



PRÁCTICA

REPORTE IV

PROYECTO: MEDICIÓN DE LAS FUERZAS EN
UNA GOTA DEFORMADA POR CÉLULAS.



Camino a seguir



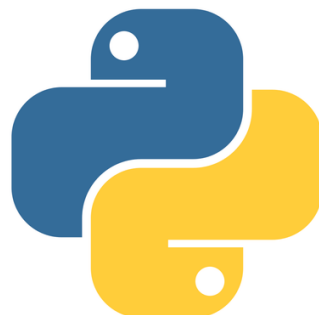
IDL

Reconstrucción 3D
de la gota. Se exporta un
archivo .OFF



MALLA TRIANGULAR

Se obtiene un conjunto de
caras y vértices. Se puede
trabajar en otro lenguaje.



PYTHON

Se busca implementar las
curvaturas y la interfaz en
Python.

Reconstrucción en IDL

Ajuste de parámetros

Aún se necesita buscar una correcta elasticidad, rigidez, viscosidad, fuerza externa, entre otros parámetros.

Relativamente sencilla

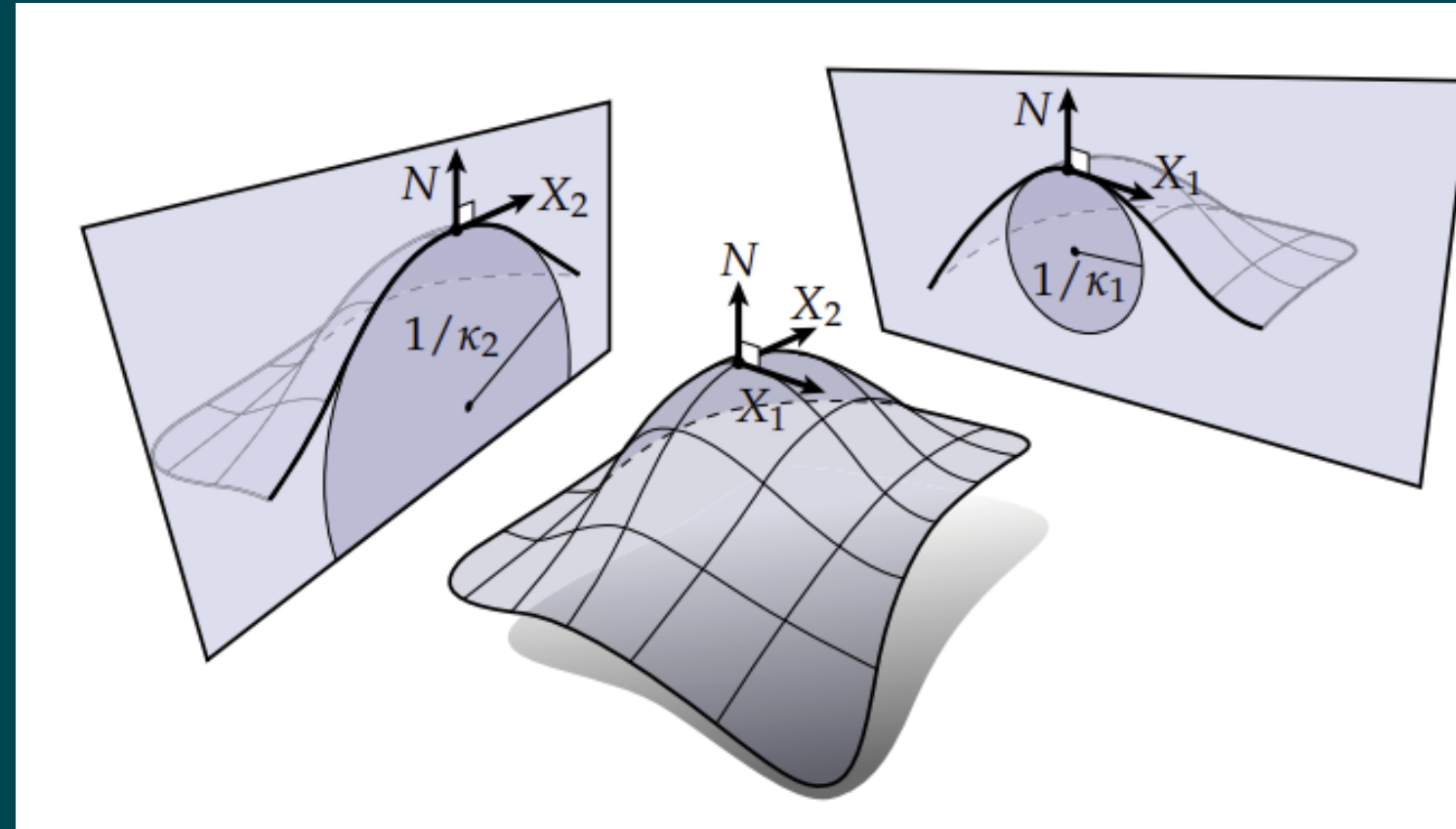
No se necesitan muchos conocimientos de programación y la exportación se hace automática.

Calidad y efectividad

El resultado obtenido es de muy buena calidad y está en un formato muy versátil, el cual tiene mucha literatura y métodos implementados

Geometría Diferencial

CURVATURAS PRINCIPALES



No depende de la parametrización

CURVATURA PROMEDIO

$$H = \frac{\kappa_1 + \kappa_2}{2}$$



ESTRÉS ANISOTRÓPICO

$$\delta\sigma_{nn}(\theta, \phi) = 2\gamma \left(H(\theta, \phi) - \frac{1}{R_o} \right)$$

Método Discreto

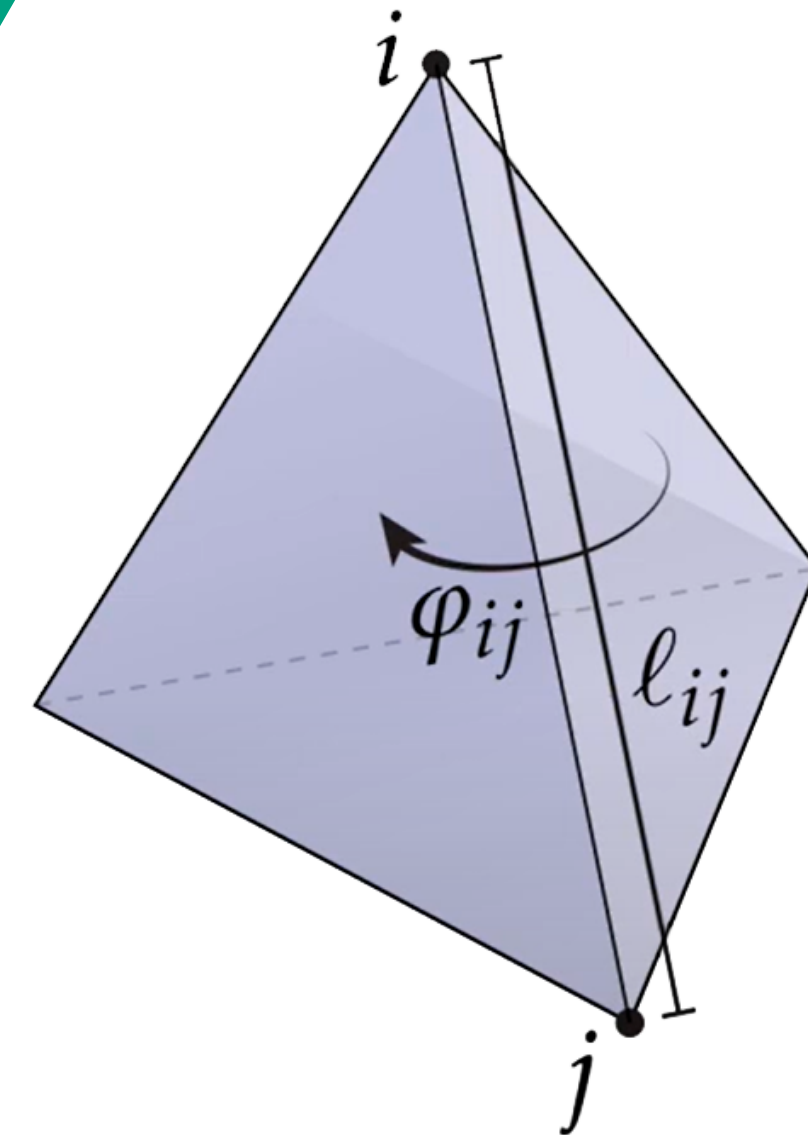
ÁNGULO ENTRE CARAS

El angulo formado entre dos caras puede determinar la cantidad de curvatura.

LARGO DEL LADO

Ponderador que puede determinar cuánto pesa ese ángulo

$$H_i = \frac{1}{4} \sum_{j \in \mathcal{N}(i)} \ell_{ij} \varphi_{ij}$$



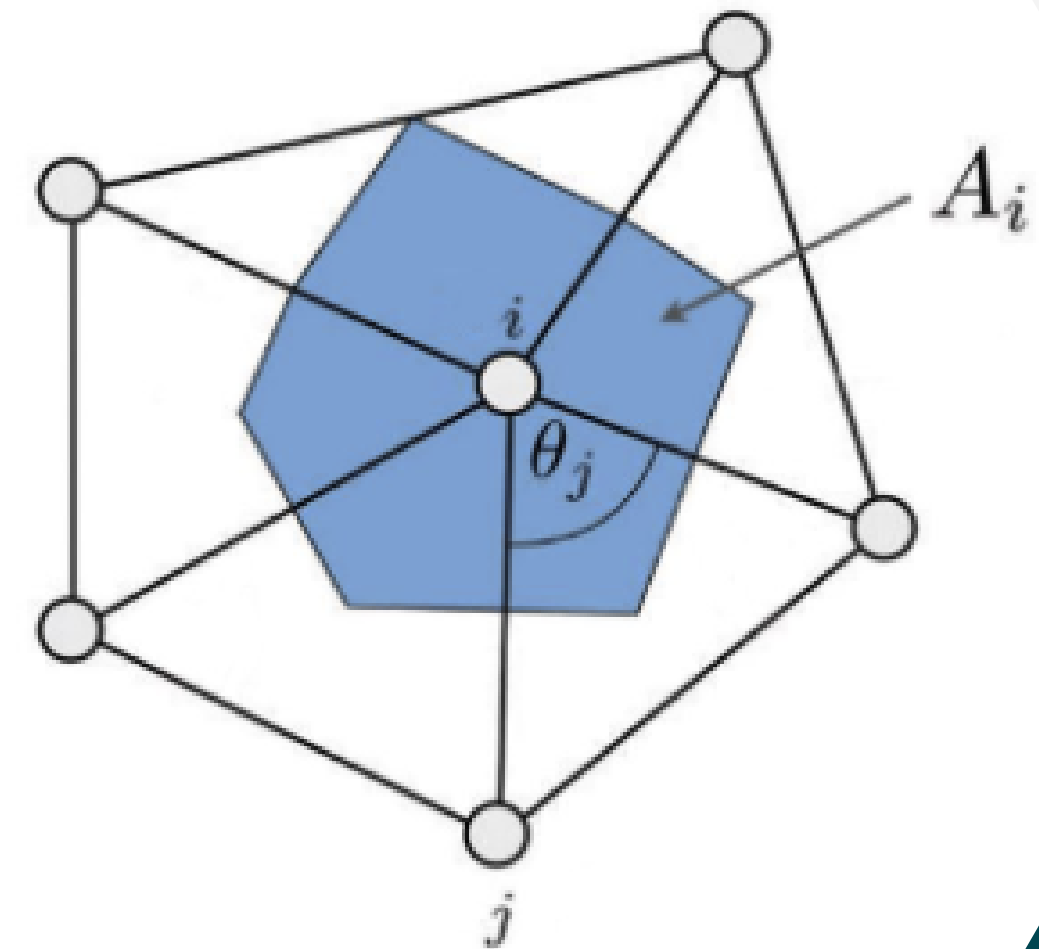
Método con el Laplaciano

LAPLACIANO

Se define como la divergencia, o “la suma de las segundas derivadas”. Una buena aproximación es:

$$(\nabla^2 u)_i = \frac{1}{2A_i} \sum_{j \in C_i} (\cot(\alpha_{i,j}) + \cot(\beta_{i,j}))(u_j - u_i)$$

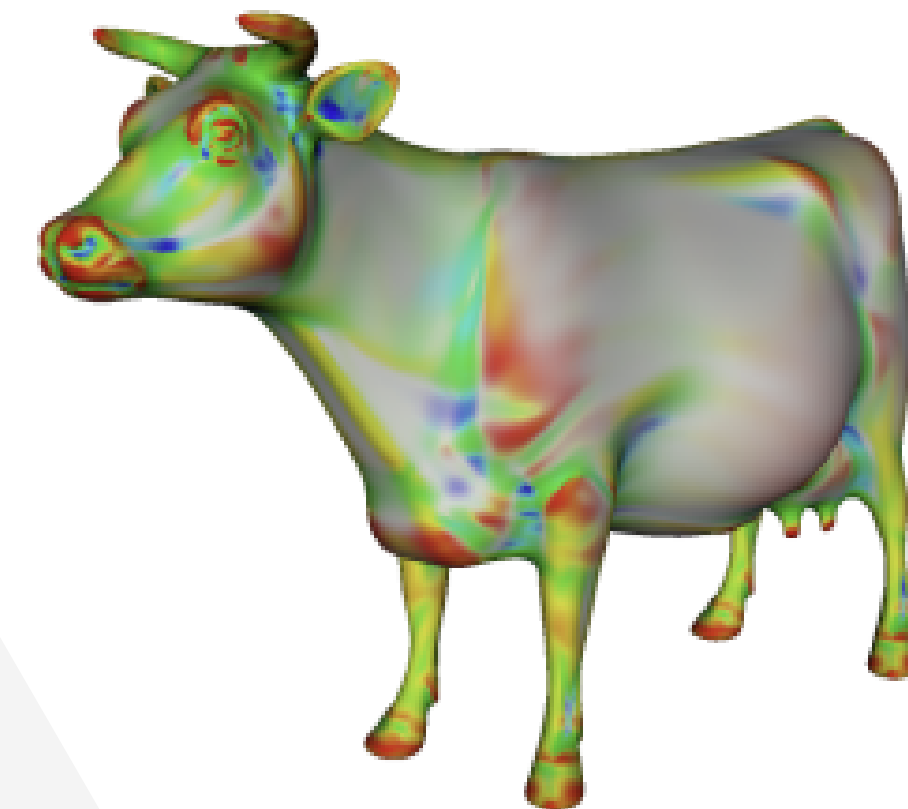
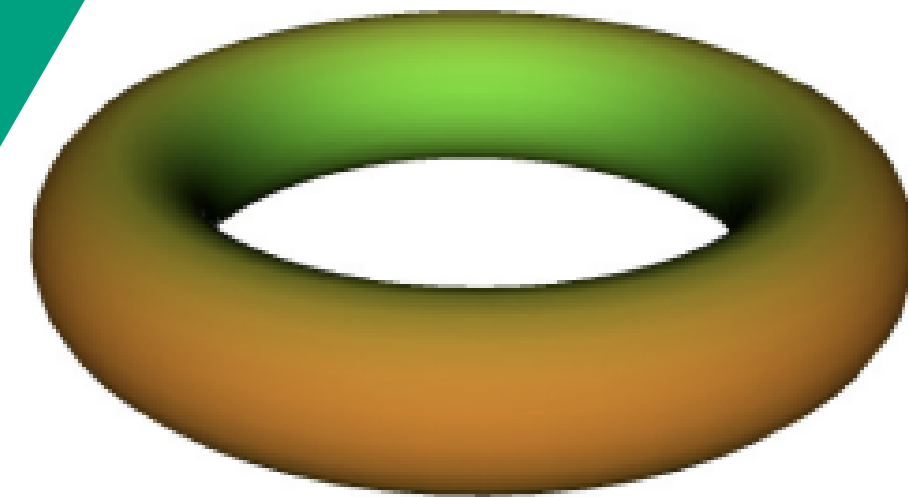
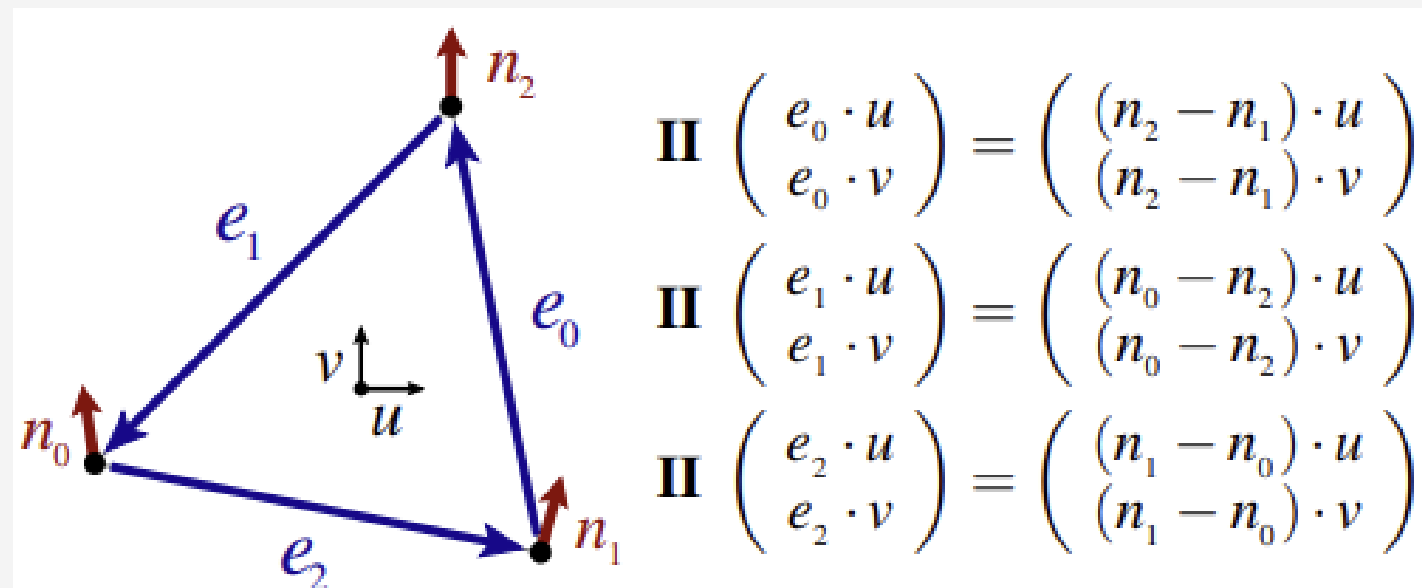
$$|H| = \frac{\|\Delta p_i\|}{2} \quad \langle S \vec{n}_i, -\Delta p_i \rangle$$



Método de Rusinkiewicz

MATRIZ DE WEINGARTEN

El autor propone un algoritmo que permita obtener una matriz, cuyos valores propios son las curvaturas principales. Tiene una implementación en C++.



Otros Métodos

TAUBIN (1995)

También se centra en la búsqueda de una matriz para obtener sus valores propios.

$$k_{ij} = \frac{2n_i \cdot (p_i - p_j)}{|p_i - p_j|^2}$$



$$M_i = \sum_{j \in \mathcal{N}(i)} k_{ij} (p_i - p_j) (p_i - p_j)^T$$

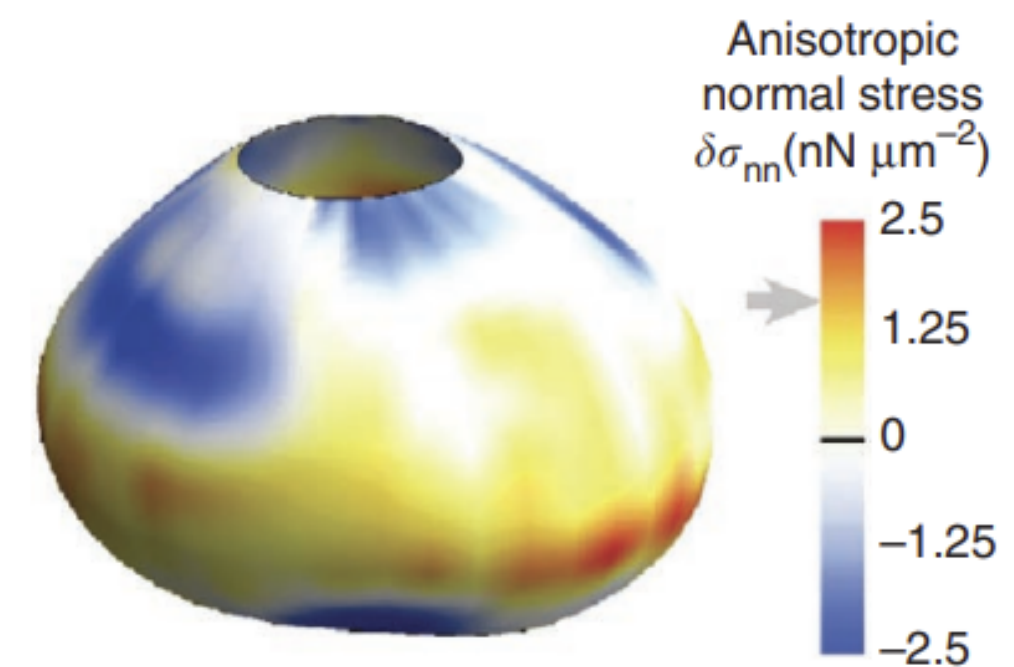
Paper Original

IMÁGEN Y CONTORNOS EN 2D

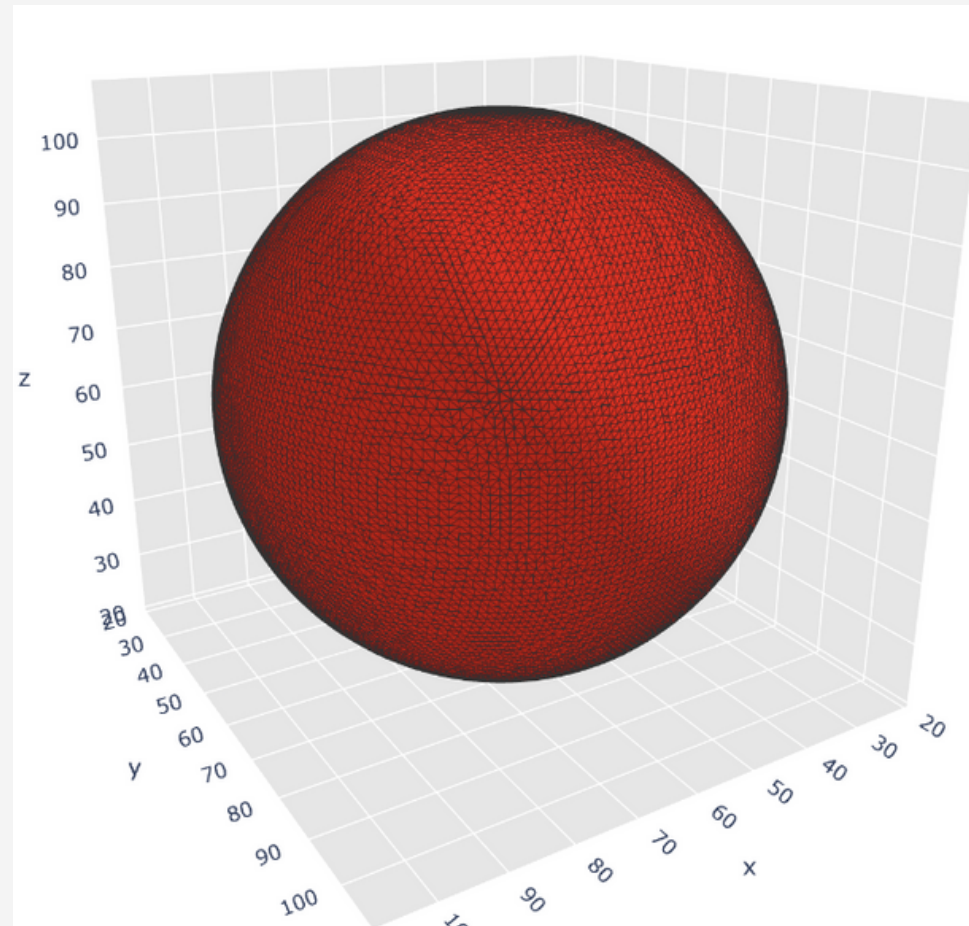
se obtiene la imagen 2D de la gota. Luego, se detecta su contorno y se aplica una serie de filtros. Se acondiciona en ImageJ y se obtienen sus coordenadas en Matlab. Por último, usando Wolfram Mathematica 8 se obtiene una curva cerrada.

PROCESAMIENTO EN 3D

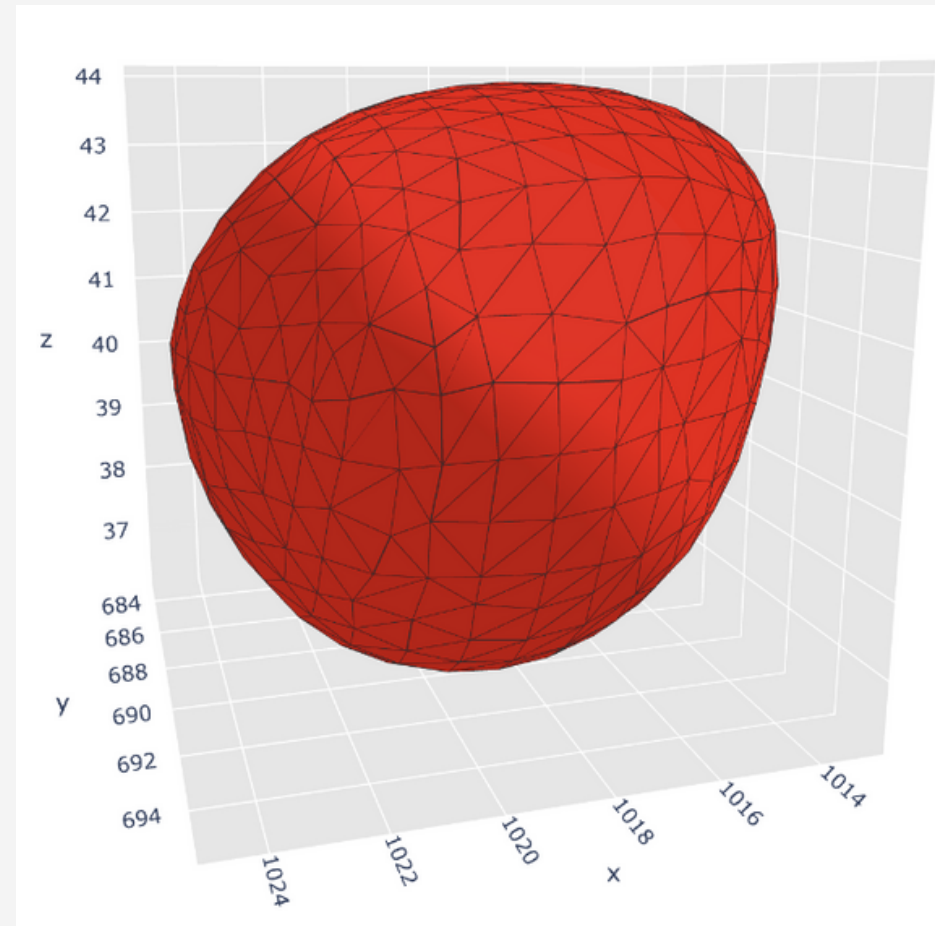
Con el mismo software se utilizó BSpline 2D para obtener toda la superficie continua. Se obtuvo H de manera teórica.



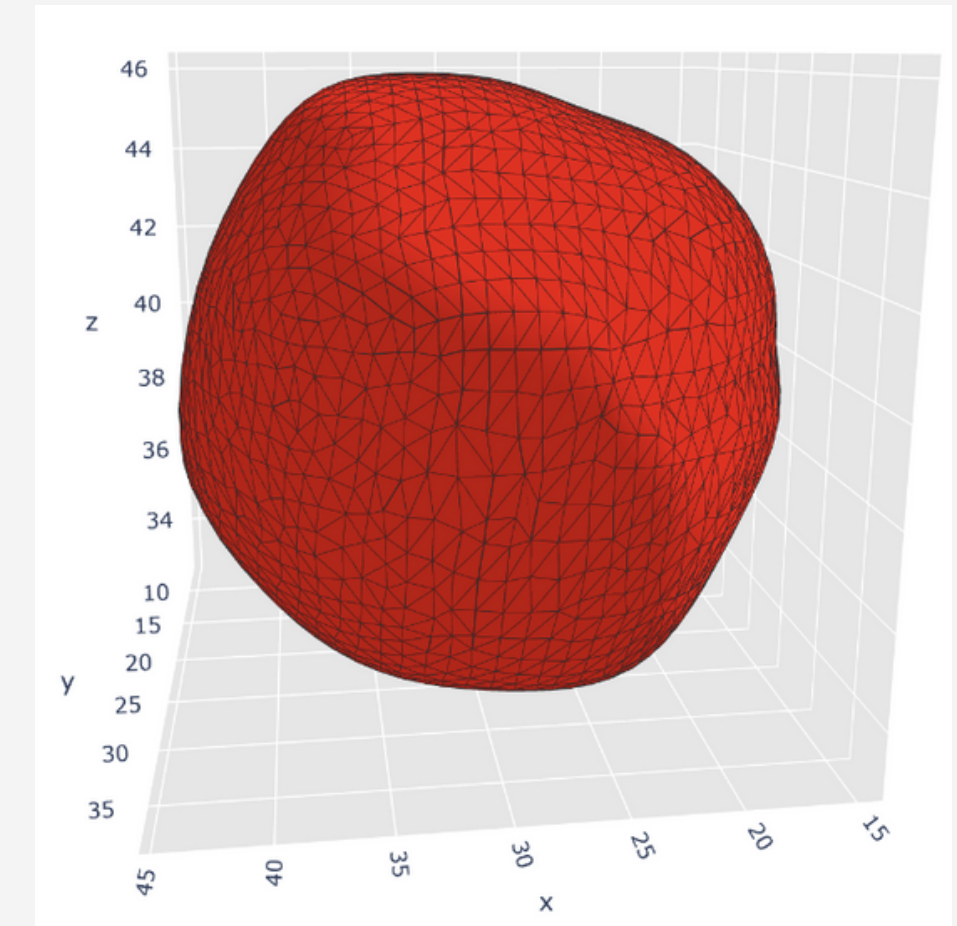
Resultados (Original)



ESFERA PERFECTA

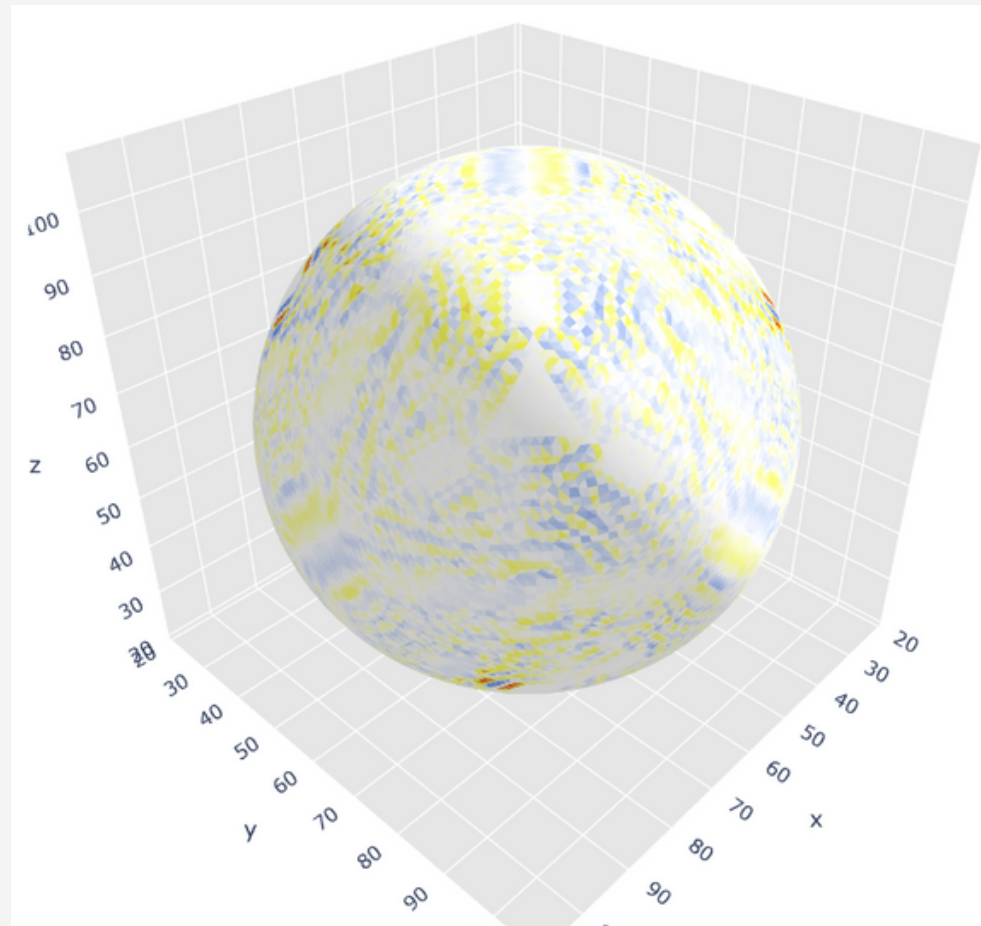


GOTA DE CONTROL

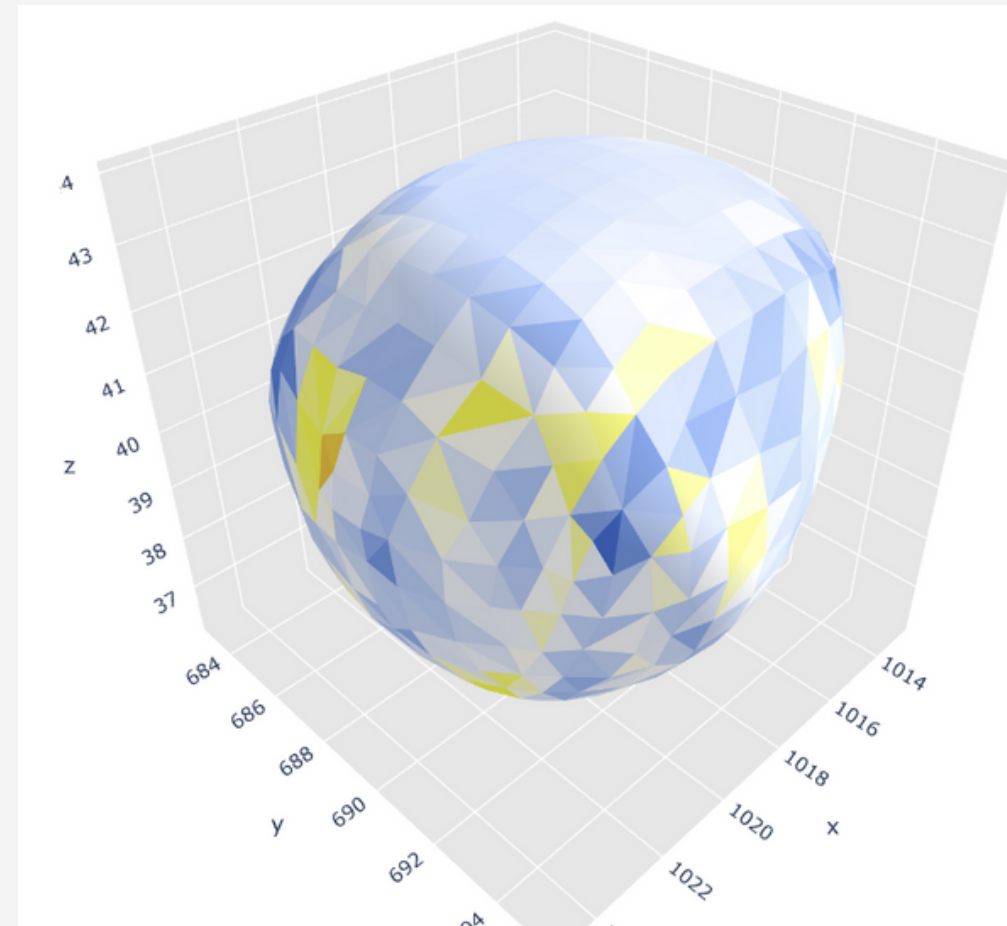


GOTA CERCA DE UNA CELULA

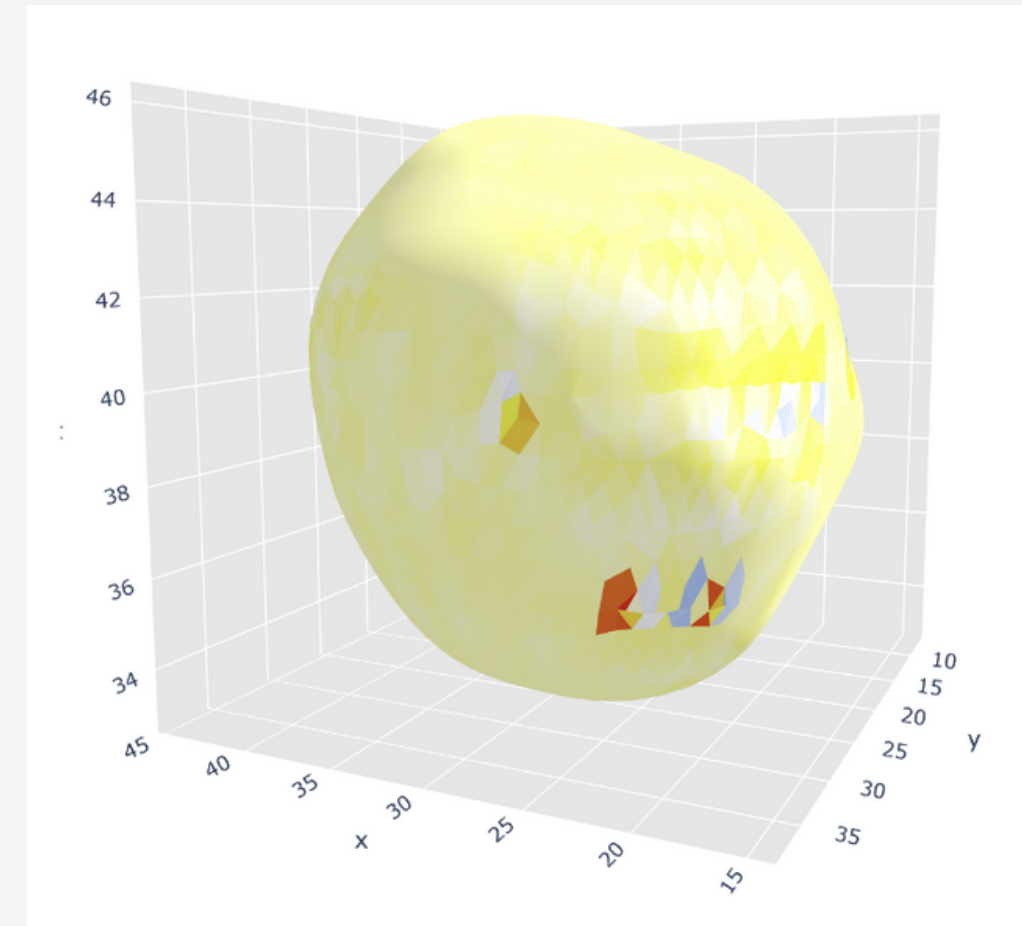
Resultados (Discreto)



ESFERA PERFECTA

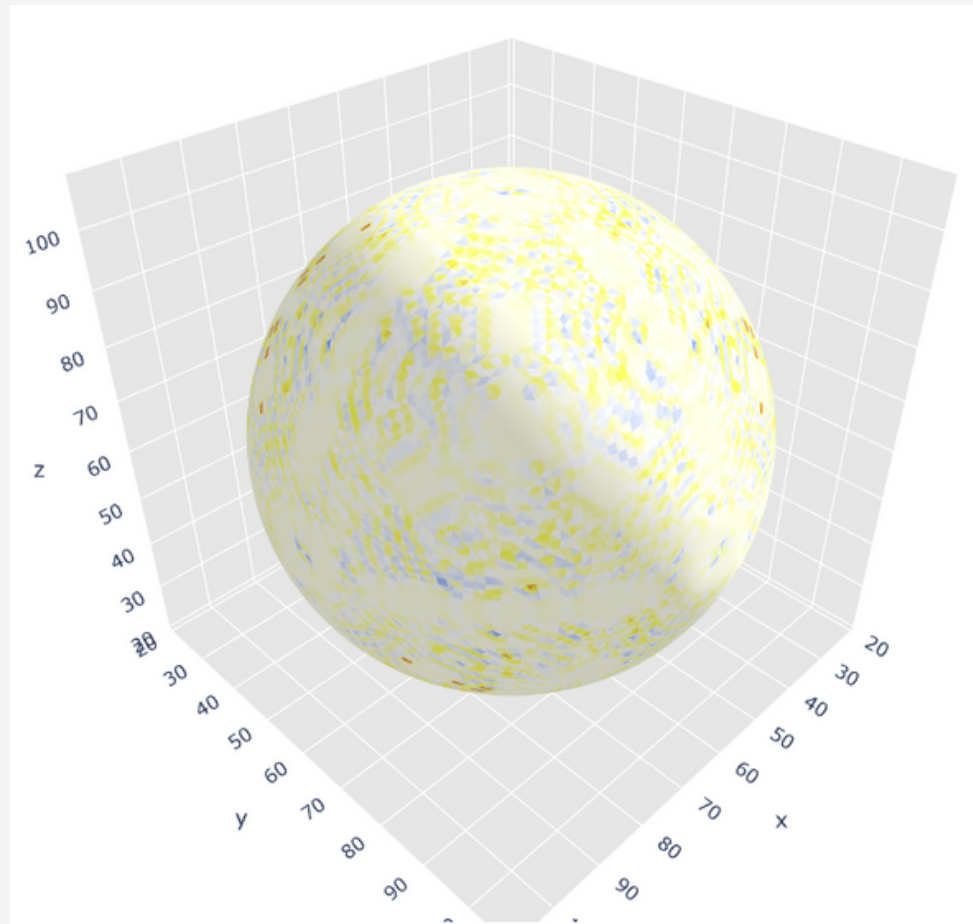


GOTA DE CONTROL

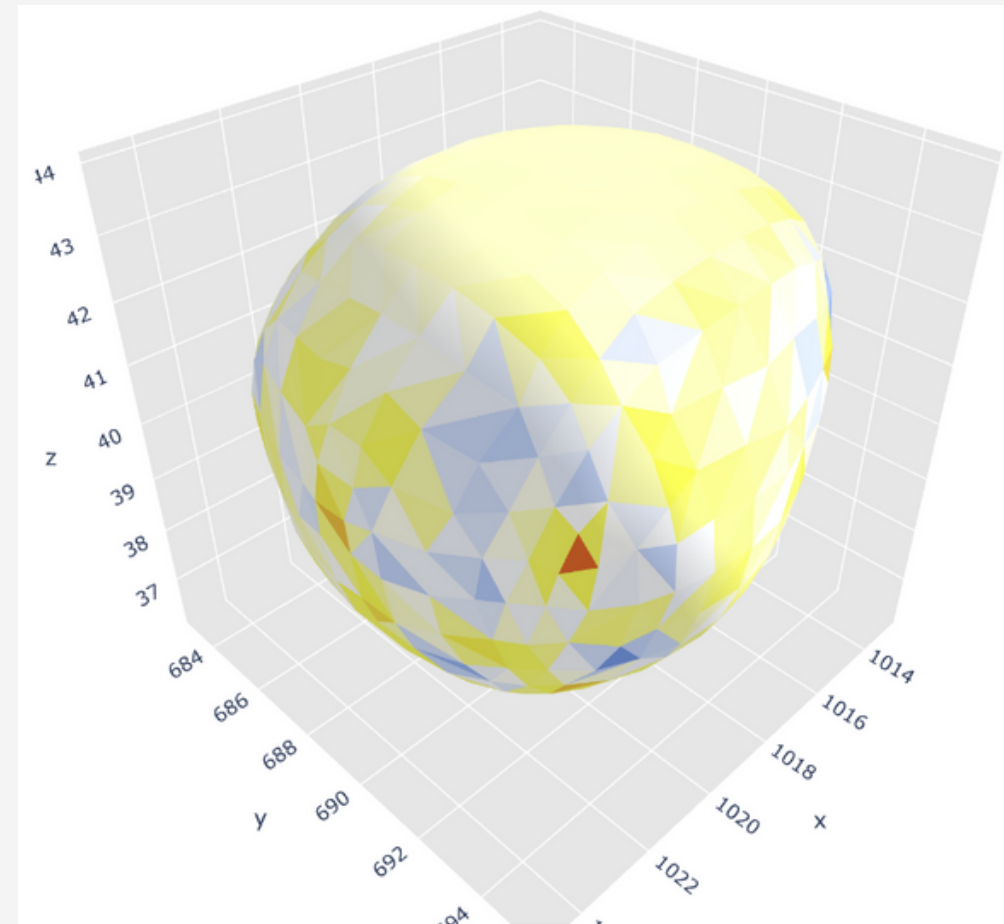


GOTA CERCA DE UNA CELULA

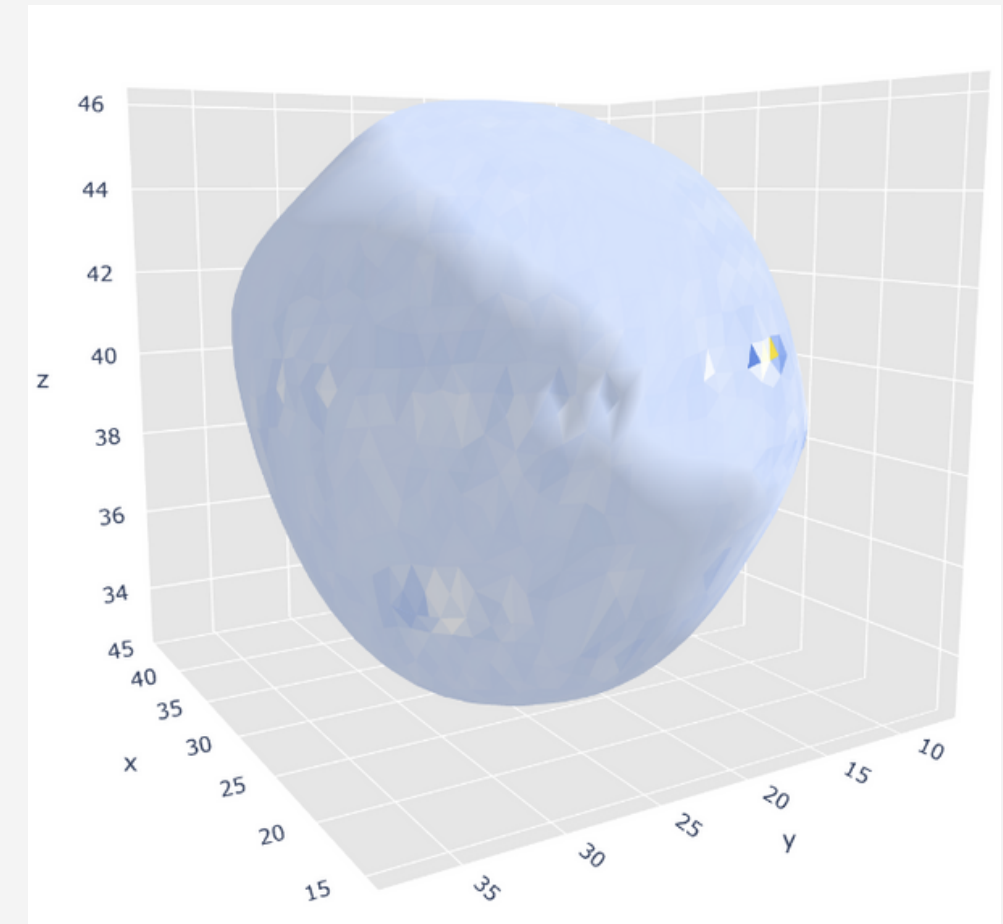
Resultados (Laplaciano)



ESFERA PERFECTA

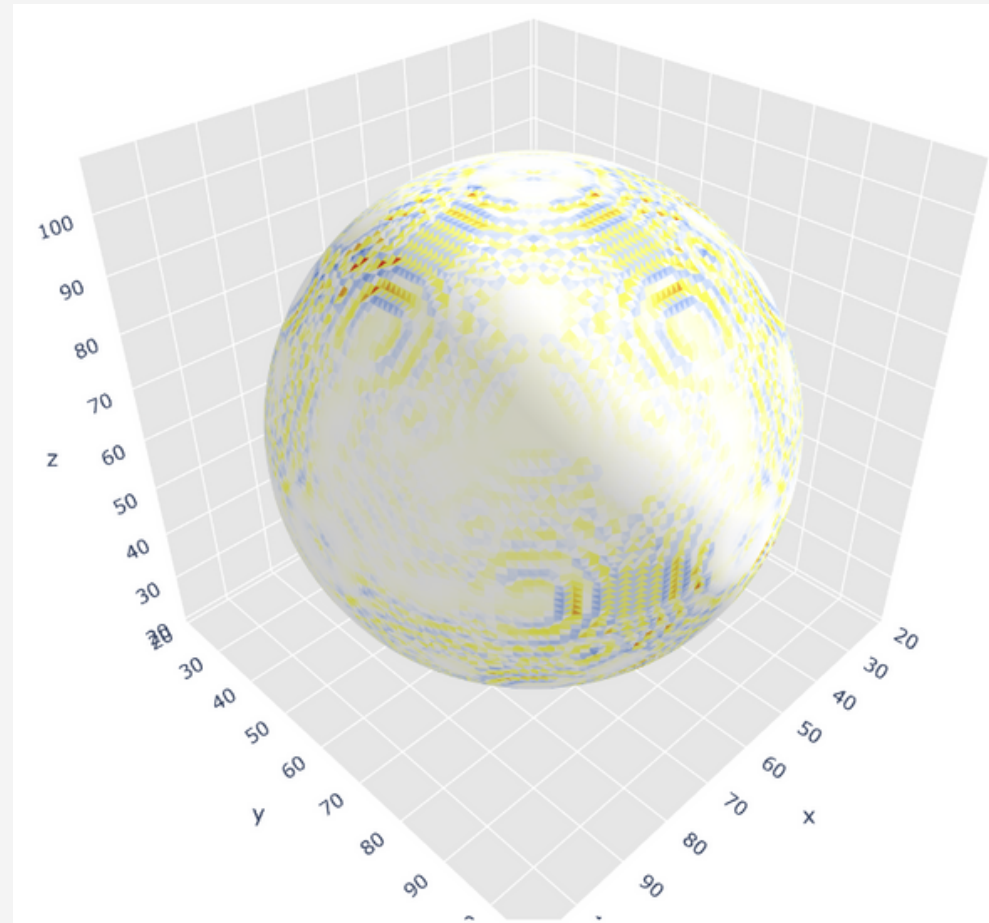


GOTA DE CONTROL

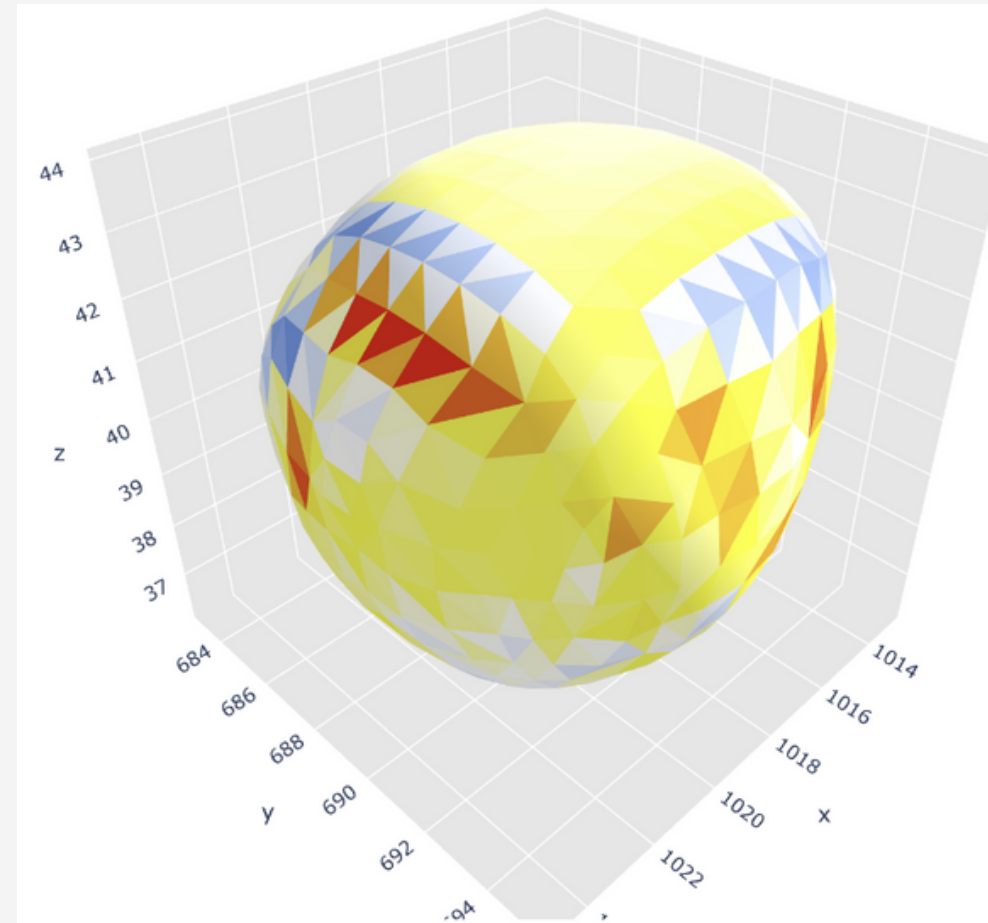


GOTA CERCA DE UNA CELULA

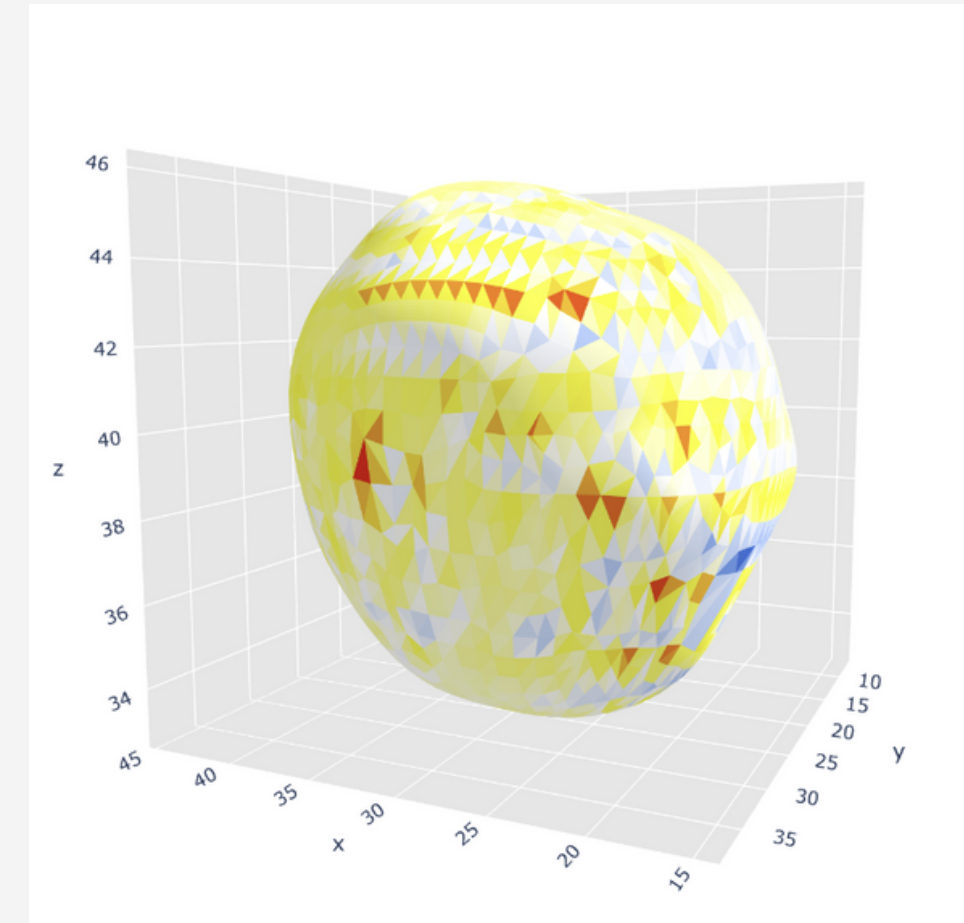
Resultados (Taubin)



ESFERA PERFECTA



GOTA DE CONTROL



GOTA CERCA DE UNA CELULA

¿Cómo seguir?

RADIO ORIGINAL

En una esfera es posible obtenerlo, pero ¿En una deformada? Se ha preferido investigar sobre las curvaturas.

CURVATURAS

Todavía se están trabajando en los algoritmos. Se tiene que verificar bien su eficacia. También, hay que decantarse por alguno de los métodos.

INTERFAZ

Ya se tiene una idea de cómo podría ser, mucho más sencilla de lo pensada, pero efectiva.



FIN