



Trabajo Fin de Máster

Estudio de la viabilidad de la segmentación de placas
de ateroma vulnerables en coronarias a partir del
campo de desplazamientos compatibles con imágenes
IVUS

On the potential of IVUS approach to vulnerable
atherosclerotic plaque reconstruction based on a
segmentation procedure using strain measurements

Autor

Álvaro Tomás Latorre Molins

Directoras

Estefanía Peña Baquedano
Myriam Cilla Hernández

Máster Universitario en Ingeniería Biomédica

Agradecimientos

Me gustaría agradecer a mis directoras Estefanía y Myriam el tiempo que han dedicado al proyecto y todo lo que me han enseñado. También reconocer la ayuda del profesor Miguel Ángel Martínez Barca, de la Universidad de Zaragoza, por su implicación en este trabajo. Por último agradecer al profesor Jacques Ohayon de la Universidad de Savoie Mont Blanc de Chambéry, Francia, cuyos conocimientos y guía han sido indispensables.

Resumen

Las enfermedades cardiovasculares ocupan el primer puesto en causas de mortalidad a nivel mundial y es la aterosclerosis una de las principales causantes de estas patologías. La aterosclerosis consiste en la deposición de colesterol en la pared arterial, esta deposición forma un núcleo lipídico que queda recubierto por una capa fibrótica que lo separa del torrente sanguíneo, dando lugar a la llamada placa de ateroma. Su ruptura puede conllevar ataques al corazón o ictus entre otras afecciones, por lo que su detección es vital para un diagnóstico precoz. Actualmente, se utiliza la ecografía intravascular (IVUS) para visualizar la pared arterial, detectar placas de ateroma y además permite obtener información sobre las deformaciones que sufre, sin embargo, diferenciar los diferentes tejidos mediante esta tecnología es muy complicado, por lo que el objetivo de este proyecto es el de segmentar, de forma automática, los tejidos de una arteria coronaria con placa de ateroma a partir de las deformaciones proporcionadas por un IVUS. Esto permitiría una rápida visualización en clínica además de abrir la posibilidad a utilizar la segmentación para crear modelos computacionales con geometrías de paciente específico. Para elaborar un método de segmentación se construyen modelos de elementos finitos con diferentes geometrías. El resultado de las simulaciones de los modelos proporciona las coordenadas y desplazamientos nodales. Se toman los resultados en dos instantes de tiempo próximos, con un incremento de presión entre ellos de aproximadamente 5mmHg, simulando así los datos que proporcionarían dos imágenes consecutivas tomadas por un IVUS. A partir de los desplazamientos es posible calcular las deformaciones y obtener diferentes variables como deformaciones principales, cilíndricas, equivalentes, etc. Aunque los modelos de elementos finitos son tridimensionales es posible tratar los datos como 2D y trabajar con una única sección, bajo la hipótesis de deformación plana (tal y como opera el IVUS) o se puede trabajar como modelo 3D, donde se analiza una pequeña rebanada. En ambos casos el proceso de segmentado consiste en combinar la representación de diferentes variables e introducir la resultante en un proceso *Watershed* que permita adquirir las diferentes partes de la arteria. Tras el segmentado se obtiene el núcleo lipídico y el lumen, permitiendo medir el espesor de la capa fibrótica o el área del núcleo. La precisión de las medidas depende del espesor de la capa fibrótica y aumentará cuanto mayor sea dicho valor. Además, dependiendo de las variables a utilizar es posible segmentar diferentes tejidos como la capa fibrótica o el tejido sano. Estos datos son de vital importancia a nivel clínico para poder clasificar una placa de ateroma como vulnerable y proporcionar un diagnóstico precoz.

Palabras Clave: Aterosclerosis, núcleo lipídico, capa fibrótica, lumen, IVUS, deformación, segmentación

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Estudios Previos	1
1.3. Objetivo	4
1.4. Metodología	4
1.5. Estructura de la Memoria	4
2. Modelo de Elementos Finitos de la Arteria Coronaria	5
2.1. Geometrías	5
2.2. Materiales	7
2.3. Ensamblado	8
2.4. Cargas y Condiciones de Contorno	9
2.5. Mallado	10
3. Procesado de Datos y Método de Segmentación	12
3.1. Post-Procesado de Datos	12
3.2. Cálculo de la Deformación y otras Variables en 2D	14
3.3. Proceso y Cálculo en 3D	17
3.4. Proceso de Segmentación	19
3.5. Cuantificación de los Errores	23
4. Resultados	24
4.1. Geometrías Idealizadas. Análisis 2D	24
4.2. Geometrías No Idealizadas	28
4.3. Geometrías IVUS	29
4.4. Geometría Idealizada. Análisis 3D	30
5. Conclusiones	31
5.1. Limitaciones	32
5.2. Líneas Futuras	32
6. Bibliografía	33
Anexo I: Materiales	37
Modelos Constitutivos	37
Modelo 1	38
Modelo 2	39
Modelo 3	39
Modelo 4	40
Modelo 5	40
Modelo 6	41
Comparación de Resultados	42
Anexo II: Metodología desarrollada en Matlab	44
Lectura de Datos	44
Cálculo de la Deformación	47
Programa Principal	49
Sección 1: Toma de datos	49
Sección 2: Obtener valores de coordenadas y desplazamientos	49
Sección 3: Mallado	49
Sección 4: Crear modelo <i>PDE</i>	50
Sección 5: Cálculo de Gradientes	50
Sección 6: Calcular Deformaciones	51

Sección 7: Gradiante de las Deformaciones	52
Sección 8: Cálculo de las Deformaciones Equivalentes	52
Sección 9: Cálculo de las Deformaciones Cilíndricas	53
Sección 10: Cálculo del Error entre Abaqus CAE y Matlab	53
Sección 11: Representación de las Variables	53
Anexo III: Validación del Método	54
Paso 1	54
Paso 2	59
Paso 3	61
Anexo IV: Resultados-Cálculo de Variables	64
Resultados del Modelo 2D	64
Resultados del Modelo 3D	71
Anexo V: Proceso de Segmentado	77
Segmentación real de las partes	77
Tratamiento de la Imagen	81
Método <i>Watershed</i>	82
Obtención de las partes	83
Ajuste de Escala	85

Índice de figuras

1.1. Proceso de formación y ruptura de la placa de ateroma en coronaria [3].	1
1.2. a) Phantom de PVA-C, b) Imagen IVUS-B segmentada manualmente, c) Mapa de deformación tangencial calculad en elementos finitos, d) Deformación tangencial estimado por LSME [9].	2
1.3. a) Deformación radial estimada a partir de una palpografía (dominio azul) b) Comparación entre el Anisotropic Index exacto del modelo de elementos finitos y el estimado [7].	3
1.4. Proceso iterativo para obtener el modulograma [kPa] de una placa de ateroma a partir del campo de deformaciones y un modelo de elementos finitos, donde 'n' es el número de iteraciones [15].	3
1.5. Pasos realizados hasta llegar al objetivo final (segmentación de la geometría).	4
2.1. Vista longitudinal del modelo, con longitud de placa (L Placa) y núcleo (L Nucleo) y espesor de la capa fibrótica (e Cap) señalizados [31].	5
2.2. Modelo geométrico 3D realizado en Solid Works. Placa de 13 mm, núcleo lipídico de 6.5 mm y capa fibrótica de 65 μm	6
2.3. Vista frontal de la sección de máxima estenosis de los modelos idealizados con diferente espesor de la capa fibrótica: a) 65 μm b) 150 μm y c) 300 μm	6
2.4. Modelos geométricos 3D realizados en Solid Works con núcleo lipídicos con formas no idealizadas, a) forma 1 y b) forma 2.	7
2.5. a) IVUS real [34], b) Segmentado manual de las diferentes partes [34], c) Dibujado del contorno en Abaqus CAE, c) Modelo 2D de la placa de ateroma y núcleo lipídico en Abaqus CAE.	7
2.6. Ensamblado de las diferentes partes de una placa de 13 mm de longitud y un núcleo de 6.5 mm; a) Adventicia, b) Adventicia y Media, c) Adventicia, Media y Placa, d) Adventicia, Media, Placa y Núcleo lipídico.	9
2.7. Partición para poder considerar la íntima sana y enferma.	9
2.8. Imposición de las condiciones de contorno de simetría a) y b), contacto con el corazón en c) y presión de 18.7 kPa en d).	10
2.9. Mallado adaptado, a)Malla fina en la zona de máxima estenosis, b)En el círculo rojo se observa la malla muy fina en la zona de la capa fibrótica c) Malla gruesa en el resto del modelo.	11
3.1. Coordenada 'Z' [mm] representada en el modelo con 65 micras de espesor. El centro de coordenadas se ha definido en el centro del lumen simulando el catéter.	12
3.2. a) Malla inicial por triangulación de Delaunay, b) Malla procesada, c) Modelo PDE.	13
3.3. Desplazamientos [mm] en X a) y en Y b), Gradientes desplazamientos en X c) y en Y d), Módulo del Gradiante de desplazamientos en X e) y en Y f).	14
3.4. a) Malla inicial, b) Malla depurada, c) Modelo PDE.	17
3.5. a) Representación de $ \nabla \varepsilon_y $ inicial, b) Recorte de la representación, c) Imagen Simétrica, d) Unión.	19
3.6. Diferentes límites para la representación de $ \nabla \varepsilon_y $: a) Límite superior de 0.02 e inferior de -0.02, b) límite superior de 0.008 e inferior de 0.	20
3.7. Efecto del disco de segmentado de tamaño 15 píxeles antes a) y después b) sobre la representación de $ \nabla \varepsilon_y $	20
3.8. Núcleo lipídico a) Tras el segmentado, b) Despues del suavizado.	21
3.9. a) Diámetro medido [pixeles] en la imagen de Matlab, b) Diámetro medido [mm] en el modelo de Abaqus CAE donde se representa las coordenadas en el eje Y.	21
3.10. a) Vista 3D del modelo, b) Vista frontal del $ \nabla \varepsilon_x $	22
3.11. a) Diámetro del lumen medido en pixeles en Matlab, b) Diámetro del lumen medido en mm en Abaqus.	22

3.12. Diferentes partes de la arteria coronaria con placa de ateroma con espesor de 150 μm . En azul aparece el tejido sano (Adventicia+Media), en amarillo la Placa o capa fibrótica, en rojo el núcleo lipídico y en verde el lumen.	23
3.13. Superposición del núcleo lipídico segmentado tras combinar $ \nabla\epsilon_y $ con $ \nabla\epsilon_{xy} $ y el núcleo real. En blanco se observa la unión entre el núcleo real y el segmentado, el morado el núcleo real no segmentado y en verde el núcleo segmentado no real.	23
4.1. Proceso de segmentado en la geometría idealizada de espesor de 150 μm : a) $ \nabla\epsilon_x $, b) $ \nabla\epsilon_y $, c) Unión de las anteriores y partes segmentadas, d) Núcleo lipídico segmentado, e) Núcleo lipídico segmentado tras el proceso de suavizado y f) Comparación entre núcleo lipídico segmentado (blanco) y real (morado).	24
4.2. Medida [mm] automática del espesor de la capa fibrótica.	25
4.3. Tabla de segmentación de 131 combinaciones posibles en la geometría idealizada de 65 μm de espesor. Se presenta el error de medida del espesor de la capa fibrótica.	25
4.4. Tabla de segmentación de 131 combinaciones posibles en la geometría idealizada de 65 μm de espesor. Se presentan los índices1/indice2.	26
4.5. Tabla de segmentación de 131 combinaciones posibles en la geometría idealizada de 150 μm de espesor. Se presenta el error de medida del espesor de la capa fibrótica.	26
4.6. Tabla de segmentación de 131 combinaciones posibles en la geometría idealizada de 150 μm de espesor. Se presentan los índices1/indice2.	26
4.7. Tabla de segmentación de 131 combinaciones posibles en la geometría idealizada de 300 μm de espesor. Se presenta el error de medida del espesor de la capa fibrótica.	26
4.8. Tabla de segmentación de 131 combinaciones posibles en la geometría idealizada de 300 μm de espesor. Se presentan los índices1/indice2.	27
4.9. Segmentación total de la geometría con espesor de 65 μm , a) Tejido sano, b) Capa fibrótica, c) Placa de ateroma y d) Lumen.	27
4.10. Segmentación total de la geometría con espesor de 150 μm , a) Tejido sano, b) Capa fibrótica, c) Placa de ateroma y d) Lumen.	28
4.11. Segmentación total de la geometría con espesor de 300 μm , a) Tejido sano, b) Capa fibrótica, c) Placa de ateroma y d) Lumen.	28
4.12. a) Segmentación de la capa fibrótica y tejido sano, b) Segmentación de capa fibrótica y c) Segmentación final del tejido sano tras restar las dos anteriores.	28
4.13. Proceso de segmentado en la geometría no idealizada (forma 1): a) dW , b) $ \nabla\epsilon_y $, c) Unión de las anteriores y partes segmentadas, d) Núcleo lipídico segmentado, e) Núcleo lipídico segmentado tras el proceso de suavizado y f) Comparación entre núcleo lipídico segmentado (blanco) y real (morado).	29
4.14. Proceso de segmentado en la geometría no idealizada (forma 2): a) $ \nabla\epsilon_{vonMises} $, b) $ \nabla\epsilon_y $, c) Unión de las anteriores y partes segmentadas, d) Núcleo lipídico segmentado, e) Núcleo lipídico segmentado tras el proceso de suavizado y f) Comparación entre núcleo lipídico segmentado (blanco) y real (morado).	29
4.15. Proceso de segmentado en geometría reconstruida de IVUS real: a) dW , b) $ \nabla\epsilon_x $, c) Unión de las anteriores y partes segmentadas, d) Núcleo lipídico segmentado, e) Núcleo lipídico segmentado y suavizado y f) Comparación entre núcleo lipídico segmentado (blanco) y real (morado).	30
4.16. Tabla de segmentación de combinaciones tridimensional en la geometría idealizada de 150 μm de espesor en el análisis 3D. Se presenta el error de medida del espesor de la capa fibrótica.	30
4.17. Tabla de segmentación de combinaciones tridimensional en la geometría idealizada de 150 μm de espesor en el análisis 3D. Se presentan los índices1/indice2.	30
6.1. Curvas de comportamiento de los materiales del modelo 1.	38
6.2. Resultado del Modelo1. Tensión Principal Máxima [kPa].	38
6.3. Resultado del Modelo 2. Tensión Principal Máxima [kPa].	39
6.4. Resultado del Modelo 3. Tensión Principal Máxima [kPa].	40
6.5. Resultado del Modelo 4. Tensión Principal Máxima [kPa].	40
6.6. Resultado del Modelo 5. Tensión Principal Máxima [kPa].	41
6.7. Resultado del Modelo 6. Tensión Principal Máxima [kPa].	42

6.8. Curvas de comportamiento de las íntimas analizadas en los modelos. σ [kPa].	42
6.9. Curvas de comportamiento de las capas fibróticas analizadas en los modelos. σ [kPa].	43
6.10. a)Modelo 2D realizado en Abaqus CAE, b) Fichero de datos.	54
6.11. Mallado Delaunay.	55
6.12. a) Numeración de los nodos y b) numeración de los elementos.	55
6.13. Ejemplo del cálculo de la deformación nodal a partir de la deformación elemental.	56
6.14. Comparación entre las deformaciones nodales calculadas mediante el la suma sin ponderación y la ponderación por área.	56
6.15. Comparación entre deformaciones cartesianas y principales calculadas en Abaqus CAE y en el método desarrollado en Matlab.	57
6.16. Deformaciones principales y equivalentes de Tresca y von Mises.	58
6.17. a) Modelo 2D de Abaqus CAE, b) Módulo de desplazamientos en los resultados.	58
6.18. Comparación entre deformaciones cilíndricas calculadas en Abaqus y Matlab.	59
6.19. a) Modelo 3D de Abaqus CAE, b) Malla de Delaunay de una sección 2D en Matlab, c) Depurado de los elementos de la malla.	59
6.20. Comparación entre deformaciones cilíndricas calculadas en Abaqus y Matlab.	60
6.21. Comparación entre deformaciones principales calculadas en Abaqus y Matlab.	60
6.22. a) Modelo 3D en Abaqus CAE, b) Mallado con tetraedros en Matlab, c) Depuración de la malla.	61
6.23. Comparación entre deformaciones cartesianas de Abaqus y Matlab.	62
6.24. Comparación entre deformaciones cilíndricas de Abaqus y Matlab.	62
6.25. Comparación entre deformaciones principales de Abaqus y Matlab.	63
6.26. Geometría idealizada tratada como modelo 2D, a) Malla inicial, b) Malla procesada y c) Modelo PDE.	64
6.27. Desplazamiento en mm en, a) el eje X , b) el eje Y , x) módulo de los desplazamientos.	64
6.28. Gradiente (vector) de los desplazamientos en a) el eje X , b) el eje Y	65
6.29. Módulo del gradiente de los desplazamientos, a) eje X , b) eje Y , c) Módulo del desplazamiento en X e Y	65
6.30. Deformaciones cartesianas calculadas con el método sobre la geometría de espesor de 150 μm bajo la hipótesis de deformación plana. a) ε_x , b) ε_y , c) ε_{xy}	66
6.31. Deformaciones obtenidas de Abaqus CAE en la geometría de espesor de 150 μm . a) ε_x , b) ε_y , c) ε_{xy}	66
6.32. Error relativo (%) entre las deformaciones calculadas en Matlab y las obtenidas en Abaqus CAE. Geometría de espesor de 150 μm . a) Error relativo de ε_x , b) Error relativo de ε_y , c) Error relativo de ε_{xy}	67
6.33. Módulo del gradiente de las deformaciones cartesianas. Geometría de espesor de 150 μm . a) $ \nabla\varepsilon_x $, b) $M \nabla\varepsilon_y $, c) $ \nabla\varepsilon_{xy} $	67
6.34. Deformaciones principales calculadas con el método sobre la geometría de espesor de 150 μm bajo la hipótesis de deformación plana. a) ε_I , b) ε_{II} , c) ε_{III}	68
6.35. Deformaciones principales obtenidas de Abaqus CAE en la geometría de espesor de 150 μm . a) ε_I , b) ε_{II} , c) ε_{III}	68
6.36. Error relativo (%) entre las deformaciones principales calculadas en Matlab y las obtenidas en Abaqus CAE. Geometría de espesor de 150 μm . a) Error relativo de ε_I , b) Error relativo de ε_{II} , c) Error relativo de ε_{III}	68
6.37. Módulo del gradiente de las deformaciones principales. Geometría de espesor de 150 μm . a) $ \nabla\varepsilon_I $, b) $ \nabla\varepsilon_{II} $, c) $ \nabla\varepsilon_{III} $	69
6.38. Deformaciones equivalentes calculadas con el método sobre la geometría de espesor de 150 μm bajo la hipótesis de deformación plana. a) $\varepsilon_{vonMises}$, b) ε_{Tresca} , c) Índice de Anisotropía (FA).	69
6.39. Módulo del gradiente de las deformaciones equivalentes. Geometría de espesor de 150 μm . a) $M \nabla\varepsilon_{vonMises} $, b) $ \nabla\varepsilon_{Tresca} $, c) $ \nabla FA $	69
6.40. Deformaciones en coordenadas cilíndricas calculadas con el método sobre la geometría de espesor de 150 μm bajo la hipótesis de deformación plana. a) ε_R , b) ε_θ , c) $\varepsilon_{R\theta}$	70
6.41. Módulo del gradiente de las deformaciones en coordenadas cilíndricas. Geometría de espesor de 150 μm . a) $ \nabla\varepsilon_R $, b) $ \nabla\varepsilon_\theta $, c) $ \nabla\varepsilon_{R\theta} $	70
6.42. a)Representación de la variable dW , b) Representación de $ \nabla dW $	71

6.43. Geometría idealizada tratada como modelo 3D, a) Malla inicial, b) Malla procesada y c) Modelo PDE.	71
6.44. Desplazamiento en mm en, a) u_x , b) u_y , c) u_z	72
6.45. Gradiente de los desplazamiento en mm, a) ∇u_x , b) ∇u_y , x) ∇u_z	72
6.46. Módulo del gradiente de los desplazamiento en mm, a) $ \nabla u_x $, b) $ \nabla u_y $, x) $ \nabla u_z $	72
6.47. Deformaciones en coordenadas cartesianas, a) ε_x , b) ε_y , c) ε_z , d) ε_{xy} , e) ε_{xz} , f) ε_{yz}	73
6.48. Error relativo entre las deformaciones en coordenadas cartesianas calculadas en Matlab y las deformaciones obtenidas en Abaqus CAE, a) Error Relativo en ε_x , b) Error Relativo en ε_y , c) Error Relativo en ε_z , d) Error Relativo en ε_{xy} , e) Error Relativo en ε_{xz} , f) Error Relativo en ε_{yz}	73
6.49. Módulo del gradiente de la deformación en coordenadas cartesianas, a) $ \nabla \varepsilon_x $, b) $ \nabla \varepsilon_y $, c) $ \nabla \varepsilon_z $, d) $ \nabla \varepsilon_{xy} $, e) $ \nabla \varepsilon_{xz} $, f) $ \nabla \varepsilon_{yz} $	74
6.50. Deformaciones principales, a) ε_I , b) ε_{II} , c) ε_{III}	74
6.51. Error relativo entre las deformaciones principales calculadas en Matlab y las deformaciones obtenidas en Abaqus CAE, a) Error Relativo en ε_I , b) Error Relativo en ε_{II} , c) Error Relativo en ε_{III}	75
6.52. Módulo del gradiente de la deformación principal, a) $ \nabla \varepsilon_I $, b) $ \nabla \varepsilon_{II} $, c) $ \nabla \varepsilon_{III} $	75
6.53. Deformaciones en coordenadas cilíndricas, a) ε_R , b) ε_θ , c) $\varepsilon_{R\theta}$	75
6.54. Módulo del gradiente de la deformación principal, a) $ \nabla \varepsilon_R $, b) $ \nabla \varepsilon_\theta $, c) $ \nabla \varepsilon_{R\theta} $	76
6.55. Representación de la variable dW	76
6.56. Núcleo lipídico real de la geometría idealizada de espesor 150 micras.	78
6.57. Núcleo lipídico donde se muestran las tres posibilidades de los elementos de la malla. Un elemento entero dentro de la geometría, uno con solo dos nodos en el interior y otro con un único nodo dentro.	79
6.58. a) Segmentación real de la placa o capa fibrótica, b) Segmentación real del tejido sano (Adventicia y Media).	81
6.59. Proceso de segmentado Watershed, a) Variable 1: dW , b) Variable 2: $ \nabla \varepsilon_y $, c) Unión de ambas variables, d) Ajuste saturado de los colores, e) Gradiente de d, f) Implementación del disco de segmentado, g) Cierre de la geometría, h) Aplicación del Watershed e i) Resultado final de la segmentación.	82
6.60. Asignación de las partes segmentadas, a) Núcleo lipídico y b) Lumen.	84
6.61. Superposición del núcleo real y el segmentado a) tras el proceso de <i>Watershed</i> , b) tras aplicar un proceso de suavizado.	84
6.62. Medida automática (rojo) del espesor de la capa fibrótica en pixeles, medida real (azul) en píxeles.	85

1. Introducción

1.1. Contexto

Las **enfermedades cardiovasculares** ocupan el primer puesto de causas de mortalidad en todo el mundo con 17.9 millones de muertes cada año [1], lo que representa el 31% del total. El 85% de las muertes por dichas enfermedades se deben a ataques al corazón y eventos cerebrovasculares [1] y aunque sus causas son diversas, la **aterosclerosis** es la más común y supone más de un 50% de las muertes súbitas [2]. La enfermedad aterosclerótica es una patología que consiste en la deposición de lípidos en el interior de la pared arterial debido a un daño endotelial previo. El endotelio, que corresponde a la capa más interna, puede verse dañado por diferentes factores [3]: biológicos (alto colesterol, diabetes), ambientales (el tabaco aumenta la rigidez arterial, dietas hipercalóricas) y mecánicos (bifurcaciones en los vasos o grandes curvaturas provocan que el flujo sanguíneo cambie y se modifique las tensiones tangenciales sobre la pared arterial). Estos factores hacen que aumente la permeabilidad endotelial y es entonces cuando el colesterol circulante por la sangre es capaz de traspasar la pared y formar la placa de ateroma. Esta placa consiste en un núcleo de grasa oxidada y fagotizada por macrófagos y recubierto por una capa fibrótica. El crecimiento de la placa conlleva un engrosamiento y endurecimiento de la pared arterial que ralentiza el flujo sanguíneo [4]. Existen diferentes factores que permiten clasificar una placa como **vulnerable**, cuanto mayor sea el **núcleo lipídico** [5] y menor la capa que lo separa del torrente sanguíneo (espesor de la **capa fibrótica**) más vulnerable y propenso a la ruptura será. Además, es posible que la placa tenga un **crecimiento positivo** [6] sin ocluir el lumen, lo que haría que pasase desapercibida en una angiografía. Al romper la placa de ateroma se forma un trombo, que puede desencadenar un ataque al corazón, un ictus u otros eventos cardiovasculares. Por este motivo, la ruptura espontánea de la placa de ateroma se considera la mayor causa de síndromes coronarios agudos. En la *Figura 1.1* se observa la progresión de la aterosclerosis.

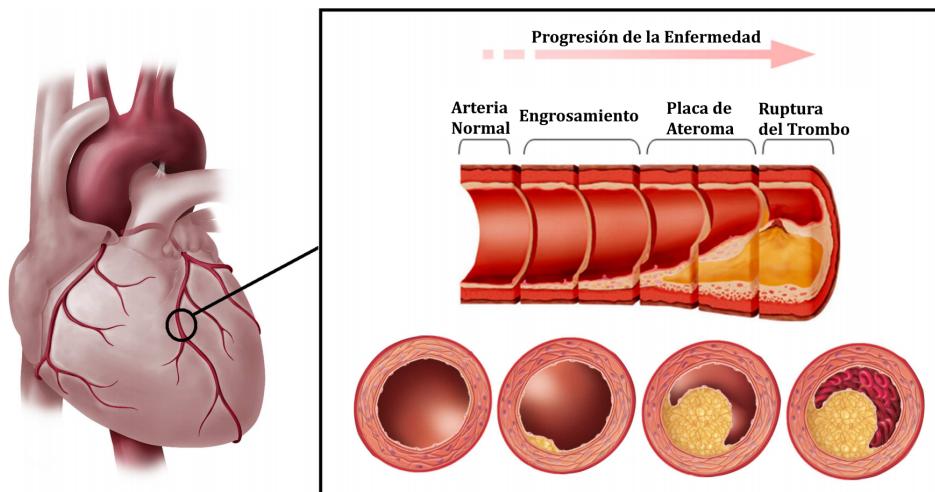


Figura 1.1: Proceso de formación y ruptura de la placa de ateroma en coronaria [3].

1.2. Estudios Previos

La caracterización morfológica y mecánica de la placa de ateroma es un paso esencial en el diagnóstico precoz de placas vulnerables [7]. Con el fin de diagnosticar estos eventos coronarios, han surgido diferentes técnicas de detección mediante imagen [7] como la **ecografía intravascular (IVUS)**, la tomografía de coherencia óptica (OCT), la resonancia magnética (MRI) y técnicas híbridas donde se combinan las anteriores [7]. El método más común para explorar las arterias coronarias es el IVUS, que permite obtener información de la luz arterial del vaso y la pared [8], y además, posibilita, mediante técnicas avanzadas, la visualización de las deformaciones en la pared arterial. Dentro de los intentos

para caracterizar las placas vulnerables se han empleado diferentes técnicas, como elastografías [9, 10, 11] y palpografías [12], cuyo objetivo es obtener mapas de elasticidad a partir de las deformaciones obtenidas mediante IVUS y/o MRI [7], ya que pequeñas variaciones en las propiedades elásticas en los tejidos provienen de ciertas patologías [7], además las diferencias de rigidez de la placa pueden cambiar las tensiones tangenciales [13] de la misma pudiendo provocar la ruptura de la placa [14].

Es habitual realizar un **modelo de elementos finitos** [7, 10, 15, 16, 17] basado en geometrías reales y poder comparar la técnica desarrollada con otras geometrías, histologías o ensayos in vitro con *phantoms* (discos de PVA que simulan la sección transversal arterial) [9, 16, 18]. En la gran mayoría de estos estudios utilizan dos imágenes consecutivas obtenidas de imágenes IVUS [7, 15], se segmentan ambas imágenes manualmente con ayuda de expertos, y toman la primera imagen como referencia [7, 15]. Introducen las geometrías a programas de elementos finitos y crean pequeñas rebanadas 3D con un pequeño espesor [7] o modelos 2D con hipótesis de deformación plana [10, 11] y se impone un incremento de presión entre 2.2 y 7.5 mmHg simulando la presión entre las imágenes tomadas por el IVUS [7, 15, 16]. Este incremento permite tratar el problema como elástico lineal con pequeñas deformaciones [7]. Es habitual simular los materiales como isótropos [17] y cuasi-incompresibles (coeficientes de Poisson $\nu \simeq 0.49$) para evitar problemas de cálculo [16].

Para desarrollar los mapas de elasticidad o modulogramas hay que poder seguir los puntos de la geometría entre las diferentes imágenes de IVUS [18], lo que es muy complicado debido al movimiento durante el ciclo cardiaco [18]. Una vez se obtienen los datos se utilizan diferentes métodos para obtener las deformaciones, como el *Lagrangian Speckle Model Estimator (LSME)* [19] que permite obtener las deformaciones tangenciales [18]. Para segmentar las partes de la placa de ateroma se utilizan diferentes métodos como el **watershed** [15, 17, 18] y se representan diferentes variables como las deformaciones tangenciales [9] que predicen el origen de la ruptura. Sin embargo, este tipo de variable depende de la colocación del catéter [11, 15], que debe situarse en el centro del lumen para la correcta representación; por este motivo se han utilizado otras variables, como las deformaciones principales [20] o von Mises [21], evitando así la dependencia del sistema de coordenadas según la posición del catéter [11]. En la *Figura 1.2* se muestra la comparación del mapa de deformaciones tangenciales [9] obtenidas por elementos finitos y por medio del método *LSME*.

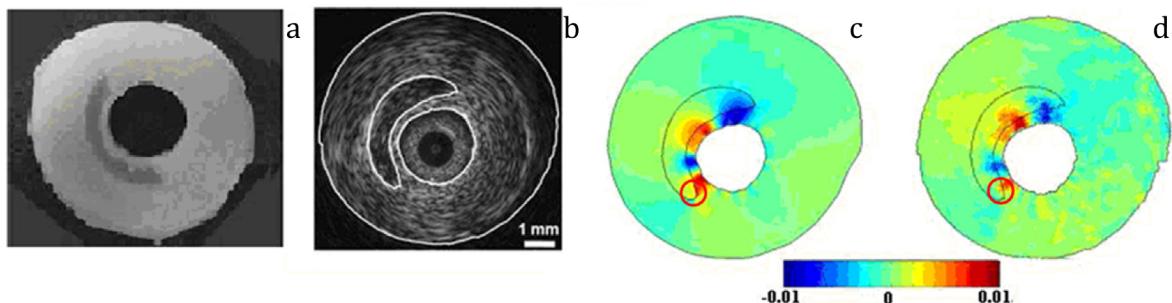


Figura 1.2: a) Phantom de PVA-C, b) Imagen IVUS-B segmentada manualmente, c) Mapa de deformación tangencial calculad en elementos finitos, d) Deformación tangencial estimado por LSME [9].

También se han definido diferentes variables como el *Anisotropic Index (AI)* [7, 12] para considerar las propiedades ortótropas del tejido vascular, ya que la amplitud de dicha variable es muy sensible a la presencia de heterogeneidades [7]. Esta variable se define como el ratio promedio de la diferencia entre las tensiones radial y circunferencial sobre la deformación radial [12] (Ecuación 1.1). En la *Figura 1.3* se muestra la representación de deformación radial estimada en una palpografía (donde el dominio no es toda la imagen) de placa real y el *AI* estimado y calculado [7].

$$AI(\Theta) = E_R^0 \cdot \left[1 - \frac{1}{4} \cdot \left(1 - \frac{E_R^0}{E_\Theta^0} + \frac{E_R^0}{E_Z^0} \right) \cdot \left(1 - \frac{E_Z^0}{E_R^0} - \frac{E_Z^0}{E_\Theta^0} \right) \right]^{-1} \quad (1.1)$$

Asimismo, es muy habitual obtener el modulograma, que es el mapa de rigidez [15, 16, 17, 18], normalmente mediante modelos iterativos en los que se usa un algoritmo [15, 17] que permite mejorar

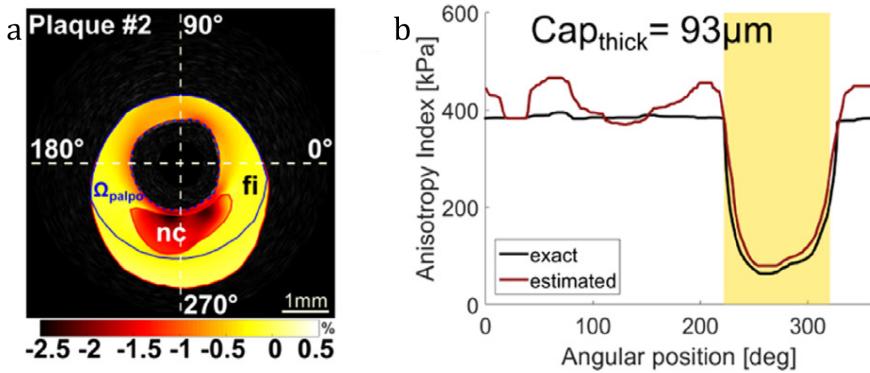


Figura 1.3: a) Deformación radial estimada a partir de una palpografía (dominio azul) b) Comparación entre el Anisotropic Index exacto del modelo de elementos finitos y el estimado [7].

la reconstrucción elástica de la placa mediante procesos de optimización. Estos se basan en el algoritmo Modified Sumi's transform (MTS) [22] en el que se relaciona el vector gradiente del módulo de Young y el tensor de deformación, y en estos casos particulares [15, 17] se toma la hipótesis de deformación plana e incompresibilidad y se define la variable del pseudo-gradiente elástico [12, 15] dW , que se expresa en la Ecuación 1.2 [15]:

$$dW = -\frac{1}{\varepsilon_R} \cdot \left(\frac{\partial \varepsilon_R}{\partial r} \right) \cdot dr - \frac{1}{\varepsilon_R} \cdot \frac{\partial \varepsilon_R}{\partial \Theta} \cdot d\Theta \quad (1.2)$$

Con la representación de esta variable es posible discernir los contornos calculados entre dos incrementos [15, 17], tras ello se realiza un segmentado watershed. Sin embargo, el mayor problema de los modulogramas no se debe tanto al proceso de optimización del algoritmo si no al precondicionamiento de las diferentes partes de la placa de ateroma para poder tener una buena estimación de los componentes y de los contornos [15]. En la Figura 1.4 se muestra un modelo donde a partir del campo de deformaciones y el modelo de elementos finitos es posible obtener el mapa elástico (módulos de Young [kPa]) de la geometría con las diferentes partes segmentadas dependiendo del número de iteraciones que se realicen (n) [15, 16, 18] y se asume un módulo elástico constante para cada parte. En estos modelos se obtiene la segmentación tras un proceso de optimización de elevado coste computacional.

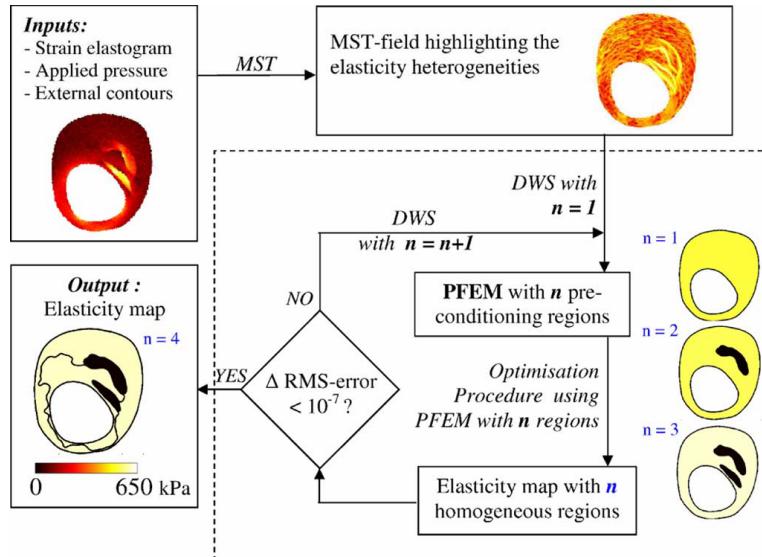


Figura 1.4: Proceso iterativo para obtener el modulograma [kPa] de una placa de ateroma a partir del campo de deformaciones y un modelo de elementos finitos, donde ' n ' es el número de iteraciones [15].

1.3. Objetivo

El objetivo del presente estudio es el de segmentar los diferentes componentes de una arteria coronaria con placa de ateroma y medir el espesor de la capa fibrótica y el área del núcleo lipídico a partir, únicamente, de las deformaciones medidas en el IVUS. Para ello se representarán diferentes variables obtenidas de las deformaciones calculadas por elementos finitos simulando los datos obtenidos mediante IVUS. Para ello, ha sido necesario abordar diferentes problemas como la caracterización de los materiales constituyentes de la coronaria con placa de ateroma y creación de modelos 3D de elementos finitos con diferentes espesores de la capa fibrótica con el objetivo de reproducir lo más fielmente posible la situación real del IVUS. Además, también se han desarrollado modelos de elementos finitos 2D a partir de imágenes de IVUS reales con el fin de validar el método. El método desarrollado considera una mayor cantidad de variables y un mayor grado de automatización respecto a los métodos ya existentes.

1.4. Metodología

Para la caracterización de los materiales se usarán curvas experimentales extraídas de diferentes publicaciones [23, 24, 25, 26, 27, 28] y se analizarán mediante los programas DataThief III [29] e Hyperfit [30]. Los diferentes modelos geométricos se realizarán en el software SolidWorks, mientras que la imposición de cargas, condiciones de contorno y análisis se realizará en el software de elementos finitos Abaqus CAE. Por último, los desplazamientos obtenidos en este programa serán la variable de entrada para trabajar en Matlab, y a partir de ella se calculará la deformación, simulando así los datos que proporcionaría un IVUS. A partir de dicha deformación se calcularán diferentes variables que permitan la segmentación.

1.5. Estructura de la Memoria

Tras el primer capítulo de introducción, donde se ha contextualizado el trabajo y se han abordado las metodologías de estudios previos, se desarrolla el segundo capítulo sobre la creación del modelo de elementos finitos, detallando la geometría (tanto analítica como modelada a partir de IVUS real), los materiales utilizados, cargas, condiciones de contorno y mallado. El tercer capítulo detalla la elaboración de una metodología, desarrollada en Matlab, que posibilita el cálculo de diferentes variables de deformación a partir de los desplazamientos obtenidos en el modelo de elementos finitos. Al final de este proceso es posible realizar la segmentación del núcleo lipídico y medición de las diferentes partes de la placa de ateroma. El cuarto y quinto capítulo son los resultados y conclusiones de todo el estudio. En la *Figura 1.5* se representan los diferentes pasos realizados hasta conseguir el objetivo final. Además, el proyecto incluye diferentes anexos: en el *Anexo I* se detalla la elección de los materiales para la creación del modelo de elementos finitos. En el *Anexo II* se expone la totalidad del código desarrollado en Matlab para el cálculo de las diferentes variables. En el *Anexo III* se muestra la validación del código del anexo anterior con modelos simples. En el *Anexo IV* se representan los resultados de aplicar el código sobre la geometría analítica de interés. Por último el *Anexo V* amplía la explicación del proceso de segmentación.

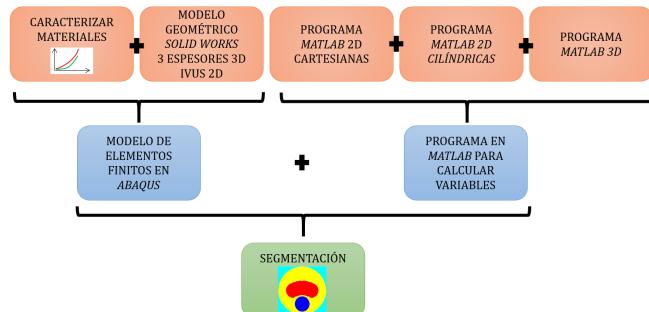


Figura 1.5: Pasos realizados hasta llegar al objetivo final (segmentación de la geometría).

2. Modelo de Elementos Finitos de la Arteria Coronaria

En este apartado se desarrollan los diferentes modelos de elementos finitos de la arteria coronaria que se han empleado en la obtención de los datos de entrada para nuestra metodología. Es importante conocer las diferentes partes que posee una arteria con placa de ateroma. Las arterias sanas están compuestas por 3 capas: la **íntima**, es la más interna y engloba al endotelio; la **media** se caracteriza por tener el mayor grosor y posee fibras musculares lisas, elastina y colágeno y es la principal encargada de soportar las cargas; y por último al **adventicia**, la capa más exterior y está compuesta de colágeno y fibroblastos. Cuando la arteria padece aterosclerosis hay que considerar el **núcleo lipídico**, que es un cúmulo de grasa muy blando. Además, la íntima que lo rodea enferma y aumenta su rigidez, pasando a denominarse **capa fibrótica**. Estos cinco tejidos deben considerarse en las geometrías y en la caracterización de los materiales.

Se crean geometrías idealizadas 3D que permitan representar, de forma idealizada, una placa de ateroma en una arteria coronaria en fase de crecimiento positivo vulnerable, en el límite y estable. Con esta geometría se elaborará la posterior metodología para realizar la segmentación y medida del núcleo lipídico y espesor de la capa fibrótica. Este proyecto toma como referencia la geometría desarrollada en proyectos anteriores [31, 32] en los que se utiliza una placa de 13 mm de longitud, un núcleo lipídico de 6,5 mm y un espesor de la capa fibrótica de **65 micras**, considerada vulnerable [33]. Estas medidas se aprecian en la *Figura 2.1*. Además, se modifica el espesor de la capa fibrótica obteniendo dos modelos con **150 y 300 micras** de espesor que se encuentran dentro de los rangos analizados en otros estudios [7, 15].

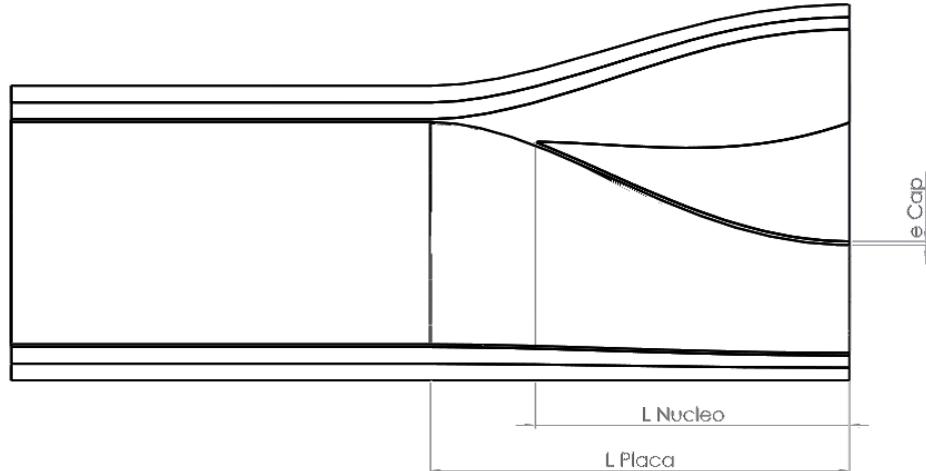


Figura 2.1: Vista longitudinal del modelo, con longitud de placa (L Placa) y núcleo (L Nucleo) y espesor de la capa fibrótica (e Cap) señalizados [31].

Asimismo, se ampliará el uso del método de segmentación con geometrías no idealizadas e IVUS reales [34] reconstruidos manualmente en Abaqus CAE. Todas las geometrías serán analizadas con materiales que siguen el modelo de Gasser [35] tal y como se detallará más adelante.

2.1. Geometrías

En este estudio se ha trabajado con tres tipos de geometrías, todas ellas modeladas en condición de cero presión en el interior del lumen:

1. Geometrías idealizadas: son geometrías 3D obtenidas de un estudio anterior [31] en el que se desarrollaban modelos con diferentes longitudes de placa y núcleo lipídico. Se ha seleccionado el modelo de 13 mm de longitud de placa y 6,5 mm de núcleo lipídico, aunque la longitud de placa y proporción de núcleo lipídico sí es importante en los términos de tensión y vulnerabilidad [31] no es influyente en la zona de máxima estenosis, que es la zona de interés a analizar. La geometría se ha desarrollado en el software Solid Works siguiendo los pasos descritos en el proyecto anterior [31]. Al ser un modelo simétrico, tan solo es necesario modelar un cuarto de la geometría (*Figura 2.2*). El modelo inicial tiene un espesor de capa fibrótica de 65 micras, se ha modificado para tener otros dos modelos con 150 y 300 micras, tal y como se muestra en la *Figura 2.3*. Con esto se pretende comprobar la influencia del espesor de la capa fibrótica en el método de segmentación propuesto.

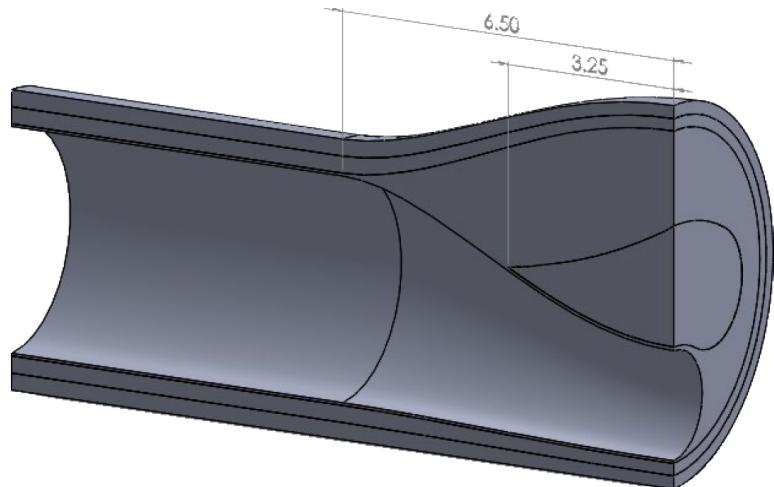


Figura 2.2: Modelo geométrico 3D realizado en Solid Works. Placa de 13 mm, núcleo lipídico de 6.5 mm y capa fibrótica de 65 μm .

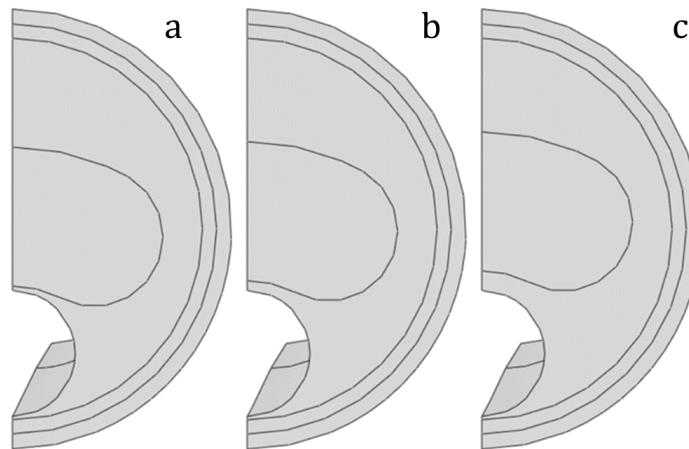


Figura 2.3: Vista frontal de la sección de máxima estenosis de los modelos idealizados con diferente espesor de la capa fibrótica: a) 65 μm b) 150 μm y c) 300 μm .

2. Geometrías no idealizadas: se han denominado así a las geometrías realizadas a partir del modelo anterior, pero variando de forma aleatoria la forma del núcleo lipídico. Tan solo se han desarrollado 2 geometrías con el fin de comprobar la metodología de segmentación desarrollada más adelante. En la *Figura 2.4* se observan las geometrías. En el caso de la *Figura 2.4b* la geometría total tendrá 2 núcleos, debido a la simetría.

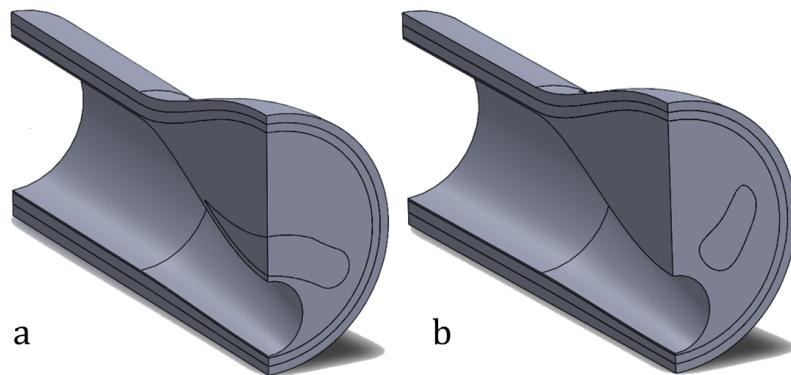


Figura 2.4: Modelos geométricos 3D realizados en Solid Works con núcleo lipídicos con formas no idealizadas, a) forma 1 y b) forma 2.

3. Geometría IVUS real: la última geometría se realiza tomando un IVUS real [34], donde en la Figura 2.5a [34] se muestra el IVUS y la segmentación manual realizada por un experto Figura 2.5b. A partir de esa segmentación se ha reconstruido manualmente la geometría en Abaqus CAE (Figura 2.5c y d). En este caso se ha construido únicamente como un modelo 2D considerando solo la información del núcleo y placa de ateroma y no se considera tejido sano (adventicia y media).

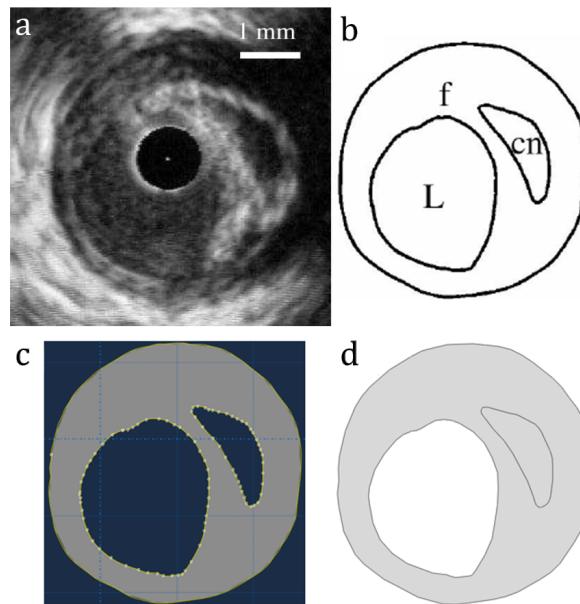


Figura 2.5: a) IVUS real [34], b) Segmentado manual de las diferentes partes [34], c) Dibujado del contorno en Abaqus CAE, c) Modelo 2D de la placa de ateroma y núcleo lipídico en Abaqus CAE.

2.2. Materiales

Debido a que los vasos sanguíneos son materiales anisótropos, fibrados y sometidos a grandes deformaciones se ha empleado un modelo de comportamiento hiperelástico, no lineal, anisótropo e incompresible. Tal y como ya se realizó en estudios anteriores [3, 31] se toma el **modelo de Gasser** [35] y se consideran dos tipos de tejidos, sanos y enfermos. Dentro del tejido sano se consideran la íntima, media y adventicia; todos ellos se modelan como materiales anisótropos fibrados, con dos familias de fibras orientadas siguiendo los resultados experimentales obtenidos en otros estudios [35]. Por otro lado, en el material enfermo se incluye la capa fibrótica (íntima enferma) y el núcleo lipídico que se modelan como material isotrópico [23]. La ecuación de comportamiento hiperelástico (Ecuación 2.1) viene definida por unos parámetros que se han obtenido mediante el ajuste de curvas obtenidas de la

bibliografía [23, 24], tal y como se detalla en el *Anexo I*.

$$\Psi = \frac{1}{D} \cdot [J - 1]^2 + \mu \cdot [I_1 - 3] + \frac{k_1}{2 \cdot k_2} \cdot \sum_{i=4,6} \exp \left(k_2 \cdot [\kappa \cdot [I_1 - 3] + [I_1 - 3 \cdot \kappa] \cdot [I_i - 1]]^2 \right) - 1 \quad (2.1)$$

Este modelo de Gasser considera dos familias de fibras, pero con un mismo comportamiento, donde k_1 corresponde a la rigidez general [kPa] y k_2 es la rigidez del colágeno (final del comportamiento del vaso). El valor D es referido al factor de incompresibilidad, en este caso es $D=0$ por ser un material incompresible. El parámetro μ es la rigidez proporcionada por la matriz [kPa] y κ aporta información sobre la dispersión de las fibras y puede obtenerse de forma experimental. En el caso particular de $\kappa=0.33$, se considera que el material es isótropo. El ángulo de las fibras está definido sobre la dirección circunferencial.

A partir de curvas obtenidas en bibliografía [23, 24] se obtienen los puntos mediante el software libre DataThief III [29] y los parámetros de ajuste se procesan en el software Hyperfit [30]. En el *Anexo I: Materiales* se detallan los diferentes materiales analizados, así como sus curvas de comportamiento y el proceso para obtener los parámetros que permiten definir el modelo. En la *Tabla 2.1* se pueden observar los parámetros utilizados para cada material.

Tejido	μ [kPa]	D [-]	k_1 [kPa]	k_2 [kPa]	κ [-]	Ángulo [º]
Adventicia	8.44	0	547.67	568.01	0.26	± 61.8
Media	1.4	0	206.16	58.55	0.29	± 28.35
Íntima	6.81	0	109.10	101.04	0.21	± 52.72
Capa Fibrótica	9.58	0	17654.91	0.51	0.33	-
Núcleo Lipídico	0.05	0	956.76	70	0.33	-

Tabla 2.1: Parámetros para definir el modelo de comportamiento de Gasser.

2.3. Ensamblado

Tras haber definido las geometrías y los materiales se procede al ensamblado. En las geometrías idealizadas y no idealizadas el proceso es el mismo, ya que se tiene el modelo 3D en Solid Works y se traspasa al software de elementos finitos Abaqus CAE. Se importan las diferentes partes y se ensamblan una a una, tal y como se muestra en la *Figura 2.6*.

El siguiente paso antes de introducir los materiales e imponer las condiciones de contorno es diferenciar entre íntima sana e íntima enferma (capa fibrótica). Para ello se realiza una partición tal y como se muestra en la *Figura 2.7*.

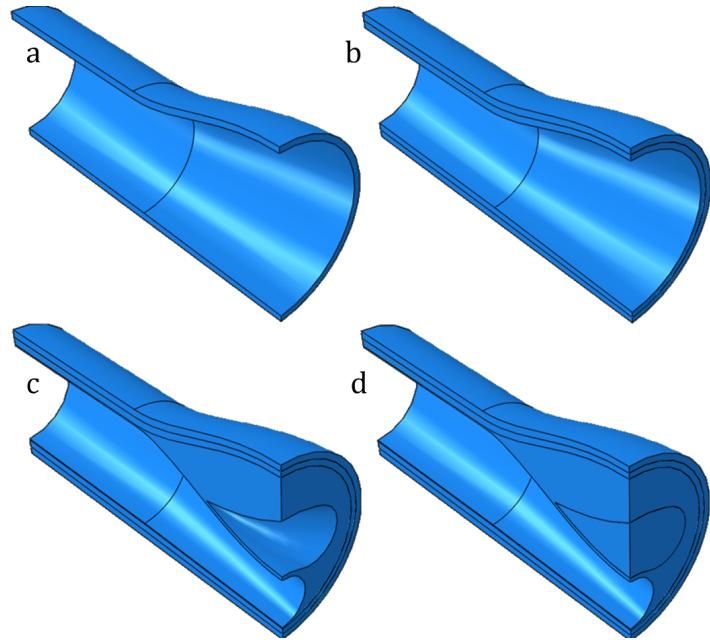


Figura 2.6: Ensamblado de las diferentes partes de una placa de 13 mm de longitud y un núcleo de 6.5 mm; a) Adventicia, b) Adventicia y Media, c) Adventicia, Media y Placa, d) Adventicia, Media, Placa y Núcleo lipídico.

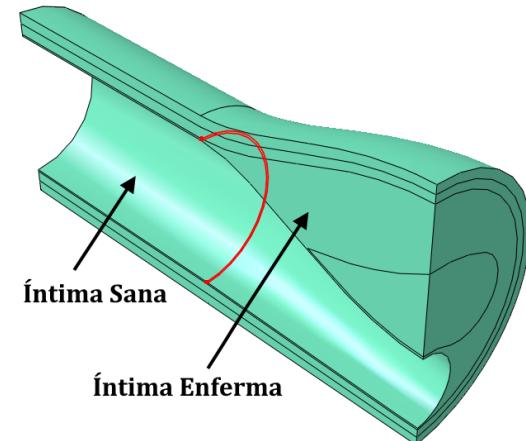


Figura 2.7: Partición para poder considerar la íntima sana y enferma.

2.4. Cargas y Condiciones de Contorno

En este paso se introducen las condiciones de contorno y cargas sobre el ensamblaje anterior. Al estar trabajando con un cuarto de la geometría es necesario imponer dichas condiciones de contorno impidiendo desplazamientos y giros en los planos de simetría (*Figura 2.8 a y b*). Además, en la generatriz exterior se impone una condición de no desplazamiento en el eje Y para impedir el desplazamiento como sólido rígido y simular el contacto con el corazón (*Figura 2.8 c*). La presión sanguínea en una coronaria sana se encuentra entre 80 y 120 mmHg, sin embargo, el pico de presión sistólica en pacientes ateroscleróticos alcanza 140 mmHg [36], que es lo mismo que 18.7 kPa. En la *Figura 2.8 d* se muestra la imposición de dicha presión sobre el lumen.

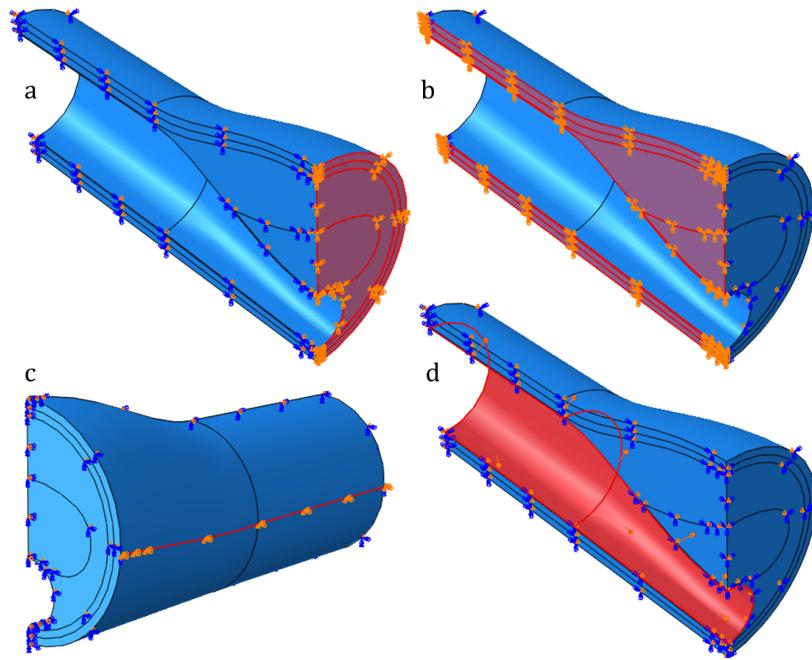


Figura 2.8: Imposición de las condiciones de contorno de simetría a) y b), contacto con el corazón en c) y presión de 18.7 kPa en d).

2.5. Mallado

El último paso antes de calcular el modelo es realizar una malla sobre el modelo. Es importante señalar que una malla muy fina permite obtener resultados más cercanos a la realidad y obtendrá un mapa de valores mucho más suavizado que con una malla gruesa. Por contra, el cálculo tardará más tiempo en finalizar y supondrá también un aumento considerable del tiempo de post-procesado de datos dentro de Matlab. En el otro extremo, una malla gruesa permitiría un cálculo y procesado de los datos rápido, pero los resultados se aproximarían peor al comportamiento real y al tener menos información nodal la segmentación sería mucho más complicada. Por lo tanto, al igual que en el estudio anterior realizado [31] donde ya se hizo un estudio de sensibilidad de mallado, se propone una malla adaptada, con un mallado global grueso y uno más fino en las zonas de interés. Además, dependiendo del modelo y del espesor de la capa fibrótica la malla también se ha modificado.

Para este estudio la zona más importante es la sección de máxima estenosis y en particular el área que rodea al núcleo lipídico, por lo tanto, esas zonas tendrán un tamaño de malla pequeño de 0,2 mm (Figura 2.9.a). De este modo, se pretende medir la capa fibrótica, por ello se necesitará mayor precisión y un mínimo de 3 elementos en el espesor entre núcleo y lumen (capa fibrótica), por lo que se usa un tamaño más fino de 0.1 mm para el espesor de 300 micras y 0.02 mm para el espesor de 65 micras (Figura 2.9.b). En el resto del modelo se toma una malla más gruesa, de 0,5 mm de tamaño (Figura 2.9.c). En la Tabla 2.2 se muestran las características de las mallas de cada modelo.

Modelo	Tipo de Elemento	Malla Global	Malla Fina	Malla Muy Fina	Nº Elementos	Nº Nodos
Geometría Idealizada	300 micras	C3D10H	0.6 mm	0.1 mm	284975	399526
	150 micras	C3D10H	0.5 mm	0.2 mm	32427	48182
	65 micras	C3D10H	0.5 mm	0.2 mm	45537	67661
Geometría no Idealizada	Forma 1	C3D10H	0.5 mm	0.2 mm	44331	65930
	Forma 2	C3D10H	0.5 mm	0.2 mm	46162	67970
Geometría Real	IVUS	CPE3H	0.05 mm	0.05 mm	8722	4569

Tabla 2.2: Información sobre las mallas de los diferentes modelos.

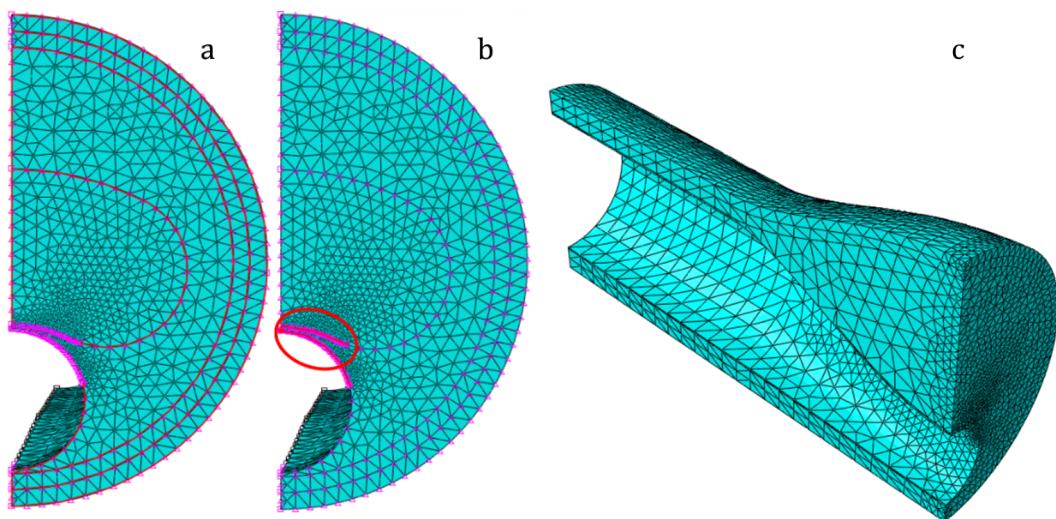


Figura 2.9: Mallado adaptado, a)Malla fina en la zona de máxima estenosis, b)En el círculo rojo se observa la malla muy fina en la zona de la capa fibrótica c) Malla gruesa en el resto del modelo.

La incompresibilidad de los tejidos es un obstáculo a la hora de simular los modelos, por lo que ha sido necesario a emplear elementos híbridos de mallado en Abaqus para poder resolver dicho problema.

Para simular correctamente la adquisición de imágenes en IVUS hay que conocer su precisión, ya que con ello sabremos el tamaño mínimo de elemento a utilizar. Actualmente, existen muchos tipos de catéteres que permiten diferentes resoluciones, desde 170 micras hasta menos de 38 micras en la dirección axial [37], por lo tanto el límite estaría en 0,038 mm de tamaño de elemento. No obstante, en el modelo de 65 micras de espesor se ha utilizado un tamaño local de 0,02 mm para obtener mejor resolución en esa zona. El resto de zonas y de modelos están dentro de la resolución de catéter comercial para IVUS [37]. Esto es una limitación que se tendrá que tener en cuenta, sobretodo en el modelo de 65 micras.

3. Procesado de Datos y Método de Segmentación

3.1. Post-Procesado de Datos

Una vez han finalizado los cálculos de todos los modelos de elementos finitos se pueden visualizar los resultados en Abaqus. Aunque el incremento de presión introducido en la simulación es de 140 mmHg, el software resuelve el problema en pequeños incrementos de presión, o *steps*, desde 0 hasta 140 mmHg, lo que permite seleccionar dos *steps* con una diferencia de presión entre ellos de aproximadamente 5 mmHg. Ambos *steps* se toman en una presión cercana a 110 mmHg, que se toma como un valor intermedio entre sístole y diástole. A partir de los desplazamientos de cada *step* se puede obtener el desplazamiento relativo entre ambos y se calcula la deformación que experimenta la placa en dicho incremento de presión. Al ser un incremento de presión pequeño, se puede utilizar la hipótesis de pequeños desplazamientos y calcular la deformación mediante elasticidad lineal. Esta metodología proporciona las deformaciones tal y como se obtendrían mediante IVUS tomando dos imágenes consecutivas [7, 16]. Las imágenes de IVUS aportan información de las coordenadas y deformaciones tomando el catéter como origen de coordenadas [15], del mismo modo se coloca el origen de coordenadas de los modelos de elementos finitos en el centro del lumen, simulando así el catéter (*Figura 3.1*).

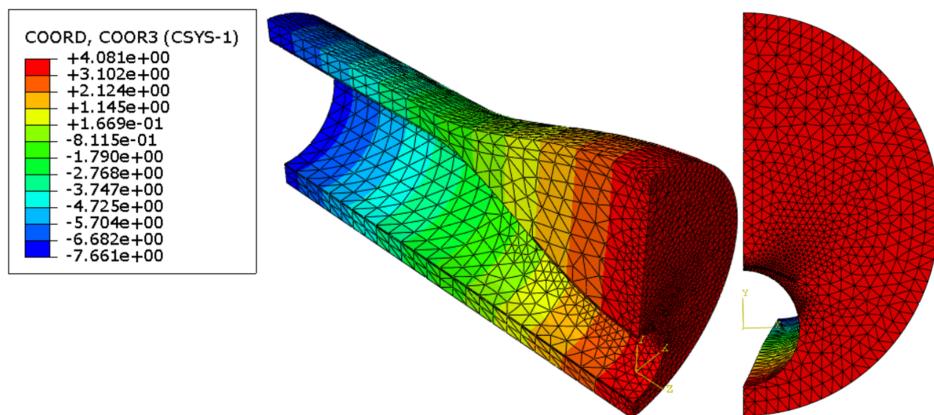


Figura 3.1: Coordenada ‘Z’ [mm] representada en el modelo con 65 micras de espesor. El centro de coordenadas se ha definido en el centro del lumen simulando el catéter.

Se exporta a Matlab información de las coordenadas y desplazamientos de todos los nodos de la malla, pero el método desarrollado dependerá de si se va a analizar como un modelo 2D o 3D. En 2D se analiza únicamente la sección de máxima estenosis, calculando todas las variables bajo la hipótesis de deformación plana ($u_z=0$), mientras que en 3D se toma una pequeña rebanada y se consideran todos los desplazamientos. Todo el código utilizado para el tratamiento de información, cálculo de la deformación, gradientes y otras variables se encuentra detallado en el *Anexo II: Metodología desarrollada en Matlab*.

El objetivo final de este proyecto es el de comparar diferentes variables que permitan la segmentación de la placa de ateroma dentro de la práctica clínica y realizar medidas del área del núcleo lipídico segmentado y medir el espesor de la capa fibrótica, ya que son las variables empleadas para determinar la vulnerabilidad de una placa de ateroma. Para ello se trabaja con los desplazamientos obtenido del análisis de elementos finitos, que permiten calcular la deformación simulando la información proporcionada por medio de imágenes clínicas como IVUS. A partir de ellos, es posible calcular diferentes variables como las deformaciones en cartesianas, cilíndricas, principales y sus gradientes entre otros, con el fin de visualizar de una forma más clara los diferentes componentes de la placa de ateroma. En el *Anexo III: Validación del Método* se muestran unos modelos geométricos muy simples sobre los que

se construyó y validó el método para poder aplicarlo en las geometrías de interés. La validación del programa de Matlab se realizó mediante la comparación entre las deformaciones calculadas en Matlab con las proporcionadas por Abaqus.

En el caso de trabajar con el modelo bidimensional se exporta, desde Abaqus CAE, la información de las coordenadas (X , Y y Z) de todos los nodos y únicamente los desplazamientos en X y en Y (u_x y u_y) ya que el desplazamiento en el eje Z se considera nulo ($u_z=0$). Con ello se impone la hipótesis de deformación plana, donde el tensor de deformación no dependerá del eje Z , tal y como se muestra en la *Ecuación 3.1*. Esta forma de proceder es la misma que se realiza clínicamente con las imágenes de IVUS de coronaria. En otro tipo de arterias, como carótidas, es posible obtener información en todos los ejes mediante IVUS longitudinales en lugar de transversales o imágenes de MRI, pero en arteria coronaria, por ahora, únicamente es posible obtener información de la sección transversal.

$$\bar{\varepsilon} = \begin{bmatrix} \varepsilon_x & \varepsilon_{xy} & 0 \\ \varepsilon_{xy} & \varepsilon_y & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.1)$$

Se importa a Matlab la información de las coordenadas (X e Y) y desplazamientos (u_x y u_y) de todos los nodos del modelo 3D. Adicionalmente se importan las deformaciones logarítmicas principales y cartesianas con el fin de realizar comparaciones entre los valores obtenidos en Abaqus CAE y Matlab (ver *Anexo II* para información más detallada). El siguiente paso es quedarse únicamente con los nodos de la sección de máxima estenosis seleccionando únicamente los nodos que tengan la coordenada Z máxima. Seguidamente se debe reconstruir los elementos de la malla uniendo los nodos mediante una triangulación de Delaunay (ver *Figura 3.2.a*). En este paso se crean elementos ficticios que cierran la geometría del lumen, por lo tanto, es preciso eliminar dichos elementos mediante un filtro que discrimine según el tamaño del radio de la circunferencia que circunscribe a cada elemento (*Figura 3.2.b*). A continuación, se crea un modelo de derivadas parciales (PDE model de la toolbox de Matlab *Differential Equation Toolbox*) que permite representar el modelo y realizar operaciones sobre él (*Figura 3.2.c*).

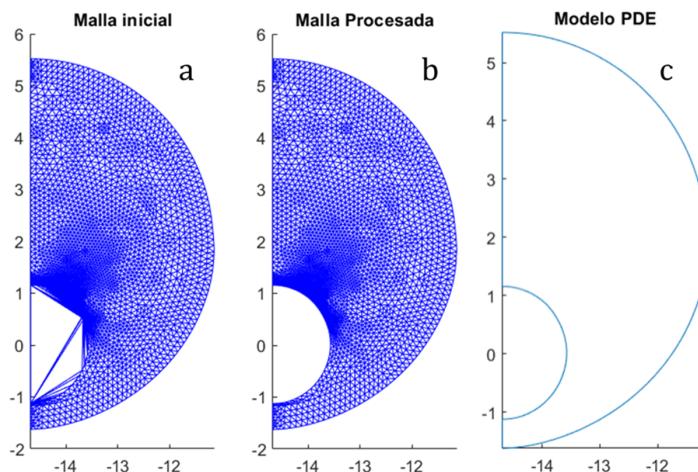


Figura 3.2: a) Malla inicial por triangulación de Delaunay, b) Malla procesada, c) Modelo PDE.

Al tener los desplazamientos nodales se pueden introducir como solución del modelo PDE directamente en Matlab, lo que posibilita representar el gradiente de dicha solución. En la *Figura 3.3* se muestran los desplazamientos (*a, b*), el gradiente de los desplazamientos (*c, d*) y el módulo del gradiente de los desplazamientos (*e, f*).

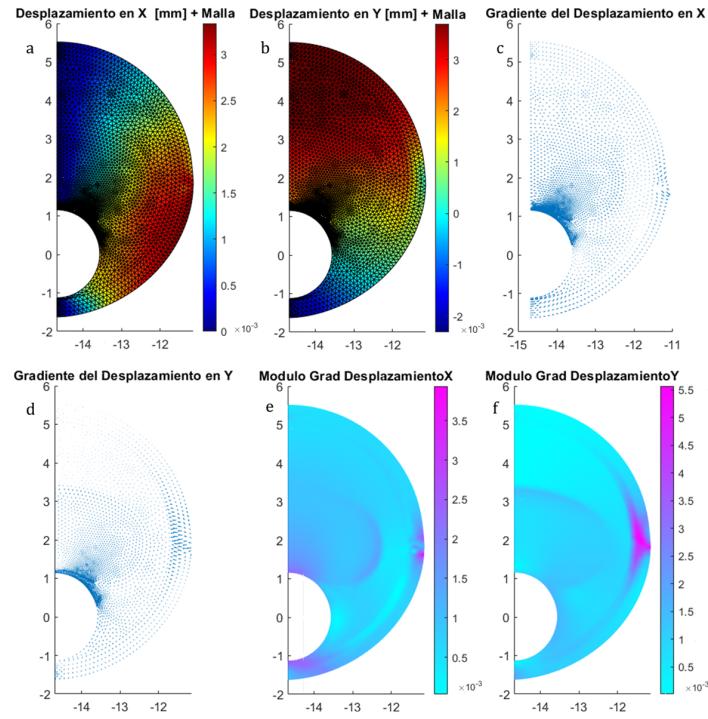


Figura 3.3: Desplazamientos [mm] en X a) y en Y b), Gradientes desplazamientos en X c) y en Y d), Módulo del Gradiente de desplazamientos en X e) y en Y f).

3.2. Cálculo de la Deformación y otras Variables en 2D

Teniendo el modelo mallado con triángulos lineales (elemento *CPE3*) y conocidos los desplazamientos en cada nodo del elemento, se introducen las funciones de forma en coordenadas naturales. Estas funciones (N_i , en las *Ecuaciones 3.2, 3.3 y 3.4*) son funciones sencillas que permiten calcular el valor de una variable dentro del elemento. En este caso se aproximan las variables deformación de forma lineal entre los nodos de cada elemento. Además se utilizan las coordenadas naturales (ξ, η) en lugar de las cartesianas (x e y), ya que permiten que las funciones de forma sean las mismas para todos los elementos de la malla.

$$N_1 = 1 - \xi - \eta \quad (3.2)$$

$$N_2 = \xi \quad (3.3)$$

$$N_3 = \eta \quad (3.4)$$

A continuación se crean las ecuaciones de posición X e Y en función de las coordenadas naturales, para ello se multiplica el valor de cada coordenada nodal (x_i, y_i) por su función de forma (N_i):

$$X(\xi, \eta) = [N_1 N_2 N_3] \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = N_1 \cdot x_1 + N_2 \cdot x_2 + N_3 \cdot x_3 \quad (3.5)$$

$$Y(\xi, \eta) = [N_1 N_2 N_3] \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = N_1 \cdot y_1 + N_2 \cdot y_2 + N_3 \cdot y_3 \quad (3.6)$$

Partiendo de las dos ecuaciones anteriores es posible despejar ξ y η en función de X e Y .

$$\xi = f(X, Y) \quad (3.7)$$

$$\eta = f(X, Y) \quad (3.8)$$

Seguidamente se construyen las funciones de los desplazamientos de igual modo:

$$U_x(\xi, \eta) = [N_1 N_2 N_3] \cdot \begin{bmatrix} u_{x1} \\ u_{x2} \\ u_{x3} \end{bmatrix} \quad (3.9)$$

$$U_y(\xi, \eta) = [N_1 N_2 N_3] \cdot \begin{bmatrix} u_{y1} \\ u_{y2} \\ u_{y3} \end{bmatrix} \quad (3.10)$$

Ahora es posible calcular la deformación en cada elemento, que, al ser triángulos lineales será un valor constante en cada elemento. En las *Ecuaciones 3.11, 3.12, 3.13* se muestra el cálculo de la deformación en ejes cartesianos.

$$\varepsilon_x = \frac{\partial U_x}{\partial x} = \frac{\partial U_x}{\partial \xi} \cdot \frac{\partial \xi}{\partial x} + \frac{\partial U_x}{\partial \eta} \cdot \frac{\partial \eta}{\partial x} \quad (3.11)$$

$$\varepsilon_y = \frac{\partial U_y}{\partial y} = \frac{\partial U_y}{\partial \xi} \cdot \frac{\partial \xi}{\partial y} + \frac{\partial U_y}{\partial \eta} \cdot \frac{\partial \eta}{\partial y} \quad (3.12)$$

$$\varepsilon_{xy} = \frac{1}{2} \cdot \left(\frac{\partial U_x}{\partial y} + \frac{\partial U_y}{\partial x} \right) = \frac{1}{2} \cdot \left(\frac{\partial U_x}{\partial \xi} \cdot \frac{\partial \xi}{\partial y} + \frac{\partial U_x}{\partial \eta} \cdot \frac{\partial \eta}{\partial y} + \frac{\partial U_y}{\partial \xi} \cdot \frac{\partial \xi}{\partial x} + \frac{\partial U_y}{\partial \eta} \cdot \frac{\partial \eta}{\partial x} \right) \quad (3.13)$$

Para aproximar en los nodos, al ser elementos lineales, basta con ponderar el valor de la deformación de cada uno de los elementos a los que pertenece y dividir por el número de elementos. Aunque no es el método más preciso (ya que no considera la cuadratura de Gauss), es más sencillo, rápido y se obtienen buenos resultados tal y como se explica en el *Anexo III*. Una vez se han obtenido las deformaciones cartesianas nodales es posible introducirlas como resultado del modelo PDE y así conseguir de forma sencilla el gradiente y el módulo del gradiente de las mismas, tal y como se ha realizado con los desplazamientos. En estos modelos es posible realizar una comparación entre la deformación calculada siguiendo este método desarrollado en Matlab y la deformación logarítmica (LE) obtenida en Abaqus CAE. Para ello se define la deformación de Abaqus *Ecuación 3.14* (ε_{Abaqus}) y el error relativo (*Ecuación 3.15*) y absoluto (*Ecuación 3.16*) entre ambos métodos. El error relativo puede salir en ocasiones muy alto, esto se debe a que los valores de deformación son muy pequeños y hacen que el denominador de la *Ecuación 3.15* tienda a cero, por ello se calcula también el error absoluto.

$$\varepsilon_{Abaqus} = \exp(LE_{step2} - LE_{step1} - 1) \quad (3.14)$$

$$ERROR_{Relativo} (\%) = \left| \frac{\varepsilon_{Abaqus} - \varepsilon_{Matlab}}{\varepsilon_{Abaqus}} \right| \cdot 100 \quad (3.15)$$

$$ERROR_{Absoluto} (\%) = |\varepsilon_{Abaqus} - \varepsilon_{Matlab}| \cdot 100 \quad (3.16)$$

Además, las deformaciones principales (ε_I o ε_{max} , ε_{II} o ε_{med} y ε_{III} o ε_{min}) se pueden calcular como un problema de valores propios a partir del tensor de deformaciones en cartesianas (3.1). A partir de las deformaciones principales se obtienen otras variables como la deformación equivalente de Von

Mises (*Ecuación 3.17*), de Tresca (*Ecuación 3.18*) o el índice de anisotropía [38] (*Ecuación 3.19*) particularizando los valores en cada nodo.

$$\varepsilon_{vonMises} = \frac{\sqrt{(\varepsilon_I - \varepsilon_{II})^2 + (\varepsilon_{II} - \varepsilon_{III})^2 + (\varepsilon_I - \varepsilon_{III})^2}}{2} \quad (3.17)$$

$$\varepsilon_{Tresca} = |\varepsilon_I - \varepsilon_{III}| \quad (3.18)$$

$$FA = \frac{\sqrt{3}}{\sqrt{2}} \cdot \frac{\sqrt{(\lambda_1 - \lambda)^2 + (\lambda_2 - \lambda)^2 + (\lambda_3 - \lambda)^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}, \text{ donde } \lambda = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3}, \text{ y } \lambda_i = \varepsilon_i + 1 \quad (3.19)$$

El IVUS proporciona las deformaciones en cilíndricas, por lo que el siguiente paso es obtener dichas deformaciones mediante una operación de cambio de base expresada en la *Ecuación 3.20*, donde $\theta = \tan^{-1}(Y/X)$:

$$\begin{bmatrix} \varepsilon_R & \varepsilon_{R\Theta} & \varepsilon_{RZ} \\ \varepsilon_{R\Theta} & \varepsilon_\Theta & \varepsilon_{\Theta Z} \\ \varepsilon_{RZ} & \varepsilon_{\Theta Z} & \varepsilon_Z \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \varepsilon_x & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_y & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_z \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.20)$$

En diferentes estudios se ha usado el criterio de la *Transformada Modificada de Sumi (MST)* [22] en los que se relaciona el vector gradiente del módulo de Young y el tensor de deformaciones [15, 18], con lo que se puede reconstruir el mapa de rigidez bajo la hipótesis de tensión plana. Este método se ha extendido a deformación plana [15] considerando los materiales incompresibles. Se expresa la ley de Hooke en elasticidad lineal y en un medio heterogéneo [15]:

$$\bar{\sigma} = -p \cdot \bar{I} + \frac{2}{3} \cdot E(\vec{x}) \cdot \bar{\varepsilon} \quad (3.21)$$

Donde $\bar{\sigma}$ y $\bar{\varepsilon}$ son tensores de tensión y deformación respectivamente, $E(\vec{x})$ es el módulo de Young que depende de la posición, ya que es para materiales heterogéneos, \bar{I} es la matriz identidad y p depende de la incompresibilidad del material. Si se impone la incompresibilidad la traza de $\bar{\varepsilon}$ es cero [15]:

$$\text{Trazza}[\bar{\varepsilon}] = \varepsilon_x + \varepsilon_y + \varepsilon_{xy} = 0 \quad (3.22)$$

Además, se desprecian las fuerzas iniciales y gravitatorias, por lo que la ecuación de equilibrio sería [15]:

$$\nabla \cdot \bar{\sigma} = \vec{0} \quad (3.23)$$

Sustituyendo las 3 ecuaciones anteriores:

$$\frac{\nabla E}{E} = \frac{3}{2} \cdot \bar{\varepsilon}^{-1} \cdot \frac{\nabla p}{E} - \bar{\varepsilon}^{-1} \cdot \nabla \cdot \bar{\varepsilon} \quad (3.24)$$

Si se define la ecuación como vector se obtiene:

$$\vec{H} = -\bar{\varepsilon}^{-1} \cdot \nabla \cdot \bar{\varepsilon} \quad (3.25)$$

Debido a la incompresibilidad (*Ecuación 3.22*) la deformación radial y circunferencial tienen el mismo valor absoluto y signo contrario. Si se escriben las componentes de \vec{H} en coordenadas cilíndricas se obtiene:

$$H_R = \frac{1}{\varepsilon_R^2 + \varepsilon_{R\Theta}^2} \cdot \left[\varepsilon_R \cdot \left(\frac{\partial \varepsilon_R}{\partial R} + \frac{1}{R} \cdot \frac{\partial \varepsilon_{R\Theta}}{\partial \Theta} + \frac{2 \cdot \varepsilon_R}{R} \right) + \varepsilon_{R\Theta} \cdot \left(\frac{\partial \varepsilon_{R\Theta}}{\partial R} - \frac{1}{R} \cdot \frac{\partial \varepsilon_R}{\partial \Theta} + \frac{2 \cdot \varepsilon_{R\Theta}}{R} \right) \right] \quad (3.26)$$

$$H_\Theta = \frac{1}{\varepsilon_R^2 + \varepsilon_{R\Theta}^2} \cdot \left[\varepsilon_{R\Theta} \cdot \left(\frac{\partial \varepsilon_R}{\partial R} + \frac{1}{R} \cdot \frac{\partial \varepsilon_{R\Theta}}{\partial \Theta} + \frac{2 \cdot \varepsilon_R}{R} \right) - \varepsilon_R \cdot \left(\frac{\partial \varepsilon_{R\Theta}}{\partial R} - \frac{1}{R} \cdot \frac{\partial \varepsilon_R}{\partial \Theta} + \frac{2 \cdot \varepsilon_{R\Theta}}{R} \right) \right] \quad (3.27)$$

Finalmente se define la variable dW [15] que cuantifica el gradiente elástico del material:

$$dW = \vec{H} \cdot \vec{dx} = [H_R \ H_\Theta] \cdot \begin{bmatrix} dr \\ d\Theta \end{bmatrix} \quad (3.28)$$

Donde $dr = \frac{r_{nodal}}{\sqrt{r_{nodal}^2 + \Theta_{nodal}^2}}$ y $d\Theta = \frac{\Theta_{nodal}}{\sqrt{r_{nodal}^2 + \Theta_{nodal}^2}}$.

En estudios anteriores también se utiliza esta variable [15, 18] pero con más simplificaciones y despreciando las deformaciones tangenciales, en este caso se consideran todos los términos de la ecuación. Todas estas variables se han calculado en Matlab y se han obtenido sus gradientes y el módulo de dichos gradientes. En el *Anexo IV* se muestran todas las variables calculadas para la geometría idealizada con 150 micras de espesor.

3.3. Proceso y Cálculo en 3D

La tecnología IVUS obtiene deformaciones a partir de la información obtenida en el plano transversal e imponiendo la hipótesis de deformación plana, tal y como se ha recreado en el apartado 2D anterior. Sin embargo, también se ha calculado un modelo 3D donde se obtienen las mismas variables de deformación pero considerando los desplazamientos en los 3 ejes. Con ello se desea comprobar si la hipótesis de deformación plana limita el proceso de segmentación. En este apartado, en lugar de trabajar únicamente con la sección 2D de máxima estenosis, se trabaja con una pequeña porción 3D que contiene dicha sección. Esta metodología permite tener en consideración los desplazamientos en todos los ejes. El procedimiento es el mismo que en el caso de 2D, en el *Anexo II* se muestran las diferencias entre el código de ambos casos.

En este proceso se importan a Matlab las coordenadas (X , Y y Z) y los desplazamientos (u_x , u_y y u_z) de todos los nodos del modelo de Abaqus CAE. Adicionalmente, se pueden importar las deformaciones logarítmicas calculadas en Abaqus para poder realizar la comparación de los resultados. El siguiente paso es quedarse con los nodos de una pequeña rebanada 3D que contenga la sección de máxima estenosis (para más información ver *Anexo II*) y se realiza el mallado con la triangulación de Delaunay y con tetraedros lineales (tipo *C3D4*) como elementos de la malla. En este caso también es necesario reconstruir o depurar la malla para eliminar elementos ficticios (*Figura 3.4a*). Para ello se calcula el radio de la esfera que contiene a los 4 nodos [39] y se limita por tamaño (*Figura 3.4b*). Por último se crea el modelo PDE de la rebanada a analizar (*Figura 3.4c*).

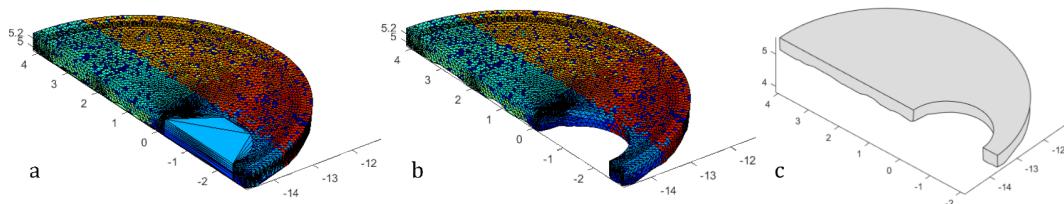


Figura 3.4: a) Malla inicial, b) Malla depurada, c) Modelo PDE.

Se definen las funciones de forma en coordenadas naturales:

$$N_1 = 1 - \xi - \eta - \zeta \quad (3.29)$$

$$N_2 = \xi \quad (3.30)$$

$$N_3 = \eta \quad (3.31)$$

$$N_4 = \zeta \quad (3.32)$$

El proceso para calcular las deformaciones es el mismo que en el caso anterior, pero con 4 funciones de forma en lugar de 3. En la *Ecuación 3.33* se expone un ejemplo de cómo se calcula la deformación, en este caso en el eje X .

$$\varepsilon_x = \frac{\partial U_x}{\partial x} = \frac{\partial U_x}{\partial \xi} \cdot \frac{\partial \xi}{\partial x} + \frac{\partial U_x}{\partial \eta} \cdot \frac{\partial \eta}{\partial x} + \frac{\partial U_x}{\partial \zeta} \cdot \frac{\partial \zeta}{\partial x} \quad (3.33)$$

El resto de deformaciones y gradientes se obtienen siguiendo el mismo procedimiento ya descrito. La variable dW también se modifica, ya que en el apartado anterior estaba calculada bajo la hipótesis de deformación plana. El planteamiento inicial será similar, sin embargo el vector \bar{H} pasará a tener 3 componentes (*Ecuaciones 3.34, 3.35 y 3.36*) en lugar de dos y serán dependientes de las deformaciones en el eje Z .

$$H_R = \frac{(-\varepsilon_{\Theta Z}^2 + \varepsilon_z \cdot \varepsilon_\Theta) \cdot \left(\frac{\partial \varepsilon_{RZ}}{\partial Z} + \frac{(\varepsilon_R - \varepsilon_\Theta) + \frac{\partial \varepsilon_{R\Theta}}{\partial \Theta}}{R} + \frac{\partial \varepsilon_R}{\partial R} \right)}{-\varepsilon_{R\Theta}^2 \cdot \varepsilon_Z + 2 \cdot \varepsilon_{RZ} \cdot \varepsilon_{R\Theta} \cdot \varepsilon_{\Theta Z} - \varepsilon_R \cdot \varepsilon_{\Theta Z}^2 - \varepsilon_{RZ}^2 \cdot \varepsilon_\Theta + \varepsilon_R \cdot \varepsilon_\Theta \cdot \varepsilon_Z} \quad (3.34)$$

$$H_\Theta = \frac{(\varepsilon_{R\Theta} \cdot \varepsilon_{\Theta Z} - \varepsilon_{RZ} \varepsilon_\Theta) \cdot \left(\frac{\partial \varepsilon_Z}{\partial Z} + \frac{\varepsilon_{RZ} + \frac{\partial \varepsilon_{\Theta Z}}{\partial \Theta}}{R} + \frac{\partial \varepsilon_{RZ}}{\partial R} \right)}{-\varepsilon_{R\Theta}^2 \cdot \varepsilon_Z + 2 \cdot \varepsilon_{RZ} \cdot \varepsilon_{R\Theta} \cdot \varepsilon_{\Theta Z} - \varepsilon_R \cdot \varepsilon_{\Theta Z}^2 - \varepsilon_{RZ}^2 \cdot \varepsilon_\Theta + \varepsilon_R \cdot \varepsilon_\Theta \cdot \varepsilon_Z} \\ - \frac{(-\varepsilon_{R\Theta} \cdot \varepsilon_Z + \varepsilon_{RZ} \varepsilon_{\Theta Z}) \cdot \left(\frac{\partial \varepsilon_{\Theta Z}}{\partial Z} + \frac{2 \cdot \varepsilon_{R\Theta} + \frac{\partial \varepsilon_\Theta}{\partial \Theta}}{R} + \frac{\partial \varepsilon_{R\Theta}}{\partial R} \right)}{-\varepsilon_{R\Theta}^2 \cdot \varepsilon_Z + 2 \cdot \varepsilon_{RZ} \cdot \varepsilon_{R\Theta} \cdot \varepsilon_{\Theta Z} - \varepsilon_R \cdot \varepsilon_{\Theta Z}^2 - \varepsilon_{RZ}^2 \cdot \varepsilon_\Theta + \varepsilon_R \cdot \varepsilon_\Theta \cdot \varepsilon_Z} \quad (3.35)$$

$$H_Z = \frac{(\varepsilon_{R\Theta} \cdot \varepsilon_{\Theta Z} - \varepsilon_{RZ} \varepsilon_\Theta) \cdot \left(\frac{\partial \varepsilon_{RZ}}{\partial Z} + \frac{(\varepsilon_R - \varepsilon_\Theta) + \frac{\partial \varepsilon_{R\Theta}}{\partial \Theta}}{R} + \frac{\partial \varepsilon_R}{\partial R} \right)}{-\varepsilon_{R\Theta}^2 \cdot \varepsilon_Z + 2 \cdot \varepsilon_{RZ} \cdot \varepsilon_{R\Theta} \cdot \varepsilon_{\Theta Z} - \varepsilon_R \cdot \varepsilon_{\Theta Z}^2 - \varepsilon_{RZ}^2 \cdot \varepsilon_\Theta + \varepsilon_R \cdot \varepsilon_\Theta \cdot \varepsilon_Z} \\ - \frac{(-\varepsilon_{RZ}^2 + \varepsilon_R \cdot \varepsilon_{\Theta Z}) \cdot \left(\frac{\partial \varepsilon_{\Theta Z}}{\partial Z} + \frac{2 \cdot \varepsilon_{R\Theta} + \frac{\partial \varepsilon_\Theta}{\partial \Theta}}{R} + \frac{\partial \varepsilon_{R\Theta}}{\partial R} \right)}{-\varepsilon_{R\Theta}^2 \cdot \varepsilon_Z + 2 \cdot \varepsilon_{RZ} \cdot \varepsilon_{R\Theta} \cdot \varepsilon_{\Theta Z} - \varepsilon_R \cdot \varepsilon_{\Theta Z}^2 - \varepsilon_{RZ}^2 \cdot \varepsilon_\Theta + \varepsilon_R \cdot \varepsilon_\Theta \cdot \varepsilon_Z} \\ - \frac{(\varepsilon_{R\Theta} \cdot \varepsilon_{\Theta Z} - \varepsilon_{RZ} \varepsilon_\Theta) \cdot \left(\frac{\partial \varepsilon_{RZ}}{\partial Z} + \frac{(\varepsilon_R - \varepsilon_\Theta) + \frac{\partial \varepsilon_{R\Theta}}{\partial \Theta}}{R} + \frac{\partial \varepsilon_R}{\partial R} \right)}{-\varepsilon_{R\Theta}^2 \cdot \varepsilon_Z + 2 \cdot \varepsilon_{RZ} \cdot \varepsilon_{R\Theta} \cdot \varepsilon_{\Theta Z} - \varepsilon_R \cdot \varepsilon_{\Theta Z}^2 - \varepsilon_{RZ}^2 \cdot \varepsilon_\Theta + \varepsilon_R \cdot \varepsilon_\Theta \cdot \varepsilon_Z} \quad (3.36)$$

$$\begin{aligned}
 & -\frac{(-\varepsilon_{R\Theta}^2 + \varepsilon_R \varepsilon_\Theta) \cdot \left(\frac{\partial \varepsilon_Z}{\partial Z} + \frac{\varepsilon_{RZ} + \frac{\partial \varepsilon_{\Theta Z}}{\partial \Theta}}{R} + \frac{\partial \varepsilon_{RZ}}{\partial R} \right)}{-\varepsilon_{R\Theta}^2 \cdot \varepsilon_Z + 2 \cdot \varepsilon_{RZ} \cdot \varepsilon_{R\Theta} \cdot \varepsilon_{\Theta Z} - \varepsilon_R \cdot \varepsilon_{\Theta Z}^2 - \varepsilon_{RZ}^2 \cdot \varepsilon_\Theta + \varepsilon_R \cdot \varepsilon_\Theta \cdot \varepsilon_Z} \\
 & -\frac{(\varepsilon_{RZ} \cdot \varepsilon_{R\Theta} - \varepsilon_R \cdot \varepsilon_{\Theta Z}) \cdot \left(\frac{\partial \varepsilon_{\Theta Z}}{\partial Z} + \frac{2 \cdot \varepsilon_{R\Theta} + \frac{\partial \varepsilon_\Theta}{\partial \Theta}}{R} + \frac{\partial \varepsilon_{R\Theta}}{\partial R} \right)}{-\varepsilon_{R\Theta}^2 \cdot \varepsilon_Z + 2 \cdot \varepsilon_{RZ} \cdot \varepsilon_{R\Theta} \cdot \varepsilon_{\Theta Z} - \varepsilon_R \cdot \varepsilon_{\Theta Z}^2 - \varepsilon_{RZ}^2 \cdot \varepsilon_\Theta + \varepsilon_R \cdot \varepsilon_\Theta \cdot \varepsilon_Z}
 \end{aligned}$$

Se toman las deformaciones en los nodos y las derivadas parciales se realizan a nivel elemental, tal y como se ha hecho en 2D. Una vez se calculan las 3 componentes para cada nodo, es posible calcular y representar dW así como el módulo de su gradiente.

$$dW = \vec{H} \cdot \vec{dx} = [H_R \ H_\Theta \ H_Z] \cdot \begin{bmatrix} dr \\ d\theta \\ dz \end{bmatrix} \quad (3.37)$$

Donde $dr = \frac{r_{nodal}}{\sqrt{r_{nodal}^2 + \theta_{nodal}^2 + z_{nodal}^2}}$, $d\theta = \frac{\theta_{nodal}}{\sqrt{r_{nodal}^2 + \theta_{nodal}^2 + z_{nodal}^2}}$ y $dz = \frac{z_{nodal}}{\sqrt{r_{nodal}^2 + \theta_{nodal}^2 + z_{nodal}^2}}$.

3.4. Proceso de Segmentación

Una vez se tienen todas las variables y los módulos de sus gradientes, el siguiente paso es obtener la imagen entera de la geometría. Es necesario recordar que en el modelo de elementos finitos tan solo se analiza un cuarto de la geometría debido a su simetría y por ello se debe reconstruir la sección entera. Mediante un tratamiento de imagen es posible representar cualquier variable, guardar la imagen, duplicarla, representar la simétrica y unirla, tal y como se muestra en la *Figura 3.5* (información ampliada en el *Anexo V: Proceso de Segmentado*). La diferencia entre haber simulado la sección entera o solo la mitad, como en este caso, ahorra muchas horas de cálculos (más de 8 horas) y de post-procesos y tan solo conlleva un error en la medida de áreas menor al 0.6 %. En el modelo basado en la geometría de IVUS no es necesario realizar esta operación, ya que en este caso se modela la totalidad de la imagen.

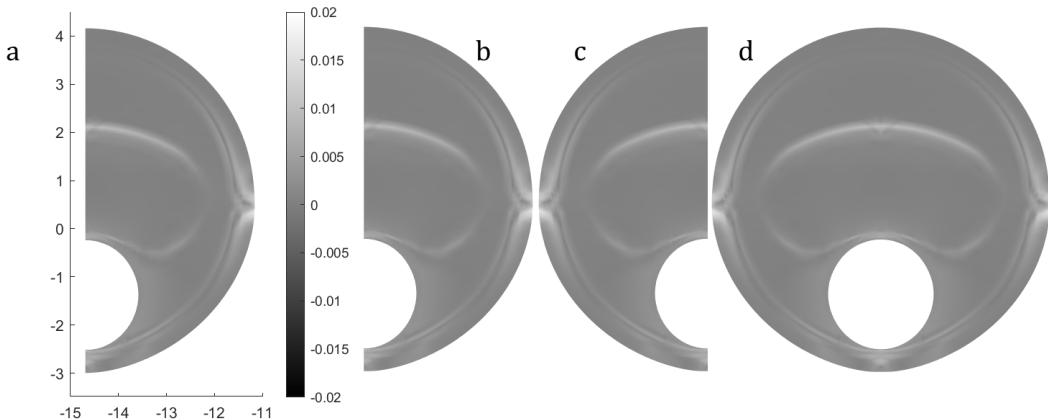


Figura 3.5: a) Representación de $|\nabla \varepsilon_y|$ inicial, b) Recorte de la representación, c) Imagen Simétrica, d) Unión.

El siguiente paso se basa en combinar variables que permitan una mejor visualización del contorno del núcleo lipídico y la capa fibrótica. Para ello se utilizan los módulos de los gradientes de las variables calculadas, esto permite intensificar los contrastes de deformación entre distintos materiales y se resaltan las fronteras de las diferentes partes. Para el método de segmentado es necesario utilizar como input una imagen en la que se pueda discernir el contorno del núcleo lipídico y el lumen. Este método utiliza una segmentación *Watershed* desarrollada en Matlab (para más información ver *Anexo V*) en la que, tras unos procesos automáticos basados en morfología matemática que permite detectar

contornos dentro de la imagen, se obtiene la segmentación del núcleo lipídico y el lumen. De cada segmentado se puede calcular el área del núcleo lipídico y medir la distancia entre el núcleo lipídico y el lumen, que es la medida de la capa fibrótica. Dependiendo de la combinación de las variables utilizadas aparecerán los contrastes en diferentes zonas de la arteria, lo que también permitirá la segmentación de otros tejidos como el tejido sano o la capa fibrótica.

Dentro del proceso de segmentado ha sido necesario introducir 3 parámetros variables. Es necesario modificarlos manualmente para obtener la segmentación de la zona deseada, el resto del proceso es automático. Los parámetros definidos son:

1. **Límites en la representación de variables:** en el paso previo al Watershed es necesario introducir límites a la hora de representar las variables. En la *Figura 3.6* se muestra la misma variable con diferentes límites. Además, es posible representar las variables con escalas de grises o con colores fríos y dependiendo de los límites de representación los contornos se marcarán más o menos. En todo el trabajo se ha utilizado la representación de variables en escala de grises.

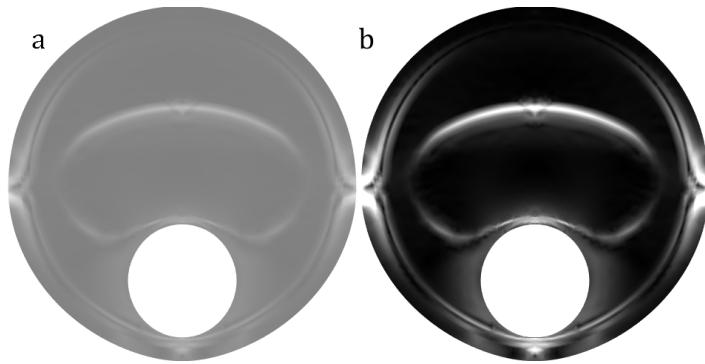


Figura 3.6: Diferentes límites para la representación de $|\nabla \varepsilon_y|$: a) Límite superior de 0.02 e inferior de -0.02, b) límite superior de 0.008 e inferior de 0.

2. **Tamaño del “disco de segmentado”:** dentro del proceso de *Watershed*, en uno de los primeros pasos se realiza una distorsión de la imagen con un disco de tamaño a definir. Dependiendo del tamaño del disco es posible resaltar unas partes u otras. En la *Figura 3.7* se muestra el efecto del proceso.

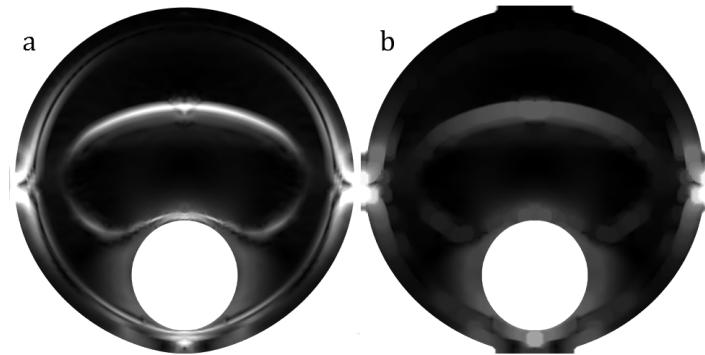


Figura 3.7: Efecto del disco de segmentado de tamaño 15 píxeles antes a) y después b) sobre la representación de $|\nabla \varepsilon_y|$.

3. **Parámetros de suavizado:** tras el proceso de segmentado se obtiene el núcleo lipídico. Sin embargo, al superponer el núcleo segmentado con la imagen original (o con el núcleo real), se observa que el resultado obtenido no es perfecto. Por lo tanto se introduce un proceso de suavizado donde primero se agranda el núcleo lipídico ligeramente y a continuación se realiza una convolución (de un número de píxeles a determinar) que permite suavizar el contorno, tal y como se aprecia en la *Figura 3.8*.

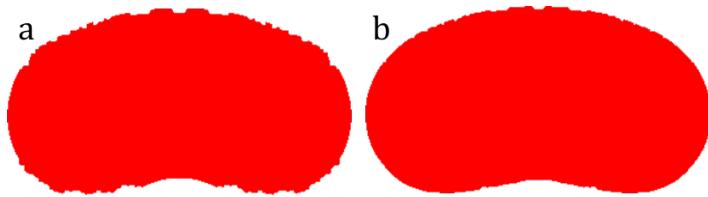


Figura 3.8: Núcleo lipídico a) Tras el segmentado, b) Despues del suavizado.

A la hora de hacer mediciones de distancias Matlab solo es capaz de medir el número de píxeles que hay entre objetos binarios, y al medir área devuelve el número de píxeles contenidos en el interior del objeto. Por lo tanto, es necesario introducir una escala para convertir las medidas de píxeles a unidades de longitud (mm). Se propone tomar como referencia el diámetro del lumen señalado en la *Figura 3.9.a* medido en el primer *step* en Abaqus CAE e igualarlo a los píxeles medidos en el mismo diámetro en la imagen *Figura 3.9.b*, lo que permite obtener la escala entre píxeles-mm. En la parte final del *Anexo V* se detalla más el cálculo de esta escala. El problema de imponer una escala no aparecería si se trabajara con imágenes IVUS reales, ya que la escala vendría dada por la resolución de la imagen.

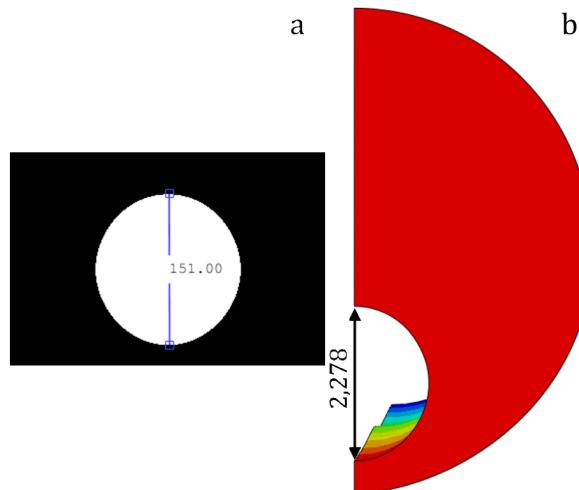


Figura 3.9: a) Diámetro medido [pixeles] en la imagen de Matlab, b) Diámetro medido [mm] en el modelo de Abaqus CAE donde se representan las coordenadas en el eje Y.

Este proceso de definición de la escala es el proceso es el mismo tanto si se parte del modelo 2D como si se parte del 3D. La única salvedad es que en el modelo 3D se tiene que situar la representación en la vista frontal para poder comparar con los resultados bidimensionales, tal y como se muestran en la *Figura 3.10*:

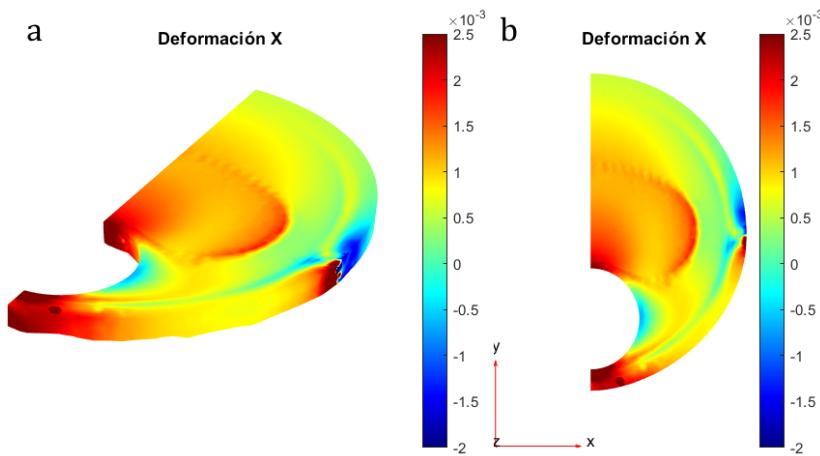


Figura 3.10: a) Vista 3D del modelo, b) Vista frontal del $|\nabla \varepsilon_x|$.

Al aplicar el proceso a la geometría real obtenida a partir de la reconstrucción del IVUS la escala se obtiene con la misma metodología. En la *Figura 3.11.a* se observa cómo se mide el diámetro del lumen en píxeles en Matlab y en la *Figura 3.11.b* cómo se mide el diámetro en mm en Abaqus CAE. Con ello se obtiene la relación píxeles-mm.

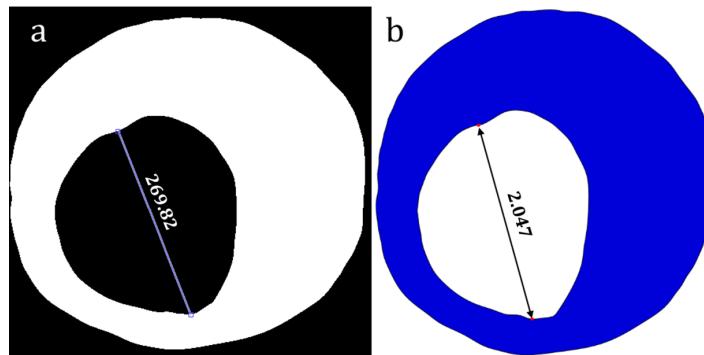


Figura 3.11: a) Diámetro del lumen medido en píxeles en Matlab, b) Diámetro del lumen medido en mm en Abaqus.

Por otro lado, en el paso de importar los nodos de Abaqus a Matlab se ha programado que a cada nodo tenga asignado un número en función del material al que pertenece. Por ejemplo, *Adventicia=1*, *Media=2*, *Núcleo Lipídico=3*... Con ello se puede representar y medir las diferentes partes reales y comparar con los resultados obtenidos tras el segmentado. En la *Figura 3.12* se muestran las diferentes partes reales en la geometría idealizada de 150 μm de espesor.

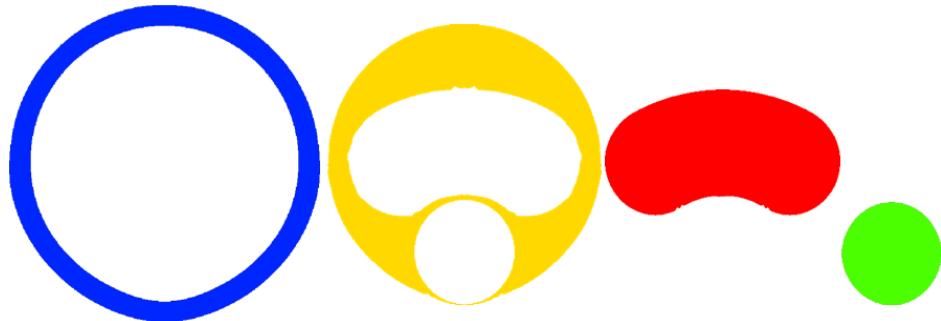


Figura 3.12: Diferentes partes de la arteria coronaria con placa de ateroma con espesor de 150 μm . En azul aparece el tejido sano (Advenciticia+Media), en amarillo la Placa o capa fibrótica, en rojo el núcleo lipídico y en verde el lumen.

3.5. Cuantificación de los Errores

Con este proceso es posible medir el espesor de la capa fibrótica, que es un criterio clave en la práctica clínica para determinar si una placa es vulnerable, y medir el área del núcleo lipídico. Sin embargo, con el fin de cuantificar la segmentación realizada se realizan diferentes cálculos que permitan valorar la validez del método desarrollado. El primer valor calculado, y que más importancia tiene, es el error relativo del espesor de la capa fibrótica (*Ecuación 3.38*), donde se compara el espesor medido con el espesor real cuyo valor se obtiene del software Abaqus. Puede ocurrir que el proceso de segmentado proporcione una medida del espesor de la capa fibrótica correcta pero la forma del núcleo obtenido sea muy diferente al real. Por este motivo se definen dos índices que cuantifican el área y su forma. El primer índice (*Ecuación 3.39*) hace referencia al área real correctamente segmentada, mientras que el segundo índice (*Ecuación 3.40*) se refiere al área segmentada que no lo es realmente. Por lo tanto, interesa que en el primer índice se obtenga un valor cercano al 100 %, mientras que en el segundo índice sea próximo a cero. En la *Figura 3.13* se muestra un ejemplo donde se superponen el núcleo real y el segmentado, donde en blanco se muestra el área que pertenece tanto al núcleo real como al segmentado, en morado el núcleo real que no se ha segmentado y en verde el núcleo segmentado que no pertenece al real.

$$\text{Error}_{\text{CapaFibrotica}} = \left| \frac{\text{EspesorMedido} - \text{EspesorReal}}{\text{EspesorReal}} \right| \cdot 100 \quad (3.38)$$

$$\text{Indice}_1 = \frac{\text{Area real que ha sido segmentada}}{\text{Area real}} \cdot 100 \quad (3.39)$$

$$\text{Indice}_2 = \frac{\text{Area ficticia que ha sido segmentada}}{\text{Area real}} \cdot 100 \quad (3.40)$$



Figura 3.13: Superposición del núcleo lipídico segmentado tras combinar $|\nabla \varepsilon_y|$ con $|\nabla \varepsilon_{xy}|$ y el núcleo real. En blanco se observa la unión entre el núcleo real y el segmentado, el morado el núcleo real no segmentado y en verde el núcleo segmentado no real.

4. Resultados

En este apartado se muestran los resultados del proceso de segmentación aplicado a diferentes geometrías (idealizadas, no idealizadas e IVUS) y se compara el método 2D basado en la hipótesis de deformación plana con el método desarrollado en 3D.

4.1. Geometrías Idealizadas. Análisis 2D

Se realiza el proceso de segmentado sobre las geometrías idealizadas con los diferentes espesores. El objetivo principal es el de segmentar el núcleo lipídico y el lumen. Tal y como se ha explicado se representan diferentes variables y la combinación de ellas se introducen como input del método watershed. En la *Figura 4.1* se representa el proceso de segmentación al combinar la variable módulo del gradiente de la deformación en X ($|\nabla \varepsilon_x|$) (*Figura 4.1.a*) y el módulo del gradiente de la deformación en Y ($|\nabla \varepsilon_y|$) (*Figura 4.1.b*). Ambas imágenes se unen y tras el proceso de segmentado se representan las diferentes partes obtenidas tal y como se observa en la *Figura 4.1.c* donde se aprecia el núcleo lipídico y el lumen. Seguidamente se selecciona únicamente el núcleo lipídico (*Figura 4.1.d*) y se realiza el proceso de suavizado (*Figura 4.1.e*). Por último, en la *Figura 4.1.f* se observa la superposición entre el núcleo segmentado y el núcleo real.

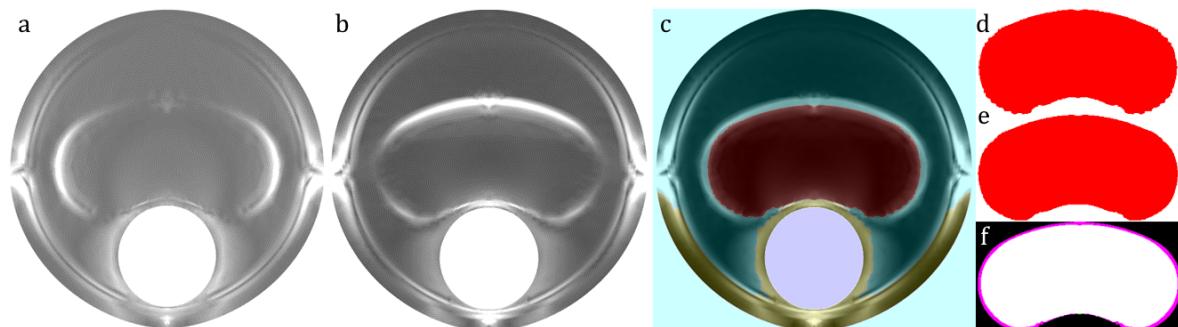


Figura 4.1: Proceso de segmentado en la geometría idealizada de espesor de 150 μm : a) $|\nabla \varepsilon_x|$, b) $|\nabla \varepsilon_y|$, c) Unión de las anteriores y partes segmentadas, d) Núcleo lipídico segmentado, e) Núcleo lipídico segmentado tras el proceso de suavizado y f) Comparación entre núcleo lipídico segmentado (blanco) y real (morado).

Una vez segmentado el núcleo lipídico y el lumen se realiza la medida de la capa fibrótica, tal y como se muestra en la *Figura 4.2*. Se calcula el error relativo entre la medida tomada y la real. Además, se calculan los índices 1 y 2 previamente definidos (*Ecuaciones 3.39 y 3.40*). La distancia obtenida está escalada y se muestra en milímetros.

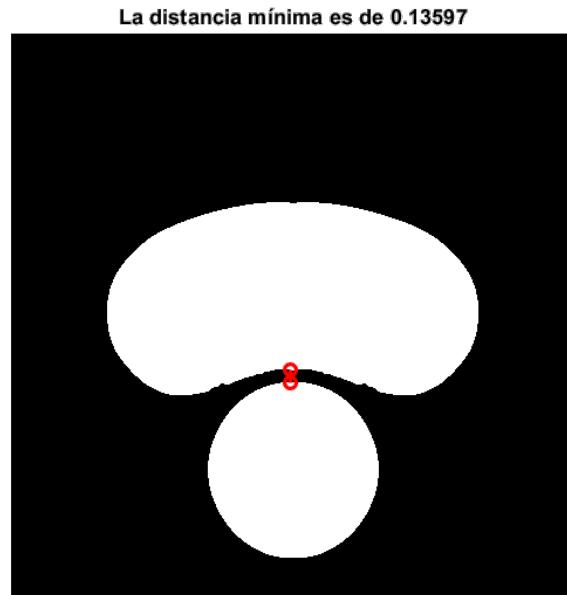


Figura 4.2: Medida [mm] automática del espesor de la capa fibrótica.

Este mismo proceso se ha seguido con las 131 combinaciones de variables a analizar y se han realizado dos tablas para cada espesor. En la primera tabla se representa el error relativo (*Ecuación 3.38*) del espesor de la capa fibrótica. La segunda tabla introduce los dos índices referentes al área (*Ecuaciones 3.39 y 3.40*). En las *Figuras 4.3, 4.4, 4.5, 4.6, 4.7 y 4.8* se muestran todas las tablas para los diferentes espesores, donde la letra *U* hace referencia a los desplazamientos, ε a las deformaciones y $|\nabla X|$ al módulo del gradiente de la variable *X*. En todas las tablas se ha seguido un código de colores:

- Rojo:** para las variables o combinaciones de variables que no permiten realizar el segmentado ni distinguir visualmente el núcleo lipídico.
- Azul:** aquellas variables o combinaciones de ellas que no obtienen el núcleo lipídico tras el segmentado *watershed* pero sí es posible predecir a simple vista su contorno aproximado.
- Verde:** para los casos en los que la segmentación *watershed* permite la obtención del núcleo lipídico. En la primera tabla se introduce el error de la medida de la capa fibrótica y en la segunda se apuntan los índices *Indice1/Indice2*.

	$ \nabla U_x $	$ \nabla U_y $	$ \nabla U $	$ \nabla E_x $	$ \nabla E_y $	$ \nabla E_{xy} $	$ \nabla E_{max} $	$ \nabla E_{med} $	$ \nabla E_{min} $	$ \nabla E_{Mises} $	$ \nabla E_{Tresca} $	$ \nabla F_A $	$ \nabla \varepsilon_x $	$ \nabla \varepsilon_y $	$ \nabla \varepsilon_{xy} $	$ \nabla \varepsilon_{max} $	$ \nabla \varepsilon_{med} $	$ \nabla \varepsilon_{min} $	$ \nabla \varepsilon_{Mises} $	$ \nabla \varepsilon_{Tresca} $	$ \nabla dW $	$ \nabla dW $		
$ \nabla U_x $	164.72%	92.33%		5.92%		191.29%	116.82%		218.41%	218.41%	218.41%	37.55%												
$ \nabla U_y $		130.27%	218.41%	180.53%		193.92%	145.93%		111.37%	111.37%	111.37%	48.48%	26.52%											
$ \nabla U $	146.90%	254.97%	37.54%		15.32%	201.70%	120.45%		249.73%	249.73%	249.73%	62.09%	70.23%	15.32%										
$ \nabla E_x $				0.00%		48.49%	97.17%		0.00%	0.00%	0.00%	0.00%									60.28%			
$ \nabla E_y $				0.00%	22.09%	21.36%	15.32%		8.31%	8.31%	8.31%	60.28%	48.49%	48.49%	2.03%	48.49%								
$ \nabla E_{xy} $						24.21%	106.74%					6.17%	17.09%											
$ \nabla E_{max} $							170.33%	106.73%				15.32%	8.87%	29.86%	192.16%	71.99%								
$ \nabla E_{med} $								148.98%		249.73%	249.73%	249.73%	84.99%	215.70%	102.81%	144.93%	219.37%							
$ \nabla E_{min} $																								
$ \nabla E_{Mises} $										26.52%	26.52%	26.52%	117.41%	184.84%							20.50%			
$ \nabla E_{Tresca} $											26.52%	26.52%	117.41%	184.84%								20.50%		
$ \nabla F_A $												26.52%	26.52%	117.41%	184.84%									
$ \nabla \varepsilon $													17.26%		52.67	52.67%								
$ \nabla \varepsilon_x $													0.00%	0.00%	8.87%	8.87%								
$ \nabla \varepsilon_y $																								
$ \nabla \varepsilon_{xy} $																								
$ \nabla \varepsilon_{max} $																								
$ \nabla \varepsilon_{med} $																								
$ \nabla \varepsilon_{min} $																								
$ \nabla \varepsilon_{Mises} $																								
$ \nabla \varepsilon_{Tresca} $																								
$ \nabla dW $																							40.18%	

Figura 4.3: Tabla de segmentación de 131 combinaciones posibles en la geometría idealizada de 65 μm de espesor. Se presenta el error de medida del espesor de la capa fibrótica.

Estudio de la viabilidad de la segmentación de placas de ateroma vulnerables en coronarias a partir del campo de desplazamientos compatibles con imágenes IVUS

	vUx	vUy	vU	vEx	vEy	vEmax	vEmed	vEmin	vEvMises	vETresca	vFa	vEr	vEe	vEre	dW	v dW	
vUx	97,94/4,98	90,91/7,63			99,84/13,87	97,09/12,5	98/13,37		96,3/13,1	96,9/13,1	96,9/13,1		99,1/13,76				
vUy		98,3/7,82	96,31/8,96	97,42/15,84		96,6/14,33	97,92/13,54		90,42/5,52	90,42/5,52	90,42/5,52	90,42/5,52	98,43/3,14	95,42/8,65	96,2/13,09		
vU			98,05/6,78	93,67/10,23	98,92/14,32	98,11/11,82	96,71/13,85	98,13,51	96,35/13,55	96,35/13,55	96,35/13,55	96,35/13,55	95,76/10,31	98,12/12,23	97,33/5,61	99,15/5,92	
vEx				99,83/4,43		98,98/6,84	95,26/5,59		99,45/5,69	99,45/5,69	99,45/5,69	99,45/5,69	99,86/6,32			94,8/3,62	
vEy					99,68/2,87	99,69/6,39	99,58/3,99	99,66/3,32	99,77/5,22	99,77/5,25	99,77/5,25	98,06/13,46	86,37/3,02	99,34/11,83	99,37/1,83	96,23/2,36	
vEmax						99,52/3,89	98,52/10,89						98,8/4,91	99,78/6,5	98,77/6,17	97,5/16,3	
vEmed						96,8/14,05	97,64/9,88						98,45/6,37	96,76/5,24	97,92/6,84	97,91/7,51	
vEmin							96,44/1,71		95,81/9,61	95,81/9,61	95,81/9,61	95,81/9,61				93,96/0,22	
vEvMises									80,72/0,55	80,72/0,55	80,72/0,55	80,72/0,55	96,15/10,63	97,37/5,51	95,45/6,09	97,38/17,98	
vETresca									80,72/0,55	80,72/0,55	80,72/0,55	80,72/0,55	96,15/10,63	97,37/5,51	95,45/6,09	97,38/17,98	
vFa										80,72/0,55	80,72/0,55	80,72/0,55	80,72/0,55	97,37/5,51	95,45/6,09		97,38/17,98
vEr											99,7/12,41					98,6/13,64	
vEe												99,87/9,5	99,15/5,17	99,49/2,45	97,06/1,91		
vEre																87,69/0,95	
dW																	
v dW																	

Figura 4.4: Tabla de segmentación de 131 combinaciones posibles en la geometría idealizada de 65 μm de espesor. Se presentan los índices 1/índice 2.

	vUx	vUy	vU	vEx	vEy	vEmax	vEmed	vEmin	vEvMises	vETresca	vFa	vEr	vEe	vEre	dW	v dW
vUx	22,66%	18,23%			2,5%	9,77%	20,17%		11,72%	11,72%	8,91%	0,00%				
vUy		26,69%			32,14%	10,53%		41,83%	41,83%	41,83%	39,98%	0,00%			4,52%	
vU			35%	7,83%	46,70%	62,70%	65,25%		84,42%	84,42%	84,42%	25,19%	11,11%	41,83%	25,42%	2,00%
vEx				0,00%	5,41%	11,11%	0,00%	148,35%	148,35%	148,35%	148,35%	148,35%	11,11%	11,11%	53,46%	28,90%
vEy					0,00%	2,00%	11,11%		0,00%	0,00%	0,00%	0,00%	0,00%	112,20%	0,00%	22,23%
vEmax						11,93%	14,99%	2,62%		0,50%	0,50%	0,50%	53,27%	10,32%	4,11%	44,44%
vEmed							0,00%	6,22%	6,22%	6,22%	6,22%	6,22%	12,59%	3,35%	16,75%	11,11%
vEmin								33,33%	33,33%	33,33%	33,33%	33,33%	26,85%	22,22%	0,00%	66,67%
vEvMises									12,79%	12,79%	12,79%	12,79%	12,79%	9,93%	12,59%	16,33%
vETresca									12,79%	12,79%	12,79%	12,79%	12,79%	9,93%	12,59%	16,33%
vFa										12,79%	12,79%	12,79%	12,79%	12,79%	12,59%	16,33%
vEr											0,00%	2,00%	0,00%	2,55%	0,00%	28,90%
vEe												0,00%	2,55%	0,00%		42,9%
vEre																9,66%
dW																
v dW																

Figura 4.5: Tabla de segmentación de 131 combinaciones posibles en la geometría idealizada de 150 μm de espesor. Se presenta el error de medida del espesor de la capa fibrótica.

	vUx	vUy	vU	vEx	vEy	vEmax	vEmed	vEmin	vEvMises	vETresca	vFa	vEr	vEe	vEre	dW	v dW	
vUx	97,61/0,76	97,92/0,37			99,37/4,78			99,03/5,49	99,68/6,52	98,81/3,48	98,81/3,48	84,01/1	99,29/3,42				
vUy		94,95/2,92	93,27/0,21			95,95/10,09	99,85/9,76		98,9/2,25	98,9/2,25	98,9/2,25	98,9/2,25	94,79/1,11	90,59/0		98,2/7,86	
vU			97,79/3,66	99,58/5,75	98,62/4,91	98,03/6,45	97,34/10,1		97,18/4,52	97,18/4,52	97,18/4,52	97,18/4,52	98,4/4,91	99,26/4,65	91,91/5,49	98,49/5,93	
vEx				92,21/0,01	96,9/4,95	93,1/0		86,81/0,19		93,29/4,88	93,29/4,88	93,29/4,88	93,29/4,88	94,49/4,69	92,07/0,01	88,97/0,65	95,19/6,79
vEy					97,22/0,57	94,92/0,45	94,99/0,38	95,51/9		94,58/0,28	94,58/0,28	94,58/0,28	94,58/0,28	90,74/0	93,82/0,05	92,67/0,02	93,5/0,04
vEmax						96,54/2,38		95,86/0		99,6/3,2	99,6/3,2	99,6/3,2	99,6/3,2	93,41/0,01	93,86/0,4	95,01/0	93,68/0
vEmed							87,09/0						90,94/0	92,73/0,23	93,11/0,4	94,51/7,33	
vEmin								88,99/0,89		70,24/7,08	70,24/7,08	70,24/7,08	70,24/7,08	80,12/0,96	87,51/0,1	99,54/5,3	96,97/9,85
vEvMises									99,05/1,58	99,05/1,58	99,05/1,58	99,05/1,58	94,37/4,58	91,12/0,01	94,2/0	96,17/8,95	
vETresca									99,05/1,58	99,05/1,58	99,05/1,58	99,05/1,58	94,37/4,58	91,12/0,01	94,2/0	96,17/8,95	
vFa										99,05/1,58	99,05/1,58	99,05/1,58	99,05/1,58	94,37/4,58	91,12/0,01	94,2/0	96,17/8,95
vEr											93,2/0	94,9/7,53					
vEe												93,52/0	96,17/0,7	92,96/0		98,7/0	
vEre													88,9/3,2				
dW																	
v dW																93,12/5,99	

Figura 4.6: Tabla de segmentación de 131 combinaciones posibles en la geometría idealizada de 150 μm de espesor. Se presentan los índices 1/índice 2.

	vUx	vUy	vU	vEx	vEy	vEmax	vEmed	vEmin	vEvMises	vETresca	vFa	vEr	vEe	vEre	dW	v dW	
vUx	35,92%	40,91%	38,67%	3,1%	24,79%	35,50%	40,92%		61,21%	61,21%	61,21%	36,11%	1,19%	3,31%	2,19%	6,80%	
vUy			34,78%	3,1%	66,45%	42,79%	38,68%		61,22%	61,22%	61,22%	21,82%	6,44%	26,66%	7,07%	2,19%	
vU				1,18%	9,82%	3,31%	24,79%	34,78%	38,68%		61,22%	61,22%	61,22%	36,12%	1,18%	2,07%	36,08%
vEx					28,92%	35,96%	2,19%	2,07%		0,00%	0,00%	0,00%	2,19%	2,07%	24,13%	35,54%	8,12%
vEy						11,04%	23,96%	26,66%	3,31%		28,92%	28,92%	28,92%	28,92%	19,42%	3,31%	18,95%
vEmax							25,81%	25,81%	25,81%				25,81%	32,53%	21,82%	23,96%	32,53%
vEmed								35,54%	35,54%	30,89%	30,89%	35,54%	28,92%	28,92%	35,54%	4,93%	
vEmin									37,68%	3,18%	3,18%	3,18%	1,42%	35,92%	35,50%	36,11%	
vEvMises										2,64%	2,64%	2,64%	26,07%	32,08%	26,07%	28,57%	6,38%
vETresca											2,64%	2,64%	2,64%	26,07%	32,08%	26,07%	28,57%
vFa											2,64%	2,64%	2,64%	26,07%	32,08%	26,07%	28,57%
vEr												2,64%	2,64%	2,64%	26,07%	32,08%	
vEe													34,92%	14,09%	35,50%	35,50%	
vEre														3,07%	23,62%	2,67%	
dW																2,53%	
v dW																	

Figura 4.7: Tabla de segmentación de 131 combinaciones posibles en la geometría idealizada de 30

	$ \nabla U_x $	$ \nabla U_y $	$ \nabla U $	$ \nabla \varepsilon_x $	$ \nabla \varepsilon_y $	$ \nabla \varepsilon_{max} $	$ \nabla \varepsilon_{med} $	$ \nabla \varepsilon_{min} $	$ \nabla \varepsilon_{visibles} $	$ \nabla \varepsilon_{tressa} $	$ \nabla F_A $	$ \nabla \varepsilon_r $	$ \nabla \varepsilon_\theta $	$ \nabla \varepsilon_\theta $	d_W	$ \nabla d_W $
$ \nabla U_x $	90.18/2.64	93.65/2.15	92.02/2.24	95.06/2.62	97.54/2.12	98.86/3.02	99.77/4.87	98.78/4.18	98.78/4.18	92.2/4.1	96.16/2.06	95.21/2.53	99.91/1.41	97.53/9.62		
$ \nabla U_y $			98.72/1.78	91.26/0.02	93.88/3.26	99.75/4.93	99.06/3.31	98.23/3.26	98.23/3.26	98.8/2.1	94.12/0.81	92.74/2.84	98.34/4.33	91.96/0.79		
$ \nabla U $			89.71/0.22	90.55/1.02	96.17/2.47	96.4/1.88	99.16/3.2	99.97/3.5	98.97/4.31	98.97/4.31	98.85/3.18	96.62/2.1	93.16/0.76	99.92/7.16	91.5/0.43	
$ \nabla \varepsilon_x $				93.74/1.72	92.38/2.55	97.01/2.68	81.29/0	94.34/0.06	94.34/0.06	85.59/1.36	88.1/0	96.66/2.14	91.31/1.13	88.55/0.7		
$ \nabla \varepsilon_y $					90.62/1.02	92.72/1.21	90.31/1.38	85.47/0.02	90.33/1.18	90.33/1.18	97.72/2.36	92.07/0.02	89.5/0.66	89.34/0	82.55/0.2	
$ \nabla \varepsilon_{max} $						92.34/2.15	91.6/1.52	92.91/1.45	91.06/1.42	91.06/1.42	94.41/2.59	89.82/0.98	90.34/1.38	92.57/2.13	93.29/6.78	
$ \nabla \varepsilon_{med} $							99.2/3.2	88.43/1.07	99.34/3.61	99.34/3.61	89.87/1.48	90.16/1.55	93.88/1.18	98.29/4.58	98.59/7.27	
$ \nabla \varepsilon_{min} $								98.81/3.48	94.9/1.03	94.9/1.03	94.9/1.03	80.86/0.13	98.91/2.32	97.19/2.23	97.22/4.11	98.83/7.67
$ \nabla \varepsilon_{visibles} $									98.58/1.78	98.58/1.78	98.58/1.78	98.22/4.49	94.23/5.97			
$ \nabla \varepsilon_{tressa} $									98.58/1.78	98.58/1.78	98.58/1.78	98.22/4.49	94.23/5.97			
$ \nabla F_A $										98.58/1.78	98.58/1.78	98.22/4.49	94.23/5.97			
$ \nabla \varepsilon_r $											96.82/3.69	94.75/0.8	99.81/5.32	99.81/5.32	99.8/0.49	
$ \nabla \varepsilon_\theta $												91.56/0.04	91.7/1.08	90.95/0	83.54/0	
d_W													91.38/1.69	88.39/0.9	91.94/5.15	
$ \nabla d_W $														81.01/0.74		
															84.32/0.16	

Figura 4.8: Tabla de segmentación de 131 combinaciones posibles en la geometría idealizada de 300 μm de espesor. Se presentan los índices 1/índice 2.

En la Figura 4.3 se observa que hay muy pocas variables que permitan realizar una correcta segmentación para el espesor de 65 μm y los errores relativos alcanzados son muy altos. Lo contrario se observa en la Figura 4.7, donde el espesor de 300 μm es posible segmentarlo con una gran cantidad de combinaciones. Se aprecia que variables como $|\nabla \varepsilon_\theta|$ o $|\nabla \varepsilon_y|$ permiten una correcta segmentación, tanto solas como en combinación con otras. Por contra $|\nabla \varepsilon_{min}|$ no es capaz de permitir una visualización del contorno del lípido. Los errores en la medida de la capa fibrótica son mayores que los obtenidos en las áreas.

Dependiendo de las variables a combinar y de los límites de representación de las mismas, es posible realizar la segmentación de las diferentes partes de la arteria con placa de ateroma. En las Figuras 4.9, 4.10 y 4.11 se observa la segmentación total de las geometrías con espesor de 65, 150 y 300 μm respectivamente. Cada parte se ha obtenido siguiendo un proceso de segmentado con unas variables diferentes. En ocasiones la segmentación directa de un tejido es difícil por lo que se realizan diferentes operaciones intermedias. Este es el caso de la Figura 4.12, donde el tejido sano se obtiene tras restar el resultado de una primera segmentación en la que se obtiene el tejido sano unido a la capa fibrótica (Figura 4.12a) y el segundo resultado que es únicamente la capa fibrótica (Figura 4.12b).

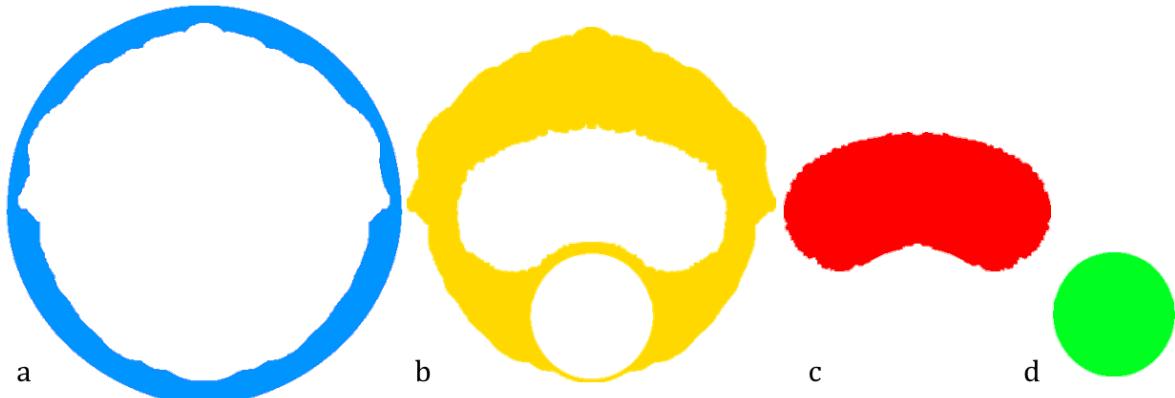


Figura 4.9: Segmentación total de la geometría con espesor de 65 μm , a) Tejido sano, b) Capa fibrótica, c) Placa de ateroma y d) Lumen.

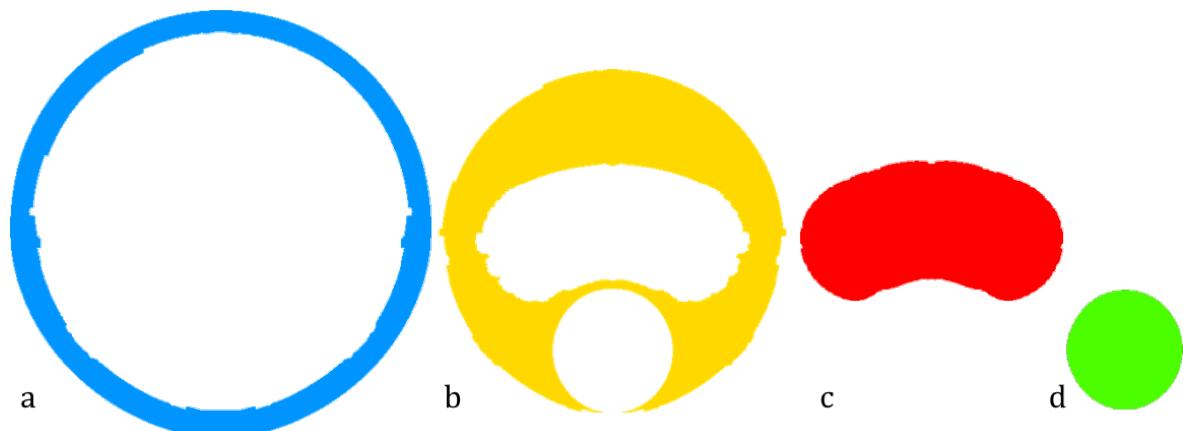


Figura 4.10: Segmentación total de la geometría con espesor de 150 μm , a) Tejido sano, b) Capa fibrótica, c) Placa de ateroma y d) Lumen.

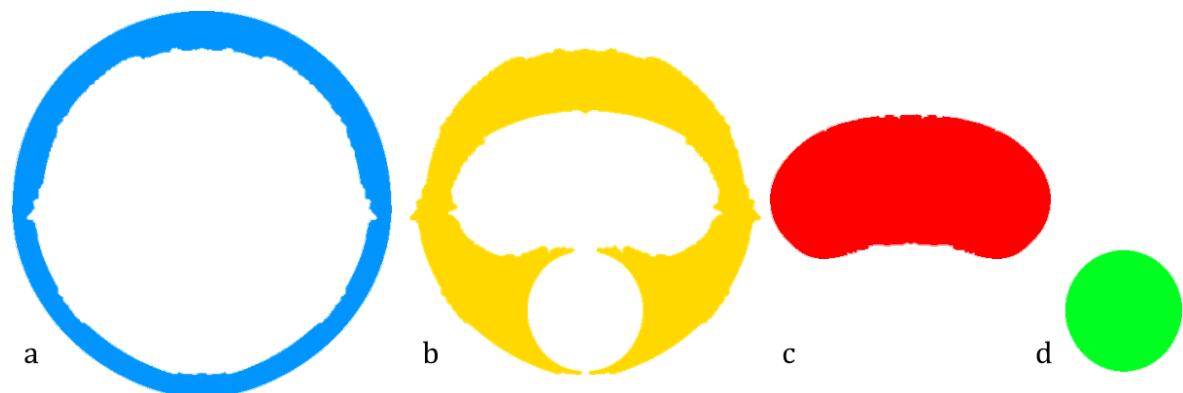


Figura 4.11: Segmentación total de la geometría con espesor de 300 μm , a) Tejido sano, b) Capa fibrótica, c) Placa de ateroma y d) Lumen.

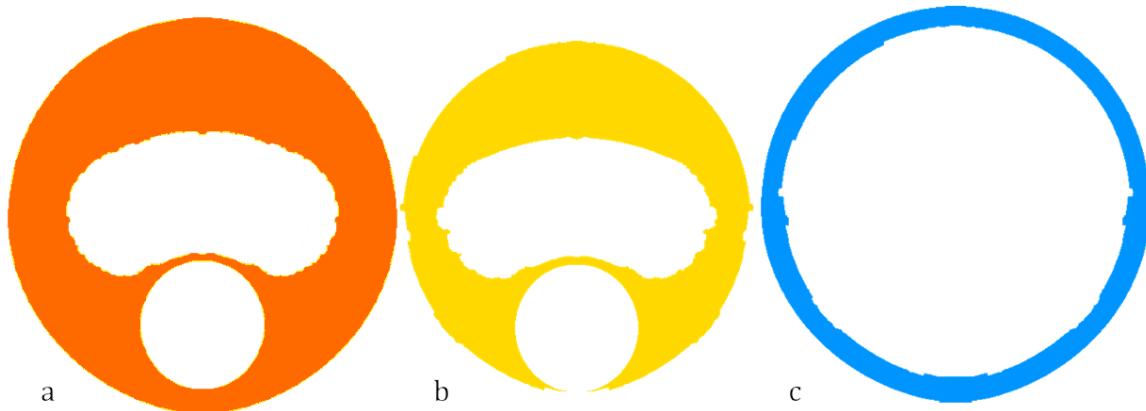


Figura 4.12: a) Segmentación de la capa fibrótica y tejido sano, b) Segmentación de capa fibrótica y c) Segmentación final del tejido sano tras restar las dos anteriores.

4.2. Geometrías No Idealizadas

En el apartado anterior se ha comprobado el proceso de segmentado en la geometría idealizada con diferentes espesores. Sin embargo, es necesario comprobar el método con otras geometrías de lípido.

En las *Figuras 4.13 y 4.14* se muestra el proceso de segmentación aplicado sobre las dos geometrías no idealizadas. Se obtiene el área del núcleo lipídico con un error del 2.04% en la forma 1 y 8.7% en la forma 2, mientras que en la medida del espesor de capa fibrótica tiene un error relativo del 6.1% y 8.3% en las formas 1 y 2 respectivamente.

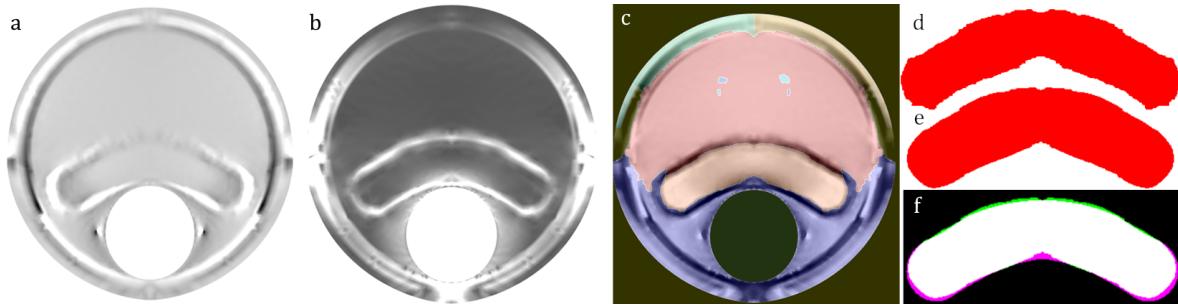


Figura 4.13: Proceso de segmentado en la geometría no idealizada (forma 1): a) dW , b) $|\nabla \varepsilon_y|$, c) Unión de las anteriores y partes segmentadas, d) Núcleo lipídico segmentado, e) Núcleo lipídico segmentado tras el proceso de suavizado y f) Comparación entre núcleo lipídico segmentado (blanco) y real (morado).

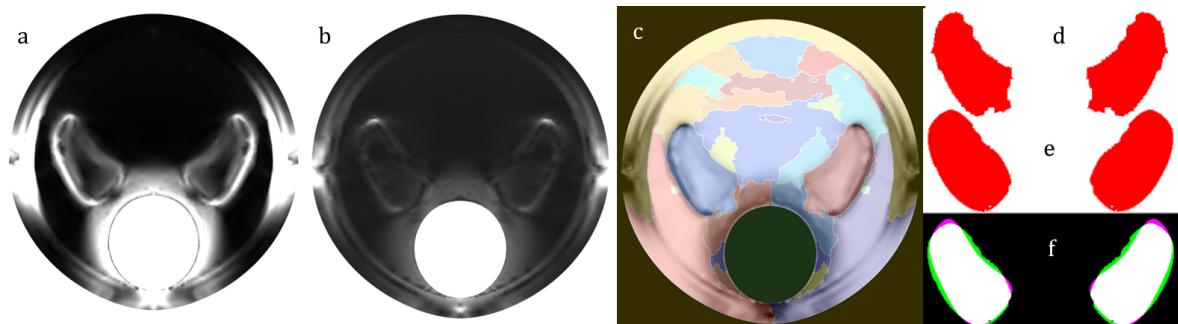


Figura 4.14: Proceso de segmentado en la geometría no idealizada (forma 2): a) $|\nabla \varepsilon_{vonMises}|$, b) $|\nabla \varepsilon_y|$, c) Unión de las anteriores y partes segmentadas, d) Núcleo lipídico segmentado, e) Núcleo lipídico segmentado tras el proceso de suavizado y f) Comparación entre núcleo lipídico segmentado (blanco) y real (morado).

4.3. Geometrías IVUS

Una vez se ha comprobado el correcto funcionamiento del método con diferentes geometrías y con el fin de demostrar que es posible aplicarlo a geometrías reales, se emplea sobre una reconstrucción real. En la *Figura 4.15* se muestra el proceso de segmentado en dicha geometría, donde el error relativo de la medida del espesor de la capa fibrótica alcanza el 7%, mientras que los índices 1 y 2 tienen valores de 86.96 % y 11.56 %. Valores inferiores a los obtenidos en otros métodos desarrollados anteriormente [15].

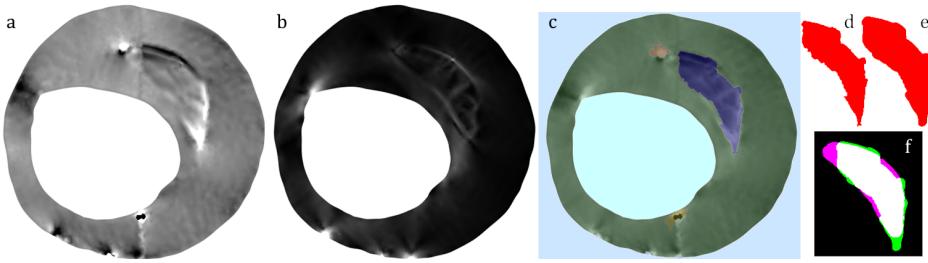


Figura 4.15: Proceso de segmentado en geometría reconstruida de IVUS real: a) dW , b) $|\nabla \epsilon_x|$, c) Unión de las anteriores y partes segmentadas, d) Núcleo lipídico segmentado, e) Núcleo lipídico segmentado y suavizado y f) Comparación entre núcleo lipídico segmentado (blanco) y real (morado).

4.4. Geometría Idealizada. Análisis 3D

Con el fin de conocer si el cálculo de las deformaciones bajo la hipótesis de deformación plana es determinante o no sobre el segmentado, se realiza el mismo proceso de segmentado sobre el modelo 3D, en el que se han considerado todas las deformaciones. Se procede del mismo modo que en el apartado de geometría idealizada, repitiendo las tablas de variables, pero en este caso se ha prestado atención únicamente a aquellas variables que o no podían calcularse en el caso 2D o su valor se ha visto modificado (todas aquellas que dependen de u_z o ϵ_z). En las Tablas 4.16 y 4.17 se muestran los resultados.

	$ \nabla U_z $	$ \nabla \epsilon_z $	$ \nabla \epsilon_{xz} $	$ \nabla \epsilon_{yz} $	$ \nabla \epsilon_{max} $	$ \nabla \epsilon_{med} $	$ \nabla \epsilon_{min} $	$ \nabla \epsilon_{Mises} $	$ \nabla \epsilon_{Tresca} $	$ \nabla FA $	dW
$ \nabla U_z $	23,93%	14,28%	17,11%	15,00%	0,00%	14,29%	43,45%	43,44%	43,44%	14,29%	
$ \nabla \epsilon_z $		6,50%	6,50%	59,32%		12,73%	7,52%	6,86%	6,86%	6,86%	5,76%
$ \nabla \epsilon_{xz} $						25,29%	8,71%	6,86%	7,25%	7,25%	
$ \nabla \epsilon_{yz} $							35,78%	34,12%	6,86%	3,57%	3,57%
$ \nabla \epsilon_{max} $							34,12%	21,58%	6,86%	22,62%	22,62%
$ \nabla \epsilon_{med} $								87,41%	6,86%	6,86%	6,86%
$ \nabla \epsilon_{min} $									4,83%	20,21%	20,21%
$ \nabla \epsilon_{Mises} $										6,50%	6,50%
$ \nabla \epsilon_{Tresca} $										6,50%	6,50%
$ \nabla FA $										6,50%	33,57%
dW											6,50%
Leyenda											
■ Imposible de Segmentar											
■ Segmentación Visual											
■ Segmentación Visual											

Figura 4.16: Tabla de segmentación de combinaciones tridimensional en la geometría idealizada de 150 μm de espesor en el análisis 3D. Se presenta el error de medida del espesor de la capa fibrótica.

	$ \nabla U_z $	$ \nabla \epsilon_z $	$ \nabla \epsilon_{xz} $	$ \nabla \epsilon_{yz} $	$ \nabla \epsilon_{max} $	$ \nabla \epsilon_{med} $	$ \nabla \epsilon_{min} $	$ \nabla \epsilon_{Mises} $	$ \nabla \epsilon_{Tresca} $	$ \nabla FA $	dW
$ \nabla U_z $	95,14/0,71	84,82/1,45	95,32/3,51	98,47/3	96,27/0,72	99,58/7,1	98,8/2,46	98,8/2,46	98,8/2,46	85,31/0,06	
$ \nabla \epsilon_z $		90,72/1,5	92,38/1,81	96,73/4,31	97,76/4,67	92,68/1,83	93,82/2,25	96,3/3,23	96,3/3,23	96,3/3,23	92,95/2
$ \nabla \epsilon_{xz} $					98,44/4,02	89,2/0,26	96,88/4,64	78,44/0,11	78,44/0,11	78,44/0,11	
$ \nabla \epsilon_{yz} $						98,36/3,58	98,25/5,56	94,15/0,05	76,19/0,18	76,19/0,18	
$ \nabla \epsilon_{max} $						98,46/3,17	97,82/4,6	98,02/5,05	99,06/6,34	99,06/6,34	
$ \nabla \epsilon_{med} $								91,73/0,12	92,76/0,08	92,76/0,08	91,53/1,6
$ \nabla \epsilon_{min} $									93,17/1,66	91,56/0,02	91,56/0,02
$ \nabla \epsilon_{Mises} $										89,1/0,11	89,1/0,11
$ \nabla \epsilon_{Tresca} $										89,1/0,11	89,1/0,11
$ \nabla FA $										89,1/0,11	73,84/0
dW											89,1/0,11
Leyenda											
■ Imposible de Segmentar											
■ Segmentación Visual											
■ Segmentación Visual											

Figura 4.17: Tabla de segmentación de combinaciones tridimensional en la geometría idealizada de 150 μm de espesor en el análisis 3D. Se presentan los índices1/indice2.

Se observa en la Figura 4.16, que al considerar todos los desplazamientos, la variable $|\nabla \epsilon_{min}|$ permite segmentar sola y en combinación con otras variables con un error relativo bajo, al contrario que el en caso 2D. Al incluir nuevas variables, dependientes de u_z se abre el abanico de combinaciones para segmentar como $|\nabla \epsilon_{yz}|$ en combinación de $|\nabla \epsilon_{Tresca}|$. Por contra $|\nabla \epsilon_{max}|$ y $|\nabla \epsilon_{min}|$ han empeorado. En términos de cuantificación de área, en la Figura 4.17 se aprecia una disminución de la precisión generalizada del método frente a los índices del modelo 2D (Figura 4.6).

5. Conclusiones

Una vez presentados todos los resultados en el apartado anterior, se exponen a continuación las conclusiones del estudio:

- Se ha conseguido desarrollar una metodología automática que permite obtener, a partir de simulaciones de elementos finitos, las deformaciones de cualquier geometría de forma análoga a cómo se obtendrían a través de un IVUS.
- El proceso de segmentado desarrollado es capaz de obtener el núcleo lipídico y medir de forma semi-automática su área y el espesor de la capa fibrótica en geometrías idealizadas con diferentes espesores, geometrías no idealizadas y en una geometría real reconstruida a partir de un IVUS. Por lo tanto, se demuestra cierta versatilidad del código desarrollado, al no depender de la geometría analizada.
- Tras analizar las 131 combinaciones de variables en las 3 geometrías idealizadas se concluye que el porcentaje de error depende de las variables elegidas. Variables como la $|\nabla \varepsilon_y|$ y $|\nabla \varepsilon_\theta|$ proporcionan resultados de gran precisión, tanto solos como en combinación con otras variables. Sin embargo, en el IVUS la resolución en la dirección circunferencial (θ) es más baja que en el caso radial, pero también se han encontrado otras combinaciones plausibles cuyo error relativo es bajo. Por otro lado la deformación principal mínima $|\nabla \varepsilon_{min}|$ no aporta ninguna información, ya que los cálculos se han realizado bajo la hipótesis de deformación plana y hace que tome valores nulos. Este problema se resuelve al considerar el análisis 3D, donde la variable permite realizar segmentados.
- El porcentaje de error también depende del espesor real de la capa fibrótica a analizar. Se ha comprobado que cuanto menor es el espesor, más difícil es realizar el segmentado ya que hay menos variables que proporcionen resultados significativos y los errores aumentan. Esto se debe a que al tener menos superficie entre el lumen, donde se aplica la presión sanguínea, y el núcleo lipídico los contornos entre tejidos quedan difuminados al existir una gran deformación. También es necesario apuntar que los errores en el caso del espesor de $65 \mu m$, además, son altos por la propia formulación del error relativo (*Ecuación 3.38*), donde un denominador pequeño hace que el error sea muy sensible a pequeños cambios en la medida.
- Aunque el estudio se ha centrado en la segmentación del núcleo lipídico también se ha conseguido una segmentación total de los diferentes tejidos que componen una arteria coronaria con placa de ateroma. La correcta segmentación de las diferentes partes está ligada al espesor de la capa fibrótica, ya que para el espesor más bajo analizado las geometrías reconstruidas presentan mayor error. Mediante la combinación de diferentes variables ha sido posible realizar segmentaciones totales en todas las geometrías idealizadas. Sin embargo, en raras ocasiones es posible visualizar el tejido sano en un IVUS.
- Al implementar el método sobre una geometría real, se observa que se puede obtener el núcleo al igual que en las geometrías idealizadas. Sin embargo, el error relativo aumenta y los índices 1 y 2 empeoran respecto a las geometrías anteriores. Parte de este aumento del error se debe a que en este caso el análisis de elementos finitos se ha realizado íntegramente en 2D y en lugar de utilizar materiales completamente incompresibles se utilizaron quasi-incompresibles para evitar problemas de cálculo. Todo ello repercute en la manera de deformarse y por lo tanto en el resultado final, donde es más difícil apreciar los contornos al analizar los módulos de los gradientes de las diferentes variables. Además, en la geometría idealizada los contornos son más suaves y permiten una mejor segmentación.
- Al comparar los análisis 2D realizados bajo la hipótesis de deformación plana con los realizados en 3D, se observa que aparecen más variables que permiten una correcta segmentación. Además, variables como la deformación principal mínima (ε_{min}) o las deformaciones equivalentes (von Mises, Tresca...) modifican su valor al poder considerar la deformación en el eje Z. Sin embargo, estas variables no presentan una ventaja significativa frente a las obtenidas en el caso 2D. Por ello, el análisis 3D, aunque es más completo y se aproxima más al comportamiento real del

tejido, necesita una gran cantidad de horas de computación sin obtener ventajas significativas frente al cálculo de deformación 2D con hipótesis de deformación plana.

- Como última conclusión, se desea señalar de la importancia que tendría una aplicación de este método en la práctica clínica, donde se trabajarían con las deformaciones obtenidas a partir del IVUS. Con ellas se podría segmentar el núcleo lipídico y tomar medidas de interés que repercutirían en el diagnóstico precoz de la placa de ateroma en arterias coronarias.

5.1. Limitaciones

- El proceso no es completamente automático, ya que existen 3 parámetros que han de modificarse manualmente: límites de representación, tamaño del disco de segmentado y parámetros de suavizado. Sin embargo, un análisis más profundo del tratamiento de la imagen podría eliminar esta parte manual.
- El análisis de elementos finitos se ha realizado sin considerar el movimiento del corazón y tensiones residuales. Para obtener unos resultados más realistas sería necesario incluirlos, sin embargo su implementación se descartó debido al alto grado de complejidad.
- El cálculo de las deformaciones se ha realizado en los nodos en lugar de utilizar los puntos de la cuadratura de Gauss. Aunque a nivel de cálculo de elementos finitos deberían considerarse los puntos de Gauss esto aumenta la complejidad del cálculo y no aporta una mejora en las deformaciones obtenidas, tal y como se ha podido comprobar en el *Anexo III*.

5.2. Líneas Futuras

Este estudio sienta las bases de un proyecto más amplio, en el que, además de solventar las limitaciones anteriormente definidas, se utilizará con datos de IVUS reales. Tras este primer paso de segmentación, se buscará obtener una elastografía que permita conocer las propiedades mecánicas de cada tejido segmentado mediante técnicas de optimización. Se aplicará el método con una mayor cantidad de geometrías reales, donde se introducirán inclusiones y homogeneidades en las geometrías para comprobar su influencia en los resultados. Además, se analizará el efecto del ruido sobre los resultados finales.

6. Bibliografía

- [1] World Health Organization. *Cardiovascular diseases (CVDs)*, 2017. URL [https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)). [Consultado el: 13 de Noviembre de 2020].
- [2] Sociedad Española de Aterosclerosis (SEA). *Las enfermedades cardiovasculares y sus factores de riesgo en España: hechos y cifras*. Visto Bueno Equipo Creativo, S.L., 2007. España.
- [3] Myriam Cilla. *Mechanical effects on the atheroma plaque appearance, growth and vulnerability*. PhD thesis, Universidad de Zaragoza, 2013.
- [4] J. Ohayon, J. G. Finet, S. Le Floc'h, A.M. Gharib G. Cloutier, J. Heroux, and R.I. Pettigrew. Biomechanics of atherosclerotic coronary plaque: Site, stability and in vivo elasticity modeling. *Annals of Biomedical Engineering*, 42:269–279, 2014.
- [5] A. Velasco, S. Mejía, JJ. Gavira, PM. Azcárate, and I. Coma-Canella. Placa vulnerable. *Revista de Medicina de la Universidad de Navarra*, 48(1):24–31, 2004.
- [6] S. Glaglov, E. Weisenberg, CK. Zarins, R. Stankunavicius, and G.J. Kolettis. Compensatory enlargement of human atherosclerotic coronary arteries. *The New England Journal of Medicine*, 316(22):1371–1375, 1987.
- [7] A. Gómez, A. Tacheau, G. Finet, M. Lagache, JL. Martiel, S. Le Floc'h, SK. Yazdani, A. Elias-Zuñiga, RI. Pettigrew, G. Cloutier, and J. Ohayon. Intraluminal ultrasonic palpation imaging technique revisited for anisotropic characterization of healthy and atherosclerotic coronary arteries: A feasibility study. *Ultrasound in Medicine and Biology*, 45(1):35–49, 2019.
- [8] JA. Schaar, JE. Muller, E. Falk, R. Virmani, V. Fuster, PW. Serruys, A. Colombo, C. Stefanadis, S. Ward Casscekk, PR. Moreno, A. Maseri, and AF. van der Steen. Terminology for high-risk and vulnerable coronary artery plaques. *European Heart Journal*, 25(12):1077–1082, 2004.
- [9] Y. Majdouline, J. Ohayon, Z. Keshavarz-Motamed, MHR Cardinal, D. García, L. Allard, S. Lérouge, F. Arsenault, G. Soulez, and G. Cloutier. Endovascular shear strain elastography for the detection and characterization of the severity of atherosclerotic plaques: In vitro validation and in vivo evaluation. *Ultrasound in Medicine and Biology*, 40(5):890–903, 2014.
- [10] J. Porée, D. García, B. Chayer, J. Ohayon, and G. Cloutier. Noninvasive vascular elastography with plane strain incompressibility assumption using ultrafast coherence compound plane wave imaging. *Institute of Electrical and Electronic Engineers: Transactions of medical imaging*, 34(12):2618–2631, 2015.
- [11] R. Nayak, S. Huntzicker, J. Ohayon, N. Carson, V. Dogra, G. Schifitto, and MM. Doyley. Principal strain vascular elastography: simulation and preliminary clinical evaluation. *Ultrasound in Medicine and Biology*, 43(3):682–699, 2017.
- [12] Armida Gómez. *A Novel Anisotropic Elasticity-Palpography IVUS Imaging Tool and Characterization of the Viscoelasto-Plastic Properties of Angioplasty Balloon : Towards a Better Assessment and Treatment of Coronary Artery Diseases*. PhD thesis, University of Grenoble, 2019.
- [13] RP. Vito, MC. Whang, DP. Giddens, CK. Zarins, and S. Glagov. Stress analysis of the diseased arterial cross-section. *American Society of Mechanical Engineers of Advance Bioengineering*, 19, 1990.
- [14] T. Idzenga, S. Holewijn, HHG. Hansen, and CL. de Korte. Estimating cyclic shear strain in the common carotid artery using radiofrequency ultrasound. *Ultrasound in Medicine and Biology*, 38(12):2229–37, 2012.

- [15] S. Le Floc'h, J. Ohayon, P. Tracqui, G. Finet, AM. Gharib, RL. Maurice, G. Cloutier, and RI. Pettigrew. Vulnerable atherosclerotic plaque elasticity reconstruction based on a segmentation-driven optimization procedure using strain measurements: Theoretical framework. *Institute of Electrical and Electronic Engineers: Transactions of medical imaging*, 28(7):1126–1137, 2009.
- [16] J. Porée, B. Chayer, G. Soulez, J. Ohayon, and G. Cloutier. Nominvasive vascular modulography method for imaging the local elasticity of atherosclerotic plaques: Simulation and in vitro vessel phantom study. *Institute of Electrical and Electronic Engineers: Transactions on ultrasonics, ferroelectrics and frequency control*, 64(12):1805–1817, 2017.
- [17] A. Tacheau, S. Le Floc'h, G. Finet, MM. Doyley, RI. Pettigrew, G. Cloutier, and J. Ohayon. The imaging modulography technique revisited for high-definition intravascular ultrasound : Theoretical framework. *Ultrasound in Medicine and Biology*, 42(3):727–741, 2016.
- [18] S. Le Floc'h, G. Cloutier, Y. Saito, G. Finet, SK. Yazdani, F. Deleaval, G. Rioufol, RI. Pettigrew, and J. Ohayon. A four-criterion selection procedure for atherosclerotic plaque elasticity reconstruction based on in vivo coronary intravascular ultrasound radial strain sequences. *Ultrasound in Medicine and Biology*, 38(12):2084–2097, 2012.
- [19] RL. Maurice, J. Ohayon, G. Finet, and G. Cloutier. Adapting the lagrangian speckle model estimator for endovascular elastography: Theory and validation with simulated radio-frequency data. *Ultrasound in Medicine and Biology*, 116(2):1276–86, 2004.
- [20] I. Zervantonakis, S. Fung-Kee-Fung, W Lee, and E. Konofagou. A novel, view-independent method for strain mapping in myocardial elastography eliminating angle and centroid dependence. *Physics in Medicine and Biology*, 52(14):4063–80, 2007.
- [21] RL. Maurice, J. Ohayon, Y. Fretigny, M. Bertrand, G. Soulez, and G. Cloutier. Non-invasive vascular elastography: A prospective study. *Institute of Electrical and Electronic Engineers: Ultrasonics Symposium*, 2:1839–1842, 2002.
- [22] C. Sumi and K. Nakayama. A robust numerical solution to reconstruct a globally relative shear modulus distribution from strain measurements. *Institute of Electrical and Electronic Engineers*, 17(3):419–28, 1998.
- [23] A. Versluis, AK. Bank, and WH. Douglas. Fatigue and plaque rupture in myocardial infarction. *Journal of the Royal Society*, 39(2):339–47, 2006.
- [24] GA. Holzapfel, G. Sommer, CT. Gasser, and P. Regitnig. Determination of layer-specific mechanical properties of human coronary arteries with nonatherosclerotic intimal thickening and related constitutive modeling. *American Journal of Physiology-Heart and Circulatory Physiology*, 289(5):2048–58, 2005.
- [25] MT. Walsh, EM. Cunnane, JJ. Mulvihill, AC. Akyildiz, FJ. Gijsen, and GA. Holzapfel. Uniaxial tensile testing approaches for characterisation of atherosclerotic plaques. *Journal of Biomechanics*, 47(4):793–804, 2014.
- [26] H M. Loree, A J. Grodzinsky, S Y. Park, L J. Gibson, and R T. Lee. Static circumferential tangential modulus of human atherosclerotic tissue. *Journal of Biomechanics*, 27(2):195–204, 1994.
- [27] Chen-Ket Chai, L. Speelman, Cees W.J. Oomens, and Frank P.T. Baaijens. Compressive mechanical properties of atherosclerotic plaques—indentation test to characterise the local anisotropic behaviour. *Journal of Biomechanics*, 47(4):784 – 792, 2014.
- [28] GV. Born and PD. Richardson. Mechanical properties of human atherosclerotic lesions. pathobiology of the human atherosclerotic plaque. *Springer New York*, pages 413–423, 1990.
- [29] B. Tummers. *DataThief III*, 2006. URL <https://datathief.org/>.
- [30] P. Skacel. *Hyperfit: Software for fitting of hyperelastic constitutive models*, 2018. URL <http://www.hyperfit.wz.cz/home.php>.

- [31] AT. Latorre. *Análisis mecánico por elementos finitos de la influencia de la longitud de la placa y el lípido en el diagnóstico de placas de ateroma en crecimiento positivo*. Master's thesis, Universidad de Zaragoza, 2018.
- [32] G. Herrero. *Análisis mecánico de placas de ateroma en fase de crecimiento positivo: Estudio de la influencia de la longitud de la placa*. Master's thesis, Universidad de Zaragoza, 2015.
- [33] G. Finet, J. Ohayon, and G. Rioufol. Biomechanical interaction between cap thickness, lipid core composition and blood pressure in vulnerable coronary plaque: impact on stability or instability. *Coronary Artery Disease*, 15(1):13–20, 2004.
- [34] S. Le Floc'h. *Modulographie vasculaire : Application à l'identification in-vivo du module de Young local des plaques d'athérosclérose*. PhD thesis, Université Joseph-Fourier - Grenoble, 2009.
- [35] T.C. Gasser, RW.Ogden, and GA. Holzapfel. Hyperelastic modelling of arterial layers with distributed collagen fibre orientations. *Journal of the Royal Society*, 3:15–35, 2006.
- [36] J. Ohayon, G. Finet, AM. Gharib, DA. Herzka, P. Tracqui, J. Heroux, G. Rioufol, MS. Kotys, A. Elagha, and RI. Pettigrew. Necrotic core thickness and positive arterial remodeling index: emergent biomechanical factors for evaluating the risk of plaque rupture. *American Journal of Physiology-Heart and Circulatory Physiology*, 295(2):717–27, 2008.
- [37] Boston Scientific. *Boston Scientific: Advancing Science for life*, 2020. URL <https://www.bostonscientific.com/en-EU/products/imaging-systems/opticross-coronary-imaging-catheter.html>. [Consultado el: 16 de Octubre de 2020].
- [38] S. Soleimanifard, KZ. Abd-Elmoniem, HK. Agarwal, MS. Tomás, T. Sasuno, E. Vonken, A. Youssef, MR. Abraham, TP. Abraham, and JL. Prince. Identification of myocardial infarction using three-dimensional strain tensor fractional anisotropy. *Institute of Electrical and Electronic Engineers. International Symposium on Biomedical Imaging: From Nano to Macro*, pages 468–471, 2010.
- [39] P. Rachakonda. *Mathworks-SphereFit*, 2020. URL <https://www.mathworks.com/matlabcentral/fileexchange/45910-sphere-fit>. [Consultado el: 27 de Agosto de 2020].
- [40] C. Díaz. *Desarrollo de un modelo constitutivo multicapa de pared arterial y su validez mediante ensayos experimentales de arteria aorta descendente y abdominal*. Master's thesis, Universidad de Zaragoza, 2019.
- [41] D. Malyuta. *Mathworks-SphereFit*, 2016. URL <http://www.regentsprep.org/regents/math/geometry/gcg6/RCir.htm>. [Consultado el: 15 de Agosto de 2020].

ANEXOS

Anexo I: Materiales

En este *Anexo* se van a definir los materiales a utilizar en el modelado de la coronaria con placa de ateroma. El modelo a analizar consta de diferentes partes: adventicia, media, íntima, capa fibrótica y núcleo lipídico. Cada parte se modelará con unos parámetros de comportamiento diferentes extraídos de gráficas experimentales recogidas en la literatura [23, 24, 25, 26, 27, 28]. Como punto de partida se toma el modelo y materiales utilizados en el proyecto anterior [31] y se denominará Modelo 1, a partir de él se desarrollarán nuevos modelos en los que habrá diferentes modificaciones:

- **Íntima:** En el Modelo 1 no se consideraba íntima, si no que se modelaba como placa o capa fibrótica, y por lo tanto en zonas alejadas a la lesión el resultado no era realista, ya que aparecía con un alto nivel tensional y por lo tanto vulnerable [31].
- **Orientación de las fibras:** En el Modelo 1 la orientación de las fibras de los materiales considerados como anisótropos (adventicia y media) se introdujeron desde las propias ventanas del software Abaqus CAE. Sin embargo, éste solo guarda la última orientación asignada. Por lo tanto, se deben introducir manualmente dentro del archivo *inp* generado por Abaqus con el siguiente código [40].

```
*parameter
Pi = 3.141592654
beta = 52.72
theta5 = beta
c5 = cos(theta5*Pi/180)
s5 = sin(theta5*Pi/180)
theta6 = -theta5
c6 = cos(theta6*Pi/180)
s6 = sin(theta6*Pi/180)
*Orientation, name=Ori-3, system=RECTANGULAR, local directions=2
Ori-3-DiscOrient
3, 0.
<c5>, <s5>, 0.0
<c6>, <s6>, 0.0
```

- Se van a comparar diferentes comportamientos de placas de ateroma e íntima. Muchas de las curvas analizadas para placa pertenecen a carótida [25] o aorta [26, 25], ya que existe poca literatura que caracterice placas en coronarias [28, 24].

Modelos Constitutivos

A continuación, se presentan los diferentes modelos estudiados, donde todos ellos se han modelado con la ecuación de comportamiento del modelo constitutivo de Gasser [35]. Las curvas se han obtenido a partir de la bibliografía utilizando el software libre DataThief III [29] que permite obtener los puntos seleccionados de una gráfica consiguiendo los puntos de las curvas, lo que permite importarlo al software Hyperfit [30] el cual permite obtener, mediante optimización, los parámetros de ajuste de los ensayos experimentales.

La comparativa entre modelos se ha realizado con una geometría diferente al resto del estudio, con una placa de 15 mm de longitud y un núcleo lipídico de 11.25 mm. Se ha realizado con este modelo para poder tener una comparativa con una geometría más grande y que permitiera visualizar mejor los cambios en el comportamiento de la placa. Se representa la tensión principal máxima [kPa] con el límite superior de 247 kPa, ya que a partir de ese valor se considera una zona muy vulnerable y con riesgo de fractura [3].

Modelo 1

Es el modelo base, utilizado en proyectos anteriores [31, 32], donde no se modela la íntima sana y el tejido sano únicamente tiene una dirección de fibras. La media y adventicia se han obtenido de los datos de *Holzapfel* [24], mientras que la placa y el núcleo de *Versluis* [23] utilizados ambos en otros trabajos [3, 31, 32]. Al considerar el tejido sano como anisótropo tienen dos curvas, longitudinal y axial, mientras que el tejido enfermo tiene solo una al ser considerado isótropo. En la *Figura 6.1* se muestran las curvas de comportamiento analizadas y en la *Figura 6.2* se observan los resultados del cálculo del este primer modelo. Se puede observar que toda la íntima es vulnerable ya que toda ella se ha considerado como tejido enfermo (capa fibrótica). A continuación se muestra la Tabla 6.1 de los parámetros de ajuste con las curvas analizadas:

Tejido	μ [kPa]	D [-]	k_1 [kPa]	k_2 [kPa]	κ [-]	Ángulo [º]
Adventicia	8.44	0	547.67	568.01	0.26	61.8
Media	1.4	0	206.16	58.55	0.29	28.35
Íntima	-	-	-	-	-	-
Capa Fibrótica	9.58	0	17654.91	0.51	0.33	-
Núcleo Lipídico	0.05	0	956.76	70	0.33	-

Tabla 6.1: Parámetros para definir el modelo de comportamiento de Gasser.

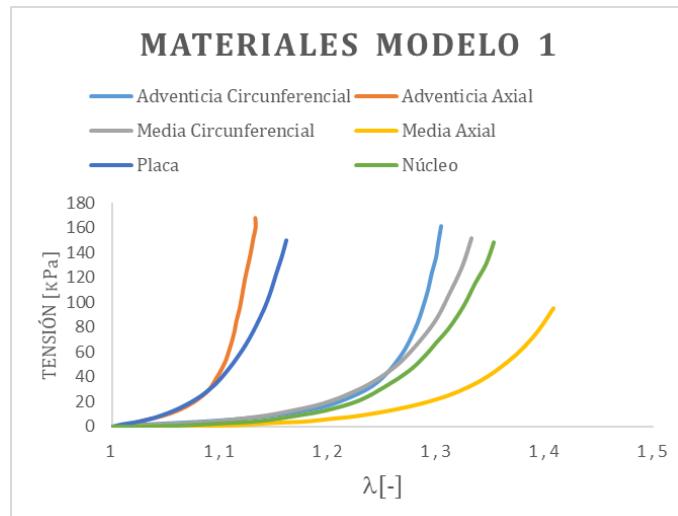


Figura 6.1: Curvas de comportamiento de los materiales del modelo 1.

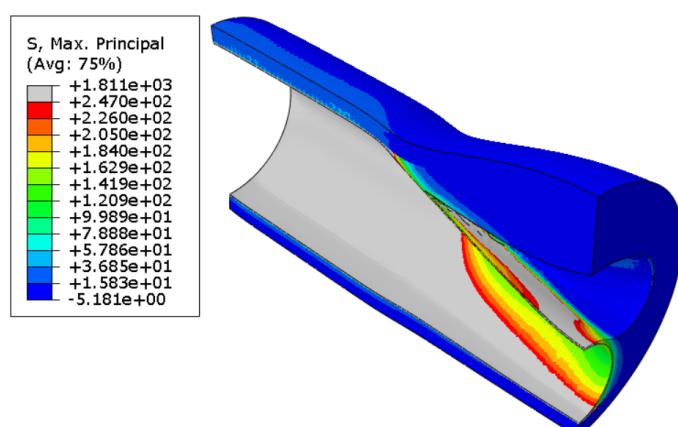


Figura 6.2: Resultado del Modelo 1. Tensión Principal Máxima [kPa].

Modelo 2

En este modelo se añaden las dos familias de fibras en la adventicia y en la media correctamente [40]. Además se incluye la íntima con propiedades de media, pero con las fibras orientadas un ángulo diferente. En este caso, tal y como se muestra en la *Figura 6.3*, la íntima tiene un comportamiento muy blando y es muy poco vulnerable, un resultado más similar al real. Los parámetros de ajuste se observan en la *Tabla 6.2*.

Tejido	μ [kPa]	D [-]	k_1 [kPa]	k_2 [kPa]	κ [-]	Ángulo [º]
Adventicia	8.44	0	547.67	568.01	0.26	± 61.8
Media	1.4	0	206.16	58.55	0.29	± 28.35
Íntima	1.4	0	206.16	58.55	0.29	± 52.13
Capa Fibrótica	9.58	0	17654.91	0.51	0.33	-
Núcleo Lipídico	0.05	0	956.76	70	0.33	-

Tabla 6.2: Parámetros para definir el modelo 2 de comportamiento de Gasser.

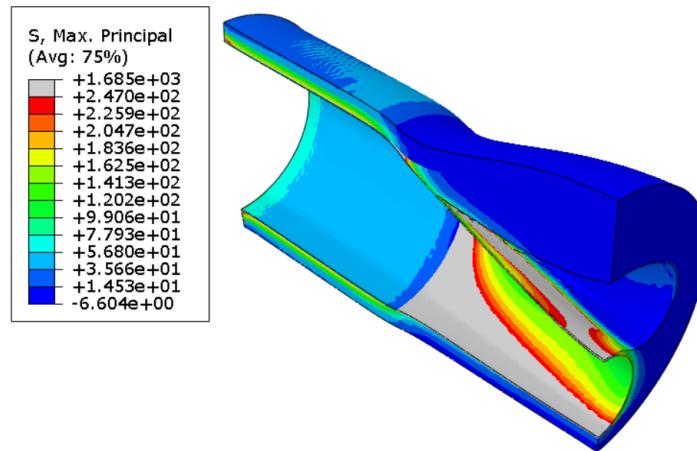


Figura 6.3: Resultado del Modelo 2. Tensión Principal Máxima [kPa].

Modelo 3

A continuación la íntima toma las propiedades de *Holzapfel* paciente VI [24] pero con el parámetro $\kappa = 1/3$ con el fin de tratar el tejido como material isótropo. Se observa en la *Figura 6.4*, que el material es mucho más rígido y es altamente vulnerable.

Tejido	μ [kPa]	D [-]	k_1 [kPa]	k_2 [kPa]	κ [-]	Ángulo [º]
Adventicia	8.44	0	547.67	568.01	0.26	± 61.8
Media	1.4	0	206.16	58.55	0.29	± 28.35
Íntima	9.68	0	70983.1	0.006	0.33	-
Capa Fibrótica	9.58	0	17654.91	0.51	0.33	-
Núcleo Lipídico	0.05	0	956.76	70	0.33	-

Tabla 6.3: Parámetros para definir el modelo 3 de comportamiento de Gasser.

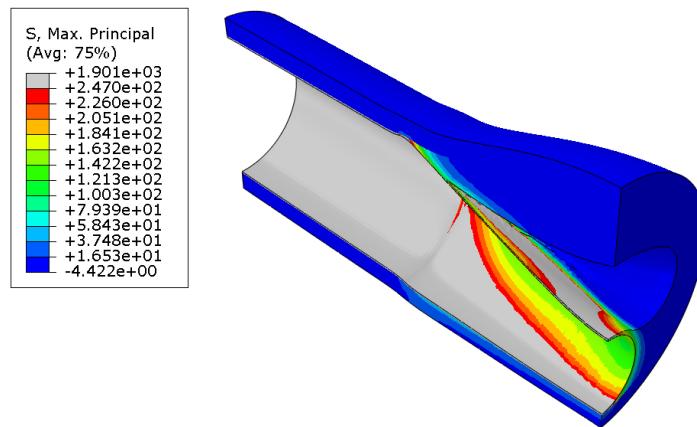


Figura 6.4: Resultado del Modelo 3. Tensión Principal Máxima [kPa].

Modelo 4

En este modelo se intenta modificar la íntima para que no adquiera un comportamiento tan rígido y se cambia el material de la placa de ateroma para ver la influencia en la vulnerabilidad de la misma. Se toma la íntima con propiedades circunferenciales del paciente VI de *Holzapfel* [24] y axiales del paciente V de *Holzapfel* [24]. Además, se utiliza la placa obtenida analizando la media de las placas celulares obtenidas por *Loree* [26]. Los resultados representados en la *Figura 6.5* indican que la íntima sigue siendo demasiado rígida, mientras que la placa es muy blanda y no es vulnerable en la zona de máxima estenosis.

Tejido	μ [kPa]	D [-]	k_1 [kPa]	k_2 [kPa]	κ [-]	Ángulo [º]
Adventicia	8.44	0	547.67	568.01	0.26	± 61.8
Media	1.4	0	206.16	58.55	0.29	± 28.35
Íntima	7.096	0	183.44	133.38	0.12	± 51.76
Capa Fibrótica	0.001	0	3906.74	32.13	0.33	-
Núcleo Lipídico	0.05	0	956.76	70	0.33	-

Tabla 6.4: Parámetros para definir el modelo 4 de comportamiento de Gasser.

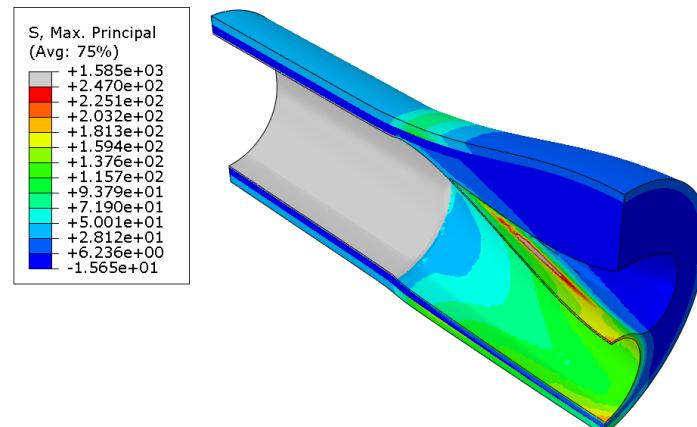


Figura 6.5: Resultado del Modelo 4. Tensión Principal Máxima [kPa].

Modelo 5

En este modelo se intenta solventar los problemas con la placa y la íntima, para ello se toma la misma curva de íntima [24] pero pretensada en su curva longitudinal un 10%, con lo que se consigue un

comportamiento más natural del esperado por este tejido ya que los pacientes utilizados en el estudio de *Holzapfel* [24] son pacientes de avanzada edad y han perdido una gran cantidad de elastina. En cuanto a la placa se toma una placa más vulnerable dentro de las estudiadas por *Loree* [26].

Tejido	μ [kPa]	D [-]	k_1 [kPa]	k_2 [kPa]	κ [-]	Ángulo [º]
Adventicia	8.44	0	547.67	568.01	0.26	± 61.8
Media	1.4	0	206.16	58.55	0.29	± 28.35
Íntima	6.81	0	109.1	101.04	0.21	± 52.72
Capa Fibrótica	7.28	0	9303.53	102.71	0.33	-
Núcleo Lipídico	0.05	0	956.76	70	0.33	-

Tabla 6.5: Parámetros para definir el modelo 5 de comportamiento de Gasser.

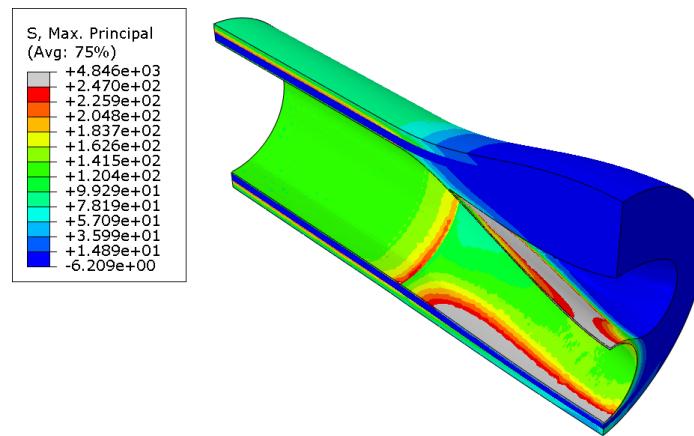


Figura 6.6: Resultado del Modelo 5. Tensión Principal Máxima [kPa].

Modelo 6

Paralelamente al modelo anterior se realiza otro, con la misma íntima que el modelo 5 pero con una placa celular de *Loree* diferente y menos rígida [26]. La diferencia con la anterior es que la placa es mucho menos vulnerable y tiene un comportamiento más blando.

Tejido	μ [kPa]	D [-]	k_1 [kPa]	k_2 [kPa]	κ [-]	Ángulo [º]
Adventicia	8.44	0	547.67	568.01	0.26	± 61.8
Media	1.4	0	206.16	58.55	0.29	± 28.35
Íntima	6.81	0	109.1	101.04	0.21	± 52.72
Capa Fibrótica	14.08	0	2952.46	25.72	0.33	-
Núcleo Lipídico	0.05	0	956.76	70	0.33	-

Tabla 6.6: Parámetros para definir el modelo 6 de comportamiento de Gasser.

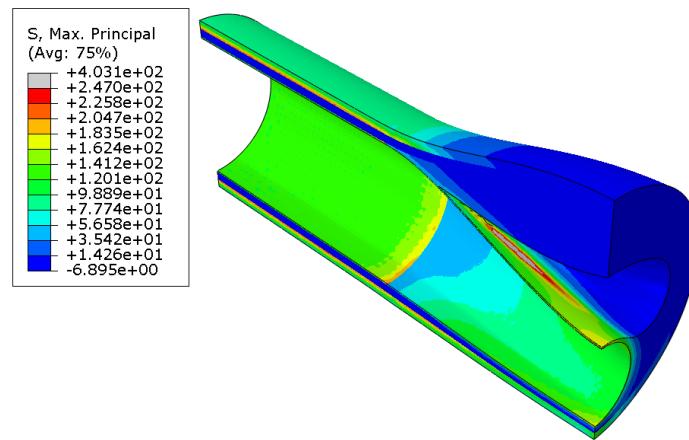


Figura 6.7: Resultado del Modelo 6. Tensión Principal Máxima [kPa].

Comparación de Resultados

En la *Figura 6.8* se muestran las curvas de comportamiento de las íntimas analizadas en los diferentes modelos, donde todos los valores se han obtenido del estudio de *Holzapfel* [24]. En la *Figura 6.9* se realiza la misma comparativa pero con las curvas de placas analizadas en los modelos.

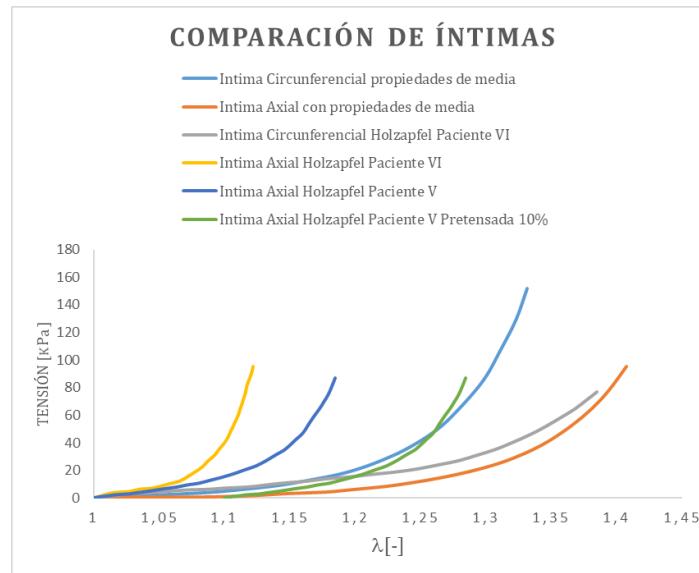


Figura 6.8: Curvas de comportamiento de las íntimas analizadas en los modelos. σ [kPa].

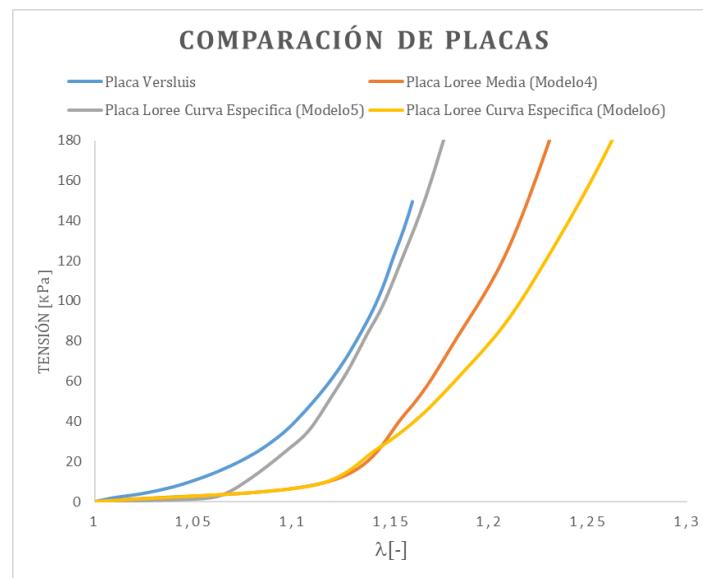


Figura 6.9: Curvas de comportamiento de las capas fibróticas analizadas en los modelos. σ/kPa .

Anexo II: Metodología desarrollada en Matlab

En este Anexo se expone y explica el código más relevante utilizado para el desarrollo del proyecto, que engloba diferentes pasos como la lectura de los datos importados desde Abaqus a Matlab, la creación de una malla a partir de los datos posicionales de los nodos, cálculo de la deformación a partir de los desplazamientos, cálculo de otras variables y representación de todas las variables obtenidas y los módulos de sus gradientes. La parte de segmentación se desarrolla íntegramente en el *Anexo V* reservando éste a la explicación del método para obtener la representación de las variables. La metodología utilizada para tratar los modelos 2D y 3D son muy similares por lo que se explica el programa 2D y se comentan las diferencias más significativas que tiene el 3D respecto a este. A continuación, se explican las funciones y diferentes partes del código programado:

Lectura de Datos

Se desarrolla una función *DatosAbaqus(Abrir, Salida)* que tiene como objetivo leer las variables de salida del cálculo de elementos finitos realizado en Abaqus CAE. Las variables de interés del modelo de Abaqus son el número de cada nodo, sus coordenadas y sus desplazamientos (aunque no son totalmente necesarios, ya que el desplazamiento se puede calcular como la diferencia entre las coordenadas entre dos instantes de tiempo). Adicionalmente se pueden introducir también resultados de deformaciones logarítmicas para comprobar resultados a modo de validación del programa. El código permite leer el archivo y clasificar cada región del modelo con un número (1, 2, 3, 4 y 5, que corresponden con núcleo, placa, media, adventicia e íntima respectivamente, pero depende de la construcción del modelo en Abaqus). Finalmente se trata el fichero para eliminar los espacios en blanco y el resto de líneas irrelevantes. Esta función tiene como parámetros de entrada “Abrir” que hace referencia al fichero ‘.txt’ con los datos de Abaqus CAE. Como “Salida” de la función se genera una matriz que tiene tantas filas como nodos posee el modelo y una columna para cada variable importada desde Abaqus (coordenadas, desplazamientos, deformaciones logarítmicas...).

```
function X= Datos_Abaqus(Abrir,Salida)
fid = fopen(Abrir);
tline = fgetl(fid);
Intermedio='Intermedio.txt';
fout = fopen(Intermedio, 'wt');
Modelo=0;
while ischar(tline)

    if startsWith(tline, 'Field Output')|startsWith(tline, ' Computation
')|startsWith(tline, ' Node')|startsWith(tline,
'Node')|startsWith(tline, ' Label')|startsWith(tline,
'Node')|startsWith(tline, ' Averaged')|startsWith(tline, ' Averaging')
    Modelo= Modelo;
    elseif startsWith (tline, '--')
    Modelo= Modelo+1;
    else
        if Modelo==1
            fprintf(fout,tline+" "+ "1.0"+ "\n");
            %1=Nucleo
        elseif Modelo==2
            fprintf(fout,tline+" "+ "2.0"+ "\n");
            %2=Placa
        elseif Modelo==3
```

```
fprintf(fout,tline+" "+ "3.0"+ "\n");
%3=Media
elseif Modelo==4
    fprintf(fout,tline+" "+ "4.0"+ "\n");
%4=Adventicia
elseif Modelo==5
    fprintf(fout,tline+" "+ "5.0"+ "\n");
%5=Intima
end
end

tline = fgetl(fid);
end
fclose(fid);
fclose(fout);
%El segundo paso es para eliminar las líneas sueltas que quedan en la
%operacion anterior
fid2 = fopen(Intermedio)
tline2 = fgetl(fid2);
fout2 = fopen(Salida, 'wt');
Modelo2=0;
while ischar(tline2)
    if startsWith(tline2, ' 1.0')|startsWith(tline2, ' 2.0')|startsWith(tline2, ' 3.0')|startsWith(tline2, ' 4.0')

        Modelo2=Modelo2+1;
    else
        fprintf(fout2,tline2+"\n");
    end
    tline2 = fgetl(fid2);
end

fclose(fid2);
fclose(fout2);
```

El siguiente paso, en el caso del programa 2D, consiste en quedarse con la sección de máxima estenosis. Por lo tanto se seleccionan únicamente los nodos que tienen la coordenada Z máxima (correspondiente en este caso con la sección de interés).

```
%% AHORA SE DEPURA PARA TENER LOS VALORES EN LA SECCION DE MAXIMA ESTENOSIS
filename = Salida;
A = importdata(filename);

valor_Section=max(A(:,4));
%Calcular el maximo para no tener que poner el valor
for i=1:length(A(:,1))
    if A(i,4)==valor_Section
        for j=1:length(A(1,:));
            B(i,j)=A(i,j);
        end
    end
end

%Ahora eliminamos los valores iguales a cero para quedarnos con la sección
v=0;
```

```
for k=1:length(B(:,1))
    if B(k,1)==0
        v=v+1;
        vec(v)=k;
    end
end
B(vec,:)=[];
```

En el caso del programa 3D se toman dos valores, uno el *valor₁* corresponde máximo de la sección, que es el mismo valor de la sección anterior, y el *valor₂* depende del tipo de malla del modelo. Deben ser dos valores lo suficientemente separados para obtener una rebanada tridimensional.

```
%% AHORA SE DEPURA PARA TENER UNA REBANADA 3D DE LA ZONA DE MÁXIMA ESTENOSIS
filename = Salida;
A = importdata(filename);
vector=sort(unique(A(:,4)), 'descend');
valor_1=vector(1);
valor_2= vector(90);
%Calcular el maximo para no tener que poner el valor
for i=1:length(A(:,1))
    if A(i,4)>valor_2 && A(i,4)<valor_1
        for j=1:length(A(1,:));
            B(i,j)=A(i,j);
        end
    end
end
%Ahora eliminamos los valores iguales a cero para quedarnos con la sección
v=0;
for k=1:length(B(:,1))
    if B(k,1)==0
        v=v+1;
        vec(v)=k;
    end
end
B(vec,:)=[];
```

Finalmente es necesario realizar un proceso para eliminar los nodos repetidos, ya que en un nodo puede pertenecer a más de una región (adventicia, media...) y por lo tanto aparecerá dos o más veces, dando problemas al mallar y procesar los resultados.

```
elementos(:)=NaN;
ve=0;
for m=1:length(B(:,1))
    if ismember(B(m,1),elementos)==1
        ve=ve+1;
        vect(ve)=m;
    end
    elementos(m)=B(m,1);
end
B(vect,:)=[];

X=B;
end
```

Cálculo de la Deformación

La siguiente función importante es aquella que permite el cálculo de las deformaciones. El cálculo en el software de elementos finitos Abaqus, permite obtener los resultados del análisis hiperelástico no lineal desde el incremento de presión 0 a 140 mmHg. Sin embargo, para simular los datos del IVUS es necesario obtener la deformación calculada de forma lineal entre un incremento próximo a los 5 mmHg, y ello se consigue con esta función. El siguiente código pertenece al programa 2D que tiene como parámetros de entrada tiene el modelo PDE desarrollado previamente (se explica en el *Programa Principal*) y una matriz $A1$ que tiene tantas filas como nodos y una columna con los nodos, otras con las coordenadas X e Y y otras dos con los desplazamientos uX y uY (la matriz $A1$ corresponde con la salida de la función anterior de lectura de datos). Las salidas de esta función son dos matrices, una con las deformaciones en coordenadas cartesianas y otra en principales. Tal y como se ha explicado en el punto 3 de la memoria, al calcular las deformaciones en 2D se asume deformación plana considerando únicamente los desplazamientos en los ejes X e Y , mientras que en el cálculo del modelo 3D se considera el desplazamiento en los 3 ejes.

Dentro de este código se desarrollan las funciones de forma elementales de elementos triangulares lineales y se calcula la deformación dentro de cada elemento (que al ser lineal el valor permanece constante en todos sus nodos), y ahí se guarda la deformación en cada uno de los nodos del elemento. Finalmente, para calcular la deformación nodal se suman las deformaciones de todos los elementos a los que pertenece cada nodo y se divide por el número total de elementos. En el siguiente código se implementan las ecuaciones desarrolladas en el punto 3 de la memoria para obtener las deformaciones en cartesianas a partir de los desplazamientos y también las deformaciones principales mediante un cálculo de valores propios.

```
function[strain_nodo,strain_principal]= Strain_X_Y(modelo,A1)
model=modelo;
A=A1;
X=A(:,2);
Y=A(:,3);
u_X=A(:,4);
u_Y=A(:,5);
%% FUNCIONES DE FORMA DE CADA ELEMENTO
syms e n
N1= 1-e-n;
N2= e;
N3= n;
N = [N1 N2 N3];
Elementos=length(model.Mesh.Elements); % numero de elementos
%% CALCULO DE LA DEFORMACIÓN EN CADA ELEMENTO
Epsilon_x=zeros([1 length(A)]);
Epsilon_y=zeros([1 length(A)]);
Epsilon_xy=zeros([1 length(A)]);
indice_nodos=zeros([1 length(A)]);
Epsilon_I=zeros([1 length(A)]);
Epsilon_II=zeros([1 length(A)]);
Epsilon_III=zeros([1 length(A)]);
for i=1:Elementos %Deformacion en XY
    nodos=model.Mesh.Elements(:,i);
    coord_x=[X(nodos(1)) X(nodos(2)) X(nodos(3))];
    coord_y=[Y(nodos(1)) Y(nodos(2)) Y(nodos(3))];
    desp_x=[u_X(nodos(1)) u_X(nodos(2)) u_X(nodos(3))];
    desp_y=[u_Y(nodos(1)) u_Y(nodos(2)) u_Y(nodos(3))];
    xx(e,n) =N*coord_x';
    yy(e,n)=N*coord_y';
    [G,b]=equationsToMatrix([xx,yy]);
    syms x y
    bb=b+[x;y];
```

```
sol=inv(G)*bb;
ee=sol(1);
nn=sol(2);
u_x2 =N*desp_x';
u_y2=N*desp_y';
format short
Epsilon_x1=double(diff(u_x2,e)*diff(ee,x)+diff(u_x2,n)*diff(nn,x));
Epsilon_y1=double(diff(u_y2,e)*diff(ee,y)+diff(u_y2,n)*diff(nn,y));
Epsilon_xy1=double(0.5*(diff(u_x2,e)*diff(ee,y)+diff(u_x2,n)*diff(nn,y)+
diff(u_y2,e)*diff(ee,x)+diff(u_y2,n)*diff(nn,x)));
tensor_def=[Epsilon_x1 Epsilon_xy1 0; Epsilon_xy1 Epsilon_y1 0; 0 0 0];
Epsilon_principal=sort(polyeig(tensor_def),'ascend');
Epsilon_max_el=Epsilon_principal(3);
Epsilon_min_el=Epsilon_principal(1);
Epsilon_med_el=Epsilon_principal(2);

for j=1:3
    indice_nodos(nodos(j))=indice_nodos(nodos(j))+1;
    Epsilon_x(nodos(j))=Epsilon_x(nodos(j))+Epsilon_x1;
    Epsilon_y(nodos(j))=Epsilon_y(nodos(j))+Epsilon_y1;
    Epsilon_xy(nodos(j))=Epsilon_xy(nodos(j))+Epsilon_xy1;
    Epsilon_I(nodos(j))=Epsilon_I(nodos(j))+Epsilon_max_el;
    Epsilon_II(nodos(j))=Epsilon_II(nodos(j))+Epsilon_med_el;
    Epsilon_III(nodos(j))=Epsilon_III(nodos(j))+Epsilon_min_el;
end
Epsilon_x1=0;
Epsilon_y1=0;
Epsilon_xy1=0;
Epsilon_max_el=0;
Epsilon_min_el=0;
Epsilon_med_el=0;
end

for i=1:length(X)
    Epsilon_x(i)=Epsilon_x(i)/indice_nodos(i);
    Epsilon_y(i)=Epsilon_y(i)/indice_nodos(i);
    Epsilon_xy(i)=Epsilon_xy(i)/indice_nodos(i);
    Epsilon_I(i)=Epsilon_I(i)/indice_nodos(i);
    Epsilon_II(i)=Epsilon_II(i)/indice_nodos(i);
    Epsilon_III(i)=Epsilon_III(i)/indice_nodos(i);
end
strain_nodo=[Epsilon_x; Epsilon_y; Epsilon_xy];
strain_principal=[Epsilon_I; Epsilon_II; Epsilon_III];
end
```

En el caso del modelo 3D en la entrada hay que introducir las coordenadas y desplazamientos en el eje Z y la salida devolverá las deformaciones en Z y todas sus componentes tangenciales. Dentro del código se incluyen las funciones de forma de tetraedros lineales y las deformaciones de los componentes tridimensionales tal y como se ha comentado en la memoria.

```
syms e n p
N1=1-e-n-p;
N2=e;
N3=n;
N4=p;
N=[N1 N2 N3 N4];
```

Programa Principal

Como su propio nombre indica, el programa principal, es el núcleo del código. Se divide en diferentes secciones dentro de las cuales se llaman las funciones anteriores.

Sección 1: Toma de datos

En esta parte se toman los datos de Abaqus de dos instantes próximos entre ellos, simulando dos imágenes consecutivas tomadas mediante IVUS, entre las que debe haber un incremento de presión entre 5 y 7.5mmHg [7, 15, 16] para poder tomar la hipótesis de comportamiento lineal. Para ello se pide por pantalla el nombre de los archivos de texto exportados de Abaqus y se llama a la función *DatosAbaqus()* explicada anteriormente.

```
prompt = 'Introduzca el nombre del fichero de entrada (instante 1): ';
Abrir1 = input(prompt);
instante1 = Datos_Abaqus(Abrir1,'Salida1.txt');
prompt = 'Introduzca el nombre del fichero de entrada (instante 2): ';
Abrir2 = input(prompt);
instante2 = Datos_Abaqus(Abrir2,'Salida2.txt');
```

Sección 2: Obtener valores de coordenadas y desplazamientos

Tras haber cargado los datos se clasifican en coordenadas y desplazamientos. Las coordenadas del instante 1 se consideran las de referencia para generar la geometría y los desplazamientos entre los dos instantes se calculan como la resta de ambos.

```
X=instante1(:,2); %Coordenadas X del instante 1
Y=instante1(:,3); %Coordenadas Y del instante 1
X_2=instante2(:,2); %Coordenadas X del instante 2
Y_2=instante2(:,3); %Coordenadas Y del instante 2
%Desplazamientos
Ux_1=instante1(:,5);
Uy_1=instante1(:,6);
Ux_2=instante2(:,5);
Uy_2=instante2(:,6);
Ux=Ux_2-Ux_1;
Uy=Uy_2-Uy_1;
```

En 3D tan solo se añaden las líneas de código referentes a la coordenada y desplazamiento en Z.

Sección 3: Mallado

Se realiza el mallado de las coordenadas nodales mediante la triangulación de Delaunay y se eliminan los elementos generados de forma errónea (aquellos que intentan malla el lumen o cerrar zonas de la geometría abierta) comparando el radio de la circunferencia que pasa por los 3 nodos de cada elemento [41].

```
P=[X Y];
T = delaunayTriangulation(P);%los nodos son las aristas de los triangulos
figure(1),hold on,
title('Malla inicial')
triplot(T)
elemt=size(T.ConnectivityList);

%Depuración de malla calculando el radio que pasa por los 3 nodos
for i=1:elemt
    punto1=T.ConnectivityList(i,1);
    punto2=T.ConnectivityList(i,2);
```

```
punto3=T.ConnectivityList(i,3);
coord1=T.Points(punto1,:);
coord2=T.Points(punto2,:);
coord3=T.Points(punto3,:);
puntos=[coord1; coord2 ;coord3];
[R,xcyc] = fit_circle_through_3_points(puntos);%Con esta funcion calculo el radio
Radios(i)=R;
if R>0.5
    Elementos_A_eliminar(i)=i;
end
end

list=T.ConnectivityList;
Elementos_A_eliminar = unique(Elementos_A_eliminar);
Elementos_A_eliminar(1)=[];
list(Elementos_A_eliminar,:)=[];
tru=triangulation(list,T.Points);
figure(2), hold on,
title('Malla Procesada')
tripplot(tru);
```

En el programa 3D se utiliza la función *tetramesh* en lugar de *tripplot*. Además, para depurar la malla se compara el radio de la esfera que contiene a los 4 nodos de cada elemento mediante la función *sphereFit()* [39].

Sección 4: Crear modelo *PDE*

En esta sección se hace uso de la toolbox de Matlab de Derivadas Parciales (*Partial Differential Equation Toolbox, PDE*) que permite crear modelos *PDE* y calcular problemas sencillos de elasticidad lineal. En este caso se introducen como geometría la malla generada en la sección anterior y como resultado del problema los desplazamientos nodales. Además se pueden representar los desplazamientos con la función *pdeplot* que en 3D pasa a ser *pdeplot3d*.

```
model = createpde(2); %el numero entre paréntesis depende del nº de
%resultados que tengamos, en este caso se van a meter los desplazamientos
% en 'X' y en 'Y', por lo tanto es 2
geometryFromMesh(model,tru.Points',tru.ConnectivityList');
u=[Ux; Uy]; %u tiene que tener desplazamientos en X e Y
results = createPDEResults(model,u,'stationary');

figure(3), hold on,
title('Modelo PDE')
pdegplot(model); %con esto el modelo queda definido tanto con la geometria como con la malla anterior

figure(4),
pdeplot(model,'XYData',Ux,'Mesh','on','ColorMap','Jet')%Aqui se ve el desplazamiento en X
hold on,
title('Desplazamiento en X [mm] + Malla')
```

Sección 5: Cálculo de Gradienes

Una de las grandes ventajas de usar Matlab para representar las variables en lugar de Abaqus CAE, es la posibilidad de representar los gradienes de cualquier variable. Puede calcularse de dos maneras, con la función *evaluateGradient* o trabajando con los resultados del modelo PDE, que directamente almacena los gradienes de cada variable. Además, se calcula el módulo de los gradienes para poder representarlo también. En el programa 3D basta con tener en cuenta las dimensiones correspondientes.

%Opcion1

```
[gradx,grady] = evaluateGradient(results,X,Y,[1,2]);
figure(8),
hold on
quiver(X(:,1),Y(:,1),gradx(:,1),grady(:,1)) %1 corresponde al desplazamiento en X
% y 2 en Y
title('Gradiente del Desplazamiento en X')
hold off

for i=1:length(gradx) %el gradiente de x corresponde al gradiente de la
% deformación principal maxima
    moduloX(i)=double(norm([gradx(i,1) grady(i,1)]));
    moduloY(i)=double(norm([gradx(i,2) grady(i,2)]));
end
figure(9), hold on,
pdeplot(model,'XYData',moduloX,'ColorMap','Jet')%Aqui se ve el modulo del
% gradiente del desplazamiento en X
title('Modulo Gradiente Desplazamiento en X')

%Opcion2

figure(8),
hold on
pdeplot(model,'XYData',results.XGradient(:,1),'ColorMap','Jet')% corresponde
% al desplazamiento en X y 2 en Y
title('Gradiente del Desplazamiento en X')
hold off

for i=1:length(results.XGradients) %el gradiente de x corresponde al
% gradiente de la deformación principal maxima
    modulo_E_max(i)=double(norm([results.XGradients(i,1) results.YGradients(i,1)]));
    modulo_E_min(i)=double(norm([results.XGradients(i,2) results.YGradients(i,2)]));
end
```

Sección 6: Calcular Deformaciones

Ahora se calculan las deformaciones ($\varepsilon_x, \varepsilon_y$ y ε_{xy}) a partir de los desplazamientos. Para ello se realiza llamada a la función *StrainXY()* que se ha explicado anteriormente en la función del cálculo de la deformación, y da como salida las deformaciones nodales en coordenadas cartesianas y las deformaciones principales.

```
Matriz=[instante1(:,1) X Y Ux Uy]; %esta matriz tiene la geometria inicial y
% los desplazamientos considerando el instante1 como referencia
```

```
[strain_nodo, strain_principal]= Strain_X_Y(model,Matriz); %Introduzco la
% funcion que he desarrollado para varios elementos, devuelve una matriz de
% deformaciones
```

```
strainX=strain_nodo(1,:)';
strainY=strain_nodo(2,:)';
strainXY=strain_nodo(3,:)';
Epsilon_I=strain_principal(1,:)';
Epsilon_II=strain_principal(2,:)';
Epsilon_III=strain_principal(3,:');
```

```
figure(14),
pdeplot(model,'XYData',strainX,'ColorMap','Jet')
```

```
hold on,  
title('Deformacion X')
```

Adicionalmente, si se han cargado como datos de Abaqus las deformaciones logarítmicas del modelo se pueden representar (la resta de los dos instantes de tiempo) para poder realizar una comprobación de los resultados.

Sección 7: Gradiente de las Deformaciones

Tal y como se ha comentado es muy interesante visualizar los gradientes de las variables, y para ello hay que crear un nuevo modelo *PDE* para generar los resultados y sus gradientes. En el nuevo modelo *PDE* se introducen las variables de las que se quiere obtener el gradiente. Este proceso se repite para todas las variables.

```
Def_nodos=[strainX; strainY; strainXY]; %Se juntan las deformaciones en un  
% vector  
modelo2 = createpde(3);  
geometryFromMesh(modelo2,tru.Points',tru.ConnectivityList');  
results2 = createPDEResults(modelo2,Def_nodos,'stationary');  
  
for i=1:length(results2.XGradients) %el gradiente de x corresponde al  
% gradiente de la deformación principal maxima  
    modulo_E_X(i)=double(norm([results2.XGradients(i,1) results2.YGradients(i,1)]));  
    modulo_E_Y(i)=double(norm([results2.XGradients(i,2) results2.YGradients(i,2)]));  
    modulo_E_XY(i)=double(norm([results2.XGradients(i,3) results2.YGradients(i,3)]));  
end
```

Sección 8: Cálculo de las Deformaciones Equivalentes

Las deformaciones principales ya se han obtenido en el paso anterior y a partir de ellas se pueden obtener las deformaciones equivalentes del criterio de von Mises (3.17) y Tresca (3.18). También se calcula el índice de anisotropía (3.19). El gradiente de estas variables se obtiene exactamente igual que en la *Sección 7* introduciendo como resultado de la *PDE* la variable en cuestión.

```
% Criterio de von Mises  
  
for i=1:length(Epsilon_I)  
    mises(i)=sqrt(((Epsilon_I(i)-Epsilon_II(i))^2+(Epsilon_II(i)-Epsilon_III(i))^2+(Epsilon_I(i)-Epsilon_III(i))^2))  
end  
  
figure(29),  
pdeplot(model,'XYData',mises,'ColorMap','Jet')  
hold on, set(gcf,'color','w'),daspect([1,1,1]),  
title('Deformacion von Mises')  
  
% Tresca  
for i=1:length(Epsilon_I)  
    Tresca(i)=(Epsilon_I(i)-Epsilon_III(i));  
end  
  
figure(27),  
pdeplot(model,'XYData',Tresca,'ColorMap','Jet')  
hold on, set(gcf,'color','w'),daspect([1,1,1]),  
title('Deformacion Tresca')  
  
% ANISOTROPIC INDEX considerando que landa=epsilon+1.
```

```
for i=1:length(Epsilon_I)
    landa1(i)=Epsilon_I(i)+1;
    landa2(i)=Epsilon_II(i)+1;
    landa3(i)=Epsilon_III(i)+1;
    landa(i)=(landa1(i)+landa2(i)+landa3(i))/3;
    FA(i)=(sqrt(3)/sqrt(2))*(sqrt((landa1(i)-landa(i))^2+(landa2(i)-landa(i))^2+
    (landa3(i)-landa(i))^2)/sqrt(landa1(i)^2+landa2(i)^2+landa3(i)^2));
end

figure(31),
pdeplot(model,'XYData',FA,'ColorMap','Jet')
hold on, set(gcf,'color','w'),daspect([1,1,1]),
title('Indice de Anisotropía')
```

Sección 9: Cálculo de las Deformaciones Cilíndricas

Se calculan las deformaciones en coordenadas cilíndricas a partir de las cartesianas tal y mediante la multiplicación por la matriz de cambio de base (*Ecuación 3.20*):

```
for i=1:length(X)
    tita(i)=atan(Y(i)/X(i));
    Epsilon_r2(i)=strainX(i)*cos(tita(i))^2+strainY(i)*sin(tita(i))^2+strainXY(i)*sin(2*tita(i));
    Epsilon_t2(i)=strainX(i)*sin(tita(i))^2+strainY(i)*cos(tita(i))^2-strainXY(i)*sin(2*tita(i));
    Epsilon_rt2(i)=sin(tita(i))*cos(tita(i))*(strainY(i)-strainX(i))+strainXY(i)*cos(2*tita(i));
end
```

Sección 10: Cálculo del Error entre Abaqus CAE y Matlab

Para poder validar el método, se realiza una comparativa entre los resultados obtenidos en el código desarrollado en Matlab y las deformaciones obtenidas en Abaqus CAE (resta de las deformaciones logarítmicas en cada instante...). Se calculan los errores relativos (*Ecuación 3.15*) y absoluto (*Ecuación 3.16*). Se puede calcular para cualquier variable de la cual se dispongan los resultados de Abaqus.

```
for i=1:length(X)
    error_EpsilonX(i)=abs((Epsilon_x(i)-instante1(i,8))/(instante1(i,8)))*100;
    error_EpsilonY(i)=abs((Epsilon_y(i)-instante1(i,9))/(instante1(i,9)))*100;
    error_EpsilonZ(i)=abs((Epsilon_z(i)-instante1(i,10))/(instante1(i,10)))*100;
    error_EpsilonXY(i)=abs((2*Epsilon_xy(i)-instante1(i,11))/(instante1(i,11)))*100;
    error_EpsilonXZ(i)=abs((2*Epsilon_xz(i)-instante1(i,12))/(instante1(i,12)))*100;
    error_EpsilonYZ(i)=abs((2*Epsilon_yz(i)-instante1(i,13))/(instante1(i,13)))*100;
end
```

Sección 11: Representación de las Variables

Una vez calculadas todas las variables es posible representarlas. Es posible modificar los colores de la representación y los límites. En el siguiente código se muestra la representación de la deformación radial para el modelo 2D y 3D.

```
figure(33),
pdeplot(model,'XYData',Epsilon_r2,'ColorMap','Jet')
hold on, set(gcf,'color','w'),caxis([-0.002, 0.0025]),daspect([1,1,1]),
title('Deformacion R')
figure(27),
pdeplot3D(model,'ColorMapData',Epsilon_R),view(2),
hold on, caxis([-0.0015 0.0025])
title('Deformación Radial (Matlab)')
```

Anexo III: Validación del Método

El objetivo de este Anexo es el de validar el código desarrollado en el *Anexo II* mediante modelos simples que permitan verificar el código base y poder aplicarlo a las geometrías de interés. En definitiva, este apartado explica la creación del código del anexo anterior y su validación con modelos simples que incrementan la dificultad. Ha sido necesarios realizar estos pasos con el fin de construir y verificar paso a paso el método antes de aplicarlo a las geometrías complejas de interés. El proceso para desarrollar el programa se basa en diferentes pasos.

- **Paso 1:** se desarrolla un programa capaz de leer los nodos de un modelo simple 2D mallado y calculado en el software Abaqus CAE y calcular la deformación cartesiana en los nodos con la hipótesis de deformación plana.
- **Paso 2:** se adapta el paso anterior para ser capaz de leer un modelo 3D de Abaqus CAE, quedarse con una sección 2D determinada, remallarla y calcular la deformación cartesiana y cilíndrica.
- **Paso 3:** se construye un código capaz de leer un modelo 3D de Abaqus CAE y analizar una rebanada 3D, mallar y calcular deformaciones cartesianas y cilíndricas.

Los Pasos 1, 2 y 3 sirven para construir el programa desarrollado en el Anexo II: Metodología desarrollada en Matlab y validar el método.

Paso 1

Se crea un modelo 2D sencillo en Abaqus CAE y se imponen unos desplazamientos en los nodos de la malla (*Figura 6.10.a*). Tras calcular, se exporta el fichero de resultados con las coordenadas X e Y y los desplazamientos en los nodos, tal y como se observa en la *Figura 6.10.b*.

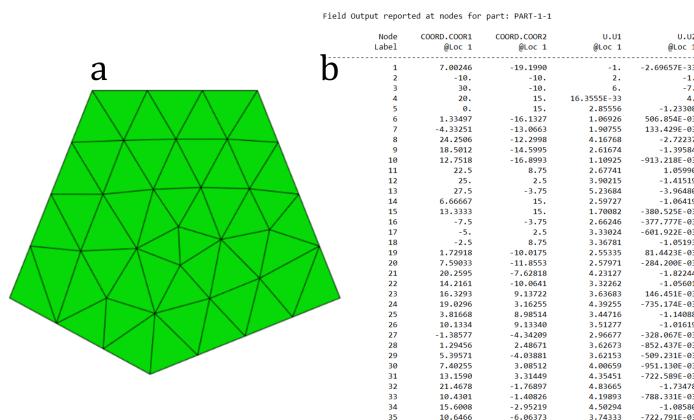


Figura 6.10: a) Modelo 2D realizado en Abaqus CAE, b) Fichero de datos.

A continuación se importa el fichero de datos en Matlab y se guardan las coordenadas de los nodos. Con ellos se realiza un mallado con triángulos lineales mediante una triangulación de Delaunay (*Figura 6.11*). En Matlab existe una toolbox que permite crear y calcular elementos en derivadas parciales (*PDE*) de problemas de elasticidad lineal sencillos con materiales isótropos. En nuestro caso, al tener geometrías complejas y ser problemas no lineales, no es posible resolver el problema. Sin embargo, se pueden introducir los desplazamientos calculados en Abaqus CAE como solución del *PDE*. Por lo tanto se consigue un modelo *PDE* con las coordenadas y los desplazamientos calculados previamente en cada uno de los nodos.

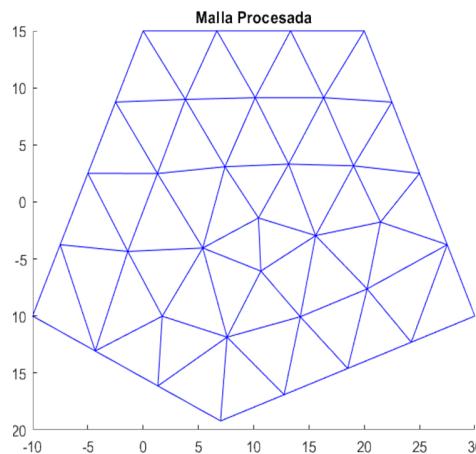


Figura 6.11: Mallado Delaunay.

Considerando los elementos triangulares lineales y tal y como se explica en el punto 3 de la memoria, se calculan las deformaciones en cartesianas. La formulación explicada (*Ecuaciones 3.2 a 3.10*) permite obtener la deformación a nivel elemental. Para particularizar en los nodos, al ser elementos lineales, basta con sumar el valor de la deformación de cada uno de los elementos a los que pertenece y dividir por el número de elementos. En la *Figura 6.12.a* se muestran los nodos de una geometría y en la *Figura 6.12.b* los elementos. Por lo tanto para calcular la deformación del nodo 17 (n_{17}), representado en la *Figura 6.13*, se utilizaría la siguiente ecuación (6.1):

$$\varepsilon_{n17} = \frac{\varepsilon_{e18} + \varepsilon_{e10} + \varepsilon_{e11}}{3} \quad (6.1)$$

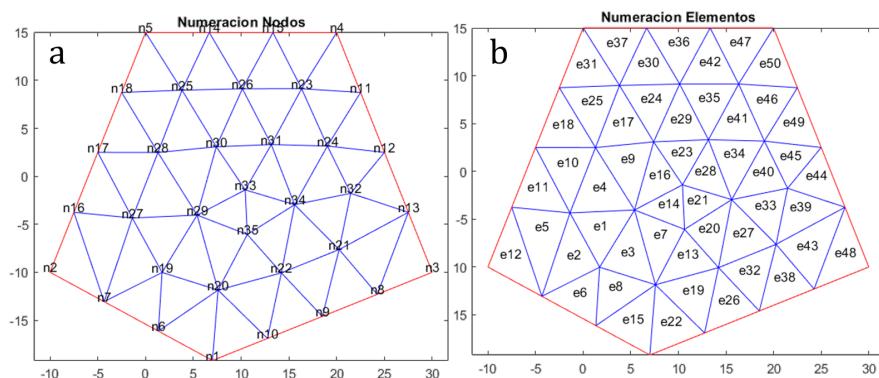


Figura 6.12: a) Numeración de los nodos y b) numeración de los elementos.

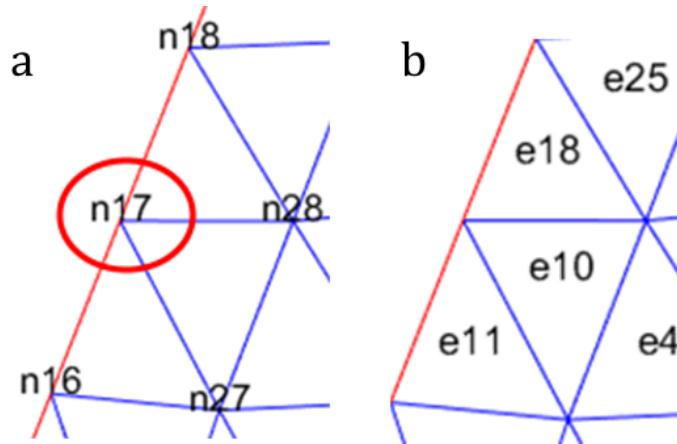


Figura 6.13: Ejemplo del cálculo de la deformación nodal a partir de la deformación elemental.

También se ha probado a realizar una ponderación con el área de cada elemento tal y como se indica en la *Ecuación 6.2*. Sin embargo los errores eran mayores que siguiendo el primer método, por lo que no se utilizó esta aproximación. En la *Figura 6.14* se muestran las deformaciones nodales en coordenadas cartesianas calculadas con y sin ponderación. Además, en dicha tabla se calculan los errores relativos entre ambos métodos y los resultados obtenido mediante el cálculo de Abaqus CAE donde se observa que al ponderar estos errores aumentan de forma considerable.

$$\varepsilon_{n17} = \frac{\varepsilon_{e18} \cdot \text{Area}_{e18} + \varepsilon_{e10} \cdot \text{Area}_{e10} + \varepsilon_{e11} \cdot \text{Area}_{e11}}{\text{Area}_{e18} + \text{Area}_{e10} + \text{Area}_{e11}} \quad (6.2)$$

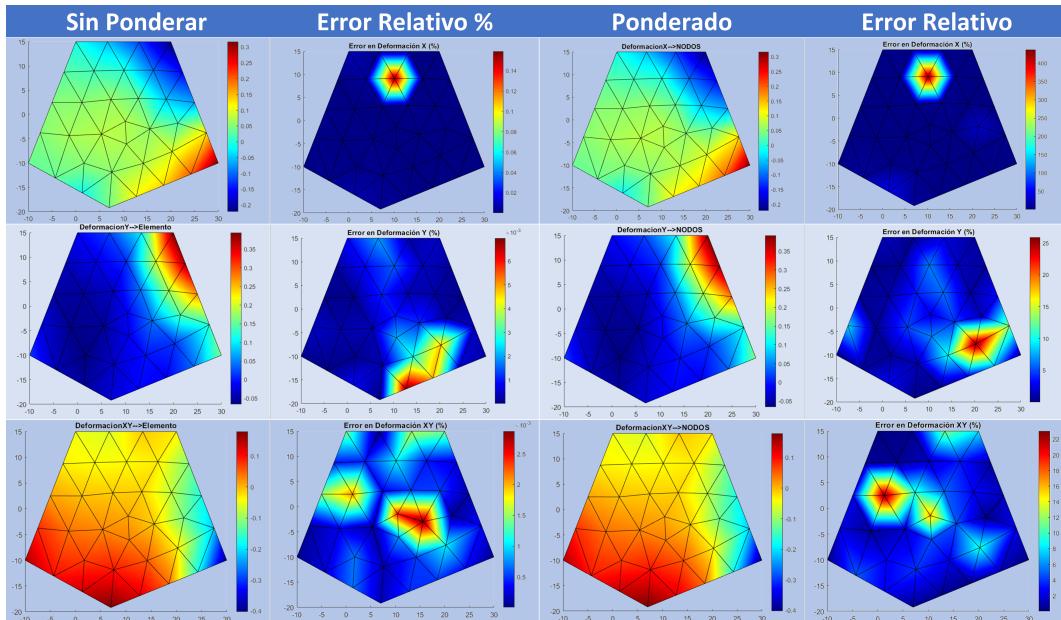


Figura 6.14: Comparación entre las deformaciones nodales calculadas mediante el la suma sin ponderación y la ponderación por área.

En este modelo, al ser un cálculo simple de deformación elástico lineal, es posible comparar la deformación obtenida en Abaqus CAE con la deformación obtenida mediante el método implantado en Matlab. En la *Figura 6.15* se realiza una comparativa de ambos resultados y se representan los errores relativos y absolutos entre ellos. Es importante señalar que en algunos nodos el error relativo es muy alto, esto se debe a que la deformación en ese nodo específico es próxima a cero y hace que el error relativo sea muy grande. Por este motivo se ha decidido representar también el error absoluto.

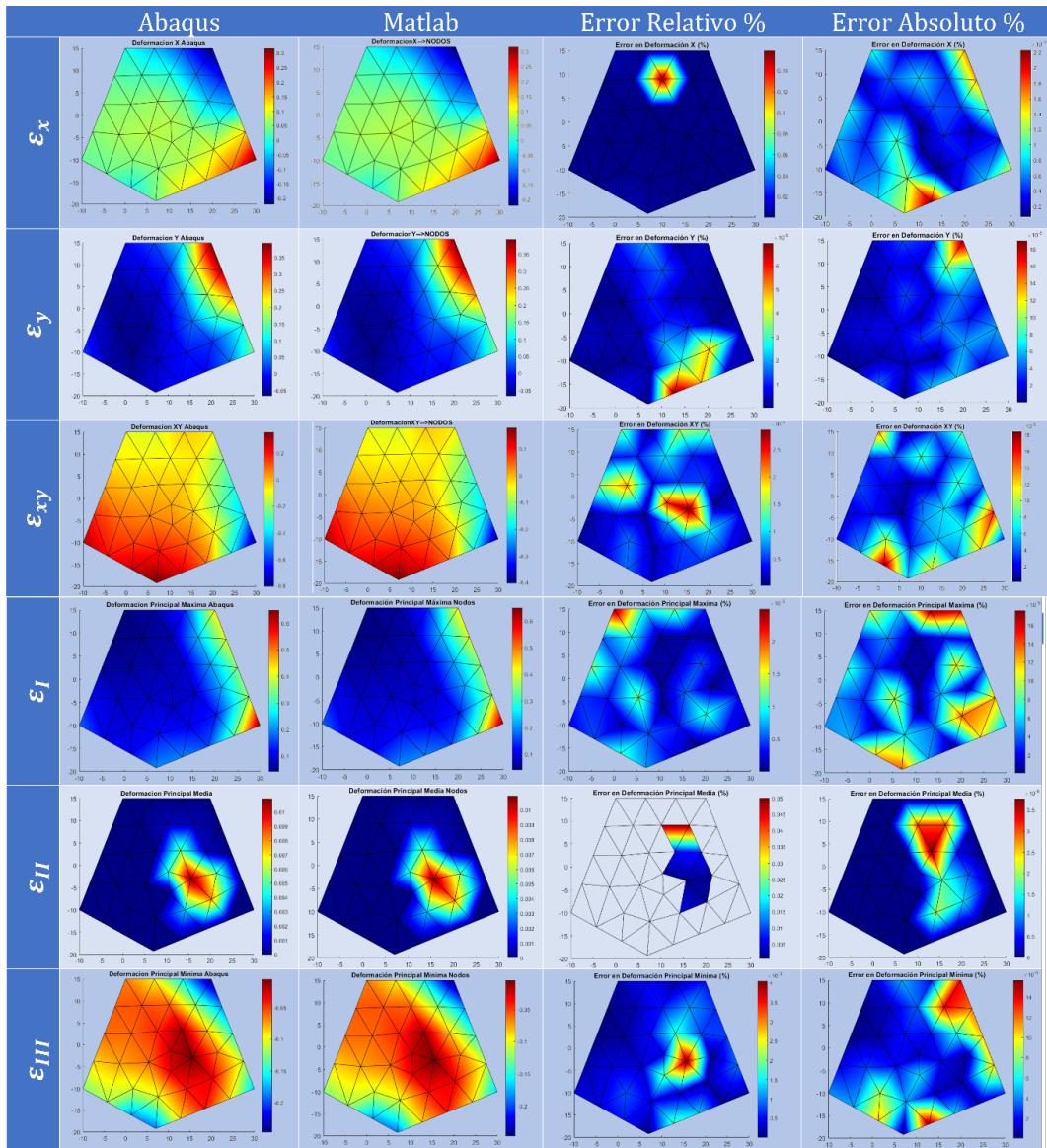


Figura 6.15: Comparación entre deformaciones cartesianas y principales calculadas en Abaqus CAE y en el método desarrollado en Matlab.

Una vez validadas las deformaciones en cartesianas se pueden obtener las deformaciones principales calculadas a nivel elemental y las deformaciones del criterio de von Mises o Tresca, que se puede apreciar en la Figura 6.16. Además, al utilizar la toolbox de PDE de Matlab, se pueden obtener con facilidad los gradientes de cada una de las variables, y al operar con ellas se puede representar el módulo de dichos gradientes, tal y como se explicará más adelante. Al realizar una comparación entre los valores de las deformaciones principales de Abaqus CAE y Matlab se ha observado un máximo error relativo inferior al 0.05 %, por lo que el método desarrollado es capaz de obtener muy buenos resultados.

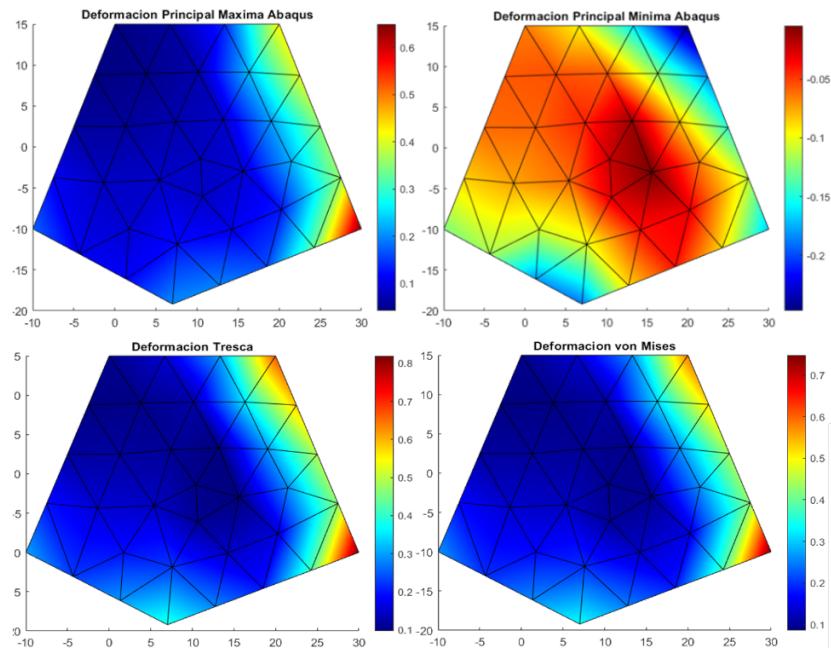


Figura 6.16: Deformaciones principales y equivalentes de Tresca y von Mises.

Con el fin de calcular y comprobar las deformaciones cilíndricas se crea otro modelo 2D muy simple (ver *Figura 6.17*) y se calculan las deformaciones cartesianas de la misma manera que en el modelo anterior, pero a continuación se calculan las deformaciones cilíndricas a partir de las cartesianas mediante las ecuaciones de cambio de base (*Ecuación 3.20*) desarrolladas en la memoria. En la *Figura 6.18* Se observan las deformaciones cilíndricas obtenidas en el código de Matlab y las calculadas en Abaqus CAE. Además se representa el error relativo y absoluto entre ambas.

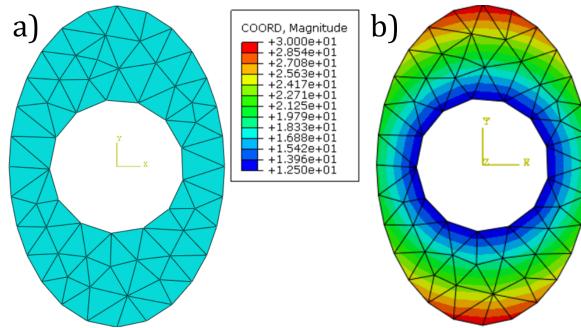


Figura 6.17: a) Modelo 2D de Abaqus CAE, b) Módulo de desplazamientos en los resultados.

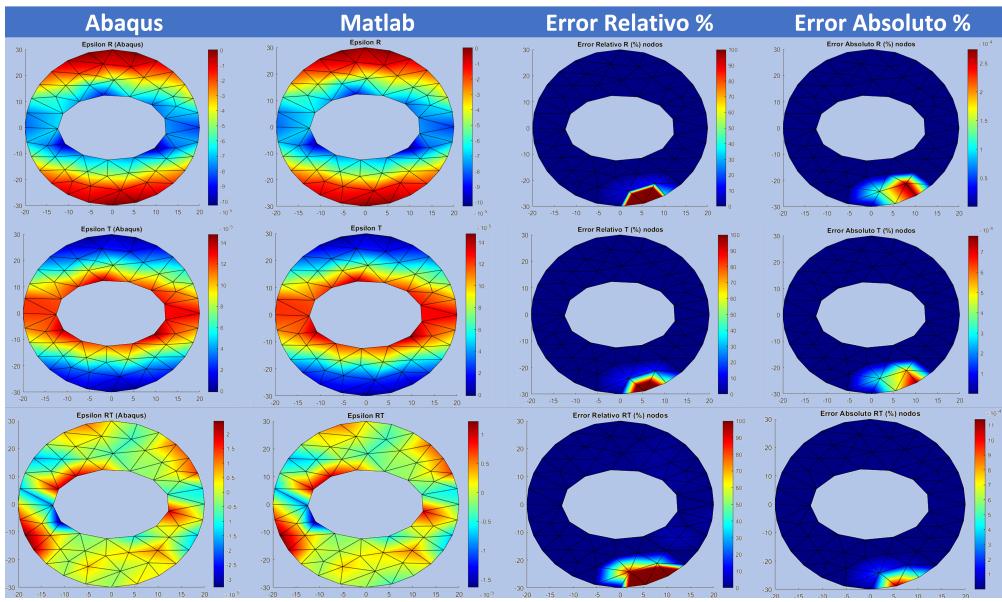


Figura 6.18: Comparación entre deformaciones cilíndricas calculadas en Abaqus y Matlab.

Paso 2

En este paso se sigue el mismo proceso que en el caso anterior, la única diferencia es que se parte de un modelo tridimensional de Abaqus CAE (mallado con tetraedros lineales en lugar de triángulos lineales), donde el sistema de coordenadas cartesianas ha sido colocado en el centro de la geometría (Figura 6.19.a). Se importan los datos de las coordenadas y desplazamientos nodales a Matlab, donde se procesan solo los de una sección 2D determinada. Al realizar el mallado con Delaunay aparecen elementos sobrantes (Figura 6.19.b) por ello se depura la malla analizando calculando el radio del círculo circunscrito en los 3 nodos de cada elemento [41] (Figura 6.19.c).

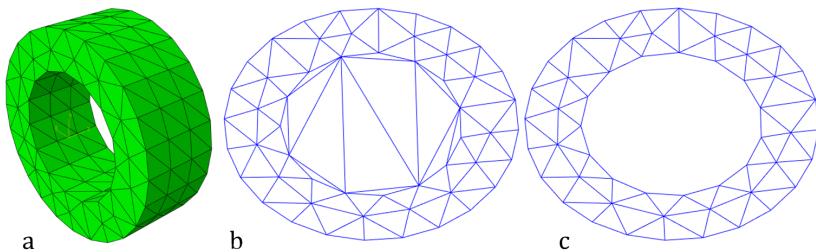


Figura 6.19: a) Modelo 3D de Abaqus CAE, b) Malla de Delaunay de una sección 2D en Matlab, c) Depurado de los elementos de la malla.

Se calculan las deformaciones en cartesianas de la misma forma que en el caso anterior y a partir de ellas se obtienen las deformaciones en coordenadas cilíndricas mediante la matriz de cambio de base desarrollada en el punto 3 de la memoria. En la Figura 6.20 se observan los resultados obtenidos en Abaqus y Matlab y los errores relativo y absoluto entre ambos. Se puede observar que el error en la deformación tangencial es muy alto, sin embargo, en el modelo 2D anterior el error es menor al 5 %. Esto se debe a que se parte de un modelo 3D de Abaqus, donde la malla y la resolución del problema se ha hecho con elementos tetraédricos lineales y al resolver en Matlab una única sección se utilizan elementos triangulares y se asume deformación plana. Esta diferencia a la hora de resolver puede causar esa variación en los resultados. También se analizan las deformaciones principales en la Figura 6.21, donde se observa que los errores son mucho más grandes que en el caso anterior. Estos errores derivan de utilizar la hipótesis de deformación plana y comparar los resultados con un cálculo 3D.

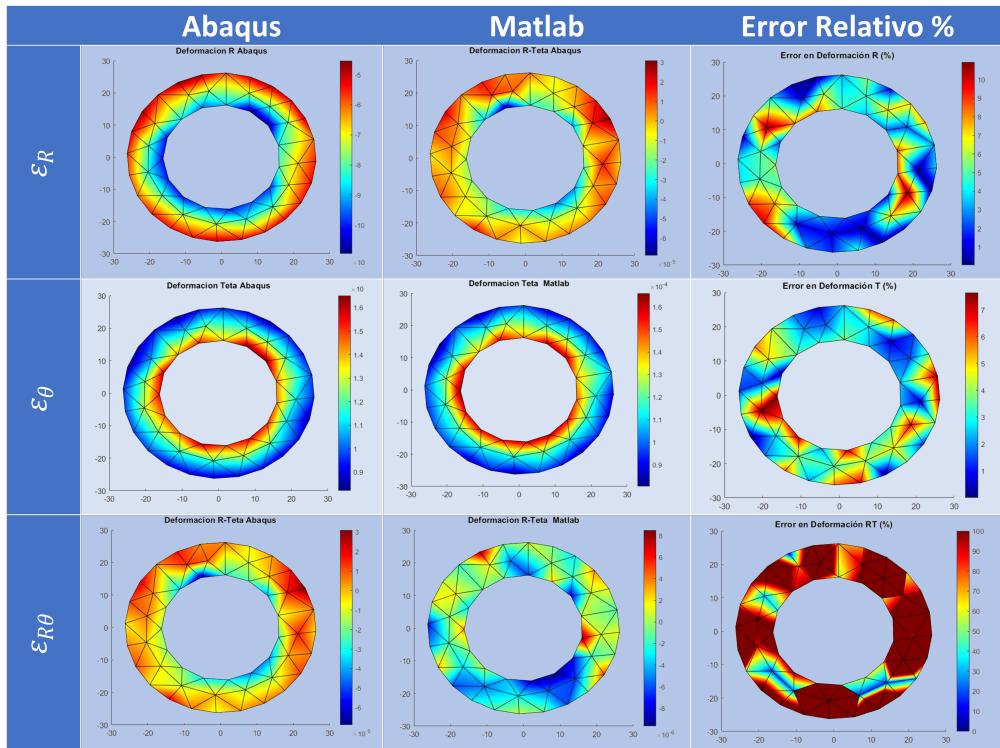


Figura 6.20: Comparación entre deformaciones cilíndricas calculadas en Abaqus y Matlab.

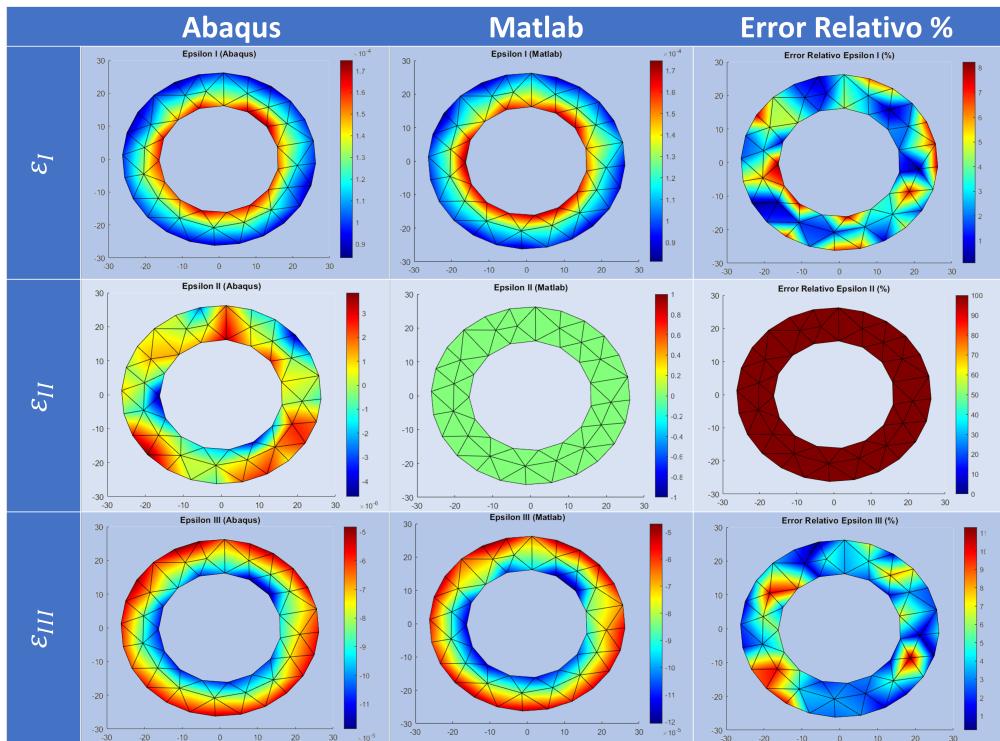


Figura 6.21: Comparación entre deformaciones principales calculadas en Abaqus y Matlab.

Paso 3

Por último se desarrolla el código que permite trabajar con modelos 3D (*Figura 6.22.a*) dentro de Matlab. Al igual que en los casos anteriores se utiliza un modelo de Abaqus, pero en este paso se trabaja con los datos de todas las variables y no con una única sección si no una rebanada 3D. Este caso la triangulación de *Delaunay* genera una red de tetraedros (*Figura 6.22.b*) y para depurar la malla (*Figura 6.22.c*) se calcula el radio de la esfera que contiene a los 4 nodos de cada elemento [39] (tetraedros lineales C3D4).

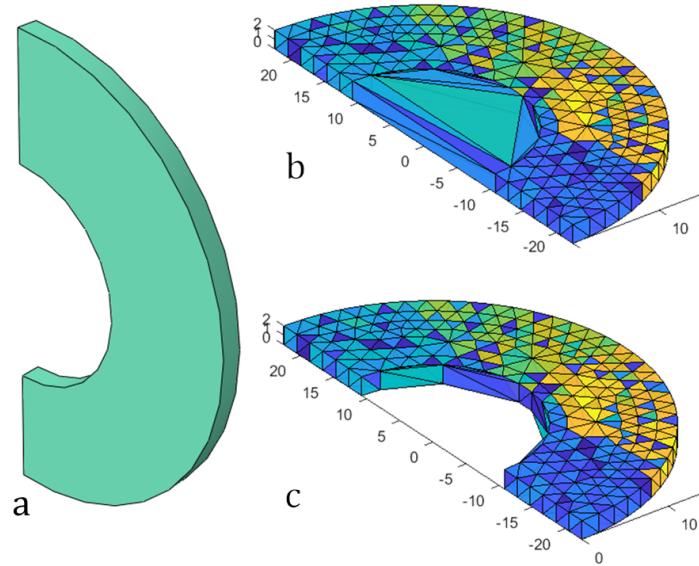


Figura 6.22: a) Modelo 3D en Abaqus CAE, b) Mallado con tetraedros en Matlab, c) Depuración de la malla.

Al utilizar elementos tridimensionales es necesario modificar la función desarrollada en los pasos anteriores para calcular la deformación, ya que hay una nueva dimensión. Se sigue el mismo método descrito en el apartado de 3D de la memoria. Se calculan las deformaciones cartesianas (*Figura 6.23*), cilíndricas (*Figura 6.24*) y principales (*Figura 6.25*), pero considerando todos los desplazamientos y se compara con los resultados obtenidos en Abaqus. En ambos casos son modelos 3D y mallados con tetraedros lineales lo que permite hacer una comparativa directa. Se puede observar que los errores relativos son altos, especialmente en las deformaciones cilíndricas, no obstante, los errores absolutos no tienen picos tan notables, por lo tanto, se debe al bajo valor de las deformaciones que hacen que el denominador del error relativo (*Ecuación 3.15*) sea próximo a cero y el error se incremente.

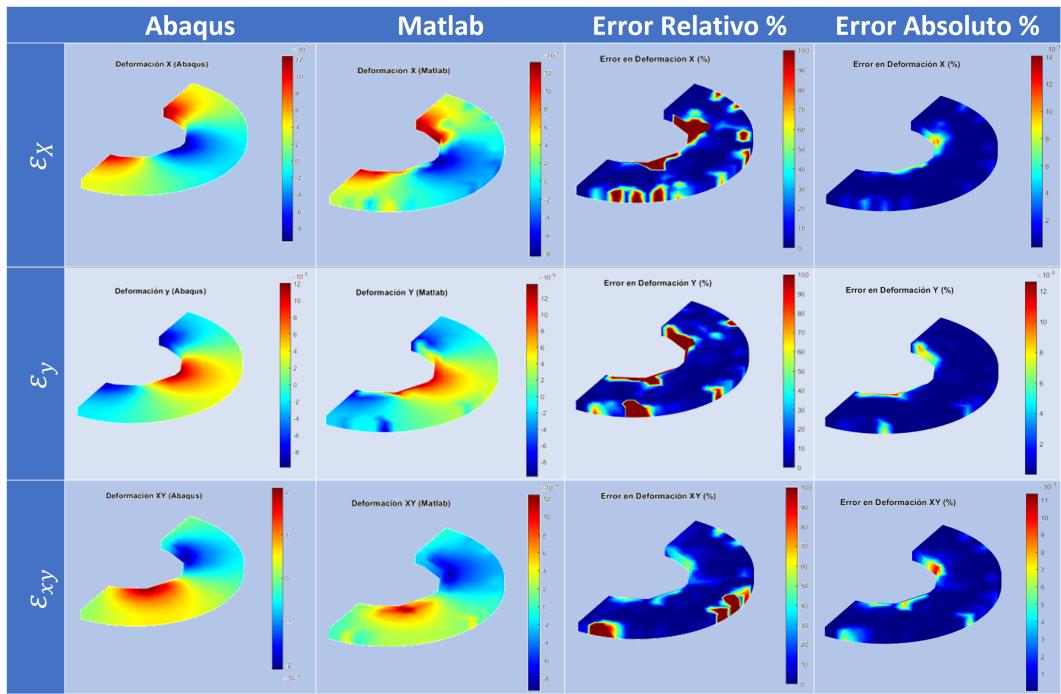


Figura 6.23: Comparación entre deformaciones cartesianas de Abaqus y Matlab.

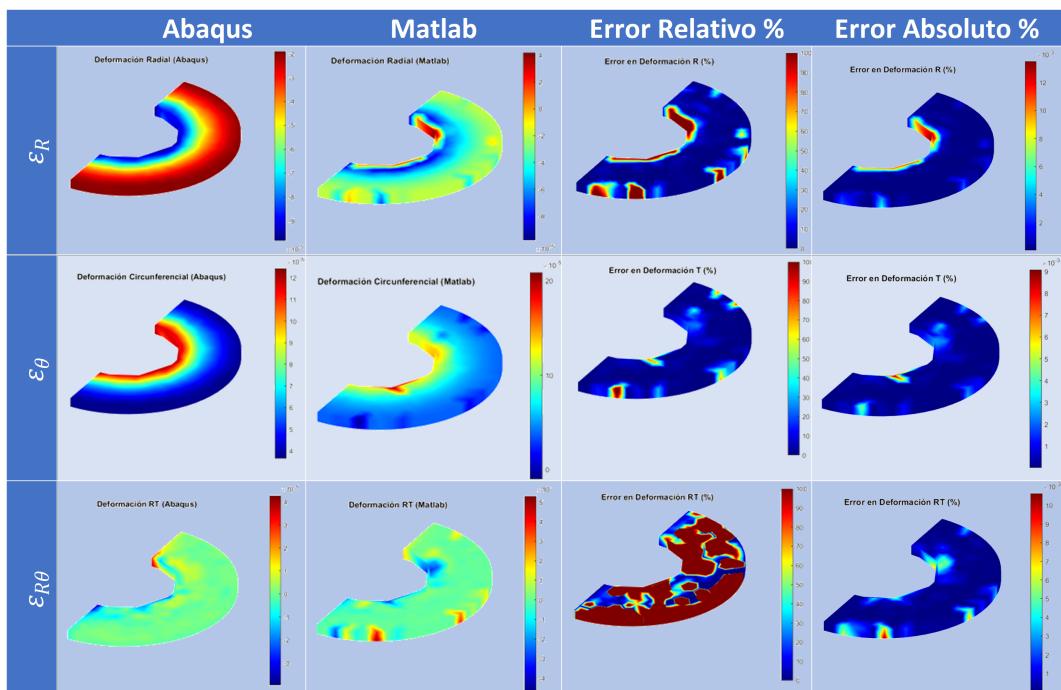


Figura 6.24: Comparación entre deformaciones cilíndricas de Abaqus y Matlab.

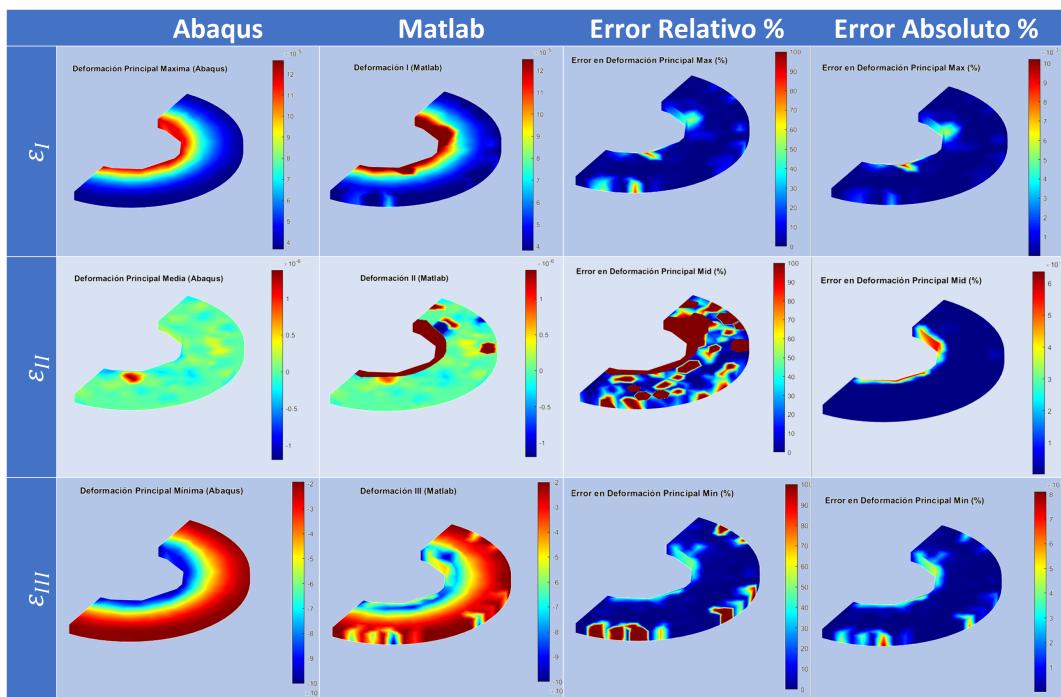


Figura 6.25: Comparación entre deformaciones principales de Abaqus y Matlab.

Anexo IV: Resultados-Cálculo de Variables

En este *Anexo* se representan los resultados del método desarrollado en el *Anexo II* particularizando en la **geometría idealizada con capa fibrótica de 150 micras de espesor**. Se muestran los resultados tanto del análisis 2D, como del análisis 3D.

Resultados del Modelo 2D

El primer proceso es el de crear la malla (*Figura 6.26.a*), procesarla para eliminar los nodos extra (*Figura 6.26.b*). Una vez se tiene la malla final se puede crear el modelo *PDE* de Matlab (*Figura 6.26.c*).

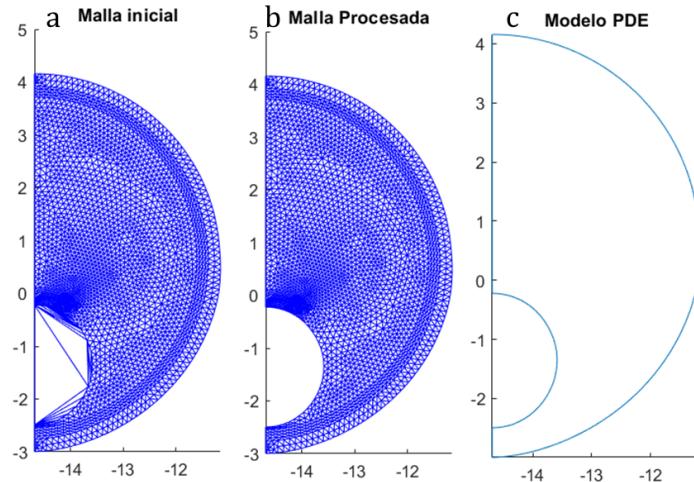


Figura 6.26: Geometría idealizada tratada como modelo 2D, a) Malla inicial, b) Malla procesada y c) Modelo PDE.

El siguiente paso es introducir los desplazamientos (calculados previamente en Abaqus CAE) como solución del modelo *PDE* y poder representar los desplazamientos (*Figura 6.27*), el gradiente de los desplazamientos (*Figura 6.28*) y el módulo del gradiente de los desplazamientos (*Figura 6.29*).

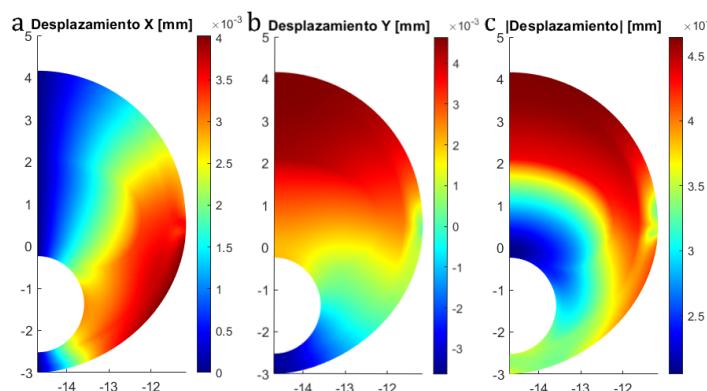


Figura 6.27: Desplazamiento en mm en, a) el eje X, b) el eje Y, y c) módulo de los desplazamientos.

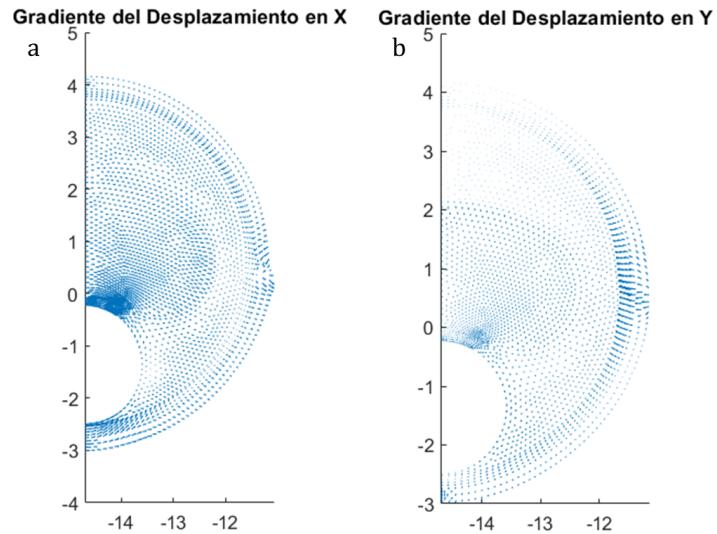


Figura 6.28: Gradiente (vector) de los desplazamientos en a) el eje X, b) el eje Y.

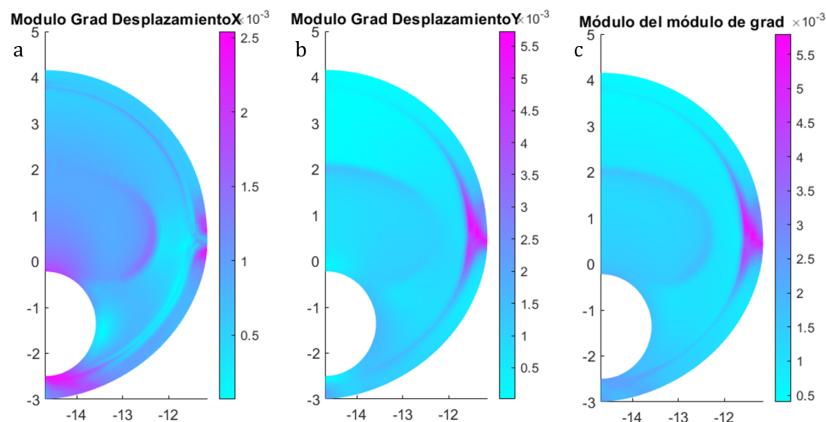


Figura 6.29: Módulo del gradiente de los desplazamientos, a) eje X, b) eje Y, c) Módulo del desplazamiento en X e Y.

Seguidamente, tal y como ya se ha explicado se calculan las deformaciones cartesianas. En la *Figura 6.30* se muestra el resultado de dicho cálculo, en la *Figura 6.31* se representan las deformaciones obtenidas en Abaqus CAE (tras restar las deformaciones logarítmicas de cada *step*) para que sirva de comparación. Al comparar la deformación tangencial se observa que los valores de Abaqus son el doble de los obtenidos en el método desarrollado en Matlab. Esto se debe a que en Abaqus no se representa la deformación tangencial como tal, se representa (γ_{xy}) que se define en la *Ecuación 6.3*.

$$\gamma_{xy} = \varepsilon_{xy} + \varepsilon_{yx} = 2 \cdot \varepsilon_{xy} \quad (6.3)$$

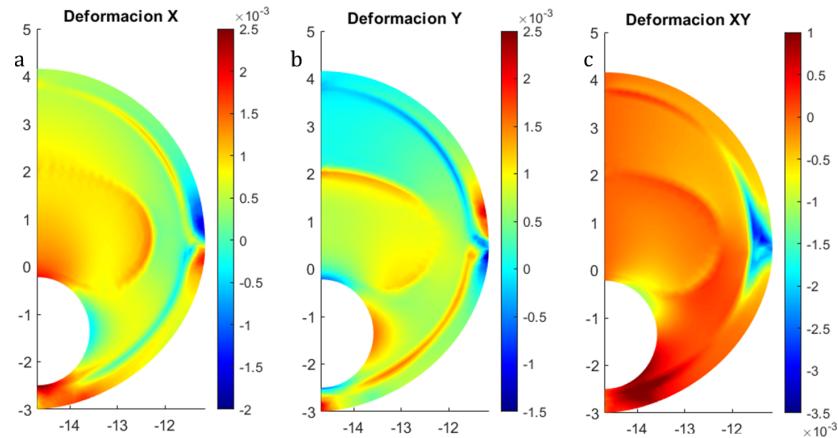


Figura 6.30: Deformaciones cartesianas calculadas con el método sobre la geometría de espesor de 150 μm bajo la hipótesis de deformación plana. a) ε_x , b) ε_y , c) ε_{xy} .

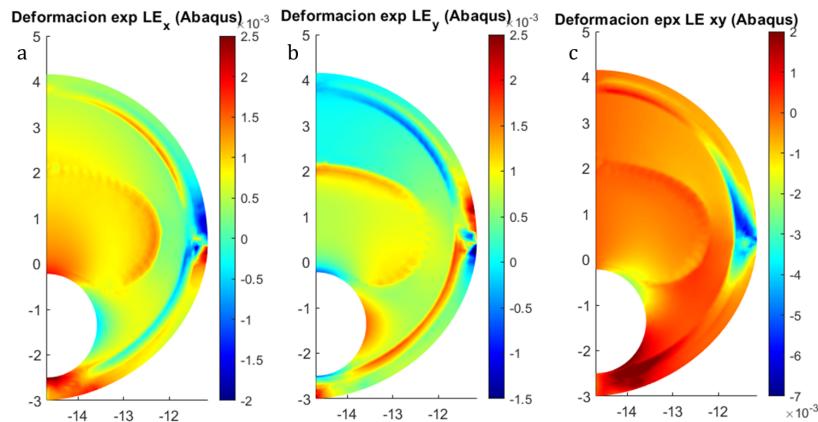


Figura 6.31: Deformaciones obtenidas de Abaqus CAE en la geometría de espesor de 150 μm . a) ε_x , b) ε_y , c) ε_{xy} .

Con el fin de realizar una comparación cuantitativa se representa en la *Figura 6.32* el error relativo (%) entre las imágenes anteriores. El error absoluto no se representa en ningún caso porque siempre se obtienen valores muy inferiores al 0 %, ya que el valor de la deformación entre los *steps* de interés es muy pequeño. El siguiente paso es obtener el módulo de los gradientes de las deformaciones calculadas (*Figura 6.33*).

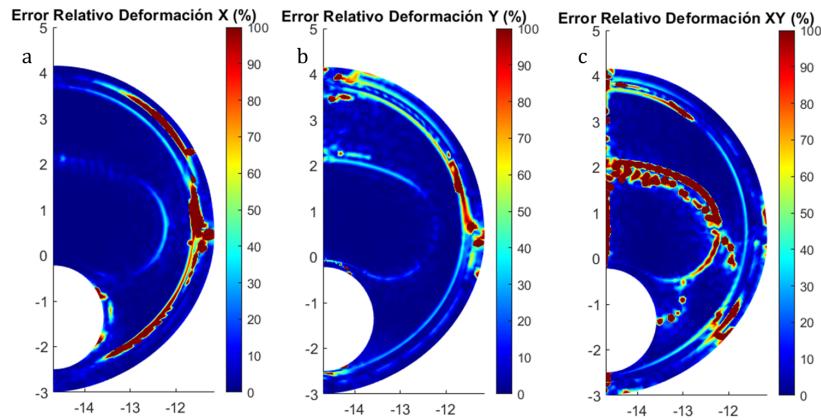


Figura 6.32: Error relativo (%) entre las deformaciones calculadas en Matlab y las obtenidas en Abaqus CAE. Geometría de espesor de 150 μm . a) Error relativo de ε_x , b) Error relativo de ε_y , c) Error relativo de ε_{xy} .

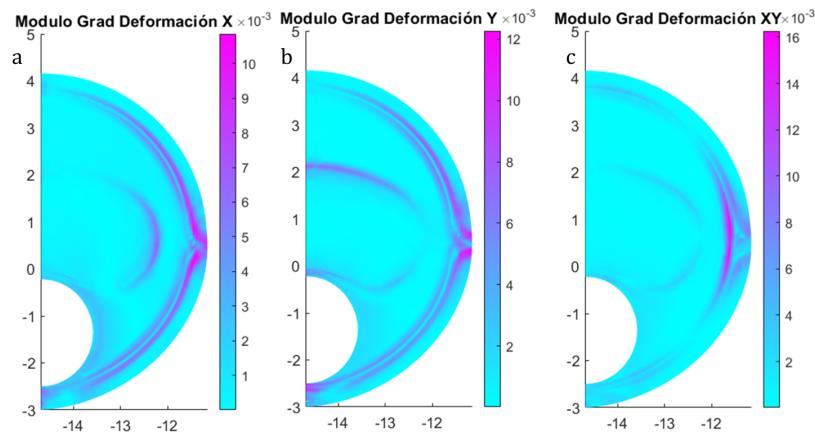


Figura 6.33: Módulo del gradiente de las deformaciones cartesianas. Geometría de espesor de 150 μm . a) $|\nabla \varepsilon_x|$, b) $M |\nabla \varepsilon_y|$, c) $|\nabla \varepsilon_{xy}|$.

Conocidas las deformaciones cartesianas es posible obtener las deformaciones principales. En la Figura 6.34 se muestran las deformaciones principales obtenidas en el método desarrollado y en la Figura 6.35 se representan las deformaciones principales obtenidas en el cálculo de Abaqus CAE. Finalmente se obtiene, en la Figura 6.36, el error relativo entre ambos cálculos. Además, en la Figura 6.37 se muestra el módulo del gradiente de la deformación principal.

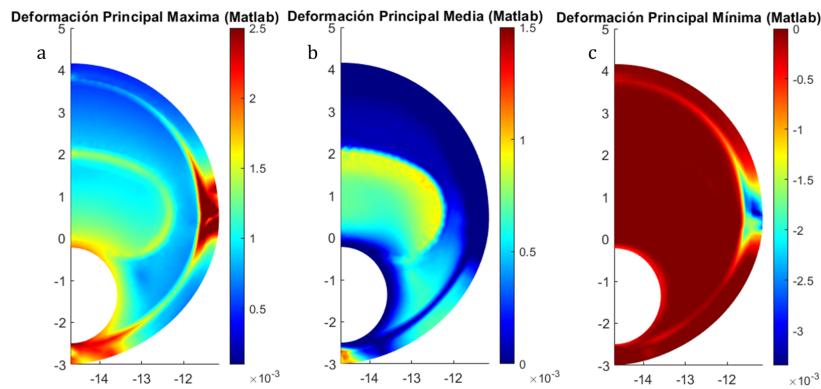


Figura 6.34: Deformaciones principales calculadas con el método sobre la geometría de espesor de 150 μm bajo la hipótesis de deformación plana. a) ε_I , b) ε_{II} , c) ε_{III} .

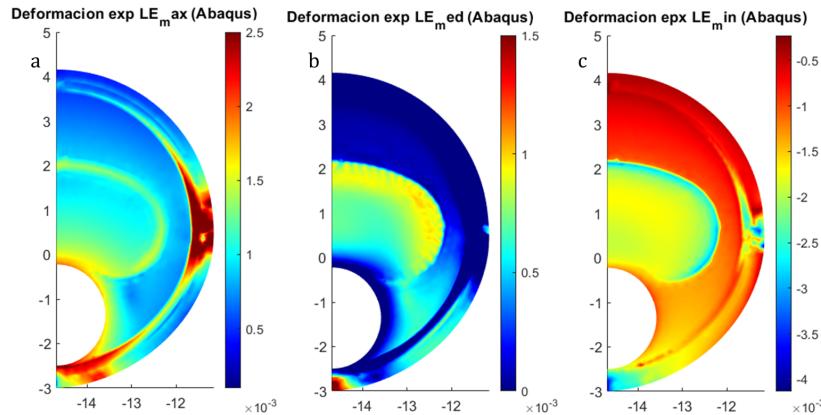


Figura 6.35: Deformaciones principales obtenidas de Abaqus CAE en la geometría de espesor de 150 μm . a) ε_I , b) ε_{II} , c) ε_{III} .

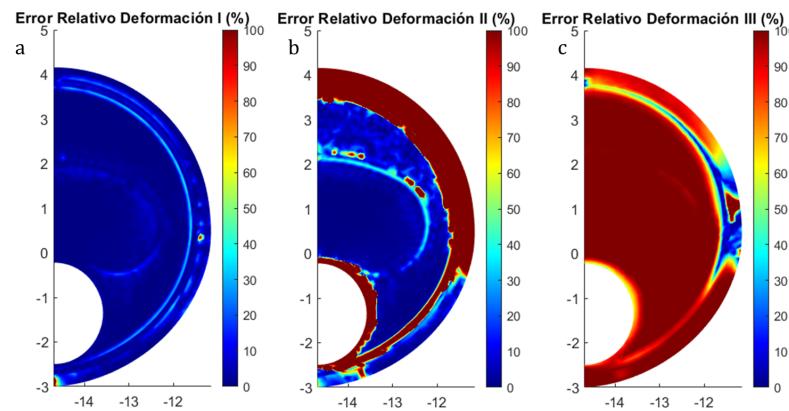


Figura 6.36: Error relativo (%) entre las deformaciones principales calculadas en Matlab y las obtenidas en Abaqus CAE. Geometría de espesor de 150 μm . a) Error relativo de ε_I , b) Error relativo de ε_{II} , c) Error relativo de ε_{III} .

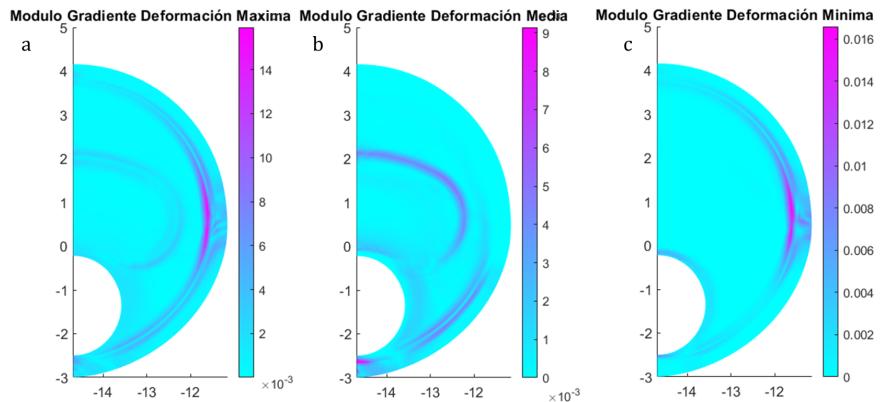


Figura 6.37: Módulo del gradiente de las deformaciones principales. Geometría de espesor de 150 μm . a) $|\nabla\varepsilon_I|$, b) $|\nabla\varepsilon_{II}|$, c) $|\nabla\varepsilon_{III}|$.

Tras calcular las deformaciones principales es posible obtener deformaciones equivalentes como la de von Mises y Tresca y calcular el índice de anisotropía (Figura 6.38). Seguidamente se obtienen sus gradientes (Figura 6.39).

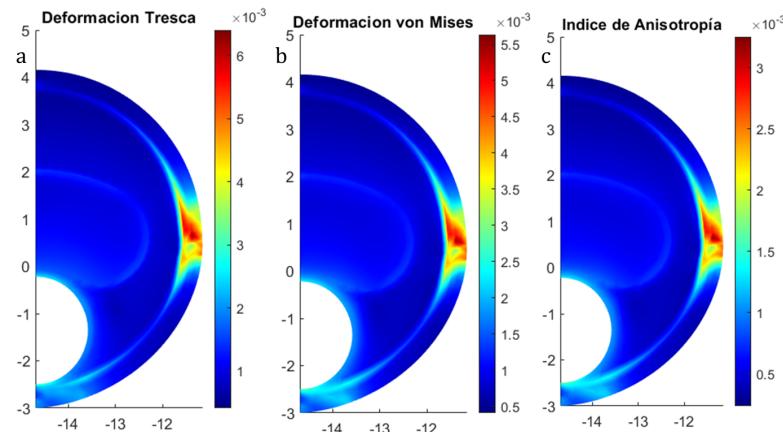


Figura 6.38: Deformaciones equivalentes calculadas con el método sobre la geometría de espesor de 150 μm bajo la hipótesis de deformación plana. a) $\varepsilon_{\text{vonMies}}$, b) $\varepsilon_{\text{Tresca}}$, c) Índice de Anisotropía (FA).

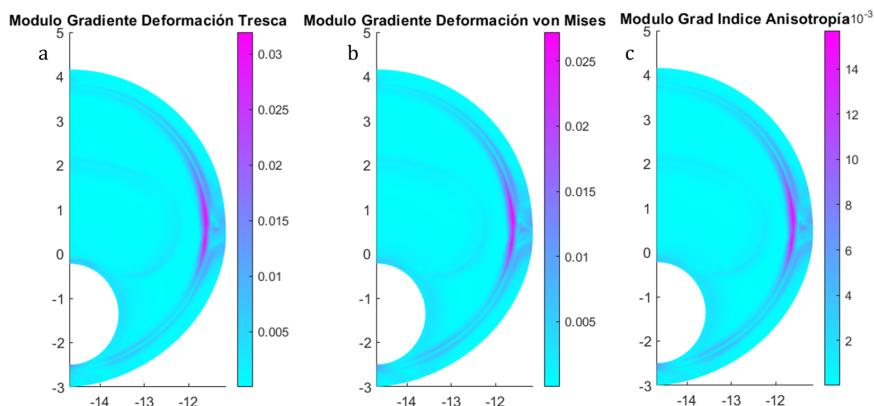


Figura 6.39: Módulo del gradiente de las deformaciones equivalentes. Geometría de espesor de 150 μm . a) $M|\nabla\varepsilon_{\text{vonMises}}|$, b) $|\nabla\varepsilon_{\text{Tresca}}|$, c) $|\nabla FA|$.

Seguidamente se calculan las deformaciones en coordenadas cilíndricas mediante la matriz cambio de base (Ecuación 3.20) representadas en la Figura 6.40. Del mismo modo se obtiene el módulo del gradiente de dichas deformaciones (Figura 6.41).

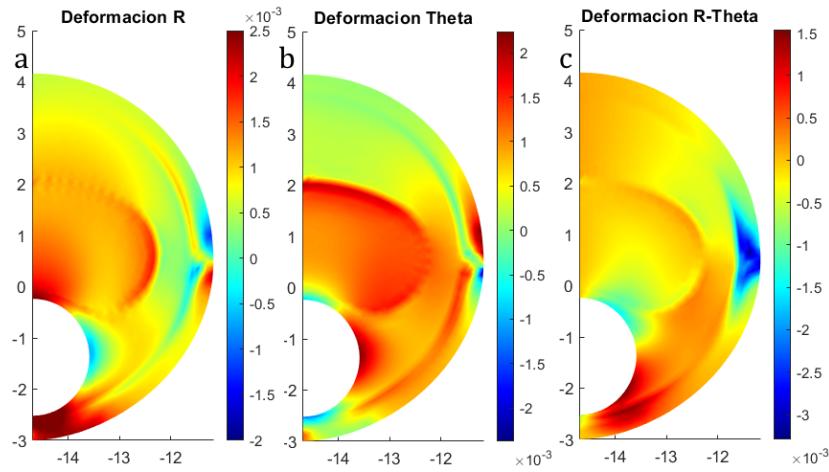


Figura 6.40: Deformaciones en coordenadas cilíndricas calculadas con el método sobre la geometría de espesor de 150 μm bajo la hipótesis de deformación plana. a) ε_R , b) ε_θ , c) $\varepsilon_{R\theta}$.

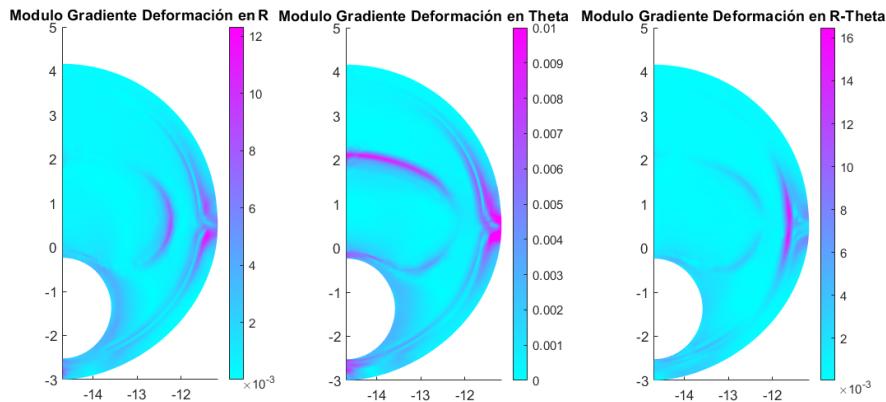


Figura 6.41: Módulo del gradiente de las deformaciones en coordenadas cilíndricas. Geometría de espesor de 150 μm . a) $|\nabla \varepsilon_R|$, b) $|\nabla \varepsilon_\theta|$, c) $|\nabla \varepsilon_{R\theta}|$.

La última variable a calcular y representar es dW (Figura 6.42.a) y el módulo de su gradiente (Figura 6.42.b).

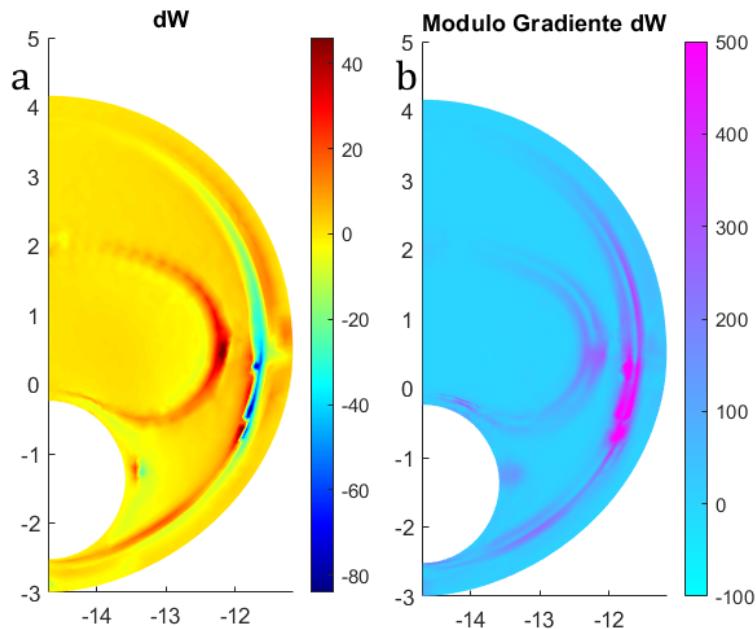


Figura 6.42: a) Representación de la variable dW , b) Representación de $|\nabla dW|$.

Resultados del Modelo 3D

En el modelo 3D se realizan los mismos cálculos y representaciones pero considerando todos los desplazamientos. Al incrementar el número de nodos y el número de operaciones dentro de cada variable el tiempo de cálculo aumenta considerablemente. Al tratar el modelo 2D el tiempo de cálculo de todas las variables es aproximadamente de 5 horas, mientras que en este caso el tiempo asciende a más de 145 horas.

En la *Figura 6.43* se muestran el proceso de creación de una malla inicial, procesado de la malla y creación del modelo *PDE*.

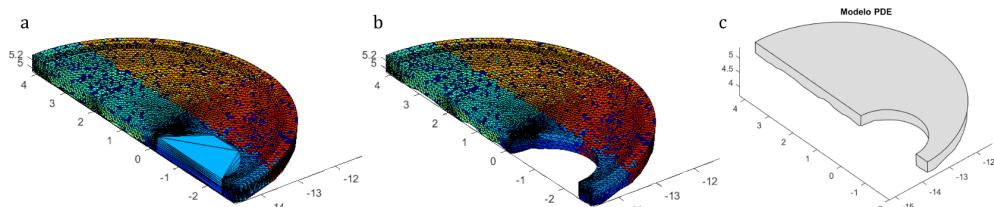


Figura 6.43: Geometría idealizada tratada como modelo 3D, a) Malla inicial, b) Malla procesada y c) Modelo PDE.

Se representan los desplazamientos *Figura 6.44*, todas las imágenes están tomadas en el plano XY para poder realizar una comparación con el método 2D. En la *Figura 6.45* se muestra el gradiente de los desplazamientos y en la *Figura 6.46* el módulo del gradiente de los desplazamientos.

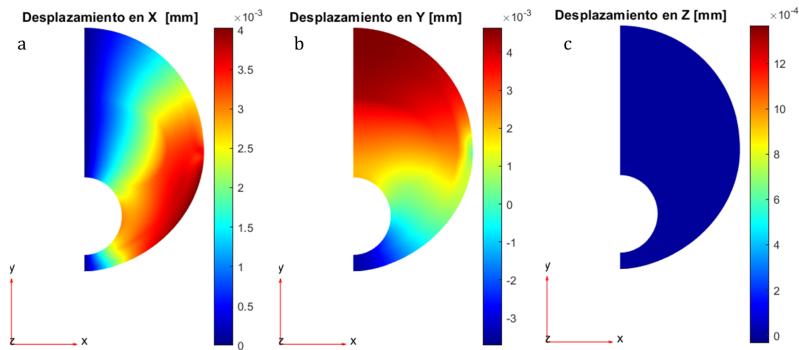


Figura 6.44: Desplazamiento en mm en, a) u_x , b) u_y , c) u_z .

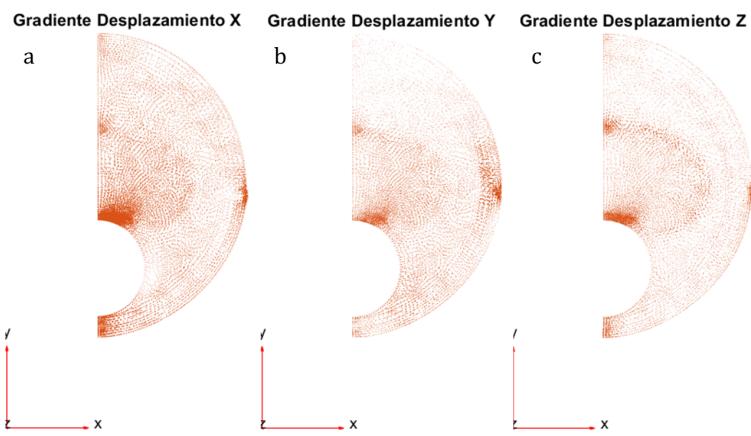


Figura 6.45: Gradiente de los desplazamiento en mm, a) ∇u_x , b) ∇u_y , x) ∇u_z .

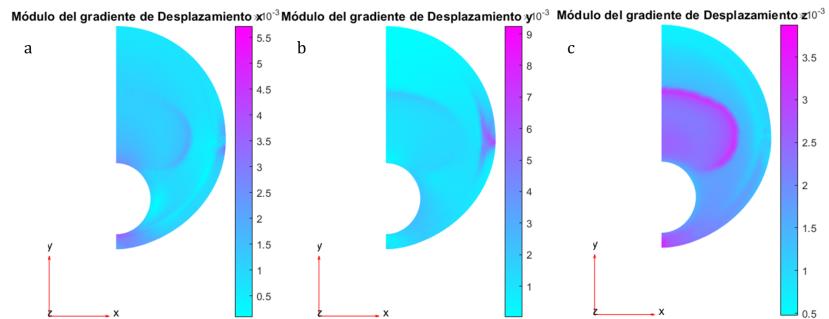


Figura 6.46: Módulo del gradiente de los desplazamiento en mm, a) $|\nabla u_x|$, b) $|\nabla u_y|$, x) $|\nabla u_z|$.

Seguidamente se calculan las deformaciones en coordenadas cartesianas (Figura 6.47), el error relativo con las deformaciones obtenidas en Abaqus (Figura 6.48) y el módulo del gradiente de las deformaciones cartesianas (Figura 6.49).

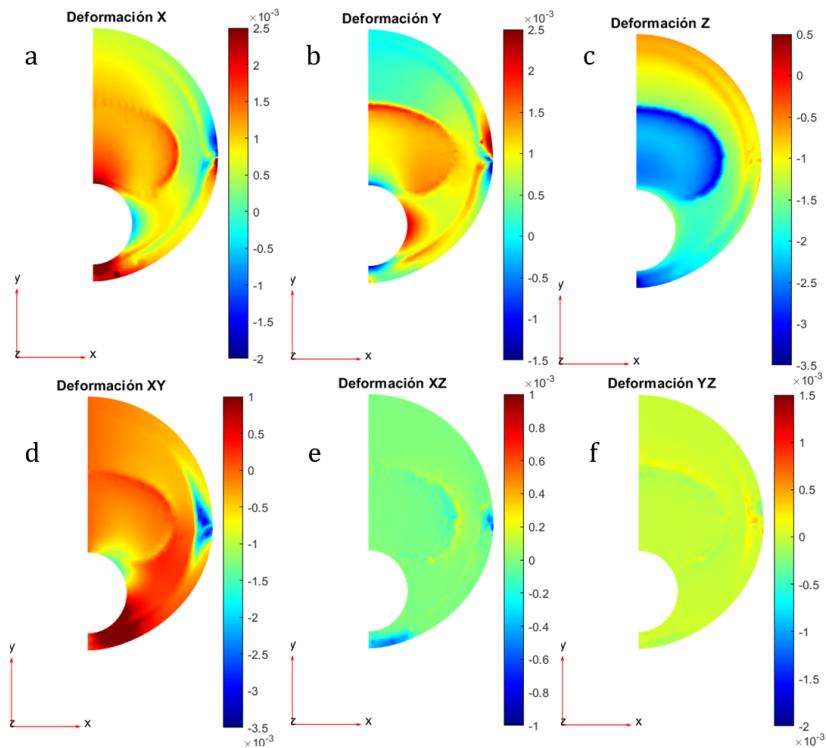


Figura 6.47: Deformaciones en coordenadas cartesianas, a) ε_x , b) ε_y , c) ε_z , d) ε_{xy} , e) ε_{xz} , f) ε_{yz} .

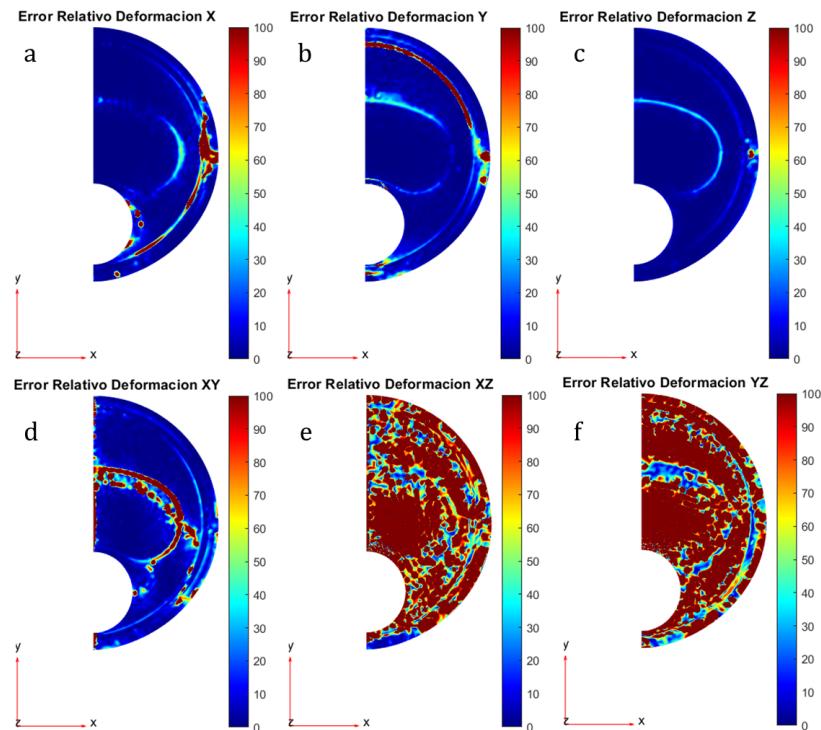


Figura 6.48: Error relativo entre las deformaciones en coordenadas cartesianas calculadas en Matlab y las deformaciones obtenidas en Abaqus CAE, a) Error Relativo en ε_x , b) Error Relativo en ε_y , c) Error Relativo en ε_z , d) Error Relativo en ε_{xy} , e) Error Relativo en ε_{xz} , f) Error Relativo en ε_{yz} .

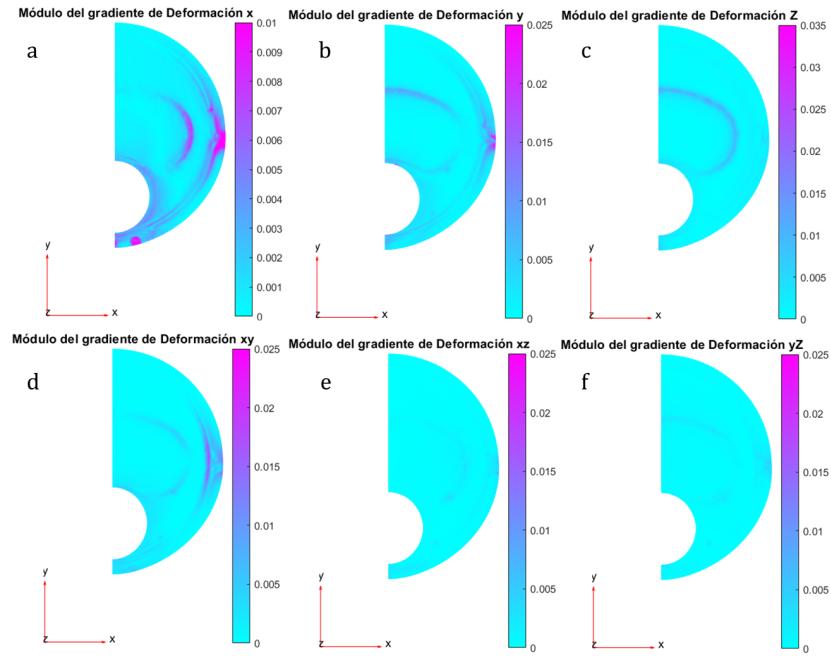


Figura 6.49: Módulo del gradiente de la deformación en coordenadas cartesianas, a) $|\nabla \varepsilon_x|$, b) $|\nabla \varepsilon_y|$, c) $|\nabla \varepsilon_z|$, d) $|\nabla \varepsilon_{xy}|$, e) $|\nabla \varepsilon_{xz}|$, f) $|\nabla \varepsilon_{yz}|$.

A continuación se procesan las deformaciones principales (*Figura 6.50*), el error relativo con las deformaciones principales de Abaqus (*Figura 6.51*) y el módulo del gradiente de la deformación principal (*Figura 6.52*).

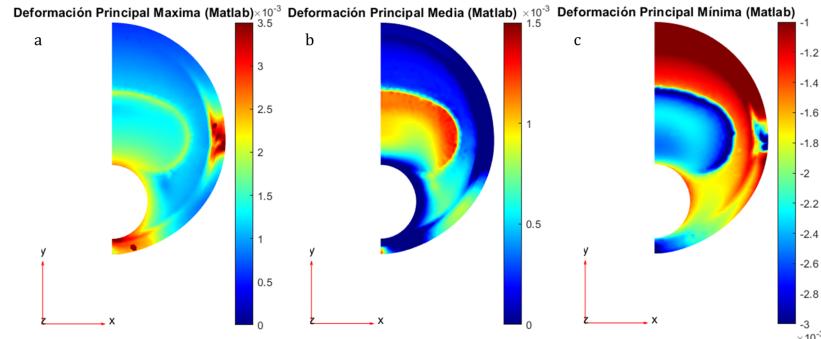


Figura 6.50: Deformaciones principales, a) ε_I , b) ε_{II} , c) ε_{III} .

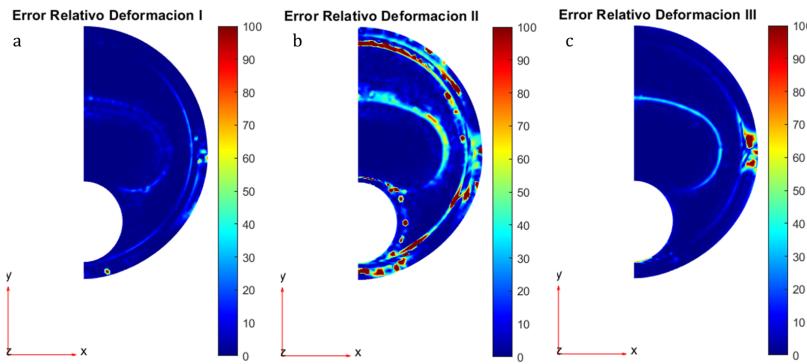


Figura 6.51: Error relativo entre las deformaciones principales calculadas en Matlab y las deformaciones obtenidas en Abaqus CAE, a) Error Relativo en ε_I , b) Error Relativo en ε_{II} , c) Error Relativo en ε_{III} .

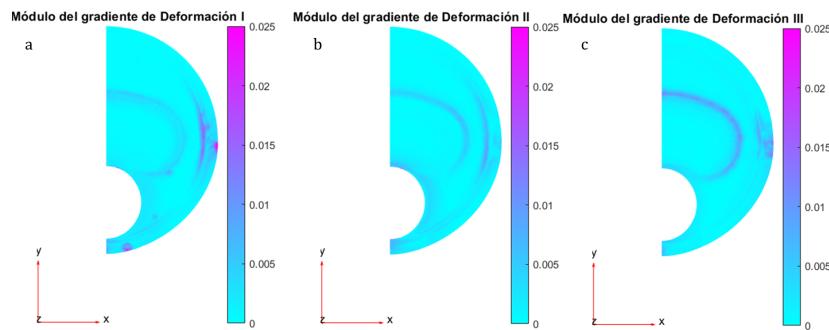


Figura 6.52: Módulo del gradiente de la deformación principal, a) $|\nabla \varepsilon_I|$, b) $|\nabla \varepsilon_{II}|$, c) $|\nabla \varepsilon_{III}|$.

Al igual que en el caso 2D, se calculan las deformaciones en coordenadas cilíndricas (Figura 6.53) y el módulo de su gradiente (Figura 6.54).

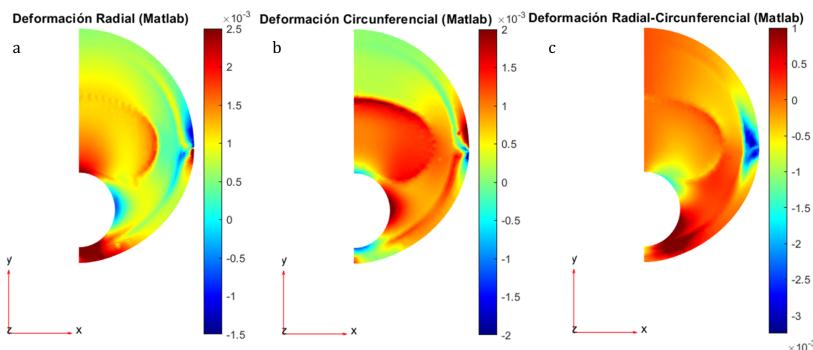


Figura 6.53: Deformaciones en coordenadas cilíndricas, a) ε_R , b) ε_θ , c) $\varepsilon_{R\theta}$.

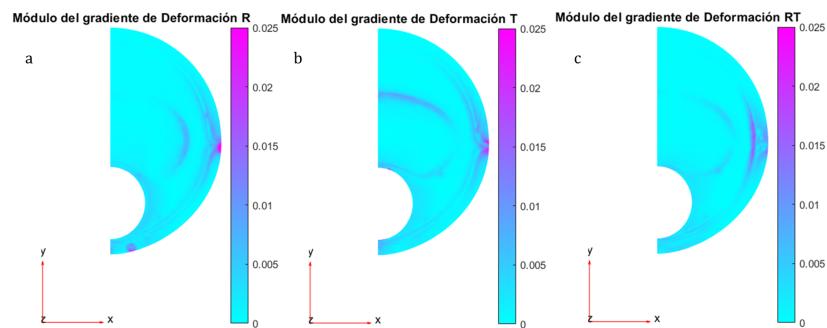


Figura 6.54: Módulo del gradiente de la deformación principal, a) $|\nabla \varepsilon_R|$, b) $|\nabla \varepsilon_\theta|$, c) $|\nabla \varepsilon_{R\theta}|$.

El último paso es calcular y representar la variable dW , tal y como se observa en la Figura 6.55.

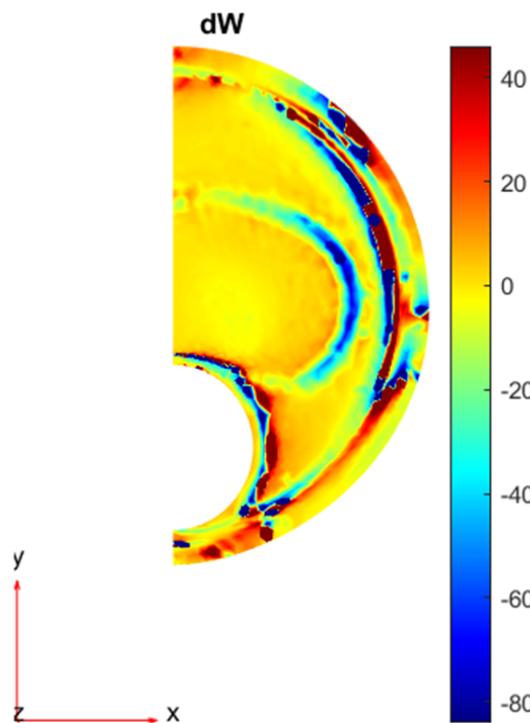


Figura 6.55: Representación de la variable dW .

Anexo V: Proceso de Segmentado

En este apartado se profundiza en el proceso de segmentado a partir de las variables calculadas (ver *Anexo IV*). Algunas de las partes del proceso se han explicado en el punto 3.4 de la memoria y se repiten en este *Anexo* con el fin de ampliar las explicaciones y posibilitar un mejor entendimiento del proceso. También se expone el código desarrollado, el ajuste de la escala para medir áreas y longitudes etc. El código es válido tanto para todas las geometrías analíticas, tanto tratadas como 2D como 3D. En el caso de la geometría de IVUS real algunos pasos no son necesarios o es necesarios modificarlos. El código se ha desarrollado en Matlab y consta de diferentes partes o apartados:

- **Segmentación real de las partes:** en el primer apartado se obtienen los contornos reales de cada una de las partes del modelo y se calcula su área (en píxeles).
- **Tratamiento de la imagen:** como se ha visto en el *Anexo IV* es posible representar todas las variables que han sido calculadas, sin embargo, se tiene solo la mitad de la sección de interés. Con el fin de simular la sección entera para simular el IVUS, se realizan unas operaciones sencillas para obtener la sección entera.
- **Método Watershed:** se modifica el método de Watershed de Matlab para poder distinguir los contornos de los tejidos de interés y poder sustraerlos.
- **Obtención de las partes:** una vez se obtiene el resultado del *Watershed* se ha de seleccionar el tejido con un elemento específico del resultado. Además se hacen operaciones como el suavizado que permite obtener un contorno del núcleo lipídico más próximo al real mediante comparaciones con el real.
- **Ajuste de Escala:** es necesario tomar una referencia para pasar las medidas de píxeles a unidades de longitud (*mm*).

Segmentación real de las partes

En el código de Matlab de *Lectura de Datos* del *Anexo II* se introduce unas líneas que permiten asignar un número a cada uno de los nodos dependiendo del tejido al que pertenecen. Por lo tanto, ahora es posible seleccionar los nodos dependiendo del tejido que se desee obtener, y con ello se puede representar el contorno real, medir el área... En el siguiente código se observa el proceso para obtener el núcleo lipídico real de la geometría analítica de espesor 150 micras.

```
for i=1:length(X)
    if instante1(i,13)==1
        colores_n(i)=1;
    elseif instante1(i,13)==2
        colores_n(i)=1;
    elseif instante1(i,13)==3
        colores_n(i)=1;
    elseif instante1(i,13)==4
        colores_n(i)=0;
    elseif instante1(i,13)==5
        colores_n(i)=1;
    end
end
figure(41), hold on, set(gcf, 'color', 'w'),
pdeplot(modelo2,'XYData',colores_n,'ColorMap', 'gray'),caxis([0
0.01]),daspect([1,1,1]),xlim([-15,-11]),ylim([-3.5,4.5])
title('Segmentación Real');
saveas(gcf,'Real150mciras.png');
Im_n=imread('Real150mciras.png');figure(1002),imshow(Im_n),
title('Figura Guardada');
```

```
Im2_n=im2double(rgb2gray(imcrop(Im_n,[286 70 240  
480])));figure(1003),imshow(Im2_n),title('Figura Cortada');  
B_n = fliplr(Im2_n);figure(1004),imshow(B_n),title('Figura  
Reflejo'); %Imagen simétrica  
If_n=ones(size(Im2_n,1),size(Im2_n,2)*2);  
for i=1:size(If_n,1)  
    for j=1:size(If_n,2)  
        if j<=size(Im2_n,2)  
            If_n(i,j)=(B_n(i,j));  
        else  
            If_n(i,j)=Im2_n(i,j-size(Im2_n,2));  
        end  
    end  
end  
close all; figure(1005),imshow(If_n);  
nucleo_real=imcomplement(im2bw(If_n));imshow(nucleo_real),  
title('Nucleo Real');  
area_nucleo_real=(regionprops(nucleo_real,'Area'));
```

El resultado es el núcleo real, representado en la *Figura 6.56*. La imagen es de tipo binaria, lo que permite medir el área del objeto blanco mediante la función *regionprops()*. El resultado de dicha medida será el número de píxeles que ocupa el núcleo lipídico.

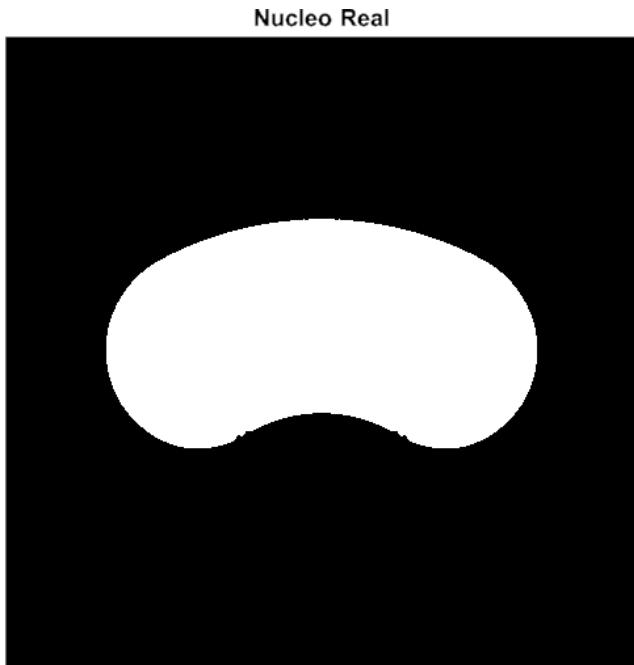


Figura 6.56: Núcleo lipídico real de la geometría idealizada de espesor 150 micras.

Es posible medir el área real ya que se conocen las coordenadas de todos los nodos. Sin embargo, esta medida dependerá de lo fina que sea la malla sobre la geometría en cuestión, ya que para calcular dicha área se suma el área de todos los elementos pertenecientes al tejido. Puede ocurrir que el elemento esté dentro de la geometría (sus tres nodos pertenecen al núcleo lipídico), que solo dos nodos estén dentro o que sólo un nodo se encuentre dentro de la geometría. En la *Figura 6.57* se muestran las tres posibilidades. Para obtener el área se opta por considerar los elementos cuyos nodos pertenecen al tejido y sumar el área de aquellos elementos que poseen únicamente 2 nodos dentro y uno fuera

ponderando su área por un factor de $2/3$, tal y como se resumen en la *Ecuación 6.4*.

$$Area_{Total} = \sum Area_{Elementos3NodosDentro} + \frac{2}{3} \cdot \sum Area_{ElementosSolo2NodosDentro} \quad (6.4)$$

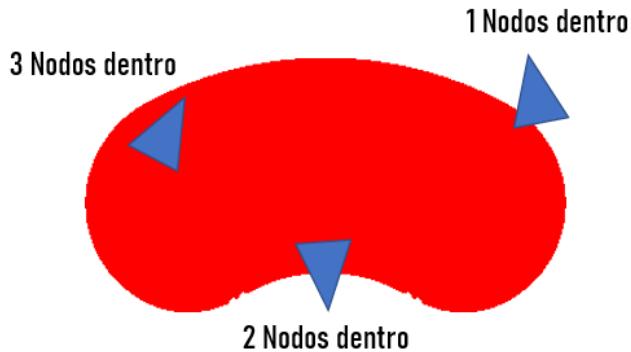


Figura 6.57: Núcleo lipídico donde se muestran las tres posibilidades de los elementos de la malla. Un elemento entero dentro de la geometría, uno con solo dos nodos en el interior y otro con un único nodo dentro.

Se aplica lo explicado en el siguiente código de Matlab, donde el resultado proporcionará el área real en mm^2 . Estas medidas se han utilizado con el fin de comparar más adelante las escalas utilizadas.

```
%% 1.1.2) Núcleo Real en malla: Se calcula el área del núcleo real
% considerando los elementos que lo componen
% Se suma el área de todos los elementos que tienen los tres nodos
% dentro del núcleo. Además se suman (multiplicado por 2/3) los
% elementos que tienen solo 2 nodos dentro del núcleo

%Area 3 nodos dentro del núcleo
for i=1:Elementos
    nodos=model.Mesh.Elements(:,i); %obtengo los nodos de cada
    % elemento
    coord_x=[X(nodos(1)) X(nodos(2)) X(nodos(3))];
    coord_y=[Y(nodos(1)) Y(nodos(2)) Y(nodos(3))];
    if instante1(nodos(1),13)==4 && instante1(nodos(2),13)==4 &&
    instante1(nodos(3),13)==4
        %Calcular area del elemento
        Area_Elemento_123(i)=polyarea(coord_x,coord_y);
    else
        Area_Elemento_123(i)=0;
    end
end

Area_nodos_123_nucleo=sum(Area_Elemento_123)*2;

%Area nodos 1 y 2 dentro del núcleo
for i=1:Elementos
    nodos=model.Mesh.Elements(:,i); %obtengo los nodos de cada
    % elemento
    coord_x=[X(nodos(1)) X(nodos(2)) X(nodos(3))];
    coord_y=[Y(nodos(1)) Y(nodos(2)) Y(nodos(3))];
    if instante1(nodos(1),13)==4 && instante1(nodos(2),13)==4
        %Calcular area del elemento

```

```
Area_Elemento_12(i)=polyarea(coord_x,coord_y);
else
    Area_Elemento_12(i)=0;
end
end

Area_nodos_12_nucleo=sum(Area_Elemento_12)*2;

%Area nodos 1 y 3 dentro del núcleo
for i=1:Elementos
    nodos=model.Mesh.Elements(:,i); %obtengo los nodos de cada
    % elemento
    coord_x=[X(nodos(1)) X(nodos(2)) X(nodos(3))];
    coord_y=[Y(nodos(1)) Y(nodos(2)) Y(nodos(3))];
    if instante1(nodos(1),13)==4 && instante1(nodos(3),13)==4
        %Calcular area del elemento
        Area_Elemento_13(i)=polyarea(coord_x,coord_y);
    else
        Area_Elemento_13(i)=0;
    end
end

Area_nodos_13_nucleo=sum(Area_Elemento_13)*2;

%Area nodos 2 y 3 dentro del núcleo
for i=1:Elementos
    nodos=model.Mesh.Elements(:,i); %obtengo los nodos de cada
    % elemento
    coord_x=[X(nodos(1)) X(nodos(2)) X(nodos(3))];
    coord_y=[Y(nodos(1)) Y(nodos(2)) Y(nodos(3))];
    if instante1(nodos(2),13)==4 && instante1(nodos(3),13)==4
        %Calcular area del elemento
        Area_Elemento_23(i)=polyarea(coord_x,coord_y);
    else
        Area_Elemento_23(i)=0;
    end
end

Area_nodos_23_nucleo=sum(Area_Elemento_23)*2;
% Se calcula el área total pero los elementos que solo tienen 2 nodos
% en el núcleo se multiplica su área por (2/3)
Area_Matlab_nucleo=Area_nodos_123_nucleo+
((Area_nodos_12_nucleo-Area_nodos_123_nucleo)+
(Area_nodos_13_nucleo-Area_nodos_123_nucleo)+
(Area_nodos_23_nucleo-Area_nodos_123_nucleo))*(2/3);
```

Se repite el mismo proceso para cada uno de los tejidos: placa y tejido sano (adventicia + media). El resultado es el obtenido en la *Figura 6.58*.

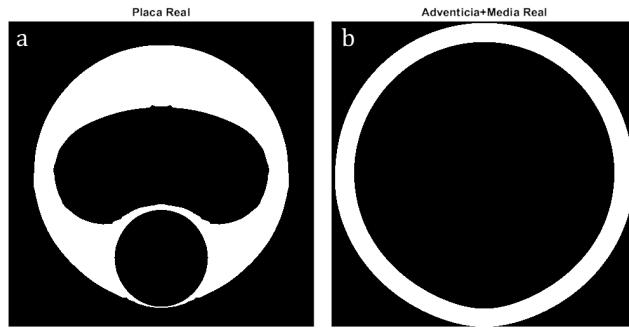


Figura 6.58: a) Segmentación real de la placa o capa fibrótica, b) Segmentación real del tejido sano (Adventicia y Media).

Tratamiento de la Imagen

Las simulaciones se han realizado con la mitad de la geometría, pero a la hora de segmentar y analizar es necesario obtener la sección entera. Este paso se ha realizado también en el apartado anterior para conseguir las geometrías reales y es necesario repetirlo a la hora de representar cualquiera de las variables calculadas. El proceso se basa en representar una variable, recortar la imagen para eliminar los ejes, crear la simétrica y unirlas. En la *Figura 3.5* de la memoria se muestra el proceso.

En este paso es necesario también fijar los límites de la representación, escala de colores a utilizar y el tamaño de la imagen. Los pasos de crear la imagen simétrica no es necesario cuando se considera la geometría de IVUS, ya que se está trabajando con la sección entera desde el primer momento.

```
%% 2) TRATAMIENTO DE LA IMAGEN PARA OBTENER LA SEGMENTACIÓN %
%% 2.1) Tratamiento de la variable
figure(1000), hold on, set(gcf,'color','w'),
pdeplot(modelo2,'XYData',dW_Total,'ColorMap','Gray'),daspect([1,1,1]),
caxis([-80,44]), xlim([-15,-11]), ylim([-3.5,4.5]),%, xlim([-15,-11]),
ylim([-3.5,4.5]) 150 micras
title('Modulo Gradiente dW')
saveas(gcf,'ImageProcesar.png');
Im=imread('ImageProcesar.png');figure(1002),imshow(Im),title('Figura
Guardada');
Im2=im2double(rgb2gray(imcrop(Im,[286 70 240
480])));figure(1003),imshow(Im2),title('Figura Cortada');
B = fliplr(Im2);figure(1004),imshow(B),title('Figura Reflejo');
%Imagen simétrica
If=ones(size(Im2,1),size(Im2,2)*2);
for i=1:size(If,1)
    for j=1:size(If,2)
        if j<=size(Im2,2)
            If(i,j)=(B(i,j));
        else
            If(i,j)=Im2(i,j-size(Im2,2));
        end
    end
end
close all;figure(1005),imshow(If),title('Figura Final: Sección
Entera');
```

Método *Watershed*

Una vez se han seleccionado las variables a combinar (*Figura 6.59 a y b*) y se ha realizado el paso anterior se pasa al método *Watershed*. En este caso se combinan la variable *dW* (*Figura 6.59.a*) y el módulo del gradiente de la deformación *Y* (*Figura 6.59.b*). Para poder explicar con mayor claridad el proceso se busca que el segmentado no sea perfecto, por ello se han elegido una combinación de variables y unos parámetros que consigan un segmentado con error. El primer paso es unir ambas imágenes (*Figura 6.59.c*), se ajustan los colores para permitir un mayor contraste (*Figura 6.59 d*) y se calcula el gradiente de la imagen (*Figura 6.59.e*). El siguiente paso es utilizar el denominado “*disco de segmentado*” que permite difuminar los bordes y hacer más notorios los cambios entre tejidos (*Figura 6.59.f*). Seguidamente se cierra de nuevo la geometría (*Figura 6.59.g*) y se realiza el proceso de segmentado con el gradiente obtenido y la imagen procesada, lo que permite encontrar las diferentes zonas marcadas en la representación de las variables elegidas (*Figura 6.59.h*). El resultado de todo el proceso es una serie de secciones (*Figura 6.59.i*) donde se puede observar que en amarillo se segmenta el núcleo lipídico y en azul oscuro el lumen. Para obtener una segmentación más precisa habría sido necesario modificar los límites de representación de cada variable y el “*disco de segmentado*”.

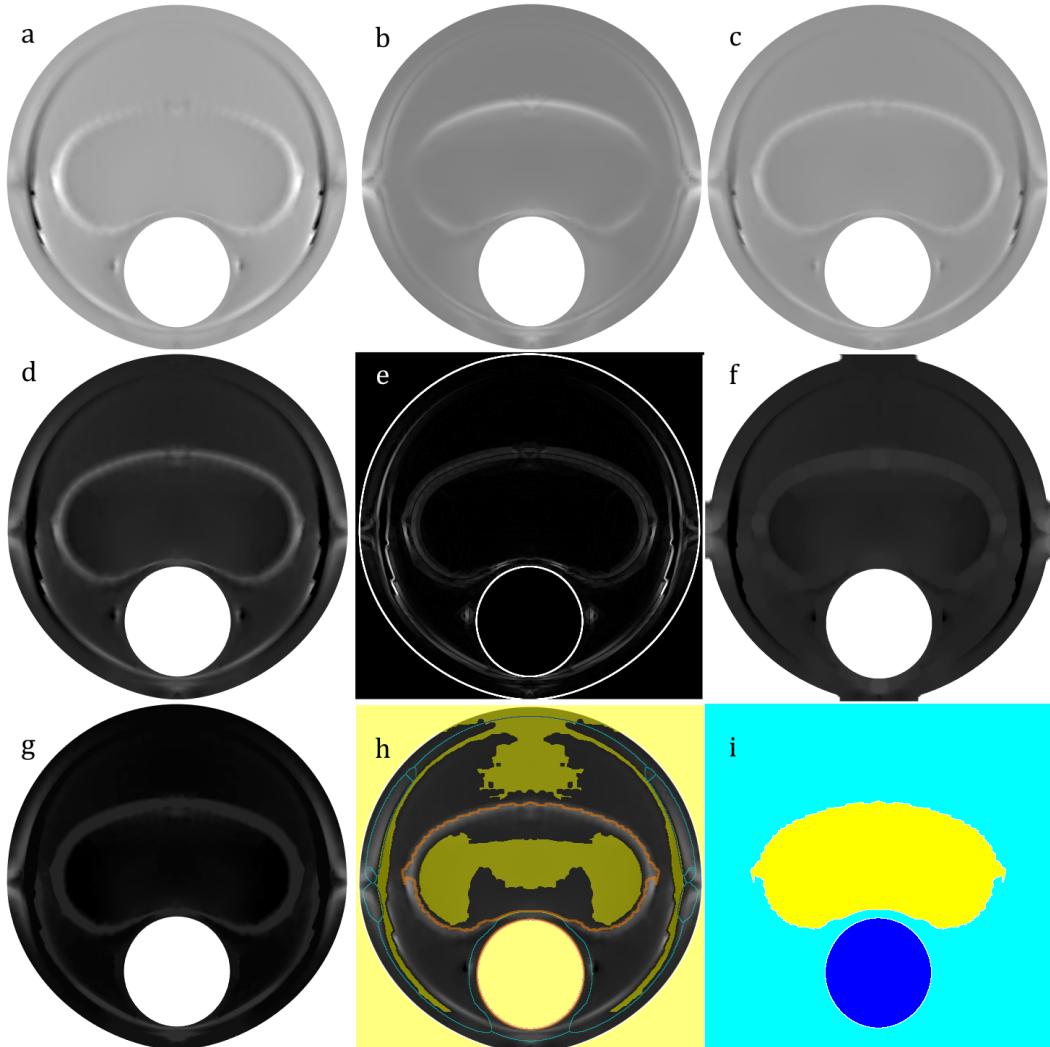


Figura 6.59: Proceso de segmentado *Watershed*, a) Variable 1: *dW*, b) Variable 2: $|\nabla \varepsilon_y|$, c) Unión de ambas variables, d) Ajuste saturado de los colores, e) Gradiente de *d*, f) Implementación del *disco de segmentado*, g) Cierre de la geometría, h) Aplicación del *Watershed* e i) Resultado final de la segmentación.

%% 2.4) Unión de las imágenes anteriores y segmentado *Watershed* de Nucleo y

% Lumen

```
imagen1 = If;
imagen2= If22;
I=imagen1*0.5+imagen2*0.5;figure(3),imshow(I), set(gcf,'color','w'),
title('Suma'); C=imresize(I,1);imshow(I);
I1=imadjust(C);figure(5),imshow(I1),set(gcf,'color','w'),title('Ajustada');
gmag = (imggradient(I1,'prewitt')); imshow(gmag), title('Gradient
Magnitude');
%Cambio los colores para hacer que el núcleo pase a un primer
%plano
I=(I1); imshow(I),set(gcf,'color','w'),title('Complementaria');
%Resaltar el primer plano
se = strel('disk',20); Io = imopen(I,se); imshow(Io);
%Apertura
Ie = imerode(I,se); Iobr = imreconstruct(Ie,I); imshow(Iobr);
%Cierre
Ioc = imclose(Io,se); imshow(Ioc);
%Apertura+Reconstruccion
Iobrd = imdilate(Iobr,se); Iobrcbr =
imreconstruct(imcomplement(Iobrd),imcomplement(Iobr)); Iobrcbr =
imcomplement(Iobrcbr); imshow(Iobrcbr);
Iobrcbr= imadjust(Iobrcbr); imshow(Iobrcbr);
%Quedarme con regiones
fgm = imregionalmax(Iobrcbr)+imregionalmin(Iobrcbr); imshow(fgm);
I2 = labeloverlay(I,fgm); imshow(I2);
%Mejorar las zonas selecciones
se2 = strel(ones(3,3)); fgm2 = imclose(fgm,se2); fgm3 =
imerode(fgm2,se2);
fgm4 = bwareaopen(fgm3,2); I3 = labeloverlay(I,fgm4); imshow(I3);
%Calcular Marcadores del fondo
bw = imbinarize(imcomplement(Iobrcbr),'adaptive'); imshow(bw)
%Esqueleto de zonas de influencia
D = bwdist(~bw); DL = watershed(D); bgm = DL == 0; imshow(bgm);
%Transformacion de cuenca hidrográfica
gmag2 = imimposemin(gmag, bgm | fgm4); L = watershed(gmag2);
%Visualizar resultados
labels = imdilate(L==0,ones(3,3)) + 2*bgm + 3*fgm4; I4 =
labeloverlay(I,labels); imshow(I4);
Lrgb = label2rgb(L,'jet','w','shuffle'); imshow(Lrgb)
figure, imshow(I), hold on, himage = imshow(Lrgb);
himage.AlphaData = 0.2;
```

Obtención de las partes

Tras obtener el resultado de la segmentación (*Figura 6.59.i*) hay que indicar qué es el núcleo lipídico y qué el lumen. En la *Figura 6.60.a* se muestra el núcleo lipídico recién segmentado, y en la *Figura 6.60.b* el lumen.

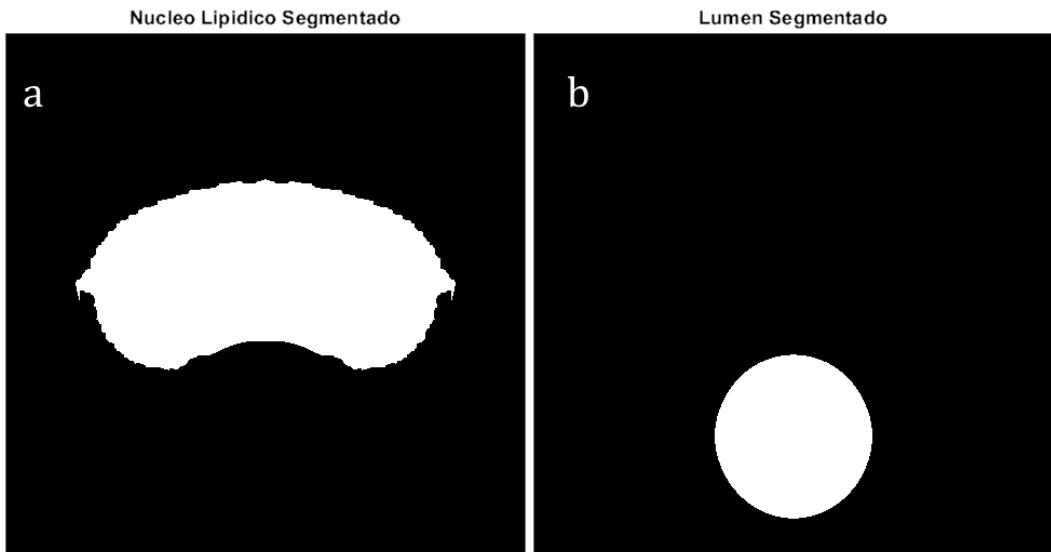


Figura 6.60: Asignación de las partes segmentadas, a) Núcleo lipídico y b) Lumen.

```
nucleo=(L==2); figure(1),imshow(nucleo),title('Nucleo Lipidico Segmentado');
lumen=(L==3);figure(2),imshow(lumen),title('Lumen Segmentado');
```

En el primer paso se ha segmentado el núcleo real, por lo que se puede realizar una comparación superponiendo el segmentado y el real. En la *Figura 6.61.a* se muestra la superposición de ambos, donde en blanco se representa el área que pertenece a ambas geometrías, en morado el núcleo lipídico que no se ha segmentado y en verde el área de más que ha sido segmentada. En un caso práctico no se tendría la geometría real, por lo que la comparación habría que realizarla con la imagen de la representación de alguna variable. Para ajustar el segmentado a la realidad disminuyendo el pixelizado es posible realizar un proceso de suavizado, donde se agranda el núcleo y se suavizan los bordes para conseguir una geometría más realista. En la *Figura 6.61.b* se muestra la superposición entre el núcleo suavizado y el real.

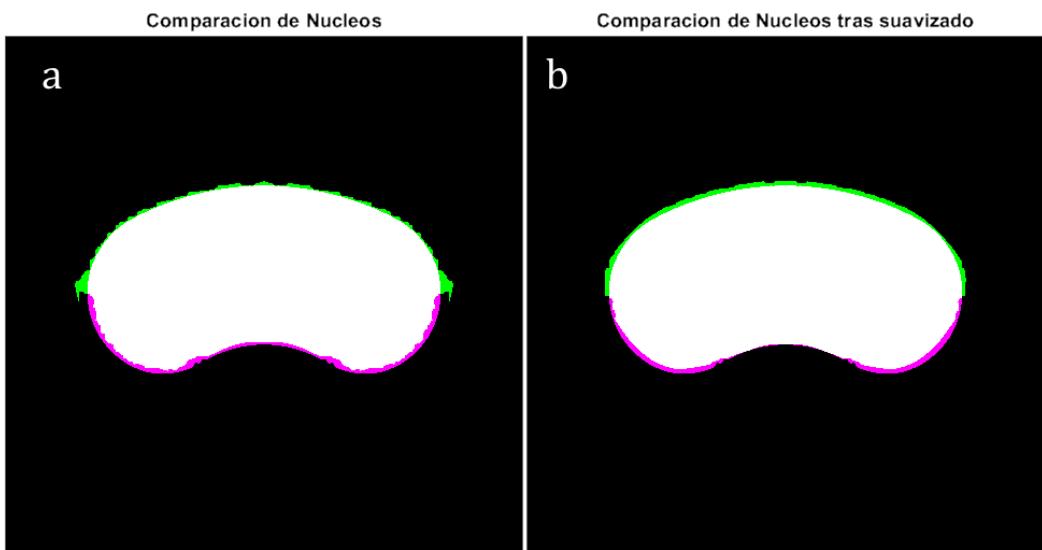


Figura 6.61: Superposición del núcleo real y el segmentado a) tras el proceso de Watershed, b) tras aplicar un proceso de suavizado.

```
nucleo=(L==2); figure(1),imshow(nucleo),title('Nucleo Lipidico Segmentado');
disco_agrandar=strel('disk',3);
```

```
nucleo=imdilate(nucleo,disco_agrandar);imshow(nucleo);
windowSize=60; % Decide as per your requirements
kernel=ones(windowSize)/windowSize^2;
result=conv2(single(nucleo),kernel,'same');
result=result>0.5;
nucleo(~result)=0;
figure, imshow(nucleo);
figure(3), imshow(imfuse(nucleo,nucleo_real)),title('Comparacion de
Nucleos tras suavizado');
```

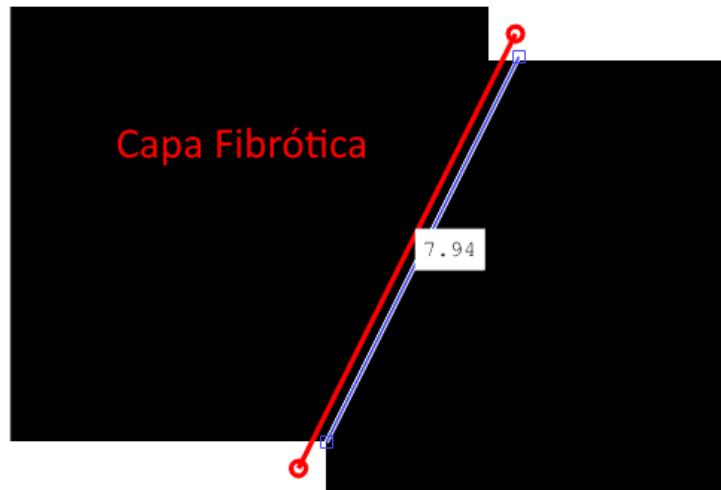
Ajuste de Escala

El último paso es realizar las medidas, de espesor de la capa fibrótica y de área del núcleo lipídico. Para ello es necesario encontrar una referencia que permita pasar de píxeles a unidades de longitud (mm). En clínica no sería necesario este apartado, ya que la imagen del IVUS ya viene con escala. En este trabajo se ha optado por utilizar el diámetro vertical del lumen (*Figura 3.9*), ya que es posible medir los píxeles mediante la función *imdistline()* de Matlab y medir los milímetros en el software Abaqus (considerando la geometría deformada en el *step* a analizar). Relacionando ambas medidas se obtiene el factor de conversión entre píxel-mm.

Otro factor a considerar es que la medida automática entre dos objetos mide la distancia entre los contornos, tomando los puntos la mitad del contorno (*Figura 6.62* línea roja), mientras que la medida real sería menor (*Figura 6.62* línea azul). Por lo tanto hay que restar 1 pixel a las medidas automáticas.

La distancia mínima (pixeles) es de 8.9443

Núcleo Lipídico



Lumen

Figura 6.62: Medida automática (rojo) del espesor de la capa fibrótica en pixeles, medida real (azul) en pixeles.

Por lo tanto las medidas quedarían definidas con el siguiente código. Al final del código se implementa también la medida de los índices 1 y 2 definidos en la memoria (*Ecuaciones 3.39 y 3.40*).

```
escala=2.30605/152; % Se considera el diámetro deformado/(nº de pixeles+1)
```

```
BW=(lumen+nucleo); figure(9), imshow(BW);
BW2 = bwperim(BW,4);
stats = struct2table(regionprops(BW2,'PixelList'));
dist = pdist2(stats.PixelList{1},stats.PixelList{2});
[dmin, idx] = min(dist(:));
[r,c] = ind2sub(size(dist),idx);
pt1 = stats.PixelList{1}(r,:);
pt2 = stats.PixelList{2}(c,:);
distancia_escalada=(dmin-2)*escala;
% Show the result
imshow(BW)
hold on
plot([pt1(1) pt2(1)],[pt1(2) pt2(2)],'r-o','LineWidth',2)
disp(sprintf('Min. distance: %f [pixel]',dmin))
hold on,
title(['La distancia mínima es de ',num2str(distancia_escalada)]);
%% 3.2) Medida de Áreas en pixeles
area_nucleo_segmentado = regionprops(nucleo, 'Area'); % Se mide el area del nucleo segmentado.
area_placa_segmentado = regionprops(placa, 'Area');
area_adventicia_media_segmentado = regionprops(adventicia_media, 'Area');
area_lumen_segmentado=regionprops(lumen,'Area');

%% 3.3) Medida de Áreas en mm2
area_nucleo_real_mm=double(area_nucleo_real.Area)*escala^2;
area_nucleo_segmentado_mm=double(area_nucleo_segmentado.Area)*escala^2;
area_lumen_segmentado_mm=double(area_lumen_segmentado.Area)*escala^2;

nucleo_segmentado_dentro_real=bitand(nucleo_real,nucleo);
nucleo_in=regionprops(nucleo_segmentado_dentro_real, 'Area');
nucleo_in=double(nucleo_in.Area)*escala^2;
nucleo_segmentado_fuera_real=nucleo-nucleo_segmentado_dentro_real;
nucleo_out=regionprops(nucleo_segmentado_fuera_real, 'Area');
nucleo_out=double(nucleo_out.Area)*escala^2;
close all;
Indice_1=(nucleo_in/area_nucleo_real_mm)*100
Indice_2=(nucleo_out/area_nucleo_real_mm)*100
```

Se pueden comparar las áreas medidas con esta escala con las áreas obtenidas en el punto de *Segmentación Real de las Partes* (donde se calculan las áreas reales a nivel elemental). Los datos arrojan errores entre ellos entorno al 2-4 %, lo que se considera válido para este proyecto.

Con todo este código desarrollado es posible medir la distancia de la capa fibrótica y el área del tejido segmentado. El método es aplicable a todas las geometrías analizadas.

