

# **Metodo de Elementos Finitos para analisis estructural**

Instructor: Carlos X. Azua Gonzalez, PhD

# Contenidos del curso

**6 sesiones FEM estatico + 1 sesion FEM dinamico:**

**8 horas (3s)**(8-9 Feb, 18:00-22:00) + **4 horas (1s + prueba)**(16 Feb,18:00-22:00)

**8 horas (3 s)** (14-15 Feb, 18:00-22:00)

**Total horas (contacto):** 8h x2 + 4h x1 = 20 horas

**1 prueba final:**

**Calculos en mathcad y parte teoria** (16 Feb,19:00-22:00)

**1 prueba inicial de conocimiento:**

**Estatica basica + Mecanica de Materiales + Algebra lineal**

# Contenido de sesiones

## **#1 Introduccion:**

- Reseña de metodos modernos de analisis de estructuras.
- Introduccion al uso de mathcad.

## **#2 Ecuaciones matriciales de equilibrio I:**

- Derivacion de ecuaciones de equilibrio en el Continuo y forma debil.
- Introduccion a interpolacion en elementos finitos.

## **#3 Ecuaciones matriciales de equilibrio II:**

- La matriz de rigidez en elementos de armadura (2D).
- EL Jacobiano y la cuadratura de Gauss.
- Evaluacion numerica de la matriz de rigidez.

# Contenido de sesiones

## **#4 Ensamblaje de elementos finitos:**

- Topologia de elementos, conectividad, y esparsidad global.
- Aplicacion al ensamblaje matricial en armaduras.

## **#5 Condiciones de borde I:**

- Modificar matriz de rigidez y vector de carga para condiciones de borde mixtas.
- Computo de deformaciones para condiciones de borde mixtas.

## **#6 Condiciones de borde II:**

- Computo de reacciones y verificacion.
- Pos-procesamiento basico

# Contenido de sesiones

## **#7 Dinamica estructural:**

- **Introduccion al analisis de problemas de vibracion libre.**
- **Frecuencias propias en sistemas estructurales con masas discretas.**

# Prueba de conocimientos

Prueba de 20 min - en linea

# Sesion 1

## Inicios de los metodos de elementos finitos:

Hrennikof, A. (1941). Solution of Problems of Elasticity by the Framework Method, J. Appl. Mech. Dec 1941, 8(4): A169-A175 (7 pages)  
<https://doi.org/10.1115/1.4009129>

**Condicion Estatica:**

$$\left. \begin{aligned} \frac{\partial \sigma_x}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + X &= 0 \\ \frac{\partial \sigma_y}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + Y &= 0 \end{aligned} \right\}$$

**Balance Momento:**

**Compatibilidad:**

$$\frac{\partial^2 \epsilon_x}{\partial y^2} + \frac{\partial^2 \epsilon_y}{\partial x^2} = \frac{\partial^2 \gamma_{xy}}{\partial x \partial y}$$

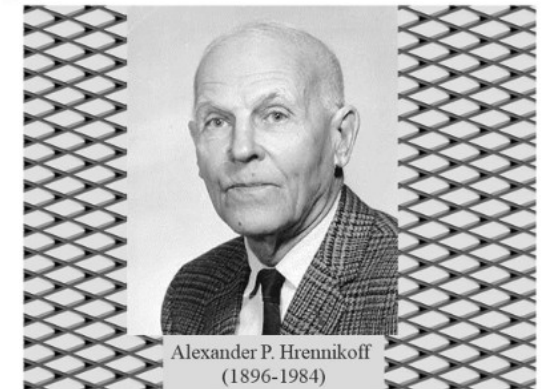
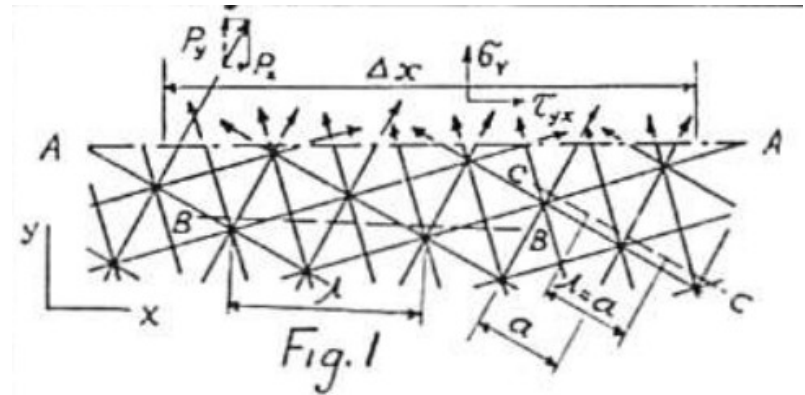
**Relacion cinematica:**

**Material elastico:**

$$\epsilon_x = \frac{1}{E} (\sigma_x - \mu \sigma_y)$$

**Relacion constitutiva:**

**+Condiciones de borde/+Condiciones borde e iniciales**



**Negro = Metodo de subdivisiones**

**Azul = Elementos finitos modernos**

Liu, WK, Li, S., Park, H. (2022). Eighty Years of the Finite Element Method: Birth, Evolution, and Future. Archives of Computational Methods in Engineering, 29:4431-4453  
<https://doi.org/10.1007/s11831-022-09740-9>

# Sesion 1

## Inicios de los metodos de elementos finitos:

### Diferencias Finitas

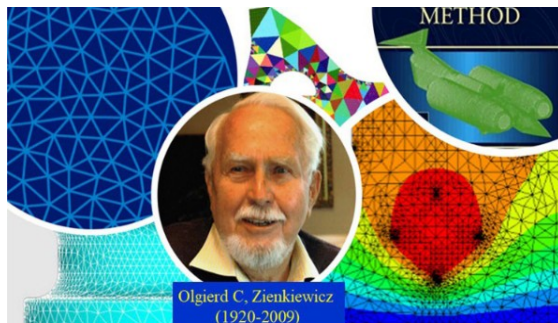
Resuelve PDE fisica

Variables en celda

Geometrias simples

Calculo directo de gradientes

Errores dependen de la malla



### Elementos Finitos

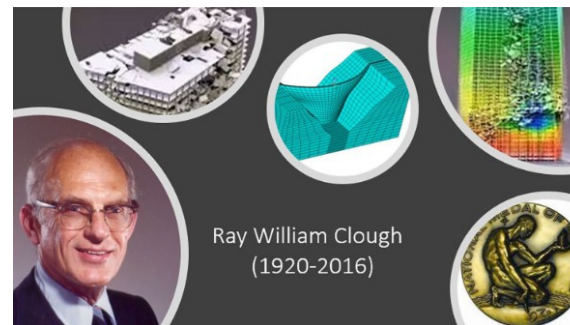
Resuelve PDE fisica

Variables unicas en nodos entre elementos

Metodo se adapta a geometrias complejas

Gradientes dependen del grado interpolacion

Errores dependen de la malla e interpolacion



### UK → Fundador UKACM

Liu, WK, Li, S., Park, H. (2022). Eighty Years of the Finite Element Method: Birth, Evolution, and Future. Archives of Computational Methods in Engineering, 29:4431-4453 <https://doi.org/10.1007/s11831-022-09740-9>

### US → Metodo Matriz Rigidez (54') / Elementos Finitos (60')

Clough RW (1960) The finite element method in plane stress analysis. In: Proceedings of 2nd ASCE conference on electronic computation. Pittsburgh PA, 8-9 Sept



# Sesion 1

## Demostracion practica de Mathcad:

$$A := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$B := \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

$$C := \begin{bmatrix} 1 & -3 \\ 3 & 4 \end{bmatrix}$$

```
MSYMCK(n,m,Ma) :=  $\begin{array}{l} \text{check} \leftarrow 1 \\ \text{for } j \in 1..m \\ \quad \text{if } \text{check} = 0 \\ \quad \quad \text{break} \\ \quad \text{for } i \in 1..n \\ \quad \quad \text{if } M_{a_i,j} \neq M_{a_j,i} \\ \quad \quad \quad \text{check} \leftarrow 0 \\ \quad \quad \quad \text{break} \end{array}$  \\ \text{check}
```

# Sesion 1

## Demostracion practica de Mathcad:

$$a_d := \det(A)$$

$$a_d = 1$$

$$d_d := \det(\text{diag}([1 \ 0 \ 1]^T))$$

$$d_d = 0$$

$$D_i := A^{-1}$$

$$D_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$E_i := \text{diag}([1 \ 0 \ 1]^T)$$

$$E_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$F_i := I_{3 \times 3}^{-1}$$

$$F_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\sigma_e := \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\text{eigenvals}(\sigma_e) = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$\text{eigenvecs}(\sigma_e) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

# Sesion 1

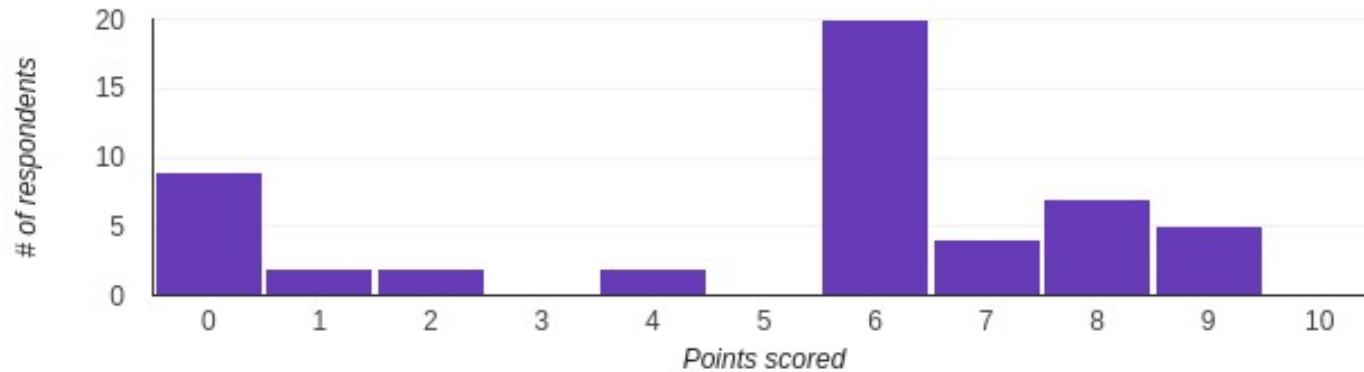
## Resultados de la prueba:

Average  
5.16 / 10 points

Median  
6 / 10 points

Range  
0 - 9 points

Total points distribution



# Sesion 1 - recapituacion

$$[0 \ 0 \ 1]^T \times r = |r| [-\sin(\theta) \ \cos(\theta) \ 0]^T; r = |r| [\cos(\theta) \ \sin(\theta) \ 0]^T$$

$$|r| = \sqrt{(r \cdot r)}; \quad \sin(\theta)/\cos(\theta) = r_y/r_x$$

=>

$$x_u' = |1| [\cos(\theta) \ \sin(\theta) \ 0]^T$$

$$y_u' = [0 \ 0 \ 1]^T \times x_u' = |1| [-\sin(\theta) \ \cos(\theta) \ 0]^T; \quad z_u' = z_u = [0 \ 0 \ 1]^T$$

$$u'_{el} = T_u u_{el}$$

$$[u_x \ u_y \ 0]^T \rightarrow [u_{x'} \ u_{y'} \ 0]^T$$

$$[u_x \ u_y \ 0]^T = (x_u' \cdot u) i' + (y_u' \cdot u) j' + (z_u' \cdot u) k' = (x_u'^T \cdot u) i' + (y_u'^T \cdot u) j' + (z_u'^T \cdot u) k'$$

$$[u_x \ u_y \ 0]^T = (\cos(\theta)u_x + \sin(\theta)u_y + 0 \cdot 0) i' + (-\sin(\theta)u_x + \cos(\theta)u_y + 0 \cdot 0) j' + (0 \cdot u_x + 0 \cdot u_y + 1 \cdot 0) k'$$

$$\begin{bmatrix} u_{x'} \\ u_{y'} \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta)u_x + \sin(\theta)u_y + 0 \cdot 0 \\ -\sin(\theta)u_x + \cos(\theta)u_y + 0 \cdot 0 \\ (0)u_x + (0)u_y + 1 \cdot 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_x \\ u_y \\ 0 \end{bmatrix}$$

# Sesion 1 - recapituacion

$$\mathbf{u}'_{el} = \mathbf{T}_u \mathbf{u}_{el}$$

$$\begin{bmatrix} \mathbf{u}_{x'} \\ \mathbf{u}_{y'} \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta)\mathbf{u}_x + \sin(\theta)\mathbf{u}_y + 0 \cdot 0 \\ -\sin(\theta)\mathbf{u}_x + \cos(\theta)\mathbf{u}_y + 0 \cdot 0 \\ (0)\mathbf{u}_x + (0)\mathbf{u}_y + 1 \cdot 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} [\mathbf{u}]'_1 \\ [\mathbf{u}]'_2 \end{bmatrix} = \begin{bmatrix} [\mathbf{T}]_b & [0] \\ [0] & [\mathbf{T}]_b \end{bmatrix} \cdot \begin{bmatrix} [\mathbf{u}]_1 \\ [\mathbf{u}]_2 \end{bmatrix}$$

$$\mathbf{T}_u = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & \cos(\theta) & \sin(\theta) \\ 0 & 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$[\mathbf{u}]'_1 = [\mathbf{u}_{1x'}, \mathbf{u}_{1y'}]^T ; [\mathbf{u}]'_2 = [\mathbf{u}_{2x'}, \mathbf{u}_{2y'}]^T$$

$$\mathbf{T}_b = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

# Sesion 2

## Balance de momento en solidos continuos:

$$\text{Div}[\sigma] + b = \rho \ddot{u}$$

- Segunda ley de Newton aplicada en un volumen diferencial
- Conservacion de momento en unidades : Fuerza / volumen
- Tensor de esfuerzo es simetrico (momento angular)

$$s_{xy} = s_{yx} \quad s_{yz} = s_{zy} \quad s_{xz} = s_{zx}$$

# Sesion 2

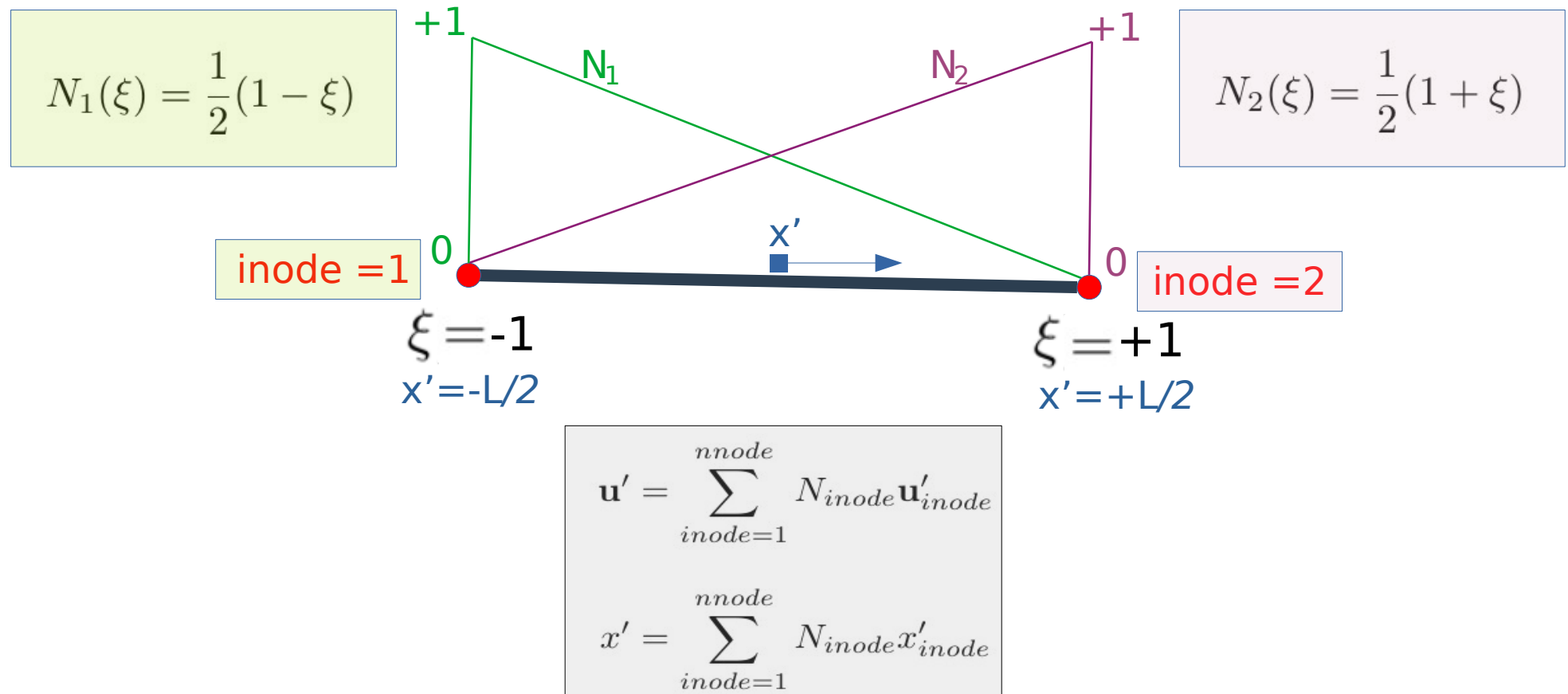
## Forma debil de balance de momento:

$$\underbrace{\int_{\Omega} \delta \mathbf{u}^T \cdot (\rho \ddot{\mathbf{u}}) d\Omega}_1 + \underbrace{\int_{\Omega} \delta \boldsymbol{\varepsilon} : \boldsymbol{\sigma} d\Omega}_2 = \underbrace{\int_{\Omega} \delta \mathbf{u}^T \cdot \mathbf{b} d\Omega}_3 + \underbrace{\int_{\Gamma_{\sigma}} \delta \mathbf{u}^T \cdot \mathbf{t} d\Gamma}_4$$

- **1: Energia debido a efectos de inercia (analisis dinamico)**
- **2: Energia interna debido a deformacion del solido**
- **3: Energia debido a la atraccion de masas (e.j. gravedad)**
- **4: Energia debido a tracciones en las superficies del solido**

# Sesion 2

## Interpolacion en FEM





# Sesion 2

## Funciones de interpolacion (mathcad)

$$N_{vec}(\xi) := \left[ \frac{1}{2} \cdot (1 - \xi) \quad \frac{1}{2} \cdot (1 + \xi) \right]^T$$

## Matriz de interpolacion (mathcad)

$$N(\xi) := \left\| \begin{array}{c} [N_1 \ N_2] \leftarrow N_{vec}(\xi)^T \\ [N_1 \ 0 \ N_2 \ 0] \\ [0 \ N_1 \ 0 \ N_2] \end{array} \right\|$$

# Sesion 2

## Ejercicio - Funciones de interpolacion

Determine el vector de desplazamiento local en  $\xi = -1$ ,  $\xi = 0$ ,  $\xi = +1$

Para los siguientes desplazamientos nodales locales

$$\mathbf{u}'_{el} = [u'_{1x} \ u'_{1y} \ u'_{2x} \ u'_{2y}]^T = [1.0 \ 0.0 \ 1.0 \ 1.0]^T$$

## Ejercicio - Matriz de interpolacion

Determine la matriz de interpolacion  $\mathbf{N}$  en  $\xi = 0$

(Nota: recuerda que la matriz de interpolacion convierte desplazamientos nodales locales directamente a vector de desplazamiento)

$$\mathbf{u}'_{el} = \mathbf{T}_u \mathbf{u}_{el}$$

$$\mathbf{u}' = \mathbf{N} \mathbf{T}_u \mathbf{u}_{el}$$

# Sesion 2

## Derivadas locales de funciones de interpolacion

$$\frac{\partial N_1(\xi)}{\partial \xi} = -\frac{1}{2}$$

$$\frac{\partial N_2(\xi)}{\partial \xi} = \frac{1}{2}$$

**En mathcad:**

$$N_{\xi}(\xi) := \begin{bmatrix} -1 & 1 \\ 2 & 2 \end{bmatrix}^T$$

# Sesion 3

## Ecuacion de movimiento (a nivel de elemento)

$$\mathbf{M}_{el}\ddot{\mathbf{u}}_{el} + \mathbf{C}_{el}\dot{\mathbf{u}}_{el} + \mathbf{K}_{el}\mathbf{u}_{el} = \mathbf{F}_{ext}^{el}$$

Matriz de amortiguacion  $\mathbf{C}$  es usualmente un artificio numerico (no fisico)

**Matriz de masa:**  $\mathbf{M}_{el} = \mathbf{T}_u^T \int_{\Omega} \rho \mathbf{N}^T \mathbf{N} d\Omega \mathbf{T}_u$

**Matriz de rigidez:**  $\mathbf{K}_{el} = \mathbf{T}_u^T \int_{\Omega} \mathbf{B}^T \mathbf{E} \mathbf{B} d\Omega \mathbf{T}_u$

**Vector de fuerzas:**  $\mathbf{F}_{ext}^{el} = \mathbf{T}_u^T \int_{\Omega} \mathbf{N}^T \mathbf{T}_b \mathbf{b} d\Omega + \mathbf{T}_u^T \int_{\Gamma_{\sigma}} \mathbf{N}^T \mathbf{t}' d\Gamma + \mathbf{T}_u^T \mathbf{F}'_p$

# Sesion 3

## Equilibrio lineal cuasi-estatico (a nivel de elemento)

**PTV:**

$$\mathbf{F}_{int}^{el} = \mathbf{F}_{ext}^{el}$$

$$\mathbf{F}_{int}^{el} = \underbrace{\mathbf{T}_u^T \int_{\Omega} \mathbf{B}^T \sigma_{x'x'} d\Omega}_{\mathbf{K}_{el} \mathbf{u}_{el}}$$

**Matriz de deformacion-desplazamiento:**  $\mathbf{B} = \begin{bmatrix} \frac{\partial N_1}{\partial x'} & 0 & \frac{\partial N_2}{\partial x'} & 0 \end{bmatrix}$

**Nota:** deformaciones infinitesimales? (=>no-lineal)

**Relacion constitutiva (esfuerzo-deformacion):**  $\sigma_{x'x'} = E \mathbf{B} \mathbf{T}_u \mathbf{u}_{el}$

**Nota:** Esfuerzos inadmisibles? (=>no-lineal)

$$\varepsilon_{x'x'} = \mathbf{B} \mathbf{T}_u \mathbf{u}_{el}$$

## Sesion 3

### Derivadas cartesianas (regla de la cadena):

$$\left\{ \frac{\partial N_1(\xi)}{\partial x'} \quad \frac{\partial N_2(\xi)}{\partial x'} \right\} = \mathbf{J}^{-1} \cdot \left\{ \frac{\partial N_1(\xi)}{\partial \xi} \quad \frac{\partial N_2(\xi)}{\partial \xi} \right\}$$

### Jacobiano (gradiente del vector posicion):

$$\mathbf{J} = \frac{\partial x'}{\partial \xi} = \left\{ \frac{\partial N_1(\xi)}{\partial \xi} \quad \frac{\partial N_2(\xi)}{\partial \xi} \right\} \cdot \begin{Bmatrix} x'_1 \\ x'_2 \end{Bmatrix}$$

$$\mathbf{X}'_1 = -L/2 \quad \& \quad \mathbf{X}'_2 = +L/2 \quad ; \quad L = \text{Longitud del elemento}$$

# Sesion 3

## Transformacion de grados de libertad:

DOF globales  $\rightarrow$  DOF local

$$\mathbf{u}'_{el} = \mathbf{T}_u \mathbf{u}_{el}$$

## Matriz de transformacion

$$\mathbf{T}_u = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & \cos(\theta) & \sin(\theta) \\ 0 & 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

# Sesion 3

## Integracion Gausiana en la matriz de rigidez:

Volumen = f(Area de la seccion)  $\rightarrow dV = A dx'$

$$\int_{\Gamma} f(x') dx' = \int_{-1}^1 f(x'(\xi)) \det(\mathbf{J}) d\xi = \sum_{i_p=1}^{n_p} f(x'(\xi_{i_p})) \cdot \det(\mathbf{J}) \cdot w_{i_p}$$

Integracion exacta si funcion es aproximable con un polinomio de orden  $(2n_p-1)$

Point-rule	Positions	Weighting
1	0.0	2.0
2	$[-1/\sqrt{3}, 1/\sqrt{3}]$	$[1.0, 1.0]$
3	$[-\sqrt{3/5}, 0.0, \sqrt{3/5}]$	$[5/9, 8/9, 5/9]$



# Sesion 3

## Integracion Gaussiana (mathcad):

Coefficientes y posicion local

$$\xi := \begin{bmatrix} \frac{-1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix}^T$$

$$w := \begin{bmatrix} 1 & 1 \end{bmatrix}^T$$

## Longitud de un elemento lineal (mathcad):

$$LBARPN(x_a, x_b) := \left\| \begin{array}{l} dx \leftarrow x_b - x_a \\ Lsqr \leftarrow dx^T \cdot dx \\ L \leftarrow \sqrt{Lsqr} \end{array} \right\|$$

# Sesion 3

## Jacobiano (mathcad):

$$J(\xi, x_{coord}) := \left| \begin{array}{l} x_a \leftarrow \begin{bmatrix} x_{coord_1} & x_{coord_2} \end{bmatrix}^T \\ x_b \leftarrow \begin{bmatrix} x_{coord_3} & x_{coord_4} \end{bmatrix}^T \\ L \leftarrow LBARN(x_a, x_b) \\ \chi \leftarrow \begin{bmatrix} -L & L \\ 2 & 2 \end{bmatrix}^T \\ N_{der} \leftarrow N_{\xi}(\xi) \\ N_{der}^T \cdot \chi \end{array} \right|$$

# Sesion 3

## Derivadas Cartesianas de funciones de interpolacion (mathcad):

$$N_{dxdy}(\xi, x_{coord}) := [J(\xi, x_{coord})]^{-1} \cdot N_{\xi}(\xi)^T$$

## Matriz de deformacion-desplazamiento (mathcad):

$$B_{mat}(\xi, x_{coord}) := \left\| \begin{array}{l} N_{cart} \leftarrow N_{dxdy}(\xi, x_{coord})^T \\ \left[ \begin{array}{cccc} N_{cart_1} & 0 & N_{cart_2} & 0 \end{array} \right] \end{array} \right\|$$

# Sesion 3

## Matriz de rigidez local (mathcad):

$$K_{L2D}(E, A, x_{coord}, n_p, ndof, nnode) := \left\| \begin{array}{l} K \leftarrow 0 \cdot \text{identity}(ndof \cdot nnode) \\ \text{if } n_p = 2 \\ \quad \text{for } i_p \in 1 \dots n_p \\ \quad \quad \left\| \begin{array}{l} detJ_{ip} \leftarrow detJ(\xi_{i_p}, x_{coord}) \\ B \leftarrow B_{mat}(\xi_{i_p}, x_{coord}) \\ dV \leftarrow A \cdot detJ_{ip} \cdot w_{i_p} \\ dK \leftarrow B^T \cdot E \cdot B \cdot dV \\ K \leftarrow K + dK \end{array} \right\| \\ K \end{array} \right\|$$

# Sesion 3

## Matriz de transformacion (mathcad):

$$T_{2D}(x_{coord}) := \left| \begin{array}{l} x_a \leftarrow [x_{coord_1} \ x_{coord_2}]^T \\ x_b \leftarrow [x_{coord_3} \ x_{coord_4}]^T \\ [dx \ dy] \leftarrow (x_b - x_a)^T \\ L \leftarrow LBARNP(x_a, x_b) \\ c \leftarrow \frac{dx}{L} \\ s \leftarrow \frac{dy}{L} \\ \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix} \end{array} \right|$$

# Sesion 3

## Matriz de rigidez (mathcad):

$$K_{2D}(E, A, x_{coord}, n_p, ndof, nnode) := \left\| \begin{array}{l} K_L \leftarrow K_{L2D}(E, A, x_{coord}, n_p, ndof, nnode) \\ T_u \leftarrow T_{2D}(x_{coord}) \\ K_{el} \leftarrow T_u^T \cdot K_L \cdot T_u \end{array} \right\|$$

# Sesion 4

## Ensamblaje de matriz y vector global:

$$\underbrace{\left( \bigcup_{ielem=1}^{nelem} \mathbf{K}_{el} \right)}_{\mathbf{K}_g} \mathbf{u}_g = \underbrace{\bigcup_{ielem=1}^{nelem} \mathbf{F}_{ext}^{el}}_{\mathbf{F}_{ext}}$$

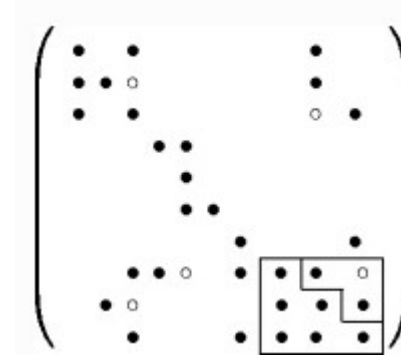
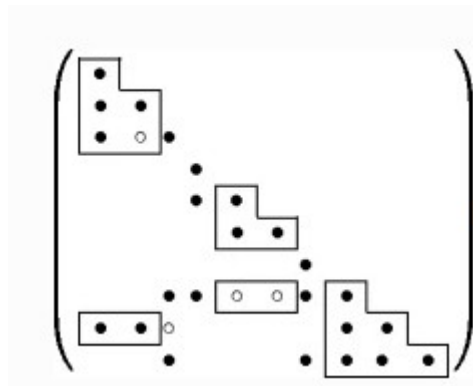
La topologia del elemento nos indica la correspondencia:

#nodo local → #nodo global

$\mathbf{Lnodes} = [\text{inodg}_{ielem} \quad \text{jnodeg}_{ielem}]$  ([nodo global #1, nodo global #2] en elemento  $ielem$ )

# Sesion 4

## Tipos de matrices globales:



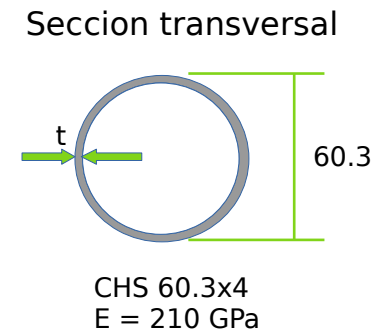
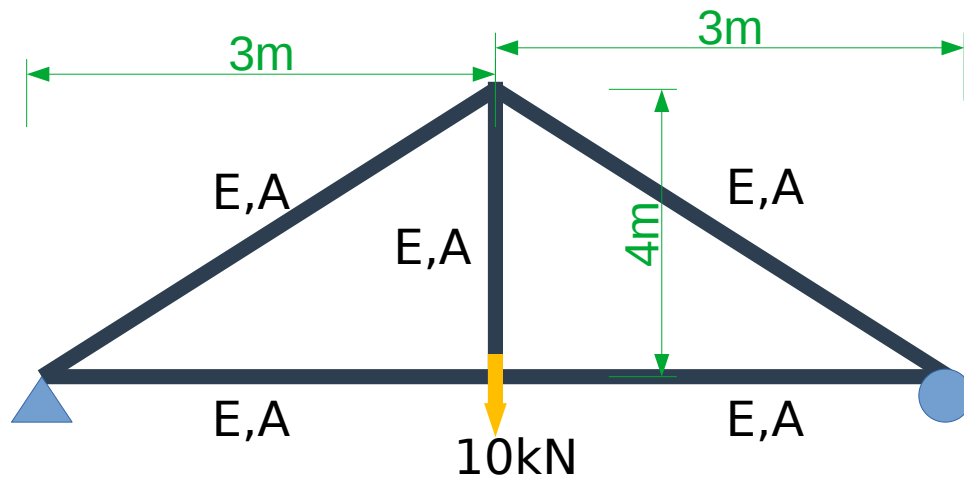
Schenk, O., Gärtner, K. (2011). PARDISO. In: Padua, D. (eds) Encyclopedia of Parallel Computing. Springer, Boston, MA. [https://doi.org/10.1007/978-0-387-09766-4\\_90](https://doi.org/10.1007/978-0-387-09766-4_90)

**Nota: Nodos numerados muy distantes en un mismo elemento genera una matriz dispersa → Eliminacion Gaussiana en algoritmo secuencial es ineficiente !**

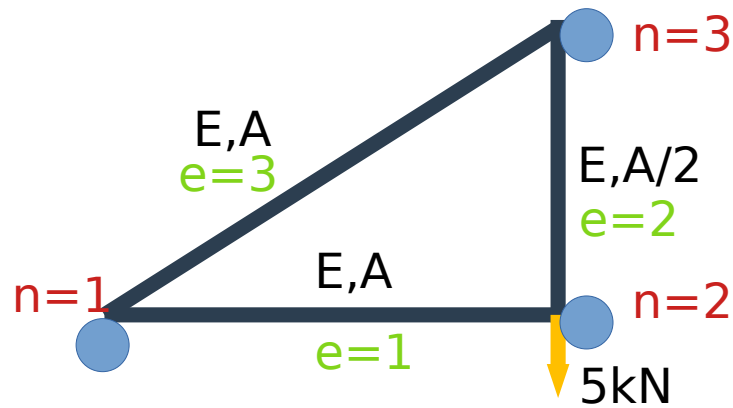


# Sesion 4

## Ejemplo de armadura:



**Nota:** Se puede reducir el sistema por simetria (sistema equivalente en fuerzas)



# Sesion 4

## Ensamblaje practico de matriz de rigidez (mathcad):

$$\begin{array}{llll} nelem := 3 & ndof := 2 & nnode := 2 & npoin := 3 \\ \\ coordg := \begin{bmatrix} 0.0 & 0.0 \\ 3.0 & 0.0 \\ 3.0 & 4.0 \end{bmatrix} & & lnodes := \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 1 \end{bmatrix} & \\ \\ E_{steel} = 2 \cdot 10^{11} & & A_{CHS60.3x4} = 7.075 \cdot 10^{-4} & \\ \\ Eelem := E_{steel} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} & & Aelem := A_{CHS60.3x4} \cdot \begin{bmatrix} 1 \\ 0.5 \\ 1 \end{bmatrix} & \end{array}$$

**nelem** = numero de elementos ; **coordg** = coordenadas globales de nodos  
**ndof** = numero de grados de libertad por nodo ; **lnodes** = topologia de elementos  
**nnode** = numero de nodos por elemento ; **Eelem** = Modulo de Young por elemento  
**npoin** = numero total de nodos en la estructura ; **Aelem** = Area transversal por elemento

# Sesion 4

## Extraer coordenadas de elemento (mathcad):

```

$$\text{coord}_{\text{extr}}(\text{coord}, \text{lnodes}, \text{ielem}, \text{ndof}, \text{nnode}) := \begin{array}{l} \text{for } \text{inode} \in 1.. \text{nnode} \\ \quad \parallel \text{ipoin} \leftarrow \text{lnodes}_{\text{ielem}, \text{inode}} \\ \quad \parallel \text{for } \text{idof} \in 1.. \text{ndof} \\ \quad \quad \parallel \text{xcoord}_{(\text{inode} - 1) \cdot \text{ndof} + \text{idof}} \leftarrow \text{coord}_{\text{ipoin}, \text{idof}} \\ \quad \parallel \\ \text{xcoord} \end{array}$$

```

# Sesion 4

## Ensamblaje de matriz global de rigidez (mathcad):

```
 $K_{2D\_g}(nelem, npoin, nnode, ndof, lnodes, coordg, Eelem, Aelem) :=$   
 $K2D\_g \leftarrow 0 \cdot \text{identity}(npoin \cdot ndof)$   
for  $ielem \in 1..nelem$   
   $coordi \leftarrow coord_{extr}(coordg, lnodes, ielem, ndof, nnode)$   
   $Ei \leftarrow Eelem_{ielem}$   
   $Ai \leftarrow Aelem_{ielem}$   
   $n_p \leftarrow 2$   
   $K2D \leftarrow K_{2D}(Ei, Ai, coordi, n_p, ndof, nnode)$   
  for  $inode \in 1..nnode$   
    for  $idof \in 1..ndof$   
      for  $jnode \in 1..nnode$   
        for  $jdof \in 1..ndof$   
           $i \leftarrow (inode - 1) \cdot ndof + idof$   
           $j \leftarrow (jnode - 1) \cdot ndof + jdof$   
           $ig \leftarrow (lnodes_{ielem, inode} - 1) \cdot ndof + idof$   
           $jg \leftarrow (lnodes_{ielem, jnode} - 1) \cdot ndof + jdof$   
           $K2D\_g_{ig, jg} \leftarrow K2D\_g_{ig, jg} + K2D_{i, j}$   
 $K2D\_g$ 
```

# Sesion 5

## Condiciones de borde mixtas:

$$\underbrace{\begin{bmatrix} \text{1-st} & \text{2-nd} & \dots & \textcolor{blue}{r\text{-th}} & \dots & \dots & \text{n-th} \\ K_{11} & K_{12} & \dots & 0 & \dots & \dots & K_{1n} \\ K_{21} & K_{22} & \dots & 0 & \dots & \dots & K_{2n} \\ \vdots & & \ddots & \vdots & & & \vdots \\ 0 & 0 & \dots & \textcolor{blue}{K_{rr}} & \dots & 0 & 0 \\ & & & \vdots & \ddots & & \\ & & & 0 & & \ddots & \\ & & & 0 & & & K_{nn} \end{bmatrix}}_{\mathbf{K}_g^{mod}} \underbrace{\begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ \textcolor{blue}{u_r} \\ \vdots \\ \vdots \\ u_n \end{Bmatrix}}_{\mathbf{u}_g} = \underbrace{\begin{Bmatrix} F_1 \\ F_2 \\ \vdots \\ F_r \\ \vdots \\ \vdots \\ F_n \end{Bmatrix}}_{\mathbf{F}_{ext}} + \underbrace{\begin{Bmatrix} -\sum^{n_{pu}} K_{1r} \cdot u_r \\ -\sum^{n_{pu}} K_{2r} \cdot u_r \\ \vdots \\ -F_r + K_{rr} \cdot u_r \\ \vdots \\ \vdots \\ -\sum^{n_{pu}} K_{nr} \cdot u_r \end{Bmatrix}}_{\mathbf{F}_{ext}^{add}}$$

$u_r$  = desplazamiento impuesto

Nota: artificio matematico para asegurar que la matriz se pueda invertir  $\det(\mathbf{K}) > 0$

# Sesion 5

## Condiciones de borde mixtas:

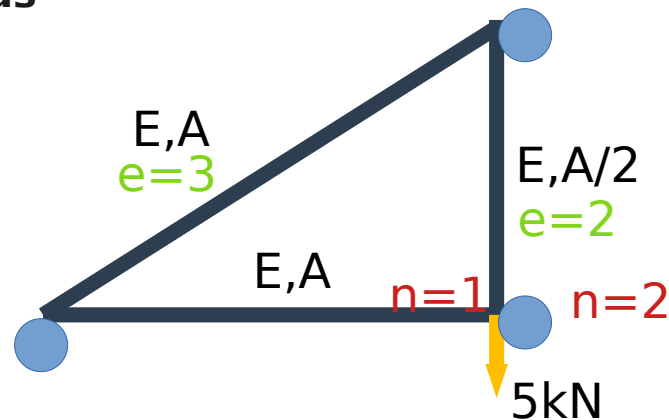
$$\begin{aligned} ntpfix &:= 3 & poinfx &:= \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 0 \\ 3 & 1 & 0 \end{bmatrix} & ufx &:= \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0 \\ 0.0 & 0 \end{bmatrix} \\ Fext &:= [0 \ 0 \ 0 \ -5000 \ 0 \ 0]^T \end{aligned}$$

**ntpfix** = numero total de nodos fijos (global)

**Poinfx** = nodos (globales) y sus grados de libertad impuestos (1= impuesto, 0=libre)

**ufx** = desplazamientos impuestos (incl. valores nulos para completar matriz)

**Fext** = vector de fuerzas externas



# Sesion 5

## Condiciones de borde mixtas (mathcad):

```

Kg_F_2D_mod(nelem,npoin,nnode,ndof,lnodes,coordg,Eelem,Aelem,ntpfix,poinfx,ufx,Fext):=
Kg ← K_2D_g(nelem,npoin,nnode,ndof,lnodes,coordg,Eelem,Aelem)
Kgmod ← Kg
Fgmod ← Fext
for ipfix ∈ 1..ntpfix
    for idof ∈ 1..ndof
        if poinfxipfix,idof+1 = 1
            ig ← (poinfxipfix,1 - 1) • ndof + idof
            for iv ∈ 1..npoin • ndof
                if ig ≠ iv
                    Kgmodig,iv ← 0
                    Kgmodiv,ig ← 0
                    Fgmodiv ← Fgmodiv - Kgiv,ig • ufxipfix,idof
            for ipfix ∈ 1..ntpfix
                for idof ∈ 1..ndof
                    if poinfxipfix,idof+1 = 1
                        ig ← (poinfxipfix,1 - 1) • ndof + idof
                        Fgmodig ← Kgig,ig • ufxipfix,idof
[Kgmod Fgmod]T

```

$$ug := Kgmod^{-1} \cdot Fgmod$$

# Sesion 6

## Calculo de reacciones:

$$\mathbf{F}_{reac} = \mathbf{K}_g \cdot \mathbf{u}_g - \mathbf{F}_{ext}$$

**Nota:** solo filas correspondientes a desplazamientos impuestos son de importancia

## Reacciones (Mathcad):

$$Freac := Kg \cdot ug - Fext$$

**Nota:** Esta forma puede ser refinada para evitar errores de truncamiento en filas sin desplazamientos impuestos. Para caso practico de este curso, esto es suficiente.



# Sesion 6

## Verificacion de analisis:

- 1. Las reacciones siempre deben estar en balance con fuerzas externas**
- 2. Verificar que la deformacion de la estructura sea consistente con las direcciones de carga. No siempre es muy intuitivo.**
- 3. Verificar que el orden de magnitud de los desplazamientos sea dentro de lo esperado**
- 4. Verificar que la malla es la adecuada. Este paso no es usualmente necesario en estructuras mas complejas.**

# Sesion 6

## Post-procesamiento (mathcad):

Extraer desplazamientos (Cartesianos) nodales por elemento

```
uext(ielem, nnode, ndof, lnodes, ug) :=  
  for inode ∈ 1..nnode  
    for idof ∈ 1..ndof  
      ig ← (lnodesielem, inode - 1) • ndof + idof  
      i ← (inode - 1) • ndof + idof  
      uelemi ← ugig  
    uelem
```

Deformaciones por elemento en puntos de integracion (puntos de Gauss)

```
snelem(ielem, lnodes, ndof, nnode, coordg, ug) :=  
  np ← 2  
  for ip ∈ 1..np  
    coordi ← coordextr(coordg, lnodes, ielem, ndof, nnode)  
    Tu ← T2D(coordi)  
    B ← Bmat(ξip, coordi)  
    uelem ← uext(ielem, nnode, ndof, lnodes, ug)  
    snip ← B • Tu • uelem  
  sn
```

# Sesion 6

## Post-procesamiento (mathcad):

Extraer desplazamientos (Cartesianos) nodales por elemento

```
uext(ielem, nnode, ndof, lnodes, ug) :=  
  for inode ∈ 1..nnode  
    for idof ∈ 1..ndof  
      ig ← (lnodesielem, inode - 1) • ndof + idof  
      i ← (inode - 1) • ndof + idof  
      uelemi ← ugig  
    uelem
```

Deformaciones por elemento en puntos de integracion (puntos de Gauss)

```
snelem(ielem, lnodes, ndof, nnode, coordg, ug) :=  
  np ← 2  
  for ip ∈ 1..np  
    coordi ← coordextr(coordg, lnodes, ielem, ndof, nnode)  
    Tu ← T2D(coordi)  
    B ← Bmat(ξip, coordi)  
    uelem ← uext(ielem, nnode, ndof, lnodes, ug)  
    snip ← B • Tu • uelem  
  sn
```

# Sesion 6

## Post-procesamiento (mathcad):

Esfuerzos por elemento en puntos de integracion (puntos de Gauss)

$$selem(ielem,lnodes,ndof,nnode,coordg,ug,Eelem) := \left\| \begin{array}{l} E \leftarrow Eelem_{ielem} \\ sn \leftarrow snelem(ielem,lnodes,ndof,nnode,coordg,ug) \\ s \leftarrow E \cdot sn \\ s \end{array} \right\|$$

Fuerzas axiales por elemento en puntos de integracion (+=tension, -=compresion)

$$fxelem(ielem,lnodes,ndof,nnode,coordg,ug,Eelem,Aelem) := \left\| \begin{array}{l} A \leftarrow Aelem_{ielem} \\ s \leftarrow selem(ielem,lnodes,ndof,nnode,coordg,ug,Eelem) \\ faxial \leftarrow A \cdot s \\ faxial \end{array} \right\|$$

# Sesion 7

## Dinamica estructural (vibraciones libres):

**Tiempo:**

$$M_{el}\ddot{u}_{el} + C_{el}\dot{u}_{el} + K_{el}u_{el} = \cancel{F_{ext}^{el}}$$

$$u_g = \tilde{u}_g \cdot e^{i\omega t}$$



**Frecuencia:**

$$\left( -\omega^2 M_g + i\omega C_g + K_g \right) \tilde{u}_g \cdot e^{i\omega t} = 0$$

**Sin disipacion** ↓

$$\left( -\omega^2 M_g + K_g \right) \tilde{u}_g \cdot e^{i\omega t} = 0$$

**Frecuencias propias y modos de vibracion**

$$T_i = 2 \pi / \omega_i \quad i=1,2,\dots \text{ numero de grados de libertad}$$

# Sesion 7

## Frecuencias propias y modos de vibracion (deformacion relativa)

$$T_n = 2\pi / \omega_n \rightarrow$$

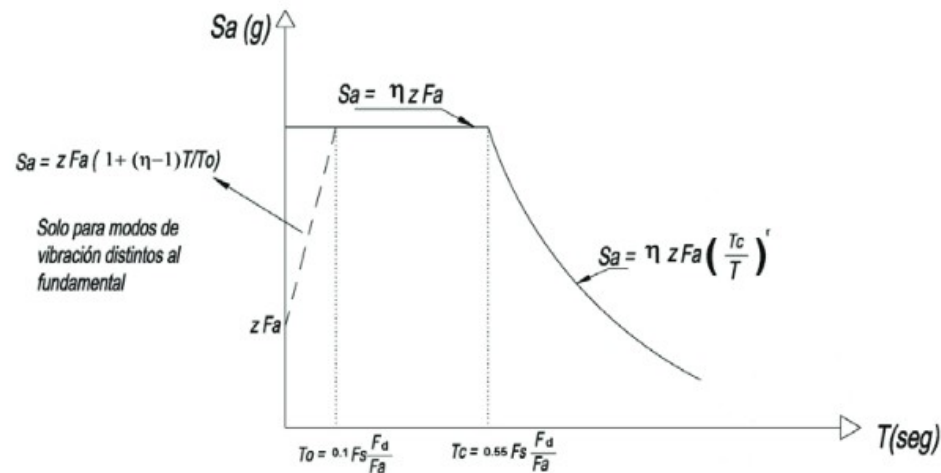
$$\left( -\omega^2 M_g + K_g \right) \tilde{u}_g \cdot e^{i\omega t} = 0$$

$\rightarrow$

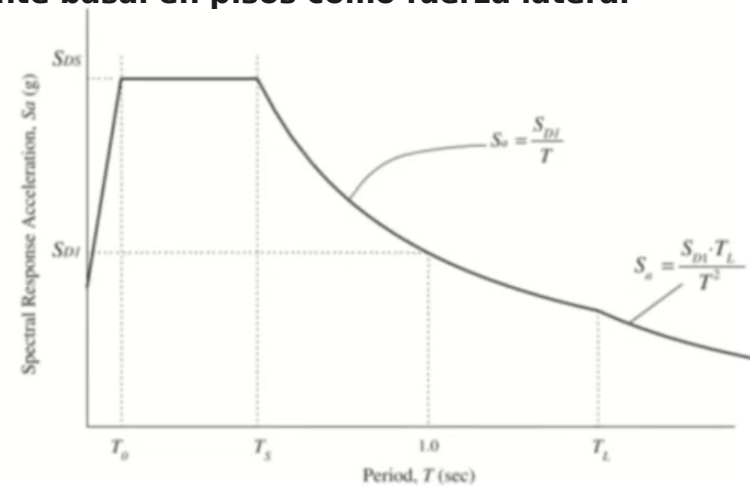
$$\det \left( M_g^{-1} K_g - \omega^2 I \right) = 0$$

**Nota:** Menor frecuencia característica corresponde al periodo fundamental  $T_n$

**Aceleracion espectral  $\rightarrow$  Cortante basal  $\rightarrow$  distribucion de cortante basal en pisos como fuerza lateral**



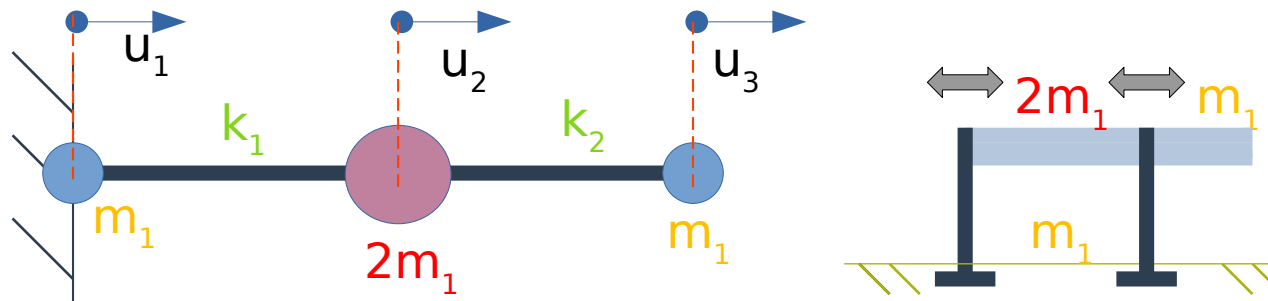
**NEC**



**ASCE 7-16 ( $T_n \approx 0.1N$  [s])**

# Sesion 7

## Frecuencias propias y modos de vibracion



$$k_1 = E_1 A_1 / L_1 ; k_2 = E_2 A_2 / L_2$$

$$c_1 = c_2 = 0$$

$$f_1 = f_2 = f_3 = 0$$

**Determinar las frecuencias propias y modos de vibracion para una constante de rigidez unitaria  $k_1 = k_2 = 1$  y masa unitaria  $m_1 = 1$**

**Nota: Eliminar filas y columnas de ceros debido a condiciones de borde para reducir sistema de ecuaciones. Deformaciones son relativas  $\rightarrow$  esfuerzos y reacciones no son absolutas!**