UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
MODELAGEM TERMODINAMICAMENTE CONSISTENTE DE MATERIAIS

# Atividade 2
# Propagação de uma trinca em uma barra 1D de seção variável

Renan Miranda Portela

3 de dezembro de 2018

## 1 OBJETIVO

O objetivo desta atividade é verificar a propagação do dano em uma seção I tracionada na ausência de um dano concentrado inicial e através das equações de movimento e dano.

## 2 METODOLOGIA

A equação 2.1 é referente a equação do movimento.

$$\rho_0 \ddot{u}_0 = div_p[F(1-\varphi)^2 KE] + \rho_0 f_0 \tag{2.1}$$

No caso de um movimento quasi-estático e sem força de corpo, consideramos a aceleração e o termo $\rho_0 f_0$ como sendo nulos como mostra a equação 2.2.

$$0 = div_p[F(1-\varphi)^2 KE] \tag{2.2}$$

Fazendo a aproximação para pequenas deformações.

$$[F][E] \simeq [E] \tag{2.3}$$

Substituindo a equação 2.3 na equação 2.2

$$0 = div_p[(1-\varphi)^2 KE]$$

Aplicando resíduos ponderados para fazer a formulação fraca primeiramente

$$0 = \int_\Omega div_p[(1-\varphi)^2 KE]w(p)d\Omega$$

Aplicando integração por partes

$$\int_{\partial\Omega}(1-\varphi)^2 C:\epsilon(u_0)Nw(p)dS - \int_\Omega [(1-\varphi)^2 KE]\frac{dw(p)}{dp}d\Omega = 0$$

Consideramos o termo de fronteira como sendo o vetor força $\hat{f}$.

$$\hat{f} = \int_{\partial\Omega}(1-\varphi)^2 C:\epsilon(u_0)Nw(p)dS$$

Forçamos o vetor força como sendo igual a força no último nó como mostrado no vetor 2.4 de tamanho $nx1$ sendo $n$ o número de nós.

$$\hat{f} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ -1000 \end{bmatrix}_{nx1} \tag{2.4}$$

$$\int_\Omega [(1-\varphi_{n-1})^2 K\frac{\partial u_n}{\partial p}]\frac{dw(p)}{dp}d\Omega = \hat{f}$$

Tornando a formulação unidimensional, encontramos:

$$\int_L\int_{\Omega(p_1)} [(1-\varphi_{n-1})^2 K\frac{\partial u_n}{\partial p}]\frac{dw(p)}{dp}dp_1dp_2dp_3 = \hat{f}$$

$$\int_L [(1-\varphi_{n-1})^2 K\frac{\partial u_n}{\partial p}]\frac{dw(p)}{dp}dp_1\int_{\Omega(p_1)}dp_2dp_3 = \hat{f}$$

$$\int_L [(1-\varphi_{n-1})^2 K\frac{\partial u_n}{\partial p}]\frac{dw(p)}{dp}dp_1A(p_1) = \hat{f}$$

Sendo $K$ e $A$ consideradas constantes no elemento:

$$KA\int_L [(1-\varphi_{n-1})^2 \frac{\partial u_n}{\partial p}]\frac{dw(p)}{dp}dp_1 = \hat{f}$$

$$[K_g] = A_{i=1}^{nel}[K_e]$$

$$[K_g][\hat{u}] = \hat{f}$$

$$[K_e] = KA^e\int_L (1-[N][\hat{\varphi}_{n-1}])^2[B]^T[B]dp$$

$$[K_e] = \frac{KA^e}{L^e}\int_{-1}^{1}(1-[N]\hat{\varphi})^2[B]^T[B]det(J)d\xi \tag{2.5}$$

Para um elemento unidimensional de três nós:

$$[N] = [\frac{-1}{2}\xi(1-\xi),(1-\xi^2),\frac{1}{2}\xi(1+\xi)]$$

$$[B] = [\frac{-1}{2}+\xi,-2\xi,\frac{1}{2}+\xi]$$

$$det(J) = [B]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Apresentando a equação do dano (2.6).

$$\lambda_1\dot{\varphi} = div_p[g_c\gamma\bigtriangledown_p\varphi_0]+(1-\varphi_0)K|E|^2-\frac{g_c}{\gamma}\varphi_0 \tag{2.6}$$

Discretizando o termo do dano em relação ao tempo, encontramos (2.7).

$$\dot{\varphi} = \frac{\varphi_n-\varphi_{n-1}}{\Delta t} \tag{2.7}$$

$$\lambda_1\frac{\varphi_n-\varphi_{n-1}}{\Delta t} = div_p[g_c\gamma\bigtriangledown_p\varphi_n]+(1-\varphi_{n-1})K|E|^2-\frac{g_c}{\gamma}\varphi_n$$

$$\varphi_n-\varphi_{n-1} = \frac{\Delta t}{\lambda_1}div_p[g_c\gamma\bigtriangledown_p\varphi_n]+\frac{\Delta tK|E|^2}{\lambda_1}(1-\varphi_{n-1})-\frac{g_c\Delta t}{\gamma\lambda_1}\varphi_n$$

$$\varphi_n+\frac{g_c\Delta t}{\gamma\lambda_1}\varphi_n-\frac{g_c\gamma\Delta t}{\lambda_1}div_p[\bigtriangledown_p\varphi_n] = \varphi_{n-1}+\frac{\Delta tK|E|^2}{\lambda_1}(1-\varphi_{n-1})$$

$$\begin{aligned}\int_\Omega\varphi_n wd\Omega+\int_\Omega\frac{g_c\Delta t}{\gamma\lambda_1}\varphi_n wd\Omega-\int_\Omega\frac{g_c\gamma\Delta t}{\lambda_1}div_p[\bigtriangledown_p\varphi_n]wd\Omega = \\ \int_\Omega\varphi_{n-1}wd\Omega+\int_\Omega\frac{\Delta tK|E|^2}{\lambda_1}(1-\varphi_{n-1})wd\Omega\end{aligned}$$

$$\begin{aligned}\int_\Omega\varphi_n wd\Omega+\frac{g_c\Delta t}{\gamma\lambda_1}\int_\Omega\varphi_n wd\Omega-\frac{g_c\gamma\Delta t}{\lambda_1}\int_\Omega div_p[\bigtriangledown_p\varphi_n]wd\Omega = \\ \int_\Omega\varphi_{n-1}wd\Omega+\frac{\Delta tK}{\lambda_1}\int_\Omega|E|^2(1-\varphi_{n-1})wd\Omega\end{aligned}$$

$$\begin{aligned}\int_\Omega\varphi_n wd\Omega+\frac{g_c\Delta t}{\gamma\lambda_1}\int_\Omega\varphi_n wd\Omega-\frac{g_c\gamma\Delta t}{\lambda_1}[\int_{\partial\Omega}[\bigtriangledown_p\varphi_n.N]wdS-\int_\Omega[\bigtriangledown_p\varphi_n]\frac{dw(p)}{dp}d\Omega] = \\ \int_\Omega\varphi_{n-1}wd\Omega+\frac{\Delta tK}{\lambda_1}\int_\Omega|E|^2(1-\varphi_{n-1})wd\Omega\end{aligned}$$

Sendo o fluxo no contorno nulo como mostra na equação (2.8).

$$\int_{\partial\Omega} [\nabla_p \varphi_n . N] w dS = 0 \tag{2.8}$$

$$\int_\Omega \varphi_n w d\Omega + \frac{g_c \Delta t}{\gamma \lambda_1} \int_\Omega \varphi_n w d\Omega + \frac{g_c \gamma \Delta t}{\lambda_1} \int_\Omega [\nabla_p \varphi_n] \frac{dw(p)}{dp} d\Omega] = \int_\Omega \varphi_{n-1} w d\Omega + \frac{\Delta t K}{\lambda_1} \int_\Omega |E|^2 (1 - \varphi_{n-1}) w d\Omega$$

Fazendo a formulação unidimensional

$$\int_L \int_{\Omega(p_1)} \varphi_n w dp_1 dp_2 dp_3 + \frac{g_c \Delta t}{\gamma \lambda_1} \int_L \int_{\Omega(p_1)} \varphi_n w dp_1 dp_2 dp_3 + \frac{g_c \gamma \Delta t}{\lambda_1} \int_L \int_{\Omega(p_1)} [\nabla_p \varphi_n] \frac{dw(p)}{dp} dp_1 dp_2 dp_3 = \int_L \int_{\Omega(p_1)} \varphi_{n-1} w dp_1 dp_2 dp_3 + \frac{\Delta t K}{\lambda_1} \int_L \int_{\Omega(p_1)} |E|^2 (1 - \varphi_{n-1}) w dp_1 dp_2 dp_3$$

$$\int_L A \varphi_n w dp_1 + \frac{g_c \Delta t}{\gamma \lambda_1} \int_L A \varphi_n w dp_1 + \frac{g_c \gamma \Delta t}{\lambda_1} \int_L A [\nabla_p \varphi_n] \frac{dw(p)}{dp} dp_1 = \int_L A \varphi_{n-1} w dp_1 + \frac{\Delta t K}{\lambda_1} \int_L A |E|^2 (1 - \varphi_{n-1}) w dp_1$$

$$A \int_L [N][\hat{\varphi}_n][N][\hat{w}] dp_1 + \frac{A g_c \Delta t}{\gamma \lambda_1} \int_L [N][\hat{\varphi}_n][N][\hat{w}] dp_1 + \frac{A g_c \gamma \Delta t}{\lambda_1} \int_L [B][\hat{\varphi}_n][B][\hat{w}] dp_1 = A \int_L [N][\hat{\varphi}_{n-1}][N][\hat{w}] dp_1 + \frac{A \Delta t K}{\lambda_1} \int_L [[B][\hat{u}_n]]^2 (1 - [N][\hat{\varphi}_{n-1}])[N][\hat{w}] dp_1$$

$$A \int_L [N]^T [N] dp_1 [\hat{\varphi}_n] + \frac{A g_c \Delta t}{\gamma \lambda_1} \int_L [N]^T [N] dp_1 [\hat{\varphi}_n] + \frac{A g_c \gamma \Delta t}{\lambda_1} \int_L [B]^T [B] dp_1 [\hat{\varphi}_n] = A \int_L [N]^T [N] dp_1 [\hat{\varphi}_{n-1}] + \frac{A \Delta t K}{\lambda_1} \int_L [N]^T [[B][\hat{u}_n]]^2 (1 - [N][\hat{\varphi}_{n-1}]) dp_1$$

Parametrizando os termos, temos:

$$\frac{A}{L^e} \int_{-1}^{1} [N]^T [N] d\xi [\hat{\varphi}_n] + \frac{A g_c \Delta t}{L^e \gamma \lambda_1} \int_{-1}^{1} [N]^T [N] d\xi [\hat{\varphi}_n] + \frac{A g_c \gamma \Delta t}{\lambda_1 L^e} \int_{-1}^{1} [B]^T [B] d\xi [\hat{\varphi}_n] = \frac{A}{L^e} \int_{-1}^{1} [N]^T [N] d\xi [\hat{\varphi}_{n-1}] + \frac{A \Delta t K}{\lambda_1 L^e} \int_{-1}^{1} [N]^T [[B][\hat{u}_n]]^2 (1 - [N][\hat{\varphi}_{n-1}]) d\xi$$

Colocando o termo do dano $[\hat{\varphi}_n]$ em evidência e somando as contribuições, encontramos:

$$[\frac{A}{L^e}\int_{-1}^{1}[N]^T[N]d\xi + \frac{Ag_c\Delta t}{L^e\gamma\lambda_1}\int_{-1}^{1}[N]^T[N]d\xi + \frac{Ag_c\gamma\Delta t}{\lambda_1 L^e}\int_{-1}^{1}[B]^T[B]d\xi][\hat{\varphi}_n] =$$

$$\frac{A}{L^e}\int_{-1}^{1}[N]^T[N]d\xi[\hat{\varphi}_{n-1}] + \frac{A\Delta tK}{\lambda_1 L^e}\int_{-1}^{1}[N]^T[[B][\hat{u}_n]]^2(1-[N][\hat{\varphi}_{n-1}])d\xi$$

$$K_{e_2} = \frac{A}{L^e}\int_{-1}^{1}[N]^T[N]d\xi + \frac{Ag_c\Delta t}{L^e\gamma\lambda_1}\int_{-1}^{1}[N]^T[N]d\xi + \frac{Ag_c\gamma\Delta t}{\lambda_1 L^e}\int_{-1}^{1}[B]^T[B]d\xi$$

$$\hat{f}_2 = \frac{A}{L^e}\int_{-1}^{1}[N]^T[N]d\xi[\hat{\varphi}_{n-1}] + \frac{A\Delta tK}{\lambda_1 L^e}\int_{-1}^{1}[N]^T[[B][\hat{u}_n]]^2(1-[N][\hat{\varphi}_{n-1}])d\xi$$

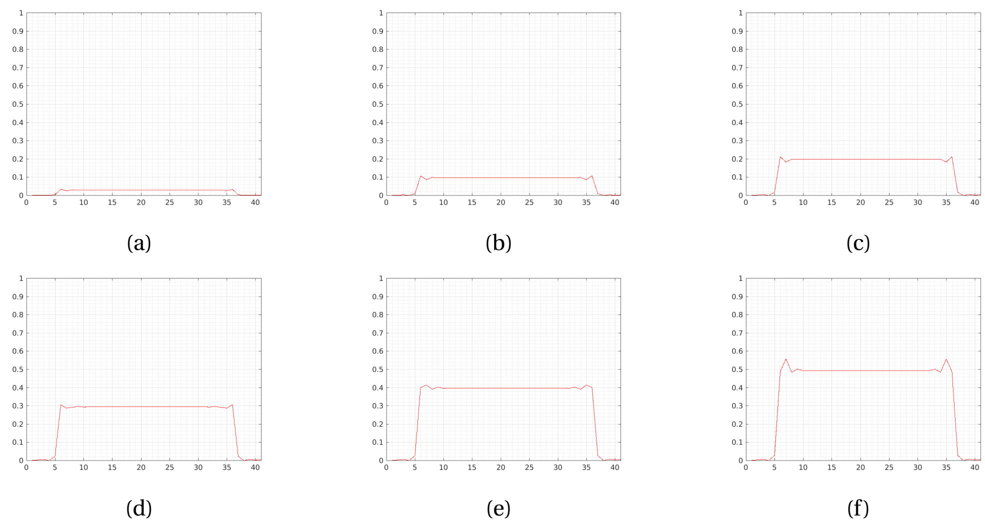
$$[K_{g_2}] = A_{i=1}^{nel}[K_{e_2}]$$

$$[K_{g_2}][\hat{\varphi}_n] = \hat{f}_2$$

# 3 Resultados

A figura (3.1) mostra a evolução do dano ao longo dos nós. Percebe-se que a mudança da geometria entre os elementos gera um fator concentrador de tensões que aumenta o dano de maneira abrupta.

Além disso, o dano é quase que constante ao longo da "alma"(parte central) da barra, com picos apenas na região de mudança de geometria. Na figura (3.2) o dano tem um progresso quase que linear, acelerando o rompimento do material nas últimas iterações.



(a) (b) (c)

(d) (e) (f)

Figura 3.1: Evolução da trinca

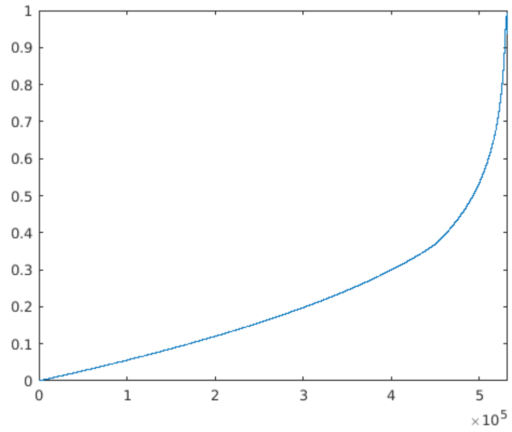



Figura 3.2: Progresso do dano ao longo das iterações

# 4 Conclusão

No caso estudado sem dano inicial e com seção de barra variável, o dano se concentra na região de mudança de área e se mantém quase que constante ao longo da alma da barra.

# Anexo

```
% @university: UNIVERSIDADE ESTADUAL DE CAMPINAS
% @school: FACULDADE DE ENGENHARIA MECANICA
% @module: MODELAGEM DE MATERIAIS TERMODINAMICAMENTE CONSISTENTES - IM437 L
% @activity: ATIVIDADE 1.1
%
% @author: DIPL.-ENG RENAN MIRANDA PORTELA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Cleaning

clear all
close all
clc

%% Coordinate matrix

nel = 20; % number of elements
coord = zeros(nel,2);
L = 1; %m
```

```
area = ones(nel,1);
base = 1:0.1*nel;
base_2 = 0.9*nel+1:nel;
delta_t = 1e-3;
gamma = L/(2*nel);
area(base) = 10;
area(base_2) = 10;
g_c = 2700;
lambda = 1e-100;
L_e = L/(2*nel);
n = 5; % gaussian points

for i = 1:2*nel+1
coord(i,1) = i; % number of nodes
coord(i,2) = (i-1)*L/(2*nel); % x-direction coordinate
end

%% incidence matrix

inci = zeros(nel,6);

for i = 1:nel
inci(i,1) = i; %number of elements
inci(i,2) = 1; %material
inci(i,3) = 1; %geometry
inci(i,4) = 2*i-1; %node 1
inci(i,5) = 2*i; %node 2
inci(i,6) = 2*i + 1; %node 3
end

%% material matrix
            %E     rho   nu
material = [200e9 7000 0.3;  %steel [Pa] [kg/m3] []
            70e9 2700 0.27]; %aluminium

%% geometry matrix

nnos = size(coord,1); % number of nodes

if mod(nnos,2) ~= 0

    k = (nnos+1)/2;

else
```

```
    k = nnos/2;

end

phi_dane = zeros(nnos,1);
% phi_dane(k,1) = 1e-2;

%% boundary conditions matrix
%   bc=[node | degrees of freedom | value]
%
%   DOF 1 --> x
%   DOF 2 --> y
%   DOF 3 --> z
%   DOF 4 --> ox
%   DOF 5 --> oy
%   DOF 6 --> oz

bc=[1 1 0;
    1 2 0];

%% load matrix

Load=[nnos 1 1000];

%% code

nbc = size(bc,1); % number of boundary conditions
nF = size(Load,1); % number of loads

%% while loop

kk = 0;

conv = 0;
iter = 0;

max_value = zeros(100,1);

[ x, w ] = legendre_set ( n );

figure('Name','Crack_Bar','NumberTitle','off');

while conv == 0
```

```
    iter = iter + 1;

    pause(1e-6)
    plot(phi_dane,'r')
%     xlim([0 nnos])
    grid on
    grid minor
     axis([0 nnos 0 1])

    max_value(iter) = max(phi_dane);

    %% Calculando kg

    alldof = 1:nnos; % degrees of freedom

    kg = zeros(nnos); % global stiffness matrix pre-location

    for i = 1 : nel

        node_1 = inci(i,4); % first node
        node_2 = inci(i,5); % second node
        node_3 = inci(i,6); % second node

        x_1 = coord(node_1,2);
        x_2 = coord(node_2,2);
        x_3 = coord(node_3,2);

        A = area(i);

        E = material(inci(i,2),1); % young's module

        loc = [node_1, node_2, node_3];

        ke = 0; % elemental stiffness matrix iniciation

        for j = 1 : n % gaussian quadrature integration

            e = x(j);
            peso = w(j);

            N = [-0.5*e*(1-e); (1-e^2); 0.5*e*(1+e)];
            B = [-0.5+e -2*e 0.5+e];
```

```
        phi = phi_dane(loc,1); % elemental dane
        det_J = B*[x_1; x_2; x_3]; % Jacobian 3 nodes element

        ke = ke + peso*E*A/L_e*((1-N'*phi)^2)*B'*B*det_J;

    end

    kg(loc,loc) = kg(loc,loc) + ke;

end

freedof = alldof;

for k = 1 : size(bc,1) % degrees of freedom
    freedof(bc(k,1)) = 0;
end

%% Calculando F

F = zeros(nnos,1);

for  i = 1 : size(Load,1)

    F(Load(i,1)) = Load(i,3);

end

u_n = zeros(size(F,1),1);

kg_aux = sparse(kg(logical(freedof),logical(freedof)));
% column & rows elimination
F_aux = sparse(F(logical(freedof),1)); % column & rows elimination

u_n(logical(freedof),1) = kg_aux\F_aux; % displacement vector

%% dano N-1

F_2 = zeros(nnos,1);

for i = 1 : nel

    node_1 = inci(i,4); % first node
    node_2 = inci(i,5); % second node
    node_3 = inci(i,6); % third node
```

```
    x_1 = coord(node_1,2); % location first node
    x_2 = coord(node_2,2); % location second node
    x_3 = coord(node_3,2);

    A = area(i);

    E = material(inci(i,2),1); % young's module

    loc = [node_1, node_2, node_3];

    fe = 0; % force vector iniciation

    for j = 1 : n % gaussian quadrature integration

        e = x(j);
        peso = w(j);

        N = [-0.5*e*(1-e); (1-e^2); 0.5*e*(1+e)];
        B = [-0.5+e -2*e 0.5+e];

        phi = phi_dane(loc,1); % elemental dane
        u_ele = u_n(loc,1); % elemental deformation

        det_J = B*[x_1; x_2; x_3]; % Jacobian 3 nodes element

        fe = fe + peso*(A/L_e*N*N'*phi*det_J +
        A * delta_t * E / (L_e*lambda) * N * (1 - N'*phi) *
        det_J * (B*u_ele)^2);

    end

    F_2(loc,1) = F_2(loc,1) + fe;

end

%% dano N

kg_2 = zeros(nnos);

for i = 1 : nel

    node_1 = inci(i,4); % first node
    node_2 = inci(i,5); % second node
```

```
    node_3 = inci(i,6); % second node

    x_1 = coord(node_1,2); % location first node
    x_2 = coord(node_2,2); % location second node
    x_3 = coord(node_3,2);

    A = area(i);

    E = material(inci(i,2),1); % young's module

    loc = [node_1, node_2, node_3];

    ke_2 = 0; % elemental stiffness matrix iniciation

    for j = 1 : n % gaussian quadrature integration

        e = x(j);
        peso = w(j);

        N = [-0.5*e*(1-e); (1-e^2); 0.5*e*(1+e)];
        B = [-0.5+e -2*e 0.5+e];

        det_J = B*[x_1; x_2; x_3]; % Jacobian 3 nodes element

        ke_2 = ke_2 + peso*(A/L_e*(N*N'*det_J) +
        g_c*A*delta_t/(gamma*lambda*L_e)*(N*N'*det_J) +
        g_c*A*delta_t*gamma/(lambda*L_e)*(B'*B*det_J));

    end

    kg_2(loc,loc) = kg_2(loc,loc) + ke_2;

end

phi_n = zeros(size(F_2,1),1);

kg_2_aux = sparse(kg_2(logical(freedof),logical(freedof)));
% column & rows elimination
F_2_aux = sparse(F_2(logical(freedof),1));
% column & rows elimination

phi_n(logical(freedof),1) = kg_2_aux\F_2_aux; % displacement vector

phi_dane = phi_dane + phi_n;
```

```
    if mod(iter, 3000) == 0
        kk = kk + 1;
        filename = sprintf('%d_%d', nel, kk);
        saveas(gcf,filename,'png')
    end

    if max(phi_dane) >= 1
        break
    end

end

figure('Name','Crack Progress','NumberTitle','off');
plot(max_value)
axis([0 iter 0 1])
```