

# PROGRAMA DE CÁLCULO ESTÁTICO DE TRELIÇAS PLANAS

```
% Programa script do MatLab
clear;
fprintf('%s\n', 'Cálculo Estático de Trelis Planas');

% Leitura de Dados
fprintf('%s\n', 'Leitura dos dados...');

% Abre o arquivo trel.txt
fid = fopen('trel.txt', 'rt');

% Lê a primeira linha do arquivo
dados = fscanf(fid, '%d', [4]);

% Obtenção do número de nós:
nnos=dados(1);

% Cálculo no número de graus de liberdade:
ngl=nnos*2;

% Obtenção do número de elementos:
nelm=dados(2);

% Obtenção do número de nós restringidos:
nnr=dados(3);

% Obtenção do número de nós carregados:
nnc=dados(4);

% Leitura das coordenadas:
coord = fscanf(fid, '%10e', [2,nnos]);

% Definição das coordenadas x dos nós: x(i)=abscissa do nó i
x=coord(1, :);

% Definição das coordenadas y dos nós: y(i)=ordenada do nó i
y=coord(2, :);

% Leitura das Propriedades dos elementos:
prop = fscanf(fid, '%10e', [4,nelm]);

% Definição do vetor nó1: no1(i)=1º nó que define o elemento i
no1=prop(1, :);

% Definição do vetor nó2: no2(i)=2º nó que define o elemento i
no2=prop(2, :);
% Definição do vetor área: area(i)=área do elemento i
area=prop(3, :);
```

```

% Definição do vetor módulo de elasticidade: mod(i)=mód. elast. do
elemento i
mod=prop(4,:);

% Leitura das restrições de apoio:
restr = fscanf(fid,'%10e',[3,nnr]);

% Definição do vetor restrições IC:
% se IC(i)=0 o GL i é livre; se IC(i)=1 o GL i está restringido
IC(nnos*2)=0;
for i=1:nnr,
    nn=restr(1,i);
    IC((nn-1)*2+1)=restr(2,i);
    IC((nn-1)*2+2)=restr(3,i);
end

% Leitura e definição do vetor de forças globais F:
forcas = fscanf(fid,'%10e',[3,nnc]);
F(ngl)=0.0;
for i=1:nnc,
    n=forcas(1,i);
    F((n-1)*2+1)=forcas(2,i);
    F((n-1)*2+2)=forcas(3,i);
end

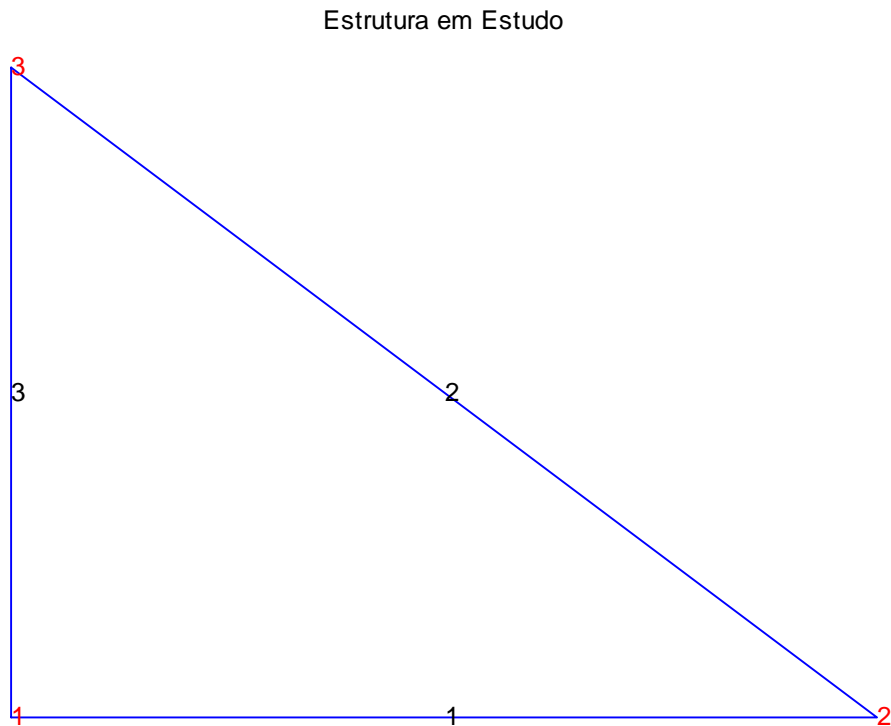
% fecha o arquivo de dados:
st = fclose(fid);

```

```

% Plota a estrutura (configuração inicial)
fprintf('%s\n', 'Desenhando a estrutura...');
figure(1);
clf;
for i=1:nelm,
    line([x(no1(i)) x(no2(i))],[y(no1(i)) y(no2(i))]);
end
axis equal;
axis off;
title('Estrutura em Estudo');
for i=1:nelm,
    text((x(no1(i))+x(no2(i)))/2,(y(no1(i))+y(no2(i)))/2,num2str(i));
end
for i=1:nnos,
    texto=text(x(i),y(i),num2str(i));
    set(texto,'Color','red')
end

```



**% Calcula a matriz de rigidez global pelo processo da rigidez direta**

fprintf('%s\n', 'Calculando a Matriz de Rigidez Global...');

Kg(ngl,ngl)=0.0;

Kg(:,:)=0.0;

**% Início do Loop**

for nel=1:nelm,

kk(1)=no1(nel);

kk(2)=no2(nel);

**% Cálculo da matriz de rotação**

dx=x(kk(2))-x(kk(1));

dy=y(kk(2))-y(kk(1));

**% Cálculo do comprimento do elemento**

L=sqrt(dx^2+dy^2);

**% Cálculo dos cossenos diretores**

cx=dx/L;

cy=dy/L;

**% Definição da Matriz de Rotação**

R=[ cx cy 0 0;

-cy cx 0 0;

0 0 cx cy;

0 0 -cy cx];

**% Definição da Matriz de Rigidez do elemento de treliça Ke**

E=mod(nel);

A=area(nel);

Ke=[E\*A/L 0 -E\*A/L 0;

0 0 0 0;

-E\*A/L 0 E\*A/L 0;

0 0 0 0];

**% Cálculo da Matriz rotacionada**

RKeR=R'\*Ke\*R;

**% Acumula na matriz de rigidez global Kg**

**% k,l= GL locais rotacionados**

**% i= nó à q pertence o GL local k**

**% j= nó à q pertence o GL local l**

**% m= nº de GLs existentes antes do nó i**

**% n= nº de GLs existentes antes do nó j**

**% k1,l1= contadores de 1 a 2**

for i=1:2,

m=2\*(kk(i)-1);

for j=1:2,

n=2\*(kk(j)-1);

k1=0;

for k=2\*i-1:2\*i,

k1=k1+1;

l1=0;

for l=2\*j-1:2\*j,

l1=l1+1;

Kg(m+k1,n+l1)=Kg(m+k1,n+l1)+RKeR(k,l);

end

end

end

end

end

**% Fim do loop**

L; α;  
E; A

Calcula  $R_\alpha$  e  $k_e$

$$k_e^G = R^T \cdot k_e \cdot R$$

k, l ≡ coords (GL) locais

p, q ≡ coords (GL) globais

p, q = k, l no referencial global

k, l = 1 até 4

p, q = (nó<sub>inicial</sub> - 1)·2 + 1 até (nó<sub>inicial</sub> - 1)·2 + 2, e  
(nó<sub>final</sub> - 1)·2 + 1 até (nó<sub>final</sub> - 1)·2 + 2

elmt	no inic	no final	i	j	m	n	k	l	k1	l1	m+k1	n+l1			
1	1	2	1	1	0	0	1	1	1	1	1	1			
			1	1	0	0	1	2	1	2	1	2	1		
			1	1	0	0	2	1	2	2	2	2	2	1	
			1	1	0	0	2	2	2	2	2	2	2	2	
			1	2	0	2	1	3	1	1	1	1	1	3	
			1	2	0	2	1	4	1	2	1	2	1	4	
			1	2	0	2	2	3	2	1	2	3	2	3	
			1	2	0	2	2	4	2	2	2	2	2	4	
			2	1	2	0	3	1	1	1	1	3	1	3	1
			2	1	2	0	3	2	1	2	2	3	2	3	2
			2	1	2	0	4	1	2	1	2	4	1	4	1
			2	1	2	0	4	2	2	2	2	4	2	4	2
			2	2	2	2	3	3	1	1	1	3	3	3	3
			2	2	2	2	3	4	1	2	2	3	4	3	4
2	2	2	2	4	3	2	1	4	4	3	4	3			
2	2	2	2	4	4	2	2	2	4	4	4	4			
2	2	3	1	1	2	2	1	1	1	1	3	3			
			1	1	2	2	1	2	1	2	3	4			
			1	1	2	2	2	1	2	1	4	3			
			1	1	2	2	2	2	2	2	4	4			
			1	2	2	4	1	3	1	1	3	5			
			1	2	2	4	1	4	1	2	3	6			
			1	2	2	4	2	3	2	1	4	5			
			1	2	2	4	2	4	2	2	4	6			
			2	1	4	2	3	1	1	1	5	3			
			2	1	4	2	3	2	1	2	5	4			
			2	1	4	2	4	1	2	1	6	3			
			2	1	4	2	4	2	2	2	6	4			
			2	2	4	4	3	3	1	1	5	5			
			2	2	4	4	3	4	1	2	5	6			
2	2	4	4	4	3	2	1	6	5						
2	2	4	4	4	4	2	2	6	6						
3	1	3	1	1	0	0	1	1	1	1	1	1			
			1	1	0	0	1	2	1	2	1	2	1		
			1	1	0	0	2	1	2	2	2	2	1		
			1	1	0	0	2	2	2	2	2	2	2		
			1	2	0	4	1	3	1	1	1	5			
			1	2	0	4	1	4	1	2	1	6			
			1	2	0	4	2	3	2	1	2	5			
			1	2	0	4	2	4	2	2	2	6			
			2	1	4	0	3	1	1	1	5	1			
			2	1	4	0	3	2	1	2	5	2			
			2	1	4	0	4	1	2	1	6	1			
			2	1	4	0	4	2	2	2	6	2			
			2	2	4	4	3	3	1	1	5	5			
			2	2	4	4	3	4	1	2	5	6			
2	2	4	4	4	3	2	1	6	5						
2	2	4	4	4	4	2	2	6	6						

$$[K]_{pq} = [K]_{pq} + [k_e^G]_{kl}$$

```
% Resolução da Equação de Equilíbrio:
```

```
U(ngl)=0.0;
```

```
U=inv(Kg)*F';
```

```
% Atualizar Geometria
```

```
x1(nnos)=0.0;
```

```
y1(nnos)=0.0;
```

```
for i=1:nnos,
```

```
    x1(i)=x(i)+U(2*i-1);
```

```
    y1(i)=y(i)+U(2*i);
```

```
end
```

```
% Plota a estrutura deformada
```

```
xd(nnos)=0.0;
```

```
yd(nnos)=0.0;
```

```
escala=input('Digite a escala a ser utilizada(100):');
```

```
if (isempty(escala)==1)
```

```
    escala=100;
```

```
end
```

```
for i=1:nnos,
```

```
    xd(i)=x(i)+escala*U(2*i-1);
```

```
    yd(i)=y(i)+escala*U(2*i);
```

```
end
```

```
fprintf('%s\n','Desenhando a estrutura deformada...');
```

```
figure(2);
```

```
clf;
```

```
for i=1:nelm,
```

```
    linha=line([x(no1(i)) x(no2(i))],[y(no1(i)) y(no2(i))]);
```

```
    set(linha,'Color','blue')
```

```
    set(linha,'LineStyle','--')
```

```
    linha=line([xd(no1(i)) xd(no2(i))],[yd(no1(i)) yd(no2(i))]);
```

```
    set(linha,'Color','red')
```

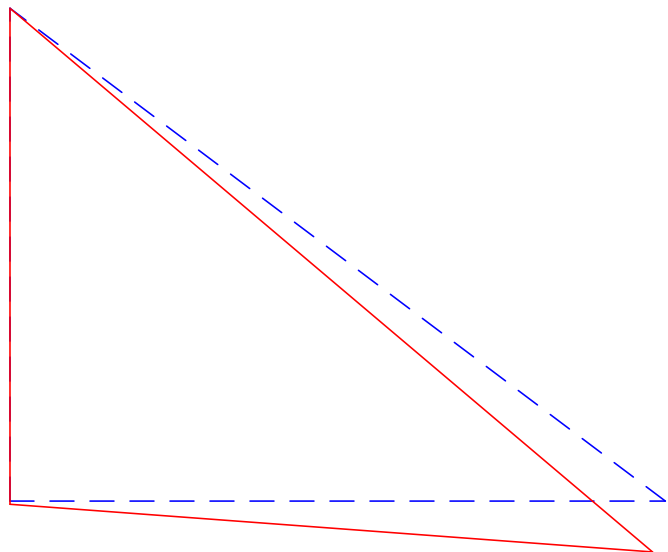
```
end
```

```
axis equal;
```

```
axis off;
```

```
title('Estrutura Deformada');
```

Estrutura Deformada



### % Obtenção dos esforços

```
% Def. vetor esforços segundo as coordenadas globais, por elemento  
Sg(4)=0.0;
```

```
% Def. vetor esforços por elementos, segundo suas coord. locais  
Sl(nelm,4)=0.0;
```

```
% Def. vetor reações  
reac(ngl)=0.0;
```

### % Início do Loop

```
for nel=1:nelm,  
    kk(1)=no1(nel);  
    kk(2)=no2(nel);
```

### % Cálculo do comprimento do elemento

```
dx=x(kk(2))-x(kk(1));  
dy=y(kk(2))-y(kk(1));  
L=sqrt(dx^2+dy^2);
```

### % Cálculo dos cossenos diretores

```
cx=dx/L;  
cy=dy/L;
```

### % Definição da Matriz de Rotação

```
R=[ cx  cy  0  0;  
    -cy  cx  0  0;  
     0   0  cx  cy;  
     0   0 -cy  cx];
```

### % Definição da Matriz de Rigidez do elemento de treliça Ke

```
E=mod(nel);  
A=area(nel);  
Kl=[E*A/L  0 -E*A/L  0;  
     0      0  0      0;  
    -E*A/L  0 E*A/L  0;  
     0      0  0      0];
```

### % Cálculo do vetor dos deloc locais segundo as coord globais ug

```
ug(4)=0.0;  
for m=1:2,  
    k=2*(kk(m)-1);  
    l=0;  
    for n=2*m-1:2*m,  
        l=l+1;  
        ug(n)=U(k+l);  
    end  
end
```

```
% Cálculo do vetor dos deloc locais segundo as coord locais ul  
ul=R*ug';
```

```

% Cálculo dos esforços segundo as coordenadas locais ul
Sl(nel,:)=(Kl*ul)';

% Cálculo dos esforços segundo as coordenadas globais
Sg(:)=0.0;
Sg=(R'*Sl(nel,:))';

% Cálculo das reações de apoio
for i=1:2,
    for j=1:2,
        if (IC(2*(kk(i)-1)+j)~=0)
            % Cálculo da coord global m
            m=2*(kk(i)-1)+j;
            n=2*(i-1)+j;
            % Cálculo das reações de apoio
            reac(m)=reac(m)+Sg(n)';
        end
    end
end
end
% Fim do Loop

```

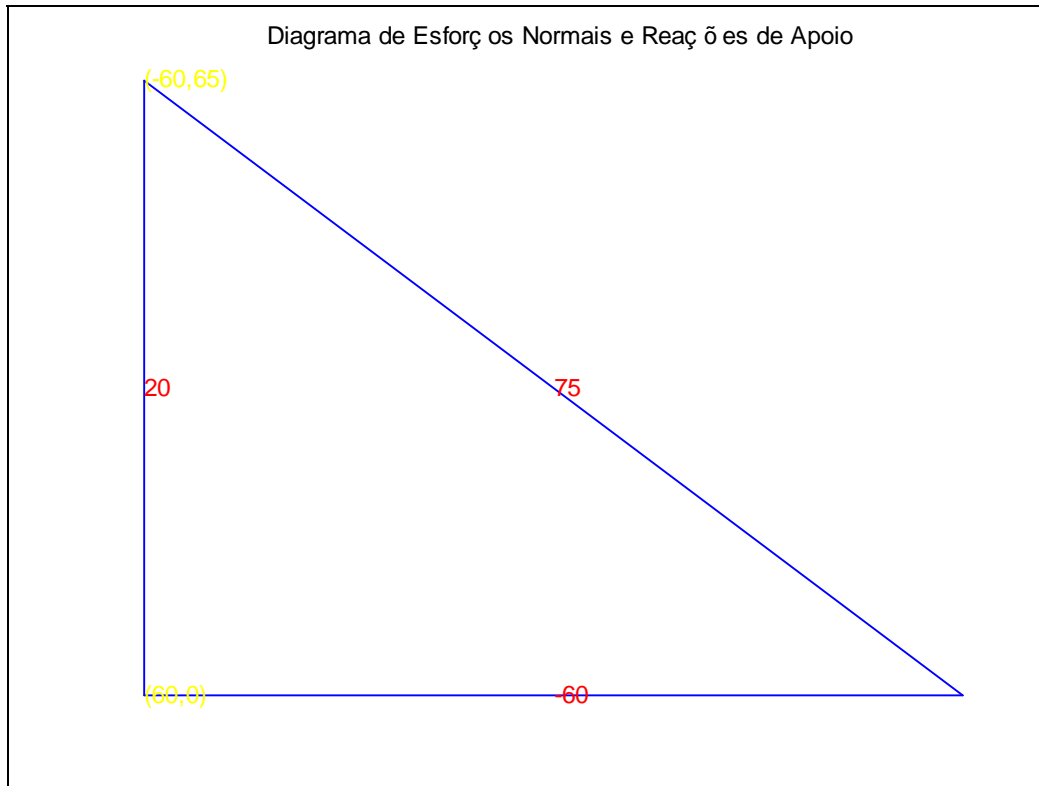
```

% Plota o Diagrama de esforços normais e reações de apoio
fprintf('%s\n', 'Desenhando o DEN...');
figure(3);
clf;

% Desenha os esforços normais
for i=1:nelm,
    linha=line([x(no1(i)) x(no2(i))],[y(no1(i)) y(no2(i))]);
    set(linha,'Color','blue')
    texto=text((x(no1(i))+x(no2(i)))/2,(y(no1(i))+y(no2(i)))/2,num2str(-
Sl(i,1),4));
    set(texto,'Color','red')
end
for i=1:nnr,
    nn=restr(1,i);
    valor=strcat('(',num2str(reac((nn-1)*2+1),3),',',',',
        num2str(reac((nn-1)*2+2),3),',')');
    texto=text(x(nn),y(nn),valor);
    set(texto,'Color','y')
end

% Desenha as reações de apoio
axis equal;
axis off;
title('Diagrama de Esforços Normais e Reações de Apoio');

```



```

% Fim do programa

```