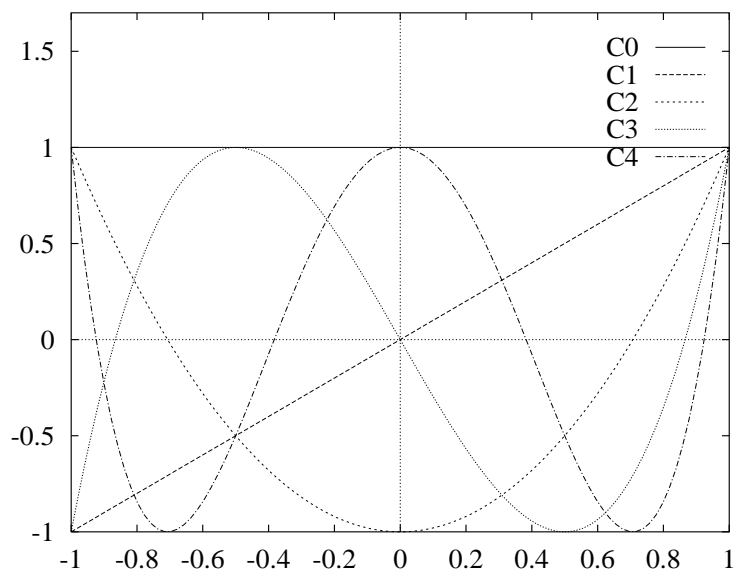


Daniel Ioan, Irina Munteanu



Metode Numerice

Algoritmi fundamentali și aplicații în
ingineria electrică

Editura
București, 2000

Daniel Ioan, Irina Munteanu
Catedra de Electrotehnică, Universitatea "Politehnica" din București

Metode Numerice

Algoritmi fundamentali și aplicații în
ingineria electrică

Editura
București, 2000

Daniel Ioan, Irina Munteanu

Metode Numerice. Algoritmi fundamentali și aplicații în ingineria electrică

Referenți științifici: xxx
yyy

Editura, București, 2000

Adresa editura

Cuprins

0	Introducere	1
0.1	Obiectul cursului	1
0.2	Locul metodelor numerice în ingineria electrică	2
1	Algoritmi și structuri de date	5
1.1	Structura sistemelor de calcul	5
1.2	Studiul algoritmilor	7
1.3	Generarea algoritmilor	9
1.3.1	Reprezentarea algoritmilor	11
1.3.2	Validarea și depanarea algoritmilor	20
1.3.3	Analiza și evaluarea algoritmilor	21
1.3.4	Implementarea algoritmilor	26
1.4	Reprezentarea datelor în calculatoarele numerice	26
1.4.1	Tipuri fundamentale de date	27
1.4.2	Agregarea datelor. Tipuri abstracte de date	31
1.4.3	Structuri dinamice de date	34
2	Erori de calcul și instabilități numerice	41
2.1	Caracterizarea erorilor de calcul	41
2.2	Erorile de rotunjire	43
2.3	Erori de trunchiere	45
2.4	Propagarea erorilor de calcul	52

3	Rezolvarea sistemelor de ecuații algebrice liniare	61
3.1	Formularea problemei. Condiționarea numerică	61
3.2	Metoda de eliminare Gauss	64
3.2.1	Descrierea algoritmului	64
3.2.2	Obținerea soluției prin substituție regresivă	66
3.2.3	Strategia de pivotare	67
3.2.4	Rezolvarea sistemelor simultane. Inversarea matricelor	69
3.2.5	Aplicație la analiza unui circuit rezistiv liniar prin metoda nodală	73
3.3	Factorizarea LU	78
3.3.1	Descrierea algoritmului de factorizare Crout	79
3.3.2	Algoritmul de factorizare Choleski	81
3.3.3	Factorizarea LU prin metoda Gauss	82
3.4	Sisteme liniare cu matrice rare	85
3.4.1	Sisteme tridiagonale și cu matrice bandă	85
3.4.2	Schema de memorare a matricelor rare	87
3.4.3	Umplerea matricelor rare	88
3.4.4	Pivotarea în cazul matricelor rare	94
3.5	Rezolvarea iterativă a sistemelor liniare	98
3.5.1	Metoda deplasărilor simultane (Jacobi)	102
3.5.2	Metoda deplasărilor succesive (Gauss - Seidel)	103
3.5.3	Convergența metodelor Jacobi și Gauss-Seidel	105
3.5.4	Metoda suprarelaxării succesive (Fränkel - Young)	110
3.5.5	Metoda iterației bloc	116
3.5.6	Alte metode iterative	118
3.6	Rezolvarea sistemelor liniare prin metode semiiterative	119
3.6.1	Metode semiiterative de tip Cebîșev	119
3.6.2	Metoda gradientilor conjugați	126
3.6.3	Precondiționarea gradientilor conjugați	130

4	Interpolarea și aproximarea funcțiilor	135
4.1	Formularea problemei	135
4.2	Interpolarea polinomială. Metoda Lagrange	139
4.3	Metoda Newton de interpolare polinomială. Diferențe divizate.	148
4.4	Interpolarea polinomială cu pas constant. Diferențe finite.	151
4.5	Alegerea nodurilor de interpolare. Metoda Cebîșev.	156
4.6	Interpolarea polinomială pe porțiuni (“spline”)	157
4.6.1	Interpolarea liniară pe porțiuni	157
4.6.2	Interpolarea pe porțiuni cu polinoame de grad superior	160
4.6.3	Analiza erorilor	167
5	Derivarea numerică a funcțiilor reale	171
5.1	Derivarea funcțiilor tabelare	172
5.1.1	Derivarea numerică folosind interpolarea liniară pe porțiuni	173
5.1.2	Derivarea numerică folosind interpolarea pe porțiuni a polinoamelor de grad superior	176
5.1.3	Derivate de ordin superior	181
5.1.4	Calculul derivatelor parțiale ale funcțiilor de mai multe variabile	182
5.2	Derivarea funcțiilor cunoscute prin cod	184
5.3	Rezolvarea numerică a ecuației Laplace	190
6	Integrarea numerică a funcțiilor reale	199
6.1	Formularea problemei	199
6.2	Cuadratură bazată pe interpolare polinomială. Formulele Newton – Cotes.	208
6.3	Metode de cuadratură de tip Gauss	217
6.4	Cuadratură compusă bazată pe interpolarea polinomială pe porțiuni	225
6.5	Metode de integrare numerică prin extrapolare. Metoda Romberg.	231
6.6	Integrale improprii	238
6.7	Integrale pe domenii nemărginite	242
6.8	Integrale multiple	247
6.8.1	Produsul cartezian al regulilor de cuadratură	250

6.8.2	Reguli de cuadratură exacte pentru monoame	254
6.8.3	Reguli compuse pentru integrale multiple	257
6.8.4	Metoda Monte Carlo de integrare numerică	258
6.9	Integrare automată	262
6.10	Transformări integrale. Serii Fourier	268
6.11	Analiza în frecvență a sistemelor liniare. Transformatele Fourier și Laplace.	276
6.12	Transformata Fourier discretă. Metoda transformării Fourier rapide.	284
7	Sisteme de ecuații algebrice neliniare	289
7.1	Introducere	289
7.2	Existența și unicitatea soluțiilor	294
7.3	Rezolvarea numerică a ecuațiilor neliniare unidimensionale	298
7.4	Complexitatea și analiza erorilor în metodele iterative	309
7.5	Metode iterative pentru rezolvarea sistemelor algebrice neliniare	311
7.6	Metoda iterației simple	314
7.7	Metoda Newton	316
7.8	Metoda Newton discretă	321
7.9	Metode quasi - Newton	325
7.10	Metode de relaxare	326
7.11	Metoda parametrului suplimentar	329
7.12	Metode de minimizare	332
7.13	Metoda aproximării liniare pe porțiuni	338
7.14	Analiza numerică a circuitelor rezistive neliniare	342
7.14.1	Caracterizarea metodei	342
7.14.2	Principiul metodei	342
7.14.3	Pseudocodul metodei	344
7.14.4	Analiza necesarului de memorie și a efortului de calcul	347

8	Ecuatii diferențiale ordinare	265
8.1	Formularea corectă a problemelor cu condiții inițiale	266
8.2	Discretizarea ecuațiilor diferențiale	268
8.3	Metode numerice cu un pas	278
8.3.1	Metoda seriei Taylor	278
8.3.2	Metoda Runge-Kutta	279
8.3.3	Controlul automat al mărimii pasului de integrare	288
8.4	Metode de integrare multipas	291
8.4.1	Metoda explicită Adams-Bashforth	293
8.4.2	Metoda implicită Adams-Moulton	295
8.4.3	Metoda Milne	298
8.4.4	Consistența, stabilitatea și convergența metodelor multipas	299
8.5	Algoritmul predictor-corector	309
8.6	Reprezentarea canonică a metodelor multipas	312
8.7	Metode cu valori multiple	316
8.7.1	Reprezentarea Nordsieck	317
8.7.2	Controlul automat al ordinului și mărimii pasului în metodele cu valori multiple	320
8.8	Integrarea numerică a ecuațiilor de tip stiff	324
8.8.1	Ecuatii diferențiale de tip stiff	324
8.8.2	Stabilitatea numerică a ecuațiilor stiff	327
8.8.3	Algoritmul Gear pentru rezolvarea ecuațiilor stiff	333
8.8.4	Alte metode numeric stiff stabile	337
8.9	Analiza numerică a circuitelor electrice în regim tranzitoriu	339
8.9.1	Principiul metodei	340
8.9.2	Pseudocodul metodei modelului discretizat	341
8.9.3	Analiza algoritmului	342
	ANEXE	345
	A	345
	Bibliografie și Webografie	348
	Index	352

Capitolul 8

Ecuatii diferențiale ordinare

Ecuatiile și sistemele de ecuații diferențiale ordinare au un rol central în modelarea fenomenelor fizice, chimice, sociologice, biologice sau de altă natură, caracterizate de un număr finit de mărimi de stare variabile în timp. Exemplul tipic în ingineria electrică îl reprezintă circuitele electrice în regim tranzitoriu, care sunt caracterizate de sisteme de astfel de ecuații. De altfel, teoria sistemelor cu o multitudine de aplicații în ingineria electrică, este bazată pe astfel de ecuații.

Forma canonică a sistemelor de ecuații diferențiale este:

$$\begin{cases} \frac{dx_1}{dt} = f_1(x_1, x_2, \dots, x_n, t) \\ \frac{dx_2}{dt} = f_2(x_1, x_2, \dots, x_n, t) \\ \vdots \\ \frac{dx_n}{dt} = f_n(x_1, x_2, \dots, x_n, t), \end{cases}$$

sau sub formă compactă:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t), \quad (8.1)$$

unde $\mathbf{f} : \mathbf{R}^{n+1} \rightarrow \mathbf{R}^n$ este o funcție vectorială cu valori în spațiul euclidian n -dimensional, $\mathbf{x} \in \mathbf{R}^n$ vectorul soluție, funcție de variabila scalară, $t \in \mathbf{R}$.

În continuare, atunci când acest lucru nu generează confuzii, vectorul soluție va fi notat și cu x , derivata cu x' , iar funcția cu f , astfel încât relația $x' = f(x, t)$ este echivalentă cu (8.1).

Trebuie remarcat că o ecuație diferențială arbitrară de ordin superior: $x^{(n)} = f(x, x', \dots, x^{(n-1)}, t)$ poate fi adusă la forma canonică prin simpla notație:

$$\begin{cases} x'_i(t) = x_{ii}(t), & i = 1, 2, \dots, n \\ f'_n(t) = f(t, x_1(t), x_2(t), \dots, x_n(t)). \end{cases}$$

După cum se va constata ulterior, pentru ca acest sistem să aibă soluție unică este necesară adăugarea unor condiții suplimentare impuse soluției, numite condiții inițiale dacă se referă la o valoare particulară a variabilei t , sau condiții **bilocale**, dacă se referă la două valori particulare ale variabilei independente.

8.1 Formularea corectă a problemelor cu condiții inițiale

Abordarea ecuației (8.1) prin metode numerice este lipsită de sens, dacă problema nu este bine formulată din punct de vedere matematic. Pentru ecuațiile care nu au soluție sau au mai multe soluții, chiar și cele mai “performante” metode de rezolvare numerică nu fac altceva decât să genereze un șir de numere lipsite de semnificație.

Din acest motiv, analiza calitativă a ecuației care trebuie rezolvată, din punctul de vedere al bunei formulări a problemei asociate, este esențială și trebuie efectuată înaintea rezolvării numerice. În formularea sa clasică, datorată lui Cauchy, problema rezolvării ecuațiilor diferențiale constă în determinarea funcției vectoriale $\mathbf{x} = \hat{\mathbf{x}}(t)$ de variabila reală t , $\hat{\mathbf{x}} : [t_0, t_m] \rightarrow \mathbf{R}^n$, funcție continuă și derivabilă care satisface ecuația (8.1) pe intervalul $[t_0, t_m]$ și condiția:

$$\hat{\mathbf{x}}(t_0) = \mathbf{x}_0, \quad (8.2)$$

cu $\mathbf{x}_0 \in \mathbf{R}^n$ dat, numită condiție inițială.

Condiția ca problema lui Cauchy să aibă soluție este dată de teorema lui Peano, care impune în principiu doar continuitatea funcției f în punctul (x_0, t_0) .

Teorema 8.1 de existență a lui Peano

Dacă funcția $f(\mathbf{x}, t)$ este continuă pe mulțimea:

$$D : \|\mathbf{x} - \mathbf{x}_0\| \leq a, \quad t_0 \leq t \leq t_m,$$

și care în plus satisface condiția inițială (8.2), atunci există cel puțin o funcție $\hat{\mathbf{x}}(t)$, care satisface ecuația diferențială (8.1) pe un interval nevid:

$$t_0 \leq t \leq t_0 + h \leq t_m.$$

Teorema lui Peano nu garantează unicitatea soluției ci doar existența acesteia. Intervalul pe care este asigurată existența soluției nu este nemărginit ci are o lungime dată de:

$$h = \min \left(t_m - t_0, \frac{a}{M} \right), \quad (8.3)$$

unde M este o margine superioară a valorii absolute a lui $\mathbf{f}(\mathbf{x}, t)$ pe D . Din acest motiv, teorema lui Peano este o teoremă de existență locală.

Demonstrația teoremei lui Peano este prezentată în [20], [30].

Pentru a obține unicitatea soluției este necesară întărirea condițiilor din teorema lui Peano, până la o condiție de tip **Lipschitz**.

Definiția 8.1 O funcție $f(\mathbf{x}, t) : \mathbf{R}^{n+1} \rightarrow \mathbf{R}$ satisface condiția Lipschitz față de \mathbf{x} pe o mulțime închisă $D \subset \mathbf{R}^{n+1}$, dacă există o constantă L , astfel încât:

$$\|f(\mathbf{x}', t) - f(\mathbf{x}'', t)\| \leq L\|\mathbf{x}' - \mathbf{x}''\|, \quad (8.4)$$

pentru orice $(\mathbf{x}', t) \in D$ și $(\mathbf{x}'', t) \in D$.

Funcțiile care satisfac condiția Lipschitz sunt funcții continue, nu neapărat derivabile, dar a căror “viteză de creștere” este mărginită. Funcțiile cu derivata continuă satisfac condiția Lipschitz dar nu și reciproc. De exemplu, funcțiile continue, liniare pe porțiuni satisfac definiția anterioară, dar nu sunt în mod necesar derivabile. Figura 8.1a reprezintă graficul unei funcții Lipschitz reale de o singură variabilă reală iar figura 8.1b reprezintă o funcție care nu satisface această condiție deoarece panta tangentei în x_0 este nemărginită.

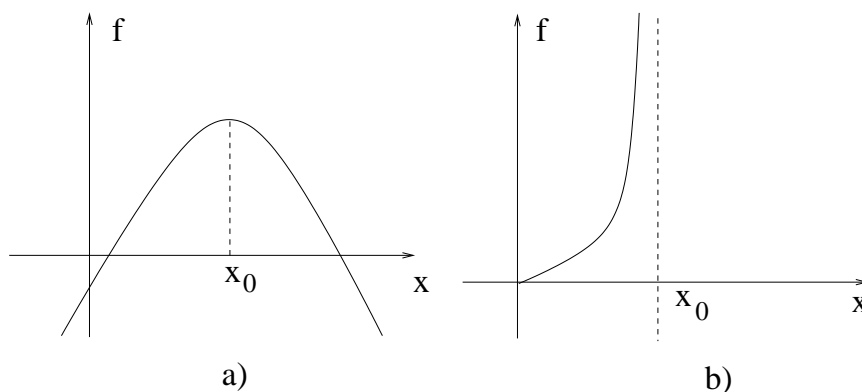


Figura 8.1: a) Funcție Lipschitziană; b) Funcție care nu satisface condiția Lipschitz

Teorema 8.2 de existență și unicitate a lui Picard

Dacă funcția $f(\mathbf{x}, t)$ este continuă pe mulțimea:

$$D : \|\mathbf{x} - \mathbf{x}_0\| \leq a, \quad t_0 \leq t \leq t_m,$$

și satisface condiția Lipschitz în raport cu \mathbf{x} pe D , atunci există o funcție unică $\hat{\mathbf{x}}(t)$, care satisface ecuația diferențială (8.1) pe un interval nevid $t_0 \leq t \leq t_0 + h \leq t_m$ și condiția inițială (8.2).

La fel ca teorema anterioară, și teorema lui Picard are un caracter local.

Întărirea condițiilor impuse funcției $f(\mathbf{x}, t)$ asigură nu numai unicitatea ci și dependența continuă a soluției $\hat{\mathbf{x}}(t)$ de datele inițiale \mathbf{x}_0 și t_0 .

Acest lucru este important mai ales când se urmărește obținerea unei soluții numerice, care pornește de la date inițiale obținute pe cale experimentală, deci susceptibile de erori.

Demonstrația teoremei de existență și unicitate a lui Picard este prezentată în [20], [30].

Pentru a obține existența soluției pe un interval de timp oricât de mare, este necesară impunerea unor condiții și mai severe pentru funcția $f(\mathbf{x}, t)$, așa cum rezultă din teorema următoare.

Teorema 8.3 de existență globală a lui Wintner

Dacă funcția $\mathbf{f}(\mathbf{x}, t)$ este continuă pe întregul spațiu $(n + 1)$ - dimensional \mathbf{R}^{n+1} și există o funcție reală de variabilă reală $L(r)$ cu proprietatea $\int_0^\infty \frac{dr}{L(r)} = \infty$, astfel încât:

$$|f_i(x_1, x_2, \dots, x_n, t)| < L(r), \quad i = 1, 2, \dots, n, \quad (8.5)$$

oricare ar fi $r = \|\mathbf{x}\| \in [0, \infty)$, atunci soluția ecuației diferențiale (8.1) cu condiția (8.2) există pentru orice $t \in \mathbf{R}$.

Teorema lui Wintner impune funcției $\mathbf{f}(\mathbf{x}, t)$ o creștere moderată spre infinit. Cu toate că polinoamele de gradul doi sau mai mare nu satisfac această condiție, totuși funcțiile continue, liniare pe porțiuni, satisfac această condiție, deci asigură existența globală a soluției.

În teoria circuitelor electrice, prezintă interes nu numai problemele cu condiții inițiale ci și cele cu condiții bilocale. Un exemplu de astfel de condiție la limită o reprezintă condiția de periodicitate:

$$\mathbf{x}(t_0) = \mathbf{x}(t_0 + T). \quad (8.6)$$

Din păcate, în cazul în care $\mathbf{f}(\mathbf{x}, t)$ nu este liniară față de \mathbf{x} această problemă nu are soluție în general unică. În schimb, dacă integrala pe o perioadă a funcției $\mathbf{f}(\mathbf{x}, t)$ satisface condiția Lipschitz $\left\| \int_{t_0}^{t_0+T} [\mathbf{f}(\mathbf{x}', t) - \mathbf{f}(\mathbf{x}'', t)] dt \right\| \leq L \|\mathbf{x}' - \mathbf{x}''\|$ cu constanta $L < 1$, atunci soluția ecuației (8.1) cu condiția (8.2) este unică.

8.2 Discretizarea ecuațiilor diferențiale

Determinarea numerică a soluției ecuației diferențiale (8.1) cu condiția inițială (8.2) presupune **discretizarea intervalului** $[t_0, t_m]$ cu pași de timp relativ mici:

$$t_0, \quad t_1 = t_0 + h_0, \quad t_2 = t_1 + h_1, \quad \dots, \quad t_{k+1} = t_k + h_k, \quad \dots, \quad t_m.$$

Dacă pasul de timp are valoare constantă $h_k = h$, atunci se obține o rețea uniformă de discretizare:

$$t_0, \quad t_1 = t_0 + h, \quad t_2 = t_0 + 2h, \quad \dots, \quad t_{k+1} = t_0 + (k + 1)h, \quad \dots, \quad t_n = t_0 + nh.$$

O metodă numerică de rezolvare reprezintă un algoritm care permite determinarea numerică a unei aproximări a soluției în nodurile rețelei de discretizare:

$$\hat{\mathbf{x}}(t_0), \quad \hat{\mathbf{x}}(t_1), \quad \hat{\mathbf{x}}(t_2), \quad \dots, \quad \hat{\mathbf{x}}(t_{k+1}), \quad \dots, \quad \hat{\mathbf{x}}(t_m).$$

Deoarece valorile soluției numerice nu sunt riguros egale cu soluția exactă, ele vor fi notate cu:

$$\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k, \dots, \mathbf{x}_m.$$

Abaterile soluției numerice \mathbf{x}_k de la pasul de timp t_k , față de soluția exactă $\mathbf{x}(t_k)$ la același moment de timp pot fi caracterizate prin eroarea absolută:

$$\varepsilon_k = \|\hat{\mathbf{x}}(t_k) - \mathbf{x}_k\|. \quad (8.7)$$

Această eroare este datorată, în principal a două cauze:

- aproximarea care stă la baza algoritmului, care determină o componentă a erorii numită eroare **de trunchiere**;
- faptul că în calculator numerele reale sunt reprezentate prin aproximări raționale, cu un număr finit de cifre semnificative, ceea ce determină erori numite **de rotunjire**.

Prima categorie de erori depinde de metoda numerică folosită iar a doua categorie este dependentă de sistemul de calcul utilizat.

În alegerea algoritmilor se urmărește ca erorile de orice natură să fie cât mai mici. Majoritatea algoritmilor asigură o scădere a erorii de trunchiere pe măsură ce pasul de discretizare a timpului, numit și pas de integrare, scade. Proprietatea unui algoritm de a asigura limita nulă a erorii de trunchiere $\varepsilon_k \rightarrow 0$ pentru orice $k = 1, 2, \dots, m$, atunci când $h \rightarrow 0$ poartă numele de **convergență**. În această definiție se presupune că soluția numerică îndeplinește exact condiția inițială ($\varepsilon_0 = \|\hat{\mathbf{x}}(t_0) - \mathbf{x}_0\| = 0$) și $h = \max(h_k)$.

Modul în care depinde eroarea de trunchiere de mărimea pasului poate fi caracterizat de *ordinul metodei*. Dacă există un număr p , real și pozitiv astfel încât eroarea de trunchiere este majorată de:

$$\varepsilon_k \leq Ch^p, \quad (8.8)$$

în care constanta C este independentă de pasul de timp h , atunci se spune că **metoda este de ordinul p** și se scrie că $\varepsilon_k = O(h^p)$. Este de așteptat ca metodele de ordin superior să asigure o eroare mai mică decât cele de ordin inferior, atunci când pasul de timp se micșorează.

Datorită caracterului recursiv al metodelor numerice de integrare a ecuațiilor diferențiale, în care \mathbf{x}_k este calculat de regulă în funcție de \mathbf{x}_{k-1} erorile apărute la pasul k se propagă la pașii ulteriori, afectând întreaga soluție. Pentru a studia propagarea erorii în procesul de calcul se consideră soluția numerică \mathbf{z}_k a ecuației diferențiale cu condiția inițială \mathbf{z}_0 , perturbată astfel încât: $\|\mathbf{x}_0 - \mathbf{z}_0\| = \varepsilon_0$.

O metodă de rezolvare se numește **numeric stabilă**, pentru un pas de timp h , dacă abaterea dintre cele două soluții este mărginită:

$$\|\mathbf{x}_k - \mathbf{z}_k\| = C\varepsilon_0, \quad (8.9)$$

în caz contrar metoda se numește *instabilă numeric*.

Dacă o metodă numerică determină “amplificarea” erorii în procesul de calcul, atunci ea nu poate fi utilizată practic în vederea obținerii unor soluții numerice cu eroare rezonabilă.

O metodă de rezolvare se numește **numeric absolut stabilă** pentru un pas de timp dat h și pentru o ecuație dată, dacă o perturbație la pasul de timp k : $\|\mathbf{x}_k - \mathbf{z}_k\| = \delta_k$, nu se amplifică la pașii de timp următori $m > k$:

$$\|\mathbf{x}_m - \mathbf{z}_m\| \leq \delta_k. \quad (8.10)$$

Din păcate, definiția stabilității numerice nu depinde numai de algoritm ci și de ecuația de rezolvat. Din acest motiv este necesară introducerea unei “ecuații test” pe care să se verifice stabilitatea absolută. Se alege ca ecuație test cea mai simplă ecuație diferențială liniară scalară:

$$\frac{dx}{dt} = \lambda x, \quad (8.11)$$

în care $\lambda \in \mathbf{C}$, deoarece multe sisteme liniare și chiar prima aproximație a celor neliniare se pot reduce la ecuații de acest tip, printr-o schimbare convenabilă de variabile. Se va numi **regiune de stabilitate numerică absolută** mulțimea valorilor pe care le poate lua pasul de timp h , la o valoare dată a parametrului λ , astfel încât o perturbație a soluției numerice la pasul k va produce modificări ulterioare care nu cresc de la pas la pas. Un exemplu de astfel de perturbări îl constituie imprecizia cu care sunt cunoscute datele inițiale sau erorile de rotunjire care intervin la fiecare pas de integrare în timp.

Pe măsură ce pasul de timp h scade, calculele sunt efectuate cu valori din ce în ce mai apropiate ($x_{k+1} \rightarrow x_k$) iar ponderea erorilor de rotunjire crește. Efectul cumulativ al erorilor de trunchiere și rotunjire este prezentat calitativ în figura 8.2. Se constată că în realitate pasul de timp nu trebuie să scadă sub o valoare minimă h_{min} deoarece erorile de rotunjire tind să crească din nou.

În concluzie, se poate afirma că prezintă interes practic doar acele metode care sunt convergente și numeric stabile, cu condiția ca pasul de integrare să nu fie mai mic decât limita impusă de eroarea de rotunjire.

În continuare va fi studiată cea mai simplă metodă de discretizare (de integrare numerică) a ecuațiilor diferențiale ordinare, și anume metoda de ordinul unu, cunoscută și sub numele de **metoda Euler**. Aceasta se obține prin aproximarea soluției exacte cu primii doi termeni din seria Taylor în care aceasta se poate descompune în fiecare punct al unei rețele uniforme de discretizare:

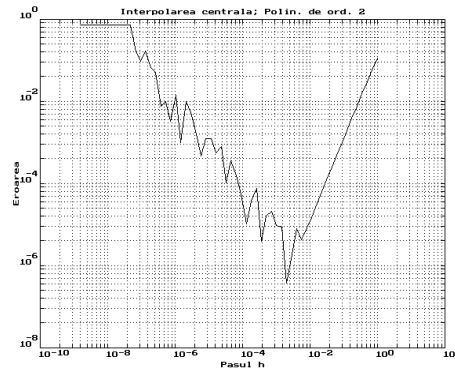


Figura 8.2: Variația erorii globale în funcție de pasul h

$$\hat{\mathbf{x}}(t_{k+1}) = \hat{\mathbf{x}}(t_k) + (t_{k+1} - t_k) \left. \frac{d\mathbf{x}}{dt} \right|_{t=t_k} + \dots \quad (8.12)$$

sau ținând seama de ecuația de rezolvat și de notațiile introduse, rezultă:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h\mathbf{f}(\mathbf{x}_k, t_k). \quad (8.13)$$

În fond, această relație se obține din ecuația de rezolvat aproximând derivata cu formula progresivă cu diferențe finite de ordinul întâi. Relația (8.13) definește o **metodă Euler explicită**, deoarece permite calculul direct al soluției numerice \mathbf{x}_{k+1} , fără să fie necesară rezolvarea vreunei ecuații sau sistem de ecuații.

Relația (8.13) poate fi obținută și prin integrarea ecuației diferențiale (8.1) pe ultimul pas de timp

$$\hat{\mathbf{x}}(t_{k+1}) - \hat{\mathbf{x}}(t_k) = \int_{t_k}^{t_{k+1}} \mathbf{f}(\mathbf{x}, t) dt, \quad (8.14)$$

și aproximarea integralei prin metoda dreptunghiurilor cu valoarea integrandului din momentul inițial t_k , al intervalului.

Dacă, în schimb, se folosește pentru evaluarea integralei valoarea din momentul final t_{k+1} , atunci se obține relația:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h\mathbf{f}(\mathbf{x}_{k+1}, t_{k+1}) \quad (8.15)$$

caracteristică **metodei Euler implicite**. Această relație se obține aproximând derivata din ecuația de rezolvat cu diferențe finite regresive. În acest ultim caz, necunoscuta \mathbf{x}_{k+1} , este prezentă atât în membrul stâng cât și în cel drept iar pentru determinarea ei este necesară rezolvarea unui sistem de ecuații neliniare. Efortul de calcul în cazul metodelor implicite este mai mare, dar se va constata că el este recompensat de o stabilitate numerică superioară a soluției.

Semnificația geometrică a metodei Euler explicită este reprezentată în figura 8.3, pentru cazul unei funcții $f(x, t)$ cu $x \in \mathbf{R}$. Se constată că soluția este aproximată cu o dreaptă care trece prin punctul inițial (x_k, t_k) și a cărei pantă este dată de valoarea funcției $f(x, t)$ în punctul inițial. Pe măsură ce pasul de timp h scade, are loc și o scădere a erorii de trunchiere ε_k .

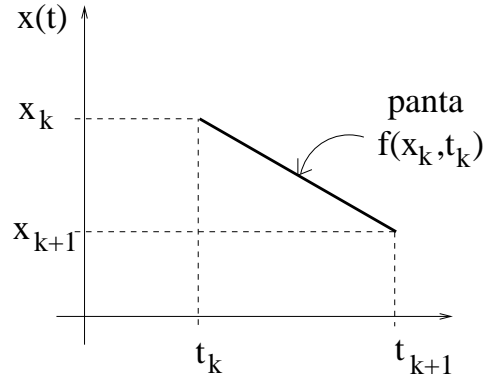


Figura 8.3: Semnificația geometrică a metodei Euler explicite

Pentru a determina **ordinul de convergență** al metodei se consideră ecuația test (8.11) cu $\lambda \in \mathbf{R}$, a cărei soluție este:

$$\hat{x}(t) = x_0 e^{\lambda t},$$

și care, în nodurile $x_k = kh$ ale unei rețele regulate are valorile:

$$\hat{x}(t_k) = x_0 e^{\lambda h t}.$$

Soluția numerică se obține folosind recursiv relația (8.13):

$$x_{k+1} = x_k + h\lambda x_k = x_0(1 + \lambda h)^{k+1}.$$

Dacă se notează:

$$\alpha = e^{\lambda h} = 1 + \lambda h + \frac{(\lambda h)^2}{2} e^{\theta \lambda h}, \quad 0 < \theta < 1 \quad \text{și} \quad (8.16)$$

$$\beta = 1 + \lambda h \leq \alpha,$$

rezultă următoarele expresii pentru soluția exactă și cea numerică:

$$\hat{x}(t_k) = x_0 \alpha^k, \quad (8.17)$$

$$x_k = x_0 \beta^k. \quad (8.18)$$

Eroarea de trunchiere are la pasul k valoarea:

$$\varepsilon_k = |\hat{x}(t_k) - x_k| = x_0 |\alpha^k - \beta^k| = x_0 |\alpha - \beta| |\alpha^{k-1} + \beta \alpha^{k-2} + \dots + \beta^{k-1}|.$$

Luând în considerație (8.16), rezultă:

$$\varepsilon_k \leq x_0 \frac{(\lambda h)^2}{2} e^{\theta \lambda h} k \alpha^{k-1} \leq x_0 h^2 k \lambda^2 e^{\lambda k h}, \quad (8.19)$$

Fixând valoarea lui $t_k = kh$, rezultă evaluarea:

$$e_k \leq h x_0 \lambda^2 t_k e^{\lambda t_k} = O(h), \quad (8.20)$$

care evidențiază faptul că metoda Euler este convergentă cu ordinul 1. Relația (8.20) evaluează eroarea cumulată pe toți pașii de timp până la t_k , motiv pentru care ea se numește **eroare globală**. Eroarea de trunchiere efectuată la fiecare pas, presupunând valoarea soluției la pasul anterior ca exactă, se numește **eroare locală** și se poate calcula folosind relația (8.20) în care se particularizează $k = 1$:

$$e_1 \leq x_0 h^2 \lambda^2 e^{\lambda h} = O(h^2). \quad (8.21)$$

Se constată că **eroarea locală** este cu un ordin de mărime mai mare decât cea globală.

Presupunând că funcția $f(x, t)$ satisface condiția Lipschitz și că soluția are derivata dublă mărginită, se obțin [47] aceleași rezultate privind ordinul erorii metodei Euler și în cazul general al ecuației (8.1).

Pentru a studia **stabilitatea numerică** a metodei Euler se consideră soluția ecuației de test (8.11) și soluția ecuației cu condiții inițiale perturbate:

$$z_k = z_0 \beta^k.$$

Abaterea soluției perturbate:

$$|x_k - z_k| = |x_0 - z_0| \beta^k \leq \varepsilon_0 \alpha^k = \varepsilon_0 e^{\lambda h k} = \varepsilon_0 e^{\lambda t_k} \quad (8.22)$$

este mărginită de abaterea condiției inițiale $\varepsilon_0 = |x_0 - z_0|$ și tinde către 0 atunci când $\varepsilon_0 \rightarrow 0$.

În consecință, se poate afirma că metoda Euler este numeric stabilă pentru orice valoare a pasului de integrare. Metoda este numeric stabilă nu numai pentru ecuațiile stabile ($Re[\lambda] < 0$) ci și pentru cele instabile, a căror soluție nu este asimptotic mărginită. Această afirmație este adevărată și în cazul ecuației diferențiale (8.1). În acest caz [26], abaterea celor două soluții este mărginită de:

$$\|\mathbf{x}_k - \mathbf{z}_k\| \leq \varepsilon_0 e^{Lh},$$

unde L este constanta Lipschitz a funcției $\mathbf{f}(\mathbf{x}, t)$.

Stabilitatea numerică a unei metode nu presupune convergența acesteia. De exemplu "metoda de ordin 0", obținută prin reținerea primului termen din seria Taylor (8.13):

$$\mathbf{x}_1 = \mathbf{x}_0, \quad \mathbf{x}_2 = \mathbf{x}_1, \quad \mathbf{x}_3 = \mathbf{x}_2, \quad \dots, \quad \mathbf{x}_{k+1} = \mathbf{x}_k, \quad \dots$$

este numeric stabilă (chiar și absolut stabilă) dar nu este convergentă.

Pentru a asigura **stabilitatea numerică absolută** este necesar ca în (\mathbf{x}_1) să fie îndeplinită condiția:

$$|x_k - z_k| \leq |x_0 - z_0|,$$

sau echivalent

$$|\beta|^k = |1 + \lambda h|^k \leq 1, \quad (8.23)$$

ceea ce corespunde în planul complex λh unui cerc de rază unitate, cu centrul în $(-1, 0)$.

Acest cerc (figura 8.4) reprezintă domeniul de stabilitate numerică absolută pentru metoda Euler.

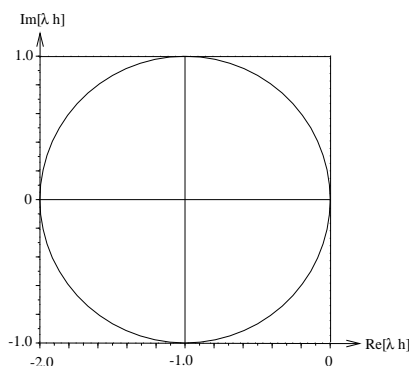


Figura 8.4: Domeniul de stabilitate numerică absolută pentru o metodă numerică

Dacă se consideră spre exemplu ecuația unui circuit electric liniar de ordinul 1, cu constanta de timp τ , care este de forma ecuației de test (8.11) cu $\lambda = -1/\tau$. În acest caz condiția de stabilitate numerică absolută este $-2 \leq \lambda h \leq 0$. Dacă se consideră pasul de timp limită $h = -\frac{2}{\lambda} = 2\tau$, atunci soluția numerică obținută prin metoda Euler este

$$x_k = (1 + \lambda h)x_{k-1} = (-1)^k x_0.$$

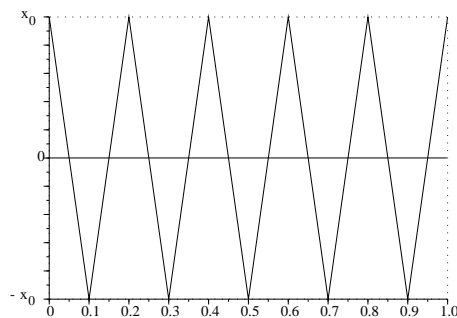


Figura 8.5: Soluția numerică Euler a unui circuit liniar de ordinul 1, pentru $h = 2\tau$

Aceasta are un caracter oscilator (figura 8.5), în timp ce soluția exactă tinde către zero în câțiva pași de timp. Dacă pasul de timp depășește această limită, atunci oscilațiile se amplifică, soluția numerică devine instabilă și se îndepărtează cu fiecare pas de integrare de soluția exactă a ecuației de rezolvat. Pentru a obține o variație monotonă a soluției numerice, așa cum variază soluția exactă, este necesar ca pasul de integrare h să fie mai mic decât constanta de timp τ a circuitului.

Pentru a caracteriza **eroarea de rotunjire** se consideră un sistem de calcul care operează cu q cifre semnificative. La fiecare operație acesta introduce o eroare relativă de rotunjire $r \in [-1/2, 1/2] \cdot 10^{-q}$. Dacă pe acest sistem se implementează metoda Euler pentru ecuația (8.11) cu $x_0 = 1$ și $\lambda \equiv -1/2$, atunci eroarea de rotunjire la pasul $t_k = \tau$ este:

$$\varepsilon_r = Nr \cong \frac{\tau}{h} 10^{-q},$$

deoarece au fost efectuați $N = \tau/h$ pași. Eroarea de trunchiere la același pas se obține pe baza relațiilor (8.19), (8.20):

$$\varepsilon_t = \frac{h}{2\tau^2} \tau e^{-1} \cong \frac{h}{\tau}.$$

Suma acestor erori:

$$\varepsilon = \varepsilon_t + \varepsilon_r = \frac{h}{\tau} + \frac{\tau}{h} 10^{-q},$$

are minimul dat de ecuația:

$$\frac{\partial \varepsilon}{\partial h} \equiv \frac{1}{\tau} - \frac{\tau}{h^2} 10^{-q} = 0,$$

ceea ce corespunde unei valori optime a pasului de timp:

$$h = \tau \sqrt{10^{-q}}.$$

De exemplu, pentru cazul uzual în care $q = 6$, rezultă că pasul metodei Euler trebuie ales mai mare de $1/1000$ din constanta de timp maximă din circuit dar mai mic decât constanta de timp minimă din circuit.

Rezolvarea prin metoda Euler explicită este descrisă de următorul pseudocod:

```

procedura euler_exp (x0, xmax, y0, h, y)
  real x0,                ;nod inițial
      xmax,                ;nod final
      y0,                  ;condiția inițială
      h                    ;pasul de integrare
  tablou real y(N)       ;vectorul soluției
                          ;numerice

  t = x0
  y(1) = y0
  n = (xmax - x0)/h      ;nr. de pași

```

```

    pentru i = 2,n          ;evaluează funcția
                           ;în cele n noduri
        y(i) = y(i-1) + hfunc(t,y(i-1))
        t = t + h
    retur

```

Procedura are următorii parametri :

- de intrare
 - x_0 = limita inferioară a intervalului de integrare;
 - x_{\max} = limita superioară a intervalului de integrare;
 - y_0 = condiția inițială;
 - h = pasul de integrare.
- de ieșire
 - $y(N)$ = vectorul soluție.

Procedura apelează funcția “**func**” ce evaluează funcția $f(t, y)$ din membrul drept al ecuației (8.1) pentru nodul i .

Rezolvarea prin metoda Euler implicită este descrisă de următorul pseudocod:

```

procedura euler_imp (x0, xmax, y0, h, err, itmax, y)
    real x0,          ;nod inițial
        xmax,        ;nod final
        y0,          ;condiția inițială
        h,           ;pasul de integrare
        err          ;eroarea maxim admisă
    întreg itmax     ;nr. max. de iterații
    tablou real y(N) ;vectorul soluției

    t = x0
    y(1) = y0
    n = (xmax - x0)/h ;nr. de pași
    pentru i = 2,n    ;evaluează funcția
                     ;în cele n noduri
        t = t + h    ;pas nou
        ;prima evaluare se face cu metoda explicită
        ynou = y(i-1)+hfunc(t,y(i-1))

        j = 0       ;contor iterații
    repetă

```

```

        yvechi = ynou
        ynou = y + hfunc(t,yvechi)
                ;evaluare nouă
        j = j + 1
        eps = abs(yvechi - ynou)
                ;evaluare eroare
    până când ( abs(eps) ≤ err sau j > itmax )
    y(i) = ynou

    retur

```

Față de metoda Euler explicită, procedura de mai sus are doi parametri suplimentari:

- de intrare
 - `err` = eroarea maximă admisă;
 - `itmax` = nr. maxim de iterații admis.

Deci, metoda de rezolvare a ecuației neliniare (8.5) este o metodă iterativă, la care inițializarea este făcută prin valori date de relația explicită (8.3).

În figurile 8.6 – 8.8 sunt prezentate soluția exactă și soluțiile numerice prin metodele Euler explicită și implicită, la rezolvarea ecuației diferențiale a unui circuit *RC* serie:

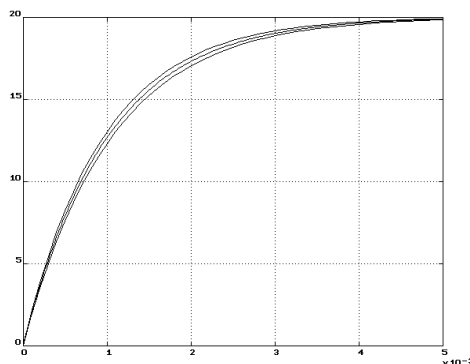
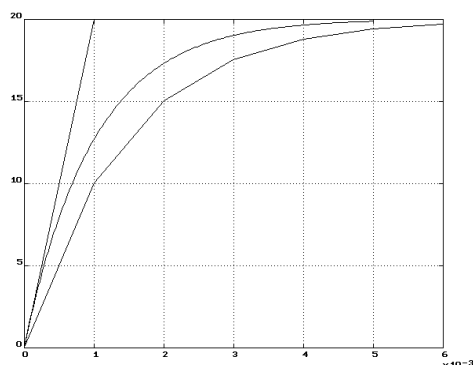
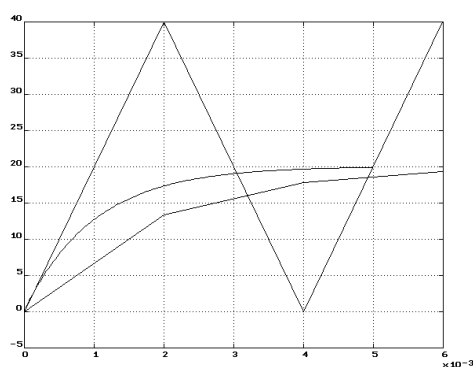


Figura 8.6: Tensiunea la bornele condensatorului pentru $h = \tau/10$

$$\begin{cases} RC \frac{du}{dt} + u = E, \\ u(0) = 0. \end{cases}$$

pentru valorile pasului de discretizare temporală $h = \tau/10$, $h = \tau$ și respectiv $h = 2\tau$.

Se constată că metoda Euler implicită este mult mai robustă, fiind mai puțin sensibilă la mărimea pasului.

Figura 8.7: Tensiunea la bornele condensatorului pentru $h = \tau$ Figura 8.8: Tensiunea la bornele condensatorului pentru $h = 2\tau$

Un alt fenomen interesant este acela că, pentru o valoare suficient de mică a pasului, soluția exactă este încadrată de cele două soluții numerice (soluția obținută prin metoda Euler explicită fiind întotdeauna mai mare, iar cea obținută prin metoda Euler implicită fiind întotdeauna mai mică decât cea exactă). Această observație permite rezolvarea unei ecuații diferențiale ordinare cu control al erorii, prin utilizarea **ambelor** metode.

8.3 Metode numerice cu un pas

8.3.1 Metoda seriei Taylor

Pentru a obține îmbunătățirea convergenței față de metoda Euler, se folosesc metode în care din seria Taylor a soluției se rețin mai mult de doi termeni. În urma trunchierii seriei Taylor la primii $(p + 1)$ termeni se obține relația:

$$x_{k+1} = x_k + hf(x_k, t_k) + \frac{h^2}{2!} f^{(1)}(x_k, t_k) + \dots + \frac{h^p}{p!} f^{(p-1)}(x_k, t_k). \quad (8.24)$$

Dacă se aplică această metodă ecuației test (8.11) cu $\lambda \in \mathbf{R}$, atunci soluția numerică se

obține de forma:

$$x_k = x_0 \left(1 + \lambda h + \frac{(\lambda h)^2}{2!} + \dots + \frac{(\lambda h)^p}{p!} \right)^k. \quad (8.25)$$

Folosind notațiile:

$$\beta = 1 + \lambda h + \frac{(\lambda h)^2}{2!} + \dots + \frac{(\lambda h)^p}{p!};$$

$$\alpha = e^{\lambda h} = \beta + \frac{(\lambda h)^{p+1}}{p!} \epsilon \theta \lambda h \geq \beta;$$

în care $\theta \in (0, 1)$, rezultă următoarele expresii pentru soluția exactă și cea numerică:

$$\hat{x}(t_k) = x_0 \alpha^k, \quad (8.26)$$

$$x_k = x_0 \beta^k. \quad (8.27)$$

Eroarea de trunchiere satisface majorările:

$$\begin{aligned} \varepsilon_k &= |\hat{x}(t_k) - x_k| = x_0 |\alpha^k - \beta^k| \leq x_0 |\alpha - \beta| k \alpha^{k-1} \leq \\ &\leq \frac{x_0 (\lambda h)^{p+1}}{p!} e^{\theta \lambda h} k e^{\lambda h(k-1)} \leq h^p x_0 |\lambda|^p t_k e^{\lambda t_k}. \end{aligned} \quad (8.28)$$

Rezultă că această metodă este convergentă și are ordinul p , eroarea de trunchiere fiind de tipul:

$$\varepsilon_k = O(h^p).$$

În această metodă, micșorarea erorii se poate realiza atât prin micșorarea pasului de timp h cât și prin mărirea ordinului p al metodei. Micșorarea excesivă a pasului h duce la creșterea erorii de rotunjire; creșterea ordinului la valori mai mari de 1 presupune calculul numeric sau analitic al valorilor funcției $f(x, t)$, fapt ce sporește sensibil efortul de calcul și este susceptibil de erori suplimentare. Din acest motiv metoda seriei Taylor trunchiată este folosită în practică doar pentru ordinul $p = 1$.

8.3.2 Metoda Runge-Kutta

O clasă de metode de ordin superior, care asigură aceeași comportare a erorii ca metoda Taylor, dar nu pretinde calculul derivatelor funcției $f(x, t)$ este cunoscută sub numele Runge-Kutta. Pentru a asigura același ordin al erorii, metodele Runge-Kutta evaluează funcția $f(x, t)$ de mai multe ori, într-un număr p de puncte în vecinătatea punctului inițial (x_k, t_k) .

Metoda Runge-Kutta de ordinul doi utilizează relația de recurență:

$$x_{k+1} = x_k + h[a_1 f(x_k, t_k) + a_2 f(x_k + b_1 h f(x_k, t_k), t_k + b_2 h)], \quad (8.29)$$

în care, constantele a_1, a_2, b_1 și b_2 se determină astfel încât eroarea de trunchiere să fie de tipul $O(h^2)$. Dacă se notează:

$$\begin{aligned} g_1 &= f(x_k, t_k), \\ g_2 &= f(x_k + b_1 h g_1, t_k + b_2 h), \end{aligned}$$

relația (8.29) devine:

$$x_{k+1} = x_k + h[a_1 g_1 + a_2 g_2].$$

Derivând funcția $f(x, t)$ față de timp:

$$f^{(1)}(x_k, t_k) = \frac{\partial f}{\partial x} f(x_k, t_k) + \frac{\partial f}{\partial t}, \quad (8.30)$$

rezultă prin reținerea primilor termeni ai dezvoltării în serie Taylor a funcției g_2 :

$$g_2 = f(x_k, t_k) + \frac{\partial f}{\partial x} f(x_k, t_k) b_1 h + \frac{\partial f}{\partial t} b_2 h + O(h^2),$$

iar prin substituție în (8.29), rezultă:

$$x_{k+1} = x_k + h(a_1 + a_2)g_1 + h^2 a_2 b_1 \frac{\partial f}{\partial x} g_1 + h^2 a_2 b_2 \frac{\partial f}{\partial t} + O(h^3). \quad (8.31)$$

Identificând termenii expresiei (8.31) cu cei ai expresiei (8.24) particularizată pentru $p = 2$:

$$x_{k+1} = x_k + h g_1 + \frac{h^2}{2} \left[\frac{\partial f}{\partial x} g_1 + \frac{\partial f}{\partial t} \right],$$

rezultă sistemul de ecuații:

$$a_1 + a_2 = 1; \quad a_2 b_1 = 1/2; \quad a_2 b_2 = 1/2. \quad (8.32)$$

Cele trei relații obținute nu pot determina în mod unic cei patru parametri și unul dintre aceștia trebuie ales arbitrar. Indiferent de această alegere, metoda obținută este convergentă cu o eroare de trunchiere de ordinul doi $\varepsilon_k = O(h^2)$. Se constată că în metoda Runge-Kutta de ordinul doi, la fiecare pas de timp se evaluează funcția $f(x, t)$ de două ori, o dată în punctul inițial (g_1) și apoi într-un punct intermediar (g_2), plasat pe direcția g_1 , urmând ca valoarea acceptată pentru soluția numerică să medieze ponderat cele două evaluări.

În practică se folosesc mai des următoarele variante ale metodei Runge-Kutta de ordinul doi:

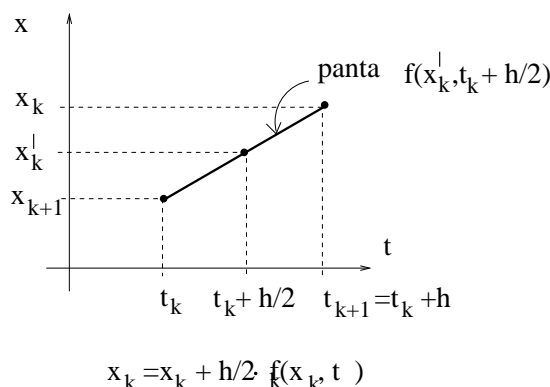


Figura 8.9: Semnificația geometrică a metodei Euler – Cauchy

- **metoda Euler-Cauchy**, obținută pentru $a_2 = 1$:

$$x_{k+1} = x_k + hf \left(x_k + \frac{h}{2} f(x_k, t_k), t_k + \frac{h}{2} \right), \quad (8.33)$$

în care se evaluează funcția $f(x, t)$ la jumătatea pasului de timp, într-un punct median, de pe segmentul de pantă $f(x_k, t_k)$, ce pornește din punctul inițial (semnificația geometrică este prezentată în figura 8.9);

- **metoda Henn**, obținută pentru $a_2 = 1/2$:

$$x_{k+1} = x_k + \frac{h}{2} [f(x_k, t_k) + f(x_k + hf(x_k, t_k), t_k + h)], \quad (8.34)$$

în care se utilizează media aritmetică a valorilor funcției $f(x, t)$ în punctul inițial (x_k, t_k) și cel final $(x_k + hf(x_k, t_k), t_k + h)$, în sensul indicat de metoda Euler (semnificația geometrică este prezentată în figura 8.10);

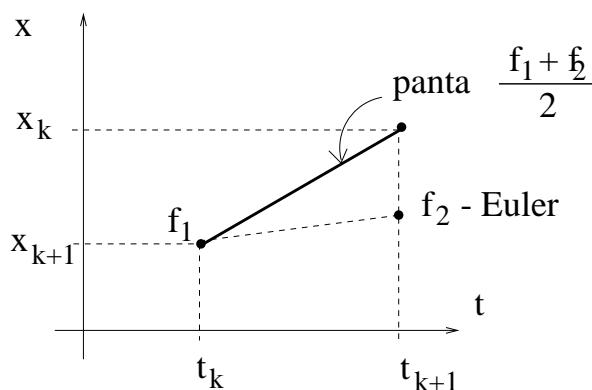


Figura 8.10: Semnificația geometrică a metodei Henn

- **metoda Ralston**, obținută pentru $a_2 = 3/4$:

$$x_{k+1} = x_k + \frac{h}{4} \left[f(x_k, t_k) + 3f \left(x_k + \frac{2}{3} hf(x_k, t_k), t_k + \frac{2}{3} h \right) \right], \quad (8.35)$$

în care se utilizează o medie ponderată a valorii inițiale (pondere $1/4$) și a valorii din punctul situat la $2/3$ din pasul h (pondere $3/4$). Această alegere asigură minimul erorii de trunchiere [26].

Folosind o tehnică asemănătoare se obține metoda Runge-Kutta de **ordinul trei** cu o eroare de trunchiere de tipul $O(h^3)$ și necesită trei evaluări ale funcției $f(x, t)$ la fiecare pas de timp. În acest caz numărul coeficienților se ridică la opt iar numărul ecuațiilor obținute prin identificarea cu seria Taylor este doar de șase, deci doi parametri rămân nedeterminați [2], [26].

Dintre toate variantele metodei Runge-Kutta, cea mai des folosită în practică este cea de **ordinul patru**:

$$x_{k+1} = x_k + h[a_1g_1 + a_2g_2 + a_3g_3 + a_4g_4], \quad (8.36)$$

cu

$$\begin{aligned} g_1 &= f(x_k, t_k); \\ g_2 &= f(x_k + b_1g_1h, t_k + c_1h); \\ g_3 &= f(x_k + b_2g_1^h + b_3g_2^h, t_k + c_2h); \\ g_4 &= f(x_k + b_4g_1^h + b_5g_2^h + b_6g_3^h, t_k + c_3h). \end{aligned}$$

În urma dezvoltării în serie Taylor a funcțiilor ce intervin în expresia (8.36) și a identificării cu relația (8.24), particularizată pentru $p = 4$, rezultă un sistem de 11 ecuații satisfăcute de cei 13 coeficienți $a_1 - a_4$, $b_1 - b_6$, $c_1 - c_4$.

Cel mai des utilizată este varianta Runge, în care se alege $b_6 = 1$ și $c_1 = 1/2$:

$$x_{k+1} = x_k + h(g_1 + 2g_2 + 2g_3 + g_4)/6 \quad (8.37)$$

cu

$$\begin{aligned} g_1 &= f(x_k, t_k); \\ g_2 &= f\left(x_k + \frac{h}{2}g_1, t_k + \frac{h}{2}\right); \\ g_3 &= f\left(x_k + \frac{h}{2}g_2, t_k + \frac{h}{2}\right); \\ g_4 &= f(x_k + hg_3, t_k + h). \end{aligned}$$

Această metodă calculează panta g_1 , în punctul inițial (x_1, t_1) , pe care o folosește pentru a avansa cu jumătate de pas $h/2$, în vederea calculului pantei g_2 . Valoarea g_2 astfel calculată este folosită pentru a recalcula panta g_3 la jumătatea pasului. Această pantă este folosită pentru calculul valorii $f(x, t)$ a funcției g_4 la sfârșitul pasului de timp. Cele patru valori obținute sunt apoi mediate cu ponderile $1/6$, $2/6$, $2/6$ și respectiv $1/6$, în vederea calculului noii stări x_{k+1} .

Această metodă are avantajul unei erori de trunchiere relativ mici, de tipul $O(h^4)$ și a faptului că permite o implementare simplă în limbajele de programare. Efortul constă, în principal, în evaluarea funcției $f(x, t)$ de patru ori la fiecare pas de timp.

Folosirea metodelor de tip Runge-Kutta de **ordin mai mare de 4** nu este justificată, deoarece pentru obținerea unei erori de tip $O(h^5)$ este necesară evaluarea funcției $f(x, t)$

de șase ori. Metodele de ordin 6, 7 și 8 necesită un număr de 7, 9 și respectiv 11 evaluări ale funcției $f(x, t)$ [?].

Pentru studiul **stabilității numerice absolute** se consideră din nou ecuația test (8.11), la care $f(x, t) = \lambda x$. Metoda Runge-Kutta de ordinul doi, aplicată acestei ecuații, conduce la:

$$x_{k+1} = x_k + h[a_1\lambda x_k + a_2\lambda(x_k + b_1h\lambda x_k)] = x_k \left[1 + \lambda h + \frac{(\lambda h)^2}{2} \right], \quad (8.38)$$

și respectiv pentru soluția perturbată:

$$z_{k+1} = z_k \left[1 + \lambda h + \frac{(\lambda h)^2}{2} \right].$$

Perturbația de la pasul k , $\delta_k = z_k - x_k$, se propagă la pasul $k + 1$, sub forma:

$$\delta_{k+1} = z_{k+1} - x_{k+1} = \left[1 + \lambda h + \frac{(\lambda h)^2}{2} \right] \delta_k.$$

Condiția ca această perturbație să se atenueze de la un pas la altul este:

$$\left[1 + \lambda h + \frac{(\lambda h)^2}{2} \right] \leq 1, \quad (8.39)$$

ceea ce corespunde în planul complex λh unei regiuni de stabilitate absolută de forma celei reprezentate în figura 8.11a. Dacă se analizează în mod asemănător metoda Runge-Kutta de ordinul patru, rezultă condiția:

$$\left[1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6} + \frac{(\lambda h)^4}{24} \right] \leq 1, \quad (8.40)$$

care este reprezentată grafic în figura 8.11b.

Prin compararea regiunilor de stabilitate numerică absolută, rezultă că metodele de ordin superior admit, pentru aceeași ecuație, valori mai mari ale pasului maxim admisibil h , decât metodele de ordin inferior.

Metoda Runge-Kutta de ordinul patru prezintă avantaje față de metoda Euler. De exemplu, pentru un pas de integrare $h = 0.1\tau$, metoda Runge-Kutta asigură o eroare cu trei ordine de mărime mai mică decât metoda Euler cu același pas. Pentru a obține o eroare comparabilă, pasul în metoda Euler trebuie micșorat la $h = 10^{-4}\tau$, ceea ce nu numai că sporește efortul de calcul de câteva sute de ori, dar și determină o creștere inadmisibilă a erorilor de rotunjire. Această afirmație se dovedește a fi adevărată în cazul în care se urmăresc erori mici de trunchiere. Dacă toleranța erorii este mare, metoda Euler poate fi mai avantajoasă din punctul de vedere al efortului de calcul.

Următorul algoritm rezolvă un sistem de N ecuații diferențiale de ordinul unu prin metoda Runge-Kutta de rang 4.

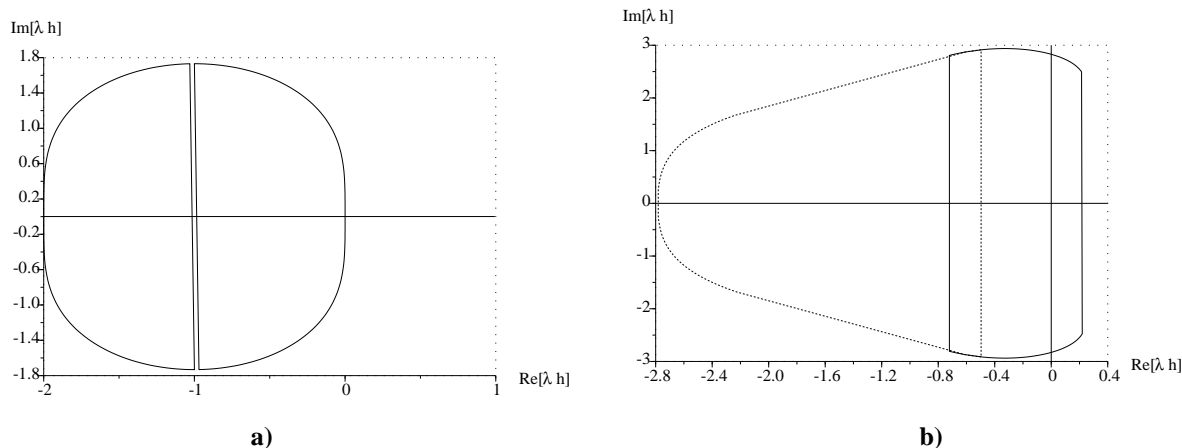


Figura 8.11: Domeniile de stabilitate absolută pentru metodele Runge-Kutta de ordinul 2 (a) și 4 (b).

```

procedura runge_4 (t0,tf,h,y) ;Rezolvă sistem de N ecuații
                                ;diferențiale de ordin I,
                                ;prin metoda Runge-Kutta de
                                ;rang 4.

real t0, ;moment inițial
        tf, ;limită interval de integrare
        h ;pas de integrare
tablou real y[N], s[N], z[N], g[N] ;N-dimensiune
                                ;sistem

t = t0
cât timp t < tf
    pentru i = 1,N
        g(i) = func (i,y(i),t) ;evaluare g1
    t = t + h/2
    pentru i = 1,N
        z(i) = y(i) + h·g(i)/2 ;argument pentru g2
        s(i) = g(i) ;salvează g1
    pentru i = 1,N
        g(i) = func (i,z(i),t) ;evaluare g2
    pentru i = 1,N
        z(i) = y(i) + h·g(i)/2 ;argument pentru g3
        s(i) = s(i) + 2·g(i) ;salvez g1+2·g2
    pentru i = 1,N
        g(i) = func (i,z(i),t) ;evaluare g3
    t = t + h/2
    pentru i = 1,N
        z(i) = y(i) + h·g(i) ;argument pentru g4
        s(i) = s(i) + 2·g(i) ;salvez

```

```

;g1+2·g2+2·g3
pentru i = 1,N
    g(i) = func (i,z(i),t) ;evaluare g4
pentru i = 1,N
    y(i) = y(i) + h·(s(i) + g(i))/6 ;soluție finală
    scrie y(i) ;salvează soluție
retur

```

În figurile 8.12 – 8.14 sunt prezentate soluția exactă și soluțiile numerice prin metoda Runge-Kutta de ordinul patru. Se constată că metoda este robustă, în mod asemănător cu metoda Euler implicită (vezi figurile 8.6 – 8.8). Totuși, metoda Runge-Kutta asigură erori mult mai mici decât metoda Euler explicită, așa cum rezultă din tabelul 8.3.2, în care sunt prezentate eroarea locală la primul pas și eroarea globală, la rezolvarea ecuației diferențiale

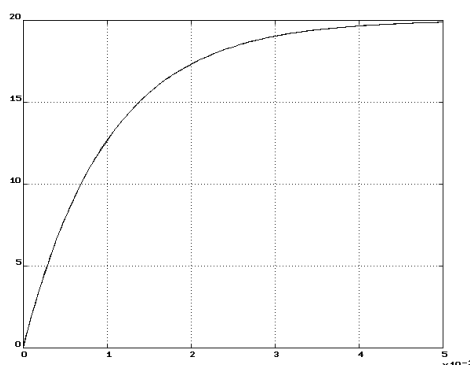


Figura 8.12: Tensiunea la bornele condensatorului pentru $h = \tau/10$

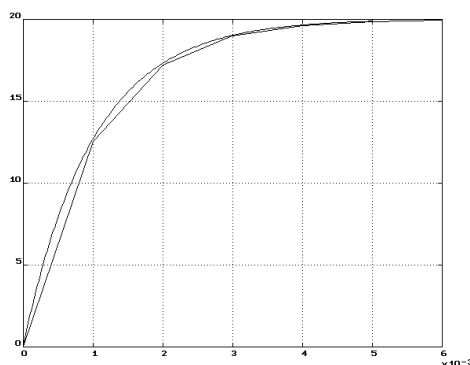


Figura 8.13: Tensiunea la bornele condensatorului pentru $h = \tau$

$$\begin{aligned} \frac{\partial y}{\partial t} &= -y, \\ y(0) &= 1, \end{aligned}$$

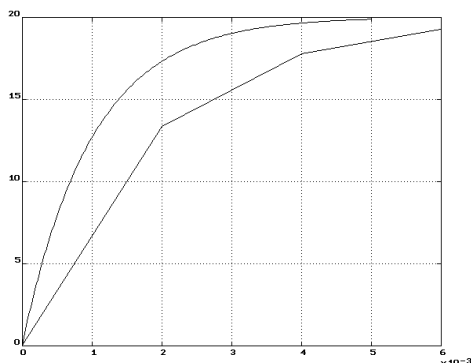


Figura 8.14: Tensiunea la bornele condensatorului pentru $h = 2\tau$

Tabela 8.1: Erori la rezolvarea ecuației diferențiale $dy/dt = -y$, $y_0 = 1$ $h \in [0, 4]$

	Euler	explicit	R-K	ord.4
h	e_l	e_g	e_l	e_g
0.001	$4.9 \cdot 10^{-7}$	$-9.8 \cdot 10^{-1}$	0	$5.3 \cdot 10^{-15}$
0.01	$4.9 \cdot 10^{-5}$	$-9.8 \cdot 10^{-1}$	$-8.3 \cdot 10^{-13}$	$-6.2 \cdot 10^{-12}$
0.1	$4.8 \cdot 10^{-3}$	$-9.8 \cdot 10^{-1}$	$-8.2 \cdot 10^{-8}$	$-6.6 \cdot 10^{-8}$
1	$3.7 \cdot 10^{-1}$	$-9.8 \cdot 10^{-1}$	$-7.1 \cdot 10^{-3}$	$-1.5 \cdot 10^{-3}$
2	1.13	$-9.8 \cdot 10^{-1}$	$-1.9 \cdot 10^{-1}$	$-9.3 \cdot 10^{-2}$

pe intervalul $[0, 4]$.

Graficele la rezolvarea cu metoda Runge-Kutta de ordinul patru a sistemelor de ecuații corespunzătoare circuitului din figura 8.15 sunt prezentate în figurile 8.16 – 8.18.

Circuitul din figura 8.15 este caracterizat de sistemul de ecuații:

$$\begin{cases} Li' + u + Ri = E, \\ Cu' = i, \end{cases}$$

sistem care poate fi adus la forma:

$$\begin{bmatrix} u' \\ i' \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{L} \\ -\frac{1}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} u \\ i \end{bmatrix} + \begin{bmatrix} E \\ 0 \end{bmatrix}.$$

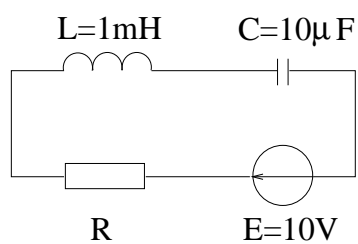
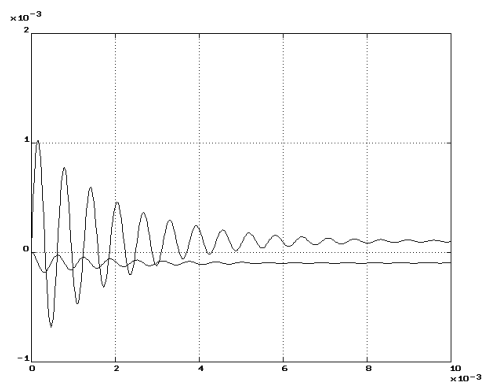
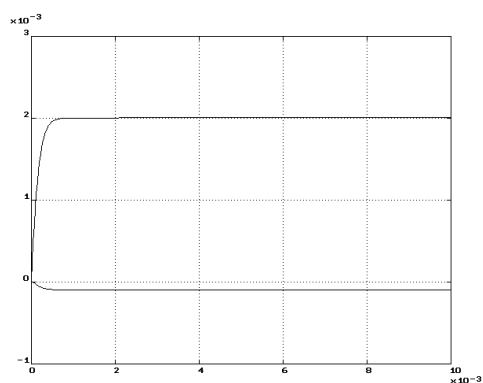
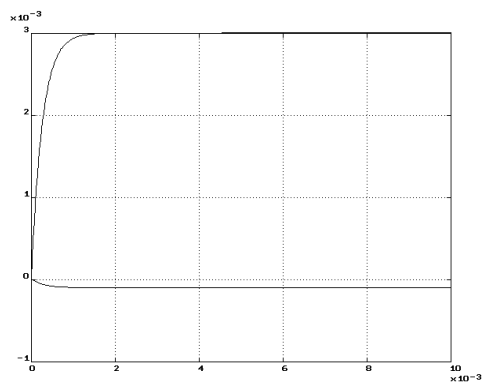


Figura 8.15: Circuit de ordinul II

Figura 8.16: Soluția circuitului din figura 8.15 pentru $R = 1\Omega$ Figura 8.17: Soluția circuitului din figura 8.15 pentru $R = 20\Omega$ Figura 8.18: Soluția circuitului din figura 8.15 pentru $R = 30\Omega$

8.3.3 Controlul automat al mărimii pasului de integrare

Criteriile prezentate privind alegerea pasului de integrare pot fi aplicate în analiza circuitelor electrice simple. În cazul circuitelor complexe, parametrice sau neliniare, constantele de timp nu sunt cunoscute aprioric sau nu pot fi definite. În aceste cazuri prezintă interes tehnicile de **alegere automată a pasului** în timpul integrării. De obicei se urmărește creșterea pasului de integrare până la limita la care erorile de trunchiere sunt încă admisibile.

O metodă relativ simplă pentru alegerea pasului și pentru controlul erorii de integrare constă în dublarea (sau înjumătățirea) pasului de timp h , astfel încât eroarea de integrare să fie cuprinsă permanent între două limite impuse $\varepsilon_m < \varepsilon < \varepsilon_M$. Pentru estimarea erorii de integrare ε se aplică relațiile (8.37) pentru doi pași succesivi:

$$\begin{aligned}\tilde{\mathbf{x}}_{k+1} &= R(\mathbf{x}_k, t_k, h), \\ \tilde{\tilde{\mathbf{x}}}_{k+2} &= R(\mathbf{x}_{k+1}, t_{k+1}, h),\end{aligned}$$

și apoi pentru un pas de lungime dublă:

$$\tilde{\tilde{\mathbf{x}}}_{k+2} = R(\mathbf{x}_k, t_k, 2h).$$

Dacă erorile de integrare sunt neglijabile, valorile $\tilde{\mathbf{x}}_{k+2}$ și $\tilde{\tilde{\mathbf{x}}}_{k+2}$ astfel calculate trebuie să fie egale. Diferența dintre aceste valori poate fi considerată ca o măsură a erorii de integrare:

$$\varepsilon = \|\tilde{\mathbf{x}}_{k+2} - \tilde{\tilde{\mathbf{x}}}_{k+2}\|. \quad (8.41)$$

Dacă $\varepsilon > \varepsilon_M$ atunci pasul de timp h se înjumătățește iar dacă $\varepsilon < \varepsilon_m$ atunci pasul de timp se dublează.

Această metodă necesită un efort de calcul sporit, la fiecare pas de timp fiind necesare în medie 5.5 evaluări ale funcției $f(x, t)$. În implementarea algoritmului trebuie acordată o atenție deosebită modului în care se calculează norma erorii. Fiecare componentă trebuie ponderată în mod corespunzător, de exemplu dacă o componentă reprezintă curentul printr-o bobină și alta sarcina unui condensator, atunci acestea trebuie raportate la valoarea tipică a curentului din bobină și respectiv la sarcina tipică din condensator. În acest fel eroarea obținută are un caracter relativ. Eroarea maximă ε_M recomandată este de circa 10^{-3} iar eroarea minimă ε_m nu trebuie să fie mai mică decât eroarea de rotunjire (se recomandă valori $\varepsilon_m = 10^{-5}$, dacă se lucrează în simplă precizie).

O îmbunătățire a erorii de calcul se poate obține dacă se adoptă pentru soluția numerică valoarea calculată prin extrapolare Richardson, pornind de la $\tilde{\mathbf{x}}_{k+2} = \mathbf{r}(t_{k+2}, h)$ și $\tilde{\tilde{\mathbf{x}}}_{k+2} = \mathbf{r}(t_{k+2}, 2h)$. Extrapolând funcția $\mathbf{r}(t_{k+2}, y)$ pentru $y = 0$ se obține:

$$\mathbf{x}_{k+2} = r(t_{k+2}, 0) = 2\tilde{\mathbf{x}}_{k+2} - \tilde{\tilde{\mathbf{x}}}_{k+2}.$$

Procedeul de extrapolare Richardson poate fi generalizat cu bune rezultate la ordine mai mari, folosind simultan mai multe rețele de discretizare cu pașii $h, h/2, h/4, h/6, h/8...$

[26]. În acest scop, se consideră că soluția numerică admite, pentru valori mici ale mărimii pasului h , o aproximare polinomială:

$$\mathbf{x}_k^{(h)} = \mathbf{r}(t_k, h) = \mathbf{y}_0 + \mathbf{y}_1 h + \mathbf{y}_2 h^2 + \dots$$

Este de așteptat ca valoarea acestei aproximări pentru $h = 0$ să fie mai aproape de soluția exactă decât pentru $h \neq 0$. Coeficienții extrapolării, din care interesează mai ales y_0 , se determină cu relațiile Aitken- Lagrange folosind rețele din ce în ce mai fine, până când abaterea dintre două soluții scade sub toleranța impusă.

Utilizând extrapolarea rațională în locul celei polinomiale, Bulirsch și Stoer au obținut una din cele mai eficiente metode de mare acuratețe pentru rezolvarea numerică a ecuațiilor diferențiale [2], [26].

Pentru a asigura permanent o valoare optimă a pasului de integrare Gear [26] propune o metodă bazată pe o singură eroare impusă ε_M . Valoarea pasului de timp h_k se modifică la fiecare etapă de integrare conform relației

$$h_{k+1} = 0,99h_k \left(\frac{15\varepsilon_M \|x_k\|}{\varepsilon} \right)^{\frac{1}{5}},$$

care asigură o eroare de trunchiere cât mai apropiată de ε_M . Dacă $\varepsilon/\|x_k\|$ depășește toleranța $15\varepsilon_M$ atunci calculul se reia cu noul pas de timp. Valoarea adoptată pentru soluția numerică este o medie ponderată a soluțiilor obținute cu cei doi pași de timp:

$$\mathbf{x}_{k+2} = (16\tilde{\mathbf{x}}_{k+2} - \tilde{\tilde{\mathbf{x}}}_{k+2})/15.$$

O metodă ingenioasă de modificare a pasului este propusă de Zonnenveld [2]. Acesta adoptă ca măsură a erorii norma ultimului termen din seria Taylor trunchiată, respectiv:

$$q_k = \left\| \frac{h_k^p}{p!} \mathbf{x}_k^{(p)} \right\|. \quad (8.42)$$

Se urmărește calculul acestui termen în funcție de evaluările deja efectuate ale funcției $\mathbf{f}(\mathbf{x}, t)$. În cazul metodei Runge-Kutta de ordinul doi (8.29) se determină coeficienții d_1 și d_2 astfel încât:

$$\frac{h^2}{2!} \mathbf{x}_k^{(2)} = h[d_1 \mathbf{g}_1 + d_2 \mathbf{g}_2]. \quad (8.43)$$

Utilizând relațiile (8.30), (8.31) rezultă:

$$\frac{h^2}{2} \left[\frac{\partial \mathbf{f}}{\partial x} \mathbf{g}_1 + \frac{\partial \mathbf{f}}{\partial t} \right] = h \mathbf{g}_1 (d_1 + d_2) + h^2 \frac{\partial \mathbf{f}}{\partial x} \mathbf{g}_1 b_1 d_2 + \frac{\partial \mathbf{f}}{\partial t} b_2 d_2,$$

ceea ce impune constantelor d_1 și d_2 relațiile:

$$d_1 + d_2 = 0 ; d_2 b_1 = 1/2 ; d_2 b_2 = 1/2. \quad (8.44)$$

Alegând, de exemplu, varianta Heun a metodei Runge-Kutta, rezultă $d_1 = -1/2$, $d_2 = 1/2$, deci eroarea de la pasul k poate fi măsurată prin:

$$q_k = \frac{h_k}{2} \|\mathbf{f}(\mathbf{x}_k, t_k) - \mathbf{f}(\mathbf{x}_k + h\mathbf{f}(\mathbf{x}_k, t_k), t_k + h)\|. \quad (8.45)$$

Folosind o tehnică asemănătoare se obține norma ultimului termen și în cazul metodei Runge-Kutta de ordinul patru (8.37):

$$q_k = \left\| \frac{h_k^4}{4!} x_k^{(4)} \right\| = h_k \left\| -\frac{2}{3}g_1 + 2g_2 + 2g_3 + 2g_4 - \frac{16}{3}g_5 \right\|, \quad (8.46)$$

cu

$$q_5 = f \left(x_k + \frac{h}{32}(5g_1 + 7g_2 + 13g_3 - g_4), t_k + \frac{3}{4}h \right).$$

Se constată că în cazul metodei de ordinul patru este necesară o evaluare suplimentară a funcției $\mathbf{f}(\mathbf{x}, t)$. Eroarea q_k de la pasul curent este comparată cu:

$$\tau_k = h_k[\varepsilon_1 \|f(x_k, t_k)\| + \varepsilon_2], \quad (8.47)$$

unde ε_1 și ε_2 sunt două valori limită date, impuse erorii.

Dacă $q_k \leq \tau_k$ atunci valoarea (x_{k+1}) calculată la pasul curent este acceptată ca fiind corectă și se trece la pasul următor h_{k+1} , care se majorează față de pasul curent. Dacă eroarea q_k depășește toleranța admisă ($q_k > \tau_k$), atunci pasul curent h_k se micșorează și se recalculează valoarea x_{k+1} cu noul pas.

Pentru coeficientul de corecție al pasului, Zonnenveld propune valoarea:

$$\mu_k = \frac{\tau_k}{\tau_k + q_k} + 0.45. \quad (8.48)$$

Dacă rezultatul este eronat ($q_k < \tau_k$) atunci pasul este corectat la valoarea $\mu_k h_k$ (cu $0.45 < \mu_k < 0.95$). Dacă rezultatul este acceptat, atunci coeficientul de corecție satisface inegalitățile $0.95 \leq \mu_k \leq 1.45$. Pentru a evita o mărire hazardată a pasului, Zonnenveld propune strategia dată de relația:

$$h_{k+1} = \begin{cases} \mu_k h_k, & \text{pentru } k = 0; \\ h_k \left[\frac{h_k}{h_{k-1}}(1 + \mu_k) - \mu_{k-1} \right], & \text{pentru } k > 0. \end{cases} \quad (8.49)$$

O variantă a metodei Runge-Kutta de ordinul patru, optimizată din punctul de vedere al efortului de calcul (și al numărului de registre procesor utilizate) se datorează lui Gill, care alege în acest sens în (8.36), $b_6 = 1 + 1/\sqrt{2}$ și $c_1 = 1/2$. În varianta sa finală,

algoritmul lui Gill are forma [48], [26]:

$$\begin{aligned}
g_1 &= hf(x_k, t_k), \\
q_1 &= q_0 + 3(g_1 - 2q_0)/2 - g_1/2, \\
y_1 &= x_k + (g_1 - 2q_0)/2, \\
g_2 &= hf(y_1, t_k + h/2), \\
q_2 &= q_1 + 3(1 - 1/\sqrt{2})(q_2 - q_1) - (1 - 1/\sqrt{2})q_2, \\
y_2 &= y_1 + (1 - 1/\sqrt{2})(q_2 - q_1), \\
g_3 &= hf(y_2, t_k + h/2), \\
q_3 &= q_2 + 3(1 + 1/\sqrt{2})(q_3 - q_2) - (1 + 1/\sqrt{2})q_3, \\
y_3 &= y_2 + (1 + 1/\sqrt{2})(q_3 - q_2), \\
g_4 &= hf(y_3, t + h), \\
q_4 &= q_3 + 3(q_4 - 2q_3)/6 - q_4/2, \\
x_{k+1} &= y_4 = y_3 + (q_4 - 2q_3)/6.
\end{aligned} \tag{8.50}$$

El asigură un număr minim de operații și folosirea cu eficiență maximă a memoriei. Mai mult, această metodă permite compensarea parțială a erorii de rotunjire, de la un pas la altul. În acest scop se alege inițial $q_0 = 0$ iar dacă precizia ar fi infinită ar trebui să rezulte $q_4 = 0$. În realitate acest lucru nu este adevărat și q_4 reprezintă circa triplul erorii de rotunjire de care este afectat rezultatul y_4 . Pentru a compensa această eroare, în pasul următor se inițializează q_0 cu valoarea q_4 de la pasul precedent. În implementarea algoritmului Runge-Kutta-Gill, dată în [48], controlul erorii și al pasului se realizează cu ajutorul relației (8.41), limita inferioară a erorii fiind aleasă $\varepsilon_m = \varepsilon_M/50$.

8.4 Metode de integrare multipas

Metodele Taylor și Runge-Kutta fac parte din categoria metodelor de integrare a ecuațiilor diferențiale cu un pas, deoarece permit estimarea soluției x_k , la pasul de timp k , folosind doar valoarea x_{k-1} de la pasul anterior. Trebuie observat că pe parcursul integrării, sunt deja cunoscute valorile soluției într-o serie de pași anteriori x_{k-1}, x_{k-2}, \dots , valori care la metodele cu un pas nu sunt luate în considerare. Utilizarea acestor valori în calculul soluției de la pasul curent poate duce la creșterea preciziei fără un efort de calcul suplimentar. Metodele care în determinarea soluției folosesc informațiile de la mai mulți pași, anteriori pasului curent se numesc *metode multipas*. Aceste metode nu au autostart, deoarece la primul pas este cunoscută doar condiția inițială x_0 . Din acest motiv, integrarea trebuie demarată cu o metodă cu un pas, de regulă cu Runge-Kutta care este utilizată la integrarea ecuației pe un număr de pași, și apoi se continuă integrarea folosind metode multipas.

Se consideră o rețea uniformă de discretizare a variabilei independente:

$$t_0, t_1 = t_0 + h, t_2 = t_0 + 2h, \dots, t_k = t_0 + kh, \dots$$

și se presupun cunoscute valorile soluției numerice în primele noduri ale rețelei:

$$x_0, x_1, x_2, \dots, x_{k-1},$$

precum și valorile derivatelor

$$x'_0 = f(x_0, t_0), x'_1 = f(x_1, t_1), \dots, x'_{k-1} = f(x_{k-1}, t_{k-1}).$$

Pentru obținerea relației de recurență specifică metodelor implicite se integrează derivata soluției pe un număr de n pași succesivi:

$$\hat{x}(t_k) - \hat{x}(t_{k-n}) = \int_{t_{k-n}}^{t_k} \hat{x}'(t) dt. \quad (8.51)$$

Folosind o metodă de cuadratură numerică, integrala va fi aproximată printr-o combinație liniară de valori ale integrandului de pe ultimele p noduri ale rețelei de discretizare:

$$x_k - x_{k-n} = h \sum_{i=1}^p b_i x'_{k-i},$$

ceea ce generează relația de recurență:

$$x_k = x_{k-n} + h[b_1 f(x_{k-1}, t_{k-1}) + \dots + b_p f(x_{k-p}, t_{k-p})]. \quad (8.52)$$

Relația (8.52) poate fi generalizată sub forma:

$$x_k = \sum_{i=1}^p (a_i x_{k-i} + b_i h x'_{k-i}), \quad (8.53)$$

în care trebuie determinați coeficienții $a_1, a_2, \dots, b_1, b_2, \dots$. Acești coeficienți, în număr de $2p$, ponderează valorile soluției și respectiv ale derivatei acesteia pe ultimii p pași de integrare. Relația (8.53) pune în evidență caracterul **explicit** al acestei clase de metode. Pentru metodele multipas, parametrul p din (8.53) indică numărul de pași ai metodei, dar în particular dacă se alege $p = 1$ se obține pentru $a_1 = b_1 = 1$ metoda cu un pas, de tip Euler explicit.

Dacă în aproximarea integralei (8.51) se folosește inclusiv valoarea integrandului din punctul final t_k , se obține:

$$x_k - x_{k-n} = h \sum_{i=0}^p b_i x'_{k-i},$$

relație care poate fi scrisă în general, sub forma implicită:

$$\sum_{i=0}^p (a_i x_{k-i} + b_i h x'_{k-i}) = 0. \quad (8.54)$$

Relația (8.54) generalizează relația (8.53), care se poate obține prin alegerea $a_0 = -1, b_0 = 0$. Spre deosebire de (8.53), relația (8.54) conține un singur termen suplimentar $x'_k = f(x_k, t_k)$.

Dacă se normalizează (8.54), astfel încât $a_0 = -1$, rezultă

$$x_k = b_0 h f(x_k, t_k) + \sum_{i=1}^p [a_i x_{k-i} + b_i h f(x_{k-i}, t_{k-i})]. \quad (8.55)$$

Se arată că (8.55) evidențiază caracterul implicit al clasei de metode obținute. Soluția numerică x_k de la pasul k intervine atât în primul membru cât și sub funcția $f(x, t)$ din membrul doi. Dacă $f(x, t)$ este o funcție neliniară, atunci determinarea soluției x_k presupune rezolvarea unui sistem de ecuații neliniare (8.55). Aplicarea metodelor implicite necesită deci un efort de calcul sporit față de cele explicite.

8.4.1 Metoda explicită Adams-Bashforth

Metoda Adams-Bashforth se bazează pe relația de recurență explicită (8.52), în care se alege $n = 1$, ceea ce corespunde alegerii $a_1 = 1, a_2 = 0, a_3 = 0 \dots$ în relația (8.53). Coeficienții b_i se pot determina prin calculul integralei (8.51) cu metoda de extrapolare polinomială de tip Newton, bazată pe diferențe regresive. În continuare, acești coeficienți vor fi determinați, pentru cazul $p = 2$ (cu doi pași), totuși cu metoda seriei Taylor trunchiate, pentru a evidenția și ordinul erorii de trunchiere. Alegând în (8.53) $p = 2$ și $a_2 = 0$, rezultă:

$$x_k = a_i x_{k-1} + h b_1 x'_{k-1} + h b_2 x'_{k-2}, \quad (8.56)$$

iar pe baza dezvoltărilor Taylor:

$$\begin{aligned} \hat{x}(t_k) &= a_1 \left[\hat{x}(t_k) - h \hat{x}'(t_k) + \frac{h^2}{2} \hat{x}''(t_k) + O(h^3) \right] + \\ &+ h b_1 \left[\hat{x}'(t_k) - h \hat{x}''(t_k) + O(h^2) \right] + \\ &+ h b_2 \left[\hat{x}'(t_k) - 2h \hat{x}''(t_k) + O(h^3) \right]. \end{aligned} \quad (8.57)$$

Identificând termenii în h_0, h_1, h_2 , rezultă sistemul:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 1 \\ \frac{1}{2} & -1 & -2 \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ b_2 \end{bmatrix}, \quad (8.58)$$

care are soluția $a_1 = 1, b_1 = \frac{3}{2}, b_2 = \frac{-1}{2}$, ceea ce corespunde relației de integrare numerică Adams-Bashforth cu doi pași:

Tabela 8.2: Coeficienții metodei Adams-Bashforth

p	i	1	2	3	4	5	6
1	b_i	1					
2	$2b_i$	3	-1				
3	$12b_i$	23	-16	5			
4	$24b_i$	55	-59	37	-9		
5	$720b_i$	1901	-2774	2616	-1274	251	
6	$1440b_i$	4277	-7293	9982	-7298	2877	-475

$$x_{k+1} = x_k + \frac{h}{2}[3x'_k - x'_{k-1}]. \quad (8.59)$$

Relația (8.57) evidențiază eroarea locală de trunchiere a metodei cu doi pași, care este de tipul $O(h^3)$. Relația (8.56) trebuie să fie satisfăcută identic de un polinom de gradul doi, deci de funcțiile $x = 1, s, s^2$ cu $s = (t - t_k)/h$. Aceasta este echivalentă cu condiția ca primii trei termeni ai seriei Taylor să se anuleze, deci condițiile de interpolare polinomială vor conduce la aceleași valori (8.59) ale coeficienților.

În general se consideră funcția polinomială:

$$\hat{x}(t) = s^n, \hat{x}'(t) = \frac{ns^{n-1}}{h},$$

unde $s = (t - t_{k-1})/h$ și $n = 1, 2, \dots, p$, care se substituie în $x_k = x_{k-1} + h \sum_{i=1}^p b_i x'_{k-i}$ și se obține:

$$1 = n \sum_{i=1}^p b_i s^{n-1} \Big|_{t=t_k-h} = n \sum_{i=1}^p (1-i)^{n-1} b_i. \quad (8.60)$$

Coeficienții b_i satisfac deci sistemul de ecuații:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 0 & -1 & -2 & -3 & \dots & -(p-1) \\ 0 & 1 & 4 & 9 & \dots & (p-1)^2 \\ 0 & -1 & -8 & -27 & \dots & -(p-1)^3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & (-1)^{p-1} & (-2)^{p-1} & (-3)^{p-1} & \dots & (1-p)^{p-1} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ \dots \\ b_p \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ 1/3 \\ 1/4 \\ \dots \\ 1/p \end{bmatrix}, \quad (8.61)$$

care pentru diferite ordine are soluțiile din tabelul 8.2:

Se poate afirma că la metoda Adams-Bashforth cu p pași eroarea locală de trunchiere satisface inegalitatea:

$$\varepsilon = \|x(t_k) - x_k\| \leq C_p h^{p+1}, \quad (8.62)$$

deci $\varepsilon = O(h^{p+1})$ [9],[4], având ordinul $p + 1$, urmând ca eroarea globală să aibă ordinul egal cu numărul de pași.

Se constată că pentru a obține o comportare a erorii de trunchiere comparabilă cu cea dată de metoda Runge-Kutta de ordinul patru, trebuie aplicată metoda Adams-Bashforth cu patru pași:

$$x_{k+1} = x_k + \frac{h}{24}[55f(x_k, t_k) - 59x'_{k-1} + 37x'_{k-2} - 9x'_{k-3}], \quad (8.63)$$

ceea ce presupune o singură evaluare a funcției $f(x, t)$ în punctul (x_k, t_k) , urmând ca celorlalte trei valori $x'_{k-i} = f(x_{k-i}, t_{k-i}), i = 1, 2, 3$ să fi fost memorate la pașii anteriori. În consecință, efortul de calcul este de circa patru ori mai mic decât la metoda Runge-Kutta. Pentru a putea demara, algoritmul generat de (8.63) are nevoie de o inițializare, care presupune aplicarea metodei Runge-Kutta pe primii trei pași: $t_1 = t_0 + h, t_2 = t_0 + 2h, t_3 = t_0 + 3h$, în vederea calcului valorilor $x'_{k-3} = f(x_0, t_0), x'_{k-2} = f(x_1, t_1), x'_{k-1} = f(x_2, t_2)$ și $x_k = f(x_3, t_3)$. pentru $k = 3$.

8.4.2 Metoda implicită Adams-Moulton

Metoda Adams-Moulton, fiind implicită, se bazează pe relația (8.55), în care ca și în cazul anterior se alege $a_1 = 1$ și $a_2 = a_3 = \dots = a_p = 0$, ceea ce corespunde alegerii $n = 1$, specifice metodelor de tip Adams:

$$x_k = x_{k-1} + h \sum_{i=1}^p b_i x'_{k-i}. \quad (8.64)$$

Dacă relația (8.64) este satisfăcută identic de un polinom arbitrar de gradul $(p + 1)$, atunci eroarea locală de trunchiere are un ordin cu o unitate mai mare decât gradul polinomului $O(h^{p+2})$. După cum s-a arătat în paragraful anterior, această condiție este echivalentă cu anularea primilor $(p + 2)$ termeni din seria Taylor asociată soluției exacte. Dacă se substituie în (8.64) polinoamele:

$$\hat{x}(t) = s^n, \text{ cu } n = 1, 2, \dots, (p + 1),$$

se obține sistemul de ecuații satisfăcut de b_i :

$$\sum_{i=0}^p (1-i)^{n-1} b_i = 1/n, n = 1, 2, \dots, (p + 1)$$

sau dezvoltat:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & -1 & -2 & \dots & -(p-1) \\ 1 & 0 & 1 & 4 & \dots & (1-p)^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & (-1)^{p-1} & (-2)^{p-1} & \dots & (1-p)^{p-1} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ 1/3 \\ \vdots \\ 1/(p+1) \end{bmatrix} \quad (8.65)$$

Tabela 8.3: Coeficienții metodei Adams-Moulton

p	i=	0	1	2	3	4	5
0	b_i	1					
1	$2b_i$	1	1				
2	$12b_i$	5	8	-1			
3	$24b_i$	9	19	-5	1		
4	$720b_i$	251	646	-264	106	-19	
5	$1440b_i$	475	1427	-798	482	-173	27

Particularizând sistemul (8.65) pentru cazul $p = 1$:

$$\begin{cases} b_0 + b_1 = 1; \\ b_0 = 1/2; \end{cases}$$

se obțin valorile coeficienților $b_0 = b_1 = 1/2$. Relația de integrare Adams-Moulton cu un pas:

$$x_{k+1} = x_k + \frac{h}{2}[f(x_k, t_k) + f(x_{k+1}, t_{k+1})] \quad (8.66)$$

este echivalentă cu metoda trapezelor aplicată integralei (8.51). Această variantă a metodei asigură o eroare locală de trunchiere de tipul $O(h^3)$, deci o comportare mai bună a erorii decât la metoda Euler implicită, care se obține formal pentru $p = 0$ și care necesită practic același efort de calcul. Metoda trapezelor fiind cu un singur pas are autostartul asigurat, dar necesită rezolvarea la fiecare pas de timp a ecuației (8.66), în vederea determinării soluției x_{k+1} . Dacă se utilizează rezolvarea iterativă, atunci se recomandă ca inițializarea să se facă cu valoarea dată de metoda Euler explicită. Se constată în acest caz că prima iterație are valoarea corespunzătoare metodei Runge-Kutta de ordinul doi în varianta Heun.

În cazul metodei implicite cu doi pași ($p = 2$) se obține $b_0 = 5/12, b_1 = 8/12, b_2 = -1/12$, ceea ce corespunde la:

$$x_{k+1} = x_k + \frac{h}{12}[5f(x_{k+1}, t_{k+1}) + 8f(x_k, t_k) - f(x_{k-1}, t_{k-1})], \quad (8.67)$$

relație pe doi pași care asigură o eroare locală de trunchiere de tipul $O(h^4)$. Tabelul 8.3 conține coeficienții metodei Adams-Moulton până la variantele cu cinci pași.

În metoda Adams-Moulton eroarea locală de trunchiere satisface inegalitatea:

$$\varepsilon = \|\hat{x}(t_k) - x_k\| \leq C_p h^{p+2}, \quad (8.68)$$

deci ordinul metodei implicite este cu o unitate mai mare decât cel al metodei explicite cu același număr de pași. Pentru a obține o comportare a erorii de trunchiere de același tip cu metoda Runge-Kutta de ordinul patru este suficientă aplicarea metodei implicite cu trei pași:

$$x_{k+1} = x_k + \frac{h}{24}[9f(x_{k+1}, t_{k+1}) + 19x'_k - 5x'_{k-1} + x'_{k-2}]. \quad (8.69)$$

Efortul de calcul în metoda Adams-Moulton este mult mai mare decât cel din metoda Runge-Kutta, deoarece presupune rezolvarea ecuației (8.69) în vederea determinării soluției numerice x_{k+1} . Pentru demararea integrării prin relația (8.69) este necesară aplicarea pe primii doi pași de timp $t_0 + h, t_1 + h$ a metodei Runge-Kutta în vederea determinării funcțiilor $x'_{k-2} = f(x_0, t_0)$, $x'_{k-1} = f(x_1, t_1)$ și $x'_k = f(x_2, t_2)$.

O metodă eficientă pentru rezolvarea ecuației implicite constă în calculul iterativ al soluției pe baza relației (8.69), pornind de la o anumită inițializare pentru x_{k+1} . Succesul acestei metode depinde de inițializare. Dacă pentru inițializare se folosește valoarea dată de o metodă explicită, atunci numărul iterațiilor necesare se poate reduce la una sau cel mult două. Algoritm bazat pe folosirea combinată a metodelor explicite și implicite este cunoscut sub numele “**predictor-corector**”. Din acest motiv metodele implicite se numesc metode de corecție. Dacă se aplică metoda predictor-corector cu un predictor de tip Adams-Bashforth pe trei pași:

$$x_{(k+1)p} = x_k + \frac{h}{12}[23f(x_k, t_k) - 16x'_{k-1} + 5x'_{k-2}] \quad (8.70)$$

și un corector de tip Adams-Moulton pe același număr de pași:

$$x_{k+1} = x_k + \frac{h}{24}[9f(x_{(k+1)p}, t_{k+1}) + 19x'_k - 5x'_{k-1} + x'_{k-2}] \quad (8.71)$$

se constată că este necesară evaluarea doar de două ori a funcției $f(x, t)$, o dată pentru predictor și a doua oară pentru corector (sau cel mult de trei ori dacă este necesară o reiterare), față de patru ori cât era necesar în metoda Runge-Kutta, pentru asigurarea unei erori comparabile. Mai mult, diferența dintre valoarea prezisă și cea corectată oferă o măsură sigură a erorii de trunchiere:

$$\varepsilon = \|x_{(k+1)p} - x_{k+1}\| \quad (8.72)$$

În consecință se poate afirma că metoda predictor-corector este mai avantajoasă din punct de vedere al efortului de calcul și al controlului erorii decât metoda Runge-Kutta.

Tabela 8.4: Coeficienții metodei Milne

p	i=	0	1	2	3	4	5
1	b_i	0	2				
3	$3b_i$	0	8	-4	8		
5	$10b_i$	0	33	-42	78	-42	11
2	$3b_i$	1	4	1			
4	$45b_i$	14	64	24	64	14	

8.4.3 Metoda Milne

Metoda Milne uzuală este o metodă implicită cu doi pași în care se alege $n = 2$, sau echivalent în ecuația (8.55), $a_1 = 0, a_2 = 1$:

$$x_k = x_{k-2} + h[b_0x'_k + b_1x'_{k-1} + b_2x'_{k-2}]. \quad (8.73)$$

Pentru determinarea celor trei coeficienți b_0, b_1, b_2 se impune condiția ca relația (8.73) să fie satisfăcută identic de un polinom arbitrar de gradul trei. Dacă se alege:

$$\hat{x}(t) = s^n$$

cu $s = (t - t_{k-2})/2h$ și $n = 1, 2, 3$, rezultă prin substituție în (8.73):

$$\begin{aligned} b_0 + b_1 + b_2 &= 2 \\ b_0 + b_1/2 &= 1 \\ b_0 + b_1/4 &= 2/3 \end{aligned}$$

sistem care admite soluția $b_0 = 1/3, b_1 = 4/3$, și $b_2 = 1/3$. În consecință metoda Milne se bazează pe relația recursivă:

$$x_{k+1} = x_{k-1} + \frac{h}{3}[f(x_{k+1}, t_{k+1}) + 4x'_k + x'_{k-1}] \quad (8.74)$$

și asigură o comportare a erorii de trunchiere de tipul $O(h^4)$. Metodele de tip Milne au pentru coeficienții din relația (8.55) valorile $a_i = 0$ cu $i \neq p$ și $a_p = 1$. Tabelul 8.4 conține valorile coeficienților b_i pentru diferite ordine. În cazul metodelor explicite ordinele impare $2p' - 1$ oferă aceeași eroare ca cele pare $2p'$, deci sunt preferabile. În cazul implicit ordinele pare $2p'$ oferă aceeași eroare ca cele de ordin $2p' + 1$, deci sunt preferate. [7]

Încercarea de a considera $a_2 \neq 0$ în metoda explicită (8.53) determină variante instabile numeric ale metodelor multipas. De exemplu, dacă se presupune $p = 2$ în (8.53) se obține:

$$x_k = a_1x_{k-1} + a_2x_{k-2} + b_1hx'_{k-1} + b_2hx'_{k-2}. \quad (8.75)$$

Tabela 8.5: Soluția ecuației de test

pasul k	0	1	2	3	4	5	6
soluția x_k	0	ε	-4ε	21ε	-104ε	521ε	-2604ε

Dacă se impune ca relația (8.75) să fie verificată identic de un polinom de gradul trei se obțin următoarele valori ale coeficienților:

$$a_1 = -4, \quad a_2 = 5, \quad b_1 = 4, \quad b_2 = 2.$$

și respectiv o metodă explicită cu doi pași:

$$x_k = -4x_{k-1} + 5x_{k-2} + h[4x'_{k-1} + 2x'_{k-2}], \quad (8.76)$$

cu o comportare a erorii de tip $O(h^4)$ dar care prezintă puternice instabilități numerice. De exemplu, în rezolvarea ecuației diferențiale

$$\frac{dx}{dt} = 0$$

cu condiția inițială nulă $x(0) = 0$, a cărei soluție exactă este $\hat{x}(t) = 0$, pentru o inițializare $x_1 = \varepsilon$ (o mică eroare de rotunjire), soluția numerică se amplifică conform tabelului 8.5, în loc să rămână constantă.

8.4.4 Consistența, stabilitatea și convergența metodelor multiple

Forma generală a relației de recurență specifică metodelor multiple este dată de (8.54). Prin particularizarea acestei relații se obțin atât metodele explicite ($b_0 = 0$) cât și cele implicite ($b_0 \neq 0$). Metodele de tip Adams reprezintă doar o parte din metodele multiple, care pot fi obținute prin particularizarea relației (8.54). Coeficienții acestora au fost determinați impunând condiția ca (8.54) să fie satisfăcută identic de polinoame arbitrare. Dacă se aplică această tehnică pentru un polinom de grad n , se pot determina un număr de coeficienți egal cu numărul coeficienților din polinom, deci $n + 1$. Relația (8.54) conține $2(p + 1)$ coeficienți $a_0, a_1, \dots, a_p, b_0, b_1, \dots, b_p$ din care unul poate fi ales unitar prin normalizarea ecuației. În consecință gradul polinomului folosit ca funcție test trebuie să satisfacă inegalitatea:

$$2p + 1 \geq n + 1, \quad (8.77)$$

diferența dintre cei doi membri reprezentând numărul coeficienților care se pot alege arbitrar. Gradul n al polinomului de test impune ordinul erorii de trunchiere, care în forma locală este de tipul $O(h^{n+1})$ iar în forma sa globală este de tipul $O(h^n)$, motiv pentru care el se numește și *ordinul metodei*.

Presupunem pentru început că soluția ecuației diferențiale este o constantă $\hat{x}(t) = 1$, ceea ce corespunde alegerii unui polinom de grad $j = 0$: $x_{k-i} = 1, x'_{k-i} = 0$. Înlocuind aceste valori în (8.54), rezultă:

$$\sum_{i=0}^p a_i = 0, \quad (8.78)$$

care este prima condiție impusă coeficienților metodei generale multipas. Dacă se alege ca polinom de test polinomul de gradul $j = 1$, de exemplu $\hat{x}(t) = t$, atunci prin substituția valorilor $x_n = t_n$ și $x'_n = 1$, în (8.54), rezultă pentru $t_{k-i} = (1-i)h$:

$$\sum_{i=0}^p (1-i)a_i + \sum_{i=0}^p b_i = 0.$$

Pentru gradul doi $\hat{x}(t) = t^2$, $x_n = t_n^2$, $x'_n = 2t_n$ rezultă:

$$\sum_{i=0}^p (1-i)^2 a_i + 2 \sum_{i=0}^p (1-i)b_i = 0,$$

iar în general coeficienții satisfac relațiile:

$$\sum_{i=0}^p (1-i)^j a_i + j \sum_{i=0}^p (1-i)^{j-1} b_i = 0, \quad (8.79)$$

pentru $j = 1, 2, \dots, n$. Dacă o metodă multipas îndeplinește condițiile (8.78) și (8.79) atunci aceasta garantează că metoda permite calculul exact (în măsura în care erorile de rotunjire sunt nule) cel puțin al soluțiilor care sunt polinoame în t , de grad maxim n . Aceste condiții sunt necesare pentru a asigura convergența metodei.

Definiția 8.2 *O metodă multipas, definită de relația (8.54), ai cărei coeficienți satisfac relațiile (8.78), (8.79) se numește consistentă.*

Ordinul unei metode consistente este dat de valoarea n a gradului maxim pentru care este îndeplinită relația (8.79).

Teorema 8.4 *O metodă de discretizare cu p pași:*

$$x_k = \sum_{i=0}^p a_i x_{n-i} + h \sum_{i=0}^p b_i f(x_{n-i}, t_{n-i}),$$

care este consistentă cu ordinul n are o eroare locală de trunchiere:

$$\varepsilon = C_n \hat{x}^{(n+1)}(\tau) h^{n+1} = O(h^{n+1}),$$

cu $-ph < \tau < h$ și:

$$C_n = \frac{1}{(n+1)!} \left[p^{n+1} - \sum_{i=1}^{p-1} a_i (p-i)^{n+1} - (n+1) \sum_{i=1}^{p-1} b_i (p-i)^n \right].$$

Această teoremă este demonstrată în [4]. Metodele de tip Adams-Bashforth sunt consistente și au ordinul egal cu numărul de pași pe care se aplică. Metodele Adams-Moulton sunt consistente dar au ordinul cu o unitate mai mare decât numărul de pași pe care se aplică. Metoda Milne (8.74) și metoda (8.76) sunt și ele consistente cu ordinul trei. Consistența este o condiție necesară dar nu suficientă pentru ca o metodă să fie utilă în practică. Consistența nu se referă în nici un fel la modul în care se propagă erorile de rotunjire. Metoda generată de relațiile (8.76) este consistentă dar s-a constatat că erorile de rotunjire sunt amplificate în așa fel încât după câțiva pași de integrare acoperă complet soluția.

Pentru a studia stabilitatea numerică a metodelor multipas se consideră din nou ecuația diferențială simplă:

$$\frac{dx}{dt} = \lambda x \quad (8.80)$$

căreia i se impune condiția inițială $x(0) = 1$, ca ecuație de test. Soluția exactă a acestei ecuații este $\hat{x}(t) = e^{\lambda t}$, funcție care în nodurile unei rețele regulate $t_k = kh$ ia valorile:

$$\hat{x}(t_k) = e^{\lambda t_k} = e^{\lambda kh} = (e^{\lambda h})^k = z_0^k. \quad (8.81)$$

Se constată că soluția exactă reprezintă o progresie geometrică cu relația:

$$z_0 = e^{\lambda h} = 1 + \lambda h + \frac{(\lambda h)^2}{2!} + \dots \quad (8.82)$$

În cazul limită în care $\lambda \rightarrow 0$ soluția exactă tinde către o constantă

$$\hat{x}(t) = x_0 = 1 \quad (8.83)$$

și $z_0 = 1$. Soluția numerică se obține aplicând succesiv relația generală (8.54), care se presupune consistentă. În cazul ecuației test (8.80) relația generală (8.54) capătă forma:

$$\sum_{i=0}^p (a_i + \lambda h b_i) x_{k-i} = 0, \quad (8.84)$$

care este o combinație liniară a ultimelor $(p+1)$ valori ale soluției numerice. Datorită caracterului liniar al relației (8.84), rezultă că și diferența dintre o soluție numerică x_k și perturbata ei z_k , respectiv $e_k = z_k - x_k$ va satisface o ecuație asemănătoare:

$$\sum_{i=0}^p (a_i + \lambda h b_i) e_{k-i} = 0 \quad (8.85)$$

Presupunem că o perturbație inițială e_0 se propagă în procesul de calcul după o progresie geometrică respectiv

$$e_j = z e_{j-1} \quad (8.86)$$

în care z este coeficientul cu care se amplifică eroarea la fiecare pas. Condiția ca o metodă să fie absolut stabilă constă în faptul că norma perturbației să nu se mărească de la un pas la altul, deci:

$$|z| \leq 1. \quad (8.87)$$

Substituind relația (8.86) în (8.85) rezultă:

$$P(z) = \sum_{i=0}^p (a_i + \lambda h b_i) z^{p-i} = 0. \quad (8.88)$$

Polinomul $P(z)$ definit de relația (8.88) este numit polinomul caracteristic al metodei iar studiul lui este de mare importanță în caracterizarea stabilității numerice a metodei de integrare. Polinomul caracteristic are gradul p și poate fi descompus în:

$$P(z) = A(z) + \lambda h B(z), \quad \text{cu} \quad A(z) = \sum_{i=0}^p a_i z^{p-i} \quad \text{și} \quad B(z) = \sum_{i=0}^p b_i z^{p-i} \quad (8.89)$$

Pentru început se studiază stabilitatea numerică în condițiile în care pasul h este suficient de mic. Pentru $h \rightarrow 0$, polinomul $P(z)$ se reduce la $A(z)$, numit polinom de stabilitate. Din punctul de vedere al ecuației (8.80), condiția $\lambda h \rightarrow 0$ este echivalentă cu degenerarea $\lambda = 0$ careia îi corespunde soluția constantă (8.83). Pentru ca ecuația (8.84) să admită în acest caz soluția constantă este necesar ca:

$$\sum_{i=1}^p a_i = 0,$$

ceea ce este adevărat, deoarece reprezintă prima condiție de consistență (8.78). Această condiție impune ca polinomul de stabilitate $A(z)$ să aibă rădăcina $z_1 = 1$. Această rădăcină a polinomului caracteristic, pentru $\lambda h = 0$ se numește **rădăcina principală** și ea asigură corectitudinea soluției. Celelalte rădăcini ale polinomului caracteristic se numesc **rădăcini parazite** și ele caracterizează propagarea erorilor de calcul. Condiția necesară pentru ca o metodă multipas să fie numeric stabilă la un pas h suficient de mic este ca toate rădăcinile parazite z_2, z_3, \dots, z_{p-1} ale polinomului de stabilitate $A(z)$

să aibă modulul mai mic decât cel al rădăcinii principale $z_1 = 1$, deci să fie cuprinse în cercul unitate:

$$|z_j| < 1 \quad (8.90)$$

Se admite și cazul $|z_j| = 1$ dar cu condiția să nu existe rădăcini multiple pe cercul unitate (toate rădăcinile de modul unitar trebuie să fie simple). Această condiție este justificată de inegalitatea (8.87). Pentru un studiu mai atent se va considera metoda consistentă cu doi pași:

$$(a_0 + \lambda hb_0)x_k + (a_1 + \lambda hb_1)x_{k-1} + (a_2 + \lambda hb_2)x_{k-2} = 0 \quad (8.91)$$

la care polinomul caracteristic este de gradul doi:

$$P(z) = A(z) + \lambda hB(z); \quad A(z) = a_0z^2 + a_1z + a_2; \quad B(z) = b_0z^2 + b_1z + b_2. \quad (8.92)$$

Polinomul de stabilitate $A(z)$ are rădăcina $z_1 = 1$ și o alta, z_2 , pe care o presupunem diferită de z_1 . Dacă se integrează ecuația (8.80) cu $\lambda = 0$ soluția exactă este constantă (8.83). Ecuația cu diferențe finite (8.91) admite o soluție generală, care este de forma unei combinații liniare de serii geometrice, cu rații date de soluțiile polinomului caracteristic:

$$x_k = c_1z_1^k + c_2z_2^k. \quad (8.93)$$

Condiția inițială exactă $x_0 = 1, x_1 = 1$ impune valorile:

$$c_1 = 1, \quad c_2 = 0, \quad (8.94)$$

cea ce ar corespunde unei soluții numerice egale cu cea exactă. Datorită erorii de rotunjire, este posibil să se obțină:

$$c_1 = 1 \text{ și } c_2 = \varepsilon \neq 0, \quad (8.95)$$

caz în care soluția numerică conține o componentă parazită a cărei valoare inițială este ε . Dacă modulul rădăcinii parazite $|z_2|$ este subunitar, această componentă nu depășește perturbația inițială ε și se anulează în timp. Dacă în schimb $|z_2| > 1$, atunci componenta parazită crește exponențial în timp și chiar dacă a plecat de la o valoare inițială neglijabilă, poate ajunge să depășească componenta exactă a soluției. Dacă rădăcina principală este dublă $z_1 = z_2$, atunci în locul relației (8.93) trebuie folosită relația:

$$x_k = c_1z_1^k + c_2kz_1^k, \quad (8.96)$$

care pentru $z_1 = z_2 = 1$ devine: $x_k = c_1 + kc_2$ ceea ce evidențiază o creștere liniară a erorii. Pentru o metodă consistentă condiția necesară și suficientă ca rădăcinile unitare să fie simple este ca polinomul $B(z)$ să nu admită rădăcina unitate, respectiv ca [4]:

$$\sum_{i=0}^p b_i \neq 0.$$

Dacă se analizează polinomul de stabilitate al metodelor de tip Adams:

$$A(z) = a_0 z^p + a_1 z^{p-1},$$

cu $a_0 = -1$ și $a_1 = 1$, rezultă că $z_1 = 1$ și $z_2 = z_3 = \dots = z_{p-1} = 0$. Deoarece toate rădăcinile parazite ale polinomului de stabilitate sunt nule, rezultă că metodele de tip Adams sunt stabile pentru h suficient de mic. Se poate afirma că din punctul de vedere al stabilității numerice la pași mici, metodele de tip Adams sunt optime deoarece au toate rădăcinile perturbante nule. Polinomul de stabilitate al metodei Milne:

$$A(z) = -(z^2 - 1),$$

are în afara rădăcinii principale $z_1 = 1$ și rădăcina parazită $z_2 = -1$, deci este stabil pentru $\lambda h \rightarrow 0$. Deoarece modulul rădăcinii parazite este unitar, rezultă că erorile de rotunjire la un pas nu sunt atenuate la pașii următori ci se propagă, fără a fi însă amplificate. Metoda Milne prezintă o stabilitate numerică la limită. În schimb, relația (8.76) are polinomul de stabilitate:

$$A(z) = -z^2 - 4z + 5 = -(z - 1)(z + 5),$$

la care rădăcina parazită $z_2 = -5$ are modulul supraunitar, ceea ce explică instabilitatea numerică a metodei pentru un pas h oricât de mic. Studiul polinomului $A(z)$ permite caracterizarea comportării metodelor multipas în situația limită $\lambda h \rightarrow 0$. În cazurile reale, de interes practic, atât λ cât și h au valori nenule. Metodele multipas care nu amplifică la pașii următori eroarea de rotunjire apărută la pasul curent, în cazul rezolvării ecuației (8.80) cu h impus se numesc *numeric absolut stabile* pentru acea valoare a pasului. Pentru studiul stabilității numerice absolute se consideră rădăcinile polinomului caracteristic $P(z)$. Deoarece rădăcinile unui polinom depind continuu de coeficienții acestuia, una din rădăcini, z_1 , este rădăcina principală și se obține prin continuitate pornind de la $z_1 = 1$, valoare corespunzătoare parametrului $\lambda h = 0$. În acest caz, rădăcina principală nu mai are valoare unitară ci aproximează valoarea (8.82). Pentru metodele consistente de ordin n , eroarea de aproximare este de tipul $O(\lambda h)^{n+1}$ [9]:

$$z_1 = 1 + \lambda h + \frac{(\lambda h)^2}{2!} + \dots + \frac{(\lambda h)^n}{n!} + O(\lambda h)^{n+1} \quad (8.97)$$

Dar și rădăcinile parazite își modifică poziția, fapt ce influențează în mod diferit propagarea erorii de rotunjire, în funcție de valoarea parametrului λh . Și în acest caz soluția ecuației liniare și omogene cu diferențe finite (8.91) este de forma (8.93). Soluția numerică cea mai apropiată de cea exactă (8.81) se obține dacă sunt realizate condițiile (8.94). Prezența erorilor la inițializare sau a erorilor de rotunjire pe parcursul calculului, face ca soluția numerică să corespundă unor condiții de tipul (8.95). Componenta perturbatoare $c_2 z_2^k$ trebuie să aibă o valoare care să nu depășească în timp componenta “exactă” $c_1 z_1^k$, ceea ce impune inegalitatea:

$$|z_j| < |z_1| \quad (8.98)$$

pentru orice rădăcină parazită $z_j, j = 1, 2, \dots, p-1$. Mulțimea punctelor din planul complex λh , pentru care inegalitatea (8.98) este asigurată poartă numele de *regiune de stabilitate numerică relativă*. În acest domeniu, eroarea chiar dacă se amplifică în timp nu este mare, relativ la soluția “exactă”. În acord cu condiția (8.87) regiunea de stabilitate numerică absolută corespunde condiției $|z_j| \leq 1$ pentru toate rădăcinile polinomului caracteristic.

Considerând spre exemplu, metoda Adams-Bashforth de ordinul doi, la care polinomul caracteristic:

$$P(z) = -z^2 + z + \frac{\lambda h}{2}(3z - 1)$$

are rădăcinile:

$$z_1 = \frac{1}{2} \left[1 + \frac{3}{2}\lambda h + \sqrt{\left(1 + \frac{3}{2}\lambda h\right)^2 - 2\lambda h} \right]$$

$$z_2 = \frac{1}{2} \left[1 + \frac{3}{2}\lambda h - \sqrt{\left(1 + \frac{3}{2}\lambda h\right)^2 - 2\lambda h} \right]$$

se constată că modulele acestor rădăcini devin egale pentru $\lambda h = -2/3$. Pentru valori reale:

$$\lambda h \geq -2/3, |z_1| \geq |z_2|,$$

metoda este numeric relativ stabilă, iar pentru valori reale:

$$\lambda h < -2/3, |z_1| < |z_2|,$$

metoda este numeric instabilă. Pentru analiza sistemelor instabile ($\lambda > 0$), metoda Adams-Bashforth de ordinul doi poate fi aplicată cu orice pas și își menține stabilitatea numerică relativă, dar pentru sisteme stabile ($\lambda = -1/\tau < 0$), metoda Adams-Bashforth de ordinul doi este stabilă doar dacă pasul de integrare este suficient de mic

$h < 2\tau/3$. Valorile reale limită ale domeniilor de stabilitate numerică relativă pentru metodele Adams-Bashforth sunt:

$$\lambda h > -0.667, \quad \text{pentru ordinul 2}$$

$$\lambda h > -0.358, \quad \text{pentru ordinul 3}$$

$$\lambda h > -0.214, \quad \text{pentru ordinul 4.}$$

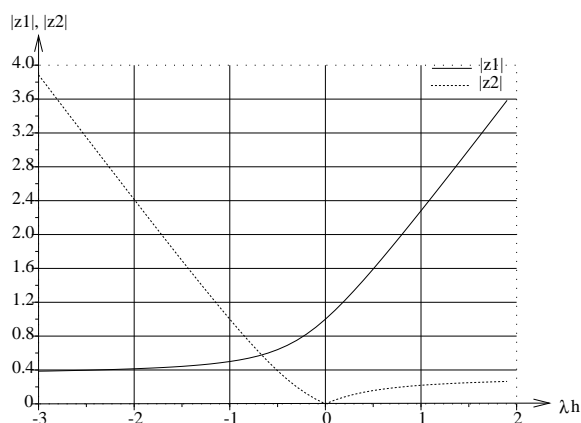


Figura 8.19: Dependența de λh a rădăcinilor polinomului caracteristic pentru metoda Adams – Bashforth de ordinul doi

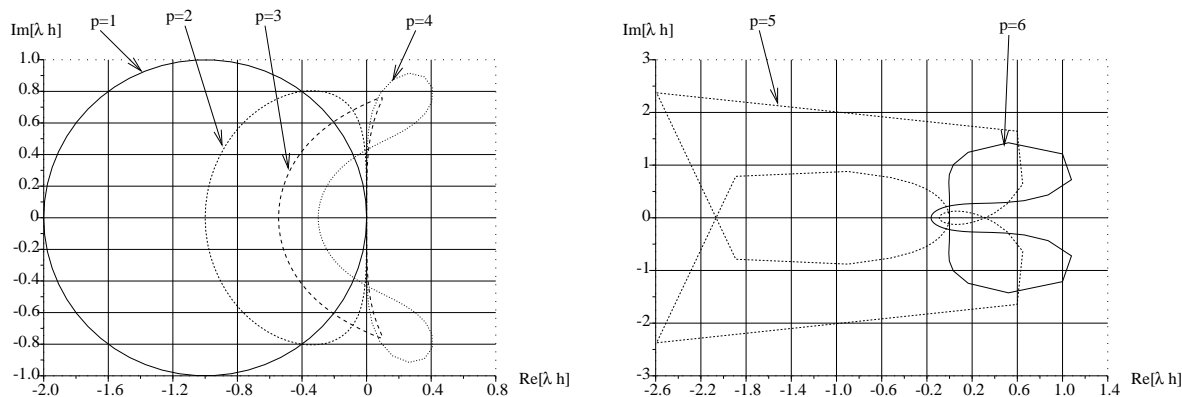


Figura 8.20: Domenii de stabilitate numerică pentru metodele Adams – Bashforth de diferite ordine p

Stabilitatea absolută impune ca atât $|z_1|$ cât și $|z_2|$ să nu depășească unitatea, ceea ce corespunde condiției $-1 < \lambda h < 0$ (figura 8.19). Domeniile de stabilitate numerică absolută pentru metodele de tip Adams-Bashforth de diferite ordine sunt reprezentate în figura 8.20. Spre deosebire de cazul metodelor de tip Runge-Kutta se constată că pe măsură ce ordinul metodei crește, atât domeniul de stabilitate relativă cât și cel de stabilitate absolută se restrâng. Începând de la ordinul trei în sus, domeniul de stabilitate absolută conține un domeniu din axa imaginară, ceea ce le face potrivite și

pentru studiul sistemelor autonome cu soluție oscilantă neamortizată. Metoda Adams-Moulton de ordinul zero (Euler implicită) are polinomul caracteristic:

$$P(z) = -z + 1 + \lambda h z,$$

cu rădăcină unică:

$$z_1 = \frac{1}{1 - \lambda h},$$

iar metoda Adams-Moulton de ordinul unu (metoda trapezelor) are polinomul caracteristic:

$$P(z) = -z + 1 + \frac{\lambda h}{2}(z + 1),$$

cu rădăcina:

$$z_1 = \frac{1 + \frac{\lambda h}{2}}{1 - \frac{\lambda h}{2}}.$$

Deoarece aceste metode implicite nu au rădăcini parazite, rezultă că sunt numeric relativ stabile pentru orice valoare a pasului de integrare. Stabilitatea numerică absolută este asigurată pentru orice valoare λh reală negativă. În semiplanul drept metoda de ordinul unu este numeric absolut instabilă, pe când metoda de ordinul zero este instabilă doar în interiorul unui cerc cu raza unitate și centrul în $(1, 0)$. Cu toate că metoda trapezelor oferă o acuratețe mai bună, metoda Euler implicită este folosită des în practică, deoarece are un domeniu de stabilitate mai extins.

Polinomul caracteristic metodei Adams-Moulton de ordinul doi:

$$P(z) = -z^2 + z + \frac{\lambda h}{12}(5z^2 + 8z - 1)$$

are rădăcinile z_1 și z_2 a căror variație față de parametrul real λh este reprezentată în figura 8.21.

Rădăcina principală z_1 care aproximează funcția $e^{\lambda h}$ este depășită în modul de rădăcina z_2 la valoarea $\lambda h = -1.5$. Modulul rădăcinii parazite $|z_2|$ depășește unitatea la valori $\lambda h < -6$ iar rădăcina principală depășește unitatea pentru $\lambda h > 0$. Rezultă că pe axa reală, regiunea de stabilitate relativă este dată de condiția $\lambda h \leq -1.5$ iar cea de stabilitate absolută este cuprinsă între limitele $-6 < \lambda h < 0$. Limitele reale ale domeniilor de stabilitate numerică relativă pentru metodele Adams-Moulton sunt:

$$\begin{aligned} \lambda h &> -1.500, & \text{pentru } p = 2; \\ \lambda h &> -0.923, & \text{pentru } p = 3; \\ \lambda h &> -0.681, & \text{pentru } p = 4. \end{aligned}$$

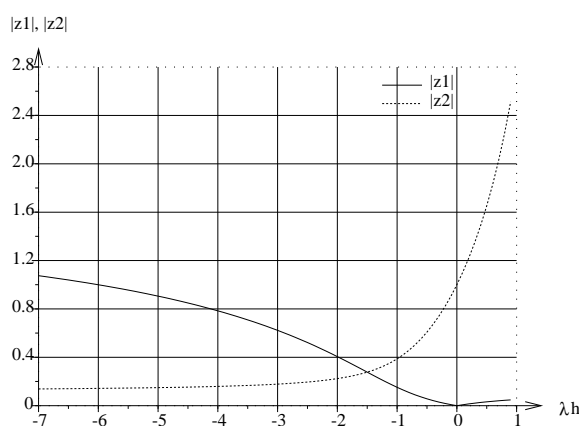


Figura 8.21: Dependenta de λh a rădăcinilor polinomului caracteristic pentru metoda Adams – Moulton de ordinul doi

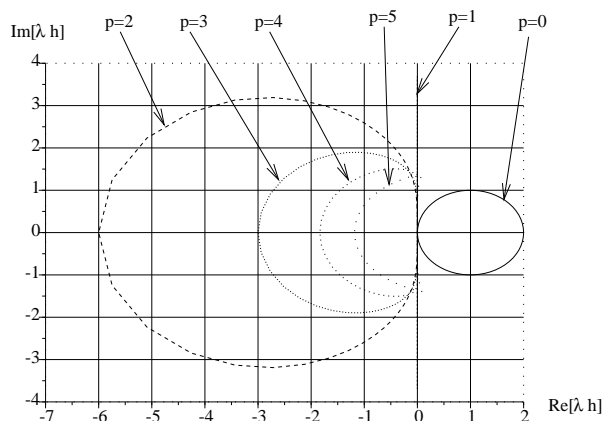


Figura 8.22: Domenii de stabilitate numerică pentru metodele Adams – Moulton de diferite ordine p

Domeniile de stabilitate numerică absolută pentru metodele de tip Adams-Moulton de diferite ordine sunt reprezentate în figura 8.22. Se constată că pe măsură ce ordinul metodei crește, domeniul de stabilitate se restrânge. Domeniile de stabilitate ale metodelor Adams implicite sunt de câteva ori mai mari decât pentru variantele explicite ale metodelor de același ordin.

Polinomul caracteristic metodei Milne:

$$P(z) = -z^2 + 1 + \frac{\lambda h}{3}(z^2 + 4z + 1)$$

are două rădăcini, cea principală [26]: $z_1 = e^{\lambda h} + O(h^5)$ și cea parazită: $z_2 = -e^{-\lambda h/3} + O(h^3)$.

Reprezentând grafic modul în care variază modulul rădăcinilor în funcție de parametrul λh , presupus real, se evidențiază că metoda Milne este numeric absolut stabilă doar

în punctul $\lambda h = 0$. Metoda este totuși numeric relativ stabilă pentru $\lambda h > 0$ ceea ce corespunde sistemelor instabile ($\lambda > 0$).

O teoremă fundamentală privind legătura strânsă între consistență, stabilitate numerică (în sensul relației (8.90), privind rădăcinile polinomului de stabilitate) și convergență este demonstrată în [9]. Dacă o metodă multipas este consistentă și numeric stabilă, atunci ea este convergentă și reciproc dacă o metodă multipas este convergentă ea este numeric stabilă. Pe baza acestei teoreme metodele de tip Adams și metoda Milne sunt stabile. Metoda (8.80) însă, este consistentă dar nu este convergentă deoarece nu este stabilă.

8.5 Algoritmul predictor-corector

Algoritmul predictor-corector folosește în mod combinat o metodă explicită pentru predicție:

$$P : x_k^{(0)} = \sum_{i=1}^p (a_i' x_{k-i} + hb_i x_{k-i}') \quad (8.99)$$

pe baza căreia pentru $j = 0$ se evaluează funcția:

$$E : x_k'^{(j)} = f(x_k^{(j)}, t_k) \quad (8.100)$$

și apoi în mod iterativ o metodă implicită pentru corecție:

$$C : x_k^{(j)} = hx_k'^{(j-1)} = \sum_{i=1}^p (a_i x_{k-i} + hb_i' x_{k-i}'), \quad (8.101)$$

cu $j = 1, 2, \dots, M$.

La fiecare pas de timp t_k sunt parcurse etapele *PECEC.....EC* sau folosind o notație compactă $P(EC)^M$. O variantă îmbunătățită a algoritmului se încheie cu o etapă de evaluare $P(EC)^M E$. Procesul iterativ continuă până când abaterea între două iterații succesive:

$$\varepsilon_j = \|x_k^{(j)} - x_k^{(j-1)}\| \quad (8.102)$$

scade sub o toleranță impusă erorii. Condiția (8.102) determină numărul de iterații M , efectuate.

Prima problemă care se formulează în legatură cu algoritmul predictor-corector constă în determinarea condițiilor în care algoritmul iterativ de corecție este convergent.

Etapele de evaluare (8.100) și corecție (8.101) pot fi prezentate compact sub forma:

$$x_k^{(j)} = F(x_k^{(j-1)}), \quad (8.103)$$

în care aplicația F este definită de:

$$F(x) = hb_0 f(x, t_k) + \sum_{i=1}^p (a_i x_{k-i} + hb_i' x_{k-i}'). \quad (8.104)$$

Se constată că limita căutată prin procedeul iterativ de corectare este chiar punctul fix al aplicației $F(x)$.

Pe baza principiului contracției, condiția suficientă ca șirul:

$$x_k^{(0)}, x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(j)}, \dots$$

definit de relația recursivă (8.103) să fie convergent, iar limita sa:

$$x_k = \lim_{j \rightarrow \infty} x_k^{(j)}$$

să fie punctul fix al aplicației (8.104) este ca $F(x)$ să fie o contracție. Cu alte cuvinte, $F(x)$ trebuie să aibă proprietatea Lipschitz cu o constantă subunitară, respectiv să existe o constantă $L < 1$, astfel încât:

$$\|F(x') - F(x'')\| \leq L \cdot \|x' - x''\|, \quad (8.105)$$

pentru orice pereche x', x'' dintr-o vecinătate a punctului fix x_k .

Presupunând că problema cu condiții inițiale este bine formulată, rezultă că funcția $f(x, t)$ specifică ecuației de rezolvat are proprietatea Lipschitz, cu o constantă care pentru momentul t_k va fi notată L_k , deci:

$$\|F(x'_k) - F(x''_k)\| = h|b_0| \cdot \|f(x'_k, t_k) - f(x''_k, t_k)\| \leq h|b_0| \cdot L_k \cdot \|x'_k - x''_k\|. \quad (8.106)$$

Pe baza relației (8.105) se deduce condiția ca procedeul iterativ să fie convergent:

$$h < \frac{1}{|b_0| \cdot L_k}. \quad (8.107)$$

Condiția de convergență nu impune derivabilitatea funcției $f(x, t)$. Dacă totuși aceasta este derivabilă, norma derivatei, egală cu constanta Lipschitz, trebuie să satisfacă inegalitatea:

$$L_k = \left\| \frac{\partial f}{\partial x} \right\| < \frac{1}{h \cdot |b_0|} \quad (8.108)$$

pentru $t = t_k$ și orice $x_k^{(j)}$ cu $j = 0, 1, 2, \dots$.

Cu cât constanta de contracție este mai mică, cu atât algoritmul este mai rapid convergent:

$$\varepsilon_{j+1} = \|x_k^{(j+1)} - x_k^{(j)}\| = \|F(x_k^{(j)}) - F(x_k^{(j-1)})\| \leq L \cdot \|x_k^{(j)} - x_k^{(j-1)}\| \leq L^j \cdot \|x^1 - x^0\| = L^j \cdot \varepsilon_1 \quad (8.109)$$

Dacă inegalitatea (8.107) nu este îndeplinită, atunci procesul iterativ poate fi divergent și norma abaterii (8.102) crește de la un pas la altul.

Analizând condițiile (8.107) sau (8.108) rezultă că pentru a obține o convergență rapidă pasul de timp trebuie să fie suficient de mic, iar dacă pasul crește, rata de convergență

scade. Pentru a optimiza efortul global de calcul trebuie făcut un compromis între valoarea pasului și rata de convergență. Metoda iterativă prezentată are avantajul că abaterea (8.102) constituie un criteriu eficient al convergenței. Se recomandă ca pasul să fie astfel ales încât să fie necesare cel mult două sau maxim trei iterații pentru a obține toleranța impusă erorii.

Comparând algoritmul predictor – corector cu algoritmul Runge - Kutta de ordinul patru se constată că primul este de cel puțin două ori mai eficient din punctul de vedere al efortului de calcul, la același ordin de mărime al erorii soluției.

În forma sa clasică, metoda predictor - corector folosea relații de tip Milne. Pentru a micșora erorile de trunchiere și pentru a elimina instabilitățile de calcul, Hamming [7], [6] propune introducerea înainte de etapa de evaluare a unei etape suplimentare numită *de modificare*. În versiunea Hamming este esențial ca predictorul și corectorul să ofere același ordin al erorii de trunchiere, chiar dacă numărul de pași este diferit în cele două relații. Modificarea operează asupra corectorului, folosind diferența dintre predictor și corector de la pasul anterior și urmărește anularea erorii de trunchiere. Pentru relațiile de ordinul patru se propune:

$$\begin{aligned} P : p_k &= x_{k-4} + \frac{4h}{3}(2x_{k-1} - x_{k-2} + 2x_{k-3}); \\ M : m_k &= p_k - \frac{11^2}{121}(p_{k-1} - c_{k-1}); \\ E : m_k &= f(m_k, t_k); \\ C : c_k &= \frac{1}{8}[9x_{k-1} - x_{k-2} + 3h(m_k + 2x_{k-1} - x_{k-2})]. \end{aligned}$$

În final, pentru soluția de la momentul t_k se adoptă valoarea:

$$x_k = c_k + \frac{9}{121}(p_k - c_k).$$

În versiunea acestui algoritm implementată în [6] nu se efectuează iterații asupra valorii prezise, în schimb se adoptă o strategie a modificării pasului de timp, bazată pe diferența dintre predictor și corector la momentul de timp k :

$$\varepsilon_k = \sum_{i=1}^n g_i \cdot |p_k^{(i)} - c_k^{(i)}|.$$

în care indicele i este asociat variabilei de stare curente, iar g_i este ponderea dată acestuia în calculul erorii.

Se deosebesc patru cazuri:

- dacă $\varepsilon_k < \varepsilon_m$, atunci pasul de timp se dublează;
- dacă $\varepsilon_m < \varepsilon_k \leq \varepsilon_M$, atunci valoarea pasului de timp se menține constantă;
- dacă $\varepsilon_k > \varepsilon_M$, atunci pasul de timp se înjumătățește;

- dacă $\varepsilon_k > \varepsilon_{\max} \geq \varepsilon_M$, atunci soluția se recalculează fără avansarea pasului.

Pentru a permite modificarea pasului, soluția este memorată pe 5 pași. La înjumătățire sunt folosite relații de interpolare care asigură erori de cel puțin același ordin ca relațiile de integrare.

8.6 Reprezentarea canonică a metodelor multipas

Prezentarea algoritmului predictor - corector într-o formă matriceală, numită *reprezentare canonică* urmărește mai multe aspecte:

- unificarea teoriei generale a metodelor cu mai mulți pași;
- programarea mai eficientă a algoritmului;
- facilitarea modificării ordinului și a pasului.

Pentru a simplifica notațiile se consideră o singură ecuație diferențială scalară:

$$\frac{dx}{dt} = f(x, t) \quad (8.110)$$

cu condiția inițială $x(t_0) = x_0 \in \mathbf{R}$. Valorile soluției numerice și ale pantei acesteia în ultimele p noduri ale unei rețele uniforme de discretizare $t_{k-p}, t_{k-p+1}, \dots, t_{k-2}, t_{k-1}$ pot fi considerate componentele unui "vector soluție numerică":

$$y_{k-1} = [x_{k-1}, x_{k-2}, \dots, x_{k-p}, hx'_{k-1}, hx'_{k-2}, \dots, hx'_{k-p}]^T. \quad (8.111)$$

Scopul unei metode multipas constă în determinarea vectorului y_k pornind de la y_{k-1} și avansând cu un pas de timp, în vederea calculului soluției numerice a ecuației (8.110). Acest proces poate fi aplicat succesiv pentru a calcula y_1, y_2, \dots, y_m , presupunând pe y_0 cunoscut. Problema calculului inițializării y_0 , numită și problema "startului", face ca metodele multipas să fie evitate. Totuși, pentru probleme mari, care solicită acuratete a soluției numerice, eficiența oferită de metodele multipas este net superioară metodelor cu un pas. O metodă uzuală în rezolvarea problemei startului constă în utilizarea metodelor cu un pas, ca de exemplu metoda Runge - Kutta, pentru a parcurge primii $(p - 1)$ pași și pentru a calcula y_0 . La start trebuie acordată o atenție deosebită controlului erorii, deoarece o inițializare inexactă, afectată de erori grosolane, poate influența negativ exactitatea soluției numerice de la pașii următori. În prima etapă a calculului, la un nou pas de timp se acceptă procedeul de predicție, bazat pe relația (8.99), prin care valoarea inițială $y_k^{(0)}$ a vectorului y_k se obține printr-o transformare liniară pornind de la vectorul y_{k-1} . Prima componentă $x_k^{(0)}$ se calculează cu relația (8.99) iar următoarele $(p - 1)$ se obțin printr-o translație. Pentru calculul pantei $hx_k^{(0)}$ se utilizează relația (8.101), în care $x_k^{(1)}$ se consideră dat de (8.99), deci:

$$hx'_k = \sum_{i=1}^p [c_i x_{k-i} + hd_i x'_{k-i}], \quad (8.112)$$

cu $c_i = (a'_i - a_i)b_0$, $d_i = (b'_i - b_i)b_0$, iar următoarele componente se obțin tot prin translație. Acest mod de scriere pentru $x_k^{(0)}$ prezintă avantaje ulterioare în calculul corectorului. În consecință, etapa poate fi reprezentată matriceal sub forma:

$$y_k^{(0)} = Y \cdot y_{k-1}, \quad (8.113)$$

sau dezvoltat:

$$\underbrace{\begin{bmatrix} x_k \\ x_{k-1} \\ x_{k-2} \\ \dots \\ x_{k-p+1} \\ hx'_{k-1} \\ hx'_{k-2} \\ \dots \\ hx'_{k-p+1} \end{bmatrix}}_{y_k^{(0)}} = \underbrace{\begin{bmatrix} a'_1 & a'_2 & \dots & a'_{p-1} & a'_p & b'_1 & b'_2 & \dots & b'_{p-1} & b'_p \\ 1 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ c_1 & c_2 & \dots & c_{p-1} & c_p & d_1 & d_2 & \dots & d_{p-1} & d_p \\ 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}}_Y \cdot \underbrace{\begin{bmatrix} x_{k-1} \\ x_{k-2} \\ x_{k-3} \\ \dots \\ x_{k-p} \\ hx'_{k-1} \\ hx_{k-2} \\ hx_{k-3} \\ \dots \\ hx_{k-p} \end{bmatrix}}_{y_{k-1}}.$$

Procedeeul de corecție se bazează pe relațiile (8.100), (8.101). Analizând relația (8.101), rezultă că termenul care conține simbolul de sumă rămâne neschimbat în timpul iterației, deci prin scăderea relațiilor pentru două iterații succesive, se obține:

$$x_k^{(j+1)} = x_k^{(j)} + hb_0[x_k^{(j)} - x_k^{(j-1)}]. \quad (8.114)$$

Această relație se poate aplica și pentru cazul $j = 0$ dacă se adoptă formal pentru $x_k^{(-1)}$ valoarea $x_k^{(0)}$ dată de relația (8.112).

Sub formă matriceală, relația (8.114) are forma:

$$y_k^{(j+1)} = y_k^{(j)} + F(y_k^{(j)}) \cdot c_y \quad (8.115)$$

cu $j = 0, 1, 2, \dots, (M - 1)$, sau dezvoltat:

$$\underbrace{\begin{bmatrix} x_k^{j+1} \\ x_{k-1} \\ \vdots \\ x_{k-p+1} \\ hx_k^{(j+1)} \\ hx'_{k-1} \\ \vdots \\ hx'_{k-p+1} \end{bmatrix}}_{y_k^{(j+1)}} = \underbrace{\begin{bmatrix} x_k^{(j)} \\ x_{k-1} \\ \vdots \\ x_{k-p+1} \\ hx_k^{(j)} \\ hx'_{k-1} \\ \vdots \\ hx'_{k-p+1} \end{bmatrix}}_{y_k^{(j)}} + h \underbrace{\left[f(x_k^{(j)}, t_k) - x_k^{(j)} \right]}_{F(y_k^{(j)})} \cdot \underbrace{\begin{bmatrix} b_0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{c_y}.$$

În concluzie, metoda predictor - corector sub formă matriceală:

$$\begin{aligned} y_k^{(0)} &= Y \cdot y_{k-1} \\ y_k^{(j+1)} &= y_k^{(j)} + F(y_k^{(j)}) \cdot c_y, j = 0, 1, \dots, (M-1) \end{aligned} \quad (8.116)$$

este caracterizată de o matrice de predicție Y , de dimensiune $2p \times 2p$ care conține toate informațiile privind atât coeficienții predictorului, cât și ai corectorului (cu excepția lui b_0). Această matrice înmulțește o singură dată vectorul y_{k-1} și nu depinde de funcția $f(x, t)$, predicția fiind invariantă la ecuația de rezolvat.

Corecția reprezintă un procedeu iterativ iar la fiecare iterație se modifică doar două componente din vectorul soluție $y_k^{(j)}$ și anume prima și componenta de rang $(p+1)$, care corespund valorilor $x_k^{(j)}$ și $x_k'^{(j)}$. Pentru aceasta se evaluează funcția $f(x, t)$ la momentul curent t_k , pentru ultima valoare a soluției numerice $x_k^{(j)}$. Valoarea soluției numerice (prima componentă a vectorului $y_{k-1}^{(j)}$) se incrementează cu:

$$b_0 h [f(x_k^{(j)}, t_k) - x_k'^{(j)}], \quad (8.117)$$

iar valoarea pantei soluției x_k (componenta de rang $(p+1)$ a vectorului $y_k^{(j)}$) se actualizează la valoarea nou calculată:

$$x_k^{(j+1)} = h f(x_k^{(j)}, t_k). \quad (8.118)$$

La prima iterație, pentru $j = 0$, valoarea $x_k'^{(0)}$ necesară relației (8.117) este preluată din vectorul $y_k^{(0)}$, de la componenta $(p+1)$, a cărei valoare este dată de (8.112). Prin substituție în (8.117) se obține:

$$x_k^{(1)} = x_k^{(0)} + b_0 \left[h f(x_k^{(0)}, t_k) - \sum_{i=1}^p (u x_{k-i} + h d_i x_{k-i}') \right],$$

sau ținând seama de expresiile coeficienților c_i, d_i precum și de expresia predictorului $x_k^{(0)}$, rezultă relația (8.101), particularizată pentru $j = 1$, care dă prima iterație a corectorului.

Acest mod de scriere matriceală a corectorului evidențiază că el este determinat în principal de funcția $f(x, t)$ și de vectorul c_y , a cărui componentă principală este coeficientul termenului implicit b_0 .

În continuare se va exemplifica scrierea matriceală a metodei predictor-corector pentru cazul utilizării relațiilor de tip *Adams* pe trei pași (8.68), (8.69). Coeficienții acestor relații au valorile:

$$\begin{aligned} a_1' &= 1, & a_2' &= 0, & a_3' &= 0; \\ b_1' &= 23/12, & b_2' &= -16/12, & b_3' &= 5/12; \\ a_1 &= 1, & a_2 &= 0, & a_3 &= 0; \\ b_1 &= 19/24, & b_2 &= -5/24, & b_3 &= 1/24 \end{aligned}$$

și $b_0 = 9/24$, care permite calculul coeficienților:

$$\begin{aligned} c_1 &= 0, & c_2 &= 0, & c_3 &= 0; \\ d_1 &= 3, & d_2 &= -3, & d_3 &= 1, \end{aligned}$$

cea ce corespunde matricei de predicție:

$$Y = \begin{bmatrix} 1 & 0 & 0 & \frac{23}{12} & -\frac{16}{12} & \frac{5}{12} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & -3 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

și matricei de corecție:

$$c_y = \left[\frac{3}{8} \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \right]^T.$$

Deoarece din vectorul soluțiilor:

$$y_k = \left[x_k \quad x_{k-1} \quad x_{k-2} \quad hx'_k \quad hx'_{k-1} \quad hx'_{k-2} \right]^T,$$

componentele x_{k-1} și x_{k-2} nu sunt utilizate nici în procesul de predicție nici în cel de corecție, acestea pot fi eliminate și odată cu ele se elimină și elementele corespunzătoare din matricele Y (liniile și coloanele 2, 3) și c_y (liniile 2, 3):

Aceasta corespunde relației de predicție:

$$\begin{bmatrix} x_k^{(0)} \\ hx_k^{(0)} \\ hx_{k-1} \\ hx_{k-2} \end{bmatrix} = \begin{bmatrix} 1 & \frac{23}{12} & -\frac{16}{12} & \frac{5}{12} \\ 0 & 3 & -3 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_{k-1} \\ hx'_{k-1} \\ hx'_{k-2} \\ hx'_{k-3} \end{bmatrix} \quad (8.119)$$

și relației de corecție cu $j = 0, 1, 2, \dots, (M - 1)$:

$$\begin{bmatrix} x_k^{(j+1)} \\ hx_k^{(j+1)} \\ hx'_{k-1} \\ hx'_{k-2} \end{bmatrix} = \begin{bmatrix} x_k^{(j)} \\ hx_k^{(j)} \\ hx'_{k-1} \\ hx'_{k-2} \end{bmatrix} + F \cdot \begin{bmatrix} \frac{3}{8} \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (8.120)$$

unde:

$$F = h \left[f(x_k^{(j)}, t_k) - x_k^{(j)} \right], \quad (8.121)$$

relații în concordanță cu cele scalare.

Constatăm că prin reprezentarea matriceală datele au căpătat o structură simplă și elegantă. Algoritmul predictor-corector este caracterizat prin matricea de predicție Y , de dimensiune 4×4 și vectorul de corecție c_y de dimensiune 4. Soluția este reprezentată la pasul curent de un vector cu patru elemente, din care primul dă valoarea soluției iar al doilea dă panta ei de variație în timp, multiplicată cu pasul de integrare h .

După inițializarea corespunzătoare a vectorului soluție y_2 la pasul de timp $k = 2$ în vederea calculului soluției numerice la pasul $k = 3$, se aplică relația de predicție (8.119) iar asupra valorilor obținute în vectorul y_k se aplică iterativ procedura de corecție (8.120), care presupune evaluarea funcției $f(x, t)$ la fiecare iterație, conform relației

(8.121). Iterațiile se opresc atunci când norma abaterii între două iterații succesive scade sub toleranța impusă. Valoarea obținută se consideră soluția numerică de la pasul k : $y_k = y_k^{(M)}$ și apoi se incrementează pasul de timp, iar algoritmul predictor-corrector se reia până la atingerea timpului final.

Algoritmul prezentat se poate generaliza ușor pentru cazul rezolvării unor sisteme de n ecuații diferențiale. În acest caz elementele vectorului soluție y_k sunt la rândul lor vectori cu n componente, relațiile (8.119), (8.120) aplicându-se în mod independent pentru fiecare componentă în parte. Dar în relația (8.121), în evaluarea funcției $f(x, t)$ pot interveni toate componentele soluției la pasul curent și la iterația curentă.

8.7 Metode cu valori multiple

În reprezentarea matriceală canonică a metodei predictor-corrector, soluția la un pas de integrare este caracterizată printr-un vector cu $2p$ componente:

$$y_k = [x_k, x_{k-1}, \dots, x_{k-p}, hx_k, hx_{k-1}, \dots, hx'_{k-p}]^T, \quad (8.122)$$

din care primele p constituie valorile soluției numerice pe ultimii pași de integrare, iar ultimele p componente reprezintă pantele soluției la acei pași, pante înmulțite cu valoarea pasului de timp h .

Această reprezentare a soluției nu este cea mai potrivită din punctul de vedere al efortului de calcul sau al erorii de rotunjire. În aceste cazuri se poate dovedi mai potrivită o reprezentare a soluției de la un moment de timp prin $2p$ combinații liniare ale componentelor canonice ale vectorului 8.122. Se presupune că aceste combinații sunt organizate într-un nou vector notat z_k . Trecerea de la vechiul vector y_k la noul vector este asigurată de o transformare liniară:

$$z_k = T \cdot y_k, \quad (8.123)$$

reprezentată de o matrice pătrată de dimensiune $2p \times 2p$. Pentru ca noua reprezentare să fie echivalentă celei vechi este necesar ca transformarea liniară (8.123) să fie inversabilă, deci matricea T să fie nesingulară. În aceste condiții trecerea inversă este asigurată de relația:

$$y_k = T^{-1}z_k. \quad (8.124)$$

Folosind noua reprezentare echivalentă, forma canonică a metodei predictor-corrector devine:

$$\begin{aligned} z_k^{(0)} &= T y_k^{(0)} = T Y y_{k-1} = T Y T^{-1} z_{k-1}; \\ z_k^{(j+1)} &= T y_k^{(j)} + F(y_k^{(j)}) T c_y = z_k^{(j)} + F(T^{-1} z_k^{(j)}) c_y, \end{aligned} \quad (8.125)$$

sau folosind relațiile:

$$Z = T Y T^{-1}; \quad G(z_k) = F(T^{-1} z_k); \quad c_z = T c_y, \quad (8.126)$$

se obține în final:

$$z_k^{(0)} = Z \cdot z_{k-1}; \quad z_k^{(j+1)} = z_k^{(j)} + G(z_k^{(j)}) c_z. \quad (8.127)$$

Relațiile (8.127) reprezintă forma matriceală canonică pentru vectorul echivalent z_k a metodei predictor-corector. Deoarece componentele vectorului z_k , soluție de la pasul t_k nu mai reprezintă valorile soluției pe ultimii pași, metoda generată de relațiile (8.127) nu mai poate fi numită o metodă multipas. Ea se va numi în continuare o *metodă cu valori multiple ale soluției*.

Metodele cu valori multiple generalizează metodele cu mai mulți pași. Dacă matricea de transformare T este matricea unitate, atunci metoda cu valori multiple devine o metodă multipas.

Trebuie menționat că o transformare de echivalență nu afectează eroarea de trunchiere sau stabilitatea metodei, singurele aspecte care se pot modifica sunt efortul de calcul și ordinul erorii de rotunjire. Pe un calculator ideal, fără eroare de rotunjire, două metode echivalente trebuie să obțină rezultate identice pentru aceeași problemă.

Un exemplu îl constituie utilizarea în locul vectorului canonic $y_k = [x_k, hx'_k, hx'_{k-1}, hx'_{k-2}]$ a vectorului echivalent $z_k = [x_k, hx'_k, x_{k-1}, hx'_{k-1}]$. Componentele acestuia reprezintă valoarea soluției pe ultimii doi pași dar deoarece el este echivalent cu o reprezentare pe trei pași nu se poate spune că metoda generată este cu doi pași, ci mai corect că este o metodă cu patru valori.

8.7.1 Reprezentarea Nordsieck

În continuare se va prezenta o tehnică de alegere a transformării T , datorată lui Nordsieck, care facilitează modificarea pasului de timp h , pe parcursul integrării.

În reprezentarea Nordsieck, soluția este caracterizată la pasul curent de vectorul echivalent:

$$z_k = \left[x_k; hx'_k; \frac{h^2}{2}x''_k; \dots; \frac{h^{2p+1}}{(2p+1)!}x_k^{(2p+1)} \right]^T, \quad (8.128)$$

care conține drept componente primii $2p$ termeni ai seriei Taylor asociate soluției exacte.

În acest caz, valorile multiple constau de fapt în valorile derivatei funcției la momentul curent, cu ordine începând de la *zero* până la $(2p+1)$, derivate ponderate corespunzător. Cu toate că o astfel de reprezentare este echivalentă formei canonice multipas, ea nu se mai poate numi așa, deoarece nu mai conține în mod explicit informații privind comportarea soluției pe mai mulți pași de timp.

Pentru a obține forma canonică a metodei predictor-corector în reprezentarea Nordsieck trebuie determinată matricea T a transformării de echivalență. În acest scop se consideră un polinom arbitrar de gradul trei:

$$x(t) = a_0 + a_1t + a_2t^2 + a_3t^3$$

ale cărui derivate sunt:

$$\begin{aligned} x'(t) &= a_1 + 2a_2t + 3a_3t^2; \\ x''(t) &= 2a_2 + 6a_3t; \\ x'''(t) &= 6a_3. \end{aligned}$$

Pe o rețea de discretizare uniformă cu $t_k = 0$, $t_{k-1} = -h$, $t_{k-2} = -2h$, componentele vectorului canonic sunt:

$$\begin{aligned}x_k &= a_0; \\x'_k &= a_1; \\x'_{k-1} &= a_1 - 2a_2h + 3a_3h^2; \\x'_{k-2} &= a_1 - 4a_2h + 12a_3h^2.\end{aligned}$$

Pentru acest caz, componentele vectorului Nordsieck sunt:

$$\begin{aligned}x_k &= a_0; \\x'_k &= a_1; \\x'_{k-1} &= a_1 - 2a_2h + 3a_3h^2; \\x'_{k-2} &= a_1 - 4a_2h + 12a_3h^2.\end{aligned}$$

Matricea de transformare are inversa dată de:

$$y_k = \begin{bmatrix} x_k \\ hx'_k \\ hx'_{k-1} \\ hx'_{k-2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -2 & 3 \\ 0 & 1 & -4 & 12 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ ha_1 \\ h^2a_2 \\ h^3a_3 \end{bmatrix} \quad (8.129)$$

Prin inversarea matricei T^{-1} se obține:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{3}{4} & -1 & \frac{1}{6} \\ 0 & \frac{1}{6} & \frac{-1}{3} & \frac{1}{6} \end{bmatrix}. \quad (8.130)$$

Folosind forma canonică a matricelor Y și c_y dată de (8.119), (8.120) precum și relațiile de transformare (8.126), rezultă:

$$Z = TYT^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad c_z = Tc_y = \begin{bmatrix} 3/8 \\ 1 \\ 3/4 \\ 1/6 \end{bmatrix}. \quad (8.131)$$

Deoarece atât vectorul canonic cât și cel Nordsieck conțin componentele x_k , hx'_k și acestea ocupă aceleași poziții (1) și (2), rezultă că funcția F nu suferă practic modificări:

$$G(z_k) = F(y_k) = h [f(x_k^{(j)}, t_k) - x_k^{(j)}]. \quad (8.132)$$

Matricea de predicție Z are forma triunghiular superioară iar elementele acestei matrice se recunosc a fi cele din triunghiul Pascal:

$$z_{i+1,j+1} = C_i^j = \frac{j!}{(j-i)! \cdot i!}, \quad 0 \leq i \leq j < 4. \quad (8.133)$$

Acest lucru nu este întâmplător deoarece relația de predicție exprimă de fapt vectorul Nordsieck de la pasul $t = t_{k+1}$:

$$y_{k+1} = \left[x_{k+1}; hx'_{k+1}; \dots; \frac{h^p x_{k+1}^{(p)}}{p!} \right]^T$$

și vectorul Nordsieck de la momentul de timp $t = t_k$:

$$y_k = \left[x_k; hx'_k; \dots; \frac{h^p x_k^{(p)}}{p!} \right]^T$$

Folosind dezvoltarea în serie Taylor trunchiată a soluției în origine se obține:

$$x(t) = \sum_{j=0}^p \frac{x_k^{(j)}}{j!} t^j = \sum_{j=0}^p \frac{z_k^{(j)}}{h^j} t^j,$$

în care s-a notat cu $z_k^{(j)}$ componenta j a vectorului Nordsieck de la pasul k . Derivata de ordin i a acestei funcții este:

$$x^{(i)}(t) = \sum_{j=0}^p \frac{j!}{(j-i)!} \frac{z_k^{(j)}}{h^j} t^{j-i}, \quad \text{cu } i \leq j,$$

sau particularizând pentru $t = h$, rezultă:

$$x_{k+1}^{(i)} = \sum_{j=0, j \geq i}^p \frac{j! h^{-i}}{(j-i)!} z_k^{(j)},$$

ceea ce permite calculul componentei i a vectorului Nordsieck la momentul $k+1$:

$$z_{k+1}^{(i)} = \frac{h^i}{i!} x_{k+1}^{(i)} = \sum_{j=0, j \geq i}^p \frac{j!}{(j-i)! \cdot i!} z_k^{(j)}.$$

Coeficienții relației de legătură formează un triunghi Pascal iar relația (8.133) este valabilă în cazul general:

$$z_{i+1, j+1} = C_i^j = \frac{j!}{(j-i)! \cdot i!}, \quad (8.134)$$

cu $0 \leq i \leq j < p$.

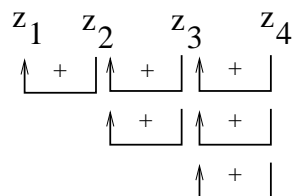
Deoarece matricea Z are mai multe elemente nule decât matricea Y s-ar părea că efortul de calcul necesar predicției este mai mare. În realitate, dacă se ia în considerare relația:

$$C_i^j + C_{i+1}^j = C_{i+1}^{j+1},$$

rezultă că produsul matricelor $Z \cdot z_{k-1}$ se poate realiza doar prin operații de adunare. Produsul dintre triunghiul Pascal și vectorul Nordsieck:

$$z_{k-1} = [z_1, z_2, \dots, z_p]^T$$

constă în operațiile (exemplificate pentru $p = 4$):



$$\begin{aligned}
 z_j &= z_j + z_{j+1}; & j &= p-1 \rightarrow 1 \\
 z_j &= z_j + z_{j+1}; & j &= p-1 \rightarrow 2 \\
 & \dots\dots\dots & & \\
 z_{p-1} &= z_{p-1} + z_p
 \end{aligned}$$

deci în total un număr de $p(p-1)/2$ adunări.

Segmentul de cod pseudocod care asigură aceste operații pentru cazul a n ecuații și p componente în vectorul Nordsieck de dimensiune $z(p, n)$ este:

```

pentru j = 2, p
  pentru j1 = j, p
    j2 = p - j1 + j - 1
    pentru i = 1, n
      z(j2, i) = z(j2, i) + z(j2 + 1, i)
  
```

Eliminarea înmulțirilor din etapa de predicție scurtează sensibil timpul de calcul asociat acestei etape.

Un alt avantaj al reprezentării Nordsieck constă în faptul că utilizatorul are la dispoziție la fiecare pas de integrare valoarea soluției și a primelor sale derivate. Cel mai evident avantaj al acestei metode constă în ușurința cu care se poate modifica pasul de timp pe parcursul integrării.

8.7.2 Controlul automat al ordinului și mărimii pasului în metodele cu valori multiple

Spre deosebire de cazul metodelor cu un pas, la care creșterea ordinului determină sporierea efortului de calcul prin mărirea numărului de evaluări ale funcției $f(x, t)$, în cazul metodelor cu valori multiple modificarea pasului nu influențează sensibil efortul de calcul.

Faptul că micșorarea mărimii pasului determină scăderea erorii de trunchiere cu atât mai rapid cu cât metoda are ordin mai înalt poate duce la ideea că metodele de ordin ridicat sunt mai bune decât cele de ordin scăzut. În realitate, metodele de ordin ridicat au un domeniu de stabilitate mai restrâns, ceea ce face ca la un pas dat probabilitatea de apariție a instabilităților numerice să fie mai mare la metodele de ordin mai înalt.

În metodele de tip predictor-corector, la care numărul de iterații corectoare este suficient de mare pentru ca soluția la pasul k să satisfacă practic exact ecuația corectorului, eroarea de trunchiere este dată de ordinul erorii corectorului. Dacă numărul de iterații este mic iar soluția numerică adoptată este relativ depărtată de punctul fix al relației de corecție, atunci eroarea de trunchiere are ordinul erorii predictorului. Aceleași afirmații sunt adevărate și cu privire la stabilitatea numerică a metodelor predictor-corector.

După cum s-a arătat, metodele Adams-Bashforth de ordinul p au o eroare locală de trunchiere de tipul $O(h^{p+1})$, care poate fi exprimată sub forma:

$$\varepsilon_k = C_k \cdot h^{p+1}.$$

Eroarea globală la momentul de timp t_m nu trebuie să depășească toleranța impusă:

$$\varepsilon_k \frac{t_m}{h} = C_k \cdot h^p \cdot t_m \leq \varepsilon_{\max}.$$

Dacă se reprezintă modul de variație a erorii globale medii pe pas $\varepsilon_{\max} = \varepsilon_k/h$ în funcție de mărimea pasului pentru diferite ordine ale metodei, se obțin grafice de tipul celor din figura 14.

Figura 14 Eroarea globală medie

Din analiza acestei figuri se constată că pentru o eroare impusă, la fiecare ordin pasul are o valoare maximă. Ordinul care permite pasul maxim depinde de mărimea erorii impuse. La erori mici, cel mai mare pas maxim îl au metodele de ordin ridicat, în schimb la erori mari sunt preferate metodele de ordin mic. În acest fel axa erorilor se divide în subdomenii, cu proprietatea că pe fiecare subdomeniu este recomandat un anumit ordin optim. Modul în care variază eroarea de trunchiere în funcție de mărimea pasului, cu condiția ca metoda să aibă ordinul optim, este dat de curba anvelopă la dreapta pentru familia de grafice din figura 14.

Din punctul de vedere al programării, micșorarea ordinului cu o unitate constă în principiu în anularea ultimelor linii din matricele Y, Z, c_y sau c_z și a ultimei coloane din matricele Y și Z . Creșterea ordinului presupune adăugarea la loc a elementelor eliminate anterior.

Cu toate că ultimul element din vectorul soluție y_k nu mai este necesar după micșorarea ordinului, el este memorat în continuare pentru cazul în care ordinul va fi din nou majorat. Acest lucru presupune ca la start inițializarea vectorului y_0 să fie făcută pentru cazul corespunzător ordinului maxim. În implementările practice, cu control automat al ordinului, acesta variază între 1 și maxim 6.

Pentru a asigura salvarea ultimelor elemente din vectorul y_k , la trecerea la pasul următor, în zona anulată a matricelor Y sau Z este necesar ca elementele subdiagonale să fie unitare. De exemplu pentru matricea (8.131):

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 0 \\ 0 & 0 & 1 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

În cazul metodelor multiple, modificarea mărimii pasului de integrare întâmpină dificultăți mult mai mari decât în cazul metodelor cu un pas. Dacă pentru pasul de timp h , la momentul t_k , vectorul soluție este:

$$y_k = [x_k, x_{k-1}, \dots, x_{k-p}, h'x'_k, h'x'_{k-1}, \dots, h'x'_{k-p}] \quad (8.135)$$

și se dorește modificarea pasului la valoarea $h' = \alpha \cdot h$, pentru calculele următoare este necesar calculul vectorului:

$$\hat{y}_k = [\hat{x}_k, \hat{x}_{k-1}, \dots, \hat{x}_{k-p}, h'\hat{x}'_k, h'\hat{x}'_{k-1}, \dots, h'\hat{x}'_{k-p}], \quad (8.136)$$

în care doar două componente se obțin direct din (8.135): $\hat{x}_k = x_k$ și $h'\hat{x}'_k = \alpha \cdot h \cdot x'_k$. Celelalte componente reprezintă valori ale soluției numerice și ale pantei acestora, valori ale variabilei independente t , care nu sunt noduri ale rețelei inițiale de discretizare. Dacă se dorește micșorarea pasului de discretizare ($\alpha < 1$), atunci aceste valori se pot calcula prin interpolare. Din punct de vedere matriceal, realizarea interpolării este echivalentă cu transformarea liniară:

$$\hat{y}_k = C(\alpha) \cdot y_k \quad (8.137)$$

în care $C(\alpha)$ este o matrice pătrată ale cărei elemente depind de α . Dacă se dorește mărirea pasului de discretizare ($\alpha > 1$), atunci valorile calculate prin extrapolare sunt susceptibile de erori de aproximare mari. Pentru a evita erorile de extrapolare este preferabilă extinderea vectorului soluție pentru memorarea rezultatelor pe un număr de pași mai mare decât cel necesar în procesul de predicție. Dificultățile întâlnite fac ca utilizatorii metodelor multiple să fie descurajați la modificarea mărimii pasului.

Nu același lucru se poate afirma despre metodele cu valori multiple. Metoda Nordsieck a fost special concepută pentru a facilita modificarea pasului. Deoarece derivatele soluției la momentul curent nu se modifică prin modificarea pasului rezultă că trecerea de la vectorul Nordsieck pentru momentul t_k și pasul h :

$$z_k = \left[x_k, hx'_k, \frac{h^2}{2}x''_k, \dots, \frac{h^p}{p!}x_k^{(p)} \right]^T \quad (8.138)$$

la vectorul corespunzător aceluiași moment, dar cu pasul $h' = \alpha h$:

$$\hat{z}_k = \left[x_k, h'x'_k, \frac{h'^2}{2}x''_k, \dots, \frac{h'^p}{p!}x_k^{(p)} \right]^T \quad (8.139)$$

se realizează prin multiplicarea componentelor cu $1, \alpha, \alpha^2, \dots$ respectiv prin aplicarea transformării (8.137), în care matricea $C(\alpha)$ este diagonală:

$$C(\alpha) = \begin{bmatrix} 1 & & & & 0 \\ & \alpha & & & \\ & & \alpha^2 & & \\ & & & \ddots & \\ 0 & & & & \alpha^p \end{bmatrix}. \quad (8.140)$$

Usurința cu care se poate modifica pasul în metoda Nordsieck permite adaptarea în permanență a mărimii acestuia la valoarea optimă, corespunzătoare erorii impuse. Modificarea pasului de integrare poate fi făcută urmărind abaterea dintre predictor și corector, abaterea dintre două iterații succesive la corector sau reducerea la doi a numărului de iterații corectoare. Toate aceste criterii nu urmăresc direct eroarea de trunchiere a metodei și pot da indicații false pentru modificarea pasului. Gear propune un algoritm de control al ordinului și mărimii pasului, algoritm bazat pe controlul erorii de trunchiere. Eroarea locală de trunchiere cu metoda Adams-Moulton de ordin p poate fi exprimată sub forma:

$$\varepsilon_t = [C_p x^{(p+2)}(\tau)] h^{p+2} \quad (8.141)$$

dependența valorii derivate de ordin $(p+2)$ a soluției într-un punct cuprins în intervalul (t_k, t_{k+1}) , cu constantele C_p dependente de ordin:

$$\begin{aligned} C_0 &= -\frac{1}{2}, & C_1 &= -\frac{1}{12}, & C_2 &= -\frac{1}{24}, \\ C_3 &= -\frac{19}{720}, & C_4 &= -\frac{3}{160}, & C_5 &= -\frac{863}{60480}. \end{aligned} \quad (8.142)$$

Metoda Nordsieck bazată pe algoritmul Adams-Moullin de ordinul p folosește vectorul soluție:

$$z_k = \left[x_k, h x'_k, \frac{h^2}{2} x''_k, \dots, \frac{h^p}{(p+1)!} x_k^{(p+1)} \right]^T. \quad (8.143)$$

Derivata de ordin $(p+2)$ a soluției poate fi calculată aproximativ, prin diferența divizată, folosind ultima componentă $z_k^{(p+1)}$ a vectorului Nordsieck la două momente succesive de timp:

$$x^{(p+2)}(\tau) \approx \frac{1}{h} [x_k^{(p+1)} - x_{k-1}^{(p+1)}] = \frac{(p+1)}{h^{p+2}} [z_k^{(p+1)} - z_{k-1}^{(p+1)}] = \frac{(p+1)}{h^{p+2}} \nabla z_k^{(p+1)}. \quad (8.144)$$

În urma modificării pasului de timp la valoarea $h' = \alpha h$, eroarea de trunchiere estimată este:

$$\varepsilon_t = C_p \cdot (p+1)! \cdot \alpha^{(p+2)} \cdot \nabla z_k^{(p+1)}. \quad (8.145)$$

Această relație permite estimarea coeficientului α de modificare a mărimii pasului de timp, în vederea obținerii erorii de trunchiere dorite:

$$\alpha \approx \left[\frac{\varepsilon_{\max}}{C_p \cdot (p+1)! \cdot \nabla z_k^{(p+1)}} \right]^{\frac{1}{p+2}}. \quad (8.146)$$

Pentru a pondera măririle hazardate de pas, aceste rezultate se ponderează cu o constantă

subunitară. Pentru pasul $p = 0, 1, \dots, 5$ se propune folosirea unui coeficient optim:

$$\alpha_p = \frac{1}{1,2} \left[\frac{\varepsilon_{\max}}{C_p \cdot (p+1)! \cdot \nabla z_k^{(p+1)}} \right]^{\frac{1}{p+2}}. \quad (8.147)$$

Dacă simultan cu modificarea pasului are loc și o micșorare cu o unitate a ordinului, atunci coeficientul de modificare se calculează cu o relație asemănătoare:

$$\alpha_{p-1} = \frac{1}{1,3} \left[\frac{\varepsilon_{\max}}{C_{p-1} \cdot (p+1)! \cdot z_k^{(p+1)}} \right]^{\frac{1}{p+1}} \quad (8.148)$$

cu $p = 1, 2, \dots, 5$.

Majorarea cu o unitate a ordinului presupune adoptarea unui coeficient de modificare a pasului:

$$\alpha_{p+1} = \frac{1}{1,4} \left[\frac{\varepsilon_{\max}}{C_{p+1} \cdot (p+1)! \cdot \nabla^2 z_k^{(p+1)}} \right]^{\frac{1}{p+3}} \quad (8.149)$$

cu $p = 0, 1, \dots, 4$. Pentru calculul coeficientului ε_{p+1} este necesară evaluarea diferenței de ordinul doi:

$$\nabla^2 z_k^{(p+1)} = z_k^{(p+1)} - 2z_{k+1}^{(p+1)} + z_{k+2}^{(p+1)}. \quad (8.150)$$

Ordinul adoptat pe următorul pas se alege corespunzător valorii maxime din cele trei: α_{p-1} (scade ordinul curent), α_p (se păstrează ordinul), α_{p+1} (crește ordinul curent). Pasul nu este crescut, dacă ordinul curent trebuie menținut și $1 < \alpha_p < 1,1$. Pentru a evita modificarea prea frecventă a ordinului sau a mărimii pasului, Gear propune calculul coeficienților α , în vederea modificării pasului, doar în următoarele condiții:

1. eroarea de trunchiere a depășit valoarea maxim admisibilă:

$$\varepsilon_t = C_p \cdot (p+1)! \cdot \nabla z_k^{(p+1)} > \varepsilon_{\max},$$

caz în care calculul soluției la pasul curent de timp se reia;

2. s-au scurs cel puțin $(p+1)$ pași de timp de la ultima modificare a ordinului sau a pasului;

3. după 10 pași de la ultima estimare a parametrilor α care nu a dus la modificarea pasului.

8.8 Integrarea numerică a ecuațiilor de tip stiff

8.8.1 Ecuații diferențiale de tip stiff

În analiza numerică a sistemelor, inclusiv a circuitelor electrice și electronice se întâlnesc

situații în care metodele de discretizare prezentate anterior nu dau rezultate satisfăcătoare. Spre exemplu, se consideră circuitul simplu din figura 8.23 a, în care cele două bobine sunt parcurse la momentul $t = 0$ de curentul $i_0 = i_1(0) = -i_2(0)$. Se urmărește determinarea variației în timp a intensității curentului $i(t) = i_1(t) + i_2(t)$, care parcurge comutatorul K , închis la momentul $t = 0$.

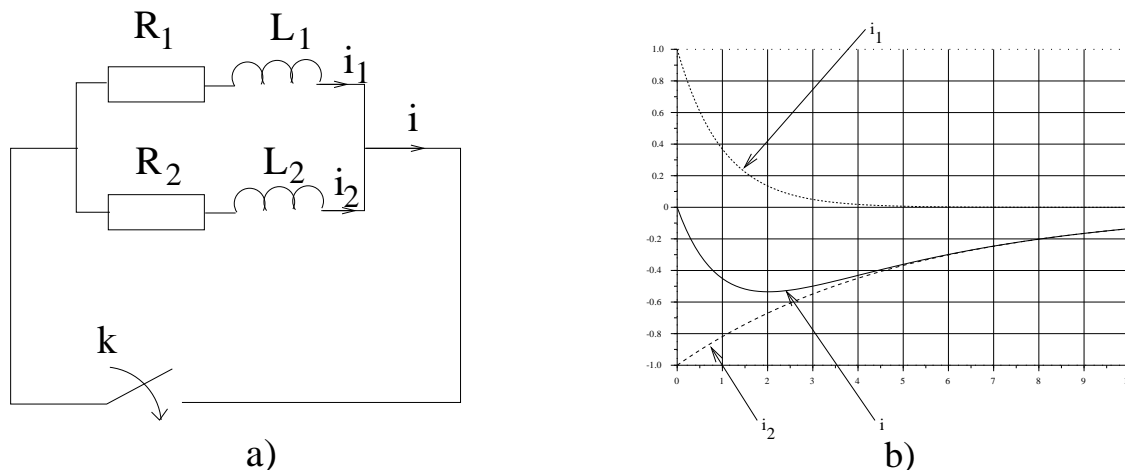


Figura 8.23: Circuit de tip stiff cu două constante de timp: a) Schema circuitului; b) Variația în timp a curenților i_1 , i_2 și i

Ecuțiile de stare asociate acestui circuit sunt:

$$\begin{aligned} \frac{di_1}{dt} &= -\frac{R_1}{L_1} i_1; \\ \frac{di_2}{dt} &= -\frac{R_2}{L_2} i_2. \end{aligned} \quad (8.151)$$

Folosind condițiile inițiale, rezultă soluțiile:

$$\begin{aligned} i_1(t) &= i_0 \cdot e^{-t/\tau_1}; \\ i_2(t) &= -i_0 \cdot e^{-t/\tau_2}; \\ i(t) &= i_0(e^{-t/\tau_1} - e^{-t/\tau_2}). \end{aligned} \quad (8.152)$$

în care cele două constante de timp au valorile:

$$\tau_1 = \frac{L_1}{R_1}, \quad \tau_2 = \frac{L_2}{R_2}. \quad (8.153)$$

Aceste funcții sunt reprezentate grafic în figura 8.15.b. Dacă se presupune că $\tau_2 \ll \tau_1$, de exemplu : $R_1 = R_2 = 1$ și $L_1 = 1$ H, $L_2 = 1$ mH, atunci intensitatea curentului $i(t)$ variază rapid în primele milisecunde și apoi relativ lent în următoarele secunde.

Rezolvarea acestei probleme cu o metodă numerică presupune alegerea unui pas de discretizare mai mic decât cea mai mică dintre constantele de timp, de exemplu:

$h = \tau_2/2 = 0,5$ ms, alegere rațională pentru studiul variației rapide din primele milisecunde. În schimb, pentru timpi mai mari de $5 \div 10$ s, această valoare a pasului este inefficientă, deoarece pentru integrare până la 5 s trebuie efectuați circa 1000 de pași. Dacă se adoptă pe acest interval o valoare a pasului mai mare de $2\tau_2 = 2$ ms, atunci rezultatele vor fi afectate de instabilități numerice puternice, datorate componentei rapid variabile, cu constanta de timp τ_2 , chiar dacă aceasta este practic "stinsă" pe acest interval.

Acest tip de circuite, caracterizate de constante de timp foarte diferite se numesc circuite de tip "stiff". Metodele clasice de integrare numerică nu se pot aplica la analiza numerică a circuitelor de tip stiff, deoarece ele impun valori ale pasului de timp nejustificat de mici. Dacă în exemplul anterior raportul celor două constante de timp τ_1/τ_2 nu are valoarea 1000, ci 10^6 sau 10^9 , atunci timpul de calcul pentru analiza acestui circuit simplu folosind un pas de integrare de ordinul τ_2 devine inacceptabil de mare. Dacă se presupune că, pentru un pas de timp, calculele durează circa $10\mu\text{s}$, atunci pentru analiza pe un interval de 1 s, timpul de calcul se ridică la $10^9 \cdot 10\mu\text{s} = 10^4\text{s} \approx 3$ ore.

În exemplul anterior, cele două componente se pot studia separat, deoarece matricea sistemului este de tip diagonal. Se pot da ușor exemple de circuite liniare caracterizate de ecuații de stare:

$$\frac{d\mathbf{x}}{dt} = A\mathbf{x} \quad (8.154)$$

de tip stiff. În acest caz valorile proprii ale matricei A au valori ale modulelor foarte diferite. Această caracteristică poate fi generalizată și pentru cazul circuitelor neliniare.

Un sistem neliniar, caracterizat de ecuația de stare:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t) \quad (8.155)$$

în care **matricea Jacobian** $J = [\partial\mathbf{f}/\partial\mathbf{x}]$ are valori proprii la care raportul dintre modulul maxim și cel minim are valori mult mai mari decât unitatea este un *sistem de tip stiff*.

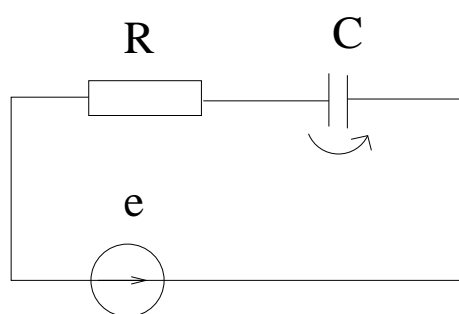
Efectele de tip stiff nu sunt legate numai de valorile proprii ale matricii caracteristice și se pot obține și pe seama excitației circuitului. De exemplu, un circuit cu o singură constantă de timp ca cel din figura (8.24)a, la care sursa variază lent în timp:

$$e(t) = E \cdot e^{-t/\tau_1}, \quad (8.156)$$

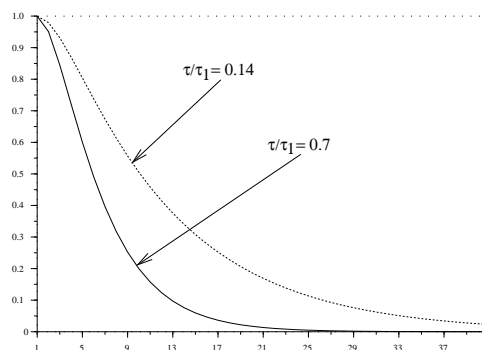
cu condiția inițială $u(0) = E$, are soluția:

$$u(t) = \frac{E}{1 - \tau/\tau_1}(e^{-t/\tau_1} - e^{-t/\tau}) + E \cdot e^{-t/\tau}, \quad \tau = RC \quad (8.157)$$

reprezentată grafic în figura (8.24)b. Dacă $\tau \ll \tau_1$, atunci analiza numerică ridică problemele menționate și deci, este de tip stiff.



a)



b)

Figura 8.24: Circuit de tip stiff cu o constantă de timp: a) Schema circuitului, cu $e(t) = E \cdot e^{-t/\tau_1}$; b) Variația în timp a tensiunii

Un caz special de probleme de tip stiff îl reprezintă **sistemele mixte de ecuații diferențiale și algebrice**. În acest caz ecuațiile algebrice pot fi privite ca ecuații diferențiale degenerate:

$$\begin{aligned} x' &= f(x, y, t) \\ \varepsilon y' &= g(x, y, t) \end{aligned} \quad (8.158)$$

în care parametrul $\varepsilon \rightarrow 0$. Valorile foarte mici ale parametrului ε corespund unor componente rapid variabile în timp, conform cărora soluția y a ecuațiilor algebrice urmărește practic instantaneu excitațiile.

Se constată că nu poate fi formulată o definiție exactă pentru *sistemele de ecuații diferențiale de tip stiff*, ci doar o caracterizare, prin afirmația că un astfel de sistem conține în soluția sa cel puțin două componente, una rapid variabilă și alta lent variabilă față de variabila independentă.

În analiza circuitelor electrice și electronice intervin deseori probleme de tip stiff, mai ales atunci când se introduc capacitățile și inductivitățile parazite, ale căror valori sunt mult mai mici decât cele utile. Modelele complexe ale dispozitivelor electronice conțin capacități sau rezistențe ale căror valori pot varia în domenii de $1/10^6$, pentru diferitele regiuni de funcționare ale componentei. Acestea fac ca un circuit care conține unul sau mai multe astfel de modele să prezinte efecte stiff.

8.8.2 Stabilitatea numerică a ecuațiilor stiff

Modul în care a fost anterior definită stabilitatea numerică nu este potrivit pentru abordarea problemelor stiff și trebuie dată o nouă definiție destinată acestei probleme speciale. Dintre definițiile posibile, cea mai larg acceptată este cea datorată lui Gear [26].

Definiția 8.3 O metodă numerică, care, aplicată ecuației test:

$$\frac{dx}{dt} = \lambda x, \quad (8.159)$$

cu $\lambda \in \mathbf{C}$, este absolut stabilă în regiunea $R_1 : \operatorname{Re}[\lambda h] \leq a \leq 0$ și este relativ stabilă în regiunea $R_2 : a < \operatorname{Re}[\lambda h] \leq b \leq 0, -\theta \leq \operatorname{Im}[\lambda h] \leq \theta$, se numește numeric stiff stabilă pentru pasul h . Cele două regiuni, R_1 și R_2 , ale planului complex \mathbf{C} sunt reprezentate grafic în figura 8.25.

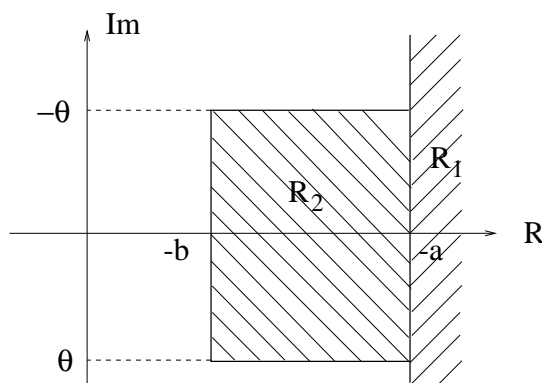


Figura 8.25: Regiuni de stabilitate ale unei metode numerice tip stiff pentru ecuația (8.159)

Analizând această definiție pentru cazul uzual al sistemelor stabile ($\operatorname{Re}[\lambda] = -1/\tau < 0$, $\operatorname{Im}[\lambda] = \omega$) rezultă că pentru pași de timp oricât de mari $h \geq a\tau$ ($\lambda h \in R_1$), metoda este absolut stabilă, ceea ce face ca eroarea propagată, corespunzătoare componentelor rapid variabile, să nu crească în timpul integrării. Mai mult, pe măsură ce valoarea acestor componente tinde către zero și eroarea asociată lor se anulează.

Pentru valori mici ale pasului de integrare $h < \min[a\tau, \theta/\omega]$, care corespund condiției $\lambda h \in R_2$, metoda este relativ stabilă, ceea ce asigură acuratețe numerică soluției obținute.

Faptul că R_2 se extinde și în semiplanul $\lambda h > 0$, permite abordarea și a problemelor instabile ($\operatorname{Re}[\lambda] = \alpha > 0$, $\operatorname{Im}[\lambda] = \omega$), caz în care pasul de timp trebuie să fie suficient de mic $h < \min[b/\alpha, \theta/\omega]$ pentru a obține soluții cu acuratețe acceptabilă și neafectate grav de instabilități numerice.

Analizând domeniile de stabilitate numerică absolută ale metodelor anterior definite se constată că regiunea R_1 este conținută doar în domeniile metodelor implicite Adams-Moulton cu $p = 0$ și $p = 1$. Doar metoda trapezelor și metoda Euler implicită satisfac condițiile de stabilitate numerică stiff, pentru care $a = 0$.

Dacă se abordează problema din figura 8.16 folosind metoda Euler explicită, atunci soluția numerică prezintă instabilități inacceptabile (figura 8.26a), pe când cu metoda Euler implicită se obțin chiar la pas mare de timp soluții numerice fără instabilități (figura 8.26b).

Dalquist a demonstrat că nu există metode multipas de ordin superior metodei trapezelor care să fie absolut stabile în semiplanul λh stâng (stiff numeric stabile cu $a = 0$).

Gear a găsit o clasă de metode numerice multipas de ordin superior, care sunt stiff numeric stabile pentru $a < 0$, pornind de la studiul comportării polinomului caracteristic (8.89) pentru $\lambda h \rightarrow -\infty$.

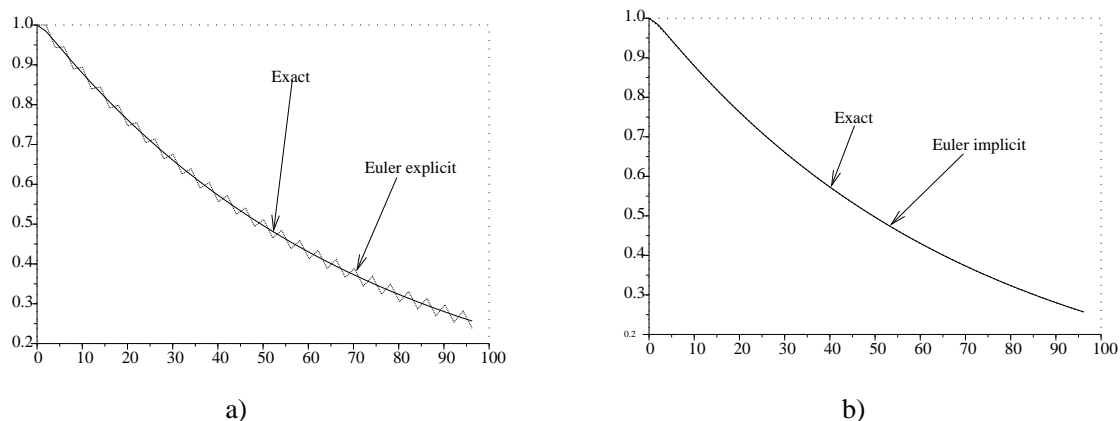


Figura 8.26: Soluțiile unei ecuații stiff prin metodele Euler explicită (a) și implicită (b)

Rădăcinile polinomului caracteristic $P(z)$ satisfac egalitatea:

$$\lambda h = -\frac{A(z)}{B(z)}, \quad (8.160)$$

iar dacă $\lambda h \rightarrow -\infty$, rezultă că:

$$\text{grad}B(z) \geq \text{grad}A(z) \quad (8.161)$$

deoarece în caz contrar punctul de la infinit este rădăcină a polinomului caracteristic și metoda își pierde stabilitatea numerică absolută. Din inegalitatea (8.161), rezultă că metodele multipas stiff sunt în mod necesar metode implicite ($b_0 \neq 0$). Nu pot exista metode explicite care să fie numeric stiff stabile.

Împărțind relația (8.89) la λh , rezultă că rădăcinile polinomului caracteristic satisfac:

$$\frac{P(z)}{\lambda h} = \frac{1}{\lambda h}A(z) + B(z) = 0, \quad (8.162)$$

sau la limită, pentru $\lambda h \rightarrow -\infty$, rezultă $B(z) = 0$, sau dezvoltat:

$$b_0 z^p + b_1 z^{p-1} + \dots + b_p = 0.$$

Pentru ca rădăcinile parazite să aibă modul minim la un pas oricât de mare, se alege:

$$b_1 = b_2 = \dots = b_p = 0, \quad (8.163)$$

ceea ce corespunde polinomului $B(z) = b_0 = z^p$ care are toate cele p rădăcini multiple și egale cu 0. Pe baza continuității rădăcinilor față de coeficienții unui polinom, este de așteptat ca și pentru alte valori ale lui λh rădăcinile parazite să aibă un modul cât mai mic. Cu aceste alegeri, rezultă relația recurentă de integrare numerică cu metoda Gear:

$$x_k = b_0 h \cdot f(x_k, t_k) + \sum_{i=1}^p a_i x_{k-i}. \quad (8.164)$$

Coeficienții b_0 și a_i se determină pe baza condițiilor de consistentă (8.74), (8.75):

$$\begin{aligned} \sum_{i=0}^p a_i &= 0; \\ \sum_{i=1}^p (1-i)^j a_i + j b_0 &= 0, \quad j = 1, 2, \dots, p, \end{aligned} \quad (8.165)$$

în care, dacă se consideră $a_0 = -1$, rezultă un sistem de $(p+1)$ ecuații:

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 0 \\ 0 & -1 & -2 & \dots & (p-1) & 1 \\ 0 & 1 & 4 & \dots & (p-1)^2 & 2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & (-1)^p & (-2)^p & \dots & (1-p)^p & p \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ b_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} \quad (8.166)$$

cu $(p+1)$ necunoscute: $b_0, a_1, a_2, \dots, a_p$.

Pentru cazul $p = 1$, se obține:

$$\begin{aligned} a_1 &= 1 \\ b_0 &= 1, \end{aligned}$$

ceea ce corespunde la $x_k = hf(x_k, t_k) + x_{k-1}$, care este chiar relația Euler implicită.

Dacă se consideră $p = 2$, se obține sistemul

$$\begin{aligned} a_1 + a_2 &= 1; \\ -a_2 + b_0 &= 1; \\ a_2 + 2b_0 &= 1, \end{aligned}$$

care are soluțiile: $a_1 = 4/3, a_2 = -1/3$ și $b_0 = 2/3$, ceea ce corespunde relației Gear de ordinul doi:

$$x_k = \frac{2}{3}h \cdot f(x_k, t_k) + \frac{1}{3}(4x_{k-1} - x_{k-2}). \quad (8.167)$$

În mod asemănător se determină coeficienții metodelor Gear de ordin superior, coeficienții ale căror valori sunt prezentate în tabelul 8.6:

Pentru determinarea domeniilor de stabilitate absolută se consideră mulțimea punctelor $z \in \mathbf{C}$ care au modul unitar: $z = e^{j\varphi}$ cu $j = \sqrt{-1}$. Dacă aceste puncte sunt rădăcini ale polinomului caracteristic, atunci locul geometric al punctelor:

$$s = \lambda h = -\frac{A(z)}{B(z)} = -\frac{A(e^{j\varphi})}{B(e^{j\varphi})} \quad (8.168)$$

reprezintă frontiera domeniului de stabilitate absolută din planul complex λh .

Pentru metodele Gear aceste curbe au ecuațiile:

$$s = -\frac{1}{b_0} \sum_{i=0}^p a_i e^{-ji\varphi}, \quad \in [0, 2\pi]. \quad (8.169)$$

Tabela 8.6: Coeficienții metodei Gear

p	ν	νb_0	νa_1	νa_2	νa_3	νa_4	νa_5	νa_6
1	1	1	1					
2	3	2	4	-1				
3	11	6	18	-9	2			
4	25	12	48	-36	16	-3		
5	137	60	300	-300	200	-75	12	
6	147	60	360	-450	400	-275	72	-10

În cazul $p = 1$, ecuația parametrică:

$$s(\varphi) = 1 - e^{j\varphi} \quad (8.170)$$

evidențiază un cerc cu centrul în $s_0 = 1$ și raza unitate. Punctele exterioare cercului determină rădăcini cu modul subunitar, de exemplu pentru $\lambda h = -1$,

$$P(z) = A(z) + \lambda h B(z) = -z + 1 + \lambda h z = z(\lambda h - 1) + 1, \quad (8.171)$$

$$z = -\frac{1}{\lambda h - 1} = \frac{1}{2}.$$

Rezultă că întregul plan complex minus discul circular menționat reprezintă domeniul de stabilitate absolută.

În cazul $p = 2$, ecuația frontierei domeniului de stabilitate este:

$$s(\varphi) = \frac{3}{2} - 2e^{j\varphi} + \frac{1}{2}e^{2j\varphi}, \quad (8.172)$$

care reprezintă o curbă închisă simetrică față de axa imaginară, ce trece prin origine și punctul $\lambda h = 4$. Si de această dată domeniul de stabilitate este exterior acestei curbe deoarece polinomul caracteristic:

$$P(z) = A(z) + \lambda h B(z) = -z^2 + \frac{4}{3}z - \frac{1}{3} + \lambda h \frac{2}{3}z^2$$

are rădăcinile:

$$z_1 = \frac{2 + \sqrt{1 + 2\lambda h}}{3 - 2\lambda h};$$

$$z_2 = \frac{2 - \sqrt{1 + 2\lambda h}}{3 - 2\lambda h}, \quad (8.173)$$

cu modul subunitar, de exemplu pentru $\lambda h = -1/2$, iar în cerc $z_1 = z_2 = 1/2$.

Domeniile de stabilitate absolută ale metodelor Gear de diferite ordine sunt prezentate în figura 8.27. Se constată că limita domeniului de stabilitate absolută are valorile $a = 0$, pentru ordinele unu și doi, iar pentru ordinele de la trei până la șase, aceste limite au

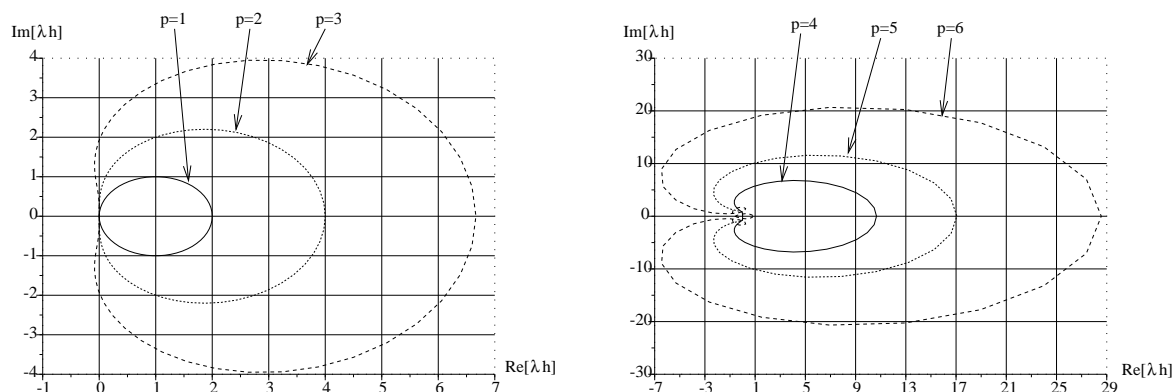


Figura 8.27: Domenii de stabilitate numerică pentru metodele Gear de diferite ordine p

valorile $a = -0.1, -0.7, -2.4$ și respectiv $a = -6.1$. Pentru ordine superioare lui 6 metodele Gear nu mai sunt numeric stiff stabile.

Dacă se reprezintă modulele celor două rădăcini $|z_1|$ și $|z_2|$ în funcție de parametrul λh presupus real se obține graficul din figura 8.28, care evidențiază caracterul numeric absolut stabil al metodelor Gear de ordinul doi, pentru $\lambda h \leq 0$ și numeric relativ stabil, pentru $0 < \lambda h < 3/2$.

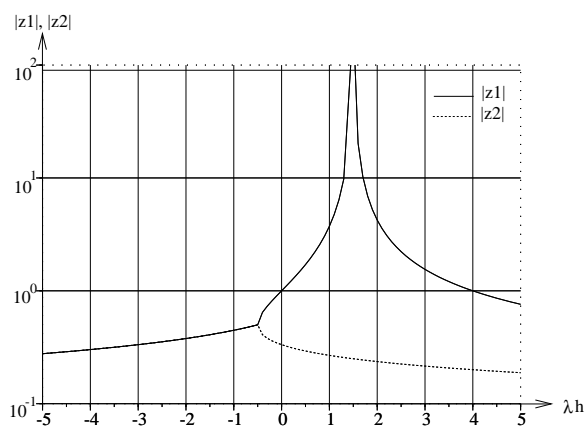


Figura 8.28: Dependența de λh a rădăcinilor polinomului caracteristic pentru metoda Gear de ordinul doi

În mod asemănător se demonstrează că pentru metodele Gear de ordine cuprinse între 1 și 6 există o regiune R_2 în vecinătatea originii, pentru care algoritmi sunt numeric relativ stabili. Rezultă că toate aceste metode se bucură de proprietatea stabilității numerice de tip stiff.

8.8.3 Algoritmul Gear pentru rezolvarea ecuațiilor stiff

Deoarece metodele Gear sunt metode multipas de tip implicit, la fiecare pas de integrare este necesară rezolvarea ecuației:

$$x_k = hb_0 \cdot f(x_k, t_k) + \sum_{i=1}^p a_i x_{k-i}. \quad (8.174)$$

Dacă pasul de integrare este suficient de mic:

$$h < \frac{1}{b_0 L_k}, \quad (8.175)$$

unde L_k este constanta Lipschitz a funcției $f(x, t)$ pentru $t = t_k$, atunci iterația simplă:

$$x_k^{(j+1)} = hb_0 \cdot f(x_k^{(j)}, t_k) + \sum_{i=1}^p a_i x_{k-i}, \quad (8.176)$$

este convergentă.

Deoarece în cazul ecuațiilor stiff pasul de timp poate fi mult mai mare decât constanta de timp a componentelor rapide, inegalitatea (8.175) nu este în general satisfăcută. Din acest motiv rezolvarea ecuației (8.174) prin metoda iterației simple, bazată pe relația (8.176) nu este folosită, fiind neconvergentă.

Ecuației neliniară implicită adusă sub forma:

$$x_k - hb_0 \cdot f(x_k, t_k) - \sum_{i=0}^p a_i x_{k-i} = 0 \quad (8.177)$$

se poate rezolva iterativ cu metoda Newton-Raphson:

$$x_k^{(j+1)} = x_k^{(j)} - [I - hb_0 \cdot J_f(x_k^{(j)}, t_k)]^{-1} \cdot \left\{ x_k^{(j)} - hb_0 \cdot f(x_k^{(j)}, t_k) - \sum_{i=1}^p a_i x_{k-i} \right\}, \quad (8.178)$$

în care I este matricea unitate de dimensiune $n \times n$, iar:

$$J_f(x_k^{(j)}, t_k) = \left. \frac{\partial f(x, t)}{\partial x} \right|_{x=x_k^{(j)}, t=t_k} \quad (8.179)$$

este matricea Jacobian a funcției $f(x, t)$, matrice de dimensiune $n \times n$.

Deoarece termenul care conține coeficienții a_1, a_2, \dots, a_p este constant pe parcursul iterațiilor, el poate fi eliminat. Dacă se scad relațiile corespunzătoare a două iterații succesive:

$$\begin{aligned} [I - b_0 h \cdot J_f(x_k^{(j)}, t_k)] \cdot [x_k^{(j+1)} - x_k^{(j)}] &= -x_k^{(j)} + hb_0 \cdot f(x_k^{(j)}, t_k) + \sum_{i=1}^p a_i x_{k-i} \\ [I - b_0 h \cdot J_f(x_k^{(j-1)}, t_k)] \cdot [x_k^{(j)} - x_k^{(j-1)}] &= -x_k^{(j-1)} + hb_0 \cdot f(x_k^{(j-1)}, t_k) + \sum_{i=1}^p a_i x_{k-i} \end{aligned}$$

rezultă:

$$[I - b_0 h \cdot J_f(x_k^{(j)}, t_k)] \cdot [x_k^{(j+1)} - x_k^{(j)}] = h b_0 [f(x_k^{(j)}, t_k) - f(x_k^{(j-1)}, t_k)] - h b_0 \cdot J_f(x_k^{(j-1)}, t_k) \cdot [x_k^{(j)} - x_k^{(j-1)}], \quad (8.180)$$

în care dacă se notează:

$$d_k^{(j)} = h \cdot f(x_k^{(j-1)}, t_k) + h \cdot J_f(x_k^{(j-1)}, t_k) [x_k^{(j)} - x_k^{(j-1)}], \quad (8.181)$$

rezultă:

$$b_0 [I - b_0 h \cdot J_f(x_k^{(j)}, t_k)] \cdot [x_k^{(j+1)} - x_k^{(j)}] = h x_k'^{(j)} - d_k^{(j)}, \quad (8.182)$$

deoarece

$$x_k'^{(j)} = f(x_k^{(j)}, t_k).$$

Scăzând două valori succesive pentru $d_k^{(j)}$ și $d_k^{(j+1)}$ se constată că și acestea satisfac o relație de recurență asemănătoare:

$$[I - b_0 h \cdot J_f(x_k^{(j)}, t_k)] \cdot [d_k^{(j+1)} - d_k^{(j)}] = h x_k'^{(j)} - d_k^{(j)}. \quad (8.183)$$

Aplicarea acestor relații iterative permite determinarea limitelor șirurilor:

$$\begin{aligned} x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(j)}, \dots &\rightarrow x_k; \\ d_k^{(1)}, d_k^{(2)}, \dots, d_k^{(j)}, \dots &\rightarrow d_k = h x_k', \end{aligned} \quad (8.184)$$

pornind de la inițializarea:

$$d_k^{(0)} = \frac{1}{b_0} \left[x_k^{(0)} - \sum_{i=1}^p a_i x_{k-i} \right], \quad (8.185)$$

obținută prin identificarea relațiilor (8.176) și (8.181) pentru $j = 0$.

Pentru ca schimbarea pasului de integrare să se facă ușor, metoda Gear se implementează în reprezentarea Nordsieck (121):

$$z_k = \left[x_k, h x_k', \frac{h^2}{2} x_k'', \dots, \frac{h^p}{p!} x_k^{(p)} \right]^T.$$

În acest caz, relația de predicție are reprezentarea matriceală:

$$z_k^{(0)} = Z z_{k-1}, \quad (8.186)$$

în care matricea Z este de tip triunghi Pascal, iar relația de corecție este:

$$z_k^{(j+1)} = z_k^{(j)} + G(z_k^{(j)}) \cdot c_z. \quad (8.187)$$

Se constată că $d_k^{(j)}$ fiind o aproximare a lui $x_k'^{(j)}$, poate fi preluat din componenta secundă a vectorului Nordsieck.

Forma matriceală a corectorului poate fi aplicată și în cazul iterației Newton-Raphson, dar cu observația că funcția de corecție G are în acest caz un caracter matriceal:

$$G(z_k^{(j)}) = [I - hb_0 \cdot J_f(x_k^{(j)}, t_k)]^{-1} \cdot [hx_k'^{(j)} - d_k^{(j)}]. \quad (8.188)$$

Pentru determinarea vectorului de corecție c_z trebuie determinată matricea de transformare T . Se consideră spre exemplificare cazul $p = 2$. Fie polinomul de grad doi:

$$x(t) = a_0 + a_1t + a_2t^2$$

cu derivatele:

$$x' = a_1 + 2a_2t; \quad x'' = 2a_2,$$

care pe o rețea regulată $t_k = 0, t_{k-1} = -h$ are vectorul Gear canonic:

$$y_k = [x_k, x_{k-1}, hx_k']^T = [a_0, a_0 - 2a, h + 2a_2h^2, ha_1]^T,$$

iar vectorul Nordsieck:

$$z_k = \left[x_k, hx_k', \frac{h^2}{2}x_k'' \right]^t = [a_0, ha_1, h^2a_2]^t.$$

Transformarea de echivalență între cei doi vectori este:

$$y_k = \begin{bmatrix} x_k \\ x_{k-1} \\ hx_k' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -2 & 2 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ ha_1 \\ h^2a_2 \end{bmatrix} = T^{-1}z_k.$$

Prin inversare rezultă:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -2 & 2 \\ 0 & 1 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix}, \quad (8.189)$$

matrice care permite calculul vectorului de corecție:

$$c_z = Tc_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2/3 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2/3 \\ 3/3 \\ 1/3 \end{bmatrix}, \quad (8.190)$$

pornind de la $c_y = [b_0 \quad 0 \quad 1]^T$.

Valorile componentelor vectorilor de corecție c_z , pentru diferite ordine, sunt date în tabelul 8.7:

Tabela 8.7: Componentele vectoriale de corecție

p	i =	1	2	3	4	5	6	7
2	3 $c_z(i)$	2	3	1				
3	11 $c_z(i)$	6	11	6	1			
4	50 $c_z(i)$	24	50	35	10	1		
5	274 $c_z(i)$	170	274	225	85	15	1	
6	1764 $c_z(i)$	720	1724	1624	735	175	21	1

Comparând cele două variante ale algoritmilor predictor-corector în reprezentarea Nordsieck se constată o mare asemănare între ele. Deosebirea de principiu constă în faptul că la metoda Gear, fiecare iterație a corectorului presupune calculul matricii Jacobian J_f și rezolvarea unui sistem de ecuații de dimensiunea acestei matrici, pe când la metodele de tip Adams este necesară doar evaluarea funcțiilor $f(x, t)$. Efortul de calcul suplimentar este “tributul” plătit pentru a asigura stabilitatea numerică a ecuațiilor stiff și pentru a putea avansa cu un pas de integrare oricât de mare după ce componentele rapid variabile s-au “stins”. Se constată că matricea $I - hb_0 J_f$ nu trebuie neapărat inversată, deoarece în majoritatea cazurilor această matrice nu variază prea mult de la o iterație la alta, motiv pentru care ea poate fi presupusă constantă și inversată o singură dată.

Tehnicile aplicate pentru controlul automat al ordinului și al pasului de integrare, prezentate în paragraful anterior pot fi aplicate și în acest caz.

În metoda Gear, eroarea de trunchiere satisface egalitatea:

$$\varepsilon = [C_p \cdot \hat{x}^{(p+1)}(\tau)] h^{p+1} = O(h^{p+1}),$$

în care coeficienții au valorile:

$$c_1 = -\frac{1}{2}, \quad c_2 = -\frac{2}{9}, \quad c_3 = -\frac{3}{22},$$

$$c_4 = -\frac{12}{125}, \quad c_5 = -\frac{10}{137}, \quad c_6 = -\frac{60}{1029}.$$

În consecință, în reprezentarea Nordsieck, eroarea de trunchiere poate fi calculată cu o relație de forma (8.139), dar această eroare se poate calcula eficient, din punctul de vedere al programatorului, cumulând corecțiile ce trebuie aplicate soluției $x_k^{(j)}$, pe parcursul iterațiilor, respectiv pentru M suficient de mare:

$$c_z = \sum_{j=0}^M G(z_k^{(j)}) = z_k^{(M)} - z_k^{(0)} = z_k - z_k^{(0)},$$

iar dacă se reține ultima componentă:

$$c_{z(p+1)} \sum_{j=0}^M G(z_k^{(j)}) = z_{k(p)} - z_{k(p)}^{(0)} = z_{k(p)} - z_{k-1(p)} = \frac{hp}{p!} (x_k^{(p)} - x_{k-1}^{(p)}) \cong \frac{h^{p+1}}{p!} x_k^{(p+1)},$$

de unde rezultă că:

$$\varepsilon = c_p \cdot p! \cdot c_{z(p+1)} \sum_{j=0}^M G(z_k^{(j)}).$$

8.8.4 Alte metode numeric stiff stabile

În afara metodelor de tip Gear și a metodei trapezelor, mai există și alte metode de discretizare numeric stiff stabile. Dintre acestea menționăm varianta implicită a metodelor Runge - Kutta. Varianta cea mai simplă a acestor metode, numită *a punctului median* se bazează pe relațiile:

$$\begin{aligned} g_1 &= h \cdot f\left(x_k + \frac{1}{2}g_1, t_k + \frac{1}{2}h\right) \\ x_{k+1} &= x_k + g_1. \end{aligned} \quad (8.191)$$

Se constată că incrementul g_1 , care trebuie aplicat soluției pentru a avansa cu un pas de timp, se determină prin rezolvarea unei ecuații neliniare implicite, corespunzătoare momentului de timp median $t_k + h/2$.

Metoda Runge - Kutta implicită cu două “evaluări” are forma generală:

$$\begin{aligned} x_{k+1} &= x_k + ha_1g_1 + a_2g_2, \quad \text{cu} \\ g_1 &= f(x_k + b_1g_1 + b_2g_2, t_k + c_1h); \\ g_2 &= f(x_k + b_3g_3 + b_4g_4, t_k + c_2h). \end{aligned} \quad (8.192)$$

Pentru următoarea alegere a coeficienților:

$$\begin{aligned} a_1 &= a_2 = 1/2; & b_2 &= 1/4 - \sqrt{3}/6; \\ b_1 &= b_4 = 1/4; & b_3 &= 1/4 + \sqrt{3}/66; \\ c_1 &= 1/2 - \sqrt{3}/6; & c_2 &= 1/2 + \sqrt{3}/6, \end{aligned}$$

metoda asigură o eroare globală de trunchiere de tipul $O(h^4)$, deci este de ordinul patru. Deoarece la fiecare moment de timp trebuie rezolvat un sistem de ecuații neliniare de dimensiune dublă față de cel din varianta anterioară, efortul de calcul asociat acestei metode este foarte mare.

Pentru a evita rezolvarea ecuațiilor neliniare, Rosenbrock propune o categorie de metode bazate pe calculul matricei Jacobian $J_f = \frac{\partial f}{\partial x}$. Varianta Calahan, cu două etape a acestei metode:

$$\begin{aligned} g_1 &= h \cdot f(x_k, t_k) + hb_1J_{f_k}g_1 \\ g_2 &= h \cdot f(x_k + b_2g_1, t_k + c_1h) + hb_1J_{f_k}g_2 \\ x_{k+1} &= x_k + a_1g_1 + a_2g_2 \end{aligned}$$

cu $a_1 = 3/4$, $a_2 = 1/4$, $b_1 = (1 + 1/\sqrt{3})/2$, $b_2 = -2\sqrt{3}$ asigură o eroare globală de tipul $O(h^3)$, deci este de ordinul 3.

Toate aceste metode conțin semiplanul stâng $\operatorname{Re} [\lambda h] < 0$ în domeniul de stabilitate numerică absolută, deci sunt numeric stiff stabile cu parametrul $a = 0$ [26].

O altă categorie de metode, cunoscută sub numele de *metode cu diferențiere regresivă*, se obțin, în principiu, prin rescrierea relației Gear (8.164) sub forma:

$$x'_k = \frac{1}{hb_0} \left[a_0 x_k - \sum_{i=1}^p a_i x_{k-i} \right]. \quad (8.193)$$

Constatând că relația (8.193) este o relație de derivare numerică cu aproximare polinomială regresivă, pentru a permite manipularea pasului variabil, Brayton propune folosirea relației de interpolare cu pași inegali, ca relație de corecție:

$$x'_k = \frac{1}{h} \sum_{i=0}^p a_i x_{k-i}, \quad (8.194)$$

în care $h = t_k - t_{k-1}$:

$$a_i = \frac{t_k - t_{k-1}}{t_k - t_{k-i}} \prod_{\substack{j=1 \\ j \neq i}}^p \frac{t_k - t_{k-i}}{t_{k-i} - t_{k-j}}, \quad j = 1 \dots p \quad \text{și} \quad \sum_{j=0}^p a_j = 0.$$

Pentru predictor se propune o relație de același ordin:

$$x_k^{(0)} = \sum_{i=0}^{p+1} c_i x_{k-i} \text{ cu } a_i = \frac{t_k - t_{k-1}}{t_k - t_{k-i}} \prod_{\substack{j=1 \\ j \neq i}}^p \frac{t_k - t_{k-i}}{t_{k-i} - t_{k-j}}, \quad j = 1, \dots, p.$$

Eroarea locală de trunchiere este dată de:

$$\varepsilon_k = \frac{h}{t_k - t_{k-p-1}} (x_k - x_k^{(0)}). \quad (8.195)$$

Printr-o transformare de echivalență, derivata se poate exprima în funcție de diferențele finite de ordinul întâi, așa cum propune Branin:

$$x'_k = \sum_{i=0}^p \frac{a_i}{h_i} \nabla x_{k-i}, \quad (8.196)$$

în care: $h_i = t_k - t_{k-p}$, $a_i = \sum_{j=0}^i p_j$, iar coeficienții p_j sunt dați de: $\sum_{j=0}^p p_j = 0$, $\sum_{j=1}^p d_j p_j = -1$,
în care $d_j = h_j/h_1$.

În acest caz eroarea locală de trunchiere este:

$$\varepsilon = \frac{h^{p+1} x_k^{(p+1)}}{p+1} \prod_{j=1}^p \frac{d_j}{j}.$$

O reprezentare, asemănătoare celei Nordsieck, dar folosind diferențe finite, este propusă de Ruhner-Petersen:

$$x'_{k+1} = \frac{1}{h} \sum_{i=0}^{p-1} a_i b_i \nabla^i x_k. \quad (8.197)$$

unde

$$a_i = \sum_{j=p-1}^i \frac{h_1}{t_{k+1} - t_{k-j}}, \quad b_i = \prod_{j=1}^i \frac{t_{k+1} - t_{k+1-j}}{t_k - t_{k-j}}, \quad b_0 = 1.$$

Relația de predicție corespunzătoare este:

$$x_{k+1}^{(0)} = \sum_{i=0}^p b_i \nabla^i x_k. \quad (8.198)$$

Aceste trei metode fiind echivalente, au aceeași eroare de trunchiere și sunt numeric stiff stabile pentru $1 \leq p \leq 6$. Fiind metode implicite, soluția se determină prin iterații Newton-Raphson, iar modificarea pasului și a ordinului se poate face prin algoritmul Gear, deoarece sunt cunoscute estimări ale erorii.

Față de metoda Gear, aceste metode oferă o stabilitate numerică mai bună, în cazul schimbării frecvente a mărimii pasului, dar ele impun calcularea coeficienților a_i la fiecare pas de integrare.

8.9 Analiza numerică a circuitelor electrice în regim tranzitoriu

Prin analiza numerică a circuitelor electrice în regim tranzitoriu se urmărește determinarea variației în timp a curenților prin laturile unui circuit cu rezistoare, bobine și condensatoare, alimentate de la surse de tensiune sau curent variabile în timp.

Pentru rezolvarea numerică a ecuațiilor diferențiale asociate unui astfel de circuit, intervalul de timp al analizei $[t_0, t_{\max}]$, este împărțit într-o rețea de discretizare. În fiecare nod al rețelei problema se reduce la rezolvarea unui circuit electric rezistiv, prin înlocuirea elementelor acumulator de energie (bobine și condensatoare) din circuitul inițial, cu modele echivalente. Aceste modele, alcătuite din rezistențe și surse, rezultă din discretizarea ecuației diferențiale ce caracterizează funcționarea elementului respectiv și au un caracter rezistiv. Dacă circuitul electric este neliniar, atunci modelul discretizat prezintă un circuit rezistiv neliniar, pentru analiza căruia se aplică de obicei metode iterative.

8.9.1 Principiul metodei

Se consideră un circuit electric cu L laturi. Fiecare latură poate conține un element pasiv (R , L , sau C) și în serie o sursă. Pentru descrierea circuitului sunt necesare următoarele informații:

- topologia circuitului electric - în acest scop fiecărei laturi de circuit i se asociază un nod inițial ni și un nod final nf , sensul de referință pentru curent, tensiunea la borne fiind de la nodul inițial către nodul final;
- tipul elementului pasiv și parametrii caracteristici ai elementelor pasive din fiecare latură R , L , sau C (pentru care nu pot avea valori nule);
- parametrii caracteristici ai surselor E (pentru care pot avea și valori nule);
- condițiile inițiale în cazul acumulatorilor de energie (tensiunile inițiale pe condensatoare u_{C_0} , respectiv curenții inițiali prin bobine i_{L_0}).

Intervalul de timp $[t_0, t_{\max}]$ este împărțit într-o rețea de discretizare uniformă cu pasul h și n noduri t_k :

$$t_0 = 0; t_1 = h; t_2 = 2h; \dots t_n = nh = t_{\max}. \quad (8.199)$$

Se consideră o latură ce conține o bobină de inductivitate L . La momentul t_k , curentul prin bobină este i_k , iar tensiunea la bornele sale este:

$$u_k = L \frac{di}{dt}. \quad (8.200)$$

Prin discretizare cu metoda Euler implicită, expresia (8.200) devine:

$$u_k = L \frac{i_k - i_{k-1}}{t_k - t_{k-1}}. \quad (8.201)$$

de unde:

$$u_k = i_k \frac{L}{h} - i_{k-1} \frac{L}{h}. \quad (8.202)$$

Ecuația (8.202) reprezintă tensiunea la bornele unei laturi ce conține un rezistor de rezistență $R_k = L/h$ și o sursă de tensiune de valoare

$$E_k = R_k i_{k-1},$$

orientată după sensul de referință al laturii.

Folosind același tip de discretizare, curentul printr-un condensator, la momentul t_k este:

$$i_k = C \frac{u_k - u_{k-1}}{h}, \quad (8.203)$$

cea ce este echivalent cu o conductanță $G_k = C/h$ în paralel cu o sursă de curent $J_k = G_k u_{k-1}$. Sursa echivalentă de tensiune a acestei grupări paralel are tensiunea electromotoare $E_k = -u_{k-1}$ și o rezistență internă $R_k = 1/G_k = h/C$. Prin urmare, pentru un condensator:

$$u_k = i_k \frac{h}{C} + u_{k-1}. \quad (8.204)$$

Circuitele R, E echivalente la fiecare pas cu elementele acumulative de energie L, C se numesc circuite discretizate.

Pornind de la condițiile inițiale i_0 , respectiv u_0 se pot determina succesiv potențialele nodurilor, v_k și curenții prin laturi i_k la orice moment $t_k (k = 1, 2, 3, \dots)$ prin rezolvarea unui circuit rezistiv (R, E) , constituit din circuite discretizate.

Cea mai simplă metodă de rezolvare a acestui circuit este metoda nodală care presupune rezolvarea următorului sistem de ecuații liniare:

$$Gv = i_s, \quad (8.205)$$

unde G este matricea conductanțelor, v este vectorul coloană al potențialelor nodurilor, iar i_s este vectorul coloană al curenților de scurtcircuit corespunzători fiecărui nod.

8.9.2 Pseudocodul metodei modelului discretizat

Următorul pseudocod descrie metoda de analiză a circuitelor electrice liniare de tip R, L, C , excitate cu surse de tensiune treaptă în regim tranzitoriu.

```

; Descriere circuit
citește L           ; număr de laturi
citește N           ; număr de noduri
pentru k = 1,L
    citește ni(k), nf(k) ; noduri inițial, final ale
                        ; laturii k
    citește tip(k)     ; tipul laturii k (R,L sau C)
    citește par(k)     ; parametrul elementului pasiv
                        ; (R(K), L(k) sau C(k))
    citește e(k)       ; t.e.m. din latura k
    dacă tip(k) ≠ R
        citește ci(k) ; condiția inițială din
                        ; latura k
    citește tmax       ; timpul maxim al simulării
    citește h           ; pasul de simulare
t = 0
; Parcurge timpul de simulare
cât timp t ≤ tmax

```

```

t = t + h
; Generează circuit discretizat
pentru k = 1,L
    dacă tip(k) = R atunci
        R(k) = par(k)
        E(k) = e(k)
    altfel dacă tip(k) = L atunci
        R(k) = par(k)/h
        E(k) = e(k) + R(k) · ci(k)
    altfel
        R(k) = h/par(k)
        E(k) = e(k) - ci(k)
; Rezolvă circuit rezistiv prin metoda nodală și
; determină potențialele V
nodal ( L, N, ni, nf, R, E, V )
; Actualizează condiții inițiale și afișează soluția
pentru k = 1,L
    u = V(ni(k)) - V(nf(k))
    i = (E(k) + u)/R(k)
    dacă tip(k) = L atunci
        ci(k) = i
    altfel ci(k) = u - e(k)
scrie k, u, i

```

8.9.3 Analiza algoritmului

Efort de calcul

Efortul de calcul este determinat în cea mai mare măsură de rezolvarea, la fiecare pas, a circuitului rezistiv prin metodă nodală. Funcția “**nodal**” generează matricele G și i_s , cu un efort de calcul de ordinul $O(6L)$ și rezolvă sistemul liniar (8.205) cu un efort de calcul de ordinul $O(N^3/3)$. Dacă se notează $n = t_{\max}/h$ numărul pașilor de timp, efortul de calcul global este de ordinul $O(n \cdot N^3/3)$ și constă în principal în rezolvarea de n ori a unui sistem liniar cu o dimensiune egală cu numărul nodurilor circuitului.

O reducere substanțială a timpului de calcul se poate obține, dacă se observă că la pasul h constant, rezistențele echivalente din circuitele discretizate au aceleași valori la toți pașii de timp, deci matricea G rămâne constantă în tot timpul simulării. În consecință matricea G poate fi generată o singură dată și factorizată LU , urmând ca la fiecare pas de timp să se determine potențialele prin rezolvarea sistemului $LUV = i_s$. În acest fel, timpul global de calcul capătă ordinul $O(N^3/3 + nN^2)$.

Analiza erorilor

Erorile soluției numerice se datoresc următoarelor cauze:

- erori inerente, datorate impreciziei datelor de intrare;
- erori de trunchiere, datorate aproximării derivatelor prin diferențe finite;
- erori de rotunjire, datorate reprezentării finite a numerelor reale apărute la rezolvarea sistemului liniar.

Prin utilizarea metodei implicite, în exprimarea derivatelor, se obține o stabilitate numerică bună.

În figura 8.30 sunt reprezentate grafic soluțiile de regim tranzitoriu ale circuitului liniar din figura 8.29, obținute în urma aplicării algoritmului prezentat anterior.

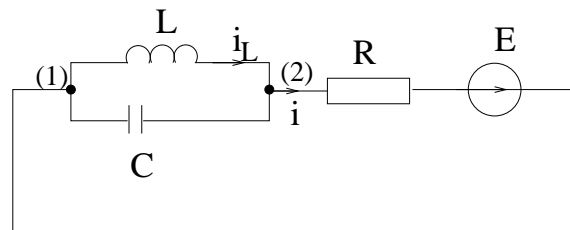


Figura 8.29: Circuit liniar în regim tranzitoriu

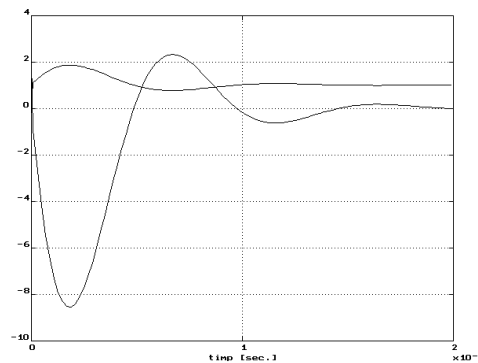


Figura 8.30: Soluția circuitului 8.29: variația în timp a curentului i și a potențialului v_1

Prin combinarea algoritmului de rezolvare a circuitelor în regim tranzitoriu cu cel de rezolvare a circuitelor neliniare (prezentat în paragraful ???) se obține un algoritm care permite rezolvarea circuitelor neliniare în regim tranzitoriu. Aplicarea acestui algoritm la rezolvarea unui circuit de tip redresor monoalternanță cu filtraj capacitiv (figura 8.31a) se obțin formele de variație ale curentului prin rezistor și tensiuni la bornele sarcinii prezentate în figura 8.32.

Variația aceluiași mărimi pentru circuitul redresor fără filtraj (figura 8.31b), este prezentată în figura 8.33.

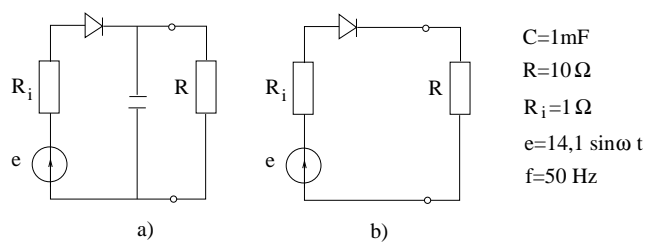


Figura 8.31: Circuite redresoare monoalternanță: a) cu filtraj capacitiv; b) fără filtraj .

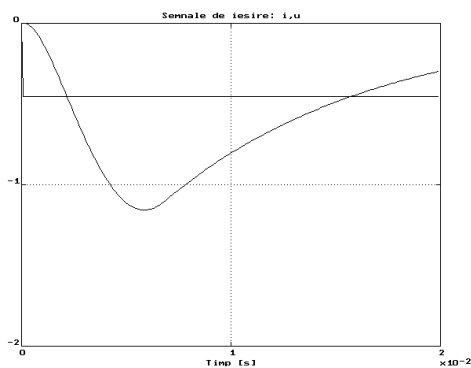


Figura 8.32: Soluția numerică a circuitului 8.31a

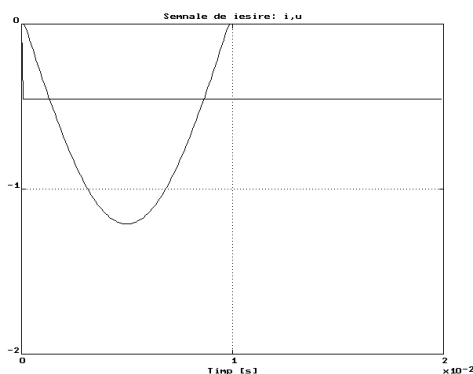


Figura 8.33: Soluția numerică a circuitului 8.31b

Anexa A

Bibliografie

- [1]
- [2] ***. *System / 360 Scientific Subroutine Package*. IBM.
- [3] Jr A. N. Wilson, editor. *Nonlinear Networks: Theory and analysis*. IEEE Press, NY, 1975.
- [4] R. K. Brayton, F. G. Gustavson și G. D. Hachtel. A new efficient algorithm for solving differential-algebraic systems using implicit backward differentiation formulas. *Proc. of the IEEE*, vol. 60, pp. 98–108, 1972.
- [5] R. P. Brent. On the Davidenko-Bronin methods for solving simultaneous nonlinear equations. *IBM J. Res. Develop.*, pp. 434–436, iul. 1972.
- [6] C. G. Broyden. A class for solving nonlinear simultaneous equations. *Math. Comp.*, vol. 19, pp. 577–593, 1965.
- [7] C. G. Broyden. A new solving nonlinear simultaneous equations ????. *Comput. J.*, vol. 12, pp. 94–99, feb. 1969.
- [8] D. A. Calahan. Numerical considerations for implementation of a nonlinear transient analysis program. *IEEE Trans. on CT*, vol. 18, nr. 1, ian 1971.
- [9] D. A. Calahan. *Computer-Aided Network Design*. Mc.Graw-Hill, New York, 1972.
- [10] B. H. Carnahan, H. A. Luther și J. O. Wilkes. *Applied Numerical Methods*. John Wiley, NJ, 1971.
- [11] K. S. Chao, D. K. Liu și C. E. Pan. A systematic search method for obtaining multiple solutions of simultaneous nonlinear equation. *IEEE Trans. on CAS*, vol. 22, nr. 9, pp. 748–752, sept. 1975.
- [12] M. J. Chien. Searching for multiple solutions of nonlinear systems. *IEEE Trans. on CAS*, vol. 26, nr. 10, pp. 817–827, oct. 1979.
- [13] M. J. Chien și E. Kuh. Solving nonlinear resistive networks using piecewise-linear analysis and simplicial subdivision. *IEEE Trans. on CAS*, vol. 24, nr. 6, pp. 305–317, iun. 1977.

- [14] I. O. Chua și N. N. Wang. A new approach to overcome the overflow problem in computer-aided analysis of nonlinear resistive circuits. *J. Of Circuit Theory and Applications*, vol. 3, pp. 261–284, 1975.
- [15] L. O. Chua și Y. F. Lem ??? Global homeomorfism of vector-valued functions. *Journal of Mathematical Analysis and Applications*, vol. 39, pp. 600–634, sept. 1972.
- [16] L. O. Chua. *Introduction to nonlinear network theory*. McGraw-Hill, NY, 1969.
- [17] L. O. Chua. Efficient computer algorithms for piecewise-linear analysis of resistive nonlinear networks. *IEEE Trans. on CT*, vol. 18, nr. 1, pp. 73–85, ian 1971.
- [18] L. O. Chua și A. C. Deng. Canonical piecewise-linear modelling. *IEEE Trans. on CAS*, vol. 33, nr. 5, pp. 511–525, mai 1986.
- [19] L. O. Chua și A. C. Deng. Canonical piecewise-linear representation. *IEEE Trans. on CAS*, vol. 35, nr. 1, pp. 101–111, ian 1988.
- [20] L. O. Chua și P. M. Lin. *Computer-aided analysis of electronic circuits*. Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [21] H. Crowder și P. Wolfe. Linear convergence of the conjugate gradient method. *IBM J. Res. Develop.*, , nr. 7, pp. 431–433, iul. 1972.
- [22] W. S. Dorn și D. D. McCracken. *Metode numerice cu programe în Fortran IV*. Editura Tehnică, București, 1976. Traducere din limba engleză.
- [23] R. Fletcher și M. J. D. Powell. A rapidly convergent descendent method for minimization. *Comput. J.*, , nr. 6, pp. 163–168, 1963.
- [24] T. Fujisawa, E. Kuh și T. Ohtsuki. A sparse matrix method for analysis of piecewise-linear resistive networks. *IEEE Trans. on CAS*, vol. 19, nr. 6, pp. 571–584, nov. 1972.
- [25] C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs, NJ, 1971.
- [26] C. W. Gear. Simultaneous numerical solutions of differential algebraic equations. *IEEE Trans. on CT*, vol. 18, nr. 1, ian 1971.
- [27] P. E. Gill și W. Murry. Algorithms for the solution of the nonlinear least-square problems. *SIAM J. Numer. Anal.*, vol. 15, nr. 5, pp. 977–992, oct. 1978.
- [28] Jr. H. F. Bronin. Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *Widely Convergent Method for Finding Multiple Solutions of Simultaneous Nonlinear Equations*, , nr. IBM J. Res. Develop., pp. 504–522, sept. 1972.
- [29] G. D. Hachel, R. K. Brayton și F. G. Gustavson. The sparse tableau approach to network analisys and design. *IEEE Trans. on CT*, vol. 18, nr. 1, ian 1971.

-
- [30] A. Halanay. *Teoria calitativă a ecuațiilor diferențiale*. Ed. Academiei, București, 1963.
- [31] A. J. Jimenez și S. W. Director. New families of algorithms for solving nonlinear circuit equations. *IEEE Trans. on CAS*, vol. 25, nr. 1, pp. 1–7, ian 1978.
- [32] J. Katzenelson. An algorithm for solving nonlinear resistive networks. *Bell Syst. Tech. J.*, vol. 44, pp. 1605–1620, oct. 1965.
- [33] S. M. Lee și K. S. Choo. Multiple solutions of piecewise-linear resistive networks. *IEEE Trans. on CAS*, vol. 30, nr. 2, pp. 84–89, feb. 1983.
- [34] W. Liniger. Multistep and one-leg methods for implicit mixed differential and algebraic systems. *IEEE Trans. on CAS*, vol. 26, nr. 9, sept. 1979.
- [35] W. Liniger și F. Odeh. A-stable, accurate averaging of multistep methods for stiff differential equations. *IBM J. of Res. and Develop.*, , nr. 7, iul. 1972.
- [36] A. R. Newton și A. Sangiovanni-Vincentelli. Relaxation-based electrical simulation. *IEEE Trans. on CAD*, vol. 3, nr. 4, pp. 3308–3331, oct. 1984.
- [37] T. Ohtsuki și N. Hoshida. Dc analysis of nonlinear networks based on generalized piecewise-linear characterization. *IEEE Trans. on CT*, vol. 18, nr. 1, pp. 146–152, ian 1971.
- [38] J. M. Ortega și W. C. Rheinholdt. *Iterative solution of nonlinear equations in several variables*. Academic Press, NY, 1970.
- [39] V. C. Prasad și V. P. Prakash. Homeomorphic piecewise-linear resistive networks. *IEEE Trans. on CAS*, vol. 35, nr. 2, pp. 251–253, feb. 1988.
- [40] J. R. Rice. *Numerical Methods, Software and Analysis. IMSL Reference Edition*. McGraw Hill, NY, 1983.
- [41] A. Rolston și H. S. Wilf. *Mathematical Methods for Digital Computers*. John Wiley, New York, 1967.
- [42] A. E. Ruehli. *Circuit Analysis Simulation and Design*. Elseviers Science Pub., 1986.
- [43] G. Russel. *Computer aided tools for VLSI system design*. Peter Peregrims, Ltd., 1987.
- [44] I. W. Sandbery. Theorems on the computation of the transient response of nonlinear networks containing transistors and diodes. *Bell Syst. Techn. Journ.*, vol. 49, pp. 1739–1776, oct. 1970.
- [45] M. A. F. Schwarz. *Computer-Aided Design of Microelectronic Circuits and Systems*. Academic Press, London, 1987.
- [46] K. N. Staton și N. Talkdar. New integration algorithms for transient stability studies. *IEEE Trans. on PAS*, vol. 89, nr. 56, mai 1970.

-
- [47] A. H. Stroud. *Numerical Quadrature and Solutions of Ordinary Differential Equations*. Springer Verlag, New York, 1974.
- [48] B. Thomas. The Runge-Kutta methods. *Byte*, , nr. 4, apr. 1986.
- [49] R. Varga. *Matrix iterative analysis*. Prentice-Hall, Englewood Cliffs, N.J., 1962.
- [50] F. F. Wu și C. A. Desoer. Global inverse function theorem. *IEEE Trans. on CT*, , nr. 3, pp. 199–201, mar. 1972.
- [51] D. A. Zein. Solution of a set of nonlinear algebraic equations for general-purpose CAD programs. *EEE Circuits and Devices Magazine*, pp. 7–20, sept. 1985.