

*Universidad de Holguín “Oscar Lucero Moya”
Facultad de Informática y Matemática*



ESTUDIO DE ALGORITMOS DE REFINAMIENTO 3D PARA MALLAS DE TETRAEDROS

**Tesis en opción al título de Máster en
Matemática Aplicada e Informática para la Administración**

Autor:

Ing. Miguel Ángel Abellón Medina

Holguín, febrero de 2014

Universidad de Holguín “Oscar Lucero Moya”
Facultad de Informática y Matemática



ESTUDIO DE ALGORITMOS DE REFINAMIENTO 3D PARA MALLAS DE TETRAEDROS

**Tesis en opción al título de Máster en
Matemática Aplicada e Informática para la Administración**

Autor:

Ing. Miguel Ángel Abellón Medina

Tutor:

Dr. José Pablo Suárez Rivero

M.Sc. Leydis Lamoth Borrero

Holguín, 2014

No se necesita de grandes hombres para poder realizar grandes cosas; se necesita solo gente consagrada y dedicada.

Philipps Brooks (1835-1893)

AGRADECIMIENTOS

A mi familia por todo el tiempo robado para dedicárselo a este trabajo.

A mi tutora Leydis por sus consejos y ayuda.

A mi otro tutor, José Pablo por brindarme documentación, código Matlab y guiarme en el camino de la generación de mallas a través de sus emails.

A la profesora María-Cecilia Rívara de la Universidad de Chile por aclararme dudas de los algoritmos, con su conocimiento y saber.

A Pedro Rodríguez Moreno de la Universidad del Bio-Bio de Chile, doctorando de Cecilia, por su ayuda en las implementaciones y envío de papers.

A Lorna Figueroa por enviarme su tesis de Magister de la Universidad de Santiago de Chile y otros temas en español.

A Rafael Montenegro y González Yuste de la Universidad de las Palmas de Gran Canaria por sus aclaraciones sobre la tesis doctoral desarrollada.

A Lucana, doctorando de José Pablo por explicarme su código de la conformidad en la malla.

A Nancy Hitschfeld por enviarme un paper sobre estructuras de datos para mallas de tetraedros.

De forma general, a todos.

Muchas Gracias.

DEDICATORIA

A mi familia, a mis amigos y a todos los apasionados del mesh generation.

RESUMEN

Los algoritmos de generación de mallas han sido objeto de numerosas investigaciones en los últimos años. Constituyen herramientas básicas en los métodos numéricos, como el método de elementos finitos o en la aproximación lineal a trozos de funciones, así como en la industria gráfica, como gráficos por ordenador, diseño y modelado de sólidos y superficies, visualización, CAD/CAM, etc. Cada vez se hace más necesario conocer cuál de estos algoritmos refina una malla con mejor calidad, donde es crítico que los algoritmos sean eficientes en términos de tiempo y espacio de cómputo.

Esta tesis investiga una clase de algoritmos de refinamiento basados en el criterio del lado mayor y otros haciendo uso de la triangulación Delaunay, sus comparativas y propiedades geométricas.

El trabajo está estructurado como sigue: en el capítulo primero se hace una introducción general de la generación de mallas, incluyendo ideas, conceptos, algoritmos y estrategias de refinamiento. Asimismo, se presentan algunas ideas básicas de diseño de estructuras de datos. Se concluye el capítulo señalando los objetivos propuestos en esta tesis.

El capítulo segundo se dedica al estudio empírico de las particiones geométricas de elementos simpliciales en 3D, estos basados en el lado mayor y delaunay. Se presentan las propiedades matemáticas y computacionales de los algoritmos propuestos. Se arriba a las conclusiones del estudio realizado y se proponen diversas líneas de trabajo futuro. Con la presente investigación se obtuvieron como resultados 3 trabajos aceptados en eventos internacionales y un artículo publicado.

ABSTRACT

In the last years, mesh generation algorithms have been largely investigated. These algorithms are basic tools in numerical methods, as in the finite element method or in the piece way linear approximation to a given function. The algorithms are also relevant in the graphic industry, as in computer graphics, surfaces and solid modeling and design, visualization, CAD/CAM, etc. Furthermore, in an increasing way, the necessity for knowing which of these algorithms refines a mesh with better quality arises, in this wide topic it is specially critical to provide algorithms with efficient performance in terms of time and space.

The present thesis investigates a class of refinement algorithms which are based on longest-edge scheme and delaunay based, their comparison and geometrical properties.

The outline of the treatment is as follows: chapter one introduces mesh generation, including basic ideas, concepts, algorithms and refinement schemes. Some preliminaries about data structures are given and the goals of the thesis close the chapter.

Second chapter studies empirical geometric properties of 3D simplicial partitions, specially the algorithms based on longest-edge and delaunay. It is presented some mathematical and computational properties related to the proposed algorithms. Some remarkable conclusions derived from the thesis are emphasized, and some possible points of research for the future are pointed out. With the current research were obtained as a result three international mesh generation events and a paper in proceedings based on this thesis.

ÍNDICE DE CONTENIDO

| | |
|---|-----------|
| INTRODUCCIÓN..... | 1 |
| CAPÍTULO 1 FUNDAMENTOS TEÓRICOS..... | 7 |
| 1.1 Introducción | 7 |
| 1.2 Estrategias de Refinamiento | 9 |
| 1.3 Estructura de datos geométricas..... | 12 |
| 1.3.1 Estructuras de datos en la representación de mallas | 13 |
| 1.3.2 Relaciones de adyacencia completas | 17 |
| 1.4 Algoritmos de refinamiento y particiones en 3D..... | 18 |
| 1.5 Algoritmos de generación de mallas basados en Delaunay..... | 23 |
| 1.6 Medidas de la calidad y la degeneración..... | 26 |
| 1.7 Librerías de código abierto para la generación de mallas..... | 28 |
| 1.8 Sistemas de apoyo a la toma de decisiones..... | 32 |
| 1.9 Conclusiones | 34 |
| CAPÍTULO 2 Estudio empírico de los algoritmos propuestos..... | 35 |
| 2.1 Introducción | 35 |
| 2.2 Algoritmo LE | 38 |
| 2.3 Algoritmo 3T-LE..... | 41 |
| 2.4 Algoritmo LE-Trisección | 46 |
| 2.5 Algoritmos basados en Delaunay..... | 54 |
| 2.5.1 Criterios de inserción de puntos..... | 55 |
| 2.5.2 Análisis del costo de tiempo | 58 |
| 2.5.3 Comparativa de la calidad con algoritmos por el lado mayor..... | 60 |
| 2.5.4 Propiedad Auto mejora de la calidad de los algoritmos basados en Delaunay..... | 62 |
| 2.6 Algoritmo 4T-Baricentro..... | 65 |
| 2.7 Conclusiones | 69 |
| CONCLUSIONES GENERALES | 70 |
| RECOMENDACIONES..... | 71 |
| BIBLIOGRAFÍA..... | 72 |
| ANEXOS | I |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1 Comparación de las librerías: NA: No Aplicable. | 32 |
| Tabla 2 Coordenadas de los tetraedros. | 41 |
| Tabla 3 Forma de medida (η), malla τ^{13} con 1, 594, 323 elementos para los 5 diferentes tetraedros t_1^1 usando la partición LE-Trisección. | 53 |
| Tabla 4 Forma de medida (η), malla τ^{10} con 1, 048, 576 elementos para los 5 diferentes tetraedros t_1^1 usando la partición LE-Cuatrisección. | 54 |
| Tabla 5 Resumen de los tiempos de ejecución de la malla de prueba. | 59 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1 Esquema que clasifica el campo de trabajo de las mallas en relación con cuatro campos de investigación. | 2 |
| Figura 2 Particiones por el lado mayor de los triángulos. | 10 |
| Figura 3 Inserción de un nuevo punto en una triangulación Delaunay | 11 |
| Figura 4 Ejemplos de mallas en 2D y 3D | 16 |
| Figura 5 Tipos de adyacencia completas | 17 |
| Figura 6 Partición similar en 3D y vista coplanaria de las caras. | 18 |
| Figura 7 Caso ilustrativo de la partición 8T-LE. | 19 |
| Figura 8 Partición Baricéntrica 3D | 19 |
| Figura 9 Patrones usados por Bey que cubren $(4 + 6 + 12 + 3) = 25$ posibilidades. .. | 20 |
| Figura 10 Tetraedros abiertos clasificados como rojo y negro. | 20 |
| Figura 11 Tetraedros tipo rojo y negro. | 21 |
| Figura 12 Clasificación de los tetraedros según Liu y Joe. | 22 |
| Figura 13 Test del círculo. | 25 |
| Figura 14 Malla inicial para el algoritmo de Watson en 3D | 25 |
| Figura 15 Distintos tipos de forma de los tetraedros | 28 |
| Figura 16 Malla de tetraedros de ejemplo. | 37 |
| Figura 17 Gráfico de la evolución de la calidad para 4 iteraciones de refinamiento. | 38 |
| Figura 18 Bisección por el lado mayor de un tetraedro. | 39 |
| Figura 19 (a) LE uniforme en un tetraedro regular. | 39 |
| Figura 19 (b) LE uniforme en un tetraedro cap. | 39 |
| Figura 19 (c) LE uniforme en un tetraedro needle. | 40 |
| Figura 19 (d) LE uniforme en un tetraedro sliver. | 40 |
| Figura 19 (e) LE uniforme en un tetraedro wedge. | 40 |
| Figura 20 Partición 3T-LE de un triángulo. | 42 |
| Figura 21 Posición relativa de la arista mayor (numerada 1) y arista mayor secundaria (numerada 2) de T. | 44 |
| Figura 22 Partición de los 3-Tetraedros obtenidos de acuerdo a la posición relativa de la arista mayor (numerada 1) y la arista mayor secundaria (numerada 2) de T. | 45 |
| Figura 23 Resultados reportados de los algoritmos 3T-LE, LE y LE-Trisección. | 46 |
| Figura 24 :Trisección por el lado mayor de un tetraedro. | 47 |

| | |
|---|----|
| Figura 25 :Resultados reportados de la LE-Cuatrisección y de la LE-Trisección 3D. | 49 |
| Figura 26 (a) : Nivel 2 de refinamiento uniforme de la LE-Trisección en un tetraedro regular. | 50 |
| Figura 26 (b) : Nivel 2 de refinamiento uniforme de la LE-Cuatrisección en un tetraedro regular. | 50 |
| Figura 27 (a) : LE-Trisección uniforme en un tetraedro regular..... | 51 |
| Figura 27 (b) : LE-Trisección uniforme en un tetraedro cap. | 51 |
| Figura 27 (c) : LE-Trisección uniforme en un tetraedro needle. | 51 |
| Figura 27 (d) : LE-Trisección uniforme en un tetraedro sliver. | 52 |
| Figura 27 (e) : LE-Trisección uniforme en un tetraedro wedge. | 52 |
| Figura 28 : Cinco tetraedros iniciales considerados y evolución de la constante de degeneración para la LE-Trisección..... | 53 |
| Figura 29 : Cinco tetraedros iniciales considerados y evolución de la constante de degeneración para la LE-Cuatrisección. | 54 |
| Figura 30 : Malla inicial, 12 tetraedros y 9 vértices..... | 59 |
| Figura 31 (a) Lado Mayor-Delaunay Uniforme en un tetraedro cap..... | 60 |
| Figura 31 (b) Lado Mayor-Delaunay Uniforme en un tetraedro needle. | 60 |
| Figura 31 (c) Lado Mayor-Delaunay Uniforme en un tetraedro regular. | 61 |
| Figura 31 (d) Lado Mayor-Delaunay Uniforme en un tetraedro sliver. | 61 |
| Figura 31 (e) Lado Mayor-Delaunay Uniforme en un tetraedro wedge. | 61 |
| Figura 32 (a) Centroides-Delaunay Uniforme en un tetraedro needle. | 63 |
| Figura 32 (b) Circunferencia-Delaunay Uniforme en un tetraedro needle. | 63 |
| Figura 32 (c) Inesfera-Delaunay Uniforme en un tetraedro needle. | 63 |
| Figura 32 (d) Cuatrisección Lado Mayor-Delaunay Uniforme en un tetraedro needle. | 64 |
| Figura 32 (e) Lado Mayor-Delaunay Uniforme en un tetraedro needle..... | 64 |
| Figura 32 (f) Trisección Lado Mayor-Delaunay Uniforme en un tetraedro needle. ... | 64 |
| Figura 32 (g) Puntos Medios-Delaunay Uniforme en un tetraedro needle. | 65 |
| Figura 33 Partición 4T-Baricentro de un tetraedro. | 66 |
| Figura 34 (a) 4T-Baricentro uniforme en un tetraedro cap..... | 66 |
| Figura 34 (b) 4T-Baricentro uniforme en un tetraedro needle. | 67 |
| Figura 34 (c) 4T-Baricentro uniforme en un tetraedro regular. | 67 |

| | |
|---|----|
| Figura 34 (d) 4T-Baricentro uniforme en un tetraedro sliver..... | 67 |
| Figura 34 (e) 4T-Baricentro uniforme en un tetraedro wedge..... | 68 |

INTRODUCCIÓN

El Procesamiento Geométrico es un área interdisciplinar de investigación y suele relacionarse con la integración de tres sub-áreas más concretas, Mallas, Subdivisión y Multi-resolución (Bischoff & Kobbelt, 2004).

Normalmente se denomina malla a un modelo poligonal (triángulos o cuadriláteros) bien sea en 3D (modelo volumétrico), o en 2D (superficie). La generación de mallas trata sobre los aspectos teóricos y prácticos de la discretización de superficies y volúmenes mediante estructuras denominadas mallas. La malla tiene la capacidad de llenar completamente la superficie o el espacio y a diferencia del resto de teselaciones la adyacencia entre elementos debe ser siempre un punto o nodo, una arista o una cara.

En la última década, el Procesamiento de Mallas poligonales se ha convertido en un área de investigación activa y muy productiva (Rivero, 2009), al extremo de que puede identificarse un campo de trabajo concreto dentro del CAGD (por sus siglas en inglés, *Computer Aided Geometric Design*), que puede denominarse Diseño y Modelado Geométrico mediante Mallas. Básicamente, el interés de las mallas puede atribuirse al desarrollo de:

1. Los modernos dispositivos de adquisición de geometría, como escaners, o Imágenes de Resonancia Magnética (MRI), los cuales producen mallas poligonales de gran tamaño y complejidad.
2. Herramientas de análisis, por ejemplo para imágenes médicas.
3. Las tarjetas graficas tipo GForce o NVIDIA utilizan estructuras de datos basadas en mallas para visualizar y/o acelerar gráficos en la pantalla del ordenador.
4. Fabricación Asistida por Ordenador, controles numéricos, prototipado rápido, etc.
5. Simulación Numérica, métodos de discretización espacial como de Elementos Finitos, Elementos de Contorno, de Volúmenes Finitos, de Diferencias Finitas, o métodos espectrales.
6. Visualización y Gráficos por Ordenador, Realidad Virtual, Visualización de Terrenos, Niveles de Detalle, compresión de imágenes, etc.
7. Sistemas de representación geográfica.

Asimismo, la investigación y desarrollo en mallas es de interés en otras áreas concretas. En la Figura 1, tomada de (Rivero, 2009), se muestra un esquema que relaciona el campo de actividad en mallas en relación con cuatro áreas conocidas, como el CAGD, los Métodos Numéricos, Gráficos por Ordenador y Geometría Computacional. En dicha figura se observa un núcleo común en torno a las mallas que comprende la generación, el procesamiento, el refinamiento y la calidad.

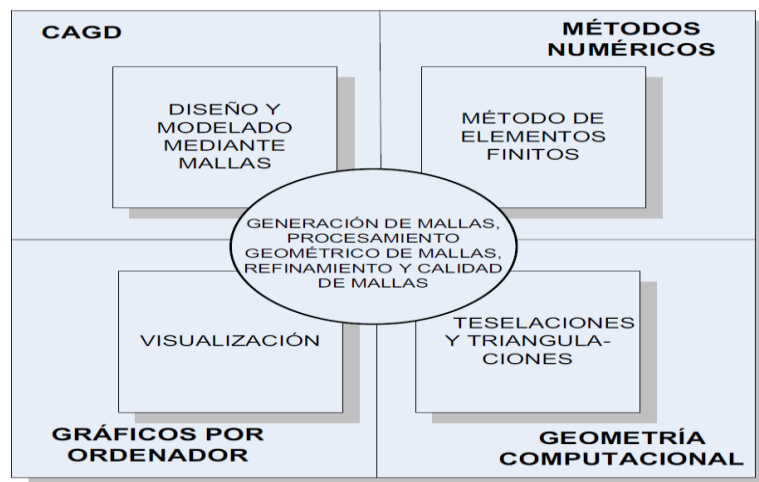


Figura 1 Esquema que clasifica el campo de trabajo de las mallas en relación con cuatro campos de investigación.

En todos estos casos de aplicación de mallas, se reconoce la necesidad de lograr una transición entre los datos de las mallas poligonales y la calidad geométrica de las mismas (Rivero, 2009). Por tanto, se necesitan algoritmos de generación de mallas que produzcan elementos triangulares con buena calidad en términos de su forma geométrica o de otros parámetros dependientes del área de aplicación.

Por ejemplo, para simulaciones numéricas u otras aplicaciones que trabajen a nivel de procesamiento de mallas, se tiene que garantizar la ausencia de elementos degenerados (M. Botsch & Kobbelt, 2001).

Para controlar la calidad de los triángulos se usa una medida que actúe como indicador de la buena o mala calidad. Esta medida de calidad es dependiente del área de aplicación (Rivero, 2009).

Fundamentalmente, los dominios 2D pueden ser divididos en triángulos o cuadriláteros; mientras que los dominios 3D, en tetraedros o hexaedros. Idealmente,

la forma y distribución de elementos en el dominio son realizados por los algoritmos de generación de mallas.

En el contexto de aplicación de las mallas, el problema del refinamiento/des refinamiento de mallas es de especial interés en áreas como Modelos Multiresolución, Niveles de Detalle o Aproximación de superficies como el terreno (Rivero, 2009).

Se define refinamiento como la técnica geométrica/algorítmica que subdivide triángulos en otros más pequeños añadiendo puntos y formando nuevos triángulos. Mediante las técnicas de refinamiento se persigue un aumento de la precisión en la malla del modelo, de forma que en zonas críticas para la simulación el refinamiento aumente la densidad de puntos triángulos, permitiendo así un estudio más detallado a nivel numérico.

La subdivisión por refinamiento no es arbitraria, ya que la distribución de los nuevos vértices en los triángulos y la generación de los nuevos triángulos han de cumplir ciertas condiciones, como ya se ha apuntado en (Velho & Zorin, 2001),(Bern & Eppstein, 1995),(Liu & Joe, 1995). El tiempo necesario para modificar la malla debería ser proporcional al número de elementos modificados. Para conseguir estos objetivos es necesario encontrar algoritmos y estructuras de datos que se ajusten a la estructura del problema y al esquema de refinamiento usado.

Para mallas triangulares, métodos de refinamiento que satisfagan conservación de la jerarquía (mallas que se anidan perfectamente en los distintos niveles de refinamiento) y conformidad (la unión de los triángulos respeta la adyacencia mediante sus lados) son bien conocidos (Rivero, 2009). Una importante clase de tales métodos lo constituyen los basados en la bisección (también denominado bipartición). La bisección, en el caso más sencillo, conecta uno de los vértices de un triángulo con el punto medio del lado opuesto. Dentro de esta clase, los algoritmos de M.C Rivara (Maria Cecilia Rivara, 1997; Maria Cecilia Rivara & Iribarren, 1996; Maria Cecilia Rivara & Levin, 1992) basados en bisección por el lado mayor son de amplia difusión y reconocimiento. Por su versatilidad y buenas propiedades, los métodos de refinamiento basados en la bisección por el lado mayor (LE) y sus variantes se han venido usando y extendiendo a muchas áreas de aplicación.

Para geometrías 3D (volúmenes), se puede extender el esquema algorítmico de la bisección dando pie a nuevos algoritmos basados en las ideas del caso

bidimensional. Ángel Plaza y colaboradores introdujeron un algoritmo de refinamiento 8T-LE_ (en ocho tetraedros por el lado mayor) (A. Plaza & Rivara, 2003). Se puede conocer entonces el número de elementos hijos que surgen en el proceso de refinamiento de tetraedros de mallas 3D.

Otros de algoritmos bien conocidos, tanto para el refinamiento como para la generación automática de mallas, son los que se agrupan con el nombre de triangulación Delaunay (Delaunay, 1934) (que proviene del tipo de triangulación que lleva su nombre). Este grupo de algoritmos posee excelentes propiedades y se aplica actualmente en muy diversas áreas de la Ciencia e Ingeniería (Rivero, 2009). Su uso en combinación con los algoritmos basados en la bisección (LE) también han sido propuestos recientemente en (M.C. Rivara, Hitschfeld, & Simpson, 2001).

Dada la cantidad de particiones existentes por el lado mayor para subdividir un triángulo, su extensión a 3D, y los criterios de inserción de puntos en combinación con Delaunay que pueden tenerse en cuenta, se plantea la necesidad de implementar una aplicación que permita asistir la toma de decisiones en un ambiente de incertidumbre en dependencia del algoritmo y de la malla en que se vaya a aplicar, debido a que muchos de estos esquemas de refinamiento no han sido estudiados en mallas tridimensionales y no se han obtenido resultados numéricos en cuanto a sus propiedades no degenerativas, tiempo de ejecución, y análisis del costo de los algoritmos. A partir de lo cual se plantea el siguiente **problema científico**: ¿cómo apoyar la toma de decisiones en el estudio de los algoritmos de refinamiento por el lado mayor y Delaunay con mallas de tetraedros?

El **objeto de estudio** en que se enmarca el problema planteado lo constituye el proceso de refinamiento de mallas de tetraedros.

La investigación tiene como **objetivo** estudiar en detalle el conjunto de algoritmos formado por LE, 3T-LE, LE-Trisección, 4T-Baricentro y los basados en la triangulación Delaunay de forma tal que permita asistir la toma de decisiones sobre cuáles algoritmos refinan una malla de tetraedros con mejor calidad o cuáles la degeneran, precisándose como **campo de acción** el proceso de refinamiento de mallas de tetraedros utilizando variantes de algoritmos 3D.

La investigación estará dirigida a demostrar la validez de la siguiente **hipótesis**:

En el conjunto de algoritmos formado por LE, 3T-LE, LE-Trisección, 4T-Baricentro y los basados en la triangulación Delaunay, al menos uno refina una malla de

tetraedros con mejor calidad evaluado por las métricas Etha, Whiteh, Ratio y Solid Angle.

Para llevar a cabo esta investigación se tendrán en cuenta las siguientes **tareas**:

1. Elaborar los fundamentos teóricos sobre tecnologías para la generación de mallas.
2. Desarrollar una estructura de datos para almacenar la malla de tetraedros.
3. Implementar los algoritmos de refinamiento global: LE, 3T-LE, LE-Trisección y 4T-Baricentro.
4. Implementar las variantes de algoritmos basados en la triangulación Delaunay.
5. Establecer las comparativas de las métricas de la calidad Etha, Whiteh, Ratio y Solid Angle en el refinamiento de mallas de tetraedros de los algoritmos implementados.

Para realizar las tareas se emplearán los siguientes métodos de investigación:

Métodos teóricos:

Análisis y síntesis: para procesar la información en la elaboración de los fundamentos teóricos y las conclusiones.

Histórico-lógico: para el estudio de la evolución del problema y la existencia de Sistemas de apoyo a la toma de decisiones similares al que se pretende elaborar.

Hipotético-deductivo: para la elaboración y prueba de la hipótesis, así como en la deducción de los resultados de la investigación.

Modelación: para el diseño de la aplicación escritorio y de las herramientas que se utilizarán en el mismo.

Enfoque sistémico: para el análisis y determinación de las relaciones funcionales y dependencias.

Métodos empíricos:

Entrevistas: para estudiar el estado de opiniones de los especialistas sobre el desarrollo de Sistemas de apoyo a la toma de decisiones, así como las posibilidades de uso de la aplicación Desktop propuesta.

Métodos Experimentales:

Pre experimentos: para estudiar los resultados numéricos de la calidad obtenida por los algoritmos de refinamiento con un mismo tipo de malla de tetraedros.

Experimentos puros: para estudiar los resultados numéricos de la calidad obtenida por los algoritmos de refinamiento con diferentes tipos de tetraedros.

Resultados esperados:

Con la implantación de este sistema de apoyo a la toma de decisiones y el estudio empírico de los algoritmos propuestos se espera obtener información oportuna sobre las propiedades no degenerativas de estos esquemas de refinamiento, robustez de los algoritmos implementados y sus propiedades computacionales, que servirán para aplicar los algoritmos en campos concretos como, simulación numérica, elementos finitos, visualización de terrenos y gráficos por ordenador.

Para destacar el interés que tiene el refinamiento de mallas dentro de un marco de aplicación, las aplicaciones de sistemas de información geográfica SIGs como Google Earth emergen con gran rapidez y todas ellas demandan una representación realista y tridimensional del terreno, las cuales necesitan de mallas optimizadas para garantizar precisión en la representación y eficiencia en la navegación virtual.

El trabajo que se presenta consta de dos capítulos. El *Capítulo 1* se titula “Fundamentos teóricos” y contiene los fundamentos de los aspectos relacionados con el objeto de estudio y del uso de conceptos del campo de generación de mallas, incluyendo ideas, algoritmos y estrategias de refinamiento. Asimismo, se presentan algunas ideas básicas de diseño de estructuras de datos.

El *Capítulo 2*, “Estudio empírico de los algoritmos propuestos”, contiene los aspectos de mayor peso de la investigación realizada, se dedica al estudio empírico de las particiones geométricas de elementos simpliciales en 3D, estos basados en el lado mayor y delaunay. Se presentan las propiedades matemáticas y computacionales de los algoritmos propuestos. Se arriba a las conclusiones del estudio realizado y se proponen diversas líneas de trabajo futuro.

CAPÍTULO 1 *FUNDAMENTOS TEÓRICOS*

1.1 Introducción

La generación de mallas es el proceso de discretización de un dominio físico en subdominios más pequeños (elementos). Las mallas se encuentran en diversas áreas del análisis numérico dentro de la Ciencia e Ingeniería aplicada. Ejemplos de estas son: ajuste de curvas y superficies en la teoría de aproximación, integración numérica, diferencias finitas, volúmenes finitos y elementos finitos (Rivero, 2001). Todas tienen en común la solución numérica de ecuaciones diferenciales en derivadas parciales, cuyo proceso en general puede ser descrito de la siguiente forma:

1. Definición inicial del dominio proporcionando algunos puntos.
2. Generación de la malla y etiquetado de nodos.
3. Solución numérica del problema en el dominio discretizado.
4. Refinamiento o redistribución de la malla y salto al paso 3 para mejorar la solución.
5. Presentación de la malla y solución (visualización).

No obstante, la generación de mallas se ha convertido actualmente en un tema interdisciplinar que aparece en muchas otras áreas. Puede citarse, por ejemplo, la industria gráfica, y más concretamente, gráficos por ordenador, diseño y modelado de sólidos y superficies, visualización, CAD/CAM, etc. Asimismo la geometría y la topología computacional hacen obligada visita al tema de generación de mallas. Otras áreas más aplicadas como cartografía por ordenador y sistemas de información geográficos (GIS) han venido a recurrir a técnicas de mallado y afines para la representación y visualización de datos espaciales.

Fundamentalmente, los dominios 2D pueden ser divididos en triángulos o cuadriláteros, mientras que los dominios 3D en tetraedros o hexaedros. Idealmente la forma y distribución de elementos en el dominio son realizados por los algoritmos de generación de mallas, automáticos o no automáticos. Serán automáticos, cuando la generación de mallas solo requiere de unos pocos parámetros iniciales y a continuación se realiza el proceso, no requiriendo la participación del usuario más

que al inicio. Serán no automáticos en caso contrario, es decir, el proceso de la generación de la malla requiere de decisiones y parámetros externos.

En décadas recientes, el método de elementos finitos ha llegado a ser un soporte fundamental en el diseño y análisis de aplicaciones industriales. Cada vez más, diseños complejos y de gran magnitud están siendo simulados usando el método de elementos finitos, lo que ha hecho aumentar la popularidad e interés por los algoritmos de mallado automático (Rivero, 2001).

El problema de la generación automática de mallas considera la mejor descripción de un dominio geométrico, sujeto a un conjunto dado de nodos iniciales, especificación de tamaño para los elementos y criterios de la forma de los elementos. La geometría del dominio se compone habitualmente de vértices, curvas, superficies y sólidos, que en el mejor de los casos se describen por paquetes de CAD o de modelado de sólidos.

Existen dos tipos fundamentales de mallas: estructuradas y no estructuradas. En las mallas estructuradas la conexión entre elementos obedece a un patrón fijo; en las no estructuradas no existe tal patrón, el número de nodos vecinos de uno dado es variable. Las mallas obtenidas por una transformación de coordenadas, de un cuadrado en dimensión dos o un cubo en dimensión tres, son ejemplos de mallas estructuradas. En ellas el nodo de índices (i, j, k) siempre tiene como vecino por la izquierda al $(i-1, j, k)$ y por la derecha al $(i+1, j, k)$, etc.

Las mallas estructuradas tienen ciertas ventajas sobre las no estructuradas. Son más simples y también más convenientes para su uso en el método de diferencias finitas. Requieren menos memoria para su almacenamiento en ordenador y es más fácil controlar la forma y tamaño de los elementos.

La gran desventaja que ofrecen las mallas estructuradas es su poca flexibilidad para adaptarse a dominios con geometrías complicadas (Rivero, 2001). Se han desarrollado numerosas técnicas para encontrar las transformaciones de coordenadas adecuadas: transformación conforme, métodos algebraicos, etc. Incluso con estas técnicas es imposible encontrar una transformación que se adapte satisfactoriamente a la forma de un dominio complicado. Existen multitud de problemas formulados en dominios cuya forma no es adecuada para una discretización mediante mallas estructuradas.

En estos casos es necesario utilizar mallas no estructuradas.

La generación automática de mallas no estructuradas es un campo relativamente reciente y que en poco tiempo ha producido grandes avances (Rivero, 2001). Esta disciplina utiliza conceptos de la geometría computacional, y por consiguiente del álgebra lineal y de la teoría de poliedros entre otros campos, y sus aplicaciones se encuentran principalmente en el método de los elementos finitos. Es frecuente que, los elementos que componen las mallas no estructuradas sean símplices, esto es, triángulos en dimensión dos y tetraedros en dimensión tres. La utilización de símplices es debida, entre otras cosas, a su capacidad para llenar el espacio y para adaptarse a los contornos del dominio sin que se produzcan discontinuidades. La malla formada por símplices recibe el nombre de triangulación y en dimensión tres teselación, o también triangulación en dimensión tres. Es sabido que es posible descomponer un dominio poliédrico (poligonal en dimensión dos) en tetraedros (triángulos), aunque esta descomposición no es única y puede requerir la inserción de puntos adicionales.

1.2 Estrategias de Refinamiento

En los últimos años variadas estrategias han sido desarrolladas para refinar mallas de triángulos y de tetraedros. Por ejemplo, se puede consultar bibliografía al respecto (Carey, 1997; Frey & George, 2000). El refinamiento basado en el lado más largo se ha popularizado en la última década en el contexto del refinamiento de mallas. En 2D, la bisección por el lado mayor de un triángulo t consiste en la partición del triángulo por el punto medio de su lado mayor y el vértice opuesto. Otra partición basada en el lado mayor es la partición de 4-triángulos (4T-LE) de un triángulo t que se define uniendo el punto medio de la arista mayor de t al vértice opuesto, y luego, uniendo ese punto medio con los puntos medios de las otras 2 aristas de t a través de segmentos paralelos a los lados, esta partición ha mostrado notables mejoras en la calidad (Maria Cecilia Rivara & Iribarren, 1996) dentro de ciertos límites estudiados recientemente en (Á. Plaza, Suárez, Padrón, Falcón, & Amieiro, 2004). La repetición arbitraria de este esquema de refinamiento por el lado mayor nunca produce un ángulo más pequeño que la mitad del ángulo mínimo original. No obstante, la aplicación recursiva de estos algoritmos de refinamiento pueden producir triangulaciones no conformes. La conformidad se refiere al requisito que la intersección de triángulos no disjuntos es o un vértice común o una arista

común. Una estrategia para garantizar la conformidad en la malla consiste en subdividir la arista mayor de los triángulos no conformes y por lo tanto el refinamiento se extiende por conformidad en una especie de efecto dominó. La propagación del refinamiento cambia de acuerdo al tipo de partición escogida, la bisección por el lado mayor propaga por una arista, la 3T-LE por la arista mayor y secundaria y la partición 4T-LE por las tres aristas del triángulo inicial. También en la búsqueda de mejores resultados de la calidad en el refinamiento iterativo de la malla, la partición 7T-LE basada en los siete triángulos por el lado mayor es obtenida primero, posicionando 2 puntos equidistantes por arista y luego, el interior del triángulo es dividido en 7 sub- triángulos de una forma compatible con la subdivisión de las aristas (Marquez, Moreno--Gonzalez, Plaza, & Suarez, 2008). Tres de los nuevos sub-triángulos son similares al original, dos son similares al triángulo nuevo generado por la 4T-LE, y los otros dos triángulos, son de forma general de mejor calidad. Para mostrar una ilustración de la familia de particiones de triángulos por el lado mayor, ver Figura 2.

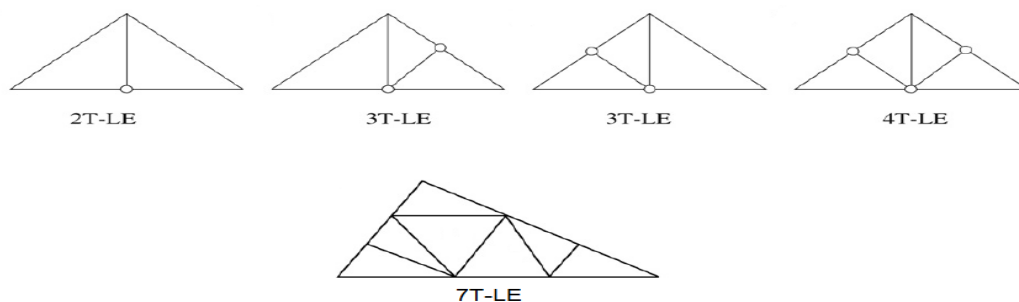


Figura 2 Particiones por el lado mayor de los triángulos.

Un número de algoritmos basados en bisección por el lado mayor de triángulos han sido implementados en software de mallas en dos dimensiones que también han sido generalizados para la tercera dimensión.

En la tercera dimensión variadas técnicas han sido desarrolladas para refinar mallas de tetraedros por medio de la bisección (A. Plaza, Padrón, & Carey, 2000; A. Plaza & Rivara, 2003; Maria Cecilia Rivara & Levin, 1992). En la literatura se puede encontrar particiones por el lado mayor bien estudiadas como LE, 4T-LE, 8T-LE, ver (A. Plaza & Rivara, 2003; Maria Cecilia Rivara & Levin, 1992), y sus respectivas referencias. Todas estas han demostrado ser particiones no degenerativas, que son factibles para la aplicación del método de elementos finitos.

Los algoritmos de generación de mallas basados en el refinamiento Delaunay han sido herramientas efectivas en teoría y práctica durante los últimos 20 años. Una nueva estrategia de refinamiento consiste en que, en vez de tener un número fijo de puntos a ser triangulados, se le da la libertad de escoger los puntos a ser insertados de forma tal que se construya una malla con una resolución deseada. La construcción de la malla se realiza de forma dinámica. Además, es posible explotar la existencia de una conocida triangulación construida por el algoritmo Delaunay de forma que se reduzca el costo computacional para construir la malla refinada.

En el contexto del problema del refinamiento de la triangulación, artículos científicos de Delaunay (Delaunay, 1934) establecen la propiedad fundamental de lo que ahora llamamos triangulación Delaunay, la propiedad de que todos los circuncentros de los triángulos no contengan puntos interiores de la triangulación.

En la mayoría de la literatura se distingue claramente entre la propiedad Delaunay, algoritmos para construir triangulaciones Delaunay y algoritmos para insertar nuevos puntos durante el refinamiento, ver Figura 3.

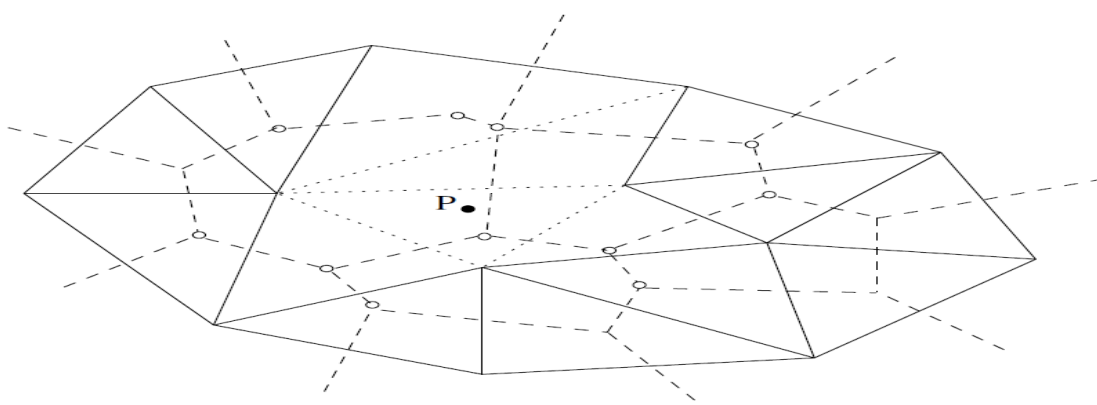


Figura 3 Inserción de un nuevo punto en una triangulación Delaunay .

Algoritmos de refinamiento Delaunay han tomado lugar en este contexto, ver referencias (Chew, 1993; Miller, Pav, & Walkington, 2002; Pav, 2003; Ruppert, 1995; Jonnathan Richard Shewchuk, 1997). La mayoría de ellos están basados en el algoritmo de Ruppert (Ruppert, 1995) .

Brevemente, de forma tal que se obtengan mallas Delaunay refinadas de calidad, se hace necesario el uso de una metodología de inserción de puntos que luego de arbitrarios pasos de refinamiento garantice la construcción de triangulaciones no degeneradas. Luego de añadir los nuevos puntos, el algoritmo Delaunay puede ser

usado para reconstruir la triangulación. La nueva malla puede ser completamente regenerada, re-triangulando todos los vértices en la malla anterior. Pueden ser considerados algoritmos eficientes de refinamiento que adecuadamente exploten las intrínsecas buenas características de la triangulación Delaunay.

1.3 Estructura de datos geométricas

Las estructuras de datos son decisivas para la eficacia de un algoritmo y de las aplicaciones relacionadas. Esta importancia es aún más notable en algoritmos geométricos, debido a la complejidad que conlleva el manejo de las entidades geométricas en 2D y 3D.

El problema de organizar y estructurar los datos en forma de estructuras de datos fue resuelto en la era inicial de la computación de forma simple. La simplicidad era respuesta necesaria ya que los dispositivos de entrada de datos al ordenador eran sencillos. Además, los primeros lenguajes de programación ofrecían posibilidades limitadas en la estructuración y manejo de los datos. Sin embargo, con el avance de la tecnología informática y el advenimiento de modernos lenguajes de programación, esta respuesta no es inmediata y ofrece múltiples consideraciones y posibilidades. Puede hablarse de que se ha pasado a una sofisticación de las estructuras de datos motivada por la necesidad de dar solución a problemas de creciente complejidad.

A continuación se enumera una serie de objetivos deseables de las estructuras de datos que tratan con objetos geométricos:

1. Capturar la información estructural (por ejemplo de la malla).
2. Permitir un procesamiento eficiente de las consultas de información
3. Permitir actualizaciones de la información lo más eficiente.
4. Optimizar el espacio de almacenamiento.
5. Almacenamiento eficiente en el sentido de minimizar el número de accesos de entrada/salida, cuando el volumen de datos es grande.

Concretamente, en el campo de la generación y refinamiento/desrefinamiento de mallas, la necesidad de una buena estructura de datos es crucial debido, entre otras, a las siguientes causas:

1. Entidades de origen espacial (geométrico y topológico).
2. Complejidad creciente en la dimensión del problema (2D,3D).

3. Volumen de datos considerable, que oscila entre miles y millones de elementos.
4. Creación, eliminación y actualización de forma dinámica de los elementos.

Todo esto ha hecho surgir áreas específicas de diseño de estructuras de datos. En este campo se denominan estructuras de datos geométricas y estructuras de datos espaciales.

Estructuras de datos clásicas como vectores, listas, árboles o grafos, no son por sí mismas muy adecuadas para la representación de objetos geométricos, ya que fundamentalmente están concebidas para problemas en dimensión 1 o bien no son capaces de representar las características espaciales de los objetos geométricos ni del algoritmo que las utiliza. Por ejemplo, en elementos finitos habitualmente es necesario mantener un orden determinado (sentido horario o anti horario) de las aristas dentro de un triángulo, o de las caras dentro de un tetraedro. Otras veces, por necesidad de los algoritmos de refinamiento o desrefinamiento, se necesita tener fácil acceso a los elementos vecinos a uno dado. Este tipo de necesidades se complican al aumentar la dimensión del problema, como se ha apuntado anteriormente. De esta forma, es preciso que la combinación adecuada de estructuras de datos o incluso el diseño de otras nuevas se adapten a los requerimientos del problema en cuestión.

Otro aspecto a tener en cuenta es que, por regla general, el diseño de estructuras de datos viene estrechamente relacionado con las características del algoritmo que las use (Goodrich & Ramaiyer, 2000). Para ello se requiere un cuidadoso diseño de las estructuras de datos, que además sean lo más genéricas posibles, pudiéndose así extender su uso a diversos problemas afines.

1.3.1 Estructuras de datos en la representación de mallas

En la representación computacional de mallas en 2D y 3D se suele distinguir tres tipos de informaciones:

1. La información geométrica, que describe la información matemática de la malla, como coordenadas de los puntos, descripción de curvas, etc.
2. Información de adyacencia, que describe las relaciones topológicas de los elementos que forman la malla. Esta información es necesaria porque a

menudo el algoritmo que trabaja con la malla necesita información de adyacencia entre las primitivas de una malla.

3. Información de atributos, que añade información semántica a los elementos o primitivas de la malla. Por ejemplo, asignar a cada vértice o cara un color, información de altura geográfica para cada nodo, asignar variables de temperatura, presión etc. Esta información de atributos depende del ámbito de aplicación de las mallas.

Se dice que una malla está completamente definida si se dispone de información suficiente de la geometría, adyacencia y atributos de la malla. Existen formas no completas de definir una malla y sin embargo la información puede ser suficiente para las necesidades de muchos problemas aplicados (Garimella, 2002).

Las estructuras de datos geométricas a menudo se denominan en función de las entidades topológicas o primitivas (vértices, aristas, caras y elementos o polígonos) que la representan. Por ejemplo, una estructura de datos basada en aristas debe su nombre a que implementa la representación de la malla en función de las aristas.

Una primera aproximación al representar una malla 2D es proporcionar la lista de los polígonos en función de los vértices. Otro enfoque simple consiste en representar las aristas de la malla en función de los vértices. En ambos casos se dispone de una lista en la que se representa para cada vértice, las coordenadas geométricas que lo definen. Por ejemplo, la malla de la Figura 4 (a) se representa, atendiendo a estas dos primeras aproximaciones, según las siguientes listas:

Lista de polígonos:

V₁,V₃,V₂

V₂,V₃,V₄

V₃,V₄,V₅

V₃,V₅,V₆

V₃,V₆,V₁

Lista de aristas:

V₁,V₂

V₂,V₄

V₃,V₄

V₄,V₅

V₃,V₅

V_3, V_1

V_1, V_6

V_6, V_5

V_3, V_2

V_6, V_3

Una extensión a 3D de dichas representaciones es inmediata si se consideran los poliedros que forman la malla en lugar de polígonos. En este caso es posible definir una lista que considere el conjunto de las caras en función de los vértices. La ventaja que tienen estas aproximaciones que definen las entidades de una malla en función de los vértices de la misma, es su simplicidad. Si el requerimiento de información es para la visualización de la malla, estas dos aproximaciones son muy adecuadas, por ejemplo en VRML (Virtual Reality Modelling Language) (Hartmann & Wernecke, 1998), que se está convirtiendo en el estándar para la representación y visualización de objetos 3D así como en OpenInventor (Wernicke, 1994). Además, la lista de polígonos ha sido la estructura de datos clásica ampliamente utilizada en elementos finitos, y por ejemplo el Partial Differential Equation Toolbox, en Matlab (Mathworks) usa este esquema para 2D. Sin embargo, estas estructuras de datos no permiten representaciones completas de mallas ya que no disponen de información de adyacencia. Aunque tampoco poseen información de atributos, esto no es un problema, ya que es inmediata sin más que asignar a cada vértice, arista o polígono (según el caso) dicha información adicional.

Un siguiente paso a la hora de representar una malla 2D es aumentar la información a almacenar. Consiste en disponer de una primera lista con las coordenadas de los vértices y dos listas adicionales, una que representa las aristas en función de los vértices y la otra que contenga las caras en función de las aristas. Una extensión a 3D implica considerar además la lista de los poliedros en función de las caras. A continuación se representa el tetraedro de la figura 4 (b) atendiendo a esta forma descrita:

- Lista de aristas + lista de caras + lista de tetraedros.

Lista de aristas:

- $e_1 : V_1, V_2$
- $e_2 : V_1, V_3$

- $e_3 : v_1, v_4$
- $e_4 : v_2, v_3$
- $e_5 : v_2, v_4$
- $e_6 : v_3, v_4$

Lista de caras:

- $f_1 : e_1, e_2, e_4$
- $f_2 : e_2, e_3, e_6$
- $f_3 : e_4, e_5, e_6$
- $f_4 : e_1, e_3, e_5$

Lista de tetraedros:

$t_1 : f_1, f_2, f_3, f_4$

La ventaja de esta representación es que almacena una información jerárquica de la malla ya que define las aristas en función de los vértices, las caras en función de las aristas y los poliedros en función de las caras. Esta cualidad es muy adecuada para los algoritmos de refinamiento basados en el esqueleto, pues dispone de forma jerárquica la información de los conjuntos k-esqueleto para una malla τ . Sin embargo, en líneas generales, esta representación supone almacenamiento excesivo y redundante además de no ser una representación completa por la falta de información de adyacencia.

En las siguientes secciones se estudian estructuras de datos para mallas 2D y $2_{1/2}D$ que permiten especificaciones completas de mallas al considerar relaciones de adyacencia entre los elementos de la malla, ver (Goodrich & Ramaiyer, 2000; Weiler, 1985).

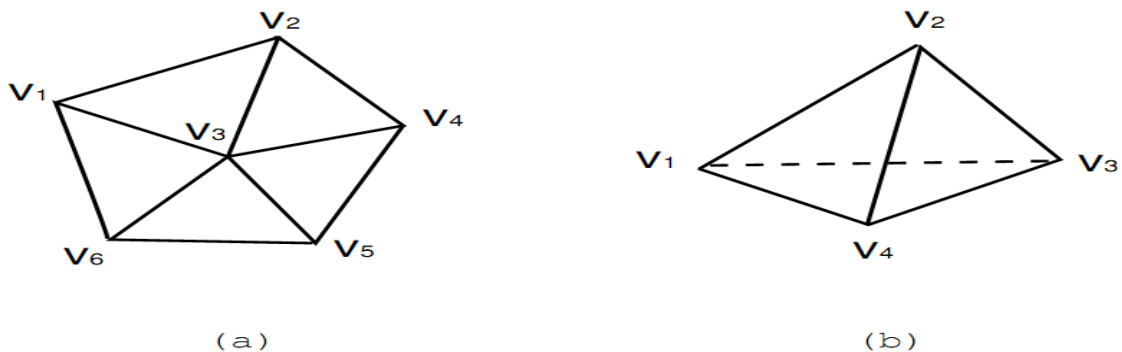


Figura 4 Ejemplos de mallas en 2D y 3D .

1.3.2 Relaciones de adyacencia completas

Una forma de representar las relaciones de adyacencia entre las primitivas topológicas del esqueleto es proporcionar enlaces de información (punteros, desde el punto de vista computacional) a cada una de las primitivas adyacentes a una dada. Según este esquema es posible derivar estructuras de datos basadas en vértices, aristas o caras. En la figura 5 se presentan esquemáticamente las tres estructuras de datos derivadas:

- (a) Basada en vértices.
- (b) Basada en aristas.
- (c) Basada en caras.

Analizando las tres variantes de la estructura de datos con adyacencia completa, puede notarse que las estructuras de datos basadas en vértices y caras, figura 5 (a) y (c) no poseen tamaño fijo, ya que el número de punteros depende del número de elementos incidentes al dado. Esta dificultad implica que el coste de almacenamiento es variable y depende de la malla en cuestión.

Asimismo, el costo para hacer una consulta depende de la complejidad local de la malla. En general sucede que las dos estructuras de datos ocupan mucho espacio y el coste para mantenerlas en todo momento actualizadas es alto.

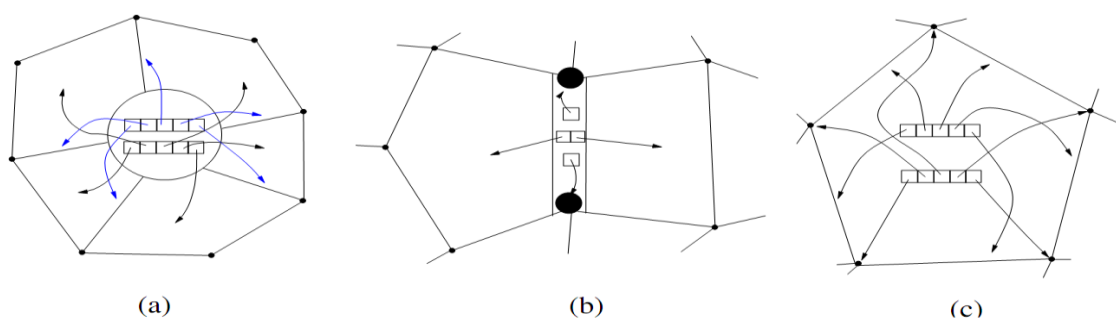


Figura 5 Tipos de adyacencia completas .

Por el contrario, la estructura de datos basada en aristas, Figura 5 (b), no posee estos problemas, ya que a lo más, una arista tendrá dos caras y dos vértices adyacentes, salvo cuando pertenece al contorno de una malla, en cuyo caso comparte una. Debido a que la estructura basada en aristas posee esta cualidad, la hace eficiente en términos de almacenamiento. Para una completa descripción de estructuras de datos basadas en aristas ver (Weiler, 1985).

1.4 Algoritmos de refinamiento y particiones en 3D

En lo que sigue se presentan diversas definiciones que corresponden a particiones de tetraedros en dimensión 3. Para una mejor apreciación gráfica de las particiones se ha optado por representar no sólo la vista en 3D, sino además una vista auxiliar análoga a la de Bansch (Bansch, 1991), que surge al desarrollar las 4 caras del tetraedro en el plano y transformar dichas caras a triángulos equiláteros.

Definición 1 (Partición Similar) El tetraedro original es dividido en ocho subtetraedros cortando las cuatro esquinas por planos paralelos a la cara opuesta y dividiendo el octaedro interior resultante en cuatro subtetraedros más, ver Figura 6.

En la división del octaedro interno en la partición similar, es relevante la elección de la arista interna, ya que ello permite optimizar la forma de los cuatro tetraedros que se generan en su interior.

Aunque se llamará a esta partición, partición similar, hay que señalar que sólo los cuatro tetraedros hijos que están en los vértices del tetraedro original son semejantes a él.

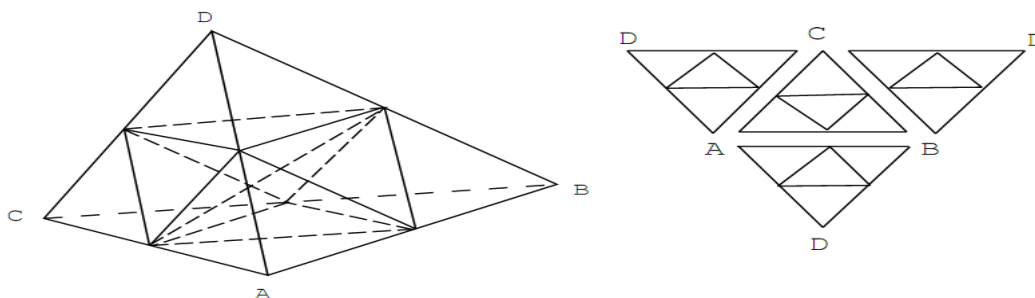


Figura 6 Partición similar en 3D y vista coplanaria de las caras.

Definición 2 (Partición 8T-LE) Para un tetraedro t con arista mayor única, la partición 8T-LE (8 Tetrahedra Longest-Edge) de t se define así:

1. Bisección por la arista mayor de t produciendo tetraedros t_1, t_2 .
2. Bisección de t_i por el punto medio de la única arista de t_i que es arista mayor de la cara común con el tetraedro original t , produciendo tetraedros t_{ij} , para $i, j=1, 2$.
3. Finalmente, bisección de cada t_{ij} por el punto medio de la única arista común con el tetraedro original t .

La Figura 7 muestra uno de los posibles casos al aplicar la partición 8T-LE.

Definición 3 (Partición 3D Bari-céntrica) Para un tetraedro t , la partición baricéntrica se define así, ver figura 8:

1. Añadir un nodo nuevo P en el baricentro de t y en los baricentros de las caras y aristas de t .
2. En cada cara de t aplicar la partición baricéntrica 2D de la cara.
3. Unir el baricentro P con todos los vértices de t y con todos los nuevos nodos añadidos anteriormente.

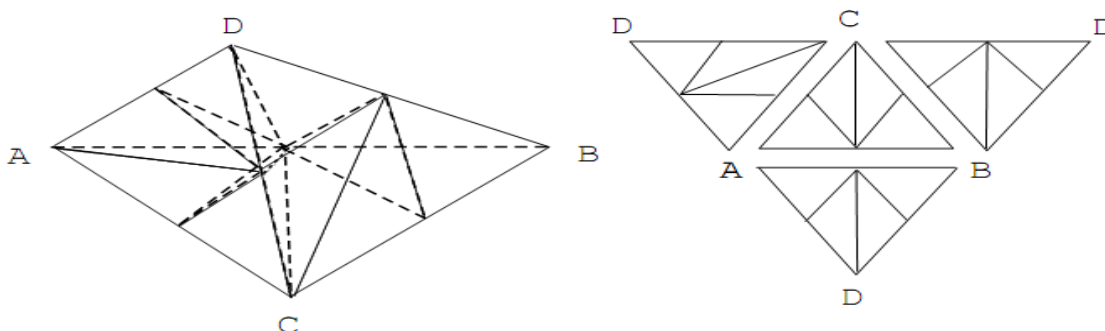


Figura 7 Caso ilustrativo de la partición 8T-LE.

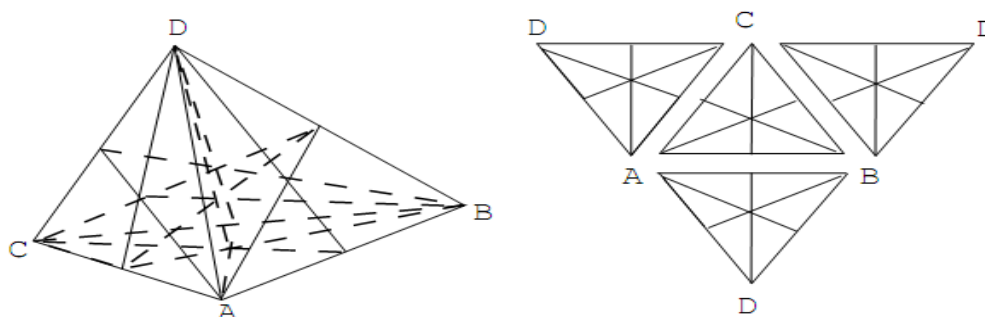


Figura 8 Partición Baricéntrica 3D .

Un algoritmo de refinamiento basado en la partición similar es el de Bey (Bey, 1995). Para refinamientos irregulares a la hora de alcanzar la conformidad, Bey considera cuatro patrones que cubren 25 de los 64 posibles, como se ve en la Figura 9 (nótese que hay $2^6 = 64$ posibilidades, porque sólo considera el número de nuevos nodos que aparecen en los puntos medios de las aristas, y no su posición relativa de acuerdo con la longitud de las mismas). Para el resto de los casos realiza refinamientos globales. Esto implica que en determinados casos puede producirse un efecto dominó, en el sentido de tener una propagación excesiva del refinamiento a través de la malla.

E. Bansch (Bansch, 1991) presenta un algoritmo basado en la elección de una arista como arista de refinamiento global en cada tetraedro, pero debe imponer pequeñas perturbaciones en las coordenadas de los nodos para evitar ciertas incompatibilidades.

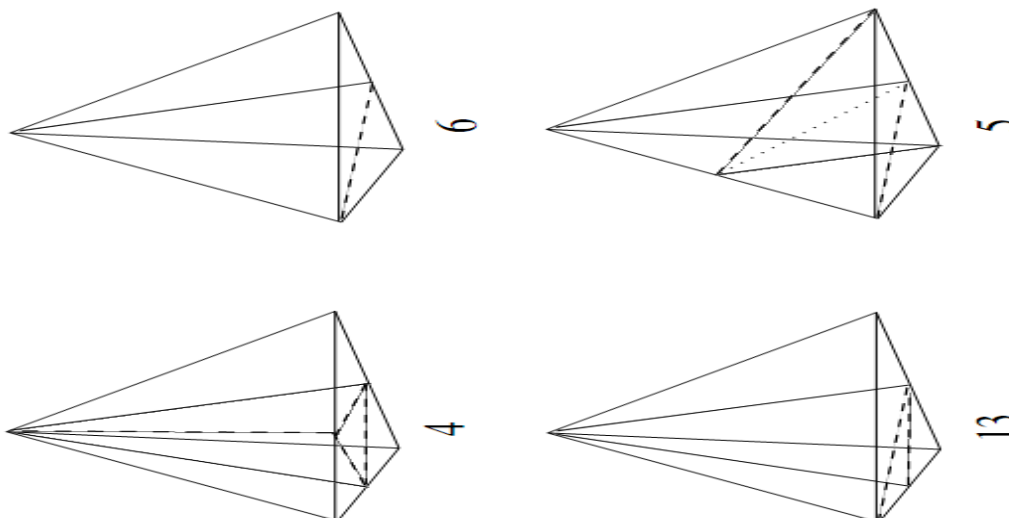


Figura 9 Patrones usados por Bey que cubren $(4 + 6 + 12 + 3) = 25$ posibilidades.

El clasifica los tetraedros en dos tipos, denominados rojo y negro, dependiendo de la posición relativa de las aristas de división y para ello al principio los abre sobre el plano.

En la Figura 10 se muestran los tetraedros tipo rojo y negro abiertos, la arista de refinamiento global con doble líneas de puntos y las siguientes en el orden de bisección con una simple línea de puntos.

Una vez cerrados los tetraedros, ver la Figura 11, se observan más claramente los tipo rojo y negro, aunque los dos tetraedros que dan cuenta de los tetraedros de tipo

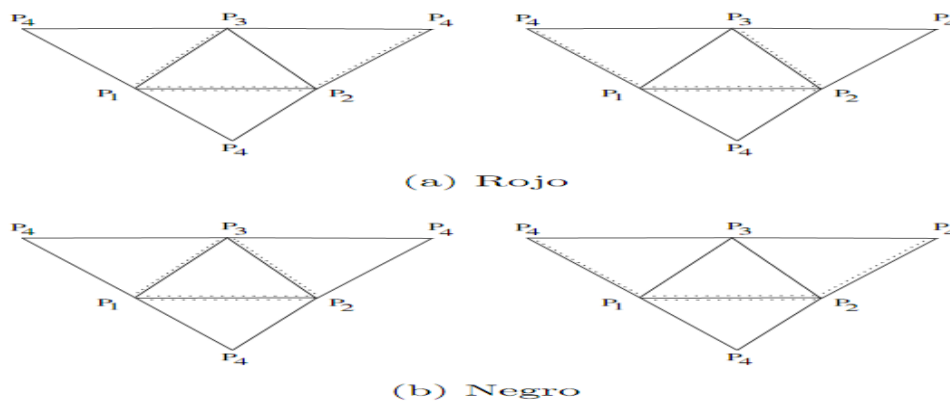


Figura 10 Tetraedros abiertos clasificados como rojo y negro.

rojo son equivalentes bajo una rotación de π radianes a través de un eje que pase por los puntos medios de las aristas con nodos P_1 y P_2 y la opuesta con nodos P_3 y P_4 .

Algoritmos de refinamiento de mallas de tetraedros basados en la bisección son los de Muthukrishnan (Muthukrishnan, Shiakolas, Nambiar, & Lawrence, 1995) y Rivara y Levin (Maria Cecilia Rivara & Levin, 1992). Ambos dividen los tetraedros por bisección por el lado mayor. Para alcanzar la conformidad, también se aplica bisección por el lado mayor de tetraedros adyacentes. Aunque no se realiza un estudio teórico-matemático sobre la no degeneración de las mallas creadas por medio de este método, los datos experimentales aportados dan a entender la buena

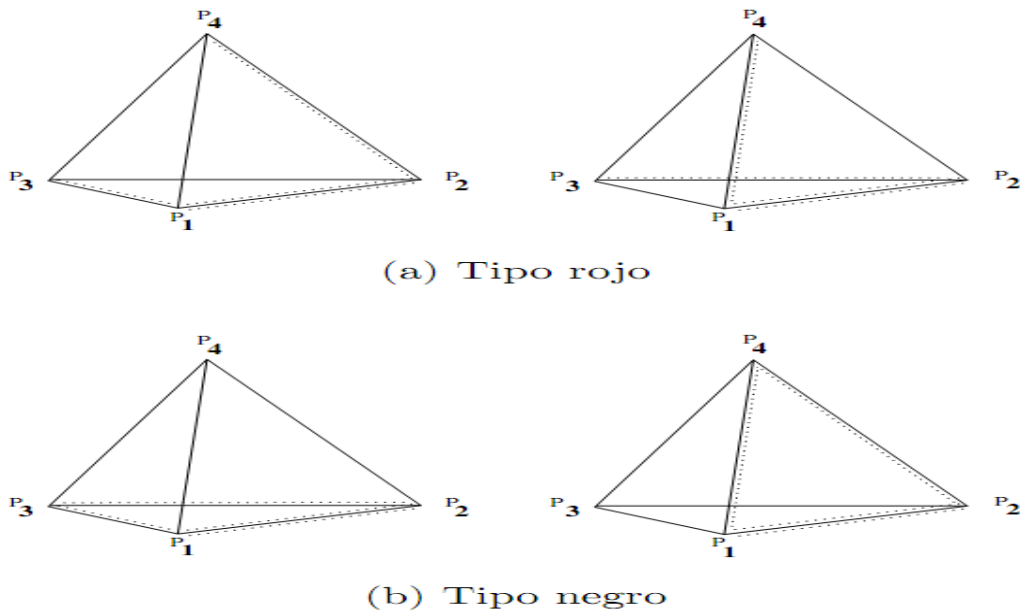


Figura 11 Tetraedros tipo rojo y negro.

calidad de las mallas obtenidas por medio de este algoritmo, y que incluso se tiene una cierta propiedad de mejoramiento de la malla cuando se parte de una malla inicial muy mala.

Recientemente, Plaza y Carey (A. Plaza & Carey, 2000) han presentado una generalización del algoritmo 4T de Rivara a tres dimensiones. Esta extensión ha sido llamada algoritmo 3D-SBR (3D-Skeleton Based Refinement). Algunas propiedades geométricas de este algoritmo han sido descritas por Rivara y Plaza (A. Plaza & Rivara, 2003). El algoritmo trabaja primero sobre las caras triangulares de los tetraedros, el 2-esqueleto de la teselación 3D y después subdivide el interior de cada tetraedro de forma congruente con la subdivisión del esqueleto. Esta idea podría ser

aplicada para desarrollar algoritmos semejantes en dimensiones mayores. El algoritmo se puede utilizar para cualquier malla inicial de tetraedros sin ningún pre proceso. Esta es una propiedad importante con relación a otros algoritmos similares. Como en el refinamiento, también el desrefinamiento, el algoritmo inverso, está basado en el 2-esqueleto y utiliza los mismo patrones de elementos usados al refinar. Ambos algoritmos son de complejidad lineal y han sido aplicados a diversos problemas (Padron, 1999; A. Plaza, et al., 2000).

Liu y Joe (Liu & Joe, 1994a, 1995) presentan un algoritmo similar al de Bansch (QLRB:Quality Local Refinement Based Algorithm). Ellos clasifican los tetraedros en cuatro tipos y clasifican los tipos de aristas dependiendo del tipo de tetraedro, como se ve en la figura 12.

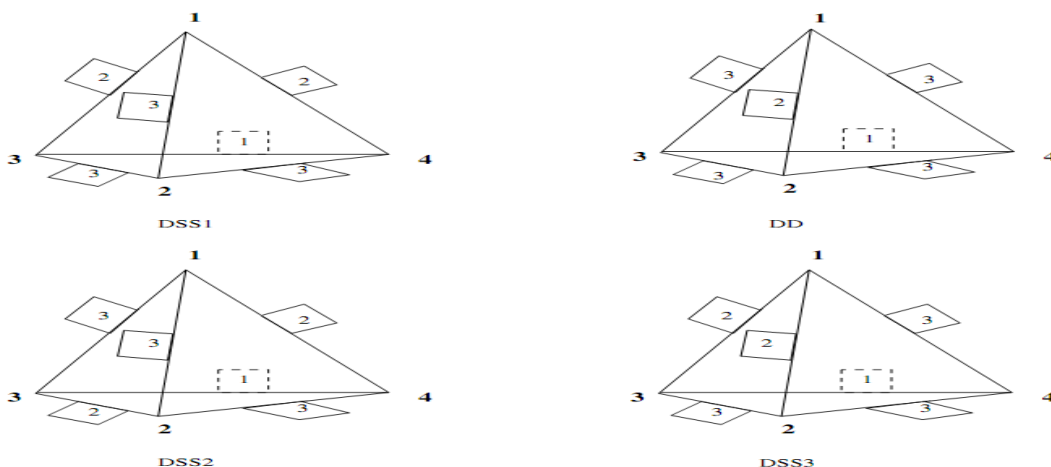


Figura 12 Clasificación de los tetraedros según Liu y Joe.

La clasificación de los tetraedros en cuatro tipos se realiza de la forma siguiente:

1. **DD:** Aparecen aristas marcadas 1 y 2 en aristas opuestas, siendo la arista etiquetada 1 la arista de referencia.
2. **DSS1:** La arista opuesta a la arista de referencia del tetraedro es etiquetada 3 y las aristas etiquetadas 2 se encuentran en la misma cara.
3. **DSS2:** Las aristas tipo 2 se encuentran en aristas opuestas.
4. **DSS3:** Las aristas tipo 2 se encuentran en la misma cara pero una de ellas es opuesta a la arista de referencia.

Se han desarrollado otros estudios similares a los de Bansch y Liu y Joe. Kossaczky (Kossaczky, 1994) desarrolla una aproximación recursiva. Su algoritmo necesita imponer ciertas restricciones y hacer un pre proceso a la malla inicial. Su propuesta es equivalente a las de Bansch y Liu y Joe en el sentido de que produce las mismas

mallas al refinar. Por otro lado, Maubach (Maubach, 1995) desarrolla un algoritmo para mallas de n -símplices generadas por reflexión. Aunque el algoritmo es válido para cualquier dimensión y el número de casos de semejanza es acotado, no puede ser aplicado a cualquier malla inicial de tetraedros. Es necesario un cierto refinamiento adicional para evitar ciertas incompatibilidades. Recientemente Mukherjee (D.N. Arnold, 2000; Mukherjee, 1996) ha presentado un algoritmo equivalente al de Bansch y prueba la equivalencia con el de Maubach.

1.5 Algoritmos de generación de mallas basados en Delaunay.

La triangulación de Delaunay es indiscutiblemente la técnica de triangulación más usada para triangular un conjunto de puntos dados o para triangular un polígono definido por sus vértices. Varios algoritmos han sido propuestos para la construcción de triangulaciones de Delaunay en el plano (Green, 1978),(Lawson, 1977),(Lee, 1980),(Bowyer, 1981),(Watson, 1981). Algunos criterios para analizar los algoritmos de triangulación son los siguientes (Lewis, 1976):**Rapidez**(el algoritmo debería ser en lo posible de costo lineal).**Calidad** (el algoritmo debería generar triángulos lo más equiláteros posible). **Generalidad** (el algoritmo debería ser capaz de tratar eficientemente geometrías complejas).

Una revisión de la literatura respecto de los métodos de construcción de triangulaciones de Delaunay (MORALES, 2000), permite distinguir cinco ideas básicas:

- a) **Test del círculo:** Consiste en insertar un punto en una triangulación de Delaunay y realizar el test del circuncírculo vacío para determinar si los nuevos triángulos generados son o no de Delaunay.
- b) **Inserción incremental de puntos:** Comenzar con un triángulo y agregar un punto a la vez, asegurándose que en cada paso la triangulación sea localmente óptima.
- c) **Intercambio de diagonales:** A partir de una triangulación inicial, iterar a través de la lista de cuadriláteros convexos y hacer intercambios donde sea necesario para asegurar optimización local.
- d) **Cavidad de inserción:** Dada una triangulación inicial, insertar un punto y verificar si existe un conjunto de triángulos cuyos circuncírculos contienen el

nuevo punto insertado, si es así, entonces formar la cavidad de inserción y generar nuevos triángulos.

- e) **Dividir para reinar:** Recursivamente dividir los datos en subconjuntos, encontrar las triangulaciones localmente óptimas para cada subconjunto y luego mezclar las triangulaciones resultantes.

A continuación, se revisa el método para construir triangulaciones Delaunay basada en el test del círculo.

Este test, llamado también test Delaunay, se usa para verificar si la nueva triangulación es localmente de Delaunay. El test básico en el algoritmo de Delaunay lo constituye la prueba del circuncírculo vacío, tres puntos constituyen un triángulo sobre una triangulación de Delaunay, si el circuncírculo a través de estos puntos no contiene otro punto en su interior.

Por ejemplo, sea el conjunto de puntos de entrada $V_n=\{a,b,c,d,e\}$ que conforman los triángulos abc y ace , compartiendo la arista ac , distribuidos como se muestra en la figura 13.

En la figura 13 , el triángulo abc es de Delaunay dado que no existe ningún otro vértice de la triangulación en el interior de su circuncírculo. En cambio, el triángulo ace no es un triángulo de Delaunay, ya que en el interior de su circuncírculo existe otro vértice d .

El test de Delaunay presenta la desventaja de ser poco robusto, es decir puede fallar debido a problemas de precisión al usar aritmética de punto flotante.

Se han abordado trabajos donde se explican estos fallos de Delaunay como por ejemplo caso en que falla el test debido a la colinearidad de los puntos (Marquez, Moreno-Gonzalez, Plaza, & Suarez, 2010).

La extensión del algoritmo a tres dimensiones trae consigo el problema de la generación de tetraedros muy deformados, los llamados Sliver, que no son detectados por el algoritmo y que pueden llevar a imprecisiones posteriores en los cálculos (Dari, 1994).

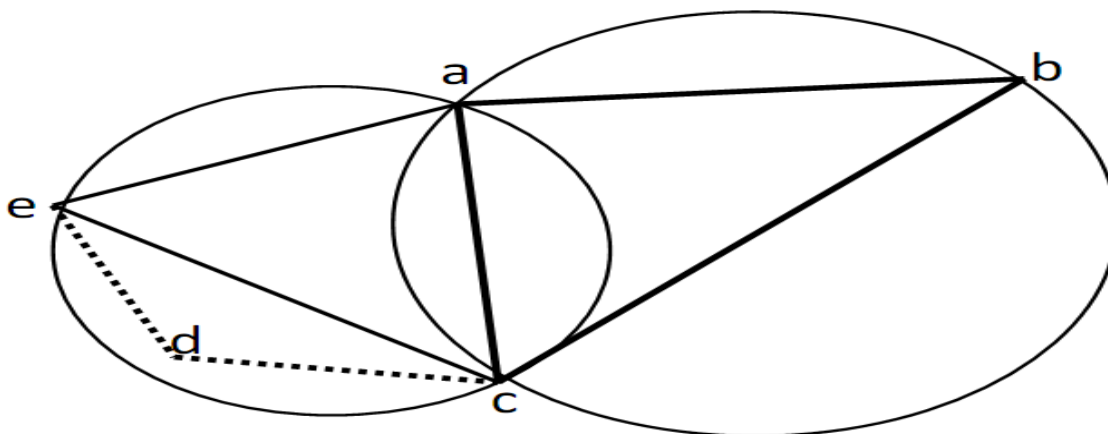


Figura 13 Test del círculo.

En tres dimensiones se sigue cumpliendo la condición del círculo circunscrito (en este caso de la esfera circunscrita), es decir, que ningún punto es interior a la esfera que pasa por los cuatro vértices de un tetraedro. Sin embargo, la propiedad de maximización del mínimo ángulo del caso bidimensional, no tiene su equivalente en las triangulaciones tridimensionales.

Diremos que un tetraedro K falla el test Delaunay con el punto P si el punto P es interior a la esfera que pasa por los cuatro vértices de K .

Llamaremos cavidad Delaunay correspondiente al punto P a la unión de los tetraedros K_j , tales que fallan el test Delaunay con el punto P .

El algoritmo de construcción de la triangulación, basado en la propiedad de la esfera circunscrita, es extensión directa del caso bidimensional: En primer lugar se genera una malla inicial que incluya a todos los puntos (ver figura 14). Luego para cada punto se determina la cavidad Delaunay correspondiente, se eliminan los tetraedros que la componen, y finalmente se agregan en la triangulación aquellos tetraedros formados por el punto y cada una de las caras de la cavidad.

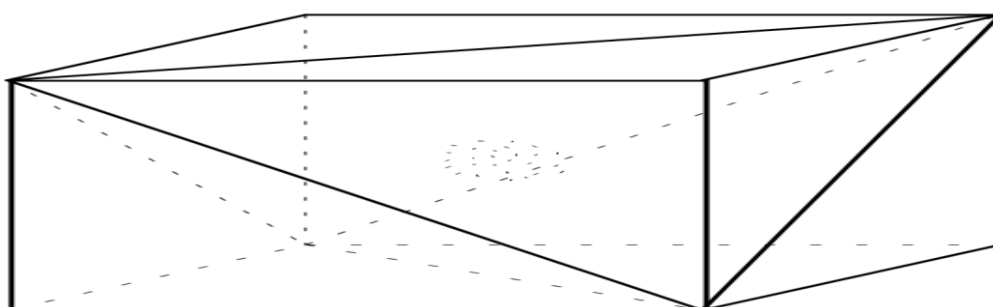


Figura 14 Malla inicial para el algoritmo de Watson en 3D .

La validez de este procedimiento para el caso de tres dimensiones se puede ver demostrada en (T.J.Baker, 1987).

1.6 Medidas de la calidad y la degeneración.

Aún permanece como un problema abierto la búsqueda de medidas de la calidad que permitan calificar a una malla no estructurada de tetraedros de forma relevante. Existen unas pocas medidas comunes que son propuestas en un formato estándar, que deben ser suficientes para caracterizar la calidad de una malla.

Nos referimos a los artículos de Liu y Joe (Liu & Joe, 1994a, 1994b, 1995) , por las definiciones, discusiones y ecuaciones de las diferentes formas de medidas de los tetraedros. Otras revisiones sobre las formas de medida de los tetraedros aparecen en Dompierre (Dompierre, Labbé, Guibault, & Camarero, 1998) y Parthasarathy (Parthasarathy, Graichen, & Hathaway, 1993).

La siguiente definición es tomada de (Liu & Joe, 1994b):

Una forma de medida de la calidad de un tetraedro es una función continua que evalúa la calidad de un tetraedro. Debe de ser invariante bajo traslación, rotación, reflexión y escalado uniforme de un tetraedro. Debe tomar valor máximo para el tetraedro regular y debe tomar valor mínimo para un tetraedro degenerado. Para facilitar las comparaciones, debe ser escalado en el intervalo $[0,1]$,se toma el valor 1 para el tetraedro regular como el máximo global y 0 para un tetraedro degenerado considerado el mínimo global.

Nos referimos a (George & Borouchaki, 1997, 1998; Liu & Joe, 1994b; Parthasarathy, et al., 1993), por la definición y clasificación de un tetraedro degenerado. De forma breve, un tetraedro degenerado es aquel cuyo volumen se hace nulo.

Cuando el volumen de un tetraedro es negativo, el tetraedro esta más que degenerado, se encuentra invertido. En este caso, algunas medidas de la calidad de los tetraedros devuelven un número negativo, otras devuelven un número positivo debido a que dependen del cuadrado del volumen.

A continuación se hace una revisión de las medidas de la calidad existentes.

Algunas medidas de la calidad de la malla o de la degeneración son las siguientes:

Whitehead (Whitehead, 1940) introdujo lo que se conoce como grosor relativo de un símplice S : $\rho(S) = \frac{\tilde{r}(S)}{\delta(S)}$ donde el radio $r(S)$ es definido como la distancia desde el centroide S hasta la frontera y $\delta(s)$ es el diámetro de S .

Stynes (Stynes, 1980) introdujo $t(S) = \frac{\tilde{area}(S)}{\delta^2(S)}$, para el método de la bisección aplicado a los triángulos. Este ratio puede ser generalizado a dimensiones superiores como por ejemplo: $t(S) = \frac{\tilde{volumen}(S)}{\delta^n(S)}$, donde n es la dimensión de símplice S .

Otros autores (ver por ejemplo (Bern & Eppstein, 1995)) han utilizado el ratio $(S) = \frac{\tilde{r}(S)}{R(S)}$, donde $r(S)$ es ahora la longitud del radio de la esfera inscrita y $R(S)$ es el radio de la esfera circunscrita.

Para un tetraedro t y para cada vértice $P \in t$, Rivara y Levin (Maria Cecilia Rivara & Levin, 1992) introdujeron la medida Φ_p asociado con el ángulo sólido Ω_p en P :

$\Phi_p = \sin^{-1}\{(1 - \cos 2\alpha_P - \cos 2\beta_P - \cos 2\gamma_P + 2 \cos \alpha_P \cos \beta_P \cos \gamma_P)^{1/2}\}$, donde α_P , β_P , γ_P son los ángulos asociados a las esquinas. Luego ellos definen $\Phi_t = \min\{\Phi_p: P \in t\}$.

La relación entre Φ_p y el ángulo sólido Ω_p en P puede ser vista en (Liu & Joe, 1994b; A. Plaza & Carey, 2000).

Liu y Joe (Liu & Joe, 1994a, 1995) introdujeron el estimado $\eta(t) = \frac{12(3v)^{\frac{2}{3}}}{\sum_{i=1}^6 l_i^2}$, para cualquier tetraedro t donde v es el volumen de t y l_i es la longitud de las aristas de t .

La figura 15 muestra los distintos tipos de formas de los tetraedros que pueden surgir al refinar las mallas (Bern & Eppstein, 1995; Parthasarathy, et al., 1993). Se le asignan nombres descriptivos a los cinco distintos tipos de elementos tridimensionales: Un tetraedro regular (Round) no tiene ningún ángulo malo de ningún tipo. Un tetraedro aguja (Needle) o delgado tiene un solo ángulo sólido pequeño. Un elemento tipo Wedge tiene pequeños pero no largos ángulos diedros y no tiene ángulos grandes de ningún tipo. Un Sliver tiene pequeños pero largos ángulos diedros pero no tiene ángulos sólidos largos. Un tetraedro tipo gorra (Cap) tiene un solo largo ángulo sólido. Para el tetraedro tipo gorra, el radio de la esfera circunscrita es más largo que la arista mayor.

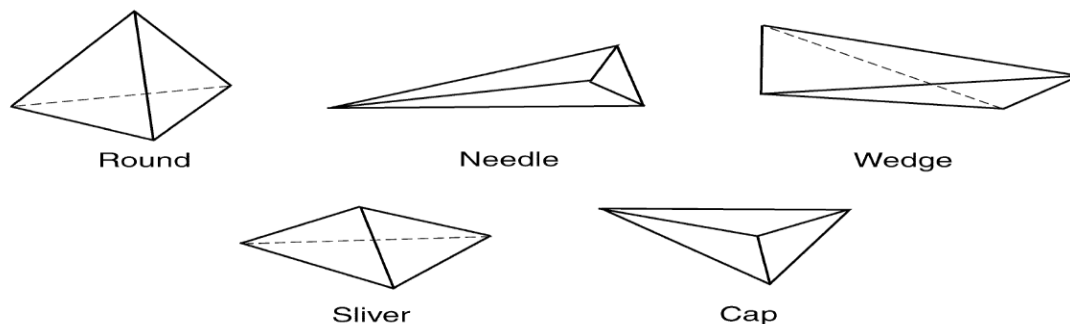


Figura 15 Distintos tipos de forma de los tetraedros .

Conjeturamos que todas estas medidas de la calidad son equivalentes en el siguiente sentido: todas ellas toman un valor máximo para el tetraedro equilátero, y si alguna de ellas tiende a 0 las demás también lo hacen indicando un tetraedro de mala calidad. En (Liu & Joe, 1994b) se demuestra la equivalencia entre todas estas medidas.

Por definición todas estas formas de medidas serán sensibles a los tetraedros degenerados o de mala calidad.

Considerar la media de todas estas formas de medida para los tetraedros como herramienta estadística garantiza que el orden de los resultados sea siempre el mismo con pocas permutaciones(Dompierre, et al., 1998).

1.7 Librerías de código abierto para la generación de mallas.

Diseñar e implementar una librería para la generación de mallas que este equipada con funcionalidades, que sea eficiente en términos de tiempo y memoria, robusta y fácil de utilizar constituye una tarea nada trivial.

Cuando una malla de elementos en 2D o 3D necesita ser desrefinada, refinada, el empleo de librerías para la generación de mallas se hace necesario para el desarrollador. En este trabajo investigamos varias librerías de código abierto que son las más utilizadas para la generación de mallas de triángulos y tetraedros. Se establece una comparación exhaustiva de todas las librerías propuestas.

CGAL (The Computational Geometry Algorithms Library) (Rineau, 2010):

Es una librería genérica escrita en C++ para geometría computacional. Proporciona primitivas geométricas básicas y operaciones, así como una colección estándar de estructuras de datos y algoritmos geométricos, incluyendo superficies de poliedros 3D e implementación de triangulaciones en 2D y 3D (Mario Botsch, Pauly, Rösxl,

Bischoff, & Kobbelt, 2006).CGAL esta específicamente diseñada para proporcionar soluciones confiables en cuestiones de eficiencia y robustez que son de importancia crucial en los algoritmos geométricos. La robustez y escalabilidad de los algoritmos se garantizan a partir del empleo de plantillas (*templates*).

Principales características:

- Algoritmos para el *Convex Hull* (2D y 3D).
- Triangulación Delaunay (2D y 3D).
- Estructura de datos basada en el esqueleto 2D.
- Estructura de datos de poliedros 3D.
- Triangulación Delaunay Restringida.
- Algoritmos de refinamiento (Catmull-Clark, Loop, Doo-Sabin and $\sqrt{3}$).

OpenMesh

Proporciona estructuras de datos eficientes para mallas poligonales y varios algoritmos estándar de procesamiento geométrico. Se encuentra altamente especializado en el procesamiento eficiente de mallas de superficie basada en estructuras de datos Half-edge (Silva, 2007) tomando algunos conceptos de CGAL. Se encuentra más enfocada en los requerimientos de modelado con las mallas poligonales y está equipada con un conjunto de algoritmos de suavizado de mallas y desrefinamiento.

Principales características:

- Desarrollada en C++.
- Estructura de datos Half-edge para mallas de triángulos.
- Módulo de desrefinamiento.
- Permite la creación de mallas multiresolución.

Mesh Library

Es una librería de propósito general para procesamiento geométrico. Ha sido probada para desrefinamiento progresivo, rastreo de siluetas y aplicaciones de cálculo para la curvatura gaussiana (Gu, 2010).

Entre algunas características de la librería podemos encontrar:

- Estructura de datos tipo Half-edge para mallas de triángulos.
- Soporta algoritmos de colapso de aristas y función de división de vértices.
- Escrita en C++.
- Lectura de ficheros.

- Visualizador de la malla integrado.

La limitación fundamental de la librería radica en que solo soporta mallas de triángulos debido a las características de la estructura de datos que implementa.

MESH2D

Es una librería de rutinas para mallas de triángulos no estructuradas aplicadas a geometrías generales 2D. Un algoritmo iterativo es incluido para optimizar la topología de la malla de forma tal que se obtengan triangulaciones de buena calidad (Engwirda, 2009).

Principales características:

- Desarrollada en Matlab.
- Soporte para mallas triangulares.
- Estructura de datos Quadtree.
- Rutinas para triangulación Delaunay.
- Algoritmos de refinamiento Quadtree.
- Módulo de optimización de mallas.

VCGLib

Es una librería portable de plantillas C++ para la manipulación y procesamiento de mallas de triángulos y de tetraedros. La librería está liberada bajo licencia GPL (General Public License) como resultado de los esfuerzos colaborativos del Visual Computing Lab del ISTI (Italian National Research Council).

VCGLib ha sido utilizado para construir herramientas exitosas en el campo investigativo como por ejemplo, Metro, una herramienta estándar para medir diferencias entre mallas de triángulos. ShadeVis, una herramienta para la computación de vértices en ambientes estáticos. Tridecimator, una herramienta eficiente para la simplificación de mallas y Trimeshinfo, diseñada para inspeccionar modelos 3D y obtener toda la información topológica relacionada.

Otros dos notables proyectos basados en esta librería son MeshLab (Cignoni, 2005-2009) que es un poderoso sistema extensible para el procesamiento de mallas triangulares no estructuradas asistido por el usuario y QuoteMol, un sistema de visualización molecular de alta calidad.

La librería es extensa y entre sus principales funcionalidades podemos encontrar:

- Desrefinamiento de alta calidad.
- Estructuras eficientes para consultas.

- Relajamiento de mallas avanzado.
- Computación de curvatura.
- Optimización de coordenadas de texturas.
- Cálculo de la distancia de Hausdorff.
- Caminos geodésicos.
- Algoritmos de mallado de avance frontal.
- Algoritmos de refinamiento.

Concluimos el epígrafe con una comparación de las librerías reportadas. En la tabla 1 se muestra un estudio comparativo tomando en cuenta algunas características fundamentales en el campo de la generación de mallas.

De forma tal que se pueda reducir el esfuerzo de desarrollo para la generación de mallas, recomendamos el uso de librerías y Apis (Application Programming Interface) que proporcionen estructuras de datos eficientes y rutinas que son esenciales para el procesamiento geométrico de mallas tanto de triángulos como de tetraedros. Cada librería tiene su campo de aplicación y fortalezas únicas y así permanecerá en el próximo futuro.

| | CGAL | OpenMesh | MeshLibrary | MESH2D | VCGLib |
|---|---|------------------------|----------------------------|--------------------|------------------------------------|
| Generalidad | Mallas 2D y 3D | Mallas poligonales | Mallas 2D | Mallas 2D | Complejos Simpliciales |
| Estructura de datos | 2D y 3D Half Edge | Half Edge | 2D Half Edge | 2D QuadTree | Indezado con adyacencias |
| Triangulación Delaunay | 2D y 3D | NA | NA | 2D | NA |
| Triangulación Delaunay Restringida | Si | NA | NA | NA | NA |
| Algoritmos Convex Hull | 2D y 3D | NA | NA | NA | NA |
| Algoritmos de refinamiento | Catmull-Clark, Loop, Doo-Sabin and $\sqrt{3}$ | NA | NA | Basada en QuadTree | Subdivisión de mallas triangulares |
| Algoritmos de desrefinamiento | Lindstrom-Turk | QSlim, normal flipping | Desrefinamiento progresivo | Ninguno | Colapso de Arista |
| Lenguaje | C++ | C++ | C++ | Matlab | C++ |

Tabla 1 Comparación de las librerías: NA: No Aplicable.**Matlab como lenguaje de programación**

Para el desarrollo de la aplicación que permite evaluar la calidad de la malla resultante así como para la implementación de la estructura de datos y los algoritmos de refinamiento se utilizó como lenguaje de programación Matlab.

Las características fundamentales que conllevaron a seleccionar Matlab como lenguaje de programación se enumeran a continuación:

1. **Modularidad del lenguaje:** El lenguaje posee un conjunto de rutinas que facilitan la visualización 3D de mallas de tetraedros, implementación de la triangulación Delaunay, esto permite la simplicidad y fomenta la reutilización de código favoreciendo una reducción notable de líneas de código.
2. **Funciones matemáticas:** Debido a la naturaleza empírica y experimental del trabajo realizado, se hace necesario el empleo de métricas de la calidad o formas de medida de las mallas, de implementación compleja en lenguajes de programación estándar como C++ o Java, Matlab proporciona una librería llamada Geometry (Burkardt, 2004) que realiza cálculos geométricos en 2D, 3D y N dimensiones. Estos cálculos involucran ángulos, área, volumen, distancia y fórmulas de la calidad en 3D.
3. **Portabilidad:** La portabilidad de las implementaciones de los algoritmos facilita la traducción de código Matlab a por ejemplo, lenguaje C o Java.
4. **Documentación:** La herramienta matemática Matlab posee una ayuda especializada y detallada, carente en muchas librerías de generación de mallas abordadas anteriormente.

1.8 Sistemas de apoyo a la toma de decisiones.

En un sentido amplio, se define a los sistemas de apoyo a las decisiones como un conjunto de programas y herramientas que permiten obtener de manera oportuna la información que se requiere durante el proceso de la toma de decisiones que se desarrolla en un ambiente de incertidumbre (Cohen, 1999).

Suelen ser intensivos en cálculos y escasos en entradas y salidas de información. Así, por ejemplo, se genera una malla inicial que requiere poca información de entrada, esta se almacena en una estructura de datos en memoria, luego como resultado de aplicar un algoritmo de refinamiento sobre la misma, genera poca

información como resultado, pero puede realizar muchos cálculos durante su proceso como resultado de la propagación del refinamiento para mantener la conformidad en la malla. Estos sistemas suelen ser interactivos y amigables, con altos estándares de diseño gráfico y visual. Apoyan a la toma de decisiones que, por su misma naturaleza son repetitivas, tal como es el caso del refinamiento iterativo sobre una malla de tetraedros para determinar con cual algoritmo se refina con mejor calidad. Se debe proporcionar la mayor cantidad de información relevante en el menor tiempo posible, con el fin de decidir lo más adecuado. En este trabajo se propone desarrollar una aplicación en Matlab teniendo en cuenta estos principios para el apoyo a la toma de decisiones.

La investigación se encuentra enmarcada en el campo de la gestión de la información, propiciando con el sistema conocimientos nuevos acerca de los algoritmos implementados, flujos de información internos y externos aumentando la eficacia del proceso de refinamiento de mallas de tetraedros.

Con la ayuda de modelos matemáticos que permitan calificar un algoritmo de refinamiento, es posible analizar un gran número de acciones alternativas, llegar a conclusiones más precisas y obtener decisiones efectivas en un tiempo justo. De esta forma se aumenta la efectividad del proceso de toma de decisiones.

La efectividad y eficiencia del sistema son asesoradas utilizando indicadores de la calidad y el rendimiento de los algoritmos implementados.

Para medir la efectividad, que expresa el nivel de conformidad de un sistema dado a los objetivos para los que fue diseñado, los indicadores de rendimiento asociados están vinculados al flujo de salida del sistema (Vercellis, 2009) .

Para medir la eficiencia, que refleja la relación entre los flujos de entrada utilizados por el sistema y los correspondientes flujos de salida, las medidas de la eficiencia están asociadas con la calidad de las nuevas mallas generadas por los algoritmos de refinamiento.

De forma general, las métricas de efectividad indican cuando un algoritmo de refinamiento es apropiado para una malla de tetraedros o no, si se obtienen valores adecuados en la calidad o si degenera la malla, mientras que las métricas de eficiencia muestran cuando los algoritmos se ejecutan en el menor tiempo y de la mejor forma posible.

1.9 Conclusiones

En el presente capítulo se realizó un estudio de los fundamentos y tecnologías relacionados con la generación de mallas, se hace un recorrido por los algoritmos y estrategias de refinamiento existentes. Asimismo, se presentan algunas ideas básicas de diseño de estructuras de datos. Se establecieron las bases teóricas sobre las cuales está enfocada la aplicación que permitirá el apoyo a la toma de decisiones sobre cuál algoritmo refina una malla de tetraedros con mejor calidad.

Se valoraron las librerías disponibles para la implementación de la estructura de datos, así como de los algoritmos de refinamiento propuestos a estudiar empíricamente, describiendo las principales características, analizando ventajas y desventajas y los motivos por los cuales se emplea Matlab como herramienta de desarrollo.

CAPÍTULO 2 Estudio empírico de los algoritmos propuestos

2.1 Introducción

Debido a la no existencia de una herramienta para la generación de mallas de tetraedros que permita estudiar empíricamente las propiedades no degenerativas de los algoritmos de refinamiento tridimensionales basados en el lado mayor y delaunay. El objetivo objetivo es desarrollar una aplicación que permita apoyar la toma de decisiones en el refinamiento con una interface gráfica amigable que puede ser descrita en unos pocos miles de líneas de código Matlab. Se pueden ofrecer implementaciones más optimizadas y el estudio de los algoritmos de refinamiento local pero la meta es asistir el análisis de los algoritmos propuestos para las mallas de tetraedros.

El software puede ser considerado como un prototipo de primera versión pero efectivo para los usuarios que deseen aplicar algunos de estos algoritmos a un campo concreto de aplicación de mallas 3D. Se describen los algoritmos, se presentan ejemplos de mallas refinadas a través de la aplicación Matlab, ver anexos. El software construye un gráfico de la evolución de la calidad para cada iteración del algoritmo estudiado. Finalmente se proponen algunas mejoras futuras para el software de mallas.

La primera tarea para implementar un software de mallas 3D lo constituye el diseño de una estructura de datos adecuada para almacenar las mallas de tetraedros. Es deseable que tales estructuras de datos cumplan determinados requisitos como ser eficientes en términos de tiempo y espacio, de forma tal que los elementos de la malla y la información de vecindad pueda ser consultada y modificada sin tener que realizar una búsqueda global, mientras se requiere un mínimo espacio de almacenamiento. Otro requisito a tener en cuenta es la simplicidad a la hora de implementar una de las estructuras de datos propuestas anteriormente en lenguajes de programación como C, C++, Java o Matlab.

Debido a que Matlab es un lenguaje orientado a matrices, se escoge la popular estructura de datos utilizada en computación numérica, la estándar lista de conectividad Elemento-Vértice empleada en elementos finitos (E.B.Becker,

G.F.Carey, & Oden, 1981). Esta estructura de datos requiere un mínimo espacio en memoria debido a que no permite almacenar puntos duplicados ni información de adyacencia. Su implementación es relativamente sencilla debido a que las tablas son soportadas como matrices en Matlab. El post-procesamiento de una malla de tetraedros es un tanto complejo, pero las rutinas de visualización de Matlab hacen este procedimiento relativamente sencillo también.

La aplicación cuenta con los siguientes algoritmos de refinamiento:

Basados en el lado mayor:

1. Bisección por el lado mayor (LE), refinamiento global y local.
2. 3-Tetraedros por el lado mayor (3T-LE), refinamiento global y local.
3. Trisección por el lado mayor, refinamiento global y local.
4. Cuatrisección por el lado mayor, refinamiento global y local.
5. 4-Tetraedros por el lado mayor (4T-LE), refinamiento global y local.
6. 8-Tetraedros por el lado mayor (8T-LE), refinamiento global y local.

Basados en Delaunay:

1. Inserción de puntos centroide con Delaunay, refinamiento global.
2. Inserción de puntos medios en las aristas con Delaunay, refinamiento global.
3. Inserción de puntos circunescfera con Delaunay, refinamiento global.
4. Inserción de puntos inescfera con Delaunay, refinamiento global.
5. Inserción de puntos por el lado mayor con Delaunay, refinamiento global.
6. Inserción de puntos por la trisección del lado mayor con Delaunay, refinamiento global.
7. Inserción de puntos por la cuatrisección del lado mayor con Delaunay, refinamiento global.

Otros:

1. 4-Tetraedros por el baricentro, refinamiento global y local.

Debe de notarse que la partición de 3-Tetraedros por el lado mayor es la extensión a tres dimensiones del algoritmo original de 3-Triángulos por el lado mayor (Maria Cecilia Rivara & Iribarren, 1996). También debe ser mencionado que en el campo del refinamiento se le ha prestado menor atención a subdivisiones basadas en la inserción de 2 ó 3 puntos equidistantes en la arista mayor de los tetraedros (Marquez, et al., 2010) que se le conoce como trisección o cuatrisección de la arista mayor, particiones estudiadas en esta tesis. En la implementación de la partición 8T-LE, se hace necesario considerar una partición intermedia 4T-LE (A. Plaza & Rivara,

2003). La partición 8T-LE es considerada la última metodología para particiones por el lado mayor (Balman, 2006). En cada caso fue realizado un estudio cuidadoso de los posibles patrones por inserción de puntos producidos por estas particiones.

Para la implementación de estos algoritmos de refinamiento por el lado mayor, la inserción de cada nuevo punto solo significa la partición del tetraedro seleccionado y el refinamiento de algunos tetraedros vecinos debido a la propagación que asegura la conformidad de la malla refinada.

Para el caso de los métodos basados en Delaunay, nuestros algoritmos refinan la malla insertando puntos y utilizando el algoritmo Delaunay para el refinamiento automático. Se obtiene rendimiento en los algoritmos utilizando el algoritmo Delaunay de CGAL (Rineau, 2010) en MatlabR2010a.

La partición de 4-Tetraedros por el baricentro inserta un punto en el baricentro del tetraedro y lo une con los 3 vértices que forman cada cara del tetraedro.

Para analizar la calidad de las formas de los nuevos tetraedros generados a través de las particiones, un tetraedro de cada tipo de forma es tomado para los experimentos, también un cubo, una malla delaunay de muestra y una malla tomada de una aplicación real, ver Figura 16, la amplia variedad de formas de los tetraedros estudiados permite conjeturar sobre las propiedades no degenerativas de algunos de estos algoritmos.

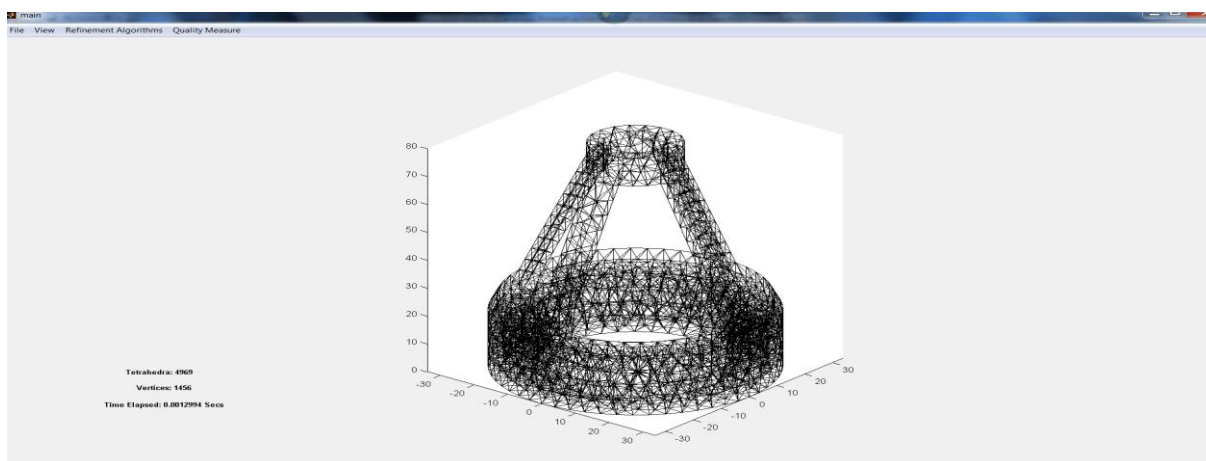


Figura 16 Malla de tetraedros de ejemplo.

Para medir la calidad o degeneración se utilizaron las siguientes medidas disponibles en el paquete Geometry (Burkardt, 2004) : Etha, Whiteh, Ratio y Solid Angle.

Luego de cada iteración de los algoritmos, se realiza la operación del cálculo de la media de la calidad sobre la estructura de datos permitiendo la construcción de un gráfico de evolución de la calidad, ver Figura 17.

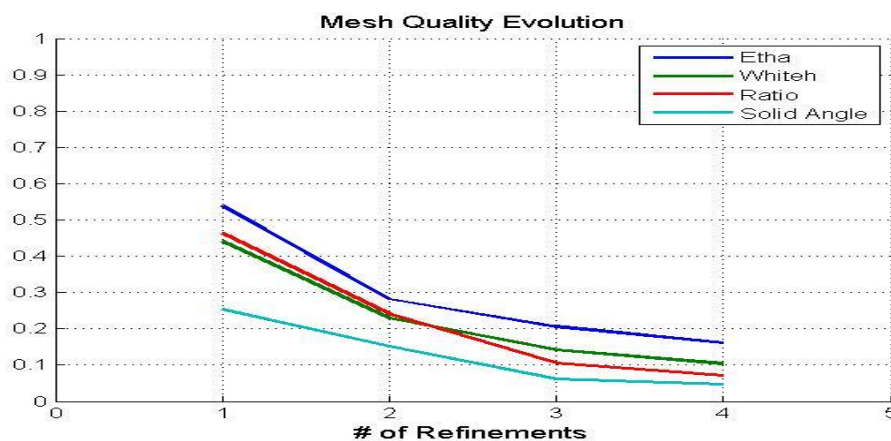


Figura 17 Gráfico de la evolución de la calidad para 4 iteraciones de refinamiento.

Debe ser notado que, a medida que las iteraciones de refinamiento proceden, también el uso de memoria por parte de la estructura de datos aumenta debido a que el costo de estos algoritmos es de complejidad lineal (M.-C. Rivara & Venere, 1994).

2.2 Algoritmo LE

El primer algoritmo a estudiar empíricamente es el conocido como bisección simple por el lado mayor, consiste en la inserción de un punto medio por el lado mayor de un tetraedro para bisecarlo en dos nuevos, ver figura 18. Fue desarrollado por Rivara y Levin en (Maria Cecilia Rivara & Levin, 1992), donde se muestra un estudio empírico de la reducción del ángulo sólido debido a repetidas subdivisiones, no obstante no se han obtenido resultados matemáticos que garanticen que los nuevos tetraedros construidos no degeneran cuando el proceso procede a infinito. Para conjeturar sobre las propiedades no degenerativas de este algoritmo de refinamiento, un tetraedro de cada tipo de forma es tomado para los experimentos, ver figura 19 (a-e) y anexo #1. Se estudia el comportamiento de las medidas de la calidad mencionadas anteriormente cuando la aplicación iterativa del algoritmo procede. Debe ser notado que la aplicación global del algoritmo sobre la malla necesita de refinamientos adicionales de forma tal que se obtenga una malla conforme.

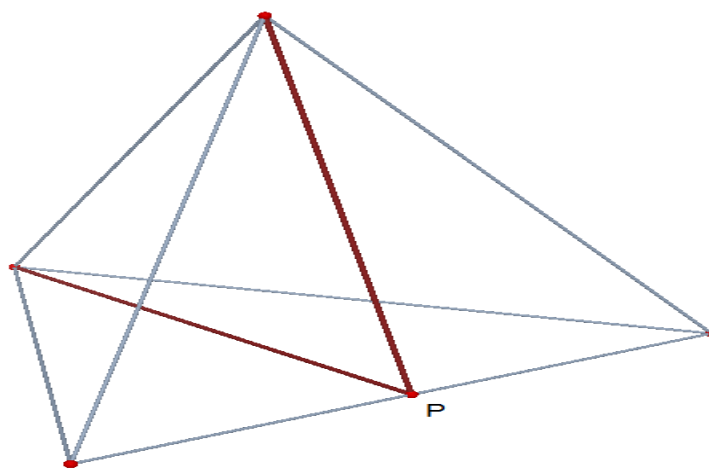


Figura 18 Bisección por el lado mayor de un tetraedro.

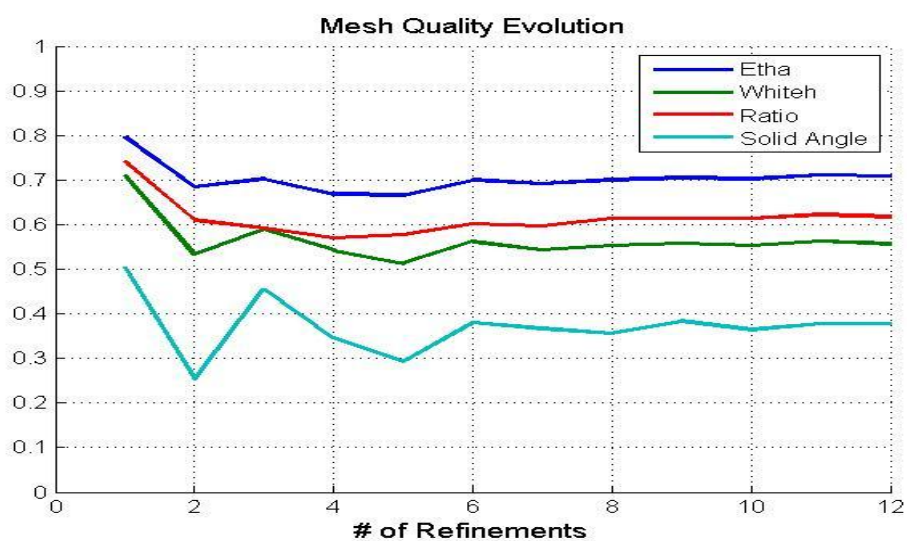


Figura 19 (a) LE uniforme en un tetraedro regular.

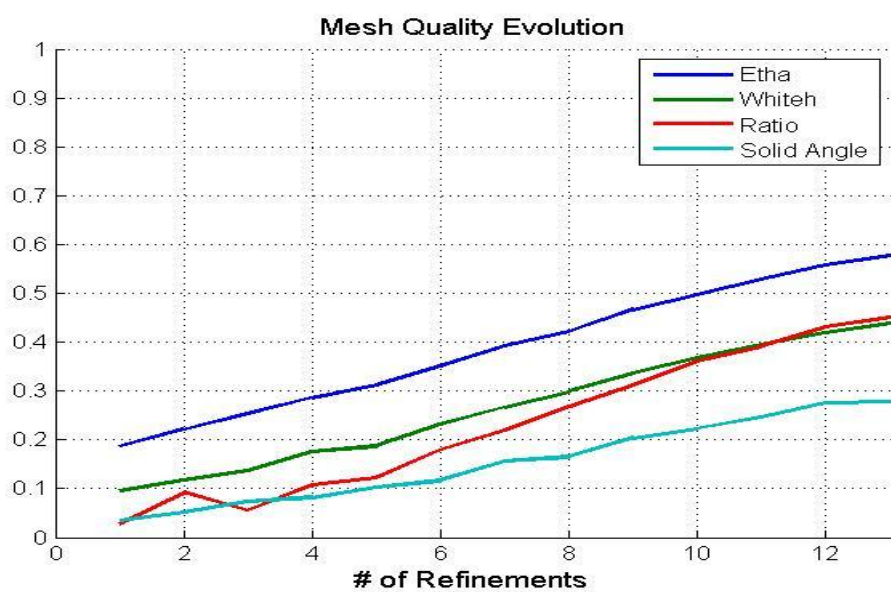


Figura 19 (b) LE uniforme en un tetraedro cap.

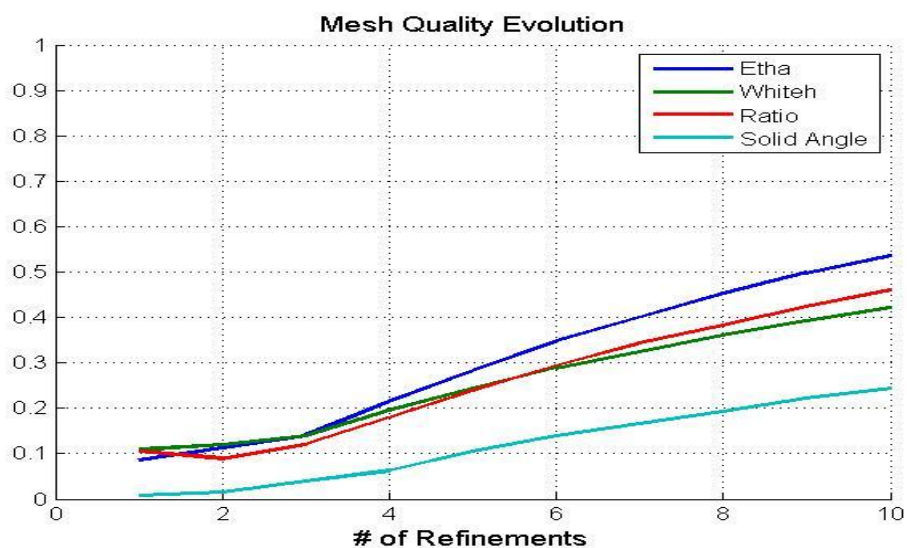


Figura 19 (c) LE uniforme en un tetraedro needle.

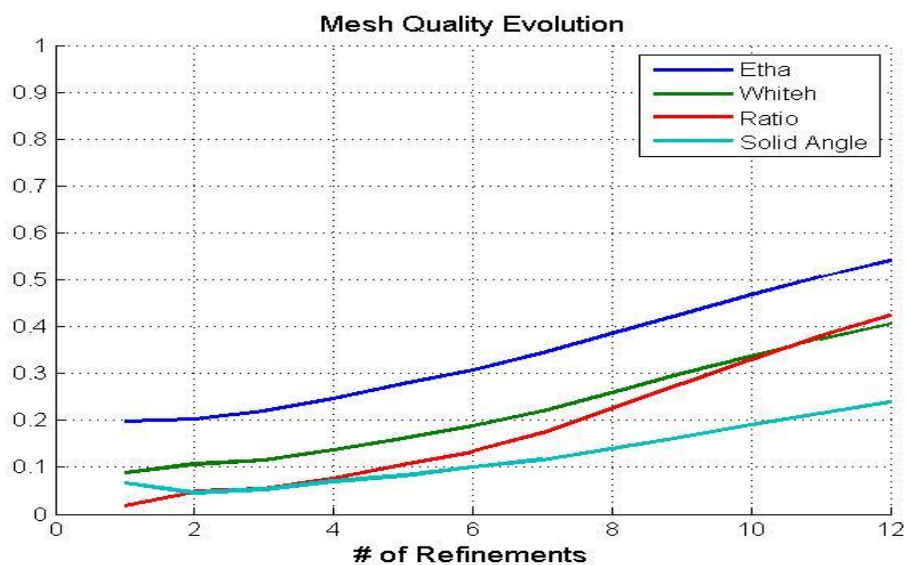


Figura 19 (d) LE uniforme en un tetraedro sliver.

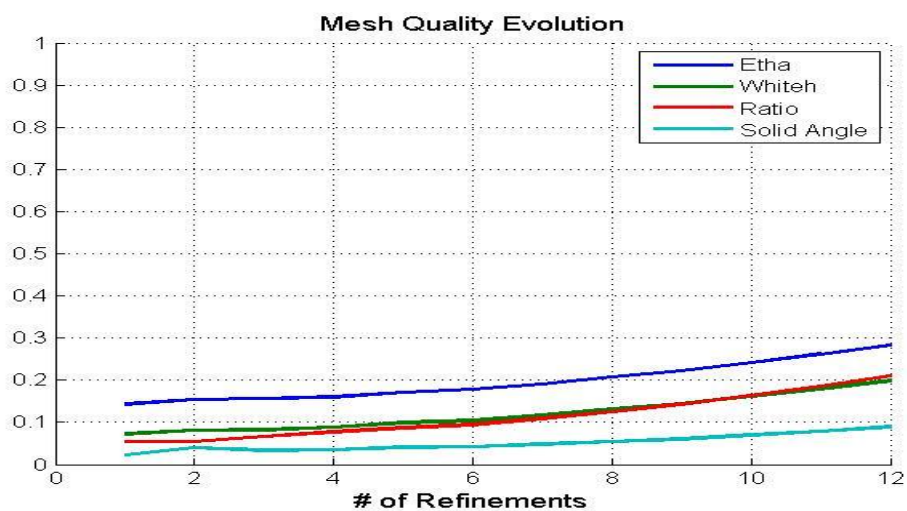


Figura 19 (e) LE uniforme en un tetraedro wedge.

Se realizaron de 10 a 12 niveles de refinamiento globales sobre una malla inicial que contiene un tipo de tetraedro diferente para cada caso hasta llegar a un millón de elementos. La Figura 19 muestra la evolución de las diferentes medidas de la calidad para los tetraedros obtenidos durante las cinco pruebas realizadas anteriormente. De los resultados reportados, en la Figura 19 (b),(c) y (d) puede ser notado la propiedad de auto mejora de la calidad cuando el algoritmo es iterativamente aplicado. Debe ser mencionado que el peor caso menos favorable para la degeneración corresponde al tetraedro wedge como malla inicial.

Para cada tetraedro de prueba utilizado en esta tesis, la Tabla 2 ha sido producida en términos de las coordenadas de sus cuatro vértices.

| Coordenadas de los Tetraedros | X | Y | Z |
|--------------------------------------|----------|----------|----------|
| | 0 | 0 | 0 |
| Tetraedro Regular | 1.7321 | 1 | 2.8284 |
| | 1.7321 | 3 | 0 |
| | 3.4641 | 0 | 0 |
| | 0 | 0 | 0 |
| Tetraedro Cap | 1.7321 | 1 | 0.1414 |
| | 1.7321 | 3 | 0 |
| | 3.4641 | 0 | 0 |
| | 0 | 0 | 0 |
| Tetraedro Needle | 1.7321 | 1 | 56.5685 |
| | 1.7321 | 3 | 0 |
| | 3.4641 | 0 | 0 |
| | 0 | 0 | 0 |
| Tetraedro Sliver | 1.7321 | -1.8197 | 0.2219 |
| | 1.7321 | 3 | 0 |
| | 3.4641 | 0 | 0 |
| | 0 | 0 | 0 |
| Tetraedro Wedge | 0.0866 | 0.0500 | 0.1414 |
| | 1.7321 | 3 | 0 |
| | 3.4641 | 0 | 0 |

Tabla 2 Coordenadas de los tetraedros.

2.3 Algoritmo 3T-LE

De manera que se comprendan las ideas que sustentan el algoritmo 3T-LE en 3D, se describe brevemente su análogo en 2D y la partición de triángulo asociada.

Definición 4.

La partición de los 3-triángulos por el lado mayor de cualquier triángulo t_0 (de lados $r_1 \geq r_2 \geq r_3$ y ángulos $\gamma_0 \geq \beta_0 \geq \alpha_0$) es obtenida uniendo el punto medio del lado mayor r_1 con el vértice opuesto y el punto medio del lado r_2 (para una ilustración, ver Figura 20).

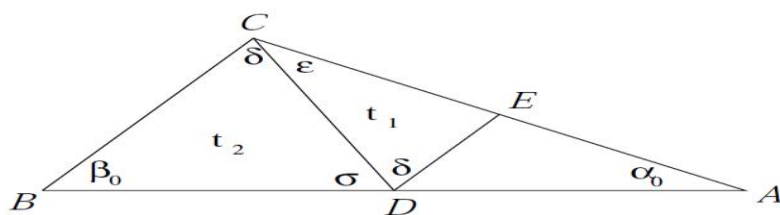


Figura 20 Partición 3T-LE de un triángulo.

La partición 3T de un triángulo t en que la segunda arista mayor de t es también subdividida se denominará 3T-LE, Figura 20.

El algoritmo de refinamiento de los tres triángulos utiliza la partición de los tres triángulos por el lado mayor para mejorar los triángulos obtusos, un teorema sobre este planteamiento puede ser encontrado en la referencia (Maria Cecilia Rivara & Iribarren, 1996). Debe ser mencionado que la partición 3T-LE, ya ha sido empleada en los algoritmos de refinamiento por el lado mayor como un patrón para realizar refinamiento por propagación en los algoritmos de refinamiento local de las triangulaciones, por ejemplo, en la bisección simple pura (Maria Cecilia Rivara, 1997), en la partición de los 4-Triángulos por el lado mayor (Á. Plaza, et al., 2004; Maria Cecilia Rivara & Iribarren, 1996) y la partición similar (Padrón, Suárez, & Plaza, 2007).

El algoritmo consiste en dos pasos básicos:

1. Refinamiento de los triángulos seleccionados uniendo el punto medio del lado mayor del triángulo con el vértice opuesto y el punto medio de la segunda arista mayor del triángulo.
2. Propagación local del refinamiento para garantizar una malla conforme debido a que el refinamiento local de un triángulo dado implica el refinamiento del propio triángulo y de algunos de sus vecinos.

Los conceptos involucrados en 2D se generalizan directamente para la tercera dimensión.

Utilizando los conceptos precedentes el algoritmo puede ser descrito esquemáticamente como sigue:

T: Conjunto de tetraedros a ser refinados.

t: malla de tetraedros.

Algoritmo 3-Tetraedros por el lado mayor (T,t)

1. Encontrar arista mayor primaria y secundaria de los tetraedros en T.
2. Insertar un punto medio en cada arista mayor primaria y secundaria.

3. Particionar cada tetraedro de T en tres subtetraedros de acuerdo a los puntos medios insertados.
4. Garantizar conformidad en la malla t.

Definición 5.

Para cualquier tetraedro T de arista mayor y arista mayor secundaria única, la partición 3T-LE (3-Tetraedros por el lado mayor) de T se define como:

1. Bisección de T por el punto medio de la arista mayor única.
2. Bisección de T por el punto medio de la arista mayor secundaria única.
3. Aplicar patrón de partición apropiado.
4. Garantizar conformidad en la malla.

Los primeros 3 pasos en el algoritmo de refinamiento 3D pueden ser computados en tiempo constante, debido a que el mayor esfuerzo de computación lo constituye la propagación del refinamiento para garantizar conformidad. La propagación es una tarea necesaria para evitar puntos no conformes, requisito que debe existir en una malla, o sea que la intersección de tetraedros no disjuntos es un vértice común, arista o cara, también para generar mallas graduales y suaves. La estrategia fundamental para implementar la propagación está basado en la propagación por el lado mayor, que consiste en repetir el refinamiento en elementos vecinos siguiendo el camino de propagación por el lado mayor (LEPP Longest-Edge Propagation Path) (Maria Cecilia Rivara & Levin, 1992) mientras existan puntos no conformes.

Varios algoritmos han sido propuestos en la literatura para garantizar la conformidad de la malla, ver referencias (A. Plaza & Carey, 2000; Maria Cecilia Rivara, 1997; Maria Cecilia Rivara & Levin, 1992; Suarez, Abad, Plaza, & Padron, 2005). En esta investigación se utiliza un algoritmo para garantizar la conformidad en la malla basado en el concepto de aristas no conformes que puede ser visto como una versión mejorada del algoritmo propuesto en (Suarez, et al., 2005). Calcular el LEPP en 3D es una tarea muy costosa, no obstante el algoritmo encuentra en tiempo real el LEPP asociado a cada arista mayor no conforme.

La partición 3T-LE está basada en el estudio de las posibles posiciones relativas de la arista mayor de T y la arista mayor secundaria de T. Claramente, solo existen tres configuraciones relativas de patrones posibles, invariantes bajo traslación, rotación, reflexión y escalado uniforme enunciados a continuación:

1. Arista mayor de T opuesta a la única arista mayor secundaria (Figura 21(a)) .
2. Arista mayor comparte un vértice con la arista mayor secundaria (Figura 21(b y c)).

En el caso del patrón 1, la partición de los tres tetraedros produce una triangulación volumétrica no conforme debido a la distribución de la arista mayor y la arista mayor secundaria.

La partición 3T-LE produce tres tetraedros mostrados en la Figura 22.

El estudio cuidadoso de los posibles patrones de particiones que surgen al insertar puntos producido por las diferentes posiciones relativas de la arista mayor y la arista mayor secundaria de T permite obtener tres patrones de particiones parciales involucrados en el algoritmo de refinamiento, ver Figura 21.

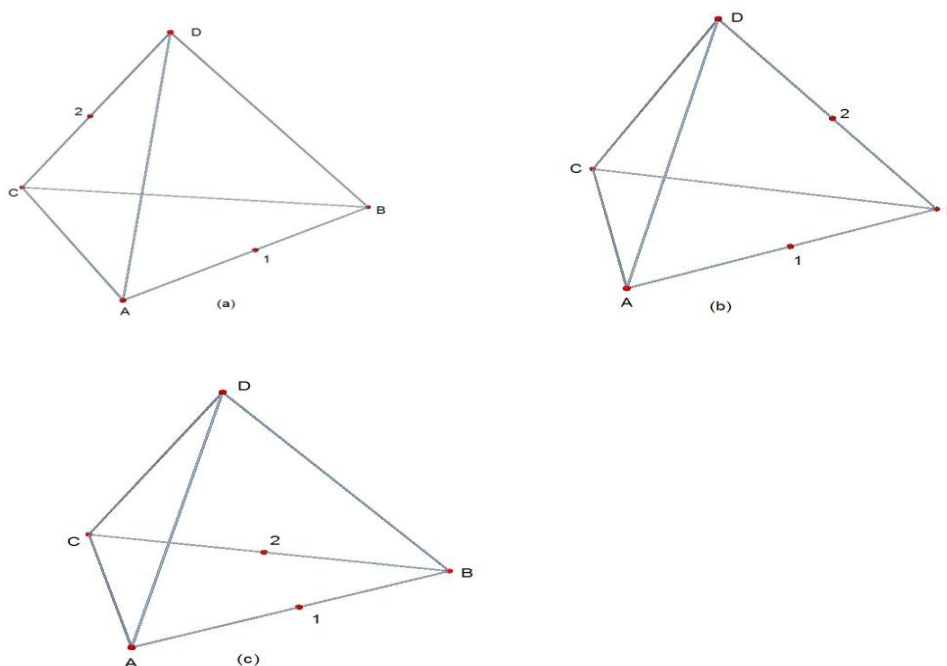


Figura 21 Posición relativa de la arista mayor (numerada 1) y arista mayor secundaria (numerada 2) de T.

A continuación aparece un estudio numérico que muestra el comportamiento práctico de la partición 3T-LE en comparación con la bisección pura por el lado mayor, LE y la trisección por el lado mayor cuando se aplica a diferentes formas de tetraedros iniciales. Luego analizamos la calidad de los nuevos tetraedros generados. La formula estándar E_{th} es utilizada para medir la calidad.

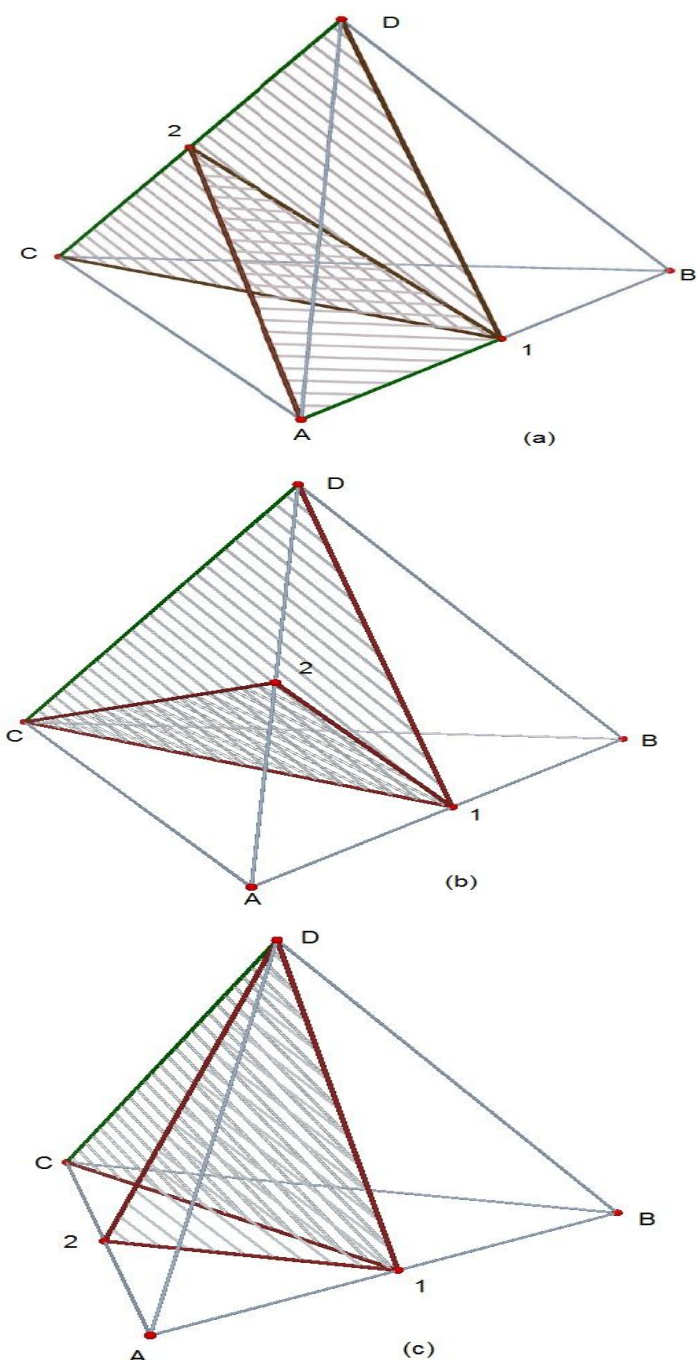


Figura 22 Partición de los 3-Tetraedros obtenidos de acuerdo a la posición relativa de la arista mayor (numerada 1) y la arista mayor secundaria (numerada 2) de T.

La tabla en la Figura 23 provee la media y el mínimo de Etha respectivamente. Un tetraedro de cada tipo de forma es tomado para los experimentos. En estos casos de prueba se realizan dos niveles de refinamiento uniforme por los algoritmos 3T-LE, LE y LE-Trisección a cada tetraedro inicial. Cada partición por el lado mayor consiste en la subdivisión de todos los tetraedros de la malla actual, seguida por la subdivisión de algunos tetraedros no conformes hasta que se obtiene una malla conforme.

| Regular | Media etha | | | Min etha | | |
|----------------|-------------------|--------|---------------|-----------------|--------|---------------|
| | 3T-LE | LE | LE-Trisección | 3T-LE | LE | LE-Trisección |
| i=0 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| i=1 | 0.7211 | 0.7957 | 0.6398 | 0.6350 | 0.7957 | 0.6181 |
| i=2 | 0.6961 | 0.6838 | 0.4033 | 0.4535 | 0.6350 | 0.2308 |
| | | | | | | |
| Needle | Media etha | | | Min etha | | |
| | 3T-LE | LE | LE-Trisección | 3T-LE | LE | LE-Trisección |
| i=0 | 0.0550 | 0.0550 | 0.0550 | 0.0550 | 0.0550 | 0.0550 |
| i=1 | 0.0781 | 0.0871 | 0.1062 | 0.0328 | 0.0378 | 0.0265 |
| i=2 | 0.1067 | 0.1140 | 0.1316 | 0.0328 | 0.0328 | 0.0181 |
| | | | | | | |
| Sliver | Media etha | | | Min etha | | |
| | 3T-LE | LE | LE-Trisección | 3T-LE | LE | LE-Trisección |
| i=0 | 0.1833 | 0.1833 | 0.1833 | 0.1833 | 0.1833 | 0.1833 |
| i=1 | 0.1786 | 0.1972 | 0.1759 | 0.1506 | 0.1713 | 0.1309 |
| i=2 | 0.2205 | 0.2028 | 0.1683 | 0.1833 | 0.1506 | 0.0627 |
| | | | | | | |
| Cap | Media etha | | | Min etha | | |
| | 3T-LE | LE | LE-Trisección | 3T-LE | LE | LE-Trisección |
| i=0 | 0.2033 | 0.2033 | 0.2033 | 0.2033 | 0.2033 | 0.2033 |
| i=1 | 0.2093 | 0.1862 | 0.1563 | 0.1841 | 0.1862 | 0.1465 |
| i=2 | 0.2609 | 0.2204 | 0.1873 | 0.1862 | 0.1841 | 0.1056 |
| | | | | | | |
| Wedge | Media etha | | | Min etha | | |
| | 3T-LE | LE | LE-Trisección | 3T-LE | LE | LE-Trisección |
| i=0 | 0.1659 | 0.1659 | 0.1659 | 0.1659 | 0.1659 | 0.1659 |
| i=1 | 0.1290 | 0.1427 | 0.1171 | 0.1032 | 0.1241 | 0.0930 |
| i=2 | 0.1590 | 0.1538 | 0.1082 | 0.1227 | 0.1032 | 0.0408 |

Figura 23 Resultados reportados de los algoritmos 3T-LE, LE y LE-Trisección.

Como principal conclusión de la comparación de los resultados obtenidos se concluye que el algoritmo 3T-LE es superior a la LE y a la LE-Trisección en cuanto a la calidad obtenida a medida que el refinamiento iterativo procede y se hace notable una mejora de la calidad para el tetraedro tipo sliver y cap, ver Anexo #2.

2.4 Algoritmo LE-Trisección

Menor atención ha sido prestada a subdivisiones basadas en la inserción de dos puntos equidistantes en la arista mayor de un tetraedro, conocido también como trisección de la arista mayor. Debe ser mencionado que no se han obtenido resultados matemáticos que garanticen la propiedad no degenerativa de la trisección 3D.

En este trabajo primero se introduce una extensión de la bisección por el lado mayor, también basada en la división de la arista mayor de un tetraedro, pero ahora en tres partes. Luego, siguiendo un argumento similar a la referencia (A. Plaza, Suárez, & Padrón, 2005), el trabajo se focaliza en la propiedad no degenerativa de la

partición (3T-LE) 3-Tetraedros por la trisección de la arista mayor, que constituye la extensión natural a tres dimensiones de la trisección por el lado mayor de los triángulos introducida por Plaza y coautores (Á. Plaza, Falcón, & Suárez, 2010).

Se diseñó una serie de experimentos numéricos que nos permiten conjeturar sobre la propiedad no degenerativa de la trisección por el lado mayor en 3D. Además, el comportamiento de la no degeneración de la partición 3T-LE es similar a otras particiones usadas en la literatura (A. Plaza & Rivara, 2003; A. Plaza, et al., 2005; Maria Cecilia Rivara & Levin, 1992).

Definición 6.

Sea t un tetraedro de nodos $ABCD$ con arista mayor AB . La trisección por el lado mayor de un tetraedro t es obtenida uniendo dos puntos equidistantes de la arista mayor de t , dígase M_1 y M_2 , con los vértices opuestos C y D (ver Figura 24).

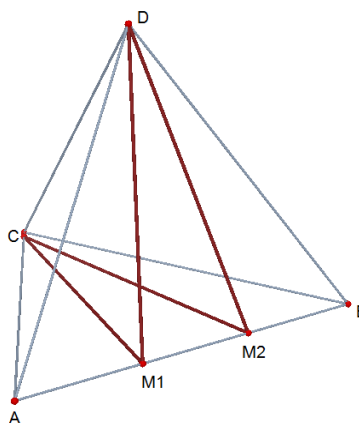


Figura 24 :Trisección por el lado mayor de un tetraedro.

La Figura 24 muestra la trisección por el lado mayor de un tetraedro $ABCD$ con arista mayor AB , donde dos puntos equidistantes en la arista mayor de AB son calculados como sigue:

$$M_1 = A + \frac{1}{3} \overrightarrow{AB} = \frac{2A+B}{3} \text{ y } M_2 = A + \frac{2}{3} \overrightarrow{AB} = \frac{A+2B}{3}$$

Se muestra aquí un estudio numérico que evidencia el comportamiento práctico de la partición LE-Trisección en comparación con el algoritmo de refinamiento basado en la cuatrisección de la arista mayor cuando se aplican ambos algoritmos a tetraedros iniciales de diferentes formas. Luego analizamos la calidad de los nuevos tetraedros generados. La formula estándar Etha es utilizada para medir la calidad. La tabla en la Figura 25 proporciona los valores de la media y el mínimo de Etha respectivamente. Un tetraedro de cada forma es empleado para los experimentos. En estas pruebas, cinco pasos de refinamientos globales por la LE-Trisección y LE-Cuatrisección son realizados a cada tetraedro inicial. Cada partición por el lado

mayor consiste en la subdivisión de todos los tetraedros de la malla, seguido por la subdivisión de los tetraedros no conformes hasta que se obtiene una malla conforme, esto explica el número distinto de tetraedros generados en cada columna de la tabla de la Figura 25. La Figura 26 muestra solo algunas de las mallas refinadas más relevantes de los experimentos.

Como principal observación de los resultados, la LE-Trisección es superior a la LE-Cuatrisección en todos los casos probados, se presenta un estudio más detallado sobre la propiedad no degenerativa de la trisección 3D.

Se procedió a estudiar mallas obtenidas de las diferentes formas de los tetraedros, la amplia variedad de formas de los tetraedros permite conjeturar sobre la propiedad no degenerativa. El objetivo es obtener un coeficiente experimental de la no degeneración de la partición de los 3-Tetraedros por la trisección del lado mayor y compararlo con la cuatrisección por el lado mayor, de manera tal que se corrobore la siguiente hipótesis, que la inserción de un gran número de puntos por el lado mayor produce degeneración de la triangulación (Marquez, et al., 2010).

La Figura 27 muestra la evolución de la media de las diferentes formas de medida para los tetraedros obtenidos durante las 5 pruebas realizadas. De los resultados reportados, la Figura 27 (b) y (c), puede ser notado la propiedad de auto mejora de la calidad de la trisección por el lado mayor cuando se aplica iterativamente al tetraedro tipo gorra (cap) o aguja (needle). Se muestra evidencia experimental para la fórmula de la calidad Etha, (η), la convergencia de no degeneración a un valor positivo fijo es garantizado para cualquier tetraedro t_i^n en τ^n , $\eta \geq 1$.

$$\eta(t_i^n) \geq c\eta(t),$$

donde c es constante e independiente de i y n . Se realizaron 13 refinamientos globales LE-Trisección sobre una malla inicial que contiene un tetraedro diferente en cada caso. El proceso de refinamiento para la partición LE-Trisección termina con el nivel de malla τ^{13} conteniendo $3^{13}=1, 594, 323$ elementos, mientras que para la cuatrisección la malla termina con nivel τ^{10} .

| Regular | Media etha | | | Min etha | | |
|---------|--------------------|---------------|------------|-----------------|---------------|------------|
| | NTET Cuatrisección | Cuatrisección | Trisección | NTET Trisección | Cuatrisección | Trisección |
| i=0 | 1 | 1.00 | 1.00 | 1 | 1.00 | 1.00 |
| i=1 | 4 | 0.5383 | 0.6398 | 3 | 0.5080 | 0.6181 |
| i=2 | 22 | 0.2822 | 0.4033 | 13 | 0.1176 | 0.2308 |
| i=3 | 160 | 0.2064 | 0.3868 | 65 | 0.0187 | 0.1023 |
| i=4 | 1549 | 0.1616 | 0.3304 | 465 | 0.0036 | 0.0494 |
| i=5 | 17854 | 0.1252 | 0.3211 | 3989 | 0.0006 | 0.0285 |
| | | | | | | |
| Needle | Media etha | | | Min etha | | |
| | NTET Cuatrisección | Cuatrisección | Trisección | NTET Trisección | Cuatrisección | Trisección |
| i=0 | 1 | 0.0550 | 0.0550 | 1 | 0.0550 | 0.0550 |
| i=1 | 4 | 0.1187 | 0.1062 | 3 | 0.0206 | 0.0265 |
| i=2 | 25 | 0.1227 | 0.1316 | 13 | 0.0078 | 0.0181 |
| i=3 | 202 | 0.1008 | 0.1382 | 79 | 0.0019 | 0.0098 |
| i=4 | 1567 | 0.1119 | 0.2078 | 537 | 7.64E-04 | 0.0087 |
| i=5 | 13789 | 0.1081 | 0.2364 | 4245 | 3.22E-04 | 0.0087 |
| | | | | | | |
| Sliver | Media etha | | | Min etha | | |
| | NTET Cuatrisección | Cuatrisección | Trisección | NTET Trisección | Cuatrisección | Trisección |
| i=0 | 1 | 0.1833 | 0.1833 | 1 | 0.1833 | 0.1833 |
| i=1 | 4 | 0.1546 | 0.1759 | 3 | 0.1049 | 0.1309 |
| i=2 | 19 | 0.1477 | 0.1683 | 11 | 0.0287 | 0.0627 |
| i=3 | 142 | 0.1055 | 0.1616 | 51 | 0.0063 | 0.0569 |
| i=4 | 1258 | 0.0944 | 0.1732 | 283 | 0.0014 | 0.0389 |
| i=5 | 11998 | 0.0782 | 0.1769 | 1659 | 5.93E-04 | 0.0250 |
| | | | | | | |
| Cap | Media etha | | | Min etha | | |
| | NTET Cuatrisección | Cuatrisección | Trisección | NTET Trisección | Cuatrisección | Trisección |
| i=0 | 1 | 0.2033 | 0.2033 | 1 | 0.2033 | 0.2033 |
| i=1 | 4 | 0.1337 | 0.1563 | 3 | 0.1200 | 0.1465 |
| i=2 | 22 | 0.1398 | 0.1873 | 11 | 0.0501 | 0.1056 |
| i=3 | 181 | 0.1347 | 0.2238 | 89 | 0.0138 | 0.0840 |
| i=4 | 1573 | 0.1122 | 0.2453 | 481 | 0.0034 | 0.0380 |
| i=5 | 17884 | 0.0936 | 0.2606 | 2907 | 7.18E-04 | 0.0289 |
| | | | | | | |
| Wedge | Media etha | | | Min etha | | |
| | NTET Cuatrisección | Cuatrisección | Trisección | NTET Trisección | Cuatrisección | Trisección |
| i=0 | 1 | 0.1659 | 0.1659 | 1 | 0.1659 | 0.1659 |
| i=1 | 4 | 0.0993 | 0.1171 | 3 | 0.0750 | 0.0930 |
| i=2 | 22 | 0.0875 | 0.1082 | 13 | 0.0194 | 0.0408 |
| i=3 | 235 | 0.0595 | 0.0954 | 69 | 0.0040 | 0.0246 |
| i=4 | 2374 | 0.0485 | 0.0886 | 441 | 5.45E-04 | 0.0105 |
| i=5 | 26278 | 0.0397 | 0.0885 | 3293 | 5.60E-05 | 0.0073 |

Figura 25 :Resultados reportados de la LE-Cuatrisección y de la LE-Trisección 3D.

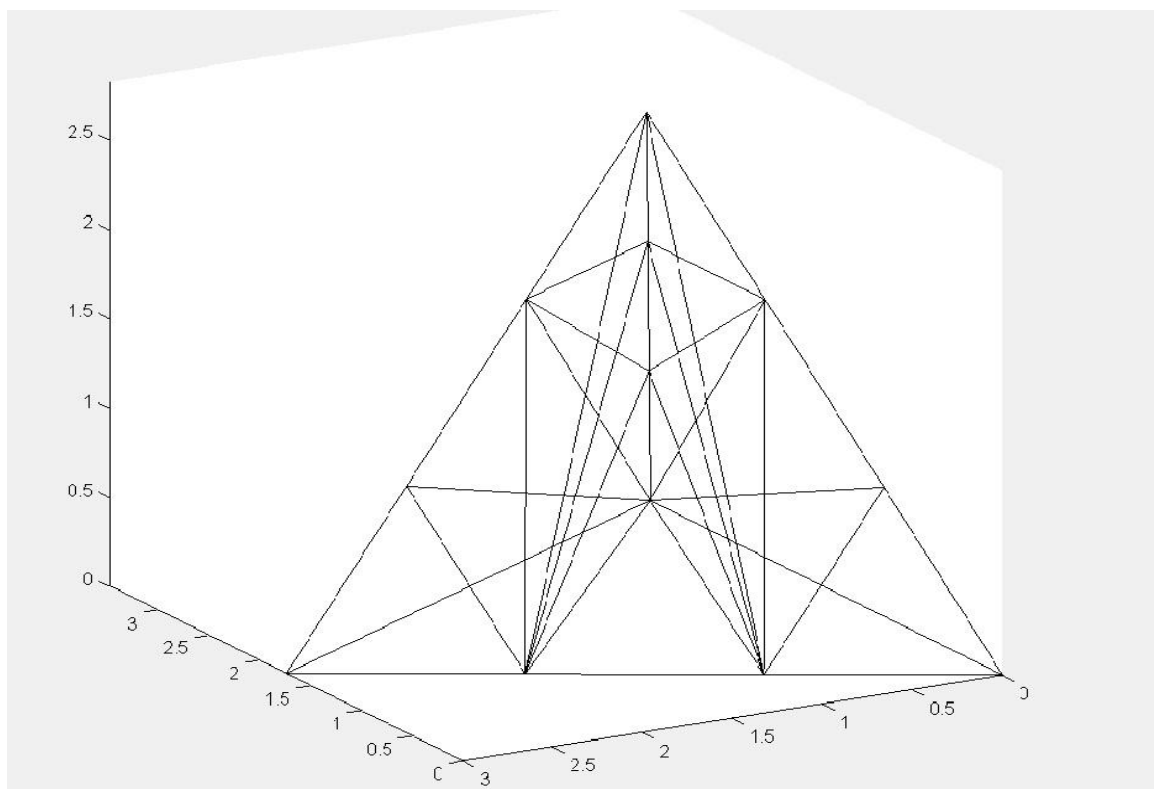


Figura 26 (a) : Nivel 2 de refinamiento uniforme de la LE-Trisección en un tetraedro regular.

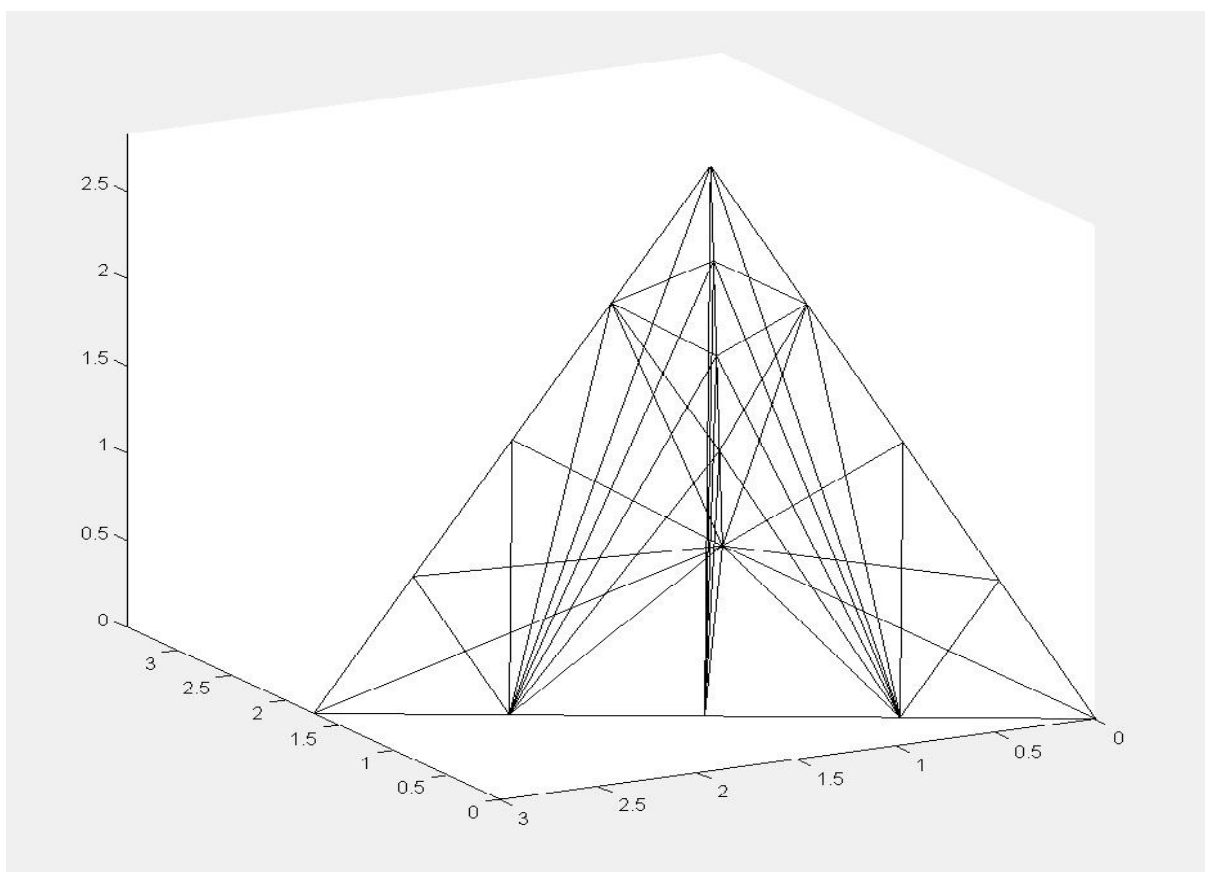


Figura 26 (b) : Nivel 2 de refinamiento uniforme de la LE-Cuatrisección en un tetraedro regular.

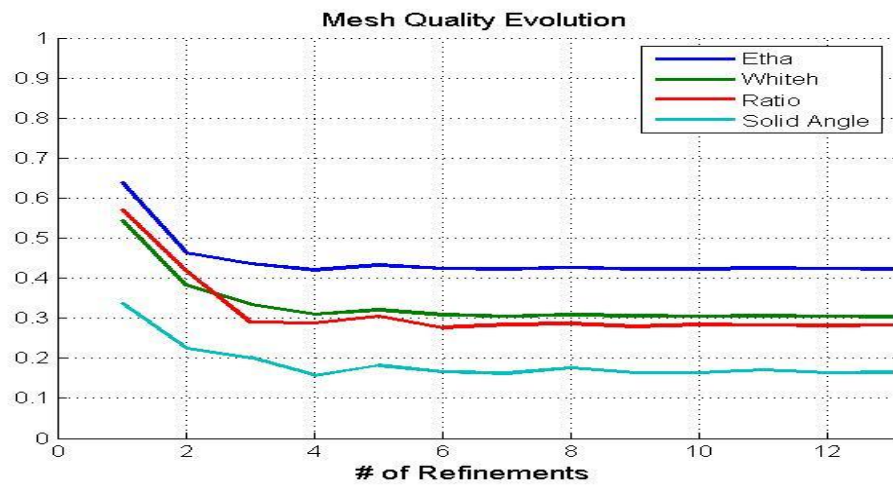


Figura 27 (a) : LE-Trisección uniforme en un tetraedro regular.

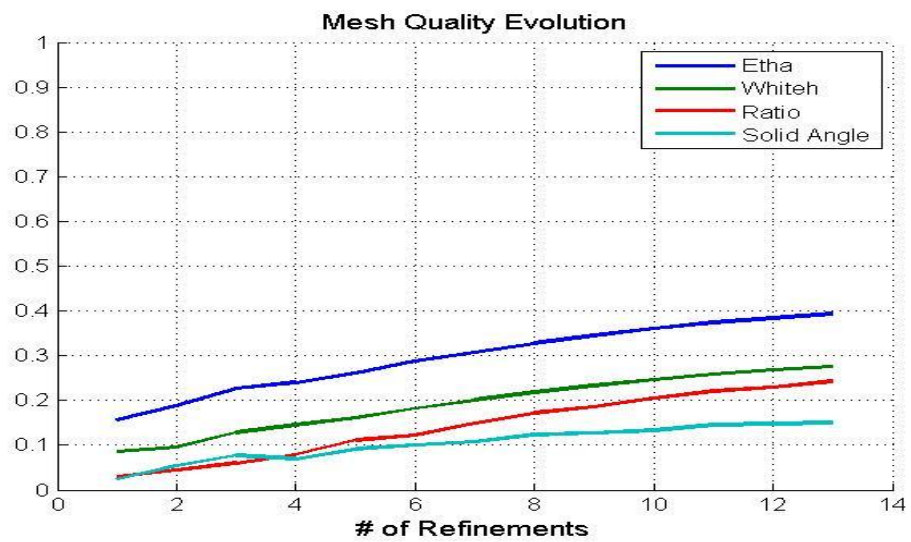


Figura 27 (b) : LE-Trisección uniforme en un tetraedro cap.

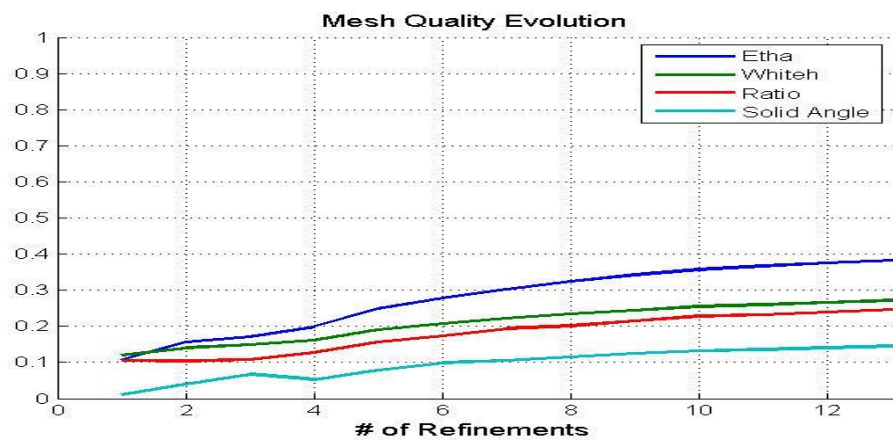


Figura 27 (c) : LE-Trisección uniforme en un tetraedro needle.

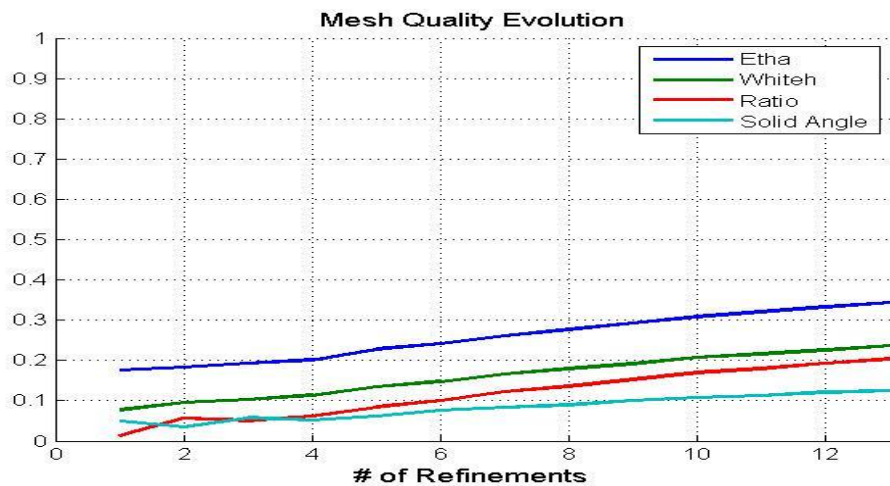


Figura 27 (d) : LE-Trisección uniforme en un tetraedro sliver.

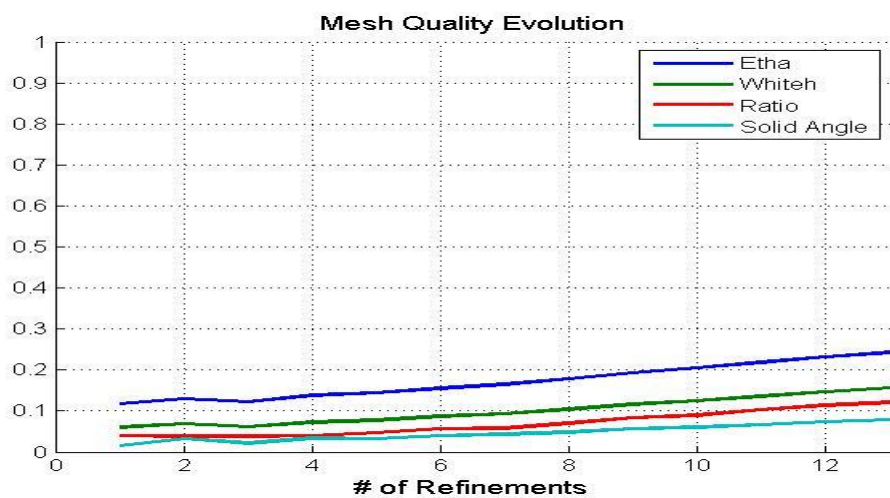


Figura 27 (e) : LE-Trisección uniforme en un tetraedro wedge.

Como medida de la calidad de τ^n definimos $\eta(\tau^n) = \min\{\eta(t_i^n)\}$ para todo $t_i^n \in \tau^n$. Luego el valor $c_\eta = \frac{\eta(\tau^n)}{\eta(\tau^1)}$ es calculado donde n es el último nivel de refinamiento iterativo.

Este valor es tomado como una estimación de c .

La tabla 3 muestra los diferentes resultados obtenidos y la Figura 28 presenta la evolución de los valores de $c_\eta = \frac{\eta(\tau^n)}{\eta(\tau^1)}$ para $n=13$.

Debe ser notado que el caso menos favorable para la constante de degeneración corresponde al tetraedro regular como malla inicial ($c=0.0174$). Por otra parte, el mejor caso para la constante de degeneración corresponde al peor tetraedro inicial, el needle, en completa correspondencia con investigaciones realizadas por otros autores (A. Plaza, et al., 2005).

En esta etapa, el interés era repetir los experimentos anteriores, pero ahora con la Cuatrisección 3D y obtener un coeficiente experimental de c . La malla refinada con

nivel τ^{10} contiene $4^{10}=1, 048, 576$ elementos. Los resultados se resumen en la Tabla 4 y la Figura 29 presenta la evolución de los valores de $c_\eta = \frac{\eta(\tau^n)}{\eta(\tau^1)}$.

La Tabla 4 muestra que el caso menos favorable de todos los experimentos corresponde al tetraedro tipo wedge como malla inicial y los mejores valores son para los tetraedros tipos cap y sliver.

| Forma Tetraedro | $\eta(t_1^1)$ | $\min\{\eta(t_i^{13})\}$ | $c = \frac{\eta(\tau^{13})}{\eta(\tau^1)}$ | Tetraedros |
|-----------------|---------------|--------------------------|--|------------|
| Regular | 1.00 | 0.0174 | 0.0174 | 1594323 |
| Cap | 0.2033 | 0.0183 | 0.0900 | 1594323 |
| Needle | 0.0550 | 0.0090 | 0.1636 | 1594323 |
| Sliver | 0.1833 | 0.0105 | 0.0573 | 1594323 |
| Wedge | 0.1659 | 0.0033 | 0.0199 | 1594323 |

Tabla 3 Forma de medida (η), malla τ^{13} con 1, 594, 323 elementos para los 5 diferentes tetraedros t_1^1 usando la partición LE-Trisección.

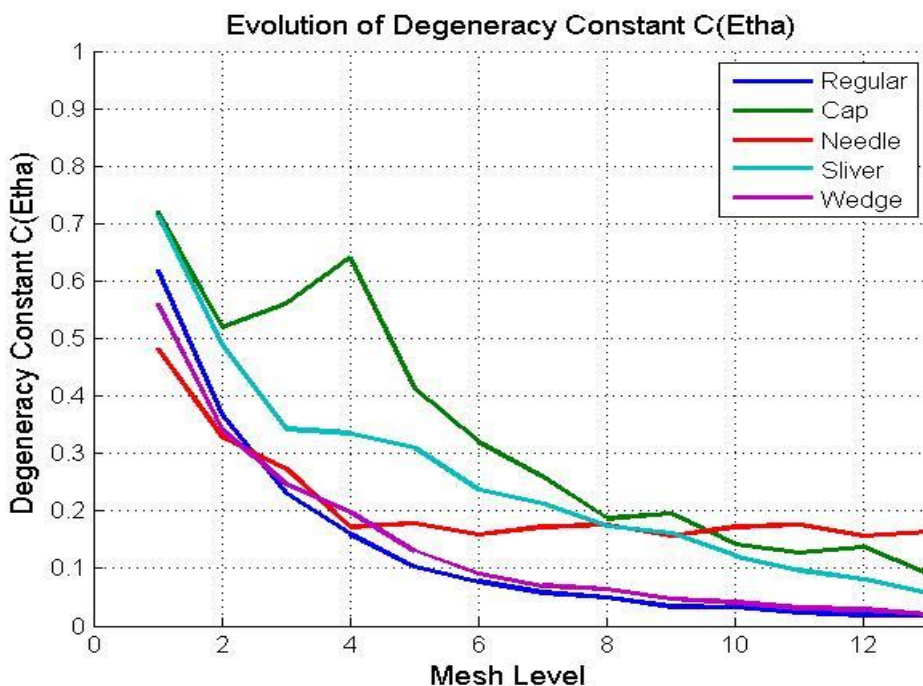


Figura 28 : Cinco tetraedros iniciales considerados y evolución de la constante de degeneración para la LE-Trisección.

| Forma Tetraedro | $\eta(t_1^1)$ | $\min\{\eta(t_i^{10})\}$ | $c = \frac{\eta(\tau^{10})}{\eta(\tau^1)}$ | Tetraedros |
|-----------------|---------------|--------------------------|--|------------|
| Regular | 1.00 | 0.0023 | 0.0023 | 1048576 |
| Cap | 0.2033 | 0.0017 | 0.0084 | 1048576 |
| Needle | 0.0550 | 0.0002 | 0.0036 | 1048576 |
| Sliver | 0.1833 | 0.0008 | 0.0044 | 1048576 |
| Wedge | 0.1659 | 0.0003 | 0.0018 | 1048576 |

Tabla 4 Forma de medida (η), malla τ^{10} con 1, 048, 576 elementos para los 5 diferentes tetraedros t_1^1 usando la partición LE-Cuatrisección.

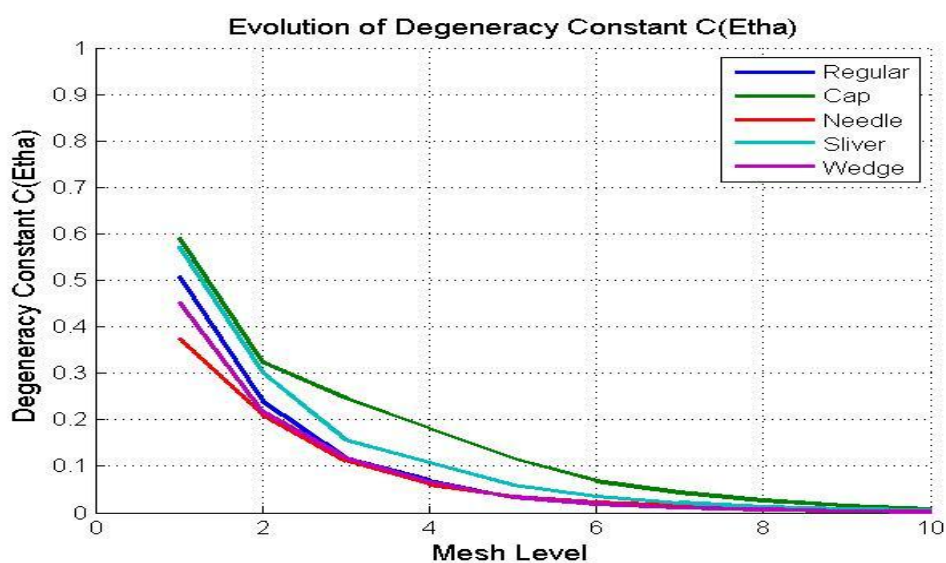


Figura 29 : Cinco tetraedros iniciales considerados y evolución de la constante de degeneración para la LE-Cuatrisección.

De las tablas 3-4 y las figuras 28-29 se deduce que el valor estimado para la constante de no degeneración es $c=0.0174$ para la partición basada en la trisección por el lado mayor y $c=0.0018$ para la cuatrisección respectivamente.

Las propiedades de auto mejora y no degeneración de la trisección por el lado mayor para mallas de tetraedros ha sido demostrado a través de evidencia empírica. Los resultados obtenidos permiten conjeturar sobre la propiedad no degenerativa de esta partición para el caso tridimensional y su comparación con la cuatrisección 3D, esta última partición si degenera la malla.

2.5 Algoritmos basados en Delaunay

Algoritmos de refinamiento eficientes que adecuadamente exploten las características de buena calidad de la triangulación Delaunay pueden ser considerados. En esta sección se discute el proceso a seguir para diseñar un

algoritmo de refinamiento Delaunay. La idea consiste en aplicar la versión incremental del algoritmo Delaunay para insertar los nuevos puntos en la malla. Debido a que el conjunto de nuevos puntos a ser insertados pertenecen a un área restringida de la triangulación (el área de refinamiento), inicialmente se conoce a cual tetraedro pertenece cada nuevo punto (existe una relación tetraedro-punto para cada punto a ser insertado), es posible explotar este conocimiento con el objetivo de reducir el tiempo de ejecución del algoritmo clásico Delaunay que tiene un costo de $O(n \log n)$ donde n es el número de vértices a ser insertados (Jonathan Richard Shewchuk, 2001). El algoritmo clásico Delaunay incremental (Algoritmo de Watson (Watson, 1981)) puede ser descrito básicamente como sigue:

Dado una simple triangulación inicial Delaunay, cada punto es incrementalmente insertado en la actual triangulación Delaunay por medio de los siguientes dos pasos:

1. Encontrar los tetraedros que contienen los puntos a ser insertados.
2. Encontrar todos los tetraedros que no satisfagan la prueba del círculo (La prueba del círculo se cumple si ningún punto es interior al circuncírculo del tetraedro). La unión de estos tetraedros define un poliedro que es re triangulado uniendo los nuevos puntos con sus vértices.

En orden de definir un criterio para decidir cuál punto debe ser añadido en cada tetraedro objetivo original, siete diferentes criterios fueron considerados y comparados: el centroide, puntos medios en las todas las aristas del tetraedro, la circunfera, la inesfera, bisección, trisección y cuatrisección de la arista mayor de cada tetraedro de la malla. Se realizó refinamiento global sobre mallas de tetraedros de muestra utilizando estas estrategias de inserción de puntos en combinación con Delaunay.

2.5.1 Criterios de inserción de puntos

Inserción punto Centroide-Delaunay

Estrategia de inserción de punto centroide con el algoritmo Delaunay. Experimentos simples con la idea de inserción de puntos por el centroide muestran que esta estrategia deteriora rápidamente la calidad de la triangulación en 2D, especialmente alrededor de las fronteras, aún cuando se aplica refinamiento global (Maria Cecilia Rivara & Inostroza, 1997). No obstante, en esta investigación se proporciona evidencia empírica que apoya la conjetura que este criterio no es válido en 3D, ver Figura 32(a).

En los experimentos desarrollados, el centroide de cada tetraedro fue iterativamente añadido como un punto a ser insertado en cada paso de refinamiento. Un algoritmo

de refinamiento Centroides-Delaunay puede ser esquemáticamente descrito como sigue donde T es el conjunto de tetraedros a ser refinado y t la malla de tetraedros:

Algoritmo de refinamiento Centroides-Delaunay(T,t)

Para cada $T \in t$

Sea P el centroide de T

Realizar inserción Delaunay del punto P de cada $T \in t$

Inserción punto Lado Mayor-Delaunay

La partición por el lado mayor ha sido la base para desarrollar algoritmos de refinamiento en 2D y 3D que aseguran la construcción de triangulaciones de buena calidad. Ambos algoritmos, algoritmos de refinamiento de bisección pura por el lado mayor y la combinación de bisección por el lado mayor con Delaunay constituyen algoritmos competitivos lineales que enfocan el problema del refinamiento de triangulaciones (M.-C. Rivara & Venere, 1994). La idea consiste en explotar el conocimiento que se tiene sobre la triangulación Delaunay, por sus buenas propiedades, introducir puntos en la malla como puntos medios del lado mayor de cada tetraedro seleccionado y utilizar el algoritmo Delaunay para que realice refinamiento automático luego de la inserción de los puntos. La bisección por el lado mayor garantiza de forma natural la construcción de triangulaciones no degeneradas y suaves, cuyas propiedades geométricas solo dependen de la malla inicial. Un algoritmo de refinamiento Lado Mayor-Delaunay puede ser esquemáticamente descrito como sigue donde T es el conjunto de tetraedros a ser refinado y t la malla de tetraedros:

Algoritmo de refinamiento Lado Mayor-Delaunay(T,t)

Para cada $T \in t$

Sea P el punto medio del lado mayor de T

Realizar inserción Delaunay del punto P de cada $T \in t$

Inserción Puntos Medios-Delaunay

La siguiente estrategia consiste en la inserción de puntos medios en todas las aristas del tetraedro en combinación con el algoritmo Delaunay. Un algoritmo de refinamiento Puntos Medios-Delaunay puede ser esquemáticamente descrito como sigue donde T es el conjunto de tetraedros a ser refinado y t la malla de tetraedros:

Algoritmo de refinamiento Puntos Medios-Delaunay(T,t)

Para cada $T \in t$

Sea P el conjunto de puntos medios de cada arista de T

Realizar inserción Delaunay del conjunto de puntos P de cada $T \in t$

Inserción Punto Circun esfera-Delaunay

La siguiente estrategia consiste en la inserción de puntos por el circuncentro de cada tetraedro en combinación con el algoritmo Delaunay. Un algoritmo de refinamiento Circun esfera-Delaunay puede ser esquemáticamente descrito como sigue donde T es el conjunto de tetraedros a ser refinado y t la malla de tetraedros:

Algoritmo de refinamiento Circun esfera-Delaunay(T,t)

Para cada $T \in t$

Sea P el punto de la circun esfera de T

Realizar inserción Delaunay del punto P de cada $T \in t$

Inserción Punto Inesfera-Delaunay

La siguiente estrategia consiste en la inserción de puntos por la esfera inscrita de cada tetraedro en combinación con el algoritmo Delaunay. Un algoritmo de refinamiento Inesfera-Delaunay puede ser esquemáticamente descrito como sigue donde T es el conjunto de tetraedros a ser refinado y t la malla de tetraedros:

Algoritmo de refinamiento Inesfera-Delaunay(T,t)

Para cada $T \in t$

Sea P el punto de la inesfera de T

Realizar inserción Delaunay del punto P de cada $T \in t$

Inserción Punto Trisección Lado Mayor-Delaunay

Estrategia de inserción de puntos por la trisección del lado mayor en combinación con el algoritmo Delaunay, que consiste en la inserción de dos puntos equidistantes en la arista mayor de cada tetraedro seleccionado y luego emplear el algoritmo Delaunay para realizar refinamiento automático. Un algoritmo de refinamiento Trisección Lado Mayor-Delaunay puede ser esquemáticamente descrito como sigue donde T es el conjunto de tetraedros a ser refinado y t la malla de tetraedros:

Algoritmo de refinamiento Trisección Lado Mayor-Delaunay(T,t)

Para cada $T \in t$

Trisección de la arista mayor de T

Sea P y Q los puntos generados

Realizar inserción Delaunay de los puntos P y Q de cada $T \in t$

Inserción Punto Cuatrisección Lado Mayor-Delaunay

Estrategia de inserción de puntos por la cuatrisección del lado mayor en combinación con el algoritmo Delaunay, que consiste en la inserción de tres puntos equidistantes en la arista mayor de cada tetraedro seleccionado y luego emplear el algoritmo Delaunay para realizar refinamiento automático. Un algoritmo de refinamiento Cuatrisección Lado Mayor-Delaunay puede ser esquemáticamente descrito como sigue donde T es el conjunto de tetraedros a ser refinado y t la malla de tetraedros:

Algoritmo de refinamiento Cuatrisección Lado Mayor-Delaunay(T,t)

Para cada $T \in t$

Cuatrisección de la arista mayor de T

Sea P, Q y R los puntos generados

Realizar inserción Delaunay de los puntos P, Q y R de cada $T \in t$

2.5.2 Análisis del costo de tiempo

El empleo iterativo de los algoritmos de refinamiento Delaunay mixtos, para resolver el problema del refinamiento de triangulaciones, introduce un número óptimo de N puntos, dentro de la región de refinamiento. En adición, el costo de tiempo del algoritmo es lineal en N . Este resultado viene directamente debido a que el tiempo esperado para cada inserción Delaunay (Guibas & Stolfi, 1985) es pequeño y prácticamente independiente de N . El tiempo de ejecución es dominado por la localización de los puntos, que es esencialmente constante (existe una conocida relación tetraedro-punto para cada punto a ser insertado). En este trabajo se proporciona evidencia empírica que muestra el comportamiento práctico del costo de los algoritmos Delaunay y su comparación con el algoritmo de refinamiento bisección pura por el lado mayor, en este contexto. Los algoritmos presentados obtienen velocidad utilizando el algoritmo Delaunay de CGAL (Rineau, 2010) en MatlabR2011a. Los experimentos se realizaron en una estación de trabajo AMD Phenom Quad-Core 3ghz con 8 gigabyte RAM, Sistema Operativo Windows 7 Ultimate 64 bit Edition. Para caracterizar el rendimiento de las implementaciones de nuestros algoritmos se compara el tiempo transcurrido para realizar cinco pasos de refinamiento global en un dominio cúbico, ver figura 30 y anexo #3. Como resumen, la tabla 5 presenta los resultados de la malla de prueba.

Como comentarios relacionados con el uso práctico de ambos esquemas de refinamiento, puede plantearse que el estudio empírico muestra que el algoritmo de

refinamiento Lado Mayor-Delaunay y la Bisección por el lado mayor constituyen algoritmos competitivos lineales que enfocan el problema del refinamiento de triangulaciones. Debe mencionarse que el trabajo computacional involucrado en el uso del algoritmo de refinamiento mixto Delaunay es superior, debido a que involucra la repetición del procedimiento incremental Delaunay (basado en la prueba del círculo Delaunay) requerida para insertar cada nuevo punto.

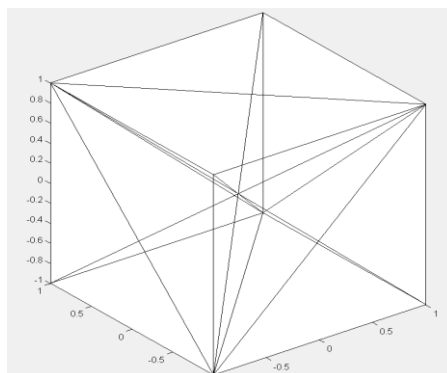


Figura 30 : Malla inicial, 12 tetraedros y 9 vértices.

| Algoritmo | Tiempo ejecución (s) | #Tetraedros | #Vértices |
|-----------------------------------|----------------------|-------------|-----------|
| Centroide-Delaunay | 6.6864 | 231667 | 34756 |
| Lado Mayor-Delaunay | 0.089313 | 6846 | 1210 |
| Puntos Medios-Delaunay | 650.3822 | 393216 | 68705 |
| Circun esfera-Delaunay | 0.42611 | 13591 | 2173 |
| Inesfera-Delaunay | 9.7045 | 229800 | 34853 |
| Trisección Lado Mayor-Delaunay | 5.6468 | 112022 | 17975 |
| Cuatrisección Lado Mayor-Delaunay | 1371.0105 | 1766012 | 272521 |
| Bisección por el Lado Mayor | 0.22139 | 384 | 125 |

Tabla 5 Resumen de los tiempos de ejecución de la malla de prueba.

En el caso del algoritmo de refinamiento por el lado mayor, la inserción de cada nuevo punto solo significa la partición del tetraedro individual y el refinamiento de algunos tetraedros vecinos que garantizan la conformidad de la malla refinada. Esto constituye una ventaja clara con respecto a los algoritmos basados en Delaunay, que son conocidos por sufrir de cuestiones no robustas como el caso de la colinearidad de n puntos alineados (Marquez, et al., 2010).

2.5.3 Comparativa de la calidad con algoritmos por el lado mayor

En este contexto, aún no existen resultados matemáticos de la comparativa de la calidad de los tetraedros generados utilizando variantes de algoritmos de refinamiento por el lado mayor versus un algoritmo Delaunay mixto con inserción de puntos por el lado mayor (Maria Cecilia Rivara & Inostroza, 1997).

Se proporciona un estudio numérico de la comparativa de la evolución de medidas de forma de la calidad cuando el algoritmo Lado Mayor-Delaunay es globalmente aplicado a mallas de tetraedros iniciales de diferentes formas.

La Figura 31 muestra la evolución de la media de algunas medidas de forma de la calidad mencionadas anteriormente para un tetraedro inicial de cada tipo de forma, en cada nivel de refinamiento cuando el algoritmo Lado Mayor-Delaunay es iterativamente globalmente aplicado.

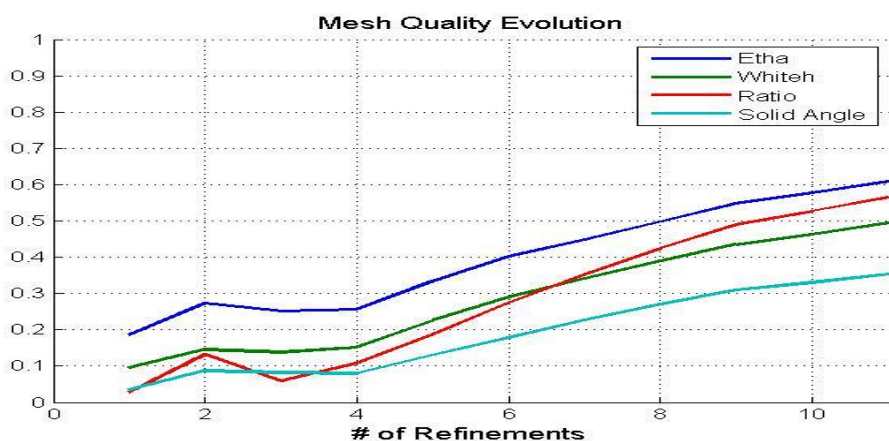


Figura 31 (a) Lado Mayor-Delaunay Uniforme en un tetraedro cap.

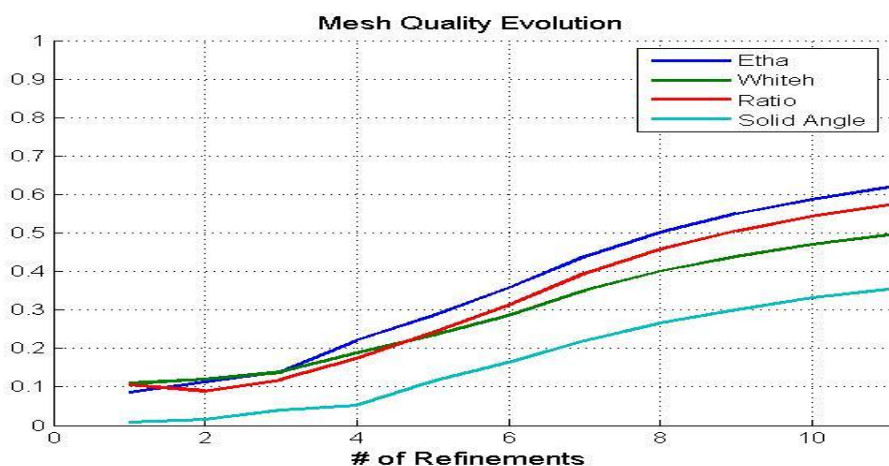


Figura 31 (b) Lado Mayor-Delaunay Uniforme en un tetraedro needle.

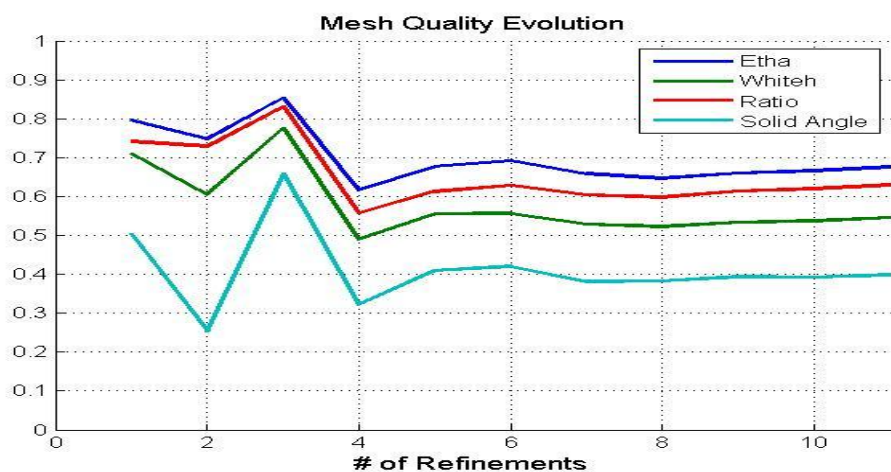


Figura 31 (c) Lado Mayor-Delaunay Uniforme en un tetraedro regular.

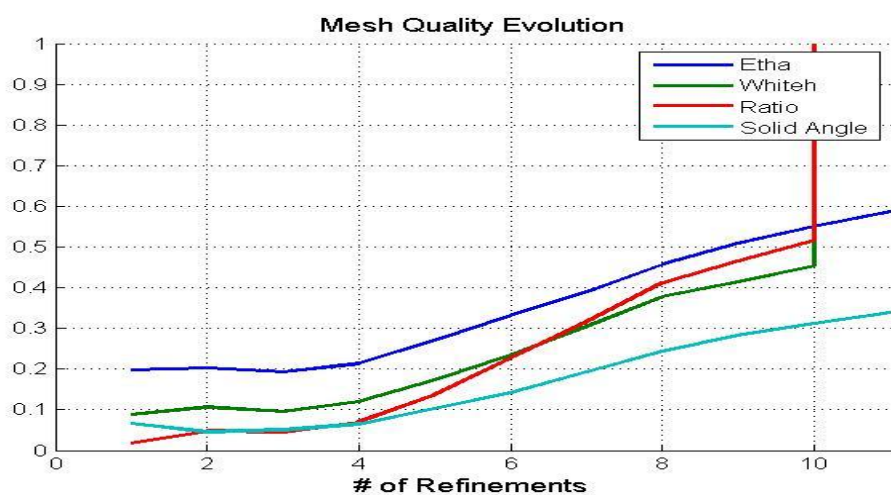


Figura 31 (d) Lado Mayor-Delaunay Uniforme en un tetraedro sliver.

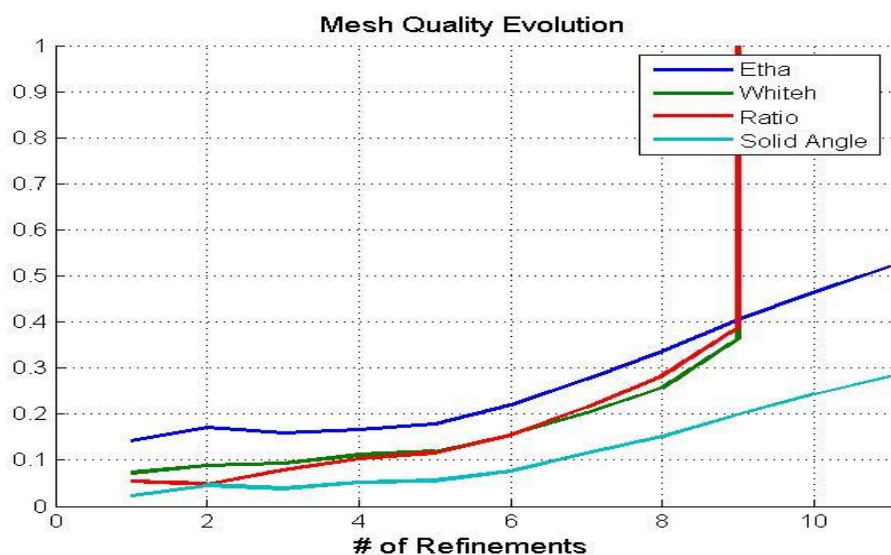


Figura 31 (e) Lado Mayor-Delaunay Uniforme en un tetraedro wedge.

Aún cuando se obtienen mejores resultados en la calidad por parte del algoritmo Lado Mayor-Delaunay con respecto a la bisección simple (Figura 19), trisección

(figura 27), cuatrisección del lado mayor (Figura 29) y en algunos casos superior a los resultados obtenidos por la partición 3T-LE (ver Anexo #2) sobre todo cuando la malla inicial es de muy mala calidad, tetraedro tipo wedge o needle (aguja), en un menor número de iteraciones, los algoritmos presentan serias dificultades debido a los errores de redondeo realizados por el ordenador (Escobar & Montenegro, 1996), especialmente en 3D, la presencia de tetraedros sliver de prácticamente volumen nulo corrompen la estructura de datos para luego poder aplicar otras variantes de algoritmos que no sean Delaunay. Las dificultades para evitarlos o removerlos han sido discutidas en numerosos artículos incluyendo (Escobar & Montenegro, 1996),(Jonathan Richard Shewchuk, 2002),(Freitag & Ollivier-Gooch, 1997),(Cheng, Dey, Edelsbrunner, Facello, & Teng, 2000). En esta tesis se permiten los tetraedros sliver en las mallas generadas, queda propuesto implementar algunas de estas técnicas de eliminación de sliver que pueden causar problemas en software de simulación numérica, ver Figura 31(d y e) donde se manifiestan valores erróneos de la calidad para las métricas Ratio y Whiteh .

Como principal comentario de nuestros experimentos, se deduce que los algoritmos de refinamiento Delaunay debido a las cuestiones no robustas, son factibles por sus propiedades de auto mejora de la calidad que serán estudiadas en la siguiente sección o para generar una malla inicial de buena calidad.

2.5.4 Propiedad Auto mejora de la calidad de los algoritmos basados en Delaunay

En esta sección se presentan algunas pruebas para comparar los resultados de la propiedad de auto-mejora de nuestras estrategias de refinamiento basadas en Delaunay. La calidad de los tetraedros generados mejora a medida que el refinamiento procede. Para nuestros experimentos, comenzamos por un tetraedro inicial de mala calidad tipo aguja (needle), se llegaron a obtener mallas de hasta 2 millones de tetraedros.

La aplicación global de estas estrategias de inserción de puntos basadas en Delaunay conduce a una mejora de la calidad de las nuevas mallas generadas. El estudio del comportamiento de algunas formas de medida de la calidad previamente mencionadas permite conjeturar sobre la propiedad de auto mejora de nuestros algoritmos.

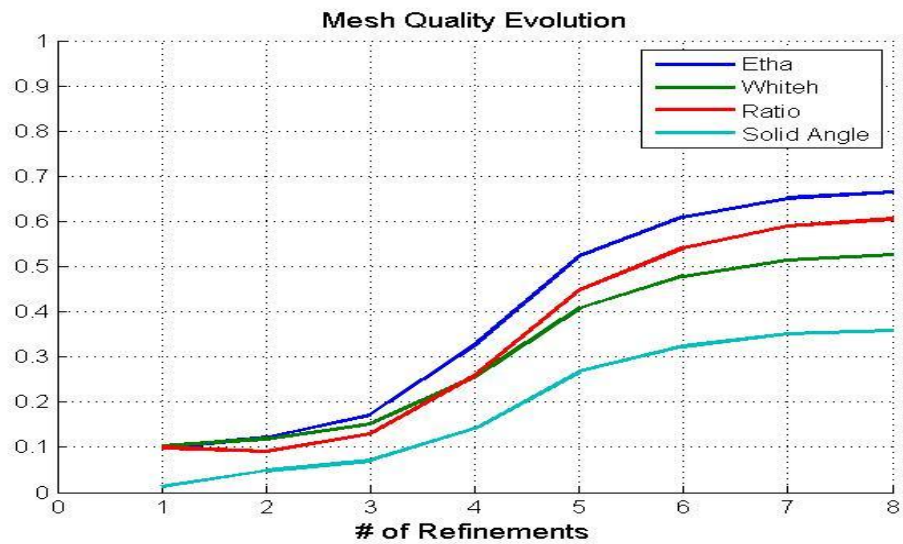


Figura 32 (a) Centroid-Delaunay Uniforme en un tetraedro needle.

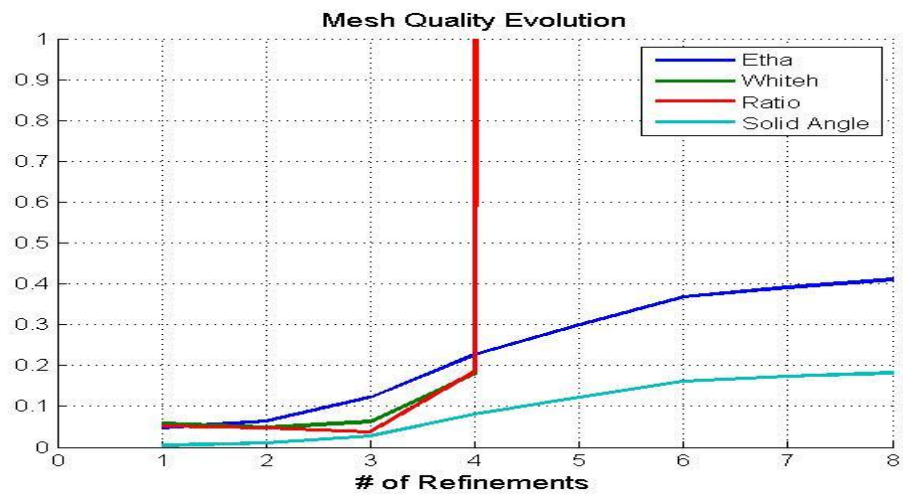


Figura 32 (b) Circunscrit-Delaunay Uniforme en un tetraedro needle.

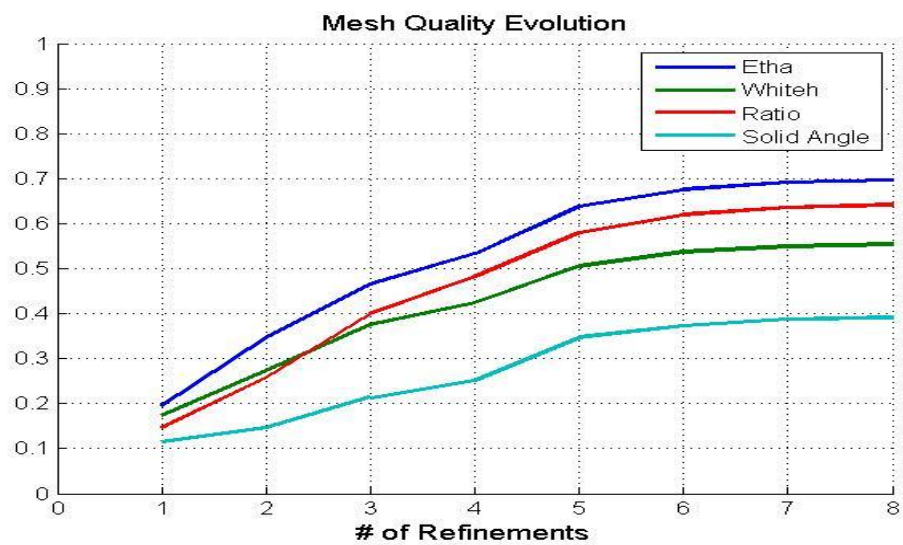


Figura 32 (c) Inscribit-Delaunay Uniforme en un tetraedro needle.

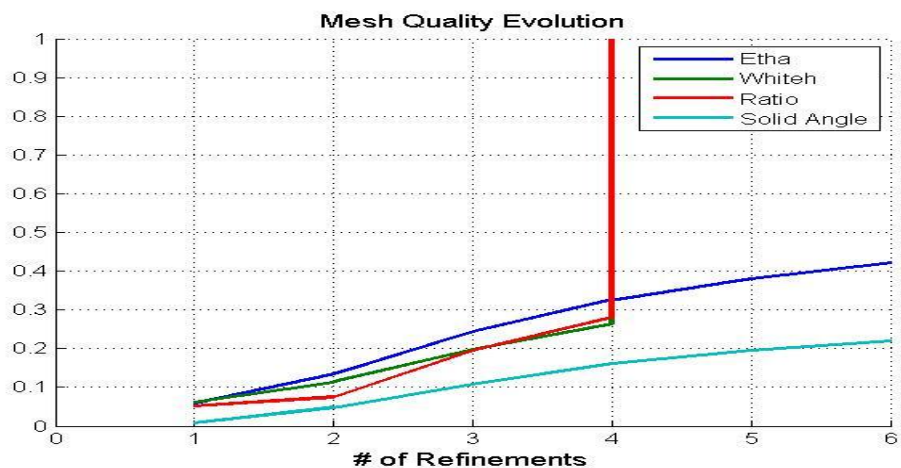


Figura 32 (d) Cuatrisección Lado Mayor-Delaunay Uniforme en un tetraedro needle.

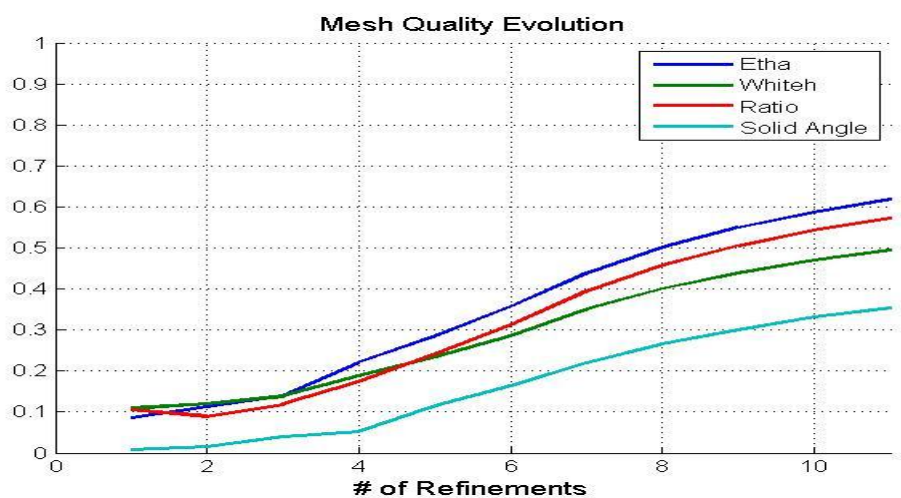


Figura 32 (e) Lado Mayor-Delaunay Uniforme en un tetraedro needle.

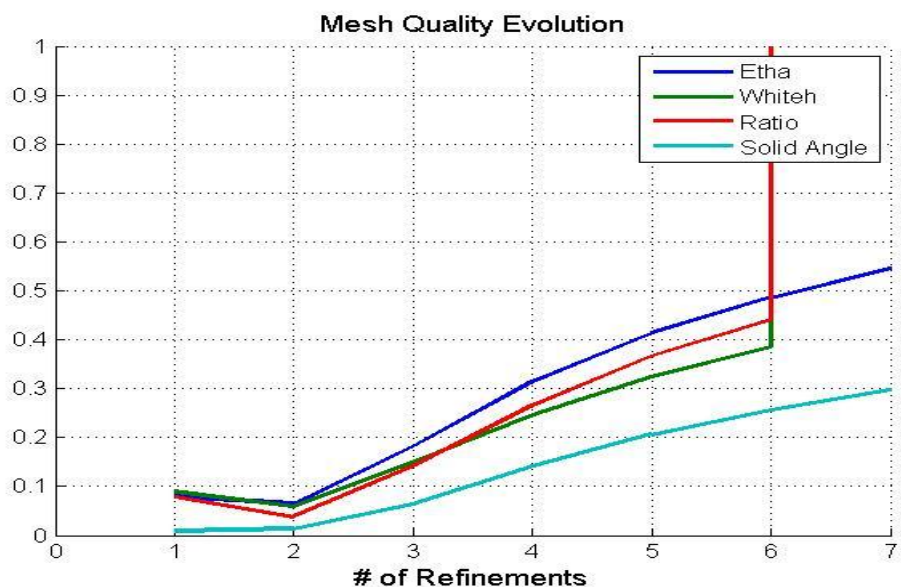


Figura 32 (f) Trisección Lado Mayor-Delaunay Uniforme en un tetraedro needle.

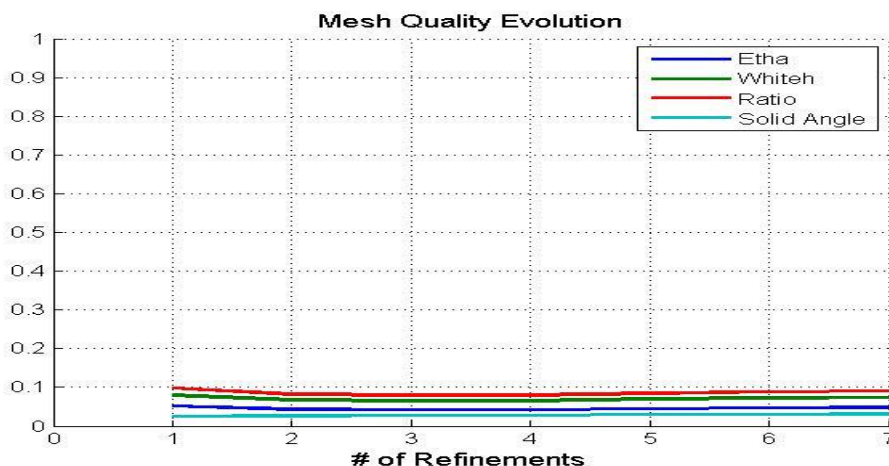


Figura 32 (g) Puntos Medios-Delaunay Uniforme en un tetraedro needle.

La Figura 32 muestra la evolución de diferentes formas de medida para los tetraedros obtenidos en el orden de 1-2 millones durante los pasos de refinamiento global de cada algoritmo explicado anteriormente. De los resultados reportados, la Figura 32 (a y c), se hace notar la propiedad de auto mejora de la calidad de estas estrategias de inserción de puntos basadas en Delaunay cuando se aplican iterativamente a una malla inicial de mala calidad. Debe ser mencionado que el caso menos favorable para la degeneración corresponde a la figura 32(g) debido a que la inserción de un gran número de puntos puede producir degeneración de la triangulación (Marquez, et al., 2010).

Luego de presentar una completa discusión del diseño de algoritmos de refinamiento Delaunay, incluyendo una sección de análisis del costo. Se concluye que las estrategias consideradas en esta tesis constituyen una excelente herramienta para la mejora de la calidad de las triangulaciones.

2.6 Algoritmo 4T-Baricentro

Definición 7. (Partición 4T(Tetraedros)-Baricentro) Para un tetraedro t con vértices ABCD la partición 4T-Baricentro se define así, ver figura 33:

1. Añadir un nodo nuevo P en el baricentro de t .
2. Unir el baricentro P con todos los vértices de t que forman las caras de t .

Para conjeturar sobre las propiedades no degenerativas de este algoritmo de refinamiento, es tomado un tetraedro de cada tipo de forma para los experimentos, ver Figura 34 (a-e). Se estudia el comportamiento de las medidas de la calidad mencionadas anteriormente cuando la aplicación iterativa del algoritmo procede. Debe ser notado que la aplicación global del algoritmo sobre la malla no necesita de

refinamientos adicionales de forma tal que se obtenga una malla conforme pues el punto se inserta en el centro de gravedad.

Se realizaron 10 niveles de refinamiento globales sobre una malla inicial que contiene un tipo de tetraedro diferente para cada caso hasta llegar a 1 048 576 elementos. La Figura 34 muestra la evolución de las diferentes medidas de la calidad para los tetraedros obtenidos durante las cinco pruebas realizadas anteriormente. De los resultados reportados, se deduce que el algoritmo degenera la malla en todos los casos posibles cuando es iterativamente aplicado, ver Anexo #4.

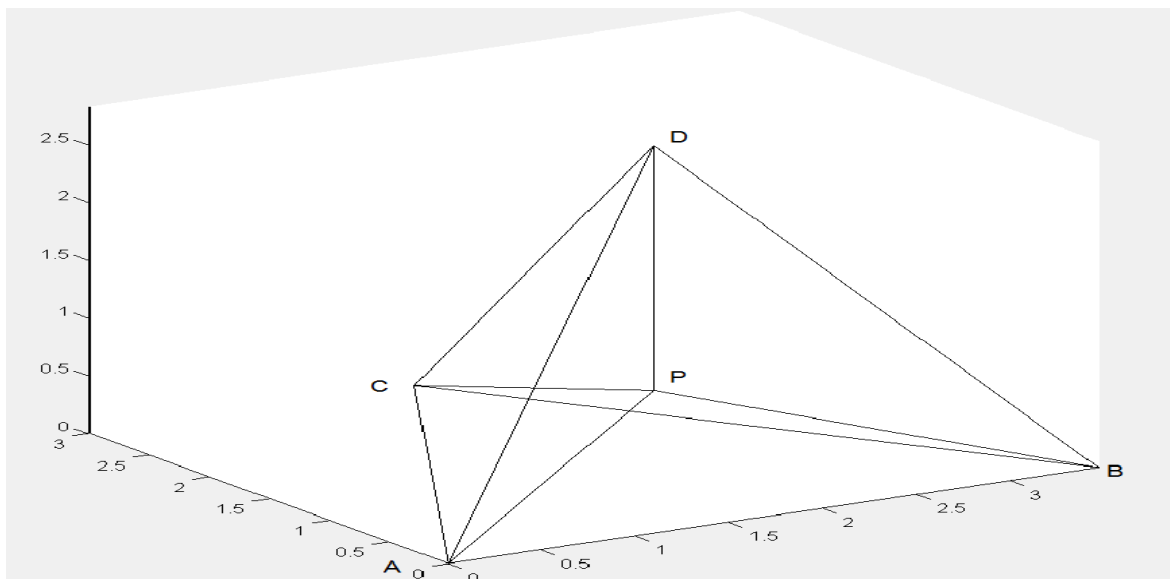


Figura 33 Partición 4T-Baricentro de un tetraedro.

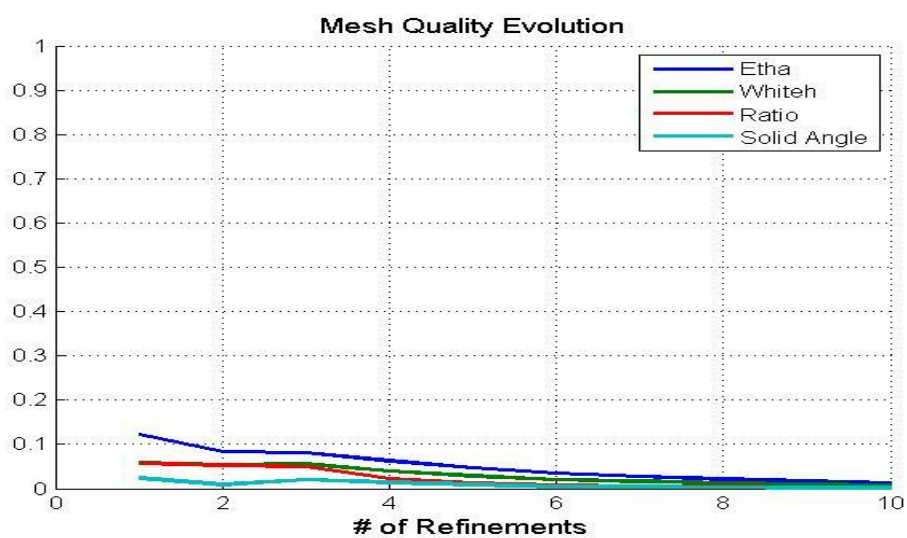


Figura 34 (a) 4T-Baricentro uniforme en un tetraedro cap.

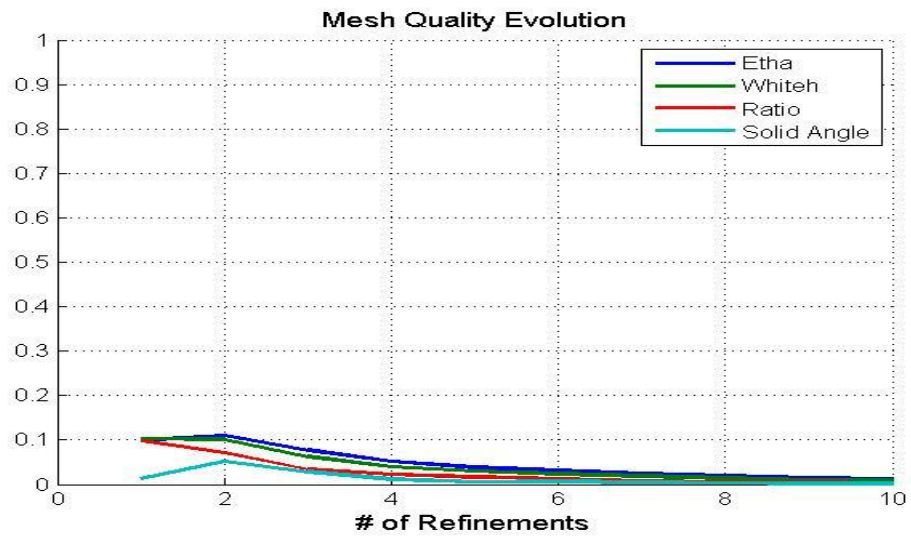


Figura 34 (b) 4T-Baricentro uniforme en un tetraedro needle.

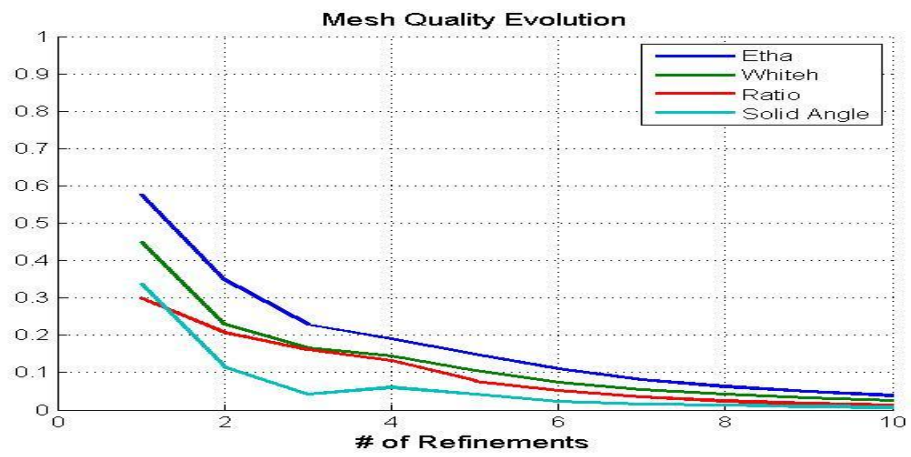


Figura 34 (c) 4T-Baricentro uniforme en un tetraedro regular.

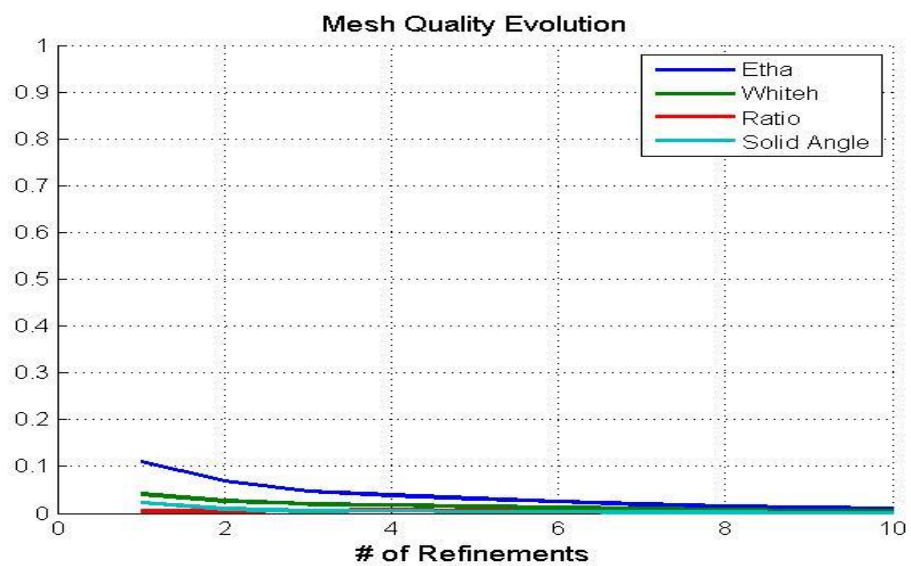


Figura 34 (d) 4T-Baricentro uniforme en un tetraedro sliver.

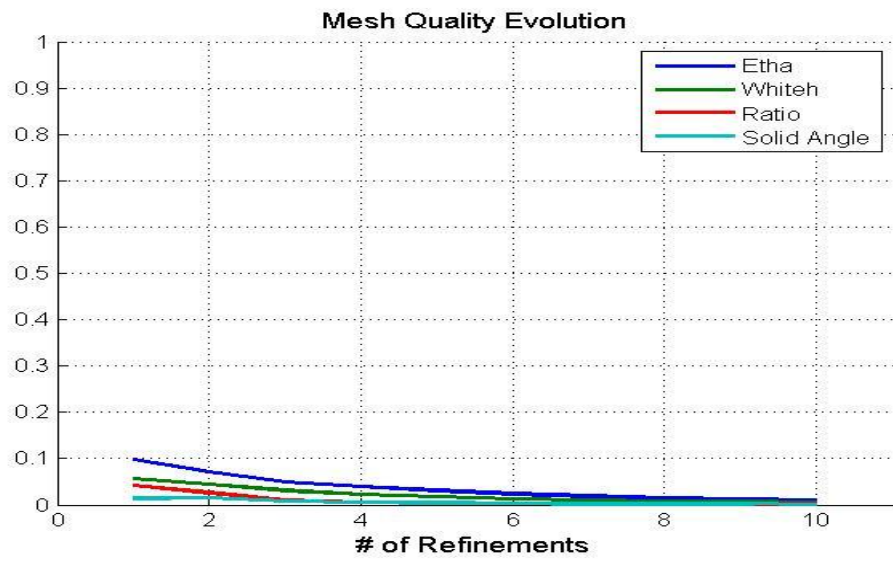


Figura 34 (e) 4T-Baricentro uniforme en un tetraedro wedge.

2.7 Conclusiones

Al finalizar este capítulo se pueden hacer afirmaciones sobre las propiedades de los algoritmos de refinamiento implementados en la aplicación Matlab. En los resultados obtenidos de la experimentación realizada se puede concluir que:

1-El algoritmo 3T-LE ofrece niveles de calidad superior a todas las variantes de algoritmos de refinamiento por el lado mayor.

2-Se verifica la propiedad no degenerativa de la trisección 3D y su comparación con la cuatrisección 3D, esta última partición si degenera la malla.

3-Los algoritmos de refinamiento Delaunay debido a las cuestiones no robustas, son factibles por sus propiedades de auto mejora de la calidad o para generar una malla inicial de buena calidad.

4-El algoritmo 4T-Baricentro degenera la malla de tetraedros.

Finalmente debe ser mencionado que en concordancia con las investigaciones realizadas por la profesora Rivara los algoritmos de refinamiento por el lado mayor son los más apropiados en los siguientes contextos:

En el refinamiento adaptivo de triangulaciones necesario en problemas complejos que dependen del tiempo y del rendimiento (M. C. Rivara, 1989).

En el uso práctico de métodos de mallados de elementos finitos sobre mallas irregulares (M. C. Rivara, 1984, 1986) debido a que estos algoritmos garantizan la construcción de secuencias de triangulaciones irregulares y anidadas.

CONCLUSIONES GENERALES

Para solucionar el problema científico planteado se asumieron los fundamentos teóricos de la disciplina Generación de Mallas como punto de partida.

La aplicación desarrollada en Matlab puede ser útil para la computación numérica de mallas de tetraedros, y permite asistir la toma de decisiones de los algoritmos de refinamiento propuestos a través de evidencia empírica. Luego de realizar el estudio experimental se arriba a la siguientes conclusiones:

1-El algoritmo 3T-LE ofrece niveles de calidad superior a todas las variantes de algoritmos de refinamiento por el lado mayor.

2-Se verifica la propiedad no degenerativa de la trisección 3D y su comparación con la cuatrisección 3D, esta última partición si degenera la malla.

3-Los algoritmos de refinamiento Delaunay debido a las cuestiones no robustas, son factibles por sus propiedades de auto mejora de la calidad o para generar una malla inicial de buena calidad.

4-El algoritmo 4T-Baricentro degenera la malla de tetraedros.

5-Cuando el conjunto de datos es del orden de millones de elementos, el tiempo de ejecución degenera el rendimiento para aplicaciones de tiempo real. Por ejemplo, estrategias de partición de dominios en dimensión 3 y arquitecturas paralelas son un campo de acción importante.

Por todo lo anterior, se considera probada la hipótesis planteada originalmente.

RECOMENDACIONES

Sobre la base de la investigación realizada y la experiencia acumulada durante la realización de este trabajo, se presentan a continuación una serie de recomendaciones para una posterior ampliación, modificación, mejora y construcción:

1. Implementación y prueba de los algoritmos sobre una estructura de datos que almacene información de adyacencia.
2. Estudio de las propiedades de los algoritmos de refinamiento local.
3. Estudio de técnicas para remover los tetraedros sliver que genera Delaunay en 3D debido a los errores de redondeo.

BIBLIOGRAFÍA

- Balman, M. (2006). PARALLEL TETRAHEDRAL MESH REFINEMENT. Master Thesis, Bogazici University.
- Bansch, E. (1991). Local mesh refinement in 2 and 3 dimensions. *IMPACT Com*, 3, 181-191.
- Bern, M., & Eppstein, D. (1995). Mesh generation and optimal triangulation. *Computing in Euclidean Geometry*(47–123).
- Bey, J. (1995). Tetrahedral grid refinement. *Computing*, 55, 355-378.
- Bischoff, S., & Kobbelt, L. (2004). Teaching meshes, subdivision and multiresolution techniques. *Comput. Aided Design* 36(14).
- Botsch, M., & Kobbelt, P. (2001). A Robust Procedure to Eliminate Degenerate Faces from Triangle Meshes.
- Botsch, M., Pauly, M., Rössl, C., Bischoff, S., & Kobbelt, L. (2006). Geometric Modeling Based on Triangle Meshes. 14-15.
- Bowyer, A. (1981). Computing Dirichlet tessellations. *The Computer Journal*, 24(162).
- Burkardt, J. (2004). Geometry Package, from http://people.sc.fsu.edu/~jburkardt/txt/gnu_lgpl.txt
- Carey, G. F. (1997). *Computational Grids: generation, adaptation and solution strategies*: Taylor & Francis.
- Cheng, S.-W., Dey, T. K., Edelsbrunner, H., Facello, M. A., & Teng, S.-H. (2000). Sliver exudation. *J. ACM*, 883–904.
- Chew, P. (1993). Guaranteed-quality mesh generation for curved surfaces. *Proceedings of the 9th Annual Symposium on Computational Geometry*, 274-280.
- Cignoni, P. (2005-2009). MeshLab (Version 1.2.3). Retrieved from <http://meshlab.sourceforge.net>
- Cohen, D. (1999). *Sistemas de información para la toma de decisiones*: McGraw-Hill.
- D.N. Arnold, A. M. a. L. P. (2000). Locally adapted tetrahedral meshes using bisection. *SIAM J. Sci. Comput*, 22(2), 431-448.

- Dari, E. A. (1994). Contribuciones a la triangulación automática de dominios tridimensionales. Tesis Doctoral, Universidad Nacional de Cuyo, San Carlos de Bariloche.
- Delaunay, B. (1934). Sur la sphere vide. *SCI*, 793-800.
- Dompierre, J., Labbé, P., Guibault, F., & Camarero, R. (1998). Benchmarks for 3D unstructured tetrahedral mesh optimization, in: 7th Int. Mes. Roundt. Sandia Report SAND, 98-2250(459–478).
- E.B.Becker, G.F.Carey, & Oden, J. T. (1981). *Finite Elements: An introduction* (Vol. 1).
- Engwirda, D. (2009). MESH2D - Automatic Mesh Generation, from <http://www.mathworks.com/matlabcentral/fileexchange/25555-mesh2d-automatic-mesh-generation>
- Escobar, J. M., & Montenegro, R. (1996). Several aspects of three-dimensional Delaunay triangulation *Advances in Engineering Software*, 27(27-39).
- Freitag, L. A., & Ollivier-Gooch, C. (1997). Tetrahedral mesh improvement using swapping and smoothing. *Internat. J. Numer. Methods Engrg*, 3979-4002.
- Frey, P. J., & George, P.-L. (2000). *Mesh Generation: application to finite elements*: Hermes Science
- Garimella, R. V. (2002). Mesh data structure selection for mesh generation and FEA applications. *International Journal for Numerical Methods in Engineering*(55), 451–478 doi: 10.1002/nme.509
- George, P.-L., & Borouchaki, H. (1997). *Triangulation de Delaunay et maillage, applications aux éléments finis*.
- George, P.-L., & Borouchaki, H. (1998). *Delaunay Triangulation and Meshing. Applications to Finite Elements*.
- Goodrich, M. T., & Ramaiyer, K. (2000). *Geometric data structures*: Elsevier.
- Green, P. J. a. S., R. (1978). Computing Dirichlet tessellations in the plane. *The Computer Journal*, 21(168).
- Gu, X. (2010). Halfedge Mesh Library from <http://www.cs.sunysb.edu/%7Eegu/software/MeshLib>
- Guibas, L., & Stolfi, J. (1985). Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graphics*, 4(74-123).
- Hartmann, J., & Wernecke, J. (1998). *The VRML 2.0 Handbook*. Addison Wesley.

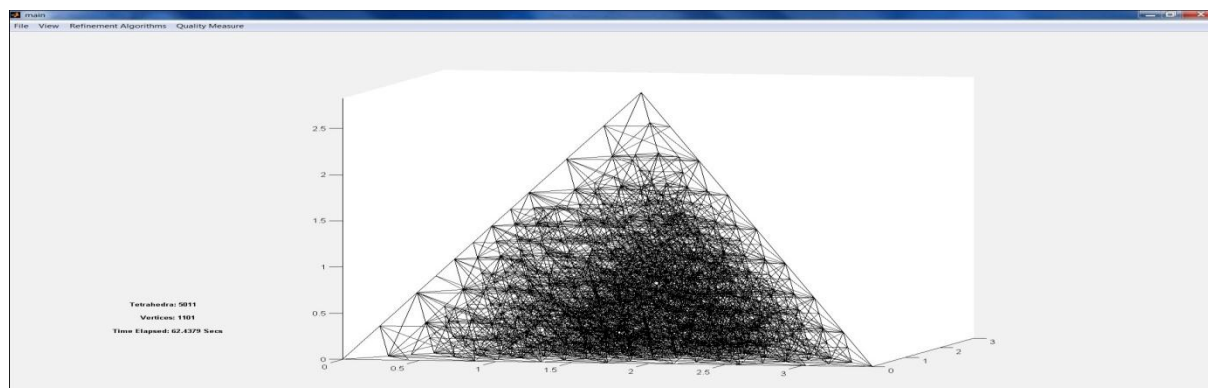
- Kossaczky, I. (1994). A recursive approach to local mesh refinement in two and three dimensions. *J. Comp. App. Math*, 55(275-288).
- Lawson, C. L. (1977). Software for C interpolation. *Mathematical Software III*, Academic Press, 161-194.
- Lee, D. T. a. S., B.J. (1980). Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer and Information Sciences*, 9(219).
- Lewis, B. A. a. R. S. (1976). Triangulation of planar regions with applications. *The Computer Journal*, 21, 324-332.
- Liu, A., & Joe, B. (1994a). On the shape of tetrahedra from bisection. *Math. Comp*, 63(141–154).
- Liu, A., & Joe, B. (1994b). Relationship between tetrahedron shape measures. *BIT*, 34(268–287).
- Liu, A., & Joe, B. (1995). Quality local refinement of tetrahedral meshes based on bisection. *SIAM J. Sci. Statist. Comput*, 16(1269–1291).
- Marquez, A., Moreno--Gonzalez, A., Plaza, A., & Suarez, J. P. (2008). The seven-triangle longest-side partition of triangles and mesh quality improvement. *Finite Elements in Analysis and Design*, 748-758.
- Marquez, A., Moreno-Gonzalez, A., Plaza, A., & Suarez, J. P. (2010). An edge-based Quasi-Delaunay refinement algorithm made simple and robust, Las Palmas de Gran Canaria, Spain.
- Mathworks, T. *Partial Differential Equations (PDE) Toolbox User's Guide*. Matlab Online manuals.
- Maubach, J. M. (1995). Local bisection refinement for n-simplicial grids generated by reflection. *SIAM J. Sci. Stat. Comp*, 16, 210-227.
- Miller, G.-L., Pav, S.-E., & Walkington, N.-J. (2002). Fully incremental 3d Delaunay mesh generation. *Proceedings of the 11th International Meshing Roundtable*, 75-86.
- Morales, L. F. (2000). *Triangulaciones delaunay y no-delaunay y particiones de triangulos en base al lado mas largo*. Tesis de magister, Universidad de Chile
- Mukherjee, A. (1996). *An adaptive finite element code for elliptic boundary value problems in three dimensions with applications in numerical relativity*. Doctoral, Penn. State University University Park.

- Muthukrishnan, E., Shiakolas, P. S., Nambiar, R. V., & Lawrence, K. L. (1995). Simple algorithm for adaptive refinement of three-dimensional finite element tetrahedral meshes. *AIAA*, 33, 928-932.
- Padron, M. A. (1999). Un algoritmo de desrefinamiento en dimension tres para mallas encajadas de tetraedros basado en el esqueleto. Universidad de Las Palmas de Gran Canaria.
- Padrón, M. A., Suárez, J. P., & Plaza, Á. (2007). Refinement based on longest-edge and self-similar four-triangle partitions. *Mathematics and Computers in Simulation*, 75(5-6), 251-262.
- Parthasarathy, V. N., Graichen, C. M., & Hathaway, A. F. (1993). A comparison of tetrahedron quality meshes. *Finite Elem. Anal.Des*, 15(225–261).
- Pav, S.-E. (2003). Delaunay refinement algorithms. PhD, Carnegie Mellon
- Plaza, A., & Carey, G. F. (2000). Local refinement of simplicial grids based on the skeleton. *Applied Numerical Mathematics*.
- Plaza, Á., Falcón, S., & Suárez, J. P. (2010). On the non-degeneracy property of the longest-edge trisection of triangles. *Applied Mathematics and Computation*, 216(862–869).
- Plaza, A., Padrón, M. A., & Carey, G. F. (2000). A 3D refinement/derefinement algorithm for solving evolution problems. *Applied Numerical Mathematics*, 32(4), 401-418.
- Plaza, A., & Rivara, M.-C. (2003). Mesh Refinement Based on 8-Tetrahedra Longest-Edge Partition. Sandia National Laboratories.
- Plaza, A., Suárez, J. P., & Padrón, M. A. (2005). Non-degeneracy study of the 8-tetrahedra longest-edge partition. *App. Num. Math*, 55 (4)(458–472).
- Plaza, Á., Suárez, J. P., Padrón, M. A., Falcón, S., & Amieiro, D. (2004). Mesh quality improvement and other properties in the four-triangles longest-edge partition. *Computer Aided Geometric Design*, 353-369.
- Rineau, L. (2010). CGAL User and Reference Manual: All Parts : 2D Conforming Triangulations and Meshes.
- Rivara, M.-C., & Venere, M. (1994). Cost analysis of the longest-side (triangle bisection) refinement algorithms for triangulations. *Engineering with Computers*.
- Rivara, M. C. (1984). Design and data structure for fully adaptive, multigrid finite-element software. *ACM Trans. Math.Software*, 10(242-264).

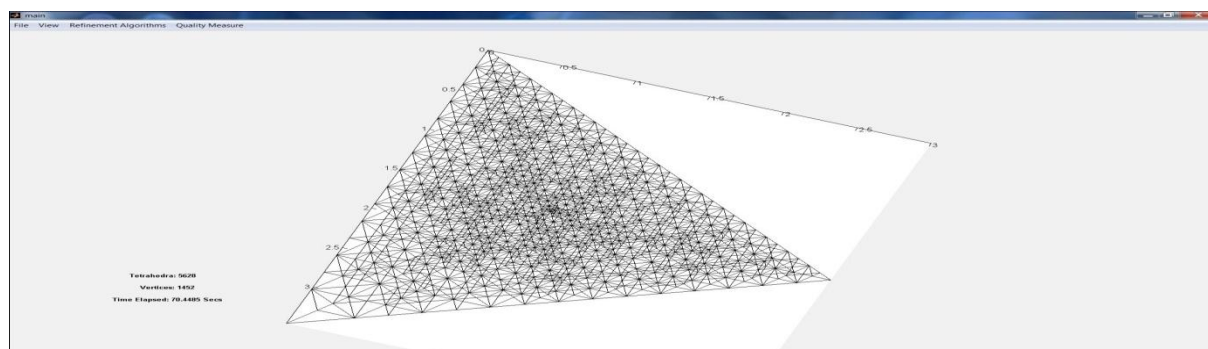
- Rivara, M. C. (1986). Adaptive finite element refinement and fully irregular and conforming triangulations. Accuracy Estimates and Adaptive Refinements in Finite Element Computations, Wiley(359-370).
- Rivara, M. C. (1989). Selective refinement /derefinement algorithms for sequences of nested triangulations. Int. j. numer.methods eng, 28(2889-2906).
- Rivara, M. C. (1997). New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. Numerical Methods in Engineering.
- Rivara, M. C., Hitschfeld, N., & Simpson, B. (2001). Terminal-edges Delaunay (small-angle based) algorithms for the quality triangulation problem. Computer Aided Design.
- Rivara, M. C., & Inostroza, P. (1997). USING LONGEST-SIDE BISECTION TECHNIQUES FOR THE AUTOMATIC REFINEMENT OF DELAUNAY TRIANGULATIONS. INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, 40(581-597).
- Rivara, M. C., & Iribarren, G. (1996). THE 4-TRIANGLES LONGEST-SIDE PARTITION OF TRIANGLES AND LINEAR REFINEMENT ALGORITHMS. MATHEMATICS OF COMPUTATION, 65(216), 1485-1502.
- Rivara, M. C., & Levin, C. (1992). A 3-D refinement algorithm suitable for adaptive and multi-grid techniques. Communications in applied numerical methods.
- Rivero, J. P. S. (2001). Estructuras de datos tipo grafo en los algoritmos de refinamiento y desrefinamiento basados en la biseccion por el lado mayor. Aplicaciones. Tesis doctoral, Universidad de Las Palmas de Gran Canaria.
- Rivero, J. P. S. (2009). Procesamiento geometrico con mallas triangulares, nuevos esquemas de refinamiento y aplicaciones. Las Palmas de Gran Canaria,España: CICYT del Ministerio de Educacion y Ciencia.
- Ruppert, J. (1995). A Delaunay refinement algorithm for quality 2-dimensional mesh generation. Journal of Algorithms, 548-585.
- Shewchuk, J. R. (1997). Delaunay refinement mesh generation. PhD, Carnegie Mellon University
- Shewchuk, J. R. (2001). Delaunay Refinement Algorithms for Triangular Mesh Generation. Berkeley.
- Shewchuk, J. R. (2002). What is a good linear element? Interpolation,conditioning, and quality measures. Proceedings of the 11th International Meshing Roundtable.

- Silva, S. d. S. (2007). Polymeco: A Polygonal Mesh Analysis and Comparison Tool Master, University of Aveiro.
- Stynes, M. (1980). On faster convergence of the bisection method for all triangles. *Math.Comp*, 152(1195–1201).
- Suarez, J. P., Abad, P., Plaza, A., & Padron, M. A. (2005). Computational aspects of the refinement of 3D tetrahedral meshes. *Computational Methods in Sciences and Engineering*.
- T.J.Baker. (1987). Three dimensional Mesh Generation by Triangulation of Arbitrary Point Sets, Hawaii.
- Velho, L., & Zorin, D. (2001). 4-8 subdivision. *Comput. Aided. Geom. Des*, 18(397-427).
- Vercellis, C. (2009). *Business Intelligence: Data Mining and Optimization for Decision Making*: John Wiley & Sons.
- Watson, D. F. (1981). Computing the n-dimensional Delaunay tessellation with applications to Voronoi polytopes. *Comput*, 24(167-172).
- Weiler, K. (1985). Edge-based data structures and concepts for solid modeling in curved-surface environments. *IEEE Computer Graphics and Applications*, 21-40.
- Wernicke, J. (1994). *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor*. Addison Wesley.
- Whitehead, J. H. C. (1940). On C1-complexes. *Ann. of Math*, 41, 809–824.

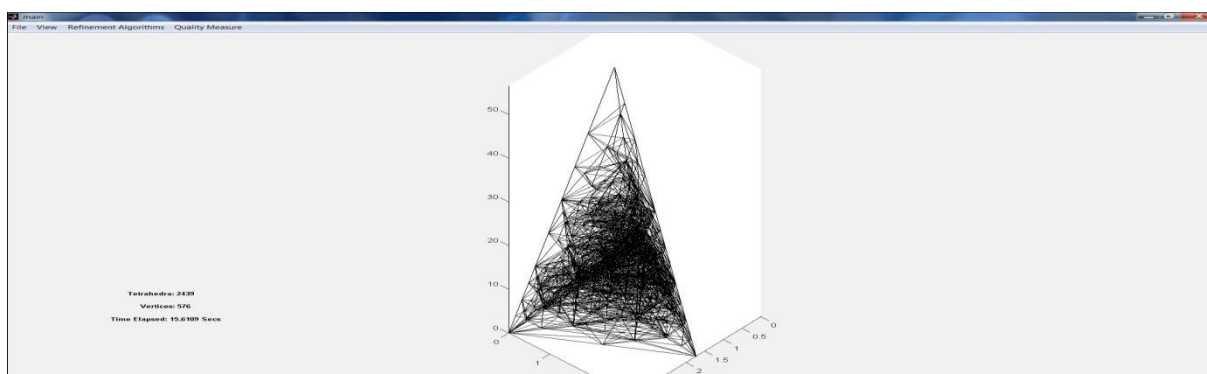
ANEXOS



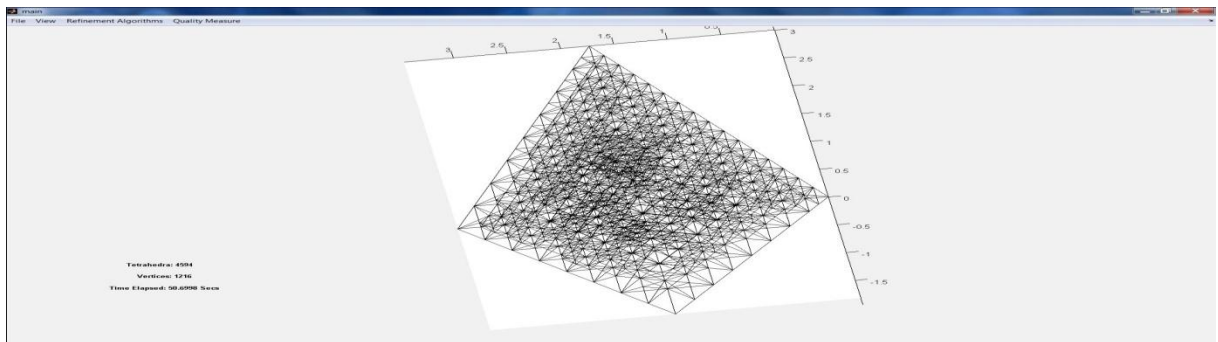
Anexo 1(a). Malla generada luego de 9 refinamientos globales por el algoritmo LE partiendo de 1 tetraedro inicial regular.



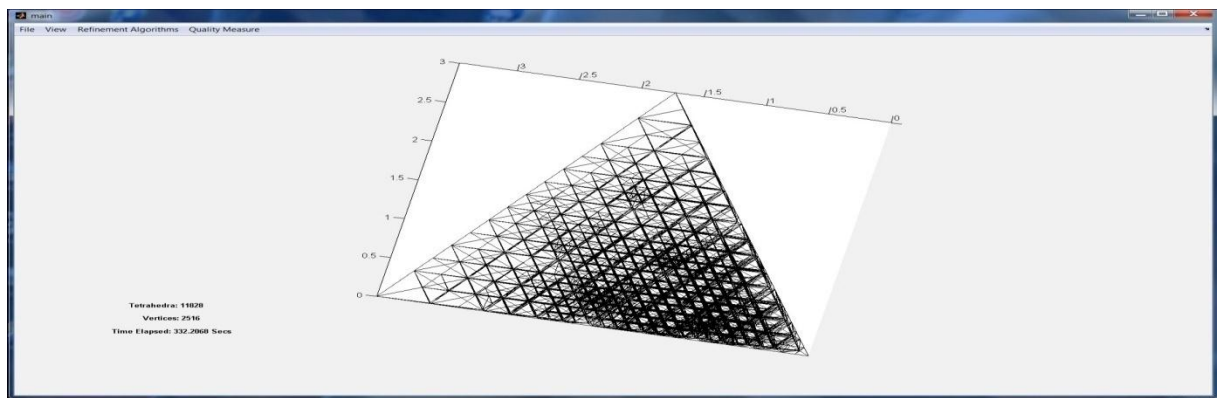
Anexo 1(b). Malla generada luego de 10 refinamientos globales por el algoritmo LE partiendo de 1 tetraedro inicial tipo cap.



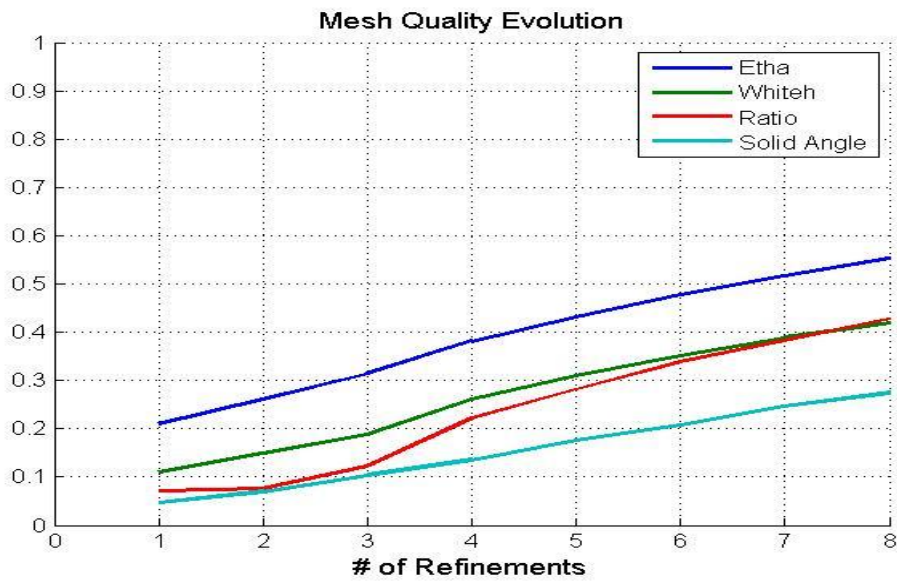
Anexo 1(c). Malla generada luego de 7 refinamientos globales por el algoritmo LE partiendo de 1 tetraedro inicial tipo needle.



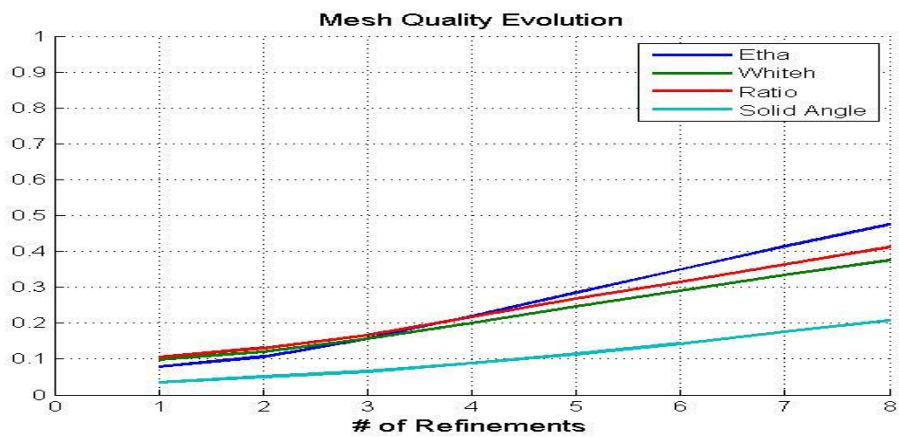
Anexo 1(d). Malla generada luego de 9 refinamientos globales por el algoritmo LE partiendo de 1 tetraedro inicial tipo sliver.



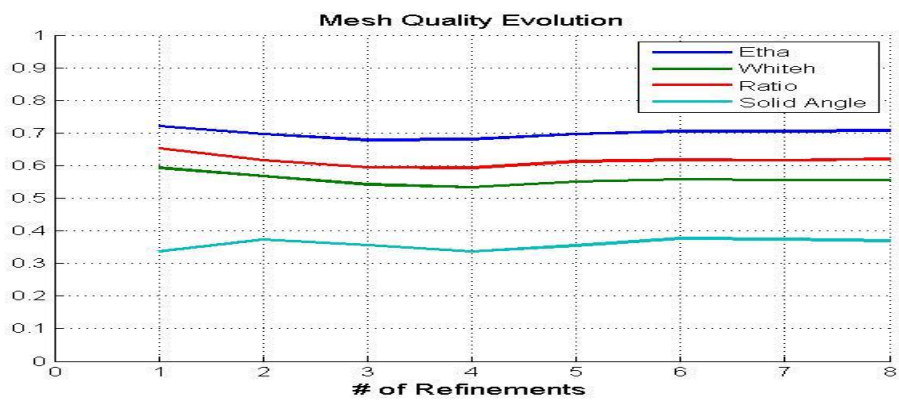
Anexo 1(e). Malla generada luego de 10 refinamientos globales por el algoritmo LE partiendo de 1 tetraedro inicial tipo wedge.



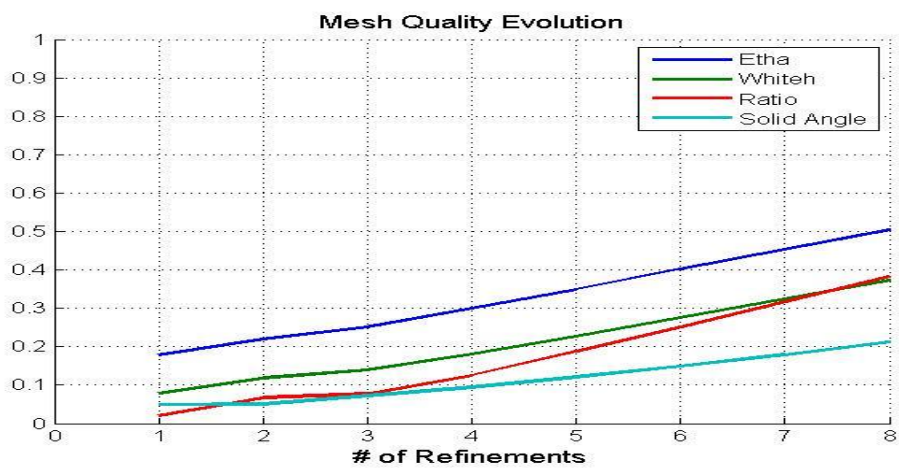
Anexo 2(a). 3T-LE uniforme en un tetraedro cap.



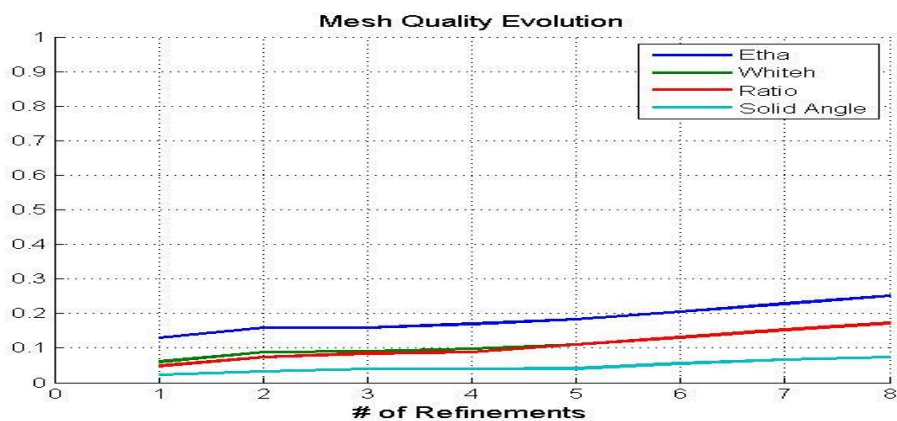
Anexo 2(b). 3T-LE uniforme en un tetraedro needle.



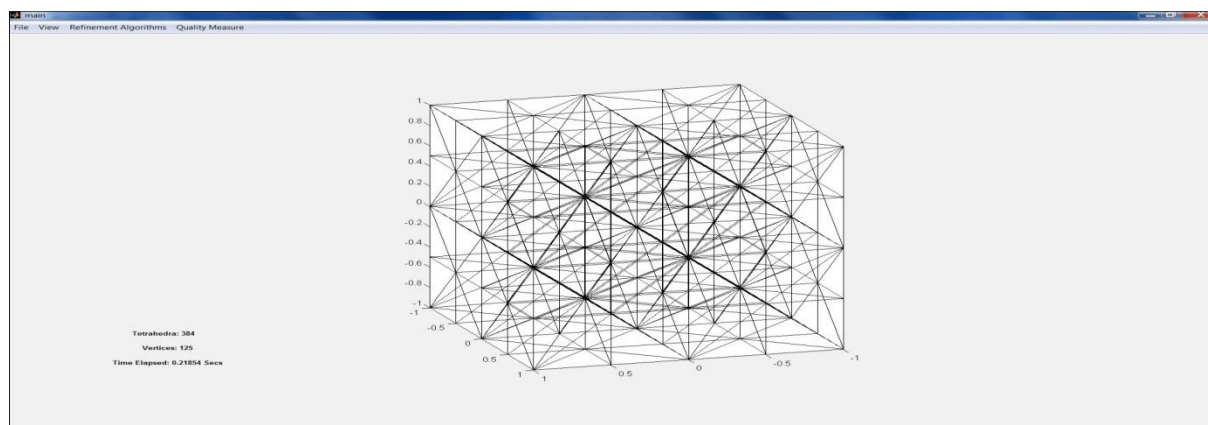
Anexo 2(c). 3T-LE uniforme en un tetraedro regular.



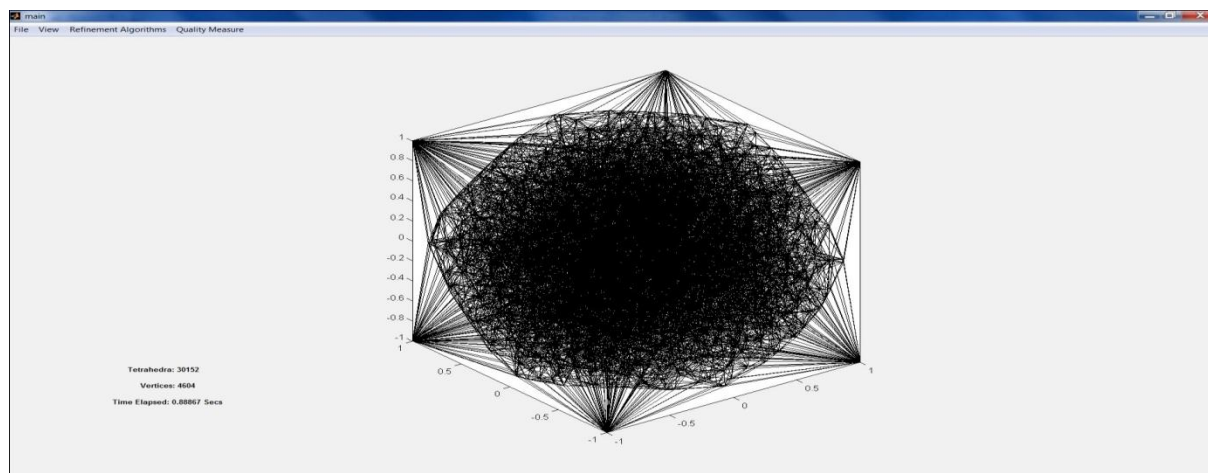
Anexo 2(d). 3T-LE uniforme en un tetraedro sliver.



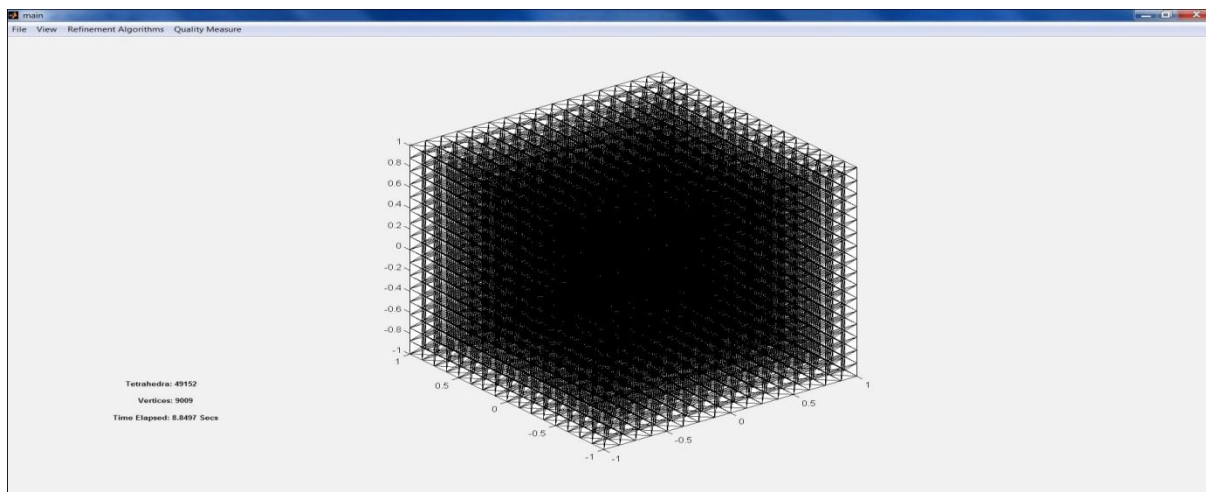
Anexo 2(e). 3T-LE uniforme en un tetraedro wedge.



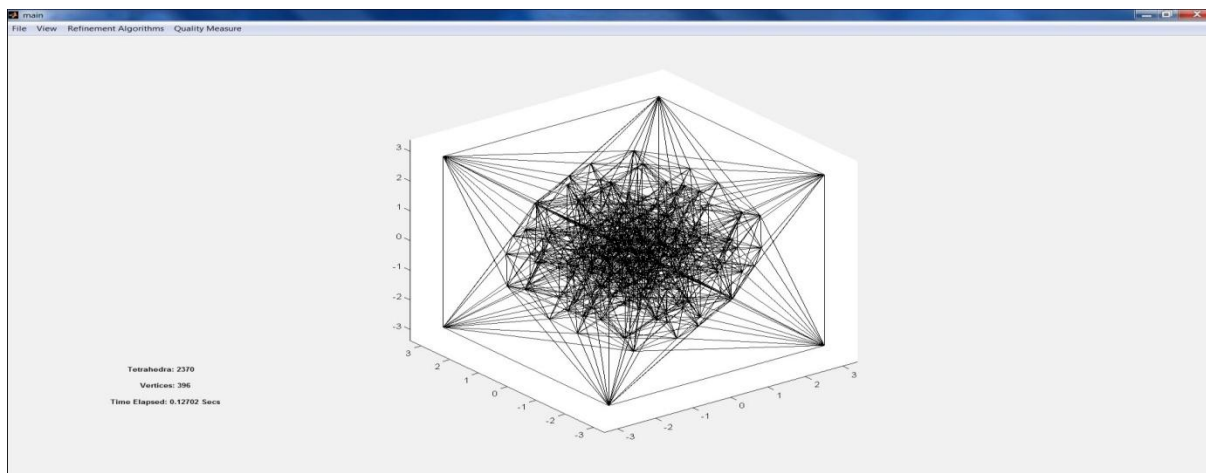
Anexo 3(a). Cubo refinado por 5 niveles del algoritmo Bisección Lado Mayor.



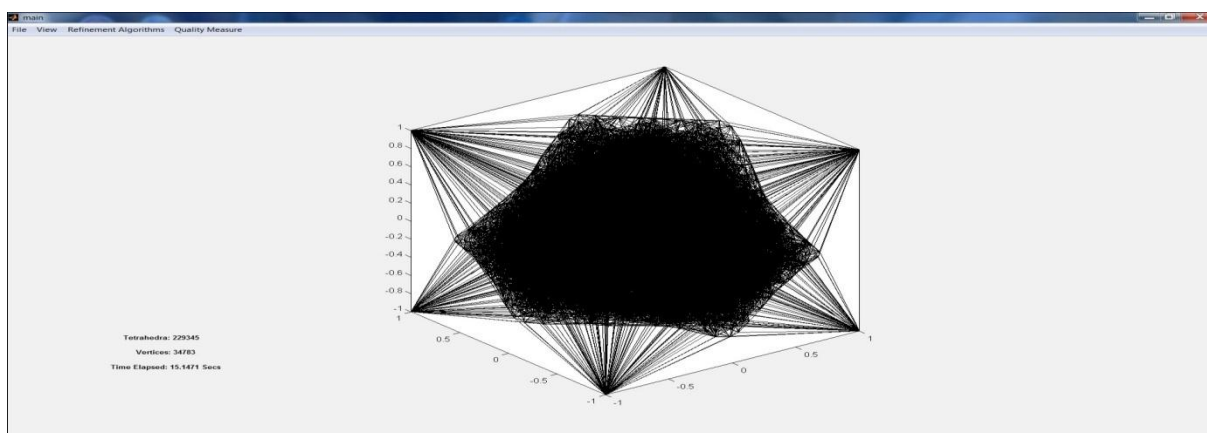
Anexo 3(b). Cubo refinado por 4 niveles del algoritmo Centroid-Delaunay.



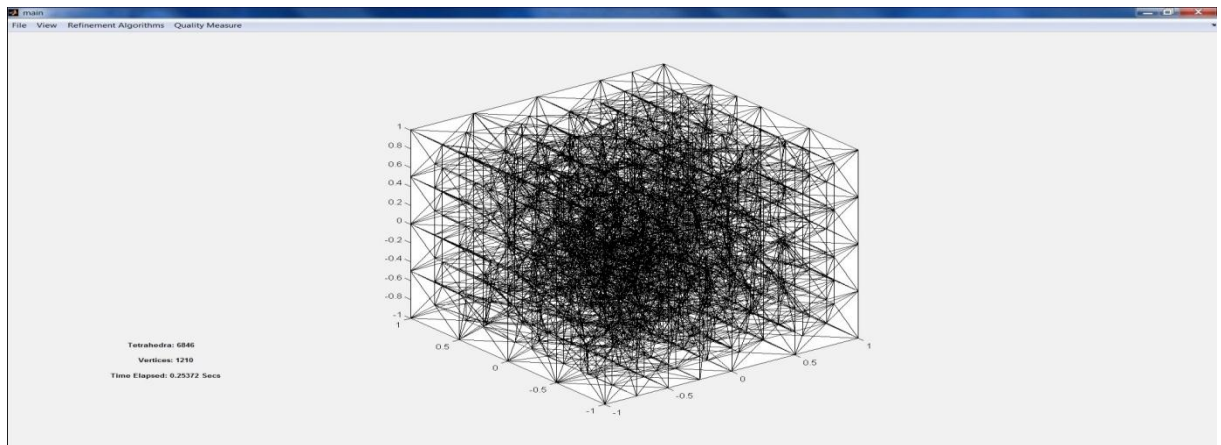
Anexo 3(c). *Cubo refinado por 4 niveles del algoritmo Puntos Medios-Delaunay.*



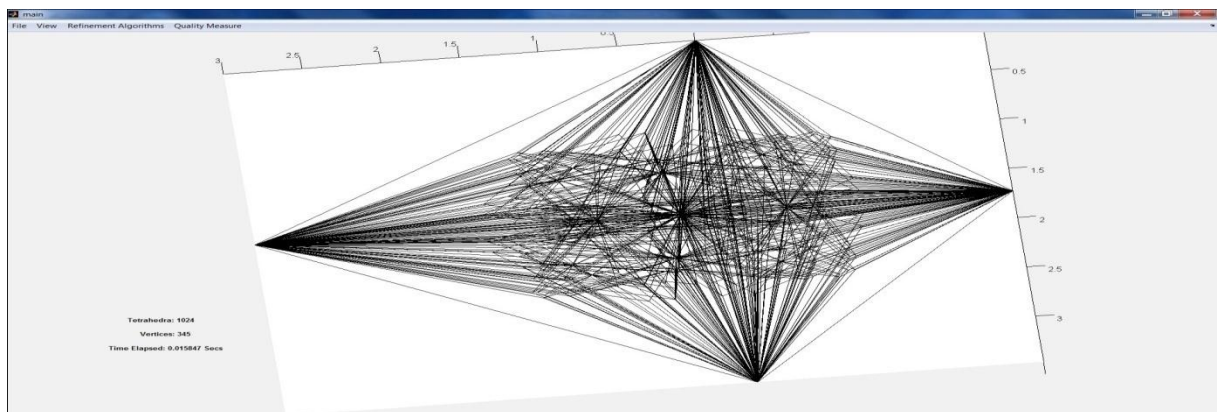
Anexo 3(d). *Cubo refinado por 4 niveles del algoritmo Circun esfera-Delaunay.*



Anexo 3(e). *Cubo refinado por 5 niveles del algoritmo Inesfera-Delaunay.*



Anexo 3(f). *Cubo refinado por 5 niveles del algoritmo Lado Mayor-Delaunay.*



Anexo 4. *Malla generada luego de 5 refinamientos globales por el algoritmo 4T-Baricentro partiendo de 1 tetraedro inicial sliver.*
