

Proyecto Fin de Máster
Máster en Ingeniería Aeronáutica

Desarrollo de técnicas de mallado MEF avanzadas en
Traction Force Microscopy 3D

Autor: Manuel Rojas Naranjo

Tutor: José Antonio Sanz Herrera

Dpto. Mecánica de Medios Continuos y Teoría de Estructuras
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019



Proyecto Fin de Máster
Máster en Ingeniería Aeronáutica

Desarrollo de técnicas de mallado MEF avanzadas en Traction Force Microscopy 3D

Autor:
Manuel Rojas Naranjo

Tutor:
José Antonio Sanz Herrera
Profesor titular

Dpto. Mecánica de Medios Continuos y Teoría de Estructuras
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2019

Proyecto Fin de Máster: Desarrollo de técnicas de mallado MEF avanzadas en Traction Force Microscopy 3D

Autor: Manuel Rojas Naranjo

Tutor: José Antonio Sanz Herrera

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal

Agradecimientos

En primer lugar quiero dar las gracias a mi tutor, José Antonio, por el tiempo dedicado durante la realización de este trabajo.

Agradecer a Gloria por su apoyo y comprensión durante todos estos años de estudios.

Por último y más importante, muchas gracias a mi familia, por ser ese gran pilar, por su apoyo incondicional y por su cariño. Muchas gracias a mi hermano, por estar siempre ahí y ser una distracción constante en los malos momentos. A mis abuelos, por su preocupación y por tantos cambios de planes instantáneos para prestarme el coche y que yo pudiera ir a la escuela por cualquier imprevisto. Y en especial a mis padres. A ti mamá, por preocuparte a veces incluso demasiado y por ser tan persuasiva e insistente cuando era pequeño para que estudiase una carrera y a ti papá, por ayudarme y solucionarme tantos problemas no solo durante mis estudios sino durante toda mi vida.

Manuel Rojas Naranjo

Sevilla, 2019

Resumen

En los últimos años, el hecho de que las fuerzas mecánicas juegan un papel central en la toma de decisiones celulares ha llegado a ser evidente y esto ha conducido al campo emergente de la mecanobiología. De esta manera, cuantificar dichas fuerzas celulares ha llegado a ser una cuestión importante en múltiples disciplinas biológicas, ya que estas regulan muchas funciones celulares como la diferenciación, crecimiento o la proliferación celular. La metodología *Traction Force Microscopy* es quizás la técnica de estimación de fuerzas más usada y exitosa y es en ella donde se enmarca este estudio.

El trabajo realizado aquí consiste en el desarrollo de un enfoque relacionado con la técnica de mallado adaptativo, para tratar de mejorar los mallados MEF realizados en la metodología TFM. Este enfoque está basado en la creación de unas capas alrededor de la célula estudiada para conseguir una aglomeración mayor de elementos en sus proximidades, que capten mejor la variación de las magnitudes. Tras esto, se obtienen los resultados tanto para un mallado sin capas como con capas y se comparan.

En base a la comparación mencionada, se llegan a una serie de conclusiones y salen a la luz ciertos estudios posibles que pueden ser de ayudas en investigaciones futuras. Por último se adjuntan los códigos generados para que puedan ser de utilidad a otros investigadores.

Abstract

In the last years, the fact that the mechanic forces play a fundamental role in cellular decisions making has come to be clear and this has led to the emerging field of mechanobiology. Therefore, quantifying this cellular forces has become an important mission in numerous biologic branches, since they regulate many cellular functions as the cell differentiation, growth or proliferation. The methodology *Traction Force Microscopy* is maybe the most popular and successful force measure technique and this report is framed in it.

This work consists in the development of an approach related to the adaptive mesh technique, in order to improve the FEM mesh made by the TFM technology. This approach is based on the creation of some layers in the surroundings of the studied cell, in order to get a bigger agglomeration in its proximities. This agglomeration let the mesh get a better magnitude variation. After that, results are obtained for a layered mesh just like for a non layered mesh, and this results are compared.

Taking into consideration the mentioned comparison, few conclusions appear as well as some possible future studies which could be helpful in following researches. Finally, the generated codes are attached in case they may be useful for future researchers.

Índice

Agradecimientos	viii
Resumen	ix
Abstract	x
Índice	xi
Índice de Tablas	xiii
Índice de Figuras	xiv
Glosario	xvii
1 Introducción	1
1.1 Contexto	1
1.2 ¿Qué es Traction Force Microscopy?	2
1.2.1 Limitaciones de la metodología TFM	5
1.2.2 Avances en tecnología y análisis	5
1.2.3 Futuro de TFM	6
1.3 Objetivos	7
1.4 Estructura del documento	7
2 3D Traction force microscopy	8
2.1 Introducción	8
2.2 Experimentación	9
2.3 Microscopía	10
2.3.1 Microscopía confocal de barrido láser (LSCM)	10
2.3.2 Microscopía confocal de disco giratorio (SDCM)	12
2.3.3 Microscopía de fluorescencia de campo amplio (FM)	12
2.4 Resolución del problema mecánico	13
2.4.1 Técnicas para el cálculo del campo de desplazamientos	14
2.4.2 Método de cálculo inverso	17
2.4.3 Método de cálculo directo	17
2.4.4 Método de cálculo inverso vs Método de cálculo directo	18
3 Resolución del problema mecánico con elementos finitos usando un método directo	19

3.1	<i>Formulación y resolución.</i>	19
3.1.1	Formulación	19
3.1.2	Resolución	23
3.2	<i>Trabajo Desarrollado.</i>	24
3.2.1	Geometría	25
3.2.2	Mallado	28
4	Resultados	38
4.1	<i>Resultados para un mallado sin capas y tetraedros de 4 nodos</i>	38
4.2	<i>Resultados para un mallado con capas y tetraedros de 4 nodos</i>	43
5	Conclusiones y trabajos futuros	51
5.1	<i>Conclusiones</i>	51
5.2	<i>Trabajos futuros</i>	53
Anexos		54
	<i>Anexo A: Script Mallado_celula_capas</i>	54
	<i>Anexo B: Script Mallado_celula_sincapas</i>	65
Referencias		74

Índice de Tablas

Tabla 1. Casos considerados de mallas con tetraedros de 4 nodos.	33
Tabla 2. Comparativa de los nodos de los casos de 4 y 10 nodos.	34
Tabla 3. Casos considerados de mallas sin capas	38

Índice de Figuras

Figura 1. TFM en 3D [5].	2
Figura 2. Esquema y desplazamiento de los beads antes y después de aplicar las fuerzas. TFM 2D [1].	3
Figura 3. Desplazamiento de los beads antes y después de aplicar las fuerzas. TFM 3D [2].	4
Figura 4. Esquema general de TFM en 3D.	8
Figura 5. Esquema de un microscopio confocal de barrido láser [9].	11
Figura 6. Esquema de un microscopio confocal de disco giratorio [13].	12
Figura 7. Esquema de un microscopio de fluorescencia de campo amplio [15].	13
Figura 8. Esquema del método de seguimiento de partículas individuales [23].	15
Figura 9. Esquema de DVC [10].	16
Figura 10. Diagrama de flujo del método directo [2].	17
Figura 11. Tracciones impuestas en el entorno de la célula.	24
Figura 12. Esquema de la idea general.	25
Figura 13. Geometría del problema (<i>izq. Célula embebida dentro de la matriz extracelular de gel. dcha. Célula aislada.</i>)	26
Figura 14. Elementos tetraédricos de 4 y 10 nodos.	26
Figura 15. Circunferencia de Delaunay [25].	27
Figura 16. Representación de las capas generadas alrededor de la célula	28
Figura 17. Diagrama de flujo del Iso2mesh [27].	29
Figura 18. Malla de la prueba 1	35
Figura 19. Malla de la prueba 2	35
Figura 20. Malla de la prueba 3	35
Figura 21. Malla de la prueba 4	35
Figura 22. Malla de la prueba 5	35
Figura 23. Malla de la prueba 6	35
Figura 24. Malla de la prueba 7	36
Figura 25. Malla de la prueba 8	36
Figura 26. Malla de la prueba 9	36
Figura 27. Malla de la prueba 10	36
Figura 28. Malla de la prueba 11	36
Figura 29. Malla de la prueba 12	36
Figura 30. Malla de la prueba 1 sin capas	38
Figura 31. Malla de la prueba 2 sin capas	38
Figura 32. Malla de la prueba 3 sin capas	39
Figura 33. Malla de la prueba 4 sin capas	39
Figura 34. Malla de la prueba 5 sin capas	39

Figura 35. Malla de la prueba 6 sin capas	39
Figura 36. Desplazamiento en la superficie celular en la prueba 1 sin capas.	40
Figura 37. Desplazamiento en la superficie celular en la prueba 2 sin capas.	40
Figura 38. Desplazamiento en la superficie celular en la prueba 3 sin capas.	40
Figura 39. Desplazamiento en la superficie celular en la prueba 4 sin capas.	40
Figura 40. Desplazamiento en la superficie celular en la prueba 5 sin capas.	40
Figura 41. Desplazamiento en la superficie celular en la prueba 6 sin capas.	40
Figura 42. Desplazamiento en un corte en la prueba 1 sin capas.	41
Figura 43. Desplazamiento en un corte en la prueba 2 sin capas.	41
Figura 44. Desplazamiento en un corte en la prueba 3 sin capas.	41
Figura 45. Desplazamiento en un corte en la prueba 4 sin capas.	41
Figura 46. Desplazamiento en un corte en la prueba 5 sin capas.	42
Figura 47. Desplazamiento en un corte en la prueba 6 sin capas.	42
Figura 48. Vector tensión en la superficie celular en la prueba 1 sin capas.	42
Figura 49. Vector tensión en la superficie celular en la prueba 2 sin capas	42
Figura 50. Vector tensión en la superficie celular en la prueba 3 sin capas.	43
Figura 51. Vector tensión en la superficie celular en la prueba 4 sin capas.	43
Figura 52. Vector tensión en la superficie celular en la prueba 5 sin capas.	43
Figura 53. Vector tensión en la superficie celular en la prueba 6 sin capas.	43
Figura 54. Desplazamiento en la superficie celular en la prueba 1 con capas.	44
Figura 55. Desplazamiento en la superficie celular en la prueba 2 con capas.	44
Figura 56. Desplazamiento en la superficie celular en la prueba 3 con capas	44
Figura 57. Desplazamiento en la superficie celular en la prueba 4 con capas.	44
Figura 58. Desplazamiento en la superficie celular en la prueba 5 con capas.	44
Figura 59. Desplazamiento en la superficie celular en la prueba 6 con capas.	44
Figura 60. Desplazamiento en la superficie celular en la prueba 7 con capas.	45
Figura 61. Desplazamiento en la superficie celular en la prueba 8 con capas	45
Figura 62. Desplazamiento en la superficie celular en la prueba 9 con capas.	45
Figura 63. Desplazamiento en la superficie celular en la prueba 10 con capas.	45
Figura 64. Desplazamiento en la superficie celular en la prueba 11 con capas.	45
Figura 65. Desplazamiento en la superficie celular en la prueba 12 con capas.	45
Figura 66. Desplazamiento en un corte en la prueba 1 con capas.	46
Figura 67. Desplazamiento en un corte en la prueba 2 con capas.	46
Figura 68. Desplazamiento en un corte en la prueba 3 con capas.	46
Figura 69. Desplazamiento en un corte en la prueba 4 con capas	46
Figura 70. Desplazamiento en un corte en la prueba 5 con capas.	47
Figura 71. Desplazamiento en un corte en la prueba 6 con capas.	47
Figura 72. Desplazamiento en un corte en la prueba 7 con capas.	47
Figura 73. Desplazamiento en un corte en la prueba 8 con capas.	47

Figura 74. Desplazamiento en un corte en la prueba 9 con capas.	47
Figura 75. Desplazamiento en un corte en la prueba 10 con capas.	47
Figura 76. Desplazamiento en un corte en la prueba 11 con capas.	48
Figura 77. Desplazamiento en un corte en la prueba 12 con capas.	48
Figura 78. Vector tensión en la superficie celular en la prueba 1 con capas.	48
Figura 79. Vector tensión en la superficie celular en la prueba 2 con capas.	48
Figura 80. Vector tensión en la superficie celular en la prueba 3 con capas.	49
Figura 81. Vector tensión en la superficie celular en la prueba 4 con capas.	49
Figura 82. Vector tensión en la superficie celular en la prueba 5 con capas.	49
Figura 83. Vector tensión en la superficie celular en la prueba 6 con capas.	49
Figura 84. Vector tensión en la superficie celular en la prueba 7 con capas.	49
Figura 85. Vector tensión en la superficie celular en la prueba 8 con capas.	49
Figura 86. Vector tensión en la superficie celular en la prueba 9 con capas.	50
Figura 87. Vector tensión en la superficie celular en la prueba 10 con capas.	50
Figura 88. Vector tensión en la superficie celular en la prueba 11 con capas.	50
Figura 89. Vector tensión en la superficie celular en la prueba 12 con capas.	50
Figura 90. Comparativa (<i>Arriba prueba 2 sin capas, en medio prueba 2 con capas y abajo prueba 8 con capas</i>).	52

Glosario

MEF	Método de elementos finitos (Finite element method)
TFM	Microscopía de fuerzas de tracción (Traction force microscopy)
ECM	Matriz extracelular
AFM	Microscopio de fuerza atómica (Atomic force microscope)
PAA	Poliacrilamida
PDMS	Polidimetilsiloxano
E	Módulo de Young
ν	Coefficiente de Poisson
PEGDA	Polietilenglicol diacrilato
LSCM	Microscopía confocal de barrido láser (laser scanning confocal microscopy)
SDCM	Microscopía confocal de disco giratorio (Spinning disk confocal microscopy)
FM	Microscopía de fluorescencia de campo amplio (Widefield fluorescence microscopy)
MLS	Método de mínimos cuadrados móviles
DVC	Correlación de volumen digital (Digital volume correlation)
DIC	Correlación de imagen digital (Digital image correlation)

1 INTRODUCCIÓN

Para comenzar, en este capítulo se va a contextualizar el tema tratado, proporcionándose una visión de sus generalidades, así como de sus limitaciones y avances recientes. Se explicarán los objetivos que se persiguen con este trabajo y se expondrá de manera breve la estructura que se ha seguido en el documento.

1.1 Contexto

En las últimas tres décadas, ha llegado a ser evidente que las fuerzas mecánicas juegan un papel central en la toma de decisiones celulares, conduciendo al emergente campo de la mecanobiología. Hoy en día, esta disciplina está apareciendo en múltiples campos en investigación biomédica. En contraste con las creencias convencionales, recientes investigaciones indican que las células regulan su mecánica celular no solo después de los sucesos desencadenados por estímulos externos o internos, sino que las células emplean una gran diversidad de mecanismos de retroalimentación que les permiten ajustar dinámicamente su mecánica para cumplir con las necesidades fisiológicas. En consecuencia, esto proporciona una imagen imprevista en donde las células activamente ejercen y resisten fuerzas para ajustar sus propiedades y así facilitar su función, la cual es particularmente importante durante las interacciones físicas con otras células o con el entorno extracelular.

Por lo tanto, cuantificar estas fuerzas celulares ha llegado a ser una misión importante en múltiples disciplinas como biofísica, biología celular o inmunología. Esto es porque dichas fuerzas, ya sean generadas por las células o aplicadas externamente, regulan muchas funciones celulares como la diferenciación, el crecimiento, la adhesión, la migración y la proliferación celular. Además, procesos fisiológicos como la embriogénesis, la metástasis del cáncer o la curación de una herida pueden verse significativamente influidos por la aparición y transmisión de fuerzas físicas entre las células y su entorno.

En su estado nativo, todas las células animales viven en un microentorno tridimensional soportadas arquitectónicamente por la matriz extracelular (ECM) y ejerciendo fuerzas sobre ella a través de los contactos célula-ECM. Como ya se ha mencionado, el equilibrio de fuerzas que surge en las interacciones célula-ECM desempeña un papel esencial en varios procesos fisiológicos y patológicos. Un ejemplo patológico ampliamente conocido es la relación entre un entorno de tejido rígido y el diagnóstico clínico de un tumor de mama. Un trabajo reciente de "Laboratorios Weaver" ha demostrado que la tumorigénesis mamaria está asociada con la interrupción de dicho equilibrio de fuerzas a través de la rigidez de la ECM y el aumento de las adherencias focales. De forma más sutil, varios trabajos han demostrado que las fuerzas mecánicas dan forma a la morfogénesis durante el desarrollo temprano del animal.

Medir estas fuerzas celulares es un desafío, y se espera que la Microscopía de fuerzas de tracción (Traction force microscopy o TFM) siga siendo la principal metodología de estimación de fuerzas. Además de la complejidad de los mecanismos de retroalimentación de la mecánica celular, la estimación de las fuerzas celulares es un desafío en sí mismo, debido a la física de la medición de dichas fuerzas mecánicas. Debido a que las células no emiten señales mecánicas que podrían ser detectadas y analizadas sin contacto con la misma, dicha cuantificación de fuerzas exige el contacto directo de la tecnología necesaria con la célula. Por ejemplo, la rigidez mecánica de las células en la forma del módulo elástico, se determina marcando físicamente su superficie con el uso de fuerzas conocidas, como por ejemplo con un microscopio de fuerza atómica (AFM). En algunos experimentos de TFM, las fuerzas celulares se cuantifican monitorizando las tracciones superficiales producidas por las células en un sustrato elástico. Para añadir mayor complejidad a esta

perspectiva, las medidas mecánicas de las células también dependen de como se realizan, ya que pueden cambiar enormemente en diferentes escalas de tiempo y longitud. En escalas de tiempos cortas (milisegundos) y escalas de longitud grandes (micrómetros), las células muestran propiedades poroelásticas y en escalas de tiempo grandes (minutos) muestran un comportamiento de ley potencial en respuesta a la aplicación de fuerzas externas. Por lo tanto, parámetros tales como desplazamientos, tracciones celulares y tasas de rotación deben ser estudiados en multitud de escalas de tiempo y longitud para caracterizar exhaustivamente las propiedades mecánicas celulares y la generación de fuerzas. Todo esto, deja patente la dificultad y el desafío que supone cuantificar las fuerzas celulares.

Por lo tanto, tras lo comentado anteriormente, aunque puede ser enormemente instructivo y útil llegar a conocer y estimar las fuerzas que las células ejercen entre ellas o en la interfase célula-matriz, es tremendamente complicado. Debido a la simplicidad de su implementación, su flexibilidad para reproducir las condiciones celulares y un proceso de análisis bien establecido, TFM ha llegado a ser una herramienta estándar en muchos laboratorios que trabajan en mecanobiología y es por eso, por lo que dicho tema es tratado en este proyecto [1-4].

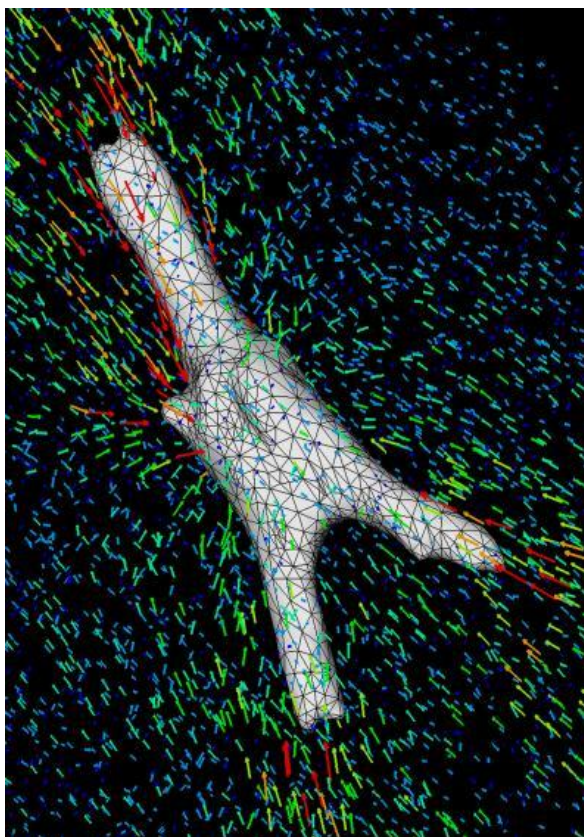


Figura 1. TFM en 3D [5].

1.2 ¿Qué es Traction Force Microscopy?

Traction force microscopy es quizás la tecnología de estimación de fuerzas más exitosa y ampliamente usada, ya que continúa ofreciendo la mayoría de los requisitos discutidos anteriormente para la cuantificación de fuerzas producidas por células vivas. En consecuencia, TFM es superior a otras tecnologías tal como se ha comentado antes, debido a la simplicidad de su implementación, su flexibilidad para imitar las condiciones celulares y un proceso de análisis bien establecido.

Las medidas cuantitativas de las fuerzas de tracción de una sola célula comenzaron hace aproximadamente tres décadas gracias a la aparición de "2D Traction force microscopy" (TFM 2D). En TFM 2D, las células animales

se cultivan en la superficie de un sustrato o hidrogel 2D elástico, delgado (entre 20 y 30 μm) y con rigidez ajustable como poliacrilamida (PAA) o polidimetilsiloxano (PDMS). Las células se incuban para permitir que se desarrollen las fuerzas de tracción. Dentro de dicho hidrogel, se introducen también, beads fluorescentes e inmóviles que sirven como marcadores de referencia. Tras esto, se utilizan fármacos para desactivar las funciones citoesqueléticas, eliminando así las tracciones celulares y se registran los desplazamientos de las perlas fluorescentes o beads usando fluorescence microscopy (microscopía de fluorescencia). Las fuerzas de tracción celulares se obtienen a partir del desplazamiento de los beads y de las propiedades mecánicas del hidrogel utilizando una función de Green o un enfoque basado en Fourier. Con el tiempo, TFM 2D se ha convertido en una metodología madura, jugando un importante rol en la comprensión de la interacción célula-sustrato y célula-célula en la adhesión y migración celular, la formación y migración de tejidos [1, 2]. En la figura 2 se aprecia a la izquierda un esquema de lo anteriormente comentado y a la derecha un ejemplo del desplazamiento de los beads antes y después de la aplicación de las fuerzas.

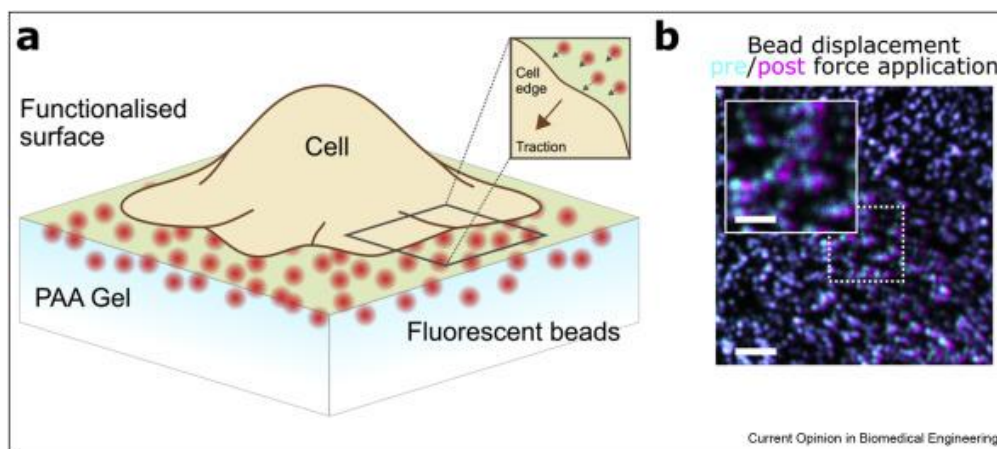


Figura 2. Esquema y desplazamiento de los beads antes y después de aplicar las fuerzas. TFM 2D [1].

El cultivo de células en 3D, donde las células están embebidas en una ECM, es cada vez más investigado por la comunidad científica, ya que muchos tipos de células dentro de una ECM requieren señales biofísicas y bioquímicas para realizar funciones fisiológicamente realistas. Se ha llegado a la conclusión de que las células se comportan de maneras muy diferentes en un sustrato 2D que dentro de biomatrices 3D. En 2D, las células se adhieren al sustrato solo en sus lados basales, mientras en 3D, las células se unen a la ECM por todas partes y son soportadas por la estructura de la ECM. Además, los mecanismos moleculares que gobiernan la adhesión y la migración celular en 2D y 3D no tienen ninguna relación aparente.

A medida que los cultivos celulares *in vitro* en 3D se convierten en la corriente principal, la metodología "3D Traction force microscopy" (TFM 3D) está avanzando rápidamente para satisfacer la necesidad de cuantificar las fuerzas mecánicas ejercidas por una única célula animal en 3D. La idea básica detrás de TFM 3D es similar a la de TFM 2D y consta de dos partes. En primer lugar, la medida de los desplazamientos de los marcadores de referencia fluorescentes o beads causados por la eliminación de las fuerzas de tracción que aplica la célula. En segundo lugar, la traducción de los desplazamientos de los beads a un campo de tensiones. A pesar de la simplicidad de estas dos ideas, TFM 3D aún se encuentra en su etapa inicial, y no se usa demasiado. Esta metodología se ve obstaculizada debido a la dificultad y coste de las imágenes de las características en escala submicrométrica en 3D, el conocimiento de las propiedades mecánicas de la ECM y la necesidad de algoritmos de computación complejos que no son fácilmente accesibles para los expertos en biología [2]. En la figura 3 se puede apreciar una célula cancerígena perteneciente a un cáncer de mama, la cual está embebida en una matriz de colágeno I. En la imagen se ve los desplazamientos de los beads que la rodean coloreados, para mostrar el cambio de posición de estos, desde el caso donde la célula está aplicando carga, hasta el caso donde

está relajada.

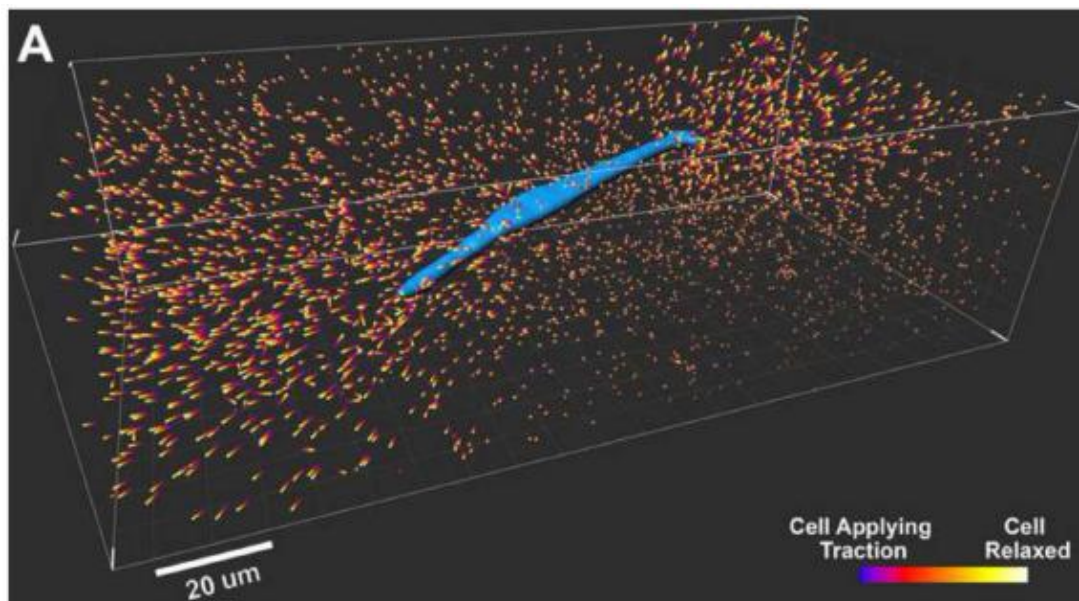


Figura 3. Desplazamiento de los beads antes y después de aplicar las fuerzas. TFM 3D [2].

¿ Por qué es necesario conocer el campo de desplazamientos en el entorno de la célula (desplazamientos de los beads) para obtener las fuerzas de tracción que las células ejercen? Desde un punto de vista general, las fuerzas no son una cantidad experimentalmente accesible de forma directa, sino que tienen que ser deducidas del hecho de que crean algún tipo de movimiento. A pesar de que este movimiento puede seguir diferentes leyes dependiendo de los detalles del sistema considerado (elástico, viscoso, ...), para medir una fuerza lo único que se necesita es monitorizar algún tipo de dinámica. Esto se ilustra mejor utilizando un muelle elástico lineal. En este caso, la fuerza se define como $F = k \cdot x$, siendo k la constante del muelle y x el desplazamiento. Sin una medida de x , no sería posible ninguna medida de la fuerza (k se obtiene mediante experimentos de calibración). Para medir x , se debe conocer el estado de referencia $x=0$ así que por lo tanto se necesita normalmente de un proceso de relajación para determinar el valor absoluto de x . De esta manera, se puede ver como incluso las situaciones aparentemente estáticas requieren de algún tipo de medida dinámica. Lo que se quiere mostrar con esto, es que cada proceso de medición directa de fuerzas celulares tiene que empezar con la identificación de un adecuada forma de medir la deformación [3].

La implementación de los experimentos de TFM convencionales se puede lograr con cualquier microscopio confocal o de epifluorescencia sin componentes ópticos adicionales. Los sustratos elásticos, como los hidrogeles de PAA que se usan con frecuencia, se pueden fabricar con elasticidades en un rango que se extiende desde menos de 1kPa hasta unos pocos de cientos de kPa, permitiendo medidas de flexibilidad que imitan multitud de diferentes rigideces de superficies y entornos celulares. Además, características nanotopologías como el tamaño de la malla del gel, pueden ser alteradas directamente al cambiar el equilibrio de monómeros y reticulantes dentro del gel PAA, sin cambiar su rigidez. Materiales alternativos como el silicio, el colágeno y el PDMS, exhiben propiedades ópticas y elásticas similares y se consideran candidatos prometedores para experimentos similares a los de TFM. El índice de refracción de los hidrogeles de silicio coincide con el índice del vidrio en el que se encuentra la muestra y, por lo tanto, permite una TFM basada en la fluorescencia de reflexión interna total (TIRF), la cual está limitada a las tracciones 2D. Dentro de la metodología TFM 3D, en aquellos casos donde se usa colágeno, es posible introducir células dentro de microambientes fisiológicos 3D, aunque su análisis aún exige algoritmos complejos debido a sus propiedades mecánicas no lineales. Específicamente, las estructuras de análisis requieren amplios conocimientos matemáticos y la mayoría de biólogos o biofísicos no dispone de ellos. En contraste, en TFM, el cálculo de las

fuerzas mecánicas de las tracciones adquiridas está bien establecido en la mayoría de los casos y las soluciones de softwares de código abierto están disponibles para usuarios sin experiencia. Además, la superficie superior de los hidrogeles puede ser funcionalizada covalentemente con lípidos y proteínas simulando condiciones fisiológicas con las que interactúan las células. Juntas, estas propiedades permiten cuantificar las tracciones celulares para una gran variedad de células en múltiples condiciones [1].

1.2.1 Limitaciones de la metodología TFM

TFM ha tenido éxito cuantificando las fuerzas celulares pero impide la caracterización de las propiedades mecánicas, lo cual naturalmente limita el método a solo una de las dos ramas de la mecánica celular. TFM no permite la interrogación activa de células a través de por ejemplo, marcas para la cuantificación de propiedades mecánicas tales como la deformabilidad, la viscosidad o rigidez de las células. La interpretación de los experimentos en 2D de TFM están fundamentalmente limitados al examen de las fuerzas en la membrana de las células. TFM en 2D solo puede proporcionar información sobre la producción de fuerzas de compartimentos celulares que ejerzan tracciones sobre la superficie apical plana del hidrogel. Los esfuerzos para extender TFM a hidrogeles tridimensionales han mostrado resultados prometedores pero con un bajo rendimiento así que el resultado estadístico es limitado y requiere un análisis especializado como simulaciones de elementos finitos para recuperar las fuerzas de los conjuntos de datos tridimensionales.

Un importante factor hardware para la precisión de TFM es la frecuencia de escaneo de la detección de los beads fluorescentes. La precisión de TFM se ve afectada por la duración de adquisición de las posiciones de los marcadores de referencia. Solo las frecuencias de adquisición suficientemente altas permiten un seguimiento preciso de las posiciones de los beads a lo largo del tiempo. De manera similar, en cualquier experimento, el muestreo del campo de desplazamiento debe cumplir con el criterio de Nyquist. Esto restringe la densidad de los marcadores de referencia que debe ser lo suficientemente alta para captar la complejidad del campo de tracciones que es aplicado por la célula. Si la densidad de marcadores es demasiado baja, habrá áreas del gel que se moverán sin ser reflejadas por el movimiento de algún bead y la información de ese movimiento se perderá. De forma inversa, si la densidad de marcadores es demasiado alta, la información también se perderá como resultado de las funciones de dispersión de los puntos de cada bead individual que se superponen con aquellos más cercanos, lo que hace que sus desplazamientos relativos sean difíciles de recuperar con precisión. Además, la máxima densidad de los marcadores dentro del gel también debe equilibrarse con la capacidad de rastrear los beads y la sensibilidad a la luz fluorescente de la muestra biológica. La localización sistemática de los marcadores podría mejorar la densidad de los mismos lo cual puede lograrse a través de patrones de impresión. En consecuencia, la calidad y la solidez de los experimentos de TFM podrían mejorarse mediante una mejor preparación de los geles con concentraciones de beads precisas, regulares y predeterminadas utilizando por ejemplo, sistemas de impresión de sustrato. En consecuencia, el mayor inconveniente de la metodología TFM clásica es su limitada sensibilidad debido a la finita densidad a la que se puede muestrear el campo de desplazamiento en el gel, la cual debe ser lo suficientemente alta como para reflejar la complejidad del campo de fuerzas aplicado por la célula [1].

1.2.2 Avances en tecnología y análisis

Para superar los desafíos anteriores, se ha mejorado recientemente la resolución espacial y la precisión en TFM usando microscopía de reducción de la emisión estimulada (STED). El incremento de la resolución espacial de STED-TFM (STFM) permite un muestreo de las fuerzas generadas por la célula más de cinco veces mayor que el del TFM convencional, lo que conduce a una estimación más precisa de las tracciones celulares. Esto debe ser equilibrado con otros factores ópticos, como la densidad máxima de los beads dentro del gel, la capacidad para rastrearlos y la sensibilidad a la luz fluorescente de la muestra biológica. Los intentos

anteriores para superar esta limitación implicaron el uso de dos colores diferentes de marcadores de referencia. Sin embargo, debido a su dependencia con la separación espectral de los beads, limita esta técnica en última instancia por el rango espectral del microscopio [1]. También respecto a la microscopía, aunque en la actualidad los métodos de imagen como la microscopía confocal o la microscopía de fluorescencia son suficientes para capturar una instantánea detallada en un instante dado, lograr este nivel de detalle dinámicamente, requerirá un método que combine mejor la resolución espacial y temporal alta con una fototoxicidad baja. En este sentido, un método prometedor es la reciente combinación de la microscopía estructurada de la iluminación (SIM) y la microscopía de lámina de luz [2].

Los aspectos teóricos de TFM son igualmente importantes para optimizar la precisión de la reconstrucción de las fuerzas. Se pueden esperar mejoras críticas del STFM a partir del análisis de imagen, y existen varios métodos mediante los cuales cuantificar el desplazamiento de los marcadores. Estos métodos pueden dividirse en aquellos que dependen del seguimiento de cada bead individual conocidas como técnicas de seguimiento de partículas individuales y los que dependen de la correlación de desplazamientos dentro de las regiones de un imagen, como la velocimetría por imagen de partículas. La elección del método depende de la naturaleza de las fuerzas esperadas y de la densidad de marcadores dentro de la imagen. Por lo tanto, es importante considerar desde el principio el análisis más adecuado para el experimento en particular. Por ejemplo, se ha demostrado recientemente que el seguimiento de flujo óptico mejora la resolución efectiva de TFM, permitiendo la cuantificación de las fuerzas generadas por pequeñas adhesiones focales de células vivas [1].

1.2.3 Futuro de TFM

La combinación de TFM y la microscopía de súper resolución como STED, podría permitir que esta metodología de cuantificación de fuerzas no solo permanezca como líder en este campo, sino que extienda su aplicabilidad a cuestiones biológicas más complejas en un futuro próximo. Por lo tanto, el impacto y el futuro de TFM es muy prometedor. Concretamente, el aumento de la sensibilidad ofrecido por STFM permitirá que la técnica aborde en escalas de longitud actualmente prohibidas para TFM convencional, proporcionando medidas de fuerzas ejercidas por pequeños orgánulos celulares o receptores individuales sin precedentes. Esto será vital para la comprensión del papel de la mecánica en los límites de muchas disciplinas como inmunología, neurociencia o biología del cáncer.

Para mejorar aún más el impacto de TFM, los enfoques futuros pueden dirigirse a incrementar la producción estadística de la técnica. Esto implica a experimentos y análisis, porque actualmente solo se pueden medir células individuales en un tiempo dado, y los procesos de análisis duran varias horas. Superar estos inconvenientes permitiría extender sistemáticamente los experimentos de TFM a una lectura estandarizada de generación de fuerzas mecánicas e investigar aspectos de la mecanobiología tales como la mecanosensibilidad, los mecanismos de retroalimentación o la capacidad de las células para ajustar la producción de fuerzas según la rigidez del hidrogel. Por ejemplo, esto podría permitir cuantificar como las células-T ajustan dinámicamente sus propiedades elásticas como la rigidez superficial alternando su tiempo de rotación y la longitud de los filamentos de actina de su dominio cortical en respuesta a las fuerzas de activación de las células presentadoras de antígenos. De la misma manera, la desregulación de la mecanosensibilidad de los miofibroblastos podría ser medida para conocer su contribución en el desarrollo de la fibrosis humana [1].

1.3 Objetivos

El objetivo global de este proyecto es un estudio para mejorar las mallas utilizadas en la metodología "Traction force microscopy 3D" para conseguir capturar unos mejores resultados de las tracciones celulares. El enfoque utilizado es generar dos capas alrededor de la célula en el dominio del gel o matriz extracelular. La razón de ser de estas capas, es porque gracias a ellas se puede controlar el tamaño del elemento en las zonas más cercanas a la célula, pudiéndose generar ahí una malla más fina, con la que se pueda captar mejor en el gradiente de tracciones en esa zona.

Para la realización del trabajo se han marcado una serie de objetivos concretos. El primero es hacer una recopilación de información para poner en contexto y explicar, aunque de forma somera, todo lo que tenga que ver con la metodología TFM 3D. Otro objetivo, es la elaboración de un código que permita mallar el entorno de una célula dada creando las capas comentadas y que dicho código permita cierto control sobre la malla al modificar algunos parámetros de entrada. Por último, un tercer objetivo es la obtención de las tracciones celulares utilizando las mallas obtenidas con el código anterior y la exposición de una serie de conclusiones tras el análisis de los resultados sacados.

1.4 Estructura del documento

En este capítulo que concluye, se ha contextualizado el tema tratado proporcionándose una visión amplia del porque de su importancia, de sus generalidades y del motivo de su estudio. Además se ha hecho un repaso acerca de las limitaciones que tiene actualmente, de los avances en tecnología y análisis y del futuro que puede deparar a esta metodología conocida como Traction force microscopy.

En el capítulo 2, se hará hincapié en los principios en lo que se basa la técnica TFM 3D, explicándose en cierta profundidad las distintas etapas en las que se divide y haciéndose un repaso de como se encuentra el estado del arte en cada una de estas fases.

A continuación, en el capítulo 3, se describirá todo el proceso seguido durante los meses de trabajo, hablando de la formulación del problema, del camino seguido para su resolución, de su geometría y del mallado del entorno celular. A medida que se vaya tocando cada una de las anteriores cuestiones, se irá haciendo referencia a las partes asociadas del código generado y sus implicaciones en él.

En el capítulo 4, se presentan los resultados obtenidos así como los comentarios que se derivan del análisis de los mismos. En el capítulo 5, se expondrán las conclusiones finales de este proyecto, además de los posibles trabajos futuros y mejoras que se podrían desarrollar a partir de este trabajo.

Por último, en el anexo se incluye el código desarrollado durante el trascurso del proyecto para que puede ser utilizado o modificado si fuese de utilidad en investigaciones futuras.

2 3D TRACTION FORCE MICROSCOPY

En este capítulo, aunque ya se ha comentado con anterioridad cuales son los principios en los que se basa TFM, se va a hacer hincapié en ellos, explicándose las distintas etapas en las que se divide esta técnica y haciéndose un repaso de como se encuentra el estado del arte en cada una de ellas.

2.1 Introducción

TFM 3D es una metodología diseñada para medir cuantitativamente el campo de tracción que una única célula incrustada dentro de una ECM puede ejercer. Existen varios pasos importantes dentro de la tecnología 3D TFM, algunos de los cuales ya han sido comentados. Primero, las células animales vivas ejerciendo tracción se cultivan dentro una ECM compuesta de un biomaterial transparente como son por ejemplo, la poliacrilamida o el colágeno, que simulan los entornos biológicos en los que se encuentran las células. También, en esta ECM, se introducen esferas fluorescentes de tamaño micrométrico, conocidas como beads. Después, se usan enzimas o fármacos para inhibir la actividad celular eliminando así las tracciones que la célula ejerce y se obtienen imágenes de los desplazamientos resultantes en 3D de los beads fluorescentes. Tras esto, se utiliza un algoritmo de cálculo para traducir los desplazamientos de los beads en el campo de tracciones ejercido por la célula en su entorno. En la siguiente figura 4 se muestran a grandes rasgos cuales son las diferentes etapas de dicha técnica [2].

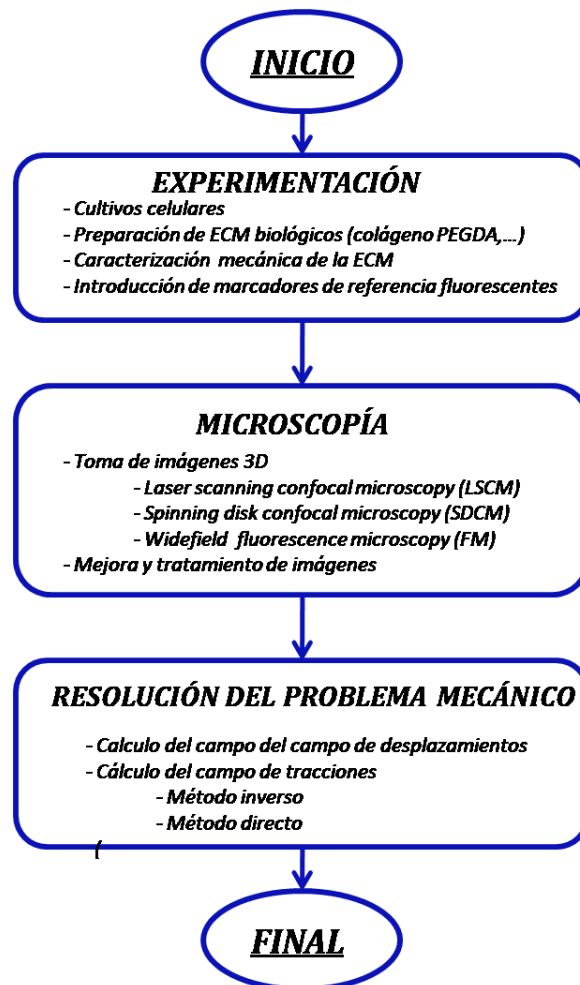


Figura 4. Esquema general de TFM en 3D.

En primer lugar, se encuentra la fase de laboratorio, donde se realizan los cultivos celulares y se introducen junto con los beads fluorescentes dentro de un biomaterial transparente, el cual además debe ser caracterizado mecánicamente. Tras esta etapa, viene la de microscopía en la cual se hace la toma de imágenes de la muestra. A continuación, se pasa a la resolución del problema mecánico, donde se realiza el cálculo de los desplazamientos utilizando algoritmos como DVC, que rastrea el desplazamiento central de los diversos subconjuntos espaciales de la imagen. Tras esto, en esta misma etapa, utilizando las propiedades del material, se es capaz de calcular las fuerzas de tracción que las células ejercen sobre su entorno.

2.2 Experimentación

Esta etapa abarca todas aquellas acciones realizadas en el laboratorio destinadas a preparar la célula para llevar a cabo el experimento. Dentro de ella se encuentran tanto la realización de los cultivos celulares, el aislamiento de las células, la generación del sustrato ECM de gel, la caracterización de las propiedades del gel o la introducción de la célula y los beads o marcadores de referencia fluorescentes en dicho gel, los cuales deben ser esparcidos aleatoriamente y en una cantidad suficiente para obtener buenos resultados.

Quizás lo más destacable en esta etapa sea la fabricación y caracterización del gel. Es importante seleccionar un material adecuado, ya que este debe ser suficientemente parecido al entorno biológico en el que se encuentran las células en la realidad, para que los resultados sean válidos. En función de la respuesta y características del gel, se podrá usar el modelo de elasticidad lineal o habrá que utilizar otros.

Los materiales más comunes para los sustratos (2D) en TFM son geles de poliacrilamida (PAA) o polidimetilsiloxano y elastómeros de silicona (PDMS). Estos tienen una rigidez que se puede ajustar variando la temperatura de curado o la cantidad de agente reticulante que incluyen los polímeros, abarcando así una amplia gama de módulos de Young. Los geles de PAA pueden tener un E entre 100 Pa y 100 kPa mientras las rigideces de las siliconas pueden ir desde $E \sim 3$ kPa a 2 MPa. Para los casos 3D, donde la célula es introducida en una matriz, los materiales más utilizados son polietilenglicol diacrilato (PEGDA) y colágeno I [8].

Hay dos formas diferentes de cultivar células en relación con la ECM en las implementaciones actuales de TFM 3D. El primer método consiste en cultivar células en la superficie de un gel de PAA funcionalizado e incrustar los beads dentro del volumen. Este método, conocido como 2.5D [12], aprovecha las propiedades mecánicas lineales y elásticas bien caracterizadas del gel de PAA y es capaz de mapear el campo de tracciones tridimensionales dentro del gel, directamente bajo la célula. La limitación de este método es que las células están en contacto con el sustrato plano solo en una superficie en lugar de en un contexto tridimensional. El segundo método consiste en introducir células completamente dentro de una ECM, lo cual es un modelo completamente tridimensional. La ventaja de este método es que las células están rodeadas y soportadas en todas las direcciones por una ECM. El uso de geles sintéticos está limitado por la dificultad de crear un gel sintético que sea compatible con el cultivo celular y retenga características fisiológicamente realistas de la ECM. En el caso del gel de colágeno, la limitación es la dificultad con la que se puede manipular el gel de origen nativo y sus propiedades mecánicas relativamente complicadas [2].

Para preparar tanto los cultivos celulares como los geles en los que se encuentran embebidas las células asemejando las condiciones de las matrices extracelulares, existen ciertas formas de proceder o protocolos que se pueden encontrar en algunos "papers" como en [6, 7].

Tal como se ha comentado antes, también es necesario conocer las propiedades del material como son el módulo de Young, E , o el coeficiente de Poisson, ν . Si la rigidez del material es $E \sim 100$ kPa, de modo que pueda soportar su propio peso, se puede realizar un ensayo de compresión entre platos y si el material es más blando se puede obtener las propiedades mediante un ensayo de reología o nanoindentador [8].

2.3 Microscopía

Cuando se habla de microscopía se hace referencia a todas aquellas acciones destinadas a obtener las imágenes necesarias para conseguir el campo de desplazamiento generado por la célula, así como la mejora de la calidad de dichas imágenes para reducir el error que se pueda cometer.

La tarea más importante llevado a cabo en esta etapa es la toma de imágenes del caso cargado y del caso libre de carga. El primero se refiere a aquel donde la célula impone un campo de desplazamiento a su entorno y a los beads que en él se encuentran, mientras el segundo hace referencia al caso donde la célula está inhibida o muerta por la adición de sustancias como detergentes o enzimas como la tripsina, proteinasa K, o tritón X-100 haciendo que su entorno y los beads vuelvan al estado indeformado.

El objetivo clave de las imágenes tal como ya se ha comentado, es medir en ellas de forma precisa, el centroíde de los marcadores de referencia fluorescentes para poder rastrear su movimiento posteriormente. Para ello estas imágenes deben ser tomadas con un buen contraste entre estos beads y el fondo y sin saturación de la imagen.

Cuando se está analizando un caso 2D, solo es necesario hacer un seguimiento de los beads en el plano y para ello es suficiente con la microscopía de fluorescencia[8]. Para TFM 3D, se han utilizado diferentes métodos de imagen para registrar las posiciones tridimensionales de los beads y de la superficie celular dentro de una ECM, intentando llegar a una solución de compromiso entre la resolución espacial, la resolución temporal, la fototoxicidad y el coste. La microscopía confocal (CM) ha sido el método de imagen principal para TFM 3D, ya que el seccionamiento óptico permitido por la apertura confocal, da como resultado una alta resolución espacial en las tres dimensiones. Esto es particularmente atractivo para obtener imágenes de la superficie celular y de los beads con una alta concentración. Comentar que es recomendable tomar las imágenes con la muestra en el medio y lejos de los límites del gel, evitando así que puedan falsearse los resultados. Sin embargo, cada voxel dentro de un volumen debe ser iluminado individualmente por el láser. Para el caso de la microscopía confocal de barrido láser (LSCM), la luz emitida es captada por un tubo fotomultiplicador de eficiencia cuántica (~10%) que compromete la resolución temporal y la viabilidad celular debido a la fototoxicidad. En la microscopía confocal de disco giratorio (SDCM), la iluminación y grabación de varios píxeles en paralelo reduce estas limitaciones a costa de una resolución espacial reducida. Otra opción es la microscopía de fluorescencia de campo amplio (FM) que se ha usado recientemente para rastrear los desplazamientos de los beads dentro de los geles blando ya sea capturando muchas imágenes bidimensionales para formar una imagen 3D o utilizando un método de imagen desenfocado. La ventaja de la FM es que puede ser rápida y al mismo tiempo limita la fototoxicidad, pero acosta de un seccionamiento óptico deficiente que limita la resolución espacial a lo largo del eje óptico (eje z), especialmente para grandes estructuras homogéneas como una célula marcada con fluorescencia. Como resultado, aún no se ha producido un verdadero campo de tensiones usando FM. Se debe tener en cuenta que la FM suele estar disponible en los laboratorios de biología, por lo que, un enfoque de TFM 3D donde solo se requiera FM podría ser adoptado más ampliamente y sería más rentable en comparación con los métodos que utilizan la microscopía confocal [2].

2.3.1 Microscopía confocal de barrido láser (LSCM)

Como ya se ha mencionado, uno de los métodos utilizados para capturar imagen en TFM 3D es la microscopía confocal de barrido láser o LSCM. Básicamente, con esta técnica lo que se hace es capturar múltiples imágenes bidimensionales a diferentes profundidades en una muestra (proceso conocido como seccionamiento óptico), las cuales permiten la reconstrucción tridimensional de dicha muestra. En la siguiente figura se muestra un esquema de su funcionamiento.

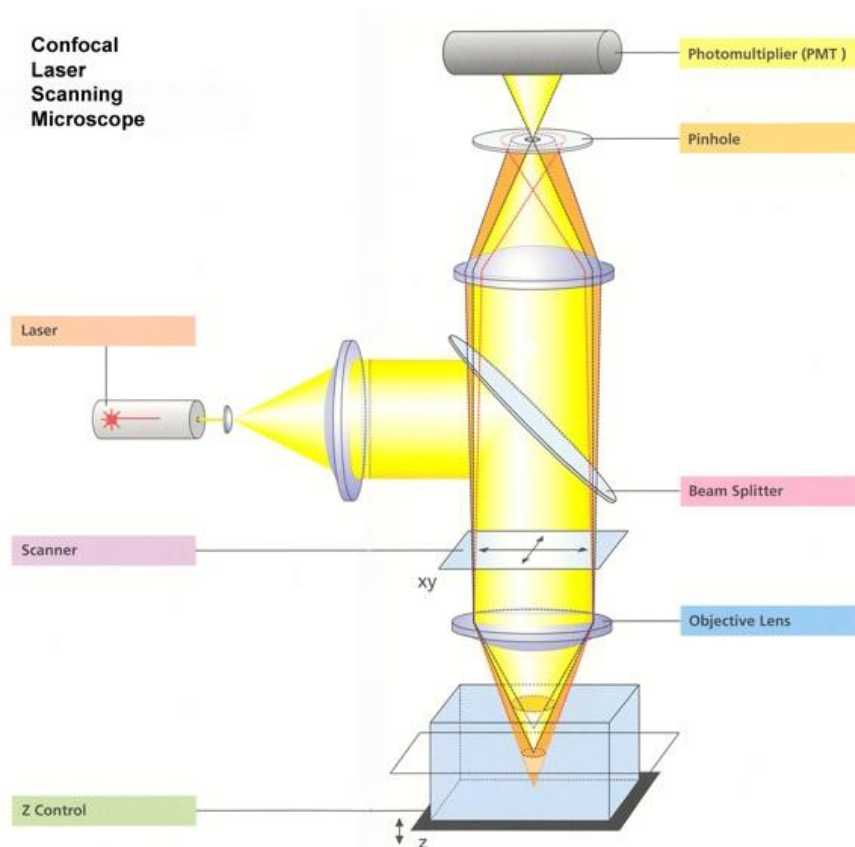


Figura 5. Esquema de un microscopio confocal de barrido láser [9].

En esta tecnología se utiliza un rayo láser, cuya intensidad se ajusta mediante filtros, que es enfocado por la lente del objetivo en un pequeño punto del plano focal y se usa para iluminar o excitar las partículas fluorescentes de la muestra. Un espejo de alta precisión permite una precisa traslación (exploración) del punto de iluminación a través de la muestra para registrar la luz emitida desde todos los puntos de la imagen. Si la muestra es fluorescente, parte de la luz volverá a la lente y viajará hacia atrás haciendo el camino inverso al láser. Esta luz pasa a través de un espejo semitransparente que la refleja lejos del láser y hacia el sistema de detección [9,13].

El primer componente del sistema de detección y el más crítico de un LSCM es el orificio ajustable (pinhole) frente a los detectores, que se encarga de permitir que solo la porción de luz deseada pase a través de él, bloqueando gran proporción de la luz no deseada fuera del foco. Por lo tanto, este orificio genera la confocalidad del sistema. Con esto se logra que la mayor parte de la información lumínica venga del mismo plano focal, lo cual mejora la resolución axial del microscopio. Tras esto, se encuentran los tubos fotomultiplicadores, al final de la trayectoria de la luz, los cuales son los detectores utilizados para amplificar la señal y cuantificar el flujo de fotones emitidos por los beads fluorescentes [9,13].

Comentar que, en TFM, se suele inyectar alguna sustancia en las células, para que al igual que los beads, sean fluorescentes y que por tanto la luz que refleje también lo sea. De esta forma, la luz reflejada por las células y beads será fluorescente y distinta a la luz del láser, pudiendo ser separadas mediante filtros.

2.3.2 Microscopía confocal de disco giratorio (SDCM)

La microscopía confocal de barrido láser es otro de los métodos utilizados para tomar imágenes de la muestra. El procedimiento seguido es similar aunque ligeramente diferente al que se sigue en el microscopio anterior. En la figura 6, se puede ver un esquema de este microscopio, mostrándose a la izquierda la trayectoria del haz de luz en la fase de excitación y a la derecha en la fase de emisión.

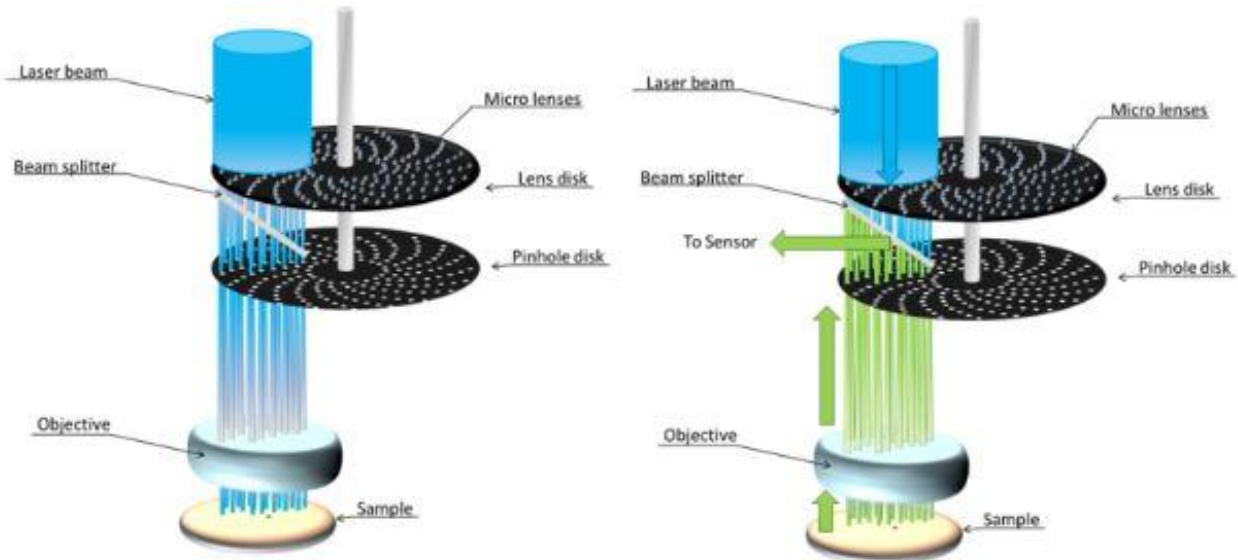


Figura 6. Esquema de un microscopio confocal de disco giratorio [13].

En este tipo, una diferencia respecto al anterior ocurre en la fase de excitación, donde en vez de dirigir un láser enfocado en un solo punto, utiliza uno de los discos giratorios que contienen un patrón de lentes pequeñas para generar múltiples rayos y enfocarlos hacia la muestra. Otra de las diferencias es en la fase de emisión, donde la luz que regresa emitida por la muestra, en lugar de pasar por un solo orificio, pasa a través de muchos repartidos a lo largo del otro disco. Con estos, al igual que en el sistema anterior, excluye aquella luz que viene emitida por puntos fuera del foco y deja pasar aquella que proviene del lugar correcto de la muestra. La rotación de ambos discos con una distribución específica de lentes y orificios genera un patrón de barrido a lo largo de la muestra que genera altas tasas de adquisición de imágenes tanto en 2D como en 3D. Por último, la luz emitida es reflejada por un espejo dicróico en una cámara altamente sensible [13].

2.3.3 Microscopía de fluorescencia de campo amplio (FM)

El microscopio de fluorescencia de campo amplio es capaz de capturar imágenes del plano focal permitiendo después una visualización tridimensional de la muestra apilando las imágenes de las secciones en el eje z. Una diferencia frente a los anteriores es que también detecta cierta luz emitida por la muestra fuera del plano focal por lo que da lugar a imágenes algo borrosas [14]. En la figura 7 se muestra un esquema de este tipo de microscopio.

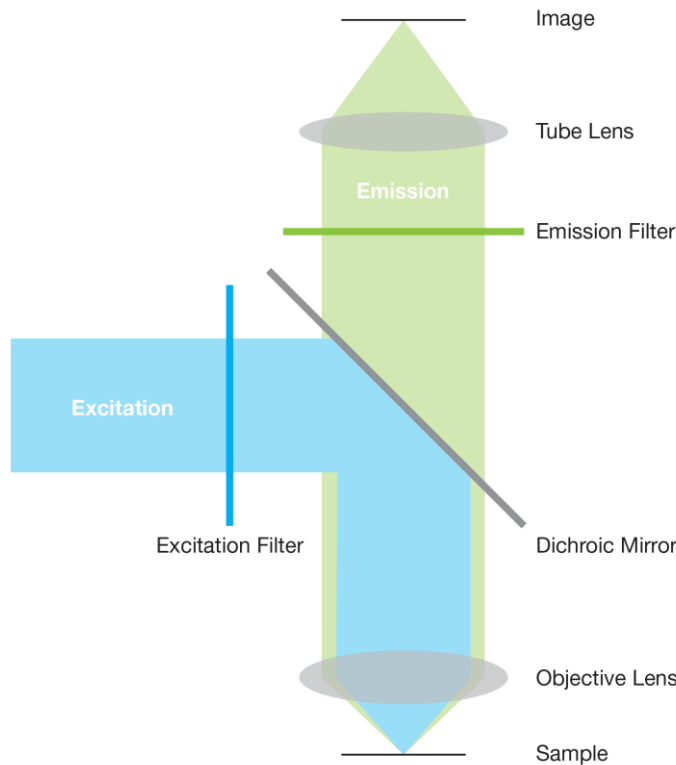


Figura 7. Esquema de un microscopio de fluorescencia de campo amplio [15].

A diferencia de la microscopía confocal, donde la muestra es excitada en uno o múltiples puntos usando uno o varios láseres, en esta técnica de obtención de imágenes, el espécimen entero es expuesto a una luz con una longitud de onda específica para excitar aquellas partes del mismo que son fluorescentes. La trayectoria de la luz es similar a la seguida en cualquier microscopio, donde la luz emitida es llevada a un detector mediante piezas oculares [15].

2.4 Resolución del problema mecánico

Tras la toma de imágenes y la mejora de las mismas, el siguiente paso es calcular el campo de desplazamiento que genera la célula y a partir de dichos desplazamientos obtener el campo de tensiones que la célula aplica sobre su entorno. Este trabajo suele ser hecho por softwares, cuya bases son las que se comentarán a continuación.

El primer paso es localizar los beads en las imágenes tanto en el caso deformado como en el indeformado. Para ello, cada pixel de una imagen tiene asignado un valor de gris como un entero sin signo de 8 bits. Los valores de gris de los marcadores de referencia debido a su fluorescencia son más altos que los del fondo y es este contraste de brillo es la base para localizar los marcadores. Sin embargo, hay que tener en cuenta que el brillo del fondo de una imagen varía a lo largo de ella por lo que no suele ser posible definir un valor límite global de gris para distinguir los beads del fondo. Los enfoques para abordar esto, suelen ser varios y algunos de ellos pueden encontrarse en [8, 10] y en artículos asociados.

Después de localizar todos los marcadores en las imágenes, el siguiente paso es emparejar los del caso indeformado con los del deformado. Los principios básicos que siguen la mayoría de softwares de reconocimiento son los que siguen a continuación. Primero se elige un bead arbitrario s_0 y se conecta con su vecino más cercano s_1 en el caso indeformado generando el segmento l_0 . Para elegir un bead como "vecino", este debe encontrarse a una distancia inferior a otra definida anteriormente como desplazamiento máximo. En

segundo lugar, siguiendo algún criterio específico del software, se detecta en el caso deformado dos conjuntos de beads entre los cuales se pueden encontrar s_0 y s_1 respectivamente. Después de esto, se traza todos los segmentos l_1 entre los distintos pares posibles y usando otro criterio específico se determina si hay alguna coincidencia entre l_0 y los l_1 posibles. Si dicha coincidencia existe, se registra las coordenadas de s_0 y s_1 en ambos casos, y si no es así se repite el proceso eligiendo otro bead como nuevo punto de origen s_0 para construir l_0 [4, 7, 10].

Una vez identificado los beads tanto en la imagen para el caso deformado como para el indeformado, ya se está en condiciones de obtener el campo de desplazamientos. Para ello, simplemente lo que se hace es la diferencia entre las coordenadas de cada marcador en el caso deformado y las del caso indeformado.

$$\vec{u} = \vec{x} - \vec{x}_0$$

Traducir los desplazamientos de los beads fluorescentes a un campo de tensiones, y calcular con este el vector tensión que la célula ejerce a su alrededor, es quizás, el paso más crítico en TFM 3D. Tras esta breve explicación, se van a exponer las técnicas más utilizadas para conseguir el campo de desplazamiento a partir de las imágenes del caso deformado e indeformado y se explicarán y compararán los dos principales métodos de computación utilizados para conseguir el campo de tensiones, usando dicho campo de desplazamiento.

2.4.1 Técnicas para el cálculo del campo de desplazamientos

Existen varias técnicas para determinar el campo de desplazamientos. Entre ellas, dos de las más importantes son el método de seguimiento de partículas y la correlación de volumen digital (DVC), las cuales se van a explicar a continuación [2].

- **Método de seguimiento de partículas**

Este método rastrea el desplazamiento de los beads fluorescentes individuales, convierte los desplazamientos discretos en un campo de desplazamiento continuo usando un algoritmo de interpolación y luego deriva para encontrar el campo de deformaciones [2]. Un esquema de este método de seguimiento de beads fluorescentes individuales a lo largo del tiempo puede verse en la figura 8 aunque en 2D, debido a la incapacidad de mostrar dicho esquema en 3D.

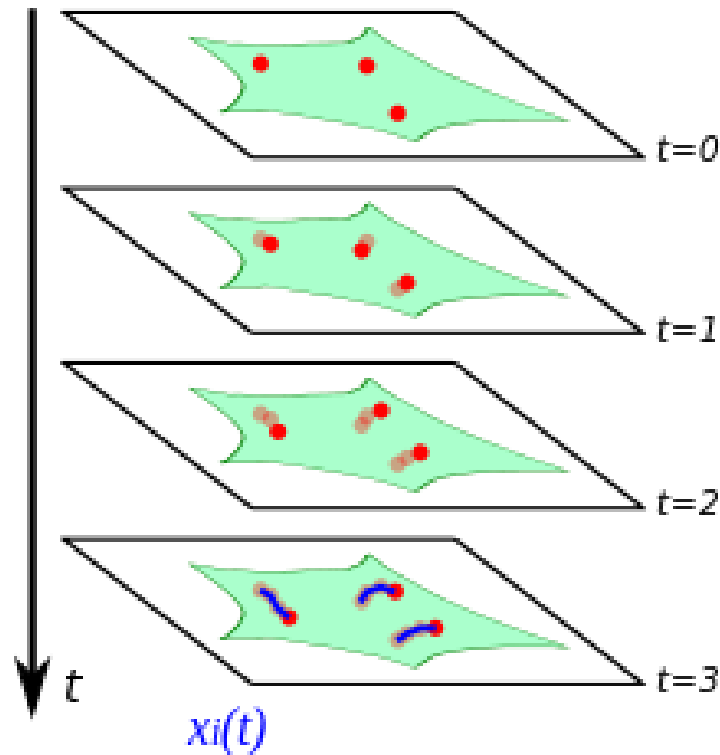


Figura 8. Esquema del método de seguimiento de partículas individuales [23].

Primero, se rastrean los marcadores de referencia fluorescentes para calcular sus desplazamientos y esto se puede hacer utilizando algoritmos como el vecino más cercano, algoritmos basados en el volumen, basados en vectores o algoritmo de regresión automática. La elección de un algoritmo de seguimiento de beads llega a ser importante en experimentos donde el desplazamiento de los beads es grande en relación con el espaciado entre los mismos. En segundo lugar, interpolar los desplazamientos de un conjunto de beads distribuidos al azar en un campo de desplazamiento continuo es una tarea difícil. En [22], los autores conectaron los beads en una malla 3D de elementos tetraédricos y se usó interpolación lineal dentro de cada elemento para calcular la energía de deformación. Hay cuestiones que tienen que ser abordadas antes de que un campo de tensión se pueda evaluar con precisión usando este método de interpolación:

- 1) Puede haber múltiples mallas 3D para conectar el mismo conjunto de beads, y no está claro como eso afectaría a la interpolación.
- 2) Debido a que el desplazamiento se interpola linealmente en cada elemento, su gradiente, o el campo de deformación, es discontinuo entre los elementos adyacentes, lo que puede limitar la resolución del campo de deformaciones y tensiones, especialmente cuando el espaciado medio entre los beads sea grande.

Recientemente, se ha propuesto utilizar un método de mínimos cuadrados móviles (MLS) para traducir los desplazamientos discretos de los beads a un campo de desplazamiento continuo. Las principales ventajas de MLS sobre otros métodos de interpolación (por ejemplo interpolación lineal) son que es un método de interpolación sin malla y que puede producir un campo de desplazamiento infinitamente derivable, asegurando así un campo de deformación suave. En resumen, primero se representa el campo de desplazamiento mediante una función polinómica 3D. A diferencia del método convencional en el que los coeficientes se establecen como constantes, en el método MLS se supone que estos coeficientes varían de un punto a otro, lo que da a la función polinómica una flexibilidad suficiente para representar un campo complicado. En cada punto de interpolación, los coeficientes polinómicos se determinan mediante un método de mínimos cuadrados ponderados en el cual los beads más cercanos al punto de interpolación tienen un mayor peso que los que están

más lejos. El campo de deformación puede ser fácilmente calculado derivando el campo de desplazamiento interpolado gracias a su suavidad. Este método, ha sido utilizado con éxito para mapear el campo de tracciones bajo un indentador (punzón), augurando buenos resultados para futuras aplicaciones en el mapeo del campo de tensiones producidas por una única célula introducida dentro de geles de colágeno [2].

- **Correlación de volumen digital (DVC)**

El método de correlación de volumen digital (DVC) es una extensión 3D del método de correlación de imagen digital (DIC) ampliamente utilizado en mecánica experimental de sólidos. El principio básico de funcionamiento del método DVC, es dividir primero el dominio de la imagen en subvolúmenes y luego correlacionar dichos subvolúmenes antes y después de la deformación mediante la comparación de patrones de imagen. Esto se observa en la figura 9.

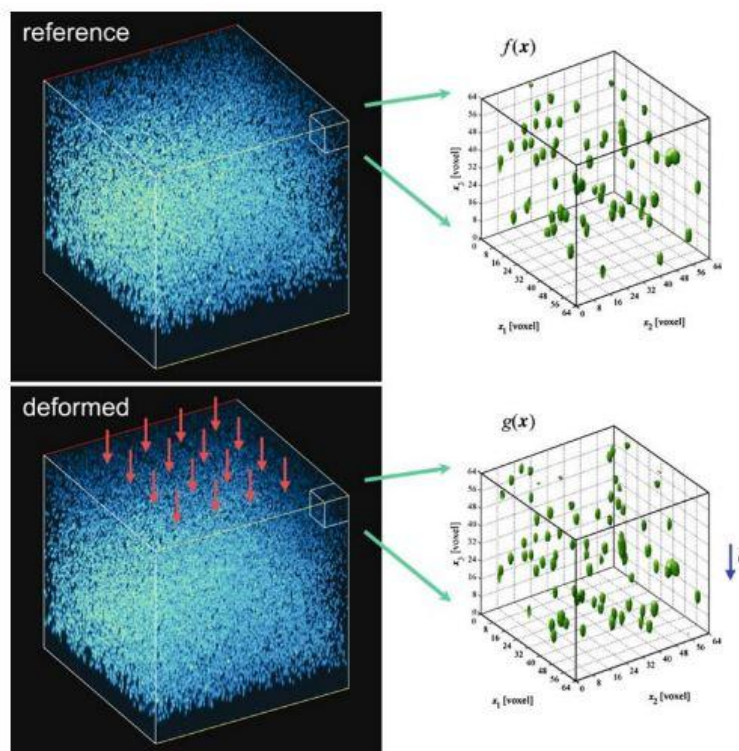


Figura 9. Esquema de DVC [10].

El proceso de correlación se realiza mediante la minimización de una función objetivo que cuantifica la coincidencia entre los subvolúmenes antes y después de la aplicación de carga. Se supone que cada subvolumen solo sufre un translación de sólido rígido. Los campos de deformación se pueden determinar primero suavizando localmente el campo de desplazamiento y luego evaluando numéricamente su gradiente. Recientemente, el método DVC se ha aplicado a geles suaves para obtener imágenes con una alta densidad de beads distribuidos aleatoriamente, las cuales proporcionan el patrón de imagen 3D. Dado que los geles pueden sufrir grandes deformaciones, la elongación y la rotación de los subvolúmenes individuales deben tenerse en cuenta al correlacionar los subvolúmenes. Esto fue ligeramente considerado en [11], ya que se incluyó la elongación de dichos subvolúmenes en el algoritmo de correlación, pero se obvió la rotación [2].

2.4.2 Método de cálculo inverso

Este método fue introducido por primera vez en [16] en TFM 2D. Aquí se modeló el sustrato de gel como un semiespacio infinito de un sólido homogéneo, isótropo y elástico lineal y se usó un enfoque de función de Green para derivar la relación entre las tracciones celulares y los desplazamientos. A causa de la geometría del semiespacio, la solución de la función de Green (o campo de desplazamiento causado por una fuerza puntual) se puede obtener de forma analítica (conocida como solución clásica de Boussinesq) de la teoría de la elasticidad lineal. Para un material isótropo solo son necesarios dos parámetros del material como son el módulo de Young y el coeficiente de Poisson.

Recientemente, este método de cálculo inverso se extendió a TFM 3D en [17]. Una complicación de esta extensión proviene del límite irregular entre la célula y el gel circundante, ya que no se dispone fácilmente de una solución analítica de la función de Green en 3D. Este problema se resolvió empleando exhaustivas simulaciones de elementos finitos para obtener numéricamente la función de Green en 3D. Esta función calculada de forma numérica se usa para formar un conjunto de ecuaciones lineales que relacionan los desplazamientos medidos de los beads con la tracción celular desconocida. Resolviendo estas ecuaciones usando un algoritmo de cálculo inverso, hace que se llegue a las tracciones celulares. Sin embargo, esto no es trivial ya que este sistema de ecuaciones suele estar mal condicionado incluso en el caso de TFM 2D como se señala por ejemplo en [18,19]. Esto significa que el resultado del campo de tracciones celulares podría no ser único y además puede ser sensible a errores en la medida en los desplazamientos de los beads. Para garantizar la robustez de las medidas de las tracciones, se han desarrollado procedimientos de regularización adicionales como en [17,19]. Hace poco, se propuso un nuevo enfoque utilizando la ecuación adjunta para el problema inverso [20,21]. En este método, el problema inverso se transforma en un conjunto de ecuaciones acopladas de las desconocidas tracciones celulares y su estado adjunto. Como resultado, no requiere la evaluación de la función de Green y esto puede llevar a alcanzar una mejor eficiencia computacional para TFM 3D [2].

2.4.3 Método de cálculo directo

Este método ofrece un camino diferente desde los desplazamientos de los beads hasta el vector tensión que aplica la célula. Lo primero que hay que hacer es determinar un campo de desplazamiento continuo a partir de las imágenes de los beads antes y después de la aplicación de la carga y entonces calcular directamente las deformaciones y luego el campo de tensiones [2]. El diagrama de flujo de este método se muestra a continuación.

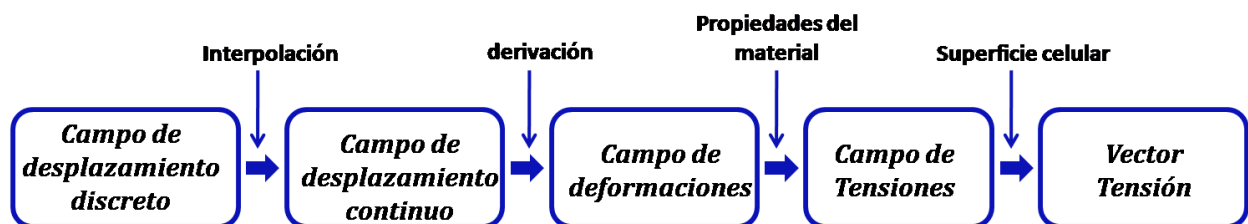


Figura 10. Diagrama de flujo del método directo [2].

Tal como se aprecia en la figura 10, el campo de desplazamiento discreto obtenido, se convierte en un campo continuo mediante un método de interpolación. Después de esto, el campo de deformación se calcula evaluando numéricamente el gradiente espacial del campo de desplazamiento. Seguidamente, el campo de tensiones es calculado a partir de la deformación utilizando las propiedades mecánicas conocidas de la biomatriz junto con un modelo constitutivo (elástico lineal, elástico no lineal, etc.). Por último, el vector tracción se consigue usando el campo de tensiones en el entorno de la célula y el vector normal exterior a la superficie celular en cada punto.

Para terminar con este apartado, remarcar que en el método de cálculo directo, ya sea utilizando la correlación de volumen digital (DVC) o el seguimiento de partículas individuales, el campo de deformación se determina directamente a partir de los datos experimentales de las posiciones de los beads sin ninguna suposición sobre las propiedades mecánicas del gel. Una vez que se tiene el campo de deformación, el siguiente paso es calcular el campo de tensiones introduciendo las deformaciones en un modelo constitutivo apropiado que describa con precisión el comportamiento mecánico del gel. Con el campo de tensiones, σ , las tracciones celulares \vec{T} , se pueden conseguir haciendo $\vec{T} = \sigma \cdot \vec{n}$, donde \vec{n} es el vector normal exterior a la superficie de interés [2].

2.4.4 Método de cálculo inverso vs Método de cálculo directo

El límite irregular entre la célula y la ECM puede plantear desafíos tanto para los métodos de cálculo directo como inverso. El método inverso actual basado en la función Green, cuando es aplicado en TFM 3D, requiere simulaciones de elementos finitos para construir numéricamente la función de Green 3D, tal como se muestra en [17]. Dichas simulaciones de elementos finitos se realizan específicamente para una geometría dada de la frontera célula-ECM. Si la célula cambia activamente su forma, la frontera cambia y se necesita un nuevo conjunto de simulaciones. Para algunas ocasiones, por ejemplo, para células en migración, esto puede llevar a un alto coste computacional y, por lo tanto, a una menor eficiencia en el cálculo del campo de tracciones. Esta dificultad se puede sortear en el método inverso basado en la ecuación adjunta y en el método de cálculo directo. En el método directo, las componentes de deformaciones y tensiones en la frontera de la célula-ECM se interpolan principalmente a partir de los desplazamientos de los beads en la vecindad de la superficie celular. Por lo tanto, la calidad de las medidas de tracciones celulares depende de la densidad de las perlas incrustadas cerca de la célula y de la capacidad de medir con precisión sus desplazamientos. Este no es el caso de los métodos inversos que pueden utilizar los desplazamientos de todas las beads para resolver la tracción celular.

Otra cuestión a tener en cuenta, es que los métodos inversos desarrollados hasta ahora, tanto los basados en la función de Green o como en la ecuación adjunta, depende de la hipótesis de elasticidad isotropa y lineal. Esto es crítico porque la mayoría de los geles biológicos no son lineales, tienden a desarrollar anisotropía cuando se deforman y tienen otras propiedades complejas del material. La característica más importante del método directo es que no se basa en un modelo de material en particular, como elasticidad lineal, para obtener el campo de deformaciones y por lo tanto puede aplicarse cuando aparecen grandes deformaciones y/o un comportamiento complejo de material. Además el método inverso requiere algoritmos de computación complejos que no son fácilmente accesibles para la comunidad biológica. Estas ventajas aún no se han aprovechado, ya que todas las tecnologías de microscopía de tracción de células 3D actuales han utilizado hasta la fecha un modelo de material elástico isotropo lineal [2].

3 RESOLUCIÓN DEL PROBLEMA MECÁNICO CON ELEMENTOS FINITOS USANDO UN MÉTODO DIRECTO

Este nuevo capítulo está dividido en dos partes bien diferenciadas. En una primera parte, se va a exponer la formulación matemática del problema mecánico asociado a TFM 3D con elementos finitos usando el método directo y después se explicará el proceso de resolución realmente seguido en este estudio. En la segunda parte, se va a describir el proceso seguido durante los meses de trabajo. Se hablará de la geometría utilizada, del código desarrollado para mallar dicha geometría haciéndose hincapié en sus partes más importantes y por último se mostrarán los distintos casos de mallados considerados para realizar el estudio.

3.1 Formulación y resolución.

3.1.1 Formulación

En este primer apartado, tal y como se ha mencionado antes, se va a hacer un repaso de la formulación matemática utilizada para resolver el problema mecánico con el método directo usando el método de elementos finitos.

Pero, ¿qué es el método de elementos finitos y cómo funciona? El método de elementos finitos es un método numérico de resolución de problemas de Mecánica de Sólidos que resulta de gran importancia por su utilidad práctica. Este método no proporciona la solución exacta a un problema dado, sino que posibilita llegar a una solución aproximada que es más que suficiente para la resolución de un problema práctico [28].

Este método, lo que hace en líneas generales, es dividir la geometría de la pieza sometida a cargas y restricciones, en partes más pequeñas conocidas como elementos, los cuales están conectados entre sí por puntos llamados nodos. Al conjunto de elementos y nodos se le denomina malla. De esta forma se está transformando un cuerpo de naturaleza continua en un modelo discreto aproximado. Por tanto, se pasa de un problema que tiene un conjunto ilimitado de incógnitas, ya que estas son funciones matemáticas, a un problema donde el conjunto de incógnitas es limitado, pues son el valor de las funciones matemáticas anteriores en los nodos. Todo esto, hace que como ya se ha comentado, las ecuaciones matemáticas que rigen el comportamiento físico no se resuelvan de manera exacta sino aproximada. La precisión de este método dependerá de la cantidad de nodos y elementos, del tamaño y de los tipos de elementos usados para dividir el dominio. Por lo tanto, cuanto menor sea el tamaño y mayor el número de elementos existentes, más se parecerá la solución del método de elementos finitos a la solución real hipotética.

En el problema que ocupa, tras las fases de experimentación, microscopía y la aplicación del algoritmo correspondiente a las imágenes antes y después de la aplicación de la carga, se llega a un campo de desplazamientos en el dominio que rodea a la célula, el cual es a nivel de vóxeles. El siguiente paso sería mallar dicho dominio de una manera adecuada que permitiera resolver el problema de la forma más precisa posible y sin que produzca un coste computacional excesivo. Una vez se tiene la malla, lo que hay que hacer es un proceso de interpolación para pasar ese campo de desplazamiento a nivel de vóxeles a los nodos de la malla de elementos finitos. La interpolación de la que se habla, puede ser un problema más o menos difícil de resolver, pero es resoluble y como no es el objeto de este trabajo no se le va a dar mayor importancia.

De esta forma, ya se tiene el valor del campo de desplazamiento en los nodos de la malla de elementos finitos, y es posible aplicar las expresiones que se mostrarán a continuación. Comentar que para realizar el siguiente desarrollo se ha utilizado [29].

Por lo tanto, el campo de desplazamiento del dominio que rodea a la célula, se puede representar de manera aproximada como:

$$\vec{u}(x, y, z) = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} N_1 u_1 + N_2 u_2 + N_3 u_3 + N_4 u_4 + \dots + N_{nn} u_{nn} \\ N_1 v_1 + N_2 v_2 + N_3 v_3 + N_4 v_4 + \dots + N_{nn} v_{nn} \\ N_1 w_1 + N_2 w_2 + N_3 w_3 + N_4 w_4 + \dots + N_{nn} w_{nn} \end{pmatrix} = \begin{pmatrix} \sum_i^{nn} N_i(x, y, z) u_i \\ \sum_i^{nn} N_i(x, y, z) v_i \\ \sum_i^{nn} N_i(x, y, z) w_i \end{pmatrix} \quad (1)$$

Donde nn es el número de nodos total de la malla, u_i , v_i y w_i son el valor de los desplazamientos en los nodos en las tres direcciones del espacio y $N_i(x, y, z)$ son las funciones de forma de cada nodo. Las funciones de forma no son más que funciones que sirven para interpolar el valor del desplazamiento en cada nodo. Estas funciones dependen del tipo de elemento que se utilice para mallar, pero lo que si se cumple siempre es que tienen que ser iguales a 1 cuando son evaluadas en el nodo asociado y 0 cuando son evaluadas en cualquier otro nodo.

Poniendo la expresión (1) de forma más compacta:

$$\vec{u}(x, y, z) = \sum_i^{nn} \mathbf{N}_i(x, y, z) \mathbf{a}_i \quad (2)$$

$$\text{siendo: } \mathbf{N}_i = \begin{pmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{pmatrix} \quad \text{y} \quad \mathbf{a}_i = \begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix} \quad (3)$$

donde \mathbf{a}_i , es el vector desplazamiento del nodo i . Con la teoría de la elasticidad tridimensional, se sabe que el vector deformación en un punto está definido por seis componentes $\epsilon = [\epsilon_x, \epsilon_y, \epsilon_z, \gamma_{xy}, \gamma_{xz}, \gamma_{yz}]^T$, siendo cada una de ellas:

$$\epsilon(x, y, z) = \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \end{pmatrix} \quad (4)$$

Si se introduce (1) en (4) se llega a lo siguiente:

$$\epsilon(x, y, z) = \sum_i^{nn} \begin{pmatrix} \frac{\partial N_i}{\partial x} u_i \\ \frac{\partial N_i}{\partial y} v_i \\ \frac{\partial N_i}{\partial z} w_i \\ \frac{\partial N_i}{\partial y} u_i + \frac{\partial N_i}{\partial x} v_i \\ \frac{\partial N_i}{\partial z} u_i + \frac{\partial N_i}{\partial x} w_i \\ \frac{\partial N_i}{\partial z} v_i + \frac{\partial N_i}{\partial y} w_i \end{pmatrix} = \sum_i^{nn} B_i a_i \quad (5)$$

siendo B_i la matriz de deformación del nodo i , con

$$B_i = \begin{pmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \end{pmatrix} \quad (6)$$

Llegado a este punto, ya es posible conocer el campo de deformaciones en un punto cualquiera del dominio o matriz extracelular. Para conocer las tensiones en ese punto cualquiera, habría que suponer un modelo constitutivo de material y aplicárselo a las deformaciones. Es en este momento, donde se hace necesario haber caracterizado el comportamiento mecánico del gel o material que compone el volumen en el cual está encerrada la célula.

$$\sigma(x, y, z) = D\epsilon = D \sum_i^{nn} B_i a_i \quad (7)$$

Normalmente las matrices extracelulares o geles suelen tener comportamientos complejos, pero aquí, para ilustrar el proceso, se supone elasticidad isotropa. Por lo tanto, la matriz constitutiva, D , viene dada por:

$$D = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{pmatrix} \quad (8)$$

Por último, para conocer el vector tracción que la célula ejerce en un punto de su superficie se utiliza la expresión (9).

$$\vec{T}(x, y, z) = \sigma \cdot \vec{n}(x, y, z) \quad (9)$$

donde \vec{n} es la normal exterior a la superficie celular. Para poder seguir el proceso descrito y obtener el vector tracción en un punto cualquiera, lo único que haría falta es conocer las funciones de forma. Como ya se ha mencionado, estas dependen del tipo de elemento usado para mallar el dominio. En este trabajo, tal como se comentará posteriormente, solo se van a usar elementos tetraédricos, en concreto, de 4 y 10 nodos. A continuación, se mostrará la manera de obtener las funciones de forma para elementos tetraédricos de 4 nodos. Para el tetraedro de 10 nodos, se obtienen de forma similar y puede ser consultado en [29].

Para el caso del elemento tetraedro de 4 nodos, tal como se ve en la figura 14, estos cuatro nodos definen una variación lineal de los desplazamientos en las coordenadas xyz de manera que se puede escribir:

$$\left. \begin{aligned} u &= \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z \\ v &= \alpha_5 + \alpha_6 x + \alpha_7 y + \alpha_8 z \\ w &= \alpha_9 + \alpha_{10} x + \alpha_{11} y + \alpha_{12} z \end{aligned} \right\} \quad (10)$$

Las constantes α_i se obtienen sustituyendo adecuadamente las coordenadas de los nodos e igualando los desplazamientos a sus valores nodales. Como se utiliza la misma aproximación para todos los desplazamientos, basta con calcular las cuatro constantes para una sola de las componentes del desplazamiento. Así, considerando la componente u:

$$\left. \begin{aligned} u_1 &= \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1 + \alpha_4 z_1 \\ u_2 &= \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2 + \alpha_4 z_2 \\ u_3 &= \alpha_1 + \alpha_2 x_3 + \alpha_3 y_3 + \alpha_4 z_3 \\ u_4 &= \alpha_1 + \alpha_2 x_4 + \alpha_3 y_4 + \alpha_4 z_4 \end{aligned} \right\} \quad (11)$$

donde u_1, u_2, u_3 y u_4 son las componentes en la dirección x del desplazamiento de cada uno de los nodos de un elemento genérico. Resolviendo el sistema (11) y sustituyendo los valores de α_i encontrados, en (10), se llega tras ordenar términos a:

$$u = \sum_i^4 \frac{1}{6V^{(e)}} (a_i + b_i x + c_i y + d_i z) u_i \quad (12)$$

de donde se deduce por comparación con (2) que la función de forma del nodo i es:

$$N_i = \frac{1}{6V^{(e)}} (a_i + b_i x + c_i y + d_i z) \quad (13)$$

siendo $V^{(e)}$ el volumen del elemento, y

$$a_i = \begin{vmatrix} x_j & y_j & z_j \\ x_k & y_k & z_k \\ x_l & y_l & z_l \end{vmatrix} ; \quad b_i = \begin{vmatrix} 1 & y_j & z_j \\ 1 & y_k & z_k \\ 1 & y_l & z_l \end{vmatrix} \quad (14)$$

$$c_i = \begin{vmatrix} x_j & 1 & z_j \\ x_k & 1 & z_k \\ x_l & 1 & z_l \end{vmatrix} ; \quad d_i = \begin{vmatrix} x_j & y_j & 1 \\ x_k & y_k & 1 \\ x_l & y_l & 1 \end{vmatrix}$$

Llegados a este punto ya se podría aplicar el método directo de elementos finitos para elementos tetraédricos de 4 nodos.

3.1.2 Resolución

En este apartado se va a explicar cuál ha sido el proceso de resolución usado para abordar este estudio, que respecto al procedimiento de la metodología TFM 3D, es un poco distinto. En TFM 3D, como ya se ha explicado, se parte de unos desplazamientos a nivel de vóxeles adquiridos mediante la aplicación de un algoritmo a las imágenes tomadas del caso deformado e indeformado.

Aquí surge el problema, también ya comentado, de la interpolación usada para pasar ese campo de desplazamientos a nivel de vóxeles a un campo de desplazamientos en los nodos de la malla de elementos finitos. A esto, hay que sumarle además que el volumen que encierra la célula en los experimentos, que es donde se calcularía el campo de desplazamiento real que la célula provoca, no tiene por qué corresponderse exactamente con el volumen que encierra la célula en la malla de elementos finitos generadas con el código que se propone en este trabajo. Esto se debe a las capas, ya que para crearlas y que no se salieran del paralelepípedo que envuelve la célula, ha sido necesario extenderlo en algunas direcciones del espacio. Por lo tanto, esto hace más difícil si cabe, ese paso de interpolación de los desplazamientos.

Para evitar esta dificultad, la cual se dejará para trabajos posteriores, y como el objeto de este estudio no es exactamente resolver un problema de TFM 3D, sino analizar la influencia que pueda tener la mejora del mallado (con el enfoque de las capas) en los resultados obtenidos, se ha seguido un proceso de resolución distinto el cual se comenta a continuación.

Lo que se hace en el problema resuelto es un enfoque que sigue un camino inverso al de TFM 3D. Se impone que la célula ejerce unas tracciones cualesquiera en unos puntos aleatorios de su superficie, y de esta forma generará un campo de tensiones, un campo de deformaciones y un campo de desplazamiento en el dominio que la rodea, los cuales serán los resultados a analizar para ver la influencia del mallado realizado. En la figura 11 se muestra estos puntos y el módulo de las tracciones aplicadas en ellos, cuyo máximo es del orden de 30 Pa.

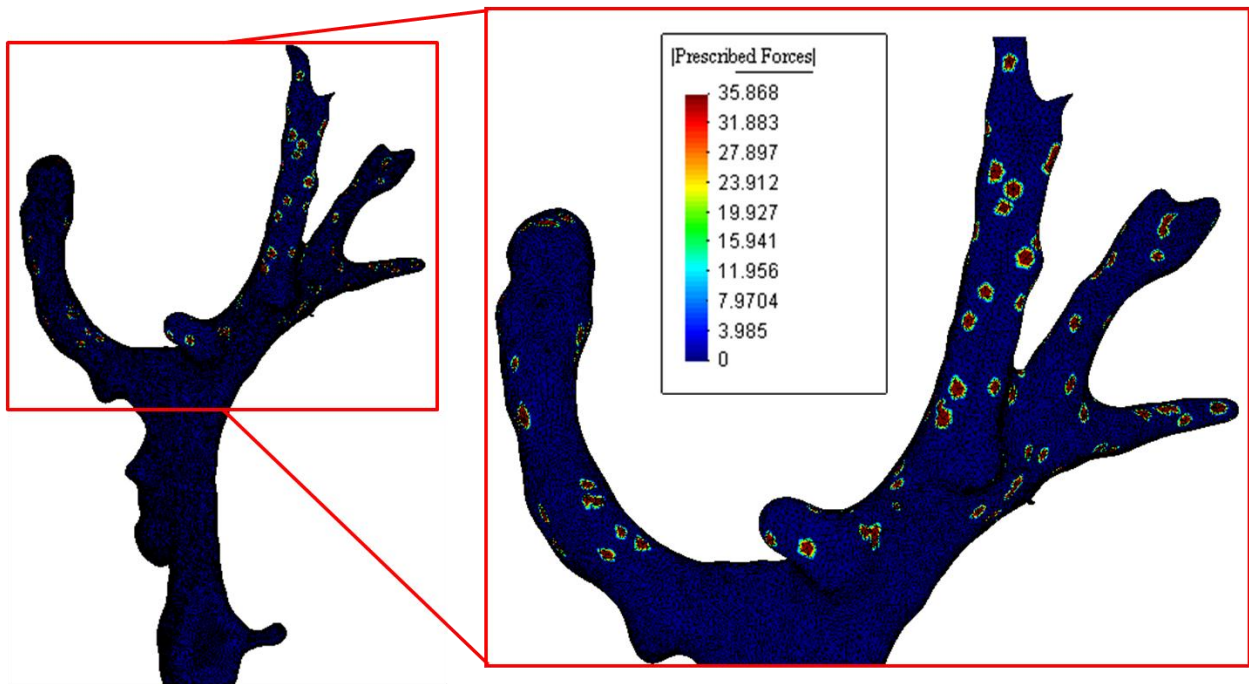


Figura 11. Tracciones impuestas en el entorno de la célula.

Comentar, que esta imposición de tracciones en unos puntos concretos en la superficie celular, no falsean ni restan credibilidad a los resultados, ya que esto es lo que ocurre en la realidad, pues las células ejercen las tracciones en unos puntos localizados de su superficie y no a través de toda la superficie como se podría llegar a pensar.

Por último, una vez se impone estas tracciones, el problema es exportado al software "Abaqus", que es un programa de cálculo por elementos finitos. Este software resolverá el problema y devolverá el campo de desplazamiento y tensiones generadas por las tracciones impuestas, y nos permitirá evaluar la bondad o no del mallado realizado.

3.2 Trabajo Desarrollado.

A grandes rasgos, lo que se persigue con este trabajo es realizar un mallado del entorno de la célula que mejore la captación de los resultados. Para ello, la idea es generar dos capas en el dominio del gel alrededor de la célula, de forma que en dichas capas la malla sea más fina, es decir que los elementos sean más pequeños en estas zonas cercanas a la célula, que al fin y al cabo son las más importantes. Así, la transición del tamaño de los elementos será más suave, lo que permitirá captar mejor el gradiente de las tracciones. Un esquema de esta enfoque es mostrado en la figura 12 donde también se señala los parámetros de entrada al código generado en este proyecto, los cuales serán explicados más adelante. Con esta idea, se realizará un trabajo de análisis modificando una serie de parámetros de entrada de la malla, como pueden ser el tipo de elemento utilizado, o los espesores de de las capas. En los siguientes apartados, además de desarrollar las ideas generales, se comentarán los aspectos concretos más importantes del código que permiten llevar este trabajo a cabo.

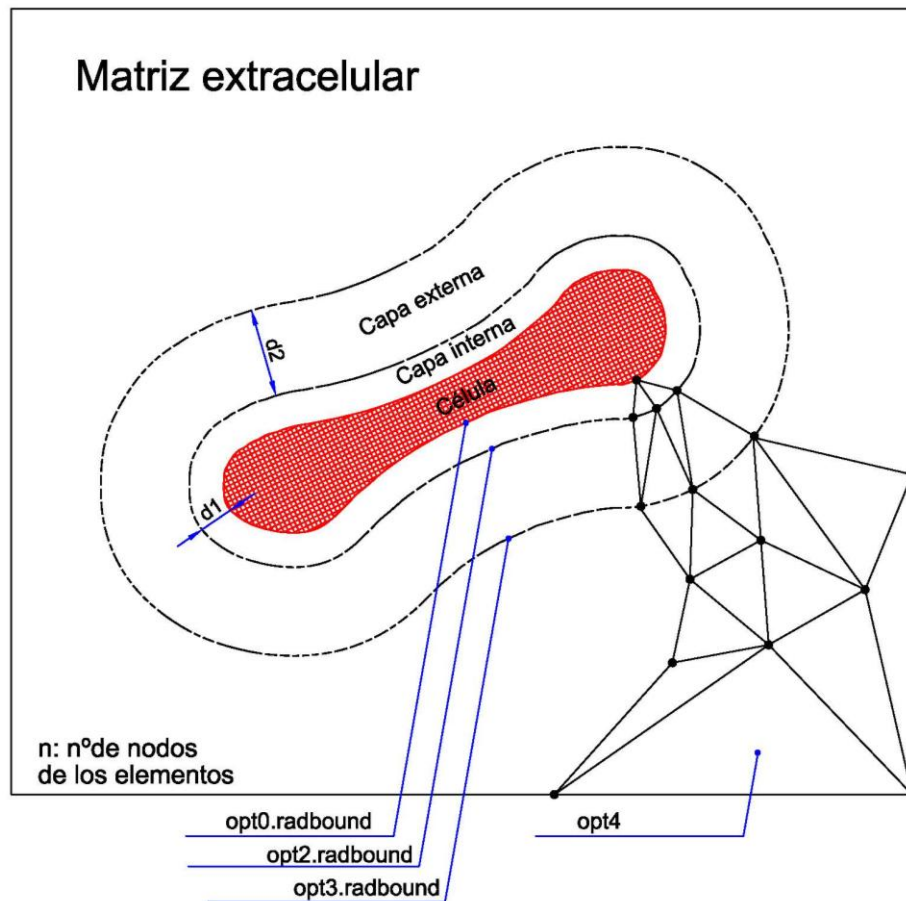


Figura 12. Esquema de la idea general.

3.2.1 Geometría

Como es evidente, en TFM 3D la geometría utilizada es la de una célula concreta encerrada dentro de un material biológico, que simule las condiciones en las que se encuentran las células en la realidad. Como la finalidad de este trabajo no era realizar ningún tipo de experimentación biológica ni ningún proceso de microscopía, la geometría fue proporcionada por el tutor de algún trabajo realizado por él anteriormente. Esta geometría fue proporcionada en un formato .mat, que está asociado al software de cálculo "matlab", que es donde se ha realizado el código que permite llevar a cabo este estudio. La geometría se encontraba en forma de hipermatriz binaria de vóxeles, que no es más que una matriz tridimensional de ceros y unos donde la posición en la que hay un cero, hace referencia al gel que rodea la célula y la posición donde aparece un uno se refiere a la célula en sí.

Para mostrar tanto la geometría, lo cual será hecho a continuación, como todos los demás resultados visuales se va a utilizar el software gratuito "GID". Este software es un pre y post-procesador de simulaciones numéricas en ingeniería, adaptable y fácil de usar. Con él, se puede realizar modelados geométricos, generación de malla, importación de datos de otros softwares, operaciones de postprocesado o visualización de resultados. En nuestro caso, como ya se ha dejado entrever, nos servirá para visualizar los resultados de malla, desplazamientos, tensiones y tracciones importados desde "matlab" o "abaqus".

Entonces, llevando a cabo una serie de pasos dentro del código, los cuales serán explicados posteriormente y poniendo los datos en un formato adecuado para que pueda ser leído por "GID" se obtiene lo mostrado en la figura 13. En la izquierda se ve un paralelepípedo que hace referencia a la matriz extracelular de gel que recubre a la célula, mientras en la derecha, se puede observar la célula. Esta será la geometría utilizada a lo largo de todo el estudio.

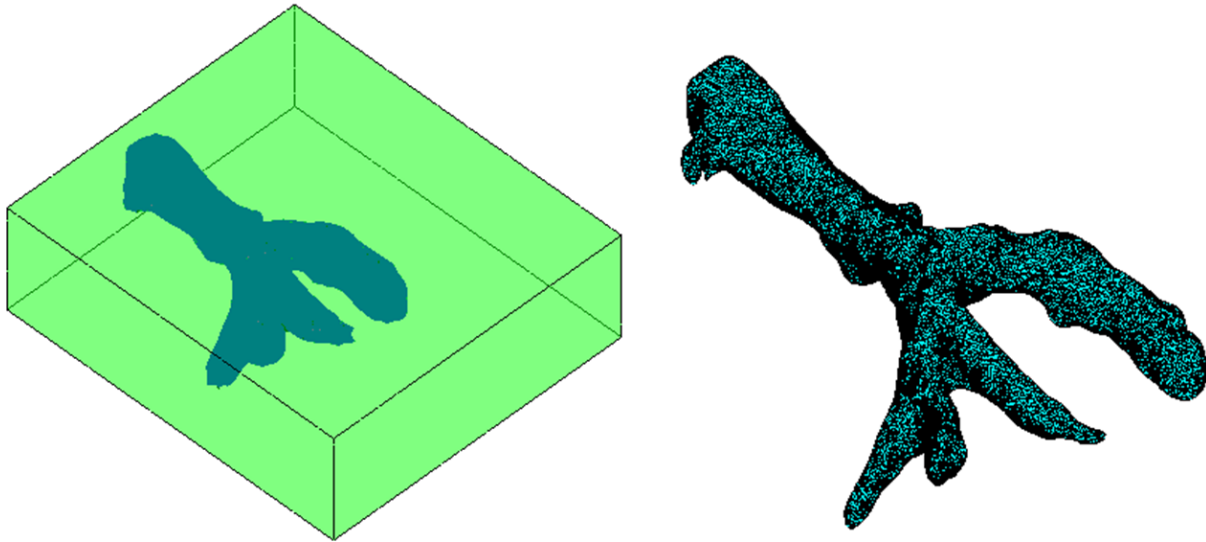


Figura 13. Geometría del problema (*izq. Célula embebida dentro de la matriz extracelular de gel. dcha. Célula aislada.*)

Lo primero que hay que introducir en el código utilizado son unos parámetros de entrada con los que se define la geometría y el tipo de malla que se va a realizar en el dominio. Estas variables de entrada o parámetros de mallado, que se pueden apreciar en la figura 12, van a ser explicadas a continuación:

- n : Esta variable hace referencia el tipo de elemento que va a usar en la malla. En nuestro estudio se ha considerado solo el uso de elementos tetraédricos ya que las mallas compuestas por dichos elementos suelen tener ventajas a la hora de mallar geometrías complejas como es el caso que nos ocupa, frente a las mallas hexaédricas. Las principales ventajas de las mallas tetraédricas frente a las hexaédricas para geometrías complejas, son que suelen emplear menores tiempos, y menores números de elementos. En el código creado, la variable n solo puede tomar el valor 4 y 10 dando lugar a mallas compuestas por elementos tetraédricos de cuatro nodos y diez nodos respectivamente. Ambos tipos de elementos se muestran en la figura 14.

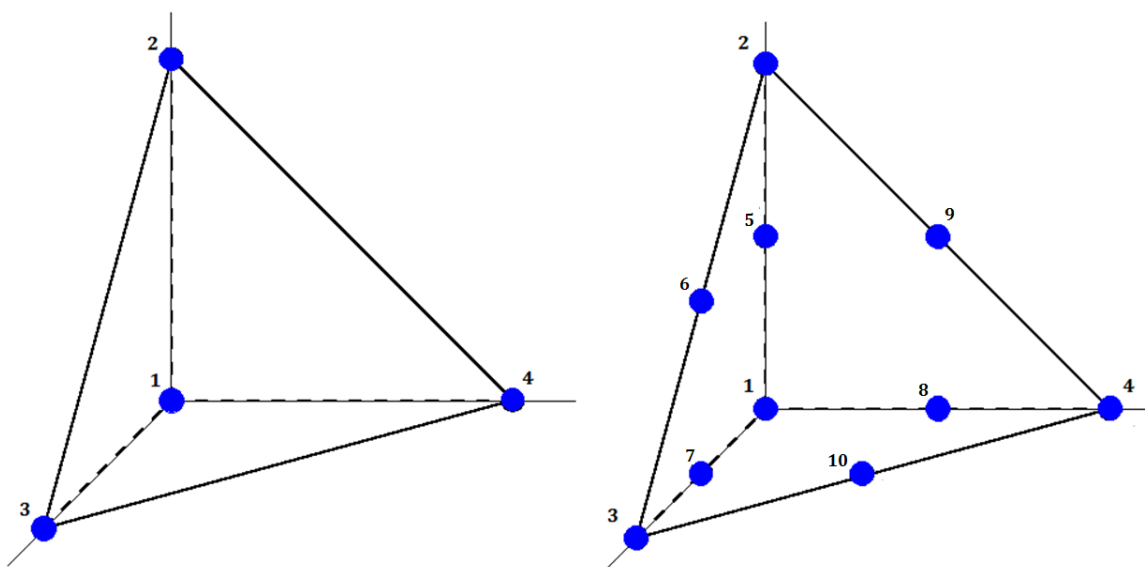


Figura 14. Elementos tetraédricos de 4 y 10 nodos.

- $d1$: Con este parámetro se hace referencia al espesor de la capa interna, es decir, a la capa que se encuentra adyacente a la célula. Se está buscando que tanto este parámetro como el espesor de la capa externa, sean la mínima distancia que permita capturar de la manera más exacta posible las tracciones que la célula ejerce en su entorno. Esto se debe a que si los espesores de las capas aumentan demasiado, el coste computacional aumentará mucho, ya que en estas capas los elementos serán más pequeños que en el exterior de ellas, por lo que habrá más elementos y por tanto más nodos. Si esto ocurre, puede darse el caso de que una malla más fina, no produzca una mejora significativa en los resultados y si un aumento considerable en el coste computacional. Los valores que se le darán tanto a esta variable como al espesor de la capa externa, son a nivel de vóxeles, es decir, representan el número de vóxeles que se le añade en las tres dimensiones del espacio a la superficie de la célula para formar dichas capas.
- $d2$: Representa el espesor de la capa externa, es decir, la capa más exterior que envuelve a la célula. Las cuestiones que afectan a esta variable son las mismas que las que afectan a la variable $d1$.
- `opt0.radbound` : Esta sintaxis en "matlab" quiere decir el campo "radbound" de la estructura "opt0". Con ella lo que se impone es el tamaño máximo del elemento (en este caso cara) que se va a usar para mallar la superficie de la célula. Más concretamente, y aunque se volverá a comentar en un apartado posterior, el valor que se le asigna hace referencia al radio máximo de la circunferencia de Delaunay (triangulación de Delaunay). Esta condición dice que la circunferencia circunscrita de cada triángulo (cara de los tetraedros) de la superficie no debe contener ningún vértice de otro triángulo[25]. Esta condición se puede ver en la figura 15. De nuevo, comentar que el valor que se asigna a esta variable es a nivel de vóxeles (píxeles), ya que esto sirve para "construir" la malla a partir de la hipermatriz de vóxeles en la que se da la geometría del problema.

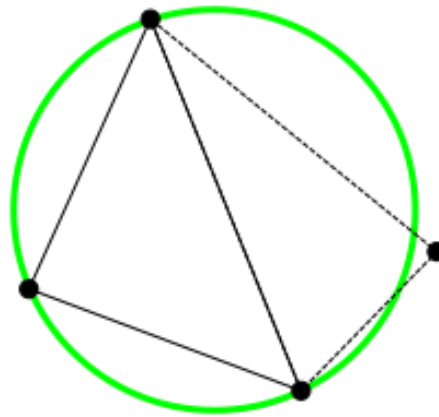


Figura 15. Circunferencia de Delaunay [25].

- `opt2.radbound` : Es igual que el parámetro anterior pero se refiere al tamaño máximo del elemento que se usa para mallar la superficie de la capa interna.
- `opt3.radbound` : Este también es igual que las dos variables anteriores pero refiriéndose al tamaño máximo del elemento que se utiliza para mallar la superficie de la capa externa.
- `opt4` : Este es el último dato de entrada que se le introduce al código y hace referencia simplemente al volumen máximo del elemento que puede usar el código. Esto afectará al dominio de gel que se encuentra fuera de las dos capas ya que ahí no se está imponiendo ningún tamaño y la única restricción que aplica es esta.

Una vez que se han introducido estos datos de entrada, la siguiente acción relevante que se realiza en el código es cargar la geometría mostrada anteriormente y generar las capas. Para la formación de las capas, como la geometría viene dada por una hipermatriz binaria donde las posiciones 1 son célula y las 0 son gel, lo que se hace es convertir todas las posiciones adyacentes de todas las posiciones 1 que se encuentran a una distancia inferior o igual a $d1$ para la capa interior y a una distancia $d1$ más $d2$ para la capa exterior, en posiciones 1, sin importar si antes había un 0 o un 1. Esto se hace en las seis direcciones del espacio. Para no perder información de la superficie de la célula y por temas relacionados con el mallado que se comentarán en un apartado posterior, las hipermatrices resultantes de la ampliación debida tanto a la capa interna como a la capa externa, se guardan en variables distintas.

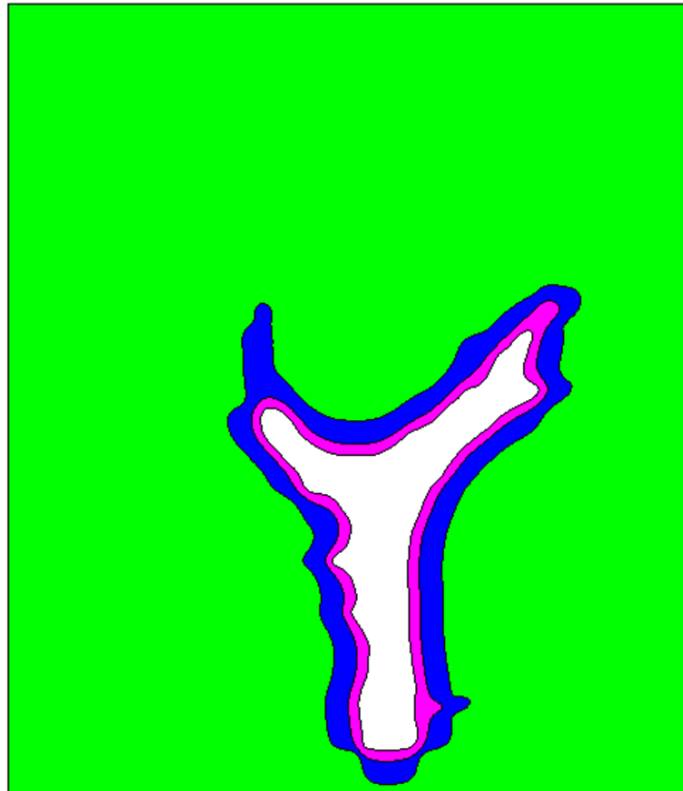


Figura 16. Representación de las capas generadas alrededor de la célula

En la figura 16 se muestra un ejemplo de cómo serían las capas generadas alrededor de la célula para un caso con unos parámetros de mallado cualquiera. Esta imagen es un corte del dominio en el que se puede distinguir el lugar que ocuparía la célula (zona blanca), la capa interna (zona magenta), la capa externa (zona azul) y el resto del gel (zona verde). Para evitar cualquier confusión, destacar que las capas forman parte del gel, es decir, no es célula y además son ficticias ya que son solo un medio para poder generar una malla más fina en esa zona.

3.2.2 Mallado

Una vez se tiene desarrollada la geometría, ya se está en condiciones de mallarla. Para ello en el presente trabajo se han utilizado una serie de funciones que permiten pasar de una matriz binaria de vóxeles a una malla tridimensional de tetraedros.

La mayoría de las funciones usadas se han obtenido del "Iso2mesh" que es un "toolbox" de código abierto para la generación de mallas para los softwares "matlab" o "octave", diseñada para la creación de superficies y

mallas tetraédricas a partir de imágenes volumétricas 3D. Este "toolbox" contiene una gran cantidad de "scripts" y funciones de procesamiento de malla, que funcionan de forma independiente o interactuando con otras funciones de mallado externas. "Iso2mesh toolbox" puede convertir directamente imágenes en 3D como imágenes binarias, segmentadas o en escalas de grises como una resonancia magnética (MRI) o tomografías computerizadas en mallas volumétricas de calidad. Esto lo hace particularmente adecuado para el análisis de datos de imágenes médicas en múltiples modalidades y modelado multifísico [26]. El diagrama de flujo que describe los diferentes caminos a través de los cuales se puede generar mallas se muestra en la figura 17.

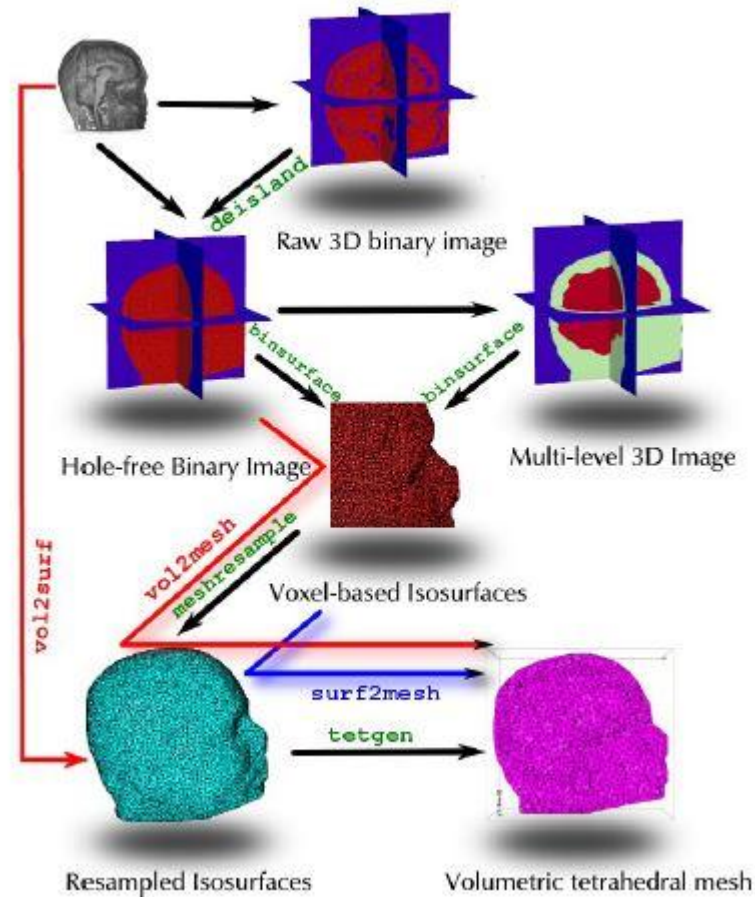


Figura 17. Diagrama de flujo del Iso2mesh [27].

Aclarado que es "Iso2mesh", ya se está en condiciones de explicar cuál ha sido el camino seguido en el código para generar la malla. Tras la generación de la geometría, donde se ha llegado a tres hipermatrices binarias que contienen la información de la célula y de ambas capas, se aplica a dichas matrices por separado la función "vol2surf", es decir, se ejecuta tres veces, una para cada hipermatriz binaria. Esta función lo que hace es detectar la interfase entre las posiciones 0 y 1 de la hipermatriz binaria (imagen volumétrica 3D) y mallar esta interfase con triángulos, de manera que lo que se consigue tras ella es una superficie formada por triángulos. La sintaxis de "vol2surf" es la siguiente:

[no, el, regions, holes] = vol2surf (img, ix ,iy , iz, opt, dofix, method, isovalues)

Las variables de entrada son:

- `img` : En este lugar lo que hay que introducir es la hipermatriz binaria.
- `ix, iy, iz` : Con estos parámetros se indica al tamaño de la hipermatriz introducida. Cada uno de ellos será un vector desde uno hasta el tamaño de la hipermatriz en la dirección correspondiente.
- `method` : Existen dos métodos para mallar la superficie aunque en este informe no se va a entrar en

detalle en las diferencias entre ambos métodos. Comentar que en el código se ha utilizado el método "cgalsurf", ya que aunque se probaron tanto este como el otro método denominado "simplify", se decidió utilizar el primero porque era el que la función utilizaba por defecto. Como no hubo problemas relacionados con esto, no hubo motivos para cambiar de método.

- `opt` : Esta variable sirve para introducirle a la función, opciones distintas a las que ella trae de forma predeterminada. El método "cgalsurf" solo permite modificar las opciones "radbound", que se refiere al radio máximo de la circunferencia de Delaunay, es decir, determina el tamaño máximo del triángulo con el que se mallan las superficies y otra denominada "distbound", que en este código no se ha usado. Es aquí donde se utilizan los parámetros de entrada "opt0.radbound", "opt2.radbound" y "opt3.radbound". Estos tres datos de entrada se utilizan para mallar la superficie de la célula, la superficie de la capa interna y la superficie de la capa externa respectivamente.
- `dofix` : Este parámetro sirve para especificar si se quiere que la propia función actúe para reparar los errores que pudieran ocurrir durante el mallado. En el código se utiliza un 1 que significa que sí.
- `isovalues` : Esta variable se puede entender como una especie de umbral, aunque la realidad es que no se tiene muy claro verdaderamente a que se refiere. En el código se le ha dado el valor de 0.5 simplemente porque tras algunas pruebas se vio que con este valor el resultado era siempre más o menos el esperado.

Ahora se va a comentar cuales son las variables de salida que proporciona esta función:

- `no` : En esta variable se devuelve una lista con las coordenadas x, y, z de los nodos que forman la malla de la superficie.
- `el` : Aquí se proporciona otra lista, pero esta con los tres nodos que forman los elementos triangulares (2D) que forman la mallas superficiales. En una cuarta columna se devuelve un identificador de la región a la que pertenece por sí se mallase varias regiones.
- `regions` : Lista de puntos para toda las subregiones.
- `holes` : Lista de puntos pertenecientes a los agujeros existentes. Tanto este parámetro como el anterior no se utilizan en este código.

Tras aplicar esta función a cada una de las tres hipermatrices obteniéndose tres superficies, el siguiente paso es aplicarle a estas superficies (cada una por separado) la función "smoothpatch". Esta es una función de matlab que sirve para suavizar mallas superficiales formadas por triángulos y es el motivo por el cual la generación de las superficies de la célula, capa interna y capa externa se ha hecho por separado, ya que esta función solo acepta una única superficie a mallar en cada llamada a la misma. La sintaxis de "smoothpatch es la siguiente:

`FV2 = smoothpatch (FV, mode, itt, lambda, sigma)`

Las variables de entrada son:

- `FV` : Esto es una estructura que debe contener el campo "FV.faces" que tiene que ser una lista de los nodos de los triángulos que forman las superficies a suavizar y el campo "FV.vértices" que es una lista de los nodos de la superficie. Las dimensiones de estas listas tienen que ser respectivamente $M \times 3$, donde M es el número de elementos triangulares y $N \times 3$, donde N es el número de nodos. Ambas listas se obtienen de la función "vol2surf" donde "FV.vértices" será la variable "no" y "FV.faces" será las primeras tres columnas de la variable "el".
- `mode` : Esta variable lo que indica es el modo en el que se realiza el suavizado. Se ha usado el modo "1" que es el que viene por defecto.
- `itt` : Especifica el número de iteraciones que se hacen en el suavizado. Se ha usado un valor de 5

iteraciones para asegurar un buen suavizado.

El resto de parámetros de entrada no se han usado en el código. La variables de salida que proporciona esta función de suavizado es:

- FV2 : Esto es de nuevo una estructura que contiene los campos "FV2.vértices" y "FV2.faces" donde se almacenan unas listas con los nuevos nodos y elementos suavizados y que tienen las mismas dimensiones que las listas de entrada a la función "smoothpatch".

Después de obtener los vértices y caras de las superficies suavizadas, lo siguiente es conformar una hipermatriz que guarde la información de la célula y capas por sí misma y unas variables de vértices y caras finales donde se guarden las coordenadas de todos los nodos que conforman las distintas superficies y los nodos que forman las caras de todas las superficies. Para ello, se suma las tres hipermatrices binarias dando lugar a una única hipermatriz donde las posiciones en las que hay un 3 hacen referencia a la célula, las posiciones con un 2 a la capa interna, las posiciones con un 1 a la capa externa y las posiciones con un 0 al resto del gel. Respecto a las variables vértices y caras finales, se obtienen combinando las variables de vértices y caras de las tres mallas superficiales suavizadas tal como se puede ver en el código del anexo, y teniendo en cuenta que hay que modificar los números correspondientes a los nodos para que en las variables vértices y caras finales, los números de los vértices no tengan repeticiones, sean consecutivos y se correspondan entre una variable y otra.

El último paso de gran importancia en el mallado, es la aplicación de la función "surf2mesh", la cual está contenida también en el "Iso2mesh toolbox" y cuya tarea es crear una malla de tetraedros a partir de una malla superficial de triángulos. La sintaxis de esta función es la siguiente:

```
[node, elem, face] = surf2mesh (v, f, p0, p1, keepratio, maxvol, regions, holes, forcebox)
```

Las variables de entrada son:

- v : En este parámetro se introduce la lista de las coordenadas de los nodos que forman todas las mallas superficiales que contengan el volumen que se quiere mallar. La dimensión de esta lista deberá ser por tanto NNx3, donde NN es el número de nodos totales.
- f : Aquí se especifica los nodos de cada elemento triangular o cara que forman las distintas superficies que puedan contener el volumen a mallar. La dimensión de esta lista será de MMx3 donde MM es el número de elementos triangulares.
- p0 : Este parámetro es un vector en el cual se especifica la coordenada de una de las esquinas del paralelepípedo que envuelve a la célula y a las capas.
- p1 : En esta variable se introduce un vector con las coordenadas de otra de las esquinas del paralelepípedo que contiene la célula y las capas. Tanto para este como para el parámetro anterior, se ha tenido especial cuidado para que las coordenadas de las esquinas no difieran demasiado de las coordenadas que tienen dichas esquinas en la hipermatriz que guarda la geometría original. Esto puede ser apreciado en el código que se encuentra en el anexo.
- keepratio : Hace referencia al porcentaje de elementos que son mantenidos tras las simplificaciones. Se utiliza el valor de 1 por lo que este input no afecta.
- maxvol : Esto sirve para definir el máximo volumen de los elementos tetraédricos. Es aquí donde se utiliza la variable de entrada del código "opt4". Como ya se comentó, esta restricción afecta básicamente a los elementos lejanos a las capas, ya que las propias capas cerca de ellas imponen ya un tamaño de elemento y es lejos de ellas donde esta restricción empieza a tener efecto.
- regions : En este input se define el número de regiones que tendrá el volumen a mallar. Esto se especifica introduciéndole las coordenadas de un punto cualquiera de cada una de las regiones

existentes, sin tener en cuenta la región exterior (paralelepípedo), por ejemplo, en el código no se tiene en cuenta la región que conforma el resto del gel exterior a las capas. Esta variable será por tanto una matriz de dimensiones $(R-1) \times 3$, donde R es el número de regiones.

- **holes** : Lista de agujeros existentes en el dominio a mallar. En el caso que ocupa, no se utiliza porque no existe ningún agujero.
- **forcebox** : Esta variable sirve para precisar si se quiere que se cree un paralelepípedo que envuelva el dominio con los datos aportados en las variables "p0" y "p1" o que lo añada de forma automática sin tener esto en cuenta. Se utiliza el valor 1 para use los datos aportados.

Las variables de salida que proporciona esta función son:

- **node** : En este parámetro se devuelve una lista con las coordenadas de todos los nodos que forman la malla de elementos tetraédricos del volumen.
- **elem** : Aquí se guarda una lista con los nodos que componen cada uno de los elementos tetraédricos que forman la malla.
- **face** : En esta variable también se guarda una lista, pero de los nodos que forman cada uno de los elementos superficiales (triángulos) existentes en la malla.

En este punto, ya se ha conseguido una malla volumétrica del dominio, incluida la célula y las capas. Después de esto, lo que se realiza es una serie de identificaciones para a continuación poner dicha malla en un formato adecuado que pueda ser exportado a "GID".

Comentar que las funciones del "Iso2mesh toolbox" crean mallas compuestas de elementos tetraédricos de cuatro nodos. Por lo tanto, si se siguen los pasos y las funciones descritas, lo que se genera es una malla volumétrica de tetraedros de cuatro nodos, es decir, cuando la variable de entrada al código "n" es igual a 4, se haría todo lo explicado anteriormente y tras esto se escribiría la malla en el formato correcto para cuatro nodos. Si lo que se quiere es una malla de tetraedros de diez nodos, lo que hay que hacer es asignarle el valor 10 a la variable "n". Entonces el programa realizará todos los pasos descritos y cuando los haya hecho, colocará nuevos nodos en el punto medio de las aristas de cada tetraedro tal como se puede ver en la figura 14, detectará y eliminará los nodos repetidos intercambiándolos por los ya existentes y escribirá la malla en el formato adecuado para tetraedros de diez nodos. Todo esto, de nuevo se puede observar en el mismo código que es aportado en el anexo.

Tras explicar a grandes rasgos el proceso mediante el cual se generan las mallas, ya se pueden mostrar las mallas creadas. En este estudio en el que se pretende ver el efecto del uso de estas capas en la malla y por consiguiente en el cálculo de las tracciones celulares, se han considerado una serie de casos a analizar. Los datos de los casos escogidos para 4 nodos se presentan en la tabla 1.

Respecto a las mallas de 10, que se han obtenido para los mismos casos que las mallas de 4 nodos, se puede ver una comparativa en la tabla 2, del número de nodos de las mallas de 4 frente a las mallas de 10 nodos. Estos datos arrojan una conclusión que no da lugar a dudas, que no es otra que las mallas de 10 nodos en estos casos considerados son inviables.

Una vez expuestos todos los casos considerados tanto para mallas de tetraedros de 4 nodos como de tetraedros de 10 nodos, ya se está en disposición de enseñar dichas mallas generadas por el código. En las siguientes figuras se va a mostrar un corte de las doce pruebas consideradas de mallas de tetraedros de 4 nodos para que se pueda apreciar la variación en el tamaño de los elementos conforme se aleja de la célula. Por simplicidad solo se mostrarán las mallas de 4 nodos ya que la variación del tamaño de los elementos para las mallas de 10 nodos, será prácticamente idéntica y lo único que varía es el número de nodos, aspecto que no se puede apreciar en estas imágenes.

Nombre del archivo	n	d1	d2	opt0 (tamaño elem sup celula)	opt2 (tamaño elem sup capa1)	opt3 (tamaño elem sup capa2)	opt4	Nº de elementos de la celula	Nº de caras de la celula	Elementos Gel				Nº de elementos totales del Gel
										Nº de elementos capa 1	Nº de elementos capa 2	Nº de elementos exterior	Nº de elementos	
Prueba1	4	5	5	1	2	2	200	227408	80064	267048	98973	206325	572346	
Prueba2	4	5	5	1	2	2	1000	227460	80064	267110	98829	139921	505860	
Prueba3	4	5	5	2	3	3	200	45922	18578	48532	39759	139700	227991	
Prueba4	4	5	5	2	3	3	1000	45828	18578	48535	39757	71630	159922	
Prueba5	4	5	5	3	4	4	200	18594	8268	24227	22334	117810	164371	
Prueba6	4	5	5	3	4	4	1000	18524	8268	24306	22360	48620	95286	
Prueba7	4	5	10	1	2	2	200	226224	80060	266062	141230	234734	642026	
Prueba8	4	5	10	1	2	2	1000	226347	80080	266954	141488	163692	572134	
Prueba9	4	5	10	2	3	3	200	45902	18512	48234	53752	157328	259314	
Prueba10	4	5	10	2	3	3	1000	45951	18512	48242	53805	83344	185391	
Prueba11	4	5	10	3	4	4	200	18472	8198	24045	27807	131856	183708	
Prueba12	4	5	10	3	4	4	1000	18472	8198	24047	27840	57010	108897	

Tabla 1. Casos considerados de mallas con tetraedros de 4 nodos.

Nombre del archivo	Nº de nodos (malla de tetraedros de 4 nodos)	Nº de nodos (malla de tetraedros de 10 nodos)	Factor de aumento del número de nodos
Prueba1	132135	1070072	8,098
Prueba2	120434	977545	8,117
Prueba3	46451	371436	7,996
Prueba4	34491	276677	8,022
Prueba5	31631	251241	7,943
Prueba6	24345	194783	8,001
Prueba7	143378	1160736	8,096
Prueba8	130846	1061664	8,114
Prueba9	51546	412602	8,005
Prueba10	38695	310939	8,036
Prueba11	34903	277863	7,961
Prueba12	21641	172779	7,984

Tabla 2. Comparativa de los nodos de los casos de 4 y 10 nodos.

La inviabilidad de las mallas de 10 nodos se debe al aumento excesivo del número de nodos, ya que se produce un aumento de un factor de 8 respecto a las mallas formadas por tetraedros de 4 nodos, en cada uno de los casos considerados. Esto da lugar a mallas inviables pues algunas incluso superan el millar de nodos y cualquier cálculo realizado con cualquiera de ellas tardaría mucho o incluso no se podrían realizar por cuestiones de memoria del ordenador usado. Por lo tanto, esto deja patente que de las dos estrategias de refinamiento de mallas posibles como son, aumentar el número de nodos (pasar de elementos tetraédricos de 4 a 10 nodos) o aumentar el número de elementos (estrategia correspondiente a la creación de capas), solo esta última sería eficiente en el contexto de TFM 3D.

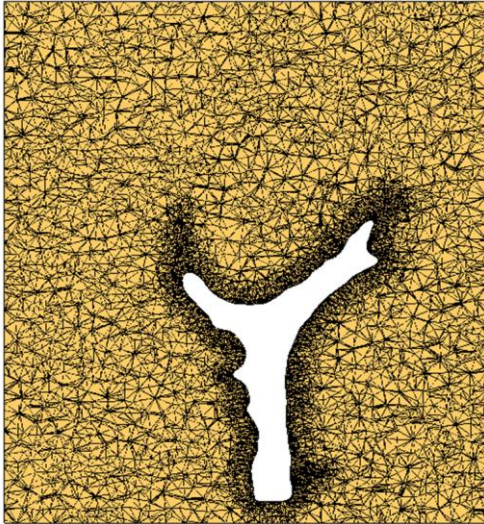


Figura 18. Malla de la prueba 1

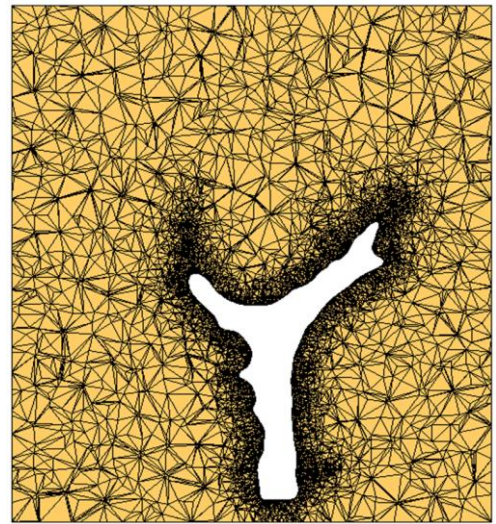


Figura 19. Malla de la prueba 2

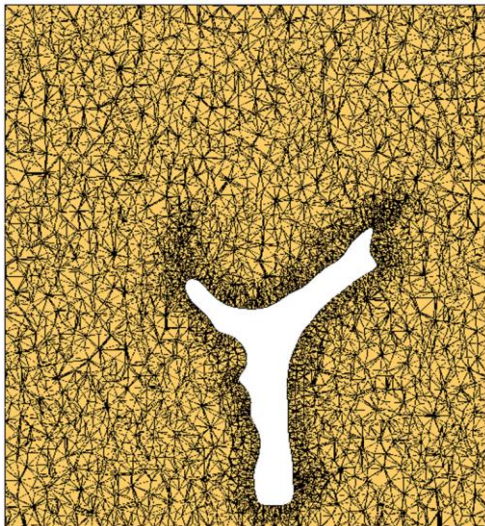


Figura 20. Malla de la prueba 3

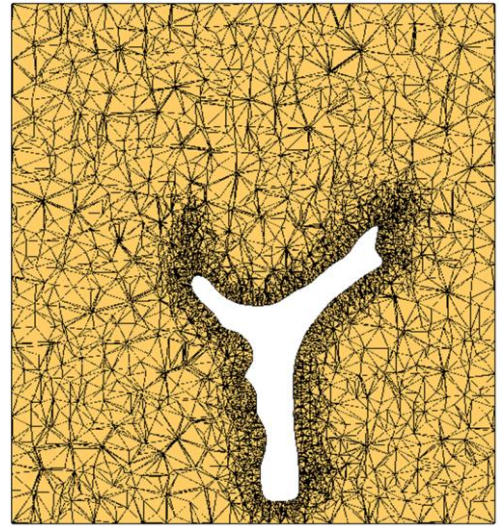


Figura 21. Malla de la prueba 4

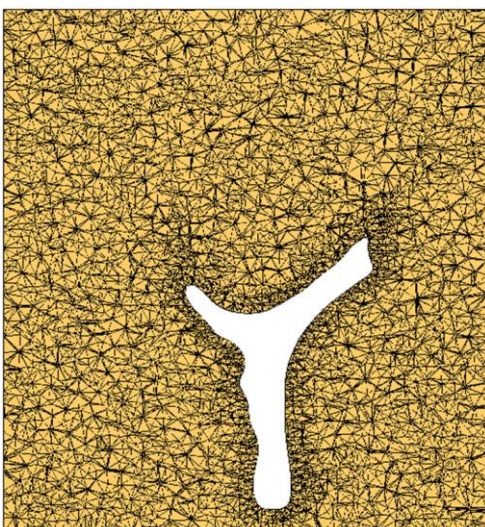


Figura 22. Malla de la prueba 5

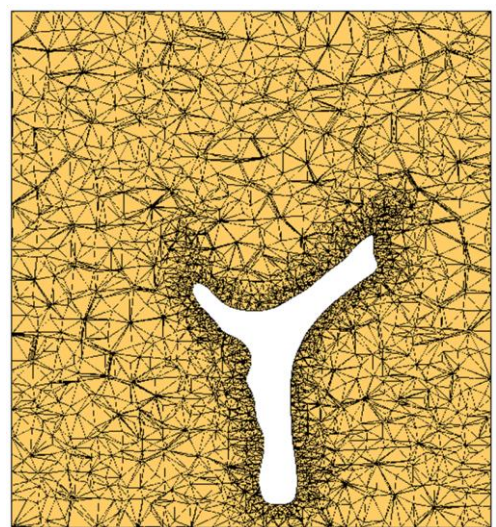


Figura 23. Malla de la prueba 6

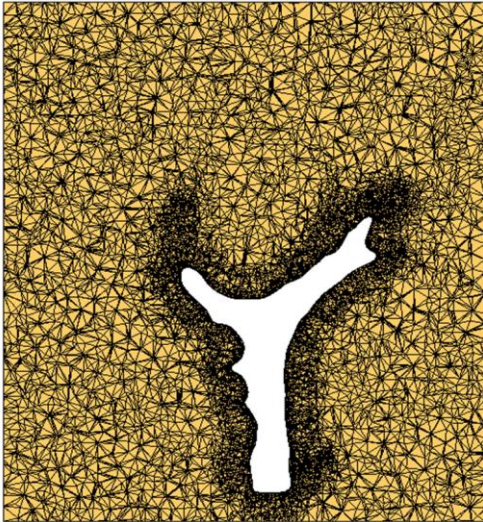


Figura 24. Malla de la prueba 7

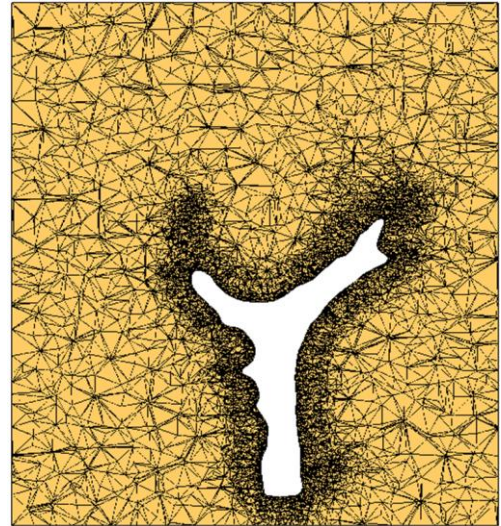


Figura 25. Malla de la prueba 8

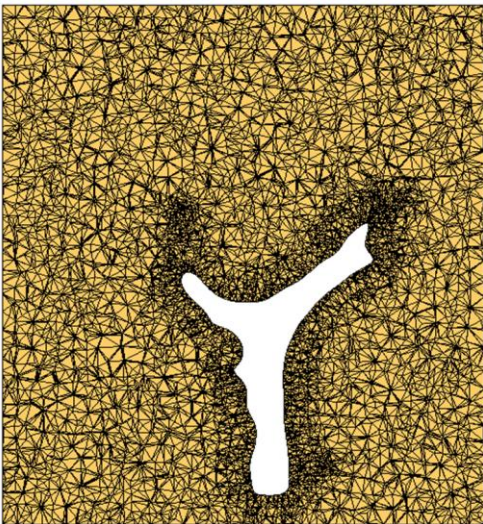


Figura 26. Malla de la prueba 9

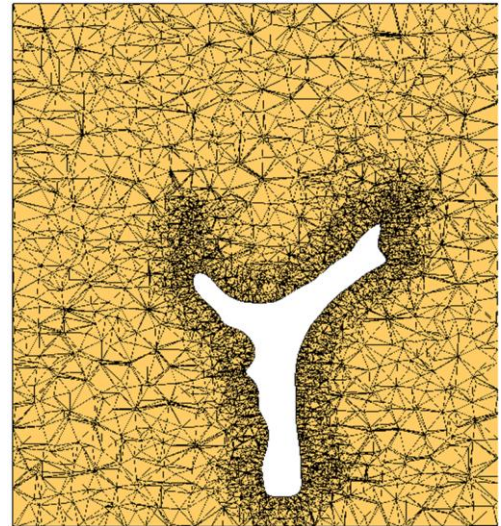


Figura 27. Malla de la prueba 10

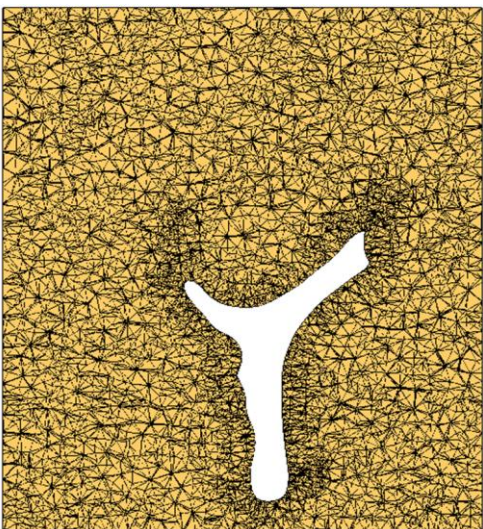


Figura 28. Malla de la prueba 11

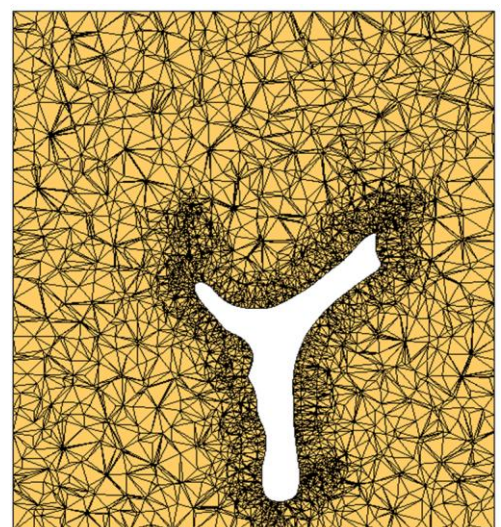


Figura 29. Malla de la prueba 12

Lo primero que salta a la vista es que se ha conseguido el efecto que se perseguía en la malla que no es otro que el uso de elementos más pequeños en las inmediaciones de la célula. También se observa que existe una buena variedad de mallas. Por ejemplo, las pruebas impares usan unos elementos más pequeños lejos de la célula que las pruebas pares, que tienen una malla más gruesa en esas zonas. Otro efecto que se puede ver es que la anchura de la zona con elementos más pequeños es menor en las primeras seis pruebas que en las últimas seis. Por último, también se aprecia como las mallas van haciéndose menos finas en las zonas cercanas a la célula de la prueba 1 a la 6 y de la prueba 7 a la 12. En el capítulo de resultados se podrá ver como se traducen estos efectos a los resultados obtenidos. Para terminar, comentar que el corte realizado en todas las mallas es el mismo, aunque pueda parecer que no. Esto se debe a que la superficie celular se malla con diferente precisión haciendo que su geometría no sea exactamente la misma en todos los caso, sino que varíe ligeramente.

4 RESULTADOS

Tras describir el proceso seguido para desarrollar el código, mallar y resolver el problema, ya se está en condiciones de mostrar los resultados obtenidos con el mallado creado, así como una serie de comentarios al respecto. Primero se mostrarán los resultados obtenidos para un mallado sin capas y después se expondrán los resultados para un mallado con capas donde se han usado elementos tetraédricos de 4 nodos.

4.1 Resultados para un mallado sin capas y tetraedros de 4 nodos

En este apartado se van a exponer los resultados obtenidos para un mallado donde no se ha aplicado el enfoque de las capas. La importancia de estos, es a la hora de comparar, ya que serán tomados como la referencia. Sin embargo, antes de esto, se van a mostrar cómo quedan las mallas generadas sin capas, y en la tabla 3, que es una tabla análoga a la tabla 1, se observarán los datos correspondientes a cada uno de los casos considerados para estos mallados.

<i>Nombre del archivo</i>	<i>n</i>	<i>opt0 (tamaño elem sup célula)</i>	<i>opt4</i>	<i>Nº de elementos de la celula</i>	<i>Nº de caras de la celula</i>	<i>Nº de elementos gel</i>
Prueba1_sincapas	4	1	200	219319	77122	353349
Prueba2_sincapas	4	1	1000	219183	77122	305487
Prueba3_sincapas	4	2	200	46375	18752	139563
Prueba4_sincapas	4	2	1000	46508	18752	90649
Prueba5_sincapas	4	3	200	18566	8248	102211
Prueba6_sincapas	4	3	1000	18567	8248	52431

Tabla 3. Casos considerados de mallas sin capas

Por lo tanto, las mallas de estos seis casos quedan como se ven en las siguientes figuras.

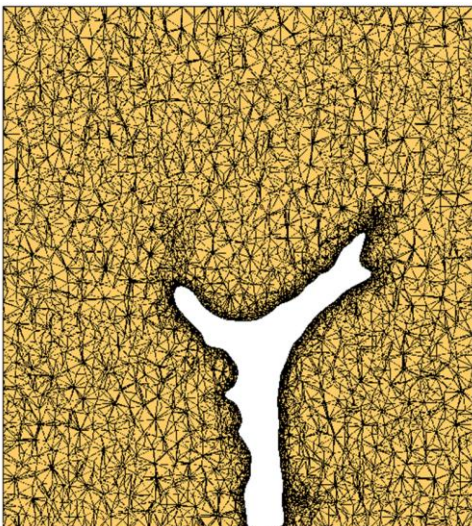


Figura 30. Malla de la prueba 1 sin capas

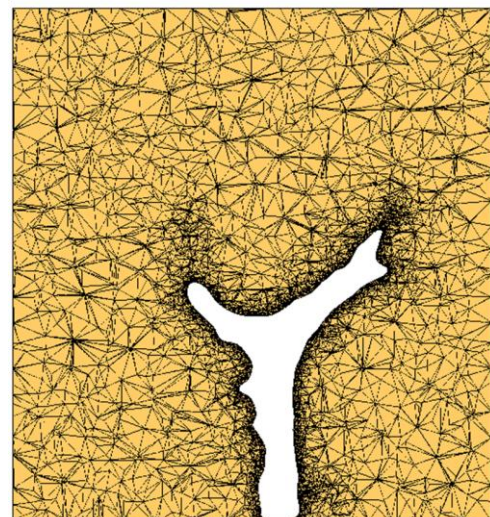


Figura 31. Malla de la prueba 2 sin capas

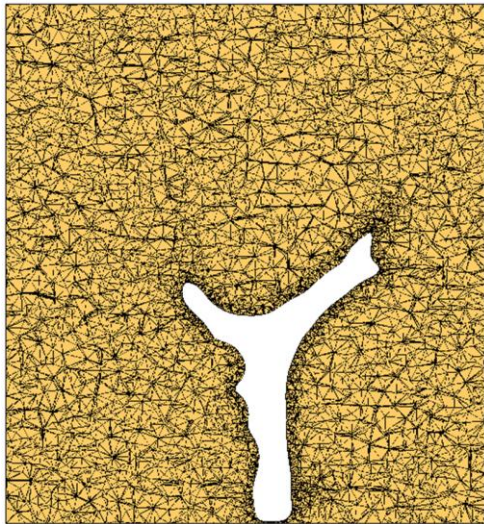


Figura 32. Malla de la prueba 3 sin capas

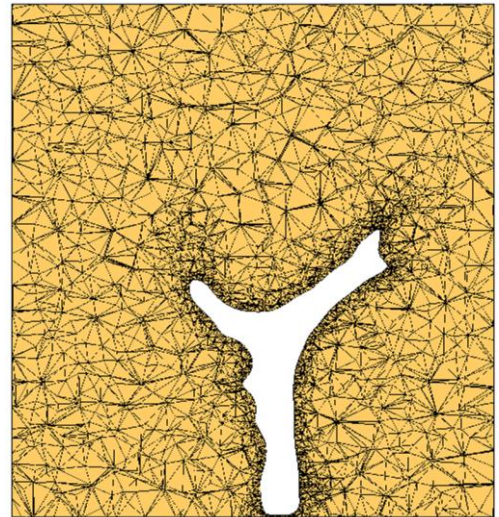


Figura 33. Malla de la prueba 4 sin capas

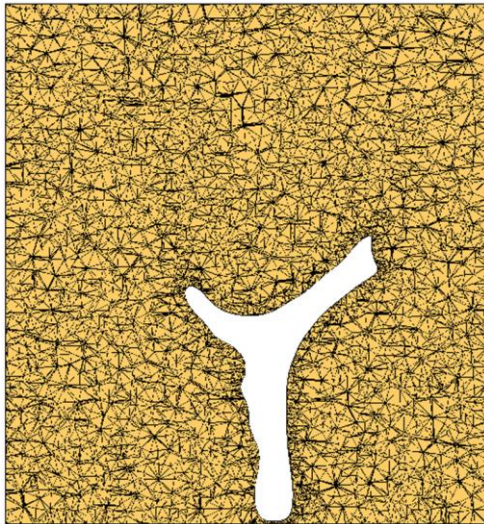


Figura 34. Malla de la prueba 5 sin capas

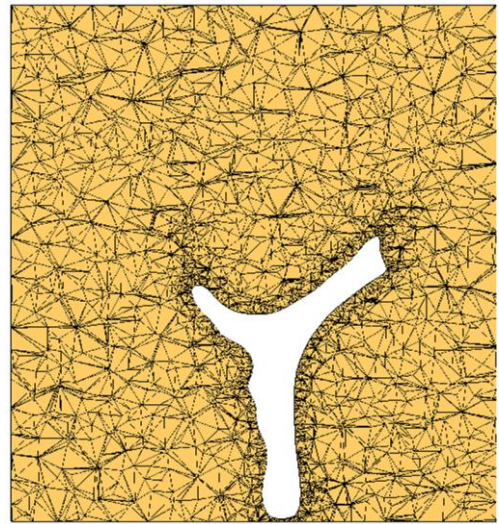


Figura 35. Malla de la prueba 6 sin capas

Por simple inspección visual se puede ver como en estos seis últimos casos, no existe esa aglomeración de elementos en las zonas adyacentes a la célula.

Antes de enseñar los resultados obtenidos, exponer que los resultados que aquí se van a mostrar son el módulo del campo de desplazamiento en la superficie de la célula y en un corte de la misma y el módulo del vector tensión en la superficie celular. También comentar que los valores que aparecerán en los códigos de colores o leyendas junto con las imágenes se encuentran en unas unidades de micras para el caso de los desplazamientos y pascales para el vector tensión. Ahora ya se está en disposición de mostrar los resultados. En primer lugar se mostrarán aquellos correspondientes a los seis casos sin capas.

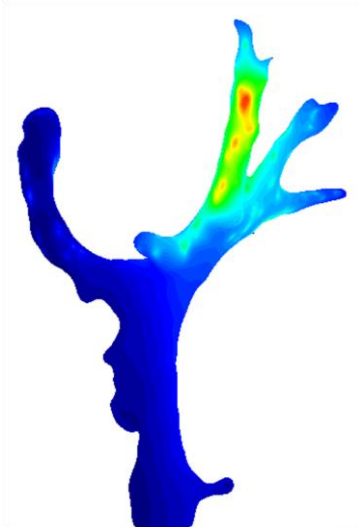


Figura 36. Desplazamiento en la superficie celular en la prueba 1 sin capas.

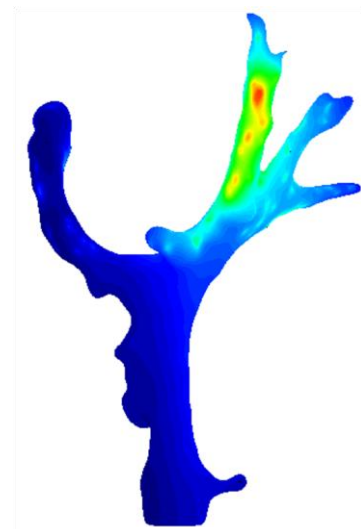


Figura 37. Desplazamiento en la superficie celular en la prueba 2 sin capas.

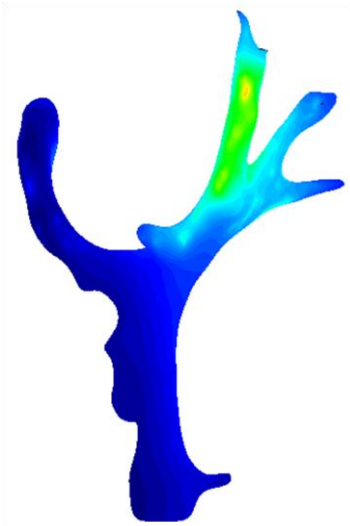


Figura 38. Desplazamiento en la superficie celular en la prueba 3 sin capas.

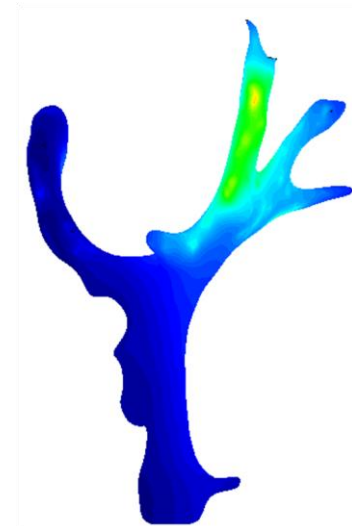
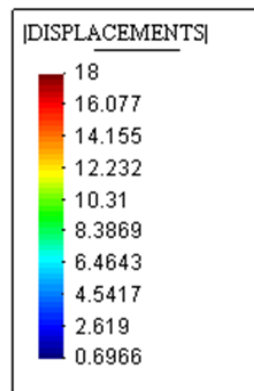


Figura 39. Desplazamiento en la superficie celular en la prueba 4 sin capas.



Figura 40. Desplazamiento en la superficie celular en la prueba 5 sin capas.

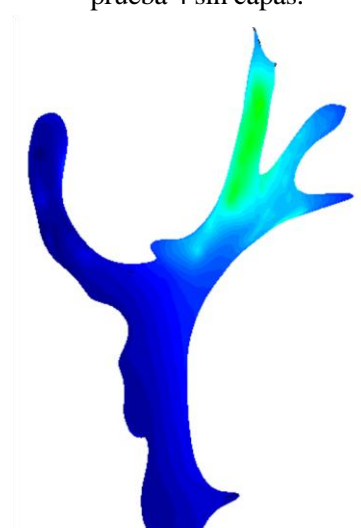


Figura 41. Desplazamiento en la superficie celular en la prueba 6 sin capas.

En estas imágenes se puede apreciar como a medida que la malla se hace mas basta en el contorno de la célula, el desplazamiento va cambiando pues se está captando con mayor error. Eso se puede ver, ya que el máximo de los desplazamientos captados en la prueba 2 es del orden de 60 micras, mientras en la prueba 6 es del orden de 12 micras. También se ve como las imágenes de la derecha son prácticamente iguales a las de la izquierda, por lo que se puede llegar a la conclusión de que para los desplazamientos no es importante hacer un mallado fino lejos de la célula, pudiéndose así disminuir el número de elementos de las mallas y conseguir la misma eficiencia.

Si se muestra ahora los desplazamientos en un corte.

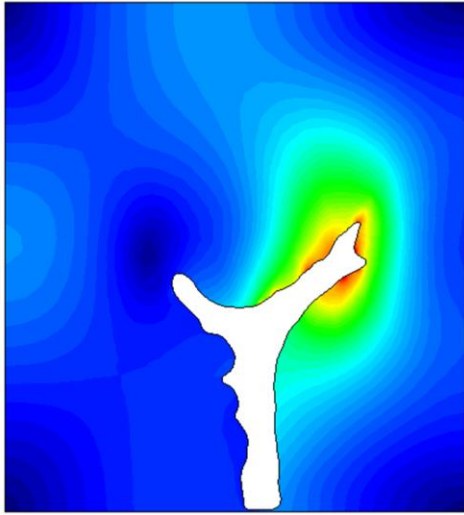


Figura 42. Desplazamiento en un corte en la prueba 1 sin capas.

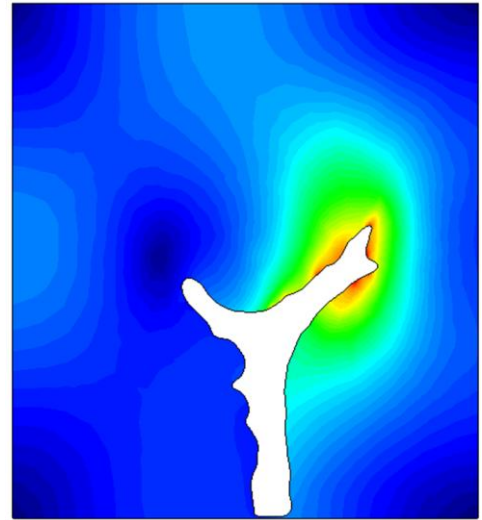


Figura 43. Desplazamiento en un corte en la prueba 2 sin capas.

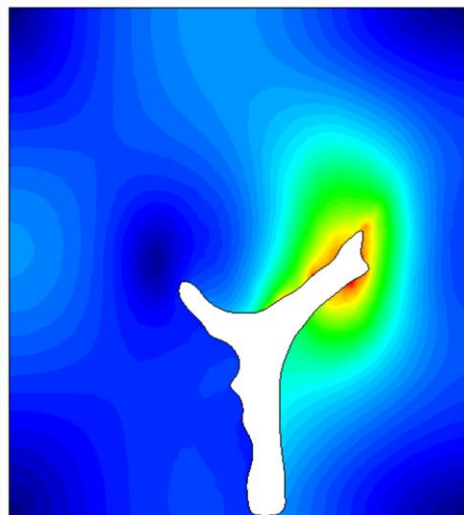


Figura 44. Desplazamiento en un corte en la prueba 3 sin capas.

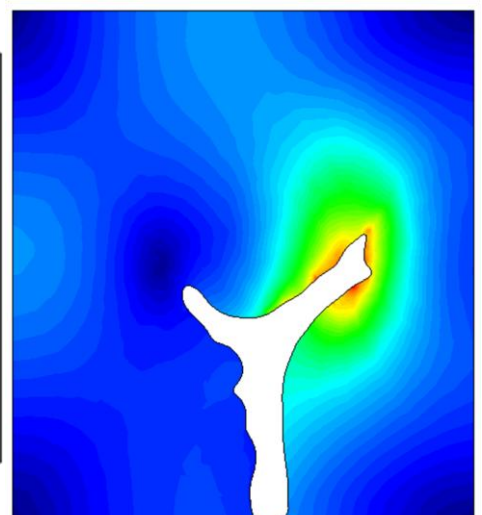
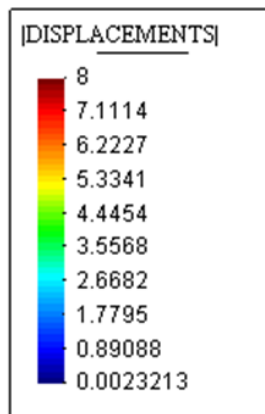


Figura 45. Desplazamiento en un corte en la prueba 4 sin capas.

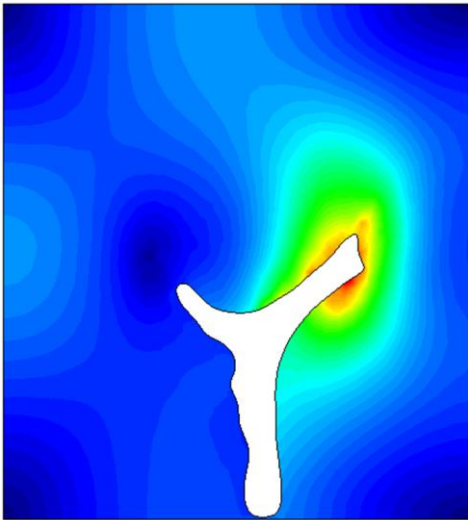


Figura 46. Desplazamiento en un corte en la prueba 5 sin capas.

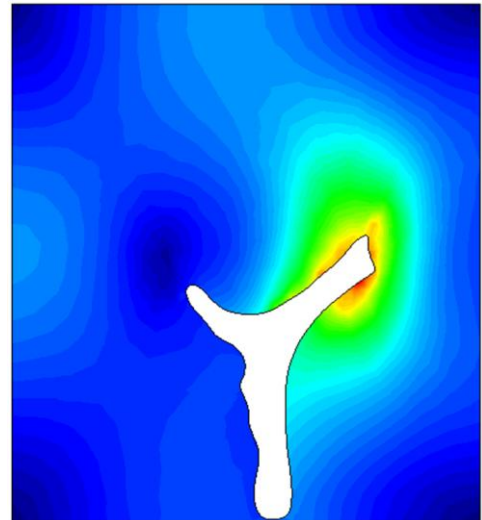


Figura 47. Desplazamiento en un corte en la prueba 6 sin capas.

En estos últimos resultados no hay una gran diferencia entre los distintos casos. La única conclusión que se puede sacar de nuevo es la poca o nula repercusión de una malla fina lejos del contorno celular, ya que no hay diferencia entre las imágenes de la izquierda y las de la derecha.

Por último se van a mostrar los resultados obtenidos para el vector tensión.

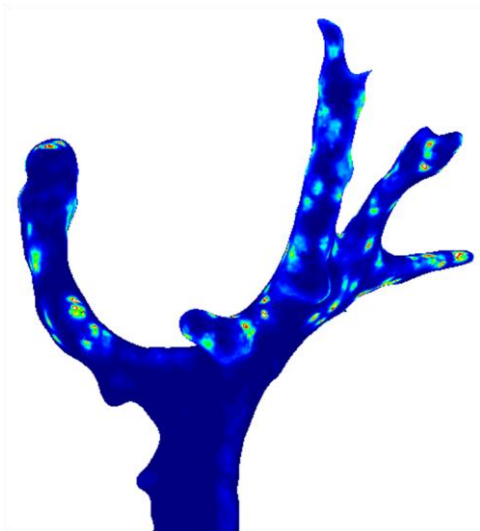


Figura 48. Vector tensión en la superficie celular en la prueba 1 sin capas



Figura 49. Vector tensión en la superficie celular en la prueba 2 sin capas.

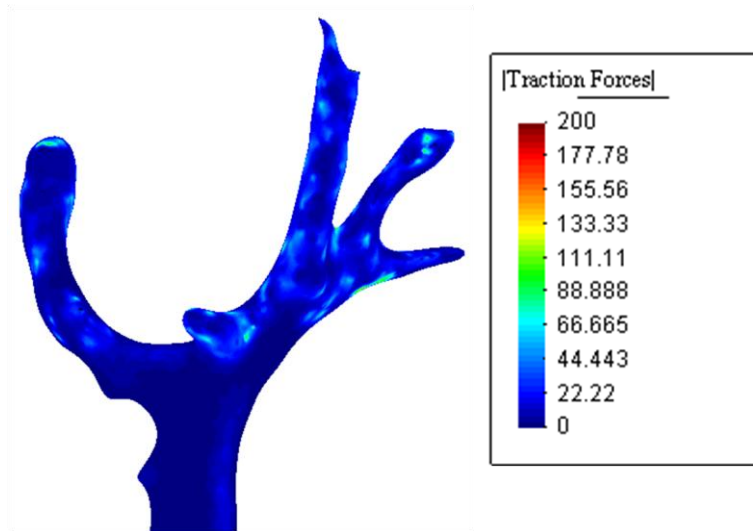


Figura 50. Vector tensión en la superficie celular en la prueba 3 sin capas.



Figura 51. Vector tensión en la superficie celular en la prueba 4 sin capas.



Figura 52. Vector tensión en la superficie celular en la prueba 5 sin capas.



Figura 53. Vector tensión en la superficie celular en la prueba 6 sin capas.

De nuevo se observa lo que se ha ido viendo en los casos anteriores. A medida que la malla se va haciendo más basta, esta capta peor los resultados. También se ve la poca importancia de una malla fina lejos de la célula. Para finalizar, comentar que es normal la gran diferencia existente aquí entre las pruebas 2 y 6, ya que al ser las tensiones proporcionales a las derivadas de los desplazamientos, es más difícil de captar su variación.

4.2 Resultados para un mallado con capas y tetraedros de 4 nodos

En este apartado se van a exponer los resultados obtenidos para un mallado donde se ha utilizado el enfoque de las capas. Dichas mallas estarán formadas por tetraedros compuestos por 4 nodos. Para comenzar se muestra el campo de desplazamiento en el contorno de la célula.



Figura 54. Desplazamiento en la superficie celular en la prueba 1 con capas.



Figura 55. Desplazamiento en la superficie celular en la prueba 2 con capas.



Figura 56. Desplazamiento en la superficie celular en la prueba 3 con capas

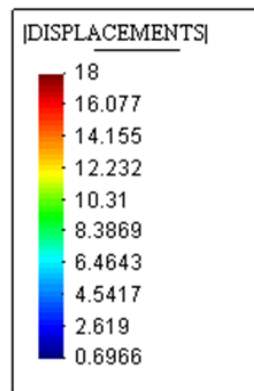


Figura 57. Desplazamiento en la superficie celular en la prueba 4 con capas.



Figura 58. Desplazamiento en la superficie celular en la prueba 5 con capas.



Figura 59. Desplazamiento en la superficie celular en la prueba 6 con capas.



Figura 60. Desplazamiento en la superficie celular en la prueba 7 con capas



Figura 61. Desplazamiento en la superficie celular en la prueba 8 con capas.



Figura 62. Desplazamiento en la superficie celular en la prueba 9 con capas.

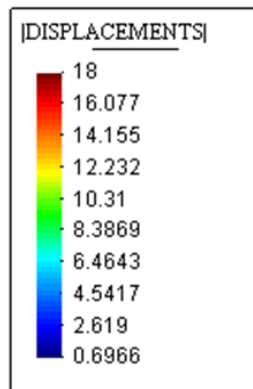


Figura 63. Desplazamiento en la superficie celular en la prueba 10 con capas.



Figura 64. Desplazamiento en la superficie celular en la prueba 11 con capas.



Figura 65. Desplazamiento en la superficie celular en la prueba 12 con capas.

Como es lógico, al utilizar el enfoque de las capas se obtiene unos resultados más precisos que sin utilizarlo. Esto da lugar a una disminución del módulo del desplazamiento respecto al caso sin capas. Por ejemplo, se puede ver los diferentes tonos de colores entre la figura 37, desplazamiento en la superficie celular en la prueba 2 sin capas y la figura 55, desplazamiento en la superficie celular en la prueba 2 con capas. En la parte superior derecha se ve como se pasa de unos valores en el entorno de 10 a unos alrededor de 6 micras. De nuevo, también se puede apreciar que los resultados empeoran al hacer las mallas menos finas y que no hace falta conseguir unas mallas muy finas en zonas lejanas a las células, ya que no afecta demasiado a los resultados, cosa que se puede ver con la escasa diferencia entre las imágenes de la izquierda y las imágenes de la derecha.

Por último, también se puede comentar algo acerca del aumento del espesor de la capa exterior de las últimas seis pruebas respecto a las primeras seis. Se puede apreciar que no hay ningún cambio significativo por ejemplo entre la prueba 2 con capas, figura 55, y la prueba 8 con capas, figura 61, por lo que se puede concluir que la aglomeración de elementos no debe extenderse muy lejos de la célula, ya que deja de tener influencia en el resultado y lo único que hace es aumentar el coste computacional.

Se muestran ahora los desplazamientos en un corte.

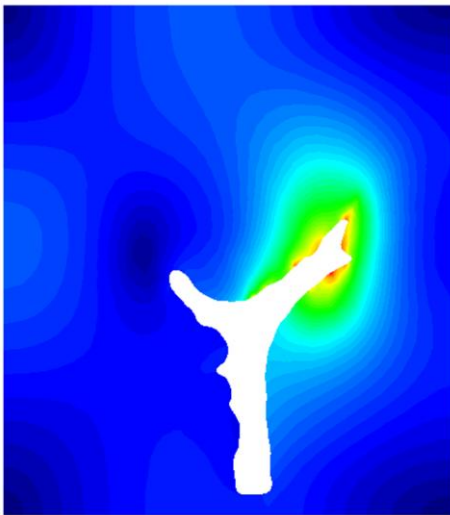


Figura 66. Desplazamiento en un corte en la prueba 1 con capas.

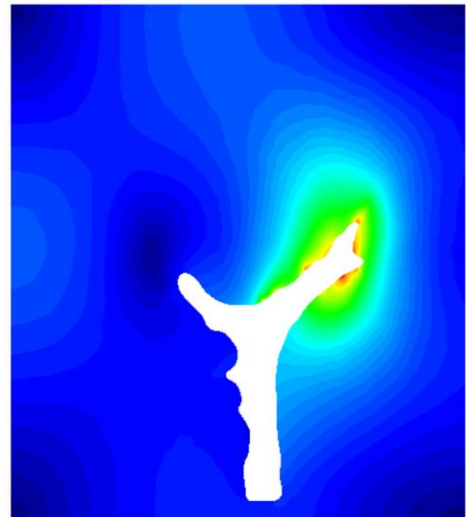


Figura 67. Desplazamiento en un corte en la prueba 2 con capas.

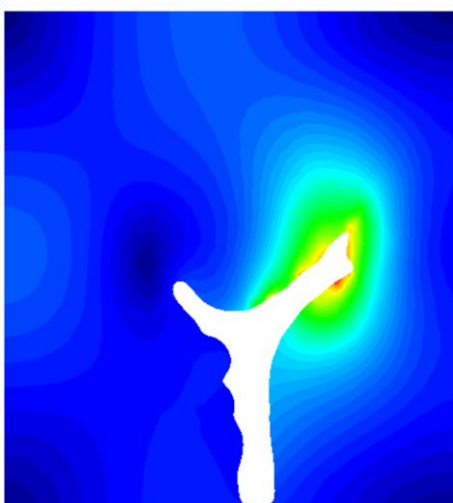


Figura 68. Desplazamiento en un corte en la prueba 3 con capas.

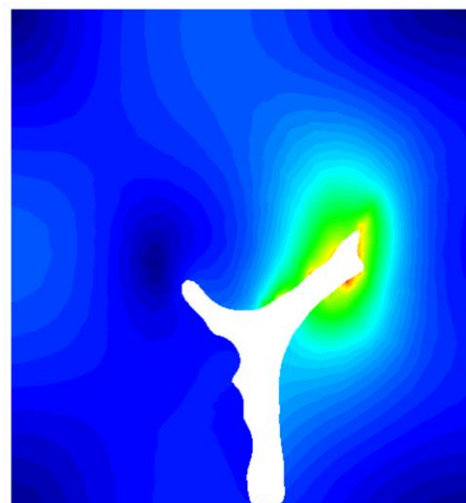
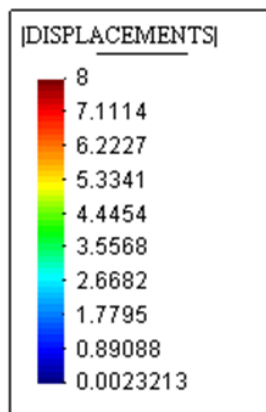


Figura 69. Desplazamiento en un corte en la prueba 4 con capas

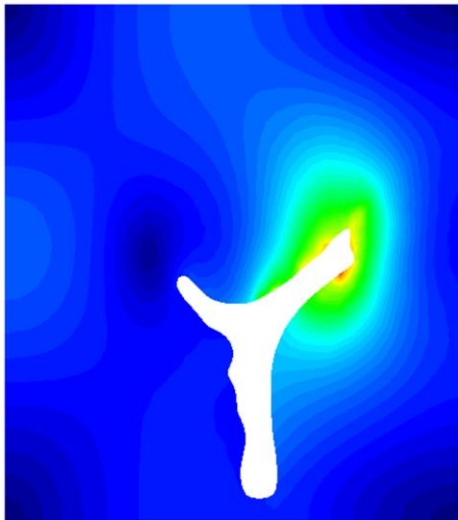


Figura 70. Desplazamiento en un corte en la prueba 5 con capas.

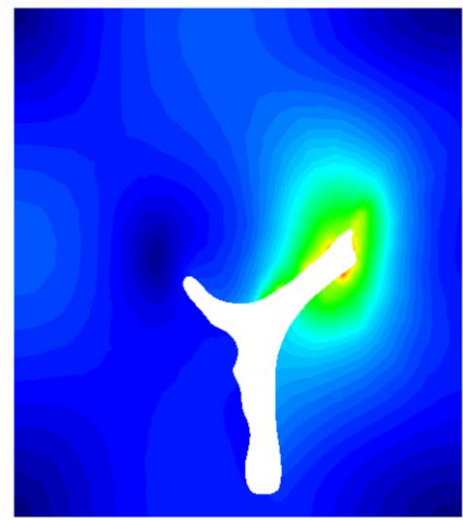


Figura 71. Desplazamiento en un corte en la prueba 6 con capas.

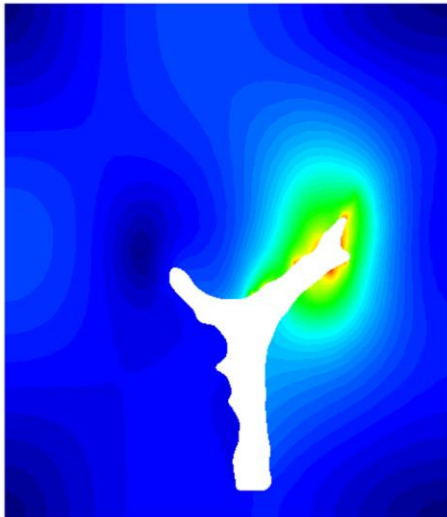


Figura 72. Desplazamiento en un corte en la prueba 7 con capas.

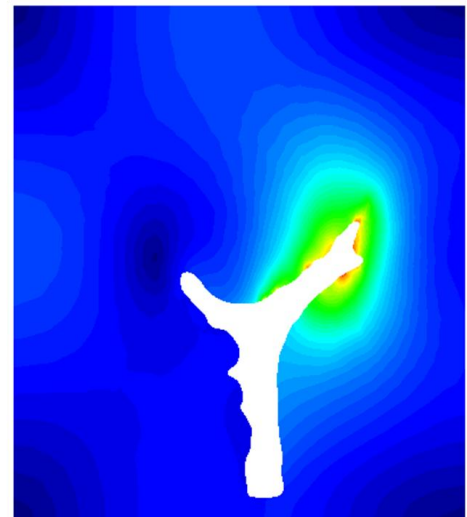


Figura 73. Desplazamiento en un corte en la prueba 8 con capas.

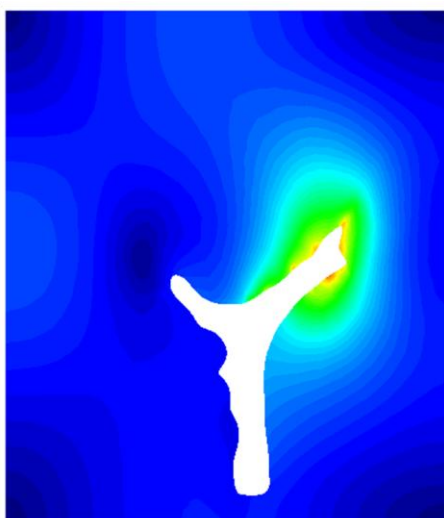


Figura 74. Desplazamiento en un corte en la prueba 9 con capas.

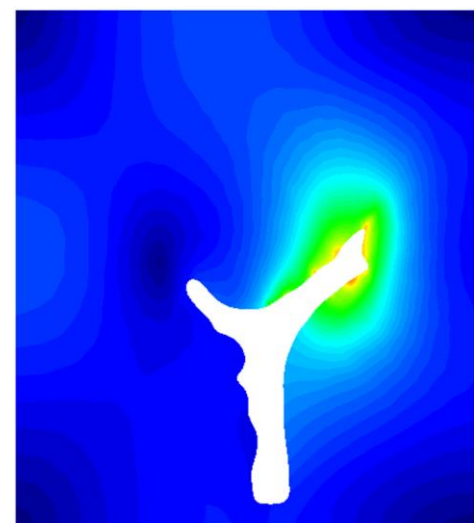
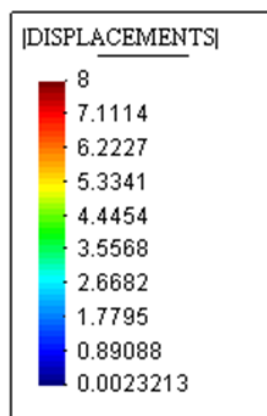


Figura 75. Desplazamiento en un corte en la prueba 10 con capas.

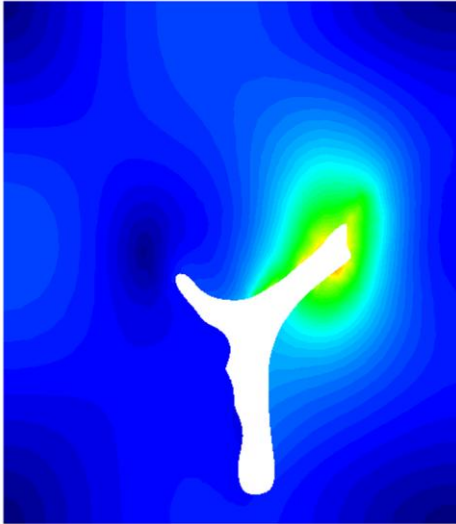


Figura 76. Desplazamiento en un corte en la prueba 11 con capas.

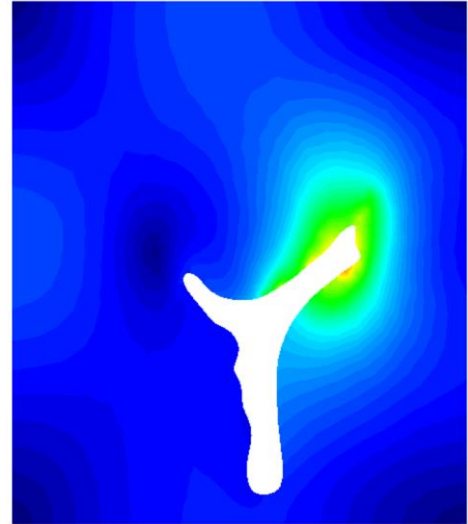


Figura 77. Desplazamiento en un corte en la prueba 12 con capas.

En estos cortes, se puede ver otra vez todo lo que se ha hablado en lo referente al desplazamiento en la superficie de la célula. El uso de las capas proporciona una mejora de la precisión con respecto a los casos donde no se utilizan. Con las capas, los desplazamientos provocados por las células disminuyen y afectan a una zona menor a la que afectaba cuando no se utiliza las capas.

De nuevo, también se ve el empeoramiento de los resultados al disminuir la precisión de la malla, la escasa influencia de una malla más fina lejos del entorno celular y la poca trascendencia de extender la aglomeración de elementos a zonas alejadas a la células, ya que se ve como los cortes de la prueba 1 con capas y de la prueba 7 con capas son prácticamente idénticos.

Ahora, para finalizar se muestra el módulo del vector tensión generado por la célula.

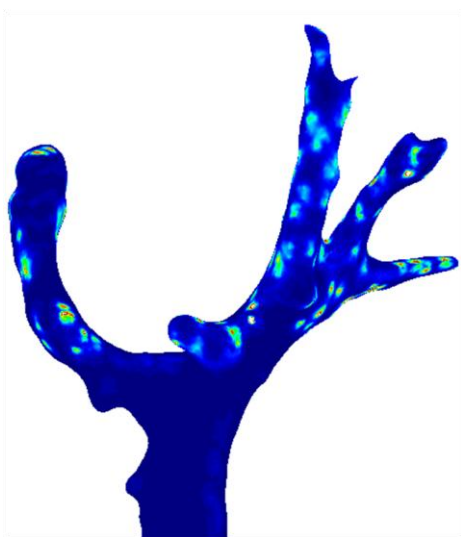


Figura 78. Vector tensión en la superficie celular en la prueba 1 con capas.



Figura 79. Vector tensión en la superficie celular en la prueba 2 con capas.

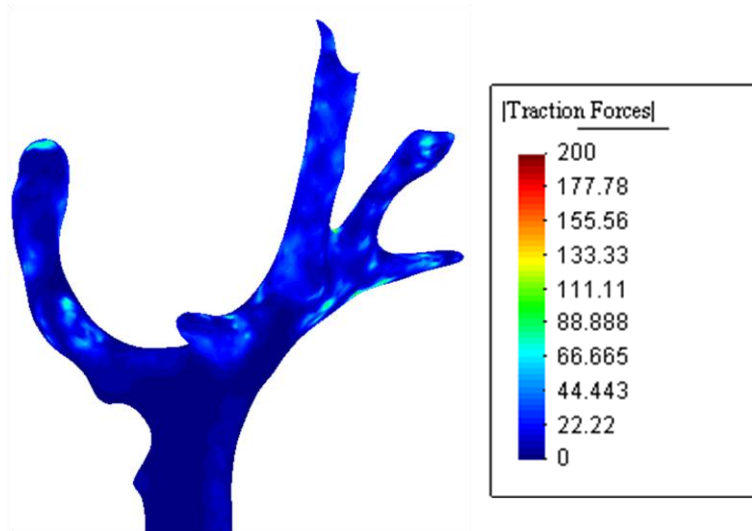


Figura 80. Vector tensión en la superficie celular en la prueba 3 con capas.



Figura 81. vector tensión en la superficie celular en la prueba 4 con capas.



Figura 82. Vector tensión en la superficie celular en la prueba 5 con capas.



Figura 83. Vector tensión en la superficie celular en la prueba 6 con capas.

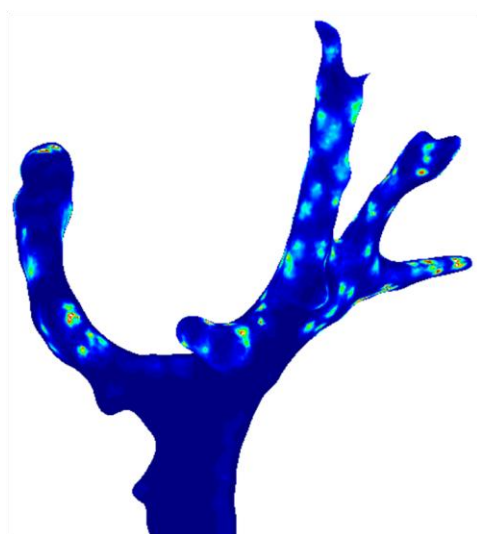


Figura 84. Vector tensión en la superficie celular en la prueba 7 con capas.

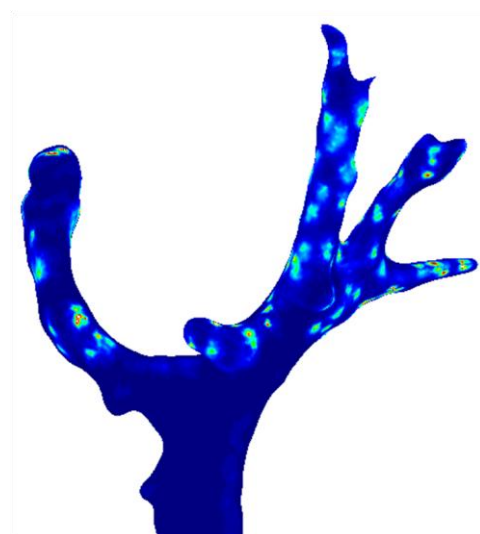


Figura 85. Vector tensión en la superficie celular en la prueba 8 con capas.

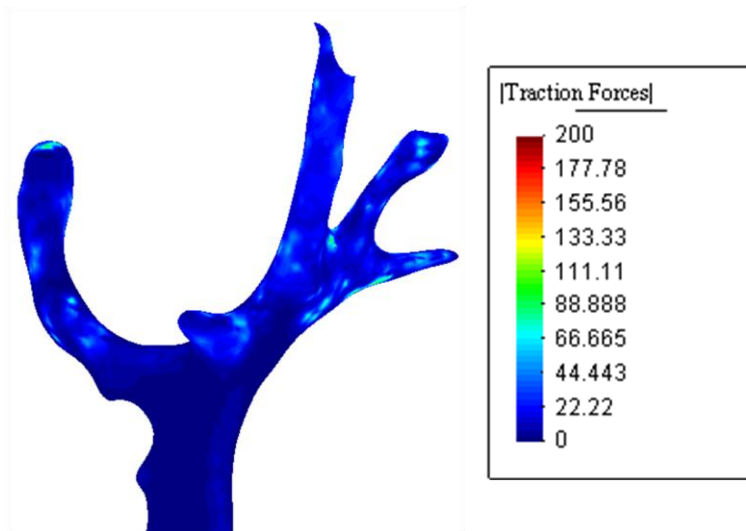


Figura 86. Vector tensión en la superficie celular en la prueba 9 con capas.



Figura 87. Vector tensión en la superficie celular en la prueba 10 con capas.

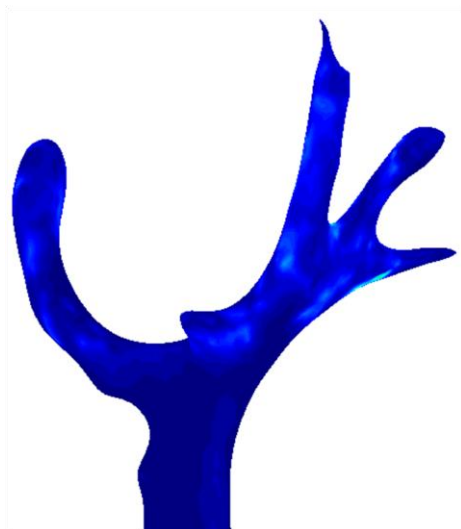


Figura 88. Vector tensión en la superficie celular en la prueba 11 con capas.



Figura 89. Vector tensión en la superficie celular en la prueba 12 con capas.

Tras ver los resultados referentes al vector tensión, se llega a las mismas conclusiones que se han ido sacando del resto de resultados. El uso de las capas, como cabe esperar, provoca una mejora de la precisión aunque quizás, esta es más difícil de apreciar en las imágenes asociadas al vector tensión adjuntadas en este informe. También se observa la pérdida de precisión conforme se hace más basta la malla y la nula utilidad de un mallado fino lejos de la célula, ya que da lugar a los mismos resultados. Por último, como ya se ha mencionado con anterioridad, tiene que existir un espesor óptimo de la zona cercana a la célula donde realizar la aglomeración de elementos, a partir del cual, los resultados sacados ya no varían. Esta conclusión se debe a la nula variación en los resultados en los casos donde se han usado un espesor de capa externa de 5 vóxeles y los casos en los cuales el espesor de la capa externa es de 10 vóxeles. Un ejemplo de esto, pueden ser las pruebas 2 y 8 con capas, correspondientes a las figuras 79 y 85.

5 CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se van a reflejar todas las ideas sacadas a raíz de la realización de este trabajo, tanto las conclusiones que de él se extraen, como los trabajos futuros a los que este da pie.

5.1 Conclusiones

Tras la realización de este trabajo y la investigación realizada en él, se puede decir que 3D TFM es una metodología que aunque es relativamente nueva, está avanzando con cierta rapidez obteniendo resultados satisfactorios en la estimación del vector tensión que aplica la célula en su entorno, y gracias a esto, puede llegar a tener un gran impacto en multitud de disciplinas biológicas como ya se ha comentado.

Entrando un poco más en detalle en lo realizado en este proyecto, en la figura 90 se puede ver una comparativa entre los resultados obtenidos para un mismo caso usando el enfoque de las capas propuesto y sin usarlo. En concreto, se van a comparar la prueba 2 sin capas, la prueba 2 con capas y la prueba 8 con capas, donde estos dos últimos casos son los que proporcionan los resultados más precisos y cuya diferencia es el espesor de las capas. En esta figura se muestra el mallado, el desplazamiento en el contorno de la célula y el vector tensión. Las conclusiones que se derivan son:

- Como cabría esperar, los resultados conseguidos sin capas son diferentes de los resultados conseguidos usando las capas. Esto quiere decir que la idea de mejorar el mallado generando una aglomeración de elementos en las proximidades de la célula da lugar a unos resultados más precisos que los obtenidos con un mallado simple.
- Como se puede ver en la figura 90, el mayor cambio en los resultados se produce en el campo de desplazamiento mientras los cambios que aparecen en el vector tensión aunque los hay, son más sutiles.
- Otra de las conclusiones que se han extraído, es la escasa importancia del mallado en las zonas más alejadas de la célula, lo cual se puede ver gracias a la nula variación de los resultados con dos mallados distintos en dichas zonas. Esto refuerza la idea de generar capas en las proximidades de la célula para aglomerar en esa zona el mayor número de elementos, ya que es ahí donde se produce las variaciones importantes de las magnitudes. Por lo tanto, se puede relajar la malla en las zonas alejadas, disminuyendo así el coste computacional.
- También en la figura 90 se observa como los resultados obtenidos para la prueba 2 con capas y para la prueba 8 con capas son prácticamente idénticos a pesar de que sus mallados difieren en el espesor de las capas usadas. Esto da lugar a que el mallado de la prueba 2 con capas sea el mejor de todos los utilizados, ya que con un coste computacional inferior que el de la prueba 8 con capas, proporciona los mismos resultados.
- En este trabajo se ha dejado patente que en el contexto de 3D TFM, el uso de tetraedros de 10 nodos en vez de usar elementos tetraédricos de 4 nodos, (estrategia de aumentar el número de nodos) es inviable ya que da lugar a mallas inservibles debido al excesivo incremento del número de nodos, que provocaría un coste computacional inasumible. Por lo tanto, la estrategia de aumentar el número de elementos seguida en este informe en la correcta.

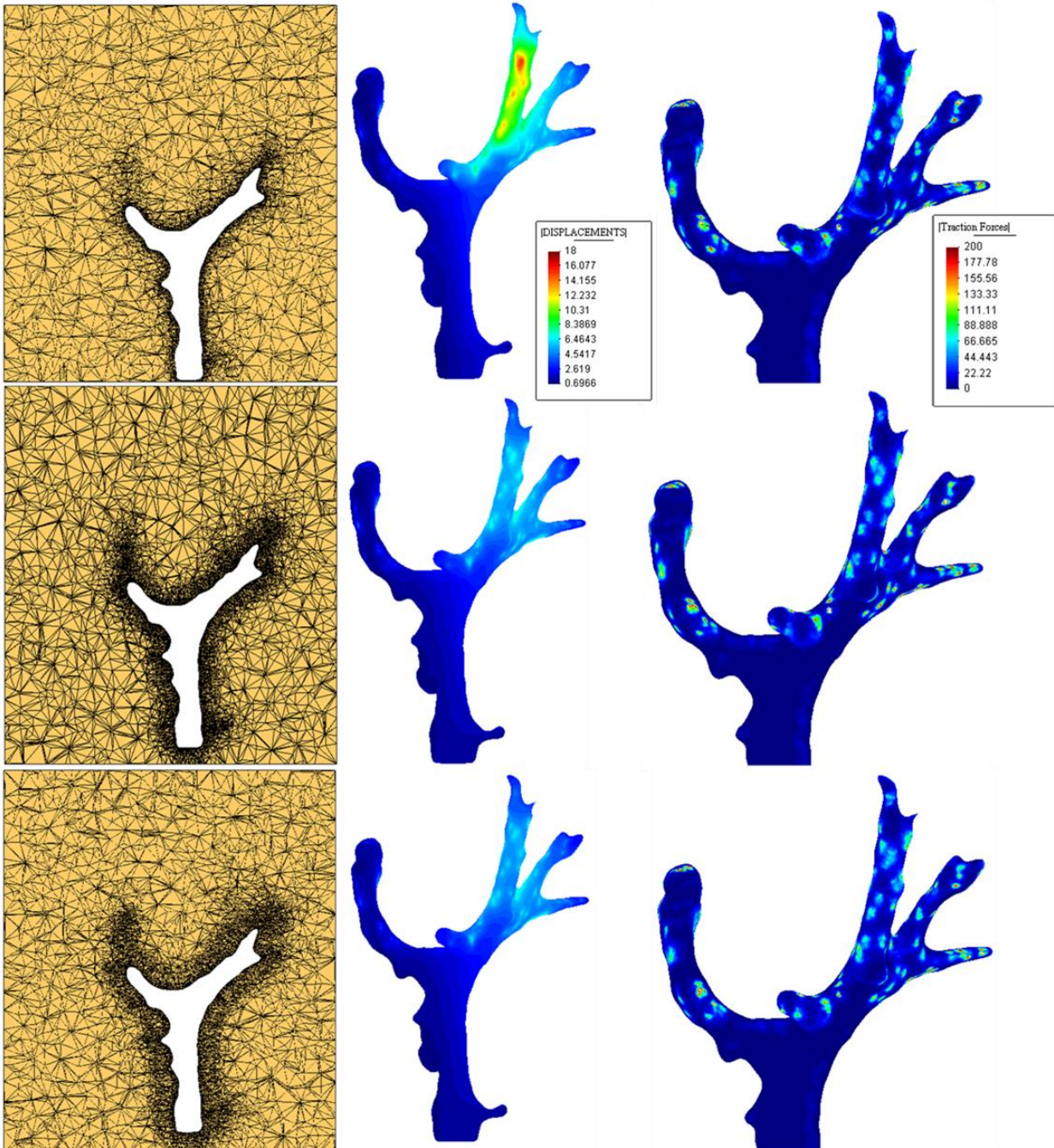


Figura 90. Comparativa (Arriba prueba 2 sin capas, en medio prueba 2 con capas y abajo prueba 8 con capas).

5.2 Trabajos futuros

Este trabajo ha dado lugar a algunas ideas con las que se podrían mejorar este mismo informe, pero por falta de tiempo o de medios, ha sido imposible llevarlas a cabo por lo que se van a exponer aquí por si en un futuro son factibles.

- Para empezar, aunque se ha llegado a la conclusión clara de que el uso de elementos tetraédricos de 10 nodos no es una posibilidad real en esta metodología, habría que validarla y ver si realmente el uso de este tipo de mallas no proporciona ninguna mejora inesperada en los resultados. Como ya se ha comentado, el excesivo coste computacional de dichas mallas, las hace casi inservibles, pero con el ordenador usado para realizar este trabajo, hubiera sido imposible comprobar dicha conclusión.
- La escasa variación de los resultados al usar un espesor de capa u otro, deja entreabierto la posibilidad de la existencia de un espesor mínimo óptimo de dichas capas con el que captar de forma precisa las variaciones de las magnitudes, aumentando mínimamente el coste computacional respecto de las mallas sin capas. Por lo tanto, esta podría ser un línea de investigación interesante.
- Aunque finalmente con el toolbox "Iso2mesh" se ha conseguido la generación de las capas buscada para conseguir la aglomeración de elementos en las proximidades de la célula, esta herramienta a lo largo del trabajo ha dado varios problemas, e incluso no se han podido obtener todos los casos de mallados que en un principio se querían. Esto abre la puerta a intentar mejorar los códigos que forman este toolbox para solucionar los problemas que han ido presentándose durante la realización de este trabajo.

Anexos

Anexo A: Script Mallado_celula_capas

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% CÓDIGO DESARROLLADO POR MANUEL ROJAS NARANJO %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% DESCRIPCIÓN:
%SCRIPT DONDE SE CARGA LA GEOMETRIA DE UNA CELULA Y SE CREA LA MALLA.
%ESTA CÉLULA ESTARA RODEADA DE DOS CAPAS Y TODO ESO ENCERRADO EN UN CUBO.
%LAS CARACTERÍSTICAS DE LA MALLA COMO TAMAÑO Y NUMERO DE NODOS DE LOS
%ELEMENTOS Y ESPESOR DE LAS DOS CAPAS SERÁN LOS PARÁMETROS DE ENTRADA A
%ESTE SCRIPT.

clear all, clc, close all;
%% DATOS DE ENTRADA
n = 10; %Número de nodos que tiene los elementos de la malla
d1 = 5; %espesor de la capa interna (1)
d2 = 10;%espesor de la capa externa (2)

opt0.radbound = 3; % tamaño del elemento de la superficie de la célula.
%Radio máximo de la esfera de delaunay.
opt2.radbound = 4; % tamaño del elemento de la superficie de la capa interna
opt3.radbound = 4; % tamaño del elemento de la superficie de la capa externa

%Le entra al surf2mesh
opt4 = 1000; % Volumen máximo de los elementos que forman el cubo.

filename='Pruebaprueba1';

%% GEOMETRÍA.
%Cargo la geometría de la célula.
load segmented_sprout.mat
load('segmented_sprout_branch_sections.mat');

Lx = res(1);
Ly = res(2);
Lz = res(3);

mask_all = segmented_sprout.mask_all;

for k=1:size(mask_all,3)
    for j=1:round(Lz/Lx,0)
        mask_all2(:, :, round(Lz/Lx,0)*(k-1)+j) = mask_all(:, :, k);
    end
end

mask3D_0 = mask_all2;

% La matriz binaria mask3D_0 tiene dimensiones 286x317x81. Uso padarray
% para generar un cubo.
x0 = 57;
y0 = 41;
% z0 = 186;
z0 = 159; %Algunos casos hay que usar este z0 para que funcione
% en vez del anterior z0.

```



```

mask3D_0 = padarray(mask3D_0,[x0 y0 z0]);

mask3D_2 = mask3D_0;
mask3D_3 = mask3D_0;
pos = find(mask3D_2 == 1);
[i,j,k] = ind2sub(size(mask3D_2),pos);

for m = 1:length(i)
    mask3D_2(i(m),j(m),k(m)+[1:d1]) = 1; % Se coloca un 1 en todos los
    %voxeles adyacentes dentro de una distancia d1
    mask3D_2(i(m),j(m),k(m)-[1:d1]) = 1;
    mask3D_2(i(m),j(m)+[1:d1],k(m)) = 1;
    mask3D_2(i(m),j(m)-[1:d1],k(m)) = 1;
    mask3D_2(i(m)+[1:d1],j(m),k(m)) = 1;
    mask3D_2(i(m)-[1:d1],j(m),k(m)) = 1;

    mask3D_3(i(m),j(m),k(m)+[1:(d1+d2)]) = 1; % Se coloca un 1 en todos
    %los voxeles adyacentes dentro de una distancia d1+d2
    mask3D_3(i(m),j(m),k(m)-[1:(d1+d2)]) = 1;
    mask3D_3(i(m),j(m)+[1:(d1+d2)],k(m)) = 1;
    mask3D_3(i(m),j(m)-[1:(d1+d2)],k(m)) = 1;
    mask3D_3(i(m)+[1:(d1+d2)],j(m),k(m)) = 1;
    mask3D_3(i(m)-[1:(d1+d2)],j(m),k(m)) = 1;
end

%% MALLADO
addpath(''); %Colocar el directorio donde se encuentre iso2mesh

%Se crea la malla de cada volumen por separado.
[no0,e10,regions0,holes0]=vol2surf(mask3D_0,1:size(mask3D_0,1),...
    1:size(mask3D_0,2),1:size(mask3D_0,3),opt0,1,'cgalsurf',0.5);

[no2,e12,regions2,holes2]=vol2surf(mask3D_2,1:size(mask3D_2,1),...
    1:size(mask3D_2,2),1:size(mask3D_2,3),opt2,1,'cgalsurf',0.5);

[no3,e13,regions3,holes3]=vol2surf(mask3D_3,1:size(mask3D_3,1),...
    1:size(mask3D_3,2),1:size(mask3D_3,3),opt3,1,'cgalsurf',0.5);

%Se Suaviza cada malla por separado.
addpath('');% Colocar directorio donde se encuentre smoothpatch_version1b

FV0.vertices = no0;    FV0.faces = e10(:,1:3);
FV2.vertices = no2;    FV2.faces = e12(:,1:3);
FV3.vertices = no3;    FV3.faces = e13(:,1:3);

FV00 = smoothpatch(FV0,1,5);
FV22 = smoothpatch(FV2,1,5);
FV33 = smoothpatch(FV3,1,5);

mask3D = mask3D_3 + mask3D_2 + mask3D_0;

IID1 = find(mask3D==1);%Para identificar un punto de cada región.
IID2 = find(mask3D==2);
IID3 = find(mask3D==3);

[ID1, JD1, KD1] = ind2sub(size(mask3D),IID1);%Identifica todos los puntos
%de cada región y lo pone en un formato necesario.
[ID2, JD2, KD2] = ind2sub(size(mask3D),IID2);

```

```
[ID3, JD3, KD3] = ind2sub(size(mask3D),IID3);

%Se juntan todos los vértices y todas las caras
FV.vertices = [FV00.vertices; FV22.vertices; FV33.vertices];
FV.faces = [FV00.faces; FV22.faces+size(FV00.vertices,1);...
            FV33.faces+size(FV00.vertices,1)+size(FV22.vertices,1)];
%Esas sumas es para cambiar el número de los vértices

% REDUCCIÓN DEL TAMAÑO DEL CUBO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Para que el cubo que envuelve todo el dominio no difiera
% demasiado del cubo original.
i_min = min(ID1);      i_max = max(ID1);
j_min = min(JD1);      j_max = max(JD1);
k_min = min(KD1);      k_max = max(KD1);

%Hay que tener en cuenta el cambio de coordenadas (x' = x0 + x)
i_min = min([i_min,x0]);      i_max = max([i_max,x0+286]);
j_min = min([j_min,y0]);      j_max = max([j_max,y0+317]);
k_min = min([k_min,z0]);      k_max = max([k_max,z0+81]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% %Represento superficie
% figure(1)
% plotsurf(FV.vertices,FV.faces);

[node,el1,fc1]=surf2mesh(FV.vertices,FV.faces,[i_min-5 j_min-5 k_min-5],...
    [i_max+5 j_max+5 k_max+5],1,opt4,[ID1(1) JD1(1) KD1(1);...
    ID2(1) JD2(1) KD2(1);ID3(1) JD3(1) KD3(1)],[],1);

% figure(2)
% plotmesh(node(:,[2 1 3]),fc1,'facealpha',0.7);% Pinta malla superficial

%% IDENTIFICACIÓN DE NODOS Y ELEMENTOS

%Identifica que elementos pertenece a cada conjunto
elem_ext=el1((el1(:,5)==0),:); %Matriz de elementos tipo 0 con los nodos
%que forman cada elemento. Ultima columna es 0
elem_capal=el1((el1(:,5)==2),:);
elem_capa2=el1((el1(:,5)==1),:);
elem_celula=el1((el1(:,5)==3),:);

ind_e_ext = find(el1(:,5)==0);%Vector donde se encuentra las filas de la
%matriz el1 donde hay elementos 0
ind_e_c1 = find(el1(:,5)==2);
ind_e_c2 = find(el1(:,5)==1);
ind_e_celula = find(el1(:,5)==3);

node = [Lx*node(:,1) Ly*node(:,2) Lz/round(Lz/Lx,0)*node(:,3)];

n_cel = intersect(elem_capal(:,1:4),elem_celula(:,1:4));
node_cel = node(n_cel,:);

node = node - min(node_cel);

%Saco algunos datos por pantalla
disp(['Número de elementos en la célula = ',num2str(length(elem_celula))])
disp(['Número de caras en la superficie de la célula = ',...
    num2str(length(FV00.faces))])
```

```

disp(['Número de elementos capa 1 = ',num2str(length(elem_capa1))])
disp(['Número de elementos capa 2 = ',num2str(length(elem_capa2))])
disp(['Número de elementos exterior = ',num2str(length(elem_ext))])

%% DISTINCIÓN SEGÚN EL N° DE NODOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CASO 4 NODOS %%%%%%%%%
switch n
  case 4
    %% DATOS DE SALIDA CASO 4 NODOS %%

    coordenadas = [[1:size(node,1)'],'node'];

    elementos_celula = [ind_e_celula,elem_celula(:,1:4)];
    elementos_capa1 = [ind_e_c1,elem_capa1(:,1:4)];
    elementos_capa2 = [ind_e_c2,elem_capa2(:,1:4)];
    elementos_ext = [ind_e_ext,elem_ext(:,1:4)];

    %% ESCRIBIR LOS DATOS (FICHERO GID) CASO 4 NODOS %%

    fileID = fopen(strcat(filename, '.flavia.msh'),'w');
    fprintf(fileID, 'MESH "TETRAEDROS 4 NODOS EXTERIOR (GEL)" DIMENSION 3
ELEMENTYPE Tetrahedra Nnode 4\r\n');
    fprintf(fileID, 'COORDINATES\r\n');
    fprintf(fileID, '%.0f %.10f %.10f %.10f\r\n', coordenadas');
    fprintf(fileID, 'END COORDINATES\r\n');
    fprintf(fileID, 'ELEMENTS\r\n');
    fprintf(fileID, '%.0f %.0f %.0f %.0f %.0f\r\n', elementos_ext');
    fprintf(fileID, 'END ELEMENTS\r\n');
    fprintf(fileID, 'MESH "TETRAEDROS 4 NODOS CAPA 2 (GEL)" DIMENSION 3
ELEMENTYPE Tetrahedra Nnode 4\r\n');
    fprintf(fileID, 'COORDINATES\r\n');
    fprintf(fileID, 'END COORDINATES\r\n');
    fprintf(fileID, 'ELEMENTS\r\n');
    fprintf(fileID, '%.0f %.0f %.0f %.0f %.0f\r\n', elementos_capa2');
    fprintf(fileID, 'END ELEMENTS\r\n');
    fprintf(fileID, 'MESH "TETRAEDROS 4 NODOS CAPA 1 (GEL)" DIMENSION 3
ELEMENTYPE Tetrahedra Nnode 4\r\n');
    fprintf(fileID, 'COORDINATES\r\n');
    fprintf(fileID, 'END COORDINATES\r\n');
    fprintf(fileID, 'ELEMENTS\r\n');
    fprintf(fileID, '%.0f %.0f %.0f %.0f %.0f\r\n', elementos_capa1');
    fprintf(fileID, 'END ELEMENTS\r\n');
    fprintf(fileID, 'MESH "TETRAEDROS 4 NODOS CÉLULA" DIMENSION 3
ELEMENTYPE Tetrahedra Nnode 4\r\n');
    fprintf(fileID, 'COORDINATES\r\n');
    fprintf(fileID, 'END COORDINATES\r\n');
    fprintf(fileID, 'ELEMENTS\r\n');
    fprintf(fileID, '%.0f %.0f %.0f %.0f %.0f\r\n', elementos_celula');
    fprintf(fileID, 'END ELEMENTS\r\n');
    fclose(fileID);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CASO 10 NODOS %%%%%%%%%
    case 10
    %% COLOCAR NUEVOS NODOS %%
    %Colocar nuevos nodos para generar elementos tetraédrico de 10 nodos.
    s = length(node);
    r = length(e11);
    %Coordenadas de los nuevos nodos
    Coord_Nodos5 = (node(e11(:,1),:)+ node(e11(:,2),:))/2; %Matriz de
    % coordenadas de todos los nodos 5 de cada elemento
    Coord_Nodos6 = (node(e11(:,2),:)+ node(e11(:,3),:))/2;
    Coord_Nodos7 = (node(e11(:,1),:)+ node(e11(:,3),:))/2;

```

```

Coord_Nodos8 = (node(e11(:,1),:)+ node(e11(:,4),:))/2;
Coord_Nodos9 = (node(e11(:,2),:)+ node(e11(:,4),:))/2;
Coord_Nodos10 = (node(e11(:,3),:)+ node(e11(:,4),:))/2;
%Esta enumeración es la que sigue abaqus

%MATRIZ1 Contiene las coordenadas de los nuevos nodos. Esta por
%bloques las primeras r filas son las coordenadas de los nodos 5,
%despues los de los nodos 6...
MATRIZ1 = [Coord_Nodos5;Coord_Nodos6;Coord_Nodos7;Coord_Nodos8;...
          Coord_Nodos9;Coord_Nodos10];
MATRIZ1 = [[s+1:s+6*r]',MATRIZ1];

MATRIZ2 = zeros(size(e11,1),11);
MATRIZ2(:,1) = [1:r]';
MATRIZ2(:,2:5) = e11(:,1:4);
MATRIZ2(:,6) = MATRIZ1(1:r,1);
MATRIZ2(:,7) = MATRIZ1(r+1:2*r,1);
MATRIZ2(:,8) = MATRIZ1(2*r+1:3*r,1);
MATRIZ2(:,9) = MATRIZ1(3*r+1:4*r,1);
MATRIZ2(:,10) = MATRIZ1(4*r+1:5*r,1);
MATRIZ2(:,11) = MATRIZ1(5*r+1:6*r,1);
%MATRIZ2 Contiene todos los elementos, los cuales se han
%pasado de 4 a 10 nodos.

%%% ELIMINAR LOS NODOS QUE ESTAN REPETIDOS EN LOS NUEVOS CREADOS%%%

[C,ia,ic] = unique(MATRIZ1(:,2:4),'rows','stable');
% Con unique lo que hago es eliminar los nodos repetidos.
ic = ic+s;
MATRIZ1 = [ic(ia),C];
MATRIZ2 = [MATRIZ2(:,1),MATRIZ2(:,2:5),ic(MATRIZ2(:,6:11))-s)];

%%% DATOS DE SALIDA CASO 10 NODOS %%%

coordenadas_10 = [[1:size(node,1)]',node; MATRIZ1];
%Contiene todos los nodos.
coordenadas = [[1:size(node,1)]',node];
%coordenadas de los nodos originales

%Datos por pantalla
disp(['Número de nodos para el caso de 4 nodos = ',...
      num2str(size(coordenadas,1))])
disp(['Número de nodos para el caso de 10 nodos = ',...
      num2str(size(coordenadas_10,1))])

elementos_celula_10 = MATRIZ2(ind_e_celula,:);
elementos_celula = MATRIZ2(ind_e_celula,1:5);

elementos_capa1_10 = MATRIZ2(ind_e_c1,:);
elementos_capa1 = MATRIZ2(ind_e_c1,1:5);

elementos_capa2_10 = MATRIZ2(ind_e_c2,:);
elementos_capa2 = MATRIZ2(ind_e_c2,1:5);

elementos_ext_10 = MATRIZ2(ind_e_ext,:);
elementos_ext = MATRIZ2(ind_e_ext,1:5);

```

```
%%% ESCRIBIR LOS DATOS (FICHERO GID) CASO 10 NODOS %%%
```

```

fileID = fopen(strcat(filename, '.flavia.msh'), 'w');
fprintf(fileID, 'MESH "TETRAEDROS 10 NODOS EXTERIOR (GEL)" DIMENSION
3 ELEMTYPE Tetrahedra Nnode 10\r\n');
fprintf(fileID, 'COORDINATES\r\n');
fprintf(fileID, '%.0f %.10f %.10f %.10f\r\n', coordenadas_10');
fprintf(fileID, 'END COORDINATES\r\n');
fprintf(fileID, 'ELEMENTS\r\n');
fprintf(fileID, '%.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f
%.0f\r\n', elementos_ext_10');
fprintf(fileID, 'END ELEMENTS\r\n');
fprintf(fileID, 'MESH "TETRAEDROS 10 NODOS CAPA 2 (GEL)" DIMENSION 3
ELEMTYPE Tetrahedra Nnode 10\r\n');
fprintf(fileID, 'COORDINATES\r\n');
fprintf(fileID, 'END COORDINATES\r\n');
fprintf(fileID, 'ELEMENTS\r\n');
fprintf(fileID, '%.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f
%.0f\r\n', elementos_capa2_10');
fprintf(fileID, 'END ELEMENTS\r\n');
fprintf(fileID, 'MESH "TETRAEDROS 10 NODOS CAPA 1 (GEL)" DIMENSION 3
ELEMTYPE Tetrahedra Nnode 10\r\n');
fprintf(fileID, 'COORDINATES\r\n');
fprintf(fileID, 'END COORDINATES\r\n');
fprintf(fileID, 'ELEMENTS\r\n');
fprintf(fileID, '%.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f
%.0f\r\n', elementos_capa1_10');
fprintf(fileID, 'END ELEMENTS\r\n');
fprintf(fileID, 'MESH "TETRAEDROS 10 NODOS ESFERA (SPROUT)" DIMENSION
3 ELEMTYPE Tetrahedra Nnode 10\r\n');
fprintf(fileID, 'COORDINATES\r\n');
fprintf(fileID, 'END COORDINATES\r\n');
fprintf(fileID, 'ELEMENTS\r\n');
fprintf(fileID, '%.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f
%.0f\r\n', elementos_celula_10');
fprintf(fileID, 'END ELEMENTS\r\n');
fclose(fileID);
otherwise
error(' El número de nodos seleccionado no es correcto')
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Includido por José Antonio Sanz %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
scal = 5.0e4;
fa_dens = 1.0;
```

```
n_cel = intersect(elem_capa1(:,1:4), elem_celula(:,1:4));
```

```
bis = ismember(fc1(:,1:3), n_cel);
[ii1, jj1] = find(bis == 1);
ffc1 = fc1(unique(ii1), 1:3);
```

```
if exist('KB_sel.mat') == 2
load('KB_sel.mat');
```

```

else
    node_cel = node(n_cel,:);
    ind_aux = find(node_cel(:,2)>45);
    KBB = n_cel(ind_aux);

    numelements = round(fa_dens/100*length(KBB));
    % get the randomly-selected indices
    indices = randperm(length(KBB));
    indices = indices(1:numelements);
    % choose the subset of a you want

%     KB_sel2 = node(KBB(indices),1:3);           % sin vecino

    neig0 = KBB(indices);                         % 1 vecino
    [indices2] = vecino3(neig0,ffc1);              % 1 vecino
    KB_sel2 = node(indices2,1:3);                 % 1 vecino

%     neig0 = KBB(indices);                       % 2 vecino
%     [neig1] = vecino3(neig0,ffc1);              % 2 vecino
%     [indices2] = vecino3(neig1,ffc1);          % 2 vecino
%     KB_sel2 = node(indices2,1:3);              % 2 vecino

    save('KB_sel','KB_sel2');
end

KB_sel = dsearchn(node,KB_sel2);

FC_sel = [zeros(length(KB_sel),1) -
ones(length(KB_sel),1)*scal/length(KB_sel) ...
zeros(length(KB_sel),1)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

coordinates = [node(:,1),node(:,2),node(:,3)];

elem0 = [elementos_capa1;elementos_capa2;elementos_ext];
elem = elementos_celula;

elements = [elem0;elem];

%%%

node_10 = coordenadas_10(:,2:end);

elem0_10 = [elementos_capa1_10;elementos_capa2_10;elementos_ext_10];
elem_10 = elementos_celula_10;

elements_10 = [elem0_10;elem_10];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xmin = min(node(:,1));
xmax = max(node(:,1));

ymin = min(node(:,2));

```

```

ymax = max(node(:,2));

zmin = min(node(:,3));
zmax = max(node(:,3));

ind_x = find(node(:,1)==xmin | node(:,1)==xmax);
ind_y = find(node(:,2)==ymin | node(:,2)==ymax);
ind_z = find(node(:,3)==zmin | node(:,3)==zmax);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Output %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fid = fopen(strcat(filename, '.inp'), 'wt');
fprintf(fid, '*HEADING');
fprintf(fid, '\n', '');
fprintf(fid, '*NODE,NSET=NTODOS');
fprintf(fid, '\n', '');

nnode_10 = 1:size(node_10,1);

rd=[nnode_10', node_10(:,1), node_10(:,2), node_10(:,3)];
fprintf(fid, '%7i, %12.7f, %12.7f, %12.7f\n', rd);

fprintf(fid, '*ELEMENT,TYPE=C3D10,ELSET=GEL');
fprintf(fid, '\n', '');

rr=[elem0_10(:,1), elem0_10(:,2), elem0_10(:,3), elem0_10(:,4),...
    elem0_10(:,5), elem0_10(:,6), elem0_10(:,7), elem0_10(:,8),...
    elem0_10(:,9), elem0_10(:,10), elem0_10(:,11)];
fprintf(fid, '%7i, %7i, %7i, %7i, %7i, %7i, %7i, %7i, %7i, %7i\n', rr);

fprintf(fid, '*ELEMENT,TYPE=C3D10,ELSET=SPROUT');
fprintf(fid, '\n', '');

clear rr

rr=[elem_10(:,1), elem_10(:,2), elem_10(:,3), elem_10(:,4), elem_10(:,5),...
    elem_10(:,6), elem_10(:,7), elem_10(:,8), elem_10(:,9),...
    elem_10(:,10), elem_10(:,11)];
fprintf(fid, '%7i, %7i, %7i, %7i, %7i, %7i, %7i, %7i, %7i, %7i\n', rr);

fprintf(fid, '*NSET,NSET=SELECTED');
fprintf(fid, '\n', '');
fprintf(fid, '%7i\n', KB_sel);

% fprintf(fid, '*NSET,NSET=BRANCH2');
% fprintf(fid, '\n', '');
% fprintf(fid, '%7i\n', nod_th_sp(KB2));
%
% fprintf(fid, '*NSET,NSET=BRANCH3');
% fprintf(fid, '\n', '');
% fprintf(fid, '%7i\n', nod_th_sp(KB3));
%

```



```

% fprintf(fid, '*NSET,NSET=BRANCH4');
% fprintf(fid, '\n', '');
% fprintf(fid, '%7i\n', nod_th_sp(KB4));
%
% fprintf(fid, '*NSET,NSET=BRANCHES');
% fprintf(fid, '\n', '');
% fprintf(fid, '%7i\n', KBB);
%
% fprintf(fid, '*NSET,NSET=SPR_SURF');
% fprintf(fid, '\n', '');
% fprintf(fid, '%7i\n', nod_th_sp);

fprintf(fid, '*SOLID SECTION, ELSET=SPROUT, MATERIAL=MATPROPS2');
fprintf(fid, '\n', '');
fprintf(fid, '*SOLID SECTION, ELSET=GEL, MATERIAL=MATPROPS');
fprintf(fid, '\n', '');
fprintf(fid, '*MATERIAL,NAME=MATPROPS');
fprintf(fid, '\n', '');
fprintf(fid, '*ELASTIC,TYPE=ISOTROPIC');
fprintf(fid, '\n', '');
fprintf(fid, '100.0, 4.50000E-01');
fprintf(fid, '\n', '');
fprintf(fid, '*MATERIAL,NAME=MATPROPS2');
fprintf(fid, '\n', '');
fprintf(fid, '*ELASTIC,TYPE=ISOTROPIC');
fprintf(fid, '\n', '');
fprintf(fid, '100.0e-6, 4.50000E-01');
fprintf(fid, '\n', '');

fprintf(fid, '*BOUNDARY');
fprintf(fid, '\n', '');

clear rr

rr=[ind_x,ones(length(ind_x),2),zeros(length(ind_x),1); ind_y,...
    2*ones(length(ind_y),2),zeros(length(ind_y),1);...
    ind_z,3*ones(length(ind_z),2),zeros(length(ind_z),1)];
fprintf(fid, '%7i, %7i, %7i, %12.7f\n', rr');

fprintf(fid, '*STEP,INC=1000');
fprintf(fid, '\n', '');
fprintf(fid, '*STATIC,DIRECT');
fprintf(fid, '\n', '');
fprintf(fid, '1.0,1.0');
fprintf(fid, '\n', '');
fprintf(fid, '*CLOAD');
fprintf(fid, '\n', '');

clear rr

rr=[KB_sel,ones(length(KB_sel),1),FC_sel(:,1); KB_sel,...
    2*ones(length(KB_sel),1),FC_sel(:,2); KB_sel,...
    3*ones(length(KB_sel),1),FC_sel(:,3)];
fprintf(fid, '%7i, %7i, %12.7f\n', rr');

%
%
% B = {'BX'};
% aux(1:size(elem,1),1) = B;

```

```

% aux2=string(aux);
%
% rr = [ind_e1 aux2 BX(:)];
%
% fprintf(fid,'%s, %s, %s\n',rr');
%
% B = {'BY'};
% aux(1:size(elem,1),1) = B;
% aux2=string(aux);
%
% rr = [ind_e1 aux2 BY(:)];
%
% fprintf(fid,'%s, %s, %s\n',rr');
%
% B = {'BZ'};
% aux(1:size(elem,1),1) = B;
% aux2=string(aux);
%
% rr = [ind_e1 aux2 BZ(:)];
%
% fprintf(fid,'%s, %s, %s\n',rr');

fprintf(fid,'*NODE FILE');
fprintf(fid,'\n','');
fprintf(fid,'U');
fprintf(fid,'\n','');
fprintf(fid,'*EL FILE,ELSET=GEL,POSITION=AVERAGED AT NODES');
fprintf(fid,'\n','');
fprintf(fid,'S');
fprintf(fid,'\n','');
fprintf(fid,'*END STEP');
fprintf(fid,'\n','');

% fprintf(fid,'*STEP');
% fprintf(fid,'\n','');
% fprintf(fid,'*MATRIX GENERATE, STIFFNESS');
% fprintf(fid,'\n','');
% fprintf(fid,'*MATRIX OUTPUT, STIFFNESS, FORMAT=COORDINATE');
% fprintf(fid,'\n','');
% fprintf(fid,'*END STEP');
% fprintf(fid,'\n','');

fclose(fid)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

command2 = strcat('abq1 & abq2 & abaqus j=',filename,' user=wrt_us.for');
status=dos(command2)

pause(10);

% Wait for file to be created.
maxSecondsToWait = 36000; % Wait 10 hours at most.
secondsWaitedSoFar = 0;
while secondsWaitedSoFar < maxSecondsToWait
    if ~exist(strcat(filename,'.lck'),'file')
        break;
    end
    pause(1); % Wait 1 second.
    secondsWaitedSoFar = secondsWaitedSoFar + 1;
end

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
u_abq(1:size(node,1),1:4) = 0;
stress(1:size(node,1),1:6) = 0;
```

```
u_abq = load('displacements.dat');
u_abq(u_abq(:,1),2:end)=u_abq(:,2:end);
u = reshape(u_abq(:,2:4)',size(u_abq,1)*3,1)';
```

```
load('stresses.dat');
%stress = stresses(:,2:end);
stress(stresses(:,1),:) = stresses(:,2:end);
```

```
TF =
tractionforces2(coordinates,elements(:,2:end),ffc1,stress,elem0(:,2:end));
```

```
svm =
sqrt(stress(:,1).*stress(:,1)+stress(:,2).*stress(:,2)+stress(:,3).*stress(:,
3)+ ...
-
(stress(:,1).*stress(:,2)+stress(:,1).*stress(:,3)+stress(:,2).*stress(:,3))+
...
3*(stress(:,4).*stress(:,4)+stress(:,5).*stress(:,5)+stress(:,6).*stress(:,6)
));
```

```
FFI(size(node,1),1:3) = 0.0;
FFI(KB_sel,:) = FC_sel;
```

```
show3(elements,coordinates,u,filename,FFI,TF',svm);
```

Anexo B: Script Mallado_celula_sincapas

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% CÓDIGO DESARROLLADO POR MANUEL ROJAS NARANJO %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% DESCRIPCIÓN:
%SCRIPT DONDE SE CARGA LA GEOMETRIA DE LA CELULA Y SE CREA LA MALLA.
%EN ESTE SRIP SE MALLARÁ LA CÉLULA SOLA, SIN CAPAS ENCERRADA EN UN CUBO.
%LAS CARACTERÍSTICAS DE LA MALLA COMO TAMAÑO Y NUMERO DE NODOS DE LOS
%ELEMENTOS SERÁN LOS PARÁMETROS DE ENTRADA A ESTE SCRIPT.

clear all, clc, close all;
%% DATOS DE ENTRADA
n = 4; %Número de nodos que tiene los elementos de la malla

opt0.radboun = 3; % tamaño del elemento de la superficie de la celula.
%Radio máximo de la esfera de delaunay.

%Le entra al surf2mesh
opt4 = 1000; % Volumen máximo de los elementos que forman el cubo.

filename='Prueba6_sincapas';

%% GEOMETRÍA.
%Cargo la geometría de la célula.
load segmented_sprout.mat
load('segmented_sprout_branch_sections.mat');

Lx = res(1);
Ly = res(2);
Lz = res(3);

mask_all = segmented_sprout.mask_all;

for k=1:size(mask_all,3)
    for j=1:round(Lz/Lx,0)
        mask_all2(:, :, round(Lz/Lx,0)*(k-1)+j) = mask_all(:, :, k);
    end
end

mask3D_0 = mask_all2;

%% MALLADO
addpath(''); %Colocar el directorio donde se encuentre iso2mesh

%Se crea la malla del volumen.
[no0,e10,regions0,holes0]=vol2surf(mask3D_0,1:size(mask3D_0,1),...
    1:size(mask3D_0,2),1:size(mask3D_0,3),opt0,1,'cgalsurf',0.5);

%Se suaviza la malla.
addpath('');% Colocar directorio donde se encuentre smoothpatch_version1b

FV0.vertices = no0;    FV0.faces = e10(:,1:3);

FV00 = smoothpatch(FV0,1,5);

mask3D = mask3D_0;

IID1 = find(mask3D==1);%Para identificar un punto de cada región.

```

```

[ID1, JD1, KD1] = ind2sub(size(mask3D),IID1);%Identifica todos los puntos
%de cada región y lo pone en un formato necesario ¿?

% %Represento superficie
% figure(1)
% plotsurf(FV.vertices,FV.faces);

[node,el1,fc1]=surf2mesh(FV00.vertices,FV00.faces,[-1 -1 -1],...
    [size(mask3D,1)+1 size(mask3D,2)+1 size(mask3D,3)+1],1,opt4,...
    [ID1(1) JD1(1) KD1(1)],[],1);

% figure(2)
% plotmesh(node(:,[2 1 3]),fc1,'facealpha',0.7);% Pinta superficies de la
malla

%% IDENTIFICACIÓN DE NODOS Y ELEMENTOS

%Identifica que elementos pertenece a cada conjunto
elem_ext=el1((el1(:,5)==0),:); %Matriz de elementos tipo 0 con los nodos
%que forman cada elemento. Ultima columna es 0
elem_celula=el1((el1(:,5)==1),:);

ind_e_ext = find(el1(:,5)==0);%vector donde se encuentra las filas de la
%matriz el1 donde hay elementos 0
ind_e_celula = find(el1(:,5)==1);

node = [Lx*node(:,1) Ly*node(:,2) Lz/round(Lz/Lx,0)*node(:,3)];

n_cel = intersect(elem_ext(:,1:4),elem_celula(:,1:4));
node_cel = node(n_cel,:);

node = node - min(node_cel);

%Datos por pantalla
disp(['Número de elementos en la célula = ',num2str(length(elem_celula))])
disp(['Número de caras en la superficie de la célula = ',...
    num2str(length(FV00.faces))])
disp(['Número de elementos exterior = ',num2str(length(elem_ext))])

%% DISTINCIÓN SEGÚN EL N° DE NODOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CASO 4 NODOS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
switch n
    case 4
        %% DATOS DE SALIDA CASO 4 NODOS %%

            coordenadas = [[1:size(node,1)'],'',node];

            elementos_celula = [ind_e_celula,elem_celula(:,1:4)];
            elementos_ext = [ind_e_ext,elem_ext(:,1:4)];

%% ESCRIBIR LOS DATOS (FICHERO GID) CASO 4 NODOS %%

            fileID = fopen(strcat(filename, '.flavia.msh'),'w');
            fprintf(fileID,'MESH "TETRAEDROS 4 NODOS EXTERIOR (GEL)" DIMENSION 3
ELEMENTYPE Tetrahedra Nnode 4\r\n');
            fprintf(fileID,'COORDINATES\r\n');
            fprintf(fileID,'%0f %0.10f %0.10f %0.10f\r\n',coordenadas');

```

```

fprintf(fileID, 'END COORDINATES\r\n');
fprintf(fileID, 'ELEMENTS\r\n');
fprintf(fileID, '%.0f %.0f %.0f %.0f %.0f\r\n', elementos_ext');
fprintf(fileID, 'END ELEMENTS\r\n');
fprintf(fileID, 'MESH "TETRAEDROS 4 NODOS CÉLULA" DIMENSION 3
ELEMENTYPE Tetrahedra Nnode 4\r\n');
fprintf(fileID, 'COORDINATES\r\n');
fprintf(fileID, 'END COORDINATES\r\n');
fprintf(fileID, 'ELEMENTS\r\n');
fprintf(fileID, '%.0f %.0f %.0f %.0f %.0f\r\n', elementos_celula');
fprintf(fileID, 'END ELEMENTS\r\n');
fclose(fileID);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CASO 10 NODOS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 10
%% COLOCAR NUEVOS NODOS %%
%Colocar nuevos nodos para generar elementos tetraédrico de 10 nodos.
s = length(node);
r = length(e11);
%Coordenadas de los nuevos nodos
Coord_Nodos5 = (node(e11(:,1),:)+ node(e11(:,2),:))/2; %Matriz de
%coordenadas de todos los nodos 5 de cada elemento
Coord_Nodos6 = (node(e11(:,2),:)+ node(e11(:,3),:))/2;
Coord_Nodos7 = (node(e11(:,1),:)+ node(e11(:,3),:))/2;
Coord_Nodos8 = (node(e11(:,1),:)+ node(e11(:,4),:))/2;
Coord_Nodos9 = (node(e11(:,2),:)+ node(e11(:,4),:))/2;
Coord_Nodos10 = (node(e11(:,3),:)+ node(e11(:,4),:))/2;
%esta enumeración es la que sigue abaqus

%MATRIZ1 Contiene las coordenadas de los nuevos nodos. Esta por
%bloques las primeras r filas son las coordenadas de los nodos 5,
%despues los de los nodos 6...
MATRIZ1 = [Coord_Nodos5;Coord_Nodos6;Coord_Nodos7;Coord_Nodos8;...
Coord_Nodos9;Coord_Nodos10];
MATRIZ1 = [[s+1:s+6*r]';MATRIZ1];

MATRIZ2 = zeros(size(e11,1),11);
MATRIZ2(:,1) = [1:r]';
MATRIZ2(:,2:5) = e11(:,1:4);
MATRIZ2(:,6) = MATRIZ1(1:r,1);
MATRIZ2(:,7) = MATRIZ1(r+1:2*r,1);
MATRIZ2(:,8) = MATRIZ1(2*r+1:3*r,1);
MATRIZ2(:,9) = MATRIZ1(3*r+1:4*r,1);
MATRIZ2(:,10) = MATRIZ1(4*r+1:5*r,1);
MATRIZ2(:,11) = MATRIZ1(5*r+1:6*r,1);
%MATRIZ2 Contiene todos los elementos, los cuales se han
%pasado de 4 a 10 nodos.

%% ELIMINAR LOS NODOS QUE ESTAN REPETIDOS EN LOS NUEVOS CREADOS%%

[C,ia,ic] = unique(MATRIZ1(:,2:4),'rows','stable');
% Con unique lo que hago es eliminar los nodos repetidos.
ic = ic+s;
MATRIZ1 = [ic(ia),C];
MATRIZ2 = [MATRIZ2(:,1),MATRIZ2(:,2:5),ic(MATRIZ2(:,6:11))-s)];

%% DATOS DE SALIDA CASO 10 NODOS %%

coordenadas_10 = [[1:size(node,1)]',node; MATRIZ1];
%Contiene todos los nodos.
coordenadas = [[1:size(node,1)]',node];

```

```

%coordenadas de los nodos originales

elementos_celula_10 = MATRIZ2(ind_e_celula,:);
elementos_celula = MATRIZ2(ind_e_celula,1:5);

elementos_ext_10 = MATRIZ2(ind_e_ext,:);
elementos_ext = MATRIZ2(ind_e_ext,1:5);

%%% ESCRIBIR LOS DATOS (FICHERO GID) CASO 10 NODOS %%%

fileID = fopen(strcat(filename, '.flavia.msh'), 'w');
fprintf(fileID, 'MESH "TETRAEDROS 10 NODOS EXTERIOR (GEL)" DIMENSION
3 ELEMTYPE Tetrahedra Nnode 10\r\n');
fprintf(fileID, 'COORDINATES\r\n');
fprintf(fileID, '%.0f %.10f %.10f %.10f\r\n', coordenadas_10');
fprintf(fileID, 'END COORDINATES\r\n');
fprintf(fileID, 'ELEMENTS\r\n');
fprintf(fileID, '%.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f
%.0f\r\n', elementos_ext_10');
fprintf(fileID, 'END ELEMENTS\r\n');
fprintf(fileID, 'MESH "TETRAEDROS 10 NODOS ESFERA (SPROUT)" DIMENSION
3 ELEMTYPE Tetrahedra Nnode 10\r\n');
fprintf(fileID, 'COORDINATES\r\n');
fprintf(fileID, 'END COORDINATES\r\n');
fprintf(fileID, 'ELEMENTS\r\n');
fprintf(fileID, '%.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f %.0f
%.0f\r\n', elementos_celula_10');
fprintf(fileID, 'END ELEMENTS\r\n');
fclose(fileID);
otherwise
error(' El número de nodos seleccionado no es correcto')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Includido por José Antonio Sanz %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

scal = 5.0e4;
fa_dens = 1.0;

n_cel = intersect(elem_ext(:,1:4), elem_celula(:,1:4));

bis = ismember(fcl(:,1:3), n_cel);
[ii1, jj1] = find(bis == 1);
ffcl = fcl(unique(ii1), 1:3);

if exist('KB_sel.mat') == 2
load('KB_sel.mat');
else
node_cel = node(n_cel, :);
ind_aux = find(node_cel(:, 2) > 45);
KBB = n_cel(ind_aux);

numelements = round(fa_dens/100*length(KBB));
% get the randomly-selected indices

```



```

indices = randperm(length(KBB));
indices = indices(1:numelements);
% choose the subset of a you want

%     KB_sel2 = node(KBB(indices),1:3);           % sin vecino

neig0 = KBB(indices);           % 1 vecino
[indices2] = vecino3(neig0,ffc1); % 1 vecino
KB_sel2 = node(indices2,1:3);    % 1 vecino

%     neig0 = KBB(indices);           % 2 vecino
%     [neig1] = vecino3(neig0,ffc1);   % 2 vecino
%     [indices2] = vecino3(neig1,ffc1); % 2 vecino
%     KB_sel2 = node(indices2,1:3);    % 2 vecino

save('KB_sel','KB_sel2');
end

KB_sel = dsearchn(node,KB_sel2);

FC_sel = [zeros(length(KB_sel),1)...
          -ones(length(KB_sel),1)*scal/length(KB_sel) zeros(length(KB_sel),1)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

coordinates = [node(:,1),node(:,2),node(:,3)];

elem0 = elementos_ext;
elem = elementos_celula;

elements = [elem0;elem];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xmin = min(node(:,1));
xmax = max(node(:,1));

ymin = min(node(:,2));
ymax = max(node(:,2));

zmin = min(node(:,3));
zmax = max(node(:,3));

ind_x = find(node(:,1)==xmin | node(:,1)==xmax);
ind_y = find(node(:,2)==ymin | node(:,2)==ymax);
ind_z = find(node(:,3)==zmin | node(:,3)==zmax);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Output %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

fid = fopen(strcat(filename, '.inp'), 'wt');
fprintf(fid, '*HEADING');
fprintf(fid, '\n', '');
fprintf(fid, '*NODE, NSET=NTODOS');
fprintf(fid, '\n', '');

nnode = 1:size(node,1);

rd=[nnode', node(:,1), node(:,2), node(:,3)];
fprintf(fid, '%7i, %12.7f, %12.7f, %12.7f\n', rd);

fprintf(fid, '*ELEMENT, TYPE=C3D4, ELSET=GEL');
fprintf(fid, '\n', '');

rr=[elem0(:,1), elem0(:,2), elem0(:,3), elem0(:,4), elem0(:,5)];
fprintf(fid, '%7i, %7i, %7i, %7i, %7i\n', rr);

fprintf(fid, '*ELEMENT, TYPE=C3D4, ELSET=SPROUT');
fprintf(fid, '\n', '');

clear rr

rr=[elem(:,1), elem(:,2), elem(:,3), elem(:,4), elem(:,5)];
fprintf(fid, '%7i, %7i, %7i, %7i, %7i\n', rr);

fprintf(fid, '*NSET, NSET=SELECTED');
fprintf(fid, '\n', '');
fprintf(fid, '%7i\n', KB_sel);

% fprintf(fid, '*NSET, NSET=BRANCH2');
% fprintf(fid, '\n', '');
% fprintf(fid, '%7i\n', nod_th_sp(KB2));
%
% fprintf(fid, '*NSET, NSET=BRANCH3');
% fprintf(fid, '\n', '');
% fprintf(fid, '%7i\n', nod_th_sp(KB3));
%
% fprintf(fid, '*NSET, NSET=BRANCH4');
% fprintf(fid, '\n', '');
% fprintf(fid, '%7i\n', nod_th_sp(KB4));
%
% fprintf(fid, '*NSET, NSET=BRANCHES');
% fprintf(fid, '\n', '');
% fprintf(fid, '%7i\n', KBB);
%
% fprintf(fid, '*NSET, NSET=SPR_SURF');
% fprintf(fid, '\n', '');
% fprintf(fid, '%7i\n', nod_th_sp);

fprintf(fid, '*SOLID SECTION, ELSET=SPROUT, MATERIAL=MATPROPS2');
fprintf(fid, '\n', '');
fprintf(fid, '*SOLID SECTION, ELSET=GEL, MATERIAL=MATPROPS');
fprintf(fid, '\n', '');
fprintf(fid, '*MATERIAL, NAME=MATPROPS');
fprintf(fid, '\n', '');
fprintf(fid, '*ELASTIC, TYPE=ISOTROPIC');

```

```

fprintf(fid, '\n', '');
fprintf(fid, '100.0, 4.50000E-01');
fprintf(fid, '\n', '');
fprintf(fid, '*MATERIAL,NAME=MATPROPS2');
fprintf(fid, '\n', '');
fprintf(fid, '*ELASTIC,TYPE=ISOTROPIC');
fprintf(fid, '\n', '');
fprintf(fid, '100.0e-6, 4.50000E-01');
fprintf(fid, '\n', '');

fprintf(fid, '*BOUNDARY');
fprintf(fid, '\n', '');

clear rr

rr=[ind_x,ones(length(ind_x),2),zeros(length(ind_x),1); ind_y,...
    2*ones(length(ind_y),2),zeros(length(ind_y),1); ind_z,...
    3*ones(length(ind_z),2),zeros(length(ind_z),1)];
fprintf(fid, '%7i, %7i, %7i, %12.7f\n',rr');

fprintf(fid, '*STEP,INC=1000');
fprintf(fid, '\n', '');
fprintf(fid, '*STATIC,DIRECT');
fprintf(fid, '\n', '');
fprintf(fid, '1.0,1.0');
fprintf(fid, '\n', '');
fprintf(fid, '*CLOAD');
fprintf(fid, '\n', '');

clear rr

rr=[KB_sel,ones(length(KB_sel),1),FC_sel(:,1); KB_sel,...
    2*ones(length(KB_sel),1),FC_sel(:,2);...
    KB_sel,3*ones(length(KB_sel),1),FC_sel(:,3)];
fprintf(fid, '%7i, %7i, %12.7f\n',rr');

%
%
% B = {'BX'};
% aux(1:size(elem,1),1) = B;
% aux2=string(aux);
%
% rr = [ind_e1 aux2 BX(:)];
%
% fprintf(fid, '%s, %s, %s\n',rr');
%
% B = {'BY'};
% aux(1:size(elem,1),1) = B;
% aux2=string(aux);
%
% rr = [ind_e1 aux2 BY(:)];
%
% fprintf(fid, '%s, %s, %s\n',rr');
%
% B = {'BZ'};
% aux(1:size(elem,1),1) = B;
% aux2=string(aux);
%
% rr = [ind_e1 aux2 BZ(:)];

```

```

%
% fprintf(fid,'%s, %s, %s\n',rr');

fprintf(fid,'*NODE FILE');
fprintf(fid,'\n','');
fprintf(fid,'U');
fprintf(fid,'\n','');
fprintf(fid,'*EL FILE,ELSET=GEL, POSITION=AVERAGED AT NODES');
fprintf(fid,'\n','');
fprintf(fid,'S');
fprintf(fid,'\n','');
fprintf(fid,'*END STEP');
fprintf(fid,'\n','');

% fprintf(fid,'*STEP');
% fprintf(fid,'\n','');
% fprintf(fid,'*MATRIX GENERATE, STIFFNESS');
% fprintf(fid,'\n','');
% fprintf(fid,'*MATRIX OUTPUT, STIFFNESS, FORMAT=COORDINATE');
% fprintf(fid,'\n','');
% fprintf(fid,'*END STEP');
% fprintf(fid,'\n','');

fclose(fid)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

command2 = strcat('abq1 & abq2 & abaqus j=',filename,' user=wrt_us.for');
status=dos(command2)

pause(10);

% Wait for file to be created.
maxSecondsToWait = 36000; % Wait 10 hours at most.
secondsWaitedSoFar = 0;
while secondsWaitedSoFar < maxSecondsToWait
    if ~exist(strcat(filename,'.lck'),'file')
        break;
    end
    pause(1); % Wait 1 second.
    secondsWaitedSoFar = secondsWaitedSoFar + 1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

u_abq(1:size(node,1),1:4) = 0;
stress(1:size(node,1),1:6) = 0;

u_abq = load('displacements.dat');
u_abq(u_abq(:,1),2:end)=u_abq(:,2:end);
u = reshape(u_abq(:,2:4)',size(u_abq,1)*3,1)';

load('stresses.dat');
%stress = stresses(:,2:end);
stress(stresses(:,1),:) = stresses(:,2:end);

```

```
TF =
tractionforces2(coordinates,elements(:,2:end),ffc1,stress,elem0(:,2:end));

svm =
sqrt(stress(:,1).*stress(:,1)+stress(:,2).*stress(:,2)+stress(:,3).*stress(:,
3)+ ...
-
(stress(:,1).*stress(:,2)+stress(:,1).*stress(:,3)+stress(:,2).*stress(:,3))+
...
3*(stress(:,4).*stress(:,4)+stress(:,5).*stress(:,5)+stress(:,6).*stress(:,6)
));

FFI(size(node,1),1:3) = 0.0;
FFI(KB_sel,:) = FC_sel;

show3(elements,coordinates,u',filename,FFI,TF',svm);
```

REFERENCIAS

- [1] Huw Colin-York and Marco Fritzsche. The future of traction force microscopy. *Current Opinion in Biomedical Engineering*. 2018; Volume 5, pages 1-5.
- [2] Matthew S. Hall, Rong Long, Xinzeng Feng, YuLing Huang, Chung-Yuen Hui, and Mingming Wu. Towards Single Cell Traction Microscopy within 3D Collagen Matrices. *Experimental Cell Research*. 2013; Volume 319, pages 2396-2408.
- [3] Xin Tang, Alireza Tofangchi, Sandeep V. Anand, Taher A. Saif1. A Novel Cell Traction Force Microscopy to Study Multi-Cellular System. *Plos Computational Biology*. 2014; Volume 10.
- [4] Ulrich S. Schwarz, Jérôme R.D. Soiné. Traction force microscopy on soft elastic substrates: A guide to recent computational advances. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research*. 2015; Volume 1853, pages 3095-3104.
- [5] <https://www.eurekalert.org/multimedia/pub/54563.php>.
- [6] Christian Franck, Stacey A. Maskarinec, David A. Tirrell, Guruswami Ravichandran. Three-Dimensional Traction Force Microscopy: A New Tool for Quantifying Cell-Matrix Interactions. *Plos One*. 2011; Volume 6.
- [7] Sung Sik Hur, Yihua Zhao, Yi-shuan Li, Elliot Botvinick, and Shu Chien. Live Cells Exert 3-Dimensional Traction Forces on Their Substrata. *Cellular and Molecular Bioengineering*. 2009; Volume 2, pages 425-436.
- [8] Robert W. Style, Rostislav Boltianskiy, Guy K. German, Callen Hyland, Christopher W. MacMinn, Aaron F. Mertz, Larry A. Wilen, Ye Xu, and Eric R. Dufresne. Traction force microscopy in physics and biology. *Soft Matter*. 2014; Volume 10, pages 4047
- [9] <https://bitesizebio.com/19958/what-is-confocal-laser-scanning-microscopy/>
- [10] Zhaochun Yanga, Jeen-Shang Linb, Jianxin Chena, James H-C. Wangc. Determining substrate displacement and cell traction fields—a new approach. *Journal of Theoretical Biology*. 2006; Volume 242, pages 607-616.
- [11] C. Franck & S. Hong & S.A. Maskarinec & D.A. Tirrell & G. Ravichandran. Three-dimensional Full-field Measurements of Large Deformations in Soft Materials Using Confocal Microscopy and Digital Volume Correlation. *Experimental Mechanics*. 2007; Volume 47, pages 427-438.
- [12] Wesley R. Legant, Colin K. Choi, Jordan S. Miller, Lin Shao, Liang Gao, Eric Betzig, and Christopher S. Chen. Multidimensional traction force microscopy reveals out-of-plane rotational moments about focal adhesions. *PNAS*. 2013; Volume 110, pages 881-886.
- [13] <https://bitesizebio.com/13408/which-light-microscope-will-i-choose-part-2-confocal-systems/>
- [14] <https://bitesizebio.com/33529/fluorescence-microscopy-the-magic-of-fluorophores-and-filters/>
- [15] <https://www.scientifica.uk.com/learning-zone/widefield-fluorescence-microscopy>

- [16] Micah Dembo and Yu-Li Wang. Stresses at the Cell-to-Substrate Interface during Locomotion of Fibroblasts. *Biophysical Journal*. 1999; Volume 76, pages 2307-2317.
- [17] Wesley R Legant, Jordan S Miller, Brandon L Blakely, Daniel M Cohen, Guy M Genin & Christopher S Chen. Measurement of mechanical tractions exerted by cells in three-dimensional matrices. *Nature Methods*. 2010; Volume 7, pages 969-971.
- [18] Benedikt Sabass, Margaret L. Gardel, Clare M. Waterman, and Ulrich S. Schwarz. High Resolution Traction Force Microscopy Based on Experimental and Computational Advances. *Biophysical Journal*. 2008; Volume 94, pages 207-220.
- [19] . U. S. Schwarz, N. Q. Balaban, D. Riveline, A. Bershadsky, B. Geiger, and S. A. Safran. Calculation of Forces at Focal Adhesions from Elastic Substrate Data: The Effect of Localized Force and the Need for Regularization. *Biophysical Journal*. 2002; Volume 83, pages 1380-1394.
- [20] G. Vitale, L. Preziosi, and D. Ambrosi. A numerical method for the inverse problem of cell traction in 3D. *Inverse Problems*. 2012; Volume 28.
- [21] G. Vitale, L. Preziosi, and D. Ambrosi. Force traction microscopy: An inverse problem with pointwise observations. *Journal of Mathematical Analysis and Applications*. *Journal of Mathematical Analysis and Applications*. 2012; Volume 395, pages 788-801.
- [22] Thorsten M. Koch, Stefan Münster, Navid Bonakdar, James P. Butler, Ben Fabry. 3D Traction Forces in Cancer Cell Invasion. *Plos One*. 2012. Volume 7.
- [23] https://en.wikipedia.org/wiki/Single-particle_tracking
- [24] http://web.mit.edu/calculix_v2.7/CalculiX/ccx_2.7/doc/ccx/node33.html
- [25] https://es.wikipedia.org/wiki/Triangulaci%C3%B3n_de_Delaunay
- [26] <http://iso2mesh.sourceforge.net/cgi-bin/index.cgi>
- [27] Qianqian Fang and David A. Boas. Tetrahedral mesh generation from volumetric binary and grayscale images. 2009.
- [28] http://ocw.uc3m.es/mecanica-de-medios-continuos-y-teoria-de-estructuras/ingenieria-estructural/material-de-clase-1/apuntes/Capitulo_9.Introduccion_al_metodo_de_los_elementos_finitos.pdf
- [29] Eugenio Oñate. Cálculo de Estructuras por el Método de Elementos Finitos. Análisis estático lineal.

