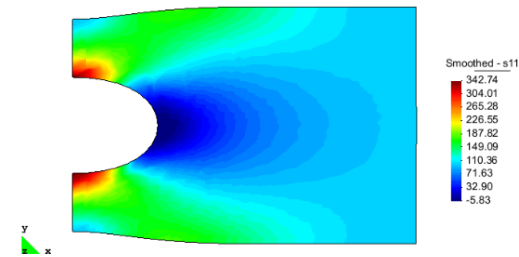
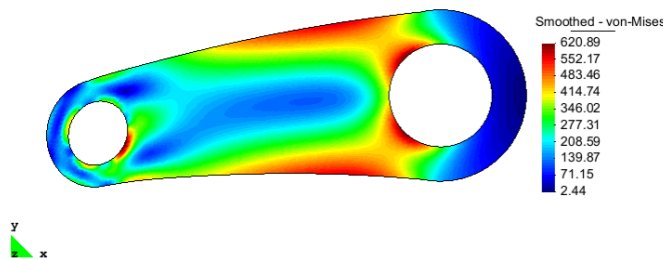
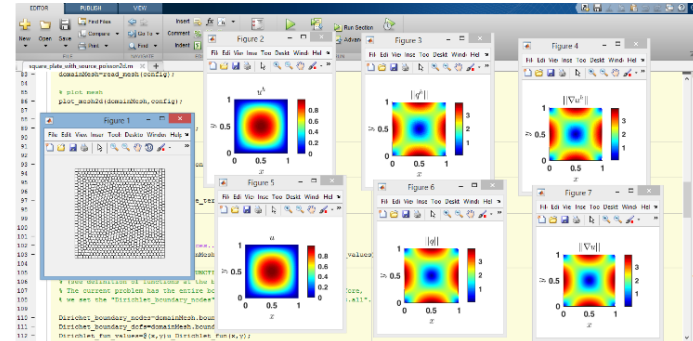
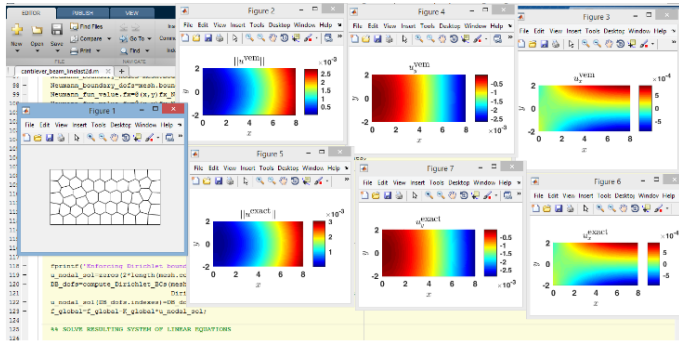


EL MÉTODO DEL ELEMENTO VIRTUAL: TEORÍA Y APLICACIONES USANDO LA LIBRERÍA VEMLab



Alejandro Ortiz-Bernardin, Edgardo Olate-Sanzana, Rodrigo Silva-Valenzuela

CAMLab - Departamento de Ingeniería Mecánica
Universidad de Chile
www.camlab.cl

(*) Financiado por CONICYT/FONDECYT No. 1181192

VEM: nueva herramienta para simulación

Primer artículo:

Beirão da Veiga, L., Brezzi, F., Cangiani, A., Manzini, G., Marini, L.D., Russo, A.: **Basic principles of virtual element methods**. Math. Models Methods Appl. Sci. 23(1), 199–214 **(2013)**

Otras principales contribuciones:

Beirão da Veiga, L., Brezzi, F., Marini, L.D.: Virtual elements for **linear elasticity** problems. SIAM J. Numer. Anal. 51(2), 794–812 **(2013)**

Brezzi, F., Marini, L.D.: Virtual Element Methods for **plate bending** problems. Comput. Methods Appl. Mech. Engrg. 253, 455–462 **(2013)**

Beirão da Veiga, L., Brezzi, F., Marini, L.D., Russo, A.: **The Hitchhiker's Guide to the Virtual Element Method**. Math. Models Methods Appl. Sci. 24(08), 1541–1573 **(2014)**

Gain, A.L., Talischi, C., Paulino, G.H.: On the virtual element method for **three-dimensional linear elasticity** problems on arbitrary polyhedral meshes. Comput. Methods Appl. Mech. Engrg. 282(0), 132–160 **(2014)**

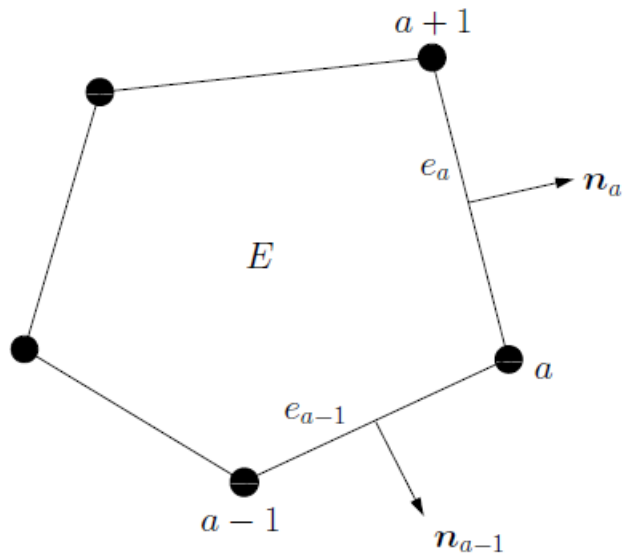
Chi, H., Beirão da Veiga, L., Paulino, G.: Some basic formulations of the virtual element method (VEM) for **finite deformations**. Comput. Methods Appl. Mech. Engrg. 318, 148–192 **(2017)**

Wriggers, P., Reddy, B.D., Rust, W., Hudobivnik, B.: Efficient virtual element formulations for compressible and incompressible **finite deformations**. Comput. Mech. 60(2), 253–268 **(2017)**

Artioli, E., Beirão da Veiga, L., Lovadina, C., Sacco, E.: Arbitrary order 2D virtual elements for polygonal meshes: part II, **inelastic problem**. Comput. Mech. 60(4), 643-657 **(2017)**

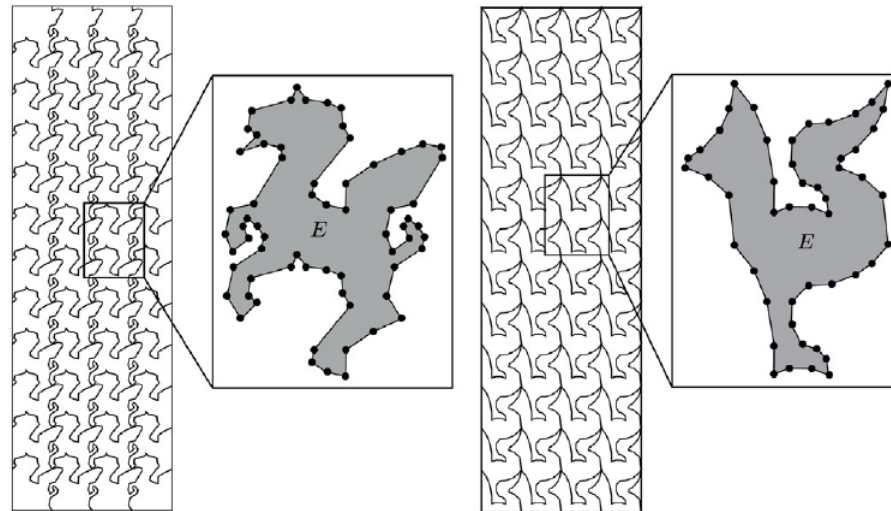
Beirão da Veiga, L., Lovadina, Vacca, G.: Virtual Elements for the **Navier-Stokes** Problem on Polygonal Meshes. SIAM J. Numer. Anal. 56(3), 1210-1242 **(2018)**

VEM: nueva herramienta para simulación



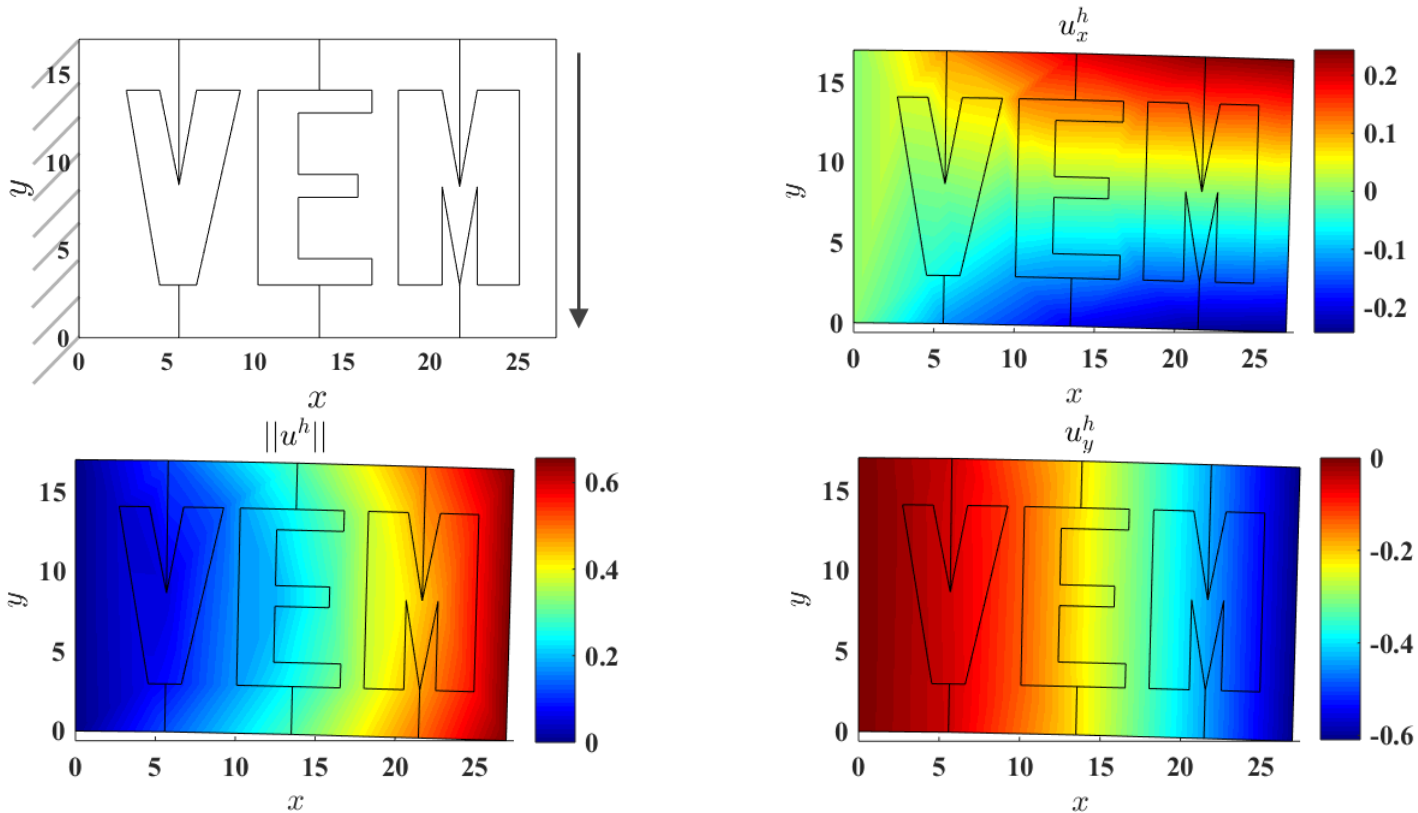
Elemento poligonal convexo

La formulación admite polígonos bien generales



(Chi et. al, CMAME, 2017)

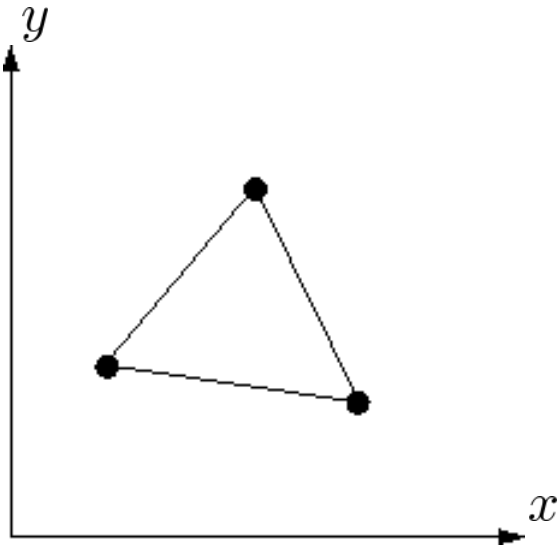
Elementos poligonales bien generales...



Malla compuesta por **7 elementos** poligonales irregulares

Método del Elemento Virtual – Elast. Lineal Estática

Polígono de tres lados



Forma bilineal \rightarrow matriz de rigidez:

$$a_E(\mathbf{u}^h, \mathbf{v}^h) = \int_E \boldsymbol{\varepsilon}^T(\mathbf{v}^h) \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{u}^h) dx$$

\mathbf{u}^h : desplazamiento (función de prueba)

\mathbf{v}^h : desplazamiento (función de peso)

Base de construcción aproximación \mathbf{u}^h y \mathbf{v}^h :

$$\mu = \{1, x, y\}$$

Movimientos de cuerpo rígido:

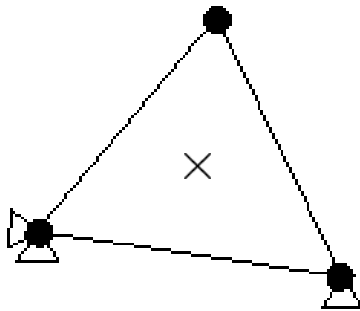
- dos traslaciones
- una rotación

Rango de la matriz de rigidez elemental:

$$R = 6 \text{ grados de libertad} - 3 \text{ movimientos de cuerpo rígido} = 3$$

Método del Elemento Virtual – Elast. Lineal Estática

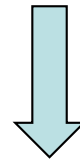
Polígono de tres lados



Integración numérica con 1 punto:

$$a_E^h(\mathbf{u}^h, \mathbf{v}^h) = \boldsymbol{\varepsilon}^T(\mathbf{v}^h(\times)) \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{u}^h(\times)) w$$

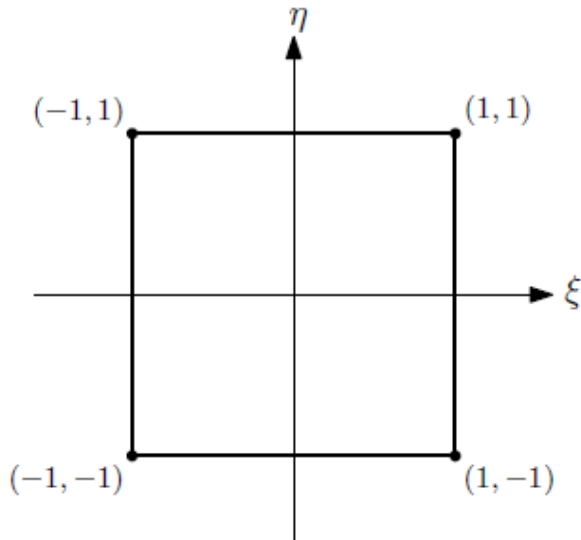
\mathbf{D} es una matriz de tamaño 3x3



- $R = 3$ (rango completo)
- Matriz de rigidez bien condicionada/estable
- **Matriz de rigidez integrada exactamente**

Método del Elemento Virtual – Elast. Lineal Estática

Polígono de cuatro lados



Forma bilineal \rightarrow matriz de rigidez:

$$a_E(\mathbf{u}^h, \mathbf{v}^h) = \int_E \boldsymbol{\varepsilon}^T(\mathbf{v}^h) \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{u}^h) dx$$

Base de construcción aproximación \mathbf{u}^h y \mathbf{v}^h :

$$\boldsymbol{\mu} = \{1, \xi, \eta, \xi\eta\}$$

Movimientos de cuerpo rígido:

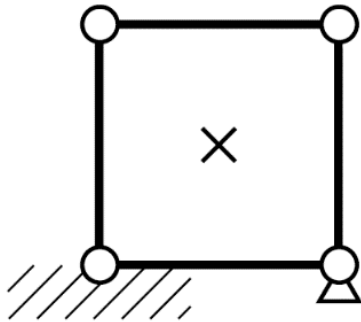
- dos traslaciones
- una rotación

Rango de la matriz de rigidez elemental:

$$R = 8 \text{ grados de libertad} - 3 \text{ movimientos de cuerpo rígido} = 5$$

Método del Elemento Virtual – Elast. Lineal Estática

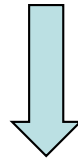
Polígono de cuatro lados



Integración numérica con 1 punto:

$$a_E^h(\mathbf{u}^h, \mathbf{v}^h) = \boldsymbol{\varepsilon}^T(\mathbf{v}^h(\times)) \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{u}^h(\times)) w$$

\mathbf{D} es una matriz de tamaño 3×3

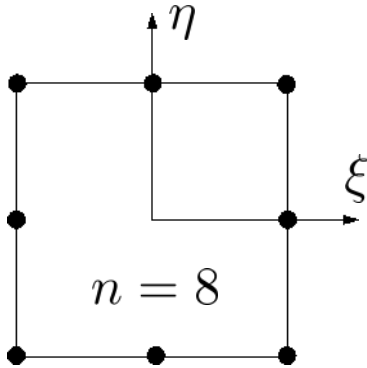


- $R = 3 < 5$ (rango incompleto)
- 2 modos espurios (energía de deformación nula)
- **Matriz de rigidez subintegrada (inexacta)**
- Monomio $\xi\eta$ responsable de los modos espurios (1 c/grado de libertad)



Método del Elemento Virtual – Elast. Lineal Estática

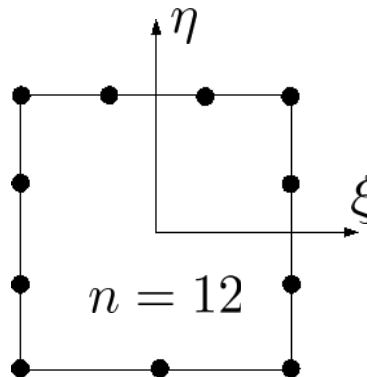
...y en polígonos “serendípitos” con cuadratura de 1 punto



$$\mu = \{1, \xi, \eta, \xi^2, \xi\eta, \eta^2, \xi^2\eta, \xi\eta^2\}$$

$$m = 2 \times (\text{monomios orden superior}) = 2 \times 5 = 10 \text{ modos}$$

(integración inexacta / error de cuadratura)



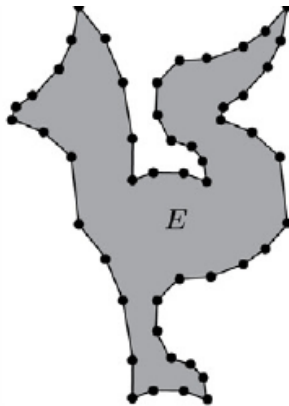
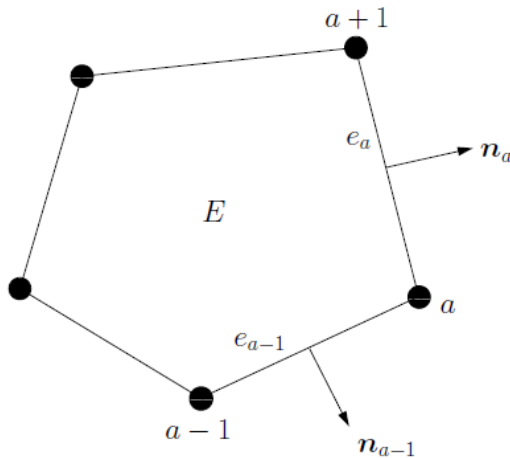
$$\mu = \{1, \xi, \eta, \xi^2, \xi\eta, \eta^2, \xi^3, \xi^2\eta, \xi\eta^2, \eta^3, \xi^3\eta, \eta\xi^3\}$$

$$m = 2 \times (\text{monomios orden superior}) = 2 \times 9 = 18 \text{ modos}$$

(integración inexacta / error de cuadratura)

Método del Elemento Virtual – Elast. Lineal Estática

Polígono arbitrario (aproximación lineal)



Base de construcción aproximación:

$$\mu := \{1, x, y\} \cup \{x^2, xy, \dots\}$$

Espacios de desplazamientos:

$\mathcal{P}(E)$ = espacio de desplazamientos lineales

$\mathcal{H}(E)$ = espacio términos de orden superior

$$\therefore \mathbf{u}^h, \mathbf{v}^h \in \mathcal{W}(E) \supseteq \mathcal{P}(E)$$

Sea $\Pi : \mathcal{W}(E) \rightarrow \mathcal{P}(E)$, $\Pi \mathbf{p} = \mathbf{p}$, $\forall \mathbf{p} \in \mathcal{P}(E)$.
Entonces,

$$\mathbf{u}^h = \underbrace{\Pi \mathbf{u}^h}_{\in \mathcal{P}} + (\mathbf{u}^h - \Pi \mathbf{u}^h)$$

$$\mathbf{v}^h = \underbrace{\Pi \mathbf{v}^h}_{\in \mathcal{P}} + \underbrace{(\mathbf{v}^h - \Pi \mathbf{v}^h)}_{\in \mathcal{H}}$$

Método del Elemento Virtual – Elast. Lineal Estática

Construcción del método

A nivel elemental se exige que la proyección satisfaga la siguiente condición de ortogonalidad:

$$a_E(\mathbf{p}, \mathbf{v}^h - \Pi \mathbf{v}^h) = 0 \quad \forall \mathbf{p} \in \mathcal{P}(E), \quad \mathbf{v}^h \in \mathcal{W}(E)$$

la que permite obtener la siguiente “descomposición del elemento virtual”:

$$\begin{aligned} a_E(\mathbf{u}^h, \mathbf{v}^h) &= a_E(\Pi \mathbf{u}^h + (\mathbf{u}^h - \Pi \mathbf{u}^h), \Pi \mathbf{v}^h + (\mathbf{v}^h - \Pi \mathbf{v}^h)) \\ &= a_E(\Pi \mathbf{u}^h, \Pi \mathbf{v}^h) + a_E(\mathbf{u}^h - \Pi \mathbf{u}^h, \mathbf{v}^h - \Pi \mathbf{v}^h) \end{aligned}$$

- Primer término puede ser calculado exactamente (términos lineales)
- Segundo término no está bien definido (contiene elementos $\in \mathcal{H}$)

Aproximando el segundo término por uno que pueda ser calculado exactamente (s_E), podemos escribir

$$a_E^h(\mathbf{u}^h, \mathbf{v}^h) := \underbrace{a_E(\Pi \mathbf{u}^h, \Pi \mathbf{v}^h)}_{\text{“consistencia”}} + \underbrace{s_E(\mathbf{u}^h - \Pi \mathbf{u}^h, \mathbf{v}^h - \Pi \mathbf{v}^h)}_{\text{“estabilidad”}}$$



Método del Elemento Virtual – Elast. Lineal Estática

La **condición de ortogonalidad** se usa para obtener Π del siguiente sistema:

$$a_E(\Pi \mathbf{v}^h, \mathbf{q}) = a_E(\mathbf{v}^h, \mathbf{q}) \quad \forall \mathbf{q} \in \mathcal{P}(E)$$

Pero como \mathbf{v}^h es lineal, la solución se obtiene solo hasta una constante. Para fijar esa constante, agregamos la siguiente ecuación al sistema:

$$\frac{1}{N} \sum_{a=1}^N \Pi \mathbf{v}^h(\mathbf{x}_a) = \frac{1}{N} \sum_{a=1}^N \mathbf{v}^h(\mathbf{x}_a)$$

donde N es el número de nodos del elemento poligonal y \mathbf{x}_a es la coordenada del nodo a .

El proyector obtenido del sistema anterior garantiza que la **forma bilineal** es:

- **consistente** (el test de la parcela se satisface)
- **estable** (limitada por abajo y por arriba)

Consistencia + Estabilidad \rightarrow Convergencia

(Teorema de equivalencia de Lax)



Método del Elemento Virtual – Elast. Lineal Estática

Usando la descomposición del elemento virtual, en su forma original y en su forma aproximada, respectivamente,

$$a_E(\mathbf{u}^h, \mathbf{v}^h) = a_E(\Pi \mathbf{u}^h, \Pi \mathbf{v}^h) + a_E(\mathbf{u}^h - \Pi \mathbf{u}^h, \mathbf{v}^h - \Pi \mathbf{v}^h)$$

$$a_E^h(\mathbf{u}^h, \mathbf{v}^h) := a_E(\Pi \mathbf{u}^h, \Pi \mathbf{v}^h) + s_E(\mathbf{u}^h - \Pi \mathbf{u}^h, \mathbf{v}^h - \Pi \mathbf{v}^h)$$

la **consistencia** se garantiza del siguiente modo:

Para $\mathbf{p} \in \mathcal{P}(E)$ implica que:

$$a_E(\mathbf{p} - \Pi \mathbf{p}, \mathbf{v}^h - \Pi \mathbf{v}^h) = 0, \quad s_E(\mathbf{p} - \Pi \mathbf{p}, \mathbf{v}^h - \Pi \mathbf{v}^h) = 0$$

Por lo tanto, $\forall \mathbf{v}^h \in \mathcal{W}(E)$ se cumple

$$a_E^h(\mathbf{p}, \mathbf{v}^h) = a_E(\Pi \mathbf{p}, \Pi \mathbf{v}^h) = a_E(\mathbf{p}, \Pi \mathbf{v}^h) = a_E(\mathbf{p}, \mathbf{v}^h)$$

Método del Elemento Virtual – Elast. Lineal Estática

Usando la descomposición del elemento virtual, en su forma original y en su forma aproximada, respectivamente,

$$a_E(\mathbf{u}^h, \mathbf{v}^h) = a_E(\Pi \mathbf{u}^h, \Pi \mathbf{v}^h) + a_E(\mathbf{u}^h - \Pi \mathbf{u}^h, \mathbf{v}^h - \Pi \mathbf{v}^h)$$

$$a_E^h(\mathbf{u}^h, \mathbf{v}^h) := a_E(\Pi \mathbf{u}^h, \Pi \mathbf{v}^h) + s_E(\mathbf{u}^h - \Pi \mathbf{u}^h, \mathbf{v}^h - \Pi \mathbf{v}^h)$$

la **estabilidad** se garantiza del siguiente modo: $\forall \mathbf{v}^h \in \mathcal{W}(E)$

$$\begin{aligned} a_E^h(\mathbf{v}^h, \mathbf{v}^h) &\leq a_E(\Pi \mathbf{v}^h, \Pi \mathbf{v}^h) + c_1 a_E(\mathbf{v}^h - \Pi \mathbf{v}^h, \mathbf{v}^h - \Pi \mathbf{v}^h) \\ &\leq \max\{1, c_1\} (a_E(\Pi \mathbf{v}^h, \Pi \mathbf{v}^h) + a_E(\mathbf{v}^h - \Pi \mathbf{v}^h, \mathbf{v}^h - \Pi \mathbf{v}^h)) = \alpha^* a_E(\mathbf{v}^h, \mathbf{v}^h) \end{aligned}$$

Similarmente, $\forall \mathbf{v}^h \in \mathcal{W}(E)$

$$a_E^h(\mathbf{v}^h, \mathbf{v}^h) \geq \min\{1, c_0\} (a_E(\Pi \mathbf{v}^h, \Pi \mathbf{v}^h) + a_E(\mathbf{v}^h - \Pi \mathbf{v}^h, \mathbf{v}^h - \Pi \mathbf{v}^h)) = \alpha_* a_E(\mathbf{v}^h, \mathbf{v}^h)$$

$\therefore \exists$ dos constantes $\alpha_* > 0$ and $\alpha^* > 0$, independientes de h y E , tal que $\forall \mathbf{v}^h \in \mathcal{W}(E)$,

$$\alpha_* a_E(\mathbf{v}^h, \mathbf{v}^h) \leq a_E^h(\mathbf{v}^h, \mathbf{v}^h) \leq \alpha^* a_E(\mathbf{v}^h, \mathbf{v}^h)$$

↑ Revela la importancia de usar s_E

Método del Elemento Virtual – Elast. Lineal Estática

Forma explícita de las proyecciones

La proyección que satisface las condición de ortogonalidad es [Gain et al., CMAME, 2014]:

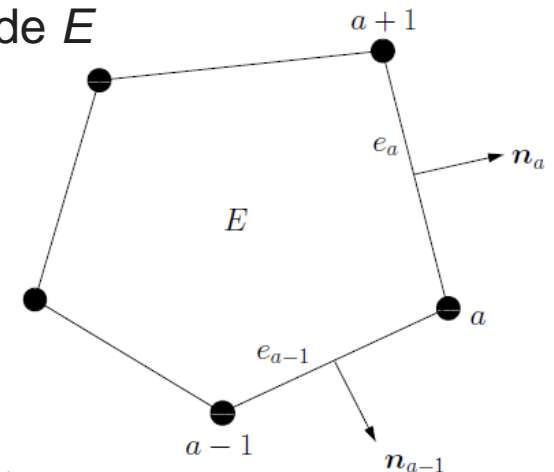
$$\Pi \mathbf{v}^h = \widehat{\boldsymbol{\varepsilon}}(\mathbf{v}^h) \cdot [\mathbf{x} - \bar{\mathbf{x}}] + \widehat{\boldsymbol{\omega}}(\mathbf{v}^h) \cdot [\mathbf{x} - \bar{\mathbf{x}}] + \overline{\mathbf{v}^h}$$

donde

$$\bar{f} = \frac{1}{N} \sum_{j=1}^N f(\mathbf{x}_j) \quad , \quad N = \# \text{ nodos de } E \quad , \quad |E| = \text{área de } E$$

$$\widehat{\boldsymbol{\varepsilon}}(\mathbf{v}) = \frac{1}{|E|} \int_E \boldsymbol{\varepsilon}(\mathbf{v}) \, d\mathbf{x} = \frac{1}{2|E|} \int_{\partial E} (\mathbf{v} \otimes \mathbf{n} + \mathbf{n} \otimes \mathbf{v}) \, ds$$

$$\widehat{\boldsymbol{\omega}}(\mathbf{v}) = \frac{1}{|E|} \int_E \boldsymbol{\omega}(\mathbf{v}) \, d\mathbf{x} = \frac{1}{2|E|} \int_{\partial E} (\mathbf{v} \otimes \mathbf{n} - \mathbf{n} \otimes \mathbf{v}) \, ds$$



Discretizando componentes del desplazamiento, $v_i^h = \phi_a$, obtenemos las expresiones discretas de $\widehat{\boldsymbol{\varepsilon}}(\mathbf{v}^h)$, $\widehat{\boldsymbol{\omega}}(\mathbf{v}^h)$, $\overline{\mathbf{v}^h}$, y por lo tanto de $\Pi \mathbf{v}^h$.

Método del Elemento Virtual – Elast. Lineal Estática

¿Por qué “virtual”?

$$\widehat{\varepsilon}_{ij}^h(\phi_a) = \frac{1}{2|E|} \int_{\partial E} (\phi_a n_i + \phi_a n_j) ds, \quad \widehat{\omega}_{ij}^h(\phi_a) = \frac{1}{2|E|} \int_{\partial E} (\phi_a n_i - \phi_a n_j) ds$$

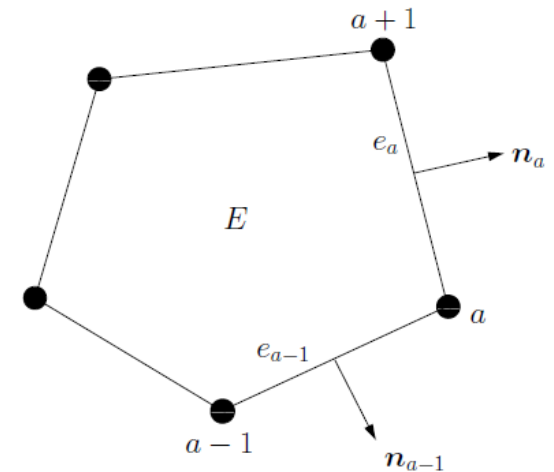
Suponiendo que ϕ_a es **lineal y continua en el contorno del elemento** y que $\phi_a(\mathbf{x}_b) = \delta_{ab}$, permite los siguientes cálculos algebraicos:

$$\frac{1}{2|E|} \int_{\partial E} \phi_a n_i ds = \frac{1}{4|E|} (|e_{a-1}| \{n_i\}_{a-1} + |e_a| \{n_i\}_a), \quad i = 1, 2$$

$$\overline{\mathbf{v}}^h \Rightarrow \overline{\phi}_a = \frac{1}{N} \sum_{j=1}^N \phi_a(\mathbf{x}_j) = \frac{1}{N}$$

$$\therefore \Pi \mathbf{v}^h = \widehat{\varepsilon}(\mathbf{v}^h) \cdot [\mathbf{x} - \overline{\mathbf{x}}] + \widehat{\omega}(\mathbf{v}^h) \cdot [\mathbf{x} - \overline{\mathbf{x}}] + \overline{\mathbf{v}}^h$$

No requiere evaluación de ϕ_a 's (son “**virtuales**”)



Método del Elemento Virtual – Elast. Lineal Estática

Construcción de la matriz de rigidez elemental

$$(\mathbf{K}_E^{\text{VEM}})_{ab} = a_E(\Pi\phi_a, \Pi\phi_b) + s_E(\phi_a - \Pi\phi_b, \phi_a - \Pi\phi_b)$$



$$(\mathbf{K}_E^{\text{VEM}})_{ab} = (\mathbf{K}_E^{\text{c}})_{ab} + (\mathbf{K}_E^{\text{s}})_{ab}$$

donde la **matriz de consistencia** se calcula algebraicamente como:

$$(\mathbf{K}_E^{\text{c}})_{ab} = a_E(\Pi\phi_a, \Pi\phi_b)$$

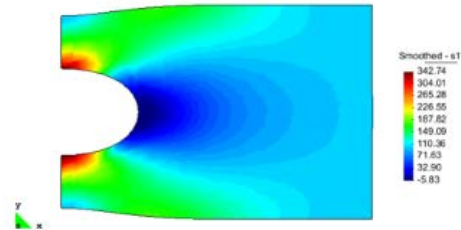
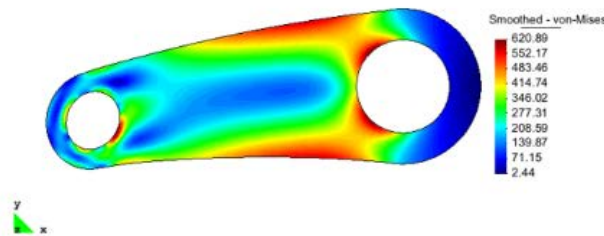
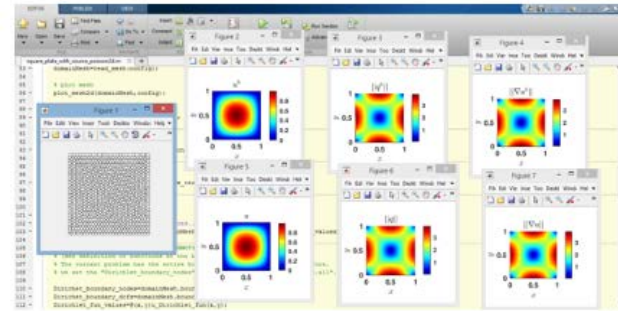
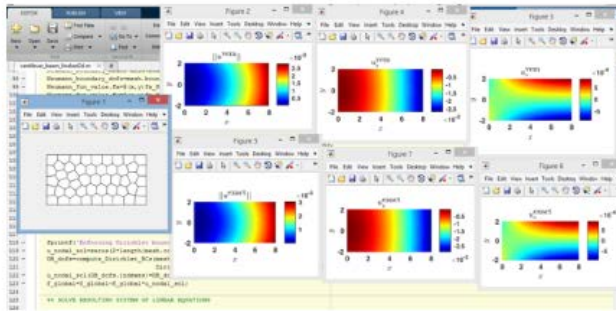
La matriz de estabilidad se construye mediante

$$(\mathbf{K}_E^{\text{s}})_{ab} = s_E(\phi_a - \Pi\phi_a, \phi_b - \Pi\phi_b) = (1 - \Pi)s_E(\phi_a, \phi_b)(1 - \Pi)$$

y definiendo $s_E(\phi_a, \phi_b) := \alpha\delta_{ab}$ (α factor de escalamiento), se obtiene la siguiente expresión algebraica para la **matriz de estabilidad**:

$$(\mathbf{K}_E^{\text{s}})_{ab} = \alpha(1 - \Pi)\delta_{ab}(1 - \Pi)$$

VEMLab: a MATLAB library for the virtual element method

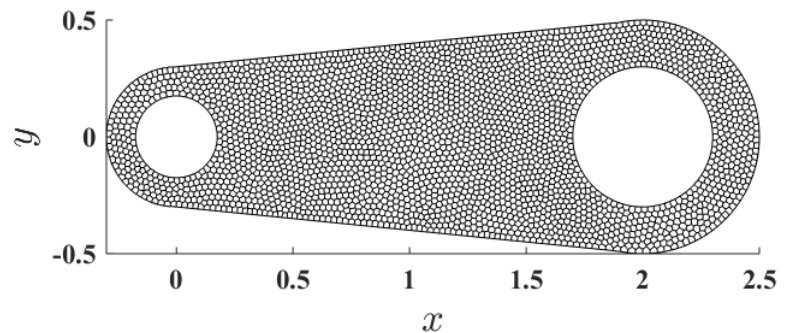
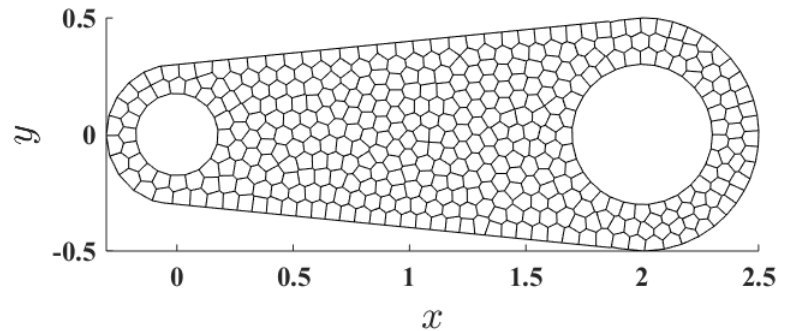


- Web: <http://camlab.cl/vemlab>
- Repositorio: <https://github.com/cemcen/vemlab>

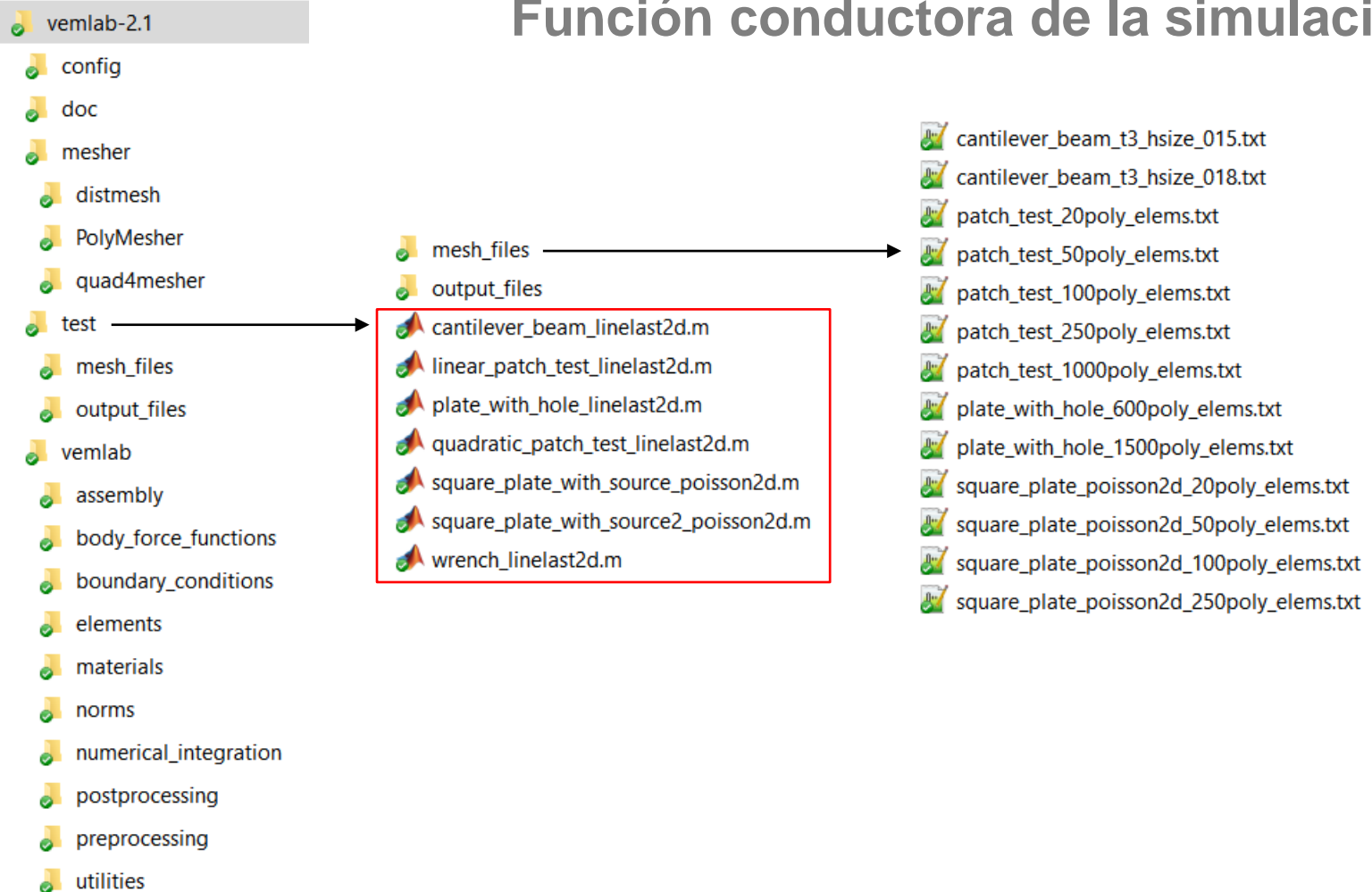
Generación de mallas poligonales

- ✓ vemlab-2.1
 - ✓ config
 - ✓ doc
 - ✓ mesher →
 - ✓ distmesh
 - ✓ PolyMesher
 - ✓ quad4mesher
 - ✓ create_polygonal_mesh.m
 - ✓ create_quadilateral_mesh.m
 - ✓ create_triangular_mesh.m
 - ✓ distmesh
 - ✓ PolyMesher
 - ✓ quad4mesher
 - ✓ test
 - ✓ mesh_files
 - ✓ output_files
 - ✓ vemlab
 - ✓ assembly
 - ✓ body_force_functions
 - ✓ boundary_conditions
 - ✓ elements
 - ✓ materials
 - ✓ norms
 - ✓ numerical_integration
 - ✓ postprocessing
 - ✓ preprocessing
 - ✓ utilities

C. Talischi, G. H. Paulino, A. Pereira, I. F. M. Menezes,
PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab,
Struct Multidisc Optim (2012) 45:309–328



Función conductora de la simulación



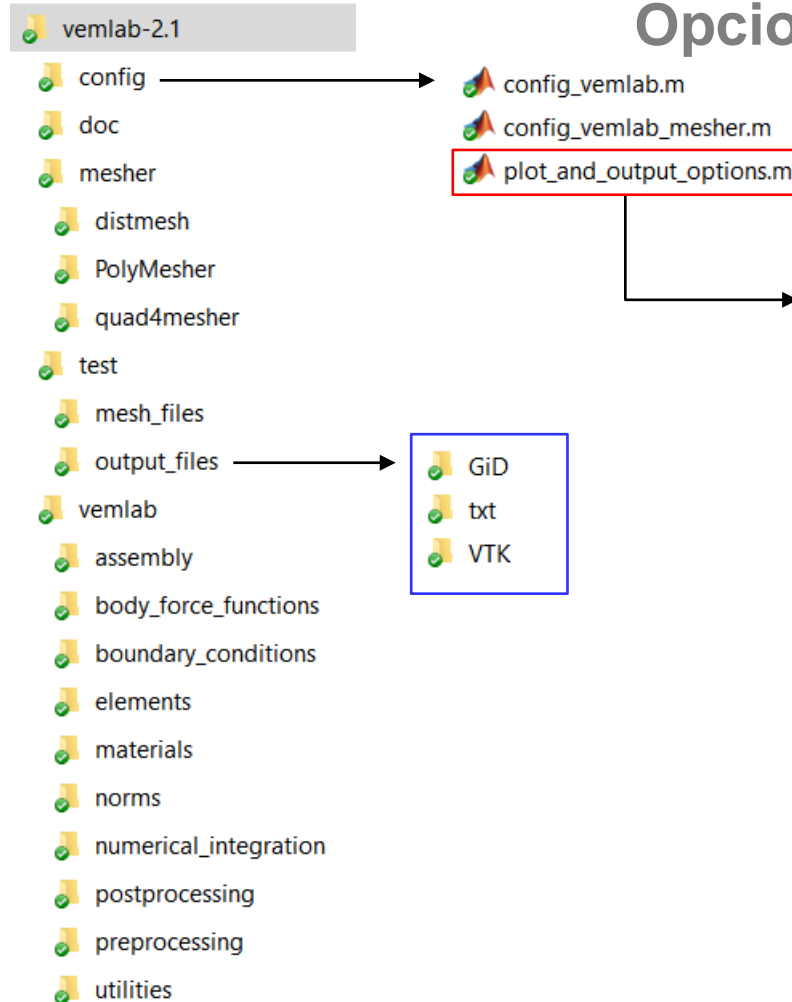


Función conductora de la simulación

- mesh_files
- output_files
- cantilever_beam_linelast2d.m** →
- linear_patch_test_linelast2d.m
- plate_with_hole_linelast2d.m
- quadratic_patch_test_linelast2d.m
- square_plate_with_source_poisson2d.m
- square_plate_with_source2_poisson2d.m
- wrench_linelast2d.m

- Definición del material
- Especificación archivo de malla
- Elección método ('VEM2D', 'FEM2DT3', 'FEM2DQ4')
- Elección módulo ('LinearElastostatics', 'Poisson')
- Definición fuerza de cuerpo
- Definición condiciones de contorno de Neumann
- Definición condiciones de contorno de Dirichlet
- Llamada función de ensamble
- Llamada función que aplica condiciones de contorno
- Llamada función que resuelve el sistema lineal
- Llamada función de postproceso

Opciones de postproceso y salidas



```

%% GENERAL

plot_mesh='yes';
plot_mesh_over_results='no';
write_solutions_to_text_file='yes';
write_solutions_to_GiD_file='yes';
write_solutions_to_VTK_file='yes';

%% POISSON MODULE

% plotting of main variables
poisson2d_plot_scalar_field.u='yes';

% plotting of fluxes
poisson2d_plot_flux.qx='no';
poisson2d_plot_flux.qy='no';
poisson2d_plot_flux.qnorm='yes'; % norm of the flux

% plotting of gradients
poisson2d_plot_grad.dx='no';
poisson2d_plot_grad.dy='no';
poisson2d_plot_grad.dnorm='yes'; % norm of the gradient

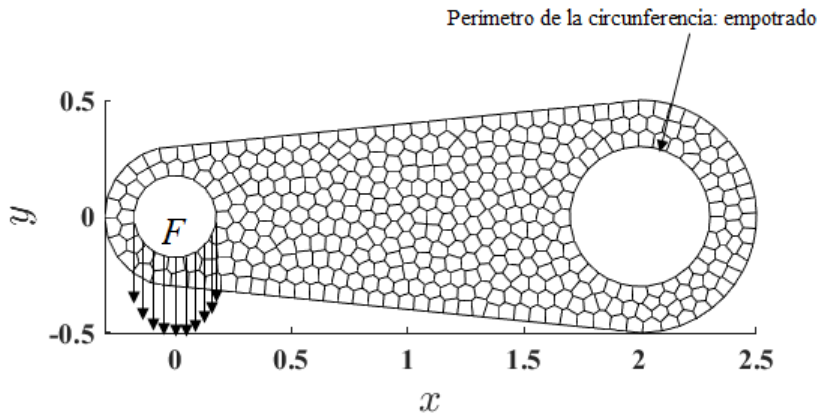
%% LINELAST MODULE

% plotting of main variables
linelast2d_plot_displacement.ux='yes';
linelast2d_plot_displacement.uy='yes';
linelast2d_plot_displacement.unorm='yes'; % norm of the displacement

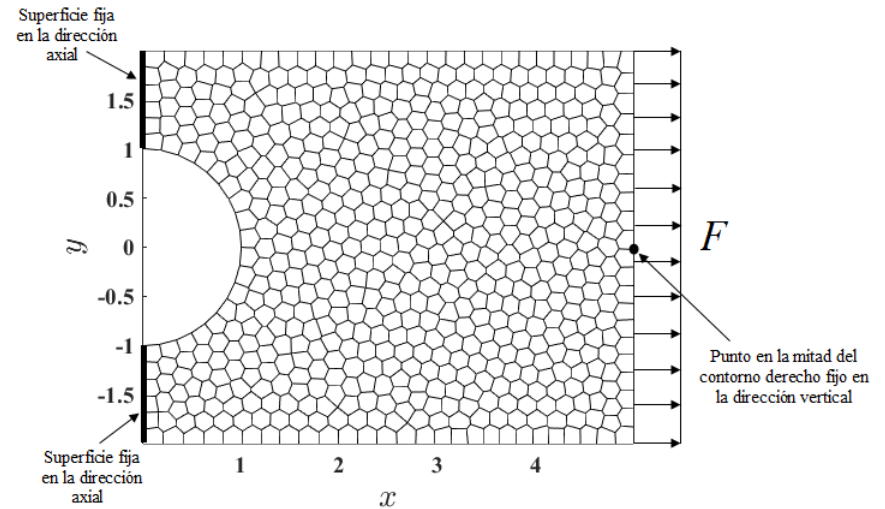
% plotting of stresses
linelast2d_plot_stress.s11='yes';
linelast2d_plot_stress.s12='no';
linelast2d_plot_stress.s22='no';
linelast2d_plot_stress.s33='no';
linelast2d_plot_stress.s1='no';
linelast2d_plot_stress.s2='no';
linelast2d_plot_stress.s3='no';
linelast2d_plot_stress.vd='no';

% plotting of strains
linelast2d_plot_strain.e11='no';
linelast2d_plot_strain.e12='no';
linelast2d_plot_strain.e22='no';
linelast2d_plot_strain.e33='no';
linelast2d_plot_strain.e1='no';
linelast2d_plot_strain.e2='no';
linelast2d_plot_strain.e3='no';
  
```


Llave de torque y placa perforada

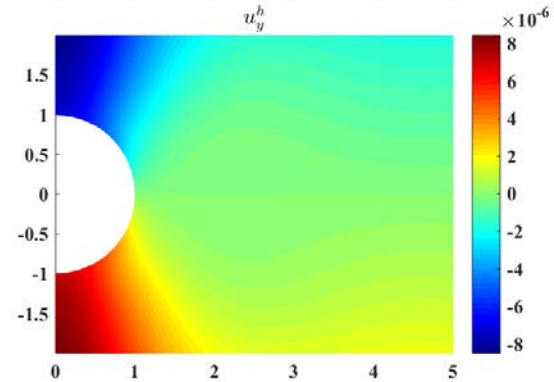
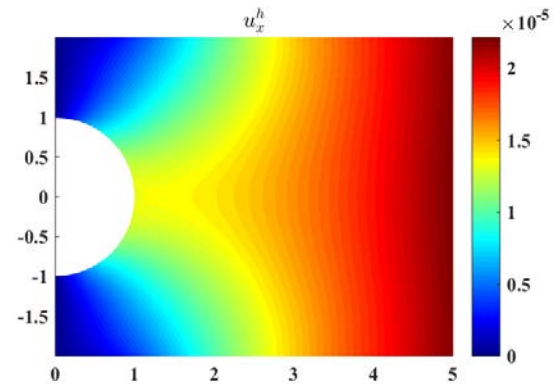
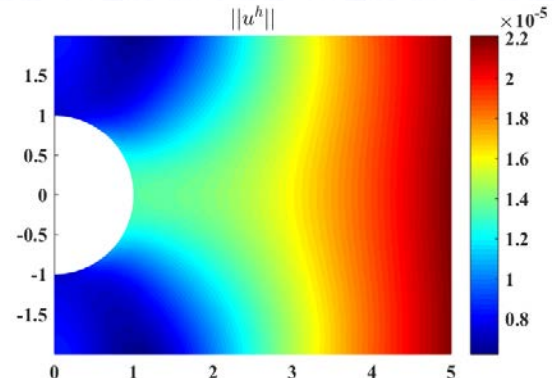
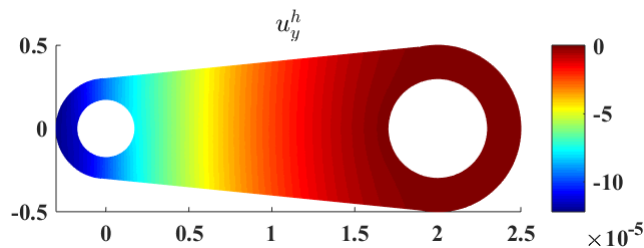
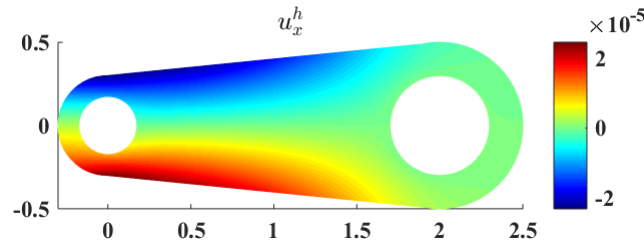
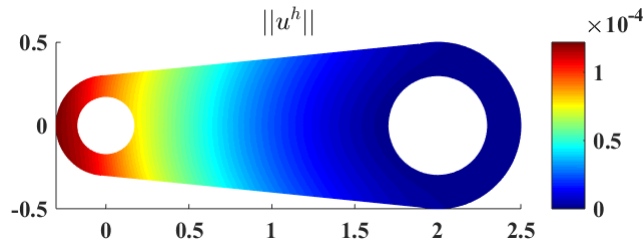


- $F = 35$
- Material: $E = 3e7$, $\nu = 0.3$
- Esfuerzo plano

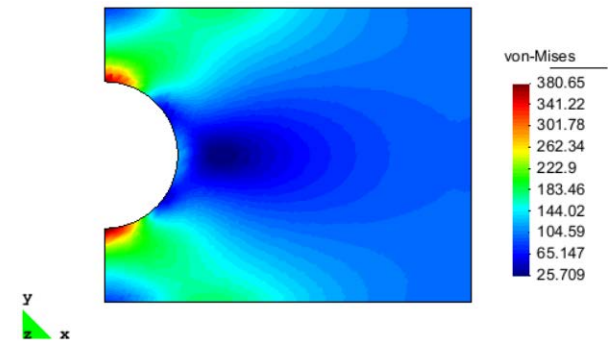
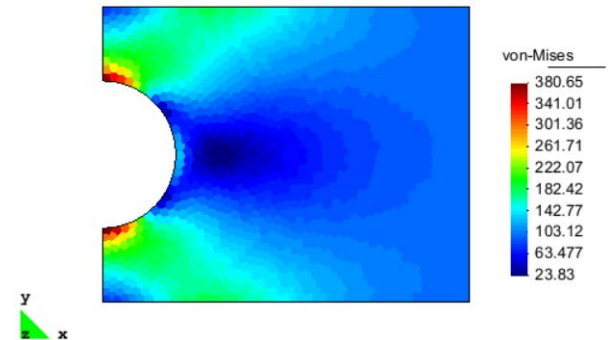
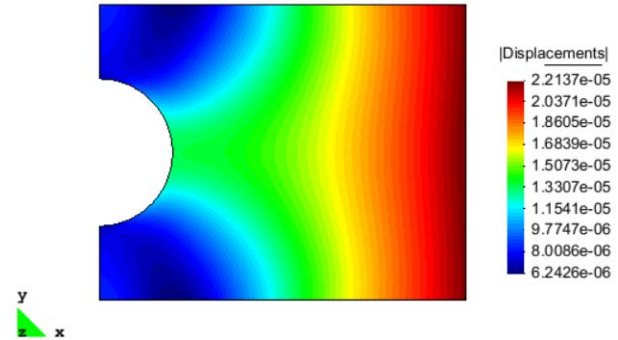
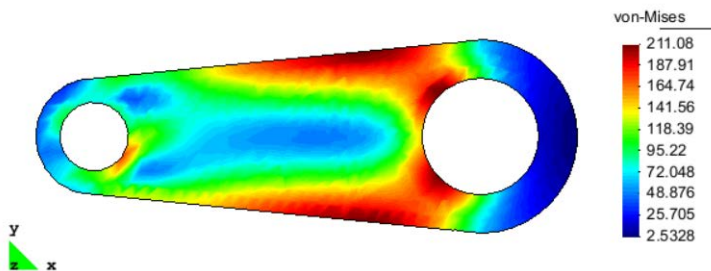
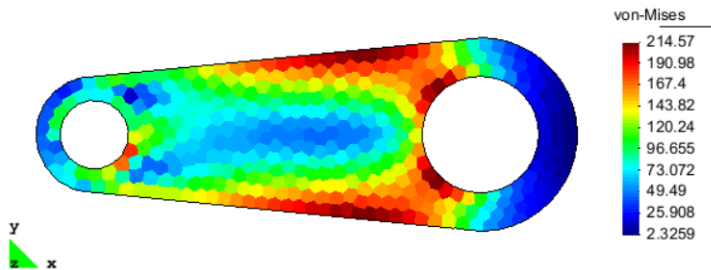
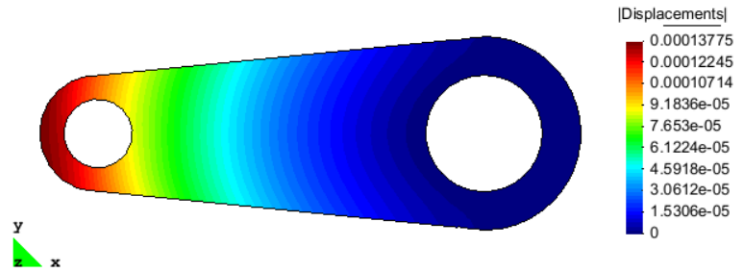


- $F = 100$
- Material: $E = 3e7$, $\nu = 0.3$
- Esfuerzo plano

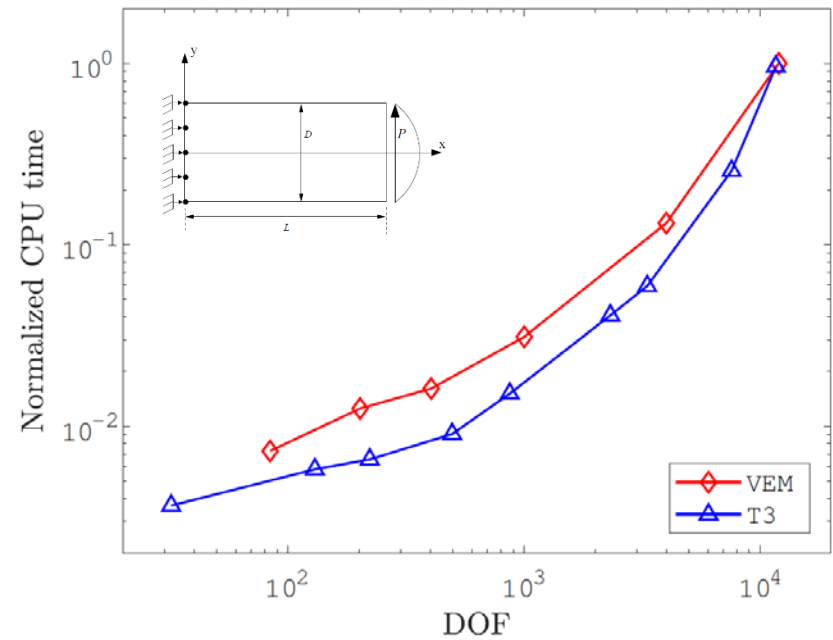
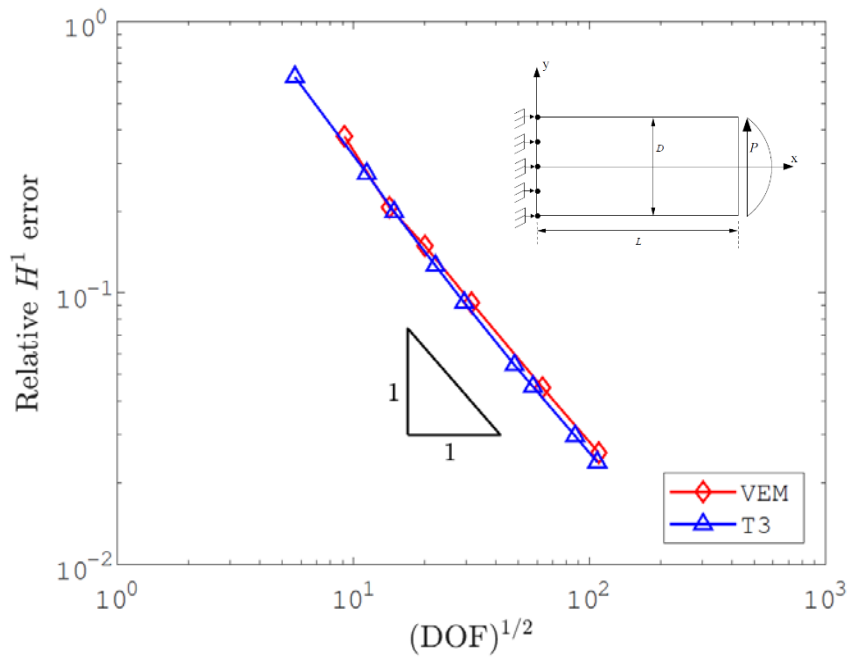
Postproceso en MATLAB



Postproceso en GiD



Convergencia





También disponible:

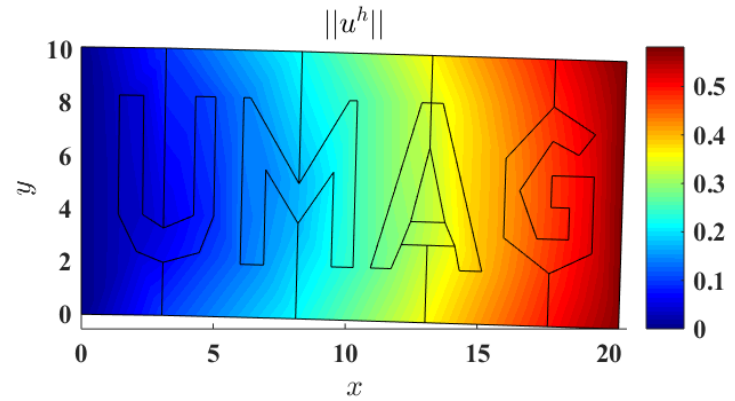
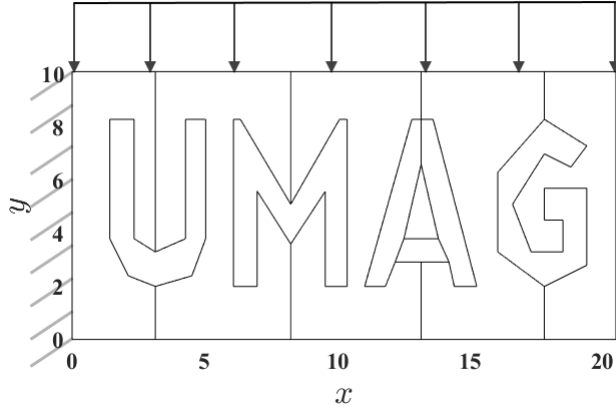
Veamy: librería para elementos virtuales escrita en C++

- Web: <http://camlab.cl/veamy>
- Repositorio: <https://github.com/cemcen/veamy>

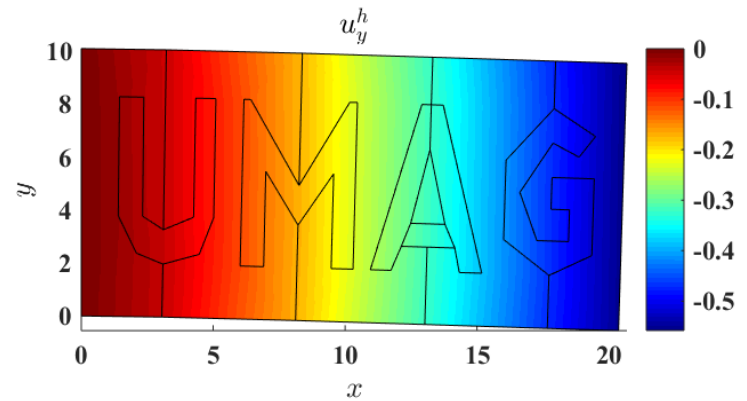
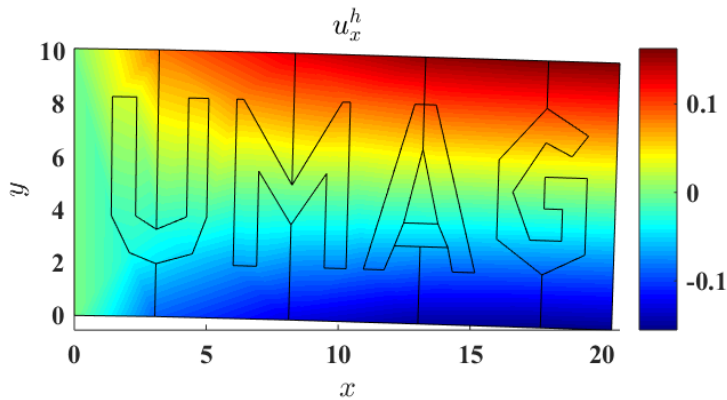
Próximamente...

VEMLia: librería para elementos virtuales escrita en Julia

AGRADECIMIENTOS: **CONICYT/FONDECYT** (Proyecto N° 1181192)



¡Gracias UMAG!



(12 elementos poligonales irregulares)