

**MDF (continuare): Model 1D, distribuit, al unui microcomutator acționat electrostatic.**  
Analiza în regim staționar elastostatic cuplat cu electrostatic.

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" București, Facultatea de Inginerie Electrică

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

Notes

---

---

---

---

---

---

---

---

---

---

---

---

## Cuprins

- 1 **Introducere**
  - Descrierea problemei
  - Formularea matematică
- 2 **Discretizarea ecuațiilor folosind MDF**
  - Discretizarea domeniului
  - Discretizarea ecuației
- 3 **Rezolvarea sistemului discretizat**
  - Sistem algebric neliniar
  - Metoda Newton
- 4 **Implementare și validare**
  - Implementare în Matlab
  - Validare

Notes

---

---

---

---

---

---

---

---

---

---

---

---

## Descrierea problemei

- 1 Înainte de a parcurge aceste slide-uri, trebuie să revedeți contextul descris la studiul de caz prezentat în [http://an.lmn.pub.ro/slides2017/06b\\_AN.pdf](http://an.lmn.pub.ro/slides2017/06b_AN.pdf)  
Acolo armătura superioară a condensatorului era rigidă, cu posibilitatea de a se mișca doar prin translație. Modelul 1D era **concentrat** într-un punct.
- 2 Acum, armătura superioară se consideră elastică, cu următorii parametri considerați cunoscuți:
  - geometrie: lungime  $l$ , lățime  $w$ , înălțime  $h$ ;
  - material: modul de elasticitate longitudinal  $E$ ;În plus, armătura superioară se consideră încastrată la ambele capete și are tensiuni inițiale mecanice distribuite uniform și constante pe parcursul deformării, descrise de mărimea locală  $S$ .

Notes

---

---

---

---

---

---

---

---

---

---

## Descrierea problemei

La aplicarea unei tensiuni electrice între armăturile unui astfel de condensator, armătura superioară este atrasă de armătura fixă, dar ea se deformează, capetele ei fiind fixate.

Vom rezolva model 1D **distribuit** de-a lungul armăturii folosind:

- 1 MDF pentru discretizarea PDE  $\Rightarrow$  sistem de ecuații algebrice neliniare;
- 2 metoda Newton pentru rezolvarea sistemului algebric asamblat.

Notes

---

---

---

---

---

---

---

---

---

---

## Formularea problemei

**Ecuțiile** care descriu fenomenele:

- Structural, elastostatic (MEC) - Ecuția Euler-Bernoulli

$$EI \frac{\partial^4 u}{\partial x^4} - S \frac{\partial^2 u}{\partial x^2} = F_{ES} \quad (1)$$

unde  $I = wh^3/12$

- Electrostatic (ES) :

$$F_{ES} = -\frac{\varepsilon_0 wV^2}{2u^2} \quad (2)$$

Vom rezolva problema făcând un cuplaj strâns între fenomene.

Notes

---

---

---

---

---

---

---

---

---

---

## Formularea problemei

**Ecuțiile** care descriu fenomenele:

- (MEC) + (ES)

$$EI \frac{\partial^4 u}{\partial x^4} - S \frac{\partial^2 u}{\partial x^2} = -\frac{\varepsilon_0 wV^2}{2u^2} \quad (3)$$

și rearanjat

- (MEC) + (ES)

$$u^2 \left( S \frac{\partial^2 u}{\partial x^2} - EI \frac{\partial^4 u}{\partial x^4} \right) - \frac{\varepsilon_0 wV^2}{2} = 0 \quad (4)$$

unde  $I = wh^3/12$ .

Notes

---

---

---

---

---

---

---

---

---

---

## Formularea problemei

### Ecuatii și condiții de frontieră

- (MEC) + (ES)

$$u'' \left( S \frac{\partial^2 u}{\partial x^2} - EI \frac{\partial^4 u}{\partial x^4} \right) - \frac{\varepsilon_0 w v^2}{2} = 0 \quad (5)$$

unde  $I = wh^3/12$ .

- Condiții la capetele domeniului

$$u(0) = u0 \quad (6)$$

$$u(l) = u0 \quad (7)$$

$$\frac{du}{dx}(0) = 0 \quad (8)$$

$$\frac{du}{dx}(l) = 0 \quad (9)$$

unde  $u0$  este distanța inițială dintre armături. 7/28

Notes

---

---

---

---

---

---

---

---

---

---

## Formularea problemei

### Se dau:

- Geometria:  $l, w, h, u0$ ;
- Materialele: (MEC):  $E$ , (ES):  $\varepsilon_0$ ;
- Sursele: (MEC):  $S$ , (ES):  $v$ ;
- Condițiile de frontieră: (MEC) încastrare la ambele capete;  
(ES) între armătura superioară, mobilă și cea inferioară,  
fixă se aplică o tensiune  $v$ .

### Se cere:

- Deplasarea armăturii  $u(x)$  pentru o tensiune  $v$  fixată,  
suficient de mică astfel încât comutatorul să nu se închidă;
- Analiza deplasarea armăturii  $u(x)$  pentru o gamă de  
tensiuni  $v$  crescătoare, începând de la valoarea 0;
- Tensiunea minimă de acționare  $V_{pi}$ .

Notes

---

---

---

---

---

---

---

---

---

---

## Discretizarea domeniului

$$x \in [0, l]$$

$n$  puncte interioare distribuite uniform

$$x_0 = 0 \quad x_1 = \Delta x \quad x_2 = 2\Delta x \quad \dots \quad x_n = n\Delta x \quad x_{n+1} = l$$

$$\Delta x = \frac{l}{n+1}$$

Notes

---

---

---

---

---

---

---

---

---

---

---

---

## Discretizarea ecuației

**Ecuatii** și **condiții de frontieră**

- (MEC) + (ES)

$$u^2 \left( S \frac{\partial^2 u}{\partial x^2} - EI \frac{\partial^4 u}{\partial x^4} \right) - \frac{\varepsilon_0 w v^2}{2} = 0 \quad (10)$$

unde  $l = wh^3/12$ .

- Condiții la capetele domeniului

$$u(0) = g_0 \quad (11)$$

$$u(l) = g_0 \quad (12)$$

$$\frac{du}{dx}(0) = 0 \quad (13)$$

$$\frac{du}{dx}(l) = 0 \quad (14)$$

unde  $g_0 = u_0$  este distanța inițială (*gap*) dintre armături.

Notes

---

---

---

---

---

---

---

---

---

---

---

---

## Derivata de ordin 2

Avem nevoie de formule de derivare numerică de ordinul 2 și de ordinul 4 pentru rețele uniforme:

$$\frac{d^2 f}{dx^2} = \frac{1}{(\Delta x)^2} (f_{i-1} - 2f_i + f_{i+1}), \quad i = 1, n \quad (15)$$

În cazul nostru este convenabil să procedăm astfel:

$$\begin{aligned} \frac{d^2 u}{dx^2}(x_i) &= \frac{d^2(u - u_0)}{dx^2}(x_i) = \\ &= \frac{1}{(\Delta x)^2} ((u_{i-1} - u_0) - 2(u_i - u_0) + (u_{i+1} - u_0)), \\ &\quad i = 1, n \end{aligned} \quad (16)$$

Acest artificiu ne permite să înlocuim cu ușurință condițiile la capete, în această ecuație.

Notes

---

---

---

---

---

---

---

---

---

---

## Derivata de ordin 2 + condiții de frontieră

$$i = 2, n - 1$$

$$\frac{d^2 u_i}{dx^2} = \frac{1}{(\Delta x)^2} ((u_{i-1} - u_0) - 2(u_i - u_0) + (u_{i+1} - u_0)) \quad (17)$$

$$i = 1$$

$$\frac{d^2 u_1}{dx^2} = \frac{1}{(\Delta x)^2} (-2(u_1 - u_0) + (u_2 - u_0)) \quad (18)$$

$$i = n$$

$$\frac{d^2 u_n}{dx^2} = \frac{1}{(\Delta x)^2} ((u_{n-1} - u_0) - 2(u_n - u_0)) \quad (19)$$

Notes

---

---

---

---

---

---

---

---

---

---

## Derivata de ordin 2 + condiții de frontieră

Scriere compactă

$$\mathbf{D2} = \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 & 0 & \dots & 0 & 0 & 0 \\ & & \dots & \dots & \dots & & & & \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 & -2 \end{bmatrix} \quad \frac{d^2 \mathbf{u}}{dx^2} = \begin{bmatrix} \frac{d^2 u_1}{dx^2} \\ \frac{d^2 u_2}{dx^2} \\ \vdots \\ \frac{d^2 u_n}{dx^2} \end{bmatrix} \quad (20)$$

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \quad \mathbf{u} - \mathbf{u}_0 = \begin{bmatrix} u_1 - u_0 \\ u_2 - u_0 \\ \vdots \\ u_n - u_0 \end{bmatrix} \Rightarrow \frac{d^2 \mathbf{u}}{dx^2} = \mathbf{D2} * (\mathbf{u} - \mathbf{u}_0) \quad (21)$$

Notes

---

---

---

---

---

---

---

---

---

---

---

---

## Derivata de ordin 4

Pentru rețele uniforme:

$$\frac{d^4 f}{dx^4} = \frac{1}{(\Delta x)^4} (f_{i-2} - 4f_{i-1} + 6f_i - 4f_{i+1} + f_{i+2}) \quad (22)$$

Este convenabil să procedăm astfel:

$$\begin{aligned} \frac{d^4 u}{dx^4}(x_i) &= \\ &= \frac{1}{(\Delta x)^4} ((u_{i-2} - u_0) - 4(u_{i-1} - u_0) + 6(u_i - u_0) - 4(u_{i+1} - u_0) + (u_{i+2} - u_0)) \\ &\quad i = 2, n-1 \end{aligned} \quad (23)$$

Notes

---

---

---

---

---

---

---

---

---

---

---

---

## Derivata de ordin 4 + condiții de frontieră

Particularizăm formula pentru

- $i = 2$  și impunem condiția de frontieră  $u(0) = u_0$
- $i = n$  și impunem condiția de frontieră  $u(l) = u_0$
- $i = 1$  și impunem condiția de frontieră  
 $du/dx(0) = 0 \Rightarrow u_{-1} = u_1$
- $i = n + 1$  și impunem condiția de frontieră  
 $du/dx(l) = 0 \Rightarrow u_{n+2} = u_n$

Notes

---

---

---

---

---

---

---

---

---

---

---

---

## Derivata de ordin 4 + condiții de frontieră

$i = 3, n-2$

$$\begin{aligned} \frac{d^4 u_i}{dx^4} &= \\ &= \frac{1}{(\Delta x)^4} ((u_{i-2} - u_0) - 4(u_{i-1} - u_0) + 6(u_i - u_0) - 4(u_{i+1} - u_0) + (u_{i+2} - u_0)) \end{aligned} \quad (24)$$

$i = 2$

$$\frac{d^4 u_2}{dx^4} = \frac{1}{(\Delta x)^4} (-4(u_1 - u_0) + 6(u_2 - u_0) - 4(u_3 - u_0) + (u_4 - u_0)) \quad (25)$$

$i = n-1$

$$\frac{d^4 u_{n-1}}{dx^4} = \frac{1}{(\Delta x)^4} ((u_{n-3} - u_0) - 4(u_{n-2} - u_0) + 6(u_{n-1} - u_0) - 4(u_n - u_0)) \quad (26)$$

Notes

---

---

---

---

---

---

---

---

---

---

---

---



## Derivata de ordin 4 + condiții de frontieră

$i = 3, n-2$

$$\begin{aligned} \frac{d^4 u_i}{dx^4} &= \\ &= \frac{1}{(\Delta x)^4} ((u_{i-2} - u_0) - 4(u_{i-1} - u_0) + 6(u_i - u_0) - 4(u_{i+1} - u_0) + (u_{i+2} - u_0)) \end{aligned} \quad (27)$$

$i = 1$

$$\frac{d^4 u_1}{dx^4} = \frac{1}{(\Delta x)^4} (7(u_1 - u_0) - 4(u_2 - u_0) + (u_3 - u_0)) \quad (28)$$

$i = n$

$$\frac{d^4 u_n}{dx^4} = \frac{1}{(\Delta x)^4} ((u_{n-2} - u_0) - 4(u_{n-1} - u_0) + 7(u_n - u_0)) \quad (29)$$

Notes

---

---

---

---

---

---

---

---

---

---

---

---

## Derivata de ordin 4 + condiții de frontieră

Scriere compactă

$$D4 = \begin{bmatrix} 7 & -4 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ -4 & 6 & -4 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & -4 & 6 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 & -4 & 6 & -4 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & -4 & 7 \end{bmatrix} \frac{d^4 \mathbf{u}}{dx^4} = \begin{bmatrix} \frac{d^4 u_1}{dx^4} \\ \frac{d^4 u_2}{dx^4} \\ \vdots \\ \frac{d^4 u_n}{dx^4} \end{bmatrix} \quad (30)$$

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \quad \mathbf{u} - \mathbf{u}_0 = \begin{bmatrix} u_1 - u_0 \\ u_2 - u_0 \\ \vdots \\ u_n - u_0 \end{bmatrix} \Rightarrow \frac{d^4 \mathbf{u}}{dx^4} = D4 * (\mathbf{u} - \mathbf{u}_0) \quad (31)$$

Notes

---

---

---

---

---

---

---

---

---

---

---

---

## Sistemul algebric neliniar obținut

$$\mathbf{F}(\mathbf{u}) = \mathbf{0} \quad (32)$$

unde

$$\mathbf{F}(\mathbf{u}) = (\mathbf{u} * \mathbf{u}) * (\mathbf{S} * \mathbf{D2} * (\mathbf{u} - \mathbf{u0}) - \mathbf{E} * \mathbf{I} * \mathbf{D4} * (\mathbf{u} - \mathbf{u0})) - \frac{\varepsilon_0 W V^2}{2} * \mathbf{1} \quad (33)$$

unde

$$\mathbf{1} = \text{ones}(n, 1) \quad \mathbf{u} * \mathbf{u} = \begin{bmatrix} u_1^2 \\ u_2^2 \\ \vdots \\ u_n^2 \end{bmatrix}$$

Ecuția obținută este **algebrică, neliniară**

Notes

---

---

---

---

---

---

---

---

---

---

## Metoda Newton

$$\mathbf{F}(\mathbf{u}) = \mathbf{0}$$

$$\mathbf{u}^{(n)} = \mathbf{u}^{(v)} - \left( \mathbf{F}'(\mathbf{u}^{(v)}) \right)^{-1} \mathbf{F}(\mathbf{u}^{(v)})$$

Algoritm - la fiecare iterație

- 1 Se calculează corecția  $\mathbf{z}$  prin rezolvarea sistemului

$$\mathbf{F}'(\mathbf{u}^{(v)})\mathbf{z} = -\mathbf{F}(\mathbf{u}^{(v)})$$

- 2 Se actualizează soluția

$$\mathbf{u}^{(n)} = \mathbf{u}^{(v)} + \mathbf{z}$$

Notes

---

---

---

---

---

---

---

---

---

---

## Calculul Jacobianului

$$\mathbf{F}'(\mathbf{u}^{(v)}) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_n} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \cdots & \frac{\partial f_n}{\partial u_n} \end{bmatrix} = ?$$

## Implementare în Matlab

Codul (comentat) este disponibil pe pagina cursului.

```
1 % program principal
2 date = citeste_date_punte();
3 date = genereaza_grid_si_matriceDER(date);
4 [out] = solve_static_beam(date);
```

Notes

---

---

---

---

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

---

---

---

---

## Implementare în Matlab

```
1 function date = citeste_date_punte ()
2 %% geometrie
3 date.l = 610e-6; % [m] lungimea membranei
4 date.w = 40e-6; % [m] latimea membranei
5 date.h = 2.2e-6; % [m] inaltimea membranei
6 date.u0 = 2.3e-6; % [m] gapul initial
7 %% materiale
8 date.E = 149e9; % [Pa] modulul lui Young
9 date.S_overHW = -3.7e6; % tensiuni reziduale [Pa] (unde S = stress coefficient [N])
10 date.epsilon0 = 8.854187817e-12; % F/m - permitivitatea vidului
11 %% tensiunea de actionare
12 date.v = 8; % V - valoare folosita daca se va face simularea pentru o anumita valoare
    a tensiunii
13 %% parametrii pentru procedura neliniara (Newton)
14 date.err_impus = 1e-3; % eroarea impusa procedurii neliniare
15 date.it_max = 100; % nr max de pasi pentru procedura iterativa neliniara
16 %% finetea gridului de discretizare
17 date.n = 40; % uniform pe OX, reprezinta nr de grade de libertate ale modelului
18 end
```

23/28

Notes

---

---

---

---

---

---

---

---

---

---

---

---

## Implementare în Matlab

```
1 function date = genereaza_grid_si_matriceDER (date)
2 %%
3 n = date.n;
4 l = date.l;
5 %
6 dx = l/(n+1);
7 dx2 = dx^2;
8 dx4 = dx^2^2;
9 date.dx = dx;
10 date.dx2 = dx2;
11 date.dx4 = dx4;
12 date.doidx = 2*dx;
13 % grid
14 date.x = linspace(0,l,n+2)';
15 %%
16 % derivata de ordinul 2 dupa x, se va aplica deplasarii (u-u0)
17 e = ones(n,1);
18 DER2 = spdiags([e -2*e e], -1:1, n, n);
19 % derivata de ordiul 4, centrata, dupa x, se va aplica deplasarii (u-u0)
20 DER2 = DER2/dx2;
21 DER4 = spdiags([e -4*e 6*e -4*e e], -2:2, n, n);
22 DER4(1,1) = 7;
23 DER4(n,n) = 7;
24 DER4 = DER4/dx4;
25 date.DER2 = DER2;
26 date.DER4 = DER4;
27 end
```

24/28

Notes

---

---

---

---

---

---

---

---

---

---

---

---

## Implementare în Matlab

```
1 function [out] = solve_static_beam(date)
2 %%
3 out.errorCode = 0;
4 out.v = date.v; out.x = date.x;
5 out.u = []; % se va completa dupa rezolvare
6 out.gapMin = []; % valoarea la mijlocul barei
7 %
8 date.S = date.S_overHW*date.h*date.w; %[Pa*m^2]
9 l = date.w*date.h^3/12;
10 date.EI = date.E*I;
11 date.v2 = date.v^2; %v^2
12 u0 = date.u0; n = date.n;
13 sol_v = u0*ones(n,1); %initializare pentru Newton
14 err_impus = date.err_impus; it_max = date.it_max;
15 pozitie_mijloc = floor(n/2) + 1; % indexul punctului din mijlocul barei
16 u0 = date.u0;
17 l = date.l;
18 err = 2*err_impus; knelin = 0;
19 while and((err > err_impus), knelin < it_max)
20     knelin = knelin + 1;
21     [Fdeu, jacobianF] = f_fder_static_matrix(sol_v, date);
22     corectie = -jacobianF\Fdeu;
23     sol_n = sol_v + corectie;
24     err = norm(corectie)/norm(sol_v);
25     fprintf(' corectia relativa = %e\n', err);
26     sol_v = sol_n; % actualizare solutie veche
27 end
28 % vedeti codul complet pe pagina cursului
```

Notes

---

---

---

---

---

---

---

---

---

---

## Implementare în Matlab

```
1 function [rhs, jacobian] = f_fder_static_matrix(u, date)
2 % calculeaza membrul drept si jacobianul sistemului de rezolvat
3 u0 = date.u0; epsilon0 = date.epsilon0;
4 w = date.w; v2 = date.v2;
5 S = date.S; EI = date.EI;
6 %%
7 n = date.n;
8 dx2 = date.dx2;
9 dx4 = date.dx4;
10 %%
11 upatrat = u.^2;
12 minusu0 = u - u0;
13 DER2 = date.DER2;
14 DER4 = date.DER4;
15 % membrul drept al sistemului de rezolvat F(uv)
16 rhs1 = S*DER2*minusu0 - EI*DER4*minusu0;
17 rhs = upatrat.*rhs1 - epsilon0*w*v2/2;
18 % jacobianul membrului drept F'(uv)
19 Sdx2 = S/dx2;
20 Eldx4 = EI/dx4;
21 diag0 = 2*u.*rhs1 + upatrat*(Sdx2*(-2) - Eldx4*6);
22 diag0(1) = diag0(1) + upatrat(1)*(-Eldx4); % ca sa se faca 7 in loc de 6
23 diag0(end) = diag0(end) + upatrat(end)*(-Eldx4);
24 diag1 = upatrat*(Sdx2 + Eldx4*4);
25 diag2 = upatrat*(-Eldx4);
26 jacobian = spdiags([diag2 diag1 diag0 diag1 diag2], [-2 -1 0 1 2], n, n);
27 jacobian = jacobian'; % spdiags il creeaza transpus, si nu e matrice simetrica
28 end
```

Notes

---

---

---

---

---

---

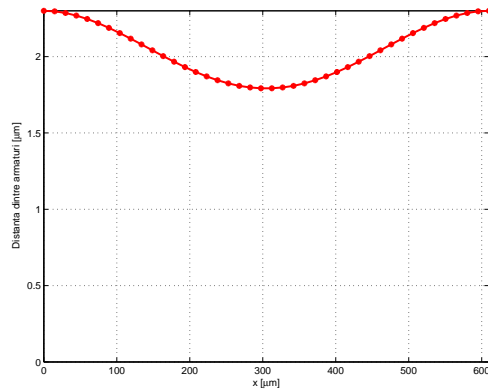
---

---

---

---

## Implementare în Matlab



## Temă

### Temă facultativă

- Investigați convergența rezolvării sistemului neliniar de ecuații pentru diferite valori ale tensiunii aplicate. și pentru diferite discretizări ale domeniului.
- Modificați codul astfel încât să puteți detecta automat tensiunea minimă de acționare.
- Ce modificări ar trebui să aduceți implementării astfel încât să exploatați simetria problemei.
- Rezolvați ecuația cu un alt software (de exemplu COMSOL) și comparați soluțiile (deformarea la o anumită tensiune, tensiunea minimă de acționare).
- Redactați un raport relevant.

Notes

---

---

---

---

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

---

---

---

---