Cap. 6. Rezolvarea numerică a ecuaţiilor şi sistemelor de ecuaţii diferenţiale ordinare

6.1. Aspecte introductive

Studiul *comportamentului dinamic* al sistemelor fizice → *modele matematice* sub forma ecuaţiilor sau sistemelor de ecuaţii diferenţiale ordinare, liniare sau neliniare.

Aspecte introductive (cont'd 1)

Însă uneori nu sunt cunoscute expresiile funcţiilor care definesc derivatele, iar alteori aceste expresii sunt complicate → nu pot fi rezolvate pe cale analitică, prin metode clasice.

De ordinul întâi ! Cu generalizări la sisteme şi la ordin superior.

Problema de rezolvare a unei ecuaţii diferenţiale de ordinul întâi:

Aspecte introductive (cont'd 2)

$\frac{dy}{dx} = f(x, y)$, $f : [a,b] x\, I \to R$, $[a,b], I \subset R$, a = x_0 , (1.1)

(se notează y' = $\frac{dy}{dx}$, I – interval) , cu condiţia iniţială

$y_0 = y(x_0)$. (1.2)

Se cere să se determine expresia functiei y(x) care verifică (1.1) şi (1.2) – ***problemă de tip Cauchy***.

Aspecte introductive (cont'd 3)

Forma implicită a ecuaţiei diferenţiale ordinare (1.1):

$$F(x, y, y') = 0,\ F: [a,b]\, x\, I_1\, x\, I_2 \to R,\ [a,b], I_1, I_2 \subset R, \qquad (1.3)$$

cu I_1 şi I_2 – intervale.

Se presupune că a fost efectuat în prealabil un studiu al problemei enunţate, constatându-se **existenţa şi unicitatea soluţiei**.

Aspecte introductive (cont'd 3)

În continuare, **pentru determinarea aproximativă a soluţiei** se poate proceda în două **moduri**:

a) Este căutată o funcţie z(x) care să aproximeze cât mai bine pe y(x) pentru x$\in\left[a,b\right]$ = ***rezolvare aproximativă analitică***.

Aspecte introductive (cont'd 4)

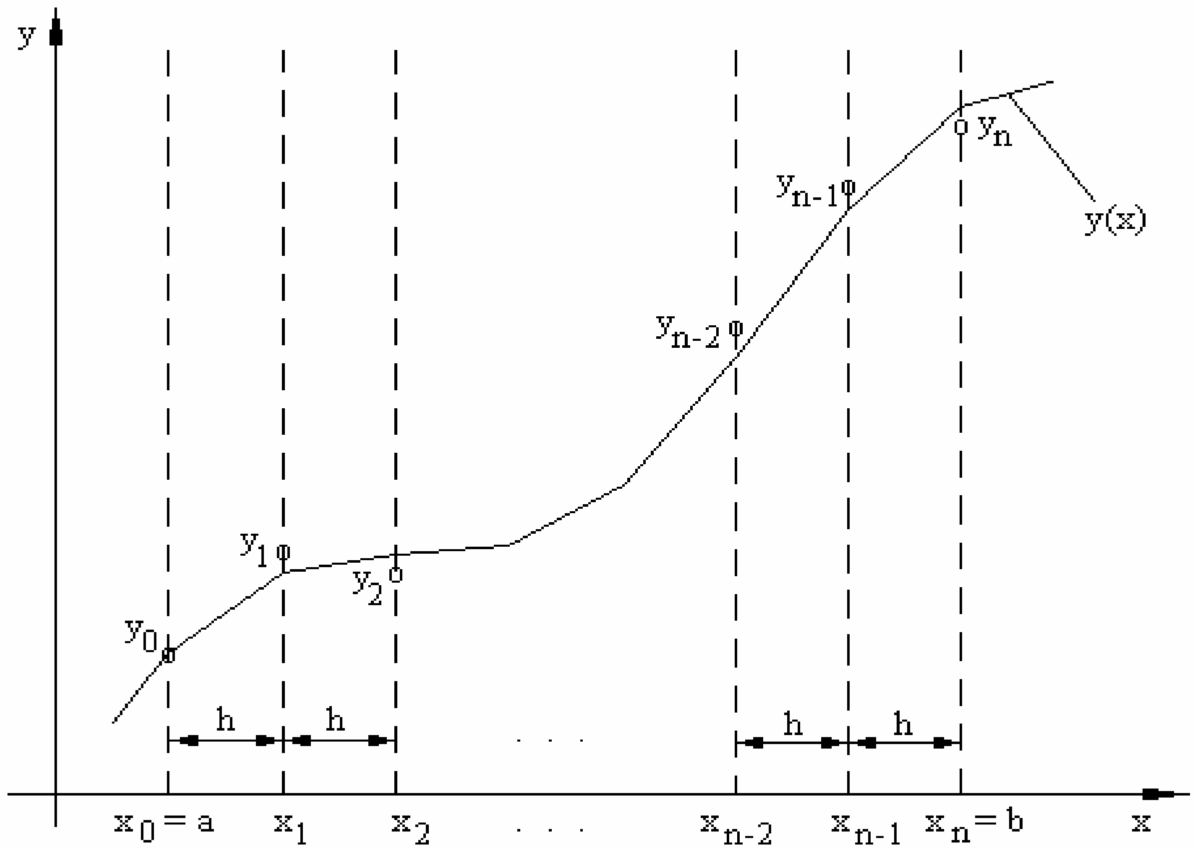
b) ***Metodă numerică propriu-zisă*** – sunt determinate valorile y_1, y_2, …, y_n care să aproximeze cât mai bine valorile exacte $y(x_1)$, $y(x_2)$, …, $y(x_n)$, ale lui $y(x)$ pentru $x \in [a, b]$ dacă punctele x_1, x_2, …, $x_n \in [a, b]$ sunt considerate echidistante:

$$x_{i+1} - x_i = h\ ,\ \ i = 0 \ldots n-1\ , \tag{1.4}$$

cu h – **pasul de discretizare** (**de integrare**) şi capetele intervalului căruia îi aparţine variabila independentă x:

$x_0 = a$, $x_n = b$.

Aspecte introductive (cont'd 5)



Aspecte introductive (cont'd 6)

Doar de tip b) ! – determinarea valorilor aproximative y_1, y_2, …..y_n – folosind relaţii de tip:

$$y_i = y_{i-1} + h \cdot g(x_j, y_j, h)\ ,\ \ i = 1 \dots n\ ,\ \ j = 1 \dots i-1\ . \qquad (1.5)$$

Categorii de metode de integrare numerică după numărul de puncte utilizate anterior punctului curent (x_i,y_i):

1) metode **monopas** (**cu paşi separaţi**) – la determinarea lui y_i utilizează informaţiile referitoare numai la un singur punct anterior, corespunzător lui x_{i-1};

2) metode **multipas** (**cu paşi legaţi**) – la determinarea lui y_i utilizează informaţiile referitoare la mai multe puncte anterioare, corespunzătoare lui x_{i-1}, x_{i-2}, ...

Ambele categorii pot utiliza:

- **algoritmi expliciţi** (**direcţi**) – punctul curent nu apare în expresia funcţiei g;
- **algoritmi impliciţi** (**iterativi**, de tip **predictor-corector**) – punctul curent apare în expresia lui g.

6.2. Metode monopas pentru ecuaţii diferenţiale

Trăsătură caracteristică: la calculul valorilor aproximative y_i, $i=1 \ldots n$, se folosesc numai informaţiile din punctul anterior, adică (x_{i-1}, y_{i-1}), cu (1.5) particularizată:

$$y_i = y_{i-1} + h \cdot g(x_{i-1}, y_{i-1}, h)\ ,\ \ i = 1 \ldots n\ . \qquad (2.1)$$

Metodele se diferenţiază între ele prin forma funcţiei g. Dar toate sunt bazate pe **dezvoltarea în serie Taylor** în vecinătatea lui x_{i-1}:

Metode monopas (cont'd 1)

$$y(x_i) = y(x_{i-1}) + \frac{h}{1!} y'(x_{i-1}) + \frac{h^2}{2!} y''(x_{i-1}) + \ldots \qquad (2.2)$$

şi reţinerea unui anumit număr de termeni din (2.2).

Algoritmii expliciţi determină valorile y_i, i = 1 ... n, prin efectuarea unui număr finit de operaţii aritmetice elementare aplicând direct o relaţie de tip (2.1).

Metode monopas (cont'd 2)

Algoritmii predictor-corector determină valorile y_i, $i=1…n$, printr-un **proces de calcul iterativ** cu convergenţă teoretic infinită, dar practic finită – **etape**:

a) se iniţializează valoarea lui y_i:

$$y_i^0 = y_{i-1} + h \cdot g_p(x_{i-1}, y_{i-1}, h) \; ; \tag{2.3}$$

b) la un pas oarecare k = 1, 2, 3, … al procesului iterativ de calcul se determină noua valoare a lui y_i:

$$y_i^k = y_{i-1} + h \cdot g_c(x_{i-1}, y_{i-1}, x_i, y_i^{k-1}, h) \; ; \tag{2.4}$$

Metode monopas (cont'd 3)

c) calculul este terminat când y_i a fost determinat cu o precizie impusă / dorită:

$$\left|y_i^k - y_i^{k-1}\right| \le \varepsilon , \qquad (2.5)$$

cu eroarea ε > 0 prestabilită.

Relaţia (2.3), aplicată o singură dată = formula de predicţie (predictor). Relaţia (2.4), aplicată în mod repetat până la atingerea preciziei dorite = formula de corecţie (corector).

Metodele de tip Euler

Trăsătură caracteristică: **metode monopas cu algoritm explicit** la care din dezvoltarea în serie Taylor (2.2) sunt reţinuţi numai primii doi termeni:

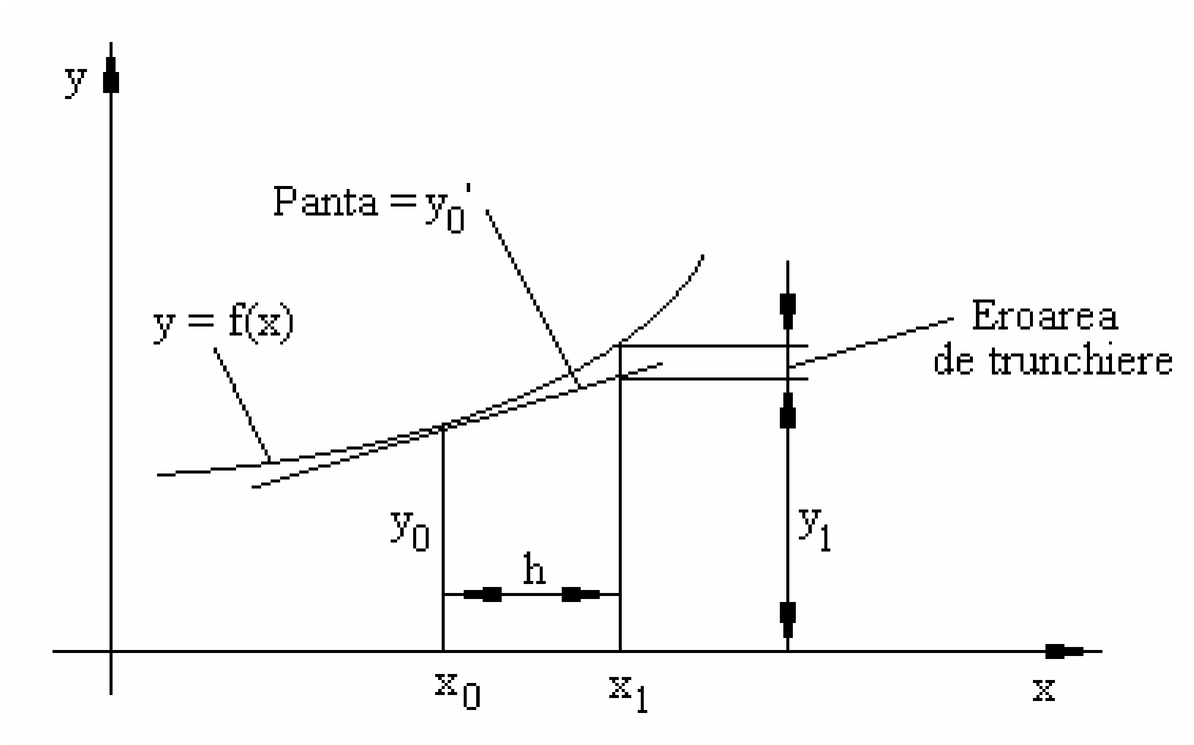
$$y_i = y_{i-1} + h \cdot y'_{i-1} \,. \tag{2.6}$$

→ pentru **versiunea clasică a metodei Euler**:

$$y_i = y_{i-1} + h \cdot f(x_{i-1}, y_{i-1}) \,. \tag{2.7}$$

Algoritm de rezolvare: pornind de la condiţia iniţială (1.2), se aplică succesiv (2.7) pentru i = 1 … n rezultând toate valorile căutate (ilustrare pentru i = 1).

Metodele de tip Euler (cont'd 1)



Exemplu: Se consideră ecuaţia diferenţială ordinară

Metodele de tip Euler (cont'd 2)

$y' = f(x, y) = y - x + 2$, cu condiţia iniţială

$y_0 = y(x_0) = y(0) = 0$.

Să se rezolve numeric pentru $x \in [0,1]$, cu pasul de discretizare $h=0.1$ ($n = 10$) utilizând versiunea clasică a metodei Euler.

Soluţie: Sunt particularizate relaţiile (1.4), (2.6) şi (2.7) $\Rightarrow$

$$f_0 = f(x_0, y_0) = f(0,0) = 0 - 0 + 2 = 2\,;$$

$$x_1 = 0 + h = 0 + 0.1 = 0.1\,;$$

$$y_1 = y_0 + hf_0 = 0 + 0.1 \cdot 2 = 0.2\,;$$

Metodele de tip Euler (cont'd 2)

$$f_1 = f(x_1, y_1) = f(0.1, 0.2) = 0.2 - 0.1 + 2 = 2.1;$$
$$x_2 = x_1 + h = 0.1 + 0.1 = 0.2;$$
$$y_2 = y_1 + hf_1 = 0.2 + 0.1 \cdot 2.1 = 0.41;$$
$$f_2 = f(x_2, y_2) = f(0.2, 0.4) = 0.41 - 0.2 + 2 = 2.21;$$
$$x_3 = x_2 + h = 0.2 + 0.1 = 0.3;$$
$$y_3 = y_2 + hf_2 = 0.41 + 0.1 \cdot 2.21 = 0.631;$$
$$f_3 = f(x_3, y_3) = f(0.3, 0.631) = 0.631 - 0.3 + 2 = 2.331, \ldots$$

Exerciţiu: calcule pentru i = 4 … 10 şi rezolvare în cazurile h=0.05 (n=20) şi h=0.025 (n=40).

Avantaj: simplă; *dezavantaj:* puţin precisă → recomandată doar în cazul rezolvărilor rapide aproximative.

Versiunea Cauchy a metodei Euler

Avantaj: îmbunătăţirea preciziei şi stabilităţii numerice; *dezavantaj:* creşterea volumului de calcule.

Trăsătură caracteristică: înlocuirea relaţiei (2.7) cu:

$$y_i = y_{i-1} + h \cdot f\left[x_{i-1} + 0.5h, y_{i-1} + 0.5h \cdot f(x_{i-1}, y_{i-1})\right], \qquad (2.8)$$

adică y' nu se mai aproximează pe intervalul $[x_{i-1,} x_i]$ cu valoarea de la începutul intervalului, ci cu o aproximaţie a valorii de la *mijlocul* acestui interval.

Versiunea Cauchy a metodei Euler (cont'd 1)

Exemplu: Să se rezolve ecuaţia diferenţială din cadrul exemplului anterior în aceleaşi condiţii cu versiunea Cauchy a metodei Euler.

Soluţie: Sunt obţinute succesiv următoarele rezultate:

$$f_0 = f(x_0, y_0) = f(0,0) = 0 - 0 + 2 = 2;$$

$$x_1 = 0 + h = 0 + 0.1 = 0.1;$$

$$f(0 + 0.5h, 0 + 05hf_0) = f(0 + 0.5 \cdot 0.1, 0 + 0.5 \cdot 0.1 \cdot 2) = 2.05;$$

$$y_1 = y_0 + h \cdot 2.05 = 0 + 0.1 \cdot 2.05 = 0.205;$$

Versiunea Cauchy a metodei Euler (cont'd 2)

$$f_1 = f(x_1, y_1) = f(0.1, 0.205) = 0.205 - 0.1 + 2 = 2.105;$$

$$x_2 = x_1 + h = 0.1 + 0.1 = 0.2;$$

$$f(x_1 + 0.5h, y_1 + 0.5hf_1) = f(0.1 + 0.5 \cdot 0.1, 0.205 + 0.5 \cdot 0.1 \cdot 2.105) = 2.1602;$$

$$y_2 = y_1 + h \cdot 2.1602 = 0.205 + 0.1 \cdot 2.1602 = 0.421.$$

Exerciţiu: calcule pentru i = 3 … 10 şi rezolvarea exemplului în cazurile h=0.05 şi h=0.025.

Versiunea Cauchy a metodei Euler (cont'd 3)

Dezavantaje ale metodelor de tip Euler:

1) Număr mare de calcule în situaţia în care intervalul [a, b] este relativ larg deoarece este nevoie de un număr foarte mare de paşi de discretizare pentru a acoperi intervalul.
2) Erorile făcute la fiecare pas (figura !) se pot acumula şi propaga imprevizibil !

Metodele de tip Runge-Kutta

Sunt **metode monopas cu algoritm explicit**.

Avantaj: asigură îmbunătăţirea în continuare a preciziei şi a erorii de trunchiere pe un pas de integrare.

Trăsătură caracteristică: pentru reducerea erorii, la determinarea lui y_i, i = 1, 2, …, n **se calculează valorile lui f(x,y) într-un număr de puncte intermediare ale intervalului** $[x_{i-1}, x_i]$, acest număr de puncte fiind legat direct de ordinul p al metodei.

Forma generală a metodelor de tip Runge-Kutta:

$$k_1 = h \cdot f(x_{i-1}, y_{i-1}) \, , \qquad (2.9)$$

Metodele de tip Runge-Kutta (cont'd 1)

$$k_j = h \cdot f(x_{i-1} + h \cdot b_j,\ y_{i-1} + \sum_{m=1}^{j-1} c_{jm} k_m)\ ,\ \ j = 2 \ldots p\ , \qquad (2.10)$$

$$y_i = y_{i-1} + \sum_{m=1}^{p} a_m k_m\ . \qquad (2.11)$$

Metodele de tip Runge-Kutta (cont'd 2)

Determinarea coeficienţilor a_m, $m = 1 \dots p$, b_j, $j = 2 \dots p$, şi c_{jm}, $m = 1 \dots j-1$, $j = 2 \dots p$ – prin dezvoltarea în serie Taylor a ambilor membri ai relaţiei (2.11) şi identificarea coeficienţilor expresiilor obţinute.

Exemplu: **metoda Runge-Kutta de ordinul 4** – algoritmul este formulat utilizând relaţiile (2.12) … (2.16), prezentate în ordinea de efectuare a calculelor:

Metodele de tip Runge-Kutta (cont'd 3)

$$k_1 = f(x_{i-1}, y_{i-1}), \qquad (2.12)$$

$$k_2 = f(x_{i-1} + 0.5h, y_{i-1} + 0.5h \cdot k_1), \qquad (2.13)$$

$$k_3 = f(x_{i-1} + 0.5h, y_{i-1} + 0.5h \cdot k_2), \qquad (2.14)$$

$$k_4 = f(x_{i-1} + 0.5h, y_{i-1} + 0.5h \cdot k_3), \qquad (2.15)$$

$$y_i = y_{i-1} + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4). \qquad (2.16)$$

Alte variante de metode de tip Runge-Kutta cu proprietăţi avantajoase: Runge-Kutta-Merson, Ralston-Runge-Kutta şi Butcher-Runge-Kutta.

Adaptarea mărimii pasului de integrare

În subdomenii ale intervalului [a, b] în care funcţia are **variaţii line** → poate fi utilizat un pas relativ mare. Dacă însă, există subdomenii în care au loc **variaţii rapide** ale lui y pentru variaţii relativ mici ale lui x → este necesar un pas mic. Dar: există funcţii pentru care pot fi prezente ambele tipuri de subdomenii menţionate; pentru acestea: în locul utilizării unui pas de integrare mic pe întreg intervalul [a, b] este preferată o soluţie mai eficientă, de ***adaptare automată*** a mărimii pasului de integrare în funcţie de valoarea gradientului funcţiei necunoscute, y'(x).

6. 3. Metode multipas pentru ecuaţii diferenţiale

Se consideră din nou ecuaţia diferenţială de ordinul întâi (1.1) cu condiţia iniţială (1.2) şi se cere să se rezolve această ecuaţie, adică să se determine expresia functiei y(x) care verifică (1.1) şi (1.2). Se pune problema rezolvării ecuaţiei menţionate printr-o metodă numerică propriu-zisă, adică se cere să se determine valorile y_1, y_2, ..., y_n care să aproximeze cât mai bine valorile exacte $y(x_1)$, $y(x_2)$, ..., $y(x_n)$, ale lui y(x) pentru $x \in [a,b]$ dacă punctele x_1, x_2, ..., $x_n \in [a,b]$ sunt considerate echidistante cu pasul de integrare h (a se vedea (1.4)).

Metode multipas (cont'd 1)

Trăsătură caracteristică: la calculul valorilor aproximative y_i, $i=1\ldots n$, se folosesc, spre deosebire de cazul metodelor monopas, **informaţiile din mai multe puncte anterioare punctului curent (x_i, y_i)** – pe baza formei particulare a relaţiei (1.5):

$$y_i=y_{i-1}+h\cdot g(x_{i-r}, y_{i-r}, \ldots, x_{i-2}, y_{i-2}, x_{i-1}, y_{i-1}, h),\ i = 1\ldots n\ , \qquad (3.1)$$

cu $r \in N$, $r \geq 2$ – numărul de puncte anterioare utilizate.

Metodele sunt diferenţiate între ele prin forma funcţiei g şi valoarea lui r, dar toate sunt bazate pe **utilizarea unor integrale** de forma

Metode multipas (cont'd 2)

$$y(x_i) - y(x_{i-r}) = \int_{x_{i-r}}^{x_i} y'(x)\,dx\ , \qquad (3.2)$$

în care funcţia $y'(x)$ este aproximată cu un polinom de interpolare.

Algoritmii expliciţi determină valorile y_i, $i = 1 \ldots n$, prin efectuarea unui număr finit de operaţii aritmetice elementare aplicând direct o relaţie de tip (3.1).

Metode multipas (cont'd 3)

Algoritmii predictor-corector determină valorile y_i, i=1…n, printr-un **proces de calcul iterativ** cu convergenţă teoretic infinită, dar practic finită – etape:

(a) se iniţializează valoarea lui y_i:

$$y_i^0 = y_{i-1} + h \cdot g_p(x_{i-r}, y_{i-r}, \ldots, x_{i-2}, y_{i-2}, x_{i-1}, y_{i-1}, h); \qquad (3.3)$$

(b) la un pas oarecare k = 1, 2, 3, … al procesului iterativ de calcul se determină noua valoare a lui y_i:

$$y_i^k = y_{i-1} + h \cdot g_c(x_{i-r}, y_{i-r}, \ldots, x_{i-2}, y_{i-2}, x_{i-1}, y_{i-1}, x_i, y_i^{k-1}, h); \qquad (3.4)$$

Metode multipas (cont'd 4)

(c) calculul este terminat când y_i a fost determinat cu o precizie impusă / dorită:

$$\left|y_i^k - y_i^{k-1}\right| \le \varepsilon , \qquad (3.5)$$

cu eroarea $\varepsilon > 0$ prestabilită.

Relaţia (3.3), aplicată o singură dată = fomula de predicţie (predictor); relaţia (3.4), aplicată în mod repetat până la atingerea preciziei dorite = formula de corecţie (corector).

Metode multipas (cont'd 5)

Avantaje comparativ cu metodele monopas:

- estimarea erorii de trunchiere este relativ simplă, eroarea de trunchiere fiind semnificativ mai mică;
- propagarea erorilor este mai redusă, fiind îmbunătăţite precizia şi stabilitatea numerică (se va reveni);
- nu este necesar calculul valorilor funcţiei f(x, y) în puncte intermediare suplimentare faţă de cele de tip (1.4).

Metode multipas (cont'd 6)

Dezavantajele metodelor multipas faţă de cele monopas:

- nu este asigurată autopornirea deoarece la primii paşi nu sunt disponibile informaţiile din punctele anterioare → se utilizează de regulă pentru **pornire** metode monopas cu eroare de trunchiere de acelaşi ordin de mărime;
- modificarea pasului de integrare h (este vorba în primul rând de reducerea acestuia, efectuată în vederea creşterii preciziei) se face relativ dificil, fiind necesare reveniri la puncte deja determinate sau reporniri cu metode monopas;
- la unele variante poate creşte volumul de calcule.

Metodele de tip Adams-Bashforth-Moulton

Sunt **metode multipas cu algoritm explicit**. Pentru metodele de tip Adams-Bashforth-Moulton ***de ordinul m***:

- relaţia (3.2) este particularizată pentru r = 1;
- integrandul din (3.2) $y'(x) = f(x, y)$ este aproximat cu un polinom de interpolare Lagrange $P_{m-1}(x)$ de gradul m–1, definit prin intermediul a m perechi de valori:

$$(x_j, f(x_j, y_j)) \ , \ \ j = i-m+1, i-m+2, \ldots , i \ . \qquad (3.6)$$

Metodele de tip Adams-Bashforth-Moulton (cont'd 1)

Algoritmul metodei Adams-Bashforth-Moulton de ordinul m = 4 (de tip predictor-corector) – **etape**, prin care se determină succesiv valorile lui y_i, i = 1 … n, pe baza unor procese de calcul iterativ:

1) Se iniţializează y_i utilizând formula de predicţie (de tip (3.3)):

$$y_i^0=y_{i-1}+h(55f(x_{i-1},y_{i-1})-59f(x_{i-2},y_{i-2})+37f(x_{i-3},y_{i-3})-9f(x_{i-4},y_{i-4}))/24 \,, \tag{3.7}$$

unde indicele superior corespunde numărului iteraţiei curente.

Metodele de tip Adams-Bashforth-Moulton (cont'd 2)

2) La un pas oarecare k, k = 1, 2, 3, … al procesului iterativ de calcul se determină noua valoare a lui y_i utilizând formula de corecţie (de tip (3.4)):

$$y_i^k = y_{i-1} + h(9f(x_i, y_i^{k-1}) + 19f(x_{i-1}, y_{i-1}) - 5f(x_{i-2}, y_{i-2}) + f(x_{i-3}, y_{i-3}))/24 \,. \tag{3.8}$$

3) Calculul este terminat când y_i a fost determinat cu o precizie impusă / dorită (condiţia de tip (3.5)):

$$\left| y_i^k - y_i^{k-1} \right| \le \varepsilon \,, \tag{3.9}$$

Metodele de tip Adams-Bashforth-Moulton (cont'd 3)

cu eroarea $\varepsilon > 0$ prestabilită.

Algoritmul este pornit pentru i = 4, deci valorile $f(x_0,y_0)$, $f(x_1,y_1)$, $f(x_2,y_2)$ şi $f(x_3,y_3)$ trebuie cunoscute înainte de pornire. În practică, aceste valori trebuie obţinute printr-o procedură de *autopornire* similară celor din cazul metodelor de tip Runge-Kutta.

6.4. Rezolvarea ecuaţiilor diferenţiale ordinare de ordin superior

Se consideră ecuaţia diferenţială de ordinul *n*:

$y^{(n)} = f(x, y, y', y'', ..., y^{(n-1)})$, (3.10)

cu $x \in [a, b]$, în condiţiile iniţiale

Ecuaţii diferenţiale ordinare de ordin superior

$$\begin{cases} y(x_0) = y_0 \\ y'(x_0) = y'_0 \\ y''(x_0) = y''_0 \\ \dots \\ y^{(n-2)}(x_0) = y^{(n-2)}{}_0 \\ y^{(n-1)}(x_0) = y^{(n-1)}{}_0 \end{cases} . \qquad (3.11)$$

Se introduc notaţiile

$$\begin{cases} y_1(x) = y^{(n-1)} \\ y_2(x) = y^{(n-2)} \\ y_3(x) = y^{(n-3)} \\ \dots \\ y_{n-1}(x) = y' \\ y_n(x) = y \end{cases} . \qquad (3.12)$$

Ecuaţii diferenţiale ordinare de ordin superior (cont'd 1)

Efectuarea substituţiilor în (3.10) → ecuaţia diferenţială de ordinul *n* (3.10) este echivalentă cu următorul sistem de *n* ecuaţii diferenţiale ordinare de ordinul întâi:

$$\begin{cases} y_1' = f(x, y_n, y_{n-1}, \ldots, y_1) \\ y_2' = y_1 \\ y_3' = y_2 \\ \ldots \\ y_{n-2}' = y_{n-3} \\ y_{n-1}' = y_{n-2} \end{cases} . \tag{3.13}$$

Ecuaţii diferenţiale ordinare de ordin superior (cont'd 2)

Condiţiile iniţiale pentru sistemul (3.13) – din (3.11) şi (3.12):

$$\begin{cases} y_1(x_0) = y^{(n-1)}{}_0 \\ y_2(x_0) = y^{(n-2)}{}_0 \\ y_3(x_0) = y^{(n-3)}{}_0 \\ \ldots \\ y_{n-1}(x_0) = y'{}_0 \\ y_n(x_0) = y_0 \end{cases} . \tag{3.14}$$

Metode de soluţionare similare celor de la ecuaţii !

6.5. Aspecte privind stabilitatea numerică şi alegerea metodelor de rezolvare numerică a ecuaţiilor diferenţiale

Pentru definirea stabilităţii numerice a unui algoritm este nevoie pentru început să se discute despre analiza condiţionării problemei asociate algoritmului.

Stabilitatea numerică şi alegerea (cont'd 1)

Analiza condiţionării unei probleme = proces matematic relativ complicat, strâns legat de teoria perturbaţiilor. O problemă este ***bine condiţionată*** dacă mici variaţii în datele problemei provoacă doar mici variaţii în soluţie. În caz contrar → problema este ***rău condiţionată***.

De regulă este studiată condiţionarea problemelor numerice din cadrul algebrei liniare, obţinându-se **numere de condiţionare**, calculabile sau estimabile.

Stabilitatea numerică şi alegerea (cont'd 2)

Analiza condiţionării pentru alte probleme – în particular, pentru rezolvarea numerică a ecuaţiilor şi sistemelor de ecuaţii diferenţiale ordinare – devine însă dificilă şi necesită un *efort relativ mare*.

În sens restrâns, se spune că un algoritm este ***numeric stabil*** dacă nu introduce o sensibilitate mai mare în raport cu datele decât cea inerentă problemei, adică nu înrăutăţeşte condiţionarea problemei asociate algoritmului. Altfel spus, un algoritm este numeric stabil dacă rezultatul calculat de acesta este – sau este apropiat de – soluţia exactă a unei mici perturbaţii a problemei iniţiale.

Stabilitatea numerică şi alegerea (cont'd 3)

Dacă problema numerică este bine condiţionată → soluţia calculată cu un algoritm numeric stabil este apropiată de soluţia exactă. Pentru mulţi algoritmi – în particular, pentru cei destinaţi rezolvării numerice a ecuaţiilor şi sistemelor de ecuaţii diferenţiale ordinare – stabilitatea numerică poate fi demonstrată matematic şi exprimată sub forma unor **condiţii de stabilitate**.

Observaţii cu referire la stabilitatea numerică a algoritmilor destinaţi rezolvării numerice a ecuaţiilor diferenţiale ordinare:

Stabilitatea numerică şi alegerea (cont'd 4)

1. În cazul metodelor de tip **Euler** – condiţia de stabilitate:

$$-2 < h \cdot \partial f/\partial y < 0 \,. \tag{4.1}$$

Concluzii:

A. Este necesar ca $\partial f/\partial y < 0$ pentru asigurarea stabilităţii numerice.

B. Poate apare instabilitate numerică pentru anumite valori ale pasului de integrare h.

Stabilitatea numerică şi alegerea (cont'd 4)

2. În cazul metodelor de tip **Runge-Kutta** – aceleaşi concluzii, rezultate din condiţia de stabilitate:

$$M < h \cdot \partial f / \partial y < 0 , \qquad (4.2)$$

în care parametrul M < 0 poate fi estimat şi diferă de la o metodă de tip Runge-Kutta la alta.

Pentru asigurarea stabilităţii numerice se poate proceda la *reducerea valorii pasului de integrare h până la o valoare suficient de mică*.

Stabilitatea numerică şi alegerea (cont'd 5)

Însă metodele de tip Runge-Kutta sunt mai puţin eficiente decât cele de tip predictor-corector datorită numărului mai mare de evaluări ale funcţiei la fiecare pas; deci, o reducere prea substanţială a valorii lui h determină creşterea considerabilă a volumului de calcule legate de evaluările funcţiei.

→ cel puţin din motivul asigurării stabilităţii numerice este nevoie de **ajustarea cu grijă a pasului de integrare** pe parcursul procesului de soluţionare e ecuaţiei diferenţiale.

Stabilitatea numerică şi alegerea (cont'd 6)

3. În cazul metodelor de tip **Adams-Bashforth-Moulton** – concluziile A. şi B. de la metodele de tip Euler. Condiţia de stabilitate:

$$-1.25 < h\cdot\partial f/\partial y < 0\ . \qquad (4.3)$$

(4.3) este din nou utilizabilă la estimarea mărimii pasului de integrare h.

Stabilitatea numerică şi alegerea (cont'd 7)

Alegerea metodei numerice de integrare a ecuaţiilor diferenţiale ordinare este strâns legată de ***analiza erorilor de calcul*** în procesul de rezolvare a ecuaţiilor diferenţiale ordinare.

Remember: Erorile de calcul pot fi de trei tipuri – erori de trunchiere, erori de rotunjire şi erori de propagare – fiind importantă analiza efectului fiecărui tip de eroare în parte asupra preciziei rezultatelor, scopul fiind evident cel de reducere a erorilor de calcul.

Stabilitatea numerică şi alegerea (cont’d 8)

Erorile de trunchiere depind de *metoda* numerică aleasă. Pentru metodele numerice analizate aici – ordine de mărime:

- ♦ h^2 pentru versiunea clasică a metodelor de tip Euler şi h^3 pentru versiunea Cauchy a metodelor de tip Euler;
- ♦ h^5 pentru metodele de tip Runge-Kutta de ordinul 4;
- ♦ h^5 pentru metodele de tip Adams-Bashforth-Moulton de ordinul 4.

Stabilitatea numerică şi alegerea (cont'd 9)

Erorile de rotunjire depind de posibilităţile *echipamentului* de calcul pe care sunt implementaţi algoritmii de rezolvare a metodelor numerice.

Aceste erori pot fi diminuate dacă se utilizează modul de lucru în **dublă precizie**. Cu toate acestea, erorile de rotunjire cresc pe măsura creşterii numărului de paşi de integrare datorită propagării erorilor de la un pas la altul.

Stabilitatea numerică şi alegerea (cont'd 10)

Erorile de propagare: efectul lor este simţit mai ales la metodele de tip **predictor-corector**. Este important să fie utilizate numai metode numeric stabile, la care erorile nu se propagă în mod imprevizibil sau chiar nemărginit.

Stabilitatea numerică şi alegerea (cont'd 11)

Pentru reducerea erorilor de trunchiere este recomandată micşorarea pasului de integrare, însă aceasta conduce la creşterea erorii de rotunjire. Pe de altă parte, micşorarea pasului de integrare conduce la creşterea numărului de paşi, cu consecinţa imediată a creşterii timpului de calcul. Deci, trebuie căutată – ca o ***soluţie de compromis*** – acea valoare a pasului de integrare pentru care erorile de calcul ("suma" celor trei tipuri de erori menţionate) să fie cât mai mici.

Stabilitatea numerică şi alegerea (cont'd 12)

Ţinând seama legătura dintre mărimea pasului de integrare şi cea a erorilor de calcul, ***alegerea unei anumite metode de rezolvare numerică a ecuaţiilor diferenţiale ordinare*** reprezintă o problemă relativ complexă – **recomandări**:

- Dacă sunt cerute *rezolvări rapide* → poate fi aleasă o metodă simplă, de tip Euler, cu un pas de integrare relativ mic, dar cu o valoare absolută mult superioară celui mai mic număr posibil a fi reprezentat în echipamentul de calcul.

Stabilitatea numerică şi alegerea (cont'd 13)

- Dacă sunt cerute *rezolvări precise fără a fi important timpul de calcul* → poate fi aleasă metoda Runge-Kutta de ordinul 4 datorită avantajelor privind autopornirea.

- Dacă sunt cerute *rezolvări precise fără modificări semnificative ale pasului de integrare* → poate fi aleasă alege metoda Adams-Bashforth-Moulton de ordinul 4 datorită avantajelor privind stabilitatea numerică.

Stabilitatea numerică şi alegerea (cont'd 14)

- La toate metodele numerice utilizate este *necesară analiza stabilităţii numerice* deoarece toate trebuie să fie numeric stabile. Dacă aceasta nu poate fi efectuată teoretic → trebuie testată în funcţie de aplicaţie.

+ experienţa utilizatorului !

+ analiza aplicaţiei !