

Curs 9: METODE NUMERICE UTILIZATE ÎN SIMULAREA SISTEMELOR DINAMICE

1 Noțiunea de „sistem dinamic”. Clasificări. Noțiunea de „simulare”

Un *sistem dinamic* este o entitate care se caracterizează printr-un mod specific de transformare a unui semnal de intrare (*cauza*) într-un semnal de ieșire (*efectul*). Matematic, aceste semnale sunt funcții de timp (figura 1).

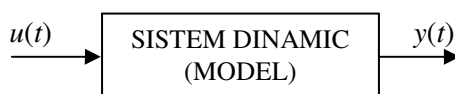


Fig. 1 Reprezentarea de tip „cutie neagră” a unui sistem dinamic

Dacă mulțimea momentelor de timp în care evoluează un sistem este o submulțime a:

- mulțimii numerelor reale, $\mathbb{R}$, sistemul este cu timp continuu (*sistem continuu*);
- mulțimii numerelor întregi, $\mathbb{Z}$, atunci sistemul se numește cu timp discret (*sistem discret* sau *sistem cu eșantionare*).

În modelarea sistemelor dinamice se utilizează legi specifice de conservare de masă și energie, descriind fenomene de acumulare și de dezacumulare, exprimate matematic sub forma ecuațiilor de bilanț, care sunt ecuații diferențiale. Așadar, se poate spune că prima imagine asupra comportamentului temporal al unui sistem dinamic este dată de *modelul de tip ecuație diferențială*. Ecuațiile diferențiale sunt fie *ordinare* – la care variabila de derivare este timpul, t – fie *cu derivate parțiale* – la care mai apare încă cel puțin o variabilă de derivare pe lângă timp (de exemplu, o coordonată spațială). Fiecare tip de ecuație diferențială este asociat respectiv unei clase de sisteme:

- *sistemele cu parametri concentrați* sunt descrise de ecuații diferențiale ordinare (exemplul 1);
- *sistemele cu parametri distribuiți* sunt descrise de ecuații diferențiale cu derivate parțiale (exemplul 2.).

Exemplul 1: Se consideră sistemul mecanic din figura 2 a), unde un corp de masă M , legat de un perete printr-un resort de constantă de elasticitate k , se deplasează cu frecare vâscoasă, caracterizată de coeficientul de frecare f_v , sub acțiunea unei forțe variabile în timp, $f(t)$.

Modelarea ca sistem dinamic începe cu identificarea drept cauză (intrare) a forței $f(t)$ și drept efect (ieșire) a deplasării $x(t)$ (figura 2 b)). Cele două mărimi sunt legate prin legea de conservare a energiei mecanice, descrisă de *ecuația diferențială ordinară* de ordinul al II-lea:

$$(1) \quad M \cdot \frac{d^2 x(t)}{dt^2} + f_v \cdot \frac{dx(t)}{dt} + k \cdot x(t) = f(t),$$

care este un model al sistemului.

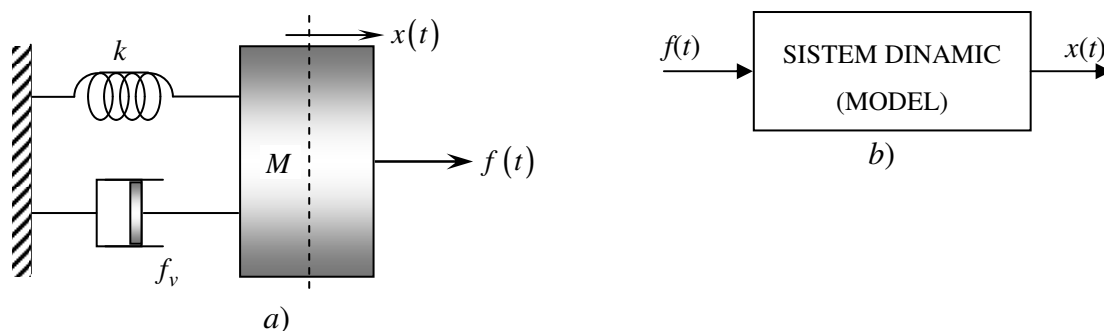


Fig. 2 Sistem mecanic cu resort: a) reprezentare fizică; b) simbolizare ca sistem dinamic

Sistemul mecanic descris este, deci, un *sistem dinamic cu parametri concentrați*.

Exemplul 2: Se consideră o bară de lungime L , a cărei temperatură se modifică drept urmare a aplicării la un moment dat, t_0 , la capetele ei a unor temperaturi cu variație cunoscută în timp, $\Theta_0(t)$ și $\Theta_L(t)$ (figura 3). Se cunosc constantele de material: densitatea materialului barei, ρ [kg/m³], căldura lui specifică, c [kcal·grd/kg] și conductibilitatea termică, λ [kcal·grd/m/h], precum și distribuția de temperaturi de-a lungul barei la momentul t_0 , $\Theta^0(x)$.

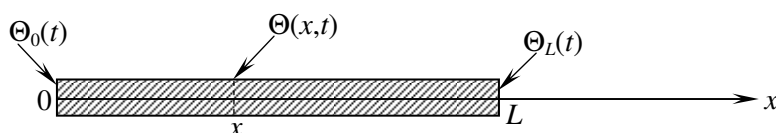


Fig. 3 Sistem guvernat de transferul unidirecțional de căldură

Temperatura în punctele barei este funcție și de timp, dar și de depărtarea de capetele barei, deci de coordonata spațială, notată cu x . Luând ca origine a coordonatei spațiale capătul din stînga al barei, se notează cu $\Theta(x,t)$ temperatura în punctul aflat la abscisa x la momentul de timp t (figura 4.3). $\Theta(x,t)$ este efectul datorat cauzelor $\Theta_0(t)$ și $\Theta_L(t)$ (numite și *forțaje de temperatură*). Evoluția temperaturii $\Theta(x,t)$ este descrisă de ecuația transferului unidirecțional de căldură, care este o *ecuație diferențială cu derivate parțiale*:

$$(2) \quad \frac{\partial \Theta(x,t)}{\partial t} = a \cdot \frac{\partial^2 \Theta(x,t)}{\partial^2 x},$$

unde $a = \lambda/(\rho \cdot c)$ se numește difuzabilitate termică [m²/s]. Sistemul descris este, deci, un *sistem dinamic cu parametri distribuiți*.

Dacă un sistem răspunde printr-o ieșire mărginită unui semnal de intrare mărginit, atunci se spune că sistemul este **stabil (intrare-ieșire)**. O clasă importantă de sisteme sunt cele **liniare**, cărora li se asociază ecuații diferențiale liniare (a se vedea exemplul 1). Sistemele liniare stabile sunt caracterizate prin aceea că lasă neschimbată forma semnalelor armonice; cu alte cuvinte, dacă la intrarea unui sistem liniar stabil se aplică semnal sinusoidal, atunci ieșirea lui va fi tot sinusoidală. Majoritatea sistemelor reale nu verifică această proprietate; se spune că sunt **neliniare**.

Termenul de “**simulare**” este asociat *evoluției temporale* a unui sistem dinamic descris printr-un model adecvat, atunci când sistemului i se aplică un semnal de intrare cunoscut; în particular, se pune problema determinării ieșirii sistemului, numite și **răspuns** (la o anume intrare cunoscută). De exemplu, simularea sistemului din exemplul (1) constă în a afla cum

variază deplasarea corpului în timp, $x(t)$, atunci când se cunoaște $f(t)$, forma de variație a forței aplicate la un moment dat, t_0 . Pentru aceasta este necesară integrarea numerică a ecuației diferențiale (1) în condiții inițiale cunoscute, $x(t_0)$ și $\dot{x}(t_0)$. Simularea sistemului din exemplul (2) presupune integrarea numerică a ecuației (2) în condiții inițiale cunoscute, $\Theta(x, t_0) = \Theta^0(x)$, pentru $x \in [0; L]$.

A *simula* un sistem dinamic este echivalent cu a *rezolva* (integra) ecuația diferențială care îl descrie, în condiții inițiale cunoscute, care reprezintă starea inițială a sistemului. Acest capitol este dedicat metodelor de **integrare numerică a ecuațiilor diferențiale**, ca primă metodă de simulare numerică a sistemelor dinamice.

O mare parte din problematica determinării răspunsului sistemelor se reduce la găsirea soluției ecuației diferențiale (sistemului de ecuații diferențiale) ca soluție (unică) a problemei cu timp inițial (problemei Cauchy).

Deoarece ecuațiile diferențiale care modelează procesele din lumea reală sunt adesea neliniare și de ordin superior, găsirea unei soluții analitice (exacte) a problemei Cauchy este dificilă. Alternativa o constituie găsirea unei *soluții aproximative*, și anume prin folosirea unui algoritm executabil pe o mașină de calcul. Astfel, soluția găsită prin aplicarea unei metode numerice este un șir de aproximații ale valorilor soluției exacte, calculate la momente discrete de timp, de obicei egal distanțate.

2 Simularea sistemelor cu parametri concentrați. Integrarea numerică a ecuațiilor diferențiale ordinare și a sistemelor de ecuații diferențiale ordinare

2.1 Fundament teoretic

În această secțiune sunt rezumate principalele rezultate teoretice care stau la baza metodelor numerice de rezolvare a ecuațiilor diferențiale ordinare (EDO) și a sistemelor de ecuații diferențiale ordinare.

În vederea rezolvării numerice a EDO sau a sistemelor de EDO, indiferent de metoda folosită, mai întâi se procedează la scrierea acestora sub forma standard:

$$(3) \quad \frac{dy}{dt} = f(t, y), \quad t \in [t_0; t_f], \quad y(t_0) = y_0,$$

unde $y = [y_1 \quad \dots \quad y_n]^T$ și condiția inițială, $y_0 = [y_{1_0} \quad \dots \quad y_{n_0}]^T$, sunt vectori de dimensiune n , unde n este ordinul ecuației sau al sistemului considerat, $f = [f_1 \quad \dots \quad f_n]^T$ este o funcție vectorială de aceeași dimensiune, iar $[t_0; t_f]$ este intervalul de integrare.

Scrierea (3) este posibilă oricare ar fi ordinul ecuației sau al sistemului considerat, în virtutea unui rezultat cunoscut din matematică: orice ecuație diferențială de ordinul $n > 1$, printr-o schimbare de variabilă, poate fi scrisă ca un sistem de n ecuații diferențiale de ordinul I. Această echivalare se face introducând n noi variabile, al căror mod de alegere nu este unic, după cum se arată mai jos.

Pentru o ecuație diferențială de ordin $n > 1$, dată sub forma:

$$(4) \quad \frac{d^n y}{dt^n} = f\left(t, y, \dot{y}, \ddot{y}, \dots, y^{(n-1)}\right), \quad t \in [t_0; t_f],$$

$$y(t_0) = y_{1_0}, \quad \dot{y}(t_0) = y_{2_0}, \dots, y^{(n-1)}(t_0) = y_{n_0}$$

se poate opta pentru alegerea noilor variabile astfel:

$$(5) \quad y^{(i-1)}(t) = z_i(t), \quad i = \overline{1, n},$$

obținându-se un sistem de ecuații diferențiale în variabila z :

$$(6) \quad \begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dots \\ \dot{z}_n = f(t, z_1, z_2, \dots, z_n) \end{cases},$$

cu condiția inițială $\underline{z}(t_0) = [z_{1_0} = y_{1_0} \quad \dots \quad z_{n_0} = y_{n_0}]^T$. Ca urmare a acestei schimbări de variabilă, sistemul (6) este de tipul (1):

$$(7) \quad \frac{d\underline{z}}{dt} = f_z(t, \underline{z}), \quad \underline{z}(t_0) = \underline{z}_0,$$

unde:

$$(8) \quad f_z = [z_2 \quad z_3 \quad \dots \quad f(t, z_1, z_2, \dots, z_n)]^T$$

Observație:

Schimbarea de variabilă (5) reprezintă numai una dintre posibilitățile de a transforma o ecuație diferențială de ordin superior într-un sistem de ecuații diferențiale de ordinul I, obținându-se astfel forma standard (1), care face obiectul metodelor numerice de simulare a răspunsului sistemelor. Se pot utiliza și alte schimbări de variabilă care au același efect, evident cu expresii diferite pentru funcția vectorială f_z .

Teorema 1 (unicitatea soluției problemei (3)):

Fie mulțimea convexă $D = \{(t, y) | t_0 \leq t \leq t_f, y \in \mathbb{R}^n\}$ și funcția vectorială $f(t, y)$ continuă pe D . Dacă există o constantă $L > 0$ astfel încât $\|f(t, y_1) - f(t, y_2)\| \leq L \cdot \|y_1 - y_2\|$ – adică f este lipschitziană (notația $\|\cdot\|$ desemnează norma vectorială) – atunci soluția problemei (3), $y(t)$, este unică pentru $t \in [t_0; t_f]$.

Se pune acum problema consecințelor pe care le au asupra soluției ecuației (4.3) eventualele perturbații – datorate erorilor de rotunjire – apărute în condițiile inițiale sau/și în funcția f . Se consideră **problema perturbată**:

$$(9) \quad \frac{d\underline{z}}{dt} = f(t, \underline{z}) + \delta(t), \quad t \in [t_0; t_f], \quad \underline{z}(t_0) = \underline{y}(t_0) + \varepsilon_0,$$

care admite soluția (exactă) $\underline{z}(t)$. Se demonstrează următorul rezultat.

Teorema 2 (soluția problemei perturbate):

Dacă f este continuă și lipschitziană pe $[t_0; t_f]$, atunci pentru orice $\varepsilon > 0$, cu $\varepsilon > \max\{\|\varepsilon_0\|, \delta(t)\}$, există $k(\varepsilon) > 0$ pentru care este îndeplinită relația:

$$(10) \quad \|\underline{z}(t) - \underline{y}(t)\| < k(\varepsilon) \cdot \varepsilon, \quad \forall t \in [t_0; t_f],$$

unde $\underline{y}(t)$ este soluția (exactă) a problemei neperturbate. Se spune că problema (4.3) este **bine formulată**.

Metodele numerice de integrare a ecuațiilor diferențiale furnizează soluții ale problemelor perturbate, aceasta datorită erorilor de rotunjire.

2.2 Elemente necesare pentru integrarea numerică a EDO

Aceste elemente trebuie văzute drept *date de intrare* care sunt furnizate unei metode numerice de integrare. Ele sunt listate mai jos:

1) EDO de rezolvat trebuie mai întâi adusă la forma (3) printr-o *schimbare de variabilă* adecvată (în particular se poate utiliza schimbarea de variabilă (5)); rezultă astfel funcția vectorială f_z , de dimensiune egală cu ordinul ecuației, n , care descrie legătura dintre

derivatele celor n noi variabile, $\dot{z}_i$, și valorile lor nederivate, z_i ;

2) *intervalul de integrare* dat prin capetele acestuia, t_0 și t_f ;

3) *condițiile inițiale*, sub forma unui vector de dimensiune n , $\underline{y}_0 = [y_{10} \quad \dots \quad y_{n0}]^T$;

4) *pasul de integrare*, notat cu h – în toate metodele de integrare se consideră o diviziune de ordinul m a intervalului $[t_0; t_f]$ (o împărțire a acestui interval în intervale de aceeași lungime, h), în ale cărei puncte, $t_k = t_0 + k \cdot h$, $k \in \{0, \dots, n\}$, se calculează valorile soluției numerice a ecuației, notate cu y_k , care sunt aproximări ale valorilor exacte, $y(t_k)$; este intuitiv evident că soluția numerică se apropie cu atât mai mult de soluția exactă cu cât diviziunea este mai fină (pasul de integrare, h , este mai mic);

5) un *subprogram* (funcție) care calculează valoarea funcției f_z pentru orice moment de timp, t , și care va fi apelat la fiecare pas de integrare, t_k , de programul unde este implementată metoda de integrare numerică;

6) alte date care se constituie ca *parametri constanți* în funcția f_z .

2.3 Metode unipas (directe)

Metodele unipas de integrare numerică a EDO sunt **metodele Taylor**, care nu se folosesc, fiind complicate, și **metodele Runge-Kutta (RK)**. Aceste metode au ca bază de pornire expresia dezvoltării polinomiale în serie Taylor a funcției $y(t)$, presupusă derivabilă de ordinul $(p+1)$ și cu derivatele continue:

$$(11) \quad y(t_{k+1}) = y(t_k) + h \dot{y}(t_k) + \frac{h^2}{2!} \ddot{y}(t_k) + \dots + \frac{h^p}{p!} y^{(p)}(t_k) + \frac{h^{p+1}}{(p+1)!} y^{(p+1)}(\xi_k),$$

unde $\xi_k \in (t_k, t_{k+1})$. Înlocuind în (4.11) expresia derivatei de ordinul i a funcției $y(t)$:

$$(12) \quad y^{(i)}(t) = f^{(i-1)}(t, y(t)), \quad i = \overline{1, (p+1)}$$

în punctul t_k , se obține:

$$(13) \quad y(t_{k+1}) = y(t_k) + hf(t_k, y(t_k)) + \frac{h^2}{2!} \dot{f}(t_k, y(t_k)) + \dots + \frac{h^{p+1}}{(p+1)!} f^{(p)}(\xi_k, y(\xi_k))$$

Relația (13) este o ecuație în diferențe ce permite calculul soluției în punctul t_{k+1} pe baza soluției în punctul t_k . Transpunerea numerică a acestei relații conduce la calculul soluției la un anumit pas de integrare, y_{k+1} , în funcție de soluția la pasul anterior, y_k . Această dependență, care ia în considerare „istoria” integrării cu un singur pas în urmă justifică numele acestei clase de metode (unipass).

În general, scopul final al oricărei metode numerice este să determine cu suficientă acuratețe o soluție aproximativă a problemei cu minimum de resurse de memorie și număr de operații. Cea mai importantă măsură a acurateții o constituie **eroarea locală de trunchiere**. Aceasta reprezintă la un anumit pas măsura în care soluția exactă a ecuației diferențiale satisface ecuația în diferențe utilizată pentru calcul.

Se demonstrează că la metodele Taylor, care sunt bazate pe relația (13), eroarea locală de trunchiere este mărginită superior de un polinom de grad p în pasul de integrare, ceea ce înseamnă că este de ordinul h^p ($O(h^p)$). Aceasta arată că, într-adevăr, acuratețea metodei crește odată cu micșorarea lui h . Pe de altă parte, micșorarea lui h antrenează creșterea numărului de operații efectuate și, deci, a erorilor de rotunjire. Se demonstrează că există o valoare a lui h pentru care se atinge minimul erorii de trunchiere în prezența erorilor de rotunjire. Concluzia este că pasul de integrare, h , rezultă dintr-un *compromis* între *acuratețea* și *precizia* metodei.

Pentru $p=0$ se obține **metoda Taylor de ordinul 1**, cunoscută și ca **metoda Euler** sau **metoda Runge-Kutta de ordinul 1**:

$$(14) \quad y_{k+1} = y_k + h \cdot f(t_k, y_k),$$

la care minimizarea erorii de trunchiere se obține pentru:

$$h = \sqrt{2\delta/M},$$

unde δ este eroarea de rotunjire la fiecare pas de integrare, iar M este o constantă care mărginește superior valorile derivatei a doua ale soluției exacte a ecuației pe intervalul de

$$\text{integrare: } \left| \ddot{y}(t) \right| \leq M, \quad \forall t \in [t_0; t_f].$$

Metodele Taylor de ordinul $p > 1$ nu sunt folosite ca atare, deoarece necesită calculul expresiei analitice a derivatelor funcției din membrul drept din (4.3). Metodele RK elimină acest dezavantaj, păstrând avantajul erorii locale de trunchiere mici.

Metoda Runge-Kutta de ordinul 2 – sau *metoda punctului median* – care are o eroare de trunchiere de $O(h^2)$, are la bază expresia:

$$(15) \quad y_{k+1} = y_k + h \cdot f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2} \cdot f(t_k, y_k)\right)$$

Alte două metode cunoscute sub numele general de metode Runge-Kutta de ordinul 2 sunt **metoda Euler modificată**, care are relația în diferențe:

$$(16) \quad y_{k+1} = y_k + \frac{h}{2} \cdot \left[f(t_k, y_k) + f\left(t_{k+1}, y_k + h \cdot f(t_k, y_k)\right) \right]$$

și **metoda Heun**, cu relația în diferențe:

$$(17) \quad y_{k+1} = y_k + \frac{h}{4} \cdot \left[f(t_k, y_k) + 3f\left(t_{k+1} + \frac{2}{3}h, y_k + \frac{2}{3}h \cdot f(t_k, y_k)\right) \right]$$

Metoda RK de ordinul 3 este puțin utilizată.

Una dintre cele mai utilizate metode unipas este **metoda Runge-Kutta de ordinul 4**, căreia îi corespund următoarele formule de calcul:

$$(18) \quad y_{k+1} = y_k + \frac{1}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4),$$

unde:

$$(19) \quad \begin{cases} k_1 = h \cdot f(t_k, y_k) & k_2 = h \cdot f\left(t_k + \frac{h}{2}, y_k + \frac{1}{2} \cdot k_1\right) \\ k_3 = h \cdot f\left(t_k + \frac{h}{2}, y_k + \frac{1}{2} \cdot k_2\right) & k_4 = h \cdot f(t_k + h, y_k + k_3) \end{cases}$$

Această metodă are eroarea locală de trunchiere $O(h^4)$, iar funcția $y(t)$ trebuie să fie de clasă C^5 (continuă și cu derivatele până la ordinul 5 continue).

Exemplu: Sa se scrie un program Matlab care sa realizeze integrarea numerica a sistemului de ecuatii diferentiale:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{1}{2}(25 - 5x_2 - 25x_1)$$

Cerinte: - Se va utiliza metoda de integrare numerica Runge-Kutta de ordinal 4.

- Se vor afisa grafic, in aceeasi figura, rezultatele obtinute prin integrare

Rezolvare:

```
% program principal pentru integrare numerica
clear all; close all;
x=[7 10];
h=0.1;
t0=0;
tf=10;
t(1)=0;
i=1;
for m=0.1:h:tf,
    k1=func(t,x(i,:));
    k2=func(t+h/2,x(i,:)+h*k1/2);
    k3=func(t+h/2,x(i,:)+h*k2/2);
    k4=func(t+h,x(i,:)+h*k3);
    x(i+1,:)=x(i,:)+h/6*(k1+2*k2+2*k3+k4);
    t(i+1)=m;
```

```

        i=i+1;
    end
    subplot(211);
    hold on;
    plot(t,x(:,1),'k');
    subplot(212);
    hold on;
    plot(t,x(:,2),'k');

% implementare sistem de ecuatii diferentiale
function xd=func(t,x)
xd(1)=x(2);
xd(2)=1/2*(25-5*x(2)-25*x(1));

```

2.4 Metode directe cu pas variabil

În general nu se poate determina eroarea globală a unei metode directe de integrare, dar se știe că există o strânsă legătură între aceasta și eroarea locală de trunchiere. Prin utilizarea adecvată a metodelor directe de diferite ordine se poate prezice eroarea locală de trunchiere și, împreună cu variația pasului, se poate controla eroarea globală.

Metodele directe cu pas variabil se mai numesc și **cu pas adaptiv** pentru că adaptează numărul și poziția punctelor în care se face calculul astfel încât eroarea locală de trunchiere să se mențină inferioară unei limite impuse. Acest paragraf este dedicat manierei în care eroarea locală de trunchiere poate fi controlată prin variația pasului de integrare.

Eroarea locală de trunchiere produsă prin aplicarea unei metode directe de ordin p poate fi controlată prin folosirea unui pas de integrare modificat. Presupunând că această modificare este de forma $h_{nou} = \delta \cdot h_{vechi}$, se demonstrează că este suficient să se aleagă:

$$(20) \quad \delta \leq \left(\frac{\varepsilon}{2|\tilde{y}_{k+1} - y_{k+1}|} \right)^{\frac{1}{p}},$$

unde cu ε s-a notat limita impusă a erorii locale de trunchiere (toleranța impusă), iar $\tilde{y}_{k+1}$ reprezintă soluția la pasul $k+1$ dată de o altă metodă de integrare decât prima folosită.

O metodă cunoscută care utilizează această tehnică, de control al erorii globale prin variația pasului de integrare, este **metoda Runge-Kutta-Fehlberg**. Această metodă constă în utilizarea unei metode Runge-Kutta cu eroarea locală de trunchiere de ordinul cinci:

$$(21) \quad \tilde{y}_{k+1} = y_k + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6,$$

pentru a estima eroarea locală de trunchiere a unei metode RK de ordinul 4 dată prin relația:

$$(22) \quad y_{k+1} = y_k + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5,$$

unde:

$$(23) \quad \begin{cases} k_1 = h \cdot f(t_k, y_k) & k_2 = h \cdot f\left(t_k + \frac{h}{4}, y_k + \frac{1}{4}k_1\right) \\ k_3 = h \cdot f\left(t_k + \frac{3h}{8}, y_k + \frac{3}{32}k_1 + \frac{9}{32}k_2\right) \\ k_4 = h \cdot f\left(t_k + \frac{12h}{13}, y_k + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right) \\ k_5 = h \cdot f\left(t_k + h, y_k + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right) \\ k_6 = h \cdot f\left(t_k + \frac{h}{2}, y_k - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right) \end{cases}$$

Un avantaj al acestei metode este că sunt necesare numai șase evaluări ale funcției f . Metodele RK de ordinul 4 și 5, utilizate împreună necesită cel puțin patru evaluări pentru cea de ordinul 4 și șase pentru cea de ordinul 5, deci în total cel puțin zece evaluări. O valoare inițială pentru h la pasul k este utilizată pentru a găsi primele valori ale lui y_{k+1} și $\tilde{y}_{k+1}$ care, la rândul lor, sunt utilizate în calculul lui δ . În mod practic, pentru alegerea unei valori inițiale pentru h se folosește o regulă mai mult sau mai puțin empirică. Pentru metoda Runge-Kutta-Fehlberg de ordinul 4 această valoare se poate obține particularizând $p=4$ în relația (4.20):

$$\delta \leq \left(\frac{\varepsilon}{2|\tilde{y}_{k+1} - y_{k+1}|} \right)^{\frac{1}{4}} = 0.84 \left(\frac{\varepsilon}{|\tilde{y}_{k+1} - y_{k+1}|} \right)^{\frac{1}{4}}$$

Algoritmul Runge-Kutta-Fehlberg (de calcul al soluției ecuației diferențiale (3) cu eroare locală de trunchiere având o toleranță dată, ε)

Date de intrare: $t_0, t_f, y_0, h_{min}, h_{max}, \varepsilon$

Pas 1. $t = t_0, y = y_0, h = h_{max}, flag = 1$

Pas 2. Cât timp $flag = 1$ repetă pașii 3-11.

Pas 3. $k_1 \leftarrow h \cdot f(t_k, y_k)$

$k_2 \leftarrow h \cdot f(t_k + h/4, y_k + k_1/4)$

$k_3 \leftarrow h \cdot f(t_k + 3h/8, y_k + 3k_1/32 + 9k_2/32)$

$k_4 \leftarrow h \cdot f(t_k + 12h/13, y_k + 1932k_1/2197 - 7200k_2/2197 + 7296k_3/2197)$

$k_5 \leftarrow h \cdot f(t_k + h, y_k + 439k_1/216 - 8k_2 + 3680k_3/513 - 845k_4/4104)$

$k_6 \leftarrow h \cdot f(t_k + h/2, y_k - 8k_1/27 + 2k_2 - 3544k_3/2565 + 1859k_4/4104 - 11k_5/40)$

Pas 4. $R \leftarrow |k_1/30 - 128k_3/4275 - 2197k_4/75240 + k_5/50 - 11k_6/40|/h$

Pas 5. Dacă $R \leq \varepsilon$, atunci $t = t + h$

$y = y + 25k_1/216 + 1408k_3/2565 + 2197k_4/4104 - k_5/5$

Pas 7. Scrie t, y .

Pas 8. $\delta \leftarrow 0.84 \cdot \left(\frac{\varepsilon}{R} \right)^{\frac{1}{4}}$

Pas 9. Dacă $\delta \leq 0.1$, atunci $h = 0.1 \cdot h$

altfel, dacă $\delta \geq 4$, atunci $h = 4 \cdot h$

altfel $h = \delta \cdot h$

Pas 10. Dacă $h > h_{max}$, atunci $h = h_{max}$

Pas 11. Dacă $t \geq t_f$,

atunci $flag = 0$

altfel, dacă $t + h > t_f$, atunci $h = t_f - t$

altfel, dacă $h > h_{min}$, atunci $flag = 0$

altfel scrie „Pas minim depășit!”

2.5 Metode multipas (indirecte)

2.5.1 Introducere

Metodele unipas, discutate în capitolele anterioare, folosesc numai informația de la momentul curent, t_k , pentru a calcula soluția EDO la momentul următor, t_{k+1} . Deși aceste metode folosesc și valori calculate în interiorul intervalului de integrare $[t_k, t_{k+1}]$, totuși acestea nu sunt reținute pentru utilizare ulterioară și, prin urmare, nu sunt disponibile utilizatorului. Deoarece informația calculată la momentele $t_0, t_1, \dots, t_k$ este disponibilă și eroarea locală de trunchiere, $|y_k - y(t_k)|$, crește odată cu k , atunci apare motivată dezvoltarea de metode care să folosească informația de la momentele de integrare anterioare. Aceste metode care, pentru a calcula soluția y_{k+1} , folosesc „istoria” integrării cu cel puțin doi pași în urmă, se numesc *metode multipas* sau *indirecte*.

Definiția 4.1: O metodă multipas de ordinul m folosită pentru a determina soluția aproximativă a problemei Cauchy (3) presupune calculul lui y_{k+1} cu relația:

$$(24) \quad y_{k+1} = a_{m-1}y_k + a_{m-2}y_{k-1} + \dots + a_0y_{k+1-m} + h \cdot (b_m f(t_{k+1}, y_{k+1}) + b_{m-1}f(t_k, y_k) + \dots + b_0 f(t_{k+1-m}, y_{k+1-m})),$$

unde a_i, b_i , $i = 0, \dots, m-1$ sunt constante.

Când $b_m = 0$, metoda se numește *explicită* sau *deschisă*, în caz contrar ea se numește *implicită* sau *închisă*.

Condiția inițială, y_0 , este cunoscută. Folosirea relației (24) într-o structură de ciclare necesită ca următoarele $m-1$ valori care urmează condiției inițiale, $y_1, y_2, \dots, y_{m-1}$, să fie calculate separat, înainte de a se intra în ciclare. De obicei, aceste valori se calculează printr-o metodă directă, Runge-Kuta. Această operațiune poartă denumirea de **amorsare**. După amorsare, valorile $y_k, y_{k-1}, \dots, y_{k+1-m}$ se calculează în ciclu utilizând formula (24) particularizată pentru metoda multipas respectivă (această fază se numește **integrarea propriu-zisă**).

Metodele implicite sunt folosite în practică mai ales pentru a îmbunătăți valoarea calculată la un pas de integrare printr-o metodă explicită. Combinația dintre metodele explicite și cele implicite dă naștere la așa numitele **metode predictor-corector** sau **cu predicție și corecție**. La aceste metode faza de integrare propriu-zisă se împarte la fiecare pas în două etape: *calculul (predicția)* valorii lui y_{k+1} prin *formula explicită* și, respectiv, *corecția* acestei valori cu *formula implicită*, până la o anumită precizie impusă. Metodele cu predicție și corecție sunt singurele metode care permit controlul direct al preciziei de calcul.

În cele ce urmează, clasa metodelor predictor-corrector este tratată separat de restul metodelor multipas, pe care le vom numi *simple*.

2.5.2 Metode multipas simple

Pașii majori ai integrării numerice prin metode multipas simple sunt dați mai jos (notația $[\cdot]$ desemnează partea întreagă).

Algoritmul general de integrare numerică a EDO prin metode multipas simple

Date de intrare: t_0, t_f, y_0, h

Pas 1. $t = t_0, y = y_0$

Pas 2. **Amorsarea:** calculul valorilor $y_1, y_2, \dots, y_{m-1}$ (printr-o metodă directă, Runge-Kutta).

Pas 3. Pentru k de la $m-1$ la $\left\lfloor \frac{t_f - t_0}{h} \right\rfloor - 1$

$$y_{k+1} = AB_m(y_k, y_{k-1}, \dots, y_{k+1-m})$$

repetă

Expresiile de calcul pentru formulele de integrare multipas se deduc pornind de la integrala funcției $\dot{y}(t)$ pe intervalul $[t_k, t_{k+1}]$:

$$(25) \quad y(t_{k+1}) - y(t_k) = \int_{t_k}^{t_{k+1}} \dot{y}(t) dt = \int_{t_k}^{t_{k+1}} f(t, y(t)) dt$$

Deoarece nu se poate integra $f(t, y(t))$ fără cunoașterea lui $y(t)$, se va utiliza un polinom de interpolare, $P(t)$, care trece prin punctele $(t_k, y_k), (t_{k-1}, y_{k-1}), \dots, (t_{k+1-m}, y_{k+1-m})$. Dacă se presupune și $y(t_k) \approx y_k$, atunci relația (25) devine:

$$(26) \quad y(t_{k+1}) \approx y_k + \int_{t_k}^{t_{k+1}} P(t) dt$$

În principiu, orice metodă de interpolare este utilizabilă pentru determinarea lui $P(t)$, dar cea mai des folosită este *interpolarea Newton de tip diferență înapoi*.

Eroarea locală de trunchiere corespunzătoare metodelor multipas se calculează similar metodelor unipas.

Fără a detalia calculele, în tabelul 1 se prezintă formulele de integrare și erorile de trunchiere pentru *metoda multipas explicită* de ordinul m , numită și **Adams-Bashforth** (AB), în cazurile $m=2, 3, 4$.

m	Formula de integrare	Eroarea locală de trunchiere la pasul $k+1$
2	(27) $y_{k+1} = y_k + \frac{h}{2} \cdot (3f(t_k, y_k) - f(t_{k-1}, y_{k-1}))$	$\tau_{k+1}(h) = \frac{5h^2}{12} \cdot y^{(3)}(\mu_k), \forall \mu_k \in (t_{k-2}, t_{k-1})$
3	(28) $y_{k+1} = y_k + \frac{h}{12} \cdot (23f(t_k, y_k) - 16f(t_{k-1}, y_{k-1}) + 5f(t_{k-2}, y_{k-2}))$	$\tau_{k+1}(h) = \frac{3h^3}{8} \cdot y^{(4)}(\mu_k), \forall \mu_k \in (t_{k-2}, t_{k-1})$

4	(29) $y_{k+1} = y_k + \frac{h}{24} \cdot (55f(t_k, y_k) - 59f(t_{k-1}, y_{k-1}) + 37f(t_{k-2}, y_{k-2}) - 9f(t_{k-3}, y_{k-3}))$	$\tau_{k+1}(h) = \frac{251h^4}{720} \cdot y^{(5)}(\mu_k), \forall \mu_k \in (t_{k-2}, t_{k-1})$
---	--	---

Tabel 1 Caracteristicile metodei multipas Adams-Bashforth pentru diferite ordine

Metodele multipas implicite, numite și **Adams-Moulton** (AM), utilizează perechea $(t_{k+1}, f(t_{k+1}, y_{k+1}))$ ca punct adițional al polinomului de interpolare în aproximarea integralei $\int_{t_k}^{t_{k+1}} f(t, y(t)) dt$.

În tabelul 2 se prezintă formulele de integrare și erorile de trunchiere pentru metodele AM de diferite ordine.

m	Formula de integrare	Eroarea locală de trunchiere la pasul $k+1$
0	(30) $y_{k+1} = y_k + \frac{h}{2} \cdot (f(t_{k+1}, y_{k+1}) + f(t_k, y_k))$	$\tau_{k+1}(h) = \frac{h^3}{12} \cdot y^{(3)}(\mu_k), \forall \mu_k \in (t_k, t_{k+1})$
1	(31) $y_{k+1} = y_k + \frac{h}{12} \cdot (5f(t_{k+1}, y_{k+1}) + 8f(t_k, y_k) - f(t_{k-1}, y_{k-1}))$	$\tau_{k+1}(h) = -\frac{h^3}{24} \cdot y^{(4)}(\mu_k), \forall \mu_k \in (t_{k-1}, t_{k+1})$
2	(32) $y_{k+1} = y_k + \frac{h}{24} \cdot (9f(t_{k+1}, y_{k+1}) + 19f(t_k, y_k) - 5f(t_{k-1}, y_{k-1}) + f(t_{k-2}, y_{k-2}))$	$\tau_{k+1}(h) = -\frac{19h^4}{720} \cdot y^{(5)}(\mu_k), \forall \mu_k \in (t_{k-2}, t_{k+1})$

Tabel 2 Caracteristicile metodei multipas Adams-Moulton pentru diferite ordine

2.5.3 Metode multipas corective (cu predicție și corecție)

Pentru a schița algoritmul de integrare numerică folosind metode corective, se introduce notația y_k^j , care semnifică soluția la pasul k de predicție și pasul j de corecție. Rezultă că valoarea predictată la fiecare pas poate fi considerată drept o corecție la pasul 0 și se notează, deci, cu y_k^0 . y_k desemnează valoarea finală (după predicție și corecție) a soluției la pasul k .

Presupunem că predicția se realizează cu o metodă AB de ordinul m , iar corecția cu o metodă AM de ordinul p (metoda corectivă rezultată se mai numește și **Adams-Bashforth-Moulton**). Precizia de corecție se notează cu ε , iar notația $\|\cdot\|$ desemnează norma vectorială.

Algoritmul general de integrare numerică a EDO prin metode PC

Date de intrare: $t_0, t_f, y_0, h, \varepsilon$

Pas 1. $t = t_0, y = y_0$

Pas 2. **Amorsarea**: calculul valorilor $y_1, y_2, \dots, y_{m-1}$ (printr-o metodă directă, Runge-Kutta).

Pas 3. Pentru k de la $m-1$ la $\left\lceil \frac{t_f - t_0}{h} \right\rceil - 1$

Predicția la pasul de integrare $k+1$:

$$y_{k+1}^0 = AB_m(y_k, y_{k-1}, \dots, y_{k+1-m})$$

Corecția la pasul 1 al pasului de integrare $k+1$:

$$y_{k+1}^1 = \text{AM}_p \left(y_{k+1}^0, y_k, y_{k-1}, \dots, y_{k+1-p} \right)$$

$j \leftarrow 1$

Cât timp $\left\| y_{k+1}^j - y_{k+1}^{j-1} \right\| > \varepsilon$

Corecția la pasul j al pasului de integrare $k+1$:

$$y_{k+1}^{j+1} = \text{AM}_p \left(y_{k+1}^j, y_k, y_{k-1}, \dots, y_{k+1-p} \right)$$

$j \leftarrow j+1$

repetă

$y_{k+1} \leftarrow y_{k+1}^{j+1}$

repetă

Acuratețea metodelor multipas corective (PC) depinde de efectul pe care îl are asupra erorilor cuplarea unei anumite formule de integrare explicite în faza de predicție cu o formulă implicită în faza de corecție. Analiza numerică a acestui efect a condus la perechile predictor-corector optime sintetizate în tabelul 3.

	Formulele de integrare	Comentarii
PC1	(33) $\begin{cases} (P): y_{k+1}^0 = y_k + \frac{h}{2} \cdot (3f(t_k, y_k) - f(t_{k-1}, y_{k-1})) \\ (C): y_{k+1}^{j+1} = y_k + \frac{h}{2} \cdot (f(t_{k+1}, y_{k+1}^j) + f(t_k, y_k)) \end{cases}$	– predicția cu AB pentru $m=2$ și corecția cu AM pentru $m=0$
PC2	(34) $\begin{cases} (P): y_{k+1}^0 = y_k + \frac{h}{12} \cdot (23f(t_k, y_k) - 16f(t_{k-1}, y_{k-1}) + 5f(t_{k-2}, y_{k-2})) \\ (C): y_{k+1}^{j+1} = y_k + \frac{h}{12} \cdot (5f(t_{k+1}, y_{k+1}^j) + 8f(t_k, y_k) - f(t_{k-1}, y_{k-1})) \end{cases}$	– predicția cu AB pentru $m=3$ și corecția cu AM pentru $m=1$
PC3	(35) $\begin{cases} (P): y_{k+1}^0 = y_k + \frac{h}{24} \cdot (55f(t_k, y_k) - 59f(t_{k-1}, y_{k-1}) + 37f(t_{k-2}, y_{k-2}) - 9f(t_{k-3}, y_{k-3})) \\ (C): y_{k+1}^{j+1} = y_k + \frac{h}{24} \cdot (9f(t_{k+1}, y_{k+1}^j) + 19f(t_k, y_k) - 5f(t_{k-1}, y_{k-1}) + f(t_{k-2}, y_{k-2})) \end{cases}$	– predicția cu AB pentru $m=4$ și corecția cu AM pentru $m=2$

Tabel 3 Formulele de integrare pentru cele mai utilizate metode predictor-corector