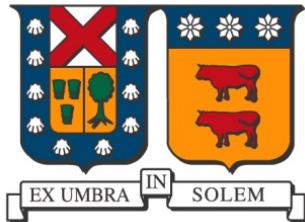


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA  
DEPARTAMENTO DE INFORMÁTICA  
VALPARAÍSO - CHILE



“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

Noviembre - 2022

## 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

<b>Resumen</b>	I
<b>Abstract</b>	I
<b>1 Glosario</b>	II
<b>Índice de Figuras</b>	v
<b>Índice de Tablas</b>	v
<b>Introducción</b>	1
<b>Capítulo 2: Definición del problema</b>	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 . . . . .	5
2.6.1 Objetivo general . . . . .	5
2.6.2 Objetivos específicos . . . . .	6
<b>Capítulo 3: Marco conceptual</b>	7
3.1 Medida de calidad para elementos en Mallas Mixtas . . . . .	7
3.2 Malla inicial . . . . .	9
3.3 Herramientas . . . . .	9
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 . . . . .	12
3.4.1 Acercamiento al algoritmo de Mixed-element mesh generator . . . . .	13
<b>Capítulo 4: Propuesta de solución</b>	15
4.1 Comprensión del problema . . . . .	15
4.2 Comprensión de elementos inválidos . . . . .	17
4.3 Análisis de la vecindad . . . . .	17
4.4 Propuesta . . . . .	18
4.4.1 Generación de malla inicial. . . . .	19
4.4.2 Identificación y etiquetado de Octantes. . . . .	21
4.4.3 Refinamiento de Octantes Identificados. . . . .	22
4.4.4 Ciclo Iterativo de Evaluación y Refinamiento. . . . .	23
<b>Capítulo 5: Validación de la solución</b>	26

5.1	Stack de tecnologías . . . . .	26
5.2	Proceso de análisis . . . . .	26
5.3	Ejecución . . . . .	27
5.3.1	Malla de corteza cerebral . . . . .	27
5.3.2	Malla de Moai . . . . .	28
5.3.3	Malla de paladar . . . . .	30
5.3.4	Malla de coxis . . . . .	32
5.4	Resultados . . . . .	32
5.4.1	Ajuste de la cantidad de iteraciones . . . . .	32
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 comportamiento de los Elementos posicionados en la vecindad a las zonas refinadas. . . . .	36
<b>Capítulo 6: Conclusiones</b>		<b>37</b>
<b>ANEXOS</b>		<b>38</b>
<b>Referencias Bibliográficas</b>		<b>39</b>

## Índice de Figuras

1	Elementos básicos . . . . .	3
2	Malla Octree parcialmente refinada. . . . .	4
3	Patrón de transición. . . . .	5
4	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. . . . .	9
5	Interfaz de software Paraview, se muestra una sección de la malla inicial con dos elementos seleccionados. Fuente: Elaboración propia. . . . .	12
6	Zona frontal de la malla donde se presentan los elementos inválidos (verde). . . . .	17
7	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. . . . .	18
8	Perspectivas de vecindario (gris) donde se encuentra un par de elementos inválidos (verde). . . . .	19
9	Diferentes localidades de refinamiento en corteza cerebral. . . . .	30
10	Diferentes localidades de refinamiento en Moai. . . . .	30
11	Representación Paladar con refinamiento en zona superior. . . . .	31
12	Representación Coxis sin refinamiento local. . . . .	33
13	Ajuste exponencial de $E_0$ para todos los casos. . . . .	40
14	Histograma agrupado por intervalo para todos los casos. . . . .	41

## Índice de Tablas

1	$E_0$ en cada iteración para los diferentes casos. . . . .	34
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

## 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, 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, primas, pirámides, etc. Esta aproximación mediante elementos geométricos recibe el nombre de malla geométrica.

#### 2.2. Mallas geométricas

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

Existen dos tipos de mallas geométricas, 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 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.

En esta memoria se utilizará la técnica Octree con elementos mixtos, malla Octree mixta, para representar y refinar las mallas geométricas volumétricas.

#### 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 en árbol de datos en la que cada nodo de este 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 Fig. 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.

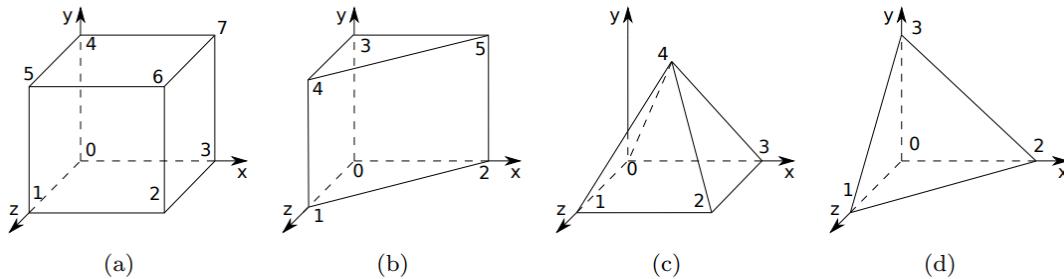


Figura 1: Elementos básicos  
(a) hexaedro, (b) prisma (cuña), (c) pirámide y (d) tetraedro.  
Fuente: [González y Lobos, 2014]

Supongamos que  $\Omega$  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  $\Omega$  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  $\Omega$ . 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  $\Omega$ , 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 Fig.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 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.

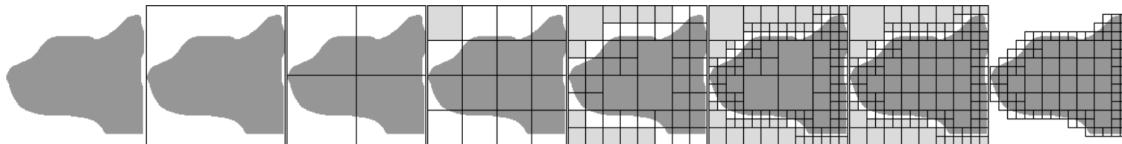


Figura 2: Malla Octree parcialmente refinada.

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  $\Omega$  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  $\Omega$  tienen su refinamiento detenido.

Fuente: [Lobos y González, 2015]

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 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 Fig. 3 se muestra un ejemplo de patrón de transición.

Una malla es conforme cuando es topológicamente correcta, es decir, tiene todos sus nodos de los bordes y caras consistentes para todos los elementos adyacentes.

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 Fig. 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, hexaedros, prismas y pirámides de base cuadrada. Todos los patrones existentes fueron diseñados y presentados en [González y Lobos, 2014]. 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.

Aquí se evidencia que los patrones de superficie para generar estas transiciones no están

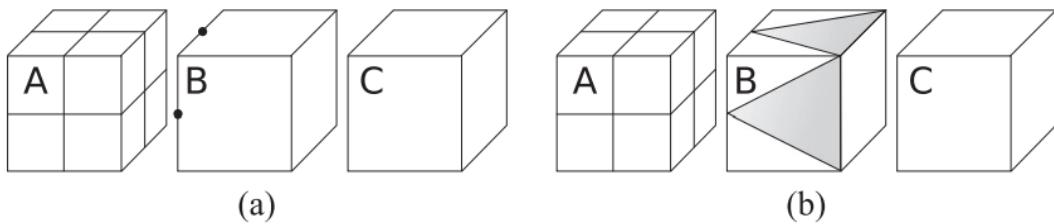


Figura 3: Patrón de transición.

(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].

diseñados para ser aplicados sobre elementos mixtos, sino que solamente para mallas compuestas por hexaedros. Esto puede surgir especialmente en sectores de la malla donde el dominio es cóncavo.

## 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

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.

## 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.

### 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 concavos.

Un octante límite es aquel que intersecta la frontera de  $\Omega$ , puede presentar nodos dentro y fuera del dominio  $\Omega$ , 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 [González y Lobos, 2014]. 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  $\Omega$ .

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.

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 ( $RA$ ). En general, el  $AR$  de un elemento es la relación entre su borde más corto y su borde más largo. Existen diferentes variaciones que utilizan otras distancias del elemento. Las longitudes utilizadas para el  $AR$  en este trabajo y el de [Daines y Lobos, 2018] se describen a continuación:

- Pirámide: Distancias entre centros de aristas opuestas en la base y la altura de la pirámide.
- Cuña: Distancias entre los centros de las aristas opuestas en las caras del cuadrilátero

y las distancias entre el centro de las caras del cuadrilátero y el centro de la arista opuesta.

- Hexaedro: Distancias entre centros de caras opuestas.

Otra variación significativa es el Aspecto Ratio Gamma (ARG). 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 dependencia dos elementos con los mismos ángulos internos pero 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 la malla.

Por tanto, para suplir la dependencia a las distancias se utiliza el Jacobiano Escalado, Scaled Jacobian ( $J_s$ ). 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 su peor  $J_S$  de sus nodos. En [Daines y Lobos, 2018] se define el rango de aceptación de la calidad de los elementos, un  $J_S < 0$  implica que el elemento está invertido,  $J_S \geq 0,2$  significa una buena calidad y un hexaedro perfecto cuando  $J_S = 1$ . Los elementos con  $J_S \in [0, 0, 2]$  se denominan cuestionables para los cálculos de [Shepherd y Johnson, 2008]

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 Elemento, más adelante denotado como  $J_{ENS}$ , Element Normalized Scaled Jacobian.

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

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

De esta forma, como se vió en [Daines y Lobos, 2018], con ayuda de estas constantes podemos calcular el  $J_{ENS}$  de cualquier nodo de un elemento sustituyendo el valor correspon-

diente en la Ec. 2. De forma similar a  $J_S$  para realizar el cálculo de  $J_{ENS}$  utilizamos la Ec. 3, donde  $i$  son los nodos del elemento  $e$ . y  $J_{ENS}^e$  es su  $J_{ENS}$ .

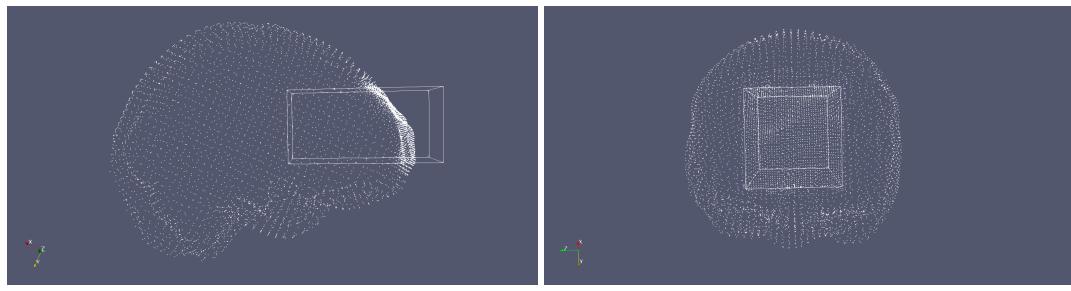
$$J_{ENS}^n = \begin{cases} (1 + K^e) - J_S, & J_S > k^e \\ J_S/K^e, & -k^e \leq J_S \leq K^e \\ -(1 + K^e) - J_S, & J_S < -k^e \end{cases} \quad (2)$$

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

### 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*.

Esta será la representación de un cerebro 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 4.



(a) Perspectiva lateral

(b) Perspectiva frontal

Figura 4: 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, un software para visualización, scripts en Bash y Python para la ejecución de la solución y el análisis de los resultados.

En esta sección se explicará la estructura necesaria para comprender la propuesta de solución, definición y usos de las otras herramientas.

### 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 linea 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 octantes entregados. Este software recibe modelos de datos de mallas en formato *mdl*.

```
1 input >      ./meshes -h
2 output >     use: ./meshes [-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.

Para exemplificar, en la figura Listing 2 se muestra el input de la herramienta para la generación de la *malla inicial*, se define la corteza cerebral, './data/cortex.mdl', con nivel 5 de refinamiento y superficie rectangular a refinar, './data/cortex\_surf\_roi.mdl', con nivel 7 de refinamiento, nombre de archivo para exportación 'c\_5r7', exportación en formato m3d y VTK, para visualización en Geomview o Paraview respectivamente.

Listing 2: Input mesher generator para la generación de la malla inicial.

Fuente: Elaboración propia.

```
input >      ./meshes -d ./data/cortex.mdl -s 5 -r ./data/cortex_surf_roi.md
```

### 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  $J_{ENS} \in \{-\infty, 1\}$ , como se muestra en la figura ??.

```

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 3: 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 4: 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.

### 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 5, 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 si una malla es válida en su construcción, etc.

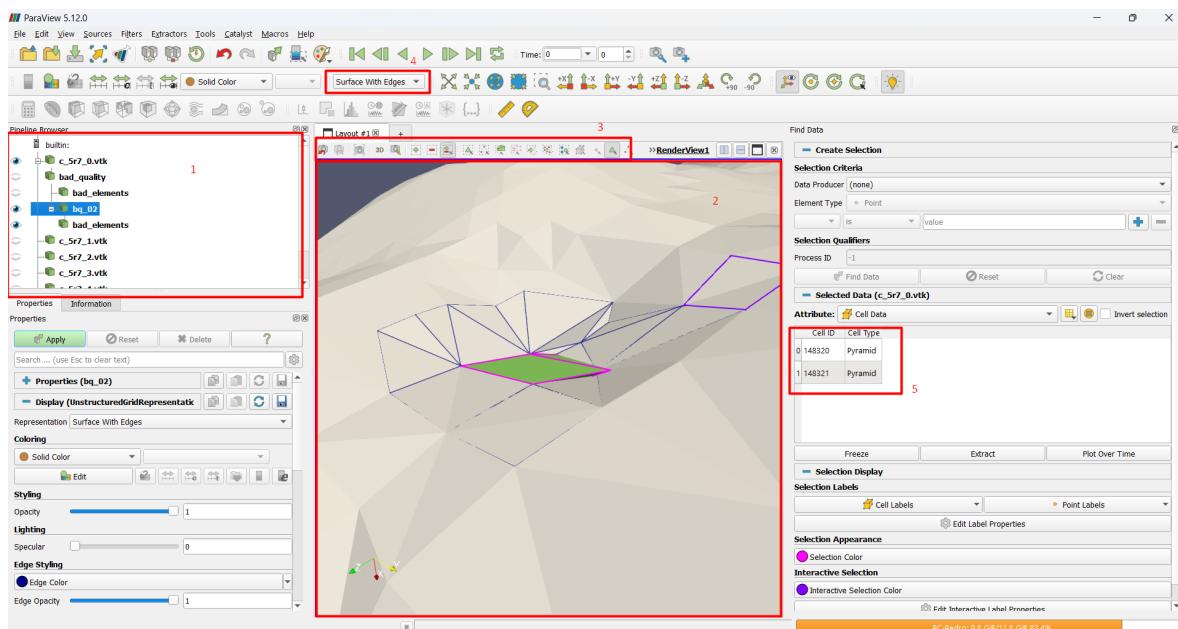


Figura 5: Interfaz de software Paraview, se muestra una sección de la malla inicial con dos elementos seleccionados.

Fuente: Elaboración propia.

## 3.4. Estructura de datos

En la construcción de la solución existen diversas estructuras de datos que se van complejiendo 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 contiene una unidad más pequeña que se le denominó Elementos, definidos en [González y Lobos, 2014] como se mencionaba al comienzo de esta sección, donde se definen patrones de transición a estos Elementos Figura 1. Luego, cada uno de estos Elementos está constituido por Puntos, que es la unidad más pequeña.

Para ejemplificar.

```
1 #include <vector>
2 class Octant
3 {
4     public:
5         Octant(vector<unsigned int> &eps, const unsigned short &
6             ref_level,
7                 const unsigned int &o_id);
8         ...
9         // indice del octante.
10        unsigned int id;
11        // indices de puntos que conforman al octante.
12        vector<unsigned int> points_indexes;
13        // indices de puntos que conforman los elementos
14        // pertenecientes al octante.
15        vector<vector<unsigned int>> sub_elements_points_indexes;
16 }
```

Listing 5: Clase Octant.

Fuente: Elaboración propia.

### 3.4.1. Acercamiento al algoritmo de Mixed-element mesh generator

Parte importante del software mencionado en la Sección 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  $\Omega$ .

**Output:** Volumetric mesh of  $\Omega$  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;
```

---

## 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.

Estos elementos inválidos podremos identificarlos por su  $J_{ENS}$  menor a cero. Entonces se ejecutó *JENS\_GENERATOR* a la malla inicial y se obtuvo la estadística de calidad de los elementos de la malla inicial, Listing 4.

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

Para ello se realizó una pequeña modificación a *MESHER\_GENERATOR* y al final de su algoritmo, cuando se exportan los datos de la malla, en específico, cuando guardan los puntos, elementos y octantes de la malla, se realiza una identificación de los índices de elementos que pertenecen a cada octante y se guardan en su clase como información del octante.

Para guardarlos en el octante se agrega un atributo a la clase llamado *sub\_elements\_indexes*, línea 16 en Listing 6.

Este atributo es consultado después, cuando se etiqueta cada octante que presente elementos inválidos.

```
1 #include <vector>
2 class Octant
3 {
4     public:
5         Octant(vector<unsigned int> &eps, const unsigned short &
6             ref_level,
7                 const unsigned int &o_id);
8         ...
9         // indice del octante.
10        unsigned int id;
11        // indices de puntos que conforman al octante.
12        vector<unsigned int> points_indexes;
13        // indices de puntos que conforman los elementos
14        // pertenecientes al octante.
15        vector<vector<unsigned int>> sub_elements_points_indexes;
16        // indices de elementos pertenecientes al octante.
17        vector<unsigned int> sub_elements_indexes;
18 }
```

Listing 6: Modificación Clase Octant.

Fuente: Elaboración propia.

Actualmente, hemos identificado 27 elementos que necesitan mejoras, 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.

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.

---

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

---

**Input :** Refinement level constraints  $RLC$ , triangular surface mesh  $\Omega$ , quality threshold  $T$ , maximum number of iterations  $I$ .

**Output:** if successful, volumetric mesh of  $\Omega$  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  $lo$ ;
```

---

Utilizando la lista de elementos involucrados, podemos visualizar los resultados en ParaView. Esto nos permite generar la Figura 6, 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 clara facilita la identificación de las áreas problemáticas dentro de la malla y demuestra cómo las transiciones entre regiones con diferentes características influyen en la calidad de los elementos.

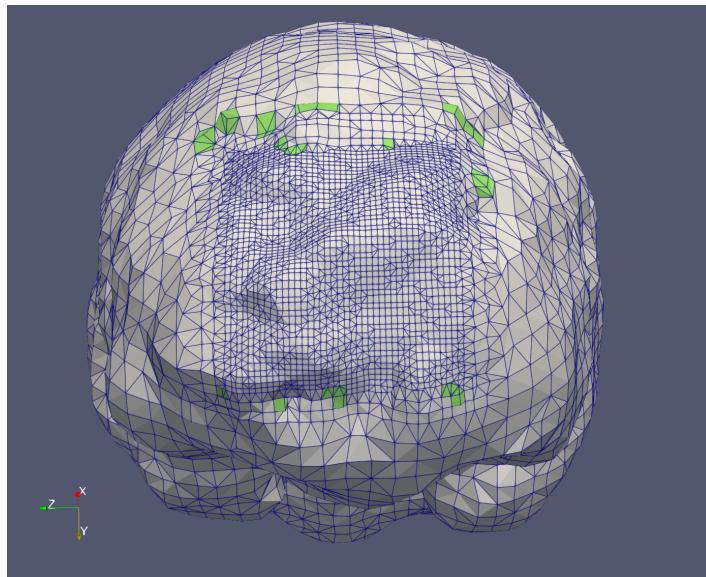


Figura 6: Zona frontal de la malla donde se presentan los elementos inválidos (verde).  
Fuente: Elaboración propia.

#### 4.2. Comprensión de elementos inválidos

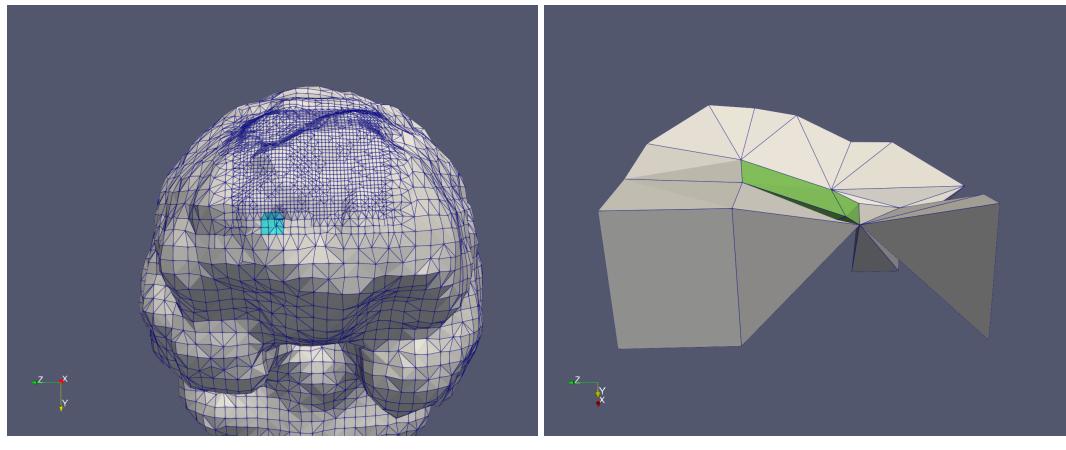
Para comprender mejor el problema, se realizará un análisis gráfico localizado, centrándose en un octante específico situado en la parte inferior izquierda del frontis de la malla. Este octante ha sido seleccionado debido a su característica cóncava en la superficie y porque contiene varios elementos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

En la Figura 7, se presenta la región de interés en el contexto de la malla inicial. La imagen a la izquierda muestra la malla en su totalidad, mientras que la imagen a la derecha proporciona una vista ampliada de la zona aislada del octante afectado. En esta vista detallada, se pueden observar todos los elementos contenidos en dicho octante, con los elementos inválidos resaltados en verde.

#### 4.3. 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 8, 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



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

Figura 7: 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.

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.

#### 4.4. 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 el *MESHER\_GENERATOR* para construir la malla, con  $RL_{gral} = 5$  y  $R_{surf} = 7$ .

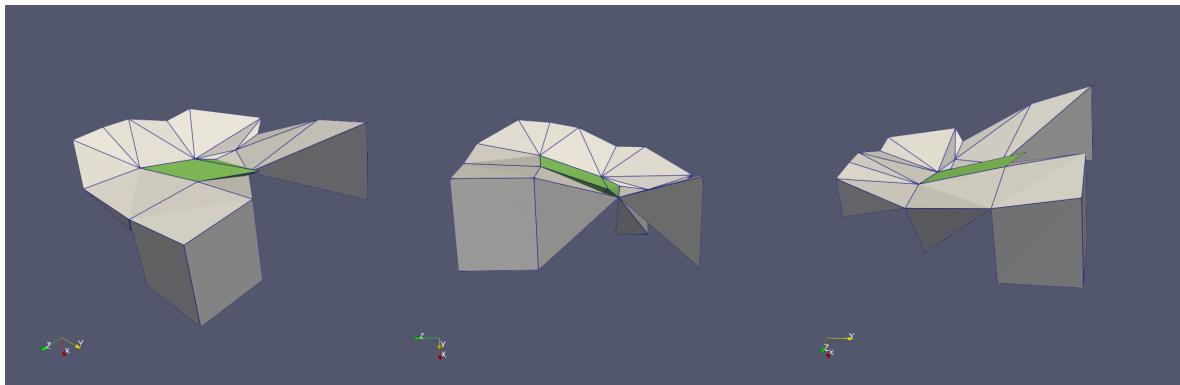


Figura 8: Perspectivas de vecindario (gris) donde se encuentra un para de elementos inválidos (verde).

Fuente: Elaboración propia.

## 2. Identificación y Etiquetado de Octantes.

El algoritmo 2 que incluye la funcionalidad *GET\_LABELED\_OCTANTS*, facilitará cumplir con esta etapa del algoritmo.

## 3. Refinamiento de Octantes Identificados.

El refinamiento de una lista de octantes puede realizarse con una de las funcionalidades de *MESHER\_GENERATOR*.

## 4. Ciclo Iterativo de Evaluación y Refinamiento.

Para realizar el proceso de iteración, se construirá un script en bash que ejecute *MESHER\_GENERATOR* con ciertas condiciones.

### 4.4.1. 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 7: 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<sub>ENS</sub>*. 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 en *MESHER\_GENERATOR*.

Puntos a destacar de esta implementación:

- Generación de Estadísticas  $J_{ENS}$ :

Al finalizar la generación de la malla, el algoritmo calcula las estadísticas  $J_{ENS}$ . Estas estadísticas proporcionan una medida de la calidad de los elementos de la malla y se registran en cada iteración, creando un historial detallado de calidad. Este proceso es crucial para monitorizar y evaluar la calidad de la malla generada de forma continua.

- Exportación de datos en formato estándar:

Para facilitar el análisis de las estadísticas  $J_{ENS}$ , los datos se almacenarán de manera persistente en un archivo con la extensión .histo. Este archivo estructurado contiene dos columnas esenciales para la evaluación de la calidad de la malla. En la primera columna, se registrarán los umbrales o cotas de calidad, mientras que la segunda columna mostrará la frecuencia de elementos cuya calidad está por debajo de cada umbral específico. La estructura de este archivo se ilustra en la Listing 8.

En el archivo **.histo**, las cotas se dispondrán en la primera fila comenzando por una cota que incluye los elementos con  $J_{ENS}$  negativo. A continuación, se utilizarán cotas progresivas como 0,03 y 0,05, seguidas de incrementos regulares de 0,05. Este formato de incremento permite una evaluación granular y sistemática de la calidad de los elementos a lo largo de un rango definido de valores  $J_{ENS}$ .

El histograma contenido en el archivo es generado por el módulo *JENS\_STATISTICS\_GENERATOR*. Este módulo es crucial porque facilita la visualización y el análisis eficiente de la distribución de la calidad de los elementos dentro de la malla. Al proporcionar una representación clara y estructurada de cómo se distribuye la calidad de los elementos, el histograma ayuda a identificar rápidamente cómo se ven afectados los elementos de la malla al realizar cada iteración.

```
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>
```

Listing 8: 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.

#### 4.4.2. 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 9: Lista de octantes a refinar en malla inicial.

Fuente: Elaboración propia.

En Listing 9, 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  $k \in \{0, \dots, NOL\}$ .

#### 4.4.3. 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 10, 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 >      ./meshing_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 10: 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.

#### 4.4.4. 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 consular 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.
          quality threshold T.
          maximum number of iterations I.
Output: if successful, volumetric mesh of  $\Omega$  that meet the RLC with quality greater than T.
Procedure GENERATE_MESH(main_surf, ref_surf, base_name, T, I):
foreach i  $\in \{0, \dots, I\}$  do
    out_name  $\leftarrow$  base_name + _ + i ;
    if i == 0 then
        | mesher_generator (main_surf, ref_surf, threshold, out_name) ;
    end if
    else
        | prev_iteration  $\leftarrow$  i - 1 ;
        | base_name_prev  $\leftarrow$  base_name + _ + prev_iteration ;
        | ref_name_prev  $\leftarrow$  base_name_prev + .ref ;
        | octs_name_prev  $\leftarrow$  base_name_prev + .oct ;
        | mesher_generator (main_surf, octs_name_prev, ref_name_prev, threshold, out_name) ;
    end if
    IMPROVER_EVALUATOR (ref_name_prev, i) ;
end foreach
lo  $\leftarrow$  GET_FILE_LINES (ref_name_prev) ;
PRINT "Finished, number of iterations exceeded." ;
PRINT "Octants to improve: lo" ;
return mesh;
Procedure IMPROVER_EVALUATOR (ref_name, i):
if FILE_EXIST (ref_name) then
    | lo  $\leftarrow$  GET_FILE_LINES (ref_name) ;
    | PRINT "Octants to improve in iteration i: lo" ;
end if
else
    | PRINT "No more Octants to improve in iteration i." ;
    | BREAK;
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 complejizar el refinamiento. 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 a el 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.

A cada una de estas mallas se les aplicará el algoritmo propuesto con un threshold cero y cien iteraciones como cota superior. 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 la sección anterior.

#### 5.3.1. Malla de corteza cerebral

Al implementar el algoritmo propuesto en la malla de corteza cerebral, entregándole como input el threshold definido en esta propuesta, el modelo de la corteza, el modelo de la superficie prismática a refinar, la base de nombre de los archivos exportados y una cota de cien iteraciones. Obtenemos el siguiente output 11, que nos muestra un identificador de cada iteración con la cantidad de Octantes a refinar al finalizar dicha iteración, finalizando exitosamente en la cuarta iteración con una malla sin elementos inválidos y reduciendo la cantidad de Octantes a refinar casi de manera exponencial.

Luego se ejecutó el algoritmo con la zona a refinar en el lóbulo occipital con el siguiente output 12.

Si bien, en esta ocasión se logró obtener una malla válida con algunas iteraciones adicionales, la cantidad de Elementos inválidos se comporta similar a el caso anterior.

```
1 input > ./script.sh 0.0 ./data/cortex.mdl ./data/cortex_surf_roi.mdl  
      c_5r7 100  
2 output > Threshold: 0.0  
3           Main surface: ./data/cortex.mdl  
4           Reference surface: ./data/cortex_surf_roi.mdl  
5           Base name: c_5r7  
6           Number of iterations: 100  
7           Octants to improve in iteration 0: 21  
8           Octants to improve in iteration 1: 7  
9           Octants to improve in iteration 2: 2  
10          Octants to improve in iteration 3: 1  
11          No more Octants to improve in iteration 4.
```

Listing 11: Output de ejecución algoritmo propuesto en malla de corteza cerebral con zona a refinar en lóbulo frontal.

Fuente: Elaboración propia.

```
1 input > ./script.sh 0.0 ./data/cortex.mdl ./data/cortex_surf_roi_2.  
      mdl c_5r7_2 100  
2 output > Threshold: 0.0  
3           Main surface: ./data/cortex.mdl  
4           Reference surface: ./data/cortex_surf_roi_2.mdl  
5           Base name: c_5r7_2  
6           Number of iterations: 100  
7           Octants to improve in iteration 0: 52  
8           Octants to improve in iteration 1: 27  
9           Octants to improve in iteration 2: 11  
10          Octants to improve in iteration 3: 6  
11          Octants to improve in iteration 4: 7  
12          Octants to improve in iteration 5: 3  
13          No more Octants to improve in iteration 6.
```

Listing 12: Output de ejecución algoritmo propuesto en malla de corteza cerebral con zona a refinar en lóbulo occipital.

Fuente: Elaboración propia.

### 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 en Listing 13 y Listing 16, respectivamente. En ambos casos se logró una malla válida en aproximadamente 5 iteraciones, pero a diferencia de los casos anteriores, como la corteza cerebral y la zona inferior en el Moai, la malla de Moai en la zona superior, se comporta aumentando en un grado menor la cantidad de Octantes por refinar entre la primera y segunda iteración, luego disminuye la cantidad de Octantes manteniendo el aparente modelo exponencial.

```
1 input > ./script.sh 0.0 ./data/moai.mdl ./data/moai_surf_roi.mdl
      moai_5r7 100
2 output >   Threshold: 0.0
3           Main surface: ./data/moai.mdl
4           Reference surface: ./data/moai_surf_roi.mdl
5           Base name: moai_5r7
6           Octants to improve in iteration 0: 22
7           Octants to improve in iteration 1: 5
8           Octants to improve in iteration 2: 2
9           Octants to improve in iteration 3: 1
10          Octants to improve in iteration 4: 1
11          No more Octants to improve in iteration 5.
```

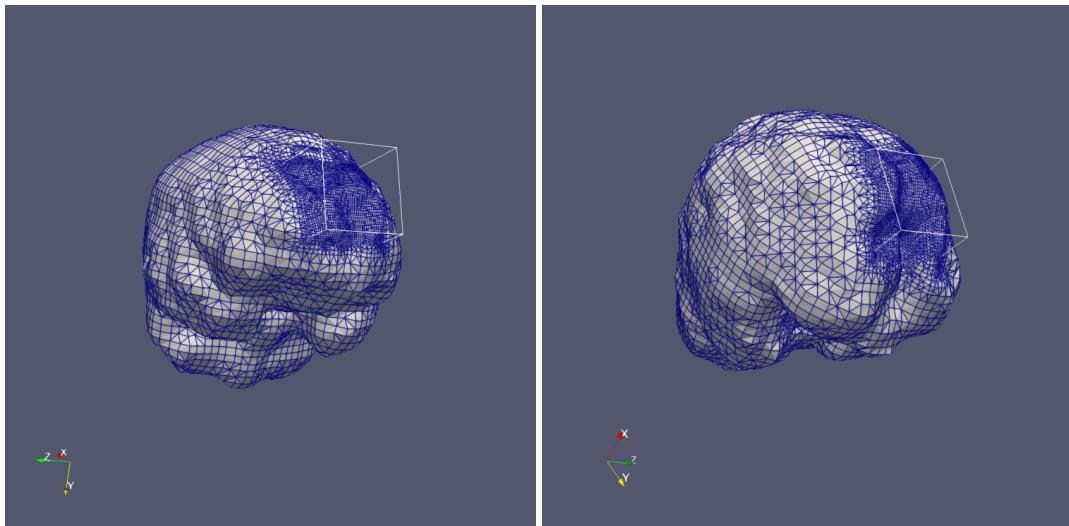
Listing 13: Output de ejecución algoritmo propuesto en malla de Moai con zona a refinar en zona inferior.

Fuente: Elaboración propia.

```
1 input > ./script.sh 0.0 ./data/moai.mdl ./data/moai_surf_roi_2.mdl
      moai_5r7_2 100
2 output >   Threshold: 0.0
3           Main surface: ./data/moai.mdl
4           Reference surface: ./data/moai_surf_roi_2.mdl
5           Base name: moai_5r7_2
6           Number of iterations: 100
7           Octants to improve in iteration 0: 10
8           Octants to improve in iteration 1: 12
9           Octants to improve in iteration 2: 9
10          Octants to improve in iteration 3: 3
11          No more Octants to improve in iteration 4.
```

Listing 14: Output de ejecución algoritmo propuesto en malla de Moai con zona a refinar en zona superior.

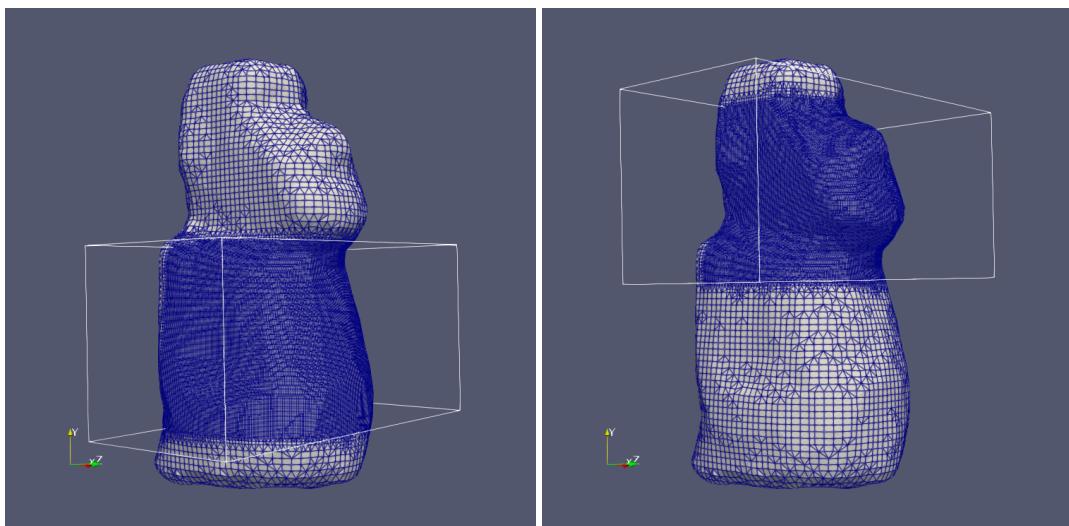
Fuente: Elaboración propia.



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

Figura 9: Diferentes localidades de refinamiento en corteza cerebral.

Fuente: Elaboración propia.



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

Figura 10: Diferentes localidades de refinamiento en Moai.

Fuente: Elaboración propia.

### 5.3.3. Malla de paladar

En el caso de la malla de paladar, podemos

```
1 input > ./script.sh 0.0 ./data/palate.mdl ./data/palate_surf_roi.mdl
      palate_5r7 100
2 output >   Threshold: 0.0
3           Main surface: ./data/palate.mdl
4           Reference surface: ./data/palate_surf_roi.mdl
5           Base name: palate_5r7
6           Number of iterations: 100
7           Octants to improve in iteration 0: 33
8           Octants to improve in iteration 1: 17
9           Octants to improve in iteration 2: 6
10          Octants to improve in iteration 3: 1
11          Octants to improve in iteration 4: 1
12          No more Octants to improve in iteration 5.
```

Listing 15: Output de ejecución algoritmo propuesto en malla de Moai con zona a refinar en zona superior.

Fuente: Elaboración propia.

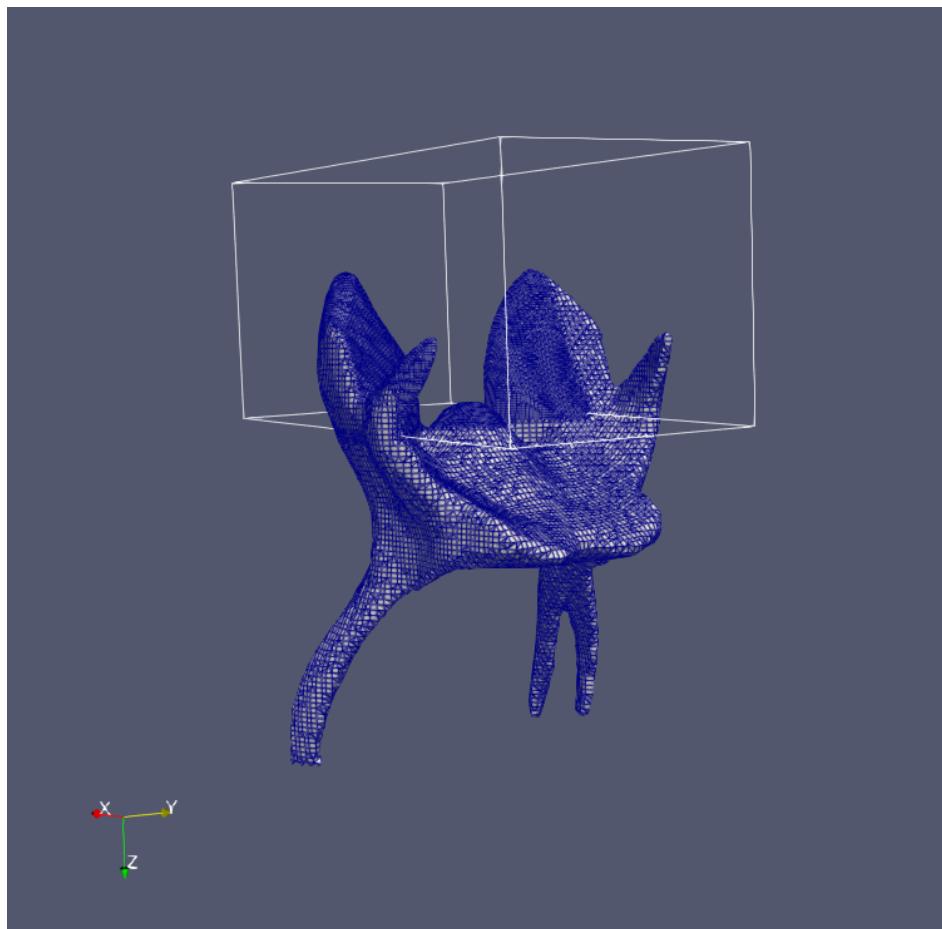


Figura 11: Representación Paladar con refinamiento en zona superior.

Fuente: Elaboración propia.

```
1 input >      ./script.sh 0.0 ./data/palate.mdl ./data/palate_surf_roi.mdl
      palate_5r7 100
2 output >    Threshold: 0.0
3           Main surface: ./data/palate.mdl
4           Reference surface: ./data/palate_surf_roi.mdl
5           Base name: palate_5r7
6           Number of iterations: 100
7           Octants to improve in iteration 0: 33
8           Octants to improve in iteration 1: 17
9           Octants to improve in iteration 2: 6
10          Octants to improve in iteration 3: 1
11          Octants to improve in iteration 4: 1
12          No more Octants to improve in iteration 5.
```

Listing 16: Output de ejecución algoritmo propuesto en malla de Moai con zona a refinar en zona superior.

Fuente: Elaboración propia.

#### 5.3.4. Malla de coxis

### 5.4. Resultados

Para analizar el comportamiento del algoritmo, nos enfocaremos en ajustar la cantidad de iteraciones 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 malla que representa el coxis, cuando logra converger, la cantidad de Octantes por refinar dando como resultado una malla válida, es posible reducir considerablemente la cantidad de Octantes por refinar en muy pocas iteraciones, a simple vista se podría entregar una hipótesis sobre el comportamiento del algoritmo en cada iteración.

Para esto, se considerará, a diferencia de lo que ya es tónica en lo antes mostrado, la cantidad de Elementos menores al threshold escogido, es decir, todos los Elementos que pertenecen a la malla y tienen  $J_{ENS} \leq 0$ , se denominará como  $E_0$ . 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 que se respaldan 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.

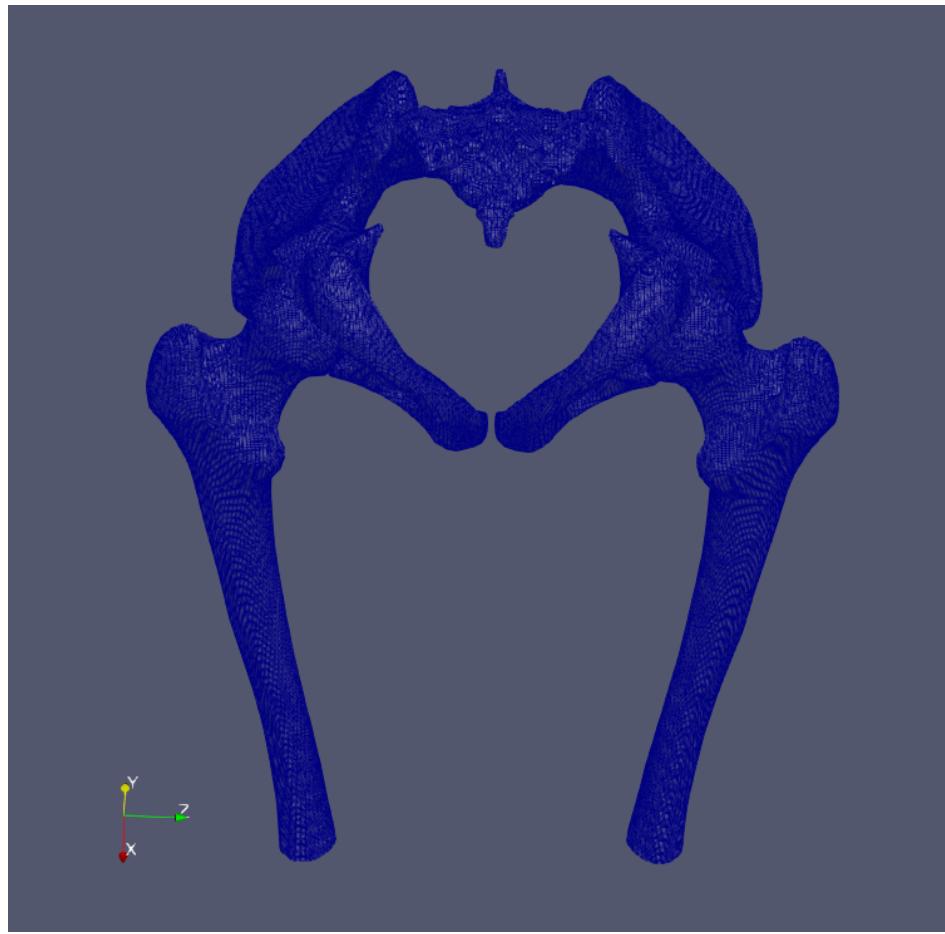


Figura 12: Representación Coxis sin refinamiento local.  
Fuente: Elaboración propia.

En la Tabla 1 se muestra el comportamiento de  $E_0$  en cada iteración, de aquí sólamente se analizarán los casos que convergen a cero.

## 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.

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.

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.

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

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

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 13. 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 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 comportamiento de los Elementos posicionados en la vecindad a las zonas refinadas.

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  $O$  con nivel de refinamiento  $rl_o$ , deben, sus vecinos directos,  $\bar{O}$ , tener un nivel de refinamiento  $rl_{\bar{o}} \in \{rl_o+1, rl_o-1, rl_o\}$ .

La estadística  $J_{ENS}$  se muestra como un histograma que representa la frecuencia de Elementos, siendo este histograma una función de escalonada que define la frecuencia de Elementos de calidad deficiente. En [Shepherd y Johnson, 2008], se define  $J_{ENS_{SJ}} \in [0, 0,2[$ , como Elementos cuestionables, es decir, Elementos de calidad deficiente. Además, destacar el estándar de ANSYS, que considera  $J_{ENS_{AN}} \in [0, 0,03[$ , como Elementos inválidos.

Por tanto, para este análisis se acotará definiendo  $EC \in \{0, 0,03, 0,05\}$  como Elementos de calidad cuestionable. Además, definiremos la frecuencia de Elementos con  $J_{ENS} \in [m, n[$ , como  $H_m^n$ .

Entonces, se graficará en Figura 14, los histogramas de cada malla para los intervalos definidos por  $EC$ . Teniendo,  $H_{-\inf}^{0,0}$ ,  $H_{0,0}^{0,03}$ ,  $H_{0,0}^{0,05}$ .

$J_{ENS}$  con la frecuencia de  $E_{EC}$ ,  $\forall x \in \{0, 0,03, 0,05\}$

## CAPÍTULO 6

### Conclusiones

Las Conclusiones son, según algunos especialistas, el aspecto principal de una memoria, ya que reflejan el aprendizaje final del autor del documento. En ellas se tiende a considerar los alcances y limitaciones de la propuesta de solución, establecer de forma simple y directa los resultados, discutir respecto a la validez de los objetivos formulados, identificar las principales contribuciones y aplicaciones del trabajo realizado, así como su impacto o aporte a la organización o a los actores involucrados. Otro aspecto que tiende a incluirse son recomendaciones para quienes se sientan motivados por el tema y deseen profundizarlo, o lineamientos de una futura ampliación del trabajo.

Todo esto debe sintetizarse en al menos 5 páginas.

## 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 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).
- [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.

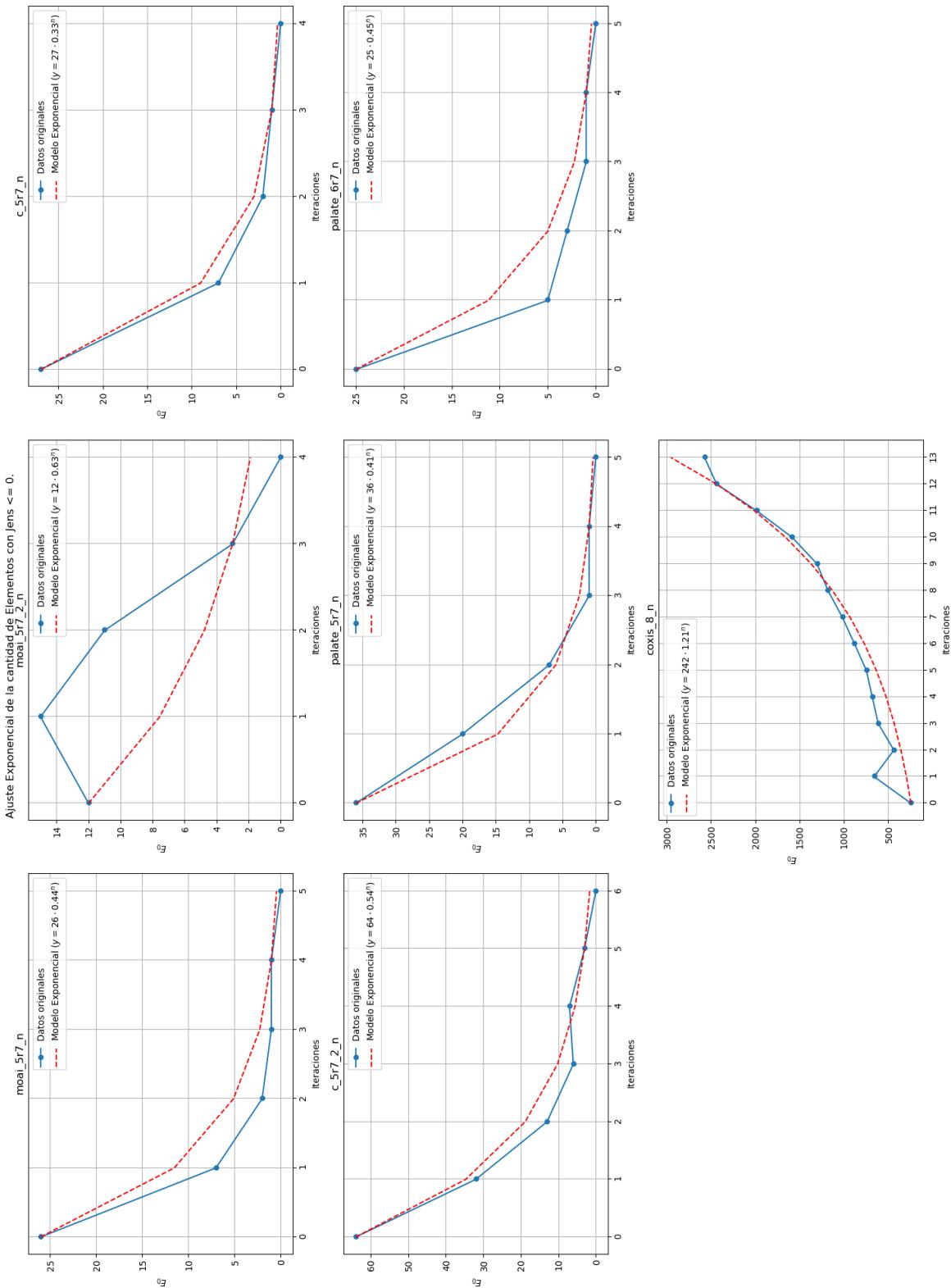


Figura 13: Ajuste exponencial de  $E_0$  para todos los casos.

Fuente: Elaboración propia.

