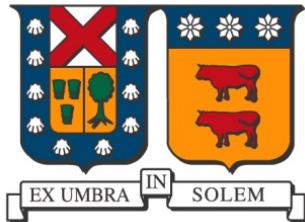


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“EXTENSIÓN DE REFINAMIENTO EN MALLAS
TRIDIMENSIONALES TIPO OCTREE PARA EVITAR
ELEMENTOS INVÁLIDOS”

PEDRO DONOSO AGUILERA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Claudio Lobos Yáñez
Profesor Correferente: Roberto León Vasquez

Agosto - 2024

Resumen

Resumen— Evitar la invalidación de mallas Octree con elementos mixtos en el proceso de refinamiento es bastante importante e incluso puede ser crítico al utilizarse en áreas que requieren cierto nivel de precisión en la representación como la medicina, es por esto que se busca generar la representación más fidedigna posible. En la presente memoria se diseñará una alternativa a la propuesta existente en la literatura [Daines y Lobos, 2018] que nos permita reparar dichos elementos inválidos, además de medir la calidad de los elementos que componen la malla y validarla, es decir, asegurar que todos los elementos resultantes son válidos.

Palabras Clave— Malla geométrica, Octree, Simulación, Refinamiento, Patrones de transición.

Abstract

Abstract— Avoiding the invalidation of Octree meshes with mixed elements in the refinement process is quite important and can even be critical when used in areas that require a certain level of accuracy in the representation such as medicine, which is why we seek to generate the most reliable representation possible. In this report we will design an alternative to the existing proposal in the literature [Daines y Lobos, 2018] that allows us to repair such invalid elements, in addition to measuring the quality of the elements that make up the mesh and validate it, i.e., ensure that all the resulting elements are valid.

Keywords— Geometric meshing, Octree, Simulation, Refinement, Transition patterns.

Glosario

RL: Nivel de refinamiento de un octante.

RA: Relación de Aspecto, medida de calidad de un elemento geométrico, relación entre su borde más corto y el más largo.

Elemento: Elementos geométricos básicos, hexaedro, prisma, pirámide y tetraedro.

Jens: Element Normalized Scaled Jacobian.

Índice de Contenidos

Resumen	I
Abstract	I
1 Glosario	II
Índice de Figuras	v
Índice de Tablas	v
Introducción	1
Capítulo 2: Definición del problema	2
2.1 Contexto	2
2.2 Mallas geométricas	2
2.3 Mallas Octree mixtas	2
2.4 Actualidad	5
2.5 Lineamiento	5
2.6 Objetivos	6
2.6.1 Objetivo general	6
2.6.2 Objetivos específicos	6
Capítulo 3: Marco conceptual	7
3.1 Medida de calidad para elementos en Mallas Mixtas	7
3.2 Malla inicial	9
3.3 Herramientas	10
3.3.1 Mixed-element mesh generator	10
3.3.2 Mixed-element J_{ENS} statistics generator	11
3.3.3 ParaView	12
3.4 Estructura de datos	13
3.4.1 Acercamiento al algoritmo de Mixed-element mesh generator	14
Capítulo 4: Propuesta de solución	16
4.1 Comprensión del problema	16
4.1.1 Estudio preliminar	16
4.1.2 Comprensión de elementos inválidos	18
4.1.3 Análisis de la vecindad	18
4.2 Propuesta	19
4.2.1 Generación de malla inicial.	20
4.2.2 Identificación y etiquetado de Octantes.	22
4.2.3 Refinamiento de Octantes Identificados.	23
4.2.4 Ciclo Iterativo de Evaluación y Refinamiento.	24

Capítulo 5: Validación de la solución	26
5.1 Stack de tecnologías	26
5.2 Proceso de análisis	26
5.3 Ejecución	28
5.3.1 Malla de corteza cerebral	28
5.3.2 Malla de Moai	28
5.3.3 Malla de paladar	29
5.3.4 Malla de coxis	29
5.4 Resultados	31
5.4.1 Ajuste de la cantidad de iteraciones	31
5.5 Análisis de los resultados	33
5.5.1 Análisis de la tasa de reducción de la cantidad de Elementos por refinar en cada iteración	33
5.5.2 Análisis del efecto en los Elementos adyacentes	35
5.5.3 Análisis de calidad frente al Algoritmo de proyección de nodos.	37
Capítulo 6: Conclusiones	42
ANEXOS	43
Referencias Bibliográficas	44

Índice de Figuras

1	Elementos básicos	3
2	Malla Octree parcialmente refinada.	4
3	Patrón de transición aplicado a tres octantes consecutivos no conformes.	4
4	Ejemplos de buena y mala configuración malla Octree.	5
5	Muestra frontal y lateral de la malla inicial con refinamiento global de nivel 5 y refinamiento en superficie prismática regular de nivel 7. Fuente: Elaboración propia.	10
6	Interfaz de software Paraview, se muestra una sección de la malla inicial con dos elementos seleccionados. Fuente: Elaboración propia.	13
7	Diagrama entidad-relación de generador de mallas. Fuente: Elaboración propia.	14
8	Zona frontal de la malla donde se presentan los elementos inválidos (verde).	17
9	Zona a analizar para evidenciar elementos de mala calidad, esta zona se extraejo de la representación de la malla con un nivel 5 de refinamiento general y nivel 7 de refinamiento en la superficie entregada. En la imagen de la izquierda se muestra una visión general de la superficie de la malla demarcada por elementos con vértices azules y la zona a estudiar de color celeste. En la imagen de la derecha, la zona a estudiar y un par de elementos de mala calidad destacados con color verde.	18
10	Perspectivas de vecindario (gris) donde se encuentra un par de elementos inválidos (verde).	19
11	Diferentes localidades de refinamiento en corteza cerebral.	29
12	Diferentes localidades de refinamiento en Moai.	30
13	Representación Paladar con refinamiento en zona superior.	31
14	Representación Coxis sin refinamiento local.	32
16	Histograma agrupado por intervalo para todos los casos.	36
21	Regiones de interés usados en instancias de corteza. De izquierda a derecha se identificarán como Región 0 a 6. En (a) se encuentra la referencia utilizada en [Daines y Lobos, 2018], en (b) homologadas en la práctica.	38
15	Ajuste exponencial de E^0 para todos los casos.	45
17	Histograma agrupado por intervalo para los dos casos de la malla Moai.	46
18	Histograma agrupado por intervalo para los dos casos de la malla Corteza.	47
19	Histograma agrupado por intervalo para los dos casos de la malla Paladar.	48
20	Histograma agrupado por intervalo para los dos casos de la malla Coxis.	49

Índice de Tablas

1	E^0 en cada iteración para los diferentes casos.	33
2	Cocientes entre cantidades de Elementos con $J_{ENS} \leq 0$ para los diferentes casos.	34
3	Promedio geométrico de los cocientes de cada caso.	35

4	Comparación de calidad de los resultados entre los algoritmos original, proyección de nodos y refinamiento de octantes.	39
5	Comparación en tiempo y calidad entre los algoritmos original, proyección de nodos y refinamiento de octantes, con threshold 0,03.	40

Introducción

La simulación asistida por computación es una herramienta muy utilizada en distintas áreas de la ciencia que tiene como objetivo describir de forma más precisa la realidad. En la medicina por ejemplo se utiliza la información obtenida de tomografías computarizadas o imágenes de resonancia magnética para obtener una malla de puntos o nodos que nos ayudará a representar virtualmente el objeto escaneado, a través de geometrías como hexaedros, prismas, pirámides, tetraedros o un grupo de estos, a esto se le denomina malla geométrica. Una malla puede ser utilizada para simulación o visualización, en el primer caso los nodos internos no son utilizados, se utilizan solo mallas superficiales, es decir, mallas conformadas generalmente por triángulos o paralelogramos que representan, como dice el nombre, solo el exterior de algún objeto a modelar, la aplicación más conocida de este tipo de mallas es en la industria del entretenimiento, animación y videojuegos. Las mallas volumétricas en su contra parte utilizan los nodos internos y elementos invisibles de la malla. La mayor aplicación es la simulación, en especial el modelamiento de fenómenos físicos para el análisis estructural.

En la presente memoria se busca integrar una etapa a posteriori de la generación de la malla Octree mixta para evitar elementos inválidos, ubicarlos y refinarlos, con el objetivo único de lograr el nivel de refinamiento requerido manteniendo la conformidad de la malla.

La estructura de esta memoria consta de 5 capítulos. En el *Capítulo 1: Definición del problema*, se detalla el contexto desde donde emerge la problemática, se explica el concepto de malla geométrica y cómo se relaciona con las simulaciones computarizadas, y se menciona el trabajo de investigación existente que aborda la misma problemática y con la que se realizará la comparación. Además, se establece el objetivo general y los objetivos específicos, así como el alcance de la solución propuesta. En el *Capítulo 2: Marco conceptual*, se hace una contextualización del tipo de malla a analizar y la problemática a abordar, se revisa en términos generales el algoritmo trabajado en [Daines y Lobos, 2018]. Por su parte en el *Capítulo 3: Propuesta de solución* se presenta el algoritmo propuesto, así como una breve introducción y explicación de la herramienta de software utilizada para construir las mallas y refinarlas. En el *Capítulo 4: Validación de la solución*, se muestran y analizan las pruebas realizadas, y se realiza la comparación con el trabajo existente [Daines y Lobos, 2018]. Finalmente, en el *Capítulo 5: Conclusiones*, se exponen las conclusiones a las cuales se llegó en base a la comparación de los algoritmos expuestos y se proponen posibles trabajos a futuro de acuerdo a las dificultades y facilidades encontradas en el proceso de validación.

CAPÍTULO 2

Definición del problema

2.1. Contexto

Para realizar una simulación computarizada de un objeto, es necesario contar con una función capaz de describirlo. Cuando la geometría de éste es compleja, generalmente es imposible encontrar dicha función y es necesario aproximarla a través una composición basada en geometrías simples como hexaedros, prismas, pirámides, etc. Esta aproximación mediante elementos geométricos recibe el nombre de malla geométrica. Existen diversos tipos de mallas geométricas, que se clasifican en dos grandes grupos, estructuradas y no estructuradas, en este trabajo nos enfocaremos en un tipo especial de malla estructurada.

2.2. Mallas geométricas

Una malla geométrica es una colección de elementos geométricos utilizados como representación computacional para el modelado tridimensional y/o animación.

Específicamente, existen dos tipos de mallas geométricas relacionadas a su alcance de representación, malla de superficie, que como dice su nombre representa sólo la superficie del dominio y donde los nodos internos no son utilizados, generalmente conformadas sólo por triángulos o paralelogramos, por otra parte, el otro tipo de malla es del tipo volumétrica, utilizada comúnmente para simulación, está compuesta generalmente por elementos geométricos volumétricos como tetraedros, hexaedros, prismas, etc. Este tipo de malla utiliza los nodos internos para análisis estructural, deformaciones, fracturas, el efecto del calor, entre otras.

En esta memoria se utilizará la estructura jerárquica regular denominada Octree y que presentará diversos tipos de elementos geométricos, malla Octree mixta, para representar y refinar las mallas geométricas volumétricas.

En ciertas aplicaciones de mallas geométricas como por ejemplo en el contexto de simulaciones biomédicas, como la técnica usada en [Lobos y González, 2015] para mallados en tiempo real en el ámbito biomédico, se requiere realizar un acercamiento a una o más zonas del objeto simulado a estudiar, este proceso se le denomina refinamiento.

2.3. Mallas Octree mixtas

La técnica Octree es una de las más utilizadas para concentrar nodos en una región determinada de una malla. Cada nodo de un Octree se denomina octante. La idea básica consiste en dividir un octante en varios octantes nuevos de forma recursiva.

El octante es una estructura de datos jerárquica y recursiva definida como un árbol de datos en la que cada nodo de éste se divide en un número constante de nodos hijos, normalmente en 8 o 27 hexaedros, estos nodos serán los encargados de generar figuras geométricas como hexaedros, prismas, pirámides y tetraedros que son mostradas en la Figura 1. Cuando se utiliza la división en 27, es posible generar una malla hexaédrica pura. Sin embargo, cuando

se utiliza el proceso 8-split, las transiciones entre regiones finas y gruesas sólo pueden gestionarse introduciendo diferentes tipos de elementos, a esto se le denomina malla Octree mixta.

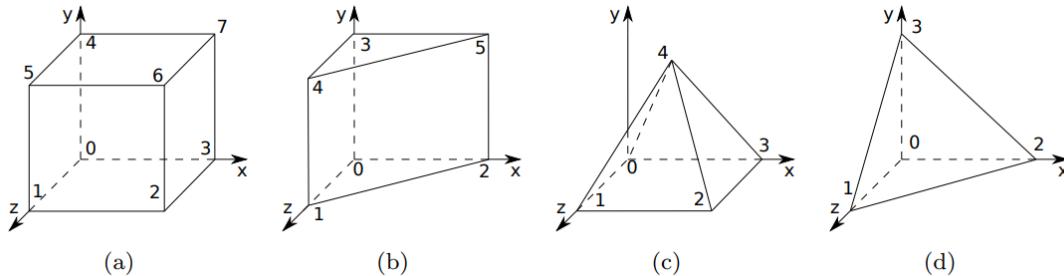


Figura 1: Elementos básicos
 (a) hexaedro, (b) prisma (cuña), (c) pirámide y (d) tetraedro.
 Fuente: [González y Lobos, 2014]

Supongamos que Ω es el dominio de entrada para el que se requiere generar una malla. La mayoría de algoritmos de mallado basados en octantes encapsulan el dominio Ω en un octante principal o primario y, a continuación, proceden a subdividir recursivamente los octantes hasta que se cumple una determinada restricción, generalmente un delta de error de representación.

El número de subdivisiones recursivas realizadas sobre un octante se denomina nivel de refinamiento (RL). Algunos algoritmos ofrecen la posibilidad de definir distintas regiones de refinamiento sobre Ω . Para cada región, puede definirse un RL mínimo y un octante puede pertenecer a varias regiones. Cuando un octante se divide y algunos de sus hijos se sitúan completamente fuera del dominio Ω , su proceso de refinamiento se detiene y se eliminan de la representación.

Un Octree equilibrado significa que dos octantes adyacentes no pueden tener una diferencia de nivel de refinamiento superior a uno, como se menciona en [Daines y Lobos, 2018]. Una malla basada en octantes suele requerir un Octree de este tipo. Para ello, algunos octantes se refinan más allá de su RL original. En la Figura 2 se muestra un ejemplo análogo en 2D para ilustrar el proceso de división en 8 hexaedros.

La tarea principal de esta investigación es el refinamiento de las mallas, es decir, aumentar la densidad de nodos que conforman la malla, tanto de forma parcial como general, para así aumentar a su vez la definición de la representación. Sin embargo este aumento en la cantidad de nodos no debe ser uniforme en toda la malla ya que afectaría considerablemente el tiempo de ejecución de las simulaciones. Es por esto que algunas técnicas de generación de mallas utilizan refinamiento parcial [Lobos y González, 2015], donde son seleccionadas ciertas regiones de la malla en las que se quiere mayor definición y solo dichas zonas de interés son refinadas. En el caso de la Figura 2, una malla en 2D de refinamiento parcial, la región de interés es el borde del dominio identificado con el color gris.

En una malla al haber distintos niveles de refinamiento, se pueden formar mallas no conformes.

Un Octree equilibrado no es conforme cuando hay múltiples RL y no se respeta la regla de

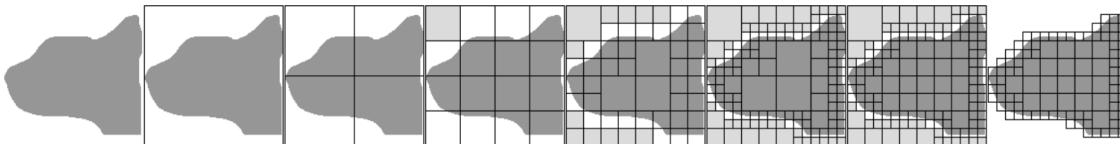


Figura 2: Malla Octree parcialmente refinada.

Ejemplo en 2D de la generación de una malla octree equilibrada utilizando un algoritmo basado en Octree. Con nivel de refinamiento RL 4 en la región Ω demarcada con gris. El resto de la malla no tiene definido un RL mínimo, por lo que los octantes que no intersectan con el límite de Ω tienen su refinamiento detenido.

Fuente: [Lobos y González, 2015]

los RL adyacentes. Para mantener la continuidad de la malla, debe aplicarse un patrón de transición sobre los octantes con vecinos de diferentes RL . Un patrón de transición sustituirá el hexaedro no conforme por diferentes tipos de elementos, de modo que se garantice que la malla de salida sea conforme. En la Figura 3 se muestra un ejemplo de patrón de transición.

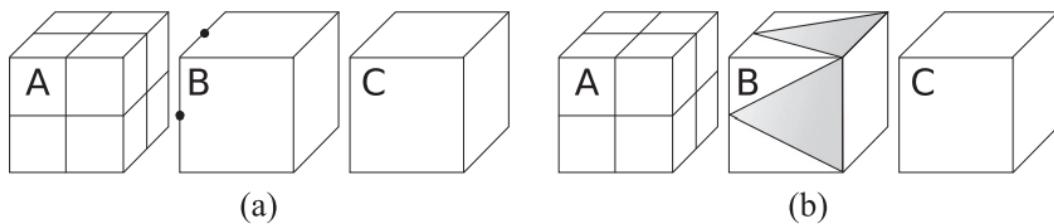


Figura 3: Patrón de transición aplicado a tres octantes consecutivos no conformes.

(a) Octante B es el único elemento que requiere un patrón de transición. (b) Malla conforme, topológicamente correcta, resultante de la transición.

Fuente: [Lobos y González, 2015].

Una malla es conforme cuando es topológicamente correcta, es decir, todos los elementos que la conforman tienen sus vértices y aristas de los bordes y caras consistentes para todos los elementos adyacentes. Un claro ejemplo se puede ver en la Figura 4 donde se muestran dos elementos consecutivos que mediante flechas de dos puntas evidencia la contraposición de vértices y aristas. En el caso del lado derecho (b), existe una arista, la que se posiciona transversalmente en la cara expuesta y que une dos vértices opuestos. En el caso de la izquierda (a), se muestra un caso de Elementos bien estructurados, donde todos sus vértices y aristas son consistentes.

Para construir una malla conforme con refinamiento parcial, se puede hacer uso de elementos especiales que forman transiciones entre regiones con distinto nivel de refinamiento. Este conjunto de elementos se le denomina patrones de transición. En la Figura 3 se puede observar que el elemento B constituye un patrón de transición entre zonas con distinto nivel de refinamiento, en donde el elemento A presenta más refinamiento que el elemento C.

La técnica de generación de mallas a utilizar en esta memoria será la técnica que utiliza elementos mixtos para construir patrones de transición. Dichos elementos son tetraedros, he-

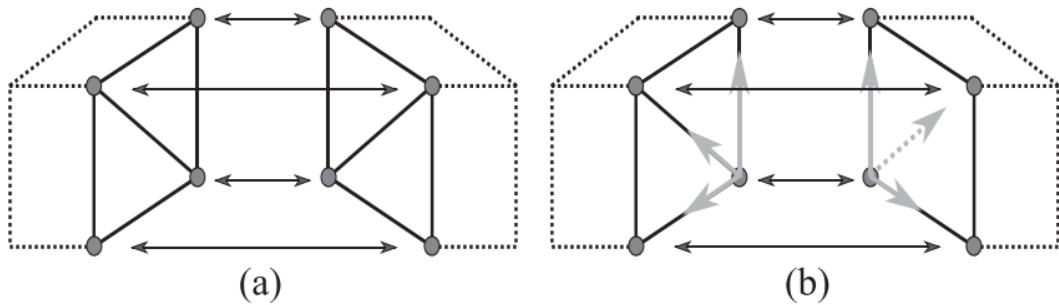


Figura 4: Ejemplos de buena y mala configuración malla Octree.

(a) Elementos bien estructurados. (b) Elementos mal estructurados, uno de sus aristas no es consistente.

Fuente: [Lobos y González, 2015].

xaedros, prismas y pirámides de base cuadrada. Todos los patrones existentes fueron diseñados y presentados en [González y Lobos, 2014], patrones de transición para mallas Octree, considerando todas las combinaciones posibles cuando un Octante y sus vecinos difieren, como máximo, de un nivel de refinamiento, evitando así, refinar el Octante cada vez que el patrón no se encuentra en su stack de patrones, como lo hacen otras técnicas. Además, también hay técnicas que los utilizan para la representación de los bordes en la malla, definiendo un conjunto de elementos, llamados patrones de superficie, que reemplazan los hexaedros encontrados en los bordes.

Algunas mallas de refinamiento parcial pueden verse afectadas negativamente por transiciones entre regiones de diferente nivel de refinamiento en la superficie, llegando muchas veces a invalidar ciertos elementos involucrados.

2.4. Actualidad

Actualmente existe un estudio [Daines y Lobos, 2018] que consiste en reparar elementos inválidos en la superficie de mallas tipo Octree con elementos mixtos y refinamiento parcial. En este estudio se propone una técnica de proyección de nodos para reparar aquellos elementos de los bordes en regiones de transición de la malla, esta técnica se aplica iterativamente debido a que en cada iteración se pueden generar nuevos elementos inválidos.

2.5. Lineamiento

De manera similar, se desarrollará desde otra perspectiva la reparación de los elementos inválidos en las regiones de transición que pueden generarse al intentar lograr una representación correcta de los límites.

Para esto se establecerán medidas de calidad ya definidas en [Daines y Lobos, 2018] y se validará la propuesta realizando un estudio de la cantidad de Elementos inválidos generados en cada iteración.

2.6. Objetivos

2.6.1. Objetivo general

Ubicar y reparar elementos inválidos o de mala calidad en la superficie de una malla geométrica tridimensional de tipo Octree mixta.

2.6.2. Objetivos específicos

- Validar la malla resultante de la refinación mediante la técnica Octree.
- Medir la calidad de la malla generada y refinar los elementos que sean inválidos.
- Establecer una estructura de datos que permita volver a un estado anterior de la malla, así en caso de que existan elementos inválidos, permitir volver a un estado válido anterior.
- Realizar pruebas para posteriormente comparar tiempos de ejecución con respecto a la versión propuesta en [Daines y Lobos, 2018].

CAPÍTULO 3

Marco conceptual

Los algoritmos basados en la técnica Octree deben gestionar el problema mencionado en la sección anterior, la representación incorrecta de los límites del dominio, por sobre todo en dominios cóncavos.

Un octante límite es aquel que intersecta la frontera de Ω^1 , puede presentar nodos dentro y fuera del dominio Ω , los nodos que estén justo en la frontera son considerados nodos externos.

Cuando el octante límite contiene sólo un hexaedro, puede emplearse un patrón de superficie, definidos en [Lobos, 2013], dicho trabajo presenta un conjunto de patrones de elementos mixtos, que se emplean en la superficie del dominio objetivo y permiten conservar la consistencia de las mallas resultantes, estos patrones están pensados para combinarse con cualquier técnica de mallado que produzca una malla hexaédrica regular o irregular, las mallas Octree son un tipo de malla hexaédrica. Un patrón de superficie es un conjunto de diferentes tipos de elementos que sustituyen al hexaedro para reducir las posibilidades de producir elementos inválidos. Dependiendo de la configuración de nodos internos/externos del octante, se empleará un patrón distinto.

En [Lobos y González, 2015] se describe una técnica de mallado que incluye patrones de superficie y transición. Esta técnica permite definir diversos RL para Ω . Esta técnica es la base del trabajo desarrollado por Daines y Lobos en [Daines y Lobos, 2018] y del presente trabajo. En [Daines y Lobos, 2018] se detectó que este algoritmo puede producir algunos elementos no válidos cuando las transiciones entre regiones de diferente RL se presentan en el borde. Se trata de un octante que debe llenar ambas regiones, gestionar una transición y, al mismo tiempo lograr la representación del borde.

3.1. Medida de calidad para elementos en Mallas Mixtas

En general, la calidad de un elemento geométrico viene determinada por el nivel de deformación en comparación con la representación geométrica más regular o perfecta del elemento. La medición de la calidad tiene dos propósitos fundamentales:

- Comparar la malla, ya sea con otras mallas producidas por el mismo dominio o con la propia noción de malla “teóricamente perfecta”.
- Mejorarla, es decir, obtener una malla “mejor” comenzando por un estado de mala calidad mediante el uso de algoritmos de reparación.

Existen múltiples medidas de calidad de los elementos geométricos utilizados, cada una de las cuales mide distintos tipos de deformaciones. Una medida utilizada habitualmente es la relación de aspecto, AR^2 . En general, el AR de un elemento es la relación entre su borde

¹ Ω : dominio de entrada para el que se requiere generar una malla.

²AR : Aspect Relation.

más corto y su borde más largo. Otra variación significativa es el Relación de Aspecto Gamma, ARG³, el ARG penaliza más las deformaciones de un tetraedro en comparación con otras variaciones del AR. Para medir la distorsión de un nodo en contraste con su vecino podemos utilizar la Jacobiana. Siendo J^a el Jacobiano del nodo a y d_i el vector del nodo a al nodo i . El Jacobiano se puede calcular realizando el producto punto entre la distancia nodal del vector altura del tetraedro y el producto cruz de las distancias nodales de los vectores basales.

El problema del Jacobiano es sus dependencias a las distancias entre nodos y a la ortogonalidad entre sus aristas, con respecto a la primera, dos elementos con los mismos ángulos internos pero con tamaños distintos de aristas obtienen distintos valores para esta medida de distorsión y sobre la segunda, la medida de calidad no se puede extender a todos los tipos de elementos geométricos que se encontrarán en las mallas mixtas.

Por tanto, para suplir la dependencia a las distancias se utilizará el Jacobiano Escalado, J_S ⁴. Esta medida de calidad normaliza los vectores del tetraedro, así sólo dependerá de la ortogonalidad de sus aristas. El J_S de un hexaedro es el peor J_S de sus nodos.

En [Daines y Lobos, 2018] se define el rango de aceptación de la calidad de los elementos. Principalmente, $J_S < 0$ son Elementos invertidos, se define en [Shepherd y Johnson, 2008] un estándar utilizado en las métricas de calidad para Jacobiano Escalado importado por la librería Veredict de Cubit [SandiaLabs, 2007] que fija $J_S < 0$ como Elementos de inaceptables y $J_S \in [0, 0,2[$, correspondería a Elementos cuestionables, es decir, Elementos de calidad deficiente. Además, se destaca el estándar de ANSYS, una herramienta robusta a la hora de realizar simulaciones computarizadas, que considera $J_S \in [0, 0,03[$, como Elementos inválidos. Elementos con $J_S \geq 0,2$ como Elementos de buena calidad y Elementos con $J_S = 1$, como un hexaedro perfecto.

Por tanto, para este análisis se acotará definiendo $J_S \in]-1, 0]$ como Elementos inválidos, estos específicamente corresponden a Elementos negativos o invertidos, que conformarían mallas con topología inconsistente. Luego, se considerará, a diferencia de los trabajos anteriores, Elementos de calidad cuestionable los que presenten $J_S \in]0, 0,05]$, este intervalo se acotará y separará en dos intervalos, $J_S \in]0, 0,03] \wedge J_S \in]0,03, 0,05]$, esto con el fin de enfocar el estudio en Elementos que produzcan una malla inconsistente, reduciendo la cota superior de 0,2 a 0,05 y estudiar las consecuencias en la propagación del cambio de la calidad del refinamiento, dividiendo el intervalo en dos, considerando el estándar de ANSYS como referencia al dividir dicho intervalo.

Por otra parte, para suplir la deficiencia en su dependencia a la ortogonalidad de sus aristas, se usará un Jacobiano para cada uno de los tipos de elementos que encontraremos en la malla, esta medida de calidad es el Jacobiano Escalado Normalizado del Elemento⁵, más adelante denotado como J_{ENS} .

El índice J_{ENS} calcula el J_S del elemento perfecto K^e , donde e denota el elemento correspondiente. Los valores para los distintos elementos utilizados en este trabajo para las mallas Octree mixtas se encuentran en la siguiente Ecuación 1.

³ARG: Aspect Ratio Gamma.

⁴ J_S : Scaled Jacobian

⁵ J_{ENS} : Element Normalized Scaled Jacobian

$$\begin{aligned}
 K^T &= \frac{\sqrt{2}}{2} \longrightarrow \text{Tetraedro} \\
 K^P &= \frac{\sqrt{6}}{3} \longrightarrow \text{Pirámide} \\
 K^W &= \frac{\sqrt{3}}{2} \longrightarrow \text{Cuña}
 \end{aligned} \tag{1}$$

De esta forma, como se vio en [Daines y Lobos, 2018], con ayuda de estas constantes podemos calcular el J_{ENS} de cualquier nodo de un elemento sustituyendo el valor correspondiente en la Ecuación 2. De forma similar a J_S para realizar el cálculo de J_{ENS} utilizamos la Ecuación 3, donde i son los nodos del elemento e . y J_{ENS}^e es su J_{ENS} .

$$J_{ENS}^n = \begin{cases} (1 + K^e) - J_S, & J_S > k^e \\ J_S/K^e, & -k^e \leq J_S \leq K^e \\ -(1 + K^e) - J_S, & J_S < -k^e \end{cases} \tag{2}$$

$$J_{ENS}^e = \begin{cases} \min\{J_{ENS}^i\}, & \forall i, J_{ENS}^i > 0 \\ \max\{J_{ENS}^i\} : J_{ENS}^i < 0, & \exists i, J_{ENS}^i < 0 \end{cases} \tag{3}$$

Para J_{ENS} se definen los mismo intervalos que para J_S para identificar Elementos.

- $J_{ENS} \in]-1, 0]$: Elementos inválidos.
- $J_{ENS} \in]0, 0,05]$: Elementos cuestionables.
- $J_{ENS} \in]0,05, 1]$: Elementos de buena calidad.

En la etapa de validación, el estudio se concentrará en contabilizar la frecuencia de Elementos que presentan una calidad no deseable(inválidos y cuestionables), por ello se definirá la frecuencia de Elementos con $J_{ENS} \in]m, n]$, como E_m^n .

Teniendo, E_{-1}^0 , como la expresión que se referirá a la frecuencia de Elementos Inválidos, $E_0^{0,03}$, a los Elementos Cuestionables y $E_{0,03}^{0,05}$, a parte de los Elementos de Buena Calidad

3.2. Malla inicial

Para comprender de mejor forma el problema buscaremos un entorno donde existan elementos inválidos, para ello procederemos a estudiar la localidad de los elementos de mala calidad, entonces se generará una malla que llamaremos la *malla inicial*.

Ésta será la representación de una corteza cerebral a un nivel 5 de refinamiento en general y se le aplicará un refinamiento de nivel 7 a una superficie prismática regular inserta en la parte frontal del cerebro. Como se muestra en la representación de puntos de la malla del cerebro y representación alámbrica de la superficie prismática de la Figura 5.

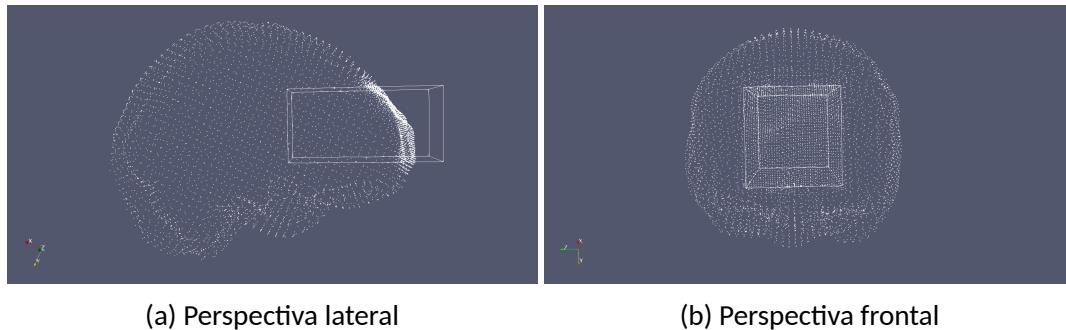


Figura 5: Muestra frontal y lateral de la malla inicial con refinamiento global de nivel 5 y refinamiento en superficie prismática regular de nivel 7.

Fuente: Elaboración propia.

3.3. Herramientas

En este trabajo se utilizarán algunas herramientas construidas por la comunidad como el generador de Mallas, el generador de estadísticas J_{ENS} , un software para visualización, scripts en Bash y Python para la ejecución de la solución y el análisis de los resultados, respectivamente.

3.3.1. Mixed-element mesh generator

El generador de mallas de elementos mixtos [Lobos y González, 2015] es una herramienta creada para facilitar la generación de mallas, a través de la línea de comandos se puede generar mallas con superficies de refinamiento en diferentes niveles, exportar en diversos formatos para visualización, iniciar con una malla refinada y refinar específicamente octantes entregados como input. Este software recibe modelos de datos de mallas en formato *mdl*.

```

1 input >      ./meshgen -h
2 output >     use: ./meshgen [-d] input.mdl [-o] input.off [-u] output
3                      [-c] volume_mesh.oct (octant mesh to start from)
4                      [-s] ref_level [-a] ref_level [-b] file.reg
5                      [-l] list_file.txt [-r] input_surface rl [-g] [-v]
6 where:
7           one of the parameters must be an input surface mesh in
8           mdl or off format. If output name is not provided it
9           will be saved in input_name.m3d. Options:
10          -s Refine octants intersecting the input surface.
11          Parameter ref_level is the refinement level
12          -a Refine all elements in the input domain.
13          Parameter ref_level is the refinement level
14          -b Refine block regions provided in file file.reg
15          -l Refine elements provided in the file by their index
16          -r Refine surface region. Will refine all the elements
17          in the provided input_surface at level rl
18          -g save output mesh in GetFem format (gmf)
19          -v save output mesh in VTK ASCII format (vtk)
20          -i save output mesh in MVM ASCII format (mvm)
```

```
21      -m save output mesh in M3D ASCII format (m3d)
```

Listing 1: Opciones de mesher generator.

Fuente: Elaboración propia.

Se denominará como *MESHER_GENERATOR* en el resto del presente trabajo.

De esta herramienta, se destacará los diferentes formatos de exportación:

- vtk: Formato utilizado para visualización en la plataforma Paraview.
- oct: Formato utilizado para persistir los datos utilizados en el algoritmo.

Adicionales a estos formatos, se agregarán algunos importantes adicionales para realizar las iteraciones y validaciones correspondientes explicados en la siguiente sección.

Otra funcionalidad a destacar es la opción de refinamiento dado un listado de Octantes y Elementos, que se explotará y mejorará para definir esta propuesta.

3.3.2. Mixed-element J_{ENS} statistics generator

El generador de estadísticas J_{ENS} para mallas Octree de elementos mixtos es una herramienta que facilita la validación de mallas, creada por la comunidad y liderado por el profesor Claudio Lobos. Puede generar histogramas con diferentes índices como Scaled Jacobian, Normalized Scaled Jacobian, Aspect Ratio, etc. En este trabajo lo utilizaremos principalmente para validar cada malla generada enfocándonos sólo en obtener la frecuencia del índice J_{ENS} . Esto es un histograma con la frecuencia de todos los elementos de la malla con $J_{ENS} \in \{-1, 1\}$, como se muestra en la figura Listing 3.

```
1 input    >   ./jens -h
2 output   >   use: ./jacobian -option input.m3d
3           options:
4           -s : Scaled Jacobian statistics
5           -e : Element Normalized Scaled Jacobian statistics
6           -j : List Jens of each Element Jens [Js]
7           -a : List Aspect Ratio for each element
8           -l : List Jens of each node for each element
```

Listing 2: Opciones de jens calculator.

Fuente: Elaboración propia.

```
1 input    >   ./jens -e ./src/c_5r7.mdl
2 output   >   negative: 27
3           <0.0333 : 4
4           <0.05   :5
5           <0.1    :13
6           <0.15   :54
7           <0.2    :219
8           <0.25   :356
9           <0.3    :9918
10          <0.35   :8431
11          <0.4    :3386
12          <0.45   :6641
```

```
13      <0.5      :2341
14      <0.55     :16891
15      <0.6      :2659
16      <0.65     :2689
17      <0.7      :2038
18      <0.75     :4872
19      <0.8      :1108
20      <0.85     :2477
21      <0.9      :457
22      <0.95     :408
23      <1        :108662
24      total: 173656
25      worst quality -0.792707
26      average quality 0.806127
```

Listing 3: Estadísticas Jens para malla inicial, muestra una lista de frecuencias para diferentes cotas superiores para la calidad Jens encontrada en la malla.

Fuente: Elaboración propia.

Se denominará como **JENS_CALCULATOR** el resto del presente trabajo.
Convenientemente, los primeros tres intervalos del histograma pertenecen a los intervalos de interés para este estudio.

3.3.3. ParaView

Paraview es un software open-source multiplataforma que facilita la visualización de representaciones en 3D. Presenta una interfaz interactiva, nos ayudará analizar visualmente el estado de la malla en todo momento. Se utilizará el formato con extensión *vtk* para visualizar mallas en esta plataforma.

Este software presenta diferentes secciones que nos ayudarán a analizar cada malla generada, la sección marcada con el número 1 en Figura 6, se presentan todas las mallas que se representarán gráficamente en la sección 2. Existe un panel de herramientas en la sección 3 que nos ayudará a marcar/desmarcar elementos o puntos en el panel 2 que se verán reflejados e identificados en el recuadro 5. En la sección 4 podremos cambiar la forma de visualizar cada representación, esto nos ayudará a identificar elementos, identificar gráficamente si una malla es válida en su construcción, etc.

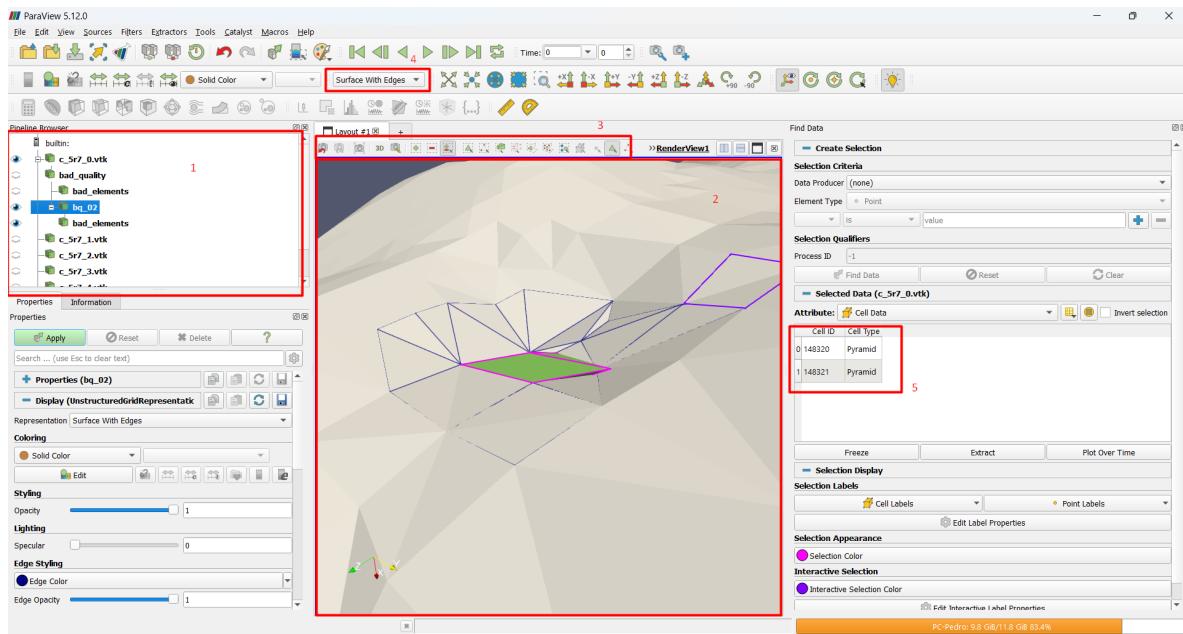


Figura 6: Interfaz de software Paraview, se muestra una sección de la malla inicial con dos elementos seleccionados.

Fuente: Elaboración propia.

3.4. Estructura de datos

En la construcción de la solución existen diversas estructuras de datos que se van complejizando a medida que se avanza en el algoritmo.

En esta sección sólo se explicará la estructura de datos elemental y de forma general.

Como se definió anteriormente, una malla Octree se divide en octantes, cada uno de estos octantes contienen Elementos mostrados en la Figura 1. Luego, cada uno de estos Elementos está constituido por Puntos, que es la unidad más pequeña.

Para exemplificar en Figura 7, se presenta un pequeño diagrama entidad relación que muestra la estructura de los objetos relacionados en el algoritmo de la generación de mallas.

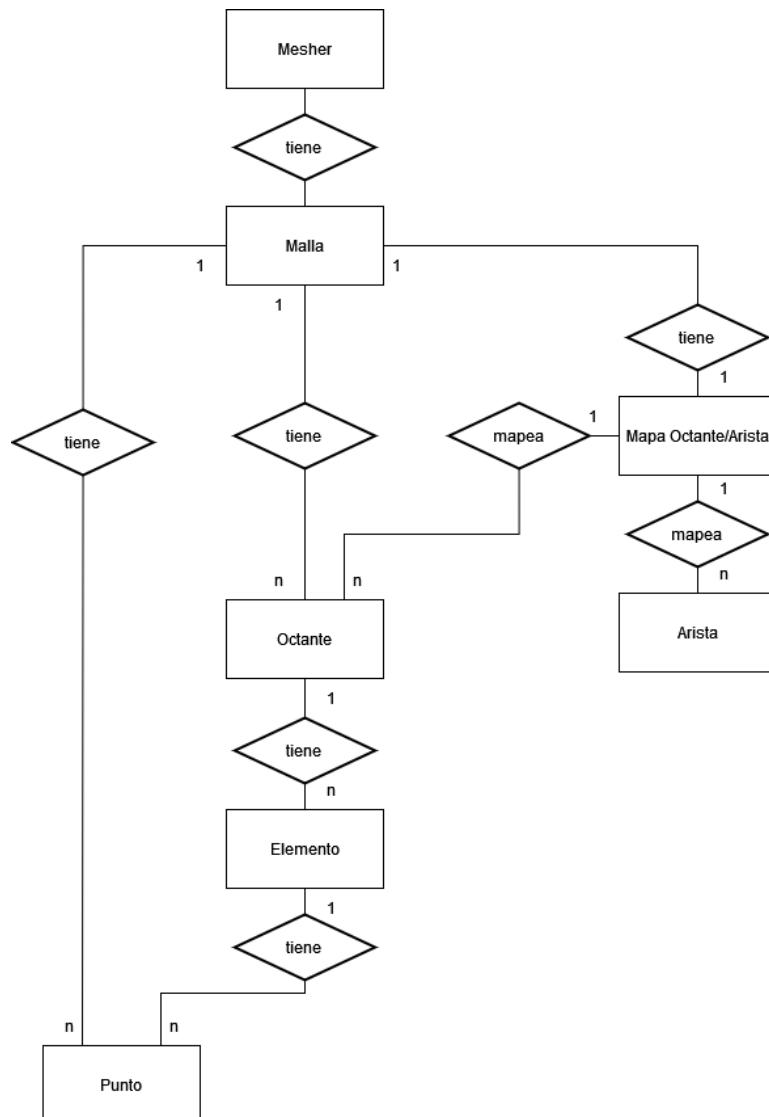


Figura 7: Diagrama entidad-relación de generador de mallas.

Fuente: Elaboración propia.

3.4.1. Acercamiento al algoritmo de Mixed-element mesh generator

Parte importante del software mencionado en la Subsubsección 3.3.1 es la generación de la malla que se define y explica brevemente en [Daines y Lobos, 2018].

Algorithm 1: Algoritmo de generación de mallas Octree con elementos mixtos y varios niveles de refinamiento.

Fuente: [Daines y Lobos, 2018]

Input : Refinement level constraints RLC , triangular surface mesh Ω .

Output: Volumetric mesh of Ω that meets RLC .

```
1 Procedure GENERATE_MESH( $RLC, \Omega$ ):
2    $mesh \leftarrow \text{GENERATE_BALANCED_OCTREE}(RLC, \Omega);$ 
3    $mesh \leftarrow \text{APPLY_TRANSITION_PATTERNS}(mesh);$ 
4    $bnodes \leftarrow \text{GET_BOUNDARY_NODES}(mesh, \Omega);$ 
5   for each node in  $bnodes$  do
6     if node  $\in l$  or (node is inside of  $\Omega$  & node is close to  $\Omega$ ) then
7       | node.PROJECT_ONTO( $\Omega$ );
8     end if
9   end for
10   $mesh \leftarrow \text{APPLY_SURFACE_PATTERNS}(mesh);$ 
11   $bnodes \leftarrow \text{GET_BOUNDARY_NODES}(mesh, \Omega);$ 
12  for each node in  $bnodes$  do
13    if node is outside of  $\Omega$  then
14      | node.PROJECT_ONTO( $\Omega$ );
15    end if
16  end for
17  return mesh;
```

En términos simples, el algoritmo consiste en generar una malla balanceada Octree, aplicar patrones de transición entre Octantes con diferencias de RL mayor a 1, luego se realizan proyecciones en los nodos que se encuentran cerca de la frontera de Ω para generar una mejor representación del borde.

Para este estudio, no es necesario profundizar en este algoritmo, pero es importante mencionarlo y entender su estructura de datos para explicar brevemente los diferentes formatos de exportación del algoritmo que utilizaremos para aplicar una versión iterativa del mismo enfocándonos específicamente en los Elementos de mala calidad.

CAPÍTULO 4

Propuesta de solución

4.1. Comprensión del problema

Inicialmente sabemos que los elementos inválidos podrían encontrarse luego de refinar una malla Octree con diferentes niveles de refinamiento, específicamente en el límite donde se producen las transiciones entre regiones de diferente nivel de refinamiento.

Entonces, se escogerá una malla frecuentemente utilizada en la comunidad, la malla de corteza cerebral, una malla de geometría suficientemente compleja por presentar múltiples concavidades y de calidad de representación suficiente por su amplia densidad de puntos que la conforman, en total 12,986 puntos, con esta malla se busca realizar un estudio gráfico preliminar.

4.1.1. Estudio preliminar

Para comenzar, se debe comprender el alcance del problema, entonces se ejecutó **JENS_CALCULATOR** a la “malla inicial” y se obtuvo el histograma con las estadísticas de calidad de los elementos de la malla inicial, Listing 3.

Nos enfocaremos sólo en elementos inválidos, por lo tanto, el threshold a estudiar será cero en esta etapa previa.

Actualmente, se identificaron 27 elementos invertidos, pero aún no se ha determinado cuáles son específicamente esos elementos. Para resolver este problema, es necesario que el algoritmo **MESHER_GENERATOR** incluya una funcionalidad para etiquetar los elementos inválidos.

Para ello se realizó una pequeña modificación a **MESHER_GENERATOR**, se le añadieron ciertas funcionalidades de **JENS_CALCULATOR**, para que al finalizar el proceso de proyección de nodos, calcule el J_{ENS} de cada Elemento presente en cada Octante e identifique cuáles son peores que el threshold definido para esta etapa.

Se propone, de manera similar a [Daines y Lobos, 2018], implementar un marcador para los octantes que contengan elementos de mala calidad, basado en una cota superior predefinida, como se muestra en algorithm 2.

Esta técnica permitirá identificar los octantes que necesitan ser refinados en etapas posteriores. En esta fase, nos enfocamos en identificar los elementos contenidos dentro de estos octantes.

Así, al obtener la lista de octantes involucrados, ahora es posible obtener los índices de los elementos que los componen, permitiendo su posterior análisis.

Utilizando la lista de elementos involucrados, podemos visualizar los resultados en ParaView. Esto nos permite generar la Figura 8, donde los elementos pertenecientes a los octantes afectados se destacan en color verde.

Con esta visualización, podemos confirmar que estos elementos se encuentran en la vecindad de las transiciones entre regiones con distintos valores de RL . Esta representación gráfica facilita la identificación de las áreas problemáticas dentro de la malla y demuestra cómo

Algorithm 2: Algoritmo para etiquetar octantes que contienen elementos con peor calidad que la cota entregada.

Input : Refinement level constraints RLC , triangular surface mesh Ω , quality threshold T , maximum number of iterations I .

Output: if successful, volumetric mesh of Ω that meet the RLC with quality greater than T .

```

1 Procedure GET_LABELED_OCTANTS( $M, T$ ):
2    $lo \leftarrow emptyset;$ 
3   for each element in  $M$  do
4      $q \leftarrow CALCULATE\_JENS(element);$ 
5     if  $q < T$  then
6        $octant \leftarrow element.GET\_OCTANT();$ 
7        $octant\_index \leftarrow octant.GET\_INDEX();$ 
8        $lo.ADD(octant\_index);$ 
9     end if
10   end for
11   return  $lo;$ 
```

las transiciones entre regiones con diferentes características influyen en la calidad de los elementos.

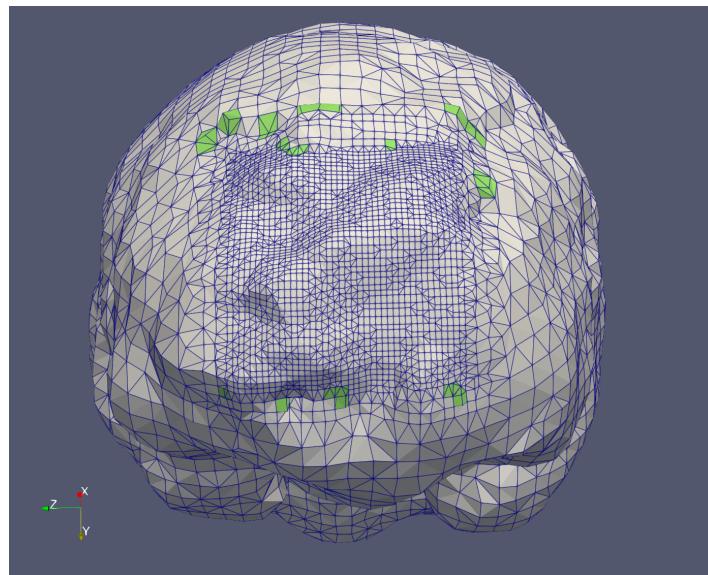
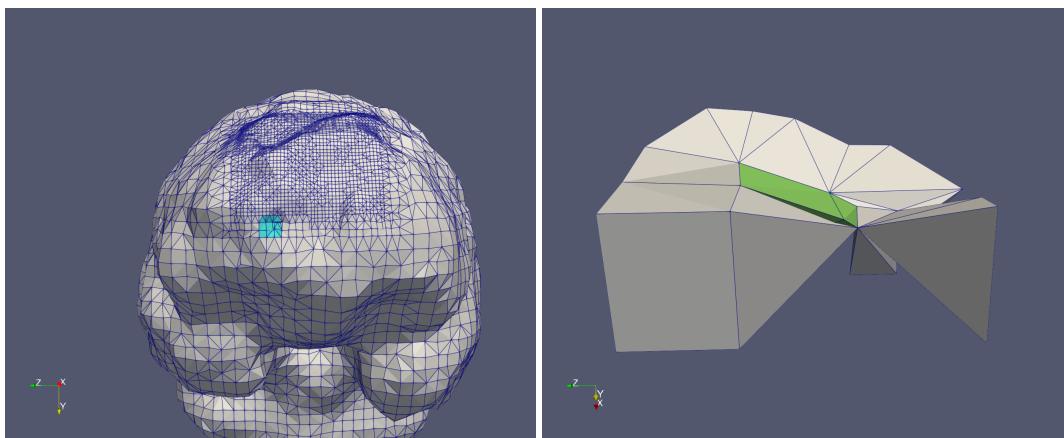


Figura 8: Zona frontal de la malla donde se presentan los elementos inválidos (verde).
Fuente: Elaboración propia.

4.1.2. Comprensión de elementos inválidos

Para comprender la naturaleza de los elementos invertidos, se realizará un análisis gráfico localizado, centrándose en un octante específico situado en la parte inferior izquierda del frontis de la malla. Este octante ha sido seleccionado debido a su característica cóncava en la superficie y porque contiene varios elementos inválidos. Además, se encuentra en la intersección de dos regiones que requieren refinamiento, cada una con diferentes valores de RL .

En la Figura 9, se presenta la región de interés en el contexto de la malla inicial. La imagen a la izquierda muestra el lóbulo frontal con la zona a estudiar en color celeste, incluye todos los Elementos del Octante, mientras que la imagen a la derecha proporciona una vista ampliada de la zona aislada del Octante afectado, la zona en celeste. En esta vista detallada, se pueden observar solo los Elementos contenidos en dicho Octante, con los Elementos inválidos resaltados en verde.



(a) Zoom out de sección extraída.
(b) Zona extraída con elementos de mala calidad resaltados.

Figura 9: Zona a analizar para evidenciar elementos de mala calidad, esta zona se extrajo de la representación de la malla con un nivel 5 de refinamiento general y nivel 7 de refinamiento en la superficie entregada. En la imagen de la izquierda se muestra una visión general de la superficie de la malla demarcada por elementos con vértices azules y la zona a estudiar de color celeste. En la imagen de la derecha, la zona a estudiar y un par de elementos de mala calidad destacados con color verde.

Fuente: Elaboración propia.

4.1.3. Análisis de la vecindad

En esta sección de la malla, se puede observar a simple vista una superficie de naturaleza cóncava. Este tipo de geometría tiende a facilitar la aparición de elementos inválidos durante el proceso de generación de la malla.

Como se muestra en la Figura 10, los elementos inválidos son fácilmente identificables. Estos elementos, destacados en verde, crean áreas en la malla que no representan fielmente

la geometría original y muestran incongruencias respecto a los elementos vecinos. En este contexto, los elementos inválidos generan una anomalía observable en relación a su entorno inmediato.

En particular, el par de elementos en verde está separado de los elementos adyacentes en su base. Esta separación indica una ruptura en la continuidad de la malla que no sigue la forma esperada de la superficie original. Este análisis resalta la importancia de prestar atención a la forma y las transiciones entre regiones de la malla para detectar y corregir estos problemas que pueden comprometer la calidad y precisión de la representación del modelo.

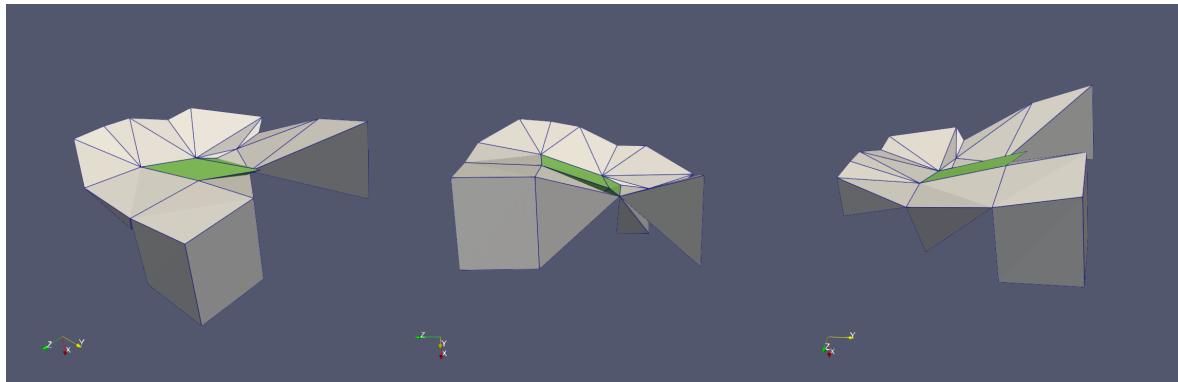


Figura 10: Perspectivas de vecindario (gris) donde se encuentra un par de elementos inválidos (verde).

Fuente: Elaboración propia.

4.2. Propuesta

Para abordar y mejorar la calidad de los Elementos inválidos en la malla, se propone refinar las localidades donde estos se encuentran. En términos específicos, esto implica realizar un refinamiento en los Octantes que están asociados con los Elementos inválidos.

Se definirá el algoritmo de esta propuesta como el siguiente:

1. Generación de malla inicial.

La generación de la malla inicial, implica utilizar la herramienta **MESHER_GENERATOR** para construir la malla inicial con las condiciones establecidas.

2. Identificación y Etiquetado de Octantes.

El etiquetador de Octantes, una modificación del algoritmo presentado en algorithm 2 que permitirá persistir los Octantes etiquetados.

3. Refinamiento de Octantes Identificados.

El refinamiento de una lista de Octantes utilizando el parámetro ‘-l’ del **MESHER_GENERATOR**.

4. Generación de estadísticas J_{ENS} .

Al finalizar la generación de la malla o refinamiento, se obtendrá y guardará de manera persistente su estadística para validar las condiciones de detención del algoritmo y generar un directorio historial con las mallas y sus correspondientes estadísticas para posterior análisis.

Es importante, en este punto, mencionar que se escogió seleccionar la última malla generada como la “malla solución” porque el algoritmo se comporta de manera exponencial en la mayoría de los casos, comportamiento analizado en la Sección 28.

4.2.1. Generación de malla inicial.

```
1 input > ./meshing_roi -d ./data/cortex.mdl -s 5 -r ./data/
    cortex_surf_roi.mdl 7 -u c_5r7_0 -m -v
2 output > All done in xxxx ms
```

Listing 4: Generador de malla, crea una malla refinada a nivel 5 y específicamente refina la superficie entregada a nivel 7, se exportará la malla con el nombre *c_5r7_0*, en formato vtk y m3d.

Fuente: Elaboración propia.

Para mejorar la calidad de la malla inicial, se implementaron modificaciones específicas en el algoritmo **MESHER_GENERATOR**. Estas modificaciones están diseñadas para evaluar y gestionar la calidad de los elementos generados, proporcionando estadísticas detalladas y facilitando la integración con otros sistemas de información. Se ha actualizado el algoritmo **MESHER_GENERATOR** para que, al finalizar la generación de la malla, se obtengan estadísticas de calidad J_{ENS} . Estas estadísticas se presentan en un formato que permite evaluar la calidad de los elementos en relación con ciertos umbrales predeterminados, además se modificó la información sobre los octantes que se guardan de manera persistente en el disco. Puntos a destacar de esta implementación:

- Generación de Estadísticas J_{ENS} :

Al finalizar la generación de la malla, el algoritmo calcula las estadísticas J_{ENS} , las cuales proporcionan una medida de la calidad de los Elementos. Estas estadísticas se registran en cada iteración, creando un historial detallado que permite monitorear la calidad de la malla de manera continua.

- Exportación de datos en formato estándar:

Para facilitar el análisis de estas estadísticas, los datos se almacenarán de forma persistente en un archivo con extensión *.histo*. Este archivo contiene dos columnas clave: la primera registra los umbrales de calidad, mientras que la segunda muestra la frecuencia de elementos cuya calidad se encuentra por debajo de esos umbrales. Un ejemplo del contenido de este archivo se ilustra en la Listing 5 , donde los intervalos de calidad están etiquetados con “<intervalo_fq_bqe>” para cada intervalo de calidad, y el nombre del archivo incluye un identificador de la iteración correspondiente.

El archivo comienza con una cota que incluye los elementos con J_{ENS} negativo, seguida de cotas progresivas, con excepciones en los primeros tres valores ($neg, 0,03, 0,05$), y luego con incrementos regulares de 0,05. Esta disposición permite una evaluación sistemática de la calidad de los elementos a lo largo de un rango definido de valores J_{ENS} . Los intervalos y cotas superiores se mantuvieron consistentes con el formato de **JENS_CALCULATOR**, facilitando su integración con otros sistemas.

El histograma generado por el módulo **JENS_CALCULATOR** es crucial, ya que proporciona una visualización clara y eficiente de la distribución de la calidad de los elementos dentro de la malla. Esto permite identificar rápidamente cómo se ven afectados los elementos en cada iteración, facilitando la toma de decisiones y el ajuste del refinamiento de la malla.

```
1 -----  
2 File: c_5r7_x.histo  
3 -----  
4 negative <neg_fq_bqe>  
5 0.030000 <0.03_fq_bqe>  
6 0.050000 <0.05_fq_bqe>  
7 0.100000 <0.10_fq_bqe>  
8 0.150000 <0.15_fq_bqe>  
9 0.200000 <0.20_fq_bqe>  
10 0.250000 <0.25_fq_bqe>  
11 0.300000 <0.30_fq_bqe>  
12 0.350000 <0.35_fq_bqe>  
13 0.400000 <0.40_fq_bqe>  
14 0.450000 <0.45_fq_bqe>  
15 0.500000 <0.50_fq_bqe>  
16 0.550000 <0.55_fq_bqe>  
17 0.600000 <0.60_fq_bqe>  
18 0.650000 <0.65_fq_bqe>  
19 0.700000 <0.70_fq_bqe>  
20 0.750000 <0.75_fq_bqe>  
21 0.800000 <0.80_fq_bqe>  
22 0.850000 <0.85_fq_bqe>  
23 0.900000 <0.90_fq_bqe>  
24 0.950000 <0.95_fq_bqe>  
25 1.000000 <1.00_fq_bqe>  
26
```

Listing 5: Estructura de archivo con extensión *.histo* que contiene un histograma con estadísticas de elementos J_{ENS} de la malla de la iteración actual.

Fuente: Elaboración propia.

- Gestión de información entre sistemas:

La exportación en formato *.histo* también permite una integración óptima de la información entre diferentes sistemas de procesamiento y análisis. Esta gestión eficiente asegura que los datos de calidad de la malla sean accesibles y utilizables en otras aplicaciones o modificaciones al algoritmo en trabajos futuros, facilitando así el flujo de

trabajo en entornos complejos de simulación o modelado.

- Respaldo de la información de la malla:

Además de las estadísticas de calidad, se necesitará disponer de un estado de la malla, por defecto, **MESHER_GENERATOR** crea un archivo de respaldo de extensión .oct, que contiene una representación detallada de la malla, incluyendo la posición de los puntos y la definición de los octantes en cada iteración. Este archivo es esencial para conservar un registro completo del estado de la malla en diferentes etapas del proceso de generación de la malla y su refinamiento.

Este archivo de respaldo, será modificado para incluir información adicional sobre los octantes, para posteriormente localizar correctamente octantes en base a sus identificadores únicos.

Este archivo .oct permite una fácil recuperación y revisión de la malla generada, facilitando la identificación de problemas y la evaluación de la calidad en etapas posteriores. También sirve como un recurso valioso para la documentación y el seguimiento de los cambios realizados durante el proceso de refinamiento de la malla.

4.2.2. Identificación y etiquetado de Octantes.

Como se explicó en la sección de *Análisis de la vecindad*, identificar y etiquetar los octantes se realizará por medio del algoritmo 2, adicional a esto, se creará un archivo con la extensión .ref para persistir la información. Este archivo es crucial porque almacenará una lista detallada de los índices de los octantes que necesitan ser refinados. Es importante que esta lista sea persistente para, además de tener un historial de los octantes refinados en cada iteración, lograr convalidarlo con el trabajo existente en **MESHER_GENERATOR**.

La persistencia de esta información permite rastrear los cambios realizados en cada iteración del refinamiento, asegurando una documentación completa y accesible del proceso. Facilita la revisión y análisis posterior pudiendo incluso, utilizar otros lenguajes de programación en el proceso.

```
1 -----
2 File: c_5r7_x.ref
3 -----
4 <octid_0>
5 <octid_1>
6 ...
7 <octid_NOL-1>
8 <octid_NOL>
```

Listing 6: Lista de octantes a refinar en malla inicial.

Fuente: Elaboración propia.

En Listing 6, se dispone un ejemplo del archivo .ref, considerando NOL la cantidad de octantes a refinar y $octid_k$ los identificadores únicos de los octantes a refinar, siendo $k \in \{0, \dots, NOL\}$.

4.2.3. Refinamiento de Octantes Identificados.

Teniendo en consideración la lista de octantes a refinar exportado en el archivo .ref, queda entonces entregar esta lista a **MESHER_GENERATOR**, junto con el listado debe entregarse también la malla de la iteración anterior.

De esta manera, en una posible iteración 3, por ejemplo, se utilizará el archivo .oct y .ref de la iteración 2, esto es comenzar con la malla generada en la iteración anterior como en Listing 7, donde se definen los siguientes elementos:

- **-d ./data/cortex.mdl**: Corteza del cerebro a analizar.
- **-c c_5r7_2.oct**: Información sobre los octantes generados en la iteración anterior.
- **-l ./c_5r7_2.ref**: Lista de Octantes a refinar.
- **-t 0.0**: Treshold definido en cero.
- **-u c_5r7_3**: Nombre base de los archivos a exportar en la iteración actual.
- **-m**: Exportación en formato m3d.
- **-v**: Exportación en formato vtk.

Lo anterior, refinará los Octantes de mala calidad de la malla en la iteración 2 según la cota definida en dicha iteración y analizará los nuevos elementos para definir una nueva lista de Octantes a refinar. Esto se exportará con el nombre de la iteración actual, exportando los siguientes archivos.

- **c_5r7_3.oct**: Información sobre los puntos y octantes de la malla generada,
- **c_5r7_3.ref**: Lista de Octantes a refinar en iteración actual.
- **c_5r7_3.m3d**: Información sobre puntos, elementos y octantes de la malla generada para visualización en Geomview.
- **c_5r7_3.vtk**: Información sobre puntos, elementos y octantes de la malla generada para visualización en Paraview.

```
1 input >      ./meshers_roi -d ./data/cortex.mdl -c ./c_5r7_2.oct -l ./  
    c_5r7_2.ref -t 0.0 -u c_5r7_3 -m -v  
2 output >     All done in xxxx ms
```

Listing 7: Generador de malla, crea una nueva malla, tomando como input la malla inicial *c_5r7_0.oct*, la lista de octantes a refinar *c_5r7_0.ref*, la cota de calidad J_{ENS} definido en cero, se exportará la malla con el nombre *c_5r7_1*, en formato vtk y m3d.

Fuente: Elaboración propia.

4.2.4. Ciclo Iterativo de Evaluación y Refinamiento.

Teniendo todo para generar un proceso iterativo, se propone entonces un algoritmo, algorithm 3, que ejecutará **MESHER_GENERATOR** las veces necesarias hasta que ya no existan elementos de mala calidad.

Como se explicó en la Sección 3.3.1, **MESHER_GENERATOR** tiene varias funcionalidades, para esta propuesta utilizaremos su funcionalidad de refinamiento general, que generará los refinamientos necesarios para las zonas definidas y definirá los elementos requeridos para consultar la calidad de la malla.

Esta primera iteración será una ejecución de **MESHER_GENERATOR**, donde se le entregará como input la malla principal a refinar que corresponde a los puntos que componen la corteza cerebral, también se le entregará la malla del área a refinar en la zona frontal de la corteza, su *threshold* para la identificación de Octantes que contengan elementos de mala calidad y el nombre base para la exportación de los archivos.

El algoritmo además, propone una ejecución de **MESHER_GENERATOR** que se ejecutará en todas las iteraciones restantes, tomará la información de la malla generada en las iteraciones previas para refinar los Octantes identificados en las iteraciones previas. Se le entregará como input la malla principal, los octantes y puntos definidos en la iteración anterior, la lista de Octantes a refinar en la iteración actual y que fueron identificados de la iteración previa, su *threshold* y el nombre base para la exportación de los archivos.

Por último, es necesario una condición de término del algoritmo, para ello se propone un evaluador de calidad denominado **IMPROVER_EVALUATOR**, este se ejecutará al final de cada iteración y consultará la existencia del archivo que contiene la lista de Octantes a refinar. Esto se realiza de esta manera, ya que, cuando no existe Octantes por refinar, es decir, no existen elementos de mala calidad, no se crea el archivo con extensión *.ref*.

Algorithm 3: Algoritmo iterativo propuesto

Input : Malla corteza $main_surf$.
Superficie a refinar ref_surf .
Nombre base para archivos de exportación $base_name$.
Cota de calidad T .
Número máximo de iteraciones I .

Output: if successful, volumetric mesh of Ω that meet the RLC with quality greater than T .

1 **Procedure** GENERATE_MESH($main_surf, ref_surf, base_name, T, I$):
2 **foreach** $i \in \{0, \dots, I\}$ **do**
3 $out_name \leftarrow base_name + _ + i$;
4 **if** $i == 0$ **then**
5 | mesher_generator ($main_surf, ref_surf, threshold, out_name$) ;
6 **end if**
7 **else**
8 | $prev_iteration \leftarrow i - 1$;
9 | $base_name_prev \leftarrow base_name + _ + prev_iteration$;
10 | $ref_name_prev \leftarrow base_name_prev + .ref$;
11 | $octs_name_prev \leftarrow base_name_prev + .oct$;
12 | mesher_generator ($main_surf, octs_name_prev, ref_name_prev, threshold, out_name$) ;
13 **end if**
14 | IMPROVER_EVALUATOR (ref_name_prev, i) ;
15 **end foreach**
16 | $lo \leftarrow GET_FILE_LINES (ref_name_prev)$;
17 | PRINT "Finished, number of iterations exceeded." ;
18 | PRINT "Octants to improve: lo " ;
19 | **return** mesh;
20 **Procedure** IMPROVER_EVALUATOR(ref_name, i):
21 **if** FILE_EXIST(ref_name) **then**
22 | $lo \leftarrow GET_FILE_LINES (ref_name)$;
23 | PRINT "Octants to improve in iteration i : lo " ;
24 **end if**
25 **else**
26 | PRINT "No more Octants to improve in iteration i ." ;
27 | BREAK;
28 **end if**

CAPÍTULO 5

Validación de la solución

5.1. Stack de tecnologías

En primera instancia existe la tentación por implementar una modificación completa de la herramienta **MESHER_GENERATOR**, esto es viable pero se descarta cuando se prioriza la escalabilidad y mantenibilidad de código. Sobre la eficiencia y rapidez de la propuesta, que son características importantes que evidencian calidad, se evaluarán en secciones posteriores. En **MESHER_GENERATOR** sólo se realizarán las modificaciones descritas en las secciones anteriores que incluyen funcionalidades para identificar y exportar los Octantes a refinar, para generar estadísticas al finalizar el refinamiento en las mallas y las pequeñas modificaciones a las funcionalidades de exportación de la información de los Octantes para luego identificarlos globalmente y así refinar los Octantes correctos.

Entonces, como debemos ejecutar **MESHER_GENERATOR**, existen varias formas de realizar esto, pero se escogerá la que utilice menos recursos para ejecutar llamadas a sistema como lo hace *bash* en linux. Ya que se requiere de funcionalidades básicas como iteraciones y manipulación de archivos, esta es una buena herramienta a utilizar.

5.2. Proceso de análisis

Para validar la propuesta, se utilizará una serie de mallas con diferentes complejidades.

- Malla de corteza cerebral con área de refinamiento prismática en la zona frontal.
Propuesta inicial, de por sí la malla de corteza cerebral posee una complejidad media, contiene algunas zonas cóncavas que podrían enriquecer la validación del algoritmo. El prisma se posicionará en la zona del lóbulo frontal.
- Malla de corteza cerebral con área de refinamiento prismática en la zona trasera.
En la zona trasera, lóbulo occipital de la corteza cerebral, de naturaleza cóncava e irregular se posicionará el prisma para refinar localmente, esto con el fin de complejizar el trabajo del algoritmo.
- Malla de Moai con área de refinamiento prismática en la zona superior.
Para este análisis se escogió una representación de un Moai, que naturalmente no posee múltiples zonas cóncavas, entonces se aplicará el área de refinamiento en la zona superior donde se presenta la mayor cantidad de zonas cóncavas.

- Malla de Moai con área de refinamiento prismática en la zona inferior.

Se utilizará también la zona inferior, que no posee mayor nivel de complejidad respecto a su concavidad, con motivo de aumentar el universo de soluciones.

- Malla de paladar con área de refinamiento prismática en la zona superior.

Esta representación posee áreas muy finas con respecto al resto de la malla, entonces se aplicó un área de refinamiento adicional en la zona superior por su complejidad en función de concavidad.

- Malla de coxis.

En general esta representación es muy compleja, presenta zonas muy pequeñas y con múltiples concavidades.

Luego de realizar algunas iteraciones básicas para tener una idea del comportamiento del algoritmo con las mallas, realizando iteraciones donde se añadieron áreas prismáticas en diferentes zonas y otras iteraciones donde se modificaron los niveles de refinamiento tanto general como del área prismática

Se identificaron dos factores importantes para tratar cada malla, y que si bien, parecen tener una correlación, este trabajo no busca encontrar dicha relación, por tanto, se supondrán como factores sin correlación.

- Complejidad geométrica Algunas mallas se notaron con alta densidad de regiones que forman depresiones o concavidades. Incluso algunas de ellas, presentaban alta densidad de estas zonas sólo en cierta parte de la representación. Entonces para aquellas que presentaban su alta densidad de concavidades en una área determinada, se le aplicó el área prismática tanto en su zona de alta densidad como también fuera de ella, como sucede en el caso de la representación del Moai y Corteza Cerebral. Para otras mallas, simplemente se insertó un área prismática con el fin de generar Elementos inválidos, como sucede en el Paladar, que si bien, presenta variedad de zonas cóncavas, no genera Elementos inválidos con sólo refinar globalmente, entonces para este caso, se realizaron variaciones en su refinamiento global presentando sólo una opción de posición en su zona prismática de refinamiento local.
- Capacidad Computacional Esto hace alusión principalmente al tiempo que demora el algoritmo en realizar refinamiento global o incluso mallas que generan detenimientos forzados del proceso donde se ejecuta el algoritmo por, quizás, presentar un alto número de vértices y aristas, tan alta que el sistema operativo probablemente queda sin memoria. Como por ejemplo, cuando se realizó un refinamiento global de nivel nueve en la malla del Coxis.

Luego de iterar las mallas, se analizará sus estadísticas sobre Elementos de mala calidad.

5.3. Ejecución

En cada malla dependiendo de su complejidad se aplicará o no, un área prismática para refinar localmente, de esta manera añadir consistencia a los resultados.

La única malla a la que no se le aplicará el área prismática es la malla de coxis debido a su compleja naturaleza explicada en el itemizado anterior.

A cada caso, se ejecutará el algoritmo con threshold $t \in \{0, 0,03, 0,05\}$ y debido a que E en la mayoría de los casos converge evidentemente en las primeras diez iteraciones, se escoge esta como cota superior.

Es necesario, antes de comenzar, definir lo siguiente. Se referirá a una malla con refinamiento global de nivel X y refinamiento local de nivel Y , como la malla XrY .

Entonces una ejecución del algoritmo para la corteza cerebral con $RL_{global} = 5$, $RL_{local} = 7$, región prismática con identificador 2, número de iteración 5 y threshold 0,03, se expondrá como “cortex_3_5r7_1_n”, teniendo la siguiente estructura “`<malla_threshold_RLglobalrRLlocal_ROIid_iteración>`”.

5.3.1. Malla de corteza cerebral

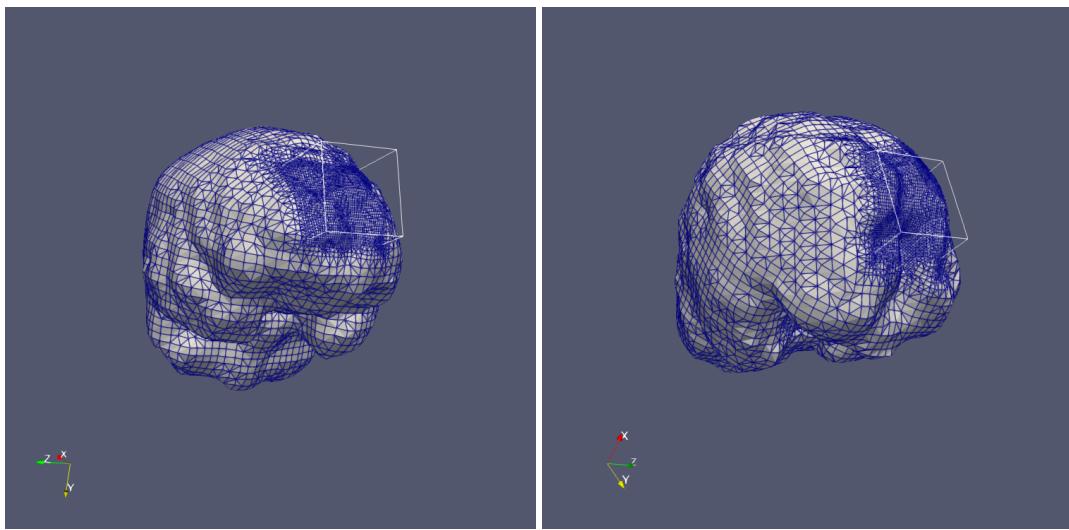
La corteza cerebral, si bien, presenta amplias zonas irregulares, el algoritmo no presenta problemas en su ejecución. Entonces se tratará como un caso para analizar complejidad geométrica.

Al implementar el algoritmo propuesto en la malla de corteza cerebral, entregándole como input las cotas definidas recientemente, el modelo de la corteza, los modelos de la superficie prismática a refinar, la base de nombre de los archivos exportados y una cota de diez iteraciones. Obtenemos los siguientes resultados Figura 18, que nos muestra un histograma agrupado por los primeros cuatro intervalos de calidad.

En la parte superior del recuadro, se muestran los histogramas para la primera superficie y en la parte inferior para la segunda superficie, frontal y occipital respectivamente, en la Página 29 se evidencia la disposición de las superficies en la corteza.

5.3.2. Malla de Moai

La ejecución del algoritmo en la malla de Moai para los casos con refinamiento inferior y superior, como podemos ver dispuestas en Figura 12, la superficie de la zona inferior se identificará como la superficie 1 y la superior se identificará como la malla 2. Se puede ver claramente la diferencia en la complejidad geométrica entre ambas superficies, teniendo en la



(a) Representación corteza cerebral con refinamiento en lóbulo frontal. (b) Representación corteza cerebral con refinamiento en lóbulo occipital.

Figura 11: Diferentes localidades de refinamiento en corteza cerebral.
Fuente: Elaboración propia.

parte superior más concavidades relacionadas a las facciones del moai. Luego, en Figura 17, se puede identificar claramente, cómo la superficie 1 se muestra más simple, reduciendo considerablemente más rápido la cantidad de Elementos de mala calidad en pocas iteraciones. No así en la superficies superior, dónde se puede notar como en las primeras iteraciones, para todos los thresholds aumenta en su segunda iteración la cantidad de Elementos negativos.

en Figura 17.

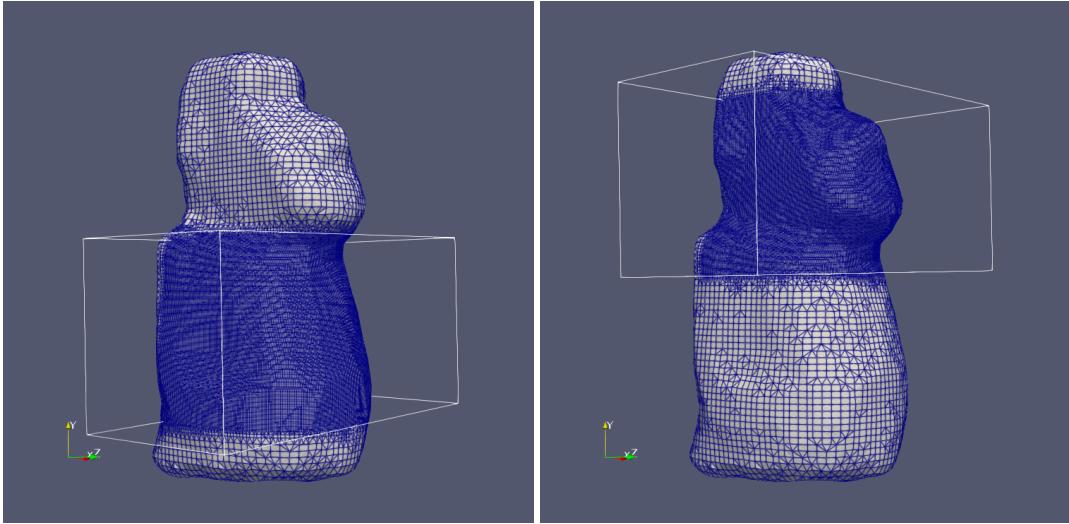
5.3.3. Malla de paladar

En el caso de la malla de paladar, se realizó un refinamiento general de nivel 5 y 6, que no generó Elementos inválidos, por tanto, se trabajará sólo con un área prismática ubicada en la zona superior por su alta densidad de zonas cóncavas, como en Figura 13.

Como se muestra en Figura 19 los resultados, todas las instancias lograron “mallas solución” en por lo menos 6 iteraciones.

5.3.4. Malla de coxis

Para la representación del Coxis, inicialmente no se utilizó una zona de refinamiento adicional al comienzo de las pruebas porque como se muestra en la Figura 14, es evidente la



(a) Representación Moai con refinamiento en zona inferior.
 (b) Representación Moai con refinamiento en zona superior.

Figura 12: Diferentes localidades de refinamiento en Moai.
 Fuente: Elaboración propia.

amplia densidad de Elementos y concavidades en la malla. Esto genera Elementos inválidos sólo aplicando el algoritmo de generación mallas. Entonces, se escogieron niveles de refinamiento global que permitieran ejecutar el algoritmo de forma eficaz, esto es entre los niveles 4 y 9, el nivel cuatro se escogió porque genera una malla con elemento inválidos pero al momento de visualizarla genera una representación con falta de detalle, no representa muy bien la realidad. El nivel nueve se escogió como cota superior porque genera un error en el sistema debido al alto número de nodos en la malla.

Para esta sección de la validación entonces se utilizaron los niveles 5 y 8 como cotas superior e inferior y se definieron dos casos con refinamiento global y local para una superficie prismática ubicada en la parte superior de la cadera, y tres casos con refinamiento global solamente, describiéndolas respectivamente como 4r7, 5r8 y 6,7 y 8

Es necesario mencionar, que se realizaron pruebas con los casos 6r8 y 5r9, pero no generaron una ejecución válida del algoritmo. Además, se colorearon los Octantes a refinar en cada iteración y siempre mostraban presencia en las articulaciones de la cadera y columna, zonas bastante pequeñas para el alcance de los niveles de refinamiento.

Es incluso una malla interesante de analizar, se logró mallas con diversos niveles de refinamiento. Niveles seis, siete, ocho y nueve de refinamiento global, además de agregar una sección prismática en la zona superior del coxis, cubriendo por completo las zonas más pequeñas de la malla, en ningún caso se logró la convergencia del algoritmo.

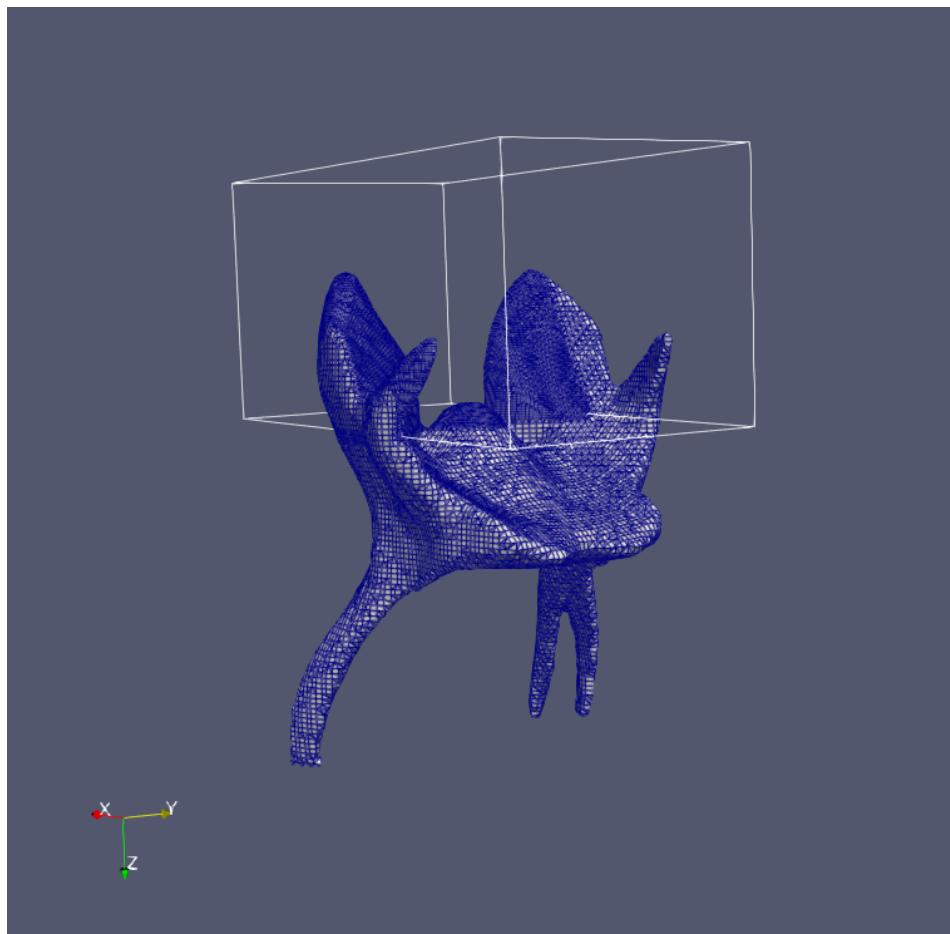


Figura 13: Representación Paladar con refinamiento en zona superior.
Fuente: Elaboración propia.

5.4. Resultados

Para analizar el comportamiento del algoritmo, nos enfocaremos en ajustar la frecuencia de Elementos inválidos a algún modelo conocido, entregar una hipótesis sobre la anomalía en la malla del Coxis y comparar con los resultados entregados en [Daines y Lobos, 2018].

5.4.1. Ajuste de la cantidad de iteraciones

El algoritmo propuesto se comporta de muy buena manera para todas las mallas a excepción de la que representa el Coxis. Cuando el algoritmo logra converger en una malla válida, es posible reducir considerablemente la cantidad de Octantes por refinar a lo más diez iteraciones, a simple vista se podría entregar una hipótesis sobre el comportamiento del algoritmo en cada iteración.

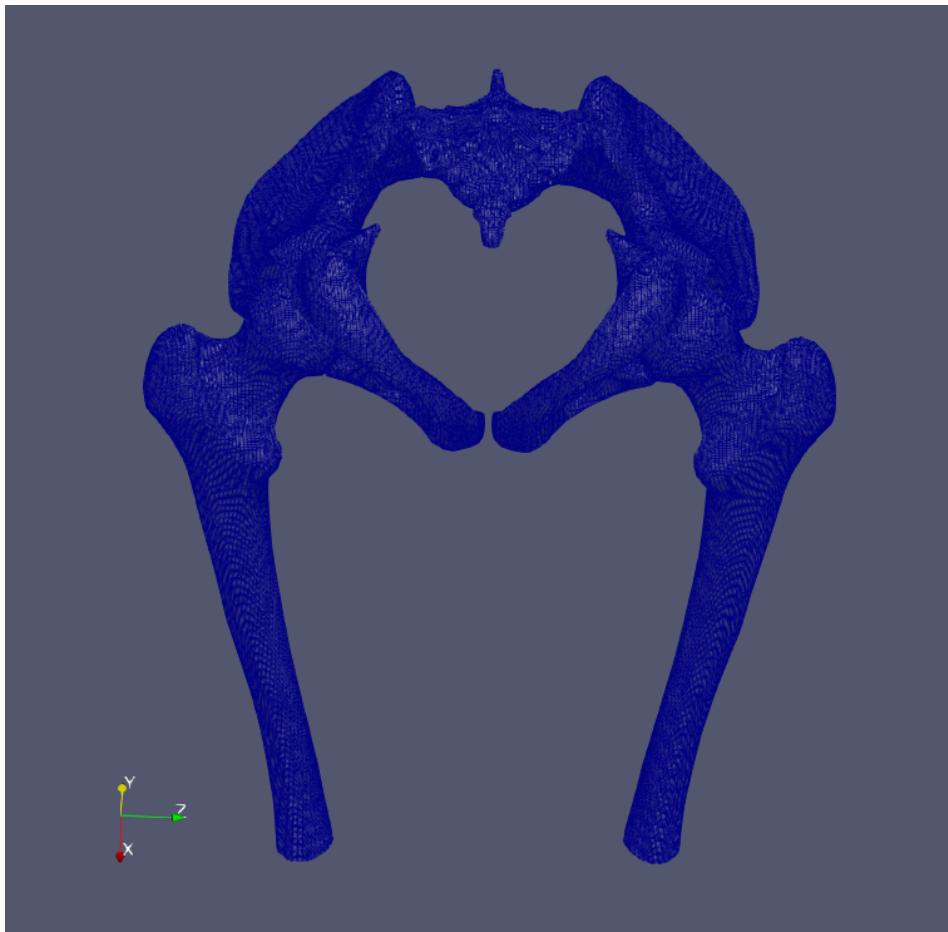


Figura 14: Representación Coxis sin refinamiento local.
Fuente: Elaboración propia.

Para esto, se considerará, a diferencia de lo que ya es tónica en lo antes mostrado, la frecuencia de Elementos que cumplen ciertas condiciones respecto a su calidad, para referirse a la frecuencia de Elementos, se denotará como E . Luego, para el análisis sobre ajuste del comportamiento del algoritmo, se considerará, la frecuencia de todos los Elementos que pertenecen a la malla y tienen $J_{ENS} \leq 0$, esto será E^0 .

Se elaboró la siguiente tabla con el E^0 en cada iteración para las diferentes mallas de los casos definidos, Tabla 1. Lo evidente es la relación estrecha entre Elementos y Octantes, dicho comportamiento se respalda en su naturaleza estructural. Luego, la aparente relación en el comportamiento de E en todos los casos, permite comenzar la búsqueda del modelo, por uno exponencial.

En la Tabla 1 se muestra el comportamiento de E^0 en cada iteración, de aquí solamente se analizarán los casos que convergen a cero.

Número de Elementos por refinar en cada iteración.						
Nº Iteración	cortex_5r7	cortex_5r7_2	moai_5r7	moai_5r7	palate_6r7	coxis_7
1	27	64	26	15	25	242
2	7	32	7	12	5	658
3	2	13	2	11	3	435
4	1	6	1	3	1	612
5	0	7	1	0	1	679
6		3	0		0	746
7		0				880
8						1017
9						1179
10						1302

Tabla 1: E^0 en cada iteración para los diferentes casos.

5.5. Análisis de los resultados

5.5.1. Análisis de la tasa de reducción de la cantidad de Elementos por refinar en cada iteración

- Definición de una secuencia exponencial.

Para determinar si E se comporta de manera exponencial, es esencial revisar cómo se comportan las reducciones entre cada par de números consecutivos y si sigue un patrón consistente con un modelo exponencial.

Un modelo exponencial sigue la forma:

$$y = a \cdot r^n$$

donde:

- y es el valor en la n-ésima posición.
- a es el valor inicial (en nuestro caso, 21).
- r es la razón de reducción (un valor menor a 1).
- n es la posición en la secuencia.

Para verificar si una secuencia sigue un modelo exponencial, podemos examinar si el cociente entre cada par de términos consecutivos es aproximadamente constante. Es decir, si $\frac{y_{n+1}}{y_n}$ es constante.

- Cálculo del cociente entre términos consecutivos

Calculamos el cociente entre cada par de números consecutivos en la secuencia de cada uno de los casos. Tabla 2

Cociente entre cantidad de Elementos por refinar consecutivos					
Nº Iteración	cortex_5r7	cortex_5r7_2	moai_5r7	moai_5r7	palate_6r7
1-2	0.259	0.500	0.269	1.25	0.2
2-3	0.286	0.406	0.285	0.733	0.6
3-4	0.500	0.462	0.5	0.273	0.333
4-5		1.166	1.0		1.0
5-6		0.429			

Tabla 2: Cocientes entre cantidades de Elementos con $J_{ENS} \leq 0$ para los diferentes casos.

- Evaluación de los Cocientes.

Para un modelo exponencial puro, los cocientes entre términos consecutivos deberán ser aproximadamente iguales. En este caso, los valores no lo son, pero están en un rango que sugiere una tendencia decreciente. Sin embargo, la variabilidad entre los cocientes da indicios de que la secuencia no se reduce de manera perfectamente exponencial con una sola razón común.

Por ejemplo, en el caso de cortex_5r7_2, sus cocientes se mantienen relativamente constante, aproximadamente 0.45, a excepción del cociente en las iteraciones 4-5 que se escapa casi al doble del promedio.

- Ajuste a un modelo exponencial

Se ajustará a un modelo exponencial cada uno de los casos. Considerando a_i , valor inicial del caso i , se buscará una razón r_i que mejor ajuste la reducción de la secuencia para el caso.

Para esto, es necesario resolver la ecuación

$$y = a_i \cdot r_i^n$$

para cada n y encontrar el r_i promedio que mejor describa la secuencia. Aunque los cocientes no son constantes, podemos ver si hay un \bar{r}_i promedio que se ajuste razonablemente.

- Cálculo promedio geométrico

El promedio geométrico de los cocientes es una manera de obtener una aproximación de la razón \bar{r}_i :

Su valor se calcula en Tabla 3.

Estos \bar{r}_i indican que las secuencias podrían ajustarse aproximadamente por un modelo exponencial con razones cercanas a las propuestas en la tabla 3.

Gráficamente podemos ver el comportamiento de E^0 en cada iteración sobre los datos reales en Figura 15. La gran mayoría de los casos se ajustan de buena manera, podríamos decir que el algoritmo se comporta bajo un modelo exponencial, sin considerar el caso excepcional del coxis. A diferencia de los demás casos, la malla del coxis diverge

Promedio geométrico de los cocientes de cada caso	
Casos i	Promedio geométrico r_i
cortex_5r7	0.333
cortex_5r7_2	0.542
moai_5r7	0.443
moai_5r7_2	0.630
palate_6r7	0.408

Tabla 3: Promedio geométrico de los cocientes de cada caso.

también de manera exponencial. Esto puede explicarse debido al comportamiento en cascada que tiene el algoritmo y la estructura continua de las vecindades cóncavas.

5.5.2. Análisis del efecto en los Elementos adyacentes

Cuando se refina cada Octante con Elementos inválidos, genera una reacción en cadena de refinamiento en los Octantes adyacentes para mantener la consistencia en los niveles de refinamiento de la malla. Debe cumplirse que, dado un Octante Objetivo O con nivel de refinamiento rl_o , deben, sus vecinos directos, \bar{O} , tener un nivel de refinamiento $rl_{\bar{o}} \in \{rl_o + 1, rl_o - 1, rl_o\}$.

Dado lo anterior, supone que cuando se refina un Octante, no debiera provocar un efecto avasallador en la calidad de los Elementos vecinos, sobre todo porque suponiendo que existen Elementos inválidos en la vecindad cercana que es el peor de los casos, al generar el refinamiento en cadena, debería mejorar al igual como sucede con el Octante objetivo.

Para validar la anterior hipótesis, se generará una serie de Histogramas agrupados por iteración de cada malla y una muestra general de los Histogramas.

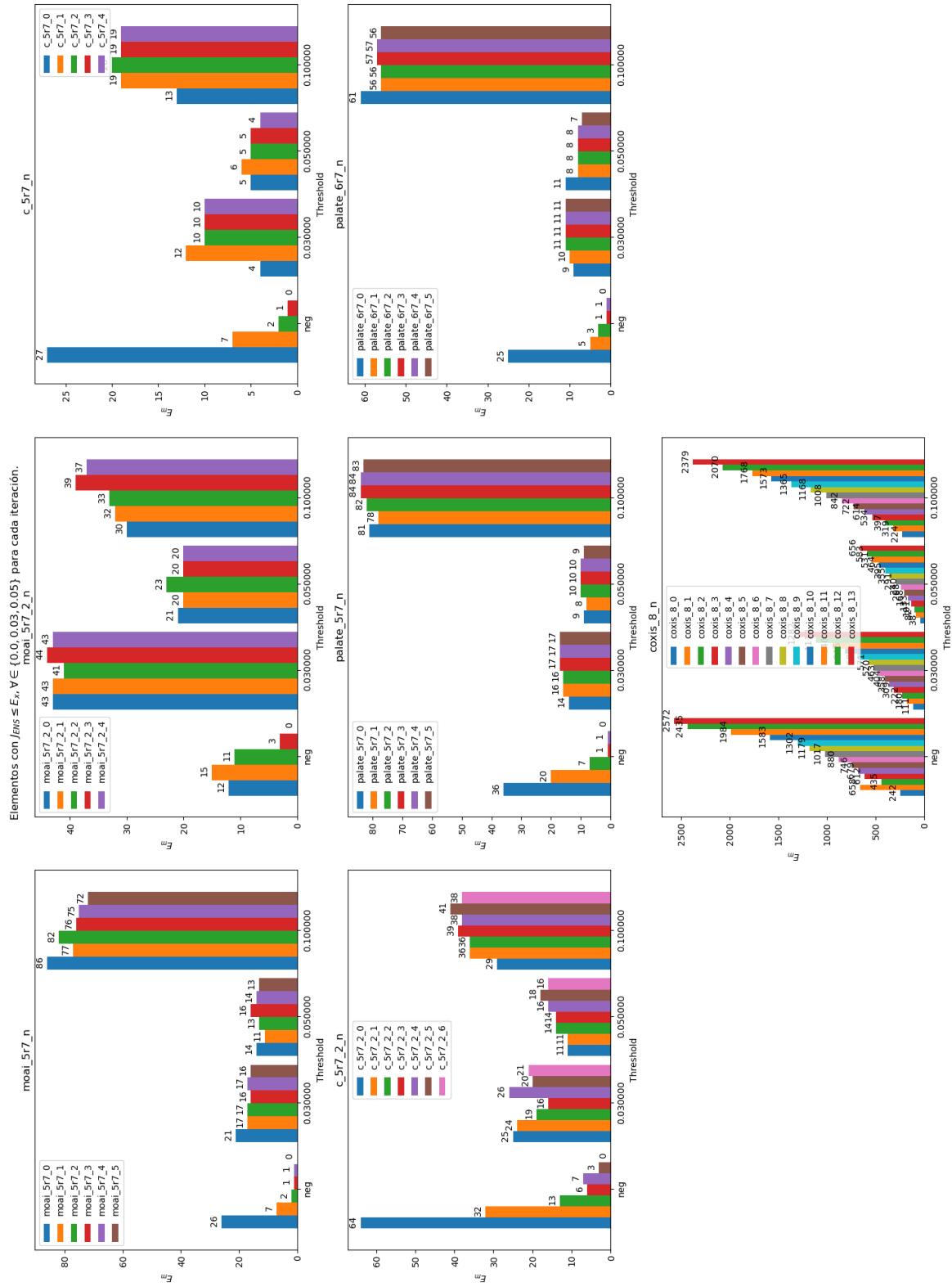


Figura 16: Histograma agrupado por intervalo para todos los casos.
Fuente: Elaboración propia.

Se puede ver como en la mayoría de los casos válidos, se mantiene constante o varía levemente considerando el orden de magnitud de los Elementos que componen las mallas.

Para la malla del Moai 1 con threshold 0, Figura 17, que contiene 693 Elementos, el efecto en cadena generá un delta positivo para el mejoramiento de la malla de aproximadamente 14 Elementos en el intervalo $E_1^{0,05}$ entre las iteraciones 0 y 5, que decae de 86 a 72 Elementos, es decir, app el 2 % de la cantidad de Elementos de la malla.

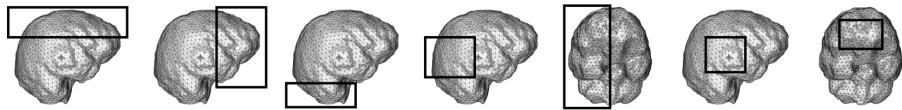
Para la malla del Moai 2 con threshold 0, el efecto generará un delta negativo para el mejoramiento de app 9 Elementos en el intervalo $E_1^{0,05}$ entre las iteraciones 0 y 3, app el 1,3 % de los Elementos de la malla. Pero, si nos fijamos en las mismas iteraciones 0 y 3, en los Octantes Objetivo, que serían los Elementos Inválidos, el delta es positivo para el mejoramiento, disminuyendo drásticamente en la iteración 3, pasando de 12 Elementos en la iteración 0 a 3 Elementos en la iteración 3, algo muy similar al delta del intervalo $E_1^{0,05}$.

Un caso similar, es el de la Corteza 1 para threshold 0, Figura 18, que en las iteraciones 0 y 5, presenta un delta negativo para el mejoramiento de app 11 Elementos en el $E_1^{0,05}$, pero en el intervalo $E_0^{-\infty}$ el delta es positivo para el mejoramiento, reduciendo la cantidad de Elementos Inválidos de 64 a 3, casi 5 veces la cantidad de Elementos que aumentaron en el intervalo $E_1^{0,05}$.

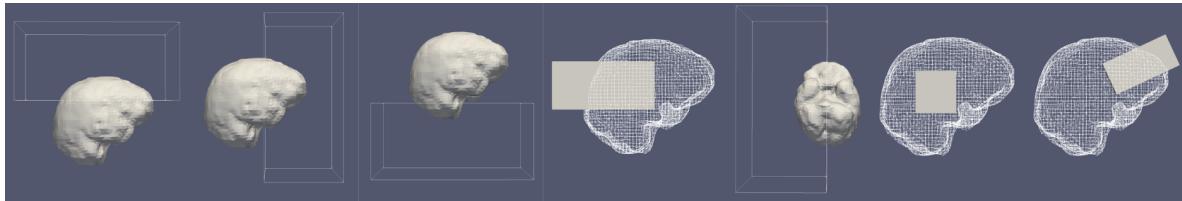
Con lo anterior, se puede determinar que muy probablemente los Elementos mejorados se distribuyen en los intervalos de calidad superiores, incluso cuando se utilizan threshold mayores a 0, es muy difícil que la calidad en los intervalos inferiores al threshold aumenten su frecuencia de Elementos, es decir, que los Elementos empeoren su calidad.

5.5.3. Análisis de calidad frente al Algoritmo de proyección de nodos.

Para analizar la calidad de las mallas generadas por el algoritmo propuesto en este estudio, se utilizará el estándar fijado en el proceso de validación en [Daines y Lobos, 2018], es decir, se utilizarán los tres tipos de refinamientos que soporta el algoritmo, general, local y superficie. Se definirán diferentes regiones de interés (ROI) mostradas en Figura 21, respecto a dichas regiones se realizará una homologación, por tanto, los resultados no serán comparables con total precisión. Para reducir la cantidad de iteraciones generadas por el efecto dominó, se utilizará el mismo criterio, un $RL = n$ en la ROI, un $RL = n - 1$ en la superficie de Ω y un $RL = n - 2$ en el resto de la malla. Se denominará “cortex_ref_reg” a las instancias, donde ref es el RL usado por el ROI, y reg el índice identificador de las regiones de interés. En este caso, se utilizará $RL = 5$ para las *regiones*, $RL = 4$ para la superficie y $RL = 3$ para todo el resto. Se definió un máximo de 20 iteraciones, debido al comportamiento inestable que presentaban algunas iteraciones cuando quedaba sólo un Octante por refinar.



(a) Regiones definidas por [Daines y Lobos, 2018].



(b) Regiones homologadas.

Figura 21: Regiones de interés usados en instancias de corteza. De izquierda a derecha se identificarán como Región 0 a 6. En (a) se encuentra la referencia utilizada en [Daines y Lobos, 2018], en (b) homologadas en la práctica.

Fuente: Elaboración propia.

El resultado se presenta en Tabla 4. Para cada instancia, se disponen cinco filas con la distribución de la calidad de los Elementos. En la primera fila se muestra la distribución del algoritmo original. Luego, las 2 filas siguientes representan a las pruebas realizadas con algoritmo de proyección de nodos con threshold 0,03 y 0,05, respectivamente y finalmente las 2 filas siguientes corresponden a las pruebas realizadas por el algoritmo propuesto con los mismos threshold definidos para el algoritmo de proyección de nodos.

En la primera columna se muestra el J_{ENS} mínimo de la malla generada, es decir, el Elemento de peor calidad, luego se muestran las frecuencias de los Elementos que pertenecen a dicho intervalo de calidad desde Elementos de calidad negativa hasta $J_{ENS} = 0,2$. Al final, la última columna nos muestra la cantidad total de Elementos de la malla.

Instance	$J_{ENS}min$	Inv	[0, 0.03[[0.03, 0.05[[0.05, 0.1[[0.1, 0.15[[0.15, 0.2[Total
cortex_5_0	-0.009743	1	8	8	14	35	31	12895
	0.037720	0	0	7	9	34	38	12891
	0.053261	0	0	0	7	35	42	12875
	0.039391	0	0	3	13	20	32	13322
	0.051298	0	0	0	12	21	33	13704
cortex_5_1	-0.000639	1	7	2	9	18	29	12005
	0.041439	0	0	2	9	18	24	11983
	0.058775	0	0	0	9	18	25	11988
	0.031648	0	0	3	8	24	49	15686
	0.062543	0	0	0	13	25	52	15828
cortex_5_2	0.002585	0	3	2	13	22	36	8329
	0.030672	0	0	4	13	21	36	8327
	0.059482	0	0	0	13	22	38	8290
	0.031067	0	0	3	11	19	40	8448
	0.058946	0	0	0	12	19	43	8515
cortex_5_3	-0.999649	3	0	1	11	44	86	11902
	0.041507	0	0	2	12	47	79	11903
	0.000000	0	2	2	22	48	84	11886
	0.032822	0	0	1	14	27	100	14340
	0.051203	0	0	0	15	28	103	14378
cortex_5_4	-0.222353	2	2	1	9	34	63	18513
	0.042440	0	0	1	7	32	64	18506
	0.056449	0	0	0	7	31	67	18495
	0.055131	0	0	0	5	16	50	18661
	0.055131	0	0	0	5	16	50	18661
cortex_5_5	-0.100899	2	2	0	8	30	55	9866
	0.072093	0	0	3	9	33	57	9871
	0.072093	0	0	0	7	33	57	9871
	0.038194	0	0	2	8	15	95	13248
	0.050810	0	0	0	8	14	94	13294
cortex_5_6	-0.491747	9	0	0	2	47	71	9635
	-0.981794	11	1	3	19	55	79	9599
	-0.981794	11	1	3	19	55	79	9599
	0.038194	0	0	1	3	10	64	10076
	0.007059	0	0	0	3	10	64	10257

$J_{ENS}min$: Jens mínimo, el Elemento de peor calidad.

Inv: Elementos con calidad menor a 0.

Total: Cantidad de Elementos que componen la malla.

En cada instancia:

- Primera fila: Algoritmo Original.
- Segunda fila: Algoritmo de proyección de nodos con threshold 0,03.
- Tercera fila: Algoritmo de proyección de nodos con threshold 0,05.
- Cuarta fila: Algoritmo de refinamiento de octantes con threshold 0,03.
- Quinta fila: Algoritmo de refinamiento de octantes con threshold 0,05.

Tabla 4: Comparación de calidad de los resultados entre los algoritmos original, proyección de nodos y refinamiento de octantes.

Se añadió el algoritmo propuesto a la comparación de tiempo y calidad realizada por [Daines y Lobos, 2018], en la Tabla 5, se evidencia, al igual que en la tabla referenciada anteriormente, las 7 regiones estudiadas para la corteza cerebral, en cada fila que representa a cada instancia, se encuentran tres filas, la primera muestra la calidad y tiempo de ejecución de la instancia con el algoritmo original, en la segunda fila se utiliza el algoritmo de proyección de nodos y en la tercera fila el algoritmo propuesto.

Para medir la calidad, se utilizó el J_{ENS} mínimo de todos los Elementos de la malla resultante y para el tiempo, medido en segundos, se definirá como el tiempo por iteración $\frac{T}{it}$, considerando como el tiempo promedio de cada iteración, esta expresión no incluye la primera iteración

porque se considera aproximadamente similar al tiempo de ejecución del algoritmo original, se define formalmente en Ecuación 4.

$$\frac{T}{it} = \frac{TO - TP}{TP - 1} \quad (4)$$

Instance	J_{ENSmin}	T [s]	It	T/it [s]
cortex_5_0	-0.009743	2.312	-	-
	0.037720	2.917	2	0.605
	0.039391	2.735	4	0.683
cortex_5_1	-0.000639	2.111	-	-
	0.041439	2.698	2	0.587
	0.031648	3.879	5	0.775
cortex_5_2	0.002585	1.776	-	-
	0.030672	2.247	2	0.471
	0.031067	1.015	2	0.507
cortex_5_3	-0.999649	2.060	-	-
	0.041507	2.595	2	0.535
	0.032822	2.655	4	0.663
cortex_5_4	-0.222353	2.819	-	-
	0.042440	3.637	2	0.818
	0.055131	1.788	2	0.894
cortex_5_5	-0.100899	1.813	-	-
	0.072093	2.275	2	0.462
	0.038194	5.743	9	0.638
cortex_5_6	-0.491747	1.800	-	-
	-0.981794	3.637	5	0.459
	0.038194	4.167	8	0.520

J_{ENS} : Jens mínimo, el Elemento de peor calidad.

T : Tiempo total de la ejecución del algoritmo.

It : Cantidad de iteraciones realizadas.

Tabla 5: Comparación en tiempo y calidad entre los algoritmos original, proyección de nodos y refinamiento de octantes, con threshold 0,03.

Al analizar los resultados de calidad y tiempo, se puede notar que el tiempo total de las ejecuciones de los casos de prueba, aumentan considerablemente, incluso en algunos casos aumentando un 100 % del tiempo de la ejecución del algoritmo de proyección de nodos, pero, si bien el tiempo total aumenta, el tiempo por iteración, se mantiene relativamente similar.

Este algoritmo y el propuesto para proyectar los nodos presentan la misma complejidad computacional $O(n \cdot 2M)$, siendo M la cantidad de nodos y n la cantidad de iteraciones,

por tanto, ambos debiesen presentar tiempos por iteración lineales y similares. Evidencia de lo anterior, es la similitud entre estos factores para ambos algoritmos, con una variación del orden de las décimas de segundo por iteración.

Por otra parte, el J_{ENSmin} para el algoritmo propuesto se mantiene más cercano al threshold que el algoritmo de proyección de nodos, esto podría interpretarse como un efecto en la vecindad por parte de este último algoritmo, no así el propuesto, que en caso de que la calidad de los Elementos no presente calidad inferior a la cota, entonces no aplicará refinamiento, es decir, priorizará mantener la calidad existente mientras no existan Elementos de peor calidad que el threshold.

CAPÍTULO 6

Conclusiones

El algoritmo propuesto fue capaz de eliminar Elementos inválidos en la mayoría de los casos, sin modificar o provocar estiramiento en los nodos de la representación ni empobrecimiento de la calidad.

Al realizar la validación del algoritmo, solo una de las instancias de prueba falló. Cuando se utilizó diversos umbrales de calidad y regiones de interés, no hubo cambios considerables en el comportamiento del algoritmo ni degradación en las estadísticas J_{ENS} .

Si bien los tiempos no son óptimos considerando la complejidad del algoritmo, se logró el objetivo de este estudio, implementar y validar el algoritmo propuesto, no así la implementación.

Se recomienda, fijar la cantidad de iteraciones en 10, ya que, la mayoría de los casos se obtuvieron mallas válidas en a lo más 7 iteraciones. Cuando requería más iteraciones, las pruebas normalmente fallaban, como en el caso de la representación del Coxis, que diverge y genera mallas de peor calidad.

Por tanto, agregar una condición de término que realice el seguimiento del deterioro de la calidad de la malla y retorno automatizado a un estado anterior, es fundamental para casos como el “moai_5_5r5_1_n” registrado en Figura 17. Aquí se muestra un posible punto intermedio en la iteración 8, donde se logran cero Elementos inválidos y dos Elementos en $E_0^{0,03}$ y uno en $E_0^{0,05}$, una calidad aceptable para un threshold 0,05.

- no leer archivos con modelo de ddatos en cada iteración - no escribir en cada iteración las representaciones gráficas de las mallas, sólo hacerlo en la malla final

intro

objetivos porqué

discusión sobre los resultados qué significó para mi (proceso de implementación) lista problemas no resueltos o complejidades de la solución

mejoras priorización del trabajo futuro

ANEXOS

En los Anexos se incluye todo aquel material complementario que no es parte del contenido de los capítulos de la memoria, pero que permiten a un lector contar con un contenido adjunto relacionado con el tema.

Referencias Bibliográficas

- [Daines y Lobos, 2018] Daines, E. y Lobos, C. (2018). Repairing octree boundary transition regions composed of different types of elements. En *2018 37th International Conference of the Chilean Computer Science Society (SCCC)*, pp. 1–8. IEEE.
- [González y Lobos, 2014] González, E. y Lobos, C. (2014). A set of mixed-element transition patterns for adaptive 3d meshing. Technical Report 2014/01, Departamento de Informática, UTFSM.
- [Lobos, 2013] Lobos, C. (2013). A set of mixed-elements patterns for domain boundary approximation in hexahedral meshes. *Studies in health technology and informatics*, 184:268–72.
- [Lobos y González, 2015] Lobos, C. y González, E. (2015). Mixed-element octree: a meshing technique toward fast and real-time simulations in biomedical applications. *International journal for numerical methods in biomedical engineering*, 31(12).
- [SandiaLabs, 2007] SandiaLabs (2007). The verdict mesh verification library. <https://github.com/sandialabs/verdict>.
- [Shepherd y Johnson, 2008] Shepherd, J. y Johnson, C. (2008). Hexahedral mesh generation for biomedical models in scirun. *Engineering with Computers*, 25(1):97–114.

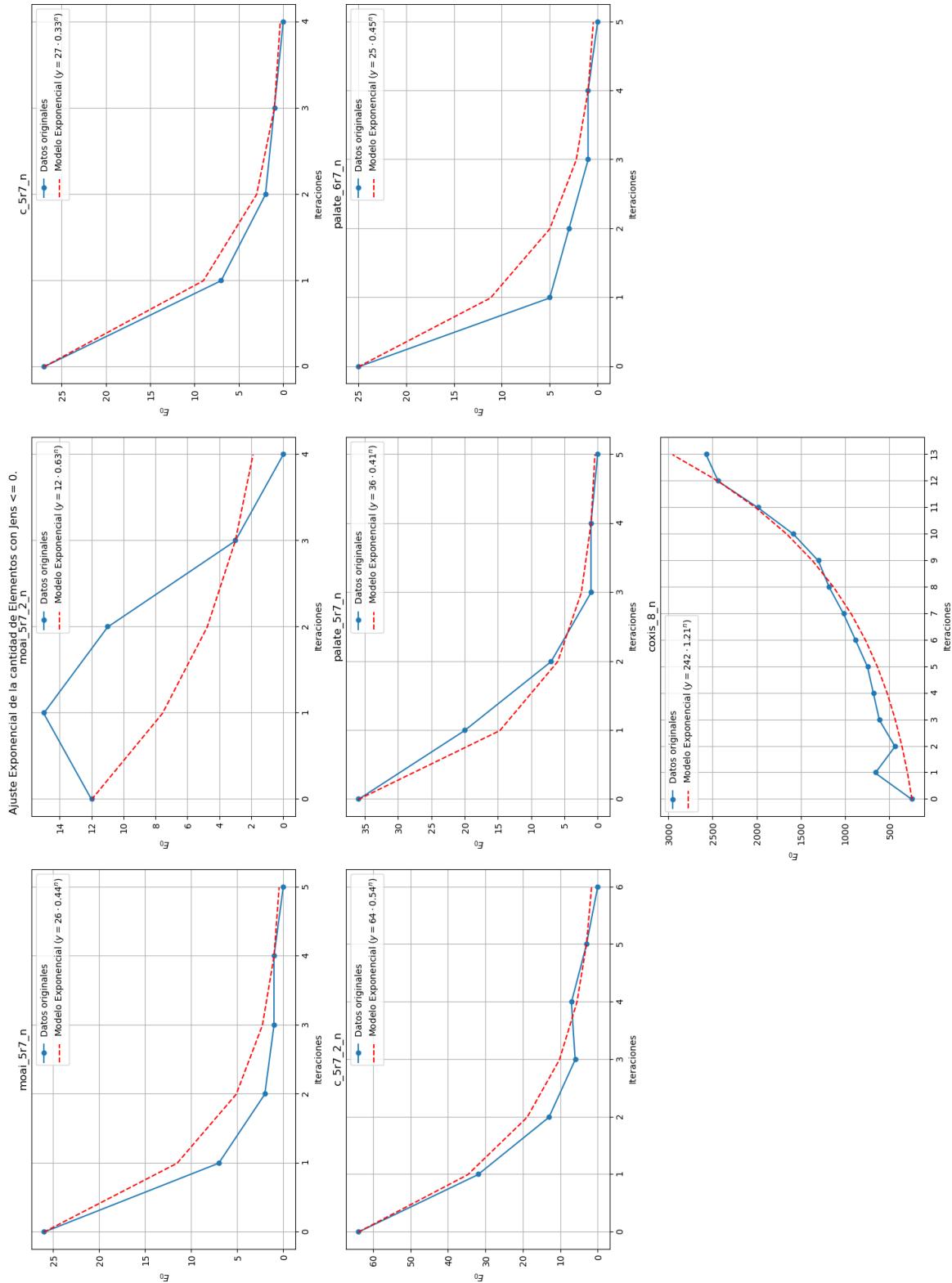


Figura 15: Ajuste exponencial de E^0 para todos los casos.
Fuente: Elaboración propia.

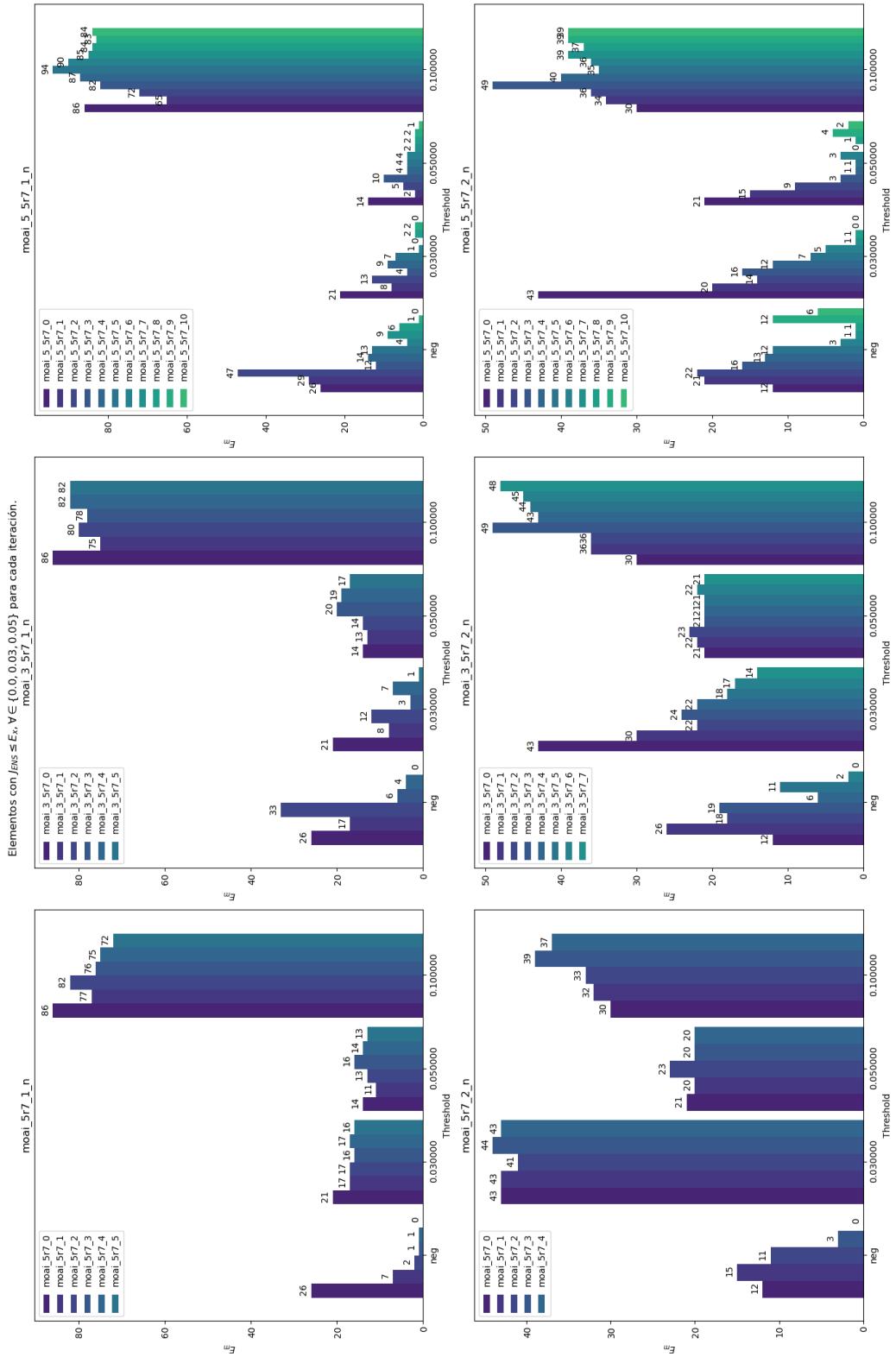


Figura 17: Histograma agrupado por intervalo para los dos casos de la malla Moai.
Fuente: Elaboración propia.

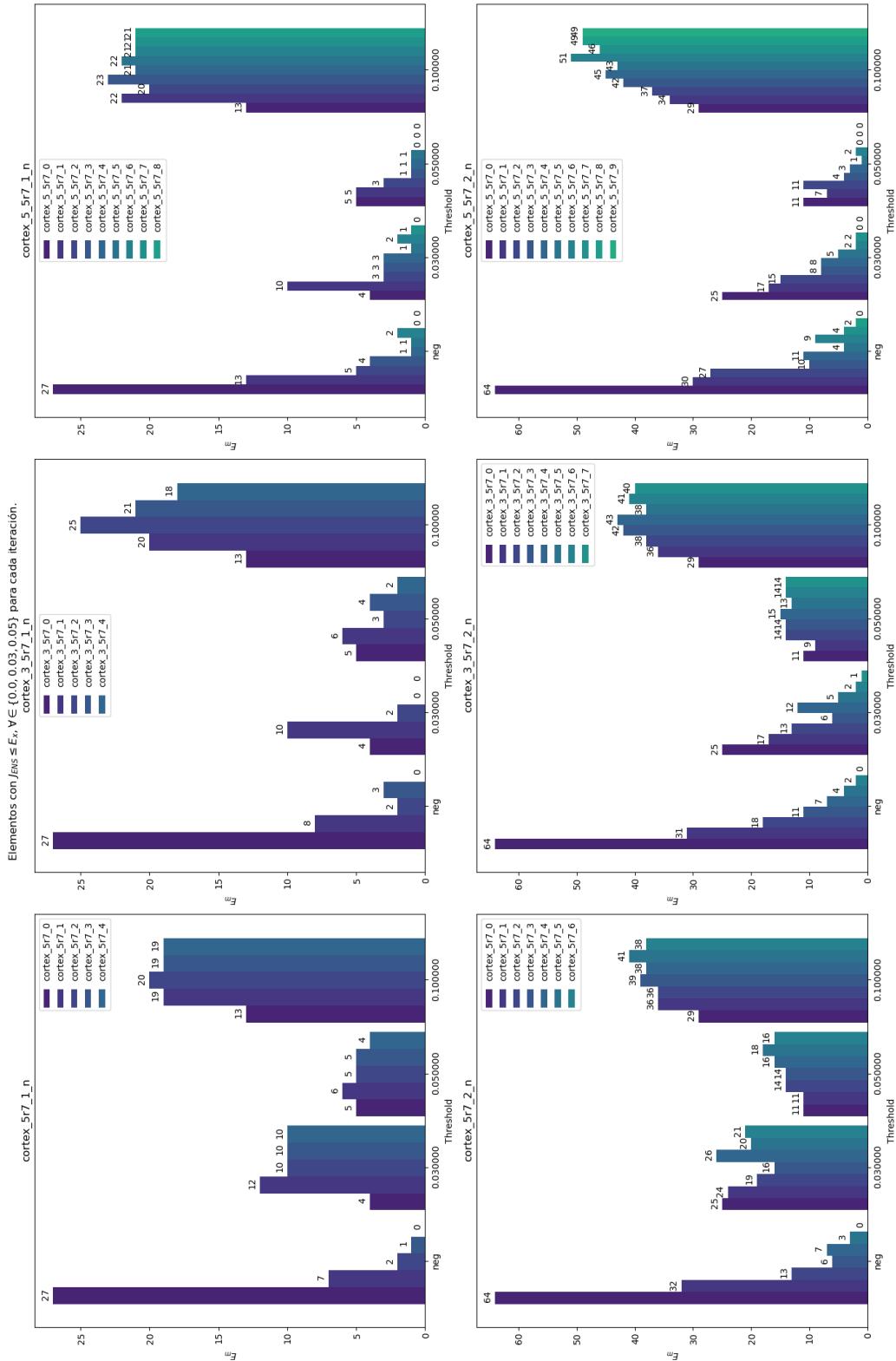


Figura 18: Histograma agrupado por intervalo para los dos casos de la malla Corteza.
Fuente: Elaboración propia.

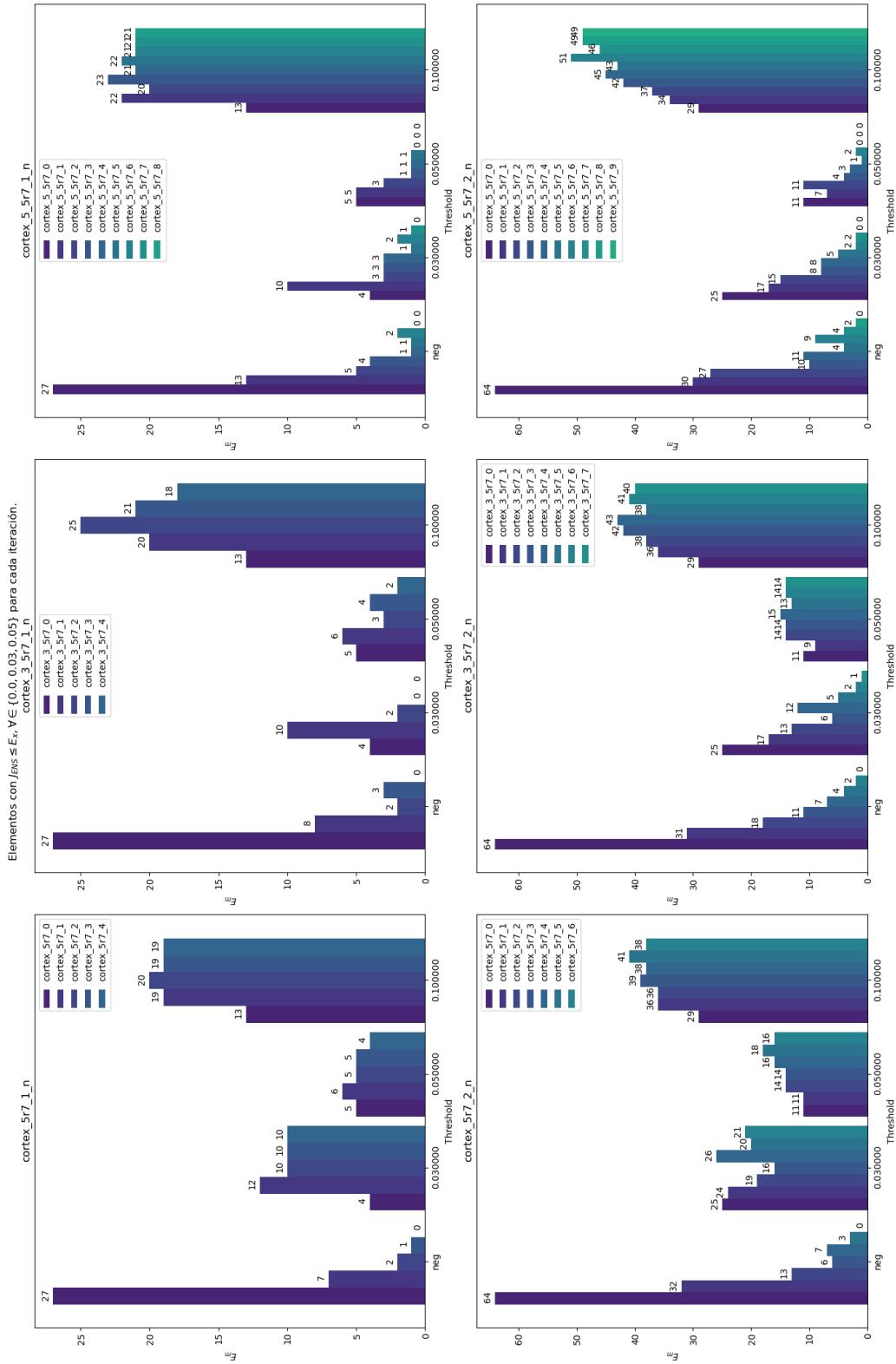


Figura 19: Histograma agrupado por intervalo para los dos casos de la malla Paladar.
Fuente: Elaboración propia.

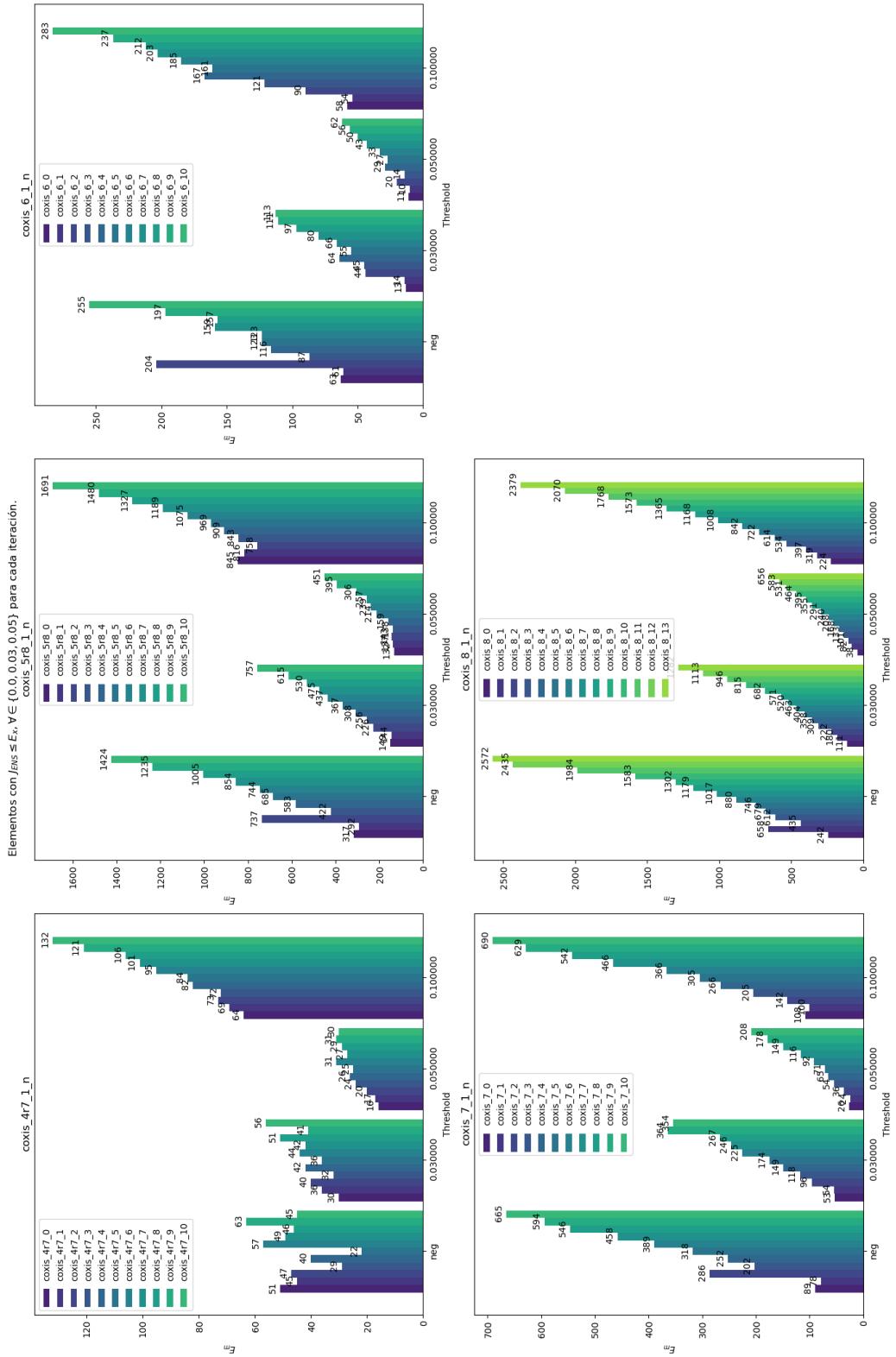


Figura 20: Histograma agrupado por intervalo para los dos casos de la malla Coxis.
Fuente: Elaboración propia.