



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
Departamento de Física

ANÁLISIS CUANTITATIVO DE LA ORGANIZACIÓN DEL TEJIDO EMBRIONARIO DEL PEZ CEBRA

Tesis para optar al grado académico
de Magíster en Ciencias con Mención en Física

Por: *Joséf Artem Guajardo Bravo*
Director/es de Tesis: *Hernán Andrés Morales Navarrete*
Juan Pablo Staforelli Vivanco

Junio 2024
Concepción, Chile

"Grief, I've learned, is really just love. It's all the love you want to give, but cannot. All of that unspent love gathers in the corners of your eyes, the lump in your throat, and in the hollow part of your chest. Grief is just love with no place to go."

- Jamie Anderson

A la memoria de mis queridos compañeros caninos que ya no están aquí, Perla, Mary, Jenny, Blade, Shiro, Kaori, Mila, Luna.

Agradecimientos

Quiero expresar mi más profundo agradecimiento a mi compañero de vida, César Riquelme, por su amor, paciencia y apoyo incondicional a lo largo de este viaje. Sin su constante aliento y confianza en mis capacidades, no habría podido llegar hasta aquí.

A mis amigos Abelor y Catalina, gracias por estar a mi lado en los momentos más difíciles. Su amistad y apoyo me dieron la fuerza para seguir adelante y superar los obstáculos que se presentaron en mi camino. A todos ustedes, les debo mucho más de lo que las palabras pueden expresar. Gracias por creer en mí.

Al director del programa de magíster, Dr. Julio Oliva, por toda la ayuda en sus gestiones administrativas, que fueron fundamentales para el desarrollo de este trabajo.

Agradezco especialmente también a la Dra. Marcela Torrejón y al Laboratorio de Señalización y Desarrollo de la Facultad de Ciencias Biológicas, Universidad de Concepción, por proporcionar un espacio seguro y confiable donde se promueve la colaboración, la discusión y el intercambio de ideas. Su apoyo constante fue fundamental para mi desarrollo académico y profesional, incluso en un entorno donde soy físico en un mundo de biólogos.

Finalmente quiero agradecer a mis 14 compañeros caninos, cuyo ánimo y amor incondicional han hecho de mi vida un lugar más hermoso, son mi más preciada familia y felicidad.

Índice

1. Introducción	10
2. Capítulo 1: Marco teórico	11
2.1. Materia condensada blanda: una visión general	11
2.2. Explorando mesofases: Cristales líquidos	12
2.2.1. Cristales líquidos formados por moléculas en forma de varilla y la fase nemática.	13
2.2.2. Parámetro de orden para nematicos uniaxiales	14
2.2.3. Parámetro de orden en nematicos biaxiales	15
2.3. Materia activa: una visión general	18
2.3.1. Nemáticos activos	19
2.3.2. Sobre otros fluidos activos	20
2.3.3. Transiciones de fase en fluidos activos	21
2.4. Tejidos biológicos como cristales líquidos nemáticos activos	24
2.5. Embriogénesis temprana del pez cebra	27
2.5.1. Movimientos morfogenéticos durante la embriogénesis del pez cebra. . .	28
2.5.2. Pez cebra como un material activo	30
3. Capítulo 2: Caracterización de la forma celular en tres dimensiones	32
3.1. Reconstrucción en mallas triangulares de imágenes de tres dimensiones	33
3.2. Desarrollo de herramientas computacionales para la cuantificación de las formas celulares en tres dimensiones	37
3.2.1. Cuantificación contacto célula-célula	37
3.2.1.1. Descomposición en modos esféricos	37
3.2.1.2. Validación del algoritmo	39
3.2.1.3. Estudio del efecto de los contactos celulares en la descripción de las formas celulares	46
3.2.2. Caracterización tensorial de la forma celular en tres dimensiones	56
3.2.2.1. Cálculo de tensores, parámetros de forma y ejes nematicos . . .	57

3.2.3.	Análisis espacial de la forma celular	58
3.2.3.1.	Modelado de grano grueso para los parámetros de orden	58
3.2.3.2.	Orden orientacional en sistemas no homogéneos	60
4.	Capítulo 3: Validación del método para datos artificiales	63
4.1.	Cascarón de células	66
4.2.	Análisis de los parámetros de orden	68
5.	Capítulo 4: Aplicación en datos reales de gastrulación temprana del pez cebra	71
5.1.	Datos reales tridimensionales del embrión del pez cebra	71
5.1.1.	Análisis de descomposición en modos esféricos	72
5.1.2.	Análisis parámetros de orden	74
6.	Cápítulo 5: Conclusiones y Discusión	78
7.	Apéndice	84
7.1.	Código para pre-procesar los datos	84
7.2.	Código para calcular la descomposición en modos esféricos y los tensores nematicos	87
7.3.	Código para calcular los parámetros de orden	101

Índice de figuras

1.	Representación de orden orientacional de liquido isotrópico, cristal liquido (orden nemático) y solido anisótropico, con n un vector director unitario.	13
2.	n es el vector director y u es el vector unitario que se define a lo largo del eje mas grande de la partícula elipsoidal / en forma de varilla. Se tendrá un alineamiento perfecto cuando $S = 1$, y no habrá alineamiento cuando $S \sim 0$	15
3.	l, m, n ejes de una molécula anisotrópica 3D, adaptado de [5]	16
4.	Defectos topológicos semi-enteros característicos de los nemáticos activos, adaptado desde [7]	19
5.	Diferentes tipos de partículas llevan a diferentes tipos de fluidos activos, adaptado de [7]	21
6.	Diagrama de transiciones vítreas, adaptado de Altieri, A. [14]	22

7.	Diagrama del eje del pez cebra, adaptado desde Marelli <i>et al.</i> , [24]	27
8.	Movimientos morfogenéticos en el desarrollo embrionario del pez cebra, adaptado de Webb [25]	28
9.	Tejido complejo que encapsula defectos topológicos, movimiento polarizado, organización nemática y diferentes fases de la materia, adaptado desde Balasubramaniam <i>et al.</i> [26]	32
10.	Reconstrucción desde imagen Watershed	35
11.	Descomposición multipolar de patrón superficial, la distribución sobre la esfera es reconstruida en la suma de los modos esféricos, adaptado de Scholich <i>et al.</i>	38
12.	Esquema de célula y sus células vecinas con opacidad mas baja, sobre ella una distribución de colores representando la distribución de contactos con sus vecinos que se proyecta sobre una esfera unitaria	39
13.	Distribución homogénea sobre una esfera.	40
14.	Distribución tipo polo norte sobre una esfera.	40
15.	Distribución tipo polo sur sobre una esfera.	40
16.	Distribución uni-polar vertical sobre una esfera.	41
17.	Distribución bipolar sobre una esfera.	41
18.	Distribución tipo anillo sobre una esfera.	41
19.	Distribución tipo anillo vertical sobre una esfera.	42
20.	Distribución tipo dos polos verticales sobre una esfera.	42
21.	Ejemplos de distribuciones de Elipsoides con distintos valores en sus semiejes	43
22.	Ejemplo de distribuciones de Cuboides de diferente elongación sin contactos.	44
23.	Dodecaedro caras.	44
24.	Análisis caso especial.	45
25.	Distribuciones diferentes caras con contacto.	46
26.	Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en esferas.	47
27.	Gráfico individual de cada modo y su respectivo nivel de contacto y elongación (todas elongaciones iguales para el caso de las esferas).	48

28.	Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en los elipsoides.	49
29.	Gráficos individuales por modo para cada elongación y nivel de contacto. Gráfico en conjunto de todos los modos y análisis en mapa de calor de los modos para cada caso.	49
30.	Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en elipsoides.	50
31.	Gráficos individuales por modo para cada elongación y nivel de contacto. Gráfico en conjunto de todos los modos y análisis en mapa de calor de los modos para cada caso.	50
32.	Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en elipsoides.	51
33.	Gráficos individuales por modo para cada elongación y nivel de contacto. Gráfico en conjunto de todos los modos y análisis en mapa de calor de los modos para cada caso.	52
34.	Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en cuboides	53
35.	Gráficos individuales por modo para cada elongación y nivel de contacto. Gráfico en conjunto de todos los modos y análisis en mapa de calor de los modos para cada caso. Se observa presencia sólo del modo F_4	53
36.	Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en cuboides	54
37.	Gráficos individuales por modo para cada elongación y nivel de contacto. Gráfico en conjunto de todos los modos y análisis en mapa de calor de los modos para cada caso. Se observa presencia sólo del modo F_4	54

38.	Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en cuboides.	55
39.	Gráficos individuales por modo para cada elongación y nivel de contacto. Gráfico en conjunto de todos los modos y análisis en mapa de calor de los modos para cada caso. Se observa presencia sólo del modo F_4	55
40.	Análisis de descomposición en modos esféricos.	64
41.	Comparación cualitativa primer autovector tensor nemático y tensor con modelado de grano grueso usando filtro gaussiano euclidiano.	64
42.	Análisis en descomposición de modos esféricos para el domo de células.	65
43.	Comparación cualitativa primer autovector tensor nemático y tensor con modelado de grano grueso usando filtro esférico compacto 3D	66
44.	Análisis en descomposición de modos esféricos para un cascarón de células.	67
45.	Comparación cualitativa primer autovector tensor nemático y tensor con modelado de grano grueso usando filtro esférico compacto 3D	67
46.	Gráficos de violin de las distribuciones de descomposición en modos esfericos para cada grupo de datos	68
47.	Descomposición en modos esféricos para diferentes tiempos	72
48.	Gráfico de primer autovector de los tensores nematicos antes del modelado de grano grueso para distintos tiempos	74
49.	Gráfico de primer autovector de los tensores nematicos de modelado de grano grueso para distintos tiempos	74
50.	Análisis parámetros de orden para diferentes tiempos	75

Resumen

En esta tesis, se realizó un análisis cuantitativo de la organización del tejido embrionario del pez cebra durante la etapa de desarrollo temprano. Para ello, se emplearon la teoría de los cristales líquidos y herramientas matemáticas avanzadas para estudiar el comportamiento de conjuntos celulares. Se desarrollaron herramientas computacionales que permitieron procesar imágenes, identificar contactos celulares y analizar sus comportamientos mediante la descomposición de modos esféricos. Posteriormente, se caracterizaron estos comportamientos de manera tensorial utilizando la teoría de tensores de textura de la ciencia de materiales. La validez de estos métodos se comprobó mediante geometrías simples y datos artificiales, aplicándolos finalmente a datos reales del artículo Shah, G. et al. « Multi-scale imaging and analysis identify pan-embryo cell dynamics of germlayer formation in zebrafish ».

Abstract

In this thesis, a quantitative analysis of the organization of zebrafish embryonic tissue during the early gastrulation stage was conducted. The theory of liquid crystals and advanced mathematical tools were employed to study the behavior of cellular assemblies. Computational tools were developed to process images, identify cellular contacts, and analyze their behaviors using spherical mode decomposition. Subsequently, these behaviors were characterized tensorially using the fabric tensor tools from materials science. The validity of these methods was verified with simple geometries and artificial data, and finally, the algorithms were applied to real data.

1. Introducción

Los sistemas biológicos tales como el tejido embrionario son estructuras complejas y altamente dinámicas, que siempre están sujetas a diferentes fenómenos tales como movimientos colectivos de sus componentes (por ejemplo, las células), transiciones de fase y propiedades emergentes. La comunicación celular y las interacciones dinámicas son cruciales para guiar toda la morfogénesis (proceso por el cual el embrión va adquiriendo la forma que le corresponde según su especie) del tejido durante el desarrollo animal.

Sin embargo, el como la morfodinámica (efectos biomecánicos involucrados en el desarrollo) de los tejidos y las propiedades de la mecánica celular se acoplan durante la embriogénesis es aun una pregunta abierta. En las ultimas décadas han habido múltiples intentos para construir campos teóricos basados en principios físicos para tratar de responder esta pregunta. En particular, nuestra actual tecnología nos permite ver procesos biológicos tales como el desarrollo del embrión con una resolución espacio-temporal sin precedentes. Todo esto en conjunto, mas el uso de herramientas computacionales puede ayudarnos a extraer información biofísica multidimensional desde las imágenes de microscopia, y analizando los resultados, ser capaces de entender más acerca de los mecanismos tras procesos complejos como el desarrollo embrionario. En esta tesis, usaremos herramientas teóricas del campo de la materia condensada blanda y la materia activa para describir el tejido embrionario del pez cebra durante su desarrollo temprano. Extraeremos datos de imágenes microscópicas de alta resolución tridimensionales mas la componente temporal, para realizar una descripción cuantitativa de la morfología y el orden estructural del sistema. Primero, introduciremos los conceptos teóricos relevantes tales como la materia activa, cristales líquidos y los defectos topológicos. Luego, explicaremos en detalle el problema biológico y los conceptos fundamentales tras el desarrollo embrionario, y finalmente, los objetivos de nuestro proyecto, metodología y la potencial contribución al área.

2. Capítulo 1: Marco teórico

2.1. Materia condensada blanda: una visión general

La materia blanda, también conocida como materia condensada blanda, es un campo que estudia una variedad de sistemas físicos que se deforman estructuralmente por tensiones térmicas o mecánicas que las hace muy sensible a factores externos. Esto incluye polímeros, coloides, geles, materiales granulares, tensoactivos, cristales líquidos y muchos otros tipos de materiales, como sistemas biológicos (por ejemplo, sangre, músculos, tejidos, bacterias, etc.). Los comportamientos interesantes de la materia blanda surgen del hecho de que no pueden predecirse directamente a partir de sus constituyentes moleculares, esto se debe a que este tipo de materia exhibe nuevas propiedades tales como la «auto-organización o auto-ensamble» en diversas estructuras a escala mesoscópica, en contraste con lo que más tratamos usualmente en física, la llamada materia «dura» condensada, que se organiza en redes cristalinas que nos permiten realizar predicciones sobre el comportamiento general del material en función de sólo sus componentes.[1]

Los materiales antes mencionados comparten en común que todos están compuestos por unidades estructurales que son muchísimo más grandes que sus átomos individuales. Las moléculas de polímeros típicamente están formadas por millones de átomos e incluso, las partículas coloidales de diámetro $0,1[\mu m]$ tienen billones de átomos. Por otro lado las moléculas que constituyen los tensoactivos y los cristales líquidos no son muy grandes (compuestas por decenas o centenas de átomos), pero forman estructuras ordenadas que se mueven en conjunto como una unidad: las moléculas tensoactivas forman micelas que se mueven como una entidad colectiva, así mismo las moléculas líquido cristalinas rotan y se desplazan en conjunto, dando la noción de una gran estructura unitaria[1].

El hecho de que la materia blanda esté formada por moléculas grandes o conjuntos de moléculas que se mueven colectivamente le da dos características importantes e interesantes que normalmente no vemos en la materia condensada usual:

- *Respuesta grande y no lineal.* La materia blanda muestra una gran respuesta a las fuerzas débiles. Por ejemplo, las moléculas de polímeros que constan de decenas de miles de átomos

se deforman fácilmente, lo que da la suavidad o «blandura» a lo que conocemos como cauchos y geles. Las partículas coloidales forman un sólido muy blando que se utiliza en cosméticos y pinturas. Las propiedades ópticas de los cristales líquidos se modifican fácilmente mediante un campo eléctrico lo que los hace útiles para el desarrollo de pantallas. Respuestas tan grandes no pueden describirse mediante relaciones lineales entre la fuerza y la respuesta. Por ejemplo, los cauchos pueden alargarse varias veces el cien por-ciento de su longitud inicial y sus respuestas mecánicas no pueden describirse mediante una relación lineal entre la tensión y la deformación.

- *Respuesta lenta y fuera del equilibrio.* La colectividad de la materia blanda ralentiza su dinámica. El tiempo de respuesta de los líquidos simples es del orden de $10^{-9}[s]$, mientras que puede ser billones o más veces más lento (desde $1[s]$ hasta $10^4[s]$) en soluciones de polímeros o coloides [1]. Consecuentemente, las propiedades y dinámicas del área estadística fuera del equilibrio son muy relevantes en la materia condensada blanda.

Estas propiedades conducen a grandes fluctuaciones térmicas, una amplia variedad de formas y patrones, sensibilidad de las estructuras en equilibrio a las condiciones externas, suavidad macroscópica y estados metaestables.

2.2. Explorando mesofases: Cristales líquidos

Habitualmente en física conocemos y estudiamos detalladamente tres estados particulares de la materia y sus transiciones de uno a otro, gaseoso, líquido y sólido, pero ¿qué pasa con esos estados intermedios? Una mesofase es un estado de la materia intermedio entre líquido y sólido, aquí tenemos las *mesofases cristalinas* que se clasifican comúnmente según su orden de traslación, orientación y conformación como cristales líquidos, cristales plásticos y cristales conformacionalmente desordenados, que representan un estado común de materia condensada. [12]. Como su nombre lo dice, el cristal líquido es un estado intermedio entre lo que se conoce como un líquido isotrópico (desordenado) y un cristal anisótropo (ordenado). Por regla general, una sustancia en este estado es fuertemente anisótropa en algunas de sus propiedades y, sin embargo, presenta un cierto grado de fluidez, que en algunos casos puede ser comparable a la de un líquido normal.

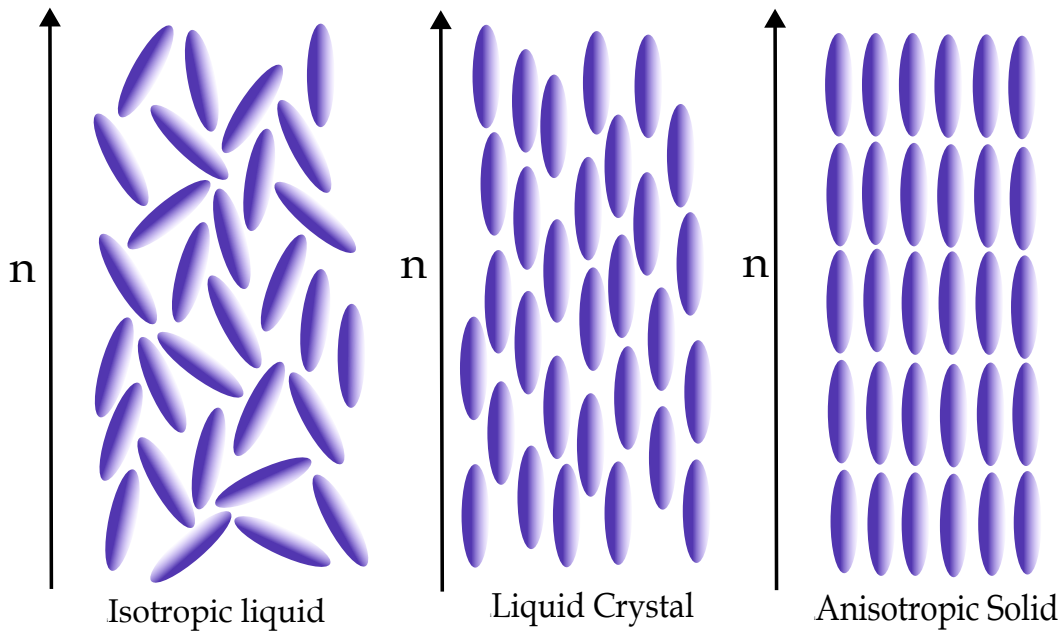


Figura 1: Representación de orden orientacional de liquido isotrópico, cristal liquido (orden nemático) y solido anisótropico, con n un vector director unitario.

Un cristal líquido es un ejemplo que muestra que la naturaleza colectiva de la materia blanda se crea mediante transiciones de fase. Las moléculas que constituyen los cristales líquidos no son grandes (su tamaño es de aproximadamente $1 \text{ [nm]}[1]$), pero en el estado cristalino líquido, estas moléculas se organizan en orden y se mueven colectivamente, creando estructuras conjuntas.

2.2.1. Cristales líquidos formados por moléculas en forma de varilla y la fase nemática.

Hay muchos tipos de orden cristalino líquido, pero aquí nos centraremos en un tipo específico, los nemáticos (Ver figura 1). Los nemáticos son el tipo de cristal líquido más simple, pero es el más utilizado, lo podemos encontrar en los LCDs (pantallas de cristal líquido), por ejemplo.

La gran mayoría de los cristales líquidos termotrópicos están compuestos por moléculas en forma de varilla, en particular, la fase nemática, aquí la orientación de las moléculas es completamente aleatoria y el sistema es un líquido isotrópico, pero, al disminuir la temperatura, las moléculas comienzan a orientarse en una dirección común y formar lo que se denomina "fase nemática". La fase nemática es un líquido anisotrópico: sus propiedades ópticas, eléctricas y magnéticas son anisotrópicas como un cristal, pero el sistema todavía está en un estado líquido que

puede fluir. Esta "mesofase" debe su fluidez a la facilidad con la que las moléculas se deslizan unas sobre otras manteniendo su paralelismo [2].

Entonces, los cristales líquidos nemáticos tienen un alto alcance en el grado de orden de orientación de las moléculas, pero ningún orden de tipo traslación a largo alcance. La dirección preferida generalmente varía de un punto a otro a lo largo del material, pero una muestra uniformemente alineada suele ser uniaxial, positiva y fuertemente birrefringente; el sistema tiene una simetría rotacional alrededor de este eje determinado y se llama director, especificado por un vector unitario \mathbf{n} , donde \mathbf{n} y $-\mathbf{n}$ se consideran totalmente equivalentes.

2.2.2. Parámetro de orden para nemáticos uniaxiales

Basado en la notación utilizada en [1]. Sea \mathbf{u} un vector unitario en la dirección del eje más largo de la molécula, en la fase isotrópica \mathbf{u} estará distribuido uniformemente en una esfera unitaria $\|\mathbf{u}\| = 1$. En la fase nemática, la distribución de \mathbf{u} se vuelve no uniforme sobre la esfera. Ahora, consideremos un parámetro que caracteriza la anisotropía de $\psi(\mathbf{u})$ en la fase nemática. Un candidato obvio para dicho parámetro suele ser el primer momento, $\langle u_\alpha \rangle$, $\langle \alpha = x, y, z \rangle$, donde $\langle \dots \rangle$ representa el promedio de la función de distribución $\psi(\mathbf{u})$,

$$\langle \dots \rangle = \int d^3u \dots \psi(\mathbf{u})$$

Sin embargo, si consideramos moléculas con forma de varilla, que pueden ser moléculas elipsoidales simétricas, el primer momento siempre es igual a cero ya que los estados especificados por \mathbf{u} y $-\mathbf{u}$ son idénticos, y por lo tanto $\psi(\mathbf{u})$ tiene una simetría de inversión $\psi(\mathbf{u}) = \psi(-\mathbf{u})$.

Entonces, nos es conveniente escoger el segundo momento $\langle u_\alpha u_\beta \rangle$, así, en el estado isotrópico tenemos que:

$$\langle u_\alpha u_\beta \rangle = \frac{1}{3} \delta_{\alpha\beta}$$

Por otro lado, si \mathbf{u} está completamente alineado con el vector director \mathbf{n} :

$$\langle u_\alpha u_\beta \rangle = n_\alpha n_\beta \quad (1)$$

Así, consideramos el parámetro,

$$Q_{\alpha\beta} = \left\langle u_\alpha u_\beta - \frac{1}{3} \delta_{\alpha\beta} \right\rangle$$

$Q_{\alpha\beta}$ representa el parámetro de orden orientacional de las moléculas en fase nemática y se le llama justamente *parámetro de orden*. Es cero en la fase isotrópica, y se vuelve no-nulo a medida que transiciona a la fase nemática.

Si la distribución de u tiene una simetría uniaxial alrededor del eje n , $Q_{\alpha\beta}$ puede ser escrito como:

$$Q_{\alpha\beta} = S \left(n_\alpha n_\beta - \frac{1}{3} \delta_{\alpha\beta} \right)$$

Donde S es el parámetro que representa que tan perfecta es la alineación de las moléculas con el eje n . Si es alineamiento es perfecto, S vale 1, si no hay alineamiento, entonces S vale 0.

$$S = \frac{3}{2} \left\langle (u \cdot n)^2 - \frac{1}{3} \right\rangle$$

Entonces, el parámetro de orden *tensorial* $Q_{\alpha\beta}$ incluye dos piezas de información: Uno es que tan fuerte están alineadas las moléculas, representado anteriormente por lo que denominamos S , y en que dirección están alineadas las moléculas, representada por el eje director " n ".

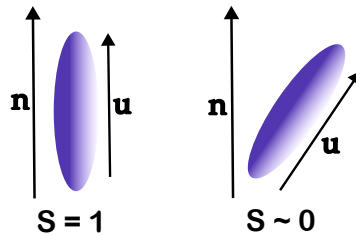


Figura 2: n es el vector director y u es el vector unitario que se define a lo largo del eje mas grande de la partícula elipsoidal / en forma de varilla. Se tendrá un alineamiento perfecto cuando $S = 1$, y no habrá alineamiento cuando $S \sim 0$.

2.2.3. Parámetro de orden en nematicos biaxiales

Cuando estudiamos sistemas que exhiben comportamientos tipo cristal liquido, usualmente sus constituyentes no tienen simetría uniaxial, al contrario suelen poseer tres ejes distintos, lo que

les da el nombre de «cristales líquidos nemáticos biaxiales».

Siguiendo la notación de Scholich en [3], denotamos la triada de auto-vectores ortonormalizados $\mathbf{l}, \mathbf{m}, \mathbf{n}$ con sus respectivos autovalores $\sigma_l, \sigma_m, \sigma_n$ para un tensor nemático que llamaremos $t_{\alpha\beta}$, que toma en cuenta la dirección de las constituyentes del sistema.

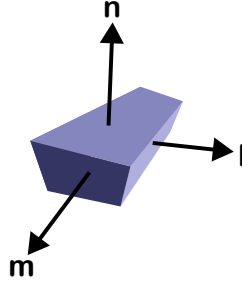


Figura 3: $\mathbf{l}, \mathbf{m}, \mathbf{n}$ ejes de una molécula anisotrópica 3D, adaptado de [5]

Cada tensor nemático \mathbf{t} de un objeto nemático individual puede escribirse como la superposición de dos tensores sin traza, irreducibles y ortogonales \mathbf{q} y \mathbf{b} :

$$\mathbf{t} = \xi_0 \mathbf{q} + \xi_1 \mathbf{b}$$

Esta descomposición separa la parte orientacional codificada en los tensores \mathbf{q} y \mathbf{b} de la anisotropía del tensor \mathbf{t} , cuantificado por dos parámetros de forma $\xi_0 = \sigma_n$ y $\xi_1 = \frac{1}{3}(\sigma_l - \sigma_m)$, que son dados como una combinación lineal de los autovalores $\sigma_l, \sigma_m, \sigma_n$ de \mathbf{t} . Los tensores de orientación \mathbf{q} y \mathbf{b} están dados como combinaciones lineales de diádicas de los vectores propios $\mathbf{l}, \mathbf{m}, \mathbf{n}$ del tensor nemático \mathbf{t} y la matriz identidad \mathbf{I} .

$$\mathbf{q} = \frac{1}{2}(3\mathbf{n} \otimes \mathbf{n} - \mathbf{I}), \quad \mathbf{b} = \frac{3}{2}(\mathbf{l} \otimes \mathbf{l} - \mathbf{m} \otimes \mathbf{m})$$

Donde \otimes denota el producto exterior. Esta descomposición nos permite llevar a cabo conjuntos de promedios sobre las partes orientacionales del sistema:

$$Q = \langle q \rangle, \quad B = \langle b \rangle$$

Los autovalores de estos tensores promediados representan las cuatro invariantes relevantes S, P, D y C del orden orientacional de los objetos que son nemáticos biaxiales, los cuales pueden

escribirse de la siguiente forma en función del promedio de los cosenos direccionales:

$$\begin{aligned}
 S &= \frac{1}{2} \langle 3(\mathbf{n} \cdot \mathbf{z})^2 - 1 \rangle, \\
 P &= \frac{3}{2} \langle (\mathbf{n} \cdot \mathbf{x})^2 - (\mathbf{n} \cdot \mathbf{y})^2 \rangle, \\
 D &= \frac{3}{2} \langle (\mathbf{l} \cdot \mathbf{z})^2 - (\mathbf{m} \cdot \mathbf{z})^2 \rangle, \\
 C &= \frac{1}{2} \langle (\mathbf{l} \cdot \mathbf{x})^2 - (\mathbf{l} \cdot \mathbf{y})^2 + (\mathbf{m} \cdot \mathbf{y})^2 - (\mathbf{m} \cdot \mathbf{x})^2 \rangle.
 \end{aligned}$$

Donde los brackets denotan el promedio del ensamble y $\mathbf{x}, \mathbf{y}, \mathbf{z}$ son *el sistema de referencia* dado por las propiedades de simetría de la fase y que es constante con respecto al promedio del ensamble. Podemos escribir este parámetro como un grupo de tensores nematicos discretos de la siguiente forma:

$$\begin{aligned}
 S &= \frac{1}{2N} \sum_i 3((\mathbf{n}^{(i)} \cdot \mathbf{z})^2 - 1), \\
 P &= \frac{3}{2N} \sum_i ((\mathbf{n}^{(i)} \cdot \mathbf{x})^2 - (\mathbf{n}^{(i)} \cdot \mathbf{y})^2), \\
 D &= \frac{3}{2N} \sum_i ((\mathbf{l}^{(i)} \cdot \mathbf{z})^2 - (\mathbf{m}^{(i)} \cdot \mathbf{z})^2), \\
 C &= \frac{1}{2N} \sum_i ((\mathbf{l}^{(i)} \cdot \mathbf{x})^2 - (\mathbf{l}^{(i)} \cdot \mathbf{y})^2 + (\mathbf{m}^{(i)} \cdot \mathbf{y})^2 - (\mathbf{m}^{(i)} \cdot \mathbf{x})^2)
 \end{aligned}$$

De aquí podemos cuantificar el estado de alineamiento de objetos que no son axialmente simétricos con una generalización de la ecuación para el parámetro de orden de los nematicos uniaxiales, que se conoce como *tensor de ordenamiento*:

$$S_{\alpha\beta}^{ij} = \frac{1}{2} \langle 3i_{\alpha}j_{\beta} - \delta_{\alpha\beta}\delta_{ij} \rangle$$

Donde $j_{\alpha} = \mathbf{e}_j \cdot E_{\alpha}$, $\mathbf{e}_j = \mathbf{a}, \mathbf{b}, \mathbf{c}$ y los ejes del laboratorio son $\mathbf{E}_{\alpha} = \mathbf{x}, \mathbf{y}, \mathbf{z}$. Este tensor de ordenamiento es por construcción definido solo para valores reales, simétrico tanto en los índices i, j como en los α, β , y sin traza en cada par de índices, específicamente $S_{\alpha\alpha}^{ij} = 0$ y $S_{\alpha\beta}^{ii} = 0$, por lo que esta diagonalizado para una elección especial de sistema de referencia ortogonal, al que llamamos anteriormente $\mathbf{l}, \mathbf{m}, \mathbf{n}$. Los elementos no-nulos son dados por combinaciones lineales de

los parámetros de orden S, P, D, C como sigue:

$$\begin{aligned}
 S^{nn}_{zz} &= S \\
 S^{nn}_{yy} &= \frac{1}{2}(-S - P) \\
 S^{nn}_{xx} &= \frac{1}{2}(-S + P) \\
 S^{mm}_{zz} &= \frac{1}{2}(-S - D) \\
 S^{mm}_{yy} &= \frac{1}{4}(S + P + D + 3C) \\
 S^{mm}_{xx} &= \frac{1}{4}(S - P + D - 3C) \\
 S^{ll}_{zz} &= \frac{1}{2}(-S + D) \\
 S^{ll}_{yy} &= \frac{1}{4}(S + P - D - 3C) \\
 S^{ll}_{xx} &= \frac{1}{4}(S - P - D + 3C)
 \end{aligned}$$

2.3. Materia activa: una visión general

La materia activa se compone de sistemas fuera de equilibrio donde sus componentes individuales convierten la energía en fuerzas y movimiento no conservativos. Esto incluye partículas autopropulsadas como coloides impulsados por reacciones químicas a biofilamentos con motores moleculares e incluso bandadas de pájaros.

Describir estos medios utilizando ecuaciones continuas como lo haríamos con la materia pasiva (fluidos y sólidos habituales) requiere un examen cuidadoso de todas las simetrías y leyes de conservación que están presentes en el sistema activo. Dado que los medios activos muestran tensiones no conservativas y flujos espontáneos, naturalmente rompen las simetrías clave que poseerían los materiales pasivos. Por ejemplo, los medios activos pueden violar la reciprocidad, definida como la simetría entre perturbación y respuesta, así como el equilibrio detallado, definido como la simetría entre el pasado y el futuro dentro de la dinámica de procesos microscópicos [6]. Las violaciones al equilibrio de cualquiera de estos principios introducen una flecha del tiempo y permiten la presencia de corrientes que no desaparecen (de energía, momento o partículas) en el estado estacionario de los sistemas activos.

2.3.1. Nemáticos activos

Una importante característica del tipo de sistema que revisaremos en esta tesis tiene que ver con la emergencia de movimientos colectivos y auto-organización, donde grupos de partículas *activas* se mueven en conjunto como una unidad, a escalas mucho mayores que sus tamaños individuales. Aquí introducimos el concepto de *Nemáticos activos*, concepto nacido desde la teoría de los cristales líquidos cuya principal diferencia con lo conocido radica en la *actividad* de su nombre, lo que conduce a la generación espontánea y a la aniquilación de algunas características robustas, que provocan la destrucción del orden nemático de largo alcance y la formación de características de tipo caótico, llamadas *turbulencia activa* sin necesidad de campos externos [6].

Defectos topológicos en nemáticos activos Algunas de estas características robustas se pueden analizar utilizando una rama de las matemáticas llamada topología. La topología describe propiedades de objetos que se conservan bajo continuas deformaciones de su forma, llamadas *invariantes topológicas*. En física de la materia condensada, una aplicación principal de los invariantes topológicos es la caracterización de *defectos topológicos*, que son objetos similares a partículas que describen deformaciones globales de un medio ordenado [7].

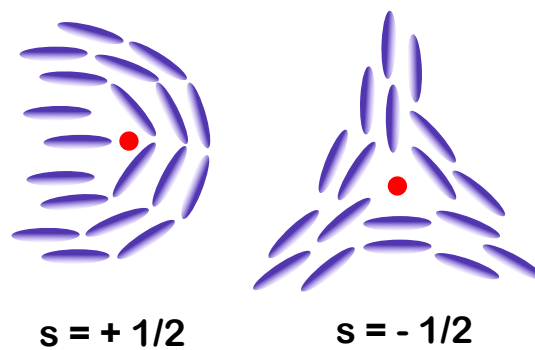


Figura 4: Defectos topológicos semi-enteros característicos de los nemáticos activos, adaptado desde [7]

En los nemáticos activos, los defectos topológicos son ceros especiales del campo de parámetros de orden, se clasifican por su número de vueltas (*winding number*) o carga topológica $s = \Delta\Theta/2\pi$ [7], donde $\Delta\Theta$ es el ángulo neto a través del cual gira el parámetro de orden cuando

se rodea el defecto. La carga s es positiva si el parámetro de orden gira en la misma dirección que el camino recorrido, negativa si gira en la dirección opuesta. En una película nemática, donde la simetría rota identifica solo la orientación (no la dirección) y el parámetro de orden es un campo lineal, los defectos de energía más bajos son variaciones de fuerza $s = \pm 1/2$.

2.3.2. Sobre otros fluidos activos

Hasta este momento acabamos de hablar de sistemas activos con simetría en forma de varilla que dan lugar a cristales líquidos, este tipo de estado de la materia existe en diferentes objetos, siempre que sean anisotrópicos; ya sean alargados, en forma de plátano, en forma de cuenco o en forma de disco, y pueden dar lugar a dos órdenes diferentes, nemáticos o esmécticos [5].

Otras simetrías pueden conducir a diferentes fluidos activos [11], si tenemos una partícula redonda sin alineación, esto nos lleva a materia activa escalar, caracterizada por separaciones de fases inducidas por motilidad descritas por las ecuaciones de Cahn-Hilliard [11] que involucran densidad. Si ahora tenemos una partícula en forma de varilla pero inducimos una polaridad, tenemos un fluido polar activo. El parámetro de orden de conjuntos de partículas activas polares será capturado por un parámetro de orden vectorial con su hidrodinámica descrita por las ecuaciones de Toner-Tu[11]. Curiosamente, las partículas polares pueden conducir al orden polar o al orden nemático.

Las partículas activas también pueden ser quirales y organizarse en estados con quiralidad macroscópica. La quiralidad colectiva es capturada por un campo escalar con rotación intrínseca; son análogos directos de los cristales líquidos colestéricos pasivos. Donde también son posibles estados quirales polares y nemáticos.

El hecho interesante de los fluidos activos con diferentes simetrías es que, dependiendo de las condiciones límite, pueden conducir a diferentes tipos de patrones por sí solos o mediante la interacción con fluidos isotrópicos u otros fluidos activos.

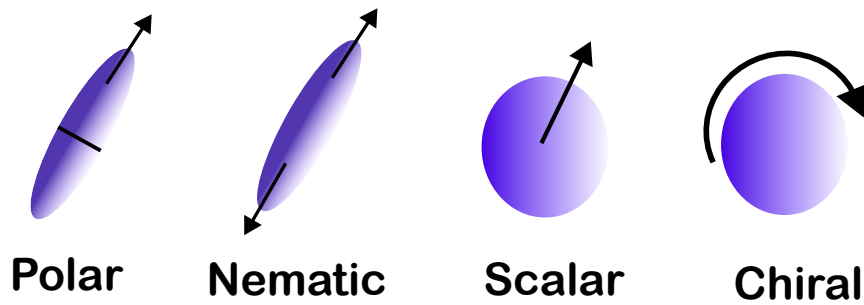


Figura 5: Diferentes tipos de partículas llevan a diferentes tipos de fluidos activos, adaptado de [7]

2.3.3. Transiciones de fase en fluidos activos

Como estado intermedio entre materiales cristalinos y amorfos, las mesofases cristalinas se asemejan a los materiales amorfos en relación a su movilidad molecular, siendo la *transición vítrea* una propiedad común, y al mismo tiempo poseyendo un cierto grado de periodicidad traslacional (con la excepción de la fase nemática). Entonces, la transición vítrea se refiere a una transición de un líquido viscoso a un sólido amorfo (desordenado, no cristalino).

En particular, *la transición de atasco (jamming transition)* es una transición vítrea que ocurre cuando un sistema se atasca, el cual sufre una transición de un estado fluido a uno sólido. A pesar de este importante cambio en la dinámica, la estructura interna del sistema permanece desordenada tanto en fase sólida como fluida, marcando una diferencia fundamental con el conocido proceso de cristalización. Lo que distingue más al sistema atascado del sólido convencional es que al aplicar incluso una pequeña tensión en una dirección determinada, el sistema se desbloquea y las partículas comienzan a fluir[15].

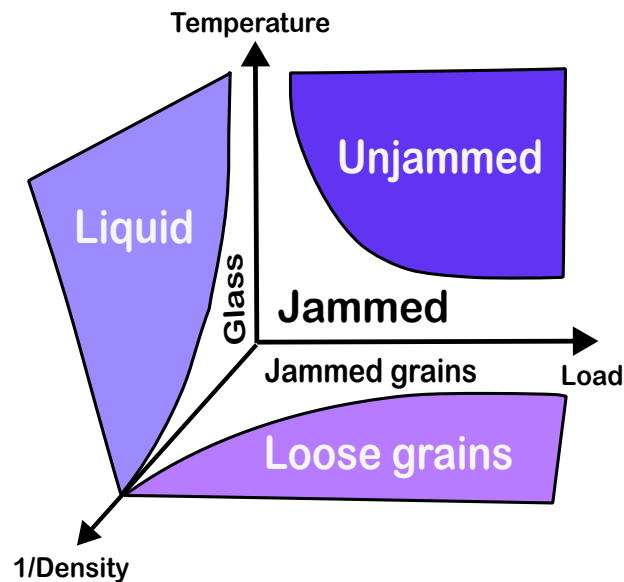


Figura 6: Diagrama de transiciones vítreas, adaptado de Altieri, A. [14]

Cuando empezamos a estudiar sistemas activos y sus transiciones de fase, es común pensar primero en partículas brownianas activas. Para partículas brownianas pasivas con repulsión eléctrica sin alineación, el proceso conduce a estados estacionarios homogéneos cuya estructura está controlada por la densidad de las partículas. Por el contrario, si ahora tenemos partículas brownianas autopropulsadas, la formación de cúmulos conducirá a separaciones de fases completas en estados diluidos y densos, que recuerdan a lo que sucede en la transición líquido-gas. Este fenómeno se denomina separación de fases inducida por la motilidad (MIPS en inglés). Cuando mezclamos este comportamiento activo con una fracción de empaquetamiento más alta, las partículas brownianas activas pueden formar estados sólidos, que exhiben un orden cristalino cuando las partículas están monodispersas, pero son vítreas en sistemas polidispersos [13].

El estudio de las interferencias activas y de los vidrios activos formados por densas colecciones de partículas brownianas activas fue estimulado en gran medida por la observación experimental de la dinámica vítrea en monocapas de células epiteliales; como veremos con más detalle en las próximas secciones, los epitelios son en realidad tejidos confluentes, es decir, son láminas de células densamente empaquetadas que cubren el plano, sin espacios entre ellas y que permanecen confluentes mientras cambian de un comportamiento líquido a sólido. En otras palabras, la fracción de empaquetamiento es siempre igual a la unidad, lo que sugiere que, a diferencia de los sistemas de partículas, la densidad puede no ser un parámetro relevante para ajustar la

aparición de rigidez en los epitelios. A diferencia de los sistemas inertes, las transiciones de fase en estos sistemas vivos pueden desencadenarse por cambios en la dinámica y las interacciones de sus constituyentes. La motilidad celular y los reordenamientos celulares son fundamentales en estas transiciones[10].

Uno de los ejemplos de la importancia de la motilidad celular y su auto-organización en las transiciones de fase más sorprendentes es como la morfología de los embriones tempranos tiene fenómenos como la extensión de la forma del cuerpo a lo largo del *eje anteroposterior* [17]. El análisis del movimiento celular en estas fases reveló que las células mesenquimatosas (un tipo de célula altamente móvil y migratoria presente en los embriones) exhiben un movimiento browniano en relación con la matriz extracelular subyacente (el sustrato, por donde se desplazan), lo que sugiere un comportamiento similar al de un gas. Por otro lado, los reordenamientos celulares provocan cambios de forma y flujos a escala tisular, donde la fluidez activa surge de cambios en la motilidad celular pero también como consecuencia de la división celular. La división celular promueve activamente los reordenamientos celulares: las células hijas se separan entre sí después de dividirse y remodelan sus uniones con las células vecinas, dando origen a la complejidad estructural que encontramos en los orígenes de la vida.

Además, la densidad celular o, equivalentemente, la fracción de volumen es un parámetro físico relevante para la transición de fase, ya que determina en gran medida la frecuencia de las interacciones entre las células. Como en sistemas tan saturados se observan estados tanto «atascados» como «no atascados», es razonable cuestionar el uso de la densidad como parámetro de control. Esto se debe a que el parámetro de control de la transición de interferencia/liberación es la relación entre el perímetro observado y la raíz cuadrada del área observada y se le denomina *índice de forma* [10]. Hasta cierto punto, la existencia del índice de forma como sello distintivo del estado sólido recuerda a un parámetro de orden estructural bien definido para los cristales. Esto sugiere que, aunque el material permanece desordenado, es posible reconocer si se comporta como un fluido o como un sólido observando la forma de sus constituyentes.

De la materia inerte conocemos que la temperatura es un parámetro físico importante y relevante para las transiciones de fase, ya que es una medida de la energía cinética promedio de

las moléculas constituyentes. En consecuencia, las moléculas de los sólidos se mueven menos en comparación con la fase líquida o gaseosa del mismo material, lo que implica efectivamente que el punto de congelación está a una temperatura más baja que el punto de ebullición. Por lo tanto, la temperatura está bien establecida como parámetro de control para las transiciones sólido-fluido.

Sólo recientemente se ha propuesto que un parámetro equivalente, denominado «cell jiggling» [10], se interprete como la temperatura efectiva. De manera similar a la imagen cinemática de la temperatura, el movimiento celular es una medida de las fluctuaciones en los contactos entre las células y es el resultado de diversos procesos moleculares que pueden modular la adhesión célula-célula o la contractibilidad celular.

También es importante tener en cuenta la conectividad célula-célula. En física sabemos que las alteraciones en la conectividad de los enlaces de una red pueden provocar cambios macroscópicos en la rigidez. El grado de conectividad entre células puede verse en analogía con la filtración a través de una sustancia porosa (fenómeno conocido como percolación).

En general, los sistemas biológicos exhiben un fuerte acoplamiento entre las escalas temporales de los cambios en las propiedades mecánicas y las deformaciones, como resultado de lo cual las tensiones pueden desempeñar un papel muy diferente en las transiciones de fase, en particular en la fase de 'átasco', cuando en comparación con materiales inertes. Está en constante juego una retroalimentación entre las señales mecánicas y bioquímicas desde las escalas moleculares hasta las tisulares. Además, es importante considerar también como el tamaño y la geometría pueden desempeñar un papel fundamental en los comportamientos colectivos de los sistemas multicelulares, creando una imagen desalentadora para poder hacer analogías desde la física inerte, dada la complejidad de los sistemas.

2.4. Tejidos biológicos como cristales líquidos nemáticos activos

Como revisamos a finales de la subsección anterior, los tejidos vivos pueden auto-organizarse y describirse como materiales activos compuestos de células que generan tensiones activas mediante la inyección continua de energía. Trabajos muy recientes han identificado sistemas biológicos donde se crean continuamente defectos topológicos como en los nemáticos activos [18]. Sorpren-

dentamente, también ha habido sugerencias de que las características nemáticas activas pueden estar relacionadas con la funcionalidad biológica. Por ejemplo, las redes moleculares reconstituidas *in vitro*, así como los citoesqueletos unicelulares, muestran que sus estructuras filamentosas pueden presentar propiedades cristalinas líquidas nemáticas y pueden promover procesos fuera del equilibrio inducidos por procesos activos en la microescala [19]. La aparición de patrones colectivos, la formación de singularidades topológicas y la transición de fase espontánea dentro del citoesqueleto celular son propiedades emergentes que impulsan las funciones celulares. Los sistemas más integrados, como los tejidos, tienen células que pueden verse como partículas nemáticas activas usando el planteamiento de grano grueso (coarse-graining method) [19] y su interacción puede dictar muchos procesos tisulares importantes, como la extrusión y migración celular, que se ha observado tanto *in vitro* como *in vivo*.

Las células epiteliales están estrechamente conectadas mediante uniones entre ellas y la actividad inherente de cada una impulsa constantemente la deformación de su estructura. Al trazar la dirección (de grano grueso) del eje largo de las células deformadas, Saw *et al.* [20] identificó un orden nemático y defectos topológicos móviles dentro de una capa confluyente bidimensional de células epiteliales. Utilizando teorías nemáticas activas continuas, pudieron predecir fielmente los campos de flujo y las tensiones alrededor de los defectos y, de acuerdo con la teoría, los experimentos muestran que existe una correlación entre el nivel de actividad y el número de defectos en la capa celular. En particular, los defectos topológicos con carga $+1/2$ tienen una fuerte relación con los sitios de apoptosis, donde una célula muere y luego es expulsada de la monocapa de tejido. Esto es resultado de la tensión de compresión relativamente alta que se genera en la "cabeza" de los defectos $+1/2$. El estrés compresivo desencadena una respuesta mecanotransductora dentro de la célula (movimiento de una proteína denominada YAP desde el núcleo al citoplasma), que desencadena la muerte celular y una expulsión simultánea de las células en los sitios de defecto. Los autores demostraron que la impresión por microcontacto en un sustrato donde se encuentren las células podría usarse para controlar el número de defectos topológicos y, por tanto, la tasa de apoptosis, en diferentes áreas de la capa celular confluyente.

Por otro lado, la morfogénesis animal implica múltiples procesos mecánicos y bioquímicos, que abarcan varios órdenes de magnitud en el espacio y el tiempo, desde la dinámica local a nivel

molecular hasta la morfología global a escala de organismo. Cómo se coordinan e integran estos numerosos procesos para formar resultados robustos y funcionales sigue siendo una pregunta pendiente. En Maroudas-Sacks *et al.* [21] desarrollaron una descripción eficaz y detallada de la morfogénesis de la regeneración de todo el cuerpo de la *Hidra*, un pequeño animal de agua dulce, que es un clásico sistema modelo para la morfogénesis, debido a su plan corporal simple y sus notables capacidades de regeneración. En [21] la hidra se describe como un sistema nemático activo, el cual contiene dos conjuntos perpendiculares distintos de fibras de actina supracelulares con un orden nemático en el ectodermo y en el endodermo (capas germinales de células), los autores siguieron *in vivo* la dinámica nemática de estas fibras de actina durante la regeneración a partir de trozos de tejido extirpados. Durante este proceso, las regiones con fibras alineadas inducen orden en las regiones vecinas, sin embargo, comienzan a aparecer defectos topológicos debido a la restricción geométrica de las zonas extirpadas de forma muy temprana, que permiten identificar y diferenciar entre los sitios distintos de formación de «la cabeza» y «el pie» de la hidra, mucho antes de la aparición de cualquier característica morfológica. Esta capacidad de predecir los sitios de formación de la cabeza y los pies a partir del campo nemático sugiere fuertemente que la distribución y la dinámica del defecto están acopladas a los mecanismos que estabilizan la alineación y la polaridad del eje del cuerpo y, por lo tanto, probablemente desempeñan un papel en el establecimiento del plan corporal. Esta relación observada entre la dinámica de los defectos y el proceso de regeneración ofrece un punto de vista biofísico complementario sobre la morfogénesis, con el campo de orientación nemática sirviendo como un morfógeno mecánico y los sitios de defectos en este campo actuando como centros de organización de la Hidra en desarrollo.

Además, como mencionamos, las transiciones de fase son un fenómeno relevante y evidente en los tejidos biológicos. Chevalier *et al.* [23] demostró que el tránsito de las células de la cresta neural entérica durante el desarrollo fetal de los mamíferos tenía una transición de fase isotrópica a nemática, donde en las primeras etapas del desarrollo embrionario de los ratones descubrieron cómo las células se distribuían de forma homogénea e isotrópica, pero en los días siguientes las células se orientaban progresivamente a lo largo de la circunferencia del tracto gastrointestinal, formando rayas o 'costillas'. Entonces, estas células pasaron de un estado de baja coherencia, con una ligera orientación longitudinal, a un estado de alta coherencia, con orientación circunferencial, haciendo una analogía con la física de los cristales líquidos que describimos en secciones anteriores.

2.5. Embriogénesis temprana del pez cebra

Nuestro sistema biológico de enfoque es el *pez cebra*, que ofrece un sistema vertebrado único y lleno de ventajas. El pez cebra es un gran modelo animal porque tiene grandes desoves, se reproduce durante todo el año y es de fácil mantenimiento. Además, una de las características más relevantes para los embriólogos y biólogos del desarrollo es el hecho de que tienen embriones transparentes que se desarrollan fuera de la madre, lo que nos permite utilizar diferentes métodos de microscopía para estudiar la dinámica celular de todo el embrión *in vivo*. Además, se pueden crear varios mutantes en esta especie en particular, mediante el uso de diferentes mutágenos que cambian algunas vías de señalización bioquímica, lo que lleva a diferentes fenotipos para los estudios.

En particular, estamos interesados en estudiar un proceso llamado *Gastrulación*, un estadio morfogénético a gran escala que define las tres principales capas germinales, el mesodermo, que formará los músculos, los elementos cardiovasculares, urogenitales y esqueléticos del cuerpo, el ectodermo, que dará origen a la epidermis y tejidos neurales, y al endodermo, que generará el tubo digestivo y sus órganos accesorios, conllevando un conjunto de movimientos morfogénéticos conservados evolutivamente; invaginación o embolia, epibolia, convergencia y extensión, definidas por su resultado morfogénético [27].

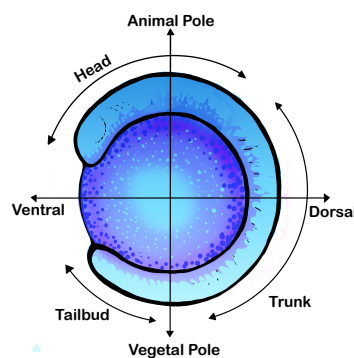


Figura 7: Diagrama del eje del pez cebra, adaptado desde Marelli *et al.*, [24]

2.5.1. Movimientos morfogenéticos durante la embriogénesis del pez cebra.

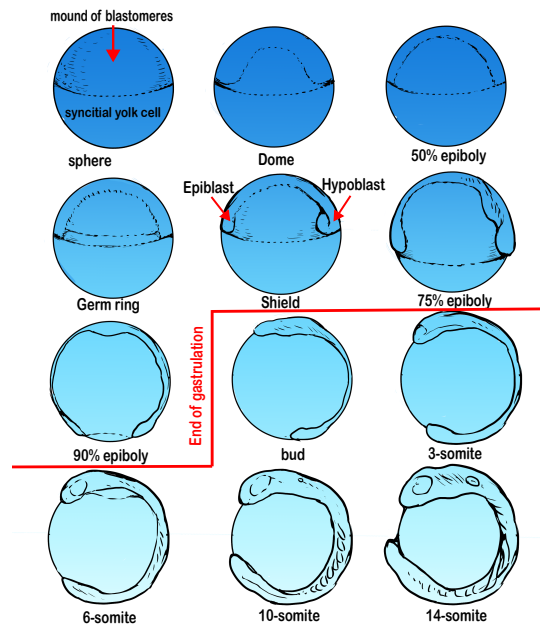


Figura 8: Movimientos morfogenéticos en el desarrollo embrionario del pez cebra, adaptado de Webb [25]

Después de 5 horas de la fertilización, al iniciar los movimientos de gastrulación, el embrión del pez cebra exhibe una arquitectura simple, con un montículo de blastómeros (un tipo de célula producida por la división celular del cigoto después de la fertilización), conocido como blastodermo, residiendo en la parte superior de la célula vitelina sincicial [27]. El blastodermo consta de una capa envolvente superficial (células epiteliales aplanadas) y células profundas (células blastodérmicas más vagamente asociadas), que darán lugar a todos los tejidos embrionarios.

El primer movimiento morfogenético durante la embriogénesis del pez cebra es *la Epibola* [27], un proceso morfogenético que resulta en la diseminación isotrópica de los tejidos, generalmente asociada con su adelgazamiento, en el pez cebra comienza cuando la célula vitelina plana forma una cúpula hacia el blastodermo y los blastómeros más profundamente ubicados se intercalan radialmente con capas más superficiales. Simultáneamente, el blastodermo se adelgaza y se expande hacia el polo vegetal. Cuando el blastodermo cubre la mitad de la célula vitelina, la blástula del pez cebra exhibe una distribución de precursores de la capa germinal (como un mapa de destino), donde las células se acumulan en la zona marginal y fluyen hacia la célula vitelina, creando un

pliegue engrosado referido como el *anillo germinal* [27].

Ahora, durante la internalización (donde se formarán las capas germinales, mesodermo y endodermo), las células se internalizan dorsalmente y luego en todos los puntos alrededor del margen [27]. Las posibles células endodérmicas residen más cerca del margen del blastodermo y están entremezcladas con precursores mesodérmicos ubicados más lejos del blastoporo. La región animal del blastodermo contiene precursores ectodérmicos. Ahora, durante la embolia [27], los precursores mesendodérmicos se mueven a través del blastoporo debajo del ectodermo potencial. En el margen dorsal del blastodermo, la internalización implica el ingreso de blastómeros individuales, mientras que, en las regiones lateroventrales, los precursores del mesendodermo internalizan de manera sincrónica que recuerda a la involución, en el proceso de ingreso sincronizado.

Una vez internalizadas, las células mesendodérmicas migran hacia el polo animal, formando una capa interna denominada *Hipoblasto* [27], mientras que las células superpuestas que no involucionan constituyen una capa ectodérmica llamada *textitepiblasto* [27] que continúa *épibolizándose*. La separación de las capas de epiblasto e hipoblasto se mantiene durante toda la gastrulación. Mientras tanto, los precursores endodérmicos también se propagan hacia el polo animal mediante un movimiento aleatorio.

Los movimientos de convergencia y extensión son altamente dinámicos y varían de manera espaciotemporal, comienzan simultáneamente con los de internalización en la mitad del proceso epibólico. La convergencia es el movimiento de las células hipoblásticas y epiblásticas hacia el futuro lado dorsal del embrión [27], lo que resulta en un estrechamiento mediolateral del eje. La extensión se refiere al alargamiento del eje anteroposterior visto en el lado dorsal. En las regiones ventrales, las células mesodérmicas no realizan estos movimientos, sino que migran hacia el polo vegetal.

El inicio del movimiento de convergencia y extensión está marcado por la compactación de células en el lado dorsal del embrión que dan lugar al *escudo*, el organizador embrionario del pez cebra [27]. Las células mesodérmicas ubicadas en el escudo forman estructuras axiales como la placa precordial y la notocorda, mientras que el mesodermo paraxial y lateral dan lugar a estructuras como los somitas y la placa lateral, respectivamente.

A medida que avanza la gastrulación, los movimientos de convergencia y extensión se restringen de modo que las células de estos diferentes subtipos mesodérmicos no se mezclan. Además, se pueden observar diferentes grados de convergencia y comportamientos de movimiento de extensión para células en diferentes posiciones a lo largo del eje dorsal-ventral [27]. En general, estos comportamientos son más pronunciados en las células del mesodermo que se encuentran en posiciones más dorsales. El mesodermo más ventral, a diferencia del mesodermo dorsal y lateral, no se mueve dorsalmente ni se extiende hacia el polo animal, sino que se mueve hacia el polo vegetal donde contribuye al brote de la cola [27]. Las células mesodermales axiales experimentan una rápida extensión debido en parte a comportamientos de elongación e intercalación mediolateral (MIB) y a la migración celular dirigida [27].

Por lo tanto, la gastrulación del pez cebra implica todos los movimientos de gastrulación conservados, que son impulsados por una variedad de comportamientos celulares, incluida la migración celular, el ingreso, las intercalaciones radiales y planas y el cambio de forma celular [29].

2.5.2. Pez cebra como un material activo

En 2017, Morita *et al.* [34] propuso un modelo mecánico del proceso de "domo" que incorpora los efectos tanto de la expansión de las células superficiales como de las intercalaciones con las células más profundas del tejido, usando una descripción de fluido activo para la deformación del embrión. La generación de fuerza en las células superficiales y profundas del embrión se caracteriza a una escala mucho mayor que la de una célula individual, es decir, abarcando el tejido a gran escala, mediante unos pocos parámetros fenomenológicos correspondientes a las viscosidades de los tejidos y la generación de fuerza interna. El estudio se centró específicamente en la interacción mecánica entre los dos tipos de tejido predominantes, una capa superficial de células epiteliales y células mesenquimatosas profundas, altamente móviles, situadas debajo de la capa epitelial, donde ambos experimentan cambios en su estructura pronunciados durante la formación del domo. A pesar de su simplicidad, su descripción física les permitió capturar los cambios de forma característicos de esos tejidos durante este proceso. Además, pudieron medir otros parámetros como la tensión superficial del embrión y la viscosidad del blastodermo mediante experimentos de compresión y explantes, ajustando parámetros internos para simular la estructura *in vivo*. Su

hallazgo proporciona evidencia de que para la propagación y movimiento de estos tejidos multicapa complejos, las células de la superficie desempeñan un papel clave al reducir su tensión superficial y, por lo tanto, expandirse activamente, y no solo impulsan la expansión del tejido, sino que también desencadenan la intercalación radial activa de las células profundas subyacentes, un proceso necesario para el adelgazamiento homogéneo del tejido durante la extensión.

Por lo tanto, al reducir simultáneamente la tensión superficial del tejido y aumentar la contracción radial del tejido, la expansión celular activa de la superficie permite que los tejidos complejos experimenten una expansión y adelgazamiento coordinados del tejido. Contándonos la importancia de la morfología y dinámica celular.

Más recientemente, en 2021, Romeo *et al.*[35] utiliza datos de seguimiento unicelular a nivel completo del embrión durante la gastrulación temprana en el pez cebra, construyendo una representación simplificada de los campos hidrodinámicos resultantes en términos de las funciones base de los armónicos esféricos escalares y vectoriales y los conecta a la dinámica de los defectos topológicos que se forman en el flujo celular. Validan este modelo utilizando datos sintéticos de partículas brownianas activas confinadas en una capa esférica. Demostrando que, de hecho, los tejidos embrionarios del pez cebra se comportan como un material activo.

Este mismo año (2021), Petridou *et al.* [37] realizó un estudio en el blastodermo del pez cebra que demostró una transición de fase de rigidez genuina que resulta de cambios en la conectividad celular mediada por la adhesión entre sus componentes. Los autores demostraron que cuando el promedio de contactos por célula es bajo, el tejido, como era de esperar, se comporta como un fluido, pero por encima de un umbral crítico de conectividad, emergen abruptamente grupos rígidos gigantes que abarcan toda la red celular, resistiéndose a la deformación, lo que hace que el comportamiento del tejido sea más sólido. Mediante una combinación de perturbación genética para alterar la conectividad celular, se demostró que la conectividad celular es un parámetro confiable para predecir cambios en la viscosidad del tejido mediante análisis de percolación de rigidez. Estas transiciones de fase de red son otro ejemplo de cómo los sistemas biológicos utilizan las transiciones de fase para esculpir tejidos durante la morfogénesis.

Aquí, conociendo la capacidad de los tejidos y las células para moverse y reorganizarse en el

proceso de la embriogénesis del pez cebra, queremos centrarnos en este complejo problema en particular, porque encapsula todos los fenómenos que revisamos en la parte teórica física de esta sección, en su organización tisular. En particular nos es importante destacar la existencia de al menos 3 diferentes comportamientos, las formación de células líderes que guían el movimiento en la extensión de los tejidos, la importancia en la adhesión e interacción entre células y como consecuencia, la existencia de diferentes fases de rigidez en el tejido, teniendo esta especie de transición de atasco, a un movimiento mas organizado y alineado, y a la polarización y creación de células líderes con un comportamiento mas fluido y dinámico.

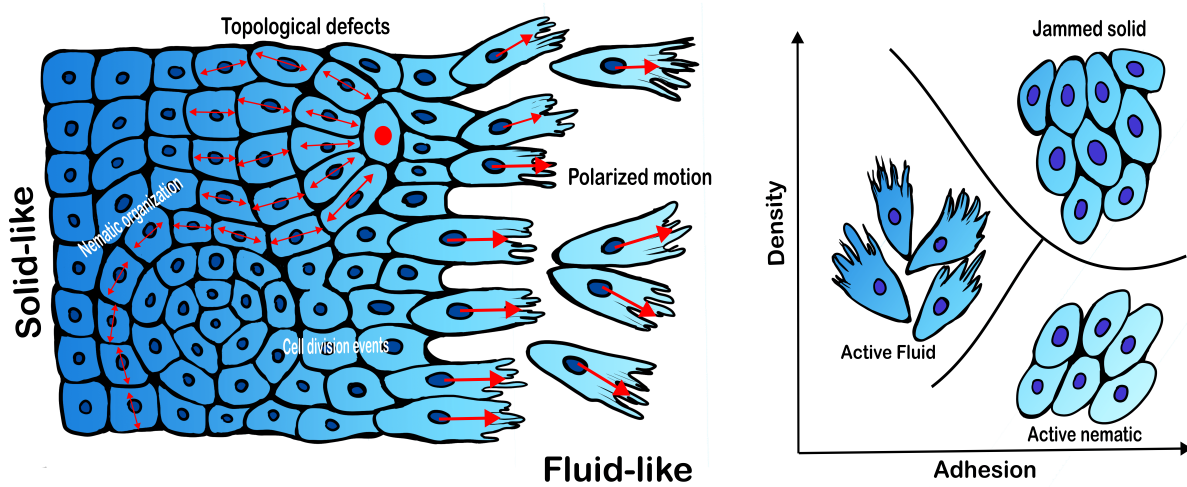


Figura 9: Tejido complejo que encapsula defectos topológicos, movimiento polarizado, organización nemática y diferentes fases de la materia, adaptado desde Balasubramaniam *et al.* [26]

3. Capítulo 2: Caracterización de la forma celular en tres dimensiones

El comportamiento colectivo del desarrollo embrionario del pez cebra muestra características notablemente similares a las de los materiales activos. Por lo tanto proponemos utilizar el marco de la materia activa para analizarlo formalmente. Sin embargo, para capturar las propiedades colectivas de los tejidos biológicos en una descripción detallada y oportuna, es necesario tener en cuenta todos los ingredientes relevantes de la dinámica celular, como la forma de las células; para migrar, las células deben adquirir una asimetría espacial que les permita generar fuerzas

para aportar en la translocación neta del cuerpo celular, donde se forman células que lideran el movimiento y guían el resto del tejido, de aquí también es importante considerar las interacciones célula-célula que también son necesarias para la adhesión celular dentro del tejido y para controlar la forma y función de las células, desempeñando un papel crucial en el desarrollo del pez cebra.

3.1. Reconstrucción en mallas triangulares de imágenes de tres dimensiones

El primer objetivo de este capítulo es transformar las coordenadas espaciales de las células en un formato adecuado para análisis posteriores, utilizando técnicas de eliminación de ruido, centrado y alineación de datos, así como segmentación de imágenes tridimensionales. Esta transformación permitirá crear representaciones visuales semi-artificiales de las trayectorias y estructuras celulares, facilitando el estudio de patrones y ensamblajes emergentes en los datos.

Pre-procesamiento de Datos El primer paso en el pre-procesamiento de los datos fue aplicar una función de eliminación de ruido mediante wavelets en Python a cada trayectoria celular. Este proceso ayuda a suavizar las trayectorias, eliminando fluctuaciones no deseadas y resaltando las tendencias subyacentes. Posteriormente, se centró la nube de células ajustando una superficie esférica mediante mínimos cuadrados y desplazando el origen en cada instante de tiempo para coincidir con el centro de la esfera. Finalmente, se rotaron los datos para alinear el polo animal del embrión con el eje z, determinado por la dirección del centro de masa de la distribución celular inicial.

Procesamiento de coordenadas y creación de imágenes El conjunto de datos, originalmente en formato CSV, contiene las coordenadas x,y,z de cada célula, junto con su identificador único y el tiempo correspondiente. Estos datos fueron procesados para generar volúmenes tridimensionales de las trayectorias celulares, siguiendo el procedimiento descrito a continuación:

- **Inicialización y lectura de datos:** Se establecieron variables iniciales y se leyó el archivo CSV en «chunks» para manejar grandes volúmenes de datos de manera eficiente. Cada chunk fue procesado para almacenar las posiciones de las células en un diccionario estructurado por tiempo e identificador celular.

- **Ajuste de Coordenadas:** Las coordenadas fueron ajustadas para asegurar que todas las posiciones fueran no negativas y estuvieran dentro de los límites de un volumen tridimensional predefinido.
- **Generación de Volúmenes 3D:** Para cada instante de tiempo, se creó un volumen tridimensional donde cada célula se representó en su posición. Estos volúmenes se guardaron en archivos .npy para su uso posterior.

Segmentación de Volúmenes Para analizar y visualizar las estructuras celulares, se realizó la segmentación de los volúmenes generados. El proceso de segmentación incluyó los siguientes pasos:

- **Carga de Volúmenes:** Los volúmenes tridimensionales fueron cargados desde los archivos .npy generados previamente.
- **Dilatación binaria y Segmentación Watershed:** Se aplicó una dilatación binaria a los volúmenes para ampliar las regiones ocupadas por células, seguida de un filtrado mediano para suavizar la máscara dilatada. Luego, se utilizó la transformación de distancia y un filtro gaussiano para preparar el volumen para el algoritmo de watershed (Este algoritmo toma los marcadores individuales e «Inunda» hasta llegar a los bordes. En muchos casos, se eligen marcadores como mínimos locales de la imagen, a partir de los cuales se inunda hasta el borde previamente definido.). El resultado de la segmentación se guardó en archivos .tif para su análisis visual y cuantitativo.

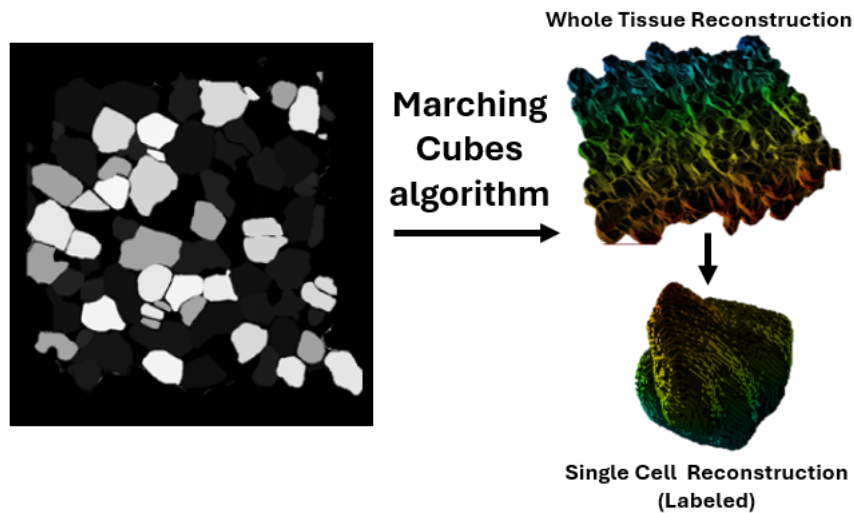


Figura 10: Reconstrucción desde imagen Watershed

Ahora habiendo obtenido la imagen con la segmentación de las células, podemos reconstruir de forma tridimensional la estructura del tejido a nivel de cada célula, para ello utilizamos el algoritmo *marching cubes* del paquete *measure* de *scikit-image*. *Marching cubes* es un algoritmo para extraer una malla de superficie 2D de un volumen 3D. Esto puede conceptualizarse como una generalización 3D de isolíneas en mapas topográficos. Funciona iterando a lo largo del volumen, buscando regiones que crucen el nivel de interés. Si se encuentran dichas regiones, se generan triangulaciones y se agregan a una malla de salida. El resultado final es un conjunto de vértices y un conjunto de caras triangulares.

Para poder utilizarlo en nuestro estudio, definimos una función que genera una malla tridimensional de la célula basada en los siguientes parámetros:

- `stepsize (float)`: Tamaño de pasos para el algoritmo de *marching cubes*.
- `smooth image wnd (int)`: Tamaño para suavizado de mediana para la máscara celular.
- `smooth mesh itt (int)`: Número de iteraciones para el suavizado de la malla tridimensional.

En estos parámetros hay varios valores que son necesarios para un pre-procesamiento de los datos, y así garantizar una mejor reconstrucción de la estructura. Consideramos la etiqueta 0 como el fondo de la imagen, y desde ahí creamos una máscara de los datos donde rellenamos agujeros (que pudiesen haber escapado del pre-procesamiento para la segmentación) y aplicamos un filtro

de mediana sobre la imagen, cuyo valor dependerá del input del usuario, utilizamos 1 y 2 en este trabajo.

```
if self.cell_id == 0:
    self.mesh = None

# Create the cell mask
cell_mask = (self.image == self.cell_id).astype(np.int8)

# Fill holes in the 3D image
cell_mask = ndi.binary_fill_holes(cell_mask)

# Median smooth image
if processing_parameters.smooth_image_wnd > 0:
    scipy.ndimage.median_filter(cell_mask, size=
        processing_parameters.smooth_image_wnd, output=cell_mask)
```

Ahora utilizando «*measure.marching_cubes*», obtenemos la reconstrucción tridimensional como vemos en 10, llamamos una función auxiliar que pueden encontrar en (citar apéndice), donde verificamos si la nueva malla tridimensional es un volumen y si no lo es, aplicamos varias correcciones que contemplan arreglar las normales, cerrar agujeros, entre otros.

Finalmente, también podemos aplicar un filtro de suavizado para una reconstrucción mas suave de la superficie y con una mejor distribución de la malla triangular.

```
# Generate mesh
verts, faces, _, _ = measure.marching_cubes(cell_mask, allow_degenerate=False,
    step_size=processing_parameters.stepsize)

del cell_mask
self.mesh = trimesh.Trimesh(vertices=verts, faces=faces)

if self.mesh.is_volume == False:
    self.fix_mesh_is_volume_False()

# Smooth mesh
if processing_parameters.smooth_mesh_itt > 0:
    self.mesh = trimesh.smoothing.filter_humphrey(self.mesh, alpha=0.1,
        beta=0.1, iterations=processing_parameters.smooth_mesh_itt,
```

3.2. Desarrollo de herramientas computacionales para la cuantificación de las formas celulares en tres dimensiones

3.2.1. Cuantificación contacto célula-célula

El contacto entre células desempeña un papel fundamental en la dinámica y la estructura celular. Estos contactos no solo son cruciales para comprender el comportamiento celular, sino que también son indicativos de la posición de las células dentro del tejido. Aquellas con múltiples contactos suelen encontrarse en el núcleo del tejido o en la región densamente poblada. En contraste, las células en los bordes del tejido exhiben un borde libre con contactos limitados en su superficie, lo que las caracteriza como células líderes.

Por otro lado, las células intermedias, alineadas y con contactos menos frecuentes, pueden mostrar hipotéticamente un comportamiento nemático. Esta fase intermedia se sitúa entre un estado sólido o "atacado" un comportamiento de fluido activo, típico de las células líderes. Por lo tanto, es de gran interés cuantificar de manera precisa la distribución espacial de estos contactos en la membrana celular.

Para abordar este análisis, presentaremos un método sistemático para caracterizar estas distribuciones de contactos celulares en una superficie compleja. Este método implica proyectar las distribuciones en una esfera unitaria y aproximarlas mediante una expansión de modos esféricos. Luego, el espectro de potencias resultante se utilizará para determinar si el comportamiento es vectorial o nemático, y para explorar la posible existencia de otros patrones de comportamiento.

Comenzaremos describiendo cómo se caracteriza una distribución en la superficie de una esfera, para luego extender este enfoque a geometrías simples con diferentes niveles de contacto.

3.2.1.1 Descomposición en modos esféricos

Basado en el artículo [4]. Sea $f(\theta, \phi)$ una densidad de superficie definida sobre una esfera unitaria, con θ el ángulo polar y ϕ el ángulo azimutal, esta puede ser descompuesta en la suma de sus modos armónicos esféricos como:

$$f(\theta, \phi) = \sum_{l=0}^{\infty} F_l(\theta, \phi)$$

con $F_l(\theta, \phi) = \sum_{m=-l}^l f_l^m Y_l^m(\theta, \phi)$ y los coeficientes f como:

$$f_l^m = \int_{S^2} d\Omega f(\theta, \phi) Y_l^{m*}(\theta, \phi)$$

Donde $d\Omega = \sin\theta d\theta d\phi$ diferencial de ángulo solido. Podemos obtener la descomposición en modos esféricos de $f(\theta, \phi)$:

$$\|F_l\|^2 = \frac{1}{4\pi} \int_{S^2} d\Omega |F_l|^2 = \frac{1}{4\pi} \sum_{m=-l}^l |f_l^m|^2, l = 0, 1, 2, \dots$$

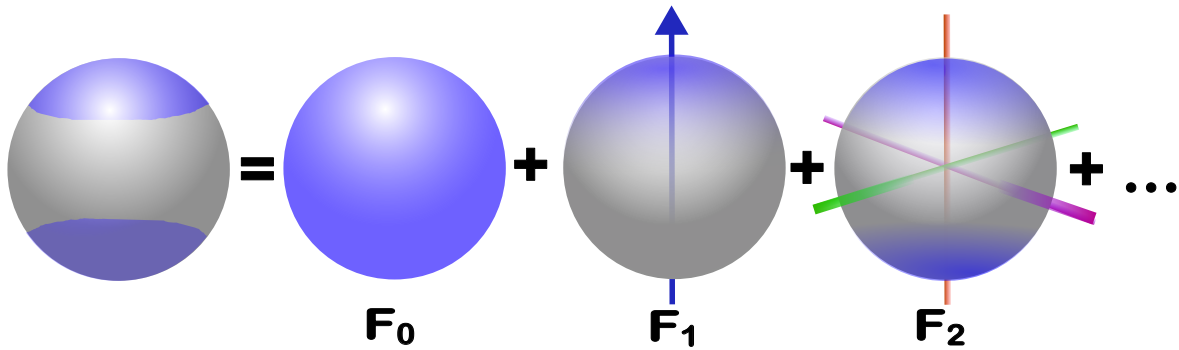


Figura 11: Descomposición multipolar de patrón superficial, la distribución sobre la esfera es reconstruida en la suma de los modos esféricos, adaptado de Scholich *et al.*

En la figura 11 podemos ver el ejemplo de una distribución sobre una superficie esférica descompuesta en sus modos esféricos. Para cuando $l = 0$, tenemos el modo F_0 que es isotrópico y cubre todo el ángulo solido al realizar la integración sobre la esfera unitaria. Cuando $l = 1$, tenemos el modo F_1 que puede ser representado con un vector en la dirección del promedio esférico de la distribución superficial [3]. Cuando $l = 2$ tenemos el modo F_2 que puede ser representada por un tensor cuyos autovectores representarían los tres principales ejes de la distribución.

El código toma la reconstrucción en mallas triangulares de la estructura a analizar (malla generalmente creada utilizando el algoritmo de *marching cubes* de la sección anterior), donde cada triangulo tendrá asignado un valor 0 o 1 dependiendo de si tiene contactos con una estructura vecina o no, siendo esta nuestra función de distribución $f(\theta, \phi)$ que se proyectará en una esfera unitaria, como vemos en la figura 12 Utilizamos el método de Simpson, que fue probado el método mas eficiente en [43] para calcular los coeficientes f_l^m , de esta forma podemos obtener los modos representativos de cada estructura $\|F_l\|^2 = \frac{1}{4\pi} \sum |f_l^m|^2$.

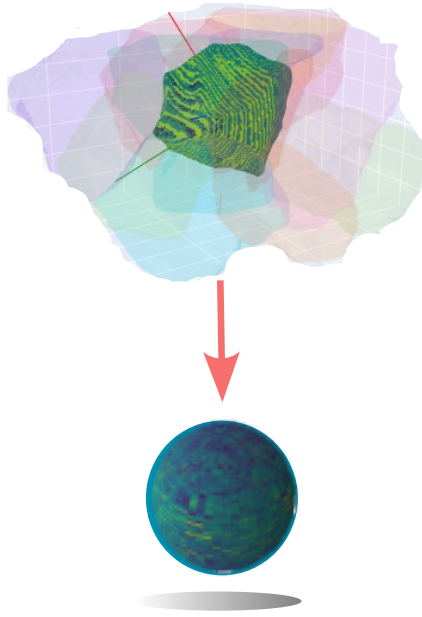


Figura 12: Esquema de célula y sus células vecinas con opacidad mas baja, sobre ella una distribución de colores representando la distribución de contactos con sus vecinos que se proyecta sobre una esfera unitaria

3.2.1.2 Validación del algoritmo

Para garantizar que el algoritmo desarrollado es funcional y produce resultados confiables, se probaron las funciones con diferentes distribuciones simples cuyos resultados son conocidos.

Distribuciones simples sobre una esfera Se utilizó una malla homogénea sobre una esfera (para evitar distorsiones en los datos) de $\cos(\theta)$ vs ϕ . Se probaron las siguientes distribuciones:

Distribución Homogénea: Una distribución uniforme sobre la esfera, que corresponde al modo F_0 . Teóricamente coincide con nuestros resultados y se utilizó para normalizar las demás distribuciones.

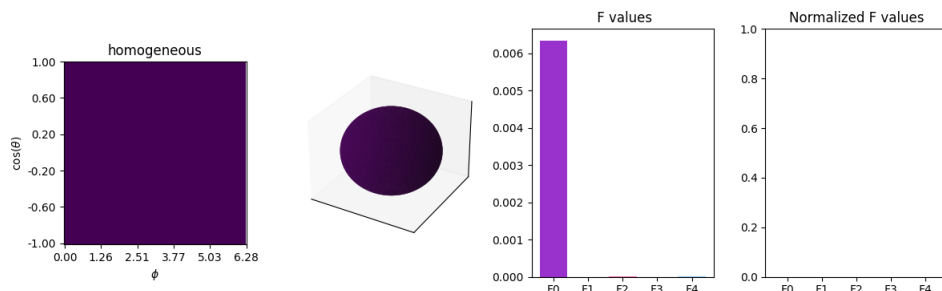


Figura 13: Distribución homogénea sobre una esfera.

Distribuciones polares: Desarrollamos distribuciones tipo polo norte y polo sur para comprobar que obteníamos el modo F_1 de una distribución vectorial o polar, lo que también coincidió con lo esperado.

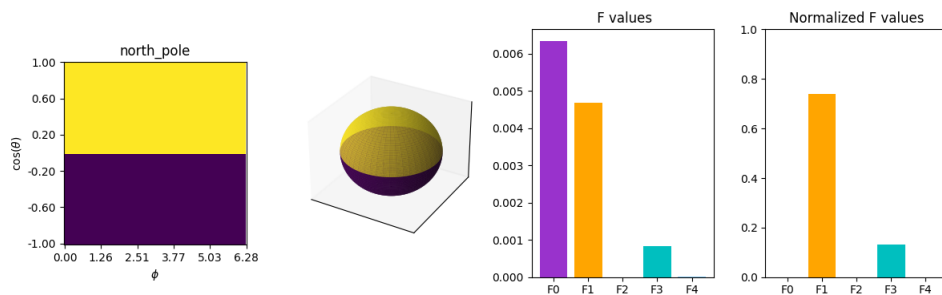


Figura 14: Distribución tipo polo norte sobre una esfera.

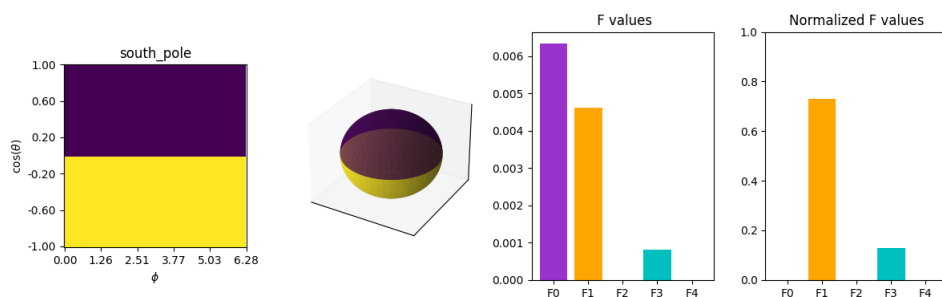


Figura 15: Distribución tipo polo sur sobre una esfera.

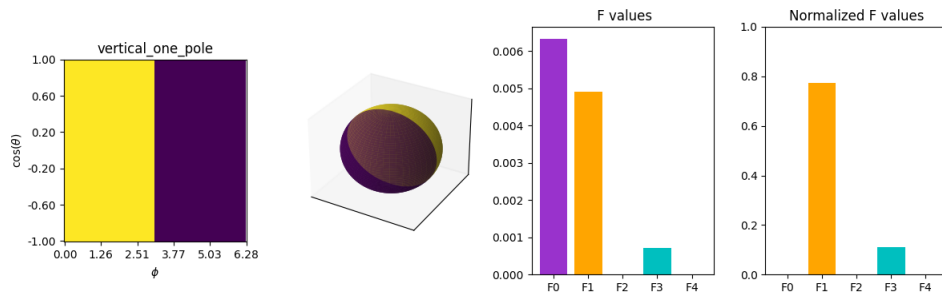


Figura 16: Distribución uni-polar vertical sobre una esfera.

Distribuciones bipolares: Se evaluaron distribuciones bipolares y de anillo sobre la esfera, donde se esperaba una contribución máxima del modo F_2 tras la normalización, caracterizado tensorialmente. Nuevamente los resultados coincidieron con lo esperado.

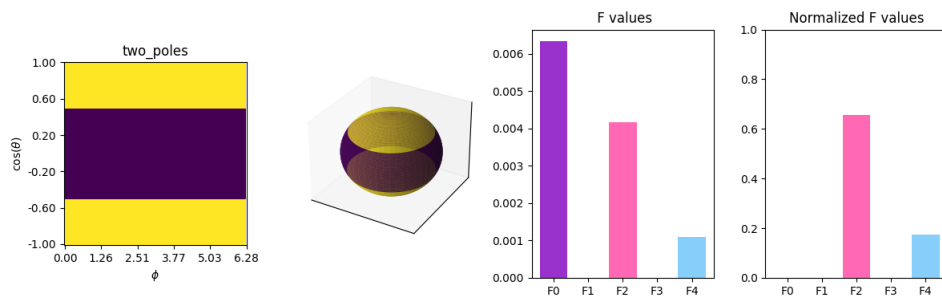


Figura 17: Distribución bipolar sobre una esfera.

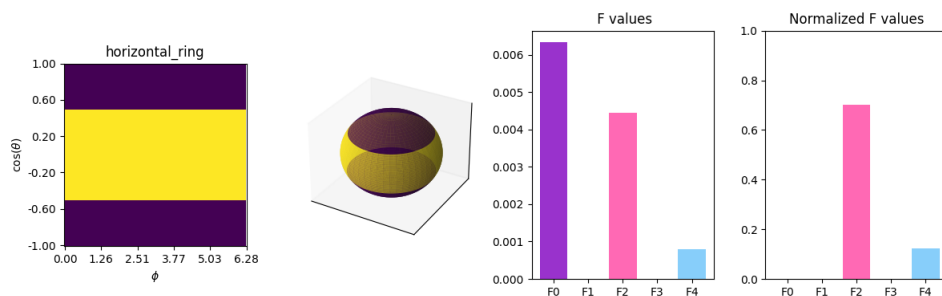


Figura 18: Distribución tipo anillo sobre una esfera.

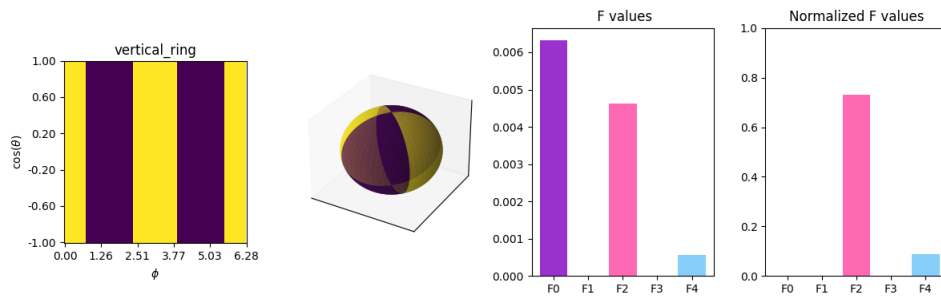


Figura 19: Distribución tipo anillo vertical sobre una esfera.

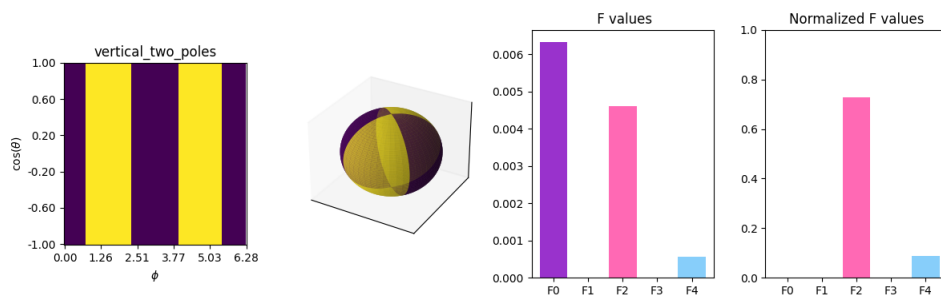


Figura 20: Distribución tipo dos polos verticales sobre una esfera.

Evaluación en geometrías simples Antes de aplicar el algoritmo a datos reales, se evaluaron diferentes geometrías para observar cómo su morfología se traducía en modos por donde había mas incidencia de los vectores normales sobre su superficie. Se probaron elipsoides y cuboides con diferentes combinaciones de elongaciones, con una distribución homogénea de contactos en su superficie (todos los valores 1).

Elipsoides Aquí en la figura 21 tenemos algunos valores representativos de elipsoides con diferentes valores en sus semiejes a,b y c. Los resultados mostraron que la incidencia de las normales contribuyó predominantemente al modo F_2 , con distribuciones similares a las distribuciones simples bipolares y tipo anillo de la sección anterior.

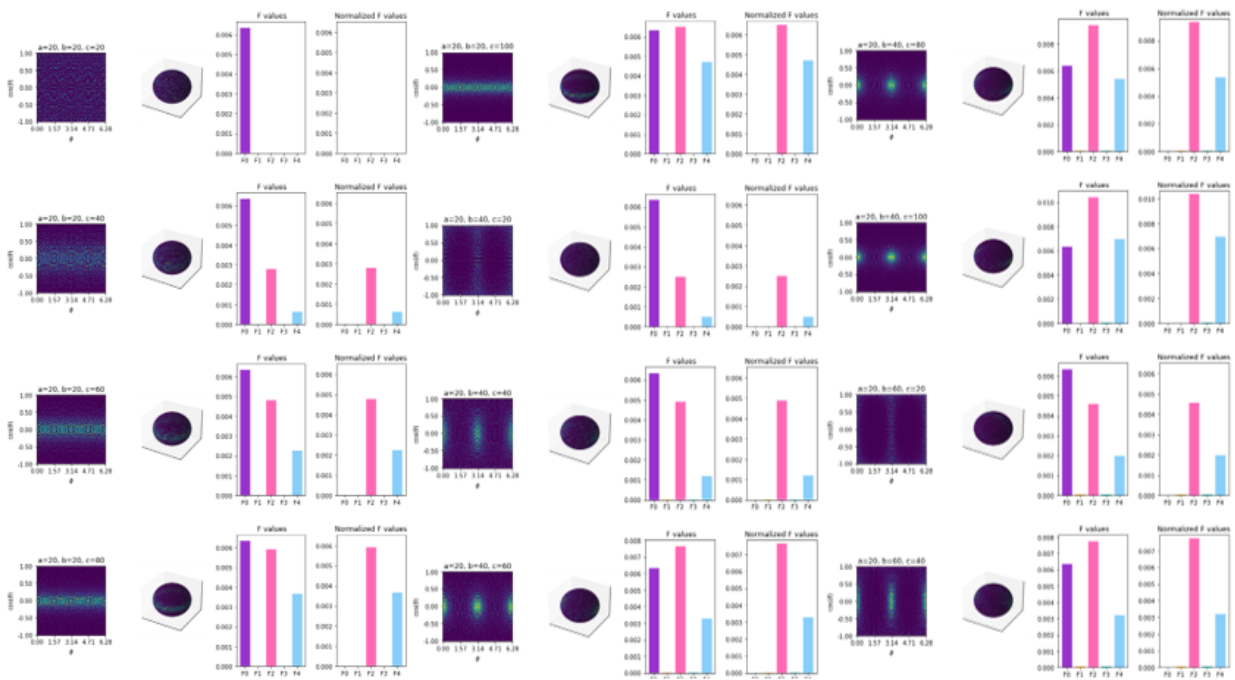


Figura 21: Ejemplos de distribuciones de Elipsoides con distintos valores en sus semiejes

Cuboides Se probó como afecta la geometría en estructuras con bordes definidos. Realizamos el mismo análisis anterior para distintos valores en los lados de los cuboides. Podemos ver algunos resultados representativos en 22.

El valor predominante es el modo F_4 , representando una distribución mucho mas compleja cuando vemos la proyección de las normales en nuestra grid $\cos(\theta)$ vs ϕ .

Sin importar la elongación obtenemos siempre que el modo F_4 es el mas alto, y puede servirnos para entender mejor los datos reales cuando la presencia de bordes puede estar influyendo en nuestro resultado, o la estructura reconstruida tiene una geometría similar a un cuboide en su compactación.

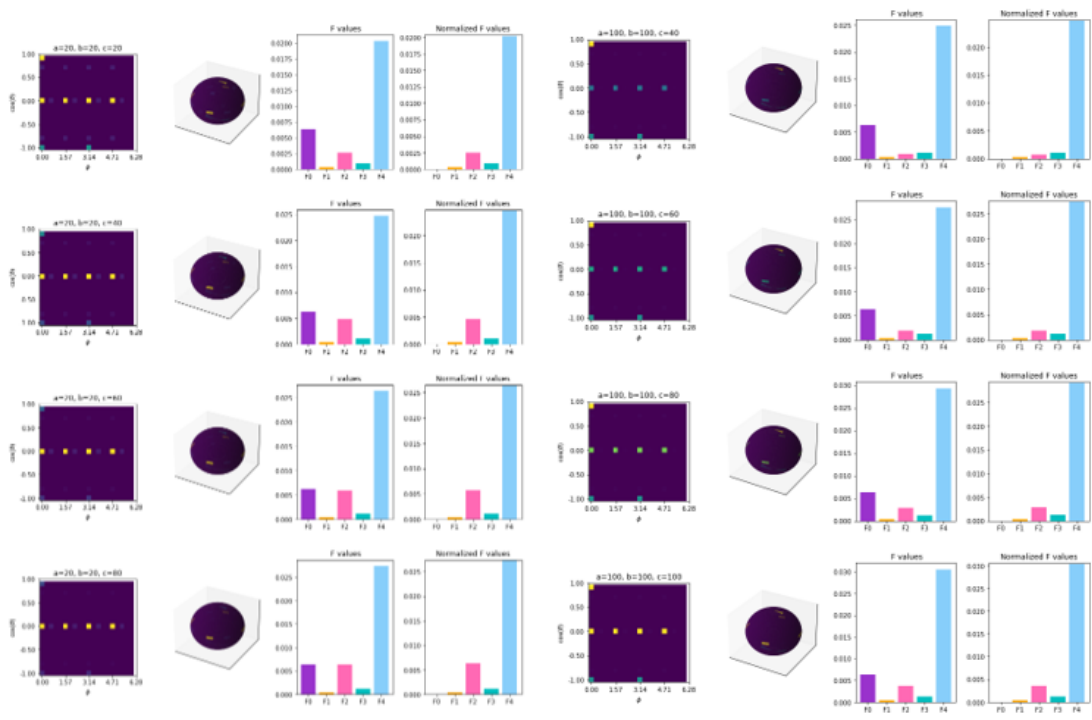


Figura 22: Ejemplo de distribuciones de Cuboides de diferente elongación sin contactos.

Dodecaedros Para finalizar esta sección, probamos con una geometría diferente: un dodecaedro, asignando diferentes pesos o «contactos» a sus caras para observar el efecto de los bordes o la existencia de caras puramente planas, como en el caso de los cuboides.

En la figura 23 están las etiquetas que asignamos a cada cara para posteriormente hacer los diferentes estudios, donde añadiremos peso 0 o 1 a alguna combinación de caras y 0 al resto.

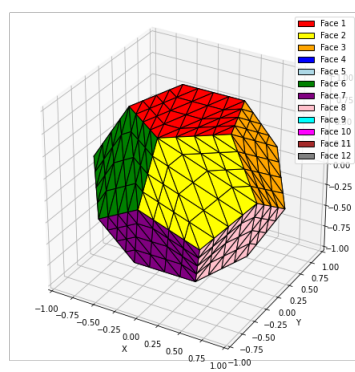


Figura 23: Dodecaedro caras.

Cuando añadimos peso a una sola cara encontramos el interesante resultado siguiente, donde todos los modos parecen aumentar de forma progresiva como observamos en 24.

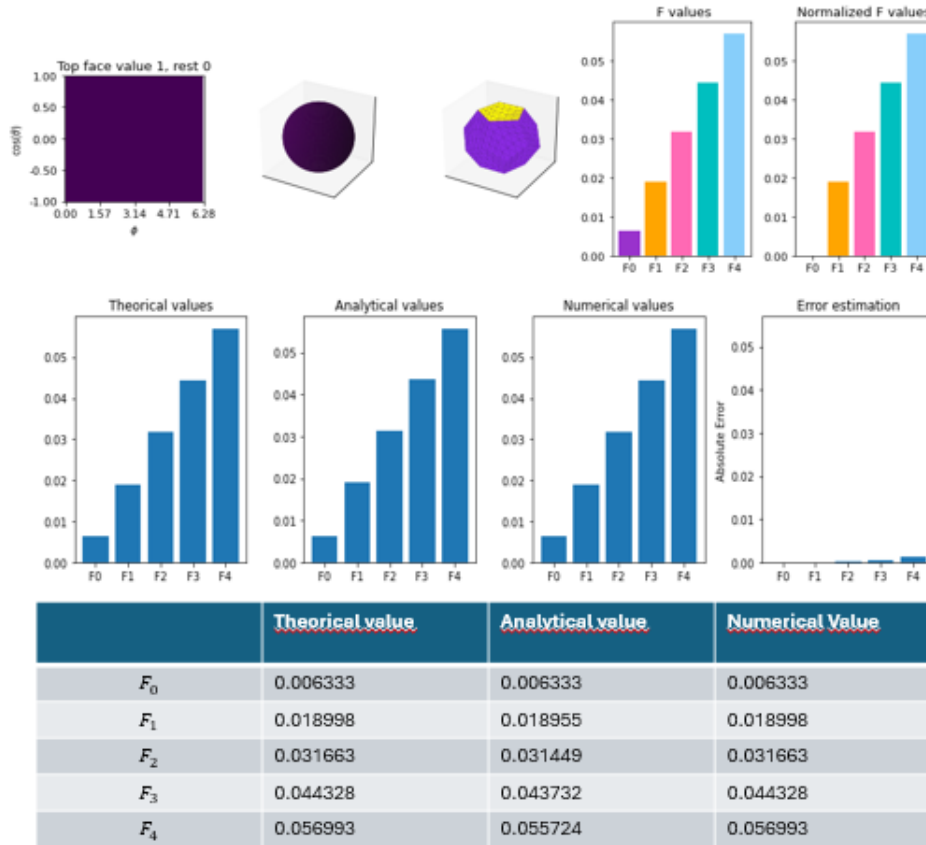


Figura 24: Análisis caso especial.

Para ello, comprobamos que los resultados numéricos coinciden exactamente con los teóricos como sigue:

Sea

$$f_l^m = \int \int_{S^2} d\Omega f(\theta, \phi) Y_l^{m*}(\theta, \phi)$$

, Donde para el caso de una cara nuestra función de distribución vendrá dada por:

$$f(\theta, \phi) = \frac{1}{\sin \theta_0} \delta(\theta - \theta_0, \phi - \phi_0),$$

con δ la delta de Dirac en coordenadas esféricas. De esta forma podemos obtener

$$S_{ff}(l) = \frac{1}{4\pi} \sum_m |f_l^m|^2$$

. Ya que $f_l^m = Y_l^{m*}(\theta_0, \phi_0)$, tenemos:

$$S_{ff}(l) = \frac{1}{4\pi} \sum_{m=-l}^l |Y_l^m|^2 \Big|_{\theta=\theta_0, \phi=\phi_0}$$

La estructura de los armónicos esféricos es conocida:

$$Y_l^m := (-1)^m \sqrt{\frac{2l+1}{4\pi}} \sqrt{\frac{(l-m)!}{(l+m)!}} P_l^m(\cos \theta) e^{im\phi}$$

Considerando que $P_l^m(1) = 1$ si $m = 0$ y $P_l^m(1) = 0$ si $m \neq 0$. Entonces, cuando tenemos una sola cara activa, podemos considerar que $\theta_0 = 0$ y $\phi_0 = 0$, con lo cual los armónicos esféricos adoptan la forma:

$$Y_l^m(0, 0) = (-1)^m \sqrt{\frac{2l+1}{4\pi}} \sqrt{\frac{(l-m)!}{(l+m)!}} \delta_{m,0}$$

Así hemos obtenido que $Y_l^0(0, 0) = \sqrt{\frac{2l+1}{4\pi}}$ y para $m \neq 0$ obtenemos $Y_l^m(0, 0) = 0$, por lo tanto:

$$S_{ff} = \frac{1}{4\pi} \frac{2l+1}{4\pi} = \frac{2l+1}{16\pi^2}$$

Así vemos que teóricamente aumenta linealmente con cada modo, como lo vimos numéricamente. También realizamos los cálculos de forma analítica usando software para calcular las integrales, y comparamos los valores en 24, viendo que nuestro código tiene valores muy cercanos a los esperados, comprobando una vez mas la validez del algoritmo.

Luego probamos con otras distribuciones, donde obtuvimos mayoritariamente el modo F_4 como para el caso del cuboide y un solo caso F_3 cuando tenemos dos caras contiguas con contacto.

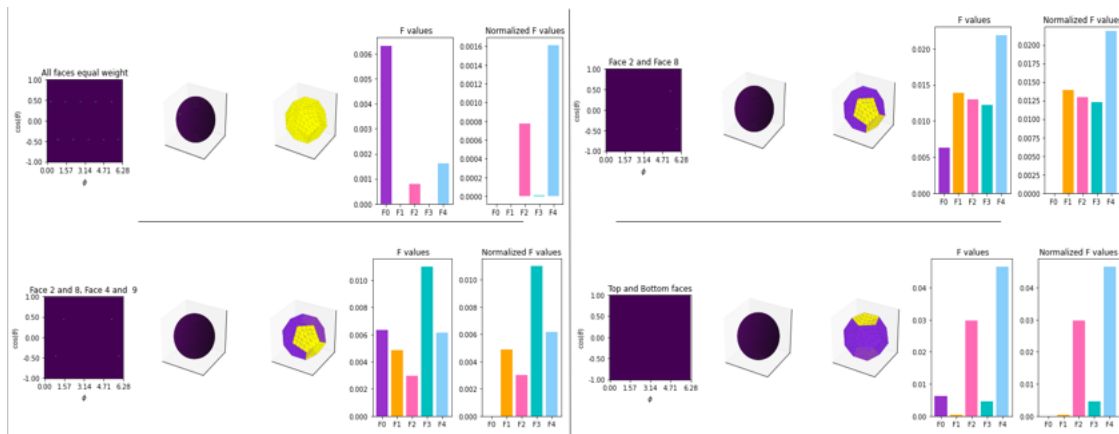


Figura 25: Distribuciones diferentes caras con contacto.

3.2.1.3 Estudio del efecto de los contactos celulares en la descripción de las formas celulares

Nos concierne ahora estudiar como distintos niveles de contacto interactúan con la morfología, ya sabemos que la forma de la estructura afecta en los modos y con los dodecaedros determinamos

que los contactos también juegan un rol importante en el resultado, por ello, analizaremos esferas, elipsoides y cuboides con distintas elongaciones, tamaños y niveles de contacto que definimos entre 0 (ningún contacto) y 1 (totalidad de contacto).

Esferas de diferentes escalas con diferentes niveles de contacto Se generaron mallas tridimensionales esféricas de diferentes tamaños, a cada tamaño le fuimos colocando un diferente nivel de contacto entre 0 y 1, el resultado fue mayoritariamente polar como era de esperado, debido a la similitud con la distribución polar simple revisada anteriormente.

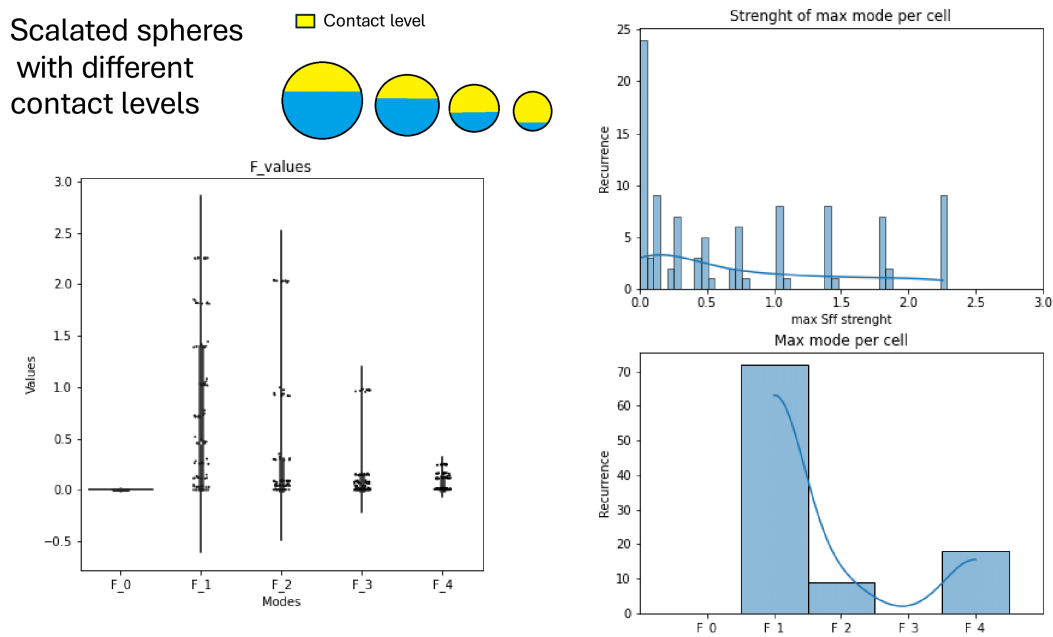


Figura 26: Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en esferas.

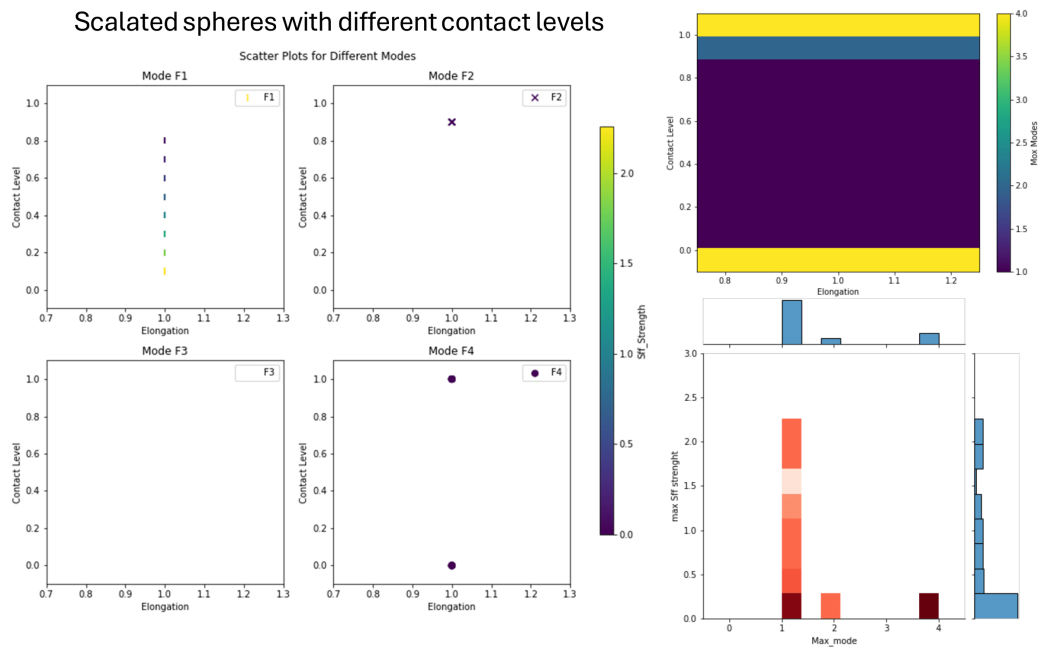


Figura 27: Gráfico individual de cada modo y su respectivo nivel de contacto y elongación (todas elongaciones iguales para el caso de las esferas).

Elipsoides con diferentes niveles de contacto en la dirección de elongación Se probaron diferentes elongaciones de estos elipsoides en la misma dirección de contacto, donde se varían niveles de contacto entre 0 (ningún contacto) y 1 (totalidad de contacto) para cada elongación distinta entre valores determinados para sus ejes a, b y c. Los resultados fueron los esperados: la morfología predominante se reflejó principalmente en el modo F_2 , mientras que el modo F_1 surgió cuando el nivel de contacto inducía un comportamiento polar en la distribución.

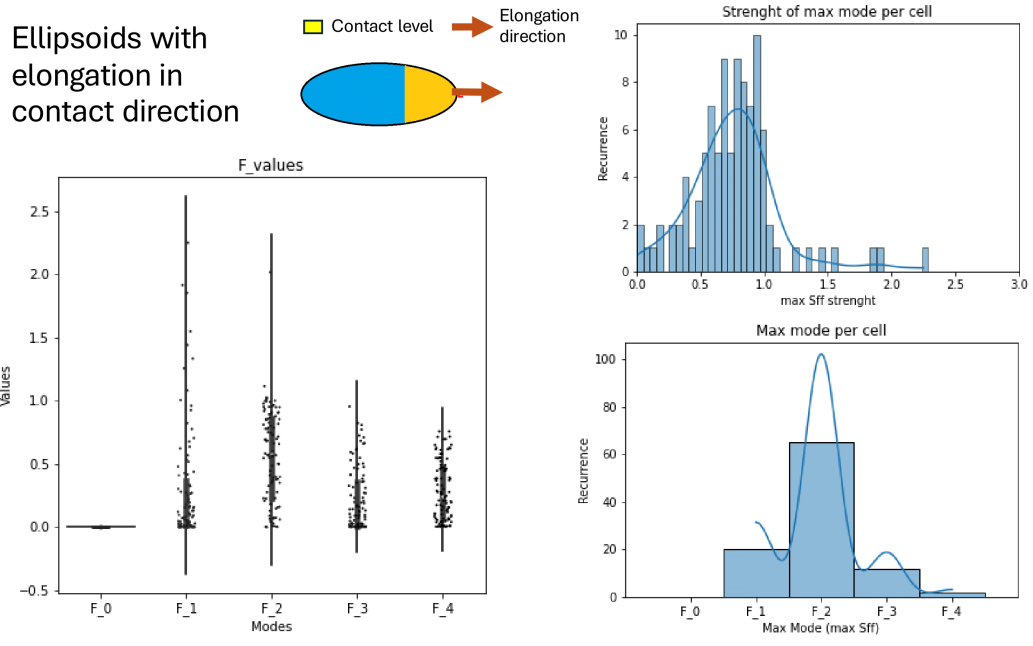


Figura 28: Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en los elipsoides.

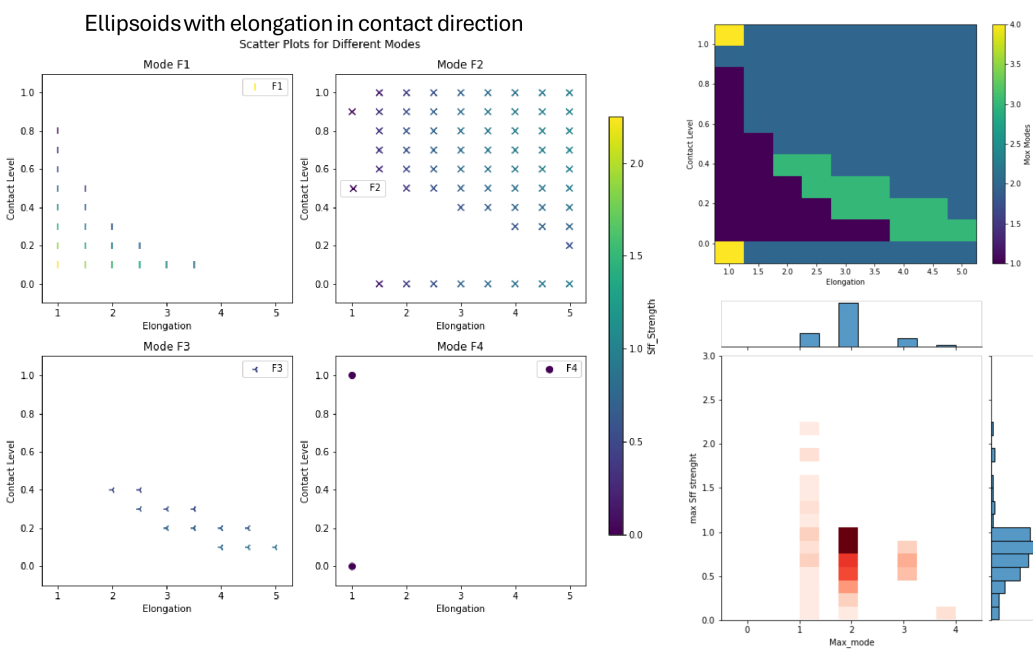


Figura 29: Gráficos individuales por modo para cada elongación y nivel de contacto. Gráfico en conjunto de todos los modos y análisis en mapa de calor de los modos para cada caso.

Elipsoides con diferentes niveles de contacto en la dirección perpendicular a la elongación

Siguiendo el mismo enfoque anterior, se evaluaron elipsoides elongados en la dirección perpendicular a los diferentes niveles de contacto, que varían entre 0 y 1. Como era de esperarse, la

morfología favoreció el modo polar F_1 , debido a que la mayoría de las normales apuntaban en la dirección de contacto. Además, cuando más de la mitad de la elipsoide tenía contacto, la morfología continuó predominando, resultando en una menor diferencia entre los modos.

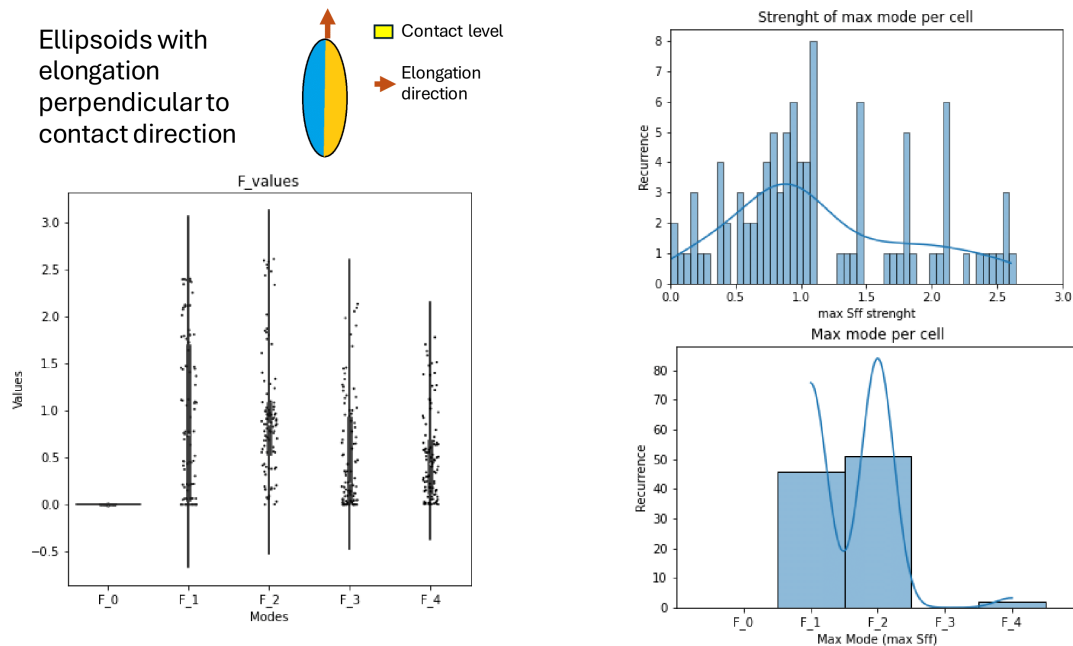


Figura 30: Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en elipsoides.

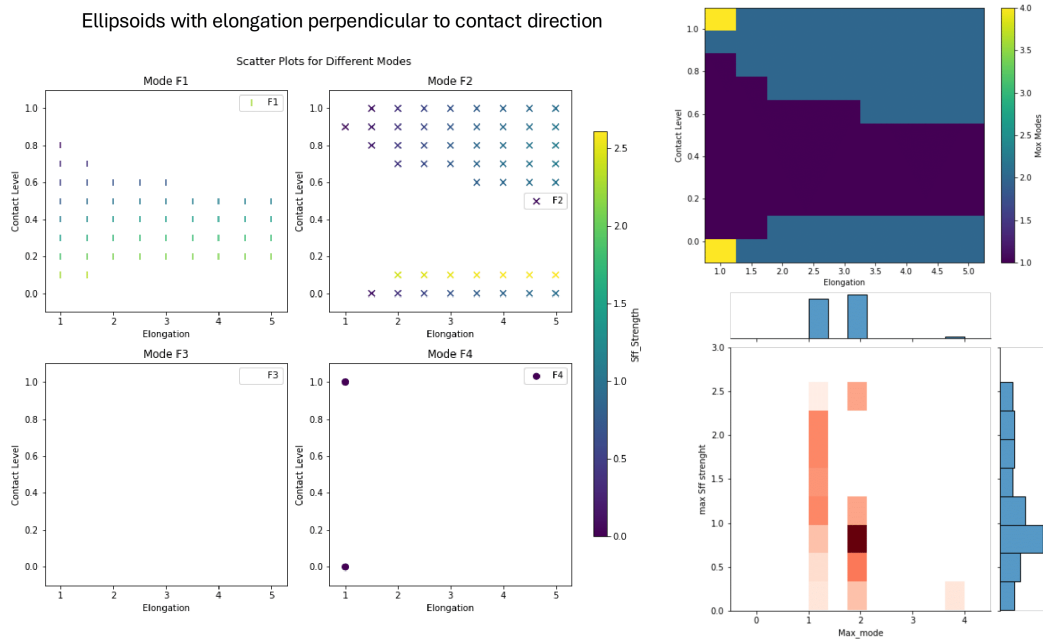


Figura 31: Gráficos individuales por modo para cada elongación y nivel de contacto. Gráfico en conjunto de todos los modos y análisis en mapa de calor de los modos para cada caso.

Elipsoides con diferentes niveles de contacto y elongación en ambas direcciones perpendiculares al contacto Concluyendo esta sección sobre elipsoides, ahora exploramos la elongación en ambas direcciones perpendiculares al punto de contacto (considerando una perspectiva tridimensional). Los resultados son consistentes con los obtenidos anteriormente, con una predominancia notable de los modos F_1 y F_2 , mostrando una mínima disparidad entre ellos. Esto sugiere que la morfología del elipsoide tiene mayor influencia que los puntos de contacto, lo cual resulta relevante para la extracción de información en los análisis posteriores.

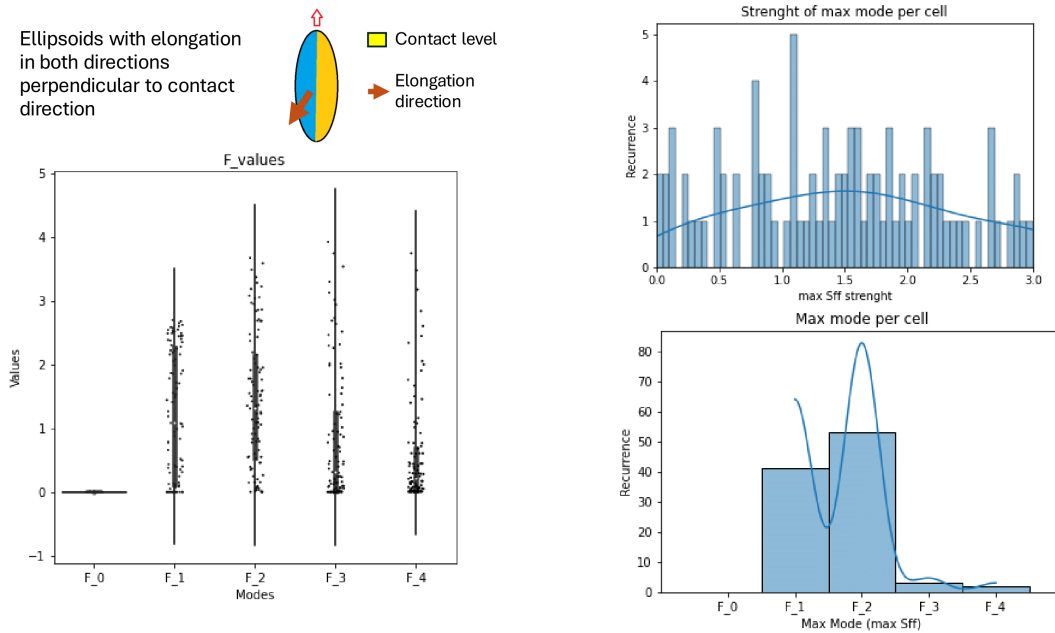


Figura 32: Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en elipsoides.

Ellipsoids with elongation in both directions perpendicular to contact direction

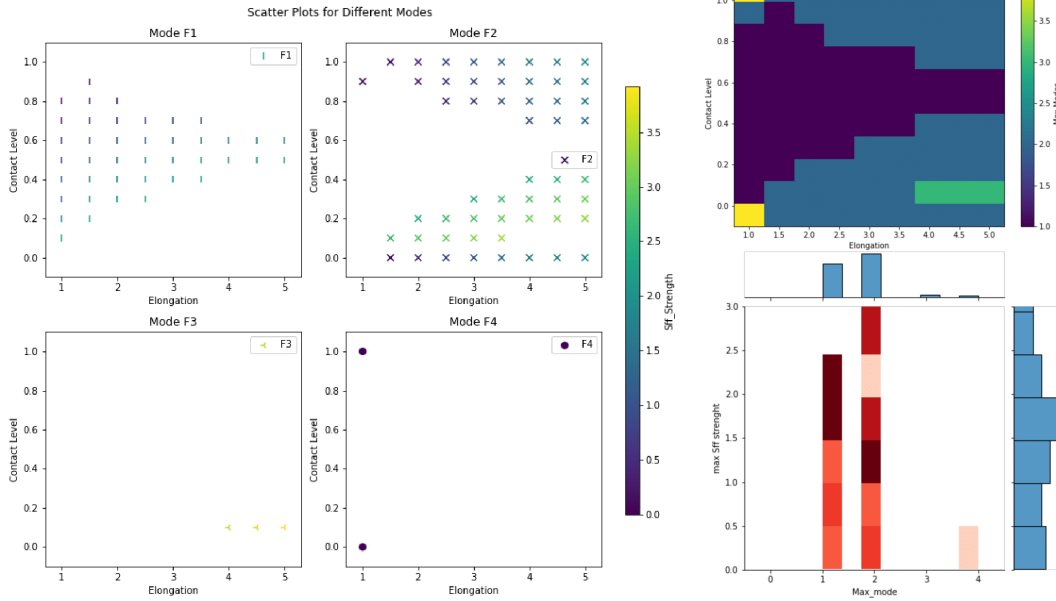


Figura 33: Gráficos individuales por modo para cada elongación y nivel de contacto. Gráfico en conjunto de todos los modos y análisis en mapa de calor de los modos para cada caso.

Cuboides con diferentes elongaciones perpendicular y paralelo a la dirección de diferentes niveles de contacto Realizamos el mismo procedimiento con los cuboides, explorando la elongación en direcciones tanto perpendiculares como paralelas a los puntos de contacto, a diferentes niveles. En todos los casos, la incidencia de otros modos fue mínima, destacando principalmente el modo F_4 . Esto sugiere que la morfología sigue siendo un factor crucial en la descripción del comportamiento estructural, independientemente de las variaciones en los puntos de contacto y la elongación.

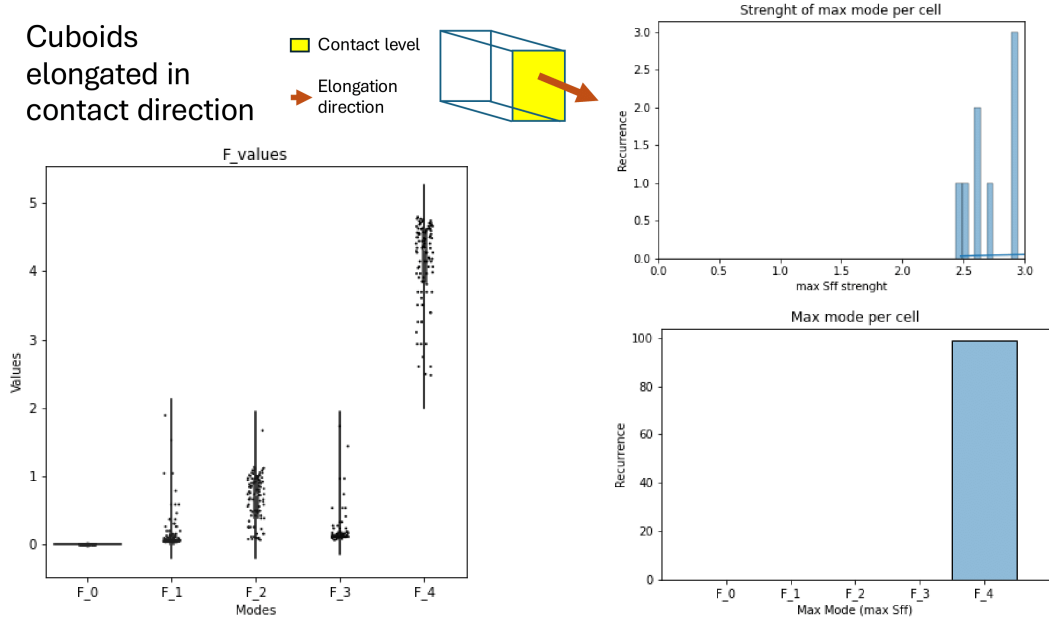


Figura 34: Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en cuboides

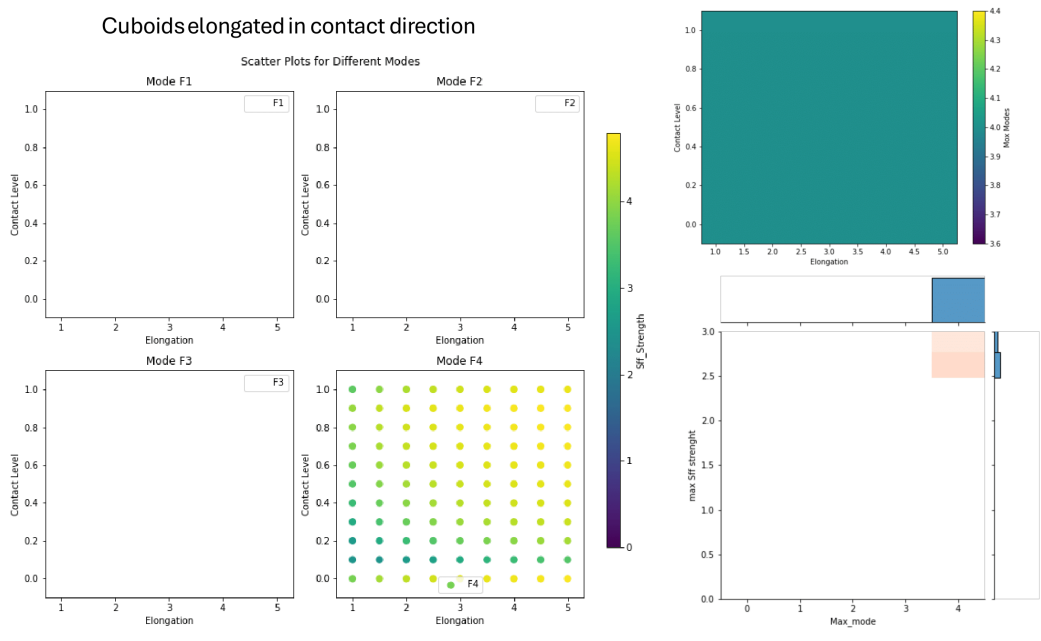


Figura 35: Gráficos individuales por modo para cada elongación y nivel de contacto. Gráfico en conjunto de todos los modos y análisis en mapa de calor de los modos para cada caso. Se observa presencia sólo del modo F_4 .

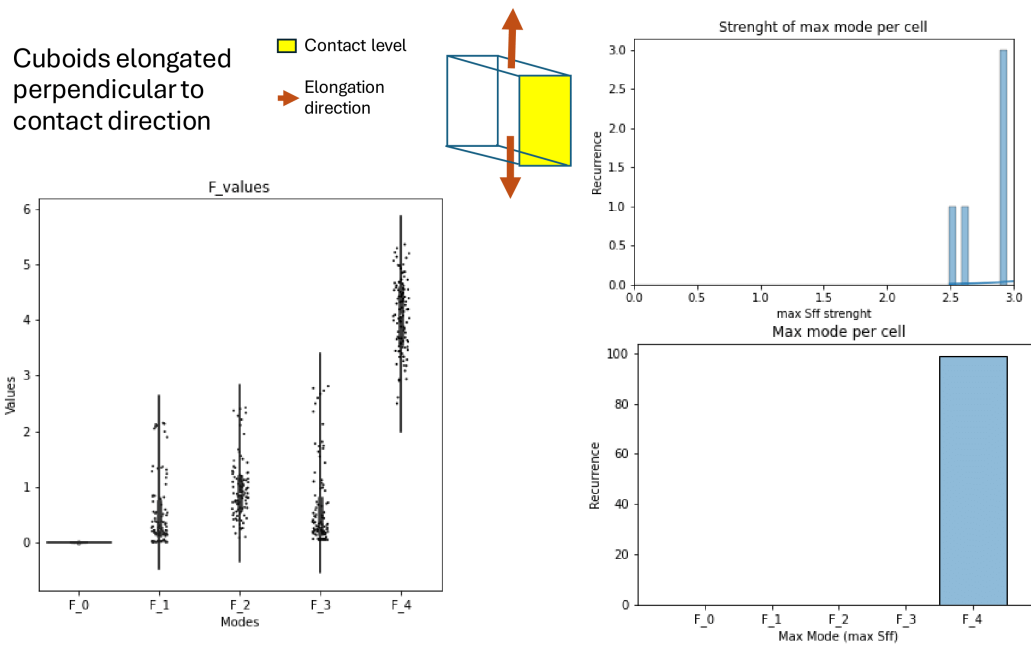


Figura 36: Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en cuboides

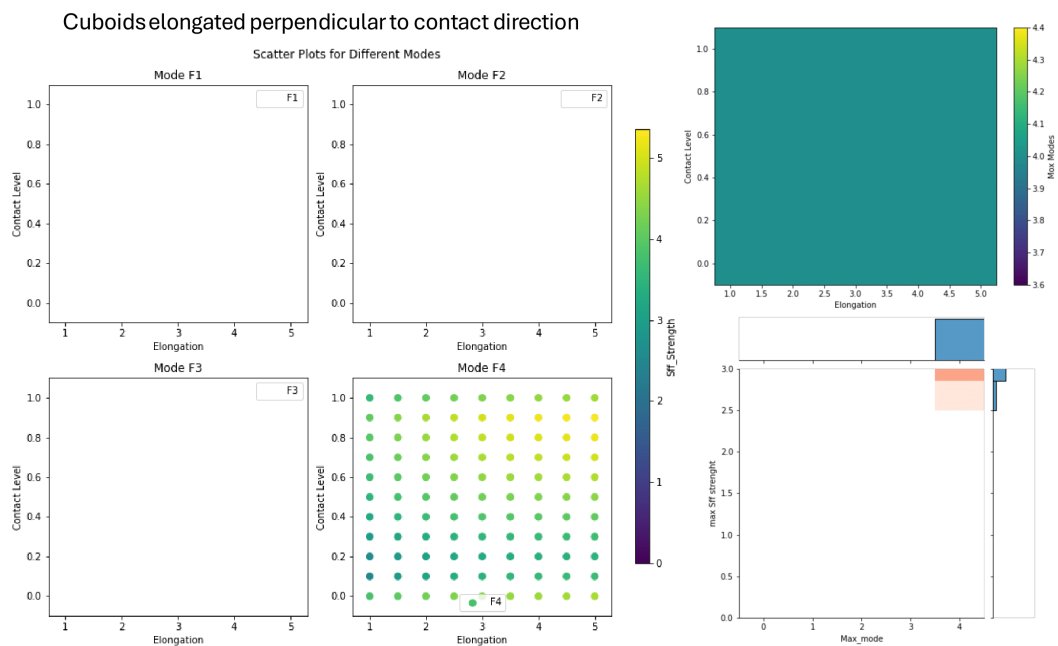


Figura 37: Gráficos individuales por modo para cada elongación y nivel de contacto. Gráfico en conjunto de todos los modos y análisis en mapa de calor de los modos para cada caso. Se observa presencia sólo del modo F_4 .

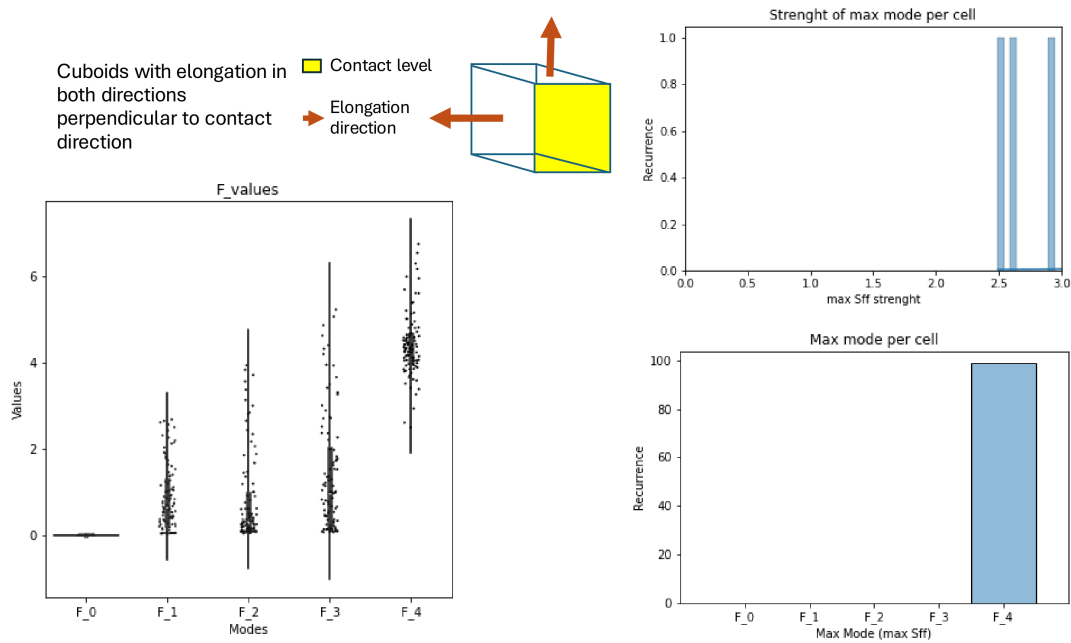


Figura 38: Gráficos de violín e histogramas para los modos máximos y su distribución dentro del ensamble de diferentes combinaciones de tamaños y niveles de contacto en cuboides.

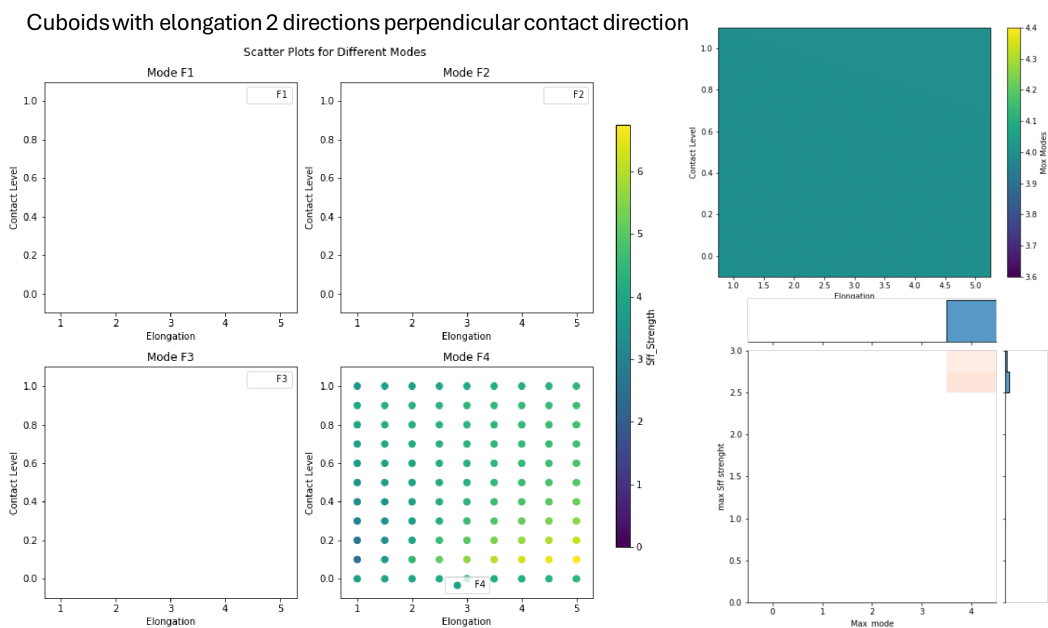


Figura 39: Gráficos individuales por modo para cada elongación y nivel de contacto. Gráfico en conjunto de todos los modos y análisis en mapa de calor de los modos para cada caso. Se observa presencia sólo del modo F_4

3.2.2. Caracterización tensorial de la forma celular en tres dimensiones

Muchas propiedades derivadas de la física y la mecánica microscópicas dependen de la dirección y, a menudo, sus componentes están distribuidas de forma aleatoria y discontinua a lo largo de diferentes orientaciones. La caracterización estadística de dichos datos direccionales es esencial para entender el problema físico subyacente, y es necesario adoptar una forma indiferente al marco de referencia o una forma tensorial que sea invariante bajo cambios de coordenadas.

Los datos direccionales pueden generalizarse como funciones de distribución de orientación («*Orientation Distribution Functions*», abreviado ODF en inglés). En su definición clásica, una ODF es en realidad una función de densidad de probabilidad de una variable de orientación aleatoria continua, que prácticamente representa la fracción del número total de elementos direccionales por unidad de ángulo a lo largo de una determinada dirección. Las ODF definidas clásicamente se pueden abstraer como funciones direccionales de valores escalares, porque desde el punto de vista del espacio vectorial, sus valores en cada orientación no son más que escalares normalizados. Las ODF de valores escalares se pueden caracterizar estadísticamente por sus tensores de textura. Estos tensores se denominan «de Textura» (*Fabric Tensors*) porque se han introducido ampliamente para describir la textura de materiales sólidos estructurados, como los granulares, medios microfisurados, policristales, polímeros y huesos, y han demostrado ser una herramienta eficaz uniendo los comportamientos multiescala de los materiales.[42]

Las aplicaciones recientes de los tensores de textura se extienden más allá de los estudios de materiales hasta dondequiera sean relevantes las variables direccionales, Los tensores de textura han sido usados por muchas décadas cuando sea que sea necesario caracterizar la distribución direccional de una colección de vectores unitarios, como en fluidos mixtos o análisis de imágenes médicas.

Las soluciones analíticas de estos tensores hasta órdenes superiores fueron obtenidos por Katanani [44], quien caracterizó los ODF con valores escalares en expansiones de tres tipos de tensores de textura y abordó su equivalencia con la expansión de la serie de Fourier en 2D y con la expansión de armónicos esféricos en 3D.

3.2.2.1 Cálculo de tensores, parámetros de forma y ejes nematicos

Discutimos los tensores de textura para caracterizar los diferentes modos en la distribución de contactos en los tejidos celulares y para obtener descriptores de forma que nos ayuden a entender cuantitativamente la morfología de las células. Estas definiciones caracterizan una sola estructura y no un conjunto.

Para un tensor de segundo orden, siguiendo la descripción de [41] (que es de interés para nosotros producto de la naturaleza nematica de las estructuras a estudiar), asumiendo que la superficie a analizar consiste en caras planares $1, 2, \dots, k$ y tienen vectores normales unitarios apuntando hacia afuera $n^{(1)}, n^{(2)}, \dots, n^{(k)}$. Cada cara tendrá áreas $A^{(1)}, A^{(2)}, \dots, A^{(k)}$, respectivamente, el tensor quedará definido como:

$$f_{ij} = \frac{1}{\sum_k A^{(k)}} \sum_k (A^{(k)} n_i^{(k)} n_j^{(k)})$$

Donde la suma en el índice (k) es en todas las caras de la malla.

Ya que es un tensor simétrico, sus autovalores son números reales y sus autovectores son a pares perpendiculares uno a otro.

Los tres autovectores de este tensor serán los tres principales ejes de la superficie analizada, que llamaremos sus «ejes nematicos» para el caso del tensor de segundo orden.

La traza del tensor es igual a 1, y sus tres autovalores son no-negativos, por lo que todos están entre 0 y 1 y suman hasta 1. Los tres autovalores del tensor $\lambda_1, \lambda_2, \lambda_3$ definen los siguientes parámetros:

- Compactación $C := \frac{\lambda_3}{\lambda_1}$
- Planitud $F := \frac{\lambda_1 - \lambda_2}{\lambda_1}$
- Elongación $E := \frac{\lambda_2 - \lambda_3}{\lambda_1}$

De manera similar a los autovalores propios $\lambda_1, \lambda_2, \lambda_3$, los tres descriptores de formas C, F y E también suman 1.

Este método de caracterización morfológica se puede aplicar fácilmente en el post-procesamiento al resultado de las técnicas habituales de obtención de imágenes de superficies en 3D, que producen una nube de puntos que representa la superficie de la estructura analizada con una cierta densidad y precisión, y a partir de ahí, se produce una malla con el software adecuado, en nuestro caso, usando el algoritmo «marching cubes» que describimos anteriormente.

El tensor de orientación de la superficie es fácil de determinar a partir de esta malla, y luego las tres métricas de compactación, planitud y elongación se pueden calcular simplemente utilizando sus valores propios. Es de destacar que la malla de superficie producida después del escaneo de la superficie puede ser triangular, cuadrilátera o sus caras pueden ser polígonos irregulares, ya que el tensor considera sólo el área y el vector normal de cada cara, mientras que la forma general del grano puede ser un poliedro arbitrario, incluso puede ser cóncavo.

3.2.3. Análisis espacial de la forma celular

En las secciones anteriores, hemos discutido cómo las células no suelen encontrarse de manera aislada. En la mayoría de los casos, forman parte de ensamblajes que constituyen tejidos o macroestructuras, las cuales se desplazan y funcionan como una unidad. Por esta razón, es fundamental estudiar las células como un material compuesto, lo que nos permite comprender su orden orientacional en el espacio.

Para capturar este orden orientacional dentro de las estructuras que analizamos, utilizamos parámetros de orden escalar, conceptos ampliamente utilizados en la teoría de cristales líquidos. Estos parámetros permiten describir de manera cuantitativa el grado de ordenamiento en un sistema de partículas o moléculas. En nuestro contexto, estos parámetros nos ayudan a entender cómo las células se orientan y organizan espacialmente dentro de un tejido.

3.2.3.1 Modelado de grano grueso para los parámetros de orden

Cuando calculamos los parámetros de orden S, P, D y C para un cristal líquido, aparte de los respectivos autovectores de cada tensor, necesitamos unos ejes de referencia x, y, z que es determinado por los ejes de simetría respectivos de la fase de nuestros objetos biaxiales. De una forma similar, podemos considerar el caso donde estos ejes de referencia estén dados por un segundo campo biaxial. Aquí introducimos el concepto de los parámetros de orden orientacional de grano grueso, también llamados parámetros de orden co-orientacional.

Parámetros de Orden Co-Orientacional Para calcular los parámetros de orden orientacional en el marco de referencia proporcionado por otro tensor, consideramos dos conjuntos de tensores nematicos $T_{\alpha\beta}^{(i)}$ y $K_{\alpha\beta}^{(i)}$. Los parámetros de orden se calculan en el marco de referencia definido por los autovectores del tensor de referencia.

Esto puede hacerse reemplazando los ejes de simetría \mathbf{x} , \mathbf{y} , \mathbf{z} por los autovectores de nuestro tensor de referencia. Denotamos como $\mathbf{l}^{(i)}$, $\mathbf{m}^{(i)}$, $\mathbf{n}^{(i)}$ los de nuestro tensor nematico $T_{\alpha\beta}^{(i)}$, y los autovectores del tensor de referencia $K_{\alpha\beta}^{(i)}$ son $\mathbf{u}^{(i)}$, $\mathbf{v}^{(i)}$, $\mathbf{w}^{(i)}$ [3].

Los parámetros de orden co-orientacional se definen como [3]:

$$co - S = \frac{1}{2N} \sum_i (3(\mathbf{n}^{(i)} \cdot \mathbf{w}^{(i)})^2 - 1), \quad (2)$$

$$co - P = \frac{3}{2N} \sum_i ((\mathbf{n}^{(i)} \cdot \mathbf{u}^{(i)})^2 - (\mathbf{n}^{(i)} \cdot \mathbf{v}^{(i)})^2), \quad (3)$$

$$co - D = \frac{3}{2N} \sum_i ((\mathbf{l}^{(i)} \cdot \mathbf{w}^{(i)})^2 - (\mathbf{m}^{(i)} \cdot \mathbf{w}^{(i)})^2), \quad (4)$$

$$co - C = ((\mathbf{l}^{(i)} \cdot \mathbf{u}^{(i)})^2 - (\mathbf{l}^{(i)} \cdot \mathbf{v}^{(i)})^2 + (\mathbf{m}^{(i)} \cdot \mathbf{v}^{(i)})^2 - (\mathbf{m}^{(i)} \cdot \mathbf{u}^{(i)})^2) \quad (5)$$

Notando explícitamente el índice i , que recorre todos los pares N de tensores nematicos ($T_{\alpha\beta}^{(i)}$, $K_{\alpha\beta}^{(i)}$). Es importante que notemos la principal diferencia que en los parámetros de orden usuales los ejes de referencia son los mismos para todo el sistema nemático, pero para este caso este sistema referencia vendrá dado por los ejes $\mathbf{u}^{(i)}$, $\mathbf{v}^{(i)}$, $\mathbf{w}^{(i)}$ que pueden variar en todo el sistema.

Interpretación de los parámetros de orden

- **Parámetro de orden co-orientacional co-S** Este parámetro mide el grado de alineamiento entre el eje principal de los tensores $T_{\alpha\beta}^{(i)}$ y $K_{\alpha\beta}^{(i)}$. Valores de $co-S$ cercanos a 1 indican un alto grado de alineación, mientras que valores cercanos a $-0,5$ sugieren una orientación perpendicular al eje. Valores cercanos a 0 representan un arreglo aleatorio.
- **Parámetro de orden co-orientacional co-P** Este parámetro cuantifica la diferencia en el alineamiento de los ejes secundarios de los tensores. Valores positivos indican que el eje principal de $T_{\alpha\beta}^{(i)}$ está mas alineado con uno de los ejes secundarios de $K_{\alpha\beta}^{(i)}$ que con el otro, mientras que valores negativos sugieren lo contrario.

- **Parámetro de orden co-orientacional co-D** Este parámetro mide la diferencia en el alineamiento de los ejes terciarios de los tensores. Un valor positivo de $co-D$ indica que el eje terciario de $T_{\alpha\beta}^{(i)}$ está más alineado con el eje principal de $K_{\alpha\beta}^{(i)}$, mientras que un valor negativo indica que el eje terciario de $T_{\alpha\beta}^{(i)}$ está más alineado con el eje secundario de $K_{\alpha\beta}^{(i)}$.
- **Parámetro de orden co-orientacional co-C** Este parámetro mide la simetría de los tensores en el plano definido por los autovectores secundarios. Un valor positivo sugiere que las orientaciones están más alineadas en una cierta dirección dentro del plano, mientras que un valor negativo indica una alineación en la dirección ortogonal dentro del mismo plano.

Estos parámetros de orden co-orientacional nos proporcionan una herramienta útil para entender la compleja estructura de materiales compuestos por componentes biaxiales, como los tejidos celulares. La variabilidad en los ejes de referencia permite capturar la dinámica y las interacciones locales entre los elementos del sistema, proporcionando una visión más detallada y precisa del orden orientacional en estos materiales.

3.2.3.2 Orden orientacional en sistemas no homogéneos

Como revisamos en las secciones anteriores de marco teórico, la gastrulación es un proceso durante el cual las células se reorganizan para formar las capas germinales que darán lugar a distintos tejidos y órganos del organismo. En el pez cebra, este proceso ocurre sobre una superficie aproximadamente esférica denominada yema, que proporciona una geometría tridimensional ideal para estudiar los movimientos y comportamientos celulares.

Para analizar las interacciones celulares y los patrones de movimiento durante la gastrulación es necesario suavizar y estimar funciones de densidad celular sobre la superficie de la yema. Los kernels son funciones que ponderan los datos en función de su distancia relativa, permitiendo una representación más continua y menos ruidosa de los datos. Un kernel con soporte compacto es particularmente útil, ya que limita su influencia a una región local alrededor de cada punto, lo que es ideal para capturar interacciones locales entre células.

En [35] fue utilizado un kernel compacto en 2D para analizar los datos de la gastrulación del pez cebra sobre una superficie esférica, el cual se define como:

$$f_k(\omega) = \frac{k+1}{2\pi} (\cos \omega)^k \cdot 1_{[\cos \omega > 0]}$$

donde ω es la distancia angular entre dos puntos en la esfera, k es un parámetro que controla la suavidad y el rango del kernel y $1_{[\cos \omega > 0]}$ es una función indicadora que asegura que el kernel solo tiene soporte en el hemisferio donde $\cos \omega > 0$.

Nuestro caso es tridimensional, por lo que necesitamos generalizar este kernel a tres dimensiones, para ello extendemos la definición de la distancia angular a coordenadas cartesianas tridimensionales. Dado un punto en coordenadas cartesianas $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ su representación en coordenadas esféricas (r, θ, ϕ) es:

$$r = \sqrt{x^2 + y^2 + z^2},$$

$$\theta = \arccos \frac{z}{r},$$

$$\phi = \arctan 2(y, x)$$

La distancia angular ω entre dos puntos en una esfera tridimensional, dados por sus ángulos (θ_1, ϕ_1) y (θ_2, ϕ_2) se calcula como:

$$\cos \omega = \frac{x_1 x_2 + y_1 y_2 + z_1 z_2}{r_1 r_2}$$

Utilizando esta distancia angular, el kernel en 3D se define de manera similar al kernel en 2D, manteniendo la función indicadora y la dependencia del parámetro k .

Es importante que notemos que el parámetro k determina la extensión del soporte y la suavidad del kernel. Para elegir un valor apropiado para k consideraremos la distribución de las distancias angulares en nuestros datos. Un enfoque común es utilizar el promedio de las distancias esféricas para estimar un k que cubra una proporción significativa de estas distancias.

```
import numpy as np

def spherical_distance_cartesian(x1, y1, z1, x2, y2, z2):
    r1 = np.sqrt(x1**2 + y1**2 + z1**2)
    r2 = np.sqrt(x2**2 + y2**2 + z2**2)
    cos_omega = (x1 * x2 + y1 * y2 + z1 * z2) / (r1 * r2)
    cos_omega = np.clip(cos_omega, -1.0, 1.0)
    omega = np.arccos(cos_omega)
    return omega
```

```

distances = []
for i in range(len(cell_positions)):
    for j in range(i + 1, len(cell_positions)):
        x1, y1, z1 = cell_positions[i]
        x2, y2, z2 = cell_positions[j]
        distance = spherical_distance_cartesian(x1, y1, z1, x2, y2, z2)
        distances.append(distance)
k = np.mean(distances)

```

De esto podemos ahora discutir la implementación del cálculo de un tensor de granulado grueso basado en un conjunto de tensores nemáticos discretos y sus posiciones espaciales. Esta metodología se fundamenta en la teoría del promediado espacial descrita anteriormente, donde utilizamos un kernel de ponderación esférico para considerar las contribuciones de los tensores vecinos.

El tensor de granulado grueso, siguiendo una estructura similar al descrito en [3] se definirá como:

$$\langle T_{\alpha\beta} \rangle_{\sigma}(r_k) = \frac{\sum_{i \neq k} g_k(\cos \omega, k) T_{\alpha\beta}^{(i)}}{\sum_{i \neq k} g_k(\cos \omega, k)} \quad (6)$$

donde $g_k(\cos \omega, k)$ es el kernel de ponderación esférico compacto que depende del coseno del ángulo ω entre las posiciones r_k y r_i , y $T_{\alpha\beta}^{(i)}$ son los tensores nemáticos discretos.

El cálculo se implementó de la siguiente forma:

- Se inicializa una lista para almacenar los tensores promediados junto con sus respectivos IDs de cada célula.
- Cálculo de Distancias Esféricas: se escribe una función que calcula el coseno del ángulo entre dos vectores en coordenadas cartesianas.
- Kernel de Ponderación: La función g_k implementa el kernel gaussiano esférico basado en la distancia angular, ajustado para k .
- Bucle Principal: Para cada tensor individual, se identifican los tensores vecinos utilizando los IDs de contacto. Se calcula el tensor ponderado acumulado y la suma de los pesos.

El tensor de granulado grueso se obtiene dividiendo el tensor ponderado acumulado por la suma de los pesos.

4. Capítulo 3: Validación del método para datos artificiales

En esta sección, se presentan y analizan datos artificiales que simulan diversas estructuras biológicas. Estos datos sirven como modelos computacionales que imitan diferentes configuraciones celulares encontradas en organismos vivos. En particular, se explorarán tres tipos de estructuras: una capa multicelular, un domo celular y un cascarón celular. Estas representaciones computacionales ofrecen una oportunidad para comprender y estudiar la organización y dinámica de las células en un contexto simulado, previo a aplicar los análisis en los datos reales.

Es importante destacar que, aunque estos datos son artificiales, ofrecen valiosas percepciones sobre la organización celular. Sin embargo, se debe tener en cuenta que las morfologías celulares en estos modelos pueden diferir de las células reales. Además, es crucial reconocer la presencia de artefactos en los datos, como caras planas o compactaciones no naturales, que pueden distorsionar la percepción de la estructura celular y deben ser considerados al interpretar los resultados.

Capas de Células El primer conjunto de datos simula una capa de células organizadas en múltiples capas. Este modelo proporciona información sobre la disposición tridimensional de las células y cómo se organizan en diferentes estratos.

Análisis de descomposición en modos esféricos Las células mostraron un comportamiento mayoritariamente del modo F_4 , lo que coincidía con las células en el «Bulk» del tejido artificial, la mayoría de las células de este modo más alto tenían contacto en todas sus caras y una apariencia más compacta y de cuboide, asemejándose a los resultados encontrados en estas geometrías.

Importante notar que este es un artefacto producto de como fueron generados los datos y de la naturaleza artificial y poco natural de las células ficticias generadas.

Por otro lado del total de células sólo 53 tuvieron un comportamiento nematico, siendo células intermedias entre los contactos con las células del borde (que tienen un comportamiento polar)

y las células del bulk que tienen un comportamiento de modo F_3 y F_4 .

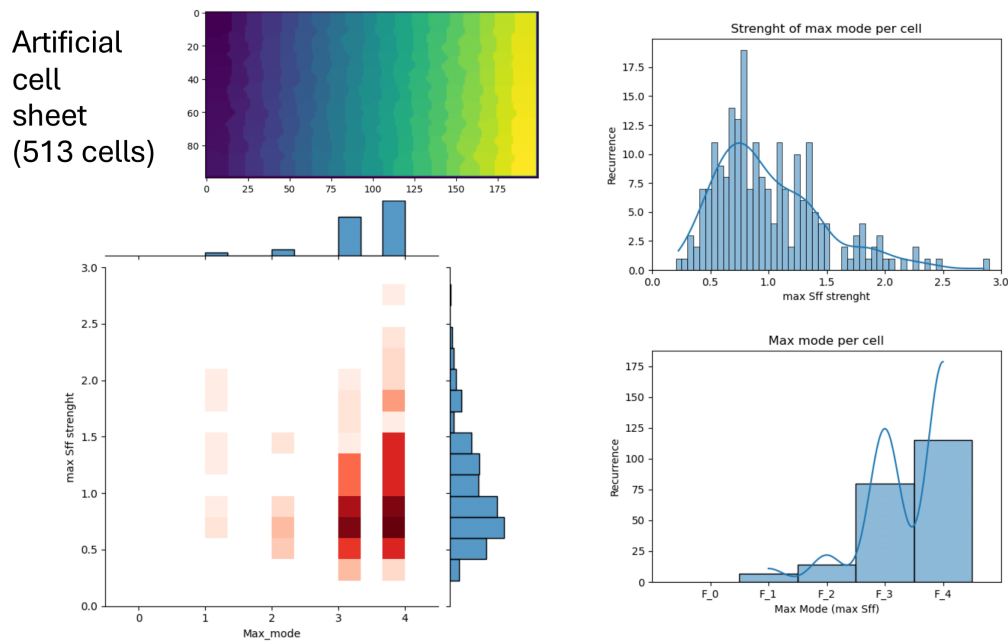


Figura 40: Análisis de descomposición en modos esféricos.

Comparación tensores nemáticos y tensores con modelado de grano grueso En este caso se aplicó un kernel gaussiano euclidiano para procesar los tensores dada la geometría del tejido artificial.

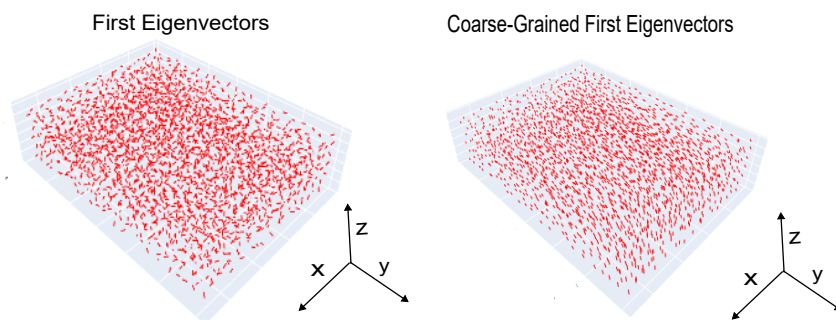


Figura 41: Comparación cualitativa primer autovector tensor nemático y tensor con modelado de grano grueso usando filtro gaussiano euclidiano.

Domo de Células El segundo conjunto de datos representa un domo celular, inspirado en la distribución de células que forman un domo sobre una yema en el embrión del pez cebra. Esta estructura tridimensional permite investigar cómo las células interactúan y se organizan para

formar una estructura morfológicamente más compleja.

Análisis en descomposición de modos esféricos Nuevamente se ve una gran incidencia del modo F_4 y F_3 producto de como fueron generados los datos artificiales. Estos coinciden con las células en la parte interior del tejido, el modo F_1 lo toman las células a los bordes (polarizadas) y el modo F_2 las células intermedias, siendo 609 células del total las con un comportamiento nemático.

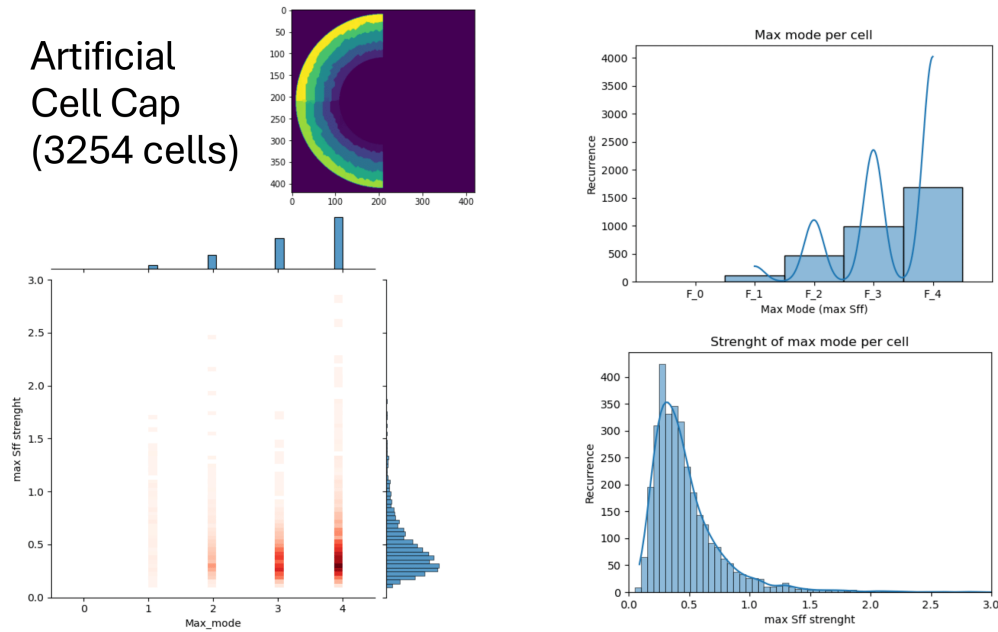


Figura 42: Análisis en descomposición de modos esféricos para el domo de células.

Comparación de tensores nemáticos y tensores de modelado de grano grueso Se utilizó el kernel esférico compacto en tres dimensiones para el procesamiento de los ejes, se puede observar un leve modelado de grano grueso para los valores estimados óptimos según el análisis de la sección anterior, al contrario de un kernel gaussiano esférico que alteraba completamente los resultados.

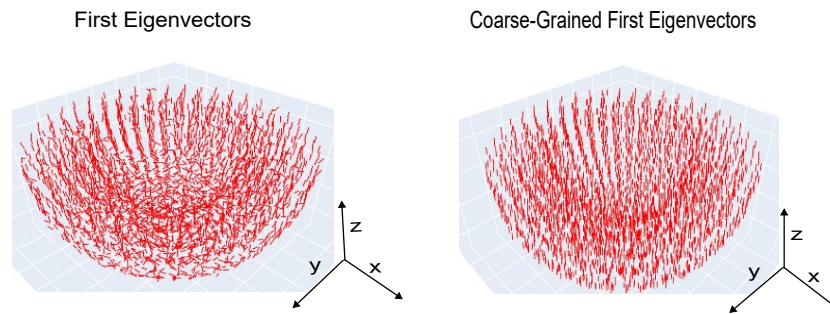


Figura 43: Comparación cualitativa primer autovector tensor nemático y tensor con modelado de grano grueso usando filtro esférico compacto 3D

4.1. Cascarón de células

Finalmente, se examina un cascarón celular, que simula la distribución homogénea de células sobre la yema del embrión del pez cebra. Este modelo ofrece perspectivas sobre la disposición espacial de las células en una estructura más uniforme.

Análisis en descomposición de modos esféricos En la misma línea de los análisis anteriores, fue predominante el modo F_4 y F_3 , correspondiente a las células más profundas del tejido con todas sus caras con contactos. Los modos F_1 y F_2 también son observados en una cantidad menor, con 1519 células del total con un comportamiento nemático.

Artificial Cell Shell (6565 cells)

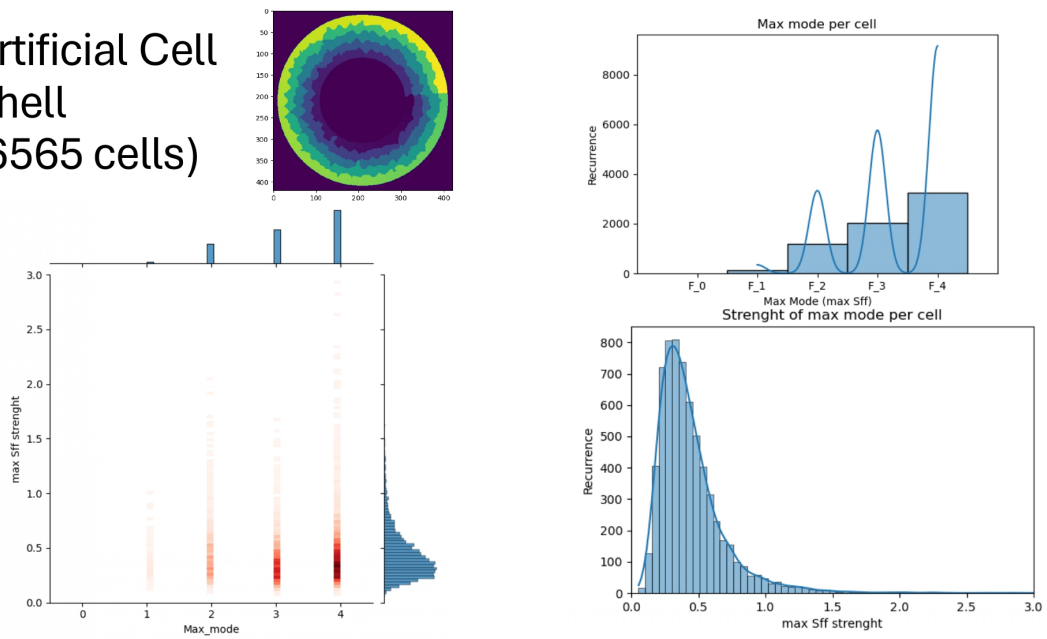


Figura 44: Análisis en descomposición de modos esféricos para un cascarón de células.

Comparación tensores nemáticos y tensores con modelado de grano grueso Se utilizó nuevamente el kernel esférico compacto en tres dimensiones dada la geometría del sistema y se compararon los resultados para asegurarnos de un suavizado de los datos que no alterara completamente la estructura del sistema.

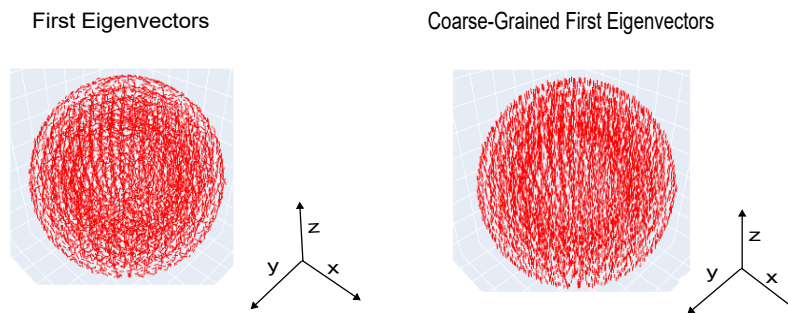


Figura 45: Comparación cualitativa primer autovector tensor nemático y tensor con modelado de grano grueso usando filtro esférico compacto 3D

4.2. Análisis de los parámetros de orden

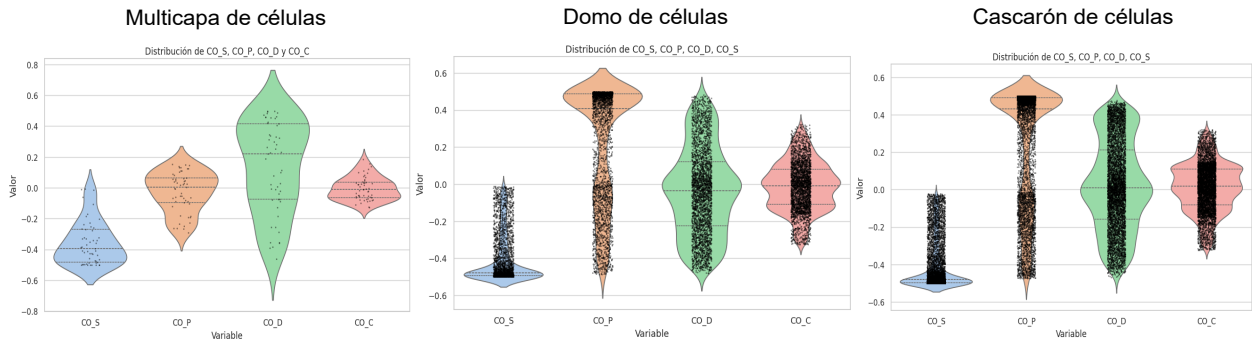


Figura 46: Gráficos de violín de las distribuciones de descomposición en modos esféricos para cada grupo de datos

Multicapa de células

CO_S

- Los valores de CO_S están concentrados cerca de -0.4 a 0. La forma de violín muestra que hay una mayor densidad de valores alrededor de -0.4, indicando que la mayoría de las orientaciones son perpendiculares o aleatorias al eje de referencia, lo que sugiere un grado bajo de alineación entre los ejes principales de los tensores $T_{\alpha\beta}^{(i)}$ y $K_{\alpha\beta}^{(i)}$.

CO_P

- Los valores están centrados alrededor de -0.2 a 0.2, con una distribución más compacta. Esto sugiere que no hay una fuerte preferencia en la alineación de los ejes secundarios de los tensores, indicando un alineamiento aleatorio en esta dimensión.

CO_D

- La forma de violín es más ancha y los valores se extienden desde -0.4 a 0.6. Un valor positivo de CO_D sugiere que el eje terciario de $T_{\alpha\beta}^{(i)}$ está más alineado con el eje principal de $K_{\alpha\beta}^{(i)}$, mientras que un valor negativo indica alineación con el eje secundario. La variabilidad observada sugiere diferentes grados de alineamiento en esta muestra.

CO_C

- Los valores están concentrados cerca de 0 con una distribución más estrecha, lo que indica una simetría moderada en el plano definido por los autovectores secundarios. Esto sugiere que las orientaciones están alineadas de manera más simétrica dentro del plano, sin una fuerte preferencia direccional.

Domo de células

CO_S

- Los valores están concentrados cerca de -0.4 a 0, similar al primer gráfico. Esto implica una orientación baja o aleatoria entre los ejes principales de los tensores $T_{\alpha\beta}^{(i)}$ y $K_{\alpha\beta}^{(i)}$.

CO_P

- Los valores están centrados alrededor de 0.4, indicando una mayor alineación polar en esta muestra. Esto sugiere que el eje principal de $T_{\alpha\beta}^{(i)}$ está más alineado con uno de los ejes secundarios de $K_{\alpha\beta}^{(i)}$.

CO_D

- La distribución es amplia y muestra una gran variabilidad. Hay diferentes grados de alineamiento del eje terciario, con muchos valores positivos, sugiriendo una alineación mayor con el eje principal de $K_{\alpha\beta}^{(i)}$.

CO_C

- Los valores están concentrados alrededor de 0, similar al primer gráfico. Esto sugiere una simetría moderada en el plano definido por los autovectores secundarios.

Cascarón de células

CO_S

- Los valores están concentrados cerca de -0.4, similar a los gráficos anteriores. Esto implica una orientación baja o aleatoria entre los ejes principales de los tensores.

CO_P

- Los valores están alrededor de 0.4, similar al gráfico del domo de células. Esto sugiere una alineación polar más fuerte en esta muestra.

CO_D

- La distribución es amplia y variada, similar a los gráficos anteriores. Esto sugiere diferentes grados de alineamiento del eje terciario, con una variabilidad considerable en el alineamiento de los ejes terciarios de los tensores.

CO_C

- Los valores están concentrados alrededor de 0, similar a los otros gráficos. Esto sugiere una simetría moderada en el plano definido por los autovectores secundarios en todas las muestras.

Resumen General

- **Consistencia en CO_S :** En todas las muestras, los valores de CO_S están bajos, lo que indica una alineación baja o aleatoria entre los ejes principales de los tensores $T_{\alpha\beta}^{(i)}$ y $K_{\alpha\beta}^{(i)}$. Esto concuerda con la posible naturaleza aleatoria de los datos.
- **Variabilidad en CO_P :** Los valores de CO_P son mayores en el domo celular y el cascarón celular, sugiriendo una alineación polar más fuerte en estas muestras, posiblemente debido a su geometría esférica.

- **Gran variabilidad en CO_D :** Este parámetro muestra una gran variabilidad en todas las muestras, indicando diferentes grados de alineamiento del eje terciario de los tensores.
- **Moderación en CO_C :** Los valores de CO_C están concentrados alrededor de 0, lo que sugiere una simetría moderada en el plano definido por los autovectores secundarios en todas las muestras.

5. Capítulo 4: Aplicación en datos reales de gastrulación temprana del pez cebra

En este capítulo se detalla el proceso de pre-procesamiento y segmentación de datos tridimensionales de trayectorias celulares, utilizando los datos publicados en el artículo «Multi-scale imaging and analysis identify pan-embryo cell dynamics of germlayer formation in zebrafish» por Shah et al. 2019. [45] Este trabajo proporciona una comprensión profunda de las dinámicas celulares a escala completa durante la formación de capas germinales en el embrión de pez cebra, y nuestros análisis buscan profundizar en estos datos mediante técnicas de procesamiento y segmentación.

5.1. Datos reales tridimensionales del embrión del pez cebra

Los datos analizados corresponden al seguimiento en tres dimensiones de los núcleos celulares desde las 4.0 horas post fertilización hasta las 18.0 horas post fertilización. Fueron analizados los 419 tiempos que fueron capturados en el artículo, por razones de espacio mostraremos los resultados con cambios mas evidentes que son el tiempo 0.0, 50.0, 100.0, 200.0 y 300.0 del archivo .csv disponible en el artículo.

5.1.1. Análisis de descomposición en modos esféricos

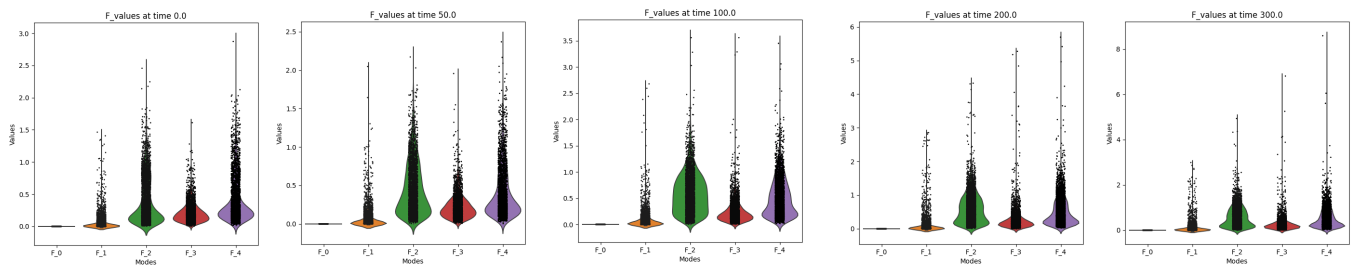


Figura 47: Descomposición en modos esféricos para diferentes tiempos

Gráfico de Tiempo 0.0

- **F_1**: Este modo tiene pocos valores, sugiriendo una baja contribución de un comportamiento polarizado pero aun así presente, lo que relacionamos con células líderes, que exhiben comportamientos de fluido activo.
- **F_2**: Se observa una mayor concentración de valores, lo que podría indicar una tendencia a una distribución más estructurada, la que podría ser nematica.
- **F_3 y F_4**: Ambos modos presentan valores altos, sugiriendo complejidad en la distribución. Sin embargo es importante recordar que al ser reconstrucciones artificiales de las membranas es probable que estos modos sean mucho mas bajos y su contribución sean artefactos como fue detectado en las secciones anteriores.

Gráfico de Tiempo 50.0

- **F_1**: Los valores siguen siendo bajos, con un leve aumento, lo cual es consistente con el tiempo 0.0.
- **F_2**: Continúa siendo un modo dominante, parece ser la distribución más grande y predominante en el conjunto de datos. Esto se debe a su mayor anchura en varias partes del violín, indicando una mayor densidad de datos en un rango más amplio de valores, mostrándonos que la tendencia estructurada persiste.

- **F_3 y F_4:** Los valores se mantienen altos y similares al tiempo 0.0, con una ligera dispersión mayor.

Gráfico de Tiempo 100.0

- **F_1:** No muestran cambios significativos.
- **F_2:** Aumenta su rango y densidad, reafirmando la presencia de una estructura nemática más pronunciada.
- **F_3 y F_4:** Ambos modos muestran un ligero incremento en su dispersión y valores máximos.

Gráfico de Tiempo 200.0

- **F_1:** No muestran cambios significativos.
- **F_2:** Comienza a tener un decrecimiento en su rango y la anchura del violín comienza a descender, mostrando una pérdida de la forma estructurada del sistema.
- **F_3 y F_4:** Ambos modos muestran un ligero decrecimiento en su dispersión y valores máximos.

Gráfico de Tiempo 300.0

- **F_1:** Permanecen prácticamente sin cambios.
- **F_2:** Presenta un decrecimiento considerable respecto a los tiempos anteriores.
- **F_3 y F_4:** Los modos también muestran un decrecimiento notable en la anchura de sus distribuciones, pero sus valores máximos no han variado mucho.

Conclusiones

- **Modo F_2:** Si bien al comienzo la estructura es predominantemente organizada de forma nemática, su presencia comienza a decrecer con el paso del tiempo perdiendo su forma estructurada.

- **Modos F_3 y F_4:** Estos modos indican una complejidad estructural que también aumenta con el tiempo.
- **F_0 y F_1:** Permanecen relativamente bajos y constantes, sugiriendo que la contribución isotrópica y la componente vectorial no cambian significativamente.

5.1.2. Análisis parámetros de orden

Utilizando un filtro Gaussiano tridimensional compacto enunciado en la sección anterior, analizamos los datos pasados por la formulación de grano grueso, como observamos en 49:

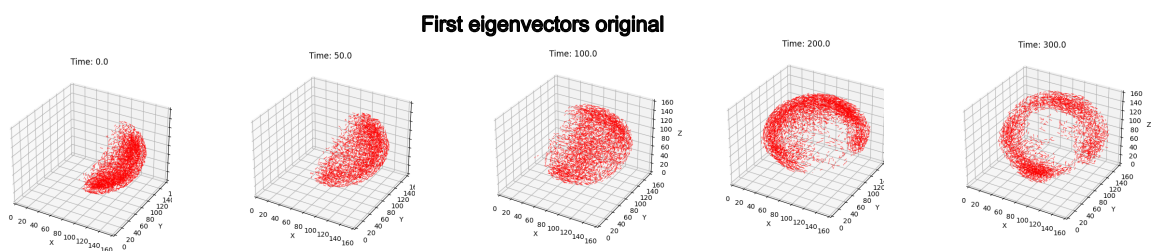


Figura 48: Gráfico de primer autovector de los tensores nematicos antes del modelado de grano grueso para distintos tiempos

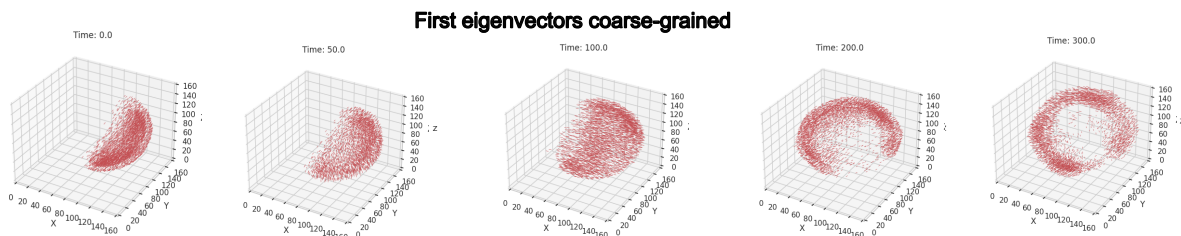


Figura 49: Gráfico de primer autovector de los tensores nematicos de modelado de grano grueso para distintos tiempos

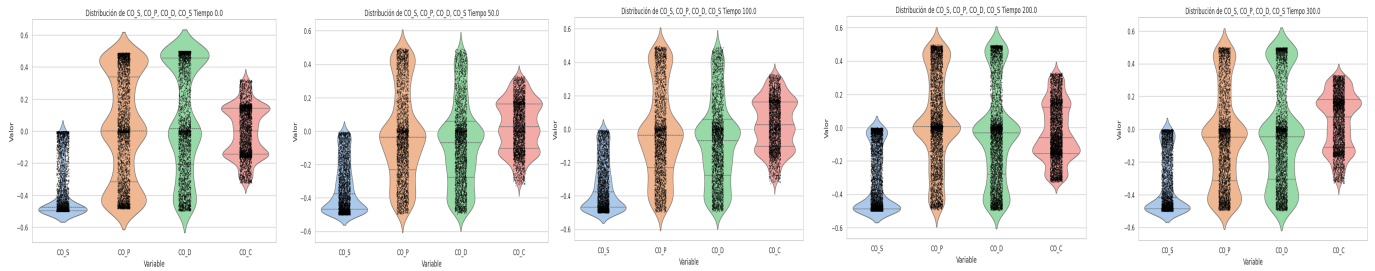


Figura 50: Análisis parámetros de orden para diferentes tiempos

Tiempo 0.0

- **co-S**: Un valor negativo de *co-S* indica que las células están inicialmente bien alineadas en una dirección específica perpendicular al eje de referencia. Esto sugiere que la estructura inicial del material es altamente organizada, con algunas células en estado aleatorio con sus valores cercanos a 0.
- **co-P** : Un valor positivo de *co-P* sugiere que las orientaciones secundarias están más alineadas con un eje específico. Esto indica una estructura biaxial con una orientación preferencial clara para los ejes secundarios entre los tensores, similar a un cristal líquido biaxial. Aun así podemos observar que existe una variabilidad amplia en los datos, con valores cercanos a 0 y también con alineamiento nulo entre los ejes secundarios. Por lo que parecen estar coexistiendo diferentes ordenamientos en este tiempo.
- **co-D**: Un valor positivo de *co-D* indica que los ejes terciarios están alineados con los ejes principales. Esto sugiere una estructura donde los planos terciarios mantienen una relación ordenada con los ejes principales, mostrando una simetría tridimensional clara. La existencia de valores negativos nos indica que el eje terciario de los tensores está más alineado con el eje secundario de nuestro tensor de referencia.
- **co-C**: Podemos ver coexistencia de orientaciones alineadas dentro del plano definido por los autovectores secundarios y otras alineadas ortogonalmente al plano.

Tiempo 50 y 100.0

- **co-S**: indica que las células empiezan a alinearse más de forma perpendicular al eje de referencia, sugiriendo una transición hacia un estado más ordenado. Esto puede reflejarse

como una transición de fase en el cristal líquido, donde el orden nemático comienza a aumentar levemente. A grandes rasgos de mantiene la configuración del tiempo 0.0 en este eje, con una muy leve variabilidad.

- **co-P** : Aumento en los valores cercanos a cero indicarían aumento en la desorganización del sistema, por otro lado aumento en valores negativos de indicarían una re-orientación opuesta a la inicial, sugiriendo que las orientaciones secundarias están cambiando significativamente, posiblemente invirtiendo su alineación preferencial, coexistiendo entre valores desordenados y una nueva organización.
- **co-D** : Cambios en *co-D* reflejarían una re-orientación en el alineamiento terciario, indicando que las células están reorganizando sus orientaciones terciarias. Esto podría sugerir un ajuste en la simetría tridimensional del material, donde vemos como se están alineando los ejes terciarios de los tensores con los ejes secundarios de los tensores de referencia.
- **co-C**: Un cambio en *co-C* indica una reconfiguración en la simetría dentro del plano secundario. Esto podría reflejar una transición hacia una estructura más desordenada en los planos secundarios del cristal líquido, pues ahora vemos un aumento en los valores cercanos a 0.

Tiempo 200.0

- **co-S**: Los valores parecen mantenerse similares, podemos observar un aumento de los valores que se acumulan alrededor de 0, aunque predominan las células alineadas perpendicularmente al eje principal, esto puede sugerir que va aumentando el desorden en el sistema.
- **co-P** : La distribución aumenta alrededor de 0, indicando un desorden en el alineamiento secundario de los ejes.
- **co-D** : Variaciones en *co-D* sugieren cambios dinámicos en la estructura terciaria del material, indicando una reorganización continua en los ejes terciarios.
- **co-C (Simetría en el Plano Secundario)**: Un cambio en *co-C* indica una reconfiguración en la simetría dentro del plano secundario. Esto podría reflejar una transición hacia una estructura más ordenada y ortogonal al plano, reflejado en el aumento de los valores negativos.

Tiempo 300.0

- **co-S**: Los valores parecen mantenerse similares con un nuevo aumento de los valores que se acumulan alrededor de 0, sugiriendo incremento en el desorden de los ejes principales.
- **co-P** : La distribución aumenta alrededor de 0, indicando un desorden en el alineamiento secundario de los ejes.
- **co-D** : Gran variabilidad en los valores de co-D, estos sugieren cambios dinámicos en la estructura terciaria del material, indicando una reorganización continua en los ejes terciarios.
- **co-C**: Un cambio en *co-C* indica una reconfiguración en la simetría dentro del plano secundario. Esto podría reflejar una transición hacia una estructura más ordenada y planar, reflejado en el aumento de los valores positivos.

Resumen General El análisis de los parámetros de orden co-orientacional revela que el tejido presenta características de un cristal líquido biaxial con variabilidad significativa en la alineación. La persistencia de valores negativos en CO-S indica una desalineación clara (o una alineación en la dirección opuesta típica de un estado altamente ordenado). CO-P y CO-D muestran que hay configuraciones biaxiales presentes, pero no de manera consistente en todo el tejido, sugiriendo una estructura interna compleja. CO-C muestra una simetría sin dirección preferida clara, reflejando una mezcla de regiones con diferentes grados de orden biaxial.

En conclusión, el material muestra una alta organización y alineación perpendicular en un comienzo. Con el tiempo, estos valores disminuyen y fluctúan, indicando una transición hacia un estado más desordenado. Finalmente, los valores bajos y fluctuantes sugieren una desorganización significativa y una falta de estabilidad en la estructura del material, similar a un cristal líquido que ha perdido su orden y ha alcanzado un estado isotrópico o altamente dinámico. Esto puede deberse a la aparición de múltiples defectos topológicos que deben analizarse con métodos más exhaustivos para obtener mejores conclusiones del comportamiento en estas fases finales donde hay gran presencia de características morfológicas.

6. Capítulo 5: Conclusiones y Discusión

En este capítulo, hemos presentado un enfoque exhaustivo para el pre-procesamiento y segmentación de datos tridimensionales de trayectorias celulares, basado en el análisis de los datos publicados por Shah et al. en 2019. Utilizando estos datos, hemos transformado las coordenadas espaciales de las células en un formato adecuado para análisis posteriores, aplicando técnicas de eliminación de ruido, centrado y alineación de datos, así como segmentación de imágenes tridimensionales. Este proceso ha permitido la creación de representaciones visuales semi-artificiales de las trayectorias y estructuras celulares, facilitando el estudio de patrones y ensamblajes emergentes en los datos.

Sin embargo, es fundamental reconocer las limitaciones y áreas de mejora en nuestro enfoque actual. Uno de los desafíos principales radica en la generación artificial de las formas de las membranas celulares. Debido a restricciones de tiempo y recursos, las membranas celulares fueron creadas artificialmente basándose únicamente en las posiciones reales de los núcleos celulares. Esta simplificación ha introducido ciertas distorsiones en los datos, generando formas celulares que pueden no reflejar completamente la realidad biológica. Las formas celulares artificiales pueden influir en los resultados de nuestro análisis, afectando la precisión y validez de las conclusiones derivadas.

Para mejorar la calidad y exactitud de este análisis, sería altamente beneficioso utilizar imágenes reales de las membranas celulares en lugar de representaciones artificiales. Las técnicas avanzadas de microscopía, como la microscopía de fluorescencia de alta resolución o la tomografía de rayos X, podrían proporcionar datos detallados y precisos sobre las estructuras de las membranas celulares. Al incorporar estas imágenes reales, podríamos reducir la incidencia de datos alterados y obtener formas celulares más naturales y representativas de la realidad biológica. Esto no solo mejoraría la precisión de nuestras representaciones visuales, sino que también proporcionaría una base más sólida para el análisis de las trayectorias y ensamblajes celulares.

Además, este análisis se puede extender para abordar cuestiones más complejas y fundamentales en la biología del desarrollo, como el estudio de defectos topológicos en tejidos celulares. Los defectos topológicos, que incluyen dislocaciones, defectos de borde y otros tipos de irregularidades en la estructura del tejido, pueden tener un impacto significativo en el comportamiento morfogenético de las células. Al rastrear y analizar estos defectos topológicos, podríamos ganar

una comprensión más profunda de cómo afectan a la organización y dinámica del tejido, y cómo contribuyen a procesos biológicos críticos como la diferenciación celular, la migración y la formación de patrones.

El análisis de defectos topológicos en el contexto de nuestro enfoque tensorial y de parámetros de orden co-orientacional podría revelar nuevas perspectivas sobre la mecánica celular y la organización del tejido. Por ejemplo, podríamos investigar cómo los defectos topológicos influyen en los parámetros de orden co-orientacional, y cómo estos cambios afectan a la estructura y funcionalidad del tejido. Esta línea de investigación podría llevarnos a identificar nuevas correlaciones entre la presencia de defectos topológicos y fenómenos biológicos específicos, proporcionando información valiosa para la biología del desarrollo y la medicina regenerativa.

En resumen, aunque nuestro enfoque actual ha proporcionado una base sólida para el análisis de trayectorias celulares y estructuras en el embrión de pez cebra, hay un potencial significativo para mejorar y expandir este trabajo mediante el uso de imágenes reales de membranas celulares y el estudio de defectos topológicos. Estas mejoras no solo aumentarían la precisión y relevancia de nuestros análisis, sino que también abrirían nuevas vías para investigar y comprender los complejos procesos que subyacen a la morfogénesis y la organización celular en tejidos biológicos. Conociendo la capacidad de los tejidos y las células para moverse y reorganizarse durante la embriogénesis del pez cebra, y centrando nuestro estudio en estos comportamientos complejos, hemos podido encapsular todos los fenómenos revisados en la parte teórica física de esta sección, en relación con su organización tisular.

Referencias

- [1] Doi, Masao. (2013) *Soft Matter Physics*, Oxford Academic. <https://doi.org/10.1093/acprof:oso/9780199652952.001.0001>
- [2] Gennes P-G, Prost J. (1995) *The Physics of Liquid Crystals* (2 Edition) Oxford: Clarendon Press.
- [3] Scholich, A. (2018). *Biaxial Nematic Order in Liver Tissue*. PhD thesis, TU Dresden
- [4] Scholich, A. et al. (2020) *Quantification of nematic cell polarity in three-dimensional tissues*, Plos Computational Biology. <https://doi.org/10.1371/journal.pcbi.1008412>

- [5] Geoffrey R. Luckhurst, Timothy J. Sluckin. (2015) *Biaxial Nematic Liquid Crystals: Theory, Simulation, and Experiment*, Wiley. <https://doi.org/10.1002/9781118696316>
- [6] Doostmohammadi, A., Ignés-Mullol, J., Yeomans, J.M. et al. (2018) *Active nematics*. Nat Commun. <https://doi.org/10.1038/s41467-018-05666-8>
- [7] Shankar, S., Souslov, A., Bowick, M.J. et al. (2022) *Topological active matter*. Nat Rev Phys. <https://doi.org/10.1038/s42254-022-00445-3>
- [8] Binysh, Jack and Kos, Žiga and Čopar, Simon and Ravnik, Miha and Alexander, Gareth P. (2020) *Three-Dimensional Active Defect Loops*, Phys. Rev. Lett <https://link.aps.org/doi/10.1103/PhysRevLett.124.088001>
- [9] Guillaume Duclos, Raymond Adkins, Debarghya Banerjee, Matthew S. E. Peterson , et al. (2020) *Topological structure and dynamics of three-dimensional active nematics*, Science. <https://doi.org/10.1126/science.aaz4547>
- [10] Lenne, PF., Trivedi, V. (2022). *Sculpting tissues by phase transitions*. Nat Commun. <https://doi.org/10.1038/s41467-022-28151-9>
- [11] Len Pismen. (2021). *Active Matter Within and Around Us*, Springer Cham.
- [12] B Wunderlich. (1999) *A classification of molecules, phases, and transitions as recognized by thermal analysis* Thermochimica Acta. [https://doi.org/10.1016/S0040-6031\(99\)00252-X](https://doi.org/10.1016/S0040-6031(99)00252-X).
- [13] Étienne Fodor, M. Cristina Marchetti. (2018) *The statistical physics of active matter: From self-catalytic colloids to living cells*, Physica A: Statistical Mechanics and its Applications. <https://doi.org/10.1016/j.physa.2017.12.137>.
- [14] Altieri, A. (2019) *The Jamming Transition*. In: *Jamming and Glass Transitions*. Springer Theses. Springer, Cham. https://doi.org/10.1007/978-3-030-23600-7_3
- [15] Bi D, Yang X, Marchetti MC, Manning ML. (2016) *Motility-driven glass and jamming transitions in biological tissues*. Phys Rev X. <https://doi.org/10.1103/physrevx.6.021011>

- [16] Linda Ravazzano. (2022) *Structure, dynamics and phase transitions of biological matter*, PhD Thesis. Università degli Studi di Milano.
- [17] Mongera, A., Rowghanian, P., Gustafson, H.J. et al. (2018) *A fluid-to-solid jamming transition underlies vertebrate body axis elongation*. Nature, (2018). <https://doi.org/10.1038/s41586-018-0479-2>
- [18] Saw, Thuan Beng & Xi, Wang & Ladoux, Benoit & Lim, C.T. *Biological Tissues as Active Nematic Liquid Crystals*. Advanced Materials. <https://doi.org/10.1002/adma.201802579>
- [19] Zemel A, Rehfeldt F, Brown AE, Discher DE, Safran SA. (2010) *Optimal matrix rigidity for stress fiber polarization in stem cells*. Nat Phys. <https://doi.org/10.1038/nphys1613>.
- [20] T.B. Saw, A. Doostmohammadi, V. Nier, L. Kocgozlu, S. Thampi, Y. Toyama, P. Marcq, C.T. Lim, J.M. Yeomans, B. Ladoux (2017) *Topological defects in epithelia govern cell death and extrusion*, Nature. <https://doi.org/10.1038/nature21718>
- [21] Y. Maroudas-Sacks, L. Garion, L. Shani-Zerbib, A. Livshits, E. Braun, K. Keren. (2021) *Topological defects in the nematic order of actin fibres as organization centres of Hydra morphogenesis*, Nat Phys. <https://doi.org/10.1038/s41567-020-01083-1>
- [22] Copenhagen, K., Alert, R., Wingreen, N.S. et al. (2021) *Topological defects promote layer formation in Myxococcus xanthus colonies*. Nat. Phys. <https://doi.org/10.1038/s41567-020-01056-4>
- [23] Chevalier, N.R., Ammouche, Y., Gomis, A. et al. (2021) *A neural crest cell isotropic-to-nematic phase transition in the developing mammalian gut*. Commun Biol. <https://doi.org/10.1038/s42003-021-02333-5>
- [24] Marelli F, Rurale G and Persani L.(2021) *From Endoderm to Progenitors: An Update on the Early Steps of Thyroid Morphogenesis in the Zebrafish*. Front. Endocrinol. <https://doi.org/10.3389/fendo.2021.664557>
- [25] Sarah E. Webb, Andrew L. Miller. (2006) *Ca²⁺ signaling and early embryonic patterning during the Blastula and Gastrula Periods of Zebrafish and Xenopus development*, Biochimica et Biophysica Acta (BBA), Molecular Cell Research. <https://doi.org/10.1016/j.bbamcr.2006.08.004>

- [26] Lakshmi Balasubramaniam, Rene-Marc Mege, Benoit Ladoux. (2022) *Active nematics across scales from cytoskeleton organization to tissue morphogenesis*, Current Opinion in Genetics and Development. <https://doi.org/10.1016/j.gde.2021.101897>
- [27] Rohde LA, Heisenberg CP. (2007) *Zebrafish gastrulation: cell movements, signals, and mechanisms*. Int Rev Cytol. [https://doi.org/10.1016/S0074-7696\(07\)61004-3](https://doi.org/10.1016/S0074-7696(07)61004-3)
- [28] , Krieg M., A. (2009) *Cell adhesion and cell mechanics during zebrafish development*. PhD thesis. , TU Dresden.
- [29] Paluch E, Heisenberg CP. (2009) *Biology and physics of cell shape changes in development*. Curr Biol. <https://doi.org/10.1016/j.cub.2009.07.029>
- [30] Schwarz, Ulrich S. and Safran, Samuel A. (2013) *Physics of adherent cells*, Rev. Mod. Phys. <https://link.aps.org/doi/10.1103/RevModPhys.85.1327>
- [31] Mayor, R., Etienne-Manneville, S. (2016) *The front and rear of collective cell migration*. Nat Rev Mol Cell Biol. <https://doi.org/10.1038/nrm.2015.14>
- [32] Maeda YT, Inose J, Matsuo MY, Iwaya S, Sano M.(2008) *Ordered Patterns of Cell Shape and Orientational Correlation during Spontaneous Cell Migration*. PLOS ONE. <https://doi.org/10.1371/journal.pone.0003734>
- [33] Lauffenburger DA, Horwitz AF. (1996) *Cell migration: a physically integrated molecular process*. Cell. [https://doi.org/10.1016/S0092-8674\(00\)81280-5](https://doi.org/10.1016/S0092-8674(00)81280-5)
- [34] Morita H, Grigolon S, Bock M, Krens SF, Salbreux G, Heisenberg CP. (2017) *The Physical Basis of Coordinated Tissue Spreading in Zebrafish Gastrulation*. Dev Cell. <https://doi.org/10.1016/j.devcel.2017.01.010>
- [35] Nicolas Romeo, Alasdair Hastewell, Alexander MietkeJörn Dunkel. (2021) *Learning developmental mode dynamics from single-cell trajectories* eLife. <https://doi.org/10.7554/eLife.68679>
- [36] Pinheiro, D., Kardos, R., Hannezo, É. et al. (2022). *Morphogen gradient orchestrates pattern-preserving tissue morphogenesis via motility-driven unjamming*. Nat. Phys. <https://doi.org/10.1038/s41567-022-01787-6>

- [37] Nicoletta I. Petridou, Bernat Corominas-Murtra, Carl-Philipp Heisenberg, Edouard Hannezo. (2021) *Rigidity percolation uncovers a structural basis for embryonic tissue phase transitions* Cell. <https://doi.org/10.1016/j.cell.2021.02.017>
- [38] Denk J, Frey E. (2020) *Pattern-induced local symmetry breaking in active-matter systems*. Proc Natl Acad Sci U S A. <https://doi.org/10.1073/pnas.2010302117>
- [39] Schmidt, U., Weigert, M., Broaddus, C., & Myers, G. (2018) *Cell Detection with Star-Convex Polygons*. In *Medical Image Computing and Computer Assisted Intervention*, Springer International Publishing. https://doi.org/10.1007/978-3-030-00934-2_30
- [40] Ershov, D., Phan, M.-S., Pylvänäinen, J. W., Rigaud, S. U., Le Blanc, L., Charles-Orszag, A., Tinevez, J.-Y. (2022) *TrackMate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines*. Nature Methods. doi:10.1038/s41592-022-01507-1
- [41] Ákos Orosz, Vasileios Angelidakis, Katalin Bagi. (2021) *Surface orientation tensor to predict preferred contact orientation and characterise the form of individual particles*, Science Direct <https://doi.org/10.1016/j.powtec.2021.08.054.1>
- [42] Herrera, L. J., Leng Kuang-dai, Yang Qiang. (2012) *Fabric Tensor Characterization of Tensor-Valued Directional Data: Solution, Accuracy, and Symmetrization* Journal of Applied Mathematics, <https://doi.org/10.1155/2012/516060>
- [43] Subía-Potosí, María Isabel. (2021) *Análisis cuantitativo y simulación de un modelo de organización de cristal líquido en tejido hepático humano*. Tesis, Escuela politécnica Nacional.
- [44] Kanatani, K. (1984) *Distribution of directional data and fabric tensors*. International Journal of Engineering Science. <https://doi.org/10.1155/2012/516060>
- [45] Shah, G., Thierbach, K., Schmid, B. et al. (2019) *Multi-scale imaging and analysis identify pan-embryo cell dynamics of germlayer formation in zebrafish*. Nat Commun <https://doi.org/10.1038/s41467-019-13625-0>

7. Apéndice

7.1. Código para pre-procesar los datos

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

# Import all necessary packages and libraries

#for calculations
import numpy as np
import scipy
import multiprocessing

# image processing
from skimage.io import imread
from skimage.measure import regionprops

#others
from tqdm import tqdm

#=====
# FUNCTIONS
#=====

class LabeledImage:

    def __init__(self, input_image):
        """
        Description:
            Initializes a LabeledImage object with the provided input image.

        Parameters:
            input_image (numpy.ndarray): The input image containing labeled regions.

        Returns:
            None
        """
        self.image = input_image.astype(np.uint16)
        self.labels = self.getLabels()
        self.nlabels = len(self.labels)

    def getLabels(self):
        """
        Description:
            Get unique labels from the input image, excluding the background label (0).

        Parameters:
            None

        Returns:
            numpy.ndarray: Sorted array of unique labels.
        """
        return np.sort(np.unique(self.image)[1:]) # The [1:] excludes 0 from the list (background)

    def custom_padding(self, dL):
        """
```

```

    Description:
        Applies custom padding to the input image.

    Parameters:
        dL (int): Padding value.

    Returns:
        None
    """
    self.image = np.pad(self.image, dL, mode='constant', constant_values=0)

def clean_up_labels(self, minVol, maxVol):
    """
    Description:
        Cleans up labels based on the specified minimum and maximum volumes.

    Parameters:
        minVol (int): Minimum volume threshold for a valid region.
        maxVol (int): Maximum volume threshold for a valid region.

    Returns:
        None
    """
    initCells = self.nlabels
    imgclean = np.zeros_like(self.image, dtype=self.image.dtype)
    imgclean.fill(0)

    print("self.image_0", np.sort(np.unique(self.image)[1:]))
    print("imgData_0", np.sort(np.unique(imgclean)[1:]))

    regions = regionprops(self.image)
    with tqdm(total=len(regions), desc="Re-Assigning IDs", unit="cells") as pbar:
        new_ID = self.image.dtype.type(1)
        for region in regions:
            if minVol <= region.area <= maxVol:
                imgclean[self.image == region.label] = new_ID
                new_ID += self.image.dtype.type(1)
            pbar.update(1)
    print("imgData", np.sort(np.unique(imgclean)[1:]))
    # Update instance values
    self.image = imgclean
    self.labels = self.getLabels()
    self.nlabels = len(self.labels)
    print("self.image_", np.sort(np.unique(self.image)[1:]))

    print(f'_{self.nlabels}_cells_extracted_from_initial_{initCells}_labels')

def get_bounding_boxes(self, dL, sortByVolume):
    """
    Description:
        Retrieves bounding boxes for labeled regions in the input image.

    Parameters:
        dL (int): Padding value to increase bounding box dimensions and avoid boundary errors.
        sortByVolume (bool): If True, sorts bounding from smallest to biggest volume.

    Returns:
        None
    """
    initCells = self.nlabels

```

```

regions = regionprops(self.image) #already labeled image with cells ids
bounding_boxes = []
labels = []
volume = []

for region in regions:
    minr, minc, minz, maxr, maxc, maxz = region.bbox #regionprops bbox is (minrow,mincolumn, maxrow, maxcolumn)
    #bounding boxes have to stay in input_image limits
    minr= max(0, minr-dL)
    minc= max(0, minc-dL)
    minz= max(0, minz-dL)
    maxr= min(self.image.shape[0]-1, maxr+dL)
    maxc= min(self.image.shape[1]-1, maxc+dL)
    maxz= min(self.image.shape[2]-1, maxz+dL)
    bounding_boxes.append((minr,minc, minz,maxr,maxc, maxz))
    labels.append(region.label)
    volume.append(region.area)

#Sort the indexes
if sortByVolume:
    sorted_indices = sorted(range(len(volume)), key = lambda i: volume[i])
    #Apply the new order of indexes for the other parameters
    bounding_boxes_sorted = [bounding_boxes[i] for i in sorted_indices]
    labels_sorted = [labels[i] for i in sorted_indices]
    volume_sorted = [volume[i] for i in sorted_indices]
    # Update instance values
    self.bounding_boxes = bounding_boxes_sorted
    self.labels = labels_sorted
    self.volumes = volume_sorted
    self.nlabels = len(self.labels)
else:
    # Update instance values
    self.bounding_boxes = bounding_boxes
    self.labels = labels
    self.volumes = volume
    self.nlabels = len(self.labels)

print(f'_{self.nlabels}_bounding_boxes_extracted_for_{initCells}_cells')

def get_subimages(self):
    """
    Description:
        Extracts subimages corresponding to each labeled region.
    Parameters:
        None
    Returns:
        List[numpy.ndarray]: List of subimages, one per labeled region.
    """
    ncells = self.nlabels

    # list of subimages – one image per cell
    subimages = [None]*ncells
    for idx, bbox in enumerate(self.bounding_boxes):
        minr,minc,minz,maxr,maxc,maxz = bbox
        subimage = self.image[minr:maxr,minc:maxc, minz:maxz]
        subimages[idx] = subimage

    print(f'_{len(subimages)}_subimages_extracted_for_{ncells}_cells')
    return subimages

#####PARALLEL PROCESSING
@staticmethod

```

```

def process_region(region, dL, image_shape):
    """
    Process a region to obtain bounding box, label, and volume.
    """
    minr, minc, minz, maxr, maxc, maxz = region.bbox
    minr = max(0, minr - dL)
    minc = max(0, minc - dL)
    minz = max(0, minz - dL)
    maxr = min(image_shape[0] - 1, maxr + dL)
    maxc = min(image_shape[1] - 1, maxc + dL)
    maxz = min(image_shape[2] - 1, maxz + dL)
    return (minr, minc, minz, maxr, maxc, maxz), region.label, region.area

def get_bounding_boxes_parallel(self, dL, sortByVolume):
    """
    Description:
        Retrieves bounding boxes for labeled regions in the input image using multiprocessing.

    Parameters:
        dL (int): Padding value to increase bounding box dimensions and avoid boundary errors.
        sortByVolume (bool): If True, sorts bounding from smallest to biggest volume.

    Returns:
        None
    """
    initCells = self.nlabels
    regions = regionprops(self.image)

    with multiprocessing.Pool() as pool:
        results = list(tqdm(pool.imap(LabeledImage.process_region, regions), total=len(regions), desc="Re-Assigning IDs", unit="cells"))

    # Sort the results if required
    if sortByVolume:
        results.sort(key=lambda x: x[2])

    # Extract bounding boxes, labels, and volumes
    self.bounding_boxes = [res[0] for res in results]
    self.labels = [res[1] for res in results]
    self.volumes = [res[2] for res in results]
    self.nlabels = len(self.labels)

    print(f'_{self.nlabels}_bounding_boxes_extracted_for_{initCells}_cells')

```

7.2. Código para calcular la descomposición en modos esféricos y los tensores nematicos

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

# Import all necessary packages and libraries

#for calculations
import numpy as np
import scipy
import cmath
import itertools
from scipy.integrate import simpson
from scipy.interpolate import griddata

```

```

#for 3D mesh creations
import trimesh
from skimage import measure

# image processing
import scipy.ndimage as ndi

#for multiprocessing
import multiprocessing as mp

#others
import pickle
import time
from tqdm import tqdm

# =====
# Class for processing single
# =====

# Classes for processing one cell

class ProcessingParameters:
    def __init__(self, stepsize, smooth_image_wnd, smooth_mesh_itt, neigh_search_dist, Nmodes, bins):
        """
        Description:
            Initializes object with specified processing parameters.

        Parameters:
            stepsize (float): Step size for marching cubes algorithm.
            smooth_image_wnd (int): Size for median smoothing of the cell mask.
            smooth_mesh_itt (int): Number of iterations for mesh smoothing.
            neigh_search_dist (float): Search distance for neighbor identification.
            Nmodes (int): Number of modes for spherical harmonics decomposition.
            bins (int): Number of bins for histogram computation.

        Returns:
            None
        """
        self.stepsize = stepsize
        self.smooth_image_wnd = smooth_image_wnd
        self.smooth_mesh_itt = smooth_mesh_itt
        self.neigh_search_dist = neigh_search_dist
        self.Nmodes = Nmodes
        self.bins = bins

class SingleCell:
    def __init__(self, image, cell_id, bounding_box):
        """
        Description:
            Initializes object with image, cell ID, and bounding box.

        Parameters:
            image (numpy.ndarray): 3D image of the cell.
            cell_id (int): Unique identifier for the cell.
            bounding_box (tuple): Bounding box coordinates (minr, minc, minz, maxr, maxc, maxz).

        Returns:
            None
        """

```

```

self.image = image
self.cell_id = cell_id
self.shift_origin = bounding_box[0:3]

def generate_mesh(self, processing_parameters):
    """
    Description:
        Generates a 3D mesh for the cell based on the provided processing parameters.

    Parameters:
        processing_parameters (ProcessingParameters): Object contain the processing parameters:
            stepsize (float): Step size for marching cubes algorithm.
            smooth_image_wnd (int): Size for median smoothing of the cell mask.
            smooth_mesh_itt (int): Number of iterations for mesh smoothing.
            neigh_search_dist (float): Search distance for neighbor identification.
            Nmodes (int): Number of modes for spherical harmonics decomposition.
            bins (int): Number of bins for histogram computation.

    Returns:
        None
    """
    if self.cell_id == 0:
        self.mesh = None

    # Create the cell mask
    cell_mask = (self.image == self.cell_id).astype(np.int8)
    # Fill holes in the 3D image
    cell_mask = ndi.binary_fill_holes(cell_mask)

    # Median smooth image
    if processing_parameters.smooth_image_wnd > 0:
        scipy.ndimage.median_filter(cell_mask, size=processing_parameters.smooth_image_wnd, output=cell_mask)

    # Generate mesh
    verts, faces, _, _ = measure.marching_cubes(cell_mask, allow_degenerate=False, step_size=processing_parameters.stepsize)
    del cell_mask
    self.mesh = trimesh.Trimesh(vertices=verts, faces=faces)

    if self.mesh.is_volume == False:
        self.fix_mesh_is_volume_False()

    # Smooth mesh
    if processing_parameters.smooth_mesh_itt > 0:
        self.mesh = trimesh.smoothing.filter_humphrey(self.mesh, alpha=0.1, beta=0.1, iterations=processing_parameters.smooth_mesh_itt, laplacian_ope

    if self.mesh.is_volume == False:
        self.fix_mesh_is_volume_False()

    # shift origin
    self.mesh.vertices += self.shift_origin

    if self.mesh.is_volume == False:
        print(f"!!!!!!!!!!!!!! Cell {self.cell_id} is not a Volume after trying fixing!!!!!!!!")

def initialize_cell_dict(self):
    """
    Description:
        Initializes a dictionary to store mesh properties and additional data.

    Parameters:

```

None

Returns:

"""

```
#Create a dictionary to save all the mesh properties
cell_dict = self.mesh.to_dict()
cell_dict['is_volume'] = self.mesh.is_volume
cell_dict['mass_properties'] = self.mesh.mass_properties
cell_dict['faces'] = self.mesh.faces
cell_dict['face_areas'] = self.mesh.area_faces.tolist()
cell_dict['triangle_centers'] = self.mesh.triangles_center
cell_dict['cellId'] = self.cell_id
cell_dict['center_mass_mesh'] = self.getCenterofMassMesh()
cell_dict['facets'] = self.mesh.facets

# Normals and save
cell_dict['face_normals'] = (1.0*(self.mesh.face_normals)).tolist()
# Save mesh shift
cell_dict['mesh_shift'] = self.shift_origin # initial position of cropped images

#Create and fill dictionaries with Nones for future data storage
cell_dict['contactIds'] = None
cell_dict['contactIds_Binary'] = None
cell_dict['neighbours_ids'] = None
cell_dict['weighted_areas'] = None

cell_dict['Distribution_Function'] = None
cell_dict['Distribution_Function_flat'] = None
cell_dict['Cos(theta)_Values'] = None
cell_dict['Phi_Values'] = None

cell_dict['Spherical_Decomposition'] = None
cell_dict['Normalized_Spherical_Decomposition'] = None
cell_dict['Sff_max'] = None
cell_dict['Sff_strenght'] = None

cell_dict['Polar_Vector'] = None
cell_dict['Second_Order_Tensor'] = None
cell_dict['eigenValues'] = None
cell_dict['eigenVectors'] = None
cell_dict['Compactness'] = None
cell_dict['Flakiness'] = None
cell_dict['Elongation'] = None

cell_dict['TrackId'] = None
cell_dict['Velocity'] = None
cell_dict['Average_Velocity'] = None

cell_dict['center_drawing'] = cell_dict['center_mass_mesh']
cell_dict['Name'] = None
cell_dict['Extra_info'] = None

self.cell_dict = cell_dict
```

```
def fix_mesh_is_volume_False(self):
```

```
    """
```

```
    Description:
```

```
        Fixes the mesh if its 'is_volume' attribute is False.
```

```
    Parameters:
```

```
        None
```

```

Returns:
    None
"""
IsVolume = self.mesh.is_volume

if IsVolume == False:
    trimesh.repair.fix_normals(self.mesh)
    IsVolume = self.mesh.is_volume

if IsVolume == False:
    trimesh.repair.fix_winding(self.mesh)
    IsVolume = self.mesh.is_volume
    if IsVolume == True:
        print("cell_is_volume_after_fix_winding")

if IsVolume == False:
    trimesh.repair.fix_inversion(self.mesh)
    IsVolume = self.mesh.is_volume
    if IsVolume == True:
        print("cell_is_volume_after_fix_inversion")

if IsVolume == False:
    trimesh.repair.fill_holes(self.mesh)
    IsVolume = self.mesh.is_volume
    if IsVolume == True:
        print("cell_is_volume_after_fill_holes")

if IsVolume == False:
    trimesh.repair.fill_holes(self.mesh)
    IsVolume = self.mesh.is_volume
    if IsVolume == True:
        print("cell_is_volume_after_fix_normals")

if IsVolume == False:
    trimesh.repair.fix_normals(self.mesh)
    IsVolume = self.mesh.is_volume
    if IsVolume == True:
        print("cell_is_volume_after_fix_normals2")

def getCenterofMassMesh(self):
    """
    Description:
        Calculates the center of mass of the mesh.

    Parameters:
        None

    Returns:
        numpy.ndarray: 3D coordinates of the center of mass.
    """

    # Get data
    weighted_area = np.array(self.mesh.area_faces.tolist())
    triangle_centers = np.array(self.mesh.triangles_center)

    # Calculate center of Mesh with weights
    facesNo = len(weighted_area)
    center = np.zeros(3)
    for k in range(0, facesNo):
        center = center + weighted_area[k] * triangle_centers[k, :]

    center = center / np.sum(weighted_area) # normalize to the total surface area

```

```

    return center

#-----
#Assign contact IDs process
#-----

def getContactIds(self, search_dist):
    """
    Calculates .....

    Parameters:
        cell_dict (dict): Dictionary containing mesh properties and additional data.
        search_dist (float): Search distance for identifying contacts.

    Returns:
        numpy.ndarray: Array containing contact IDs for each face.
    """
    # Extract relevant data from cell_dict
    shift_origin = self.cell_dict['mesh_shift']
    center_mesh = np.round(np.array(self.cell_dict['mass_properties']['center_mass']), decimals=0).astype(int) - shift_origin
    img_data = self.image

    # Verify if the center mass is within the image bounds
    if (0 <= center_mesh[0] < img_data.shape[0] and
        0 <= center_mesh[1] < img_data.shape[1] and
        0 <= center_mesh[2] < img_data.shape[2]):

        verts = np.array(self.cell_dict['vertices']) - shift_origin
        faces = np.array(self.cell_dict['faces'])
        face_normals = np.array(self.cell_dict['face_normals'])

        # Initialize vector with default value 0
        contact_ids = np.zeros((face_normals.shape[0]), dtype=int)

        # Calculate contacts
        for idx, (face, normal) in enumerate(zip(faces, face_normals)):
            # Centroid of the triangle
            centroid = np.mean(verts[face], axis=0)

            # Point of contact
            contact_point = np.round(centroid + search_dist * normal).astype(int)

            # Verify if the contacts are in the limits of the image
            if (0 <= contact_point[0] < img_data.shape[0] and
                0 <= contact_point[1] < img_data.shape[1] and
                0 <= contact_point[2] < img_data.shape[2]):
                contact_ids[idx] = img_data[tuple(contact_point)]
            else:
                raise ValueError(f"Contact_point_{contact_point} is outside the image bounds")

        return contact_ids
    else:
        raise ValueError(f"Center_mass_{center_mesh} is outside the image bounds for cell_{cell_dict['cellId']}")

def assign_contact_ids(self, processing_parameters):
    # Evaluate number of cells
    img_data = self.image
    img = img_data[img_data > 0]
    nCells = len(np.unique(img))

```

```

# Assign contactIds
if nCells > 1:
    self.cell_dict['contactIds'] = self.getContactIds(search_dist=processing_parameters.neigh_search_dist)
else:
    # Assigns 1 to all faces
    face_normals = np.array(self.cell_dict['face_normals'])
    self.cell_dict['contactIds'] = np.ones((face_normals.shape[0],), dtype=np.int8)

# Assign neighbours ids
cells_contact_ids = self.cell_dict['contactIds']
cells_contact_ids = cells_contact_ids[cells_contact_ids > 0] # Only neighbouring cells
self.cell_dict['neighbours_ids'] = np.unique(cells_contact_ids)

if not self.cell_dict['neighbours_ids'].any():
    cellId = self.cell_dict['cellId']
    neigh = self.cell_dict['neighbours_ids']
    print(f'No neighbours were identified for cell {cellId}: {neigh}')

# Assign contactIds binary
self.cell_dict['contactIds_Binary'] = self.cell_dict['contactIds'].copy()
self.cell_dict['contactIds_Binary'][self.cell_dict['contactIds_Binary'] > 0] = 1

return self.cell_dict

```

```

#-----
#Spherical decomposition process
#-----

```

```

@staticmethod
def spherical_simp(phi, Costheta, distribution_function):
    """
    Calculates the integral of a distribution_function over an homogeneous costheta-phi spherical grid

    Parameters:
    - Costheta: 1D array, the sampling points for cos(theta) (cosine of polar angle).
    - phi: 1D array, the sampling points for phi (azimuthal angles).
    - distribution_function: The matrix to integrate (numpy.ndarray).

    Returns:
    Integral value

    """
    # Integral calculation
    integral = simpson(simpson(distribution_function, Costheta), phi)
    return integral

```

```

@staticmethod
def normalize_hist(phi, Costheta, distribution_function):
    """
    Does the normalization of a distribution_function

    Parameters:
    - phi: 1D array, the sampling points for phi (azimuthal angles).
    - Costheta: 1D array, the sampling points for cos(theta) (cosine of polar angle).
    - distribution_function: The matrix to integrate (numpy.ndarray).

    Returns:
    The normalized distribution
    """

```

```

"""
Area = SingleCell.spherical_simp(phi, Costheta, distribution_function)
distribution_normalized=distribution_function/Area
return distribution_normalized

@staticmethod
def spherical_projection(face_normals):
    """
    Calculates the spherical coordinates costheta and phi of vectors

    Parameters:
    - face_normals: 3D array

    Returns:
    Costheta and Phi coordinates of the vectors.

    """
    # Spherical projection of the face normals
    costheta_data = face_normals[:,2:3] # polar angle (0 <= theta <= pi)
    phi_data = np.arctan2(face_normals[:,1:2], np.array(face_normals[:,0:1])) # azimuthal angle (-pi <= phi <= pi)
    phi_data = np.where(phi_data < 0, phi_data + 2*np.pi, phi_data) # change negative phi values to their positive equivalent
    return costheta_data, phi_data

@staticmethod
def distribution_function(costheta_data, phi_data, weight, bins):
    """
    Calculates the distribution function associated to costheta and phi projections of the vectors.

    Parameters:
    - costheta_data: 1D array, output of spherical_projection.
    - phi_data: spherical_projection
    - weight: weight associated to every bin
    - bins: number of bins

    Returns:
    distribution function, costheta and phi values

    """
    #Get the distribution function associated to the projection of the normals
    H, xedges, yedges = np.histogram2d(costheta_data.ravel(), phi_data.ravel(), bins=bins, range=[[-1, 1], [0, 2*np.pi]], weights=weight)

    #get centers of the bins
    dt = (xedges[1]-xedges[0])/2
    costheta = xedges[0:bins]+ dt
    dp = (yedges[1]-yedges[0])/2
    phi = yedges[0:bins]+ dp

    return H, costheta, phi

@staticmethod
def get_flm(distribution_normalized, l, m, Costheta, theta, phi):
    """
    Calculates the  $f_l^m$  coefficients for the Spherical power spectrum analysis

    Parameters:
    - distribution_normalized: The matrix normalized (numpy.ndarray).
    - l: int, value of the mode (0,1,2,3...)
    - m: int, goes between -l,+l.
    - Costheta: 1D array, the sampling points for cos(theta) (cosine of polar angle).
    - phi: 1D array, the sampling points for phi (azimuthal angles).

    Returns:

```

```

    Sf_l: f_l^m

"""

integrand = np.ndarray(shape = (len(distribution_normalized), len(distribution_normalized)), dtype=complex)
for i in range(len(distribution_normalized)):
    for j in range(len(distribution_normalized)):
        integrand[i][j]= distribution_normalized[i][j] * np.conj(scipy.special.sph_harm(m, l, phi[j], theta[i]))
flm=SingleCell.spherical_simp(phi, Costheta, integrand)
return flm

@staticmethod
def gen2D_distribution(num_points, distribution_type, distribution_width):
    """
    Creates different ideal distributions over a homogeneous spherical grid.

    Parameters:
    - num_points: number of points of the grid (int)
    - distribution_type: string, 'homogeneous', 'north_pole', 'south_pole', 'two_poles', 'horizontal_ring', 'vertical_ring',
    'vertical_two_poles', 'vertical_one_pole'.
    - distribution_width: float, between 0 and 1, percentage of the area that the distribution has.

    Returns:
    The distribution function

    """
    # Define a homogeneous grid over a sphere
    Costheta_i = np.linspace(-1, 1, num_points+1)
    phi_i = np.linspace(0, 2*np.pi, num_points+1) # Azimuthal angle [0,2pi]

    #Correct the grid points
    dt = (Costheta_i[1]-Costheta_i[0])/2
    Costheta= Costheta_i[0:num_points]+ dt
    dp = (phi_i[1]-phi_i[0])/2
    phi = phi_i[0:num_points]+ dp

    # Define the distribution function based on the distribution type
    if distribution_type == 'homogeneous':
        distribution_function = np.ones((num_points, num_points))

    return distribution_function, Costheta, phi

@staticmethod
def spherical_Analysis(phi, Costheta, distribution_function, l, normalize_histogram):
    """
    Calculates the Spherical Analysis of a distribution function

    Parameters:
    - phi: 1D array, the sampling points for phi (azimuthal angles).
    - Costheta: 1D array, the sampling points for cos(theta) (cosine of polar angle).
    - distribution_function: The matrix to do the spherical analysis(numpy.ndarray).
    - l: int, value of the mode (0,1,2,3...)
    - normalize_histogram: bool, wants the histogram to be normalized

    Returns:
    Sf_l: Mode l of the distribution function

    """
    if normalize_histogram == True :
        distribution_normalized = SingleCell.normalize_hist(phi, Costheta, distribution_function)
    else:
        distribution_normalized = SingleCell.distribution_function

```

```

theta = np.arccos(Costheta)
Sf=0.0
for m in range(-1, l+1):
    fml = SingleCell.get_flm(distribution_normalized, l, m, Costheta, theta, phi)
    Sf += (1/(4*np.pi))*((np.absolute(fml))**2)
return Sf

def spherical_decomposition(self, processing_parameters):
    # Extract data from cell_dict
    face_normals = np.array(self.cell_dict['face_normals'])
    face_areas = np.array(self.cell_dict['face_areas'])
    contacts = self.cell_dict['contactIds_Binary']
    weighted_area = np.multiply(face_areas, contacts)
    modes = processing_parameters.Nmodes
    bins = processing_parameters.bins

    # Calculate spherical projection
    costheta_i, phi_i = SingleCell.spherical_projection(face_normals) # Llama al método estático con la clase SingleCell
    distribution_cell, costheta_f, phi_f = SingleCell.distribution_function(costheta_i, phi_i, weighted_area, bins= processing_parameters.bins)
# Llama al método estático con la clase SingleCell
    self.cell_dict['Distribution_Function'] = distribution_cell
    self.cell_dict['Cos(theta)_Values'] = costheta_f
    self.cell_dict['Phi_Values'] = phi_f

    # Calculate values for homogeneous distribution using bins and modes for the actual distribution so we can normalize it
    distribution_flat, costheta, phi = SingleCell.gen2D_distribution(bins, 'homogeneous', 0) # Asumiendo que gen2D_distribution es una función de
    self.cell_dict['Distribution_Function_flat'] = distribution_flat

    # Calcula la descomposición esférica para los modos requeridos
    F_values = np.zeros(modes)
    F_flat = np.zeros(modes)
    for l in range(modes):
        F_values[l] = SingleCell.spherical_Analysis(phi_f, costheta_f, distribution_cell, l, True) # Llama al método estático con la clase SingleCell
        F_flat[l] = SingleCell.spherical_Analysis(phi, costheta, distribution_flat, l, True) # Llama al método estático con la clase SingleCell

    self.cell_dict['Spherical_Decomposition'] = F_values

    # Calcula la descomposición esférica normalizada
    # Normaliza por el primer modo de la distribución plana
    F0 = F_flat[0]
    for l in range(modes):
        F_values[l] /= F0
        F_flat[l] /= F0

    F_norm = F_values - F_flat
    self.cell_dict['Normalized_Spherical_Decomposition'] = F_norm

    # Obtiene el ID del modo más grande y la fuerza de los valores normalizados
    max_modes = np.argmax(F_norm)
    self.cell_dict['Sff_max'] = max_modes

    # Calcula la 'fuerza del modo': Sff_max de los valores normalizados
    sorted_F = np.sort(F_norm, kind='quicksort')[::-1]
    self.cell_dict['Sff_strenght'] = sorted_F[0]

    return self.cell_dict

#-----
#Anistropy descriptors
#-----

```

```

def first_order_vector(self):
    face_normals = np.array(self.cell_dict['face_normals'])
    face_areas = np.array(self.cell_dict['face_areas'])
    contacts = self.cell_dict['contactIds_Binary']
    weighted_area = np.multiply(face_areas, contacts)

    total_weighted_area = np.sum(weighted_area)

    # Check if the total weighted area is greater than zero
    if total_weighted_area > 0:
        # Calculate polar vector
        faces_no = len(weighted_area)
        V = np.zeros(3)
        for k in range(0, faces_no):
            V = V + weighted_area[k] * face_normals[k, :]
        V = V / np.sum(weighted_area) # normalize to the total surface area
        return V

def second_order_tensor(self):
    face_normals = np.array(self.cell_dict['face_normals'])
    face_areas = np.array(self.cell_dict['face_areas'])
    contacts = self.cell_dict['contactIds_Binary']
    weighted_area = np.multiply(face_areas, contacts)

    total_weighted_area = np.sum(weighted_area)

    # Check if the total weighted area is greater than zero
    if total_weighted_area > 0:
        faces_no = len(weighted_area)
        F = np.zeros((3, 3))

        for k in range(0, faces_no):
            F = F + weighted_area[k] * np.tensordot(face_normals[k, :], face_normals[k, :], axes=0) # outer product
        F = F / np.sum(weighted_area) # normalize to the total surface area
        return F

@staticmethod
def FTensorAnalysis(F):
    # Eigenvalues
    eigen_values, vectors = np.linalg.eig(F) # eigen analysis
    index = np.argsort(eigen_values)[::-1] # sort eigenvalues in descending order
    eigen_values = eigen_values[index]

    f1 = eigen_values[0] # Largest eigenvalue
    f2 = eigen_values[1] # Intermediate eigenvalue
    f3 = eigen_values[2] # Smallest eigenvalue

    # Eigenvectors
    v1 = vectors[:, index[0]] # First eigenvector
    v2 = vectors[:, index[1]] # Second eigenvector
    v3 = vectors[:, index[2]] # Third eigenvector

    eigen_vectors = {
        "v1": v1,
        "v2": v2,
        "v3": v3}

    C = f3 / f1 # Compactness
    F = (f1 - f2) / f1 # Flakiness
    E = (f2 - f3) / f1 # Elongation

    return C, F, E, eigen_values, eigen_vectors

```

```

def get_anisotropy_descriptors(self):

    # Get first order descriptor
    PolarVector = self.first_order_vector()
    self.cell_dict['Polar_Vector'] = PolarVector

    # Get second order descriptor
    Ftensor = self.second_order_tensor()
    C, F, E, eigenValues, eigenVectors = SingleCell.FTensorAnalysis(Ftensor)
    self.cell_dict['Second_Order_Tensor'] = Ftensor
    self.cell_dict['Compactness'] = C
    self.cell_dict['Flakiness'] = F
    self.cell_dict['Elongation'] = E
    self.cell_dict['eigenValues'] = eigenValues
    self.cell_dict['eigenVectors'] = eigenVectors

    return self.cell_dict

=====
# Class for processing all cells
=====

class CellsProcessor:
    def __init__(self, subimages, bounding_boxes, labels, processing_parameters, num_cores, output_path):
        """
        Description:
            Initializes object with input data and parameters.

        Parameters:
            subimages (List[numpy.ndarray]): List of subimages, one per labeled region.
            bounding_boxes (List[tuple]): List of bounding boxes for each labeled region.
            labels (List[int]): List of labels corresponding to each cell.
            processing_parameters (ProcessingParameters): Object containing the following parameters:
                stepsize (float): Step size for marching cubes algorithm.
                smooth_image_wnd (int): Size for median smoothing of the cell mask.
                smooth_mesh_itt (int): Number of iterations for mesh smoothing.
                neigh_search_dist (float): Search distance for neighbor identification.
                Nmodes (int): Number of modes for spherical harmonics decomposition.
                bins (int): Number of bins for histogram computation.

            num_cores (int): Number of CPU cores to use for parallel processing.
            output_path (str): Path to save the processed data.

        Returns:
            None
        """
        self.subimages = subimages
        self.bounding_boxes = bounding_boxes
        self.labels = labels
        self.num_cores = num_cores
        self.output_path = output_path
        self.ncells = len(subimages)
        self.processing_parameters = processing_parameters

    def run_processing(self):
        """
        Description:
            Runs the cell processing workflow, either in parallel or serial mode.

        Parameters:
            None

```

```

Returns:
    None
"""
# Process data
start_time = time.time()
if self.num_cores > 1:
    cells_data = self.run_parallel_processing()
else:
    cells_data = self.run_serial_processing()

elapsed_time = time.time() - start_time

# Save output
init_cells = len(cells_data)
with open(self.output_path, "wb") as file:
    pickle.dump(cells_data, file, protocol=pickle.HIGHEST_PROTOCOL)

print(f'Process done!')
print(f"Elapsed Time: {elapsed_time:.4f} seconds")
print(f'[{init_cells} cells saved in {self.output_path}']

def run_parallel_processing(self):
    """
    Description:
        Runs the cell processing workflow in parallel.

    Parameters:
        None

    Returns:
        List[dict]: List of dictionaries containing processed cell data.

    """
    cells_data = [None] * self.ncells
    pool = mp.Pool(processes=self.num_cores)
    input_data = [(self.subimages[idx], self.labels[idx], self.bounding_boxes[idx], self.processing_parameters) for idx, id_value in enumerate(self.subimages)]

    total_tasks = len(input_data)
    progress_bar = tqdm(total=total_tasks, desc="Processing", unit="cells")
    results = []

    for input_args in input_data:
        result = pool.apply_async(self.process_cell, args=input_args, callback=lambda x: self.update_progress(progress_bar))
        results.append(result)

    pool.close()
    pool.join()

    for result in results:
        result_value = result.get()
        if result_value is not None: # Check if the result is valid
            cell_ID = int(result_value['cellID'])
            cells_data[cell_ID - 1] = result_value
        else:
            print(f"Skipping cell due to processing error.")

    # Filter out None values from cells_data
    cells_data = [cell for cell in cells_data if cell is not None]

    return cells_data

```

```

def update_progress(self, progress_bar):
    """
    Description:
        Callback function to update the progress bar during parallel processing.

    Parameters:
        progress_bar (tqdm.tqdm): Progress bar object.

    Returns:
        None
    """
    progress_bar.update()

def run_serial_processing(self):
    """
    Description:
        Runs the cell processing workflow in serial.

    Parameters:
        None

    Returns:
        List[dict]: List of dictionaries containing processed cell data.
    """
    cells_data = [] # Lista vacía para almacenar los datos de las celdas procesadas
    progress_bar = tqdm(total=self.ncells, desc="Processing", unit="cells")

    for idx, id_value in enumerate(self.labels):
        result = self.process_cell(self.subimages[idx], self.labels[idx], self.bounding_boxes[idx], self.processing_parameters)
        if result is not None: # Verificar si el resultado es None
            cell_ID = int(result['cellID'])
            cells_data.append((cell_ID, result)) # Almacenar una tupla (cellID, resultado)
            progress_bar.update(1)

    progress_bar.close()

    # Ordenar los datos por cellID antes de retornarlos
    cells_data.sort(key=lambda x: x[0])
    sorted_cells_data = [data[1] for data in cells_data]

    return sorted_cells_data

@staticmethod
def process_cell(image, cell_id, bounding_box, processing_parameters):
    """
    Description:
        Process a single cell, including mesh generation and data extraction.

    Parameters:
        image (numpy.ndarray): 3D image of the cell.
        cell_id (int): Unique identifier for the cell.
        bounding_box (tuple): Bounding box coordinates (minr, minc, minz, maxr, maxc, maxz).
        processing_parameters (ProcessingParameters): Object containing processing parameters.

    Returns:
        dict: Dictionary containing processed cell data.
    """
    try:

```

```

cell = SingleCell(image, cell_id, bounding_box) # Create SingleCell objet
cell.generate_mesh(processing_parameters) # Generate mesh from an image
cell.cell_dict = {}
cell.initialize_cell_dict() # Initialize dictionary from mesh

# Get cell contacts
cell.assign_contact_ids(processing_parameters)

# Spherical decomposition
cell.spherical_decomposition(processing_parameters)

# Get anisotropy descriptors
cell.get_anisotropy_descriptors()

# Get extra data
cell.cell_dict['Name'] = f'Cell_{cell.cell_id}'

return cell.cell_dict
except Exception as e:
    print(f"Error_{processing}_{cell}_{cell_id}:_{str(e)}")
    return None # Return None if an error occurs, indica

```

7.3. Código para calcular los parámetros de orden

```

import pickle
import numpy as np
from tqdm import tqdm

#Input
inputFile = 'input_path'
# Import data
with open(inputFile, "rb") as pickle_file:
    cellsdata = pickle.load(pickle_file)

# Extraer los datos necesarios de las células
individual_tensors = [cell['Second_Order_Tensor'] for cell in cellsdata]
cell_positions = [cell['center_mass_mesh'] for cell in cellsdata]
eigenvectors_tensor= [cell['eigenVectors'] for cell in cellsdata]

# Importa la característica "cellIds" junto con otras características
cell_ids = [cell['cellId'] for cell in cellsdata]

# Crea una lista de tuplas que contenga el tensor individual y su ID de célula correspondiente
individual_tensors_with_ids = [(tensor, cell_id) for tensor, cell_id in zip(individual_tensors, cell_ids)]

def spherical_distance_cartesian(x1, y1, z1, x2, y2, z2):
    """
    Calcula la distancia esférica usando coordenadas cartesianas en 3D.
    """
    r1 = np.sqrt(x1**2 + y1**2 + z1**2)
    r2 = np.sqrt(x2**2 + y2**2 + z2**2)
    cos_omega = (x1 * x2 + y1 * y2 + z1 * z2) / (r1 * r2)
    cos_omega = np.clip(cos_omega, -1.0, 1.0) # Asegurar que cos_omega esté en el rango [-1, 1]
    omega = np.arccos(cos_omega)
    return omega

# Calcular las distancias esféricas entre las posiciones de las células
distances = []
for i in range(len(cell_positions)):
    for j in range(i + 1, len(cell_positions)):

```

```

    x1, y1, z1 = cell_positions[i]
    x2, y2, z2 = cell_positions[j]
    distance = spherical_distance_cartesian(x1, y1, z1, x2, y2, z2)
    distances.append(distance)

# Calcular el promedio de las distancias esféricas
average_distance = np.mean(distances)
k = average_distance

def f_k(cos_omega, k):
    return (k + 1) / (2 * np.pi) * (cos_omega ** (k + 1)) * (cos_omega > 0)

def g_k(cos_omega, k):
    return k / (2 * np.pi) * (cos_omega ** (k - 1)) * (cos_omega > 0)

#calculo cos(omega)
def spherical_distance_cartesian(x1, y1, z1, x2, y2, z2):
    r1 = np.sqrt(x1**2 + y1**2 + z1**2)
    r2 = np.sqrt(x2**2 + y2**2 + z2**2)
    cos_omega = (x1 * x2 + y1 * y2 + z1 * z2) / (r1 * r2)
    return cos_omega

# Inicializar una lista para los tensores coarse grained junto con sus IDs de célula correspondientes
coarse_grained_tensors = []

# Actualiza el cálculo de la distancia en el bucle principal
for i, (tensor, cell_id) in enumerate(tqdm(individual_tensors_with_ids, desc="Calculando tensores coarse grained")):
    cumulative_tensor = np.zeros_like(tensor)
    sum_of_weights = 0

    for j, (neighbor_tensor, neighbor_id) in enumerate(individual_tensors_with_ids):
        if i != j: # Excluir la célula actual
            x1, y1, z1 = cell_positions[i]
            x2, y2, z2 = cell_positions[j]
            cos_omega = spherical_distance_cartesian(x1, y1, z1, x2, y2, z2)
            weight = f_k(cos_omega, k)
            cumulative_tensor += weight * neighbor_tensor
            sum_of_weights += weight

    # Calcular el tensor promedio en la vecindad sin verificar la suma de pesos
    coarse_grained_tensor = cumulative_tensor / sum_of_weights
    coarse_grained_tensors.append((coarse_grained_tensor, cell_id))

def FTensorAnalysis(F):
    # Eigenvalues and eigenvectors
    eigen_values, vectors = np.linalg.eig(F) # Eigen analysis

    # Sort eigenvalues and eigenvectors
    index = np.argsort(eigen_values)[::-1] # Sort eigenvalues in descending order
    eigen_values = eigen_values[index]
    vectors = vectors[:, index]

    # Extract eigenvectors
    v1 = vectors[:, 0] # First eigenvector
    v2 = vectors[:, 1] # Second eigenvector
    v3 = vectors[:, 2] # Third eigenvector

    eigen_vectors = {
        "v1": v1,
        "v2": v2,

```

```

        "v3": v3
    }

    return eigen_vectors

eigen_vectors_ref= []
for i in range(len(coarse_grained_tensors)):
    eigen_vectors_ref.append(FTensorAnalysis(coarse_grained_tensors[i][0]))

# Función para calcular los parámetros COOPS
def calculate_COOPS(eigenvectors_tensor, eigenvectors_reference):

    EigenVectors= eigenvectors_tensor
    EigenVectors_ref= eigenvectors_reference

    # Autovectores del tensor individual
    n = EigenVectors['v1'] # First eigenvector
    m = EigenVectors['v2'] # Second eigenvector
    l = EigenVectors['v3'] # Third eigenvector

    # Autovectores del tensor coarse grained
    u = EigenVectors_ref['v1'] # First eigenvector
    v = EigenVectors_ref['v2'] # Second eigenvector
    w = EigenVectors_ref['v3'] # Third eigenvector

    # Cálculo de los parámetros COOPS
    co_S = 0.5 * np.mean(3 * (n * w)**2 - 1)
    co_P = 1.5 * np.mean((n * u)**2 - (n * v)**2)
    co_D = 1.5 * np.mean((l * w)**2 - (m * w)**2)
    co_C = 0.5 * np.mean((l * u)**2 - (l * v)**2 + (m * v)**2 - (m * u)**2)

    return co_S, co_P, co_D, co_C

co_S_values = []
co_P_values = []
co_D_values = []
co_C_values = []
# Calcular COOPS
for i in range(len(individual_tensors)):
    co_S, co_P, co_D, co_C= calculate_COOPS(eigenvectors_tensor[i], eigen_vectors_ref[i])
    co_S_values.append(co_S)
    co_P_values.append(co_P)
    co_D_values.append(co_D)
    co_C_values.append(co_C)

```