

Modelado geométrico de estructuras anatómicas para análisis por elementos finitos

Modelado geométrico de estructuras anatómicas para análisis por elementos finitos

Miller Gómez Mora



Resumen

Las estructuras anatómicas son objetos físicos que representan cualquier parte del cuerpo humano, las cuales ocupan un espacio y se pueden distinguir de su entorno. El método de elementos finitos (EF) ha sido utilizado recientemente para el estudio de algunas de estas estructuras, que generalmente hacen parte del sistema óseo.

En este libro se presentan las características principales de un *framework* que integra diferentes algoritmos y métodos para la generación de mallas de superficie de estructuras anatómicas complejas aptas para simulaciones por EF. Este *framework* recibe como entrada una imagen médica, a partir de la cual se reconstruye la geometría de las diferentes partes que conforman la estructura anatómica. Para completar el modelo, se realizan dos procesos adicionales: uno de suavizado y otro de simplificación.

Los modelos geométricos obtenidos contienen un alto porcentaje de triángulos de buena calidad, de tamaño uniforme y adaptados a la curvatura de la estructura anatómica, lo que facilita la investigación y aplicación práctica en simulaciones por EF. Así mismo, cuando la estructura anatómica está compuesta de varias regiones, el modelo geométrico garantiza que las interfaces de las regiones no se traslapen entre sí, lo que ayuda, por un lado, a evitar que haya triángulos invertidos y, por otro, a garantizar la convergencia de las simulaciones por EF.

Palabras clave: modelado geométrico, estructura anatómica, simulación por elementos finitos.

¿Cómo citar este libro?

Gómez Mora, M. (2024). *Modelado geométrico de estructuras anatómicas para el análisis por elementos finitos*. Universidad Distrital Francisco José de Caldas-Editorial UD. <https://doi.org/10.69740/UD.9789587877205.9789587877229>

Abstract

Anatomical structures are physical objects that represent any part of the human body, which occupy space and can be distinguished from their environment. The finite element (FE) method has recently been used to study some of these structures, which are generally part of the bone system.

This book presents the main characteristics of a framework that integrates different algorithms and methods for surface mesh generation of complex anatomical structures suitable for FE simulations. This framework receives as input a medical image, from which the geometry of the different parts that make up the anatomical structure is reconstructed. To complete the model, two additional processes are carried out: one for smoothing and another for simplification.

The geometric models obtained contain a high percentage of good quality triangles, of uniform size and adapted to the curvature of the anatomical structure, which facilitates research and practical application in FE simulations. Likewise, when the anatomical structure is composed of several regions, the geometric model guarantees that the interfaces of the regions do not overlap each other, which helps, on the one hand, to avoid inverted triangles and, on the other, to guarantee the convergence of FE simulations.

Keywords: geometric modeling, anatomical structure, finite element simulation.

How to cite this book?

Gómez Mora, M. (2024). *Geometric modeling of anatomical structures for finite element analysis*. Universidad Distrital Francisco José de Caldas-Editorial UD. <https://doi.org/10.69740/UD.9789587877205.9789587877229>



© Universidad Distrital Francisco José de Caldas
© Oficina de Investigaciones
© Miller Gómez Mora

ISBN: 978-958-787-720-5
ISBN digital: 978-958-787-721-2
ISBN ePub: 978-958-787-722-9

DOI: <https://doi.org/10.69740/UD.9789587877205.9789587877229>

Primera edición, noviembre de 2024

Líder Unidad de Publicaciones
Rubén Eliécer Carvajalino C.

Gestión editorial
Catalina Moreno Correa

Corrección de estilo
HIPERTEXTO SAS

Diagramación
HIPERTEXTO SAS

Diseño de cubierta
Johana Delgado Macías

Editorial UD

Universidad Distrital Francisco José de Caldas
Carrera 24 n.º 34-37 Bogotá, D. C., Colombia
Teléfono: 6013239300 ext. 6202
Correo electrónico: publicaciones@udistrital.edu.co

Cláusula de responsabilidad:

Editorial UD no es responsable de las opiniones y de la información recogidas en el presente texto. Los autores asumen de manera exclusiva y plena toda responsabilidad sobre su contenido, ya sea este propio o de terceros, declarando en este último supuesto que cuentan con la debida autorización de terceros para su publicación. Igualmente, expresan que no existe conflicto de interés alguno en relación con los resultados de la investigación propiedad de tales terceros. En consecuencia, los autores serán responsables civil, administrativa o penalmente, frente a cualquier reclamo o demanda por parte de terceros, relativa a los derechos de autor u otros derechos que se vulneren como resultado de su contribución.

Todos los derechos reservados.

Esta obra no puede ser reproducida sin el permiso previo escrito de la Unidad de Publicaciones de la Universidad Distrital Francisco José de Caldas.
Hecho en Colombia.

Sistema de Bibliotecas de la Universidad Distrital Francisco José de Caldas
Catalogación en la publicación (CEP)

Gómez Mora, Miller

Modelado geométrico de estructuras anatómicas para análisis por elementos finitos / Miller Gómez Mora. -- Primera edición. -- Bogotá : Universidad Distrital Francisco José de Caldas, 2024.
142 páginas : ilustraciones, gráficas ; 24 cm. (Espacios)

ISBN: 978-958-787-720-5 ISBN digital: 978-958-787-721-2
ISBN ePub: 978-958-787-722-9

1. Ingeniería biomédica 2. Método de elementos finitos -- Anatomía humana 3. Ecuaciones diferenciales parciales 4. Generación numérica de mallas (análisis numérico) 5. Métodos de simulación I. Serie.

610.28: CDD 23 edición.

Contenido

Introducción	19
Capítulo 1. Marco teórico	25
1.1. Generación de mallas de superficie	25
1.1.1. Segmentación de imágenes 3D	26
1.1.1.1. Técnicas estructurales	28
1.1.1.2. Técnicas estocásticas	28
1.1.1.3. Técnicas híbridas	29
1.1.2. Generación de la geometría	30
1.1.2.1. Algoritmo <i>Marching cubes</i>	30
1.1.2.2. Algoritmo <i>Dual contouring</i>	32
1.2. Suavizado	34
1.2.1. Relaciones de vecindario	35
1.2.2. Suavizado laplaciano	36
1.2.3. Laplaciano con corrección	39
1.2.4. Filtro pasa-bajos	40
1.3. Simplificación	41
1.3.1. Agrupación de vértices	42
1.3.2. Clasificación de vértices	43
1.3.3. Colapso de aristas	43
1.4. Requerimientos geométricos	44
1.4.1. Métricas de calidad de los triángulos	45
1.4.2. Métricas de calidad de los tetraedros	51
1.4.3. Efecto de un elemento malo	56
Capítulo 2. <i>Framework</i> para el modelado geométrico de estructuras anatómicas	59
2.1. Modelado matemático	60
2.2. Reconstrucción geométrica de las regiones	61
2.3. Reconstrucción geométrica de las estructuras anatómicas	65
2.4. Suavizado	69
2.5. Simplificación	74

Capítulo 3. Simulaciones por elementos finitos y resultados	77
3.1. Modelado de estructuras simples	78
3.1.1. Modelado geométrico	79
3.1.2. Diseño del modelo de EF	80
3.1.3. Resultados	86
3.2. Modelado de estructuras compuestas	87
3.2.1. Modelado geométrico	88
3.2.2. Diseño del modelo de EF	89
3.2.3. Resultados	93
3.3. Calidad de las mallas de superficie	96
Conclusiones	103
Referencias	105
Anexos	113
Anexo A. Introducción al método de elementos finitos	113
A.1. Discretización	114
A.2. Funciones de interpolación	116
A.3. Ensamblaje de las ecuaciones de los elementos	119
A.4. Condiciones de frontera	120
A.5. Solución de las ecuaciones	121
A.6. Resultados derivados	122
A.6.1. Desplazamientos	122
A.6.2. Deformación	122
A.6.3. Tensión	123
Anexo B. Introducción al análisis por elementos finitos con FEBio	125
B.1. Modelado geométrico	126
B.2. Preprocesamiento	127
B.2.1. Ensamblado	128
B.2.2. Cargas y condiciones de frontera	130
B.2.3. Propiedades de material	132
B.2.4. Mallado y tipo de elemento	133
B.2.5. Paso de análisis	135
B.3. Procesamiento o fase de solución	136
B.4. Posprocesamiento y visualización	138
Sobre el autor	141

Índice de figuras

Figura Intro1.	Malla de superficie de un hígado humano	21
Figura Intro2.	Acercamiento de la región demarcada por el recuadro rojo de la figura Intro1	21
Figura 1.1.	Imagen médica volumétrica	26
Figura 1.2.	Tipos de segmentación	27
Figura 1.3.	Casos base del algoritmo <i>Marching cubes</i>	30
Figura 1.4.	Algoritmo <i>Marching cubes</i>	31
Figura 1.5.	Ilustración de la dualidad en 2D, entre el algoritmo <i>Marching cubes</i> (izquierda) y el algoritmo <i>Dual contouring</i> (derecha)	33
Figura 1.6.	Imagen 2D y su correspondiente representación mediante un árbol <i>quadtrees</i>	33
Figura 1.7.	Vecindario topológico en una malla de triángulos	35
Figura 1.8.	Suavizado de un caballito de mar representado por un contorno cerrado	36
Figura 1.9.	Suavizado de una curva poligonal 2D	38
Figura 1.10.	Filtro laplaciano con corrección aplicado a una curva poligonal 2D	40
Figura 1.11.	Resultado de la aplicación del filtro de suavizado pasa-bajos	41
Figura 1.12.	Agrupación de vértices	42
Figura 1.13.	Clasificación de vértices para simplificación y retriangulación	43
Figura 1.14.	Colapso de aristas	44
Figura 1.15.	Numeración de vértices y aristas en un elemento triangular	46
Figura 1.16.	Relación entre el <i>inradius</i> y el <i>circumradius</i> de un triángulo	48
Figura 1.17.	Vértices de un tetraedro	51
Figura 1.18.	Ilustración de la altura h_3 del vértice 3	53
Figura 2.1.	<i>Framework</i> para el modelado geométrico de estructuras anatómicas	59

Figura 2.2.	Orientación de vértices y triángulos de una malla de superficie	61
Figura 2.3.	Sistema de coordenadas en el algoritmo FE	63
Figura 2.4.	Resultado de aplicar el método AEAP	65
Figura 2.5.	Grafo MRML	67
Figura 2.6.	Generación errónea de un modelo geométrico M	68
Figura 2.7.	Generación de la malla de superficie de una vértebra humana	71
Figura 2.8.	Suavizado de la malla de superficie M_i	72
Figura 2.9.	Modelo geométrico compuesto por tres mallas de superficie M_i	73
Figura 2.10.	Resultado de simplificar una malla de superficie M_i sin suavizar	75
Figura 2.11.	Resultado de simplificar una malla de superficie M_i suavizada	75
Figura 3.1.	Metodología para el diseño de simulaciones por EF	78
Figura 3.2.	Estructura base de la columna vertebral	78
Figura 3.3.	Metodología para el diseño de simulaciones por EF	79
Figura 3.4.	Modelado geométrico de un DIV	80
Figura 3.5.	Mallado volumétrico del DIV	81
Figura 3.6.	Asignación de las condiciones de frontera del DIV	83
Figura 3.7.	Asignación del material elástico lineal al DIV	84
Figura 3.8.	Configuración del paso de análisis del DIV	85
Figura 3.9.	Información del avance en la ejecución de la simulación	86
Figura 3.10.	Resultados de la simulación del DIV	87
Figura 3.11.	Metodología para el diseño de simulaciones por EF	88
Figura 3.12.	Modelado geométrico de la UEF L1-L2	89
Figura 3.13.	Asignación del material en el <i>software</i> de diseño FEBio	91
Figura 3.14.	Configuración de las interfaces de contacto	92
Figura 3.15.	Configuración del paso de análisis	94
Figura 3.16.	Información del avance en la ejecución de la simulación	95
Figura 3.17.	Distribución de la tensión efectiva de Lagrange de la UEF	95
Figura 3.18.	Distribución de la tensión efectiva de Lagrange del DIV	96
Figura 3.19.	Comparación de la calidad de las mallas de superficie de la vértebra L1	98

Figura 3.20.	Porcentaje de conservación de la superficie y el volumen	101
Figura A.1.	Modelo físico y matemático de una varilla	114
Figura A.2.	Discretización en elementos	114
Figura A.3.	Funciones de interpolación lineal para una barra de dos nodos	117
Figura A.4.	Error de discretización	118
Figura A.5.	Conectividad entre 3 elementos de 2 nodos	119
Figura A.6.	Notación para componentes de desplazamiento	122
Figura B.1.	Los componentes principales de la interfaz gráfica de FEBio Studio	126
Figura B.2.	Geometría del modelo que se va a simular	128
Figura B.3.	Configuración del contacto entre un cuerpo rígido y un cuerpo deformable	130
Figura B.4.	Configuración de las cargas y condiciones de frontera	131
Figura B.5.	Asignación de materiales	133
Figura B.6.	Configuración de proceso de mallado	134
Figura B.7.	Configuración del paso de análisis	136
Figura B.8.	Ejecución de la simulación	137
Figura B.9.	Terminación exitosa del proceso de simulación	138
Figura B.10.	Visualización de los resultados	139

Índice de tablas

Tabla 2.1. Tipos de consultas de proximidad	73
Tabla 3.1. Resultados del proceso de mallado de la UEF	90
Tabla 3.2. Estadísticas de las simulaciones del DIV	99
Tabla 3.3. Estadísticas del proceso de simulación de la UEF	99
Tabla 3.4. Distancia de Hausdorff	101
Tabla 3.5. Estadísticas de las mallas de superficie	101

Introducción

Las estructuras anatómicas son objetos físicos que representan cualquier parte del cuerpo humano, que normalmente ocupan un espacio y se pueden distinguir de su entorno. Los órganos, huesos, tejidos y células son todas estructuras anatómicas. Muchas de estas están compuestas o divididas en partes o regiones con ciertas características y funciones asociadas (Wu *et al.*, 2008).

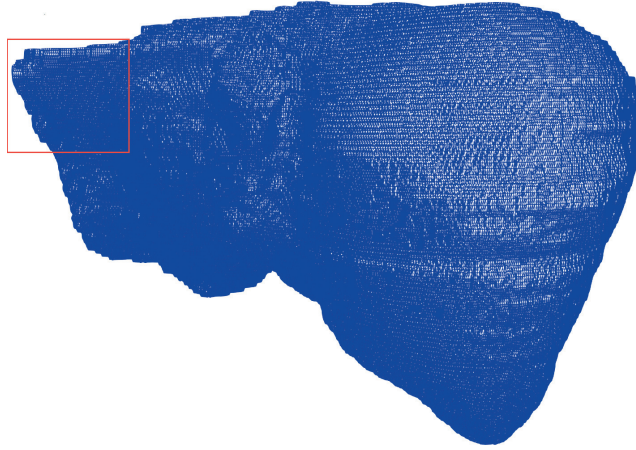
Los métodos para el estudio de las estructuras anatómicas han avanzado rápidamente, desde la disección de animales y cadáveres frescos o preservados, hasta la creación de modelos computacionales tridimensionales (3D). Especialmente, el modelado computacional está contribuyendo a la resolución de problemas complejos que normalmente no tienen soluciones analíticas o que no ofrecen posibilidad de estudio experimental. Estos problemas pueden describirse en términos de ecuaciones diferenciales parciales, y el método de elementos finitos (EF) es un enfoque numérico con el que estas ecuaciones pueden resolverse en términos aproximados (De *et al.*, 2010), (Maas *et al.*, 2012). Probablemente los ejemplos más comunes se encuentran en la biomedicina, donde los modelos computacionales que utilizan el método de EF han analizado con éxito sistemas biomecánicos y han realizado importantes contribuciones para comprender el comportamiento mecánico de diferentes estructuras anatómicas (Witte *et al.*, 2017). Esto gracias a que el análisis por EF es capaz de capturar los parámetros biomecánicos internos de las diferentes partes que conforman la estructura anatómica, como huesos y tejidos blandos, lo que es difícil de medir mediante enfoques experimentales (Little y Adam, 2015).

Un paso importante en el diseño de un modelo para análisis por EF es la construcción de la geometría y la discretización del dominio de interés, lo cual se realiza generalmente con ayuda de un programa de diseño asistido por computador CAD (Sun *et al.*, 2005), (Panagiotopoulou, 2009). La geometría de las diferentes partes de la estructura anatómica es representada por una malla de triángulos. La unión de estas mallas forma el dominio de la simulación, donde cada malla representa la superficie de un subdominio (Jalalian *et al.*, 2013), (Huynh *et al.*, 2015), (Gao *et al.*, 2015). Sin embargo, las estructuras anatómicas tienen gran complejidad geométrica lo que hace muy difícil modelar fielmente la geometría de estos objetos. La estrategia más usada es generar la geometría directamente a partir de imágenes médicas como resonancia magnética (RM) o tomografía computarizada (TC) (Jalalian *et al.*, 2013), (Botsch *et al.*, 2010), (Gavidia *et al.*, 2011). Pero estas mallas tienen dos problemas relacionados con las inexactitudes que surgen y son inherentes al proceso de adquisición de las imágenes médicas. El primer problema está relacionado con la aparición de artefactos de *aliasing*, también conocidos como efecto escalón o sierra, debido a que los algoritmos de extracción de mallas siguen los contornos de la imagen médica y el resultado final no refleja la suavidad que generalmente tienen las estructuras anatómicas. El segundo problema está relacionado con los miles o cientos de miles de triángulos que se necesitan para capturar los detalles de alta complejidad geométrica que generalmente tienen las estructuras anatómicas. Esto no es práctico en términos computacionales ya que el tiempo y la memoria requeridos para el análisis y despliegue de los modelos 3D son proporcionales al número de triángulos.

La figura Intro1 muestra la malla de superficie, extraída mediante el algoritmo *Marching Cubes* (Lorensen y Cline, 1987), de un hígado humano con 221 402 vértices y 442 788 triángulos, lo cual impide incluso una buena visualización.

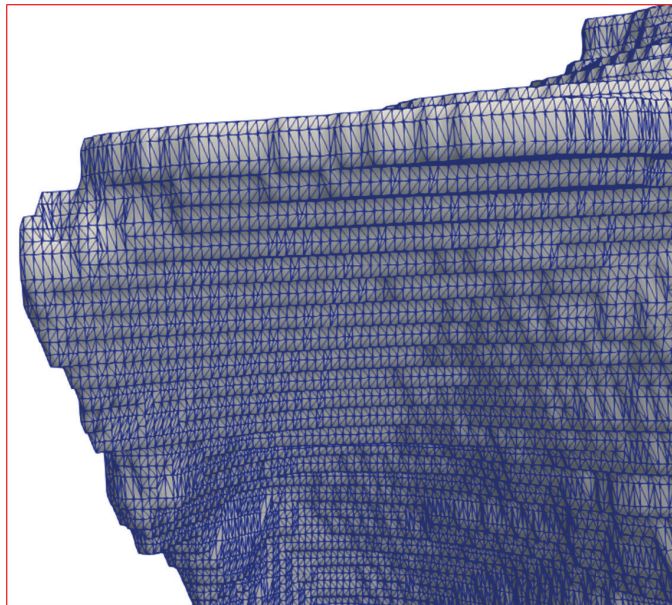
En la figura Intro2 se puede observar un acercamiento de la región demarcada en la parte superior izquierda por el recuadro rojo, donde se evidencian de mejor manera los dos problemas en mención.

Figura Intro1. Malla de superficie de un hígado humano



Fuente: elaboración propia.

Figura Intro2. Acercamiento de la región demarcada por el recuadro rojo de la figura Intro1



Fuente: elaboración propia.

Desde un punto de vista práctico, la capacidad de generar mallas de superficie de buena calidad es la clave del éxito del análisis numérico (Du *et al.*, 2009). Una buena malla generalmente logra un equilibrio entre la precisión de

los modelos geométricos y el tiempo de cálculo requerido para procesarlos. Adicionalmente, las mallas de superficie se utilizan como entrada para los generadores de mallas de volumen, lo que influye considerablemente en la calidad de estas últimas. Cuando se tienen mallas de volumen con elementos de baja calidad (extremadamente delgados o muy distorsionados) a menudo dificultan la convergencia de la simulación por EF y aumentan los errores de análisis. Esto obliga a que en muchas ocasiones sea necesario aplicar procesos de suavizado y diezmado a las mallas de superficie con el fin de obtener modelos con una geometría simplificada y manejable, pero que además que cuente con elementos de buena calidad (Garland y Heckbert, 1997), (Luebke, 2001), (Levine *et al.*, 2012).

En caso de que la estructura anatómica esté compuesta de varias partes, tanto el suavizado como la simplificación de la geometría requieren una atención especial, ya que la modificación de la malla perteneciente a una parte de la estructura puede afectar a las mallas vecinas. Por ejemplo, si una operación de simplificación mediante el colapso de aristas se produce cerca del límite de una malla, los bordes de los límites se moverán y, en consecuencia, las mallas vecinas también deberán moverse para evitar generar un agujero o distorsión entre las mallas (d'Otreppe *et al.*, 2012). Adicionalmente, se debe garantizar que las mallas sean cerradas, sin agujeros o grietas, y completamente *2-manifold*, es decir, cada borde de la malla debe ser compartido exactamente por dos caras y la vecindad de cada vértice debe ser el equivalente topológico de un disco. Una malla cerrada *2-manifold* S debe satisfacer la condición de Euler-Poincaré, $n + t - k = 2$.

Los problemas mencionados hacen que la generación de modelos geométricos de estructuras anatómicas adecuados para el análisis por EF sea una tarea difícil. Sin embargo, al no cumplir con estos requerimientos el modelo resultante de EF puede ser insoluble o propenso a imprecisiones numéricas (Feng *et al.*, 2010), (Maas *et al.*, 2016). Estos problemas pueden resolverse en gran parte combinando diferentes algoritmos dentro de un *framework* (marco de trabajo) capaz de generar modelos geométricos aptos para simulaciones por EF, manteniendo los detalles geométricos relevantes, así como la naturaleza multiregión de las estructuras anatómicas.

En el capítulo 1 se analizan los métodos y algoritmos más utilizados para la obtención y optimización de las mallas de superficie. Así mismo, se analizan los requerimientos geométricos necesarios para simulaciones por EF. En el capítulo 2 se integran, dentro de un *framework*, diferentes métodos y algoritmos computacionales para el modelado geométrico de estructuras anatómicas, permitiendo replicar de manera realista las características geométricas relevantes de la estructura bajo estudio. En el capítulo 3 se presentan los resultados de aplicar el *framework* propuesto en diferentes simulaciones por EF de estructuras anatómicas, tanto simples como compuestas. En el capítulo 4 se plasman las conclusiones más significativas de la aplicación del *framework* propuesto. Finalmente, en el anexo A se presenta una introducción al método de EF y en el anexo B se hace una introducción al análisis por EF mediante el diseño de un modelo de EF de una barra de torsión utilizando el *software* FEBio Studio (<http://febio.org>) (Maas *et al.*, 2012).

Capítulo 1. Marco teórico

Con el reciente desarrollo de las técnicas de modelado y adquisición de datos 3D, las mallas de triángulos se han convertido en la representación de superficies más popular debido a su simplicidad, eficiencia y flexibilidad (Botsch *et al.*, 2010), lo que ha beneficiado enormemente diferentes áreas científicas y de ingeniería como el diseño asistido por computador, la simulación física, la computación biomédica, la ingeniería inversa y la visualización científica, solo por nombrar algunas. En este capítulo se hace un estudio de los algoritmos y métodos que se pueden aplicar al modelado geométrico 3D de estructuras anatómicas, comenzando con los métodos para la generación de mallas de superficie a partir de imágenes médicas (numeral 1.1). La generación de las mallas es solo el comienzo, la calidad de las mismas a menudo determina la precisión del análisis por elementos finitos. Sin embargo, las mallas resultantes pueden tener defectos de *aliasing*, lo que hace necesario aplicar un método de suavizado mediante la reubicación iterativa de los vértices (numeral 1.2). Adicionalmente, las mallas pueden estar compuestas de miles o cientos de miles de triángulos, muchos de los cuales tienen una mala calidad. En el numeral 1.3 se analizan los algoritmos para reducir el número de triángulos que conforman dichas mallas. Finalmente, en el 1.4 se describen los requerimientos geométricos que deben cumplir los elementos de las mallas necesarios en los procesos de análisis por EF.

1.1. Generación de mallas de superficie

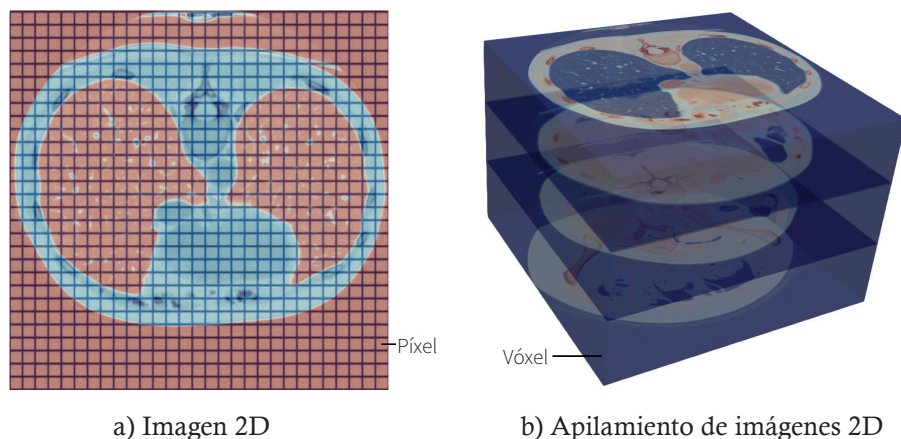
El modelado geométrico comienza generalmente con la reconstrucción de la superficie de las diferentes partes que componen la estructura anatómica que

se quiere modelar. La selección de la estructura anatómica se realiza a partir de imágenes médicas mediante un proceso llamado segmentación, el cual descompone los datos de la imagen médica en regiones o partes significativas (numeral 1.1.1). Con base en el resultado del proceso de segmentación se genera la geometría de las diferentes partes de la estructura que se representa mediante mallas de superficie (numeral 1.1.2).

1.1.1. Segmentación de imágenes 3D

Las imágenes médicas como la RM o TC combinan imágenes 2D en una cuadrícula 3D (figura 1.1). Cada imagen 2D representa una porción delgada de la parte de la anatomía escaneada y está compuesta de píxeles individuales, donde la distancia entre dos píxeles suele ser constante en cada dirección (figura 1.1a). Los píxeles dentro de cada imagen 2D están representados por valores escalares que se pueden interpretar como valores de intensidad (figura 1.1b). Cada valor de intensidad dentro de esa matriz de píxeles es una medida promediada de las propiedades del material en esa ubicación particular de la estructura escaneada. En la TC, por ejemplo, se mide la tasa de absorción de los rayos X, normalizada a la tasa de absorción del agua, expresada en las llamadas unidades Hounsfield. Las regiones con valores de intensidad homogéneos suelen representar estructuras anatómicas, mientras que los gradientes fuertes son indicadores de los límites de los tejidos.

Figura 1.1. Imagen médica volumétrica

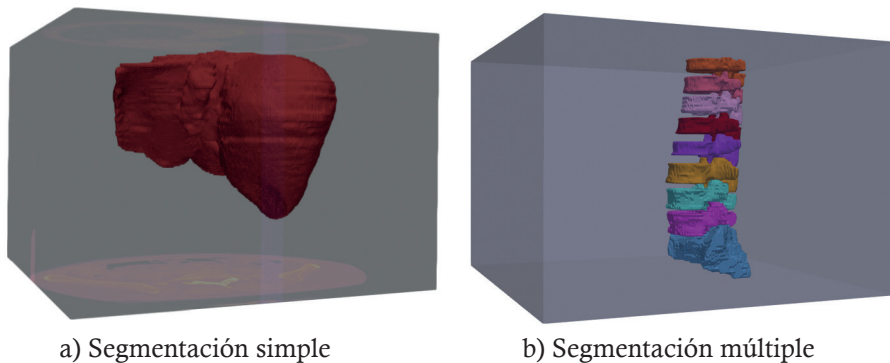


Fuente: elaboración propia.

Por otro lado, los elementos de la cuadrícula 3D se denominan vóxeles (elementos de volumen) y se obtienen mediante el apilamiento de las imágenes 2D (figura 1.1b) formando un campo escalar 3D. Aunque los campos escalares 3D ya se pueden visualizar directamente utilizando las denominadas técnicas de renderizado de volumen, estos datos no se pueden modificar de forma sencilla y no proporcionan una entrada razonable para las simulaciones de EF. Así, el mayor desafío es recuperar las formas inherentes de las estructuras anatómicas a partir de datos de imágenes médicas 3D con la mayor precisión posible y convertirlos en geometría. Dichas estructuras se pueden describir por sus límites, y estos límites se dan implícitamente mediante la clasificación de los vóxeles en un proceso llamado segmentación, donde a cada vóxel se le asigna una etiqueta única para indicar a qué parte o región particular pertenece (figura 1.2). En aplicaciones médicas, generalmente es suficiente segmentar solo la estructura anatómica relevante y no toda la anatomía que aparece en la imagen.

Las segmentaciones tienen el mismo tamaño que la imagen médica original y pueden ser simples o múltiples. El resultado de las segmentaciones simples es una máscara binaria (figura 1.2a), mientras que el resultado de las segmentaciones múltiples es un conjunto de etiquetas que corresponden a diferentes partes o regiones de una estructura (figura 1.2b). Existen diferentes algoritmos y técnicas de segmentación que pueden aplicarse. Lakare y Kaufman (2000) clasificaron las técnicas de segmentación en tres clases: estructurales, estocásticas e híbridas.

Figura 1.2. Tipos de segmentación



Fuente: elaboración propia.

1.1.1.1. Técnicas estructurales

Las técnicas estructurales intentan encontrar propiedades estructurales de la región que se va a segmentar. Las propiedades estructurales como las superficies de intersección (aristas en 2D) se detectan en el volumen y luego se combinan para segmentar la región. En algunos algoritmos, la información de la estructura se guarda y luego se recupera para realizar la segmentación en un conjunto de datos similar. Como ejemplo se tienen las técnicas de detección de bordes 3D, las técnicas morfológicas, los algoritmos de búsqueda de grafos, los modelos deformables y los conjuntos de nivel.

Las técnicas estructurales tienen la ventaja de que pueden segmentar un solo órgano o región de interés, sin embargo, la precisión de la segmentación depende de la calidad de la imagen 3D. Dado que estos algoritmos ubican información estructural, cualquier ruido en los datos de entrada podría afectar el resultado. Estas técnicas también necesitan un buen contraste entre las regiones que se van a segmentar. Un bajo contraste puede causar dificultades para encontrar bordes, lo que podría afectar igualmente el resultado. Otra desventaja es que son difíciles de automatizar lo que obliga a la intervención manual, principalmente para seleccionar puntos de semilla dentro de la región de interés.

1.1.1.2. Técnicas estocásticas

Por otro lado, las técnicas estocásticas no tienen en cuenta la información estructural. Su objetivo es realizar la segmentación con base solo en un análisis matemático (estadístico) de los datos que contiene la imagen 3D. Las técnicas más representativas de esta clasificación son: métodos basados en umbrales, técnicas de clasificación, algoritmos de agrupamiento y campos aleatorios de Markov. El umbral es probablemente la técnica más simple de segmentación para volúmenes con valores escalares. En esta técnica, se utiliza un único valor llamado umbral para crear una partición binaria de intensidades de los vóxeles. Todos los vóxeles con intensidades superiores al umbral se agrupan en una clase y aquellos con intensidades por debajo del umbral se agrupan en otra clase. El uso de un solo umbral da como resultado una segmentación binaria del volumen.

Las técnicas estocásticas realizan la segmentación en toda la imagen 3D dando como resultado diferentes regiones. Aunque este proceso de segmentación es totalmente automático (excluyendo la selección de parámetros iniciales), la selección de la región de interés entre las regiones segmentadas puede ser tediosa y necesitar de intervención manual. Por ejemplo, si se quiere segmentar la sección torácica de un ser humano, se obtendrán todos los huesos clasificados como una región. Si solo se está interesado en uno de ellos, por ejemplo, una costilla, se tendrá que localizar manualmente y eliminar todos los demás huesos. Esto generalmente se hace pasando el conjunto de datos a través de otro algoritmo como crecimiento de regiones para marcar la región seleccionada y eliminar todo lo demás.

La precisión y la calidad de la segmentación en estos algoritmos dependen de la selección de los parámetros iniciales. Esto podría ser bueno y malo a la vez. La buena parte es que, con la selección adecuada de parámetros, el algoritmo podría hacerse robusto contra el ruido para un problema particular. También podría ajustarse para realizar la segmentación en conjuntos de datos de bajo contraste. Por otro lado, la selección de parámetros es muy crítica y cualquier elección incorrecta podría conducir a una segmentación no deseada.

1.1.1.3. Técnicas híbridas

Finalmente, existen algoritmos de segmentación que no se pueden clasificar completamente en las dos categorías anteriores, pero sí usan algo de estos algoritmos. Entre ellos se tiene: crecimiento de regiones, división y fusión de regiones, enfoques guiados por atlas, y redes neuronales artificiales. Crecimiento de regiones es la técnica más simple para extraer una región conectada de una imagen 3D con base en un criterio de conexión predefinido. Este criterio puede ser tan simple como la intensidad del vóxel o podría ser la salida de cualquier otro algoritmo de segmentación. En la forma más simple, el crecimiento de regiones requiere un punto de partida, llamado semilla, a partir del cual comienza a crecer hasta que se satisfacen los criterios de conexión.

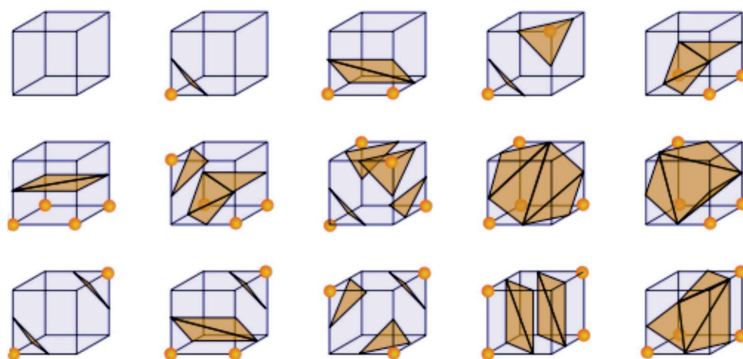
1.1.2. Generación de la geometría

Los resultados de la segmentación se almacenan comúnmente en un volumen etiquetado 3D (figura 1.2). Aunque puede ser factible utilizar directamente los resultados del proceso de segmentación de imágenes 3D en ciertas aplicaciones, para el caso de visualización 3D y de análisis por EF es necesaria una representación geométrica por medio de mallas de superficie que constan de decenas o cientos de miles de triángulos conectados, generados mediante algoritmos como *Marching cubes* (MC) (Lorensen y Cline, 1987) o *Dual contouring* (DC) (Ju *et al.*, 2002). Con el proceso de reconstrucción de la geometría se pasa del espacio 3D de los vóxeles a superficies 2D incrustadas en R3 lo que permite reducir drásticamente el tamaño de los datos (Gómez-Mora y Rodríguez, 2018).

1.1.2.1. Algoritmo *Marching cubes*

El algoritmo MC subdivide el espacio 3D en una serie de pequeños cubos conocidos como vóxeles. Básicamente, este algoritmo examina cada vóxel y genera una triangulación en caso de que la isosuperficie se cruce con el vóxel. Una isosuperficie es otra forma de llamar a una superficie definida por la ecuación implícita $f(x, y, z) = d$, donde f es una función del espacio y d una constante cuyo valor (isovalor) es a menudo 0 (Hansen y Johnson, 2011). El prefijo *iso* indica que la función f toma el mismo valor (d) en toda la superficie.

Figura 1.3. Casos base del algoritmo *Marching cubes*



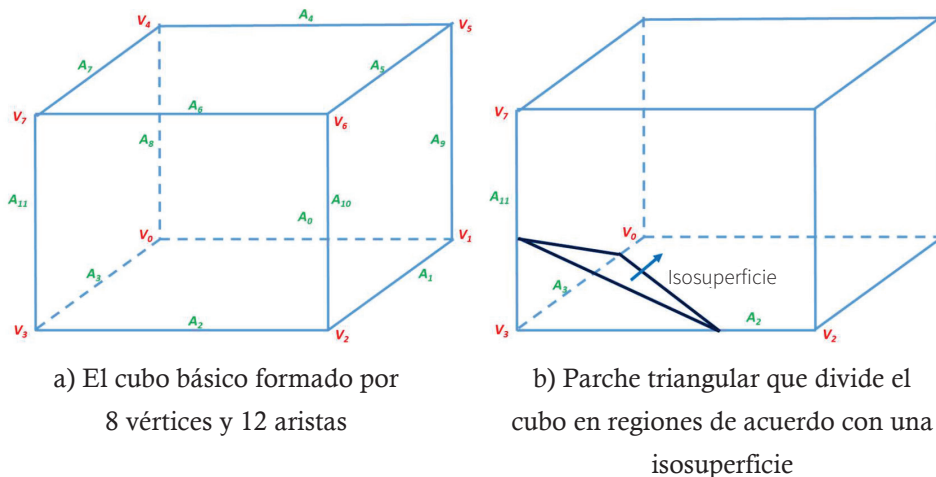
Fuente: elaboración propia.

Cada vértice de un vóxel (cubo) es un punto de la malla 3D, y todos estos puntos están definidos por el mallado del volumen. Dado que cada uno de los ocho vértices de un cubo puede ser marcado, existen 256 (2^8) combinaciones posibles del estado de una esquina. La figura 1.3 muestra que estas combinaciones se pueden simplificar a 15, teniendo en cuenta que las celdas se duplican en las siguientes condiciones:

- Al reflejar la forma en cualquiera de los tres ejes principales.
- Al invertir el estado de todas las esquinas y voltear las normales de los polígonos relacionados.
- Al rotar en cualquier grado sobre cualquiera de los tres ejes principales.

En cada iteración del algoritmo MC se avanza a través de los vóxeles, examinando los puntos de las esquinas y reemplazando el vóxel con un conjunto apropiado de polígonos (figura 1.4). Cada vértice y cada arista del cubo (figura 1.4a) se indexan en tablas a las que accede el algoritmo MC para buscar y determinar qué aristas del vóxel se cruzan con la isosuperficie, para así poder crear parches triangulares que dividen el cubo en diferentes partes (figura 1.4b). Al conectar los parches de todos los vóxeles en el límite de la isosuperficie, se obtiene una representación de superficie que se aproxima a la que describe el conjunto de datos volumétricos.

Figura 1.4. Algoritmo *Marching cubes*



Fuente: elaboración propia.

La figura 1.4a muestra el orden de indexación de las aristas A_i y los vértices V_i , mientras que la figura 1.4b corresponde a un caso en el que el vértice V_3 está por debajo del valor de isosuperficie y esta última corta las aristas A_2 , A_3 y A_{11} . La posición exacta de los vértices del parche triangular, en color azul oscuro, depende de la relación del valor de la isosuperficie y de los valores en los vértices 3-2, 3-0 y 3-7, respectivamente.

Cuando un vértice está por encima de la isosuperficie y otro vértice adyacente está por debajo, la posición en la que la isosuperficie corta la arista es interpolada linealmente (figura 1.4b). La relación de la longitud entre los dos vértices será la misma que la relación del valor de la isosuperficie con los valores en los vértices de la celda de la grilla. Los puntos de intersección se pueden calcular por interpolación lineal. Si V_1 y V_2 son los vértices de una arista de corte y val_1 y val_2 son los valores escalares de cada vértice, el punto de intersección P viene dado por:

$$P = V_1 + \frac{(isovalor - val_1)(V_2 - V_1)}{val_2 - val_1} \quad (1.1)$$

Antes de terminar, el algoritmo MC calcula la normal de las caras de la triangulación para obtener una representación de la superficie lo más correcta posible. La flecha azul, en la figura 1.4b, indica el vector normal a la superficie del triángulo generado.

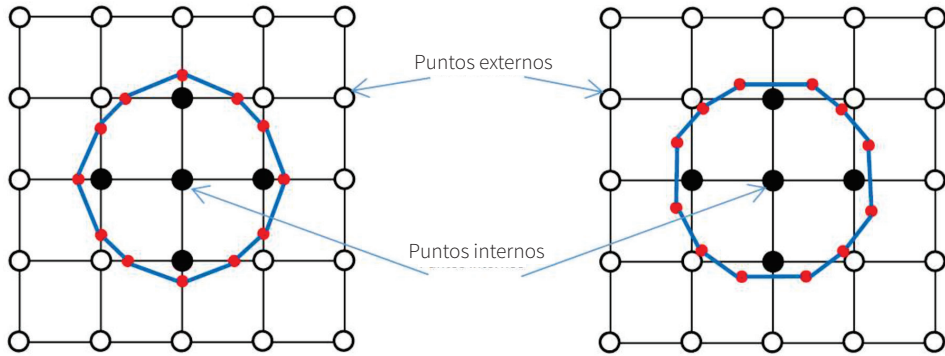
1.1.2.2. Algoritmo *Dual contouring*

El algoritmo *Dual contouring* (DC) (Ju *et al.*, 2002) se utiliza para extraer la superficie de un volumen implícito. El método es dual en el sentido de que los vértices generados por DC son topológicamente duales a las caras generadas por el algoritmo MC (Lorensen y Cline, 1987), como se muestra en la figura 1.5.

El algoritmo DC es un híbrido de métodos basados en cuadrículas y en árboles (*octrees* para 3D y *quadtrees* para 2D). Un volumen etiquetado de entrada se subdivide primero utilizando una cuadrícula uniforme de un tamaño apropiado. La figura 1.6a muestra una superficie 2D con múltiples etiquetas

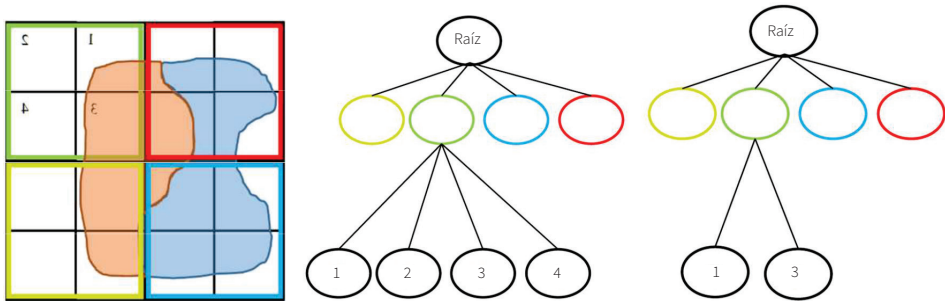
que es dividida usando una cuadrícula uniforme de cuadrados. Luego se usa un *octree/quadtree* para representar y analizar los cubos de la cuadrícula.

Figura 1.5. Ilustración de la dualidad en 2D, entre el algoritmo *Marching cubes* (izquierda) y el algoritmo *Dual contouring* (derecha)



Fuente: elaboración propia.

Figura 1.6. Imagen 2D y su correspondiente representación mediante un árbol *quadtree*



a) Imagen multi-etiqueta dividida en cuadrados

b) Representación mediante un árbol *quadtree*

c) Árbol *quadtree* después de la simplificación

Fuente: elaboración propia.

Una vez que el volumen se divide utilizando la cuadrícula uniforme, se anotan y almacenan los valores de las esquinas de cada cubo. En el DC estándar de un solo material, las esquinas están etiquetadas como dentro o fuera del volumen. Esto da como resultado una malla de un solo material. En el DC

multimaterial, a cada esquina del cubo se le asigna un valor entero, donde cada número entero representa un material particular del volumen etiquetado.

Las hojas del árbol representan los cubos de la cuadrícula en su nivel más fino de división, como se muestra en la figura 1.6b. Un cubo o una hoja cuyas 8 esquinas tienen el mismo valor de material implica que el cubo está completamente dentro de ese material en particular. Por lo tanto, el árbol se puede simplificar aún más eliminando dichas hojas. Esto se demuestra en la figura 1.6c, donde las hojas numeradas 2 y 4 se retiran del *quadtrees* porque sus 4 esquinas tienen el mismo valor, es decir, los cubos se encuentran fuera del volumen.

Además de los enfoques de *Marching cubes* y *Dual contouring* se han desarrollado otros algoritmos implícitos para el mallado de superficies basados en la triangulación de Delaunay restringida como el *software* desarrollado por Rineau y Yvinec (2007), que está disponible en la Biblioteca de Algoritmos de Geometría Computacional (CGAL, por sus iniciales en inglés) (Fabri y Pion, 2009). Yu *et al.* (2012) propusieron un algoritmo de mejoramiento de límites para el seguimiento de superficie basado en un gradiente fraccional derivado. Para esto, utilizaron una integral diferencial para extraer las partes importantes de la superficie límite. Luego, el fraccional diferencial mejora la característica del borde y la textura para rastrear con éxito la estructura de la textura y extraer la superficie del límite en la imagen. La obtención del límite de la superficie generada en Yu *et al.* (2012) es mejor que el algoritmo MC.

1.2. Suavizado

Un aspecto importante de los modelos geométricos es la fidelidad geométrica y visual que las mallas de superficie deben mantener para parecerse lo más posible a sus formas originales. Por ejemplo, la apariencia natural de las estructuras anatómicas es generalmente suave sin bordes afilados. Sin embargo, la malla generada por los algoritmos de extracción de mallas tiene solo una aproximación continua C^0 , debido a que estos algoritmos siguen los contornos de la imagen 3D segmentada que está compuesta de vóxeles. Esto limita la representación de las superficies curvas que generalmente tienen las estructuras anatómicas y, por lo tanto, solo representan una aproximación de la estructura anatómica real de la persona.

Para muchas aplicaciones que involucran mallas de superficie, a veces es necesario realizar algún tipo de suavizado para mejorar las características deseables o eliminar los artefactos y el ruido no deseados. En esta parte se examinan las relaciones de vecindario y tres métodos de suavizado, cada uno de los cuales ha demostrado ser fácil de implementar, rápido y efectivo (Preim y Botha, 2013).

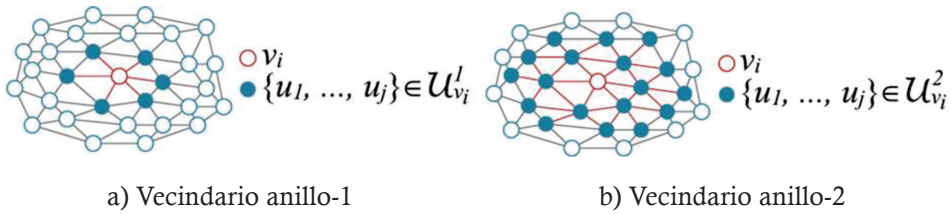
1.2.1. Relaciones de vecindario

La mayoría de los algoritmos que procesan mallas de superficie, tanto para suavizar como para simplificar la misma, operan en una vecindad local de cada vértice. Existen principalmente dos tipos diferentes de vecindario: vecindario topológico y vecindario euclidiano.

Sea V el conjunto de vértices de una malla de superficie. La vecindad topológica del vértice $v_i \in V$ comprende todos los vértices $u_j \in V$ que están conectados a v_i a través de una ruta de cierta longitud (figura 1.7).

La vecindad topológica 1 comprende exactamente todos los u_i que comparten una arista con v_i , por lo que la ruta de conexión tiene la longitud de 1 (figura 1.7a). De manera similar, el vecindario topológico 2 contiene todos los vértices conectados a v_i con un camino que consiste en uno o dos aristas (figura 1.7b). Estos vecindarios a menudo se conocen como anillo-1, anillo-2 y así sucesivamente.

Figura 1.7. Vecindario topológico en una malla de triángulos



Fuente: elaboración propia.

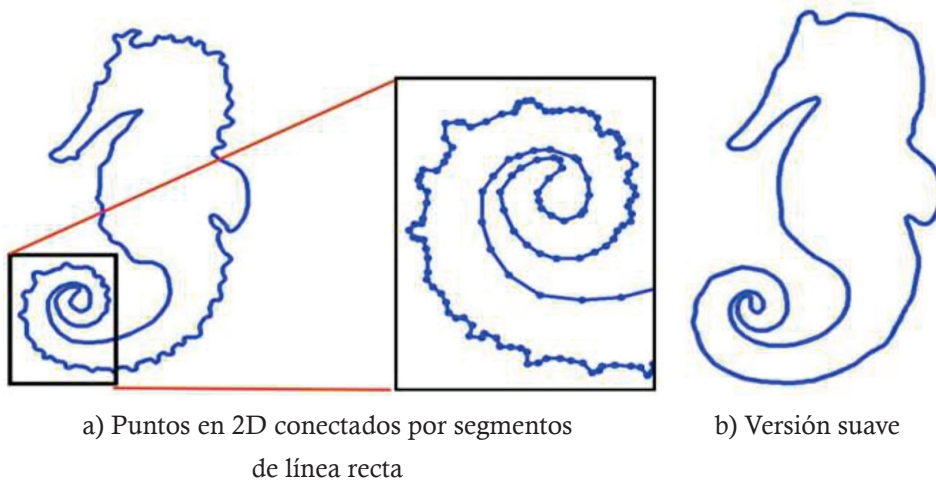
De otro lado, el vecindario euclidiano del vértice v_i comprende todos los vértices $u_j \in V$ que tienen una distancia (euclidiana) $\leq d$ a v_i , donde d es

un umbral de distancia. Mientras que la vecindad topológica puede contener vértices que son bastante distantes, la vecindad euclidiana tiene una forma circular (en 2D) o esférica (en 3D), limitando efectivamente la distancia. En la práctica, el vecindario anillo-1 tiene un par de ventajas para muchas aplicaciones. Los vértices vecinos u_j pueden determinarse más rápido (sin cálculos de distancia) y siempre contienen un cierto número de elementos que generalmente se necesitan, por ejemplo, para un suavizado de malla robusto. Sin embargo, las distancias euclidianas también se emplean para algunos problemas de representación de la superficie en medicina, por ejemplo, en la visualización de vasos con superficies implícitas.

1.2.2. Suavizado laplaciano

La figura 1.8a muestra la forma de un caballito de mar representada por un contorno cerrado mediante una secuencia de puntos 2D que están conectados por segmentos de línea recta, como se ilustra en el acercamiento. Ahora se quieren eliminar las características rugosas de la forma del caballito de mar y obtener una versión suavizada, como la que se muestra en la figura 1.8b.

Figura 1.8. Suavizado de un caballito de mar representado por un contorno cerrado



Fuente: elaboración propia.

Un procedimiento simple para lograr esto es conectar repetidamente los puntos medios de segmentos sucesivos de contorno, lo que se conoce como suavizado de punto medio. Como se puede ver en la figura 1.9a, después de dos pasos de suavizado de punto medio, cada vértice de contorno $\mathbf{v}_i$ se mueve al punto medio del segmento de línea que conecta los puntos medios de los segmentos de contorno originales adyacentes a $\mathbf{v}_i$. Específicamente, sean $\mathbf{v}_{i-1} = (x_{i-1}, y_{i-1})$, $\mathbf{v}_i = (x_i, y_i)$ y $\mathbf{v}_{i+1} = (x_{i+1}, y_{i+1})$ tres vértices de contorno consecutivos. Luego, el nuevo vértice $\mathbf{v}_i$ después de dos pasos de suavizado del punto medio viene dado por un promedio local,

$$\mathbf{v}'_i = \frac{1}{2 \left[\frac{1}{2(\mathbf{v}_{i-1} + \mathbf{v}_i)} \right]} + \frac{1}{2 \left[\frac{1}{2(\mathbf{v}_i + \mathbf{v}_{i+1})} \right]} = \frac{1}{4} \mathbf{v}_{i-1} + \frac{1}{2} \mathbf{v}_i + \frac{1}{4} \mathbf{v}_{i+1} \quad (1.2)$$

El vector entre $\mathbf{v}_i$ y el punto medio del segmento de línea que conecta los dos vértices adyacentes a $\mathbf{v}_i$, que se muestra como una flecha roja en la figura 1.9b, se llama el laplaciano discreto en $\mathbf{v}_i$, denotado por $\delta(\mathbf{v}_i)$, donde:

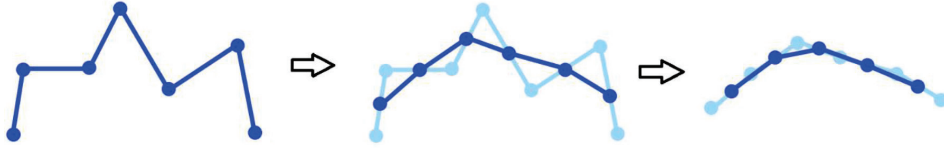
$$\delta(\mathbf{v}_i) = \frac{1}{2(\mathbf{v}_{i-1} + \mathbf{v}_{i+1})} - \mathbf{v}_i \quad (1.3)$$

Como se puede observar en la figura 1.9, después de dos pasos de suavizado del punto medio, cada vértice del contorno se desplaza por la mitad de su laplaciano discreto. Esto se conoce como suavizado laplaciano (Field, 1988). La versión suavizada del caballito de mar en la figura 1.8b se obtuvo aplicando diez pasos de suavizado laplaciano.

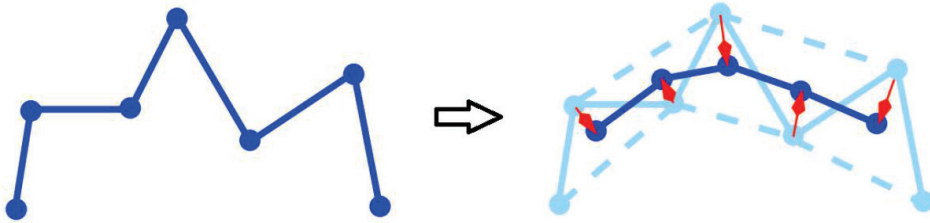
El filtro laplaciano se puede aplicar iterativamente a una malla poligonal y produce superficies lisas después de un número apropiado de iteraciones. En la mayoría de las implementaciones, se aplica al vecindario anillo-1, ya que es más rápido en comparación con los vecindarios más extendidos. Debido a esto, también se conoce como operador paraguas. Este término se refiere al hecho de que un paraguas tiene un vértice central, conectado a una

serie de vértices en los alrededores. En cada iteración, un vértice se mueve en la dirección del centro geométrico de sus vecinos. Un factor de ponderación λ determina la influencia de la posición del vértice original en su valor después del suavizado.

Figura 1.9. Suavizado de una curva poligonal 2D



a) Dos pasos de suavizado de punto medio



b) Un paso de suavizado laplaciano

Fuente: elaboración propia.

La transformación de cada vértice se realiza de acuerdo con la siguiente ecuación:

$$\mathbf{v}'_i = \mathbf{v}_i + \frac{\lambda}{m} \sum_{j=1}^m (\mathbf{u}_j - \mathbf{v}_i), \mathbf{u}_j \in V, \mathbf{v}_i \neq \mathbf{u}_j, \forall \mathbf{u}_j \in U_{\mathbf{v}_i} m = \|U_{\mathbf{v}_i}\| \quad (1.4)$$

donde $\mathbf{v}_i$ es la posición anterior de un vértice, y $\mathbf{v}'_i$ es su nueva posición y $\mathbf{u}_j$ son vértices de la vecindad desde $\mathbf{v}_i$. Por lo tanto, el operador laplaciano es una versión discretizada del laplaciano para funciones analíticas continuas, aplicable a las mallas poligonales. El número de iteraciones y el factor de ponderación λ son los dos parámetros que determinan la cantidad de suavizado.

Dado que todos los vecinos son tratados por igual en el proceso de suavizado, el método conduce a deformaciones de la superficie cuando las

regiones con bordes significativamente más pequeños están cerca de las regiones con bordes grandes. En los modelos de superficie, extraídos directamente de cuadrículas de volumen regular, este problema no es grave. Sin embargo, si las mallas se simplifican y comprimen, el suavizado posterior debería ponderar adecuadamente la contribución del vértice de acuerdo con la distancia desde el vértice actual.

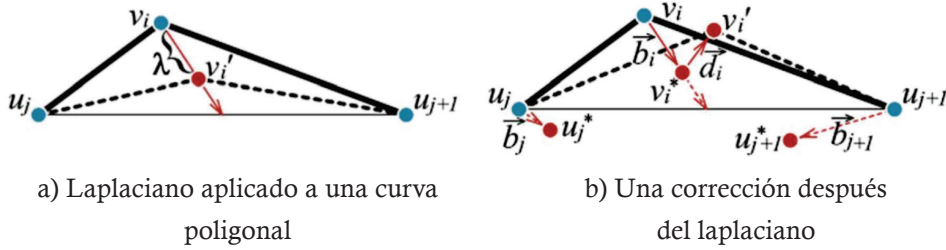
Aunque el filtro u operador laplaciano es el método de suavizado más simple y efectivo que se puede encontrar en diferentes herramientas de modelado geométrico, este no garantiza una mejora de la calidad de la malla, donde se pueden generar elementos invertidos. Adicionalmente, se pueden perder pequeños detalles, ya que el método no es adaptativo, similar al filtro gaussiano en el suavizado de imágenes. Por lo tanto, hubo necesidad de introducir diferentes versiones de suavizado laplaciano o de combinarlo con otros métodos.

1.2.3. Laplaciano con corrección

El laplaciano con corrección adicional fue introducido por Vollmer *et al.* (1999) para reducir la contracción del volumen. En este enfoque, los vértices se trasladan aproximadamente en la dirección original en una cierta cantidad después del suavizado laplaciano (figura 1.10).

La figura 1.10a ilustra el principio de funcionamiento del laplaciano aplicado a una curva poligonal 2D que mueve el vértice $\mathbf{v}_i$ en la dirección de sus vecinos de acuerdo con el factor de ponderación λ . La figura 1.10b muestra la corrección aplicada después del laplaciano que reduce el efecto del suavizado. La corrección se mueve en una dirección derivada de los vectores normales locales. La dirección específica y la cantidad se derivan de la posición anterior de ese vértice y la traslación promedio de sus vecinos directos.

Sin embargo, este algoritmo consume bastante tiempo, ya que cada paso requiere más tiempo de cómputo y se requieren más pasos en comparación con el laplaciano original, aunque se logran obtener resultados más precisos que este último. Adicionalmente, la configuración del laplaciano con corrección tiene cuatro parámetros (dos más que el laplaciano).

Figura 1.10. Filtro laplaciano con corrección aplicado a una curva poligonal 2D

Fuente: elaboración propia.

1.2.4. Filtro pasa-bajos

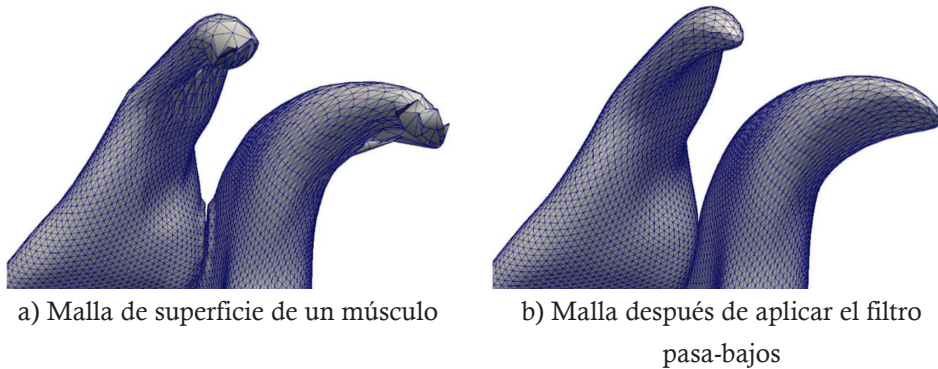
El filtro pasa-bajos, también denominado el método Taubin (Taubin, 1995b), cuenta con tres parámetros: dos filtros tipo Laplace uno con peso positivo λ y otro con peso negativo μ y un contador de iteraciones N . Las propiedades del filtro pasa-bajos son la frecuencia de paso de banda k_{PB} , la ondulación de paso de banda δ_{PB} , la frecuencia de banda de parada k_{SB} y el rizado de banda de parada δ_{SB} , los cuales están relacionados de la siguiente manera:

$$\begin{aligned}
 \lambda &< \frac{1}{k_{SB}} \\
 \frac{1}{\lambda} + \frac{1}{\mu} &= k_{PB} \\
 \left[\frac{(\lambda - \mu)^2}{-4\lambda\mu} \right]^N &< 1 + \delta_{PB} \\
 \left[(1 - \lambda k_{SB})(1 - \mu k_{SB}) \right]^N &< \delta_{SB}
 \end{aligned} \tag{1.5}$$

En términos generales, el filtro pasa-bajos combina un filtro laplaciano positivo λ , que suaviza el contorno reduciendo su tamaño y uno negativo μ , que lo vuelve a inflar (Taubin, 1995a). El resultado es similar al laplaciano con corrección y, al igual que este, conduce a resultados más precisos que el laplaciano original reduciendo fuertemente la contracción del volumen. Además, se ajusta muy bien al tipo de superficies biológicas, que son cerradas y con una distribución de vértices casi uniforme (figura 1.11). En la figura 1.11 se presenta el resultado de la operación de suavizado usando el filtro λ / μ , de un músculo. En la figura 1.11a se muestra la malla original obtenida de la imagen médica y en la figura 1.11b se observa la malla final suavizada.

Los tres filtros de suavizado descritos se pueden beneficiar de una estructura de datos que permite un recorrido rápido de vértices con acceso inmediato a sus vecinos de anillo-1. Adicionalmente, los algoritmos de suavizado no cambian el número de vértices ni la topología de la malla sino que solo se cambian las coordenadas de los vértices. En Taubin *et al.* (1996) se presenta una versión mejorada del filtro pasa-bajos λ / μ , el cual produce el mismo efecto que el filtro λ / μ descrito en Taubin (1995a y 1995b), pero en un tiempo mucho menor.

Figura 1.11. Resultado de la aplicación del filtro de suavizado pasa-bajos



Fuente: elaboración propia.

1.3. Simplificación

Para muchas aplicaciones, entre ellas las de análisis por EF, se debe reducir la complejidad de la malla relacionada con el número de vértices y triángulos, mediante un proceso de simplificación. La simplificación o decimación de mallas de superficie tiene como objetivo transformar una malla M en una malla de menor resolución M' , que esté lo más cerca posible de la malla original M pero que abarque un número significativamente menor de vértices y triángulos. Este proceso también reduce la complejidad computacional involucrada en los pasos adicionales de procesamiento para la mejora de la calidad de las mallas. Para este fin se han propuesto varios algoritmos que se pueden clasificar principalmente en tres tipos:

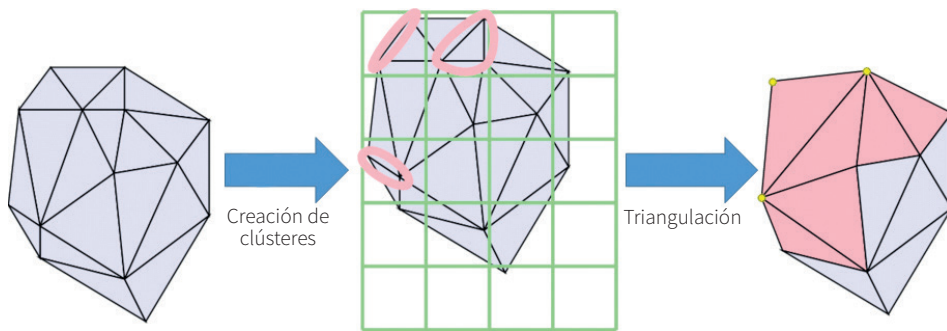
- Agrupamiento de vértices (Rossignac y Borrel, 1993).
- Clasificación de vértices (Schroeder *et al.*, 1992).
- Colapso iterativo de aristas (Hoppe *et al.*, 1993).

Generalmente, los algoritmos de simplificación de mallas toman como parámetro de entrada un cierto número objetivo de polígonos, un cierto porcentaje de reducción de tamaño, o un error que no debe superarse.

1.3.1. Agrupación de vértices

La agrupación de vértices simplifica una malla al fusionar un grupo de vértices en uno solo y eliminar los elementos degenerados (figura 1.12).

Figura 1.12. Agrupación de vértices



Fuente: elaboración propia.

En Rossignac y Borrel (1993) se usó un vértice representativo para reemplazar múltiples vértices de la malla dentro de una celda 3D, usando un volumen subdividido uniformemente para particionar una malla 3D. Shaffer y Garland (2001) propusieron una mejora a este método mediante el uso de particiones adaptativas del espacio basadas en el análisis de la malla. Sin embargo, las mallas simplificadas eran de baja calidad con un límite de error de aproximación muy deficiente.

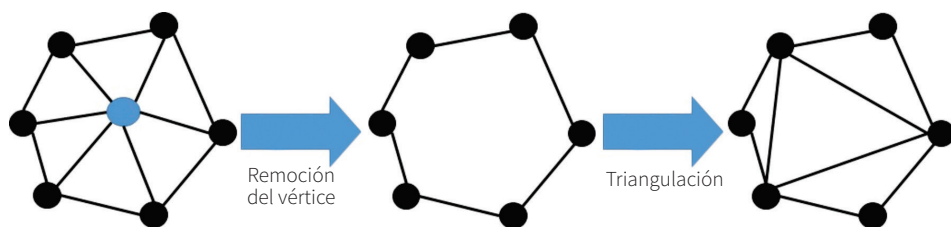
Lindstrom (2000) y Lindstrom y Silva (2001) exploraron la decimación de malla fuera del núcleo basada en la agrupación de vértices. En su primer trabajo, los autores aplicaron la métrica de error cuadrático para guiar la agrupación de vértices. El beneficio fue un mejor posicionamiento de estos y la necesidad de un solo paso sobre la malla de entrada para generar la versión simplificada. Si bien este enfoque inicialmente requería que la memoria del sistema tuviera la versión simplificada de la malla, su segundo trabajo eliminó este requisito almacenando información auxiliar en el disco de forma compacta para evitar costosos accesos aleatorios al mismo.

Los métodos de agrupación de vértices son sencillos de implementar, pero tienen dificultades para controlar la malla simplificada. Adicionalmente, pueden cambiar la topología de la malla de salida y dar como resultado mallas no *manifold*.

1.3.2. Clasificación de vértices

Estos algoritmos permiten clasificar los vértices o aristas con respecto a su relevancia al evaluar el error que se introduciría al eliminarlos. Según esta evaluación, los detalles geométricos se eliminan hasta que se alcanza el tamaño o porcentaje objetivo. Hay varios criterios para seleccionar los vértices que se van a eliminar. Un criterio notable es *la distancia a un eje plano*, como lo describen Schroeder *et al.* (1992). La intuición detrás de este enfoque es eliminar los vértices extraños que están en un eje plano, ya que no contribuyen mucho a la geometría de la malla. Por ejemplo, si un vértice (en color azul en la figura 1.13) está rodeado por un ciclo de triángulos y está *razonablemente* cerca del plano promedio de este grupo de triángulos, este vértice se puede priorizar para su eliminación. Existen otros criterios de eliminación de vértices que se aplican a dimensiones superiores, como es reportado en Renze y Oliver (1996).

Figura 1.13. Clasificación de vértices para simplificación y retriangulación

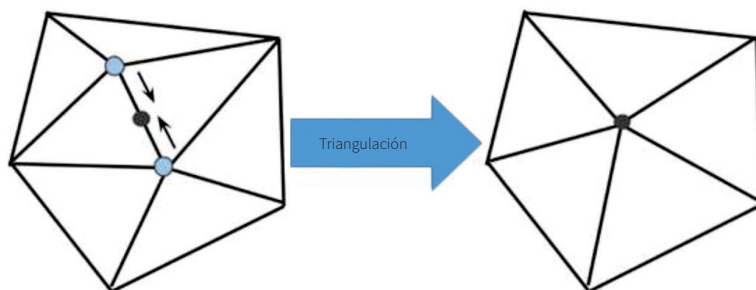


Fuente: elaboración propia.

1.3.3. Colapso de aristas

El colapso de aristas funciona al contraer repetidamente las aristas con los vértices y fusionar las aristas coincidentes resultantes (figura 1.14).

Figura 1.14. Colapso de aristas



Fuente: elaboración propia.

Esto permite que un vértice $\mathbf{v}_s$ se mueva a un vértice adyacente $\mathbf{v}_t$ y luego se elimina. Trotts *et al.* (1998 y 1999) seleccionaron las aristas candidatas basándose en la predicción de las desviaciones resultantes de triángulos y tetraedros locales si estos se colapsaran. Garland y Heckbert (1997) utilizaron métricas de error cuadrático (QEM) para guiar el proceso de colapso de las aristas que han proporcionado una desviación baja del error medio de la malla. Luego, Garland y Zhou (2005) extendieron este trabajo a dimensiones superiores. Lindstrom (2000) propuso una simplificación basada en imágenes, que utiliza imágenes del modelo original contra las de un modelo simplificado para determinar el costo del colapso de una arista. Este enfoque es particularmente útil para aplicaciones que requieren que el modelo simplificado sea visualmente similar al original.

1.4. Requerimientos geométricos

Después de aplicar los métodos de suavizado y simplificación a una malla de superficie generada a partir de imágenes médicas generalmente se le debe optimizar para adecuarla a los requerimientos de las aplicaciones de elementos finitos. Por ejemplo, cuando se aplica un método de simplificación, es esencial que después de eliminar un vértice se realice un remallado local que proporcione una triangulación local en esa área. Entre las posibles triangulaciones, se busca una en la que los triángulos sean lo más equiláteros posible.

En general, los elementos como triángulos, tetraedros y cuadriláteros, que son casi isotrópicos, tienen una alta calidad, mientras que los elementos

altamente anisotrópicos con fuertes deformaciones tienen una calidad baja. Por lo tanto, un triángulo equilátero tiene la máxima calidad y los triángulos con ángulos muy bajos o muy grandes tienen una calidad baja. Básicamente hay dos medidas para la calidad del triángulo:

- La simetría equi-angular (SEA).
- La relación entre el círculo interior y el círculo exterior.

La SEA para triángulos se calcula de acuerdo con la siguiente ecuación:

$$Q_{SEA} = \max \left\{ \frac{\theta_{max} - 60^\circ}{120^\circ} \quad \frac{60^\circ - \theta_{min}}{60^\circ} \right\} \quad (1.6)$$

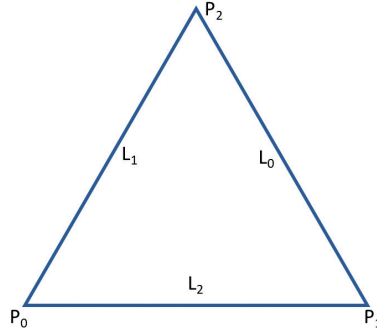
donde θ_{min} y θ_{max} son el ángulo mínimo y máximo de un triángulo T . Para los cuadriláteros, los componentes de 60° y 120° deben reemplazarse por 90° . Los valores de la SEA están en el intervalo $[0,1]$, donde 0 representa la mejor calidad. Sin embargo, en algunos sistemas, la SEA se calcula de manera diferente y los valores de 1 representan allí la mejor calidad.

La relación entre el círculo interior (el círculo máximo que se halla completamente dentro de un triángulo) y el círculo exterior (el círculo más pequeño que contiene completamente un triángulo) es 0,5 para un triángulo equilátero. Por lo tanto, 0,5 representa la calidad óptima y 0 una calidad muy baja (el intervalo está entre $[0, 0,5]$).

1.4.1. Métricas de calidad de los triángulos

Todas las métricas en esta sección se definen en un elemento triangular, como se ilustra en la figura 1.15, en la que los vectores de los lados del triángulo se denotan por:

$$\begin{aligned} \vec{L}_0 &= \vec{P}_2 - \vec{P}_1 \\ \vec{L}_1 &= \vec{P}_0 - \vec{P}_2 \\ \vec{L}_2 &= \vec{P}_1 - \vec{P}_0 \end{aligned} \quad (1.7)$$

Figura 1.15. Numeración de vértices y aristas en un elemento triangular

Fuente: elaboración propia.

Las longitudes de los lados del triángulo se denotan de la siguiente manera:

$$\begin{aligned} L_0 &= \|\vec{L}_0\| \\ L_1 &= \|\vec{L}_1\| \\ L_2 &= \|\vec{L}_2\| \end{aligned} \quad (1.8)$$

y las longitudes de los lados más grandes y más pequeños son, respectivamente,

$$\begin{aligned} L_{min} &= \min(L_0, L_1, L_2) \\ L_{max} &= \max(L_0, L_1, L_2) \end{aligned} \quad (1.9)$$

La calidad $Q(t)$ de un triángulo t se puede calcular como:

$$Q(t) = \frac{6}{\sqrt{3}} \frac{A_t}{S_t E_t} \quad (1.10)$$

donde A_t es el área del triángulo t , S_t es su perímetro medio y E_t es la longitud de su lado más largo (Frey y Borouchaki, 1997). Normalmente, Q_{min} y Q_{avg} se utilizan para la medición de la calidad, que representan la calidad mínima y media de un triángulo, respectivamente.

Área

El área de un triángulo es la mitad de la magnitud del producto transversal de cualquier par de vectores de las aristas adyacentes:

$$A = \frac{1}{2} \|\vec{L}_0 \times \vec{L}_1\| = \frac{1}{2} \|\vec{L}_1 \times \vec{L}_2\| = \frac{1}{2} \|\vec{L}_2 \times \vec{L}_0\| \quad (1.11)$$

Relación de aspecto

La relación de aspecto (RA) es otra métrica utilizada para medir la calidad de los triángulos. Hay varias definiciones de la RA de un triángulo. Por ejemplo, $RA = \frac{\sqrt{3}L}{4A_t}$, donde L es el lado más largo y A_t es el área del triángulo. En algunos artículos simplemente se calcula como la relación entre la longitud del borde más largo y la más corta de un triángulo (Huang y Xu, 2017). En Pébay y Baker (2003) se establece la RA de un triángulo como:

$$RA = \frac{L_{max}}{2\sqrt{3}r} = \frac{L_{max}(L_0 + L_1 + L_2)}{4\sqrt{3}A} \quad (1.12)$$

De manera similar, en algunos casos, se define como la relación entre el *circumradius* y el *inradius* de un triángulo (figura 1.16), calculado como:

$$RA = \frac{8(p - L_0)(p - L_1)(p - L_2)}{L_0 L_1 L_2} \quad (1.13)$$

$$\text{donde } p = \frac{L_0 + L_1 + L_2}{2}$$

Con frecuencia se usará n para representar algún lado arbitrario L_n o vértice p_n del triángulo. Al referirse a la siguiente entrada en sentido antihorario ($n + 1$) o entrada en sentido horario ($n - 1$), se toma el resultado módulo 3

de modo que, por ejemplo, si $(n = 1)$, $n + 1 = 2$ y $n + 2 = 0$. En todos los casos, el valor de RA para el triángulo unitario equilátero es igual a 1.

Inradius

El *inradius* r de un triángulo (figura 1.16a) se puede calcular como:

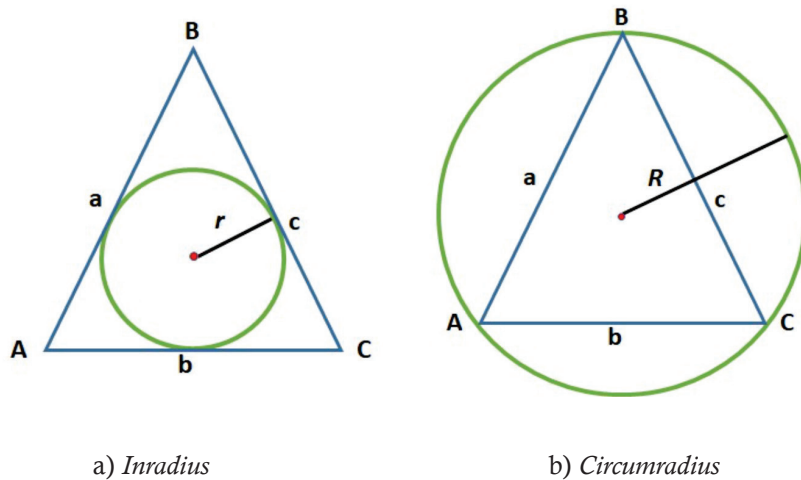
$$r = \frac{2A}{\|\vec{L}_0\| + \|\vec{L}_1\| + \|\vec{L}_2\|} \quad (1.14)$$

Circumradius

El *circumradius* R del triángulo (figura 1.16b) se calcula como:

$$R = \frac{\|\vec{L}_0\| \|\vec{L}_1\| \|\vec{L}_2\|}{2r \|\vec{L}_0\| + \|\vec{L}_1\| + \|\vec{L}_2\|} \quad (1.15)$$

Figura 1.16. Relación entre el *inradius* y el *circumradius* de un triángulo



Fuente: elaboración propia.

Aspecto de Frobenius

El aspecto de Frobenius es la suma de las longitudes de las aristas al cuadrado dividida por el área y normalizada de modo que un triángulo equilátero unitario tenga un valor de 1 (Pébay y Baker, 2003).

$$aspecto_{frobenius} = \frac{\|\vec{L}_0\|^2 + \|\vec{L}_1\|^2 + \|\vec{L}_2\|^2}{4A\sqrt{3}} \quad (1.16)$$

Relación de arista

La relación de arista de un triángulo está dada por la ecuación cuyo valor para un triángulo unitario equilátero es 1 (Pébay y Baker, 2003):

$$relación_{arista} = \frac{L_{max}}{L_{min}} \quad (1.17)$$

Ángulo máximo

El ángulo máximo incluido del triángulo medido en grados está dado por la ecuación cuyo valor para un triángulo unitario equilátero es 60:

$$ángulo_{máximo} = \max_{n \in \{0,1,2\}} \left\{ \arccos \left(\frac{\vec{L}_n \cdot \vec{L}_{n+1}}{\|\vec{L}_n\| \|\vec{L}_{n+1}\|} \right) \left(\frac{180^\circ}{\pi} \right) \right\} \quad (1.18)$$

Ángulo mínimo

El ángulo mínimo incluido del triángulo medido en grados esta dado por la ecuación cuyo valor para un triángulo unitario equilátero es 60 (Pébay y Baker, 2003):

$$\text{ángulo}_{\text{mínimo}} = \min_{n \in \{0,1,2\}} \left\{ \arccos \left(\frac{\overrightarrow{L_n} \cdot \overrightarrow{L_{n+1}}}{\|\overrightarrow{L_n}\| \|\overrightarrow{L_{n+1}}\|} \right) \left(\frac{180^\circ}{\pi} \right) \right\} \quad (1.19)$$

En los últimos años el mejoramiento en los ángulos máximos y mínimos (Erten y Üngör, 2009) y, especialmente, el remallado no obtuso (Khan *et al.*, 2018; Wang *et al.*, 2018; Xu *et al.*, 2019) han recibido una atención significativa.

Relación de aspecto

La relación de aspecto es otra métrica utilizada que está dada por la ecuación:

$$\text{relación}_{\text{aspecto}} = \frac{R}{2r} \quad (1.20)$$

Un valor para un triángulo unitario equilátero es 1, mientras que un valor cero indica un triángulo degenerado (Dietrich *et al.*, 2008).

Número de condición

El número de condición de la matriz jacobiana ponderada está dado por:

$$\text{condición} = \frac{\overrightarrow{L_2} \cdot \overrightarrow{L_2} + \overrightarrow{L_1} \cdot \overrightarrow{L_1} + \overrightarrow{L_1} \cdot \overrightarrow{L_2}}{2A\sqrt{3}} \quad (1.21)$$

En teoría, el número de condición es invariable según qué nodo se calcula, pero el error de truncamiento de coma flotante puede contribuir a diferencias entre los valores calculados para cada nodo (Knupp, 2000; Knupp, 2003). El valor para un triángulo unitario equilátero es 1.

Forma

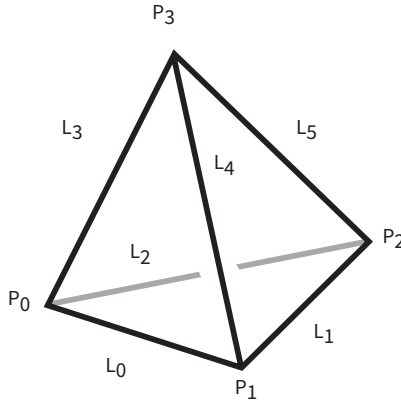
La forma métrica es:

$$\text{forma} = \frac{1}{\text{condición}} \quad (1.22)$$

1.4.2. Métricas de calidad de los tetraedros

Sean $\overline{P}_0, \overline{P}_1, \overline{P}_2, \overline{P}_3$ puntos en $\mathbb{R}^3$ que no están contenidos en un mismo plano, el tetraedro con vértices $\overline{P}_0, \overline{P}_1, \overline{P}_2, \overline{P}_3$ es el poliedro convexo cerrado más simple que tiene 4 caras triangulares y 6 aristas (figura 1.17).

Figura 1.17. Vértices de un tetraedro



Fuente: elaboración propia.

Todas las métricas en esta sección se definen sobre este elemento tetraédrico, donde los vectores de las aristas están dados por:

$$\begin{aligned}\overline{L}_0 &= \overline{P}_1 - \overline{P}_0 & \overline{L}_3 &= \overline{P}_3 - \overline{P}_0 \\ \overline{L}_1 &= \overline{P}_2 - \overline{P}_1 & \overline{L}_4 &= \overline{P}_3 - \overline{P}_1 \\ \overline{L}_2 &= \overline{P}_0 - \overline{P}_2 & \overline{L}_5 &= \overline{P}_3 - \overline{P}_2\end{aligned}\tag{1.23}$$

Las longitudes de las aristas del tetraedro se denotan de la siguiente manera:

$$\begin{aligned}L_0 &= \|\overline{L}_0\| & L_3 &= \|\overline{L}_3\| \\ L_1 &= \|\overline{L}_1\| & L_4 &= \|\overline{L}_4\| \\ L_2 &= \|\overline{L}_2\| & L_5 &= \|\overline{L}_5\|\end{aligned}\tag{1.24}$$

Las longitudes de la arista más grande y más pequeña son:

$$\begin{aligned}L_{\min} &= \min(L_0, L_1, L_2, L_3, L_4, L_5) \\ L_{\max} &= \max(L_0, L_1, L_2, L_3, L_4, L_5)\end{aligned}\tag{1.25}$$

El volumen

El volumen se puede definir en términos de los vectores de las aristas como (Parthasarathy *et al.*, 1994):

$$V = \frac{(\vec{L}_2 \times \vec{L}_0) \cdot \vec{L}_3}{6} \quad (1.26)$$

Inradius

Además, sea r el *inradius*

$$r = \frac{2A}{A} \quad (1.27)$$

donde A es el área de superficie del tetraedro.

$$A = \frac{1}{2} \|\vec{L}_2 \times \vec{L}_0\| + \|\vec{L}_3 \times \vec{L}_0\| + \|\vec{L}_4 \times \vec{L}_1\| + \|\vec{L}_3 \times \vec{L}_2\| \quad (1.28)$$

Circumradius

Sea R el *circumradius* del tetraedro

$$R = \frac{\|\vec{L}_3\|^2 (\|\vec{L}_2 \times \vec{L}_0\|) + \|\vec{L}_2\|^2 (\|\vec{L}_3 \times \vec{L}_0\|) + \|\vec{L}_0\|^2 (\|\vec{L}_3 \times \vec{L}_2\|)}{12V} \quad (1.29)$$

Relación de arista

La relación de arista de un tetraedro es:

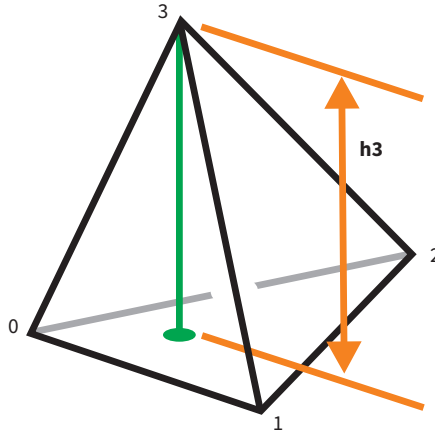
$$relación_{arista} = \frac{L_{max}}{L_{min}} \quad (1.30)$$

El valor para un tetraedro equilátero es 1.

Aspecto delta

El aspecto δ es un número adimensional definido como la relación más pequeña de la altura de un vértice sobre su triángulo opuesto a la raíz cuadrada del área del triángulo en todos los vértices del tetraedro (figura 1.18).

Figura 1.18. Ilustración de la altura h_3 del vértice 3



Fuente: elaboración propia.

En general, se toma (i, j, k, l) como una permutación de $\{0, 1, 2, 3\}$, es decir, $(i, j, k, l) \in G_4$ y $\|\bar{L}_{ab}\|$ es la longitud de la arista que conecta los vértices a y b. Entonces la relación de aspecto δ puede escribirse como:

$$aspecto_{\delta} = \min_i \left\{ condición \frac{h_i}{\sqrt{A_{jkl}}} \right. \quad (1.31)$$

donde A_{jkl} es el área del triángulo opuesto al vértice i y $condición = \frac{\sqrt[4]{108}}{4} \approx 0,805927$ es elegido para que un tetraedro equilátero tenga $q = 1$.

Aspecto de Frobenius

La matriz de aristas del elemento tetraédrico se define de la siguiente manera:

$$T_0 = (L_0 L_1 L_2) \quad (1.32)$$

y sea W la matriz de aristas del tetraedro regular de referencia. Considere la matriz que mapea W en T_0 :

$$A_0 = T_0 W^{-1} \quad (1.33)$$

La norma de Frobenius de A_0 es:

$$|A_0|_F = \sqrt{\text{tr}(A_0^T A_0)} \quad (1.34)$$

y el aspecto de Frobenius es el número de condición asociado con esta norma (Knupp, 2000). Es decir, el aspecto de Frobenius del tetraedro se define como el número de condición de Frobenius normalizado de A_0 , que será igual a 1 cuando el elemento es regular.

Relación de aspecto

La relación de aspecto de un tetraedro T es (Frey y George, 2007):

$$relación_{\text{aspecto}} = \frac{L_{\max}}{2\sqrt{6}r} \quad (1.35)$$

El valor para un tetraedro equilátero es 1.

Jacobiano

Esta métrica se define de la siguiente manera (Knupp, 2003):

$$jacobiano = (\vec{L}_2 \times \vec{L}_0) \cdot \vec{L}_3 \quad (1.36)$$

El valor para un tetraedro equilátero es $\frac{\sqrt{2}}{2}$

Ángulo mínimo

El ángulo diédrico (no orientado) de 2 caras del tetraedro adyacentes a lo largo de la arista i ($0 \leq i \leq 5$) se mide en grados,

$$\alpha_i = \frac{180^\circ}{\pi} \arccos(\vec{n}_{i1} \cdot \vec{n}_{i2}) \quad (1.37)$$

donde $\vec{n}_{i1}$ y $\vec{n}_{i2}$ son vectores unitarios normales a las 2 caras del tetraedro adyacentes a la arista i . Por lo tanto, el ángulo diédrico mínimo (no orientado) del tetraedro, medido en grados, es:

$$\text{ángulo}_{\text{mínimo}} = \min_{i \in \{0, 1, 2, 3, 4, 5\}} \alpha_i \quad (1.38)$$

El valor para un tetraedro equilátero es $\frac{180^\circ}{\pi} \arccos \frac{1}{3} \approx 70,528779^\circ$.

Relación de radios

Esta medida se conoce comúnmente como la relación de radios, ya que es la relación normalizada del radio de la esfera inscrita con respecto al radio de la circunferencia (Parthasarathy *et al.*, 1994).

$$\text{relación}_{\text{radios}} = \frac{R}{2r} = \frac{\|\vec{L}_3\|^2 (\vec{L}_2 \times \vec{L}_0) + \|\vec{L}_2\|^2 (\vec{L}_3 \times \vec{L}_0) + \|\vec{L}_0\|^2 (\vec{L}_3 \times \vec{L}_2) A}{108V^2} \quad (1.39)$$

La relación de radios es el cociente de estos dos radios normalizados por $\frac{1}{3}$, de modo que un tetraedro equilátero tiene una calidad de 1.

La forma

Knupp (2003) define la métrica de calidad de forma como:

$$\text{forma} = \frac{3(\text{jacobiano}\sqrt{2})^{\frac{2}{3}}}{\frac{3}{2}(\vec{L}_0 \cdot \vec{L}_0 + \vec{L}_2 \cdot \vec{L}_2 + \vec{L}_3 \cdot \vec{L}_3) - (\vec{L}_0 \cdot -\vec{L}_2 + \vec{L}_0 \cdot \vec{L}_3 + -\vec{L}_2 \cdot \vec{L}_3)} \quad (1.40)$$

El valor para un tetraedro equilátero es 1.

1.4.3. Efecto de un elemento malo

Se requieren varias propiedades de regularidad de las mallas de triángulos y tetraedros, generalmente prescritas en términos de características geométricas, para garantizar resultados adecuados de interpolación y convergencia del método de elementos finitos (anexo A). Zlámal (1968) introdujo la llamada *condición de ángulo mínimo* que asegura la convergencia de las aproximaciones de los elementos finitos para resolver problemas de valores de frontera elípticos lineales para $d = 2$. Esta condición requiere que exista una constante $\alpha_0 > 0$ tal que el ángulo mínimo α_T de cada triángulo T en toda la triangulación satisfaga:

$$\alpha_T \geq \alpha_0 \quad (1.41)$$

La condición de Zlámal se puede generalizar en R^d para cualquier $d \in 2, 3, \dots$, de modo que todos los ángulos diedros de los elementos de menor dimensión están delimitados desde abajo por una constante positiva (Brandts *et al.*, 2008). Es decir, existe una constante $\alpha_0 > 0$ tal que para cualquier tetraedro se tiene:

$$\alpha_D^T \geq \alpha_0 \quad y \quad \alpha_F^T \geq \alpha_0 \quad (1.41)$$

donde $\alpha_D^T > 0$ es el mínimo de valor de los ángulos diedros entre las caras de T y $\alpha_F^T > 0$ es el ángulo mínimo en las 4 caras triangulares de T .

Es común en el análisis de EF, al igual que en la geometría discreta y computacional, distinguir cualitativamente entre los llamados elementos *bien formados*, es decir, casi regulares con ángulos cerca a los 60 grados y los *mal formados*. Para procesos de interpolación, los efectos de un triángulo o tetraedro mal formado son completamente locales. Los grandes errores puntuales no se extienden más allá del elemento malo. Sin embargo, en los métodos numéricos el error de discretización se comporta de manera diferente. La solución $u(x)$ producida por el método de EF asume un error $f - u(x)$ que está limitado de manera global. Si hay demasiados elementos de baja

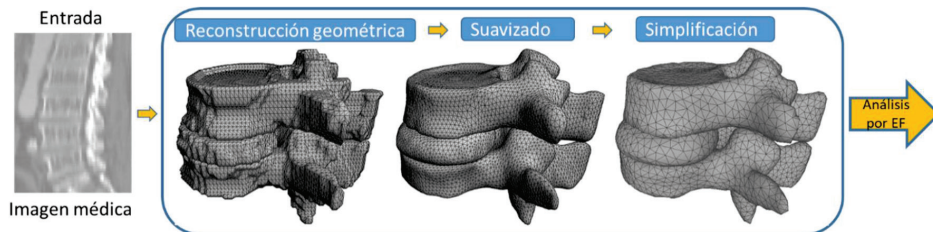
calidad en la malla, el error de solución aumentará y el tiempo de solución se alargará o incluso no podrá calcularse (anexo A).

Por otro lado, la matriz de rigidez de un elemento con un valor propio máximo grande eleva igualmente el valor propio máximo de la matriz de rigidez global K . Si una malla tiene solo uno o dos elementos malos impondrá solo unos pocos valores propios grandes. El resto del espectro de K puede estar dentro de un rango limitado (anexo A). Algunos procesadores de EF iterativos pueden aprovechar esto, y otros no. Los métodos primitivos, como el método jacobiano o el método del descenso más pronunciado, se comportan mal cuando el número de condición de K es grande, incluso si solo un valor propio malo es el responsable. Por el contrario, algunos métodos del subespacio de Krylov funcionan bien en estas circunstancias. Por ejemplo, el método de gradiente conjugado, que solo funciona con matrices de rigidez simétricas y, por lo tanto, solo con ecuaciones diferenciales autoadjuntas, como la ecuación de Poisson, se puede sortear cada valor propio malo con una iteración adicional, después de lo cual funciona tan bien como lo haría si el valor propio malo no estuviera en el espectro. Así las cosas, el efecto de algunos elementos malos depende en gran medida del método utilizado para resolver las ecuaciones lineales.

Capítulo 2. *Framework* para el modelado geométrico de estructuras anatómicas

En este capítulo se presentan las características principales del *framework* desarrollado para el apoyo al proceso de modelado geométrico, a partir de imágenes médicas, de estructuras anatómicas (figura 2.1).

Figura 2.1. *Framework* para el modelado geométrico de estructuras anatómicas



Fuente: elaboración propia.

Este *framework* recibe como entrada una imagen médica segmentada, lo cual permite recuperar rápidamente, dada la etiqueta de un punto p (vóxel), el componente conectado o región a la que pertenece, de acuerdo con el modelo matemático del numeral 2.1. A partir de la imagen médica se reconstruye la geometría de las diferentes partes que conforman la estructura anatómica Ω utilizando el algoritmo *flying edges* (FE), propuesto por Schroeder *et al.* (2015) (numeral 2.2). Las mallas de superficie obtenidas se deben configurar dentro del modelo geométrico M de la estructura anatómica resolviendo el posible problema de traslape entre las regiones (numeral 2.3). Para completar

el modelo geométrico apto para el análisis por EF, se realiza un proceso de suavizado (numeral 2.4) y otro de simplificación (numeral 2.5). Finalmente, el modelo geométrico es exportado al *software* de EF donde se le asignan cargas, condiciones de frontera y se convierten en mallas de volumen.

2.1. Modelado matemático

Una estructura anatómica se puede describir matemáticamente como:

$$\Omega = \bigcup_{i=1}^n \Omega_i \quad (2.1)$$

donde Ω es la estructura anatómica modelada en $\mathbb{R}^3$, Ω_i es la i -ésima parte o región que pertenece a Ω y n es el número de partes. Cuando $n = 1$ se dice que la estructura anatómica es *simple*, en caso contrario, se dice que la estructura es *compuesta*.

El espacio euclidiano $\mathbb{R}^3$ puede extenderse a $\mathbb{R}^4$ para representar la etiqueta de la región m de cualquier punto p de la estructura anatómica Ω mediante la expresión:

$$\mathbf{p} = \{x, y, z, m\}^T \quad (2.2)$$

Dado que cada parte Ω_i tiene una superficie límite cerrada $\Gamma_i = \partial \Omega_i$, la interfaz entre 2 partes i y j se puede denotar por:

$$\Omega_i \bigcap_{i \neq j} \Omega_j = \Gamma_i \bigcap_{i \neq j} \Gamma_j = \Gamma_{ij} = \Gamma_{ji} \quad (2.3)$$

Si 2 partes i y j no tienen frontera en común, entonces la interfaz estará vacía.

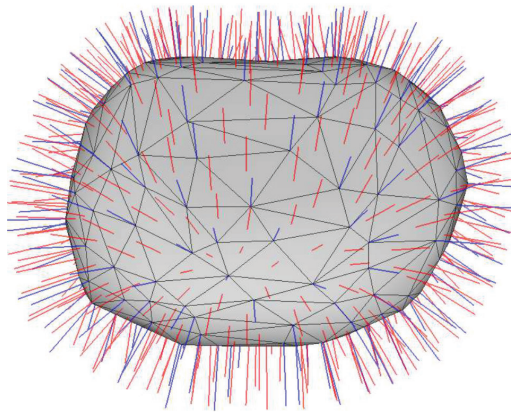
$$\Omega_i \bigcap_{i \neq j} \Omega_j = \Gamma_i \bigcap_{i \neq j} \Gamma_j = 0 \quad (2.4)$$

El *framework* propuesto se basa en esta propiedad teniendo en cuenta que en los cuerpos vivos las partes de las estructuras anatómicas no se tocan directamente entre sí con una interfaz discontinua. En cambio, las regiones en contacto a menudo están cubiertas de grasa, tejido conectivo o líquido. Bajo este supuesto, las regiones que conforman una estructura anatómica estarían separadas por un material de fondo, que contiene todas las partes de la estructura y siempre evita su contacto directo mediante una capa de separación, lo que hace que la interfaz entre dos regiones sea siempre vacía.

2.2. Reconstrucción geométrica de las regiones

El primer paso en el modelado de una estructura anatómica Ω es la generación de las geometrías de las regiones Ω_i que conforman dicha estructura. A partir de cada región Ω_i se genera una malla de superficie M_i , que aproxima la superficie de la región mediante un conjunto de triángulos conectados por sus lados o vértices comunes. Es decir, una malla de superficie $M_i = (V, T)$ es una malla triangular cerrada que rodea el interior de una región de interés Ω_i , donde $V = \{v_r : 1 \leq r \leq n_v\}$ denota el conjunto de vértices, y $T = \{t_k : 1 \leq k \leq n_t\}$ representa el conjunto de triángulos que deben ser orientados. La figura 2.2 muestra la orientación (las normales) de los vértices en color azul y de los triángulos en color rojo de una malla de superficie.

Figura 2.2. Orientación de vértices y triángulos de una malla de superficie



Fuente: elaboración propia.

Los triángulos t_k^i de la malla M_i deben cumplir que:

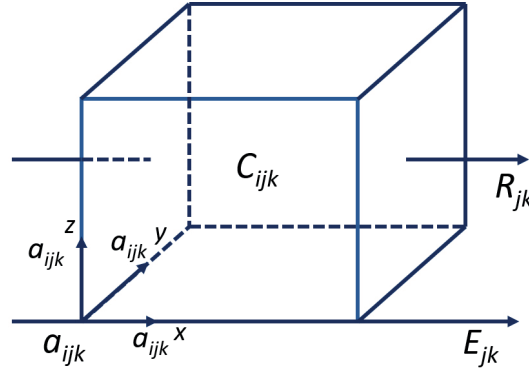
- $M_i = \bigcup_{k=1}^{n_T} t_k^i$
- Cada triángulo t_k^i es cerrado y con interior no vacío.
- Para diferentes triángulos t_p^i y t_q^i la intersección de sus interiores es vacía.
- Para diferentes triángulos t_p^i y t_q^i su intersección es un lado común, un vértice común o es vacía.

La mayoría de los procesos de extracción de mallas de superficie M_i a partir de imágenes médicas se basan en los algoritmos MC y DC (numeral 1.1.2). Ambos algoritmos toman como entrada un conjunto de datos volumétricos y forman un conjunto de cubos lógicos para examinar cada celda y así generar una triangulación a partir de una iso-superficie que se cruza con la celda.

Para el *framework* se utiliza el algoritmo *flying edges* (FE), propuesto por Schroeder *et al.* (2015), el cual utiliza una estrategia que adapta localmente el tamaño de los elementos t_k^i de la malla M_i . El algoritmo FE implementa, de forma paralela, el algoritmo MC en una CPU que cuente con varios núcleos. Este algoritmo opera en una cuadrícula estructurada con N valores escalares dispuestos topológicamente en una cuadrícula regular de puntos de dimensión $(l \times m \times n)$ con valores s_{ijk} . Las celdas de la cuadrícula se definen a partir de los ocho puntos adyacentes asociados con valores escalares en las direcciones $i + (0,1)$, $j + (0,1)$, $k + (0,1)$. Los bordes de la cuadrícula E_{ijk} van en las direcciones de la fila E_{jk} , la columna E_{ik} y la pila E_{ij} ; y se componen de los bordes de la celda e_{ijk} . Las filas de celdas de la cuadrícula R_{ijk} consisten en todas las celdas v_{ijk} que tocan tanto E_{ijk} como $E_{(i+1)(j+1)(k+1)}$. Por ejemplo, un borde de fila x de longitud n con $0 < i < n$:

$$E_{ij} = \bigcup_{i=0}^{n-1} e_{ijk} \quad y \quad R_{ij} = \bigcup_{i=0}^{n-1} v_{ij} \quad (2.5)$$

donde cada v_{ijk} tiene un eje de celda asociado a_{ijk} que se refiere a los tres bordes de celda que emanan del punto ubicado en s_{ijk} en las direcciones positivas de fila x , columna y y pila z (figura 2.3).

Figura 2.3. Sistema de coordenadas en el algoritmo FE

Fuente: elaboración propia.

La figura 2.3 muestra cómo el algoritmo procesa la celda a_{ijk} en paralelo a lo largo de los bordes E_{jk} de x para contar las intersecciones y el número de triángulos de salida. Las primeras y últimas intersecciones a lo largo de E_{jk} se utilizan para realizar un recorte computacional cuando se genera una salida en paralelo sobre las filas de celdas de la cuadrícula R_{jk} .

El isocontorno C se define por un isovalor v : $C(v) = \{p | F(p) = v\}$, donde $F(p)$ mapea el punto $p \in \mathbb{R}^n$ a un número real (es decir, el isovalor v). La malla de superficie resultante M_i es una superficie lineal continua por partes. Se debe tener en cuenta también que C se cruza con e_{ijk} cuando algún p se encuentra a lo largo del borde de la celda e_{ijk} . Se supone que el valor escalar varía linealmente sobre e_{ijk} , por lo que C puede cruzarse como máximo en un solo punto a lo largo del borde.

Para realizar el análisis por EF de cualquier estructura anatómica Ω se requiere que cada región se represente con mallas M_i de buena calidad, lo cual se logra mediante el ajuste del tamaño y el ángulo mínimo de los triángulos t_k^i que conforman la malla M_i , como se explica en el numeral 1.4. En este trabajo se tienen en cuenta 2 tipos de triángulos, los agudos y los planos. Un triángulo agudo se refiere a aquel que tiene 1 ángulo cercano a 0 y 2 cerca de 90 grados, mientras que un triángulo plano tiene 1 ángulo cercano a 180 grados y 2 cercanos a 0. Adicionalmente, las mallas de superficie M_i deben representar los detalles de la superficie de las regiones Ω_i utilizando elementos t_k^i lo más homogéneos posible (numeral 1.4).

Algoritmo 2.1. Remallado isotrópico con base en el método AEAP

Entrada: malla triangular *2-manifold* M_i representando la región Ω_i , número de vértices n definido por el usuario.

Salida: malla M_i' con triángulos lo más equiláteros posible.

1. Inicialización:

- a. Insertar un número de vértices de muestreo.
- b. Eliminar los vértices de la malla original.
- c. Volver a triangular los vértices de muestreo recién insertados creando una teselación mutua.
- d. Identificar las aristas y vértices importantes para su conservación utilizando un ángulo diédrico de 40 grados que se mantiene durante el proceso de remallado.

2. Optimización de la conectividad:

Para cada lado l en la malla M_i'

- a. Calcular el costo de la optimización de valencia de cada vértice con 4 vecinos.
- b. Voltar el borde l .
- c. Volver a calcular el costo de voltear el borde l en términos de optimización de valencia con los nuevos vecinos.
- d. Si el costo de la optimización de valencia coincide con el valor ideal después de voltear el borde l , aceptar el cambio de borde, de lo contrario, deshacer el paso 2-b.

3. Implementación del AEAP:

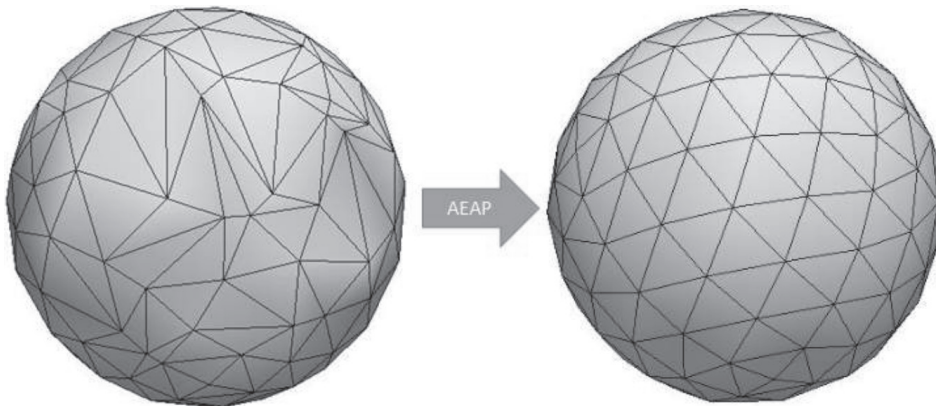
- a. Calcular la matriz de transformación local para resolver la función de minimización de energía.
- b. Proyectar los vértices recién generados en la malla original.

4. Repetir el paso 2 hasta que los vértices de muestra recién insertados n sean menores que el número original de vértices en la malla N .

De esta manera, en este *framework* se quiere maximizar el ángulo mínimo de los triángulos t_k^i de las mallas de superficie M_i buscando que sean lo más equilátero posible AEAP (*As Equilateral As Possible*) (Du *et al.*, 2015). El algoritmo 2.1 describe el funcionamiento del método AEAP, el cual utiliza un proceso iterativo de dos pasos. La conectividad de la malla se optimiza para reducir la cantidad de vértices, seguida de un ajuste local y la agregación de un triángulo equilátero t_k^i .

La figura 2.4 muestra cómo todos los triángulos equiláteros encajan en una malla coherente con formas y características geométricas parecidas a la malla original. Obviamente, algunos se deformarán durante este proceso, sin embargo, si la deformación hace que el triángulo original esté cerca del equilátero, no cuenta como deformación. Esto permite que la deformación transforme cada triángulo lo más equilátero posible.

Figura 2.4. Resultado de aplicar el método AEAP



Fuente: elaboración propia.

2.3. Reconstrucción geométrica de las estructuras anatómicas

Una vez se obtiene la malla de superficie M_i , se debe agregar al modelo geométrico M que va a representar a toda la estructura anatómica Ω . El algoritmo 2.2 muestra los diferentes pasos para obtener un modelo geométrico M de una estructura anatómica Ω compuesta de una o varias regiones Ω_i .

Algoritmo 2.2. Modelado geométrico de estructuras anatómicas compuestas

Entrada: imagen médica segmentada con un conjunto de n regiones que componen la estructura anatómica Ω .

Salida: modelo geométrico M compuesto por un conjunto de mallas de superficie M_i .

Para todas las regiones $\Omega_i \in \Omega : 1 \leq i \leq n$.

Extraer región Ω_i .

Generar malla de superficie M_i .

Agregar M_i al modelo geométrico M .

Verificar y resolver traslape de M_i .

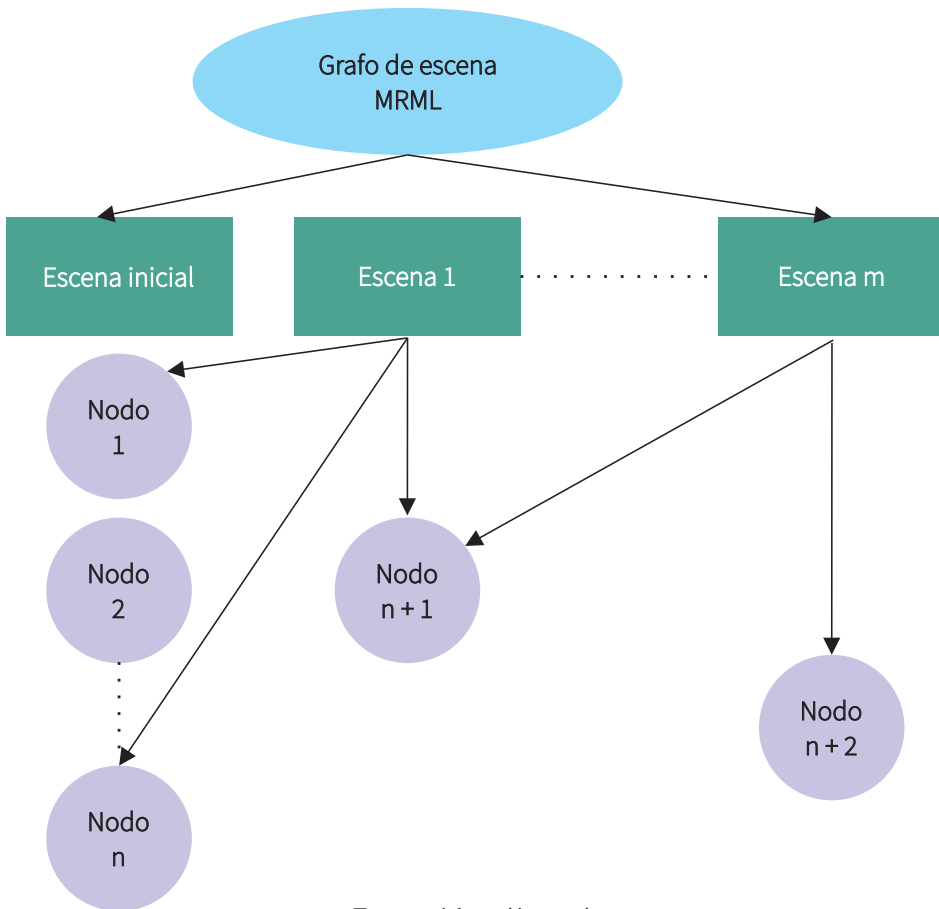
Cuando la estructura anatómica está compuesta por una sola región Ω_i , entonces $\Omega = \Omega_i$, $i = 1$. En este caso, el modelo geométrico M está compuesto por una única malla de superficie M_i . En otros casos, las estructuras anatómicas están compuestas por varias regiones Ω_i para lo cual es importante manipular todas las mallas M_i del modelo geométrico M como un todo y no como partes aisladas.

En trabajos anteriores (Gómez-Mora y Flórez-Valencia, 2016; Gómez-Mora, 2018) se presentó el método RAM. Este, combina las características de un conjunto de superficies Γ_i y regiones Ω_i mediante la relación $\Gamma_i = \partial \Omega_i$. El método RAM implementa un vector de funciones implícitas (VFI) para representar y manipular tanto la información de la geometría de las regiones individuales Ω_i , como de toda la estructura. Sin embargo, la generación del modelo geométrico M a partir del VFI es compleja y propensa a errores debido a que la representación implícita de las regiones agrega un suavizado a las mallas M_i generando en algunas ocasiones solapamiento entre las regiones.

Para este *framework* se optó por la utilización de un grafo de escena MRML (*Medical Reality Markup Language*), el cual permite mantener sincronizadas las diferentes mallas M_i . Adicionalmente, el cambio en una región se ve reflejado en las demás regiones del grafo, lo que permite hacer un posprocesamiento en las mallas M_i antes de generar el modelo geométrico final M . Cada región Ω_i está asociada a un nodo MRML, que es un contenedor de datos (figura 2.5).

El modelo geométrico M es representado por los nodos activos en la última escena. Se puede observar en la figura 2.5 que en la escena 1 se encuentran activos el nodo 1, el nodo n y el nodo $n+1$, mientras que en la escena m se encuentran activos el nodo 1, el nodo $n+1$ y el nodo $n+2$. La información relacionada de esta última escena sería la que se guarde como parte del modelo geométrico M de la estructura Ω .

Figura 2.5. Grafo MRML



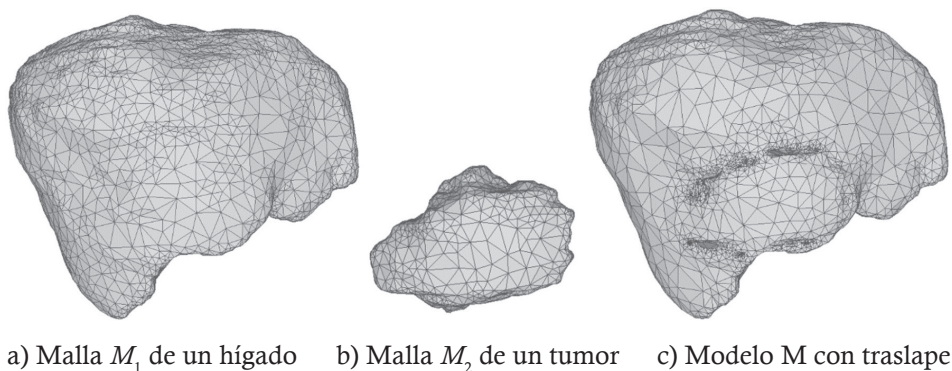
Fuente: elaboración propia.

Adicionalmente, es importante que las geometrías de las regiones Ω_i no se traslapen entre sí. Cuando las regiones de una estructura compuesta se encuentran en una única imagen médica con segmentaciones múltiples (figura 2.2b), se puede generar cada malla de superficie M_i sin inconvenientes

de solapamiento entre las mismas mallas. Sin embargo, si las regiones que conforman la estructura anatómica Ω se encuentran en diferentes volúmenes (figura 2.2a), el modelo geométrico M podría tener mallas de superficie M_i que se traslapen o intercepten, lo que obliga a realizar un proceso de resolución de dichos traslapes.

En la figura 2.6 se muestra el modelado de una estructura anatómica compuesta por dos regiones: un hígado (figura 2.6a) y un tumor hepático (figura 2.6b), que se encuentran en diferentes segmentaciones. En la figura 2.6c se puede ver el resultado, donde es evidente que el tumor sobresale de la malla del hígado, cuando debería estar contenido dentro de esta.

Figura 2.6. Generación errónea de un modelo geométrico M



a) Malla M_1 de un hígado b) Malla M_2 de un tumor c) Modelo M con traslape

Fuente: elaboración propia.

Para resolver los traslapes entre las mallas M_i se usa una estrategia de restas o sustracciones booleanas utilizando *vtkbool*, que es una extensión de la biblioteca de gráficos VTK (<http://www.vtk.org/>) (Schroeder *et al.*, 1998). *Vtkbool* puede tomar dos mallas poligonales M_i y M_j y produce una malla M'_i , de la cual se elimina la parte que se cruza con M_j . Se debe tener en cuenta que el orden en el que se ejecutan las operaciones booleanas de resta sobre las mallas M_i determina la configuración final del modelo geométrico M . La malla M_i que se procesa primero probablemente perderá la mayor parte de su geometría y no tendrá influencia en las demás mallas. La malla M_i que se procesa en último lugar no se verá afectada en absoluto por las otras mallas.

Así mismo, se elige siempre en último lugar para ser procesadas las mallas M_i que representan los huesos, ya que calidad de su segmentación es la más alta debido a que son relativamente fáciles de distinguir en una imagen médica, por lo tanto, el resultado de la segmentación es más confiable y menos propenso a errores. Adicionalmente, las mallas M_i que representan músculos y huesos están ordenadas entre sí de manera descendente de acuerdo con su volumen. De esta forma, las mallas M_i que sean más pequeñas se procesarán más tarde, lo que significa que serán las últimas en ser restadas. Esto es apropiado, ya que el impacto de los cambios en los objetos pequeños suele ser mayor.

Finalmente, para reducir el número de restas booleanas se usa el método de cuadro delimitador (*bounding box*) para predeterminar si dos mallas se pueden traslapar y así evitar comparar mallas que estén distantes una de la otra.

2.4. Suavizado

En la generación de un modelo geométrico M para simulaciones por EF, es esencial contar con mallas M_i no solo precisas sino suaves, porque los detalles de la superficie, relacionados con defectos de *aliasing* (figura 2.7a), pueden tener un impacto negativo en el análisis numérico e incluso pueden dificultar la convergencia de la simulación. Sin embargo, se debe tener en cuenta que el proceso de suavizado puede reducir tanto el área de la superficie (Taubin, 1995b) como el volumen de la estructura anatómica Ω (Zhou *et al.*, 2005; Zhang *et al.*, 2009), que depende principalmente de la cantidad de suavizado que se realice (numeral 1.2).

La preservación del volumen es otra característica importante de los modelos 3D requerida en diferentes campos, desde la computación gráfica para hacer que las deformaciones parezcan más naturales (Roussellet, 2018), hasta sistemas dinámicos para lograr simulaciones más estables (Ozbolat y Koc, 2011). El algoritmo de suavizado laplaciano con corrección (Vollmer *et al.*, 1999) y el filtro pasa-bajos (Taubin, 1995a) compensan la pérdida de volumen logrando resultados muy similares.

Taubin (1995b) propuso la siguiente función de transferencia de segundo grado para resolver el problema de la contracción del volumen:

$$f(k) = (1 - \lambda k)(1 - \mu k) \quad (2.6)$$

Esta función se puede implementar como dos pasos consecutivos de suavizado laplaciano con diferentes factores de escala λ y μ (algoritmo 2.3). El primero con $\lambda > 0$, y el segundo con $\mu < -\lambda < 0$. Es decir, después de realizar el paso de suavizado laplaciano con factor de escala positivo λ (paso de contracción), se realiza un segundo paso de suavizado laplaciano con factor de escala negativo μ (paso de expansión).

Algoritmo 2.3. Filtro de suavizado pasa-bajos de Taubin

Entrada: malla triangular *2-manifold* M_i representando la región Ω_i , número de iteraciones N , valor de la banda de paso a ser suavizada $\mathbf{x}$.

Salida: malla M_i' suavizada aplicando filtro pasa-bajos de Taubin.

Para todas las iteraciones $i : 0 \leq i < N$

$$M_i' = \text{Laplaciano}(M_i, \mathbf{x})$$

si i es par:

$$\mathbf{x} = \mathbf{x} + \lambda M_i'$$

en caso contrario:

$$\mathbf{x} = \mathbf{x} + \mu M_i'$$

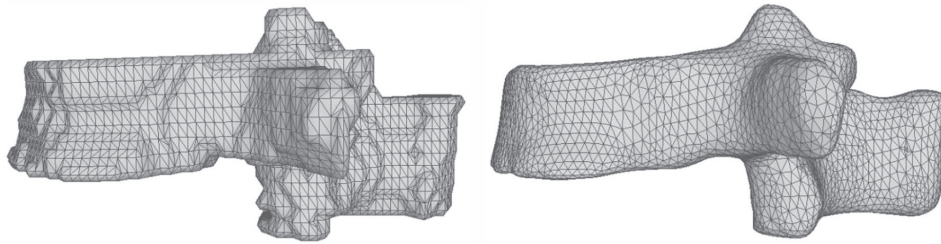
En este *framework* se utiliza la versión del filtro pasa-bajos propuesta por Taubin *et al.* (1996). Este algoritmo produce el mismo efecto que el filtro λ/μ descrito en Taubin (1995b), pero en menor tiempo. Además, se ajusta muy bien al tipo de superficies biológicas, que son cerradas y con una distribución de vértices casi uniforme.

El proceso de suavizado se puede controlar mediante dos variables (algoritmo 2.3). La primera es el número máximo de pasadas de suavizado determinado por el número de iteraciones N . Entre diez y veinte iteraciones es todo lo que normalmente se necesita y este valor tendrá correspondencia directa con el grado del polinomio que se utiliza para aproximar la función sinc rectangular (Gómez-Mora *et al.*, 2021). La segunda variable es la especificación de la banda de paso $\mathbf{x}$ para el filtro sinc rectangular. El valor de

esta variable se especifica como un número de punto flotante entre 0 y 2, donde un buen valor es 0,1, ya que los valores bajos producen más suavizado (Taubin, 2000).

En la figura 2.7a se muestra una malla de superficie M_i correspondiente a una vértebra humana generada mediante el algoritmo FE. Aunque se observa que el algoritmo FE genera una buena superficie de la región segmentada Ω_i y la malla M_i tiene elementos homogéneos, el resultado en general muestra efectos de escalón. La figura 2.7b muestra el modelo de superficie M'_i , aplicando el filtro de suavizado pasa-bajos λ / μ , el cual se adapta a la curvatura de la vértebra, reflejando mejor la suavidad que generalmente tienen las vértebras y, en general, todas las estructuras anatómicas.

Figura 2.7. Generación de la malla de superficie de una vértebra humana



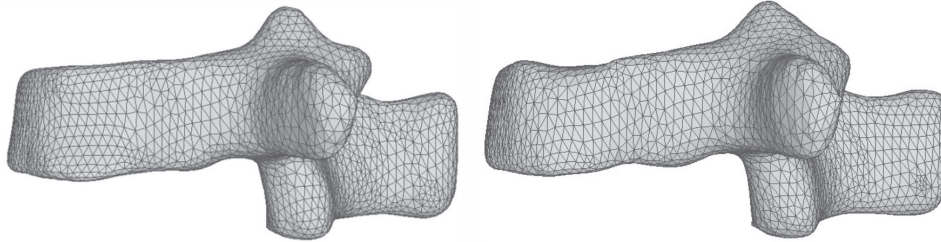
a) Malla M_i , con 6 506 vértices y 13 012 caras, generada con el algoritmo FE.

b) Malla M'_i , con un suavizado del 50 %. 6 506 vértices y 13 012 caras.

Fuente: elaboración propia.

Adicionalmente, se puede observar en la figura 2.7 que el tamaño de la malla M_i es casi el mismo, solo cambia la suavidad que aplica el algoritmo pasa-bajos en la figura 2.7b.

La figura 2.8a muestra el resultado de aplicar un porcentaje de suavizado del 70 % sobre la malla M_i de la figura 2.7a, mientras que la figura 2.8b muestra el resultado de un suavizado del 80 %. Aunque la pérdida de volumen de la malla de superficie M_i obtenida mediante el algoritmo FE es mínima con respecto al volumen de la región segmentada Ω_i , se debe aclarar que las mallas suavizadas M'_i , M''_i y M'''_i sí lo hacen, lo que se ve reflejado en la reducción del número de vértices y caras de las malla suavizadas en la figura 2.8.

Figura 2.8. Suavizado de la malla de superficie M_i 

a) Malla M_i'' con un suavizado
del 70 %. 6198 vértices, 12 396 caras

b) Malla M_i''' con un suavizado
del 80 %. 5948 vértices, 11 896 caras

Fuente: elaboración propia.

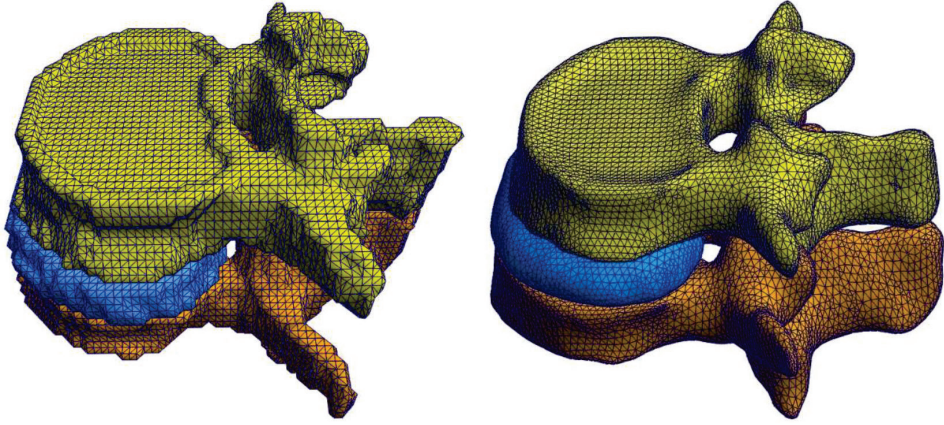
Para el caso de estructuras anatómicas compuestas por varias regiones Ω_i , se deben generar las mallas de superficie M_i sin aplicar el filtro de suavizado λ/μ (figura 2.9a). En este caso, el filtro de suavizado se debe aplicar a todo el modelo geométrico M (figura 2.9b). Esto permite conservar la interfaz cerrada entre las diferentes mallas de superficie M_i al tiempo que evita que haya solapamientos de las mismas, conservando la topología de todo el modelo geométrico M .

La figura 2.9 muestra el modelado de una estructura anatómica compuesta de tres regiones: dos vértebras y un disco intervertebral. Se puede observar que el proceso de suavizado asegura que todas las mallas de superficie M_i se suavicen y encajen sin problemas, como las piezas de un rompecabezas.

Sin embargo, para evitar que las malla M_i se superpongan al momento de aplicar el algoritmo de suavizado se debe verificar en cada paso de suavizado la ocurrencia de eventos de contacto entre dos mallas M_i y M_j mediante consultas de proximidad (Machado *et al.*, 2015).

En la tabla 2.1 se listan las cinco medidas de proximidad más comunes con su definición y condición.

Figura 2.9. Modelo geométrico compuesto por tres mallas de superficie M_i



a) Modelo geométrico M sin suavizado b) Modelo con un suavizado del 50 %

Fuente: elaboración propia.

Tabla 2.1. Tipos de consultas de proximidad

Consulta	Definición	Condición
Detección de colisión	Comprueba si dos regiones M_i y M_j se superponen en el espacio o si sus límites comparten al menos un punto en común	$col(M_i, M_j) = M_i \cap M_j$ (2.7)
Distancia de separación	Longitud del segmento de línea más corto que une dos conjuntos de puntos	$d_{sep}(M_i, M_j) = \min_{p \in M_i} \min_{q \in M_j} p - q $ (2.8)
Distancia de influencia	Desviación máxima de un conjunto del otro	$d_{inf}(M_i, M_j) = \max_{p \in M_i} \max_{q \in M_j} p - q $ (2.9)
Distancia de Hausdorff	Distancia máxima entre los puntos de dos conjuntos	$d_{haus}(M_i, M_j) = \max_{p \in M_i} \min_{q \in M_j} p - q $ (2.10)
Profundidad de penetración	Distancia mínima necesaria para trasladar un conjunto para separarlo del otro	$d_{pen}(M_i, M_j) = \min \ v\ $ tal que $\min_{p \in M_i} \min_{q \in M_j} p - q + v > 0$ (2.11)

Fuente: elaboración propia.

2.5. Simplificación

En simulaciones por EF el número de elementos (tetraedros) define los sistemas de ecuaciones que se van a resolver. Un modelo geométrico denso M conduce a una malla de volumen con un gran número de elementos. Para mantener el número de elementos de volumen razonablemente bajo y así evitar tiempos de simulación demasiado largos, es necesario limitar el número de elementos del modelo geométrico M mediante un proceso de diezmado o simplificación de las mallas de superficie M_i (numeral 1.3).

Para reducir el número de triángulos de una malla M_i se necesita eliminar información redundante (Kobbelt *et al.*, 1998). En general, un vértice v_i es redundante si se puede cambiar su vecindad por otra con un número menor de triángulos. Al igual que en el proceso de suavizado (numeral 2.4), el proceso de simplificación también cambia ligeramente la forma de la región (Schroeder, 1997), esto puede hacer que las mallas de superficie M_i se solapen o intercepten dentro del modelo geométrico M de las estructuras anatómicas compuestas, por lo que esta acción se debe aplicar a estructuras anatómicas simples o a estructuras anatómicas compuestas antes del proceso de suavizado.

Algoritmo 2.4. Simplificación mediante clasificación de vértices

Entrada: malla triangular *2-manifold* M_i representando la región Ω_i , porcentaje de reducción de tamaño n .

Salida: malla M'_i simplificada.

Mientras haya un vértice v_i que se pueda quitar

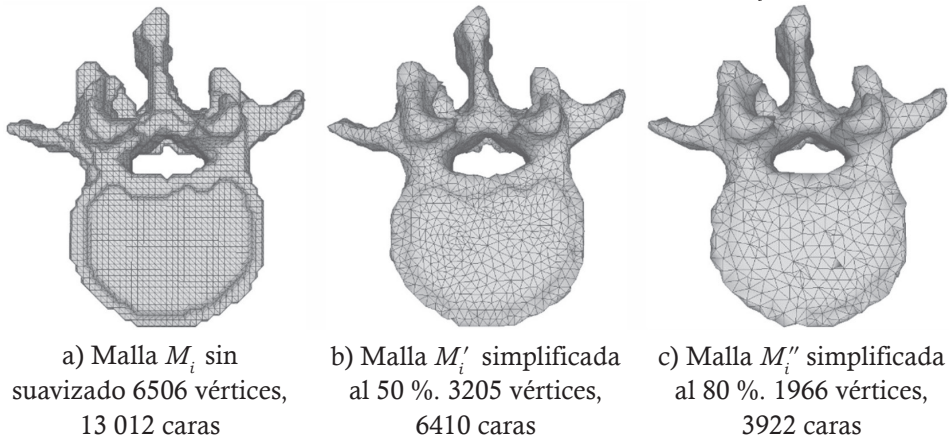
Aplicar el operador de eliminación de vértices a v_i

Volver a triangular el agujero creado de lado k

En este *framework* se aplica una técnica de clasificación de vértices (Renze y Oliver, 1996). El algoritmo 2.4 muestra que al eliminar un vértice se eliminan también sus bordes y caras adyacentes, creando un agujero de lado k , donde k es la valencia del vértice. Este agujero se triangula nuevamente agregando $k-2$ triángulos. Esto permite reducir el número de vértices y el número de triángulos en 1 y 2, respectivamente.

La figura 2.10 muestra el resultado de simplificar una malla de superficie M_i sin suavizar, mientras que la figura 2.11 muestra el resultado de aplicar la misma simplificación a una malla de superficie previamente suavizada. Las figuras 2.10b y 2.10c muestran cómo la forma de las mallas M'_i y M''_i se distorsiona a medida que aumenta la simplificación de la malla inicial M_i cuando esta no ha sido suavizada.

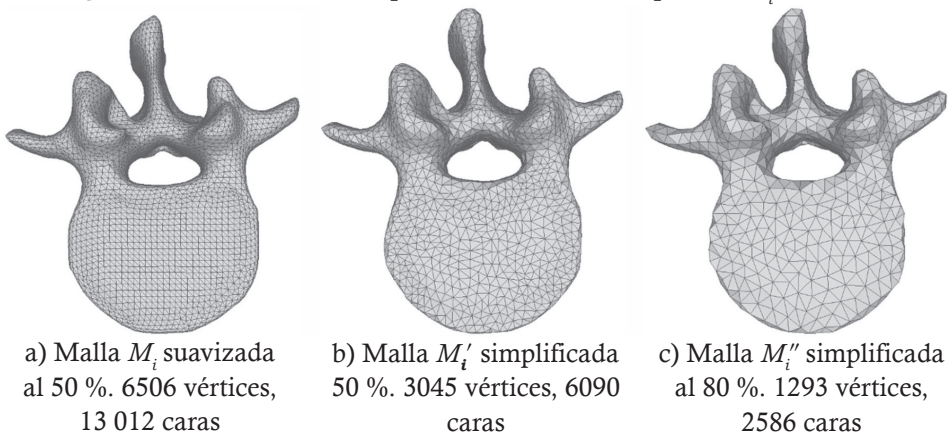
Figura 2.10. Resultado de simplificar una malla de superficie M_i sin suavizar



Fuente: elaboración propia.

Por otro lado, la distorsión producida por la simplificación es menor en las mallas M'_i y M''_i cuando la malla M_i ha sido suavizada previamente, como se muestra en las figuras 2.11b y 2.11c.

Figura 2.11. Resultado de simplificar una malla de superficie M_i suavizada



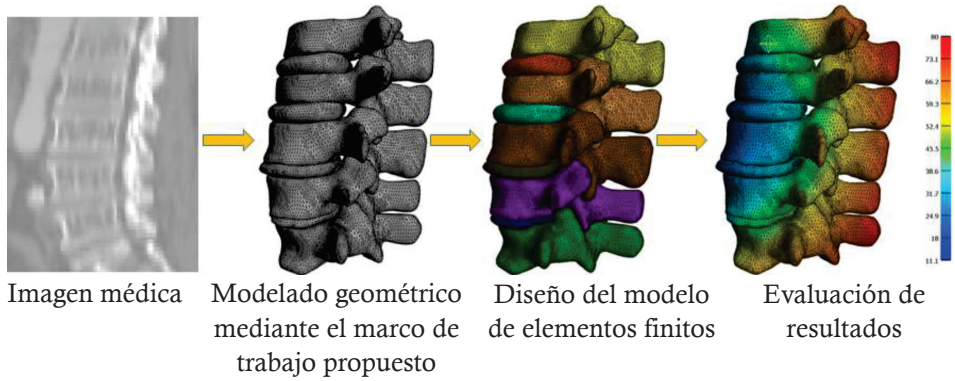
Fuente: elaboración propia.

Capítulo 3. Simulaciones por elementos finitos y resultados

En este capítulo se presenta el diseño y los resultados de cuatro simulaciones por EF de la columna lumbar, creadas para demostrar la capacidad del *framework* propuesto en el capítulo 2, el cual busca manejar geometrías complejas y producir mallas de alta calidad, en especial en aquellos casos cuando los modelos geométricos tienen interfaces de contacto entre regiones dentro de las estructuras anatómicas.

Las simulaciones se realizaron en un computador con un microprocesador Pentiumi7 de 3.40 GHz y 8 gigas de memoria RAM. Las herramientas y bibliotecas que proporcionan una funcionalidad de soporte a esta investigación pertenecen al grupo de tecnologías de *software de código abierto*, por lo que su aplicación está exenta de taras económicas y legales (Pavarino *et al.*, 2013). Entre las más importantes se tiene: VTK (<http://www.vtk.org/>) (Schroeder *et al.*, 1998) para utilizar el algoritmo MC y la extensión vtkbool, 3D Slicer (<http://www.3dslicer.org/>) (Fedorov *et al.*, 2012) para la generación de las mallas de superficie, la simplificación y el suavizado de las mismas, y FEBio Studio (<http://febio.org>) (Maas *et al.*, 2012) para el modelado y simulación por EF y para la visualización de los resultados.

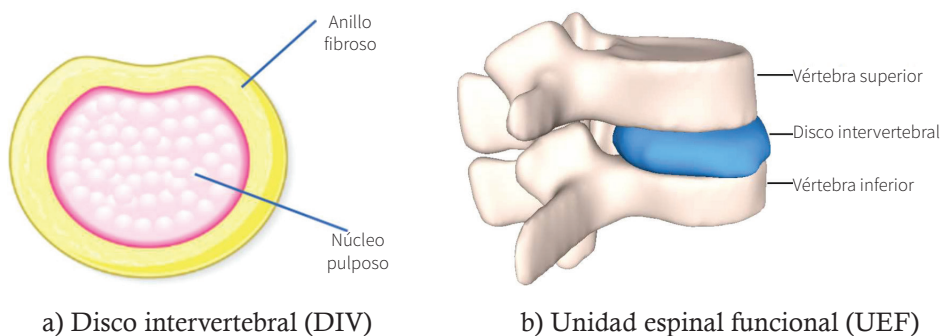
Los modelos geométricos se generaron a partir de una imagen médica segmentada de la columna lumbar, proveída por el Laboratorio de Planeación Quirúrgica de la Universidad de Harvard (<https://spl.harvard.edu/>). La figura 3.1 muestra la metodología que se aplica para el diseño de simulaciones por medio de EF. Esta metodología se aplica en el numeral 3.1 al modelado de estructuras anatómicas simples y en el numeral 3.2, al modelado de estructuras anatómicas compuestas.

Figura 3.1. Metodología para el diseño de simulaciones por EF

Fuente: elaboración propia.

3.1. Modelado de estructuras simples

La columna vertebral es una estructura compleja formada por vértebras y discos intervertebrales (DIV) que actúan juntos para proporcionar una variedad de movimientos y soportar cargas considerables (figura 3.2). Los DIV son estructuras viscoelásticas que soportan las cargas y absorben energía cuando la columna está bajo compresión (Miele *et al.*, 2012). Todos los DIV de la columna tienen estructuras idénticas, aunque varían en tamaño y forma según la zona de la columna donde se encuentren ubicados. La parte exterior de los DIV está rodeada por un anillo fibroso de colágeno y su parte central es el núcleo que tiene un contenido de agua mucho más alto que el anillo, pero también tiene mucho menos colágeno (figura 3.2a).

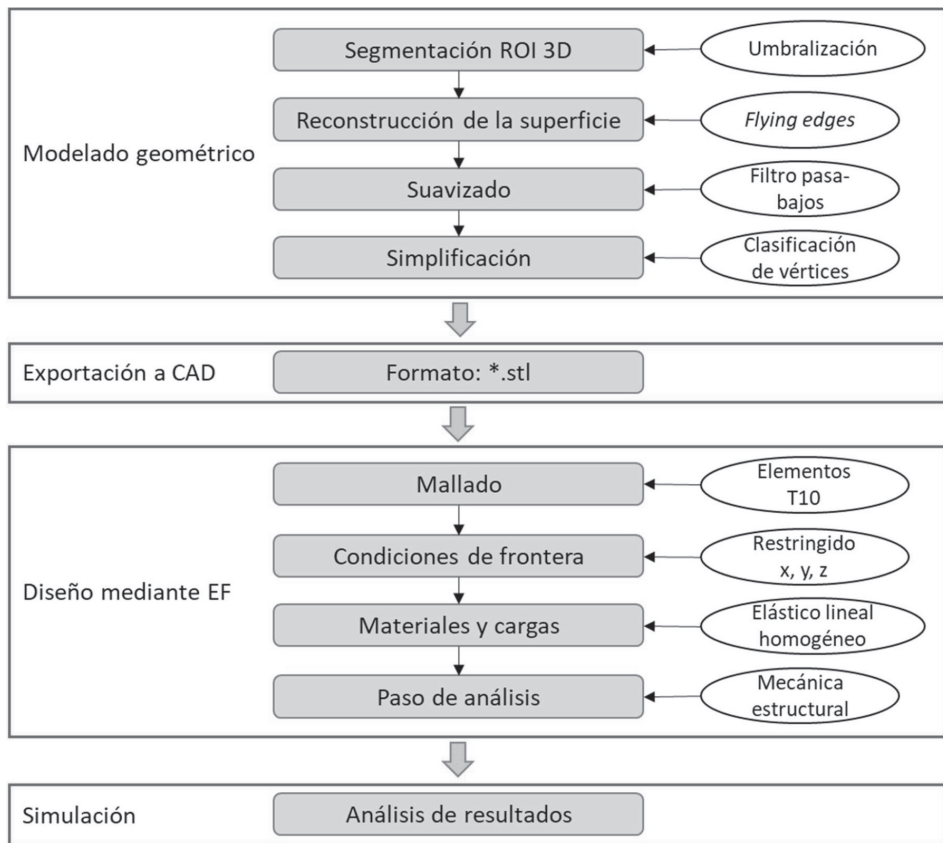
Figura 3.2. Estructura base de la columna vertebral

Fuente: elaboración propia.

Las partes superior e inferior de los DIV se ubican en las placas inferior y superior de cada vértebra adyacente formando una unidad espinal funcional (UEF) (figura 3.2b). La UEF se considera la parte más pequeña de la columna que representa las principales cualidades biomecánicas y las características de toda la columna (Benzel, 2001).

El diagrama de flujo de la figura 3.3 presenta los pasos para el diseño de la simulación por EF de un disco intervertebral.

Figura 3.3. Metodología para el diseño de simulaciones por EF



Fuente: elaboración propia.

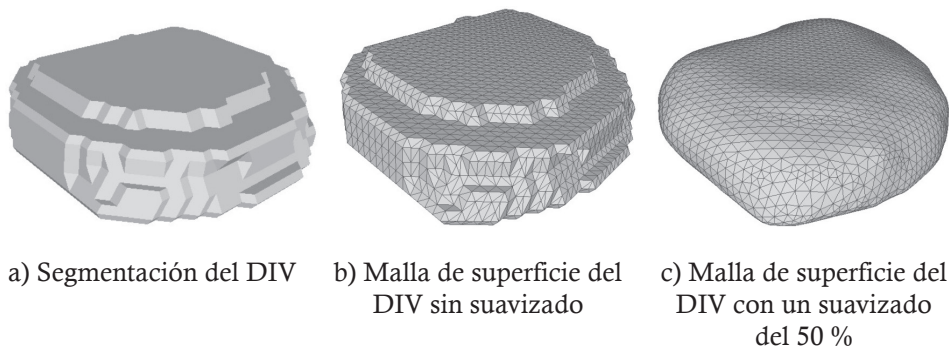
3.1.1. Modelado geométrico

La imagen médica segmentada de la columna lumbar se importó al *software* 3D Slicer, donde se extrajo la región de interés (ROI) del DIV, ubicado entre

las vértebras lumbares L1 y L2 (figura 3.4a), usando una técnica estocástica de segmentación con base en umbrales (numeral 1.1.1). A partir de esta ROI se procedió a generar dos mallas de superficie utilizando el algoritmo *flying edges* (Schroeder *et al.*, 2015). La figura 3.4b muestra la malla sin suavizado, mientras que la figura 3.4c muestra la malla con un suavizado del 50 %. Se puede observar que el efecto de escalón, que aparece en la figura 3.4b, se ha eliminado y en su lugar la malla del DIV presenta una superficie suave y redondeada. Una vez se obtienen las mallas de superficie, se procede a guardarlas en formato *stl*.

Dado que la imagen médica no hace diferencia entre el anillo y el núcleo del DIV, se asume que la estructura anatómica del DIV está compuesta de una sola región Ω_i y, por lo tanto, el modelo geométrico M está compuesto de una única malla de superficie M_1 extraída de dicha región.

Figura 3.4. Modelado geométrico de un DIV



Fuente: elaboración propia.

3.1.2. Diseño del modelo de EF

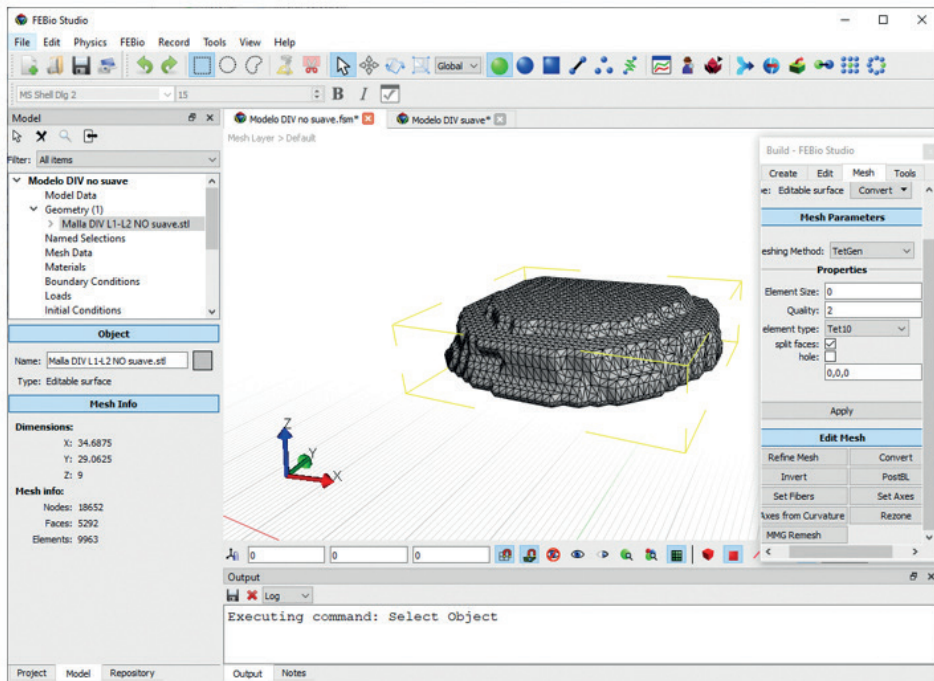
Con base en los dos modelos geométricos M del DIV, obtenidos en la fase de modelado geométrico, se diseñaron dos simulaciones con los mismos parámetros con el fin de comparar los resultados de estas. Para cada simulación se crea un modelo de EF en el *software* FEBio Studio con los nombres: Modelo DIV suave y Modelo DIV no suave. Para cada modelo se importaron los archivos en formato *stl* que contienen los datos de las dos mallas de superficie del DIV M_1 , la suavizada y la no suavizada.

Mallado y tipo de elemento

Dado que la simulación de EF se basa en una representación volumétrica de los elementos, se necesita convertir la representación de la superficie triangular del modelo geométrico M en una representación de volumen tetraédrico. Para esto, en FEBio Studio se selecciona el panel *Mesh* en la ventana de construcción y se escoge el tipo de elemento tetraédricos cuadráticos Tet10 (figura 3.5).

Estos elementos ubican nodos en los vértices y adicionalmente en el medio de las aristas, proporcionando los beneficios de los elementos tetraédricos lineales, pero mejorando la precisión (Finley *et al.*, 2018). El mallado volumétrico de FEBio Studio se basa en el programa TetGen (Si y Gärtner, 2005; Si, 2015).

Figura 3.5. Mallado volumétrico del DIV



Fuente: elaboración propia.

FEBio organiza los tetraedros cuadráticos en un sistema de coordenadas local (llamadas coordenadas isoparamétricas). El vector de posición global x se puede escribir en función de las coordenadas isoparamétricas en el siguiente sentido:

$$x(r, s, t) = \sum_{i=1}^n N_i(r, s, t) x_i \quad (3.1)$$

Aquí, n es el número de nodos, r , s y t son las coordenadas isoparamétricas, N_i son las funciones de forma del elemento y x_i son las coordenadas espaciales de los nodos del elemento tetraédrico.

Las coordenadas del área se relacionan con las coordenadas isoparamétricas de la siguiente manera:

$$\begin{aligned} t_1 &= 1 - r - s - t \\ t_2 &= r \\ t_3 &= s \\ t_4 &= t \end{aligned} \quad (3.2)$$

Y las funciones de forma de la siguiente manera:

$$\begin{aligned} N_i &= t_i(2t_i - 1), i = 1 \dots 4 \\ N_5 &= 4t_1t_2 & N_6 &= 4t_2t_3 \\ N_7 &= 4t_3t_1 & N_8 &= 4t_1t_4 \\ N_9 &= 4t_2t_4 & N_{10} &= 4t_3t_4 \end{aligned} \quad (3.3)$$

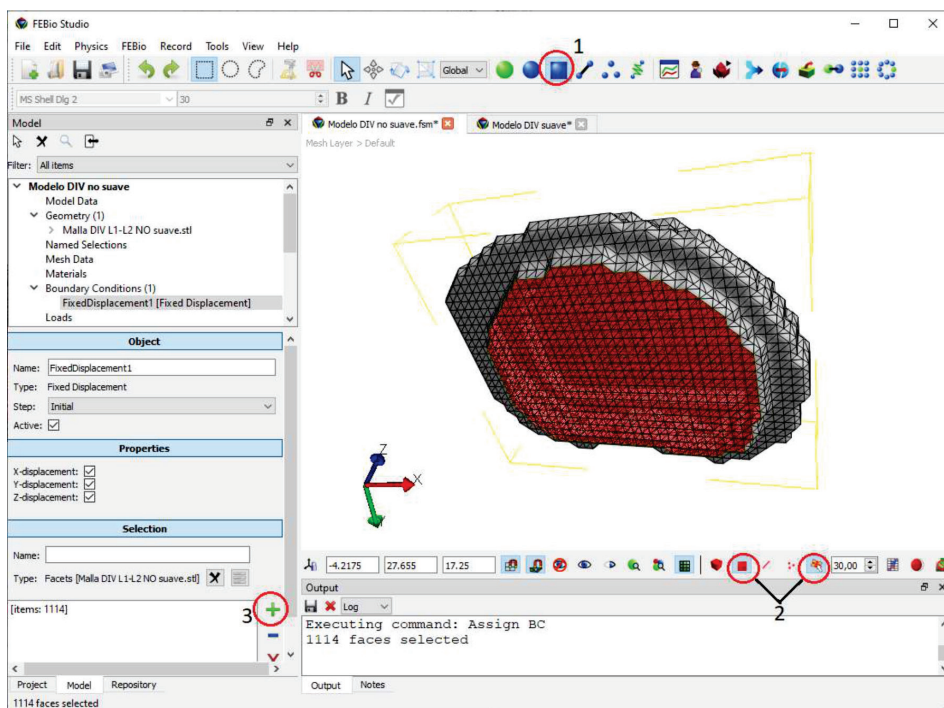
Condiciones de frontera

Las condiciones de frontera son los valores conocidos de los grados de libertad en la frontera que generalmente son desplazamientos y rotaciones.

Para las dos simulaciones se restringieron todos los grados de libertad de la parte inferior del modelo geométrico del DIV. En FEBio Studio se hace clic en el menú *Physics->Add Boundary Condition* y se escoge la opción desplazamiento fijo. En el panel Propiedades se deben marcar todas las casillas de desplazamiento, como se muestra en la figura 3.6, lo que restringe los

grados de libertad en x, y, z. A continuación, se debe aplicar esta restricción a la parte inferior del DIV. Para esto se activa la opción seleccionar caras en la Barra principal de herramientas (botón con el cuadrado azul) marcado con el número 1 en la parte superior de la figura 3.6. Luego, en la Barra de herramientas de gráficos se activan las botones de selección de caras y selección de conectados, marcados con el número 2 en la parte inferior de la figura 3.6. Finalmente, se asignan las caras seleccionadas a la condición de frontera haciendo clic en el botón +, marcado con el número 3 en la parte inferior izquierda de la figura 3.6.

Figura 3.6. Asignación de las condiciones de frontera del DIV



Fuente: elaboración propia.

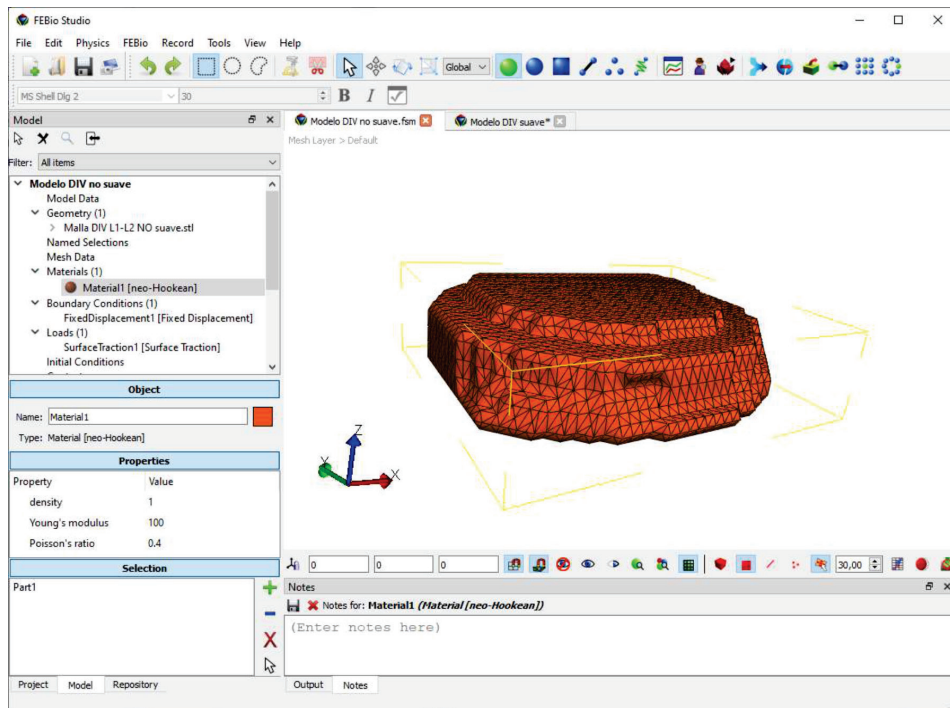
Asignación de materiales y cargas

Al DIV se le configuró un material elástico lineal homogéneo, que representa el comportamiento de los componentes combinados del núcleo y el anillo del DIV, con un módulo elástico de 100 MPa y un valor de relación de Poisson de 0,4 (Yamamoto *et al.*, 1989; Lavaste *et al.*, 1992).

Para esto se hace clic en el menú *Physics->Add Material* y se escoge el material neo-Hookean asignando en la ventana de propiedades, ubicada en el lado izquierdo de la figura 3.7, los valores ya mencionados. Esta misma configuración de material es asignada al dominio de la simulación del DIV suavizado. La figura 3.7 muestra el DIV con el color del material asignado y la malla de superficie del DIV visible.

La configuración de las cargas de los dos modelos se hace mediante el menú *Physics->Add Surface Load*, asignando una carga aproximada de 450 newtons a la parte superior del DIV. Esta carga equivale al peso corporal de una persona de 70 kg.

Figura 3.7. Asignación del material elástico lineal al DIV



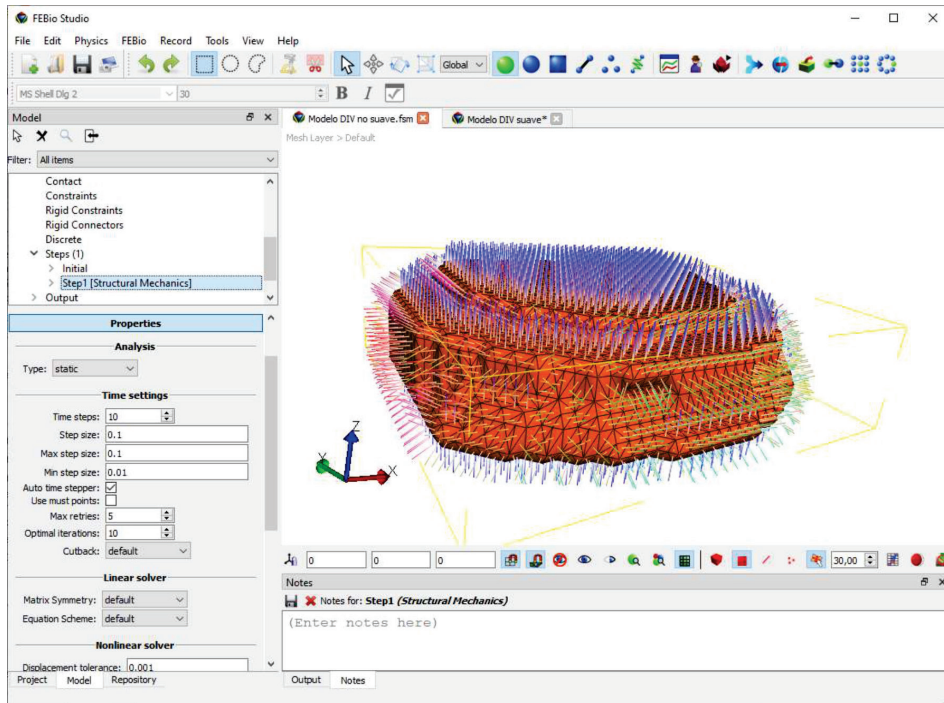
Fuente: elaboración propia.

Paso de análisis

Finalmente, se necesita crear un *paso de análisis* para definir qué tipo de análisis se quiere hacer. En este caso se hace clic en el menú *Physics->Add Analysis*

Step y se escoge el análisis de Mecánica estructural. Las propiedades y los valores del paso de análisis se muestran al lado izquierdo de la figura 3.8.

Figura 3.8. Configuración del paso de análisis del DIV



Fuente: elaboración propia.

Se puede observar que aparecen 2 pasos de análisis en la lista: el de Mecánica estructural que se acaba de crear y uno llamado Inicial. El paso inicial no es realmente un paso de análisis, ya que no se puede establecer la configuración del paso de tiempo para el paso inicial. Este se utiliza para incluir las condiciones de frontera permanentes, es decir, las condiciones de frontera que se propagarán a través de todos los pasos del análisis.

El modelo final de EF se muestra en la figura 3.8, donde, además de la malla de superficie, se activó la visualización de las normales de cada triángulo de la malla. Una vez terminado el diseño del modelo de EF se procede a guardar el modelo en un archivo con extensión *fsm*.

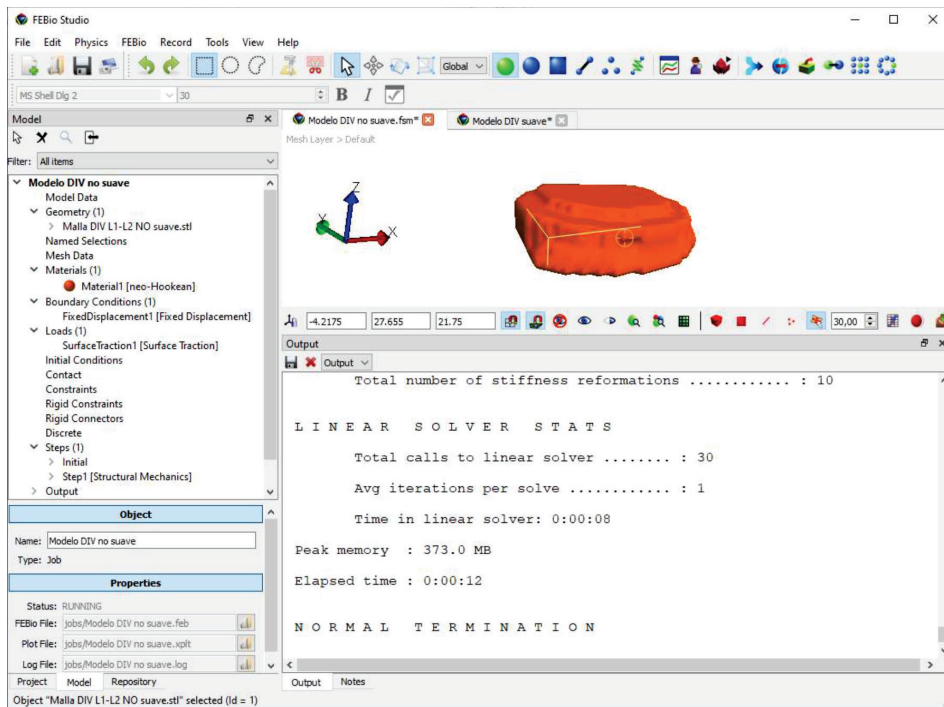
3.1.3. Resultados

Simulación

Para ejecutar la simulación del modelo se puede acceder desde el entorno gráfico al procesador en el menú *FEBio->Run FEBio*. Sin embargo, se debe tener en cuenta que la simulación es un proceso que se realiza fuera del entorno gráfico.

Antes de comenzar la simulación, el procesador comprueba el archivo en formato *fsm* en busca de errores. La figura 3.9 muestra la terminación exitosa de la simulación en la ventana de información ubicada debajo del entorno gráfico.

Figura 3.9. Información del avance en la ejecución de la simulación



Fuente: elaboración propia.

Para hacer los cálculos de la simulación, todos los elementos tetraédricos del modelo de EF se integran numéricamente. Esto implica que las integrales sobre el volumen del elemento v^e se aproximan mediante una suma:

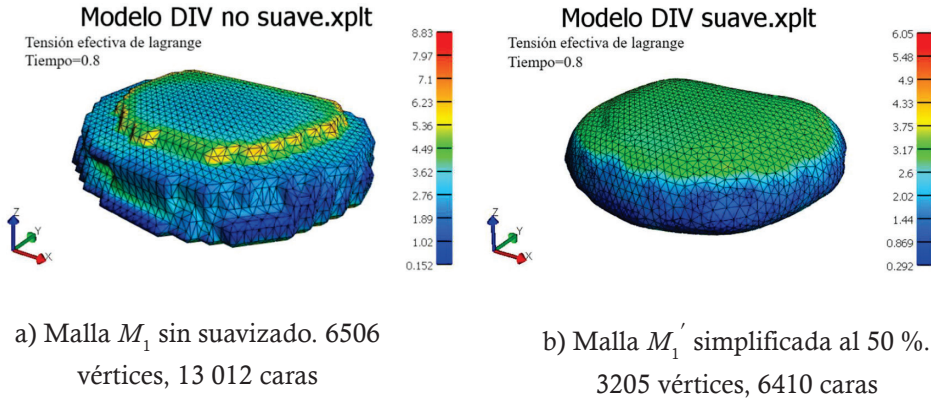
$$\int_{\square} f(x) dv = \int_{\square_e} f(r) J(r) d\square \cong \sum_{i=1}^m f(r_i) J_i w_i \quad (3.4)$$

Aquí, $\square$ es el cubo biunit, m es el número de puntos de integración, r_i son la ubicación de los puntos de integración en coordenadas isoparamétricas, J es el jacobiano de la transformación $x = x(r, s)$ y w_i es un peso asociado con el punto de integración, la cual se realiza sobre el volumen del elemento en el sistema de coordenadas naturales.

Visualización

La figura 3.10 muestra el resultado de las dos simulaciones. En la figura 3.10a se puede observar el estrés efectivo para el tiempo 0,8 con un valor de 8,83 MPa, llegando a 11,0 MPa en el tiempo 1 cuando termina la simulación. Para el caso del DIV suave el estrés efectivo es de 6,05 MPa en el tiempo 0,8 (figura 3.10b), alcanzando un valor de 7,56 MPa en el tiempo 1 al finalizar la simulación.

Figura 3.10. Resultados de la simulación del DIV



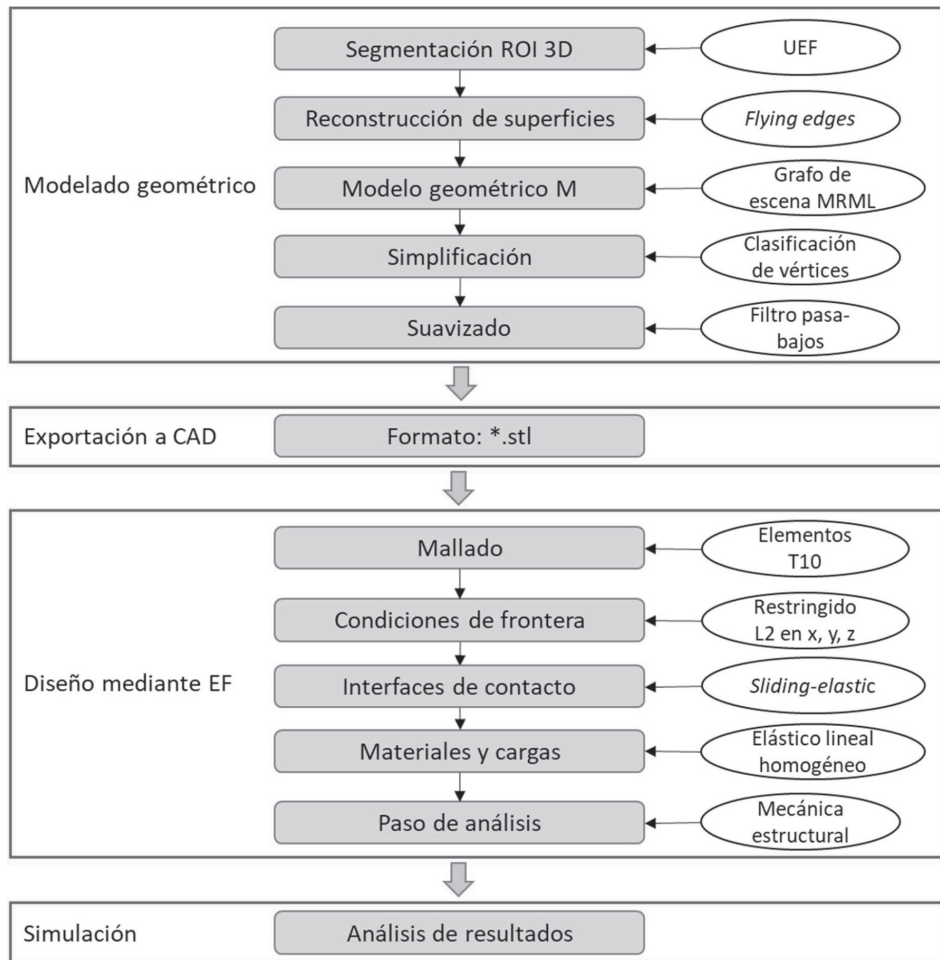
Fuente: elaboración propia.

3.2. Modelado de estructuras compuestas

El diseño de las simulaciones por EF del numeral 3.1 y los respectivos resultados han demostrado la capacidad del *framework*, propuesto en el capítulo 2, para generar modelos geométricos aptos para el análisis por EF de

estructuras anatómicas simples. En este numeral se presenta el modelado de la UEF L1-L2 de la columna lumbar (figura 3.2b). El diagrama de flujo de la figura 3.11 presenta los pasos para el diseño de la simulación por EF.

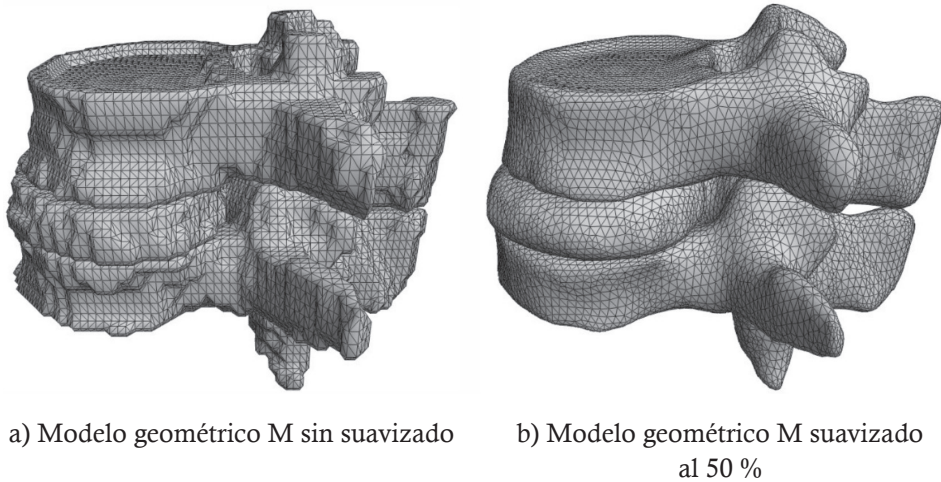
Figura 3.11. Metodología para el diseño de simulaciones por EF



Fuente: elaboración propia.

3.2.1. Modelado geométrico

Al igual que en el numeral 3.1, la imagen médica segmentada de la columna lumbar se importó al *software* 3D Slicer. A partir de la imagen médica se extrajeron tres regiones de interés Ω_i asociadas a la UEF L1-L2: las dos vértebras adyacentes L1 y L2 y el DIV L1-L2 (figura 3.12).

Figura 3.12. Modelado geométrico de la UEF L1-L2

Fuente: elaboración propia.

Con base en estas regiones se generaron las mallas de superficie M_i y el modelo geométrico M, observando que las mallas M_i sean cerradas y que no se traslapen entre sí. En la figura 3.12a se puede observar el modelo M sin suavizado, mientras que la figura 3.12b muestra el modelo M final con un suavizado del 50 %. Una vez se obtiene el modelo geométrico M se procede a guardarlo en formato *stl*.

3.2.2. Diseño del modelo de EF

Los archivos en formato *stl*, obtenidos en la fase de modelado geométrico, son importados al *software* FEBio Studio para diseñar dos simulaciones, una con los archivos del modelo geométrico sin suavizar y otra con los archivos del modelo suavizado. Para esto se crean dos modelos de EF: Modelo UEF no suave y Modelo UEF suave, que se guardan en formato *fsm*. Los parámetros de configuración de estos modelos son los mismos con el fin de comparar los resultados de las dos simulaciones (figura 3.13).

Mallado y tipo de elemento

El mallado de todo el dominio de simulación se realizó utilizando elementos tetraédricos cuadráticos T10. Como ya se ha mencionado, esto proporciona

los beneficios de los elementos tetraédricos lineales, pero mejorando la precisión (Finley *et al.*, 2018). El número de nodos, caras y elementos de cada subdominio del modelo se muestra en la tabla 3.1.

Tabla 3.1. Resultados del proceso de mallado de la UEF

Tipo de UEF	Región	N.º nodos	N.º caras	N.º tetraedros
No suave	Vértebra L1	46 921	13 012	25 471
	Vértebra L2	53 504	14 748	29 136
	DIV	18 652	5292	9963
Suave	Vértebra L1	7611	13 012	25 198
	Vértebra L2	8630	14 748	28 708
	DIV	3090	5292	9910

Fuente: elaboración propia.

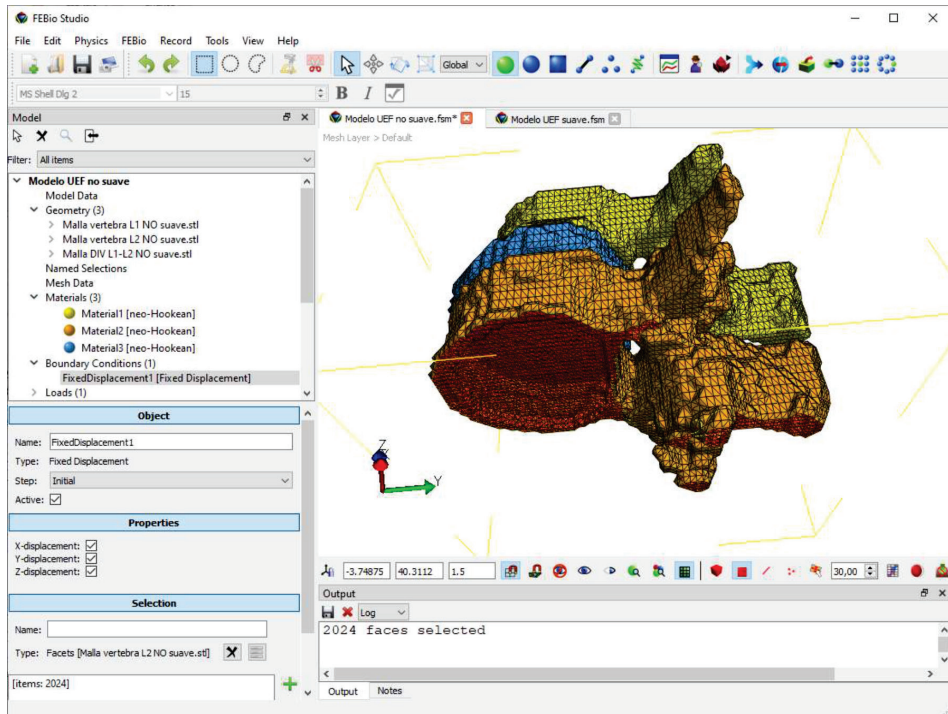
Asignación de materiales y cargas

Las vértebras están compuestas por 2 tipos de hueso llamados cortical y esponjoso. Sin embargo, las vértebras se modelaron como sólidos con un material elástico lineal homogéneo isotrópico. Esto permitió representar el comportamiento de los componentes combinados del hueso cortical y del hueso esponjoso con un solo material (figura 3.13). El módulo elástico para las vértebras fue de 12 000 MPa y la relación de Poisson fue de 0,3 (Yamamoto *et al.*, 1989; Lavaste *et al.*, 1992). El subdominio correspondiente al DIV se modeló como se muestra en el numeral 3.1.2.

Para la configuración de las cargas, el cuerpo vertebral superior y las dos apófisis articulares superiores recibieron aproximadamente 450 newtons, que equivale al peso corporal de una persona de 70 kg (Nabhani y Wake, 2002).

Condiciones de frontera

Para las dos simulaciones, todos los grados de libertad de la parte inferior del cuerpo vertebral L2 y las dos apófisis articulares inferiores fueron restringidas, como se muestra en la figura 3.13.

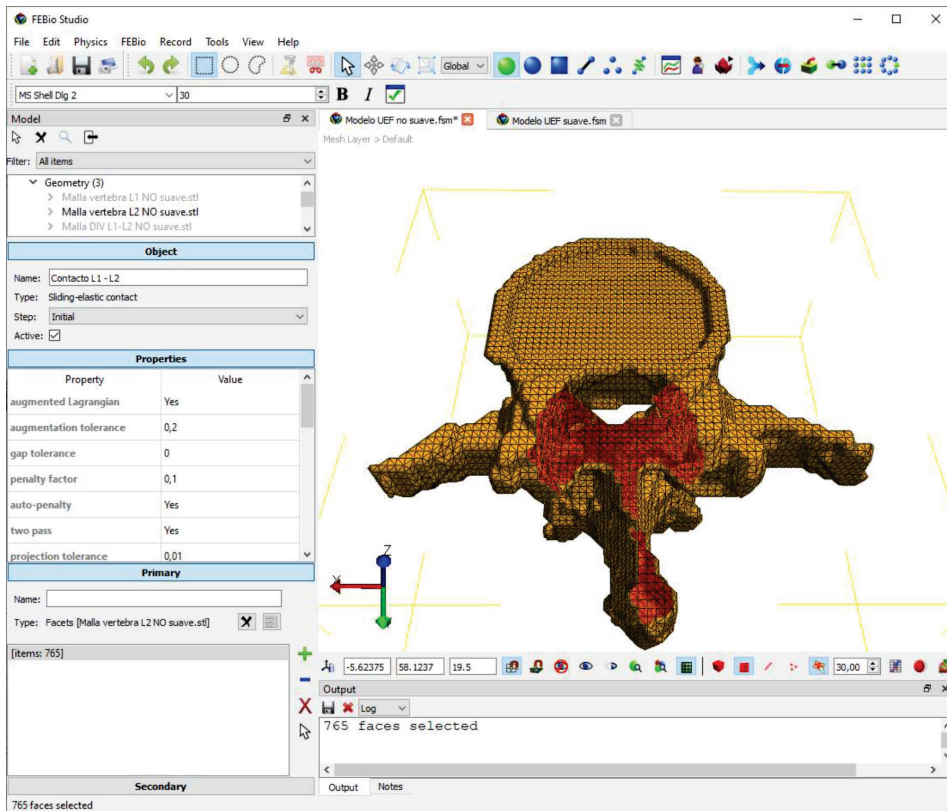
Figura 3.13. Asignación del material en el *software* de diseño FEBio

Fuente: elaboración propia.

Configuración de las interfaces de contacto

El *software* FEBio Studio permite conocer la interacción o contacto entre las diferentes regiones que componen las estructuras anatómicas mediante *interfaces de contacto*. En estas simulaciones se utilizaron *sliding-elastic*, que son interfaces de contacto que se pueden configurar tanto con fricción como sin fricción entre dos superficies. Las superficies de contacto deben extenderse lo suficiente para incluir todas las partes que pueden entrar en contacto durante el proceso de simulación. Cada interfaz *sliding-elastic* consta de una superficie maestra (*Primary*) y una superficie esclava (*Secondary*) (figura 3.14). La superficie *Primary* permite medir la función de separación y definir las cantidades cinemáticas como normales de superficie y tangentes. Sobre la superficie *Secondary* se integran las ecuaciones de contacto y se calculan las fuerzas de tracción del contacto.

Figura 3.14. Configuración de las interfaces de contacto



Fuente: elaboración propia.

Se debe tener en cuenta que las dos superficies no pueden incluir los mismos nodos y que se debe elegir la superficie esclava y la maestra, de acuerdo con las siguientes pautas:

- La mayor de las dos superficies debe actuar como superficie maestra.
- Si las superficies son de tamaño similar, la superficie del cuerpo más rígido debe actuar como superficie maestra.
- Si las superficies son de tamaño y rigidez similar, la superficie con la malla menos densa debe actuar como superficie maestra.
- Si una de las superficies de contacto es rígida, es mejor elegir esta superficie como superficie maestra.

De acuerdo con estas pautas se configuraron tres interfaces de contacto: Contacto L1-L2, Contacto L1-DIV y Contacto L2-DIV. Para esto se hace clic en el menú *Physics->Add Contact* y se escoge la opción *sliding elastic*.

En la ventana de propiedades se activan las opciones de *augmented Lagrangian*, *autopenalty* y *two-pass*. Luego se agregan las caras que harán parte del contacto primario y secundario haciendo clic en el botón . Para esto se debe dejar visible solo la geometría de la región de la que se van a seleccionar las caras, ocultando las regiones restantes como se muestra en la figura 3.14.

Paso de análisis

Finalmente, se crea el paso de análisis de Mecánica estructural. Las propiedades y los valores del paso de análisis se muestran en la ventana del visor de modelos de la figura 3.15. Con la configuración del paso de análisis se termina el diseño del modelo de EF.

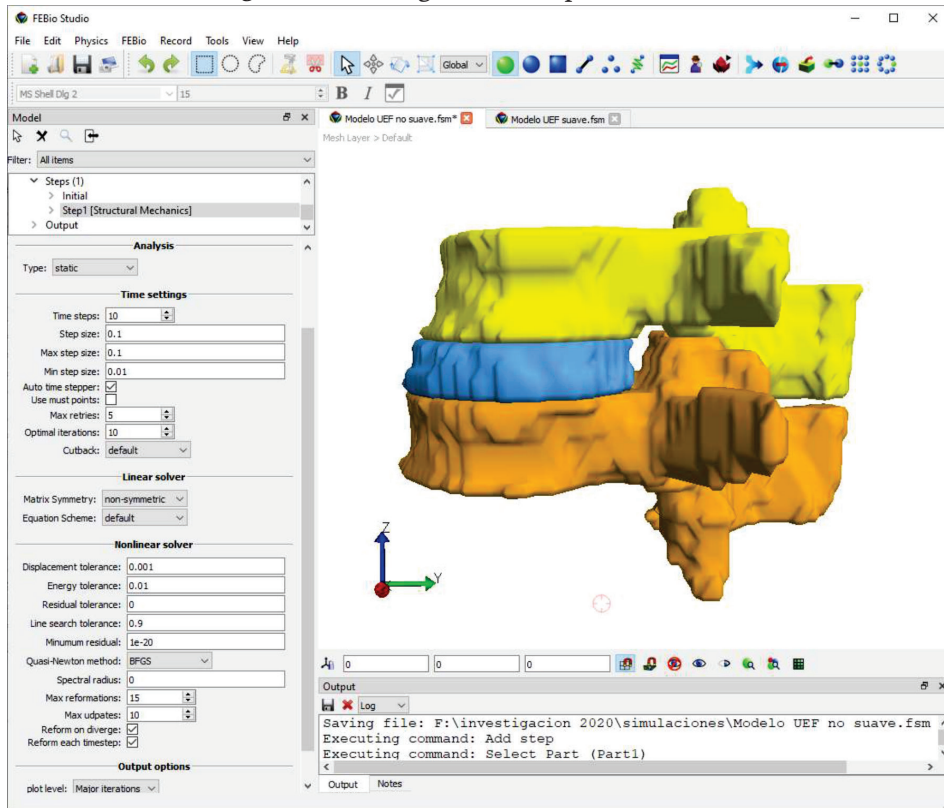
3.2.3. Resultados

Simulación

En el panel de registro ubicado en la parte inferior derecha de la figura 3.16 se muestra el estado del proceso de simulación y algunas estadísticas. El procesador FEBio a menudo tiene dificultades cuando las estructuras anatómicas tienen regiones con mucha separación inicial. Esto es así cuando el análisis es impulsado por la fuerza, lo que significa que la deformación del modelo es impulsada por una fuerza (a diferencia de los análisis impulsados por el desplazamiento). En este caso, es mejor evitar la separación inicial, si es posible.

Otra forma de solucionar este problema es usar primero un desplazamiento para poner las superficies en contacto y luego reemplazar el desplazamiento con una fuerza.

Figura 3.15. Configuración del paso de análisis



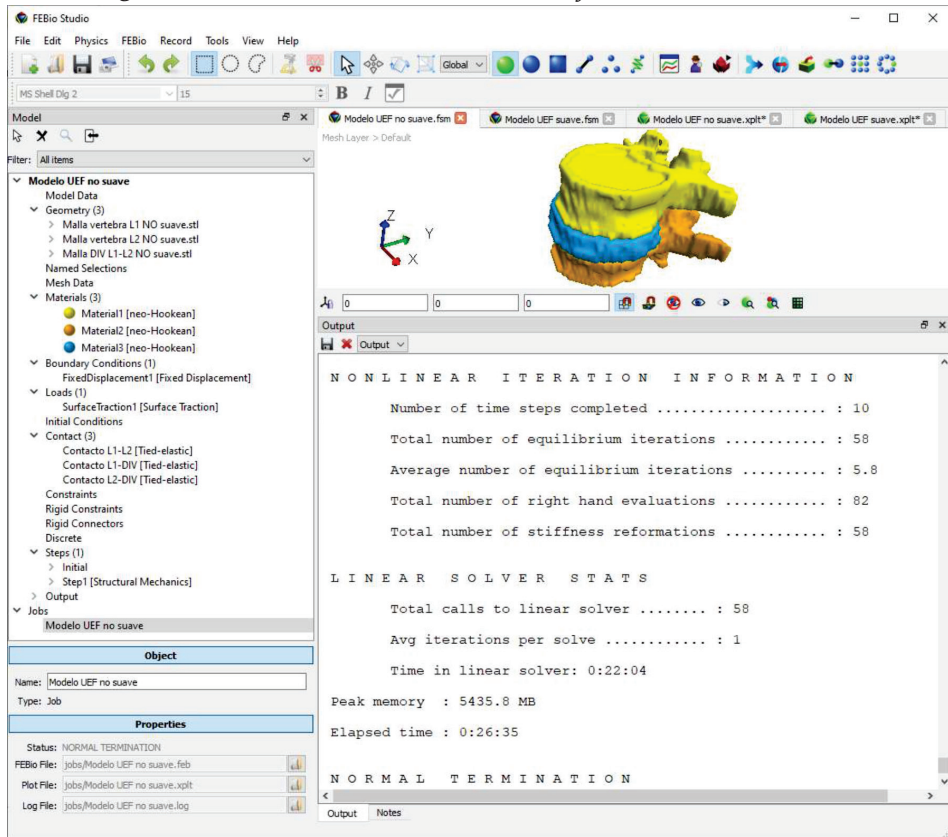
Fuente: elaboración propia.

Visualización

La figura 3.17 muestra la tensión efectiva de lagrange de la UEF no suave (figura 3.17a) y de la UEF suave (figura 3.17b) compuestas respectivamente por las vértebras L1 y L2 y del DIV L1-L2.

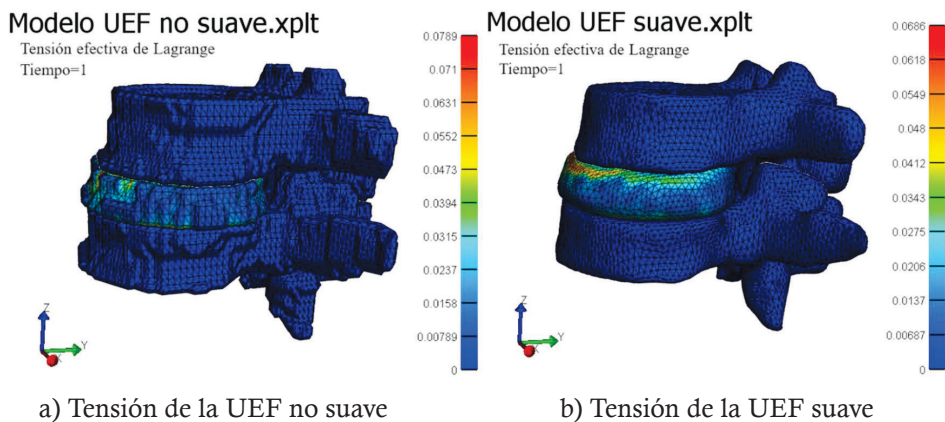
La tensión del DIV tiene dos causas principalmente: el contacto entre el DIV y la vértebra L1 y del contacto entre el DIV y la vértebra L2. La figura 3.17a muestra la UEF no suave con una tensión máxima de 0,079 Mpa al finalizar la simulación, mientras que la figura 3.17b muestra una tensión máxima de 0,068 Mpa para la UEF suave. Tal como se demuestra en Caínzos López (2017), las tensiones máximas para las dos vértebras se producen en la porción anteroposterior del cuerpo vertebral próxima a los pedículos, mientras que en el disco se localizan en la zona exterior del anillo fibroso (figura 3.18).

Figura 3.16. Información del avance en la ejecución de la simulación



Fuente: elaboración propia.

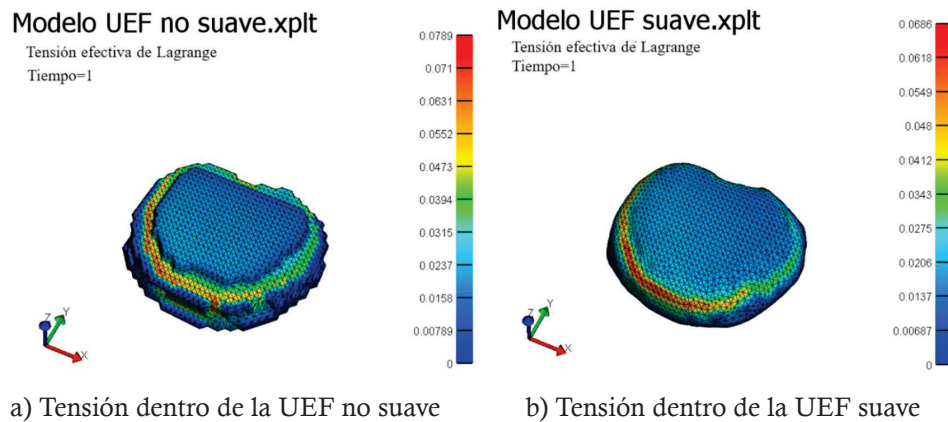
Figura 3.17. Distribución de la tensión efectiva de Lagrange de la UEF



Fuente: elaboración propia.

Aunque la configuración de las dos simulaciones es igual, la tensión de los contactos del DIV de la UEF no suave es mayor (figura 3.18a) que la tensión del DIV de la UEF suave (figura 3.18b), debido a que los contactos de este último se distribuyen más uniformemente.

Figura 3.18. Distribución de la tensión efectiva de Lagrange del DIV



Fuente: elaboración propia.

Como ya se ha mencionado, el disco intervertebral soporta y distribuye las cargas a las que se somete la columna. Esto se evidencia en el desplazamiento máximo que experimenta y que se marca en la porción anteroposterior de la UEF, donde el desplazamiento de las vértebras coincide con el desplazamiento del disco intervertebral.

3.3. Calidad de las mallas de superficie

Para simulaciones por EF es esencial que todos los triángulos del modelo geométrico M tengan al menos una calidad mínima, ya que de lo contrario la malla de volumen, derivada de la malla de la superficie, también tendrá elementos de baja calidad y la simulación numérica podría no converger. Si el tamaño de los elementos varía para ajustarse a las propiedades locales de la región de interés, igualmente se deben garantizar transiciones graduales en el tamaño de los elementos adyacentes.

Se pueden definir varios índices para indicar qué tan buena es la calidad de un triángulo. En el numeral 1.4 se ofrece una descripción general de estos.

La mayoría se basan en el supuesto de que el triángulo de mejor calidad es el equilátero. Dado que la selección de una métrica de evaluación de calidad es crucial, aquí se ha adoptado un índice de calidad tomando como base la relación de aspecto del triángulo, denotado por Q , de la forma:

$$Q = \frac{R}{2r}, 0 \leq Q \leq 1 \quad (3.5)$$

donde R y r denotan el *circumradius* y el *inradius* del triángulo, respectivamente (Brandts *et al.*, 2008). El índice se ha normalizado para que sea $Q = 1$ para un triángulo equilátero. Los valores de Q cercanos a cero indican triángulos de mala calidad con el caso límite $Q = 0$ correspondiente a un triángulo de área cero. El índice de calidad Q se puede evaluar mediante:

$$Q = \frac{abc}{8(p-a)(p-b)(p-c)}, p = \frac{a+b+c}{2} \quad (3.6)$$

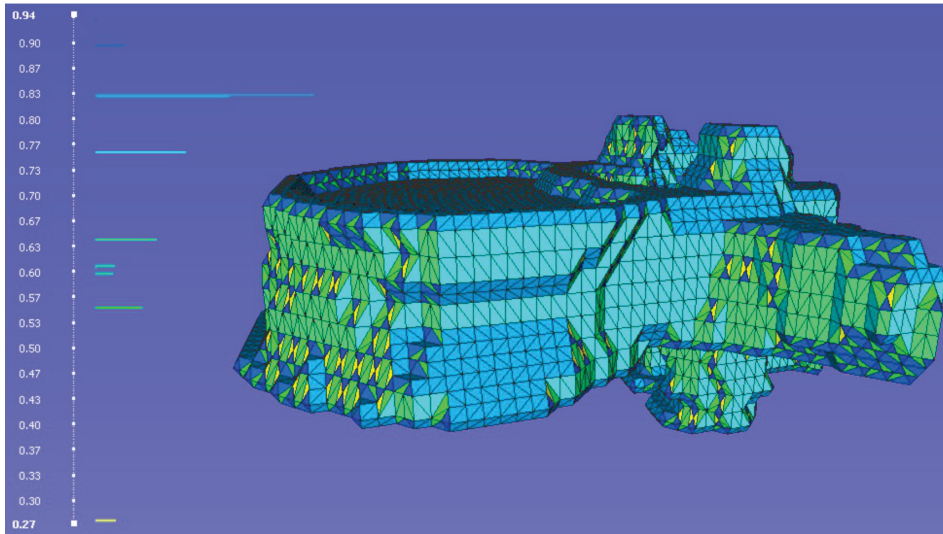
donde a , b y c representan las longitudes de los lados del triángulo.

En la figura 3.19 se compara la calidad de la malla de triángulos inicial generada por el algoritmo *Marching cubes* (Lorensen y Cline, 1987) y la malla suavizada mediante el algoritmo FE (Schroeder *et al.*, 2015) de una vértebra lumbar L1. Se puede observar en la figura 3.19a que, aunque visualmente los triángulos generados por el algoritmo MC parecen tener una buena forma, la relación de aspecto de los mismos demuestra que la calidad de estos no es tan buena y que la distribución de los valores de Q no es uniforme.

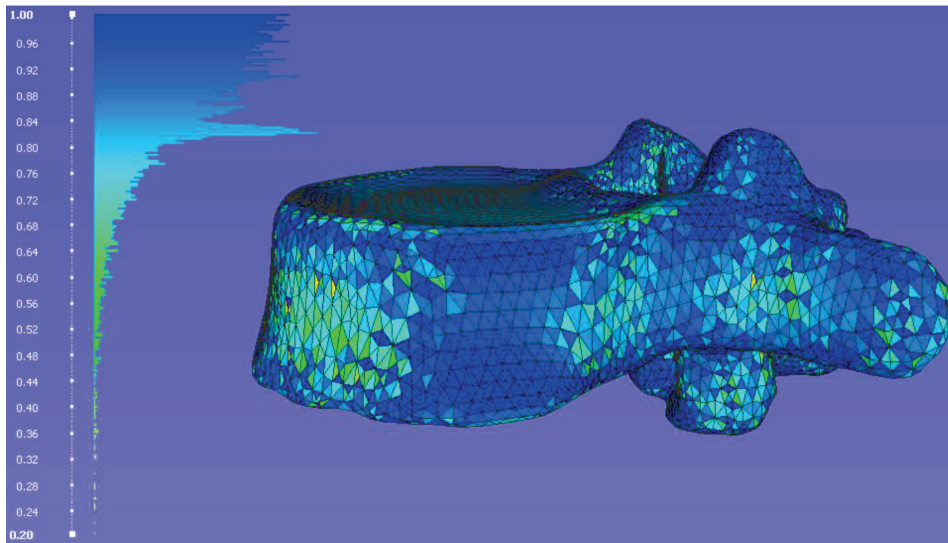
Por otro lado, la malla de triángulos generada por el *framework* propuesto muestra claramente un cambio hacia una distribución uniforme de los valores de Q , la mayoría con $Q > 0,6$, que evidencia una excelente distribución de la calidad de los triángulos de la malla, como se puede observar en la figura 3.19b.

Esta mejora en la calidad de las mallas se ve reflejada en las estadísticas relacionadas con las diferentes simulaciones por EF realizadas en este capítulo.

Figura 3.19. Comparación de la calidad de las mallas de superficie de la vértebra L1



a) Malla generada por el algoritmo MC



b) Malla suavizada con el algoritmo FE

Fuente: elaboración propia.

En la tabla 3.2 se muestran algunas estadísticas de las dos simulaciones diseñadas para el DIV. Se puede observar que, aunque el número de tetraedros generados a partir de las mallas de superficie es casi el mismo, no es así el

número de nodos de EF generados por FEBio, como tampoco la memoria requerida y el tiempo de simulación que para el caso del DIV no suave es casi diez veces mayor. Estas estadísticas y otras más relacionadas con el proceso de simulación FEBio las guarda en un archivo con extensión *.log*.

Tabla 3.2. Estadísticas de las simulaciones del DIV

Estadística	DIV no suave	DIV suave
N.º de nodos	18 652	3090
N.º de tetraedros	9963	9910
Memoria pico	374 MB	59,1 MB
Tiempo transcurrido	12,2 segundos	1,3 segundos

Fuente: elaboración propia.

En la tabla 3.3 se muestran las mismas estadísticas, pero relacionadas con las dos simulaciones diseñadas para la UEF. Se puede evidenciar que el tiempo de simulación para la UEF con las mallas suavizadas es de tan solo 1,42 segundos, para la UEF con las mallas generadas por el algoritmo MC es 26,37 minutos. Este tiempo está igualmente relacionado con el número de nodos y la memoria necesaria para llevar a cabo las simulaciones que, como se observa en la tabla 3.3 es mucho mayor para UEF con las mallas no suaves.

Tabla 3.3. Estadísticas del proceso de simulación de la UEF

Estadística	DIV no suave	DIV suave
N.º de dominios	3	3
N.º de nodos	119 077	19 331
N.º de tetraedros	64 570	36 816
Memoria pico	5435,8 MB	565,9 MB
Tiempo transcurrido	26,37 minutos	1,42 minutos

Fuente: elaboración propia.

Se podría pensar que la mejora en la calidad de las mallas, propuesta en este *framework*, mejora los tiempos de la convergencia de las simulaciones a expensas de una reducción en el volumen de las mismas. Para mostrar que la mejora de las mallas causa poca pérdida de resolución espacial, se hace una comparación entre la región de interés segmentada Ω_i y la malla

de superficie mejorada M_i de dicha región. La técnica de comparación más utilizada es el coeficiente de similitud de Dice (Dice, 1945); sin embargo, aquí no es aplicable debido a la desalineación geométrica que pueden presentar la región y la malla de superficie, ya que el coeficiente de Dice es altamente dependiente de la forma. Otra técnica de comparación es la distancia de Hausdorff, que al ser una métrica invariante de la forma es utilizada frecuentemente para comparar la superposición espacial de dos objetos (Huttenlocher *et al.*, 1993).

La distancia de Hausdorff ($D_{Hausdorff}$) mide la diferencia entre dos modelos geométricos en términos de las distancias entre muestras de puntos en las superficies de ambos modelos. Dados dos conjuntos de puntos finitos $A = \{a_1, \dots, a_n\}$ y $B = \{b_1, \dots, b_m\}$, para cada punto en la superficie A la distancia a la superficie B se define como la distancia euclidiana al punto más cercano en la superficie B :

$$D_{Hausdorff}(A, B) = \max(d(A, B), d(B, A)) \quad (3.7)$$

donde $d(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$ y $\|\cdot\|$ es la distancia entre los puntos a y b medida por alguna norma (L_1, L_2, L_∞).

Los valores de la distancia de Hausdorff van desde 0 (cero), cuando hay una total coincidencia entre los dos modelos, hasta cualquier valor que indica la distancia de un punto de A al punto más cercano de B . Es importante observar que esta distancia es en general no simétrica, es decir, $d(A, B) \neq d(B, A)$.

Para los cálculos se toma la malla de superficie sin suavizar (A) y la malla de superficie suavizada (B). La tabla 3.4 muestra los resultados de calcular la distancia de Hausdorff, utilizando el *software* MeshLab (www.meshlab.net). Esta distancia es inferior a 0,89 y 0,99 para las regiones de las vértebras L1 y L2, e inferior a 0,82 para la región del DIV. Adicionalmente, se puede observar que la media de la distancia de Hausdorff está en un promedio de 0,24 para todas las mallas de superficie M_i .

Tabla 3.4. Distancia de Hausdorff

Región de interés	$d(A,B)$	media
Vértebra L1	0,88	0,23
Vértebra L2	0,99	0,23
Disco intervertebral L1-L2	0,82	0,23

Fuente: elaboración propia.

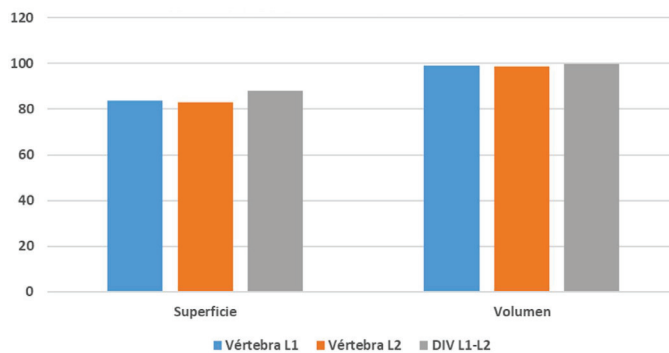
En la tabla 3.5 se listan las estadísticas relacionadas con el número de vértices y triángulos de las mallas de superficie, así como el área y el volumen de cada una de ellas.

Tabla 3.5. Estadísticas de las mallas de superficie

Región de interés	Malla sin suavizar				Malla suavizada			
	Puntos	Triángulos	Área (mm ²)	Volumen (mm ³)	Puntos	Triángulos	Área (mm ²)	Volumen (mm ³)
Vértebra L1	6506	13 012	5880,1	12 165,4	6506	13 012	4914,7	12 044
Vértebra L2	7374	14 748	6648,9	13 109,5	7374	14 748	5518	12 951,3
Disco L1-L2	2648	5292	2323,6	6022,4	2648	5292	2048	6017,1

Fuente: elaboración propia.

Otra característica que buscan los modelos 3D es la conservación de la superficie y del volumen. En la Figura 3.20 se puede observar que el porcentaje de conservación de la superficie de las diferentes mallas está entre el 83 % y el 88 %, mientras que la conservación del volumen es casi del 100 %.

Figura 3.20. Porcentaje de conservación de la superficie y el volumen

Fuente: elaboración propia.

Conclusiones

En este libro se realizaron varias actividades con el fin de obtener un *framework* (marco de trabajo) para generar mallas de superficie de alta calidad a partir de imágenes médicas que facilite la investigación de estructuras anatómicas y la aplicación práctica de las simulaciones por EF a las mismas. Todos los métodos y algoritmos integrados en el *framework* se estudiaron en el capítulo 1 y para su implementación se utilizó la plataforma de visualización y análisis de imágenes médicas 3D Slicer (www.slicer.org) (Fedorov *et al.*, 2012). Para el diseño, ejecución y visualización de las simulaciones por EF se utilizó el *software* FEBio Studio (<http://febio.org>) (Maas *et al.*, 2016). De acuerdo con los resultados presentados en el capítulo 3 se pueden extraer las siguientes conclusiones:

Para asegurarse de que la simulación por EF se pueda resolver de manera eficiente, las mallas de superficie M_i , que representan la geometría de las regiones Ω_i que conforman la estructura anatómica Ω , deben cumplir tres requisitos mínimos: una buena calidad de los triángulos o tetraedros, un número apropiado de estos para representar las características tanto de la superficie como del volumen, y un cambio gradual en el tamaño de los mismos. El *framework* propuesto en el capítulo 2 permite obtener mallas de superficie M_i que satisfacen las necesidades geométricas ya mencionadas. Las mallas de superficie contienen un alto porcentaje de triángulos de buena calidad, de tamaño uniforme y adaptados a la curvatura de la estructura anatómica. Así mismo, cuando la estructura anatómica está compuesta de varias regiones el modelo geométrico M , obtenido mediante el *framework* propuesto, garantiza que las interfaces de las regiones no se traslapen entre

sí, lo que ayuda a evitar, por un lado, que haya triángulos invertidos y, por otro, ayuda a la convergencia de la simulación por EF.

El modelo geométrico M es utilizado por el procesador de EF para calcular la solución de la simulación, la cual generalmente es independiente de la resolución de la malla. Sin embargo, para que la solución esté lo más cerca posible de la solución exacta es necesario un estudio de convergencia. Esto significa que la misma simulación debe ejecutarse 80 veces con diferentes resoluciones de malla (más fina o más gruesa dependiendo del caso) (Gómez-Mora *et al.*, 2021). En el capítulo 3 se diseñaron dos simulaciones con los mismos parámetros, pero con diferente suavizado y resolución de malla, por cada estructura anatómica analizada. Los modelos geométricos obtenidos con el *framework* propuesto en el capítulo 2 garantizaron la convergencia de las simulaciones, aunque el tiempo y la memoria necesaria para cada simulación fueron diferentes como debería esperarse.

Se considera importante trabajar más en el diseño de modelos de EF propios ya que, por ejemplo, la definición del material para una UEF, especialmente la del DIV que se conoce es anisotrópico, no coincide en su totalidad con los parámetros del material elástico lineal homogéneo utilizado en las simulaciones. Por otro lado, el haber utilizado una carga de fuerza para simular un proceso estático en lugar de un proceso dinámico repercute en la respuesta mecánica de la UEF para replicar de la mejor manera el trabajo diario al que esta podría estar expuesta.

Finalmente, es oportuno enfatizar que el modelado geométrico de estructuras anatómicas a partir de imágenes médicas no se limita solo a aplicaciones en el área médica y de bioingeniería. Otras disciplinas, como los ensayos de materiales no destructivos, el prototipado rápido y las geociencias también se pueden beneficiar de la estrategia de ingeniería inversa y las técnicas de modelado geométrico que se describen en este libro.

Referencias

- Benzel, E. C. (2001). Rolling Meadows. En E. Menzel (Ed.), *Biomechanics of Spine Stabilization* (pp. 437- 440). Thieme.
- Botsch, M., Kobbelt, L., Pauly, M., Alliez, P. y Lévy, B. (2010). *Polygon Mesh Processing*. CRC press.
- Brandts, J., Korotov, S. y Křížek, M. (2008). On the Equivalence of Regularity Criteria for Triangular and Tetrahedral Finite Element Partitions. *Computers & Mathematics with Applications*, 55(10), 2227-2233.
- Caínzos López, V. (2017). Simulación y análisis por elementos finitos de la columna vertebral humana. [tesis de grado, Universidade da Coruña].
- Cebal, J. R. y Lohner, R. (1999). From Medical Images to CFD Meshes. En K. Shimada (Ed.), *Proceedings of 8th International Meshing Roundtable* (pp. 321-331). Springer. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=5e724ea6dae2f1bf0e2941399fb206ce83c31026>
- De, Guilak, F. y Mofrad, M. R. (2010). *Computational Modeling in Biomechanics*. Springer.
- Dice, L. R. (1945). Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3), 297-302.
- Dietrich, C. A., Scheidegger, C. E., Schreiner, J., Comba, J. L., Nedel, L. P. y Silva, C. T. (2008). Edge Transformations for Improving Mesh Quality of Marching Cubes. *IEEE Transactions on Visualization and Computer Graphics*, 15(1), 150-159.
- d'Otreppe, V., Boman, R. y Ponthot, J.-P. (2012). Generating Smooth Surface Meshes from Multi-region Medical Images. *International Journal for Numerical Methods in Biomedical Engineering*, 28(6-7), 642-660.

- Du, J., Jin, Y. y Tong, R. (2015). As-equilateral-as-possible Surface Remeshing. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 9(4), 1- 12.
- Du, Q., Wang, D. y Zhu, L. (2009). On Mesh Geometry and Stiffness Matrix Conditioning for General Finite Element Spaces. *SIAM Journal on Numerical Analysis*, 47(2), 1421-1444.
- Erten, H. y Üngör, A. (2009, 23-36 de junio). *Computing Triangulations Without Small and Large Angles*, [conferencia]. Sixth International Symposium on Voronoi Diagrams, Copenhagen, Denmark.
- Fabri, A. y Pion, S. (2009). CGAL: The Computational Geometry Algorithms Library. En *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 538-539). Association for Computing Machinery.
- Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.-C., Pujol, S., Bauer, C., Jennings, D., Fennessy, F. y Sonka, M. (2012). 3d Slicer as an Image Computing Platform for the Quantitative Imaging Network. *Magnetic Resonance Imaging*, 30(9), 1323-1341.
- Feng, P., Ju, T. y Warren, J. (2010). Piecewise Tri-linear Contouring for Multimaterial Volumes. En B. Mourrain, S. Schaefer y G. Xu (Eds.), *Advances in Geometric Modeling and Processings* (pp. 43-56). Springer. https://doi.org/10.1007/978-3-642-13411-1_4
- Field, D. A. (1988). Laplacian Smoothing and Delaunay Triangulations. *International Journal for Numerical Methods in Biomedical Engineering*, 4(6), 709-712.
- Finley, S. M., Brodke, D. S., Spina, N. T., DeDen, C. A. y Ellis, B. J. (2018). Febio Finite Element Models of the Human Lumbar Spine. *Computer Methods in Biomechanics and Biomedical Engineering*, 21(6), 444-452.
- Frey, P. y Borouchaki, H. (1997). *Surface Mesh Evaluation* [Conferencia]. 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Nueva York, Estados Unidos.
- Frey, P. J. y George, P.-L. (2007). *Mesh Generation: Application to Finite Elements*. Iste.
- Gao, Z., Gibson, I., Ding, C., Wang, J. y Wang, J. (2015). Virtual Lumbar Spine of Multi-body Model Based on Simbody. *Procedia Technology*, 20, 26-31.

- Garland, M. y Heckbert, P. S. (1997). Surface Simplification Using Quadric Error Metrics. En M. Garland y P. S. Heckbert, *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, (pp. 209-216). ACM Press/Addison-Wesley Publishing Co.
- Garland, M. y Zhou, Y. (2005). Quadric-based Simplification in Any Dimension. *ACM Transactions on Graphics (TOG)*, 24(2), 209-239.
- Gavidia, G., Soudah, E., Martín-Landrove, M. y Cerrolaza, M. (2011). Generación de modelos discretos de tejidos del ser humano a través del preprocesamiento y segmentación de imágenes médicas. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 27(3), 200-226.
- Gómez-Mora, M. (2018). Geometric Modeling of Complex Anatomical Structures Based on Implicit Functions. *Contemporary Engineering Sciences*, 11(43), 2121-2133.
- Gómez-Mora, M. y Flórez-Valencia, L. (2016). Modeling of Heterogeneous Objects: An Approach Based on Implicit Functions. *Revista Tecnura*, 20, 15-28.
- Gómez-Mora, M. y Rodríguez, R. (2018, 11-13 de octubre). Modelado tridimensional de objetos heterogéneos a partir de imágenes médicas: herramientas y algoritmos. En N. Becerra, *Congreso Internacional México-Colombia Cicom 2018* (pp. 67-75).
- Gómez-Mora, M., Rodríguez, R. y Vanegas, C. (2021). Modelado geométrico de estructuras anatómicas para el análisis por elementos finitos. En *Congreso Internacional México-Colombia Cicom 2018* (pp. 127-321).
- Hansen, C. D. y Johnson, C. R. (2011). *Visualization Handbook*. Elsevier.
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. y Stuetzle, W. (1993). Mesh Optimization. En Whitton, M. C., *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, (pp. 19-26). Association for Computing Machinery. <https://doi.org/10.1145/166117.166119>
- Huang, X. y Xu, D. (2017). Aspect-ratio Based Triangular Mesh Smoothing. En *ACM SIGGRAPH 2017 Posters*, (pp. 1-2). Association for Computing Machinery. <https://doi.org/10.1145/3102163.3102189>
- Huttenlocher, D. P., Klanderman, G. A. y Rucklidge, W. J. (1993). Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9), 850-863.

- Huynh, K., Gibson, I., Jagdish, B. y Lu, W. (2015). Development and Validation of a Discretised Multi-body Spine Model in Lifemod for Biodynamic Behaviour Simulation. *Computer Methods in Biomechanics and Biomedical Engineering*, 18(2), 175-184.
- Jalalian, A., Gibson, I. y Tay, E. H. (2013). Computational Biomechanical Modeling of Scoliotic Spine: Challenges and Opportunities. *Spine Deformity*, 1(6), 401-411.
- Ju, T., Losasso, F., Schaefer, S. y Warren, J. (2002). Dual Contouring of Hermite Data. *ACM Transactions on Graphics*, 21(3), 339-346.
- Khan, D., Yan, D.-M., Wang, Y., Hu, K., Ye, J. y Zhang, X. (2018). High-quality 2D Mesh Generation Without Obtuse and Small Angles. *Computers & Mathematics with Applications*, 75(2), 582-595.
- Knupp, P. M. (2000). Achieving Finite Element Mesh Quality via Optimization of the Jacobian Matrix Norm and Associated Quantities. *International Journal for Numerical Methods in Engineering*, 48(8), 1165-1185.
- Knupp, P. M. (2003). Algebraic Mesh Quality Metrics for Unstructured Initial Meshes. *Finite Elements in Analysis and Design*, 39(3), 217-241.
- Kobbelt, L., Campagna, S. y Seidel, H.-P. (1998). A General Framework for Mesh Decimation. En K. S. Booth y A. Fournier, *Proceedings of Graphics Interface '98*, (pp. 43-50). <https://doi.org/10.20380/GI1998.06>
- Lakare, S. y Kaufman, A. (2000). *3D Segmentation Techniques for Medical Volumes*. Center for Visual Computing, Department of Computer Science y State University of New York.
- Lavaste, F., Skalli, W., Robin, S., Roy-Camille, R. y Mazel, C. (1992). Three-dimensional Geometrical and Mechanical Modelling of the Lumbar Spine. *Journal of Biomechanics*, 25(10), 1153-1164.
- Levine, J. A., Paulsen, R. R. y Zhang, Y. (2012). Mesh Processing in Medical-image Analysis: a Tutorial. *IEEE Computer Graphics and Applications*, 32(5), 22-28.
- Lindstrom, P. (2000). Out-of-core Simplification of Large Polygonal Models. En J. R. Brown y K. Akeley, *Proceedings of The 27th Annual Conference on Computer Graphics and Interactive Techniques*, (pp. 259-262). ACM Press/Addison-Wesley Publishing Co. <https://doi.org/10.1145/344779.344912>
- Lindstrom, P. y Silva, C. T. (2001). A Memory Insensitive Technique for Large Model Simplification. En Bailey, M., Hansen, C., Pfister, H. y Swan, E., *VIS '01: Proceedings Visualization* (pp. 121-550). IEEE.

- Little, J. y Adam, C. (2015). Geometric Sensitivity of Patient-specific Finite Element Models of the Spine to Variability in User-selected Anatomical Landmarks. *Computer Methods in Biomechanics and Biomedical Engineering*, 18(6), 676-688.
- Lorensen, W. E. y Cline, H. E. (1987). Marching Cubes: A High Resolution 3d Surface Construction Algorithm. *ACM Siggraph Computer Graphics*, 21, 163-169.
- Luebke, D. P. (2001). A Developer's Survey of Polygonal Simplification Algorithms. *IEEE Computer Graphics and Applications*, 21(3), 24-35.
- Maas, S. A., Ellis, B. J., Ateshian, G. A. y Weiss, J. A. (2012). Febio: Finite Elements for Biomechanics. *Journal of Biomechanical Engineering*, 134(1), 1-10.
- Maas, S. A., Ellis, B. J., Rawlins, D. S. y Weiss, J. A. (2016). Finite Element Simulation of Articular Contact Mechanics with Quadratic Tetrahedral Elements. *Journal of Biomechanics*, 49(5), 659-667.
- Machado, M., Flores, P. y Ambrósio, J. (2015). Techniques for Geometrical Detection of Contact within Multibody Systems. En F. Viadero y M. Ceccarelli, *New Trends in Mechanism and Machine Science*, (pp. 471-478). Springer.
- Miele, V. J., Panjabi, M. M. y Benzel, E. C. (2012). Anatomy and Biomechanics of the Spinal Column and Cord. En *Handbook of Clinical Neurology*, 109, 31-43.
- Nabhani, F. y Wake, M. (2002). Computer Modelling and Stress Analysis of the Lumbar Spine. *Journal of Materials Processing Technology*, 127(1), 40-47.
- Ozbolat, I. T. y Koc, B. (2011). Multi-directional Blending for Heterogeneous Objects. *Computer-Aided Design*, 43(8), 863-875.
- Panagiotopoulou, O. (2009). Finite Element Analysis (fea): Applying an Engineering Method to Functional Morphology in Anthropology and Human Biology. *Annals of Human Biology*, 36(5), 609-623.
- Parthasarathy, V., Graichen, C. y Hathaway, A. (1994). A Comparison of Tetrahedron Quality Measures. *Finite Elements in Analysis and Design*, 15(3), 255-261.
- Pavarino, E., Neves, L. A., Machado, J. M., de Godoy, M. F., Shiyou, Y., Momente, J. C., Zafalon, G. F., Pinto, A. R. y Valêncio, C. R. (2013). Free Tools and Strategies for the Generation of 3d Finite Element Meshes: Modeling of the Cardiac Structures. *Journal of Biomedical Imaging*, 5, 1-11.

- Pébay, P. y Baker, T. (2003). Analysis of Triangle Quality Measures. *Mathematics of Computation*, 72(244), 1817-1839.
- Preim, B. y Botha, C. P. (2013). *Visual Computing for Medicine: Theory, Algorithms, and Applications*. Newnes.
- Reddy, J. N. (2017). *Energy Principles and Variational Methods in Applied Mechanics*. John Wiley & Sons.
- Renze, K. J. y Oliver, J. H. (1996). Generalized Unstructured Decimation. *IEEE Computer Graphics and Applications*, 16(6), 24-32.
- Rico, H. A. B. (2018). *Diseño y simulación de un tanque tipo II revestido con alambre de acero SA-905* [tesis de maestría, Universidad de Guanajuato]. <http://www.repositorio.ugto.mx/handle/20.500.12059/392>
- Rineau, L. y Yvinec, M. (2007). A Generic Software Design for Delaunay Refinement Meshing. *Computational Geometry*, 38(1-2), 100-110.
- Rossignac, J. y Borrel, P. (1993). Multi-resolution 3D Approximations for Rendering Complex Scenes. En B. Falcidieno y T. L. Kunii (Eds.), *Modeling in Computer Graphics*, (pp. 455-465). Springer.
- Roussellet, V. (2018). *Implicit Muscle Models for Interactive Character Skinning*. [Tesis doctoral, Université Paul Sabatier-Toulouse III].
- Schroeder, W., Martin, K. M. y Lorensen, W. E. (1998). *The Visualization Toolkit: an Object-oriented Approach to 3D Graphics*. Prentice-Hall, Inc.
- Schroeder, W., Maynard, R. y Geveci, B. (2015, 25-26 de octubre). Flying Edges: A High-performance Scalable Isocontouring Algorithm. En J. Bennett, H. Childs y M. Hadwiger, *2015 IEEE 5th Symposium on Large Data Analysis and Visualization (LDAV)*, (pp. 33-40). <https://ieeexplore.ieee.org/document/7348069>
- Schroeder, W. J. (1997). A Topology Modifying Progressive Decimation Algorithm. En *Proceedings. Visualization '97*, (pp. 205-212). <https://ieeexplore.ieee.org/document/663883>
- Schroeder, W. J., Zarge, J. A. y Lorensen, W. E. (1992). Decimation of Triangle Meshes. En J. J. Thomas (Ed.), *SIGGRAPH92: 19th Annual Conference and Exhibition on Computer Graphics and Interactive Techniques*, (pp. 65-70).
- Shaffer, E. y Garland, M. (2001). Efficient Adaptive Simplification of Massive Meshes. En *Proceedings Visualization 2001*, (pp. 127-551). <https://ieeexplore.ieee.org/document/964503/authors#authors>
- Si, H. (2015). Tetgen, a Delaunay-based Quality Tetrahedral Mesh Generator. *ACM Transactions on Mathematical Software (TOMS)*, 41(2), 1-36.

- Si, H. y Gärtner, K. (2005). Meshing Piecewise Linear Complexes by Constrained Delaunay Tetrahedralizations. En *Proceedings of the 14th International Meshing Roundtable*, (pp. 147-163). Springer.
- Sun, W., Starly, B., Nam, J. y Darling, A. (2005). Bio-cad Modeling and its Applications in Computer-aided Tissue Engineering. *Computer-aided Design*, 37(11), 1097-1114.
- Taubin, G. (1995a). Curve and Surface Smoothing without Shrinkage. En *Proceedings of IEEE International Conference on Computer Vision*, (pp. 852-857). IEEE.
- Taubin, G. (1995b). A Signal Processing Approach to Fair Surface Design. En *Proceedings of The 22nd Annual Conference on Computer Graphics and Interactive Techniques*, (pp. 351-358). Association for Computing Machinery.
- Taubin, G. (2000). Geometric Signal Processing on Polygonal Meshes. En H. R. B. Peroche (Ed.). *Rendering Techniques 2000: Proceedings of the Eurographics Workshop*, (pp. 1-11), Brno, Czech Republic.
- Taubin, G., Zhang, T. y Golub, G. (1996). Optimal Surface Smoothing as Filter Design. En B. Buxton y R. Cipolla, *European Conference on Computer Vision*, (pp. 283-292), Cambridge, Reino Unido. Springer.
- Trotts, I., Hamann, B., Joy, K. y Wiley, D. (1998). *Simplification of Tetrahedral Meshes*. Department of Computer Engineering and Institute of Engineering and Sciences of Bilkent University.
- Trotts, I. J., Hamann, B. y Joy, K. I. (1999). Simplification of Tetrahedral Meshes with Error Bounds. *IEEE Transactions on Visualization and Computer Graphics*, 5(3), 224-237.
- Vollmer, J., Mencl, R. y Mueller, H. (1999). Improved Laplacian Smoothing of Noisy Surface Meshes. En *Computer Graphics Forum*, 18, pp. 131-138. Wiley Online Library.
- Wang, Y., Yan, D.-M., Liu, X., Tang, C., Guo, J., Zhang, X. y Wonka, P. (2018). Isotropic Surface Remeshing without Large and Small Angles. *IEEE Transactions on Visualization and Computer Graphics*, 25(7), 2430-2442.
- Wittek, A., Joldes, G., Nielsen, P. M., Doyle, B. J. y Miller, K. (2017). *Computational Biomechanics for Medicine: From Algorithms to Models and Applications*. Springer.
- Wu, X., Liu, W. y Wang, M. Y. (2008). A CAD Modeling System for Heterogeneous Object. *Advances in Engineering Software*, 39(5), 444-453.

- Xu, Q.-C., Yan, D.-M., Li, W. y Yang, Y.-L. (2019). Anisotropic Surface Remeshing without Obtuse Angles. En *Computer Graphics Forum*, 38, pp. 755-763. Wiley Online Library.
- Yamamoto, I., Panjabi, M. M., Crisco, T. y Oxland, T. (1989). Three-dimensional Movements of the Whole Lumbar Spine and Lumbosacral Joint. *Spine*, 14(11), 1256-1260.
- Yu, M., Yanning, Z., Huiling, L., Shuang, X. y Ting, Y. (2012). A Tracking Algorithm Based on Fractional Differential Gradient within 3d Image. En *2012 International Conference on Biomedical Engineering and Biotechnology*, (pp. 696-699), Macau, China. IEEE.
- Zhang, Y., Bajaj, C. y Xu, G. (2009). Surface Smoothing and Quality Improvement of Quadrilateral/Hexahedral Meshes with Geometric Flow. *Communications in Numerical Methods in Engineering*, 25(1), 1-18.
- Zhou, K., Huang, J., Snyder, J., Liu, X., Bao, H., Guo, B. y Shum, H.-Y. (2005). Large Mesh Deformation Using the Volumetric Graph Laplacian. En *ACM Transactions on Graphics (TOG)*, 24, pp. 496-503.
- Zlámal, M. (1968). On the Finite Element Method. *Numerische Mathematik*, 12(5), 394-409.

Anexos

Anexo A. Introducción al método de elementos finitos

Actualmente, el análisis por EF hace parte básica de la ingeniería moderna y es de gran utilidad en la resolución de problemas biomédicos relacionados con la planificación y simulación quirúrgica, la planificación de terapias, el seguimiento del progreso de enfermedades y el diseño de prótesis, entre otras (De *et al.*, 2010). Para un uso eficaz de cualquier *software* para análisis por EF es importante una comprensión básica del método de EF mediante el cual se resuelve numéricamente cualquier fenómeno físico que esté descrito mediante ecuaciones diferenciales parciales.

El método de EF es una técnica numérica que se utiliza para análisis por EF de cualquier fenómeno físico dado. Considérese la deformación axial de una varilla (figura A.1). La ecuación diferencial ordinaria (EDO) que describe esta deformación es:

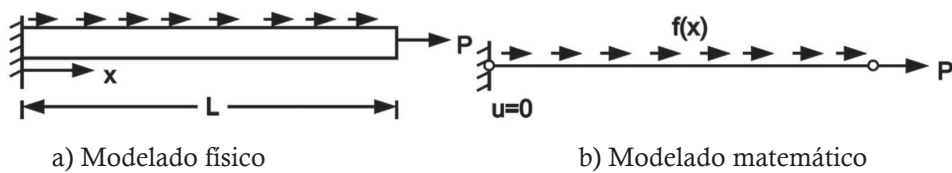
$$-\frac{d}{dx}\left(EA\frac{du}{dx}\right) - f = 0; 0 \leq x \leq L \quad (\text{A.1})$$

donde A es el área de la sección transversal, f es la fuerza distribuida y E es el módulo de la varilla. Las condiciones de frontera para el caso de la figura A.1 son:

$$\begin{aligned} u(0) &= 0 \\ \left[\left(EA \frac{du}{dx} \right) \right]_{x=L} &= P \end{aligned} \quad (\text{A.2})$$

La barra real que se muestra en la figura A.1a se modela matemáticamente como una línea en la figura A.1b, ocupando el dominio $[0; L]$ a lo largo del eje real x .

Figura A.1. Modelo físico y matemático de una varilla

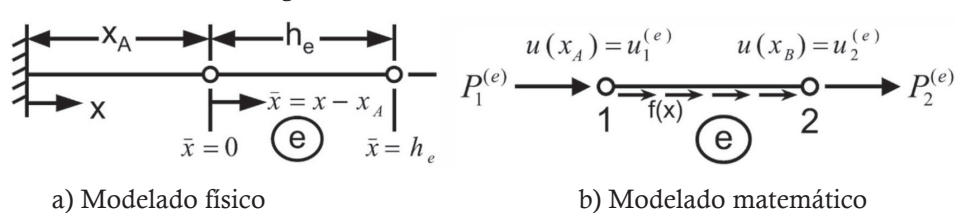


Fuente: elaboración propia.

A.1. Discretización

En el paso siguiente se divide el dominio en elementos discretos, como se muestra en la figura A.2.

Figura A.2. Discretización en elementos



Fuente: elaboración propia.

Para obtener las ecuaciones de los elementos, se utiliza una forma integral de la ecuación diferencial ordinaria (EDO), que se obtiene integrando el producto de la EDO por una función de peso v de la siguiente manera:

$$0 = \int_{x_A}^{x_B} v \left[-\frac{d}{dx} \left(EA \frac{du}{dx} \right) - f \right] dx \quad (\text{A.3})$$

Esto se llama *forma débil* porque la solución $u(x)$ no tiene que satisfacer la EDO (A.1) para todos y cada uno de los valores infinitos de x en $[0; L]$, en un sentido estricto (*fuerte*). En cambio, la solución $u(x)$ solo tiene que satisfacer la EDO en (A.3), en un sentido de promedios ponderados (*débil*).

Por tanto, es más fácil encontrar una solución débil que una fuerte. Aunque para el caso de la varilla se conoce la solución fuerte (exacta), la mayoría de los problemas de la mecánica por componentes no tienen una solución exacta. La ecuación gobernante se obtiene integrando por partes (A.3) de la siguiente manera:

$$0 = \int_{x_A}^{x_B} EA \frac{dv}{dx} \frac{du}{dx} dx - \int_{x_A}^{x_B} v f dx - \left[v \left(EA \frac{du}{dx} \right) \right]_{x_A}^{x_B} \quad (\text{A.4})$$

donde $v(x)$ es una función de ponderación, que generalmente se establece igual a la variable primaria $u(x)$. Esto permite concluir que:

- Especificar $v(x)$ en x_A o x_B es una condición de frontera esencial.
- Especificar $\left(EA \frac{du}{dx} \right)$ en cualquier extremo es la condición de frontera natural.

Si $u(x)$ es la variable principal y $\left(EA \frac{du}{dx} \right) = EA \epsilon_x = A \sigma_x$ es la variable secundaria, entonces (figura A.2b)

$$\begin{aligned} u(x_A) &= u_1^e \\ u(x_B) &= u_2^e \\ - \left[\left(EA \frac{du}{dx} \right) \right]_{x_A} &= P_1^e \\ \left[\left(EA \frac{du}{dx} \right) \right]_{x_B} &= P_2^e \end{aligned} \quad (\text{A.5})$$

donde la ecuación gobernante se convierte en

$$0 = \int_{x_A}^{x_B} EA \frac{dv}{dx} \frac{du}{dx} dx - P_1^e v(x_A) - P_2^e v(x_B) = B(v, u) - l(v) \quad (\text{A.6})$$

con

$$\begin{aligned} B(u, v) &= \int_{x_A}^{x_B} EA \frac{dv}{dx} \frac{du}{dx} dx \\ l(v) &= \int_{x_A}^{x_B} v f dx + P_1^e v(x_A) + P_2^e v(x_B) \end{aligned} \quad (\text{A.7})$$

Así la función desconocida $u(x)$ se aproxima como una combinación lineal de funciones conocidas $N_i^e(x)$, mientras que los coeficientes desconocidos a_j^e se aproximan como $u_e(x) = \sum_{j=1}^n a_j^e N_j^e(x)$ donde a_j^e son los coeficientes que se deben encontrar y $N_i^e(x)$ son funciones de interpolación. Para la función de ponderación $v(x)$, se puede utilizar el método de Ritz (Reddy, 2017), en el que $v(x) = N_i^e(x)$. Sustituyendo en la ecuación gobernante (A.6) se obtiene:

$$\sum_{j=1}^n \left(\int_{x_A}^{x_B} EA \frac{dN_i^e}{dx} \frac{dN_j^e}{dx} dx \right) a_j^e = \int_{x_A}^{x_B} N_i^e f dx + P_1^e N_i^e(x_A) - P_2^e N_i^e(x_B) \quad (\text{A.8})$$

que se puede escribir como

$$\sum_{j=1}^n K_{i,j}^e a_j^e(x) = F_i^e \quad (\text{A.9})$$

o en forma de matriz

$$[K^e] \{a^e\} = \{F^e\} \quad (\text{A.10})$$

donde $[K^e]$ es la matriz de rigidez del elemento, $\{F^e\}$ es la fuerza equivalente del vector del elemento y $\{a^e\}$ son los parámetros desconocidos del elemento.

A.2. Funciones de interpolación

Aunque cualquier conjunto completo de funciones linealmente independientes podría usarse como funciones de interpolación, es conveniente elegir la función de tal manera que los coeficientes desconocidos representen los desplazamientos de los nodos, es decir $a_i = u_i$. Para el caso de un elemento con 2 nodos, que abarque el intervalo $x_e \leq x \leq x_e + 1$, se pueden utilizar las siguientes funciones de interpolación lineal (figura A.3):

$$N_1^e = \frac{x_e + 1 - x}{h_e}$$

$$N_2^e = \frac{x - x_e}{h_e}$$
(A.11)

donde $h_e = x_{e+1} - x_e$ es la longitud del elemento.

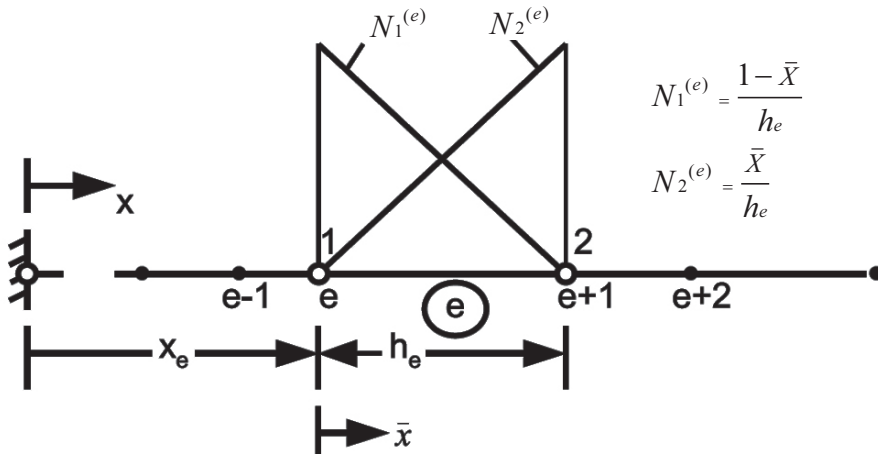
Estas funciones de interpolación satisfacen las siguientes condiciones:

$$N_i^e(x_j) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$
(A.12)

$$\sum_{i=1}^2 N_i^e(x)$$
(A.13)

lo que garantiza que los coeficientes desconocidos representen los desplazamientos nodales, es decir, $a_i = u_i$.

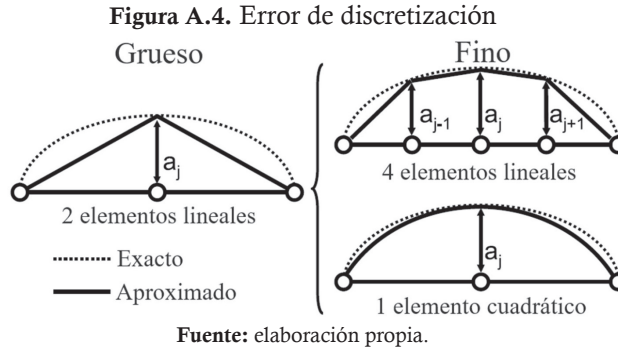
Figura A.3. Funciones de interpolación lineal para una barra de dos nodos



Fuente: elaboración propia.

Se pueden utilizar diferentes funciones de interpolación, con sus ventajas y desventajas. Sin embargo, estas están íntimamente relacionadas con el

número de nodos del elemento. Entonces, más nodos por elemento implican más precisión y menos necesidad de una malla fina, pero también implican un mayor costo en términos de tiempo computacional. La figura A.4 ilustra cómo la solución aproximada converge a la exacta cuando el número de elementos aumenta de 2 a 4, o cuando el número de nodos en el elemento aumenta de 2 para el elemento lineal a 3 para el elemento cuadrático.



Cuando las funciones de interpolación satisfacen las condiciones en las ecuaciones A.12 y A.13, es posible reescribir la ecuación A.10 como:

$$[K^e] \{u^e\} = \{F^e\} \quad (\text{A.14})$$

donde $\{u^e\}$ son los desplazamientos nodales, $[K^e]$ es la matriz de rigidez del elemento dada por:

$$[K^e] = \begin{bmatrix} \int_{x_A}^{x_B} EA \frac{dN_1^e}{dx} \frac{dN_1^e}{dx} dx & \int_{x_A}^{x_B} EA \frac{dN_1^e}{dx} \frac{dN_2^e}{dx} dx \\ \int_{x_A}^{x_B} EA \frac{dN_2^e}{dx} \frac{dN_1^e}{dx} dx & \int_{x_A}^{x_B} EA \frac{dN_2^e}{dx} \frac{dN_2^e}{dx} dx \end{bmatrix} \quad (\text{A.15})$$

y $\{F^e\}$ es el vector de fuerzas (cargas) del elemento:

$$\{F^e\} = \begin{Bmatrix} \int_{x_A}^{x_B} N_1^e f dx + P_1^e \\ \int_{x_A}^{x_B} N_2^e f dx + P_2^e \end{Bmatrix} \quad (\text{A.16})$$

Para un elemento e de una varilla de dos nodos, son fijos tanto el área de sección transversal constante A_e y la longitud del elemento h_e como el módulo E . Estos valores definen la rigidez del elemento de tracción-compresión mediante:

$$k^e = \frac{EA_e}{h_e} \quad (\text{A.17})$$

Las cargas externas sobre el elemento son la fuerza distribuida f_e , la fuerza en el extremo 1, P_1^e , y la fuerza en el extremo 2, P_2^e (figura A.2b). Usando estos valores, las funciones de interpolación lineal (ecuación A.11), así como A.15 y A.16, la rigidez de la matriz del elemento y las fuerzas nodales equivalentes se convierten en:

$$[K^e] = \begin{bmatrix} k^e & -k^e \\ -k^e & k^e \end{bmatrix} = \frac{EA_e}{h_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (\text{A.18})$$

$$\{F^e\} = \frac{f_e h_e}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} + \begin{Bmatrix} P_1^e \\ P_2^e \end{Bmatrix} \quad (\text{A.19})$$

A.3. Ensamblaje de las ecuaciones de los elementos

Los parámetros desconocidos del elemento corresponden a los desplazamientos en los nodos del elemento. Dado que un nodo debe tener el mismo desplazamiento en los elementos adyacentes, se deduce que el valor es el mismo. Por ejemplo, usando la conectividad de elementos que se muestra en la figura A.5, se asignan etiquetas únicas a los desplazamientos, usando letras mayúsculas.

Figura A.5. Conectividad entre 3 elementos de 2 nodos



Fuente: elaboración propia.

Si el superíndice denota el número del elemento y el subíndice indica el número del nodo, entonces se tiene que:

$$\begin{aligned} u_1^1 &= U_1 \\ u_2^1 &= U_2 = u_1^2 \\ u_1^2 &= U_3 = u_1^3 \\ u_2^3 &= U_4 \end{aligned} \quad (\text{A.20})$$

Ahora, las ecuaciones de los elementos se pueden ensamblar en el sistema global. Primero, la contribución del elemento uno es:

$$\begin{bmatrix} k^1 & -k^1 & 0 & 0 \\ -k^1 & k^1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} f_1 h_1 / 2 \\ f_1 h_1 / 2 \\ 0 \\ 0 \end{Bmatrix} + \begin{Bmatrix} P_1^1 \\ P_2^1 \\ 0 \\ 0 \end{Bmatrix} \quad (\text{A.21})$$

A continuación, se agrega la contribución del elemento dos, como sigue:

$$\begin{bmatrix} k^1 & -k^1 & 0 & 0 \\ -k^1 & k^1 + k^2 & -k^2 & 0 \\ 0 & -k^2 & k^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} f_1 h_1 / 2 \\ f_1 h_1 / 2 + f_2 h_2 / 2 \\ 0 \\ 0 \end{Bmatrix} + \begin{Bmatrix} P_1^1 \\ P_2^1 + P_1^2 \\ P_2^2 \\ 0 \end{Bmatrix} \quad (\text{A.22})$$

Finalmente, se agrega el elemento tres para obtener el sistema completamente ensamblado, de la siguiente manera:

$$\begin{bmatrix} k^1 & -k^1 & 0 & 0 \\ -k^1 & k^1 + k^2 & -k^2 & 0 \\ 0 & -k^2 & k^2 & -k^3 \\ 0 & 0 & -k^3 & k^3 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} = \frac{1}{2} \begin{Bmatrix} f_1 h_1 / 2 \\ f_1 h_1 / 2 + f_2 h_2 / 2 \\ f_2 h_2 + f_3 h_3 \\ f_3 h_3 \end{Bmatrix} + \begin{Bmatrix} P_1^1 \\ P_2^1 + P_1^2 \\ P_2^2 + P_1^3 \\ P_2^3 \end{Bmatrix} \quad (\text{A.23})$$

A.4. Condiciones de frontera

Teniendo en cuenta la ley del equilibrio (figura A.2), las cargas internas se cancelan siempre que dos elementos comparten un nodo, es decir:

$$\begin{aligned} P_2^1 + P_1^2 &= 0 \\ P_2^2 + P_1^3 &= 0 \end{aligned} \quad (\text{A.24})$$

Los restantes P_1^1 y P_2^3 son las fuerzas en el extremo de la varilla. Si cualquiera de los extremos de la varilla está fijo, entonces el desplazamiento debe establecerse en cero en ese extremo. Si el final en $x = 0$ es fijo, entonces $U_1 = 0$. Si el final en $x = L$ está libre, entonces se debe especificar P_2^3 , ya que $U_4 = 0$. Si no se especifica, entonces se supone que la fuerza es cero.

A.5. Solución de las ecuaciones

Como $U_1 = 0$, eliminando la primera fila y columna de la matriz de rigidez, se obtiene un sistema de ecuaciones algebraicas de 3×3 , y se resuelven 3 incógnitas: U_2 , U_3 , U_4 . Una vez que se encuentra una solución para U_2 , la reacción P_1^1 se calcula a partir de la primera ecuación de (A.23), como sigue:

$$-k^1 U_2 = \frac{f_1 h_1}{2} + P_1^1 \quad (\text{A.25})$$

Ahora, la solución U_i en los 4 puntos a lo largo de la barra está disponible. A continuación, la solución en cualquier ubicación x se puede calcular mediante las funciones de interpolación, como sigue:

$$U^e(x) = \sum_{j=1}^2 U_j^e N_j^e(x) \quad (\text{A.26})$$

o lo que es igual,

$$(x) = \begin{cases} U_1 N_1^1(x) + U_2 N_2^1(x) & \text{if } 0 \leq x \leq h_1 \\ U_2 N_1^2(x) + U_3 N_2^2(x) & \text{if } h_1 \leq x \leq h_1 + h_2 \\ U_3 N_1^3(x) + U_4 N_2^3(x) & \text{if } h_1 + h_2 \leq x \leq h_1 + h_2 + h_3 \end{cases} \quad (\text{A.27})$$

La derivación de las ecuaciones de elementos, ensamblaje y solución para cualquier tipo de elementos es similar a la del elemento de varilla unidimensional descrito hasta este punto, con la excepción de que se usa el principio

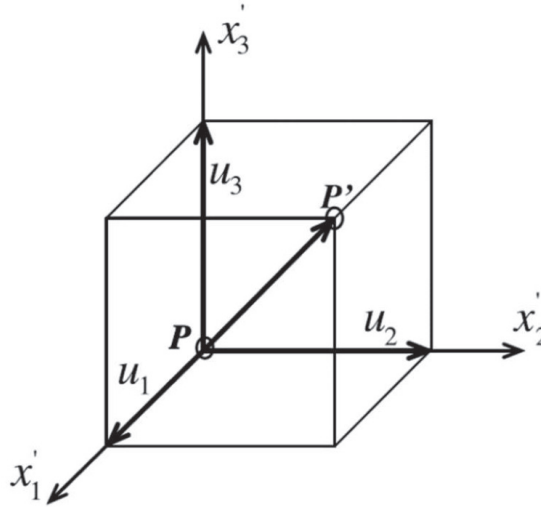
de trabajo virtual en lugar de la ecuación gobernante A.1. El principio de trabajo virtual proporciona una forma débil, similar a la de la ecuación A.4.

A.6. Resultados derivados

A.6.1. Desplazamientos

Cada punto de un objeto puede trasladarse y girar como un cuerpo rígido, así como deformarse para ocupar un nuevo espacio, bajo la acción de diferentes fuerzas.

Figura A.6. Notación para componentes de desplazamiento



Fuente: elaboración propia.

Así, los desplazamientos u_i de cualquier punto P del objeto (figura A.6) se definen en términos de los 3 componentes del vector u_i (en un sistema de coordenadas cartesiano rectangular) como $u_i = (u_1, u_2, u_3)$.

Siendo el desplazamiento una cantidad vectorial o tensorial de primer orden:

$$\mathbf{u} = u_i = (u_1, u_2, u_3) \quad (\text{A.28})$$

A.6.2. Deformación

Para el caso de análisis geométrico no lineal, los componentes del tensor de deformación lagrangiana se obtienen mediante:

$$L_{i,j} = \frac{1}{2} (u_{i,j}, u_{j,i}, u_{r,i} u_{r,j}) \quad (\text{A.29})$$

donde

$$u_{i,j} = \frac{\partial u_i}{\partial u_j} \quad (\text{A.30})$$

Cuando los gradientes de desplazamiento son muy pequeños y los productos de las derivadas parciales de u_i son despreciables en comparación con los términos de la derivada lineal (de primer orden), entonces el tensor de deformación $\varepsilon_{i,j}$ se calcula mediante:

$$\varepsilon = \varepsilon_{i,j} = \frac{1}{2} (u_{i,j} + u_{j,i}) \quad (\text{A.31})$$

La deformación ε_x se calcula directamente a partir de los desplazamientos conocidos dentro del elemento (Rico, 2018). Por ejemplo,

$$\varepsilon_x = \frac{du}{dx} = \sum_{j=1}^2 U_j^e \frac{N_j^e}{dx} \quad (\text{A.32})$$

Si se tiene en cuenta que $N_j^e(x)$ son funciones, entonces las deformaciones son constantes sobre el elemento. Por lo tanto, el valor de las deformaciones es un orden de magnitud menor que la variable primaria (desplazamientos).

A.6.3. Tensión

El vector de tensiones es la fuerza por unidad de área que actúa sobre un plano que pasa por un punto, el cual permite describir completamente el estado de tensión en ese punto. El vector de tensiones se puede expresar en términos de los componentes que actúan sobre 3 planos mutuamente perpendiculares alineados con las direcciones de las coordenadas ortogonales.

La notación tensorial para la tensión es $\sigma_{i,j}$ con $(i, j = 1, 2, 3)$, donde el primer subíndice corresponde a la dirección de la normal al plano de interés y el segundo subíndice corresponde a la dirección de la tensión. Las tensiones normales de tracción ($i = j$) se definen como positivas cuando la normal

al plano y las direcciones del componente de tensión son ambas positivas o negativas.

Los valores de tensión generalmente se calculan a partir de las deformaciones mediante las ecuaciones constitutivas

$$\sigma_x = E\varepsilon_x \quad (\text{A.33})$$

Se debe tener en cuenta que el valor de las tensiones debe ser el mismo que el de las deformaciones.

Anexo B. Introducción al análisis por elementos finitos con FEBio

Existen diferentes *softwares* con capacidades de análisis por EF que se utilizan en las diferentes disciplinas de la ingeniería. Estos *softwares* ayudan a resolver una gran variedad de problemas, desde el análisis estático lineal simple hasta el análisis transitorio no lineal. Al comienzo solo algunos programas comerciales como ANSYS (www.ansys.com) y Abaqus (www.3ds.com/products-services/simulia/) tenían capacidades especiales para analizar estructuras anatómicas complejas y materiales compuestos. Sin embargo, en los últimos tiempos se han desarrollado programas de código abierto, como FEBio (www.febio.org), que igualmente permiten trabajar con este tipo de objetos y aceptan ecuaciones constitutivas y formulaciones de elementos personalizadas y programadas por el usuario.

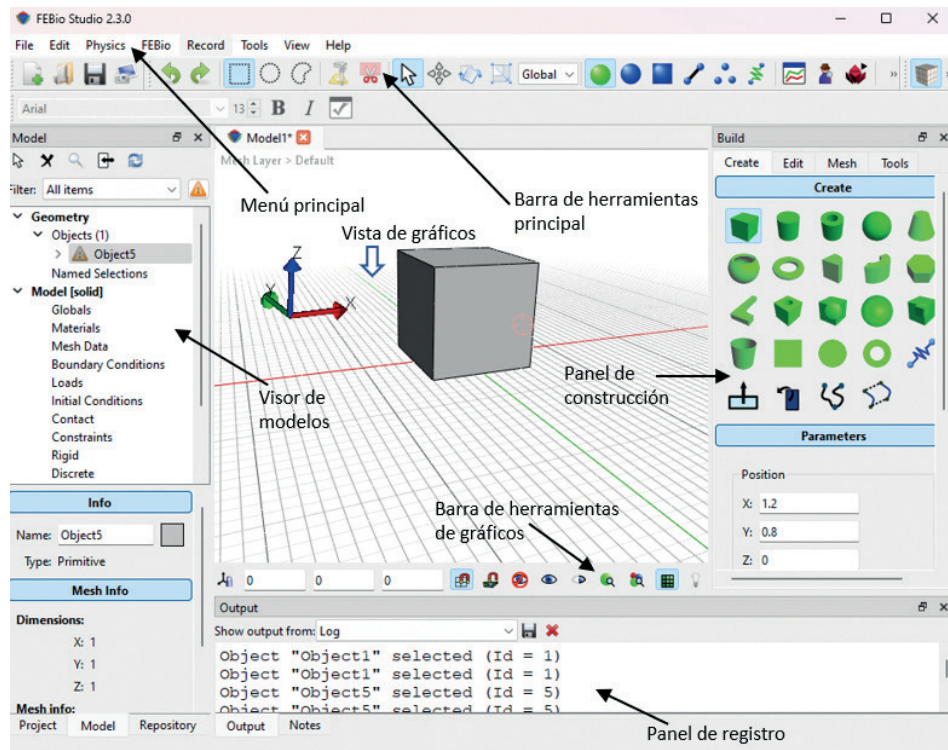
En este anexo se presenta una introducción a la interfaz gráfica de usuario (GUI) de FEBio para familiarizarse con los procedimientos típicos utilizados en el modelado de EF. El proceso para análisis por EF se organiza comúnmente en cuatro etapas: modelado geométrico (numeral B.1), preprocesamiento (numeral B.2), procesamiento (numeral B.3) y posprocesamiento (numeral B.4).

La figura B.1 muestra una descripción general de los componentes de la interfaz gráfica de FEBio Studio en la versión 1.4.2. El menú principal presenta todos los elementos de menú disponibles. La barra de herramientas principal, ubicada directamente debajo del menú, ofrece una forma alternativa de invocar algunos de estos elementos del menú. El visor de modelos muestra una descripción general de los componentes del modelo. La vista

de gráficos cubre la mayor parte de la interfaz gráfica y muestra una vista 3D del modelo. La barra de herramientas de gráficos se encuentra debajo de la vista de gráficos y muestra información sobre esta vista. El panel de construcción es donde se pueden encontrar la mayoría de las herramientas de edición de mallas y modelado de sólidos de FEBio.

La mayoría de los paneles en FEBio Studio son acoplables, es decir, se pueden mover y cerrar. Si se cierra un panel, se puede volver a mostrar buscando su nombre en la parte inferior del menú *View*.

Figura B.1. Los componentes principales de la interfaz gráfica de FEBio Studio



Fuente: elaboración propia.

B.1. Modelado geométrico

El primer paso en el proceso de diseño por EF es el modelado geométrico para obtener mallas de superficie que aproximen el dominio de la simulación.

Mediante el modelado de sólidos, el usuario crea una representación geométrica del objeto que se va a simular utilizando constructores como volúmenes, áreas, líneas y puntos. En la parte derecha de la figura B.1 se muestra el panel de construcción (*Build*) de FEBio, que cuenta con todas las herramientas de creación y edición de geometrías para la construcción de modelos sólidos. Sin embargo, se puede utilizar otro *software* de modelado como SolidWorks (www.solidworks.com) para generar el modelo sólido, que luego debe ser importado a FEBio para agregar las condiciones de frontera y las cargas, invocar el procesador (*solver*) y finalmente visualizar los resultados. Igualmente, el modelo sólido creado en FEBio se puede exportar en un formato que puede ser leído por otros procesadores como ANSYS y Abaqus.

Sin embargo, el modelado de sólidos es difícil de aplicar al modelado por EF de estructuras anatómicas, debido a que no existen constructores que permitan modelar la complejidad geométrica de dichas estructuras. En este caso, la información de la geometría se deriva de imágenes médicas, como la TC y la RM (Cebal y Lohner, 1999). Mediante un proceso de filtrado y suavizado de imágenes se obtiene una segmentación de la estructura de interés, la cual se puede convertir en un modelo de superficie 3D, como se mostró en el capítulo 2, que luego debe ser importado a FEBio para agregar las condiciones de frontera y las cargas, simular el modelo y visualizar los resultados.

B.2. Preprocesamiento

En esta parte se construye un modelo de FE para una prueba de torsión de una barra. Para esto se crean 2 cajas rectangulares y un cuerpo rígido. Luego se aplican los desplazamientos fijos y las rotaciones prescritas apropiadas para girar un extremo de una caja.

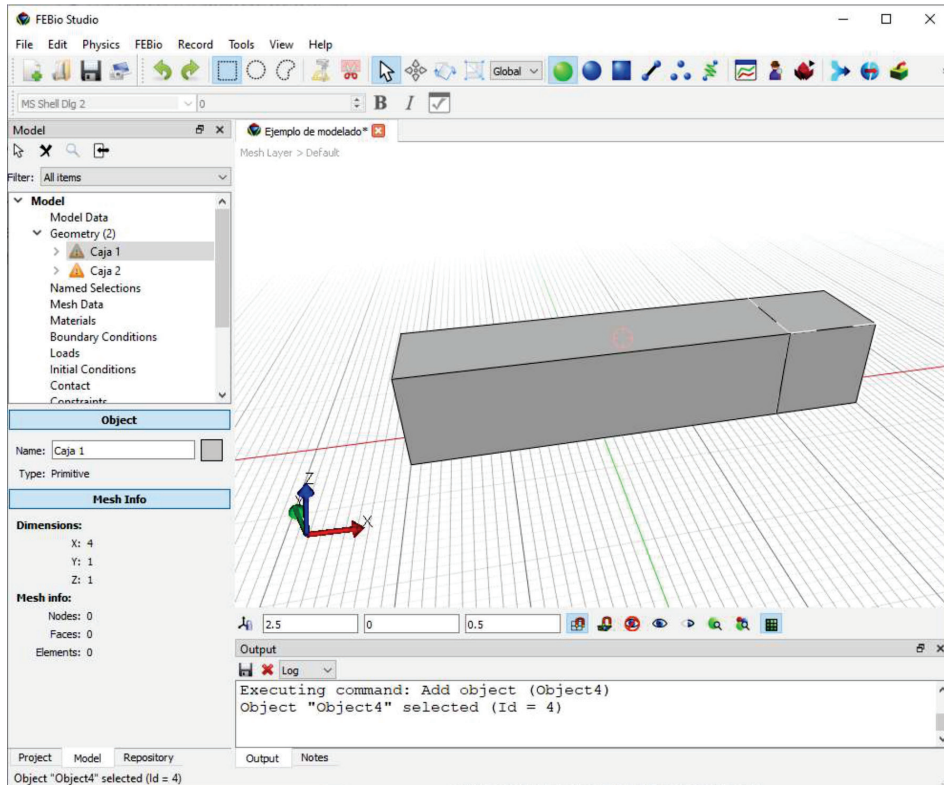
La geometría consta de 2 cajas: una para la malla deformable y otra para el cuerpo rígido. Primero, se crea la caja 1 para la malla deformable. Para esto se activa el panel de construcción y se crea una caja que se ubica en el origen (0,0,0) con los parámetros: *ancho* = 4, *alto* = 1 y *profundidad* = 1. A continuación, se crea la caja 2, ubicada en la posición (2,5,0,0) y con parámetros: *ancho* = 1, *alto* = 1 y *profundidad* = 1. Esto colocará la caja 2 justo al lado de la caja 1, como aparece en la figura B.2.

B.2.1. Ensamblado

Si existe en el modelo más de una parte (figura B.2), es necesario poner las partes juntas mediante un proceso de ensamble y así poder representar de forma adecuada el objeto físico que se quiere analizar.

Para unir la malla deformable al cuerpo rígido se necesita seleccionar la(s) cara(s) que se desea(n) unir. Antes de hacer eso, se debe agregar un cuerpo rígido al modelo activando el menú *Physics->Add material* y seleccionando el cuerpo rígido de la lista, en la categoría otro (*Other*).

Figura B.2. Geometría del modelo que se va a simular



Fuente: elaboración propia.

La creación de un material rígido también crea implícitamente un cuerpo rígido. Aparecerá un ícono en la Vista de gráficos para indicar dónde se encuentra el cuerpo rígido en el espacio. El cuerpo rígido, que se coloca

inicialmente en el origen, el cual corresponde con su centro de rotación, se debe mover al centro de la caja 2. Para esto, se selecciona el material del cuerpo rígido en el Visor de modelos y en la Lista de propiedades se establece el parámetro de centro de masa en (2,5, 0, 0,5). Esto coloca el cuerpo rígido en el lugar correcto.

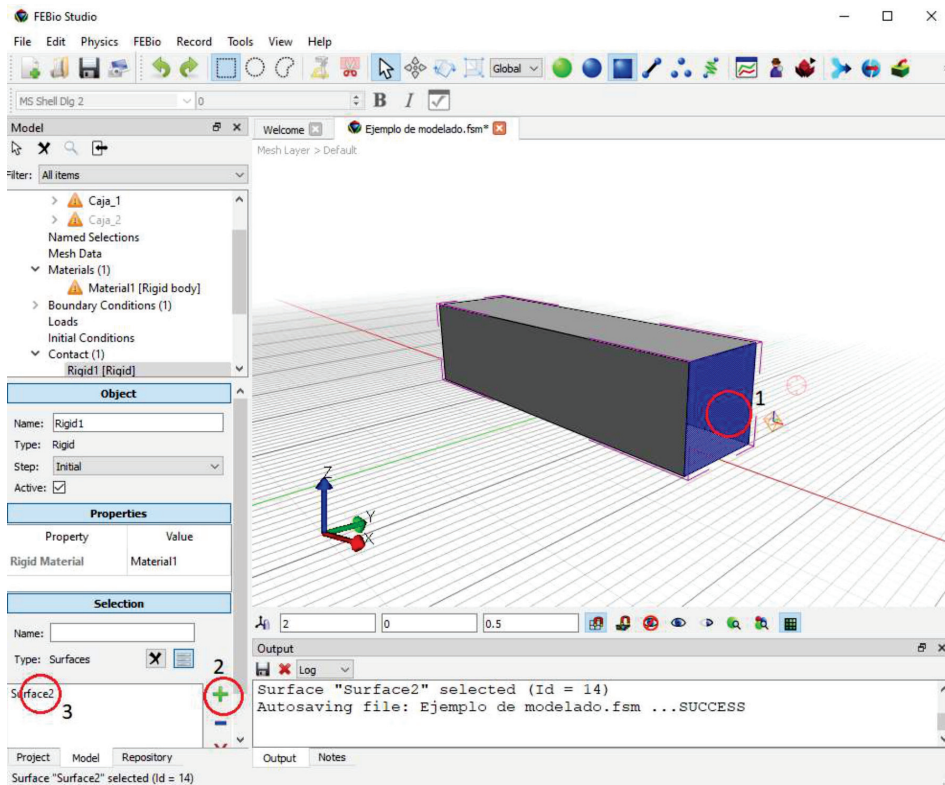
A continuación, se crea una interfaz rígida que conecta un cuerpo deformable a un cuerpo rígido. En el menú *Physics->Add Contact* se selecciona *Rigid* en el cuadro de diálogo que aparece. Una interfaz rígida solo requiere un parámetro y es el material del cuerpo rígido. Para esto se selecciona el cuerpo rígido, que se creó anteriormente, en la lista desplegable de las propiedades. Ahora se tiene que definir a qué parte de la caja 1 se aplicará esta interfaz rígida. Para facilitar la selección, primero se oculta la caja 2, la más pequeña. Se puede ocultar un objeto ubicándolo en el Visor de modelos y haciendo clic derecho sobre él. En el menú emergente, se selecciona ocultar, lo que atenúa el nombre del objeto (caja 2) en el Visor de modelos. Luego, se selecciona la cara derecha de la caja 1, es decir, la cara que está tocando la caja 2 (antes de que fuera ocultada).

En la barra de herramientas principal, ubicada en la parte superior de la interfaz de usuario (figura B.1), se ven cinco íconos de selección. El primero (una bola verde) es el botón de selección de objetos. Los cuatro botones siguientes permiten seleccionar partes, caras, aristas y nodos, respectivamente. Se hace clic en el ícono Seleccionar caras (cuadrado azul) en la barra de herramientas y luego en la cara derecha de la caja 1 para resaltar esta cara como se muestra en la figura B.3 con el número 1. En la parte inferior del Visor de modelos se enumerarán todas las superficies (o bordes o nodos) a las que se aplica esta condición de frontera. Con la cara derecha de la caja 1 resaltada, se hace clic en el signo  en el cuadro de selección, marcado con el número 2 en la figura B.3. Esto agregará la selección actual a la lista de selección de la interfaz rígida, número 3 en la figura B.3. Ahora ya se puede hacer visible la caja 2.

Hay 3 botones más disponibles en el cuadro de selección. El botón con el signo  quita los elementos que están seleccionados actualmente en la Vista de gráficos. El botón con el signo  eliminará todas las líneas seleccionadas

y los componentes del modelo correspondientes de la lista de selección. Finalmente, cuando se presiona el botón con la flecha, los elementos correspondientes a las líneas seleccionadas en el cuadro de selección se seleccionarán en la Vista de gráficos.

Figura B.3. Configuración del contacto entre un cuerpo rígido y un cuerpo deformable



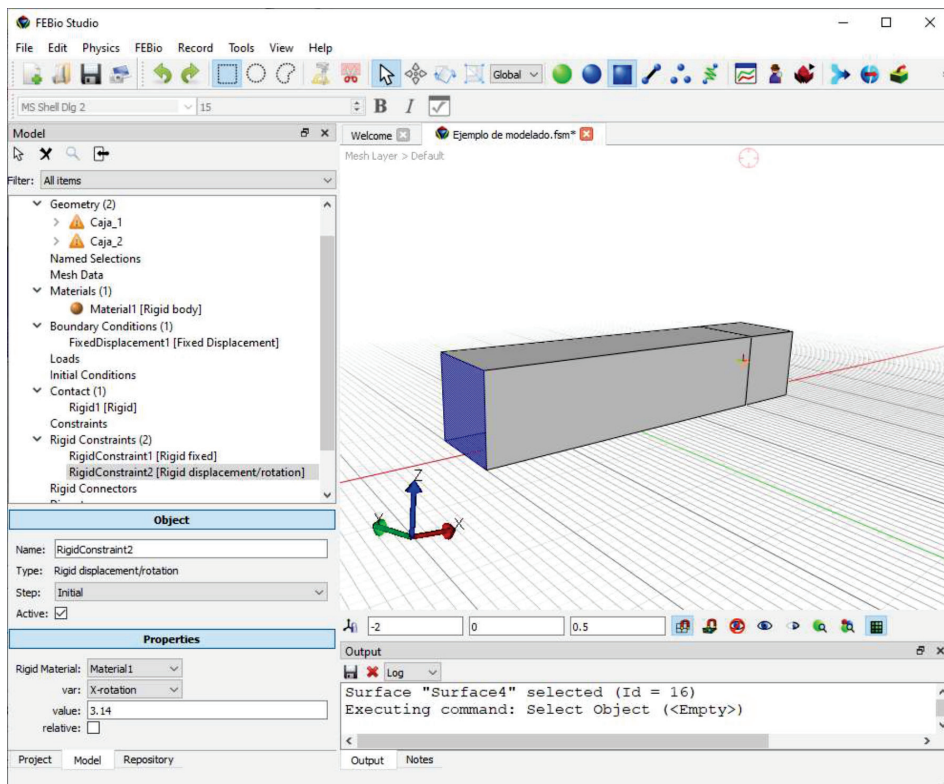
Fuente: elaboración propia.

B.2.2. Cargas y condiciones de frontera

Las cargas se definen por fuerzas, presiones, fuerzas de inercia (como la gravedad) y desplazamientos especificados, todos aplicados al modelo. Las cargas se pueden aplicar a los nodos mediante fuerzas y momentos concentrados y se pueden distribuir entre los elementos como: cargas superficiales, cargas de cuerpo, cargas de inercia u otras cargas acopladas como deformaciones térmicas.

Por otro lado, las condiciones de frontera son los valores conocidos de los grados de libertad en la frontera. En análisis estructural, los grados de libertad son los desplazamientos y las rotaciones. Restringir diferentes conjuntos de grados de libertad da como resultado la aplicación de diferentes condiciones de frontera. La primera condición de frontera es asignar a la cara izquierda de la caja 1 un desplazamiento fijo en los grados de libertad (x, y, z) . Para esto, se selecciona un desplazamiento fijo en el menú *Physics* $\rightarrow$ *Add Boundary Condition*. La condición de frontera ahora será visible en el Visor de modelos en cuya parte inferior aparecen visibles varios paneles que permiten personalizar esta condición de frontera. En el panel Propiedades se deben marcar todas las casillas de desplazamiento, lo que restringe los grados de libertad en (x, y, z) . A continuación, se debe aplicar esta restricción a la cara izquierda de la caja 1.

Figura B.4. Configuración de las cargas y condiciones de frontera



Fuente: elaboración propia.

El cuerpo rígido inicialmente no tiene restricciones, para agregar una se va al menú *Physics->Add Rigid Constraint*. En el cuadro de diálogo que aparece se selecciona el material rígido y la opción de desplazamiento/rotación fija. Después de hacer clic en Aceptar, las restricciones se mostrarán en el Visor de modelos. En la lista de propiedades, se marcan todas las casillas excepto la casilla de rotación x, a la cual se la aplicará una restricción diferente. Se selecciona de la lista que aparece en el menú *Physics->Add Rigid Constraint* la opción *Prescribe Displacement / Rotation*. En la lista de propiedades, se selecciona el grado de libertad de rotación x ingresando el valor 3,14 en el parámetro de *valor*. La configuración del modelo por EF ahora debe parecerse al de la figura B.4.

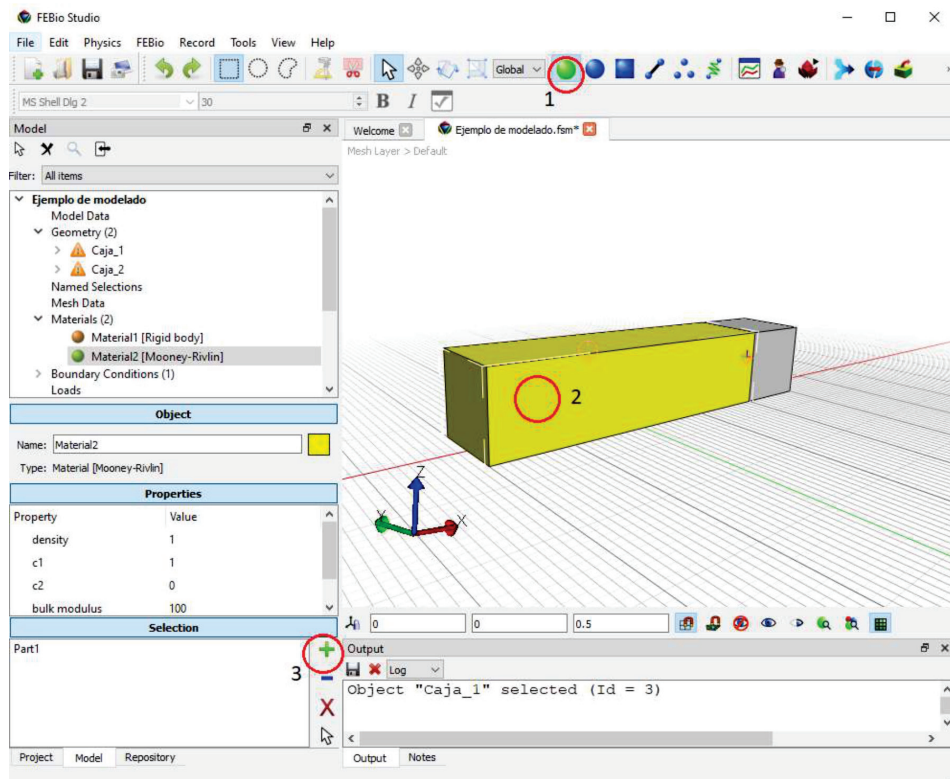
B.2.3. Propiedades de material

Todas las partes del modelo por EF deben estar asociadas a un material. Dependiendo del tipo de análisis que se va a realizar, las propiedades del material pueden ser lineales (análisis elástico lineal) o no lineales (por ejemplo, análisis de mecánica de daños), isotrópicas u ortotrópicas, constantes o dependientes de la temperatura. Introducir las propiedades correctas de los materiales es uno de los aspectos más importantes de un análisis exitoso de materiales. Otras propiedades mecánicas, como los coeficientes de resistencia, la densidad y la expansión térmica son opcionales y su definición depende de los objetivos del análisis.

Para agregar un material al modelo en FEBio Studio, se abre el Explorador de materiales seleccionando el menú *Physics/Add Material*. También se puede abrir el Explorador de materiales haciendo clic con el botón derecho en el elemento Materiales del Visor de modelos y seleccionando Agregar material en el menú emergente. Es bueno recordar que ya se ha creado un material relacionado con el cuerpo rígido, ahora se crea el segundo material que se asignará a la caja 1. En la opción Categoría, seleccione *uncoupled elastic*. Luego, de la lista de modelos de materiales disponibles, se selecciona el material *Mooney-Rivlin* y se hace clic en Aceptar. El material recién agregado ahora debería estar resaltado en el Visor de modelos y las propiedades del material ahora se muestran en el panel de Propiedades, ubicado debajo del Visor de modelos. Luego, se establecen las siguientes propiedades del material: *densidad=1*, *c1=1*, *c2=0*, *bulkmodulus=100*.

En la barra de herramientas principal, ubicada en la parte superior de la interfaz de usuario (figura B.1), se elige el botón de selección de objetos (una bola verde) como se muestra en la figura B.5 con el número 1. Luego, en la vista de gráficos se selecciona la caja 1, marcada con el número 2 en la figura B.5. Finalmente, se asigna al material *Mooney-Rivlin* haciendo clic en el botón con el símbolo $+$ en el cuadro de selección del material con la casilla elegida en la Vista gráfica, número 3 en la figura B.5. Ahora, se selecciona la caja 2 y se asigna, haciendo clic en el botón con el símbolo $+$ en el cuadro de selección, al material del cuerpo rígido.

Figura B.5. Asignación de materiales



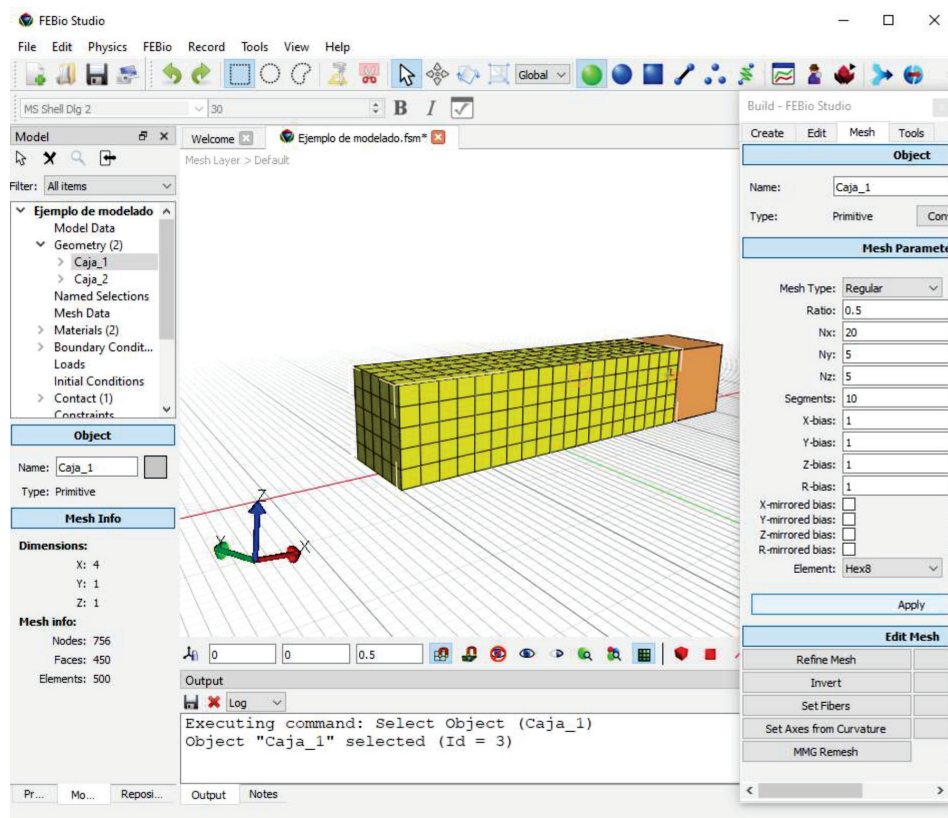
Fuente: elaboración propia.

B.2.4. Mallado y tipo de elemento

A continuación, se debe mallar el modelo, lo que permite obtener información de todos los elementos, su posición y la conectividad de cada uno.

La información de conectividad permite al programa ensamblar la matriz de rigidez del elemento y el vector de fuerza equivalente del elemento para obtener las ecuaciones de equilibrio global. Con base en la geometría de la superficie de los objetos (subdominios), se genera una malla de volumen que debe ser continua y no superpuesta, y con una buena calidad de elemento. En FEBio Studio las mallas de volumen generalmente consisten en tetraedros, hexahedros o una mezcla de ambos tipos de elementos, conocidos como cuadrículas de volúmenes híbridos.

Figura B.6. Configuración de proceso de mallado



Fuente: elaboración propia.

En FEBio Studio se selecciona la caja 1 y luego se selecciona la pestaña *mesh* en el panel de construcción (figura B.6). El panel *mesh* muestra todos los parámetros que afectarán el algoritmo de mallado. Se establecen los siguientes

parámetros, dejando todos los demás en sus valores predeterminados: $N_x = 20$, $N_y = 5$, $N_z = 5$. Antes de aplicar el mallado, se debe elegir el tipo de elemento, que para este caso es Hex8. Este proceso se repite para la caja 2, configurando los parámetros de la siguiente manera: $N_x = 1$, $N_y = 1$, $N_z = 1$. En el último caso, se crea efectivamente una malla de un solo elemento para el cuerpo rígido. El modelo completo ahora debería parecerse a la figura B.6.

Es importante señalar que la interfaz rígida le permite conectar un cuerpo rígido a una malla deformable no conforme. Esto significa que los nodos del cuerpo rígido no necesariamente coinciden con los nodos de la malla deformable. Esto es conveniente, ya que permite combinar los dos objetos por separado sin preocuparse por la continuidad en la interfaz.

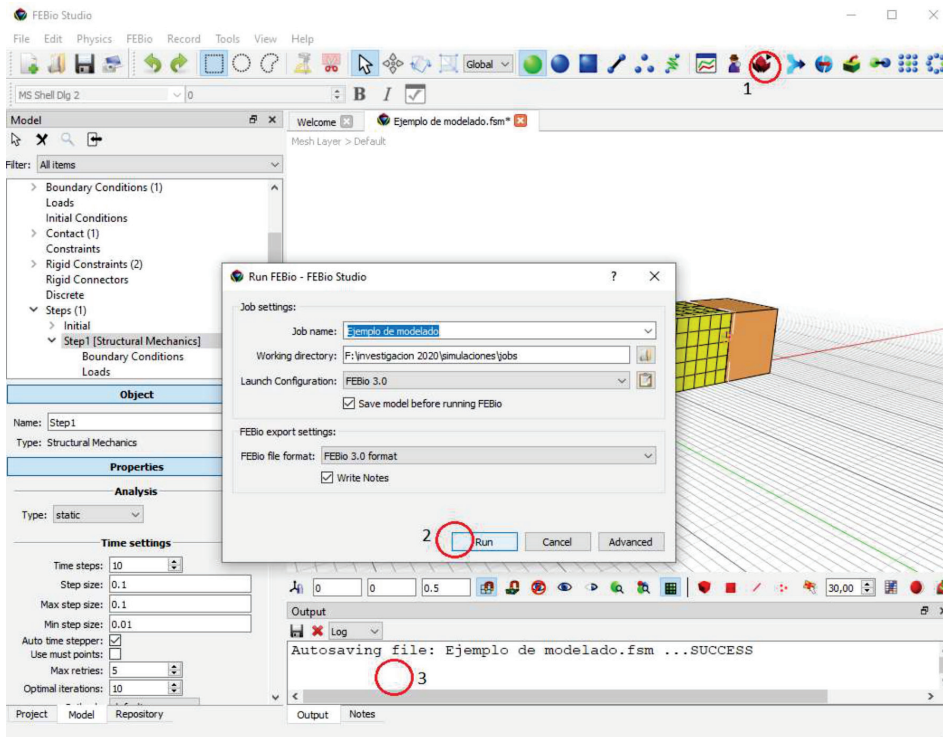
Dado que en FEBio Studio las condiciones de frontera, las cargas y las propiedades del material se pueden asignar al modelo sólido antes de mallar, se puede volver a mallar sin perder, o tener que quitar, las cargas y las condiciones de frontera. Sin embargo, los modelos se deben mallar antes de la fase de procesamiento.

B.2.5. Paso de análisis

Finalmente, se necesita crear un paso de análisis para definir qué tipo de análisis se quiere hacer. Se abre el menú *Physics/Add Analysis Step*, seleccionando el análisis de Mecánica estructural de entre los diferentes análisis disponibles. Las propiedades del paso de análisis ahora se muestran en el Visor de modelos (figura B.7). La configuración predeterminada es lo suficientemente buena por lo que no es necesario cambiar ningún parámetro.

Hay que tener en cuenta que en el Visor de modelos aparecen dos pasos en la lista: el paso que se acaba de crear y un paso llamado Inicial. El paso inicial no es un paso de análisis propiamente dicho ya que no se puede establecer la configuración del paso de tiempo. En su lugar, se utiliza para definir las condiciones de frontera permanentes, es decir, las condiciones de frontera que se propagarán a través de todos los pasos del análisis.

Figura B.7. Configuración del paso de análisis



Fuente: elaboración propia.

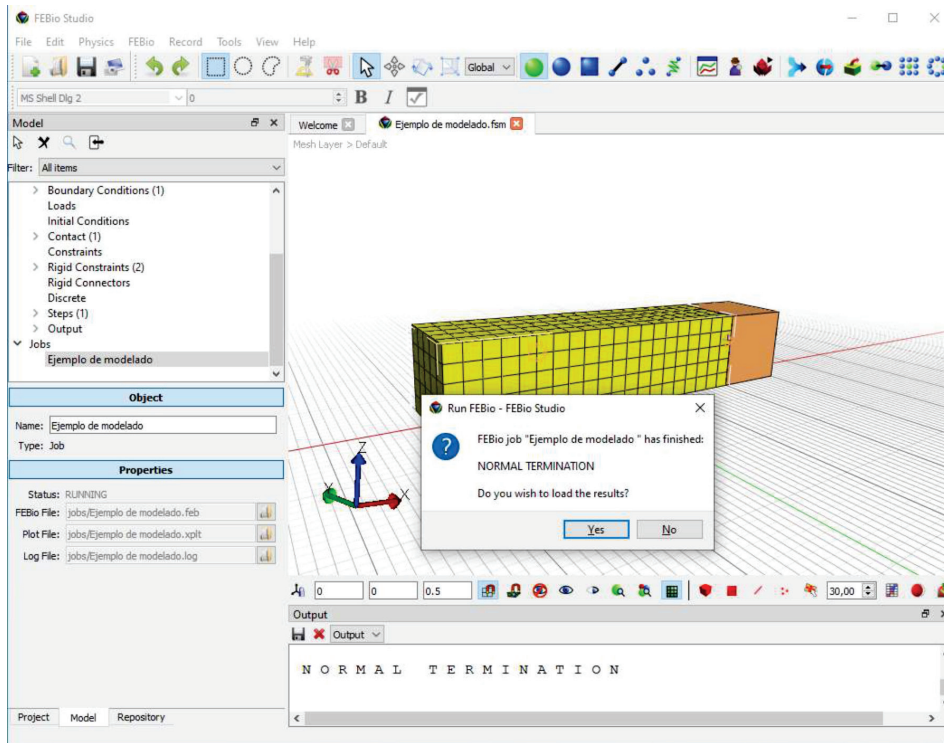
B.3. Procesamiento o fase de solución

En este punto, es una buena idea guardar el modelo. Se debe seleccionar el menú Archivo-> Guardar para guardar el modelo. En esta fase, el procesador de FEBio resuelve un conjunto simultáneo de ecuaciones que genera el método de EF. Por lo general, la solución primaria se obtiene resolviendo los valores de los grados de libertad de los nodos, es decir, los desplazamientos y las rotaciones. Luego se calculan los resultados derivados en los puntos de integración como tensiones y deformaciones.

El FEBio es un procesador de elementos finitos no lineal implícito que está diseñado especialmente para aplicaciones en biomecánica. Ofrece análisis, modelos constitutivos y condiciones de frontera que son relevantes para este campo en particular. Se puede ejecutar el procesador de FEBio desde el

entorno gráfico de FEBio Studio seleccionando el menú *FEBio->Run FEBio*, o desde la barra de herramientas principal haciendo clic en el botón correspondiente, como se muestra en la figura B.8 con el número 1.

Figura B.8. Ejecución de la simulación



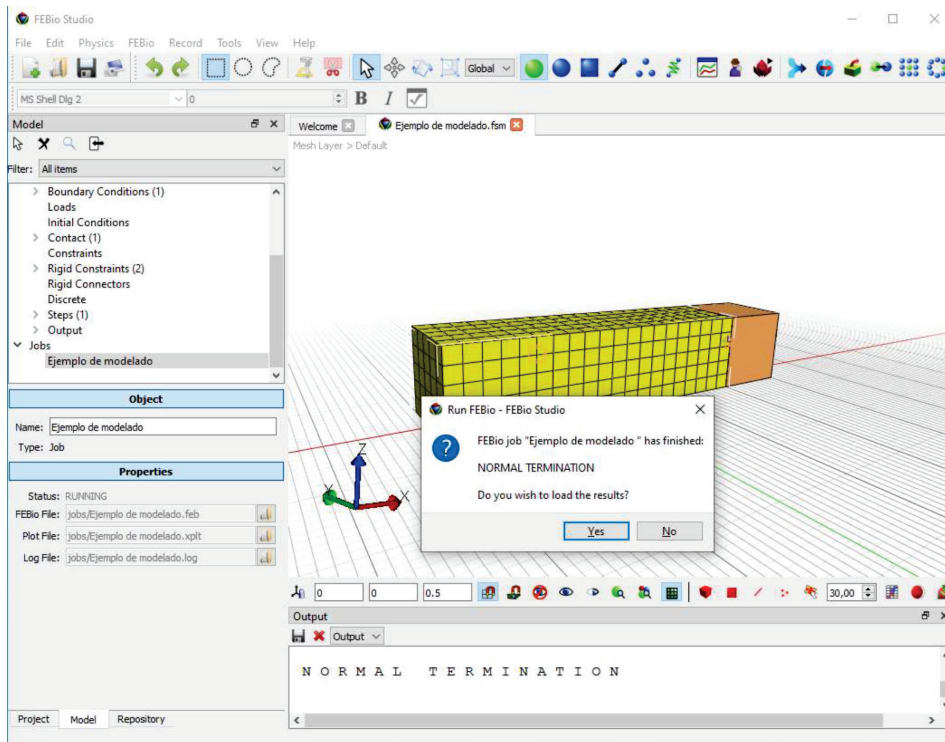
Fuente: elaboración propia.

El cuadro de diálogo de Ejecutar de FEBio, marcado con el número 2 en la figura B.8, permite configurar diferentes campos relacionados con la ubicación y el formato de los archivos de la simulación. Al presionar *Run*, se exportará el proyecto actual a la ubicación especificada, luego se llamará a FEBio. La salida de FEBio se mostrará en el panel de registro, marcado con el número 3 en la figura B.8.

Al completar el proceso de simulación, el programa guardará los resultados en un archivo *xplt*, cuyos datos se pueden visualizar en cualquier momento. Así mismo, se puede invocar el módulo de visualización directamente desde

la ventana de terminación de la simulación, haciendo clic en la opción Sí. Este último paso cambiará de la zona de trabajo al módulo de visualización, mostrando una visualización preliminar, como se observa en la figura B.9.

Figura B.9. Terminación exitosa del proceso de simulación



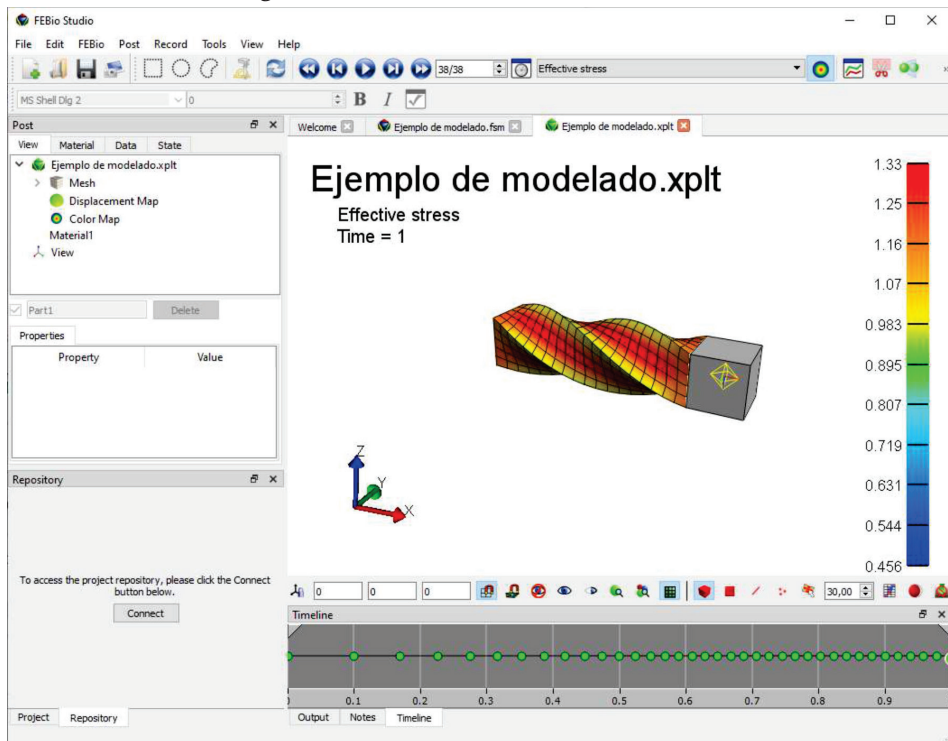
Fuente: elaboración propia.

B.4. Posprocesamiento y visualización

Una vez que se obtiene la solución al modelo por EF, el posprocesamiento se puede utilizar para analizar los resultados, los cuales se pueden revisar gráficamente (figura B.10) o listando los valores numéricamente. Dado que un modelo generalmente contiene una cantidad considerable de resultados, puede ser mejor utilizar herramientas gráficas. FEBio Studio incluye cálculos derivados, como estimación de errores, combinaciones de casos de carga u operaciones de trayectoria.

La figura B.10 muestra la tensión de Von Mises sobre el modelo deformado por la torsión de la barra. El valor máximo de tensión es de 1,33 MPa en el tiempo 1.

Figura B.10. Visualización de los resultados



Fuente: elaboración propia.

Sobre el autor

Miller Gómez Mora

Profesor titular de carrera adscrito a la Facultad Tecnológica de la Universidad Distrital Francisco José de Caldas. Director del grupo de investigación en Simulaciones y Sistemas Telemáticos (IMAGINET). Ingeniero de sistemas (Universidad Técnica de Georgia), especialista en Teleinformática y magíster en Ciencias de la Información y las Comunicaciones (Universidad Distrital Francisco José de Caldas), Ph. D. en Ingeniería (Universidad Javeriana). Realiza investigación en computación gráfica y realidad virtual, modelado y fabricación 3D y sistemas telemáticos.

Correo electrónico: mgomezm@udistrital.edu.co

ORCID: <https://orcid.org/0000-0002-4310-8893>

ES



Este libro se terminó de imprimir en noviembre de 2024
en XPRESS ESTUDIO GRÁFICO Y DIGITAL S. A. S.,
Bogotá, Colombia.