

7. Anexos

Anexo A. Obtención del campo de desplazamientos: Comparación de los diferentes métodos de extrapolación de Matlab

Para la obtención del campo de desplazamientos en las coordenadas nodales de la malla del modelo de elementos finitos generado, es necesario realizar una extrapolación, ya que los puntos donde se conocen los desplazamientos iniciales no coinciden con dichas coordenadas. Se ha utilizado el software Matlab para realizar esta tarea y, dado que en él se ofrecen diferentes métodos de interpolación de datos, se ha realizado un estudio comparando los resultados obtenidos con cada uno de estos métodos.

Además, se realizará un estudio de la sensibilidad de la malla ante los desplazamientos impuestos. Es decir, se compararán varios tamaños de malla a la hora de extrapolar para ver si la precisión con respecto al campo de desplazamientos inicial varía. Se probarán mallas con un tamaño máximo de elemento de ocho, diez y doce micras.

Tanto la comparación de métodos como el análisis de sensibilidad solo han sido realizados para la malla del primer caso de célula relajada con tripsina. Es de suponer que el método seleccionado para este caso será válido también para el resto, pues no hay ninguna diferencia significativa que haga pensar lo contrario.

Inicialmente, disponemos de un campo de desplazamientos con una distribución de puntos uniforme en forma cúbica (los cuales han sido obtenidos experimentalmente) con las dimensiones mostradas en la tabla 1. El campo final posee unas dimensiones mayores y, además, se trata de puntos incluidos en el rectángulo delimitado por el gel pero distribuidos en su interior aleatoriamente por el espacio, sin ningún tipo de patrón geométrico. La variación del tamaño de malla no cambia las dimensiones de la misma, solo lo poblada de puntos que está.

En primer lugar, es necesario centrar los puntos iniciales con las coordenadas de la malla para poder realizar correctamente la extrapolación. Para ello, se traslada el centro geométrico del campo inicial de desplazamientos hasta la posición del centro geométrico de las coordenadas nodales. Ha de tenerse en cuenta que se ha añadido un volumen extra al gel para que la célula no estuviese tan próxima al extremo del modelo por lo que, para centrar los desplazamientos en torno a la posición real de la célula, se trasladará el campo inicial hasta el centro geométrico de las coordenadas nodales antes de ampliar su volumen (Tabla A1) (Figura A1).

	Puntos Iniciales			Coordenadas Nodales			
	Eje X	Eje Y	Eje Z	Eje X	Eje Y	Eje Z	
Valor máximo	325,18	329,9	54,81	Valor máximo	321,9	321,9	57,96
Valor mínimo	27,58	42,22	3,15	Valor mínimo	0	0	0
Dimensiones	297.6	287.68	51.66	Dimensiones	321.9	321.9	57.96
centro	176,38	186,06	28,98	centro	160,95	160,95	28,98

Tabla A1. Dimensiones de las matrices de puntos.

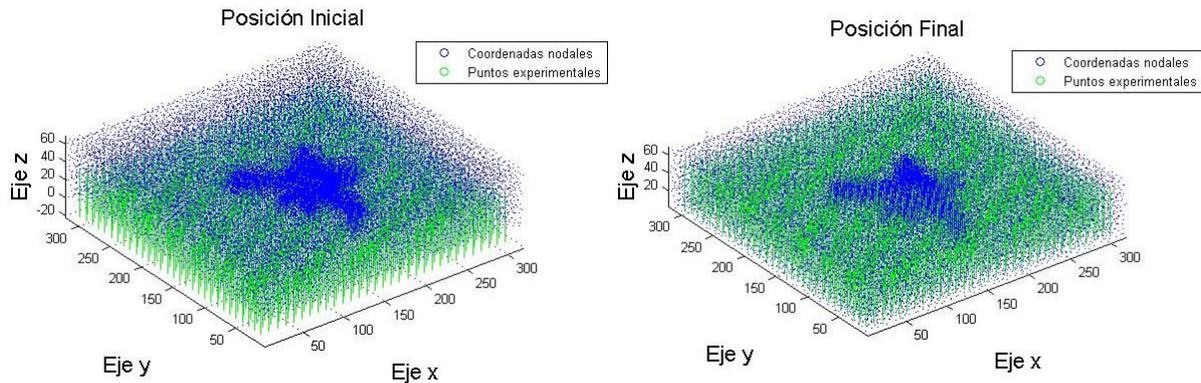


Figura A1 Posicionamiento relativo del campo inicial de desplazamientos.

Una vez posicionados entre sí correctamente los puntos, pasamos a extrapolar los desplazamientos. Matlab ofrece las siguientes órdenes de interpolación y extrapolación:

- `interp`
- `griddata`
- `scatteredInterpolant`
- `spline`
- `pchip`

Los métodos *spline* y *pchip* se han descartado directamente por ser solamente válidos para interpolaciones y extrapolaciones en 2D. Además, una vez estudiado el método *griddata*, se puede apreciar que este método no es capaz de extrapolar fuera del dominio del volumen inicial y, como nuestro campo final tiene mayor dominio que el inicial, es necesario realizar una extrapolación a los puntos exteriores. Lo mismo sucede con la orden *interp*, que también es capaz solamente de interpolar por lo que estos dos métodos también se descartan. Por lo tanto, solamente se estudiará la orden *scatteredInterpolant* con todas sus posibles variantes.

Lo primero es crear una función, para cada uno de los métodos, que extrapole los desplazamientos automáticamente introduciendo como parámetros de entrada las coordenadas de los puntos iniciales, los desplazamientos iniciales y las coordenadas de los puntos en los que queremos saber los desplazamientos. La

función *scatteredInterpolant* ofrece varios métodos distintos de interpolación y extrapolación (Tabla A2), se probarán todas ellas para ver cuál se aproxima mejor a los datos iniciales.

Método	Descripción	Continuidad
lineal	Triangulación basada en interpolación lineal. Válida para 2D y 3D	C^0
nearest	Triangulación basada en una interpolación en función del valor tomado por el punto más cercano (nearest neighbour interpolation). Válida para 2D y 3D	Discontinua
natural	Triangulación basada en una interpolación en función de los valores tomados en su entorno (natural neighbour interpolation). Válida para 2D y 3D	C^1

Tabla A2. Métodos ofrecidos en *scatteredInterpolant*

Teniendo esto en cuenta crearemos una función para cada una de las variantes de nuestra orden de la forma:

```
function VF = extrapolate3D(PI,VI,PF)
%Realiza una interpolación y una extrapolación 3D usando la orden
%scatteredInterpolant con los métodos 'linear', 'linear'
%VARIABLES DE ENTRADA Y DE SALIDA:
%PI=Puntos del campo de desplazamientos inicial
%VI=Desplazamientos iniciales en los puntos PI
%PF=Puntos en donde se desea conocer los desplazamientos
%VF=Matriz de desplazamientos interpolados

%Separa la matrices introducidas en columnas
[PIx,PIy,PIz]=sepxyz(PI);
[VIx,VIy,VIz]=sepxyz(VI);
[PFx,PFy,PFz]=sepxyz(PF);
%Crea las funciones de interpolación/extrapolación
Fx=scatteredInterpolant(PIx(:),PIy(:),PIz(:),VIx(:),'linear','linear');
Fy=scatteredInterpolant(PIx(:),PIy(:),PIz(:),VIy(:),'linear','linear');
Fz=scatteredInterpolant(PIx(:),PIy(:),PIz(:),VIz(:),'linear','linear');
%Realiza la interpolación/extrapolación en cada uno de los ejes
VFx=Fx(PFx,PFy,PFz);
VFy=Fy(PFx,PFy,PFz);
VFz=Fz(PFx,PFy,PFz);
%Concatena todas las columnas en una sola matriz
VF=[VFx,VFy,VFz];
end
```

Esta es la función para el método de interpolación-extrapolación lineal, las otras dos funciones serán idénticas pero cambiando la especificación de método "*linear*" por "*nearest*" y "*natural*" respectivamente.

Para el funcionamiento de la orden *extrapolate3D* se utiliza otra función llamada *sepxyz*, esta es una orden que permite separar por columnas una matriz de forma automática:

```
%Separa por columnas la matriz introducida
function [x,y,z] = sepxyz(M)
c=size(M,2);
if c==3
    x=M(:,1);
    y=M(:,2);
    z=M(:,3);
else
    disp('Introduzca una matriz de 3 columnas.');
```

Una vez realizadas las interpolaciones con cada uno de los métodos, es necesario definir una función que represente en 3D el campo vectorial, para poder visualizar los resultados. Para ello, se ha utilizado la función "quiver3c" que crea una escala de color dependiente del módulo del desplazamiento en cada punto. Esta función y había sido creada previamente [12] (Figuras A2-A11).

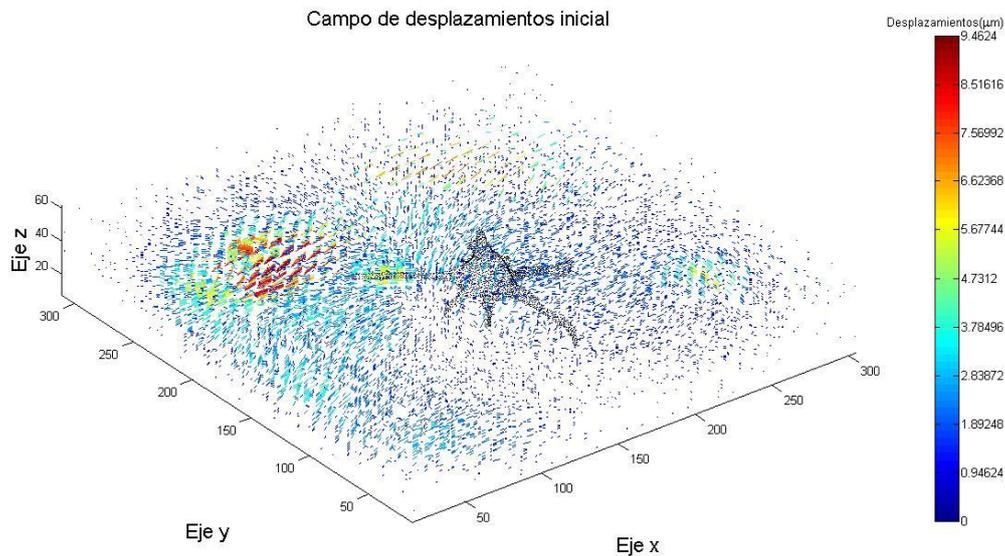


Figura A2. Representación del campo de desplazamientos inicial con la orden quiver3color.

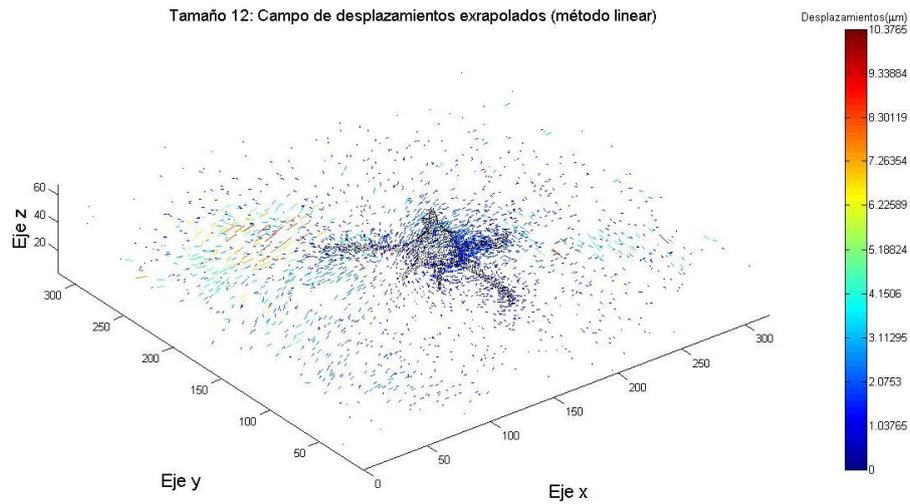


Figura A3. Campo de desplazamientos extrapolado con la función `scatteredInterpolant` para el tamaño de malla de 12 micras (método lineal). Representación con la orden `quiver3color`.

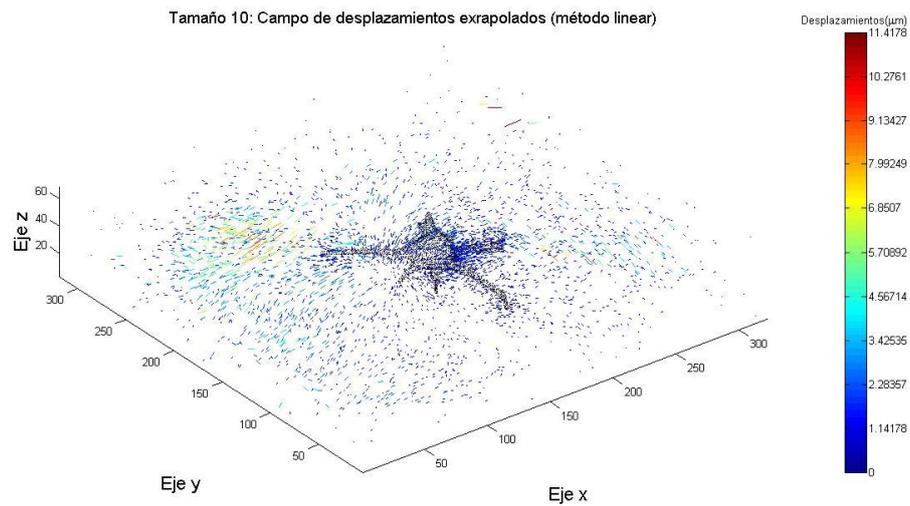


Figura A4. Campo de desplazamientos extrapolado con la función `scatteredInterpolant` para el tamaño de malla de 10 micras (método lineal). Representación con la orden `quiver3color`.

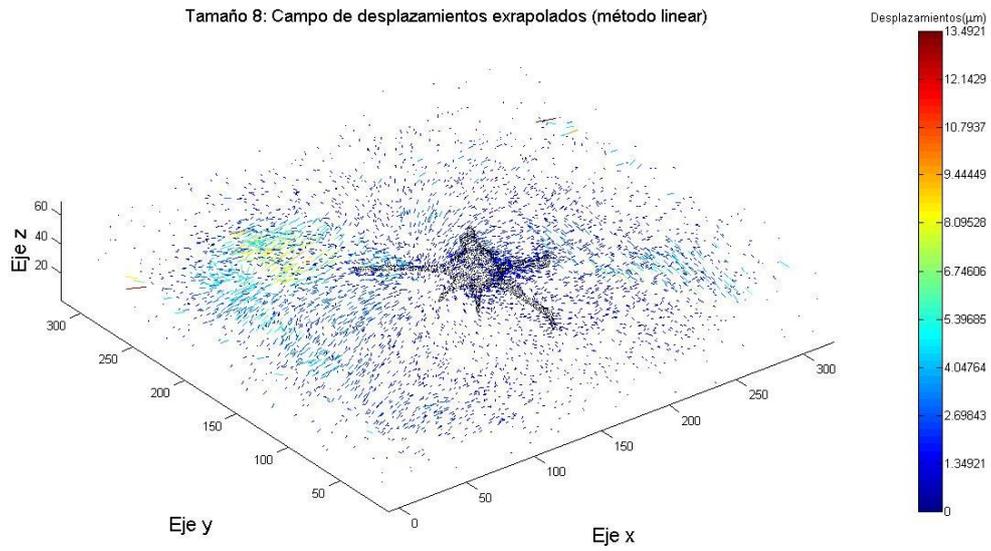


Figura A5. Campo de desplazamientos extrapolado con la función *scatteredInterpolant* para el tamaño de malla de 8 micras (método lineal). Representación con la orden *quiver3color*.

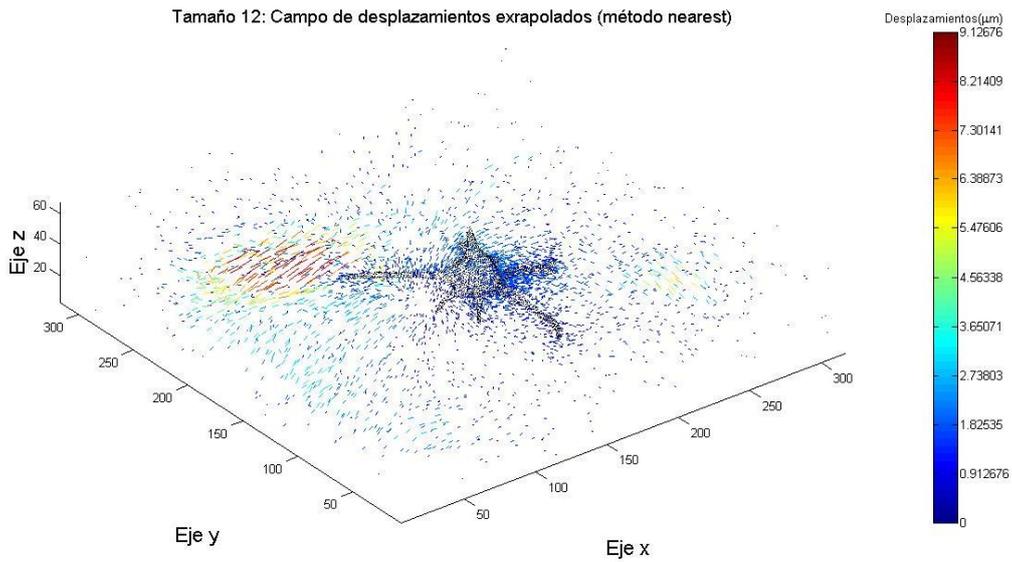


Figura A6. Campo de desplazamientos extrapolado con la función *scatteredInterpolant* para el tamaño de malla de 12 micras (método nearest). Representación con la orden *quiver3color*.

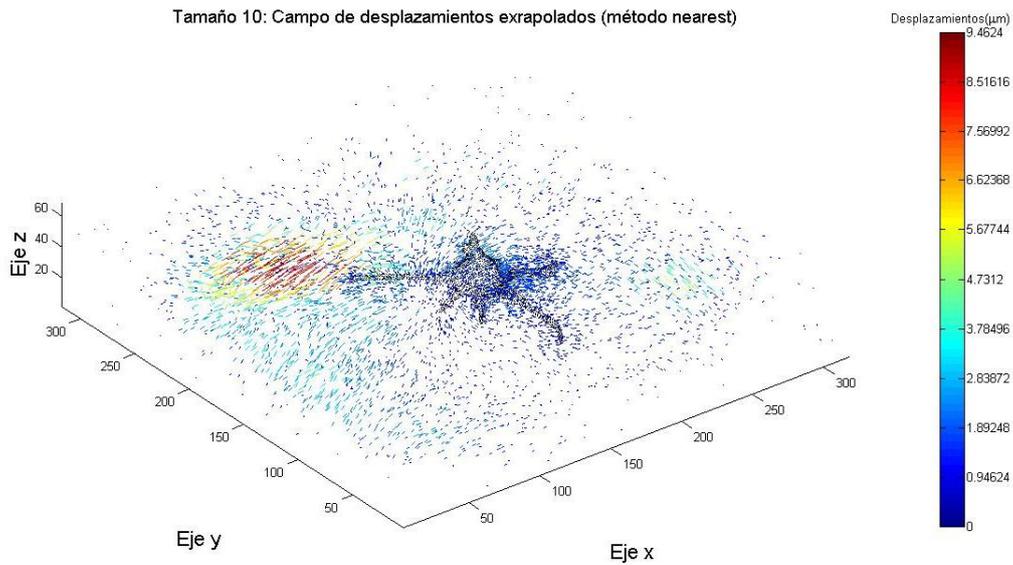


Figura A7. Campo de desplazamientos extrapolado con la función `scatteredInterpolant` para el tamaño de malla de 10 micras (método nearest). Representación con la orden `quiver3color`.

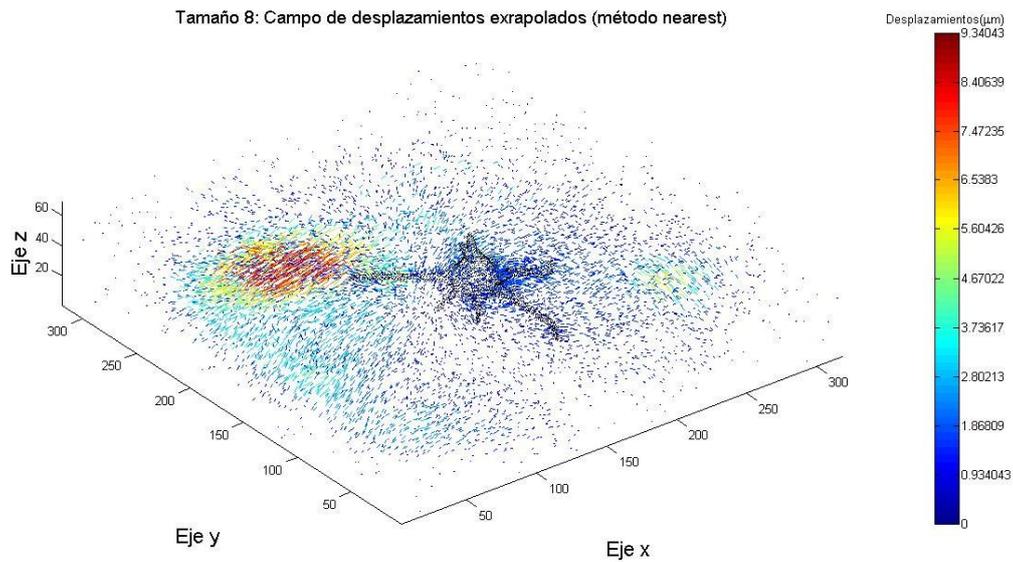


Figura A8. Campo de desplazamientos extrapolado con la función `scatteredInterpolant` para el tamaño de malla de 8 micras (método nearest). Representación con la orden `quiver3color`.

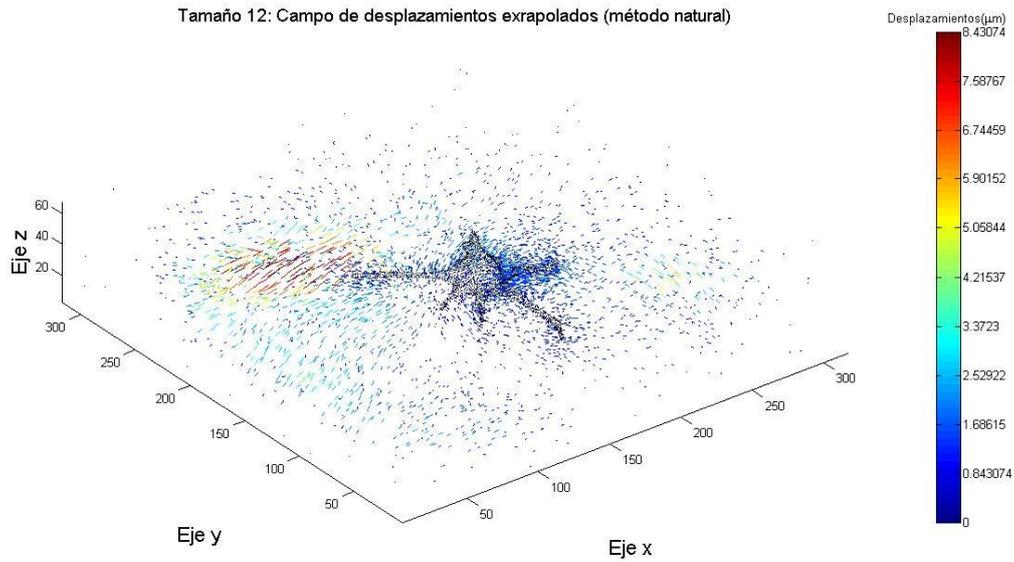


Figura A9. Campo de desplazamientos extrapolado con la función `scatteredInterpolant` para el tamaño de malla de 12 micras (método natural). Representación con la orden `quiver3color`.

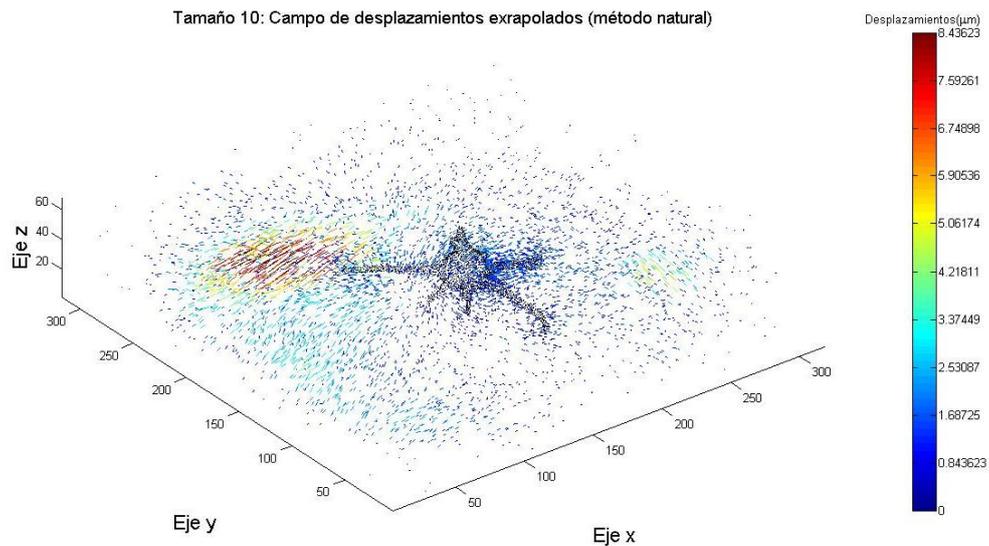


Figura A10. Campo de desplazamientos extrapolado con la función `scatteredInterpolant` para el tamaño de malla de 10 micras (método natural). Representación con la orden `quiver3color`.

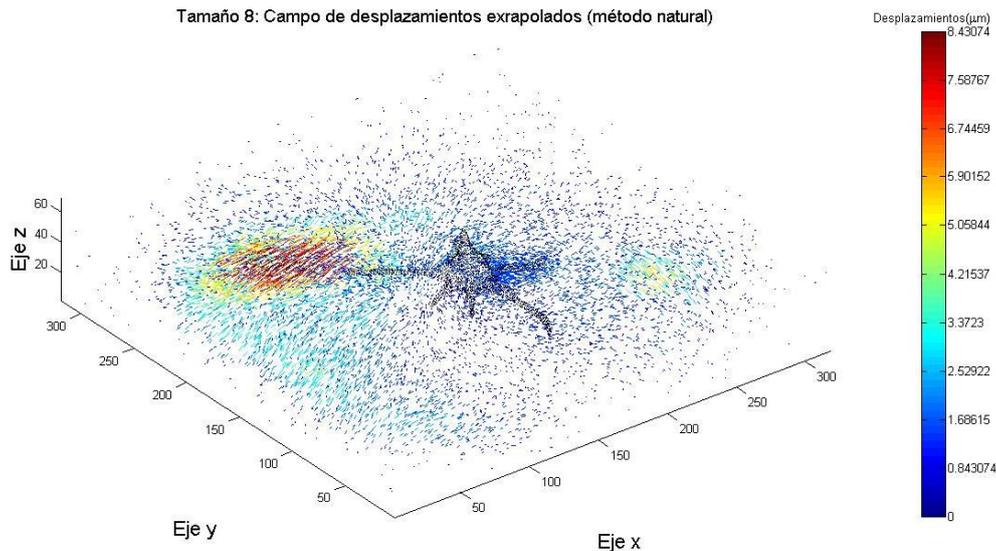


Figura A11. Campo de desplazamientos extrapolado con la función *scatteredInterpolant* para el tamaño de malla de 8 micras (método natural). Representación con la orden *quiver3color*.

Si se observa la escala de color a la derecha de cada gráfica, se aprecia que los valores máximos obtenidos es bastante similar exceptuando el caso de la interpolación lineal. Por este motivo, se descarta este método debido a la mala calidad de los resultados obtenidos con él.

Para poder apreciar con más detalle la distribución de desplazamientos en cada caso, se han implementado otras dos funciones (*maximo* y *minimo*) que nos permiten obtener la localización de los máximos y mínimos obtenidos en cada caso y otra función extra (*histograma*) que nos permite ver una función de distribución de los desplazamientos:

```
function [pmax,dmax] = maximo(points,desp)
%Permite obtener los puntos donde los desplazamientos son máximos
%(0.85*maximo) y el valor de dichos desplazamientos (opcional)
%VARIABLES DE ENTRADA Y DE SALIDA:
%points=puntos a estudiar
%desp= valor del desplazamiento en cada punto
%pmax=puntos donde los desplazamientos superan el 85% del valor máximo
%dmax=valor de los desplazamientos en los puntos pmax (opcional)

[fp]=size(points,1);
[fd]=size(desp,1);
%Calcula el modulo de cada desplazamiento
mod=sqrt(sum(desp.^2,2));
%Calcula el valor modular máximo
max_mod=max(mod);
```

```

%Inicializa la matriz auxiliar y la va completando con los valores
máximos
if fp==fd
    M=horzcat(points, desp, mod);
    aux=zeros(1,7);
    [f]=size(M,1);
    for i=1:f
        if M(i,7)>=(0.85*max_mod)
            aux=[aux;M(i,:)];
        end
    end
    %Especifica las diferentes variables de salida
    S=aux(2:end,:);
    pmax=S(:,1:3);
    dmax=S(:,4:6);
else
    disp('points y desp deben tener el mismo número de filas');
end
end

function [pmin,dmin] = minimo(points, desp)
%Permite obtener los puntos donde los desplazamientos son mínimos
%(0.01*maximo) y el valor de dichos desplazamientos (opcional)
%VARIABLES DE ENTRADA Y DE SALIDA:
%points=puntos a estudiar
%desp= valor del desplazamiento en cada punto
%pmin=puntos donde los desplazamientos no superan el 1% del valor
máximo
%dmax=valor de los desplazamientos en los puntos pmin (opcional)
[fp]=size(points,1);
[fd]=size(desp,1);
%Calcula el módulo de cada desplazamiento
mod=sqrt(sum(desp.^2,2));
%Calcula el valor modular máximo
max_mod=max(mod);
%Inicializa la matriz auxiliar y la va completando con los valores
mínimos
if fp==fd
    M=horzcat(points, desp, mod);
    aux=zeros(1,7);
    [f]=size(M,1);
    for i=1:f
        if M(i,7)<=(0.01*max_mod)
            aux=[aux;M(i,:)];
        end
    end
    %Especifica las diferentes variables de salida
    S=aux(2:end,:);
    pmin=S(:,1:3);
    dmin=S(:,4:6);
else
    disp('points y desp deben tener el mismo número de filas');
end
end

```

```

function S = histograma(desp,color,anchuraLinea)
%Crea una función de distribución donde se representa la cantidad de
nodos que alcanzan
%un determinado valor de desplazamiento.
% VARIABLES DE ENTRADA Y SALIDA:
% desp=Campo de desplazamientos a estudiar
% color=color deseado para el gráfico
% anchuraLinea=anchura de línea deseada para el gráfico (opcional)
%S=vector de salida que contiene la distribución porcentual de los
%desplazamientos
if nargin<=2
    anchuraLinea=1;
end
%Calcula el módulo del desplazamiento para cada nodo y su valor máximo
mod=sqrt(sum(desp.^2,2));
max_mod=max(mod);
Lm=length(mod);
%Inicializa los escalares y las variables necesarias para el bucle
n_nodos=0;
j=1;
aux=zeros(101,1);
mod_c=zeros(101,1);
%Compara cada valor de 'mod' con un tanto porcentual del
desplazamiento
%máximo y contabiliza en 'aux' cuántos nodos (porcentualmente) superan
o igualan ese valor, es decir,
%si hay 3000 nodos que superan el 65% del valor máximo y en total
tenemos 6000 nodos, en la posición 65
%de 'aux' obtendremos el valor 50(%).
for k=0:0.01:1
    for i=1:Lm
        if mod(i)>=k*max_mod
            n_nodos=n_nodos+1;
        end
    end
    aux(j)=((n_nodos/Lm)*100);
    mod_c(j)=(k*max_mod);
    j=j+1;
    n_nodos=0;
end
%Crea la variable de salida
S=aux;
%Aproxima con la orden 'pchip' una curva que describa la trayectoria
%marcada por los valores puntuales obtenidos en el bucle.
p=0:0.001:max_mod;
graf=pchip(mod_c,S,p);
plot(p,graf,color,'LineWidth',anchuraLinea)
xlabel('Desplazamientos (\mum)','FontSize',16)
ylabel('Número de nodos (%)','FontSize',16)
end

```

Con esta última función ya creada, se ejecuta el siguiente programa para que represente las funciones de distribución de los diferentes tamaños de malla y que estudie el error cometido:

```

%%NEAREST%%
%Representa, en este caso para el método 'nearest', en una misma
%figura, las funciones de distribución de los diferentes tamaños de
%mallla, el error que cometen con respecto a la distribución inicial y
%el error cometido con respecto al tamaño de malla anterior mediante
%la orden 'subplot'

%Representación de las funciones de distribución
subplot(2,2,[1,2]);
Si=histograma(desp_def,'k',2);
hold on
S12_nn=histograma(vintspnn_12,'r');
S10_nn=histograma(vintspnn_10,'g');
S8_nn=histograma(vintspnn_8,'b');

legend('Inicial','Tamaño 12','Tamaño 10','Tamaño 8');
title('Comparación de la distribución de desplazamientos
(nearest)','FontSize',16)
hold off

%Representación del error con respecto a la inicial
subplot(2,2,3);
x=0:1:100;
Si_12_nn=abs((Si-S12_nn)/Si)*100;
Si_10_nn=abs((Si-S10_nn)/Si)*100;
Si_8_nn=abs((Si-S8_nn)/Si)*100;
plot(x,Si_12_nn,'r','LineWidth',2)
hold on
plot(x,Si_10_nn,'g','LineWidth',2)
plot(x,Si_8_nn,'b','LineWidth',2)
hold off
xlabel('Desplazamientos (%)','FontSize',16)
ylabel('Error (%)','FontSize',16)
title('Error respecto a la distribución inicial','FontSize',16)
legend('Tamaño 12','Tamaño 10','Tamaño 8');

%Representación del error con respecto a la malla anterior
subplot(2,2,4);
S12_10_nn=abs((S12_nn-S10_nn)/S12_nn)*100;
S10_8_nn=abs((S10_nn-S8_nn)/S10_nn)*100;
plot(x,S12_10_nn,'r','LineWidth',2)
hold on
plot(x,S10_8_nn,'g','LineWidth',2)
hold off
xlabel('Desplazamientos (%)','FontSize',16)
ylabel('Error (%)','FontSize',16)
title('Error entre mallas','FontSize',16)
legend('Tamaño 12-Tamaño 10','Tamaño 10-Tamaño 8');

```

Usando estas funciones, se obtienen los valores máximos, los valores mínimos y la función de distribución de los desplazamientos para cada caso. A continuación, se comparan con los valores originales para los diferentes tamaños de malla (Figuras A12-A17).

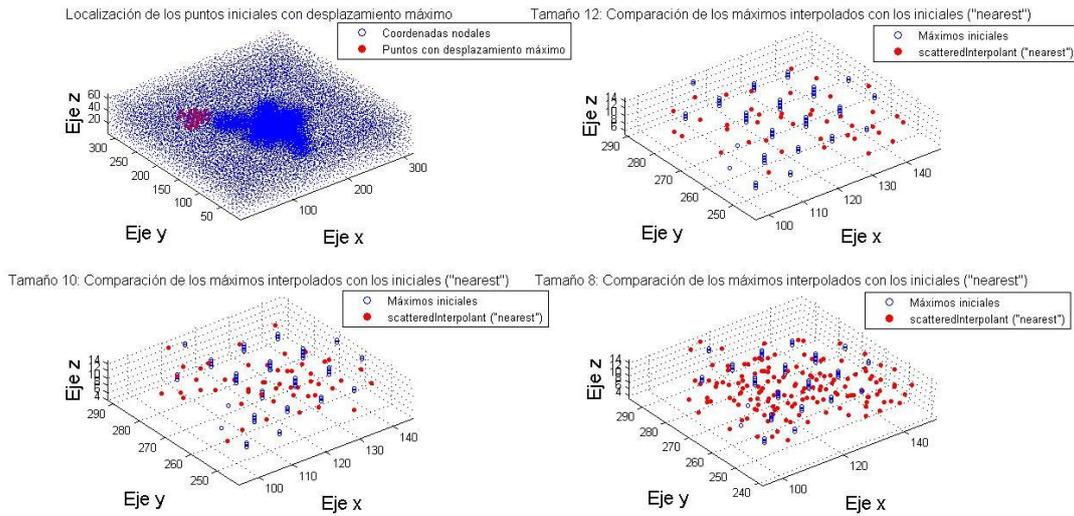


Figura A12. Comparación de los máximos iniciales con los extrapolados para los diferentes tamaños de malla. (Método Nearest)

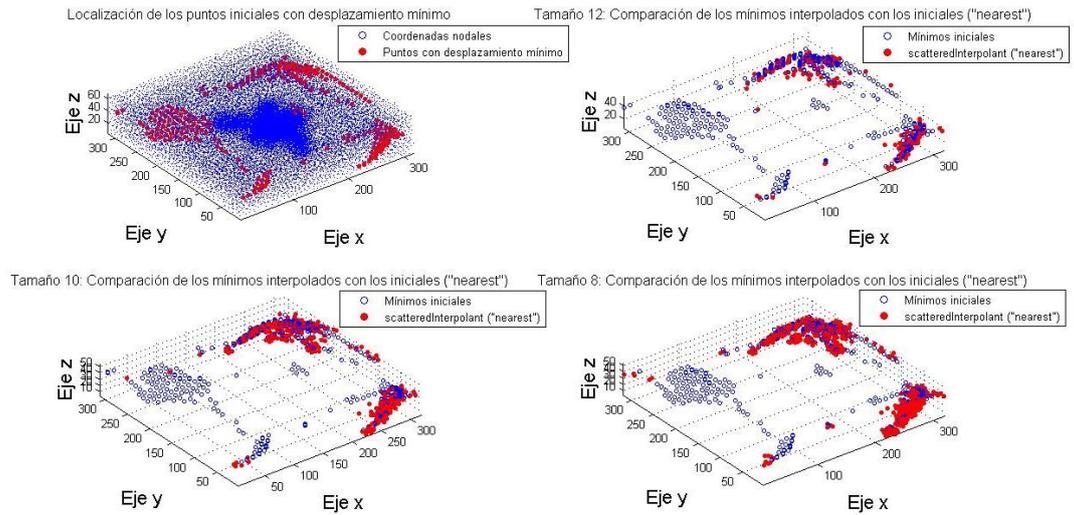


Figura A13. Comparación de los mínimos iniciales con los extrapolados para los diferentes tamaños de malla. (Método Nearest)

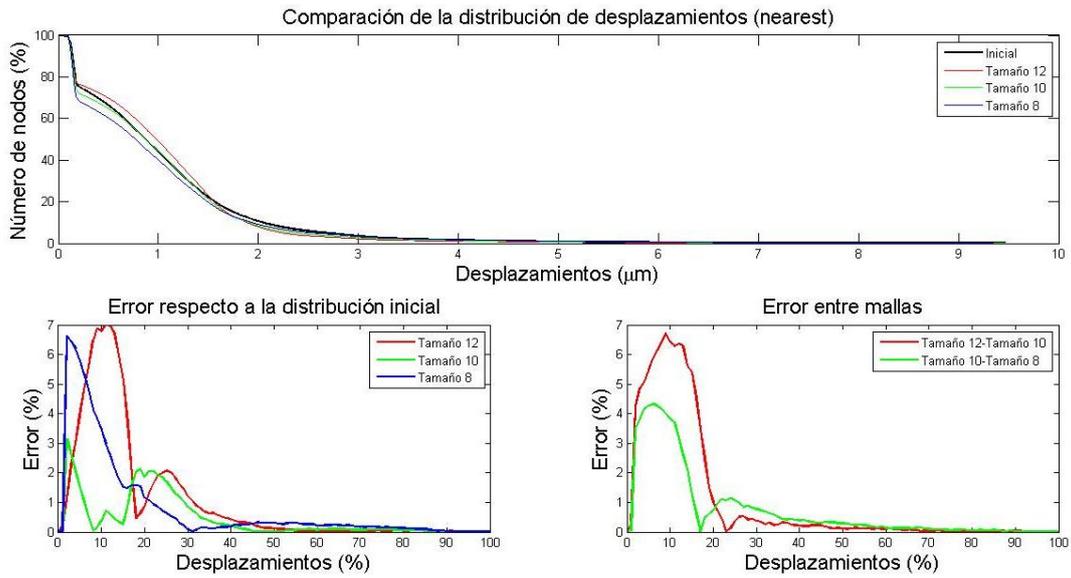


Figura A14. Función distribución de los puntos extrapolados para los diferentes tamaños de malla. (Método Nearest)

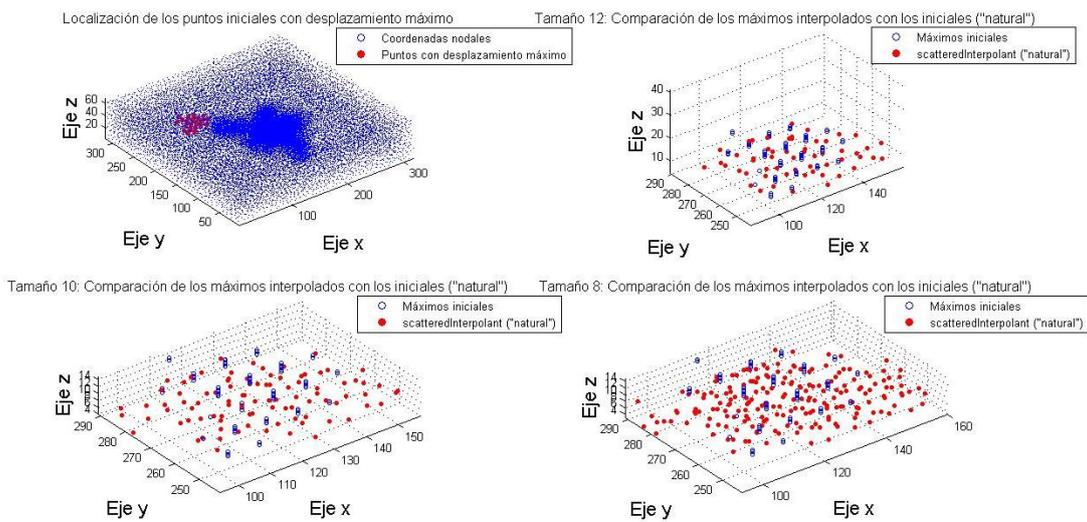


Figura A15. Comparación de los máximos iniciales con los extrapolados para los diferentes tamaños de malla. (Método Natural)

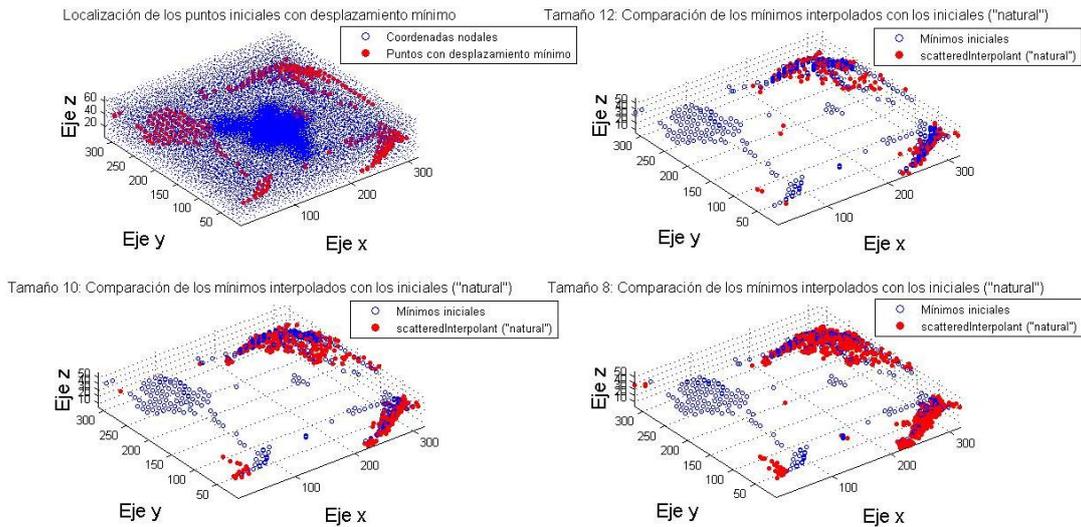


Figura A16. Comparación de los mínimos iniciales con los extrapolados para los diferentes tamaños de malla. (Método Natural)

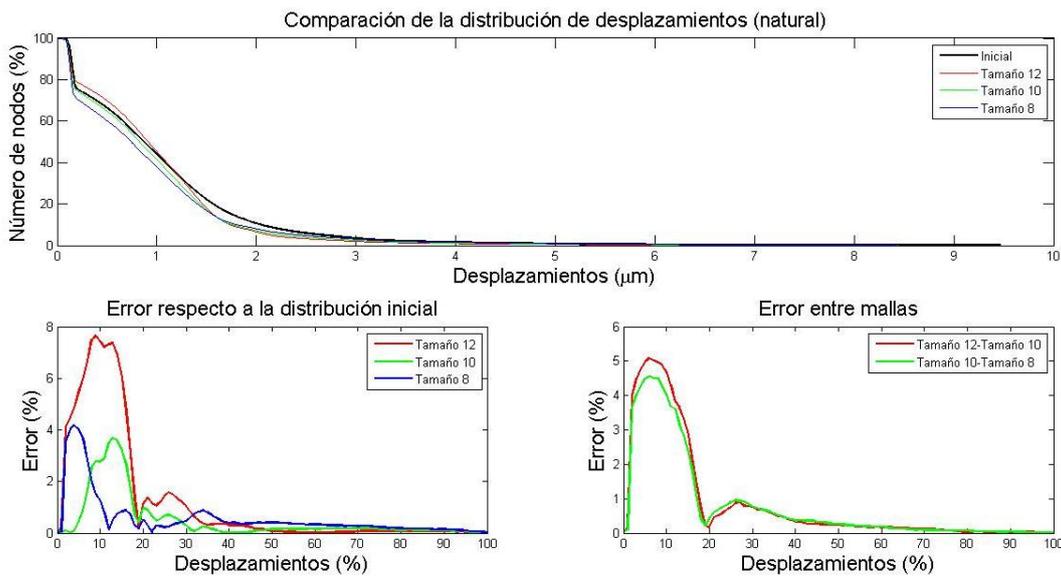


Figura A17. Función distribución de los puntos extrapolados para los diferentes tamaños de malla. (Método Natural)

Como se puede apreciar, la distribución de máximos y mínimos parece ser fiel a la original, y un poco más precisa la conseguida con el método "nearest". Además, en la función de distribución de desplazamientos, se puede ver que se llega a producir un error hasta del 8 % con respecto a la distribución del campo inicial en los puntos con desplazamiento más bajo para el caso más desfavorable. Aunque este error en un principio puede parecer demasiado alto, en la mayoría del dominio se estabiliza y toma un valor razonable del error. Tanto la distribución de

máximos y mínimos como la función de distribución creada hacen pensar que el modelo más preciso es el obtenido con el método nearest para el tamaño 10.

Si se observa bien en el campo inicial, parece que hay una zona de desplazamientos máximos que se pierde al extrapolar (Figura A18), esto, unido a los errores que se acaban de comentar, hace pensar que el método de extrapolación no es suficientemente preciso.

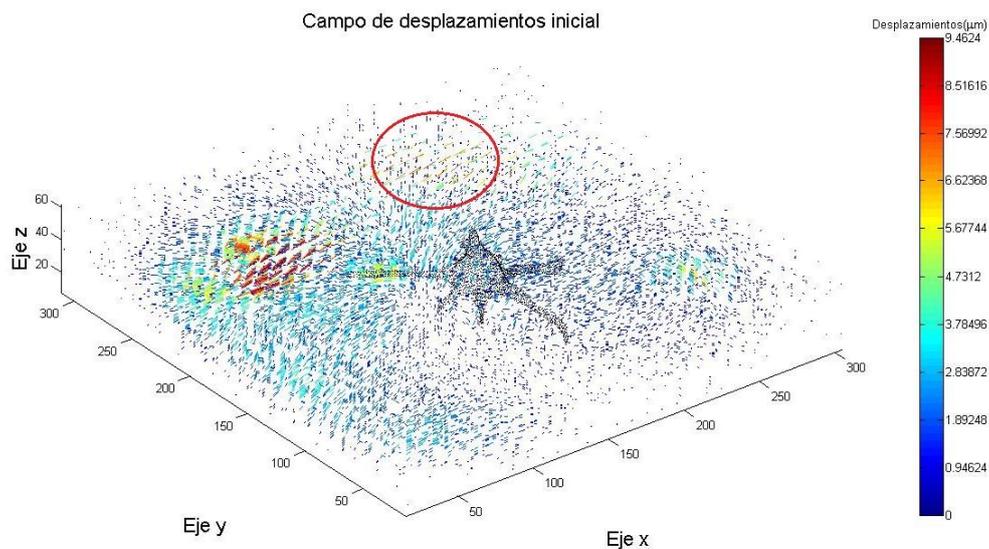


Figura A18. Zona problemática a la hora de extrapolar.

Para confirmar esta pérdida, se ha utilizado la orden *maximo* pero esta vez seleccionando los puntos que superan el 65% del valor máximo en vez del 85% (Figura A19). Los resultados obtenidos confirman que la zona señalada contiene valores mayores que el umbral en el campo inicial y que se están perdiendo al extrapolar:

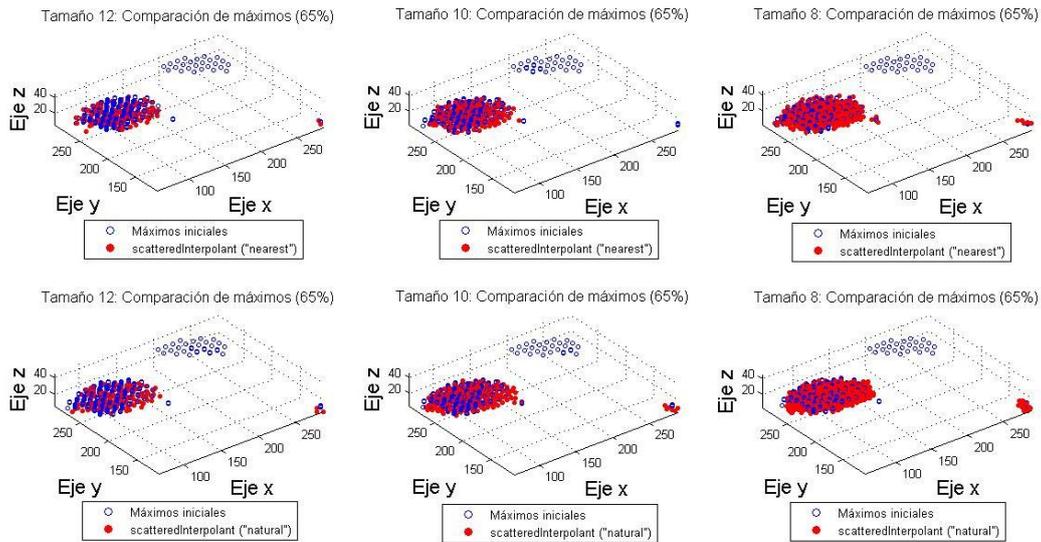


Figura A19. Comparación de los máximos extrapolados con los iniciales (65%). Arriba Método Nearest y abajo Método Natural.

Por este motivo, se ha implementado un método de "extrapolación local" que subdivide los campos de desplazamientos en cuatro zonas para que la función de desplazamientos sea más fácil de extrapolar:

```
function VFL = extrapLocal1_NN(points_i, desp_i, Coord, x_sep, y_sep)
%Realiza una extrapolación local dividiendo los datos iniciales y los
%puntos finales en cuatro submatrices de las mismas dimensiones para
%ganar más precisión ante funciones de desplazamientos complejas. En
%este caso, se utiliza el método 'nearest' para interpolar y
%extrapolar
%VARIABLES DE ENTRADA Y SALIDA:
%points_i=puntos iniciales desde donde se desea extrapolar los
%desplazamientos
%desp_i=desplazamientos correspondientes a los puntos iniciales
%Coord=Puntos finales a los cuales queremos extrapolar los datos
%x_sep=distancia en el eje X a la que queremos hacer la división
%y_sep=distancia en el eje Y a la que queremos hacer la división
%VFL=Matriz de salida que incluye los puntos y los desplazamientos
finales.

%Separa los puntos iniciales y finales. Hay que tener en cuenta que,
%en la orden 'separar' implementada anteriormente, se concatenaban las
%matrices 'points_i' y 'desp_i' en una sola matriz. De esta forma,
%cada punto irá siempre con su desplazamiento correspondiente y no
%cometeremos ningún error a la hora de extrapolar
[PDI1, PDI2, PDI3, PDI4]=separar(points_i, desp_i, x_sep, y_sep);
[C1, C2, C3, C4]=separar(Coord, Coord, x_sep, y_sep);
```

```

%Delimita las dimensiones de las submatrices y las zonas de
%intersección
[ cx, cy, cz ] = sepxyz (Coord);
maxx_c = max (cx);
minx_c = min (cx);
maxy_c = max (cy);
miny_c = min (cy);
maxz_c = max (cz);
minz_c = min (cz);
d = 5;
I12 = [ (x_sep-d), miny_c, minz_c; (x_sep+d), (y_sep-d), maxz_c ];
I13 = [ minx_c, (y_sep-d), minz_c; (x_sep-d), (y_sep+d), maxz_c ];
I24 = [ (x_sep+d), (y_sep-d), minz_c; maxx_c, (y_sep+d), maxz_c ];
I34 = [ (x_sep-d), (y_sep+d), minz_c; (x_sep+d), maxy_c, maxz_c ];
IT = [ (x_sep-d), (y_sep-d), minz_c; (x_sep+d), (y_sep+d), maxz_c ];

%Crea las variables necesarias para la extrapolación separando las
%matrices que previamente hemos concatenado
P1 = PDI1 (:, 1:3);
P2 = PDI2 (:, 1:3);
P3 = PDI3 (:, 1:3);
P4 = PDI4 (:, 1:3);
D1 = PDI1 (:, 4:6);
D2 = PDI2 (:, 4:6);
D3 = PDI3 (:, 4:6);
D4 = PDI4 (:, 4:6);
C1 = C1 (:, 1:3);
C2 = C2 (:, 1:3);
C3 = C3 (:, 1:3);
C4 = C4 (:, 1:3);

%Extrapolamos los desplazamientos finales con la orden 'extrapolate3D_NN'
%implementada previamente. Para probar otros métodos como el 'linear'
%o el 'natural' tan solo hemos de llamar a las órdenes ya creadas
%'extrapolate3D_LL' o 'extrapolate3D_NaN' respectivamente en vez de
%usar 'extrapolate3D_NN'.
vint1 = extrapolate3D_NN (P1, D1, C1);
vint2 = extrapolate3D_NN (P2, D2, C2);
vint3 = extrapolate3D_NN (P3, D3, C3);
vint4 = extrapolate3D_NN (P4, D4, C4);

%Concatena los puntos y los desplazamientos finales en una sola matriz
para poder trabajar con ellos
MF1 = [ C1, vint1 ];
MF2 = [ C2, vint2 ];
MF3 = [ C3, vint3 ];
MF4 = [ C4, vint4 ];
%Debido al solapado que hemos impuesto en la intersección para ganar
%precisión, es necesario ahora hacer la media en esta zona entre las
%interpolaciones realizadas en cada una de las submatrices. Para ello,
%con la orden 'contenido' determinamos qué puntos de cada submatriz se
%encuentran dentro de la zona de intersección y generamos una matriz
%para cada intersección con los datos referentes a cada una de las
%distintas submatrices

```

```

MI12_1=contenido(MF1,I12);
MI12_2=contenido(MF2,I12);
MI12=[MI12_1,MI12_2];
MI13_1=contenido(MF1,I13);
MI13_3=contenido(MF3,I13);
MI13=[MI13_1,MI13_3];
MI24_2=contenido(MF2,I24);
MI24_4=contenido(MF4,I24);
MI24=[MI24_2,MI24_4];
MI34_3=contenido(MF3,I34);
MI34_4=contenido(MF4,I34);
MI34=[MI34_3,MI34_4];
MIT_1=contenido(MF1,IT);
MIT_2=contenido(MF2,IT);
MIT_3=contenido(MF3,IT);
MIT_4=contenido(MF4,IT);
MIT=[MIT_1,MIT_2,MIT_3,MIT_4];

%Finalmente, realizamos la media de los desplazamientos finales en
%cada una de las intersecciones y los almacenamos en una matriz 'PDFi'
d12=zeros(size(MI12,1),3);
for i=1:size(MI12,1)
d12(i,:)=[(MI12(i,4)+MI12(i,10))/2],[(MI12(i,5)+MI12(i,11))/2],[(MI12
(i,6)+MI12(i,12))/2];
end
PDF12=[MI12(:,1:3),d12];

d13=zeros(size(MI13,1),3);
for i=1:size(MI13,1)
d13(i,:)=[(MI13(i,4)+MI13(i,10))/2],[(MI13(i,5)+MI13(i,11))/2],[(MI13
(i,6)+MI13(i,12))/2];
end
PDF13=[MI13(:,1:3),d13];

d24=zeros(size(MI24,1),3);
for i=1:size(MI24,1)
d24(i,:)=[(MI24(i,4)+MI24(i,10))/2],[(MI24(i,5)+MI24(i,11))/2],[(MI24
(i,6)+MI24(i,12))/2];
end
PDF24=[MI24(:,1:3),d24];
d34=zeros(size(MI34,1),3);
for i=1:size(MI34,1)
d34(i,:)=[(MI34(i,4)+MI34(i,10))/2],[(MI34(i,5)+MI34(i,11))/2],[(MI34
(i,6)+MI34(i,12))/2];
end
PDF34=[MI34(:,1:3),d34];

dT1=zeros(size(MIT,1),1);
dT2=zeros(size(MIT,1),1);
dT3=zeros(size(MIT,1),1);
for i=1:size(MIT,1)
dT1(i,:)=(MIT(i,4)+MIT(i,10)+MIT(i,16)+MIT(i,22))/4;
dT2(i,:)=(MIT(i,5)+MIT(i,11)+MIT(i,17)+MIT(i,23))/4;
dT3(i,:)=(MIT(i,6)+MIT(i,12)+MIT(i,18)+MIT(i,24))/4;
dT=[dT1,dT2,dT3];
end
PDFT=[MIT(:,1:3),dT];

```

```

%Una vez interpolados y mediados todos los desplazamientos, pasamos a
%reconstruir el campo final de desplazamientos. Éste se dividirá en 9
%zonas distintas: las cinco zonas de intersección existentes y las
%cuatro zonas de los extremos. En las zonas de los extremos (A1, A2,
%A3 y A4) los desplazamientos serán los interpolados inicialmente,
%mentras que en las zonas de intersección (I12, I13, I24, I34,IT) los
%desplazamientos serán las medias de las diferentes submatrices, no
%los interpolados simplemente.
A1=[minx_c,miny_c,minz_c;(x_sep-d),(y_sep-d),maxz_c];
A2=[(x_sep+d),miny_c,minz_c;maxx_c,(y_sep-d),maxz_c];
A3=[minx_c,(y_sep+d),minz_c;(x_sep-d),maxy_c,maxz_c];
A4=[(x_sep+d),(y_sep+d),minz_c;maxx_c,maxy_c,maxz_c];

MA1=contenido(MF1,A1);
MA2=contenido(MF2,A2);
MA3=contenido(MF3,A3);
MA4=contenido(MF4,A4);

%El último detalle a tener en cuenta, es que tanto la orden 'separar'
%como la orden 'contenido' utilizadas para generar las diferentes
%submatrices necesarias, utilizan el criterio 'menor o igual' y 'mayor
%o igual', es decir, al delimitar una zona, incluye los puntos del
%extremo. Por lo tanto, hemos generado puntos duplicados
%correspondientes a cada una de las zonas de intersección entre las
%submatrices. Para solucionar este problema, utilizamos la orden
%'eliminaIguales' para eliminar los puntos duplicados.
PDFT=eliminaIguales(PDFT,PDF12);
PDFT=eliminaIguales(PDFT,PDF13);
PDFT=eliminaIguales(PDFT,PDF24);
PDFT=eliminaIguales(PDFT,PDF34);
PDF12=eliminaIguales(PDF12,MA1);
PDF12=eliminaIguales(PDF12,MA2);
PDF13=eliminaIguales(PDF13,MA1);
PDF13=eliminaIguales(PDF13,MA3);
PDF24=eliminaIguales(PDF24,MA2);
PDF24=eliminaIguales(PDF24,MA4);
PDF34=eliminaIguales(PDF34,MA3);
PDF34=eliminaIguales(PDF34,MA4);
%Para terminar, concatenamos todas las submatrices ahora que son
%correctas en una sola.
VFL=[MA1;PDF12;MA2;PDF13;PDFT;PDF24;MA3;PDF34;MA4];
end

```

Se han implementado por tanto dos métodos distintos de extrapolación local. Uno que subdivide en cuatro zonas tanto los puntos iniciales como los finales y otro que tan solo subdivide en cuatro zonas los puntos iniciales para ver si afecta a los resultados conseguidos. Debido a la gran similitud entre los resultados obtenidos, se ha optado por comentar solo el primero de los dos métodos.

Como se explica en la descripción de la función, se separan las matrices introducidas por las distancias "*x_sep*" e "*y_sep*" definidas por el usuario. En este caso esas distancias son 170 micras tanto para "*x_sep*" como para "*y_sep*" (Figura A20).

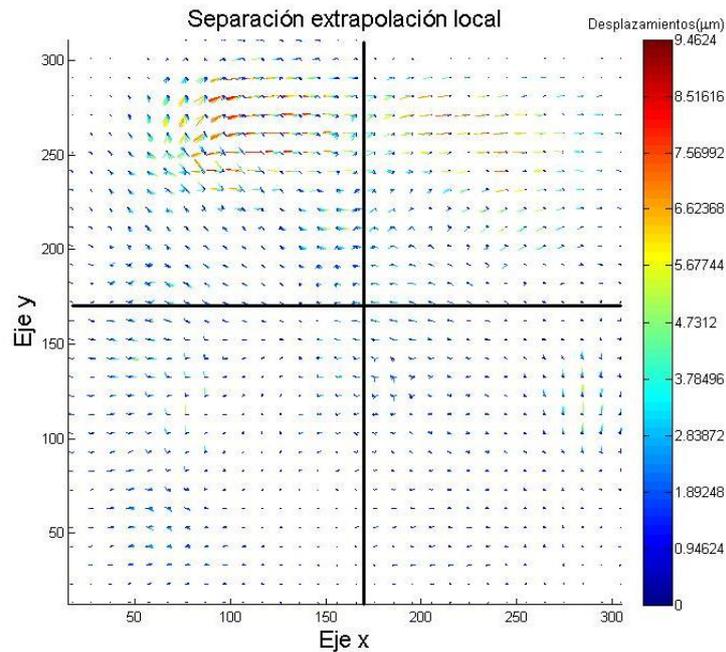


Figura A20. Subdivisiones realizadas para la función de extrapolación local.

Además, en las zonas de intersección se genera un pequeño desfase para que se solapen las diferentes submatrices y así poder hacer la media aritmética en la unión para mejorar la continuidad de la función. Por lo tanto, realmente las matrices quedarán divididas en nueve submatrices como se muestra en la siguiente imagen (Figura A21):

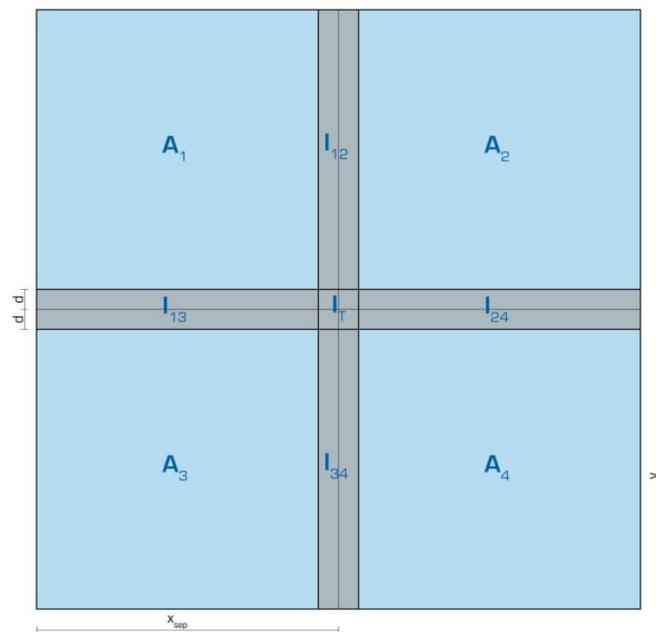


Figura A21. Subdivisiones realizadas realmente por el método local teniendo en cuenta el desfase aplicado.

Donde las diferentes submatrices creadas abarcan un poco más del área que deberían para poder proceder de la manera comentada. Ese área extra se regula con la variable "desfase" introducida en la función de Matlab (Figura A22).

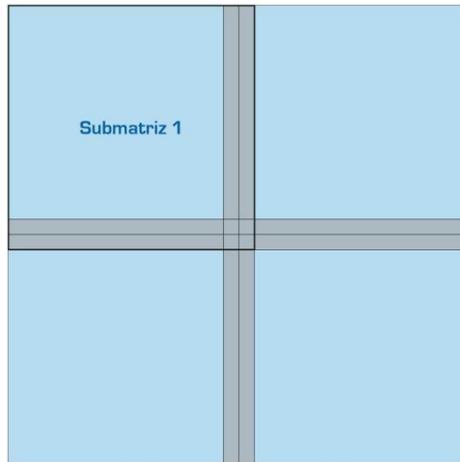


Figura A22. Detalle del funcionamiento del método local.

Para el funcionamiento de esta función, ha sido necesario implementar tres órdenes distintas: *contenido*, *eliminaIguales* y *separar*, a continuación se adjunta el código de su implementación:

```
function P = contenido(M,ref)
%Determina qué puntos de un matriz inicial se encuentran contenidos en
%un volumen 'ref'
% VARIABLES DE ENTRADA Y SALIDA:
% M=Matriz de puntos a estudiar
% ref=Volumen de referencia donde queremos localizar los puntos
%iniciales contenidos
% P=listado de puntos pertenecientes a 'M' y contenidos en 'ref'
%Determina los valores máximos y mínimos de 'ref' en cada dirección
[x,y,z]=sepxyz(ref);
maxcx_ref=max(x);
mincx_ref=min(x);
maxcy_ref=max(y);
mincy_ref=min(y);
maxcz_ref=max(z);
mincz_ref=min(z);
[f]=size(M,1);
j=1;
```

```

%Comprueba para cada punto de 'M' si sus coordenadas x,y,z son menores
%que el valor máximo de 'ref' y mayores que su valor mínimo para
%coordenada respectivamente y, si es así, copia el punto en una matriz
%auxiliar, 'aux'
for i=1:f
    if ((mincx_ref<=M(i,1))&&(M(i,1)<=maxcx_ref))
        if ((mincy_ref<=M(i,2))&&(M(i,2)<=maxcy_ref))
            if ((mincz_ref<=M(i,3))&&(M(i,3)<=maxcz_ref))
                aux(j,:)=M(i,:);
                j=j+1;
            end
        end
    end
end
end
%Crea la variable de salida
P=aux;
end

```

```

function [ML1,ML2,ML3,ML4] = separar(points_i,desp_i,x_sep,y_sep)
%Concatena las matrices 'points_i' y 'desp_i' en una matriz 'M' y
%subdivide esta matriz 'M' en cuatro zonas en función de las
%coordenadas de 'points_i' y las distancias 'x_sep' e 'y_sep' a las
%que queremos separar la matriz
% VARIABLES DE ENTRADA:
% points_i=Matriz de coordenadas a separar
% desp_i=Desplazamientos correspondientes a la matriz de coordenadas a
%separar
%x_sep=distancia en el eje X a la que queremos hacer la división
%y_sep=distancia en el eje Y a la que queremos hacer la división

%Concatena las matrices iniciales y establece los limites para la
%separación
M=[points_i,desp_i];
[cx,cy,cz]=sepxyz(points_i);
maxx_c=max(cx);
minx_c=min(cx);
maxy_c=max(cy);
miny_c=min(cy);
maxz_c=max(cz);
minz_c=min(cz);
%Define los cuatro dominios en los que se separará la matriz. Para una
%mayor precisión en la interpolación local posterior, solapamos la
%unión con un desfase de 5 unidades para poder hacer la media en la
%zona de intersección y obtener una continuidad mejor en nuestros
%resultados
desfase=5;
ref_1=[minx_c, miny_c, minz_c; (x_sep+desfase), (y_sep+desfase),
maxz_c];
ref_2=[(x_sep-desfase), miny_c, minz_c; maxx_c, (y_sep+desfase),
maxz_c];
ref_3=[minx_c, (y_sep-desfase), minz_c; (x_sep+desfase), maxy_c,
maxz_c];
ref_4=[(x_sep-desfase), (y_sep-desfase), minz_c; maxx_c, maxy_c,
maxz_c];

```

```

%Definición de las variables de salida
ML1=contenido (M,ref_1);
ML2=contenido (M,ref_2);
ML3=contenido (M,ref_3);
ML4=contenido (M,ref_4);
end

function M = eliminaIguales(matriz_borrar,M_2)
%Compara dos matrices y elimina de 'matriz_borrar' los elementos que
%también pertenezcan a 'M_2'
% VARIABLES DE ENTRADA Y SALIDA:
% matriz_borrar=Matriz a comparar con 'M_2' de la que se eliminarán
los
% valores coincidentes
% M_2=Matriz de referencia
% M=Matriz actualizada sin los elementos coincidentes

%Crea las variables y constantes necesarias para el programa
fmb=size(matriz_borrar,1);
fm2=size(M_2,1);
aux1=matriz_borrar;
aux2=ones(1,6);
% k=0;

%Compara los puntos de ambas matrices individualmente, si las
coordenadas
%x,y,z coinciden en ambos, elimina esa fila de 'aux1' (que es igual a
'matriz_borrar')
for i=1:fmb
    for j=1:fm2
        if
            (matriz_borrar(i,1)==M_2(j,1)) && (matriz_borrar(i,2)==M_2(j,2)) && (matriz_borrar(i,3)==M_2(j,3))
                break
            elseif j~=fm2
                continue
            end
            aux2=[aux2;aux1(i,:)];
        end
    end
end
%Crea la variable de salida
M=aux2(2:end,:);
end

```

Siguiendo estos criterios, aplicando el método *extrapLocal* se obtienen los siguientes resultados (Figuras A23-A28):

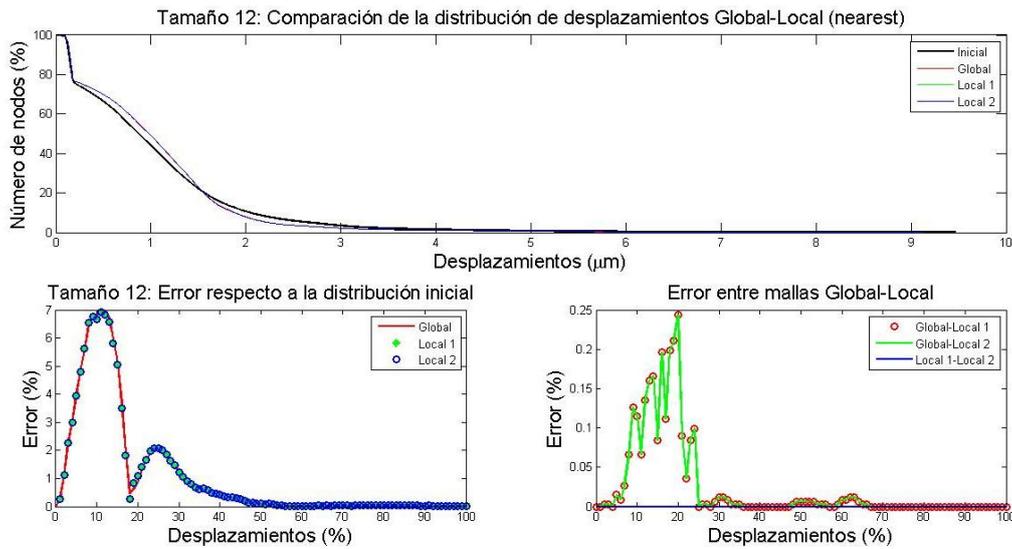


Figura A23. Comparación de las distribuciones de puntos entre el método global y el local para tamaño de malla 12. (Método Nearest)

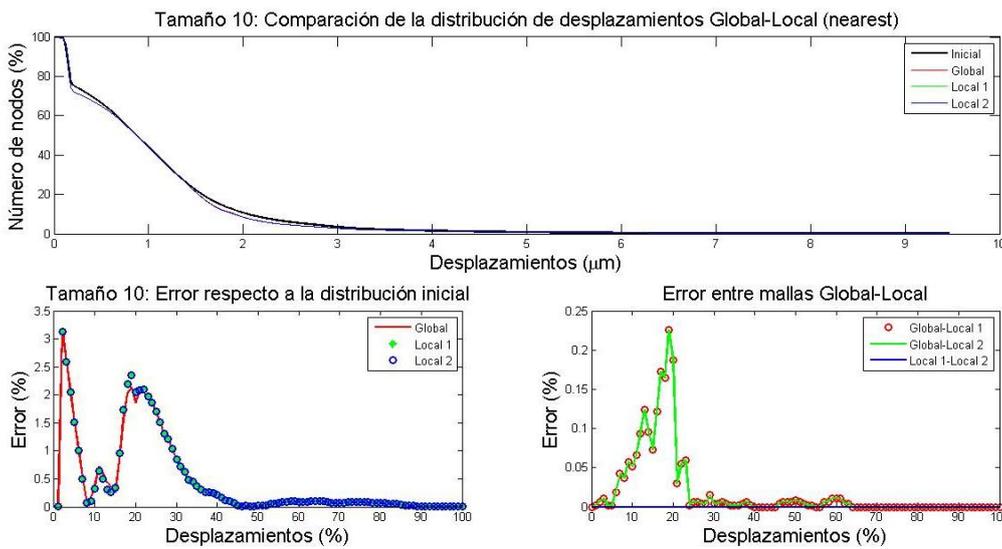


Figura A24. Comparación de las distribuciones de puntos entre el método global y el local para tamaño de malla 10. (Método Nearest)

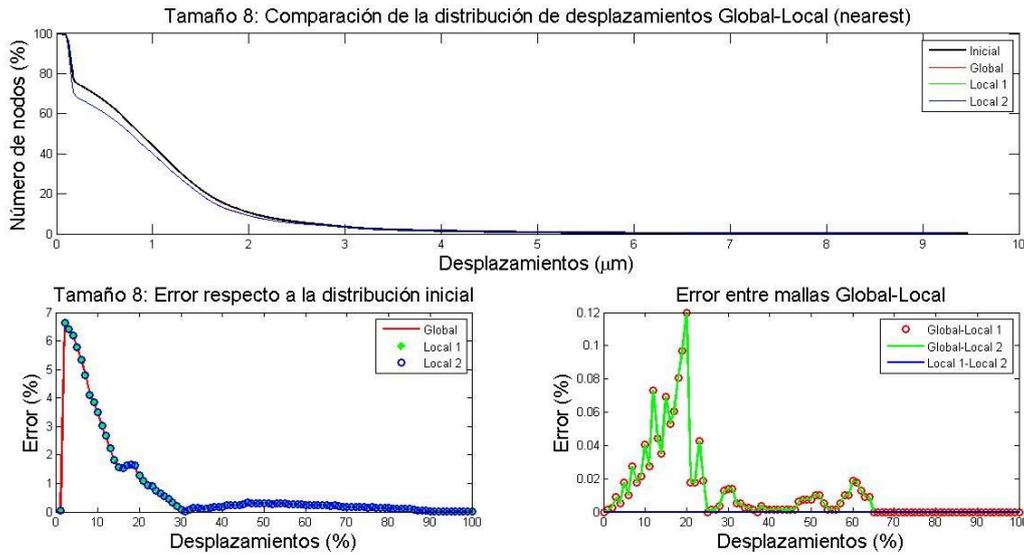


Figura A25. Comparación de las distribuciones de puntos entre el método global y el local para tamaño de malla 8. (Método Nearest)

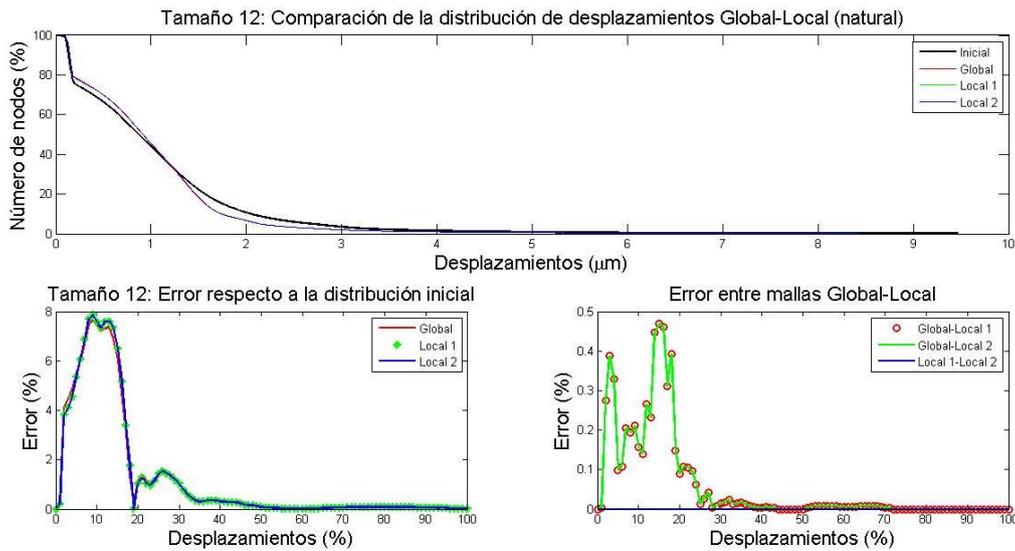


Figura A26. Comparación de las distribuciones de puntos entre el método global y el local para tamaño de malla 12. (Método Natural)

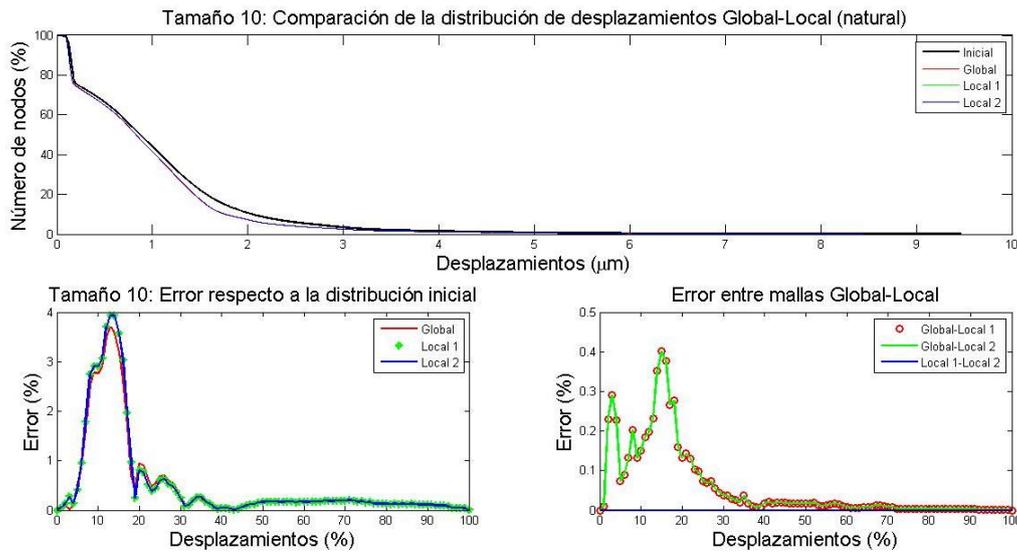


Figura A27. Comparación de las distribuciones de puntos entre el método global y el local para tamaño de malla 10. (Método Natural)

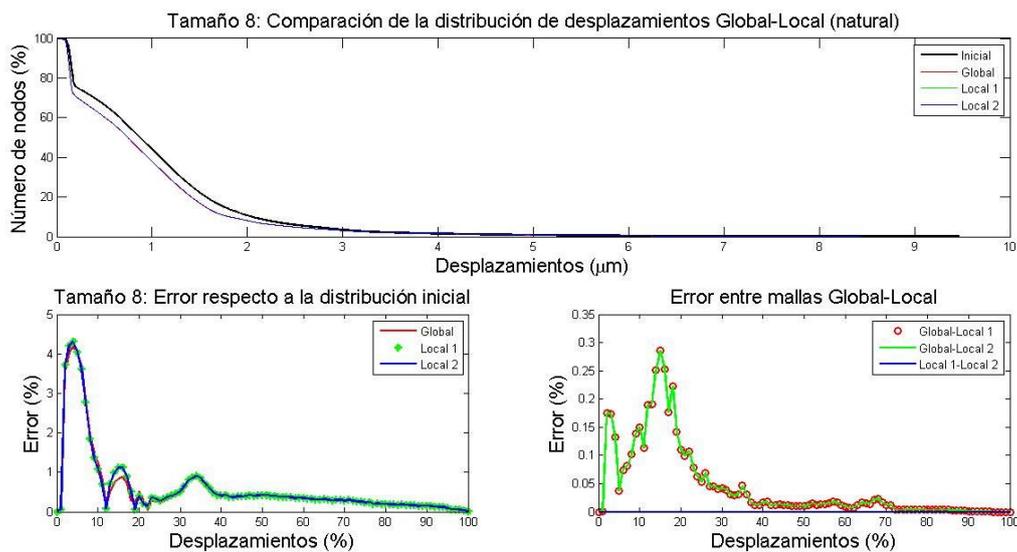


Figura A28. Comparación de las distribuciones de puntos entre el método global y el local para tamaño de malla 8. (Método Natural)

Se ha decidido incluir solamente las funciones de distribución como método de contraste debido a la gran similitud de resultados entre el método global y el local. En la primera gráfica de cada imagen se muestra la comparación entre el campo inicial, el campo extrapolado con el método global y el obtenido con el método local. La coincidencia entre resultados es tal, que las gráficas del método global y la del local se solapan en una sola línea. En las dos gráficas de la parte inferior de cada imagen, se muestra el error que comete cada método con respecto inicial y el error entre el método global y el local respectivamente. El máximo error entre métodos es de un 0.5% en el caso más desfavorable.

Salta a la vista que el método local es prácticamente igual al global y que la zona que se perdía del campo inicial no logra extrapolarse correctamente ni siquiera con este método. Sin embargo, si se seleccionan solamente los puntos iniciales máximos de la zona problemática en vez de en todo el dominio (Figura A30), se observa lo siguiente:

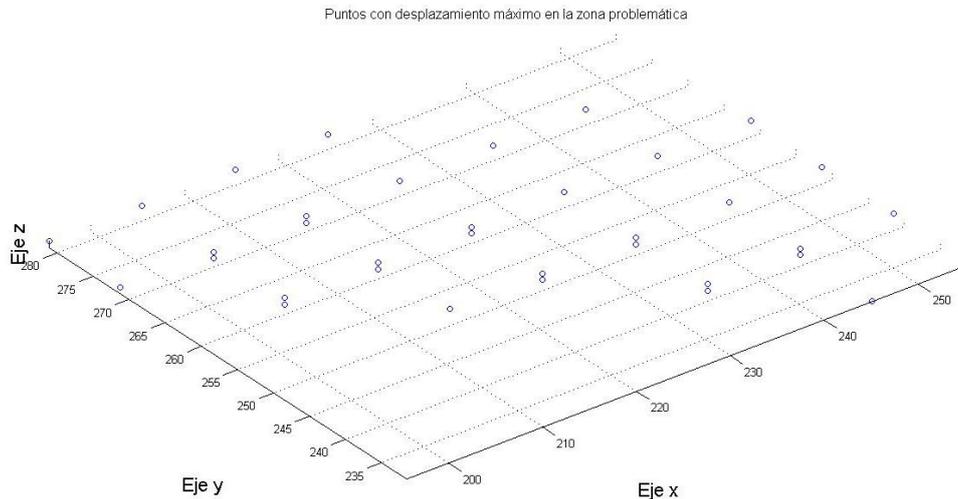


Figura A29. Detalle de los puntos con desplazamiento máximo en la zona problemática.

Pese a que en las imágenes anteriores parecía que la zona problemática era un volumen significativo, al reducir el volumen representado se puede observar que esta zona es un conjunto muy pequeño de puntos y que además se encuentran prácticamente en el mismo plano. Debido a la gran cantidad de puntos que tiene el campo de desplazamientos inicial, estos puntos se pierden al extrapolar a la malla.

Ya que se trata de un conjunto tan pequeño y con una geometría tan plana, se considera que no son relevantes con respecto al campo global y que la extrapolación es suficientemente buena.

Teniendo en cuenta todos estos datos, podemos concluir que:

- los métodos "nearest" y "natural" generan modelos que se aproximan adecuadamente al original.
- El método local no produce ninguna mejoría significativa con respecto al global y, además, tiene un mayor coste computacional.
- Pese a algún problema local, se considera que todos los métodos no descartados tienen una precisión suficientemente buena y que pueden aportar una solución válida.
- El método "nearest" es el que obtiene unos resultados más realistas ya que la amplitud de desplazamientos obtenida es la más cercana a la inicial y la localización de sus valores máximos y mínimos es también la más fiel con respecto al modelo experimental.

- El tamaño de malla que obtiene unos valores más fieles a los originales es el tamaño 10 ya que su función de distribución es la que más se asemeja a la del modelo inicial.
- Por lo tanto, para la extrapolación de desplazamientos se utilizará la orden **scatteredInterpolant** con el método "**nearest neighbour interpolation**" y un tamaño de malla de **10 micras**.

Tras haber definido el método de extrapolación, es necesario crear un último programa que genere un fichero de entrada con el campo de desplazamientos a aplicar al modelo y en el formato requerido por Abaqus. Para ello, se ha programado la función *inpAbaqus*:

```
function inpAbaqus(numNodos, desp_inp, nombreArchivoSalida)
%Crea un archivo con el formato necesario para introducir los
%desplazamientos en el fichero *.inp de Abaqus.
% VARIABLES DE ENTRADA:
% numNodos=nodos correspondientes a los desplazamientos
% desp_inp=desplazamientos nodales en las coordenadas de la malla
% nombreArchivo=nombre deseado para el fichero de salida *.txt

%Comprueba si las dimensiones de las matrices de entrada coinciden
[f]=size(desp_inp,1);
Ln=length(numNodos);
if Ln==f
    %Da el formato requerido donde, si u_nodo1=(ux,uy,uz), crea:
    %nodo_1 1 1 ux
    %nodo_1 2 2 uy
    %nodo_1 3 3 uz
    %Y así sucesivamente con todos los nodos
    aux=zeros(3*Ln,4);
    for i=1:f
        j=(i-1)*3+1;
        aux(j,:)= [numNodos(i),1,1,desp_inp(i,1)];
        aux((j+1),:)= [numNodos(i),2,2,desp_inp(i,2)];
        aux((j+2),:)= [numNodos(i),3,3,desp_inp(i,3)];
    end
    %Crea el fichero de salida en el que cada línea tiene la forma:
    %NombreModelo.NombreSetNodos.nodo_i,1,1,ux_i
    %NombreModelo.NombreSetNodos.nodo_i,2,2,uy_i
    %NombreModelo.NombreSetNodos.nodo_i,3,3,uz_i
    %Para este caso, el nombre deseado del modelo es:
    %'GLOBAL_10_3_GEL_EXTRA-1.Nodos_Solo_Gel.'
    ABAQ=aux';
    fileID = fopen([nombreArchivoSalida '.txt'],'w');

    fprintf(fileID, 'GLOBAL_10_3_GEL_EXTRA1.Nodos_Solo_Gel.%d,%d,%d,%.4f\n',
    ABAQ);
        fclose(fileID);
        disp('El fichero se ha escrito correctamente.')
else
    disp('Las dimensiones de las matrices no coinciden.')
end
end
```

Anexo B. Documentación gráfica de la distribución de las variables estudiadas en el apartado de resultados

B.1. Caso I: Simulaciones para el modelo de célula relajada con tripsina

B.1.1. Efecto de la variación de la rigidez del citoplasma

Para optimizar el espacio, se ha representado en cada figura los resultados de todas las variaciones de un mismo parámetro. Por ejemplo, en las Figuras B1-B4 se recogen tanto los parámetros obtenidos para la primera versión (todos los materiales hiperelásticos) de la simulación en la que se reduce en un orden de magnitud el módulo elástico del citoplasma (a la izquierda) como las del caso en el que se aumenta esta misma magnitud en un orden de magnitud (a la derecha). En el centro de cada una de las imágenes aparecerá siempre el modelo base con las propiedades sin variar para que sea más fácil comparar visualmente los resultados.

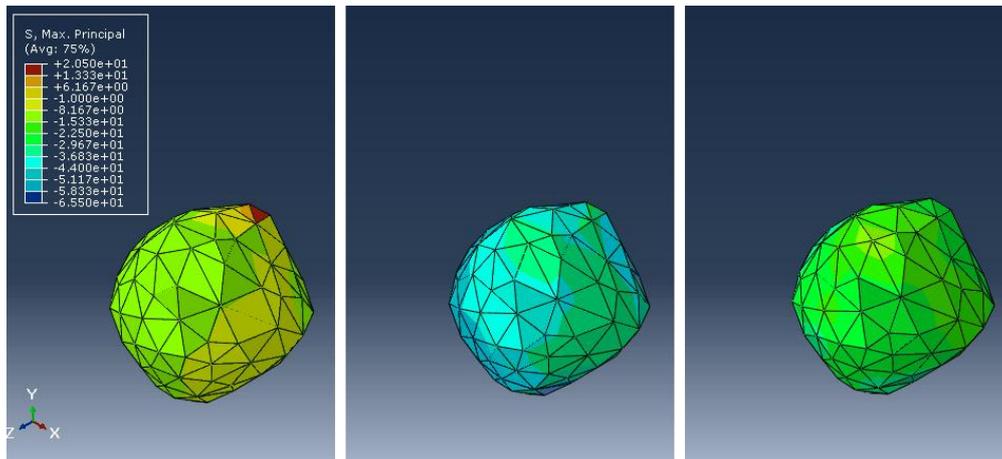


Figura B1. Tensión Máxima Principal con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

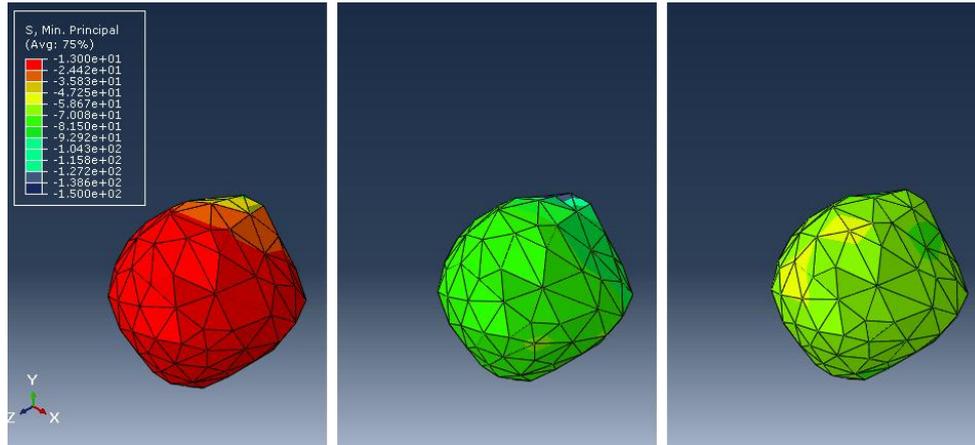


Figura B2. Tensión Mínima Principal con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

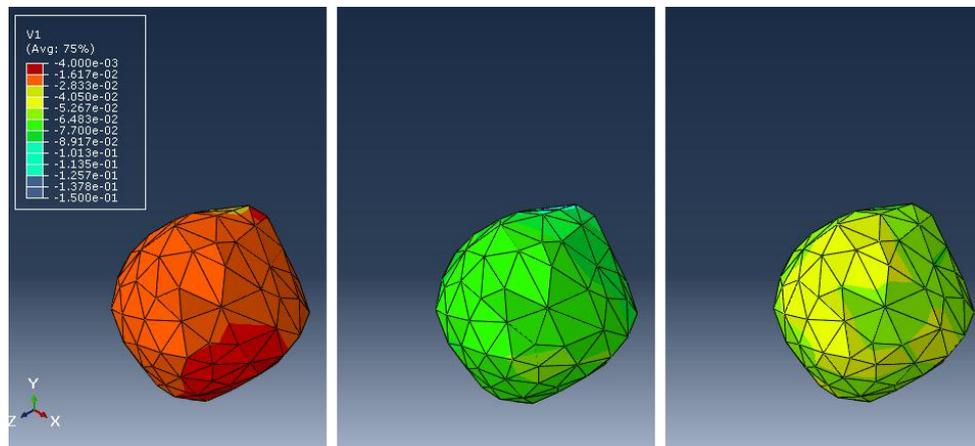


Figura B3. Variación volumétrica con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

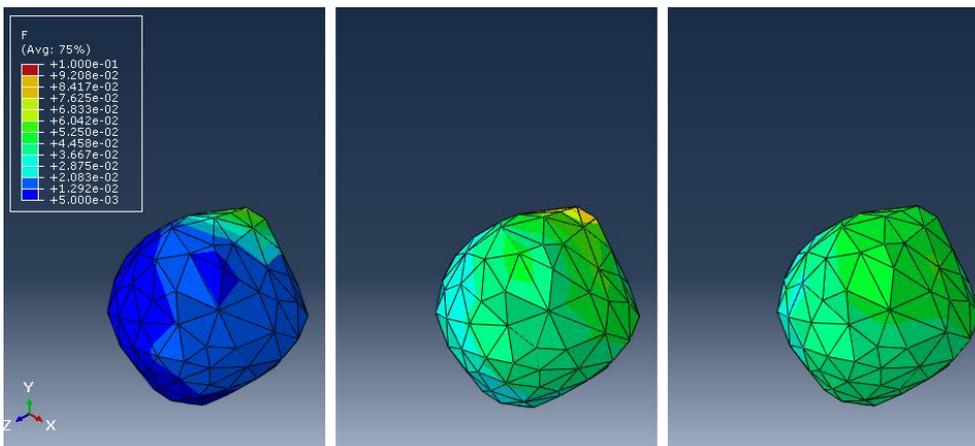


Figura B4. Variación de forma con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

Con la segunda versión planteada (donde el material del gel se considera elástico lineal y el resto de materiales permanecen hiperelásticos), se obtienen los resultados mostrados en las Figuras B5-B8.

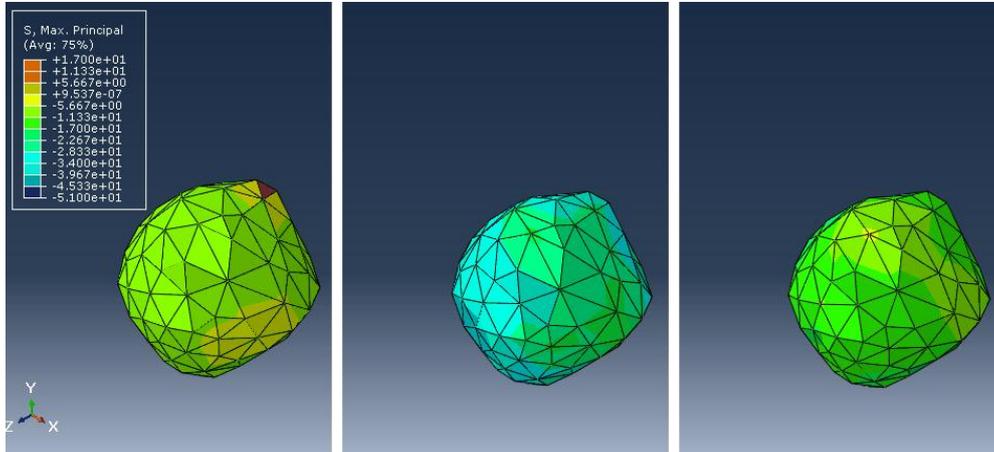


Figura B5. Tensión Máxima Principal con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

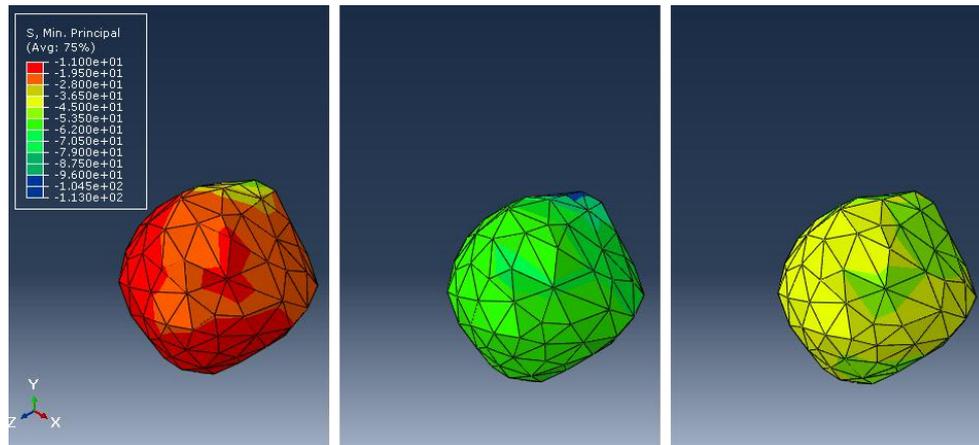


Figura B6. Tensión Mínima Principal con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

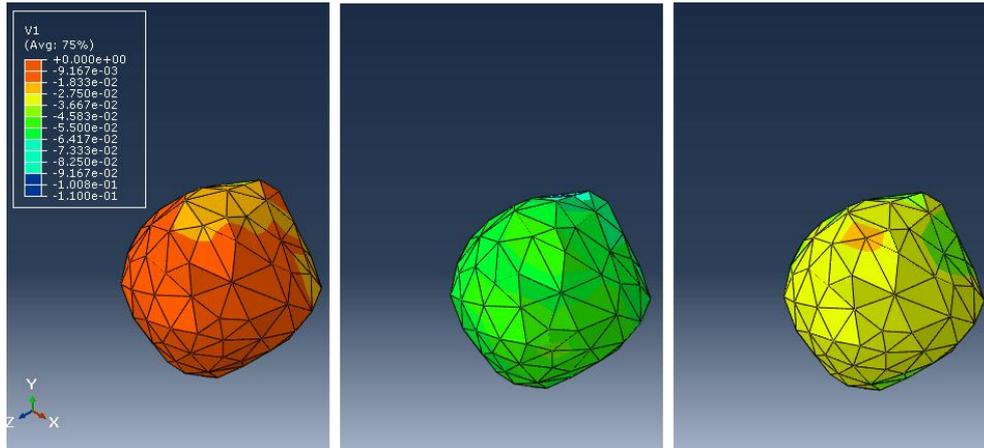


Figura B7. Variación volumétrica con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

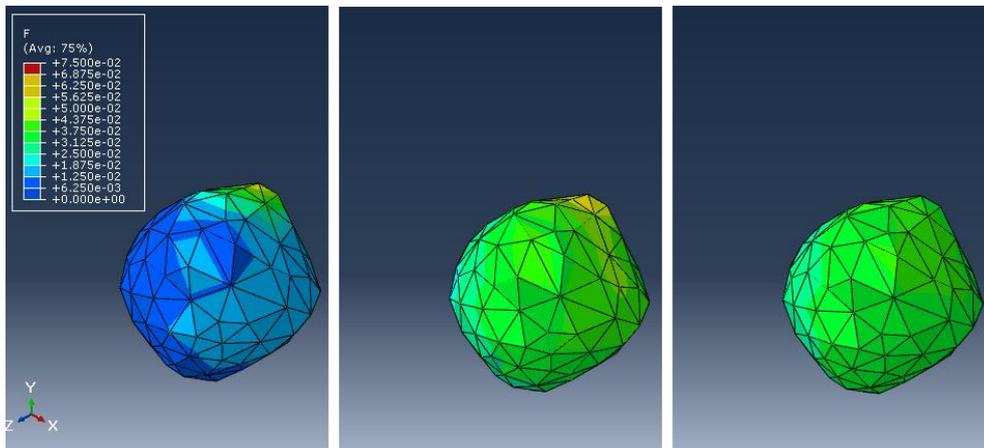


Figura B8. Variación de forma con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

Como se ha comentado en la memoria, la variación de los resultados obtenidos no parece guardar una relación de proporcionalidad con respecto al módulo elástico. Puesto que no hay ninguna evolución uniforme en cada grupo de imágenes.

Si se compara la Versión 1 con la Versión 2, se observa que las distribuciones obtenidas son prácticamente iguales para cada variable y, por lo general, la Versión 2 obtiene valores un poco inferiores a la Versión 1.

B.1.2. Efecto de la variación de la rigidez del núcleo

Una vez estudiada la influencia de la rigidez del citoplasma, se procede a estudiar la del núcleo. Procediendo de la misma manera que en el apartado anterior, se han obtenido los resultados mostrados en las Figuras B9-B12 para la primera versión.

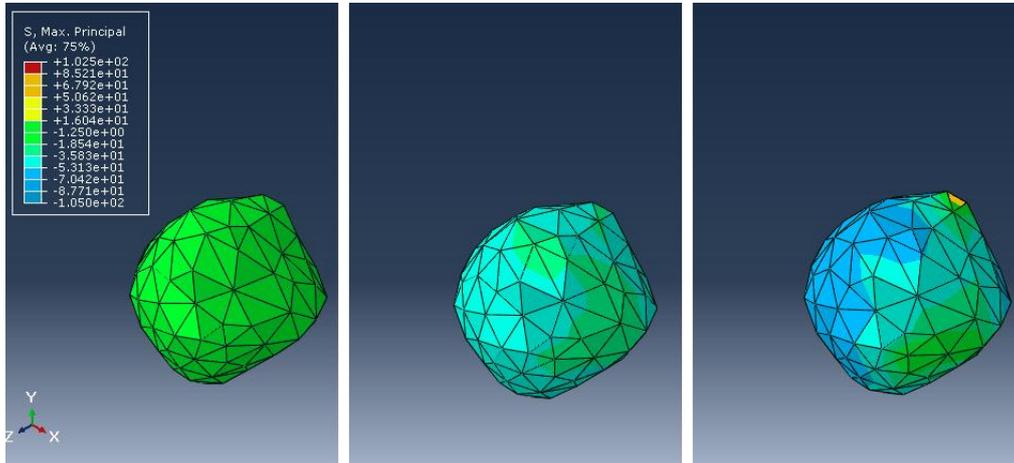


Figura B9. Tensión Máxima Principal con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

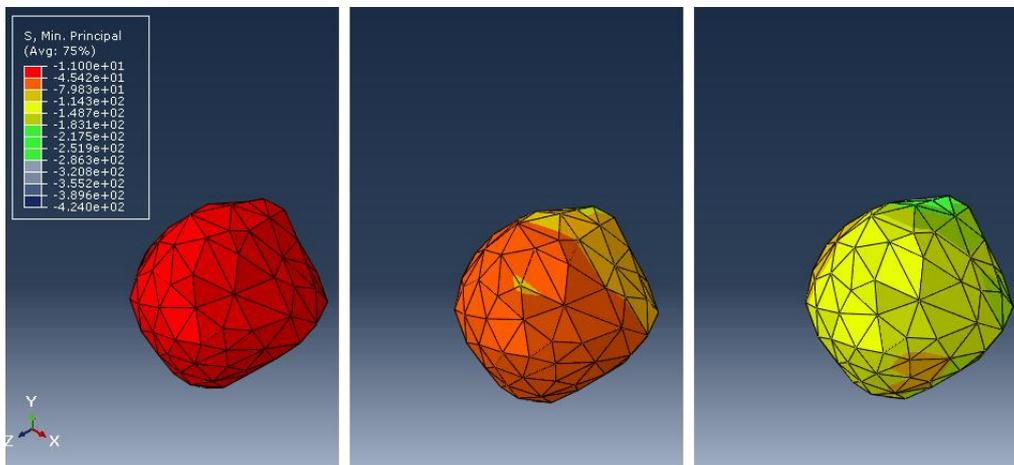


Figura B10. Tensión Mínima Principal con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

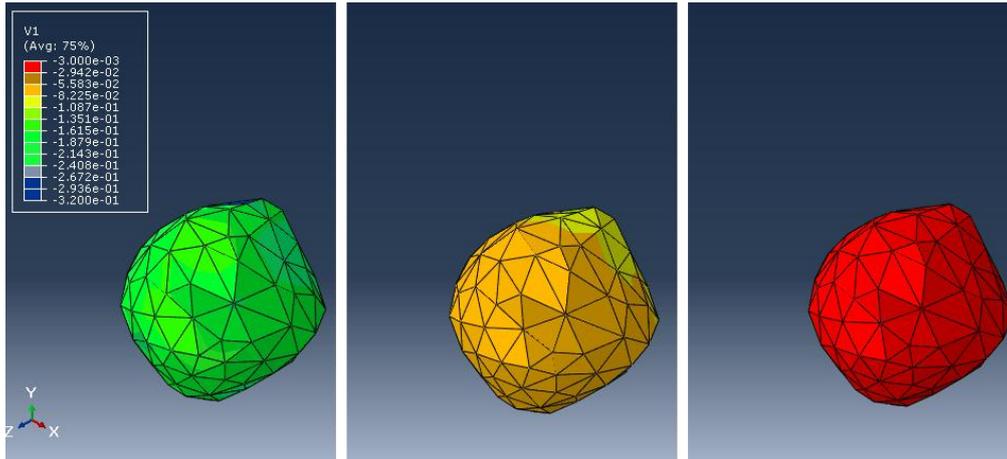


Figura B11. Variación volumétrica con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

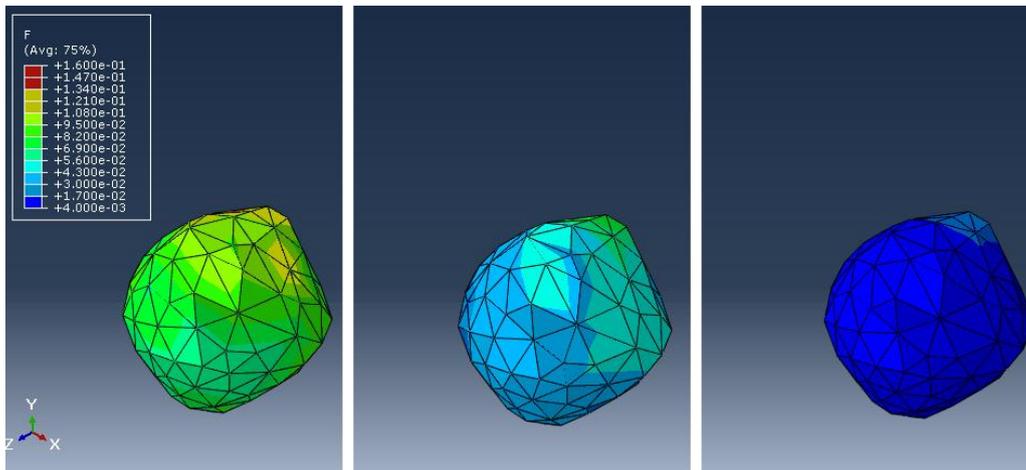


Figura B12. Variación de forma con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

En comparación con estos resultados, con la segunda versión del modelo se obtienen los siguientes resultados (Figuras B13-B16).

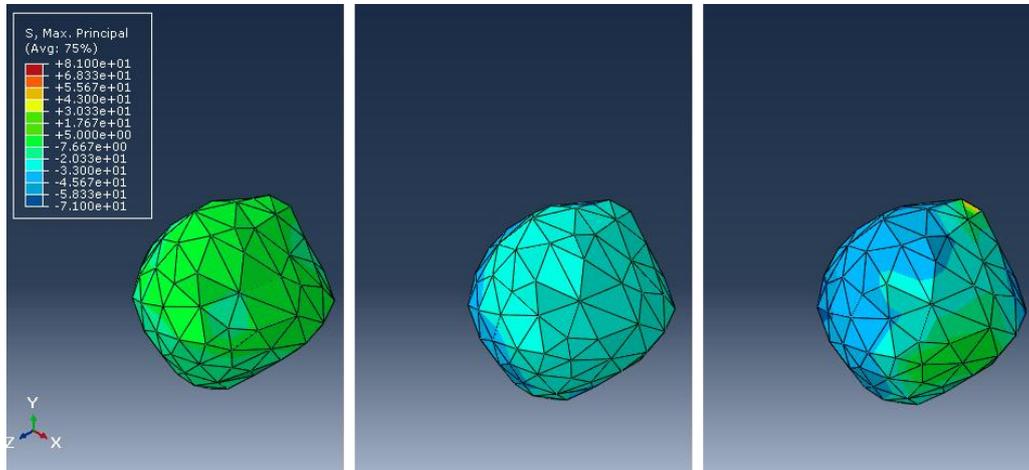


Figura B13. Tensión Máxima Principal con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

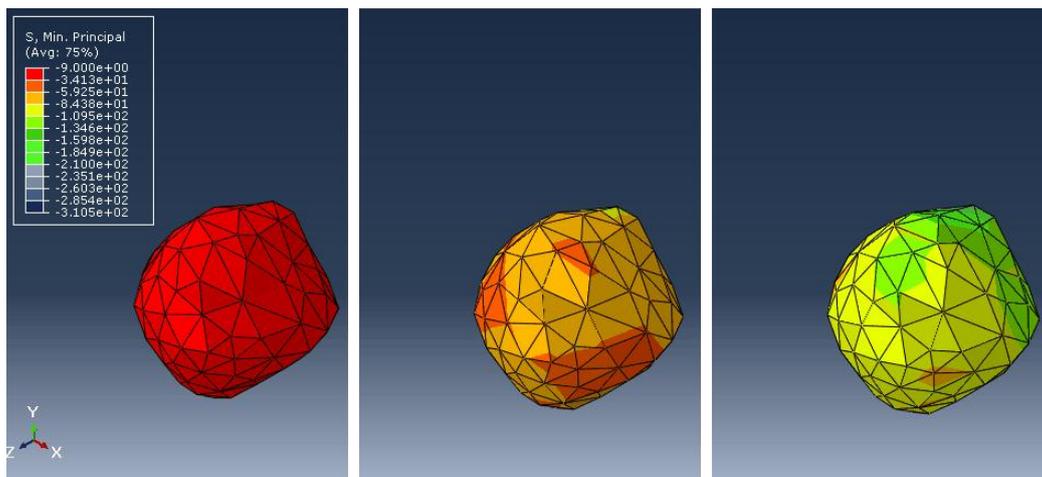


Figura B14. Tensión Mínima Principal con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

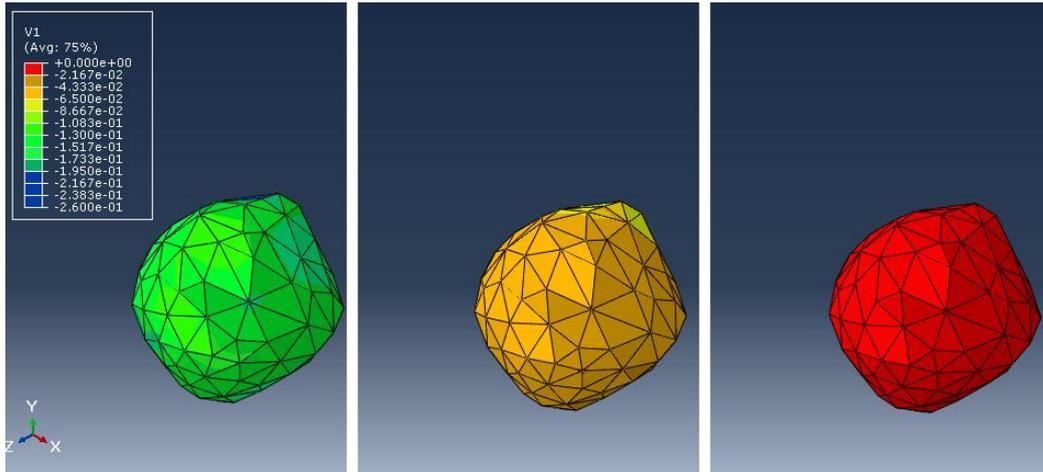


Figura B15. Variación volumétrica con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

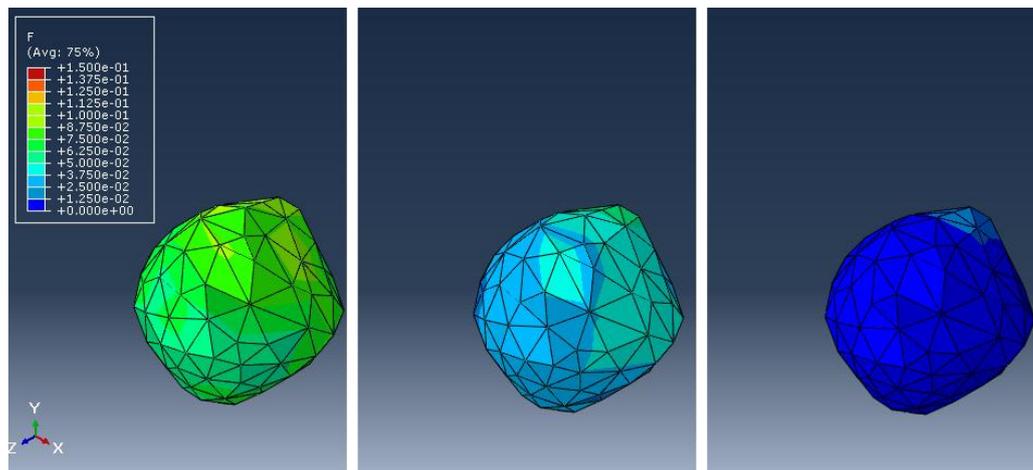


Figura B16. Variación de forma con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

En este caso, las distribuciones varían bastante con el cambio del módulo elástico. Conforme aumentamos el módulo elástico las tensiones y la variación de forma se ven reducidas. La variación de volumen aumenta con el módulo elástico.

Las distribuciones obtenidas con la versión 1 son prácticamente idénticas a las que se obtienen con la versión 2.

B.1.3. Efecto de la variación de la rigidez tanto del citoplasma como del núcleo

Por último, se ha probado a variar los dos parámetros a la vez para observar que efectos produce. Resultados obtenidos para la primera (Figuras B17-B20) y la segunda versión (Figuras B21-B24):

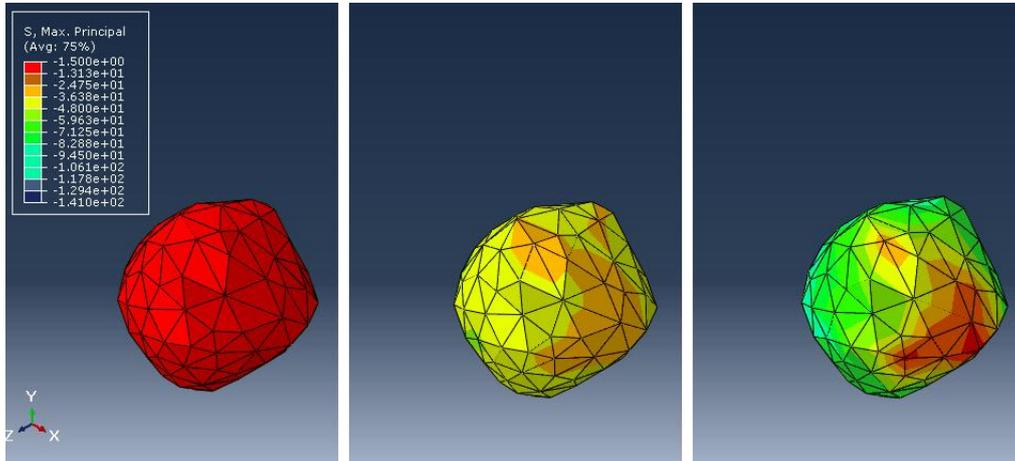


Figura B17. Tensión Máxima Principal con respecto a la variación del módulo elástico tanto del citoplasma como del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

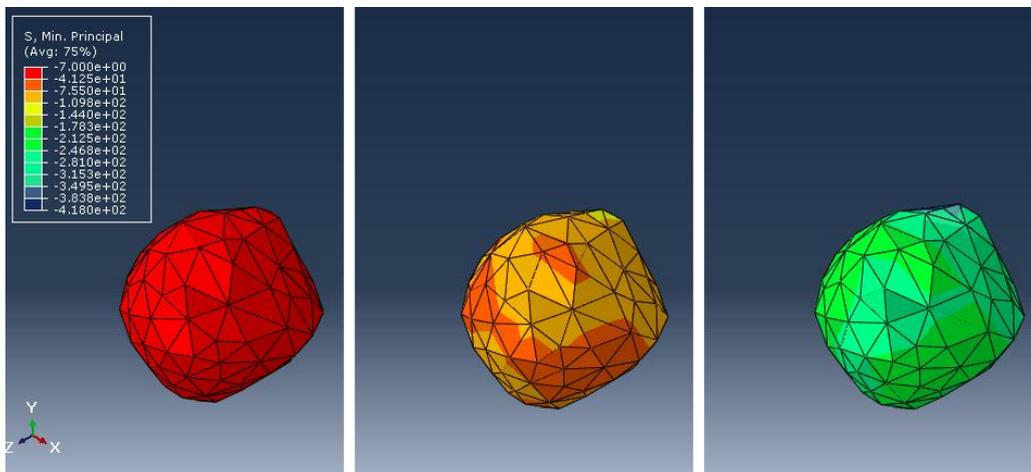


Figura B18. Tensión Mínima Principal con respecto a la variación tanto del módulo elástico del citoplasma como del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

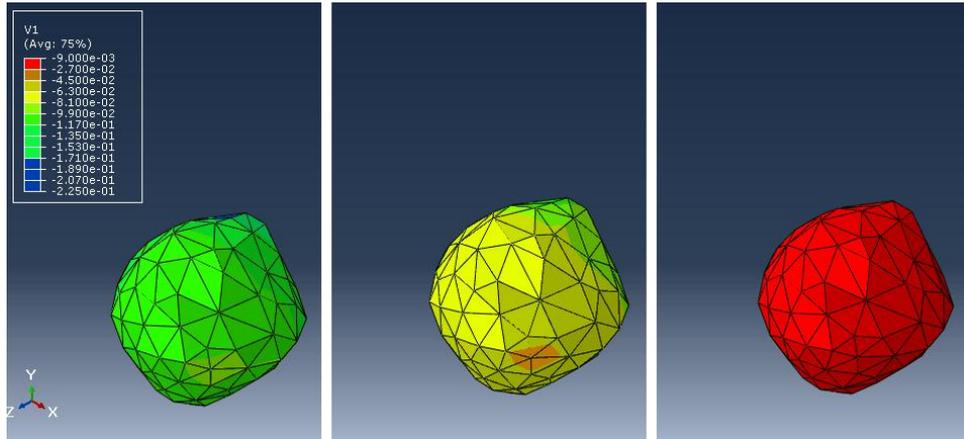


Figura B19. Variación volumétrica con respecto a la variación tanto del módulo elástico del citoplasma como del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

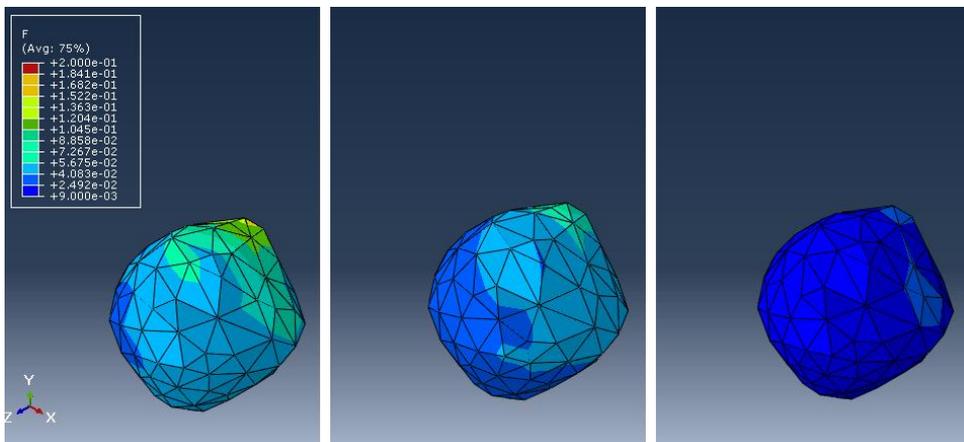


Figura B20. Variación de forma con respecto a la variación tanto del módulo elástico del citoplasma como del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

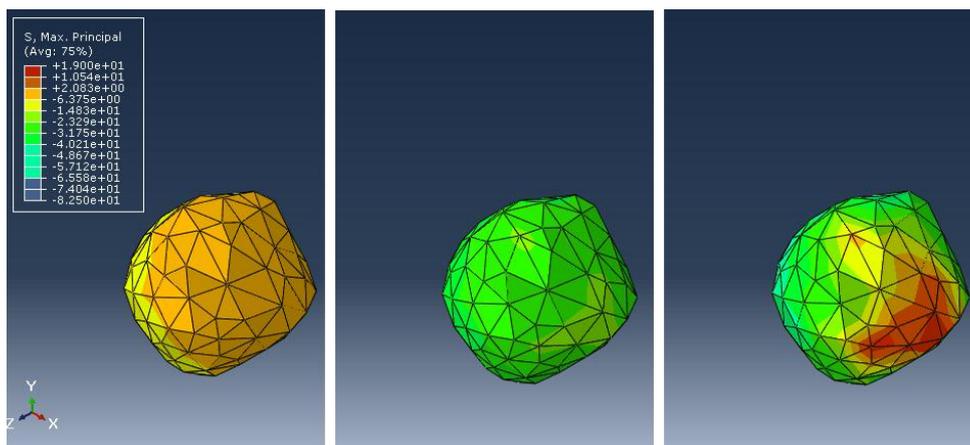


Figura B21. Tensión Máxima Principal con respecto a la variación tanto del módulo elástico del núcleo como del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

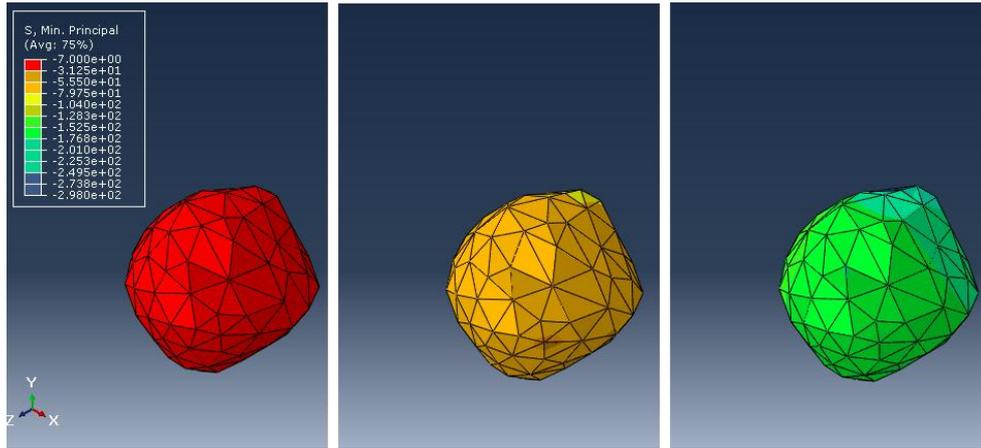


Figura B22. Tensión Mínima Principal con respecto a la variación tanto del módulo elástico del núcleo como del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

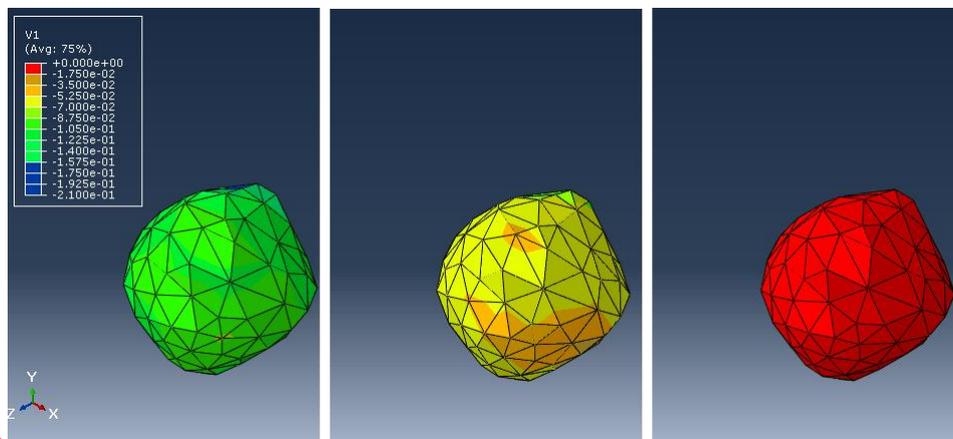


Figura B23. Variación volumétrica con respecto a la variación tanto del módulo elástico del núcleo como del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

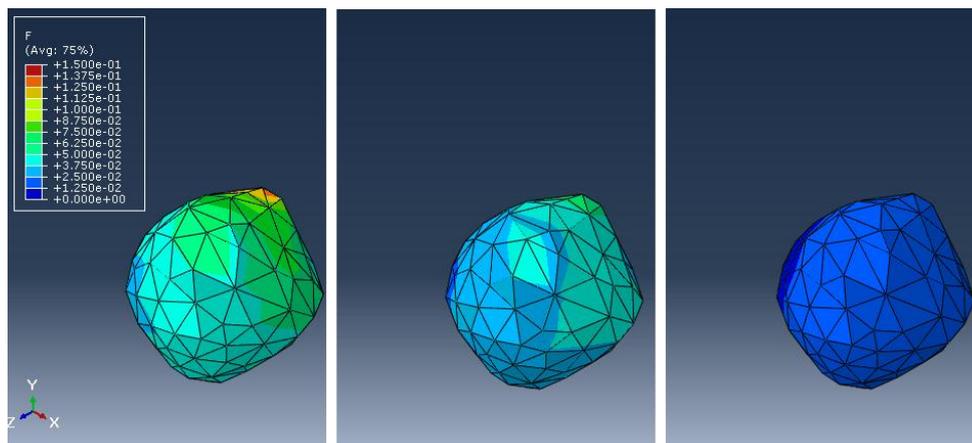


Figura B24. Variación de forma con respecto a la variación tanto del módulo elástico del núcleo como del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

La evolución de las diferentes distribuciones es muy similar a la obtenida en el caso anterior para casi todas las variables. La variable que más se diferencia es la Tensión Principal Máxima, que parece tener una variación más intensa en este caso que en que solo se variaba el módulo elástico del núcleo.

B.2. Caso II: Simulaciones para el modelo de célula relajada con citocalasina

A continuación, se presentan los resultados de las simulaciones para el caso de la célula relajada con citocalasina. El criterio para representar las figuras es el mismo que en el Caso I, se ha mostrado en cada una de ellas tres elementos siendo el central el modelo base con las propiedades intactas.

Dado que no hay ninguna diferencia significativa entre el modelo de la célula relajada con tripsina y la de citocalasina, las distribuciones obtenidas se comportan de una forma muy similar a la comentada en el Caso I. Exceptuando algún detalle como el que se ha comentado en la Memoria.

B.2.1. Efecto de la variación de la rigidez del citoplasma

Resultados de la primera (Figuras B25-B28) y segunda versión (Figuras B29-B32).

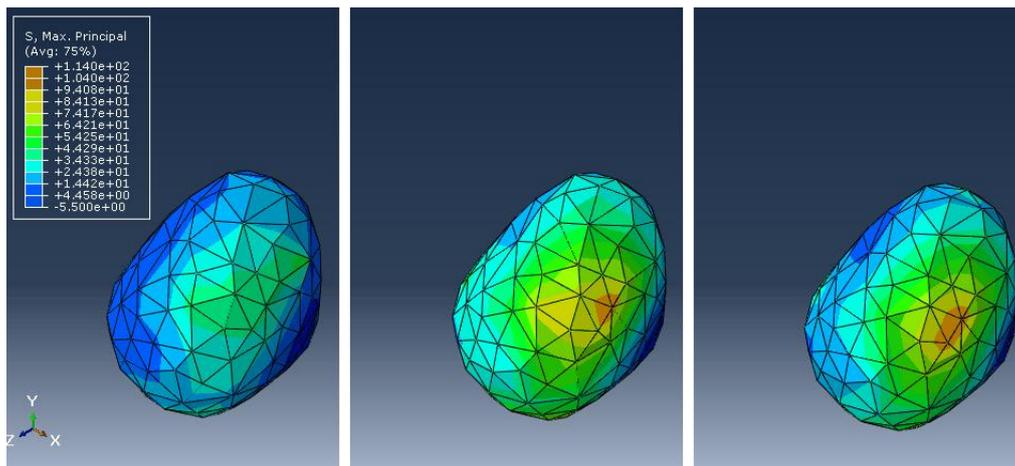


Figura B25. Tensión Máxima Principal con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

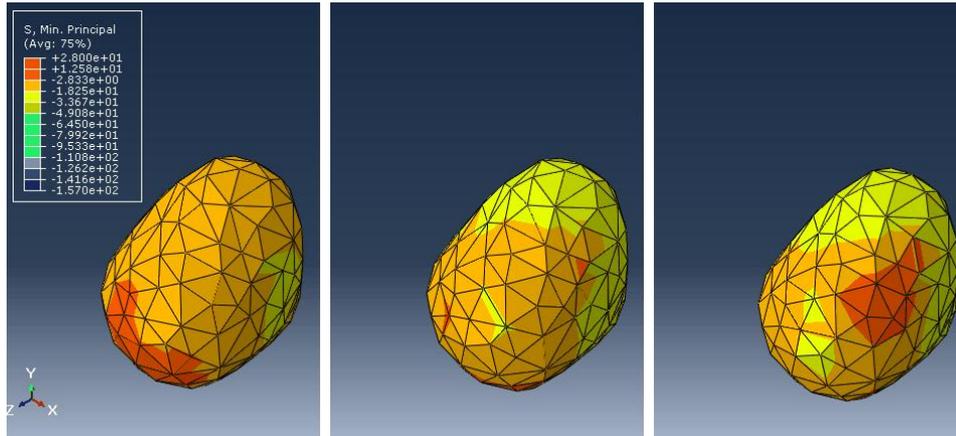


Figura B26. Tensión Mínima Principal con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

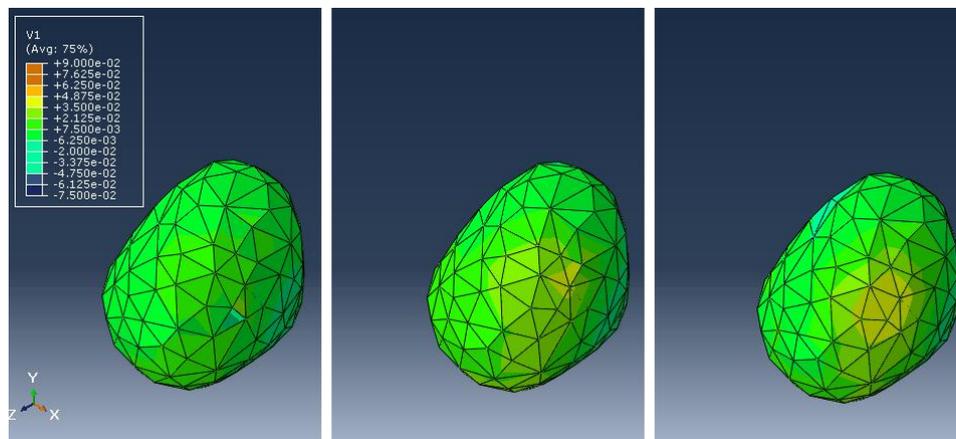


Figura B27. Variación volumétrica con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

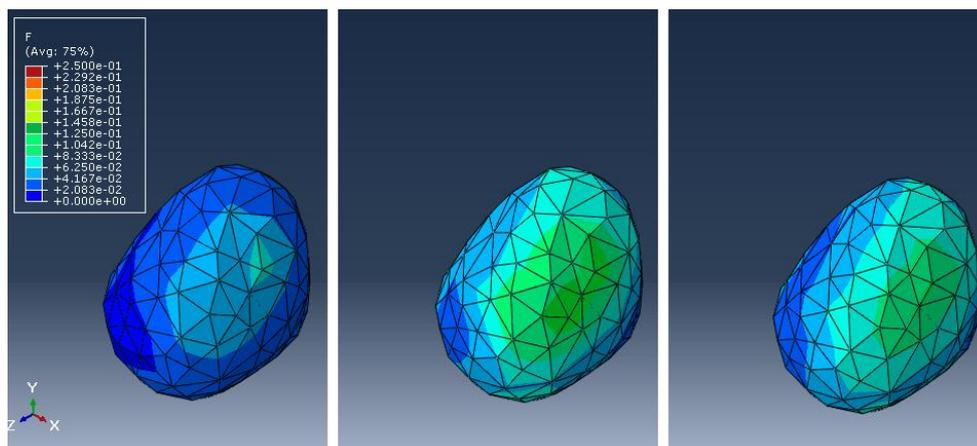


Figura B28. Variación de forma con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

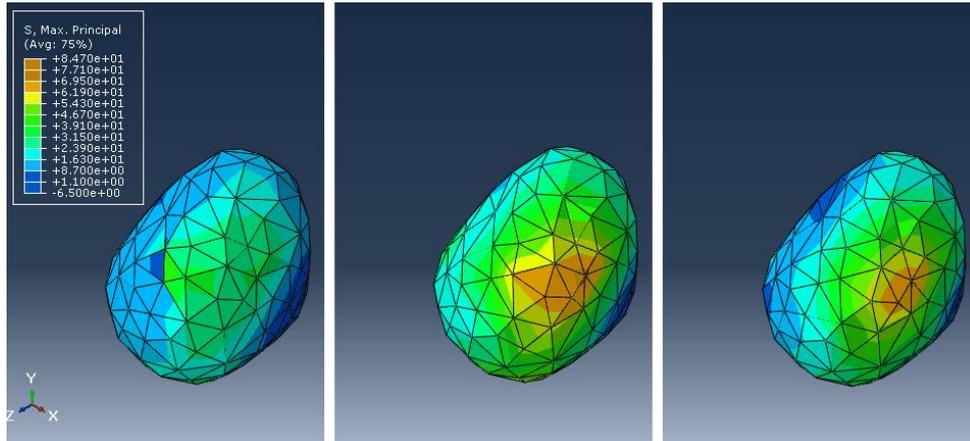


Figura B29. Tensión Máxima Principal con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

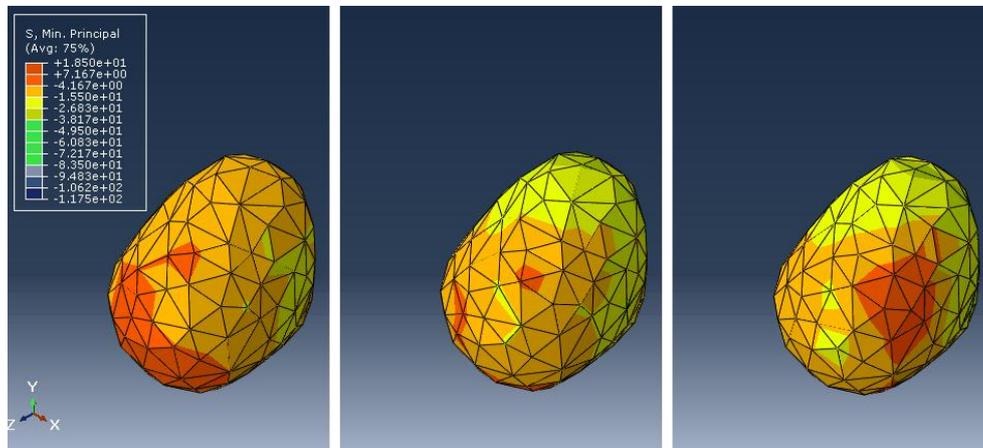


Figura B30. Tensión Mínima Principal con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

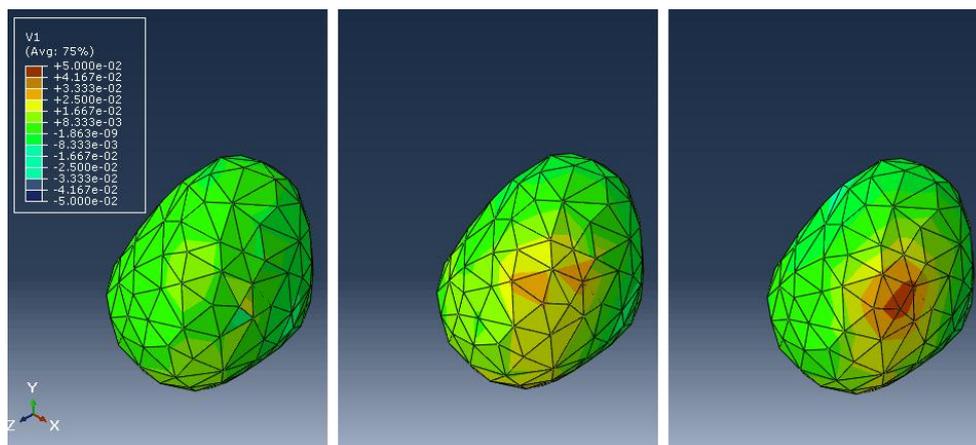


Figura B31. Variación volumétrica con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

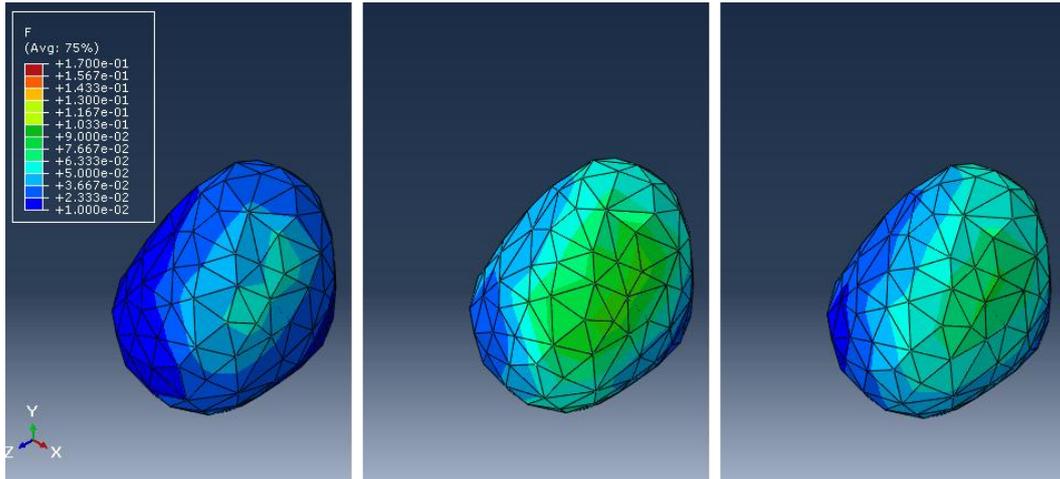


Figura B32. Variación de forma con respecto a la variación del módulo elástico del citoplasma. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

B.2.1. Efecto de la variación de la rigidez del núcleo

Resultados de la primera (Figuras B33-B36) y segunda versión (Figuras B37-B40).

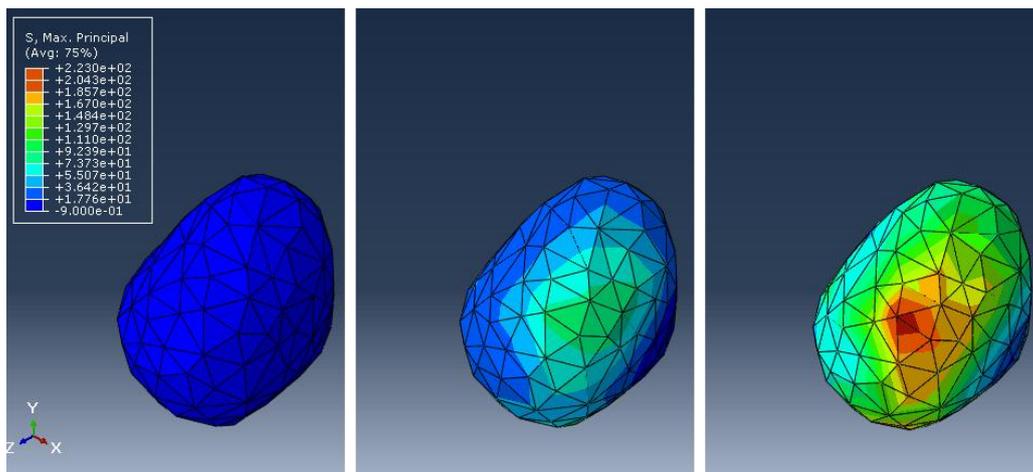


Figura B33. Tensión Máxima Principal con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

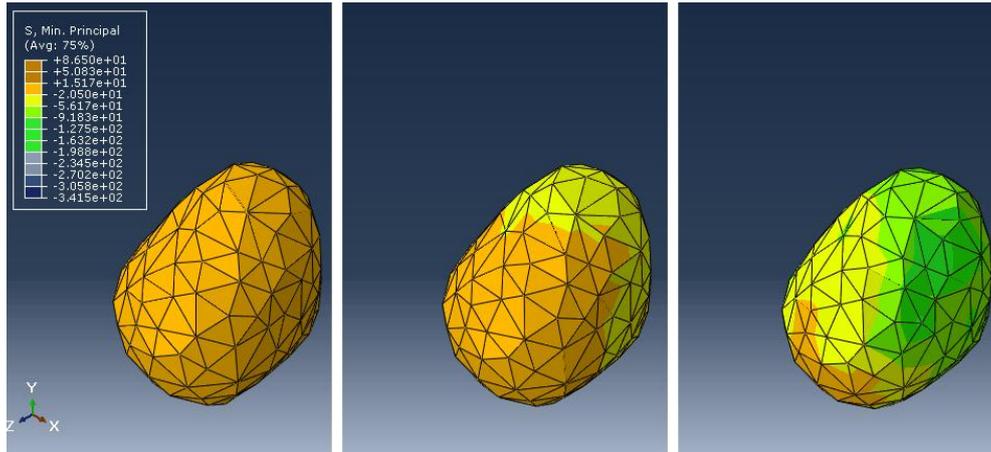


Figura B34. Tensión Mínima Principal con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

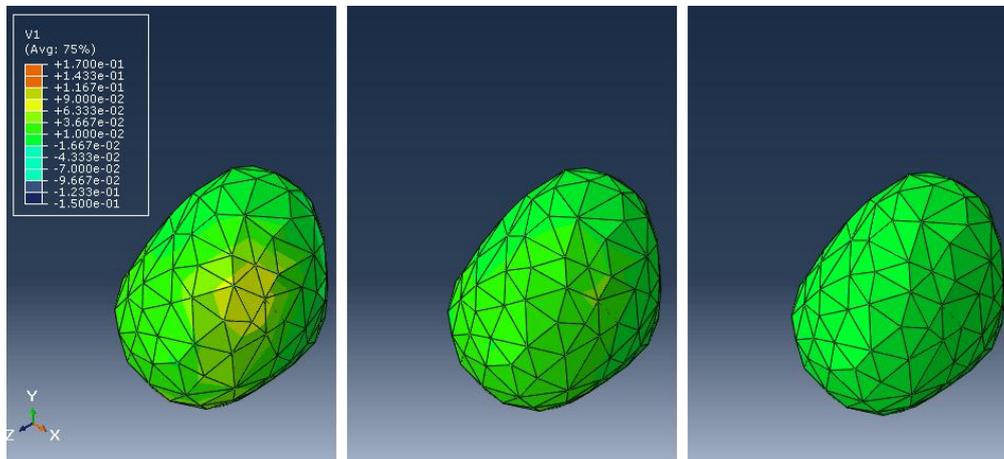


Figura B35. Variación volumétrica con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

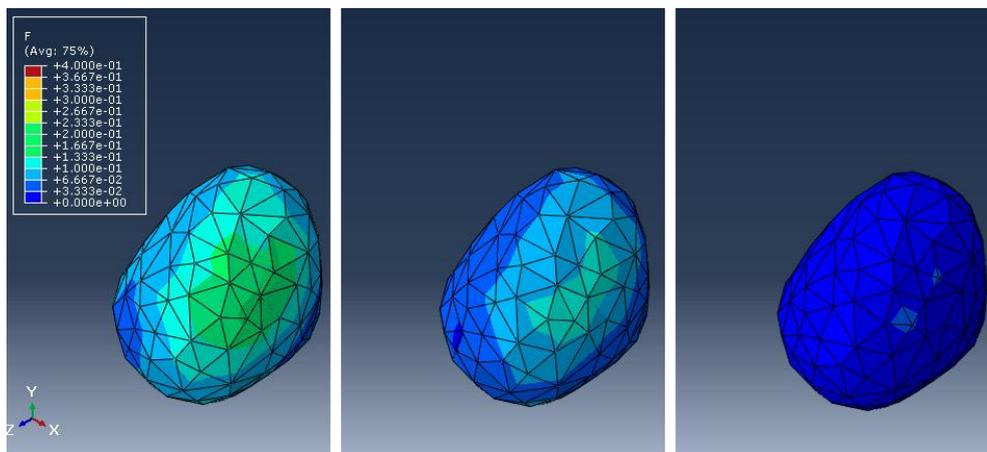


Figura B36. Variación de forma con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

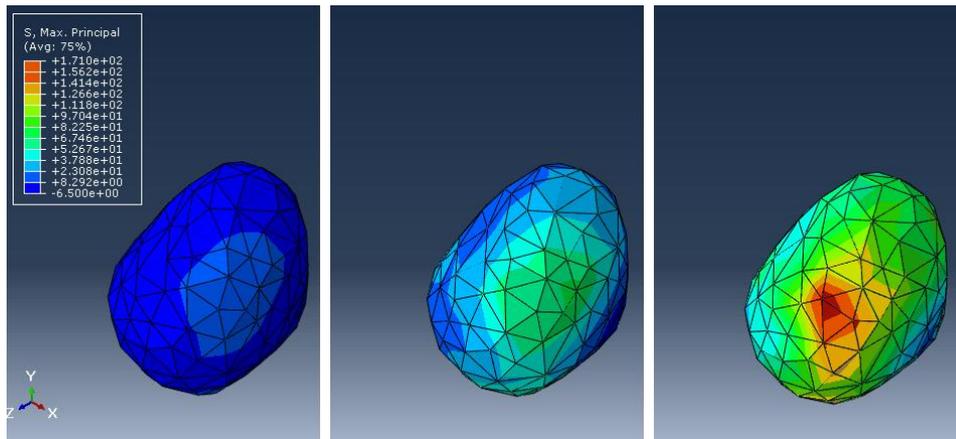


Figura B37. Tensión Máxima Principal con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

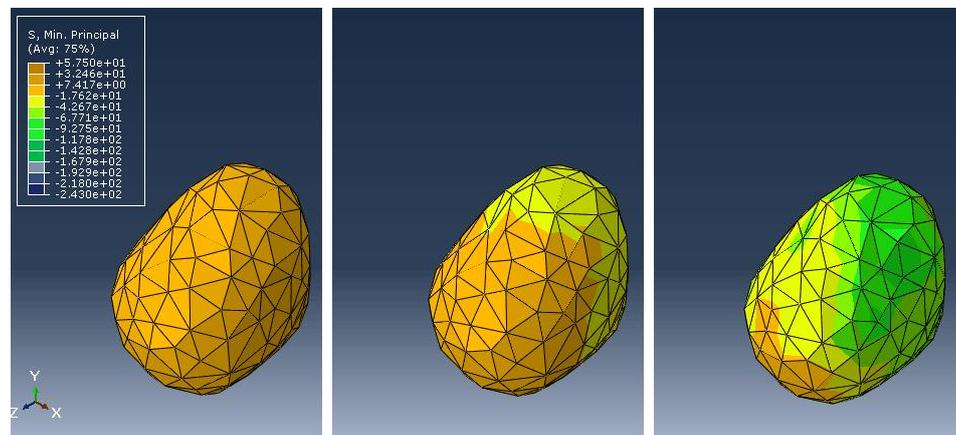


Figura B38. Tensión Mínima Principal con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

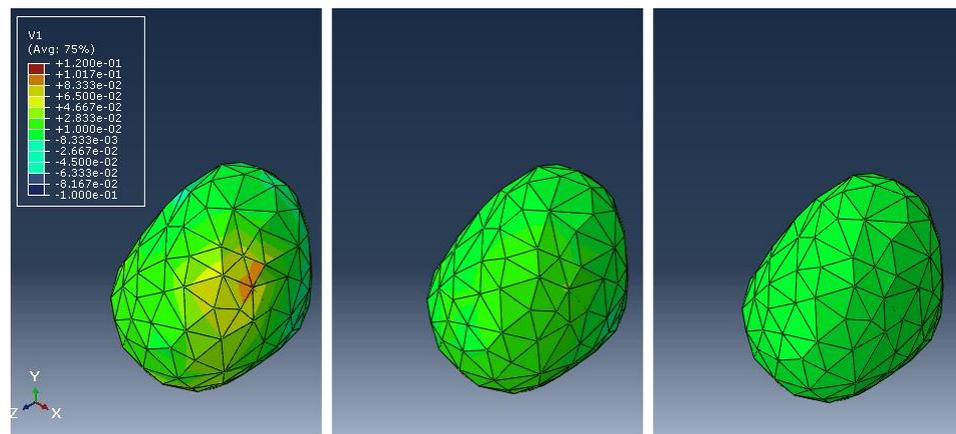


Figura B39. Variación volumétrica con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

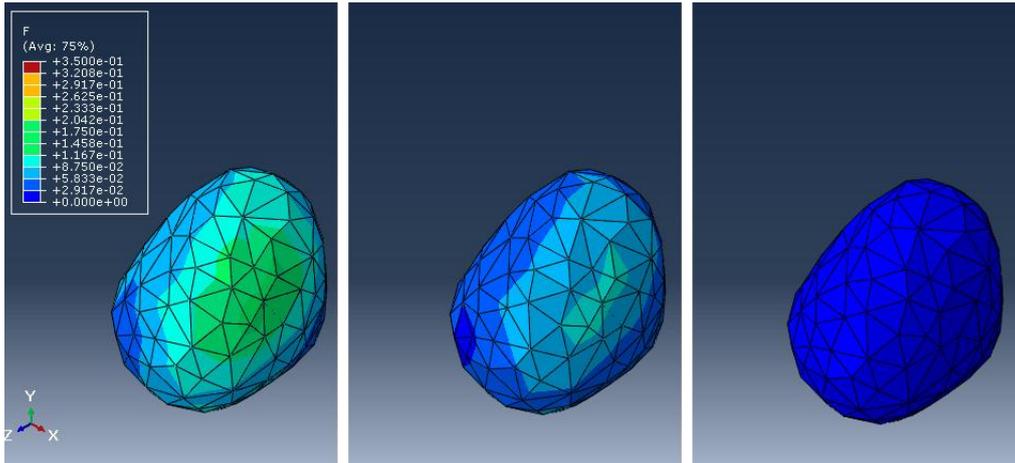


Figura B40. Variación de forma con respecto a la variación del módulo elástico del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

B.2.1. Efecto de la variación de la rigidez tanto del citoplasma como del núcleo

Resultados de la primera (Figuras B41-B44) y segunda versión (Figuras B45-B48).

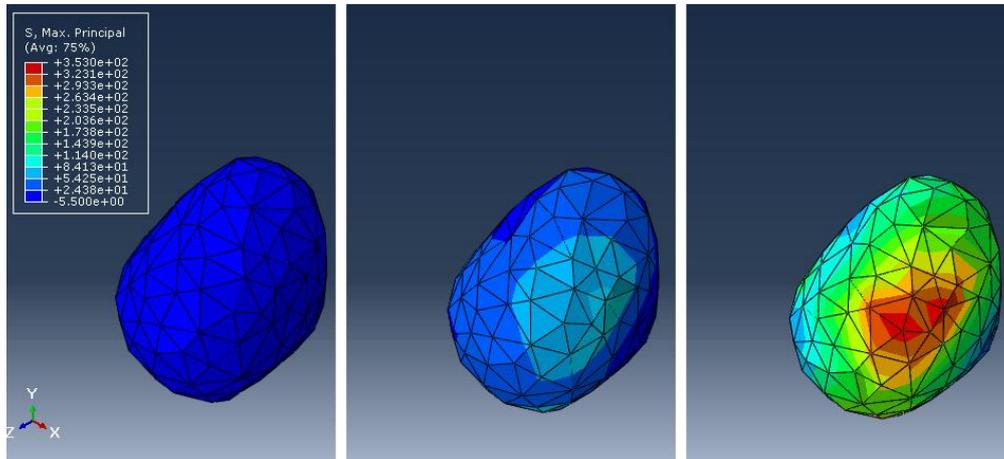


Figura B41. Tensión Máxima Principal con respecto a la variación del módulo elástico tanto del citoplasma como del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

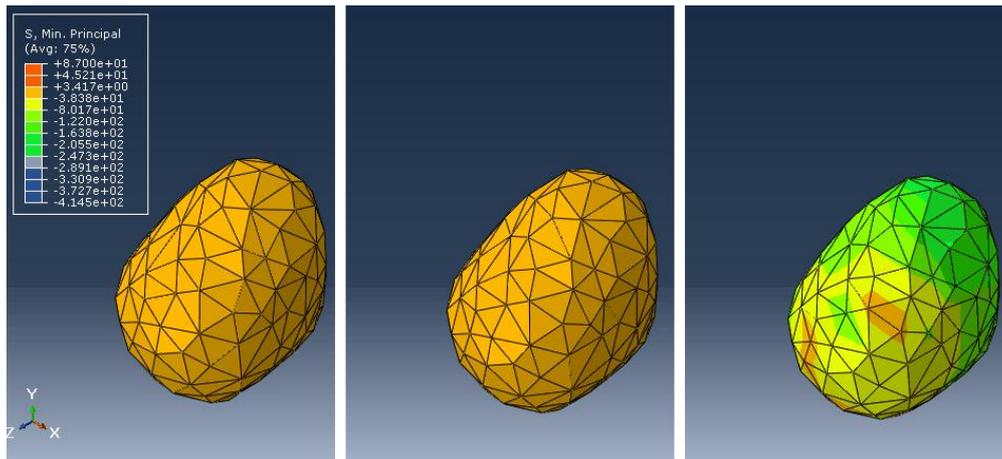


Figura B42. Tensión Mínima Principal con respecto a la variación del módulo elástico tanto del citoplasma como del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

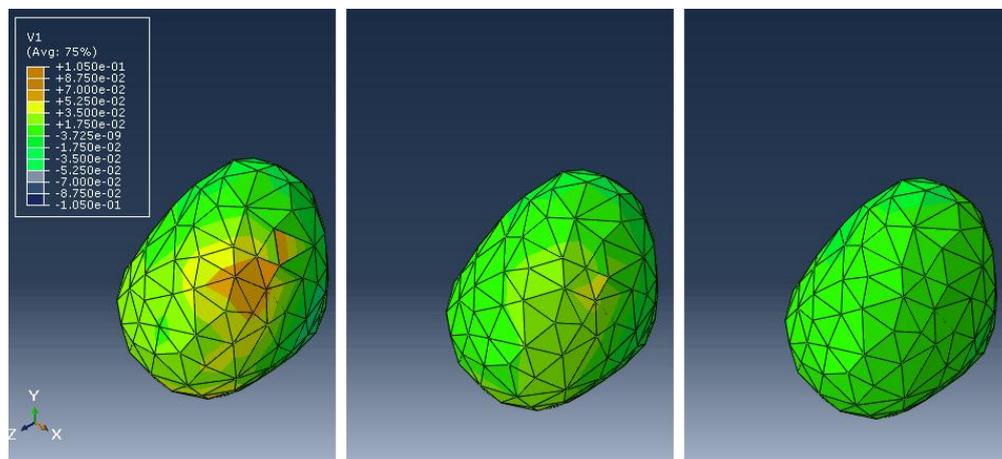


Figura B43. Variación volumétrica con respecto a la variación del módulo elástico tanto del citoplasma como del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

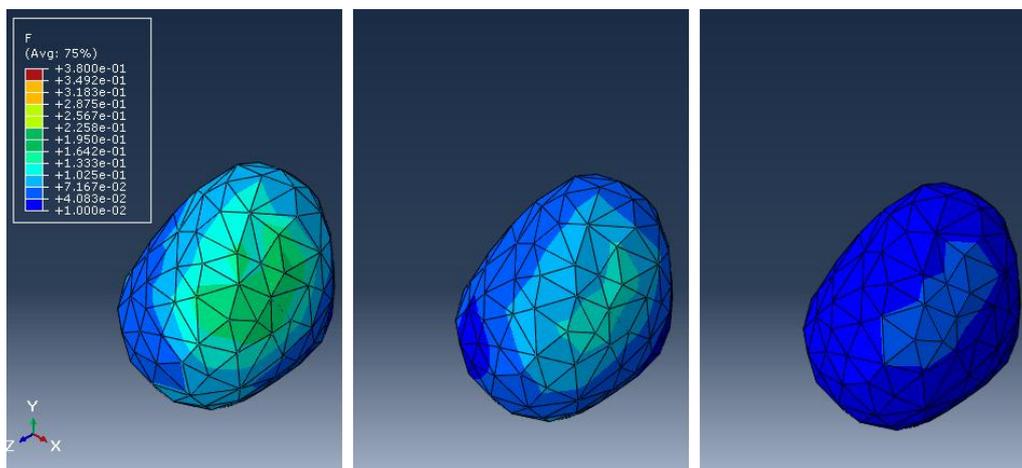


Figura B44. Variación de forma con respecto a la variación del módulo elástico tanto del citoplasma como del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 1.

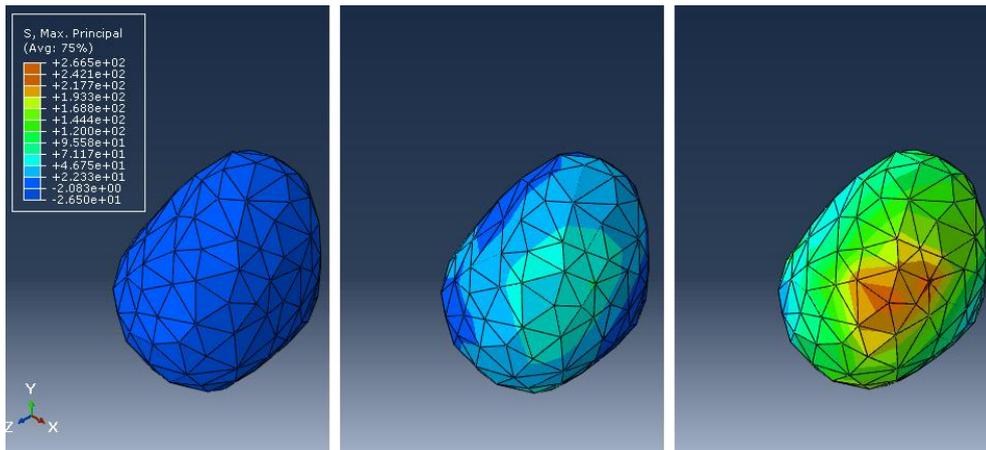


Figura B45. Tensión Máxima Principal con respecto a la variación del módulo elástico tanto del citoplasma como del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

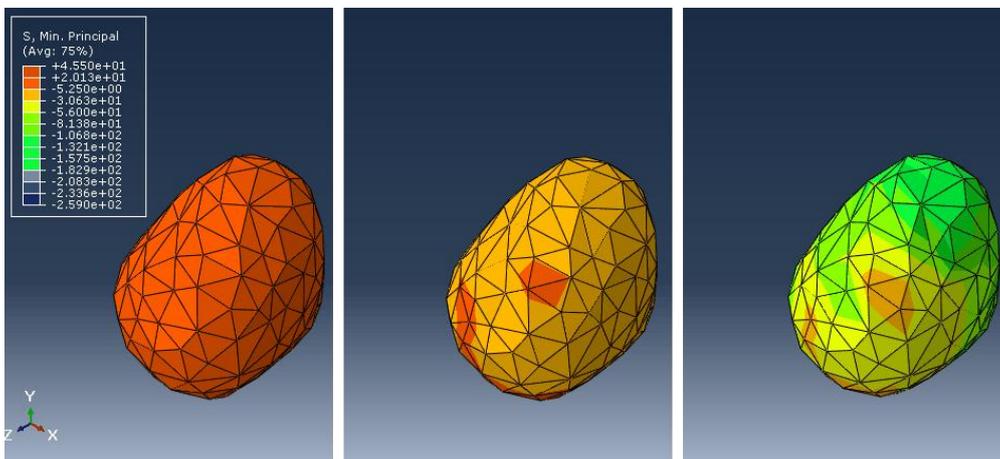


Figura B46. Tensión Mínima Principal con respecto a la variación del módulo elástico tanto del citoplasma como del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

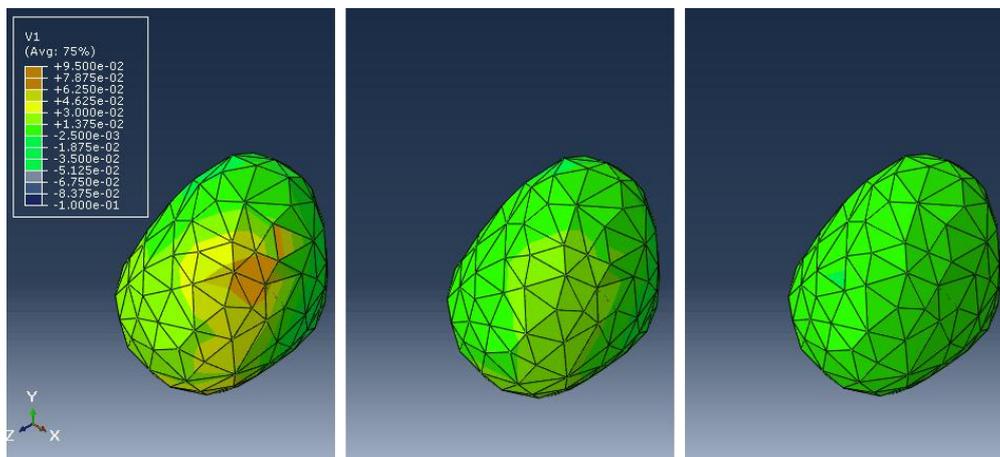


Figura B47. Variación volumétrica con respecto a la variación del módulo elástico tanto del citoplasma como del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

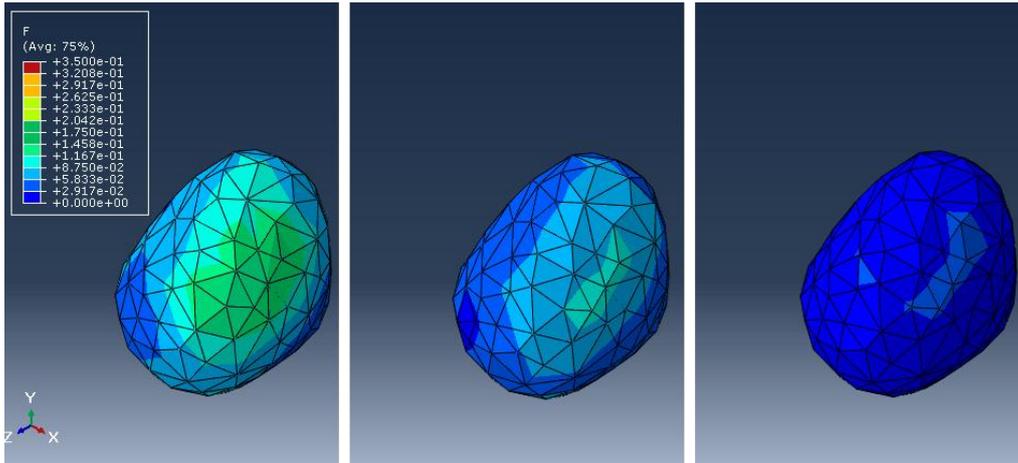


Figura B48. Variación de forma con respecto a la variación del módulo elástico tanto del citoplasma como del núcleo. Reducción de un orden de magnitud (izquierda), propiedades base (centro) y aumento en un orden de magnitud (derecha). Versión 2.

B.3. Caso III: Introducción de las microesferas de poliestireno en el modelo de célula relajada con tripsina

Para comprender mejor los resultados se muestran dos vistas donde se puede ver la localización relativa entre la microesfera y el núcleo (Figuras B49 Y B50). En la primera de ellas se observa la localización relativa entre ambos cuerpos con la misma perspectiva con la que se han mostrado los resultados. Además, se ha añadido una segunda vista lateral auxiliar para ayudar a la visualización.

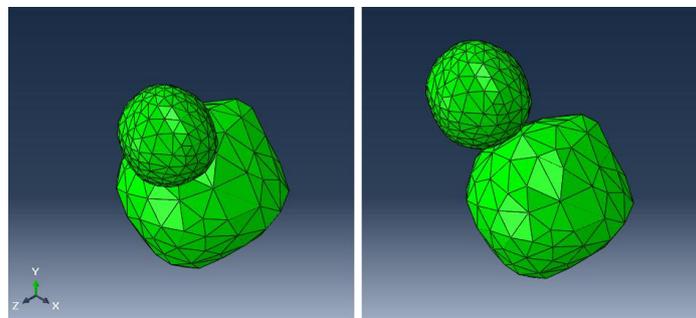


Figura B49. Vista de la posición relativa entre el núcleo y la microesfera en la perspectiva en la que se muestran los resultados. a) Microesfera próxima al núcleo. b) Microesfera lejana al núcleo

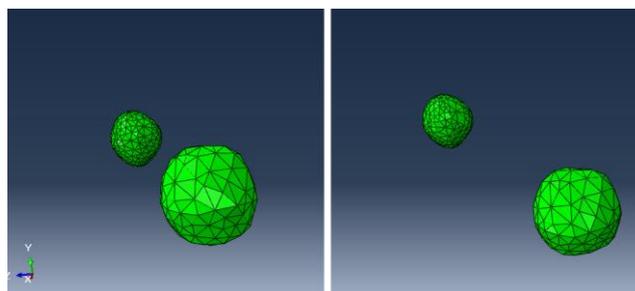


Figura B50. Vista lateral de la posición relativa entre el núcleo y la microesfera. a) Microesfera próxima al núcleo. b) Microesfera lejana al núcleo

En este último caso, se muestra en cada imagen el caso inicial a la izquierda de cada figura, el caso de la esfera más cerca del núcleo en el centro y el caso de la esfera más lejana al núcleo en la derecha (Figuras B51-B54). En general, pese a que los valores alcanzados varían notablemente de un caso a otro, las distribuciones no varían en exceso. Esto es debido a que, como se ha comentado en la memoria, se producen variaciones más puntuales.

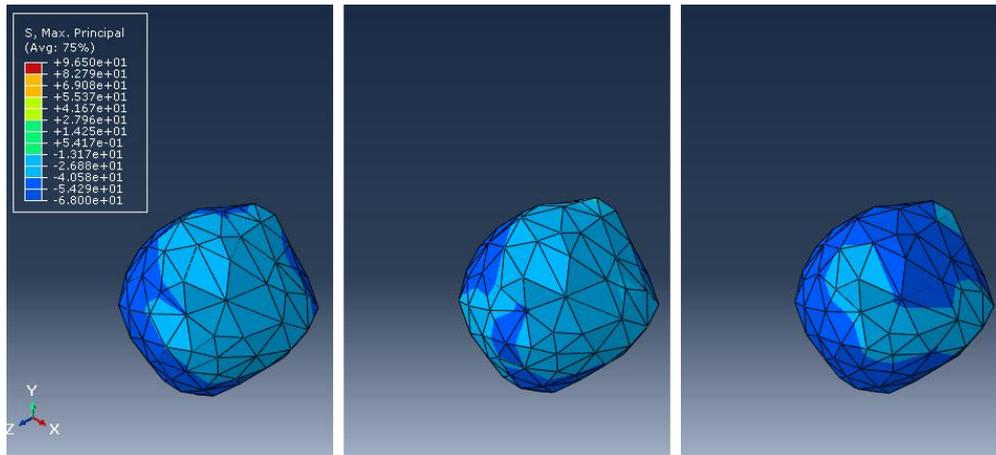


Figura B51. Distribución de la Tensión Máxima Principal. Modelo original sin microsferas (izquierda), modelo con la microsfera cercana (centro), modelo con la microsfera lejana (derecha).

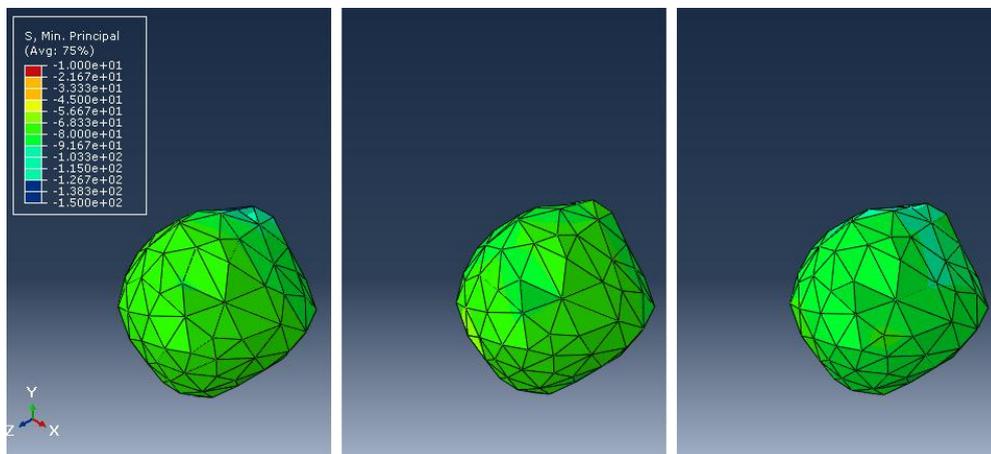


Figura B52. Distribución de la Tensión Mínima Principal. Modelo original sin microsferas (izquierda), modelo con la microsfera cercana (centro), modelo con la microsfera lejana (derecha).

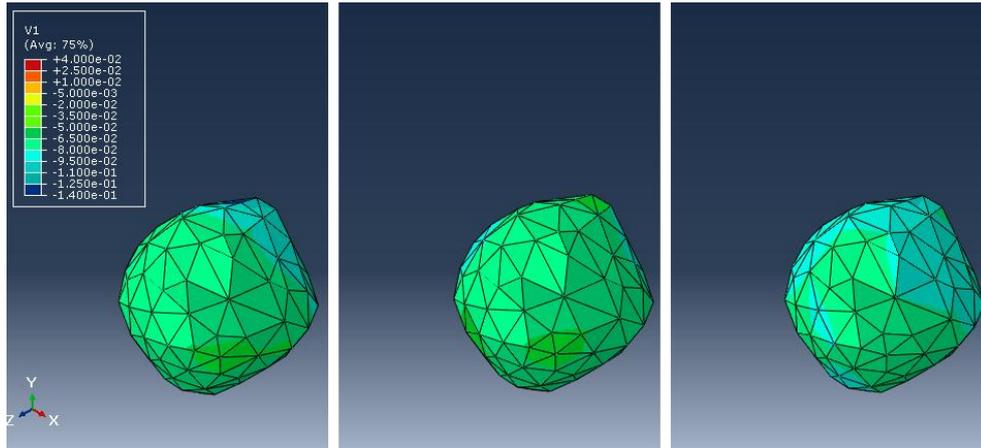


Figura B53. Distribución de la variación volumétrica. Modelo original sin microsferas (izquierda), modelo con la microsfera cercana (centro), modelo con la microsfera lejana (derecha).

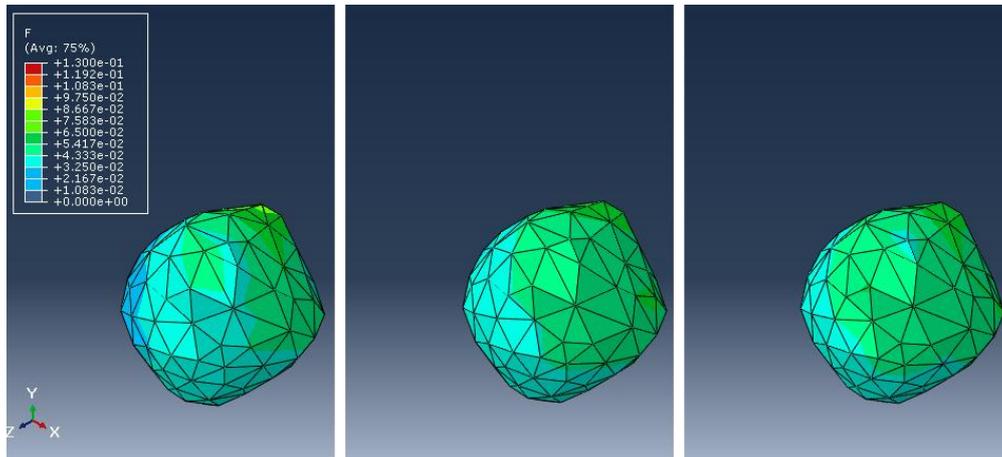


Figura B54. Distribución de la variación de forma. Modelo original sin microsferas (izquierda), modelo con la microsfera cercana (centro), modelo con la microsfera lejana (derecha).