



TAREA 3 ASPECTOS COMPUTACIONALES DEL MÉTODO DE ELEMENTOS FINITOS (525538)

NICOLÁS NÚÑEZ LIRA

ÍNDICE

1. Ecuación de Elasticidad	1
1.1. Formulación Primal	1
1.2. Formulación Mixta	1
2. Formulación Variacional	2
2.1. Existencia y unicidad de la formulación variacional continua	3
3. Formulación discreta	3
4. Formulación matricial	4
4.1. Matriz de rigidez elemental	5
4.2. Vector de carga elemental	5
5. Implementación	6
6. Resultados numéricos	12
6.1. Testeo	12
Referencias	13

1. ECUACIÓN DE ELASTICIDAD

1.1. Formulación Primal. Consideremos el siguiente problema de valores de contorno:

Hallar $\mathbf{u} : \Omega \rightarrow \mathbb{R}^3$ tal que:

$$(P) \begin{cases} -\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) &= \mathbf{f} & \text{en } \Omega \\ \mathbf{u} &= \mathbf{u}_D & \text{en } \Gamma_D \\ \boldsymbol{\sigma}(\mathbf{u})\mathbf{n} &= \mathbf{g} & \text{en } \Gamma_N \end{cases}$$

donde $\Omega \subset \mathbb{R}^3$ es un subconjunto abierto, acotado, conexo, de frontera poliédrica. El tensor de esfuerzo está dado por

$$\boldsymbol{\sigma}(\mathbf{u}) = 2\mu\boldsymbol{\varepsilon}(\mathbf{u}) + \lambda(\nabla \cdot \mathbf{u})\mathbf{I}$$

con λ y μ los coeficientes de Lamé del material que compone Ω , $\mathbf{f} \in L^2(\Omega)^3$, $\mathbf{g} \in L^2(\Gamma)^3$ y \mathbf{u}_D suficientemente regular.

1.2. Formulación Mixta. Se considerará la formulación variacional mixta desplazamiento-presión. Para esto consideremos

$$p = -\lambda \nabla \cdot \mathbf{u}$$

De donde, se sigue que el problema de valores de contorno a resolver es:

Hallar $\mathbf{u} : \Omega \rightarrow \mathbb{R}^3, p : \Omega \rightarrow \mathbb{R}$ tal que:

$$(P)' \begin{cases} -2\mu \nabla \cdot \varepsilon(\mathbf{u}) + \nabla p &= \frac{\mathbf{f}}{2\mu} & \text{en } \Omega \\ p + \lambda(\nabla \cdot \mathbf{u}) &= 0 & \text{en } \Omega \\ \mathbf{u} &= \mathbf{u}_D & \text{en } \Gamma_D \\ \boldsymbol{\sigma}(\mathbf{u}, p)\mathbf{n} &= \mathbf{g} & \text{en } \Gamma_N \end{cases}$$

2. FORMULACIÓN VARIACIONAL

Multiplicando la primera ecuación de $(P)'$ por $\mathbf{v} : \Omega \rightarrow \mathbb{R}^3$ suficientemente suave en Ω , tal que $v|_{\Gamma_D} = 0$, e integrando sobre Ω , se tiene que

$$\begin{aligned} \int_{\Omega} \mathbf{f} \cdot \mathbf{v} &= -2\mu \int_{\Omega} (\nabla \cdot \mathbf{u}) \cdot v + \int_{\Omega} \nabla p \cdot \mathbf{v} \\ &= 2\mu \left(\int_{\Omega} \varepsilon(\mathbf{u}) : \nabla \mathbf{v} - \int_{\Gamma_N} (\varepsilon(\mathbf{u})\mathbf{n}) \cdot \mathbf{v} \right) - \int_{\Omega} p(\nabla \cdot \mathbf{v}) + \int_{\Gamma_N} p\mathbf{v} \cdot \mathbf{n} \\ &= 2\mu \int_{\Omega} \varepsilon(\mathbf{u}) : \nabla \mathbf{v} - \int_{\Omega} p(\nabla \cdot \mathbf{v}) - \int_{\Gamma_N} (2\mu\varepsilon(\mathbf{u}) - p\mathbf{I})\mathbf{n} \cdot \mathbf{v} \end{aligned}$$

Aplicando la condición de Neumann y reordenando la ecuación se obtiene que

$$2\mu \int_{\Omega} \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) - \int_{\Omega} p(\nabla \cdot \mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} + \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{v}$$

De forma análoga, considerando $q : \Omega \rightarrow \mathbb{R}$ suficientemente suave, multiplicando dicha función en la segunda ecuación, e integrando sobre Ω , se sigue que

$$\int_{\Omega} (\nabla \cdot \mathbf{u})q + \frac{1}{\lambda} \int_{\Omega} pq = 0$$

Consideremos los espacios $H := [H_{\Gamma_D}^1(\Omega)]^3$ y $Q := L^2(\Omega)$. Si además, consideramos el hecho de que $\mathbf{u}_D \in [H^{\frac{1}{2}}(\Gamma_D)]^3$, entonces existe $\hat{\mathbf{u}}_D \in H^1(\Omega)$ tal que $\hat{\mathbf{u}}_D|_{\Gamma_D} = \mathbf{u}_D$, entonces podemos considerar el cambio de variable $\mathbf{w} = \mathbf{u} - \hat{\mathbf{u}}_D \in H$. De esta forma es posible escribir la formulación variacional mixta del problema $(P)'$, como sigue:

Hallar $(\mathbf{w}, p) \in H \times Q$ tal que:

$$(FV) \begin{cases} a(\mathbf{w}, \mathbf{v}) + b(\mathbf{v}, p) &= F(\mathbf{v}) & \forall \mathbf{v} \in H \\ b(\mathbf{w}, q) - c(p, q) &= G(q) & \forall q \in Q \end{cases}$$

Donde las formas bilineales $a : H \times H \rightarrow \mathbb{R}$, $b : H \times Q \rightarrow \mathbb{R}$, $c : Q \times Q \rightarrow \mathbb{R}$ están definidas por:

$$a(\mathbf{u}, \mathbf{v}) = 2\mu \int_{\Omega} \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v})$$

$$b(\mathbf{u}, p) = - \int_{\Omega} (\nabla \cdot \mathbf{u})p$$

$$c(p, q) = \epsilon \int_{\Omega} pq$$

donde $\epsilon := \frac{1}{\lambda}$.

Y los funcionales $F \in H'$ y $G \in Q'$ se definen como

$$F(\mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} + \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{v} - 2\mu \int_{\Omega} \varepsilon(\mathbf{v}) : \varepsilon(\hat{\mathbf{u}}_D)$$

$$G(q) = - \int_{\Omega} (\nabla \cdot \hat{\mathbf{u}}_D)q$$

2.1. Existencia y unicidad de la formulación variacional continua. Para asegurar la existencia y unicidad de (F) se procederá comprobando las hipótesis de una generalización del Teorema de Babuška-Brezzi(ver [1]).

Notemos que las formas bilineales y los funcionales involucrados en la formulación variacional son acotados. En efecto, para todo $\mathbf{u}, \mathbf{v} \in H$ y $p, q \in Q$, se sigue que

$$\begin{aligned} |a(\mathbf{u}, \mathbf{v})| &\leq 2\mu \|\nabla^s \mathbf{u}\|_{0,\Omega} \|\nabla^s \mathbf{v}\|_{0,\Omega} \leq 2\mu \|\mathbf{u}\|_{1,\Omega} \|\mathbf{v}\|_{1,\Omega} \\ |b(\mathbf{u}, p)| &\leq \|\nabla \cdot \mathbf{u}\|_{0,\Omega} \|p\|_{0,\Omega} \leq 3 \|\mathbf{u}\|_{1,\Omega} \|p\|_{0,\Omega} \\ |c(p, q)| &\leq \epsilon \|p\|_{0,\Omega} \|q\|_{0,\Omega} \\ |F(\mathbf{v})| &\leq (\|\mathbf{f}\|_{0,\Omega} + C_{\Gamma_N} \|\mathbf{g}\|_{0,\Gamma_N} + 2\mu \|\hat{\mathbf{u}}_D\|_{0,\Omega}) \|\mathbf{v}\|_{1,\Omega} \\ |G(q)| &\leq \|\nabla \cdot \hat{\mathbf{u}}_D\|_{0,\Omega} \|q\|_{0,\Omega} \end{aligned}$$

donde C_{Γ_N} es la constante de la desigualdad de trazas.

Notemos además que trivialmente la forma bilineal es H -elíptica. Y que por la primera desigualdad de Korn, y la equivalencia de normas entre $|\cdot|_{1,\Omega}$ y $\|\cdot\|_{1,\Omega}$, la forma bilineal a también lo es.

Ahora, si consideramos el problema: *Hallar $\mathbf{v}_q \in \{u \in [H_0^1(\Omega)]^3 : \nabla \cdot u = 0\}^\perp$ tal que*

$$\nabla \cdot \mathbf{v}_q = q$$

Donde $q \in L_0^2$. Por el corolario 2.4 de [2], podemos asegurar que tiene solución única que cumple con la dependencia continua con constante C_{dc} . De esta forma, considerando que $[H_0^1(\Omega)]^3 \subset [H_{\Gamma_D}^1(\Omega)]^3$, se sigue que

$$\sup_{\mathbf{v} \in H \setminus \{\theta\}} \frac{|b(\mathbf{v}, q)|}{\|\mathbf{v}\|_{1,\Omega}} \geq \frac{|b(\mathbf{v}_q, q)|}{\|\mathbf{v}_q\|_{1,\Omega}} \geq \frac{\|q\|_{0,\Omega}^2}{C_{dc} \|q\|_{0,\Omega}} = C_{dc}^{-1} \|q\|_{0,\Omega}$$

Lo que demuestra la condición ínf-sup para la forma bilineal b . Demostrando así todas las hipótesis para el Teorema que nos asegura la existencia y unicidad del problema (FV) .

3. FORMULACIÓN DISCRETA

Sea \mathcal{T}_h una triangularización de $\bar{\Omega}$, además consideremos los espacios de aproximación

$$H_h := \{v \in [C(\Omega)]^3 : v|_T \in \mathbb{P}^1(T), \forall T \in \mathcal{T}_h\} \cap [H_{\Gamma_D}^1(\Omega)]^3$$

$$Q_h := \{q \in C(\Omega) : q|_T \in \mathbb{P}^1(T), \forall T \in \mathcal{T}_h\}$$

entonces es posible formular el problema a nivel discreto estabilizado(vease [3]):

Hallar $(u_h, q_h) \in H_h \times Q_h$ tal que:

$$(FD)(\forall (v, q) \in H_h \times Q_h) \quad \mathcal{B}(u_h, p_h; v, q) = \mathcal{G}(v, q)$$

donde para todo $(\mathbf{u}, p), (\mathbf{v}, q) \in H_h \times Q_h$

$$\mathcal{B}(u, p; v, q) = 2\mu \langle \varepsilon(\mathbf{u}), \varepsilon(\mathbf{v}) \rangle - \langle \nabla \cdot \mathbf{v}, p \rangle - \langle \nabla \cdot \mathbf{u}, q \rangle - \epsilon \langle p, q \rangle - \alpha \sum_{T \in \mathcal{T}_h} h_T^2 \langle \nabla p, \nabla q \rangle_T$$

$$\mathcal{G}(v, q) = \langle \mathbf{f}, \mathbf{v} \rangle + \langle \mathbf{g}, \mathbf{v} \rangle_{\Gamma_N} - 2\mu \langle \varepsilon(\hat{\mathbf{u}}_D), \varepsilon(\mathbf{v}) \rangle - \langle \nabla \cdot \hat{\mathbf{u}}_D, q \rangle - \alpha \sum_{T \in \mathcal{T}_h} h_T^2 \langle \mathbf{f}, \nabla q \rangle_T$$

donde $\alpha \in (0, C_I^k)$, con C_I^k es la mayor constante positiva de modo que se satisfaga la desigualdad inversa

$$C_I^k \sum_{T \in \mathcal{T}_h} h_T^2 \|\Delta \mathbf{v}\|_{0,T}^2 \leq \|\nabla \mathbf{v}\|_{0,\Omega}^2 \quad \forall \mathbf{v} \in H_h$$

Notemos que puesto que se están considerando espacios de aproximación continuos y $\mathbb{P}_1 - \mathbb{P}_1$, los términos de salto en el operador \mathcal{B} y las divergencias del operador \mathcal{G} se anulan. Además por el mismo argumento anterior, α puede ser cualquier número real.

4. FORMULACIÓN MATRICIAL

Consideremos \hat{T} el tetraédro de referencia como sigue

$$\hat{\mathbf{a}}_1 = \theta \quad , \quad \hat{\mathbf{a}}_2 = e_1 \quad , \quad \hat{\mathbf{a}}_3 = e_2 \quad , \quad \hat{\mathbf{a}}_4 = e_3$$

y $[P], [DP]$ la base de Q_h sobre \hat{T} y sus derivadas, respectivamente, como sigue

$$[P]_{1 \times 4} = [1 - x - y - z, x, y, z]$$

$$[DP]_{3 \times 4} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

La base ocupada para H_h y sus derivadas está dada por

$$[\mathbf{P}]_{3 \times 12} = \begin{bmatrix} [P] & & \\ & [P] & \\ & & [P] \end{bmatrix}$$

$$[\mathbf{DP}]_{9 \times 12} = \begin{bmatrix} [DP] & & \\ & [DP] & \\ & & [DP] \end{bmatrix}$$

De esta forma podemos expresar las funciones \mathbf{u} y p restringidas a \hat{T} como

$$\mathbf{u} = [\mathbf{P}]_{3 \times 12} [\mathbf{u}]_{12 \times 1} \quad , \quad \nabla \mathbf{u} = [\mathbf{DP}]_{9 \times 12} [\mathbf{u}]_{12 \times 1}$$

$$p = [P]_{1 \times 4} [p]_{4 \times 1} \quad , \quad \nabla p = [DP]_{3 \times 4} [p]_{4 \times 1}$$

Donde

$$[p] = [p(\hat{\mathbf{a}}_1), p(\hat{\mathbf{a}}_2), p(\hat{\mathbf{a}}_3), p(\hat{\mathbf{a}}_4)]^T$$

$$[\mathbf{u}] = [\mathbf{u}_1(\hat{\mathbf{a}}_1), \mathbf{u}_1(\hat{\mathbf{a}}_2), \mathbf{u}_1(\hat{\mathbf{a}}_3), \mathbf{u}_1(\hat{\mathbf{a}}_4), \mathbf{u}_2(\hat{\mathbf{a}}_1), \mathbf{u}_2(\hat{\mathbf{a}}_2), \mathbf{u}_2(\hat{\mathbf{a}}_3), \mathbf{u}_2(\hat{\mathbf{a}}_4), \mathbf{u}_3(\hat{\mathbf{a}}_1), \mathbf{u}_3(\hat{\mathbf{a}}_2), \mathbf{u}_3(\hat{\mathbf{a}}_3), \mathbf{u}_3(\hat{\mathbf{a}}_4)]^T$$

Se considera además la transformación lineal afín invertible $\mathcal{F} : \hat{T} \rightarrow T$ para todo $T \in \mathcal{T}_h$, tal que $\mathcal{F}(\hat{\mathbf{a}}_i) = \mathbf{a}_i$, donde \mathbf{a}_i son los vértices de T , de la forma

$$\mathcal{F}(\hat{\mathbf{x}}) = B\hat{\mathbf{x}} + b = \mathbf{x}$$

Así por regla de la cadena se sigue que

$$[DP'] = B^{-T}[DP] \quad [\mathbf{DP}'] = [\mathbf{B}]^{-T}[\mathbf{DP}]$$

donde $[DP']$ y $[\mathbf{DP}']$ son las bases de las derivadas de los polinomios en Q_h y H_h , respectivamente. y

$$[\mathbf{B}]_{9 \times 9} = \begin{bmatrix} [B] & & \\ & [B] & \\ & & [B] \end{bmatrix}$$

donde, en nuestro caso

$$[B]_{3 \times 3} = [\mathbf{a}_2 - \mathbf{a}_1, \mathbf{a}_3 - \mathbf{a}_1, \mathbf{a}_4 - \mathbf{a}_1] \quad , \quad [b]_{3 \times 1} = [\mathbf{a}_1] \quad , \quad J_{\mathcal{F}} = \det([B])$$

Además, notemos que por definición

$$\varepsilon(\mathbf{u}) = \nabla^s \mathbf{u} = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T)$$

Lo cual puede ser escrito en forma matricial como sigue

$$\varepsilon(\mathbf{u}) = \begin{bmatrix} \varepsilon_{1,1} \\ \varepsilon_{2,2} \\ \varepsilon_{3,3} \\ 2\varepsilon_{1,2} \\ 2\varepsilon_{1,3} \\ 2\varepsilon_{2,3} \end{bmatrix} = \begin{bmatrix} \partial_1 \mathbf{u}_1 \\ \partial_2 \mathbf{u}_2 \\ \partial_3 \mathbf{u}_3 \\ \partial_1 \mathbf{u}_2 + \partial_2 \mathbf{u}_1 \\ \partial_1 \mathbf{u}_3 + \partial_3 \mathbf{u}_1 \\ \partial_2 \mathbf{u}_3 + \partial_3 \mathbf{u}_2 \end{bmatrix} = [D][\mathbf{DP}][\mathbf{u}]$$

donde, por construcción, se tiene

$$[D]_{6 \times 9} = [e_1 \mid e_5 \mid e_9 \mid e_2 + e_4 \mid e_3 + e_7 \mid e_6 + e_8]^T$$

4.1. Matriz de rigidez elemental. Con lo descrito anteriormente, es posible escribir todos los productos interiores presentes en \mathcal{B} restringidos a cada elemento T , son posible escribirlos como se muestra a continuación.

$$\begin{aligned}
\langle \varepsilon(\mathbf{u}), \varepsilon(\mathbf{v}) \rangle_T &= [\mathbf{v}]^T \underbrace{\left(\int_{\hat{T}} [\mathbf{D}\mathbf{P}]^T [\mathbf{B}]^{-1} [\mathbf{D}]^T [\mathbf{D}] [\mathbf{B}]^{-T} [\mathbf{D}\mathbf{P}] J_{\mathcal{F}} \right)}_{\mathbf{K}_1} [\mathbf{u}]^T \\
\langle \nabla \cdot \mathbf{v}, p \rangle_T &= [\mathbf{v}]^T \underbrace{\left(\int_{\hat{T}} [\mathbf{D}\mathbf{P}]^T [\mathbf{B}]^{-1} [z]^T [\mathbf{P}] J_{\mathcal{F}} \right)}_{\mathbf{K}_2} [p]^T \\
\langle \nabla \cdot \mathbf{u}, q \rangle_T &= [q]^T \underbrace{\left(\int_{\hat{T}} [\mathbf{P}]^T [z] [\mathbf{B}]^{-T} [\mathbf{D}\mathbf{P}] J_{\mathcal{F}} \right)}_{\mathbf{K}_3} [\mathbf{u}]^T \\
\langle p, q \rangle_T &= [q]^T \underbrace{\left(\int_{\hat{T}} [\mathbf{P}]^T [\mathbf{P}] J_{\mathcal{F}} \right)}_{\mathbf{K}_4} [p]^T \\
\langle \nabla p, \nabla q \rangle_T &= [q]^T \underbrace{\left(\int_{\hat{T}} [\mathbf{D}\mathbf{P}]^T [\mathbf{B}]^{-1} [\mathbf{B}]^{-T} [\mathbf{D}\mathbf{P}] J_{\mathcal{F}} \right)}_{\mathbf{K}_5} [p]
\end{aligned}$$

De esta forma podemos ensamblar localmente, de lo que se obtiene que para el elemento T , las matrices de rigidez local será

$$[\mathbf{K}_T]_{16 \times 16} = \left[\begin{array}{c|c} 2\mu \mathbf{K}_1 & -\mathbf{K}_2 \\ \hline -\mathbf{K}_3 & -\epsilon \mathbf{K}_4 - \alpha h_T^2 \mathbf{K}_5 \end{array} \right]$$

4.2. Vector de carga elemental. Análogamente, es posible escribir todos los productos interiores presentes en \mathcal{B} restringidos a cada elemento T , de lo que se sigue que

$$\begin{aligned}
\langle \mathbf{f}, \mathbf{v} \rangle_T &\approx [\mathbf{v}]^T \underbrace{\left(\int_{\hat{T}} [\mathbf{P}]^T [\mathbf{P}] J_{\mathcal{F}} \right)}_{\mathbf{L}_1} [\mathbf{f}] \\
\langle \varepsilon(\hat{\mathbf{u}}_D), \varepsilon(\mathbf{v}) \rangle_T &\approx [\mathbf{v}]^T \underbrace{\left(\int_{\hat{T}} [\mathbf{D}\mathbf{P}]^T [\mathbf{B}]^{-1} [\mathbf{D}]^T [\mathbf{D}] [\mathbf{B}]^{-T} [\mathbf{D}\mathbf{P}] J_{\mathcal{F}} \right)}_{\mathbf{L}_3} [\hat{\mathbf{u}}_D] \\
\langle \nabla \cdot \hat{\mathbf{u}}_D, q \rangle_T &\approx [q]^T \underbrace{\left(\int_{\hat{T}} [\mathbf{P}]^T [z] [\mathbf{B}]^{-T} [\mathbf{D}\mathbf{P}] J_{\mathcal{F}} \right)}_{\mathbf{L}_4} [\hat{\mathbf{u}}_D] \\
\langle \mathbf{f}, \nabla q \rangle_T &\approx [q]^T \underbrace{\left(\int_{\hat{T}} [\mathbf{D}\mathbf{P}]^T [\mathbf{B}]^{-1} [\mathbf{P}] J_{\mathcal{F}} \right)}_{\mathbf{L}_5} [\mathbf{f}]
\end{aligned}$$

Para el producto interior en el sentido de trazas, consideremos un elemento T , tal que posea una cara en Γ_N , con vértices $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4$. De esta forma se puede parametrizar la cara exterior en función del triángulo de referencia $\hat{t} \in \mathbb{R}^2$, dependiendo del nodo interior, la cual viene dada por

$$r(s, t) = \begin{cases} s(\mathbf{a}_2 - \mathbf{a}_4) + t(\mathbf{a}_3 - \mathbf{a}_4) + \mathbf{a}_4 & \text{si } i = 1 \\ \mathbf{a}_1 + s(\mathbf{a}_3 - \mathbf{a}_1) + t(\mathbf{a}_4 - \mathbf{a}_1) & \text{si } i = 2 \\ \mathbf{a}_1 + s(\mathbf{a}_2 - \mathbf{a}_1) + t(\mathbf{a}_4 - \mathbf{a}_1) & \text{si } i = 3 \\ \mathbf{a}_1 + s(\mathbf{a}_2 - \mathbf{a}_1) + t(\mathbf{a}_3 - \mathbf{a}_1) & \text{si } i = 4 \end{cases}$$

Así, sin pérdida de generalidad, para todo i , se sigue que

$$\begin{aligned}\langle \mathbf{g}, \mathbf{v} \rangle_{T \cap \Gamma_N} &\approx [\mathbf{v}]^T \left(\int_{T \cap \Gamma_N} [\mathbf{P}]^T [\mathbf{P}] \right) [\mathbf{g}] \\ &= [\mathbf{v}]^T \underbrace{\left(\int_t [\mathbf{P} \circ \mathbf{r}]^T [\mathbf{P} \circ \mathbf{r}] \|\mathbf{r}_s \times \mathbf{r}_t\|_2 \right)}_{L_2} [\mathbf{g}]\end{aligned}$$

De esta forma podemos ensamblar localmente, de lo que se obtiene que para el elemento T , el vector de carga local estará dado por

$$[\mathbf{L}_T]_{16 \times 1} = \begin{bmatrix} \mathbf{L}_1 + \mathbf{L}_2 - 2\mu \mathbf{L}_3 \\ -\mathbf{L}_4 - \alpha h_T^2 \mathbf{L}_5 \end{bmatrix}$$

5. IMPLEMENTACIÓN

La implementación de la formulación matricial fue realizada en entorno MATLAB, es fácil ver que en la sección de *Datos del problema* es posible modificar los parámetros del problema de valores de contorno que se desee resolver con el método estabilizado anteriormente planteado.

El script `linearelasticity_pressure.m`, listado a continuación, está programado con las suposiciones de que el programa `tetgen` está en el mismo directorio, con el cual se crea el mallado de un archivo con extensión `.poly` a elegir. Además abre directamente la visualización de la solución, exportada a formato `.vtk` en el programa `paraview`, el cual también se presupone instalado.

LISTING 1. Elasticidad lineal

```

1  % Resolucion del problema de valores de contorno de elasticidad lineal
2  % / \nabla \cdot \sigma(u) = -f en \Omega
3  % < u = u_D en \Gamma_D
4  % \ \sigma \cdot n = g en \Gamma_N
5  % En su formulacion mixta desplazamiento lineal-presion con Mef
   estabilizado P1-P1
6
7  clear all; close all; clc
8  %% Configuracion
9  d=3; %Dimension
10 Neu=[1]; %Identificador Neumann
11 Dir=[2]; %Identificador Dirichlet
12
13 %% Datos del problema
14
15 % Calculo de los coeficientes de Lamé:
16 % (el material que se considerara sera el acero estructural
17 % Modulo elastico (E) y modulo de poisson (nu)
18 E=2.1e5; nu=0.26;
19 % Coeficientes de Lamé (lambda, mu)
20 lambda=(E*nu)/((1+nu)*(1-2*nu));
21 mu=E/(2*(1+nu));
22 eps=1/lambda;
23
24 %Parametro estabilizacion
25 alf=1;
26
27 % Funciones involucradas
28 f = @(x,y,z) [(x-x);(y-y);-9.8*x.^0.*y.^0.*z.^0];
29 ud = @(x,y,z) [(x-x);(y-y);(z-z)];
30 g = {@(x,y,z) [(x-x);(y-y);(z-z)], @(x,y,z) [(x-x);(y-y);1000*x.^0.*y.^0.*z.^0]};

```

```

31
32 %% Creacion del mallado
33 [archivo,ruta]=uigetfile('*.poly','Seleccione archivo de malla');%
    Lectura archivo de malla
34 name=archivo(1:end-5);%Rescatar el nombre del .poly
35
36 mmax=.01;mmax=num2str(mmax);%Medida maxima como string
37 c=strcat('./tetgen -pqa',mmax,{' '},archivo);%Comando para crear mallado
38 system(c{1});%Ejecucion comando anterior
39
40 fid=fopen(strcat(ruta,[name, '.1.node']), 'r');%Lectura nodos
41 nid = fscanf(fid, '%i',4);
42 mesh.p = fscanf(fid, '%f', [nid(3)+(nid(4)+1+d) nid(1)]);
43 mesh.p = (mesh.p(2:end,:))';
44 fclose(fid);
45
46 fid=fopen(strcat(ruta,[name, '.1.ele']), 'r');%Lectura elementos
47 nele = fscanf(fid, '%i',3);
48 mesh.t = fscanf(fid, '%f', [nele(3)+(2+d) nele(1)]);
49 mesh.t = (mesh.t(2:end,:))';
50 fclose(fid);
51
52 fid=fopen(strcat(ruta,[name, '.1.face']), 'r');%Lectura caras
53 nface = fscanf(fid, '%i',2);
54 mesh.f = fscanf(fid, '%f', [1+d+nface(2) nface(1)]);
55 mesh.f = (mesh.f(2:end,:))';
56 fclose(fid);
57
58 %% Base de espacio de funciones discreto tetraedro de referencia
59 %Base polinomios v=[P]*[v]
60 p=@(x,y,z) [1-x-y-z,x,y,z];
61 P=@(x,y,z) blkdiag(p(x,y,z),p(x,y,z),p(x,y,z));
62 %Base gradiente \nabla u=[DP]*[u]
63 dp=[-1 -1 -1;1 0 0;0 1 0;0 0 1]';
64 DP=blkdiag(dp,dp,dp);
65 z=[1,0,0,0,1,0,0,0,1];%Divergencia \nabla\cdot u=[z]*[DP]*[u]
66 %Base tensorial \varepsilon(u)=[D]*[DP]*[u]
67 D=[1 0 0 0 0 0 0 0 0; %\partial_1 u_1
68     0 0 0 0 1 0 0 0 0; %\partial_2 u_2
69     0 0 0 0 0 0 0 0 1; %\partial_3 u_3
70     0 1 0 1 0 0 0 0 0; %2*(\partial_2 u_1 + \partial_1 u_2)
71     0 0 1 0 0 0 1 0 0; %2*(\partial_3 u_1 + \partial_1 u_3)
72     0 0 0 0 0 1 0 1 0]; %2*(\partial_3 u_2 + \partial_2 u_3)
73
74 %% Ensamble global
75 nnodos=size(mesh.p,1);
76 nelem=size(mesh.t,1);
77 K=sparse((d+1)*nnodos,(d+1)*nnodos);
78 F=sparse((d+1)*nnodos,1);
79 I=speye(9);
80
81 % Matrices de masa
82 % \int P'*P
83 m=@(x,y,z) P(x,y,z)'*P(x,y,z);
84 M=gauss(m,d); %[M]_{9\times 9}
85 % \int p'*p

```

```

86 m=@(x,y,z)p(x,y,z)'*p(x,y,z);
87 m=gauss(m,d); %[m]_{3\times 3}
88 % \int p
89 pp=gauss(p,3);
90 % \int P
91 PP=gauss(P,d);
92
93 % Ensamble global
94 for i=1:nelem
95     %Identificar elemento i
96     ind=mesh.t(i,:);
97     xx=(mesh.p(ind,1:d))';
98     %Cambio de variable
99     b=xx(:,2:end)-xx(:,1);
100     Jf=abs(det(b));
101     B=blkdiag(b,b,b);
102     bb=xx(:,1);
103     %Diametro elemento i
104     h=0;
105     for j=1:d+1
106         for k=2:d+1
107             if norm(xx(:,j)-xx(:,k))>h
108                 h=norm(xx(:,j)-xx(:,k));
109             end
110         end
111     end
112     h=h^2;
113
114     %% Matriz de rigidez
115
116     %Matriz \int \varepsilon(u):\varepsilon(v)
117     K1=2*mu*(Jf/6)*DP'*(B\I)*D'*D*(B'\DP); %V01(elem referencia)=1/6;
118     % Matriz \int (\nabla\cdot v)p
119     K2=-Jf*DP'*(B\I)*z'*pp;
120     % Matriz \int (\nabla\cdot u)q
121     K3=-Jf*pp'*z*(B'\DP);
122     % Matriz \int pq
123     K4=-eps*Jf*m;
124     % Matriz \int \nabla p \nabla q ---- Termino estabilizacion
125     K5= -alf*h*(Jf/6)*dp'*(b\I(1:3,1:3))*(b'\dp);
126     %Ensamble K
127     Ind=[mesh.t(i,:),mesh.t(i,)+nnodos,mesh.t(i,)+2*nnodos,mesh.t(i,)+3*nnodos];
128     K(Ind,Ind)=K(Ind,Ind)+[K1,K2;K3,K4+K5];
129
130     %% Vector de carga
131
132     %Vector \int f\cdot v
133     Ind=Ind(1:end-4);
134     F(Ind)=F(Ind)+Jf*M*f(xx(1,:) ',xx(2,:) ',xx(3,:) ');
135     %Vector \int \varepsilon(u_d):\varepsilon(v)
136     F(Ind)=F(Ind)-2*mu*(Jf/6)*DP'*(B\I)*D'*D*(B'\DP)*ud(xx(1,:) ',xx(2,:) ',xx(3,:) ');
137     %Vector \int q(\nabla\cdot u_D)
138     ind=mesh.t(i,)+3*nnodos;
139     F(ind)=F(ind)-Jf*pp'*z*(B'\DP)*ud(xx(1,:) ',xx(2,:) ',xx(3,:) ');

```



```

140     %Vector \int f\cdot (\nabla p) ----- Termino de estabilizacion
141     F(ind)=F(ind)-alf*h*dp'*(b\I(1:3,1:3))*PP*f(xx(1,:) ',xx(2,:) ',xx
        (3,:) ');
142 end
143
144 %% Imposicion de condiciones de frontera
145
146 %Condiciones Dirichlet
147 u=zeros((d+1)*nnodos,1);
148 nodirich=1:(d+1)*nnodos;
149 for i=1:length(Dir)
150     dirich=unique(mesh.f(mesh.f(:,end) == Dir(i),1:3));
151     Dirich=[dirich ; dirich+nnodos ; dirich+2*nnodos];
152     u(Dirich)=ud(mesh.p(dirich,1),mesh.p(dirich,2),mesh.p(dirich,3));
153     nodirich=setdiff(nodirich,Dirich);
154 end
155
156 %Condiciones de Neumann
157 for i=1:length(Neu)
158     gn=g{i};
159     neu=mesh.f(find(mesh.f(:,end)==Neu(i)),1:3);
160     for j=1:length(neu)
161         ind=find(sum((ismember(mesh.t,neu(j,:)))')==3); %Indice elemento
            de la cara actual
162         ind2=find(ismember(mesh.t(ind,:),neu(j,:)));
163         ind3=mesh.t(ind,ind2); %Indices nodo de la cara actual
164         ind4=find(~ismember(mesh.t(ind,:),neu(j,:))); %Nodo interior
            elemento actual
165         %Nodos cara actual
166         a=mesh.p(ind3,1:end-1)';
167         %Vector normal cara actual
168         n=norm(cross(a(:,2)-a(:,1),a(:,3)-a(:,1)));
169         %Parametrizacion
170         switch ind4 %Composicion [P\circ r]
171             case 1
172                 R=@(s,t) P(s,t,1-s-t)'*P(s,t,1-s-t);
173                 R=gauss(R,d-1);
174             case 2
175                 R=@(s,t) P(0,s,t)'*P(0,s,t);
176                 R=gauss(R,d-1);
177             case 3
178                 R=@(s,t) P(s,0,t)'*P(s,0,t);
179                 R=gauss(R,d-1);
180             case 4
181                 R=@(s,t) P(s,t,0)'*P(s,t,0);
182                 R=gauss(R,d-1);
183         end
184         %Nodos elemento actual
185         w=(mesh.p(mesh.t(ind,:),1:d))';
186         %Interpolacion g en elemento actual
187         G=gn(w(1,:) ',w(2,:) ',w(3,:) ');
188         %Producto interior de traza local:
189         %\int_{\Gamma_N\cap T} g\cdot v
190         F2=n*R*G;
191         %Ensamble vector de carga

```

```

192         Ind=[mesh.t(ind,:) , mesh.t(ind,:)+nnodos , mesh.t(ind,:)+2*
193             nnodos];
194         F(Ind)=F(Ind)+F2;
195     end
196 end
197 %% Resolucion sistema
198
199 u(nodirich) = K(nodirich,nodirich)\(F(nodirich)-K(nodirich,Dirich)*u(
200     Dirich));
201 %% Exportar VTK y visualizacion
202
203 %Parte vectorial---Defleccion
204 sol.type='vector';
205 sol.name='Deflection';
206 sol.data=[u(1:nnodos),u(nnodos+1:2*nnodos),u(2*nnodos+1:3*nnodos)];
207 %Parte escalar---Presion
208 sol(2).type='scalar';
209 sol(2).name='Pressure';
210 sol(2).data=u(3*nnodos+1:end);
211
212 name=strcat(name, '.vtk');
213 vtk_write_tetrahedral_grid_and_data(name, 'Linear_elasticity', ...
214     mesh.p(:,1:end-1), mesh.t, sol, false)
215
216 %Visualizar en paraview
217 c=strcat('paraview --data=', name, '');
218 system(c);

```

Se ocupa además una cuadratura de Gauss, dependiendo de la dimensión de la región de integración, la cual fue programada en el script `gauss` listado a continuación

LISTING 2. Cuadratura de Gauss 2D y 3D

```

1 function int = gauss(f,d,B,b)
2 %Aproximacion de integrales con cuadratura de Gauss en 2D y 3D, en
3   triangulos y tetraedros respectivamente,
4   considerando un cambio de variable lineal, afin e invertible de la
5   forma
6   %  $F:\hat{K} \rightarrow K$ 
7   %  $F(\hat{x})=B*\hat{x}=x$ 
8   % donde  $\hat{K}$  es el triangulo o el tetraedro de referencia con nodos
9      $\theta, e_i, i=1:d$ 
10  % Parametros de entrada:
11  % f:Integrando
12  % d:Dimension del espacio
13  % B,b:Constantes de F
14  %-----
15  %Si nargin=2, se considera que f solo se debe evaluar  $\hat{K}$ .
16  %Si nargin=4, se considera que f depende de vectores en  $\hat{K}$  y K
17  if nargin==2
18      switch d
19          case 2
20              ww = [1/6, 1/6, 1/6];
21              xx = {2/3,1/6;1/6,1/6;1/6,2/3};
22              int = 0;

```

```

20         if nargin==2
21             for j = 1:d+1
22                 int = int + ww(j)*f(xx{j,:});
23             end
24         end
25         if nargin==4
26             for j = 1:d+1
27                 int = int + ww(j)*f(xx{j,:),[xx{j,:}]*B'+b);
28             end
29         end
30     case 3
31         ww = [1/24,1/24,1/24,1/24];
32         xx = {0.5854101966249685    0.1381966011250105
33              0.1381966011250105;
34              0.1381966011250105    0.1381966011250105    0.1381966011250105;
35              0.1381966011250105    0.1381966011250105    0.5854101966249685;
36              0.1381966011250105    0.5854101966249685    0.1381966011250105};
37         int = 0;
38         if nargin==2
39             for j = 1:d+1
40                 int = int + ww(j)*f(xx{j,:});
41             end
42         if nargin==4
43             for j = 1:d+1
44                 int = int + ww(j)*f(xx{j,:),[xx{j,:}]*B'+b);
45             end
46         end
47     end
48 end

```

Además para el calculo de los errores $\| \cdot \|_0$, $\| \cdot \|_1$ y $\| \cdot \|_1$, es posible recurrir al script `err.m`, listado a continuación

LISTING 3. Error en espacios de Sobolev de orden 0 y 1

```

1 function [L2,H1,sH1]=err(f,df,p,t,uu,d)
2 %Calculo de los errores en normas L^2 y H^1 y seminorma H^1 dados los
   datos de ingreso:
3 % f:Funcion solucion
4 % df:Gradiente de la solucion
5 % p:Nodos del mallado generado por Triangle/Tengen
6 % t:Elementos del mallado generado por Triangle/Tengen
7 % uu:Solucion numerica
8 % d:Dimension del espacio(d=2,3)
9 L2=0;sH1=0;
10 for i=1:size(t,1)
11     switch d
12         case 2
13             P=@(x,y)[1-x-y,x,y];
14             DP=[-1 1 0;-1 0 1];
15             ind=t(i,:);
16             xx=p(ind,1:2)';
17             B=xx(:,2:end)-xx(:,1);
18             Jf=abs(det(B));
19             b=xx(:,1);
20             l=@(x,y,w) (f(w(1),w(2))-P(x,y)*uu(ind)).^2*Jf;

```

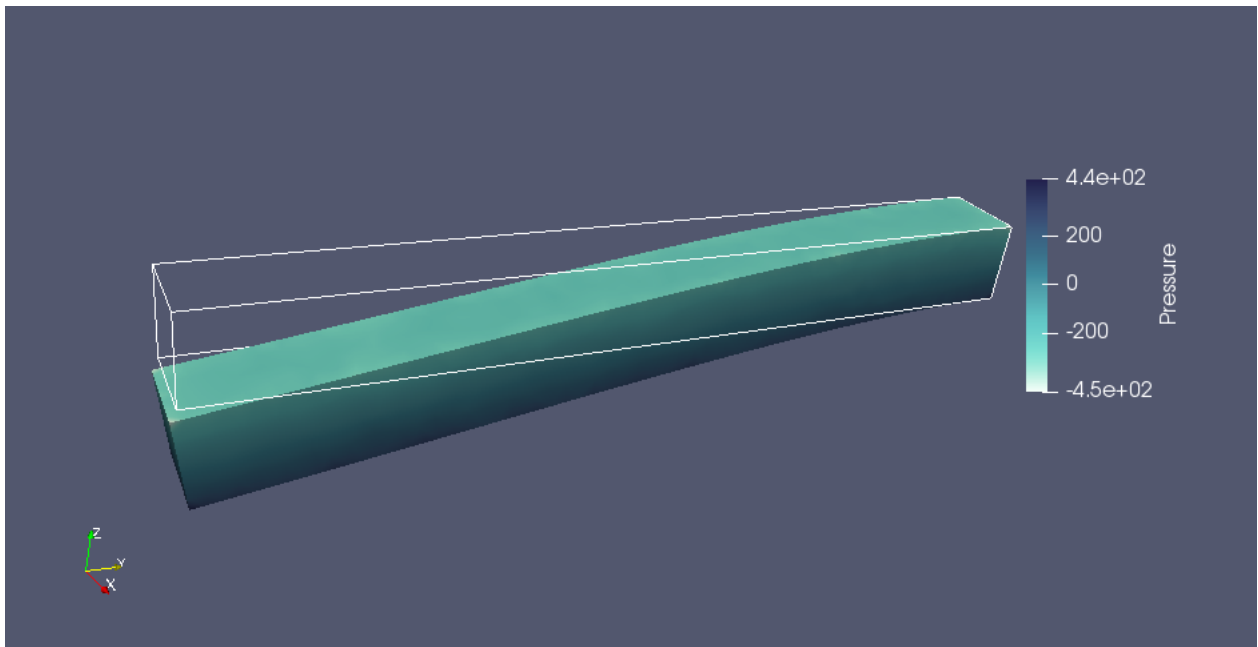
```

21         L2=L2+gauss(1,d,B,b);
22         sn=@(x,y,w) (df(w(1),w(2))-(B'\DP)*u(ind)).^2*Jf;
23         sH1=sH1+gauss(sn,d,B,b);
24     case 3
25         P=@(x,y,z) [1-x-y-z;x;y;z]';
26         DP=[-1 1 0 0;-1 0 1 0;-1 0 0 1];
27         ind=t(i,:);
28         xx=p(ind,1:3)';
29         B=xx(:,2:end)-xx(:,1);
30         Jf=abs(det(B));
31         b=xx(:,1);
32         l=@(x,y,w) (f(w(1),w(2),w(3))-P(x,y,z)*uu(ind)).^2*Jf;
33         L2=L2+gauss(1,d,B,b);
34         sn=@(x,y,w) (df(w(1),w(2),w(3))-(B'\DP)*u(ind)).^2*Jf;
35         sH1=sH1+gauss(sn,d,B,b);
36     end
37 end
38 H1=sqrt(L1+sH1);
39 sH1=sqrt(sH1);
40 L2=sqrt(L2);

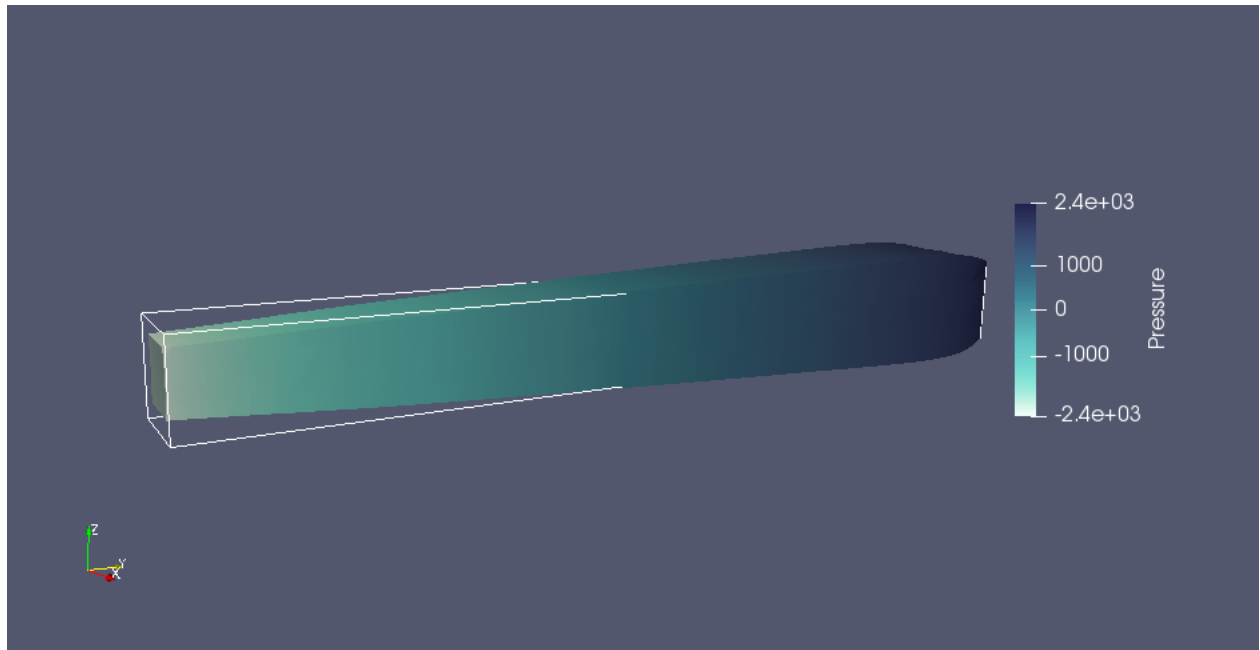
```

6. RESULTADOS NUMÉRICOS

6.1. Testeo. Se considerara la barra de acero estructural(módulo elástico $E = 2,1e5$, módulo de Poisson $\nu = 0,26$) Ω como el paralelepipedo con vertices opuesto $(0,0,0)$ y $(1,10,1)$, dado en `viga.poly`, empotrado en la cara $\{(x,y,z) \in \Omega : y = 10\}$, con condiciones Dirichlet nulas en la recta $\{(x,0,1) \in \Omega\}$, sometida a la fuerza $\mathbf{f} = (0,0,-9,8)$.



Y para las mismas condiciones, pero con $\mathbf{f} = (0,10,0)$.



Todo lo anterior con el parámetro de estabilización $\alpha = 1$.

REFERENCIAS

- [1] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods, volume 15 of Springer Series in Computational Mathematics*. Springer-Verlag, New York, 1991
- [2] L. P. Franca and R. Stenberg. *Error analysis of Galerkin least squares methods for the elasticity equations*. *SIAM J. Numer. Anal.* 1991
- [3] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations, volume 5 of Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1986. Theory and algorithms.