Rezolvarea sistemelor neliniare

Sistemele de ecuații neliniare conțin, în exprimarea analitică a ecuațiilor, necunoscute la puteri diferite de unitate. Aceste reprezentări sunt ecuații ale curbelor de diferite ordine.

Rezolvarea sistemelor de ecuații neliniare este posibilă atât exact (folosind, în MATLAB, facilitățile toolbox-ului Symbolic Math Toolbox, prin funcția *[x,y]=solve(ecuația_1, ecuația_2, ..., ecuația_n)*), cât şi folosind diferite metode iterative, pornind de la o soluție inițială de aproximare (metoda Seidel, metoda Newton-Raphson).

Rezolvarea sistemelor neliniare prin utilizarea funcției **solve()** este deosebit de avantajoasă, prin faptul că oferă soluții exacte şi rapide, în cazul sistemelor cu număr mare de ecuații. Dezavantajul major al folosirii acestei funcții constă în faptul că, în multe cazuri sunt oferite soluții redundante, provenind din situații de simetrie a soluțiilor. De exemplu, pentru sistemul de ecuații $\begin{cases} \mathbf{a^3 \cdot b^2 = 0} \\ \mathbf{a - 2 \cdot b = p} \end{cases}$ se obțin, datorită primei ecuații, soluții redundante $\mathbf{a_1=a_2=a_3=0}$, conform succesiunii:

```
» syms a b p
» [a b]=solve('a^3*b^2=0','a-2*b=p',a,b)
a =
    [ 0]
    [ 0]
    [ 0]
    [ p]
    [ p]
b =
    [ -1/2*p]
```

```
[ -1/2*p]
[ -1/2*p]
[      0]
[      0]
```

Pentru conformitate şi completarea informaţiilor, se sugerează consultarea capitolului II al prezentei lucrări.

De exemplu, rezolvarea sistemului $\begin{cases} 7 \cdot x^3 - 10 \cdot x - y - 1 = 0 \\ 8 \cdot y^3 - 11 \cdot y + x - 1 = 0 \end{cases}$ oferă **nouă soluţii**, care sunt, de fapt, punctele de intersecţie între curbele reprezentate analitic prin cele două ecuaţii ale sistemului. Pentru exemplificare, se va rezolva sistemul, folosind funcţia **solve()**, şi se vor reprezenta grafic cele două curbe, pe aceeaşi figură, ilustrându-se intersecţia în cele nouă puncte.

```
» syms x y
» [x,y]=solve(7*x^3-10*x-y-1,8*y^3-11*y+x-1)
x =
    [   -1.1980103893746355864034307510181]
    [   -1.1531141693400621144345138714630]
    [   -1.0604369649024398560611207640276]
    [   -.23114478759741808537998918326327]
    [ -.90533053970812374150742462794171e-1]
    [ .12485152998404704822816298527946e-1]
    [    1.1860751212838186219075928192669]
    [    1.2433857533161570535543994026046]
    [    1.2912933375869876361449885121668]
y =
    [   -1.0558299812819083326283376342552]
    [    -.20170599392974627723151142661l]
    [   1.2569429491012192993810497430700]
    [   1.2250007910285437430503287387984]
    [ -.9986367086952217754320754065747e-1]
    [   -1.1248379067679301599243037612649]
    [    -1.1809721090414812605409103563 20]
    [    .22133863796246260433589763463e-1]
    [     1.1591320579645789050033024737 80]
» x=eval(x)
x =
      -1.19801038937464
```

```
        -1.15311416934006
        -1.06043696490244
        -0.23114478759742
        -0.09053305397081
         0.01248515299840
         1.18607512128382
         1.24338575331616
         1.29129333758699
» y=eval(y)
y =
        -1.05582998128191
        -0.20170599392975
         1.25694294910122
         1.22500079102854
        -0.09986367086952
        -1.12483790676793
        -1.18097210904148
         0.02213386379625
         1.15913205796458
» ezplot('7*x^3-10*x-y-1')
» hold on
» ezplot('8*y^3-11*y+x-1')
```

Rezultatele grafice sunt afişate în figura 4.12:

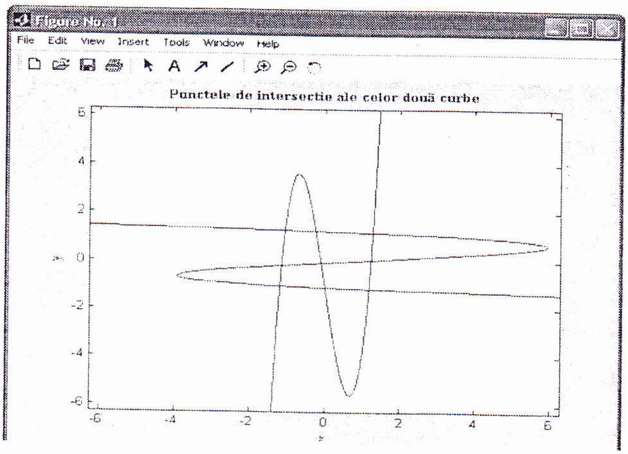

Fig. 4.12 – Intersecţia a două curbe de gradul III

Metoda aproximațiilor succesive pentru rezolvarea sistemelor de ecuații neliniare

Ca şi în cazul sistemelor liniare, şi în acest caz, sunt utilizate formule iterative simple pentru aproximarea iterativă a valorilor soluțiilor unui sistem. Metoda are la bază stabilirea unei formule care să genereze un şir de valori ($\mathbf{m}_k$, $\mathbf{p}_k$) care să fie convergent la valorile (**m**, **p**). De exemplu, pentru sistemul $\begin{cases} \mathbf{x^2 - 2 \cdot x - y + 0.5 = 0} \\ \mathbf{x^2 + 4 \cdot y^2 - 4 = 0} \end{cases}$, se poate constata uşor, că există două perechi de soluții (figura 4.13):

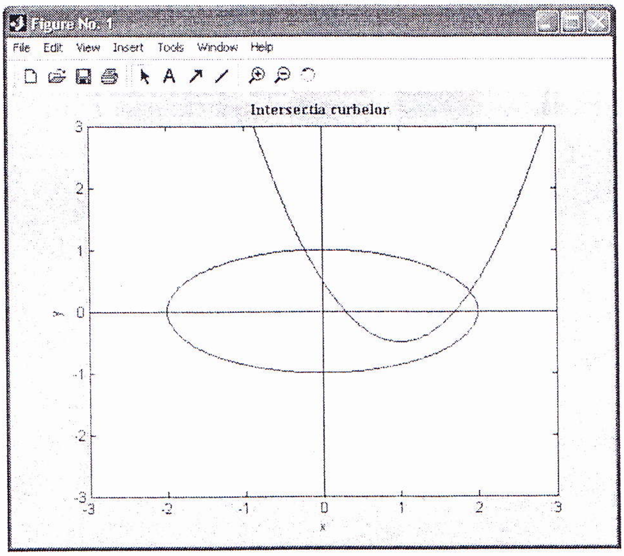

Fig. 4.13 – Intersecția curbelor (soluția sistemului)

Metoda folosită pentru rezolvarea numerică a acestui sistem neliniar de ecuații este o metodă asemănătoare metodei aproximațiilor succesive (metoda iterației la punct fix). Pentru aceasta se foloseşte prima ecuație a sistemului, pentru a exprima pe **x** în funcție de cealaltă variabilă, apoi, se adună, în cea de-a doua ecuație, în fiecare membru, **8y**, obținându-se expresiile:

$$\begin{cases} x = \dfrac{x^2 - y + 0.5}{2} \\ \\ y = \dfrac{-x^2 - 4 \cdot y^2 + 8 \cdot y + 4}{8} \end{cases}$$

, care conduc la definirea unor formule recursive, ale căror expresii sunt:

$$\begin{cases} m_{k+1} = g_1(m_k, p_k) = \dfrac{m_k^2 - p_k + 0.5}{2} \\ \\ p_{k+1} = g_2(m_k, p_k) = \dfrac{-m_k^2 - 4 \cdot p_k^2 + 8 \cdot p_k + 4}{8} \end{cases}$$

. Pornind de la perechea $(\mathbf{m_1}, \mathbf{p_1})=(\mathbf{0},\mathbf{1})$, folosind secvența următoare:

```
» m(1)=0;p(1)=1;
» for k=1:9
   m(k+1)=(m(k)^2-p(k)+0.5)/2;
   p(k+1)=(-m(k)^2-4*p(k)^2+8*p(k)+4)/8;
» end
» m,p
```

se obțin șirurile de valori din tabelul alăturat:

k	m_k	p_k
1	0	1.00000000000000
2	-0.25000000000000	1.00000000000000
3	-0.21875000000000	0.99218750000000
4	-0.22216796875000	0.99398803710938
5	-0.22231471538544	0.99381210235879
6	-0.22219413484094	0.99380287587678
7	-0.22221632115953	0.99380951863159
8	-0.22221471262096	0.99380832729650
9	-0.22221447439565	0.99380840928141
10	-0.22221456832524	0.99380842302332

Pentru aceleaşi formule iterative, folosind ca pereche de valori, de start, perechea $(\mathbf{m_1}, \mathbf{p_1})=(\mathbf{0},\mathbf{1})$, se obțin valorile din tabelul următor:

k	m_k	p_k
1	1.0e+021 * 0.00000000000000	1.0e+021 * 0
2	1.0e+021 * 0.00000000000000	1.0e+021 * 0
3	1.0e+021 * 0.00000000000000	1.0e+021 * - 0.00000000000000
4	1.0e+021 * 0.00000000000000	1.0e+021 * - 0.00000000000000
5	1.0e+021 * 0.00000000000000	1.0e+021 * - 0.00000000000000
6	1.0e+021 * 0.00000000000000	1.0e+021 * - 0.00000000000000
7	1.0e+021 * 0.00000000000000	1.0e+021 * - 0.00000000000000
8	1.0e+021 * 0.00000000000000	1.0e+021 * - 0.00000000000000
9	1.0e+021 * 0.00000000013121	1.0e+021 * - 0.00000000005391
10	1.0e+021 * 8.60762524637768	1.0e+021 * - 3.60518365504342

Se constată că, în acest caz, şirul este divergent. Pentru a rezolva problema divergenţei acestui şir, se va folosi un algoritm aritmetic simplu: se adună, în fiecare membru al primei ecuaţii **-2x**, iar, în cea de-a doua ecuaţie, în fiecare membru, se adună **-3y**, ceea ce duce la schimbarea formulelor iterative, asociate sistemului:

$$\begin{cases} m_{k+1} = g_1(m_k, p_k) = \dfrac{-m_k^2 + 4\cdot m_k + p_k - 0.5}{2} \\ \\ p_{k+1} = g_2(m_k, p_k) = \dfrac{-m_k^2 - 4\cdot p_k^2 + 11\cdot p_k + 4}{11} \end{cases}$$

, atunci, folosind o secvenţă MATLAB,

```
m(1)=2;p(1)=0;
for k=1:9
   m(k+1)=(-m(k)^2+4*m(k)+p(k)-0.5)/2;
   p(k+1)=(-m(k)^2-4*p(k)^2+11*p(k)+4)/8;
end
m,p
```

se obţin rezultatele din tabel:

k	m_k	p_k
1	2.00000000000000	0
2	1.75000000000000	0
3	1.71875000000000	0.08522727272727
4	1.75306285511364	0.17766760776672
5	1.80834482712104	0.25044101782332
⋮	⋮	⋮
11	1.90231369749638	0.31239810081194
12	1.90142774355756	0.31156467378958
⋮	⋮	⋮
22	1.90067728708090	0.31121921719266
23	1.90067710794553	0.31121887589946
24	1.90067691950670	0.31121867376012
25	1.90067679972072	0.31121858249344
26	1.90067674218986	0.31121855327949

Soluția (**1.9006767, 0.3112185**) verifică cele două ecuații ale sistemului inițial, formulele de iterare ale șirurilor ($\mathbf{m_k}$) și ($\mathbf{p_k}$) fiind convergente:

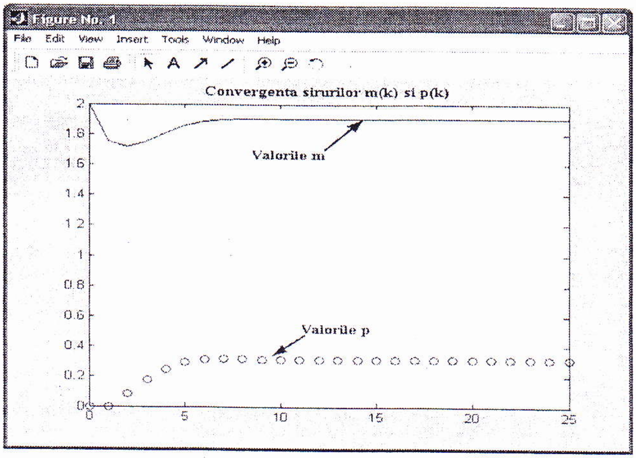

Fig. 4.14 - Convergența șirurilor m_k și p_k