CURS 6

METODE NUMERICE PENTRU ECUAȚII DIFERENȚIALE ORDINARE

Partea II

4 Operatori în mai mulți paşi

4.1 Definiții. Operatori liniari

Considerăm din nou, problema cu valori inițiale (1,2)

$$x' = f(t,x)\,; \qquad x(t_0) = x^{(0)} \tag{41}$$

pentru care se cere soluția pe intervalul $[t_0, TT]$, şi nodurile $t_j,\ j = \overline{0,N}$ (unde $t_N = TT$). Notăm, ca înainte, cu $x(t_j)$ soluția exactă şi cu x_j soluția calculată în $t_j,\ j \geq 1$, iar $x_0 = x^{(0)}$. Un operator în *k*-paşi este o formulă de forma

$$x_{i+1} = g(x_{i+1}, x_i, x_{i-1}, \ldots, x_{i-k+1})\,, \tag{42}$$

care calculează soluția x_{i+1} în funcție de *k* valori $x_j,\ j = i, i-1, \ldots, i-k+1$ calculate anterior. La primul pas al metodei, aceste valori trebuie determinate printr-o procedură specială de start. Dacă x_{i+1} apare în membrul doi operatorul este *implicit*, altfel este *explicit*. În ceea ce urmează vom considera numai cazul operatorilor multi-pas *liniari* şi cu *pas constant h*, adică:

- $t_j = t_0 + jh, \quad j \geq 1$;
- Funcția *g* este liniară în x_j şi în $f_j = f(t_j, x_j)$, $j = i+1, i, \ldots, i-k+1$.

Pentru convenienţă, ecuația (42) se scrie sub forma

$$\begin{aligned} x_{i+1} &= a'_{k-1}x_i + a'_{k-2}x_{i-1} + \ldots + a'_0 x_{i-k+1} \\ &\quad + h(b_k f(t_{i+1}, x_{i+1}) + b_{k-1} f(t_i, x_i) + \ldots b_0 f(t_{i-k+1}, x_{i-k+1})) \end{aligned} \tag{43}$$

În (43) vom presupune că cel puțin unul dintre a'_0, b_0 este diferit de zero (operatorul are *k* paşi); în rest, oricare alt coeficient poate fi zero. Dacă $b_k \neq 0$ operatorul este implicit, iar dacă $b_k = 0$ el este explicit. Condensat, (43) se scrie:

$$x_{i+1} = \sum_{l=0}^{k-1} a'_l x_{i-k+1+l} + h\sum_{l=0}^{k} b_l f_{i-k+1+l}, \quad i \geq k-1 \tag{43a}$$

Observaţie

Pentru simplificarea notaţiei indexate, să notăm

$i+1=k$,

rezultă $i-k+1=0$. Adică:

- Valoarea care se calculează este x_k, şi cele *k* valori anterioare sunt

  $x_{k-1}, x_{k-2}, \ldots, x_0$.

  (Aceasta nu restrânge generalitatea, coeficienţii a'_l, b_l nedepinzând de *i*).

- Momentul curent devine t_k, iar momentele anterioare sunt $t_{k-1}, \ldots, t_0$.

Astfel, relaţia (43a) se scrie:

$$x_k = \sum_{l=0}^{k-1} a_l x_l + h\sum_{l=0}^{k} b_l f_l \tag{44}$$

Forma generală a unei metode multi-pas (44) este

$$\sum_{l=0}^{k} a_l x_l = h\sum_{l=0}^{k} b_l f_l \tag{45}$$

în care s-a pus $a_l = -a'_l, \; l = \overline{0, k-1}$.

În (45) se pun condiţiile:

$a_k = 1$ (mai general, $a_k \neq 0$, pentru explicitare în raport cu x_k)

$|a_0| + |b_0| \neq 0$ (metoda are *k* paşi).

Explicit:

$$a_0 x_0 + a_1 x_1 + \ldots + a_{k-1} x_{k-1} + a_k x_k = h(b_0 f_0 + b_1 f_1 + \ldots + b_{k-1} f_{k-1} + b_k x_k)$$

■

Exemple:

1) Metoda mijlocului este definită de formula

$$x_{i+1} = x_{i-1} + 2hf(t_i, x_i), \quad i \geq 1$$

şi este un operator explicit în 2 paşi.

2) Metoda trapezului este definită de

$$x_{i+1} = x_i + \frac{h}{2}(f(t_i, x_i) + f(t_{i+1}, x_{i+1})), \quad i \geq 0$$

şi este un operator implicit într-un singur pas ■

4.2 Ordin

Formula (44) sau (45) se zice "exactă" pentru o funcție $x(t)$ dacă, din ipoteza că în membrul întâi avem $x_l = x(t_l), \quad l = \overline{i, i-k+1}$, rezultă ca avem şi $x_{i+1} = x(t_{i+1})$ – în limita erorilor de rotunjire.

Definiție

Dacă formula (43) sau (44) este exactă pentru polinoamele de grad *p*, zicem că operatorul are ordinul *p* (sau, formula are ordinul de precizie *p*) ■

Lucrăm pe forma (45)

$$\sum_{l=0}^{k} a_l x_l = h \sum_{l=0}^{k} b_l f_l$$

Să presupunem că cerem ca (45) să aibă ordinul *p*. În acest caz $f(t, x(t)) = x'(t)$ este un polinom de grad $p-1$. Condiția pusă revine la condiția că formula să fie exactă pentru polinoamele $x(t) = t^q$, $q = \overline{0, p}$ (acestea alcătuiesc o bază pentru polinoamele de grad ≤ *p*). Avem $x'(t) = qt^{q-1}$. Putem presupune $t_0 = 0$ (coeficienții nu pot depinde de t_0), avem $t_l = lh$, şi rezultă:

$x_l = x(t_l) = l^q h^q, \quad q \geq 0;$

$f_l = x'(t_l) = q l^{q-1} h^{q-1}, \quad q \geq 1$, şi $f_l = 0$, pentru $q = 0$.

Ținând cont de faptul că termenul al doilea în (44) conține $hf_l = qlh^q$, urmează că h^q se va simplifica. Putem pune atunci, *h* = 1, şi avem:

$x_l = l^q, \quad q \geq 0;$

$f_l = q l^{q-1}, \quad q \geq 1;$ $\qquad f_l = 0$, pentru $q = 0$.

Înlocuind în (45), rezultă, pentru *q* = 0, 1 şi *q* = 2, … , *p*:

1) $q = 0 \quad (x_l = 1, f_l = 0)$:

$$\sum_{l=0}^{k} a_l = 0 \tag{46a}$$

2) $q = 1 \quad (x_l = l, f_l = 1)$:

Rezultă, ținând cont de (45a):

$$\sum_{l=0}^{k} l a_l - \sum_{l=0}^{k} b_l = 0 \tag{46b}$$

3) $q = 2, \ldots, p; p+1 \qquad (x_l = l^q,\ f_l = l j^{q-1})$:

$$\sum_{l=0}^{k} l^q a_l - q \sum_{l=0}^{k} l^{q-1} b_l = 0, \quad q = 2, \ldots, p \tag{46c}$$

$$\sum_{l=0}^{k} l^{p+1} a_l - (p+1) \sum_{l=0}^{k} l^p b_l \neq 0 \qquad (q = p+1) \tag{46d}$$

Rezumând, avem condițiile:

$$\begin{aligned} d_0 &= \sum_{l=0}^{k} a_l = 0 && (q = 0) \\ d_1 &= \sum_{l=0}^{k} l a_l - \sum_{l=0}^{k} b_l = 0 && (q = 1) \\ d_q &= \sum_{l=0}^{k} l^q a_l - q \sum_{l=0}^{k} l^{q-1} b_l = 0, && q = 2, \ldots, p \end{aligned} \tag{46}$$

$$d_{p+1} = \sum_{l=0}^{k} l^{p+1} a_l - (p+1) \sum_{l=0}^{k} l^p b_l \neq 0 \quad (q = p+1)$$

În (46) avem $l^0 = 1, l \geq 0$.

În particular, formulele explicite pentru q = 0, 1, 2, 3 sunt cele de mai jos, în care sumele se opresc la termenii de indice k, inclusiv:

$$d_0 = a_0 + a_1 + a_2 + a_3 + a_4 + \ldots$$

$$d_1 = a_1 + 2a_2 + 3a_3 + 4a_4 + \ldots - (b_0 + b_1 + b_2 + b_3 + b_4 + \ldots)$$

$$d_2 = a_1 + 4a_2 + 9a_3 + 16a_4 + \ldots - 2(b_1 + 2b_2 + 3b_3 + 4b_4 + \ldots)$$

$$d_3 = a_1 + 8a_2 + 27a_3 + 64a_4 + \ldots - 3(b_1 + 4b_2 + 9b_3 + 16b_4 + \ldots)$$

Observaţie

Cu coeficienţii din (43), $a_l = -a'_l$, $l = \overline{1, k-1}$, şi $a_k = a'_k = 1$ condiţiile (46) sunt:

$$d_0 = 1 - \sum_{l=0}^{k-1} a'_l = 0 \qquad (q = 0)$$

$$d_1 = k - \sum_{l=0}^{k-1} l a'_l - \sum_{l=0}^{k} b_l = 0 \qquad (q = 1)$$

$$d_q = k^q - \sum_{l=0}^{k} l^q a'_l - q \sum_{l=0}^{k} l^{q-1} b_l = 0, \qquad q = 2, \ldots, p$$

$$d_{p+1} = k^{p+1} - \sum_{l=1}^{k-1} l^{p+1} a'_l - (p+1) \sum_{l=0}^{k} l^p b_l \neq 0 \quad (q = p+1)$$

■

Cu cele de mai sus avem următoarea propoziţie:

Propozitie

Ordinul unei metode (45) este numărul natural p pentru care avem

$d_0 = 0, d_1 = 0, \ldots, d_p = 0$ şi $d_{p+1} \neq 0$.

Coeficienţii d_q sunt definiţi de (46) ■

Polinoamele generatoare ale metodei

Polinoamele obţinute prin înlocuirile $x_l, f_l \to r^l$ în cei doi membri din (45) se zic polinoamele generatoare ale metodei, şi anume:

$$\begin{aligned} \rho(r) &= \sum_{l=0}^{k} a_l r^l \\ \sigma(r) &= \sum_{l=0}^{k} b_l r^l \end{aligned} \qquad (47)$$

Observaţi că:

$d_0 = \rho(1)$; $\qquad d_1 = \rho'(1) - \sigma(1)$ ■

Consistenţă

O metodă pentru care coeficienţii verifică primele două relaţii (46) – adică, condiţiile pentru $q = 0, 1$:

$d_0 = 1, \quad d_1 = 0$,

se zic *consistentă*. Aceasta echivalează cu condiția ca metoda să fie exactă pentru polinoame de gradul unu. Condiția de consistență se poate exprima şi sub forma următoare, utilizând polinoamele generatoare:

$$\rho(1) = 0, \quad \rho'(1) = \sigma(1) . \tag{48}$$

Exemple-2

1) Reluăm metoda mijlocului (Exemple-1), în care $k = 2$:

$x_{i+1} - x_{i-1} = 2hf_i, \quad i \geq 1$

Avem: $a_0 = -1, a_1 = 0, a_2 = 1; b_0 = 2$. Rezultă:

$d_0 = a_0 + a_1 + a_2 = 0$; $d_1 = a_1 + 2a_2 - b_0 = 2 - 2 = 0$

$d_2 = a_1 + 4a_2 - 2(0 \cdot b_0) = 4 \neq 0$

Astfel, ordinul metodei este $p = 1$.

2) Metoda trapezului ($k = 1$):

$x_{i+1} - x_i = h(\frac{1}{2} f_{i+1} + \frac{1}{2} f_i), \quad i \geq 0$

Avem: $a_0 = -1, a_1 = 1; b_0 = \frac{1}{2}, b_1 = \frac{1}{2}$, şi

$d_0 = 0$; $d_1 = 1 - (\frac{1}{2} + \frac{1}{2}) = 0$

$d_2 = a_1 - 2(b_1) = 1 - 2(\frac{1}{2}) = 0$

$d_3 = a_1 - 3(b_1) = 1 - 3\frac{1}{2} \neq 0$

Rezultă $p = 2$ ■

4.3 Construcția operatorilor în mai mulți paşi

Coeficienții în (44), (45) se determină astfel ca formula să fie exactă pentru polinoamele de un grad dat. Dacă, din condițiiile puse, rămân coeficienți liberi (parametri), aceştia se vor determina astfel ca să avem îndeplinite una sau mai multe din condițiile:

- Eroarea de trunchiere să fie cât mai mică;
- Propagarea erorilor să fie cât mai mică;

- Formula să fie cât mai simplă, de exemplu unii coeficienţi să fie zero.

În afară de aceste condiții, vom cere ca metoda să fie stabilă, v. mai jos.

Determinarea coeficienților în (45) se face prin una din următoarele metode:

- metoda coeficienților nedeterminați
- prin integrare numerică
- prin derivare numerică

Acestea se expun în continuare.

4.3.1 Metoda coeficienților nedeterminați

Relațiile (46) reprezintă un sistem de $p+1$ ecuații în cel mult $2k+1$ coeficienți a_l, b_l (conform $a_k = 1$). Dacă numărul coeficienților este egal cu $p+1$ atunci, sistemul poate fi rezolvat în a_l, b_l. Dacă acest număr este mai mare decât $p+1$, unii coeficienți rămân ca parametri.

Exemple-3

1) Să determinăm metodele 1-pas, de ordin doi ($k = 1$, şi $p = 2$):

$x_{i+1} = a_0' x_i + h(b_1 f_{i+1} + b_0 f_i)$.

În forma (45), metoda se scrie:

$x_{i+1} - a_0' x_i = h(b_1 f_{i+1} + b_0 f_i)$

Cu $a_1 = 1$, ecuațiile (46) sunt: $-a_0' + 1 = 0$, $1 - (b_0 + b_1) = 0$, $1 - 2b_1 = 0$. Din acestea rezultă $a_0' = 1$, $b_1 = b_0 = 1/2$, astfel că metoda căutată este metoda trapezului:

$$x_{i+1} = x_i + \frac{h}{2}(f_{i+1} + f_i).$$

2) Metode 2-pas, explicite, de ordin 1 ($k = 2, p = 1;\ b_1 = 0$)

$x_{i+1} = a_1' x_i + a_0' x_{i-1} + h b_0 f_i$.

Sau, în forma (45): $x_{i+1} - a_1' x_i - a_0' x_{i-1} = h b_0 f_i$. Condițiile (46) sunt $1 - a_0' - a_1' = 0_1$, $1 - b_0 = 0$, astfel că metoda este

$x_{i+1} = (1 - a_0')x_i + a_0' x_{i-1} + hf_i$,

în care $a_0' \neq 0$ rămâne un parametru ■

4.3.2 Metode bazate pe integrare numerică (metodele Adams şi Milne)

Integrând ecuația (41) pe intervalul $[t_i, t_{i+1}]$, avem:

$$x(t_{i+1}) = x(t_i) + \int_{t_i}^{t_{i+1}} f(t, x(t))\, dt$$

Cu *k* aproximații cunoscute $x_i, x_{i-1}, \ldots, x_{i-k+1}$, pe nodurile $t_i, t_{i-1}, \ldots, t_{i-k+1}$, găsim aproximațiile funcției *f* pe aceste noduri:

$f_j = f(t_j, x_j), \quad j = \overline{i, i-k+1}$.

Metode Adams explicite

În membrul doi, înlocuim funcția necunoscută $f(t, x(t))$ cu polinomul de interpolare Newton pe nodurile $t_i, t_{i-1}, \ldots, t_{i-k+1}$. (*k* noduri, polinom de grad $k-1$). Se obține metodele Adams explicite, referite şi ca metode *Adams-Bashforth*:

$$x_{i+1} = x_i + h\sum_{l=0}^{k-1} c_l f_{i-l} = x_i + h(c_0 f_i + c_1 f_{i-1} + c_2 f_{i-2} + c_3 f_{i-3} + \ldots) \qquad (49)$$

Coeficienții c_l din (49) sunt dați în tabelul următor, pentru *k* = 1, 2, 3, 4.

Coeficienții pentru c_l pentru metodele Adams explicite – ecuația (49)

k	f_i	f_{i-1}	f_{i-2}	f_{i-3}	f_{i-4}
1	1				
2	3/2	-1/2			
3	23/12	-16/12	5/12		
4	55/24	-59/24	37/24	-9/24	
5	1901/720	-2774/720	2616/720	-1274/720	251/720

De exemplu, metoda pentru $k = 4$ este:

$x_{i+1} = x_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3})$. Formulele (49) sunt de tipul (44),

cu $a_0 = 1, a_l = 0, l \geq 1$, şi $b_{-1} = 0$, $b_l = c_l$.

Ordin:

Metodele Adams explicite au ordinul $p = k$.

Metode Adams implicite

Analog, cu aproximațiile $f_j = f(t_j, x_j)$ pe nodurile $t_{i+1}, t_i, \ldots, t_{i-k+1}$, utilizăm polinomul de interpolare pentru f.($k+1$ noduri, polinom de grad k).

Se obțin metodele Adams implicite, referite şi ca metode *Adams-Moulton*:

$$x_{i+1} = x_i + h\sum_{l=0}^{k} c_l f_{i+1-l} = x_i + h(c_0 f_{i+1} + c_1 f_i + c_2 f_{i-1} + c_3 f_{i-2} + \ldots) \qquad (50)$$

Pentru $k = 0, 1, 2, 3, 4$, coeficienții din (50) se dau în tabelul următor:

Coeficienții c_l pentru metodele Adams implicite – ecuația (50)

k	f_{i+1}	f_i	f_{i-1}	f_{i-2}	f_{i-3}
0	1				
1	1/2	1/2			
2	5/12	8/12	-1/12		
3	9/24	19/24	-5/24	1/24	
4	251/720	646/720	-264/720	106/720	-19/720

Cazul special $k = 0$ produce metoda Euler implicită: $x_{i+1} = x_i + hf_{i+1}$. Formulele obținute sunt de tipul (44) – cu $a_0 = 1, a_l = 0, l \geq 1$, şi $b_l = c_{l+1}$. Metodele implicite au o precizie mai mare decât cele explicite. Determinarea lui x_{i+1} din formulele de mai sus, se face cu o metodă pentru ecuații neliniare (de exemplu, pentru h = mic, metoda punctului fix).

Ordin:

Ordinul metodei este $p = k + 1$ ■

Metode Milne-Simpson (implicite)

Ecuația (41) se integrează pe intervalul $[t_{i-1}, t_{i+1}]$, obținând

$$x(t_{i+1}) = x(t_{i-1}) + \int_{t_{i-1}}^{t_{i+1}} f(t, x(t))\, dt$$

Sub integrală, înlocuim *f* cu polinomul p_k utilizat în metodele Adams implicite (polinomul de interpolare pe nodurile $t_{i+1}, \ldots, t_{i-k+1}$).

Se obțin metodele:

$$x_{i+1} = x_{i-1} + h\sum_{l=0}^{k} c_l f_{i+1-l} \qquad (51)$$

Coeficienții c_l în (51), pentru $k = 0, 1, 2, 3, 4$, sunt dați mai jos.

Coeficienții c_l pentru metodele Milne-Simpson – ecuația (51)

K	f_{i+1}	f_i	f_{i-1}	f_{i-2}	f_{i-3}
0	2				
1	0	2			
2, 3	1/3	4/3	1/3		
4	29/90	124/90	24/90	4/90	-1/90

Metoda pentru $k = 2$ este numită metoda Milne, şi este dată de

$$x_{i+1} = x_{i-1} + \frac{h}{3}(f_{i+1} + 4f_i + f_{i-1})\,.$$

Ordin:

Metodele Milne-Simpson au ordinul $p = k + 1$ ■

4.3.3 Metode bazate pe derivare numerică (BDF)

În metodele anterioare s-a integrat ecuația (41) şi s-a utilizat polinomul de interpolare pentru funcția $f(t, x(t))$. Considerăm acum ecuația (41), şi polinomul de

interpolare pentru funcția $x(t)$, pe nodurile $t_{i+1}, t_i, \ldots, t_{i-k+1}$ şi valorile $x_{i+1}, \ldots, x_{i-k+1}$ (polinom de grad k):

Avem:

$$\sum_{j=1}^{k} \frac{1}{j} \nabla^j x_{i+1} = h f_{i+1} \qquad (54)$$

Formulele (54) se numesc "formule de derivare înapoi" (*backward differentiation formulae*) şi metodele cu aceste formule se zic *metode BDF*. Ele se utilizează în integrarea numerică a ecuațiilor diferențiale rigide – v. 5.

Formulele (54) explicitate în termenii x_{i+1-l} sunt:

$$\sum_{l=0}^{k} c_l x_{i+1-l} = h f_{i+1} \qquad (54')$$

Pentru $k = \overline{1,6}$, coeficienții c_l sunt dați mai jos. Pentru $k > 6$, metodele BDF sunt instabile (Hairer et al. (1987)).

Coeficienții c_l pentru metodele BDF – ecuația (54')

k	x_{i+1}	x_i	x_{i-1}	x_{i-2}	x_{i-3}	x_{i-4}	x_{i-5}
1	1	-1					
2	3/2	-2	1/2				
3	11/6	-3	3/2	-1/3			
4	25/12	-4	3	-4/3	1/4		
5	137/60	-5	5	-10/3	5/4	-1/5	
6	147/60	-6	15/2	-20/3	15/4	-6/5	1/6

Ordin:

Metodele BDF au ordinul $p = k$ ■

4.4 Stabilitatea metodelor în mai mulți paşi

Stabilitatea metodei analizează comportarea soluției pentru $n \to \infty$ şi $h \to 0$, cu condiția *nh* = constant ($nh = TT - t_0$ = lungimea intervalului de integrare). Pentru $h \to 0$, din (43') se obține:

$$x_{i+1} - a_0 x_i - a_1 x_{i-1} - \ldots - a_{k-1} x_{i-k+1} = 0 \qquad (55)$$

"x_{i+1}" din această ecuație, poate fi interpretat ca soluția dată de metodă, pentru ecuația diferențială $x' = 0$.

Pentru a rezolva ecuația liniară (şi omogenă) cu diferențe (55), căutăm soluții de forma $x_j = r^j$. Înlocuind, obținem

$$r^{i+1} - a_0 r^i - a_1 r^{i-1} - \ldots a_{k-1} r^{i-k+1} = 0,$$

şi împărțind cu r^{i-k+1} rezultă

$$\rho(r) = r^k - a_0 r^{k-1} - a_1 r^{k-2} - \ldots a_{k-1} = 0. \qquad (56)$$

Remarcăm că $\rho(r)$ este primul polinom generator definit în (47). Fie $r_\nu, \nu = 1, k$ rădăcinile polinomului $\rho(r)$. Dacă rădăcinile sunt simple, un sistem de soluții fundamentale este $\{r_1^j, \ldots, r_k^j\}$, şi soluția generală este o combinație liniară de acestea:

$$x_j = \sum_{\nu=1}^{k} c_{j\nu} r_\nu^j, \quad j = i+1, \ldots, i-k+1.$$

Dacă există rădăcini multiple r_ν, cu ordinul de multiplicitate $m_\nu \geq 1$, un sistem de soluții fundamentale corespunzând rădăcinii r_ν sunt $\{r_\nu, jr_\nu, \ldots, j^{m_\nu - 1} r_\nu\}$. Soluția generală are forma

$$x_j = \sum_{\nu=1}^{k} p_{j\nu}(j) r_\nu^j,$$

în care $p_{j\nu}(j)$ sunt polinoame de grad $m_\nu - 1$ în j. În ambele cazuri, pentru ca soluția să rămână mărginită pentru $n \to \infty$, trebuie ca rădăcinile ecuației (56) să se situeze în discul unitate ($|r| \leq 1$), iar rădăcinile de modul 1 să fie simple.

Remarcăm că, pentru metode consistente polinomul $\rho(r)$ are întotdeauna o rădăcină $r = 1$, conform (48).

Definiție

Metoda multi-pas (43) se zice *stabilă*, dacă polinomul generator $\rho(r)$ satisface *condiția de rădăcini*, şi anume:

a) Rădăcinile sunt situate în discul unitate: $|r_v| \le 1$;

b) Rădăcinile de modul 1 sunt simple: dacă $|r_v| = 1$, atunci $\rho'(r_v) \neq 0$ ■

Exemple

1) Stabilitatea metodelor Adams (§ 4.2.2):

Polinomul generator pentru metodele Adams (explicite şi implicite) este $\rho(r) = r^k - r^{k-1}$. Radăcinile sunt $r = 1$ – simplă, şi $r = 0$ – multiplă de ordinul (k-1). Polinomul satisface condiția de rădăcini şi, în consecință, metodele Adams sunt stabile.

2) Stabilitatea metodelor Milne-Simpson (§ 4.2.2):

Polinomul generator $\rho(r) = r^k - r^{k-2}$ satisface condiția de rădăcini, şi deci, metodele sunt stabile. Existența rădăcinii $r = -1$ duce însă la fenomenul de "instabilitate slabă" – v. Hairer & Wanner (1991).

3) Stabilitatea metodelor BDF (§ 4.2.3):

Se arată că metodele BDF sunt stabile pentru $k \le 6$, şi instabile pentru $k \ge 7$.

■

Ordinul maxim al unei metode stabile

Ordinul maxim pentru care metoda este stabilă este dat de următorul rezultat datorat lui Dalquist, şi numit "prima barieră Dalquist".

Propoziție

Ordinul p al unei metode liniare în k-paşi stabilă, satisface:

a) $p \le k+2$ … pentru k = par

b) $p \le k+1$ … pentru k = impar

c) $p \le k$ … pentru $b_{-1} \le 0$ (în particular, pentru o metodă explicită) ■

4.5 Convergența metodelor în mai mulți paşi

Considerăm din nou, problema cu valori inițiale (41), în care presupunem că $f(t,x)$ satisface condițiile din secțiunea 6-I, 1, şi în consecință problema are soluție unică. Considerăm integrarea numerică pe intervalul $[t_0, TT]$, inclus în intervalul de existență a soluției. Reamintim că notăm prin $x(t_j)$ şi x_j, respectiv soluția exactă şi calculată pe punctul $t_j = t_0 + jh$. Pentru ceea ce urmeaza vom presupune că vrem să calculăm soluția pe punctul *fixat* $t \in [t_0, TT]$, cu paşi *h* din ce în ce mai mici. Punem atunci: $t - t_0 = nh = constant$, şi avem $n = (t - t_0)/h$, astfel că $h \to 0 \Leftrightarrow n \to \infty$. Notăm cu $x_t(h)$, soluția calculată pe punctul fixat *t*, cu pasul *h*. Convergența va cere ca, sub anumite ipoteze, să avem limita:

$$\lim_{\substack{h\to 0 \\ t=\text{fixat}}} x_t(h) = x(t)$$

Practic, pentru a calcula $x_t(h)$, utilizăm metoda (43) în care punem $t = t_{i+1}$, $n = i+1$, şi $(i+1)h = t_{i+1} - t_0 =$ constant. În acest caz vom scrie $x_{i+1}(h)$ în loc de $x_{t_{i+1}}(h)$.

Definiție

1) Metoda liniară multi-pas (43) se zice convergentă dacă, pentru orice problemă cu valori inițiale (41), următoarea condiție este îndeplinită:

Dacă valorile de start $x_j(h)$ (pe punctele t_j, $0 \le j \le k-1$), satisfac

$x(t_j) - x_j(h) \to 0$ … pentru $h \to 0$, $j = \overline{0, k-1}$

atunci avem şi

$\forall t \in [t_0, TT]$, t = fixat, $x(t) - x_t(h) \to 0$ … pentru $h \to 0$.

2) Metoda se zice convergentă de ordinul *p* dacă, pentru orice problemă (41) cu *f* suficient derivabilă, există constantele pozitive h_0, C_0, C, astfel ca să avem:

Dacă valorile de start satisfac

$\| x(t_j) - x_j(h) \| \le C_0 h^p$ … pentru $h \le h_0$, $j = \overline{0, k-1}$

atunci

$\forall t \in [t_0, TT]$, t = fixat, $\| x(t) - x_t(h) \| \le Ch^p$ … pentru $h \le h_0$.

■

Rezultatul principal din următoarele teoreme este că proprietățile de consistență + stabilitate ale unei metode, sunt condiții necesare şi suficiente pentru convergența acesteia: *Convergență* ⇔ *Consistență* + *Stabilitate*.

Teorema 1

Dacă metoda (43) este convergentă, atunci ea este stabilă şi consistentă ■

Teorema 2

1) Dacă metoda (43) este stabilă şi de ordinul $p = 1$ (adică, consistentă), atunci ea este convergentă.
2) Dacă metoda (43) este stabilă şi de ordinul p, atunci ea este convergentă de ordinul p ■

4.6 Stabilitate relativă şi stabilitate slabă

Considerăm ecuația de test

$x' = \lambda x$, $x(0) = 1$

care are soluția $x(t) = e^{\lambda t}$.

Definiție

Fie o metodă (43) consistentă, şi r_0 rădăcina principală a polinomului caracteristic (57). Metoda se zice:

- *Relativ stabilă* pe intervalul $[\alpha, \beta] \ni 0$, dacă pentru orice $h\lambda$ în acest interval, rădăcinile polinomului caracteristic satisfac condițiile:

$| r_\nu(h\lambda) | \le | r_0(h\lambda) |, \quad \nu = \overline{1, k-1}$ (61a)

şi, dacă $| r_\nu | = | r_0 |$, atunci r_ν este rădăcină simplă. (61b)

- *Absolut stabilă* pe intervalul $[\alpha, \beta]$ dacă, pentru orice $h\lambda$ în acest interval:

$$|r_\nu(h\lambda)|<1, \quad \nu=\overline{0,k-1} \tag{62}$$

- Metoda satisface condiția *tare* de rădăcini, dacă:

$$|r_\nu(0)|<1, \quad \nu=\overline{1,k-1} \tag{63}$$

O metodă stabilă dar care nu este relativ stabilă, se zice *slab stabilă*.

■

Observații

- Stabilitatea absolută poate avea loc numai în cazul $\lambda<0$ (mai general, λ are partea reală negativă) – v. relația (60). În acest caz, stabilitatea absolută echivalează cu condiția ca $t_j\to\infty\Rightarrow x_j\to 0$.

- Condiția tare de rădăcini implică stabilitatea relativă: cu $r_0(0)=1$, condiția (63) se scrie $|r_\nu(0)|<r_0(0)$, iar din continuitatea rădăcinilor ca funcții de $h\lambda$, rezultă că aceasta are loc pe o vecinătate a lui 0. Reciproca nu este, în general adevărată.

- Întrucât definițiile anterioare se aplică şi la sisteme, λ se consideră, în general, complex. Mulțimea valorilor $h\lambda$ pentru care au loc (60), respectiv (61), se zic *regiunea de stabilitate* relativă, respectiv absolută, ale metodei. Determinarea regiunii de stabilitate absolută este mai simplă decât cea de stabilitate relativă, şi ea se poate face pe criterii algebrice (Hurwitz-Routh, Schur – v. Ralston & Rabinowitz (1978)). Regiunile de stabilitate absolută (în planul complex), pentru metodele Adams-Bashforth şi Adams-Moulton sunt reproduse în Atkinson (1978). Din aceste diagrame rezultă că, cu cât ordinul este mai mare, regiunea de stabilitate este mai mică – v. şi Exemple-3. Totuşi, chiar pentru ordine mari, $|h\lambda|$ rămâne relativ mare, şi nu introduce restricții majore asupra lui h, cu excepția cazului în care $|\lambda|$ este "mare" ■

Exemple

1. Metoda mijlocului este dată de $x_{i+1}=x_{i-1}+2hf(t_i,x_i), \quad i\geq 1$, şi cu $f(t,x)=\lambda x$ devine $x_{i+1}=x_{i-1}+2h\lambda x_i$. Polinomul caracteristic este: $r^2-2(h\lambda)r-1=0$. Punem $\lambda=-K$, unde $K>0$, şi avem

$r_{0,1}(hK) = -hK \pm \sqrt{(hK)^2 + 1}$. Metoda este stabilă (| r | < 1) pentru $hK < 1$, dar avem $|r_1| > |r_0|$, deci metoda nu este relativ stabilă.

2. Metodele Adams: Verificăm condiția (62). Pentru $\lambda = 0$, polinomul caracteristic este $\rho(r) = r^k - r^{k-1}$, cu rădăcinile $r_0 = 1$ şi $r_\nu = 0, \nu = \overline{1, k-1}$. Condiția tare (63) este satisfăcută şi deci, metoda este relativ stabilă.

3. Să determinăm intervalul (presupunem λ real, şi negativ) de stabilitate absolută pentru metodele Adams-Moulton $k = 2$, $k = 3$ (de ordinele 3, 4) – v. Tabelul din 4.2.2. f_j se înlocuieşte cu λx_j. Pentru metoda $k = 2$ avem: $x_{i+1} = x_i + h\lambda(5x_{i+1} + 8x_i - x_{i-1})/12$. Punem $h\lambda = z$ ($z < 0$) şi $x_{i-2} = 1, x_{i-1} = r, \ldots$, rezultă $p(z) = (12 - 5z)r^2 + (12 + 8z)r + z$. Condițiile pentru $|r| < 1$ sunt: $\Delta > 0$; $p(-1) > 0$; $p(1) > 0$; $-1 < (6+4z)/(12-5z) < 1$, care conduc la $-6 < z < 0$.

   Pentru metoda $k = 3$ avem $x_{i+1} = x_i + h\lambda(9x_{i+1} + 19x_i - 5x_{i-1} + x_{i-2})/24$. Cu notațiile anterioare avem $p(z) = (24 - 9z)r^3 - (24 + 19z)r^2 + 5zr - z$. Condiții *necesare* pentru $|r| < 1$ sunt (avem $z < 0$): $p(-1) < 0$; $p(1) > 0$, care conduc la $-3 < z < 0$. Se arată că acesta este rezultatul final. (Exercițiu: verificați că $p''(r) > 0$, $\forall z < 0$.) ■

4.6 Eroarea de trunchiere

Considerăm metoda în k-paşi (44), în care punem acum $x'(t) = f(t, x(t))$

$$x_{i+1} = \sum_{l=0}^{k-1} a_l x_{i-l} + h \sum_{l=-1}^{k-1} b_l x'_{i-l}, \quad i \geq k-1 \tag{64}$$

În (64), x_j şi x'_j sunt aproximații pentru $x(t_j)$ şi $x'(t_j)$. Înlocuind x_j, x'_j cu valorile exacte, formula va avea o eroare pe care o notăm T_{i+1} şi care reprezintă eroarea de trunchiere locală pe pasul i+1:

$$x(t_{i+1}) = \sum_{l=0}^{k-1} a_l x(t_{i-l}) + h \sum_{l=-1}^{k-1} b_l x'(t_{i-l}) + T_{i+1} \tag{65}$$

Presupunem că avem $T_{i+1} = 0$, pentru cazul când x este un polinom de grad $\leq p$ (ordinul metodei este p).

Se arată că eroarea T_{i+1} are forma

$$T_{i+1} = d_p h^{p+1} x^{(p+1)}(\xi), \quad \xi \in (t_{i-k+1}, t_{i+1}) \tag{69}$$

Expresia coeficientului d_p se dă mai jos.

$$(p+1)!\, d_p = k^{p+1} - \sum_{l=0}^{k-2} a_l (k-1-l)^{p+1} - (p+1) \sum_{l=-1}^{k-2} b_l (k-1-l)^p \tag{74}$$

Explicit:

$$\begin{aligned}(p+1)!\, d_p = {} & k^{p+1} - a_0 (k-1)^{p+1} - a_1 (k-2)^{p+1} - \ldots - a_{k-2} \cdot 1 \\ & - (p+1)[b_{-1} k^p + b_0 (k-1)^p + b_1 (k-2)^p + \ldots + b_{k-2} \cdot 1]\end{aligned} \tag{74'}$$

■

Exemple

1. Metode Adams-Bashforth ($p = k$):

Nr. paşi k	Ordin p	Coeficient d_p
1	1	1/2
2	2	5/12
3	3	3/8
4	4	251/720

2. Metode Adams-Moulton ($p = k+1$):

Nr. paşi k	Ordin p	Coeficient d_p
1	2	-1/12
2	3	-1/24
3	4	-19/720
4	5	-3/160

4.7 Metode predictor-corector

4.7.1 Predictori şi corectori

Eroarea unei metode de ordin *p* este dată de (69). Pentru o ecuație dată, la acelaşi pas şi acelaşi ordin, mărimea erorii este dată de $|d_p|$, unde d_p este dat de (74). În general, $|d_p|$ este mai mic pentru o metodă implicită decât pentru una explicită.

Un exemplu îl constituie metodele Adams – compară valorile din exemplele 1 şi 2 de mai sus. Astfel, este preferabil a se determina soluția cu o metodă implicită. Aceasta are forma generală $x_{i+1} = g(x_{i+1}, \ldots, x_{i-k+1})$ şi determină soluția pe un pas cu o metodă iterativă pentru rezolvarea ecuației în x_{i+1}. Astfel, se cere o estimare a aproximației inițiale (la fiecare pas). Cea mai bună cale este de a calcula această aproximație cu o metodă explicită – care va fi numită *predictor*. Apoi se va "corecta" valoarea prin iterație în metoda implicită – aceasta va fi numită *corector*. Metoda obținută prin cuplarea unui predictor şi a unui corector, se va numi o metodă *predictor-corector*. Criterii pentru alegerea predictorului şi corectorului se vor discuta în 4.7.4.

4.7.2 Convergența iterației de punct fix

Explicitând în membrul doi termenul în x_{i+1}, metoda (64) se scrie:

$$x_{i+1} = \sum_{l=0}^{k-1} (a_l x_{i-l} + h b_l x'_{i-l}) + h b_{-1} f(t_{i+1}, x_{i+1}) \equiv g(x_{i+1}) \qquad (75)$$

Să rezolvăm (75) prin metoda punctului fix. Fie la pasul $i+1$ (fixat) estimarea $x_{i+1}^{(0)}$ a lui x_{i+1}, cu aceasta calculăm $f(t_{i+1}, x_{i+1}^{(0)})$ şi înlocuim în (74) obținând $x_{i+1}^{(1)}$, etc. În general, la pasul *j* al iterației avem:

$$x_{i+1}^{(j+1)} = \sum_{l=0}^{k-1} (a_l x_{i-l} + h b_l x'_{i-l}) + h b_{-1} f(t_{i+1}, x_{i+1}^{(j)}), \quad j \geq 0 \qquad (76)$$

şi remarcăm că de la pasul *j* la pasul *j*+1, suma din membrul doi nu se modifică. Condiția de convergență în (76) este $|g'(x)| \leq \lambda < 1$, pe o vecinătate a lui $x_{i+1}^{(0)}$. Avem

$$g'(x) = hb_{-1}\frac{\partial f(t,x)}{\partial x},$$

Presupunând că derivata $\partial f / \partial x$ este mărginită într-o vecinătate I a lui $(t_{i+1}, x_{i+1}^{(0)})$ care conține punctele $(t_{i+1}, x_{i+1}^{(j)})$:

$$\left|\frac{\partial f(t,x}{\partial x}\right| \le \lambda, \quad (t,x) \in I \tag{77}$$

rezultă că trebuie să avem:

$$hb_{-1}\lambda < 1 \tag{78}$$

Mai mult, rata convergenței este $hb_{-1}\lambda$, astfel că pentru o iterație rapidă vom cere să avem $hb_{-1}\lambda << 1$. În (78) s-a presupus $b_{-1} > 0$, ceea ce are loc pentru toate metodele considerate anterior. Pentru convergența pe orice pas (i+1) corespunzând lui $t_{i+1} \in [t_0, TT]$, în (77) se va lua $I = [t_0, TT]$.

Observație

Pentru ecuații diferențiale *rigide* (v. 5), unde λ este "mare", nu putem avea (77) decât pentru un pas h excesiv de mic. În acest caz se va utiliza, în loc de iterația de punct fix, metoda Newton

4.7.3 Estimarea de tip Milne a erorii. Modificarea pasului

Diferența între valoarea corectată $x_{i+1}^{(c)}$ şi valoarea prezisă $x_{i+1}^{(0)}$, constituie o estimare a erorii pe pasul i+1, numită *estimare de tip Milne*:

$$\varepsilon_{i+1} = x_{i+1}^{(c)} - x_{i+1}^{(0)}$$

Dacă în raport cu o toleranță *tol* impusă, avem:

- $|\varepsilon_{i+1}| \le tol$: $x_{i+1}^{(c)}$ este acceptată (ca aproximație pentru x_{i+1}) şi calculul continuă. Dacă $|\varepsilon_{i+1}| << tol$, pasul h poate fi mărit – pentru calculul valorii următoare x_{i+2}.
- $|\varepsilon_{i+1}| > tol$: $x_{i+1}^{(c)}$ nu este acceptată şi se recalculează cu un pas h mai mic;

Utilizarea estimării de mai sus este însă supusă condiției ca predictorul şi corectorul să aibă *acelaşi ordin*.

4.7.4 Eroarea de trunchiere a metodei predictor-corector

Fie predictorul şi corectorul definiți de (44), explicită şi respectiv implicită:

$$x_{i+1}^{(0)} = \sum_{l=0}^{k-1} \alpha_l x_{i-l} + h \sum_{l=0}^{k-1} \beta_l f_{i-l} \qquad \text{- predictor} \tag{79a}$$

$$x_{i+1} = \sum_{l=0}^{k-1} a_l x_{i-l} + h \sum_{l=0}^{k-1} b_l f_{i-l} + h b_{-1} f(t_{i+1}, x_{i+1}^{(0)}) \qquad \text{- corector} \tag{79b}$$

şi presupunem că facem *o singură iterație* în corector adică, aplicăm (79b) cu valoarea $x_{i+1}^{(0)}$ furnizată de (79a), obținând $x_{i+1} = x_{i+1}^{(1)}$. Eroarea de trunchiere a metodei (79a, b) se obține cum urmează.

Eroarea de trunchiere a metodei predictor-corector este de ordinul erorii de trunchiere a corectorului. Pentru aceasta trebuie să avem:

$$p^P \geq p^C - 1$$

unde p^P şi p^C sunt ordinele predictorului, respectiv corectorului.

4.7.5 Alegerea predictorului şi corectorului. Exemple

Alegerea predictorului este legată de o cât mai bună estimare a lui $x_{i+1}^{(0)}$. Între predictori de acelaşi ordin, criteriul este coeficientul d_p al erorii (cât mai mic).

Alegerea predictorului este mai puțin critică decât cea a corectorului. Alegerea corectorului este dictată în principal de proprietățile lui de stabilitate. Între corectori de acelaşi ordin, criteriile de alegere sunt: coeficientul erorii (cât mai mic) şi regiunea de stabilitate absolută (cât mai mare). Dacă predictorul este suficient de exact – v. 4.7.4 , ordinul metodei predictor-corector este ordinul corectorului (dacă ar fi utilizat singur).

Vom da câteva exemple de metode predictor-corector, dintre cele mai utilizate. În predictor, notăm:

$$f_{i+1}^{(j)} = f(t_{i+1}, x_{i+1}^{(j)})$$

Primul exemplu este cel al unei metode de cel mai mic ordin (corector 1-pas, $p = 2$).

0. Metodă de ordin 2:

Predictor (Euler, ordin 1):

$$x_{i+1}^{(0)} = x_i + hf_i$$

Corector (Metoda trapezului, ordin 2):

$$x_{i+1}^{(j+1)} = x_i + \frac{h}{2}(f_i + f_{i+1}^{(j)})$$

1. Metodele Milne şi Hamming

Predictor (ordin 4):

$$x_{i+1}^{(0)} = x_{i-3} + \frac{4h}{3}(f_i - f_{i-1} + 2f_{i-2})$$

Corector (Milne, ordin 4):

$$x_{i+1}^{(j+1)} = x_{i-1} + \frac{h}{3}(f_{i+1}^{(j)} + 4f_i + f_{i-1}), \qquad j \geq 0$$

Coeficienţii erorii sunt: 14/45 (predictor) şi -1/90 (corector).

Corectorul nu este însă relativ stabil, pentru valori $\lambda > 0$ (ecuaţia de test este $x' = \lambda x$ – v. § 4.4).

O modificare a corectorului conduce la metoda Hamming care este stabilă pentru $h\lambda \leq 0.69$. Corectorul devine:

$$\tilde{x}_{i+1}^{(0)} = x_{i+1}^{(0)} + \tfrac{112}{121}(x_i - x_i^{(0)}) \qquad \text{- modificare}$$

$$x_{i+1}^{(j+1)} = \frac{1}{8}(9x_i - x_{i-2}) + \frac{3h}{8}(\tilde{f}_{i+1}^{(j)} + 2f_i - f_{i-1}) \qquad \text{- corector}$$

unde $\tilde{f}$ este calculat în valoarea modificată $\tilde{x}$.

2. Metode Adams – ordin 4:

Predictor (Adams-Bashforth, ordin 4):

$$x_{i+1}^{(0)} = x_i + \frac{h}{25}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3})$$

Corector (Adams-Moulton, ordin 4):

$$x_{i+1}^{(j+1)} = x_i + \frac{h}{24}(9f_{i+1}^{(j)} + 19f_i - 5f_{i-1} + f_{i-2})$$

3. Metode Adams – ordin 5:

Predictor (Adams-Bashforth, ordin 5):

$$x_{i+1}^{(0)} = x_i + \frac{h}{720}(1901 f_i - 2774 f_{i-1} + 2616 f_{i-2} - 1274 f_{i-3} + 251 f_{i-4})$$

Corector (Adams-Moulton, ordin 5):

$$x_{i+1}^{(j+1)} = x_i + \frac{h}{720}(251 f_{i+1}^{(j)} + 646 f_i - 264 f_{i-1} + 106 f_{i-2} - 19 f_{i-3})$$

4.7.6 Implementarea metodelor predictor-corector

Implementarea clasică urmează, în principal, două scheme:

a) PECE: Predicție – Evaluare – Corectare (1 iterație) – Evaluare

Aceasta înseamnă, pe un pas $i+1$:

1. $x_{i+1}^{(0)}$ este estimat de predictor; (P)

2. Se calculează $f(t_{i+1}, x_{i+1}^{(0)})$; (E)

3. Se face *o singură* iteratie în corector, rezultă $x_{i+1}^{(1)}$; (C)

4. Se calculează $f_{i+1} = f(t_{i+1}, x_{i+1})$ (dacă pasul este acceptat). (E)

Schema conduce la 2 evaluări de funcții pe un pas – sub-paşii 2, 4. Dacă predictorul este de ordin ≤ (ordinul corectorului – 1), eroarea de trunchiere a schemei este de ordinul erorii corectorului.

b) P(EC)^M:

1) $x_{i+1}^{(0)}$ este estimat de predictor; (P)

2.0) Testare $| x_{i+1}^{(j+1)} - x_{i+1}^{(j)} | < tol$. Dacă este satisfăcută, se trece la pasul *i*+2. Dacă nu, se face:

2.1) Evaluare: $f_{i+1}^{(j)} = f(t_{i+1}, x_{i+1}^{(j)})$; (E)

2.2) Iterare în corector: rezultă $x_{i+1}^{(j+1)}$, şi se revine la pasul 2.0. (C)

M este numărul de iteraţii pe pasul $i+1$, şi poate varia de la un pas la altul. Schema (b) conduce la *M* evaluări de funcţii pe un pas, dar numărul total de

evaluări de funcții (pentru întregul interval de integrare) poate fi mai mic decât în schema (a), şi în acest caz, schema (b) este mai eficientă.

Observaţie

O altă schemă (c), constă în a impune un număr fixat de iterații M în schema (b). În acest caz, la pasul 2.0 se testează dacă *numărul de iterații* = M. În cazul satisfacerii testului 2.0, pasul este urmat de evaluarea 2.1. În acest caz, schema se simbolizează prin $\mathbf{P}(\mathbf{EC})^M\mathbf{E}$. Cazul $M = 1$ revine la schema (a) ■

Stabilitate:

În schema (a) – PECE, utilizarea unei singure iterații modifică caracteristicile de stabilitate ale metodei în raport cu cele ale schemei (b), şi stabilitatea este influențată de predictorul utilizat. Concret, schema (a) micşorează regiunea de stabilitate pe care o are corectorul utilizat singur. În schema (b) – $\mathrm{P(EC)}^M$, iterarea până la convergență nu modifică stabilitatea metodei şi aceasta nu este influențată de predictorul utilizat. V. o discuție mai amplă în Ralston & Rabinowitz (1978), şi reprezentări ale regiunii de stabilitate pentru schema (a) în Hairer et al. (1991).

Observaţie

Codurile care implementează aceste scheme utilizează procedeul *pas variabil – ordin variabil* (abreviat *VSVO*), adică în funcție de un test pe un pas $i+1$ acceptat, la pasul următor se poate varia atât ordinul metodei cât şi mărimea pasului ■

4.7.7 Determinarea valorilor de start

La primul pas ($i = 0$), metoda în k-paşi cere k valori de start $x_0, x_{-1}, \ldots, x_{=k+1}$.

$x_0 = x^{(0)}$ este dat de condițiile inițiale, dar pentru celelalte valori trebuie utilizată o procedură de determinare. Aceasta poate fi:

a) Seria Taylor

b) Metode Runge-Kutta

c) Metode multi-pas de ordin mai mic

(a) Se utilizează desvoltarea lui $x(t)$ în serie Taylor, în jurul lui t_0:

$$x(t_0 + jh) = x_0 + jhx_0' + \frac{(jh)^2}{2!}x_0'' + \ldots \qquad j = -1, \ldots, -k+1$$

Derivata x_0' se calculează, cu condițiile inițiale, din ecuația dată $x' = f(t, x)$, iar derivatele $x_0^{(n)} = x^{(n)}(t_0)$ – prin derivarea ecuației. Desvoltatea în serie se face până la termenul de ordinul *p* inclusiv, unde *p* este ordinul metodei.

(b) Metodele Runge-Kutta sunt auto-start şi pot astfel furniza valorile de start. Se va utiliza o metodă al cărei ordin este egal cu ordinul metodei multi-pas.

(c) Procedurile a, b, au fost utilizate înainte de apariția implementării metodelor *VSVO* (*pas variabil - ordin variabil*). În acestea din urmă, se începe integrarea cu o metodă 1-pas (v. Exemplul 0 în 4.7.5) care calculează x_1, şi se continuă cu metode în 2, 3, …, (*k*-1) paşi. Cu cele *k* valori calculate, se continuă cu metoda în *k*-paşi. Astfel, aceste metode devin metode auto-start.

4.8 Comparația metodelor predictor-corector (PC) cu metodele Runge Kutta (RK)

Comparația se face luând în considerare următoarele criterii:

a) Necesitatea valorilor de start;

b) Precizie;

c) Număr de evaluări de funcții / pas;

d) Numărul de evaluări suplimentare de funcții, necesare pentru controlul erorii locale de trunchiere.

(a) Metodele R-K sunt auto-start, pe când metodele PC cer o procedură pentru valorile de start. Totuşi, în implementarea predictor-corector, ordin variabil-pas variabil, metodele PC devin auto-start.

(b) La ordine egale, precizia metodelor RK este uşor superioară. Compară rezultatele din 4.9 (v. mai jos) cu cele din 3.3.9.

(c) Pentru metodele RK, numărul de evaluări/pas este ≥ ordinul metodei. Pentru metodele PC, în implementarea PECE acest număr este 2, dar în implementarea P(EC)^M, respectiv P(EC)^{M}E, acesta depinde de numărul *M* de iterații (fiind egal cu *M*, respectiv *M*+1).

(d) Controlul erorii locale de trunchiere: cere evaluări suplimentare în metodele RK, în timp ce la metodele PC nu cere evaluări suplimentare.

Codurile actuale se orientează mai mult spre metodele RK, sau metode de tip similar pentru ecuații diferențiale de ordinul doi (metodele Nyström) – cu excepția ecuațiilor diferențiale rigide, pentru care se utilizează metode BDF şi metode RK implicite.

4.9 Exemple numerice

Vom relua problema celor două corpuri din 3.3.9, pentru $e = 0.9$, calculând soluția pe [0, 20] în dublă precizie, cu următoarele metode predictor-corector:

- Adams de ordinele 4 şi 5 (4.7 – Exemplele 2 şi 3), cu pas constant şi cu 1 iterație în corector – utilizând codul din Anexa, 4.2.
- Adams, ordin variabil-pas variabil, ordinul ≤ 12, lucrând cu subrutina DIVPAG din IMSL. S-au considerat două cazuri: ordinul ≤ 5, şi ≤ 12. Întervalul de integrare s-a împărțit în 20, respectiv 200, sub-intervale. Rezultate mai precise se obțin pentru 20 sub-intervale (pasul maxim este astfel 1.).

Rezultatele se dau în tabelele următoare.

$e = 0.9$, Metoda Adams ordin 4, pas constant: Erori absolute extreme la $t = 18.84$

Pasul h	Eroarea absolută Maximă ($\dot{x}$)	Minimă (x)	Nr. apeluri DERIVS
0.01	3.65 D0[†]	3.30 D-1[†]	4010
0.001	2.70 D-2	1.14 D-5	40010
0.0005	2.09 D-3	1.14 D-6	80010

[†] Eroarea maximă are loc în $\dot{y}$ şi cea minimă în y.

$e = 0.9$, Metoda Adams ordin 5, pas constant: Erori absolute extreme la $t = 18.84$

Pasul h	Eroarea absolută Maximă ($\dot{x}$)	Minimă (x)	Nr. apeluri DERIVS
0.01	4.29 D0[†]	2.97 D-1[†]	4013
0.001	6.64 D-4	3.69 D-7	40013
0.0005	3.33 D-5	1.86 D-8	80013

[†] Eroarea maximă are loc în $\dot{y}$ şi cea minimă în $\dot{x}$.

Observaţie

Exemplele din tabelele anterioare s-au rulat şi iterând în corector până la convergență, cu toleranța *tol* = 1D-10 (pentru iterația de punct fix). Ordinul erorii a fost aproximativ acelaşi, iar numărul de iterații a crescut nesemnificativ: de exemplu, pentru Adams ordinul 5, $h = 0.001$, valorile din tabelul de mai sus sunt: 1.71 D-4, 9.85 D-8, 40942 ■

$e = 0.9$, Metoda Adams, ordin maxim 5, ordin variabil-pas variabil (DIVPAG): Erori absolute extreme la $t = 18.0$

Toleranța *TOL*	Eroarea absolută Maximă (x)	Minimă (y)	Număr de paşi	Număr apeluri FCN
1 D-10	1.11 D-6	1.05 D-8	2437	2577
1 D-15	6.84 D-11	4.44 D-13	16276	16424

$e = 0.9$, Metoda Adams, ordin maxim 12, ordin variabil-pas variabil (DIVPAG): Erori absolute extreme la $t = 18.0$

Toleranța *TOL*	Eroarea absolută Maximă (x)	Minimă (y)	Număr de paşi	Număr apeluri FCN
1 D-10	2.07 D-07	1.44 D-08	1611	1760
1 D-15	6.28 D-12	3.25 D-13	4228	4417

Observaţii

- Pasul maxim care a putut fi utilizat de DIVPAG a fost 1, adică lungimea sub-intervalului (nu s-au impus nici pasul maxim, nici pasul minim, care pot fi specificaţi în parametrii de intrare `hmax`, `hmin`). Aceasta explică numărul redus de paşi cu care lucrează rutina chiar pentru *TOL* foarte mic.
- *TOL* este un parametru care serveşte la controlul normei erorii locale, astfel ca eroarea globală să fie proporţională cu *TOL*. Norma aleasă în exemplele de mai sus a fost norma-∞. *TOL* şi tipul de normă sunt parametri de intrare ai rutinei DIVPAG (în tabloul `param`). A nu se confunda parametrul *TOL* cu toleranța *tol* pentru iterația de punct fix în 4.7.6.

- Se observă superioritatea implementării în forma ordin variabil-pas variabil, cu controlul erorii, față de implementarea cu pas fix.
- Comparând rezultatele cu cele obținute prin metoda Runge-Kutta 8(7) – v. 3.3.9, se constată o precizie mai mare a acesteia din urmă, cu prețul unui număr mai mare de evaluări de funcții: de exemplu, pentru toleranța 2.22D-15, ordinul erorii este în plaja D-13 … D-14, la un număr de 1380 paşi şi cca. 18000 evaluări ■

5 Ecuații diferențiale "rigide"

Vom face în continuare o scurtă introducere în problematica ecuațiilor diferențiale rigide. Pentru o tratare extensivă trimitem la tratatul Hairer et al. (1991), dedicat integrării ecuațiilor diferențiale rigide.

În aplicarea unei metode numerice, dimensiunea pasului se alege dintr-o condiție de precizie a soluției, impunând o toleranță pentru eroarea de trunchiere locală. De obicei, pasul rezultat este în regiunea de stabilitate a metodei. Dacă însă dimensiunea pasului este dictată mai degrabă de condiția de stabilitate decât de condiția de precizie, zicem că avem o problemă *rigidă*.

Exemplu – 1

Să considerăm următoarea problemă:

$$x'' + 101x' + 100x = 0; \quad x(0) = 1,\ x'(0) = 0$$

Punând ecuația sub forma unui sistem de ordinul întâi, avem

$$x' = u, \quad u' = -100x - 101u; \qquad x(0) = 1,\ u(0) = 0,$$

sau matriceal, $\mathbf{x}' = \mathbf{A}\mathbf{x}$, unde $\mathbf{x} = [x \quad u]^T$, iar $\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -100 & -101 \end{pmatrix}$.

Valorile proprii ale lui **A** sunt $\lambda_1 = -1, \lambda_2 = -100$. Soluția exactă este de forma

$$x(t) = C_1 e^{-t} + C_2 e^{-100t} \tag{84}$$

şi cu condițiile inițiale date rezultă

$$x(t) = \tfrac{100}{99} e^{-t} - \tfrac{1}{99} e^{-100t}. \tag{85}$$

Termenul în e^{-100t} tinde rapid spre 0 (se amortizează): de exemplu, pentru $t = 0.1$ avem $e^{-10} = 4.54 * 10^{-5}$, astfel că pentru $t > 0.1$ avem:

$$x(t) \approx \tfrac{100}{99} e^{-t} \qquad (86)$$

Soluția dată de al doilea termen din (85) zice *tranzitorie*. Soluția (86) reprezintă soluția *staționară*. Pentru t suficient de mare (de exemplu, $t > 0.1$), soluția tranzitorie nu mai contribuie la soluția (85), dar determină stabilitatea metodei. Chiar şi în cazul în care termenul al doilea dispare din soluția (84) – condițiile inițiale sunt astfel că rezultă $C_2 = 0$ – valoarea proprie λ_2 determină intervalul de stabilitate al metodei. Într-adevăr, să presupunem că integrăm sistemul cu metoda RK4. Intervalul de stabilitate absolută a metodei este (v. 3.3.9): $-2.785296 < h\lambda < 0$, sau $0 < h(-\lambda) < 2.785296$, ceea ce conduce (pentru λ_2) la $h < 0.02785$. Întrucât ordinul erorii globale a metodei RK4 este h^4, dacă am vrea să calculăm soluția cu o eroare de ordinul 10^{-4}, ar fi suficient un pas de ordinul $h = 0.1$ – care este însă în afara intervalului de stabilitate. Într-adevăr, calculând cu $h = 0.025$ (pas constant) se obține $x(10) = 4.5858514950$ E-5 care are o eroare de –6.21E-13, în timp ce cu $h = 0.28$ şi 0.3, rezultă respectiv $x(9.996) = -2.74…$E+1, şi $x(9.990) = -1.1459…$E+44 – care probează instabilitatea ■

Fie un sistem liniar $\mathbf{x}' = \mathbf{A}\mathbf{x}$, de m ecuații, şi λ_i valorile proprii ale matricii **A**. Faptul că sistemul este rigid se defineşte prin următoarele condiții:

$$\mathrm{Re}(\lambda_i) < 0, \quad i = \overline{1,m}$$

$$\max_{i=1,m} |\mathrm{Re}(\lambda_i)| >> \min_{i=1,m} |\mathrm{Re}(\lambda_i)|$$

Cu alte cuvinte, sistemul este rigid dacă are un punct fix stabil, şi **A** are valori proprii de mărimi foarte diferite. Definiția de mai sus are mai multe inconveniente, şi anume:

- Nu mai convine în cazul în care matricea **A** are o valoare proprie egală cu zero;
- Nu se poate aplica pentru un sistem neliniar, sau pentru o singură ecuație.

Se dă atunci următoarea definiție mai puțin precisă, dar aplicabilă atât sistemelor liniare cât şi celor neliniare, cât şi pentru o singură ecuație (Lambert, v. Cartwright and Piro (1992)):

"Dacă o metodă numerică este forţată să utilizeze, într-un anumit interval, un *pas excesiv de mic* pentru o problemă a cărei soluţie exactă este netedă în acel interval, atunci problema se zice *rigidă* în acel interval"

(O funcţie este netedă în [*a*, *b*], dacă are derivată continuă în [*a*, *b*].)

O problemă poate fi rigidă pe unele sub-intervale ale soluţiei şi non-rigidă pe altele. Definiţia de mai sus permite codurilor pentru integrarea numerică, să "recunoască" rigiditatea problemei pe intervalul pe care se calculează soluţia (sau pe sub-intervale ale acestuia), prin faptul că rutina este forţată să micşoreze excesiv pasul, pentru a satisface toleranţa impusă erorii de trunchiere. Utilizarea unui pas foarte mic poate crea probleme datorate acumulării erorilor de rotunjire sau creşterii timpului de calcul.

Pentru o problemă rigidă controlată de un parametru, s-ar cere ca metoda de integrare să fie stabilă pentru orice dimensiune a pasului, la orice valoare a parametrului pentru care problema este stabilă. De exemplu, problema de test pentru stabilitatea absolută, $x' = \lambda x$, este stabilă pentru $\mathrm{Re}(\lambda) < 0$, iar metoda ar trebui să fie stabilă pentru orice h, oricare ar fi λ cu $\mathrm{Re}(\lambda) < 0$. Regiunea de stabilitate absolută este atunci tot semi-planul (complex) stâng. Aceasta conduce la definiţia *A*-stabilităţii:

"O metodă se zice *A*-stabilă dacă regiunea ei de stabilitate liniară absolută conţine întreg semi-planul stâng".

Metodele RK explicite nu au *A*-stabilitate, deoarece regiunile lor de stabilitate absolută sunt finite – v. 3.3.7. În schimb, unele metode RK implicite sunt *A*-stabile şi se pot aplica la ecuaţii rigide. Inconvenientul este că ele cer rezolvarea unui sistem neliniar ceea ce măreşte numărul de evaluări de funcţii. *A*-stabilitatea este o cerinţă severă pentru o metodă. Pentru metodele multi-pas, ordinul maxim al unei metode *A*-stabile este $p = 2$ (a doua "barieră Dalquist"). De exemplu, metodele Adams-Moulton sunt *A*-stabile numai pentru $k = 1$ (metoda trapezului implicită), iar metodele BDF numai pentru $k = 1$ (Euler implicită) şi $k = 2$. V. Hairer et al. (1991).

În mod curent, problemele rigide se rezolvă cu metode BDF (4.2.3), numite şi metode *Gear*. Acestea sunt tot implicite. Metodele BDF sunt implementate în subrutina DIVPAG, care permite, prin codul metodei *param*(10), alegerea

metodelor Adams (v. 3.10), sau BDF până la ordinul 5. Ordinul este limitat din considerente de stabilitate. Codul *param*(13) permite alegerea rezolvitorului sistemului neliniar (iterația de punct fix, metoda Newton, şi metode Newton modificate), recomandându-se alegerea metodei Newton sau Newton-modificată (Secțiunea 3-IV, 2).

Exemplu – 2

Considerăm ecuația van der Pol

$$x'' - \lambda(1 - x^2)x' + x = A\cos(\omega t)$$

în cazul vibrației libere $A = 0$, pentru valori "mici" şi "mari" ale parametrului λ – de exemplu, $\lambda = 1$ şi $\lambda = 100$. Luăm condițiile inițiale: $x(0) = 1, x'(0) = 0$ şi integrăm ecuația pe intervalul [0, 100], în dublă precizie, cu următoarele metode:

- Metoda RK4, pas constant (codul din ANA_EcDif): Cazul $\lambda = 1$ se integrează cu un pas $h = 0.1$ (1000 paşi). În cazul $\lambda = 100$, pasul maxim pentru stabiliate este $h = 0.0083$ (12048 paşi); cu $h = 0.084$, la $t = 0.5628$, soluția (x, x') calculată este (3.035E+88, - 6.173E+181), şi apoi (*NaN*, *NaN*), care probează instabilitatea metodei pentru acest pas. Pasul mult mai mic necesar în cazul $\lambda = 100$, arată că ecuația este rigidă. Pentru o precizie comparabilă cu cea a metodelor RKV şi BDF, în cazul $\lambda = 1$ a fost necesar un pas de 2.5E-3, iar în cazul $\lambda = 100$, un pas de 2.5E-4.
- Metoda RK-Verner 5(6), pas variabil între 1 şi 2.22E-15, cu toleranța $tol = 1D-10$ (v. 3.10): Cazul $\lambda = 1$ este integrat cu 21375 evaluări (2594 paşi, pasul de încercare (trial step) la $t = 100$, $h = 5.9E-2$). Cazul $\lambda = 100$ este integrat cu 63522 evaluări (6941 paşi, pasul de încercare la $t = 100$ este $h = 1.59E-2$).
- Metoda BDF, pas variabil – ordin variabil (ordin ≤ 5), pasul între 1.0 şi 1E-5, toleranță 1D-10, rezolvitor Newton cu jacobianul calculat numeric (subrutina DIVPAG): Cazul $\lambda = 1$ cere 14109 evaluări (12427 paşi, pasul de încercare la $t = 100$, $h = 1.24E-2$). Cazul $\lambda = 100$ cere 4015 evaluări (3256 paşi, pasul de încercare la $t = 100$, $h = 9.86E-1$).

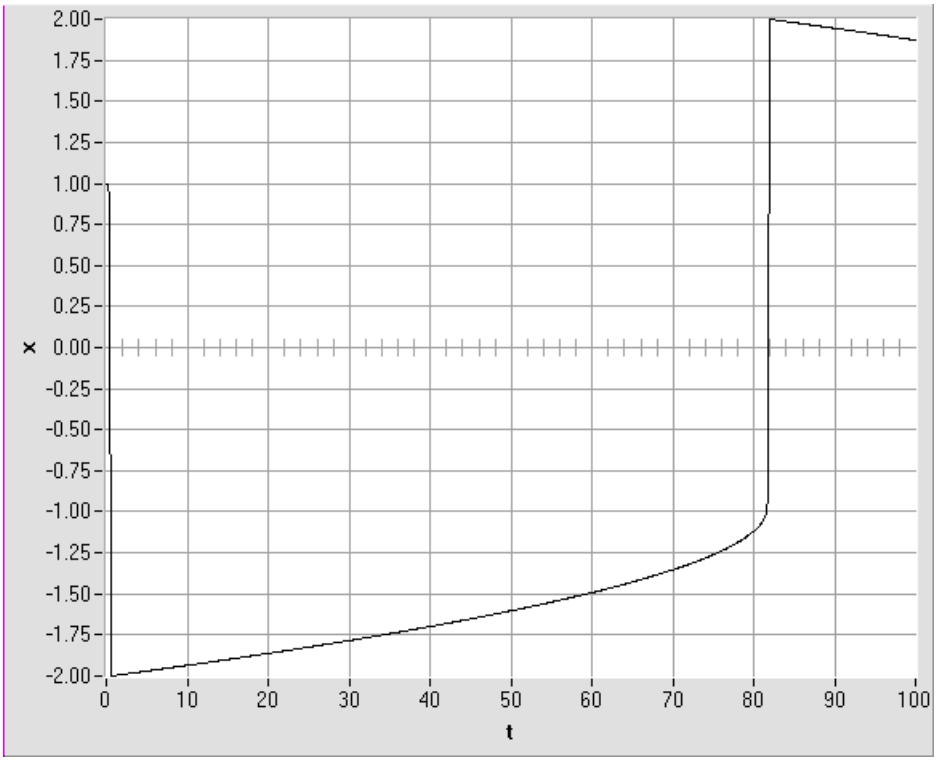
Rezultate comparative pentru ultimele două metode se dau în tabelul următor. Metodele s-au rulat cu aceeaşi toleranţă 1D-10, şi produc la $t = 100$, soluţii x care au 8 cifre semnificative identice. (Soluţiile x' au tot 8 cifre semnificative identice).

Ecuaţia van der Pol – Număr de evaluări de funcţii şi pas de încercare la $t = 100$

Metoda	$\lambda = 1$	$\lambda = 100$
RKV 5(6), *tol* = 1D-10	21375; 5.91 E-2	63522; 1.59 E-2
BDF, *tol* = 1D-10	14109; 1.24 E-2	4021; 9.86 E-1

Comparaţia eficienţei după numărul de evaluări de funcţii arată net superioritatea metodei BDF pentru cazul ecuaţiei rigide, dar şi pentru cazul ecuaţiei non-rigide.

În graficul următor se reprezintă soluţia $x(t)$ a problemei, pentru $\lambda = 100$.



Ecuaţia van der Pol, cazul $\lambda = 100$.

■

BIBLIOGRAFIE (selectivă)

Manuale de analiză numerică:

1. Atkinson K.E., "An Introduction to Numerical Analysis", John Wiley & Sons, N.Y., 1978. 2nd edition, 1989.
2. Curtis F.G., "Applied Numerical Analysis", Addison-Wesley Publishing Company, Inc., 1978.
3. Isaacson E., and Keller H.B., "Analysis of Numerical Methods", John Wiley & Sons, N.Y., 1966.
4. Kincaid D., and Cheney W., "Numerical analysis", 2nd edition, Brooks/Cole Publ. Co., 1996.
5. Ralston A., and Rabinowitz Ph., " A First Course in Numerical Analysis", McGraw-Hill, Inc., 1983.

Ecuații diferențiale:

6. Cartwright J.H.E & Piro O., "The Dynamics of Runge-Kutta Methods", Int. J. Bifurcation and Chaos, 2, 427-449, 1992, http://lec.ugr.es/~julyan/publications.html
7. Chisăliță A., Lung N, Chisăliță G.-A., "Criterii numerice şi procedee analitice pentru identificarea răspunsului haotic", Contract 34/1998, Tema 25/155, Universitatea Tehnică din Cluj-Napoca, 1998.
8. Dormand J. R., "Numerical Methods for Differential Equations", CRC Press LLC, (1996).
9. Hairer E., Nørsett S.P., and Wanner G., "Solving Ordinary Differential Equations I (Nonstiff Problems)", Springer-Verlag, 1987.
10. Hairer E., and Wanner G., "Solving Ordinary Differential Equations II (Stiff and Differential-Algebraic Problems)", Springer-Verlag, 1991.
11. Lambert J.D., "Numerical Methods for Ordinary Differential Systems. The Initial Value Problem.", J. Wiley & Sons, 1991.

Biblioteci şi coduri:

12. Chisăliță A, "ANA_EcDif", 2006, ftp.utcluj.ro/pub/users/chisalita/

13. "IMSL Mathematical and Statistical Libraries", Compaq Visual Fortran 6.6, IMSL Help, 1999.
14. Brankin R.W. and Gladwell I., "RKSUITE_90 Release 1.0 June 1994", http://www.netlib.org/ode/rksuite/.
15. Brankin R.W., Gladwell I., and Shampine L.F. , "RKSUITE Release 1.0 November 1991", http://www.netlib.org/ode/rksuite/.