

REZOLVAREA NUMERICĂ A ECUAȚILOR ȘI SISTEMELOR DE ECUAȚII DIFERENȚIALE ORDINARE

1. Aspecte introductive

Studiul *comportamentului dinamic* al sistemelor fizice \rightarrow *modele matematice* sub forma ecuațiilor sau sistemelor de ecuații diferențiale ordinare, liniare sau neliniare. Însă uneori nu se cunosc expresiile funcțiilor care definesc derivatele, iar alteori aceste expresii sunt complicate \rightarrow nu pot fi rezolvate pe cale analitică, prin metode clasice.

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

Problema de rezolvare a unei ecuații diferențiale de ordinul întâi:

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

(se notează $y' = \frac{dy}{dx}$, I – interval), cu condiția inițială:

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

Se cere să se determine expresia funcției $y(x)$ care verifică relațiile (1.1) și (1.2) – *problemă de tip Cauchy*.

Forma implicită a ecuației diferențiale ordinare (1.1):

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

cu I_1 și I_2 – intervale.

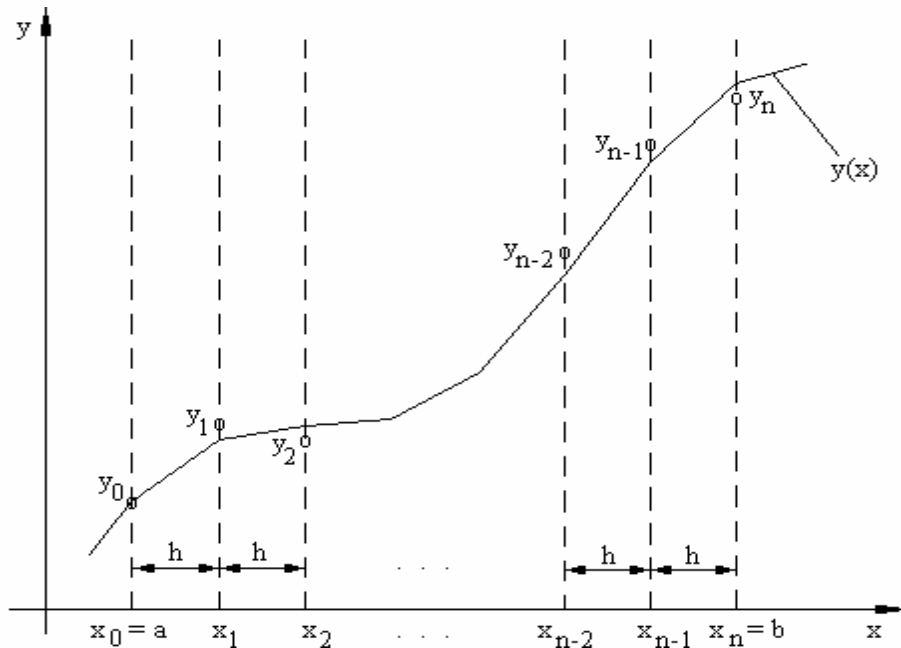
Se presupune că s-a efectuat în prealabil un studiu al problemei enunțate, constatându-se **existența și unicitatea soluției**. În continuare, **pentru determinarea aproximativă a soluției** se poate proceda în două **moduri**:

a) se caută o funcție $z(x)$ care să aproximeze cât mai bine pe $y(x)$ pentru $x \in [a, b]$ = *rezolvare aproximativă analitică*;

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

$$x_{i+1} - x_i = h, \quad i = 0 \dots n-1, \quad (1.4)$$

cu h – **pasul de discretizare (de integrare)** și capetele intervalului căruia îi aparține variabila independentă x sunt: $x_0 = a, x_n = b$.



Doar de tip b) ! – determinarea valorilor aproximative y_1, y_2, \dots, y_n se va face folosind relații de tip (1.5):

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

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

- 1) metode **monopas (cu pași separați)** – la determinarea lui y_i utilizează informațiile referitoare numai la un singur punct anterior, corespunzător lui x_{i-1} ;
- 2) metodele **multipas (cu pași legați)** – la determinarea lui y_i utilizează informațiile referitoare la mai multe puncte anterioare, corespunzătoare lui x_{i-1}, x_{i-2}, \dots .

Abele categorii pot utiliza:

- **algoritmi expliți (directi)** – punctul curent nu apare în expresia funcției g ;
- **algoritmi impliți (iterativi, de tip predictor-corector)** – punctul curent apare în expresia lui g .

2. Metode monopas pentru ecuații diferențiale

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

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

Metodele se diferențiază între ele prin forma funcției g , dar toate sunt bazate pe **dezvoltarea în serie Taylor** în vecinătatea lui x_i :

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

și reținerea unui anumit număr de termeni din (2.2).

Algoritmii expliți determină valorile y_i , $i = 1 \dots n$ prin efectuarea unui număr finit de operații aritmetice elementare aplicând direct o relație de tip (2.1).

Algoritmi predictor-corector determină valorile y_i , $i=1\dots n$ printr-un **proces de calcul iterativ** cu convergență teoretic infinită, dar practic finită – etpe:

a) se inițializează valoarea lui y_i :

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

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

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

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

$$|y_i^k - y_i^{k-1}| \leq \varepsilon , \quad (2.5)$$

cu eroarea $\varepsilon > 0$ prestabilită.

Relația (2.3), aplicată o singură dată = formula de predicție (predictor); relația (11.2.4), aplicată în mod repetat până la atingerea preciziei dorite = formula de corecție (corector).

Metodele de tip Euler

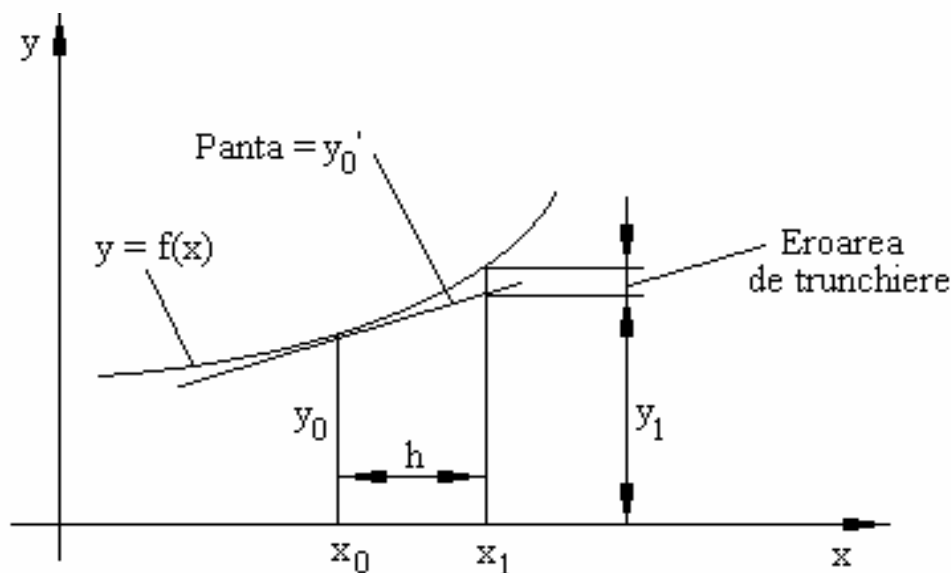
Trăsătură caracteristică: **metode monopas cu algoritmi explicit** la care din dezvoltarea în serie Taylor (2.2) se rețin numai primii doi termeni:

$$y_i = y_{i-1} + h \cdot y'_{i-1} . \quad (2.6)$$

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

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

Algoritm de rezolvare: pornind de la condiția inițială (1.2), se aplică succesiv (2.7) pentru $i = 1 \dots n$ rezultând toate valorile căutate (ilustrare pentru $i = 0$).



Exemplu: Se consideră ecuați diferențială ordinară:

$$y' = f(x, y) = y - x + 2 , \text{ cu condiția inițială:}$$

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

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

Soluție: Se particularizează relațiile (1.4), (2.6) și (2.7)

⇒

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

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

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

$$f_1 = f(x_1, y_1) = f(0.1, 0.2) = 0.2 - 0.1 + 2 = 2.1;$$

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

$$y_2 = y_1 + hf_1 = 0.2 + 0.1 \cdot 2.1 = 0.41;$$

$$f_2 = f(x_2, y_2) = f(0.2, 0.4) = 0.41 - 0.2 + 2 = 2.21;$$

$$x_3 = x_2 + h = 0.2 + 0.1 = 0.3;$$

$$y_3 = y_2 + hf_2 = 0.41 + 0.1 \cdot 2.21 = 0.631;$$

$$f_3 = f(x_3, y_3) = f(0.3, 0.631) = 0.631 - 0.3 + 2 = 2.331, \dots$$

Exercițiu: calcule pentru $i = 4 \dots 10$ și rezolvare în cazurile $h=0.05$ ($n=20$) și $h=0.025$ ($n=40$).

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

Versiunea Cauchy a metodei Euler

Avantaj: îmbunătățirea preciziei și stabilității numerice;
dezavantaj: creșterea volumului de calcule.

Trăsătură caracteristică: înlocuirea relației (2.7):

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

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

Exemplu: Să se rezolve ecuația diferențială din cadrul exemplului anterior în aceleași condiții cu versiunea Cauchy a metodei Euler.

Soluție: Se obțin succesiv următoarele rezultate:

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

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

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

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

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

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

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

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

Exercițiu: calcule pentru $i = 3 \dots 10$ și rezolvarea exemplului în cazurile $h=0.05$ și $h=0.025$.

Dezavantaje ale metodelor de tip Euler:

- 1) Număr mare de calcule în situația în care intervalul $[a, b]$ este relativ larg deoarece este nevoie de un număr foarte mare de pași de discretizare pentru a acoperi intervalul.

2) Erorile făcute la fiecare pas (figura !) se pot acumula și propaga imprevizibil !

Metodele de tip Runge-Kutta

Sunt **metode monopas cu algoritm explicit**.

Avantaj: asigură îmbunătățirea în continuare a preciziei și a erorii de trunchiere pe un pas de integrare.

Trăsătură caracteristică: pentru reducerea erorii, la determinarea lui y_i , $i = 1, 2, \dots, n$ 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:

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

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

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

Determinarea coeficienților a_m , $m = 1 \dots p$, b_j , $j = 2 \dots p$ și c_{jm} , $m = 1 \dots j-1$, $j = 2 \dots p$ se face prin dezvoltare în serie

Taylor a ambilor membri ai relației (2.11) și identificarea coeficienților expresiilor obținute.

Exemplu: **metoda Runge-Kutta de ordinul 4** – algoritm bazat pe utilizarea relațiilor (2.12) ... (2.16), prezentate în ordinea de efectuare a calculelor:

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

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

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

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

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

Alte variante de metode de tip Runge-Kutta cu proprietăți avantajoase: Runge-Kutta-Merson, Ralston – Runge-Kutta și Butcher-Runge-Kutta.

Adaptarea mărimii pasului de integrare

În subdomenii ale intervalului $[a, b]$ în care funcția are variații line \rightarrow poate fi utilizat un pas relativ mare. Dacă însă, există subdomenii în care au loc variații rapide ale lui y pentru variații relativ mici ale lui $x \rightarrow$ este necesar un pas mic. Dar: există funcții pentru care pot fi prezente ambele tipuri de

subdomenii menționate; pentru acestea: în locul utilizării unui pas de integrare mic pe întreg intervalul $[a, b]$ este preferată o soluție mai eficientă, de adaptare automată a mărimii pasului de integrare în funcție de valoarea gradientului funcției necunoscute, $y'(x)$.

3. Metode multipas pentru ecuații diferențiale

Se consideră din nou ecuația diferențială de ordinul întâi (1.1) cu condiția inițială (1.2) și se cere să se rezolve această ecuația, adică să se determine expresia funcției $y(x)$ care verifică (1.1) și (1.2). Se pune problema rezolvării ecuației menționate printr-o metodă numerică propriu-zisă, adică se cere să se determine valorile y_1, y_2, \dots, y_n care să aproximeze cât mai bine valorile exacte $y(x_1), y(x_2), \dots, y(x_n)$, ale lui $y(x)$ pentru $x \in [a, b]$ dacă punctele $x_1, x_2, \dots, x_n \in [a, b]$ sunt considerate echidistante cu pasul de integrare h (vezi (1.4)).

Trăsătură caracteristică: la calculul valorilor aproximative $y_i, i = 1 \dots n$ se folosesc, spre deosebire de cazul metodelor monopas, **informațiile din mai multe puncte anterioare punctului curent (x_i, y_i)** , relația (1.5) obținând forma particulară (3.1):

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

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

Metodele se diferențiază între ele prin forma funcției g și valoarea lui r , dar toate sunt bazate pe **utilizarea unor integrale** având forma (3.2):

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

unde funcția $y'(x)$ este aproximată cu un polinom de interpolare.

Algoritmii expliți determină valorile $y_i, i = 1 \dots n$ prin efectuarea unui număr finit de operații aritmetice elementare aplicând direct o relație de tip (3.1).

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

(a) se inițializează valoarea lui y_i :

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

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

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

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

$$|y_i^k - y_i^{k-1}| \leq \varepsilon, \quad (3.5)$$

cu eroarea $\varepsilon > 0$ prestabilită.

Relația (3.3), aplicată o singură dată = formula de predicție (predictor); relația (3.4), aplicată în mod repetat până la atingerea preciziei dorite = formula de corecție (corector).

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 îmbunătățite precizia și stabilitatea numerică (se va reveni);
- nu este necesar calculul valorilor funcției $f(x, y)$ în puncte intermediare suplimentare față de cele de tip (1.4).

Dezavantajele metodelor multipas față de cele monopas:

- nu este asigurată autopornirea deoarece la primii pași nu sunt disponibile informațiile din punctele anterioare → se utilizează de regulă pentru pornire metode monopas cu eroare de trunchiere de același ordin de mărime;
- modificarea pasului de integrare h (este vorba în primul rând de reducerea acestuia, efectuată în vederea creșterii

preciziei) se face relativ dificil, fiind necesare reveniri la puncte deja determinate sau reporniri cu metode monopas;

- la unele variante poate crește 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*:

- ❖ relația (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:

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

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

- 1) Se inițializează y_i utilizând formula de predicție (de tip (3.3)):

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

unde indicele superior corespunde numărului iterației curente.

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

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

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

$$|y_i^k - y_i^{k-1}| \leq \varepsilon, \quad (3.9)$$

cu eroarea $\varepsilon > 0$ prestabilită.

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

Rezolvarea ecuațiilor diferențiale ordinare de ordin superior

Se consideră ecuația diferențială de ordinul n :

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

cu $x \in [a, b]$, în condițiile inițiale (3.11):

$$\left\{ \begin{array}{l} y(x_0) = y_0 \\ y'(x_0) = y'_0 \\ y''(x_0) = y''_0 \\ \dots \\ y^{(n-2)}(x_0) = y^{(n-2)}_0 \\ y^{(n-1)}(x_0) = y^{(n-1)}_0 \end{array} \right. \quad (3.11)$$

Se introduc notațiile:

$$\left\{ \begin{array}{l} y_1(x) = y^{(n-1)} \\ y_2(x) = y^{(n-2)} \\ y_3(x) = y^{(n-3)} \\ \dots \\ y_{n-1}(x) = y' \\ y_n(x) = y \end{array} \right. \quad (3.12)$$

Prin efectuarea substituțiilor în (3.10), se obține că ecuația diferențială de ordinul n (3.10) este echivalentă cu următorul sistem de n ecuații diferențiale ordinare de ordinul întâi:

$$\left\{ \begin{array}{l} y'_1 = f(x, y_n, y_{n-1}, \dots, y_1) \\ y'_2 = y_1 \\ y'_3 = y_2 \\ \dots \\ y'_{n-2} = y_{n-3} \\ y'_{n-1} = y_{n-2} \end{array} \right. \quad (3.13)$$

Condițiile inițiale pentru sistemul (3.13) se obțin din relațiile (3.11) și (3.12):

$$\left\{ \begin{array}{l} y_1(x_0) = y^{(n-1)}_0 \\ y_2(x_0) = y^{(n-2)}_0 \\ y_3(x_0) = y^{(n-3)}_0 \\ \dots \\ y_{n-1}(x_0) = y'_0 \\ y_n(x_0) = y_0 \end{array} \right. \quad (3.14)$$

Metode de soluționare similare celor de la ecuații !

4. Aspecte privind stabilitatea numerică și alegerea metodelor de rezolvare numerică a ecuațiilor diferențiale

Pentru definirea stabilității numerice a unui algoritm este nevoie mai întâi să se discute despre analiza condiționării problemei asociate algoritmului. *Analiza condiționării* unei probleme = proces matematic relativ complicat, strâns legat de teoria perturbațiilor. O problemă este *bine condiționată* dacă mici variații în datele problemei provoacă doar mici variații în soluție. În caz contrar → problema este *rău condiționată*.

De regulă este studiată condiționarea problemelor numerice din cadrul algebrei liniare, obținându-se **numere de condiționare**, calculabile sau estimabile. Analiza condiționării pentru alte probleme – în particular, pentru rezolvarea numerică a ecuațiilor și sistemelor de ecuații diferențiale ordinare – devine însă dificilă și 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 înrăutățește condiționarea problemei asociate algoritmului. Altfel spus, un algoritm este numeric stabil dacă rezultatul calculat de acesta este – sau este apropiat de – soluția exactă a unei mici perturbații a problemei inițiale.

Dacă problema numerică este bine condiționată → soluția calculată de un algoritm numeric stabil este apropiată de soluția exactă. Pentru mulți algoritmi – în particular, pentru cei destinați rezolvării numerice a ecuațiilor și sistemelor de ecuații diferențiale ordinare – stabilitatea numerică poate fi demonstrată matematic și exprimată sub forma unor **condiții de stabilitate**.

Observații cu referire la stabilitatea numerică a algoritmilor destinați rezolvării numerice a ecuațiilor diferențiale ordinare:

1. În cazul metodelor de tip Euler condiția de stabilitate poate fi exprimată sub forma (4.1):

$$-2 < h \cdot \partial f / \partial y < 0, \quad (4.1)$$

de unde se poate concluziona că:

A. este necesar ca $\partial f / \partial y < 0$ pentru asigurarea stabilității numerice,

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

2. Și în cazul metodelor de tip Runge-Kutta se pot trage aceleași concluzii, rezultate din condiția de stabilitate (4.2):

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

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

Pentru asigurarea stabilității numerice se poate proceda la *reducerea valorii pasului de integrare h până la o valoare suficient de mică*. Însă, metodele de tip Runge-Kutta sunt mai puțin eficiente decât cele de tip predictor-corector datorită numărului mai mare de evaluări ale funcției la fiecare pas;

deci, o reducere prea substanțială a valorii lui h determină creșterea considerabilă a volumului de calcule legate de evaluările funcției.

→ cel puțin din motivul asigurării stabilității numerice este nevoie de ajustarea cu grijă a pasului de integrare pe parcursul procesului de soluționare a ecuației diferențiale.

3. În cazul metodelor de tip Adams-Bashforth-Moulton se pot trage concluziile A. și B. de la metodele de tip Euler. De data aceasta condiția de stabilitate are expresia (4.3):

$$-1.25 < h \cdot \partial f / \partial y < 0, \quad (4.3)$$

care este de asemenea utilă la estimarea mărimii pasului de integrare h .

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

Remember: erorile de calcul pot fi de trei tipuri – erori de trunchiere, erori de rotunjire și 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.

Erorile de trunchiere depind de metoda numerică aleasă. Pentru metodele numerice analizate aici, erorile de trunchiere sunt:

- ◆ de ordinul de mărime al lui h^2 pentru versiunea clasică a metodelor de tip Euler și de ordinul de mărime al lui h^3 pentru versiunea Cauchy a metodelor de tip Euler;
- ◆ de ordinul de mărime al lui h^5 pentru metodele de tip Runge-Kutta de ordinul 4;
- ◆ de ordinul de mărime al lui h^5 pentru metodele de tip Adams-Bashforth-Moulton de ordinul 4.

Erorile de rotunjire depind de posibilitățile echipamentului de calcul pe care sunt implementați algoritmi 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 creșterii numărului de pași de integrare datorită propagării erorilor de la un pas la altul.

Erorile de propagare: efectul lor este simțit 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.

Pentru reducerea erorilor de trunchiere este recomandată micșorarea pasului de integrare, însă aceasta conduce la creșterea erorii de rotunjire. Pe de altă parte, micșorarea pasului de integrare conduce la creșterea numărului de pași, cu consecința imediată a creșterii timpului de calcul. Deci, trebuie căutată – ca o *soluție de compromis* – acea valoare a pasului de integrare pentru care erorile de calcul (“suma” celor trei tipuri de erori menționate) să fie cât mai mici.

Ținând seama legătura dintre mărimea pasului de integrare și cea a erorilor de calcul, *alegerea unei anumite metode de rezolvare numerică a ecuațiilor diferențiale ordinare* reprezintă o problemă relativ complexă – **recomandări:**

- Dacă se cer *rezolvări rapide* → se poate alege 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.
- Dacă se cer *rezolvări precise fără a fi important timpul de calcul* → se poate alege metoda Runge-Kutta de ordinul 4 datorită avantajelor legate de autopornire.
- Dacă se cer *rezolvări precise fără modificări semnificative ale pasului de integrare* → se poate alege metoda Adams-Bashforth-Moulton de ordinul 4 datorită avantajelor legate de stabilitate numerică.
- La toate metodele numerice utilizate este *necesară analiza stabilității numerice* deoarece toate trebuie să fie numeric stabile. Dacă aceasta nu poate fi efectuată teoretic → trebuie testată în funcție de aplicație.

+ experiența utilizatorului !

+ analiza aplicației !