

DESARROLLO DE UNA HERRAMIENTA PARA CONTROLAR EL ERROR Y
MEJORAR LA PRECISIÓN DE APROXIMACIONES BASADAS EN ELEMENTOS
FINITOS CON FORMAS POLIGONALES ARBITRARIAS EN 3D.

JUAN DIEGO DUARTE FERRUCHO
JUAN SEBASTIÁN PARRA HERNÁNDEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2018

DESARROLLO DE UNA HERRAMIENTA PARA CONTROLAR EL ERROR Y
MEJORAR LA PRECISIÓN DE APROXIMACIONES BASADAS EN ELEMENTOS
FINITOS CON FORMAS POLIGONALES ARBITRARIAS EN 3D

JUAN DIEGO DUARTE FERRUCHO
JUAN SEBASTIÁN PARRA HERNÁNDEZ

Trabajo de grado para optar al título de
Ingeniero Mecánico

DIRECTOR
OCTAVIO ANDRÉS GONZÁLEZ ESTRADA
Ingeniero Mecánico, PhD.

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2018

*Agradezco a toda mi familia por su apoyo incondicional, especialmente a mis padres pues les
debo lo que soy.*

*A mis compañeros, les agradezco su compañía y a los profesores por la orientación durante
esta etapa que hoy se culmina,*

Juan Diego Duarte Ferrucho

A DIOS, por permitirme terminar esta etapa de mi vida, y entregarme todas las herramientas para afrontar cada uno de los retos superados en ésta.

*A mi madre Nubia, por su inagotable esfuerzo en forjar a sus hijos como personas de bien.
Por ese inmenso amor y apoyo, que siempre me brindo la confianza para seguir adelante.
Pero, sobre todo por creer en mí, a pesar de todas las circunstancias.*

A mi padre Ricardo, por siempre estar cuando necesitaba un consejo, o alguien a quien acudir en momentos difíciles. Por su grande deseo de cuidar y dar lo mejor a sus hijos.

A mi familia, por siempre estar pendiente de mí, orientándome y ayudándome a crecer como persona. Porque lo más importante para mí, fue sentir su apoyo y orgullo.

*A mis amigos, que siempre me apoyaron y me brindaron una mano cuando más lo necesitaba.
Ellos, que siempre me dijeron si puedes, cuando tenía el ánimo en el piso.*

Juan Sebastián Parra Hernández

CONTENIDO

	Pág.
INTRODUCCIÓN	13
1. FORMULACIÓN DEL PROBLEMA	16
1.1 PLANTEAMIENTO DEL PROBLEMA.....	16
1.2 JUSTIFICACIÓN PARA SOLUCIONAR EL PROBLEMA	16
2. OBJETIVOS.....	18
2.1 OBJETIVO GENERAL	18
2.2 OBJETIVOS ESPECÍFICOS.....	18
3. APROXIMACIÓN DE LOS ELEMENTOS FINITOS PARA MALLAS POLIGONALES	19
3.1 DISCRETIZACIÓN DE LOS ELEMENTOS POLIGONALES	19
3.2 INTEGRACIÓN NUMÉRICA EN ELEMENTOS FINITOS POLIGONALES	22
4. FORMULACIÓN DEL PROBLEMA ELÁSTICO - LINEAL	25
5. ERROR ESTIMADO EN NORMA ENERGÉTICA.....	27
5.1 RECONSTRUCCIÓN DE TENSIONES	27
6. RESULTADOS NUMÉRICOS.....	30
6.1 PLACA CUADRADA SOMETIDA A CARGA EXTERNAS	31
6.2 CILINDRO BAJO PRESIÓN INTERNA.....	35
7. CONCLUSIONES	39
REFERENCIAS	40
ANEXOS.....	45

LISTA DE TABLAS

	Pág.
Tabla 1. Norma energética y efectividad global del cilindro hueco para diferentes grados de libertad.	32
Tabla 2. Norma energética y efectividad global del cilindro hueco para diferentes grados de libertad.	36

LISTA DE FIGURAS

	Pág.
Figura 1. Elemento Voronoi 3D: (a) Teselación Voronoi para un cubo, (b) Vecinos naturales para nodo P, según la función de forma de Laplace.	20
Figura 2. Mapeo en el elemento poligonal global y su poliedro de referencia.	23
Figura 3. Transformación isoparamétrico para un triángulo.....	23
Figura 4. Cubo sometido a presión externa.	31
Figura 5. Efectividad global vs grados de libertad.....	32
Figura 6. Energía de deformación y tendencia del error vs grados de libertad.	33
Figura 7. Efectividad local a nivel del elemento.	33
Figura 8. Cilindro sometido a presión interna.....	35
Figura 9. Efectividad global vs Grados de libertad.....	36
Figura 10. Energía de deformación y tendencia del error vs grados de libertad.	37
Figura 11. Efectividades locales a nivel de elemento.	38

LISTA DE ANEXOS

	Pág.
Anexo A. Simulación de tensión para el cubo.....	52
Anexo B. Simulación de tensión para el cilindro.	54
Anexo C. Convergencia de la energía de deformación según los grados de cuadratura.....	56
Anexo D. Análisis vectorial para la obtención del vector de fuerza.....	57
Anexo E. Código de entrada para Voro++	58
Anexo F. Descripción del código de Matlab.....	63

RESUMEN

TÍTULO: DESARROLLO DE UNA HERRAMIENTA PARA CONTROLAR EL ERROR Y MEJORAR LA PRECISIÓN DE APROXIMACIONES BASADAS EN ELEMENTOS FINITOS CON FORMAS POLIGONALES ARBITRARIAS EN 3D.¹

AUTORES:

Juan Diego Duarte Ferrucho

Juan Sebastián parra Hernández²

PALABRAS CLAVES: Método de elementos finitos, estimación del error, mallas poliédricas, mallas 3D.

DESCRIPCIÓN:

En este trabajo se presenta una técnica de estimación del error de discretización de la solución para problemas elástico-lineales en tres dimensiones, resueltos mediante una aproximación de elementos finitos con formas poliédricas (polyFEM). La estimación del error está basada en la reconstrucción de la solución en tensiones mediante un procedimiento de promediado en nodos.

Se evalúan dos problemas con solución exacta: placa cuadrada sometida a cargas externas y cilindro con presión interna. Con la aproximación numérica de la solución se valida la efectividad de la estimación a partir del error obtenido. Con la inclusión de los elementos poliédricos en tres dimensiones se busca estudiar problemas más reales y simplificar el mallado.

¹ Trabajo de grado

² Facultad de Ingeniería Físico-Mecánicas. Escuela de Ingeniería Mecánica. Director Dr. Octavio Andrés González Estrada, PhD, Ingeniero Mecánico.

ABSTRACT

TITLE: DEVELOPMENT OF A TOOL TO CONTROL THE ERROR AND IMPROVE THE ACCURACY OF FINITE ELEMENTS BASED ON ARBYTRARY POLIGONAL SHAPES IN 3D.³

AUTHORS:

Juan Diego Duarte Ferrucho

Juan Sebastián parra Hernández⁴

KEYWORDS: Finite elements method, estimation of error, polyhedral mesh, 3D mesh.

DESCRIPTION:

In this paper, we present an error estimator technique of solution discretization for linear elastic problems in three dimensions is shown, solved through an approximation of finite element with polygonal shapes (polyFEM). The error estimate is based in recovery type of the stress solution by means of the node averaging procedure.

Two problems are evaluated with exact solution: a square plate subjected to external loads and a cylinder with internal pressure. With the approximation of the solution, the effectiveness of the estimation is validated based on the error obtained. The inclusion of polyhedral elements in three dimensions seeks to study more real problems and simplify meshing.

³ Bachelor thesis.

⁴ Faculty of Physico-Mechanical Engineering, School of Mechanical Engineering. Advisor: Dr. Octavio Andrés González Estrada, PhD, Mechanical Engineer.

INTRODUCCIÓN

Tradicionalmente, los elementos triangulares y cuadráticos han sido las primeras formas de elementos de estudio. Recientemente, ha surgido interés en los elementos con más de 4 vértices [1], que se destacan por su capacidad de flexibilidad de mallado, precisión de resultados y permiten manejar geometrías arbitrarias [2]. Esta técnica es útil para problemas más reales donde se facilita su modelado [3], [4], [5]. Se han desarrollado generadores de mallado para elementos poligonales que facilitan realizar la discretización de diferentes formas geométricas [6].

Los elementos finitos requieren del cálculo de funciones de interpolación para llegar a la solución, las funciones planteadas para los elementos tradicionales no se pueden usar en elementos poligonales, en [7] se introdujeron los primeros conceptos analíticos básicos, luego se estableció un tipo de coordenadas baricéntricas [8] [9], que permiten la construcción de funciones para formas poligonales, pero se limitaba a triángulos paramétricos. Después Floater [10] muestra las posibles formas de ampliar la capacidad a n-lados basados en el trabajo de Wachspress, del cual han desarrollado investigaciones recientemente, dando paso a nuevas construcciones extendidas que satisfacen la propiedad de Lagrange, denominadas como familia de suavidad de tres-coordenadas, coordenadas de Wachspress, coordenadas de valor medio, coordenadas armónicas discretas, que permiten el modelado geométrico con las mallas adecuadas logrando así nuevas implementaciones más robustas. También se han implementado nuevas cuadraturas serendipity en los elementos poligonales, que permiten un número reducido de funciones base en las mallas cuadriláteras arbitrarias, y establecen convergencia cuadrática. Con gran aplicabilidad a la familia de suavidad de tres-coordenadas, a diferencia del elemento lagrangiano pierde precisión, pero es más eficiente [11]. La aplicación serendipity en los elementos facilita la flexibilidad en el mapeo del dominio sin aumentar el costo computacional, ideal para refinados.

Se desarrollaron diversas variaciones en los modelos basados en estas estructuras poligonales, destacando la teselación de celdas Voronoi, que permiten mayor flexibilidad de discretización [12], inicialmente fueron empleados para modelar estructuras porosas en materiales con formas regulares [13]. Talischi [6] introdujo funciones de densidad en las configuraciones Voronoi, para graduar la calidad del mallado de la geometría, resultando ideal para estructuras más complicadas, materiales compuestos o porosos [14].

Otros estudios se extendieron en entornos geométricos 3D [15], donde se encuentra que la aproximación es más acertada comparada con el modelamiento en 2D [16], inicialmente se han propuesto metodologías de mallado de geometrías 3D que funcionan creando un campo de mallas random dentro de las fronteras declaradas en el dominio [17]. Una metodología planteada para dar solución a un problema con características 3D, trata en la descomposición de los estados de deformación de los elementos 3D [18], debido a que, con la energía de deformación lineal se puede lograr convergencia en la solución numérica. Para el caso donde se presentan funciones no polinómicas dentro del problema, en [19] se trabajaron dos técnicas de integración para elementos poligonales 3D XFEM, ideal para problemas con singularidades y discontinuidades en el mallado, caso común en la grietas. Las técnicas de mallado que se han desarrollado en trabajos recientes combinan los elementos de celdas Voronoi con topologías en 3D [20].

Las soluciones obtenidas por métodos numéricos inducen un error en la solución del problema, este error debe ser estimado para validar el modelo. Se han desarrollado diferentes técnicas para calcular dicho error [21], [22], donde se estima el error en norma energética empleando el concepto reconstrucción de la solución a partir de diferentes procedimientos, por ejemplo aproximación nodal, mínimos cuadrados ponderados [23], [24], entre otros. Recientemente se estudia la obtención del error a posteriori empleando los elementos finitos extendidos (XFEM), planteando una relación entre el campo de deformación bruto y mejorado con MLS, en [25], [26], se aplica en problemas de tipo fractura elástica lineal. Recientemente,

se han desarrollado técnicas que se basan en nuevas estrategias de discretización, en [27] se estima el error residual para mallas poligonales con un alto orden BEM, empleando operadores de cuasi-interpolación, y [28] para su extensión en mallas 3D la estimación de la n -error es realizada a partir de una formulación mixta equivalente. Estas técnicas proveen confiabilidad, eficiencia y calidad a una aproximación, lo que es esencial para estudios y aplicaciones prácticas posteriores como el mallado adaptativo [29] y refinamiento [30], [31], realizados en las distintas formas de elementos poligonales.

Este trabajo está organizado de la siguiente forma: en primer lugar, se presenta una introducción a los elementos finitos poligonales, incluyendo una descripción de las técnicas de construcción e integración numérica. Posteriormente, se presenta el procedimiento de reconstrucción de tensiones (este estudio se hace con un promediado de nodos), seguido de los resultados numéricos. Finalmente, se indican las conclusiones más relevantes.

1. FORMULACIÓN DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

En los últimos años se han desarrollado nuevos métodos numéricos en el campo de los elementos finitos, que permiten la construcción de mallas de elementos definidas mediante elementos poligonales con un número arbitrario de lados. El propósito fundamental es aliviar las restricciones de los algoritmos de mallado, a la vez que mejorar la precisión y flexibilidad del método numérico.

Los métodos numéricos como herramienta numérica siempre llevan un error asociado de aproximación que es necesario controlar para validar los resultados de la simulación.

1.2 JUSTIFICACIÓN PARA SOLUCIONAR EL PROBLEMA

Controlar el error es un punto esencial dentro de los métodos numéricos. Un problema desarrollado numéricamente se valida a partir de la estimación de los errores obtenidos, de esta manera se puede saber que tan exactos fueron los resultados con respecto a una solución teórica ya aprobada, comparando éstos con el fin de disminuir el error. Partiendo que este método es empleado en 2D, se quiere lograr la inclusión de otra dimensión en el proceso de mallado y cálculo. Se podrán tratar problemas más reales y ampliar su área de aplicación.

Por otra parte, se pretende fomentar la investigación de la mecánica computacional dentro de la Escuela de Ingeniería Mecánica, UIS.

El análisis del error en mallas de elementos finitos convencionales en 2D es de gran ayuda para dar solución de problemas de mecánica de materiales, lo que hace pensar que con un análisis del error en mallas de elementos finitos poligonales arbitrarios en 3D se pueden lograr resultados numéricos con mayor exactitud en situaciones más reales. En la Escuela se tiene experiencia en el desarrollo de estimadores de error basados en la reconstrucción de la solución FEM, por lo cual se escogen como punto de partida para la creación de un algoritmo de control del error.

Por otro lado, los proyectos de investigación enfocados al desarrollo de software para ingeniería pueden motivar a la comunidad estudiantil a vincularse más en este tipo de actividades, con un posible impacto en la región al estimular otros sectores productivos.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar una herramienta que permita el control del error y mejorar la precisión de aproximaciones basadas en elementos finitos con formas poligonales arbitrarias 3D contribuyendo con la misión de la escuela de ingeniería mecánica de la Universidad Industrial de Santander haciendo referencia a la investigación y aplicación de los conocimientos previamente adquiridos.

2.2 OBJETIVOS ESPECÍFICOS

- Implementar el método de elementos finitos para mallas poligonales en 3D para problemas de elasticidad lineal en Matlab.
- Desarrollar una técnica para estimar el error en mallas poligonales de elementos finitos en 3D basada en la reconstrucción de la solución en tensiones.
- Evaluar el desempeño de la técnica propuesta utilizando dos problemas con solución analítica: placa cuadrada sometida a cargas externas y cilindro con presión interna.

3. APROXIMACIÓN DE LOS ELEMENTOS FINITOS PARA MALLAS POLIGONALES

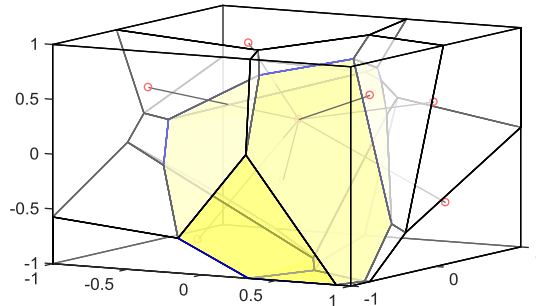
El análisis de elementos finitos funciona como una aproximación de soluciones, esta herramienta de estudio se utiliza en problemas de contorno frecuentes en la ingeniería, y permite comprender de forma más clara el comportamiento de los sistemas. Este método basado en elementos finitos ha surgido como una analogía entre elementos discretos y particiones finitas en un dominio continuo [32].

3.1 DISCRETIZACIÓN DE LOS ELEMENTOS POLIGONALES

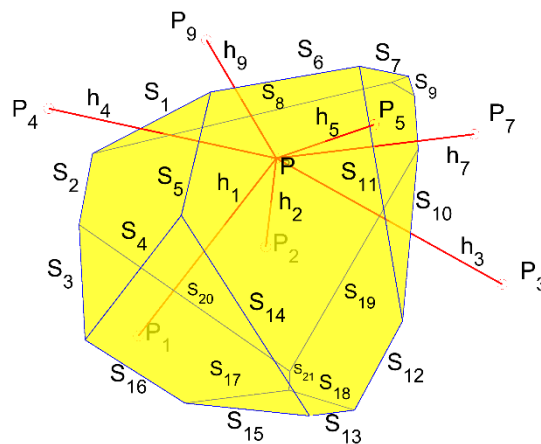
Las mallas poliédricas están compuestas por elementos con más de 4 lados, como hexágonos, pentágonos, etc. Con el estudio de elementos poliédricos se permiten construir mallas complicadas que requieren alto grado de flexibilidad, geometrías complicadas sin la necesidad de aumentar el número de elementos o el número de funciones de forma, ya que requieren mayor capacidad computacional para resolver el problema. Actualmente, existen programas enfocados a la construcción de mallas arbitrarias poligonales en 2D y poliédricas en 3D, los cuales son implementados en el estudio y la aplicación de elementos finitos a problemas de ingeniería, por ejemplo, Neper [33] , Voro++ [34] o Cantiléver [6] que se destacan por realizar mallas arbitrarias Voronoi y extenderlas a formas poliedras.

Considere la región del dominio limitado $\Omega \in \mathbb{R}^3$, que esta descrita por nodos dispersos $N = \{p_1, p_2, p_3, \dots, p_n\}$ en Ω y siendo $V(N)$ subdivisiones del dominio celdas Voronoi, entonces, en cualquier punto dentro $V(p_i)$ esta más cercano al nodo p_i que a cualquier otro nodo $p_j \in N$ ($i \neq j$) [2], donde las geometrías $V(p_i)$ tienen dominios convexos \mathbb{R}^3 . Los poliedros son construidos conectando los nodos que tienen en común los bordes del elemento Voronoi como se muestra en la Figura 1.

Figura 1. Elemento Voronoi 3D: (a) Teselación Voronoi para un cubo, (b) Vecinos naturales para nodo P, según la función de forma de Laplace.



(a)



(b)

La interpolación de Laplace, se propuso inicialmente para reducir las interpolaciones en los elementos finitos vista desde un punto más general, en [35] se propuso la interpolación para elementos en tres dimensiones, poliedros, elementos con dominios en el espacio físico, $x_i \in \Omega$. En la figura 1, se muestra una celda Voronoi donde el punto p posee un lado Voronoi común con otro punto p_i , se dice que estos dos son vecinos naturales, y para geometrías en 2D cumplen la propiedad de los círculos denominada triangulaciones de Delaunay, para el caso 3D esta se adapta a esferas.

Para dar solución a un problema, gobernado por ecuaciones diferenciales parciales con modelamientos geométricos, se dispone de un esquema de interpolación que

trata en declaraciones variacionales representadas como funciones de forma, en [36] se formuló la interpolación de Laplace para el punto p .

$$\phi_i^l(\mathbf{x}) = \frac{\alpha_i(\mathbf{x})}{\sum_{j=1}^n \alpha_j(\mathbf{x})}, \quad \alpha_j(\mathbf{x}) = \frac{s_j(\mathbf{x})}{h_j(\mathbf{x})}, \quad \mathbf{x} \in \mathbb{R}^2, \quad (1)$$

Donde $\phi_i^l(\mathbf{x})$ es la función de forma (Wachspress) para el nodo i , $\alpha_i(\mathbf{x})$ es la función de ponderación de Laplace, $s_i(\mathbf{x})$ es la longitud del borde de Voronoi asociado a los nodos p y p_i y $h_i(\mathbf{x})$ es la distancia Euclidiana entre p y p_i .

Partiendo de la definición del problema elástico, en busca de los desplazamientos \mathbf{u}^h se establece el sistema de ecuaciones y el tipo de frontera Dirichlet.

$$\mathbf{F} = \mathbf{K}\mathbf{u} \quad (2)$$

$$\mathbf{K} = \sum_e \mathbf{K}^e = \sum_e \int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} dV \quad (3)$$

Donde \mathbf{K} es la matriz global de rigidez, \mathbf{F} es el vector de fuerzas global nodal, \mathbf{D} es la matriz de propiedades del material, \mathbf{B} es la matriz de deformación-desplazamiento y se define como la derivada de las funciones de forma (Wachspress) $\mathbf{B} = \nabla \phi(\mathbf{x})$.

Para el vector de fuerzas, se tomará únicamente en cuenta la acción de una fuerza superficial \mathbf{t}

$$\mathbf{F} = \int_{\Omega} \mathbf{N}^T \mathbf{t} dA \quad (4)$$

El vector de tracción \mathbf{t} , se determina con:

$$\mathbf{t} = P * \mathbf{n} \quad (5)$$

donde P es el valor de la presión en la superficie y \mathbf{n} es el vector normal a la superficie.

Para las geometrías que presentan formas planas o curvas, el vector \mathbf{t} , se establece en función de ds , según la dirección descrita por vectores unitarios \mathbf{e} . En el Anexo D, se muestra la técnica de transformación de sistema de referencia, para conseguir las componentes de las fuerzas que actúan en el contorno Dirichlet dA , finalmente, se interpolan los valores de Gauss, en cada punto de muestreo localizados en la superficie donde actúa P , para obtener el valor nodal de la fuerza.

3.2 INTEGRACIÓN NUMÉRICA EN ELEMENTOS FINITOS POLIGONALES

Los métodos de interpolación se establecen para resolver las funciones polinómicas incógnita del problema, y según la geometría del elemento se disponen de cuadraturas de integración, para el caso de elementos poligonales se emplea el tratamiento de funciones baricéntricas, cuyas características permiten tratar diferentes geometrías convexas y no convexas [37]. Para funciones no polinómicas, discontinuas, las cuadraturas de integración resultan de baja calidad y precisión en la aproximación, para dar solución a este problema en [1], [38] se propone dividir el elemento en subdominios conocidos (polígonos de referencia), como triángulos o rectángulos para 2D y tetraedros o cubos en 3D. Con estas técnicas de integración se pueden aplicar las cuadraturas ya establecidas de estas formas geométricas.

En [2] se presenta una técnica de transformación paramétrica, donde la integración de Laplace se hace tomando los polígonos de referencia como base para realizar el mapeo en el elemento poligonal global usando funciones de forma.

$$\begin{aligned}
 x &= \sum N_i(\xi, \eta, \tau) * x_i \\
 y &= \sum N_i(\xi, \eta, \tau) * y_i \\
 z &= \sum N_i(\xi, \eta, \tau) * z_i
 \end{aligned}
 \tag{6}$$

Las funciones de forma N_i , son evaluadas para los polígonos de referencia. Las variables (x, y, z) representan las coordenadas globales de los puntos nodales del polígono global, y (x_i, y_i, z_i) definen las coordenadas locales del polígono de referencia. En la Figura 2 se muestra el poliedro global y un subdominio tetraedro tomado como poliedro de referencia.

Figura 2. Mapeo en el elemento poligonal global y su poliedro de referencia.

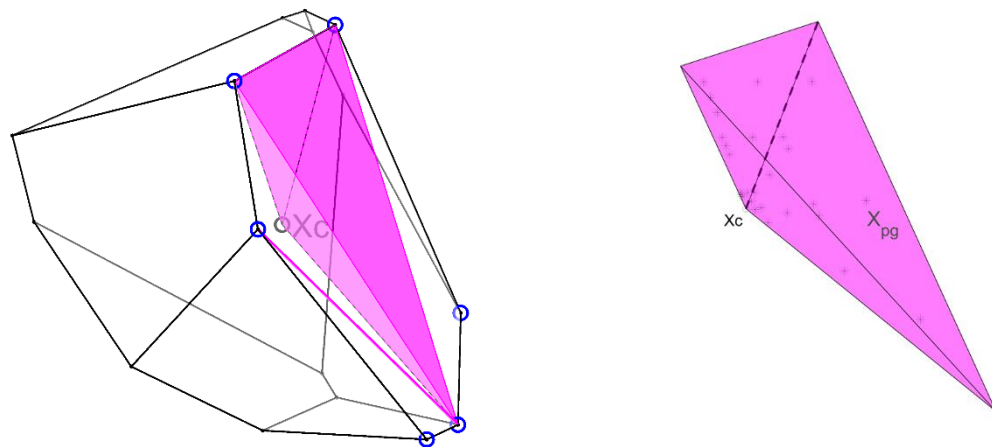
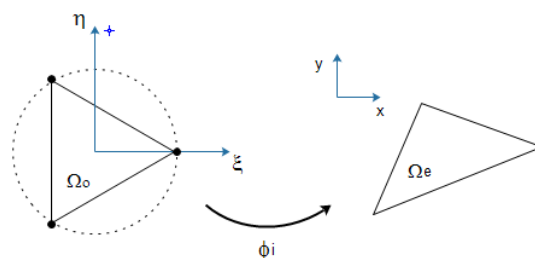


Figura 3. Transformación isoparamétrica para un triángulo.



En la Figura 3, se muestra el mapeo de un triángulo en la geometría normalizada para 2D, y su respectiva representación matemática es:

$$(x, y) = \phi(\xi, \eta) \leftrightarrow (\xi, \eta) = \phi^{-1}(x, y) \quad (7)$$

Donde ϕ , es la función de interpolación y (ξ, η) son las coordenadas locales del dominio. Entonces la integración adoptando las tres coordenadas espaciales normalizadas resulta:

$$\int_{\Omega_e} \phi(x, y, z) dx dy dz = \int_{\Omega_o} \phi(\xi, \eta, \tau) * |J| d\xi d\eta d\tau \quad (8)$$

De acuerdo con la regla de cuadratura Gaussiana, la integral se puede ver de la siguiente forma:

$$I = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \phi(\xi, \eta, \tau) |J| d\xi d\eta d\tau \approx \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^p \phi(\xi_i, \eta_j, \tau_k) |J| H_i H_j H_k \quad (9)$$

Donde H son evaluados en los puntos de integración de la región Ω , y corresponden al peso de cuadratura en cada punto. Los poliedros de grado N pueden descomponerse en N sub-dominios, entonces la integración de cualquier función ϕ en un dominio Ω resulta:

$$\int_{\Omega} \phi d\Omega = \sum_N \int_{\Omega_i} \phi |J| |J_i| d\bar{\xi} d\bar{\eta} d\bar{\tau} \quad (10)$$

La subdivisión del elemento implica un mapeo isoparamétrico, donde las coordenadas (ξ, η, τ) preparan las funciones de forma de Laplace para su integración [1].

4. FORMULACIÓN DEL PROBLEMA ELÁSTICO - LINEAL

En esta sección, se presenta el análisis del problema lineal elástico 3D. A partir de la notación de Voigt, se establece un dominio $\Omega \subset \mathbb{R}^3$, que es descrito por n nodos. Se define la matriz esfuerzos en el espacio como:

$$\boldsymbol{\sigma} = \{\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx}\}^T$$

El desplazamiento en sus tres componentes $\mathbf{u} = \{u, v, w\}^T$, cada una corresponde a una dirección de las coordenadas x, y, z del espacio, y la deformación por $\boldsymbol{\varepsilon} = \{\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}\}^T$. Estos campos se encuentran definidos sobre el dominio Ω establecido. De las fuerzas que actúan sobre el cuerpo, se definen las fuerzas por unidad de superficie, denotadas como $\mathbf{t} = \{t_x, t_y, t_z\}^T$, y son impuestas sobre los límites de Neumann Γ_N , las fuerzas por unidad de volumen están denotadas como $\mathbf{b} = \{b_x, b_y, b_z\}^T$, y los desplazamientos $\bar{\mathbf{u}}$ hacen parte de las condiciones de Dirichlet del problema, $\mathbf{u} = \bar{\mathbf{u}}$ en Γ_D .

Para el problema elástico, se establecen las condiciones a satisfacer

$$\mathbf{L}^T \boldsymbol{\sigma} + \mathbf{b} = 0 \quad \text{En } \Omega \quad (11)$$

$$\mathbf{G} \boldsymbol{\sigma} = \mathbf{t} \quad \text{En } \Gamma_N \quad (12)$$

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \mathbf{L} \mathbf{u} \quad \text{En } \Omega \quad (13)$$

Donde \mathbf{L} corresponde al operador diferencial de la elasticidad lineal y se expresa.

$$\mathbf{L}^T = \begin{bmatrix} \partial/\partial x & 0 & 0 & 0 & \partial/\partial z & \partial/\partial y \\ 0 & \partial/\partial y & 0 & \partial/\partial z & 0 & \partial/\partial x \\ 0 & 0 & \partial/\partial z & \partial/\partial y & \partial/\partial x & 0 \end{bmatrix}$$

\mathbf{G} es el operador que proyecta el campo de esfuerzo en tracciones sobre los límites del dominio.

También en la frontera, se tiene en cuenta la ley de Hooke como una relación entre tensión y deformación, donde el operador \mathbf{D} es la matriz de elasticidad del material.

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u}) \quad (14)$$

Para la solución del problema se dispone del método integral con un enfoque variacional, que se basa en el planteamiento de la energía potencial total del cuerpo elástico, para encontrar $\mathbf{u} \in \mathbf{V}$ de forma que $\forall \mathbf{v} \in \mathbf{V}$:

$$\begin{aligned} \Pi_p = & \int_{\Omega} \frac{1}{2} \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u}) d\Omega - \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D}\boldsymbol{\varepsilon}_0 d\Omega + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \boldsymbol{\sigma}_0 d\Omega - \int_{\Omega} \mathbf{v}^T \mathbf{b} d\Omega \\ & - \int_{\Gamma_N} \mathbf{v}^T \mathbf{t} d\Gamma \end{aligned} \quad (15)$$

La ecuación 15 se puede desarrollar de forma discreta, empleando las ecuaciones 13,14, $\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{u}^e$ y $\mathbf{B} = \mathbf{L}\mathbf{N}$. Con las funciones de forma evaluadas en cada elemento, la energía potencial total del sistema se obtiene:

$$\begin{aligned} \Pi_p = & \frac{1}{2} \mathbf{u}_e^T \int_{\Omega_e} \mathbf{B}^T \mathbf{D}\mathbf{B} dV * \mathbf{u}_e - \mathbf{u}_e^T \int_{\Omega_e} \mathbf{B}^T \mathbf{D}\boldsymbol{\varepsilon}_0 dV + \\ & \mathbf{u}_e^T \int_{\Omega_e} \mathbf{B}^T \boldsymbol{\sigma}_0 dV - \mathbf{u}_e^T \int_{\Omega_e} \mathbf{N}^T \mathbf{b} dV - \mathbf{u}_e^T \int_{S_e} \mathbf{N}^T \mathbf{t} dV \end{aligned} \quad (16)$$

$$\begin{aligned} \Pi_p = & \frac{1}{2} * \sum_{e=1}^n u_e^T K_e u_e \\ & - \sum_{e=1}^n u_e^T \left[\int_{\Omega_e} \mathbf{B}^T \mathbf{D}\boldsymbol{\varepsilon}_0 dV - \int_{\Omega_e} \mathbf{B}^T \boldsymbol{\sigma}_0 dV + \int_{\Omega_e} \mathbf{N}^T \mathbf{b} dV + \int_{S_e} \mathbf{N}^T \mathbf{t} dV \right] \end{aligned} \quad (17)$$

Donde $\mathbf{V} = \{\mathbf{v} | \mathbf{v} \in [H^1(\Omega)]^2, \mathbf{v}|_{\Gamma_D}(\mathbf{x}) = \mathbf{0}\}$ es un espacio de prueba del problema elástico [26]. La solución se desarrolla basada en el planteamiento variacional de Galerkin, y necesita como requisito que las funciones de forma sean no negativas, que cumplan la partición de la unidad, completitud, y que garanticen la interpolación nodal y la linealidad en las fronteras.

5. ERROR ESTIMADO EN NORMA ENERGÉTICA

El error en una aproximación de elementos finitos se define como diferencia de soluciones, para desplazamientos $\mathbf{e} = \mathbf{u} - \mathbf{u}^h$, donde \mathbf{e} es el error de discretización, \mathbf{u} es la solución de referencia y \mathbf{u}^h es evaluada por elementos finitos. El error se puede cuantificar a partir de la norma energética, que resulta en una forma de interpretación global del error.

$$\|\mathbf{e}\|^2 = \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{e})^T \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{e}) \, d\Omega \quad (18)$$

De igual forma para la diferencia de soluciones, en el campo de tensiones el error $\mathbf{e}_{\sigma} = \boldsymbol{\sigma} - \boldsymbol{\sigma}^h$, se puede cuantificar como:

$$\|\mathbf{e}\|^2 = \int_{\Omega} \mathbf{e}_{\sigma}^T \mathbf{D}^{-1} \mathbf{e}_{\sigma} \, d\Omega \quad (19)$$

5.1 RECONSTRUCCIÓN DE TENSIONES

Para problemas complejos, la solución exacta es difícil de determinar, por tanto, su error exacto solo puede ser estimado. Zienkiewicz y Zhu [39] definieron la estimación del error exacto, $\boldsymbol{\varepsilon}$, para una aproximación como:

$$\|\mathbf{e}\|^2 \approx \boldsymbol{\varepsilon} = \int_{\Omega} \mathbf{e}_{\sigma}^{*T} \mathbf{D}^{-1} \mathbf{e}_{\sigma}^* \, d\Omega \quad (20)$$

Donde el error $\mathbf{e}_{\sigma}^* = \boldsymbol{\sigma}^* - \boldsymbol{\sigma}^h$, siendo $\boldsymbol{\sigma}^*$ el campo de tensiones reconstruido, que se define como la solución más aproximada a la exacta, por tanto, la calidad de la técnica de reconstrucción de la solución influye en la precisión del estimador del error.

La estimación del error en norma energética, se basa en técnicas de reconstrucción, y son diversos los procedimientos [21], [22], [23], [24], [40], por ejemplo, la

aproximación nodal que es comúnmente usada para conseguir campos continuos de tensiones o tensiones alisadas (es usada en el trabajo actual), otra técnica es la reconstrucción de tensiones por super-convergencia de la solución FEM (SPR), que consiste en el uso de funciones de interpolación que son aplicadas a patch formados por los elementos que contienen al nodo de consideración. Otra técnica usualmente empleada en estimación del error, es la reconstrucción de tensiones por mínimos cuadrados ponderados (MLS), que funciona como una forma alternativa de SPR, puesto que considera dentro de estos patch, solamente los puntos cercanos al punto de consideración. Las funciones que contienen las tensiones se ajustan por mínimos cuadrados. Ahora se sigue trabajando en aplicaciones de estos procedimientos, porque demuestran ser robustos en los estimadores de error, obteniendo variaciones y mejoras en estas técnicas de reconstrucción en [25], [26].

El método de reconstrucción basado en promediado en nodos, se calcula el campo de tensiones reconstruido a partir de un alisado de las soluciones, interpolando en los puntos de muestreo la tensión en cada elemento, usando las funciones de forma como aproximación de elementos finitos.

$$\sigma^*(x) = \sum_{I=1}^{n_e} N_I(x) \sigma_I^*(x_I) \quad (21)$$

En la cual n_e es el numero de nodos en el elemento, $\sigma_I^*(x_I)$ son los campos de tensiones evaluados en el nodo I , por una técnica de alisado de soluciones, que consiste en asignar a cada nodo de la malla el valor promedio de las tensiones evaluadas en dicho nodo como correspondiente a cada uno de los elementos a los que está conectado [40].

La tensión alisada en el nodo j , para el elemento e , se escribe:

$$\sigma_I^*(x_I) = \frac{\sum_{j=1}^{M_j} \sigma_j^h(x)}{M_j} \quad (22)$$

Donde M_j , es el número de elementos que contienen al nodo j , y $\sigma_j^h(x)$ es la tensión del elemento j evaluada por FEM.

6. RESULTADOS NUMÉRICOS

Se proponen dos problemas, un cubo sometido a carga externa y un cilindro hueco expuesto a presión interna. Cuyas soluciones exactas serán comparadas con la obtenida por la técnica de reconstrucción de tensiones de promediado de nodos, también se presentan los resultados de comparación por medio de gráficas de efectividad local y error estimado en norma energética. Finalmente, para cada problema se muestra la evolución del error, según los grados de libertad en un gráfico logarítmico.

La efectividad representa que tan confiable es el estimador del error, haciendo una comparación entre el error obtenido con el cálculo de tensiones FEM y el error obtenido con un método de reconstrucción de tensiones. Un estimador de error, para trabajar con métodos adaptativos debe presentar efectividades dentro del rango [0.8-1.2]. La efectividad del estimador de error se calcula:

$$\theta = \frac{\varepsilon^{PN}}{\|e\|} \quad (23)$$

Se introduce un elemento de indexación D , definido como

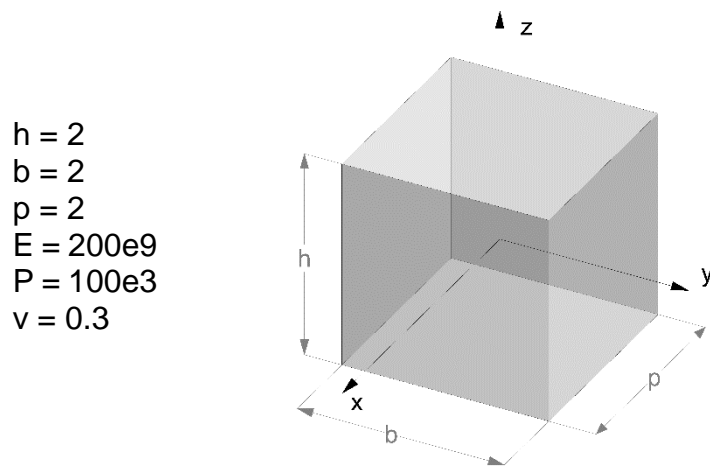
$$D = \begin{cases} \theta^e - 1, & \text{si } \theta^e \geq 1 \\ 1 - \frac{1}{\theta^e}, & \text{si } \theta^e < 1 \end{cases} \quad (24)$$

Donde θ^e es la efectividad en el elemento. Para calcular θ^e , se utiliza la ecuación 22, pero con el dominio de integración ahora siendo el elemento en vez que el dominio físico.

6.1 PLACA CUADRADA SOMETIDA A CARGA EXTERNAS

En la figura 5, se muestra un cubo sólido sometido a tensión por acción de una presión P normal a la cara lateral positiva XZ y se restringen los desplazamientos en y , los nodos de la cara opuesta XZ negativa. Las condiciones geométricas y propiedades del material se muestran a continuación.

Figura 4. Cubo sometido a presión externa.



La solución del problema elástico se efectúa con una malla arbitraria de elementos poligonales, evaluando el muestreo en los puntos de Gauss. La validación del problema se hará inicialmente a partir de la solución exacta, cuyo valor se calcula como:

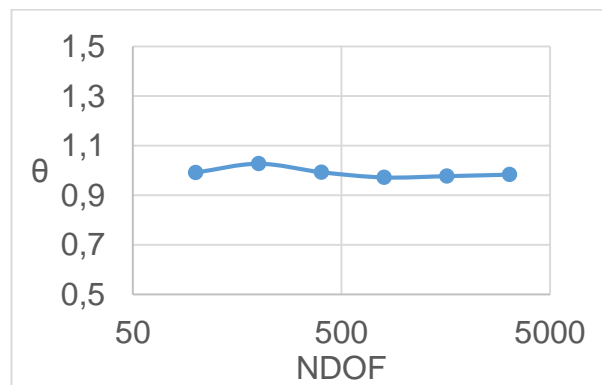
$$\sigma = \frac{F}{A} \quad (25)$$

En la tabla 1 se muestran los resultados de efectividad global y norma energética para cada uno de los grados de libertad propuestos. Se observa una efectividad global que tiende a estabilizarse en 1 (Figura 5), lo que significa que el análisis tiene un alto grado de confiabilidad.

Tabla 1. Efectividad global y norma energética del cilindro hueco para diferentes grados de libertad.

NDOF ⁵	θ	$\ U\ $
100	0,9919	0,0251
200	1,0273	0,0254
400	0,9933	0,0251
800	0,9722	0,0254
1600	0,9774	0,0252
3200	0,9835	0,0251

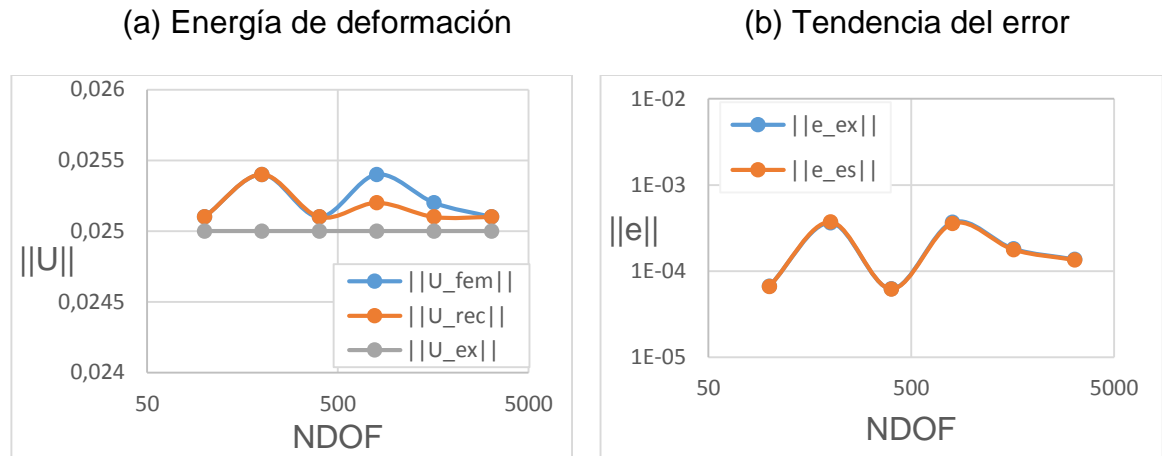
Figura 5. Efectividad global vs grados de libertad.



En la Figura 6.a se presentan los resultados obtenidos de la energía de deformación para cada solución (FEM, exacta y reconstruida). En la Figura 6.b se muestra la variación de las dos tendencias del error (exacto y estimado). Todo esto en los grados de libertad propuestos.

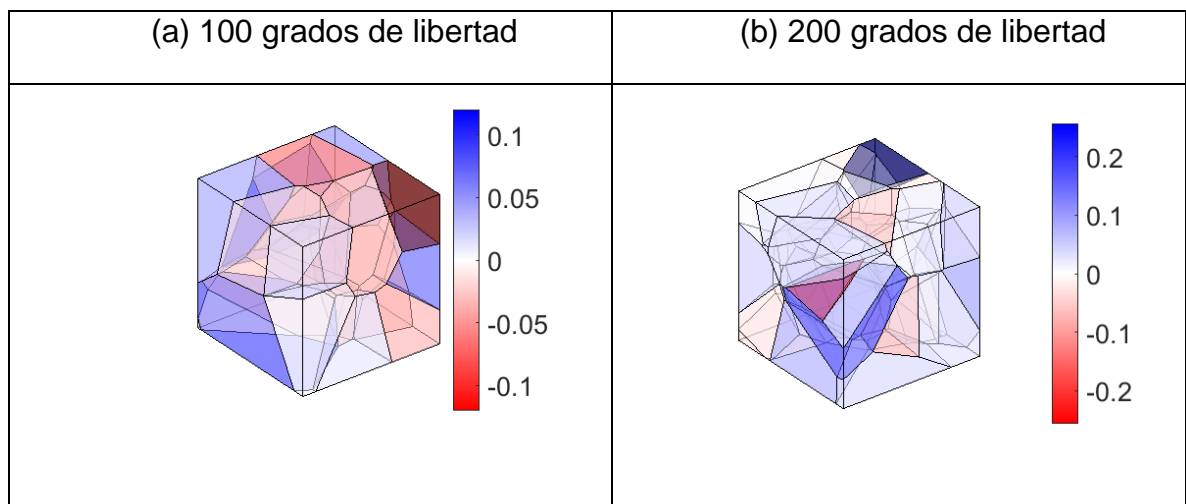
⁵ NDOF: Number degrees of freedom (número de grados de libertad).

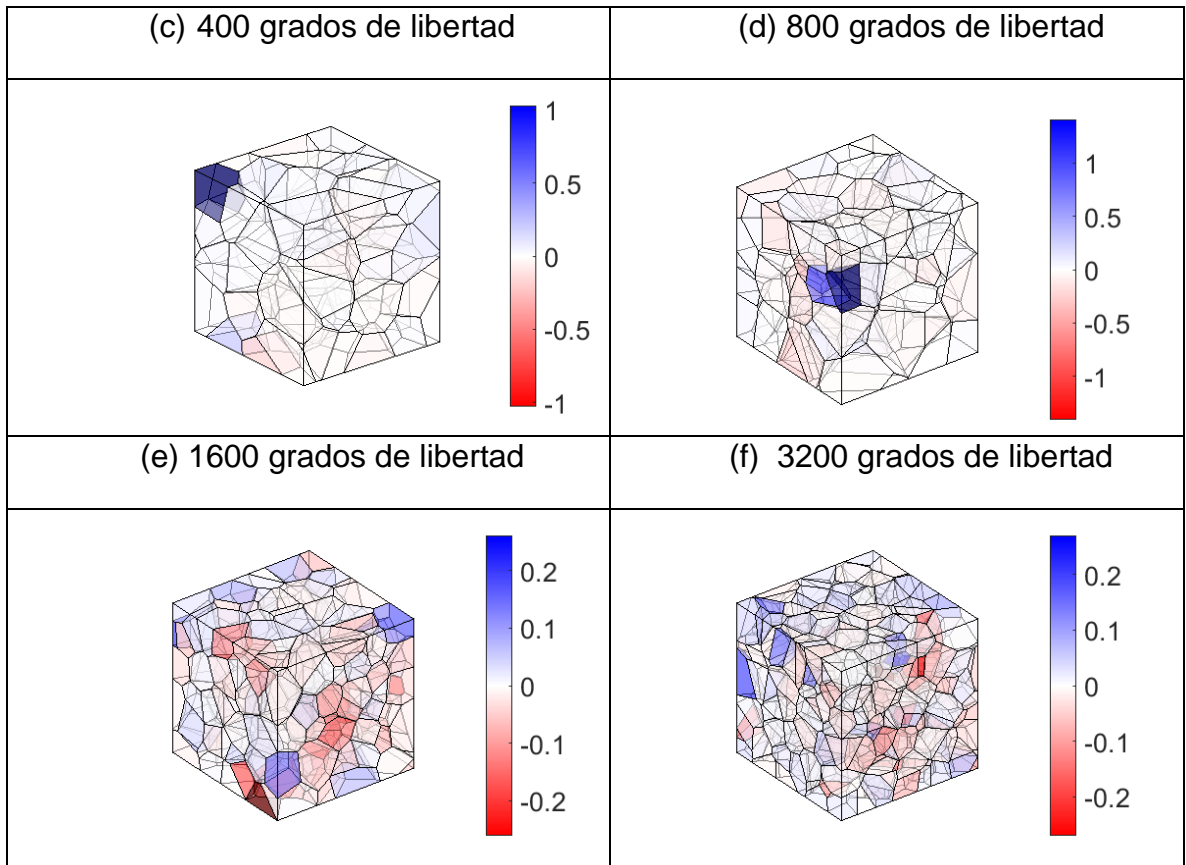
Figura 6. Energía de deformación y tendencia del error vs grados de libertad.



Al realizar los cálculos de efectividad local a nivel del elemento, se observa como a medida que aumentan los grados de libertad la efectividad local tiende a 0. Esto se debe, a que, al incrementar el número de datos de análisis utilizados para realizar los cálculos, el mallado se asemeja más a la situación real, aumentando la precisión de los resultados y disminuyendo el error a la hora de hacer una comparación con la solución exacta. Ver Figura 7.

Figura 7. Efectividad local a nivel del elemento.

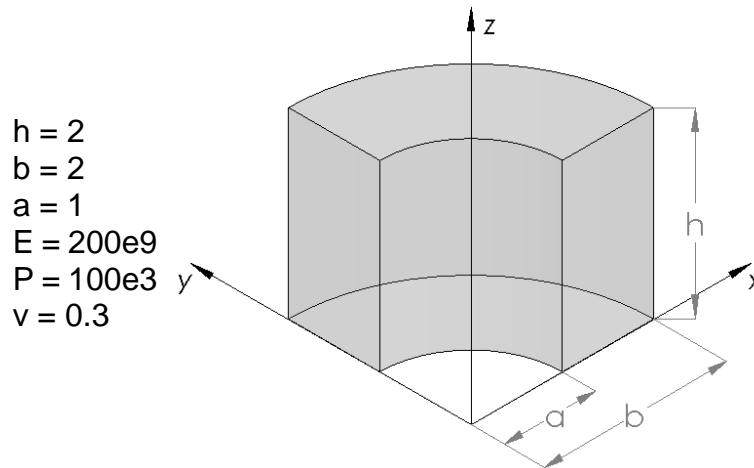




6.2 CILINDRO BAJO PRESIÓN INTERNA

Se modela un cuarto de cilindro hueco sometido a una presión interna P en la cara circular interna, cuyas condiciones geométricas y propiedades del material se muestran en la Figura 8. El cuerpo es restringido simétricamente en las caras XZ y YZ . Se fijan sus desplazamientos en dirección Z en las caras normales a XY .

Figura 8. Cilindro sometido a presión interna.



Este problema elástico se analiza con una malla de poliedros arbitrarios lineales, se conoce su solución exacta, la cual se obtiene asumiendo un comportamiento de deformación plana. El valor de la tensión exacta es obtenido a partir de las siguientes ecuaciones:

$$k = \frac{b}{a}$$

$$\sigma_r = \frac{P}{k^2 - 1} \left(1 - \frac{b^2}{r^2} \right); \quad \sigma_t = \frac{P}{k^2 - 1} \left(1 + \frac{b^2}{r^2} \right) \quad (26)$$

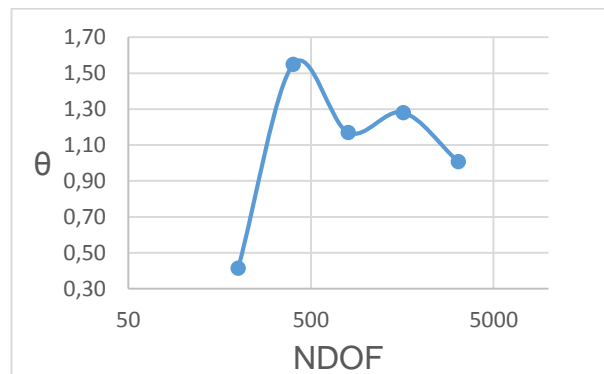
$$\begin{aligned}
\sigma_{xx} &= \sigma_r \cos^2 \phi + \sigma_t \sin^2 \phi; & \sigma_{yy} &= \sigma_r \sin^2 \phi + \sigma_t \cos^2 \phi \\
\sigma_{zz} &= \nu(\sigma_{xx} + \sigma_{yy}); & \tau_{xy} &= (\sigma_r - \sigma_t) \sin \phi \cos \phi \\
\tau_{xy} &= (\sigma_r - \sigma_t) \sin \phi \cos \phi & & \\
\tau_{xz} &= 0; & \tau_{yz} &= 0
\end{aligned}
\tag{27}$$

Al igual que sucedía en el análisis del cubo, los resultados obtenidos en el problema del cilindro presentan una efectividad global que tiende a estabilizarse en 1 (Figura 9). En la Tabla 3 se muestra la efectividad global y norma energética para cada uno de los grados de libertad propuestos.

Tabla 2. Norma energética y efectividad global del cilindro hueco para diferentes grados de libertad.

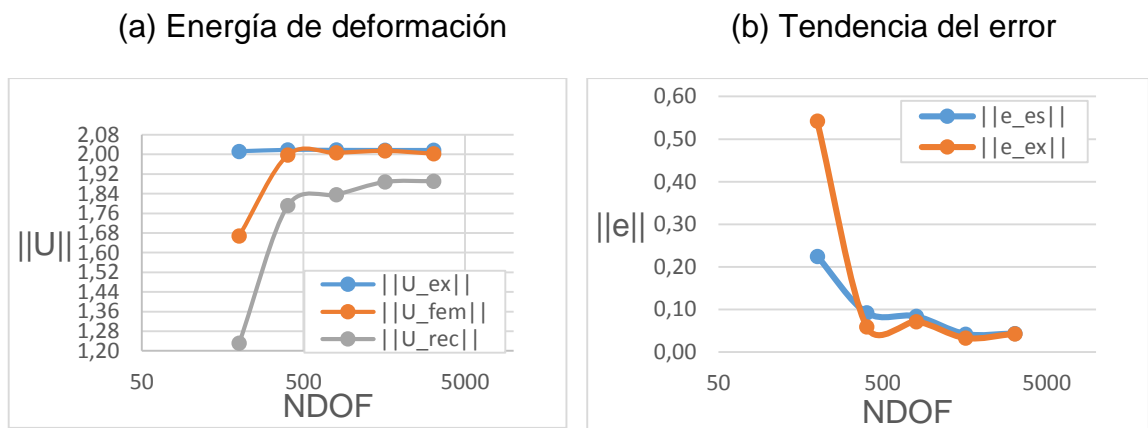
NDOF	θ	$\ U\ $
200	0,4144	1,6688
400	0,5490	1,9975
800	1,1689	2,0069
1600	1,2808	2,0147
3200	1,0073	2,0034

Figura 9. Efectividad global vs Grados de libertad.



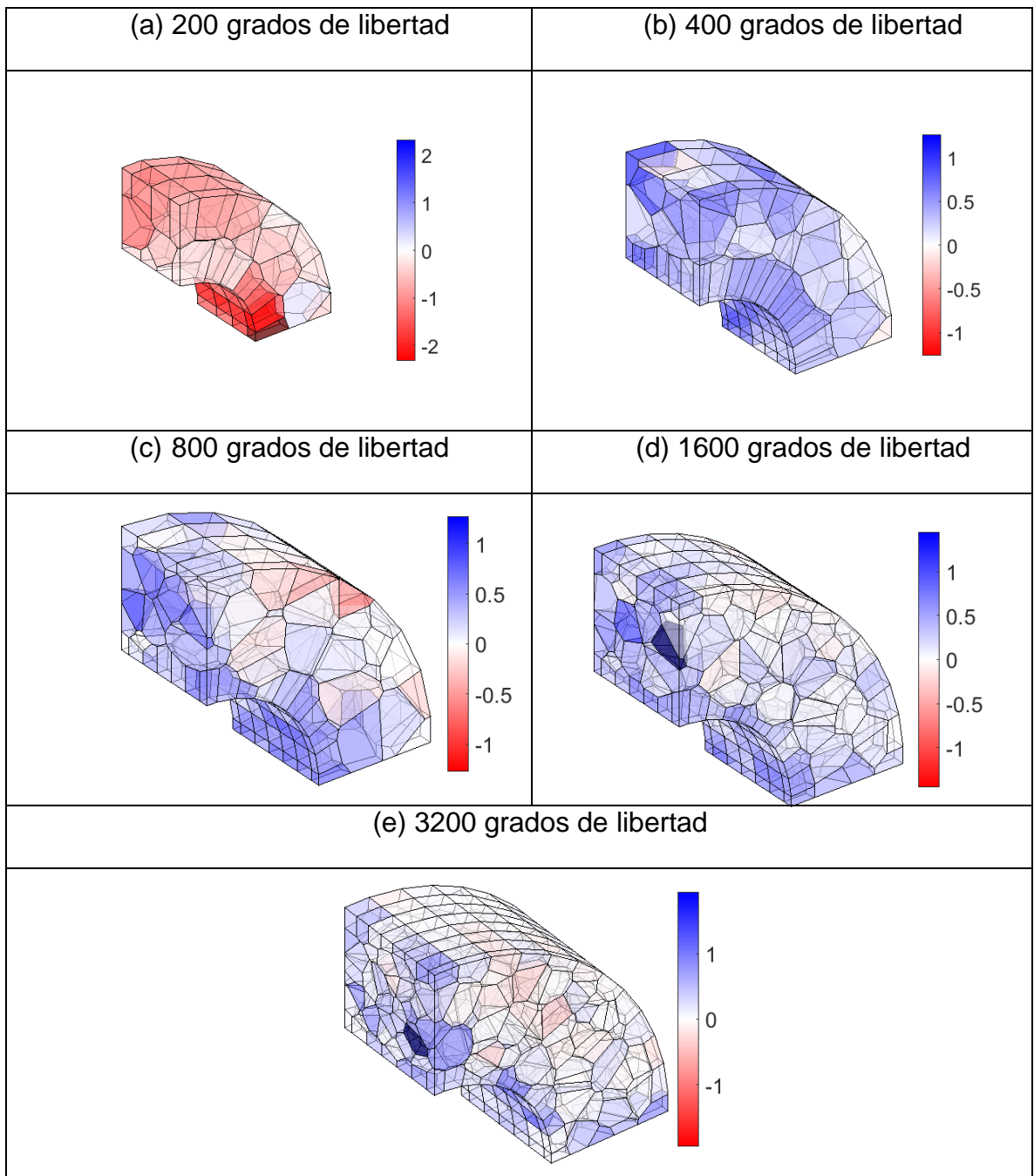
En la Figura 10.a se presentan los resultados obtenidos de la energía de deformación para cada solución (FEM, exacta y reconstruida). En la Figura 10.b se muestra la variación de las dos tendencias del error (exacto y estimado). Todo esto en los grados de libertad propuestos.

Figura 10. Energía de deformación y tendencia del error vs grados de libertad.



Al igual que en el problema anterior, al realizar los cálculos de efectividad local a nivel del elemento, se observa como a medida que aumentan los grados de libertad la efectividad local tiende a 0 (Figura 11). Cabe resaltar que al tener solo 200 grados de libertad se presenta bastante margen de error, la efectividad local para este caso llega a un valor superior a 2, y una efectividad global de 0,4144. Esto indica una estimación poco confiable a comparación de las siguientes pruebas, por lo que se recomienda no trabajar con menos de 400 grados de libertad en casos como este.

Figura 11. Efectividades locales a nivel de elemento.



7. CONCLUSIONES

- En el método de elementos finitos empleado en problemas de elasticidad lineal, se observa que la aplicación de elementos poligonales para mallados 3D puede ser de gran ayuda, especialmente al momento de estudiar problemas con geometrías no regulares donde se requiera flexibilidad en el mallado.
- Se observó la importancia de la calidad del mallado en el análisis de elementos finitos, y como afecta la solución con la variación del tamaño de los elementos. Cabe resaltar la dificultad que actualmente existen en los malladores 3D con teselaciones Voronoi para crear diferentes figuras con superficies curvas, pues las fronteras no son uniformemente cubiertas y fue necesario deformar los elementos originales que podría introducir un error en la solución.
- La técnica de reconstrucción empleada de promediado en nodos, a pesar de no ser una técnica robusta a la hora de suavizar la solución, logra obtener resultados cercanos a los esperados, por ejemplo, la efectividad global cercana a 1. Esto se podría mejorar con el uso de otras técnicas más robustas como MLS o SPR, donde se hacen un análisis más detallado de los puntos de muestreo vecinos, aproximando mejor a la solución.
- Se puede estudiar el número de la cuadratura de Gauss para los puntos de muestreo, debido a que existen casos donde su tamaño sobra pues no hay mejora en la aproximación de la solución, de esta forma se puede evitar tiempos de cálculo (ver Anexo C).

REFERENCIA BIBLIOGRAFICA

- [1] KRAUS, Markus; RAJAGOPAL, Amirthan y STEINMANN, Paul. Investigations on the polygonal finite element method: Constrained adaptive Delaunay tessellation and conformal interpolants. En: Computers and Structures, 2013, vol. 120, p. 33–46.
- [2] SUKUMAR, N. y TABARRAEI A. Conforming polygonal finite elements. En: international Journal for Numerical Methods in Engineering, Noviembre 2004, vol. 61, no. 12, p. 2045–2066.
- [3] JAYABAL, K., *et al.* Micromechanical modelling of switching phenomena in polycrystalline piezoceramics: Application of a polygonal finite element approach. En: Computational Mechanics, Abril 2011, vol. 48, no. 4, p. 421–435.
- [4] MALSCH, Elisabeth y DASGUPTA, Gautam. Interpolations for temperature distributions : a method for all non-concave polygons. En: International Journal of Solids and Structures, 2004, vol. 41, p. 2165–2188.
- [5] BIABANAKI, S. y KHOEI, A. A polygonal finite element method for modeling arbitrary interfaces in large deformation problems. En: Computational Mechanics, 2012, vol. 50, no.1, p. 19-33.
- [6] TALISCHI, Cameron, *et al.* PolyMesher : a general-purpose mesh generator for polygonal elements written in Matlab. En: Structural and Multidisciplinary Optimization, 2012, vol. 45, no. 3, p. 309–328.
- [7] WACHSPRESS, Eugene. A rational basis for function approximation. En: IMA Journal of Applied Mathematics, 1973, vol. 11, no. 1, p. 83-104.
- [8] WACHSPRESS, Eugene. Barycentric coordinates for polytopes. En: Computers and Mathematics with Applications, 2011, vol. 61, no. 11, p. 3319–3321.
- [9] WARREN, Joe, Barycentric Coordinates for Convex Polygons. En: Advances in Computational Mathematics, 1996, vol. 6, no. 2, p. 1–14.
- [10] FLOATER, Michael; HORMANN, Kay y KÓŠ, Géza. A General Construction

- of Barycentric Coordinates over Convex Polygons. En: *Advances in Computational Mathematics*, 2006, vol. 24, no. 1, pp. 311–331.
- [11] RAND, Alexander; GILLETTE, Andrew y BAJAJ, Chandrajit. Quadratic serendipity finite elements on polygons using generalized barycentric coordinates. En: *The Institute for Computational Engineering and Sciences*, 2011, vol. 100, no. 2, p. 130–134.
- [12] CHI, Heng, *et al.* Polygonal finite elements for finite elasticity. En: *International Journal for Numerical Methods in Engineering*, 2014, vol. 101, no. 4, p. 305–328.
- [13] GHOSH, Somnath y LIU, Yunshan. Voronoi cell finite element model based on micropolar theory of thermoelasticity for heterogeneous materials. En: *International Journal for Numerical Methods in Engineering*, 1995, vol. 38, no. 8, p. 1361–1398.
- [14] MOORTHY, Suresh y GHOSH, Somnath. Adaptivity and convergence in the Voronoi cell finite element model for analyzing heterogeneous materials. En: *Computer Methods in Applied Mechanics and Engineering*, 2000, vol. 185, p. 37–74.
- [15] NATARAJAN, Sundararajan, *et al.* A new one point quadrature rule over arbitrary star convex polygon/polyhedron. En: *arXiv - Mathematics - Numerical Analysis*, 2017, p. 1-19.
- [16] GOZIN, M. y AGHAIE-KHAFRI, M. 2D and 3D finite element analysis of crack growth under compressive residual stress field. En: *International Journal of Solids and Structures*, 2012, vol. 49, no. 23–24, p. 3316–3322.
- [17] EBEIDA, Mohamed y MITCHELL, Scott. Uniform Random Voronoi Meshes. En: *International Meshing Roundtable*, 2011, vol. 20, p. 1–18.
- [18] GAIN, Arun; TALISCHI, Cameron y PAULINO, Glaucio. On the Virtual Element Method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. En: *Computer Methods in Applied Mechanics and Engineering*, 2014, vol. 282, p. 132–160.
- [19] GONZÁLEZ, Vicente; GINER, Eugenio y TARANCON, Jose. Comparación de

- esquemas de integración 3D para elementos enriquecidos en XFEM. En: Revista UIS Ingenierías, 2016, vol. 15, no. 2, pp. 7–16.
- [20] JAYABAL, Kaliappan y MENZEL, Andreas. Voronoi-based three-dimensional polygonal finite elements for electromechanical problems. En: Computational Materials Science, 2012, vol. 64, p. 66–70.
- [21] RÓDENAS, J. *et al.* A recovery-type error estimator for the extended finite element method based on singular + smooth stress field splitting. En: International Journal for Numerical Methods in Engineering, Octubre 2008, vol. 76, no. 4, p. 545–571.
- [22] GILLETTE, Andrew; RAND, Alexander y BAJAJ, Chandrajit. Error estimates for generalized barycentric interpolation. En: Advances in Computational Mathematics, 2012 vol. 37, no. 3, p. 417–439.
- [23] GONZÁLEZ, Octavio; NATARAJAN, Sundararajan y GRACIANO, Carlos. Recovery based error estimation for the polygonal finite element method. En: Revista UIS Ingenierías, 2017, vol. 1, p. 1–10.
- [24] XIAO, Q y KARIHALOO, B. Statically admissible stress recovery for crack problems. En: ICF XI - 11th International Conference on Fracture, Italia, 2005.
- [25] BORDAS, Stéphane y DUFLOT, Marc. Derivative recovery and a posteriori error estimate for extended finite elements. En: Computer Methods in Applied Mechanics and Engineering, Julio 2007, vol. 196, no. 35–36, p. 3381–3399.
- [26] RÓDENAS, J. *et al.* Enhanced error estimator based on a nearly equilibrated moving least squares recovery technique for FEM and XFEM. En: Computational Mechanics, 2013, vol. 52, no. 2, p. 321–344.
- [27] WEIßER, Steffen. Residual error estimate for BEM-based FEM on polygonal meshes. En: Computers and Mathematics with Applications, 2016, vol. 118, no. 4, p. 765–788.
- [28] HOFREITHER, C. L² error estimates for a nonstandard finite element method on polyhedral meshes, En: Journal of Numerical Mathematics 2011 vol. 19, no. 1, p. 27–39.
- [29] MORA, David; RIVERA, Gonzalo y Rodríguez, Rodolfo. A posteriori error

- estimates for a Virtual Elements Method for the Steklov eigenvalue problem. En: arXiv - Mathematics - Numerical Analysis, 2016, vol. 1, p. 26.
- [30] CANGIANI, Andrea, *et al.* A posteriori error estimates for the virtual element method. En: arXiv - Mathematics - Numerical Analysis, 2017, vol. 2, p. 27.
- [31] VERFÜRTH, Rüdiger. A posteriori error estimation and adaptive mesh-refinement techniques. En: Journal of Computational and Applied Mathematics, 1994, vol. 50, no. 1, p. 67–83.
- [32] ZIENKIEWICZ, O. y TAYLOR, R. *The Finite Element Method: The Basis*,. Barcelona: McGraw-Hill, 2000, ISBN 0 7506 5049 4, p 18-37,112-139, 365-398.
- [33] QUEY, R.; DAWSON, P. y BARBE, F. Large-scale 3D random polycrystals for the finite element method: Generation, meshing and remeshing. En: Computer Methods in Applied Mechanics and Engineering, 2011, vol. 200, no. 17–20, p. 1729–1745.
- [34] RYCROFT, Chris. VORO++: A three-dimensional Voronoi cell library in C++. En: *Chaos*, 2009, vol. 19, no. 4, p. 1–16.
- [35] IDELSOHN, Sergio, *et al.* The meshless finite element method. En: International Journal for Numerical Methods in Engineering, 2003, vol. 58, no. 6, p. 893–912.
- [36] SUKUMAR, N. y MALSCH, E. Recent Advances in the Construction of Polygonal Finite Element Interpolants. En: Archives of Computational Methods in Engineering, 2006, vol. 13, no. 1, p. 129–163.
- [37] HORMANN, Kay y FLOATER, Michael. Mean Value Coordinates for Arbitrary Planar Polygons. En: ACM Transactions on Graphics, 2006, vol. 25, no. 4, p. 1424–1441.
- [38] NATARAJAN, Sundararajan, *et al.* Convergence and accuracy of displacement based finite element formulations over arbitrary polygons: Laplace interpolants, strain smoothing and scaled boundary polygon formulation. En: Finite Elements in Analysis and Design, 2014, vol. 85, p. 101–122.

- [39] ZIENKIEWICZ, O. y ZHU, J. A simple error estimator and adaptive procedure for practical engineering analysis. En: International Journal for Numerical Methods in Engineering, 1987, vol. 24, no. 2, p. 337–357.
- [40] J. E. ARENAS y J. J. RODENAS, Estimación de error y mejora de la solución de desplazamientos de elementos finitos y del rendimiento de solvers iterativos mediante técnicas recovery. Tesis de Máster en Ingeniería Mecánica y Materiales. Valencia.: Universidad Politécnica de Valencia, Departamento de Ingeniería Mecánica y de Materiales, 2011. p. 3–56.

BIBLIOGRFIA

ARENAS, J. y RODENAS, J. Estimación de error y mejora de la solución de desplazamientos de elementos finitos y del rendimiento de solvers iterativos mediante técnicas recovery. Tesis de Máster en Ingeniería Mecánica y Materiales. Valencia.: Universidad Politécnica de Valencia, Departamento de Ingeniería Mecánica y de Materiales, 2011. p. 3–56.

BIABANAKI, S. y KHOEI, A. A polygonal finite element method for modeling arbitrary interfaces in large deformation problems. En: Computational Mechanics, 2012, vol. 50, no.1, p. 19-33.

BORDAS, Stéphane y DUFLOT, Marc. Derivative recovery and a posteriori error estimate for extended finite elements. En: Computer Methods in Applied Mechanics and Engineering, Julio 2007, vol. 196, no. 35–36, p. 3381–3399.

CANGIANI, Andrea, *et al.* A posteriori error estimates for the virtual element method. En: arXiv - Mathematics - Numerical Analysis, 2017, vol. 2, p. 27.

CHI, Heng, *et al.* Polygonal finite elements for finite elasticity. En: International Journal for Numerical Methods in Engineering, 2014, vol. 101, no. 4, p. 305–328.

EBEIDA, Mohamed y MITCHELL, Scott. Uniform Random Voronoi Meshes. En: International Meshing Roundtable, 2011, vol. 20, p. 1–18.

FLOATER, Michael; HORMANN, Kay y KÓŠ, Géza. A General Construction of Barycentric Coordinates over Convex Polygons. En: *Advances in Computational Mathematics*, 2006, vol. 24, no. 1, pp. 311–331.

GAIN, Arun; TALISCHI, Cameron y PAULINO, Glaucio. On the Virtual Element Method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. En: *Computer Methods in Applied Mechanics and Engineering*, 2014, vol. 282, p. 132–160.

GHOSH, Somnath y LIU, Yunshan. Voronoi cell finite element model based on micropolar theory of thermoelasticity for heterogeneous materials. En: *International Journal for Numerical Methods in Engineering*, 1995, vol. 38, no. 8, p. 1361–1398.

GILLETTE, Andrew; RAND, Alexander y BAJAJ, Chandrajit. Error estimates for generalized barycentric interpolation. En: *Advances in Computational Mathematics*, 2012 vol. 37, no. 3, p. 417–439.

GONZÁLEZ, Octavio; NATARAJAN, Sundararajan y GRACIANO, Carlos. Recovery based error estimation for the polygonal finite element method. En: *Revista UIS Ingenierías*, 2017, vol. 1, p. 1–10.

GONZÁLEZ, Vicente; GINER, Eugenio y TARANCON, Jose. Comparación de esquemas de integración 3D para elementos enriquecidos en XFEM. En: *Revista UIS Ingenierías*, 2016, vol. 15, no. 2, pp. 7–16.

GOZIN, M. y AGHAIE-KHAFRI, M. 2D and 3D finite element analysis of crack growth under compressive residual stress field. En: *International Journal of Solids and Structures*, 2012, vol. 49, no. 23–24, p. 3316–3322.

HOFREITHER, C. L 2 error estimates for a nonstandard finite element method on polyhedral meshes, En: *Journal of Numerical Mathematics* 2011 vol. 19, no. 1, p. 27–39.

HORMANN, Kay y FLOATER, Michael. Mean Value Coordinates for Arbitrary Planar Polygons. En: *ACM Transactions on Graphics*, 2006, vol. 25, no. 4, p. 1424–1441.

IDELSOHN, Sergio, *et al.* The meshless finite element method. En: *International Journal for Numerical Methods in Engineering*, 2003, vol. 58, no. 6, p. 893–912.

JAYABAL, K., *et al.* Micromechanical modelling of switching phenomena in polycrystalline piezoceramics: Application of a polygonal finite element approach. En: *Computational Mechanics*, Abril 2011, vol. 48, no. 4, p. 421–435.

JAYABAL, Kaliappan y MENZEL, Andreas. Voronoi-based three-dimensional polygonal finite elements for electromechanical problems. En: *Computational Materials Science*, 2012, vol. 64, p. 66–70.

KRAUS, Markus; RAJAGOPAL, Amirthan y STEINMANN, Paul. Investigations on the polygonal finite element method: Constrained adaptive Delaunay tessellation and conformal interpolants. En: *Computers and Structures*, 2013, vol. 120, p. 33–46.

MALSCH, Elisabeth y DASGUPTA, Gautam. Interpolations for temperature distributions: a method for all non-concave polygons. En: International Journal of Solids and Structures, 2004, vol. 41, p. 2165–2188.

MOORTHY, Suresh y GHOSH, Somnath. Adaptivity and convergence in the Voronoi cell \otimes nite element model for analyzing heterogeneous materials. En: Computer Methods in Applied Mechanics and Engineering, 2000, vol. 185, p. 37–74.

MORA, David; RIVERA, Gonzalo y Rodríguez, Rodolfo. A posteriori error estimates for a Virtual Elements Method for the Steklov eigenvalue problem. En: arXiv - Mathematics - Numerical Analysis, 2016, vol. 1, p. 26.

NATARAJAN, Sundararajan, et al. A new one point quadrature rule over arbitrary star convex polygon/polyhedron. En: arXiv - Mathematics - Numerical Analysis, 2017, p. 1-19.

NATARAJAN, Sundararajan, *et al.* Convergence and accuracy of displacement based finite element formulations over arbitrary polygons: Laplace interpolants, strain smoothing and scaled boundary polygon formulation. En: Finite Elements in Analysis and Design, 2014, vol. 85, p. 101–122.

QUEY, R.; DAWSON, P. y BARBE, F. Large-scale 3D random polycrystals for the finite element method: Generation, meshing and remeshing. En: Computer Methods in Applied Mechanics and Engineering, 2011, vol. 200, no. 17–20, p. 1729–1745.

RAND, Alexander; GILLETTE, Andrew y BAJAJ, Chandrajit. Quadratic serendipity finite elements on polygons using generalized barycentric coordinates. En: The Institute for Computational Engineering and Sciences, 2011, vol. 100, no. 2, p. 130–134.

RÓDENAS, J. *et al.* A recovery-type error estimator for the extended finite element method based on singular + smooth stress field splitting. En: International Journal for Numerical Methods in Engineering, Octubre 2008, vol. 76, no. 4, p. 545–571.

RÓDENAS, J. *et al.* Enhanced error estimator based on a nearly equilibrated moving least squares recovery technique for FEM and XFEM. En: Computational Mechanics, 2013, vol. 52, no. 2, p. 321–344.

RYCROFT, Chris. VORO++: A three-dimensional Voronoi cell library in C++. En: *Chaos*, 2009, vol. 19, no. 4, p. 1–16.

SUKUMAR, N. y MALSCH, E. Recent Advances in the Construction of Polygonal Finite Element Interpolants. En: Archives of Computational Methods in Engineering, 2006, vol. 13, no. 1, p. 129–163.

SUKUMAR, N. y TABARRAEI A. Conforming polygonal finite elements. En: international Journal for Numerical Methods in Engineering, Noviembre 2004, vol. 61, no. 12, p. 2045–2066.

TALISCHI, Cameron, *et al.* PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab. En: Structural and Multidisciplinary Optimization, 2012, vol. 45, no. 3, p. 309–328.

VERFÜRTH, Rüdiger. A posteriori error estimation and adaptive mesh-refinement techniques. En: Journal of Computational and Applied Mathematics, 1994, vol. 50, no. 1, p. 67–83.

WACHSPRESS, Eugene. A rational basis for function approximation. En: IMA Journal of Applied Mathematics, 1973, vol. 11, no. 1, p. 83-104.

WACHSPRESS, Eugene. Barycentric coordinates for polytopes. En: Computers and Mathematics with Applications, 2011, vol. 61, no. 11, p. 3319–3321.

WARREN, Joe, Barycentric Coordinates for Convex Polygons. En: Advances in Computational Mathematics, 1996, vol. 6, no. 2, p. 1–14.

WEIßER, Steffen. Residual error estimate for BEM-based FEM on polygonal meshes. En: Computers and Mathematics with Applications, 2016, vol. 118, no. 4, p. 765–788.

XIAO, Q y KARIHALOO, B. Statically admissible stress recovery for crack problems. En: ICF XI - 11th International Conference on Fracture, Italia, 2005.

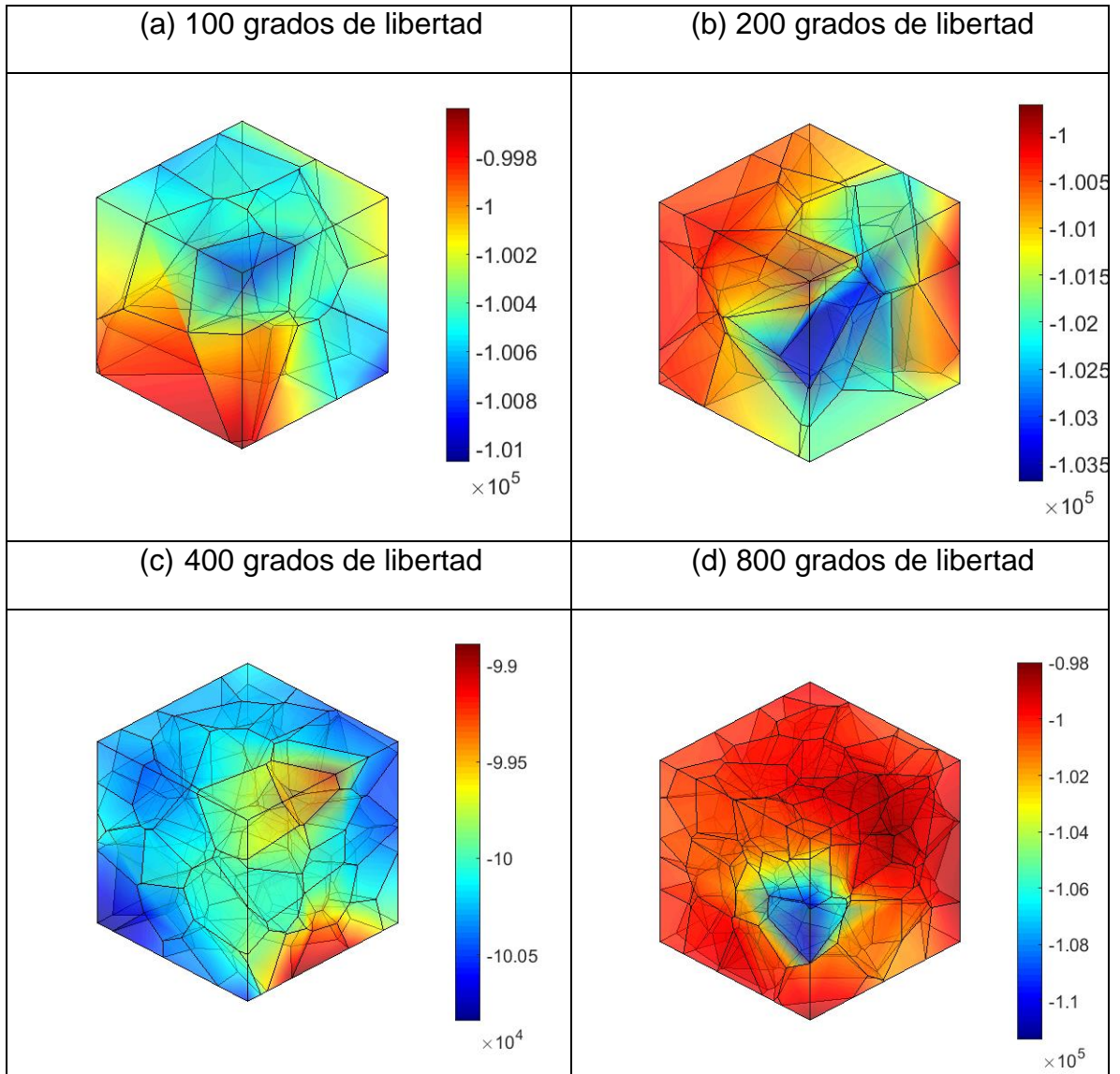
ZIENKIEWICZ, O. y TAYLOR, R. The Finite Element Method: The Basis,. Barcelona: McGraw-Hill, 2000, ISBN 0 7506 5049 4, p 18-37,112-139, 365-398.

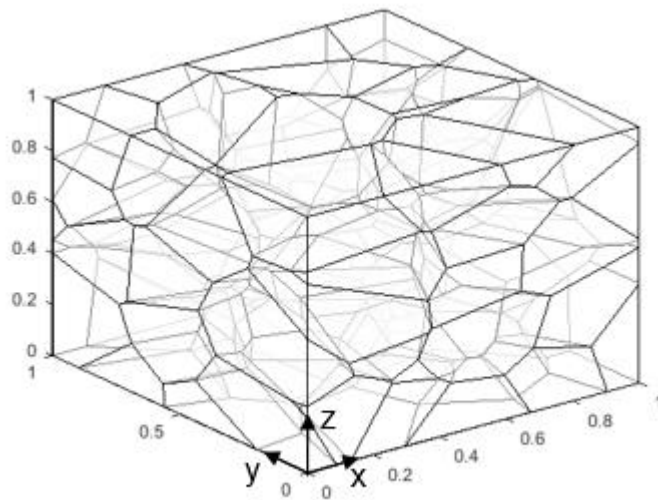
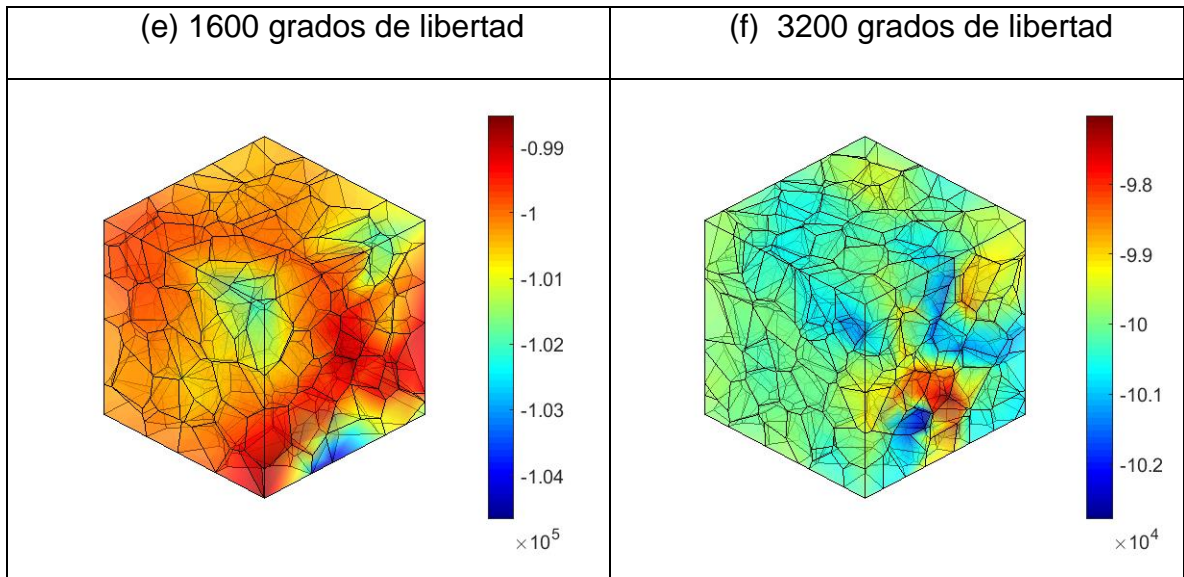
ZIENKIEWICZ, O. y ZHU, J. A simple error estimator and adaptive procedure for practical engineering analysis. En: International Journal for Numerical Methods in Engineering, 1987, vol. 24, no. 2, p. 337–357.

ANEXOS

Anexo A. Simulación de tensión para el cubo.

Tension en direccion Y , para diferentes tamaños de elementos, la presión esta aplicada normalmente a la cara XZ , de forma que la fuerza sobre la cara resulta con la componente negativa Y y las otras dos X y Z nulas.

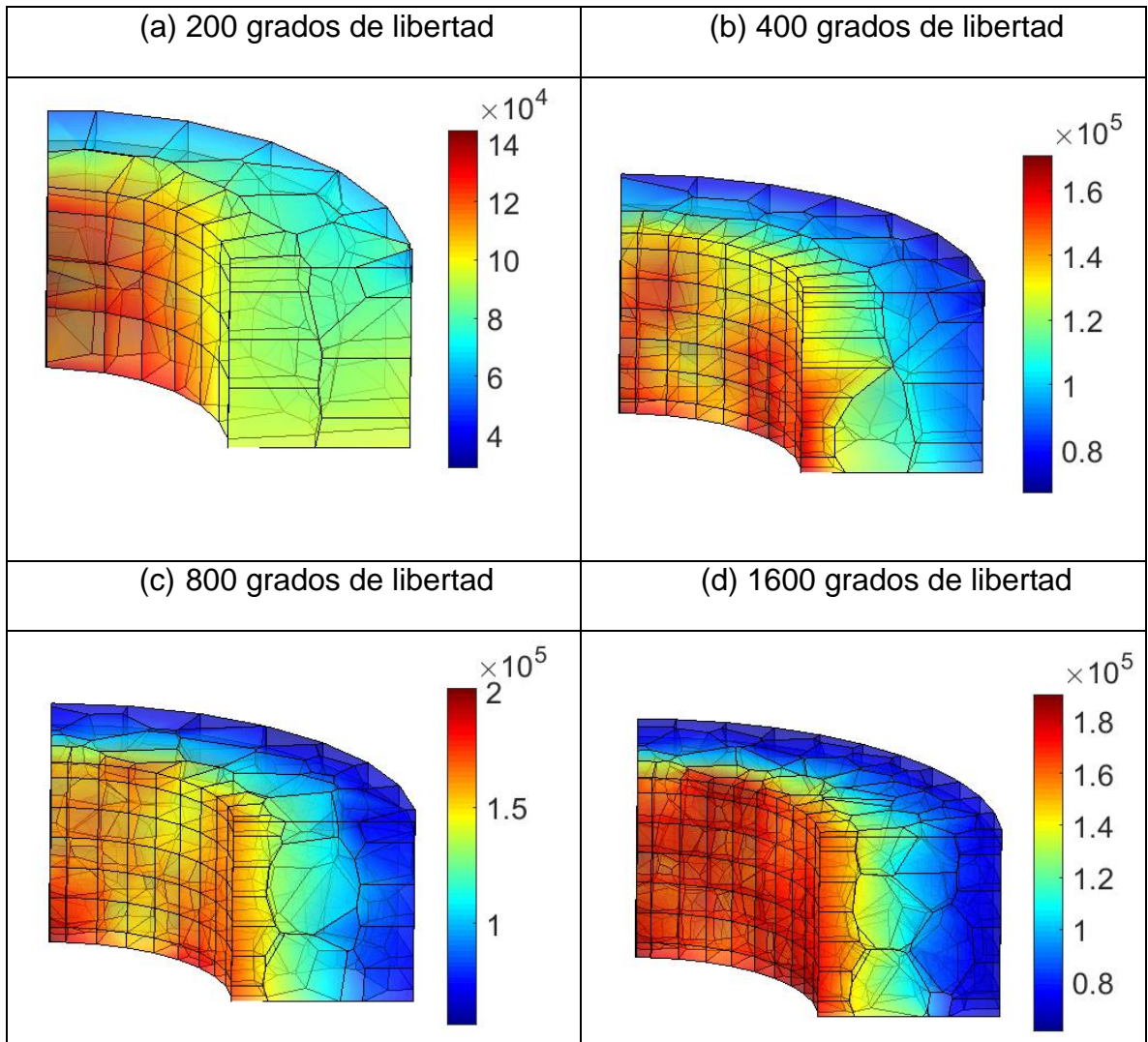


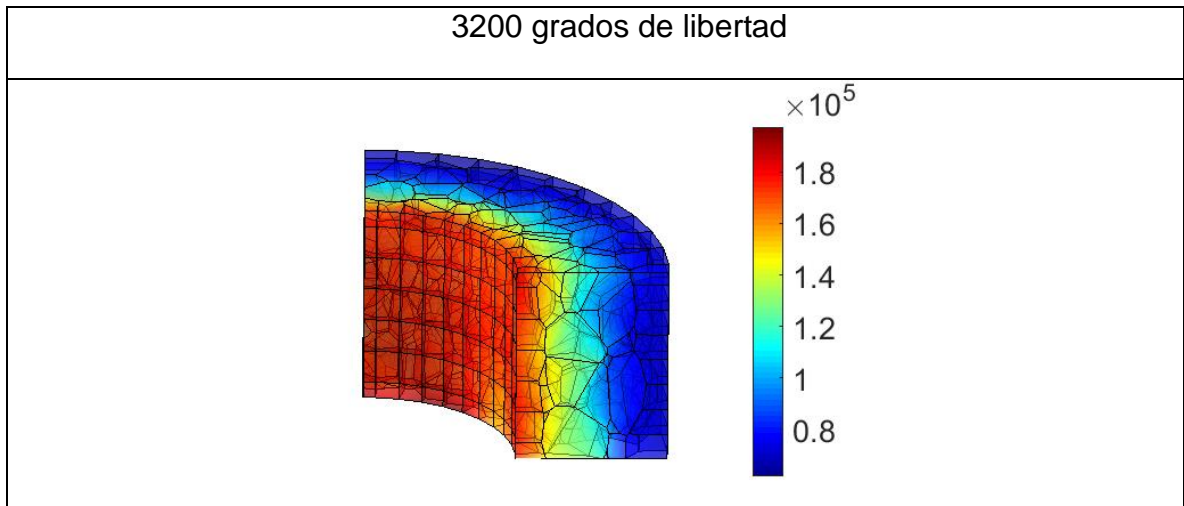


Teselación Voronoi para cubo realizado en el programa Neper.

Anexo B. Simulación de tensión para el cilindro.

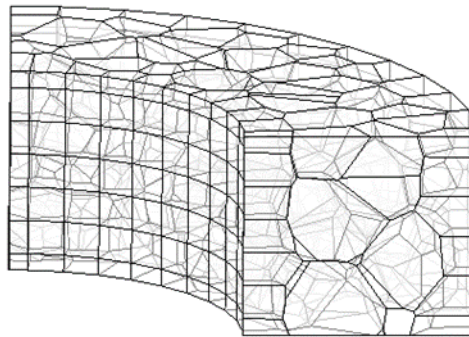
Tensión Von Mises para diferentes grados de libertad. La presión es aplicada en la cara circular interna del cilindro en dirección radial hacia afuera. De modo que la fuerza resultante es radial de igual manera.





A medida que aumentan los grados de libertad la presión se distribuye en la zona de aplicación de una forma más uniforme, teniendo una aproximación real bastante aceptable, además de que su valor de tensión tiende al real. Esto genera confianza a la hora de realizar los cálculos ya que su precisión es óptima.

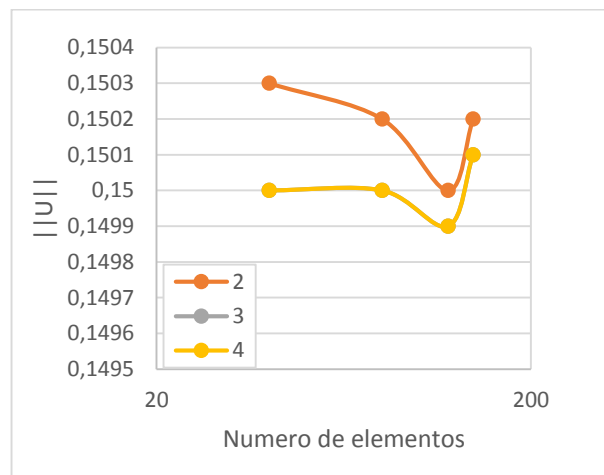
Teselación Voronoi para el cilindro realizado en el programa Vorop++:



Anexo C. Convergencia de la energía de deformación según los grados de cuadratura.

Resultados de energía de deformación para diferentes tipos de cuadraturas con variación en el número de elementos: convergencia

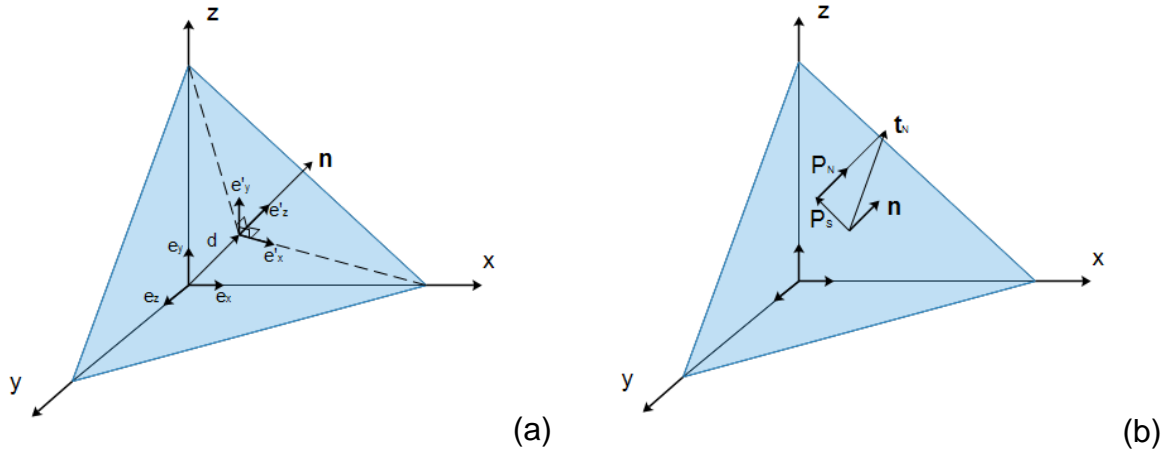
U NDOF	Cuadratura de Gauss			
	1	2	3	4
40	0,2102	0,1503	0,15	0,15
80	0,207	0,1502	0,15	0,15
120	0,2061	0,15	0,1499	0,1499
140	0,2063	0,1502	0,1501	0,1501



Se observa como con cuadraturas de tercero y cuarto orden para diferentes pruebas se presenta el mismo valor de energía de formación, lo cual indica que para ahorrar tiempo de computación es óptimo utilizar una cuadratura de orden 3.

Anexo D. Análisis vectorial para la obtención del vector de fuerza.

Sistemas de referencia para la fuerza en las caras: (a) transformación de vector unitario, (b) aplicación de la fuerza normal al plano.



$$X' = X + X_m * U' * V$$

Para encontrar la ubicación de los puntos de Gauss en la superficie 3D, se establece un nuevo sistema de referencia (SR_1) con vector unitario $V(e'_x, e'_y, e'_z)$, $U(e_x, e_y, e_z)$, se describe el sistema de referencia anterior (SR_2), $X'(x', y', z')$, contiene las nuevas coordenadas en V para el punto $X(x, y, z)$, y $X_m = d$ según la grafica 2 que indica al vector desplazamiento entre los dos SR . Con los valores de cuadratura de Gauss H_{pg} y $G(x, y, z)$ en SR_1 ya se obtiene el vector unitario $\mathbf{n} = (n_x, n_y, n_z)$, para cada punto de Gauss a partir de $G(x, y, z)$, pasando a reemplazar en la ecuación 5.

Anexo E. Código de entrada para Voro++

Para cubo:

El código `random_points.cc` proporciona el mallado Voronoi de un cubo, donde sus dimensiones se pueden modificar en las líneas [11-13]. En la línea 20, el número de partículas se refiere a los puntos centrales de cada elemento, entonces, se puede entender como el número de elementos.

Para compilar se puede desde Linux, Mac OS y Windows usando (cygwin), con las siguientes líneas ejecutadas en el terminal de pc.

- `make random_points.cc`
- `./random_points`

En la carpeta contenedora del código se generan 5 archivos `.custom`, que se debe copiar en la carpeta que se quiera llamar desde la entrada de Matlab, por ejemplo, dentro de cubo y 100Nelem



Code listing – random_points.cc

```
1: // Voronoi calculation example code
2: //
3: // Author   : Chris H. Rycroft (LBL / UC Berkeley)
4: // Email    : chr@alum.mit.edu
5: // Date     : August 30th 2011
6:
7: #include "voro++.hh"
8: using namespace voro;
9:
10: // Set up constants for the container geometry
11: const double x_min=-1,x_max=1;
12: const double y_min=-1,y_max=1;
13: const double z_min=-1,z_max=1;
14: const double cvol=(x_max-x_min)*(y_max-y_min)*(x_max-x_min);
15:
16: // Set up the number of blocks that the container is divided into
17: const int n_x=6,n_y=6,n_z=6;
```

```

18:
19: // Set the number of particles that are going to be randomly
    introduced
20: const int particles=20;
21:
22: // This function returns a random double between 0 and 1
23: double rnd() {return double(rand())/RAND_MAX;}
24:
25: int main() {
26:     int i;
27:     double x,y,z;
28:
29:     // Create a container with the geometry given above, and make
    it
30:     // non-periodic in each of the three coordinates. Allocate
    space for
31:     // eight particles within each computational block
32:     container
con(x_min,x_max,y_min,y_max,z_min,z_max,n_x,n_y,n_z,
33:         false,false,false,8);
34:
35:     // Randomly add particles into the container
36:     for(i=0;i<particles;i++) {
37:         x=x_min+rnd()*(x_max-x_min);
38:         y=y_min+rnd()*(y_max-y_min);
39:         z=z_min+rnd()*(z_max-z_min);
40:         con.put(i,x,y,z);
41:     }
42:
43:     // Sum up the volumes, and check that this matches the
    container volume
44:     double vvol=con.sum_cell_volumes();
45:     printf("Container volume : %g\n"
46:         "Voronoi volume : %g\n"
47:         "Difference : %g\n",cvol,vvol,vvol-cvol);
48:
49:     // Output the particle positions in gnuplot format
50:     con.draw_particles("random_points_p.gnu");
51:
52:     // Output the Voronoi cells in gnuplot format
53:     con.draw_cells_gnuplot("random_points_v.gnu");
48: }

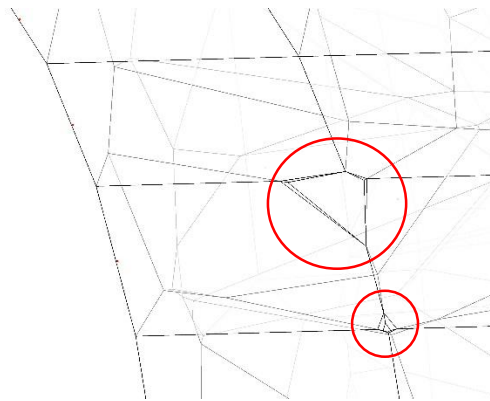
```

Para un cuarto de cilindro hueco:

El código `cylinder_inv.cc` proporciona el mallado Voronoi de un cuarto cilindro hueco, donde sus dimensiones se pueden modificar en las líneas [10-16]. Para este caso hay que tener cuidado en el mallado, al menos en la cara interna, pues fue

necesario limitar las fronteras curvas con una malla de puntos (línea [72-85]), y si durante el mallado se generan defectos dentro de la cara interna, se debe arreglar ya sea modificando la malla de puntos o el número de partículas que se van a colocar (línea.18).

De igual forma que en el cubo, los archivos que se generan .costum, se llevan a las carpetas donde importa los datos Matlab (xxxNelem), por ejemplo, en cilindro 200Nelem.



```

1: // Cylindrical wall example code
2: //
3: // Author   : Chris H. Rycroft (LBL / UC Berkeley)
4: // Email    : chr@alum.mit.edu
5: // Date     : August 30th 2011
6:
7: #include "voropp.h"
8: using namespace voro;
9: // Set up constants for the container geometry
10: const double x_min=0,x_max=7;
11: const double y_min=0,y_max=3;
12: //const double z_min=-6,z_max=6;
13: const double pi=3.1415926535897932384626433832795;
14: const double r_min=3.4,r_max=5.6;
15: const double t_min=0,t_max=90;
16: const double z_min=0,z_max=7;
17: // Set the number of particles that are going to be randomly
   introduced
18: const int particles=10;
19: // This function returns a random double between 0 and 1
20: double rnd() {return double(rand())/RAND_MAX;}
21: // Set the computational grid size
22: const int n_x=6,n_y=6,n_z=6;
23: struct wall_cylinder_inv : public wall {

```

```

24:         public:
25:             wall_cylinder_inv(double xc_, double yc_, double zc_,
26: double xa_, double ya_, double za_, double rc_, int w_id_=-99)
27:                 : w_id(w_id_), xc(xc_), yc(yc_), zc(zc_),
xa(xa_), ya(ya_), za(za_),
28:                 asi(1/(xa_*xa_+ya_*ya_+za_*za_)), rc(rc_) {}
29:             bool point_inside(double x, double y, double z) {
30:                 double xd=x-xc, yd=y-yc, zd=z-zc;
31:                 double pa=(xd*xa+yd*ya+zd*za)*asi;
32:                 xd-=xa*pa; yd-=ya*pa; zd-=za*pa;
33:                 return xd*xd+yd*yd+zd*zd>rc*rc;
34:             }
35:             template<class v_cell>
36:             bool cut_cell_base(v_cell &c, double x, double y,
double z) {
36:                 double xd=x-xc, yd=y-yc, zd=z-zc;
37:                 double pa=(xd*xa+yd*ya+zd*za)*asi;
38:                 xd-=xa*pa; yd-=ya*pa; zd-=za*pa;
29:                 pa=xd*xd+yd*yd+zd*zd;
40:                 if(pa>1e-5) {
41:                     pa=2*(sqrt(pa)*rc-pa);
42:                     return c.nplane(-xd, -yd, -zd, -pa, w_id);
43:                 }
44:                 return true;
45:             }
46:             bool cut_cell(voronoicell &c, double x, double y, double
z) {return cut_cell_base(c, x, y, z);}
47:             bool cut_cell(voronoicell_neighbor &c, double x, double
y, double z) {return cut_cell_base(c, x, y, z);}
48:         private:
                 const int w_id;
                 const double xc, yc, zc, xa, ya, za, asi, rc;
49: };
50:
51: int main() {
52:
53:     // Create a container with the geometry given above, and make
it
54:     // non-periodic in each of the three coordinates. Allocate
space for
55:     // eight particles within each computational block.
56:     container con(x_min, x_max, y_min, y_max, z_min, z_max, n_x, n_y, n_z,
57:                 false, false, false, 8);
58:
59:     // Add a cylindrical wall to the container
60:     wall_cylinder_inv cyl(0, 0, 0, 0, 1, 0, 3);
61:     con.add_wall(cyl);
62:
63:     //wall_cylinder_inv cyl2(0, 0, 0, 0, 1, 0, 6.5);
64:     //con.add_wall(cyl2);
65:     wall_cylinder cyl2(0, 0, 0, 0, 1, 0, 6);
66:     con.add_wall(cyl2);
67:
68:     int i;

```

```

69:     double x,y,z,r,t;
70:
71:     //400E
72:     for(t=0;t<=90;t+=15) for(y=0;y<=3;y+=0.9999) {
73:         r=6;
74:         x=r*cos(t/180*pi);
75:         z=r*sin(t/180*pi);
76:         con.put(i,x,y,z);
77:     };
78:
79:     //400E
80:     for(t=0;t<=90;t+=13) for(y=0;y<=3;y+=0.7999) {
81:         r=3;
82:         x=r*cos(t/180*pi);
83:         z=r*sin(t/180*pi);
84:         con.put(i,x,y,z);
85:     };
86:
87:     for(i=0;i<particles;i++) {
88:         y=y_min+rnd()*(y_max-y_min);
89:         r=r_min+rnd()*(r_max-r_min);
90:         t=t_min+rnd()*(t_max-t_min);
91:         x=r*cos(t/180*pi);
92:         z=r*sin(t/180*pi);
93:         if (con.point_inside(x,y,z)) {
94:             con.put(i,x,y,z);
95:         };
96:     };
97:
98:     // Output the particle positions in gnuplot format
99:     con.draw_particles("cylinder_inv_p.gnu");
100:
101:     // Output the Voronoi cells in gnuplot format
102:     con.draw_cells_gnuplot("cylinder_inv_v.gnu");
103:
104:     // ID and position, more
105:     con.print_custom("%i %q %s %F","packing.custom1");
106:
107:     // more
108:     con.print_custom("%a","packing.custom2");
109:
110:     //more
111:     con.print_custom("%t","packing.custom3");
112:
113:     //more
114:     con.print_custom("%w %g %s","packing.custom4");
115:
116:     //more
117:     con.print_custom("%P","packing.custom5");
118: }

```

Anexo F. Descripción del código de Matlab.

Estimador de error, basado en reconstrucción de tensiones por promediado en Nodos, para mallas arbitrarias(Voronoi) en 3D

```
% Escrito por: Juan Diego Duarte F.  
%             Juan Sebastián Parra H.  
% Escuela de Ing. Mecánica  
% Universidad Industrial de Santander  
  
% *En este trabajo hay códigos realizados por otros autores, que fueron  
% necesarios para completar el código final*
```

Contenido

- Entrada y presentación del código
- Selección del problema.
- Cuerpo del código

Entrada y presentación del código

Ubicación de la carpeta actual y reinicio de variables.

```
close all  
clear; clc  
DataRoot=pwd;  
reset_callruta;
```

Selección del problema

Selección de la malla. El mallado de la pieza solo está dispuesto:

Nombre Caso:

- Cubo
- Cilindro

Numero de Nodos:

- Cubo: 100, 200, 400, 800, 1600, 3200
- Cilindro: 200, 400, 800, 1600, 3200.

```

% *Si se quiere otras mallas, extraer lo datos de voro++*

str=input('Nombre Caso: ','s');
switch str
    case 'cubo'
        fold='graf\ntipos\cubo';
        Nelem=str2num(input('Numero de Nodos: ','s'));
        type=1;
    case 'cilindro'
        fold='graf\ntipos\cilindro';
        Nelem=str2num(input('Numero de Nodos: ','s'));
        type=2;
        global r1 r2
        r1=3;
        r2=6;
end

```

Cuerpo del código

```

folder=[DataRoot,bar, fold];
voro3D;

```

Voro3D

Estimación del error de un problema elástico lineal, su mallado se consigue de software Neper, la técnica de reconstrucción de tensiones es promediado en nodos.

Contenido

- Importar Geometría
- Propiedades del material
- Stiffness computation – loop over elements.
- Processing
- Calculo, solución exacta
- Post-processing
- Reconstrucción de la tensión
- Calculo, estimación de error
- Revisión de la efectividad Local

Importar Geometría

En graphdatos se toman los archivos de voro++, donde se contienen los datos del mallado de la pieza, y son organizados de forma que se puedan trabajar

```
tic
path(path, './graf')

graphdatos;
Dmax = zeros(numelem,1);
%Max diámetro del elemento
for el = 1:numelem
    econ = Element{el};
    V = Node(econ,:);
    [dmax] = maxpolydia(V);
    Dmax(el,1) = dmax;
end
h = max(Dmax)
numnode
numelem
```

Propiedades del material

Se definen el tipo de material junto a sus propiedades: Material isótropo, Modulo de Young y coeficiente de poisson.

```
path(path, './functions_cantilever')
path(path, './polymesh')

E = 200e9;
nu = 0.3;
C= E/((1+nu)*(1-2*nu))* ...
    [(1-nu) nu nu 0 0 0;
    nu (1-nu) nu 0 0 0;
    nu nu (1-nu) 0 0 0;
    0 0 0 (1-2*nu)/2 0 0;
    0 0 0 0 (1-2*nu)/2 0;
    0 0 0 0 0 (1-2*nu)/2];
```

Stiffness computation - Loop over elements

Código Natarajan Sundar. Cantilever 3D, modificado para voro++.

Calculo de la matriz de rigidez.

```

ndof = 3;
totalUnknown= numnode*ndof ;
kmat = sparse(totalUnknown,totalUnknown);
vol = zeros(1,numelem);

for iel = 1:numelem
    clear un g ncoord
    econ = Element{iel};
    ncoord = Node(econ,:);
    nn = length(econ);
    face = faces{1, iel}';
    quadorder = 3;
    [un,g,nfaces] = getUnitNormg(face,ncoord);
    [G,W,MatN] = GetgaussPolyhedraMod_mod(ncoord,quadorder,face);
    corrVarg;

    Pgauss.G{1,iel}=G;
    Pgauss.W{1,iel}=W;
    Pgauss.N{1,iel}=MatN;

    clear gindex
    for in = 1:length(econ)
        gindex(ndof*in-2) = ndof*econ(in)-2;
        gindex(ndof*in-1) = ndof*econ(in)-1;
        gindex(ndof*in) = ndof*econ(in);
    end

    for igp = 1:size(W,1)
        pt = G(igp,:);
        [phi,dphi] = wachspress3d(ncoord,g,un,pt);

        B = zeros(6,ndof*nn);
        B(1,1:ndof:ndof*nn) = dphi(:,1) ;
        B(2,2:ndof:ndof*nn) = dphi(:,2) ;
        B(3,3:ndof:ndof*nn) = dphi(:,3) ;

        B(4,1:ndof:ndof*nn) = dphi(:,2) ;
        B(4,2:ndof:ndof*nn) = dphi(:,1) ;

        B(5,2:ndof:ndof*nn) = dphi(:,3) ;
        B(5,3:ndof:ndof*nn) = dphi(:,2) ;

        B(6,1:ndof:ndof*nn) = dphi(:,3) ;
        B(6,3:ndof:ndof*nn) = dphi(:,1) ;

        kmat(gindex,gindex) = kmat(gindex,gindex) +B'*C*B*W(igp);

    % Volumen

```

```

        vol(iel) = vol(iel) + W(igp);

    % Matriz DB
    CBe= C*B;
    CB{1,iel}{igp,1}= CBe;
end

end

V_FEM=sum(vol)

```

Processing

En `problemCub` y `problemCil`, contienen los códigos donde se aplican las fuerzas y las restricciones al problema y ajuste de la matriz `Kmat`.

```

global K F
K=kmat;

% Cargar el modelo con fuerzas y restricciones
if type==1      % Cubo
    problemCub
elseif type==2  % Cilindro
    problemCil
end

kk=full(K);FF=full(F);

```

Solve

Solución FEM, para establecer los desplazamientos (Sol). Calculo de las tensiones por elemento en los puntos de gauss, tensiones nodales, y sus respectivas graficas.

```

Sol= K\F;

% desplazamientos u, grafica
uu=full(Sol);
Untitled3;

% figure: desplazamiento
[u] = desp_gr(uu,numnode,Node,faces);
% Calculo de Esfuerzos en elementos

```

```

S = cal_s(numelem,Element,uu,CB);
% Calculo de Esfuerzos en nodos (promediado directo)
SN = cal_se(numelem,Element,S,numnode);
% % figure: Esfuerzos en direccion dSig
% Descomentar que grafica se quiere mostrar
def='Fem';
% dSig='x';
% graf_s(SN,faceess,u,dSig,def);
% dSig='y';
% graf_s(SN,faceess,u,dSig,def);
% dSig='SVM';
% graf_s(SN,faceess,u,dSig,def);

```

Calculo de solución exacta

Para el cubo la tensión será la misma presión, pues la fuerza es normal a la cara, y para el cilindro la solución se describe en ExactStressMod, problema de deformación plana.

```

if type==1 % Cubo
    eSigExGP=[0 -100000 0 0 0 0]';
    SNEx=ones(1,numnode) '*[0 -100000 0 0 0 0];
elseif type==2 % Cilindro
    for iel = 1:numelem
        clear un g ncoord Sigpg sr st
        G = []; WG = [];

        G=Pgauss.G{1,iel};
        W=Pgauss.W{1,iel};
        Problem=1; % Stress exact cylindre 3d
        for igp = 1:size(W,1)
            Xpg=G(igp,1);Zpg=G(igp,3);
            [Sigpg(:,igp) sr(igp)
st(igp)]=ExactStressMod(Xpg,Zpg,Problem,C,E,nu,r1,r2,Carga);
            end
            SigExpG{1,iel}=Sigpg;
            Sigrpg{1,iel}=sr;
            Sigtpg{1,iel}=st;
        end
        % Calculo de Esfuerzos en nodos(promediado directo)
        SNEx = cal_se(numelem,Element,SigExpG,numnode);
        % figure: Esfuerzos en direccion dSig
        % Descomentar que grafica se quiere mostrar
        dSig='SVM';
        % def='Exacta';
        % graf_s(SNEx,faceess,Node,dSig,def);
        % dSig='x';

```

```

    % graf_s(SNEx, facess, Node, dSig, def);
    % dSig='y';
    % graf_s(SNEx, facess, Node, dSig, def);
end

```

Post-processing

Calculo de la energía de deformación y el error FEM.

```

FENorm = zeros(1,numelem);
ErrNormEx = zeros(1,numelem);
NormEx = zeros(1,numelem);
invC=inv(C);
Problem=1; % Stress exact cylindre 3d

%error exacto
for iel = 1:numelem
    clear un g ncoord
    G = []; WG = [];
    %
    G=Pgauss.G{1,iel};
    W=Pgauss.W{1,iel};
    %
    eFieldFE_Elm=S{1,iel};
    for igp = 1:size(W,1)
        Xpg=G(igp,1); Zpg=G(igp,3);

        if type==1 % Cubo
            eSigExGP=[0 -100000 0 0 0 0]';
        elseif type==2 % Cilindro
            eSigExGP=ExactStressMod(Xpg,Zpg,Problem,C,E,nu,r1,r2,Carga);
        end
        eFieldFEGP=eFieldFE_Elm(:,igp);
        % FENorm(iel) = FENorm(iel) + eSigExGP' * invC * eSigExGP *W(igp);
        FENorm(iel) = FENorm(iel) + eFieldFEGP' * invC * eFieldFEGP *W(igp);
        eIncrSigEx=eSigExGP-eFieldFEGP;
        ErrNormEx(iel) = ErrNormEx(iel) + eIncrSigEx' * invC * eIncrSigEx *
        W(igp);
        NormEx(iel) = NormEx(iel) + eSigExGP' * invC * eSigExGP *
        W(igp);
    end
end

% exacta
FENorm2ex = NormEx;
NormEx = sqrt(NormEx);

```

```

GlobNormex = sqrt(sum(FENorm2ex))
UFemex = 1/2*sum(FENorm2ex)
% estimado
FENorm2 = FENorm;
FENorm = sqrt(FENorm); SFENorm2=sum(FENorm2);
GlobNorm = sqrt(SFENorm2)
UFem = 1/2*SFENorm2 % 1/2*int(sigma^T*D^-1*sigma*ddom)
% error FEM
ErrNormEx2=ErrNormEx;
ErrNormEx=sqrt(ErrNormEx); SErrNormEx2=sum(ErrNormEx2);
GlobNormEx=sqrt(SErrNormEx2)
ErrorFem = 1/2*SErrNormEx2 % 1/2*int(Dsigma^T*D^-1*Dsigma*ddom)

```

Post-Processing

En SigRecovMod, se hace la reconstrucción de la tensión, por el método promediado en nodos, se calcula sus valores nodales y se realizan sus respectivas graficas.

```

SigRecovModVoro; % promediado en nodos SUM(N*SN)pg
% Calculo de Esfuerzos en nodos - alisado
SNRec = cal_se(numelem,Element,SigRec,numnode);
% figure: Esfuerzos en direccion dSig
% Descomentar que grafica se quiere mostrar
def='Rec.';
% dSig='SVM';
% graf_s(SNRec,faceess,u,dSig,def);
% dSig='x';
% graf_s(SNRec,faceess,u,dSig,def);
% dSig='y';
% graf_s(SNRec,faceess,u,dSig,def);

```

Calculo de la Estimación del error

A partir de la tensión reconstruida, se calcula el error estimado, comparando con el error FEM que se calculó anteriormente.

```

FENormRec = zeros(1,numelem);
ErrNormRec = zeros(1,numelem);
NormRec = zeros(1,numelem);
for iel = 1:numelem
    clear un g ncoord
    econ = Element{iel};
    ncoord = Node(econ,:);

```

```

G = []; WG = [];
G=Pgauss.G{1,iel};
W=Pgauss.W{1,iel};

eFieldRec_Elm=SigRec{1,iel}; % Rec.
eFieldFE_Elm=S{1,iel};      % FEM

for igp = 1:size(W,1)
    eFieldFEGP=eFieldFE_Elm(:,igp);
    eSigRecGP=eFieldRec_Elm(:,igp);
%    FENorm(iel) = FENorm(iel) + eSigExGP' * invC * eSigExGP *W(igp);
    FENormRec(iel) = FENormRec(iel) + eFieldFEGP' * invC * eFieldFEGP
*W(igp);
    eIncrSigRec=eSigRecGP-eFieldFEGP;
    ErrNormRec(iel) = ErrNormRec(iel) + eIncrSigRec' * invC * eIncrSigRec *
W(igp);
    NormRec(iel) = NormRec(iel) + eSigRecGP' * invC * eSigRecGP *
W(igp);
end
end

Norm2Rec = NormRec;
NormRec = sqrt(NormRec);SNorm2Rec=sum(Norm2Rec);
GlobNormRec = sqrt(SNorm2Rec)
UFemRec = 1/2*SNorm2Rec % 1/2*int(sigma^T*D^-1*sigma*ddom)

ErrNormRec2=ErrNormRec;
ErrNormRec=sqrt(ErrNormRec);SErrNormRec2=sum(ErrNormRec2);
GlobErrNormRec=sqrt(SErrNormRec2)
ErrorFemRec = 1/2*SErrNormRec2 % 1/2*int(Dsigma^T*D^-1*Dsigma*ddom)

```

Revisión de la efectividad Local

Finalmente, se calcula la efectividad Global, y en `pruebagragefeVoro`, se determina la efectividad a nivel de elemento para ser graficada en la pieza.

```

[UFem;UFemRec;UFemex]
ef2=ErrorFemRec/ErrorFem

% Efectividad Local
Efect=ErrNormRec./ErrNormEx;

pruebagragefeVoro;
toc

```