Derivarea numerică. Introducere în metoda diferenţelor finite.

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" Bucureşti, Facultatea de Inginerie Electrică

Suport didactic pentru disciplina *Metode numerice în ingineria electrică*, 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.

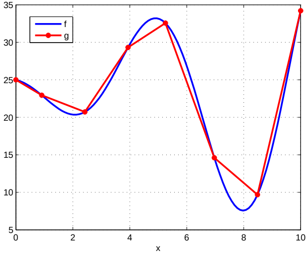
Ideea: $g(x)$ interpolarea sau aproximarea funcţiei şi

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

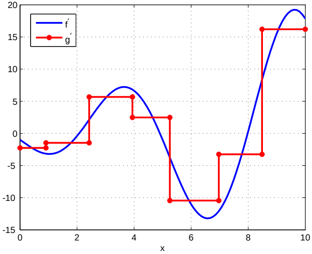
Precizia cu care se face interpolarea sau aproximarea va determina precizia de calcul a derivatei numerice.

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

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

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

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

Eroarea de trunchiere

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

Pp. $|f''(x)| \leq M_2$

Notăm $h = x_{k+1} - x_k$

$$|e_t| \leq \frac{M_2}{2} h, \tag{8}$$

Formule de derivare numerică - cu eroare de ordinul 1

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

$$|e_t| \leq \frac{M_2}{2} h, \tag{9}$$

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

(1 = puterea lui h, sau gradul polinomului de interpolare folosit)

Formule de derivare numerică - cu eroare de ordinul 1

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

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \mathrm{O}((x - x_k)^2) \tag{10}$$

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

$$y_{k-1} = y_k - f'(x_k)h + \mathrm{O}(h^2), \tag{11}$$

$$\frac{y_k - y_{k-1}}{h} - f'(x_k) = \mathrm{O}(h). \tag{12}$$

$|e_t|$ =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:

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

$$\begin{aligned} g'(x) &= \frac{(x-x_k)+(x-x_{k+1})}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})}y_{k-1} + \frac{(x-x_{k-1})+(x-x_{k+1})}{(x_k-x_{k-1})(x_k-x_{k+1})}y_k + \\ &+ \frac{(x-x_{k-1})+(x-x_k)}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)}y_{k+1}. \end{aligned} \tag{14}$$

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 (34) în x_k se obţine formula de derivare centrată de ordinul 2

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

Evaluând polinomul derivat (34) în punctele x_{k-1} şi x_{k+1} se obţine formula de derivare progresivă de ordinul 2

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

şi, respectiv, de derivare regresivă de ordinul 2

$$y'_{k+1} = -\frac{h_2}{h_1(h_1+h_2)} y_{k-1} - \frac{h_1 + h_2}{h_1 h_2} y_k + \frac{h_1 + 2h_2}{h_2(h_1+h_2)} y_{k+1}. \quad (17)$$

Formule de derivare numerică - cu eroare de ordinul 2

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

Derivare centrată

$$y'_k = \frac{y_{k+1} - y_{k-1}}{2h} \tag{18}$$

Derivare progresivă

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

Derivare regresivă

$$y'_{k+1} = \frac{-y_{k-1} - 4y_k + 3y_{k+1}}{2h} \tag{20}$$

Formule de derivare numerică - cu eroare de ordinul 2

$$f(x) = f(x_k)+f'(x_k)(x-x_k)+\frac{f''(x_k)}{2!}(x-x_k)^2+\mathrm{O}((x-x_k)^3). \quad (21)$$

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

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

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

$\Rightarrow$

$$f(x_{k+1}) - f(x_{k-1}) = 2f'(x_k)h + \mathrm{O}(h^3), \quad (24)$$

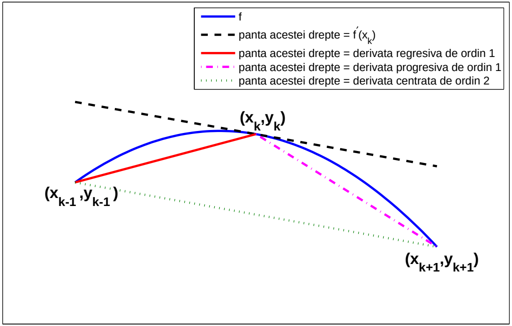
$\Rightarrow$

$$\frac{y_{k+1} - y_{k-1}}{2h} - f'(x_k) = \mathrm{O}(h^2). \quad (25)$$

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($n, x, dy, xcrt$)

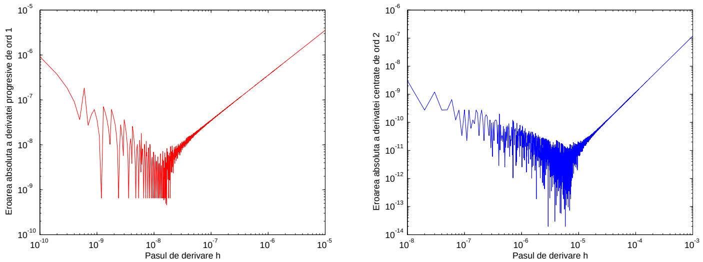
Algoritmi numerici - funcţie dată prin cod

- 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.

$$y'_k = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} = \frac{y_{k+1} - y_k}{h} \tag{26}$$

$$e_t = |y'_k - f'(x_k)| \leq \frac{M_2}{2} h, \tag{27}$$

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

$$a_t = \frac{M_2}{2} h \tag{28}$$

Algoritmi numerici - funcţie dată prin cod

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

$$|e_r| \le \frac{|e_{y_{k+1}}| + |e_{y_k}|}{h} \le \frac{\mathrm{eps}(|y_{k+1}| + |y_k|)}{h} \tag{29}$$

Dacă $|f(x)| \le M_0$ atunci

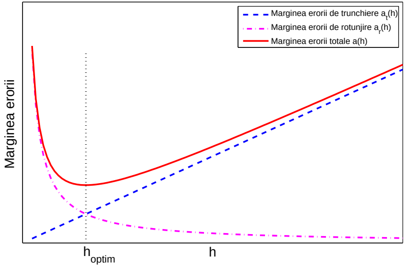
$$|e_r| \le \frac{2M_0\mathrm{eps}}{h} \overset{not}{=} a_r \tag{30}$$

Algoritmi numerici - funcţie dată prin cod

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

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

$$m(h) = \frac{M_2}{2} h + \frac{2 M_0 \text{eps}}{h^2} \qquad \min(m(h)) = m(h_{\text{optim}}). \tag{31}$$



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

$$m'(h) = 0$$

$\Rightarrow$

$$h_{\text{optim}} = 2\sqrt{\frac{M_0 \text{eps}}{M_2}}. \tag{32}$$

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
```

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.

$$\begin{aligned} g(x) &= \frac{(x-x_k)(x-x_{k+1})}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})}y_{k-1} + \frac{(x-x_{k-1})(x-x_{k+1})}{(x_k-x_{k-1})(x_k-x_{k+1})}y_k + \\ &+ \frac{(x-x_{k-1})(x-x_k)}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)}y_{k+1}, \end{aligned} \tag{33}$$

$$\begin{aligned} g'(x) &= \frac{(x-x_k)+(x-x_{k+1})}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})}y_{k-1} + \frac{(x-x_{k-1})+(x-x_{k+1})}{(x_k-x_{k-1})(x_k-x_{k+1})}y_k + \\ &+ \frac{(x-x_{k-1})+(x-x_k)}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)}y_{k+1}. \end{aligned} \tag{34}$$

$$\begin{aligned} g''(x) &= \frac{2}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})}y_{k-1} + \frac{2}{(x_k-x_{k-1})(x_k-x_{k+1})}y_k + \\ &+ \frac{2}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)}y_{k+1}, \end{aligned} \tag{35}$$

Derivate de ordin superior

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

$$g''(x) = \frac{y_{k-1} - 2y_k + y_{k+1}}{h^2}. \tag{36}$$

Derivate de ordin superior - evaluarea erorii

Seria Taylor cu un număr suplimentar de termeni:

$$f(x_{k+1}) = f(x_k) + f'(x_k)h + \frac{f''(x_k)}{2!}h^2 + \frac{f'''(x_k)}{3!}h^3 + \mathrm{O}(h^4), \quad (37)$$

$$f(x_{k-1}) = f(x_k) - f'(x_k)h + \frac{f''(x_k)}{2!}h^2 - \frac{f'''(x_k)}{3!}h^3 - \mathrm{O}(h^4). \quad (38)$$

$$f(x_{k+1}) + f(x_{k-1}) = 2f(x_k) + f''(x_k)h^2 + \mathrm{O}(h^4). \quad (39)$$

$$\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} - f''(x_k) = \mathrm{O}(h^2). \quad (40)$$

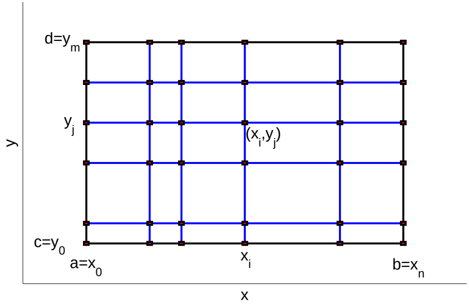
Derivate de ordin superior - evaluarea erorii

- 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":

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

$$i=\overline{0,n} \quad j=\overline{0,m}$$

Derivate parţiale

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

$$\left.\frac{\partial f}{\partial x}\right|_{x=x_0, y=y_0} = \lim_{x \to x_0} \frac{f(x, y_0) - f(x_0, y_0)}{x - x_0} = \lim_{x \to x_0} \frac{f1(x) - f1(x_0)}{x - x_0} = \frac{\mathrm{d}f1}{\mathrm{d}x}, \tag{41}$$

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

$$\left.\frac{\partial f}{\partial y}\right|_{x=x_0, y=y_0} = \lim_{y \to y_0} \frac{f(x_0, y) - f(x_0, y_0)}{y - y_0} = \lim_{y \to y_0} \frac{f2(y) - f2(y_0)}{y - y_0} = \frac{\mathrm{d}f2}{\mathrm{d}y}, \tag{42}$$

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

$$\frac{\partial g}{\partial x}(x_i, y_j) = \frac{z_{i+1,j} - z_{i,j}}{x_{i+1} - x_i}, \tag{43}$$

$$\frac{\partial g}{\partial y}(x_i, y_j) = \frac{z_{i,j+1} - z_{i,j}}{y_{j+1} - y_j} \tag{44}$$

etc.

Derivate parţiale

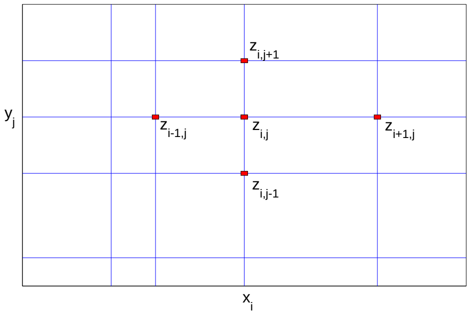
Exemplu: evaluarea numerică a operatorului Laplace aplicat unei funcţii de două varibile.

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \tag{45}$$

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

$$\begin{aligned} \Delta g &= \frac{z_{i-1,j} - 2z_{i,j} + z_{i+1,j}}{h^2} + \frac{z_{i,j-1} - 2z_{i,j} + z_{i,j+1}}{h^2} = \\ &= \frac{z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1} - 4z_{i,j}}{h^2}, \end{aligned} \tag{46}$$

Derivate parţiale



$$\Delta g = \frac{z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1} - 4z_{i,j}}{h^2} \quad (47)$$

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*.

Lectură obligatorie

- Cap.7 din

[1] Gabriela Ciuprina, Mihai Rebican, Daniel Ioan - Metode numerice in ingineria electrica - Indrumar de laborator pentru studentii facultatii de Inginerie electrica, Editura Printech, 2013, disponibil la http://mn.lmn.pub.ro/indrumar/IndrumarMN_Printech2013.pdf