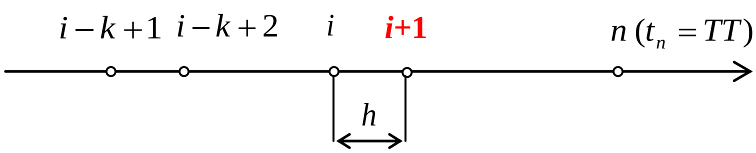
4.3.3 Metode bazate pe derivare numerică (BDF)

Considerăm din nou, problema cu valori iniţiale

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

În metodele anterioare s-a integrat ecuaţia (41) şi s-a utilizat polinomul de interpolare pentru funcţia $f(t, x(t))$. Considerăm acum ecuaţia (41), şi polinomul de interpolare (Newton, cu diferențe înapoi) pentru funcţia $x(t)$, pe nodurile $t_{i+1}, t_i, \ldots, t_{i-k+1}$ şi valorile $x_{i+1}, \ldots, x_{i-k+1}$ ($k+1$ noduri, de indici: $i+1, i, i-1, \ldots, i-(k-1)$; polinom de grad k.)



$$q_k(t) = q_k(t_i + sh) = \sum_{j=0}^{k} (-1)^j \binom{-s+1}{j} \nabla^j x_{i+1}\,,$$

unde $t = t_i + sh$, $s = (t - t_i)/h$.

[Reamintim că:

Diferența regresivă a funcției f pe punctul x, cu pasul h, este

$$\nabla f(x) = f(x) - f(x-h)$$

și că:

$$\nabla f(x) = \Delta f(x-h)\,;$$

$\nabla^r f(x) = \Delta^r f(x - rh)$, sau $\nabla^r f_j = \Delta^r f_{j-r}$.]

Cerem ca polinomul $q_k(t)$ să satisfacă ecuaţia diferenţială pe punctul t_{i+1}:

$$q_k'(t_{i+1}) = f(t_{i+1}, x_{i+1}) \tag{52}$$

Ecuaţia (52) este ecuaţia implicită din care se determină necunoscuta x_{i+1}. Avem:

$$q_k'(t) = \frac{dq_k(s)}{ds} \cdot \frac{1}{h}, \qquad \frac{dq_k(s)}{ds} = \sum_{j=0}^{k+1} (-1)^j \frac{d}{ds} \binom{-s+1}{j} \nabla^j x_{i+1},$$

de unde, ţinând cont de $t = t_{i+1} \Leftrightarrow s = 1$, rezultă

$$q_k'(t_{i+1}) = \frac{1}{h} \sum_{j=0}^{k} (-1)^j \frac{d}{ds} \binom{-s+1}{j} \Bigg|_{s=1} \nabla^j x_{i+1}$$

Cu aceasta, ecuaţia (52) devine formula implicită

$$\sum_{j=0}^{k} \delta_j \nabla^j x_{i+1} = hf_{i+1}, \tag{53}$$

unde s-a pus

$$\delta_j = (-1)^j \frac{d}{ds} \binom{-s+1}{j} \Bigg|_{s=1}$$

Cu definiţia coeficientului binomial avem

$$\binom{-s+1}{j} = (-1)^j \frac{(s-1)s(s+1)\ldots(s+j-2)}{j!},$$

cu care se obţine

$$\delta_j = \frac{1}{j!} \prod_{l=-1}^{j-2} \frac{(s-1)s(s+1)\ldots(s+j-2)}{s+l} \Bigg|_{s=1},$$

sau:

$$\delta_0 = 0; \qquad \delta_j = \frac{1}{j}, \quad j \geq 1$$

Formula (53) devine:

$$\sum_{j=1}^{k} \frac{1}{j} \nabla^j x_{i+1} = hf_{i+1} \tag{54}$$

Reamintim notațiile:

$f_j = f(t_j, x_j)$

(iar: $x_j = x(t_j)$.)

Formulele (54) se numesc “formule de derivare înapoi” (*Backward Differentiation Formulae*) şi metodele bazate pe aceste formule se zic *metode BDF*. Ele se utilizează în integrarea numerică a ecuaţiilor diferenţiale rigide – v. 5.

Utilizăm expresia diferențelor înapoi ale funcției *f* pe t_i :

$$\nabla^j f_i = \sum_{l=0}^{j} (-1)^l \binom{j}{l} f_{i-l} \ ,$$

pe care o scriem sub forma

$$\nabla^j f_i = \sum_{l=0}^{k} (-1)^l c_l^j f_{i-l}$$

unde definim

$$c_l^j = \begin{cases} \binom{j}{l} & l \le j \\ 0 & l > j \end{cases}$$

Luăm acum, $f_i = x_{i+1}$.

Rezultă

$$\nabla^j x_{i+1} = \sum_{l=0}^{k} (-1)^l c_l^j x_{i+1-l}$$

(54) devine:

$$\sum_{j=1}^{k} \frac{1}{j} \sum_{l=0}^{k} (-1)^l c_l^j x_{i+1-l} = hf_{i+1}$$

Intervertind ordinea de sumare, rezultă:

$$\sum_{l=0}^{k} (-1)^l \sum_{j=1}^{k} \frac{1}{j} c_l^j x_{i+1-l} = hf_{i+1}$$

În fine, cu:

$$c_l = (-1)^l \sum_{j=l}^{k} \frac{1}{j} c_l^j \text{ pentru } l \geq 1, \text{ şi } c_0 = \sum_{j=1}^{k} \frac{1}{j}. \tag{55}$$

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

$$\sum_{l=0}^{k} c_l x_{i+1-l} = hf_{i+1} \tag{56}$$

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

Coeficienţii c_l pentru metodele BDF – ecuaţia (56)

k	c_0 (x_{i+1})	c_1 (x_i)	c_2 (x_{i-1})	c_3 (x_{i-2})	c_4 (x_{i-3})	c_5 (x_{i-4})	c_6 (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

Din deducerea formulelor BDF rezultă că acestea sunt exacte pentru cazul în care soluţia este un polinom de grad k (x şi q_k coincid). Cu un raţionament analog cu cel din 4.3.2, rezultă că metodele BDF au ordinul $p = k$.

NOTE

1. Implementare și Valori de start

1.1 Valori de start

Să considerăm, spre exemplu, formula de ordinul 4 (k = 4):

$$c_{40}x_{i+1} + c_{41}x_i + c_{42}x_{i-1} + c_{43}x_{i-2} + c_{44}x_{i-3} = hf_{i+1}$$

Coeficienții sunt notați acum c_{kl} pentru a evidenția și ordinul k al metodei.

Pentru $k = 4$, coeficienții sunt:

25/12; -4; 3; -4/3; 1/4

Concret:

$$\frac{25}{12}x_{i+1} - 4x_i + 3x_{i-1} - \frac{4}{3}x_{i-2} + \frac{1}{4}x_{i-3} = hf_{i+1}$$

În primul moment, i = 0, formula este

$$\frac{25}{12}x_1 - 4x_0 + 3x_{-1} - \frac{4}{3}x_{-2} + \frac{1}{4}x_{-3} = hf_1$$

x_0 este dat (condiția inițială), dar valorile de start x_{-1}, x_{-2}, x_{-3} trebuie produse de altă metodă. Concret, se utilizează metoda Runge-Kutta, sau metodele BDF de ordin mai mic.

Cu BDF de ordine 1...3, avem:

$$c_{10}x_{i+1} + c_{11}x_i = hf_{i+1}$$

$$c_{20}x_{i+1} + c_{21}x_i + c_{22}x_{i-1} = hf_{i+1}$$

$$c_{30}x_{i+1} + c_{31}x_i + c_{32}x_{i-1} + c_{33}x_{i-2} = hf_{i+1}$$

Se scriu acestea pentru $i = -1$:

$$c_{10}x_0 + c_{11}x_{-1} = hf_0 \qquad \Rightarrow x_{-1} = (hf_0 - c_{10}x_0)/c_{11}$$

$$c_{20}x_0 + c_{21}x_{-1} + c_{22}x_{-2} = hf_0 \qquad \Rightarrow x_{-2} = (hf_0 - c_{20}x_0 - c_{21}x_{-1})/c_{22}$$

$$c_{30}x_0 + c_{31}x_{-1} + c_{32}x_{-2} + c_{33}x_{-3} = hf_0 \qquad \Rightarrow x_{-3} = (hf_0 - c_{30}x_0 - c_{31}x_{-1} - c_{32}x_{-2})/c_{33}$$

Coeficienții pentru k = 1; 2; 3; sunt (tabelul de mai sus):

$$\begin{array}{llll} 1 & -1 & & \\ 3/2 & 1/2 & & \\ 11/6 & -3 & 3/2 & -1/3 \end{array}$$

1.2 Implementare

Pentru implementare (în cod), în formula metodei (56) se face $i+1=1$ (sau $i=0$). Indicii devin:

$i+1$	i	$i-1$	$\ldots$	$i+1-k$
1	0	-1	$\ldots$	$-k+1$

Astfel valorile "generale" $x_{i+1}, x_i, x_{i-1}, \ldots, x_{i+1-k}$, devin valorile curente $x_1, x_0, x_{-1}, \ldots, .x_{-k+1}$.

Necunoscuta curentă este x_1, iar valorile anterioare (cunoscute) sunt $x_0, x_{-1}, \ldots, x_{-k+1}$.

Formula (56) devine

$$\sum_{l=0}^{k} c_l x_{1-l} = hf_1$$

Explicit:

$$c_0 x_1 + c_1 x_0 + c_2 x_{-1} + \ldots + c_k x_{-k+1} = hf_1$$

De exemplu, formula anterioară pentru k = 4 este:

$$\frac{25}{12} x_1 - 4x_0 + 3x_{-1} - \frac{4}{3} x_{-2} + \frac{1}{4} x_{-3} = hf_1$$

În cod: valorile curente se stochează în tabloul `xa(-k+1:1, n)`; în acesta, `xa(1, n)` va stoca valoarea calculată pe pasul curent, anume x_1; n reprezintă dimensiunea vectorului **x** din (41) – v. mai jos 2.2.

■

2. Rezolvarea ecuației neliniare în x_{i+1}

2.1 O singură ecuație (41)

Să considerăm din nou, pentru exemplificare, cazul $k = 4$. Ecuația (56) – v. mai sus – se poate scrie:

$$-hf(t_{i+1}, x_{i+1}) + c_{40}x_{i+1} + [c_{41}x_i + c_{42}x_{i-1} + c_{43}x_{i-2} + c_{44}x_{i-3}] = 0 \qquad (56')$$

Sau:

$$\tilde{f}(x_{i+1}) = 0 \qquad (57a)$$

$$\tilde{f}(x) = -hf(t_{i+1}, x) + c_{40}x + [c_{41}x_i + c_{42}x_{i-1} + c_{43}x_{i-2} + c_{44}x_{i-3}] \qquad (57b)$$

S-a notat cu $\tilde{f}$ membrul întâi în (56′); termenii in parantezele mari nu depind de $x = x_{i+1}$.

Ecuația neliniară (57) se rezolvă cu metoda Newton (de exemplu). Derivata funcției $\tilde{f}$ este:

$$\frac{d}{dx}\tilde{f}(x) = -h\frac{\partial}{\partial x}f(t_{i+1}, x) + c_{40}$$

2.2 <u>Cazul sistemului</u>

În locul ecuației scalare (41) avem un sistem de ordinul întâi

$$\mathbf{x}' = \mathbf{f}(t, \mathbf{x}); \qquad \mathbf{x}(t_0) = \mathbf{x}^{(0)}, \qquad (58)$$

unde **x** și **f** sunt vectori cu *n* coordonate:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}; \qquad \mathbf{f}(t, \mathbf{x}) = \begin{bmatrix} f_1(t, x_1) \\ f_2(t, x_2) \\ \vdots \\ f_n(t, x_n) \end{bmatrix}.$$

Atunci, ecuații (56) pot fi scrise pentru fiecare coordonată, astfel că (56) devine

$$\sum_{l=0}^{k} c_l \mathbf{x}^{i+1-l} = h\mathbf{f}^{i+1} \qquad (59)$$

Utilizăm indicii superiori pentru valorile $\mathbf{x}^j = \mathbf{x}(t_j)$ și $\mathbf{f}^j = \mathbf{f}(t_j, \mathbf{x}^j)$.

Indicii inferiori desemnează coordonatele:

$$\mathbf{x}^j = \begin{bmatrix} x_1^j \\ x_2^j \\ \vdots \\ x_n^j \end{bmatrix}$$

Sau:

$$-h\mathbf{f}(t_{i+1}, \mathbf{x}^{i+1}) + c_0\mathbf{x}^{i+1} + \sum_{l=1}^{k} c_l \mathbf{x}^{i+1-l} = \mathbf{0} \qquad (59')$$

Sau:

$$\tilde{\mathbf{f}}(\mathbf{x}^{i+1}) = \mathbf{0} \qquad (60a)$$

$$\tilde{\mathbf{f}}(\mathbf{x}) = -h\mathbf{f}(t_{i+1}, \mathbf{x}) + c_0\mathbf{x} + \sum_{l=1}^{k} c_l \mathbf{x}^{i+1-l} \qquad (60b)$$

Pentru rezolvarea ecuației (60) cu metoda Newton, jacobianul lui $\tilde{\mathbf{f}}$ va fi:

$$\frac{\partial \tilde{f}_m}{\partial x_j}(\mathbf{x}) = -h\frac{\partial f_m}{\partial x_j}(t_{i+1}, \mathbf{x}) + c_0\delta_j^m; \qquad m, j = \overline{1, n}.$$

Unde: $\delta_j^m = \begin{cases} 1 & j = m \\ 0 & j \neq m \end{cases}$ este simbolul Kronecker.

<u>Exemplu</u>

Fie ecuația liniară

$$\ddot{u} + k_1\dot{u} + ku = 0$$

În particular, pentru $k = 100$ și $k_1 = 101$, ecuația este rigidă.

Aceasta se pune sub forma unui sistem de ordinul I, astfel:

$$\dot{u} = v$$
$$\dot{v} = -ku - k_1 v$$

Cu $\mathbf{x} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$,

sistemul devine

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \qquad \text{sau:} \qquad \begin{cases} \dot{x}_1 = f_1(x_1, x_2) \\ \dot{x}_2 = f_2(x_1, x_2) \end{cases}$$

Unde:

$$f_1(\mathbf{x}) = x_2$$
$$f_2(\mathbf{x}) = -kx_1 - k_1 x_2$$

Relația (60b) devine:

$$\tilde{\mathbf{f}}(\mathbf{x}) = -h\mathbf{f}(\mathbf{x}) + c_0\mathbf{x} + \sum_{l=1}^{k} c_l \mathbf{x}^{i+1-l} \text{ , sau}$$

$$\tilde{f}_1(\mathbf{x}) = -hx_2 + c_0 x_1 + \sum_{l=1}^{k} c_l x_1^{i+1-l}$$

$$\tilde{f}_2(\mathbf{x}) = hkx_1 + hk_1 x_2 + c_0 x_2 + \sum_{l=1}^{k} c_l x_2^{i+1-l}$$

Jacobianul $\mathbf{F}$ lui $\tilde{\mathbf{f}}$ va fi dat de:

$$F_{11} = c_0 \qquad F_{12} = -h$$
$$F_{21} = hk \qquad F_{22} = hk_1 + c_0$$

Observație – Cod în **`ANA\BDF`***:*

- Funcția **f**:

În cod: f = `ff`; $\mathbf{x}$ = `y(1:2)`; $\tilde{\mathbf{f}}$ = `f_bdf`;
Tabloul `xa(-k+1:1, 1:2)` conține valorile $\mathbf{x}^j$;
Tabloul `c(1:6, 0:6)` conține coeficienții c_{kl}. Exemplu: c_{k0} = `c(k,0)`;
"k" desemnează ordinul metodei BDF: $1 \le k \le 6$.

```
ff(1) =y(2)
ff(2) =-100*y(1) -101*y(2)


f_bdf =-h*ff(:) +c(k,0)*y(:)

do i =1, k
f_bdf =f_bdf +c(k, i)*xa(-i+1,:)
enddo
```

■

- Jacobianul **F**:

În cod: **F** = `jf`

```
jf(1,1) =c(k,0); jf(1,2) =-h;
jf(2,1) =h*100;  jf(2,2) =c(k,0) +h*101
```

■