Derivarea numerică. Introducere în metoda diferenţelor finite.

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" Bucureşti, Facultatea de Inginerie Electrică

Suport didactic pentru disciplina *Algoritmi Numerici*, 2017-2018

Cuprins

Importanţa evaluării derivatelor

- Relaţii utile pentru evaluarea unor mărimi:

$$i(t) = -\frac{\mathrm{d}q_{\mathcal{D}_\Sigma}}{\mathrm{d}t}$$

$$u(t) = -\frac{\mathrm{d}\phi_{S_\Gamma}}{\mathrm{d}t}$$

- Evaluarea senzitivităţilor:

$$\frac{\partial \mathcal{F}}{\partial p}$$

Ideea: derivarea numerică se bazează pe interpolare.

- Ecuaţii sau sisteme de ecuaţii diferenţiale

$\Rightarrow$ Metoda diferenţelor finite

Formularea problemei - cazul cel mai simplu

Se dă funcţia $f : [a, b] \to \mathbb{R}$ (cunoscută prin date sau prin cod)
Se cere evaluarea numerică a derivatei $f'(x_0)$, unde $x_0 \in [a, b]$.
Matematic:

$$f'(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}. \tag{1}$$

Nu se poate aplica aceasta formulă:

- nu se poate trece la limita $\Rightarrow$ erori de trunchiere;
- nedeterminare 0/0 $\Rightarrow$ probleme numerice când x este ales prea aproape de x_0.

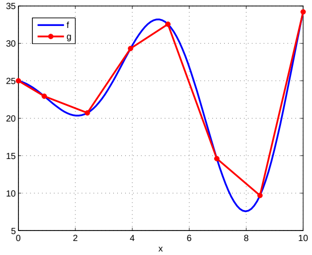
Ideea: $g(x)$ interpolarea sau aproximarea funcţiei şi

$$g'(x) = \frac{\mathrm{d}g}{\mathrm{d}x} \approx f'(x) = \frac{\mathrm{d}f}{\mathrm{d}x}. \tag{2}$$

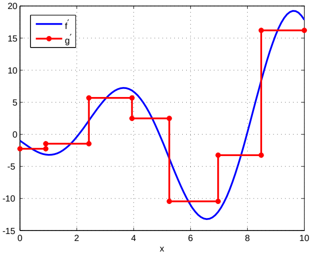
Precizia cu care se face interpolarea sau aproximarea va determina precizia de calcul a derivatei numerice.

Formule de derivare numerică - cu eroare de ordinul 1

g - interpolare liniară pe porţiuni.



Funcţia reală f şi interpolarea ei pe porţiuni g.



Aproximarea derivatei f' cu valoarea g'.

Formule de derivare numerică - cu eroare de ordinul 1

Derivare progresivă de ordinul 1:

$$y'_k = \frac{y_{k+1} - y_k}{x_{k+1} - x_k}, \tag{3}$$

Derivare regresivă de ordinul 1:

$$y'_k = \frac{y_k - y_{k-1}}{x_k - x_{k-1}}. \tag{4}$$

Ordinul erorii de trunchiere specifice acestei aproximări se poate estima cu ajutorul dezvoltării în serie Taylor.

Formule de derivare numerică - cu eroare de ordinul 1

Derivata progresivă $y'_k = (y_{k+1} - y_k)/(x_{k+1} - x_k)$

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{f''(\xi)}{2!}(x - x_k)^2 \qquad (5)$$

$x = x_{k+1} \Rightarrow$

$$y_{k+1} = y_k + f'(x_k)(x_{k+1} - x_k) + \frac{f''(\xi)}{2!}(x_{k+1} - x_k)^2, \qquad (6)$$

Eroarea de trunchiere

$$\frac{y_{k+1} - y_k}{x_{k+1} - x_k} - f'(x_k) = \frac{f''(\xi)}{2!}(x_{k+1} - x_k), \qquad (7)$$

Pp. $|f''(x)| \leq M_2$
Notăm $h = x_{k+1} - x_k$

$$|e_t| \leq \frac{M_2}{2} h, \qquad (8)$$

Formule de derivare numerică - cu eroare de ordinul 1

Derivata regresivă $y'_k = (y_k - y_{k-1})/(x_k - x_{k-1})$

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \mathrm{O}((x - x_k)^2) \tag{9}$$

$x = x_{k-1}$ şi notând $x_k - x_{k-1} = h$

$$y_{k-1} = y_k - f'(x_k)h + \mathrm{O}(h^2), \tag{10}$$

$$\frac{y_k - y_{k-1}}{h} - f'(x_k) = \mathrm{O}(h). \tag{11}$$

$|e_t|$ =O(h), eroare de ordin 1

Formule de derivare numerică - cu eroare de ordinul 2

Dezavantajul g - lpp: nu este derivabilă în noduri.
Remediu: creşterea gradului polinomului de interpolare.
Polinom de interpolare de grad 2:

$$\begin{aligned} g(x) &= \frac{(x-x_k)(x-x_{k+1})}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})} y_{k-1} + \frac{(x-x_{k-1})(x-x_{k+1})}{(x_k-x_{k-1})(x_k-x_{k+1})} y_k + \\ &+ \frac{(x-x_{k-1})(x-x_k)}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)} y_{k+1}, \end{aligned} \tag{12}$$

$$\begin{aligned} g'(x) &= \frac{(x-x_k)+(x-x_{k+1})}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})} y_{k-1} + \frac{(x-x_{k-1})+(x-x_{k+1})}{(x_k-x_{k-1})(x_k-x_{k+1})} y_k + \\ &+ \frac{(x-x_{k-1})+(x-x_k)}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)} y_{k+1}. \end{aligned} \tag{13}$$

Formule de derivare numerică - cu eroare de ordinul 2

Notăm: $h_1 = x_k - x_{k-1}$ şi $h_2 = x_{k+1} - x_k$.
Evaluând polinomul derivat (58) în x_k se obţine formula de derivare centrată de ordinul 2

$$y'_k = -\frac{h_2}{h_1(h_1+h_2)}y_{k-1} - \frac{h_1-h_2}{h_1h_2}y_k + \frac{h_1}{h_2(h_1+h_2)}y_{k+1}. \quad (14)$$

Evaluând polinomul derivat (58) în punctele x_{k-1} şi x_{k+1} se obţine formula de derivare progresivă de ordinul 2

$$y'_{k-1} = -\frac{2h_1+h_2}{h_1(h_1+h_2)}y_{k-1} + \frac{h_1+h_2}{h_1h_2}y_k - \frac{h_1}{h_2(h_1+h_2)}y_{k+1}. \quad (15)$$

şi, respectiv, de derivare regresivă de ordinul 2

$$y'_{k+1} = -\frac{h_2}{h_1(h_1+h_2)}y_{k-1} - \frac{h_1+h_2}{h_1h_2}y_k + \frac{h_1+2h_2}{h_2(h_1+h_2)}y_{k+1}. \quad (16)$$

Formule de derivare numerică - cu eroare de ordinul 2

În cazul unui pas echidistant $h_1 = h_2 = h$

Derivare centrată

$$y'_k = \frac{y_{k+1} - y_{k-1}}{2h} \tag{17}$$

Derivare progresivă

$$y'_{k-1} = \frac{-3y_{k-1} + 4y_k - y_{k+1}}{2h} \tag{18}$$

Derivare regresivă

$$y'_{k+1} = \frac{-y_{k-1} - 4y_k + 3y_{k+1}}{2h} \tag{19}$$

Formule de derivare numerică - cu eroare de ordinul 2

$$f(x) = f(x_k)+f'(x_k)(x-x_k)+\frac{f''(x_k)}{2!}(x-x_k)^2+\mathrm{O}((x-x_k)^3). \quad (20)$$

Evaluăm în x_{k-1} şi x_{k+1} (pp. $x_{k+1} - x_k = x_k - x_{k-1} = h$)

$$f(x_{k+1}) = f(x_k) + f'(x_k)h + \frac{f''(x_k)}{2!}h^2 + \mathrm{O}(h^3), \quad (21)$$

$$f(x_{k-1}) = f(x_k) - f'(x_k)h + \frac{f''(x_k)}{2!}h^2 - \mathrm{O}(h^3). \quad (22)$$

$\Rightarrow$

$$f(x_{k+1}) - f(x_{k-1}) = 2f'(x_k)h + \mathrm{O}(h^3), \quad (23)$$

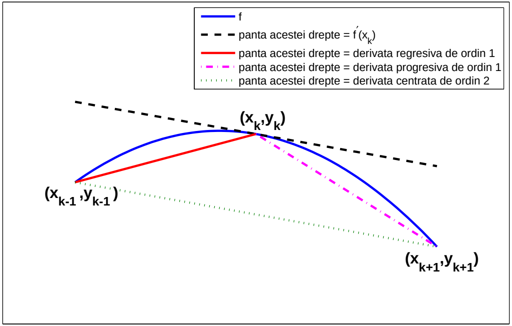
$\Rightarrow$

$$\frac{y_{k+1} - y_{k-1}}{2h} - f'(x_k) = \mathrm{O}(h^2). \quad (24)$$

Formula de derivare centrată are eroarea de trunchiere de ordinul 2

Formule de derivare numerică

Formulele de ordinul 2 sunt mai precise decât formulele de ordinul 1.



Semnificaţia geometrică a celor mai simple formule de derivare numerică (pas echidistant).

Modul de definire al funcţiei

Este important modul în care este cunoscută funcţia.

- Funcţia dată tabelar (prin date)
  1. se evaluează derivatele în punctele din tabel
  2. se evaluează derivata într-un punct oarecare, facând o interpolare liniară pe porţiuni a acestor valori.
- Funcţia dată printr-un cod
  1. se alege pasul optim de derivare pentru o anumită formulă de derivare numerică;
  2. se evaluează derivata cu formula de derivare numerică într-un punct oarecare.

Algoritmi numerici - funcţie dată tabelar

1. Se evaluează derivatele în punctele din tabel
2. Se evaluează derivata într-un punct oarecare, facând o interpolare liniară pe porţiuni a acestor valori.

<u>procedură</u> pregătire_derivate(n, x, y, dy)
; calculează derivatele numerice în noduri
<u>întreg</u> n ; numărul de puncte din tabel minus 1
<u>tablou</u> <u>real</u> $x[n]$, $y[n]$ tabelul de valori, indici de la 0
<u>tablou</u> <u>real</u> $dy[n]$; valorile derivatelor
...
; la capătul din stânga diferenţe progresive de ordinul 2
$h1 = x_1 - x_0$
$h2 = x_2 - x_1$
$dy_0 = -(2h1 + h2)/(h1(h1 + h2))dy_0 + (h1 + h2)/(h1h2)dy_1 - h1/(h2(h1 + h2))dy_2$
; la capătul din dreapta diferenţe regresive de ordinul 2
$h1 = x_{n-1} - x_{n-2}$
$h2 = x_n - x_{n-1}$
$dy_n = -h2/(h1(h1 + h2))dy_{n-2} - (h1 + h2)/(h1h2)dy_{n-1} + (h1 + 2h2)/(h2(h1 + h2))dy_n$
; în nodurile interioare diferenţe centrate de ordinul 2
<u>pentru</u> $k = 1, n - 1$
  $h1 = x_k - x_{k-1}$
  $h2 = x_{k+1} - x_k$
  $dy_k = -h2/(h1(h1 + h2))dy_{k-1} - (h1 - h2)/(h1h2)dy_k + h1/(h2(h1 + h2))dy_{k+1}$

•

rez = interpolare_lpp($n, x, dy, xcrt$)

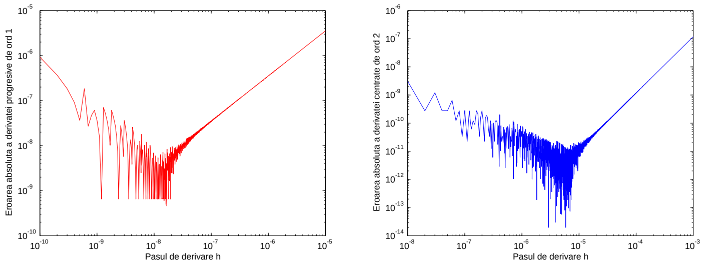
Algoritmi numerici - funcţie dată prin cod

- Pasul de discretizare trebuie *ales* de utilizator.
- *Eroarea de trunchiere* - cu atât mai mică cu cât pasul este mai mic.

Dar,

Există şi *eroare de rotunjire*, - cu atât mai mare cu cât pasul este mai mic.

Algoritmi numerici - funcţie dată prin cod



Teste numerice pentru $\sin'(\pi/4)$: Eroarea în funcţie de pasul de derivare pentru formula de derivare progresivă de ordinul 1 (stânga) şi formula de derivare centrată de ordinul 2 (dreapta).

Pas optim de derivare numerică = pasul de la care încep să predomine erorile de trunchiere. Atenţie: Nu este pasul pentru care eroarea este minimă!.

Eroarea "optimă" în cazul derivării centrate este mai mică decât în cazul derivării progresive.

Algoritmi numerici - funcţie dată prin cod

Estimarea h_{optim} în cazul formulei de derivare progresivă de ordinul 1.

$$y'_k = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} = \frac{y_{k+1} - y_k}{h} \tag{25}$$

$$e_t = |y'_k - f'(x_k)| \leq \frac{M_2}{2} h, \tag{26}$$

unde $|f''(x)| \leq M_2$. Notăm marginea erorii de trunchiere:

$$a_t = \frac{M_2}{2} h \tag{27}$$

Algoritmi numerici - funcţie dată prin cod

Pp. $|e_{y_k}/y_k| < \mathrm{eps}$, şi $e_h = 0$

$$|e_r| \leq \frac{|e_{y_{k+1}}| + |e_{y_k}|}{h} \leq \frac{\mathrm{eps}(|y_{k+1}| + |y_k|)}{h} \tag{28}$$

Dacă $|f(x)| \leq M_0$ atunci

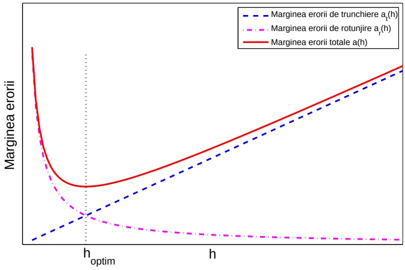
$$|e_r| \leq \frac{2M_0\mathrm{eps}}{h} \overset{not}{=} a_r \tag{29}$$

Algoritmi numerici - funcţie dată prin cod

Eroarea totală $e = e_t + e_r$ va fi majorată de

$$|e| = |e_t + e_r| \leq |e_t| + |e_r| \leq a_t + a_r = m(h)$$

$$m(h) = \frac{M_2}{2} h + \frac{2M_0 \text{eps}}{h^2} \qquad \min(m(h)) = m(h_{\text{optim}}). \qquad (30)$$



Pasul optim de derivare numerică este pasul de la care încep să predomine erorile de trunchiere.

Algoritmi numerici - funcţie dată prin cod

Condiţia de minim

$$m'(h) = 0$$

$\Rightarrow$

$$h_{\text{optim}} = 2\sqrt{\frac{M_0 \text{eps}}{M_2}}. \tag{31}$$

Pasul optim de derivare depinde de:

- eroarea de rotunjire
- marginea funcţiei
- marginea derivatei a doua.

Este pasul pentru care marginea erorii totale este minimă şi nu trebuie confundat cu pasul pentru care eroarea totală este minimă, pas care s-ar putea să fie mai mic decât pasul optim, dar care nu poate fi estimat apriori.

Algoritmi numerici - funcţie dată prin cod

Algoritm pentru derivata progresivă de ordinul 1 a unei funcţii date prin cod, folosind un pas de derivare optim. $M_2 \approx \frac{f(x)-2f(x+h)+f(x+2h)}{h^2}$.

```
funcţie derivata_f(x, h0, maxit)
; calculează derivata numerică a funcţiei f folosind diferenţe progresive de ordinul 1 şi pas optim de derivare
real x                                  ; punctul în care se va evalua derivata
real h0                                 ; pasul iniţial
întreg maxit                            ; numărul maxim de iteraţii pentru calculul pasului
...
eps = zeroul_maşinii ()
f0 = f(x)                               ; valoarea funcţiei în punctul de derivare
h = h0
k = 0
repetă
        k = k + 1
        f1 = f(x + h)
        f2 = f(x + 2h)
        M2 = |f0 − 2f1 + f2|/h^2        ; estimarea marginii derivatei a doua
        dacă M2 < eps                   ; testează dacă derivata a doua este zero
                hoptim = h              ; funcţia e liniară, eroarea de trunchiere e zero
        altfel
                M0 = max(|f0|, |f1|, |f2|)      ; estimarea marginii funcţiei
                hoptim = 2 sqrt(M0eps/M2)
        r = h/hoptim                    ; rata de modificare a pasului
        h = hoptim
până când (k > maxit) sau (r > 0.5 şi r < 2)
întoarce (f(x + hoptim) − f0)/hoptim
```

Extrapolare Richardson

Pentru a obţine formule de derivare numerică de ordin mai mare decât 2:

1. Se poate reface raţionamentul pornind de la un polinom de interpolare de ordin mai mare ca 2;
2. Se poate folosi extrapolarea Richardson.

Extrapolare Richardson

Reamintim:

$$f(x_{k+1}) = f(x_k) + f'(x_k)h + \frac{f''(x_k)}{2!}h^2 + \mathrm{O}(h^3), \quad (32)$$

$$f(x_{k-1}) = f(x_k) - f'(x_k)h + \frac{f''(x_k)}{2!}h^2 - \mathrm{O}(h^3). \quad (33)$$

$\Rightarrow$

$$f(x_{k+1}) - f(x_{k-1}) = 2f'(x_k)h + \mathrm{O}(h^3), \quad (34)$$

$\Rightarrow$

$$\frac{f(x_{k+1}) - f(x_{k-1})}{2h} - f'(x_k) = \mathrm{O}(h^2). \quad (35)$$

Schimbăm notaţiile:

x_k devine x

x_{k+1} devine $x + h$

x_{k-1} devine $x - h$

Extrapolare Richardson

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2!}h^2 + \mathrm{O}(h^3), \tag{36}$$

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2!}h^2 - \mathrm{O}(h^3). \tag{37}$$

$\Rightarrow$

$$f(x+h) - f(x-h) = 2f'(x)h - a_2 h^3 - a_4 h^5 + \cdots, \tag{38}$$

$\Rightarrow$

$$\frac{f(x+h) - f(x-h)}{2h} - f'(x) = -a_2 h^2 - a_4 h^4 + \cdots. \tag{39}$$

Extrapolare Richardson

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + a_2 h^2 + a_4 h^4 + \cdots . \qquad (40)$$

Dacă considerăm f şi x fixate, se defineşte funcţia

$$\varphi(h) = \frac{f(x+h) - f(x-h)}{2h}. \qquad (41)$$

$\varphi(h)$ este o aproximare a lui $f'(x)$ cu eroarea $O(h^2)$.
Scop: să calculăm $\lim_{h\to 0} \varphi(h)$ (care este $f'(x)$).
Variante

1. Să calculăm $\varphi(h)$, $\varphi(h/2)$, $\varphi(h/4)$,. . ..
2. Să folosim expresia erorii pentru a îmbunătăţi convergenţa (extrapolare Richardson).

Extrapolare Richardson

Ideea:

$$f'(x) = \varphi(h) + a_2 h^2 + a_4 h^4 + a_6 h^6 \cdots . \tag{42}$$

$$\varphi(h) = f'(x) - a_2 h^2 - a_4 h^4 - a_6 h^6 \cdots . \tag{43}$$

$$\varphi(h/2) = f'(x) - a_2 (h/2)^2 - a_4 (h/2)^4 - a_6 (h/2)^6 \tag{44}$$

Se elimină termenul dominant din eroare, prin operaţii algebrice.

$$\varphi(h) - 4\varphi(h/2) = -3f'(x) - 3a_4 h^4/4 - 15 a_6 h^6/16 \cdots . \tag{45}$$

$$-1/3\varphi(h) + 4/3\varphi(h/2) = f'(x) + a_4 h^4/4 + 5a_6 h^6/16 \cdots . \tag{46}$$

Extrapolare Richardson

Se poate continua:

$$-1/3\varphi(h) + 4/3\varphi(h/2) = f'(x) + a_4 h^4/4 + 5a_6 h^6/16 \cdots . \quad (47)$$

$$\Phi(h) = -1/3\varphi(h) + 4/3\varphi(h/2) \quad (48)$$

$\Phi(h)$ este o aproximare a lui $f'(x)$ cu eroarea $O(h^4)$.

$$\begin{aligned} \Phi(h) &= f'(x) + b_4 h^4 + b_6 h^6 \cdots . & (49) \\ \Phi(h/2) &= f'(x) + b_4 (h/2)^4 + b_6 (h/2)^6 & (50) \end{aligned}$$

$$\Phi(h) - 16\Phi(h/2) = -15f'(x) - 7b_6 h^6/8 \cdots . \quad (51)$$

$$-1/15\Phi(h) + 16/15\Phi(h/2) = f'(x) + 7b_6 h^6/120 + \cdots . \quad (52)$$

Extrapolare Richardson - Ideea generală

Se poate aplica pentru aproximarea cu acurateţe din ce în ce mai mare a unei mărimi L, pentru care există o funcţie $\varphi(h)$. a.î.

- $L = \lim_{h\to 0} \varphi(h)$
- $\varphi(h)$ poate fi evaluată pentru orice h;
- are loc proprietatea:

$$\varphi(h) = L - \sum_{k=1}^{\infty} a_{2k} h^{2k} \tag{53}$$

unde coeficienţii a_{2k} nu sunt cunoscuţi.

Se alege un h potrivit şi se calculează numerele

$$D(i,0) = \varphi(h/2^i), \quad i = 0, \ldots, n. \tag{54}$$

Extrapolare Richardson - Ideea generală

$D(i,0)$ reprezintă estimări ale lui L, dar estimări mai precise se pot obţine prin extrapolare Richardson.
Se demonstrează că [Cheney]:

$$D(i,j) = D(i,j-1) + (D(i,j-1) - D(i-1,j-1))/(4^i - 1), \quad j = 0,\dots,i. \tag{55}$$

$$\begin{array}{ccccc} D(0,0) & & & & \\ D(1,0) & D(1,1) & & & \\ D(2,0) & D(2,1) & D(2,2) & & \\ \vdots & \vdots & \vdots & & \\ D(n,0) & D(n,1) & D(n,2) & \cdots & D(n,n) \end{array} \tag{56}$$

Extrapolare Richardson pentru evaluarea derivatelor

Se dau:

- funcţia dată prin cod f;
- punctul în care dorim să calculăm derivata ei x;
- pasul iniţial pentru formula de derivare numerică h;
- informaţia despre ordinul erorii la care dorim să ajungem n.

...

<u>pentru</u> $i = 0, n$

$D_{i,0} = (f(x+h) - f(x-h))/(2h)$

<u>pentru</u> $j = 1, i$

$D_{i,j} = D_{i,j-1} + (D_{i,j-1} - D_{i-1,j-1})/(4^i - 1)$

•

$h = h/2$

•

...

Derivate de ordin superior

$f''(x) = ?$

Pot fi privite ca aplicaţii recursive ale derivatei de ordinul unu.

pregătire_derivate$(n, x, dy, d2y)$

rez = interpolare_lpp$(n, x, d2y, xcrt)$

Experimentele numerice arată că procedând astfel, acurateţea rezultatului se pierde pe măsură ce ordinul derivatei creşte.

Derivate de ordin superior

O altă posibilitate: deducerea formulelor de derivare pornind de la polinomul de interpolare iniţial.

$$\begin{aligned} g(x) &= \frac{(x-x_k)(x-x_{k+1})}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})} y_{k-1} + \frac{(x-x_{k-1})(x-x_{k+1})}{(x_k-x_{k-1})(x_k-x_{k+1})} y_k + \\ &+ \frac{(x-x_{k-1})(x-x_k)}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)} y_{k+1}, \end{aligned} \tag{57}$$

$$\begin{aligned} g'(x) &= \frac{(x-x_k)+(x-x_{k+1})}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})} y_{k-1} + \frac{(x-x_{k-1})+(x-x_{k+1})}{(x_k-x_{k-1})(x_k-x_{k+1})} y_k + \\ &+ \frac{(x-x_{k-1})+(x-x_k)}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)} y_{k+1}. \end{aligned} \tag{58}$$

$$\begin{aligned} g''(x) &= \frac{2}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})} y_{k-1} + \frac{2}{(x_k-x_{k-1})(x_k-x_{k+1})} y_k + \\ &+ \frac{2}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)} y_{k+1}, \end{aligned} \tag{59}$$

Derivate de ordin superior

În cazul particular $x_{k+1} - x_k = x_k - x_{k-1} = h$

$$g''(x) = \frac{y_{k-1} - 2y_k + y_{k+1}}{h^2}. \tag{60}$$

Derivate de ordin superior - evaluarea erorii

Seria Taylor cu un număr suplimentar de termeni:

$$f(x_{k+1}) = f(x_k) + f'(x_k)h + \frac{f''(x_k)}{2!}h^2 + \frac{f'''(x_k)}{3!}h^3 + \mathrm{O}(h^4), \quad (61)$$

$$f(x_{k-1}) = f(x_k) - f'(x_k)h + \frac{f''(x_k)}{2!}h^2 - \frac{f'''(x_k)}{3!}h^3 - \mathrm{O}(h^4). \quad (62)$$

$$f(x_{k+1}) + f(x_{k-1}) = 2f(x_k) + f''(x_k)h^2 + \mathrm{O}(h^4). \quad (63)$$

$$\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} - f''(x_k) = \mathrm{O}(h^2). \quad (64)$$

Derivate de ordin superior - evaluarea erorii

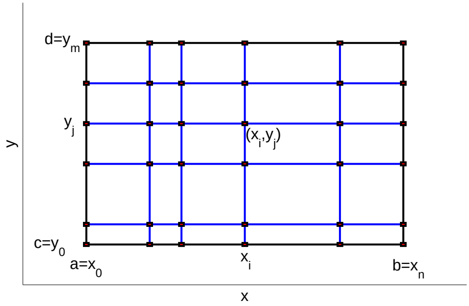
- Polinomul de interpolare iniţial trebuie să aibă un grad suficient de mare pentru ca derivata de ordin superior să fie nenulă.
- Ar putea apare fenomenului Runge care va afecta şi acurateţea derivatelor numerice. De aceea, calculul numeric al derivatelor de ordin superior trebuie evitat.

Derivate parţiale

$$f(x,y) : [a,b] \times [c,d] \to \mathbb{R}. \qquad \frac{\partial f}{\partial x} = ? \quad \frac{\partial f}{\partial y} = ?$$

$$a = x_0, x_1, \ldots, x_i, \ldots, x_n = b$$

$$c = y_0, y_1, \ldots, y_j, \ldots, y_m = d$$



Valorile din "tabel":

$$f(x_i, y_j) = z_{i,j}$$

$$i = 0, n \quad j = 0, m$$

Derivate parţiale

Evaluarea numerică a derivatelor parţiale se reduce la evaluarea numerică a derivatelor totale.

$$\left.\frac{\partial f}{\partial x}\right|_{x=x_0, y=y_0} = \lim_{x \to x_0} \frac{f(x, y_0) - f(x_0, y_0)}{x - x_0} = \lim_{x \to x_0} \frac{f1(x) - f1(x_0)}{x - x_0} = \frac{\mathrm{d}f1}{\mathrm{d}x}, \tag{65}$$

unde $f1(x) = f(x, y_0)$ este o funcţie care depinde de o singură variabilă reală. Similar,

$$\left.\frac{\partial f}{\partial y}\right|_{x=x_0, y=y_0} = \lim_{y \to y_0} \frac{f(x_0, y) - f(x_0, y_0)}{y - y_0} = \lim_{y \to y_0} \frac{f2(y) - f2(y_0)}{y - y_0} = \frac{\mathrm{d}f2}{\mathrm{d}y}, \tag{66}$$

unde $f2(y) = f(x_0, y)$ este o funcţie care depinde de o singură variabilă reală.

Derivate parţiale

Formulele de derivare progresivă de ordinul 1

$$\frac{\partial g}{\partial x}(x_i, y_j) = \frac{z_{i+1,j} - z_{i,j}}{x_{i+1} - x_i}, \tag{67}$$

$$\frac{\partial g}{\partial y}(x_i, y_j) = \frac{z_{i,j+1} - z_{i,j}}{y_{j+1} - y_j} \tag{68}$$

etc.

Derivate parţiale

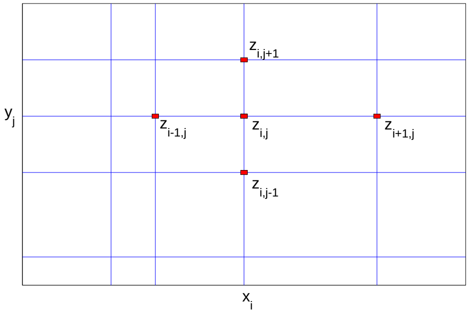
Exemplu: evaluarea numerică a operatorului Laplace aplicat unei funcţii de două varibile.

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \tag{69}$$

Pp. $x_{i+1} - x_i = y_{j+1} - y_j = h$, pentru orice $i = 0, n-1$, $j = 0, m-1$

$$\begin{aligned} \Delta g &= \frac{z_{i-1,j} - 2z_{i,j} + z_{i+1,j}}{h^2} + \frac{z_{i,j-1} - 2z_{i,j} + z_{i,j+1}}{h^2} = \\ &= \frac{z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1} - 4z_{i,j}}{h^2}, \end{aligned} \tag{70}$$

Derivate parţiale



$$\Delta g = \frac{z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1} - 4z_{i,j}}{h^2} \tag{71}$$

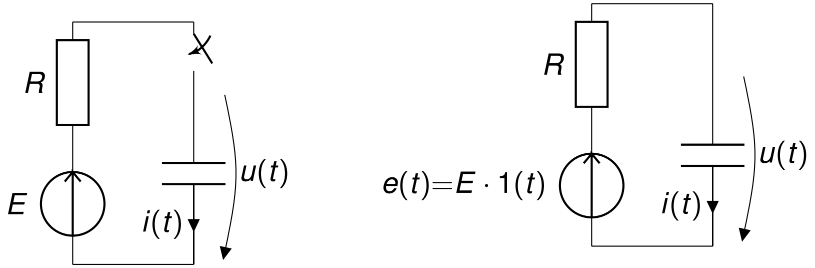
Dacă astfel de raţionamente se folosesc pentru rezolvarea numerică a problemelor de inginerie formulate cu ajutorul unor ecuaţii cu derivate totale (ODE) sau parţiale (PDE) atunci se spune că pentru rezolvarea problemei se aplică *metoda diferenţelor finite*.

Metoda diferenţelor finite

- **Pasul 1:** Se alege o schemă de diferenţe finite pentru aproximarea derivatelor din ecuaţii şi se rescrie ecuaţia ca ecuaţie cu diferenţe finite.
- **Pasul 2:** Se stabileşte grila de discretizare şi se scrie ecuaţia discretizată pentru fiecare nod.
- **Pasul 3:** Se rezolvă sistemul de ecuaţii pentru determinarea valorilor necunoscute în nodurile grilei.
- **Pasul 4:** Calculul altor valori, în afara celor din noduri, se face prin interpolare.

1 şi 2 = *preprocesare*
3 = *rezolvare*
4 = *postprocesare*

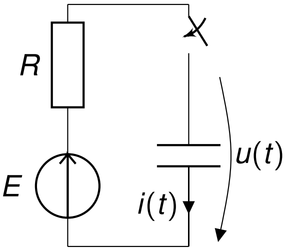
Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)

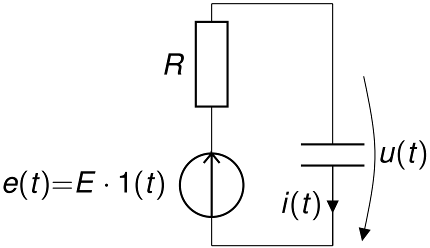


$C = 4\mu$F, $E = 20$ mV, $R = 10\ \Omega$, $u(0) = 0$.

$$Ri(t) + u(t) = E \qquad i(t) = C\frac{\mathrm{d}u(t)}{\mathrm{d}t}, \tag{72}$$

$$RC\frac{\mathrm{d}u(t)}{\mathrm{d}t} + u(t) = E. \qquad u(0) = u_0 = 0. \tag{73}$$





Soluţie analitică

$$u(t) = (u_0 - E)\exp(-t/\tau) + E. \tag{74}$$

unde $\tau = RC$ este *constanta de timp* a circuitului.

Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)

Ecuaţia (73) o rescriem ca

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} + \frac{1}{\tau}u(t) = \frac{1}{\tau}E. \tag{75}$$

Vom urmări calculul numeric în intervalul de timp $[0, t_{max}]$ unde $t_{max} = 10\tau$ într-o reţea echidistantă de N puncte t_k, unde pasul de discretizare h este

$$t_{k+1} - t_k = h, \quad \text{pentru} \quad k = 1, \ldots, N-1. \tag{76}$$

Vom nota valorile discrete obţinute prin rezolvare numerică cu u_k. Ele vor fi aproximaţii ale mărimii reale u.

$$u_k \approx u(t_k). \tag{77}$$

Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} + \frac{1}{\tau}u(t) = \frac{1}{\tau}E. \tag{78}$$

Varianta I - diferenţe finite progresive de ordinul 1

$$\frac{u_{k+1} - u_k}{h} + \frac{1}{\tau}u_k = \frac{1}{\tau}E, \tag{79}$$

$\Rightarrow$ u_{k+1} poate fi calculată explicit cu formula

$$u_{k+1} = u_k \left(1 - \frac{h}{\tau}\right) + \frac{h}{\tau}E. \tag{80}$$

Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)

$$u_{k+1} = u_k \left(1 - \frac{h}{\tau}\right) + \frac{h}{\tau} E. \tag{81}$$

<u>procedură</u> mdf_ode_p1(u0,E,tau,h,N,u)
; rezolvă ecuaţia $\frac{\mathrm{d}u(t)}{\mathrm{d}t} + \frac{1}{\tau} u(t) = \frac{1}{\tau} E$ cu metoda diferenţelor finite
; d/dt se discretizează folosind diferenţe progresive de ordinul 1

```
real u0                    ; condiţia iniţială - dată
real E                     ; coeficient în ecuaţie - dată
real tau                   ; constantă de timp - dată
real h                     ; pas de discretizare al intervalului de timp - dat
întreg N                   ; număr de valori de timp - dat
tablou real u[N]           ; soluţia discretă - rezultat
u(1) = u0
pentru k = 1,N-1
        u(k+1) = u(k)*(1-h/tau) + h*E/tau
•
retur
```

Această metodă, cunoscută şi sub numele de *Euler explicită*, este instabilă pentru $h > \tau$.

Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} + \frac{1}{\tau}u(t) = \frac{1}{\tau}E. \tag{82}$$

Varianta a II-a - diferenţe finite regresive de ordinul 1

$$\frac{u_k - u_{k-1}}{h} + \frac{1}{\tau}u_k = \frac{1}{\tau}E, \tag{83}$$

$\Rightarrow$ u_k poate fi calculată explicit:

$$u_k = \left(u_{k-1} + \frac{h}{\tau}E\right) / \left(1 + \frac{h}{\tau}\right). \tag{84}$$

Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)

$$u_k = \left(u_{k-1} + \frac{h}{\tau} E \right) / \left(1 + \frac{h}{\tau} \right). \tag{85}$$

<u>procedură</u> mdf_ode_r1(u0,E,tau,h,N,u)
; rezolvă ecuaţia $\frac{\mathrm{d}u(t)}{\mathrm{d}t} + \frac{1}{\tau} u(t) = \frac{1}{\tau} E$ cu metoda diferenţelor finite
; d/dt se discretizează folosind diferenţe regresive de ordinul 1
<u>real</u> u0 ; condiţia iniţială - dată
<u>real</u> E ; coeficient în ecuaţie - dată
<u>real</u> tau ; constantă de timp - dată
<u>real</u> h ; pas de discretizare al intervalului de timp - dat
<u>întreg</u> N ; număr de valori de timp - dat
<u>tablou real</u> u[N] ; soluţia discretă - rezultat
u(1) = u0
<u>pentru</u> k = 2,N
    u(k) = (h*E/tau + u(k-1))/(1 + h/tau)
• <u>retur</u>

Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)

OBS: Ecuaţia generală este (unde f - funcţie în gen. neliniară):

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y)$$

Metoda *Euler explicită* - derivată progresivă:

$$\frac{y_{k+1} - y_k}{h} = f(t_k, y_k)$$

$$y_{k+1} = y_k + hf(t_k, y_k)$$

$$y_n = y_v + hf(t_v, y_v)$$

Calculul soluţiei la un nou pas de timp se face explicit în funcţie de informaţia de la pasul vechi de timp.

Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)

OBS: Ecuaţia generală este (unde f - funcţie în gen. neliniară):

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y)$$

Metoda *Euler implicită* - derivată regresivă:

$$\frac{y_k - y_{k-1}}{h} = f(t_k, y_k)$$

$$y_k = y_{k-1} + hf(t_k, y_k)$$

$$y_n = y_v + hf(t_n, y_n)$$

$$y_n - hf(t_n, y_n) - y_v = 0$$

Calculul soluţiei la un nou pas de timp presupune rezolvarea unei ecuaţii neliniare.
Numai dacă f este liniară, atunci aceasta rezolvare se finalizează cu un calcul explicit (cz. exemplului studiat).

Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} + \frac{1}{\tau}u(t) = \frac{1}{\tau}E. \tag{86}$$

Varianta a III-a - diferenţe finite centrate de ordinul 2

$$\frac{u_{k+1} - u_{k-1}}{2h} + \frac{1}{\tau}u_k = \frac{1}{\tau}E, \tag{87}$$

$$-u_{k-1} + \frac{2h}{\tau}u_k + u_{k+1} = \frac{2h}{\tau}E. \tag{88}$$

$k = 2, \ldots, N-1 \Rightarrow N-2$ ecuaţii cu N necunoscute.
Trebuie adăugate relaţiile la capete.

Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)

La $t = 0$: $u(1) = u_0$
Pentru $k = N$ diferenţe regresive de ordinul 2:

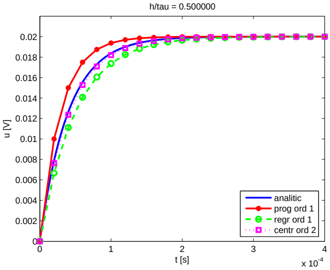
$$\frac{u_{N-2} - 4u_{N-1} + 3u_N}{2h} + \frac{1}{\tau}u_N = \frac{1}{\tau}E, \qquad (89)$$

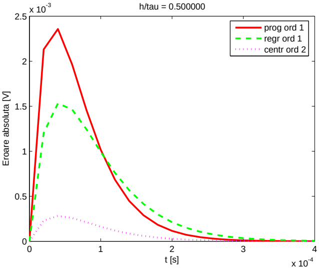
$$u_{N-2} - 4u_{N-1} + \left(3 + \frac{2h}{\tau}\right)u_N = \frac{2h}{\tau}E. \qquad (90)$$

Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)

```
procedură mdf_ode_c2(u0,E,tau,h,N,u)
; rezolvă ecuaţia du(t)/dt + (1/τ) u(t) = (1/τ) E cu metoda diferenţelor finite
; d/dt se discretizează folosind diferenţe centrate de ordinul 2
real u0                    ; condiţia iniţială - dată
real E                     ; coeficient în ecuaţie - dată
real tau                   ; constantă de timp - dată
real h                     ; pas de discretizare al intervalului de timp - dat
întreg N                   ; număr de valori de timp - dat
tablou real u[N]           ; soluţia discretă - rezultat
tablou real A[N,N]         ; matricea coeficienţilor sistemului asamblat - stocata rar
tablou real tl[N]          ; vectorul termenilor liberi ai sistemului asamblat
A = 0                      ; pseudocod simplificat
tl = 0
A(1,1) = 1                 ; conditia initiala tl(1) = u0
pentru k = 2,N-1           ; noduri interioare
        A(k,k-1) = -1
        A(k,k) = 2*h/tau
        A(k,k+1) = 1
        tl(k) = 2*h*E/tau
•
A(N,N-2) = 1               ; conditia la tmax = N*h
A(N,N-1) = -4
A(N,N) = (3 + 2*h/tau)
tl(N) = 2*h*E/tau
u = "A^-1 tl"              ; rezolva sistemul algebric liniar asamblat
retur
```

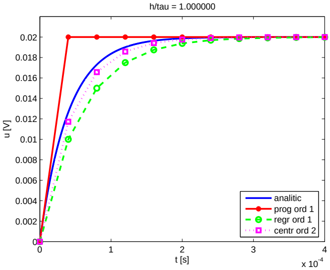
Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)

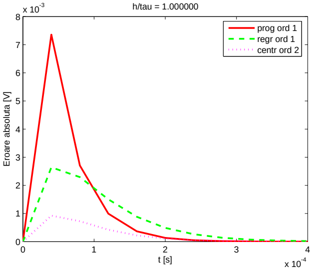




Cazul $h = \tau/2$.

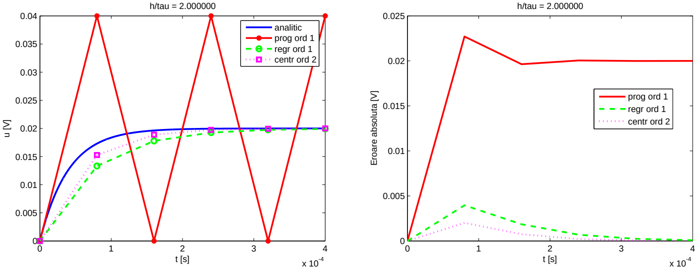
Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)





Cazul $h = \tau$.

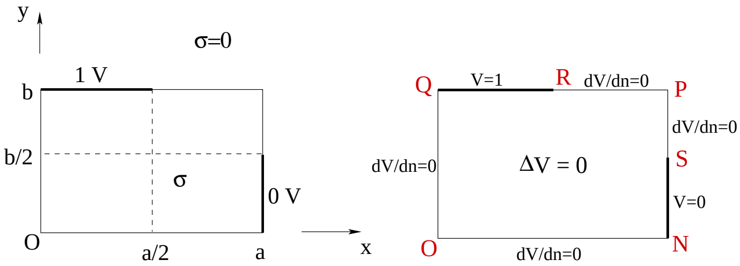
Exemplul 1 - rezolvarea unei ec.dif. ordinare (ODE)



Cazul $h = 2\tau$.

Obs: Discuţia despre rezolvarea ecuaţiilor şi sistemelor ODE va fi reluată.

Exemplul 2 - ec. derivate parţiale (eliptic)



Problema 2D de regim electrocinetic: domeniul de calcul (stânga), condiţii de frontieră (dreapta).

Se dă un conductor omogen, de conductivitate σ, situat într-un mediu perfect izolant. Conductorul are o dimensiune după axa Oz mult mai lungă decât celelalte două. Figura reprezintă o secţiune perpendiculară pe direcţia dimensiunii foarte mari. Această secţiune este un dreptunghi, de dimensiuni *a*, *b*. Conductorul are două borne supraconductoare, una aflată la potenţial $V_0 = 1$V, iar cealaltă aflată la potenţial nul. Se cere (pentru început) câmpul în acest domeniu.

Exemplul 2 - ec. derivate parţiale (eliptic)

Formularea în V - potenţial electrocinetic, unde $\mathbf{E} = -\mathrm{grad}\, V$, $\mathbf{E}$ este intensitatea câmpului electric.
Ecuaţia de ordinul doi satisfăcută de V:

$$-\operatorname{div}(\sigma \,\mathrm{grad}\, V) = 0 \tag{91}$$

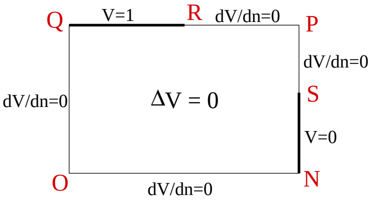
$V : \mathcal{D} \to \mathbb{R}$; $\mathcal{D}$ = ONPQ; (91) - ecuaţie eliptică, de tip Laplace generalizată. Domeniul omogen $\Rightarrow$ ecuaţie de tip Laplace:

$$\Delta V = 0, \tag{92}$$

$\Delta = \operatorname{div}\,\mathrm{grad}$ este operatorul Laplace; în 2D, xOy:

$$\Delta V = \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2}. \tag{93}$$

Exemplul 2 - ec. derivate parţiale (eliptic)

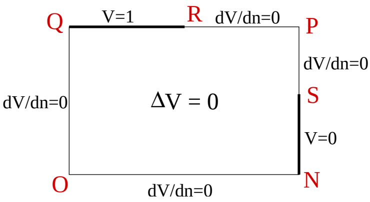


$\Rightarrow$ Ecuaţia de rezolvat:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0, \tag{94}$$

unde $V = V(x, y) : \mathcal{D} \to \mathbb{R}$

Exemplul 2 - ec. derivate parţiale (eliptic)



Condiţii de frontieră:

$$V = V_0 \quad \text{pe} \quad \mathrm{QR} \quad (\mathrm{Dirichlet}) \tag{95}$$

$$V = 0 \quad \text{pe} \quad \mathrm{SN} \quad (\mathrm{Dirichlet}) \tag{96}$$

$$\frac{\partial V}{\partial n} = 0 \quad \text{pe} \quad \mathrm{NOQ} \cup \mathrm{RPS} \quad (\mathrm{Neumann}) \tag{97}$$

Exemplul 2 - ec. derivate parţiale (eliptic)

Gridul de discretizare bidimensional:

$$0 = x_1 < x_2 < \ldots < x_{n_x} = a$$
$$0 = y_1 < y_2 < \ldots < y_{n_y} = b.$$

Un nod al gridului:

$$(x_i, y_j) \quad i = 1, \ldots, n_x, \quad j = 1, \ldots, n_y.$$

MDF determină valorile potenţialului în nodurile acestui grid:

$$V(x_i, y_j) \stackrel{not}{=} V_{i,j} \quad i = \overline{1, n_x}, \quad j = \overline{1, n_y}. \tag{98}$$

Exemplul 2 - ec. derivate parţiale (eliptic)

Ecuaţia asociată unui nod interior: Pp h_x, h_y paşii de discretizare:

$$\frac{\partial^2 V}{\partial x^2}(x_i, y_j) = \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{h_x^2}, \tag{99}$$

$$\frac{\partial^2 V}{\partial y^2}(x_i, y_j) = \frac{V_{i,j+1} - 2V_{i,j} + V_{i,j-1}}{h_y^2}. \tag{100}$$

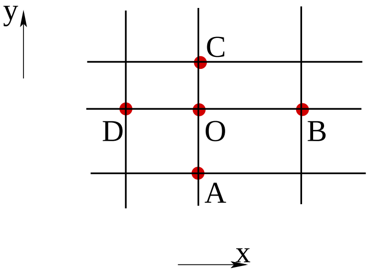
$$\frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{h_x^2} + \frac{V_{i,j+1} - 2V_{i,j} + V_{i,j-1}}{h_y^2} = 0, \tag{101}$$

$$2\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right) V_{i,j} - \frac{1}{h_x^2} V_{i+1,j} - \frac{1}{h_x^2} V_{i-1,j} - \frac{1}{h_y^2} V_{i,j+1} - \frac{1}{h_y^2} V_{i,j-1} = 0. \tag{102}$$

Potenţialul într-un nod interior = combinaţie liniară a potenţialelor nodurilor învecinate (media lor dacă paşii sunt egali).

Exemplul 2 - ec. derivate parţiale (eliptic)

Ecuaţia asociată unui nod interior - reţea neuniformă:



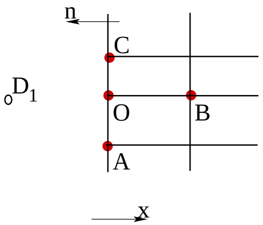
Nodurile marcate intervin în scrierea ecuaţiei unui nod interior.

$$V_O\left(\frac{1}{h_B h_D}+\frac{1}{h_A h_C}\right)- -V_A\frac{1}{h_A(h_A+h_C)}- -V_B\frac{1}{h_B(h_B+h_D)}- -V_C\frac{1}{h_C(h_A+h_C)}- -V_D\frac{1}{h_D(h_B+h_D)}=0, \quad (103)$$

$h_A = \|OA\|$, $h_B = \|OB\|$, $h_C = \|OC\|$, $h_D = \|OD\|$.

Exemplul 2 - ec. derivate parţiale (eliptic)

Ecuaţia asociată unui nod pe frontieră dreaptă:



Nodurile marcate intervin în scrierea ecuaţiilor unui nod pe frontieră Neumann dreaptă.

Fie $g = -\sigma \frac{\partial V}{\partial n}$ şi g_O valoarea medie în O:

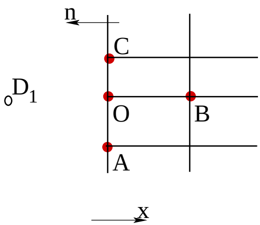
$$g_O = \frac{g_{OA} h_A + g_{OC} h_C}{h_A + h_C} \qquad (104)$$

unde g_{OA} este CF asociată lui OA, iar g_{OC} este CF asociată lui OC.
Nod “fantomă" D_1, $h_D = \|OD_1\|$.
Pentru simplitate $h_D = h_B$.

$$\frac{\partial V}{\partial x} = \frac{g}{\sigma} \quad \Rightarrow \quad V_{D_1} = V_B - 2h_B \frac{g_O}{\sigma}. \qquad (105)$$

Exemplul 2 - ec. derivate parţiale (eliptic)

Ecuaţia asociată unui nod pe frontieră dreaptă:



Nodurile marcate intervin în scrierea ecuaţiilor unui

nod pe frontieră Neumann dreaptă.

Pentru O se scrie o ecuaţie ca pentru un nod interior şi înlocuind expresia anterioară $\Rightarrow$:

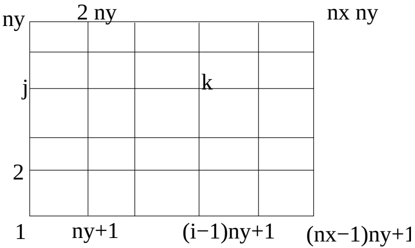
$$V_O \left(\frac{1}{h_B^2} + \frac{1}{h_A h_C} \right) - V_A \frac{1}{h_A(h_A + h_C)} - V_B \frac{1}{h_B^2} - V_C \frac{1}{h_C(h_A + h_C)} = -\frac{g_O}{\sigma h_B}. \tag{106}$$

Exemplul 2 - ec. derivate parţiale (eliptic)

- Discretizarea problemei conduce la un sistem de ecuaţii algebric liniar, prin a cărui rezolvare se obţin valorile potenţialelor în nodurile gridului.
- Pentru asamblarea sistemului este util ca nodurile să fie numerotate a.î. necunoscutele să reprezinte un vector.

Exemplul 2 - ec. derivate parţiale (eliptic)

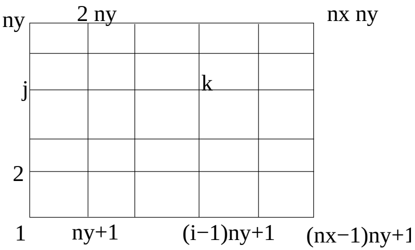
Exemplu de numerotare a nodurilor:



Relaţia între numărul nodului k şi poziţiile proiecţiilor sale pe cele două axe i şi j este

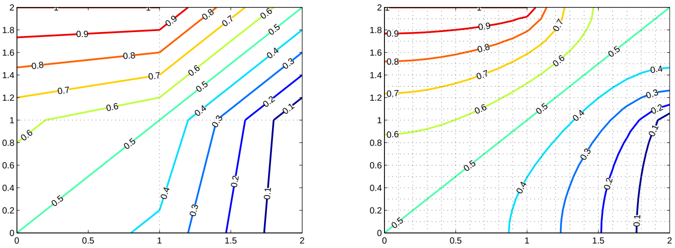
$$k = (i-1)n_y + j, \quad i = 1, \ldots, n_x, \quad j = 1, \ldots, n_y$$

Exemplul 2 - ec. derivate parţiale (eliptic)



- Se asamblează un sistem de ecuaţii algebrice liniare, de dimensiune egală cu numărul de noduri ($n = nx * ny$);
- Matricea este rară, cu structură bandă diagonală. Efortul de rezolvare O(n).

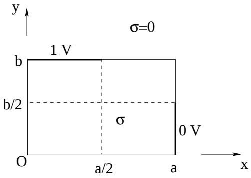
Exemplul 2 - ec. derivate parţiale (eliptic)

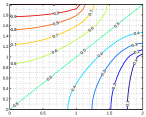


Linii echipotenţiale pentru un grid cu $n_x = n_y = 3$ (stânga) şi un grid cu $n_x = n_y = 21$ (dreapta).

- Soluţia numerică este mai precisă pe măsură ce scade pasul h. Eroarea $O(h^2)$.

Exemplul 2 - ec. derivate parţiale (eliptic)





Scopul unei astfel de aplicaţii:
Cât este rezistenţa echivalentă?

- Metoda liniară: $R = U/I$, $U = 1 - 0 = 1\,V$,
  $I = \int_{\text{terminal}} \mathbf{J} \cdot \mathbf{n}\, \mathrm{d}A$, $\mathbf{J} = \sigma \mathbf{E}$, $\mathbf{E} = -\mathrm{grad}\, V$.
- Metoda energetică: $R = U^2/P$
  $P = \int_{\text{domeniu}} \mathbf{J} \cdot \mathbf{E}\, \mathrm{d}v$

Cum se evaluează numeric integralele? (Vom reveni asupra acestei aplicaţii.)
Rezultate din problemele de câmp devin date de intrare pentru problemele de circuite electrice.

Exemplul 3 - ec. derivate parţiale (hiperbolic)

Ecuaţia undei scalare:

$$\frac{\partial^2 u}{\partial t^2} = v^2 \frac{\partial^2 u}{\partial x^2}, \tag{107}$$

$u(x,t) : [0,a] \times [0,T] \to \mathbb{R}$, v este o constantă cunoscută.
Soluţia acestei ecuaţii este o *undă* care se propagă cu viteză v.
Buna formulare a acestei probleme necesită impunerea

- *condiţiei iniţiale* $u(x,0) = h_0(x)$,
- *condiţiilor la capete* - relaţii în care intervin mărimile $u(0,t) = h_1(t)$ şi $u(a,t) = h_2(t)$.

Exemplul 3 - ec. derivate parţiale (hiperbolic)

Se poate demonstra că

$$u(x,t) = ud(x,t) + ui(x,t) + U_0, \tag{108}$$

unde $ud(x,t)$ se numeşte *undă directă* şi se poate scrie sub forma

$$ud(x,t) = f(x - vt), \tag{109}$$

iar $ui(x,t)$ se numeşte *undă inversă* şi se poate scrie sub forma

$$ud(x,t) = g(x + vt). \tag{110}$$

f şi g rezultă în mod univoc, din impunerea condiţiilor iniţiale şi condiţiilor la capete.

Exemplul 3 - ec. derivate parţiale (hiperbolic)

Avem nevoie de o discretizare spaţială şi de una temporală.
Pp. că ambele discretizări sunt uniforme şi vom nota cu

- Δz pasul discretizării spaţiale
- Δt pasul discretizării temporale
- N numărul de puncte de discretizare spaţiale
- M numărul de paşi de timp simulaţi $\Rightarrow$ timpul maxim de simulare este $T = M\Delta t$

Exemplul 3 - ec. derivate parţiale (hiperbolic)

Vom considera următoarele condiţii de unicitate:

- *condiţia iniţială* nulă $u(x, 0) = h_0(x) = 0$,
- *condiţia la capătul din stânga* - excitaţia cu un impuls Gauss: $u(0, t) = h_1(t) = exp(-(t - T/10).^2/(2 * (T/50)^2))$
- *condiţia la capătul din dreapta* $u(a, t) = h_2(t) = 0$.

În problemele în care apare propagare, este util uneori ca frontiera domeniului spaţial să fie modelată ca o frontieră absorbantă, "invizibilă" din punct de vedere al propagării mărimilor în domeniul spaţial modelat. Condiţia de frontieră absorbantă pentru capătul din dreapta al problemei studiate este

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial z} = 0.$$

Exemplul 3 - ec. derivate parţiale (hiperbolic)

Mărimea $u(x, t)$ este discretizată atât în spaţiu cât şi în timp.
Prin rezolvarea cu MDF, vom obţine aproximaţii:

$$u(x_k, t_j) \approx u_k^{(j)}, \quad k = 1, \ldots, N, \quad j = 1, \ldots, M. \tag{111}$$

Formule de derivare de ordin 2:

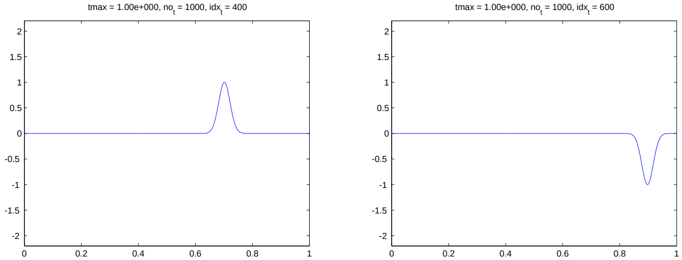
$$\frac{u_k^{(j-1)} - 2u_k^{(j)} + u_k^{(j+1)}}{(\Delta t)^2} = v^2 \frac{u_{k-1}^{(j)} - 2u_k^{(j)} + u_{k+1}^{(j)}}{(\Delta x)^2}, \tag{112}$$

$\Rightarrow$

$$u_k^{(j+1)} = \left(\frac{v\Delta t}{\Delta x}\right)^2 \left(u_{k-1}^{(j)} - 2u_k^{(j)} + u_{k+1}^{(j)}\right) + 2u_k^{(j)} - u_k^{j-1}, \quad k = 2, \ldots, N- \tag{113}$$

Mărimea u, într-un anumit punct k şi la un anumit moment de timp $j+1$ depinde explicit de valoarea în acel punct şi în cele învecinate la momentul de timp anterior j şi de valoarea în acel punct la momentul $j-1$

Exemplul 3 - ec. derivate parţiale (hiperbolic)



Propagarea unui impuls Gauss (stânga) şi rezultatul reflexiei după atingerea unei frontiere pe care s-a impus condiţie Dirichlet nulă (dreapta).

Exemplul 3 - ec. derivate parţiale (hiperbolic)

Atunci când o problemă necesită atât o discretizare spaţială cât şi una temporală, pentru ca soluţia numerică să fie stabilă, este necesar să fie îndeplinită condiţia lui Courant

$$|v|\Delta t \leq \Delta x. \tag{114}$$

Concluzii - MDF

1. În rezolvarea ecuaţiilor diferenţiale cu metoda diferenţelor finite, trebuie avute în vedere aspectul stabilitaţii algoritmului şi aspectul instabilitătii numerice datorate intrării în zona în care predomină erorile de rotunjire.
2. Stabilitatea se analizează diferit dacă ecuaţia este cu derivate ordinare (ODE) sau cu derivate parţiale (PDE).
3. Formulat pe scurt, trebuie ca paşii de discretizare să fie suficient de mici pentru ca algoritmul să fie stabil, dar nu prea mici ca să nu se intre în zona erorilor de rotunjire.
4. Un algoritm eficient ar trebui să poată să folosească paşi de discretizare neuniformi, adaptaţi tipului de variaţie a soluţiei.

Asupra rezolvării sistemelor ODE şi PDE vom reveni.

Referinţe

- [Ciuprina13a] Gabriela Ciuprina, *Algoritmi numerici pentru calcule ştiintifice în ingineria electrică* Editura MatrixROM, 2013, pag. 169-209

  Disponibilă la http://www.lmn.pub.ro/∼gabriela/books/AlgNr_MatrixRom2013.pdf

- [Cheney08] Ward Cheney and David Kincaid, *Numerical Mathematics and Computing*, Brooks/Cole publishing Company,2000. (Capitolul 4.3. *Estimating derivatives and Richardson extrapolation*)

  Disponibilă la http://www.physics.brocku.ca/Courses/5P10/References/cheneykincaid.pdf

- [Rumpf] Raymond Rumpf. Lecture 9 (CEM) – Finite-Difference Method video.

  Disponibil la https://www.youtube.com/watch?v=v-exTNOSG3g. Alte materiale la

  http://emlab.utep.edu/ee5390cem.htm