

Derivarea numerică. Introducere în metoda diferențelor finite.

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" București, Facultatea de Inginerie Electrică

Suport didactic pentru disciplina *Metode numerice în ingineria electrică*, 2017-2018

Cuprins

- 1 Introducere
 - Importanța evaluării derivatelor
 - Formularea problemei derivării numerice
- 2 Prima derivată (derivată totală de ordinul 1)
 - Formule de derivare numerică - cu eroare de ordinul 1
 - Formule de derivare numerică - cu eroare de ordinul 2
 - Algoritmi
- 3 Alte derivate
 - Derivate (totale) de ordin superior
 - Derivate parțiale

Notes

Notes

Importanța evaluării derivatelor

- Relații utile pentru evaluarea unor mărimi:

$$i(t) = -\frac{dq_{D\Sigma}}{dt}$$

$$u(t) = -\frac{d\phi_{S_T}}{dt}$$

- Evaluarea sensibilităților:

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

Ideea: **derivarea numerică se bazează pe interpolare.**

- Ecuații sau sisteme de ecuații diferențiale

⇒ **Metoda diferențelor finite**

Notes

Formularea problemei - cazul cel mai simplu

Se dă funcția $f : [a, b] \rightarrow \mathbb{R}$ (cunoscută prin date sau prin cod)

Se cere evaluarea numerică a derivatei $f'(x_0)$, unde $x_0 \in [a, b]$.

Matematic:

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

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 funcției și

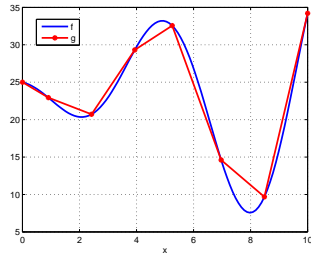
$$g'(x) = \frac{dg}{dx} \approx f'(x) = \frac{df}{dx}. \quad (2)$$

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

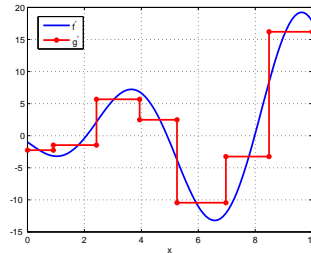
Notes

Formule de derivare numerică - cu eroare de ordinul 1

g - interpolare liniară pe porțiuni.



Funcția reală f și interpolarea ei pe porțiuni g .



Aproximarea derivatei f' cu valoarea g' .

Notes

Formule de derivare numerică - cu eroare de ordinul 1

Derivare progresivă de ordinul 1:

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

Derivare regresivă de ordinul 1:

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

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

Notes

Formule de derivare numerică - cu eroare de ordinul 1

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

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

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

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

Eroarea de trunchiere

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

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

Notăm $h = x_{k+1} - x_k$

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

Formule de derivare numerică - cu eroare de ordinul 1

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

$$|e_t| \leq \frac{M_2}{2} h, \quad (9)$$

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

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

Notes

Notes

Formule de derivare numerică - cu eroare de ordinul 1

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

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + O((x - x_k)^2) \quad (10)$$

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

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

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

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

Notes

Formule de derivare numerică - cu eroare de ordinul 2

Dezavantajul g - lpp: nu este derivabilă în noduri.

Remediu: creșterea gradului polinomului de interpolare.

Polinom de interpolare de grad 2:

$$g(x) = \frac{(x - x_k)(x - x_{k+1})}{(x_{k-1} - x_k)(x_{k-1} - x_{k+1})}y_{k-1} + \frac{(x - x_{k-1})(x - x_{k+1})}{(x_k - x_{k-1})(x_k - x_{k+1})}y_k + \frac{(x - x_{k-1})(x - x_k)}{(x_{k+1} - x_{k-1})(x_{k+1} - x_k)}y_{k+1}, \quad (13)$$

$$g'(x) = \frac{(x - x_k) + (x - x_{k+1})}{(x_{k-1} - x_k)(x_{k-1} - x_{k+1})}y_{k-1} + \frac{(x - x_{k-1}) + (x - x_{k+1})}{(x_k - x_{k-1})(x_k - x_{k+1})}y_k + \frac{(x - x_{k-1}) + (x - x_k)}{(x_{k+1} - x_{k-1})(x_{k+1} - x_k)}y_{k+1}. \quad (14)$$

Notes

Formule de derivare numerică - cu eroare de ordinul 2

Notăm: $h_1 = x_k - x_{k-1}$ și $h_2 = x_{k+1} - x_k$.

Evaluând polinomul derivat (34) în x_k se obține 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} și x_{k+1} se obține formula de **derivare progresivă de ordinul 2**

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

și, respectiv, de **derivare regresivă de ordinul 2**

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

Notes

Formule de derivare numerică - cu eroare de ordinul 2

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

Derivare centrată

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

Derivare progresivă

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

Derivare regresivă

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

Notes

Formule de derivare numerică - cu eroare de ordinul 2

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

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

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

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

⇒

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

⇒

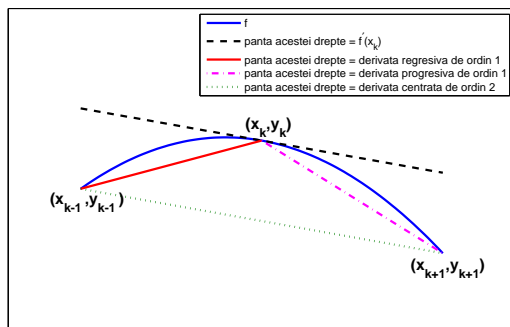
$$\frac{y_{k+1} - y_{k-1}}{2h} - f'(x_k) = O(h^2). \quad (25)$$

Formula de derivare centrată are eroarea de trunchiere de ordinul 2

Notes

Formule de derivare numerică

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



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

Notes

Modul de definire al funcției

Este important modul în care este cunoscută funcția.

- **Funcția dată tabelar (prin date)**
 - 1 se evaluează derivatele în punctele din tabel
 - 2 se evaluează derivata într-un punct oarecare, făcând o interpolare liniară pe porțiuni a acestor valori.
- **Funcția 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 - funcție dată tabelar

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

```
procedură pregătire_derivate(n, x, y, dy)
; calculează derivatele numerice în noduri
întreg n ; numărul de puncte din tabel minus 1
tablou_real x[n], y[n] tabelul de valori, indici de la 0
tablou_real dy[n] ; valorile derivatelor
...
; la capătul din stânga diferențe progresive de ordinul 2
h1 = x1 - x0
h2 = x2 - x1
dy0 = -(2h1 + h2)/(h1(h1 + h2))dy0 + (h1 + h2)/(h1h2)dy1 - h1/(h2(h1 + h2))dy2
; la capătul din dreapta diferențe regresive de ordinul 2
h1 = xn-1 - xn-2
h2 = xn - xn-1
dyn = -h2/(h1(h1 + h2))dyn-2 - (h1 + h2)/(h1h2)dyn-1 + (h1 + 2h2)/(h2(h1 + h2))dyn
; în nodurile interioare diferențe centrate de ordinul 2
pentru k = 1, n - 1
    h1 = xk - xk-1
    h2 = xk+1 - xk
    dyk = -h2/(h1(h1 + h2))dyk-1 - (h1 - h2)/(h1h2)dyk + h1/(h2(h1 + h2))dyk+1
```

rez = interpolare_lpp(n, x, dy, xcr)

Notes

Notes

Algoritmi numerici - funcție dată prin cod

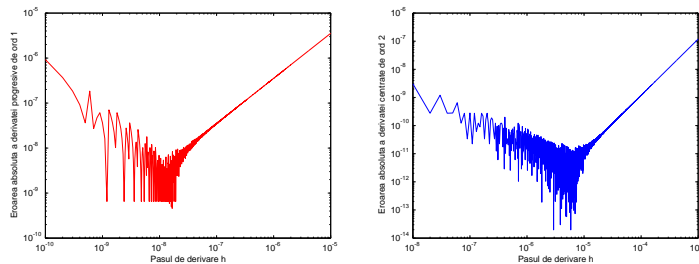
- Pasul de discretizare trebuie ales de utilizator.
- Eroarea de trunchiere - cu atât mai mică cu cât pasul este mai mic.

Dar,

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

Notes

Algoritmi numerici - funcție dată prin cod



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

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

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

Notes

Algoritmi numerici - funcție dată prin cod

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

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

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

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

$$a_t = \frac{M_2}{2}h \quad (28)$$

Notes

Algoritmi numerici - funcție dată prin cod

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

$$|e_r| \leq \frac{|e_{y_{k+1}}| + |e_{y_k}|}{h} \leq \frac{\text{eps}(|y_{k+1}| + |y_k|)}{h} \quad (29)$$

Dacă $|f(x)| \leq M_0$ atunci

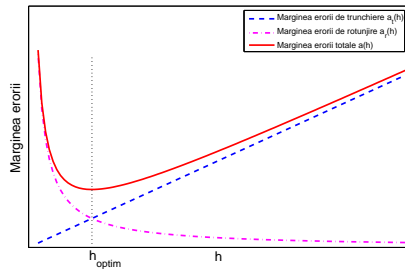
$$|e_r| \leq \frac{2M_0\text{eps}}{h} \stackrel{\text{not}}{=} a_r \quad (30)$$

Notes

Algoritmi numerici - funcție dată prin cod

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

$$|e| = |e_t + e_r| \leq |e_t| + |e_r| \leq a_t + a_r = m(h)$$
$$m(h) = \frac{M_2}{2}h + \frac{2M_0\text{eps}}{h^2} \quad \min(m(h)) = m(h_{\text{optim}}). \quad (31)$$



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

Algoritmi numerici - funcție dată prin cod

Condiția de minim

$$m'(h) = 0$$

⇒

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

Pasul optim de derivare depinde de:

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

Este pasul pentru care marginea erorii totale este minimă și 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.

Notes

Notes

Algoritmi numerici - funcție dată prin cod

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

```
funcție derivata_f(x, h0, maxit)
; calculează derivata numerică a funcției f folosind diferențe progresive de ordinul 1 și pas optim de derivare
real x ; punctul în care se va evalua derivata
real h0 ; pasul inițial
intreg maxit ; numărul maxim de iterații pentru calculul pasului
...
eps = zeroul_mașinii ()
f0 = f(x) ; valoarea funcției 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 ; funcția e liniară, eroarea de trunchiere e zero
    altfel
        M0 = max(|f0|, |f1|, |f2|) ; estimarea marginii funcției
        hoptim = 2*sqrt(M0eps/M2)
    r = h / hoptim ; rata de modificare a pasului
    h = hoptim
până când (k > maxit) sau (r > 0.5 și r < 2)
întoarce (f(x + hoptim) - f0) / hoptim
```

Notes

Derivate de ordin superior

$f''(x) = ?$

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

pregătire_derivare(n, x, dy, d2y)

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

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

Notes

Derivate de ordin superior

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

$$g(x) = \frac{(x-x_k)(x-x_{k+1})}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})}y_{k-1} + \frac{(x-x_{k-1})(x-x_{k+1})}{(x_k-x_{k-1})(x_k-x_{k+1})}y_k + \frac{(x-x_{k-1})(x-x_k)}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)}y_{k+1}, \quad (33)$$

$$g'(x) = \frac{(x-x_k) + (x-x_{k+1})}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})}y_{k-1} + \frac{(x-x_{k-1}) + (x-x_{k+1})}{(x_k-x_{k-1})(x_k-x_{k+1})}y_k + \frac{(x-x_{k-1}) + (x-x_k)}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)}y_{k+1}. \quad (34)$$

$$g''(x) = \frac{2}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})}y_{k-1} + \frac{2}{(x_k-x_{k-1})(x_k-x_{k+1})}y_k + \frac{2}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)}y_{k+1}, \quad (35)$$

Derivate de ordin superior

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

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

Notes

Notes

Derivate de ordin superior - evaluarea erorii

Seria Taylor cu un număr suplimentar de termeni:

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

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

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

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

Derivate de ordin superior - evaluarea erorii

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

Notes

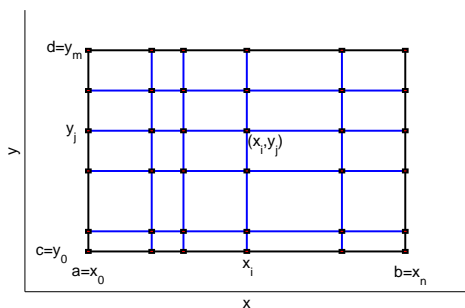
Notes

Derivate parțiale

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

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

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



Valorile din
"tabel":

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

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

Notes

Derivate parțiale

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

$$\frac{\partial f}{\partial x} \Big|_{x=x_0, y=y_0} = \lim_{x \rightarrow x_0} \frac{f(x, y_0) - f(x_0, y_0)}{x - x_0} = \lim_{x \rightarrow x_0} \frac{f_1(x) - f_1(x_0)}{x - x_0} = \frac{df_1}{dx}, \quad (41)$$

unde $f_1(x) = f(x, y_0)$ este o funcție care depinde de o singură variabilă reală. Similar,

$$\frac{\partial f}{\partial y} \Big|_{x=x_0, y=y_0} = \lim_{y \rightarrow y_0} \frac{f(x_0, y) - f(x_0, y_0)}{y - y_0} = \lim_{y \rightarrow y_0} \frac{f_2(y) - f_2(y_0)}{y - y_0} = \frac{df_2}{dy}, \quad (42)$$

unde $f_2(y) = f(x_0, y)$ este o funcție care depinde de o singură variabilă reală.

Notes

Derivate parțiale

Formulele de derivare progresivă de ordinul 1

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

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

etc.

Notes

Derivate parțiale

Exemplu: evaluarea numerică a operatorului Laplace aplicat unei funcții de două variabile.

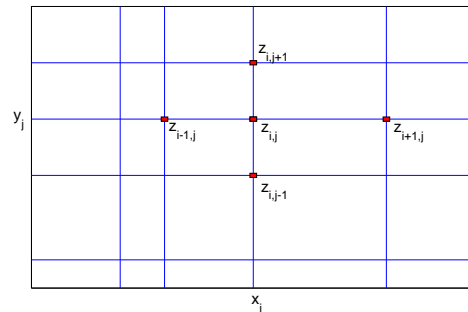
$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \quad (45)$$

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

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

Notes

Derivate parțiale



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

Dacă astfel de raționamente se folosesc pentru rezolvarea numerică a problemelor de inginerie formulate cu ajutorul unor ecuații cu derivate totale (ODE) sau parțiale (PDE) atunci se spune că pentru rezolvarea problemei se aplică *metoda diferențelor 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

Notes

Notes
