



**UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA**

**PROGRAMA DE DOCTORADO EN INGENIERÍAS QUÍMICA, MECÁNICA Y DE  
FABRICACIÓN**

**TESIS DOCTORAL**

**MODELIZACIÓN Y SIMULACIÓN DEL PROCESO DE DEGRADACIÓN  
DE SCAFFOLDS FABRICADOS CON FABRICACION ADITIVA Y  
NANOMATERIALES.**

Autor: Jacob Abdelfatah Ndioubnane

Dirigida por: Gabriel Winter Althaus

Rubén Paz Hernández





**ULPGC**

**Universidad de  
Las Palmas de  
Gran Canaria**



Doctorado en Ingenierías  
Química, Mecánica y de Fabricación

## **TESIS DOCTORAL**

**MODELIZACIÓN Y SIMULACIÓN DEL PROCESO DE  
DEGRADACIÓN DE SCAFFOLDS FABRICADOS CON  
FABRICACION ADITIVA Y NANOMATERIALES.**

Doctorando

Jacob Abdelfatah  
Ndioubnane

Director

Gabriel Winter  
Althaus

Codirector

Rubén Paz  
Hernández

Las Palmas de Gran Canaria, Julio de 2025

---

## RESUMEN

Durante la realización de este trabajo se ha abordado el desarrollo de modelos que permitan predecir el comportamiento degradativo y erosivo de scaffolds elaborados a partir de polímeros bioabsorbibles. Inicialmente el trabajo se centra en el estudio de la erosión superficial debida al movimiento del medio circundante bajo condiciones in-vitro. Sin embargo, se observa que este modelo es aplicable bajo unas condiciones muy específicas (solo para velocidades de flujo altas), por lo que se decide valorar otros modelos de degradación/erosión más versátiles y realistas. Más adelante, se aborda el proceso erosivo bajo condiciones estáticas desarrollando un modelo probabilístico-experimental basado en autómatas celulares que toma el grado de exposición al medio como variable, así como los valores de referencia del ensayo experimental para medir la pérdida del material debida a la degradación enzimática. Estos ensayos se extrapolan a geometrías más complejas como scaffolds tomando la relación entre el volumen disuelto y el superficial. Sin embargo, al establecer la comparativa con los valores experimentales no se observa una correspondencia lo suficientemente clara. Es en este momento que se aborda la variación de propiedades intrínsecas del material, primero haciendo una revisión de las técnicas existentes, enfocándose primero en la degradación hidrolítica y luego en el cálculo de propiedades mecánicas. Por último, se propone un modelo integrado fenomenológico-probabilístico que permite cuantificar el proceso degradativo y erosivo del material.

## AGRADECIMIENTOS

El desarrollo de este trabajo no hubiese sido posible sin la fundamental supervisión que han ejercido mis directores de tesis, Gabriel Winter y Rubén Paz, por el constante apoyo que han mostrado durante la realización de este trabajo, agradecimiento extensivo al tutor académico Mario Monzón, por su disposición a solventar cualquier cuestión que permitiese progresar en el desarrollo de este trabajo.

Agradecer también a los compañeros del Grupo de Investigación de Fabricación Integrada y Avanzada, especialmente a Ricardo Donate y Gisela Vega, por su valiosa colaboración y asesoramiento, facilitarme los datos de los ensayos de degradación para la validación de los modelos desarrollados, así como de los modelos geométricos de las piezas que se han simulado.

También agradecer al Instituto de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería y a los diferentes miembros del mismo por su asesoramiento cuando fueron requeridos, y al técnico del Instituto Juan Ignacio González por facilitarme el acceso a los recursos computacionales del Centro de Procesamiento de Datos para la ejecución de las diferentes simulaciones.

También agradecer a diversos investigadores que se han prestado a colaborar, o que a través de su asesoramiento o la facilitación de recursos han podido dar un desarrollo adicional a este trabajo. Destacar a Andy Gleadall (Universidad de Loughborough), Gaizka Erkizia (Universidad del País Vasco), José Antonio Sanz Herrera (Universidad de Sevilla) y Marcin Heljak (Warsaw University of Technology).

Agradecer también a La Fundación Universitaria de Las Palmas por la financiación a través del Fondo de Ayudas a la Investigación que ha servido para financiar la primera fase de este trabajo y al proyecto BioAM del Grupo de Investigación de Fabricación Integrada y Avanzada.

Y, por último, expresar mi absoluta gratitud a mi familia, especialmente a mis padres, por erigirse en un apoyo constante y por ser un pilar fundamental para que lleve a término éste trabajo.



# ÍNDICE

1.	ESTRUCTURA DEL DOCUMENTO .....	1
2.	ANTECEDENTES.....	4
3.	OBJETIVOS.....	8
4.	ESTADO DEL ARTE.....	9
4.1.	DEGRADACIÓN HIDROLÍTICA.....	11
4.2.	ESTUDIO DE LA EROSIÓN DINÁMICA DEBIDA A LAS TENSIONES TANGENCIALES.....	12
4.3.	TIPOS DE MODELOS PARA SIMULAR EL COMPORTAMIENTO DEGRADATIVO DEL MATERIAL POLIMÉRICO.....	12
4.3.1.	MODELOS DETERMINISTAS.....	13
4.3.2.	MODELOS ESTOCÁSTICOS/PROBABILÍSTICOS .....	14
4.3.3.	MODELOS EMPÍRICOS.....	14
5.	MÉTODOS Y HERRAMIENTAS .....	15
5.1.	MÉTODOS NUMÉRICOS EMPLEADOS EN LOS CÁLCULOS COMPUTACIONALES.....	15
5.1.1.	MÉTODO DE LOS ELEMENTOS FINITOS.....	15
5.1.2.	SIMULACIONES DE MONTE CARLO.....	16
5.1.3.	AUTÓMATAS CELULARES.....	16
5.1.4.	MODELOS DE SIMULACIÓN POR DOMINIOS FICTICIOS.....	17
5.2.	HERRAMIENTAS UTILIZADAS.....	17
5.3.	PROCESAMIENTO A GRAN ESCALA.....	19
5.3.1.	CENTRO DE PROCESAMIENTO DE DATOS DEL IUSIANI.....	19
5.3.2.	CLUSTER LEIBNIZ.....	19
5.3.3.	RIEMANN.....	20
6.	ESQUEMA GENERAL DEL TRABAJO.....	21
7.	TRABAJOS PRELIMINARES .....	23
7.1.	APLICACIÓN DE UN MODELO DE EROSION A UN SCAFFOLD CILÍNDRICO BAJO CONDICIONES DINÁMICAS.....	23
7.1.1.	INTRODUCCIÓN .....	23
7.1.2.	METODOLOGÍA.....	24
7.1.3.	MODELADO GEOMÉTRICO.....	24
7.1.4.	DEFINICIÓN DEL MODELO .....	26

7.2.	RESULTADOS.....	33
7.3.	CONCLUSIONES OBTENIDAS DE LA APLICACIÓN DEL MÉTODO. ....	37
8.	ANÁLISIS ESTOCÁSTICO DEL COMPORTAMIENTO EROSIVO SUPERFICIAL DE LOS BIOPOLÍMEROS BAJO CONDICIONES ESTÁTICAS. ESTUDIO DE LA DEGRADACIÓN ENZIMÁTICA DE FILAMENTOS Y SCAFFOLDS.....	39
8.1.	METODOLOGÍA DE MONTE CARLO ASOCIADA AL CÁLCULO DE AUTÓMATAS CELULARES. ....	40
8.2.	SIMULACIÓN DE MONTE CARLO DE LA DEGRADACIÓN DE UN FILAMENTO DE COMPOSITE POLICAPROLACTONA-MICROCELULOSA (PCL-MCC). ....	44
8.2.1.	ANTECEDENTES .....	44
8.2.2.	METODOLOGÍA.....	45
8.2.3.	RESULTADOS .....	50
8.3.	SIMULACIÓN DE MONTE CARLO DE LA DEGRADACIÓN ENZIMÁTICA DE UN SCAFFOLD PRISMÁTICO.....	53
8.3.1.	RESULTADOS .....	55
8.4.	EXTRAPOLACIÓN DE LOS RESULTADOS OBTENIDOS EN LA SIMULACIÓN DEL FILAMENTO A UNA GEOMETRÍA GENERAL. ....	58
8.4.1.	METODOLOGÍA.....	58
8.4.2.	RESULTADOS .....	61
8.5.	CONCLUSIONES GENERALES.....	62
9.	TÉCNICAS DE MODELIZACIÓN DE LA DEGRADACIÓN HIDROLÍTICA DE UN MATERIAL POLIMÉRICO.....	64
9.1.	REACCIÓN DIFUSIÓN .....	64
9.1.1.	IMPLEMENTACIÓN ESTANDAR DE LAS ECUACIONES DE REACCIÓN DIFUSIÓN EN FREEFEM++ .....	67
9.1.2.	IMPLEMENTACION MATRICIAL DE LAS ECUACIONES DE REACCIÓN DIFUSION .....	74
10.	MODELIZACIÓN DE LA EVOLUCIÓN DE LAS PROPIEDADES INTRÍNSECAS.....	79
10.1.	INTRODUCCIÓN.....	79
10.2.	METODOLOGÍA ASOCIADA AL CÁLCULO DE LAS PROPIEDADES MECÁNICAS DE LA PIEZA USANDO EL CRITERIO DE MÓDULO DE ELASTICIDAD GLOBAL. ....	81

10.2.1.	APLICACIÓN DEL CRITERIO DE MODULO DE ELASTICIDAD GLOBAL SOBRE UNA PIEZA .....	83
10.3.	SIMULACIÓN DE LA ROTURA DE CADENAS POLIMÉRICAS Y CÁLCULO PROMEDIO DE PROPIEDADES MECÁNICAS .....	86
10.3.1.	DISTRIBUCIÓN INICIAL DE PESO MOLECULAR.....	86
10.3.2.	SIMULACIÓN DEL PROCESO DE DEGRADACIÓN DE LAS PROPIEDADES MECÁNICAS: .....	88
10.3.3.	ESCISIÓN MOLECULAR.....	89
10.3.4.	ESCISIÓN EN LOS EXTREMOS. ....	91
10.3.5.	ACTUALIZACIÓN DE PROPIEDADES MECÁNICAS.....	92
11.	ACOMPLAMIENTO DE LOS MODELOS DE DEGRADACIÓN Y EROSIÓN.....	95
11.1.	INTRODUCCIÓN.....	95
11.2.	PRIMERA PROPUESTA DE INTEGRACIÓN DEL MODELO DEGRADATIVO CON EL MODELO HIDROLÍTICO. ....	96
11.3.	SEGUNDA PROPUESTA DE MODELO INTEGRADO: APLICANDO REFINAMIENTO LOCAL A LA MALLA.....	102
11.4.	INTEGRACIÓN DE LOS MODELOS DE DEGRADACIÓN Y EROSION MEDIANTE LA APLICACIÓN DE DOMINIOS FICTICIOS.....	109
11.4.1.	INTRODUCCION .....	109
11.4.2.	ESQUEMA GENERAL DEL MODELO.....	110
11.4.3.	PREPROCESADO DEL DOMINIO GEOMÉTRICO. ....	121
11.4.4.	CONDICIONES DE CONTORNO .....	127
11.4.5.	MODELIZACIÓN DE LA POROSIDAD.....	130
11.4.6.	RESULTADOS .....	135
12.	CONCLUSIONES .....	138
13.	LÍNEAS FUTURAS .....	143
14.	BIBLIOGRAFÍA.....	145
15.	ANEXOS.....	153
15.1.	METODOLOGÍA PARA EL CÁLCULO DE PROPIEDADES MECÁNICAS POR MEDIO DEL SOFTWARE ABAQUS. ....	153
15.1.1.	DEFINICIÓN DE CARGAS Y CONDICIONES DE CONTORNO 170	
15.1.2.	EJECUCIÓN DE LA SIMULACIÓN .....	173
15.2.	SCRIPT CORRESPONDIENTE AL CÁLCULO DE LA EROSIÓN DEBIDO A LAS TENSIONES TANGENCIALES EN UNA GEOMETRÍA SENCILLA.....	179



<b>15.3. SCRIPT CORRESPONDIENTE A LA GENERACIÓN DE LA MALLA ESTRUCTURADA DEL FILAMENTO POLICAPROLACTONA(PCL).....</b>	<b>184</b>
<b>15.3. SCRIPT CORRESPONDIENTE AL MODELO CORRESPONDIENTE AL MODELO DE MONTE CARLO PARA LA EROSIÓN ESTÁTICA DE UN FILAMENTO. ....</b>	<b>185</b>
<b>15.4. SCRIPT CORRESPONDIENTE A LA PRIMERA PROPUESTA DEL MODELO INTEGRADO PARA LA DEGRADACIÓN Y EROSIÓN DEL POLÍMERO .....</b>	<b>193</b>
<b>15.5. SCRIPT CORRESPONDIENTE AL REFINAMIENTO DE MALLA EN EL MODELO DE REACCIÓN DIFUSIÓN. ....</b>	<b>206</b>
<b>15.6. SCRIPT CORRESPONDIENTE A LA GENERACIÓN DE LA MALLA ESTRUCTURADA PARA LA CELDA UNITARIA.....</b>	<b>212</b>
<b>15.7. SCRIPT CORRESPONDIENTE AL MODELO INTEGRADO APLICADO A UNA CELDA UNITARIA CON DISCRETIZADO NO ESTRUCTURADO.....</b>	<b>218</b>

## ÍNDICE DE FIGURAS

Figura 2-1. Representación de las distintas topologías que representan los scaffolds. Extraída de [10].	5
Figura 4-1. Representación del mecanismo de erosión en polímeros bioabsorbibles. Figura extraída de [32].	10
Figura 4-2. Esquema conceptual de las diferentes metodologías empleadas para la modelización de la degradación de polímeros.	13
Figura 6-1. Esquema general del trabajo	22
Figura 7-1. Esquema general de cálculo de la erosión y de propiedades mecánicas del scaffold bajo condiciones dinámicas.	27
Figura 7-2. Modelo de scaffold.	28
Figura 7-3. Modelo geométrico del biorreactor para la simulación (basado en [26])	28
Figura 7-4. Mallado del scaffold.	29
Figura 7-5. Malla representativa del espacio fluido.	29
Figura 7-6. Malla representativa del dominio líquido. Sección transversal.	30
Figura 7-7. Líneas de corriente de velocidad	34
Figura 7-8. Mapa de tensiones tangenciales	34
Figura 7-9. Geometría del scaffold en el instante inicial	35
Figura 7-10. Geometría del scaffold en el instante inicial	35
Figura 7-11. Evolución de la degradación en el instante $t=0.02$ .	36
Figura 7-12. Evolución de la degradación en $t=0.037$ .	36
Figura 7-13. Evolución de la degradación en $t=0.05$ .	37
Figura 8-1. Representación en 3D de la vecindad de Von Neumann usando vóxeles.	41
Figura 8-2. Ejemplo de discretizado de una pieza cónica usando elementos tetraédricos(izquierda) y elementos hexaédricos(derecha) realizado en GMSH, basado en [87]	42
Figura 8-3. Representación de la vecindad de Von Neumann para elementos tetraédricos.	42
Figura 8-4. Representación de la vecindad de Von Neumann para tetraedros regulares en 3D.	43
Figura 8-5. Representación ortogonal de la vecindad de Von Neumann en 3d.	44
Figura 8-6. Esquema del modelo de cálculo estocástico de la degradación enzimática.	46
Figura 8-7. Selección de los elementos superficiales de la geometría de la malla.	47
Figura 8-8. Diseño del filamento (de diámetro 1,9 mm y de longitud 50,3 mm) y vistas parciales de la geometría mallada (con 4258800 elementos y 730431 nodos). a) Geometría inicial del filamento; b) Sección transversal de la geometría discretizada; c) Vista longitudinal de la geometría mallada.	49
Figura 8-9. Detalle de las geometrías erosionadas. a) a las 24 horas. Y b) a las 48 horas.	53
Figura 8-10. Modelo 3d de un scaffold de tipo prismático.	54
Figura 8-11. Evolución de la degradación (Etapa 1)	56
Figura 8-12. Evolución de la erosión (Etapa 2)	56
Figura 8-13. Evolución de la erosión (etapa 3)	57
Figura 8-14. Evolución de la erosión (etapa 4)	57

Figura 8-15. Diagrama de flujo del modelo extrapolado de Monte Carlo.....	59
Figura 8-16. Representación de la malla del filamento.....	60
Figura 8-17. Representación de malla del scaffold.....	61
Figura 9-1. Implementación de las ecuaciones fenomenológicas de Wang en términos de la variación de la concentración de monómeros.....	68
Figura 9-2. Implementación de las ecuaciones fenomenológicas de Wang en términos de la variación de la concentración de ésteres.....	68
Figura 9-3. Evolución temporal de la concentración de monómeros en el proceso degradativo.....	74
Figura 9-4. Implementación matricial de las ecuaciones hidrolíticas de Wang en FreeFem++.....	76
Figura 9-5. Comparativa de la evolución de monómeros en el modelo standard y matricial con valores del parámetro $n = 0,2$ .....	77
Figura 9-6. Comparativa de la evolución de monómeros en el modelo standard y matricial con valores del parámetro $n = 0,5$ .....	78
Figura 10-1. Esquema de proceso propuesto para cálculo de propiedades mecánicas globales.....	82
Figura 10-2. Representación 3d del modelo de scaffold (a)) y vista frontal con las cargas y desplazamientos aplicados(b)).....	83
Figura 10-3. Distribución de peso molecular.....	87
Figura 10-4. Distribución de longitud de cadena.....	88
Figura 10-5. Tipos de escisión de una cadena polimérica: A) Escisión aleatoria; B) Escisión en los extremos.....	89
Figura 10-6. Diagrama del proceso de obtención del módulo de elasticidad con la rotura molecular.....	90
Figura 10-7. Representación gráfica de la relación entre los valores estandarizados Módulo de Elasticidad vs Peso Molecular Normalizado en diferentes instantes del proceso degradativo.....	93
Figura 10-8. Distribución del módulo de Young en la celda unitaria en las diferentes etapas de degradación hidrolítica.....	94
Figura 11-1. Diagrama de flujo correspondiente a la implementación numérica del primer modelo propuesto modelo integrado de degradación hidrolítica y erosión.....	98
Figura 11-2. Distribución del peso molecular normalizado de la celda unitaria en diferentes etapas de degradación.....	101
Figura 11-3. Diagrama de flujo correspondiente a la implementación numérica del modelo integrado de degradación hidrolítica y erosión con refinamiento de malla.....	103
Figura 11-4. Representación del dominio discretizado utilizado como caso base para evaluar el comportamiento numérico de la segunda propuesta.....	104
Figura 11-5. Evolución del proceso erosivo al aplicar un refinamiento local en el contorno.....	106
Figura 11-6. Imagen de la celda unitaria discretizada y la distribución del peso molecular en distintas etapas del proceso de degradación.....	109



Figura 11-7. Diagrama de flujo del modelo integrado de degradación hidrolítica y erosión superficial. ....	111
Figura 11-8. Representación dominio con dos regiones simplificado.....	112
Figura 11-9. Representación de la distancia signada respecto a la interfase .....	114
Figura 11-10. Representación en vista de sección de la función característica usada para aplicar condiciones de contorno. ....	118
Figura 11-11. Vista de la sección frontal del dominio ficticio discretizado.....	122
Figura 11-12. Representación 3D de la celda unitaria .....	123
Figura 11-13. Representación frontal(a) y en perspectiva(b) de un filamento semicilíndrico. ....	123
Figura 11-14. Representación de la sección transversal de la celda unitaria.....	124
Figura 11-15. Representación de las regiones de la celda unitaria. ....	124
Figura 11-16. Representación geométrica de la celda unitaria del scaffold para un a) 80%, 87%, y 92% de porosidad .....	126
Figura 11-17. Representación de la celda unitaria contenida en dominio ficticio. ....	126
Figura 11-18. Vista en sección frontal de la celda unitaria embebida en el volumen ficticio .....	127
Figura 11-19. Representación de los dominios de estudio del problema. ....	128
Figura 11-20. Representación de las condiciones de frontera en el scaffold (a), la celda unitaria (b) y en la frontera ficticia (c). ....	130
Figura 11-21. Representación de la variación de la porosidad en función del tiempo de degradación expresado en semanas. Tomado con permiso de Agrawal [28]. ....	131
Figura 11-22. Expresión empírica para el cálculo de porosidad de acuerdo con Agrawal. ....	132
Figura 11-23. Representación de la interfase calculada usando el método de los dominios ficticios (80% porosidad).....	135
Figura 11-24. Representación de la región líquida de la celda unitaria del dominio ficticio. ....	136
Figura 11-25. Comparativa evolución porosidad simulada vs experimental. ....	136
Figura 12-1. Esquema general del modelo integrado finalmente adoptado.....	141
Figura 15-1. Apertura de ficheros en gmsh.....	153
Figura 15-2. Especificación de directorio del fichero .geo.....	154
Figura 15-3. Especificación del motor de geometría en gmsh .....	154
Figura 15-4. Importación de la malla en el bloc de notas. ....	154
Figura 15-5. Visualización de la geometría importada dentro del programa.....	155
Figura 15-6. Activación de la opción "Show bounding boxes" para la comprobación de las dimensiones de la pieza.....	156
Figura 15-7. Definición del tamaño de malla-1. ....	156
Figura 15-8. Definición del tamaño de malla-2 .....	157
Figura 15-9. Visualización de la malla generada. ....	158
Figura 15-10. Malla importada en ABAQUS.....	159
Figura 15-11. Cambiar de módulo mesh en el menú de árbol de ABAQUS.....	160
Figura 15-12. Selección del módulo de edición de mallas.....	160
Figura 15-13. Selección del método de eliminación de elementos de la malla.....	161
Figura 15-14. Selección de elementos 2D en la barra de selección de entidades. ....	161
Figura 15-15. Visualización de los elementos de superficie seleccionados en color rojo... ..	162
Figura 15-16. Tecla "Done" para la eliminación de los elementos seleccionados.....	162

Figura 15-17. Selección del módulo de propiedades en ABAQUS .....	162
Figura 15-18. Módulo de propiedades en ABAQUS.....	163
Figura-15-19. Ventana principal del gestor de propiedades .....	163
Figura 15-20. Selección del tipo de material para el estudio.....	164
Figura -15-21. Definición de las propiedades del material.....	164
Figura 15-22. Panel principal del gestor de secciones .....	165
Figura-15-23. Asignación de material a las secciones .....	165
Figura 15-24. Representación de la pieza en color turquesa tras asignación de propiedades .....	166
Figura 15-25. Panel principal en el modulo step .....	166
Figura 15-26. Creación de análisis .....	167
Figura 15-27. Ventana de edición del step. ....	168
Figura 15-28. Ventana de edición de la salida de datos en "Field Output" .....	168
Figura 15-29. Ventana de edición de la salida de datos en "History Output".....	169
Figura 15-30. Ventana de inserción de "Time points". ....	169
Figura 15-31. Selección del modo "Last Increment" en Field Output Manager. ....	170
Figura 15-32. Ventana de creación de condiciones de contorno .....	171
Figura 15-33. Definición de restricciones en la base de la pieza .....	172
Figura 15-34. Definición de restricciones en un punto central de la base.....	172
Figura 15-35. Definición de desplazamiento en la cara superior del scaffold.....	173
Figura 15-36. Editor de trabajos .....	174
Figura 15-37. Gestor de trabajos .....	174
Figura 15-38. Mapa de fuerzas de reacción sobre la pieza.....	175
Figura 15-39. Ventana de creación de la tabla de resultados. Selección de la casilla "ODB field output".....	175
Figura 15-40. selección de las fuerzas de reacción como variable a almacenar en la tabla	176
Figura 15-41. Selección de los nodos donde queremos registrar la fuerza de reacción.....	176
Figura 15-42. Utilización del operador sum() para sumar los valores de la fuerza de reacción en la base. ....	177
Figura 15-43. Representación de la fuerza resultante respecto al tiempo.....	177
Figura 15-44. Localización del punto de fuerza resultante en la gráfica. ....	178
Figura 15-45. Medición de las dimensiones de la pieza.....	179

## ÍNDICE DE TABLAS

Tabla 7-1. Ecuaciones implícitas para la generación de superficies TPMS. ....	26
Tabla 8-1. Resultados de pérdida de masa y módulo de elasticidad debido a la degradación enzimática. ....	50
Tabla 8-2. Resultados simulados de degradación de un filamento de PCL. ....	51
Tabla 8-3. Resultados simulados de degradación de un filamento de PCL-MCC 98:2. ....	53
Tabla 8-4. Tabla de resultados experimentales de pérdida de masa porcentual y del módulo de Young promedio correspondientes al scaffold rectangular. ....	55
Tabla 8-5. Tabla de resultados obtenidos del modelo de Monte Carlo de la pieza degradada. ....	55
Tabla 8-6. Tabla de resultados obtenidos de $v_r/v_s$ en cada etapa de degradación del filamento. ....	61
Tabla 8-7. Valores representativos de pérdida volumétrica en las diferentes etapas de degradación para el modelo simulado. ....	62
Tabla 10-1. Datos relativos a la pieza utilizada para las simulaciones, así como el desplazamiento aplicado al mismo en el ensayo mecánico. ....	84
Tabla 10-2. Valores de referencia de módulo de elasticidad intrínseco del filamento en diferentes etapas de degradación. ....	84
Tabla 10-3. Tabla de resultados de módulo de elasticidad de la pieza y comparativa con valores de referencia. ....	85
Tabla 10-4. Variación porcentual relativa del módulo de elasticidad global respecto al estado inicial de la pieza. ....	85
Tabla 11-1 Distancia entre centro de filamentos en función de la porosidad. ....	126





# 1. ESTRUCTURA DEL DOCUMENTO

## **ANTECEDENTES.**

En este capítulo se contextualiza de forma general las contribuciones realizadas hasta la fecha en la temática de la ingeniería tisular y en la modelización del proceso degradativo y erosivo de polímeros, resaltando su aplicación en celdas unitarias de scaffolds bioabsorbibles. De aquí se concluye la necesidad de desarrollar una metodología numérica que permita estimar el comportamiento degradativo y erosivo de scaffolds bioabsorbibles con el paso del tiempo.

## **OBJETIVOS.**

En este apartado se trata de definir los diferentes objetivos necesarios para llevar a término este trabajo. Para ello se pretende utilizar una metodología de diseño integrada que, utilizando software de simulación de código abierto basado en C++, integrado con otros programas para el preprocesado y postproceso de la malla, permite determinar los parámetros de diseño e indicadores del estado de degradación de la pieza para su posterior ajuste a modelos experimentales de referencia.

## **ESTADO DEL ARTE.**

Se hace una revisión punto por punto de las diferentes técnicas y metodologías propuestas por los diferentes autores que han servido para predecir el proceso de degradación y erosión polimérica de scaffolds con el paso del tiempo, la degradación de materiales biopoliméricos bajo condiciones estáticas y dinámicas, así como la evolución de las propiedades mecánicas a lo largo del tiempo.

## **ESQUEMA GENERAL DEL TRABAJO.**

En este capítulo se describe esquemáticamente y de manera cronológica cada una de las etapas llevadas a cabo en el desarrollo de este trabajo.

## **MÉTODOS Y HERRAMIENTAS.**

En este capítulo, se describen los principales métodos numéricos empleados en el desarrollo de este trabajo. Asimismo, se listan las principales herramientas informáticas utilizadas para las implementaciones

## **TRABAJOS PRELIMINARES.**

En este capítulo se describe la propuesta inicial de este trabajo utilizando el criterio de tensiones tangenciales para el modelado de la erosión superficial. Más adelante se discuten las limitaciones encontradas a nivel computacional y conceptual en la ejecución de este modelo.

## **ANÁLISIS ESTOCÁSTICO DEL COMPORTAMIENTO EROSIVO SUPERFICIAL DE LOS BIOPOLÍMEROS BAJO CONDICIONES ESTÁTICAS. ESTUDIO DE LA DEGRADACIÓN ENZIMÁTICA DE FILAMENTOS Y SCAFFOLDS.**

En este capítulo se describe una propuesta de un modelo estocástico-experimental basado en autómatas celulares que permita describir el comportamiento erosivo superficial del material bajo condiciones estáticas y tomando el grado de exposición como variable de referencia en el proceso erosivo.

## **TÉCNICAS DE MODELIZACIÓN DE LA DEGRADACIÓN HIDROLÍTICA DE UN MATERIAL POLIMÉRICO.**

Se hace una revisión de los principales modelos presentados hasta la fecha para la degradación hidrolítica de materiales poliméricos realizando una implementación de uno de los modelos presentados. Se realiza una implementación numérica en elementos finitos de la ecuación fenomenológica de Wang [1] y se compara con los resultados obtenidos por este. Se añade una segunda propuesta en la que se linealiza la ecuación con métodos iterativos de punto fijo. En una tercera propuesta se realiza la notación matricial consiguiendo una mejor eficiencia computacional del modelo propuesto en relación con las otras técnicas.

## **MODELIZACIÓN DE LA EVOLUCIÓN DE LAS PROPIEDADES INTRÍNSECAS.**



En este capítulo se procede a estudiar la evolución de las propiedades intrínsecas del material bio-polimérico. Se establece una correlación entre el peso molecular y el módulo de elasticidad promedio del material que permitirá determinar la evolución temporal de las pérdidas mecánicas con el tiempo.

## **ACOPLAMIENTO DE LOS MODELOS DE DEGRADACIÓN Y EROSIÓN.**

En este capítulo se propone una metodología integrada de degradación y erosión superficial, realizando sendas implementaciones numéricas con el fin de obtener una metodología lo suficientemente estable que permita determinar un proceso completo de erosión y degradación bajo diferentes condiciones dinámicas y estáticas.

## **CONCLUSIONES.**

En este capítulo se extraerán las principales conclusiones derivadas de los resultados obtenidos, poniendo en valor aquellos aspectos que suponen una mejora de lo hecho hasta ahora en la literatura, así como incidir en las limitaciones o dificultades encontradas durante la realización de este trabajo.

## **LÍNEAS FUTURAS.**

En este último capítulo del trabajo se trazan las futuras líneas de trabajo con el objetivo de expandir y mejorar el trabajo presentado.

## 2. ANTECEDENTES

La Ingeniería Tisular es un campo multidisciplinar que aplica los principios de la ingeniería y las ciencias biológicas en aras del desarrollo de sustitutos biológicos que restauren, mantengan y/o mejoren la función del tejido a reparar o sustituir [2]. Esta combina los principios de los materiales con la trasplante celular para desarrollar tejidos sustitutos y/o promover la regeneración endógena [3].

El objetivo principal de la Ingeniería Tisular es el de la fabricación de construcciones funcionales que permitan la restauración o mejora de los tejidos dañados [4]. Entre los campos de aplicación se encuentran el de los tejidos óseos, y tejidos blandos, entre otros. La disciplina se basa en la asociación de los biomateriales, las células y de los factores biológicos o de entorno [5]. Los materiales empleados en ingeniería tisular pueden obtenerse de diversas fuentes, principalmente naturales o sintéticas, y todos deben cumplir unos requisitos de biocompatibilidad para evitar la inducción de la respuesta inmune además de ser esterilizable para facilitar una integración segura con el tejido del paciente [4];

En el campo de la ingeniería tisular ósea (Bone Tissue Engineering, en inglés) se recurre a materiales como el titanio y derivados, pero, también se emplean materiales poliméricos. En el caso de los tejidos blandos se hace uso de polímeros bioabsorbibles. Los polímeros son por amplia mayoría los materiales más usados en ingeniería tisular, entre los cuales se encuentran el ácido poliláctico (PLA), el ácido poliglicólico (PGA) y copolímeros de los anteriores [6].

Gracias a la fabricación aditiva, se pueden construir una serie de estructuras huecas con diferentes configuraciones, en el que se replican patrones. Estas estructuras, que tienen un patrón periódico, se denominan scaffolds, y cuentan con una amplitud de campos de aplicación en Ingeniería Tisular.

Los scaffolds consisten en estructuras porosas con diferentes configuraciones o arquitecturas, que permiten la adhesión, el crecimiento y la proliferación celular [7]. Estos dispositivos se fabrican generalmente mediante técnicas de fabricación aditiva [8]. Recientemente, la aparición de nuevas técnicas de fabricación como el Modelado por Deposición Fundida (FDM, por sus siglas en inglés, o extrusión de material, MEX, según la norma ISO 52900), el Sinterizado Selectivo por Láser (SLS, por sus siglas en inglés, o fusión de lecho de polvo con láser y polímero, PBF-LB/P, según la norma ISO 52900), Impresión 3D y Deposición por Extrusión de Precisión (PED, por sus siglas en inglés) permiten fabricar piezas con arquitecturas más complejas [9] (Figura 2-1).

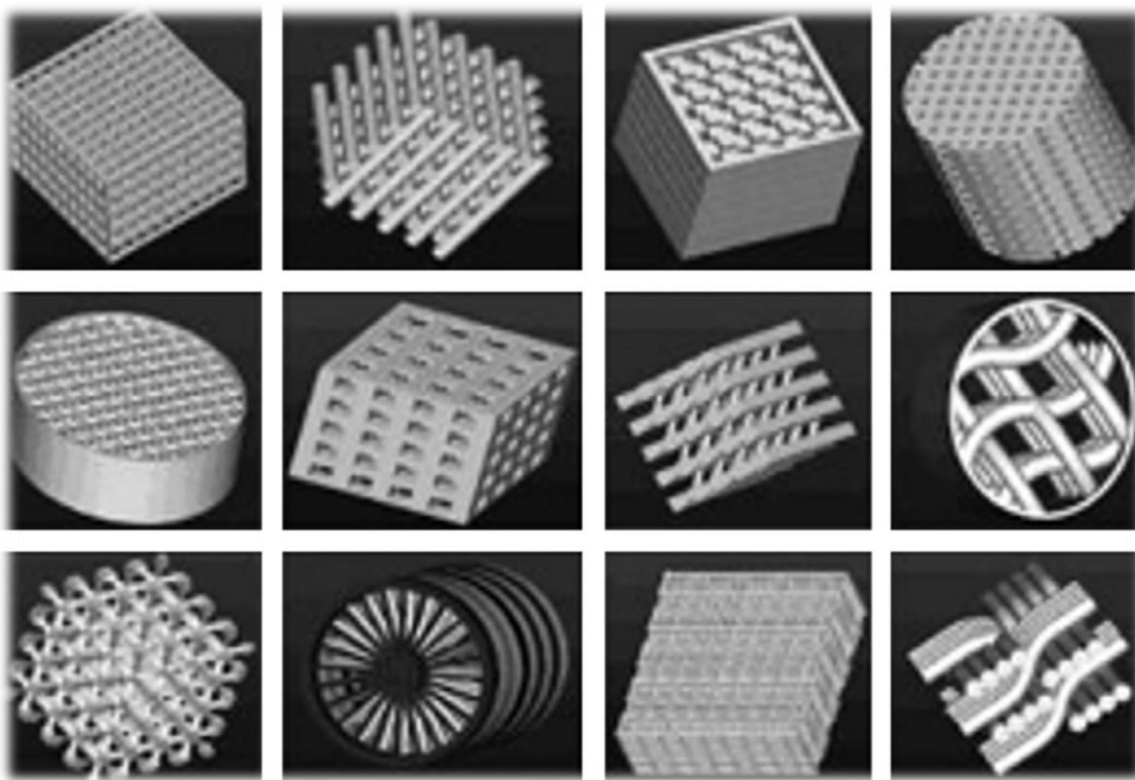


Figura 2-1. Representación de las distintas topologías que representan los scaffolds. Extraída de [10].

En relación con los tejidos óseos, los materiales empleados suelen ser polímeros (de origen natural y sintético), materiales metálicos derivados del titanio, así como materiales cerámicos con propiedades bioactivas [11], [12], [13]. Por su parte, los tejidos blandos tanto para la aplicación en tejidos blandos como para evitar intervenciones posteriores intrínsecas a los implantes metálicos (que requieren su sustitución cada cierto tiempo), la tendencia más efectiva es el empleo de materiales bioabsorbibles como algunos polímeros alifáticos, principalmente PLA y PGA, o combinaciones de ambos, entre otros. Estos materiales están pensados para proveer de rigidez mecánica al tejido dañado que, a medida que se va regenerando, va recuperando su capacidad portante, en detrimento del scaffold que será progresivamente disuelto (materiales biocompatibles y bioabsorbibles). Asimismo, una vez implantados, los scaffolds experimentan un cambio en su estructura química debido a la acción de las moléculas de agua que se difunden en el interior de la estructura, provocando las roturas de las cadenas poliméricas, así como una erosión debida a la exposición del material al medio y/o al paso del flujo sobre la superficie.

En las últimas décadas, ha aumentado notablemente el interés por emplear **polímeros bioabsorbibles con diferentes propiedades** en la ingeniería tisular, especialmente para la regeneración de tejidos blandos como los cartilaginosos de las articulaciones [14], [15]. Es crucial estudiar cómo se comportan estos dispositivos mientras están insertados en el tejido dañado, ya que esto es fundamental para lograr una reparación óptima del tejido afectado. En la fase de diseño de los scaffolds bioabsorbibles, **es fundamental considerar que la**

**velocidad de descomposición de la estructura debe estar en sintonía con la velocidad de regeneración del tejido. Dada la importancia de esta característica, resulta interesante contar con herramientas numéricas que permitan estimar la degradación de la geometría y propiedades intrínsecas de scaffolds bioabsorbibles con el paso del tiempo.**

Para estimar el comportamiento de los polímeros bioabsorbibles es necesario tener en cuenta dos fenómenos básicos: la erosión y la degradación. La erosión hace referencia a la pérdida efectiva de masa del material polimérico, y se clasifica en dos partes: erosión en masa y erosión superficial [16]. En el primero no se manifiestan cambios significativos en la morfología del material, mientras que en el segundo caso se hacen más visibles los cambios de morfología del polímero biodegradable en las diferentes etapas del proceso. Burkersroda ha definido una formulación teórica para determinar, en términos promedio, si un material experimenta erosión superficial o erosión en masa [17] a través de un parámetro denominado *número de erosión*.

En relación con las condiciones del proceso degradativo y erosivo de scaffolds bioabsorbibles, pueden darse en in-vivo o in-vitro.

En lo que respecta al proceso degradativo de los polímeros, este puede ser del tipo hidrolítico, enzimático, foto degradativo, térmico y oxidativo [18], [19].

Diversos autores han planteado diferentes modelos que describen la degradación de los polímeros, y en gran medida se ha asociado a un proceso de hidrólisis de los polímeros bioabsorbibles. Este proceso puede ser hidrolítico puro o enzimático. Esto consiste en que el agua se inserta en la estructura polimérica a través de los poros y rompe las cadenas por donde se presentan los radicales de éster. Históricamente, se ha recurrido a un modelo de degradación que define el fenómeno de la hidrólisis como un evento probabilístico, por medio del modelo de Montecarlo [20], [21]. Erkizia et al. [7], por su parte, demostraron con esta metodología un buen ajuste a los resultados experimentales.

El proceso de erosión se considera como un proceso de mayor complejidad, que depende de varios factores, incluido la degradación, que interactúa con este de forma continua [22]. El proceso de erosión de una partícula se considera como un evento aleatorio que no se puede predecir con total seguridad, pero la probabilidad de que se produzca es conocida en todo momento. Göpferich et al. [21], [23] definieron un modelo de erosión en el que se subdivide la geometría [8] en píxeles iguales, asignando a cada uno de ellos un valor de probabilidad de acuerdo con una función matemática según el tipo de estructura molecular, cristalina o amorfa.

Diversos autores han abordado el proceso de degradación mecánica de los polímeros. Se ha demostrado en base a ensayos experimentales que existe un valor de tensión tangencial límite [24], por encima del cual se inicia la degradación mecánica del material. Esta se produce principalmente en la entrada del flujo, pero también en zonas en la que este se encuentra totalmente desarrollado [25].



M.Saad et al. [26], a partir de ensayos in vitro dentro de un biorreactor, reconstruyeron la geometría del scaffold a partir de microtomografía computarizada en distintos instantes de tiempo y simularon por CFD el correspondiente mapa de tensiones tangenciales en las diferentes etapas en las que se ha realizado mediciones. Chen et al [27] plantearon este fenómeno como una consecuencia del movimiento del flujo circundante, a través del campo de tensiones tangenciales en la superficie externa del material y modificando la geometría de tal manera que el campo resultante fuera lo más homogéneo posible. Sin embargo, el valor seleccionado de tensiones tangenciales es arbitrario. Se ha observado que la degradación del material se acelera para valores menores de porosidad [28] además de que bajo condiciones dinámicas el material se degrada a menor tasa con respecto a las condiciones estáticas debido a la liberación de subproductos.

Diferentes modelos han intentado representar con mayor o menor exactitud el proceso de degradación al que se somete el dispositivo en contacto con el líquido. Principalmente estos procesos se centran en la obtención de la degradación hidrolítica a la que se somete el implante al estar en contacto con el líquido.

Por otra parte, está el fenómeno erosivo, que se representa en muchos casos como una consecuencia de la exposición del material al medio acuoso.

La degradación y erosión llevan consigo una modificación en las propiedades mecánicas del polímero, aunque no se presenta de una manera inmediata. En el caso de la degradación, existe una relación que permite calcular el módulo de elasticidad en base al número de cadenas poliméricas cuyo módulo de elasticidad sobrepase el valor umbral o crítico del peso molecular [29]. A esto hay que añadir que las cadenas resultantes del proceso de hidrólisis no se suman al número de cadenas que aparecen. Con respecto al aspecto erosivo no hay un estudio claro de cómo funciona el proceso. En relación con las propiedades mecánicas, hay diferentes metodologías que permiten estimar la pérdida de rigidez del material en las diferentes fases del proceso.

### 3. OBJETIVOS

En este epígrafe se definen los objetivos del presente trabajo, cuya finalidad principal es desarrollar una herramienta numérica que sea capaz de predecir la degradación geométrica y de propiedades mecánicas (módulo de elasticidad) de un scaffold biodegradable, partiendo de una serie de inputs o parámetros de partida como el modelo geométrico o las propiedades del material. De esta manera, se pretende que el resultado de esta herramienta numérica permita obtener la geometría y propiedades intrínsecas estimadas de un scaffold biodegradable en distintos momentos del proceso de degradación. Esto a su vez permitiría estimar el comportamiento mecánico con otras herramientas numéricas, creando así un flujo de trabajo y metodología que sería útil para optimizar el diseño de scaffolds bioabsorbibles en los que se busca, entre otros aspectos, que la tasa de degradación de la estructura porosa esté en consonancia con la tasa de regeneración del tejido.

Inicialmente, el plan de investigación de este trabajo contemplaba como objetivos el desarrollo de una herramienta numérica que permitiera predecir la degradación del polímero bajo condiciones *in vitro* dinámicas basadas en la modificación de la geometría, definiendo para ello un valor umbral de tensiones tangenciales a partir del cual se erosiona el material y, posteriormente, analizar la pérdida de propiedades mecánicas de la geometría degradada mediante simulaciones de elasticidad por elementos finitos. Después de realizar una revisión exhaustiva se ha visto que un mayor flujo del medio normalmente reduce la tasa de degradación debido a la remoción de los oligómeros derivados del proceso de degradación [28], [30]. Por tanto, la hipótesis de partida y objetivos especificados inicialmente se fueron adaptando en base a los hallazgos encontrados durante el desarrollo de la tesis. En concreto, tras ver las limitaciones de esa hipótesis inicial (más tensiones tangenciales conllevan mayor degradación), se planteó como objetivo desarrollar un modelo estocástico, basado en autómatas celulares, que permitiera predecir la erosión superficial en base al grado de exposición al medio. Este modelo se alimentó, en un principio, de valores experimentales, para luego extrapolar esa metodología usando criterios puramente geométricos. Tras ello, al no observarse un ajuste lo suficientemente razonable entre los resultados obtenidos de la simulación y los valores de referencia de los ensayos, se orientó el trabajo en realizar un enfoque combinado que estudiara, por una lado, las variaciones a nivel morfológico de la pieza debidas a la erosión superficial, y, por otro, la variación de propiedades intrínsecas debidas a la degradación del material, siendo este último el que se produciría en primer lugar para dar luego paso al proceso erosivo.

## 4. ESTADO DEL ARTE.

Un scaffold bioabsorbible, durante la vida útil, sufre una serie de procesos fisicoquímicos en EROSIÓN SUPERFICIAL Y EROSIÓN EN MASA

Durante el proceso de degradación, dependiendo del grado de hidrofobicidad del material, la pérdida de material se producirá bien desde la superficie o a través de su núcleo. A este proceso de reducción de la masa se le llama erosión y es un aspecto que juega un papel clave de forma paralela al proceso de degradación.

La erosión de los polímeros se produce de dos maneras:

A través de que afectan tanto a su morfología y propiedades macroscópicas, así como a sus propiedades intrínsecas, culminando en última instancia en la completa disolución del polímero. Atendiendo al proceso de pérdida efectiva de masa, encontramos dos fenómenos claramente diferenciados: erosión superficial y erosión en masa. En este capítulo se ahondará en cada uno de ellos y en los fenómenos más importantes que intervienen en el proceso degradativo del material.

Por otra parte, se encuentra el fenómeno degradativo, en el que al material va disminuyendo sus propiedades intrínsecas tales como el peso molecular o el módulo de elasticidad. Entre los mecanismos degradativos existentes se prestará especial atención al fenómeno hidrolítico, en el que el medio acuoso al insertarse dentro de la estructura del material produce la rotura de las cadenas poliméricas dentro del mismo.

- la superficie, donde el material se desprende progresivamente hasta obtener una diferencia en las densidades alta. Esto se denomina **erosión superficial**. Esto puede deberse principalmente a la exposición del material al medio o a las tensiones tangenciales debidas al movimiento del flujo en la superficie.
- De forma interna, cuando las moléculas de agua se difunden en el medio provocando, mediante la rotura sucesiva de cadenas, la reducción del peso molecular del material y la pérdida de masa por medio de la liberación de oligómeros a través de los poros hacia el medio acuoso. Esto se traduce, además de un decremento en las propiedades intrínsecas del material, en una eventual pérdida de la masa efectiva del polímero, sin modificar su morfología externa, pero sí su grado de porosidad microscópica. Mediante la difusión de las moléculas de agua en la estructura polimérica, se produce la rotura de las cadenas, y, consecuentemente, difusión de oligómeros a través de la superficie. Este fenómeno se denomina **erosión en masa** (bulk erosion en inglés).

Según Burkersroda et al [31], dos parámetros juegan un papel importante en la cinética de erosión (Figura 4-1):

1. La difusión del agua en el núcleo polimérico (D)
2. La tasa de degradación de la estructura ( $\lambda$ ).

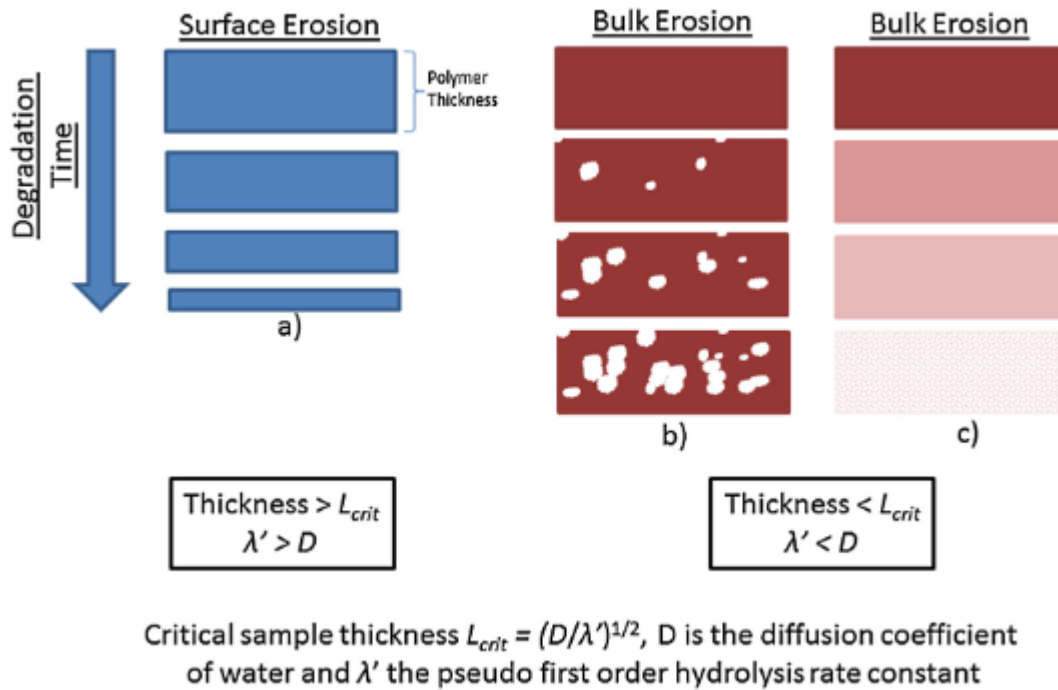


Figura 4-1. Representación del mecanismo de erosión en polímeros bioabsorbibles. Figura extraída de [32].

En base a esto, si un polímero presenta una tasa de degradación más elevada que la correspondiente a la difusividad, predominará una erosión superficial. En cambio, si la difusión de las moléculas de agua en el medio se produce a una velocidad mayor que la degradación, predominará la erosión en masa. Este proceso no se produce de forma independiente, es decir, que puede haber materiales en los que se produzca al mismo tiempo la degradación y erosión superficial [33]. Para ver si en un punto del material hay erosión superficial o en masa, Burkersroda et al. [31] definió un parámetro denominado número de erosión ( $\epsilon$ ) que relaciona el tiempo de difusión ( $t_{diffusion}$ ) y el tiempo de erosión ( $t_{erosion}$ ), para decidir si una matriz polimérica se somete a erosión o degradación, de la siguiente forma:

$$\text{Erosion Number} = \frac{t_{diffusion}}{t_{erosion}} = \frac{\langle x \rangle^2 \pi * k}{4D * (\ln \langle x \rangle - \ln \sqrt[3]{\frac{M}{N_A(N-1)*p}})} \quad (4-1)$$

donde

- $\langle x \rangle$  representa la distancia media que deben viajar las moléculas de agua para difundirse en la matriz polimérica,

- $\lambda$  es una constante que tiene en cuenta las diferencias en reactividad de los distintos grupos funcionales.
- $D$  es el coeficiente de difusión efectiva.
- $N_A$  es el número de Avogadro.
- $M$  representa el valor promedio (en número) del peso molecular.
- $N$  es el valor promedio del grado de polimerización
- $\rho$  es la densidad del material polimérico.

#### 4.1. DEGRADACIÓN HIDROLÍTICA.

La degradación hidrolítica de polímeros es un proceso complejo que involucra la interacción de medios acuosos con polímeros bioabsorbibles como polisacáridos, poliésteres y sus copolímeros alifáticos y aromáticos, y poliamidas. Estos polímeros pueden degradarse a través de varios mecanismos, incluyendo la foto-degradación, o la degradación térmica, mecánica y química, que pueden actuar solos o en combinación, a menudo de manera sinérgica [34].

La tasa de degradación hidrolítica depende de varios factores, como la longitud de la cadena polimérica, la cristalinidad, la distribución del peso molecular, la capacidad de hinchamiento, el tamaño, la forma y la geometría de la matriz polimérica, el pretratamiento de la superficie, la porosidad, el tamaño y la distribución de los poros, la geometría de los poros, las dimensiones generales del dispositivo, las condiciones de procesamiento y la difusividad del agua en la matriz polimérica [35], [36]. La cristalinidad, en particular, limita la difusión del agua y, por lo tanto, restringe el acceso a los enlaces hidrolíticamente lábiles, haciendo que las regiones amorfas se degraden más rápidamente [37].

Un modelo fenomenológico destacado es el de Wang et al. (2008), que presenta un modelo de difusión-reacción para la biodegradación de polímeros bioabsorbibles [1]. Este modelo considera factores como la hidrólisis, la cristalinidad del material y la auto catálisis, donde ciertos productos de la hidrólisis pueden acelerar aún más la degradación del material.

La evaluación de la degradación se realiza mediante estudios *in vitro* e *in vivo*. Los estudios *in vitro* proporcionan información sobre las tasas de degradación y el comportamiento de la erosión en soluciones acuosas, mientras que los estudios *in vivo* son más relevantes para evaluar la utilidad de los dispositivos biomédicos en el entorno fisiológico. La erosión física que acompaña a la degradación hidrolítica se clasifica como erosión en masa o erosión superficial, dependiendo de si la pérdida de masa ocurre en todo el material o se limita a la superficie del espécimen.



#### **4.2. ESTUDIO DE LA EROSIÓN DINÁMICA DEBIDA A LAS TENSIONES TANGENCIALES.**

Como hemos dicho, la erosión es debida principalmente a que la velocidad de degradación supera la velocidad de difusión del líquido circundante. Algunos autores han visto una relación causa efecto entre las tensiones tangenciales presentes en la superficie del polímero debidas al flujo, y la erosión que se produce en éste. Aunque es cierto que está mucho más extendida la idea de que un cierto nivel de tensiones tangenciales favorece la adhesión celular y la respuesta mecano-biológica, algunos autores remarcaron el papel que juegan las tensiones tangenciales en el proceso de degradación polimérica, concretamente en la fase de erosión superficial.

#### **4.3. TIPOS DE MODELOS PARA SIMULAR EL COMPORTAMIENTO DEGRADATIVO DEL MATERIAL POLIMÉRICO.**

Según la literatura, los diferentes modelos presentados para modelar o predecir el comportamiento de un polímero biodegradable se pueden definir en los representados en el siguiente esquema [38], [39] (Figura 4-2):

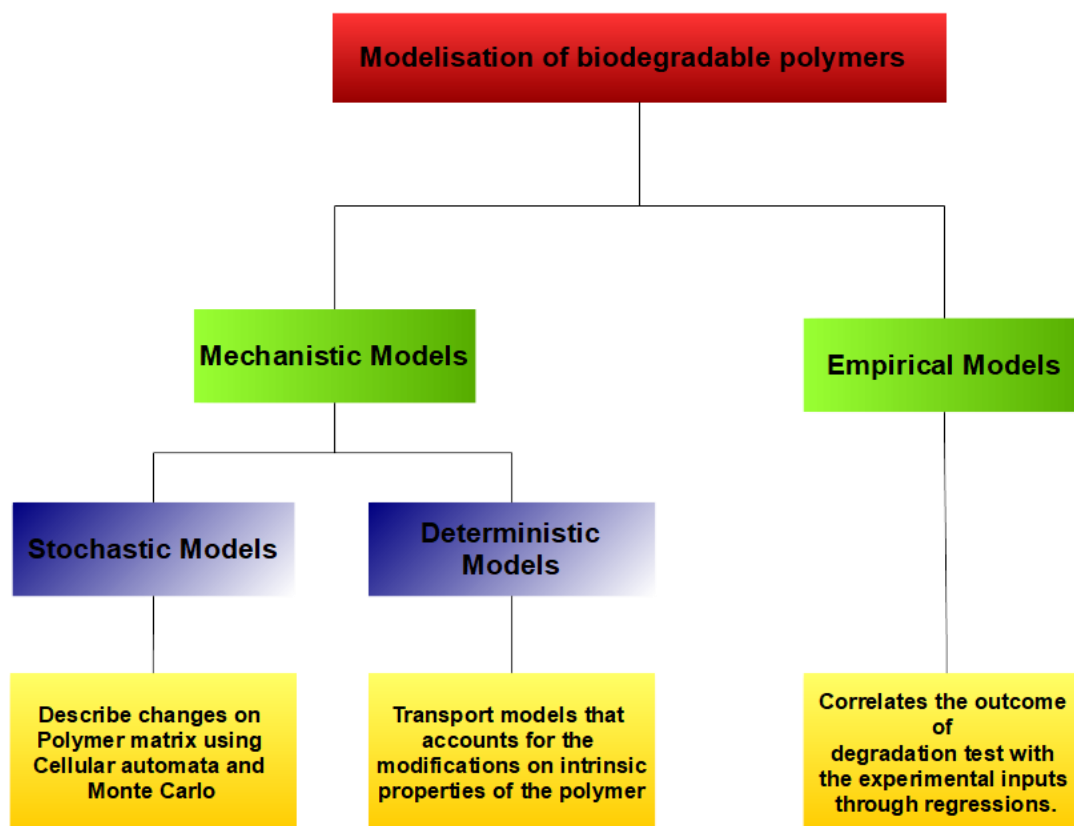


Figura 4-2. Esquema conceptual de las diferentes metodologías empleadas para la modelización de la degradación de polímeros.

Los diferentes modelos que se han presentado en la literatura se pueden clasificar en dos grandes grupos: los modelos mecanicistas y los modelos empíricos o experimentales. Los modelos mecanicistas se tratan de modelos que permiten describir el comportamiento de sistemas de mayor o menor complejidad a partir de la comprensión del mecanismo físico o químico que rige el comportamiento del sistema. Esto se traduce en el uso de ecuaciones de gobierno deterministas o modelos de carácter estocástico o probabilístico en los que se describe el comportamiento del sistema bajo condiciones dinámicas o estáticas. Estos modelos tienen aplicabilidad en diferentes ámbitos como pueden ser la ingeniería y las ciencias médicas, entre otras.

#### 4.3.1. MODELOS DETERMINISTAS.

Estos modelos deterministas o fenomenológicos describen el proceso de degradación a través de modelos de transporte regidos por los procesos de difusión, reacción y disolución [1]. Fenómenos como la erosión superficial debida a la aparición de tensiones tangenciales como consecuencia del movimiento del líquido circundante también han sido abordados [27]. Para el cálculo de las propiedades mecánicas también se recurren a modelos deterministas que relacionan el módulo de elasticidad en función de las propiedades

intrínsecas (por ejemplo el peso molecular y el número de cadenas presentes cuyo peso molecular quede por encima de un umbral) [40], [41].

#### 4.3.2. MODELOS ESTOCÁSTICOS/PROBABILÍSTICOS

Los modelos estadísticos describen los cambios en la matriz polimérica (concentración polimérica, cristalinidad, densidad, etc.) sobre una base estocástica/probabilística. Göpferich introdujo el estudio de una metodología basada en una distribución de elementos cristalinos y amorfos. Las simulaciones de Monte Carlo han sido frecuentemente utilizadas para describir estos modelos [7], [21]. En el caso del cálculo de la evolución de las propiedades mecánicas, de la rotura de cadenas poliméricas se recurre también a un modelo estadístico/probabilístico, para decidir qué tipo de rotura se produce [40], [41]. Para decidir donde se realiza la escisión de la cadena se asigna una probabilidad de rotura dependiendo de si esta se produce en los extremos o de forma aleatoria. La metodología asociada al método de Monte Carlo se ha utilizado también en el estudio de la liberación de fármacos durante el proceso degradativo del material polimérico [38], [42].

#### 4.3.3. MODELOS EMPÍRICOS

Los modelos empíricos se basan en enfoques heurísticos que utilizan correlaciones estadísticas tales como regresiones, etc., para relacionar analíticamente los datos numéricos obtenidos de las observaciones de un experimento dado. Un aspecto a remarcar de los modelos empíricos es que son capaces de determinar cuál o cuáles son los factores de mayor peso en el proceso que se estudia [38]. Estos modelos aportan una comprensión menor acerca del mecanismo del proceso, estableciendo una relación específica que se cumple para las condiciones específicas que se están ensayando.

En relación a la modelización de los polímeros bioabsorbibles, Cooney et al. [43] estableció un modelo experimental que relaciona la tasa de liberación de fármacos con la superficie del material polimérico a través de la cual se libera. Por su parte, Hopfenberg et al. [44], estudió la influencia de la erosión superficial en el proceso de liberación controlado de fármacos aplicándolo a diferentes topologías: esferas, cilindros y placas rectangulares. Un mayor número de datos iniciales permitirá que el modelo se comporte de manera más representativa.

Tanto los modelos fenomenológicos como los estadísticos son mecanicistas, puesto que establecen una relación causal entre el proceso que ocurre y el resultado que se da. En cambio, los modelos empíricos establecen una relación específica que se cumple para el caso concreto que se está ensayando y rara vez se puede extrapolar a diferentes casos que se estudien.

Por último, existen modelos que combinan las simulaciones estadísticas, simulando la erosión superficial, con los modelos fenomenológicos para el cálculo de propiedades hidrolíticas [45]. Recientemente se han utilizado los resultados experimentales como parámetro de entrada para simular la degradación enzimática por Monte Carlo [46].

## 5. MÉTODOS Y HERRAMIENTAS

### 5.1. MÉTODOS NUMÉRICOS EMPLEADOS EN LOS CÁLCULOS COMPUTACIONALES.

#### 5.1.1. MÉTODO DE LOS ELEMENTOS FINITOS.

Se trata de una técnica de resolución numérica de ecuaciones en derivadas parciales. Consiste en discretizar el dominio en elementos de una tipología determinada (triángulos, elementos cuadráticos, tetraedros, hexaedros, etc.) y resolver las ecuaciones diferenciales en cada uno de los nodos usando una notación matricial. Para ello, este modelo se sirve de la formulación débil o variacional para su resolución, describiendo el problema en derivadas parciales a través de integrales de dominio y contorno multiplicando cada uno de los términos de la ecuación por una función test. Con el incremento de las capacidades de los ordenadores, así como de las estaciones de cálculo y clústeres, este modelo se ha erigido como uno de los principales métodos para la modelización de diversos sistemas que queden descritos a través de sus ecuaciones diferenciales. Inicialmente este método fue concebido para el cálculo mecánico de estructuras metálicas.

Existen diferentes herramientas que nos permiten simular por elementos finitos. Muy conocidos son los softwares comerciales, ANSYS, Abaqus, SolidWorks, etc., [47] softwares robustos que, a través de diferentes modelos preinstalados, simulan el comportamiento de diferentes sistemas de diversa complejidad, generalmente imponiendo unas condiciones iniciales y de contorno, así como ajustando una serie de parámetros. Asimismo, estas herramientas permiten la posibilidad, aunque con ciertas limitaciones, de definir modelos personalizados, así como sus parámetros a través de una ventana de código. Por otra parte, existen herramientas de que permiten modelar el comportamiento de casi cualquier sistema a través de líneas de código. En este grupo se encuadran herramientas genéricas como Matlab, Python, Octave, que disponen de librerías o extensiones para el modelado por elementos finitos, con funciones específicas, así como otras herramientas más específicas que permiten modelar casi cualquier sistema de forma relativamente sencilla a través de líneas de código para la resolución de ecuaciones diferenciales, a través de funciones específicas. En este subgrupo se encuentran también otros programas, como FreeFem++ [48], ElmerFem y OpenFoam [49], basados en lenguajes de programación de alto nivel, que permiten modelar cualquier sistema de una forma relativamente sencilla a partir de sus ecuaciones de gobierno. La desventaja de estas herramientas es que, al carecer de interfaz gráfica, o tener una interfaz muy limitada, la parte correspondiente al preprocesado del dominio y discretizado y el postprocesado o análisis de los resultados se obtiene generalmente a través de terceros programas como Gmsh [50], [51], y programas CAD.

### 5.1.2. SIMULACIONES DE MONTE CARLO.

Las simulaciones de Monte Carlo se conciben como una colección de técnicas computacionales que permiten simular procesos de naturaleza arbitraria, mediante el uso de muestras aleatorias, y con el uso de series no relacionadas de números aleatorios que forman una cadena de Markov [52], [53]. Las simulaciones de Monte Carlo, junto con las cadenas de Markov, juegan un rol importante en la estadística Bayesiana. Los métodos de Monte Carlo confían en la ley de “grandes números” (*law of large numbers*, en inglés): la distribución de una muestra cada vez más grande debe converger a la par con la distribución de la población subyacente [54].

### 5.1.3. AUTÓMATAS CELULARES.

El método de los autómatas celulares (CA, por sus siglas en inglés) ha sido una metodología que ha capturado el interés de diversas disciplinas debido a su estructura simple y su capacidad para modelar sistemas complejos. Su origen se remonta a los años 40 , cuando J. Von Neumann y Stanislaw Ulam, trabajaban en el *Proyecto Manhattan* en el laboratorio de Los Álamos). Este trabajo dio forma al primer diseño del modelo de Von Neumann [55]. Los CA han sido objeto de rigurosos análisis matemáticos y físicos. Este interés se ha manifestado en la publicación de numerosos artículos de investigación anuales, la organización de conferencias especializadas y la oferta de cursos universitarios sobre CA.

El atractivo de los CA radica en su simplicidad y su potencial para modelar fenómenos naturales sofisticados. Un autómata celular es un modelo de un sistema descentralizado compuesto por celdas individuales que interactúan localmente. A pesar de su simple estructura, las iteraciones de estas interacciones locales pueden generar patrones complejos. Von Neumann demostró que un CA puede ser universal, capaz de realizar cualquier operación computable, aunque su complejidad impidió su implementación en computadoras.

Investigadores como John Conway y Stephen Wolfram han realizado contribuciones significativas al campo. Conway, en la década de 1970, publicó el Juego de la Vida [56], mientras que Wolfram, en los años 80, estudió reglas unidimensionales simples que demostraron la capacidad de los CA para emular comportamientos complejos.

La investigación en CA se ha enfocado en dos objetivos principales: predecir el comportamiento global a partir de las reglas locales y diseñar reglas locales para alcanzar un comportamiento global específico. Este estudio presenta un panorama actualizado de la teoría y aplicaciones de los CA, destacando la diversidad de conceptos y enfoques propuestos por los investigadores. En resumen, se busca entender cómo las reglas locales pueden influir en las dinámicas globales y diseñar CA para modelar sistemas complejos de manera práctica.

#### 5.1.4. MODELOS DE SIMULACIÓN POR DOMINIOS FICTICIOS.

También llamados Métodos de Dominio Embebido [57]. Se trata de una técnica numérica derivada del método de los elementos finitos especialmente diseñada para el modelado de ecuaciones en derivadas parciales definidas en dominios con fronteras complejas y/o deformables. El método consiste en extender el dominio original a una región simplificada, facilitando el discretizado y evita el uso de un remallado constante, resultando en un sistema más eficiente para la modelización de sistemas. Tiene una gran aplicabilidad en el caso de los flujos particulados.

En relación con el ámbito de estudio de esta tesis, diversos autores han realizado estudios computacionales usando el modelo de dominios ficticios [58], [59] para estimar el comportamiento de la erosión del material bajo condiciones dinámicas y/o estáticas, diversos autores han optado por esta metodología para representar el comportamiento erosivo del material en aras de lograr un modelo numéricamente más estable.

## 5.2. HERRAMIENTAS UTILIZADAS.

Durante la realización de este trabajo se han optado por las diferentes herramientas de modelizado, discretizado, simulación y postproceso que se listan a continuación:

# SALOME

o SALOME:

Desarrollada inicialmente  
fundamentalmente por una

asociación entre la empresa eléctrica EDF (Electricité de France) y el organismo estatal de Investigación CEA (Commissariat à l'Énergie Atomique et aux Énergies Alternatives) y la empresa de tecnologías de comunicación OPENCASCADE [60], se trata de una herramienta integrada de código destinada principalmente al preprocesamiento y posprocesamiento de modelos [61], aunque cuenta con algunos módulos para resolución de problemas mecánico-estructurales y fluidodinámicos, entre otros. Además, cuenta con un módulo de Python para la definición de funciones específicas. Dispone de una herramienta para el post-procesamiento llamado ParaVIS [62], basado en Paraview [63]. Se ha utilizado esta herramienta para la preparación y reparación de los modelos, así como el modelado de algunas piezas sencillas para su posterior utilización para el cálculo de las ecuaciones correspondientes de degradación y erosión del material.





# Gmsh

con los que se ha trabajado en esta tesis.

- GMSH [50], [51], [64]: Es un generador de mallas por elementos finitos con funcionalidades integradas de pre- y post-procesado. Permite la importación de ficheros geométricos de múltiples formatos como .step o .iges, aunque permite también la definición de geometrías a través de una serie de comandos por medio de un editor (Notepad, Notepad++, Visual Studio Code, etc.). Asimismo, dispone de diferentes modelos de mallador preinstalados como el modelo de Delaunay, el Frontal Delaunay, o el Netgen. Se ha utilizado esta herramienta para el discretizado y el etiquetado de los diferentes modelos



derivadas parciales por el método de los elementos finitos [48]. Tiene un mallador nativo en dos dimensiones llamado BAMG [65] que, a su vez, permite el tratamiento de mallas ya generadas como, por ejemplo, la adaptación de mallas, y que es compatible con varios programas de código abierto como GMSH o Paraview. En 3d también utiliza malladores nativos, aunque limitados, por lo que se recurre al uso de un mallador externo como Tetgen [66], [67], [68]. Para el refinamiento en 3d, se recurre a la librería anterior o bien a la librería MMG3D [69], [69], [70]. Permite el modelado del dominio de estudio a partir de la introducción de contornos de forma paramétrica variando la coordenada de un punto inicial u “origen” a un punto final o “extremo”. La programación de las ecuaciones diferenciales es bastante intuitiva puesto que se asemeja bastante a la formulación variacional clásica. Se trata de la principal herramienta de trabajo de esta tesis para el cómputo de los diferentes modelos aplicados.

- Freefem++:

Desarrollado inicialmente por Olivier Pironneau, se trata de un lenguaje de programación que se utiliza para el modelado de ecuaciones en



en simulaciones científicas, proporcionando un entorno interactivo para el postprocesado y

- ParaView [63] es una herramienta de visualización diseñada para manejar grandes volúmenes de datos generados

análisis de resultados. Su arquitectura permite el procesamiento paralelo y el renderizado distribuido, optimizando el uso de memoria y recursos computacionales en sistemas con memoria compartida o distribuida. Además, incorpora técnicas como el nivel de detalle y el streaming de datos para mejorar la eficiencia en el manejo de conjuntos de datos masivos. Estas características permiten a los científicos explorar sus simulaciones de manera interactiva y generar visualizaciones de alta calidad con un rendimiento optimizado, facilitando el análisis y la interpretación de resultados complejos.

### 5.3. PROCESAMIENTO A GRAN ESCALA.

La ejecución con geometrías más complejas en un entorno local implica una limitación a nivel de recursos, debido a la alta demanda de memoria necesaria que se puede utilizar para la ejecución de cada caso. Por este motivo, la utilización de recursos computacionales de altas prestaciones se antoja clave para la resolución de este tipo de simulaciones de forma eficiente.

#### 5.3.1. CENTRO DE PROCESAMIENTO DE DATOS DEL IUSIANI.

En una sala grande de sus instalaciones, el Instituto de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (IUSIANI) tiene un Centro de Procesamiento de Datos (CPD) para acceder a la información que necesita para sus operaciones y asegurar el servicio continuo a clientes, empleados, proveedores y empresas colaboradoras, protegiendo físicamente los equipos informáticos o de comunicaciones y los servidores de bases de datos con información relevante del instituto.

La primera parte del trabajo se realizó en el clúster Leibniz mientras que la segunda se realizó en el clúster Riemann los cuales se detallan a continuación.

#### 5.3.2. CLUSTER LEIBNIZ

Está formado por 28 nodos de cálculo y un nodo de acceso o *front-end*. Los nodos de cálculo se agrupan en siete equipos Bullx R424E2, cada uno con cuatro nodos de cálculo, y fuentes de alimentación redundantes. Cada nodo de cálculo está constituido por:

- 2 procesadores [Intel Xeon E5645 Westmere-EP](#), con 6 *cores* cada uno.
- 48 GB de memoria RAM.
- Un disco duro de 500 GB.
- Interfaz Infiniband a 40 Gbs.

El sistema operativo linux instalado, tanto en los nodos de cálculo como en el nodo de acceso, es Ubuntu 19.10 *Eoan Ermine* en su versión *server*.

Este recurso está formado por dos servidores:

- El servidor **vector** con las siguientes características:
  - 2 procesadores [Intel Xeon E5620](#) con cuatro *cores* cada uno.
  - 24 GB de Memoria RAM.
  - 1 disco duro de 1 TB.
  - 2 tarjetas NVIDIA Tesla M2050, de microarquitectura [Fermi](#).
  - Sistema Operativo Ubuntu 16.04.5 LTS *xenial*.
- El servidor **vectork40** con las siguientes características:
  - 1 procesador [Intel Xeon E5-2630 v4](#), con diez *cores*.
  - 256 GB de memoria RAM.
  - 1 disco duro de 3 TB.
  - 2 tarjetas NVIDIA Tesla K40, de microarquitectura [Kepler](#).
  - Sistema Operativo Ubuntu 16.04.5 LTS *xenial*.

En ambos servidores puede utilizarse el modelo de programación [CUDA](#) (*Compute Unified Device Architecture*) y el compilador de [PGI Community Edition](#).

### 5.3.3. RIEMANN

Dicho clúster se compone de los siguientes puntos:

Dos nodos de acceso, cada uno con:

- 1 CPU Intel Xeon E-2314
- 32 GB de RAM
- Almacenamiento:
  - 1 disco SSD de 450 GB
  - 1 disco HDD de 2 TB

24 nodos de cómputo, cada uno con:

- 2 CPU Intel Xeon Silver 4314
- 128 GB de RAM
- Almacenamiento:
  - 1 disco SSD de 450 GB
- Red de servicio Ethernet a 1 Gbps
- Red de aplicaciones, Ethernet a 100 Gbps

## 6. ESQUEMA GENERAL DEL TRABAJO.

El presente trabajo se estructura en 5 bloques principales, cada uno de los cuales se divide a su vez en 5 partes: revisión bibliográfica, definición matemática del problema, preprocesado del dominio, simulación del problema y post-procesado (Figura 6-1). En el primer bloque (correspondiente al [capítulo 7](#)), el efecto de la erosión mecánica se considera como consecuencia de las fuerzas de cizalladura que ejercen el medio dinámico sobre el material. A continuación, dadas las limitaciones observadas en la metodología anterior, se pasa a un segundo bloque ([capítulo 8](#)) con enfoque estocástico, en el que se estudia la influencia de los vecinos líquidos para proceder a remover la pieza polimérica. Se prosigue en los capítulos 9 y 10 con un enfoque fenomenológico, en el que se estudia la evolución de las propiedades intrínsecas (peso molecular y módulo de Young), poniendo especial énfasis en las propiedades mecánicas (módulo de Young). Por último, las técnicas anteriores (enfoque estocástico y fenomenológico) se integran en un método único ([capítulo 11](#)) que aporta una perspectiva más global de lo que ocurre en el material.

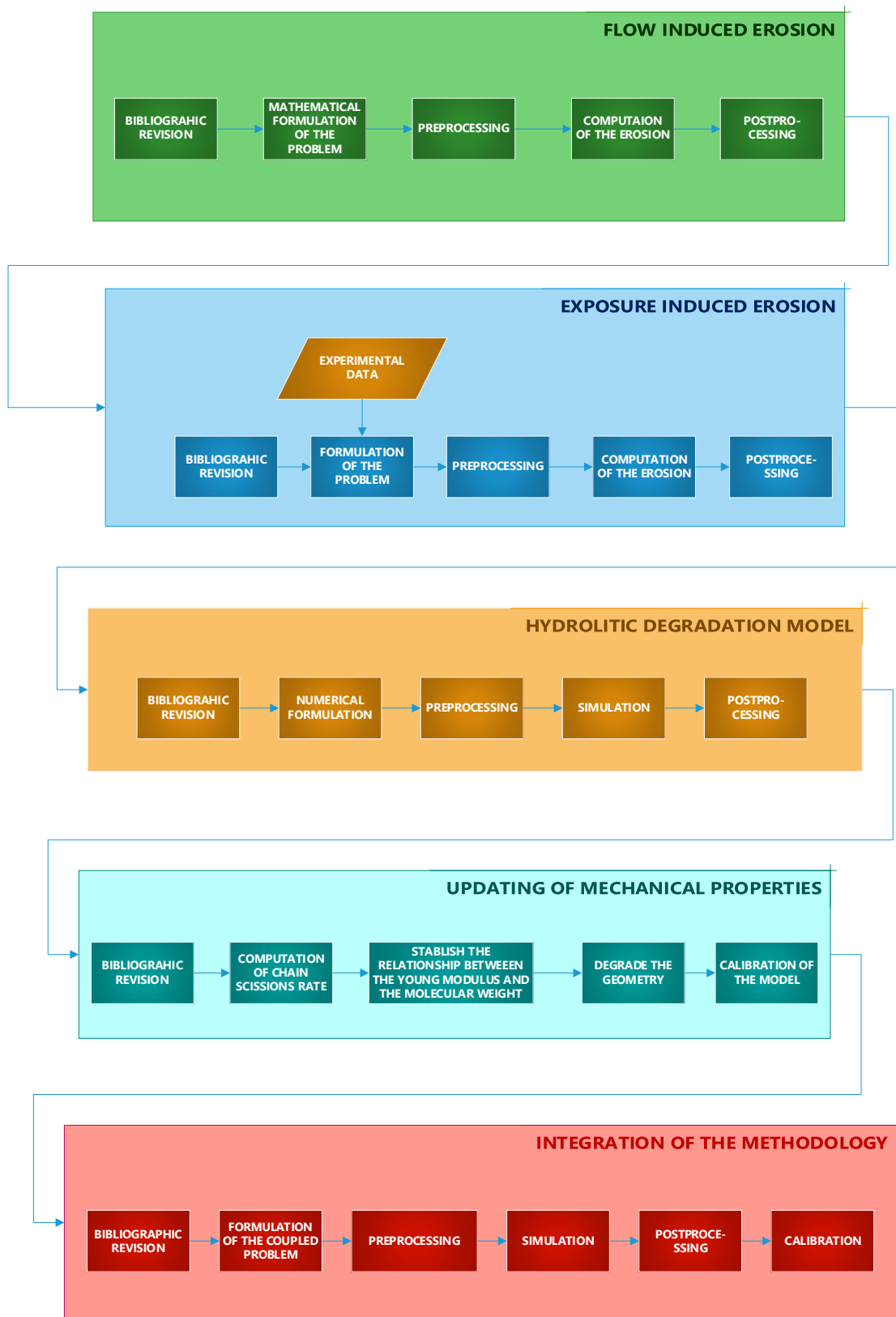


Figura 6-1. Esquema general del trabajo

## 7. TRABAJOS PRELIMINARES

### 7.1. APLICACIÓN DE UN MODELO DE EROSION A UN SCAFFOLD CILÍNDRICO BAJO CONDICIONES DINÁMICAS.

#### 7.1.1. INTRODUCCIÓN

Como se ha expuesto anteriormente, el diseño de scaffolds con materiales bioabsorbibles requiere, entre otros aspectos, de un buen ajuste entre la tasa de degradación del scaffold y la de regeneración del tejido nativo. A pesar de que en los capítulos anteriores se han presentado diferentes técnicas degradativas, este apartado se centra en el desarrollo de un modelo degradativo y erosivo del material basado en el efecto de las tensiones tangenciales, alineándose con la hipótesis de partida.

Se ha demostrado que las tensiones tangenciales conocidas como *Wall Shear Stress* (WSS, por sus siglas en inglés) ayudan a promover la adhesión celular y la respuesta mecano-biológica de los tejidos [71], [72] bajo condiciones in vitro dinámicas controladas. Se destaca que la **tensión tangencial de pared** es un **estímulo mecánico clave** en el cultivo celular dentro de **biorreactores de perfusión**, ya que influye directamente en la **diferenciación y proliferación de células madre** dentro de los scaffolds. Se ha demostrado que diferentes niveles de WSS pueden inducir la diferenciación de células madre mesenquimales (MSCs) hacia osteoblastos y osteocitos, lo que sugiere que la optimización de la arquitectura del andamio para controlar la distribución de la WSS podría mejorar la bioactividad y la eficiencia del cultivo celular. Diferentes autores han destacado la importancia del proceso de las tensiones tangenciales en el proceso degradativo. Culter encontró una relación entre las tensiones tangenciales y la disminución del peso molecular bajo condiciones de flujo [24]. Así, tenemos que la erosión mecánica es debida, en parte, a la presencia de tensiones tangenciales. Para que se produzca la erosión del material hace falta que la tensión tangencial local sobrepase un valor umbral previamente definido. Este valor umbral se debe ajustar a los valores experimentales de pérdida de masa. Chen et al [27] definieron una metodología en la que, a partir de la distribución de tensiones tangenciales, al definir una tensión umbral, cambia el estado de aquellos elementos ubicados en la frontera de sólido a líquido.

En este capítulo se expone un modelo numérico simplificado que aborda el proceso de erosión superficial desde la perspectiva expuesta por Chen.



### 7.1.2. METODOLOGÍA

En la etapa inicial de este trabajo se abordó la modelización numérica del proceso de erosión de scaffolds cilíndricos bajo condiciones in vitro, en base a las siguientes etapas:

1. **Preprocesado:** Aquí se llevan a cabo las tareas iniciales. Se modela la forma del dispositivo y se definen los espacios de estudio. Luego, se prepara geométricamente el entorno, se genera una malla para representar la estructura y se identifican las áreas donde se aplicarán condiciones específicas para el análisis. Este paso es crucial, ya que establece las bases geométricas y de condiciones para la simulación.
2. **Simulación:** Esta etapa constituye el núcleo del proceso. Se implementan los modelos que representarán la degradación y erosión de los materiales. Aquí se realizan los cálculos necesarios para cuantificar el avance y las variables relacionadas con el proceso de degradación. En este punto, se observa cómo los materiales se comportan bajo diferentes condiciones y se evalúa su comportamiento conforme a los modelos establecidos.
3. **Postprocesado:** Después de obtener los resultados de la simulación, se procede a su visualización y análisis. Esta fase incluye la representación gráfica de los datos obtenidos, lo que permite entender de manera más clara y visual el proceso de degradación. Se analizan los resultados para extraer conclusiones relevantes que ayuden a mejorar el diseño y comprender el comportamiento del material en estudio.

En resumen, el preprocesado establece las bases, la simulación ejecuta los cálculos y modelos para representar la degradación, y el postproceso permite visualizar y entender los resultados obtenidos de la simulación.

### 7.1.3. MODELADO GEOMÉTRICO

Entre las técnicas más utilizadas para el diseño de scaffold se encuentra el modelo de Ingeniería de Tejidos Asistida por Computadora (CATE por sus siglas en inglés), que es la aplicación de tecnologías asistidas por computador, tales como el CAD, el procesamiento de imágenes fabricación asistida por computador (CAM), prototipado rápido (Rapid Pototyping, RP), y/o la producción de geometrías libres sólidas (Solid Freeform Fabrication, SFF) para el modelado, diseño, simulación y fabricación de tejidos biológicos y sustitutivos de órganos [73].

La finalidad es conseguir que el dispositivo cumpla con ciertos requerimientos biológicos, mecánicos y geométricos. En lo referente a **requisitos biológicos**, la estructura debe facilitar la deposición y proliferación celular, así como la regeneración del tejido, el transporte de nutrientes y señales. Para ello se requiere de un tamaño de porosidad óptimo que garantice la interconectividad en el interior, así como un material que sea biocompatible. En el caso de **los requisitos mecánicos**, el scaffold debe proveer estructuralmente al tejido de un soporte estructural mientras se regenera. Los scaffolds deben poseer rigidez y resistencia equivalentes

a la estructura o tejido que se va a sustituir. Y por último, se encuentran los requerimientos anatómicos, que implican que el tamaño del dispositivo debe ajustarse al lugar que va a ocupar el nuevo tejido [74].

La estructura de los scaffolds consiste en la repetición de una celda unitaria en un dominio cilíndrico, prismático o tetraédrico. La forma de la celda es la que determina el tipo de scaffold. La elección de un tipo u otro dependerá de las condiciones a las que se vea sometido en el ensayo in vitro o directamente implantado en el órgano a reparar. Entre las arquitecturas utilizadas, las más utilizadas son la P- Schwarz, la giroide y el diamante.

Las estructuras de los scaffolds pueden ser periódicas o irregulares. Dentro de las estructuras regulares, existen varias formas para reconstruir la estructura del scaffold [10]:

- A. **Metodología basada en CAD:** es el método más utilizado actualmente. las tecnologías CAD generalmente trabajan con sistemas de modelado de sólidos y de superficies, como el de geometría sólida constructiva (CSG, Constructive Solid Geometry) y el de representación de límites (B-REP, Boundary REPresentation). El primero basado en la combinación de estructuras primitivas básicas (cilindro, cubo, esfera, etc.) y la aplicación de operadores booleanos (unión, diferencia, intersección, etc.), mientras el segundo el sólido es representado por su superficie externa, consistente en un conjunto de vértices, líneas y agrupaciones. En la Figura 2-1 se pueden ver algunos ejemplos de scaffolds elaborados mediante estos sistemas.
- B. **Diseño basado en toma de imágenes:** Es un método alternativo, inicialmente ideado para la fabricación de dispositivos personalizados para un paciente en concreto. Este método combina las técnicas actuales de procesamiento de imágenes, y los softwares de diseño con técnicas de fabricación de forma libre sólida (SFF, por sus siglas en inglés) para simplificar y acelerar el proceso de diseño de scaffolds. Las técnicas basadas en imágenes consisten en la intersección de dos imágenes binarias tridimensionales, una representa la forma de la patología a reparar, y la otra representa un conjunto de celdas binarias unitarias. El resultado obtenido definirá el patrón de los poros dentro de la estructura.
- C. **Diseño basado en superficies implícitas:** permite la creación de superficies complejas a partir de ecuaciones implícitas [75], [76], especificando propiedades como el tamaño de poro, y la variación de tamaños del poro. Estas tipologías se asemejan a diferentes superficies presentes en la naturaleza, tales como los caparazones de los escarabajos o las alas de las mariposas. Son denominadas superficies mínimas triplemente periódicas. Considerando una superficie  $S$  definida por:

$$S: F(X) = 0, X \in R^3 \quad (7-1)$$

$X$  es un punto de coordenadas  $x,y,z$ . Gabrielli et al [75] consideró una suma de productos de funciones trigonométricas de la siguiente forma:

$$1 + \sum_{c=1}^d a_c \sin^i x \sin^j y \sin^k z \cos^l x \cos^m y \cos^n z = 0 \quad (7-2)$$

Esto da lugar a superficies de nivel periódicas triplicadas: la primitiva (P), la giroide (G) y la diamante (D) (Tabla 7-1):

$P: a_1(\cos x + \cos y + \cos z) + a_2(\cos x \cos y + \cos y \cos z + \cos x \cos z) + 1 = 0 \quad (7-3)$	
$G: a_3(\cos x \sin y + \cos y \sin z + \cos z \sin x) + a_4(\cos 2x \cos 2y + \cos 2y \cos 2z + \cos 2x \cos 2z) + 1 = 0 \quad (7-4)$	
$D: a_5(\sin x \sin y \sin z + \sin x \cos y \cos z + \cos x \sin y \cos z + \cos x \cos y \sin z) + a_6(\cos(4x) + \cos(4y) + \cos(4z)) + 1 = 0 \quad (7-5)$	

Tabla 7-1. Ecuaciones implícitas para la generación de superficies TPMS.

Existen algunas herramientas que permiten modelar estas superficies ajustando el número de celdas unitarias como el tamaño de porosidad. Uno de los más utilizados en la literatura es el K3DSurf [77], que es un programa de código abierto que tiene una librería con diferentes tipologías de superficies implícitas. Recientemente, este programa ha sido sustituido por el MathMod [78], que está basado en el anterior.

#### 7.1.4. DEFINICIÓN DEL MODELO

El esquema de simulación se representa en la Figura 7-1. Inicialmente se parte de un modelo geométrico de la pieza a simular. En la etapa de preprocesado se imponen las condiciones de contorno de velocidad al dominio circundante, así como a la superficie. En la simulación se calculan las velocidades y se aplica el criterio correspondiente.

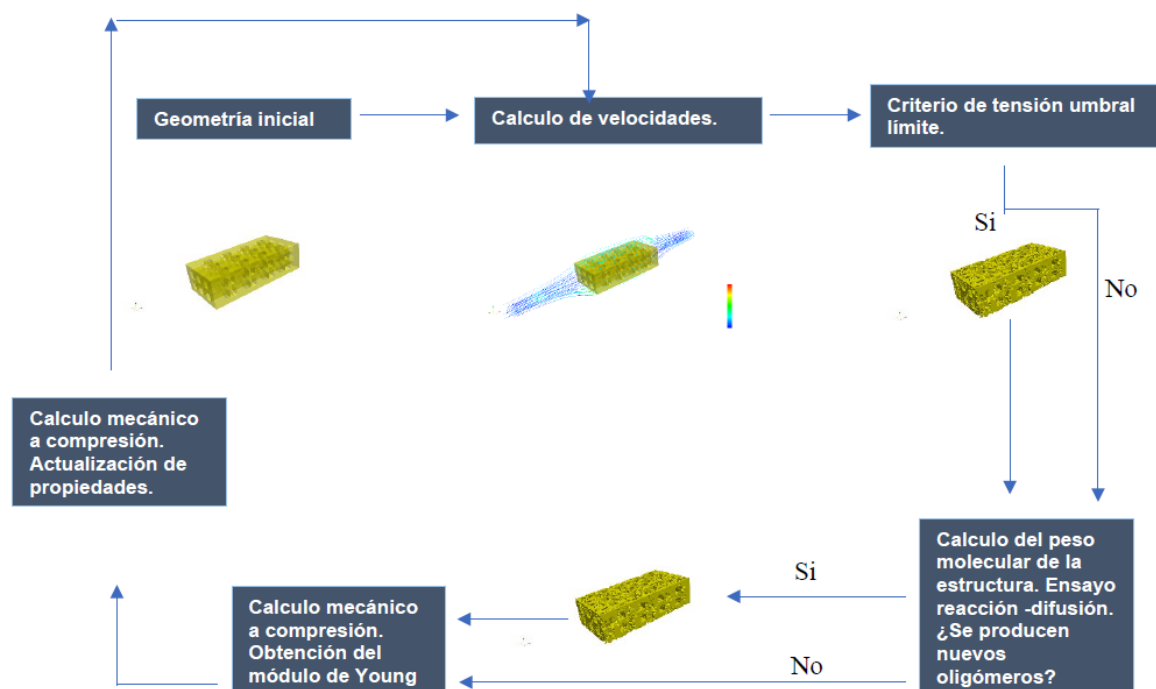


Figura 7-1. Esquema general de cálculo de la erosión y de propiedades mecánicas del scaffold bajo condiciones dinámicas.

En esta primera fase del trabajo, se ha trabajado con un modelo de scaffold cilíndrico definido mediante programas CAD por el grupo de Fabricación Integrada y Avanzada [18]. Desde el punto de vista de fabricación aditiva, normalmente se parte de un diseño sólido (cilindro macizo en este caso) al que posteriormente se le aplican los parámetros de fabricación deseados, como el tipo y densidad de relleno. A partir de esta definición de los parámetros de fabricación, se obtiene un fichero G-code que se usará en la impresora 3D. En este caso, la geometría empleada se obtuvo a partir de dicho fichero de fabricación, modelado posteriormente en programas CAD con metodologías que se han desarrollado dentro del grupo de Fabricación Integrada y Avanzada (Figura 7-2). Específicamente, se decidió trabajar con estas geometrías puesto que los resultados experimentales empleados como referencia en este trabajo (procedentes del proyecto BioAM) se obtuvieron a partir de ciertas geometrías obtenidas mediante fabricación aditiva, que luego se modelaron en CAD.

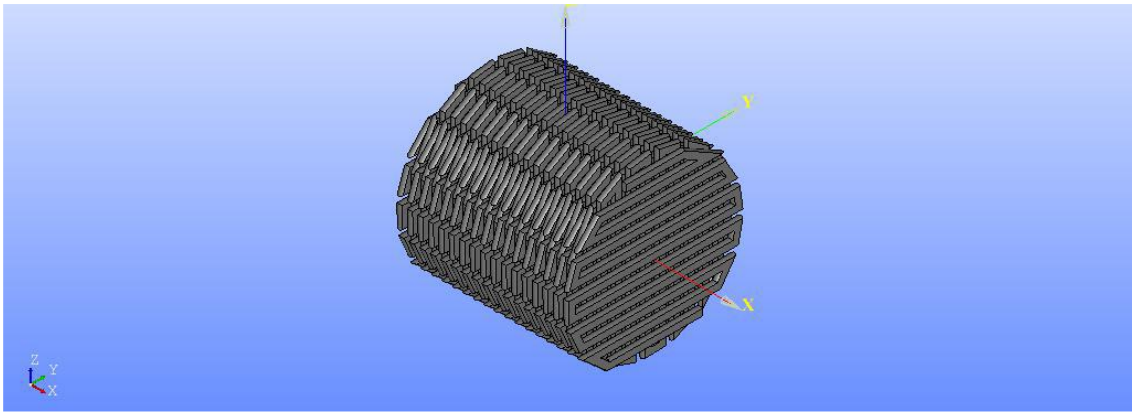


Figura 7-2. Modelo de scaffold.

El scaffold tiene una longitud en la dirección del eje X de 10 mm, así como un diámetro de 10mm y un tamaño de poro de 0,5 mm, aproximadamente.

Para reproducir las condiciones de un ensayo in vitro, el scaffold estará ubicado en el interior de una cámara de un biorreactor de acuerdo a las condiciones de ensayo definidas por Basri et al [26].

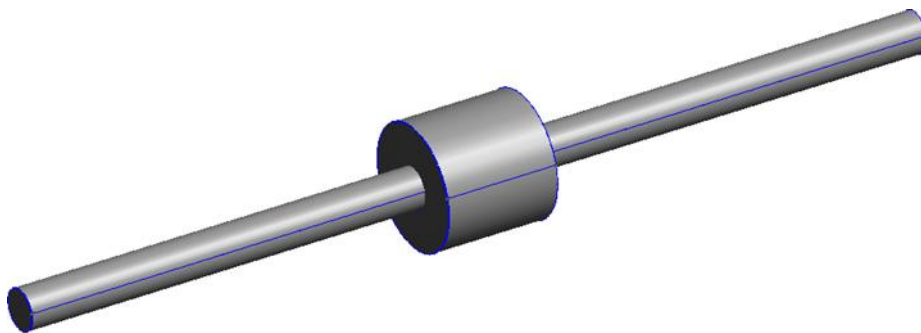


Figura 7-3. Modelo geométrico del biorreactor para la simulación (basado en [26])

Para las simulaciones, se ha creado el volumen que ocupa teóricamente el fluido, definido como el espacio comprendido entre la superficie del scaffold y el de la cámara y los conductos. Para ello se aplicará la resta booleana entre el conjunto de la cámara y los conductos y el dispositivo.

## DISCRETIZADO

Una vez definido el modelo geométrico se procede a realizar la malla del mismo para su posterior cálculo mediante método de elementos finitos. Para ello se ha utilizado el software GMSH [50], que dispone de diferentes algoritmos para realizar mallas de diferentes formas y tamaños. Para el scaffold se ha elegido una malla tetraédrica con un tamaño de elemento uniforme de 0,1 mm (Figura 7-4).



Figura 7-4. Mallado del scaffold

Por su parte para la definición de la malla del volumen del líquido se definirá un tamaño de elemento de 0.5 mm para los contornos exteriores, mientras que para los contornos del scaffold se mantiene el mallado de 0.1 mm (Figura 7-5 y Figura 7-6).

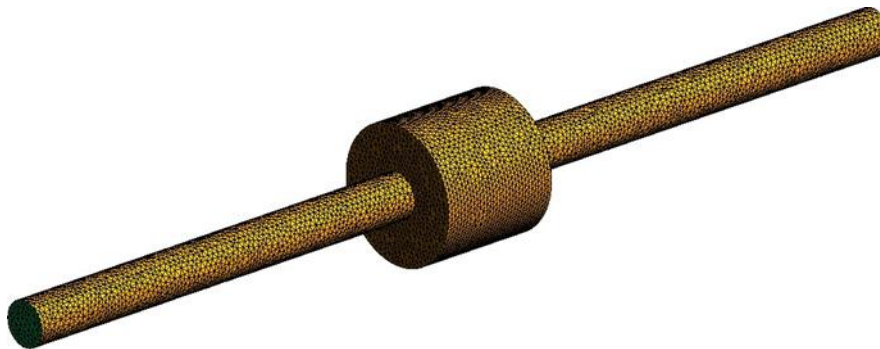


Figura 7-5. Malla representativa del espacio fluido.



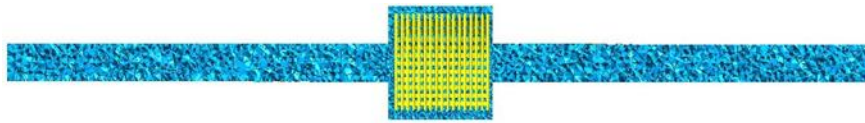


Figura 7-6. Malla representativa del dominio líquido. Sección transversal.

## IMPLEMENTACIÓN NUMÉRICA DE LAS ECUACIONES DE NAVIER STOKES. ESQUEMA NUMÉRICO.

Para la simulación fluidodinámica del medio dinámico se ha recurrido el software FreeFem++, que se diseña para el cálculo de ecuaciones diferenciales en derivadas parciales. Entre los modelos existentes para la simulación por Navier Stokes se ha recurrido a los modelos de Chorin Rannacher y el mdelo de Uzawa con preconditionamiento de Cahout-Chabard. La resolución de los modelos de Navier Stokes ha sido llevado a cabo en FreeFem++ [48], que es un programa principalmente diseñado para la resolución de ecuaciones diferenciales en derivadas parciales, el cual ha sido desarrollado en C++ y es mantenido por investigadores de la Universidad de Pierre- Marie Curie y el laboratorio Jacques-Louis Lions.

Existen varios modelos para representar el flujo en FreeFem++, dentro de los cuales se encuentra el modelo de Navier Stokes- Uzawa [79], [80] y el modelo de Chorin-Rannacher [81].

Dada la ecuación de Navier Stokes en forma vectorial:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nu \Delta \mathbf{u} = \mathbf{f} \quad (7-6)$$

Donde  $p$ =presión/densidad,  $\nu$  es viscosidad cinemática

Asumiendo un flujo incompresible:

$$\nabla \cdot \mathbf{u} = 0 \quad (7-7)$$

El algoritmo de Chorin Rannacher [81] está basado en la metodología propuesta por Chorin [82] para desacoplar las ecuaciones de Navier Stokes, resultando un sistema de ecuaciones

independientes para cada una de las componentes de la velocidad y la presión. Presenta un buen comportamiento numérico, similar al método de Taylor Hood o de penalización, con la ventaja de ser altamente paralelizable, la implementación que consideramos permite obtener buenas exactitudes usando polinomios de grado uno tanto para las velocidades como para las presiones, véase en G.Winter, J.Abdelafatah et al [83], dónde se testea en varios benchmark.

Los dos primeros términos, correspondientes a la derivada total, se aproximan mediante el método de las características, para evaluar la velocidad en el tiempo anterior en la posición de la partícula fluida. Así en lenguaje y soporte FreeFEM++, para el cálculo de las derivadas totales respecto al tiempo se utiliza la instrucción convect, pasando en sucesivas iteraciones de tiempo como argumentos la velocidad en el instante actual, la velocidad anterior, y el paso de tiempo, así sea  $u \circ X(x) = u(x - u(x)\delta t)$ , así la ecuación de Navier Stokes la resolvemos mediante la aproximación de la derivada total (que engloba el término convectivo):

$$\frac{1}{\delta t} [u^m - u^{m-1} \circ X^{m-1}] + \nabla p^{m+1} - \nu \Delta u^m = 0, u|_{\Gamma} = u_T \quad (7-8)$$

Rannacher propuso desacoplar el cálculo de las velocidades y presiones, de modo que en la ecuación de Navier Stokes la presión es calculada previamente con una presión intermedia, según exponemos a continuación. Considera una presión corregida que se relaciona con la velocidad según la siguiente expresión:

$$-\Delta q = \nabla \cdot u^m + \nabla \cdot u^{m+1} \quad (7-9)$$

Formulación variacional en las ecuaciones :

$$\int_{\Omega} \nabla q \nabla w d\Omega - \int_{\Omega} \frac{\nabla \cdot u^m + \nabla \cdot u^{m+1}}{\Delta t} w d\Omega = 0 \quad (7-10)$$

en la que se aproxima el término  $\nabla \cdot u^{m+1}$  por su valor promedio de la iteración de tiempo anterior:

$$\int_{\Omega} \nabla q \nabla w d\Omega - \int_{\Omega} \frac{\nabla \cdot u^m - \frac{(\nabla u)^m}{area}}{\Delta t} w d\Omega = 0 \quad (7-11)$$

Estableciéndose, para el cálculo de la presión [83]:

$$p^{m+1} = p^m - q - \overline{p^m - q} \quad (7-12)$$

La velocidad finalmente es obtenida mediante la expresión que nos es familiar en los métodos de proyección:

$$\mathbf{u}^{m+1} = \mathbf{u}^m + \nabla q \delta t \quad (7-13)$$

Por otra parte, el modelo numérico propuesto por Uzawa [80], presenta una mayor estabilidad en 3 dimensiones y se realiza en ecuaciones en derivadas parciales

-Las ecuaciones de Navier Stokes en la forma de la expresión (7-6) pueden ser reescritas de forma matricial de la siguiente forma:

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix} \quad (7-14)$$

Donde  $\mathbf{u}$  es el vector de velocidad y  $p$  la presión. La expresión (7-14) puede ser reescrita como:

$$\begin{aligned} A\mathbf{u} + B^T p &= b \\ B\mathbf{u} &= c \end{aligned} \quad (7-15)$$

Multiplicando ambos miembros en (7-15) por  $BA^{-1}$  :

$$BA^{-1}A\mathbf{u} + BA^{-1}B^T p = BA^{-1}b$$

$$B\mathbf{u} + BA^{-1}B^T p = BA^{-1}b$$

$$BA^{-1}B^T p = BA^{-1}b - c$$

Nombrando  $BA^{-1}B^T = E$  y  $BA^{-1}b - c = \beta$ , se tiene que:

$$E\mathbf{p} = \beta \quad (7-16)$$

E es simétrica y definida positiva, por lo que es posible aplicar la metodología del Gradiente Conjugado para resolver el problema.

-La presión puede obtenerse fácilmente de la expresión (7-16). Una vez obtenida, la velocidad se puede determinar de la expresión (7-14).

Una forma de acelerar la convergencia es haciendo uso de preconditionadores. Para este algoritmo se recurre al preconditionador Cahouet-Chabart [24].

Una vez calculadas las velocidades, se determinan las tensiones tangenciales, que en el caso de fluidos newtonianos, viene dada por la siguiente expresión:

$$\tau_w = \sqrt{\left[\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right]^2 + \left[\mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\right]^2 + \left[\mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)\right]^2} \quad (7-17)$$

La tensión en los distintos puntos se evalúa con un valor umbral previamente definido [27]. Si en algún punto, se supera el valor obtenido de tensión tangencial, se procede a eliminar el elemento en el que se encuentra.

## 7.2. RESULTADOS

A continuación, se muestran algunos resultados obtenidos. Usando la metodología de Chorin-Rannacher, el mapa de velocidades en los primeros instantes se puede ver en la siguiente figura. Se puede observar que la velocidad es mayor en la entrada a la cámara:

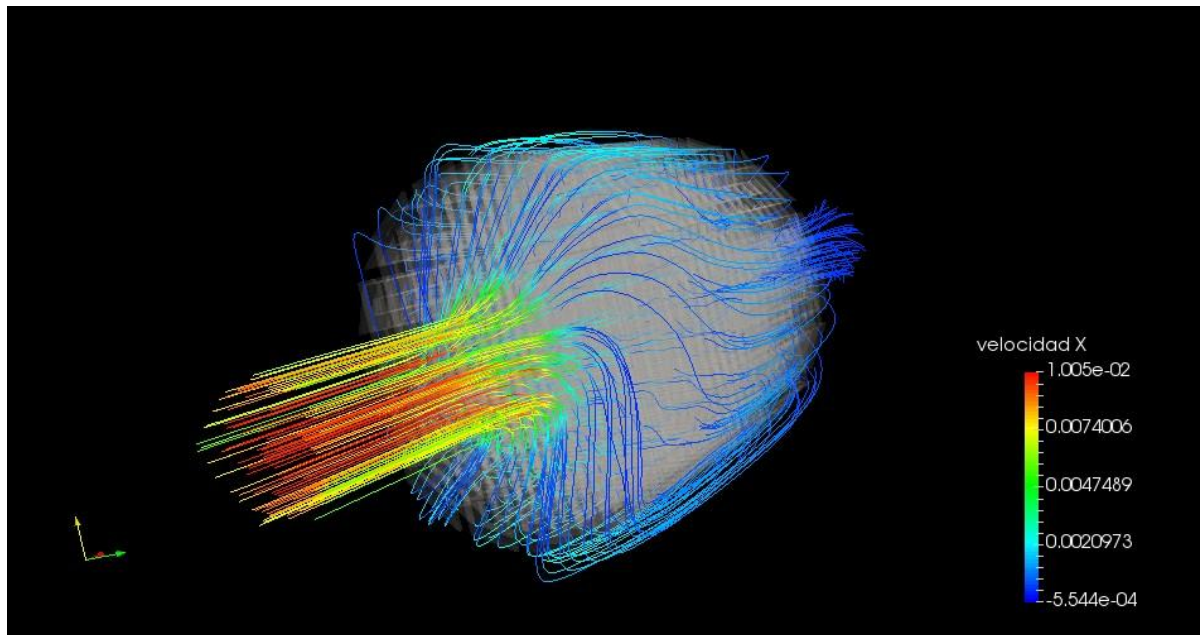


Figura 7-7. Líneas de corriente de velocidad

Esto incide en que las tensiones tangenciales son mayores en las zonas centrales de la cara de la entrada debido a la mayor incidencia del flujo por esas zonas (Figura 7-8). Por lo tanto, en las sucesivas figuras se ve una erosión progresiva cada vez más intensa en esas zonas que se va propagando hacia el resto de la pieza.

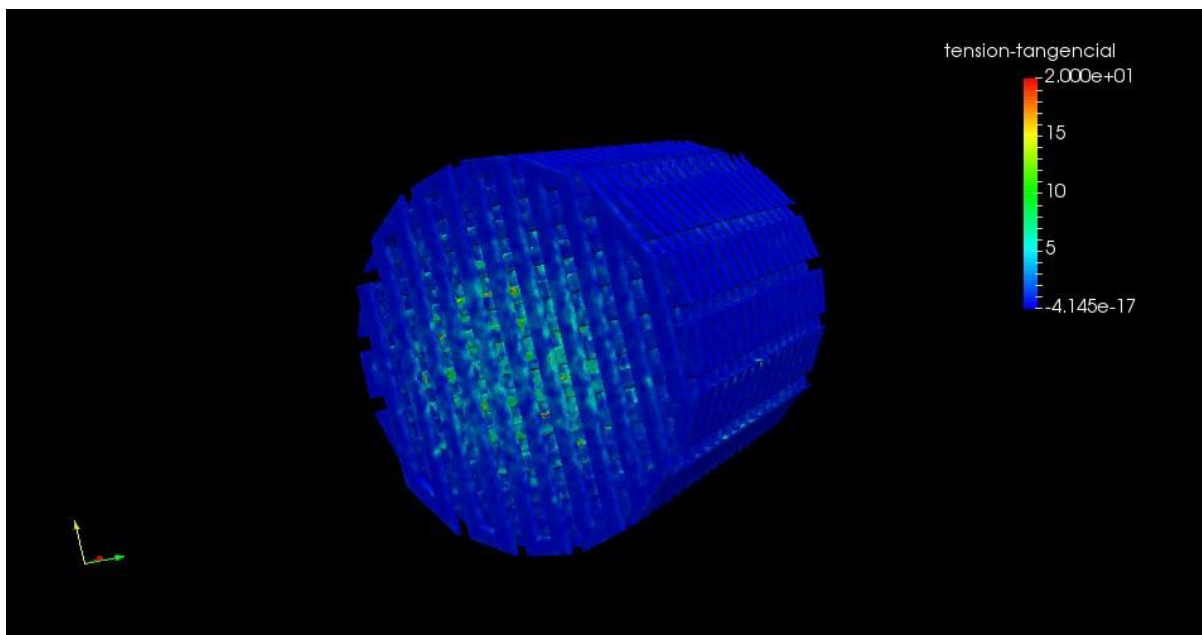


Figura 7-8. Mapa de tensiones tangenciales

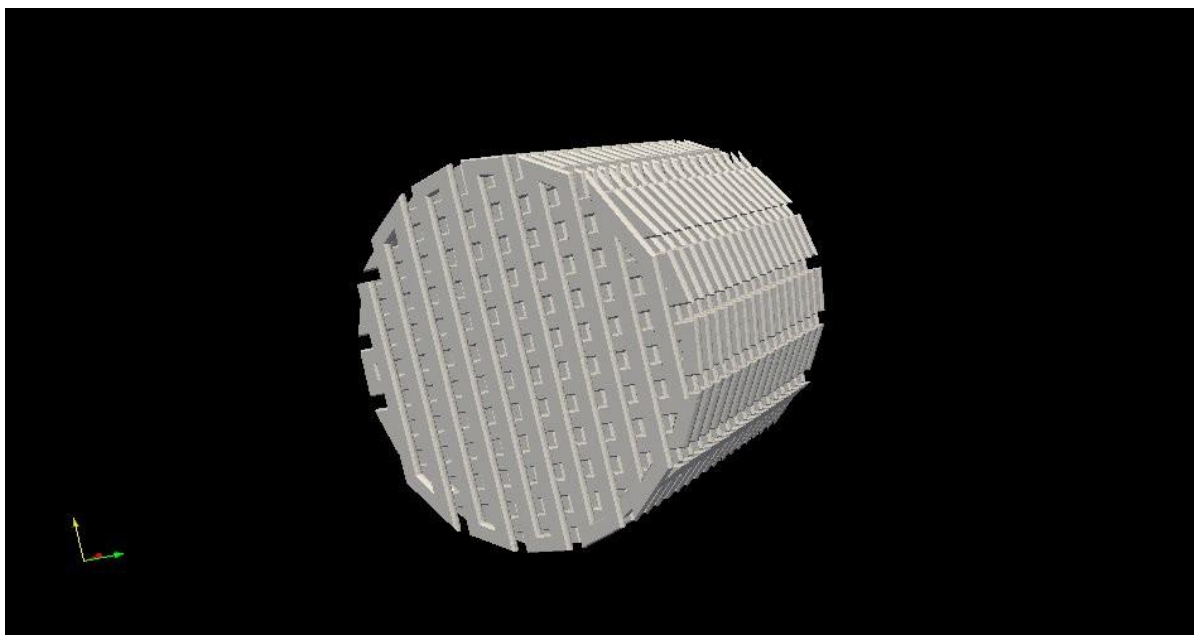


Figura 7-9. Geometría del scaffold en el instante inicial

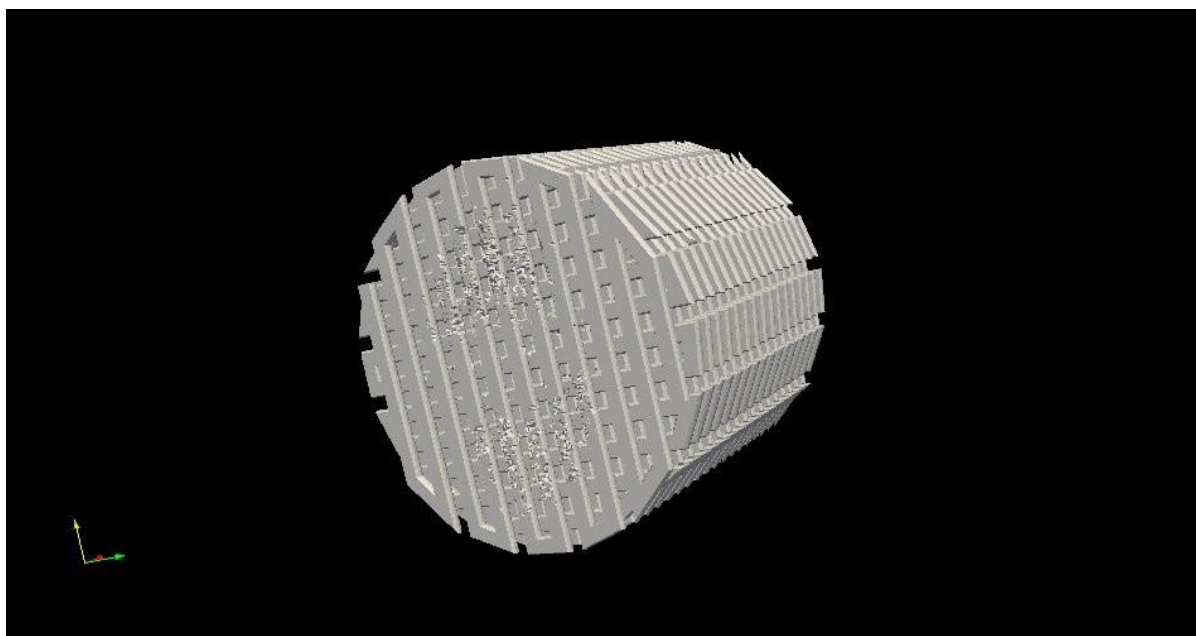


Figura 7-10. Geometría del scaffold en el instante inicial

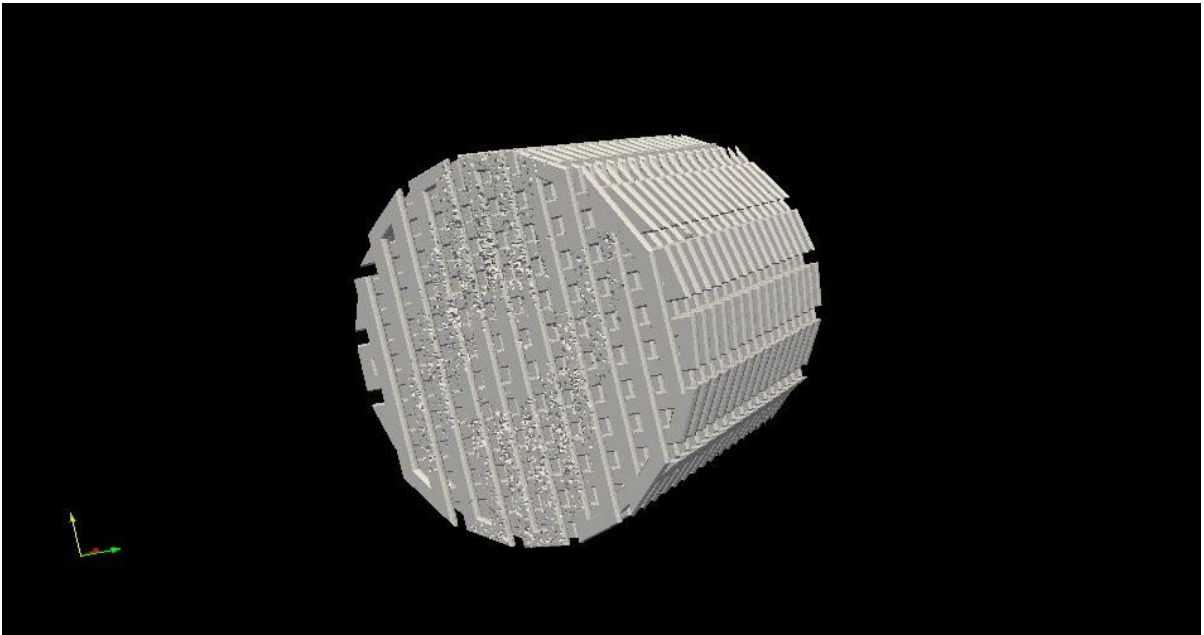


Figura 7-11. Evolución de la degradación en el instante  $t=0.02$

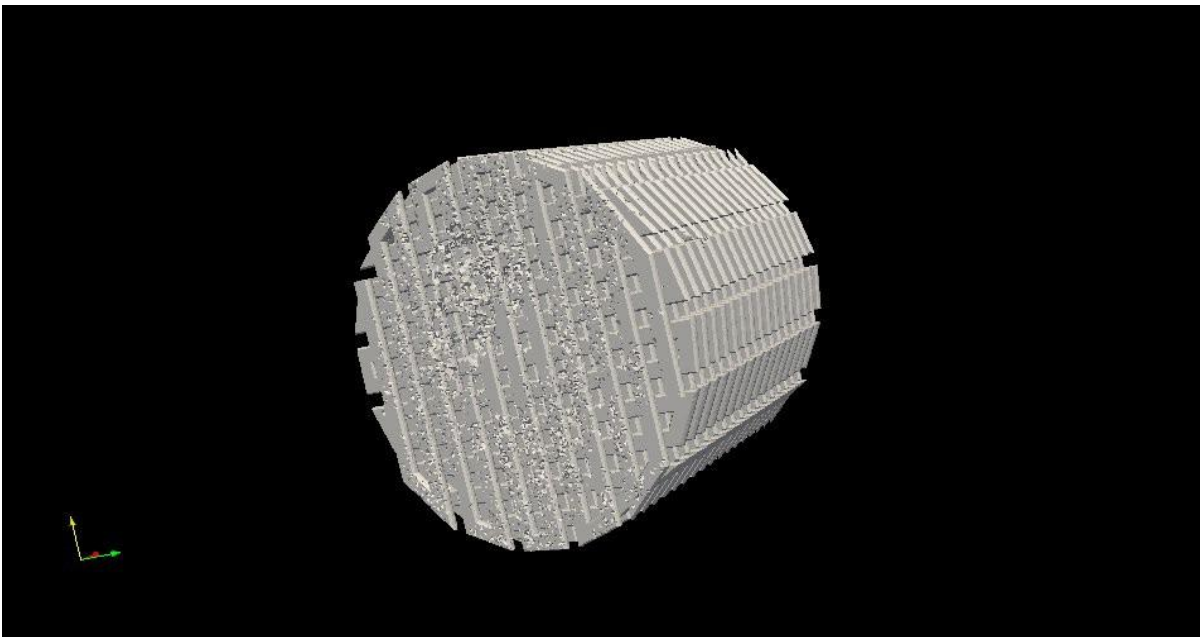


Figura 7-12. Evolución de la degradación en  $t=0.037$



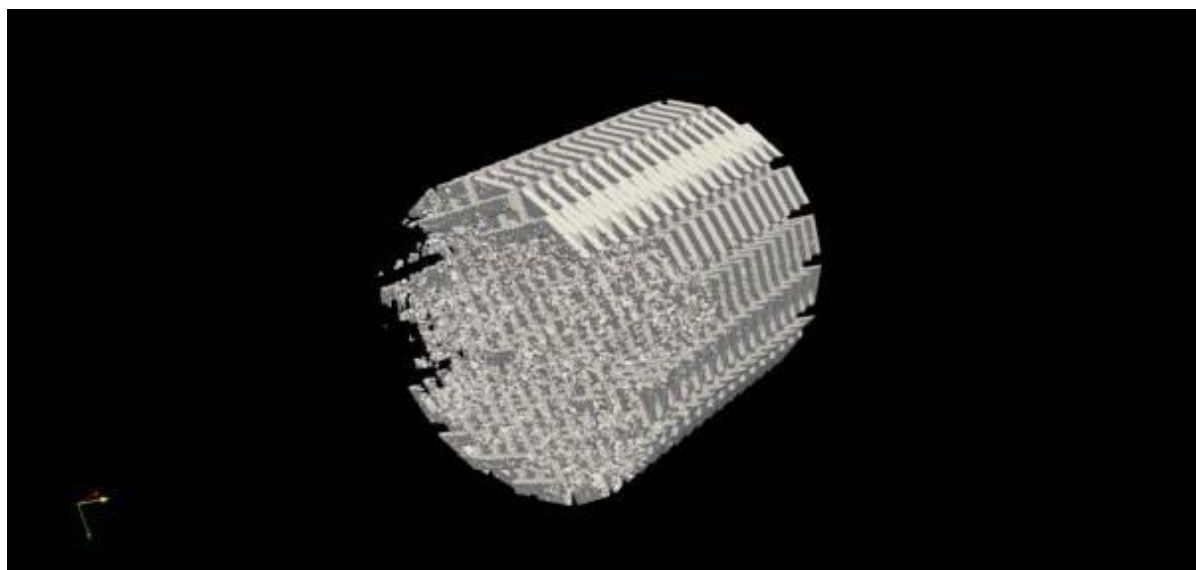


Figura 7-13. Evolución de la degradación en  $t=0.05$

### 7.3. CONCLUSIONES OBTENIDAS DE LA APLICACIÓN DEL MÉTODO.

A nivel computacional, el modelo presenta un buen comportamiento durante las primeras iteraciones. Sin embargo, a partir de cierto punto se observa cierta inestabilidad a medida que la geometría se erosiona, atribuible a la creciente abruptez del contorno resultante de eliminar dichos elementos. Este fenómeno se deriva de la limitación del método de los elementos finitos para calcular las integrales en el contorno después de un número específico de iteraciones. Existe una manera de mitigar este efecto a partir del refinamiento local de las regiones erosionadas.

Por otro lado, las condiciones dinámicas no desempeñan un papel predominante en el proceso de degradación, sino más bien lo contrario. Según Agrawal et al. [6], el movimiento del flujo tiene un efecto de ralentización del proceso hidrolítico, causado por el arrastre de los oligómeros producidos, que contribuyen a una mayor acidez del medio, lo que deriva en un proceso autocatalítico del proceso. En contraste, en condiciones estáticas, la presencia de estos subproductos acelera catalíticamente el proceso debido a un mayor nivel de acidez en el medio.

Además, esta metodología no aborda o considera adecuadamente el fenómeno erosivo experimentado por los biopolímeros en condiciones estáticas. Diversos autores consideraron el fenómeno erosivo desde la perspectiva de la exposición al medio o de la propensión a la erosión de los elementos debida a la existencia de regiones cristalinas y amorfas. Para la modelización del comportamiento del material bajo estas condiciones, se requiere un enfoque alternativo, que no requiera del cálculo del flujo sino que cuantifique la pérdida de material a través de parámetros estocásticos implique la degradación del polímero bajo condiciones estáticas [16]. La presencia de un medio acuoso afecta la respuesta del material de manera considerable. La degradación enzimática de la malla se debe a la presencia de mallas. En otras palabras, esto afecta a la degradación del material bajo condiciones dinámicas.

Un caso particular del fenómeno degradativo-erosivo bajo condiciones estáticas es cuando por procesos enzimáticos [19]. Muchos de los polímeros sintéticos tales como el PLA y el PCL son susceptibles a procesos de degradación enzimática [84]. La presencia de material enzimático ayuda al proceso de erosión de la pieza. Debido a la voluminosidad de las moléculas enzimáticas el proceso se asume también como erosivo. La acción de las enzimas facilita la ruptura del material u contribuye a su progresiva disolución.

En el siguiente capítulo se abordará una metodología estocástica basada en autómatas celulares que ayude a predecir el comportamiento erosivo del material bajo dichas condiciones anteriormente mencionadas.

## 8. ANÁLISIS ESTOCÁSTICO DEL COMPORTAMIENTO EROSIVO SUPERFICIAL DE LOS BIOPOLÍMEROS BAJO CONDICIONES ESTÁTICAS. ESTUDIO DE LA DEGRADACIÓN ENZIMÁTICA DE FILAMENTOS Y SCAFFOLDS.

A lo largo de los años ha habido un interés creciente en la modelización de la erosión superficial de los materiales poliméricos bajo diferentes condiciones tanto estáticas como dinámicas. Para caracterizar este comportamiento se recurren a modelos mecanicistas, de entre los cuales se encuentran métodos basados probabilísticos basados en Monte Carlo y métodos deterministas. De entre los modelos existentes destaca el propuesto por Göpferich et al [21], [42], en el que divide la matriz polimérica en una región rectangular bidimensional de  $n_x \times n_y$  píxeles, cada uno de ellos con un estado (cristalino, amorfo y erosionado) sometida a unos cambios de su morfología. La erosión superficial va asociado a un tiempo de vida. Dependiendo de si la región del material es cristalina o amorfa, el material tendrá una mayor o menor probabilidad de disolución.

A cada elemento se le asume una probabilidad de disolución a través de un parámetro definido como tiempo de vida  $t_{vida}$ . Göpferich, en primer lugar y Siepmann, más adelante, desarrollaron unos modelos estocásticos en los que el material se subdivide en regiones amorfas y cristalinas. A los elementos cristalinos se les asigna una menor probabilidad de disolución, es decir, que se les asigna un mayor tiempo de vida. La función de densidad empleada para modelar este tipo de fenómenos temporales en términos estocásticos se define según una función de Poisson de primer orden.

$$P_{dissolution} = \lambda e^{-\lambda t}; \quad t \geq 0 \quad (8-1)$$

Donde  $\lambda$  es el ritmo medio de que se materialice el proceso de disolución. Para una matriz polimérica de dimensiones  $n_x \times n_y$  el valor esperado de tiempo de disolución es [21]:

$$e_{n_y}(t) = \lambda n_y e^{-t \lambda n_y} \quad (8-2)$$

Por otra parte, este modelo fue simulado mediante el uso de autómatas celulares. Este método se basa en estimar la influencia de los vecinos líquidos que rodean un elemento central. Cuantos más elementos haya, mayor probabilidad de disolución tendrá el elemento en cuestión.

La matriz está definida en una matriz virtual en un espacio bidimensional o cúbico, con un número de estados siendo modelados. El tiempo de vida de un polímero cambia con el número de celdas vecinas y el número de estados que han sido modelados. Esta metodología se ha encontrado como una forma razonable de ajustar los valores experimentales excepto con la degradación está asociada a la erosión superficial.

## **8.1. METODOLOGÍA DE MONTE CARLO ASOCIADA AL CÁLCULO DE AUTÓMATAS CELULARES.**

En este apartado se presenta una metodología basada en los autómatas celulares. Se tratará de evaluar el grado de exposición al líquido de los diferentes elementos bajo condiciones de ensayo. Diversos autores han desarrollado modelos que permiten predecir la erosión frontal usando esta metodología. Por ejemplo, Zigourakis et al [20] fueron uno de los primeros autores que adoptaron esta metodología en el proceso de erosión frontal para la liberación de fármacos. Por otra parte, Erkizia et al [7] definieron un modelo en el que la metodología está basada en el grado de curvatura y la distancia de los centros a los vecinos líquidos.

Los elementos totalmente expuestos al medio se disolverán automáticamente. Por el contrario, aquellos elementos que se encuentren totalmente rodeados de elementos sólidos permanecerán en la pieza.

La evaluación del número de caras está basada en la metodología de Von Neumann [27, 28]. Esta metodología contabiliza, para cada elemento, el número de elementos adyacentes como aquellos que comparten al menos un lado o cara con el elemento central. En 2D la representación del modelo viene dado por la Figura 8-1.

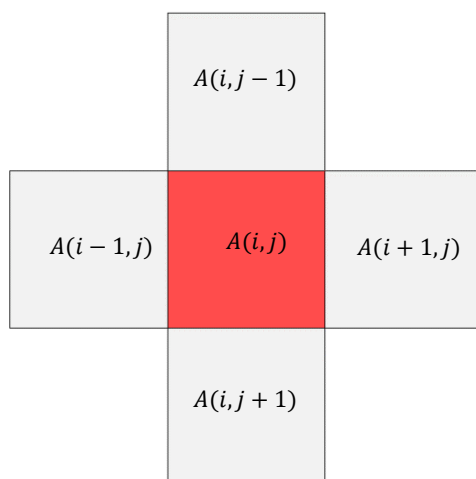


Figura 8-1. Representación en 2D de la vecindad de Von Neumann usando vóxeles

Como se puede ver, para un elemento central cuadrado, existen 4 vecinos directos posibles. Si extrapolamos esto al caso de un elemento central cúbico, tendríamos un máximo de seis vecinos directos (Figura 8-2):

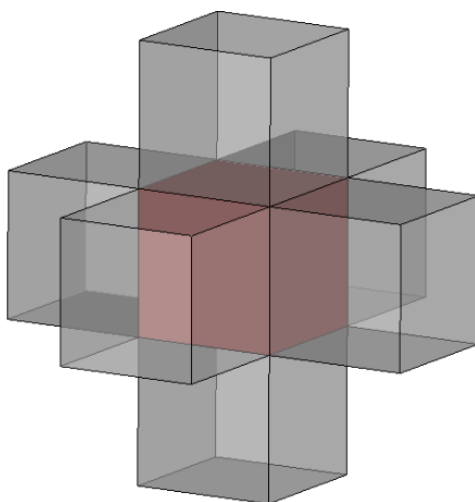


Figura 8-2. Representación en 3D de la vecindad de Von Neumann usando vóxeles.

Esta metodología presentan ciertos inconvenientes ya que el uso de vóxeles hexaédricos presentan algunos problemas a la hora de discretizar geometrías más complejas, ya que exige un refinado mayor de la pieza , o bien, pueden aparecer elementos no-cúbicos [87].

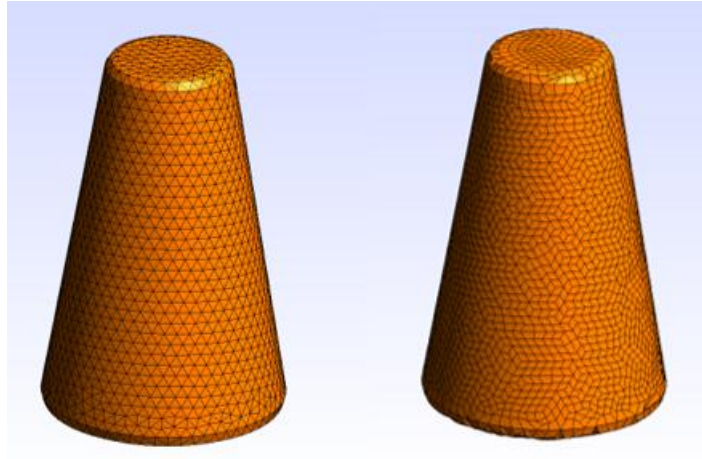


Figura 8-3. Ejemplo de discretizado de una pieza cónica usando elementos tetraédricos(izquierda) y elementos hexaédricos(derecha) realizado en GMSH, basado en [87]

Recientemente, Deng et al [87] han propuesto una versión del modelo de Von Neumann en la que las vecindades entre elementos se establece en base a elementos tetraédricos, por lo tanto se establece una vecindad entre elementos directos que puede llegar a 3, en el caso 2D y 4 en el caso en 3 dimensiones(Figura 8-4 y Figura 8-5):

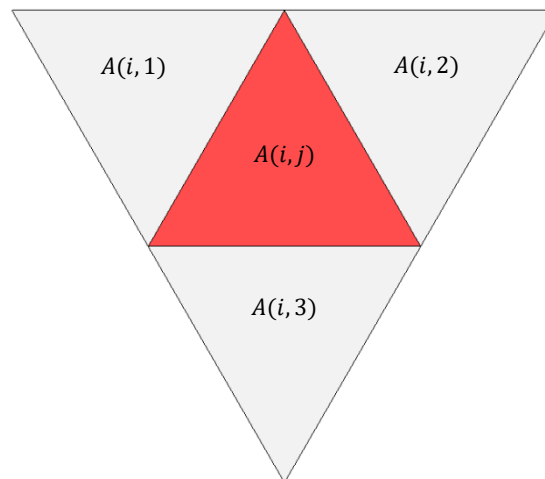


Figura 8-4. Representación de la vecindad de Von Neumann para elementos tetraédricos.

En el caso 2D existe un número de vecinos directos igual a 3, 1 menos que para el caso elementos cuadriláteros. Esto, a efectos computacionales, presenta una mayor eficiencia, aunque esto podría resultar en una menor precisión de cálculo.

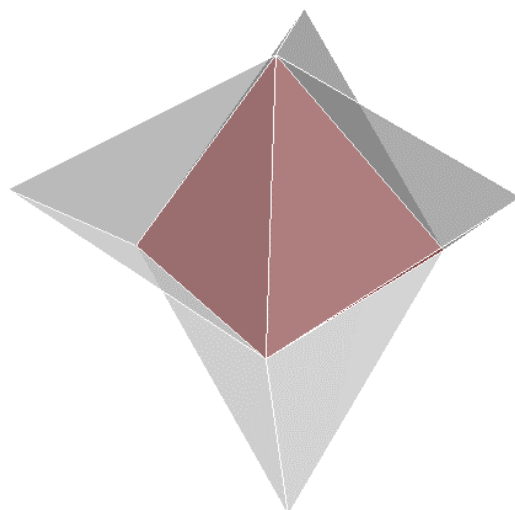


Figura 8-5. Representación de la vecindad de Von Neumann para tetraedros regulares en 3D.

A pesar de la periodicidad que generalmente presentan los scaffolds, se tratan de geometrías con un grado de curvatura y complejidad elevado, por lo que el uso de elementos hexaédricos o vóxeles requiere de un refinamiento lo suficientemente alto, con el consiguiente coste computacional. Por este motivo, a veces se opta por realizar un análisis local sobre una celda unitaria, y extrapolar los resultados a una geometría más compleja. Por este motivo el tamaño de los diferentes elementos será desigual, por lo que se requiere de incorporar un factor de peso del tipo:



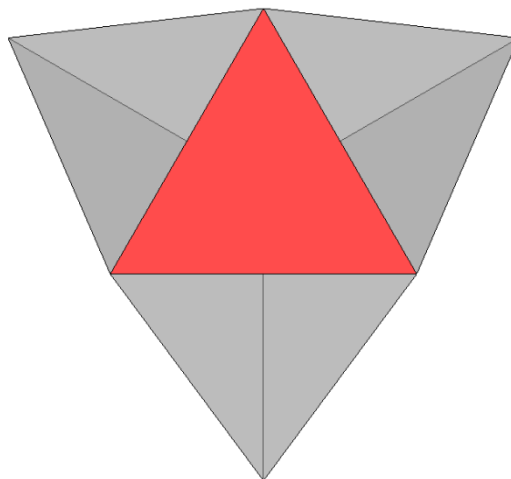


Figura 8-6. Representación ortogonal de la vecindad de Von Neumann en 3d

Como se puede ver existen cuatro celdas contiguas al tetraedro central. Las cuatro celdas comparten una cara con el elemento central

## 8.2. SIMULACIÓN DE MONTE CARLO DE LA DEGRADACIÓN DE UN FILAMENTO DE COMPOSITE POLICAPROLACTONA-MICROCELULOSA (PCL-MCC).

### 8.2.1. ANTECEDENTES

Las enzimas son proteínas que tienen una compleja estructura tridimensional, y su actividad está íntimamente relacionada con la estructura específica que conforma. Aquellas regiones relacionadas con la actividad enzimática son aquellas que se relacionan con las secuencias características primarias. Estas regiones son aquellas en las que se produce la interacción con el sustrato.

En cuanto a los mecanismos que se producen, las enzimas se clasifican en diferentes tipos: las endoenzimas, provocan la escisión aleatoria de los enlaces principales de la cadena, lo que lleva a una aceleración del proceso de degradación; las exoenzimas, por su parte, atacan secuencialmente los enlaces terminales de las cadenas. La absorción y la velocidad de reacción dependen de las características intrínsecas de la enzima, así como de factores externos, tales como el pH y la temperatura. Por lo general, al ser la estructura de las enzimas muy voluminosa, le es muy complicado penetrar dentro de la matriz polimérica, por lo que la degradación enzimática es un proceso eminentemente erosivo. La erosión depende de varios factores, entre los que se encuentran la degradación polimérica, el hinchamiento del material, la cristalinidad, la hidrofobicidad, la longitud de cadena y la difusividad del medio.

Entre las enzimas capaces de degradar el PLA están la lipasa, esterasa y Alcalase. Para la policaprolactona, la lipasa ha conseguido catalizar la hidrólisis del material, por lo que es posible observar cambios significativos en la degradación del material en periodos cortos de tiempo.

Se han planteado diferentes estrategias para poder modificar la velocidad de degradación de un polímero, sin modificar sus ventajas como termoplásticos bioabsorbibles, tales como el desarrollo de mezclas, el uso de materiales compuestos ("composites" en inglés) y copolímeros. Los composites se forman por medio de la introducción de un material de relleno en la matriz polimérica, modificando las propiedades del material base, entre las que se encuentra la cristalinidad, la cual tiene una gran influencia en la tasa de degradación de la pieza. Además, la presencia del relleno hace que se modifiquen otras propiedades como su aspereza, [88], y su mojabilidad, lo cual puede modificar la tasa de adsorción del material, lo que constituye un paso determinante en el mecanismo de degradación enzimática de los polímeros.

El uso de los materiales de relleno se ha analizado principalmente en términos de mejora de la bioactividad de los matrices derivadas del PCL. Sin embargo, apenas aparecen estudios que empleen esta técnica para el ajuste de la cinética de degradación de estos procesos [89]. En estudios anteriores, se ha demostrado que el uso de celulosa microcristalina (MCC), junto al PCL tiene una experiencia probada en el campo de la ingeniería de tejidos óseos [90]

### 8.2.2. METODOLOGÍA

El esquema resumido de esta metodología se muestra en el siguiente diagrama de flujo. La metodología fue implementada en FreeFem++ [48]. El primer paso consiste en pasar la geometría en formato malla al programa. La geometría es discretizada, usando el software Gmsh [50] y cargada dentro del programa. Adicionalmente, la pérdida experimental de masa (expresadas en tanto por ciento para cada etapa de degradación) se introduce también como parámetro de entrada. Inicialmente, por simplicidad la metodología asume una densidad constante, por lo que el porcentaje de volumen de polímero disuelto coincide con el de masa.

$$\%m_{disuelto} = \%V_{disuelto} \quad (8-3)$$

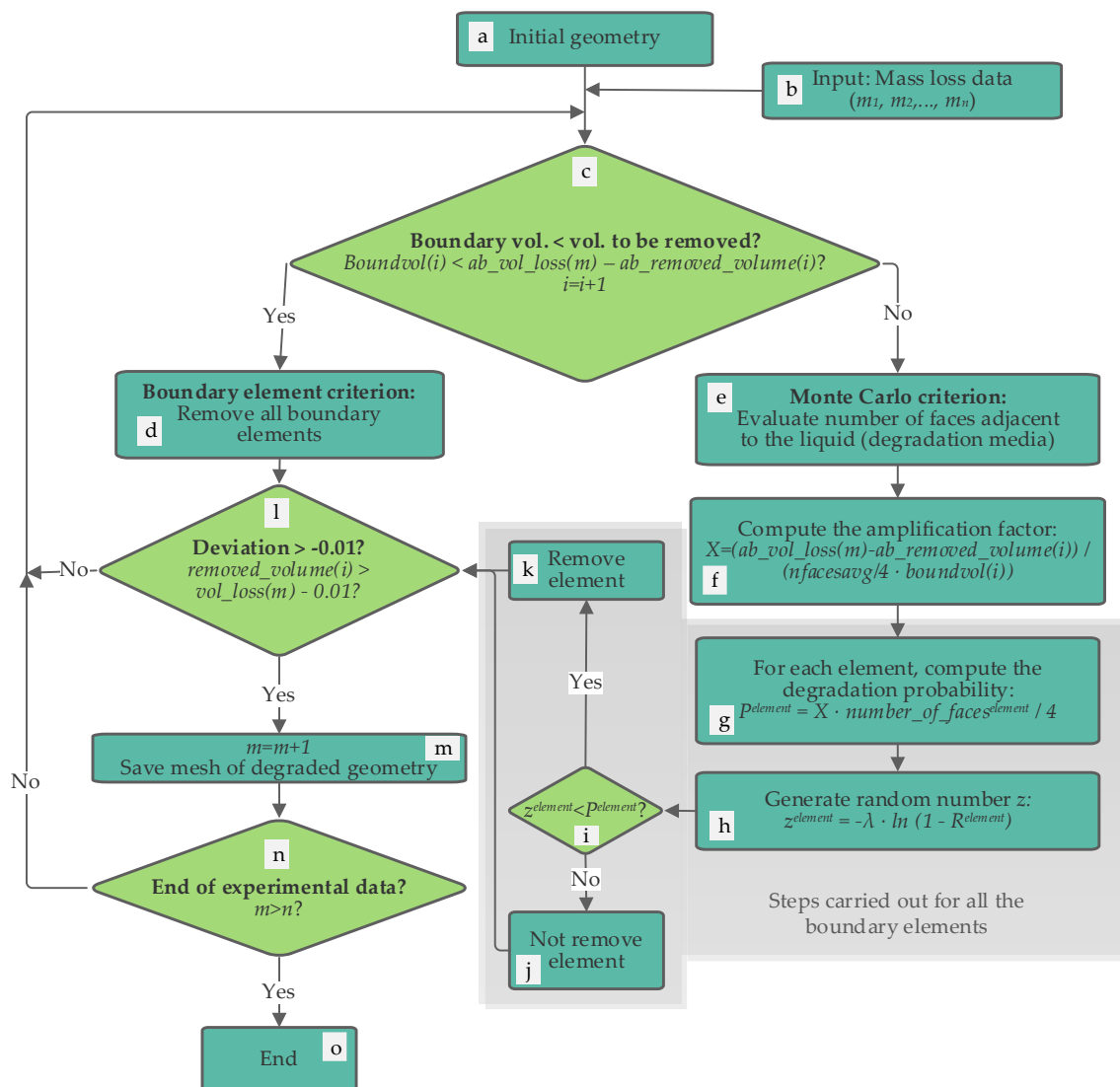


Figura 8-7. Esquema del modelo de cálculo estocástico de la degradación enzimática.

A continuas sería hacer un barrido e identificar aquellos elementos superficiales que se encuentran en la capa más externa asignándoles una variable de estado B que toma valores 0 o 1 dependiendo de si el elemento linda con el medio o no.

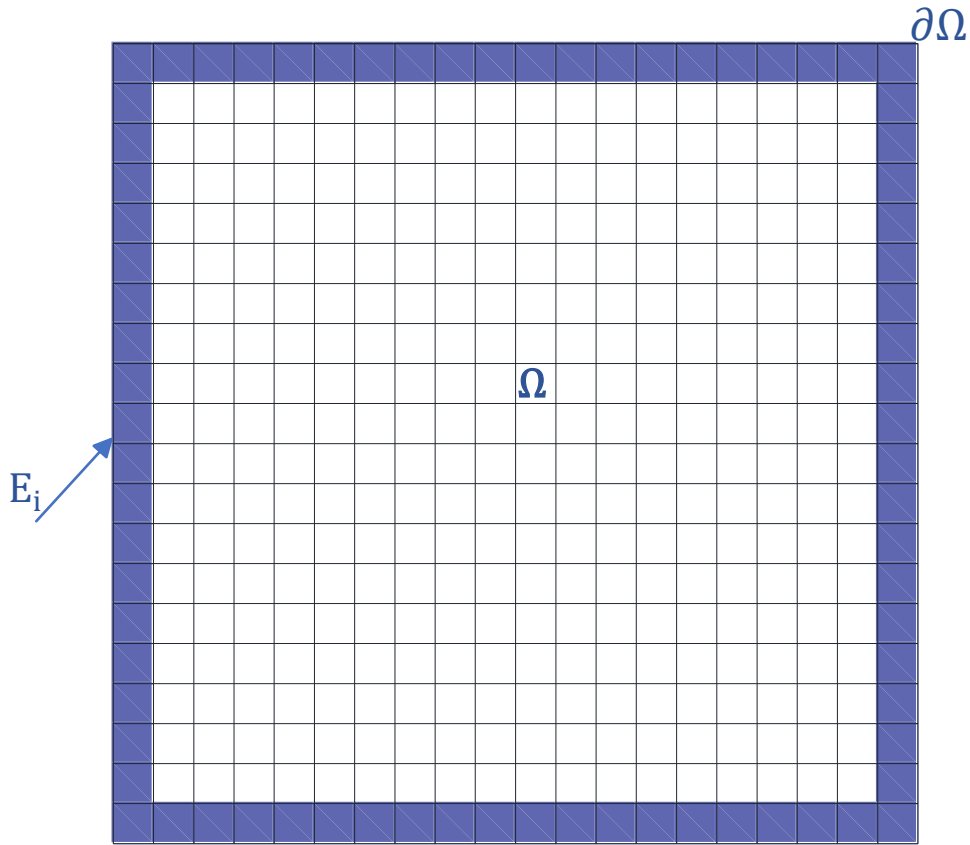


Figura 8-8. Selección de los elementos superficiales de la geometría de la malla.

$$\begin{cases} B_i = 0 & \text{si } E_i \cap \partial\Omega = 0 \\ B_i = 1 & \text{si } E_i \cap \partial\Omega \neq 0 \end{cases} \quad (8-4)$$

Una vez asignados los valores correspondientes a cada uno de los elementos del contorno, el algoritmo calcula, sobre la geometría discretizada, el volumen resultante de los elementos superficiales y lo compara con el volumen de necesario para alcanzar el valor de referencia. en la etapa de estudio (Figura 8-7c). Para llevarlo a cabo, el método compara el volumen de los elementos de contorno (boundvol(i)(mm<sup>3</sup>)), con la diferencia entre el volumen ya extraído (ab\_removed\_volume(i), in mm<sup>3</sup>) y el volumen a extraer en la etapa i de degradación (ab\_vol\_loss(m)), que, al principio, se corresponde con el primer dato de pérdida de masa en valor absoluto.

$$\text{Boundvol}(i) > \text{ab\_vol\_loss}(m) - \text{ab\_removed\_volume}(i) \quad (8-5)$$

Siendo “i” el Contador de iteraciones y “m” el identificador del volumen de referencia, que se corresponde con los diferentes tiempos de medición.

Si no se alcanza la condición anterior, se eliminan todos los elementos de contorno, repitiendo el proceso tantas veces como sea necesario hasta que la diferencia sea lo suficientemente pequeña como para aplicar el método de Monte Carlo, en el que los elementos están expuestos a la influencia del medio circundante. En ese caso se aplica el método de Monte Carlo para alcanzar el volumen de referencia (Figura 8-7d, criterio de elementos de contorno). Si, por el contrario, se cumple la condición previa, entonces se aplica el criterio de Monte Carlo (Figura 8-7e). Para el caso en que se aplica la simulación de Monte Carlo, la metodología propuesta se basa en parte en la desarrollada por Erkizia [7]. En dicho trabajo, se considera la influencia de las caras y aristas de los elementos líquidos adyacentes para determinar su probabilidad. En la metodología que se presenta aquí, se incorpora un término adicional, denominado “factor de amplificación”, para ajustar la probabilidad de degradación a los valores experimentales de pérdida de masa, lo que consigue una aceleración de la convergencia del proceso a los valores experimentales. En esta metodología se omite la influencia de las aristas en la probabilidad del método. Por tanto, el primer paso es evaluar el número de caras adyacentes al medio circundante. Esto consiste en evaluar, para cada elemento superficial (en este caso tetraédrico), el número de caras en contacto con el medio. Los elementos con al menos una cara en contacto con el medio se consideran como elementos de contorno (Figura 8-7e). En este paso, se etiquetan todos los elementos de contorno a través de una variable lógica, y también se calcula el valor promedio de caras en contacto con el líquido ( $n_{facesavg}$ ).

El siguiente paso consiste en calcular el “factor de amplificación”,  $X$  (Figura 8-7f). Este factor tiene en cuenta el valor actual de referencia de pérdida de masa ( $ab_{vol_{loss}(m)}$ , en  $mm^3$ ), el volumen ya extraído ( $ab_{removed\_volume(i)}$ , en  $mm^3$ ), el número promedio de caras en contacto con el líquido ( $n_{facesavg}$ ):

$$X = \frac{ab_{vol_{loss}(m)} - ab_{removed\_volume(i)}}{(n_{facesavg}/4 \cdot bound_{vol}(i))} \quad (8-6)$$

El factor de amplificación representa la proporción de volumen de elementos que deben ser extraídos para alcanzar la pérdida de masa de referencia, pero dividido entre el número promedio de caras en contacto con el líquido. Esto significa que, para un valor alto del factor de amplificación, se necesita extraer una mayor cantidad de masa para alcanzar el valor deseado.

Después de determinar el valor de  $X$ , el algoritmo evalúa, para cada elemento de contorno, la probabilidad de degradación (Figura 8-7g) como:

$$P_{element} = X \frac{n_{faces_{element}}}{4} \quad (8-7)$$

$number\_of\_faces_{element}$  es el número de caras del elemento en contacto con el líquido o el medio de degradación. O lo que es lo mismo, la probabilidad de degradación de cada

elemento depende de la relación entre el número de caras en contacto con el medio y el número de caras totales, multiplicado por el factor de amplificación.

Para decidir si el elemento evaluado, se disuelve o no, se compara en base a un número aleatorio  $Z_{element}$ , que se genera de acuerdo a una distribución exponencial (Figura 8-7h):

$$Z_{element} = -\lambda \cdot \ln(1 - R_{element}) \quad (8-8)$$

Siendo  $\lambda$  un parámetro de ajuste para ralentizar o acelerar el proceso, y  $R_{element}$  es un número aleatorio generado en base a una distribución uniforme.

Por consiguiente, para cada elemento de contorno, se calcula la probabilidad de degradación y se genera un número aleatorio  $Z_{element}$ . Así, los dos valores son comparados (Figura 8-7i). Si este último se queda por debajo del valor de probabilidad, se procede a la disolución del elemento correspondiente. Por el contrario, el elemento permanecerá en la pieza.

El siguiente paso, que es común para los dos criterios (Figura 8-7.j), evalúa la desviación entre el valor simulado de pérdida de masa  $z$  el valor de referencia. Si la diferencia es menor que un -0.01%, entonces se considera que se ha alcanzado dicho valor.

$$removed\_volume(i) > vol\_loss(m) - 0.01 \quad (8-9)$$

La metodología se aplica sobre dos filamentos de materiales distintos, el PCL puro y el composite PCL: MCC 98:2. Para ambos casos, el modelo geométrico de partida es el mismo, correspondiente a las dimensiones reales de filamentos obtenidos en los ensayos experimentales (diámetro de 1,9 mm y longitud de 50,3 mm). Una vez definida la geometría, se procede al discretizado de la misma. Para ello, se importa el modelo en GMSH [50], para la obtención de una malla regular. La geometría fue discretizada usando elementos tetraédricos, obteniendo una malla estructurada, representada en la Figura 8-9:

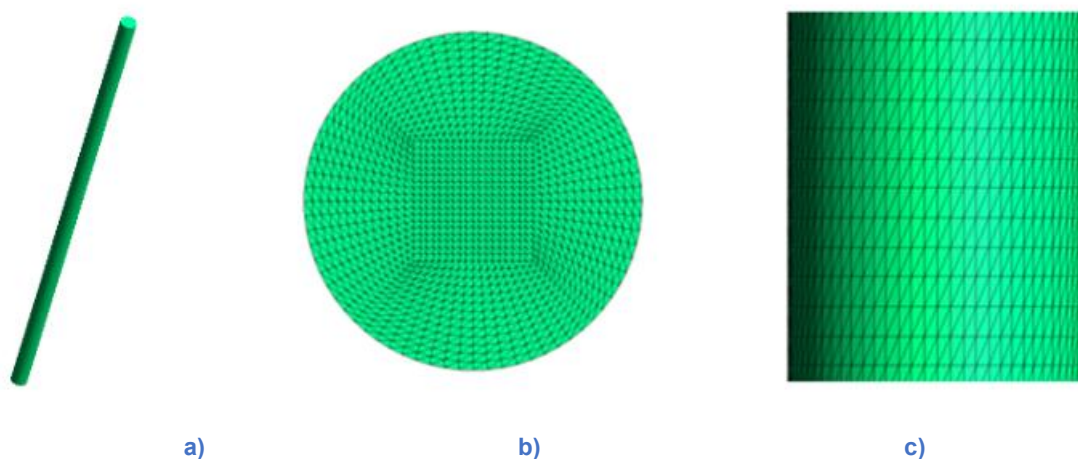


Figura 8-9. Diseño del filamento (de diámetro 1,9 mm y de longitud 50,3 mm) y vistas parciales de la geometría mallada (con 4258800 elementos y 730431 nodos). a) Geometría inicial del filamento; b) Sección transversal de la geometría discretizada; c) Vista longitudinal de la geometría mallada.

La malla fue importada en FreeFem++ [48]. Los datos de pérdida de masa (expresados en porcentaje) fueron añadidos como valores de entrada de acuerdo con la Tabla 8-1. Tanto la degradación de los filamentos de PCL puro como del composite se utilizó un cluster ubicado en el Centro de Procesamiento de Datos del SIANI, el cual cuenta con 28 nodos de cálculo y un nodo adicional para el acceso al mismo. Los nodos de cálculo están agrupados en siete equipos Bullx R424E2, cada uno con 4 nodos. Cada nodo de cálculo consiste en:

- Un total de 2 procesadores Intel Xeon E5645 Westmere EP, con seis núcleos cada uno.
- Un total de 48 GB of RAM.
- Un disco duro de 500 GB
- Interfaz infiniband a 40 Gbs.

Material	Tiempo de degradación (h)	Pérdida de masa (%)	Módulo de elasticidad (MPa)	pH
PCL	0	-	306±46	7,59
PCL	4	0,84±0,11	304±29	7,00
PCL	24	6,16±0,28	265±37	6,28
PCL	48	5,88±0,28	270±25	5,59
PCL:MCC 98:2	0	-	295±17	7,59
PCL:MCC 98:2	4	1,12±0,15	288±36	6,74
PCL:MCC 98:2	24	13,68±1,31	239±16	4,70
PCL:MCC 98:2	48	16,15±5,28	216±23	4,35

Tabla 8-1. Resultados de pérdida de masa y módulo de elasticidad debido a la degradación enzimática.

### 8.2.3. RESULTADOS

En la Tabla 8-2 se muestran los resultados de degradación simulados para un filamento de PCL:MCC 98:2. Cada fila de la tabla los diferentes datos obtenidos a lo largo de las sucesivas iteraciones de Monte Carlo. La primera columna representa el contador de iteraciones; la segunda representa el volumen de elementos superficiales que se encuentran al inicio de la iteración correspondiente; en la tercera columna se registran los valores porcentuales de pérdida de volumen de acuerdo con los valores de pérdida de masa. Nótese que aquí se ha



asumido una densidad constante, por lo que los valores porcentuales de pérdida de masa y de volumen son idénticos. En el ensayo; la cuarta columna describe la metodología utilizada para extraer los elementos en función de la relación entre el diferencial para llegar al volumen de referencia y el volumen de elementos superficiales. Si esta relación es mayor que 1, es decir, que el volumen necesario para alcanzar el valor de referencia es mayor que el volumen de los elementos superficiales, entonces se procede a la extracción de los últimos, en caso contrario se aplicaría la metodología de Monte Carlo para cada elemento por separado. En la quinta columna se representa el volumen extraído acumulado, en mm<sup>3</sup>, al final de la iteración. La sexta columna es igual que la anterior, solo que representándolo en términos porcentuales. Por lo tanto, tanto la tercera como la sexta columna deben coincidir para que el modelo tenga un buen ajuste. En relación con este aspecto, la última columna representa la desviación entre los valores calculados en la simulación y los valores de referencia del ensayo realizado.

Iteración	Volumen de elementos de contorno (mm <sup>3</sup> )	Volumen de referencia extraído (%)	Criterio Aplicado	Volumen extraído (mm <sup>3</sup> )	Volumen extraído (%)	Desviación (%)
1	4,62	0,84	Monte Carlo	1,21	0,85	0,01
2	5,45	6,16	Elementos de contorno	6,66	4,67	-1,49
3	5,10	6,16	Monte Carlo	8,88	6,17	0.01

Tabla 8-2. Resultados simulados de degradación de un filamento de PCL

En la primera iteración, se aplica el criterio de Monte Carlo, y se observa que el volumen extraído coincide prácticamente con el correspondiente de referencia, siendo la desviación de un 0,01% respecto al valor experimental. Por consiguiente, el valor de referencia se actualiza, tomando siguiente dato de medición experimental, que en este caso es 5,45%. En este caso, la diferencia entre los valores experimentales y los simulados son lo suficientemente grandes, el algoritmo aplica el criterio de elementos de contorno a la extracción de la totalidad de los elementos superficiales presentes en el modelo. Se observa un desfase de un -1,49% respecto al valor de referencia. Así, en la tercera iteración, el ratio entre el volumen a extraer y el volumen superficial vuelve a ser menor que uno y, por ende, el algoritmo aplica la metodología de Monte Carlo, con un factor de amplificación ajustado, logrando así un buen

ajuste entre el valor simulado (6,17%) y el valor experimental (6,16%). Por lo tanto, vemos que la metodología converge con rapidez a los valores experimentales de referencia de pérdida de masa. Nótese que el último valor de referencia de pérdida de masa para PCL (Tabla 8-2) no se incluyó como valor de entrada en el método numérico ya que se produjo la saturación enzimática.

Como la iteración 1 alcanzó el primer valor de referencia de pérdida de volumen, la malla resultante de esta iteración corresponde a la geometría degradada a las 4 h. Asimismo, la malla resultante de la iteración 3 corresponde a la geometría degradada del segundo valor de referencia, a las 24 h.

Por otro lado, la Tabla 8-3 muestra los resultados correspondientes al filamento compuesto (PCL:MCC 98:2). El tiempo de cálculo fue de 3549 s (aproximadamente 59 min), claramente superior al tiempo necesario para la PCL pura (25 min). Esta diferencia en el tiempo de cálculo se debe principalmente a la mayor tasa de degradación y al mayor tiempo de degradación del filamento compuesto (en PCL, el último tiempo de degradación fue de 24 h en lugar de 48 h, ya que la última pérdida de masa experimental no se consideró para la simulación). En este caso, el primer valor de referencia de volumen removido (1,12%) se alcanzó en la primera iteración, con el método de Monte Carlo. Sin embargo, la segunda referencia se logró después de cuatro iteraciones adicionales (tres con la eliminación de todos los elementos de contorno correspondientes y la última aplicando el método de Monte Carlo). Esto se debe a la alta diferencia entre el primer valor de referencia de pérdida de masa (1,12%) y el segundo (13,68%). De hecho, la eliminación completa de los elementos de contorno es útil para casos como este, en los que existen grandes diferencias entre valores de referencia consecutivos de pérdida de masa (la eliminación completa de los elementos de contorno permite una convergencia más rápida a los datos experimentales). Finalmente, el tercer dato de referencia de pérdida de masa se logró luego de dos iteraciones adicionales con el criterio de Monte Carlo (iteraciones 6 y 7). En la iteración 6, el volumen eliminado (16,13 %) casi alcanzó el valor de referencia correspondiente de pérdida de masa (6,15 %) (desviación de -0,02 %). Como no se cumple el criterio establecido para el cambio de valor de referencia (o final del programa si no hay más valores de referencia) (la desviación debe ser superior al -0,01% para cambiar el valor de referencia), el método aplica otra iteración con el criterio de Monte Carlo (iteración 7). En esa última iteración, se puede observar claramente la influencia del factor de amplificación en la función de probabilidad. En la iteración anterior, el volumen removido fue la diferencia entre 16.13% y 13.67% (2.46% de material removido en la iteración 6). Sin embargo, en la última iteración, el porcentaje de material eliminado fue solo del 0,02 % (16,15–16,13 %). Por tanto, se demuestra de nuevo que el factor de amplificación permite un ajuste muy preciso a los resultados experimentales en un número reducido de iteraciones.

Iteración	Volumen de elementos de contorno (mm <sup>3</sup> )	Volumen de referencia extraído (%)	Criterio Aplicado	Volumen extraído (mm <sup>3</sup> )	Volumen extraído (%)	Desviación (%)
1	4,62	1,12	Monte Carlo	1,60	1,12	0,00
2	5,59	13,68	Elementos de contorno	7,19	5,04	-8,64
3	5,11	13,68	Elementos de Contorno	12,30	8,63	-5,05
4	4,38	13,68	Elementos de Contorno	16,67	11,70	-1,98
5	4,81	13,68	Monte Carlo	19,49	13,67	-0,01
6	5,46	16,15	Monte Carlo	22,99	16,13	-0,02
7	4,84	16,15	Monte Carlo	23,02	16,15	0,00

Tabla 8-3. Resultados simulados de degradación de un filamento de PCL-MCC 98:2

La Figura 8-10 muestra la pérdida de elementos del filamento PCL a las 24 H y a las 48 h.

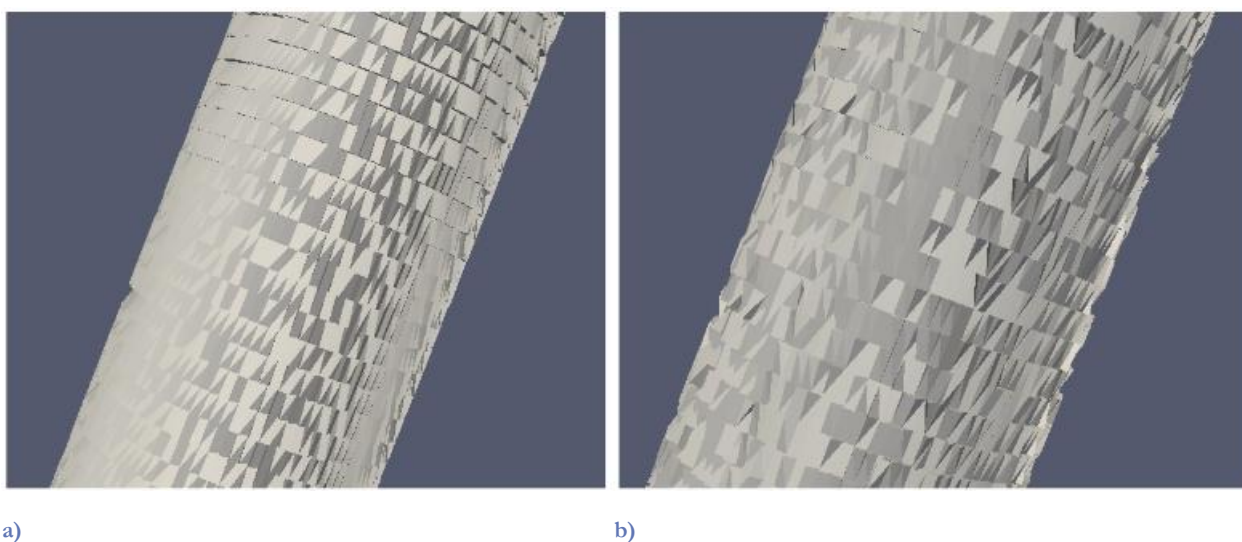


Figura 8-10. Detalle de las geometrías erosionadas. a) a las 24 horas. Y b) a las 48 horas.

### 8.3. SIMULACIÓN DE MONTE CARLO DE LA DEGRADACIÓN ENZIMÁTICA DE UN SCAFFOLD PRISMÁTICO

Para probar la capacidad de la metodología descrita anteriormente se aplica el ensayo a un scaffold de tipología rectangular, de geometrías determinadas para evaluar la

idoneidad del modelo. En este caso se realizará de forma independiente del caso del filamento. La segunda parte del trabajo se centra en la obtención de un modelo que permita la simulación de la degradación hidrolítica basándose en el grado de exposición de la pieza. Sobre esto diversos autores han realizado diferentes metodologías para la degradación hidrolítica. Una metodología ampliamente extendida consiste en la realización de una simulación de Montecarlo de la estructura en función del tipo de la configuración de la red polimérica. Generalmente la degradación se inicia en aquellas zonas que se encuentren más expuestas al líquido.

La estructura del scaffold se compone de zonas amorfas y cristalinas. Las primeras se caracterizan por poseer unos enlaces más débiles, por lo que las partículas de agua pueden penetrar y romper las cadenas. Por su parte las estructuras cristalinas se caracterizan por redes generalmente más fuertes

Gracias a la colaboración del Grupo de Investigación de fabricación Integrada y Avanzada, se dispone de un ensayo experimental en el que se produce la degradación de un scaffold de PCL a partir de geometrías CAD (), y obtenido mediante fabricación de extrusión de material. En un ensayo de degradación enzimática, se ha cuantificado la cantidad de masa disuelta y el módulo de elasticidad, expresada en tanto por ciento, en intervalos entre 24 y 48 horas hasta las 96 horas.

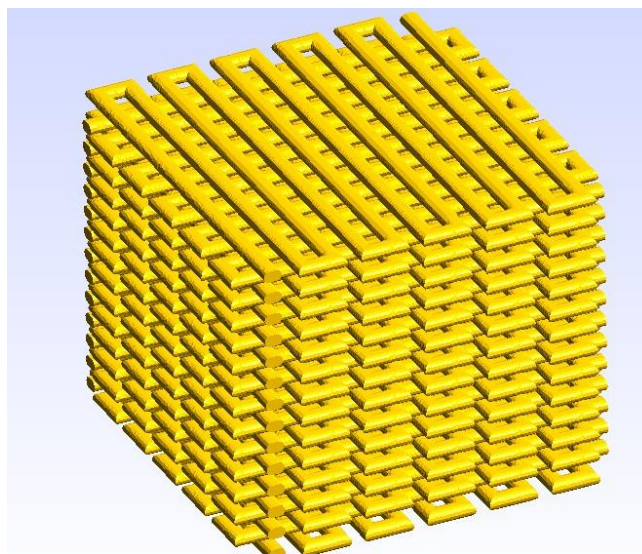


Figura 8-11. Modelo 3d de un scaffold de tipo prismático

### 8.3.1. RESULTADOS

Tiempo de degradación (h)	Promedio % variación de masa	Desv % variación de masa	Promedio E compresión (MPa)	Desv E compresión (MPa)
0,00	0,00	0,00	76,02	1,75
24,00	5,61	1,01	69,83	2,96
48,00	6,13	0,18	68,64	2,73
96,00	11,99	0,46	59,61	2,43

Tabla 8-4. Tabla de resultados experimentales de pérdida de masa porcentual y del módulo de Young promedio correspondientes al scaffold rectangular.

Aplicando la metodología de forma iterativa, y pasando la masa a volumen se observa una rápida convergencia hacia los valores de referencia obtenidos en los ensayos.

Número de Elementos removidos	Volumen disuelto acumulado (m <sup>3</sup> )	Porcentaje de Volumen removido (%)	Porcentaje experimental Pérdida de masa de referencia	Desviación de los datos (%)
144734	2,067E-08	5,370	5,61	-0,24
151844	2,172E-08	5,643	5,61	+0,033
166452	2,387E-08	6,203	6,13	+0,073
332349	4,850E-08	12,601	11,99	+0,611

Tabla 8-5. Tabla de resultados obtenidos del modelo de Monte Carlo de la pieza degradada

Se observa que la variación porcentual entre el volumen extraído y el obtenido en las diferentes etapas de degradación oscila se encuentra entre -0.24 y +0.611, lo cual supone una buena aproximación a los ensayos experimentales.



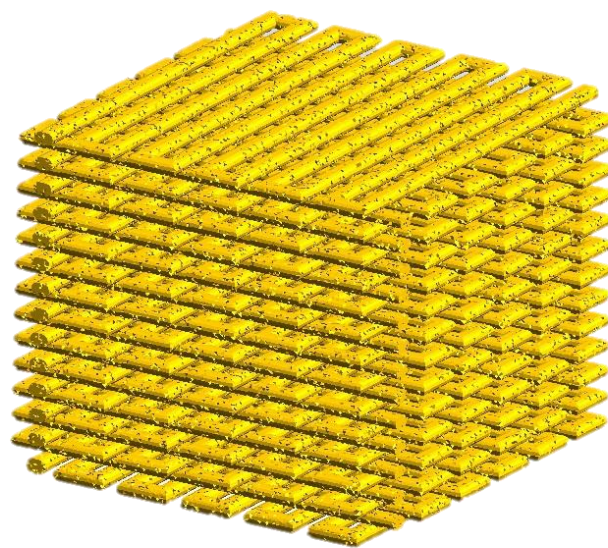


Figura 8-12. Evolución de la degradación (Etapa 1)

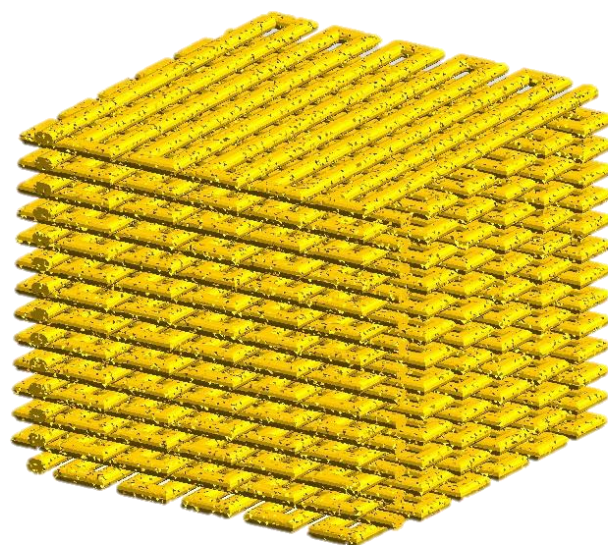


Figura 8-13. Evolución de la erosión (Etapa 2)

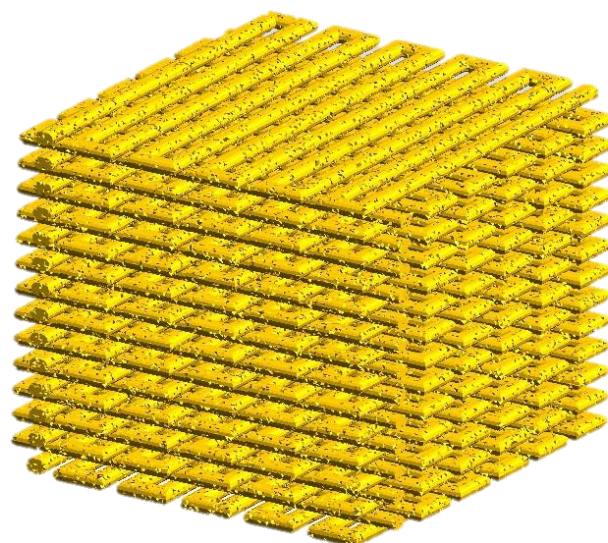


Figura 8-14. Evolución de la erosión (etapa 3)

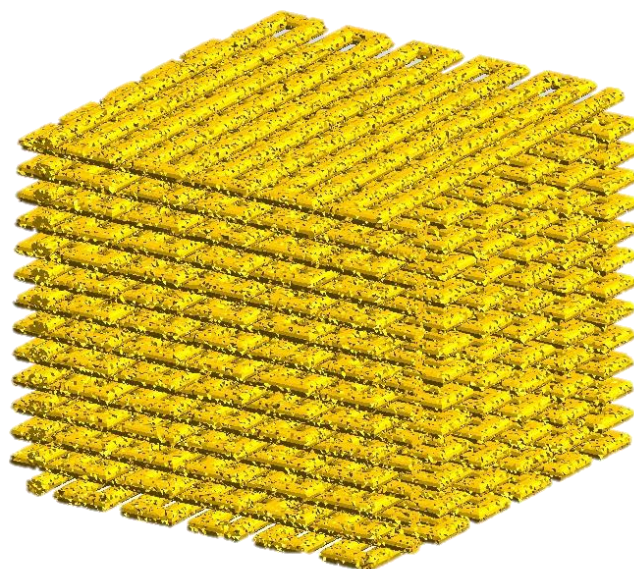


Figura 8-15. Evolución de la erosión (etapa 4)



## 8.4. EXTRAPOLACIÓN DE LOS RESULTADOS OBTENIDOS EN LA SIMULACIÓN DEL FILAMENTO A UNA GEOMETRÍA GENERAL.

Se ha observado que el modelo anterior presenta un buen comportamiento general cuando se le suministran los datos obtenidos de la experimentación. Para poder generalizar el comportamiento obtenido en el modelo del filamento al modelo del scaffold del apartado anterior y que no dependa de los valores experimentales de referencia, se opta por un enfoque predominantemente geométrico que relaciona la cantidad de elementos disueltos con la cantidad de elementos que tras aplicar Monte Carlo serán removidos. Esta relación, se almacena para usarlo como referencia en el modelo nuevo.

### 8.4.1. METODOLOGÍA

Se replica el procedimiento descrito en el diagrama de la Figura 8-7, incluyendo un paso adicional de almacenar la ratio entre el volumen removido y el volumen superficial (Figura 8-16):

$$r_i = \frac{V_{removido_i}}{V_{superficial_{i-1}}} \quad (8-10)$$

Estos se tomarán como valor de referencia en cada una de las etapas simuladas para el filamento, para tomarlo luego de referencia en la geometría del scaffold, evaluando el volumen superficial de la capa exterior en la etapa anterior y el volumen a remover en la etapa actual. Esto se tomará de referencia para la construcción del modelo de degradación superficial.

En aras de conseguir esto, hace falta que la relación inicial entre elementos superficiales en cada caso sea la misma.

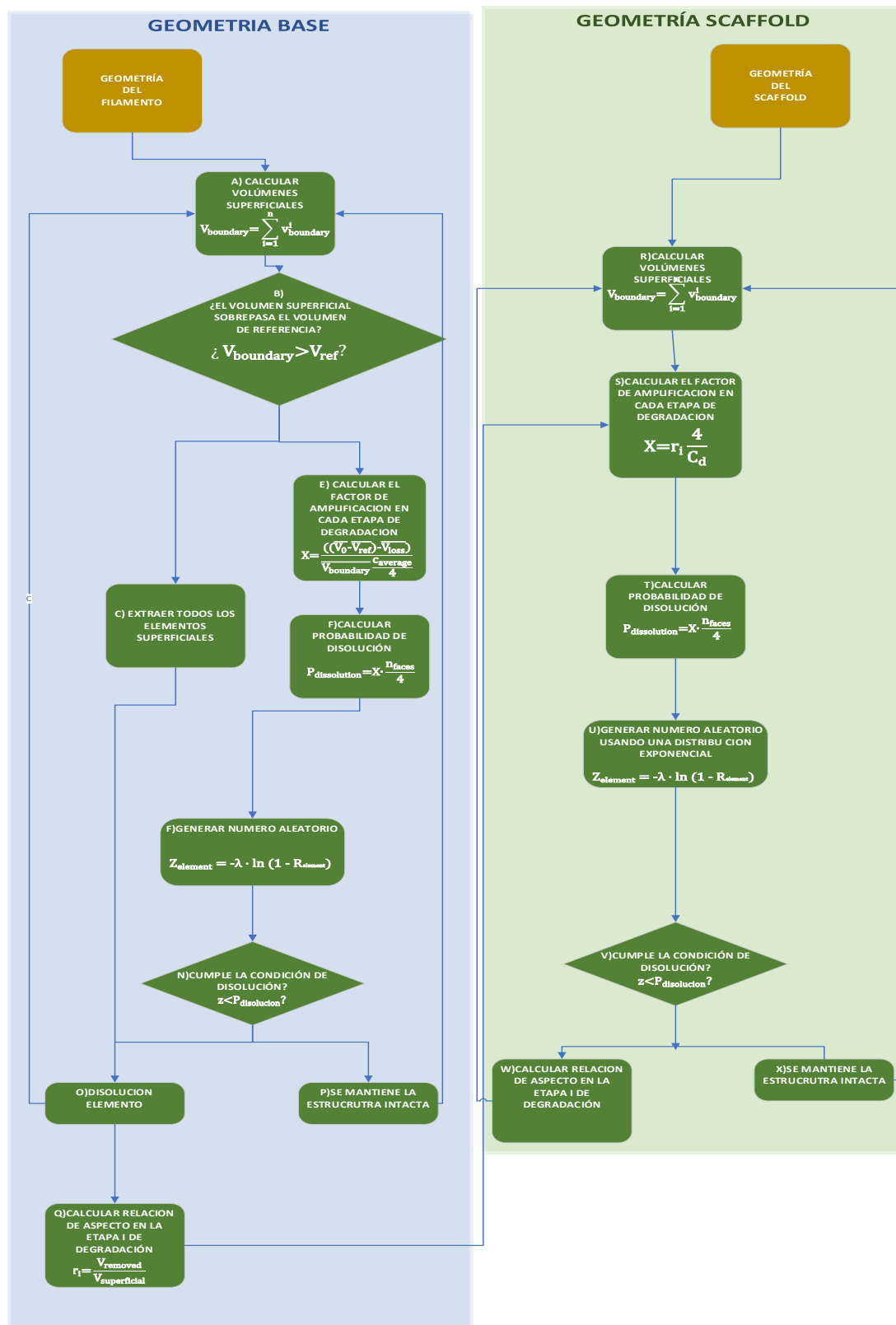


Figura 8-16. Diagrama de flujo del modelo extrapolado de Monte Carlo.

Los pasos del a1) al q1) son los mismos pasos descritos para un filamento (Figura 8-7). En el paso m se calcula la relación entre el volumen extraído y el superficial. Una vez disuelto cada elemento se extrae aplicando una relación de  $r_i = \frac{v_{removido}}{v_{superficial}}$ , estos valores quedan representados en la penúltima columna de la tabla. Se observa que en el caso de aplicar una remoción total entre el valor total y el removido se obtendría una cantidad igual a 1 en el caso de que el volumen de los elementos superficiales sea menor al valor requerido para alcanzar el volumen de referencia.

La relación  $r_i$  se utilizará en las etapas homólogas de degradación de la segunda geometría (etapa s) siendo el nuevo factor de amplificación del tipo:

$$X_i = \frac{r_i}{\frac{C_{avg}}{4}} \quad (8-11)$$

Una vez calculado dicho factor se multiplicará por el ratio de caras en contacto con el medio para la obtención de la probabilidad elemental de disolución para el elemento  $i$ .

$$P_{elemental}^i = X_i \cdot \frac{c_e^i}{4} \quad (8-12)$$

El valor de la probabilidad obtenida se comparará con un numero aleatorio. Si este número queda por debajo del anterior se procederá a extraer dicho elemento de la geometría.

$$Z_{elemental} < P_{elemental}^i \rightarrow V_{scaffold} = V_{scaffold} - v_i \quad (8-13)$$

El modelo geométrico a simular del filamento es el mismo utilizado en el epígrafe 8.2, aplicando un tamaño de mallado de 20 mm:

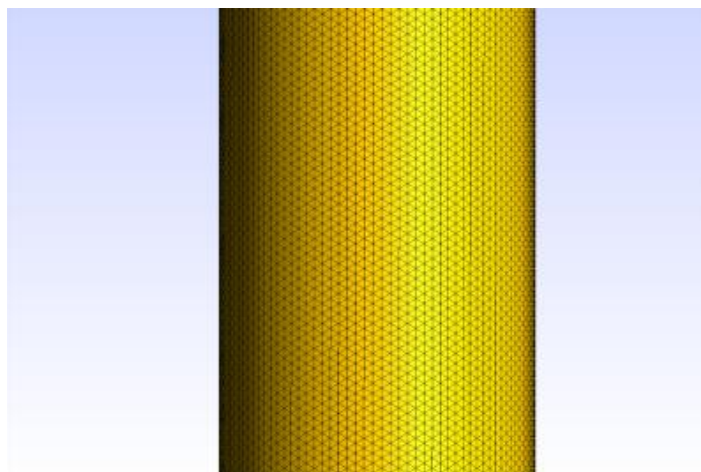


Figura 8-17. Representación de la malla del filamento.

El modelo usado para validar la metodología del modelo extrapolado es el scaffold del epígrafe 8.3, aplicando un discretizado uniforme de 0,06 mm:

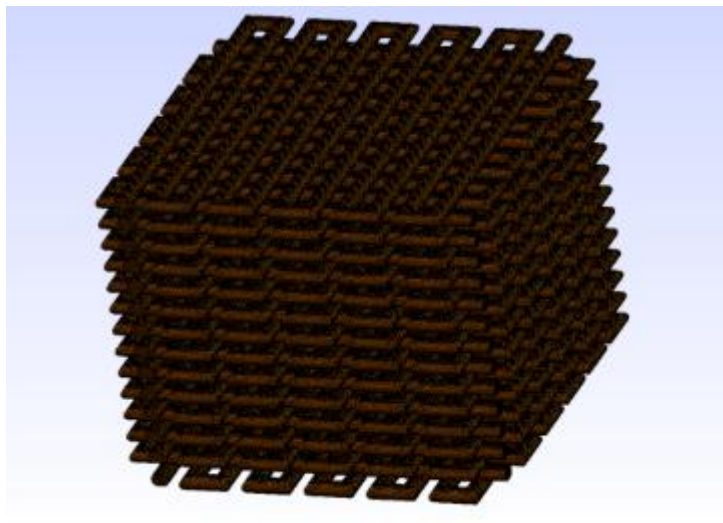


Figura 8-18. Representación de malla del scaffold.

#### 8.4.2. RESULTADOS

En la Tabla 8-6 se representan los resultados de referencia de la ratio  $vr/vs$  de la simulación de Monte Carlo del filamento de PCL- MCC92 con un tamaño de elemento de 20 mm:

vr/vs filament
0,17928
0,848924
0,118736
0,00190122
0,0426273

Tabla 8-6. Tabla de resultados obtenidos de vr/vs en cada etapa de degradación del filamento

En la Tabla 8-7 se observa que a partir de el volumen extraído se aproxima al primer valor de referencia de volumen de 5,61 tomando

La penúltima columna de la tabla se usa como referencia para calcular el proceso de degradación de un scaffold del mismo material.

Iteración	Número de elementos extraídos	Volumen de los elementos superficiales (mm <sup>3</sup> )	Porcentaje de volumen extraído de referencia	Metodología aplicada	Volumen disuelto (mm <sup>3</sup> )	Vr/Vs	Porcentaje de volumen extraído
0	8.645.395	89,0575	5,61	Monte Carlo	15,9799	0,179411	4,08406
1	6.421.476	97,8416	5,61	Monte Carlo	89,1253	0,74761	22,781
2	6.125.437	103,153	11,99	Monte Carlo	100,891	0,114063	25,7884
3	6.120.925	103,849	11,99	Monte Carlo	101,075	0,0017723	25,8355
4	6.017.375	103,849	11,99	Monte Carlo	105,31	0,040735	26,9178

Tabla 8-7. Valores representativos de pérdida volumétrica en las diferentes etapas de degradación para el modelo simulado.

## 8.5. CONCLUSIONES GENERALES

En este trabajo se ha observado la aplicación de un modelo estocástico-experimental de Monte Carlo en el cual se aplica la erosión superficial estática de un filamento, en primer lugar, y de un scaffold en segundo lugar, mostrando un buen ajuste a los resultados experimentales. En lo que respecta al modelo numérico, una de las ventajas de aplicar este modelo reside en la presencia del factor de amplificación, que acelera la búsqueda de la solución aproximarse a un valor de referencia, consiguiendo así un buen ajuste experimental en un número relativamente bajo de iteraciones.

El modelo presenta una limitación clara: los valores experimentales se usan para alimentar dicho modelo, lo cual limita la posibilidad de extrapolar los resultados a geometrías más complejas. Esto se traduce en una dependencia significativa de los datos experimentales, lo que reduce la flexibilidad y aplicabilidad del modelo en escenarios donde tales datos no están disponibles o son difíciles de obtener.

Se ha intentado realizar esta metodología tomando el caso base del filamento como referencia y, mediante el registro de la ratio entre el número de elementos superficiales removibles y el número de elementos totales en el contorno sin poder encontrar una similitud importante.

Otra desventaja del modelo es la ausencia de relación de los resultados obtenidos con las propiedades intrínsecas del material. El modelo únicamente indica la pérdida de masa y volumétrica, sin proporcionar información detallada sobre el proceso de degradación en sí. Esto significa que, aunque el modelo puede predecir de manera efectiva la cantidad de material degradado, no ofrece una comprensión profunda de los mecanismos subyacentes que dirigen dicho proceso.

Para solventar este problema, se propone la integración de los modelos anteriores con modelos que den cuenta de la evolución de las propiedades intrínsecas del material a lo largo del proceso de degradación. Esto podría incluir la incorporación de parámetros que reflejen cambios en la estructura molecular, la cristalinidad, y otras propiedades fisicoquímicas del polímero. Además, sería beneficioso desarrollar modelos híbridos que combinen elementos estocásticos y deterministas, permitiendo una mejor representación tanto de la variabilidad inherente a los datos experimentales como de los principios fundamentales que gobiernan la degradación del material.

Asimismo, es crucial que futuros trabajos se enfoquen en la validación de estos modelos integrados mediante pruebas experimentales exhaustivas. Esto no solo mejoraría la precisión y confiabilidad de las predicciones del modelo, sino que también proporcionaría una base más sólida para la extrapolación de los resultados a diferentes polímeros y condiciones de uso.

En resumen, aunque el modelo estocástico-experimental de Monte Carlo presenta ciertas ventajas en términos de ajuste a datos experimentales y rapidez de convergencia, sus limitaciones en cuanto a la extrapolación de resultados y la comprensión de los mecanismos de degradación deben ser abordadas. La integración con modelos que consideren las propiedades intrínsecas del material y la validación experimental son pasos necesarios para mejorar la aplicabilidad y precisión de estos modelos en el estudio de la degradación hidrolítica de polímero.

## 9. TÉCNICAS DE MODELIZACIÓN DE LA DEGRADACIÓN HIDROLÍTICA DE UN MATERIAL POLIMÉRICO.

Diferentes modelos se han presentado para la modelización de la degradación hidrolítica del polímero. Estos modelos se basan, en su mayoría en la segunda ley de Fick[91], incorporando términos fuente que sirven para capturar la dinámica del proceso hidrolítico. De entre los modelos existentes, se optará por el modelo propuesto por Wang et al [1], entre otros motivos, por tratarse de un modelo determinista que se ha usado ampliamente en la literatura, y cuya resolución en 2 pasos permite obtener una solución numérica de forma eficiente. La implementación de estas ecuaciones servirá como base para la realización de las sucesivas propuestas de modelización que se tratan en el presente trabajo.

### 9.1. REACCIÓN DIFUSIÓN

Según Wang et al [1], las ecuaciones (adimensionales) que rigen la degradación hidrolítica de un polímero vienen definidas por las concentraciones normalizadas de monómeros ( $\overline{C_m}$ ) y ésteres ( $\overline{C_e}$ ) para cada instante  $t$ , de acuerdo a las siguientes expresiones adimensionalizadas:

$$\begin{aligned} \frac{d\overline{C_m}}{d\bar{t}} &= \overline{k_1}\overline{C_e} + \overline{C_e}\overline{C_m^n} + \nabla \cdot \{\overline{D_o}[1 + \alpha(1 - (\overline{C_m} + \overline{C_e}))]\nabla(\overline{C_m})\} \\ \frac{d\overline{C_e}}{d\bar{t}} &= -(\overline{k_1}\overline{C_e} + \overline{C_e}\overline{C_m^n}) \end{aligned} \quad (9-1)$$

Donde  $\overline{k_1}$ ,  $n$  y  $\alpha$  son, respectivamente, la tasa de reacción hidrolítica no catalítica, la constante de disociación (que suele tomar como valor 0,5) y  $\alpha$  un parámetro de difusión fijado normalmente a 4,5.



### Condiciones de contorno:

- $C_m$

$$\begin{aligned} \frac{\partial C_m}{\partial x} &= 0 \text{ in symmetry planes} \\ C_m &= 0 \text{ in free surface} \end{aligned} \quad (9-2)$$

La primera condición es del tipo Neumann y significa que el flujo de monómeros a través de los planos de simetría es 0. En el contexto que nos ocupa, se asumen como planos de simetría aquellos que corresponden a la sección de corte de la pieza.

La segunda condición, del tipo Dirichlet asume que una vez los monómeros alcanzan la superficie en contacto con el agua, se desprenden del dispositivo.

- $C_e$

La condición de contorno relativa a la concentración de ésteres no está especificada en el artículo de Wang. Sin embargo, se puede inferir integrando la segunda ecuación diferencial teniendo en cuenta que para la superficie libre  $C_m=0$ . Así, sustituyendo este valor en la segunda ecuación del sistema (9-1):

$$\frac{d\bar{C}_e}{d\bar{t}} = -(\bar{k}_1 \bar{C}_e + \bar{C}_e * 0) = -\bar{k}_1 \bar{C}_e \quad (9-3)$$

Separando variables e integrando a ambos lados de la expresión anterior:

$$\int_{\bar{C}_{e0}}^{\bar{C}_e} \frac{d\bar{C}_e}{\bar{C}_e} = - \int_0^{\bar{t}} \bar{k}_1 d\bar{t} \quad (9-4)$$

Teniendo en cuenta que para  $\bar{t} = 0$ ,  $\bar{C}_{e0}=1$ :

$$(9-5)$$

$$\ln(\overline{C_e}) = -\overline{k_1}\overline{t}$$

aplicando inversas de logaritmos:

$$\overline{C_e} = e^{-\overline{k_1}\overline{t}} \quad (9-6)$$

Se observa, pues, que se trata, por tanto, de una condición de contorno no estacionaria y uniforme en la frontera del material con el medio.

### Condiciones iniciales:

La concentración de monómeros y ésteres, en sus valores estándar durante el instante inicial, toma los valores 1 y 0 respectivamente:

$$\begin{aligned} \overline{C_{e0}} &= 1 \\ \overline{C_{m0}} &= 0 \end{aligned} \quad (9-7)$$

### Formulación variacional del problema:

Para la resolución del sistema de ecuaciones en derivadas parciales (9-1) por medio del uso de elementos finitos, se deberá pasar los términos de las ecuaciones al primer miembro y multiplicar sendas ecuaciones funciones auxiliares o de ensayo, definidas en el mismo subespacio que las variables del proceso con la particularidad de que se anulan en la frontera:

$$\begin{aligned} \left( \frac{d\overline{C_m}}{d\overline{t}} - (\overline{k_1}\overline{C_e} + \overline{C_e}\overline{C_m^n}) - \nabla \cdot \{ \overline{D_o}[1 + \alpha(1 - (\overline{C_m} + \overline{C_e}))]\nabla(\overline{C_m}) \} \right) \cdot v &= 0 \\ \left( \frac{d\overline{C_e}}{d\overline{t}} + (\overline{k_1}\overline{C_e} + \overline{C_e}\overline{C_m^n}) \right) \cdot w &= 0 \end{aligned} \quad (9-8)$$

Integrando respecto al dominio y aplicando el teorema de la divergencia para el termino difusivo  $\nabla \cdot \{ \overline{D_o}[1 + \alpha(1 - (\overline{C_m} + \overline{C_e}))]\nabla(\overline{C_m}) \}$  tenemos:

$$\begin{aligned} \int_{\Omega} \left( \frac{d\overline{C_m}}{d\bar{t}} - (\overline{k_1 C_e} + \overline{C_e C_m^n}) \right) \cdot v + \int_{\Omega} [\overline{D_o} [1 + \alpha(1 - (\overline{C_m} + \overline{C_e}))]] \cdot \nabla(\overline{C_m}) \cdot \nabla(v) \\ = 0 \\ \int_{\Omega} \left( \frac{d\overline{C_e}}{d\bar{t}} + (\overline{k_1 C_e} + \overline{C_e C_m^n}) \right) \cdot w = 0 \end{aligned} \quad (9-9)$$

La resolución de las ecuaciones (9-9) se pueden realizar de forma combinada sumando las dos expresiones anteriores resultando en una expresión única:

$$\begin{aligned} \int_{\Omega} \left( \frac{d\overline{C_m}}{d\bar{t}} - (\overline{k_1 C_e} + \overline{C_e C_m^n}) \right) \cdot v + \int_{\Omega} \left( \frac{d\overline{C_e}}{d\bar{t}} + (\overline{k_1 C_e} + \overline{C_e C_m^n}) \right) \cdot w \\ + \int_{\Omega} [\overline{D_o} [1 + \alpha(1 - (\overline{C_m} + \overline{C_e}))]] \cdot \nabla(\overline{C_m}) \cdot \nabla(v) = 0 \end{aligned} \quad (9-10)$$

De ahora en adelante, para las simulaciones, se empleará un esquema numérico que trata de forma secuencial las ecuaciones de reacción-difusión (9-9). Aunque el sistema original es acoplado- ya que las concentraciones de  $\overline{C_m}$  y  $\overline{C_e}$  aparecen en sendas expresiones- su resolución se puede hacer numéricamente siguiendo un esquema desacoplado, por su mayor simplicidad en su implementación y el menor costo computacional por iteración, al resolverse en dos etapas.

#### 9.1.1. IMPLEMENTACIÓN ESTANDAR DE LAS ECUACIONES DE REACCIÓN DIFUSIÓN EN FREEFEM++

En la Figura 9-1 y la Figura 9-2 se representan de forma esquematizada como pasar de las ecuaciones diferenciales de parte que sirve para pasar del modelo variacional. La estructura básica para plantear una ecuación diferencial en su forma variacional en FreeFem++ es por medio de la instrucción **problem** en el que se indica el nombre del problema a resolver, la variable a determinar y la función auxiliar que se usa para el modelo variacional **problem monomer (Cm, v)**, es usando, para cada término de la ecuación, el comando **int3d**, o, en su defecto, **int2d** para un estudio bidimensional que contiene como argumentos de entrada la malla de la pieza discretizada, y, opcionalmente, un argumento de optimización del tipo **optimize = 1**.

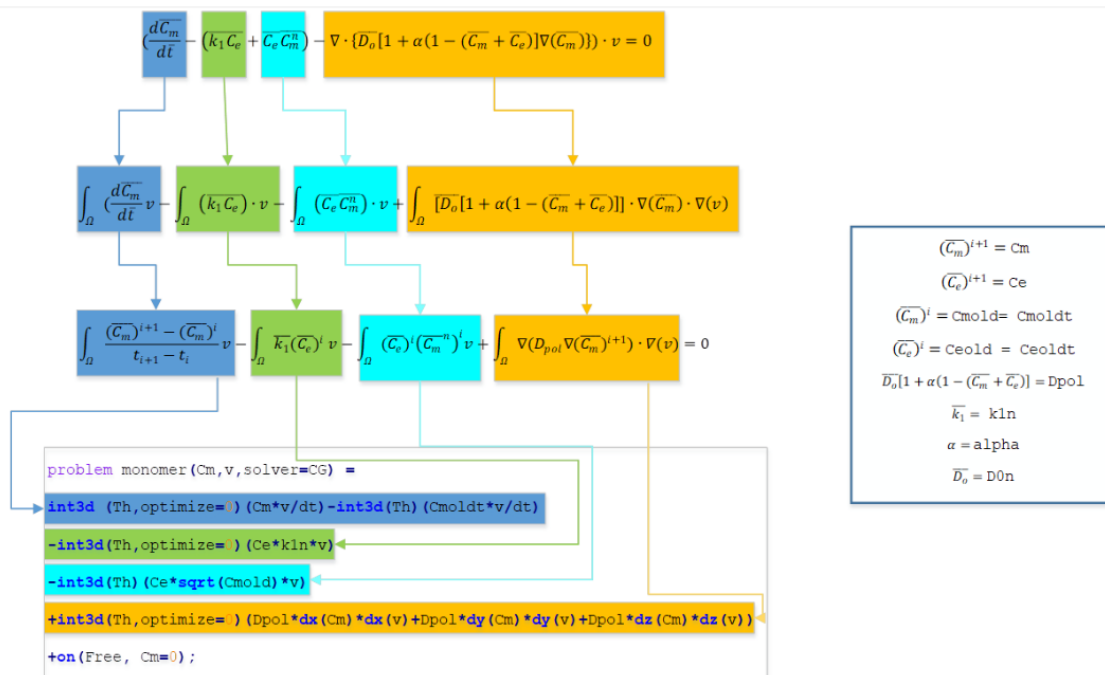


Figura 9-1. Implementación de las ecuaciones fenomenológicas de Wang en términos de la variación de la concentración de monómeros.

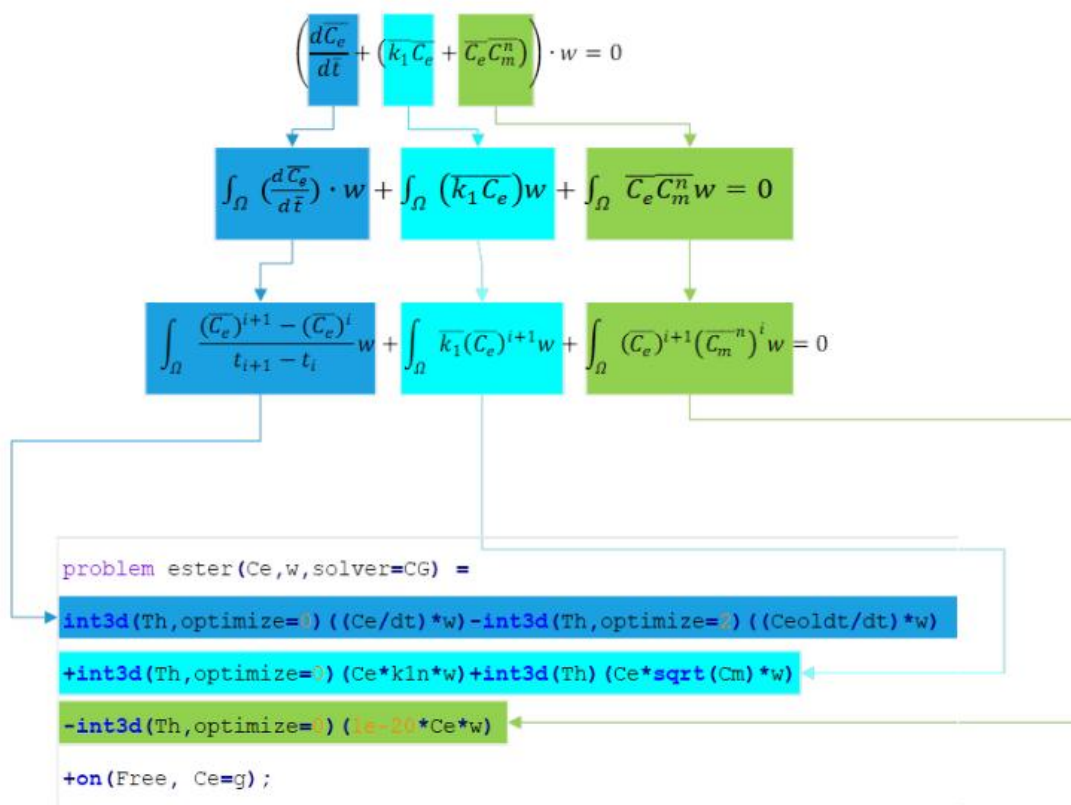


Figura 9-2. Implementación de las ecuaciones fenomenológicas de Wang en términos de la variación de la concentración de ésteres

Para el cálculo de la ecuación correspondiente a la concentración de monómeros, teniendo en cuenta que el término fuente  $\overline{k_1 C_e} + \overline{C_e C_m^n}$  es no lineal, se procederá a su linealización utilizando el método del punto fijo [92], con un criterio de convergencia establecido de tal manera que la suma de la norma H1 del error absoluto  $e$  entre 2 iteraciones consecutivas sea menor que una tolerancia establecida[93]:

$$\|e\|_{H_1}^2 = \sqrt{\|e\|_{L^2}^2 + |e|_{H^1}^2} < \epsilon \quad (9-11)$$

El error absoluto acumulado en el dominio  $\Omega$  entre 2 mediciones sucesivas de la concentración de monómeros definida en la malla de estudio se expresa como:

$$e = \int_{\Omega} (C_m^i - C_m^{i-1}) d\Omega \quad (9-12)$$

La norma  $L^2$  del error se expresa en como la integral extendida al dominio de estudio  $\Omega$  del cuadrado del error:

$$\|e\|_{L^2}^2 = \int_{\Omega} |e(x, y, z)|^2 d\Omega \quad (9-13)$$

El cuadrado de la seminorma H1 del error se expresa como la integral extendida al dominio del cuadrado del gradiente del error:

$$|e|_{H^1}^2 = \int_{\Omega} |\nabla e|^2 d\Omega \quad (9-14)$$

Para la implementación en FreeFem++, que se muestra a continuación, se inicializa los errores cuadráticos de la norma L2 y la seminorma H1 a la unidad. Como criterio de parada se asume un valor de  $\epsilon = 10^{-6}$ . Dentro de la instrucción “while” se van ejecutando iterativamente hasta que la condición previa se satisfaga:

```
real ErrorL2 = 1.;
real ErrorH1 = 1.;
```

```

while(sqrt(ErrorL2+ErrorH1) > 1.e-6)    // bucle de linealización método
de punto fijo
{

    Dpol=D0n*(1+alpha*(1-((Ce+Cmold))))); //Actualización difusividad

    reactiondiffusion; // se obtiene Cm , Ce es el calculado con problem
    ester

    error=Cm-Cmold;
    ErrorH1 = int3d(Th)(dx(error)^2+dy(error)^2+dz(error)^2);
    ErrorL2 = int3d(Th)(error^2);
    cout << " Error seminorma H1 = " <<sqrt( ErrorH1) <<endl;
    cout << " Error L2 = " << sqrt(ErrorL2) <<endl;
    cout << " Error norma H1 = " << sqrt(ErrorL2+ErrorH1) <<endl;

    Cmold=Cm;
}      // fin iteracion de linealización

```

La difusividad  $\overline{D_o}[1 + \alpha(1 - (\overline{C_m} + \overline{C_e}))]$ , al no ser constante se actualizará en cada etapa de cálculo con los valores previos de  $\overline{C_m}$  y  $\overline{C_e}$ .

Nos queda por tanto la siguiente implementación numérica en FreeFem++:

```

load "msh3" load "iovtk" load "gmsh" load "MUMPS_seq"

mesh3 Th=readmesh3("Model 1_2.mesh");

plot(Th);
real k1=0.003;
real k2=0.002;
real n=0.5;
real D=0.1;
real l=0.5;
real Free=1;
real Symmetry=2;
real total = 0;

//NORMALIZED VALUES

real t=0;
real dt=0.01;
real T=4;

```

```
fespace Vh(Th,P13d);
Vh Cm, Ce, v, w, Cmold,Cmoldt,Cm0,Ce0,Ceoldt,Dpol,error,g;

Ce0=1;
Ceoldt=Ce0;
Ce=1;
Cm0=0;
Cmoldt=Cm0;
Cmold=Cm0;

real k1n=0.0008;
real D0n=1;

real alpha=4.5;

problem reactiondiffusion(Cm,v) =
  int3d(Th,optimize=0)(Cm*v/dt)
  -int3d(Th)(Cmoldt*v/dt)
  -int3d(Th,optimize=0)(Ce*k1n*v)
  -int3d(Th)(Ce*(Cmold^0.2)*v)
  +int3d(Th,optimize=0)(Dpol*dx(Cm)*dx(v)+Dpol*dy(Cm)*dy(v)+Dpol*dz(Cm)*dz(
  v))
  +int2d(Th,Symmetry)(1e-10*v) //Condición Neumann en el plano de simetría
  +on(Free, Cm=0);

problem reaction(Ce,w) =
  int3d(Th,optimize=0)((Ce/dt)*w)-int3d(Th,optimize=2)((Ceoldt/dt)*w)
  +int3d(Th,optimize=0)(Ce*k1n*w)+int3d(Th)(Ce*(Cm^0.2)*w)
  // Cm es conocidoes Cm0 en t=0 y es el Cm calculado en
  reactiondiffusion
  -int3d(Th,optimize=0)(1e-20*Ce*w)
  +on(Free, Ce=g);

Cmold=Cm0;

ofstream dataunit ( "monomerversustime_standard.txt");
dataunit << " "<<"Time"
          << "Monomer_conc"<< " \n";

{
  ofstream resultsstandard( "Resultsstandard. txt" );
  resultsstandard << " " <<" Time "<<" , "
                << " " <<" Cmax "<<" , "
                << " " <<" Cmin "<<" , "
                << " " <<" Diffcoef "<<" , "
                << " " <<" Ceavg "<<" , "
```

```
        << " " <<" Cemax " <<"\n";
    }

    for(real t=0;t<T;t+=dt) // bucle de iteraciones en el tiempo t desde t=0
    hasta t=T

    {

        g=exp(-(k1n)*t);

        real ErrorL2 = 1.;
        real ErrorH1 = 1.;

        while(sqrt(ErrorL2+ErrorH1) > 1.e-6) // bucle de linealización método
        de punto fijo
        {

            Dpol=D0n*(1+alpha*(1-((Ce+Cmold))))); //Actualización difusividad

            reactiondiffusion; // se obtiene Cm , Ce es el calculado con problem
            reaction

            error=Cm-Cmold;
            ErrorH1 = int3d(Th)(dx(error)^2+dy(error)^2+dz(error)^2);
            ErrorL2 = int3d(Th)(error^2);
            cout << " Error seminorma H1 = " <<sqrt( ErrorH1) <<endl;
            cout << " Error L2 = " << sqrt(ErrorL2) <<endl;
            cout << " Error norma H1 = " << sqrt(ErrorL2+ErrorH1) <<endl;

            Cmold=Cm;
        } // fin iteracion de linealización

        for(int j=0;j<Th.nv;j++){
            if (Cm[][j]<0){
                Cm[][j]=abs(Cm[][j]);}
        }

        Cmoldt=Cm;

        real Cmavg = int3d(Th)(Cm)/Th.measure;
        cout<<"el valor medio de Cm es " <<Cmavg<< endl;

        ofstream dataunit ( "monomerversustime_standard.txt",append);
```



```

        dataunit << " " << t
                << " " << Cmavg<< "\n";

reaction;
Ceoldt=Ce;

for(int j=0;j<Th.nv;j++){
if (Ce[][j]<0){
Ce[][j]=0;}}

cout<<"el valor maximo de la concentracion de monomeros es
"<<Cm[].max<<endl;
cout<<"el valor maximo de la concentracion de esteres es
"<<Ce[].max<<endl;

real Ceavg = int3d(Th)(Ce)/Th.measure;

real Dpolavg=D0n*(1+alpha*(1-((Ceavg+Cmavg)/1.)));

ofstream resultsstandard ( "Resultsstandard. txt" ,append);
resultsstandard << " " << t << " , "
                << " " << Cmavg << " , "
                << " " << Cm[].max << " , "
                << " " << Dpol[].max << " , "
                << " " << Ceavg << " , "
                << " " << Ce[].max << "\n";

int[int] order = [1, 1]; // order (default in 0)
cout << " t = " << t <<endl;
cout<<"el valor medio de Dpol = "<<Dpolavg<< endl;
cout<<" concentracion promedio de monomeros es "<<Cmavg<<endl;
cout<<" concentracion promedio de esteres es "<<Ceavg<<endl;

}
// FIN ITERACION DE tiempo

ofstream commandunit ( "gnuplot_contour_commands_xy_data.txt" );
commandunit << "# gnuplot_contour_commands_xy_data.txt\n";
commandunit << "# usage: gnuplot <
gnuplot_contour_commands_xy_data.txt\n";
commandunit << "#\n";
commandunit << "set term png\n";
commandunit << "set terminal png size 1200,800";
commandunit << "set output 'gnuplot_contour.png'\n";
commandunit << "set nokey\n";
commandunit << "set xlabel 'Time'\n";

```

```
commandunit << "set ylabel 'normalized monomer concentration'\n";
commandunit << "set title 'Solution contours'\n";
commandunit << "set timestamp\n";
commandunit << "plot 'monomerversustime.txt' with lines lt 1\n";
```

En la Figura 9-3 se observa la evolución del valor promedio de la concentración de monómeros. Se observa, una fase inicial de subida, que corresponde a la acumulación de los subproductos en la matriz polimérica. En torno a  $\bar{t} = 0,5$  se alcanza el valor máximo de concentración, dando paso a una fase de descenso que se traduce a una liberación de las partículas al medio.

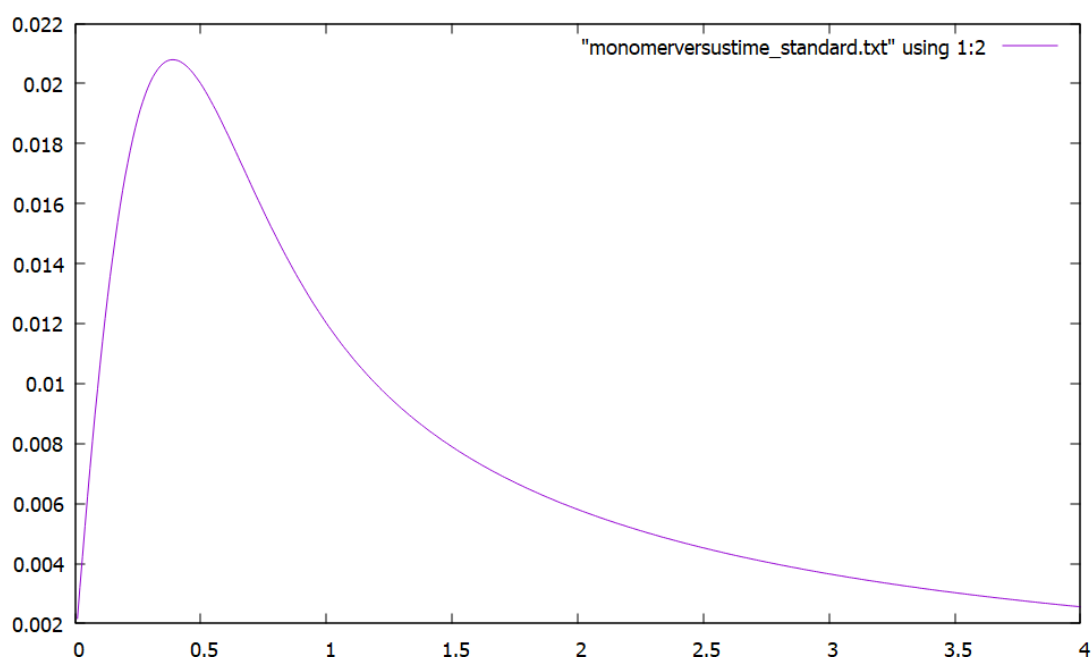


Figura 9-3. Evolución temporal de la concentración de monómeros en el proceso degradativo para  $n = 0,2$

### 9.1.2. IMPLEMENTACION MATRICIAL DE LAS ECUACIONES DE REACCIÓN DIFUSION

El método matricial del FreeFEM [94] ofrece varias ventajas respecto al método estándar, especialmente en la resolución de problemas de ecuaciones en derivadas parciales (EDP).

Aquí hay algunas razones clave:

1. **Eficiencia Computacional:** El método matricial permite una formulación más eficiente y compacta de los problemas, lo que reduce el tiempo de cálculo y mejora el rendimiento computacional.

2. **Flexibilidad:** FreeFEM permite definir y manipular matrices de manera flexible, lo que facilita la implementación de condiciones de contorno y la adaptación a diferentes tipos de problemas.
3. **Precisión:** La formulación matricial permite un control más preciso sobre los errores numéricos y la estabilidad de las soluciones, lo que resulta en soluciones más exactas y fiables.
4. **Modularidad:** La estructura matricial facilita la reutilización de código y la implementación de esquemas de solución más complejos, como métodos iterativos y técnicas de optimización.

En resumen, el método matricial del FreeFEM es superior al método estándar en términos debido a su eficiencia, flexibilidad y modularidad.

En la implementación de las ecuaciones de reacción difusión la instrucción `problem`, vista en Figura 2-1, se sustituye por `varf`:

```
varf monomer(Cm, v)
```

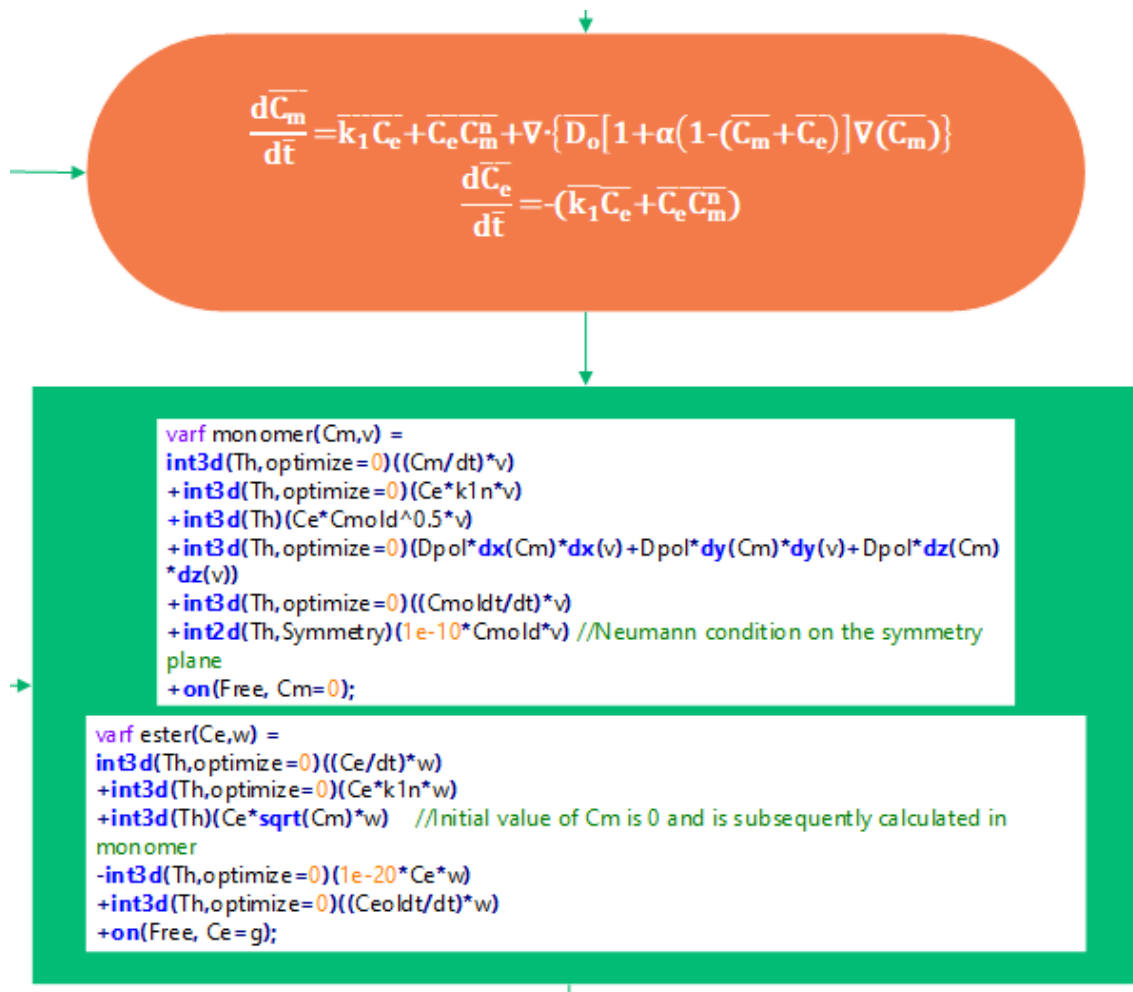


Figura 9-4. Implementación matricial de las ecuaciones hidrolíticas de Wang en FreeFem++

Asimismo, para poder resolver el sistema numérico resultante de forma matricial se crean sendas matrices  $A_m$  y  $A_e$ , respectivamente.

```

matrix Am = monomer(Vh, Vh, solver = GMRES);
matrix Ae = ester(Vh, Vh, solver = GMRES);
  
```

Además, es necesario definir el vector de términos independientes, que contendrá a los valores definidos en la frontera como condición de contorno Dirichlet:

```

Cm[] = Am^-1 * Bm;
Ce[] = Ae^-1 * Be;
  
```

La resolución de las ecuaciones se realizará de la forma

Para la validación de esta implementación se recurrirá se utilizará el parámetro  $n$ , que representa el exponente del segundo término de la primera ecuación del sistema (9-1), y se verá la evolución del promedio de  $C_m$  a lo largo del proceso degradativo de la celda unitaria.

Representando la evolución de monómeros para  $n$  utilizando esta técnica (Figura 9-5) se observa una tendencia similar a la que se obtuvo en la implementación anterior (Figura 9-3). Hay un tramo inicial de subida de la concentración  $\overline{C_m}$ . En el momento de la liberación de monómeros al medio circundante, se alcanza un pico de la concentración, para dar luego a una disminución. En este sentido, se observa que el uso de una implementación estándar y una implementación matricial no afecta el resultado obtenido de  $C_m$ . El momento en el que se inicia la fase de descenso se produce en torno a 0,3:

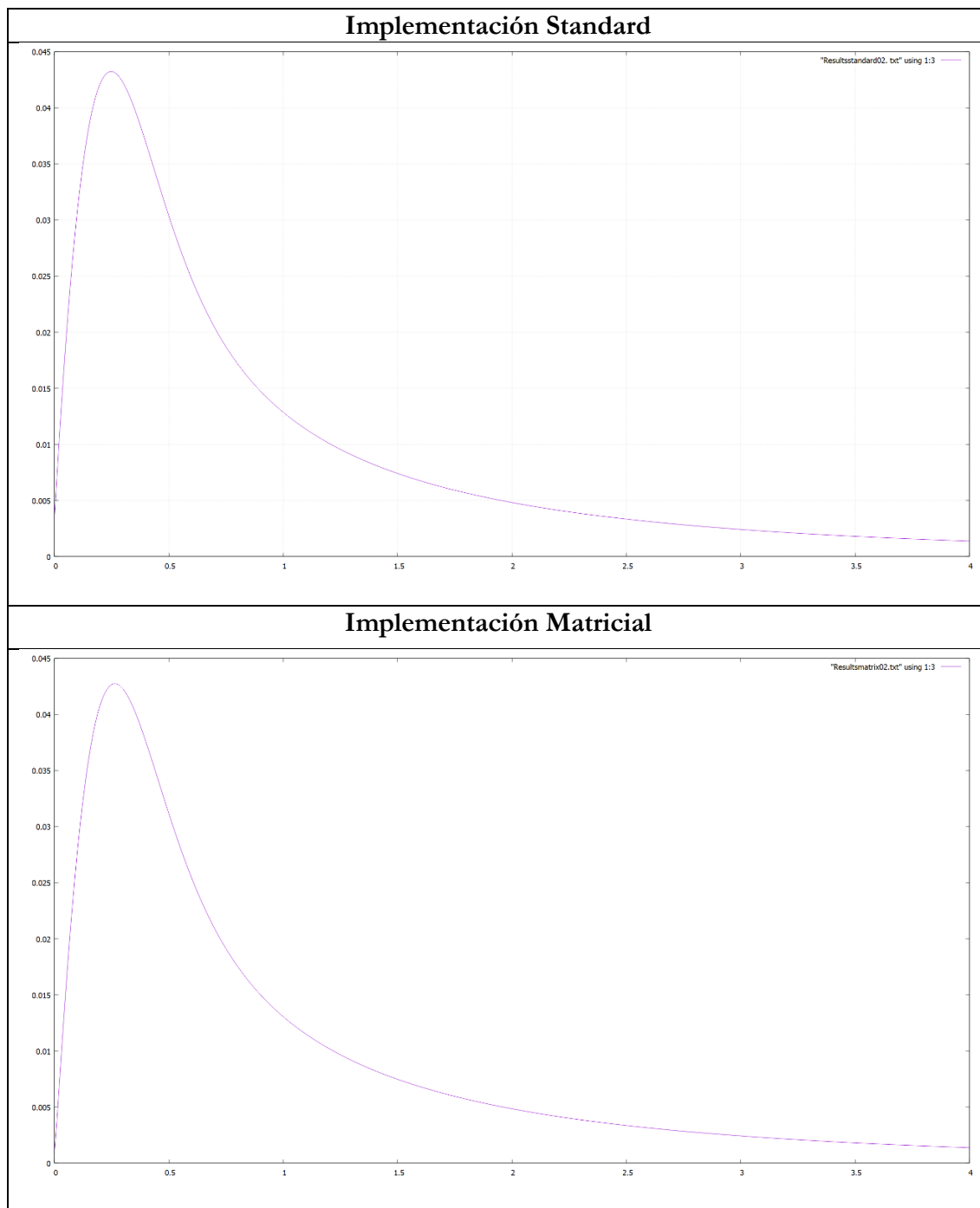


Figura 9-5. Comparativa de la evolución de monómeros en el modelo standard y matricial con valores del parámetro  $n = 0,2$

Para  $n = 0,5$  se observa una tendencia similar en ambas implementaciones (Figura 9-6), observándose un máximo para el parámetro  $\bar{c}_m$  ligeramente por encima de 0,02, para  $\bar{t}$  en torno a 0,4:

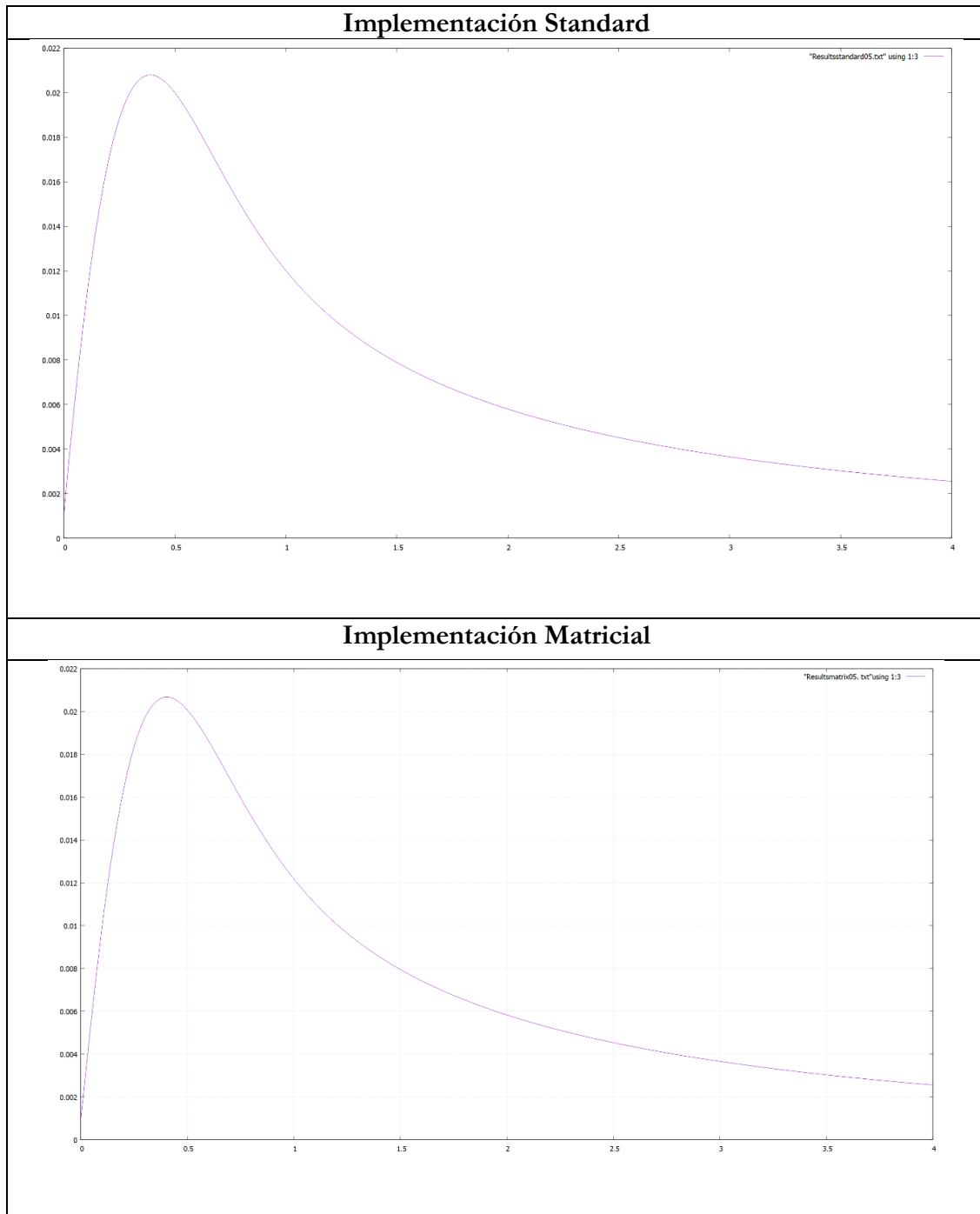


Figura 9-6. Comparativa de la evolución de monómeros en el modelo standard y matricial con valores del parámetro  $n = 0,5$

## 10. MODELIZACIÓN DE LA EVOLUCIÓN DE LAS PROPIEDADES INTRÍNSECAS.

### 10.1. INTRODUCCIÓN.

A lo largo de los años, la mayor parte de los modelos presentados se centraron en cuantificar la degradación a través de los cambios en el peso molecular o el grado de dispersión [1], [39], [95], [96]. Sin embargo, ha habido una escasez de modelos que relacionen los cambios de la degradación con las propiedades del material [32]. En los últimos 15 años, se ha experimentado una serie de avances en la modelización de la degradación de las propiedades mecánicas de polímeros en ingeniería tisular, asociándolos a parámetros intrínsecos del proceso.

En relación a la degradación hidrolítica, la modelización del proceso de difusión acuosa dentro de la matriz polimérica, previo al proceso de rotura de cadenas, se ajusta a una Ley de Fick [91] generalizada de la siguiente forma:

$$\frac{\partial[H_2O]}{\partial t} = D_x \frac{\partial^2[H_2O]}{\partial x^2} + D_y \frac{\partial^2[H_2O]}{\partial y^2} + D_z \cdot \frac{\partial^2[H_2O]}{\partial z^2} \quad (10-1)$$

Donde  $[H_2O]$  representa la concentración del espécimen difusivo en el instante  $t$ , mientras que los coeficientes  $D_x$ ,  $D_y$  y  $D_z$  representan la difusividad del material dentro del polímero en las direcciones  $x$ ,  $y$  y  $z$  respectivamente. Para materiales isotrópicos los coeficientes de difusión se igualan, siendo  $D = D_x = D_y = D_z$ :

$$\frac{\partial[H_2O]}{\partial t} = D \left[ \frac{\partial^2[H_2O]}{\partial x^2} + \frac{\partial^2[H_2O]}{\partial y^2} + \frac{\partial^2[H_2O]}{\partial z^2} \right] = D \cdot \Delta[H_2O] \quad (10-2)$$

Para polímeros alifáticos, este proceso se asume como instantáneo puesto que la duración del proceso hidrolítico es comparativamente mucho más rápido que el proceso degradativo [97], [98].

La ecuación de Fick se utiliza también, bajo ciertas condiciones, para el modelado de la difusión de los subproductos de la degradación u oligómeros en el proceso de difusión [99]. Generalizando la ecuación (10-2) a un material genérico cuya concentración es  $C$ , tenemos la ecuación general de Fick para medios isotrópicos:



$$\frac{\partial C}{\partial t} = D\Delta C$$

A los materiales que rigen su comportamiento degradativo según la anterior ley se les denomina *Fickianos*. La asunción de que el coeficiente de difusión  $D$  es constante, es una simplificación del proceso degradativo, puesto que cuando el material se degrada el espacio liberado por los oligómeros que se difunden al medio acuoso son ocupados por las moléculas de agua, lo que deriva en un incremento del proceso difusivo de los subproductos de la degradación.

En este caso la difusividad se asume como constante, pero en realidad es dependiente de varios factores como la porosidad o la concentración de enlaces de éster o de oligómeros [19].

Por regla general, en la gran mayoría de los polímeros reabsorbibles que se utilizan para implantes ortopédicos, el medio acuoso se infiltra de forma casi instantánea en el material.

Por otra parte, los modelos anteriormente mencionados no contemplan la pérdida de propiedades mecánicas debidas a la erosión superficial, es decir, no hay cambio volumétrico de la pieza durante el proceso de pérdida de propiedades mecánicas.

Por otra parte, estos modelos mencionados permiten estimar la evolución de las propiedades mecánicas cuando el material se encuentra por encima de la temperatura de transición vítrea. Por este motivo, Ding por una parte y Gleadall por otra, desarrollaron sendos modelos moleculares que permite predecir el comportamiento del material polimérico cuando este se encuentra por debajo de la temperatura de transición vítrea. [100], [101].

Asimismo, se han establecido relaciones para la determinación de la evolución de las propiedades mecánicas del material o la tensión límite o de rotura de los polímeros bioabsorbibles, pudiéndose definir mediante una expresión de tipo exponencial:

$$\sigma = \sigma_0 e^{-\lambda[E][H_2O]t} \quad (10-3)$$

Donde  $[E]$  y  $[H_2O]$  representan la concentración de grupos de éster y de agua, respectivamente, en el núcleo del material [102].

Por otra parte, Wang [40] en primer lugar y, posteriormente, Shirazi [41], establecieron un modelo que permite predecir la evolución de las propiedades mecánicas debidas a la erosión en masa del polímero usando un modelo basado en la variación entrópica del material [29], calculando el número de cadenas poliméricas del material a partir de un valor crítico del peso molecular.

Recientemente, Vieira et al. integraron el modelo acoplado de Sevim y Pam para el cálculo de la erosión superficial y en masa [45] con la metodología propuesta por dicho autor para calcular las propiedades mecánicas debidas a la erosión superficial [103].

En esta sección se presenta una implementación del cálculo de propiedades mecánicas por medio de la utilización del criterio del módulo de elasticidad global. Y luego se utilizará el modelo de muelle entrópico propuesto por Ward et al. [29] para el cálculo de la evolución de las propiedades mecánicas debidas al proceso hidrolítico.

## 10.2. METODOLOGÍA ASOCIADA AL CÁLCULO DE LAS PROPIEDADES MECÁNICAS DE LA PIEZA USANDO EL CRITERIO DE MÓDULO DE ELASTICIDAD GLOBAL.

El cálculo de propiedades mecánicas se hará considerando en los ensayos de compresión que el material se comporta de forma elástica. El material empleado para las simulaciones es el PCL, cuyo módulo de elasticidad es de 350 MPa y tiene un coeficiente Poisson de 0.46. La metodología se detalla paso a paso en el siguiente apartado.

Se empleará la geometría degradada de la pieza y, en diferentes etapas de degradación, se realizará una carga equivalente a un desplazamiento unitario en la superficie situada a la entrada del flujo, y una restricción de movimiento vertical en la cara situada a la salida del mismo. Se evaluarán las resultantes de la fuerza de reacción en distintos instantes de tiempo. El valor derivado (fuerza total entre el área) es el módulo de elasticidad (módulo de elasticidad equivalente, puesto que se asume sección completamente maciza).

Se disponen de diferentes herramientas utilizadas para la realización de los cálculos mecánicos. Algunas vienen integradas en programas de diseño CAD como pueden ser SolidWorks, Catia o Inventor, aunque las modelos que presentan son válidas para ciertos tipos de materiales. También presentan ciertas limitaciones a la hora de trabajar con geometrías más complejas que requieran de mallados más finos. Para estudios más complejos, se precisa el uso de softwares de simulación multifísico, tales como ANSYS, Abaqus y Comsol. También existen versiones de código abierto que tienen implementados modelos de elasticidad lineal y no lineal, como SALOME-MECA, Code Aster [104] y FreeFem++. Dentro de la literatura relacionada con el estudio de las propiedades mecánicas de scaffolds poliméricos bioabsorbibles [102], [105], para este estudio se recurre al uso del ABAQUS por su mayor fiabilidad a la hora de realizar cálculos para diferentes tipos de geometrías.

El trabajo a realizar consistirá en importar la estructura del scaffold degradada en la simulación previa, y realizar sucesivos ensayos de compresión. El procedimiento a seguir es secuencial, y se esquematiza en la Figura 10-1:

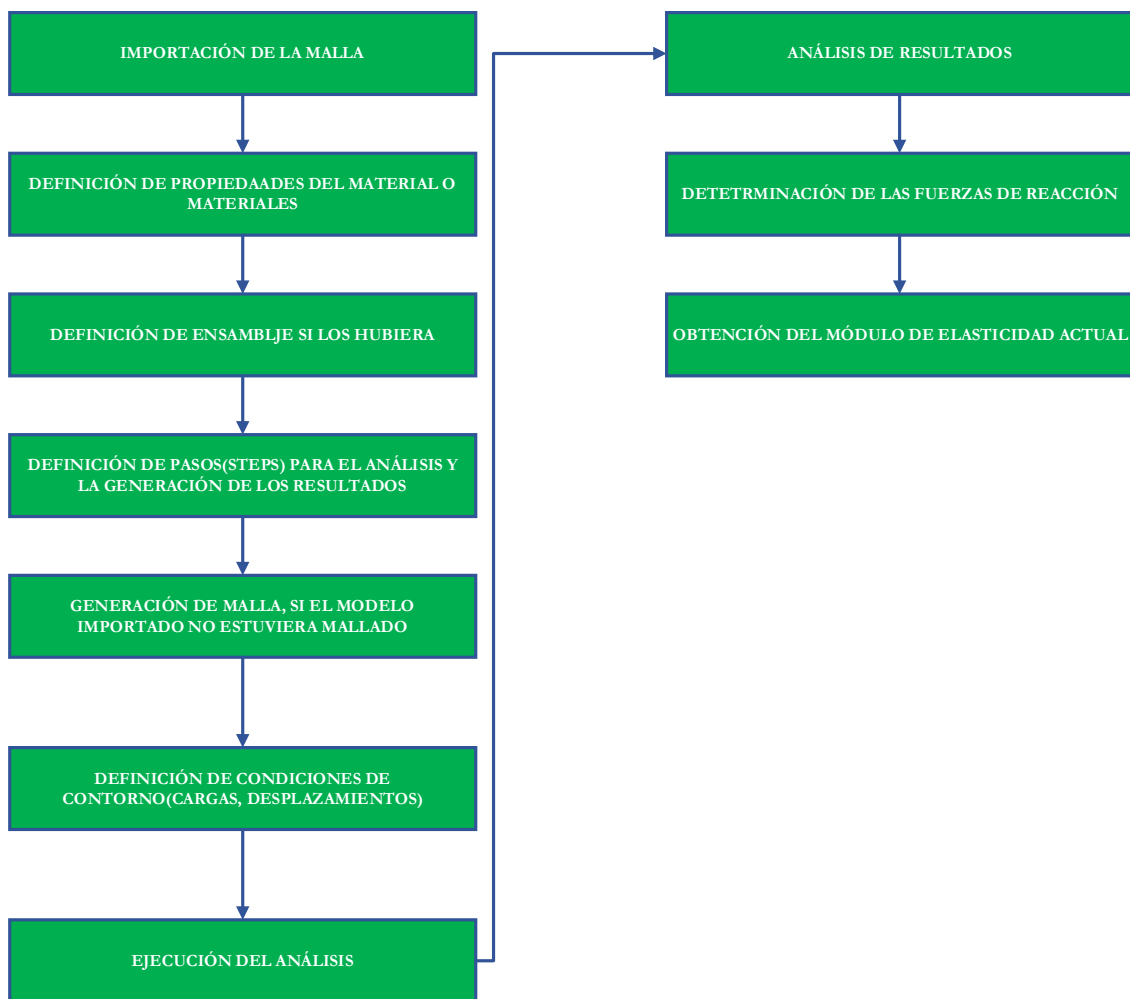


Figura 10-1. Esquema de proceso propuesto para cálculo de propiedades mecánicas globales.

En el Anexo 15.1 se describe paso a paso cada una de las etapas descritas en el diagrama anterior mediante el uso de las herramientas GMSH y ABAQUS para el discretizado y simulación de las propiedades mecánicas.

El cálculo del módulo de elasticidad se hace dividiendo el valor obtenido de la fuerza de reacción de la simulación entre la superficie de aplicación de la carga, según la siguiente expresión:

$$E_{global} = \frac{F_{reacción} / A_{equivalente}}{desplazamiento / Longitud} \quad (10-4)$$

Donde:

- $F_{reacción}$  es la carga aplicada en la base del dispositivo para contrarrestar el desplazamiento aplicado.
- $A_{equivalente}$  es la superficie sobre la que se aplica dicha carga de reacción
- *desplazamiento* es, como su propio nombre indica es la cantidad que se impone como desplazamiento a los nodos de la cara superior.
- *Longitud* es la distancia máxima medida en la pieza a ensayar en la dirección de aplicación del desplazamiento.

### 10.2.1. APLICACIÓN DEL CRITERIO DE MODULO DE ELASTICIDAD GLOBAL SOBRE UNA PIEZA

Para este cálculo se utilizará el modelo de scaffold prismático que se ha empleado en el epígrafe 8.3, utilizando los 4 modelos del dispositivo en las diferentes etapas de degradación. En la Tabla 10-1 se pueden ver datos relativos a las dimensiones de la pieza sobre el que se han hecho cálculos de erosión estática por Monte Carlo, así como el desplazamiento que se le aplica a la parte superior del dispositivo durante el ensayo mecánico, que es de 0,4 mm (Figura 10-2).

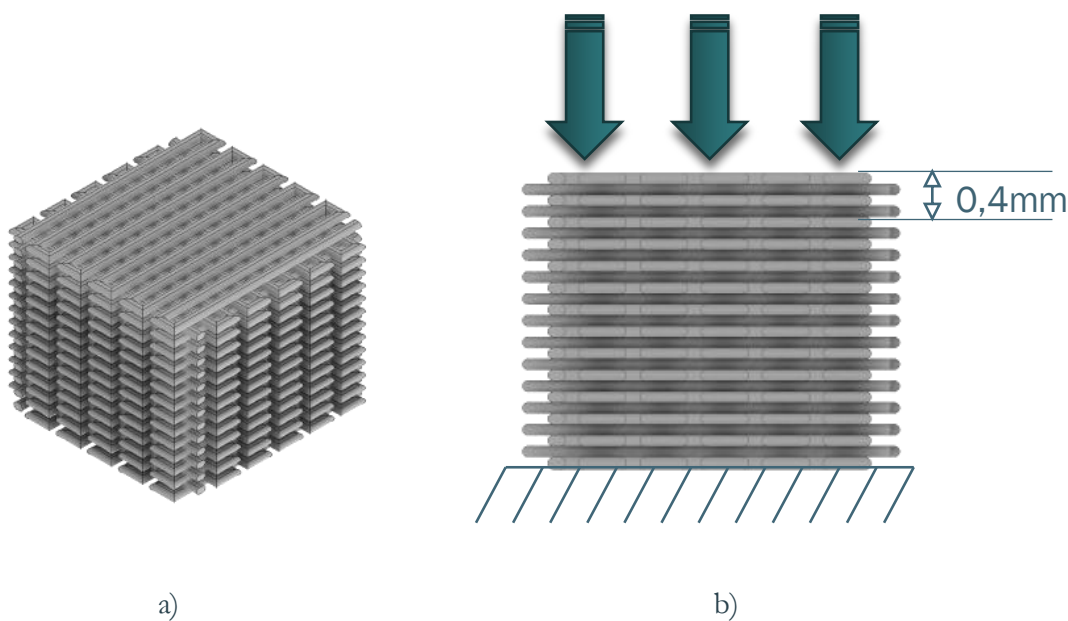


Figura 10-2. Representación 3d del modelo de scaffold (a)) y vista frontal con las cargas y desplazamientos aplicados(b))

Dimensiones nominales de las piezas	10x10x8 mm x mm x mm
Altura real de la pieza=	8,13858 mm
Área equivalente=	105,577 mm <sup>2</sup>
Desplazamiento aplicado=	0,4 mm

Tabla 10-1. Datos relativos a la pieza utilizada para las simulaciones, así como el desplazamiento aplicado al mismo en el ensayo mecánico.

A la hora de realizar el estudio es necesario indicar el valor del módulo de elasticidad intrínseco del material como parámetro de entrada en el software. Para ello, se tomará, para cada caso, los valores de pérdida de módulo de elasticidad de un filamento en diferentes los instantes de medición, datos obtenidos de un experimento realizado por el grupo de Fabricación Integrada y Avanzada [dentro del proyecto BioAM](#). Los datos aparecen representados en la Tabla 10-2:

Módulo de elasticidad Filamento (MPa)
306
304
265
270

Tabla 10-2. Valores de referencia de módulo de elasticidad intrínseco del filamento en diferentes etapas de degradación.

En la última columna de la Tabla 10-3 se pueden ver los valores obtenidos de módulo de elasticidad de la pieza en diferentes instantes de tiempo, a partir de los valores de fuerza de reacción (tercera columna). Se puede ver que, efectivamente se produce un descenso de los valores de módulo de elasticidad global en los diferentes instantes de tiempo. Atendiendo a la comparativa con los resultados experimentales se observa que dichos resultados obtenidos son del orden de 20 MPa menores en términos absolutos que los valores obtenidos del ensayo mecánico. Una de las razones que se pueden inferir de esta discrepancia es la no consideración del cambio en las propiedades intrínsecas del material tales como el peso molecular, habida cuenta de la pérdida de longitud de las cadenas poliméricas en el proceso de degradación. También el tomar como valor intrínseco del módulo de elasticidad el correspondiente al filamento en diferentes instantes de tiempo permite obtener un ajuste similar, aunque con un desfase significativo entre los valores simulados y los obtenidos de la experimentación.

Tiempo(h)	Porcentaje de masa disuelto	Fuerza de Reacción (N)	E global simulado (MPa)	Módulo de Elasticidad global Pieza experimental (MPa)*	Módulo de elasticidad Filamento (MPa)*	Desviación resultados (MPa)
0	0	241,003	46,0095	63,27	306	17,2605
24	5,61	228,513	43,625	69,82	304	26,195
48	6,13	227,185	43,371	68,84	265	25,649
96	11,99	211,817	40,438	59,61	270	19,172

Tabla 10-3. Tabla de resultados de módulo de elasticidad de la pieza y comparativa con valores de referencia.

En la Tabla 10-4 se representa una comparativa de la variación porcentual entre módulos de elasticidad en etapas en la etapa  $i$  y la etapa  $i + 1$  tomadas en la simulación y en el ensayo experimental, como aparece representado en la expresión (10-5):

$$\Delta E_{global}(\%) = \frac{E_{global}^i - E_{global}^{i+1}}{E_{global}^i} \cdot 100 \quad (10-5)$$

E global simulado (MPa)*	Variación porcentual simulada	Módulo de Elasticidad global Pieza experimental (MPa)*	Variación porcentual de referencia (%)	Desfase
46,0095	0	63,27	0	0
43,625	5,183	69,82	10,352	5,169
43,371	5,735	68,84	8,8035	3,0685
40,438	12,109	59,61	5,7847	-6,3243

Tabla 10-4. Variación porcentual relativa del módulo de elasticidad global respecto al estado inicial de la pieza

A la luz de los resultados, se observa que hay un desfase entre las variaciones porcentuales que van entre 3 y 6,5 en valor absoluto. En relación con el modelo simulado, la variación reflejada es de 5,183 en etapas 2 y 3, correspondientes a las mediciones en 24 y 48 h, respectivamente, son bastante similares. Sin embargo, se observa una discrepancia con respecto a los valores porcentuales obtenidos a partir de las mediciones durante el ensayo mecánico son significativamente mayores. En la última etapa, el descenso en las propiedades

mecánicas en el modelo simulado se duplica, mientras que en el ensayo experimental se reduce algo más de 3 puntos. De aquí se puede observar que no hay tampoco una correspondencia clara entre ambas mediciones.

Por este motivo, se plantea en el siguiente epígrafe, una perspectiva que está basada en un acoplamiento del modelo hidrolítico visto en el epígrafe 9 con un modelo de rotura de cadenas propuesto por Ward y Sweeney.

### 10.3. SIMULACIÓN DE LA ROTURA DE CADENAS POLIMÉRICAS Y CÁLCULO PROMEDIO DE PROPIEDADES MECÁNICAS

En el siguiente caso se aplicará la rotura de cadenas poliméricas bajo la acción de las moléculas de agua y su efecto en el cambio de propiedades mecánicas.

Inicialmente se parte de una cadena única compuesta de  $n$  moléculas de ácido láctico de peso molecular igual a 72 g/mol.

#### 10.3.1. DISTRIBUCIÓN INICIAL DE PESO MOLECULAR

Inicialmente se considera una cadena de PLA de  $3,55 \cdot 10^7$  moléculas y se aplican 9999 roturas aleatorias para la obtención de 10000 cadenas. Se fija como valor umbral de 12 moléculas para aquellas cadenas que son absorbidas por el medio, por lo que se contabilizan solo aquellas que queden por encima de dicho valor.

Al haber cadenas con diferentes pesos moleculares la distribución del peso molecular no es uniforme. Para obtener valores representativos del conjunto se definen los pesos moleculares promedio en número y en peso (ecuaciones (10-6) y (10-7) respectivamente).

Los pesos moleculares promedio numérico y en peso son:

$$M_n = \frac{N_{TOTAL} M_0}{n} \quad (10-6)$$

$$M_w = \sum_{i=1}^{i=N} \frac{N_i^2}{N_{TOTAL}} M_0 \quad (10-7)$$

Donde  $N_{TOTAL}$  es la suma de unidades monoméricas de la muestra,  $M_0$  el peso molar de una unidad monomérica y  $n$  el número total de cadenas.



El **peso promedio en número** es el valor peso molecular promedio de una cadena. El **peso promedio en peso** es la suma para todas las cadenas de la masa molar  $M_0$  multiplicada por la correspondiente longitud de la cadena  $N_i$ , y la probabilidad de que una unidad monomérica pertenezca a dicha cadena,  $\frac{N_i}{N_{TOTAL}}$ .

**Paso 1. Generar una distribución inicial de peso molecular.**

Se dispone de una cadena de  $N=3.55 \cdot 10^7$  monómeros y se producen 10000 roturas aleatorias de cadenas para la obtención de las 10000 cadenas poliméricas.

La distribución del peso molecular en función de la longitud de cadena se expresa en las siguientes gráficas:

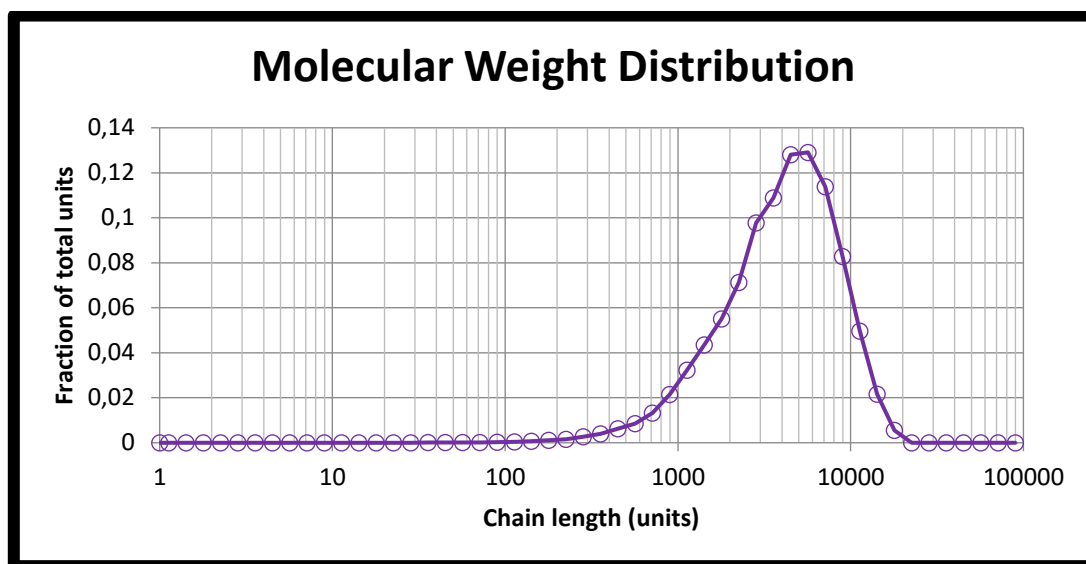


Figura 10-3. Distribución de peso molecular

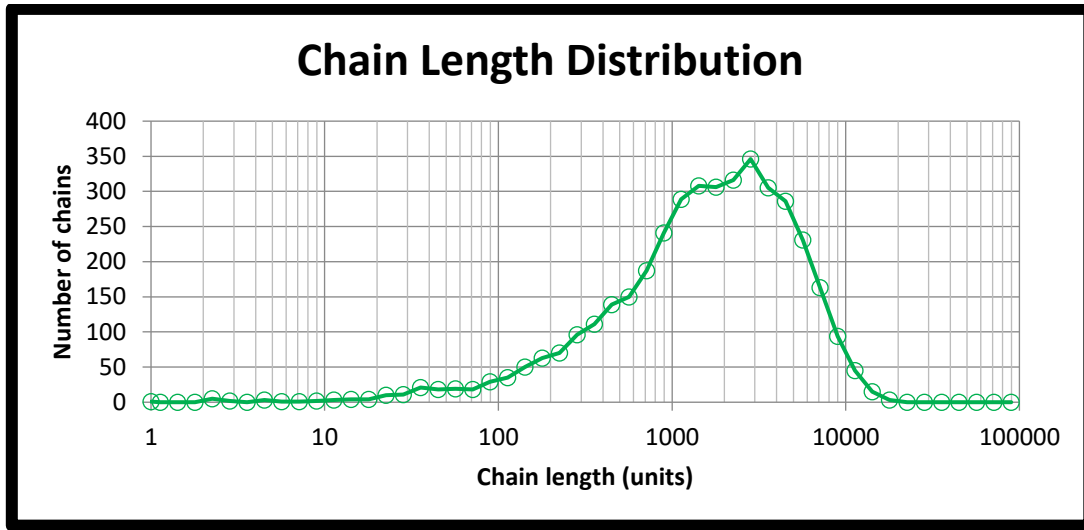


Figura 10-4. Distribución de longitud de cadena.

### 10.3.2. SIMULACIÓN DEL PROCESO DE DEGRADACIÓN DE LAS PROPIEDADES MECÁNICAS:

La teoría entrópica [29] establece una relación entre el módulo de elasticidad y el número de cadenas presentes en un polímero (3):

$$E = 3KNT \quad (10-8)$$

Donde N es el número de cadenas por unidad de volumen, K es la constante de Boltzmann y T la temperatura absoluta en grados Kelvin.

Cada cadena original y las cadenas resultantes de aplicar una escisión a la anterior se contabilizan como una sola cadena. Si la subcadena de longitud mayor tiene un peso molecular por debajo del valor crítico,  $M_n^{crit}$  definido previamente, se resta al número total de cadenas.

$$N_{chains}^i = N_{chains}^{i-1} - 1 \quad (10-9)$$

Siendo  $N_{chains}^i$  el número total de cadenas en la etapa i de degradación y  $N_{chains}^{i-1}$  el correspondiente a la etapa anterior. En este caso, se ha definido un peso molecular límite de 12 unidades [106].

### 10.3.3. ESCISIÓN MOLECULAR

En relación con la rotura de cadenas, existen dos tipos (Figura 10-5): la **escisión aleatoria**, que se produce en una posición aleatoria dentro de la cadena, derivando en la formación de dos subcadenas y la **escisión en los extremos**, en donde la rotura se produce en el enlace inicial o final de la cadena. Este último provoca la formación de monómeros.

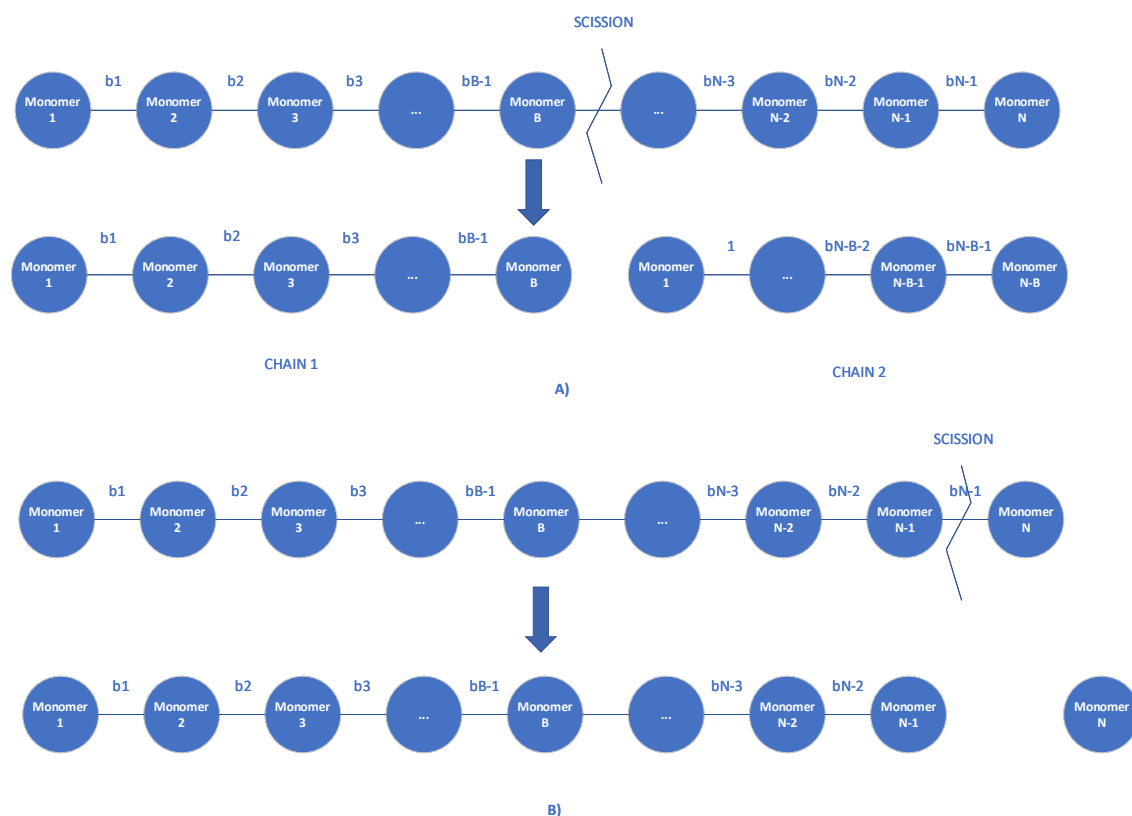


Figura 10-5. Tipos de escisión de una cadena polimérica: A) Escisión aleatoria; B) Escisión en los extremos.

Normalmente se establece una relación entre el número de veces que se produce una escisión aleatoria por cada vez que se realiza una escisión en los extremos. Esta relación,  $R_{scissions}$ , suele tomar los valores 3:1, 1:1 y 1:50 [40], [41].

Para decidir el tipo de escisión que se produce se utiliza el parámetro  $R_{scissions}$  que se compara con un número generado aleatoriamente. Si el valor obtenido queda por debajo de  $R_{scissions}$ , entonces se aplica una escisión aleatoria. En caso contrario, se aplicará una rotura en los extremos, que deriva en la formación de un monómero.

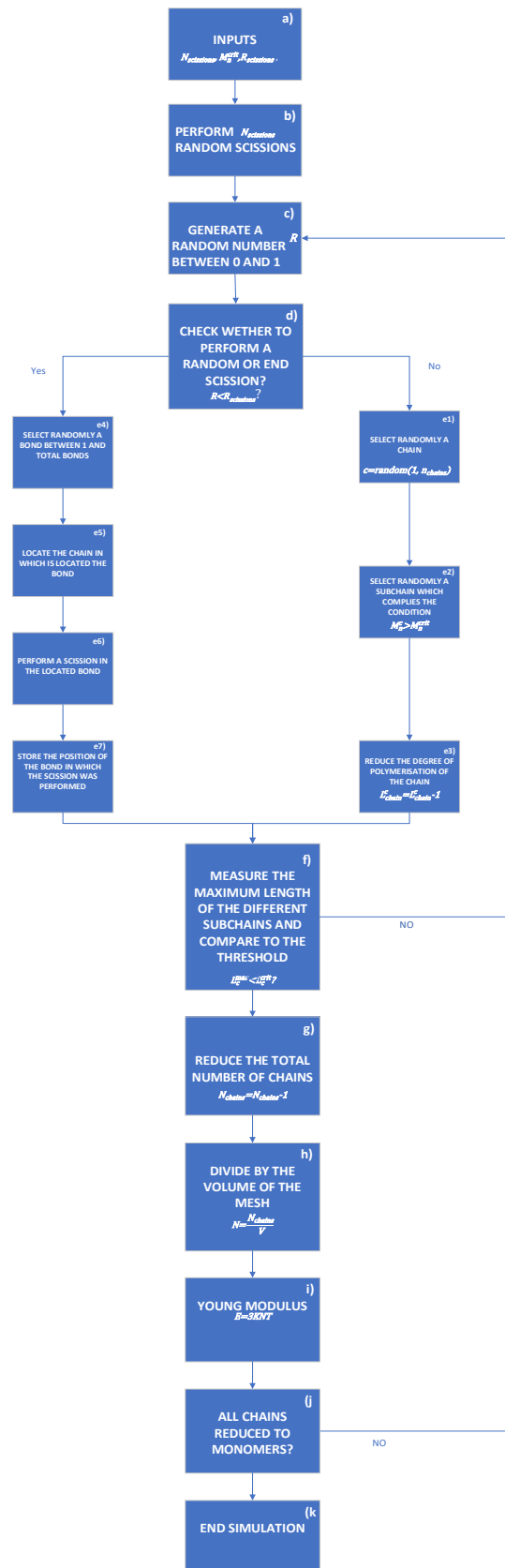


Figura 10-6. Diagrama del proceso de obtención del módulo de elasticidad con la rotura molecular.

La secuencia detallada de obtención del módulo de elasticidad se explica en la Figura 10-6. Inicialmente (paso a)) se deben especificar como parámetros de entrada el valor crítico de peso molecular,  $M_n^{crit}$ , el ratio entre escisiones aleatorias y los extremos  $R_{scissions}$  y el numero de escisiones iniciales  $N_{scissions}$ , para la obtención del número de cadenas iniciales. En el paso b), se aplican 9999 escisiones aleatorias a la cadena inicial.

En la segunda parte del proceso, se decide el tipo de rotura que se va a aplicar. Para ello, se genera un número aleatorio  $r$  entre 0 y 1 (paso c). Este valor se recalcula como un valor  $c$ , que depende del anterior:

$$c = \frac{r}{1 - r} \quad (10-10)$$

El valor de  $c$  se compara con la relación  $R_{scissions}$ (paso d)). Si este queda por debajo, entonces se aplicaría una escisión aleatoria. En caso contrario la escisión se produciría en los extremos.

#### Escisión aleatoria.

Si la escisión se produce de manera aleatoria, se selecciona al azar un enlace sobre el cual aplicar dicha rotura (paso e4)).

$$bond = random(1, n - 1) \quad (10-11)$$

Acto seguido se localiza la cadena en la cual se encuentra dicho enlace (paso e5)) y se aplica la rotura(e6). Las longitudes de las cadenas resultantes se almacenan en las sucesivas filas del vector  $chain_{array}$

$$chain_{array}(i, :) = (l_{i1}, l_{i2}, \dots, l_{in}) \quad (10-12)$$

#### 10.3.4. ESCISIÓN EN LOS EXTREMOS.

Para aplicar una escisión en los extremos, se selecciona aleatoriamente una de las cadenas que cumpla la condición de no ser un monómero (e1). Dentro de la cadena, se selecciona arbitrariamente aquella subcadena cuya longitud sea mayor al valor umbral (e2) y se reduce su grado de polimerización en una unidad(e3).

$$l_{ij}^k = l_{ij}^{k-1} - 1 \quad (10-13)$$

Una vez aplicada la rotura se localiza la subcadena de mayor longitud y se compara con el valor umbral (f). Si la longitud queda por debajo de  $M_n^{crit}$ , entonces se reduce el número de cadenas (g). Si por el contrario la condición no se cumple se volvería al paso (c).

$$n_{chains}^i = n_{chains}^{i-1} - 1. \quad (10-14)$$

### 10.3.5. ACTUALIZACIÓN DE PROPIEDADES MECÁNICAS

Una vez reducido el número de cadenas, se divide entre el volumen de la pieza (h):

$$N = \frac{n_{chains}^i}{V} \quad (10-15)$$

Este valor se utiliza para determinar el módulo de Young promedio de la pieza a través de la expresión (10-8):

$$E = 3KNT$$

Por último, se comprueba si todas las cadenas se han convertido en monómeros (j). En caso afirmativo, se da por finalizada la simulación (k). Si no, se volvería al paso c).

Para establecer la relación entre el peso molecular y el módulo de elasticidad, calculamos los valores obtenidos de ambos parámetros en diferentes instantes de tiempo, obteniendo una tendencia del tipo:

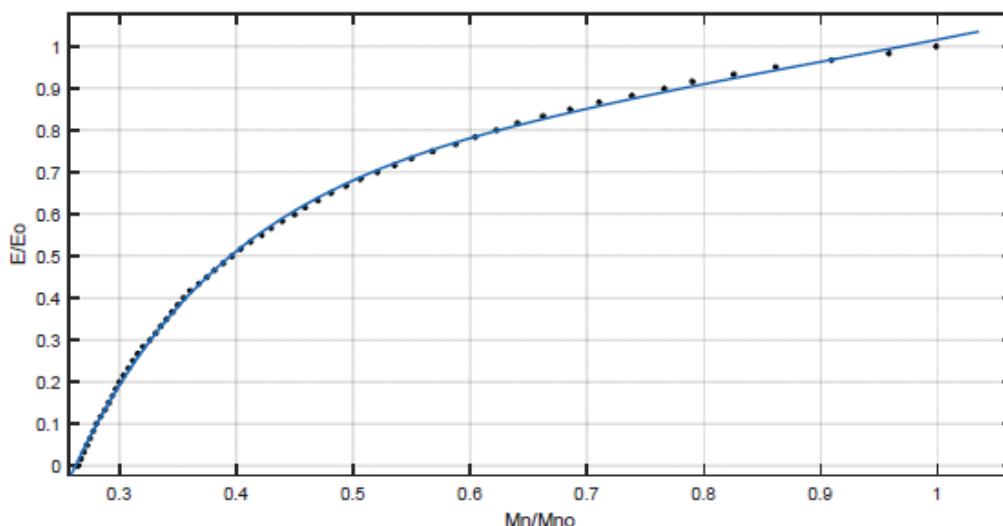


Figura 10-7. Representación gráfica de la relación entre los valores estandarizados Módulo de Elasticidad vs Peso Molecular Normalizado en diferentes instantes del proceso degradativo.

En la Figura 10-7 se observa la representación de los puntos correspondientes a los pares  $(M_n^*, E^*)$  obtenidos en diferentes instantes del proceso de rotura de cadenas. Para establecer una relación entre ambas variables se recurre a un ajuste regresivo por medio de la herramienta *Curve Fitting Toolbox* de MATLAB [107]. En este caso, la expresión que más se aproxima a la serie de puntos es la suma de dos expresiones de tipo exponencial del tipo:

$$E = ae^{bM_n^*} + ce^{dM_n^*} \quad (10-16)$$

Para lo cual se requiere de 4 parámetros de ajuste. En este caso la combinación de valores que proporciona un mejor ajuste es  $a=0,6131$ ,  $b=0,5077$ ,  $c=-5,408$  y  $d=7,796$ , para una raíz del error cuadrático medio o RMSE de 0,006717

Teniendo en cuenta que  $\overline{C_e} = M_n^*$  [30] podemos utilizar esta relación en la implementación del modelo de Reacción Difusión (9.1.2) para obtener la distribución del módulo de Young en diferentes instantes de tiempo. En la Figura 10-8 se observa la distribución del módulo de Young normalizado en diferentes etapas del proceso de degradación.



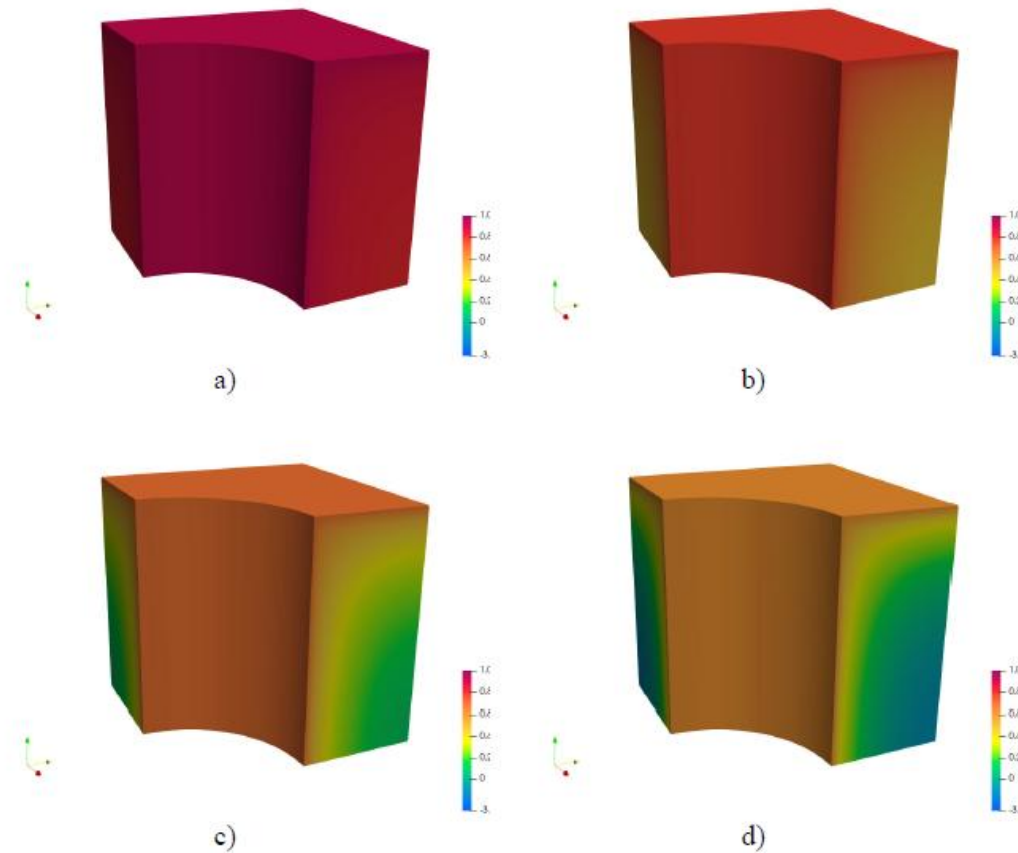


Figura 10-8. Distribución del módulo de Young en la celda unitaria en las diferentes etapas de degradación hidrolítica.

A la luz de los datos los resultados obtenidos nos dicen que la pérdida mayor del módulo de elasticidad se produce principalmente en las regiones internas de la matriz polimérica. Esta pérdida localizada de rigidez se explica por la dependencia exponencial, del módulo de elasticidad con el peso molecular. Dado que el peso molecular promedio  $\overline{M}_n$  es directamente proporcional a la concentración de enlaces éster  $C_e$  [30], [96] y atendiendo al mapa de peso molecular obtenido por Wang en el modelo hidrolítico [1], esto deriva en que la distribución espacial del módulo de Young refleja fielmente la evolución de  $C_e$  en el dominio polimérico.

# 11. ACOMPLAMIENTO DE LOS MODELOS DE DEGRADACIÓN Y EROSIÓN.

## 11.1. INTRODUCCIÓN

Se ha demostrado que el material no solo manifiesta un proceso únicamente degradativo o erosivo, sino que normalmente es una combinación de ambos efectos. También se necesita considerar las condiciones a las que está sometida la pieza (si el proceso es in-vivo existe una predominancia del efecto degradativo frente al erosivo o la erosión se produce en masa). Por el contrario, bajo condiciones in vitro existe una mayor presencia del proceso erosivo superficial. Por tanto, existe la necesidad de realizar un enfoque más holístico del proceso incluyendo todos aquellos fenómenos que describen mejor el comportamiento del material. Varios autores intentaron representar el comportamiento combinado del efecto degradativo y erosivo del material considerando diferentes perspectivas a nivel numérico. Sevim y Pam modelaron un modelo combinado fenomenológico-probabilístico que permite estimar el comportamiento del material bajo condiciones dinámicas. Niu y Pan [108] por su parte desarrollaron un modelo fenomenológico en elementos finitos para la determinación de la erosión en masa y la degradación hidrolítica con una aplicación en scaffold vasculares bio-reabsorbibles. Sin embargo, de entre los modelos presentados en la literatura para modelar la erosión superficial mediante el uso de modelos fenomenológicos y/o estocásticos normalmente se tiende a aplicarlos sobre geometrías sencillas. En este respecto, Shockley et al han considerado un enfoque predominantemente fenomenológico para el estudio combinado de los fenómenos erosivos y degradativos del material para diferentes configuraciones geométricas, asumiendo que la difusión del agua no se produce de forma instantánea [59]. El enfoque utilizado de conjuntos de nivel [109] le otorga una mayor estabilidad a este modelo permitiendo extrapolar este modelo a diferentes geometrías de mayor complejidad. Sin embargo, éste último modelo se centra en el comportamiento del material bajo condiciones estáticas, obviando la influencia del movimiento del medio circundante bajo condiciones dinámicas.

Aquí se presentan diferentes metodologías o propuestas, en las que se ha valoran su nivel de convergencia para luego determinar cuál de las opciones ensayadas presenta una mayor estabilidad numérica a la hora de representar la totalidad o la mayor parte del proceso combinado de la erosión superficial y la erosión hidrolítica, con el fin de obtener un modelo lo suficientemente robusto para determinar la evolución de las propiedades intrínsecas y de los cambios en la configuración geométrica de la pieza para diferentes instantes de tiempo.

Se han modelado diferentes propuestas numéricas que intentan describir el comportamiento del material incluyendo el modelo por medio del modelo propuesto por Wang [1] y el modelo erosivo desarrollado por Abdelfatah et al [46] . Se valorarán la idoneidad de cada propuesta y se discutirá su estabilidad en términos de convergencia. Estos modelos se aplicarán

inicialmente a geometrías básicas y luego se replicará su comportamiento sobre modelos de mayor complejidad, una vez se validen sobre los modelos base.

## 11.2. PRIMERA PROPUESTA DE INTEGRACIÓN DEL MODELO DEGRADATIVO CON EL MODELO HIDROLÍTICO.

La primera propuesta consiste en elaborar un modelo integrado que inicialmente considera una degradación hidrolítica del material considerando un umbral para el inicio del proceso erosivo de acuerdo al modelo propuesto por Heljak [30] que relaciona la pérdida de masa con la concentración máxima de subproductos acumulados en la matriz polimérica.

$$\frac{\partial m(\bar{t}, \bar{x})}{\partial t} = -k_m (C_m^{max})^2 \quad (11-1)$$

Siendo:

$-C_m^{max}$  la concentración normalizada de monómeros resultantes del proceso de degradación hidrolítica.

$-k_m$  es un coeficiente que se utilizará como parámetro de ajuste para calibrar el modelo.

Considerando  $m_0$  como la masa inicial del dispositivo y  $m$  como la masa en el instante  $\bar{t}$  y en el espacio  $\bar{x}$ . Para la obtención de la masa aplicamos separación de variables e integramos la expresión resultante para obtener la masa de referencia:

$$\int_{m_0}^m dm = -k_m (C_m^{max})^2 \int_0^{\bar{t}} dt$$

Teniendo la siguiente expresión de la masa en función del tiempo.

$$m = m_0 - k_m (C_m^{max})^2 t \quad (11-2)$$

Estos valores de masa se usarán de referencia para estimar la pérdida de masa del material debido a la erosión.

En la Figura 11-1 se representa el esquema de trabajo del primer modelo propuesto. Como parámetros de entrada se toman los parámetros del modelo hidrolítico de Wang  $k_1, k_2, C_{e0}, C_{m0}, D_0, \alpha$ . La implementación usada aquí corresponde al modelo matricial,

como se ha visto en la sección 9.1.2. La resolución iterativa de estas ecuaciones se realizará hasta que el promedio sobre el dominio de la concentración de monómeros sea menor que el correspondiente a la iteración anterior (momento en el cual se produce la liberación de los mismos al medio, y se activa el proceso erosivo):

$$\overline{C}_m^i < \overline{C}_m^{i-1} \quad (11-3)$$

Esto se traduce en que el valor de  $\overline{C}_m$  en la iteración previa se corresponde con el máximo:

$$\overline{C}_m^i < \overline{C}_m^{max} \quad (11-4)$$

Una vez ha sido verificada la condición anterior, se actualiza el valor de masa debida a la liberación de monómeros a través de la ecuación (11-2).

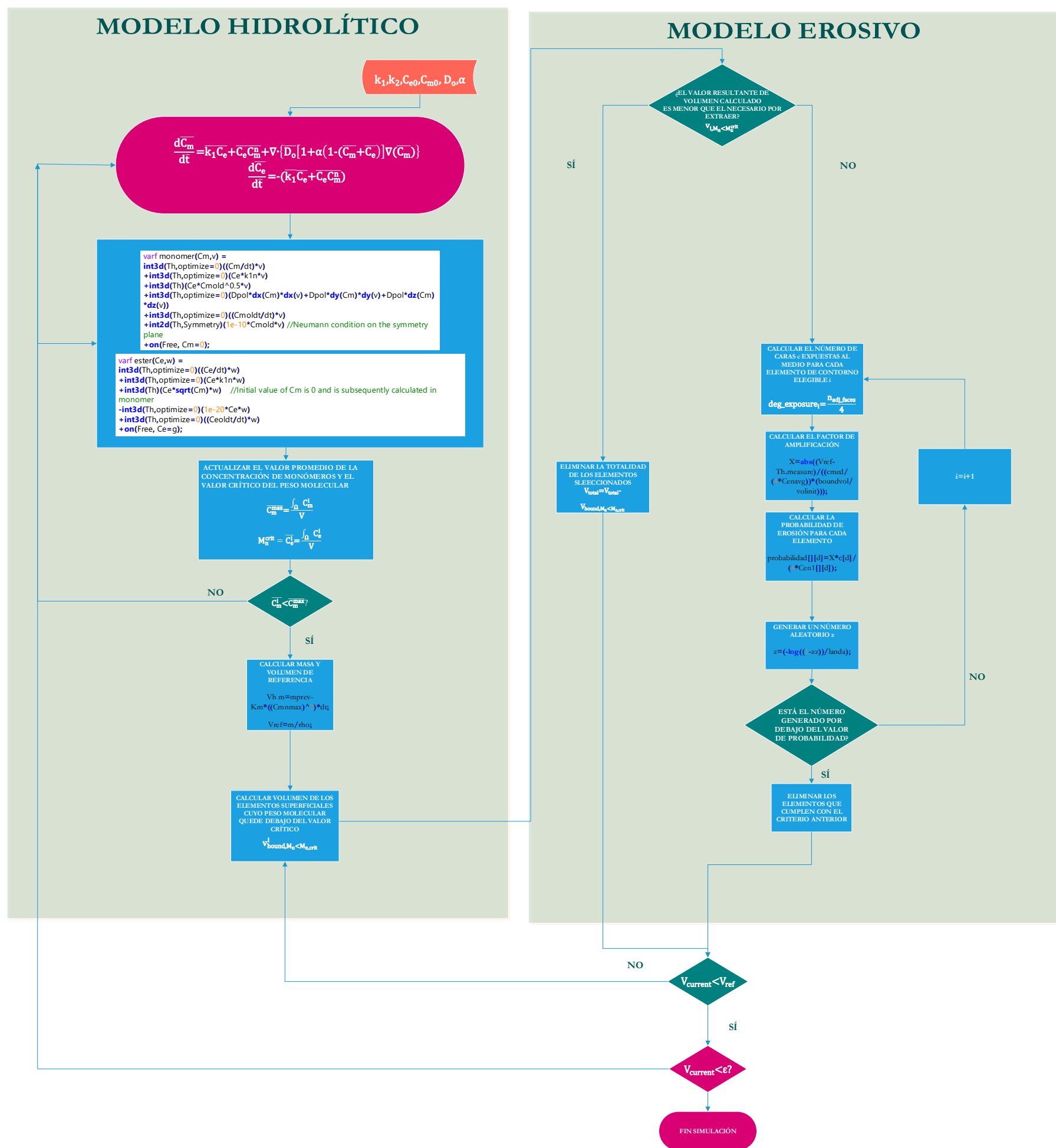


Figura 11-1. Diagrama de flujo correspondiente a la implementación numérica del primer modelo propuesto modelo integrado de degradación hidrolítica y erosión.

Una vez alcanzado el criterio de erosión, se aplica el Método de Monte Carlo usando una ecuación modificada en el que se incorpora el termino de peso molecular, que, de acuerdo a Han [96], su valor normalizado coincide con el valor normalizado de la concentración de enlaces ésteres:

$$\overline{M}_n = \overline{C}_e \quad (11-5)$$

La erosión de Monte Carlo se basa en la expresión (8-7) incluyendo el término de peso molecular promedio en el denominador de la siguiente forma.

$$P_i = X \frac{c}{4\overline{M}_n} \quad (11-6)$$

Introduciendo dicho término en el denominador logramos que la probabilidad se incremente para valores de peso molecular por debajo del crítico.

Asimismo, el factor de amplificación, como se ha visto anteriormente (véase sección 8.2) se usa para mejorar, a efectos computacionales el tiempo de cálculo necesario para obtener el valor de referencia de volumen calculado previamente. Este valor, dependerá del valor que tomen tanto el número de caras que cada elemento tiene en contacto con el medio y el valor normalizado de peso molecular para cada tetraedro. La incorporación de la masa (volumen) de referencia, calculado en el paso anterior, y el valor promedio del meso molecular  $\overline{M}_{n\text{average}}$ , obtenido a partir de las expresiones hidrolíticas, en el denominador de la expresión, dividiendo al término de caras promedio  $c_{\text{average}}$ , nos da una expresión del tipo:

$$X = \frac{((\overline{V}_0 - \overline{V}_{ref}) - \overline{V}_{loss})}{\overline{V}_{boundary} \frac{c_{\text{average}}}{4\overline{M}_{n\text{average}}}} \quad (11-7)$$

Donde:

- $\overline{V}_0$  es el volumen inicial normalizado de la pieza.
- $\overline{V}_{loss}$  es el volumen a extraer por la pieza.
- $c_{\text{average}}$  hace referencia al número promedio de caras que tiene cada elemento.
- $\overline{V}_{boundary}$  es el volumen superficial de los elementos de la muestra. El volumen máximo se obtiene cuando la relación entre este y el volumen total es 1.
- $\overline{M}_{n\text{average}}$  es el valor promedio del peso molecular normalizado.

El criterio establecido para erosionar la pieza es que el numero aleatorio generado quede por debajo del valor de probabilidad definido.

El modelo se inicializa con los parámetros de entrada del modelo hidrolítico de Wang en el que se establecen las condiciones necesarias para la determinación de las condiciones de contorno.

```
real k1=0.003;  
real k2=0.002;  
real n=0.5;  
real D=1.41e-14;  
real l=0.5;  
real Free=1;  
real Symmetry=2;  
real k1n=0.1;  
real D0n=1;  
real alpha=4.5;  
real Cavg;  
real Ceavg;
```

La implementación de las ecuaciones de degradación en formato matricial utilizando el comando `varf` tal como se ha visto en el epígrafe 9.1.2:

```
varf monomer(Cm,v) =  
int3d(Th,optimize=0)((Cm/dt)*v)  
+int3d(Th,optimize=0)(Ce*k1n*v)  
+int3d(Th)(Ce*(Cmold^0.5)*v)  
+int3d(Th,optimize=0)(Dpol*dx(Cm)*dx(v)+Dpol*dy(Cm)*dy(v)+Dpol*dz(Cm)*dz(v))  
+int3d(Th,optimize=0)((Cmoldt/dt)*v)  
+int2d(Th,Symmetry)(1e-10*Cmold*v) //Neumann condition on the symmetry plane  
+on(Free, Cm=0);
```

```
varf ester(Ce,w) =
  int3d(Th,optimize=0)((Ce/dt)*w)
+int3d(Th,optimize=0)(Ce*k1n*w)
+int3d(Th)(Ce*(Cm^0.5)*w)    //Initial value of Cm is 0 and is subsequently
calculated in monomer
-int3d(Th,optimize=0)(1e-20*Ce*w)
+int3d(Th,optimize=0)((Ceoldt/dt)*w)
+on(Free, Ce=g);
```

Después se procede a su resolución en FreeFem++ utilizando una notación matricial:

```
matrix Am = monomer(Vh, Vh);
real[int] Bm = monomer(0, Vh);
```

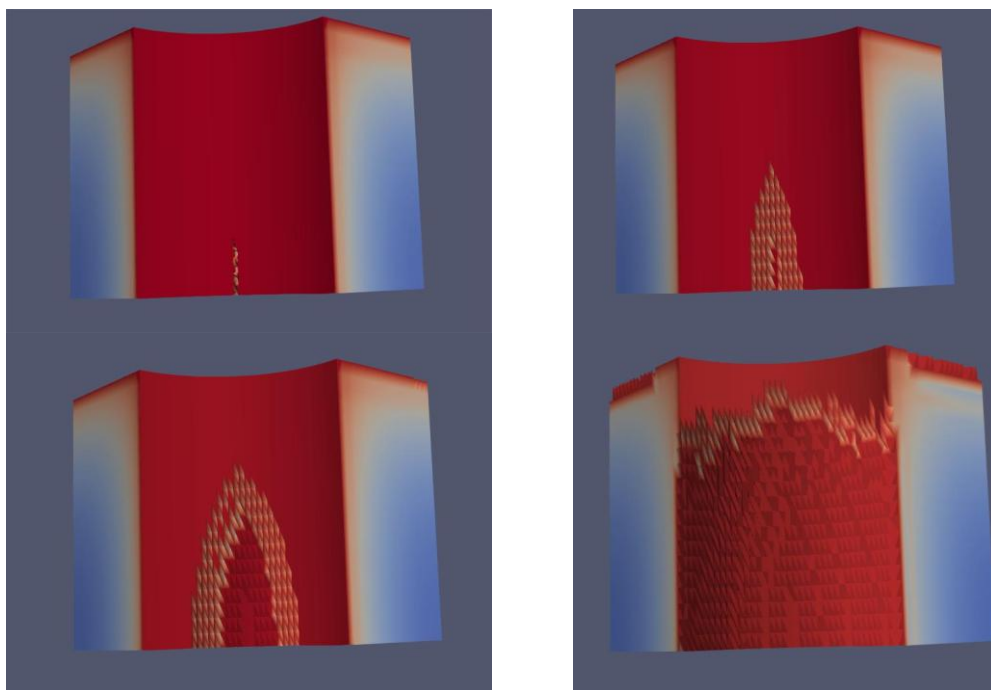


Figura 11-2. Distribución del peso molecular normalizado de la celda unitaria en diferentes etapas de degradación.



Los resultados obtenidos muestran una erosión uniforme y progresiva del material, en aquellas regiones en las que el peso molecular sobrepasa el umbral preestablecido tras un número determinado de iteraciones, logrando obtener una evolución bastante razonable del frente erosivo. Sin embargo, esta tendencia se detiene abruptamente al cabo de unas iteraciones debido a la imposibilidad del algoritmo de aplicar las integrales de contorno sobre una superficie muy irregular. Una de las causas que se aducen en un inicio es de una discretización que no es lo suficientemente refinada, sobre todo en aquellas regiones potenciales o susceptibles de ser erosionadas de acuerdo con el criterio de erosión establecido.

Por este motivo, se plantea una evolución de la metodología propuesta introduciendo una etapa de refinamiento local utilizando el criterio adoptado para la erosión de los elementos de la matriz polimérica.

### **11.3. SEGUNDA PROPUESTA DE MODELO INTEGRADO: APLICANDO REFINAMIENTO LOCAL A LA MALLA.**

Para intentar solventar los problemas que han surgido en el modelo computacional anterior se introduce un paso adicional para el refinamiento previo de aquellos elementos de frontera en las que el peso molecular esté por debajo del valor crítico. En la Figura 11-3 se representa el diagrama de flujo de la metodología integrada con refinamiento.

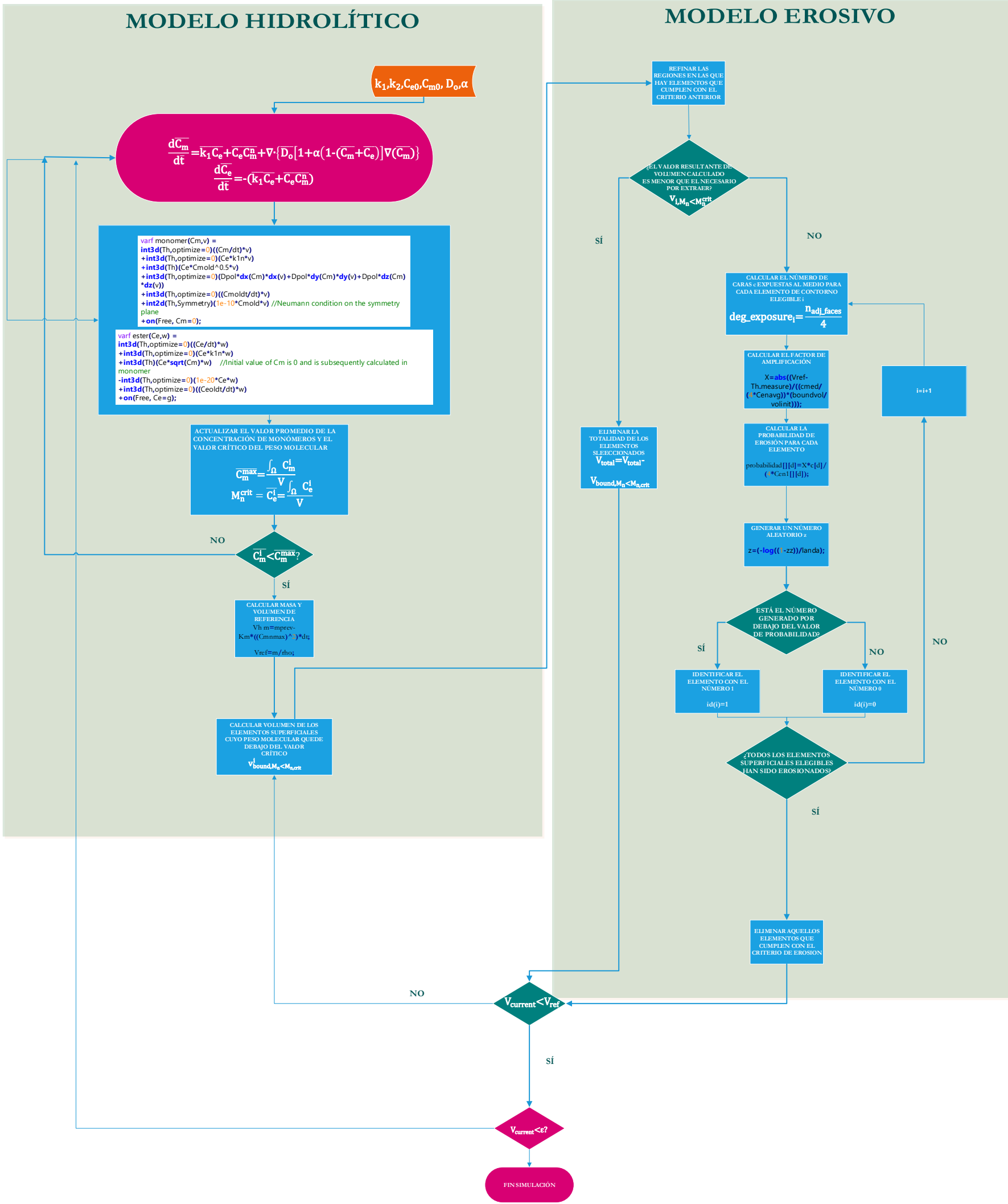
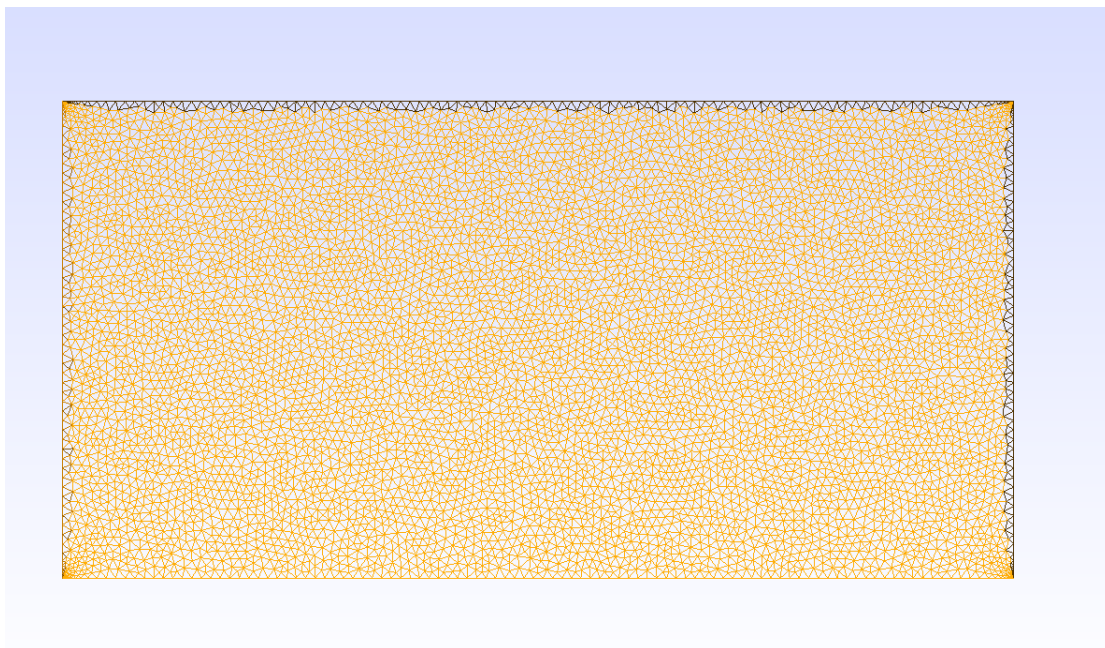


Figura 11-3. Diagrama de flujo correspondiente a la implementación numérica del modelo integrado de degradación hidrolítica y erosión con refinamiento de malla.

## CASO BASE. MODELADO DE LA EROSIÓN DE UNA PLACA RECTANGULAR

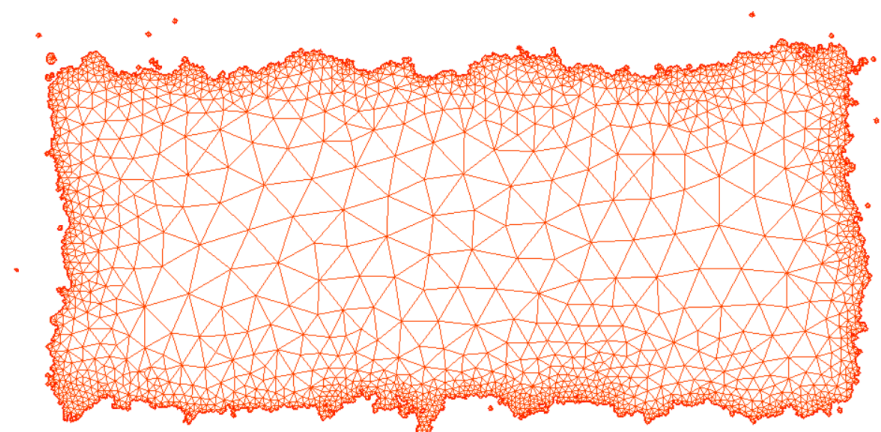
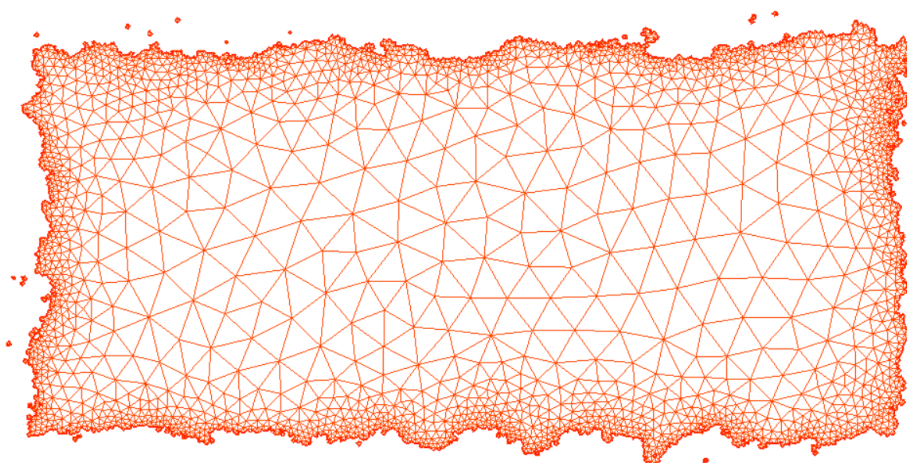
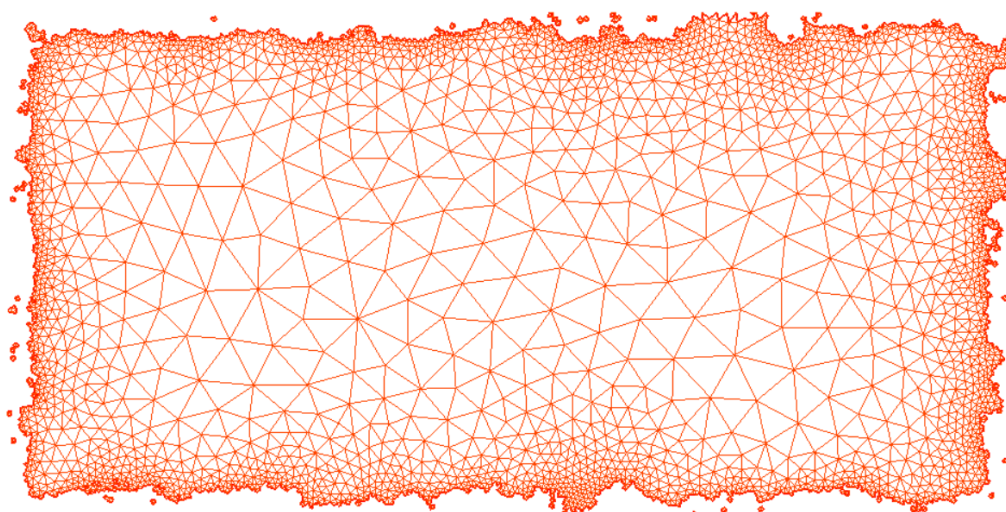
Para evaluar la idoneidad, en términos computacionales del modelo propuesto, analizamos su comportamiento en una dominio rectangular discretizado de forma no estructurada como se observa en la Figura 11-4.



**Figura 11-4. Representación del dominio discretizado utilizado como caso base para evaluar el comportamiento numérico de la segunda propuesta.**

Inicialmente, la malla presenta un refinamiento localizado en los vértices del dominio. Esto se debe a que, al aplicar un refinamiento uniforme sobre las aristas, los puntos de intersección (esquinas) reciben implícitamente una mayor densidad de nodos, generando un mallado más fino en esas zonas.

En este caso no se establece diferenciación en torno a las regiones puesto que la distribución de peso molecular no se calcula en este caso, por este motivo el refinamiento se producirá a lo largo de la frontera. desplazamiento de los nodos es susceptible de ser producido en cualquier punto de la frontera. En la Figura 11-5 se observa cómo el dominio se va erosionando de forma uniforme empezando por la frontera. También se observa la aparición de elementos flotantes, que se justifica en términos de la presencia de



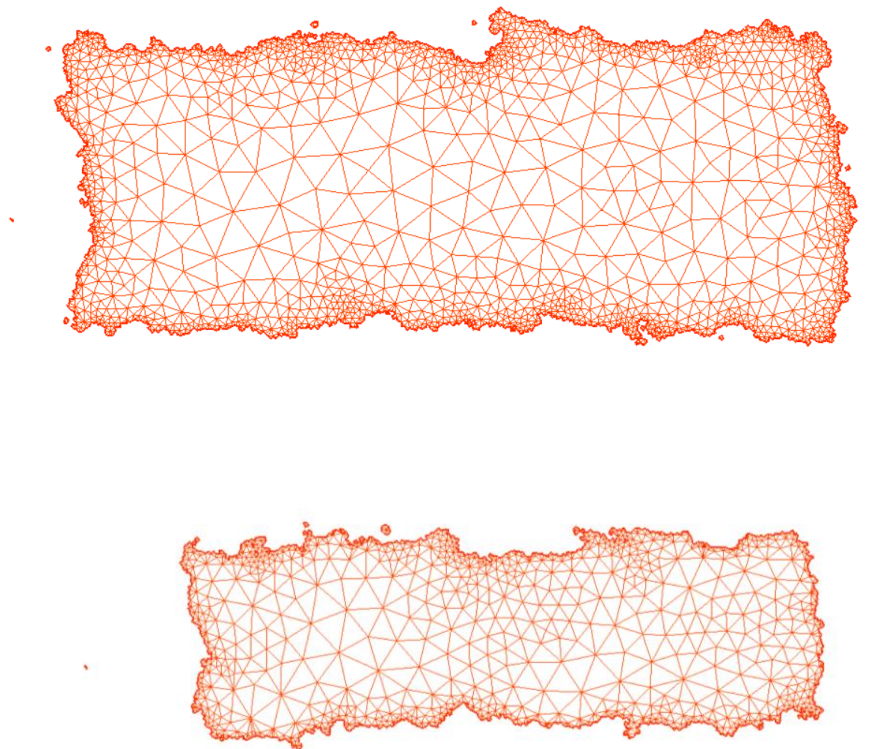


Figura 11-5. Evolución del proceso erosivo al aplicar un refinamiento local en el contorno.

Para realizar un refinamiento en 3d, se empleará la librería externa **tetgreconstruction** de FreeFem++.

Para ello será necesario definir métrica  $h$  en base a aquellas regiones que cumplan con la condición establecida de. La métrica calculada es un campo escalar que nos da el parame

$$M_n \leq M_{n,crit} \quad (11-8)$$

Para ello se define una variable de estado **Bid** de tal manera que se asignan a 1 aquellos elementos que cumplan la condición de ser elementos de frontera, y, a su vez, presenten un valor calculado de  $\overline{C}_e$ , o en su defecto,  $\overline{M}_n$  menor que el valor crítico. En caso contrario, la variable se asigna a 0. Inicialmente la variable toma valores nulos, y se irá modificando en cada elemento que cumpla las requisitos previamente establecidos:



```
Xh Bid=0;
```

```
for (int k=0; k<Th.nbe; ++k){
    int d = Th.be(k).Element;
    if ((Bid[][d]==0 && Th.be(k).label==1 && Cen1[][d]<Mncrit)){
        Bid[][d]=1;
        boundvol=boundvol+vol[][d];
    }
}<<<<<
```

Este argumento, se utilizará como input en la obtención de la métrica se usa el comando mshmet como sigue:

```
h[]=mshmet(Th,Bid,normalization=1,aniso=0,nbregul=1,hmin=Th.measure/40,h
max=Th.measure/10,err=0.0001);
```

Donde:

- Th es la malla de estudio, sobre la cual se calcula la métrica.
- **normalization** escala los valores de h para lograr una métrica mas coherente y estable numéricamente,
- **aniso** indica que si en la reconstrucción de la malla pueden aparecer tetraedros no equiláteros.
- **Nbregul** indica el número de iteraciones de suavizado de la malla. En este caso se deja por defecto a 1.
- **hmin** representa el tamaño mínimo de elemento.
- **hmax** indica el tamaño máximo de elemento en las regiones que no verifican las condiciones de refinamiento.
- **Err**, representa el umbral de error considerado como aceptable para la solución del problema de refinamiento en la malla para el campo h definido.

Esta métrica, que depende de la posición espacial dentro de la pieza o dominio de estudio, se pasa como argumento de entrada en el comando “tetgenreconstruction” en términos del tamaño del tetraedro. Por otra parte, el volumen del tetraedro, para una malla isotrópica debe aproximarse al valor, que de acuerdo a la métrica definida, sería  $h^3/6$ , donde  $h$  representa el tamaño local definido por dicha métrica. La implementación quedaría como sigue:

```
Th=tetgreconstruction(Th,switch="raAQ",sizeofvolume=h*h*h/6.);
```

Donde:

- Th es la malla de estudio, sobre la cual se aplica el refinamiento.
- Switch = “raAQ” representa, en este orden, refinamiento, tamaño objetivo por volumen, autoetiquetado para mantener diferenciación tras la reconstrucción, y mejora de calidad geométrica.
- Sizeofvolume que se asocia al volumen local del tetraedro previamente indicado.

Replicando esto sobre el modelo de la primera propuesta desarrollada, se observa, en la Figura 11-6 un primer refinamiento local situado en la parte central inferior de la cara circular, que es la misma en la que se establecía la erosión en el primer modelo integrado, resultando en una erosión superficial en dicha región. Sin embargo, al cabo de la primera iteración el programa no es capaz de continuar las ejecuciones. Esto supone un empeoramiento respecto de la primera propuesta. Esto puede ser debido a que el algoritmo empleado para el refinamiento local en 3 dimensiones no es el mismo que el empleado en el caso bidimensional. FreeFem++ trabaja nativamente con una librería para el discretizado de geometrías en 2D llamada Bidimensional Anisotropic Mesh Generator (BAMG) que a su vez, dispone de un comando para la adaptación de malla llamado **adaptmesh** que permite obtener refinamientos locales bastante homogéneos. Sin embargo, en 3d las librerías externas, entre las que se encuentran MMG3D y Tetgen, permiten hacer un refinamiento indicando una métrica para el campo de altura  $h$  del tetraedro. Sin embargo, en ambos casos, el funcionamiento del refinamiento local depende en gran parte de cómo se encuentre discretizada la geometría inicialmente. El refinamiento de la geometría introduce patrones irregulares de mallado, como se observa en la imagen izquierda de la Figura 11-6, lo cual puede generar una desestructuración interna localizada de la malla que puede inducir a enredos lo que puede derivar en última instancia a un error de cómputo. Otro motivo que puede constituirse como causante del fallo es la utilización de mallas estructuradas, como es el caso que nos ocupa, y en el que un refinamiento localizado puede derivar en una desestructuración, dando lugar a distorsiones excesivas de la malla, y en última instancia, en un fallo numérico.

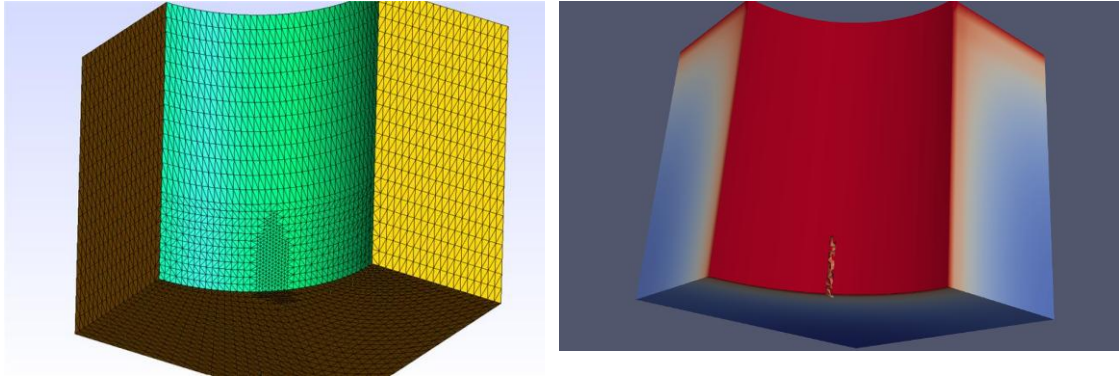


Figura 11-6. Imagen de la celda unitaria discretizada y la distribución del peso molecular en distintas etapas del proceso de degradación.

Dicho esto, a continuación se presenta una propuesta que, trabajando con fronteras ficticias, es capaz de describir mejor los cambios en la interfase del dispositivo durante el proceso de erosión superficial.

## 11.4. INTEGRACIÓN DE LOS MODELOS DE DEGRADACIÓN Y EROSION MEDIANTE LA APLICACIÓN DE DOMINIOS FICTICIOS.

### 11.4.1. INTRODUCCION

Desde la perspectiva computacional, el trabajar con fronteras irregulares que se van actualizando deriva en una inestabilidad creciente del modelo que resulta finalmente en un error de cómputo debido a la incapacidad del algoritmo de aplicar las integrales de contorno, como se ha visto en los capítulos anteriores. Diversos autores han abordado el uso de la optimización topológica para el tratamiento de mallas deformables que se actualizan durante la simulación. Glowinsky et al. [110], [111], [112] desarrollaron modelos matemáticos basados en la técnica de dominios ficticios en los que el dominio de estudio se contiene en un dominio simplificado. Para que la información de la interfase quede bien contenida en la malla ficticia es importante que exista una discretización lo suficientemente fina para que la geometría real quede capturada con precisión dentro del dominio extendido. Este enfoque permite mantener una malla estática, sin cambios, mientras que la evolución de la frontera se modela con funciones auxiliares, como funciones de conjuntos de nivel o *level-set* [109], [113], que, mediante el tratamiento implícito de las interfases, evita que las integrales de contorno se apliquen directamente sobre las fronteras reales, lo cual, como se ha constatado en los dos epígrafes anteriores, deriva, eventualmente, en un error de cómputo.



#### 11.4.2. ESQUEMA GENERAL DEL MODELO.

En la Figura 11-7 se representa el esquema correspondiente al tercer modelo propuesto:

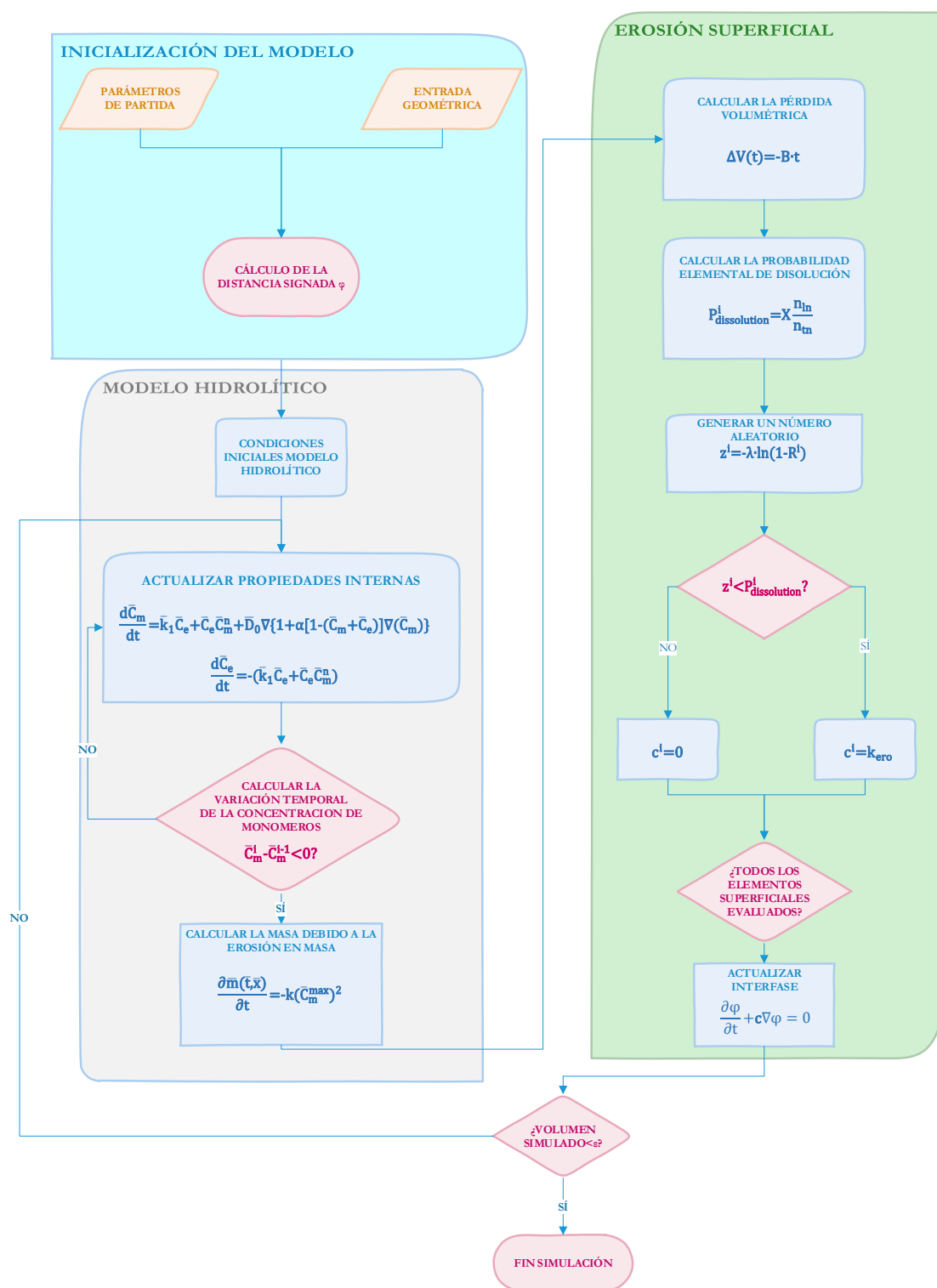


Figura 11-7. Diagrama de flujo del modelo integrado de degradación hidrolítica y erosión superficial.

Teóricamente, el modelo inicial está constituido por dos regiones  $\Omega_s$  y  $\Omega_f$  siendo  $\Gamma_s$  la interfase común a ambas (Figura 8-10). La frontera externa viene representada por  $\Gamma_f$  y es en la cual se aplicarán las condiciones globales de frontera.

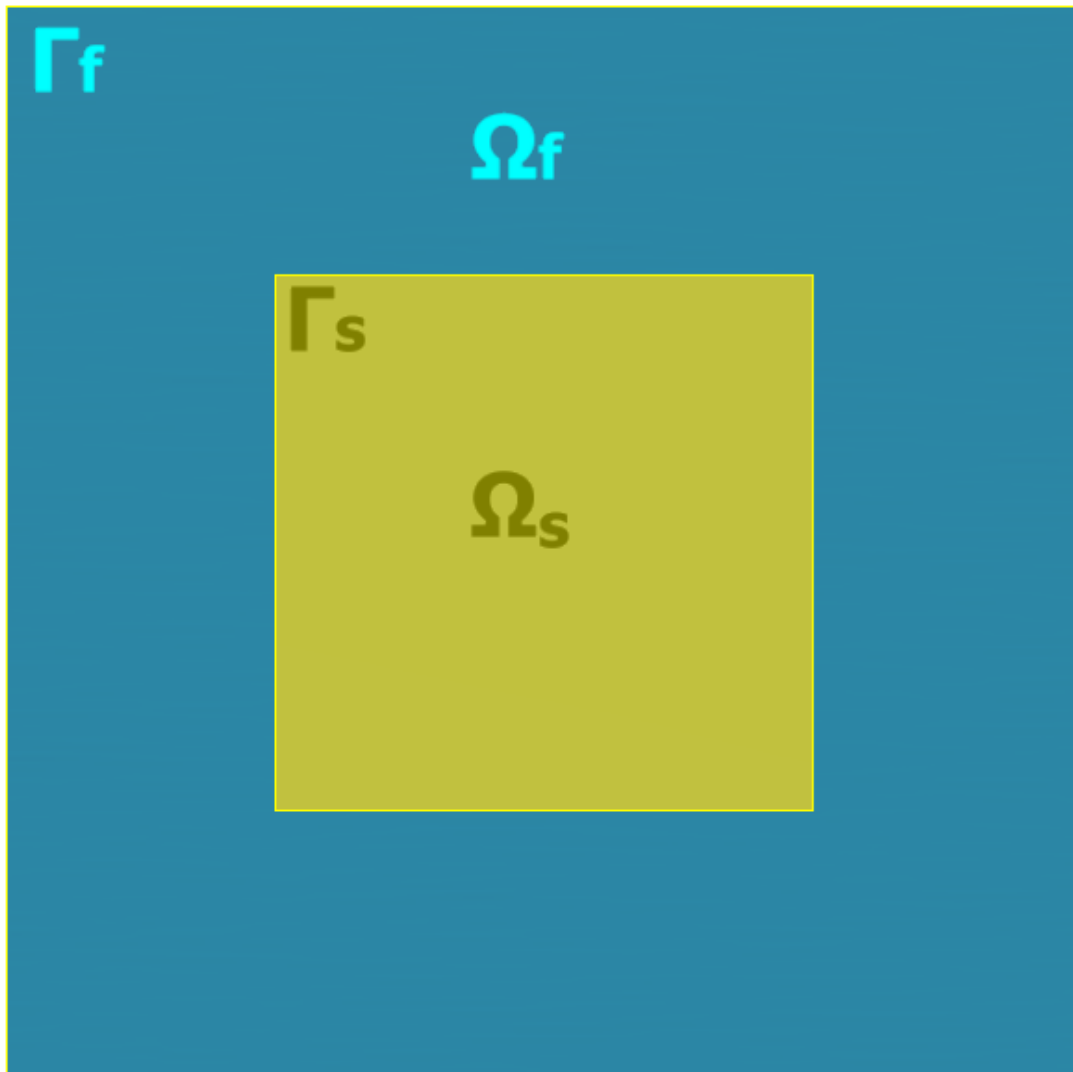


Figura 11-8. Representación dominio con dos regiones simplificado.

La unión de ambos dominios se denominará como:  $\Omega_t = \Omega_s \cup \Omega_f$ .

Sea  $V_h$  un subespacio de  $H^1(\Omega_t)$ , se define la distancia signada  $\varphi$  como la mínima distancia respecto de la interfase  $\Gamma_s$ , de tal manera que sea negativa en la región  $\Omega_s$  y positiva en  $\Omega_f$ :

$$\varphi(x, y) = \begin{cases} -dist & si (x, y) \in \Omega_f \\ dist & si (x, y) \in \Omega_s \\ 0 & si (x, y) \in \Gamma_s \end{cases} \quad (11-9)$$

Donde *dist* es el valor mínimo de la distancia euclídea con respecto a la interfase común  $\Gamma_s$

$$dist: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^+ \quad (11-10)$$

Para hacer esto existe una instrucción en Freefem ++ que nos permite hallar la distancia respecto de la interfase:

```

//////////////////////////////////DISTANCE
COMPUTATION//////////////////////////////////

varf vb(uaux, vaux) = on(1, uaux = 1);
VVh ub = vb(0, VVh, tgv=1);
plot(ub, value=1);

ub[] = ub[] ? 0:1;
plot(ub,value=1);

distance(Th,ub,phi[],distmax=100);
phi=-phi*(region==OMEGA)+ phi*(region==omega);

plot(phi);

H = 1*(phi >= 0) + 0*(phi < 0);
rho = 1300*(phi >= 0) + 1000*(phi < 0);
mu= 1e10*(phi >= 0) + 1e-3*(phi < 0);

```

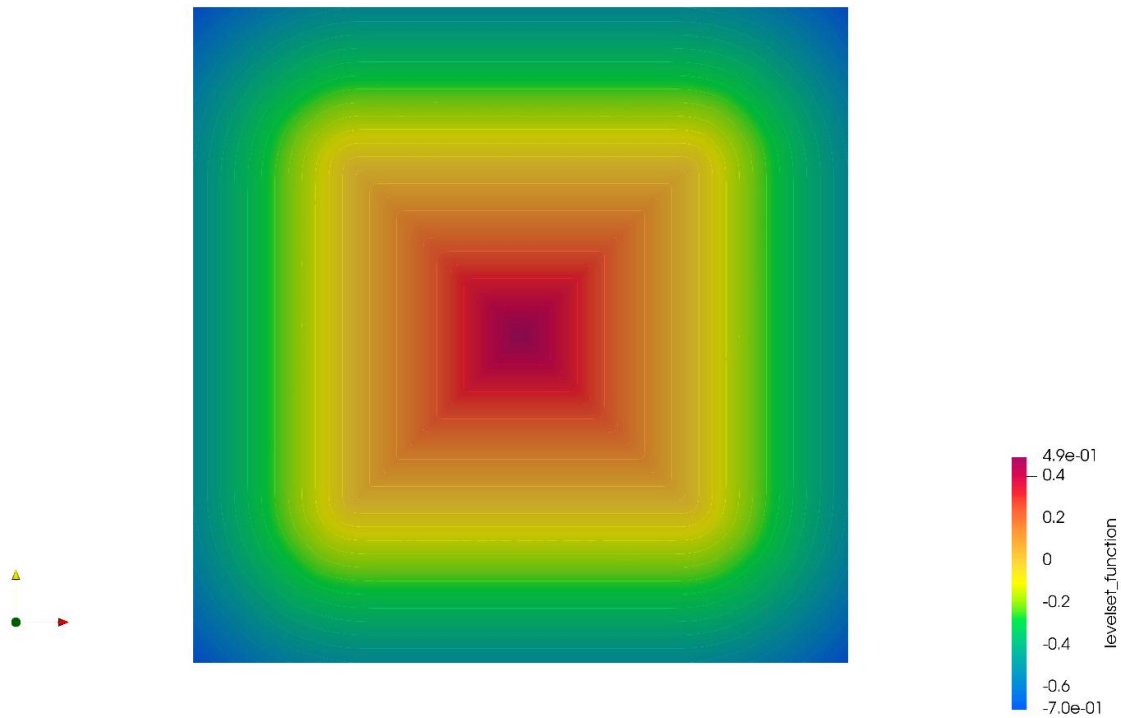


Figura 11-9. Representación de la distancia signada respecto a la interfase

Una vez definida la distancia, la interfase quedará definida por el lugar geométrico de los puntos donde  $\varphi = 0$ . Por tanto, se podrá prescindir de la interfase física definida al principio. Para ello, en el FreeFem existe una instrucción llamada “*change*” que permite modificar parámetros de la malla. En este caso, para eliminar la interfase interior habrá que agregar el parámetro  $rmInternalEdges = 1$  o  $rmInternalFaces = 1$  de la forma:

```
Tht = change(Tht, rmInternalFaces = 1);
```

Para modelar la degradación hidrolítica, se usan las ecuaciones de reacción difusión de Wang et al [1] definieron un sistema de ecuaciones diferenciales (11-6) adimensionales que permiten estimar en diferentes instantes de tiempo la concentración de ésteres y monómeros .

$$\begin{aligned} \frac{d\bar{C}_m}{dt} &= \bar{k}_1 \bar{C}_e + \bar{C}_e \bar{C}_m^n + \bar{D}_0 \nabla \{1 + \alpha [1 - (\bar{C}_m + \bar{C}_e)] \nabla (\bar{C}_m)\} \\ \frac{d\bar{C}_e}{dt} &= -(\bar{k}_1 \bar{C}_e + \bar{C}_e \bar{C}_m^n) - H \frac{\bar{k}_1}{\varepsilon} C \end{aligned} \quad (11-11)$$

Las concentraciones normalizadas de monómeros y ésteres se denotan como  $\bar{C}_m$  y  $\bar{C}_e$ , respectivamente. Ambas se definen en el subespacio  $H^1(\Omega_s)$ . Además, la constante no catalítica se representa como  $\bar{k}_1$ . Los términos  $\bar{k}_1\bar{C}_e$  y  $\bar{C}_e\bar{C}_m^n$  dan cuenta de los procesos no catalíticos y catalíticos, respectivamente. Inicialmente, los valores de  $\bar{C}_m$  y  $\bar{C}_e$  son:

$$\begin{aligned}\bar{C}_m|_{t=0} &= 0 \\ \bar{C}_e|_{t=0} &= 1\end{aligned}\tag{11-12}$$

Asimismo, en la frontera libre  $\Gamma_s$ , se asume que la liberación de subproductos de la reacción química se produce de forma instantánea. Por tanto, se asume que:

$$\bar{C}_m|_{\Gamma_s} = 0\tag{11-13}$$

Sustituyendo esta expresión en la segunda ecuación del sistema (11-11) tenemos que el valor de la concentración de ésteres en la interfase común es:

$$\begin{aligned}\frac{d\bar{C}_e}{dt} &= -\bar{k}_1\bar{C}_e \\ \frac{d\bar{C}_e}{\bar{C}_e} &= -\bar{k}_1 dt \\ \int_0^{\bar{C}_e} \frac{d\bar{C}_e}{\bar{C}_e} &= \int_0^t -\bar{k}_1 dt \\ \ln\bar{C}_e &= -\bar{k}_1 t \\ \bar{C}_e|_{\Gamma_s} &= e^{-\bar{k}_1 t}\end{aligned}\tag{11-14}$$

Tanto la condición (11-13) como la condición (11-14) están definidas en el subespacio  $H^{-\frac{1}{2}}(\Gamma_s)$ . Sean  $v$  y  $w$  dos funciones definidas en el subespacio  $H_1^0(\Omega_s)$ . Integrando las ecuaciones respecto del dominio  $\Omega$  y aplicando el teorema de la divergencia para el término difusivo  $\nabla\{\bar{D}_o[1 + \alpha(1 - (\bar{C}_m + \bar{C}_e))]\nabla(\bar{C}_m)\}$ . En este caso la difusividad se aproximará con los valores de  $\bar{C}_m$  y  $\bar{C}_e$  para el instante anterior:

$$D = \bar{D}_o[1 + \alpha(1 - (\bar{C}_m^* + \bar{C}_e^*))]\tag{11-15}$$

La formulación variacional es la siguiente:

$$a_{\Omega_s}(\overline{C_m}, v) = \int_{\omega_s} \left( \frac{d\overline{C_m}}{d\bar{t}} - (\overline{k_1 C_e} + \overline{C_e C_m^n}) \right) \cdot v + \int_{\omega_s} [\overline{D_o}[1 + \alpha(1 - (\overline{C_m} + \overline{C_e}))]] \cdot \nabla(\overline{C_m}) \cdot \nabla(v)$$

$$a_{\Omega_s}(\overline{C_e}, w) = \int_{\omega_s} \left( \frac{d\overline{C_e}}{d\bar{t}} + (\overline{k_1 C_e} + \overline{C_e C_m^n}) \right) \cdot w$$

Existen varios planteamientos para la resolución numérica de ecuaciones diferenciales usando el método de los dominios ficticios [114] . Uno de ellos consistiría en transformar la condición Dirichlet descrita en el modelo variacional en una integral definida en el contorno de la interfase. A efectos numéricos, en el FreeFem++ no existe una instrucción que permita definir la integral sobre una interfase que no esté previamente identificada a través de las etiquetas o labels.

Por tanto, para poder transformar estas expresiones en funciones que puedan ser resueltas por el programa, se transformará la integral de contorno calculando una función característica en la interfase

Para extrapolar las expresiones anteriores al dominio conjunto  $\Omega_f + \Omega_s$  es necesario transformar la condición Dirichlet en cada una de las expresiones. Considerando una función característica  $G$  que toma que el valor unitario prefijado en un subconjunto pequeño  $G$  alrededor de la interfase y 0 en cualquier otro punto

$$G\{x \in \Omega : \begin{cases} 1 & \text{si } d(x, \Gamma_s) \leq \epsilon \\ 0 & \text{de otra manera} \end{cases}$$

Esto puede expresarse en términos de la función signada de la siguiente manera:

$$G = \begin{cases} 1 & \text{si } -\epsilon < \varphi < \epsilon \\ 0 & \text{de otra manera} \end{cases} \quad (11-16)$$

de dominio del producto de la función característica por el valor de la condición, por la función test:

$$\int_{\Omega_t} G \cdot C_m|_{\gamma} \cdot v$$

El problema para resolver sería: calcular el par  $(\widetilde{C_m}, \widetilde{C_e}) \in V_h \times H^{-\frac{1}{2}}(\Gamma_s)$  tal que:

$$a_{\Omega}(\widetilde{C}_m, v) = \int_{\omega_s} \left( \frac{d\widetilde{C}_m}{d\bar{t}} - (\overline{k_1 C_e} + \overline{C_e C_m^n}) \right) \cdot v + \int_{\omega_s} [\overline{D_o} [1 + \alpha(1 - (\overline{C_m} + \overline{C_e}))]] \cdot \nabla(\overline{C_m}) \cdot \nabla(v)$$

Donde  $\widetilde{C}_m$  y  $\widetilde{C_e}$  son las variables extendidas de  $\overline{C_m}$  en el dominio  $\Omega_t = \Omega_s + \Omega_f$ :

$$\widetilde{C}_m, \widetilde{C_e} \in$$

E incluyendo este término en la ecuación tendríamos:

$$\begin{aligned} \int_{\omega_s} \left( \frac{d\widetilde{C}_m}{d\bar{t}} - (\overline{k_1 C_e} + \overline{C_e C_m^n}) \right) \cdot v + \int_{\omega_s} [\overline{D_o} [1 + \alpha(1 - (\overline{C_m} + \overline{C_e}))]] \cdot \nabla(\overline{C_m}) \cdot \nabla(v) \\ - \int_{\Omega_t} G(0) \end{aligned}$$

Replicando el mismo procedimiento para la ecuación de la variación de concentraciones éster.

$$\int_{\omega_s} \left( \frac{d\widetilde{C_e}}{d\bar{t}} + (\overline{k_1 C_e} + \overline{C_e C_m^n}) \right) \cdot w - \int_{\omega_s} H$$



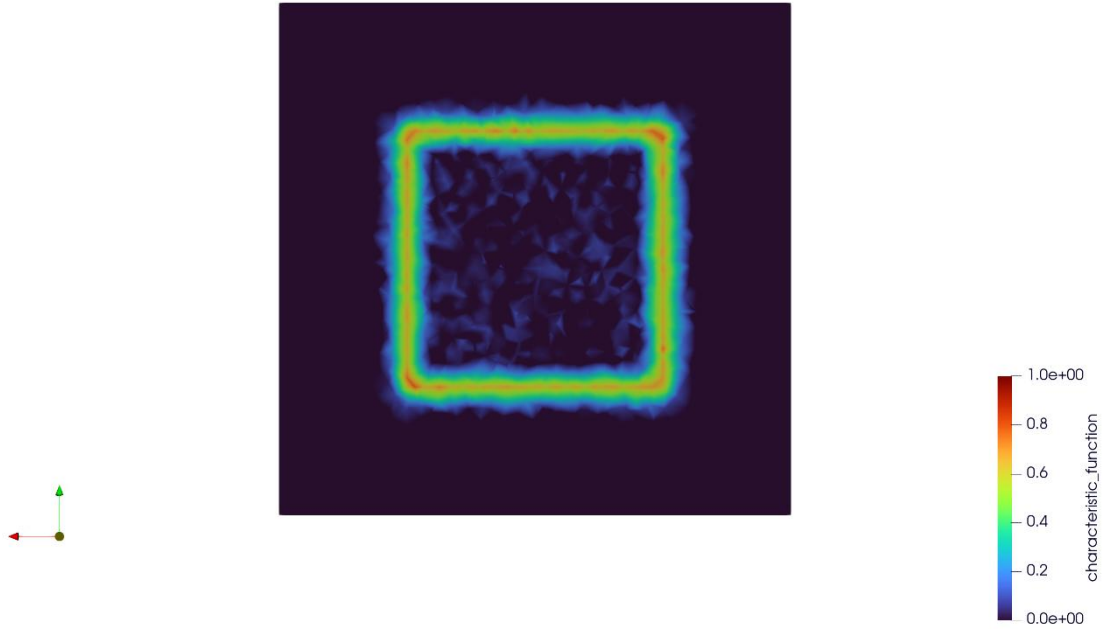


Figura 11-10. Representación en vista de sección de la función característica usada para aplicar condiciones de contorno.

Extrapolando las ecuaciones (11-11) al dominio  $\Omega_s + \Omega_f$  se incluye un término de penalización a la primera de las ecuaciones del tipo:

$$G \frac{k}{\varepsilon \bar{c}_m} \quad (11-17)$$

Donde G es una función característica que vale 0 en el medio sólido y 1 en el medio acuoso.  $\varepsilon \bar{c}_m$  es un término de valor muy pequeño de tal manera que la función solo calculará las expresiones del tipo:

$$\frac{d\bar{c}_m}{dt} = \bar{k}_1 \bar{c}_e + \bar{c}_e \bar{c}_m^n + \bar{D}_0 \nabla \{1 + \alpha [1 - (\bar{c}_m + \bar{c}_e)] \nabla (\bar{c}_m)\} - G \frac{\bar{k}_1}{\varepsilon \bar{c}_m} \bar{c}_m \quad (11-18)$$

Esto simula la degradación de las partículas. De acuerdo con Heljak et al. [30] el material empieza a perder masa cuando la concentración de monómeros alcanza su máximo, y establece una relación entre la variación de masa y el valor máximo de concentración:

$$\frac{\partial \bar{m}(\bar{t}, \bar{x})}{\partial t} = \begin{cases} 0, & \bar{t} \leq \bar{t}_{max} \\ -\bar{k}_m(\bar{C}_m^{max})^2 & \bar{t} \geq \bar{t}_{max} \end{cases} \quad (11-19)$$

Este tipo de pérdida de masa no implica una pérdida volumétrica de material. Por tanto, no se tiene en cuenta para el cálculo de volumen de referencia.

Retomando la expresión de probabilidad introducida en 11.2:

$$P_{dissolution} = X( \quad (11-20)$$

$$X = \frac{((\bar{V}_0 - \bar{V}_{ref}) - \bar{V}_{loss})}{\bar{V}_{boundary} \frac{C_{average}}{4\bar{M}_{naverage}}} \quad (11-21)$$

Para algunos polímeros la erosión comienza tan pronto como el material está en contacto con el medio [115]. La tasa de pérdida volumétrica de un material para algunos polímeros viene definida por:

$$\Delta V(t) = -B * t \quad (11-22)$$

Esto representa un comportamiento lineal del material en el proceso erosivo cuando la fracción volumétrica tiende de forma asintótica a un valor mínimo:

$$\Delta V(t) = -V_0 \cdot e^{-B \cdot t} \quad (11-23)$$

Sustituyendo en la expresión (11-21) tenemos que la probabilidad elemental de disolución es:

$$P_{dissolution}^i = \frac{(B \cdot t \cdot \bar{V}_{loss})}{\bar{V}_{boundary} \frac{C_{average}}{4\bar{M}_{naverage}}} \frac{n_{ln}}{n_{tn}} \quad (11-24)$$

Una vez calculada la probabilidad de disolución se genera un número aleatorio z basado en una ley exponencial del tipo:

$$z_i = -\lambda \ln(1 - R^i) \quad (11-25)$$

Donde R toma valores definidos en el intervalo [0,1].

Si el valor de  $z$  queda por debajo de la probabilidad el elemento se procederá a eliminarse. Esto se modelará con una función  $c_\gamma$  denominada velocidad de deformación interpolada según la variable  $\varphi$ . Se definirá un valor de  $c$  de acuerdo con la elegibilidad de los elementos disueltos de la siguiente manera:

$$c_\gamma^i = \begin{cases} h_{triangle}/\delta_t & \text{if } z_i \leq P_{dissolution}^i \\ 0 & \text{de otra manera} \end{cases} \quad (11-26)$$

Para actualizar la interfase de acuerdo al modelo de erosión, se recurre a modelos basados en Conjuntos de Nivel (Level Set Method, en inglés) introducidos inicialmente por Stanley Osher et al. [109] en los que la distancia signada  $\varphi$  se deforma, en la interfase, de forma convectiva, por medio de un campo de velocidad  $\mathbf{c}$ , que dependerá de la variable que produce el fenómeno (en este caso la erosión) y que será calculada a través de las ecuaciones multiplicando se calcula igualando su derivada total a 0:

$$\frac{\partial \varphi}{\partial t} + \mathbf{c} \nabla \varphi = 0 \quad (11-27)$$

Donde  $\mathbf{c}$  es una generalización de la variable  $c_\gamma$  definida en la interfase entre el polímero y el medio y es igual al producto de su módulo por un vector unitario normal a la interfase:

$$\mathbf{c} = c \cdot \mathbf{n} \quad (11-28)$$

Siendo  $\mathbf{n}$  el campo vectorial unitario definido en  $\Omega_s \cup \Omega_f$  de la normal a la frontera que se hace 1 en la interfase:

$$\mathbf{n} = \frac{\nabla \varphi}{\|\nabla \varphi\|} \quad (11-29)$$

Una vez la interfase queda actualizada se volverán a aplicar las ecuaciones hidrolíticas. Al finalizar cada etapa combinada de degradación-erosión se evaluará el volumen del dominio  $\Omega_f$ , de tal manera que, si queda por debajo de un valor  $\varepsilon$ , que en este caso se asumirá igual a  $10^{-3}$ , se dará por concluida la ejecución.

Este modelo presenta conforme la interfase se va actualizando una serie de inestabilidades debidas al flujo circundante. En primer lugar, las ecuaciones de transporte que modelan la difusión de las partículas en el medio cuando se trabaja con dominios más complejos requiere de una condición de estabilidad para que las soluciones obtenidas en las diferentes etapas no

diverjan a un valor excesivamente alto. El uso de ecuaciones paramétricas es una condición imprescindible. En términos de paso de tiempo y de la discretización esto se logra a través del número de Courant-Friedrich-Lewy [114] que definieron, a través de una expresión que relaciona el valor de la variable del problema en derivadas parciales con el paso de tiempo  $\Delta t$ , y la discretización espacial a lo largo de los ejes cartesianos,  $\Delta \mathbf{s} = (\Delta x, \Delta y, \Delta z)$ :

$$\frac{u_x \Delta t}{\Delta x} + \frac{u_y \Delta t}{\Delta y} + \frac{u_z \Delta t}{\Delta z} < C \quad (11-30)$$

El valor C generalmente toma valores que se encuentran entre 0 y 1.

#### 11.4.3. PREPROCESADO DEL DOMINIO GEOMÉTRICO.

Aprovechando la simplicidad del modelo, con la finalidad de obtener los resultados más precisos posibles cada una de las tres regiones que componen el modelo de estudio será discretizada en elementos con el fin de lograr un mallado lo más estructurado posible.

- **Mallas Estructuradas:** Ofrecen simplicidad y eficiencia, requieren menos memoria y el software puede acceder a celdas vecinas de manera eficiente.
- **Desafíos:** Puede ser difícil generar una malla estructurada para dominios geométricos complicados y pueden requerir más elementos que una malla no estructurada.
- **Métodos de Generación:** Clasificados en enfoques manuales, métodos algebraicos o de interpolación, y métodos de EDP o variacionales.
- **Mapeo Conforme:** Utilizado en la generación de mallas, preserva ángulos y puede inducir una malla estructurada con elementos que tienden a ángulos de 90°.

El mallado estructurado en el análisis por elementos finitos (FEA) es un proceso crucial que implica dividir una geometría compleja en formas más simples y bien definidas conocidas como elementos. Estos elementos están conectados en puntos llamados nodos, que definen las coordenadas en el espacio. La principal ventaja del mallado estructurado es su simplicidad y eficiencia computacional. Permite un acceso directo a las celdas vecinas, lo cual puede ser particularmente beneficioso para simulaciones que requieren la recuperación frecuente de datos de elementos adyacentes.

Sin embargo, crear una malla estructurada puede ser un desafío para geometrías con formas complejas. En tales casos, puede ser difícil generar una malla estructurada y pueden requerirse más elementos que en una malla no estructurada para lograr una precisión comparable. Los métodos de generación de mallas se clasifican generalmente en enfoques manuales, métodos

algebraicos o de interpolación, y métodos basados en ecuaciones diferenciales parciales (EDP) o variacionales.

Uno de los métodos utilizados en la generación de mallas es el mapeo conforme, que preserva los ángulos y puede inducir una malla estructurada con elementos que tienden a ángulos de  $90^\circ$ . Este método es especialmente útil para aplicaciones que requieren una alta precisión en la representación de la geometría y en la solución de las EDP.

Es importante destacar que el uso de mallas estructuradas está sujeto a las necesidades específicas de cada proyecto de FEA y debe considerarse cuidadosamente para garantizar resultados precisos y eficientes.

Se ha logrado realizar una malla estructurada de la celda unitaria subdividiendo la geometría en elementos hexaédricos (Figura 11-11), pero luego no se ha considerado en los cálculos posteriores debido al alto coste computacional por iteración.

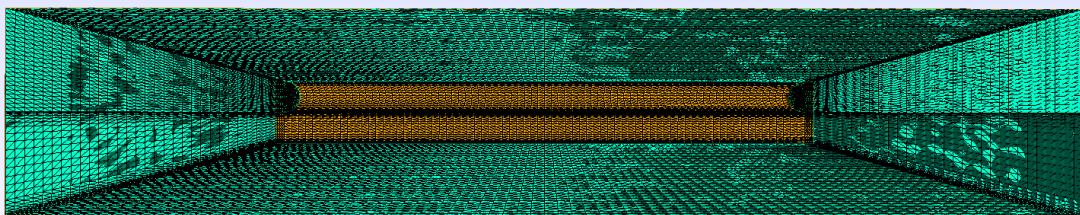


Figura 11-11. Vista de la sección frontal del dominio ficticio discretizado.

En este caso se tomarán los datos propuesto por Agrawal et al. para el ajuste del modelo [28]. En este caso el material es PLGA y los scaffolds tienen diferentes grados de porosidad: baja (80%), media (87%) y alta (92%). Al tratarse de un modelo adimensional podemos representar esto con diferentes topologías que básicamente cumplan el criterio de porosidad descrito anteriormente.

Para modelar el proceso de degradación del scaffold, se considera que la celda unitaria se contiene en un dominio simplificado con forma de paralelepípedo de dimensiones  $10,4 \times 10,4 \times 2 \text{ mm}^3$ , centrado en el centro de gravedad de la celda. El dominio se divide en una malla tridimensional de elementos finitos, donde cada elemento representa una pequeña porción del material. La Figura 11-12 muestra la geometría y la malla utilizada para los diferentes valores de porosidad.

En este caso el volumen representativo viene constituido por de 4 semi-filamentos cilíndricos paralelos 2 a 2 colocados uno encima del otro.

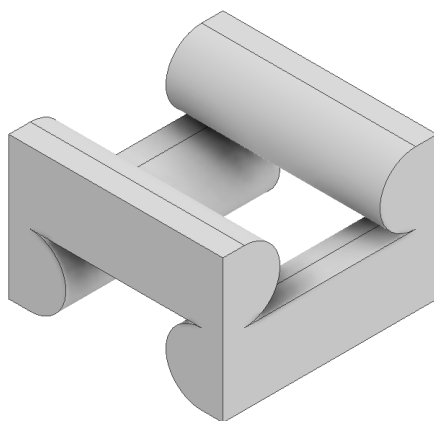


Figura 11-12. Representación 3D de la celda unitaria

Cada uno de los filamentos completo viene definido por dos semicírculos de radio  $r$  y un rectángulo de base  $d$  y altura el doble del radio (Figura 11-13). La longitud viene representada por  $z$  y se calibrará en función de la porosidad que se quiera analizar.

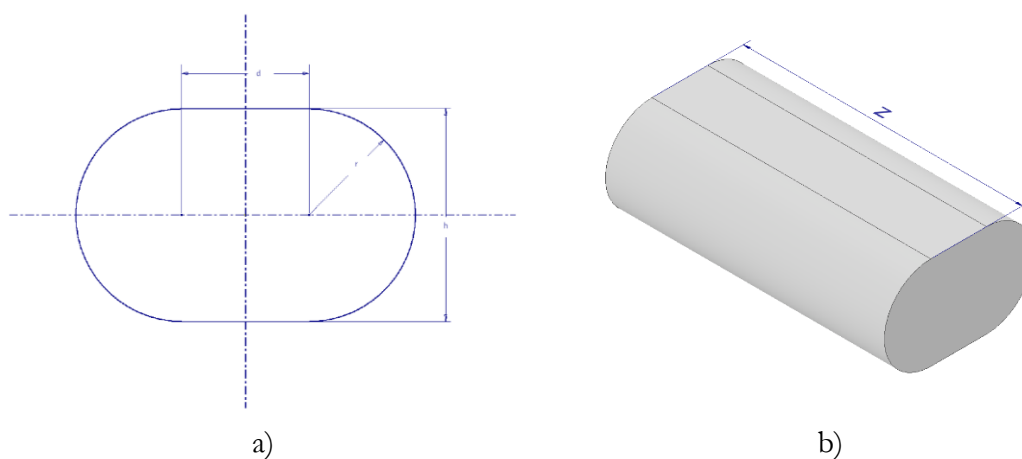


Figura 11-13. Representación frontal(a) y en perspectiva(b) de un filamento semicilíndrico.

Así, el volumen de filamento  $V_{fil}$  será igual a:

$$V_{fil} = (2 \cdot d \cdot r + \pi r^2) \cdot z \quad (11-31)$$

En este caso, para las dimensiones de la sección transversal se tomará la distancia  $d$  igual a  $0,09 \text{ mm}$  el radio  $r$  es igual a  $0,15$  y  $z$  será una variable que se ajustará para lograr la porosidad objetivo y que coincidirá con la distancia entre centros de dos filamentos consecutivos (Figura 11-14).

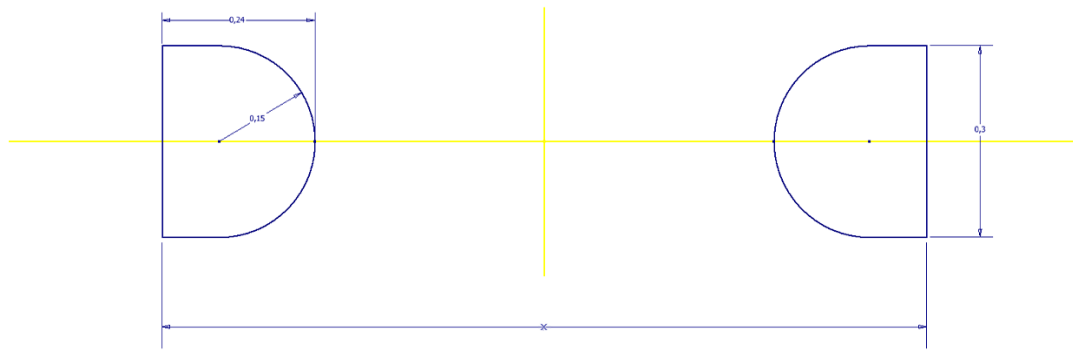


Figura 11-14. Representación de la sección transversal de la celda unitaria.

Para considerar la influencia del medio en el material, los filamentos estarán rodeados en un cubo envolvente al que se le denomina celda unitaria. De tal manera que el volumen de la celda  $V_{cell}$  vendrá definido como la suma del volumen unitario menos el volumen de la pieza.

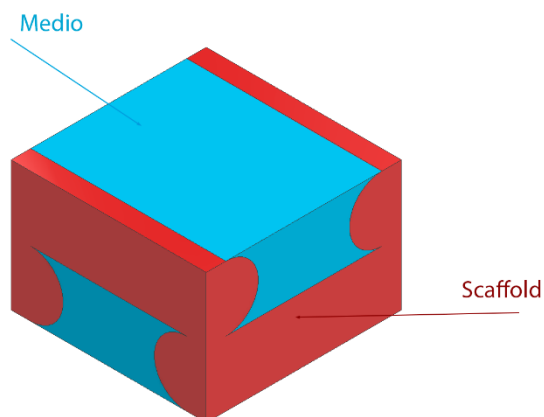


Figura 11-15. Representación de las regiones de la celda unitaria.

Para lograr las diferentes porosidades del ensayo de Agrawal[28], la distancia  $x$  entre los centros de los filamentos será la variable a ajustar, puesto que en función del valor que tome la porosidad tomará un valor u otro. La porosidad,  $p$  se define como la variación porcentual, entre el volumen total de la celda unitaria y el volumen del material:

$$p = \frac{V_{cell} - V_{polymer}}{V_{cell}} \cdot 100 \quad (11-32)$$

Siendo el volumen de la celda:

$$V_{cell} = x^2 h = 2x^2 \cdot 0,3 = 0,6x^2 \quad (11-33)$$

Y el volumen representativo del scaffold en función de la distancia entre centros es:

$$V_{scaffold} = 4 \cdot 0,5 \cdot ((0,09 \cdot 0,3 + \pi \cdot 0,15^2) \cdot x) = 0,1953 \cdot x \quad (11-34)$$

Sustituyendo en la expresión y simplificando nos queda que la expresión de la porosidad en función de la distancia  $x$  es:

$$p = \frac{100x - 32,55}{x} \quad (11-35)$$

De aquí se puede despejar fácilmente la variable  $x$  en función del parámetro  $p$  como:

$$x = \frac{32,55}{100 - p} \quad (11-36)$$

Tomando como referencia los valores de porosidad definidos por Agrawal [28] se obtiene una separación entre centros de 2,1, 3,2 y 5,2 mm respectivamente(Tabla 11-1). Estos vienen definidos por las celdas de la Figura 11-16.

Porosidad	Distancia entre filamentos(x)
80 %	2,1 mm
87 %	3,2 mm



92 %	5,2 mm
------	--------

Tabla 11-1 Distancia entre centro de filamentos en función de la porosidad.

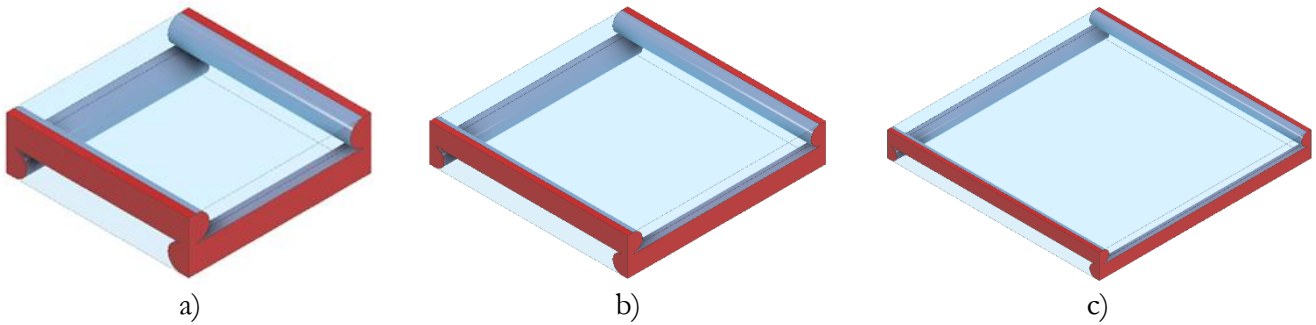


Figura 11-16. Representación geométrica de la celda unitaria del scaffold para un a) 80%, 87%, y 92% de porosidad

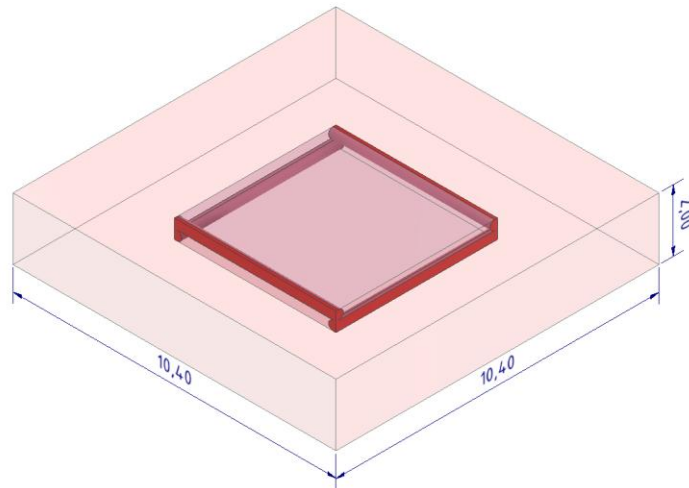


Figura 11-17. Representación de la celda unitaria contenida en dominio ficticio.

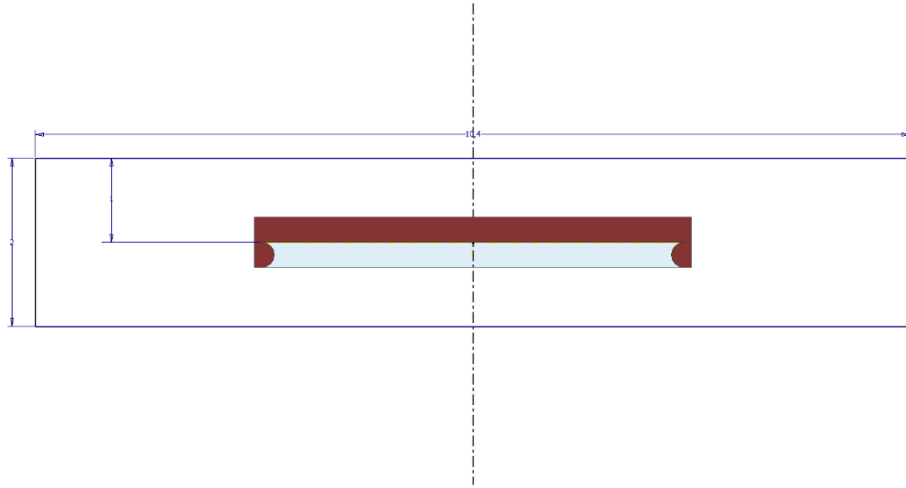


Figura 11-18. Vista en sección frontal de la celda unitaria embebida en el volumen ficticio

#### 11.4.4. CONDICIONES DE CONTORNO

Para la aplicación de las condiciones de frontera se establecerán tres regiones diferenciadas según el orden de importancia: la primera es la región sólida del polímero  $\omega_s$ , seguida de la región correspondiente al medio  $\omega_f$ , y por último, la diferencia del dominio ficticio  $\Omega$  con la celda unitaria  $\omega = \omega_f - \omega_s$  (Figura 11-19).

Asimismo, las variables extendidas del problema,  $\widetilde{C}_m$  y  $\widetilde{C}_e$  tomarán distintos valores en función del lugar en el espacio  $\mathbf{x} = (x, y, z)$  que se considere dentro del dominio ficticio. En el caso de  $\widetilde{C}_m$ , este tomará los valores correspondientes a  $C_m$  en  $\omega$ .

$$\begin{aligned}\widetilde{C}_m(\mathbf{x}) &= \begin{cases} 0 & \text{si } \mathbf{x} \in (\Omega - \omega) \\ C_m(\mathbf{x}) & \text{en otro caso} \end{cases} \\ \widetilde{C}_e(\mathbf{x}) &= \begin{cases} 0 & \text{si } \mathbf{x} \in (\Omega - \omega) \\ 0 & \text{si } \mathbf{x} \in \omega_f \\ C_e(x, y, z) & \text{en otro caso} \end{cases}\end{aligned}\tag{11-37}$$

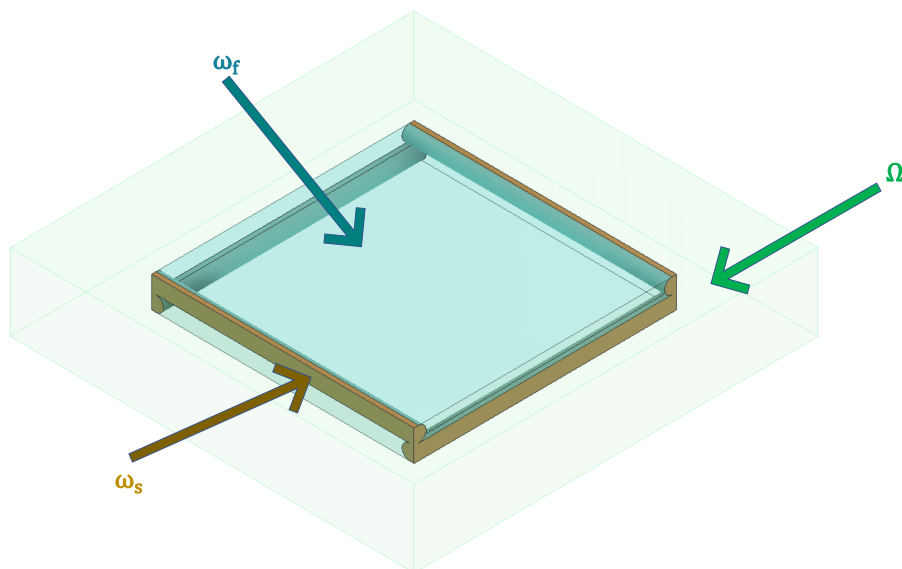
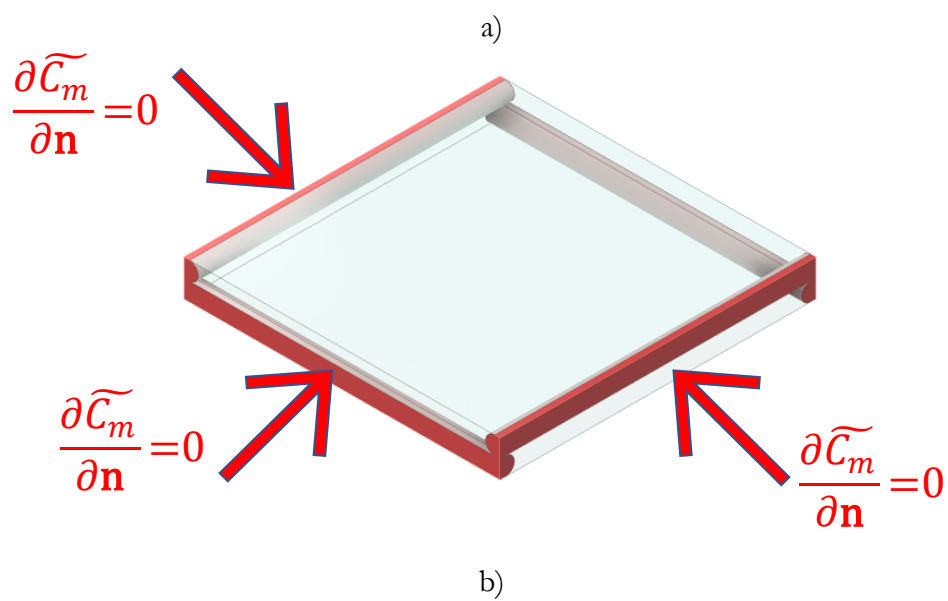
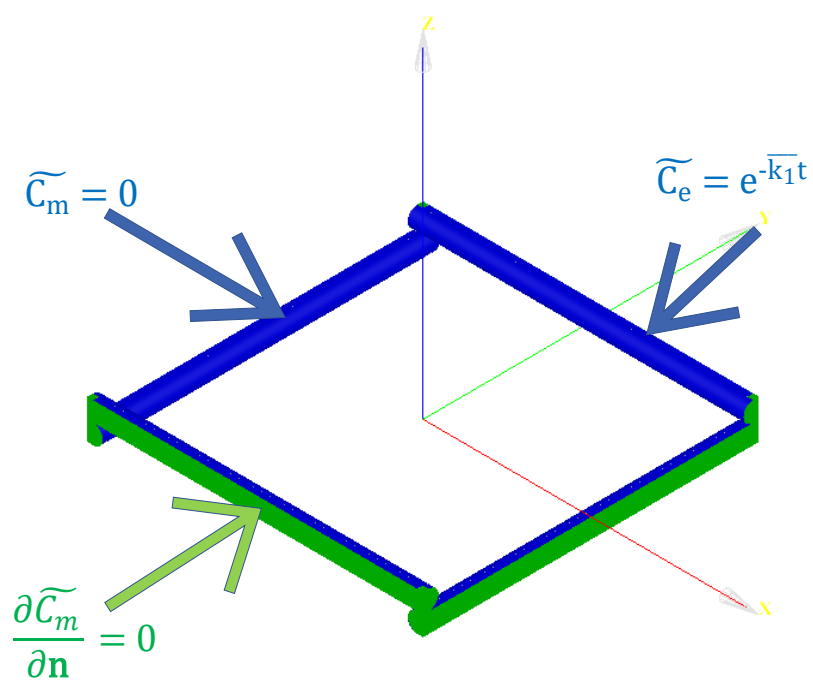


Figura 11-19. Representación de los dominios de estudio del problema.

Las condiciones de contorno dependerán de las fronteras a considerar. En el caso de la interfase del material con el medio  $\gamma_s$  se asumirá una condición de que la concentración de oligómeros es igual a 0 y la concentración de esteres igual a la exponencial negativa del producto de la constante no auto-catalítica normalizada y el tiempo. En la frontera de la celda unitaria  $\omega$  se asumirá una concentración de subproductos igual a 0. En primer lugar, se asume una concentración de oligómeros nula en la superficie libre del material. Para la celda unitaria se asume una condición de frontera periódica y uniforme igual al gradiente de la concentración de subproductos igual a 0.

En la frontera ficticia se asumirá una condición de frontera igual a la concentración de monómeros igual a 0, mientras que en la frontera se asume una condición de contorno de que la concentración de ésteres coincide con la exponencial de la concentración de la constante no auto catalítica por el tiempo:

En las caras exteriores de la celda líquida se asumirán unas condiciones de frontera periódicas igual a que el gradiente de la concentración de subproductos sea igual a 0. Por su parte, en el scaffold, las condiciones a aplicar son de que el gradiente de la concentración sea 0 en las superficies que son secciones y que la concentración de subproductos sea libre en la interfase común con el volumen ficticio y el volumen líquido.



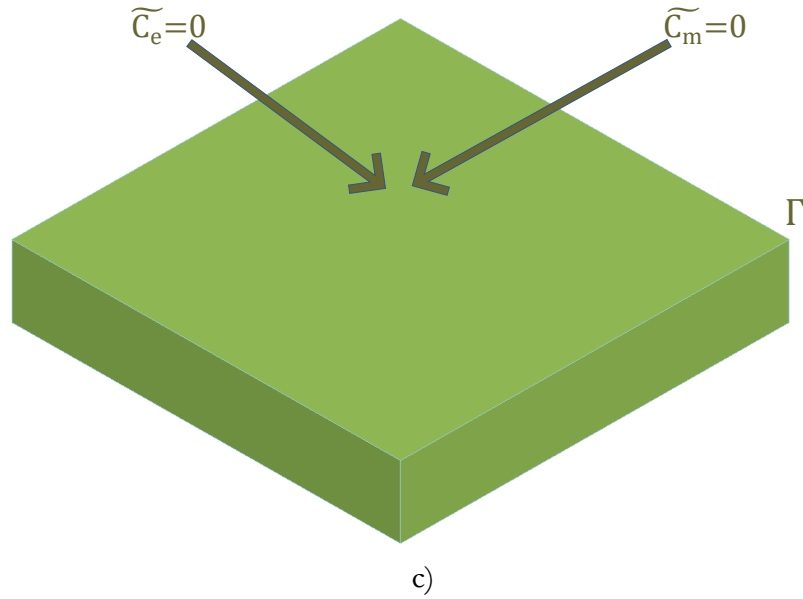


Figura 11-20. Representación de las condiciones de frontera en el scaffold (a), la celda unitaria (b) y en la frontera ficticia (c).

#### 11.4.5. MODELIZACIÓN DE LA POROSIDAD

Para la modelización de la erosión del material. Tanto por masa como por superficie, los autores recurren al cálculo de la porosidad. Pan et al [1], [96], [116], estableció diferentes modelos que daban cuenta de la variación de la degradación hidrolítica. En la literatura existen varios modelos que permiten establecer. Por una parte, Göpferich definió la porosidad como una variación de una. Estos modelos solo dan cuenta de la variación en masa (bulk erosión del material) pero no reflejan la variación del mismo debido a la erosión superficial.

Göpferich ha definido una metodología que establece una relación entre la degradación hidrolítica y la del material en cuestión.

Según los experimentos obtenidos por Agrawal [28], se observa inicialmente una tendencia decreciente en la porosidad del material, para luego incrementarse hasta llegar a su completa disolución. En términos de la degradación hidrolítica, esto es debido principalmente al aumento inicial que tienen los oligómeros, el cual tiene una tasa de producción que viene definida por [117]:

$$\frac{dR_s}{dt} = k_1 C_e + k_2 C_e C_m^n \quad (11-38)$$

Siendo  $C_e$  la concentración de enlaces de ester y  $C_m$  la de monómeros.

Asimismo, la variación de la de la concentración de oligómeros, la cual se puede modelar a partir de las leyes de Fick

Por otra parte, a lo largo del proceso hidrolítico el material se reordena dando lugar a una estructura más cristalina y por consiguiente más opaca, lo que hace que el material reduzca su porosidad interna. Esto es debido en parte, al efecto de la acumulación de los oligómeros en la estructura polimérica, así como en un aumento de la configuración de las cadenas cristalinas con respecto de las amorfas.

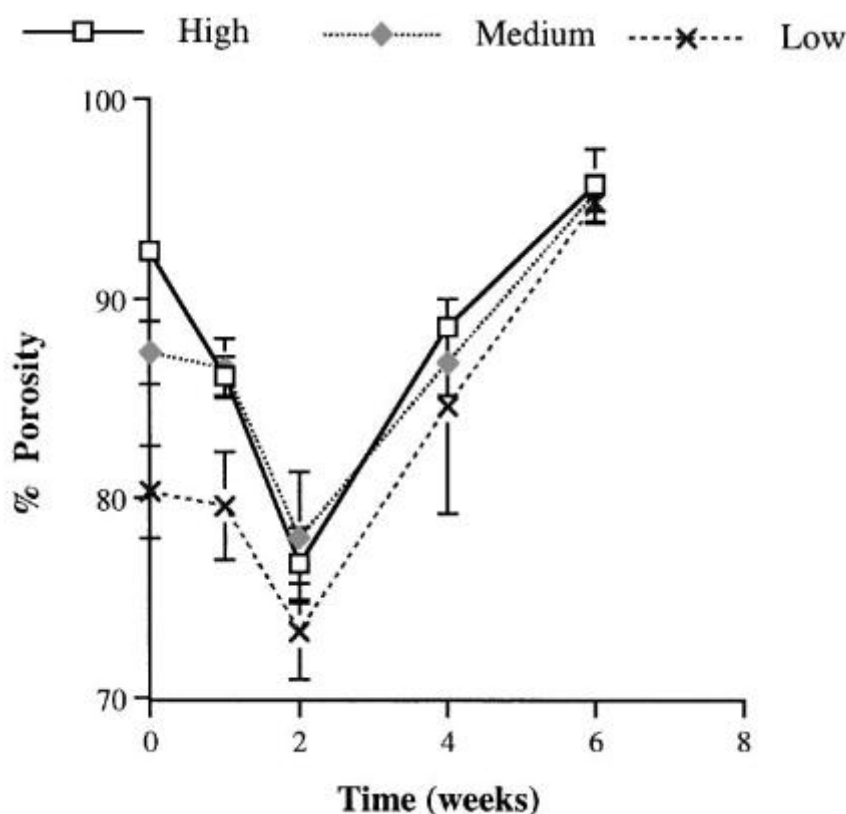


Figura 11-21. Representación de la variación de la porosidad en función del tiempo de degradación expresado en semanas. Tomado con permiso de Agrawal [28].

Después de realizar varios ajustes, se ha visto que la mejor expresión que refleja dicho comportamiento es:

$$p = 100 \cdot (1 - \alpha(t + \beta)^{\gamma} \cdot e^{-\delta(t+\beta)}) \quad (11-39)$$

donde  $\alpha, \beta, \gamma$  y  $\delta$  son parámetros que cuyos valores dependerán de las características intrínsecas del material: cristalinidad, porosidad inicial, volumen inicial y concentración de monómeros (oligómeros).

Por tanto, la resolución de los parámetros vendrá en función de la expresión de la porosidad.

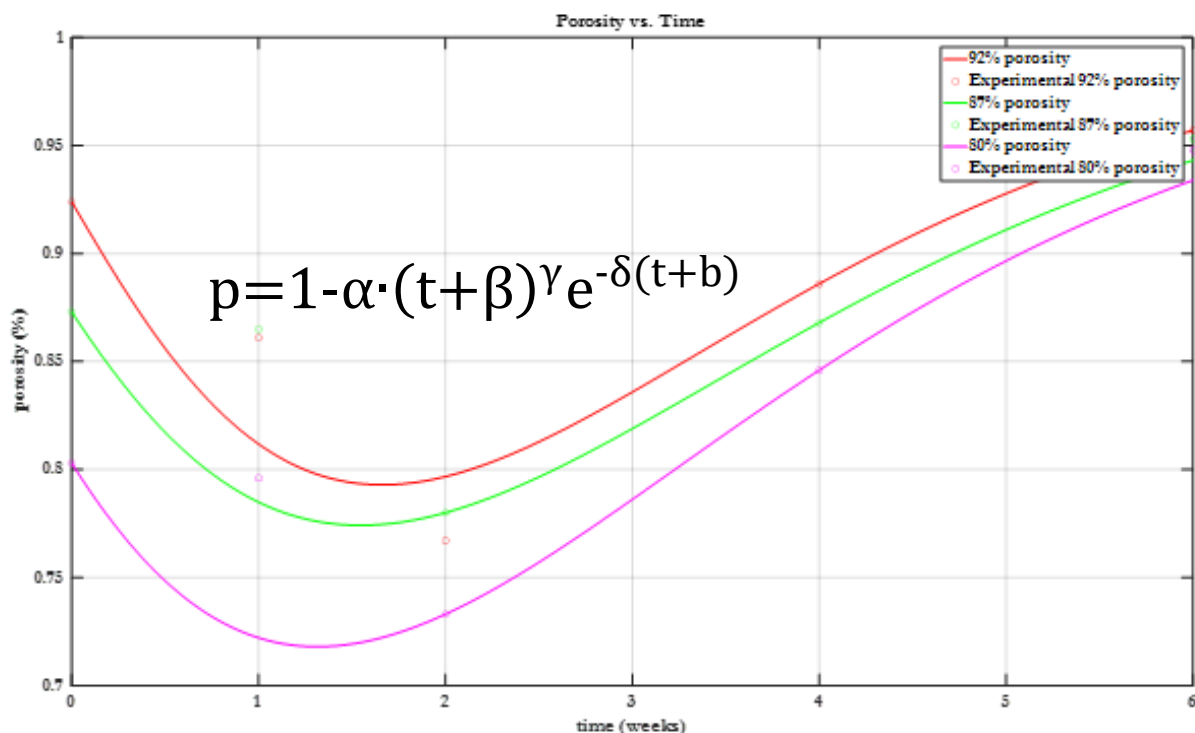


Figura 11-22. Expresión empírica para el cálculo de porosidad de acuerdo con Agrawal

Sin embargo, esta expresión carece de generalidad porque solo tendría validez para el ensayo experimental que se tome de referencia. El objetivo es obtener una expresión de porosidad que dependa de las variables físicas del proceso degradativo y erosivo.

Se define el término porosidad global  $p_{global}$  como la suma ponderada de las porosidades  $p_1$  y  $p_2$ , que depende de los parámetros intrínsecos del material como son la concentración de subproductos  $c_m$  o la cristalinidad del material  $x_c$ . La ponderación está expresada en términos de  $f_{trans}$ , de tal manera que el peso asignado es  $1 - f_{trans}$  para  $p_1$  y  $f_{trans}$  para  $p_2$  ::

$$p_{global} = (1 - f_{trans}) \cdot p_1 + f_{trans} \cdot p_2$$

Aquí  $p_1$  y  $p_2$  son las porosidades en las fases degradativas y erosiva, respectivamente. Por su parte,  $f_{trans}$  es la función de transición para cambiar de la expresión  $p_1$  a la  $p_2$ :

$$p_1 = p_{surface} \cdot \left( 1 - \left( 1 - \frac{\chi}{1 + e^{\gamma(x_c - x_{c_0})}} \right) \left( \frac{c_{m_{avg}}^{i-1}}{c_{m_{avg}}^i + \epsilon} \right)^\psi \right) \quad (11-40)$$

$$p_2 = 100 - p_{surface} \cdot \beta \cdot e^{-\alpha \frac{\chi}{1 + e^{\gamma(x_c - x_{c_0})}} \frac{c_{m_{avg}}^{i-1}}{c_{m_{avg}}^i + \epsilon}} \quad (11-41)$$

$$f_{trans} = v(1 + \tanh(\beta(t - t_{trans}))) \quad (11-42)$$

Donde:

$\gamma$ : parámetro de ajuste para lograr una mejor aproximación a los resultados de referencia.

$x_c$ : es la cristalinidad del material en la etapa actual del proceso de degradación. Se calculará de acuerdo con la fórmula de recristalización de Avrami

$x_{c_0}$ : es la cristalinidad inicial del material. Se asume un valor de 0.8 para la cristalinidad del material

$c_{m_{avg}}^{i-1}$ : es el valor promedio de la concentración de monómeros en la etapa anterior del proceso hidrolítico.

$c_{m_{avg}}^i$ : es el valor promedio de concentración de monómeros en la etapa actual de degradación.

$x_c$  representa una expresión algebraica de la cristalinidad a partir de la ecuación de Avrami:

$$x_c = x_{c_0} + (x_c^{max} - x_{c_0}) \cdot (1 - e^{-k \cdot t + v \cdot C_m}) \quad (11-43)$$

Por otra parte, la porosidad debida a la erosión superficial se expresa como la relación entre el volumen hueco en el espacio sólido:

$$p_{surface} = \frac{V_{cell} - V_{scaffold}}{V_{cell}} \cdot 100 \quad (11-44)$$

En las expresiones de  $p_1$  y  $p_2$  los términos  $\frac{2}{1 + e^{\gamma(x_c - x_{c_0})}}$  y  $\left( \frac{c_{m_{avg}}^{i-1}}{c_{m_{avg}}^i + \epsilon} \right)$  se pueden renombrar como  $F_{Crist}$  y  $F_{Monom}$  respectivamente. Así nos queda que:



$$p_1 = p_{surface} \cdot \left(1 - \left(1 - \frac{\chi}{F_{cris}}\right) F_{monom}^{\psi}\right) \quad (11-45)$$

$$p_2 = 100 - p_{surface} \cdot \beta \cdot e^{-\alpha F_{cris} F_{monom}}$$

## DATOS Y PARÁMETROS DEL MODELO

Los valores utilizados de los parámetros del modelo son:

### 1) MODELO POROSIDAD:

$$\begin{aligned} \alpha &= 1,7 & \psi &= 1,5 \\ t_{trans} &= 3,1 & \chi &= 1,9 \\ v &= 0,5 & \lambda &= 0,45 \\ \phi &= 2,9 & \beta &= 0,055 \\ \omega &= 10 \\ \gamma &= 0,00001 \end{aligned} \quad (11-46)$$

### 2) REACCIÓN DIFUSIÓN:

$$\begin{aligned} k_1 &= 0.03 \\ k_2 &= 2 \cdot 10^{1.5} \\ D_0 &= 0.5 \\ \alpha &= 4.5 \\ C_{e_0} &= 1 \\ C_{m_0} &= 0 \end{aligned} \quad (11-47)$$

### 3) MODELO EROSIVO PROBABILÍSTICO:

$$B = 6.5 \cdot 10^{-1} \quad (11-48)$$

#### 11.4.6. RESULTADOS

### DISTANCIA SIGNADA PHI

En la primera gráfica (Figura 11-23) se muestra la vista en sección transversal de la distancia signada de la región sólida de la celda unitaria:

Con el cálculo de la distancia signada respecto de la interfase tenemos una distribución en la que se diferencian claramente la región del material polimérico, coloreada en rojo, y la región azul que representa el dominio ficticio.

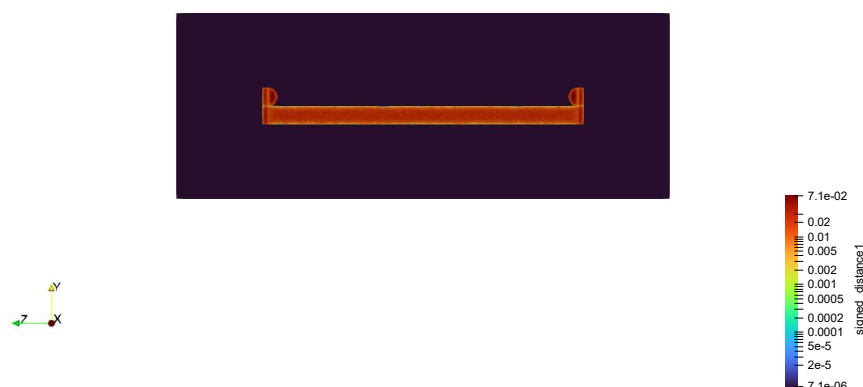


Figura 11-23. Representación de la interfase calculada usando el método de los dominios ficticios (80% porosidad).

De igual forma, se observa una partición clara entre el espacio líquido que rodea a la pieza y el resto del conjunto ficticio (Figura 11-24):



Figura 11-24. Representación de la región líquida de la celda unitaria del dominio ficticio.

## EVOLUCIÓN DE LA POROSIDAD BAJO CONDICIONES DINÁMICAS.

En relación con la porosidad tenemos un ajuste cualitativamente similar al de Agrawal, a partir de las 2 semanas aproximadamente:

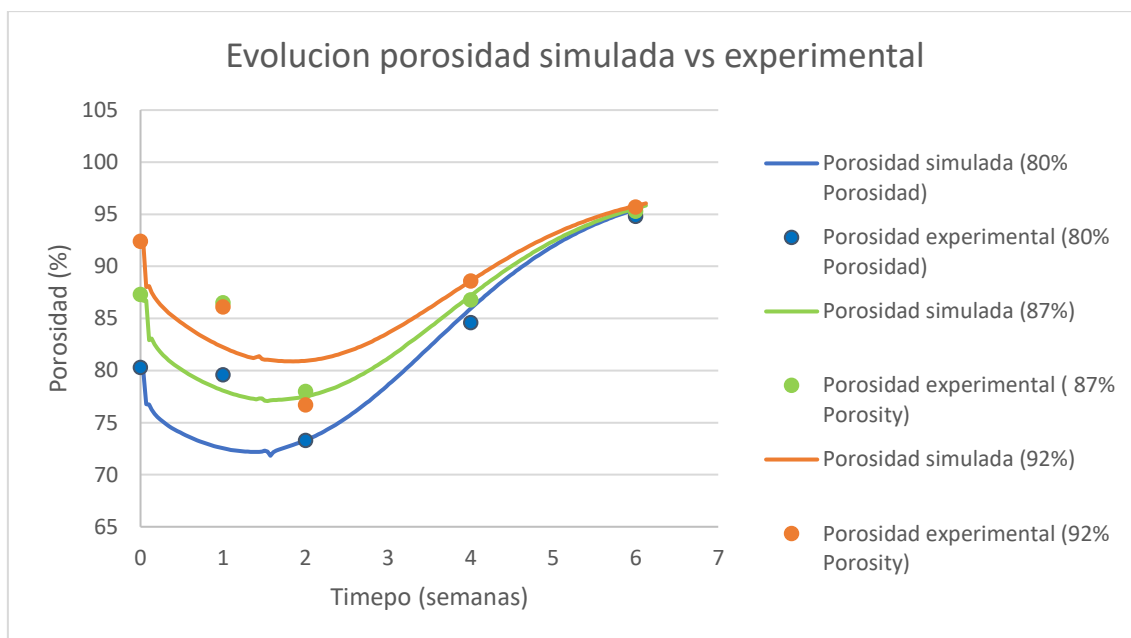


Figura 11-25. Comparativa evolución porosidad simulada vs experimental.

En la Figura 11-25 se observa las líneas de tendencia de los valores simulados para las tres configuraciones (80% porosidad, 87% de porosidad y 92% de porosidad). Los puntos representan los valores experimentales del modelo de Agrawal. En general, se observa un buen ajuste, con la salvedad de que en la primera semana la disminución es más acusada en el modelo simulado que lo que reflejan los valores experimentales. Asimismo, los resultados obtenidos en el modelo para una porosidad inicial de 92% presentan una diferencia del 4% con respecto al valor de referencia en la segunda semana. A la vista de los resultados del modelo, se observa una tendencia cualitativamente similar a los datos de referencia, con una **fase inicial de disminución de la porosidad**, seguida por una fase de recuperación, lo que sugiere que los principales mecanismos de degradación están bien representados.

## 12. CONCLUSIONES

Durante el desarrollo de este trabajo se ha profundizado en diferentes metodologías para la degradación polimérica, de menor a mayor complejidad. En un inicio, el objetivo del trabajo se centraba en el cálculo de la pérdida de masa debida a la erosión superficial del material bajo condiciones dinámicas *in vitro*. Las consideraciones adoptadas para el modelado de la erosión eran puramente geométricas, centrándose únicamente en la distribución del campo de tensiones tangenciales, como causante principal del proceso. Sin embargo, a nivel físico el comportamiento dinámico del medio no juega un papel principal en el proceso degradativo, debido al arrastre de los subproductos de la degradación, que actúan como agentes principales en el proceso degradativo/erosivo del material [28], [30].

Para solventar esta cuestión, se procedió a realizar un estudio estocástico bajo condiciones estáticas que permitieran predecir el comportamiento del material, tomando para ello el grado de exposición al medio de cada elemento numérico como variable de decisión para determinar, mediante un criterio probabilístico, si un elemento superficial de la matriz polimérica se disuelve o no. En este respecto, este segundo modelo tomaba los valores de pérdida de masa de referencia (a partir de resultados experimentales), y, a partir de un factor de ajuste, denominado factor de amplificación, se lograba alcanzar los valores objetivo en pocas iteraciones. Sin embargo, al basarse en datos experimentales, esta metodología suponía una limitación importante a la hora de generalizar su aplicación para otras configuraciones y materiales. En relación con esto, se almacenaron, en cada etapa, las relaciones entre el volumen extraído y el volumen de elementos superficiales, para una geometría base (en este caso un filamento). Estos valores se usaron de referencia para simular el proceso en un scaffold, usando la relación del volumen removido y el volumen superficial de los resultados obtenidos en el filamento como referencia. Sin embargo, no se observó un patrón replicable entre ambos modelos (el inicial y el extrapolado). Además del uso de valores experimentales de referencia, otra de las razones por la cual este modelo no es del todo generalizable, es la ausencia de variables descriptivas del comportamiento intrínseco del material. Por este motivo, esta aproximación no resulta eficaz a la hora de estimar el comportamiento degradativo del material ni la evolución de las propiedades intrínsecas del material, que, en realidad, juegan un papel más significativo en el proceso.

Ante ello, se consideró un criterio fenomenológico para predecir la variación de las propiedades internas del material. En este caso, se asumió el modelo propuesto por Wang et al. para la estimación temporal de las propiedades intrínsecas, concretamente de peso molecular y concentración de monómeros. Dicho modelo, que se basa en la segunda Ley de Fick, se trata de un modelo para cuantificar el proceso de degradación de los materiales

poliméricos en el que se asume que la formación de subproductos de la degradación se realiza en forma de monómeros, siendo los oligómeros los productos que generalmente se generan del proceso hidrolítico. Además, no se asume el proceso de recristalización del material polimérico [96].

Paralelamente, en relación con la caracterización numérica del comportamiento mecánico del material en las sucesivas etapas del proceso degradativo, se partía inicialmente de las configuraciones del scaffold en las sucesivas etapas del proceso erosivo-enzimático. Tomando los valores de referencia medidos en un filamento en las diferentes etapas del proceso, se realizaban sucesivas simulaciones mecánicas usando ABAQUS y se registraba el valor de módulo de elasticidad global. No obstante, los valores obtenidos en la simulación y en el ensayo experimental no guardaban una similitud razonable. Uno de los principales motivos que se aducen, al igual que en el modelo dinámico y estocástico descrito anteriormente, es el empleo de un enfoque puramente geométrico en la modelización del comportamiento, desatendiendo la aparición de variables descriptivas del comportamiento interno del material.

A raíz de esto, se aplicó un modelo probabilístico para la cuantificación de la rotura de cadenas [106], en el que se obtiene el valor promedio del peso molecular. Cuantificando aquellas cadenas cuya longitud quede por encima del umbral establecido, y usando un criterio entrópico [29], [40], se obtiene una relación entre los valores normalizados del peso molecular y el módulo de elasticidad. Usando el modelo hidrolítico descrito previamente, y aprovechando la equivalencia entre el valor estandarizado del peso molecular y de la concentración de ésteres [30], se obtiene una distribución del módulo de elasticidad en diferentes etapas de la degradación [118]. Esta técnica solo se aplicó a una celda unitaria asumiendo únicamente condiciones hidrolíticas.

Por último, se implementó el modelo integrado propuesto anteriormente, que en esencia se trata de una técnica que sirve para predecir el comportamiento de un material polimérico cuando se encuentra sometido al mismo tiempo a una degradación hidrolítica y a una erosión superficial. En este modelo, el material sufre un proceso de degradación hidrolítica hasta llegar a un punto en el cual los oligómeros resultantes de la degradación se liberan al medio. Es en este momento en el cual el material comienza a perder masa efectiva, lo que se traduce en un proceso erosivo que ocasiona un cambio de forma. A partir de entonces, se producen simultáneamente los fenómenos degradativos (degradación interna asociada a la pérdida de masa molecular) y erosivos (pérdida efectiva de masa y volumen). En relación con el proceso erosivo, se trata de un proceso de erosión superficial en el que, dependiendo del grado de exposición del polímero al medio basándonos en un criterio de vecindades (autómatas celulares), se cuantifica la cantidad de material liberada al medio a través de la superficie.

Para aplicar este modelo integrado, en primer lugar, se valoró trabajar directamente con la geometría discretizada de manera uniforme, y combinar la implementación numérica de las ecuaciones hidrolíticas de reacción difusión con el modelo estocástico, una vez sabido que los subproductos de la reacción hidrolítica se difunden en el medio, condición en la cual la erosión se inicia. Esto permitiría simular ambos procesos de forma conjunta. Sin embargo, esto ocasiona una creciente inestabilidad numérica, especialmente en aquellas regiones donde el frente erosivo está más desarrollado, lo que deriva en último término en un error de cómputo. Para solventar esto, se planteó el suavizado, por medio de un refinamiento local de malla, a aquellas regiones que contengan elementos susceptibles de ser erosionados. Los resultados obtenidos no fueron nada alentadores, puesto que el modelo era únicamente capaz de realizar una o dos etapas sucesivas, empeorando el resultado del caso anterior.

Después de varias implementaciones que permitían definir los fenómenos anteriormente descritos, se llegó a la conclusión de que la implementación más estable y eficiente numéricamente era la basada en el método de dominios ficticios. A nivel numérico, una de las ventajas que presenta este modelo es la robustez que tiene a la hora de poder trabajar con geometrías de mayor complejidad, ya que no se trabaja directamente con dicha geometría a la hora de resolver las ecuaciones del proceso, sino que se encuentra contenida dentro de un dominio simplificado. Asimismo, incluye algunos parámetros que pueden jugar un papel en el proceso de degradación como pueden ser el peso molecular y la concentración de subproductos (monómeros). Con este modelo integrado se realizaron diferentes simulaciones y minuciosos ajustes en el modelo hasta llegar a disponer de un modelo con suficiente estabilidad computacional, y con potencial para predecir el comportamiento tanto a nivel macroscópico (pérdida volumétrica y masa) como a nivel interno (pérdida de peso molecular y propiedades mecánicas).

En definitiva, tras el desarrollo de esta tesis se llega a la conclusión de que el mejor modelo conseguido se corresponde con el modelo integrado (degradación hidrolítica y erosión superficial), el cual trabaja con una celda unitaria del scaffold (simplificando así el modelo hasta la unidad fundamental), y al que se le aplica una geometría extendida (dominios ficticios) para conseguir la estabilidad numérica. Con ello, se consigue una herramienta de simulación para determinar la geometría degradada de un scaffold polimérico y bioabsorbible en cualquier instante del tiempo (Figura 12-1). Además, los resultados del modelo de degradación hidrolítica pueden ser extraídos y usados para determinar la distribución del módulo de elasticidad del material en distintos instantes de tiempo. Por tanto, las salidas o resultados de estos modelos (geometría degradada y módulo de elasticidad del material en cada instante) pueden ser datos muy útiles para estimar el comportamiento mecánico del scaffold mediante simulaciones numéricas. Esto lo convierte en una herramienta muy interesante para la optimización del diseño, en la que se busca conseguir una estructura porosa que no solo soporte las cargas mecánicas correspondientes en los momentos más críticos de la regeneración del tejido, sino que además tenga una tasa de degradación en consonancia con la tasa de regeneración del tejido nativo.

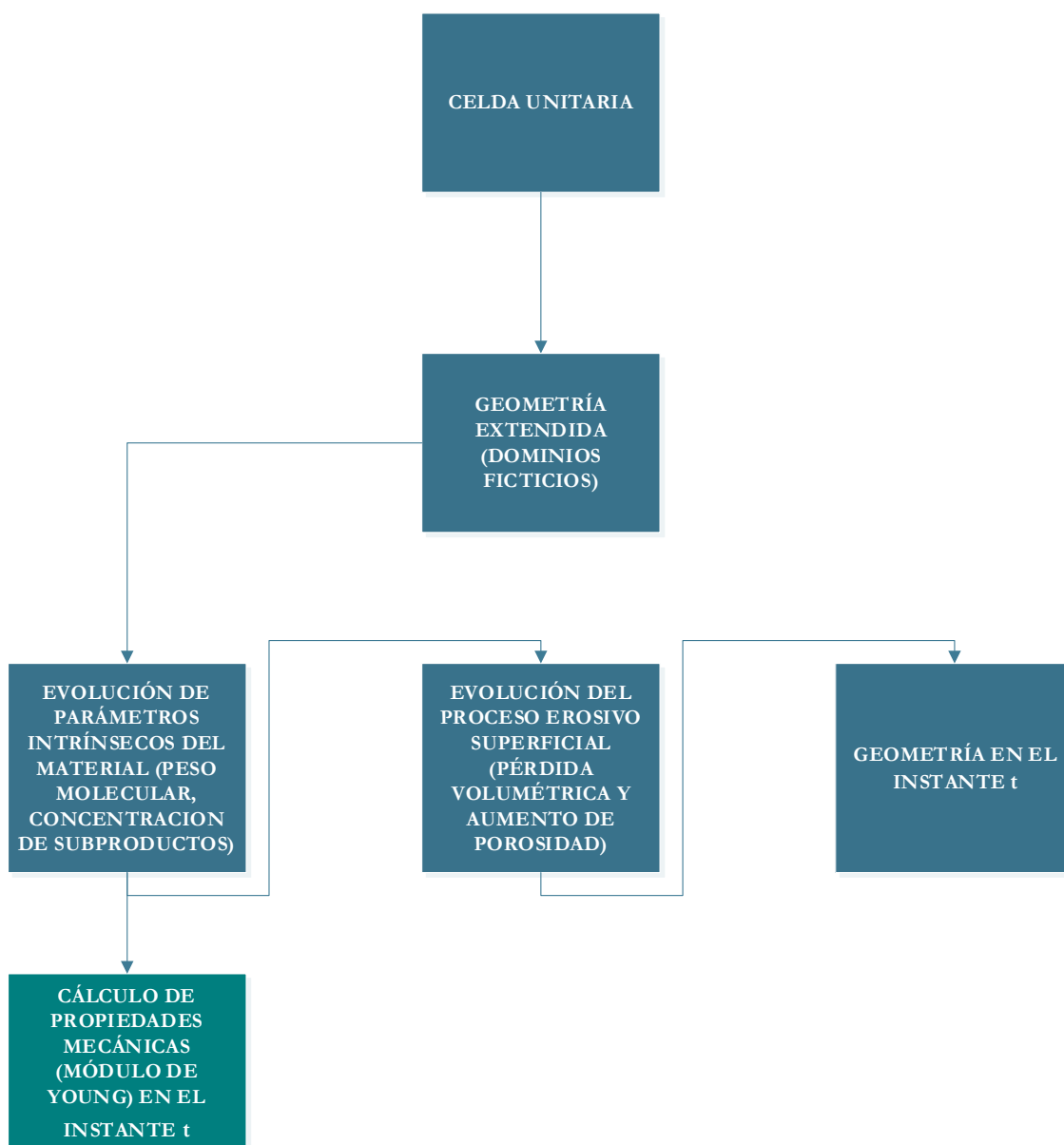


Figura 12-1. Esquema general del modelo integrado finalmente adoptado.

En base a todo lo expuesto anteriormente, cabría destacar que la principal contribución de esta tesis es precisamente este modelo integrado explicado anteriormente. Además, dentro de ese modelo y desde el punto de vista numérico, también cabe destacar la formulación e implementación con el método de elementos finitos, linealizando el problema no lineal de reacción difusión y aplicando un esquema matricial que aporta mayor modularidad y eficiencia computacional. Asimismo, también habría que recalcar la implementación en dominios ficticios para dar más estabilidad al modelo integrado, sobre todo en su fase erosiva. Otra contribución que cabe destacar ha sido la caracterización no lineal de la pérdida volumétrica debida a la erosión superficial a través de la porosidad. En contraposición con otros modelos propuestos hasta la fecha según el estado del arte, se ha tenido en cuenta el



efecto de parámetros como la cristalinidad en el descenso de la porosidad durante el proceso combinado degradativo y erosivo.

## 13. LÍNEAS FUTURAS

Los resultados obtenidos en el modelo integrado reflejan un ajuste razonable a lo largo del proceso experimental, con una fase de descenso de porosidad seguido de un tramo ascendente en el que la pieza aumenta el valor porcentual de dicho parámetro. Sin embargo, en el tramo inicial el descenso de la porosidad calculado por el modelo propuesto es mucho más pronunciado en las fases iniciales si se compara con los valores obtenidos por Agrawal [28]. En este respecto, sería interesante obtener un modelo más refinado que replique el comportamiento degradativo-erosivo del material, sobre todo en las fases iniciales del proceso. Una opción de realizar esto es mediante el uso de una función de porosidad inicial que sea capaz de capturar aquellas fases más atenuadas del comportamiento inicial de la pieza. Esto, unido al uso de funciones de transición, se lograría una expresión única que representaría, de manera más fidedigna, la evolución de porosidad a lo largo del proceso.

Asimismo, sería preciso hacer una validación más exhaustiva del método, en términos del proceso degradativo, para validar el modelo con diferentes configuraciones geométricas, ajustando todas las variables que intervienen en el proceso y contrastando los resultados con nuevos estudios experimentales, con el fin de lograr una mayor bondad del modelo con diferentes geometrías, materiales, condiciones in vitro, etc.

En relación con esto, el creciente impacto de la inteligencia artificial permitiría validar la metodología propuesta usando un método más automatizado. Pero esto requiere de una base de datos lo suficientemente grande para inicializar el entrenamiento. Por ello, se podría valorar el uso de modelos de Machine Learning que permitan entrenar inicialmente el modelo a partir de datos experimentales de ensayos de degradación, y después se usarían esos modelos entrenados para predecir el proceso degradativo del material.

Otro de los puntos que se podrían tratar en el futuro de este trabajo es la obtención de las propiedades mecánicas del scaffold evaluado en diferentes etapas del proceso erosivo. Para ello se integrarían los modelos que relacionan el módulo de elasticidad con el peso molecular obtenido en la sección 10.3 y se verían los efectos del cambio de propiedades mecánicas durante el proceso erosivo. Dentro de la literatura existente, Vieira et al. [103] desarrolló un modelo que permite estudiar la evolución de las propiedades mecánicas debidas a la erosión. Sin embargo, este modelo no considera el efecto del movimiento del flujo alrededor del material y su influencia en el proceso erosivo.

En relación con esto, sería interesante la obtención, en diferentes instantes de tiempo, de un modelo que calcule el módulo de elasticidad equivalente, mediante una simulación de

compresión mediante elementos finitos, usando las propiedades mecánicas y la geometría estimada en diferentes instantes de tiempo que se obtienen del modelo integrado previamente descrito.

En última instancia, sería ideal la obtención de forma simultánea de la evolución del proceso de regeneración del tejido y la evolución de las propiedades mecánicas del dispositivo mientras se encuentra insertado en el tejido. Este tipo de herramientas numéricas para estimar este comportamiento sería bastante útil para llevar a cabo la optimización del diseño de scaffolds en términos topológicos y en términos de material, para lograr un dispositivo que sea capaz de mantener la capacidad portante en todo momento, disminuyendo progresivamente su capacidad portante en consonancia con la regeneración del tejido dañado.

## 14. BIBLIOGRAFÍA

- [1] Y. Wang, J. Pan, X. Han, C. Sinka, y L. Ding, «A phenomenological model for the degradation of biodegradable polymers», *Biomaterials*, vol. 29, n.º 23, pp. 3393-3401, ago. 2008, doi: 10.1016/j.biomaterials.2008.04.042.
- [2] R. Skalak y C. F. Fox, «Tissue engineering: proceedings of a workshop held at Granlibakken, Lake Tahoe, California, February 26-29, 1988», (*No Title*), 1988.
- [3] M. E. Furth y A. Atala, «Chapter 6 - Tissue Engineering: Future Perspectives», en *Principles of Tissue Engineering (Fourth Edition)*, R. Lanza, R. Langer, y J. Vacanti, Eds., Boston: Academic Press, 2014, pp. 83-123. doi: 10.1016/B978-0-12-398358-9.00006-9.
- [4] A. E. Eldeeb, S. Salah, y N. A. Elkasabgy, «Biomaterials for Tissue Engineering Applications and Current Updates in the Field: A Comprehensive Review», *AAPS PharmSciTech*, vol. 23, n.º 7, p. 267, sep. 2022, doi: 10.1208/s12249-022-02419-1.
- [5] C. Vinatier y J. Guicheux, «Cartilage tissue engineering: From biomaterials and stem cells to osteoarthritis treatments», *Annals of Physical and Rehabilitation Medicine*, vol. 59, n.º 3, pp. 139-144, jun. 2016, doi: 10.1016/j.rehab.2016.03.002.
- [6] R. Sakaguchi, J. Ferracane, y J. Powers, Eds., «Chapter 16 - Tissue Engineering», en *Craig's Restorative Dental Materials (Fourteenth Edition)*, Philadelphia: Elsevier, 2019, pp. 313-326. doi: 10.1016/B978-0-323-47821-2.00016-0.
- [7] G. Erkizia, A. Rainer, E. M. D. Juan-Pardo, y J. Aldazabal, «Computer Simulation of Scaffold Degradation», *J. Phys.: Conf. Ser.*, vol. 252, p. 012004, nov. 2010, doi: 10.1088/1742-6596/252/1/012004.
- [8] A. Standard y others, «Standard terminology for additive manufacturing technologies», *ASTM International F2792-12a*, vol. 46, pp. 10918-10928, 2012.
- [9] M. K. Heljak, W. Świążkowski, C. X. F. Lam, D. W. Hutmacher, y K. J. Kurzydłowski, «Evolutionary design of bone scaffolds with reference to material selection», *International Journal for Numerical Methods in Biomedical Engineering*, vol. 28, n.º 6-7, pp. 789-800, 2012, doi: 10.1002/cnm.2487.
- [10] S. M. Giannitelli, D. Accoto, M. Trombetta, y A. Rainer, «Current trends in the design of scaffolds for computer-aided tissue engineering», *Acta Biomaterialia*, vol. 10, n.º 2, pp. 580-594, feb. 2014, doi: 10.1016/j.actbio.2013.10.024.
- [11] M. P. Nikolova y M. S. Chavali, «Recent advances in biomaterials for 3D scaffolds: A review», *Bioactive Materials*, vol. 4, pp. 271-292, dic. 2019, doi: 10.1016/j.bioactmat.2019.10.005.
- [12] T. Ghassemi, A. Shahroodi, M. H. Ebrahimzadeh, A. Mousavian, J. Movaffagh, y A. Moradi, «Current Concepts in Scaffolding for Bone Tissue Engineering», *Arch Bone Jt Surg*, vol. 6, n.º 2, pp. 90-99, mar. 2018.
- [13] A. Moghaddam *et al.*, «Recent trends in bone tissue engineering: a review of materials, methods, and structures», *Biomed Mater*, vol. 19, n.º 4, may 2024, doi: 10.1088/1748-605X/ad407d.
- [14] M. Modrák, M. Trebuňová, A. F. Balogová, R. Hudák, y J. Živčák, «Biodegradable Materials for Tissue Engineering: Development, Classification and Current Applications», *Journal of Functional Biomaterials*, vol. 14, n.º 3, Art. n.º 3, mar. 2023, doi: 10.3390/jfb14030159.

- [15] S. Singh, C. Chittasupho, B. G. Prajapati, y A. S. Chandel, «Editorial: Biodegradable polymeric materials in tissue engineering and their application in drug delivery», *Front. Bioeng. Biotechnol.*, vol. 11, sep. 2023, doi: 10.3389/fbioe.2023.1296119.
- [16] A. Göpferich y J. Tessmar, «Polyanhydride degradation and erosion», *Advanced Drug Delivery Reviews*, vol. 54, n.º 7, pp. 911-931, oct. 2002, doi: 10.1016/S0169-409X(02)00051-0.
- [17] F. V. Burkensroda, L. Schedl, y A. Göpferich, «Why degradable polymers undergo surface erosion or bulk erosion», *Biomaterials*, vol. 23, n.º 21, pp. 4221-4231, nov. 2002, doi: 10.1016/S0142-9612(02)00170-9.
- [18] N. F. Zaaba y M. Jaafar, «A review on degradation mechanisms of polylactic acid: Hydrolytic, photodegradative, microbial, and enzymatic degradation», *Polymer Engineering & Science*, vol. 60, n.º 9, pp. 2061-2075, 2020, doi: 10.1002/pen.25511.
- [19] B. Laycock *et al.*, «Lifetime prediction of biodegradable polymers», *Progress in Polymer Science*, vol. 71, pp. 144-189, ago. 2017, doi: 10.1016/j.progpolymsci.2017.02.004.
- [20] K. Zygorakis, «Development and temporal evolution of erosion fronts in bioerodible controlled release devices», *Chemical Engineering Science*, vol. 45, n.º 8, pp. 2359-2366, ene. 1990, doi: 10.1016/0009-2509(90)80116-V.
- [21] A. Göpferich y R. Langer, «Modeling of polymer erosion», *Macromolecules*, vol. 26, n.º 16, pp. 4105-4112, ago. 1993, doi: 10.1021/ma00068a006.
- [22] A. Göpferich, «Mechanisms of polymer degradation and erosion», *Biomaterials*, vol. 17, n.º 2, Art. n.º 2, ene. 1996, doi: 10.1016/0142-9612(96)85755-3.
- [23] A. Göpferich y R. Langer, «Modeling of polymer erosion in three dimensions: Rotationally symmetric devices», *AIChE Journal*, vol. 41, n.º 10, Art. n.º 10, 1995, doi: 10.1002/aic.690411012.
- [24] J. D. Culter, J. L. Zakin, y G. K. Patterson, «Mechanical degradation of dilute solutions of high polymers in capillary tube flow», *Journal of Applied Polymer Science*, vol. 19, n.º 12, pp. 3235-3240, 1975, doi: 10.1002/app.1975.070191210.
- [25] J. Abdelfatah Ndioubnane, R. Paz Hernández, M. D. Monzón Verón, y G. Winter Althaus, «Modelisation and simulation of the degradation process of scaffolds developed by additive manufacturing and biomaterials», 2019, Accedido: 23 de febrero de 2025. [En línea]. Disponible en: <https://accedacris.ulpgc.es/jspui/handle/10553/113678>
- [26] H. Basri *et al.*, «The effect of flow rate characteristic on biodegradation of bone scaffold», *Malaysian Journal of Fundamental and Applied Sciences*, vol. Special Issue on Medical Device and Technology (IMEDITECH 2017), pp. 546-552, dic. 2017, doi: 10.11113/mjfas.v13n4-2.843.
- [27] M. Schellekens, S. Zhou, J. Cadman, W. Li, R. Appleyard, y Q. Li, «Design Optimization of Scaffold Microstructures Using Wall Shear Stress Criterion Towards Regulated Flow-Induced Erosion», *Journal of biomechanical engineering*, vol. 133, p. 081008, ago. 2011, doi: 10.1115/1.4004918.
- [28] C. M. Agrawal, J. S. McKinney, D. Lanctot, y K. A. Athanasiou, «Effects of fluid flow on the in vitro degradation kinetics of biodegradable scaffolds for tissue engineering», *Biomaterials*, vol. 21, n.º 23, pp. 2443-2452, dic. 2000, doi: 10.1016/S0142-9612(00)00112-5.
- [29] I. M. Ward y J. Sweeney, *Mechanical Properties of Solid Polymers: Third Edition*. Chichester, UK: John Wiley & Sons, Ltd, 2012. doi: 10.1002/9781119967125.
- [30] M. K. Heljak, W. Swieszkowski, y K. J. Kurzydłowski, «Modeling of the degradation kinetics of biodegradable scaffolds: The effects of the environmental conditions», *Journal of Applied Polymer Science*, vol. 131, n.º 11, 2014, doi: 10.1002/app.40280.

- [31] F. von Burkersroda, L. Schedl, y A. Göpferich, «Why degradable polymers undergo surface erosion or bulk erosion», *Biomaterials*, vol. 23, n.º 21, pp. 4221-4231, nov. 2002, doi: 10.1016/S0142-9612(02)00170-9.
- [32] B. Laycock *et al.*, «Lifetime prediction of biodegradable polymers», *Progress in Polymer Science*, vol. 71, pp. 144-189, ago. 2017, doi: 10.1016/j.progpolymsci.2017.02.004.
- [33] K. E. Uhrich, S. M. Cannizzaro, R. S. Langer, y K. M. Shakesheff, «Polymeric Systems for Controlled Drug Release», *Chem. Rev.*, vol. 99, n.º 11, pp. 3181-3198, nov. 1999, doi: 10.1021/cr940351u.
- [34] J. Medina, Y. Roche, O. Maldonado, J. C. Hernández, y C. Zapata, «Degradación hidrolítica y biodegradación de mezclas binarias de ácido poliláctico (PLA) con residuos plásticos», *Revista INGENIERÍA UC*, vol. 25, n.º 2, 2018, Accedido: 29 de agosto de 2024. [En línea]. Disponible en: <https://www.redalyc.org/journal/707/70757669013/html/>
- [35] R. R. A. Silva, C. S. Marques, T. R. Arruda, S. C. Teixeira, y T. V. de Oliveira, «Biodegradation of Polymers: Stages, Measurement, Standards and Prospects», *Macromol*, vol. 3, n.º 2, Art. n.º 2, jun. 2023, doi: 10.3390/macromol3020023.
- [36] C. Yang, X. Xing, Z. Li, y S. Zhang, «A Comprehensive Review on Water Diffusion in Polymers Focusing on the Polymer–Metal Interface Combination», *Polymers*, vol. 12, n.º 1, Art. n.º 1, ene. 2020, doi: 10.3390/polym12010138.
- [37] J. Trifol, D. Plackett, P. Szabo, A. E. Daugaard, y M. Giacinti Baschetti, «Effect of Crystallinity on Water Vapor Sorption, Diffusion, and Permeation of PLA-Based Nanocomposites», *ACS Omega*, vol. 5, n.º 25, pp. 15362-15369, jun. 2020, doi: 10.1021/acsomega.0c01468.
- [38] C. K. Sackett y B. Narasimhan, «Mathematical modeling of polymer erosion: Consequences for drug delivery», *International Journal of Pharmaceutics*, vol. 418, n.º 1, pp. 104-114, oct. 2011, doi: 10.1016/j.ijpharm.2010.11.048.
- [39] Y. Chen, S. Zhou, y Q. Li, «Mathematical modeling of degradation for bulk-erosive polymers: Applications in tissue engineering scaffolds and drug delivery systems», *Acta Biomaterialia*, vol. 7, n.º 3, pp. 1140-1149, mar. 2011, doi: 10.1016/j.actbio.2010.09.038.
- [40] Y. Wang, X. Han, J. Pan, y C. Sinka, «An entropy spring model for the Young's modulus change of biodegradable polymers during biodegradation», *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 3, n.º 1, pp. 14-21, ene. 2010, doi: 10.1016/j.jmbbm.2009.02.003.
- [41] R. N. Shirazi, W. Ronan, Y. Rochev, y P. McHugh, «Modelling the degradation and elastic properties of poly(lactic-co-glycolic acid) films and regular open-cell tissue engineering scaffolds», *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 54, pp. 48-59, feb. 2016, doi: 10.1016/j.jmbbm.2015.08.030.
- [42] J. Siepmann, N. Faisant, y J.-P. Benoit, «A New Mathematical Model Quantifying Drug Release from Bioerodible Microparticles Using Monte Carlo Simulations», *Pharm Res*, vol. 19, n.º 12, Art. n.º 12, dic. 2002, doi: 10.1023/A:1021457911533.
- [43] D. O. Cooney, «Effect of geometry on the dissolution of pharmaceutical tablets and other solids: Surface detachment kinetics controlling», *AIChE Journal*, vol. 18, n.º 2, pp. 446-449, 1972, doi: 10.1002/aic.690180234.
- [44] H. B. HOPFENBERG, «Controlled Release from Erodible Slabs, Cylinders, and Spheres», en *Controlled Release Polymeric Formulations*, vol. 33, 0 vols., en ACS Symposium Series, no. 33, vol. 33., AMERICAN CHEMICAL SOCIETY, 1976, pp. 26-32. doi: 10.1021/bk-1976-0033.ch003.

- [45] K. Sevim y J. Pan, «A model for hydrolytic degradation and erosion of biodegradable polymers», *Acta Biomaterialia*, vol. 66, pp. 192-199, ene. 2018, doi: 10.1016/j.actbio.2017.11.023.
- [46] J. Abdelfatah, R. Paz, M. E. Alemán-Domínguez, M. Monzón, R. Donate, y G. Winter, «Experimental Analysis of the Enzymatic Degradation of Polycaprolactone: Microcrystalline Cellulose Composites and Numerical Method for the Prediction of the Degraded Geometry», *Materials*, vol. 14, n.º 9, Art. n.º 9, ene. 2021, doi: 10.3390/ma14092460.
- [47] I. A. Magomedov y Z. S. Sebaeva, «Comparative study of finite element analysis software packages», *J. Phys.: Conf. Ser.*, vol. 1515, n.º 3, p. 032073, abr. 2020, doi: 10.1088/1742-6596/1515/3/032073.
- [48] F. Hecht, «New development in freefem++», *Journal of Numerical Mathematics*, vol. 20, n.º 3-4, pp. 251-266, dic. 2012, doi: 10.1515/jnum-2012-0013.
- [49] H. G. Weller, G. Tabor, H. Jasak, y C. Fureby, «A tensorial approach to computational continuum mechanics using object-oriented techniques», *Computers in Physics*, vol. 12, n.º 6, pp. 620-631, nov. 1998, doi: 10.1063/1.168744.
- [50] C. Geuzaine y J.-F. Remacle, «Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities», *International Journal for Numerical Methods in Engineering*, vol. 79, n.º 11, pp. 1309-1331, 2009, doi: 10.1002/nme.2579.
- [51] J. Remacle, N. Chevaugneon, É. Marchandise, y C. Geuzaine, «Efficient visualization of high-order finite elements», *Numerical Meth Engineering*, vol. 69, n.º 4, pp. 750-771, ene. 2007, doi: 10.1002/nme.1787.
- [52] «International Encyclopedia of Education», ScienceDirect. Accedido: 19 de marzo de 2024. [En línea]. Disponible en: <http://www.sciencedirect.com:5070/referencework/9780080448947/international-encyclopedia-of-education>
- [53] «Computational Materials Science: The Simulation of Materials, Microstructures and Properties | Wiley», Wiley.com. Accedido: 21 de julio de 2022. [En línea]. Disponible en: <https://www.wiley.com/en-us/Computational+Materials+Science%3A+The+Simulation+of+Materials%2C+Microstructures+and+Properties-p-9783527601943>
- [54] M. Gilli, D. Maringer, y E. Schumann, *Numerical methods and optimization in finance*, Second edition. London ; San Diego, CA: Academic Press is an imprint of Elsevier, 2019.
- [55] J. von Neumann, «The general and logical theory of automata», en *Cerebral Mechanisms in Behaviour*, L. A. Jeffress, Ed., Wiley, 1951.
- [56] M. Games, «The fantastic combinations of John Conway's new solitaire game "life" by Martin Gardner», *Scientific American*, vol. 223, pp. 120-123, 1970.
- [57] C. Börgers, «Domain imbedding methods for the Stokes equations», *Numer. Math.*, vol. 57, dic. 1990, doi: 10.1007/BF01386422.
- [58] Y. Chen, S. Zhou, J. Cadman, y Q. Li, «Design of cellular porous biomaterials for wall shear stress criterion», *Biotechnology and Bioengineering*, vol. 107, n.º 4, pp. 737-746, 2010, doi: 10.1002/bit.22842.
- [59] M. F. Shockley y A. H. Muliana, «Modeling temporal and spatial changes during hydrolytic degradation and erosion in biodegradable polymers», *Polymer Degradation and Stability*, vol. 180, p. 109298, oct. 2020, doi: 10.1016/j.polymdegradstab.2020.109298.
- [60] «History of SALOME», SALOME PLATFORM. Accedido: 10 de febrero de 2025. [En línea]. Disponible en: [https://www.salome-platform.org/?page\\_id=676](https://www.salome-platform.org/?page_id=676)
- [61] «Functionalities», SALOME PLATFORM. Accedido: 10 de febrero de 2025. [En línea]. Disponible en: [https://www.salome-platform.org/?page\\_id=23](https://www.salome-platform.org/?page_id=23)



- [62] «Scientific visualisation», SALOME PLATFORM. Accedido: 10 de febrero de 2025. [En línea]. Disponible en: [https://www.salome-platform.org/?page\\_id=377](https://www.salome-platform.org/?page_id=377)
- [63] J. Ahrens, B. Geveci, y C. Law, «36 - ParaView: An End-User Tool for Large-Data Visualization», en *Visualization Handbook*, C. D. Hansen y C. R. Johnson, Eds., Burlington: Butterworth-Heinemann, 2005, pp. 717-731. doi: 10.1016/B978-012387582-2/50038-1.
- [64] E. Marchandise, J.-F. Remacle, y C. Geuzaine, «Optimal parametrizations for surface remeshing», *Engineering with Computers*, vol. 30, n.º 3, pp. 383-402, jul. 2014, doi: 10.1007/s00366-012-0309-3.
- [65] F. Hecht, «BAMG: bidimensional anisotropic mesh generator», *User Guide*. INRIA, Rocquencourt, vol. 17, 1998.
- [66] G. Miller, D. Talmor, S.-H. Teng, N. Walkington, y H. Wang, «Control Volume Meshes Using Sphere Packing: Generation, Refinement and Coarsening», en *Proceedings of the Fifth International Meshing Roundtable*, 1996, pp. 47-61.
- [67] H. Si, *TetGen: A Quality Tetrahedral Mesh Generator and 3D Delaunay Triangulator*. Weierstrass Institute for Applied Analysis and Stochastics (WIAS), 2006.
- [68] H. Si, «TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator», *ACM Trans. Math. Softw.*, vol. 41, n.º 2, pp. 1-36, feb. 2015, doi: 10.1145/2629697.
- [69] C. Dobrzynski y P. Frey, «Anisotropic Delaunay Mesh Adaptation for Unsteady Simulations», en *Proceedings of the 17th International Meshing Roundtable*, R. V. Garimella, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 177-194. doi: 10.1007/978-3-540-87921-3\_11.
- [70] G. Balarac *et al.*, «Tetrahedral remeshing in the context of large-scale numerical simulation and high performance computing», *MathematicS In Action*, vol. 11, n.º 1, pp. 129-164, sep. 2022, doi: 10.5802/msia.22.
- [71] R. J. McCoy y F. J. O'Brien, «Influence of Shear Stress in Perfusion Bioreactor Cultures for the Development of Three-Dimensional Bone Tissue Constructs: A Review», *Tissue Engineering Part B: Reviews*, vol. 16, n.º 6, pp. 587-601, dic. 2010, doi: 10.1089/ten.teb.2010.0370.
- [72] T. Pires, J. W. C. Dunlop, P. R. Fernandes, y A. P. G. Castro, «Challenges in computational fluid dynamics applications for bone tissue engineering», *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 478, n.º 2257, p. 20210607, ene. 2022, doi: 10.1098/rspa.2021.0607.
- [73] W. Sun, A. Darling, B. Starly, y J. Nam, «Computer-aided tissue engineering: overview, scope and challenges», *Biotechnology and Applied Biochemistry*, vol. 39, n.º 1, pp. 29-47, 2004, doi: 10.1042/BA20030108.
- [74] B. Starly *et al.*, «Computer-aided bone scaffold design: a biomimetic approach», en *2003 IEEE 29th Annual Proceedings of Bioengineering Conference*, mar. 2003, pp. 172-173. doi: 10.1109/NEBC.2003.1216047.
- [75] R. Gabbrielli, I. G. Turner, y C. R. Bowen, «Development of Modelling Methods for Materials to be Used as Bone Substitutes», *Key Engineering Materials*. Accedido: 6 de septiembre de 2019. [En línea]. Disponible en: <https://www.scientific.net/KEM.361-363.903>
- [76] F. P. W. Melchels, A. M. C. Barradas, C. A. van Blitterswijk, J. de Boer, J. Feijen, y D. W. Grijpma, «Effects of the architecture of tissue engineering scaffolds on cell seeding and culturing», *Acta Biomaterialia*, vol. 6, n.º 11, pp. 4208-4217, nov. 2010, doi: 10.1016/j.actbio.2010.06.012.
- [77] «K3DSurf», SourceForge. Accedido: 1 de octubre de 2019. [En línea]. Disponible en: <https://sourceforge.net/projects/k3dsurf/>



- [78] «MathMod», SourceForge. Accedido: 1 de octubre de 2019. [En línea]. Disponible en: <https://sourceforge.net/projects/mathmod/>
- [79] J. E. Roberts y J.-M. Thomas, «Mixed and hybrid methods», en *Handbook of Numerical Analysis*, vol. 2, en Finite Element Methods (Part 1), vol. 2. , Elsevier, 1991, pp. 523-639. doi: 10.1016/S1570-8659(05)80041-9.
- [80] M. Fortin y A. Fortin, «A generalization of Uzawa's algorithm for the solution of the Navier-Stokes equations», *Communications in Applied Numerical Methods*, vol. 1, n.º 5, pp. 205-208, 1985, doi: 10.1002/cnm.1630010504.
- [81] R. Rannacher, «On chorin's projection method for the incompressible navier-stokes equations», en *The Navier-Stokes Equations II — Theory and Numerical Methods*, J. G. Heywood, K. Masuda, R. Rautmann, y V. A. Solonnikov, Eds., en Lecture Notes in Mathematics. Berlin, Heidelberg: Springer, 1992, pp. 167-183. doi: 10.1007/BFb0090341.
- [82] A. J. Chorin, «A Numerical Method for Solving Incompressible Viscous Flow Problems», *Journal of Computational Physics*, vol. 135, n.º 2, pp. 118-125, ago. 1997, doi: 10.1006/jcph.1997.5716.
- [83] G. Winter, J. Abdelfatah, Ramos, Alejandr, y Henríquez, Vicente, «Algunas propuestas para la simulación 3D de la velocidad del aire y la evolución de la temperatura en los edificios», presentado en XVIII Escuela Hispano-Francesa Jacques-Louis Lions sobre Simulación Numérica en Física e Ingeniería, Las Palmas de Gran Canaria, jun. 2018. [En línea]. Disponible en: <http://ehf2018.iusiani.ulpgc.es/speakers.html>  
[http://ehf2018.iusiani.ulpgc.es/speakers\\_files/DeliveredTalk\\_GWinter.pdf](http://ehf2018.iusiani.ulpgc.es/speakers_files/DeliveredTalk_GWinter.pdf)
- [84] G. Zidan, C. Ann Greene, y A. Seyfoddin, «2 - Formulation design in drug delivery», en *Engineering Drug Delivery Systems*, A. Seyfoddin, S. M. Dezfooli, y C. A. Greene, Eds., en Woodhead Publishing Series in Biomaterials. , Woodhead Publishing, 2020, pp. 17-41. doi: 10.1016/B978-0-08-102548-2.00002-0.
- [85] «Theory of self-reproducing automata: John von Neumann (edited by A.W. Burks). University of Illinois Press, Urbana, 1966. xiii + 388pp., \$10.00», *Information Storage and Retrieval*, vol. 5, n.º 3, p. 151, oct. 1969, doi: 10.1016/0020-0271(69)90026-6.
- [86] E. W. Weisstein, «von Neumann Neighborhood». Accedido: 20 de julio de 2022. [En línea]. Disponible en: <https://mathworld.wolfram.com/>
- [87] X. Deng *et al.*, «Topology optimization design of three-dimensional multi-material and multi-body structure based on irregular cellular hybrid cellular automata method», *Scientific Reports*, vol. 12, abr. 2022, doi: 10.1038/s41598-022-09249-y.
- [88] J. Gopinathan *et al.*, «Preparation, characterisation, and in vitro evaluation of electrically conducting poly( $\epsilon$ -caprolactone)-based nanocomposite scaffolds using PC12 cells», *Journal of Biomedical Materials Research Part A*, vol. 104, n.º 4, pp. 853-865, 2016, doi: 10.1002/jbm.a.35620.
- [89] M. Bartnikowski, T. R. Dargaville, S. Ivanovski, y D. W. Hutmacher, «Degradation mechanisms of polycaprolactone in the context of chemistry, geometry and environment», *Progress in Polymer Science*, vol. 96, pp. 1-20, sep. 2019, doi: 10.1016/j.progpolymsci.2019.05.004.
- [90] M. E. Alemán-Domínguez, E. Giusto, Z. Ortega, M. Tamaddon, A. N. Benítez, y C. Liu, «Three-dimensional printed polycaprolactone-microcrystalline cellulose scaffolds», *Journal of Biomedical Materials Research Part B: Applied Biomaterials*, vol. 107, n.º 3, pp. 521-528, 2019, doi: 10.1002/jbm.b.34142.
- [91] A. Fick, «On liquid diffusion», *Journal of Membrane Science*, vol. 100, n.º 1, pp. 33-38, mar. 1995, doi: 10.1016/0376-7388(94)00230-V.
- [92] F. Hecht, «FreeFEM Documentation», p. 673.

- [93] G. Winter, «Aplicación de los métodos de Quasi-Newton a problemas no lineales de transferencia de calor», Universidad de Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, 1984. [En línea]. Disponible en: <https://accedacris.ulpgc.es/handle/10553/2198>
- [94] «Tutorial to write a transient Stokes solver in matrix form». Accedido: 26 de febrero de 2025. [En línea]. Disponible en: <https://doc.freefem.org/tutorials/timeDependentStokes.html>
- [95] H. Antheunis, J.-C. van der Meer, M. de Geus, W. Kingma, y C. E. Koning, «Improved Mathematical Model for the Hydrolytic Degradation of Aliphatic Polyesters», *Macromolecules*, vol. 42, n.º 7, pp. 2462-2471, abr. 2009, doi: 10.1021/ma802222m.
- [96] X. Han y J. Pan, «A model for simultaneous crystallisation and biodegradation of biodegradable polymers», *Biomaterials*, vol. 30, n.º 3, pp. 423-430, ene. 2009, doi: 10.1016/j.biomaterials.2008.10.001.
- [97] J. S. Wiggins, M. K. Hassan, K. A. Mauritz, y R. F. Storey, «Hydrolytic degradation of poly(d,l-lactide) as a function of end group: Carboxylic acid vs. hydroxyl», *Polymer*, vol. 47, n.º 6, pp. 1960-1969, mar. 2006, doi: 10.1016/j.polymer.2006.01.021.
- [98] E. A. Schmitt, D. R. Flanagan, y R. J. Linhardt, «Importance of distinct water environments in the hydrolysis of poly(DL-lactide-co-glycolide)», *Macromolecules*, vol. 27, n.º 3, pp. 743-748, ene. 1994, doi: 10.1021/ma00081a019.
- [99] X. Han y X. Zhang, «15 - Modelling degradation of biodegradable polymers», en *Science and Principles of Biodegradable and Bioresorbable Medical Polymers*, X. Zhang, Ed., Woodhead Publishing, 2017, pp. 427-448. doi: 10.1016/B978-0-08-100372-5.00015-5.
- [100] A. Gleadall, J. Pan, y M.-A. Kruff, «An atomic finite element model for biodegradable polymers. Part 2. A model for change in Young's modulus due to polymer chain scission», *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 51, pp. 237-247, nov. 2015, doi: 10.1016/j.jmbbm.2015.07.010.
- [101] L. Ding, R. L. Davidchack, y J. Pan, «A molecular dynamics study of Young's modulus change of semi-crystalline polymers during degradation by chain scissions», *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 5, n.º 1, pp. 224-230, ene. 2012, doi: 10.1016/j.jmbbm.2011.09.002.
- [102] A. C. Vieira, A. T. Marques, R. M. Guedes, y V. Tita, «Material model proposal for biodegradable materials», *Procedia Engineering*, vol. 10, pp. 1597-1602, ene. 2011, doi: 10.1016/j.proeng.2011.04.267.
- [103] A. F. C. Vieira, E. H. P. Da Silva, y M. L. Ribeiro, «Numerical Approach to Simulate the Mechanical Behavior of Biodegradable Polymers during Erosion», *Polymers*, vol. 15, n.º 9, Art. n.º 9, ene. 2023, doi: 10.3390/polym15091979.
- [104] «Code\_Aster». Accedido: 4 de octubre de 2019. [En línea]. Disponible en: <https://code-aster.org/spip.php?rubrique2>
- [105] A. C. Vieira, R. M. Guedes, y V. Tita, «On Different Approaches to Simulate the Mechanical Behavior of Scaffolds during Degradation», *Procedia Engineering*, vol. 110, pp. 21-28, ene. 2015, doi: 10.1016/j.proeng.2015.07.005.
- [106] Andrew Gleadall y Pan, Jingzhe, «Computer Simulation of Polymer Chain Scission in Biodegradable Polymers», *J Biotechnol Biomater*, vol. 03, n.º 01, 2013, doi: 10.4172/2155-952X.1000154.
- [107] «Curve Fitting Toolbox - MATLAB». Accedido: 2 de julio de 2025. [En línea]. Disponible en: <https://es.mathworks.com/products/curvefitting.html>
- [108] W. Niu y J. Pan, «A model of polymer degradation and erosion for finite element analysis of bioresorbable implants», *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 112, p. 104022, dic. 2020, doi: 10.1016/j.jmbbm.2020.104022.

- [109] S. Osher y J. A. Sethian, «Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations», *Journal of Computational Physics*, vol. 79, n.º 1, Art. n.º 1, nov. 1988, doi: 10.1016/0021-9991(88)90002-2.
- [110] R. Glowinski, T.-W. Pan, y J. Periaux, «A fictitious domain method for Dirichlet problem and applications», *Computer Methods in Applied Mechanics and Engineering*, vol. 111, n.º 3, pp. 283-303, ene. 1994, doi: 10.1016/0045-7825(94)90135-X.
- [111] R. Glowinski, T.-W. Pan, T. I. Hesla, D. D. Joseph, y J. Périaux, «A distributed Lagrange multiplier/fictitious domain method for flows around moving rigid bodies: Application to particulate flow», *International Journal for Numerical Methods in Fluids*, vol. 30, n.º 8, pp. 1043-1066, 1999, doi: 10.1002/(SICI)1097-0363(19990830)30:8<1043::AID-FLD879>3.0.CO;2-Y.
- [112] R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, y J. Périaux, «A Fictitious Domain Approach to the Direct Numerical Simulation of Incompressible Viscous Flow past Moving Rigid Bodies: Application to Particulate Flow», *Journal of Computational Physics*, vol. 169, n.º 2, pp. 363-426, may 2001, doi: 10.1006/jcph.2000.6542.
- [113] S. Osher y R. P. Fedkiw, «Level Set Methods: An Overview and Some Recent Results», *Journal of Computational Physics*, vol. 169, n.º 2, pp. 463-502, may 2001, doi: 10.1006/jcph.2000.6636.
- [114] R. Courant, K. Friedrichs, y H. Lewy, «On the Partial Difference Equations of Mathematical Physics», *IBM Journal of Research and Development*, vol. 11, n.º 2, pp. 215-234, mar. 1967, doi: 10.1147/rd.112.0215.
- [115] A. G. Thombre y K. J. Himmelstein, «Modelling of drug release kinetics from a laminated device having an erodible drug reservoir», *Biomaterials*, vol. 5, n.º 5, pp. 250-254, sep. 1984, doi: 10.1016/0142-9612(84)90068-1.
- [116] K. Sevim y J. Pan, «A model for hydrolytic degradation and erosion of biodegradable polymers», *Acta Biomaterialia*, vol. 66, pp. 192-199, ene. 2018, doi: 10.1016/j.actbio.2017.11.023.
- [117] G. L. Siparsky, K. J. Voorhees, y F. Miao, «Hydrolysis of Polylactic Acid (PLA) and Polycaprolactone (PCL) in Aqueous Acetonitrile Solutions: Autocatalysis», *Journal of Polymers and the Environment*, vol. 6, n.º 1, pp. 31-41, ene. 1998, doi: 10.1023/A:1022826528673.
- [118] J. Abdelfatah Ndioubnane, R. Paz Hernández, M. D. Monzón Verona, A. Gleadall, y G. Winter Althaus, «An overview on degradation of biopolymeric scaffolds and the effect on their mechanical properties», presentado en Congress on Numerical Methods in Engineering (CMN 2022), Las Palmas de Gran Canaria: International Center for Numerical Methods in Engineering (CIMNE), sep. 2022. [En línea]. Disponible en: <https://accedacris.ulpgc.es/handle/10553/119802>

## 15. ANEXOS

### 15.1. METODOLOGÍA PARA EL CÁLCULO DE PROPIEDADES MECÁNICAS POR MEDIO DEL SOFTWARE ABAQUS.

#### 4) Generación de la malla.

Se abre el programa GMSH. El primer paso para dar es crear un fichero nuevo .geo. Para ello, se va a File; New en la barra superior ( Figura 15-1):

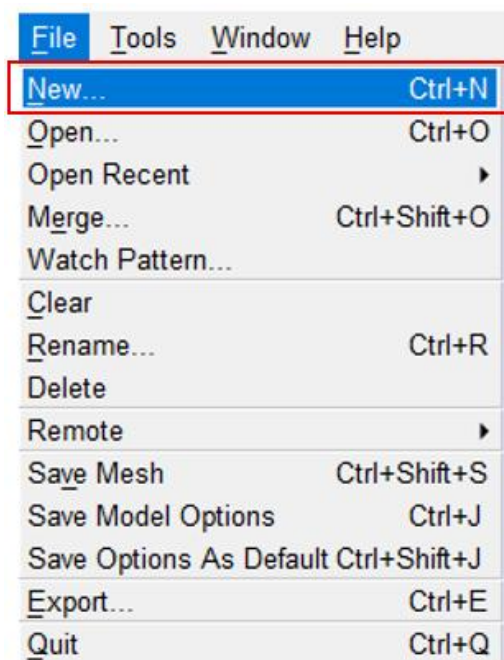


Figura 15-1. Apertura de ficheros en gmsh

En la ventana emergente (Figura 15-2), se especifica el directorio en el que se quiere guardar (importante, debe estar en el mismo que el de la geometría que se vaya a importar), se escribe un nombre seguido de .geo y finalmente le damos a “Guardar”:

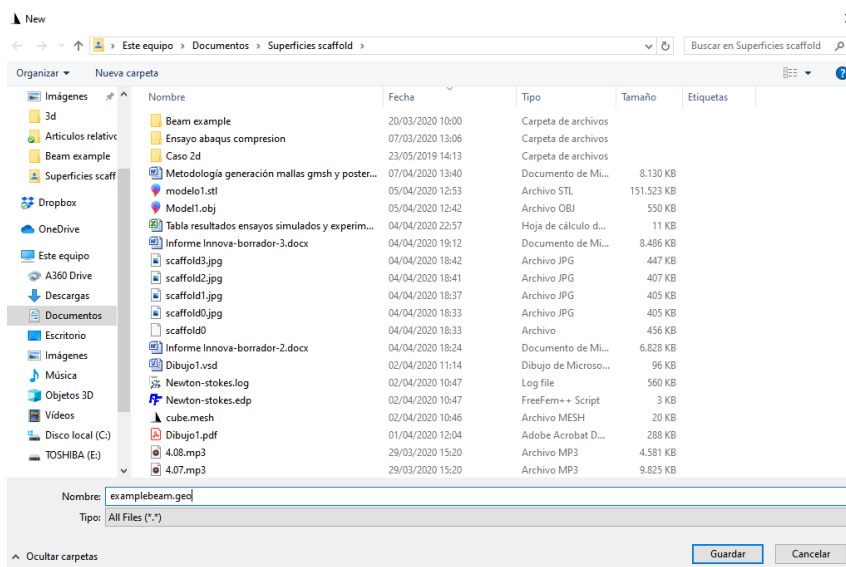


Figura 15-2. Especificación de directorio del fichero .geo

Una vez creado el fichero, aparece una ventana preguntando qué tipo de motor de geometría se quiere usar openCASCADE o built-in (Figura 15-3). El primero está enfocado a las geometrías CAD comunes a los programas de diseño 3D tales como Solidworks, Inventor, Catia, etc., permitiendo la utilización de operadores booleanos como la unión, intersección, y diferencia. El segundo es el motor nativo del programa, presente en las primeras ediciones del programa y que permite crear geometrías básicas, pero no permite hacer modificaciones muy complejas sobre el modelo. Por lo que se optará por la opción OPENCASCADE.

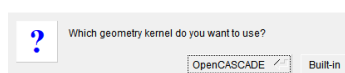


Figura 15-3. Especificación del motor de geometría en gmsh

Para importar una geometría se va al panel izquierdo Geometry; Edit Script y se escribe la instrucción "Merge" seguida del nombre del fichero entre comillas, como se ve en la Figura 15-4. Los formatos CAD que permite abrir el programa son .step o .stp, .iges o .igs y .brep. También permite importar modelos de malla de superficie como .stl o .vtk:



Figura 15-4. Importación de la malla en el bloc de notas.



Para visualizar la geometría en el visor, vamos a Geometry; Reload Script. Aparecerá la geometría representada en forma alámbrica ( Figura 15-5)

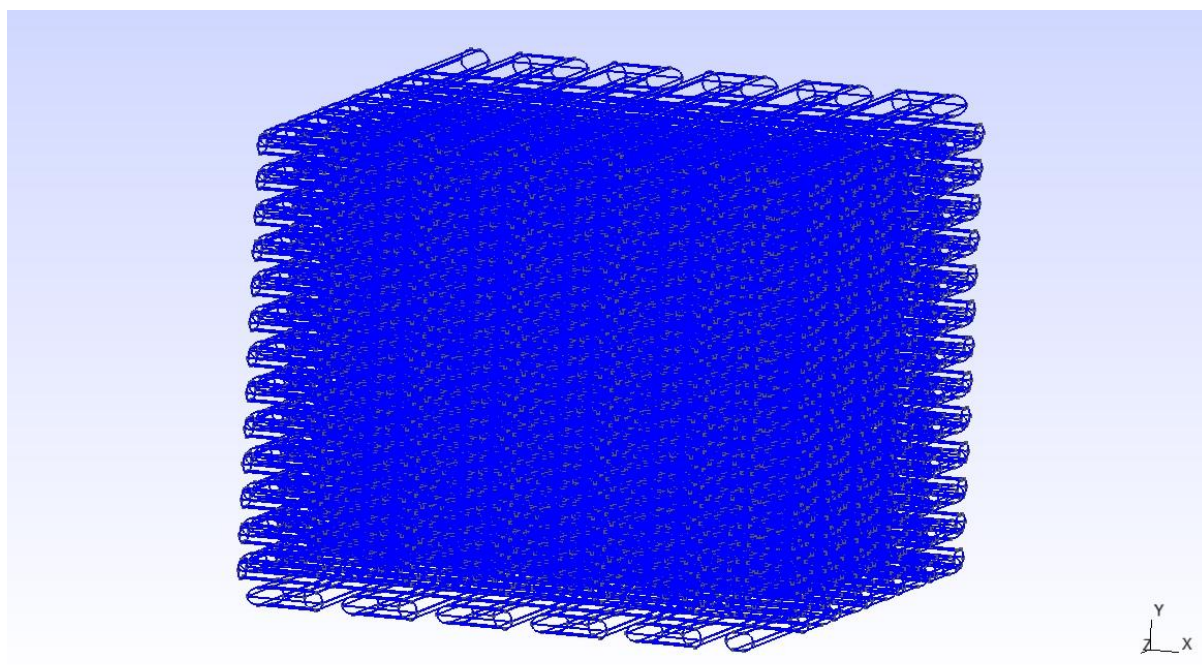


Figura 15-5. Visualización de la geometría importada dentro del programa.

Para comprobar las dimensiones de la geometría importada se va a Tools;Options (Figura 15-6). En la ventana emergente, dentro de la pestaña General, se selecciona la casilla Show Bounding Boxes.

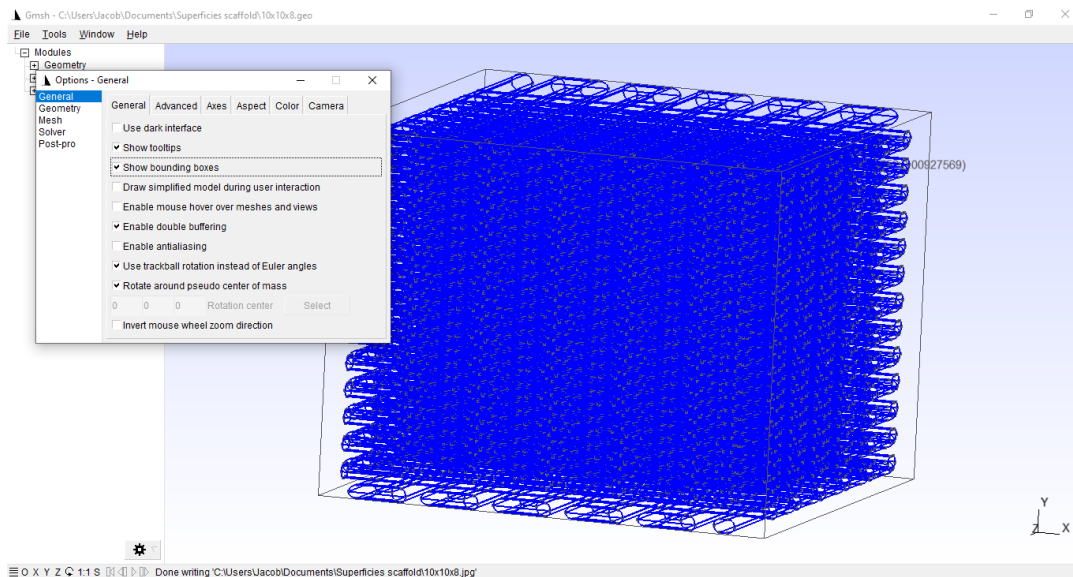


Figura 15-6. Activación de la opción "Show bounding boxes" para la comprobación de las dimensiones de la pieza

Como se puede ver en el panel izquierdo, hay tres módulos de trabajo dentro del programa (Figura 15-7):

El módulo geometry, para la creación y modificación de geometrías; el módulo Mesh para definición y generación de mallas, y el módulo Solver para postprocesado y visualización de resultados. Lo importante en esta etapa es la generación de mallas por lo que se enfocará en el módulo Mesh.

En primer lugar, se asignará un tamaño de elemento de malla a la pieza. Para ello, se despliega el módulo mesh pulsando sobre el botón "+" situado a su izquierda, y luego a Define; Size at Points.

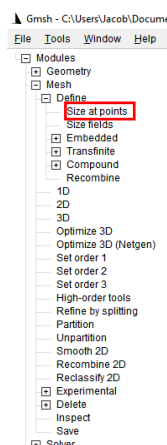


Figura 15-7. Definición del tamaño de malla-1.

En el visor se seleccionan los vértices de la pieza con la tecla Ctrl y arrastrando con el ratón hasta que cubra la totalidad de la misma, manteniendo pulsado el botón izquierdo del ratón (Figura 15-8). Aparecerán los puntos en color rojo como se muestra en la figura.

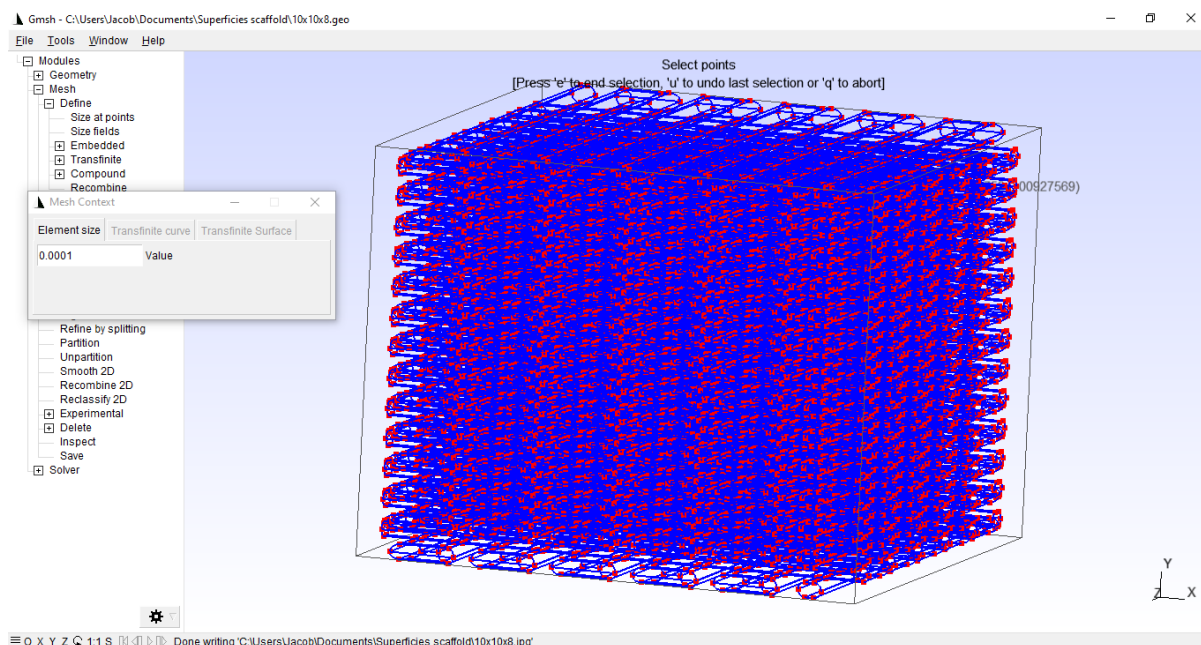


Figura 15-8. Definición del tamaño de malla-2

En la ventana “Mesh Context” se asigna a la pieza un tamaño de elemento lo suficientemente bajo para la obtención de resultados precisos pero que no sea demasiado bajo a la vez permita al equipo resolver el modelo. Por ello, de acuerdo a las dimensiones de la pieza se ha optado por un tamaño de elemento de 0.001.

Una vez hecho esto, clicar fuera de la ventana, y pulsar la tecla “e”. Los cambios se verán reflejados en la información contenida en cada nodo.

El paso final será generar el discretizado de la geometría. Dentro del módulo mesh, se selecciona la opción 3D y el programa generará la malla. También se puede generar la malla pulsando sobre la tecla 3. Asimismo en el script, escribiendo al final la instrucción “Mesh 3;”, es posible también realizar el mallado. Una vez aplicado se visualizará la pieza en el visor (Figura 15-9):



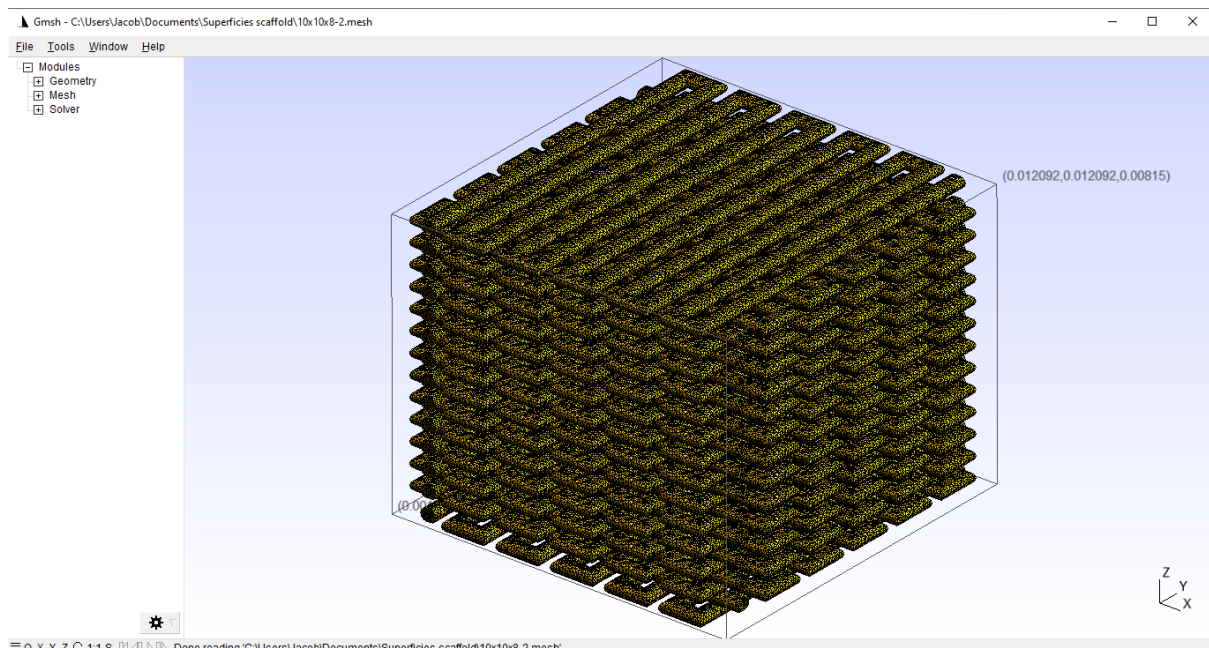


Figura 15-9. Visualización de la malla generada.

Una vez generada la malla, si se desea que los tetraedros tengan vértices intermedios (elementos C3D10 en Abaqus), se selecciona la opción “Set order 2”, en el menú contextual “Mesh” del panel izquierdo, lo que hará que los elementos sean cuadráticos.

En el siguiente paso se explicará la fase de exportación.

## 5) Importación del modelo

Una vez obtenida la geometría erosionada en un instante de tiempo determinado en FreeFEM++ (véase epígrafe 8.3), se exporta a un formato legible por el software de simulación. En este caso, FreeFEM++ no permite exportar a un formato que se pueda utilizar como modelo en ABAQUS. Por tanto, se utilizará un paso intermedio, que es el importar la pieza en una etapa específica de degradación, en GMSH, para luego, exportarlo en ABAQUS. El FreeFEM permite exportar las mallas en el formato nativo MEDIT, con la extensión .mesh, que a su vez es legible por GMSH. La instrucción para ello es:

```
Savemesh(scaffold, “scaffold.mesh”);
```

Una vez exportada la malla, se abre en GMSH. El siguiente paso será exportar la malla en formato compatible con Abaqus, para lo cual el GMSH permite dos opciones: Nastran bulk data file(.bdf) y Abaqus input file(.inp). Ambos formatos son compatibles, pero finalmente se ha optado por la realización del proceso en formato .inp ya que es nativa al programa y no genera mensajes de error en la lectura.

Para ello se va a File; Export y se le da un nombre al fichero seguido de la extensión .inp.

Ya en ABAQUS, se va a File;Import;Model y se escoge .inp como formato. Hecho esto aparecerá el modelo representado en el visor (Figura 15-10):

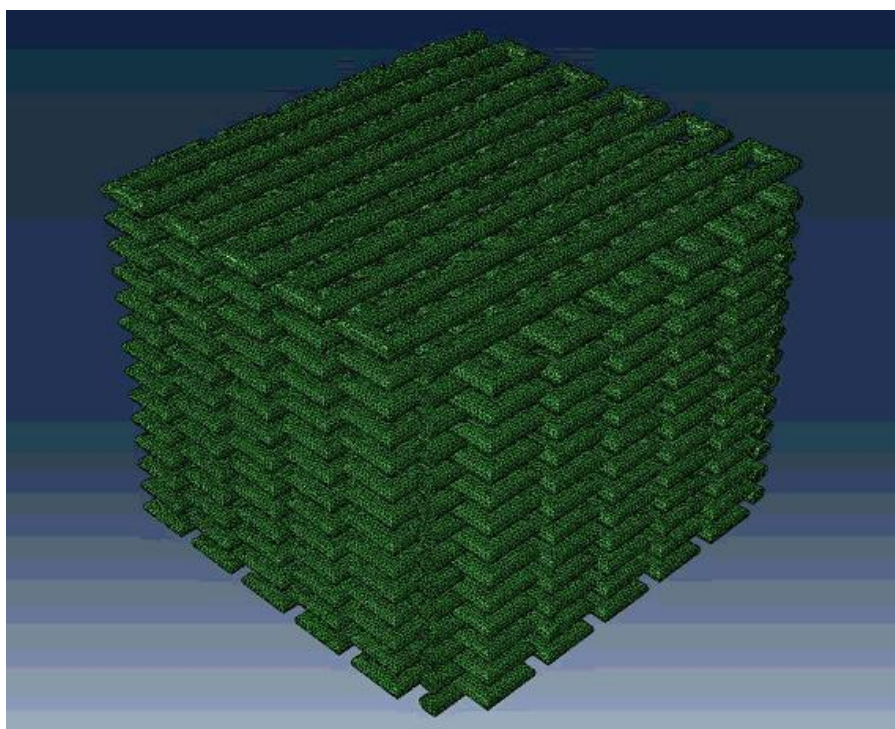


Figura 15-10. Malla importada en ABAQUS

Antes de poder trabajar con la geometría, hay que eliminar los elementos de superficie, que se importan por defecto desde el Gmsh y que generalmente acarrearán problemas en cuanto a la asignación de propiedades.

Abaqus denomina a aquellas mallas que no han sido generadas dentro del programa sino que han sido importadas de otros softwares, mallas huérfanas (orphan mesh en inglés). Para editar dichas mallas, el programa solo permite hacerlo dentro del módulo mesh yendo a la opción “mesh;edit”.

En primer lugar, en el árbol tree, se despliega la pieza y haciendo clic sobre el icono mesh, con la combinación Ctrl+espacio se cambiará al módulo mesh (Figura 15-11):

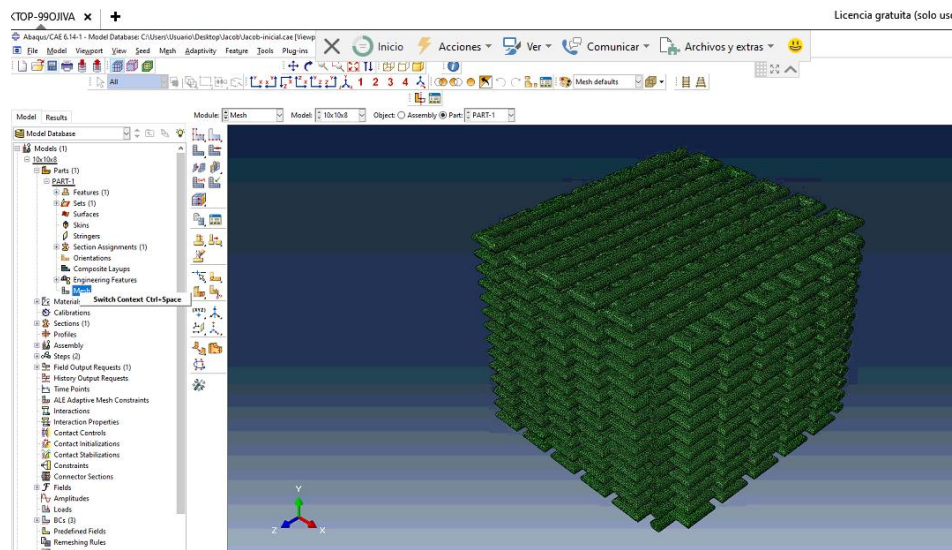


Figura 15-11. Cambiar de módulo mesh en el menú de árbol de ABAQUS.

Después, en la barra superior ir a Mesh>Edit (Figura 15-12):

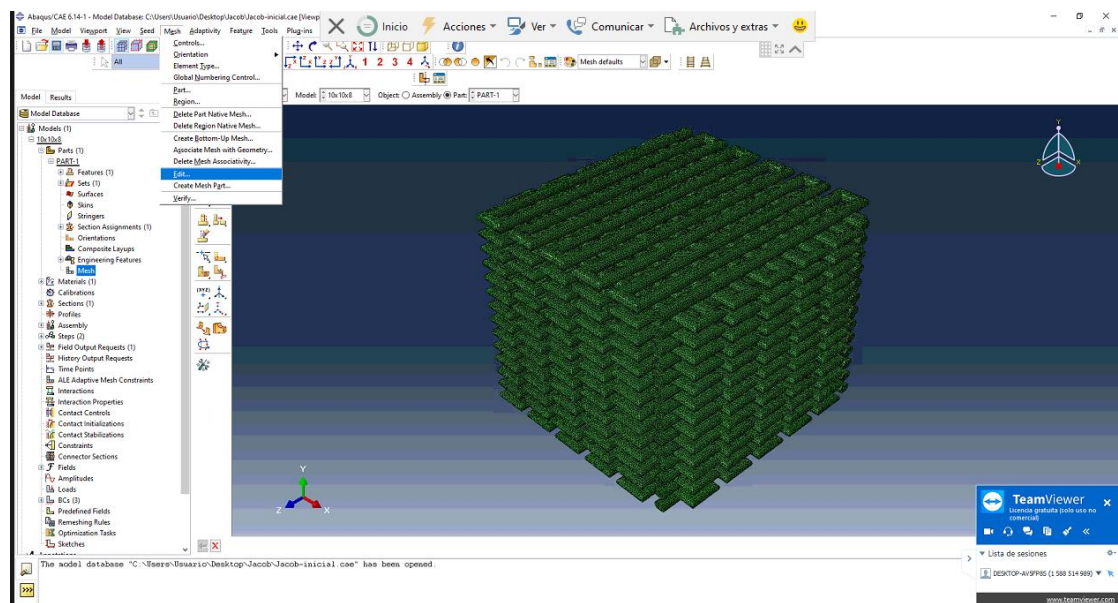


Figura 15-12. Selección del módulo de edición de mallas.

En la ventana emergente, en “Category” se va a “Element” y en “Method” se selecciona “Delete” (Figura 15-13):

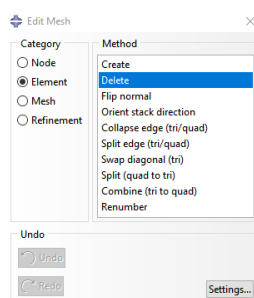


Figura 15-13. Selección del método de eliminación de elementos de la malla.

En el panel superior, se cambia el tipo de elementos a seleccionar de All a 2D elements (Figura 15-14):

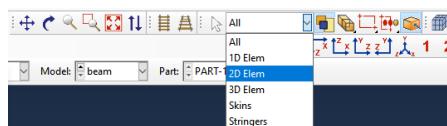


Figura 15-14. Selección de elementos 2D en la barra de selección de entidades.

Hecho esto, en el visor se seleccionan todos los elementos por lo que se quedará la figura remarcada en rojo (Figura 15-15):

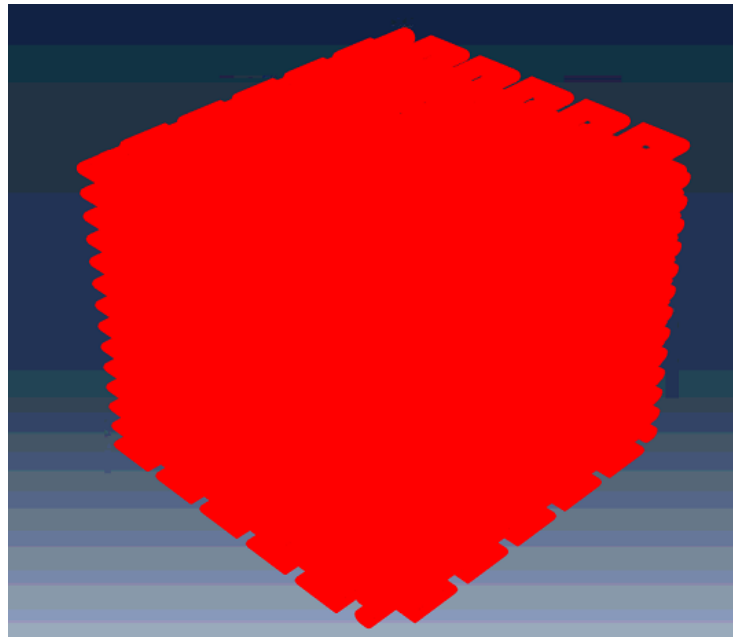


Figura 15-15. Visualización de los elementos de superficie seleccionados en color rojo

Finalmente se pulsa el botón Done situado en la barra inferior y ya quedarán todos los elementos de superficie removidos (Figura 15-16):

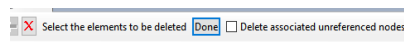


Figura 15-16. Tecla "Done" para la eliminación de los elementos seleccionados.

## 6) Edición de propiedades

Para la edición de propiedades se va primero a Module;Properties (Figura 15-17):

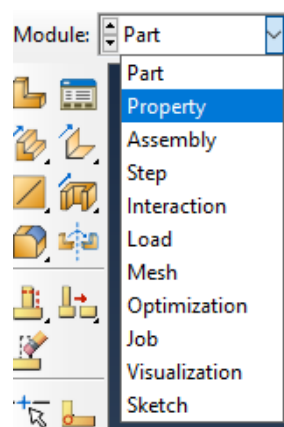


Figura 15-17. Selección del módulo de propiedades en ABAQUS

Dentro del módulo los pasos a seguir para la correcta asignación de propiedades son los siguientes (Figura 15-18):

- Propiedades de material
- Sección
- Asignación de sección.

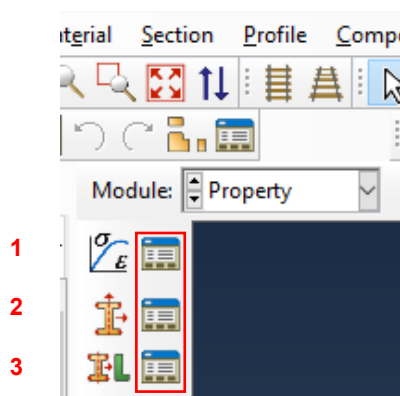


Figura 15-18. Módulo de propiedades en ABAQUS.

Para ello, se va al icono “Material Manager” y se selecciona el botón “Create” (Figura-15-19):

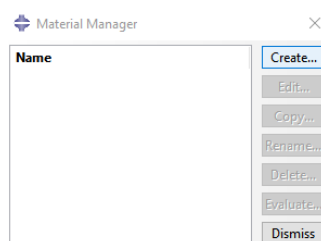


Figura-15-19. Ventana principal del gestor de propiedades

En la ventana emergente, se selecciona el tipo de Material. En este caso, se irá a Mechanical; Elasticity; Elastic (Figura 15-20):

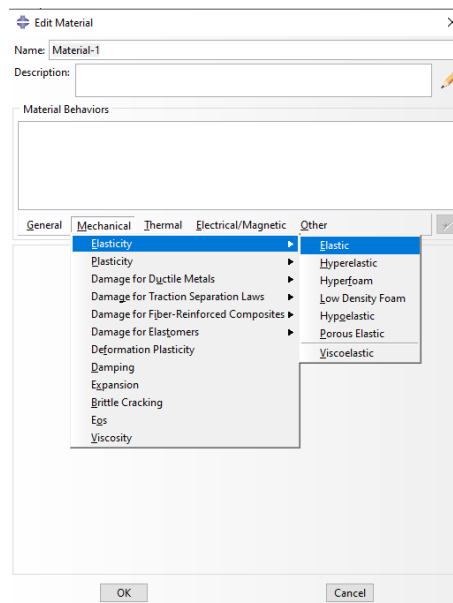


Figura 15-20. Selección del tipo de material para el estudio.

En este material se asignan las propiedades del módulo de elasticidad y de Poisson (Figura -15-21). En este caso, un Módulo de Young de  $3,5 \cdot 10^8$  Pa y un coeficiente de Poisson de 0,46.

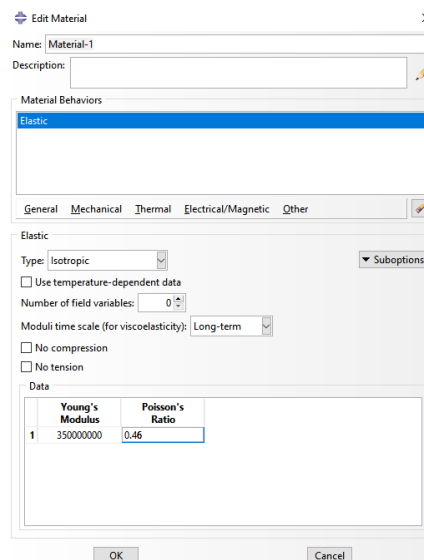


Figura -15-21. Definición de las propiedades del material

El siguiente paso es asignar una sección. Para ello hay que ir al segundo icono en el panel de propiedades. En category se escoge la opción “Solid” y en type “Homogeneous” y finalmente se pulsa sobre “continue” (Figura 15-22):

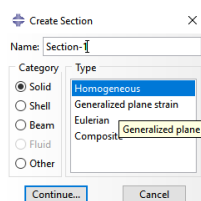


Figura 15-22. Panel principal del gestor de secciones

El último paso sería asignar la sección al material definido previamente, que ya se ha asignado dentro del fichero en cuestión. Para ello se va al tercer icono, y en la ventana emergente, se selecciona el botón “Create”. Se selecciona la malla en el visor, y se pulsa sobre la tecla “Done” (Figura-15-23):

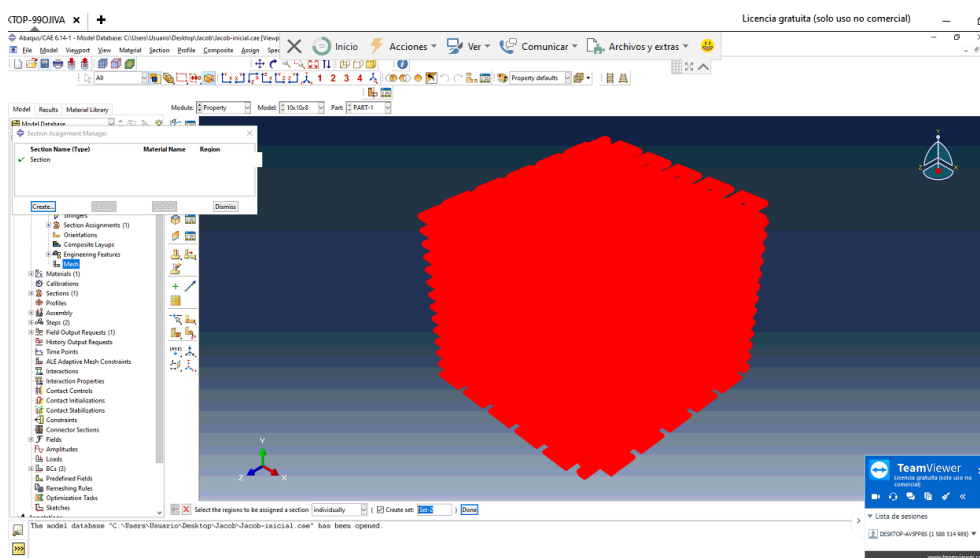


Figura-15-23. Asignación de material a las secciones

Una vez hecho esto, el modelo cambiará su color a turquesa, por lo que la asignación de material habrá sido realizada de forma correcta (Figura 15-24)



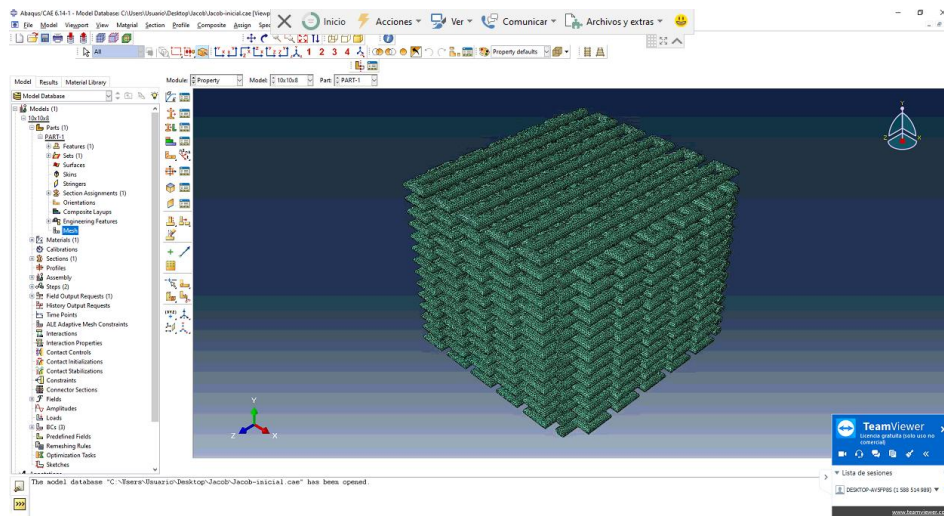


Figura 15-24. Representación de la pieza en color turquesa tras asignación de propiedades

## 7) Asignación de steps o pasos en la simulación para la posterior creación de condiciones de contorno

Aquí se definirán el tipo de simulación que se quiere realizar y los resultados que se quieren almacenar. Dentro del módulo step se tienen tres iconos como se muestra en la figura, y que se irán editando uno a uno, según el orden mostrado en la Figura 15-25:

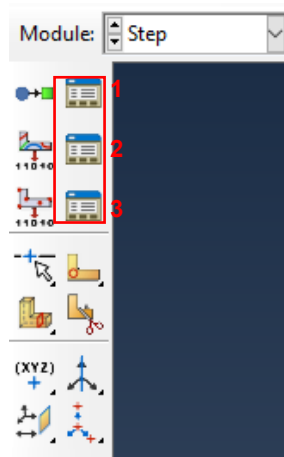


Figura 15-25. Panel principal en el modulo step

Por defecto, el ABAQUS tiene predefinido un step denominado “initial” que sirve para la definición de aquellos puntos en los que el desplazamiento es nulo. También sirve para definir aquellos puntos donde se van a definir condiciones de contorno constantes durante el estudio.

Además, se procede a la definición de un step adicional en el que se definirá el tipo de cálculo, se almacenarán los resultados del análisis y se definen aquellas condiciones de contorno que requieran de un cálculo del instante inicial y final. Para ello hay que ir al primer icono de la figura, y en la ventana emergente se selecciona el tipo de análisis que se va realizar. En este caso se elegirá el tipo Static, general (Figura 15-26):

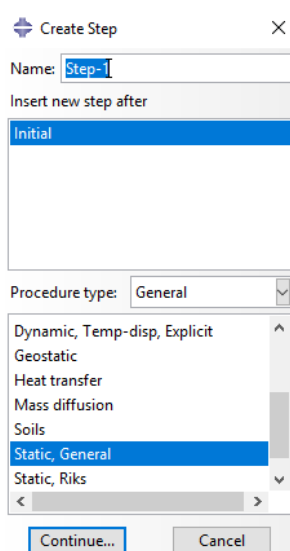


Figura 15-26. Creación de análisis

En la siguiente ventana se activa la opción NLGEOM a ON para que el programa considere tensiones y deformaciones reales, en vez de los valores nominales. También es adecuado tomar esta opción para el caso de deformaciones que sobrepasen el 5% (Figura 15-27):

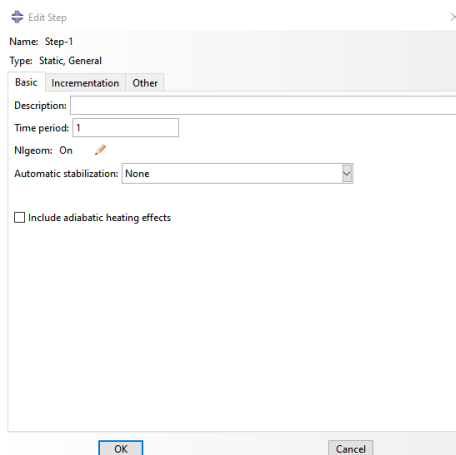


Figura 15-27. Ventana de edición del step.

Por último, se registran las variables que se desean tener en los resultados. Los resultados registrados pueden pertenecer al dominio entero, o a una parte del mismo. En el segundo icono del panel, “Field Output”, se seleccionan las variables que se desean registrar para su posterior visualización y análisis. Para este caso, se escogerán las variables de salida definidas por defecto, clicando en “Preselected defaults” (Figura 15-28):

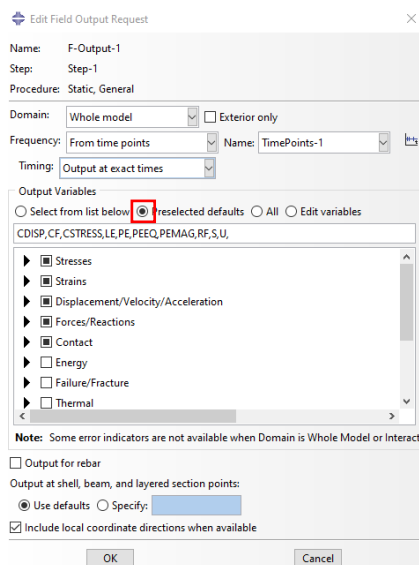


Figura 15-28. Ventana de edición de la salida de datos en "Field Output"

En History Output se seleccionan los puntos sobre los que se quiere obtener información específica. En este caso, se quiere conocer el valor de la fuerza de reacción en la base, por lo

que se seleccionan los puntos de la cara inferior. Para ello, se selecciona un set previamente creado que contenga dichos puntos ( Figura 15-29):

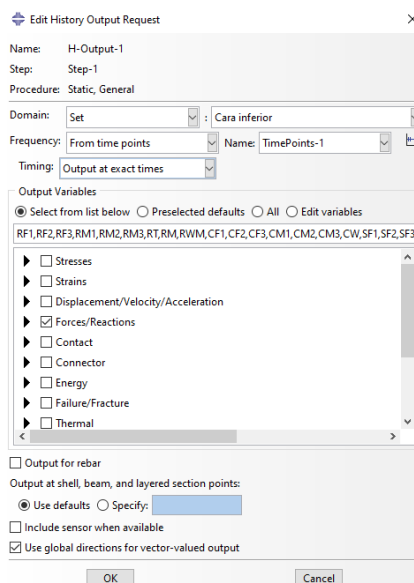


Figura 15-29. Ventana de edición de la salida de datos en "History Output"

Por otra parte, para la obtención de resultados en diferentes momentos de la simulación según se vaya aplicando el desplazamiento, Abaqus permite la introducción de "Time points" (puntos temporales) que van variando desde el inicio de la simulación ( $t=0$ ) hasta el final de la misma ( $t=1$ ). En la ventana (Figura 15-30), se activa primero la casilla "Specify using delimiters", y se especifica el inicio, el fin y el incremento.

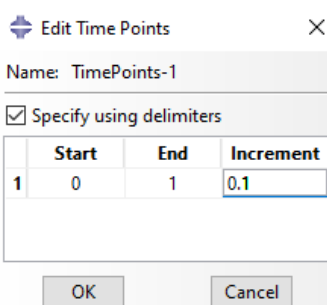


Figura 15-30. Ventana de inserción de "Time points".

Si solo se quiere representar aquellos valores en la gráfica del último paso, en la ventana de "Field Output" se selecciona "Last increment" como se ve en la Figura 15-31:

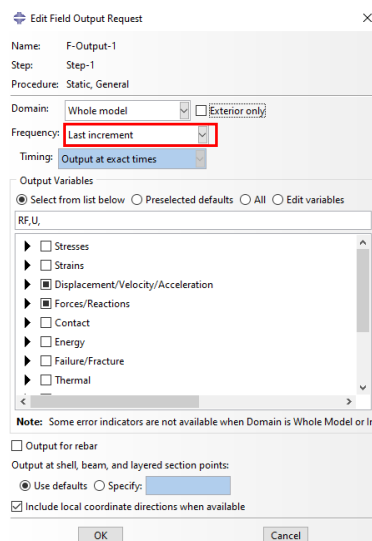


Figura 15-31. Selección del modo "Last Increment" en Field Output Manager.

### 15.1.1. DEFINICIÓN DE CARGAS Y CONDICIONES DE CONTORNO

Para la definición de condiciones de contorno y las cargas, hay que trabajar en el módulo "Load". En este trabajo se considera un desplazamiento vertical en la cara superior, mientras que en la cara inferior se considera una restricción de desplazamiento nulo en el eje longitudinal de la pieza, y una condición de encastre o desplazamiento y rotación nulo en dos puntos. Resumiendo, tenemos las siguientes condiciones de contorno a aplicar son:

- Desplazamiento longitudinal negativo de -0.4 mm en la cara superior de la pieza.
- Desplazamiento nulo en dirección longitudinal de la pieza en su base.
- Restricción de desplazamiento y rotación nulo en todas las direcciones de uno o dos puntos centrales de la base.

Para la aplicación de las diferentes condiciones de contorno, ABAQUS las distingue según su naturaleza. Para aquellas que se mantienen constantes o son nulas durante el análisis, para su definición, se usa el step "initial". En cambio, para aquellas que se van aplicando de forma progresiva, se recurre al uso del step-1 definido en el paso 4 del epígrafe 14.1.

En primer lugar, se selecciona el icono "BCs", señalado en el menú de árbol, situado en el panel izquierdo de la interfaz principal de usuario (Figura 15-24). Aparecerá una ventana emergente,

en la que se definirán las condiciones de frontera a aplicar. Para las condiciones de contorno de desplazamiento nulo en la vertical y de desplazamiento nulo en todas las direcciones de los puntos de encastre, en el campo “step”, se selecciona el campo “Initial”. En “Category”, se selecciona “Mechanical” y en “Types for selected step”, Displacement/Rotation para el primero (Figura 15-32), y Symmetry/Antisymmetry/Encastre para el segundo.

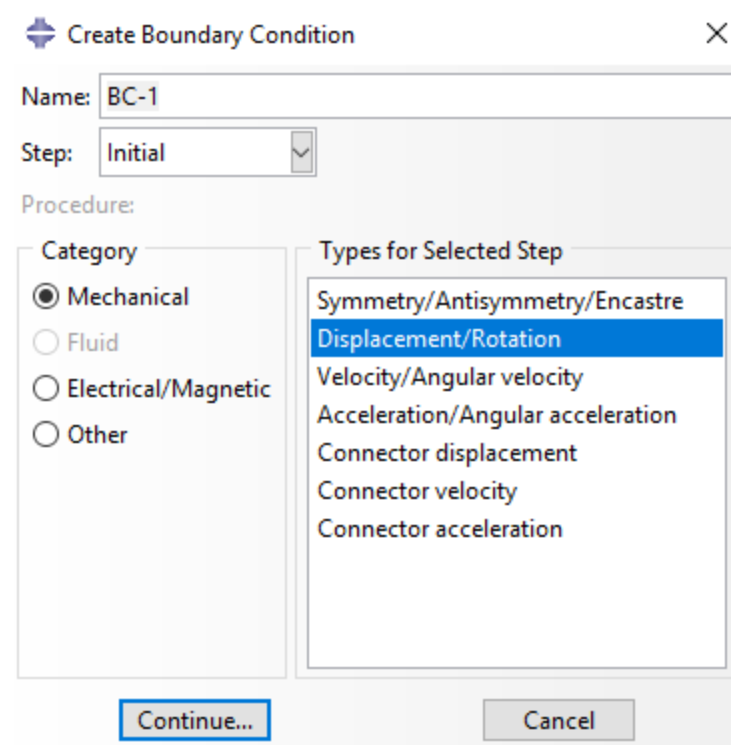


Figura 15-32. Ventana de creación de condiciones de contorno

Se le da a continue, y en la ventana emergente que aparece se editan los desplazamientos que se quieren restringir, en este caso el desplazamiento longitudinal U1 para la base (Figura 15-33) y condición de encastre para los puntos creados en dicha superficie (Figura 15-34):

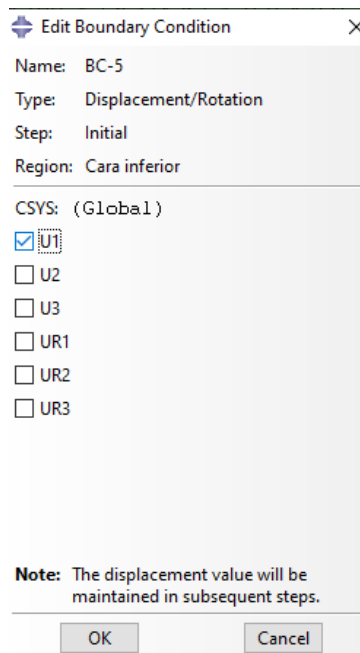


Figura 15-33. Definición de restricciones en la base de la pieza

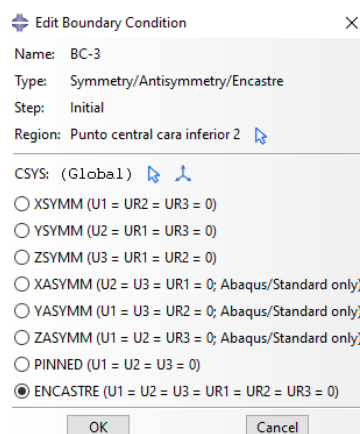


Figura 15-34. Definición de restricciones en un punto central de la base.

Para la imposición del desplazamiento en la cara superior, se crea una nueva condición de contorno (Figura 15-35), se selecciona el step creado en el y en el panel de tipos, “Displacement/Rotation”. En la ventana emergente, se activa la casilla correspondiente al desplazamiento U2, que es el que se ha considerado como eje vertical, y se define el valor asignado.

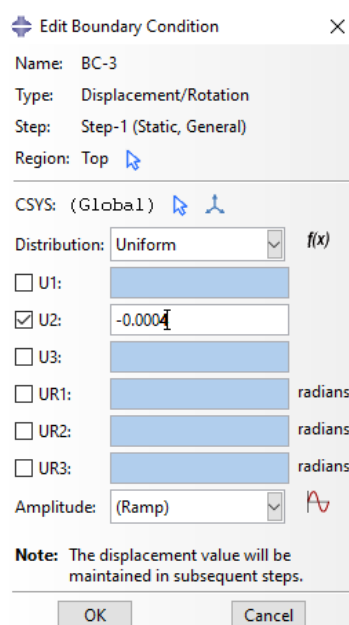


Figura 15-35. Definición de desplazamiento en la cara superior del scaffold.

### 15.1.2. EJECUCIÓN DE LA SIMULACIÓN

Para la ejecución de la simulación es preciso definir una serie de parámetros relativos al tipo de trabajo a ejecutar y a los recursos destinados a utilizar durante el transcurso del proceso. En el módulo job se selecciona el icono “Job manager”. En la ventana emergente (Figura 15-36), se editan parámetros tales como la memoria asignada para el cálculo, el número de procesadores para el trabajo en paralelo. Una vez editados los cambios, se pulsa el botón OK, y se cierra la ventana.



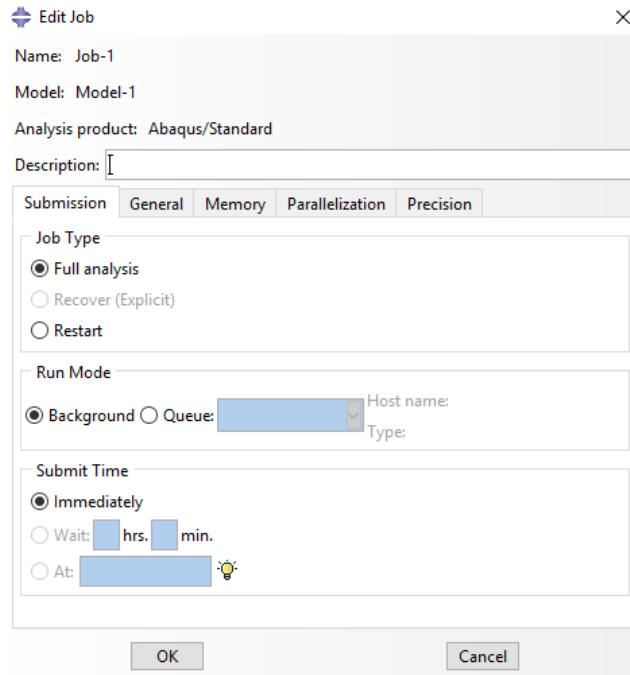


Figura 15-36. Editor de trabajos

El siguiente paso es mandar el trabajo. Para realizar esto, es preciso es crear el input file .inp, yendo a Write Input como se ve en la esquina superior derecha de la Figura 15-37. Una vez creado el archivo, se selecciona la casilla “data check”, que sirve para comprobar si nuestro modelo tiene algún fallo. Por último, si la comprobación no tiene ningún problema, se pulsa la casilla “Submit” para ejecutar el trabajo.

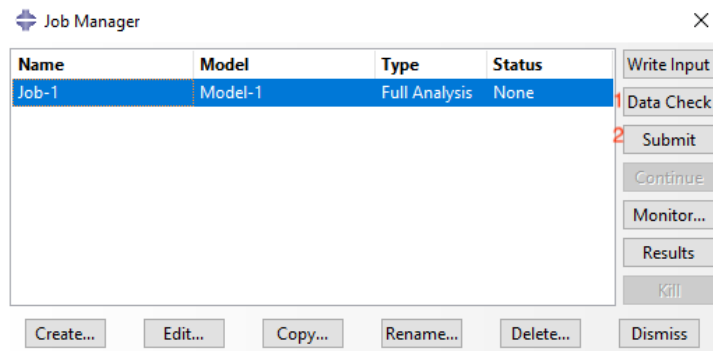


Figura 15-37. Gestor de trabajos

Una vez finalizados los cálculos se procede a la visualización de los resultados obtenidos y al análisis de estos. El mapa correspondiente a las fuerzas de reacción muestra los siguientes resultados (Figura 15-38):

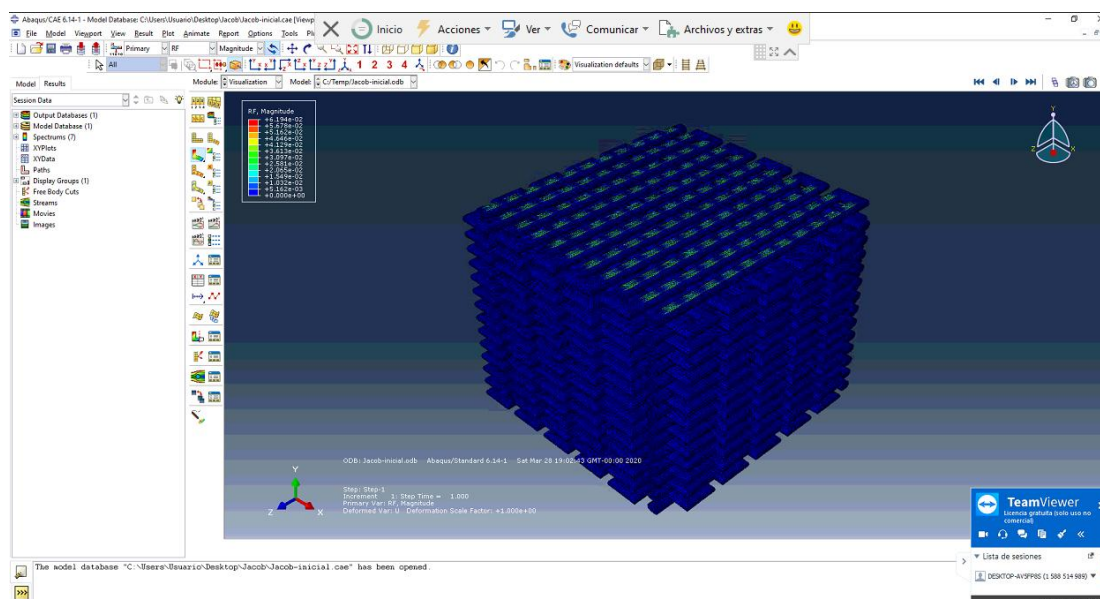


Figura 15-38. Mapa de fuerzas de reacción sobre la pieza.

Para obtener la resultante sobre la base, en primer lugar, hay que registrar los datos obtenidos en una tabla. Para ello, se va al icono “Create XY data”, señalado en rojo en la figura anterior.

En la ventana emergente seleccionamos “ODB field output”, y se pulsa sobre continue (Figura 15-39).

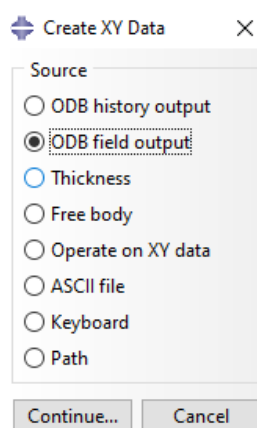


Figura 15-39. Ventana de creación de la tabla de resultados. Selección de la casilla "ODB field output".

A continuación, en la pestaña “Variables”, se selecciona en el campo “Position” la opción “Unique Nodal” y se activa la casilla “Reaction Force” (Figura 15-40):

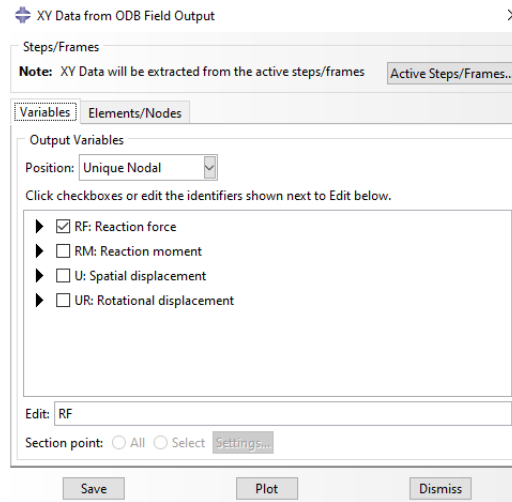


Figura 15-40. selección de las fuerzas de reacción como variable a almacenar en la tabla

En la pestaña “Elements/Nodes” se selecciona como método de selección “Node sets” y el set correspondiente a la base, como se ve en la Figura 15-41. Se guardan los datos y se cierra la ventana.

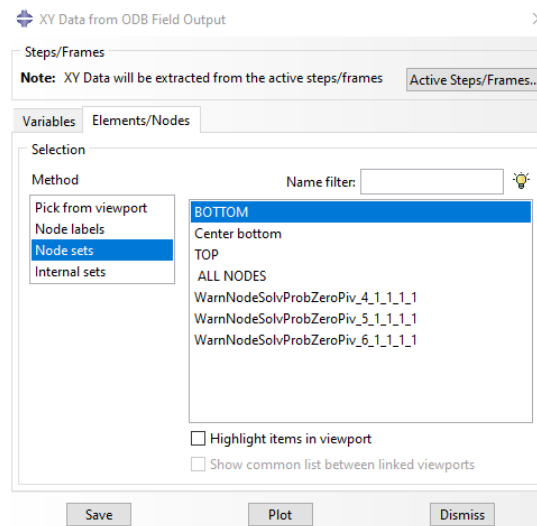


Figura 15-41. Selección de los nodos donde queremos registrar la fuerza de reacción.

Obtenidos los valores de la fuerza de reacción en los distintos nodos de la base, se procede al cálculo de su resultante (Figura 15-42). Para ello, se va nuevamente a “Create XY data” y se selecciona la opción “Operate on XY data”. En la ventana que aparece hay que seleccionar el operador  $\text{sum}((A,A,...))$  en el panel de la derecha. Después, en el recuadro “XY data” se eligen todos los datos relativos a la fuerza de reacción vertical. Una vez seleccionados, clicar sobre el icono “Add to expresión”, para incluirlo en la expresión del recuadro superior:

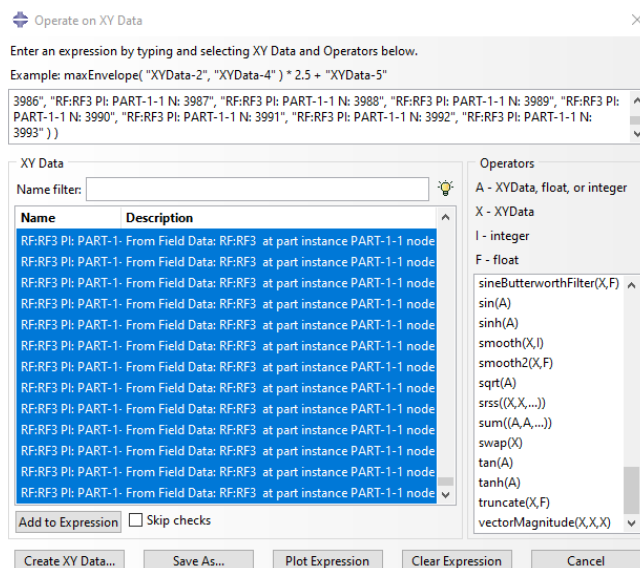


Figura 15-42. Utilización del operador  $\text{sum}()$  para sumar los valores de la fuerza de reacción en la base.

Para ver la gráfica se pulsa sobre “Plot Expression”. La gráfica muestra una especie de rampa. Esto es debido a que la carga se aplica de forma gradual ajustándose a una recta (Figura 15-43).

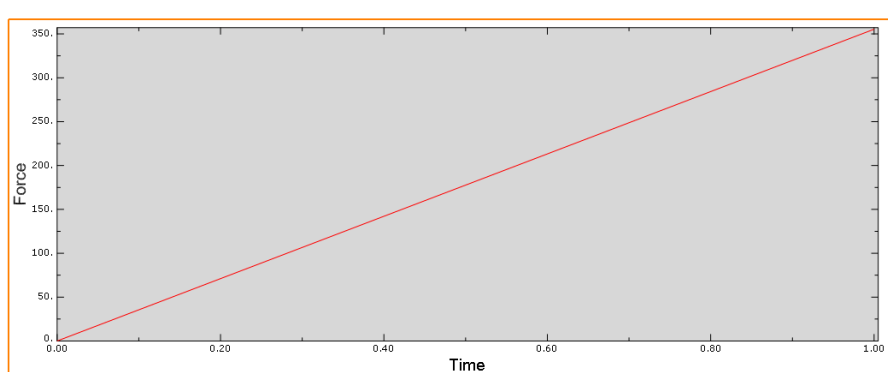


Figura 15-43. Representación de la fuerza resultante respecto al tiempo.

Para consultar este valor, se va a Tools;Query, y en Visualization Module Queries, se selecciona “Probe values”. Después en la gráfica, se coloca el cursor sobre el punto máximo de la curva y dará las coordenadas del punto en la gráfica. **La coordenada vertical del punto señalado en verde será el valor resultante de la fuerza de reacción** (Figura 15-44).

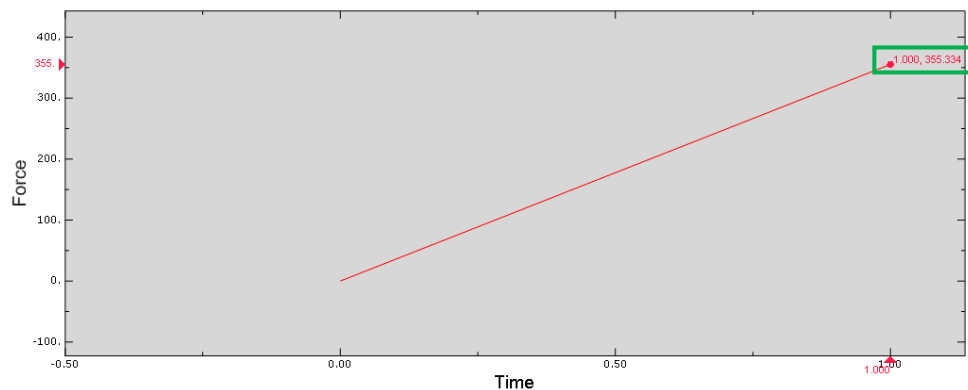


Figura 15-44. Localización del punto de fuerza resultante en la gráfica.

El cálculo del módulo de elasticidad se hace dividiendo el valor obtenido de la fuerza de reacción entre la superficie de aplicación de la carga, según la expresión ( 15-1):

$$E_{global} = \frac{F_{reacción} / A_{equivalente}}{desplazamiento / Longitud} \quad (15-1)$$

El área equivalente  $A_{equivalente}$  representa la superficie de aplicación del desplazamiento y se determina a partir de la distancia máxima del prisma envolvente al dominio en el plano XZ. La longitud corresponde con la distancia máxima medida en la dirección de aplicación del desplazamiento.

1. Para medir , se va a Tools; query; Distance, y se seleccionan 2 puntos situados en los extremos de la superficie según la dirección de uno de los ejes, como se ve en la siguiente figura. Esto dará la distancia vertical de la pieza (Figura 15-45).

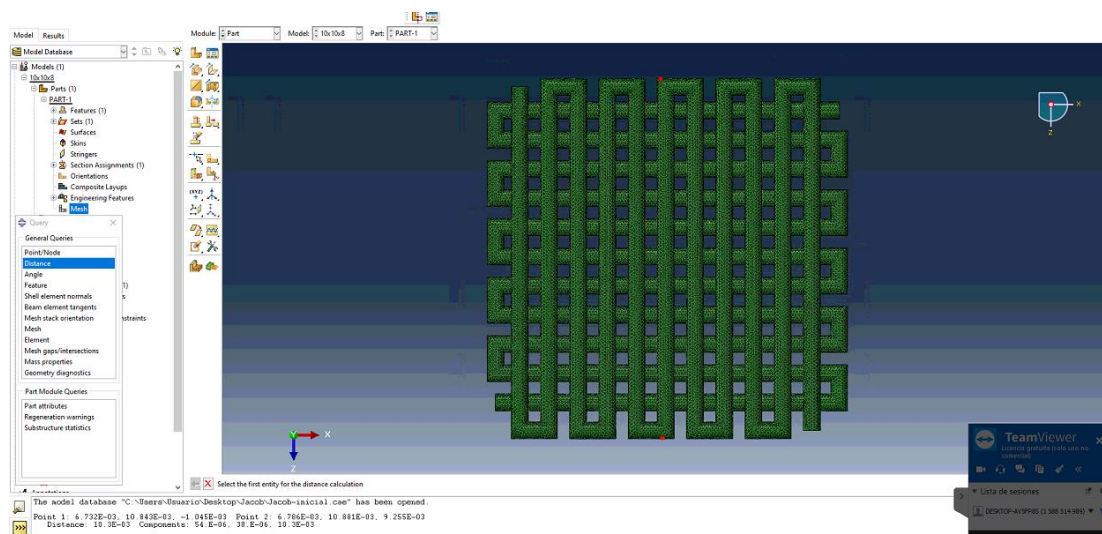


Figura 15-45. Medición de las dimensiones de la pieza

2. Se repite el mismo procedimiento para la dirección ortogonal.
3. Se mide la altura correspondiente al dispositivo en dirección perpendicular al plano de la superficie de la misma manera que antes.
4. Una vez obtenidos, se sustituyen en la expresión anterior para la obtención del correspondiente módulo de elasticidad.

Este proceso se repite para cada una de las etapas de degradación de la pieza.

## 15.2. SCRIPT CORRESPONDIENTE AL CÁLCULO DE LA EROSIÓN DEBIDO A LAS TENSIONES TANGENCIALES EN UNA GEOMETRÍA SENCILLA.

```

/* APLICACION :
// CALCULA VELOCIDAD CON MODELO DE NAVIER-STOKES ALGORITMO DE CHORIN
MODIFICADO, realizando una sola adaptacion de malla

// MODELA UN VERTIDO PUNTUAL INSTANTANEO

// MODELA CALCULANDO EN CADA PASO DE TIEMPO la dispersion del vertido
sometido al campo de velocidades resultante del calculado anteriormente

*/

load "gmsh"

```

```
load "msh3"
load "iovtk"
load "mshmet"
load "tetgen"

/*ETIQUETAS
1 ENTRADA
2 PAREDES LATERALES
3 SALIDA
4 SCAFFOLD
5 VOLUMEN DE CONTROL
*/

real TIME;

mesh3 malla=readmesh3("Volumen de control.mesh");
mesh3 scaffold=readmesh3("scaffoldsencillo.mesh");

savemesh(malla,"malla.mesh");

plot(malla);

fespace Vh(malla,P1);

/* en el algoritmo que estamos usando casi siempre, el de Chorin modificado
para resolver calcular velocidades de un fluido incompresible , se aproximan
por polinomios de grado uno velocidades y tambien presiones cuestion que
no va bien con otros algoritmos de resolucio de Navier Stokes que las
velocidades se aproximan por P2 y presiones por P1 */

real dt = 0.001, tf=20.;
//tf tiempo final de la simulacion de calculo de velocidades

real alpha=1/dt;

fespace Vh2(malla,P2);

Vh2 Coldt,C,vh;
// C es concentracion que se dispersa y se difunde
// vh es la funcion test de la formulacion variacional del MEF

real aa=-1.25,bb=0.000001;

/* A CONTINUACION MODELO DE LA FUENTE PUNTUAL (VERTIDO PUNTUAL DE UN
CONTAMINANTE O ALGUNA VARIABLE QUE SE DISPERSA */
```

```

// comprobamos que el valor promedio integral de fp (valor medio de fp
// donde esta definido)
//en su mallado malladofp da el valor A

Vh w,u = 0, v =0,ww=0, p = 0, q=0; // esta linea debe estar aqui porque se
nombra a continuacion la variable u

Vh tao,taomax;

// definicion modelo dispersion de contaminante

// ***** AHORA resolucioN Navier Stokes *****

fespace Wh(scaffold,P2);

real nu = 0.000016; // nu es la viscosidad cinematica del fluido
Vh uold = u, vold = v, wwold=ww, pold=p;
real volume= int3d(malla)(1.);
real t;
macro ut(u1,u2,u3)([N.(N.u1),N.(N.u2),N.(N.ww)])//EOM
macro dn(u) (N.y*dy(u) )//EOM

for(int n=0;n<700;n++)

{
Vh      f=convect([u,v,ww],-dt,uold),      g=convect([u,v,ww],-dt,vold),
h=convect([u,v,ww],-dt,wwold);
Vh uold = u, vold = v, wwold=ww, pold=p;

t=t+dt;

solve pb4u(u,w,init=n,solver=CG)
=int3d(malla)(u*w/dt +nu*(dx(u)*dx(w)+dy(u)*dy(w)+dz(u)*dz(w)))
-int3d(malla)((f/dt-dx(p))*w)
+ on(1,u = 2) + on(2,4,u = 0)
+ on(3,u=f);

//on(5,u=-nv*y); // CONTORNO LABEL 5 ESTA EN ROTACION

//en la entrada del fluido o contornos de label=1 , se ha definido un perfil
parabolico de velocidades

```



[illegible]

```

malla=tetgreconstruction(malla,switch="raAQ",sizeofvolume=h2*h2*h2/6);
*/

tao1=tao1;

savemesh(scaffold,"scaffold-"+n+".mesh");
savevtk("scaffold"+n+".vtk",scaffold,region,dataname="scaffold");

mesh3 mallar=trunc(scaffold,abs(tao1)>3.5,split=1,label=4);
//mesh3 mallar2=trunc(malla,abs(tao-5)=1,split=1,label=4);
scaffold=trunc(scaffold,abs(tao1)<=3.5,split=1,label=4);

malla=malla+mallar;
malla=change(malla,rmInternalFaces=1,fregion=5);
u=u;
v=v;
ww=ww;
w=w;
p=p;
q=q;
tao=tao;
plot(malla);
plot(mallar);
savemesh(malla,"malla-"+n+".mesh");
savevtk("malla"+n+".vtk",malla,region,dataname="regiones");
savemesh(mallar,"mallar.mesh");
//savemesh(scaffold,"scaffold-"+n+".mesh");
//savevtk("scaffold"+n+".vtk",scaffold,region,dataname="scaffold");

if ( !(n % 11))
plot(tao,cmm=" esfuerzo cortante =" +t, nbiso=30,value=1);

//if ( !(n % 31))
//plot(coef=0.5,cmm=" velocidad =" +t,[u,v],value=1);

TIME+=dt;
cout << " dt = " << dt << " TIME = " <<TIME<<endl;
savevtk("velocidad"+n+".vtk",malla,[u,v,ww],tao,dataname="velocidad
tension-tangencial");
}

// ***** fin resolucion Navier Sokes *****

//plot([u,v], coef=0.5, value=1, ps="velocidad.ps");

```

```
plot(tao,cmm=" esfuerzo cortante =", nbiso=30,value=1);
```

```
savevtk("velocidadfinal.vtk",malla,[u,v,ww],dataname="velocidad");
```

### 15.3. SCRIPT CORRESPONDIENTE A LA GENERACIÓN DE LA MALLA ESTRUCTURADA DEL FILAMENTO POLICAPROLACTONA(PCL)

```
SetFactory("OpenCASCADE");
```

```
Merge "Filamento_PCL.brep";
```

```
//+
Transfinite Curve {5, 7, 20, 22, 27, 11, 13, 15, 17, 28, 8, 10, 18, 19, 26,
2, 4, 23, 24, 25} = 14 Using Progression 1;
//+
Transfinite Curve {14, 21, 12, 9, 6, 3, 1, 16} = 14 Using Progression 1;
//+
Transfinite Surface {8};
//+
Transfinite Surface {7};
//+
Transfinite Surface {4};
//+
Transfinite Surface {3};
//+
Transfinite Surface {2};
//+
Transfinite Surface {1};
//+
Transfinite Surface {6};
//+
Transfinite Surface {5};
//+
Transfinite Surface {11};
//+
Transfinite Surface {9};
//+
Transfinite Surface {10};
//+
Transfinite Surface {12} Right;
//+
```

```

//Recombine Surface {8, 7, 5, 4, 11, 12, 9, 10, 6, 3, 1, 2};
//+
Extrude {0, 0, 50.3} {
    Point{16}; Point{12}; Point{10}; Point{11}; Point{15}; Point{9};
    Point{14}; Point{13}; Point{17}; Point{7}; Point{8}; Point{1}; Point{3};
    Point{5}; Point{2}; Point{6}; Point{4}; Curve{24}; Curve{22}; Curve{14};
    Curve{21}; Curve{23}; Curve{12}; Curve{20}; Curve{17}; Curve{15};
    Curve{16}; Curve{28}; Curve{25}; Curve{27}; Curve{11}; Curve{13}; Curve{9};
    Curve{18}; Curve{26}; Curve{2}; Curve{8}; Curve{5}; Curve{19}; Curve{1};
    Curve{6}; Curve{10}; Curve{3}; Curve{4}; Curve{7}; Surface{8}; Surface{7};
    Surface{5}; Surface{4}; Surface{11}; Surface{12}; Surface{9}; Surface{10};
    Surface{6}; Surface{3}; Surface{1}; Surface{2}; Layers{350};
}
//+

//+
Physical Surface("Boundary", 1) = {47, 29, 52, 41, 34, 21, 38, 32, 50, 17,
44, 25, 28, 46, 20, 39, 33, 48, 36, 30, 37, 16, 31, 42, 14, 23, 45, 27, 51,
3, 26, 4, 40, 19, 2, 18, 10, 12, 7, 49, 13, 9, 35, 11, 22, 1, 15, 8, 43,
6, 5, 24};
//+
Physical Volume("Region", 2) = {10, 4, 12, 8, 2, 6, 7, 11, 5, 1, 9, 3};
//+
//+
Transfinite Volume{1} = {16, 12, 11, 15, 21, 20, 19, 18};
//+
Transfinite Volume{2} = {10, 16, 15, 9, 23, 21, 18, 22};
//+
Transfinite Volume{3} = {12, 14, 13, 11, 20, 25, 24, 19};
//+
Transfinite Volume{4} = {10, 9, 7, 8, 23, 22, 26, 27};

```

### 15.3. SCRIPT CORRESPONDIENTE AL MODELO CORRESPONDIENTE AL MODELO DE MONTE CARLO PARA LA EROSIÓN ESTÁTICA DE UN FILAMENTO.

```

load "gmsh"
load "msh3"
load "iovtk"
load "mshmet"
load "tetgen"
load "ffrandom"

//verbosity=11;

```

```

{
ofstream file("resultados-PLA.txt");
file << "          iteration          "<< " "<< "          number of removed
elements          "<< "          "<< "          volume of boundary
elements          "<< "          "<< "          reference removed
volume          "<< "          "<< "          Metodologia
aplicada          "<< "          "<< "          dissolved
volume          "<< " "<< " percentage of removed elements "<<endl;
file
<< "_____ "<< " "<< "_____
_____ "<< " "<< "_____ "<< " "<< "_____
_____ "<< " "<< "_____ "<< " "<< "_____
_____ "<<endl;
}

//mesh3 malla=readmesh3("Fluid Domain.mesh");
mesh3 scaffold=readmesh3("Filamento_PCL_2.mesh");
mesh3 scaffoldr;
mesh3 scaffoldrr;

real rho=1.24e^-6;
real volsc=scaffold.measure;
cout<<"el volumen del scaffold es "<<volsc<<endl;

//REFERENCE DATA OF LOSS MASS VALUES
real[int] massloss=[1.62,6.38,13.8];
real[int] volloss=massloss;

//savemesh(scaffold,"malla.mesh");

int numeroftetrahedra=scaffold.nt;

cout<<"el numero de elementos es"<<numeroftetrahedra<<endl;

//real[int] probabilidad(numeroftetrahedra);

int NbBoundaryElements = scaffold.nbe;
cout << "Number of boundary element = " << NbBoundaryElements << endl;

fespace Ph(scaffold,P0);
//Ph ok=0;
Ph okc=0;

```

```

//Ph Be=0;
Ph vol=volume;

int m=0;
//int n=0;
real d=0;
real boundvol=0;
real X;
real t=0;
real dt=0.5;

volsc=scaffold.measure;
srandomdev();
for(int n=0; n<500; n++){
cout<<"ITERATION "<<n<<endl;
//CREATE A VARIABLE TO COMPUTE THE NUMBER OF FACES OF EACH ELEMENT IN
CONTACT OF THE SURROUNDING MEDIA
real[int] c(scaffold.nt);
for (int i=0;i<scaffold.nt;i++){
c[i]=0;}
real csum=0;
real cmed=0;
real nbt=0;
vol=volume;
real cc=0;
Ph Be;
Be=0;
    Ph ok=0;
Ph probabilidad=0;

//IDENTIFY THE VOLUME OF EACH TETHRAEDRA AND THEIR RESPECTIVE VOLUMES AND
THE SUM OF THEM.
for (int k=0; k<scaffold.nbe; ++k)

{

    int Tetrahedra = scaffold.be(k).Element;
    int d =Tetrahedra;
    if ((Be[][d]==0)){
Be[][d]=1;
boundvol=boundvol+vol[][d];

        for (int e = 0, ee; e < 4; e++){
            //set ee to e, and ee is change by method adj,
            //cout << d << " " << e << " <=> " << int(scaffold[d].adj((ee=e)))
<< " " << ee
            //<< ", adj: " << (scaffold[d].adj((ee=e)) > 0) << endl;

```

```

        if (scaffold[d].adj((ee=e)) < 0){
//INCREASE THE VALUE OF C IN CASE OF THE FACE IS IN CONTACT WITH THE
LIQUID
            c[d]=c[d]+1;}
    }
    //cout<<"el numero de caras en contacto es "<<c[d]<<endl;
    csum=csum+c[d];
    nbt=nbt+1;
    //cout<<"el valor medio de caras en contacto es "<<cmed<<endl;
    }

    }
    savevtk("scaffoldbe-"+n+".vtu", scaffold, Be, dataname="shear-stress");

    cmed=csum/nbt;
    //Be=0;
    cout<<"the volume of the boundary elements is "<<boundvol<<endl;
    cout<<"the experimental removed volume is "<<0.01*volloss[m]*volsc<<endl;

    real remainingmass=volloss[m]*volsc*0.01;

    if(boundvol>0.01*volloss[m]*volsc-scaffoldr.measure){

    for (int k=0; k<scaffold.nbe; ++k)
    {
        //int kk=k;
        int Tetrahedra = scaffold.be(k).Element;
        int d=Tetrahedra;
        //cout<<"el tetraedro "<<Tetrahedra<<"tiene un identificador
        "<<Be[][d]<<endl;

        //EVALUATE THE FACES OF THE ELEMENT.

        /*
            for (int e = 0, ee; e < 4; e++){
                //set ee to e, and ee is change by method adj,
                //cout << d << " " << e << " <=> " << int(scaffold[d].adj((ee=e)))
                << " " << ee
                //<< ", adj: " << (scaffold[d].adj((ee=e)) > 0) << endl;
                if (scaffold[d].adj((ee=e)) < 0){
//INCREASE THE VALUE OF C IN CASE OF THE FACE IS IN CONTACT WITH THE
LIQUID
                    c=c+1;}
            }
        */
        //cout<<"el valor de c es "<<c<<endl;
    
```

```

real diff=abs(0.01*volloss[m]*volsc-scaffoldr.measure);
real diffmax=(0.01*volloss[2]*volsc-scaffoldr.measure);
real diff0=abs(0.01*volloss[m]*volsc);
real diff0max=abs(0.01*volloss[2]*volsc);
//CALCULATE THE AMPLIFICATION FACTOR

/*
if (t=0){

else{
real X=diff/((boundvol*0.5)*diffmax);
}
*/
//COMPUTE THE PROBABILITY
//probabilidad[][d]=(X*c/4)*t*exp(-X*c/4*t);
//////////

cout<<"el valor medio de caras en contacto es "<<cmed<<endl;

real X=abs(0.01*volloss[m]*volsc-scaffoldr.measure)/((cmed/4)*boundvol);
//probabilidad[][d]=1-exp(-0.00812*t);
probabilidad[][d]=X*c[d]/4;
cout<<"The probability of dissolution is "<<probabilidad[][d]<<endl;

//////////////////////GENERATE A RANDOM NUMBER BETWEEN 0 AND ONE
TO COMPARE WITH THE PROBABILITY OF EACH ELEMENT THAT FOLLOWS AN EXPONENTIAL
DISTRIBUTION.//////////////////////

//GENERATE A RANDOM NUMBER BASED ON AN UNIFORMAL DISTRIBUTION
real zz=randreal1();
//LANDA IS A VALUE USED TO ADJUST THE REMOVAL RATE OF THE ELEMENTS
real landa=0.00812;
real z;
z=(-log((1-zz))/landa);

if(z>1){
while(z>1){
zz=randreal1();
z=(-log((1-zz))/landa);
//cout<<"el valor del numero aleatorio es "<<z<<endl;
}
}

```



```
// INCLUIR AQUI QUE GENERE UN RANDOM QUE SIGA LA DISTRIBUCION EXPONENCIAL  
(VER MATLAB)
```

```
cout<<"THE VALUE OF THE RANDOM NUMBER IS "<<z<<endl;
```

```
//IDENTIFICATION OF THE ELEMENTS WHICH PROBABILITY IS LOWER THAN THE RANDOM  
VALUE Z TO PROCEED OF ITS REMOVAL BY USING THE OK FUNCTION
```

```
if ((z<probabilidad[][d])){  
    ok[][d]=1;  
    cc=cc+1;
```

```
    //d=d+1;
```

```
    //cout<<"el tetraedro "<<d<<" se disolvera"<<endl;  
    //cout<<"se eliminaran "<<cc<<" elementos"<<endl;  
}
```

```
else{  
    //cout<<"el tetraedro "<<Tetrahedra<<" no se disolvera"<<endl;  
    ok[][d]=0;  
}
```

```
//c=0;  
}
```

```
}
```

```
//Be[][d]=1;//NO SE QUE ES ESTO
```

```
//cout<<"el valor de c es "<<c<<endl;
```

```
//Be[][d]=1;  
//scaffoldr=trunc(scaffold,ok==1,split=1,label=4);
```

```

//
//savemesh(fluid,"fluid"+n+".mesh");
//plot(scaffold);
//plot(scaffoldr);
savevtk("scaffold-"+n+".vtu",scaffold,ok,Be,dataname="shear-stress");
//REMOVING THE TRIANGLES WHICH COMPLIES THE LATTER CONDITION
//if(cc>0){
if((boundvol<volloss[m]*volsc*0.01-scaffoldr.measure)){
    cout<<"se procede a la remocion de los elementos de contorno"<<endl;
    //volloss[m]=(volloss[m]*volsc*0.01-boundvol)/(volsc*0.01);
    scaffoldrr=trunc(scaffold,Be==1,split=1,label=1);
    scaffold=trunc(scaffold,Be==0,split=1,label=1);
    //volloss[m]=(volloss[m]*volsc*0.01-boundvol)/(volsc*0.01);
    scaffoldr=scaffoldr+scaffoldrr;
    savemesh(scaffold,"scaffold"+n+".mesh");

scaffoldr=change(scaffoldr,rmInternalFaces=1);
savevtk("scaffold-residual-"+n+".vtu",scaffoldr,ok,Be,dataname="shear-
stress");
savevtk("scaffold-eroded-"+n+".vtu",scaffold,ok,Be,dataname="shear-
stress");
if(mpirank==0){
ofstream file("resultados-PLA.txt", append);
{
    file<<"          "<<n<<"          "<<" "<<"          "
<<scaffoldr.nt<<"          "<<" "<<"          "<<boundvol<<"
          "<<" "<<"          "<<volloss[m]<<"          "<<" "<
<"          "Elements
contorno          "<<" "<<"          "<<scaffoldr.measure<<"
          "<<" "<<"          "<<(scaffoldr.measure/volsc)*100<<"
          "<<endl;}}

}
boundvol=0;
}

else{
    cout<<"se procede a la remocion de los elementos que cumplen con el
criterio"<<endl;
    scaffoldrr=trunc(scaffold,ok==1,split=1,label=1);
    scaffold=trunc(scaffold,ok==0,split=1,label=1);

    scaffoldr=scaffoldr+scaffoldrr;
    savemesh(scaffold,"scaffold"+n+".mesh");

    scaffoldr=change(scaffoldr,rmInternalFaces=1);

```

```

savevtk("scaffold-residual-"+n+".vtu", scaffoldr, ok, Be, dataname="shear-
stress");
savevtk("scaffold-eroded-"+n+".vtu", scaffold, ok, Be, dataname="shear-
stress");

if(mpirank==0){
ofstream file("resultados-PLA.txt", append);
{
file<<"          "<<n<<"          "<<" "<<"          "<<sc
affoldr.nt<<"          "<<" "<<"          "<<boundvol<<"
          "<<" "<<"          "<<volloss[m]<<"          "<<" "<<"
                                     Monte
Carlo          "<<" "<<"          "<<scaffoldr.measure<<"
          "<<" "<<"          "<<(scaffoldr.measure/volsc)*100<<"
          "<<endl; }

}
boundvol=0;
}
//}

//      cout<<"se disolvieron "<<d<<"elementos"<<endl;

//if(scaffoldr.measure>(volloss[m]*0.01*volsc)&&(0.01*volloss[m]*volsc-
scaffoldr.measure<0.005)){
if(((0.01*volloss[m]*volsc-scaffoldr.measure)/volsc)*100<0.01){
m=m+1;

cout<<"el valor de m es "<<m<<endl;
}

if(m>2){
break;
}

boundvol=0;
t=t+dt;
}

```

## 15.4. SCRIPT CORRESPONDIENTE A LA PRIMERA PROPUESTA DEL MODELO INTEGRADO PARA LA DEGRADACIÓN Y EROSIÓN DEL POLÍMERO

```

load "msh3" load "iovtk" load "gmsh" load "tetgen" load "mshmet" load
"medit" load "mmg" load "ffrandon" load "distance"

//include "MonteCarlo.idp"

mesh3 Th=readmesh3("Model 1_2.mesh");
mesh3 Th;

plot(Th);
//real k1=0.003;
real k2=0.002;
real n=0.5;
real D=1.41e-14;
real l=0.5;
real Free=1;
real Symmetry=2;
real a=0.6131;
real c=0.5077;
real d=-5.408;
real e=-7.796;
real delta;
real Km=11;
real boundvol=0;
real csum=0;
real cmed=0;
real nbt=0;
real cc=0;
real Mncrit=0;
real eps= 1e-8;
real D0n=1;
real Ks = 1e-10;

////////////////////////////////////REACTION
DIFFUSION////////////////////////////////////
////////////////////////////////////

//NORMALIZED VALUES

/*

```

```

real k1n=k1/(k2*Ce0^n);
real D0n=D0/((Ce0^n)*k2*1^2);
*/

real t=0;
real dt=0.01;
real T=8;

fespace Vh(Th,P13d);
fespace Xh(Th,P0);
fespace Mh(Th,[P1,P1,P1,P1,P1,P1]);
Mh [m11,m21,m22,m31,m32,m33];
real lerr=0.05;

Vh k1n, Cm, Ce, v, w, Cmold, Cmoldt, Cm0, Ce0, Ceold, Ceoldt, Dpol, error,
g, m, b, phi, phiold, Nx, Ny, h, H1, nabla ;
Xh vol=volume;
m[]=1/vol[];
real mttotal=1;
real mttotalprev;
real volnorm;
real Cmavg,Ceavg,Cmavgprev,Dpolavg;

Xh Ce1, Cm1;

Ce0=1;
Ce=Ce0;

Cmold=0;
Ceold=1;
Cm=0;
real Cmmax=0;
real alpha=4.5;
Vh f;

////////MACRO DEFINITIONS: GRADIENT AND DIVERGENCE
macro grad(u) [dx(u), dy(u), dz(u)]//
macro div(u) (dx(u)+dy(v)+dz(w)) //
//Vh sol=exp(-(k1n+Cm^n)*t);

////////////////////////////////////////COMPUTING INITIAL VOLUME OF THE UNIT
CELL////////////////////////////////////////
real volinit=Th.measure;
cout<<"El volumen de la pieza es "<<volinit<<endl;
real vol0=Th.measure/volinit;

```

```

////////////////////////////////////
////////////////////////////////////

```

```

////////////////////////////////////INITIALIZATION OF THE
AVERAGE NORMALISED MONOMER CONCENTRATION////////////////////////////////////
Cmavg=int3d(Th)(Cm*v)/volinit;

```

```

////////////////////////////////////DISTANCE
COMPUTATION////////////////////////////////////

```

```

varf vb(uaux, vaux) = on(1, uaux = 1);////////////////////////////////Setting uaux to 1
in the interfacce
Vh ub = vb(0, Vh, tgv=1);
plot(ub, value=1);

```

```

ub[] = ub[] ? 0:1;
plot(ub,value=1);

```

```

distance(Th,ub,phi[],distmax=100);//compute absolute distance(phi) respect
to the interface
phi=-phi*(region==OMEGA)+ phi*(region==omega);////assign signed distance
respect to the interface (phi < 0 in OMEGA and phi>0 in omega)

```

```

plot(phi);

```

```

Th = change(Th, rmInternalFaces = 1); //FEMOVE PHYSICAL INTERFACES

```

```

H = 1*(phi >= -epsilon) + 0*(phi < epsilon);////////////////CHARACTERISTIC
FUNCTION //////////////////////////////////
////////////////////////////////////ASSIGN RHO AND MU VALUES IN EACH
REGION////////////////////////////////////
k1n = 0.1*(phi >= 0) + 0*(phi < 0);

```

```

////////////////////////////////////
////////////////////////////////////

```

```

////////////////////////////////////HYDROLITIC DEGRADATION
EQUATIONS(Wang et al. 2008)////////////////////////////////////
varf monomer(Cm,v) =
int3d(Th,optimize=0)((Cm/dt)*v)
+int3d(Th,optimize=0)(Ce*k1n*v)
-int3d(Th)(Ce*sqrt(Cmold)*v)
+int3d(Th,optimize=0)(Dpol*dx(Cm)*dx(v)+Dpol*dy(Cm)*dy(v)+Dpol*dz(Cm)*dz(
v))

```

```

+int3d(Th,optimize=0)((Cmoldt/dt)*v)-int3d(H*K1*Cm/eps)
-int2d(Th,Symmetry)(1e-10*v) //Neumann condition on the symmetry plane
-int2d(Th, 1)(1e-6*v);

//real[int] bm = bbm(0,Vh);

varf ester(Ce,w) =
int3d(Th,optimize=0)((Ce/dt)*w)
+int3d(Th,optimize=0)(Ce*k1n*w)
-int3d(Th)(Ce*sqrt(Cmold)*w) //Initial value of Cm is 0 and is
subsequently calculated in monomer
-int3d(Th,optimize=0)(1e-20*Ce*w)
+int3d(Th,optimize=0)((Ceoldt/dt)*w)
-int3d(H*Cm/Ks)
+on(Free, Ce=g);

////////////////////MATRIX
ASSEMBLY////////////////////////////////////
matrix Am = monomer(Vh, Vh);
matrix Ae = ester(Vh, Vh);

//verbosity=5;

Cmold=Cm0;

////////////////////////////////TIME DEPENDENT
SIMULATION////////////////////////////////
for (int i=0;i<T/dt;i++){

    g=exp(-(k1n)*t);////////////////////////////////ESTHER
CONCENTRATION DIRICHLET CONDITION////////////////////////////////

    real ErrorL2 = 1.;
    real ErrorH1 = 1.;

    /*real[int] solm(Vh.ndof),sole(Vh.ndof), auxm(Vh.ndof),auxe(Vh.ndof);
    Vh w1 = Cmold;

    solm = w1[];*/

    real[int] Bm = monomer(0, Vh);
    real[int] Be = ester(0, Vh);

```

```

//Xh w1 = Cmold;
//Xh w2 = Ceold;
//////////////////////LINEALIZATION OF
DIFFUSIVITY////////////////////////////////////
while(sqrt(ErrorL2+ErrorH1) > 1.e-11)
{
    Dpol=D0n*(1+alpha*(1-((Ce+Cmold))))); // CORREGIDO Actualización
difusividad
    //cout<<"el valor de Dpol es "<<Dpol<<endl;
    //auxm = Bm*solm; auxm += bm;
    Cm[] = Am^-1 * Bm; // Obtaining of Ce from the problem reaction
    //w1[]=solm; Cm=w1;
    error=Cm-Cmold;
    ErrorH1 = int3d(Th)(dx(error)^2+dy(error)^2+dz(error)^2);
    ErrorL2 = int3d(Th)(error^2);
    cout << " Error seminorma H1 = " <<sqrt( ErrorH1) <<endl;
    cout << " Error L2 = " << sqrt(ErrorL2) <<endl;
    cout << " Error norma H1 = " << sqrt(ErrorL2+ErrorH1) <<endl;

    Cmold[]=Cm[];

} // fin iteracion de linealización

for(int j=0;j<Th.nv;j++){
    if (Cm[][j]<0){
        Cm[][j]=abs(Cm[][j]);}
}

Cmoldt[]=Cm[];

//Vh w2 = Ceoldt;
//sole = w2[];
//auxe = Be*sole; auxe += be;

Ce[] = Ae^-1 * Be; // se obtiene Cm , Ce es el calculado con problem
reaction
//w1[]=sole; Ce=w1;

//reaction;

//Ce=exp(-(k1n+Cmoldt)*dt);

Cmavg = int3d(Th)(Cm)/Th.measure;
Ceavg = int3d(Th)(Ce)/Th.measure;

```



```

//Dpolavg=D0n*(1+alpha*(1-((Ceavg+Cmavg)/Ce0)));
//DISPLAYING      COMPUTED      VALUES      IN
//TERMINAL//
cout << " t = " << t << endl;
cout<<"el valor medio de Dpol = "<<Dpolavg<< endl;
cout<<"el valor medio de Cm es "<<Cmavg<< endl;
cout<<"El valor medio previo de Cm es "<<Cmavgprev<<endl;
cout<<"el valor medio de Ce es "<<Ceavg<< endl;
    cout<<"el valor maximo de la conCetracion de monomeros es
"<<Cm[.max<<endl;
    cout<<"el valor maximo de la conCetracion de esteres es
"<<Ce[.max<<endl;

Ceoldt[]=Ce[];

int[int] order = [1, 1]; // order (default in 0)
    //savevtk("result-"+t/dt+".vtk", Th, Cm, Ce,dataname="monomer
ester",order=order);

Ceavg=int3d(Th)(Ce)/Th.measure;

fespace PV(Th, P1);
PV uu=Cm, vv=Ce; // interpolate in P1
string dataname = "monomer ester"; // variable name in Paraview
//int[int] order = [1, 1]; // order (default in 0)

Cmavgprev=Cmavg;
//cout<<"El valor medio previo de Cmavg es "<<Cmavgprev<<endl;

//UPDATING      AVERAGE      MONOMER      CONCENTRATION
//VALUE//
Cmavg=int3d(Th)(Cm)/Th.measure;
cout<<"El valor medio de Cm es "<<Cmavg<<endl;

//COMPUTE DIFFERENCE BETWEEN THE NEWLY AND THE PREVIOUS VALUE
//OF THE MONOMER CONCENTRATION//
real diff=Cmavg-Cmavgprev;
cout<<"La diferencia entre valores medios es "<<diff<<endl;

if (diff >= 0){

    //Cmmax=Cmavg;
    Mncrit=Ceavg;
}

```

```

else {
    //if (diff<0){

        /*if(Cmmax == 0){
            Cmmax=Cmavg;

        }*/

        mtotprev = mtotal;
        mtotal = mtotprev-Km*((Cmmax)^2)*dt;/////////////////////////UPDATING
THE MASS DUE TO THE BULK EROSION////////////////////////////////////
        Xh vol;
        vol = volume;

                                                                    m[] =
vol[]/(Th.measure/mtotal);/////////////////////////COMPUTE      THER
ELEMNTARY VOLUME////////////////////////////////////
        cout<<"el valor actual de masa es "<<m[].max<<endl;
        Vh rho = mtotprev/(Th.measure/volinit);
        Vh rhoprev=rho;
        real Vref=Th.measure/volinit;
        vol[] = vol[]/Vref;
        rho = m/Vref;

        Xh Be=0;/////////////////////////INITIALIZE THE BOUNDARY ELEMENT
IDENTIFIER////////////////////////////////////
        csum=0;/////////////////////////INITIALIZE THE ACCUMULATIVE
NUMBER OF LIQUID NEIGHBOURS////////////////////////////////////
        cmed=0;/////////////////////////INITIALIZE THE AVERAGE VALUE OF
LIQUID NEIGHBOPURS////////////////////////////////////
        nbt=0;/////////////////////////INITIALIZE THE TOTAL NUMBER OF
BOUNDARY ELEMENTS////////////////////////////////////

        ///////////////////////////INITIALIZE      ARRAY      OF      LIQUID
NEIGHBOURS////////////////////////////////////
        real[int] c(Th.nt);
        for (int i = 0; i < Th.nt; i++){
            c[i] = 0;
        }

        ////IDENTIFYING      AND      EVALUATING      BOUNDARY      ELEMENTS
NEIGHBOURHOODS.
        for (int k = 0; k<Th.nbe; ++k){
            int d = Th.be(k).Element;

```

```
if ((Be[][d] == 0 && Th.be(k).label == 1 && Ce1[][d]<Mncrit)){

    Be[][d] = 1;
    boundvol = boundvol+vol[][d];
    cout<<"el volumen de los elementos de contorno es
"<<boundvol<<endl;

    for (int e = 0, ee; e < 4; e++){
        if ((Th[d].adj((ee=e)) < 0)){
            //INCREASE THE VALUE OF C IN CASE OF THE FACE IS IN CONTACT
            WITH THE LIQUID
            c[d]=c[d]+1;
        }
    }
    cout<<"el numero de caras en contacto es "<<c[d]<<endl;
    csum=csum+c[d];
    nbt=nbt+1;
    Vref=Vref-vol[][d];
    //cout<<"el valor medio de caras en contacto es "<<cmed<<endl;
}

}

cout<<"el valor total de caras en contacto es "<<csum<<endl;

if (boundvol > 0 ){
    //REALIZAMOS SUCEсивAS SIMULACIONES DE MONTECARLO PARA EROSIONAR LA
    PIEZA HASTA ALCANZAR EL VOLUMEN DE REFERENCIA
    while(Th.measure/volinit>Vref ){

        cout<<"el volumen actual de la pieza es "<<Th.measure/volinit<<
        " y el volumen de referencia es "<<Vref<<endl;
        cc=0;
        Xh ok=0;
        Xh probabilidad=0;
        cmed=csum/nbt;

        cout<<"the volume of the boundary elements is "<<boundvol<<endl;
        cout<<"the experimental removed volume is "<<Vref<<endl;

        real remainingmass=Vref;
        //if(boundvol>Vref-Thr.measure){

        for (int k=0; k<Th.nbe; ++k){

            if(Th.be(k).label==1){
```

```

//int kk=k;
int d = Th.be(k).Element;
//int d=Tetrahedra;

if (c[d]==4){
    ok[][d]=1;
    cc=cc+1;
}

else{

    //cout<<"el valor medio de caras en contacto es
" << cmed << endl;

    real X=abs((Vref-
Th.measure)/((cmed/(4*Ceavg))*(boundvol/volinit)));
    //probabilidad[][d]=1-exp(-0.00812*t);
    probabilidad[][d]=X*c[d]/(4*Ce1[][d]);
    cout<<"The probability of dissolution is
" << probabilidad[][d] << endl;

    ////////////////////////////////////GENERATE A RANDOM
NUMBER BETWEEN 0 AND ONE TO COMPARE WITH THE PROBABILITY OF EACH ELEMENT
THAT FOLLOWS AN EXPONENTIAL
DISTRIBUTION.////////////////////////////////////

    //GENERATE A RANDOM NUMBER BASED ON AN UNIFORMAL
DISTRIBUTION

    real zz=randreal1();
    //LANDA IS A VALUE USED TO ADJUST THE REMOVAL RATE
OF THE ELEMENTS

    real landa=0.00812;
    real z;
    z=(-log((1-zz))/landa);

    if(z>1){
        while(z>1){
            zz=randreal1();
            z=(-log((1-zz))/landa);
        }
    }

    //cout<<"el valor del numero
aleatorio es " << z << endl;
}
}

```

```

//cout<<"THE VALUE OF THE RANDOM NUMBER IS
"<<z<<endl;

//IDENTIFICATION OF THE ELEMENTS WHICH PROBABILITY
IS LOWER THAN THE RANDOM VALUE Z TO PROCEED OF ITS REMOVAL BY USING THE OK
FUNCTION

        if ((z<probabilidad[][d] && Ce1[][d]<Mncrit &&
Be[][d]==1)){

            ok[][d]=1;
            cc=cc+1;

            //d=d+1;

            cout<<"el tetraedro "<<d<<" se disolvera"<<endl;
            //cout<<"se eliminaran "<<cc<<" elementos"<<endl;
        }

        else{
            //cout<<"el tetraedro "<<Tetrahedra<<" no se
disolvera"<<endl;

            ok[][d]=0;
        }
    }

    //c=0;
}

//}

fespace Nh(Th,[P1,P1,P1]);
varf vn1([nx,ny,nz],[mx,my,mz]) = on(1,2, nx=N.x,ny=N.y,
nz=N.z);

varf disp([dispx,dispy,dispz],[mx,my,mz]) = on(1,2,
dispx=ok*N.x,dispy=ok*N.y, dispz=ok*N.z);
Nh [dispx,dispy,dispz]; // here the DoF of dispx,dispy are
scramble .
dispx[]=disp(0.,Nh);// set [dispx,dispy] DoF.
Nh [nx,ny,nz]; // here the DoF of nx,ny are scramble .
nx[]=vn1(0.,Nh);// set [nx,ny] DoF.

Vh nxx=nx, nyy=ny, nzz=nz;
real dispmax=sqrt(dispx[].max^2+dispy[].max^2+dispz[].max^2);
Vh dispmod=sqrt(dispx^2+dispy^2+dispz^2);

```

```

//savevtk("Th-pcl-
"+i+".vtu",Th,[dispx,dispy,dispz],ok,Be,dataname="normals
erodingtetrahedras");

    cout<<"La componente normal de mayor valor es "<<dispmx;
    /*if((boundvol<Vref-Thr.measure/volinit)){
        cout<<"se procede a la remocion de los elementos de
contorno"<<endl;
        //volloss[m]=(volloss[m]*vol*0.01-boundvol)/(vol*0.01);
        //real minArea=checkmovemesh(Th,[x,y+f]);
        Thr=trunc(Th,Be==1,split=1);
        Th=trunc(Th,Be==0,split=1);
        //volloss[m]=(volloss[m]*vol*0.01-boundvol)/(vol*0.01);
        Thr=Thr+Thrr;
        savemesh(Th,"Th-pcl-mcc-982-"+i+".mesh");

        Thr=change(Thr,rmInternalFaces=1);
        savevtk("Th-residual-pcl-mcc-982-
"+i+".vtu",Thr,ok,Be,dataname="shear-stress");
        savevtk("Th-eroded-pcl-mcc-982-
"+i+".vtu",Th,ok,Be,dataname="shear-stress");
        //if(mpirank==0){
        ofstream file("resultados-PCL-MCC-982.txt", append);
        {
            file<<"
                                "<<i<<"
                                "<<"
                                "<<"
            "<<Thr.nt<<"
                                "<<"
                                "<<"
                                "<<boundvo
l<<"
                                "<<"
                                "<<"
                                "<<Vref<<"
                                "<<"
                                "<<"
                                "Elementos
contorno
"<<"
"<<"
                                "<<Thr.measure<<"
                                "<
"<"
"<<"
                                "<<(Thr.measure/vol)*100<<"
                                "<<endl;}"

        //}
        boundvol=0;
    }*/

    //else{
        cout<<"se procede a la remocion de los elementos que cumplen
con el criterio"<<endl;
        /*Thrr=trunc(Th,ok==1,split=1,label=1);
        Th=trunc(Th,ok==0,split=1,label=1);

        Thr=Thr+Thrr;
        savemesh(Th,"Th-pcl-mcc-982-"+i+".mesh");

        Thr=change(Thr,rmInternalFaces=1);*/

```

```

if(i%5){
    Vh u, ww,hh;
    Xh condition;
    real q =1;
    for(int ii=0; ii<3; ii++)
    {

        solve Poisson(u, ww, solver=CG, eps=5e-5)
        = int3d(Th)(
            100*grad(u)'*grad(ww))
        - int3d(Th)(q * ww)
        + on(1, u=dispmo)
        ;
        plot(u,value=1,ps="figure"+ii+".eps");
        savevtk("Poisson"+ii+".vtk",Th, u,Cm, Ce,
dataname="results monomer ester");
        hh=0;

        hh[]=mshmet(Th,(u[]*.max-
u)^2,normalization=1,aniso=0,nbregul=1,hmin=Th.measure/40,hmax=Th.measure
/10,err=lerr);

        [m11,m21,m22,m31,m32,m33]=[hh,hh,hh,hh,hh,hh];
        lerr=5e-2;//change the level of error
        //Th=mmg3d(Th,metric=h[],hgrad=6);//("oo/Th.o.mesh");
        Th=tetgreconstruction(Th,switch="raAQ",sizeofvolume=h
h*hh*hh/6.);

        //h=hTriangle;
    }
    savemesh(Th,"Adaptedmesh"+i+".mesh");
}

if(dispmx>0){
    Th=movemesh(Th,[x-0.005*dispx/dispmx,y-
0.005*dispy/dispmx,z-0.005*dispz/dispmx]);
}
else{
    Thr=trunc(Th,ok==1,split=1,label=1);
    Th=trunc(Th,ok==0,split=1,label=1);

    Thr=Thr+Thr;
    savemesh(Th,"Th-pcl-mcc-982-"+i+".mesh");
    Thr=change(Thr,rmInternalFaces=1);
}

/*

```

```

        //savevtk("Th-residual-"+i+".vtu",Thr,Cm,dataname="shear-
stress");
        savevtk("Th-eroded-"+i+".vtu",Th,Cm,dataname="shear-
stress");}*/

//if(mpirank==0){

/*if(Thr.measure>0){
    ofstream file("resultados-PCL-MCC-982.txt", append);
    {
        file<<"                "<<n<<"                "<<" "<<"
"<<Thr.nt<<"                "<<" "<<"                "<<bound
vol<<"                "<<" "<<"                "<<Vref<<"                "<<"
"<<"                Monte
Carlo                "<<" "<<"                "<<Thr.measure<<"                "<<
" "<<"                "<<(Thr.measure/vol)*100<<"                "<<endl;}}

    }*/
    //}

//}

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
//}

}
boundvol=0;
}
savemesh(Th,"Th-pcl-mcc-982-"+i+".mesh");
//savevtk("result-"+i+".vtu", Th, uu, vv, order=order);
//savevtk("resultaveraged-"+n+".vtu", Th, Cm, Ce, order=order);
//}

}
}

```



## 15.5. SCRIPT CORRESPONDIENTE AL REFINAMIENTO DE MALLA EN EL MODELO DE REACCIÓN DIFUSIÓN.

```
load "msh3" load "iovtk" load "gmsh" load "tetgen" load "mshmet" load  
"medit" load "mmg" load "ffrandom" load "UMFPACK64"
```

```
mesh3 Th=readmesh3("Model 1_2.mesh");  
plot(Th);
```

```
//PARAMETERS////////////////////////////////////
```

```
//real Ce0=1.73e4;  
real k1 = 0.003;  
real k2 = 0.002;  
real n = 0.5;  
real D = 1.41e-14;  
real l = 0.5;  
real Free = 1;  
real Symmetry = 2;  
real a = 0.6131;  
real c = 0.5077;  
real d = -5.408;  
real e = -7.796;  
real k1n = 0.1;  
real D0n = 1;  
real alpha = 4.5;  
real errm = 0.01;  
real Cmavg;  
real Ceavg;  
real Cmax = 0;  
real Mncrit=0;
```

```
//TIME STEP SETTING////////////////////////////////////
```

```
real t=0;  
real dt=0.01;  
real T=8;
```

```
////////SUBSPACE DEFINITIONS////////////////////////////////////
```

```
fespace Vh(Th,P1);  
fespace Xh(Th,P0);
```

```

fespace PV(Th, P1);

//////////FE FUNCTIONS
DEFINITIONS//////////
Vh Cm, Ce, v, w, Cmold,Cmolddt,Cm0,Ceolddt,Dpol,error,g, u, ww,hh, m, mprev;

Xh vol=volume;
m[]=1/vol[];
real mttotal=1;
real mttotalprev;
real volnorm;
real volinit=Th.measure;
cout<<"El volumen de la pieza es "<<volinit<<endl;
real vol0=Th.measure/volinit;

real Ce0=1;
Ceolddt=Ce0;
Ce=1;

Cm0=0;
Cmolddt=Cm0;
Cmold=Cm0;

//////////DEFINING
MACROS//////////
macro grad(u) [dx(u), dy(u), dz(u)]//
macro div(u) (dx(u)+dy(v)+dz(w)) //

varf monomer(Cm,v) =
    int3d(Th,optimize=0)((Cm/dt)*v)
    +int3d(Th,optimize=0)(Ce*k1n*v)
    +int3d(Th)(Ce*(Cmold^0.5)*v)
    +int3d(Th,optimize=0)(Dpol*dx(Cm)*dx(v)+Dpol*dy(Cm)*dy(v)+Dpol*dz(Cm)
    *dz(v))
    +int3d(Th,optimize=0)((Cmolddt/dt)*v)
    +int2d(Th,Symmetry)(1e-10*Cmold*v) //Neumann condition on the symmetry
plane
    +on(Free, Cm=0);

varf ester(Ce,w) =
    int3d(Th,optimize=0)((Ce/dt)*w)
    +int3d(Th,optimize=0)(Ce*k1n*w)
    +int3d(Th)(Ce*(Cm^0.5)*w) //Initial value of Cm is 0 and is
subsequently calculated in monomer
    -int3d(Th,optimize=0)(1e-20*Ce*w)
    +int3d(Th,optimize=0)((Ceolddt/dt)*w)
    +on(Free, Ce=g);

```

```

ofstream dataunit ( "monomerversustime.txt");
dataunit << " "<<"Time"
          << "Monomer_conc"<< " \n";

{
    ofstream resultsmatrix ( "Resultsmatrix. txt" );
    resultsmatrix << " " <<" Time "<<" , "
                  << " " <<" Cavg "<<" , "
                  << " " <<" Cmax "<<" , "
                  << " " <<" Diffcoef "<<" , "
                  << " " <<" Ceavg "<<" , "
                  << " " <<" Cemax "<<"\n";
}

Cmold=Cm0;

for (int i=0;i<T/dt;i++){

    g=exp(-(k1n)*t);

    matrix Am = monomer(Vh, Vh);
    real[int] Bm = monomer(0, Vh);

    real ErrorL2 = 1.;
    real ErrorH1 = 1.;

    while(sqrt(ErrorL2+ErrorH1) > 1.e-6)
    {

        Dpol=D0n*(1+alpha*(1-((Ce+Cmold))))); //Actualizacion difusividad
        cout<<"el valor maximo de Dpol es "<<Dpol[].max<<endl;

        Cm[] = Am^-1 * Bm; // se obtiene Cm , Ce es el calculado con problem
monomer

        error=Cm-Cmold;

        ErrorH1 = int3d(Th)(dx(error)^2+dy(error)^2+dz(error)^2);
        ErrorL2 = int3d(Th)(error^2);

        cout << " Error seminorma H1 = " <<sqrt( ErrorH1) <<endl;
    }
}

```

```

    cout << " Error L2 = " << sqrt(ErrorL2) <<endl;
    cout << " Error norma H1 = " << sqrt(ErrorL2+ErrorH1) <<endl;

    Cmold=Cm;

}          // fin iteracion de linealización

for(int j=0;j<Th.nv;j++){
    if (Cm[][j]<0){
    Cm[][j]=abs(Cm[][j]);}
}

Cmavg = int3d(Th)(Cm)/Th.measure;

Cmoldt=Cm;

matrix Ae = ester(Vh, Vh);
real[int] Be = ester(0, Vh);

Ce[] = Ae^-1 * Be; // se obtiene Ce , Ce es el calculado con problem
reaction

for(int j=0;j<Th.nv;j++){
    if (Ce[][j]<0){
    Ce[][j]=0;}}

    cout<<"el valor maximo de la conCetracion de monomeros es
"<<Cm[.max<<endl;
    cout<<"el valor maximo de la conCetracion de esterres es
"<<Ce[.max<<endl;

ofstream dataunit ( "monomerversustime.txt",append);
dataunit << " " << t
    << " " << Cmavg << "\n";

Ceoldt=Ce;

Ceavg = int3d(Th)(Ce)/Th.measure;

ofstream resultsmatrix ( "Resultsmatrix. txt" ,append);
resultsmatrix << " " << t <<" , "
    << " " << Cmavg <<" , "
    << " " << Cm[.max <<" , "
    << " " << Dpol[.max <<" , "
    << " " << Ceavg <<" , "

```

```
<< " " << Ce[] .max << "\n";
```

```
real Dpolavg=D0n*(1+alpha*(1-((Ceavg+Cmavg)/Ce0)));
```

```
cout << " t = " << t << endl;
```

```
cout<<"el valor medio de Dpol = "<<Dpolavg<< endl;
```

```
cout<<" concentracion promedio de monomeros es "<<Cmavg<<endl;
```

```
cout<<" concentracion promedio de esteres es "<<Ceavg<<endl;
```

```
cout<<"el valor maximo de la concentracion de monomeros es  
"<<Cm[] .max<<endl;
```

```
cout<<"el valor maximo de la concentracion de esteres es  
"<<Ce[] .max<<endl;
```

```
////////////////////////////////////MONTE CARLO  
SIMULATION////////////////////////////////////  
////
```

```
Vh uu=Cm, vv=Ce; // interpolate in P1
```

```
string dataname = "monomer ester"; // variable name in Paraview
```

```
int[int] order = [1, 1]; // order (default in 0)
```

```
savevtk("result-"+i+".vtu", Th, uu, vv, order=order);
```

```
real Cmavgprev=Cmavg;
```

```
Cmavg=int3d(Th)(Cm)/vol;
```

```
cout<<"El valor medio de Cm es "<<Cmavg<<endl;
```

```
real diff=Cmavg-Cmavgprev;
```

```
cout<<"La diferencia entre valores medios es "<<diff<<endl;
```

```
if (diff>0){
```

```
    Cmax=Cmavg;
```

```
    Mncrit=Ceavg;
```

```
}
```

```
else
```

```
{
```

```
problem Poisson(u,v,solver=CG) = int3d(Th)( grad(u)'*grad(v) ) //  
' ) for emacs
```

```
-int3d(Th)( 1*v ) + on(1,u=Ce[] .max-Ce);
```

```

cout<<"el valor de Ce es "<<Ce[][i]<<endl;

//MESH ADAPTATION
for(int ii=0; ii<3; ii++)
{
    Poisson;
    cout <<" u min, max = " << u[].min << " "<< u[].max << endl;
    Vh h;

    h[]=mshmet(Th,(u[].max-
u)^2,normalization=1,aniso=0,nbregul=1,hmin=Th.measure/40,hmax=Th.measure
/10,err=0.01);
    cout <<" h min, max = " << h[].min << " "<< h[].max << " " <<
h[].n << " " << Th.nv << endl;
    // FFC3: add 3D view parameters
    plot(u,wait=0,fill=0,boundary=0,CutPlane=0,ShowMeshes=1,LabelCo
lors=0);
    errm*= 0.8;// change the level of error
    cout << " Th" << Th.nv < " " << Th.nt << endl;
    Th=tetgreconstruction(Th,switch="raAQ",sizeofvolume=h*h*h/6.);
    //medit("U-adap-iso-"+ii,Th,u,wait=1);

//INTERPOLATE TERMS
Cm=Cm;
Ce=Ce;
g=g;
Cmold=Cmold;
Ceoldt=Ceoldt;
uu=uu;
vv=vv;

}
Am.resize(Vh.ndof, Vh.ndof);
Bm.resize(Vh.ndof);
Ae.resize(Vh.ndof, Vh.ndof);
Be.resize(Vh.ndof);
savevtk("result-refined"+i+".vtu", Th, uu, vv, order=order);
//Th=trunc(Th, Ce<0.8, split=1, label=[1,1,2,2]);
//savevtk("resultaveraged-"+i+".vtu", Th, Cm1, Ce1, order=order);
}
t = t + dt;
}

```

## 15.6. SCRIPT CORRESPONDIENTE A LA GENERACIÓN DE LA MALLA ESTRUCTURADA PARA LA CELDA UNITARIA.

```
SetFactory("OpenCASCADE");
//+
Geometry.OCCAutoFix = 0;
//+
Geometry.Tolerance = 1e-2;

Merge "Scaffold_92_Porosity_3.brep";
//+
//Merge "Scaffold_92_Porosity_3_1.brep";
BooleanFragments{ Volume{:}; Delete; }{ }
//+

//+

Merge "Fluid_92_Porosity_3.brep";
BooleanFragments{ Volume{:}; Delete; }{ }

Merge "Fictitious_92_Porosity_3.brep";
Geometry.OCCAutoFix = 0;
//BooleanFragments{ Volume{:}; Delete; }{ }Geometry.OCCFastUnbind = 0;
BooleanFragments{ Volume{:}; Delete; }{ }

Volume_index= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
37, 38, 39,
    40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57,
58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76,
77, 78,
    79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96,
97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111,
112, 113,
    114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127,
128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,
143, 144,
    145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158,
159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173,
174, 175,
```

```

176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189,
190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204,
205, 206,
207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220,
221, 222, 223, 224};

```

```

a = 0;
b = 0;
Physical Volume("Scaffold", 1) = {Volume_index[a]};
For i In {a + 1: a + 83}
    Physical Volume("Scaffold", 1) += {Volume_index[i]};
    b = i;
EndFor

```

```

a = b + 1;
Physical Volume("Fluid", 2) = {Volume_index[a]};
For i In {a + 1: a + 17}
    Physical Volume("Fluid", 2) += {Volume_index[i]};
    b = i;
EndFor

```

```

a = b + 1;
Physical Volume("Fictitious", 3) = {Volume_index[a]};
For i In {a + 1: a + 121}
    Physical Volume("Fictitious", 3) += { Volume_index[i]};
    b = i;
EndFor

```

```

a = 150;
b = 5;
c = 70;
d = 10;
e = 8;
f = 7;
//+
Transfinite Curve {1288, 1315, 1299, 1275, 1003, 1042, 1305, 1004, 1019,
768, 769, 774, 775, 967, 966, 965, 770, 964, 627, 631, 482, 156, 147, 186,
185, 507, 505, 508, 506, 188, 187, 152, 159, 484, 960, 621, 625, 773, 772,
771, 963, 776, 777, 961, 962, 1017, 985, 1025, 1061, 1088, 1114, 1129,
1148, 986, 1219, 1013, 1213, 1054, 1195, 1201, 1207, 1012, 1018, 971, 970,
969, 923, 922, 919, 918, 637, 633, 586, 924, 968, 356, 347, 386, 385, 615,
613, 616, 614, 388, 387, 352, 639, 588, 643, 927, 972, 359, 973, 920, 921,
925, 926, 975, 974, 1016, 1015, 1217, 1223, 1210, 1205, 1199, 1055, 1014}
= a Using Progression 1;
//+
Transfinite Curve {813, 811, 830, 753, 755, 730, 732, 814, 816, 831, 752,
756, 729, 733, 821, 817, 835, 751, 749, 728, 726, 820, 818, 834, 747, 748,

```



```
723, 725, 857, 855, 867, 865, 802, 787, 784, 796, 901, 905, 884, 760, 762,
738, 740, 900, 898, 883, 767, 763, 745, 741, 902, 904, 885, 759, 758, 737,
735, 895, 897, 881, 766, 764, 744, 742, 792, 790, 805, 800, 951, 937, 934,
945, 917, 913, 893, 684, 682, 661, 659, 909, 907, 888, 686, 714, 688, 716,
663, 705, 702, 665, 916, 955, 914, 942, 892, 940, 948, 680, 681, 656, 658,
910, 912, 889, 685, 689, 662, 666, 842, 840, 825, 654, 650, 678, 674, 844,
872, 846, 874, 827, 863, 860, 646, 645, 670, 668, 843, 847, 826, 647, 649,
671, 673, 837, 839, 823, 653, 698, 651, 696, 677, 710, 708, 675, 997, 1145,
1133, 1119, 1099, 1073, 1036, 999, 1037, 984, 1024, 998, 1167, 1161, 1154,
1143, 1131, 1038, 976, 1139, 1126, 1111, 1084, 1057, 1049, 979, 1164, 1158,
1150, 1137, 1123, 1050, 978, 1051, 977, 1026, 989, 1029, 992, 1285, 1272,
1263, 1257, 1251, 1030, 991, 1320, 1309, 1296, 1282, 1270, 1031, 990, 1032,
1002, 1041, 1008, 1293, 1280, 1267, 1261, 1255, 1046, 1007, 1044, 1009,
1324, 1313, 1302, 1291, 1278, 1045, 1303, 1329, 1226, 1186, 1122, 1156,
1279, 1312, 1216, 1160, 1075, 1134, 1298, 1326, 1225, 1185, 1117, 1152,
1273, 1310, 1214, 1157, 1066, 1127, 1243, 1264, 1295, 1300, 1248, 1266,
1238, 1252, 1269, 1276, 1240, 1254, 1112, 1149, 1110, 1115, 1118, 1153,
1058, 1124, 1056, 1062, 1072, 1130, 1340, 1104, 1333, 1077, 1331, 1064,
1339, 1096, 1227, 1234, 1229, 1232, 1187, 1190, 1175, 1183} = b Using
Progression 1;
```

```
//+
```

```
Transfinite Curve {981, 982, 994, 995, 1318, 1098, 983, 993, 996, 980,
1338, 1094, 1322, 1095, 1287, 1221, 1165, 1141, 1332, 1069, 1070, 1311,
1274, 1215, 1159, 1128, 1068, 1307, 1113, 1071, 1209, 1297, 1151, 1265,
1189, 1203, 1259, 1245, 1172, 1230, 1087, 1138, 1284, 1086, 1173, 1231,
1239, 1253, 1060, 1125, 1059, 1271, 1176, 1197, 1035, 1033, 1053, 1052,
1028, 1023, 1034, 1022, 988, 1005, 1006, 987, 1319, 1105, 1000, 1001, 1010,
1011, 1342, 1106, 1325, 1109, 1224, 1294, 1168, 1147, 1081, 1335, 1314,
1080, 1135, 1218, 1162, 1281, 1308, 1082, 1120, 1301, 1212, 1155, 1079,
1268, 1206, 1192, 1236, 1249, 1262, 1181, 1144, 1101, 1090, 1290, 1256,
1180, 1242, 1235, 1132, 1074, 1277, 1063, 1200, 1184, 1039, 1048, 1047,
1040, 1020, 1021, 1027, 1043} = c Using Progression 1;
```

```
//+
```

```
Transfinite Curve {1292, 1323, 1341, 1222, 1166, 1103, 1108, 1146, 1316,
1102, 1286, 1317, 1321, 1337, 1220, 1163, 1091, 1093, 1097, 1140, 1202,
1228, 1246, 1258, 1283, 1289, 1204, 1233, 1250, 1260, 1085, 1136, 1170,
1193, 1083, 1089, 1100, 1142, 1178, 1194, 779, 780, 788, 808, 162, 171,
509, 778, 49, 89, 441, 690, 691, 692, 718, 720, 782, 783, 791, 809, 165,
178, 511, 781, 54, 91, 443, 693, 694, 695, 719, 721, 852, 853, 877, 879,
262, 299, 559, 851, 365, 378, 619, 931, 932, 933, 941, 959, 849, 850, 876,
878, 257, 297, 557, 848, 362, 371, 617, 928, 929, 930, 938, 958} = d Using
Progression 1;
```

```
//+
```

```
Transfinite Curve {824, 828, 522, 822, 566, 887, 890, 891, 864, 543, 599,
949, 841, 836, 906, 911, 845, 838, 908, 915, 873, 553, 609, 957, 819, 856,
537, 812, 603, 896, 903, 952, 815, 810, 894, 899, 832, 833, 866, 526, 547,
```

829, 562, 593, 880, 882, 886, 944, 1273, 1310, 1331, 1214, 1157, 1064, 1066, 1127, 1298, 1326, 1339, 1225, 1185, 1096, 1117, 1152, 1303, 1329, 1340, 1226, 1186, 1104, 1122, 1156, 1279, 1312, 1333, 1216, 1160, 1075, 1077, 1134, 1227, 1243, 1264, 1295, 1300, 1234, 1248, 1266, 1229, 1238, 1252, 1269, 1276, 1232, 1240, 1254, 750, 754, 803, 463, 495, 746, 390, 421, 644, 648, 652, 697, 727, 731, 795, 485, 722, 724, 431, 667, 669, 672, 676, 709, 761, 765, 807, 470, 501, 757, 415, 437, 679, 683, 687, 715, 801, 491, 427, 706, 739, 743, 734, 736, 655, 657, 660, 664, 1112, 1149, 1187, 1110, 1115, 1118, 1153, 1190, 1058, 1124, 1175, 1056, 1062, 1072, 1130, 1183} = e Using Progression 1;

//+

**Transfinite Curve** {1343, 1107, 1334, 1076, 1336, 1092, 1330, 1065, 875, 291, 379, 943, 862, 274, 377, 939, 861, 542, 598, 947, 871, 870, 954, 956, 1304, 1067, 1328, 1116, 1306, 1078, 1327, 1121, 858, 854, 950, 953, 859, 267, 370, 936, 868, 282, 369, 935, 869, 548, 594, 946, 1188, 1191, 1169, 1179, 1171, 1177, 1174, 1182, 1208, 1211, 1244, 1247, 1237, 1241, 1196, 1198, 804, 786, 785, 797, 799, 789, 793, 806, 712, 711, 700, 699, 713, 717, 704, 703, 169, 170, 496, 74, 59, 422, 59, 707, 436, 83, 66, 701, 500, 179, 177, 798, 496, 170, 169, 794} = f Using Progression 1;

//+

Surface\_index = {869, 851, 837, 842, 866, 863, 846, 855, 859, 1180, 1001, 1178, 1072, 1183, 1201, 1202, 1156, 1074, 1149, 999, 994, 1030, 960, 954, 1028, 745, 1008, 856, 930, 1011, 944, 1175, 980, 1172, 986, 852, 845, 862, 847, 848, 849, 861, 850, 860, 844, 853, 1066, 858, 854, 857, 1069, 838, 762, 768, 809, 816,

818, 868, 753, 839, 840, 1143, 841, 1135, 843, 867, 865, 864, 681, 695, 1191, 1199, 955, 721, 709, 961, 990, 802, 798, 817, 815, 814, 810, 808, 807, 801, 795, 514, 1169,

1181, 1182, 710, 1200, 1136, 785, 1150, 711, 1075, 1145, 682, 288, 1157, 292, 489, 492, 677, 678, 679, 680, 793, 698, 707, 708, 800, 799, 697, 797, 796, 1179, 794, 750,

749, 748, 747, 746, 744, 743, 751, 720, 696, 722, 723, 724, 766, 780, 767, 1177, 765, 764, 763, 761, 1173, 693, 694, 754, 752, 1192, 981, 962, 956, 1195, 1000, 995,

987, 1067, 1013, 1029, 1031, 1184, 1071, 1073, 1197, 935, 932, 943, 1010, 945, 940, 934, 989, 1196, 1168, 1194, 215, 218, 1198, 274, 701, 700, 699, 1190, 282, 1193,

691, 275, 277, 287, 289, 290, 291, 685, 1012, 931, 933, 1131, 1137, 942, 946, 756, 1144, 741, 1122, 735, 702, 967, 728, 727, 726, 725, 1163, 776, 1068, 1065, 1061,

1070, 786, 784, 783, 782, 781, 779, 778, 777, 775, 912, 774, 920, 773, 772, 771, 770, 769, 1167, 760, 705, 1009, 1186, 706, 1004, 1007, 717, 975, 982, 713, 1174,

1176, 703, 704, 988, 1160, 488, 820, 819, 813, 433, 436, 485, 811, 493, 496, 499, 500, 508, 516, 519, 970, 941, 936, 921, 911, 732, 804, 1170, 803, 716, 730, 729,

1171, 791, 731, 787, 714, 1189, 806, 805, 1187, 689, 736, 959, 734, 733, 737, 1006, 1002, 1003, 1005, 1185, 1188, 998, 993, 712, 715, 976, 974, 718, 969, 968, 719,

966, 1158, 469, 462, 458, 1164, 425, 421, 1159, 473, 1155, 1152, 821, 1148, 1165, 1166, 1130, 1129, 520, 1162, 638, 634, 623, 615, 563, 547, 522, 1123, 683, 505, 684,

686, 812, 487, 486, 758, 919, 687, 922, 688, 1161, 690, 759, 692, 918, 757, 949, 952, 755, 742, 740, 739, 1062, 1121, 1117, 1113, 792, 790, 789, 1064, 1063, 738, 788,

1060, 1058, 1055, 910, 913, 914, 388, 530, 576, 541, 666, 668, 828, 542, 826, 834, 589, 836, 389, 592, 591, 590, 565, 548, 546, 545, 561, 562, 564, 578, 566, 575, 577,

655, 665, 667, 669, 670, 619, 613, 614, 616, 617, 618, 620, 621, 622, 624, 633, 635, 636, 637, 650, 128, 132, 313, 325, 543, 363, 390, 387, 379, 365, 544, 375, 1056,

1054, 1053, 1049, 1037, 1034, 1059, 1057, 652, 648, 581, 574, 573, 572, 571, 649, 651, 647, 653, 654, 656, 570, 569, 568, 567, 559, 596, 626, 611, 603, 632, 639, 640,

641, 642, 53, 643, 595, 594, 593, 586, 644, 645, 646, 114, 130, 129, 127, 122, 117, 131, 115, 56, 551, 383, 329, 332, 391, 597, 598, 599, 600, 835, 830, 827, 823,

538, 536, 527, 386, 398, 397, 394, 582, 585, 658, 663, 674, 675, 676, 673, 1024, 1153, 1015, 1151, 1032, 1033, 1035, 1036, 996, 985, 991, 992, 997, 1154, 1096, 1077,

1083, 1133, 1090, 1091, 1092, 1095, 628, 1097, 1107, 1116, 1115, 1114, 1111, 1112, 1147, 1146, 1142, 1139, 979, 584, 583, 631, 612, 630, 625, 833, 832, 831, 829,

610, 627, 825, 824, 629, 822, 602, 587, 588, 601, 604, 605, 606, 607, 608, 609, 948, 549, 657, 550, 664, 552, 659, 660, 553, 560, 947, 661, 950, 951, 662, 554, 953,

558, 557, 957, 958, 555, 556, 579, 671, 580, 672, 938, 928, 925, 1078, 1106, 1108, 1109, 1110, 923, 917, 1120, 1105, 908, 1125, 905, 973, 1128, 1132, 1134, 1082,

924, 926, 1101, 964, 1099, 1098, 1094, 1093, 1089, 927, 929, 937, 1085, 939, 1017, 1023, 1022, 1021, 1141, 1140, 1138, 1016, 1014, 984, 983, 978, 977, 1027, 1020,

1076, 1079, 1052, 1051, 1050, 1080, 1048, 1047, 1081, 1043, 1041, 1040, 1039, 1038, 899, 883, 882, 881, 880, 1127, 898, 885, 1124, 879, 878, 900, 897, 896, 895,

894, 893, 892, 884, 891, 890, 889, 888, 887, 886, 876, 875, 874, 873, 872, 877, 871, 870, 1044, 1045, 1046, 1026, 1025, 1019, 1018, 1042, 915, 1088, 1100, 1102,

```
1103, 1104, 1118, 1119, 916, 1087, 909, 907, 906, 1126, 904, 903,
902, 972, 971, 901, 965, 963, 1084, 1086};
```

```
For i In {0: 732}
    Transfinite Surface {Surface_index[i]} Left;
EndFor
```

```
For i In {0:223}
    Transfinite Volume {Volume_index[i]};
EndFor//+
```

```
//+
```

```
//+
```

```
Physical Surface("Section", 2) = {698, 724, 732, 715, 728, 706, 687, 544,
633, 569, 647, 556, 625, 602, 640, 614, 692, 702, 682, 711, 730, 726, 719,
606, 660, 674, 667, 652, 642, 618, 689, 701, 678, 761, 777, 755, 670, 676,
656, 664, 612, 646, 624, 734, 770, 744, 748, 796, 804, 789, 782, 772, 737,
620, 636, 645, 648, 608, 629, 550, 572, 562, 742, 776, 754, 800, 786, 792,
806, 583, 600, 592, 566, 596, 554, 574, 739, 775, 750, 765, 778, 758, 694,
704, 684, 548, 570, 560, 578, 594, 588, 598, 753, 695, 745, 681, 609, 605,
553, 557};
```

```
//+
```

```
Physical Surface("Free Surface", 1) = {638, 389, 398, 672, 630, 580, 599,
759, 487, 500, 802, 768, 757, 486, 499, 762, 801, 709, 729, 718, 662, 675,
712, 731, 721, 790, 805, 798, 628, 671, 397, 388, 634, 589, 673, 666, 659,
788, 803, 795, 587, 597, 576, 668};
```

```
//+
```

```
Physical Surface("Fluid section", 3) = {809, 818, 816, 836, 835, 832, 817,
820, 821, 810, 813, 812, 831, 833, 824, 828, 827, 825};
```

```
//+
```

```
Physical Surface("Fictitious Boundary", 4) = {1072, 1066, 1054, 1060, 1048,
1149, 1135, 1121, 1090, 1112, 1093, 1076, 1084, 1106, 1042, 1100, 1014,
1032, 1038, 1018, 977, 991, 1002, 1008, 1028, 903, 923, 947, 910, 934, 954,
989, 918, 940, 960, 944, 986, 999, 974, 930, 994, 980, 966, 969, 1001, 951,
927, 907, 915, 937, 957, 869, 842, 837, 863, 846, 855, 866, 859, 851, 895,
963, 875, 870, 899, 879, 888, 901, 891, 884, 1178, 1172, 1162, 1185, 1191,
1201, 1194, 1180, 1168, 1158, 1151, 1138, 1124, 1118, 1146, 1132, 1196,
1188, 1175, 1165, 1154, 1141, 1183, 1199, 1202, 1127, 971, 983, 996, 1005,
1011, 1030, 1074, 1069, 1063, 1035, 1021, 1040, 1057, 1051, 1045, 1025,
1103, 1087, 1080, 1098, 1109, 1115, 1095, 1129, 1143, 1156};
```

## 15.7. SCRIPT CORRESPONDIENTE AL MODELO INTEGRADO APLICADO A UNA CELDA UNITARIA CON DISCRETIZADO NO ESTRUCTURADO.

```
load "msh3" load "iovtk" load "gmsh" load "tetgen" load "mshmet" load
"medit" load "mmg" load "ffrandon" load "distance" load "mshmet"

//include "MonteCarlo.idp"

mesh3
Tht=readmesh3("Fictitious_domain_80_porosity_nonstructured.mesh");//Read
the mesh of the fictitious domain
mesh3 Ths;//Initialisation of the solid mesh
mesh3 Thf;//Initialisation of the nonsolid mesh
mesh3 Thcell;//Initialisation of the fluid mesh

////////////////////LABELS////////////////////////////////////
int omegas = 1;
int omegaf = 2;
int OMEGAF = 3;
int gammasf = 1;
int gammasf2 = 5;
int gammas = 2;
int gammaf = 3;
int GAMMA = 4;
////////////////////

plot(Tht);
//real k1=0.003;

real k2 = 2*10^1.5;
real n = 0.5;
real D = 1.41e-14;
real l = 0.5;
real Free = 1;
real Symmetry = 2;
real a = 0.6131;
real c = 0.5077;
real d = -5.408;
real e = -7.796;
real delta;
real Km = 11;
real boundvol = 0;
real csum = 0;
```

```

real cmed = 0;
real nbt = 0;
real cc = 0;
real Mncrit = 0.6;
real eps = 1e-8;
real eps1 = 1e-6;
real D0n = 0.45;
real Ks = 1e-2;
real lambda = 0.45;//parameter for modulate the material compaction in
the first stages of the degradation, in order to get a porosity
decreasing
real beta = 0.5;
real beta1 = 2;
real epsilon = 2e-4;
real Ker = 10;
real B = 0.9;
real PorosityBulkErosion;
real PorositySurfaceErosion;
real PorosityInit = 0.8;
real porosity = PorosityInit;
real tTrans = 2.0;
real Ce0real = 1.73e-5;
real xc0 = 0.8;
real xcMax = 1;
real nu = 0.6;
real nu1 = 0.25;
bool eroInit = 0;
real k1n = 0.1;
real alpha1;
real gamma1 = 1;
int j;
bool nonsection;

//////////////////////////////////////REACTION
DIFFUSION//////////////////////////////////////
//////////////////////////////////////

//NORMALIZED VALUES

/*
real k1n=k1/(k2*Ce0^n);
real D0n=D0/((Ce0^n)*k2*1^2);
*/

```

```

real t = 0;
real dt = 0.009;
real T = 200;
real tReal = t/(Ce0real^0.5*k2);

fespace Vh(Tht,P13d);
fespace Sh(Tht, P0);
fespace Xh(Ths,P0);

real lerr = 0.05;
Vh Cm, Ce, v, w, Cmold, Cmoldt, Cm0, Ce0, Ceold, Ceoldt, Dpol, error, g,
m, b, phi1, phi1h, phi2, phi2h, phi1old, phi2old, C, Nx, Ny, Nz, h,
nabla, okt, length, d1, e1;
Sh G, H;
real mttotal = 1;
real mttotalinit = 1;
real mttotalprev;
real volnorm;
real Cavg, Ceavg, Cavgprev, Dpolavg;
Cavgprev = 0;

ofstream remainingvolume ( "remainingvolume.txt");
remainingvolume<<" "<<"time"<<" "<<"simulated
volume"<<" "<<"reference
volume"<<" "<<"difference"<<" "<<"porosity"<<"\n";

Xh Ce1, Cm1, ok;

////////MACRO DEFINITIONS: GRADIENT AND DIVERGENCE
macro grad(u) [dx(u), dy(u), dz(u)]//
macro div(u) (dx(u)+dy(v)+dz(w)) //
//Vh sol=exp(-(k1n+Cm^n)*t);

//////////INITIALIZATION OF THE
AVERAGE NORMALISED MONOMER
CONCENTRATION//////////
Cavg = int3d(Tht)(Cm*v)/Tht.measure;
cout<<"el valor maximo de Cavg es "<<Cavg<<endl;
//////////DISTANCE
COMPUTATION//////////

varf vb(uaux, vaux) = on(gammasf, gammas, uaux =
1);//////////Setting uaux to 1 in the interfacce

```

```

Vh ub = vb(0, Vh, tgv=1);
plot(ub, value = 1);

ub[] = ub[] ? 0:1;
plot(ub,value = 1);

distance(Tht, ub, phi1[], distmax=100); //compute absolute distance(phi)
respect to the interface
phi1 = -phi1*(region == OMEGAF)- phi1*(region == omegaf) + phi1*(region
==omegas); //assign signed distance respect to the interface (phi < 0 in
OMEGA and phi>0 in omega)

plot(phi1);

varf symmetry(sect,sectaux) = on(gammas, gammaf, sect = 1);
Vh sect1 = symmetry(0, Vh, tgv=1);
sect1[] = sect1[] ? 0:1;
plot(sect1, value = 1);

Tht = change(Tht, rmInternalFaces = 1); //REMOVE PHYSICAL INTERFACES
H = 1*(phi1>=0) + 0*(phi1<0);
G = (phi1>=- epsilon && phi1< epsilon); //CHARACTERISTIC FUNCTION
//length = mshmet(Tht, phi1, normalization = 1, aniso = 0, hmin = 0.01,
hmax = 0.08, err = 1e-3);
real hmin = 0.02;
real hmax = 0.1;
//length = max(hmin, min(hmax,abs(phi)));

//Tht = tetgreconstruction(Tht, switch = "raAQ", sizeofvolume =
length*length*length/6);
//Tht = mmg3d(Tht, metric = length[]);
savemesh(Tht, "refinedmesh.mesh");
//ASSIGN RHO AND MU VALUES IN EACH
REGION

savevtk("signed_distance.vtk", Tht, phi1, G, sect1,
dataname="signed_distance characteristic_function section");

//DISTANCE COMPUTATION CELL UNIT(ASSIGNING
PROPERTIES TO DIFFUSIVITY)

varf vbf(uaux, vaux) = on(gammaf, gammasf, gammas, uaux =
1); //Setting uaux to 1 in the interface

Vh ubf = vbf(0, Vh, tgv=1);

```



```

plot(ubf, value = 1);

ubf[] = ubf[] ? 0:1;
plot(ubf,value = 1);

distance(Tht, ubf, phi2[], distmax=100);//compute absolute distance(phi1)
respect to the interface
phi2 = -phi2*(region == OMEGAF) + phi2*(region == omegaf) - phi2*(region
== omegas);////assign signed distance respect to the interface (phi < 0
in OMEGA and phi>0 in omega)

plot(phi2);

Tht = change(Tht, rmInternalFaces = 1); //REMOVE PHYSICAL INTERFACES
H = 1*(phi1>=0) + 0*(phi1<0);
G = (phi1>= 0 && phi1< epsilon);//////////CHARACTERISTIC FUNCTION
//////////
//length = mshmet(Tht, phi1, normalization = 1, aniso = 0, hmin = 0.01,
hmax = 0.08, err = 1e-3);

//length = max(hmin, min(hmax,abs(phi1)));

//Tht = tetgreconstruction(Tht, switch = "raAQ", sizeofvolume =
length*length*length/6);
//Tht = mmg3d(Tht, metric = length[]);
//savemesh(Tht, "refinedmesh.mesh");
savevtk("signed_distance_fluid.vtk", Tht, phi1, phi2, G, k1n, sect1,
dataname="signed_distance1 signed_distamce2 characteristic_function_solid
non_catalitic_constant");

//////////ASSIGN RHO AND MU VALUES IN EACH
REGION//////////
//k1n = 0.1*(phi1 >= 0) + 0*(phi1 < 0);

//////////
//////////

Ths = trunc(Tht, phi1 >= 0, label = 1 );
Thcell = trunc(Tht, phi1 >= 0 || phi2 >=0, label = 2);
savemesh(Thcell, "CeldaUnitaria.mesh");
//Thf = trunc(Tht, phi < 0, label = 2);
real VolScaffoldInit = Ths.measure;
real VolScaffold = VolScaffoldInit;

```

```

real VolCell = Thcell.measure;
real PorositySurfaceErosionInit = (VolCell - VolScaffold)/VolCell*100;
PorositySurfaceErosion = PorositySurfaceErosionInit;

varf ConvectLevelScaffold(phi1, phi1h) =

    int3d(Th)(phi1*phi1h/dt)
+ int3d(Th)(convect([(Nx)*C, (Ny)*C, (Nz)*C], -dt, phi1old)*phi1h);
//+ int3d(Th)(eps*dx(phi1)*dx(phi1h)+eps*dy(phi1)*dy(phi1h));

varf ConvectLevelFluid(phi2, phi2h) =

    int3d(Th)(phi2*phi2h/dt)
+ int3d(Th)(convect([- (Nx)*C, - (Ny)*C, - (Nz)*C], -dt, phi2old)*phi2h);
//+ int3d(Th)(eps*dx(phi2)*dx(phi2h)+eps*dy(phi2)*dy(phi2h));

//////////HYDROLITIC DEGRADATION
EQUATIONS(Wang et al. 2008)//////////
varf monomer(Cm , v) =

    int3d(Th, optimize = 0)((Cm/dt)*v)
+ int3d(Th, optimize = 0)(H*Ce*k1n*v)
+ int3d(Th, optimize = 0)(H*Ce*sqrt(Cmold)*v)
+ int3d(Th, optimize =
0)(Dpol*dx(Cm)*dx(v)+Dpol*dy(Cm)*dy(v)+Dpol*dz(Cm)*dz(v))
+ int3d(Th, optimize = 0)((Cmoldt/dt)*v)
//+ int3d(Th, optimize = 0)(1e-10*Cm*v)
- int3d(Th, optimize = 0)(G*(e1-0)*v)
//+ int3d(Th, optimize = 0)(epsilon*dx(Ce)*dx(v) + epsilon*dy(Ce)*dy(v)
+ epsilon*dz(Ce)*dz(v)) // Diffusion Term
+ on(GAMMA, Cm = 0); //Dirichlet condition of periodicity on external
boundary

//real[int] bm = bbm(0,Vh);

varf ester(Ce, w) =

    int3d(Th, optimize = 0)((Ce/dt)*w)
+ int3d(Th, optimize = 0)(H*Ce*k1n*w)
+ int3d(Th, optimize = 0)(H*Ce*sqrt(Cm)*w) //Initial value of Cm is
0 and is subsequently calculated in monomer
- int3d(Th, optimize = 0)(1e-20*Ce*w)
+ int3d(Th, optimize = 0)((Ceoldt/dt)*w)
//- int3d(Th, optimize = 0)((H*k1n/Ks)*Ce*w)

```

```

- int3d(Tht, optimize = 0)(G*(d1 - g)*w) // STRONGER
ENFORCEMENT
//+ int3d(Tht, optimize = 0)(epsilon*dx(Ce)*dx(w) +
epsilon*dy(Ce)*dy(w) + epsilon*dz(Ce)*dz(w)) // Diffusion Term
+ on(GAMMA, Ce = 0);

////////////////////MATRIX
ASSEMBLY////////////////////////////////////

//matrix Ae = ester(Vh, Vh, solver = CG);

//verbosity=5;
Ce0 = 1;
Ce = Ce0*(phi1 >= 0) + 0*(phi1 < 0);
savevtk("Distribucion inicial.vtk", Tht, Cm, Ce, dataname = "monomer
ester");
phi1old = phi1;

Cmold = 0;
Ceoldt = Ce;
Cm = 0;
real Cmmax = 0;
real alpha = 4.5;
Vh f;
//Cmold = Cm0;

////////////////////TIME DEPENDENT
SIMULATION////////////////////////////////////
for (int i = 0;i<T/dt;i++){

    g=exp(-(k1n)*t);////////////////////ESTHER
CONCENTRATION DIRICHLET CONDITION////////////////////////////////////

    cout<<"el valor maximo de g es "<<g[.max<<endl;

    real ErrorL2 = 1.;
    real ErrorH1 = 1.;

    phi1old = phi1;

    matrix Am = monomer(Vh, Vh, solver = CG);
    real[int] Bm = monomer(0, Vh);

```

```

Vh h = hTriangle;
//k1n = 0.1*(phi1 >= 0) + 0*(phi1 < 0);
d1 = Ce;
e1 = Cm;
H = 0*(phi1 < -0.02 ) + 1*(phi1 >= -0.02);
G = (phi1 >=-0.02 && phi1 < h && sect1> 0.4);//////////CHARACTERISTIC
FUNCTION ////////////
    savevtk("characteristicfunction.vtk", Tht, phi1, G,
dataname="signed_distance characteristic_function");

//////////LINEALIZATION OF
DIFFUSSIVITY//////////

while(sqrt(ErrorL2+ErrorH1) > 1.e-8)
{

    cout<<"el valor maximo de Cmoldd es "<<Cmoldd[.max<<endl;
    Dpol = D0n*(1+alpha*(1-((Ce+Cmoldd)))); // CORREGIDO
Actualización difusividad
    savevtk("difusividad.vtk", Tht, Dpol, dataname ="difusividad");
    cout<<"el valor de Dpol es "<<Dpol[.max<<endl;
    Cm[ ] = Am^-1 * Bm; // Obtaining of Ce from the problem reaction
    cout<<"el valor maximo de Cm es "<<Cm[.max<<endl;
    //w1[ ]=solm; Cm=w1;
    error = Cm-Cmoldd;
    cout<<"el error entre Cm y el valor previo es
"<<error[.max<<endl;
    ErrorH1 = int3d(Tht)(dx(error)^2+dy(error)^2+dz(error)^2);
    cout<<"el error obtenido en H1 es "<<ErrorH1<<endl;
    ErrorL2 = int3d(Tht)(error^2);
    cout << " Error seminorma H1 = " << sqrt( ErrorH1) <<endl;
    cout << " Error L2 = " << sqrt(ErrorL2) <<endl;
    cout << " Error norma H1 = " << sqrt(ErrorL2+ErrorH1) <<endl;
    Cmoldd[ ] = Cm[ ];

} // fin iteracion de linealización

for(int j = 0;j < Tht.nv;j++){

    if (Cm[[j]<0){

        Cm[[j] = 0;
    }

}

Cmolddt[ ] = Cm[ ];

```

```

Cmold[] = Cm[];

matrix Ae = ester(Vh, Vh, solver = CG);
real[int] Be = ester(0, Vh);

Ce[] = Ae^-1 * Be; // se obtiene Cm , Ce es el calculado con problem
reaction
//Ce = g*(phi1 >= 0 && phi1 < 0.01) + Ce*(phi1 >= 0.01) + 0*(phi1 <
0);
//w1[]=sole; Ce=w1;
cout<<"el valor maximo de Ce es "<<Ce[].max;

for(int j = 0;j<Tht.nv;j++){

    if (Ce[][j] < 0){

        Ce[][j] = 0;

    }

    /*if (Ce[][j] > 1){

        Ce[][j] = 1;

    }*/

}

//reaction;
Ceoldt[] = Ce[];
tReal = t/(Ce0real^0.5*k2);
//Ce=exp(-(k1n+Cmoldt)*dt);
//Ths = trunc(Tht, phi1 >= 0, label = 1);
Thf = trunc(Tht, phi1 < 0, label = 1);
Thcell = trunc(Tht, phi1 >= 0 || phi2 >=0, label = 2);

savemesh(Ths, "solido-"+i+".mesh");
savemesh(Thf, "fluido-"+i+".mesh");

fespace Vvh(Ths, P13d);
Vvh Cmint, Ceint, sectint;

```

```

Cmint = Cm;
Ceint = Ce;
sectint = sect1;

savevtk("Interpolated values"+i+".vtk", Ths, Cmint, Ceint, sectint,
dataname ="monomer ester section");

Ceavg = int3d(Ths)(Ceint)/Ths.measure;

Dpolavg=D0n*(1+alpha*(1-((Ceavg+Cmavg)/Ce0)));
//////////DISPLAYING COMPUTED VALUES IN
TERMINAL//////////
cout << " t = " << t <<endl;

int[int] order = [1, 1]; // order (default in 0)

if(i % 5 == 0){

    savevtk("result-"+i+".vtk", Tht, Cm, Ce,dataname="monomer
ester",order = order);

}

Cmavgprev = Cmavg;
//cout<<"El valor medio previo de Cmavg es "<<Cmavgprev<<endl;
Cmavg = int3d(Ths)(Cmint)/Ths.measure;
//////////UPDATING BULK POROSITY//////////
//PorosityBulkErosion = -lambda*(1 - mtotal / mtotalinit);
real Xc = xc0 + (xcMax-xc0)*(1-exp(-(0.5*t+2*(1-Ceavg))));
//////////UPDATING AVERAGE MONOMER CONCENTRATION
VALUE//////////
if (Xc > xcMax){
    Xc = xcMax;
}

cout<<"El valor medio de Cm es "<<Cmavg<<endl;
cout<<"el valor medio de Dpol = "<<Dpolavg<< endl;
cout<<"el valor medio de Cm es "<<Cmavg<< endl;
cout<<"El valor medio previo de Cm es "<<Cmavgprev<<endl;
cout<<"el valor medio de Ce es "<<Ceavg<< endl;

```

```

    cout<<"el valor maximo de la concentracion de monomeros es
"<<Cm[.max<<endl;
    cout<<"el valor maximo de la concentracion de esteres es
"<<Ce[.max<<endl;

    ////////////COMPUTE DIFFERENCE BETWEEN THE NEWLY AND THE PREVIOUS
    VALUE OF THE MONOMER CONCENTRATION////////////////////////////////////
    real diff = Cmax - Cmaxprev;
    cout<<"La diferencia entre valores medios es "<<diff<<endl;

    if (diff >= 0 && eroInit == 0){
        tTrans = t;
        Cmax = Cmax;
        //Mncrit = Ceavg;
        cout<<"el valor critico del peso molecular es "<<Mncrit<<endl;
    }

    else {
        eroInit = 1;

        ////////////COMPUTING INITIAL VOLUME OF THE
        UNIT CELL////////////////////////////////////
        cout<<"check volume"<<endl;
        cout<<"El volumen de la pieza es "<<VolScaffoldInit<<endl;
        real vol0 = Ths.measure/VolScaffoldInit;
        cout<<"end check
volume"<<endl;    //////////////////////////////////////
        //////////////////////////////////////

        mtotprev = mtot;
        mtot = mtotprev -
Km*((Cmax)^2)*dt;/////////////////////////////////UPDATING THE MASS DUE TO THE BULK
EROSION/////////////////////////////////
        Xh vol, Ce1, Cm1;
        cout<<"start interpolating concentrations"<<endl;
        Cm1 = Cm;
        Ce1 = Ce;
        cout<<"end interpolating concentrations"<<endl;
        cout<<"compute volume"<<endl;
        ///////////////////////////////////TETRAHEDRAL
        VOLUMES/////////////////////////////////
        vol = volume;

```

```

        m[] =
vol[]/(Ths.measure/mtotal);//////////////////////////COMPUTE THE
ELEMENTARY VOLUME//////////////////////////

        cout<<"el valor actual de masa es "<<m[].max<<endl;
        Vh rho = mtotallprev/(Ths.measure/VolScaffold);
        Vh rhoprev = rho;
        real Vref = Ths.measure/VolScaffold;
        vol[] = vol[]/Vref;
        rho = m/Vref;

        Xh Bound = 0;//////////////////////////INITIALIZE THE BOUNDARY
ELEMENT IDENTIFIER//////////////////////////
        csum = 0;//////////////////////////INITIALIZE THE ACCUMULATIVE
NUMBER OF LIQUID NEIGHBOURS//////////////////////////
        cmed = 0;//////////////////////////INITIALIZE THE AVERAGE VALUE
OF LIQUID NEIGHBOURS//////////////////////////
        nbt = 0;//////////////////////////INITIALIZE THE TOTAL NUMBER
OF BOUNDARY ELEMENTS//////////////////////////

        ////////////////////////////INITIALIZE ARRAY OF LIQUID
NEIGHBOURS//////////////////////////
        real[int] c(Ths.nt);
        for (int i = 0; i < Ths.nt; i++){

                c[i] = 0;

        }

        ////IDENTIFYING AND EVALUATING BOUNDARY ELEMENTS
NEIGHBOURHOODS.

        cout<<"el valor total de caras en contacto es "<<csum<<endl;
        cout<<"La diferencia entre el volumen inicial y el actual es
"<<VolScaffoldInit - Ths.measure<<endl;
        real voldiff;
        if (VolScaffoldInit - Ths.measure > 1e-3){
                voldiff = VolScaffoldInit - Ths.measure;
        }
        else{
                voldiff =0;
        }

        ok = 0;

```



```

    cout<<"el volumen a disolver es "<<VolScaffoldInit*(1-exp(-B*(t-
tTrans)) - voldiff) > 1e-3<<" y estamos en el tiempo "<< t<< endl;

    //REALIZAMOS SUCEATIVAS SIMULACIONES DE MONTECARLO PARA EROSIONAR
    LA PIEZA HASTA ALCANZAR EL VOLUMEN DE REFERENCIA

    //while(VolScaffoldInit*(1-exp(-B*(t-tTrans)) - voldiff) > 1e-
3    && t>0){

        nabla = (dx(phi1))^2+(dy(phi1)^2+dz(phi1)^2);
        Nx = dx(phi1)/(sqrt(nabla + eps1^2));
        Ny = dy(phi1)/(sqrt(nabla + eps1^2));
        Nz = dz(phi1)/(sqrt(nabla + eps1^2));
        savevtk("normal"+ i +".vtk", Tht, [Nx, Ny, Nz], dataname =
"normal_vector_magnitude");

        for (int k = 0; k<Ths.nbe; ++k){

            int d = Ths.be(k).Element;
            int v0 = Ths.be(k)[0];
            int v1 = Ths.be(k)[1];
            int v2 = Ths.be(k)[2];
            nonsection = (sectint[][v0]>0.35) && (sectint[][v1]>0.35)
&& (sectint[][v2]>0.35);
            cout<<"el valor logico de nonsection es
"<<nonsection<<endl;

            //cout<<"el identificador del elemento es "<<d<<endl;

            if ((Bound[][d] == 0 && Ths.be(k).label == 1 && Ce1[][d]
< Mncrit && nonsection)){//If those elements comply the condition of
elegibiity of surface boundary and moleculatr weight boundary element
                Bound[][d] = 1;
                boundvol = boundvol+vol[][d];
                cout<<"el volumen de los elementos de contorno es
"<<boundvol<<endl;
                for (int e = 0, ee; e < 4; e++){

                    if ((Ths[d].adj((ee=e)) < 0)){
                        //INCREASE THE VALUE OF C IN CASE OF THE FACE IS IN
CONTACT WITH THE LIQUID
                        c[d]=c[d]+1;
                    }
                }
            }
        }
    }

```

```

    }

    cout<<"el numero de caras en contacto es
" << c[d] << endl;

    csum = csum + c[d];
    nbt = nbt + 1;
    Vref = Vref - vol[][d];

}

}
ok = 0;
cout<<"el volumen actual de la pieza es
" << Ths.measure/VolScaffoldInit << " y el volumen de referencia es
" << Vref << endl;
cc = 0;
phi1old = phi1;

Xh probabilidad = 0;

cmed = csum/nbt;

cout<<"el valor medio de caras en contacto es " << cmed << endl;
cout<<"the volume of the boundary elements is " <<
boundvol << endl;
cout<<"the experimental removed volume is " << Vref << endl;

real remainingmass = Vref;
//if(boundvol>Vref-Thr.measure){

srandomdev();
for (int k = 0; k < Ths.nbe; ++k){

    if(Ths.be(k).label == 1){

        //int kk=k;
        int d = Ths.be(k).Element;
        int v0 = Ths.be(k)[0];
        int v1 = Ths.be(k)[1];
        int v2 = Ths.be(k)[2];

```

```

        nonsection = (sectint[][v0]>0.35) &&
(ssectint[][v1]>0.35) && (sectint[][v2]>0.35);
        cout<<"el valor logico de nonsection es
"<<nonsection<<endl;

        if (nonsection == 0){

            continue;
        }

        //int d=Tetrahedra;

        if (c[d] == 4){

            ok[][d] = 1;
            cc = cc+1;

        }

        else{

            //cout<<"el valor medio de caras en contacto es
"<<cmed<<endl;

            real X = (VolScaffoldInit*(2-exp(-B*(tReal-
tTrans)))-Ths.measure)/VolScaffoldInit;

            probabilidad[][d] = X*c[d]/(4);

            cout<<"el decremento de volumen teorico es
"<<(exp(-B*(t-tTrans))-1)<<endl;
            cout<<"el numero promedio de caras en contacto es
"<<cmed<<endl;

            cout<<"el volumen de elementos de contorno es
"<<boundvol<<endl;

            cout<<"el numero de caras en contacto es
"<<c[d]<<endl;

            cout<<"el valor del parametro B es "<<B<<endl;

            cout<<"The probability of dissolution is
"<<probabilidad[][d]<<endl;

            ////////////////////////////////////GENERATE A RANDOM
NUMBER BETWEEN 0 AND ONE TO COMPARE WITH THE PROBABILITY OF EACH ELEMENT

```

```

    THAT FOLLOWS AN EXPONENTIAL
    DISTRIBUTION.///////////////////////////////////////////////////

    //GENERATE A RANDOM NUMBER BASED ON AN UNIFORMAL
    DISTRIBUTION
    real zz = randreal1();
    //LANDA IS A VALUE USED TO ADJUST THE REMOVAL
    RATE OF THE ELEMENTS
    real landa = 0.00812;
    real z;
    z = (-log((1-zz))/landa);

    if(z > 1){
        while(z > 1){
            zz = randreal1();
            z = (-log((1-
zz))/landa);
            //cout<<"el valor del numero aleatorio es
"<<z<<endl;
        }
    }
    cout<<"THE VALUE OF THE RANDOM NUMBER IS
"<<z<<endl;

    //IDENTIFICATION OF THE ELEMENTS WHICH
    PROBABILITY IS LOWER THAN THE RANDOM VALUE Z TO PROCEED OF ITS REMOVAL BY
    USING THE OK FUNCTION

    if ((z<probabilidad[][d] && Ce1[][d] < Mncrit &&
Bound[][d] == 1)){

        ok[][d] = 1;
        cc = cc+1;

        cout<<"el elemento "<<d<<" se
disolvera"<<endl;

        //cout<<"se eliminaran "<<cc<<"
elementos"<<endl;
    }

    else{
        cout<<"el tetraedro "<<d<<" no se
disolvera"<<endl;

        ok[][d] = 0;
    }
}

```

```

    }
}

//Tht = Ths + Thf;

/*
//ADAPTACION DE MALLA PARA SUAVIZAR LA SUPERFICIE DE
EROSION(OPCIONAL)
Tht = change(Tht, rmInternalFaces = 1);
length = max(hmin, min(2*hmax,abs(phi)));
Tht = mmg3d(Tht, metric = length[]);
length = max(hmin, min(2*hmax,abs(phi)));
Tht = mmg3d(Tht, metric = length[]);
*/

//hTriangle = hTriangle;
savevtk("checkbefore-"+i+".vtk",Tht, phi1, okt, dataname =
"phi erodibility");
Vh height = hTriangle;
/////////////////////////INTERPOLATING COMPUTED
VALUES////////////////////////////////////
/*Cm = Cm;
Cmold = Cmold;
Ce = Ce;
Ceold = Ceold;
okt = okt;
Cm1 = Cm1;
Ce = Ce;
phi1 = phi1;
phi1old = phi1old;
C = C;*/

/////////////////////////RESIZING MATRICES
Am.resize(Vh.ndof, Vh.ndof);
Bm.resize(Vh.ndof);
Ae.resize(Vh.ndof, Vh.ndof);
Be.resize(Vh.ndof);

okt = ok;

matrix Als = ConvectLevelScaffold(Vh, Vh, solver = CG);
real[int] Bls = ConvectLevelScaffold(0, Vh);

matrix Alf = ConvectLevelFluid(Vh, Vh, solver = CG);

```

```

real[int] Blf = ConvectLevelFluid(0,Vh);

for (int k = 0; k < Tht.nv; k++){

    if (okt[][k] == 0 || !nonsection){

        C[][k] = 0;

    }
    else{

        C[][k] = 10;

    }

}

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
//}

savevtk("height"+i+".vtk", Tht, height, C, dataname = "height
ErodibleHeight");

/*phi1[] = Als^-1*BlS;
phi2[] = Alf^-1*Blf;*/

phi1=convect([-C*Nx,-C*Ny, -C*Nz],dt,phi1);
phi2=convect([C*Nx, C*Ny, C*Nz],dt,phi2);
savevtk("checkafter-"+i+".vtk",Tht, phi1, phi2, okt, dataname
= "PhiScaffold Phifluid PotentialErodibleElements");

//Tht = tetgreconstruction(Tht, switch = "raAQ", sizeofvolume
= length*length*length/6);
//Tht = mmg3d(Tht, metric = length[]);

//savemesh(Tht, "refinedmesh"+i+".mesh");

Ths = trunc(Tht, phi1 > 0, label = 1);
VolScaffold = Ths.measure;
PorositySurfaceErosion = (1 - (VolScaffold/VolCell))*100;
voldiff = VolScaffoldInit - Ths.measure;

```

```

    real CheckBeforePorosity = (B*t - voldiff)/VolScaffoldInit;

    PorositySurfaceErosion = (1 - (VolScaffold/VolCell))*100;
    voldiff = VolScaffoldInit - Ths.measure;
    real CheckAfterPorosity = (B*t - voldiff)/VolScaffoldInit;

    //ofstream remainingvolume ( "remainingvolume.txt", append);
    //remainingvolume
    <<"          "<<Ths.measure<<"          "<<Vref<<"          "<<
    CheckAfterPorosity<<"          "<<"\n";

    cout<<"la diferencia relativa entre el valor simulado y el de
referencia es "<< B*t - voldiff<< endl;
    savemesh(Ths,"Eroded"+ i +".mesh");
    savevtk("current_front_erosion"+ i +".vtk", Ths, phi1,
dataname = "interface");

    //} //fin bucle while

    boundvol=0;

}
Ce = g*(G == 1) + Ce*(G == 0);

if (i == 0 || i%5 == 0){

    //savemesh(Ths,"Eroded"+ i/5 +".mesh");
    savevtk("results-"+ i +".vtk", Tht, Cm, Ce, phi1, okt, dataname =
"monomer ester front_erosion erodibility");
    savevtk("Eroded"+ i +".vtk", Ths, Cm1, Ce1, ok, dataname =
"monomer ester erodibility");

}

t = t + dt;
tReal = t/(Ce0real^0.5*k2);
cout<<"el tiempo actual es "<<t<<endl;
Cm = Cm*(phi1 < 0) + Cm*(phi1 >= 0);
Ce = 0*(phi1 < 0) + Ce*(phi1 >= 0);
Ce = g*(G == 1) + Ce*(G == 0);

real phi1avg= int3d(Tht)(phi1)/Tht.measure;

```

```

real hmax = h[.max];
//dt = 0.2*hmax/abs(phi1[.max/phi1avg);
real alpha1 = 0.09;
cout<<"el valor del paso de tiempo es "<<dt<<endl;
real FCrist = (1+exp(4.5*(Xc-xc0)));
real FMonom = (Cmavgprev/(Cmavg+1e-7));
//real FMonom = (1 - tanh(6 * Cm));
real Porosity1 = abs(PorositySurfaceErosion*(1-((1-
(2/FCrist))*FMonom^(10))));
real Porosity2 = abs(PorositySurfaceErosion*(1-
alpha1*(2/FCrist)*FMonom));
real tanhTrans = 1 * (1 + tanh(2 * (tReal - 2)));
porosity = (1 - tanhTrans) * Porosity1 + tanhTrans * Porosity2;

/*if (diff>=0 && eroInit == 0){
    porosity = abs(PorositySurfaceErosion*(1-((1-
abs(2/FCrist))*0.8*FMonom^500));

}

else{
    real alpha1 =0.1;
    real FMonom = (Cmavgprev/(Cmmax));
    porosity = abs(PorositySurfaceErosion*(1-
alpha1*(2/FCrist)*FMonom));
}*/
//
ofstream remainingvolume ( "remainingvolume.txt", append);
remainingvolume
<<"                "<<tReal<<"                "<<Ths.
measure<<"                "<<Thcell.measure<<"
                "<<Cmavgprev<<"                "<<"
                "<<Xc<<"
"<<FCrist<<"                "<<FMonom<<"
                "<<porosity<<"                "<<"\n";

}

```