



**Estudio del método de Schneiders para la generación de mallas cuadrangulares no estructuradas y su aplicación en la simulación de la electrofisiología del corazón.**

Estiven Flórez Castro

Director: Ph.D. César Augusto Acosta Minoli

PERTINENTE CREATIVA INTEGRADORA

 @uniquindio  unquindioconectada  unquindioconectada

[www.uniquindio.edu.co](http://www.uniquindio.edu.co)



UNIVERSIDAD  
DEL QUINDÍO



# Universidad del Quindío

FACULTAD DE CIENCIAS DE LA EDUCACIÓN

LICENCIATURA EN MATEMÁTICAS

ESTUDIO DEL MÉTODO DE SCHNEIDERS PARA LA GENERACIÓN DE MALLAS  
CUADRANGULARES NO ESTRUCTURADAS Y SU APLICACIÓN EN LA SIMULACIÓN  
DE LA ELECTROFISIOLOGÍA DEL CORAZÓN.

Estiven Flórez Castro

Director: Ph.D. César Augusto Acosta Minoli

Modalidad: Trabajo de Grado bajo la modalidad de Desarrollo

Área de profundización: Matemática aplicada

Grupo de Investigación: Grupo de Estudio y Desarrollo de Software GEDES

Junio 2022

1

## 1. Resumen

El presente trabajo pretende estudiar y aplicar el código SpecMesh2D en la representación de la electrofisiología de secciones planas del corazón, permitiendo la comprensión del algoritmo utilizado en la generación de mallas y la interpretación de los resultados. El trabajo de grado cuenta con una introducción en la que se expone la importancia de la generación de mallas en la solución de ecuaciones diferenciales parciales en dominios bidimensionales por métodos espectrales, así como el por qué del uso de elementos cuadráticos. Luego se presenta la importancia de la generación de mallas en diferentes investigaciones que se encuentran en el estado del arte, además de los objetivos y cómo se realizó el trabajo de grado utilizando la metodología SCRUM, la cual se utilizó en la versatilidad del tiempo para tener una revisión constante al trabajo. Finalmente los resultados de los mallados propuestos y simulaciones a partir de los ceros de los polinomios de Legendre encontrados en cada mallado.

**Palabras clave:** Generación de mallas, elementos cuadrangulares, electrofisiología cardíaca, método de Schneiders.

## 2. Abstract

The present work aims to study and apply the SpecMesh2D code in the representation of the electrophysiology of flat sections of the heart, allowing the understanding of the algorithm used in the generation of meshes and the interpretation of the results. The job of degree has an introduction in which the importance of mesh generation in the solution of partial differential equations in two-dimensional domains by spectral methods is exposed, as well as the reason for the use of quadratic elements. Then the importance of mesh generation is presented in different investigations that are in the state of the art, in addition to the objectives and how the work of degree was be carried out using the SCRUM methodology, which will be used in the versatility of time to have a constant review to the work. Finally the results of the proposed grids and simulations based on the zeros of the Legendre polynomials found in each mesh.

**keywords:** Mesh generation, quadratic elements, cardiac electrophysiology, Schneiders method.

## Índice

<b>1. Resumen</b>	<b>2</b>
<b>2. Abstract</b>	<b>3</b>
<b>3. Introducción</b>	<b>1</b>
<b>4. Planteamiento del Problema</b>	<b>2</b>
<b>5. Objetivos</b>	<b>3</b>
5.1. Objetivo general . . . . .	3
5.2. Objetivos específicos . . . . .	3
<b>6. Justificación</b>	<b>4</b>
<b>7. Estado del arte</b>	<b>5</b>
<b>8. Marco conceptual</b>	<b>10</b>
8.1. Métodos espectrales . . . . .	10
8.2. Generación de mallas . . . . .	11
8.3. Mallas estructuradas . . . . .	12
8.4. Mallas estructuradas y métodos algebraicos . . . . .	12
8.5. Mallas no estructuradas . . . . .	13
8.5.1. Métodos de Superposición-Deformación de Retícula (Scheneiders) . .	13
8.5.2. Método de Delaunay-Voronoi . . . . .	15
8.5.3. Método de Frente de Avance . . . . .	16
8.6. Interpolación por splines cúbicos . . . . .	17
8.7. Condiciones de continuidad paramétrica . . . . .	18
8.8. Cuadratura Gaussiana . . . . .	19
8.9. Anatomía del corazón . . . . .	20

8.9.1. Fibras Musculares . . . . .	23
<b>9. Metodología SCRUM</b>	<b>24</b>
<b>10.Resultados</b>	<b>26</b>
10.1. Generación de mallas . . . . .	26
10.1.1. SpecMesh2D . . . . .	26
10.1.2. Cygwin . . . . .	27
10.1.3. Code::blocks . . . . .	27
10.1.4. VisIt . . . . .	27
10.1.5. MATLAB . . . . .	28
10.2. Modelo y dominio . . . . .	28
10.3. Resolución de la ecuación diferencial . . . . .	33
<b>11.Conclusiones</b>	<b>38</b>
<b>12.Referencias bibliográficas</b>	<b>39</b>
<b>13.Anexos</b>	<b>42</b>

### 3. Introducción

La generación de mallas hace parte del proceso de los métodos numéricos de ecuaciones diferenciales parciales, como por ejemplo los elementos finitos o elementos espectrales, manejando un dominio geométrico bidimensional o tridimensional, este dominio se divide en unos subdominios conocidos como elementos, formando la geometría computacional; es una disciplina de las ciencias de la computación, logrando sus inicios con la tesis doctoral de Shamos, (1978) donde se realizó una expansión del estudio de la geometría computacional, abarcando áreas de la matemática como: álgebra, topología, ecuaciones diferenciales, probabilidad y análisis numérico.

Kopriva, (2011) y Morcillo, (2000), mencionan que las dificultades más comunes para implementar un método en el proceso de discretización, es tener un bajo coste computacional, una buena calidad en la malla, las alternancias de manejar elementos de la malla o en su defecto discretizar elementos con geometrías complejas, dando lugar, a que actualmente no exista un método que se destaque frente a los demás.

En este trabajo de grado se aplicará una secuencia metodológica apoyada por el software SpecMesh2D, que es autoría de Kopriva, (2011) la cual ha sido utilizada por el grupo de investigación GEDES de la Universidad del Quindío, con el fin de explorar y construir mallas en dominios bidimensionales, generando algunos ejemplos y aplicándolos a la solución de ecuaciones diferenciales parciales. Con la exploración de este software, se pretende comprender los métodos de mallado, elaboración del mallado y presentar una serie de ejemplos aplicados a las ecuaciones de problemas de monodominio hiperbólico en el estudio de la electrofisiología del corazón bidimensional.

Se utiliza el procedimiento de discretización basado en el método de Schneiders, y se aplica la técnica en la discretización de dominios bidimensionales que corresponde a planos

transversales de la fisiología del corazón; además, de validar cuales de los mallados en la resolución de una ecuación diferencial parcial obteniendo varias simulaciones del pulso cardíaco en tiempos diferentes.

#### 4. Planteamiento del Problema

Existe un grupo de profesionales en el área de las matemáticas y la computación que se han dedicado a la modelación de fenómenos, usando la generación automática de mallas como herramienta para la aproximación de ecuaciones diferenciales parciales mediante el método de elementos finitos; para esto es necesario discretizar el dominio geométrico en formas simples de dos dimensiones como los triángulos y los cuadriláteros o de tres dimensiones como los tetraedros y hexaedros, siendo estos graficados en software computacionales, llevando en su mayoría al objeto a la formulación de problemas, generando mallas y dando solución a ecuaciones numéricas imprescindibles.

Los diversos sucesos han originado la necesidad de generar mallas, entre ellas la geografía y la cartografía, también se usa en la simulación física para la solución numérica de ecuaciones diferenciales, siendo esta una de las principales necesidades del uso de las mallas. Por esto, es preciso la realización de ejemplos y uso del código mediante los diversos tipos de dominios geométricos bien sean complejos o no complejos, para graficar mediante ecuaciones paramétricas o *splines* cúbicos.

En investigaciones como la de Paya, (2013) se presenta una de las aplicaciones de la generación de mallas para el área de medicina, donde se hizo uso del programa Scilab para simular los modelos matemáticos del bidominio y desfibrilación, que mediante el ingreso de datos para emular la señal por medio de ecuaciones, se pudieron visualizar y detectar los problemas del corazón, concluyendo que con la aplicación del método de volúmenes finitos se puede presentar una mayor aproximación a la realidad, asociándolo a la geometría de

malla y a las funciones *test*. Similar a lo que se realizará en esta investigación con el ejemplo de la generación de mallas del tejido del corazón.

De lo anterior surge la siguiente pregunta de investigación: ¿Cómo discretizar mediante la generación de mallas cuadrangulares no estructuradas, un dominio bidimensional que represente diferentes estructuras internas del corazón, para utilizarla en la aproximación de soluciones de modelos de la actividad eléctrica cardíaca mediante un modelo de alto orden de precisión?

## 5. Objetivos

### 5.1. Objetivo general

Aplicar el método de Schneiders en la discretización de dominios bidimensionales en secciones transversales de imágenes del corazón para la aproximación numérica de ecuaciones diferenciales parciales, mediante métodos de alto orden de precisión.

### 5.2. Objetivos específicos

1. Estudiar las técnicas y fundamentos que se implementaron en el código SpecMesh2D.
2. Implementar el código SpecMesh2D en los procesos de discretización de dominios bidimensionales.
3. Presentar y validar un conjunto de ejemplos aplicados a la ecuación del problema monodominio hiperbólico.

## 6. Justificación

Para lograr un avance en la ejecución y construcción de mallas, se hace necesario tener algunos conocimientos que permiten precisar su desarrollo y este caso se hablará del lenguaje de programación Fortran, donde no es únicamente necesaria la fundamentación teórica, para esto se requiere de un equipo computacional adaptable que permita la compilación del ejecutable del código SpecMesh2D, en el cual se pueden graficar los dominios mediante *splines* cúbicos o parametrización de curvas, logrando modelar los diferentes tipos de dominios geométricos conocidos.

Para esta investigación se tuvo en cuenta, la fundamentación teórica de las ecuaciones diferenciales parciales, sin embargo, sus soluciones no serán desarrolladas, solamente se hace uso de las mismas para la implementación de la geometría computacional que permite la definición del dominio usado; Por otra parte, debido a la capacidad de las mallas en la ilustración geométrica de los diversos campos investigativos y de las ciencias diferentes a las matemáticas, se ilustrarán algunos casos reales y modelamientos matemáticos que finalizan en una mallado.

Además, este trabajo de grado tuvo como objetivo analizar la discretización bidimensional utilizado en la aproximación numérica mediante métodos de alto orden de precisión, lo que permite que la malla sea más cercana a la realidad, implementando una metodología que evidencia el uso adecuado del software para extraer información que finalmente, da la posibilidad de definir los parámetros que serán estudiados y graficados.

Finalmente, el trabajo de grado presentó una forma de entendimiento y conocimiento acerca de la codificación de software por medio de ejemplos, mostrando la importancia del uso de estas herramientas computacionales para el proceso de la generación de mallas.

## 7. Estado del arte

Al realizar una revisión de las investigaciones acerca de la generación automática de mallas se encontró que:

En la investigación de Quispe Ccapacca, (2018) explica que es la generación de mallas:

*La aplicación del método de elementos finitos requiere discretización de los dominios en pequeños polígonos de forma triangular o cuadrilateral. Este procedimiento es llamado “generación de mallas triangulares o cuadrilaterales” según corresponda. La generación de malla requiere que las curvas que delimitan los dominios hayan sido subdivididas en varios segmentos.*

Según, Melo Lagos, (2008) en su tesis afirma que:

*La generación de mallas de 2D (dos dimensiones), tiene como enfoques más comunes a usar triángulos o cuadriláteros. Los triángulos son los polígonos más simples y poseen importantes propiedades (por ejemplo, sobre el tamaño de sus lados, medianas y puntos notables) y relaciones con circunferencias (inscritas, circunscritas). Por otra parte, los cuadriláteros son figuras más complejas que los triángulos y, a diferencia de éstos, pueden no estar completamente contenidos en un plano (utilizando la definición genérica de cuadriláteros, que no exige que sean polígonos).*

*Sin embargo, el uso de mallas compuestas por cuadriláteros es un enfoque que ha ganado adeptos, pues se considera que estas mallas son más precisas y flexibles. Una dificultad de abordar este enfoque es que no existe abundante literatura al respecto como en el caso de las mallas de triángulos.*

Ahora, Bern and Plassmann, (2000) en su artículo hace un aporte respecto a los tipos de malla:

Tipo I: *Una malla estructurada es aquella en la que todos los vértices interiores son topológicamente iguales.*

Tipo II: *Una malla no estructurada es aquella en la que los vértices pueden tener vecindarios locales que varían arbitrariamente.*

Tipo III: *Una malla híbrida está formada por una serie de pequeñas mallas estructuradas, combinadas en un conjunto de patrones no estructurados*

Luego, define sus características de la siguiente manera:

*En general, las mallas estructuradas ofrecen simplicidad y un fácil acceso de datos, mientras que las mallas no estructuradas ofrecen una adaptabilidad de malla más conveniente y un mejor encaje para dominios complicados, (como se podría esperar, las mallas híbridas se encuentran en un punto intermedio).*

En su trabajo Elkinn Adrián Calderón, (2014) nombra los métodos para generar las mallas:

*En la etapa de pre-proceso de los Métodos de Elementos Finitos (MEF) o de los Métodos de Elementos Espectrales (MES) se sitúa la generación de la malla junto a su regularización y pre-condicionamiento; esta etapa es la primera y tal vez, más importante fase, pues de ella dependen los recursos de memoria necesaria, la velocidad de generación de la solución y la exactitud de los resultados numéricos obtenidos. En ella, el dominio se divide en pequeños subdominios llamados elementos.*

A partir de esto, algunos autores han comparado el uso de mallas cuadrangulares y triangulares; algunos de ellos son:

- Lavender and Hayhurst, (1986) dice que si el tamaño de los elementos cuadriláteros se controlan cuidadosamente, y se generan de ese modo cuadriláteros bien formados, se puede lograr un aumento en la eficiencia de las mallas.
- Liang et al., (2010) afirma que los triángulos normalmente no se desempeñan tan bien como los cuadriláteros, lo cual es especialmente cierto para los análisis lineales.
- Schneiders and Bünten, (1995) presentan un ejemplo en planificación y optimización de los procesos de formación del metal, donde utilizan elementos cuadriláteros en dos dimensiones o elementos hexaédricos en tres dimensiones para la discretización en elementos finitos de la ecuación de este problema. Ellos concluyeron que estos pares de elementos son más precisos en predicción de grandes deformaciones plásticas que los elementos triangulares o tetraedros.

Respecto a la aplicación de las mallas en otros campos diversos a la matemática aplicada se encontró que:

En el trabajo de Cherrez Monar and Miranda Masaquiza, (2020) el cual trata de "la simulación de los modelos matemáticos bidominio y desfibrilización para estudiar el restablecimiento del ritmo cardíaco" donde su objetivo era simular dichos modelos matemáticos ingresando datos para emular la señal mediante ecuaciones donde se lograra visualizar y detectar los problemas del corazón, se concluye que con la aplicación del método de volúmenes finitos se puede presentar una mayor aproximación a la realidad.

Por otra parte Jonas et al., (2009) en su investigación: "Generación de mallas de elementos finitos para arterias coronarias a partir de imágenes ultrasonido intravascular (IVUS)";

presenta una herramienta computacional para la detección de las paredes de arterias coronarias a partir de series de imágenes tomadas por un sensor de ultrasonido intravascular (IVUS). Debido a las características de los datos analizados, previamente a la segmentación se aplica un algoritmo de filtrado espacial que permite la reducción de ruido, cuya conclusión de las pruebas experimentales realizadas muestran que el algoritmo permite obtener mallas de calidad superior a la que se puede generar a partir de otros tipos de imágenes, con una excelente definición de la sección arterial. Finalmente, en base a los contornos individuales detectados es posible reconstruir una serie de mallas de superficie que describen la estructura arterial en los diferentes ciclos cardiacos de sístole y diástole.

Además, Paya, (2013) en su tesis doctoral: “Un método de diferencias finitas generalizadas para simular la actividad eléctrica de un tejido cardíaco” muestra un mallado distinto al de otros *softwares*, incluso del que será usado en esta investigación, evidenciando la dificultad de realizar mediciones directas en la electrocardiología, siendo dicha tesis una motivación para la simulación numérica de modelos cardiacos, ya que aplica un esquema explícito con el método de diferencias finitas generalizadas y mínimos cuadrados móviles (MDFG-MCM) al caso de la modelación monodominio de problemas de conductividad eléctrica.

Finalmente, Flores Yépez, (2013) se enfoca en la resolución numérica del modelo monodominio de conducción eléctrica en el tejido cardíaco. Allí afirman:

*Este modelo comprende los aspectos referentes a la electrofisiología del corazón y permite describir la dinámica de la propagación mediante la resolución numérica usando los métodos de multimalla no lineales aplicados a la discretización del modelo.*

Este estudio bibliográfico permite hacer una construcción ideal de lo que se trabajó en este trabajo de investigación, donde es necesario tener una fundamentación teórica estructu-

rada, para lograr desarrollarla haciendo uso del código SpecMesh2D basado en el lenguaje Fortran siendo ejecutable en el software cygwin64. Se hace alusión a estos documentos para comprender todos los temas en los cuales se puede profundizar y hacer una ejemplificación adecuada y acertada de estos estudios.

## 8. Marco conceptual

### 8.1. Métodos espectrales

Los métodos espectrales se pueden utilizar para resolver numéricamente ecuaciones diferenciales parciales, brindando soluciones aproximadas en la forma de polinomios sobre todo su dominio.

Este método se utiliza para resolver varios tipos de ecuaciones diferenciales parciales, un ejemplo tiene la siguiente forma:

$$\begin{aligned} \frac{\partial u}{\partial t}(x, t) &= \gamma(x, t)u(x, t), \quad x \in D, t \geq 0, \\ B u(x, t) &= 0, x \in \partial D, \quad t > 0, \\ u(x, 0) &= g(x), \quad x \in D, \end{aligned}$$

Donde  $D$  es el dominio espacial con frontera,  $\gamma(x, t)$  es un operador diferencial (no necesariamente lineal),  $B$  es el operador de frontera y  $g(x)$  es la condición inicial.

Los métodos espectrales son una forma de aproximar las soluciones de las ecuaciones diferenciales. Las componentes claves para su formulación son las funciones base (llamadas también funciones de aproximación o expansión) y las funciones de prueba. Las funciones base se usan para dar una representación aproximada de la solución. Las funciones de prueba se usan para asegurar que la ecuación diferencial y quizás algunas condiciones de frontera se cumplan tanto como sea posible por la serie truncada de expansión.

Para encontrar una solución con métodos espectrales, se utiliza una serie truncada de funciones ortogonales suaves (funciones base)  $\Phi_n(x)$ , de acuerdo con la expresión:

$$u_N(x, t) = \sum_{n=0}^N \hat{u}_n(t) \Phi_n(x).$$

Donde los coeficientes  $\hat{u}_n(t)$ . Aparte representan diversas formas de obtener estos coeficientes, y cada manera de encontrarlos define diferentes esquemas de métodos espectrales. Los más conocidos son: el método de Galerkin, el método de Tau y el método de Colocación, Sáenz López, (2016).

## 8.2. Generación de mallas

La generación de mallas se ha convertido en una aplicación de los métodos espectrales y métodos finitos, este último entendido como la aproximación de problemas continuos, de tal manera, que se especifique un número finito de parámetros asociados y su solución cumpla con los parámetros de los problemas discretos, convirtiéndose y así éste en un requisito para que la malla no quede mal estructura desde el tamaño y la forma para no producir errores en su solución.

Por otra parte, los mallados permiten desarrollar modelos que se usan para la experimentación, siendo éstos herramientas científicas de aproximación a la determinación de estructuras óptimas de diseño en sistemas de monitoreo, automatización, control y diagnóstico de fallas. Usados para la caracterización y creación de conocimientos sobre el comportamiento de fenómenos de nuestro entorno.

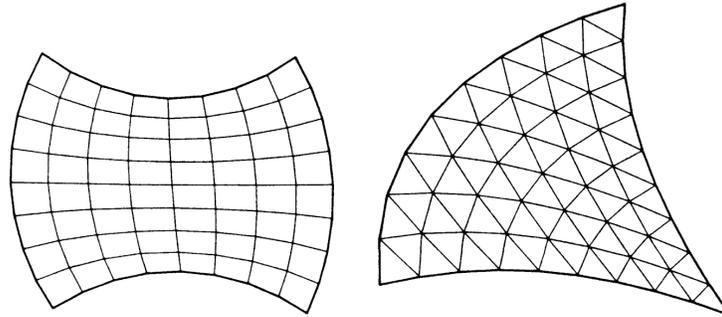
Entre los usos de los mallados se encuentran las mallas para simular el tejido del miocardio y la simulación de arritmias, estableciendo riesgos de los pacientes para decidir tratamientos adecuados y a tiempo. Varios métodos computacionales intentan ofrecer una visión más amplia de las imágenes para generar una geometría tridimensional con regiones únicas de cada paciente, González, (2021).

### 8.3. Mallas estructuradas

Por malla estructurada se entiende toda discretización cuya conectividad sigue un patrón reticular. Por esta razón, en este tipo de mallas se utilizan principalmente elementos cuadriláteros (2D), o hexaédricos (3D), aunque a partir de ellos, se pueden generar triángulos y tetraedros, respectivamente. La creación de esta retícula impone unas fuertes condiciones sobre el contorno del dominio, por lo que en muchos casos este tipo de discretización no es realizable o, siéndolo, presenta una baja calidad, (Morcillo, 2000).

### 8.4. Mallas estructuradas y métodos algebraicos

Dado que en la generación de mallas estructuradas, la conectividad es conocida de antemano, los métodos tienen como objetivo primordial el cálculo de la posición de los nodos que forman la retícula, de modo que los elementos resultantes presentan unas determinadas características de tamaño y regularidad. Las técnicas más sencillas y, por tanto, más rápidas en el cálculo de estas coordenadas nodales son las denominadas algebraicas, entre las que destacan, fundamentalmente, las de transformación en geometrías canónicas o “*mapping*” y, las de interpolación transfinita. Estas técnicas generalmente se aplican sobre geometrías cuadriláteras, aunque en ocasiones también han sido aplicadas en dominios triangulares, donde se generan respectivamente elementos triangulares, tetraédricos o pentaédricos. En la **Figura 1.1** se presentan algunos ejemplos de este tipo de discretización, Morcillo, (2000)



**Figura 1.1:** Mallas generadas mediante métodos algebraicos, Morcillo (2000).

### 8.5. Mallas no estructuradas

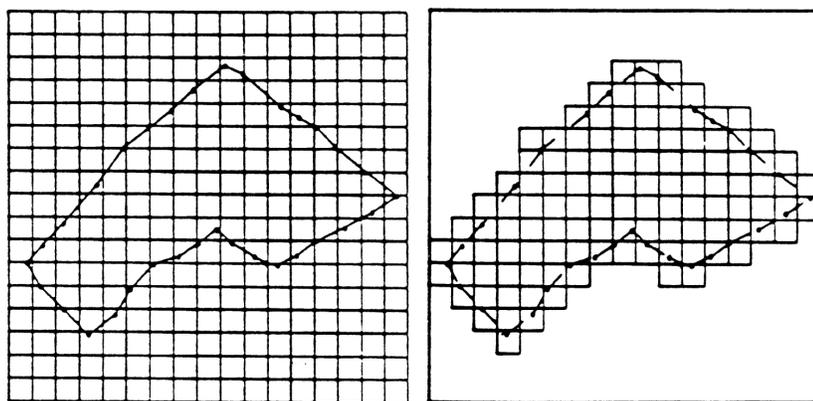
Las mallas no estructuradas, al contrario que las estructuradas, no presentan un patrón de conectividad predeterminado, pues ésta viene determinada por el contorno del problema, la situación de los nodos interiores y el método de discretización utilizado. Una de las características principales de este tipo de discretización es el elemento triangular (2D), aunque en los últimos años se han desarrollado técnicas de generación de malla no estructurada cuadrilátera. Esta libertad en la conectividad hace posible la discretización de cualquier dominio a partir de una discretización del contorno arbitrario. Por lo tanto, este tipo de mallas son de aplicación más general que las estructuradas, pero esto repercute en un mayor coste computacional, Bern and Plassmann (2000).

#### 8.5.1. Métodos de Superposición-Deformación de Retícula (Scheneiders)

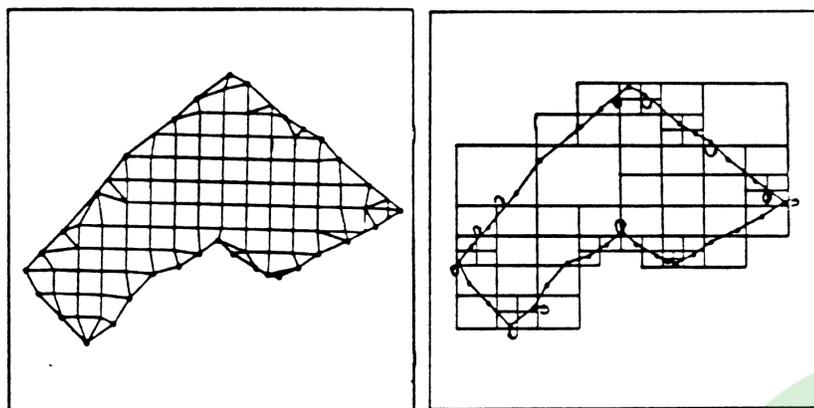
Estos métodos generan la malla a partir de una retícula cuadrada, como la mostrada en la **Figura 1.2.** que contiene al dominio del problema, mediante las siguientes etapas:

1. Eliminación de los cuadrados (2D) que no intersecten el dominio.
2. Procesamiento de los cuadrados o cubos que son intersectados parcialmente por el contorno del dominio. Si este proceso no se da, el ajuste al contorno es muy deficiente.

te, por lo que resulta más apropiado para métodos de diferencias finitas que para el MEF. Generalmente, para mejorarlo se modifican los cuadrados de forma que se ajusten perfectamente al contorno **Figura 1.3.**, o bien se utiliza una retícula jerárquica no conforme (en la terminología inglesa “*quadtree*” u “*octree*”), como muestra la **Figura 1.3.**



**Figura 1.2:** La imagen de la izquierda es la retícula inicial, en la imagen de la derecha la intersección del dominio, Morcillo, (2000).



**Figura 1.3:** La imagen de la izquierda es el ajuste al contorno, en la imagen de la derecha es la retícula jerárquica, Morcillo, (2000).

### 8.5.2. Método de Delaunay-Voronoi

La triangulación de Delaunay fue de las primeras técnicas de discretización triangular utilizadas. Cada una de las regiones propuestas por Dirichlet se denominan región de Voronoi, donde, la unión de los puntos por pares de las regiones poligonales generan otra discretización del dominio, la cual lleva por nombre triangulación de Delaunay, donde la generación de mallas presenta una máxima regularidad de ángulos en los triángulos generados. El método de Delaunay garantiza una triangulación óptima. Sin embargo, esta triangulación óptima no garantiza que los tetraedros generados sean óptimos, es decir, no genera resultados precisos para la solución de la ecuación diferencial parcial, por lo que, en general, tras la generación de la malla son necesarias técnicas de detección y corrección de las figuras defectuosas, Bern and Plassmann, (2000).

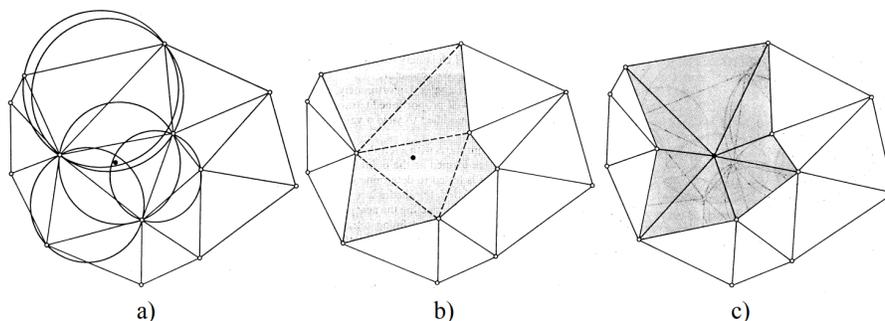
La generación de este tipo de discretización consiste en un método sistemático dividido en las siguientes etapas:

1. En caso de que el dominio del problema no sea convexo, la forma más sencilla es mediante la especificación de 4 puntos que formen un cuadrado en el caso bidimensional u 8 puntos que formen un cubo en el caso volumétrico.
2. Introducción de un nuevo nodo del dominio.
3. Determinación de los elementos cuyas circunferencias o esferas circunscritas contienen al nuevo nodo. Estos elementos deben ser eliminados, así como sus aristas (y caras) comunes.
4. Determinación de los nodos pertenecientes a esos elementos.
5. Determinación de las aristas (o caras) exteriores de los elementos eliminados, es decir, del hueco que resulta tras la eliminación de los elementos anteriores.
6. Generación de nuevos elementos mediante la unión de las aristas (o caras) detectadas

en el punto 5 con el nuevo nodo.

7. Repetición de la secuencia desde el punto 2 hasta que todos los nodos hayan sido introducidos.

8. Eliminación de las aristas y elementos (2D) o aristas, caras y elementos (3D) que no pertenezcan al dominio.



**Figura 1.4.:** Triangulación de Delaunay: a) inserción de nodo, b) detección y eliminación de elementos, c) triangulación local, Morcillo, (2000)

### 8.5.3. Método de Frente de Avance

En este tipo de métodos la generación de la malla se realiza desde el contorno hacia el interior del dominio, siendo los elementos utilizados triángulos en dos dimensiones y tetraedros en problemas tridimensionales. El proceso de creación de la malla es iterativo: se analiza un frente, inicializado con los datos del contorno, para determinar una zona de partida desde la que se crean uno o varios elementos internos. Luego, se actualiza el frente con los nuevos nodos y aristas generadas y repite el proceso hasta que el dominio quede completamente mallado.

George and Frey, (2013) plantean un ejemplo de criterio geométrico en dos dimensiones,

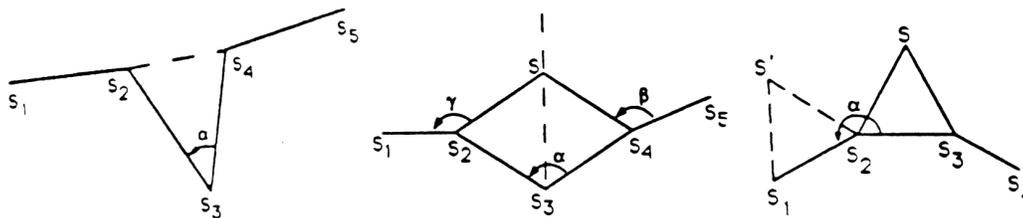
según el cual, si  $\alpha$  es el ángulo formado por dos arista consecutivas del frente, se pueden identificar tres posibles situaciones:

Si  $\alpha < \frac{\pi}{2}$ : Se crea una arista y un elemento formado por ésta y las aristas que definen el ángulo  $\alpha$ .

Si  $\frac{\pi}{2} < \alpha < \frac{2\pi}{3}$ : a partir de las dos aristas con ángulo  $\alpha$  se genera un nodo interno, tres aristas y dos elementos.

Si  $\alpha < \frac{2\pi}{3}$ : se crea un punto interno, las dos aristas que forma éste con los nodos extremos de la arista del frente, y un elemento formado por estas tres aristas.

La posición del nodo interno generado en los dos últimos casos viene determinada por criterios de regularidad y tamaño de los elementos.



**Figura 1.5.:** Ejemplo de tratamiento del frente en mallas bidimensionales, Morcillo (2000).

## 8.6. Interpolación por splines cúbicos

Una función polinómica de n-ésimo grado se define como:

$$y = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n$$

donde  $n$  es un entero no negativo y  $a_k$  son constantes. Siendo  $a_n \neq 0$ , si  $n = 1$  la ecuación representa una recta, si  $n = 2$  la función es cuadrática; si  $n = 3$  se tiene un polinomio cúbico.

Dado un conjunto de puntos, podemos ajustar una curva que pase por ellos. Se construye una sección curva (con un polinomio cúbico) entre cada par de puntos dado. Así, cada sección de curva se define como:

$$x_i(t) = a_0 + a_1t + a_2t^2 + a_3t^3$$

$$y_i(t) = b_0 + b_1t + b_2t^2 + b_3t^3$$

Donde  $t$  varía entre 0 y 1. Los valores de  $t$  se determinan según las condiciones de frontera en las secciones curvas.

- Dos secciones de curva adyacentes deben tener igual coordenada en los extremos
- Es necesario adaptar la pendiente de cada par de curvas adyacentes para obtener una curva global (formada por unión de varias secciones de curva) continua y suave, (Burden, 1996).

### 8.7. Condiciones de continuidad paramétrica

Para asegurar una transición suave en la curva en un intervalo, se imponen las siguientes condiciones de continuidad en los puntos de conexión:

- Continuidad de orden cero  $C^0$ ; únicamente implica que las curvas se unen.
- Continuidad de primer orden  $C^1$ ; las primeras derivadas (tangentes a la curva) son iguales en su punto de unión.
- Continuidad de segundo orden  $C^2$ ; además de lo dicho para  $C^1$ , se tiene que cumplir que la variación de los vectores tangentes al punto de unión de ambas curvas por la derecha y por la izquierda es equivalente. Esto se logra forzando a que la segunda derivada sea igual en la frontera de ambas curvas. Así, se consigue una transición

suave de una sección de curva a la siguiente.

$i$	0	1	2	3	4
$t_i$	0	0.25	0.5	0.75	1
$x_i$	-1	0	1	0	1
$y_i$	0	1	0.5	0	-1

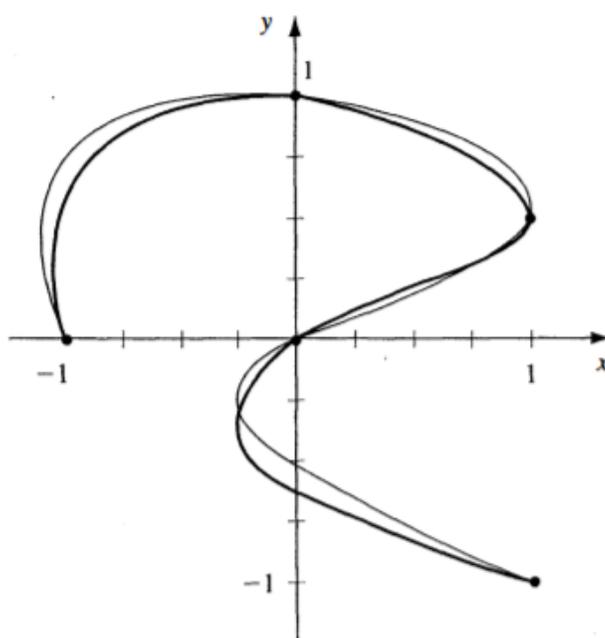


Figura 1.6.: Ejemplo de interpolación por splines cúbicos, (Burden, 1996).

### 8.8. Cuadratura Gaussiana

La cuadratura Gaussiana es una aproximación de una integral definida de una función. Una cuadratura de Gauss  $n$ , es una cuadratura construida para obtener el resultado exacto

al integrar polinomios de grado  $2n - 1$  o menos. Para esto se seleccionan los puntos de evaluación  $x_i$  y los coeficientes  $c_i$  de forma conveniente. La expresión suele expresarse para una integral con límite inferior (-1) y superior de (1) de la siguiente manera:

$$\int_{-1}^1 f(x) dx \approx \sum_{i=1}^n c_i f(x_i)$$

Se tiene que

$$c_i = \int_{-1}^1 L_i(x) dx, \quad i = 1, \dots, n$$

Además, se hace referencia a la interpolación de Lagrange, donde  $L_i(x)$  es el  $n$ -ésimo polinomio interpolante de Lagrange, encontrando aproximaciones que se determinan con sólo especificar determinados puntos en el plano por donde deben pasar, Burden (1996).

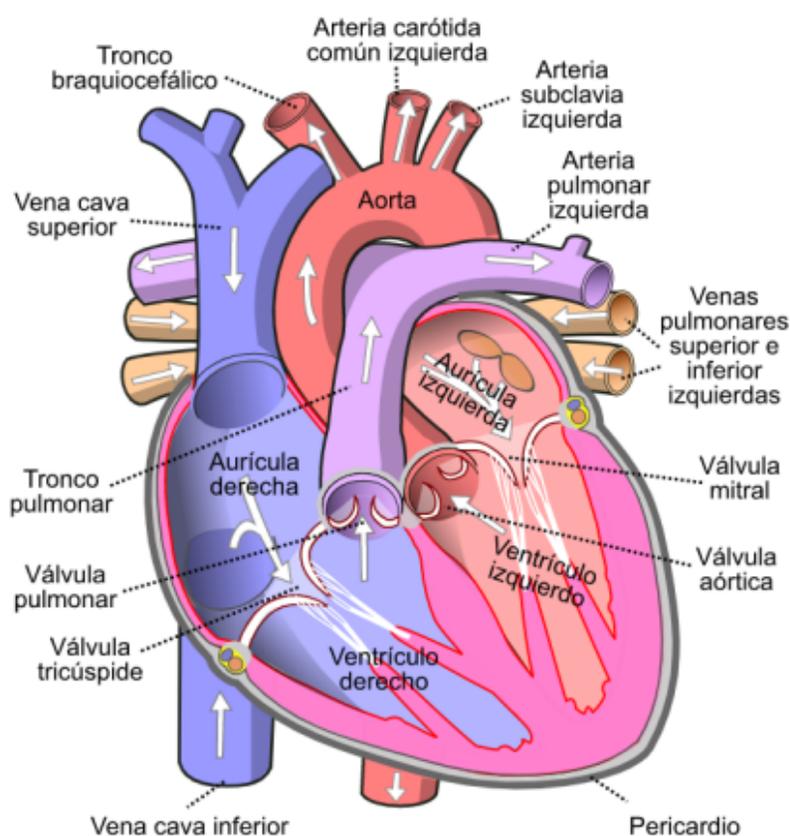
$$L_i(x) = \frac{(x - x_1) \dots (x - x_{i-1}) \dots (x - x_{i+1}) \dots (x - x_n)}{(x_i - x_1) \dots (x_i - x_{i-1}) \dots (x_i - x_{i+1}) \dots (x_i - x_n)}$$

### 8.9. Anatomía del corazón

El corazón es un músculo hueco que circunscribe cavidades en las cuales circula la sangre, cuando se relaja (diástole), atrae hacia sí la sangre que circula en las venas y cuando se contrae (sístole) expulsa la sangre hacia las arterias: aorta o tronco pulmonar, el cual está formado por un músculo con propiedades particulares, llamado miocardio, tapizado interiormente por el endocardio y exteriormente por el epicardio; el corazón está rodeado por el pericardio que es un conjunto fibroso cuya función es separarlo de los órganos vecinos.

El corazón tiene una cara anterior, una posterior y dos bordes: derecho e izquierdo; en la superficie cardíaca se encuentra la grasa por la que avanzan las arterias y las venas que irrigan el corazón, es decir las arterias coronarias, que llevan sangre al músculo cardíaco

y las venas coronarias que la sacan, el peso del corazón varía según la edad, el tamaño y el propio peso de la persona; se considera que el corazón pesa aproximadamente el 0,45 % del peso corporal en el hombre y el 0,40 % del peso corporal en la mujer, de tal modo que en un adulto de estatura media el peso del corazón oscila entre 250 – 350 gramos en los hombres y entre 200 – 300 gramos en las mujeres, hay que tener presente que cuando se trata de deportistas profesionales, habitualmente el corazón muestra un aumento fisiológico o natural de su peso.



**Figura 1.7:** Vista frontal de un corazón humano. Las flechas blancas indican el flujo normal de la sangre. Modificado de Netter, (2003)

La parte interna del corazón está constituida por cuatro cavidades: dos en el lado derecho y dos en el izquierdo, de ahí que sea común hablar de corazón derecho y corazón izquierdo, las cavidades situadas en la parte superior se denominan aurículas y las dispuestas en la parte inferior ventrículos; en condiciones normales, las cavidades derechas no se comunican con las izquierdas, pues se encuentran divididas por un tabique muscular, denominado tabique interauricular que separa ambas aurículas; ahora el tabique que distancia los ventrículos se denomina interventricular.

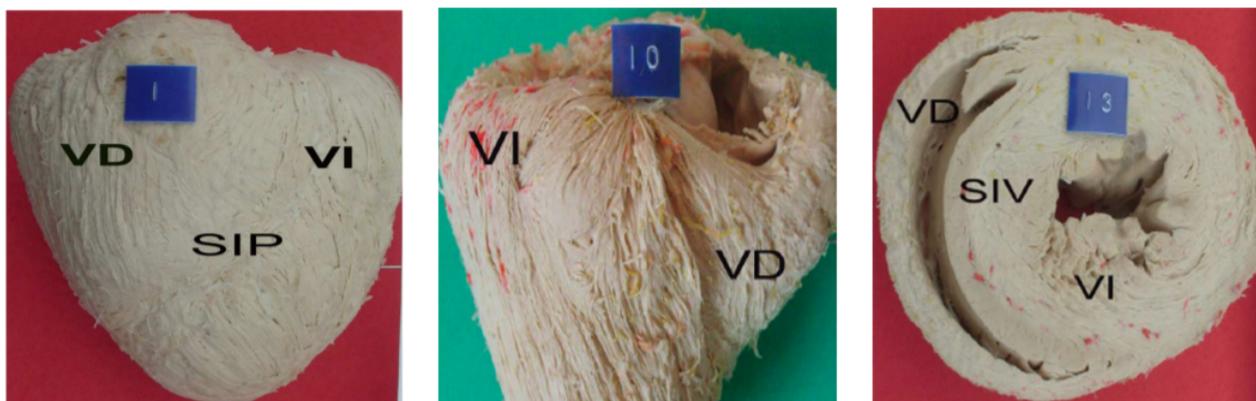
La parte derecha del corazón consta de una aurícula en la parte superior y un ventrículo en la inferior, a la aurícula derecha llega la sangre venosa (no oxigenada) de todo el cuerpo a través de las venas cavas, que desembocan en ellas, ambas se encuentran en la pared posterior próximas al tabique: la superior en la zona más alta, y la inferior en la baja, también desemboca en la aurícula derecha el seno venoso, conducto que recoge la sangre venosa del corazón, en la cara anterior se ubica la orejuela derecha, de forma triangular.

Las aurículas tienen las paredes finas y están constituidas, de fuera hacia dentro, por el pericardio, la hoja interna o miocardio y una capa muy fina o endocardio, esta última reviste toda la superficie interna del corazón, incluidas las válvulas y está formada por una capa de células endoteliales, semejantes a las de los vasos sanguíneos y fibras de colágeno elásticas; la estructura de los ventrículos es semejante, la diferencia estriba en el grosor de la capa muscular mientras que el ventrículo derecho tiene un espesor de 3 a 4 mm, el izquierdo alcanza aproximadamente los 10mm, esta diferencia se debe a que al expulsar la sangre durante la sístole, el ventrículo izquierdo se encuentra con una resistencia mayor, lo que se conoce como: la presión arterial. Las válvulas situadas en los orificios que comunican las aurículas y los ventrículos, llamadas tricúspide y mitral, tienen una morfología diferente de las válvulas que se encuentran entre los ventrículos y las arterias pulmonares y aorta,

es decir, las válvulas pulmonar y aórtica, todas tienen la misma función: se abren y dejan pasar la sangre, para después cerrarse e impedir que la sangre retroceda, Hall, (2011).

### 8.9.1. Fibras Musculares

El músculo cardíaco tiene una estructura de fibras celulares, los iones se mueven con facilidad en el líquido intracelular a lo largo del eje longitudinal de las fibras musculares cardíacas, de modo que los potenciales de acción viajan fácilmente desde una célula muscular cardíaca a la siguiente, por tanto la orientación de estas fibras es importante en la electrofisiología cardíaca. En base a lo anterior se muestra en la **Figura 1.8** la descripción geométrica de las fibras ventriculares y auriculares, además de diferentes cortes en cada sección transversal del corazón de aveztruz, a partir de esta geometría resulta de gran importancia para la elaboración de los dominios para generar las mallas y simulaciones.



**Figura 1.8:** Descripción de las fibras ventriculares en el corazón de aveztruz, Carvallo and Henríquez Pino, (2010).

## 9. Metodología SCRUM

El presente trabajo de grado se realizó bajo la metodología SCRUM Rodríguez and Dorado, (2015). SCRUM es un marco de trabajo iterativo e incremental para el desarrollo de proyectos y se estructura en ciclos de trabajo llamados SPRINTS; estos son iteraciones de 1 a 4 semanas, y se suceden una detrás de otra, la cual permitió interactuar continuamente con el desarrollo. Se representó una solución numérica en el estudio de la electrofisiología del corazón, representando un dominio bidimensional de la **Figura 1.8**, generando un archivo cuya función será visualizar la malla y un archivo donde se encontrarán las características del mallado entre ellas: número de nodos, elementos y una serie de parámetros que se verán reflejados en lo construido por el código, posterior a ello se realizaron simulaciones con los datos obtenidos para la resolución de una ecuación diferencial de onda.

Se realizaron reuniones denominadas "SPRINTS", las cuales fueron beneficiosas para discutir el progreso del trabajo de grado dando una flexibilidad a la hora de cambios o mejores adaptaciones y poder cumplir el objetivo general.

Específicamente, se hizo un estudio del código SpecMesh2D perteneciente a Kopriva, (2011) usado por el grupo de investigación GEDES de la Universidad del Quindío, presentando ejemplos y un manual que sirva como guía para la ejecución y comprensión del código mencionado anteriormente, permitiendo interactuar en el campo de la matemática aplicada, dando respuesta a la pregunta de investigación.

En la siguiente tabla se muestre los sprints utilizados en este trabajo de grado definiendo un tiempo esperado para la realización de cada uno, además de las condiciones necesarias para el cumplimiento.

Tabla de sprints		
Objetivo del sprint	Tiempo de ejecución	Estrategia para el cumplimiento
Estudio del lenguaje de programación Fortran90, Visit, Matlab y terminal cygwin64.	4 meses	Revisión de textos explicativos, exploración de las aplicaciones, diseño de ejemplos simples, uso de las aplicaciones en función del trabajo de grado.
Estudio de textos para la comprensión matemática de la técnica empleada en SpecMesh2D. Además, de la elaboración de los dominios bidimensionales.	3 meses	Revisión del manual SpecMesh2D, donde especifica el método de Schneiders para la generación de mallas y la elaboración de dominios por parametrización o splines cúbicos, teniendo una comprensión de cada método utilizado y hacer uso de ello en el trabajo.
Selección de las secciones del corazón, elaboración de los dominios para su discretización y compilación del mallado en SpecMesh2D.	2 meses	Se hizo una búsqueda de secciones transversales encontrando sin cortes, con un corte y dos cortes, luego se paso a crear el dominio de cada sección mediante interpolación por splines cúbicos y posterior a ello compilar el código en SpecMesh2D.
Uso de los pares ordenados obtenidos en los mallados de cada sección para la resolución de la ecuación diferencial parcial.	1 mes	Se tomaron los resultados obtenidos en cada mallado para poder usarse en la resolución de la ecuación diferencial parcial; luego al obtener las simulaciones se validan la correcta ejecución de los mallados.

## 10. Resultados

Durante el desarrollo se logró la generación de mallados que se adapten a secciones transversales del corazón, específicamente de la **Figura 1.8**; para la elaboración de los mallados se usó el código SpecMesh2D en el cual se elaboraron los dominios para cada sección mediante splines cúbicos, lo cual beneficia a una mayor precisión en las imágenes propuestas; a continuación se explica lo necesario para usar el código y la forma en que se elaboraron los archivos para la generación de los mallados.

### 10.1. Generación de mallas

En esta sección se mostrará el proceso para generar los mallados propuestos, desde la instalación de los programas necesarios, la ejecución y el uso de cada uno, además de como se crea cada archivo que va a compilar y así generar los mallados, visualización de la malla y sus características generales.

#### 10.1.1. SpecMesh2D

SpecMesh2D es un código escrito en Fortran90 diseñado para generar mallas cuadriláteras en geometrías bidimensionales, por lo cual se hace necesario de un compilador de Fortran90, además de un editor de código. Las instrucciones para la compilación de SpecMesh2D se encuentra en, (Kopriva 2011, p. 19).

Cuando se compila un archivo en SpecMesh2D nos genera tres archivos diferentes, con extensiones de archivo (.tec, .txt, .mesh), la primera se usa en Visit para la visualización de las mallas, el segundo presenta las características que se pueden visualizar en un bloc de notas y el tercero es un archivo para visualizar en Matlab obteniendo los zeros (raíces) que corresponden a los polinomios de Legendre.

### 10.1.2. Cygwin

La generación de mallas en este trabajo de grado se hizo por medio del código SpecMesh2D hecho en Fortran90 el cual es autoría de David A. Kopriva, para su instalación se necesita un compilador de fortran en esta caso se trabajó en el sistema operativo Windows 10, se hizo necesario la instalación de Cygwin, que es una gran colección de herramientas GNU y de código abierto que brindan una funcionalidad similar a una distribución de Linux en Windows, funcionando como terminal. Luego de su instalación se instalaron los paquetes de “gfortranz “make-devel” lo cual nos brinda un compilador de fortran y un comando especial para la compilación del código con el paquete devel.

### 10.1.3. Code::blocks

Code::Blocks es un IDE (Entorno de Desarrollo Integrado) gratuito de C/C++ y Fortran creado para satisfacer las necesidades más exigentes de sus usuarios. Está diseñado para ser muy extensible y totalmente configurable.

Al usar Cygwin se necesita un editor de código, en el cual se guardaran la extensión de archivo como (.control).

### 10.1.4. VisIt

VisIt es una herramienta de análisis gráfico y visualización paralela interactiva gratuita para ver datos científicos en plataformas Unix y PC. Los usuarios pueden generar rápidamente visualizaciones a partir de sus datos, animarlas a lo largo del tiempo, manipularlas y guardar las imágenes resultantes para presentaciones. VisIt contiene un amplio conjunto de funciones de visualización para que pueda ver sus datos de varias maneras. Se puede utilizar para visualizar campos escalares y vectoriales definidos en mallas estructuradas y no estructuradas de dos y tres dimensiones (2D y 3D).

VisIt será la herramienta gráfica donde se pueden observar la generación automática de mallas, la extensión de archivo que reconoce VisIt es (.tec).

#### 10.1.5. MATLAB

MATLAB es una plataforma de programación y cálculo numérico, en la cual se aprecian muchos usos en el entorno matemático, mediante esta aplicación podremos usar la extensión de archivo (.mesh), necesaria para ver las soluciones que genera el mallado y poder dar ejemplos en la ecuación monodominio.

### 10.2. Modelo y dominio

Una vez instalados todos los programas requeridos, se pasa a la elaboración de mallas creando un dominio bidimensional en el editor de código. Se empezará con los parámetros de ejecución, el cual define el nombre de los archivos generados tras compilar la extensión (.control), además del formato de archivo de la malla, en este caso se usa ISM (*Implementing Spectral Methods*), luego se plantea el orden polinomial y por último el tamaño de la cuadrícula de fondo que hace referencias al Método de Schneiders, ver Anexo 1.

Luego, se inicia con el modelo, el dominio va a estar conformado por una curva exterior (*Outerboundary*) y curvas interiores (*Innerboundary*), las curvas interiores no son necesarias en la generación de la malla solo basta con una curva exterior. Para elaborar las curvas interiores o exteriores hay tres formas diferentes de hacerlo, al realizar cualquiera de las tres curvas hay que definir cual se utiliza.

- **Ecuaciones paramétricas:** El primer tipo de curva es un sistema de ecuaciones paramétricas y permite representar una curva o superficie en el plano o en el espacio, mediante valores que recorren un intervalo de números reales, mediante una variable.

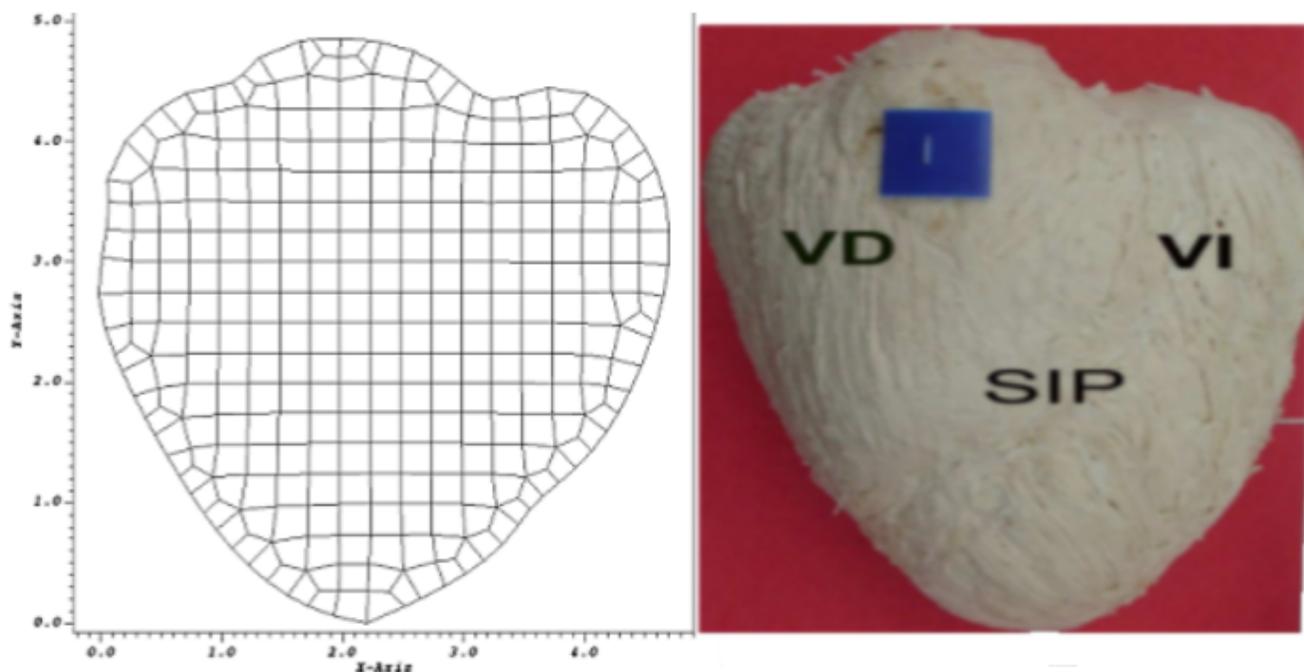
Un ejemplo de una curva definida por ecuaciones paramétricas se encuentra en el Anexo 2.

- **Splines cúbicos:** El segundo tipo de curva es el tipo Spline Cúbico, que ajusta una spline cúbico a un conjunto de nodos en valores de parámetros dados. La parametrización no tiene por qué ser uniforme. Un ejemplo de una curva definida por un spline cúbico se encuentra en el Anexo 3.
- **Rectas por nodos:** El último tipo de curva es el tipo de curva de puntos finales que toma un par de parejas ordenadas inicial y un par final, luego forma una línea recta entre ellos. Un ejemplo de una recta por nodos se encuentra en el Anexo 4.

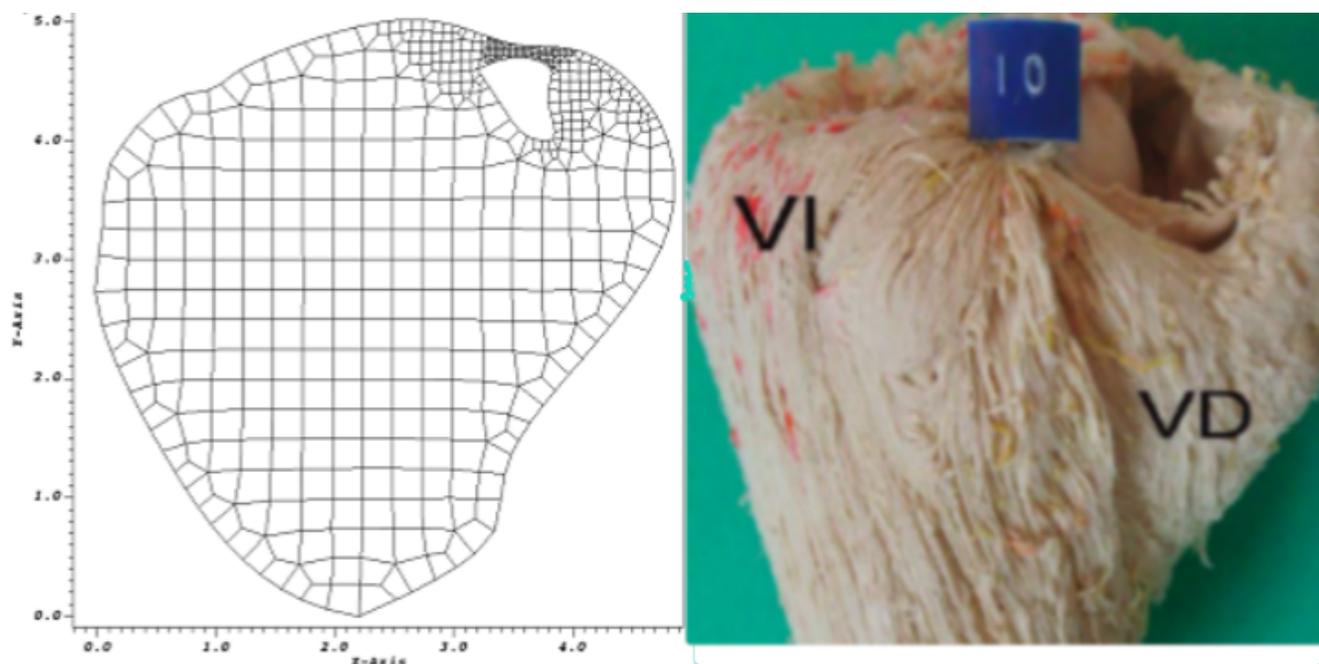
Al definir los parámetros de la malla y establecer el dominio bidimensional, se pasa a compilar el archivo en cygwin con el siguiente comando (`./SpecMesh2D < Examples/corazon.control`), primero se ejecuta el código SpecMesh2D, luego se llamó a la carpeta “Examples” en la cual se encontrarán los archivos de control que se ejecutarán, en este ejemplo se compilará el archivo de control llamado “corazón”, después se mostrarán las características de la malla generada en el terminal de cygwin, ver Anexo 5.

Cuando se compila el archivo se obtiene la visualización de la malla, además de sus características. Se puede visualizar en Visit, Matlab y un bloc de notas, ver Anexo 6.

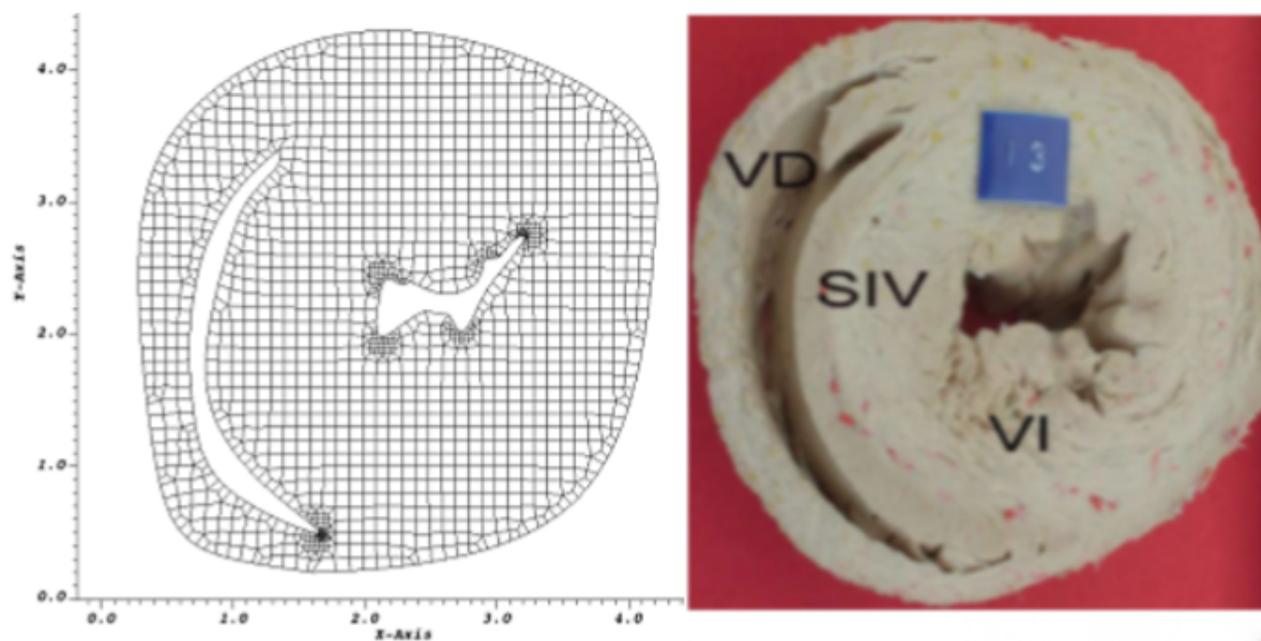
Al generar los mallados se obtienen las características como el número de elementos, los nodos internos de la malla que están representados en términos de los ceros (raíces) de los polinomios de Legendre, las mallas son necesarias para el soporte de la solución numérica para la resolución de la ecuación de onda; en Fortran se obtuvo una simulación que describe el comportamiento eléctrico en diferentes secciones del corazón teniendo en cuenta la expansión de la onda en las ventrículas en diferentes momentos; a continuación se presentarán los ejemplos propuestos, clasificándose por cortes.



**Figura 2:** Mallado cuadrangular no estructurado de la primer sección del corazón de la **Figura 1.8** donde no se evidencian cortes en el interior, teniendo solo el dominio exterior, el cual se realizó por interpolación de *splines* cúbicos. El código de ejecución se puede observar en el Anexo 7. La malla se generó con un orden polinomial de grado 6 sobre los nodos de Legendre, obteniendo un total de 325 nodos y 289 elementos.



**Figura 2.1:** Mallado cuadrangular no estructurado de la segunda sección del corazón de la **Figura 1.8** evidenciando un corte arriba del ventrículo derecho, elaborando el dominio exterior y interior el cual representa el corte en la sección, los dos dominios se realizaron por interpolación de *splines* cúbicos y el código de ejecución se puede observar en el Anexo 8. La malla se generó con un orden polinomial de grado 6 sobre los nodos de Legendre, obteniendo un total de 650 nodos y 566 elementos.



**Figura 2.2:** Mallado cuadrangular no estructurado de la tercera sección del corazón de la **Figura 1.8** evidenciando un corte arriba del ventrículo izquierdo y abajo del ventrículo derecho, elaborando el dominio por interpolación de splines cúbicos teniendo un dominio exterior y dos curvas interiores que representan los cortes mencionados, el código de ejecución se puede observar en el Anexo 9. La malla se generó con un orden polinomial de grado 6 sobre los nodos de Legendre, obteniendo un total de 1801 nodos y 1630 elementos.

### 10.3. Resolución de la ecuación diferencial

Para la simulación no se especificó en la resolución de la ecuación diferencial parcial ya que el objetivo de este trabajo de grado no era la solución de la ecuación como tal; las mallas son la parte más importante de la solución numérica, así que con los resultados se esperaba la correcta simulación de las tres secciones.

Al momento de generar los mallados se da verificación que los dominios que se elaboraron no presenten discontinuidad en los cuadriláteros que se generaron en el dominio. Una vez obtenidos los mallados, se verifica en el código de Fortran el cual representa la resolución de la ecuación de onda con los nodos que generó el código en cada mallado; para la resolución se usó el método de Galerkin discontinuo nodal para propagar y reflejar ondas planas, donde se propagó una única onda gaussiana a través de la malla. Se define con el siguiente vector de onda:

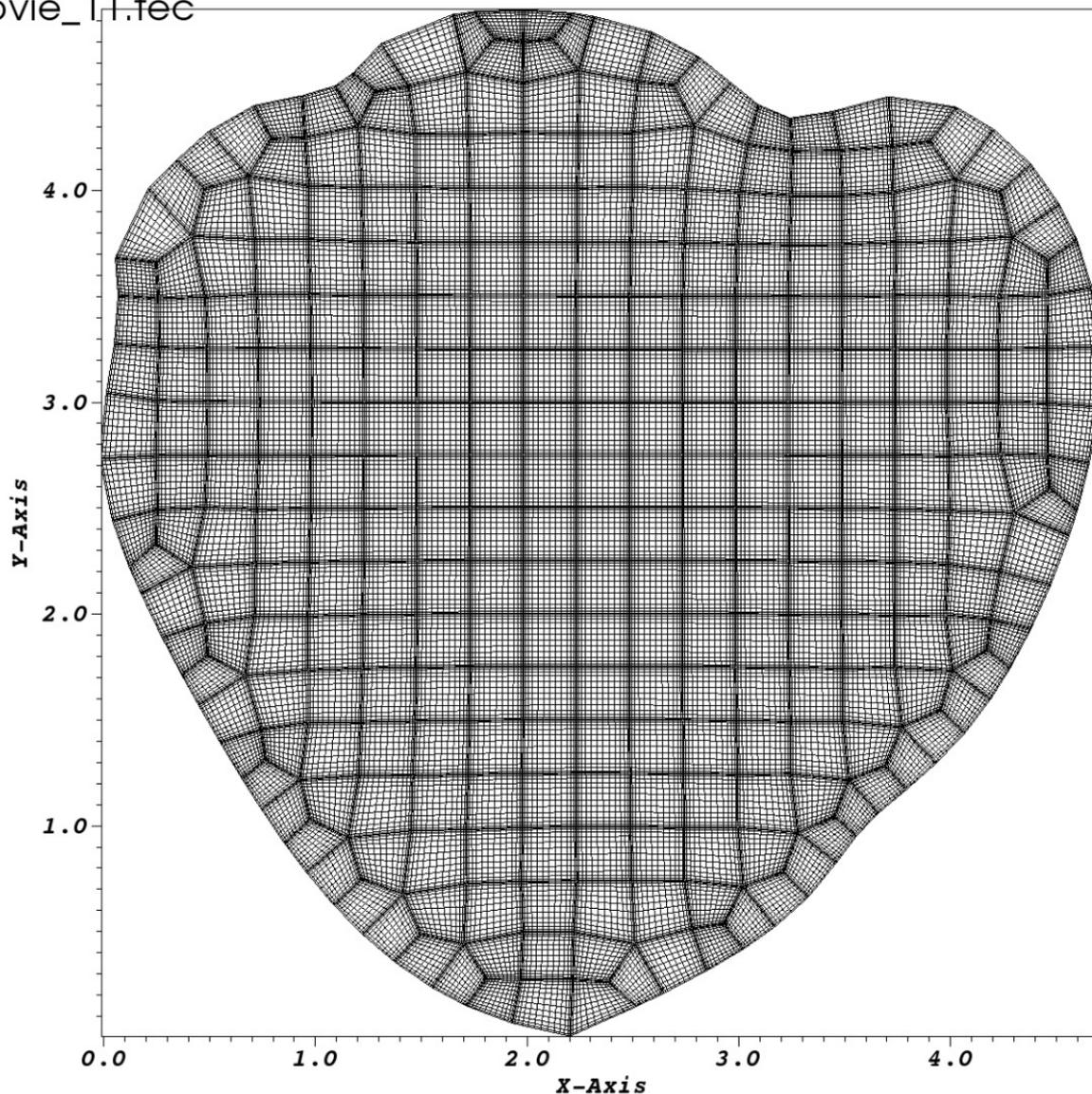
$$\begin{bmatrix} p \\ u \\ v \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{k_x}{c} \\ \frac{k_y}{c} \end{bmatrix} e^{-\frac{(k_x(x-x_0)+k_y(y-y_0)-ct)^2}{d^2}}$$

Con el vector de onda  $k$  normalizado para satisfacer  $k_x^2 + k_y^2 = 1$ . Se trata de una onda con forma gaussiana.

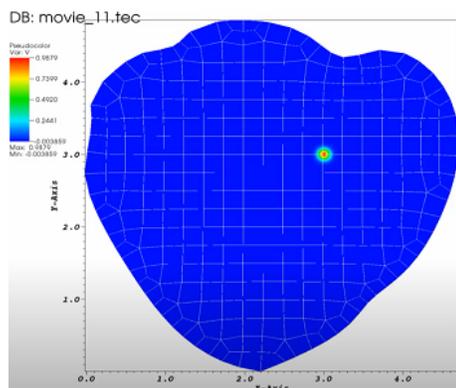
A continuación se presenta un mallado que surge de la discretización mediante el método de Galerkin, luego los resultados que se obtuvieron en las simulaciones las cuales se muestran en diferentes unidades de tiempo, presentando cada segundo transcurrido.

DB: movie\_11.tec

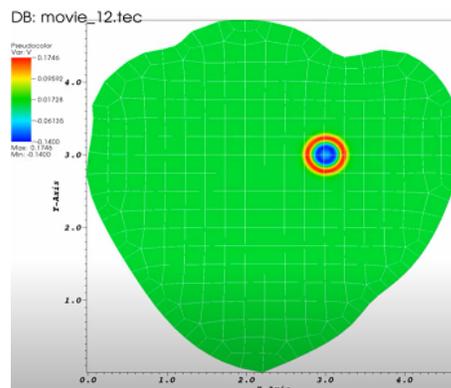
Mesh  
Var: mesh



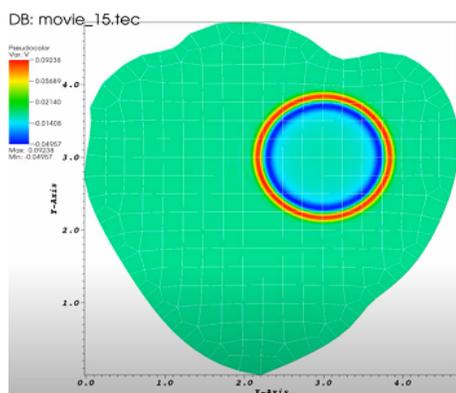
**Figura 2.3:** Corresponde a la malla que surge de aplicar la discretización mediante el método de Galerkin, ya que cada elemento está discretizado con nodos que corresponden a los ceros de un polinomio de Legendre de grado 6.



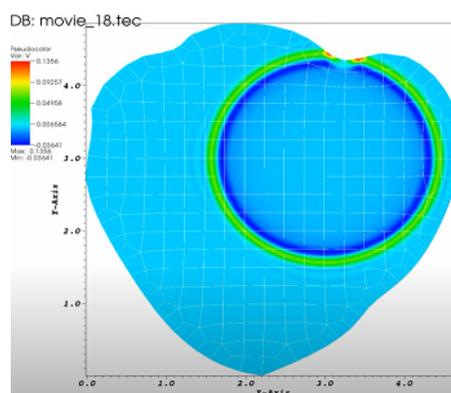
a)  $t = 0s$



b)  $t = 0.2s$

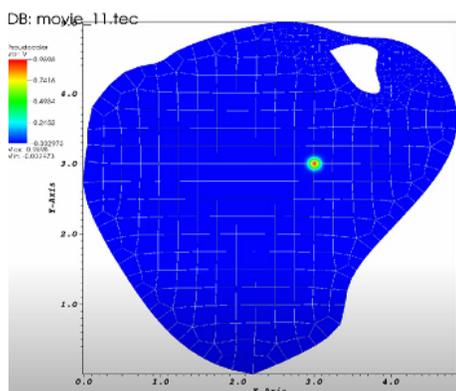


c)  $t = 0.8s$

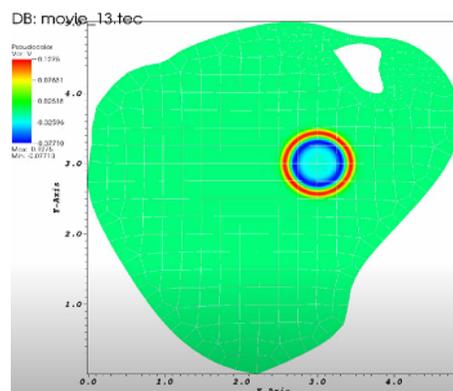


d)  $t = 1.4s$

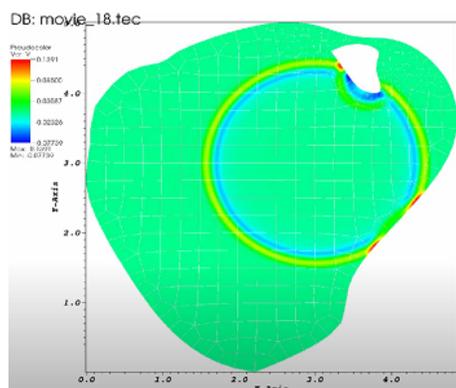
**Figura 2.4:** Simulación de la primer sección del corazón donde no presenta cortes en su interior, se evidencia una expansión de la onda desde el ventrículo izquierdo, desde un tiempo de cero segundos hasta que la onda colisiona en la frontera.



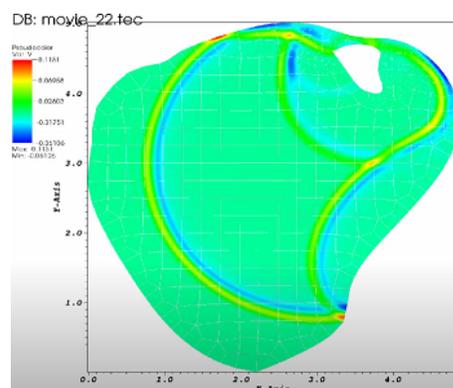
a)  $t = 0s$



b)  $t = 0.4s$

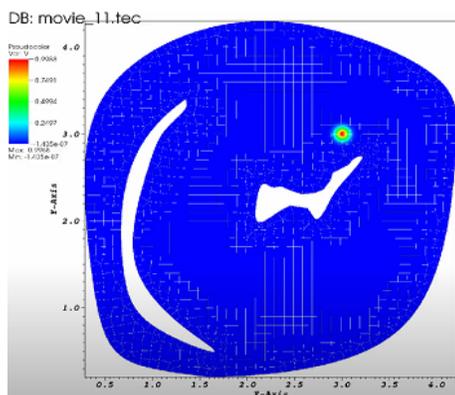


c)  $t = 1.4s$

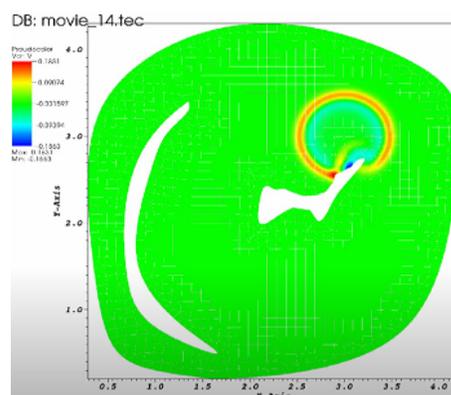


d)  $t = 2.2s$

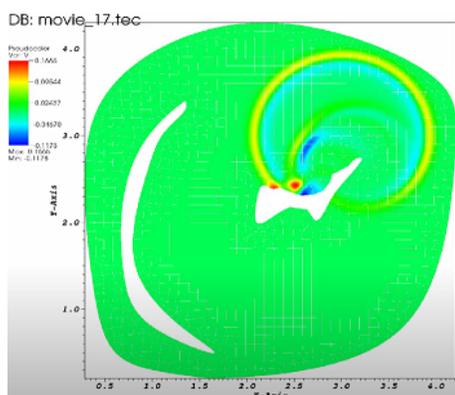
**Figura 2.5:** Simulación de la segunda sección del corazón evidenciando un corte arriba del ventrículo derecho, se evidencia una expansión de la onda un poco más arriba del ventrículo derecho, desde un tiempo de cero segundos hasta que la onda colisiona con el borde exterior.



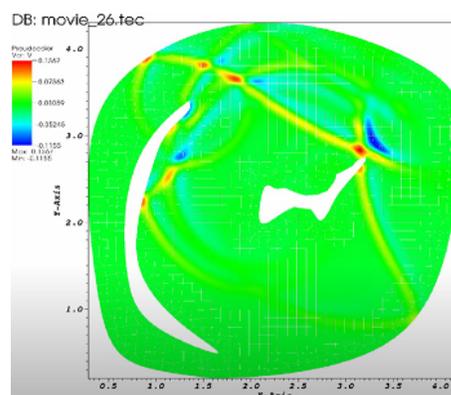
a)  $t = 0s$



b)  $t = 0.45s$



c)  $t = 0.9s$



d)  $t = 2.25s$

**Figura 2.6:** Simulación de la tercera sección del corazón evidenciando un corte arriba del ventrículo izquierdo y abajo del ventrículo derecho, se evidencia una expansión de la onda un poco más arriba del corte del ventrículo izquierdo, desde un tiempo de cero segundos hasta que la onda colisiona con el borde exterior.

Todas las simulaciones se hicieron con condiciones de frontera reflectivas, donde se observa el acoplamiento de la solución de la ecuación de onda.

## 11. Conclusiones

1. En la figura 2.4 se observa que en los apartados a), b), c) y d) se encuentra un punto, llamado punto de control donde se propaga la onda de un corazón sano, la cual se propaga de forma progresiva por todo el tejido cardíaco, mostrando que el método de Schneiders aplicado en SpecMesh2D es apropiado para este tipo de simulaciones.
2. En la figura 2.5 y 2.6 en donde se presenta uno y dos cortes respectivamente, se observa en el apartado d) de ambas figuras que el paso progresivo de la onda se interrumpe, ya que después de un tiempo la onda rebota en las condiciones de frontera y ya no tiene una forma adecuada, en comparación a la onda de la figura 2.4.
3. El método de Schneiders utilizado en SpecMesh2D es apropiado en la elaboración de mallas con cuadriláteros en un dominio bidimensional, exigiendo poco coste computacional en la generación de mallados, sin embargo, su coste computacional aumenta en la realización de simulaciones, ya que a medida en que aumenta la cantidad de elementos de cada mallado, el tiempo en compilar cada simulación puede llegar a tardar alrededor de 2 días.
4. Los ejemplos propuestos en las secciones transversales del corazón fueron adaptadas satisfactoriamente en SpecMesh2D, obteniendo una aproximación numérica en el dominio planteado, generando las características del mallado y el archivo que corresponde a los zeros de los polinomios de Legendre, para la resolución de la ecuación del problema monodominio.

## 12. Referencias bibliográficas

### Referencias

- [1] Marshall W Bern and Paul E Plassmann. Mesh generation. *Handbook of computational geometry*, 38, 2000.
- [2] Richard L Burden, J Douglas Faires, Rafael Iriarte Balderrama, et al. *Análisis numérico*. 1996.
- [3] Pamela Carvallo and Jorge Henríquez-Pino. Arquitectura de las fibras ventriculares en el corazón de avestruz (*struthio camelus*). *International Journal of Morphology*, 28(2):353–356, 2010.
- [4] Elkinn Adrián Calderón César Augusto Acosta Minoli. Generador automático de mallas cuadrangulares para la solución numérica de problemas de propagación de ondas. 2015.
- [5] Leydi Michelle Cherrez Monar and Joselyn Alexandra Miranda Masaquiza. Simulación de los modelos matemáticos bidominio y desfibrilación para estudiar el restablecimiento del ritmo cardiaco, utilizando el software libre scilab. B.S. thesis, Escuela Superior Politécnica de Chimborazo, 2020.
- [6] César Augusto Acosta Minoli Elkinn Adrián Calderón. Estudio e implementación de un generador automático de mallas no estructuradas en 2d mediante el método de schneiders para simular numéricamente modelos en ecuaciones diferenciales parciales. 2014.
- [7] Williams Fabricio Flores Yépez. Simulación numérica del modelo de excitación cardíaca de aliev-panfilov con métodos multimalla. B.S. thesis, Quito, 2013., 2013.

- [8] Paul Louis George and Pascal Frey. *Mesh generation: application to finite elements*. John Wiley & Sons, 2013.
- [9] Tomás González González. Estudio mediante simulación de la generación de arritmias reentrantes en distintos estadios tras un infarto de miocardio. 2021.
- [10] John E Hall. *Guyton y Hall. Tratado de fisiología médica*. Elsevier Health Sciences, 2011.
- [11] Ignacio Jonas, Luis J Romero, Mariana del Fresno, Juan P D’Amato, and Marcelo J Véner. Generación de mallas de elementos finitos para arterias coronarias a partir de imágenes ivus. *Mecánica Computacional*, 28(15):1237–1246, 2009.
- [12] David A Kopriva. *Implementing spectral methods for partial differential equations: Algorithms for scientists and engineers*. Springer Science & Business Media, 2009.
- [13] David A. Kopriva. Specmesh2d user manual. *Department of Mathematics, The Florida State University*, 2011.
- [14] DA Lavender and DR Hayhurst. An assessment of higher-order isoparametric elements for solving an elastic problem. *Computer methods in applied mechanics and engineering*, 56(2):139–165, 1986.
- [15] Xinghua Liang, Mohamed S Ebeida, and Yongjie Zhang. Guaranteed-quality all-quadrilateral mesh generation with feature preservation. *Computer Methods in Applied Mechanics and Engineering*, 199(29-32):2072–2083, 2010.
- [16] Cristina Sonia Melo Lagos. Desarrollo de una herramienta que genera mallas de superficie compuestas de cuadriláteros para modelar el crecimiento de árboles. 2008.
- [17] Alejandro Díaz Morcillo. *Métodos de mallado y algoritmos adaptativos en dos y tres dimensiones para la resolución de problemas electromagnéticos cerrados mediante el método de los elementos finitos*. PhD thesis, Universitat Politècnica de València, 2000.

- [18] Frank Henry Netter, Stephanie Engelhardt, and Martin Kortenhaus. *Medicina interna*. Elsevier España, 2003.
- [19] D Francisco Vicente Paya. *Un método de diferencias finitas generalizadas para simular la actividad eléctrica de un tejido cardiaco*. PhD thesis, Universidad Politécnica de Madrid, 2013.
- [20] Edgar Quispe Ccapacca. Algoritmos para generación de mallas cuadrilaterales por descomposición jerárquica de dominio. 2018.
- [21] César Rodríguez and Rubén Dorado. ¿ por qué implementar scrum? *Revista Ontare*, 3(1):125–144, 2015.
- [22] David Sáenz López. Estudio de los métodos espectrales en ecuaciones diferenciales de una dimensión y su comparación con el método de diferencias finitas. 2016.
- [23] Robert Schneiders and Rolf Bünten. Automatic generation of hexahedral finite element meshes. *Computer Aided Geometric Design*, 12(7):693–707, 1995.
- [24] Michael Ian Shamos. Computational geometry. *Ph. D. thesis, Yale University*, 1978.

## 13. Anexos

### Anexo 1: Parámetros de ejecución y formato de la malla.

```
\begin{ControlInput}

  \begin{RunParameters}
    mesh file name = "MeshFiles/corazon.mesh"
    plot file name = "PlotFiles/corazon.tec"
    stats file name = "Statistics/corazon.txt"
    mesh file format = "ISM"
    polynomial order = 6
  \end{RunParameters}

  \begin{MeshParameters}
    mesh type = "conforming"
    background grid size = [0.25,0.25]
  \end{MeshParameters}
\end{ControlInput}
```

### Anexo 2: Ejemplo de ecuaciones paramétricas.

```
\begin{Model}
\begin{OuterBoundary}
  \begin{ParametricEquationCurve}
    name = "B1"
    xEqn = "f(t) = t"
    yEqn = "f(t) = 0"
  \end{ParametricEquationCurve}
  \begin{ParametricEquationCurve}
    name = "B2"
    xEqn = "f(t) = 1"
    yEqn = "f(t) = t"
  \end{ParametricEquationCurve}
  \begin{ParametricEquationCurve}
    name = "B3"
    xEqn = "f(t) = 1 - t"
    yEqn = "f(t) = 1"
  \end{ParametricEquationCurve}
  \begin{ParametricEquationCurve}
    name = "B4"
    xEqn = "f(t) = 0"
    yEqn = "f(t) = 1-t"
  \end{ParametricEquationCurve}
\end{OuterBoundary}
\end{Model}
```

```
\begin{InnerBoundaries}

  \begin{Chain}
    name = "CircleCenter"
    \begin{ParametricEquationCurve}
      name = "CircleCenter"
      xEqn = "f(t) = 0.5 + 0.25*cos(2*pi*t)"
      yEqn = "f(t) = 0.5 + 0.25*sin(2*pi*t)"
    \end{ParametricEquationCurve}
  \end{Chain}

\end{InnerBoundaries}
```

**Anexo 3: Ejemplo de *spline* cúbico.**

```
\begin{SplineCurve}
name = "SplineBoundary"
9
0.0000000000000000 -3.5000000000000000 3.5000000000000000
3.846153846153846E-002 -3.2000000000000000 5.0000000000000000
7.692307692307693E-002 -2.0000000000000000 6.0000000000000000
0.769230769230769 0.0000000000000000 -1.0000000000000000
0.807692307692308 -1.0000000000000000 -1.0000000000000000|
0.846153846153846 -2.0000000000000000 -0.8000000000000000
0.884615384615385 -2.5000000000000000 0.0000000000000000
0.923076923076923 -3.0000000000000000 1.0000000000000000
1.000000000000000 -3.5000000000000000 3.5000000000000000
\end{SplineCurve}
```

**Anexo 4: Ejemplo de rectas por nodos.**

```
\begin{EndPointsCurve}
name = "B1"
xStart = [0.0,1.0]
xEnd = [2.0,1.0]
\end{EndPointsCurve}
```

Anexo 5: Ejemplo al generar una malla.

```

ESTIVEN@DESKTOP-Q10S5F4 ~/Mesh
$ ./SpecMesh2D < Examples/corazon.control
Mesh Statistics:
Total time           = 0.11000000000000000
Number of nodes      = 558
Number of Edges      = 1044
Number of Elements   = 486

Mesh Quality:
Measure              Minimum           Maximum           Average           Acceptable Low
Acceptable High Reference
Signed Area          0.00032796       0.00883195       0.00165483       0.00000000
999.99900000        1.00000000
Aspect Ratio         1.00167161       3.32196759       1.22732807       1.00000000
999.99900000        1.00000000
Condition            1.00000965       4.18069692       1.18768052       1.00000000
4.00000000          1.00000000
Edge Ratio           1.00294831       7.65830081       1.54751724       1.00000000
4.00000000          1.00000000
Jacobian             0.00020912       0.00734897       0.00139242       0.00000000
999.99900000        1.00000000
Minimum Angle        30.81743053      89.88538126      76.26204135      40.00000000
90.00000000         90.00000000
Maximum Angle        90.12657079      159.45200215     103.44271049     90.00000000
135.00000000        90.00000000
Area Sign            1.00000000       1.00000000       1.00000000       1.00000000
1.00000000          1.00000000

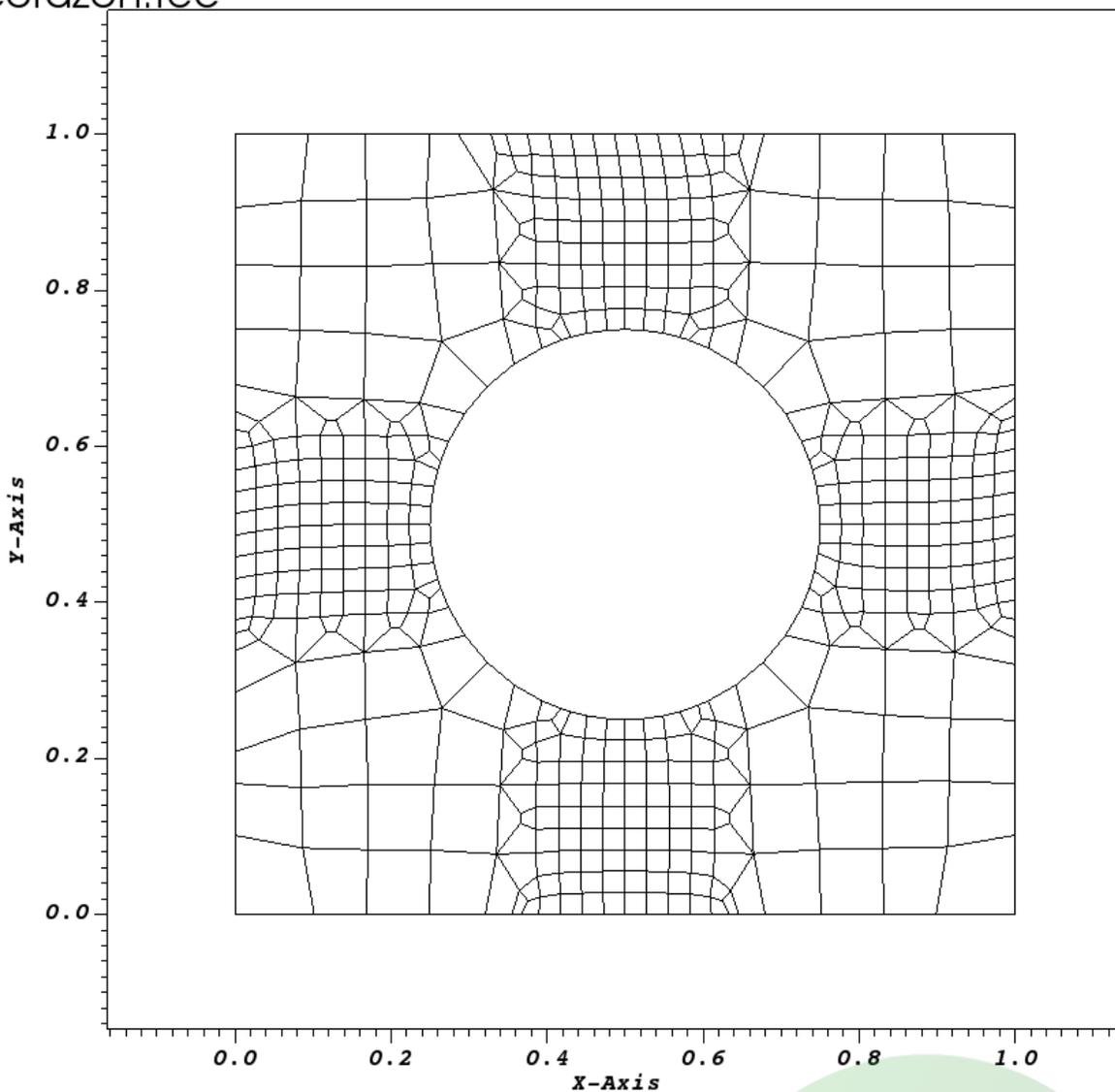
Testing for duplicate nodes...
No duplicate nodes found. Mesh OK
Test complete.

```

Anexo 6: Visualización de la malla en Visit, estadísticas en bloc de notas y Matlab.

DB: corazon.tec

Mesh  
Var: mesh

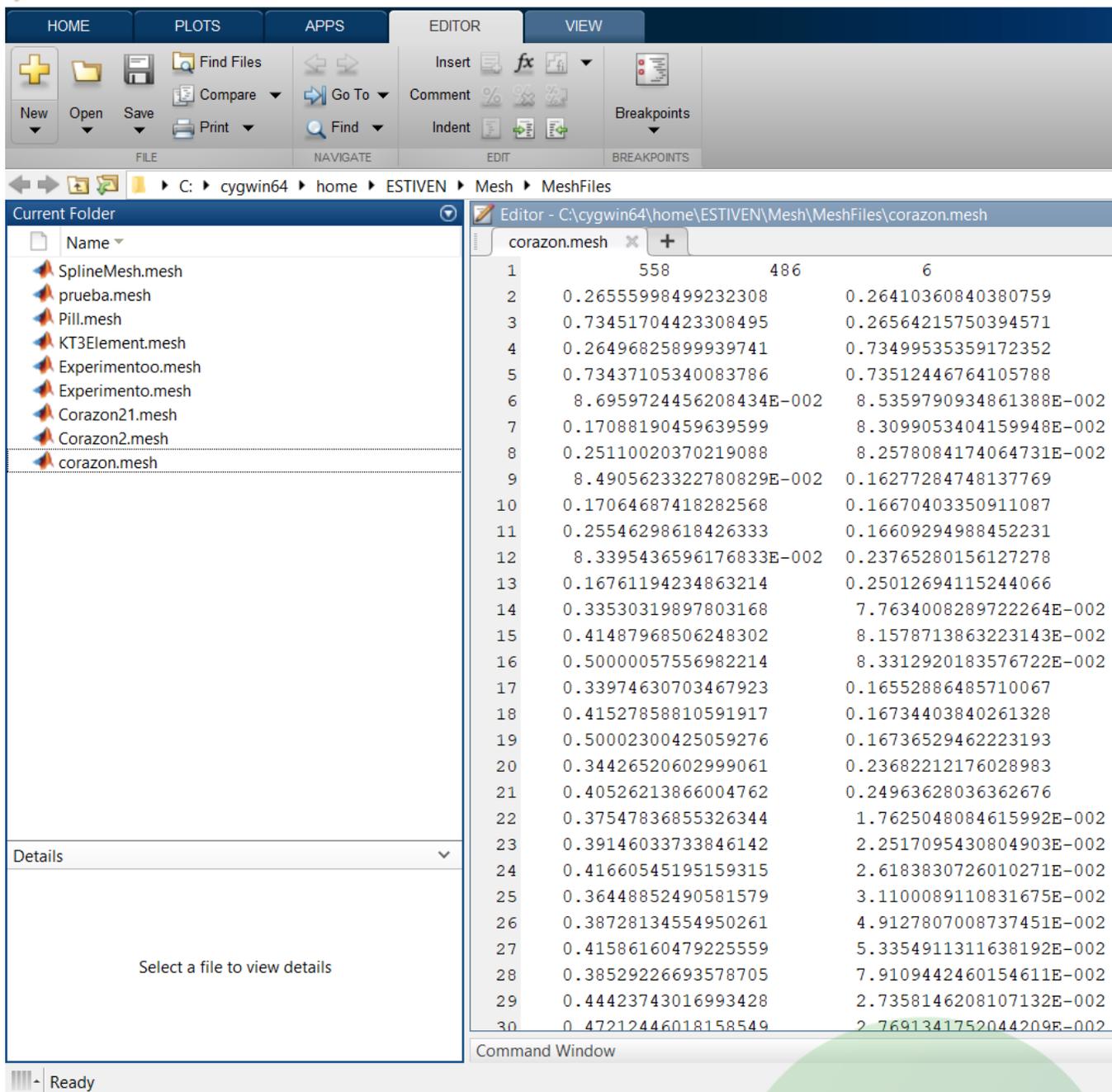


corazon: Bloc de notas

Archivo Edición Formato Ver Ayuda

Signed Area	Aspect Ratio	Condition	Edge Ratio	Jacobian	Minimum Angle	Maximum Angle	Area Sign
6.8307E-03	1.0393E+00	1.0055E+00	1.1084E+00	6.4920E-03	8.7536E+01	9.3063E+01	1.0000E+00
6.8963E-03	1.0217E+00	1.0019E+00	1.0573E+00	6.7017E-03	8.7422E+01	9.2618E+01	1.0000E+00
6.7448E-03	1.0478E+00	1.0164E+00	1.1460E+00	6.3250E-03	8.3658E+01	9.7270E+01	1.0000E+00
8.2663E-03	1.0945E+00	1.0306E+00	1.1852E+00	7.0738E-03	7.5997E+01	9.6037E+01	1.0000E+00
7.2322E-03	1.0352E+00	1.0019E+00	1.0524E+00	7.0414E-03	8.7490E+01	9.2607E+01	1.0000E+00
7.0004E-03	1.1877E+00	1.0500E+00	1.3793E+00	5.7344E-03	7.6764E+01	1.0549E+02	1.0000E+00
4.7550E-04	1.3766E+00	1.1513E+00	1.7387E+00	2.6912E-04	6.0583E+01	1.1218E+02	1.0000E+00
7.3208E-04	1.0697E+00	1.0092E+00	1.1370E+00	6.8445E-04	8.3155E+01	9.6728E+01	1.0000E+00
1.5442E-03	1.4588E+00	1.2168E+00	1.8902E+00	7.1922E-04	5.9595E+01	1.2455E+02	1.0000E+00
8.5139E-04	1.0311E+00	1.0066E+00	1.0640E+00	8.1079E-04	8.5382E+01	9.6421E+01	1.0000E+00
7.6322E-04	1.0181E+00	1.0012E+00	1.0345E+00	7.5166E-04	8.7398E+01	9.1806E+01	1.0000E+00
7.7411E-04	1.0097E+00	1.0002E+00	1.0184E+00	7.6926E-04	8.9223E+01	9.0473E+01	1.0000E+00
7.7495E-04	1.0051E+00	1.0000E+00	1.0088E+00	7.7307E-04	8.9807E+01	9.0127E+01	1.0000E+00
7.9765E-04	1.0193E+00	1.0011E+00	1.0331E+00	7.8254E-04	8.8192E+01	9.2612E+01	1.0000E+00
7.8469E-04	1.0105E+00	1.0002E+00	1.0182E+00	7.8084E-04	8.9313E+01	9.0812E+01	1.0000E+00
7.7892E-04	1.0056E+00	1.0001E+00	1.0097E+00	7.7764E-04	8.9817E+01	9.0194E+01	1.0000E+00
1.2529E-03	1.5054E+00	1.4655E+00	2.4069E+00	5.8200E-04	5.0482E+01	1.1198E+02	1.0000E+00
8.6784E-04	1.0262E+00	1.0033E+00	1.0554E+00	8.4170E-04	8.5437E+01	9.4180E+01	1.0000E+00
1.4223E-03	3.0156E+00	3.5345E+00	7.0942E+00	3.3142E-04	3.4934E+01	1.4326E+02	1.0000E+00
4.0349E-04	1.5008E+00	1.2756E+00	2.3312E+00	2.4050E-04	6.4757E+01	1.1466E+02	1.0000E+00
8.3801E-04	1.0166E+00	1.0015E+00	1.0307E+00	8.2651E-04	8.7015E+01	9.2271E+01	1.0000E+00
1.1310E-03	1.4560E+00	1.3024E+00	2.1540E+00	5.5377E-04	5.5088E+01	1.1512E+02	1.0000E+00
8.2187E-04	1.0110E+00	1.0015E+00	1.0258E+00	8.1534E-04	8.7270E+01	9.2888E+01	1.0000E+00
8.1723E-04	1.0119E+00	1.0010E+00	1.0310E+00	8.0680E-04	8.8165E+01	9.2147E+01	1.0000E+00
7.9321E-04	1.0089E+00	1.0002E+00	1.0187E+00	7.8939E-04	8.9415E+01	9.0809E+01	1.0000E+00
7.8277E-04	1.0041E+00	1.0000E+00	1.0083E+00	7.8196E-04	8.9872E+01	9.0187E+01	1.0000E+00
8.1920E-04	1.0106E+00	1.0007E+00	1.0251E+00	8.1339E-04	8.8303E+01	9.1538E+01	1.0000E+00
7.9862E-04	1.0054E+00	1.0002E+00	1.0130E+00	7.9573E-04	8.9214E+01	9.0703E+01	1.0000E+00
7.8635E-04	1.0018E+00	1.0000E+00	1.0032E+00	7.8473E-04	8.9713E+01	9.0198E+01	1.0000E+00
8.2248E-04	1.0056E+00	1.0004E+00	1.0112E+00	8.2024E-04	8.8370E+01	9.1508E+01	1.0000E+00
8.0399E-04	1.0058E+00	1.0002E+00	1.0105E+00	7.9910E-04	8.8928E+01	9.0774E+01	1.0000E+00
7.8784E-04	1.0089E+00	1.0002E+00	1.0203E+00	7.8225E-04	8.9402E+01	9.0326E+01	1.0000E+00
1.0526E-03	1.5398E+00	1.5165E+00	2.4104E+00	5.3902E-04	4.9120E+01	1.1582E+02	1.0000E+00
8.3673E-04	1.0102E+00	1.0030E+00	1.0219E+00	8.2477E-04	8.5563E+01	9.3947E+01	1.0000E+00
1.0731E-03	2.7028E+00	3.1927E+00	6.1502E+00	2.7693E-04	3.6105E+01	1.4023E+02	1.0000E+00
3.4930E-04	1.4524E+00	1.3304E+00	2.2474E+00	2.1120E-04	6.2948E+01	1.1837E+02	1.0000E+00
9.1263E-04	1.1659E+00	1.0695E+00	1.3297E+00	7.6174E-04	6.9965E+01	1.0661E+02	1.0000E+00
9.0991E-04	1.4996E+00	1.1795E+00	2.1010E+00	4.8539E-04	6.1598E+01	1.1834E+02	1.0000E+00
6.0039E-04	1.3638E+00	1.1409E+00	1.7092E+00	3.7255E-04	7.2101E+01	1.1767E+02	1.0000E+00
8.4484E-04	1.0121E+00	1.0014E+00	1.0219E+00	8.3849E-04	8.7129E+01	9.2471E+01	1.0000E+00
8.1297E-04	1.0314E+00	1.0013E+00	1.0513E+00	7.9661E-04	8.8237E+01	9.1284E+01	1.0000E+00
7.7679E-04	1.0222E+00	1.0014E+00	1.0537E+00	7.6319E-04	8.8764E+01	9.0736E+01	1.0000E+00

MATLAB R2017b



HOME PLOTS APPS EDITOR VIEW

FILE NAVIGATE EDIT BREAKPOINTS

C:\cygwin64\home\ESTIVEN\Mesh\MeshFiles

Current Folder

- SplineMesh.mesh
- prueba.mesh
- Pill.mesh
- KT3Element.mesh
- Experimentoo.mesh
- Experimento.mesh
- Corazon21.mesh
- Corazon2.mesh
- corazon.mesh

Editor - C:\cygwin64\home\ESTIVEN\Mesh\MeshFiles\corazon.mesh

Line	Value 1	Value 2	Value 3
1	558	486	6
2	0.26555998499232308		0.26410360840380759
3	0.73451704423308495		0.26564215750394571
4	0.26496825899939741		0.73499535359172352
5	0.73437105340083786		0.73512446764105788
6	8.6959724456208434E-002		8.5359790934861388E-002
7	0.17088190459639599		8.3099053404159948E-002
8	0.25110020370219088		8.2578084174064731E-002
9	8.4905623322780829E-002		0.16277284748137769
10	0.17064687418282568		0.16670403350911087
11	0.25546298618426333		0.16609294988452231
12	8.3395436596176833E-002		0.23765280156127278
13	0.16761194234863214		0.25012694115244066
14	0.33530319897803168		7.7634008289722264E-002
15	0.41487968506248302		8.1578713863223143E-002
16	0.50000057556982214		8.3312920183576722E-002
17	0.33974630703467923		0.16552886485710067
18	0.41527858810591917		0.16734403840261328
19	0.50002300425059276		0.16736529462223193
20	0.34426520602999061		0.23682212176028983
21	0.40526213866004762		0.24963628036362676
22	0.37547836855326344		1.7625048084615992E-002
23	0.39146033733846142		2.2517095430804903E-002
24	0.41660545195159315		2.6183830726010271E-002
25	0.36448852490581579		3.1100089110831675E-002
26	0.38728134554950261		4.9127807008737451E-002
27	0.41586160479225559		5.3354911311638192E-002
28	0.38529226693578705		7.9109442460154611E-002
29	0.44423743016993428		2.7358146208107132E-002
30	0.47212446018158549		2.7691341752044209E-002

Command Window

Ready

Anexo 7: Código para ejecución del mallado de la Figura 2 sin cortes, dominio mediante interpolación por splines cúbicos.

```

\begin{RunParameters}
  mesh file name = "MeshFiles/Experiment.mesh"
  plot file name = "PlotFiles/Experiment.tec"
  stats file name = "Statistics/Experiment.txt"
  mesh file format = "ISM"
  polynomial order = 6
\end{RunParameters}

\begin{MeshParameters}
  mesh type = "conforming"
  background grid size = [0.5,0.5]
\end{MeshParameters}

\begin{Smoother}
  smoothing = "ON"
  smoothing type = "LinearAndCrossBarSpring"
  spring constant = 1.0
  mass = 1.0
  rest length = 0.0
  damping coefficient = 5.0
  number of iterations = 30
  time step = 0.1
\end{Smoother}

\end{ControlInput}

\begin{Model}
\begin{OuterBoundary}
\begin{SplineCurve}
name = "SplineBoundaryCurve"
16
0.000000000000000 4.000000000000000 0.500000000000000|
0.021450663216333 2.000000000000000 1.800000000000000
0.098736027838293 0.300000000000000 5.000000000000000
0.132600000000000 0.900000000000000 6.100000000000000
0.198800083627838 2.000000000000000 6.000000000000000
0.264000000012333 2.900000000000000 6.700000000000000
0.317824843244332 4.200000000000000 6.500000000000000
0.380000003076923 5.000000000000000 5.800000000000000
0.458637823980333 5.800000000000000 6.000000000000000
0.530153846153846 7.100000000000000 5.800000000000000
0.609230769230769 7.600000000000000 5.000000000000000
0.749230769244444 7.100000000000000 3.600000000000000
0.827692307692308 7.200000000000000 3.200000000000000
0.890153846153846 6.800000000000000 2.500000000000000
0.966423784678585 5.600000000000000 1.200000000000000
1.000000000000000 4.000000000000000 0.500000000000000
\end{SplineCurve}

```

Anexo 8: Código para ejecución del mallado de la Figura 2.1 con un corte, curvas interiores y exteriores mediante interpolación por splines cúbicos.

```

background grid size = [0.25,0.25]
\end{MeshParameters}

\begin{Smoother}
smoothing = "ON"
smoothing type = "LinearAndCrossBarSpring"
spring constant = 1.0
mass = 1.0
rest length = 0.0
damping coefficient = 5.0
number of iterations = 35
time step = 0.1
\end{Smoother}

\end{ControlInput}

\begin{Model}
\begin{OuterBoundary}
\begin{SplineCurve}
name = "SplineBoundaryCurve"
16
0.000000000000000 2.200000000000000 0.000000000000000
0.002145663216333 1.600000000000000 0.200000000000000
0.007873627838293 0.600000000000000 1.300000000000000
0.026000000000000 0.000000000000000 3.000000000000000
0.081800083627838 0.100000000000000 3.800000000000000
0.104000000012333 0.400000000000000 4.200000000000000
0.137824843244332 0.800000000000000 4.400000000000000
0.240000003076923 1.200000000000000 4.600000000000000
0.355863782398333 2.400000000000000 5.000000000000000
0.530153846153846 3.600000000000000 4.800000000000000
0.609230769230769 4.200000000000000 4.700000000000000
0.749230769244444 4.700000000000000 3.000000000000000
0.827692307692308 3.650000000000000 1.640000000000000
0.890153846153846 3.400000000000000 1.000000000000000
0.966423784678585 3.000000000000000 0.400000000000000
1.000000000000000 2.200000000000000 0.000000000000000
\end{SplineCurve}
\end{OuterBoundary}

\begin{InnerBoundaries}
\begin{SplineCurve}
name = "SplineBoundaryCurve2"
8
0.000000000000000 3.200000000000000 4.600000000000000
0.123145663216333 3.500000000000000 4.700000000000000
0.337873627838293 3.820000000000000 4.600000000000000
0.596000000000000 3.800000000000000 4.300000000000000
0.781800083627838 3.800000000000000 4.000000000000000
0.864000000012333 3.600000000000000 4.050000000000000
0.954000000012333 3.400000000000000 4.300000000000000
1.000000000000000 3.200000000000000 4.600000000000000
\end{SplineCurve}

```

Anexo 9: Código para ejecución del mallado de la Figura 2.2 con dos cortes, curvas interiores y exteriores mediante interpolación por splines cúbicos.

```

\begin{Model}
\begin{OuterBoundary}
\begin{SplineCurve}
name = "SplineBoundaryCurve"
11
0.000000000000000 1.600000000000000 0.200000000000000
0.0334615384615386 1.000000000000000 0.300000000000000
0.1162307692307693 0.400000000000000 1.100000000000000
0.153656737762300 0.300000000000000 2.000000000000000
0.262307692307693 0.600000000000000 3.600000000000000
0.395230769230769 2.200000000000000 4.300000000000000
0.517692307692308 3.800000000000000 3.800000000000000
0.6706153846153846 4.200000000000000 2.800000000000000
0.834615384615385 3.800000000000000 1.000000000000000
0.923076923076923 3.000000000000000 0.400000000000000
1.000000000000000 1.600000000000000 0.200000000000000
\end{SplineCurve}
\end{OuterBoundary}

\begin{InnerBoundaries}
\begin{SplineCurve}
name = "SplineBoundaryCurve2"
13
0.000000000000000 1.350000000000000 3.400000000000000
0.053145663216333 0.900000000000000 2.900000000000000
0.116545134234432 0.700000000000000 2.300000000000000
0.231800083627838 0.700000000000000 1.400000000000000
0.32400000012333 1.000000000000000 0.800000000000000
0.39600000012333 1.600000000000000 0.500000000000000
0.46600000012333 1.550000000000000 0.600000000000000
0.578372178368772 1.100000000000000 1.000000000000000
0.668372178368772 0.800000000000000 1.600000000000000
0.778372178368772 0.950000000000000 2.600000000000000
0.8538372178368772 1.100000000000000 3.000000000000000
0.958372178368772 1.350000000000000 3.300000000000000
1.000000000000000 1.350000000000000 3.400000000000000
\end{SplineCurve}

\begin{SplineCurve}
name = "SplineBoundaryCurve3"
17
0.000000000000000 2.120000000000000 2.350000000000000
0.043145663216333 2.100000000000000 2.000000000000000
0.106545134234432 2.320000000000000 2.100000000000000
0.181800083627838 2.600000000000000 2.180000000000000
0.25400000012333 2.730000000000000 2.020000000000000
0.310000000123334 2.830000000000000 2.150000000000000
0.39600000012333 2.970000000000000 2.400000000000000
0.468372178368772 3.200000000000000 2.700000000000000
0.558372178368772 3.100000000000000 2.690000000000000
0.618372178368772 3.020000000000000 2.590000000000000
0.683837217836877 2.900000000000000 2.550000000000000
0.758372178368772 2.820000000000000 2.370000000000000
0.808372178368772 2.560000000000000 2.310000000000000
0.853837217836877 2.370000000000000 2.390000000000000
0.908372178368772 2.270000000000000 2.380000000000000
0.948372178368772 2.200000000000000 2.420000000000000
1.000000000000000 2.120000000000000 2.350000000000000
\end{SplineCurve}

```