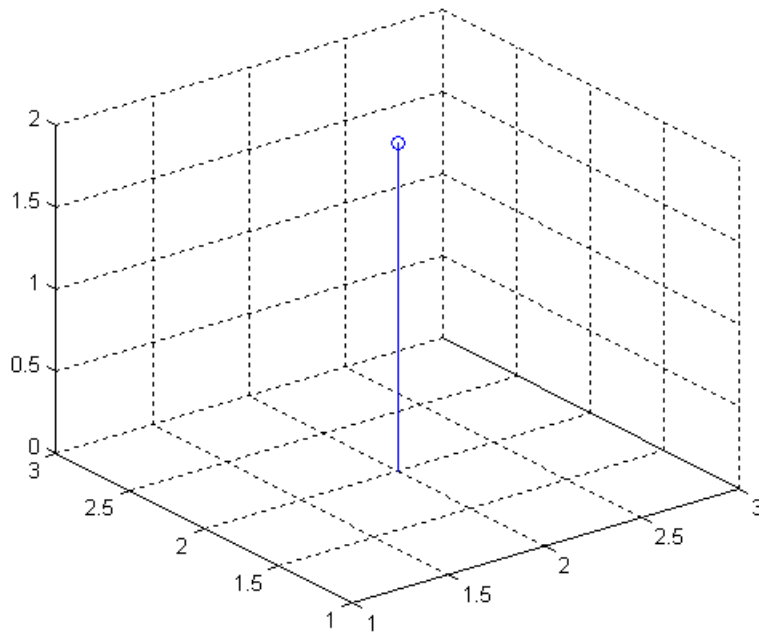


Figura 2.14. Reprezentarea în sistemul de coordonate 3D a punctului $M(2,2,2)$

Imaginea din fig.2.14 se obține foarte ușor apelând la funcția MATLAB **plot3**. Aceasta, la fel ca și echivalenta sa în 2D, *plot*, are sintaxa: `plot3(x, y, z, s)` în care: x , y și z sunt 3 vectori de aceeași lungime reprezentând coordonatele punctelor ce se doresc a fi reprezentate, iar s este un șir de caractere de lungime 1, 2 sau 3 cu aceleași semnificații ca în cazul funcției **plot**. Comenzile MATLAB sunt următoarele:

```
>> plot3(2, 2, 2, 'bo');
>> line([2 2],[2 2],[0 2]);
```

Dacă se dorește reprezentarea mai multor puncte ale căror coordonate satisfac ecuația dată de o funcție de două variabile, atunci trebuie generate mai multe perechi (x, y) care să conducă la o grilă de puncte. Acest lucru este posibil în MATLAB folosind funcția **meshgrid**.

Exemplul 2.8. Se consideră aceeași funcție din exemplul 2.7 în care atât x cât și y iau valori din intervalul $[2, 5]$.

Soluție:

```
>> [x,y] = meshgrid(2:5);
>> z = x+2*y-4;
>> plot3(x, y, z, 'bo');
```

Punctele sunt reprezentate în figura următoare:

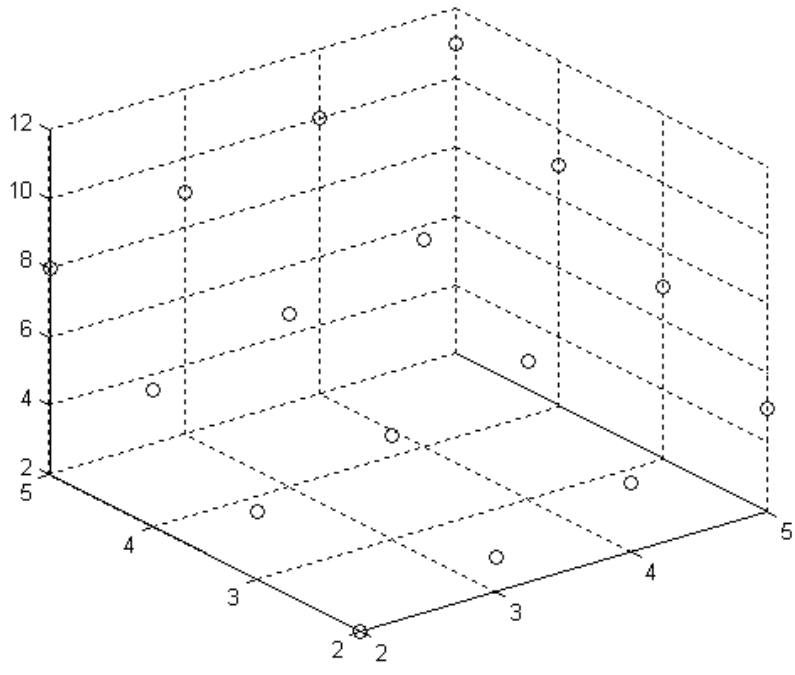


Figura 2.15. Diverse puncte ale căror coordonate satisfac ecuația $z = f(x, y) = x + 2y - 4$

Dacă, mai mult, se dorește schițarea suprafeței a cărei ecuație este dată de funcția de două variabile de mai sus, atunci se poate folosi funcția **mesh**. Această funcție afișează suprafață în spațiul 3D, colorând doar liniile ce unesc punctele de definiție a suprafeței (punctele definite de tripletele (x, y, z) din lista de argumente primită de funcție).

```
>> [x,y] = meshgrid(2:5);
>> z = x+2*y-4;
>> mesh(x,y,z);
```

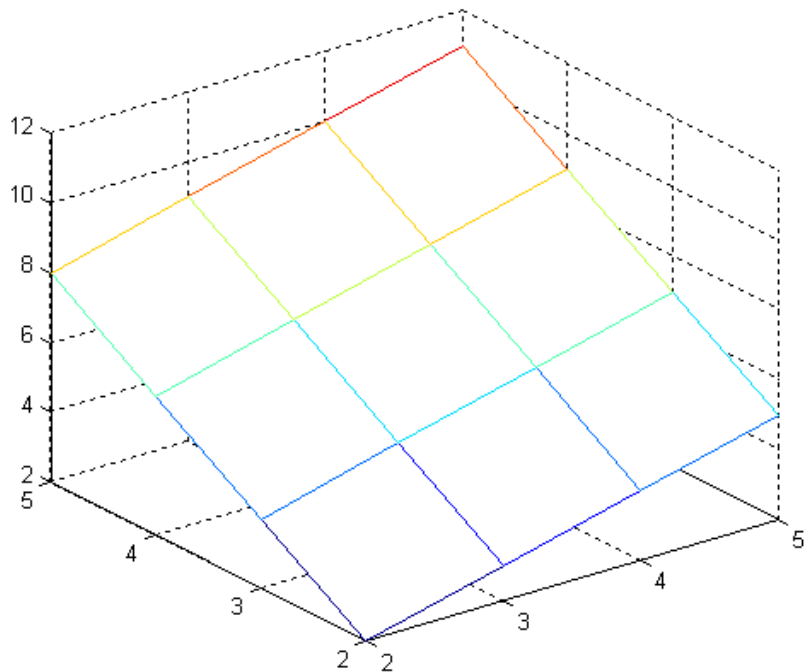


Figura 2.16. Suprafața de ecuație $z = x + 2y - 4$

Dacă se dorește vizualizarea suprafeței din alt unghi, imaginea poate fi rotită folosind mouse-ul în mod interactiv, după apăsarea butonului **Rotate 3D**, marcat cu roșu în figura următoare:



Figura 2.17. Butonul Rotate 3D

2.2.2 Reprezentarea în spațiu a suprafețelor date prin ecuații parametrice

În această secțiune se discută modul de reprezentare grafică a suprafețelor în spațiul tridimensional atunci când acestea sunt date prin ecuații parametrice.

Exemplul 2.9. Să se schițeze suprafața dată prin ecuațiile parametrice:

$$\begin{aligned}x &= r \cos q \\y &= r \sin q \\z &= r\end{aligned}$$

în care $0 \leq r \leq 1, 0 \leq q \leq 2\pi$.

Soluție:

Mai întâi, folosind deja cunoscuta funcție *linspace*, se generează valorile pentru parametrii r și q , în limitele domeniilor acestora.

```
>> r = linspace(0, 1, 100);
>> teta = linspace(0, 2*pi, 100);
```

Apoi, folosind funcția **meshgrid** se generează grila perechilor ordonate (r, θ) :

```
>> [r, teta] = meshgrid(r, teta);
```

și se calculează valorile coordonatelor carteziene ale punctelor de pe suprafață:

```
>> x = r.* cos(teta);
>> y = r.* sin(teta);
>> z = r;
```

după care se reprezintă suprafața folosind funcția **mesh**.

```
>> mesh(x, y, z);
```

O altă funcție oferită de MATLAB pentru reprezentarea suprafețelor este **surf**. Aceasta are același efect cu funcția *mesh* cu deosebirea că pe lângă colorarea liniilor ce unesc punctele de definiție, umple folosind culoare poligoanele care alcătuiesc suprafața. Culoarea poligoanelor sunt determinate în funcție de valoarea coordonatei din vectorul z și de o hartă a culorilor (**colormap** – listă ordonată de culori). Funcția poate primi suplimentar o serie de argumente care să reprezinte proprietăți ale suprafeței rezultate.


```
>> surf(x, y, z, ...
        'FaceColor', 'interp', ...
        'EdgeColor', 'none', ...
        'FaceLighting', 'phong');
```

De asemenea, reprezentarea grafică poate fi îmbunătățită acționând asupra muchiilor, adăugând culoare și lumină, modificând scara pe axele de coordonate sau modificând punctul de vizualizare al obiectului grafic. Comenzile MATLAB care realizează acest lucru fac apel la funcțiile **camlight**, **lighting**, **view** și **axis**.

Illuminarea reprezintă o tehnică prin care un obiect este reprezentat ca fiind expus unei surse de lumină direcțională. În anumite cazuri această tehnică, alăturată eventual cu modificarea punctului de observație, poate conduce la scoaterea în evidență a unor diferențe subtile în forma suprafeței, fiind mai ușor de observat. De asemenea, iluminarea poate adăuga un anumit realism formelor 3D.

Specificarea punctului de vizualizare se face stabilind unghiul din care un observator vede obiectul grafic. Pentru această trebuie furnizate două valori unghiulare (ambele în grade): azimutul sau rotația orizontală și elevația verticală.

```
>> camlight left;
>> view (160,10);
>> axis equal;
>> axis off;
```

Rezultatul poate fi observat în figura următoare, figură în care se poate observa comparativ suprafața schițată cu ajutorul funcției *mesh* (în stânga) și aceeași suprafață, rotită și iluminată, generate cu ajutorul funcției *surf* (în dreapta).

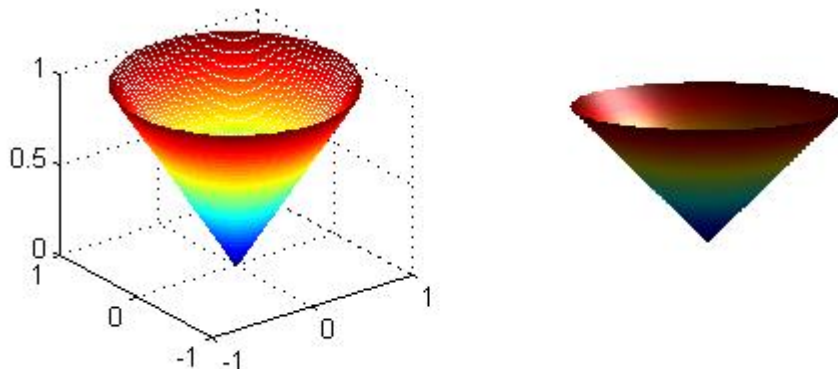


Figura 2.18. Suprafața parametrică din exemplul 2.9

Efecte similare de rotație, translație și mărire/micșorare (*zoom*) se pot obține și utilizând butoanele din bara de instrumente a ferestrei în care este reprezentată imaginea.



Figura 2.19. Suprafața parametrică din exemplul 2.9

2.2.3 Primitive grafice pentru suprafețe

Așa cum pentru dreaptă în plan există primitiva **line** și pentru cele mai cunoscute suprafețe în spațiu (sferă, elipsoid și cilindru) există o serie de primitive.

Sintaxa acestor comenzi este următoarea:

$[X, Y, Z] = \text{sphere}(N)$ – generează trei matrici de dimensiune $(N+1) \times (N+1)$ care reprezintă coordonatele unor puncte situate pe sfera unitate.

$[X, Y, Z] = \text{cylinder}(R, N)$ – generează trei matrici de puncte care reprezintă coordonatele unor puncte aflate pe o suprafață cilindrică bazată pe curba generatoare din vectorul R . Acest vector conține razele suprafeței cilindrice în diverse puncte egal distanțate.

$[X, Y, Z] = \text{ellipsoid}(XC, YC, ZC, XR, YR, ZR, N)$ – generează trei matrici de dimensiune $(N+1) \times (N+1)$ care reprezintă coordonatele unor puncte situate pe elipsoidul cu centrul în punctul de coordonate (XC, YC, ZC) și de raze (XR, YR, ZR) .

După generarea coordonatelor mulțimii de puncte de pe suprafețe, pentru reprezentarea grafică a suprafețelor generate cu ajutorul primitivelor grafice se folosesc funcțiile *surf* sau *mesh*.

Exemplul 2.10. Să se schițeze curba de intersecție unei suprafețe sferice de cu una cilindrică de rază jumătate din raza sferei și care conține centrul sferei (curba lui Viviani).

Soluție:

Comenzile MATLAB care reprezintă grafic această intersecție sunt următoarele:

```
[x1, y1, z1] = sphere(1000);  
mesh(2*x1, 2*y1, 2*z1, 'EdgeColor', [0.3 0.3 0.6]);  
hold on;  
[x2, y2, z2] = cylinder([1 1 1],1000);  
mesh(x2, y2-1, 4*z2-2, 'EdgeColor', [0.3 0.6 0.3]);
```

Deoarece funcția *sphere* generează o sferă unitate (de diametru 1), a fost necesar să se înmulțească cu 2 coordonatele punctelor de pe suprafață pentru a obține o sferă de rază 2, centrată în origine.

De asemenea, coordonatele punctelor de pe suprafața cilindrică au fost deplasate (translatate) pentru a include centrul sferei pe suprafață, iar înălțimea cilindrului a fost mărită pentru a se evidenția intersecția celor două suprafețe.

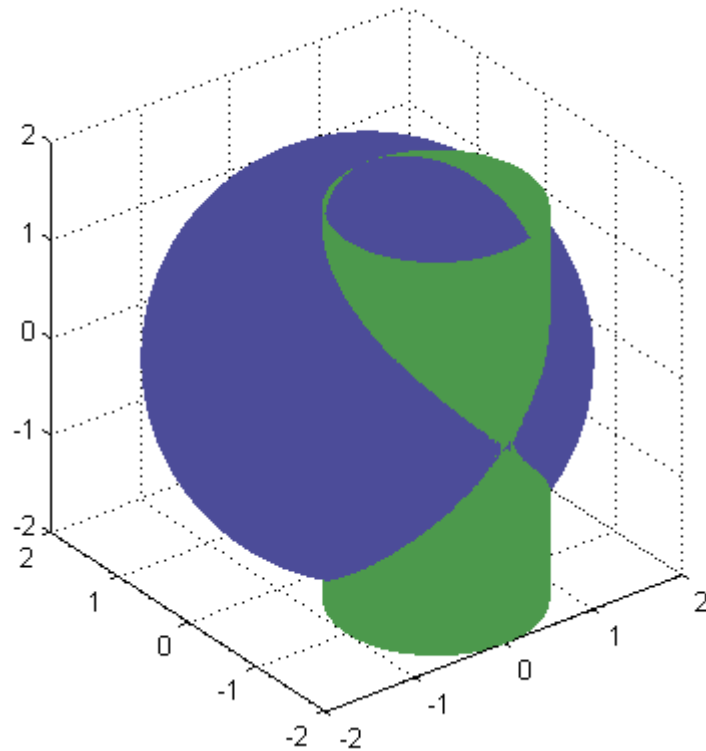


Figura 2.20. Intersecția unei suprafețe sferice cu una cilindrică

2.2.4 Reprezentarea în spațiu a curbelor date prin ecuații parametrice

Comenzile *plot* și *comet* utilizate pentru reprezentarea curbelor în plan au un echivalent atunci când se discută despre reprezentarea curbelor în spațiu: *plot3* și respectiv, *comet3*.

Exemplul 2.11. Să se reprezinte grafic curba dată prin următoarele ecuații parametrice:

$$x = a \cos wt$$

$$y = a \sin wt$$

$$z = bt$$

unde $a = 2$, $b = 0.1$, $w = 2$ și $0 \leq t \leq 12\pi$.

Soluție:

Se inițializează valorile constantelor care dau amplitudinea a și rata de creștere b .

```
>> a = 2;
>> b = 0.1;
>> w = 2;
```

Se încarcă un vector cu valorile parametrului.

```
>> t = linspace(0, 12*pi, 1000);
```

Se calculează valorile tripletelor (x, y, z) care determină coordonatele punctelor curbei, în funcție de cele ale parametrului t .

```
>> x = a * cos(w * t);  
>> y = a * sin(w * t);  
>> z = b * t;
```

Pentru a identifica sensul mișcării se folosește comanda:

```
>> comet3(x, y, z);
```

Pentru a reprezenta punctele curbei se folosește comanda:

```
>> plot3(x, y, z);
```

Eventual, se adaugă informații suplimentare pentru axe și titlu:

```
>> xlabel('axa Ox');  
>> ylabel('axa Oy');  
>> zlabel('axa Oz');  
>> title('x=2 cos(t), y=2 sin(t), z=0.1t');
```

Curba dată parametric, prin ecuațiile parametrice de mai sus, se numește *helix* și poate fi observată în figura următoare:

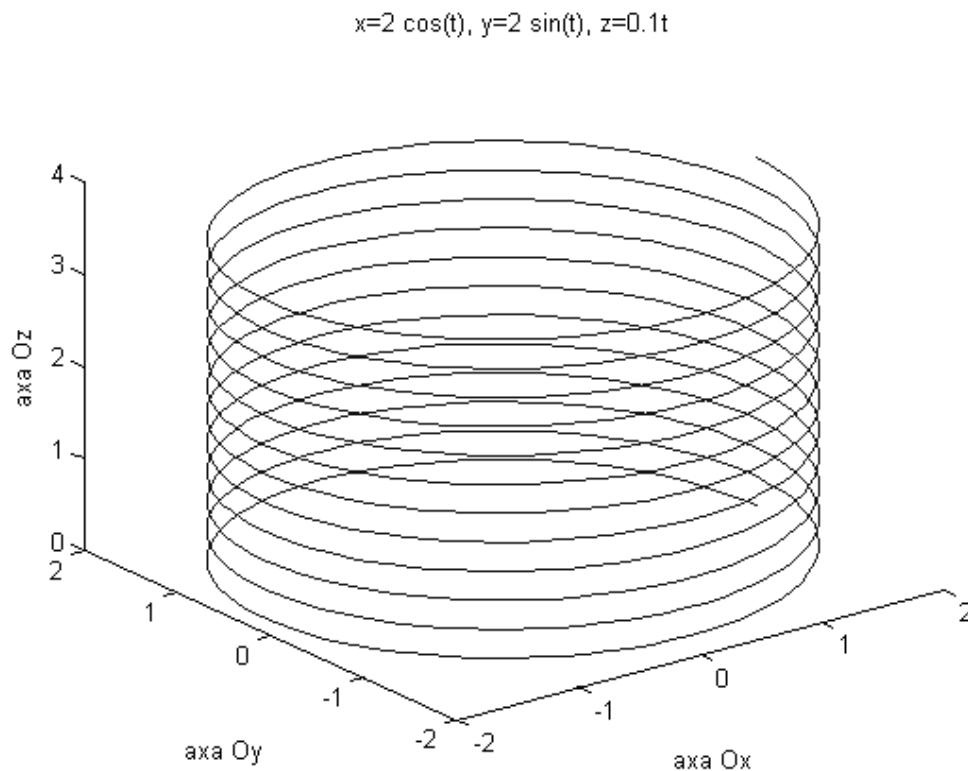


Figura 2.21. *Curba helix*

2.2.5 Reprezentarea în spațiu a curbelor și suprafețelor date implicit

La fel ca și în cazul curbelor plane, în cazul curbelor și suprafețelor în spațiul tridimensional se pot ivi situații în care relațiile între coordonatele punctelor de pe curbă sau suprafață sunt date implicit și nu explicit. Sistemul MATLAB pune la dispoziție o serie de funcții simple numite EZ (*easy*) care permit schițarea unei curbe sau a unei suprafețe cu un minim de cerințe suplimentare. Aceste funcții nu oferă aceleași performanțe ca funcțiile *plot* sau *surf*, însă pot fi extrem de utile atunci când se dorește doar o reprezentare grafică rapidă și fără pretenții.

Funcțiile care pot fi utilizate în cazul reprezentărilor grafice în spațiul 3D sunt prezentate în următorul tabel.

Funcții	Utilizare
ezcontour	desenează curbele de nivel corespunzătoare unei funcții reale de două variabile reale $f(x, y)$ pe un domeniu specificat sau pe domeniul implicit $-2\pi < x, y < 2\pi$
ezcontourf	are același efect ca <i>ezcontour</i> la care se adaugă umplerea cu culoare a spațiilor dintre curbele de nivel
ezmesh	schitează o suprafață tridimensională corespunzătoare unei funcții reale de două variabile reale $f(x, y)$ dată implicit sau parametric pe un domeniu specificat sau pe un domeniu implicit
ezmeshc	schitează suprafața și curbele de nivel corespunzătoare
ezplot3	desenează o curbă reprezentată parametric un domeniu specificat sau pe un domeniu implicit al parametrului
ezsurf	desenează o suprafață tridimensională colorată, corespunzătoare unei funcții reale de două variabile reale $f(x, y)$ dată implicit sau parametric pe un domeniu specificat sau pe un domeniu implicit
ezsurf	desenează suprafața și curbele de nivel corespunzătoare

Tabelul 2.3 Funcțiile din categoria EZ

Funcțiile pot fi definite separat și apoi pot fi apelate cu ajutorul unui *handler* sau pot fi specificate sub forma unui șir de caractere. De exemplu:

```
>> ezplot3(@sin,@cos,'t')
sau
>> ezplot3('cos(t)', 'sin(t)', 't')
```

Exemplul 2.12. Să se schițeze suprafața și curbele de contur în cazul suprafeței dată prin ecuația implicită $x^2 - y^2 = 0$.

Soluție:

```
>> ezsurf('x^2-y^2')
```

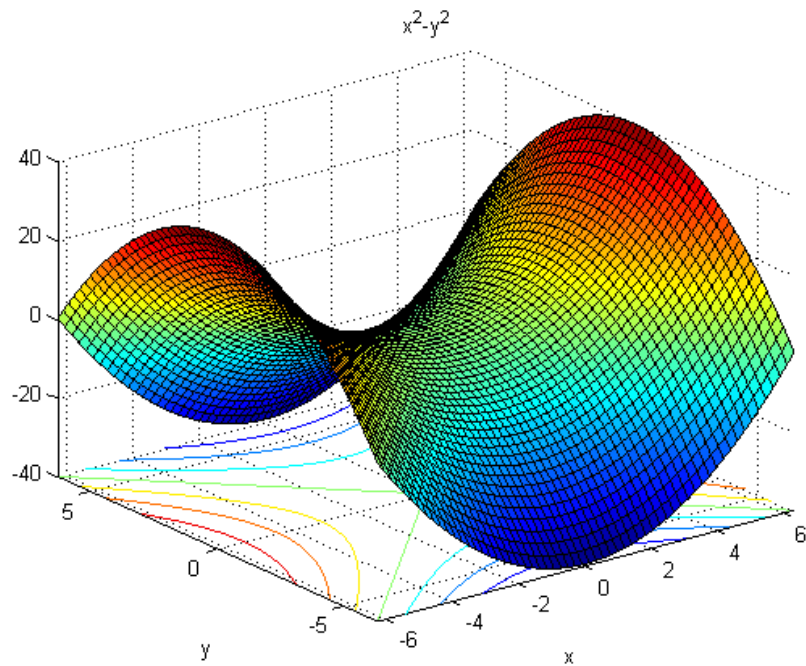


Figura 2.22. Reprezentarea unei suprafețe și curbelor de contur corespunzătoare

2.2.6 Reprezentarea în sistemele de coordonate sferice și cilindrice

În cazul în care suprafețele și curbele din spațiul tridimensional sunt date în formă explicite, dar în coordonate sferice (sau cilindrice), înainte de reprezentarea grafică propriu-zisă se poate apela la ajutorul funcțiilor de conversie de la sistemul de coordonate sferic (sau cilindric) la cel cartezian: **sph2cart** (**pol2cart**).

Exemplul 2.13. Să se reprezinte grafic suprafața de ecuație $r = 2$, în sistemul de coordonate sferic, pentru domeniile de valori $0 \leq \varphi \leq 2\pi$, $-\pi/2 \leq \theta \leq \pi/2$.

Soluție:

Pașii ce trebuie urmați sunt descriși în cele ce urmează:

1. se generează valorile pentru φ și θ .

```
>> teta = linspace(0,2*pi,100);
```

2. se generează grila pentru variabilele de intrare

```
>> fi = linspace(-pi,pi,100);
```

3. se calculează valoarea lui r

```
>> ro = 2;
```

4. se trece la coordonate carteziene

```
>> [x, y, z] = sph2cart(teta,fi,ro);
```

5. se reprezintă suprafața în sistemul de coordonate carteziene

```
>> surf(x, y, z);
```

Bineînțeles, la fel ca și în cazul bidimensional, există și funcțiile de conversie inversă care fac transformarea de la reprezentarea în coordonate carteziene la cele sferice (sau cilindrice): **cart2sph** (**pol2cart**).

2.3 Alte tipuri de reprezentări grafice

În această secțiune vor fi exemplificate câteva funcții MATLAB utilizate pentru diverse reprezentări în plan, funcții care se pot dovedi extrem de utile în aplicațiile ingineresti: Acestea sunt prezentate în tabelul următor.

Funcții	Utilizare
Bar, barh, bar3, bar3h	funcții utilizate pentru a reprezenta 2D sau 3D o valorile numerice sub forma unor coloane verticale respectiv, orizontale
hist	utilizată pentru generarea și reprezentarea sub formă de coloane a histogramei unui tablou de numere, în funcție de distribuția acestora într-un număr intervale implicite sau definite de utilizator
rose	echivalenta funcției <i>hist</i> în sisteme de coordonate polare
area	utilizată pentru a trasa un grafic și a umple suprafață dintre acesta și axa orizontală, Ox
Pie, pie3	utilizată pentru o reprezentare sub formă unui disc (<i>plăcintă</i>) 2D, respectiv 3D în care fiecare valoare este exprimată prin raportarea la suma tuturor valorilor
Stem, stem3	funcții utilizate pentru reprezentarea secvențelor discrete de date
stairs	afișează o secvență discretă de date sub formă de trepte
compass	afișează în vectori plecând din origine
feather	afișează vectorii plecând din puncte de pe axa orizontală
quiver, quiver3	afișează vectori 2D, respectiv 3D
contour, contour3	afișează curbele de contur (izolinii) 2D, respective 3D

Tabelul 2.4 Alte tipuri de reprezentări grafice

Exemplul 2.14. Să se reprezinte grafic notele obținute de o grupă de 20 studenți la examenele susținute în anul II la disciplina Matematică și distribuția acestora. Se consideră următoarele note: 2, 4, 7, 2, 8, 3, 10, 3, 8, 2, 3, 1, 6, 7, 6, 5, 7, 7, 7, 7.

Soluție:

Comenzile MATLAB sunt următoarele:

```
>> a = [2,4,7,2,8,3,10,3,8,2,3,1,6,7,6,5,7,7,7,7];
>> subplot(3,1,1);
>> bar(a);
>> subplot(3,1,2);
>> hist(a);
```

```
>> subplot(3,1,3);
>> area(hist(a));
```

Graficele rezultate sunt reprezentate în figura următoare:

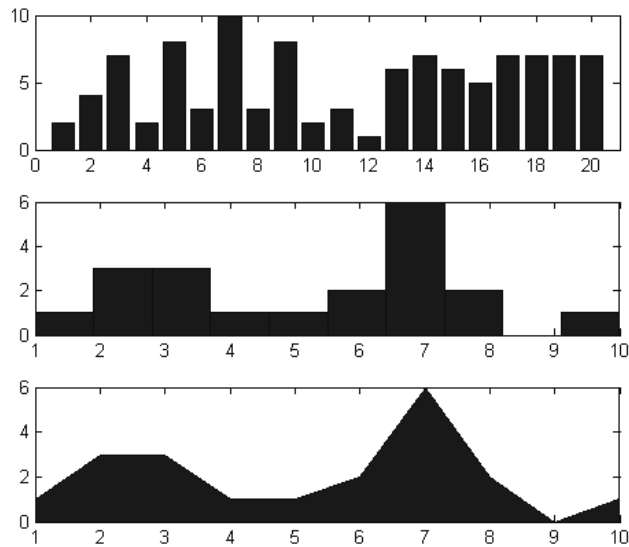


Figura 2.23. Reprezentarea grafică a notelor și a distribuțiilor acestora

Dacă se dorește și o reprezentare procentuală a distribuției notelor, se pot folosi funcțiile **pie** sau **pie3**:

```
>> pie3(hist(a) , [1 0 0 0 0 0 0 0 0 0]);
```

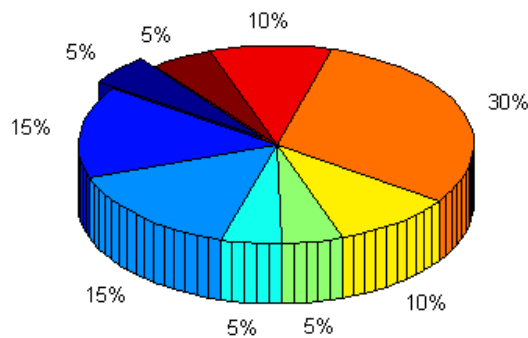


Figura 2.24. Reprezentarea procentuală a distribuțiilor acestora

Vectorul binar furnizat drept al doilea argument al funcției *pie3* este folosit pentru a scoate în evidență o anumită felie a graficului, în cazul de față cea corespunzătoare notei 1. De asemenea, trebuie observat că se afișează doar procentele nenule, adică nu se reprezintă nicio felie corespunzătoare notei 9, notă care nu a fost obținută de niciunul din studenți.

O reprezentare grafică similară celei obținute cu funcția *bar* se poate obține cu ajutorul funcției *stem*. De data aceasta dreptunghiurile vor fi înlocuite cu segmente, corespunzătoare fiecărei valori discrete. Fiecare punct poate fi marcat cu o formă stabilă de utilizator, conform specificațiilor de formatare din tabelul 2.1.

Reprezentări grafice foarte interesante și utile pot fi obținute cu ajutorul funcției **stem3**.

```
>> t = linspace(0,5,100);
>> plot3(sint(t), cost(t), t, 'r');
>> hold on;
>> stem3(sint(t), cos(t), t, 'd')
```

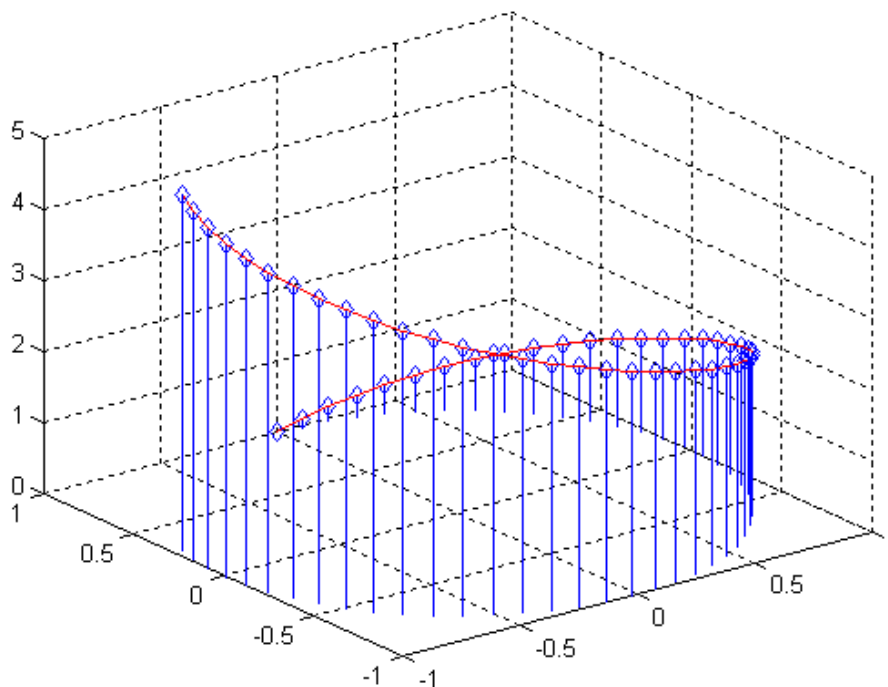


Figura 2.25. Reprezentarea unor puncte discrete de pe curba helix

Pentru reprezentarea grafică a poziției și vitezei unor obiecte, în probleme de mecanică și nu numai, extrem de utile se pot dovedi funcțiile pentru desenarea vectorilor: **compass**, **feather**, **quiver**, **quiver3**.

Exemplul 2.15. Un corp se deplasează după o traiectorie dată de următoarea ecuație:

$$r(t) = v t + a t^2 / 2$$

Să se reprezinte vectorul viteză în 10 puncte de pe traiectorie, în primele 2 secunde de la lansare, știind componentele accelerației $a_x = 0 \text{ m/s}^2$, $a_y = 0 \text{ m/s}^2$, $a_z = -32 \text{ m/s}^2$ și vitezele inițiale $v_{0x} = 2 \text{ m/s}$, $v_{0y} = 3 \text{ m/s}$, $v_{0z} = 10 \text{ m/s}$.

Soluție:

Secvența de comenzi MATLAB care rezolvă această problemă este următoarea

```
>> t = linspace(0,1,10);
>> az = -32; ax = 0; ay = 0;
>> v0x = 2; v0y = 3; v0z = 10;
>> rx = v0x.*t+1/2*ax*t.^2;
>> ry = v0y.*t+1/2*ay*t.^2;
>> rz = v0z.*t+1/2*az*t.^2;
>> vx = gradient(rx);
>> vy = gradient(ry);
>> vz = gradient(rz);
```

```

>> plot3(rx, ry, rz, 'r', 'LineWidth', 2);
>> hold on;
>> scale = 0;
>> quiver3(rx,ry,rz,vx,vy,vz,scale)
>> grid on;
>> view([70 18])

```

Reprezentarea grafică poate fi observată în figura următoare. Linia roșie reprezintă traiectoria iar săgețile reprezintă vectorul viteză în cele 10 puncte de pe traiectorie. Dacă se dorește și valoarea vitezei, aceasta se găsește stocată, pe componente, în elementele vectorilor v_x , v_y și v_z .

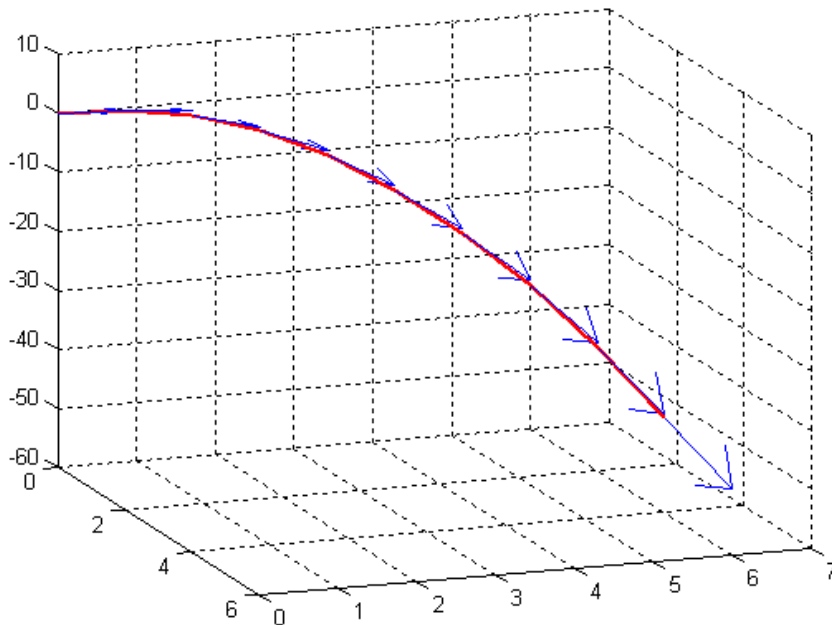


Figura 2.26. *Reprezentarea traiectoriei și a vitezei în 10 puncte*

2.4 Utilizarea instrumentelor de desenare pentru editarea graficelor

Sistemul MATLAB permite formatarea graficelor pentru o mai bună comprehensibilitate, permite configurarea scării de-a lungul axelor și includerea de marcaje pe acestea, utilizarea culorilor și a diferitelor tipuri de linii pentru a distinge cu ușurință anumite obiecte grafice.

Graficele obținute pot fi editate interactiv, atât în ceea ce privește modul de prezentare prin adăugarea de adnotări, titlu, comentarii, legende pentru un grafic deja construit cât și în ceea ce privește adăugarea de alte reprezentări grafice în aceeași fereastră, folosind utilitarul numit **Plot Tools**. Acest utilitar permite, de asemenea, controlarea multor aspecte ale reprezentărilor grafice prin adăugarea de noi zone grafice, vizualizarea grafică a variabilelor din spațiul de lucru, identificarea coordonatelor diferitelor puncte de pe grafice, mărirea sau micșorarea imaginii etc.

În MATLAB există două modalități prin care pot fi editate graficele:

- utilizând mouse-ul pentru a selecta și edita interactiv obiectele grafice create; în această abordare se selectează obiectele grafice (curbe, suprafețe, axe, texte, legendă etc.) care se doresc a fi editate și se efectuează dublu-clic pentru a activa editorul de proprietăți sau se efectuează clic-dreapta asupra obiectelor grafice după ce a fost pornit utilitarul *Plot Tools*.
- utilizând funcțiile MATLAB în linie de comandă sau într-un fișier script (M-file); dacă se preferă lucrul în linie de comandă sau cu fișiere script, această abordare presupune utilizarea sistemului de manipulare a obiectelor grafice oferit de MATLAB (*MATLAB Handle Graphics*). Acest sistem se bazează pe valoarea întoarsă de majoritatea funcțiilor grafice. Această valoare reprezintă un „mâner” (*handler*) al obiectele grafice și poate fi utilizat împreună cu comenzile **set** și **get** pentru a modifica proprietățile acestora.

Utilitarul *Plot Tools* poate fi accesat apăsând butonul marcat cu roșu în interfața grafică utilizator prezentată în figura următoare sau efectuând un dublu-clic asupra unui obiect grafic reprezentat în fereastră.

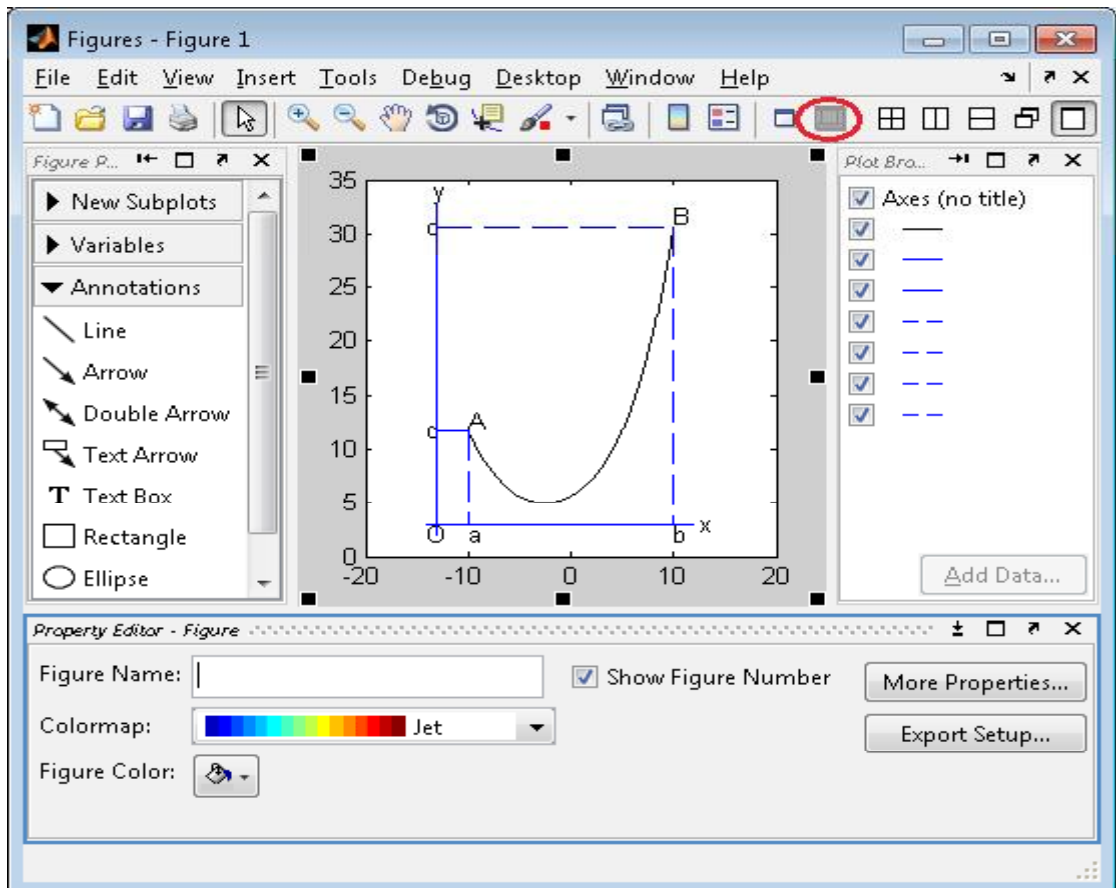


Figura 2.27. Utilitarul *Plot Tools*

2.5 Manipularea obiectelor grafice în MATLAB

Obiectele grafice reprezintă elementele de bază utilizate pentru a reprezenta grafice de funcții și elemente de interfață utilizator. Lista obiectelor grafice include, printre altele:

Obiect	Descriere
root	rădăcina ierarhiei, corespunzătoare ecranului calculatorului
figure	fereastra utilizată pentru reprezentarea graficelor și componentelor interfeței utilizator
axes	axele pentru reprezentarea grafică în cadrul unei figuri; cadrul care conține reprezentarea grafică
uicontrol	componentă a interfeței utilizator care răspunde la acțiunea acestuia
uimenu	meniul definit de utilizator în interfața grafică
uicontextmenu	meniul contextual <i>pop-up</i> care apare la apăsarea butonului dreapta al <i>mouse</i> -ului
image	imagini bidimensionale bazate pe un tablou de pixeli
light	sursă de lumină care afectează modul de colorare al unor obiecte grafice
line	obiect grafic liniar utilizat pentru reprezentări grafice de către funcții cum sunt <i>plot</i> , <i>plot3</i> etc.
patch	poligon 2D cu muchiile și interiorul colorate
rectangle	formă bidimensională care poate reprezenta un dreptunghi, un dreptunghi cu colțurile rotunjite sau o elipsă
surface	reprezentare tridimensională a unei matrici de date create prin interpretarea valorilor datelor ca fiind înălțimea deasupra planului Oxy
text	șir de caractere

Tabelul 2.5 Obiecte grafice

Aceste obiecte grafice sunt organizate într-o ierarhie reprezentată în diagrama următoare:

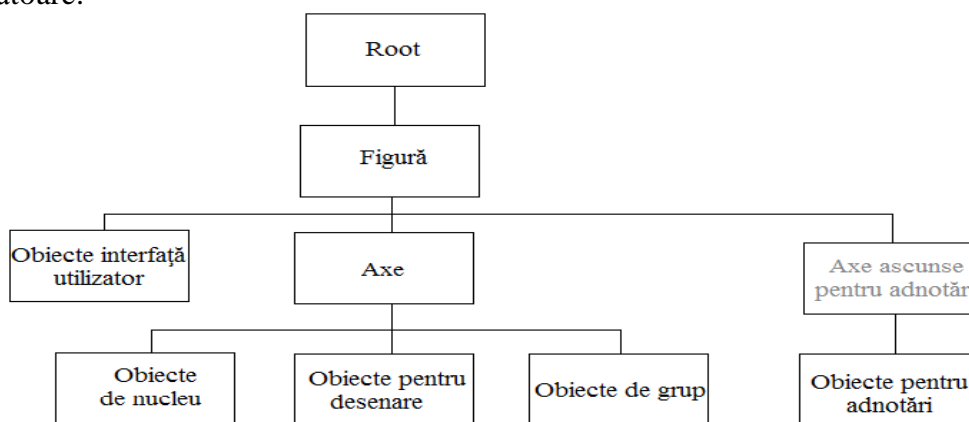


Figura 2.28. Ierarhia obiectelor grafice în MATLAB

Fiecare obiect grafic are asociată o funcție primitivă (*low-level*) care creează obiectul. Aceste funcții au aceleași nume cu obiectele pe care le creează. Funcțiile de nivel înalt, cum ar fi *plot*, apelează la funcțiile primitive corespunzătoare pentru a desena graficele respective.

Lista funcțiilor care pot fi utilizate în lucrul cu obiectele grafice include:

Funcție	Utilizare
allchild	găsește toate obiectele copil ale unui anumit obiect grafic
ancestor	găsește părintele unui anumit obiect grafic
copyobj	copiază un obiect grafic
delete	șterge un obiect grafic
findall	găsește toate obiectele grafice
findobj	găsește obiectele având o anumită proprietate
gca	returnează <i>handler</i> -ul axelor curente
gcf	returnează <i>handler</i> -ul figurii curente
gco	returnează <i>handler</i> -ul axelor curente
get	returnează valorile anumitor proprietăți ale obiectului grafic
set	stabilește valorile anumitor proprietăți ale obiectului grafic

Tabelul 2.6 Funcții pentru obiectele grafice

Exemplul 2.16. Să se găsească toate dreptunghiurile.

Soluție:

```
>> h = findobj('Type','rectangle');
```

Exemplul 2.17. Să se găsească toate liniile punctate de culoare roșie.

Soluție:

```
>> h = findobj('Type','line',...
    'Color','r','LineStyle',':');
```

Toate obiectele grafice au anumite proprietăți cu valori implicite. Totuși, în cazul în care se dorește ca un anumit obiect grafic să poată fi utilizat în aplicațiile MATLAB sau în cazul în care se dorește modificarea anumitor proprietăți ale sale se poate utiliza valoarea întoarsă de funcțiilor grafice. Stabilirea valorilor anumitor proprietăți ale obiectelor grafice se poate face fie la crearea acestora, fie ulterior, apelând la funcția **set**.

Ori de câte ori MATLAB creează un obiect grafic, pe lângă stabilirea proprietăților implicite sau a celor definite de utilizator, îi atribuie acestuia un identificator (*handler* - „mâner”). Această valoare poate fi utilizată pentru a configura ulterior obiectul grafic sau pentru a obține valorile anumitor proprietăți în scopul modificării aspectului unei reprezentări grafice sau pentru a crea utilitare personalizate pentru desenare sau manipulare directă a obiectelor grafice.

Exemplul 2.18. Să se modifice culoarea și transparența muchiilor reprezentării grafice din exemplul 2.9 (suprafața conică).

Soluție:

```
>> hndl = mesh(x,y,z);
>> set(hndl, 'EdgeColor', [0.6 0.6 0.6],...
    'EdgeAlpha', 0.5);
```

Pentru construirea unei interfețe grafice se pot utiliza comenzile care generează obiecte grafice de tip **uicontrol**:

```
>> buton = uicontrol('Style', 'pushbutton', ...  
                    'Position', [50 50 200 100], ...  
                    'String', 'Ajutor');
```

sau se poate apela la un instrument specializat de proiectare a unei interfețe grafice, *GUI Design Environment* sau *GUIDE* (utilizând comanda *guide* în fereastra de comenzi MATLAB) care simplifică crearea de interfețe utilizator (fig. 2.29).

După crearea și includerea componentei în fereastră se definește funcția de răspuns a acestuia, cea care se execută atunci când utilizatorul acționează asupra componentei grafice. Acest lucru se poate face stabilind încă de la crearea sa valoarea pentru proprietatea 'Callback' sau apelând la funcția *set*:

```
>> set(buton, 'Callback', 'help');
```

Astfel, ori de câte ori este acționată componenta grafică, se va afișa în fereastra de comenzi rezultatul comenzii *help*.

Dacă se dorește efectuarea unei alte acțiuni se va defini funcția respectivă și se va apela transmițând numele ei pentru funcția de răspuns 'Callback'.

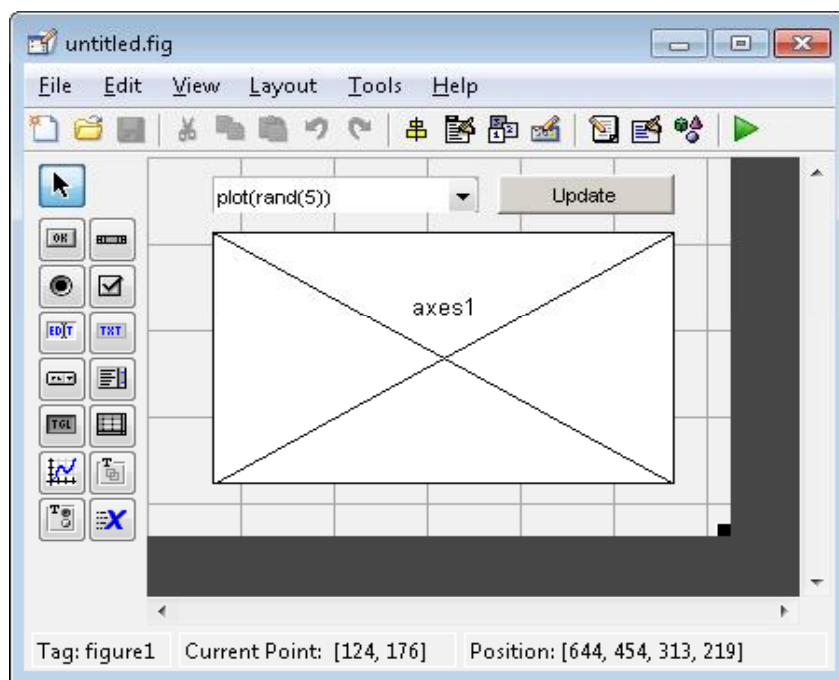


Figura 2.29. Utilitarul *GUIDE*

Pentru mai multe detalii privind crearea interfețelor grafice se pot consulta paginile de manual pentru funcțiile **uicontrol**, **uimenu**, **uicontextmenu** și **guide**.

2.6 Exerciții propuse spre rezolvare

Exercițiul 2.1. Să se reprezinte grafic, pe intervalul $[-10, 10]$, cât mai neted posibil, funcția pătratică:

$$y = x^2 + 3x + 2.$$

Exercițiul 2.2. Să se reprezinte grafic, pe intervalul $[-5, 5]$, cât mai neted posibil, funcția:

$$y = 1 / (1 + e^{-x}).$$

Exercițiul 2.3. Să se reprezinte pe același grafic funcțiile $y = \sin 2x$ și $y = \cos x$, pe intervalul $[0, 4\pi]$.

Exercițiul 2.4. Să se reprezinte grafic, pentru valori ale parametrului în intervalul $[0, 2\pi]$, curba definită de următoarele ecuații parametrice:

$$\begin{aligned}x &= 2 \cos 2t \\y &= 3 \sin 3t\end{aligned}$$

Exercițiul 2.5. Să se reprezinte grafic, mai întâi în coordonate polare și apoi în coordonate carteziane, funcția $r = 1 + 2 \cos \theta / 2$.

Exercițiul 2.6. Să se reprezinte grafic, pe intervalul $[-4, 4]$ curba dată prin ecuația

$$16y^2 = x^3(8 - x).$$

Exercițiul 2.7. Să se schițeze suprafața de ecuație
 $y = 1$
pe domeniul

$$D = \{(x, y) / -2 \leq x \leq 2, -1 \leq y \leq 1\}.$$

Exercițiul 2.8. Să se schițeze suprafața de ecuație
 $z = f(x, y) = 1 - x^2 - y^2$
pe domeniul

$$D = \{(x, y) / -3 \leq x, y \leq 3\}.$$

Exercițiul 2.9. Să se schițeze suprafața dată prin ecuațiile parametrice:
 $x = \sin f r \cos q$
 $y = \sin f \sin q$
 $z = \cos f$
în care $0 \leq f \leq \pi$, $0 \leq \theta \leq 2\pi$.

Exercițiul 2.10. Să se reprezinte suprafața dată prin ecuațiile parametrice:
 $x = (r + \cos u/2 \sin v - \sin u/2 \sin 2v) \cos u$
 $y = (r + \cos u/2 \sin v - \sin u/2 \sin 2v) \sin u$
 $z = \sin u/2 \sin v + \cos u/2 \sin 2v$
în care $r = 1$ și $0 \leq u, v \leq 2\pi$.

Exercițiul 2.11. Să se reprezinte grafic curba dată prin ecuațiile parametrice:

$$x = \frac{\cos t}{\sqrt{1+a^2t^2}}$$
$$y = \frac{\sin t}{\sqrt{1+a^2t^2}}$$
$$z = \frac{-at}{\sqrt{1+a^2t^2}}$$

în care $a = 1$ și $-12p \leq t \leq 12p$.

Exercițiul 2.12. Să se reprezinte grafic suprafața de ecuație:

$$r = q/2,$$

în sistemul de coordonate sferic, pentru domeniile de valori $0 \leq q \leq 4p$, $0 \leq f \leq p/2$.

Exercițiul 2.13. Să se reprezinte grafic suprafețele din exemplul 2.12 și exercițiul 2.12 în cazul în care se utilizează sistemul de coordonate cilindric, iar r este înlocuit cu coordonata z .

Capitolul 3

Modele statistice

În înțelegerea completă a realității este de multe ori necesar să cunoaștem, să înțelegem și să stăpânim legăturile existente între două sau mai multe fenomene, cuantificate prin variabile. Spre exemplu, în vederea aplicării unei strategii corecte de marketing, este necesar să cunoaștem dacă există o corelație între profitul unei întreprinderi și cheltuielile cu publicitatea și în ce fel se manifestă. De asemenea, pentru controlul investițiilor este necesar să cunoaștem în ce fel evoluează profitul pe o perioadă imediat următoare de timp. Asta presupune de fapt să putem construi și mai apoi utiliza pentru prognoză, așa numitele *modele statistice* sau *modele de regresie*, acestea fiind modele care descriu *corelația* existentă între două variabile oarecare și în caz particular, între o variabilă și timp. De cele mai multe ori, fundamentarea acestor modele are la bază un volum mare de date și aici își dovedesc utilitatea pachetele de programe concepute pentru a asista calculele de prognoză.

În modelarea prin regresie pornim de la următoarea situație: fiind date două variabile X și Y , studiate într-o populație A , se pune problema dacă între cele două variabile, respectiv între fenomenele descrise de acestea, există o anumită dependență numită și corelație. O primă concluzie se poate obține reprezentând grafic într-un sistem de coordonate XY , cele două șiruri de date observate la nivelul populației pentru cele două variabile (corelograma). Dacă punctele graficului se împrăștie pe toată suprafața fără a urma o anumită regulă, atunci vom spune că cele două variabile nu sunt corelate. Dacă în schimb punctele descriu o anumită curbă, numita și *curbă empirică de regresie*, atunci vom spune că există corelație și ea este cu atât mai intensă, cu cât domeniul pe care se întind punctele este mai îngust. Mai mult, dacă punctele se așează pe o curbă care poate fi aproximată de o curbă clasică (dreaptă, parabolă, exponențială, etc.) atunci vom spune că legătura dintre cele două variabile este una liniară sau parabolică sau exponențială, etc. și vom folosi ecuația acelei curbe clasice pentru prognoză.

O fundamentare riguroasă a existenței unei corelații și mai apoi a modelului care descrie corelația, numit și *model de regresie*, se poate face pe baza calculului și interpretării unor indicatori statistici. Se vor parcurge câteva etape, precum: verificarea existenței unei corelații, stabilirea formei matematice a modelului, în urma analizei curbei empirice de regresie, determinarea parametrilor care intervin în ecuația modelului și utilizarea modelului pentru calcule de prognoză. Există și parametri care masoară corelația (gradul de asociere) între două variabile calitative, parametri bazați pe frecvențele de apariție ale valorilor variabilelor și nu pe valori. Odată stabilită existența corelației între variabile, se poate trece la stabilirea modelului de regresie care descrie corelația. Un caz particular este acela când cuplul X, Y este înlocuit de cuplul t, Z , în care t reprezintă timpul și Z reprezintă o variabilă urmărită în timp. Un *model cronologic*, este de fapt un model de regresie în care se studiază dependența unui fenomen de timp sau mai bine zis, evoluția sa în timp. În calcule, variabila timp t ia valorile implicite 1, 2, 3, ... iar prognoza pe baza unui astfel de model presupune de fapt previziunea fenomenului pe o perioadă următoare de timp, care nu a avut încă loc. Desigur, atât în cazul unui model de regresie oarecare, cât și în cazul unui model cronologic, prognoza pe baza modelului este cu atât mai veridică, cu cât modelul este mai bine ales să se potrivească datelor și cu cât orizontul de previziune este mai mic. Spre exemplu, dacă ne hazardăm să previzionăm un fenomen

pe o perioadă foarte îndepărtată de timp, față de perioada din care s-au cules datele pe care se bazează modelul, atunci riscăm ca eroarea de previziune să fie foarte mare.

În cele ce urmează vom vedea pe parcursul a patru paragrafe, atât elementele teoretice pentru fundamentarea unui model de regresie, cât și suportul computațional asigurat de pachetele specializate din MATLAB 6.5., o mare parte din fundamentarea teoretică și aplicațiile prezentate aici fiind preluate din lucrările [5] și [6]. Ultimele trei paragrafe ale acestui capitol se finalizează cu câte o temă propusă pentru laborator. De asemenea, pe parcursul capitolului sunt prezentate șase probleme reale, care sunt apelate cu aceeași notație în tot capitolul, fie ca justificări ale unor modele, fie ca bază pentru aplicațiile expuse, respectiv pentru temele propuse pentru laborator.

3.1. Problema regresiei

Modelele de regresie fac parte din categoria *modelelor stochastice* (statistice), în care toți factorii explicativi ai unui fenomen, care nu își găsesc locul direct în model, apar cumulați sub forma unei variabile aleatoare numită *eroare*. O variabilă Y (*parametrul de ieșire*) care cuantifică fenomenul studiat poate fi explicată prin *regresia* pe unul sau mai mulți *factori explicativi* (*parametrii de intrare*). Toți factorii explicativi care nu sunt suficient de relevanți pentru Y , intră în model sub forma cumulativă a erorii.

În cazul când factorii explicativi relevanți se rezumă la un singur factor X , avem de-a face cu *modelul de regresie simplă*, de forma,

$$Y = f(X) + e, \quad (3.1)$$

unde e reprezintă *eroarea* din model, componentă care înglobează toți ceilalți factori de care depinde y , în afara lui x , iar f este funcția care descrie legătura dintre variabile, numită și *funcție de regresie*.

Dacă avem doi sau mai mulți factori explicativi (predictivi), X_1, X_2, \dots, X_p , atunci *regresia* se numește *multiplă* și modelul corespunzător va fi

$$Y = f(X_1, X_2, \dots, X_p) + e \quad (3.2)$$

Problema regresiei pleacă de la existența unui set de date privind două sau mai multe variabile aleatoare, scopul modelării fiind descrierea relației dintre ele, adică determinarea funcției f , în vederea prognozării valorilor variabilei dependente în raport cu valorile variabilelor explicative. Această problemă se pune doar atunci când între variabile există o legătură reală, bazată pe natura fenomenelor care stau la baza lor. Altfel, este posibil ca formal, datele numerice să pară corelate (spre exemplu, cresc pe aceeași perioadă de timp), fără ca fenomenele pe care le cuantifică să fie corelate. Într-un astfel de caz, studiul regresiei ar fi lipsit de sens. În cazul în care, funcția de regresie este parametrică, adică este de forma, $f(X_1, X_2, \dots, X_p, a_1, a_2, \dots, a_s)$, atunci determinarea funcției revine la determinarea parametrilor, a_1, a_2, \dots, a_s . Condiția naturală care apare este ca abaterile valorilor rezultate din model față de valorile empirice, rezultate prin observare, să fie minime. Cel mai cunoscut criteriu care permite satisfacerea unei astfel de condiții este *criteriul*

celor mai mici pătrate, care constă în minimizarea sumei pătratelor acelor abateri. Astfel, parametrii optimi vor fi aceia pentru care, modelul satisface condiția,

$$\sum_{i=1}^n e_i^2 = \min, \quad n - \text{numarul de date}, \quad (3.3)$$

condiție ce revine la

$$F(a_1, a_2, \dots, a_s) = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - f(x_{1i}, x_{2i}, \dots, x_{pi}, a_1, a_2, \dots, a_s))^2 = \min. \quad (3.4)$$

Mai departe, problema de minim revine la condițiile:

$$\frac{\partial F(a_1, a_2, \dots, a_s)}{\partial a_i} = 0, \quad i = \overline{1, n} \quad (3.5)$$

Într-un model de regresie simplă, pentru o valoare observată x_i a lui X , avem două tipuri de date corespunzătoare lui Y : un y_i observat și un $y_i' = f(x_i) = \overline{y(x)}$, calculat pe baza modelului. Determinarea funcției de regresie constă în acest caz în etapele :

-alegerea unei clase de funcții, care aproximează cel mai bine realitatea datelor reprezentată grafic prin norul statistic (corelograma între x și y) ; de exemplu dreapta, parabola, exponentiala, etc

-determinarea curbei de regresie (a parametrilor ei) ca fiind aceea curbă din clasa stabilită care satisface criteriul celor mai mici pătrate.

În cazul regresiei simple, criteriul celor mai mici pătrate (suma pătratelor abaterilor valorilor observate față de valorile rezultate din model este minimă) se scrie de forma :

$$\sum_{i=1}^N e_i^2 = \sum_{i=1}^N [y_i - f(x_i)]^2 = \min. \quad (3.6)$$

Cele mai simple *modele* de regresie sunt cele *liniare*, adică cele în care f depinde liniar de variabilele predictive. O mare parte dintre dependențele reale sunt de tip liniar sau pot fi reduse la modelul liniar motiv pentru care studierea unui astfel de model ocupă un loc important în modelarea matematică.

3.2. Modelul de regresie liniară simplă

3.2.1. Definirea modelului liniar simplu

Vom considera în cele ce urmează, cazul unui model de regresie liniară simplă, plecând de la presupunerea că forma corelogramei între Y și X (o singură variabilă factor) a indicat dreapta ca cea mai potrivită curbă de regresie.

Modelul de regresie liniară simplă este de forma

$$Y = a + bX + e, \quad (3.7)$$

unde a și b se numesc *parametrii dreptei de regresie* (a –*termenul liber* sau *intersecția dreptei cu axa OY* și b –*panta dreptei*) X se numește *predictor* sau *variabilă explicativă*, Y , *variabilă răspuns* sau *efect* iar e , *eroare* sau *reziduu*.

Problema regresiei liniare simple: Se consideră un set bidimensional de date $x_i, y_i, i = \overline{1, n}$, reprezentând valori observate ale celor două variabile, X și Y . În vederea prognozării valorilor lui Y corespunzătoare unor valori noi ale lui X (altele decât cele n observate), se pune problema determinării parametrilor de regresie a și b , în așa fel, încât dintre toate dreptele posibile, dreapta de ecuație,

$$\hat{y} = f(x) = a + bx, \quad (3.8)$$

să descrie cel mai bine legătura dintre variabile și față de care punctele de coordonate $x_i, y_i, i = \overline{1, n}$, să se apropie cel mai mult. Deoarece în realitate, mai există și alți factori predictivi pentru Y , pe lângă X , dependența directă dintre X și Y nu este una matematică, pur funcțională, ci una statistică (stochastică) de forma,

$$y(x) = a + bx + e. \quad (3.9)$$

Apariția erorii e este evidențiată de acest aspect, eroarea înglobând practic efectul celorlalți factori de influență asupra lui Y , diferiți de X , plecându-se de fapt de la premisa că influența acestora și prin urmare și eroarea este neglijabilă. Din punct de vedere tehnic, apariția componentei e , în model, este motivată și de faptul că unei singure valori a lui X , îi corespunde nu o singură valoare a lui Y (ca în cazul unei legături funcționale) ci o serie de valori ale lui Y . Spre exemplu, pentru un venit de 2500 lei, la nivelul unei populații se înregistrează mai multe valori ale cheltuielilor. Totuși, unei singure valori a lui X , îi corespunde, în mod unic, o singură valoare medie a lui Y (media seriei de valori ale lui Y corespunzătoare valorii fixate pentru X , adică media condiționată a lui Y , de către valoarea fixată a lui X). Prin urmare, legătura directă, matematică, se poate stabili între X și media lui Y , deci avem

$$\hat{y} = f(x) = \bar{y}(x) = a + bx. \quad (3.10)$$

Cum, în general, valoarea unei variabile diferă de medie cu o anumită eroare, neglijabilă când media este reprezentativă, vom avea

$$y(x) = \bar{y}(x) + e. \quad (3.11)$$

Prin urmare, vom face referire la modelul liniar prin una din următoarele forme

$$\hat{y} = f(x) = \bar{y}(x) = a + bx, \quad (3.12)$$

$$y(x) = a + bx + e. \quad (3.13)$$

Determinarea și utilizarea modelului pentru previziuni presupune următoarele tehnici de regresie liniară simplă: identificarea modelului, estimarea parametrilor modelului prin metoda celor mai mici pătrate-ajustarea modelului, validarea modelului și în cele din urmă utilizarea lui pentru calcule de prognoză.

3.2.2. Verificarea existenței unei corelații și identificarea modelului

Un rol important în identificarea unei corelații și a formei ei îl joacă așa numitul coeficient de corelație liniară, precum și raportul de corelație (determinare).

Coeficientul de corelație liniară: Pentru a analiza dacă între variabilele cantitative X și Y există o legătură (de tip liniar) se calculează *coeficientul de corelație liniară*, dat de formula:

$$r = \frac{\text{cov}(x, y)}{S_x \cdot S_y} \in [-1, 1], \quad (3.14)$$

unde $\text{cov}(x, y) = M(x \cdot y) - M(x) \cdot M(y)$ cu $M(x \cdot y) = \frac{\sum_{i=1}^N x_i \cdot y_i}{N}$ se numește *covarianță*. Aici, $x_i, y_i, i = \overline{1, N}$ sunt valorile celor două variabile, observate pe populația de volum N , iar $M(\)$ și S reprezintă media și abaterea medie pătratică. În cazul când cele două variabile sunt independente, covarianța este nulă.

Interpretarea coeficientului de corelație liniară: Valorile coeficientului de corelație sunt în intervalul $[-1, 1]$. Dacă $r = 0$, între cele două variabile nu există corelație. Dacă $r = 1$, corelația între cele două variabile este maximă și directă. Dacă $r = -1$, corelația între cele două variabile este maximă și inversă. Cu cât avem o valoare mai apropiată de 1 sau -1 cu atât corelația e mai puternică (directă pentru valori pozitive și inversă pentru valori negative), cu cât avem o valoare apropiată de 0 corelația este mai slabă.

Raportul de corelație: Pentru fundamentarea raportului de corelație, R , se pornește de la următoarea relație numită regula de adunare a varianțelor:

$$\text{Variația totală} = \text{variația explicată prin regresie} + \text{variația reziduală},$$

unde variația totală este exprimată de suma totală de pătrate, $S_T = \sum_{i=1}^N [y_i - \bar{y}]^2$,

variația reziduală este exprimată de suma reziduurilor pătratice, $S_R = \sum_{i=1}^N e^2$ iar

variația explicată este exprimată de diferența între cele două sume,

$S_E = S_T - S_R = \sum_{i=1}^N [y_i - \bar{y}]^2 - \sum_{i=1}^N e^2$. *Raportul de corelație* în care variația lui y se exprimă prin regresia pe x este dat de

$$R^2 = \frac{S_E}{S_T}. \quad (3.15)$$

Interpretare: Dacă acest raport este peste 0,5 atunci modelul de regresie se consideră potrivit la date, o valoare egală cu 1 indicând potrivire maximă, iar una egală cu 0, nepotrivire a modelului la date. Un model de regresie adecvat datelor este caracterizat de S_E mare, S_R mic, R^2 mare (apropiat de 1).

În ce privește *identificarea modelului*, se justifică alegerea unui model liniar (identificarea modelului) prin:

- calculul *coeficientului de corelație liniară* pe baza datelor, $x_i, y_i, i = \overline{1, n}$ și compararea valorii sale absolute cu 1 (pentru ca modelul liniar să fie potrivit datelor, coeficientul de corelație liniară trebuie să aibă o valoare absolută cât mai apropiată de 1);

- *reprezentarea punctelor* de coordonate $x_i, y_i, i = \overline{1, n}$ - corelograma asociată datelor și interpretarea ei (într-un sistem ortogonal de axe-alura liniei poligonale obținute prin unirea cu segmente a punctelor, trebuie să fie cât mai apropiată de cea a unei drepte).

3.2.3. Ajustarea modelului prin determinarea unor estimatori ai parametrilor săi

Forma modelului este

$$Y = a + bX + e, \tag{3.16}$$

iar pentru $x_i, y_i, i = \overline{1, n}$, devine,

$$y_i = a + bx_i + e_i, i = \overline{1, n}. \tag{3.17}$$

Se pune *problema estimării parametrilor a și b*, astfel încât dreapta determinată de aceștia să minimizeze suma pătratelor reziduurilor, e_i , (abaterile valorilor reale y_i , față de valorile rezultate din model, $\hat{y}_i = a + bx_i$)-*criteriul celor mai mici pătrate*. Criteriul celor mai mici pătrate particularizat pe cazul modelului liniar simplu constă în faptul că dintre toate dreptele posibile o vom alege pe aceea, specificată printr-un anumit a și b , față de care suma pătratelor abaterilor punctelor care descriu datele este minimă. Un astfel de criteriu minimizează de fapt, global, diferența, $y - f(x)$, unde $f(x) = a + bx$. Prin urmare, din condiția,

$$\sum_{i=1}^N e_i^2 = \sum_{i=1}^N [y_i - a - bx_i]^2 = \min \tag{3.18}$$

ilustrată și în graficul de mai jos,

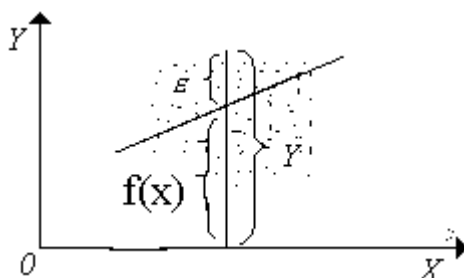


Figura 3.1. Corelograma și dreapta de regresie

rezultă *estimatorii de cele mai mici pătrate ai parametrilor de regresie* (valorile reale pentru a și b , nu pot fi determinate exact deoarece, dispunem doar de eșantionul de observații, $x_i, y_i, i = \overline{1, n}$),

$$\begin{aligned}\hat{b} &= \frac{\text{cov}(x, y)}{s_x^2} \\ \hat{a} &= \bar{y} - \bar{x} \cdot \hat{b}.\end{aligned}\tag{3.19}$$

Pe lângă aceste estimări punctuale este necesar să cunoaștem și *inferențele asupra acestor parametrii*, mai precis intervalele de încredere corespunzătoare. Un cadru bun pentru astfel de estimări presupune câteva condiții asupra modelului, numite și *ipotezele Gauss-Markov*.

3.2.4. Ipotezele fundamentale-Gauss-Markov. Validarea și estimarea modelului

Vom prezenta în acest capitol modelul liniar simplu clasic cu erori normale, independente și identic distribuite, de medie zero, rezultat în urma aplicării următoarelor condiții, numite și ipotezele fundamentale-Gauss-Markov :

- normalitatea erorilor : $e_i \in N, \forall i = \overline{1, n}$;
- independența erorilor : $\text{cov}(e_i, e_j) = 0, \forall i, j = \overline{1, n}, i \neq j$;
- erori identic distribuite (model homoscedastic) : $M(e_i) = 0, V(e_i) = s^2, \forall i = \overline{1, n}$.

Altfel spus, reziduurile trebuie să aibă comportamentul „zgomotului gaussian alb”. Toate aceste ipoteze pot fi verificate printr-o serie de teste statistice, care presupun validarea modelului. Orice abatere de la aceste ipoteze duce la alte tipuri de modele liniare, decât cel clasic. O metodă de verificare este și cea prin care se *proiectează reziduurile* și se analizează dacă *graficul este cel al unui zgomot alb-oscilații în jurul lui 0*.

În problema stabilirii modelului de regresie între variabilele X și Y , apare aspectul privind extinderea informațiilor de la nivelul eșantionului pe care se presupune ca avem datele pentru X și Y , la nivelul populației. Astfel, toate rezultatele obținute (valorile parametrilor de regresie a și b , valorile indicatorilor calității regresiei, r, R, c^2) se consideră doar niște estimatori pentru adevăratele valori valabile pe întreaga populație. În ipotezele Gauss-Markov, se pot realiza inferențe asupra parametrilor modelului, atât prin determinarea intervalelor de încredere, cât și prin teste de ipoteză asupra modelului. De asemenea se pot face teste de validare asupra modelului, testând existența corelației la nivelul întregii populații, precum și potrivirea modelului liniar la date.

Intervale de încredere de tip $1 - \alpha$, pentru coeficienții de regresie sunt de forma:

$$P\left(\hat{b} - s_b t_{n-2, 1-\frac{\alpha}{2}} < b < \hat{b} + s_b t_{n-2, 1-\frac{\alpha}{2}}\right) = 1 - \alpha,\tag{3.20}$$

$$P\left(\hat{a} - s_a t_{n-2, 1-\frac{a}{2}} < a < \hat{a} + s_a t_{n-2, 1-\frac{a}{2}}\right) = 1 - a, \quad (3.21)$$

unde $t_{n-2, 1-\frac{a}{2}}$ este cuantila de ordin $1 - \frac{a}{2}$, pentru o variabilă de tip Student cu $n-2$ grade de libertate, iar

$$s_b^2 = \frac{s^2}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad s_a^2 = \frac{s^2}{n} \frac{\sum_{i=1}^n x_i^2}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad s^2 = \frac{1}{n-2} \sum_{i=1}^n e_i^2 = \frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{a} - \hat{b}x_i)^2. \quad (3.22)$$

Teste T de semnificație a parametrilor modelului : În cazul parametrului b , valoarea corespunzătoare eșantionului este valoarea \hat{b} . Pentru această valoare se poate vedea dacă e diferită de 0, adică dacă dependența lui Y de X există (dacă ar fi 0 atunci $y = 0x + b = b$, deci Y nu ar depinde de X). Se pune problema dacă această concluzie, se poate extinde și la nivelul populației, adică pentru b . Vom testa atunci ipoteza nulă $H_0 : b = 0$, plecând de la realitatea că pe eșantion, situația e următoarea : $\hat{m} \neq 0$. Pentru a testa ipoteza se aplica un test de tip T (bazat pe legea Student). Decizia, adică acceptarea sau respingerea ipotezei se va lua cu un risc (prag de semnificație) a , pe care îl alegem mai mic decât 0,05. Pentru parametrul a se procedează la fel ca și mai sus doar că semnificația ipotezei $H_0 : a = 0$, se referă la existența termenului liber.

Testul F de semnificație a regresiei: Modelul propus este semnificativ (potrivit) pentru datele eșantionului dacă valoarea lui \hat{R} (raportul de corelație) este cât mai aproape de 1 și mai departe de 0. Pentru a vedea dacă acest rezultat se păstrează și pe întreaga populație, se face un test de tip F , bazat pe legea Fisher, pentru un prag de semnificație a , pe care îl alegem mai mic decât 0,05. Ipoteza pe care o vom testa se referă la adevărata valoare a lui R , valabilă pe întreaga populație. Mai precis, testăm ipoteza $H_0 : R = 0$, ceea ce ar presupune că la nivelul întregii ipoteze modelul nu este semnificativ datelor.

3.2.5. Utilizarea modelului pentru calcule de prognoză

Odată specificată forma modelului, împreună cu valoarea numerică a parametrilor a și b , putem utiliza modelul pentru prognoză. Astfel, pentru o nouă valoare, neobservată, a lui X , notată cu x_0 , valoarea lui Y corespunzătoare, conform modelului de regresie stabilit, va fi

$$y_0 = y(x_0) = a + bx_0 + e, \quad (3.23)$$

respectiv, valoarea sa medie,

$$y_p = \overline{y(x_0)} = a + bx_0, \quad (3.24)$$

aceasta din urmă numindu-se și valoare previzionată. Practic, calculul de prognoză răspunde la întrebarea: “Care este valoarea medie a lui y pentru o valoare nouă, neobservată (în afara celor N valori), a lui x , egală cu x_0 ?” Se observă că nu se poate calcula, exact, decât valoarea medie, valoarea reală depinzând de eroarea e , care nu poate fi calculată ci doar presupusă a fi neglijabilă (minimă). Ținând cont de faptul că nu se cunosc adevăratele valori ale lui a și b (la nivelul populației) ci doar estimatorii \hat{a} și \hat{b} , previziunea de mai sus este de fapt de forma

$$\hat{y}_p = \hat{a} + \hat{b} \cdot x_0. \quad (3.25)$$

Pentru valoarea y se poate determina următorul *interval de încredere pentru previziune* :

$$P\left(\hat{y} - s_0 t_{n-2, 1-\frac{\alpha}{2}} < y < \hat{y} + s_0 t_{n-2, 1-\frac{\alpha}{2}}\right) = 1 - \alpha, \quad (3.26)$$

unde

$$s_0^2 = \frac{s^2}{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}. \quad (3.27)$$

Pe același grafic se pot reprezenta *dreapta de regresie și curbele ce corespund capetelor intervalului de încredere pentru o valoare previzionată*.

3.2.6. Funcții specifice și aplicații privind modelul liniar simplu în MATLAB

În cele ce urmează vom prezenta câteva funcții MATLAB, utile în implementarea tehnicilor de regresie, pentru modelul liniar simplu. Nu vom prezenta aici sintaxele cele mai generale ale funcțiilor ci doar cele care sunt mai potrivite în cadrul modelării matematice. Cele mai utilizate funcții în acest sens sunt funcțiile *corrcoef*, *polyfit*, *polyval*, *refline*, *polytool*. Funcțiile *corrcoef*, *polyfit*, *polyval*, *polytool* se regăsesc printre funcțiile Matlab generale, în timp ce funcția *refline* este o funcție specifică pachetului Matlab “Statistics”

Sintaxa funcției corrcoef: Funcția *corrcoef* se utilizează pentru a calcula coeficientul de corelație liniară dintre două variabile:

```
>>rx = corrcoef(x,y)
```

- *parametrii de intrare*: x -matrice de tipul (n,1)-coloană; y -matrice de tipul (n,1)-coloană, matrici care stochează valorile celor două variabile pentru care se calculează coeficientul de corelație;

- *parametrii de ieșire*: rx -matrice de tipul (2,2) care conține pe poziția 11 și 22 coeficientul de corelație dintre variabila x și x , respectiv y și y , adică valoarea 1, iar pe poziția 12 și 21 coeficientul de corelație dintre x și y , respectiv y și x , care sunt de fapt, valori egale între ele, coeficientul de corelație fiind o funcție simetrică.

Prin urmare, valoarea coeficientului de corelație între x și y se citește pe diagonala a doua a matricii răspuns, indicând o corelație mare cu atât mai mult cu cât se apropie de 1, în modul, respectiv, o corelație mică, atunci când se apropie mai mult de 0.

Pentru a putea analiza grafic corelația dintre puncte, respectiv variabile se poate utiliza funcția `plot`, cu sintaxa `plot(x,y)` care va desena linia poligonală dată de punctele de coordonate x, y sau `plot(x, y, '*')` care afișează graficul de puncte x, y . În urma analizei formei acsetui grafic putem afirma cu o oarecare precizie dacă e vorba de o corelație liniară (curba poligonală poate fi ușor aproximată cu o dreaptă) sau nu. În cazul în care răspunsul este afirmativ se poate afișa pe același grafic și dreapta de regresie, utilizând funcția `refline`.

Sintaxa funcției `refline`:

```
>>refline
```

Comanda de mai sus, trebuie dată după comanda `plot`, înainte de închiderea figurii generate și reprezintă grafic în figura curentă, obținută cu funcția `plot(x,y, '*')`, dreapta de regresie corespunzătoare punctelor.

Sintaxa funcției `polyfit` – modelul liniar : Funcția `polyfit` calculează parametrii drepte de regresie corespunzătoare datelor, utilizând metoda celor mai mici pătrate.

```
>>p = polyfit(x, y, 1)
```

- *parametrii de intrare*: x, y - vectori-matrici coloană de aceeași dimensiune; 1-reprezintă gradul 1 al polinomului de regresie, grad care corespunde modelului liniar;

- *parametrii de ieșire*: p -vector de dimensiune 2, cu prima componentă reprezentând panta drepte de regresie, iar a doua, termenul liber sau altfel spus, prima componentă este coeficientul lui x iar a doua este termenul liber din polinomul $p_1 \cdot x + p_2$.

Sintaxa funcției `polyval` – modelul liniar: Funcția `polyval` calculează previziunea pe baza drepte de regresie, determinată de `polyfit`.

```
>>ypreviz = polyval(p, xnou)
```

- *parametrii de intrare*: p -vector de dimensiune 2, care stochează parametrii drepte de regresie, începând cu coeficientul lui X și terminând cu termenul liber; reprezintă rezultatul funcției `polyfit`; $xnou$ -matrice care conține valorile noi ale variabilei explicative, valori pe care se dorește a fi calculată previziunea;

- *parametrii de ieșire*: $ypreviz$ -matrice de aceeași dimensiune cu $xnou$, care conține valorile variabilei răspuns y (corespunzătoare valorilor noi ale lui x), previzionate din dreapta de regresie dată prin vectorul coeficienților de regresie, p .

Sintaxa funcției `polytool`– modelul liniar : Funcția `polytool` creează o interfață grafică pentru utilizator, care permite controlul ajustării de cele mai mici pătrate printr-un polinom, implicit și printr-o dreaptă. Astfel, funcția `polytool` acoperă toate facilitățile oferite de funcțiile de mai sus, prin integrarea graficelor și a răspunsurilor numerice într-o interfață interactivă:

```
>>polytool(x, y, 1, alfa)
```

- *parametrii de intrare*: x , y - vectori - matrici coloană - de aceeași dimensiune; 1-reprezintă gradul 1 al polinomului de regresie, grad care corespunde modelului liniar; alfa reprezintă pragul de semnificație admis pentru intervalele de încredere determinate pentru coeficienții modelului și pentru previziune;

- *parametrii de ieșire*: informațiile fundamentale legate de modelul liniar de regresie simplă (coeficienți, reprezentare grafică, previziune, intervale de încredere, etc.), încorporate într-o interfață grafică interactivă; mai precis, funcția *polytool* creează un grafic interactiv al dreptei de regresie pentru variabila răspuns y și variabila explicativă x , împreună cu punctele observate și cu intervalele de încredere de tip $(1 - \alpha)100\%$ pentru previziune (valoarea implicita este de 95%), precum și cu câteva informații numerice legate de modelul liniar; interfața grafică permite controlul asupra alegerii valorii pentru care dorim previziunea și exportul unor parametri precum valoarea previzionată împreună cu intervalul de încredere, parametrii de regresie împreună cu intervalele de încredere și reziduurile.

Vom prezenta în cele ce urmează două probleme reale preluate din [9], care reclamă modelarea, acestea fiind reluate și abordate prin diverse metode de modelare și în următoarele paragafe. Se va vedea în cele din urmă, că modelul liniar se pretează foarte bine la prima problemă, în timp ce pentru cea de-a doua problemă este necesar și studiul altor modele. Totuși, ori de câte ori, particularitatea fenomenului pe care încercăm să-l modelăm nu reclamă strict o anumită formă matematică a modelului, se poate pleca de la testarea datelor cu un model liniar, tehnicile asociate aceluia fiind foarte simple și apoi se poate concluziona dacă datele necesită sau nu un model mai complicat.

Problema 1. Corelația dintre greutatea și înălțimea unei persoane

În mod natural, se presupune că există o strânsă legătură între *înălțimea* și *greutatea* unei persoane. Se pune problema determinării unui model regresiv care să permită *calcularea greutății normale a unei persoane, în raport cu înălțimea sa*. Vom considera un lot de 15 persoane, pentru care se înregistrează înălțimea și greutatea ([9]).

Nr.	1	2	3	4	5	6	7	
înălțime	1.67	1.8	1.76	1.68	1.78	1.76	1.67	
greutate	70	82	75	69	82	78	65	
Nr.	8	9	10	11	12	13	14	15
înălțime	1.82	2	1.67	1.56	1.78	1.82	1.6	1.76
greutate	95	113	67	61	81	90	62	78

Tabelul 3.1. Date privind înălțimea și greutatea a 15 persoane

Încărcarea datelor în MATLAB se poate face fie direct de la tastatură, fie prin importarea lor dintr-un fișier de tip *ascii*, cu extensia „*dat*”. Următorul program (*aplicatie1.m*), permite încărcarea datelor stocate în fișierul *hgdate.dat*, ordonarea

crescătoare a datelor după înălțime, definirea variabilelor de lucru x și y , pe baza variabilei MATLAB, $hgdate$ și apoi reprezentarea grafică a greutății versus înălțime.

```
>>%incarca datele pentru inaltime si greutate dintr-
un fisier cu extensia 'dat', se creeaza astfel
variabila hgdate-matrice de tip (15, 2)
>>load hgdate.dat -ascii
>>%ordoneaza datele dupa inaltime (prima coloana)si
le pune in matricea 'dateord'
>>dateord=sortrows(hgdate);
>>%defineste variabila X ca fiind prima coloana din
matricea datelor ordonate
>>x=dateord(1:15,1);
>>%defineste variabila Y ca fiind a doua coloana din
matricea datelor ordonate
>>y=dateord(1:15,2);
>>% reprezinta datele intr-un sistem de coordonate
>>plot(x,y)
```

Se obține următorul grafic:

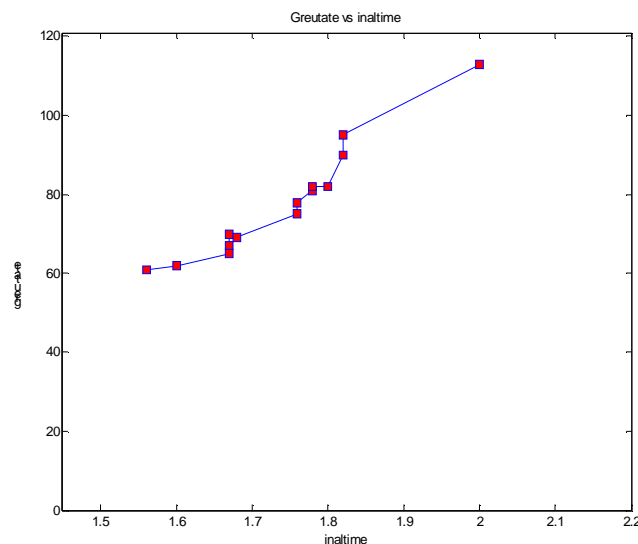


Figura 3.2. Diagrama corelației dintre greutate și înălțime

Din analiza graficului se poate avansa ipoteza unei regresii liniare simple între greutate și înălțime. Mai mult, calculând coeficientul de corelație cu funcția *corrcoef*, obținem $r = 0.97$ (0.9668), deci o valoare foarte apropiată de 1, prin urmare o corelație puternică. Apelând funcția *polytool*, obținem o serie de informații relativ la modelul liniar de regresie. Parametrii de regresie sunt 125.84 și -141.35. Așadar, avem modelul $y = -141.35 + 125.84x + e$. Având în vedere că parametrii s-au obținut pe baza unui eșantion de volum 15, cele două valori reprezintă doar niște estimări ale adevăratelor valori, pentru care se obțin intervalele de încredere pentru coeficienți (parametri de regresie): (105.92, -176.12) și (145.77, -106.59) adică, $P(105.92 < b < 145.77) = 0,95$ respectiv $P(-176.12 < a < -106.59) = 0,95$. Pentru cele 15 perechi de valori observate se pot calcula și reziduurile, adică diferența între valoarea reală a lui y_i , măsurată și valoarea teoretică, rezultată din model,

$\hat{y}_i = -141.35 + 125.84x_i$. Se obțin următoarele valori reziduale, numeric și apoi grafic cu funcția *plot*: 6.0371, 2.0033, -3.8058, -1.8058, 1.1942, -1.0643, -5.1319, -2.1319, -2.1319, -1.6488, -0.64876, -3.1657, 2.3174, 7.3174, 2.6654.

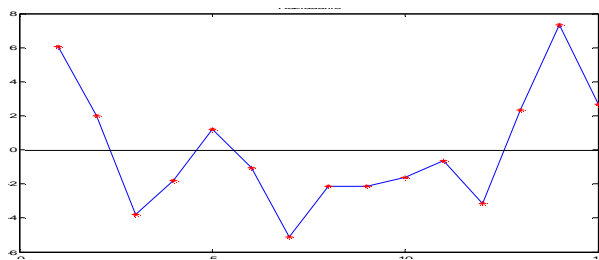


Figura 3.3. Graficul rezidurilor

Rezidurile vor descrie un model cu atât mai bun cu cât sunt mai apropiate de axa 0 și nu manifestă o tendință, în sensul că două puncte consecutive nu sunt ambele pozitive sau ambele negative. În acest caz, reziduurile arată că ar putea exista un alt model, care să se preteze mai bine la date. Odată determinat modelul se poate folosi pentru previziune. Spre exemplu, dorim să aflăm care este greutatea normală pentru o persoană de 1.78 m. Înlocuind, obținem, $\hat{y}_i = -141.35 + 125.84 \cdot 1.78 = 82.649$ kg. Din nou, previziunea obținută este doar o estimatie a valorii reale, deoarece, modelul s-a determinat pe baza unui eșantion și în plus, nu s-au luat în calcul toate aspectele, gen sex, rasă, etc., toate acestea intrând în variabila eroare, e . Prin urmare, este necesar să determinăm și un interval de încredere pentru previziune, de tip 95%, în care se află greutatea reală, corespunzătoare unei înălțimi de 1.78 m. $P(74.419 < y_i < 90.879) = 0,95$. În plus, funcția *polytool*, permite afișarea pe același grafic a dreptei de regresie, împreună cu punctele observate și împreună cu curbele care descriu intervalele de încredere pentru fiecare previziune.

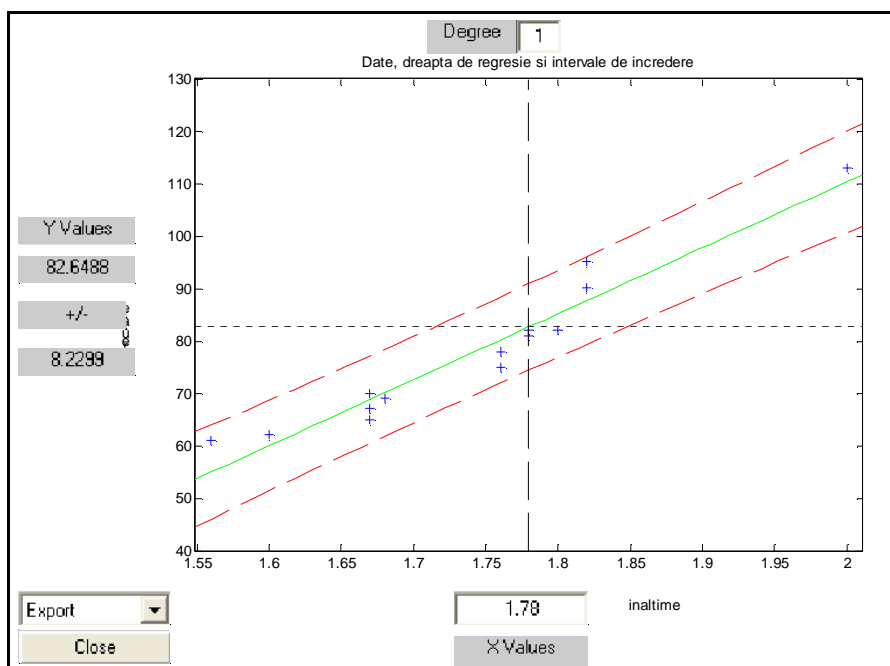


Figura 3.4. Interfața funcției *polytool*

Problema 2. Corelația dintre viteza de reducere circumferențială a ventriculului stâng și nivelul glucozei în sânge

Pentru un lot format din 24 de bolnavi de diabet de tip I, s-a măsurat, pe de o parte, *dinamica nivelului glucozei în sânge*-GS(mmol/l) și pe de altă parte, *viteza medie de reducere a circumferinței ventriculului stâng*-VcV(%/sec), în scopul predicției variabilei răspuns VcV, în funcție de nivelul GS, mai precis *în scopul predicției problemelor cardiovasculatorii în funcție de evoluția glicemiei*. Datele au fost următoarele ([9]):

Pacient nr.	GS(mmol/l)	VcV(%/sec)
1	15.3	1.76
2	10.8	1.34
3	8.1	1.27
4	19.5	1.47
5	7.2	1.27
6	5.3	1.49
7	9.3	1.31
8	11.1	1.09
9	7.5	1.18
10	12.2	1.22
11	6.7	1.25
12	5.2	1.19
13	19	1.95
14	15.1	1.28
15	6.7	1.52
16*	8.6	-
17	4.2	1.12
18	10.3	1.37
19	12.5	1.19
20	16.1	1.05
21	13.3	1.32
22	4.9	1.03
23	8.8	1.12
24	9.5	1.7

Tabelul 3.2. Date privind viteza de reducere circumferențială a ventriculului stâng și nivelul glucozei în sânge pentru 24 de bolnavi

Din tabel se poate observa că pacientul cu numărul 16 nu are înregistrată valoarea pentru VcV. După validarea unui model de regresie, se va putea estima această valoare lipsă, ca previziune pe baza modelului, pentru $X = GS = 8.6$. Ținând seama de dificultatea de a analiza direct problema contracției ventriculului stâng față de stabilirea nivelului glucozei în sânge, în sensul că nu putem găsi o formulă matematică directă de legătură, se impune studiul unui model regresiv. Rezolvarea acestei probleme prin studiul unui model liniar asupra datelor va constitui temă de laborator.

3.2.7. Teme de laborator în MATLAB

Tema 1

1. Încărcați în spațiul de lucru MATLAB, fișierul *GVdate.dat*, în care ați stocat datele Problemei 2 din subparagraful 3.2.6. Pe baza funcțiilor MATLAB, *corrcoef*, *refline*, *polyfit*, *polyval*, analizați cât de potrivit este un model liniar la aceste date, calculând coeficientul de corelație, reprezentați în același grafic, punctele și dreapta de regresie, calculați parametrii dreptei de regresie, calculați valoarea variabilei răspuns VcV , estimată (previzionată) pe baza modelului pentru o valoare a glicemiei $GS=8.6$.
2. Aplicând funcția *polytool*, determinați în plus intervalele de încredere pentru parametrii de regresie și pentru previziune și de asemenea, reziduurile. Reprezentați grafic dreapta de regresie împreună cu punctele și cu intervalele de încredere pentru previziune. Folosind rezultatele numerice obținute cu *polytool*, reprezentați reziduurile și evidențiați comportamentul acestora față de 0.

3.3. Alte modele de regresie simplă

Deși tehnicile specifice modelului liniar sunt simple și ușor de aplicat, în realitate, există și date a căror modelare reclamă altă formă decât cea liniară. O parte dintre aceste modele, cum sunt, de pildă, cele polinomiale, sau cele exponențiale, pot fi reduse prin anumite transformări la modelul liniar, motiv pentru care ele se mai numesc generic modele liniarizabile sau pur și simplu modele liniare, atunci când nu există pericol de confuzie cu modelul liniar propriu zis, altfel numindu-se modele neliniare. Celelalte modele, care nu pot fi reduse la forma liniară se numesc întotdeauna, modele neliniare.

3.3.1. Modelul polinomial

Modelul de regresie polinomială simplă este un model de regresie simplă în care funcția de regresie este un polinom în variabila predictivă, X . Forma modelului este următoarea:

$$Y = a_N X^N + a_{N-1} X^{N-1} + \dots + a_1 X + a_0 + e. \quad (3.28)$$

Ca și la modelul liniar se pune *problema determinării coeficienților modelului*, formulele teoretice fiind asemănătoare cu cele de la regresia liniară. În mod analog, *se pot determina intervale de încredere pentru coeficienți, previziuni pe baza modelului, interval de încredere pentru previziune și reziduurile*.

Coeficienții se pot afla fie direct aplicând criteriul celor mai mici pătrate, fie aplicând criteriul pe modelul liniarizat, după *transformarea modelului într-un model liniar (liniarizarea modelului)*. Pentru liniarizare, se folosesc următoarele substituții:

$$X^N = X_N, X^{N-1} = X_{N-1}, \dots, X = X_1, \quad (3.29)$$

modelul reducându-se la următorul model liniar multiplu:

$$Y = a_N X_N + a_{N-1} X_{N-1} + \dots + a_1 X_1 + a_0 + e. \quad (3.30)$$

Prin urmare, dintr-un model de regresie polinomială simplă, de gradul N , cu variabila efect Y și o singură variabilă predictivă X , s-a ajuns la un model de regresie liniară multiplă cu N variabile predictive, cu variabila efect Y și variabilele predictive, X_N, X_{N-1}, \dots, X_1 , definite pe baza lui X . Problema regresiei polinomiale constă în determinarea polinomului de regresie care descrie cel mai bine corelația dintre două variabile, X și Y , date prin n valori, $x_i, y_i, i = \overline{1, n}$. Un model polinomial este o alternativă simplă la modelul liniar, pentru cazurile când dreapta nu reușește să descrie destul de bine corelația dintre variabile.

Un exemplu de *motivație a modelului de regresie polinomială* este dat de Problema 2 descrisă în subparagraful 3.2.6 și propusă spre rezolvare în Tema 1 de laborator. Analizându-se un model liniar simplu pentru problema dată, se obține un coeficient de corelație liniară de $r = 0.416$, care indică o corelație liniară destul de slabă. Mai mult, analizând graficul punctelor vizavi de dreapta de regresie se poate observa că multe puncte sunt departe de dreapta.

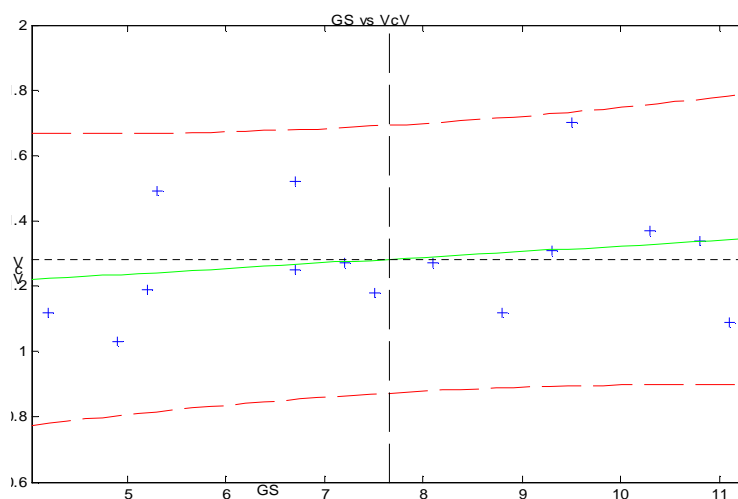
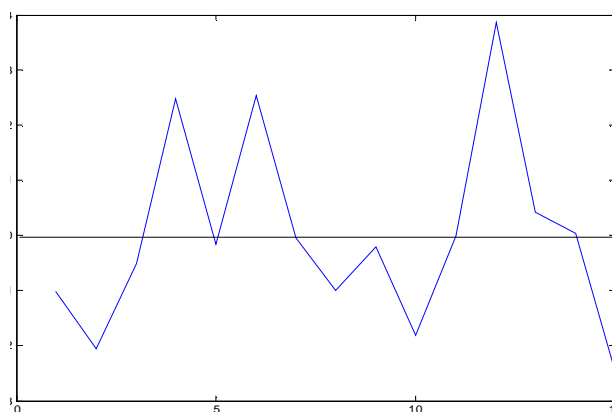


Figura 3.5. Corelograma, dreapta de regresie și intervale de încredere pentru predicție

În plus, reziduurile nu au un comportament aleatoriu și sunt destul de mari, după cum se poate observa în graficul următor.



Aceste aspecte ne îndreptăţesc demersul de a căuta un model mai potrivit pentru problema respectivă, prin urmare un model neliniar. Analiza unui model polinomial raportat la această problemă, se justifică prin faptul că dacă dreapta, care presupune un polinom de gradul 1, nu a fost suficient de flexibilă încât să se apropie de date, atunci, crescând gradul polinomului, creşte şi flexibilitatea curbei de regresie, prin urmare ne vom aştepta la o mai bună apropiere a acesteia de date.

Cu cât creşte gradul polinomului, cu atât creşte şansa ca acea curbă de regresie care este reprezentarea grafică a polinomului, să se apropie mai mult de punctele de coordonate $x_i, y_i, i = \overline{1, n}$. Totuşi există câteva restricţii care limitează alegerea unui grad mare. Având în vedere că un polinom de grad N presupune $N+1$ necunoscute, este nevoie de cel puţin $N+1$ condiţii pentru a le determina, prin urmare, pentru a modela datele printr-un polinom de grad N , este nevoie de cel puţin $N+1$ date, adică, $n \geq N+1 > N$. Cu alte cuvinte gradul polinomului trebuie să fie mai mic decât volumul datelor. Dacă alegem chiar gradul n , atunci toate punctele se vor afla pe curba de regresie polinomială, de grad n . Dacă alegem un grad mai mare decât n , polinomul nu poate fi complet determinat, ceea ce conduce la erori mari de aproximare.

Determinarea polinomului optim, presupune două aspecte :

- stabilirea gradului optim N al polinomului de regresie;
- stabilirea coeficienţilor polinomului optim.

Stabilirea gradului optim N al polinomului de regresie: Se reprezintă punctele de coordonate $x_i, y_i, i = \overline{1, n}$, într-un sistem ortogonal de axe şi se analizează forma liniei poligonale obţinute prin unirea cu segmente a punctelor. Dacă aceasta nu impune evident un anumit grad al polinomului (aproximativ o dreaptă-gradul 1, aproximativ o parabolă-gradul 2), se determină modele de regresie de diverse grade, se reprezintă grafic curbele aferente şi se păstrează curba faţă de care punctele se apropie cel mai mult. Pentru o selecţie optimă riguroasă a gradului polinomului, se pot compara şi statisticile de regresie aferente, pentru fiecare model.

Stabilirea coeficienţilor polinomului optim : Odată stabilit gradul polinomului, rămâne problema determinării coeficienţilor care ţin de forma analitică a funcţiei polinomiale, acest aspect rezolvându-se prin metoda celor mai mici pătrate, fie direct pe forma polinomială, fie pe forma liniarizată a modelului.

Analiza comparativă a statisticilor adiţionale pentru polinoame de diverse grade, este prezentată, pe un cadru mai general, pe orice model de regresie simplă, la finalul acestui paragraf.

3.3.2. Alte modele de regresie simplă neliniară

Toate modelele care se bazează pe alte funcţii de regresie decât cea liniară, se numesc modele de regresie neliniară. Modelul de regresie polinomială simplă este un astfel de exemplu, dar la fel de bine funcţia de regresie poate fi exponenţială, logaritmică, serie de puteri, funcţie trigonometrică, etc. De altfel, un model polinomial este de dorit atunci când scopul modelării este să copiem comportamentul datelor, ştiut fiind faptul că o curbă polinomială este cu atât mai flexibilă, cu cât, creşte gradul său. Însă un grad mare presupune şi un număr mare de coeficienţi necunoscuţi care cresc eroarea. În plus, un model polinomial este bun pentru previziuni pe valori care aparţin intervalului de date observate (intrapolare), dar din cauza puterilor mari ale

variabilei predictive X , dă erori mari la previziuni pe valori care nu aparțin intervalului de date observate (extrapolare). Vom considera în cele ce urmează, trei probleme care justifică abordarea altor modele decât modelul liniar și în unele cazuri, altul decât cel polinomial. De altfel, un model polinomial este abordat atunci când se caută un model simplu pentru date care nu par a fi liniare. Mai mult, diverse fenomene reale cer prin însăși natura problemei, o anumită formă a modelului, bazată pe alte funcții decât cele liniare, cum ar fi funcția exponențială, funcțiile trigonometrice, etc.

În cele ce urmează prezentăm câteva exemple de motivație a unui model de regresie neliniară. În acest sens, reluăm pentru început Problema 2 descrisă în cadrul subparagrafului 3.2.6, abordată inițial prin modelarea cu un model liniar simplu și mai apoi cu un model polinomial. Având în vedere anumite dezavantaje ale modelării polinomiale, se impune și analiza unui model neliniar precum și realizarea unui studiu comparativ pe mai multe modele, inclusiv cel polinomial. Spre exemplu, pentru modelul polinomial, din punct de vedere grafic cea mai bună ajustare pare să fie cea de gradul 5. Un polinom de grad mai mic nu va fi destul de flexibil pentru date. În schimb, pentru un polinom de grad mai mare nu vom avea suficientă informație în date pentru a estima cu destulă precizie coeficienții, fapt care se poate observa și în lățimea excesivă a intervalului de încredere pentru previziune, pe porțiunile unde nu avem date. Aceste aspecte se pot observa spre exemplu, pe un polinom de grad 9, pe graficul de mai jos.

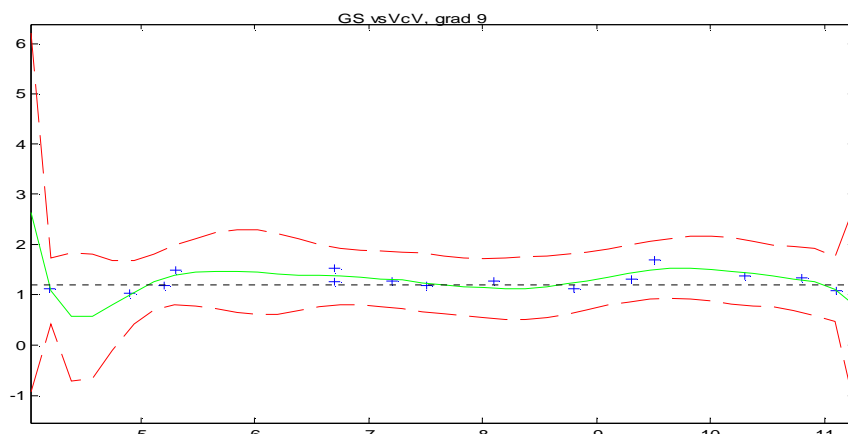


Figura 3.7. Corelograma, modelul de regresie polinomială de grad 9 și intervalele pentru predicție

Alte două probleme reale care reclamă prin specificitatea lor, utilizarea unui alt model decât cel liniar, respectiv decât cel polinomial sunt următoarele:

Problema 3. Corelația dintre rata de consumare a unui reactant într-o reacție chimică și concentrația reactantului.

Problema 4. Corelația dintre coeficientul de dilatare termică a cuprului și temperatura din cuptorul de prelucrare, în grade Kelvin.

În timp ce Problema 2, ridică necesitatea de a compara diverse modele, inclusiv cel polinomial, în vederea alegerii modelului optim, pentru Problema 3 și Problema 4, este din start, necesar un alt model decât cel polinomial. Astfel, practica arată că pentru Problema 3 este de dorit un model bazat pe funcții putere, în timp ce,

Problema 4 reclamă un model bazat pe funcții raționale, ambele modele simple neliniare.

Un *model de regresie simplă neliniară* este un model de forma,

$$Y = f(X, a_1, a_2, \dots, a_p) + e \quad (3.31)$$

unde funcția de regresie are altă formă decât cea a funcției liniare de gradul 1. Dacă, în urma unor transformări (logaritmare, substituții), modelul se poate reduce la un model de regresie liniară simplă sau multiplă, atunci modelul se numește *liniarizabil*.

Totuși într-o bună parte din literatura de specialitate, dacă funcția de regresie este neliniară în variabila explicativă X , dar liniară în coeficienții, a_1, a_2, \dots, a_p , atunci modelul este denumit generic, *model liniar*. Din această categorie fac parte și modelele polinomiale. Pentru astfel de modele, liniarizabile, se obțin aceleași rezultate, indiferent dacă se ajustează direct modelul prin metoda celor mai mici pătrate sau după liniarizarea modelului. Din acest punct de vedere, din clasa *modelelor neliniare*, fac parte acele modele, în care funcția de regresie nu este liniară în coeficienți. În general, pentru un astfel de model, în cazul când este liniarizabil, ajustarea liniară făcută după liniarizare este de preferat, deoarece ajustarea directă duce la un sistem de ecuații neliniare, care presupune de cele mai multe ori, doar aproximare și nu determinarea exactă a soluțiilor. Desigur, atunci când există un soft computațional adecvat aproximării numerice, se poate adopta și ajustarea neliniară, de cele mai mici pătrate. Obținerea unor soluții se face în acest caz, aplicând un algoritm iterativ de aproximare, cum ar fi metoda Gauss-Newton. Se pornește cu o presupunere inițială asupra coeficienților necunoscuți, aceștia se introduc în model, pe baza lor se calculează valorile prezise pentru Y . În următorul pas, se introduc în model, valorile previzionate ale lui Y , pe baza valorilor inițiale ale coeficienților, pe post de valori observate și se obține prin aplicarea ajustării liniare, de cele mai mici pătrate, un nou set de coeficienți. Pasul se reiterează până la îndeplinirea unui criteriu de suficiență (spre exemplu, diferența între coeficienții obținuți în doi pași consecutivi, să fie mai mică decât o valoare dată). Ca și concluzie, în ajustarea neliniară, aplicarea directă a criteriului celor mai mici pătrate, nu mai asigură aceleași calități pentru estimatori și de multe ori, nu se ajunge la forma analitică a acestora, ei fiind doar estimați.

Din clasa modelelor liniarizabile amintim modelul exponențial, hiperbolic, logaritmic și modelul funcției, putere.

Modelul exponențial: Modelul face parte din clasa modelelor neliniare (în parametri), funcția de regresie fiind dată prin,

$$\hat{y} = f(x) = ab^x. \quad (3.32)$$

Liniarizarea modelului exponențial se face prin logaritmare, obținându-se, $\log \hat{y} = \log a + x \log b$, de unde cu substituțiile, $\hat{z} = \log \hat{y}$, $A = \log a$, $B = \log b$, se obține modelul liniar în X și Z , $\hat{z} = A + Bx$. Valorile noii variabile Z , rezultă din ecuația de substituție, pe baza valorilor variabilei Y . Odată determinați A și B , din ecuațiile de substituție vor rezulta și valorile pentru a și b , modelul fiind astfel determinat. O variantă asemănătoare este dată de modelul $\hat{y} = ae^{bx}$ care se liniarizează prin logaritmare, $\ln \hat{y} = \ln a + bx$ și prin substituțiile, $\hat{z} = \ln \hat{y}$, $A = \ln a$. Modelele exponențiale sunt cerute pentru modelarea proceselor, în care rata de schimb a unei cantități este proporțională cu o valoare inițială legată de aceea

cantitate. Se utilizează frecvent în studiile epidemiologice, pentru a modela răspândirea unei boli infecțioase pentru care nu există tratament sau pentru a modela diverse creșteri în populații biologice, care nu țin cont de factorii externi.

Modelul hiperbolic (regresia reciprocă): Modelul face parte din clasa modelelor liniare (în parametri), funcția de regresie fiind dată prin,

$$\hat{y} = f(x) = \frac{b}{x} + a. \quad (3.33)$$

Liniarizarea modelului se poate face prin substituția, $z = \frac{1}{x}$, obținându-se modelul liniar, $\hat{y} = bz + a$, valorile noii variabile Z , fiind obținute din ecuația de substituție, pe baza valorilor lui X , iar parametrii fiind aceiași ca și în modelul inițial. Variante

asemănătoare sunt date de modelele, $\frac{1}{\hat{y}} = \frac{b}{x} + a$ și $\hat{y} = \frac{1}{\frac{b}{x} + a}$, care se rezolvă cu

substituțiile $u = \frac{1}{x}$, $\hat{v} = \frac{1}{\hat{y}}$, doar că aceste modele nu mai sunt liniare (în parametri).

Variante de modele combinate neliniare (în parametri), între modelul exponențial și cel hiperbolic sunt date de modelele, $\hat{y} = ae^{\frac{b}{x}}$ și $\frac{1}{\hat{y}} = be^{-x} + a$ sau $\hat{y} = \frac{1}{be^{-x} + a}$, care

se liniarizează prin logaritizarea, $\ln \hat{y} = \ln a + \frac{b}{x}$ și substituțiile,

$u = \frac{1}{x}$, $\hat{v} = \ln \hat{y}$, $A = \ln a$, respectiv, prin substituțiile $u = e^{-x}$, $\hat{v} = \frac{1}{\hat{y}}$.

Modelul logaritm: Modelul face parte din clasa modelelor liniare (în parametri), funcția de regresie fiind dată prin,

$$\hat{y} = f(x) = a + b \log x. \quad (3.34)$$

Liniarizarea modelului se poate face prin substituția, $\hat{z} = \log x$. Atât modelul hiperbolic, cât și modelul logaritm, se aplică atunci când datele inițiale nu se pretează la o regresie liniară, dar o anumită transformare a lor (inversul datelor, logaritmul datelor), poate fi potrivită pentru o astfel de regresie.

Modelul funcției putere: Modelul face parte din clasa modelelor neliniare (în parametri), funcția de regresie fiind dată prin,

$$\hat{y} = f(x) = ax^b. \quad (3.35)$$

Liniarizarea modelului se poate face prin logaritizarea, $\log \hat{y} = \log a + b \log x$ și prin substituțiile $u = \log x$, $\hat{v} = \log \hat{y}$. Un exemplu de aplicare a unui astfel de model, este și cel al unei reacții chimice, în care se știe că rata la care se consumă un reactant este în general, proporțională cu concentrația reactantului, ridicată la o anumită putere.

Din clasa modelelor ce nu pot fi liniarizate și pentru care se aplică tehnicile ajustării neliniare, bazate pe metode iterative, amintim modelul exponențial-putere, $\hat{y} = bx^a e^{cx}$, modelul putere-logaritm, $\hat{y} = bx^a + c \log x$, modelul dublu exponențial, $\hat{y} = ae^{bx} + ce^{dx}$, modelul bazat pe serii de puteri-exemplu cu doi termeni,

$$\hat{y} = a + bx^c, \text{ modelul bazat pe serii Fourier, } \hat{y} = a_0 + \sum_{i=1}^p a_i \cos(pwx) + b_i \sin(pwx).$$

Acest ultim tip de modele este folosit pentru a descrie un semnal periodic, de frecvență w . Un alt model neliniarizabil este și *modelul bazat pe funcții raționale*,

$$\hat{y} = \frac{\sum_{i=1}^{k+1} p_i x^{k+1-i}}{x^m + \sum_{i=1}^m q_i x^{m-i}}, \text{ unde gradul polinomului de la numărător este } k \text{ iar cel al}$$

polinomului de la numărător este m , acesta având și termen liber. Datorită acestui termen liber, cele două polinoame nu coincid niciodată, chiar dacă au același grad. Spre exemplu, un model rațional, cubic/gradul 4, va fi de forma

$$\hat{y} = \frac{p_1 x^3 + p_2 x^2 + p_3 x + p_4}{x^4 + q_1 x^3 + q_2 x^2 + q_3 x + q_4}. \text{ Acest tip de modele se folosește ca și modelele}$$

polinomiale, atunci când se caută un model simplu, pentru date care cer flexibilitate dar nu au o structură complicată. Probleme în aplicarea modelului apar atunci când numărătorul are valori apropiate de zero.

3.3.3. Analiza comparativă a modelelor de regresie

Ca și în cazul modelului liniar, studiul unui model de regresie oarecare presupune următoarele aspecte: determinarea coeficienților, a intervalelor de încredere pentru aceștia, determinarea și studiul repartiției reziduurilor, determinarea previziunii și a intervalelor previzionale, reprezentarea grafică a curbei de regresie vizavi de datele observate, împreună cu intervalele de încredere pentru previziune și analiza cantitativă a calității regresiei pe baza unor statistici adiționale. În cazul când se dorește alegerea modelului optim din mai multe modele, după analiza statisticilor de regresie, se va alege acel model care are cât mai multe statistici optime. Dintre statisticile de regresie mai des utilizate, amintim: suma de pătrate reziduale, coeficientul de corelație specifică, coeficientul ajustat, eroarea standard de regresie.

Suma de pătrate reziduale: Această statistică măsoară deviația totală a datelor observate față de datele prezise prin model, fiind de dorit să aibă o valoare cât mai mică, apropiată de 0. Formula de calcul este

$$S_{Rez} = \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (3.36)$$

Coeficientul de corelație specifică: Această statistică măsoară cât de mult este explicată variația variabilei răspuns prin model. Poate fi privită și ca și pătratul corelației între valorile observate și valorile prezise, ale variabilei răspuns. Pentru un model liniar, această statistică este egală cu pătratul coeficientului de corelație liniară, a lui Pearson. Formula de calcul este

$$R^2 = \frac{V_{\text{explicata prin regresie}}}{V_{\text{total}}} = \frac{S_{\text{explicata prin regresie}}}{S_{\text{total}}} = \quad (3.37)$$

$$\frac{S_{\text{total}} - S_{\text{rezidual}}}{S_{\text{total}}} = 1 - \frac{S_{\text{rezidual}}}{S_{\text{total}}}$$

adică,

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \in (0,1). \quad (3.38)$$

Este de dorit ca valoarea sa să fie cât mai mare, mai precis, cât mai aproape de 1. Spre exemplu, o valoare de 0,85 arată că un procent de 85% din variația datelor este explicată prin model. Dezavantajul acestei statistici este acela că va crește odată cu numărul de parametri necunoscuți, prin urmare nu prezintă siguranță în selecția unui model de un anumit grad dintr-o clasă de modele de același tip. Spre exemplu, ea va crește odată cu gradul polinomului de ajustare, dar asta nu înseamnă neapărat că un grad mai mare e mai bun pentru anumite date.

Coeficientul ajustat de corelație: Această statistică vine să înlăture neajunsul statisticii precedente, așa încât, să nu mai depindă de numărul de parametri necunoscuți, din model. Se obține prin ajustarea statisticii anterioare, cu gradele de libertate, $n = n - p$, unde p este numărul de parametri necunoscuți care intră în model și care trebuie estimați, iar n , reprezintă numărul de informații cu privire la model, adică, numărul datelor observate. Formula de calcul este

$$\bar{R}^2 = 1 - \frac{S_{rezidual} \cdot (n-1)}{S_{total} \cdot (n)} \quad (3.39)$$

Se interpretează la fel ca și R^2 , o valoare mare, apropiată de 1, indicând o bună ajustare a datelor, în timp ce o valoare apropiată de 0, indică o slabă ajustare. Este, de asemenea, cel mai indicat criteriu numeric, atunci când vrem să introducem un nou parametru în model (spre exemplu, în regresia polinomială, un grad mai mare presupune un nou parametru care rezează coeficientul puterii introduse prin acel grad) și dorim să vedem dacă în acest fel s-a îmbunătățit ajustarea. Dintre două regresii se va păstra aceea care are valoarea acestei statistici, mai mare.

Eroarea standard de regresie: Această statistică este de fapt radicalul erorii medii pătratice și este dată de formula,

$$E = \sqrt{\frac{S_{rezidual}}{n}}, \quad (3.40)$$

cu v , precizat mai sus. O valoare apropiată de 0, indică o bună ajustare a datelor, de către model.

În final, dintre toate modelele luate în calcul, într-o astfel de analiză, se va păstra modelul pentru care sunt satisfăcute cele mai multe din cele patru condiții și anume, să aibă cea mai mică sumă de pătrate reziduale și cea mai mică eroare standard, respectiv, cel mai mare, coeficient de corelație specifică și cel mai mare coeficient ajustat.

Distanțarea mare a punctelor față de curba de regresie, arată că modelul nu reușește să ajusteze suficient de bine datele (subajustează datele). Obținerea unor intervale largi de încredere pentru coeficienții unui model arată lipsa preciziei în estimarea coeficienților. Atunci când în zona datelor, intervalul este îngust, dar se

lărgeste simptomatic, în zona în care nu există date, înseamnă că modelul propus suprajustează datele sau altfel spus, datele de care dispunem nu sunt suficiente pentru a le ajusta cu un astfel de model. O astfel de situație se întâmplă atunci când încercăm să modelăm datele printr-un polinom sau un model rațional, de grad prea mare sau printr-o serie (Fourier, de puteri, exponențială) cu prea mulți termeni. Un interval larg de încredere (nesimultan, pentru o nouă observație) pentru previziune, arată că modelul nu prezintă siguranță în previziune. Existența unui anumit patern în alura reziduurilor proiectate față de axa $y=0$, arată faptul că erorile nu sunt întâmplătoare și este de dorit îmbunătățirea modelului sau studiul unui model mai bun. De asemenea, o prea mare depărtare a reziduurilor față de axa orizontală, $y=0$, arată erori prea mari și reclamă îmbunătățirea modelului sau schimbarea tipului de model. Este cazul când modelul subajustează datele.

3.3.4. Funcții și aplicații privind alte modele de regresie simplă în MATLAB

Vom prezenta pentru început, câteva funcții MATLAB, utile în implementarea tehnicilor de regresie, pentru modelul simplu polinomial, care include și cazul particular al modelului polinomial de gradul 1, prin urmare liniar, iar apoi vom prezenta o altă funcție cu extindere la oric tip de regresie simplă. Nu vom prezenta aici sintaxele cele mai generale ale funcțiilor ci doar cele care sunt mai potrivite în cadrul modelării matematice. Cele mai utilizate funcții în acest sens sunt funcțiile *polyfit*, *polyval*, *refcurve*, *polytool*, *cftool*. Funcțiile *polyfit*, *polyval*, *polytool* se regăsesc printre funcțiile MATLAB generale, în timp ce funcția *refcurve* este funcție specifică pachetului MATLAB “Statistics”, iar funcția *cftool* este specifică pachetului MATLAB “Curve Fitting” (pachet specializat pe ajustarea curbelor).

Sintaxa funcției polyfit: Funcția *polyfit* calculează parametrii polinomului de regresie de grad N , corespunzător datelor, utilizând metoda celor mai mici pătrate.

```
>>p = polyfit(x, y, N)
```

- *parametrii de intrare:* x , y -vectori-matrici coloană de aceeași dimensiune; N -reprezintă gradul polinomului de regresie;
- *parametrii de ieșire:* p -vector de dimensiune $N+1$, cu prima componentă reprezentând coeficientul puterii celei mai mari a lui x , iar ultima, termenul liber din $p_N \cdot x^N + p_{N-1} \cdot x^{N-1} + \dots + p_1 \cdot x + p_0$.

Sintaxa funcției refcurve: Pentru a putea analiza grafic corelația dintre puncte, respectiv variabile se poate utiliza funcția *plot*, cu sintaxa *plot(x,y)* care va desena linia poligonală dată de punctele de coordonate x , y sau *plot(x, y, '*')* care afișează graficul de puncte x , y . În urma analizei formei acestui grafic putem avansa ipoteze cu privire la forma funcției de regresie, în funcție de curba clasică prin care putem aproxima curba poligonală (curbă liniară, polinomială de un anumit grad, exponențială, putere, rațională, etc.) În cazul în care este vorba de o aproximare polinomială, pe același grafic se poate reprezenta și curba polinomială de regresie, utilizând funcția *refcurve*,

```
>>refcurve(p)
```

-*parametrii de intrare*: p -vectorul coeficienților polinomului de regresie, determinat prin metoda celor mai mici pătrate, corespunzător punctelor; vectorul începe cu coeficientul celei mai mari puteri a lui X și se termină cu termenul liber.

Comanda de mai sus, trebuie dată după comanda *plot*, înainte de închiderea figurii generate și reprezintă grafic în figura curentă, obținută cu funcția *plot(x,y,'*')*, curba polinomială de regresie, corespunzătoare punctelor, dată prin polinomul specificat prin coeficienții din vectorul p .

Sintaxa funcției polyval: Funcția *polyval* calculează previziunea pe baza polinomului de regresie, determinat de *polyfit*.

```
>>ypreviz = polyval(p,xnou)
```

-*parametrii de intrare*: p -vector de dimensiune $N+1$, care stochează parametrii polinomului de regresie, începând cu coeficientul lui x^N și terminând cu termenul liber; reprezintă rezultatul funcției *polyfit*; *xnou*-matrice care conține valorile noi ale variabilei explicative, valori pe care se dorește a fi calculată previziunea;

-*parametrii de ieșire*: *ypreviz*-matrice de aceeași dimensiune cu *xnou*, care conține valorile variabilei răspuns y (corespunzătoare valorilor noi ale lui x), previzionate din polinomul de regresie dat prin vectorul coeficienților de regresie, p .

Sintaxa funcției polytool: Funcția *polytool* acoperă toate facilitățile oferite de funcțiile de mai sus, prin integrarea graficelor și a răspunsurilor numerice într-o interfață interactivă, care permite controlul ajustării de cele mai mici pătrate printr-un polinom.

```
>>polytool(x, y, N, alfa)
```

-*parametrii de intrare*: x , y -vectori - matrici coloană - de aceeași dimensiune; N -reprezintă gradul polinomului de regresie; $alfa$ reprezintă pragul de semnificație admis pentru intervalele de încredere determinate pentru coeficienții modelului și pentru previziune;

-*parametrii de ieșire*: informațiile fundamentale legate de modelul polinomial de regresie simplă (coeficienți, reprezentare grafică, previziune, intervale de încredere, etc.), încorporate într-o interfață grafică interactivă; mai precis, funcția *polytool* creează un grafic interactiv al polinomului de regresie de grad N , pentru variabila răspuns y și variabila explicativă x , împreună cu punctele observate și cu intervalele de încredere de tip $(1-a)100\%$ pentru previziune (valoarea implicită este de 95%), precum și cu câteva informații numerice legate de modelul polinomial; interfața grafică permite controlul asupra alegerii valorii pentru care dorim previziunea și exportul unor parametri precum valoarea previzionată împreună cu intervalul de încredere, parametrii de regresie împreună cu intervalele de încredere și reziduurile.

În figura de mai jos, se poate observa interfața grafică pentru *polytool*, care afișează curba de regresie polinomială de gradul 3, precum și submeniul pentru exportul de date (coeficienții de regresie, intervale de încredere pentru coeficienți, valoarea lui y prezisă, corespunzătoare valorii lui x , setată pe orizontală, $x=5.5$, intervalul de încredere pentru previziune și reziduurile) către spațiul de lucru. În grafic se observă trei tipuri de curbe, curba empirică a datelor, reprezentată prin puncte, curba polinomului de regresie, de grad 3, de cele mai mici pătrate, corespunzător datelor, reprezentată prin linie continuă și cele două curbe semnificând intervalul de încredere pentru previziune, reprezentate prin curbe punctate. În partea

de sus, se observă afișajul din care se poate schimba gradul polinomului, schimbându-se implicit și graficul, iar în bara de instrumente de sus, putem schimba tipul de interval de încredere din “Bounds”, metoda de ajustare din “Methods”, inserare titlu și alte etichete din “Insert”, editarea imaginii din “Tools”, etc.

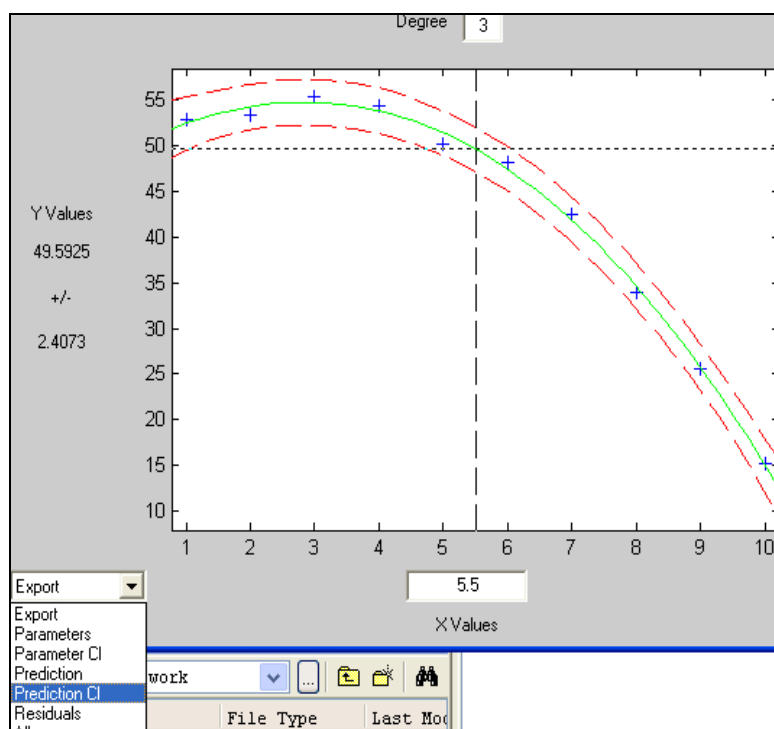


Figura 3.8. Modelul de regresie polinomială de grad 3. Exportul intervalelor pentru predicție

Sintaxa funcției cftool: Comanda *cftool*, permite deschiderea unei interfețe grafice proiectată pentru ajustarea datelor de tipul x , y , și implicit pentru determinarea unui model de regresie simplă între variabila răspuns și variabila predictivă. Aplicația dispune de o serie de modele clasice, liniare în parametrii sau neliniare, precum și de posibilitatea de a lucra cu modele scrise de utilizator. Dacă în spațiul de lucru există setul de date și acesta este alocat corespunzător în două variabile, x și y , este suficient ca de la tastatură să apelăm comanda:

```
>>cftool.
```

Prin această comandă se deschide interfața grafică prin intermediul căreia putem încărca datele din spațiul de lucru, în sesiunea *cftool*. Rezultatele obținute se pot salva din interfață, ca obiect MATLAB cu extensia “*cfit*” și pot fi apoi încărcate tot din interfață, la o nouă apelare a comenzii *cftool*, prin opțiunea “*load*”. În imaginea de mai jos, se poate observa interfața pentru *cftool*, în care optăm pentru submeniul “*Data*” și apoi din fereastra de dialog “*Data*”, încercăm datele din spațiul de lucru, pentru x și y . Prin click pe butoanele “*Create data set*” și “*View*” putem vizualiza în interfață, datele, atât grafic cât și numeric. S-au utilizat datele din Problema 1, descrisă în subparagraful 3.2.6.

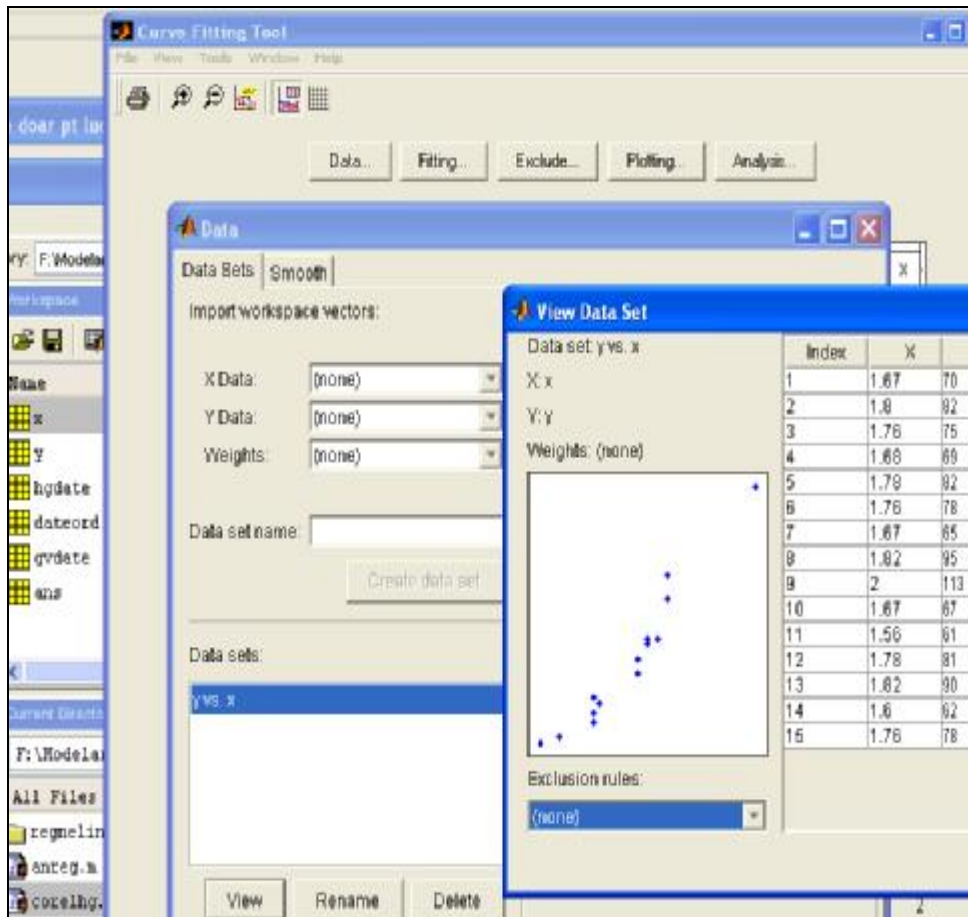


Figura 3.9. Interfața pentru cftool. Construcția setului de date

Prin utilizarea butonului “Fitting” se va deschide o altă fereastră de dialog, în care se realizează efectiv ajustarea, funcția *cftool* având încorporate diverse tipuri de modele de ajustare (polinomial, exponențial, putere, rațional, etc.) precum și posibilitatea de a defini un model de către utilizator. Din noua fereastră de dialog, selectăm tipul ajustării, spre exemplu, așa cum se poate observa în imagine, s-a selectat tipul polinomial de gradul 3 și prin utilizarea butonului „Apply”, primim două tipuri de informații, numerice și grafice. Informațiile numerice sunt afișate în fereastra de dialog “Fitting”, se referă la forma analitică a modelului, la coeficienții funcției de regresie, împreună cu intervalele de încredere, precum și la câteva statistici de regresie cum ar fi R^2 , toate aceste informații putând fi exportate spre spațiul de lucru, prin click pe butonul “Save to workspace”. Informația grafică sub formă de curba de ajustare împreună cu datele reprezentate prin puncte se afișează în fereastra principală de dialog pentru *cftool*; prin click pe butonul „View” din bara de instrumente, putem opta pentru a afișa suplimentar curbele pentru intervalele de încredere și reziduurile. Informația grafică poate fi salvată ca imagine, în vederea inserării într-un document word.

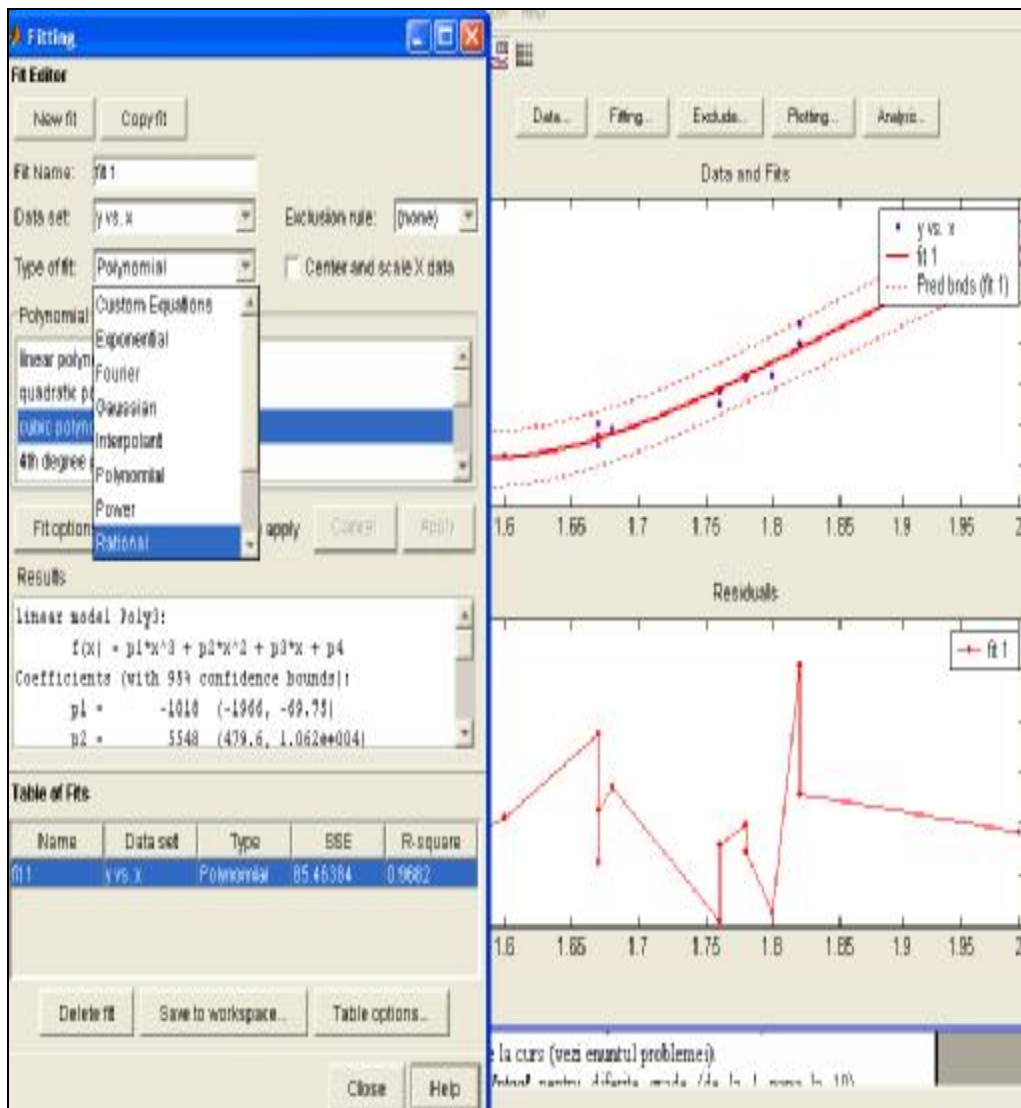


Figura 3.10. Interfața pentru cftool. Alegerea modelului

Prin click pe butonul „New fit”, se poate continua cu alte ajustări care se vor afișa pe același grafic. Astfel, se poate ajunge la cea mai bună modelare, comparând pe aceeași interfață, atât grafic cât și numeric prin statisticile care dau calitatea regresiei, diverse tipuri de modele pentru același set de date. Interfața dispune și de alte facilități precum evaluarea funcției de regresie în anumite puncte (“Analysis”), reprezentarea doar a anumitor curbe de regresie din cele încercate, pentru a înlesni comparația (Plotting), etc. Pe lângă biblioteca de modele încorporate în interfață, cftool dispune de două tipuri de subinterfețe pentru modele definite de utilizator (din “Type of fit”, alegem “Custom Equations”. Prima interfață, se poate vizualiza în imaginea de mai jos și presupune posibilitatea definirii ecuației unui model liniar în parametrii, utilizatorul introducând în interfață doar subfuncțiile din expresia liniară a modelului $a \cdot f_1(x) + b \cdot f_2(x) + \dots + c$.

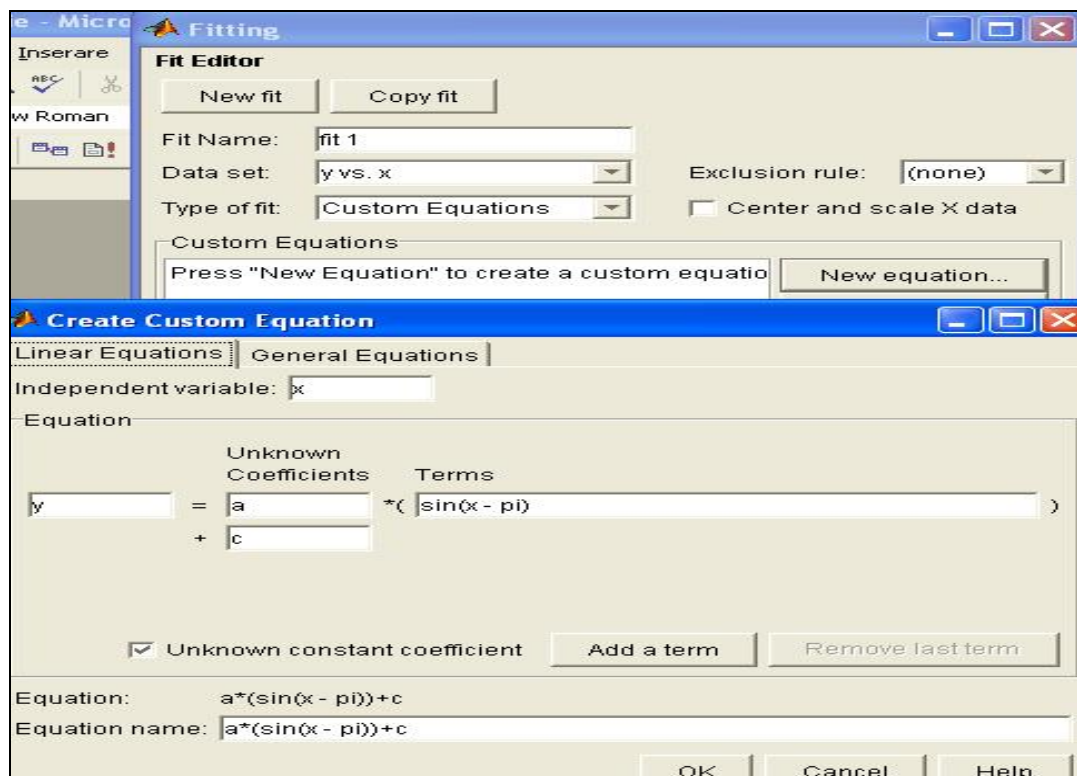


Figura 3.11. Interfața pentru cftool. Construcția unui model liniar în parametri definit de utilizator

În imaginea de mai jos, este prezentată interfața pentru celălalt caz al ecuației definite de utilizator și anume un model neliniar în parametri, în acest caz, utilizatorul specificând direct ecuația modelului și valorile inițiale pentru parametri necunoscuți, necesare în metoda iterativă Gauss-Newton (modelul nemaifiind liniar, soluția pentru coeficienți nu mai este exactă).

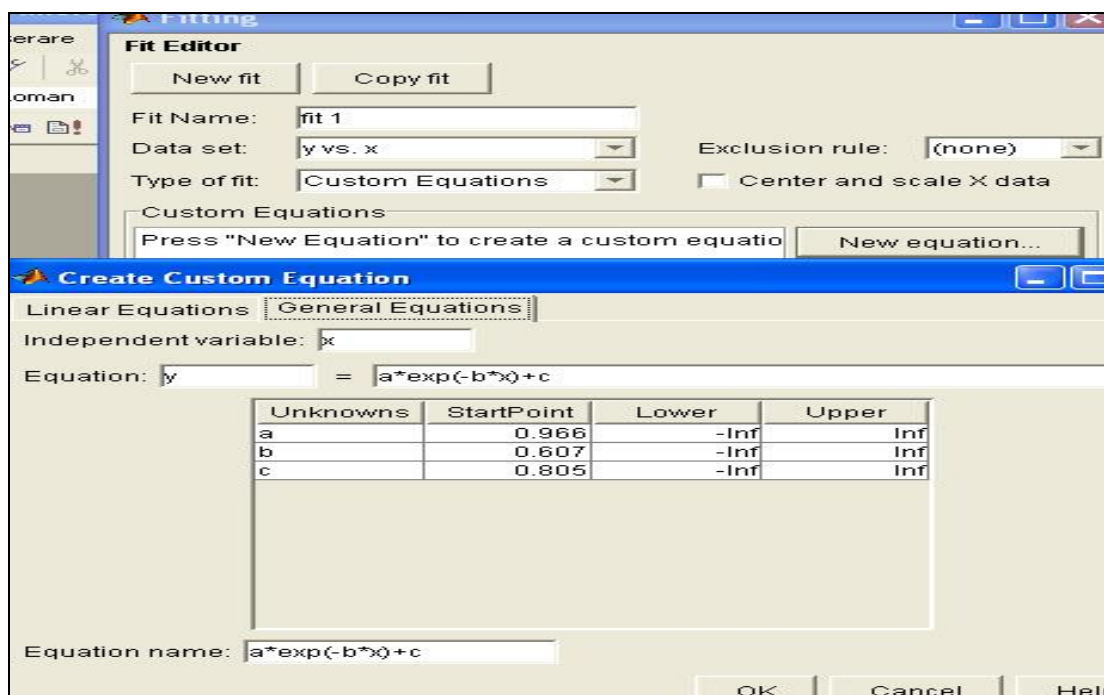


Figura 3.12. Interfața pentru cftool. Construcția unui model neliniar în parametri definit de utilizator

Spre exemplu, expresia MATLAB $a \cdot \sin(x - \pi) + b \cdot (x^2) + d \cdot (x - 3) + c$ este liniară în raport cu toți parametrii, în timp ce expresiile $a \cdot \exp(-b \cdot x) + c$, $a \cdot \sin(-b \cdot x) + c \cdot \cos(-d \cdot x)$, $a \cdot \sin(-b \cdot x) + c \cdot \sin(-d \cdot x)$ sunt neliniare, prima fiind liniară în a, c dar neliniară în b, deci pe ansamblu neliniară, a doua liniară în a, c, neliniară în b și d, iar a treia, la fel.

În cele ce urmează vom considera Problema 4 din subparagraful 3.3.2., referitoare la corelația dintre coeficientul de dilatare termică a cuprului și temperatura din cuptorul de prelucrare, în grade Kelvin. Ca și date observate vom folosi fișierul *hahn1.mat* din librăria de date a MATLAB-ului și vom vedea că cel mai potrivit model este modelul rațional 3-2 (cubic/cuadratic). Încărcând datele în interfața cftool, obținem următoarea reprezentare pentru cele 236 de date ([25]):

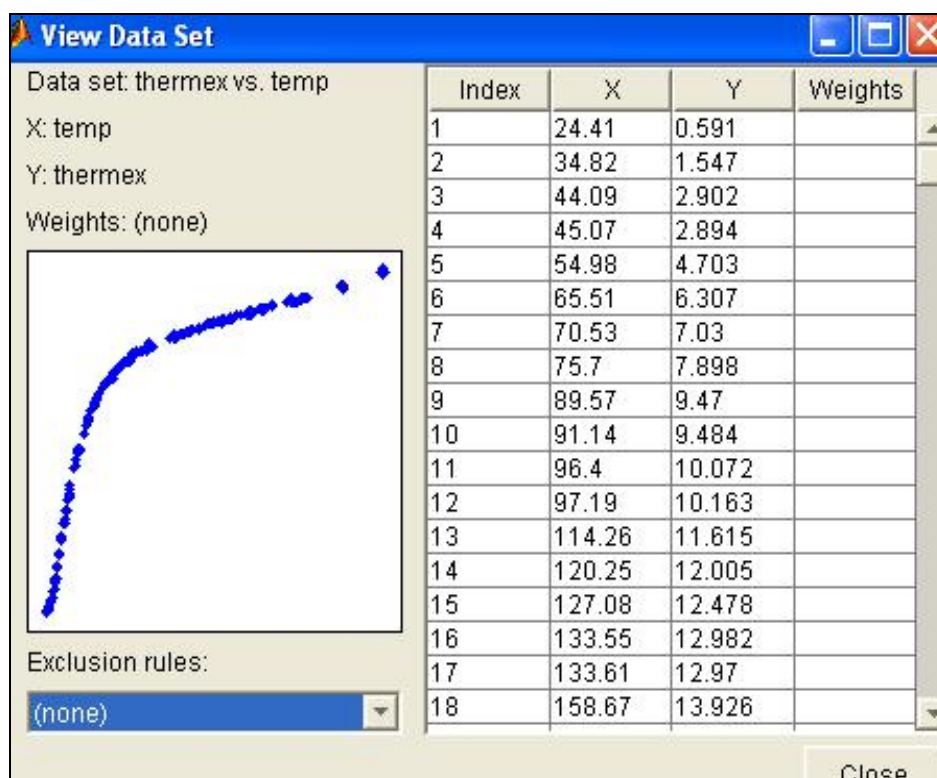


Figura 3.13. Interfața pentru cftool. Vizualizarea setului de date "hahn1"

Pentru început, se pot încerca (spre exemplu ca temă de laborator, diverse ajustări cum ar fi polinoame (spre exemplu, gradul 3, 4, 5), putere, exponențial, Fourier, rațional. În cadrul modelului polinomial se fac diverse ajustări, spre exemplu de tipul, 5-5, 1-5, 5-1, 2-2, 3-3, 2-3, 3-2 și se analizează direct sau prin comparație calitatea ajustării, ajungându-se la concluzia că modelul 3-2 este cel mai potrivit pentru date. De exemplu, pentru 3-3 (cubic pe cubic) obținem următoarea reprezentare grafică în care se poate observa că această curbă de regresie nu reușește să se apropie foarte bine chiar de toate datele și mai mult reziduurile manifestă o tendință, adică nu sunt întâmplătoare (sunt multe reziduuri consecutive peste 0 și de asemenea, multe consecutive peste 0).

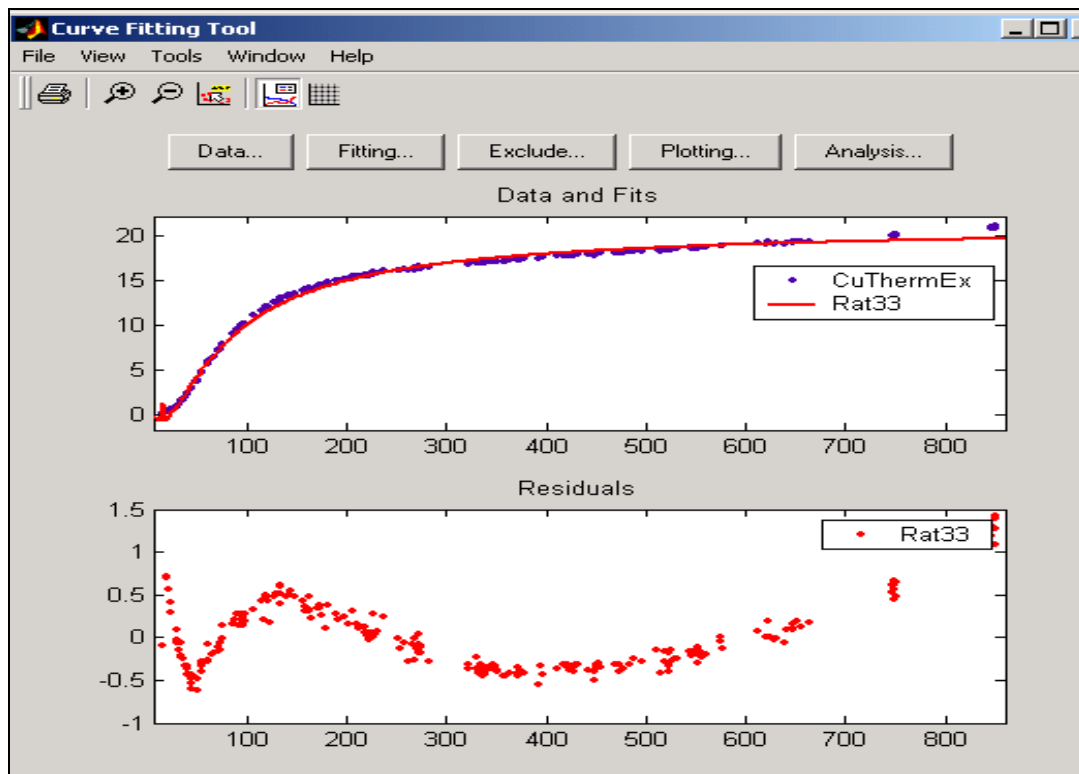


Figura 3.14. Modelul rațional de grad 3-3, corelograma, curba de regresie și reziduurile

De asemenea, rezultatele numerice prezentate mai jos, ne arată că rezultatele nu converg, prin urmare modelul este o alegere slabă pentru date.

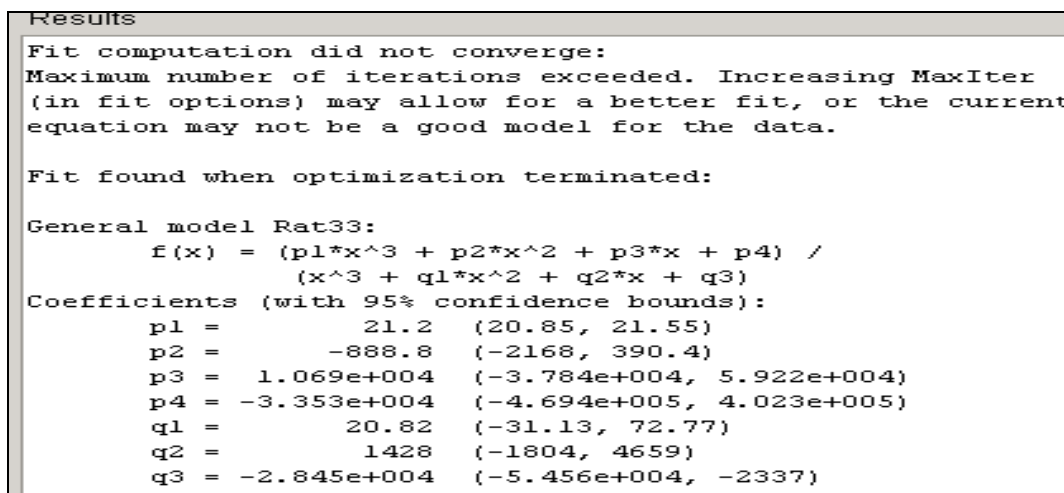


Figura 3.15. Afişajul rezultatelor numerice pentru coeficienții modelului și intervalele de încredere

În schimb, datele grafice prezentate mai jos pentru modelul 3-2, indică o bună ajustare a datelor (reziduuri fără patern, aleatoriu împrăștiate în jurul lui 0):

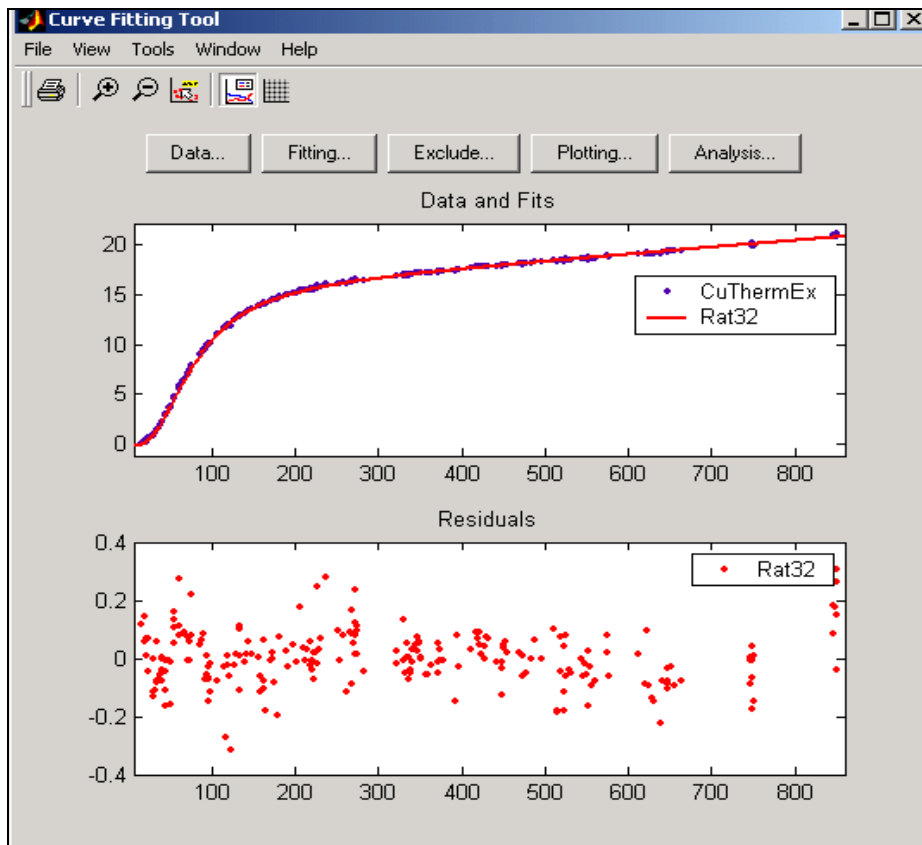


Figura 3.16. Modelul rațional 3-2

3.3.5. Teme de laborator în MATLAB

Tema 2

1. Încărcăți în spațiul de lucru MATLAB, fișierul GVdata.dat (un fișier care conține datele Problemei 2 din subparagraful 3.2.6). Aplicând funcția *polytool* pentru diferite grade (de la 1 până la 10), comparați din punct de vedere grafic, diferite modele polinomiale de regresie. Realizați și analizați de asemenea comparativ, graficele reziduurilor pentru modelele polinomiale de gradul 1, 5 și 9. În urma acestor analize, alegeți cel mai potrivit grad.
2. Pe baza funcțiilor MATLAB, *refcurve*, *polyfit*, *polyval*, reprezentați în același grafic, punctele și polinomul de regresie, pentru gradul ales la punctul 1, calculați parametrii polinomului de regresie, pentru același grad, calculați valoarea variabilei răspuns VcV, estimată (previzionată) pe baza modelului pentru o valoare a glicemiei, $GS = 8.6$.
3. În vederea alegerii unui model optim, aplicați funcția *cftool* pentru aceleași date, utilizând modele polinomiale de diverse grade, modele exponențiale, modele putere, modele Fourier, modele raționale. Comentați potrivirea fiecărui model la date, prin analiza grafică (a curbei de regresie, a intervalelor pentru previziune și a reziduurilor) și prin analiza cantitativă (a intervalelor de încredere pentru coeficienți și a statisticilor care dau calitatea regresiei).
4. Aplicând *cftool* creați două modele noi, neexistente în baza de modele, unul care să fie liniar în parametri, respectiv un model neliniar în parametri. Realizați ajustarea datelor și comentați potrivirea acestor modele la date.

3.4. Modele de regresie multiplă

3.4.1. Fundamentarea modelului de regresie multiplă

Atunci când variabila răspuns, Y , depinde de doi sau mai mulți factori predictivi (explicativi), X_1, X_2, \dots, X_p , *regresia* se numește *multiplă* și modelul corespunzător va fi

$$Y = f(X_1, X_2, \dots, X_p) + e. \quad (3.41)$$

Ca și în cazul modelului de regresie simplă, cel mai utilizat model de regresie multiplă este *modelul liniar multiplu*. Se numește *model de regresie, multiplu, liniar*, între variabila Y și variabilele X_1, X_2, \dots, X_p , modelul

$$Y = \sum_{k=1}^p a_k X_k + e. \quad (3.42)$$

Problema regresiei multiple constă în studiul comportării variabilei Y , în raport cu factorii X_1, X_2, \dots, X_p . Acest studiu revine la evaluarea parametrilor (coeficienților) de regresie, a_1, a_2, \dots, a_p . Estimarea coeficienților de regresie se face pe baza unui eșantion de volum n .

În cele ce urmează va fi tratat modelul liniar, însă multe dintre aspecte sunt valabile și pentru un model oarecare. Dacă scriem modelul pentru fiecare observație, $i = \overline{1, n}$, $y_i, x_{i1}, x_{i2}, \dots, x_{ip}$, avem $y_i = \sum_{k=1}^p a_k x_{ik} + e_i$. Modelul poate fi scris sub formă matriceală, folosind notațiile:

$$x = (x_1, x_2, \dots, x_p) = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}, n > p, \quad (3.43)$$

$$\begin{aligned} y' &= (y_1, y_2, \dots, y_n) \in \mathbb{R}^n, \\ a' &= (a_1, a_2, \dots, a_p) \in \mathbb{R}^p, \\ e' &= (e_1, e_2, \dots, e_n) \in \mathbb{R}^n. \end{aligned}$$

Astfel, modelul se poate scrie sub forma matriceală $y = xa + e$. În vederea estimării lui a , se ajustează modelul prin criteriul celor mai mici pătrate care constă în minimizarea expresiei:

$$e'e = \sum_{i=1}^n e_i^2. \quad (3.44)$$

Dacă $\text{rang}(x) = p$ (coloanele matricei x , adică vectorii, x_1, x_2, \dots, x_p , sunt vectori liniar independenți), atunci soluția ajustării prin criteriul celor mai mici pătrate este dată de formula

$$a = (x'x)^{-1} x'y. \quad (3.45)$$

Noțiuni utilizate în teoria modelelor de regresie: Se numește *valoare ajustată* sau *prezisă* a lui y (în modelul liniar), valoarea $\hat{y}' = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n) \in \mathbb{R}^n$, definită de $\hat{y} = xa$. Se numește *matrice de influență a modelului*, matricea H care transformă valoarea y , în valoarea ajustată, \hat{y} , adică, $\hat{y} = Hy$. Se numește *reziduu*, valoarea $e = y - \hat{y} = (I - H)y$. În cazul modelului liniar, matricea de influență are forma $H = x(x'x)^{-1} x'$. Un caz particular al modelului liniar, care face mai ușor trecerea la modelul liniar simplu (cu o singură variabilă explicativă), este modelul liniar cu termen constant, care se obține din modelul liniar înlocuind unul dintre predictorii cu vectorul constant $(1, 1, \dots, 1)$.

Ipotezele modelului: Pentru ca estimatorii coeficienților să aibă anumite proprietăți este necesar ca modelul să satisfacă anumite proprietăți referitoare la eroare. Astfel, avem condiția: $e \in N(q, S^2 I)$, adică $e \in N$, $E(e) = q$, $q = (0, 0, \dots, 0)' \in \mathbb{R}^n$ și $V(e) = E(e \cdot e') = S^2 I$. Dacă erorile îndeplinesc această condiție atunci modelul se numește modelul clasic Gauss Markov iar fără condiția de normalitate, modelul se numește doar Gauss Markov. Pentru eroarea care satisface aceste ipoteze se folosește și denumirea de zgomot alb. Ipoteză că eroarea este de medie nulă, arată faptul că erorile din model nu sunt erori sistematice ci întâmplătoare. Se mai spune că modelul este cu erori normale, independente și identic distribuite (i.i.d.). Condiția $V(e) = E(e \cdot e') = S^2 I$ se poate scrie și sub forma, $V(e_i) = S^2, \forall i = \overline{1, n}$, (model homoscedastic) împreună cu condiția $\text{cov}(e_i, e_j) = 0, \forall i \neq j, i, j = \overline{1, n}$ (model cu erori necorelate). O altă condiție, pe lângă cele privind eroarea, este ca predictorii (variabilele explicative) să nu fie corelați între ei. Pentru cazurile când modelul nu satisface una din condiții există reguli de corectare a modelului, dar estimatorii obținuți nu mai au aceleași calități. Nevalidarea acestor ipoteze duce la erori mari în model, de aceea, pentru a nu compromite din start modelul, se încearcă satisfacerea acestor condiții. E bine de specificat că, aceste condiții nu au legătură cu caracterul liniar al modelului, de aceea le putem întâlni și la alte modele, unde vor fi analizate în mod asemănător.

În vederea analizei calității modelului de regresie, se face inferența asupra coeficienților estimați și asupra coeficientului de corelație. Ca și în modelul simplu, intervalele de încredere pentru coeficienți și pentru previziune, trebuie să fie cât mai mici și statisticile de regresie, care se definesc similar cu cele de la cazul simplu, trebuie să fie optime. Se poate determina și un interval de încredere pentru eroare, interval care trebuie să-l conțină pe 0, pentru un model în care erorile sunt acceptabile. Atunci când există observații pentru care intervalele de încredere pentru reziduu corespunzător, nu-l conține pe 0, se preferă corectarea modelului prin eliminarea acelor observații din model. Metodele de comparație grafică întâlnite la modelul simplu, nu mai pot fi folosite aici, deoarece ar trebui să reprezentăm date

multidimensionale. Se pot utiliza însă grafice bidimensionale care să redea pe rând corelația dintre variabila răspuns și câte un predictor.

Teste pentru coeficienții modelului și pentru coeficientul de corelație: Ipotezele acestui test sunt ipoteza nulă, $H_0 : a_k = a_k^{(0)}$ și alternativa ei, $H_1 : a_k \neq a_k^{(0)}$, ceilalți coeficienți fiind în afara ipotezelor. Testul se fundamentează pe o statistică de tip Student, cu $n - p$ grade de libertate, n , numărul datelor, iar p , numărul coeficienților. Un caz particular al acestui test este cel al semnificației coeficientului a_k , test bazat pe ipotezele, $H_0 : a_k = 0$ și $H_1 : a_k \neq 0$. În cazul în care ipoteza nulă se acceptă, avem o justificare pentru a elimina predictorul X_k , din model, coeficientul lui, nefiind semnificativ diferit de 0.

Pentru a realiza inferența asupra coeficientului de corelație, se folosește o statistică F , de tip Fisher Snedecor, cu $p-1$ și $n-p$ grade de libertate, și se emite ipoteza nulă, $R = 0$, ipoteză ce infirmă regresia liniară. Pentru testarea ipotezei se ia în calcul un anumit prag de semnificație. Spre exemplu, în Matlab, se întoarce ca și informație de regresie, valoarea calculată a statisticii F și p -valoarea asociată acestei statistici, adică probabilitatea critică, $c = 1 - F_{p-1, n-p}(F_{calculata})$. Dacă $c \leq j$, j , prag de semnificație, se respinge ipoteza nulă.

În cazul regresiei multiple, apare *problema ierarhizării gradului de importanță a predictorilor*, în model păstrându-se doar acei factori care au o influență semnificativă a variabilei răspuns. Există două metode de selectare a factorilor care vor intra în model și anume *regresia pas cu pas, directă* (forward stepwise regression), respectiv *regresia pas cu pas inversă* (backward stepwise regression).

În *regresia directă*, se analizează corelația simplă dintre variabila răspuns și fiecare predictor în parte prin coeficientul de corelație, alegând cel mai puternic predictor ca fiind variabila care are cel mai mare coeficient de corelație și cea mai mică probabilitate critică la testul F , pentru coeficientul de corelație. Cel mai puternic predictor va fi prima variabilă explicativă care intră în model. Dintre variabilele rămase se alege aceea care are cea mai mare corelație (abstracție făcând de semn) cu reziduurile modelului de la pasul anterior. Se repetă procedeul până când efectul introducerii unei noi variabile în model devine nesemnificativ. În *regresia inversă*, se procedează analog dar se pornește cu toate variabilele în model și se elimină pe rând cele care nu au efect semnificativ. Pentru decizia finală se iau în calcul ambele metode.

Modele neliniare: În aplicațiile științifice, există de obicei suficientă teorie relevantă care să impună o anumită formă așa zisă mecanică a modelului, de obicei neliniară. Modelele neliniare sunt dificil de ajustat, necesitând presupuneri inițiale asupra coeficienților și metode iterative. Un astfel de caz va fi prezentat în partea de aplicații a paragrafului.

În cele ce urmează vom prezenta alte două probleme reale care necesită utilizarea unui model multiplu și care vor fi apelate pe parcursul paragrafelor cu aplicații și teme de laborator.

Problema 5. Corelația dintre anumiți indicatori biologici pentru bolnavii de fibroză chistică.

Se consideră un lot de 25 de bolnavi cu fibroză chistică pentru care s-au înregistrat datele privind presiunea statică maximă respiratorie P_{Emax} (cm H_2O), considerată ca variabilă răspuns, Y , vârsta (ani), sexul (codat, 0-M, 1-F), înălțimea H (cm), greutatea G (kg), masa corpului $BMP(G/H^2)$, volumul forțat expiratoriu într-o

secundă FEV, volumul residual RV, capacitatea funcțională reziduală FRC, capacitatea totală a plămânului TLC. Din regresia pas cu pas, directă s-a obținut modelul $PE \max = 124.83 + 1.64 \cdot G - 1.005 \cdot BMP$. Din regresia pas cu pas inversă, modelul rezultat este : $PE \max = 126.334 + 1.536 \cdot G - 1.465 \cdot BMP + 1.109 \cdot FEV$. În general, se păstrează în modelul final, predictorii care apar în ambele modele, cu coeficient semnificativ, adică, în acest caz, greutatea G și BMP. Datele sunt prezentate mai jos, ultima coloană reprezentând valorile variabilei răspuns ([9]):

	Vârsta	Sexul	H	G	BMP	FEV	RV	FRC	TLC	PEmax
1	7	0	109	13.1	68	32	258	183	137	95
2	7	1	112	12.9	65	19	449	245	134	85
3	8	0	124	14.1	64	22	441	268	147	100
4	8	1	125	16.2	67	41	234	146	124	85
5	8	0	127	21.5	93	52	202	131	104	95
6	9	0	130	17.5	68	44	308	155	118	80
7	11	1	139	30.7	89	28	305	179	119	65
8	12	1	150	28.4	69	18	369	198	103	110
9	12	0	146	25.1	67	24	312	194	128	70
10	13	1	155	31.5	68	23	413	225	136	95
11	13	0	156	39.9	89	39	206	142	95	110
12	14	1	153	42.1	90	26	253	191	121	90
13	14	0	160	45.6	93	45	174	139	108	100
14	15	1	158	51.2	93	45	158	124	90	80
15	16	1	160	35.9	66	31	302	133	101	134
16	17	1	153	34.8	70	29	204	118	120	134
17	17	0	174	44.7	70	49	187	104	103	165
18	17	1	176	60.1	92	29	188	129	130	120
19	17	0	171	42.6	69	38	172	130	103	130
20	19	1	156	37.2	72	21	216	119	81	85
21	19	0	174	54.6	86	37	184	118	101	85
22	20	0	178	64	86	34	225	148	135	160
23	23	0	180	73.8	97	57	171	108	98	165
24	23	0	175	51.1	71	33	224	131	113	95
25	23	0	179	71.5	95	52	225	127	101	195

Tabelul 3.3. Valorile indicatorilor medicali pentru 25 de bolnavi cu fibroză chistică

Problema 6. Corelația dintre rata reacției și concentrațiile reactanților în modelul Hougen-Watson.

Modelul Hougen-Watson de corelație a ratei reacției cu concentrațiile reactanților ([25]) este dat de relația, $Rata\ reactiei = \frac{a_1 x_2 - x_3 / a_5}{1 + a_2 x_1 + a_3 x_2 + a_4 x_3}$, unde

x_1, x_2, x_3 , reprezintă concentrațiile de hidrogen, n-pentan și izopentan. Valorile ratei de schimb a reacției sunt 8.55, 3.79, 4.82, 0.02, 2.75, 14.39, 2.54, 4.35, 13, 8.5, 0.05, 11.32, 3.13, iar valorile concentrației reactanților sunt prezentate, în ordinea descrisă, prin datele de mai jos.

<i>Hidrogen</i>	<i>n-Pentan</i>	<i>Izopentan</i>
285	80	10
470	300	120
470	80	120
470	80	10
100	190	10
100	80	65
470	190	65
100	300	54
100	300	120
100	80	120
285	300	10

Tabelul 3.4. Valorile concentrației reactanților pentru modelul Hougen-Watson

3.4.2. Funcții și aplicații privind modelul de regresie multiplă în MATLAB

Funcția *nlinfit*, analoaga funcției *polyfit*, se poate aplica și pentru cazul regresiei multiple, fiecare coloană din matricea X, conținând valorile pentru câte o variabilă explicativă, motiv pentru care va fi prezentată în acest paragraf, împreună cu funcțiile destinate regresiei multiple, *regress*, *rcoplot* și *stepwise*. Ca exemplu, vom considera pentru început datele stocate în matricea „moore” din MATLAB ([25]):

<i>X1</i>	<i>X2</i>	<i>X3</i>	<i>X4</i>	<i>X5</i>	<i>Y</i>
1125	232	7160	85.9	8905	1.5563
920	268	8804	86.5	7388	0.8976
835	271	8108	85.2	5348	0.7482
1000	237	6370	83.8	8056	0.716
1150	192	6441	82.1	6960	0.313
990	202	5154	79.2	5690	0.3617
840	184	5896	81.2	6932	0.1139
650	200	5336	80.6	5400	0.1139
640	180	5041	78.4	3177	-0.2218
583	165	5012	79.3	4461	-0.1549
570	151	4825	78.7	3901	0
570	171	4391	78	5002	0
510	243	4320	72.3	4665	-0.0969
555	147	3709	74.9	4642	-0.2218
460	286	3969	74.4	4840	-0.3979
275	198	3558	72.5	4479	-0.1549
510	196	4361	57.7	4200	-0.2218
165	210	3301	71.8	3410	-0.3979
244	327	2964	72.5	3360	-0.5229
79	334	2777	71.9	2599	-0.0458

Tabelul 3.5. Valorile celor 6 variabile din matricea „moore”

Cu următoarele comenzi vom construi parametrii de intrare pentru funcțiile destinate regresiei multiple:

```
>>load moore
>>X = [ones(size(moore,1),1) moore(:,1:5)]
>>y = moore(:,6)
```

Sintaxa funcției regress: Funcția *regress* face parte din Statistics Toolbox și permite determinarea parametrilor unui model multiplu de regresie liniară, împreună cu intervalele de încredere. De asemenea, se mai poate returna ca rezultat, șirul reziduurilor împreună cu intervalele de încredere și cu câteva statistici care dau calitatea regresiei. Sintaxa cea mai complexă este

```
>>[b,bint,r,rint,stats] = regress(y,X,alpha).
```

-În parametrii de intrare *y*, respectiv *X*, se stochează datele înregistrate pentru variabila răspuns, respectiv pentru variabilele factori, coloanele matricii *X* reprezentând câte un factor. În parametrul de intrare *alpha* se stochează pragul de semnificație pentru intervalele de încredere determinate în cadrul regresiei.

-În parametrii de ieșire *b*, respectiv *bint* se stochează vectorul parametrilor modelului, ordinea fiind corespunzătoare ordinii din *X*, adică prima valoare din *b* este coeficientul primului factor X_1 , iar *bint* stochează capetele intervalelor de încredere corespunzătoare parametrilor de regresie. În *r*, respectiv *rint* se stochează vectorul reziduurilor (de dimensiune egală cu numărul de date), respectiv intervalele de încredere pentru reziduuri, iar în *stats* se returnează în această ordine, valoarea lui R^2 , valoarea statisticii F asociate ipotezei că toți coeficienții de regresie sunt 0, respectiv valoarea de semnificație p , asociată acestui test. Pentru aplicarea corectă este necesară redefinirea matricii *X* ca fiind $X=[\text{ones}(\text{size}(y) X)]$, astfel încât regresia să fie cu termen liber și în acest caz primul parametru estimat este termenul liber. Spre exemplu, pentru datele din matricea "moore", $\text{stats} = 0.8107 \quad 11.9886 \quad 0.0001$, adică 0.8107 indică faptul că modelul acoperă datele în proporție de 80%, iar valoarea $F=12$ cu nivelul 0.0001 indică faptul că este destul de improbabil că toți coeficienții sunt zero.

Sintaxa funcției rcoplot: Funcția *rcoplot* face parte din Statistics Toolbox și are sintaxa :

```
>>rcoplot(r, rint).
```

Prin această comandă se reprezintă grafic reziduurile calculate cu *regress*, împreună cu intervalele de încredere, cele mai acceptabile reziduuri fiind acelea la care intervalul de încredere îl conține pe 0. Spre exemplu, la datele stocate în matricea *moore*, graficul va fi următorul:

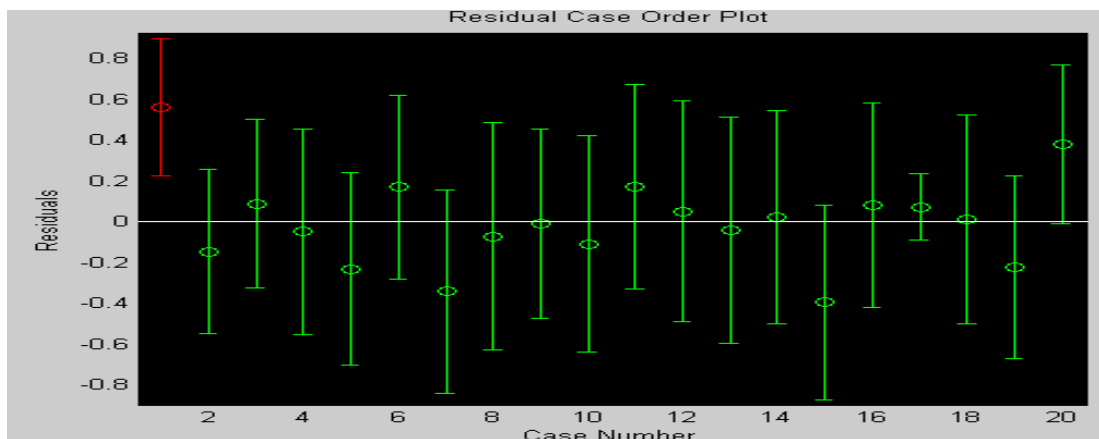


Figura 3.17. Vizualizarea datelor de tip „outlier” cu ajutorul intervalelor de încredere pentru reziduri

În fereastra graficului elementul „outlier” al cărui interval de încredere pentru reziduu nu-l conține pe 0 este desenat cu roșu, celelalte segmente fiind desenate cu verde.

Sintaxa funcției stepwise: Funcția *stepwise* face parte din Statistics Toolbox și permite utilizarea unei interfețe interactive pentru determinarea factorilor cei mai semnificativi pentru considerarea lor în modelul liniar. Se poate aplica și pentru regresia forward, plecând de la un factor și adăugând pe cei mai importanți, în funcție de anumite statistici de regresie obținute, fie pentru regresia backward, plecând de la toți factorii și eliminând pe rând câte unul care este mai puțin semnificativ. Sintaxa este

```
>>stepwise (X, Y) .
```

Pentru exemplificare vom considera o altă bază de date „*hald*” în care valorile variabilelor factori sunt stocate în matricea „*ingredients*” ([25]),

<i>X1</i>	<i>X2</i>	<i>X3</i>	<i>X4</i>
7	26	6	60
1	29	15	52
11	56	8	20
11	31	8	47
7	52	6	33
11	55	9	22
3	71	17	6
1	31	22	44
2	54	18	22
21	47	4	26
1	40	23	34
11	66	9	12
10	68	8	12

Tabelul 3.6. Valorile variabilelor „*ingredients*” din matricea „*hald*”

iar valorile pentru variabila răspuns sunt stocate în vectorul „*heat*”:

78.5, 74.3, 104.3, 87.6, 95.9, 109.2, 102.7, 72.5, 93.1, 115.9, 83.8, 113.3, 109.4.

Comanda

```
>>stepwise(ingredients, heat)
```

va duce la deschiderea unei interfețe interactive cu trei ferestre:

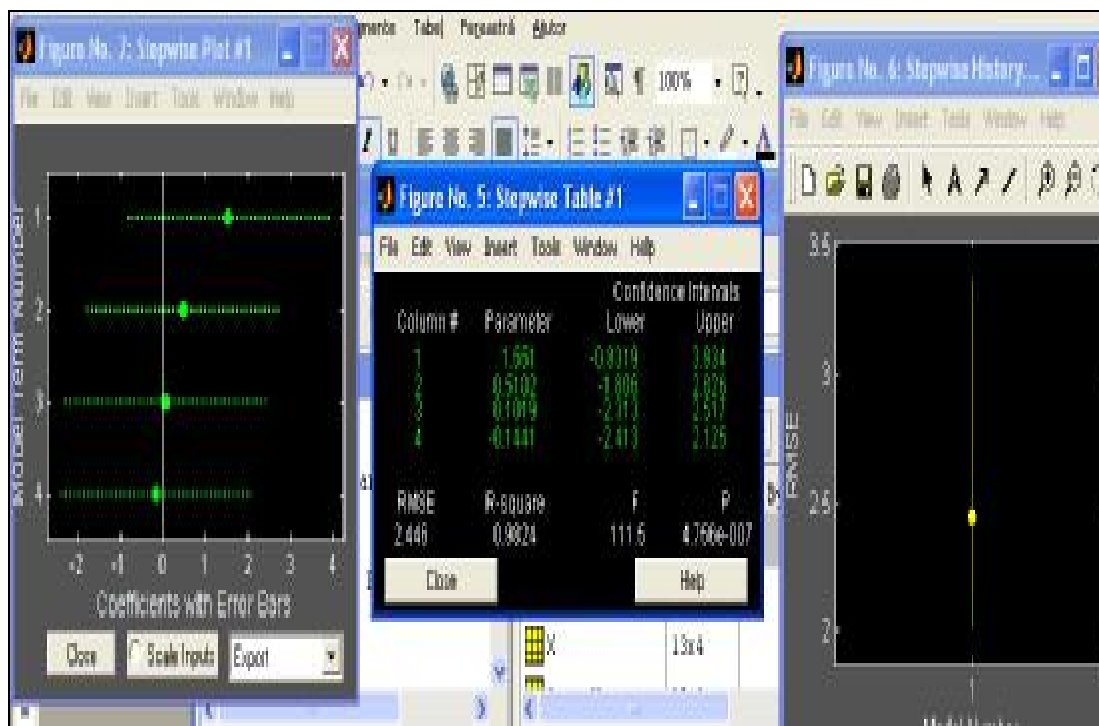


Figura 3.18. Interfața funcției stepwise

În fereastra din stânga sunt afișate cu culoare verde, coeficienții împreună cu intervalele de încredere. Un factor este nesemnificativ pentru model, deci poate fi exclus, atunci când este nesemnificativ diferit de zero. În fereastra din stânga, un astfel de coeficient va avea intervalul de încredere așa încât să-l conțină pe 0, iar linia intervalului este reprezentată punctat. De asemenea, un interval de încredere scurt va arăta precizie în estimarea coeficienților. În fereastra din mijloc avem informațiile numerice privind coeficienții, intervalele de încredere, precum și câteva statistici de regresie. Spre exemplu, modelul este cu atât mai bun cu cât are RMSE (eroarea) mai mică și R-square mai mare. În ultima fereastră sunt reprezentate erorile RMSE cu intervalele de încredere corespunzătoare, consecutiv pentru fiecare model încercat. Pentru a încerca și alte modele, adică a elimina sau adăuga un termen, se face click pe termenul corespunzător în fereastra din stânga, culoarea acestuia schimbându-se din verde în roșu, ceea ce înseamnă că factorul respectiv nu mai este în model, iar pentru ceilalți factori se schimbă informațiile în funcție de determinarea acestui nou model. Dacă vrem să reintroducem factorul în model, facem din nou click pe acel factor. Atunci când avem regresie backward, excluzând pe rând câte un factor va fi de preferat să-l excludem pe cel care are valoarea mai apropiată de zero. Când comparăm modelele îl vom păstra pe cel care are cel mai mare R-square, cea mai mică RMSE (pe ultima fereastră se pot compara grafic), respectiv care are cât mai puțini coeficienți reprezentați punctat.

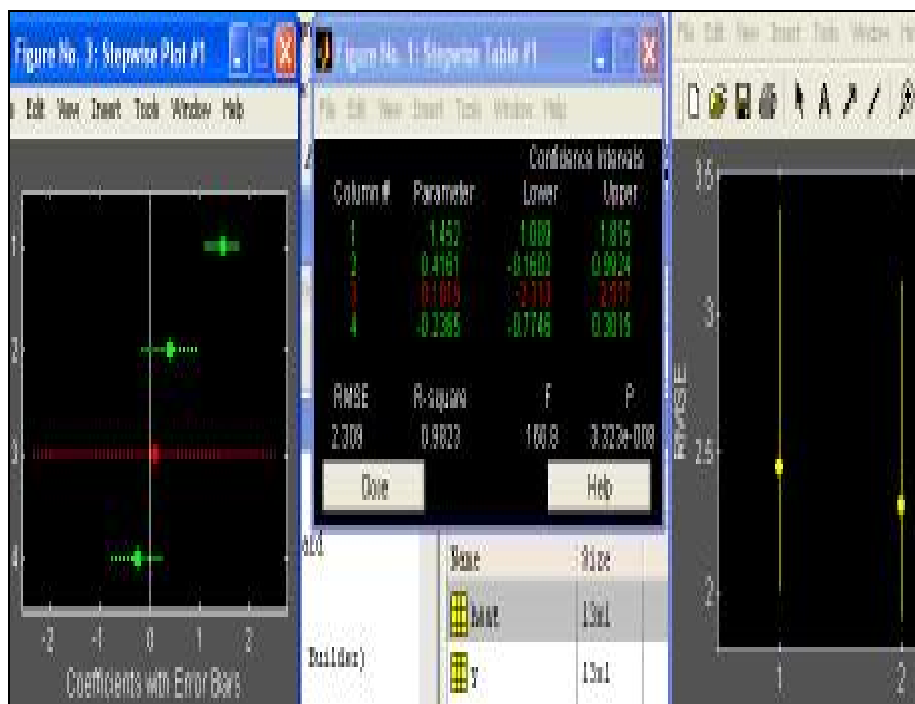


Figura 3.19. Interfața funcției stepwise. Model cu trei termeni

În imaginea de mai sus, este reprezentat modelul cu trei termeni, obținut prin excluderea celui de-al treilea termen din cei patru, care era cel mai apropiat de 0. Se vede că primul coeficient devine astfel semnificativ diferit de zero, RMSE scade, iar R-square scade nesemnificativ (artificial, datorită scăderii numărului de coeficienți). Dacă vrem să avem în model doar coeficienți semnificativi, atunci mai trebuie să continuăm cu excluderea factorilor neimportanți. Vom încerca două variante, una în care excludem factorul al doilea din primii patru și alta în care excludem ultimul factor. Pentru factorul al doilea exclus, obținem,



Figura 3.20. Interfața funcției stepwise. Model cu primul și ultimul termen

adică toți factorii rămași sunt semnificativi, iar RMSE crește, respectiv R-square scade nesemnificativ, RMSE=2,734, R-square=0,9725. În cazul în care excludem ultimul factor și îl reintroducem în model pe cel de-al doilea, obținem:

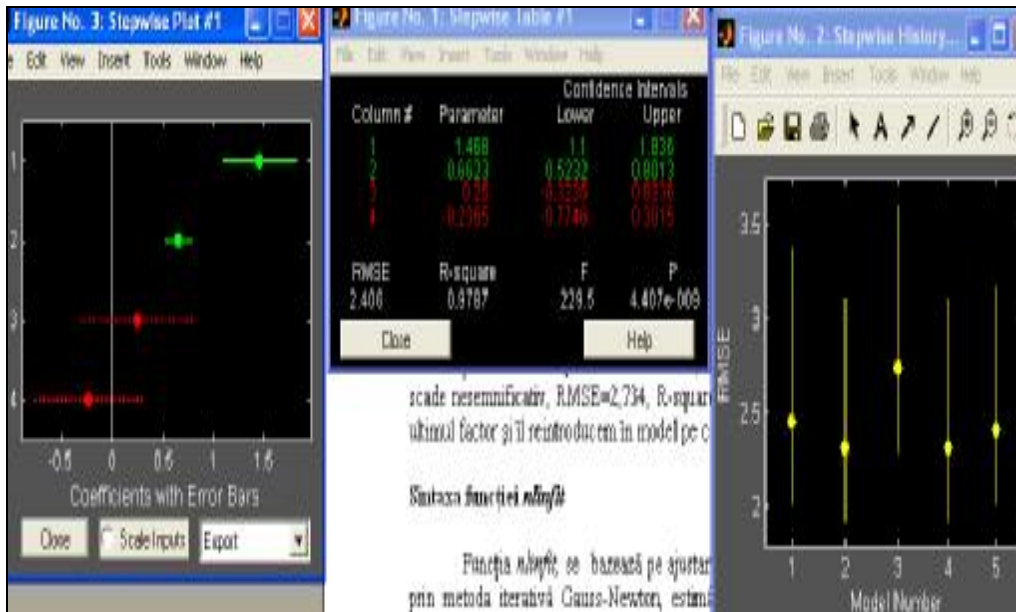


Figura 3.21. Interfața funcției stepwise. Model cu primii doi termeni

adică, avem de asemenea ambii factori rămași semnificativi, iar $RMSE=2,406$, $R\text{-square}=0,9787$. Deci, cum $RMSE$ este mai mic și $R\text{-square}$ mai mare, preferăm această ultimă variantă. Pe graficul din ultima fereastră prima bară reprezintă versiunea cu patru termeni, a doua versiunea cu trei, a treia versiunea cu doi factori, prin excluderea celui de-al doilea, a patra din nou versiunea cu trei factori după reintroducerea în model a factorului al doilea, iar ultima versiunea cu doi factori după excluderea celui de-al patrulea. Dacă am încerca să excludem mai mult, eliminând și factorul al doilea, rămânând cu un model cu un singur factor, primul, care pare cel mai semnificativ, se poate observa că $R\text{-square}$ scade exagerat de mult, în timp ce $RMSE$ crește prea tare, așa cum se poate observa și în figura următoare, prin urmare, păstrăm în model primii doi factori.



Figura 3.22. Interfața funcției stepwise. Model cu un singur termen

Sintaxa funcției nlinfit: Funcția se bazează pe ajustarea neliniară prin metoda iterativă Gauss-Newton, estimând coeficienții unui model neliniar:

```
>> betahat = nlinfit(X,y,FUN,beta)
```

- *parametrii de intrare:* X , respectiv y , stochează valorile variabilelor factor, respectiv ale variabilei răspuns, FUN conține funcția prin care se dorește ajustarea și care trebuie definită cu procedura FUNCTION din MATLAB, de forma $yhat = myfun(beta,X)$ iar $beta$ conține vectorul valorilor de inițializare pentru coeficienții ce urmează a fi determinați.

- *parametrii de ieșire:* $betahat$ este vectorul estimațiilor coeficienților modelului descris de funcția FUN .

Pentru exemplificare, vom considera modelul Hougen-Watson descris în Problema 6 din subparagraful 3.4.1, de ecuația $rata\ reactiei = \frac{a_1 x_2 - x_3 / a_5}{1 + a_2 x_1 + a_3 x_2 + a_4 x_3}$.

și datele din fișierul MATLAB “*reaction*” cu variabilele factori “*reactants*” și variabila răspuns „*rate*”, care conțin date simulate pentru acest exemplu ([25]). Se poate observa că factorii nu intră liniar în model. Funcția Hougen dată de ecuația de mai sus se definește într-un fișier cu extensia *.m*, conținând următoarele linii de cod:

```
>>function yhat = hougen(beta,x)
>>b1 = beta(1);
>>b2 = beta(2);
>>b3 = beta(3);
>>b4 = beta(4);
>>b5 = beta(5);
>>x1 = x(:,1);
>>x2 = x(:,2);
>>x3 = x(:,3);
>>yhat = (b1*x2 - x3/b5)./(1+b2*x1+b3*x2+b4*x3);
```

Pentru a ajusta datele cu această funcție se precizează un vector pentru $beta$, de exemplu, $beta = [1; 2; 5; 1; 3]$ și se dă comanda

```
>>betahat = nlinfit(reactants,rate,'hougen',beta)
```

Se obține rezultatul $betahat = (1.2526 \ 0.0628 \ 0.0400 \ 0.1124 \ 1.1914)$.

3.4.3. Teme de laborator în MATLAB

Tema 3

1. Incărcați fișierul care conține datele Problemei 5, descrise în subparagraful 3.4.1. Alegeți cel mai bun model liniar pentru aceste date, utilizând funcția *stepwise*.
2. Determinați caracteristicile modelului (valoarea coeficienților, reziduurile, intervale de încredere pentru aceștia, etc.) cu funcția *regress*. Reprezentați grafic reziduurile.
3. Propuneți pentru aceste date, un model neliniar în parametrii și lucrați pe acel model cu funcția *nlinfit*.

Capitolul 4

Procese stochastice staționare aplicate în procesarea semnalelor

Modelele probabiliste formează fundamentul teoriei informației. Informația ca atare este cuantificată în termeni de logaritimi și probabilități. Modelele probabiliste sunt folosite pentru a caracteriza și prezice apariția unui eveniment aleator în arii diverse precum predicția numărului de apeluri telefonice de pe o magistrală într-o anumită perioadă a zilei, modelarea traficului auto, modelarea datelor financiare, predicția efectelor medicamentelor în funcție de datele experimentale etc. În procesarea semnalelor modelele probabiliste sunt folosite pentru a descrie variațiile semnalelor aleatoare în aplicații precum recunoașterea formelor, codarea semnalelor și estimarea lor. În acest context procesele stochastice sunt clase de semnale a căror fluctuație în timp este parțial sau complet aleatoare, ca de exemplu vorbitul, muzica, imaginile, zgomotele și semnalele video. O descriere completă a acestor semnale se realizează cu ajutorul unui model probabilistic, dar ele pot fi caracterizate și cu ajutorul unor instrumente statistice simple precum medie, corelație sau densitate spectrală.

4.1. Semnale aleatoare și procese stochastice

Semnalele, în funcție de caracteristicile lor fundamentale pot fi clasificate în două categorii: semnale *deterministe* și semnale *aleatoare*. În limbajul teoriei procesării semnalelor funcțiile aleatoare de timp se mai numesc *semnale stochastice*. În fiecare clasă în parte, un semnal poate fi continuu sau discret (ca funcție de timp) și poate avea de asemenea amplitudine continuu sau discret valuată.

Un semnal deterministic poate fi definit ca unul de parcurge o traiectorie predefinită în timp și spațiu. Fluctuațiile exacte ale unui semnal deterministic pot fi complet descrise în termeni de funcții de timp, iar valoarea exactă a semnalului la un anumit moment este predictibilă în funcție de descrierea funcțională a istoriei trecute a semnalului. De exemplu o undă sinusoidală $x(t)$ poate fi modelată și prezisă cu acuratețe fie printr-un model predictiv liniar de ordinul doi fie prin mult mai familiara formă ecuațională $x(t) = A \sin(2\pi ft + f)$.

Semnalele aleatoare au fluctuații impredictibile; ca atare nu se poate formula o ecuație care să-i prezică valoarea *exactă* în funcție de istoria trecută a semnalului. Majoritatea semnalelor, ca vorbitul sau zgomotele sunt, cel puțin în parte, semnale aleatoare. Conceptul de aleator este strâns legat de conceptele de informație și zgomot. Într-adevăr, o mare parte din procesarea semnalelor aleatoare are de a face cu extragerea informației din observarea zgomotelor. Dacă un semnal ar fi să aibă capacitatea de a cuprinde informație, atunci el trebuie să aibă un anumit grad de aleator: un semnal predictibil nu mai conține informație. De aceea partea aleatoare a unui semnal este conținutul informațional al semnalului sau zgomot sau ambele. Deși un semnal aleator nu este complet predictibil, acesta prezintă un set de caracteristici statistice bine definite precum minimul, maximul, media, mediana, varianța și densitatea spectrală.

4.1.1. Tipuri de procese stochastice

În general prin proces stochastic înțelegem un șir infinit de variabile aleatoare $(X(0), X(1), \mathbf{K})$ cu câte o variabilă aleatoare pentru fiecare moment de timp și valorile (sau realizările) acestora se reprezintă ca un șir infinit de numere $(x[0], x[1], \mathbf{K})$. Un proces aleator se va nota succint $X[n]$ având realizările $x[n]$, n parcurgând mulțimea numerelor naturale, adică $n=0,1,\mathbf{K}$. Folosim parantezele drepte pentru a indica faptul că lucrăm cu *parametru de timp discret*. În consecință, folosim și terminologia *proces aleator cu parametru de timp discret* vs. *proces aleator cu parametru de timp continuu*.

De multe ori însă avem nevoie să privim un proces ca având trecut infinit, deci cu parametrul de timp parcurgând mulțime întregilor și nu „doar” mulțimea numerelor naturale. Un astfel de proces îl vom numi *proces infinit* spre diferență de cel definit anterior, care se mai numește *proces semi-infinit*.

După valorile proceselor, discrete sau continue, distingem între *proces cu parametru de timp discret și valori discrete* (DTDV), respectiv celelate posibile combinații, anume *proces cu parametru de timp discret și valori continue* (DTCV), *proces cu parametru de timp continuu și valori discrete* (CTDV) sau *proces cu parametru de timp continuu și valori continue* (CTCV). Acestea nu sunt însă singurele posibile, ci mai distingem multe alte tipuri de procese atât în funcție de mulțimea parcursă de parametrul de timp cât și de mulțimea unde procesul ia valori (vezi [16]).

Un alt tip de clasificare este cel dat de anumite proprietăți ale procesului, în special de proprietăți ale funcției sale de corelație. Astfel dacă funcția de (auto)corelație este invariantă la translație vorbim despre *proces staționare*, iar dacă funcția de (auto)corelație a procesului are o reprezentare integrală, vorbim despre *proces armonizabile* (vezi și [16 §3.1, §3.4]). Având în vedere însă că în procesarea semnalelor cea mai bună modelare o conferă procesele staționare, vom introduce în secțiunea următoare mai în detaliu acest tip de proces aleator.

4.1.2. Caracteristici ale proceselor aleatoare

Vom începe prin a aminti câteva mărimi ce caracterizează o *variabilă aleatoare*. *Funcția de probabilitate* a unei variabile aleatoare X , notată $P_X(\cdot)$, va fi o funcție definită prin relația $P_X[B] = P[X \in B]$, pentru fiecare boreliană B .

Două variabile aleatoare X și Y se zic *identice distribuite*, $X \sim Y$, dacă funcțiile lor de probabilitate sunt egale, i.e. $P_X[B] = P_Y[B]$, pentru orice boreliană B .

Funcția de repartiție a unei variabile aleatoare X va fi dată de relația $F_X(x) = P[X \leq x]$, pentru orice x .

În cazul unei variabile aleatoare discrete distingem *funcția de probabilitate în masă* (sau **Probability Mass Function**) dată de $p_X(x) = P[X = x]$, care permite exprimarea funcției de probabilitate ca o sumă $P_X[B] = P[X \in B] = \sum_{p_X(x) > 0} p_X(x)$.

Media unei variabile aleatoare, $E[X]$ se definește prin integrala Stieltjes $E[X] = \int_{-\infty}^{\infty} x dF_X(x)$, în cazul unei variabile aleatoare continue sau prin suma $E[X] = \sum_{p_X(x) > 0} xp_X(x)$, în cazul unei variabile aleatoare discrete.

În funcție de medie definim *dispersia* (sau *varianța*) unei variabile aleatoare prin $Var[X] = E[(X - E[X])^2] = E[X^2] - E^2[X]$.

Pentru două variabile aleatoare definim *covarianța*, ca fiind dată de $Cov[X_1, X_2] = E[X_1 X_2] - E[X_1]E[X_2] = E[(X_1 - E[X_1])(X_2 - E[X_2])]$. Dacă $X_1 = X_2$, atunci avem $Cov[X, X] = Var[X]$.

Dacă notăm cu $s_X^2 := Var[X]$ (*deviația standard* a variabilei aleatoare X) atunci *corelația* a două variabile aleatoare X_1, X_2 se definește ca $r(X_1, X_2) = \frac{Cov[X_1, X_2]}{s_{X_1}^2 s_{X_2}^2}$.

Având în vedere că un proces aleator poate fi asimilat unei funcții definite pe mulțimea parametrului de timp cu valori variabile aleatoare, toate mărimile definite mai sus se aplică componentelor proceselor aleatoare. Astfel corelația dintre două variabile aleatoare se transformă în *funcție de (auto)corelație*. Dacă prin abuz de notație vom folosi pentru un proces aleator notația generică X , atunci funcția de (auto)corelație a procesului se va defini prin

$$r_X(s, t) = r(X(s), X(t)),$$

unde $X(s)$ și $X(t)$ reprezintă variabilele aleatoare asumate de procesul stochastic X la momentul s , respectiv t .

4.2. Procese staționare cu parametru discret de timp

Am amintit la finele lui 4.1.1 proprietatea unui proces aleator de a fi staționar. De fapt distingem între două tipuri de staționaritate. Un proces aleator este *staționar* dacă funcția de probabilitate în masă (PMF) comună pentru orice eșantion finit este invariantă la translația în timp. Această condiție poate fi „relaxată” prin impunerea invarianței la translația în timp a funcției de (auto)corelație. În aceste condiții, vorbim despre *staționaritate slabă*. Invarianța la translația în timp a funcției de (auto)corelație se traduce prin dependența acesteia de diferența dintre variabilele sale și nu de fiecare variabilă în parte. Putem așadar spune că pentru un proces slab staționar are loc

$$r_X[k] = r(X[n], X[n+k]) = E[X[n]X[n+k]], \quad (4.1.1)$$

de unde deducem că $r_X[0] > 0$. În plus, este ușor de observat că funcția de (auto)corelație astfel definită este o funcție pară, adică are loc $r_X[k] = r_X[-k]$.

Exemplul 4.1 (Zgomot alb gaussian): Zgomotul alb gaussian este un proces aleator $X[n]$, independent identic distribuit (IID) cu distribuția dată de

$$p_X(x) = \frac{1}{\sqrt{2\pi s^2}} \exp\left(-\frac{1}{2s^2} x^2\right).$$

Fiecare variabilă aleatoare componentă a unui zgomot alb $X[n_0]$ este de medie nulă și are aceeași varianță. Având în vedere că este vorba de un proces IID, acesta este și staționar. Staționaritatea fiind însă o condiție prea restrictivă, vom defini în continuare zgomotul alb mai general (adică să fie slab staționar). Astfel, *un zgomot alb este un proces slab staționar de medie nulă cu varianța identică s^2 cu eșantioane necorelate*. În acest caz obținem pentru funcția de (auto)corelație

$$\begin{aligned} r_X[k] &= E[X[n]X[n+k]] \\ &= E[X[n]]E[X[n+k]] \quad k \neq 0 \quad (\text{eșantioane necorelate și de medie nulă}) \\ &= E[X^2[n]] = s^2 \quad k = 0 \quad (\text{eșantioane de varianța egală}). \end{aligned}$$

Ca atare avem

$$r_X[k] = s^2 d[k]. \quad (4.1.2)$$

Exemplul 4.2 (Proces medie mobilă): Procesul *medie mobilă* este un proces definit prin

$$X[n] = \frac{1}{2}(U[n] + U[n-1]) \quad -\infty < n < \infty,$$

unde $U[n]$ este un zgomot alb de varianță s_U^2 . Următorul cod MATLAB este folosit pentru a obține o realizare a unui proces medie mobilă.

```
randn('state', 0)
u = randn(21,1);
for i=1:21
    if i==1
        x(i,1)=0.5*(u(1)+randn(1,1)); %initializarea sirului
    else
        x(i,1)=0.5*(u(i)+u(i-1));
    end
end
end
```

Procesul medie mobilă exprimându-se ca și combinație liniară a unui zgomot alb, deducem cu ușurință că acesta este de medie nulă și staționar, având funcția de (auto)corelație

$$r_X[k] = \begin{cases} \frac{s_U^2}{2} & k = 0 \\ \frac{s_U^2}{4} & k = \pm 1 \\ 0 & \text{in rest.} \end{cases} \quad (4.1.3)$$

Exemplul 4.3 (Proces autoregresiv): Un proces autoregresiv este un proces slab staționar de medie nulă ce satisface ecuația cu diferențe recursivă

$$X[n] = aX[n-1] + U[n] \quad -\infty < n < \infty, \quad (4.1.4)$$

unde $|a| < 1$ și $U[n]$ un zgomot alb. Un proces de acest tip se dezvoltă după modelul

M

$$X[0] = aX[-1] + U[0]$$

$$X[1] = aX[0] + U[1]$$

$$X[2] = aX[1] + U[2]$$

M

Să observăm că $X[n]$ depinde doar de valorile prezente și trecute ale zgomotului alb $U[n]$. Se poate chiar arăta că

$$E[X[n]U[n+k]] = 0, k \geq 1. \quad (4.1.5)$$

Condiția $|a| < 1$ este necesară pentru a asigura staționaritatea procesului. Dacă considerăm varianța zgomotului alb $S_U^2 = 1 - a^2$ obținem $r_X[0] = 1$. Dăm în continuare un cod MATLAB pentru generarea unor realizări de procese autoregresive.

```
clear all
randn('state',0)
a1 = 0.25; a2 = 0.98;
varu1 = 1-a1^2; varu2 = 1-a2^2;
varx1 = varu1/(1-a1^2); varx2 = varu2/(1-a2^2);
x1(1,1) = sqrt(varx1)*randn(1,1);
x2(1,1) = sqrt(varx2)*randn(1,1);
for n = 2:31
    x1(n,1) = a1*x1(n-1) + sqrt(varu1)*randn(1,1);
    x2(n,1) = a2*x2(n-1) + sqrt(varu2)*randn(1,1);
end
```

Să determinăm acum funcția de (auto)corelație. Din (4.1.4) avem pentru $k \geq 1$

$$\begin{aligned} r_X[k] &= E[X[n]X[n+k]] \\ &= E[X[n](aX[n+k-1] + U[n+k])] \\ &= aE[X[n]X[n+k-1]] \\ &= ar_X[k-1] \end{aligned} \quad (4.1.6)$$

Pentru trecerea de la a doua egalitate la a treia am folosit (4.1.5). Soluția ecuației liniare cu diferențe recursivă se observa imediat a fi $r_X[k] = ca^k$, cu c o constantă și $k \geq 1$. Cum pentru $k=1$ rezultă că $r_X[1] = ca$, deducem că $c = r_X[0]$. Se poate chiar arăta că

$$r_X[k] = \frac{S_U^2}{1-a^2} a^{|k|} \quad -\infty < k < \infty. \quad (4.1.7)$$

4.3. Procese aleatoare la ieșirea dintr-un filtru liniar

Considerăm efectele unui sistem liniar invariant la translație (linear shift invariant system – LSI) asupra unui proces slab staționar cu parametru de timp discret. Vom mai numi un astfel de sistem *filtru*. Fie așadar $U[n]$ procesul staționar care intră (input-ul) în filtru și $X[n]$ procesul staționar la ieșire (output-ul). Un bun exemplu este procesul medie mobilă (Moving Average) care apare ca output la un proces de tip zgomot alb gaussian (white Gaussian noise). Mai precis un proces MA are forma $X[n] = (U[n] + U[n-1])/2$, unde $U[n]$ este un proces de tip zgomot alb gaussian (wGn) și poate fi obținut prin acțiunea sistemului (filtrului)

$$h[k] = \begin{cases} \frac{1}{2} & k = 0 \\ \frac{1}{2} & k = 1 \\ 0 & \text{în rest} \end{cases}$$

asupra unui zgomot alb (proces wGn) $U[n]$. În adevăr, dacă acțiunea sistemului se

traduce prin produsul de convoluție discret $X[n] = \sum_{k=-\infty}^{\infty} h[k]U[n-k]$ obținem output-ul

$$\begin{aligned} X[n] &= h[0]U[n] + h[1]U[n-1] \\ &= \frac{1}{2}U[n] + \frac{1}{2}U[n-1] \\ &= \frac{1}{2}(U[n] + U[n-1]) \end{aligned}$$

adică un proces MA.

În general un sistem liniar SI (Shift Invariant) va fi specificat în forma de mai sus cu $-\infty < k < \infty$ sau în mod echivalent prin *funcția sa de sistem*, definită ca transformata în z (Laplace) a răspunsului la impuls. Funcția de sistem este așadar dată de

$$H(z) = \sum_{k=-\infty}^{\infty} h[k]z^{-k}. \quad (4.3.1)$$

În plus mai avem nevoie de *funcția de transfer frecvențială* a sistemului liniar SI care se definește ca fiind transformata Fourier discretă a răspunsului în impuls, deci dată de

$$H(f) = \sum_{k=-\infty}^{\infty} h[k]\exp(-2\pi ifk). \quad (4.3.2)$$

Această funcție ne indică efectul sistemului asupra unei secvențe de input sinusoidale complexe $u[n] = \exp(2\pi if_0 n)$ pentru $-\infty < n < \infty$. Se poate arăta că răspunsul sistemului la acest input este $x[n] = H(f_0)\exp(2\pi if_0 n) = H(f_0)u[n]$. Observăm că acțiunea sistemului modifică amplitudinea sinusoidalei complexe prin $|H(f_0)|$ și faza cu $\angle H(f_0)$, dar păstrează în rest secvența sinusoidală complexă. De asemenea, este de notat că funcția de transfer frecvențială se poate obține imediat din funcția de sistem prin formula $H(f) = H(\exp(2\pi if))$. Pentru procesul MA din exemplul de mai sus

funcția de sistem este $H(z) = \frac{1}{2} + \frac{1}{2}z^{-1}$, iar funcția de transfer frecvențială este funcția de sistem în care z este înlocuit cu $\exp(2\pi if)$, ceea ce ne conduce la

$$H(f) = \frac{1}{2} + \frac{1}{2}\exp(-2\pi if).$$

Vom introduce cu ajutorul unui exemplu caracteristicile unui proces aleator de output.

Exemplul 4.4. Considerăm $U[n]$ un proces staționar de medie m_U și funcție de (auto)corelație $r_U[k]$ ca input într-un sistem liniar SI cu răspunsul la impuls dat de

$$h[k] = \begin{cases} h[0] & k = 0 \\ h[1] & k = 1 \\ 0 & \text{în rest.} \end{cases}$$

Acest sistem liniar se mai numește *filtru cu răspuns la impuls finit*, deoarece are un număr finit de eșantioane nenule ca răspuns la impuls. Căutăm să determinăm dacă output-ul din acest filtru este de asemenea staționar și dacă da atunci cum se modifică media și funcția de (auto)corelație.

Procesul de output va fi așadar

$$X[n] = h[0]U[n] + h[1]U[n-1],$$

iar șirul mediilor va fi

$$\begin{aligned} E[X[n]] &= h[0]E[U[n]] + h[1]E[U[n-1]] \\ &= h[0]m_U + h[1]m_U \\ &= (h[0] + h[1])m_U \end{aligned}$$

deci constantă în timp, dată de

$$m_X = (h[0] + h[1])m_U.$$

De asemenea din (4.3.2) aceasta se poate scrie sub forma

$$m_X = \sum_{k=-\infty}^{\infty} h[k]\exp(-2\pi ifk) \Big|_{f=0} m_U = H(0)m_U.$$

Așadar media output-ului este modificată de funcția de transfer frecvențială evaluată în $f = 0$. Rămâne de văzut dacă $E[X[n]X[n+k]]$ depinde sau nu de n . Calculând

$$\begin{aligned} E[X[n]X[n+k]] &= E[(h[0]U[n] + h[1]U[n-1])(h[0]U[n+k] + h[1]U[n+k-1])] \\ &= h^2[0]E[U[n]U[n+k]] + h[0]h[1]E[U[n]U[n+k-1]] \\ &\quad + h[1]h[0]E[U[n-1]U[n+k]] + h^2[1]E[U[n-1]U[n+k-1]] \\ &= (h^2[0] + h^2[1])r_U[k] + h[0]h[1]r_U[k-1] + h[1]h[0]r_U[k+1], \end{aligned}$$

deci nu depinde de n . Ca atare $X[n]$ este proces staționar cu funcția de (auto)corelație

$$r_X[k] = (h^2[0] + h^2[1])r_U[k] + h[0]h[1]r_U[k-1] + h[1]h[0]r_U[k+1]. \quad (4.3.3)$$

Să arătăm în continuare că funcția de (auto)corelație a procesului output dintr-un sistem liniar SI se poate scrie sub forma unei convoluții multiple de șiruri. Pentru aceasta considerăm (4.3.3) și fie

$$\begin{aligned}g[0] &= h^2[0]h^2[1] \\g[1] &= h[0]h[1] \\g[-1] &= h[1]h[0]\end{aligned}$$

cu zero în rest. Atunci

$$\begin{aligned}r_x[k] &= g[0]r_v[k] + g[1]r_v[k-1] + g[-1]r_v[k+1] \\&= \sum_{j=-1}^1 g[j]r_v[k-j] \\&= g[k] * r_v[k] \quad (\text{definiția sumei de convoluție}).\end{aligned}\tag{4.3.4}$$

De asemenea, este ușor de arătat (prin calcul direct) că

$$\begin{aligned}g[k] &= \sum_{j=-1}^0 h[-j]h[k-j] \\&= h[-k] * h[k]\end{aligned}\tag{4.3.5}$$

și ca atare din (4.3.4) și (4.3.5) obținem rezultatul final

$$\begin{aligned}r_x[k] &= (h[-k] * h[k]) * r_v[k] \\&= h[-k] * h[k] * r_v[k].\end{aligned}\tag{4.3.6}$$

Am omis parantezele în (4.3.6) ținând cont de asociativitatea și comutativitatea operației de convoluție.

Pentru a obține densitatea spectrală a procesului $X[n]$ observăm din (4.3.2) că transformata Fourier a răspunsului la impuls este funcția de transfer frecvențială și deci

$$F \{h[k]\} = H(f)$$

$$F \{h[-k]\} = H^*(f)$$

unde cu F am notat transformata Fourier discretă. Transformata Fourier aplicată relației (4.3.6) ne dă

$$P_x(f) = H^*(f)H(f)P_v(f),$$

sau

$$P_x(f) = |H(f)|^2 P_v(f).$$

Aceasta este relația de bază între densitatea spectrală a procesului de output și cea a procesului de input, adică *densitatea spectrală de output este densitatea spectrală de input înmulțită cu pătratul magnitudinii funcției de transfer frecvențiale*. Faptele enunțate mai sus pot fi cumulate în următoarea teoremă.

Teorema 4.1 (Caracteristici ale procesului aleator la output-ul unui sistem liniar SI). *Dacă un proces slab staționar $U[n]$ de medie m_U și funcția de (auto)corelație $r_U[k]$ este input-ul unui sistem liniar SI care are un răspuns la impuls $h[k]$ și răspuns în frecvență $H(f)$, atunci procesul de output $X[n] = \sum_{k=-\infty}^{\infty} h[k]U[n-k]$ este de asemenea slab staționar și*

$$\mathbf{m}_X = \sum_{k=-\infty}^{\infty} h[k] \mathbf{m}_U = H(0) \mathbf{m}_U \quad (4.3.7)$$

$$r_X[k] = h[-k] * h[k] * r_U[k] \quad (4.3.8)$$

$$P_X(f) = |H(f)|^2 P_U(f). \quad (4.3.9)$$

Demonstrație. Media la output este dată de

$$\begin{aligned} \mathbf{m}_X &= E[X[n]] = E\left[\sum_{k=-\infty}^{\infty} h[k]U[n-k]\right] \\ &= \sum_{k=-\infty}^{\infty} h[k]E[U[n-k]] \\ &= \sum_{k=-\infty}^{\infty} h[k] \mathbf{m}_U = H(0) \mathbf{m}_X \quad (U[n] \text{ este staționar}) \end{aligned}$$

și nu depinde de n . Pentru a determina dacă putem defini densitatea spectrală considerăm $E[X[n]X[n+k]]$. Aceasta devine

$$\begin{aligned} E[X[n]X[n+k]] &= E\left[\sum_{i=-\infty}^{\infty} h[i]U[n-i] \sum_{j=-\infty}^{\infty} h[j]U[n+k-j]\right] \\ &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h[i]h[j]E[U[n-i]U[n+k-j]] \end{aligned}$$

deoarece $U[n]$ este staționar. Cum expresia anterioară nu depinde de n deducem că $X[n]$ este de asemenea staționar. Funcția de (auto)corelație este așadar

$$\begin{aligned} r_X[k] &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h[i]h[j]r_U[(k+i)-j] \\ &= \sum_{i=-\infty}^{\infty} h[i] \sum_{j=-\infty}^{\infty} h[j]r_U[(k+i)-j], \end{aligned}$$

unde am notat

$$g[m] = h[m] * r_U[m]. \quad (4.3.10)$$

Pe de altă parte avem că

$$\begin{aligned} r_X[k] &= \sum_{i=-\infty}^{\infty} h[i]g[k+i] \\ &= \sum_{l=-\infty}^{\infty} h[-l]g[k-l] \quad (\text{punem } l = -i) \\ &= h[-k] * g[k]. \end{aligned}$$

Din (4.3.10) însă $g[k] = h[k] * r_U[k]$ și ca atare

$$\begin{aligned} r_X[k] &= h[-k] * (h[k] * r_U[k]) \\ &= h[-k] * h[k] * r_U[k] \end{aligned} \quad (4.3.11)$$

Pentru formula de densitate spectrală (4.3.9) aplicăm transformata Fourier la (4.3.11) și ținem cont de faptul că $F\{h[-k]\} = H^*(f)$.

!

Un caz particular special apare când procesul de input este zgomot alb. Folosind particularitatea zgomotului alb și anume, că densitatea spectrală este $P_U(f) = S_U^2$, conform (4.3.9) obținem

$$P_X(f) = |H(f)|^2 S_U^2. \quad (4.3.12)$$

Folosind apoi $r_U[k] = S_U^2 d[k]$ în (4.3.8) funcția de (auto)corelație la output devine

$$r_X[k] = h[-k] * h[k] * S_U^2 d[k]$$

și ținând cont că $h[k] * d[k] = h[k]$

$$\begin{aligned} r_X[k] &= S_U^2 h[-k] * h[k] \\ &= S_U^2 \sum_{i=-\infty}^{\infty} h[-i] h[k-i]. \end{aligned}$$

Punând $m = -i$ obținem în cele din urmă

$$r_X[k] = S_U^2 \sum_{i=-\infty}^{\infty} h[m] h[m+k] \quad -\infty < k < \infty. \quad (4.3.13)$$

Exemplul 4.5 (Proces autoregresiv (AR)): Vom determina funcția de (auto)corelație și densitatea spectrală pentru acest tip de procese folosind conceptele descrise anterior de sistem liniar SI. Reamintim că un proces AR se definește prin $X[n] = aX[n-1] + U[n]$ și poate fi privit ca outputul unui sistem liniar SI, având funcția de sistem dată de

$$H(z) = \frac{1}{1 - az^{-1}}$$

și input-ul fiind un zgomot alb $U[n]$. Fie acum $u[n]$ un șir determinist cu transformata z $U(z)$ și $x[n]$ șirul determinist de output cu transformata z corespunzătoare $X(z)$. Din definiția funcției de sistem avem

$$H(z) = \frac{X(z)}{U(z)}$$

de unde deducem formula pentru transformata z a procesului de output

$$\begin{aligned} X(z) &= H(z)U(z) \\ &= \frac{1}{1 - az^{-1}} U(z). \end{aligned}$$

Astfel,

$$X(z) - az^{-1}X(z) = U(z)$$

ceea ce prin transformata z inversă se traduce în ecuația cu diferențe

$$x[n] - ax[n-1] = u[n] \quad (4.3.14)$$

care este echivalent cu definiția procesului AR când înlocuim input-ul și output-ul cu procese aleatoare.

Densitatea spectrală la output se obține din (4.3.12) ca

$$\begin{aligned} P_X(f) &= |H(\exp(2\pi if))|^2 S_U^2 \\ &= \frac{S_U^2}{|1 - a \exp(-2\pi if)|^2} \end{aligned} \quad (4.3.15)$$

cea ce coincide cu rezultatele anterioare. Pentru a determina acum funcția de (auto)corelație putem lua transformata Fourier inversă la (4.3.15) sau putem folosi (4.3.13). Cea de-a doua abordare este în general mai ușoară. Pentru a găsi răspunsul la impuls putem folosi (4.3.14) cu input-ul $d[n]$ astfel ca output-ul să fie prin definiție $h[n]$. Cum sistemul liniar SI se presupune a fi causal, va trebui să determinăm o soluție pentru ecuația cu diferențe $h[n] = ah[n-1] + d[n]$ pentru $n > 0$ cu condiția inițială $h[-1] = 0$. Condiția inițială este zero din cauza presupunerii că sistemul liniar este causal, un asemenea sistem neputând produce un output nenul $-h[-1]$ – înainte de a putea fi aplicat unui input, în cazul de față la momentul $n = 0$ cu inputul $d[n]$. Obținem astfel $h[n] = a^n u_s[n]$, unde $u_s[n]$ este pasul unitate. Așadar (4.3.13) devine, pentru $k \geq 0$

$$\begin{aligned} r_x[k] &= S_U^2 \sum_{m=-\infty}^{\infty} a^m u_s[m] a^{m+k} u_s[m+k] \\ &= S_U^2 a^k \sum_{m=0}^{\infty} a^{2m} \quad (m \geq 0 \text{ si } m+k \geq 0 \text{ pentru termenii nenuli ai sumei}) \\ &= S_U^2 \frac{a^k}{1-a^2} \quad (\text{deoarece } |a| < 1) \end{aligned}$$

și ca atare avem pentru k arbitrar

$$r_x[k] = S_U^2 \frac{a^k}{1-a^2}.$$

Să mai notăm că sistemul liniar SI amintit aici este un filtru cu răspuns la impuls infinit, deoarece răspunsul la impuls dat de $h[n] = a^n u_s[n]$ este de lungime infinită.

Exemplul 4.6. Proces aleator medie mobilă (Moving Average).

Folosim (4.3.13) pentru a determina funcția de (auto)corelație a unui proces MA. Reamintim definiția unui proces MA ca $X[n] = \frac{U[n] + U[n-1]}{2}$, cu $U[n]$ un zgomot alb (wGn). Este deja evident ca funcția de sistem este dată de $H(z) = \frac{1}{2} + \frac{1}{2}z^{-1}$ și răspunsul la impuls este $h[m] = \frac{1}{2}$ pentru $m = 0, 1$ și zero în rest. Din (4.3.13) obținem

$$\begin{aligned} r_x[k] &= S_U^2 \sum_{m=-\infty}^{\infty} h[m]h[m+k] \\ &= S_U^2 \sum_{m=0}^1 h[m]h[m+k], \end{aligned}$$

deci pentru $k \geq 0$

$$r_x[k] = \begin{cases} s_U^2 \sum_{m=0}^1 h^2[m] & k=0 \\ s_U^2 \sum_{m=0}^1 h[m]h[m+1] & k=1. \\ 0 & k \geq 2 \end{cases}$$

În cele din urmă obținem

$$r_x[k] = \begin{cases} s_U^2 \left(\left(\frac{1}{2} \right)^2 + \left(\frac{1}{2} \right)^2 \right) = \frac{s_U^2}{2} & k=0 \\ s_U^2 \left(\frac{1}{2} \right) \left(\frac{1}{2} \right) = \frac{s_U^2}{4} & k=1 \\ 0 & k \geq 2 \end{cases}$$

ceea ce coincide cu ce am obținut anterior.

4.4. Interpretarea densității spectrale

Vom arăta că *densitatea spectrală, integrată pe o bandă de frecvență ne dă puterea medie în acea bandă*. Aceasta permite densității spectrale să fie privită ca puterea medie pe unitatea de frecvență. Vom mai aminti o metodă de măsurare a puterii medii a unui proces slab staționar într-o bandă îngustă de frecvență. Pentru aceasta vom filtra procesul cu ajutorul unui filtru de bandă îngustă cu funcție de transfer frecvențială

$$H(f) = \begin{cases} 1 & -f_0 - \frac{\Delta f}{2} \leq f \leq -f_0 + \frac{\Delta f}{2}, f_0 - \frac{\Delta f}{2} \leq f \leq f_0 + \frac{\Delta f}{2} \\ 0 & \text{in rest} \end{cases}$$

Lățimea benzii de trecere a filtrului Δf se presupune foarte mică. Dacă un proces slab staționar $X[n]$ este input-ul în acest filtru, atunci procesul staționar de output-ul $Y[n]$ va fi compus din componentele de frecvență situate în interiorul benzii Δf , restul fiind „filtrate”. Puterea medie totală la procesul de output $Y[n]$ este $r_Y[0]$ și reprezintă suma puterilor medii a procesului $X[n]$ din range-ul benzilor $\left[-f_0 - \frac{\Delta f}{2}, -f_0 + \frac{\Delta f}{2} \right]$ și $\left[f_0 - \frac{\Delta f}{2}, f_0 + \frac{\Delta f}{2} \right]$. Acestea se deduc din fomula

$$r_Y[0] = \int_{-\frac{\Delta f}{2}}^{\frac{\Delta f}{2}} P_Y(f) df .$$

Folosind acum (4.3.9) și definiția răspunsului în frecvență a filtrului de bandă îngustă obținem

$$\begin{aligned}
r_Y[0] &= \int_{-\frac{\Delta f}{2}}^{\frac{\Delta f}{2}} P_Y(f) df \\
&= \int_{-\frac{\Delta f}{2}}^{\frac{\Delta f}{2}} |H(f)|^2 P_X(f) df \\
&= \int_{-f_0 - \frac{\Delta f}{2}}^{-f_0 + \frac{\Delta f}{2}} 1 \cdot P_X(f) df + \int_{f_0 - \frac{\Delta f}{2}}^{f_0 + \frac{\Delta f}{2}} 1 \cdot P_X(f) df \\
&= 2 \int_{f_0 - \frac{\Delta f}{2}}^{f_0 + \frac{\Delta f}{2}} 1 \cdot P_X(f) df \quad (\text{deoarece } P_X(-f) = P_X(f)).
\end{aligned}$$

Dacă lăsăm $\Delta f \rightarrow 0$, astfel încât $P_X(f) \rightarrow P_X(f_0)$ în intervalul de integrare, aceasta devine

$$r_Y[0] = 2P_X(f_0)\Delta f$$

sau

$$P_X(f_0) = \frac{1}{2} \frac{r_Y[0]}{\Delta f}.$$

Cum însă $r_Y[0]$ este puterea medie totală în raport cu componentele de frecvență din cadrul benzilor considerate, care este dublul puterii medii totale în banda pozitivă de frecvență, avem că

$$P_X(f_0) = \frac{\text{Puterea medie totala in banda } \left[f_0 - \frac{\Delta f}{2}, f_0 + \frac{\Delta f}{2} \right]}{\Delta f}. \quad (4.4.1)$$

Aceasta înseamnă că *densitatea spectrală* $P_X(f_0)$ este *puterea medie a lui* $X[n]$ *într-o bandă îngustă de frecvență în jurul lui* $f = f_0$ *împărțită la lățimea benzii*. Se justifică astfel terminologia de *densitate*. Pentru a obține puterea medie din cadrul unei benzi de frecvență din densitatea spectrală inversăm relația (4.4.1) pentru a obține

$$\text{Puterea medie totala in banda } \left[f_0 - \frac{\Delta f}{2}, f_0 + \frac{\Delta f}{2} \right] = P_X(f_0)\Delta f,$$

ceea ce reprezintă *aria* de sub graficul densității spectrale. Mai general, pentru o bandă de frecvență arbitrară avem

$$\text{Puterea medie totala in banda } [f_1, f_2] = \int_{f_1}^{f_2} P_X(f) df$$

după cum am enunțat mai devreme.

4.5. Filtrare Wiener

Abordarea generică a determinării mediei și funcției de (auto)corelație sau a mediei și densității spectrale se va numi *filtrare Wiener*, cu toate că vor fi de fapt patru probleme distincte care se vor grupa sub această denumire și anume *filtrare*, *netezire*, *predicție* și *interpolare*. În problema de filtrare presupunem ca un semnal $S[n]$ este perturbat (corupt) aditiv de un zgomot $W[n]$ rezultând un proces $X[n]$. Dorim să estimăm $S[n]$ prin filtrarea lui $X[n]$ cu ajutorul unui filtru liniar SI cu

răspunsul la impuls $h[k]$. Intenționăm ca filtrul să reducă zgomotul și să permită semnalului să „treacă”. Filtrul estimează un eșantion particular al semnalului, de ex. $S[n_0]$, procesând eșantionul curent $X[n_0]$ precum și eșantioanele trecute $\{X[n_0-1], X[n_0-2], \mathbf{K}\}$. Ca atare, filtrul se presupune a fi cauzal, adică cu răspuns în impuls $h[k]=0$ pentru $k < 0$. Aceasta ne dă estimarea

$$\hat{S}[n_0] = \sum_{k=0}^{\infty} h[k]X[n_0-k] \quad (4.5.1)$$

care depinde de eșantionul curent și eșantioanele anterioare observate. Presupunem eșantioanele anterioare corelate cu semnalul prezent, ceea ce ar trebui să îmbunătățească considerabil estimarea. Această problemă de *filtrare* poate fi implementată în *timp real*.

A doua problemă, *netezirea*, se deosebește de filtrare prin faptul că filtrul aplicat nu este obligatoriu cauzal. Ca atare, estimarea ia forma

$$\hat{S}[n_0] = \sum_{k=-\infty}^{\infty} h[k]X[n_0-k] \quad (4.5.2)$$

unde $\hat{S}[n_0]$ depinde de eșantioanele trecute, prezente și *viitoare* ale lui $X[n]$. Este clar că aceasta nu poate fi realizată în timp real, totuși poate fi aproximată dacă permitem o întârziere înaintea estimării. Această întârziere este necesară acumulării eșantioanelor $\{X[n_0], X[n_0+1], \mathbf{K}\}$ înaintea estimării lui $\hat{S}[n_0]$.

În cadrul celorlalte două probleme observăm eșantioane ale procesului slab staționar $X[n]$ și dorim să estimăm eșantioanele neobservate. Pentru *predicție*, numită și *extrapolare*, observăm eșantioanele prezente și trecute $\{X[n_0], X[n_0-1], \mathbf{K}\}$ și dorim să estimăm un eșantion viitor, $X[n_0+L]$, pentru L un întreg pozitiv. Această problemă se mai numește *predicție de pas L* . Pentru simplitate vom considera doar predicție de pas 1, adică $L=1$. Predictorul (liniar) va avea forma

$$\hat{X}[n_0+1] = \sum_{k=0}^{\infty} h[k]X[n_0-k] \quad (4.5.3)$$

care folosește, evident, un filtru cauzal.

Pentru *interpolare*, observăm eșantioanele $\{\mathbf{K}, X[n_0-1], X[n_0+1], \mathbf{K}\}$ și dorim să estimăm $X[n_0]$. Interpolatorul va fi dat de

$$\hat{X}[n_0] = \sum_{\substack{k=-\infty \\ k \neq 0}}^{\infty} h[k]X[n_0-k] \quad (4.5.4)$$

care este un filtru necauzal. Pentru implementarea practică a formulelor (4.5.1) – (4.5.4) va trebui să trunchiem răspunsul în impuls la un număr finit de eșantioane.

Pentru a determina răspunsurile în impuls optime vom adopta metoda erorii în medie pătratică. Estimările care folosesc aceste metode se numesc generic *filtre Wiener*. Din cele patru probleme vom aborda netezirea și predicția.

4.5.1. Netezire Wiener

Observând $X[n] = S[n] + W[n]$ pentru $-\infty < n < \infty$, dorim să estimăm $S[n_0]$ folosind (4.5.2). Vom presupune că atât $S[n]$ cât și $W[n]$ sunt ambele procese slab staționare de medie nulă de funcție de (auto)corelație (densitate spectrală) dată. De asemenea, deoarece nu avem motive de a presupune altceva, vom considera că semnalul și zgomotul sunt necorelate, adică $E[S[n_1]W[n_2]] = 0$ pentru orice n_1 și n_2 . Eroarea în medie pătratică pentru această problemă se definește ca

$$\text{emp} = E[e^2[n_0]] = E[(S[n_0] - \hat{S}[n_0])^2]$$

unde $e[n_0] = S[n_0] - \hat{S}[n_0]$ este eroarea. Pentru a minimiza EMP (Eroarea în Medie Pătratică) folosim faptul că eroarea trebuie să fie ortogonală (necorelată) cu datele. Aceasta se traduce prin condiția

$$E[e[n_0]X[n_0 - l]] = 0 \quad -\infty < l < \infty.$$

Obținem astfel

$$\begin{aligned} E[(S[n_0] - \hat{S}[n_0])X[n_0 - l]] &= 0 \\ E\left[\left(S[n_0] - \sum_{k=-\infty}^{\infty} h[k]X[n_0 - k]\right)X[n_0 - l]\right] &= 0 \end{aligned}$$

ceea ce implică

$$E[S[n_0]X[n_0 - l]] = \sum_{k=-\infty}^{\infty} h[k]E[X[n_0 - k]X[n_0 - l]]. \quad (4.5.5)$$

Dar

$$\begin{aligned} E[S[n_0]X[n_0 - l]] &= E[S[n_0](S[n_0 - l] + W[n_0 - l])] \\ &= E[S[n_0]S[n_0 - l]] \quad (\text{zgomotul și semnalul sunt necorelate și de medie nulă}) \\ &= r_s[l] \end{aligned}$$

și

$$\begin{aligned} E[X[n_0 - k]X[n_0 - l]] &= E[(S[n_0 - k] + W[n_0 - k])(S[n_0 - l] + W[n_0 - l])] \\ &= E[S[n_0 - k]S[n_0 - l]] + E[W[n_0 - k]W[n_0 - l]] \\ &= r_s[l - k] + r_w[l - k]. \end{aligned}$$

Mulțimea infinită de ecuații liniare simultane devine din (4.5.5)

$$r_s[l] = \sum_{k=-\infty}^{\infty} h[k](r_s[l - k] + r_w[l - k]) \quad -\infty < l < \infty. \quad (4.5.6)$$

Să observăm că ecuațiile nu depind de n_0 , deci soluția pentru răspunsul în impuls optimal va fi același pentru orice n_0 . Observând că partea dreaptă în (4.5.6) este de fapt un produs de convoluție discret avem că

$$r_s[l] = h[l] * (r_s[l] + r_w[l]).$$

Aplicăm transformata Fourier și obținem

$$P_s(f) = H(f)(P_s(f) + P_w(f))$$

de unde funcția de transfer frecvențială a filtrului Wiener optimal de netezire este

$$H_{\text{opt}}(f) = \frac{P_s(f)}{P_s(f) + P_w(f)}. \quad (4.5.7)$$

Răspunsul în impuls optimal se obține luând transformata Fourier inversă a relației (4.5.7).

Exemplul 4.7. (netezire Wiener pentru un semnal AR cu zgomot alb)

Considerăm un semnal reprezentat printr-un proces AR perturbat de un zgomot alb cu varianța s_w . Densitățile spectrale vor fi date de

$$P_s(f) = \frac{s_u^2}{|1 - a \exp(-2\pi j f)|^2}.$$

$$P_w(f) = s_w^2$$

Densitatea spectrală și netezirea Wiener a răspunsurilor în frecvență.

Pentru a implementa netezirea Wiener pentru exemplul considerat, datele au fost filtrate în domeniul de frecvență și convertite apoi în domeniul temporal. Aceasta se realizează cu ajutorul transformatei Fourier discrete inverse

$$\hat{s}[n] = \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{P_s(f)}{P_s(f) + s_w^2} X_N(f) \exp(2\pi j f n) df \quad n = 0, 1, 2, \dots, N-1$$

unde $X_N(f)$ este transformata Fourier a datelor disponibile $\{x[0], x[1], \dots, x[N-1]\}$, care este de fapt

$$X_N(f) = \sum_{n=0}^{N-1} x[n] \exp(2\pi j f n)$$

(în exemplul anterior s-a luat $N = 50$). Implementarea folosește de fapt o transformată Fourier rapidă (FFT) inversă pentru aproximarea integralei cum se vede din codul MATLAB de mai jos. Pentru a folosi FFT și FFT inversă, intervalul de frecvență a fost luat $[0, 1]$. Din cauza periodicității transformatei Fourier, aceasta nu va afecta rezultatul.

```
clear all
randn('state', 0)
a = 0.9; varu=0.5; vars=varu/(1-a^2); varw=1; N=50;
for n=0:N-1
    nn = n+1;
    if n==0
        s(nn,1) = sqrt(vars)*randn(1,1);
    else
        s(nn,1) = a*s(nn-1)+sqrt(varu)*randn(1,1);
    end
end
end
```

```

x = s+sqrt(varw)*randn(N,1);
Nfft = 1024
Ps = varu./(abs(1-a*exp(-j*2*pi*[0:Nfft-1]'/Nfft)).^2);
Hf = Ps./(Ps+varw);
sestf = Hf.*fft(x,Nfft);
sest = real(ifft(sestf,Nfft));

```

Putem de asemenea determina EMP minimă pentru a vedea cât de bine funcționează filtru de netezire. Acesta este dată de

$$\begin{aligned} \text{emp}_{\min} &= E\left[\left(S[n_0]-\hat{S}[n_0]\right)^2\right] \\ &= E\left[\left(S[n_0]-\hat{S}[n_0]\right)S[n_0]\right]-E\left[\left(S[n_0]-\hat{S}[n_0]\right)\hat{S}[n_0]\right]. \end{aligned}$$

Din ipoteza de ortogonalitate avem însă că termenul al doilea este nul, adică

$$\begin{aligned} E\left[\left(S[n_0]-\hat{S}[n_0]\right)\hat{S}[n_0]\right] &= E\left[e[n_0]\sum_{k=-\infty}^{\infty}h_{opt}[k]X[n_0-k]\right] \\ &= \sum_{k=-\infty}^{\infty}h_{opt}[k]E\left[e[n_0]X[n_0-k]\right]=0. \end{aligned}$$

Așadar, avem

$$\begin{aligned} \text{emp}_{\min} &= E\left[\left(S[n_0]-\hat{S}[n_0]\right)S[n_0]\right] \\ &= r_s[0]-E\left[\sum_{k=-\infty}^{\infty}h_{opt}[k]X[n_0-k]S[n_0]\right] \\ &= r_s[0]-\sum_{k=-\infty}^{\infty}h_{opt}[k]E\left[\left(S[n_0-k]+W[n_0-k]\right)S[n_0]\right] \\ & \qquad \qquad \qquad =E[S[n_0-k]S[n_0]]=r_s[k] \end{aligned}$$

deoarece $S[n]$ și $W[n]$ sunt necorelate și de medie nulă. Așadar EMP minimă este

$$\text{emp}_{\min} = r_s[0]-\sum_{k=-\infty}^{\infty}h_{opt}[k]r_s[k]. \quad (4.5.8)$$

Folosind teorema lui Parseval putem transpune această eroare în domeniul de frecvență prin

$$\begin{aligned} \text{emp}_{\min} &= \int_{-\frac{1}{2}}^{\frac{1}{2}}P_S(f)df - \int_{-\frac{1}{2}}^{\frac{1}{2}}H_{opt}(f)P_S(f)df \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}}\left(1-H_{opt}(f)\right)P_S(f)df \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}}\left(1-\frac{P_S(f)}{P_S(f)+P_W(f)}\right)P_S(f)df \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}}\frac{P_W(f)}{P_S(f)+P_W(f)}P_S(f)df \end{aligned}$$

și luând $r(f) = \frac{P_S(f)}{P_W(f)}$ ca fiind raportul semnal-zgomot în domeniul de frecvență avem

$$\text{emp}_{\min} = \int_{-\frac{1}{2}}^{\frac{1}{2}}\frac{P_S(f)}{1+r(f)}df. \quad (4.5.9)$$

Observăm că benzile de frecvență pentru care contribuția la EMP minimă este maximă sunt benzile unde raportul semnal-zgomot este minim sau pentru care $r(f) \rightarrow 1$.

4.5.2. Predicție

Vom considera doar predicția de pas 1. Ca mai sus criteriul EMP este folosit pentru a determina predictorul astfel încât din (4.5.3)

$$\begin{aligned} \text{emp} &= E \left[\left(X[n_0 + 1] - \hat{X}[n_0 + 1] \right)^2 \right] \\ &= E \left[\left(X[n_0 + 1] - \sum_{k=0}^{\infty} h[k] X[n_0 - k] \right)^2 \right] \end{aligned}$$

să fie minimă după $h[k]$ pentru $k \geq 0$. Folosind din nou ipoteza de ortogonalitate suntem conduși către o mulțime infinită de ecuații simultane

$$E \left[\left(X[n_0 + 1] - \sum_{k=0}^{\infty} h[k] X[n_0 - k] \right) X[n_0 - l] \right] = 0 \quad l = 0, 1, \mathbf{K} .$$

Aceste ecuații devin

$$E[X[n_0 + 1]X[n_0 - l]] = \sum_{k=0}^{\infty} h[k] E[X[n_0 - k]X[n_0 - l]]$$

sau

$$r_X[l + 1] = \sum_{k=0}^{\infty} h[k] r_X[l - k] \quad l = 0, 1, \mathbf{K} . \quad (4.5.10)$$

Observăm și de această dată că răspunsul în impuls optimal nu depinde de n_0 , deci obținem același predictor pentru fiecare eșantion. Pentru a rezolva aceste ecuații nu mai este suficientă transformata Fourier, dar din cauză că acestea trebuie verificate doar pentru $l \geq 0$, nu putem folosi transformata z .

EMP minimă poate fi evaluată folosind o metodă similară cu netezirea Wiener, anume

$$\begin{aligned} \text{emp}_{\min} &= E \left[\left(X[n_0 + 1] - \sum_{k=0}^{\infty} h_{\text{opt}}[k] X[n_0 - k] \right) X[n_0 + 1] \right] \\ &= r_X[0] - \sum_{k=0}^{\infty} h_{\text{opt}}[k] r_X[k + 1] \end{aligned} \quad (4.5.11)$$

unde $h_{\text{opt}}[k]$ este răspunsul în impuls ce verifică (4.5.10).

Exemplul 4.8 (predicția unui proces AR)

Considerăm un proces AR cu funcția de (auto)corelație dată de

$$r_x[k] = \left(\frac{S_U^2}{(1-a^2)} \right) a^{|k|} = r_x[0]a^{|k|}. \text{ Atunci din (4.5.10) avem}$$

$$r_x[0]a^{|l+1|} = \sum_{k=0}^{\infty} h[k]r_x[0]a^{|l-k|} \quad l = 0, 1, \mathbf{K}$$

și dacă punem $h[k] = 0$ pentru $k \geq 1$, avem

$$a^{|l+1|} = h[0]a^{|l|} \quad l = 0, 1, \mathbf{K} .$$

Cum $l \geq 0$, soluția este

$$h_{opt}[0] = \frac{a^{l+1}}{a^l} = a$$

sau

$$\hat{X}[n_0 + 1] = aX[n_0].$$

Având în vedere ne-dependența de n_0 a relațiilor de mai sus putem înlocui eșantionul specific cu unul mai general prin înlocuirea lui n_0 cu $n-1$. Aceasta se traduce prin

$$\hat{X}[n] = aX[n-1]. \quad (4.5.12)$$

Dacă ne amintim că un proces AR se definește ca $X[n] = aX[n-1] + U[n]$, vedem că predictorul de pas 1 optimal se obține omițând termenul $U[n]$. Aceasta se întâmplă deoarece $U[n]$ nu poate fi prezis din eșantioanele trecute $\{X[n-1], X[n-2], \dots\}$, care sunt necorelate cu $U[n]$. În plus, eroarea de predicție este dată de $e[n] = X[n] - \hat{X}[n] = X[n] - aX[n-1] = U[n]$. În final să observăm că predicția depinde doar de cel mai recent eșantion și nu de eșantioanele trecute ale lui $X[n]$. Adică, pentru a prezice $X[n_0 + 1]$ toate informațiile despre eșantioanele trecute sunt conținute în $X[n_0]$.

Soluția generală pentru (4.5.10) fiind mai complicată alegem să nu o prezentăm aici în detaliu. Vom recapitula modul de obținere a soluției și prezentăm un alt exemplu.

1. Presupunem că transformata z a funcției de corelație, care este dată de

$$P_X(z) = \sum_{k=-\infty}^{\infty} r_x[k]z^{-k}$$

poate fi scrisă ca

$$P_X(z) = \frac{S_U^2}{A(z)A(z^{-1})} \quad (4.5.13)$$

unde

$$A(z) = 1 - \sum_{k=1}^{\infty} a[k]z^{-k}.$$

Avem nevoie ca $A(z)$ să aibă toate zerourile în interiorul discului unitate, adică filtrul de transformată z egală cu $\frac{1}{A(z)}$ este stabil și cauzal.

2. Soluția lui (4.5.10) pentru răspunsul în impuls este

$$h_{opt}[k] = a[k+1] \quad k = 0, 1, \mathbf{K}$$

iar EMP minimă este

$$\text{emp}_{\min} = E \left[\left(X[n_0+1] - \hat{X}[n_0+1] \right)^2 \right] = S_U^2.$$

3. Predictorul liniar optimal devine din (4.5.3)

$$\hat{X}[n_0+1] = \sum_{k=0}^{\infty} a[k+1]X[n_0-k] \quad (4.5.14)$$

și are EMP minimă, $\text{emp}_{\min} = S_U^2$.

În mod clar, cea mai dificilă parte a soluției este punerea lui $P_X(z)$ în forma (4.5.13). În termeni de densitate spectrală această cerință se traduce prin

$$\begin{aligned} P_X(f) &= P_X(\exp(2pif)) = \frac{S_U^2}{A(\exp(2pif))A(\exp(-2pif))} \\ &= \frac{S_U^2}{A(\exp(2pif))A^*(\exp(2pif))} \\ &= \frac{S_U^2}{|A(\exp(2pif))|^2} \\ &= \frac{S_U^2}{\left| 1 - \sum_{k=1}^{\infty} a[k] \exp(-2pifk) \right|^2}. \end{aligned}$$

Dar această formă a densității spectrale este o generalizare a densității spectrale a unui proces AR. De fapt, dacă trunchiem suma astfel încât densitatea spectrală căutată devine

$$P_X(f) = \frac{S_U^2}{\left| 1 - \sum_{k=1}^{\infty} a[k] \exp(-2pifk) \right|^2}$$

atunci obținem densitatea spectrală a unui așa numit *proces AR de ordin p*, care se mai notează AR(p). În acest caz,

$$X[n] = \sum_{k=1}^p a[k]X[n-k] + U[n] \quad (4.5.15)$$

unde $U[n]$ reprezintă ca de obicei un zgomot alb Gaussian de varianță S_U^2 . În mod cert pentru $p=1$ obținem definiția anterioară dată pentru procese AR. Dacă considerăm un proces AR(p) astfel încât $a[l]=0$ pentru $l > p$ soluția pentru predictorul liniar de pas 1 optimal este din (4.5.14)

$$\hat{X}[n_0+1] = \sum_{l=0}^{p-1} a[l]X[n_0-l]$$

de unde punând $k = l + 1$ rezultă

$$\hat{X}[n_0 + 1] = \sum_{k=1}^p a[k]X[n_0 + 1 - k] \quad (4.5.16)$$

iar EMP minimă este S_U^2 .

Exemplul 4.9 (predictor liniar de pas 1 pentru proces MA)

Considerăm procesul staționar dat de $X[n] = U[n] - bU[n-1]$, unde $|b| < 1$ și $U[n]$ este zgomot alb Gaussian (wGn) de varinată S_U^2 (proces cunoscut și sub numele de *medie mobilă*). Acesta este un caz special al procesului MA din Exemplul 4.1. pentru care $h[0] = 1$ și $h[1] = -b$, iar $U[n]$ este wGn. Pentru a găsi predictorul liniar optim punem transformata z a funcției de corelație în forma cerută. Să determinăm întâi densitatea spectrală. Deoarece funcția de sistem este $H(z) = 1 - bz^{-1}$, funcția de transfer frecvențială este $H(f) = 1 - b \exp(2\pi if)$. Din (4.3.12) densitatea spectrală devine

$$P_X(f) = H(f)H^*(f)S_U^2 = (1 - \exp(-2\pi if))(1 - b \exp(2\pi if))S_U^2$$

și deci înlocuind $\exp(2\pi if)$ cu z , obținem

$$P_X(z) = (1 - bz^{-1})(1 - bz)S_U^2. \quad (4.5.17)$$

Aducând (4.5.17) la forma dorită pentru $P_X(z)$ dată în (4.5.13) avem că

$$A(z) = \frac{1}{1 - bz^{-1}}.$$

Pentru a converti aceasta în $1 - \sum_{k=1}^{\infty} a[k]z^{-k}$, considerăm transformata z inversă, pentru a obține

$$Z^{-1}\{A(z)\} = \begin{cases} b^k & k \geq 0 \\ 0 & k < 0 \end{cases}$$

deci $a[k] = -b^k$ pentru $k \geq 1$. (Observăm aici necesitatea de a lua $|b| < 1$; în caz contrar $a[n]$ nu ar mai fi stabil.) Predictorul optimal este din (4.5.14)

$$\begin{aligned} \hat{X}[n_0 + 1] &= \sum_{k=0}^{\infty} a[k+1]X[n_0 - k] \\ &= \sum_{k=0}^{\infty} (-b^{k+1})X[n_0 - k] \\ &= -bX[n_0] - b^2X[n_0 - 1] - \mathbf{L} \end{aligned}$$

iar EMP minimă este

$$\text{emp}_{\min} = S_U^2.$$

Ca un caz particular special vom mai considera predictorul liniar de pas 1 de lungime finită. Lungimea finită se referă la faptul că predicția depinde doar de eșantionul prezent și eșantioanele trecute până la $M-1$. Analog ca în cazul predictorului de lungime infinită se poate arăta că, dacă predictorul este dat de

$$\hat{X}[n_0+1] = \sum_{k=0}^{M-1} h[k]X[n_0-k]$$

care este (4.5.3) cu $h[k]=0$ pentru $k \geq M$, atunci răspunsul în impuls optim satisface cele M ecuații lineare simultane

$$r_x[l+1] = \sum_{k=0}^{M-1} h[k]r_x[l-k] \quad l = 0, 1, \mathbf{K}, M-1.$$

Scrise în formă matricială aceste ecuații devin

$$\begin{bmatrix} r_x[0] & r_x[1] & \mathbf{K} & r_x[M-1] \\ r_x[1] & r_x[0] & \mathbf{K} & r_x[M-2] \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ r_x[M-1] & r_x[M-2] & \mathbf{K} & r_x[0] \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \\ \mathbf{M} \\ h[M-1] \end{bmatrix} = \begin{bmatrix} r_x[1] \\ r_x[2] \\ \mathbf{M} \\ r_x[M] \end{bmatrix}. \quad (4.5.18)$$

\mathbf{R}_x

EMP minimă corespunzătoare este dată de

$$\text{emp}_{\min} = r_x[0] - \sum_{k=0}^{M-1} h_{\text{opt}}[k]r_x[k+1]. \quad (4.5.19)$$

Aceste ecuații se mai numesc ecuațiile *Wiener-Hopf*. În general, ele sunt rezolvate numeric și există numeroși algoritmi pentru aceasta (vezi [19]). Majoritatea acestor algoritmi se folosesc de faptul că matricea are o structură de matrice de (auto)corelație. Ca atare este simetrică, pozitiv definită și are proprietatea *Toeplitz*. Aceasta spune de fapt că elementele de-a lungul fiecărei diagonale principale sunt identice. O altă conexiune importantă între ecuațiile predicției liniare și un proces AR(p) este stabilită în cazul în care punem $M=p$ în (4.5.18). Atunci, deoarece pentru un proces AR(p) avem că $h[n]=a[n+1]$ pentru $n=0, 1, \mathbf{K}, p-1$ (să ne reamintim din (4.5.16) că $\hat{X}[n_0+1] = \sum_{k=1}^p a[k]X[n_0+1-k]$), ecuațiile Wiener-Hopf devin

$$\begin{bmatrix} r_x[0] & r_x[1] & \mathbf{K} & r_x[p-1] \\ r_x[1] & r_x[0] & \mathbf{K} & r_x[p-2] \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ r_x[p-1] & r_x[p-2] & \mathbf{K} & r_x[0] \end{bmatrix} \begin{bmatrix} a[0] \\ a[1] \\ \mathbf{M} \\ a[p-1] \end{bmatrix} = \begin{bmatrix} r_x[1] \\ r_x[2] \\ \mathbf{M} \\ r_x[p] \end{bmatrix}. \quad (4.5.20)$$

Să notăm un fapt important și anume că pentru un proces AR(p) predictorul liniar este bazat pe o infinitate de eșantioane trecute sau pe un număr finit de eșantioane trecute. Ecuațiile (4.5.20) mai sunt cunoscute ca ecuațiile *Yule-Walker*. În această formă acestea leagă funcția de (auto)corelație de parametrii filtrului AR. Dacă eșantioanele funcției de (auto)corelație sunt cunoscute, atunci parametrii filtrului AR pot fi obținuți

prin rezolvarea ecuațiilor. Pe de altă parte dacă parametrii filtrului au fost determinați din (4.5.20), varianța zgomotului alb $U[n]$ va fi dată de

$$s_U^2 = \text{emp}_{\min} = r_x[0] - \sum_{k=1}^p a[k]r_x[k], \quad (4.5.21)$$

formă care rezultă punând $h_{opt}[k] = a[k+1]$ cu $M = p$ în (4.5.19).

4.6. Exemplu concret – sintetizarea vocii

Este deja larg răspândit ca informațiile (telefonice sau GPS) să fie transmise electronic cu ajutorul unui convertor (sintetizator) text – voce. Una din primele aplicații în acest sens a fost dezvoltată de Texas Instruments prin Speak and Spell¹. În general sunetele vorbite se împart în două categorii: *vorbire vocală* – când rostim o vocală și *vorbire nevocală* – când rostim o consoană. Pentru sunetele vocale excitația se modelează cu ajutorul unui șir de impulsuri care produc un sunet periodic, iar sunetele nevocale se modelează prin transformarea unui zgomot alb într-un sunet de tip zgomot. Excitația este dată de modificarea tractului vocal uman, ce poate fi asimilată unui filtru liniar SI. Cunoașterea formei undei excitate și a funcției de sistem a tractului vocal ne permite să sintetizăm vorbirea. Pentru sunetele nevocale trecem un zgomot alb gaussian discret printr-un filtru liniar SI având funcția de sistem $H_{nv}(z)$. Vom trata în continuare sintetizarea semnalului nevocal, sintetizarea sunetelor vocale decurgând analog.

S-a stabilit că o bună modelare a tractului vocal este filtrul liniar SI având funcția de sistem

$$H_{nv}(z) = \frac{1}{1 - \sum_{k=1}^p a[k]z^{-k}}$$

care este un *filtru de poli*. Pentru considerente practice, *ordinul filtrului*, care este egal cu numărul de poli ai funcției complexe de mai sus, se alege ca fiind $p=12$. Output-ul $X[n]$ al acestui filtru, pentru un zgomot alb $U[n]$ cu varianța s_U^2 este procesul staționar

$$X[n] = \sum_{k=1}^p a[k]X[n-k] + U[n],$$

care reprezintă ecuațiile cu diferențe definatorii pentru un proces AR(p). Așadar vorbirea nevocală se poate sintetiza folosind aceste ecuații cu diferențe pentru o alegere convenabilă de parametri $\{a[1], a[2], \dots, a[p], s_U^2\}$. Parametrii vor fi diferiți pentru fiecare sunet nevocal în parte. Pentru a determina parametrii pentru un anumit sunet se folosește un segment din sunetul vorbit dorit pentru a estima funcția de (auto)corelație. Apoi parametrii $a[k]$ pentru $k=1, 2, \dots, p$ se pot obține rezolvând ecuațiile Yule-Walker. Funcția de corelație teoretică este înlocuită de funcția de (auto)corelație estimată pentru a obține din (4.5.20) sistemul

¹ Marcă înregistrată Texas Instruments

$$\begin{bmatrix} \hat{r}_x[0] & \hat{r}_x[1] & \mathbf{K} & \hat{r}_x[p-1] \\ \hat{r}_x[1] & \hat{r}_x[0] & \mathbf{K} & \hat{r}_x[p-2] \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ \hat{r}_x[p-1] & \hat{r}_x[p-2] & \mathbf{K} & \hat{r}_x[0] \end{bmatrix} \begin{bmatrix} a[0] \\ a[1] \\ \mathbf{M} \\ a[p-1] \end{bmatrix} = \begin{bmatrix} \hat{r}_x[1] \\ \hat{r}_x[2] \\ \mathbf{M} \\ \hat{r}_x[p] \end{bmatrix} \quad (4.5.22)$$

folosit pentru determinarea parametrilor $\hat{a}[k]$. Determinăm apoi varianța estimată a zgomotului alb din (4.5.21) ca fiind

$$\hat{S}_U^2 = \hat{r}_x[0] - \sum_{k=1}^p \hat{a}[k] \hat{r}_x[k] \quad (4.5.23)$$

unde $\hat{a}[k]$ sunt determinați ca soluția ecuațiilor Yule-Walker (4.5.22). Algoritmul descris conține o modificare adusă estimării funcției de (auto)corelație, care se face după formula

$$\hat{r}_x[k] = \frac{1}{N} \sum_{n=0}^{N-1-k} x[n]x[n+k] \quad k = 0, 1, \mathbf{K}, p \quad (4.5.24)$$

Diferită de abordarea clasică prin faptul că factorul normalizator este N în loc de $N-k$. Pentru $N \gg p$ aceasta are un efect neglijabil asupra estimării parametrilor, dar are avantajul că polii funcției de sistem estimate $\hat{H}_m(z)$ se află în interiorul cercului unitate. Această metodă de estimare se mai numește *metoda autocorelării predicției liniare*.

În mod normal pentru analiza sunetelor de vorbire pentru a estima parametrii AR, vom eșantiona la 8 kHz și vom folosi un bloc de date de lungime 20 msec (aprox. 160 de eșantioane).

Din datele experimentale s-a ajuns la concluzia că sunetul sintetizat este similar sunetului original dacă densitățile spectrale corespunzătoare sunt similare. Așadar densitatea spectrală estimată

$$\hat{P}_x(f) = \frac{\hat{S}_U^2}{\left| 1 - \sum_{k=1}^p \hat{a}[k] \exp(-2pifk) \right|^2} \quad (4.5.25)$$

ar corespunde cerințelor.

În final pentru a sintetiza sunetul calculăm

$$x[n] = \sum_{k=1}^p \hat{a}[k] x[n-k] + u[n]$$

unde $u[n]$ reprezintă un zgomot alb gaussian *pseudoaleator* cu varianța \hat{S}_U^2 pentru aprox. 20 msec.

Vom da în continuare codul MATLAB pentru aceste operații.

```
N = length(xseg); % xseg reprezinta esantionul de sunet
Nfft = 1024; %seteaza lungimea FFT pentru transformata Fourier
Freq=[0:Nfft-1]'/Nfft-0.5; %pct in care se repr densit spectr
P_per=(1/N)*abs(fftshift(fft(xseg,Nfft))).^2%calc periodograma
p = 12; %dimensiunea matricii de autocorelatie
```

```

for k=1:p+1 %estimarea fct de autocorelatie ptr p=0,1,...,p
    rX(k,1) = (1/N)*sum(xseg(1:N-k+1).*xseg(k:N));
end
r = rX(2:p+1) %completarea vectorului din dreapta
for i=1:p
    for j=1:p
        R(i,j) = rX(abs(i-j)+1);
    end
end
a = inv(R)*r; %rezolvarea ect lin ptr det param filtru AR
varu = rX(1)-a'*r; %gasirea variantei excitatiei de zgomot
den = abs(fftshift(fft([1;-a],Nfft))).^2; %calc numitor
                                %densit spectrala AR
P_AR = varu./den; %calc densit spectrala AR

```

4.7. Exerciții și probleme propuse spre rezolvare

1. Un sistem liniar SI cu funcția de sistem $H(z) = 1 - z^{-1} - z^{-2}$ este utilizat pentru a filtra un proces aleator zgomot alb cu parametru de timp discret de varianță $S_U^2 = 1$. Determinați funcția de (auto)corelație și densitatea spectrală a procesului rezultat după filtrare.
2. Un proces staționar cu parametru de timp discret de medie $m_U = 2$ este input-ul unui sistem liniar SI cu răspuns la impuls $h[n] = \left(\frac{1}{2}\right)^n$, pentru $n \geq 0$ și $h[n] = 0$, pentru $n < 0$. Găsiți media output-ului.
3. Un zgomot alb cu parametru de timp discret $U[n]$ reprezintă input-ul la un sistem liniar care produce output-ul dat de procesul $X[n] = a^{|n|}U[n]$ pentru $|a| < 1$. Să se determine densitatea spectrală a procesului de output.
4. Un proces slab staționar cu parametru de timp discret este definit prin ecuația cu diferențe $X[n] = aX[n-1] + U[n] - bU[n-1]$, unde $U[n]$ este un zgomot alb cu parametru de timp discret de varianță $S_U^2 = 1$. Reprezentați grafic densitatea spectrală a procesului $X[n]$ pentru $a = 0.9, b = 0.2$ și pentru $a = 0.2, b = 0.9$. Explicați rezultatele.
5. Un proces slab staționar cu parametru discret de timp este dat de ecuația cu diferențe $X[n] = 0.5X[n-1] + U[n] - 0.5U[n-1]$, unde $U[n]$ este un zgomot alb discret de varianță $S_U^2 = 1$. Găsiți funcția de (auto)corelație și densitatea spectrală a procesului $X[n]$.
6. Un proces aleator cu densitatea spectrală dată de

$$P_x(f) = \frac{1}{\left|1 - \frac{1}{2}\exp(-2\pi if)\right|^2}$$

este filtrat printr-un sistem liniar SI pentru a produce un zgomot alb $U[n]$ de varianță $S_U^2 = 4$. Care este exprimarea sistemului ca ecuație cu diferențe?

7. Un proces AR de ordinul 2 este dat de ecuația cu diferențe recursivă $X[n] = 2r \cos(2p f_0) X[n-1] - r^2 X[n-2] + U[n]$, unde $U[n]$ este zgomot alb discret de varianță $S_U^2 = 1$. Pentru $r = 0.7, f_0 = 0.1$ și pentru $r = 0.95, f_0 = 0.1$ reprezentați grafic densitatea spectrală a lui $X[n]$. Indicație: determinați locațiile polilor lui $H(z)$.
8. Un semnal de medie nulă cu densitatea spectrală $P_S(f) = 2 - 2 \cos(2p f)$ este încastrat într-un zgomot alb de varianță $S_W^2 = 1$. Reprezentați grafic funcția de transfer frecvențială a netezitorului Wiener optimal. Determinați de asemenea și EMP_{\min} . Indicație: pentru determinarea EMP folosiți o aproximare „în sumă” a integralei.
9. Această problemă este menită să simuleze un netezitor Wiener. Mai întâi generați $N = 50$ eșantioane ale unui semnal $S[n]$, care este un proces AR ($U[n]$ este presupus zgomot alb) cu $a = 0.25$ și $S_U^2 = 0.5$. A nu se uita de condiția inițială $S[-1] \sim N\left(0, \frac{S_U^2}{1-a^2}\right)$. În continuare se adună zgomotul alb $W[n]$ de varianță $S_W^2 = 1$ realizării procesului AR. Se aplică apoi codul MATLAB pentru netezirea semnalului distorsionat. Reprezentați grafic semnalul și semnalul netezit.
10. Considerăm un proces AR(2) dat de $X[n] = -r^2 X[n-2] + U[n]$, unde $U[n]$ este un zgomot alb de varianță S_U^2 și $0 < r < 1$. Acest proces se deduce din (4.5.15) cu $p = 2$ și $a[1] = 0, a[2] = -r^2$. Funcția de (auto)corelație a acestui proces se poate arăta că este dată de $r_x[k] = \left(\frac{S_U^2}{1-r^4}\right) r^{|k|} \cos\left(\frac{kp}{2}\right)$ (vezi [19]). Găsiți predictorul liniar optimal de pas 1 în baza eșantioanelor prezente și trecute ale lui $X[n]$. Simulați predictorul pentru cazurile $r = 0.5, S_U^2 = 1 - r^4$ și $r = 0.95, S_U^2 = 1 - r^4$ astfel ca puterea medie în fiecare caz să fie aceeași ($r_x[0] = 1$). Generați 150 de eșantioane și eliminați primele 100 pentru asigurarea staționarității. Reprezentați grafic valorile prezise pentru fiecare caz. Care din valorile lui r dă un proces mai predictibil?
11. Pentru $M=1$ rezolvați ecuațiile Wiener-Hopf date în (4.5.18) pentru a determina $h[0]$.
12. Procesul MA descris în Exemplul 4.7. și dat de $X[n] = U[n] - bU[n-1]$ are funcția de (auto)corelație pentru $S_U^2 = 1$
- $$r_x[k] = \begin{cases} 1+b^2 & k=0 \\ -b & k=1. \\ 0 & k \geq 2 \end{cases}$$
- Pentru $M=2$ rezolvați ecuațiile Wiener-Hopf pentru a găsi predictorul liniar de pas finit și determinați apoi EMP minimă.
13. Dorim să prezicem un zgomot alb. Pentru aceasta să se rezolve ecuațiile Wiener-Hopf pentru $r_x[k] = S_X^2 d[k]$.

14. Pentru un proces MA $X[n] = U[n] - \frac{1}{2}U[n-1]$, unde $U[n]$ este un zgomot alb de varianță $\sigma_U^2 = 1$, să se găsească predictorul de lungime finită optimal $\hat{X}[n_0 + 1] = h[0]X[n_0] + h[1]X[n_0 - 1]$ și EMP minimă.
15. Un filtru cu funcție de transfer frecvențială $H(F) = \exp(2\pi i F t_0)$ se folosește pentru a filtra un proces slab staționar cu densitatea spectrală $P_X(F)$. Care este densitatea spectrală la ieșirea din filtru și de ce?

Notă. Prezentul capitol este redactat după Steven Kay, *Intuitive Probability and Random Processes using MATLAB*, Springer, New York, 2006, cap. 18.

Capitolul 5

Probleme de mecanică rezolvate în MATLAB

În acest capitol vom prezenta câteva probleme de mecanică abordate în MATLAB, ca o exemplificare a utilității acestui soft matematic pentru rezolvarea aproximativă a ecuațiilor algebrice, a problemelor Cauchy pentru ecuații diferențiale, pentru reprezentări grafice, pentru simularea unor fenomene mecanice, etc. În fiecare secțiune am prezentat câteva probleme, pentru fiecare problemă fiind dat enunțul, abordarea din punct teoretic și tratarea problemei în MATLAB.

5.1. Calcul vectorial

Problema 5.1 (Operații cu vectori): *Să se traducă operațiile cu vectori fizici în operații cu vectori / matrice.*

Soluție:

Din punct de vedere teoretic, vectorului

$$\mathbf{x} = x^1 \mathbf{u}_1 + x^2 \mathbf{u}_2 + x^3 \mathbf{u}_3,$$

reprezentat într-o bază ortonormată $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ i se asociază, în MATLAB, vectorul linie (matrice 1×3)

$$\mathbf{x} = [x(1) \ x(2) \ x(3)].$$

Pentru simplificare, în continuare, vom nota componentele x_1, x_2, x_3 , adică, vectorul \mathbf{x} este $\mathbf{x} = [x_1 \ x_2 \ x_3]$. Operațiile de adunare a vectorilor, înmulțire a unui vector cu un scalar și scădere a vectorilor se traduc prin operațiile corespunzătoare cu matrice.

Evaluarea *produsului scalar a doi vectori* se poate realiza:

- prin utilizarea expresiei analitice:

$$x_1 y_1 + x_2 y_2 + x_3 y_3;$$

- prin operații cu matrice:

$$\mathbf{x} * \mathbf{y}' = [x_1 \ x_2 \ x_3] \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{sum}(\text{diag}(\mathbf{x}' * \mathbf{y})) = \text{trace}(\mathbf{x}' * \mathbf{y})$$

- prin utilizarea funcției `dot` din MATLAB:

$$\text{dot}(\mathbf{x}, \mathbf{y})$$

Evaluarea *produsului vectorial a doi vectori* se poate realiza:

- prin utilizarea expresiei analitice:

$$[x_2 y_3 - x_3 y_2, \ x_3 y_1 - x_1 y_3, \ x_1 y_2 - x_2 y_1]$$

- prin operații cu matrice:

- prima variantă: se construiesc matricele

$$\mathbf{m} = \mathbf{x}' * \mathbf{y} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & x_1 y_3 \\ x_2 y_1 & x_2 y_2 & x_2 y_3 \\ x_3 y_1 & x_3 y_2 & x_3 y_3 \end{bmatrix}, \quad \mathbf{q} = \mathbf{m} - \mathbf{m}' ,$$

atunci produsul vectorial este vectorul

$$[q_{23} + q_{31} + q_{12}];$$

- a doua variantă: se consideră vectorii

$$u = [1 \ 0 \ 0], \quad v = [0 \ 1 \ 0], \quad w = [0 \ 0 \ 1]$$

atunci componentele produsului vectorial sunt

$$\det([u; x; y]), \det([v; x; y]), \det([w; x; y]),$$

- prin utilizarea funcției `cross` din MATLAB:

$$\text{cross}(x, y)$$

Evaluarea *produsului mixt a trei vectori* se face prin utilizarea funcției `det` din:
 $\det([x; y; z])$.

Ca exemplificare de programare și calcul în MATLAB pentru problema enunțată se poate considera programul următor:

```
x = [1 2 3], y = [4 5 3]
ps1 = x(1)*y(1)+x(2)*y(2)+x(3)*y(3), ps2 = x*y'
ps3 = sum(diag(x'*y)), ps4 = dot(x,y)
m = x'*y, q = m-m', pv1 = [q(2,3) q(3,1) q(1,2)]
u = [1 0 0], v = [0 1 0], w = [0 0 1]
C1 = [u; x; y], C2 = [v; x; y], C3 = [w; x; y]
pv2 = [det(C1), det(C2), det(C3)]
pv3 = cross(x,y), z = [1 1 1]
pm = det([x; y; z])
```

Problema 5.2 (Coliniaritate): *Cunoscându-se vectorii de poziție ai punctelor A, B, C să se verifice dacă cele trei puncte sunt coliniare, să se calculeze aria triunghiului ABC și să se determine al patrulea vârf al paralelogramului ABCD.*

Soluție:

Necoliniaritatea punctelor A, B și C se poate verifica arătând că cel puțin două din rapoartele

$$\frac{x_B^1 - x_A^1}{x_C^1 - x_A^1}, \quad \frac{x_B^2 - x_A^2}{x_C^2 - x_A^2}, \quad \frac{x_B^3 - x_A^3}{x_C^3 - x_A^3}$$

sunt diferite (B nu se află pe dreapta determinată de A și C). De exemplu, considerând $\mathbf{x}_A = 2\mathbf{u}_1 - 3\mathbf{u}_2 + 4\mathbf{u}_3$, $\mathbf{x}_B = 3\mathbf{u}_1 + 2\mathbf{u}_2 - \mathbf{u}_3$, $\mathbf{x}_C = \mathbf{u}_2 - 2\mathbf{u}_3$, se obține

$$\frac{x_B^1 - x_A^1}{x_C^1 - x_A^1} = -\frac{1}{2}, \quad \frac{x_B^2 - x_A^2}{x_C^2 - x_A^2} = \frac{5}{4}, \quad \frac{x_B^3 - x_A^3}{x_C^3 - x_A^3} = \frac{5}{6}.$$

Necoliniaritatea punctelor A, B și C rezultă și din faptul că produsul vectorial $\vec{AB} \times \vec{AC}$ este nenul.

Același produs vectorial este utilizat și pentru calculul ariei triunghiului ABC

$$\begin{aligned} \text{aria}(\Delta ABC) &= \frac{1}{2} \left\| \vec{AB} \times \vec{AC} \right\| = \frac{1}{2} \left\| \begin{matrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \\ 1 & 5 & -5 \\ -2 & 4 & -6 \end{matrix} \right\| = \\ &= \frac{1}{2} \sqrt{10^2 + 16^2 + 14^2} = \sqrt{138} \approx 11.747340 \end{aligned}$$

Condiția ca $ABCD$ să fie paralelogram este $\vec{AB} = \vec{DC}$ sau $\mathbf{x}_B - \mathbf{x}_A = \mathbf{x}_C - \mathbf{x}_D$.
Deci, vectorul de poziție al celui de al 4-lea vârf al paralelogramului $ABCD$ este

$$\mathbf{x}_D = \mathbf{x}_A - \mathbf{x}_B + \mathbf{x}_C = -\mathbf{u}_1 - 4\mathbf{u}_2 + 3\mathbf{u}_3.$$

Programul MATLAB pentru problema 5.2 este următorul:

```
a = [2 -3 4]; b = [3 2 -1]; c = [0 1 -2];
ab = b-a, ac = c-a
k1 = ab(1)/ac(1), k2 = ab(2)/ac(2), k3 = ab(3)/ac(3)
pv = cross(ab,ac)
if pv==0
    disp('coliniare')
else
    disp('necoliniare');
end
d = c-b+a
M = [a;b;c;d;a];
x = M(:,1); y = M(:,2); z = M(:,3);
plot3(x,y,z,'r','LineWidth',2)
aria_abc = sqrt(pv*pv')/2
```

Problema 5.3 (Coplanaritate): Se dau vârfurile unui patrulater $ABCD$. Să se arate că acest patrulater este plan, chiar paralelogram, și să i se calculeze aria.

Soluție:

Considerăm că vectorii de poziție ai vârfurilor patrulaterului sunt

$$\mathbf{x}_A = 3\mathbf{u}_1 + \mathbf{u}_2 + 5\mathbf{u}_3, \quad \mathbf{x}_B = 2\mathbf{u}_1 - \mathbf{u}_2 + 2\mathbf{u}_3,$$

$$\mathbf{x}_C = 5\mathbf{u}_1 + 3\mathbf{u}_2 + 7\mathbf{u}_3, \quad \mathbf{x}_D = 6\mathbf{u}_1 + 5\mathbf{u}_2 + 10\mathbf{u}_3.$$

Coplanaritatea punctelor A, B, C, D rezultă din faptul că A se află în planul determinat de B, C, D , deoarece

$$\begin{vmatrix} x_A^1 & x_A^2 & x_A^3 & 1 \\ x_B^1 & x_B^2 & x_B^3 & 1 \\ x_C^1 & x_C^2 & x_C^3 & 1 \\ x_D^1 & x_D^2 & x_D^3 & 1 \end{vmatrix} = - \begin{vmatrix} x_B^1 - x_A^1 & x_B^2 - x_A^2 & x_B^3 - x_A^3 \\ x_C^1 - x_A^1 & x_C^2 - x_A^2 & x_C^3 - x_A^3 \\ x_D^1 - x_A^1 & x_D^2 - x_A^2 & x_D^3 - x_A^3 \end{vmatrix} = - \begin{vmatrix} -1 & -2 & -3 \\ 2 & 2 & 2 \\ 3 & 4 & 5 \end{vmatrix} = 0.$$

Se observă că

$$\mathbf{x}_D = \mathbf{x}_A - \mathbf{x}_B + \mathbf{x}_C,$$

deci $ABCD$ este un paralelogram, adică un patrulater plan.

Aria paralelogramului se calculează cu ajutorul produsului vectorial

$$\begin{aligned} \text{aria}(ABCD) &= \left\| \vec{AB} \times \vec{AC} \right\| = \left\| \begin{vmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \\ -1 & -2 & -3 \\ 3 & 4 & 5 \end{vmatrix} \right\| = \\ &= \left\| 2\mathbf{u}_1 - 4\mathbf{u}_2 + 2\mathbf{u}_3 \right\| = \sqrt{2^2 + 4^2 + 2^2} = 2\sqrt{6} \approx 4.8989794 \end{aligned}$$

Cu MATLAB problema se poate analiza cu următorul program:

```

a = [3 1 5], b = [2 -1 2], c = [5 3 7], d = [6 5 10]
cond1 = det([d 1; a 1; b 1; c 1])
if cond1==0
    disp('coplanare')
else
    disp('necoplanare')
end
cond2 = a-b+c-d
if cond2==0
    disp('coplanare: ABCD este paralelogram')
else
    disp('necoplanare')
end
M = [a;b;c;d;a];
x = M(:,1);y = M(:,2);z = M(:,3);
plot3(x,y,z,'r','LineWidth',2)
pv = cross(b-a,d-a);
aria_ABCD_1 = sqrt(pv*pv')
aria_ABCD_2 = norm(pv)

```

5.2. Probleme de cinematică

Problema 5.4 (Problema întâlnirii a două corpuri): Presupunând că un mobil M_1 se deplasează rectiliniu și uniform cu viteza constantă \mathbf{v}_1 , $\|\mathbf{v}_1\| = v_1 = \text{const.}$, să se determine traiectoria plană a unui mobil M_2 , ce se deplasează uniform cu viteza (constantă) \mathbf{v}_2 , $\|\mathbf{v}_2\| = v_2 = \text{const.}$, orientată mereu spre M_1 .

Soluție:

Fie planul determinat de traiectoria rectilinie a mobilului M_1 și poziția inițială a lui M_2 și fie în acest plan un reper cu originea O în poziția inițială a mobilului M_1 și axa Ox^1 dreapta pe care se mișcă M_1 . Dacă notăm $\overrightarrow{M_1M_2} = x\mathbf{u}_1 + y\mathbf{u}_2$, atunci poziția mobilelor va fi caracterizată prin

$$x_1^1 = v_1 t, x_1^2 = 0, \quad (5.1)$$

$$x_2^1 = v_1 t + x, x_2^2 = y > 0.$$

Cum viteza lui M_2 este $\mathbf{v}_2 = -v_2 \frac{\overrightarrow{M_1M_2}}{\|\overrightarrow{M_1M_2}\|}$ atunci pe componente avem

$$v_1 + \dot{x} = -\frac{v_2 x}{\sqrt{x^2 + y^2}}, \quad \dot{y} = -\frac{v_2 y}{\sqrt{x^2 + y^2}}. \quad (5.2)$$

Cu substituția

$$x = uy, \quad (5.3)$$

sistemul (5.2) devine

$$\dot{x} = -\frac{v_2}{\sqrt{1+u^2}}, \quad \dot{y} = -v_1. \quad (5.4)$$

Se obține ecuația diferențială

$$\frac{dy}{du} = \frac{v_2}{v_1} \frac{y}{\sqrt{1+u^2}}, \quad (5.5)$$

care admite soluția

$$u = \frac{1}{2} \left[(cy)^{v_1/v_2} - (cy)^{-v_1/v_2} \right] \quad (5.6)$$

unde $c = \frac{\left(u_o + \sqrt{1+u_o^2} \right)^{v_2/v_1}}{y_o}$ și $u_o = \frac{x_o}{y_o}$.

a) Presupunând $v_1 \neq v_2$, din $\frac{dy}{dt} = -\frac{2v_2}{(cy)^{v_1/v_2} + (cy)^{-v_1/v_2}}$ se deduce

$$t(y) = t_* - \frac{1}{2c} \left[\frac{(cy)^{1+v_1/v_2}}{v_1+v_2} - \frac{(cy)^{1-v_1/v_2}}{v_1-v_2} \right] \quad (5.7)$$

unde $t_* = \frac{1}{2c} \left[\frac{(cy_o)^{1+v_1/v_2}}{v_1+v_2} - \frac{(cy_o)^{1-v_1/v_2}}{v_1-v_2} \right]$.

Cum $x(y) = \frac{y}{2} \left[(cy)^{v_1/v_2} - (cy)^{-v_1/v_2} \right]$ se estimează limitele

$$\lim_{y \rightarrow 0} x(y) = 0, \quad \lim_{y \rightarrow 0} t(y) = t_* \quad \text{pentru } v_1 < v_2, \quad (5.8)$$

$$\lim_{y \rightarrow 0} x(y) = -\infty, \quad \lim_{y \rightarrow 0} t(y) = +\infty \quad \text{pentru } v_1 > v_2,$$

adică mobilele se întâlnesc la momentul $t = t_*$ dacă $v_1 < v_2$ și nu se întâlnesc dacă $v_1 > v_2$. În acest din urmă caz, distanța minimă dintre mobile

$$\frac{1}{c} \frac{v_1}{\sqrt{v_1^2 - v_2^2}} \left(\frac{v_1 - v_2}{v_1 + v_2} \right)^{v_2/v_1}$$

este realizată la momentul $t = t_*$.

b) Dacă $v_1 = v_2$, din $\frac{dy}{dt} = -\frac{2v_2}{cy + (cy)^{-1}}$ se obține

$$t = t_* - \frac{(cy)^2 + 2 \ln(cy)}{4cv_1}, \quad \text{unde } t_* = \frac{(cy_o)^2 + 2 \ln(cy_o)}{4cv_1} \quad (5.9)$$

și din $x(y) = \frac{y}{2} \left(cy - \frac{1}{cy} \right)$ rezultă

$$\lim_{y \rightarrow 0} x(y) = -\frac{1}{2c}, \quad \lim_{y \rightarrow 0} t(y) = -\infty, \quad (5.10)$$

adică "întâlnirea" are loc asimptotic în trecut.

În programul MATLAB pentru rezolvarea acestei probleme se utilizează funcțiile de calcul simbolic `dsolve`, `simplify`, `limit`, `subs` pentru integrarea ecuațiilor de mișcare cu condiții inițiale date (obținerea soluției analitice), simplificarea expresiei pentru soluție și calcularea limitelor și a celorlalte mărimi caracteristice. De asemenea, se fac reprezentări grafice în plan utilizând funcțiile `plot`, `subs` și `ezplot` pentru reprezentarea grafică a traiectoriei pentru diferite valori ale raportului $k = v_2 / v_1$ și ale poziției inițiale y_0 .

Pentru a nu face apel la date numerice dimensionale, se introduc variabilele adimensionale

$$X = \frac{x}{y_0}, \quad Y = \frac{y}{y_0}, \quad T = \frac{v_1 t}{y_0} \quad (5.11)$$

și notațiile

$$k = v_2/v_1, \quad C = cy_0 = \left(u_0 + \sqrt{1+u_0^2}\right)^k, \quad T_* = \frac{1}{2} \left[\frac{C^{1/k}}{1+k} - \frac{C^{-1/k}}{1-k} \right].$$

Soluțiile (5.6) și (5.7) se scriu

$$u = \frac{1}{2} \left[(CY)^{1/k} - (CY)^{-1/k} \right] = \sinh \left(\frac{\ln(CY)}{k} \right), \quad (5.12)$$

$$T = T_* - \frac{1}{2} \left[\frac{(CY)^{1/k}}{1+k} - \frac{(CY)^{-1/k}}{1-k} \right].$$

Integrând ecuațiile de mișcare (5.2) după schimbarea de variabilă (5.3):

```
syms u0
u = dsolve('Du=sqrt(1+u^2)/y/k', 'u(y0)=u0', 'y'),
simplify(u)
```

se obține

```
u = [sinh((log(y)+log(u0-(u0^2+1)^(1/2))*k-log(y0))/k)]
     [sinh((log(y)+log(u0+(u0^2+1)^(1/2))*k-log(y0))/k)]
ans = [sinh((log(y)+log(u0-(u0^2+1)^(1/2))*k-log(y0))/k)]
      [sinh((log(y)+log(u0+(u0^2+1)^(1/2))*k-log(y0))/k)]
```

Observăm că sistemul de ecuații diferențiale (5.2) nu a putut fi integrat simbolic de MATLAB, dar în același timp dă soluția ecuației (5.5) în complex, prima parte a soluției neavând sens.

Programul care determină traiectoriile pornind din aceeași poziție inițială pentru problema întâlnirii, este următorul:

```
x0 = 1; y0 = 1; u0 = y0/x0;
y = 0:0.0001:y0;
k1 = 1.25;
c1 = (u0+sqrt(1+u0*u0))^k1;
lt1 = (c1^(1+1/k1)/(1+k1)-c1^(1-1/k1)/(1-k1))/2/c1
t1 = lt1-((c1*y).^(1+1/k1)/(1+k1)-...
         (c1*y).^(1-1/k1)/(1-k1))/2/c1;
u1 = ((c1*y).^(1/k1)-(c1*y).^(-1/k1))/2;
x1 = t1+y.*u1;
k2 = 1.5;
```

```

c2 = (u0+sqrt(1+u0*u0))^k2;
lt2 = (c2^(1+1/k2)/(1+k2)-c2^(1-1/k2)/(1-k2))/2/c2
t2 = lt2-((c2*y).^(1+1/k2)/(1+k2)-...
      (c2*y).^(1-1/k2)/(1-k2))/2/c2;
u2 = ((c2*y).^(1/k2)-(c2*y).^(-1/k2))/2;
x2 = t2+y.*u2;
k3 = 1.75;
c3=(u0+sqrt(1+u0*u0))^k3;
lt3 = (c3^(1+1/k3)/(1+k3)-c3^(1-1/k3)/(1-k3))/2/c3
t3 = lt3-((c3*y).^(1+1/k3)/(1+k3)-...
      (c3*y).^(1-1/k3)/(1-k3))/2/c3;
u3 = ((c3*y).^(1/k3)-(c3*y).^(-1/k3))/2;
x3 = t3+y.*u3;
plot(x1,y,'r-',x2,y,'g-.',x3,y,'b--','LineWidth',2);
axis equal;
Xlabel('x^1/y_0','FontSize',14);
Ylabel('x^2/y_0','FontSize',14);
legend('k_1=1.25','k_1=1.50','k_1=1.75')

```

iar rezultatele grafice obținute din acest program sunt prezentate în figura 5.1.

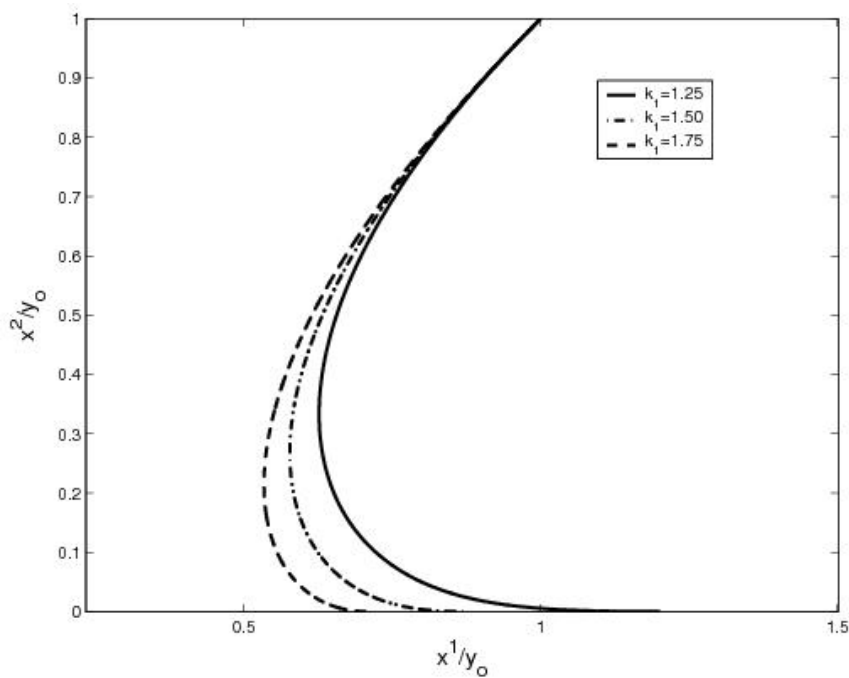


Figura 5.1: Traiectoriile pentru diferite raporturi ale vitezelor în problema întâlnirii a două corpuri.

Programul care determină traiectoriile pornind din poziții inițiale diferite pentru problema întâlnirii, este următorul:

```

k = 1.5; y0 = 1;
y = 0:0.0001:y0;
x01 = 1; u01 = x01/y0;
c1 = (u01+sqrt(1+u01*u01))^k/y0;
lt1 = (c1^(1+1/k)/(1+k)-c1^(1-1/k)/(1-k))/2/c1

```

```

t1 = lt1-((c1*y).^ (1+1/k)/(1+k)-(c1*y).^ (1-1/k)/(1-k))/2/c1;
u1 = ((c1*y).^ (1/k)-(c1*y).^ (-1/k))/2;
x1 = t1+y.*u1;
x02 = 1.5; u02 = x02/y0;
c2 = (u02+sqrt(1+u02*u02))^k/y0;
lt2 = (c2^(1+1/k)/(1+k)-c2^(1-1/k)/(1-k))/2/c2
t2 = lt2-((c2*y).^ (1+1/k)/(1+k)-(c2*y).^ (1-1/k)/(1-k))/2/c2;
u2 = ((c2*y).^ (1/k)-(c2*y).^ (-1/k))/2;
x2 = t2+y.*u2;
x03 = 2; u03 = x03/y0;
c3 = (u03+sqrt(1+u03*u03))^k/y0;
lt3 = (c3^(1+1/k)/(1+k)-c3^(1-1/k)/(1-k))/2/c3
t3 = lt3-((c3*y).^ (1+1/k)/(1+k)-(c3*y).^ (1-1/k)/(1-k))/2/c3;
u3 = ((c3*y).^ (1/k)-(c3*y).^ (-1/k))/2;
x3 = t3+y.*u3;
plot(x1,y,'r-',x2,y,'g-.',x3,y,'b--','LineWidth',2);
axis equal;
Xlabel('x^1/y_o','FontSize',14);
Ylabel('x^2/y_o','FontSize',14);
legend(['v_1/y_o*t_*_1=',num2str(lt1)],['v_1/y_o*t_*_2=',...
num2str(lt2)],['v_1/y_o*t_*_3=',num2str(lt3)])

```

Rezultatele obținute cu acest program pentru $k = 1.5$ sunt prezentate în figura 5.2.

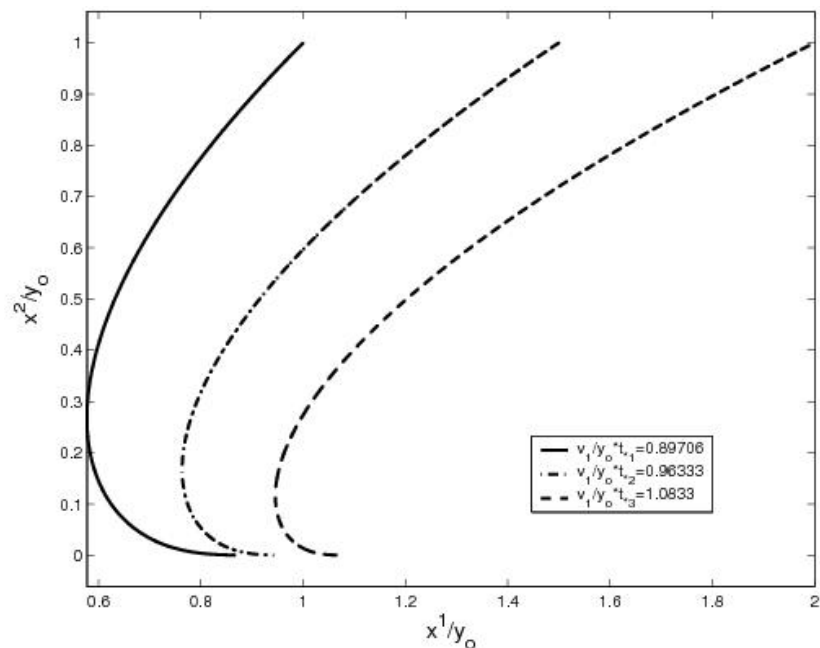


Figura 5.2: Traiectoriile pentru diferite poziții inițiale în problema întâlnirii.

Programul următor este un program de animație pentru problema întâlnirii.

```

x0 = 2; y0 = 1; u0 = x0/y0; k = 1.5;
c = (u0+sqrt(1+u0*u0))^k;
lt = (c^(1+1/k)/(1+k)-c^(1-1/k)/(1-k))/2/c;
y1 = 0.01:0.01:y0;
t1 = lt-((c*y1).^ (1+1/k)/(1+k)-(c*y1).^ (1-1/k)/(1-k))/2/c;
u1 = ((c*y1).^ (1/k)-(c*y1).^ (-1/k))/2; x1 = t1+y1.*u1;

```

```

ncadre = 15;
syms yy
for j = 1:ncadre
    timp=lt-((c*yy).^(1+1/k)/(1+k)-(c*yy).^(1-1/k)/(1-k))/2/c
    t(j)=(j-1)*lt/ncadre;
    yj = solve(t(j)-timp);
    for i = 1:length(yj)
        if (imag(yj(i))==0)
            y(j) = eval(yj(i))
        end
    end
end
end
M = moviein(ncadre)
for j=1:ncadre-1
    u(j) = ((c*Y(j)).^(1/k)-(c*Y(j)).^(-1/k))/2;
    x(j) = t(j)+y(j)*u(j);
    xtg = [t(j) x(j)]; ytg=[0 y(j)];
    plot(xl,y1,'-k',x(j),y(j),'*r',t(j),0,'*b',xtg,ytg,'.k')
    axis equal;
    axis([0,2.1*lt ,0,1.1*y0]);
    M(j) = getframe;
end
movie(M)

```

Problema 5.5 (Mișcarea uniformă pe o elice circulară): *Un punct material M se mișcă uniform cu viteza v_0 (valoare algebrică) pe o elice circulară. Cunoscând raza r a cilindrului și panta α a elicei, să se studieze mișcarea punctului în coordonate carteziene, cilindrice și intrinseci.*

Soluție: Ecuațiile parametrice ale elicei sunt

$$\begin{cases} x^1 = r \cos q, \\ x^2 = r \sin q, \\ x^3 = r q \operatorname{tg} \alpha. \end{cases}$$

Componentele vitezei în coordonate carteziene

$$\dot{x}^1 = -r\dot{q} \sin q, \quad \dot{x}^2 = r\dot{q} \cos q, \quad \dot{x}^3 = r\dot{q} \operatorname{tg} \alpha.$$

Din condiția de mișcare uniformă rezultă

$$v_0^2 = r^2 \dot{q}^2 (1 + \operatorname{tg}^2 \alpha) \Rightarrow \dot{q} = \dot{q}_0 = \frac{v_0}{r} \cos \alpha = \operatorname{const.} \Rightarrow q = q_0 + \frac{v_0}{r} t \cos \alpha.$$

Componentele accelerației în coordonate carteziene vor fi

$$\ddot{x}^1 = -r\dot{q}^2 \cos q, \quad \ddot{x}^2 = -r\dot{q}^2 \sin q, \quad \ddot{x}^3 = 0,$$

deci accelerația este perpendiculară pe axa Ox^3

$$\mathbf{a} = -r\dot{q}^2 (\cos q \mathbf{u}_1 + \sin q \mathbf{u}_2).$$

În coordonate cilindrice, având în vedere rezultatul din [11] (problema 2.5.2), se obține

$$v_r = 0, v_q = r\dot{q}, v_z = r\dot{q} \operatorname{tg} \alpha; \mathbf{v} = r\dot{q} \mathbf{u}_q + r\dot{q} \operatorname{tg} \alpha \mathbf{u}_z,$$

$$v_0^2 = r^2 \dot{q}^2 (1 + \operatorname{tg}^2 \alpha) \Rightarrow \dot{q} = \dot{q}_0 = \frac{v_0}{r} \cos \alpha = \operatorname{const.},$$

$$a_r = -r\dot{q}^2, a_q = 0, a_z = 0; \mathbf{a} = -r\dot{q}^2 \mathbf{u}_r.$$

Triedrul lui Frenet asociat elicei este definit prin

$$\left(\frac{ds}{dq}\right)^2 = \left(\frac{dx^1}{dq}\right)^2 + \left(\frac{dx^2}{dq}\right)^2 + \left(\frac{dx^3}{dq}\right)^2 = \frac{r^2}{\cos^2 a} \Rightarrow \frac{ds}{dq} = e \frac{r}{\cos a},$$

$$\boldsymbol{\tau} = \frac{d\mathbf{x}}{ds} = \frac{dq}{ds} \frac{d\mathbf{x}}{dq} = e \cos a (-\sin q \mathbf{u}_1 + \cos q \mathbf{u}_2 + \operatorname{tg} a \mathbf{u}_3),$$

$$\frac{d\boldsymbol{\tau}}{ds} = \frac{\mathbf{v}}{r} \Rightarrow r = \frac{r}{\cos^2 a} > r, \mathbf{v} = -(\cos q \mathbf{u}_1 + \sin q \mathbf{u}_2),$$

$$\boldsymbol{\beta} = \boldsymbol{\tau} \times \mathbf{v} = -e \sin a (-\sin q \mathbf{u}_1 + \cos q \mathbf{u}_2 - \operatorname{ctg} a \mathbf{u}_3).$$

Din condiția de mișcare uniformă se obține

$$\dot{q} = v_o = \text{const.},$$

iar din formulele

$$\mathbf{v} = \frac{d\mathbf{x}}{dt} = \frac{d\mathbf{x}}{ds} \frac{ds}{dt} = \dot{q} \boldsymbol{\tau},$$

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = \dot{q} \frac{d\boldsymbol{\tau}}{dt} = \dot{q} \frac{d\boldsymbol{\tau}}{ds} \frac{ds}{dt} = \dot{q}^2 \frac{d\boldsymbol{\tau}}{ds} = \frac{(\dot{q})^2}{r} \boldsymbol{\beta},$$

rezultă

$$\mathbf{v} = v_o \boldsymbol{\tau}, \mathbf{a} = \frac{v_o^2}{r} \boldsymbol{\beta}.$$

Cu ajutorul programului următor este reprezentată elicea și triedrul Frenet în diferite puncte ale elicei. Reprezentarea grafică este dată în figura 5.3.

```

r = 4; a = 0.5; q = 0:pi/72:4*pi;
ex = r*cos(q); ey = r*sin(q); ez = a*q;
plot3(ex,ey,ez, '-k', 'LineWidth', 2); hold on
u = 0:pi/4:4*pi;
x = r*cos(u); y = r*sin(u); z = a*u;
tx = -r*sin(u)/sqrt(r^2+a^2);
ty = r*cos(u)/sqrt(r^2+a^2);
tz = a/sqrt(r^2+a^2);
quiver3(x,y,z,tx,ty,tz, .5, 'r')
nx = -cos(u); ny = -sin(u); nz = 0;
quiver3(x,y,z,nx,ny,nz, .5, 'g')
bx = a*sin(u)/sqrt(r^2+a^2);
by = -a*cos(u)/sqrt(r^2+a^2);
bz = r/sqrt(r^2+a^2);
quiver3(x,y,z,bx,by,bz, 0.5, 'b')
axis equal;
view([120,20])
xlabel('x^1', 'FontSize', 14);
ylabel('x^2', 'FontSize', 14);
zlabel('x^3', 'FontSize', 14);
text(-1,-1,7, ['r=', num2str(r), ', a=', num2str(a)], 'FontSize', 14);
title('Triedru Frenet asociat elicei
circularare', 'FontSize', 14);

```

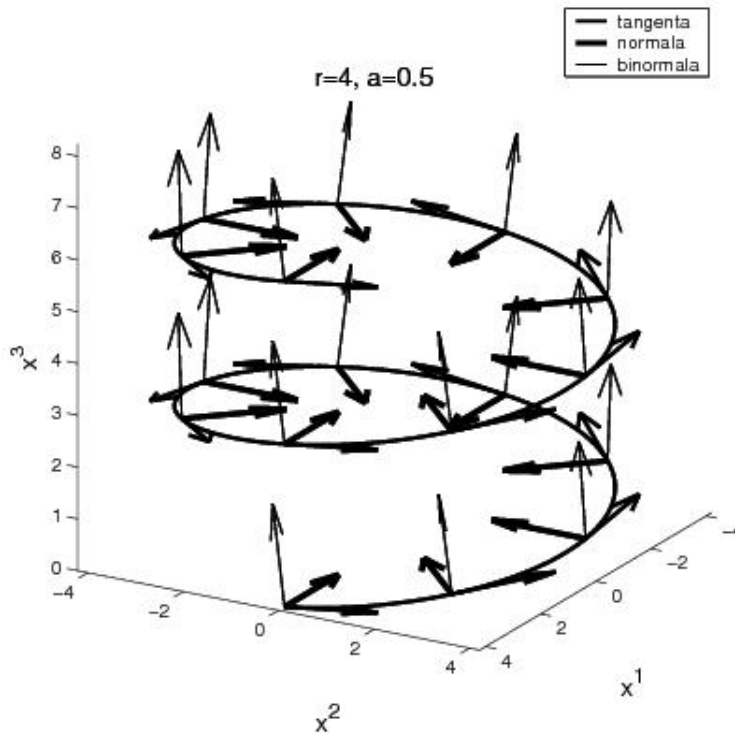


Figura 5.3: Triedrul Frenet.

Problema 5.6 (Loxodroma): *Un vapor se mișcă pe suprafața oceanului cu viteză constantă, unghiul dintre viteză și meridian fiind constant α . Să se determine traiectoria vaporului.*

Soluție:

Pământul este presupus sferic, cu raza R . Utilizând coordonate sferice

$$v_r = 0, v_q = R\dot{q}, v_j = Rj\dot{j}\cos q,$$

dar

$$v_q = v \cos \alpha, v_j = v \sin \alpha.$$

Prin urmare ecuația diferențială a traiectoriei este

$$\frac{dq}{dj} = \cos q \operatorname{ctg} \alpha.$$

Integrând cu condițiile inițiale la $t = 0$

$$q(0) = \frac{p}{2}, j(0) = 0,$$

se obține

$$\operatorname{tg} \frac{q}{2} = \exp(j \operatorname{ctg} \alpha).$$

Traectoria numită *loxodromă* este prezentată în figura 5.4 pentru $\alpha \in \left\{ \frac{p}{6}, \frac{p}{4}, \frac{p}{3} \right\}$,

folosind programul dat în continuare:


```

alpha = [pi/6 pi/4 pi/3];
n_alpha = [6 4 3];
for i=1:3
    subplot(1,3,i);
    [sx,sy,sz] = sphere(20);
    mesh(sx,sy,sz)
    colormap(gray);
    hold on
    q = pi/30:pi/30:pi*29/30;
    x = -sin(q).*cos(tan(alpha(i))*log(tan(q/2)));
    y = -sin(q).*sin(tan(alpha(i))*log(tan(q/2)));
    z = cos(q);
    plot3(x,y,z,'-r','LineWidth',2)
    title('\alpha=\pi/')
    axis equal;
    view([-120,20])
    title(['\alpha=\pi','/', num2str(n_alpha(i))])
end

```

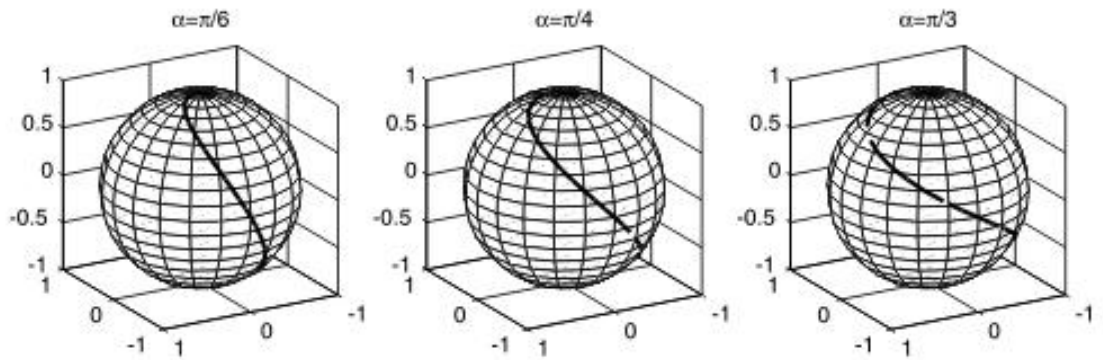


Figura 5.4: Loxodroma.

Problema 5.7 (Cicloida): Să se studieze mișcarea unui punct fixat P de pe o circumferință de rază a , care se rostogolește fără să alunece pe o dreaptă din planul circumferinței, centrul circumferinței având viteza constantă v_0 .

Soluție:

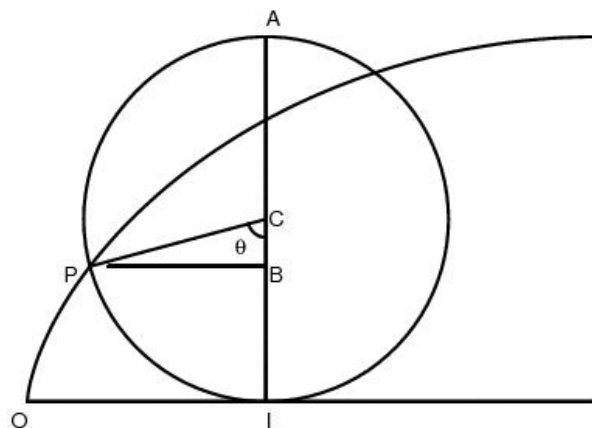


Figura 5.5: Generare cicloidă.

Fie reperul ortogonal (plan) cu originea în poziția inițială a punctului considerat, axa Ox^1 dreapta pe care alunecă cercul, ca în figura 5.5. Vectorul de poziție al punctului material $\mathbf{x} = \overrightarrow{OC} + \overrightarrow{CP}$ dă coordonatele punctului geometric P

$$\begin{cases} x^1 = OI - PB = aq - a \sin q = a(q - \sin q), \\ x^2 = CI - CB = a - a \cos q = a(1 - \cos q). \end{cases}$$

Acestea conduc la ecuațiile parametrice ale traiectoriei

$$\begin{cases} x^1 = a(q - \sin q), \\ x^2 = a(1 - \cos q), \end{cases}$$

care este o *cicloidă* (gr. *Kyklos* = cer, *eidos* = aspect).

Viteza punctului P are componentele în coordonate carteziene

$$\begin{cases} \dot{x}^1 = a\dot{q}(1 - \cos q), \\ \dot{x}^2 = a\dot{q}\sin q. \end{cases}$$

Se obține

$$v_o t = aq \Rightarrow \dot{q} = \frac{v_o}{a} = \text{const.}, \dot{q} = 0,$$

iar viteza punctului P are mărimea

$$\sqrt{(\dot{x}^1)^2 + (\dot{x}^2)^2} = 2a\dot{q}\sin \frac{q}{2} = 2v_o \sin \frac{q}{2} = \frac{v_o}{a} \pi$$

și este dirijată după direcția PA .

Accelerația punctului P are componentele în coordonate carteziene

$$\begin{cases} \ddot{x}^1 = a\dot{q}^2 \sin q = \frac{v_o^2}{a} \sin q, \\ \ddot{x}^2 = a\dot{q}^2 \cos q = \frac{v_o^2}{a} \cos q. \end{cases}$$

Traietoriei cicloidă i se asociază triedrul Frenet

$$\left(\frac{ds}{dq}\right)^2 = \left(\frac{dx^1}{dq}\right)^2 + \left(\frac{dx^2}{dq}\right)^2 = a^2[(1 - \cos q)^2 + \sin^2 q] = 2a^2,$$

$$\boldsymbol{\tau} = \frac{d\mathbf{x}}{ds} = \sin \frac{q}{2} \mathbf{u}_1 + \cos \frac{q}{2} \mathbf{u}_2,$$

$$\frac{d\boldsymbol{\tau}}{ds} = \frac{\mathbf{v}}{r} = \frac{1}{4a \sin \frac{q}{2}} \left(\cos \frac{q}{2} \mathbf{u}_1 - \sin \frac{q}{2} \mathbf{u}_2 \right) \Rightarrow$$

$$\Rightarrow \mathbf{v} = \cos \frac{q}{2} \mathbf{u}_1 - \sin \frac{q}{2} \mathbf{u}_2, \quad r = 4a \sin \frac{q}{2} = 2mas \left(\hat{PI} \right)$$

$$\boldsymbol{\beta} = \boldsymbol{\tau} \times \mathbf{v} = \mathbf{u}_3.$$

Triedrul lui Frenet și raza de curbură depind numai de elementele geometrice ale curbei, nu și de caracteristicile cinematice ale mișcării.

Proiecțiile vitezei și accelerației pe triedrul lui Frenet sunt

$$\mathbf{v} = 2v_o \sin \frac{q}{2} \boldsymbol{\tau}, \quad \mathbf{a} = \frac{v_o^2}{a} \left(\cos \frac{q}{2} \boldsymbol{\tau} - \sin \frac{q}{2} \mathbf{v} \right)$$

Prin programul următor se face o reprezentare animată a cicloidei. Rezultatul final al programului este dat în figura 5.6.

```

r = 5;
figure(1)
axis equal, axis([-6 100 -2 12]), hold on
line([-6,100],[0,0],'Color','y')
line([0, 0],[-2,12],'Color','y')
unghi = 0:0.01:2*pi; x1 = r*cos(unghi); x2 = r+r*sin(unghi);
cerc = plot(x1,x2)
punct = plot(0,0,'or','MarkerSize',2,'MarkerFaceColor','r')
pause
v0 = 1; timp = 0:0.5:95; m = length(timp);
for i=1:m
    x1 = v0*timp(i)+r*cos(unghi); x2 = r+r*sin(unghi);
    set(cerc,'Xdata',x1,'Ydata',x2)
    theta = v0*timp(i)/r;
    xp1 = v0*timp(i)-r*sin(theta); xp2 = r-r*cos(theta);
    set(punct,'Xdata',xp1,'Ydata',xp2)
    plot(xp1,xp2,'om','MarkerSize',2)
    drawnow
end

```

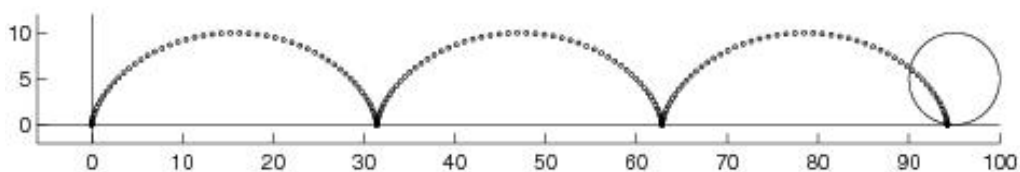


Figura 5.6: Câteva arce de cicloidă.

Problema 5.8 (Epicicloida): Să se studieze mișcarea unui punct fixat P de pe un cerc de rază a , care se rostogolește uniform fără să alunece pe un cerc fix de rază b , cele două cercuri fiind tangente exterior, centrul cercului mobil având viteza constantă v_0 . Mișcarea se desfășoară în planul celor două cercuri.

Soluție:

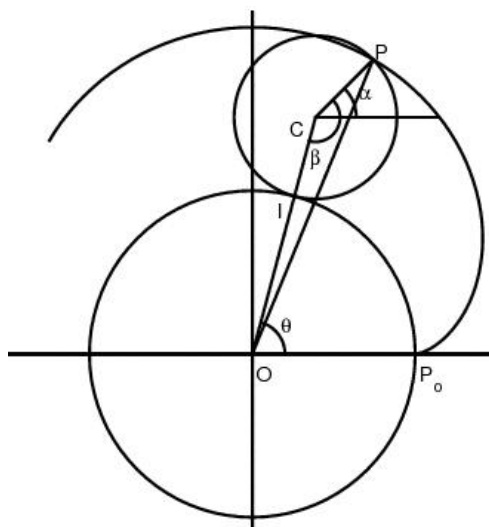


Figura 5.7: Generare epicicloidă.

Fie reperul ortogonal (plan) cu originea în centrul O al cercului fix și axa Ox^1 determinată de O și poziția inițială a punctului solidar cu cercul mobil, ca în figura 5.7
Vectorul de poziție al punctului material

$$\mathbf{x} = \overline{OC} + \overline{CP}$$

dă coordonatele punctului geometric P

$$\begin{cases} x^1 = (a+b)\cos q + a \cos a, \\ x^2 = (a+b)\sin q + a \sin a. \end{cases}$$

Întrucât nu există alunecare

$$\text{mas}\left(\begin{matrix} \cdot \\ IP_o \end{matrix}\right) = \text{mas}\left(\begin{matrix} \cdot \\ IP \end{matrix}\right) \Rightarrow ab = bq,$$

iar

$$a = b + q - p = \frac{a+b}{a}q - p$$

Acestea conduc la ecuațiile parametrice ale traiectoriei

$$\begin{cases} x^1 = (a+b)\cos q - a \cos\left(\frac{a+b}{a}q\right), \\ x^2 = (a+b)\sin q - a \sin\left(\frac{a+b}{a}q\right), \end{cases}$$

care este o *epicicloida* (gr. *epi*=pe, peste, *kyklos*=cerc, *eidōs*=aspect).

Viteza punctului P are componentele în coordonate carteziene

$$\begin{cases} \dot{x}^1 = -\dot{q}(a+b) \left[\sin q - \sin\left(\frac{a+b}{a}q\right) \right], \\ \dot{x}^2 = \dot{q}(a+b) \left[\cos q - \cos\left(\frac{a+b}{a}q\right) \right]. \end{cases}$$

Mișcarea centrului C fiind uniformă, se obține

$$v_o t = (a+b)q \Rightarrow \dot{q} = \frac{v_o}{a+b} = \text{const.}, \ddot{q} = 0,$$

iar viteza punctului P are mărimea

$$\sqrt{(\dot{x}^1)^2 + (\dot{x}^2)^2} = 2v_o \left| \sin\left(\frac{b}{2a}q\right) \right|.$$

Accelerația punctului P are componentele în coordonate carteziene

$$\begin{cases} \ddot{x}^1 = -(a+b)\ddot{q} \left[\cos q - \frac{a+b}{a} \cos\left(\frac{a+b}{a}q\right) \right], \\ \ddot{x}^2 = -(a+b)\ddot{q} \left[\sin q - \frac{a+b}{a} \sin\left(\frac{a+b}{a}q\right) \right]. \end{cases}$$

Traietoriei (epicicloida) i se asociază triedrul Frenet

$$\frac{ds}{dq} = \sqrt{\left(\frac{dx^1}{dq}\right)^2 + \left(\frac{dx^2}{dq}\right)^2} = 2(a+b) \left| \sin\left(\frac{b}{2a}q\right) \right|,$$

$$\boldsymbol{\tau} = \frac{d\mathbf{x}}{ds} = \frac{-\left[\sin q - \sin\left(\frac{a+b}{a}q\right)\right]\mathbf{u}_1 + \left[\cos q - \cos\left(\frac{a+b}{a}q\right)\right]\mathbf{u}_2}{2\left|\sin\left(\frac{b}{2a}q\right)\right|} =$$

$$\frac{2\sin\left(\frac{b}{2a}q\right)\cos\left(\frac{2a+b}{2a}q\right)\mathbf{u}_1 + 2\sin\left(\frac{b}{2a}q\right)\sin\left(\frac{2a+b}{2a}q\right)\mathbf{u}_2}{2\left|\sin\left(\frac{b}{2a}q\right)\right|}.$$

În ipoteza că $\sin\left(\frac{b}{2a}q\right) > 0$, se obține

$$\boldsymbol{\tau} = \cos\left(\frac{2a+b}{2a}q\right)\mathbf{u}_1 + \sin\left(\frac{2a+b}{2a}q\right)\mathbf{u}_2$$

$$\frac{d\boldsymbol{\tau}}{ds} = \frac{\mathbf{v}}{r} = \frac{\frac{2a+b}{2a}\left[-\sin\left(\frac{2a+b}{2a}q\right)\mathbf{u}_1 + \cos\left(\frac{2a+b}{2a}q\right)\mathbf{u}_2\right]}{2(a+b)\sin\left(\frac{b}{2a}q\right)} \Rightarrow$$

$$\Rightarrow \mathbf{v} = -\sin\left(\frac{2a+b}{2a}q\right)\mathbf{u}_1 + \cos\left(\frac{2a+b}{2a}q\right)\mathbf{u}_2, \quad r = \frac{4a(a+b)}{2a+b}\sin\left(\frac{b}{2a}q\right)$$

$$\boldsymbol{\beta} = \boldsymbol{\tau} \times \mathbf{v} = \mathbf{u}_3.$$

Triedrul lui Frenet și raza de curbură depind numai de elementele geometrice ale curbei, nu și de caracteristicile cinematice ale mișcării.

Proiecțiile vitezei și accelerației pe triedrul lui Frenet sunt

$$\mathbf{v} = 2v_o \sin\left(\frac{b}{2a}q\right)\boldsymbol{\tau}, \quad \mathbf{a} = \frac{v_o^2}{a+b}\left[\frac{b}{a}\cos\left(\frac{b}{2a}q\right)\boldsymbol{\tau} + \frac{2a+b}{a}\sin\left(\frac{b}{2a}q\right)\mathbf{v}\right].$$

Exemplificarea formei traiectoriilor, dată în figura 5.8 pentru diferite valori ale raportului $\frac{b}{a}$, se obține cu programul următor:

```
q = 0:pi/120:2*pi;
x1 = 2*cos(q)-cos(2*q); y1 = 2*sin(q)-sin(2*q);
x2 = 3*cos(q)-cos(3*q); y2 = 3*sin(q)-sin(3*q);
x3 = 4*cos(q)-cos(4*q); y3 = 4*sin(q)-sin(4*q);
xc1 = cos(q); yc1 = sin(q);
xc2 = 2*cos(q); yc2 = 2*sin(q);
xc3 = 3*cos(q); yc3 = 3*sin(q); subplot(1,3,1);
plot(xc1,yc1,'-b',x1,y1,'-r','LineWidth',2)
axis equal; axis([-5,5,-5,5]); axis off
title('b/a=1 cardioida','FontSize',14)
subplot(1,3,2); plot(xc2,yc2,'-b',x2,y2,'-r','LineWidth',2)
axis equal; axis([-5,5,-5,5]); axis off
title('b/a=2','FontSize',14)
subplot(1,3,3); plot(xc3,yc3,'-b',x3,y3,'-r','LineWidth',2)
axis equal; axis([-5.5,4.5,-5,5]); axis off
title('b/a=3','FontSize',14)
```

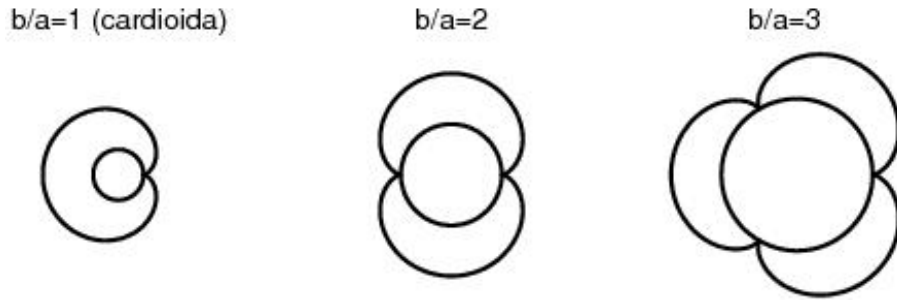


Figura 5.8: Exemple de epicloide.

Vizualizarea mișcării prin animație pentru $a = b$ se poate realiza cu programul:

```

a = 1; b = a;
ncadre = 180;
M = moviein(ncadre);
for j = 1:ncadre
    q = j*pi/90; qc = 0:pi/90:q;
    x = (a+b)*cos(qc)-a*cos((a+b)*qc/a);
    y = (a+b)*sin(qc)-a*sin((a+b)*qc/a);
    xp = (a+b)*cos(q)-a*cos((a+b)*q/a);
    yp = (a+b)*sin(q)-a*sin((a+b)*q/a); u = 0:pi/90:2*pi;
    xa = (a+b)*cos(q)+a*cos(u); ya = (a+b)*sin(q)+a*sin(u);
    xb = b*cos(u); yb=b*sin(u);
    plot(x,y,'r',xa,ya,'b',xb,yb,'b',xp,yp,'ob','LineWidth',2)
    axis equal; axis([-3.5,3.5,-3.5,3.5]); axis off
    M(j) = getframe;
end
movie(M)

```

Problema 5.9 (Hipocicloida): Să se studieze mișcarea unui punct fixat P de pe un cerc de rază a , care se rostogolește uniform fără să alunece pe un cerc fix de rază b , $a < b$, cercul mobil tangent în interiorul cercului fix, centrul cercului mobil având viteza constantă v_0 . Mișcarea se desfășoară în planul celor două cercuri.

Soluție:

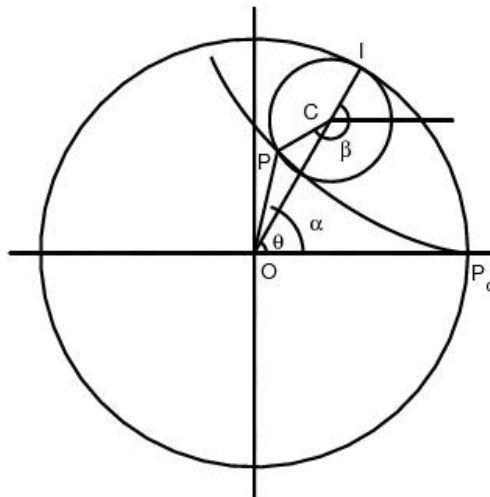


Figura 5.9: Generare hipocicloidă.

Fie reperul ortogonal (plan) cu originea în centrul O al cercului fix și axa Ox^1 determinată de O și poziția inițială a punctului solidar cu cercul mobil, ca în figura 5.9. Vectorul de poziție al punctului material

$$\mathbf{x} = \overline{OC} + \overline{CP}$$

dă coordonatele punctului geometric P

$$\begin{cases} x^1 = (b-a)\cos q + a\cos j, \\ x^2 = (b-a)\sin q + a\sin j. \end{cases}$$

Întrucât nu există alunecare

$$\text{mas}\left(\begin{matrix} \cdot \\ \text{IP}_o \end{matrix}\right) = \text{mas}\left(\begin{matrix} \cdot \\ \text{IP} \end{matrix}\right) \Rightarrow a\alpha = bq,$$

iar

$$j = 2p - \alpha + q = 2p - \frac{b-a}{a}q.$$

Acestea conduc la ecuațiile parametrice ale traiectoriei

$$\begin{cases} x^1 = (b-a)\cos q + a\cos\left(\frac{b-a}{a}q\right), \\ x^2 = (b-a)\sin q - a\sin\left(\frac{b-a}{a}q\right), \end{cases}$$

care este o *hipocicloidă* (gr. *hipo*=sub, dedesubt, *kyklos*=cerc, *eidōs*=aspect).

Viteza punctului P are componentele în coordonate carteziene

$$\begin{cases} \dot{x}^1 = -(b-a)\dot{q}\left[\sin q + \sin\left(\frac{b-a}{a}q\right)\right], \\ \dot{x}^2 = \dot{q}(b-a)\left[\cos q - \cos\left(\frac{b-a}{a}q\right)\right]. \end{cases}$$

Mișcarea centrului C fiind uniformă, se obține

$$v_{ot} = (b-a)\dot{q} \Rightarrow \dot{q} = \frac{v_o}{b-a} = \text{const.}, \ddot{q} = 0,$$

iar viteza punctului P are mărimea

$$\sqrt{(\dot{x}^1)^2 + (\dot{x}^2)^2} = 2v_o \left| \sin\left(\frac{b}{2a}q\right) \right|.$$

Accelerația punctului P are componentele în coordonate carteziene

$$\begin{cases} \ddot{x}^1 = -(b-a)\ddot{q}\left[\cos q - \frac{b-a}{a}\cos\left(\frac{b-a}{a}q\right)\right], \\ \ddot{x}^2 = -(b-a)\ddot{q}\left[\sin q - \frac{b-a}{a}\sin\left(\frac{b-a}{a}q\right)\right]. \end{cases}$$

Traietoriei (hipocicloida) i se asociază triedrul Frenet

$$\frac{ds}{dq} = \sqrt{\left(\frac{dx^1}{dq}\right)^2 + \left(\frac{dx^2}{dq}\right)^2} = 2(b-a) \left| \sin\left(\frac{b}{2a}q\right) \right|,$$

$$\boldsymbol{\tau} = \frac{d\mathbf{x}}{ds} = \frac{-\left[\sin q + \sin\left(\frac{b-a}{a}q\right)\right]\mathbf{u}_1 + \left[\cos q - \cos\left(\frac{b-a}{a}q\right)\right]\mathbf{u}_2}{2\left|\sin\left(\frac{b}{2a}q\right)\right|} =$$

$$\frac{2\sin\left(\frac{b}{2a}q\right)\cos\left(\frac{b-2a}{2a}q\right)\mathbf{u}_1 + 2\sin\left(\frac{b}{2a}q\right)\sin\left(\frac{b-2a}{2a}q\right)\mathbf{u}_2}{2\left|\sin\left(\frac{b}{2a}q\right)\right|}.$$

În ipoteza că $\sin\left(\frac{b}{2a}q\right) > 0$, se obține

$$\boldsymbol{\tau} = \cos\left(\frac{b-2a}{2a}q\right)\mathbf{u}_1 + \sin\left(\frac{b-2a}{2a}q\right)\mathbf{u}_2$$

$$\frac{d\boldsymbol{\tau}}{ds} = \frac{\mathbf{v}}{r} = \frac{\frac{b-2a}{2a}\left[-\sin\left(\frac{b-2a}{2a}q\right)\mathbf{u}_1 + \cos\left(\frac{b-2a}{2a}q\right)\mathbf{u}_2\right]}{2(b-a)\sin\left(\frac{b}{2a}q\right)} \Rightarrow$$

$$\Rightarrow \mathbf{v} = -\sin\left(\frac{b-2a}{2a}q\right)\mathbf{u}_1 + \cos\left(\frac{b-2a}{2a}q\right)\mathbf{u}_2, \quad r = \frac{4a(b-a)}{b-2a}\sin\left(\frac{b}{2a}q\right)$$

$$\boldsymbol{\beta} = \boldsymbol{\tau} \times \mathbf{v} = \mathbf{u}_3.$$

Triedrul lui Frenet și raza de curbură depind numai de elementele geometrice ale curbei. Proiecțiile vitezei și accelerației pe triedrul lui Frenet sunt

$$\mathbf{v} = 2v_o \sin\left(\frac{b}{2a}q\right)\boldsymbol{\tau}, \quad \mathbf{a} = \frac{v_o^2}{b-a}\left[\frac{b}{a}\cos\left(\frac{b}{2a}q\right)\boldsymbol{\tau} + \frac{b-2a}{a}\sin\left(\frac{b}{2a}q\right)\mathbf{v}\right].$$

Exemplificarea formei traiectoriilor pentru diferite valori ale raportului b/a este reprezentată în figura 5.10 și se obține cu programul:

```
q = 0:pi/120:2*pi;
x1 = 2*cos(q)+cos(2*q); y1 = 2*sin(q)-sin(2*q);
x2 = 3*cos(q)+cos(3*q); y2 = 3*sin(q)-sin(3*q);
x3 = 4*cos(q)+cos(4*q); y3 = 4*sin(q)-sin(4*q);
xc1 = 3*cos(q); yc1 = 3*sin(q);
xc2 = 4*cos(q); yc2 = 4*sin(q);
xc3 = 5*cos(q); yc3 = 5*sin(q);
subplot(1,3,1); plot(x1,y1,'-r',xc1,yc1,'-b','LineWidth',2)
axis equal; axis([-3,3,-3,3]); axis off
title('b/a=3','FontSize',14)
subplot(1,3,2); plot(x2,y2,'-r',xc2,yc2,'-b','LineWidth',2)
axis equal; axis([-4,4,-4,4]); axis off
title('b/a=4 (astroida)','FontSize',14)
subplot(1,3,3); plot(x3,y3,'-r',xc3,yc3,'-b','LineWidth',2)
axis equal; axis([-5,5,-5,5]); axis off
title('b/a=5','FontSize',14)
```

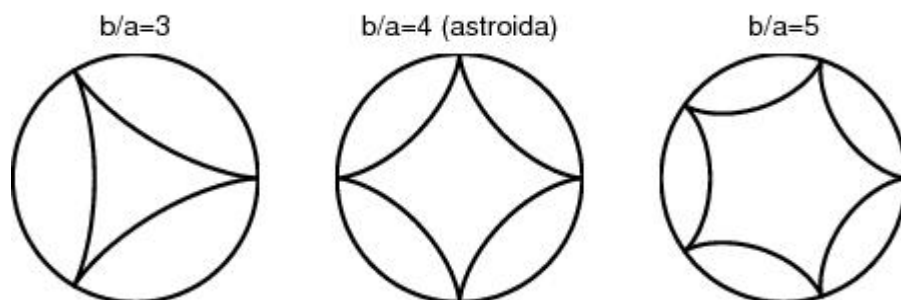



Figura 5.10. Exemple de hipocicloide.

Vizualizarea mișcării prin animație pentru $b = 4a$ se poate realiza cu programul următor:

```

a = 1; b = 4*a;
ncadre = 180;
M = moviein(ncadre);
for j = 1:ncadre
    q = j*pi/90; qc = 0:pi/90:q;
    x = (b-a)*cos(qc)+a*cos((b-a)*qc/a);
    y = (b-a)*sin(qc)-a*sin((b-a)*qc/a);
    xp = (b-a)*cos(q)+a*cos((b-a)*q/a);
    yp = (b-a)*sin(q)-a*sin((b-a)*q/a);
    u = 0:pi/90:2*pi;
    xa = (b-a)*cos(q)+a*cos(u); ya = (b-a)*sin(q)+a*sin(u);
    xb = b*cos(u); yb = b*sin(u);
    plot(x,y,'-r',xa,ya,'b',xb,yb,'b',xp,yp,'ob','LineWidth',2)
    axis equal; axis([-4.5,4.5,-4.5,4.5]); axis off
    M(j) = getframe;
end
movie(M)

```

5.3. Mișcarea în câmp central

Problema 5.10 (Mișcarea kepleriană): Să se studieze mișcarea punctului material sub acțiunea forței de atracție universală.

Soluție:

Forța de atracție universală este definită prin

$$\mathbf{F} = -f \frac{Mm \mathbf{x}}{r^2 r}, \quad r = \|\mathbf{x}\|,$$

unde s-au notat: m - masa punctului material atras,

M - masa centrului atractiv,

f - constanta de atracție universală.

Pentru mișcarea în câmp central sunt cunoscute următoarele rezultate:

- traiectoria este plană conținută în planul determinat de poziția inițială și viteza inițială date prin

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{r}_o, & \|\mathbf{r}_o\| &= r_o \\ \mathbf{v}(0) &= \mathbf{v}_o, & \|\mathbf{v}_o\| &= v_o, & \angle(\mathbf{r}_o, \mathbf{v}_o) &= \alpha \end{aligned}$$

- în planul traiectoriei mișcarea are loc cu viteză areolară constantă

$$r^2 \dot{q} = C = r_o v_o \sin a;$$

- ecuația traiectoriei în coordonate polare se obține din ecuația lui Binet

$$\frac{d^2}{dq^2} \left(\frac{1}{r} \right) + \frac{1}{r} = - \frac{r^2 F(r)}{mC^2},$$

unde $F(r) = \|\mathbf{F}\|$.

În cazul mișcării sub acțiunea forței de atracție universală se pot stabili câteva proprietăți particulare.

1. Traiectoria unui punct material supus acțiunii forței de atracție universală este o conică care depinde de condițiile inițiale:

- traectoria este elipsă dacă $k = \frac{r_o v_o^2}{fM} < 2$ (adică $e < 1$);
- traectoria este parabolă dacă $k = \frac{r_o v_o^2}{fM} = 2$ (adică $e = 1$);
- traectoria este hiperbolă dacă $k = \frac{r_o v_o^2}{fM} > 2$ (adică $e > 1$).

Într-adevăr, în cazul forței de atracție universală, formula lui Binet devine

$$\frac{d^2}{dq^2} \left(\frac{1}{r} \right) + \frac{1}{r} = \frac{fM}{C^2}$$

și are soluția

$$\frac{1}{r} = A \cos q + B \sin q + \frac{fM}{C^2}.$$

Condițiile inițiale în coordonate polare (r, θ) se scriu astfel

$$r(0) = r_o, \quad \dot{r}(0) = \dot{r}_o = v_o \cos a, \quad r(0) \dot{q}(0) = r_o \dot{q}_o = v_o \sin a$$

Cum $\frac{d}{dq} \left(\frac{1}{r} \right) = - \frac{1}{r^2} \dot{r}$, pentru determinarea constantelor A și B se obține sistemul

$$\begin{cases} A \cos q_o + B \sin q_o = \frac{1}{r_o} - \frac{fM}{C^2} \\ -A \sin q_o + B \cos q_o = - \frac{v_o \cos a}{C}, \end{cases}$$

de unde rezultă

$$\begin{cases} A = \left(\frac{1}{r_o} - \frac{fM}{C^2} \right) \cos q_o + \frac{v_o \cos a}{C} \sin q_o, \\ B = \left(\frac{1}{r_o} - \frac{fM}{C^2} \right) \sin q_o - \frac{v_o \cos a}{C} \cos q_o. \end{cases}$$

Soluția obținută se poate scrie sub forma

$$r = \frac{p}{1 + e \cos(q - q_1)},$$

ceea ce arată că traiectoria este o conică, cu centrul atractiv în focar, și unde

$$p = \frac{C^2}{fM} = \frac{r_o^2 v_o^2 \sin^2 a}{fM}, \quad e \cos(q_o - q_1) = \frac{p}{r_o} - 1, \quad e \sin(q_o - q_1) = \frac{p \cos a}{r_o \sin a},$$

$$e^2 = 1 + \frac{p}{r_o} \left(\frac{p}{r_o \sin^2 a} - 2 \right) = 1 + \frac{p}{r_o} \left(\frac{r_o v_o^2}{fM} - 2 \right), \quad \operatorname{tg}(q_o - q_1) = \frac{r_o v_o^2 \sin a \cos a}{r_o v_o^2 \sin^2 a - fM}.$$

de unde rezultă natura conice.

Același rezultat se putea obține utilizând integrala primă a energiei. Întâi să observăm că forța de atracție universală este o forță conservativă (potențială), deoarece

$$\begin{aligned} P = \mathbf{F} \cdot \mathbf{v} &= -f \frac{Mm \mathbf{x}}{r^2} \cdot \frac{d\mathbf{x}}{dt} = -f \frac{Mm}{r^3} \frac{d}{dt} \left(\frac{1}{2} \mathbf{x} \cdot \mathbf{x} \right) = -f \frac{Mm}{r^3} \frac{d}{dt} \left(\frac{1}{2} r^2 \right) = \\ &= -f \frac{Mm}{r^3} r \frac{dr}{dt} = -f \frac{Mm}{r^2} \frac{dr}{dt} = \frac{d}{dr} \left(f \frac{Mm}{r} \right) \frac{dr}{dt} = \frac{d}{dt} \left(f \frac{Mm}{r} \right), \end{aligned}$$

de unde rezultă că potențialul forței de atracție universală (pentru un punct material atractiv) este

$$U(r) = f \frac{Mm}{r}.$$

Integrala primă a energiei se scrie:

$$\frac{1}{2} m (\dot{\mathbf{r}}^2 + r^2 \dot{\mathbf{q}}^2) - U(r) = h,$$

unde $h = \frac{1}{2} m v_o^2 - \frac{fMm}{r_o}$ este constanta energiei. Rezultă

$$\frac{1}{2} m C^2 \left(\left(\frac{d}{dq} \left(\frac{1}{r} \right) \right)^2 + \frac{1}{r^2} \right) - \frac{fMm}{r} = h,$$

unde notând $u = \frac{1}{r}$, se obține

$$\begin{aligned} \left(\frac{du}{dq} \right)^2 &= \frac{2}{mC^2} (h + fMmu) - u^2 \Rightarrow \left(\frac{du}{dq} \right)^2 = \frac{1}{C^2} \left(\frac{2h}{m} + \frac{(fM)^2}{C^2} \right) - \left(u - \frac{fM}{C^2} \right)^2 \Rightarrow \\ &= \operatorname{arccos} \left(\frac{C \left(u - \frac{fM}{C^2} \right)}{\sqrt{\frac{2h}{m} + \frac{(fM)^2}{C^2}}} \right) = q - q_1 \Rightarrow u = \frac{fM}{C^2} + \sqrt{\frac{2h}{mC^2} + \frac{(fM)^2}{C^4}} \cos(q - q_1). \end{aligned}$$

Deci, rezultă

$$r = \frac{p}{1 + e \cos(q - q_1)},$$

unde

$$p = \frac{C^2}{fM}, \quad e = \sqrt{1 + \frac{2hC^2}{m(fM)^2}}.$$

2. Dacă traiectoria este elipsă ($e < 1$), atunci se definesc apsidele pentru $\theta = \theta_1$ și $\theta = \theta_1 + \pi$:

$$a = \frac{1}{2} \left(\frac{p}{1+e} + \frac{p}{1-e} \right) = \frac{p}{1-e^2}, \quad b = a\sqrt{1-e^2} = \frac{p}{\sqrt{1-e^2}}.$$

ținând seama de legea ariilor $r^2 \dot{q} = C$, pentru perioada mișcării rezultă

$$t = \frac{p ab}{C}$$

Se obține astfel **legea a III-a a lui Kepler: raportul dintre pătratul perioadei și cubul semiaxei mari nu depinde de punctul material în câmpul de atracție al aceluiași centru**

$$\frac{t^2}{a^3} = \frac{\left(p a^2 \sqrt{1-e^2} \right)^2}{a^3 C^2} = \frac{p^2 a (1-e^2)}{C^2} = \frac{p^2 p}{C^2} = \frac{p^2}{fM} = \text{const.}$$

3. Dacă sunt adevărate **legile lui Kepler:**

I: traiectoria punctului material este o elipsă;

II: raza vectoare mătură arii egale în intervale de timp egale (viteză areolară constantă);

III: raportul dintre pătratul perioadei și cubul semiaxei mari este constant, atunci mișcarea are loc sub acțiunea forței de atracție universală, trecând prin focarul elipsei. Într-adevăr, traiectoria fiind elipsă, este plană și ecuația ei în coordonate polare este

$$r = \frac{p}{1 + e \cos(q - q_1)}, \quad \text{cu } e < 1.$$

Legea a doua a lui Kepler în coordonate polare se scrie

$$r^2 \dot{q} = C,$$

iar ecuațiile de mișcare în coordonate polare sunt

$$m \left(\ddot{r} - r \dot{q}^2 \right) = F_r, \quad m \left(r \ddot{q} + 2 \dot{r} \dot{q} \right) = F_q.$$

Din ecuația a doua de mișcare și legea a doua a lui Kepler, rezultă $F_q = 0$, deci mișcarea se efectuează sub acțiunea unei forțe centrale trecând prin focar.

Aplicând formula lui Binet, se obține

$$F = F_r = -\frac{mC^2}{r^2} \left(\frac{d^2}{dq^2} \left(\frac{1}{r} \right) + \frac{1}{r} \right) = -\frac{mC^2}{r^2} \left(-\frac{e \cos(q - q_1)}{p} + \frac{1 + e \cos(q - q_1)}{p} \right) = -\frac{mC^2}{pr^2}.$$

Pentru cazul în care traiectoria este elipsă, știm că

$$\frac{t^2}{a^3} = \frac{p^2 p}{C^2} = \text{const.},$$

de unde rezultă că $\frac{C^2}{p}$ este constantă independentă de punctul material, deci F este invers proporțională cu pătratul distanței r .

Se introduce unghiul q prin relația

$$r \sin(q - q_1) = a\sqrt{1-e^2} \sin q.$$

Rezultă $r \cos(q - q_1) = a(\cos q - e)$ și $r = a(1 - e \cos q)$, $\text{tg} \frac{q - q_1}{2} = \sqrt{\frac{1+e}{1-e}} \text{tg} \frac{q}{2}$.

Se obține

$$r = a\sqrt{1 - e^2 \cos^2 q}$$

iar din integrala primă a ariilor

$$C = ab(1 - e \cos q)$$

de unde rezultă

$$q - e \sin q = \frac{C}{ab}(t - t_0),$$

relație utilă în stabilirea dependenței de timp a coordonatelor polare ale punctului material.

În figura 5.11 sunt prezentate exemple de conice depinzând de condițiile inițiale, pentru aceeași poziție inițială r_0 (aleasă drept unitate de lungime) și pentru o

viteză inițială a cărei marime verifică relația $k = \frac{r_0 v_0^2}{fM}$ și care face cu direcția inițială

θ_0 un unghi α . Am considerat $k \in \{1.7, 2, 2.5\}$ (elipsă, parabolă, hiperbolă), $\theta_0 = 0$ și $\alpha = \pi / 2$ (figura 5.11a), respectiv $\alpha = \pi / 4$ (figura 5.11b). Rezultatele sunt obținute prin programul:

```

ke = 1.7; kp = 2; kh = 2.5;
ue1 = -pi:pi/360:pi; re1 = ke./(1+(ke-1)*cos(ue1));
xe1 = re1.*cos(ue1); ye1 = re1.*sin(ue1);
up1 = -2.5:0.01:2.5; rp1 = kp./(1+(kp-1)*cos(up1));
xp1 = rp1.*cos(up1); yp1 = rp1.*sin(up1);
uh1 = -2.05:0.01:2.05; rh1=kh./(1+(kh-1)*cos(uh1));
xh1=rh1.*cos(uh1); yh1=rh1.*sin(uh1);
ue2 = -pi:pi/360:pi; re2=ke./(2+(ke-2)*cos(ue2)-ke*sin(ue2));
xe2 = re2.*cos(ue2); ye2 = re2.*sin(ue2);
up2 = 2:0.01:7.5; rp2 = kp./(2-kp*sin(up2));
xp2 = rp2.*cos(up2); yp2 = rp2.*sin(up2);
uh2 = -3.8: 0.01: 1; rh2=kh./(2+(kh-2)*cos(uh2)-kh*sin(uh2));
xh2 = rh2.*cos(uh2); yh2 = rh2.*sin(uh2);
subplot(1,2,1);
plot(xe1,ye1,'-r',xp1,yp1,':g',xh1,yh1,'-.b','LineWidth',2);
Xlabel('x_1/r_o','FontSize',14);
Ylabel('x_2/r_o','FontSize',14);
text(-5.5,0,'k=1.7','FontSize',14);
text(-6,-4.2,'k=2','FontSize',14);
text(-2,6,'k=2.5','FontSize',14);
text(-2,-6,'\alpha=\pi/2','FontSize',14);
axis equal; axis([-8 2 -8 8])
subplot(1,2,2);
plot(xe2,ye2,'-r',xp2,yp2,':g',xh2,yh2,'-.b','LineWidth',2);
Xlabel('x_1/r_o','FontSize',14);
Ylabel('x_2/r_o','FontSize',14);
text(-0.5,4.4,'k=1.7','FontSize',14);
text(1.9,6.5,'k=2','FontSize',14);
text(-4,-1,'k=2.5','FontSize',14);
text(-1,-4,'\alpha=\pi/4','FontSize',14);
axis equal
axis([-5 5 -8 8])

```

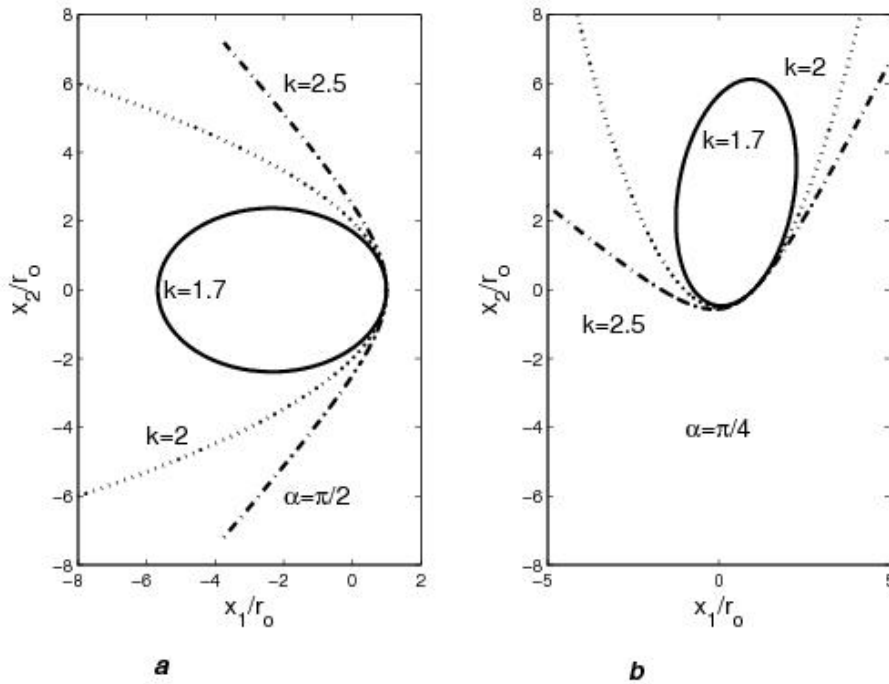


Figura 5.11: Tipul conicei în funcție de viteza inițială $k = \frac{v_o r_o}{k f M}$

Programul următor cuprinde animația a mișcării pe elipsă:

```

a = 1; e = 0.85; c = e*a; b = a*sqrt(1-e^2); r = 0.05;
ncadre = 72; M = moviein(ncadre);
syms x
s = 0:pi/36:2*pi
for j = 1:72
    t = (j-1)*pi/36;
    u = solve(x-e*sin(x)-t); p(j)=eval(u);
end
for j = 1:72
    x1 = a*cos(s); y1 = b*sin(s);
    xc = a*cos(p(j))+r*cos(s)/2; xc = eval(xc)
    yc = b*sin(p(j))+r*sin(s)/2; yc = eval(yc)
    xs = c+r*cos(s); ys = r*sin(s)
    plot(x1,y1,'-r',xc,yc,'.r',xs,ys,'.b')
    axis equal; axis([-1.1,1.1,-1,1]);
    M(j) = getframe;
end
movie(M,3)

```

Problema 5.11 (Problema sateliților artificiali): Să se studieze mișcarea în câmp central în cazul în care corpul atractiv este Pământul, iar corpul atras un satelit artificial, sub acțiunea forței de atracție universală. Să se stabilească condițiile de satelizare.

Soluție: La suprafața Pământului, forța de atracție universală este egală cu greutatea punctului material, deci avem

$$F = mg = f \frac{Mm}{R^2} \Rightarrow fM = gR^2,$$

unde $R = R_p$ este raza Pământului. Traectoria "punctului material" atras trebuie să fie o elipsă, pentru ca acesta să fie satelit. Deci

$$r_o v_o^2 < 2fM = 2gR^2.$$

Prin urmare, viteza de lansare a satelitului de la distanța $r_o > R$ trebuie să fie astfel încât

$$r_o v_o^2 < 2gR^2. \quad (5.13)$$

Traectoria satelitului (elipsa) nu trebuie să intersecteze suprafața Pământului, deci pentru

$$r = \frac{p}{1 + e \cos q}, \quad p = \frac{r_o^2 v_o^2 \sin^2 a}{fM}, \quad e^2 = 1 + \frac{p}{r_o} \left(\frac{r_o v_o^2}{fM} - 2 \right)$$

trebuie să avem

$$r > R \quad \forall q \in [0, 2p]$$

de unde rezultă că

$$r_{\min} = \frac{p}{1 + e} > R.$$

Astfel, obținem

$$\frac{p}{1 + e} > R \Rightarrow \frac{p}{R} > 1 + e \Rightarrow \left(\frac{p}{R} - 1 \right)^2 > e^2 \Rightarrow \frac{p^2}{R^2} - 2\frac{p}{R} + 1 > 1 + \frac{p}{r_o} \left(\frac{r_o v_o^2}{fM} - 2 \right) \Rightarrow$$

$$\frac{p}{R^2} - \frac{v_o^2}{fM} > 2 \left(\frac{1}{R} - \frac{1}{r_o} \right) \Rightarrow \frac{v_o^2}{fM} \left(\frac{r_o^2 \sin^2 a}{R^2} - 1 \right) > 2 \left(\frac{1}{R} - \frac{1}{r_o} \right)$$

Cum $r_o > R$ rezultă că

$$r_o^2 \sin^2 a - R^2 > 0;$$

și

$$v_o^2 > 2fM \frac{(r_o - R)}{Rr_o} \frac{R^2}{(r_o^2 \sin^2 a - R^2)} \Rightarrow v_o^2 > \frac{2gR^2}{r_o} \frac{(r_o - R)R}{(r_o^2 \sin^2 a - R^2)}$$

de unde folosind și condiția (5.13) se obține

$$\frac{2gR^2}{r_o} \frac{(r_o - R)R}{(r_o^2 \sin^2 a - R^2)} < v_o^2 < \frac{2gR^2}{r_o} \quad (5.14)$$

În consecință, condițiile de satelizare sunt

$$\begin{cases} r_o^2 \sin^2 a > R^2, \\ \frac{2gR^3(r_o - R)}{r_o(r_o^2 \sin^2 a - R^2)} < v_o^2 < \frac{2gR^2}{r_o}. \end{cases} \quad (5.15)$$

Din (5.15)₂ și echivalența

$$2r_o \sin^2 a (r_o - R) > r_o^2 \sin^2 a - R^2 \Leftrightarrow (r_o \sin a - R)^2 + 2r_o R \sin a (1 - \sin a) > 0$$

se obține inegalitatea

$$gR^3 < \frac{2gR^2 r_o (r_o - R) \sin^2 a}{r_o^2 \sin^2 a - R^2} < r_o^2 v_o^2 \sin^2 a,$$

Din inegalitatea $\frac{p}{1 + e} > R$ rezultă și că $p > R$.

Dacă intrarea pe traiectorie se face sub unghi $\alpha = \pi / 2$, atunci din (5.15) se obține

$$\frac{2gR^3}{r_o(r_o + R)} < v_o^2 < \frac{2gR^2}{r_o}.$$

Dacă lansarea se realizează aproximativ de la suprafața Pământului, $r_o \approx R$, atunci

$$gR < v_o^2 < 2gR.$$

Mărimea $\sqrt{gR} \approx 7.29$ km/s reprezintă *prima viteză cosmică*, iar mărimea $\sqrt{2gR} \approx 11.2$ km/s reprezintă *a doua viteză cosmică*. Orice corp lansat de la suprafața Pământului, perpendicular pe raza vectoare, cu o viteză cuprinsă între cele două viteze cosmice, devine satelit al Pământului. Pentru a evita efectele frecării cu atmosfera, se ia r_o încât să includă și grosimea atmosferei (intrarea pe orbită la limita superioară a atmosferei); în acest caz, vitezele cosmice se modifică ușor.

Pentru abordarea în MATLAB considerăm variabilele adimensionale

$$X = \frac{r_o}{R} > 1, Y = \frac{v_o}{\sqrt{gR}} > 0.$$

Condiția de satelizare (5.15)₂ se scrie în variabile adimensionale, astfel

$$\begin{cases} XY^2 < 2, \\ Y^2(X^2 \sin^2 a - 1) > 2\left(1 - \frac{1}{X}\right) \end{cases}$$

Aceste relații sugerează considerarea în plan (X, Y) a curbelor

$$\Gamma_0: Y = \sqrt{\frac{2}{X(X+1)}}, \quad \Gamma_1: Y = \sqrt{\frac{2}{X}}, \quad \Gamma_2: Y = \sqrt{\frac{2(X-1)}{X(X^2 \sin^2 a - 1)}}$$

unde Γ_0 este, de fapt, Γ_2 pentru $a = \frac{\pi}{2}$. Aceste curbe sunt reprezentate grafic în figura

5.12 rezultată din programul

```
x1 = 0.2:0.001:6; y1 = sqrt(2./x1); y0 = sqrt(2./x1./(x1+1));
all = pi/3; x210 = 1/sin(all)^2;
x21 = x210:0.01:6;
y21 = sqrt(2*(x21-1)./x21./(x21.*x21*sin(all)^2-1));
al2 = pi/4; x220 = 1/sin(al2)^2; x22 = x220:0.01:6;
y22 = sqrt(2*(x22-1)./x22./(x22.*x22*sin(al2)^2-1));
al3 = pi/6; x230 = 1/sin(al3)^2; x23 = x230:0.01:6;
y23 = sqrt(2*(x23-1)./x23./(x23.*x23*sin(al3)^2-1));
yas = 0:0.1:3;
plot(7,0,1,yas,':k',x1,y0,'-k',x1,y1,'-k',x21,y21,':k',...
     x22,y22,':k',x23,y23,':k','LineWidth',2);
text(1.45,1.1,'\leftarrow\alpha=\pi/3','FontSize',14);
text(2.7,0.7,'\leftarrow\alpha=\pi/4','FontSize',14);
text(5.5,0.56,'\leftarrow\alpha=\pi/6','FontSize',14);
text(0.4,1, '(\Gamma_0)', 'FontSize', 14);
text(0.45,2.2, '(\Gamma_1)', 'FontSize', 14);
text(6.2,0.3, '(\Gamma_2)', 'FontSize', 14);
xlabel('X=r_o/R', 'FontSize', 14);
ylabel('Y=v_o/sqrt(gR)', 'FontSize', 14);
text(2,3, '\Gamma_0: Y=sqrt(2/X/(X+1))', 'FontSize', 14);
```



```

text(2,2.75,'\Gamma_1: Y=sqrt(2/X)', 'FontSize',14);
text(2,2.5,'\Gamma_2: Y=sqrt(2(X-1)/X/(X^2sin^2(\alpha)-1))');
title('Conditii de satelizare', 'FontSize',14)

```

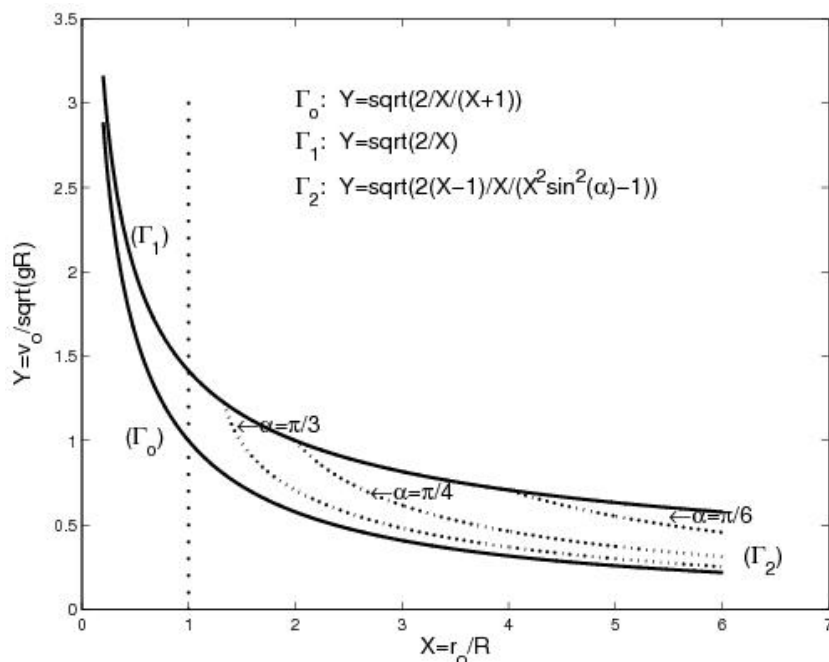


Figura 5.12: Condiții de satelizare.

Domeniul de satelizare $D = D_1 \cap D_2(\alpha)$, domeniul D_1 fiind delimitat inferior de axa OX , la stânga de dreapta $X=1$ și superior de curba Γ_1 , iar $D_2(\alpha)$ inferior de curba Γ_2 , la stânga de dreapta $X = \frac{1}{\sin \alpha}$. În programul următor sunt calculate, ca exemplu, câteva orbite în funcție de vitezele inițiale. Reprezentarea grafică a orbitelor este în figura 5.13.

```

X0 = 5.5; Y0 = 0.4;
cond = (2*(X0-1)/X0/Y0^2+1)/X0^2;
cd = sqrt(cond/(1-cond));
alinf = atan(cd); alsup = pi-alinf;
alo = pi/2; ale = pi/4; u=0:pi/180:2*pi;
xc = cos(u); yc = sin(u);
lo=1/(X0*Y0*sin(alo))^2+(1/X0-1/(X0*Y0*sin(alo)))^2*cos(u)-...
    cot(alo)/X0*sin(u);
xo = cos(u)./lo; yo = sin(u)./lo;
le=1/(X0*Y0*sin(ale))^2+(1/X0-1/(X0*Y0*sin(ale)))^2*cos(u)...
    -cot(ale)/X0*sin(u);
xe = cos(u)./le; ye = sin(u)./le;
linf=1/(X0*Y0*sin(alinf))^2+(1/X0-1/(X0*Y0*sin(alinf)))^2*cos(u)-...
    cot(alinf)/X0*sin(u);
xinf = cos(u)./linf; yinf = sin(u)./linf;
lsup=1/(X0*Y0*sin(alsup))^2+(1/X0-1/(X0*Y0*sin(alsup)))^2*cos(u)-...
    cot(alsup)/X0*sin(u);
xsup = cos(u)./lsup; ysup = sin(u)./lsup;
plot(xc,yc,'-k','LineWidth',3);
axis equal; hold on

```

```

plot(0,7,0,-7,xo,yo,'-k',xinf,yinf,'--k',xsup,ysup,'--k',...
     xe,ye,':k','LineWidth',2);
axis([-6 12 -7 7]); axis off
text(-0.2,0,'T','FontSize',14);
text(6,6,'\leftarrow \alpha=\pi/4','FontSize',14);
text(6.7,0.9,['\leftarrow \alpha_{i_n_f}=' ,num2str(alinf)],...
      'FontSize',14);
text(6.7,-1.2,['\leftarrow \alpha_{s_u_p}=' , num2str(alsup)],...
      'FontSize',14);
text(-5.2,-4,'\alpha=\pi/2 \rightarrow','FontSize',14);
title('Orbite in functie de vitezele initiale','FontSize',14)
hold off

```

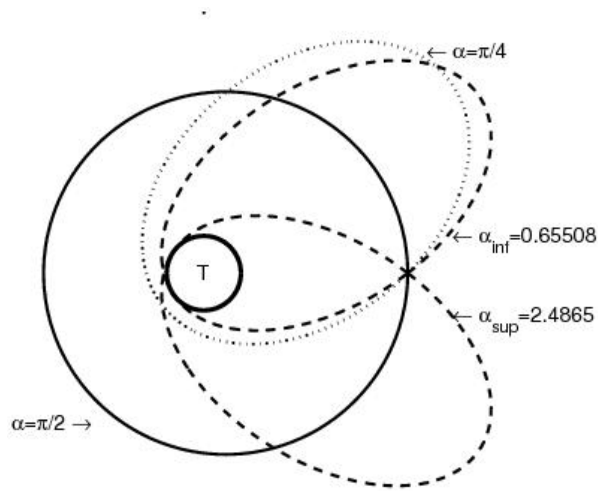


Figura 5.13: Orbite în funcție de vitezele inițiale.

Problema 5.12 (Mișcarea pe semicircumferință): *Un punct material se mișcă pe o circumferință de diametru $2a$ sub acțiunea unei forțe de atracție din partea unui punct fix O al circumferinței (figura 5.14). Să se deducă expresia forței centrale care determină această mișcare și viteza în punctul A diametral opus lui O pentru care mișcarea să fie posibilă.*

Soluție:

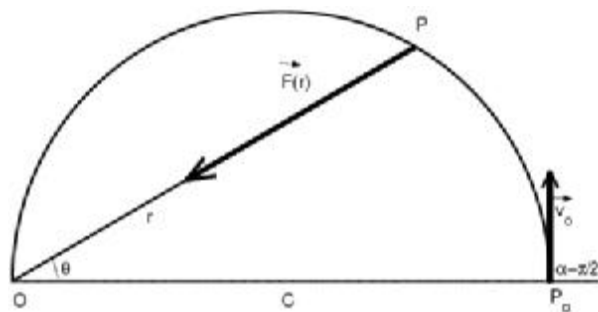


Figura 5.14: Mișcarea pe circumferință.

Fie OA axa Ox^1 . Datele inițiale ale problemei (cu A poziție inițială) sunt

$$r_o = 2a, q_o = 0, \dot{\theta} = 0, r_o \dot{\phi}_o = v_o, C = 2av_o.$$

Reprezentarea parametrică a traiectoriei fiind

$$r = 2a \cos q,$$

din ecuația lui Binet rezultă

$$F(r) = -\frac{8ma^2 C^2}{r^5};$$

prin urmare forța care determină mișcarea are expresia

$$F(r) = -\frac{mk}{r^5}, k > 0,$$

dacă viteza inițială v_o este astfel încât $k = 8a^2 C^2$, adică

$$v_o = \frac{1}{(2a)^2} \sqrt{\frac{k}{2}} = \frac{1}{r_o^2} \sqrt{\frac{k}{2}}.$$

Punctul material ajunge din A în O (și mișcarea încetează) după un interval de timp ce se obține din legea ariilor

$$4a^2 \int_0^q \cos^2 q \, dq = 2av_o,$$

adică

$$t = \frac{2a}{v_o} \int_0^q \cos^2 q \, dq = \frac{a}{v_o} \left(q + \frac{1}{2} \sin 2q \right)$$

și deci

$$t_{tot} = \frac{2a}{v_o} \int_0^{p/2} \cos^2 q \, dq = \frac{pa}{2v_o}.$$

Reciproc, se poate pune problema determinării traiectoriei punctului material sub acțiunea forței centrale

$$\mathbf{F}(r) = -\frac{8ma^2 C^2}{r^5} \mathbf{r},$$

știind datele inițiale la $t = 0$

$$r(0) = r_o = 2a, q(0) = q_o = 0, v(0) = v_o = \frac{1}{r_o^2} \sqrt{\frac{k}{2}}, \mathbf{a} = \angle(\mathbf{r}_o, \mathbf{v}_o) = \frac{p}{2},$$

unde $k = 8a^2 C^2$ și $C = 2av_o$.

Ecuația parametrică a traiectoriei se obține integrând ecuația Binet în care înlocuind expresia forței $F(r)$ se obține

$$\frac{d^2}{dq^2} \left(\frac{1}{r} \right) + \frac{1}{r} = \frac{2r_o^2}{r^3},$$

unde înmulțind cu $\frac{d}{dq} \left(\frac{1}{r} \right)$, rezultă

$$\frac{1}{2} \frac{d}{dq} \left(\left(\frac{d}{dq} \left(\frac{1}{r} \right) \right)^2 \right) + \frac{1}{2} \frac{d}{dq} \left(\left(\frac{1}{r} \right)^2 \right) = \frac{r_o^2}{2} \frac{d}{dq} \left(\left(\frac{1}{r} \right)^4 \right)$$

de unde, ținând seama de datele inițiale, se obține

$$\frac{1}{2} \left(\frac{d}{dq} \left(\frac{1}{r} \right) \right)^2 = \frac{r_o^2}{2} \left(\frac{1}{r^4} - \frac{1}{r_o^4} \right) - \frac{1}{2} \left(\frac{1}{r^2} - \frac{1}{r_o^2} \right) \Rightarrow \left(\frac{d}{dq} \left(\frac{1}{r} \right) \right)^2 = \frac{r_o^2}{r^4} - \frac{1}{r^2} \Rightarrow$$

$$\left(-\frac{1}{r^2} \frac{dr}{dq}\right)^2 = \frac{r_o^2}{r^4} - \frac{1}{r^2} \Rightarrow \left(\frac{dr}{dq}\right)^2 = r_o^2 - r^2 \Rightarrow \frac{dr}{dq} = -\sqrt{r_o^2 - r^2} \Rightarrow$$

$$-\int_{r_o}^r \frac{dr}{\sqrt{r_o^2 - r^2}} = q \Rightarrow q = \arccos\left(\frac{r}{r_o}\right) \Rightarrow r = r_o \cos q.$$

Deci, traiectoria este o circumferință de rază $\frac{r_o}{2}$.

Același rezultat se putea obține și pornind de la integrala primă a energiei.

Dacă $v_o \neq \frac{1}{r_o^2} \sqrt{\frac{k}{2}}$ pentru $\angle(\mathbf{r}_o, \mathbf{v}_o) = \frac{p}{2}$, atunci punctul traiectoria punctului nu mai este circulară.

Programul MATLAB următor cuprinde reprezentarea animată a mișcării pe circumferință.

```
rb = 0.05; ncadre = 288; M = moviein(ncadre);
for j=1:ncadre
    tc = j*pi/2/ncadre;
    syms z
    zc = solve(z+sin(2*z)-tc);
    qc = eval(zc); xc = 2*cos(qc)^2; yc = sin(2*qc);
    s = 0:pi/90:pi/2; x = 2*cos(s).^2; y = sin(2*s);
    u = 0:pi/72:2*pi; xb = rb*cos(u); yb = rb*sin(u);
    plot(x,y,'-b','LineWidth',2);
    hold on
    plot(xc+xb,yc+yb,'-r','LineWidth',3);
    axis equal; axis([-0.1 2.1 -0.1 1.1])
    hold off
    M(j) = getframe;
end
movie(M)
```

Problema 5.13 (Mișcarea pe lemniscată): Un punct material se mișcă sub acțiunea unei forțe centrale pe lemniscata $r^2 = a \cdot \cos 2\theta$ (figura 5.15). Să se deducă expresia forței care determină această mișcare și relația ce trebuie să existe între datele inițiale și mărimea forței.

Soluție:

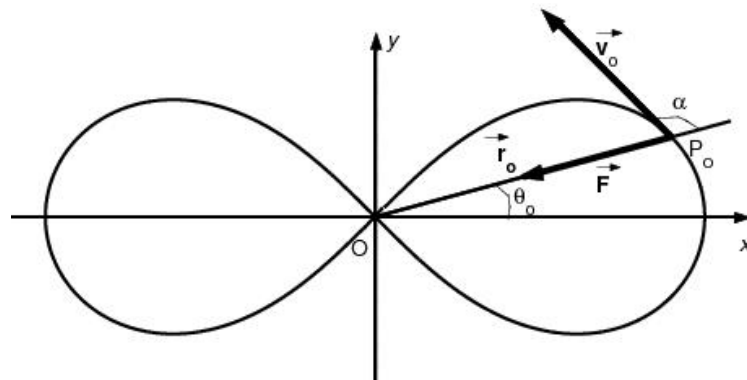


Figura 5.15: Mișcarea pe lemniscată.

Poziția inițială a punctului material este caracterizată prin

$$r_o = |\mathbf{r}_o| = \sqrt{a \cos 2q_o}, \quad q_o = \angle(\mathbf{r}_o, Ox), \quad v_o = |\mathbf{v}_o|,$$

unde pentru $a > 0$ avem $q_o \in \left(-\frac{p}{4}, \frac{p}{4}\right) \cup \left(\frac{3p}{4}, \frac{5p}{4}\right)$.

Viteza inițială fiind \mathbf{v}_o , unghiul ei cu \mathbf{r}_o se poate determina din relația

$$\cos a = \cos(\mathbf{r}_o, \mathbf{v}_o) = -\frac{2 \sin 2q_o}{\sqrt{1 + 3 \sin^2 2q_o}}.$$

ținând seama de reprezentarea parametrică a traiectoriei, din ecuația lui Binet rezultă

$$F(r) = -\frac{3ma^2 C^2}{r^7},$$

deci expresia forței centrale care determină această mișcare este

$$F(r) = -\frac{mk}{r^7},$$

unde $k = 3 a^2 C^2 > 0$ și $C = r_o v_o \sin a$.

Mișcarea sub acțiunea unei asemenea forțe are loc pe curba dată numai dacă viteza inițială a punctului material are valoarea

$$v_o = \sqrt{\frac{k}{3a^3 \cos 2q_o \sin^2 a}}.$$

Punctul material parcurge arcul $P_o P$ după un interval de timp ce se obține din legea ariilor

$$a q \cos 2q = r_o v_o \sin a,$$

adică

$$t = \frac{r_o v_o \sin a}{a} \int_{q_o}^q \cos 2q \, dq = \frac{r_o v_o \sin a}{2a} (\sin 2q - \sin 2q_o).$$

Punctul ajunge în O (și mișcarea încetează) după intervalul de timp

$$t_{tot} = \frac{r_o v_o \sin a}{a} \int_{q_o}^{p/4} \cos 2q \, dq = \frac{r_o v_o \sin a}{2a} (1 - \sin 2q_o).$$

Reciproc, se poate pune problema determinării traiectoria unui punct material sub acțiunea forței centrale atractive $F(r) = -\frac{mk}{r^7}$, cu condițiile inițiale la $t = 0$:

$$r(0) = r_o, \quad q(0) = q_o = 0, \quad v(0) = v_o = \frac{1}{r_o^3} \sqrt{\frac{k}{3}}, \quad a = \frac{p}{2}.$$

Prelucrând integrala primă a energiei

$$\frac{1}{2} m (\dot{r}^2 + r^2 \dot{q}^2) - U(r) = h,$$

unde $U(r) = \frac{mk}{6r^6}$ și $h = \frac{1}{2} m v_o^2 - \frac{mk}{6r_o^6} = 0$, se obține

$$\frac{1}{2}m(\dot{r}^2 + r^2\dot{\varphi}^2) - \frac{mk}{6r^6} = 0,$$

unde utilizând că $\dot{r} = \frac{dr}{dq}\dot{q}$ și din legea ariilor $\dot{\varphi} = \frac{C}{r^2}$ cu $C = r_0 v_0 = \frac{1}{r_0^2} \sqrt{\frac{k}{3}}$, rezultă

$$\left(\frac{dr}{dq}\right)^2 + r^2 = \frac{r^4}{C^2} \frac{k}{3r^6} \Rightarrow \left(\frac{dr}{dq}\right)^2 + r^2 = \frac{r^4}{C^2} \frac{3r_0^4 C^2}{3r^6} \Rightarrow$$

$$\left(\frac{dr}{dq}\right)^2 + r^2 = \frac{r_0^4}{r^2} \Rightarrow \left(r \frac{dr}{dq}\right)^2 = r_0^4 - r^4.$$

Notând $r^2 = r_0^2 w$, se obține ecuația

$$\frac{1}{4} \left(\frac{dw}{dq}\right)^2 = 1 - w^2.$$

Deoarece $r \leq r_0$, scade, deci $w \leq 1$ scade, rezultă ecuația $\frac{dw}{dq} = -2\sqrt{1-w^2}$, din care, ținând seama că $w(0) = 1$ și integrând pe $[0, t]$, se obține

$$\arccos w - \arccos 1 = 2(\theta - \theta_0),$$

adică, se obține $w = \cos 2\theta$. Deci, se obține ecuația traiectoriei

$$r^2 = r_0^2 \cos 2q,$$

ceea ce înseamnă că traiectoria este o lemniscată.

Traiectoria punctului material sub acțiunea forței centrale atractive de tipul $F(r) = -\frac{mk}{r^7}$ pentru o viteză inițială $v_0 \neq \frac{1}{r_0^3} \sqrt{\frac{k}{3}}$ pentru $a = \frac{p}{2}$, nu mai este o lemniscată.

Programul următor cuprinde reprezentarea animată a mișcării pe lemniscată.

```

a = 1; q0 = 0; tfinal = 1; rb = 0.01;
ncadre = 300; M = moviein(ncadre);
for j=1:ncadre
    tc = j*tfinal/ncadre; qc = asin(tc)/2;
    xc = cos(qc)*sqrt(a*cos(2*qc));
    yc = sin(qc)*sqrt(a*cos(2*qc));
    s = -pi/4:pi/180:pi/4;
    x=cos(s).*sqrt(a*cos(2*s)); y=sin(s).*sqrt(a*cos(2*s));
    u = 0:pi/72:2*pi; xb = rb*cos(u); yb = rb*sin(u);
    plot(x,y,'-b','LineWidth',2);
    hold on
    plot(xc+xb,yc+yb,'-r','LineWidth',4)
    axis equal
    axis([-0.05 sqrt(a)+0.05-sqrt(a)/2 sqrt(a)/2])
    axis off
    hold off
    M(j) = getframe;
end
movie(M)

```

Problema 5.14 (Mișcarea pe spirala lui Arhimede): *Un punct material se mișcă sub acțiunea unei forțe centrale pe spirala logaritmică $r = ae^{lq}$ (figura 5.16). Să se deducă expresia forței care determină această mișcare și relația ce trebuie să existe între datele inițiale și mărimea forței.*

Soluție:

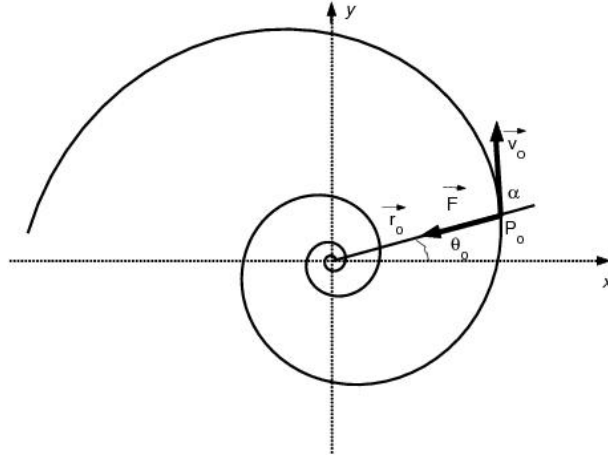


Figura 5.16: Mișcarea pe spirala logaritmică.

Poziția inițială a punctului material este caracterizată prin

$$r_o = |\mathbf{r}_o| = ae^{lq_o}, \quad q_o, v_o = |\mathbf{v}_o|.$$

Viteza inițială fiind \mathbf{v}_o , unghiul ei cu \mathbf{r}_o rezultă din

$$\cos \alpha = \cos(\mathbf{r}_o, \mathbf{v}_o) = \frac{l}{\sqrt{l^2 + 1}}.$$

ținând seama de reprezentarea parametrică a traiectoriei, din ecuația lui Binet rezultă

$$F(r) = -\frac{mC^2(l^2 + 1)}{r^3},$$

deci expresia forței centrale care determină această mișcare este

$$F(r) = -\frac{mk}{r^3},$$

unde $k = C^2(l^2 + 1) > 0$ și $C = r_o v_o \sin \alpha$.

Mișcarea sub acțiunea unei asemenea forțe are loc pe curba dată numai dacă viteza inițială a punctului material are valoarea

$$v_o = \frac{\sqrt{k}}{r_o}.$$

Punctul material parcurge arcul P_oP după un interval de timp ce se obține din legea ariilor

$$a^2 \frac{d^2 q}{dt^2} = \sqrt{\frac{k}{l^2 + 1}},$$

de unde

$$t = \frac{a^2}{2} \sqrt{\frac{l^2 + 1}{k}} \left(e^{2lq} - e^{2lq_0} \right)$$

Reciproc, se pune problema determinării traiectoriei unui punct material sub acțiunea forței centrale $F(r) = -\frac{mk}{r^3}$, cu condițiile inițiale la $t = 0$ date prin

$$r(0) = r_0, q(0) = q_0 = 0, v_0 = \frac{\sqrt{k}}{r_0}, a.$$

Ecuția lui Binet

$$\frac{d^2}{dq^2} \left(\frac{1}{r} \right) + \frac{1}{r} = \frac{k}{C^2} \frac{1}{r}$$

se scrie sub forma

$$\frac{d^2}{dq^2} \left(\frac{1}{r} \right) - \text{ctg}^2 a \frac{1}{r} = 0$$

și admite soluția

$$\frac{1}{r} = A e^{q \text{ctg} a} + B e^{-q \text{ctg} a}.$$

Din condițiile inițiale pentru constantele A și B se obține

$$A = 0, B = \frac{1}{r_0} \text{ctg} a,$$

deci

$$r = r_0 e^{q \text{ctg} a},$$

adică, traiectoria este o spirală logaritmică.

Traectoria punctului material sub acțiunea forței centrale $F(r) = -\frac{mk}{r^3}$ pentru o viteză inițială $v_0 \neq \frac{\sqrt{k}}{r_0}$ nu mai este o spirală logaritmică.

Programul următor cuprinde reprezentarea animată a mișcării pe spirala logaritmică.

```

a = 0.01; lb = 0.2; q0 = 4*pi;
ttot = exp(14*lb*pi) - exp(2*lb*q0);
rb = 0.01; ncadre=300; M=moviein(ncadre)
for j = 1:ncadre
    tc = j*ttot/ncadre; qc = log(tc+exp(2*lb*q0))/2/lb;
    xc = a*exp(lb*qc)*cos(qc); yc = a*exp(lb*qc)*sin(qc);
    s = 0:0.005:7*pi*1.01;
    x = a*exp(lb*s).*cos(s); y = a*exp(lb*s).*sin(s);
    u = 0:pi/72:2*pi; xb = rb*cos(u); yb = rb*sin(u);
    plot(x,y,'-b','LineWidth',2); hold on
    plot(xc+xb,yc+yb,'-r','LineWidth',3); axis equal
    axis([-0.9 0.5 -0.4 0.65])
    hold off
    M(j) = getframe;
end
movie(M)

```


Problema 5.15 (Oscilatorul anizotrop - Curbe Lyssajous): Să se studieze mișcarea unui punct material supus unei forțe $\mathbf{F} = -\mathbf{C}\mathbf{x}$, \mathbf{C} fiind un tensor de ordinul al doilea simetric pozitiv definit.

Soluție:

Ecuția de mișcare este

$$m\ddot{\mathbf{x}} = -\mathbf{C}\mathbf{x}.$$

Fie $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$ direcțiile principale ale tensorului \mathbf{C} și k_1, k_2, k_3 valorile principale cu $k_j > 0, j = 1, 2, 3$. În baza formată din direcțiile principale avem

$$\mathbf{C} = \sum_{j=1}^3 k_j \mathbf{m}_j \otimes \mathbf{m}_j \quad \text{și} \quad \mathbf{C}\mathbf{x} = \sum_{j=1}^3 k_j x^j \mathbf{m}_j.$$

Deci ecuațiile scalare de mișcare sunt

$$m\ddot{x}^j = -k_j x^j, \quad j = 1, 2, 3, \quad \text{sau} \quad \ddot{x}^j + w_j^2 x^j = 0, \quad \text{cu} \quad w_j^2 = \frac{k_j}{m}, \quad j = 1, 2, 3,$$

și admit soluțiile

$$x^j = x_o^j \cos w_j t + \frac{\mathfrak{x}_o^j}{w_j} \sin w_j t, \quad j = 1, 2, 3, \quad \text{sau} \quad x^j = A^j \cos(w_j t - a_j), \quad j = 1, 2, 3,$$

$$\text{unde } A^j = \sqrt{(x_o^j)^2 + \left(\frac{\mathfrak{x}_o^j}{w_j}\right)^2}, \quad \cos a_j = \frac{x_o^j}{A^j} \quad \text{și} \quad \sin a_j = \frac{\mathfrak{x}_o^j}{w_j A^j} \quad \text{pentru } j = 1, 2, 3, \text{ iar}$$

x_o^j și \mathfrak{x}_o^j sunt din condițiile inițiale

$$\mathbf{x}(0) = \mathbf{x}_o = \sum_{j=1}^3 x_o^j \mathbf{m}_j \quad \text{și} \quad \dot{\mathbf{x}}(0) = \dot{\mathbf{x}}_o = \sum_{j=1}^3 \mathfrak{x}_o^j \mathbf{m}_j.$$

Forma soluțiilor indică faptul că mișcarea este mărginită, traiectoriile sunt cuprinse în interiorul unui paralelipiped, centrat în origine, cu laturile $2A^1, 2A^2, 2A^3$. Mișcarea este periodică numai dacă există numerele naturale n_1, n_2, n_3 astfel încât

$$\frac{n_1}{w_1} = \frac{n_2}{w_2} = \frac{n_3}{w_3}.$$

Curbele obținute se numesc **curbe Lyssajous**.

Pentru cazul bidimensional, în figura 5.17 sunt reprezentate traiectoriile realizate prin programul următor:

```
t = 0:.01:2*pi; y = cos(t);
x1 = 2*cos(2*t); x2 = 2*cos(2*t-pi/4); x3 = 2*cos(2*t-pi/2);
x4 = 2*cos(3*t); x5 = 2*cos(3*t-pi/3); x6 = 2*cos(3*t-pi/2);
x7 = 2*cos(3*t-2*pi/3); x8 = 2*cos(4*t); x9 = 2*cos(4*t-pi/4);
x10 = 2*cos(4*t-pi/2); x11 = 2*cos(7*t); x12 = 2*cos(7*t-0.4);
y12 = cos(3*t);
subplot(4,4,1); plot(x1,y); axis equal tight; title('2,1,0');
subplot(4,4,2); plot(x2,y); axis equal tight; title('2,1,\pi/4');
subplot(4,4,3); plot(x3,y); axis equal tight; title('2,1,\pi/2');
subplot(4,4,5); plot(x4,y); axis equal tight; title('3,1,0');
subplot(4,4,6); plot(x5,y); axis equal tight; title('3,1,\pi/3');
subplot(4,4,7); plot(x6,y); axis equal tight; title('3,1,\pi/2');
subplot(4,4,8); plot(x7,y); axis equal tight; title('3,1,2\pi/3');
subplot(4,4,9); plot(x8,y); axis equal tight; title('4,1,0');
subplot(4,4,10); plot(x9,y); axis equal tight; title('4,1,\pi/4');
subplot(4,4,11); plot(x10,y); axis equal tight; title('4,1,\pi/2');
```

```
subplot(4,4,13); plot(x11,y); axis equal tight; title('7,1,0');
subplot(4,4,14); plot(x12,y12); axis equal tight; title('7,3,0.4');
set(findobj('Type','line'),'LineWidth',2)
```

În programul de mai sus am considerat câteva perechi de rapoarte ale frecvențelor și defazaje inițiale conform tabelului următor:

ω_1	2	2	2	3	3	3	3	4	4	4	7	7
ω_2	1	1	1	1	1	1	1	1	1	1	1	3
α_1	0	$\pi/4$	$\pi/2$	0	$\pi/3$	$\pi/2$	$2\pi/3$	0	$\pi/4$	$\pi/2$	0	0.4

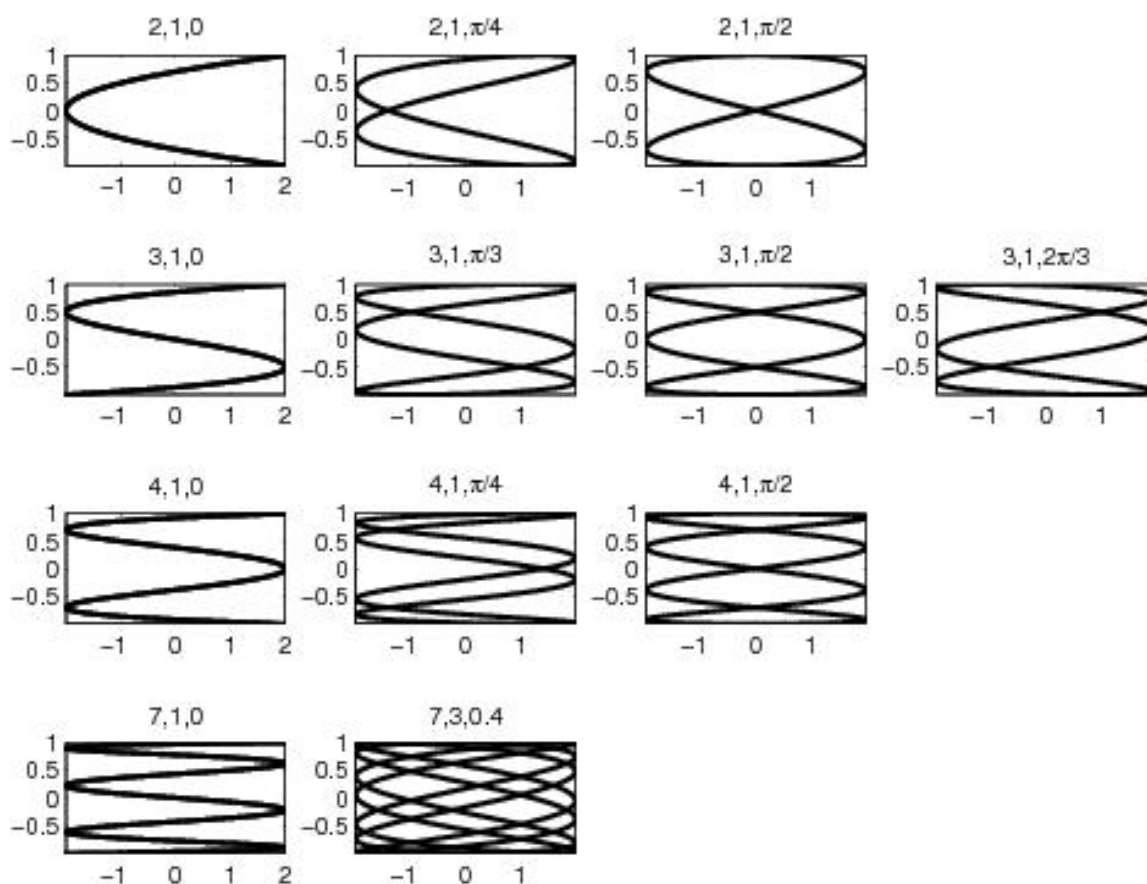


Figura 5.17: Exemple de curbe Lyssajous.

5.4. Mișcarea pe o curbă

Problema 5.16 (Pendulul matematic): Să se studieze mișcarea unui punct material greu, de masă m , pe o circumferință de rază l , în plan vertical. Se cere:

a) deducerea legii de mișcare a punctului material (legea lui Newton în proiecție pe axele referențialului inerțial cartezian, legea lui Newton în proiecție pe triedrul lui Frenet, teorema momentului cinetic, teorema energiei cinetice);

b) mărimea forței de legătură;

c) tipuri de mișcări ale punctului material greu pe circumferința verticală, discuție după condițiile inițiale.

Frecarea și rezistența mediului se neglijează.

Soluție:

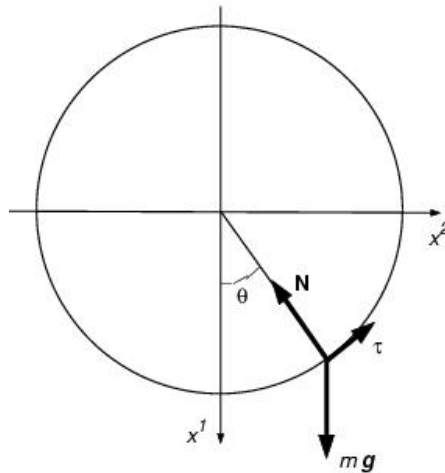


Figura 5.18: Pendulul matematic.

Rezolvarea teoretică a problemei presupune mai multe etape.

a) Deducerea legii de mișcare a punctului material.

Se consideră referențialul inerțial $Ox^1x^2x^3$, având originea în centrul circumferinței verticale, planul vertical al circumferinței fiind planul Ox^1x^2 , cu Ox^1 verticala descendentă (figura 5.18). Coordonatele punctului material trebuie să satisfacă restricțiile

$$\begin{cases} (x^1)^2 + (x^2)^2 = l^2, \\ x^3 = 0. \end{cases}$$

Asupra punctului material acționează forța de greutate $m\mathbf{g}$ și forța de legătură (reacțiunea) normală la circumferință \mathbf{N} , aflată în planul vertical al circumferinței.

Legea lui Newton în proiecție pe axele referențialului inerțial cartezian.

În acest caz, legea lui Newton se scrie astfel

$$m\mathbf{\ddot{x}} = m\mathbf{g} + \mathbf{N}$$

și prin proiecție pe direcțiile referențialului ales conduce la ecuațiile

$$\begin{cases} m\ddot{x}^1 = N^1 + mg, \\ m\ddot{x}^2 = N^2, \\ m\ddot{x}^3 = N^3. \end{cases}$$

Cum mișcarea are loc în plan vertical, $x^3 = 0$, și reacțiunea normală \mathbf{N} este în planul vertical al circumferinței, $N^3 = 0$, a treia ecuație scalară este adevărată. În continuare se va lucra numai în planul Ox^1x^2 .

Deoarece direcția reacțiunii este cunoscută, ecuațiile de mișcare se pot scrie sub forma

$$\begin{cases} m\ddot{x}^1 = -N\frac{x^1}{l} + mg, \\ m\ddot{x}^2 = -N\frac{x^2}{l}. \end{cases}$$

Trecând la coordonate polare

$$\begin{cases} x^1 = l \cos q, \\ x^2 = l \sin q, \end{cases}$$

ecuațiile de mișcare devin

$$\begin{cases} ml(-\ddot{q} \sin q - \dot{q}^2 \cos q) = -N \cos q + mg, \\ ml(\ddot{q} \cos q - \dot{q}^2 \sin q) = -N \sin q. \end{cases}$$

Prin eliminarea lui N , rezultă **ecuația pendulului matematic**

$$\ddot{q} + \frac{g}{l} \sin q = 0 \quad (5.16)$$

și mărimea algebrică a forței de legătură

$$N = mg \cos q + ml\dot{q}^2. \quad (5.17)$$

Pentru forța de legătură normală la traiectorie - în absența frecării - se putea scrie

$$N = l \frac{\text{grad}\left(\left(x^1\right)^2 + \left(x^2\right)^2 - l^2\right)}{\left\|\text{grad}\left(\left(x^1\right)^2 + \left(x^2\right)^2 - l^2\right)\right\|} + m \frac{\text{grad}\left(x^3\right)}{\left\|\text{grad}\left(x^3\right)\right\|} = l \left(\frac{x^1}{l} \mathbf{v}_1 + \frac{x^2}{l} \mathbf{v}_2 \right) + m \mathbf{v}_3,$$

deci $\lambda = N$, $\mu = 0$. Multiplicatorul N nu este o constantă, ci depinde de poziție (prin intermediul vitezei) (dacă N ar fi fost constant, ecuațiile diferențiale în coordonate carteziane ar fi fost liniare, ușor integrabile, în contradicție cu ceea ce se întâmplă cu ecuația pendulului matematic).

Legea lui Newton în proiecție pe triedrul lui Frenet.

Legea lui Newton în proiecție pe triedrul lui Frenet $(\boldsymbol{\tau}, \mathbf{v}, \boldsymbol{\beta})$ conduce la ecuațiile

$$\begin{cases} m\ddot{q} = -mg \sin q, \\ m\frac{v^2}{l} = -mg \cos q + N. \end{cases}$$

Deoarece $\boldsymbol{\beta} = \mathbf{v}_3$ sunt doar două ecuații scalare.

Având în vedere că

$$v = l\dot{q},$$

se obține imediat ecuația de mișcare (5.16) și mărimea forței de legătură (5.17).

Teorema momentului cinetic.

Momentul cinetic al punctului material este în acest caz

$$\mathbf{K}_O = \mathbf{x} \times m\mathbf{v} = l\mathbf{r} \times ml\dot{q}\mathbf{v}_q = ml^2\dot{q}\mathbf{v}_z,$$

iar momentul rezultat al forțelor dată și de legătură

$$\mathbf{M}_O = \mathbf{x} \times (m\mathbf{g} + \mathbf{N}) = \mathbf{x} \times m\mathbf{g} = -mgl \sin q \mathbf{v}_z.$$

Teorema momentului cinetic

$$\frac{d\mathbf{K}_O}{dt} = \mathbf{M}_O$$

conduce direct la ecuația de mișcare (5.16).

Teorema energiei cinetice.

Energia cinetică a punctului material este

$$T = \frac{1}{2}mv^2 = \frac{1}{2}ml^2\dot{q}^2,$$

iar puterea mecanică a forțelor dată și de legătură

$$P = \mathbf{v} \cdot (m\mathbf{g} + \mathbf{N}) = \mathbf{v} \cdot m\mathbf{g} = mg\dot{q} = -mgl\dot{q}\sin q.$$

Teorema energiei cinetice

$$\frac{dT}{dt} = P$$

conduce, din nou, direct, la ecuația de mișcare (6.16).

Prin teorema momentului cinetic și prin teorema energiei se obține direct ecuația de mișcare, dar nu mai sunt informații despre forța de legătură.

b) Ecuația de mișcare (5.16) admite integrala primă

$$\frac{1}{2}ml^2\dot{q}^2 - mgl \cos q = \text{const.},$$

numită *integrala primă a conservării energiei totale*, constanta fiind determinată din condițiile inițiale.

c) Tipuri de mișcări ale punctului material greu pe circumferința verticală.

Caracteristicile mișcării punctului material greu pe circumferința verticală vor fi determinate de condițiile inițiale impuse la $t = 0$

$$\mathbf{x}(0) = \mathbf{x}_o, \dot{\mathbf{x}}(0) = \mathbf{v}_o \quad \text{sau} \quad q(0) = q_o, \dot{q}(0) = \dot{q}_o. \quad (5.18)$$

1. Mici oscilații.

Dacă poziția inițială și viteza inițială sunt astfel încât punctul material să rămână în vecinătatea poziției de echilibru stabile $q = 0$, atunci ecuația de mișcare se poate liniariza

$$\sin q; q \Rightarrow \ddot{q} + \frac{g}{l}q = 0. \quad (5.19)$$

Soluția ecuației de mișcare liniarizate (5.19), cu condițiile inițiale (5.18), este

$$q = q_o \cos\left(t\sqrt{\frac{g}{l}}\right) + \dot{q}_o \sin\left(t\sqrt{\frac{g}{l}}\right) \quad \text{sau} \quad q = \sqrt{q_o^2 + \frac{l}{g}\dot{q}_o^2} \cos\left(t\sqrt{\frac{g}{l}} + a\right),$$

unde

$$\cos a = \frac{q_o}{\sqrt{q_o^2 + \frac{l}{g}\dot{q}_o^2}}, \quad \sin a = \frac{\sqrt{\frac{l}{g}\dot{q}_o^2}}{\sqrt{q_o^2 + \frac{l}{g}\dot{q}_o^2}}.$$

Condiția ca liniarizarea să aibă sens ($\theta < \theta_* = 5^\circ$) este ca

$$q_o^2 + \frac{l}{g}\dot{q}_o^2 \leq q_*^2.$$

Liniarizarea nu are sens în vecinătatea poziției de echilibru $\theta = \pi$, deoarece este poziție de echilibru instabilă.

Integrala primă de conservare a energiei totale, cu condițiile inițiale considerate, se explicitează astfel

$$l^2 \dot{\varphi}^2 = 2gl \cos \varphi + h, \quad h = v_o^2 - 2gx_o^1.$$

Fie x_A^1 o abscisă definită prin

$$x_A^1 = x_o^1 - \frac{v_o^2}{2g},$$

astfel încât integrala primă se scrie sub forma

$$l^2 \dot{\varphi}^2 = 2g(l \cos \varphi - x_A^1).$$

Această relație arată că în cursul mișcării

$$x_A^1 \leq x^1 \leq l.$$

2. Cazul $-l < x_A^1$ (mișcare oscilatorie).

În acest caz există pe circumferință două puncte A și A' cu abscisa x_A^1 și se poate nota

$$x_A^1 = l \cos a, \quad 0 < a < \pi$$

și ecuația de mișcare devine

$$l \dot{\varphi} = \pm \sqrt{2gl(\cos \varphi - \cos a)}.$$

înlocuind

$$\cos \varphi = 1 - 2 \sin^2 \frac{\varphi}{2}, \quad \cos a = 1 - 2 \sin^2 \frac{a}{2},$$

se obține ecuația diferențială

$$\frac{d\varphi}{dt} = \pm 2 \sqrt{\frac{g}{l}} \sqrt{\sin^2 \frac{a}{2} - \sin^2 \frac{\varphi}{2}}.$$

Fie $\dot{\varphi}_o > 0$. Din ecuația de mai sus, în care se alege semnul +, rezultă că φ va crește până la valoarea a când $\dot{\varphi}$ se anulează. Luând apoi semnul - în fața radicalului, se vede că φ va descreește până la valoarea $-a$, când iarăși se anulează etc.

Intervalul de timp în care φ și $\dot{\varphi}$ revin la aceeași valoare este perioada de oscilație

$$t_o = 2 \sqrt{\frac{l}{g}} \int_{-a}^a \left(\sin^2 \frac{a}{2} - \sin^2 \frac{\varphi}{2} \right)^{-\frac{1}{2}} d\left(\frac{\varphi}{2}\right)$$

Notând

$$\sin \frac{\varphi}{2} = u \sin \frac{a}{2},$$

se exprimă t_o sub forma integralei eliptice

$$t_o = 4 \sqrt{\frac{l}{g}} \int_0^1 \frac{du}{\sqrt{(1-u^2)(1-k^2 u^2)}}$$

(forma normală Legendre), unde parametrul $k^2 = \sin^2 \frac{a}{2} < 1$, depinde de condițiile inițiale.

Pentru calculul numeric al lui τ_o se poate folosi dezvoltarea în serie

$$\frac{1}{\sqrt{1-k^2u^2}} = 1 + \frac{1}{2}k^2u^2 + \frac{1 \cdot 3}{2 \cdot 4}k^4u^4 + \mathbf{L} + \frac{1 \cdot 3 \cdot 5 \cdot \mathbf{L} \cdot (2n-1)}{2 \cdot 4 \cdot 6 \cdot \mathbf{L} \cdot 2n}k^{2n}u^{2n} + \mathbf{L},$$

valabilă pentru $k^2u^2 < 1$, urmată de integrarea termen cu termen a seriei uniform convergente. Dar

$$\int_0^1 \frac{u^{2n} du}{\sqrt{1-u^2}} = \frac{1 \cdot 3 \cdot 5 \cdot \mathbf{L} \cdot (2n-1) p}{2 \cdot 4 \cdot 6 \cdot \mathbf{L} \cdot 2n \cdot 2}$$

și prin urmare

$$t_o = 2p \sqrt{\frac{l}{g}} \left(1 + \frac{1}{2^2} \sin^2 \frac{a}{2} + \mathbf{L} + \left(\frac{1 \cdot 3 \cdot 5 \cdot \mathbf{L} \cdot (2n-1)}{2 \cdot 4 \cdot 6 \cdot \mathbf{L} \cdot 2n} \right)^2 \sin^{2n} \frac{a}{2} + \mathbf{L} \right)$$

În cazul oscilațiilor mici

$$\sin \frac{a}{2}; \frac{a}{2} \quad \text{deci} \quad t_o = 2p \sqrt{\frac{l}{g}} \left(1 + \frac{1}{16} a^2 \right)$$

Dacă și a^2 este neglijabil, atunci se regăsește formula lui Galileu

$$t_o = 2p \sqrt{\frac{l}{g}}$$

3. Cazul $x_A^1 < -l$ (mișcare de rotație).

În acest caz

$$v_o^2 > 2g(x_o^1 + l)$$

și punctul descrie complet circumferința în același sens, impus de semnul lui \dot{q}_o . Ecuația diferențială care descrie mișcarea este

$$l\ddot{q} = \pm \sqrt{2g(l - x_A^1) \left(1 - k_1^2 \sin^2 \frac{q}{2} \right)},$$

unde

$$k_1^2 = \frac{2l}{l - x_A^1} < 1.$$

Intervalul de timp în care punctul plecat din poziția q_o revine în aceeași poziție cu $q = q_o + 2p$ este dat de

$$t_o = \frac{l}{\sqrt{2g(l - x_A^1)}} \int_{q_o}^{q_o+2p} \frac{dq}{\sqrt{1 - k_1^2 \sin^2 \frac{q}{2}}} = \frac{2l}{\sqrt{2g(l - x_A^1)}} \int_0^1 \frac{du}{\sqrt{(1-u^2)(1-k_1^2u^2)}}.$$

4. Cazul $x_A^1 = -l$ (mișcare asimptotică).

Pentru realizarea acestui caz, pur teoretic, ar trebui ca

$$v_o^2 = 2g(x_o^1 + l)$$

În această ipoteză, se aplică formulele din cazul 2., cu $a = +p$. Alegând, de exemplu, semnul +

$$l\ddot{q} = \sqrt{2gl(1 + \cos q)}$$

și prin integrare, după separarea variabilelor

$$t = \sqrt{\frac{l}{g}} \int_{q_0}^q \frac{d\left(\frac{q}{2}\right)}{\cos\left(\frac{q}{2}\right)} = \ln\left(\operatorname{tg}\left(\frac{q}{4} + \frac{p}{4}\right)\right)_{q_0}^q.$$

Punctul material se mișcă în sensul în care q crește și ajunge în punctul $q = p$ într-un timp infinit

$$\lim_{q \rightarrow p} t(q) = +\infty.$$

Dependența cazurilor analizate de condițiile inițiale $\{q_0, v_0\}$ se poate prezenta grafic, pe abscisă fiind poziția inițială q_0 , pe ordonată raportul vitezei inițiale la

viteza critică $\frac{v_0}{\sqrt{gl}}$, iar cota reprezentând raportul $\frac{x_A^1}{l}$. Pe suprafața astfel obținută s-a

trasat curba corespunzând cazului mișcărilor asimptotice în figura 5.19 folosind programul:

```
[X,Y] = meshgrid(-pi:pi/25:pi, -1.5:0.1:1.5);
Z = cos(X)-Y.^2/2; meshz(X,Y,Z)
hold on
x = -pi:pi/50:pi; y1 = sqrt(cos(x)+1); z1 = cos(x)-y1.^2/2;
y2 = -sqrt(cos(x)+1); z2 = cos(x)-y2.^2/2;
plot3(x,y1,z1,'-k',x,y2,z2,'-k','LineWidth',2)
xlabel('\theta','FontSize',14),
ylabel('v_0/sqrt(g*l)','FontSize',14),zlabel('x_1_A',...
'FontSize',14)
title('Studiul conditiilor initiale','FontSize',14)
text(0,-1,1,'\leftarrow miscari oscilatorii','FontSize',14);
text(-1,-1.5,0,'\leftarrow miscari asimptotice',...
'FontSize',14);
text(-1.8,-1.5,-1,'\leftarrow miscari derotatie',...
'FontSize',14); hold off
```

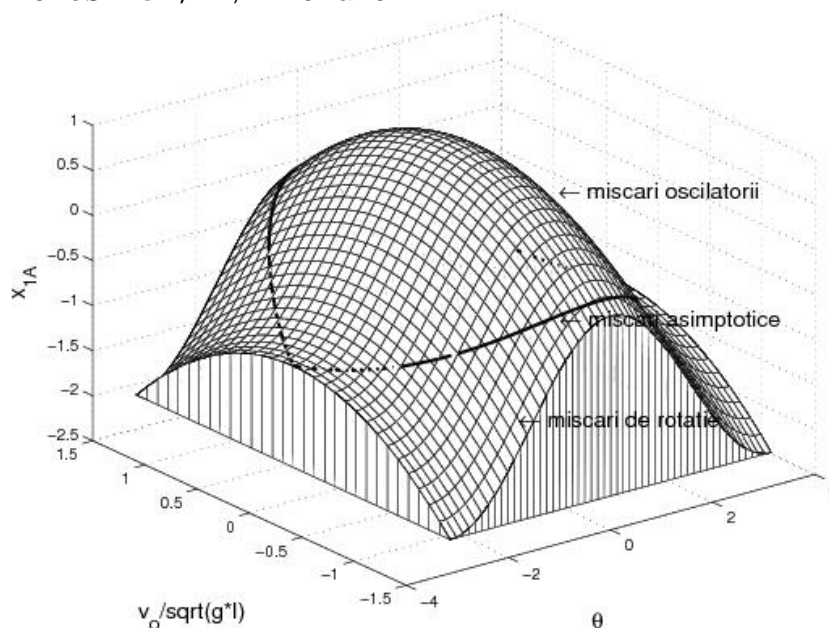


Figura 5.19: Studiul condițiilor inițiale.

Integrarea ecuației pendului matematic (5.16) nu se poate face analitic. Încercând acest lucru în prin apelarea funcției `dsolve`

```
v = dsolve('D2v+sin(v)', 't')
```

conduce la răspunsul

```
Warning: Explicit solution could not be found;
implicit solution returned.
v = [ Int(1/(2*cos(a)+C1)^(1/2), a=`..v)-t-C2=0, ...
      -Int(1/(2*cos(a)+C1)^(1/2), a=`..v)-t-C2=0]
```

Forma implicită a soluției reprezintă de fapt integrala primă a energiei. În acest caz vom integra numeric ecuația (5.16) asociind prin notațiile

$$q_1 = q, q_2 = \dot{q},$$

sistemul cvasiliniar de ordinul întâi

$$\begin{cases} \dot{q}_1 = q_2, \\ \dot{q}_2 = -\frac{g}{l} \sin q_1. \end{cases}$$

Pentru integrarea numerică am folosit funcția `pendmat`

```
function dqdt = pendmat(t,q)
    dqdt = [q(2); -sin(q(1))];
end
```

și programul următor

```
k1 = 0.5; per1 = 4*ellipke(k1)
k2 = 0.6; per2 = 4*ellipke(k2)
k3 = 0.8; per3 = 4*ellipke(k3)
[t1,q1] = ode45(@pendmat,[0 3*per3],[0; 2*k1]);
[t2,q2] = ode45(@pendmat,[0 3*per3],[0; 2*k2]);
[t3,q3] = ode45(@pendmat,[0 3*per3],[0; 2*k3]);
s = 0:0.01:3.1*per3;
figure(1)
plot(t1,q1(:,1),'k-',t2,q2(:,1),'k-',t3,q3(:,1),'k-',...
      s,0,'k-','LineWidth',2)
axis([0 30 -2.5 2.5])
xlabel('t*sqrt(g/l)'),ylabel('\theta')
gtext('1','FontSize',14), gtext('2','FontSize',14),
gtext('3','FontSize',14),
text(12.5,2,'\theta_0=0','FontSize',14)
text(1,-2.25,['1 \rightarrow \tau_1=',num2str(per1),...
              ' 2 \rightarrow \tau_2=',num2str(per2),...
              ' 3 \rightarrow \tau_3=',num2str(per3)],'FontSize',14)
title('Variatia unghiului \theta in functie de viteza
initiala','FontSize',14)
figure(2)
q40 = pi/6;
k4 = sqrt((1-cos(q40))/2); per4 = 4*ellipke(k4) q50 = pi/4;
k5 = sqrt((1-cos(q50))/2); per5 = 4*ellipke(k5) q60 = pi/3;
k6 = sqrt((1-cos(q60))/2); per6 = 4*ellipke(k6)
[t4,q4] = ode45(@pendmat,[0 3*per6],[q40; 0]);
[t5,q5] = ode45(@pendmat,[0 3*per6],[q50; 0]);
[t6,q6] = ode45(@pendmat,[0 3*per6],[q60; 0]);
```

```

s = 0:0.01:3.1*per3;
plot(t4,q4(:,1),'k-',t5,q5(:,1),'k-',t6,q6(:,1),'k-',...
      s,0,'k-','LineWidth',2)
xlabel('t*sqrt(g/l)'), ylabel('\theta')
gtext('4','FontSize',14), gtext('5','FontSize',14),
gtext('6','FontSize',14)
text(12.5,1.25,'(d\theta/dt)_o=0','FontSize',14)
text(1,-1.25,['4 \rightarrow \tau_4=',num2str(per4),...
              ', 5 \rightarrow \tau_5=',num2str(per5),...
              ', 6 \rightarrow \tau_6=',num2str(per6)],'FontSize',14)
title('Variatia unghiului \theta in functie de pozitia ...
      initiala','FontSize',14)

```

Reprezentările grafice obținute cu acest program sunt prezentate în figurile 5.20 și 5.21.

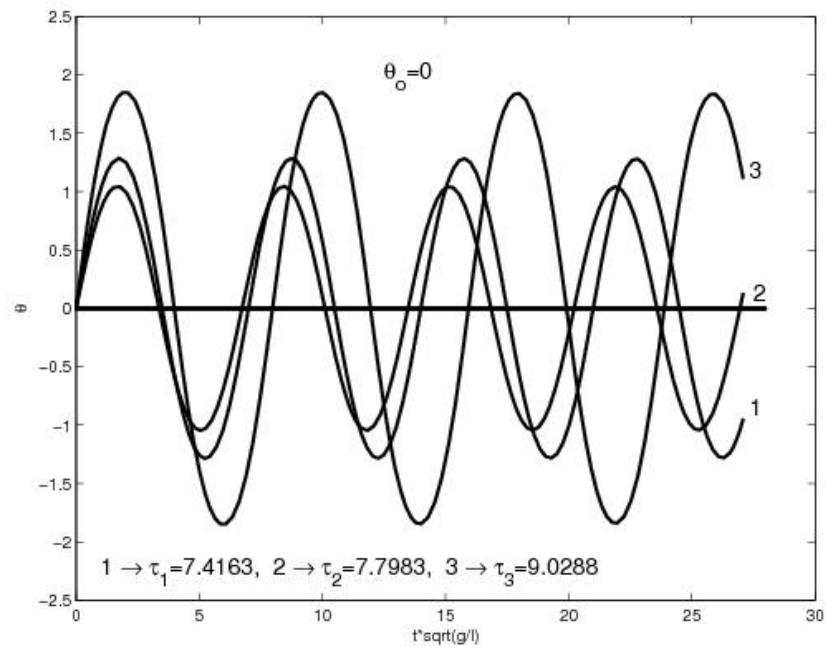


Figura 5.20: Variația unghiului q în funcție de viteza inițială.

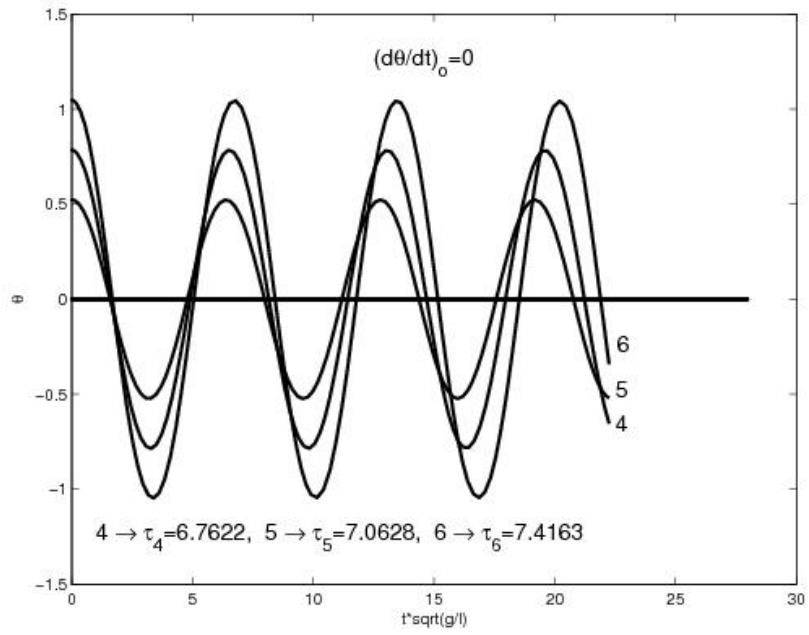


Figura 5.21: Variația unghiului q în funcție de poziția inițială.

În figura 5.22 este reprezentată variația unghiului q pentru diverse viteze inițiale, iar în figura 5.23 este reprezentată comparația între soluția numerică și soluția analitică simplificată.

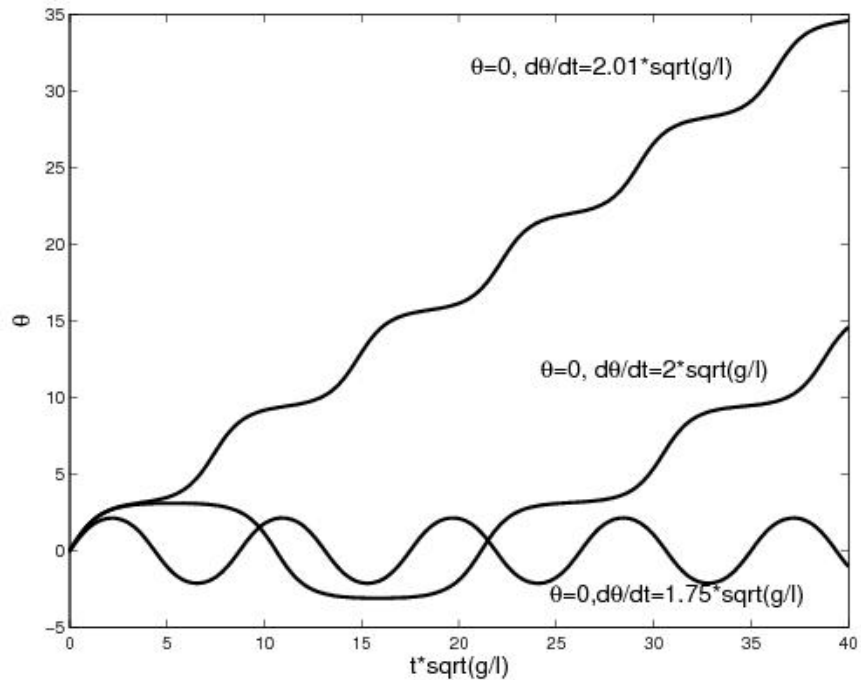


Figura 5.22: Variația unghiului q în funcție viteza inițială.

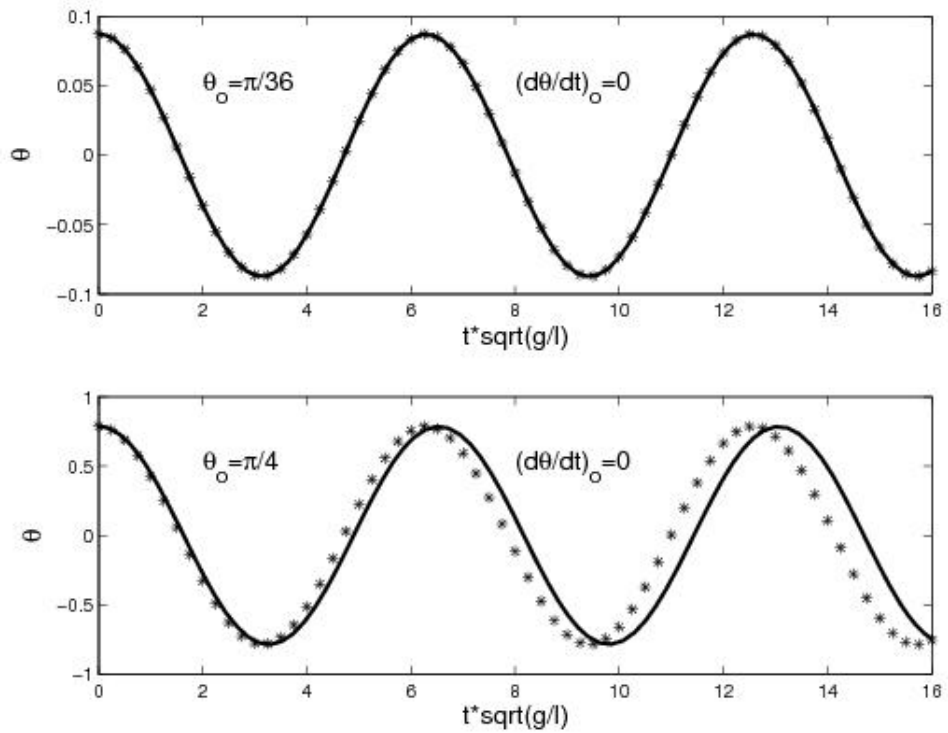


Figura 5.23: Comparație soluție analitică și soluție numerică.

Problema 5.17 (Pendulul cicloidal): Să se studieze mișcarea fără frecare a unui punct material greu, de masă m , pe o cicloidă cu baza orizontală, situată în plan vertical, cu concavitatea în sus.

Soluție:

Se alege un reper $Ox^1x^2x^3$ cu originea în punctul cel mai de jos, Ox^1x^2 planul vertical cu axa x^2 verticală ascendentă. Fie a raza cercului generator.

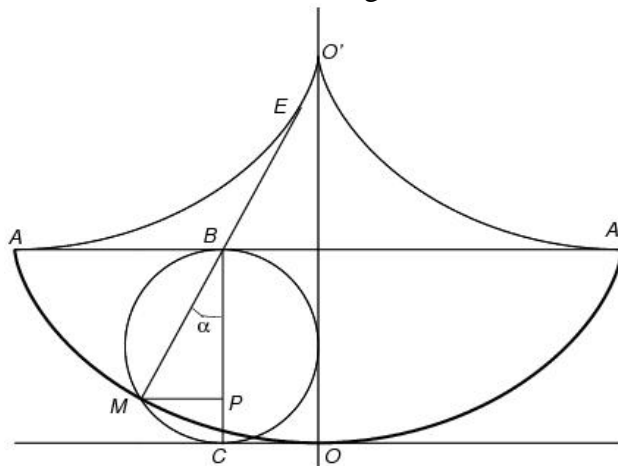


Figura 5.24: Pendul cicloidal.

Precizăm câteva proprietăți geometrice ale cicloidei (figura 5.24): fie o poziție a cercului generator cu M punctul curent de pe cicloidă; normala la curbă este MB , centrul de curbura în E , simetricul lui M față de B ; locul geometric al lui E este o

cicloida egală cu cicloida considerată, având vârfurile în A și A' ; tangenta MC este jumătate din arcul $\overset{\frown}{OM}$, notat cu s . În triunghiul dreptunghic BMC avem

$$MC^2 = BC \cdot CP,$$

adică

$$\frac{s^2}{4} = 2ax^2 \quad \text{de unde rezulta} \quad \frac{dx^2}{ds} = \frac{s}{4a}.$$

Reprezentarea parametrică a cicloidei este dată prin ecuațiile

$$\begin{cases} x^1 = a(q + \sin q), \\ x^2 = a(1 - \cos q), \end{cases}$$

cu $q \in [-p, p]$.

Proiecția greutatei pe tangentă este $-mg \frac{dx^2}{ds} = -mg \frac{s}{4a}$. Deci ecuația de mișcare ~~$m\ddot{x} = mg + N$~~ ,
 $m\ddot{x} = mg + N$,

în proiecție pe tangentă, va fi

$$\frac{d^2s}{dt^2} = \frac{g}{4a}s.$$

Soluția acestei ecuații este

$$s = s_0 \cos\left(t \sqrt{\frac{g}{4a}}\right)$$

dacă mobilul este lăsat liber (viteză inițială nulă) din poziția inițială $s(0) = s_0$. Timpul necesar mobilului să ajungă în O este independent de poziția inițială

$$t_* = p \sqrt{\frac{a}{g}},$$

deci mișcarea este tautocronă.

Reacțiunea normală rezultă din proiecția ecuației de mișcare pe direcția normalei principale

$$m \frac{v^2}{r} = mg \cdot \mathbf{v} + N$$

cu

$$v^2 = 2g(x_0^2 - x^2), \quad r = 2MB, \quad mg \cdot \mathbf{v} = -mg \cos a = -mg \frac{2a - x^2}{MB},$$

adică

$$N = mg \frac{x_0^2 - x^2}{MB} + mg \frac{2a - x^2}{MB} = \frac{mg(x_0^2 - 2x^2 + 2a)}{MB}.$$

Dacă $x_0^2 = 2a$, atunci

$$N = \frac{mg(4a - 2x^2)}{MB} = -2mg \cdot \mathbf{v}.$$

În programele următoare sunt prezentate animații pentru mișcarea tautocronă a punctului material greu pe cicloida și pendulul cicloidal.

```

r = 0.05; per = pi/2;
th01 = pi/9; th02 = pi/3; th03 = pi/4;
th04 = pi/6; th05 = pi/2; ncadre = 100;
M = moviein(ncadre);
for j = 1:ncadre+1
    t = (j-1)*per/ncadre;
    u = 0:pi/100:pi; v = 0:pi/60:2*pi;
    cx = u+sin(u); cy = 1-cos(u); u1 = 2*asin(sin(th01)*cos(t));
    xb1 = u1+sin(u1)+r*cos(v); yb1 = 1-cos(u1)+r*sin(v);
    u2 = 2*asin(sin(th02)*cos(t));
    xb2 = u2+sin(u2)+r*cos(v); yb2 = 1-cos(u2)+r*sin(v);
    u3 = 2*asin(sin(th03)*cos(t));
    xb3 = u3+sin(u3)+r*cos(v); yb3 = 1-cos(u3)+r*sin(v);
    u4 = 2*asin(sin(th04)*cos(t));
    xb4 = u4+sin(u4)+r*cos(v); yb4 = 1-cos(u4)+r*sin(v);
    u5 = 2*asin(sin(th05)*cos(t));
    xb5 = u5+sin(u5)+r*cos(v); yb5 = 1-cos(u5)+r*sin(v);
    plot(cx,cy,'-k','LineWidth',2)
    hold on
    plot(xb1,yb1,'-m',xb2,yb2,'-c',xb3,yb3,'-r',...
        xb4,yb4,'-g',xb5,yb5,'-b','LineWidth',4);
    axis equal tight; axis([-0.05 pi -0.05 2])
    hold off
    M(j) = getframe;
end
movie(M)

```

```

r = 0.05; per = 2*pi; th0 = pi/4;
ncadre = 100;
M = moviein(ncadre);
for j = 1:ncadre+1
    t = (j-1)*per/ncadre;
    u = -pi:pi/100:pi; v = 0:pi/60:2*pi;
    cx = u+sin(u); cy = 1-cos(u);
    u = 2*asin(sin(th0)*cos(t));
    xb = u+sin(u)+r*cos(v); yb = 1-cos(u)+r*sin(v);
    plot(cx,cy,'-k','LineWidth',2);
    hold on
    plot(xb,yb,'r','LineWidth',4);
    axis equal tight; axis([-pi pi -0.05 2]);
    hold off
    M(j) = getframe;
end
movie(M,3)

```

Bibliografie

1. D. Arnold, B. Wagner, *Matlab Programming*, 2007, Department of Mathematics, College of the Redwoods, <http://msenux.redwoods.edu/Math4Textbook/>
2. V. Arnold, *Méthodes mathématiques de la mécanique classique*, Editura Mir, Moscova, 1976.
3. Ș. Bălan, *Culegere de probleme de mecanică*, Editura Didactică și Pedagogică, București, 1977.
4. P. Blaga, *Statistică prin...MATLAB*, Presa Universitară Clujeană, Cluj-Napoca, 2002.
5. N. Breaz, *Modele de regresie bazate pe funcții spline*, Presa Universitară Clujeană, 2007.
6. N. Breaz, *Modelare asistată de software matematic, Note de curs și teme de laborator*, Seria Didactică a Universității „1 Decembrie 1918” Alba Iulia, 2010.
7. L. Dragoș, *Principiile mecanicii analitice*, Editura Tehnică București, 1976.
8. M. Ghinea, V. Fireșteanu, *MATLAB-calcul numeric, grafică, aplicații*, Teora, 2003.
9. F. Gorunescu, A. Prodan, *Modelare stochastică și simulare*, Ed. Albastră, Cluj-Napoca, 2001.
10. H. Kammerer, *Classical Mechanics with Maple*, ediție electronică, 2003, <http://www.adeptscience.co.uk/products/mathsim/maple/powertools/mechanical>.
11. D. Massier, *Mecanică. Aplicații, Partea I: Mecanica newtoniană a sistemelor discret de puncte materiale*, Editura Universității București, 1998.
12. D. Massier, *Mecanică. Aplicații. Partea II: Mecanica newtoniană a solidului rigid*, Editura Universității București, 2001.
13. *MATLAB - The Language of Technical Computing, MATLAB 6 (Release 12)*, The MathWorks Inc., 2000.
14. C. Moler, *Numerical computing with MATLAB*, Electronic Edition: The MathWorks, Inc., Natick, MA, 2004, <http://www.mathworks.com/moler>.

15. I. Paraschiv-Munteanu, D. Massier, *Probleme de mecanică rezolvate în MATLAB*, Editura Universității București, 2008.
16. Păstorel Gașpar, *Analiza armonică a proceselor stochastice*, Ed. Univ. de Vest, Timișoara, 2008.
17. A.I Rus, C.Iancu, *Modelare matematică*, Editura Transilvania Press, Cluj-Napoca, 2000.
18. Steven Kay, *Intuitive Probability and Random Processes using MATLAB*, Springer, New York, 2006.
19. Steven Kay, *Modern Spectral Estimation: Theory and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
20. Dan Ștefănoiu, *Enciclopedia Matematică, Cap. 20*, coordonatori: M. Iosifescu, O. Stănășilă, D. Ștefănoiu, Ed. AGIR, București, 2010.
21. O.Yakimenko, *Engineering Computations and Modeling in Matlab/Simulink*, AIAA Education, 2011.
22. <http://aparate.elth.ucv.ro/index.php?path=CAUTIL%2FCURSURI%2FOptimizari/>
23. The MathWorks, Inc., *MATLAB – The Language of Technical Computing, Getting Started with MATLAB*, www.mathworks.com
24. Matlab & Simulink a tutorial by Tom Nguyen, <http://edu.levitas.net/Tutorials/Matlab/plotting.html>
25. ***, *Documentation for MathWorks Products, R2009a-* <http://www.mathworks.com/>