



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

TESELACIÓN DELAUNAY DE CUBOIDES PARA MALLAS MIXTAS 3D

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

DAVID GABRIEL CONTRERAS LOBOS

PROFESOR GUÍA:  
NANCY HITSCHFELD KAHLER

MIEMBROS DE LA COMISIÓN:  
MARÍA CECILIA RIVARA ZÚÑIGA  
PATRICIO INOSTROZA FAJARDIN

Este trabajo ha sido parcialmente financiado por Proyecto Fondecyt N°1120495

SANTIAGO DE CHILE  
ENERO 2013

## Resumen

El modelamiento de problemas que requieren una solución aplicando métodos numéricos como el método de volúmenes finitos necesita una malla geométrica (discretización compuesta de celdas simples tales como triángulos en 2D y tetraedros en 3D) que represente de la mejor forma posible al dominio a estudiar. Un método de generación de mallas geométricas basado en octrees genera, en su proceso de refinamiento, cuboides que presentan a lo más un punto extra en cada arista, conocidos como elementos 1-irregulares, los cuales deben ser teselados en elementos finales. Las mallas mixtas estudiadas son teselaciones de Delaunay compuestas de 7 elementos coesféricos (todos sus vértices están en la superficie de una esfera) finales: tetraedro, cuboide, prisma, pirámide, complemento de tetraedro, prisma deformado y complemento de tetraedro deformado.

Esta memoria se enfoca en el ámbito de teselaciones Delaunay sobre cuboides 1-irregulares para mallas mixtas en 3D con el objetivo de minimizar la cantidad de elementos usados en cada teselación, ya que esto afecta directamente el desempeño de un simulador. Se desarrollaron dos herramientas que facilitan el análisis de las teselaciones resultantes de cuboides 1-irregulares, generados por bisección (los puntos extra bisectan a las aristas) y generados por intersección (los puntos extra se ubican arbitrariamente). La primera herramienta desarrollada se encarga de la generación de datos y estadísticas que permiten analizar la teselación de cualquier cuboide 1-irregular y clasificar a qué tipo de poliedro corresponden sus elementos. Se implementó un algoritmo que permite encontrar una teselación Delaunay formada por elementos coesféricos sobre un conjunto de puntos arbitrarios, el cual fue aplicado posteriormente sobre los distintos cuboides estudiados. La segunda herramienta consiste en una aplicación GUI que permite la visualización de la teselación de cualquier cuboide 1-irregular generado por bisección especificando el tamaño del cuboide base, permitiendo interactuar independientemente con cada elemento final generado y visualizar su centroide y circuncentro.

El análisis de las teselaciones de los cuboides 1-irregulares arrojó la identificación de 31 elementos coesféricos adicionales a los 7 elementos previamente conocidos, 17 en cuboides generados por bisección y 14 exclusivos de cuboides generados por intersección. Sin embargo, los 7 elementos originales representan la gran mayoría del total de apariciones en todos los cuboides estudiados: más del 95 % en cuboides generados por bisección y cerca del 97 % en el caso de intersección, donde además, menos del 0,07 % corresponde a elementos encontrados exclusivamente en este caso. Al mismo tiempo, se realizó un análisis de las teselaciones de cuboides 1-irregulares generados por bisección bajo distintos criterios de tal manera de minimizar el número de elementos finales distintos permitiendo agregar arcos y caras interiores a la teselación obtenida. Por ejemplo, agregando caras interiores, cerca del 96,9 % del total de apariciones corresponde a los 7 elementos originales. Además, se identifican proporciones óptimas del cuboide 1-irregular base para obtener teselaciones generadas por bisección que permiten la aplicación del método de volúmenes finitos. Todas las teselaciones son apropiadas si el cuboide 1-irregular tiene una razón entre su lado más largo y su lado más corto menor o igual a  $\sqrt{2}$ .

## Agradecimientos

Quisiera agradecer a la profesora Nancy, por todos sus *hints* y sugerencias que ayudaron a hacer posible este trabajo. También a mi familia, amigos y compañeros, que estuvieron conmigo durante estos años. Gracias a todos.

Sí, soy de pocas palabras.

# Tabla de Contenido

<b>Resumen</b>	<b>i</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Aspectos Generales . . . . .	1
1.2. Motivación . . . . .	2
1.2.1. Configuraciones 1-irregulares generadas por bisección . . . . .	3
1.2.2. Configuraciones 1-irregulares generadas por intersección . . . . .	4
1.3. Objetivos . . . . .	5
1.3.1. Objetivo General . . . . .	5
1.3.2. Objetivos Específicos . . . . .	5
1.4. Contenido . . . . .	5
<b>2. Revisión Bibliográfica y Antecedentes</b>	<b>7</b>
2.1. Teselación de cuboides con puntos de Steiner . . . . .	7
2.1.1. Antecedentes sobre configuraciones 1-irregulares generadas por bisección . . . . .	7
2.1.2. Antecedentes sobre configuraciones 1-irregulares generadas por intersección . . . . .	8
2.2. Generalización de octrees modificados para modelamiento geométrico . . . . .	8
2.2.1. Elementos coesféricos base . . . . .	9
<b>3. Diseño de la Solución</b>	<b>13</b>
3.1. Ambiente de Desarrollo . . . . .	13
3.1.1. Qhull . . . . .	13
3.1.2. PStreams . . . . .	14
3.1.3. Qt . . . . .	14
3.1.4. QGLViewer . . . . .	15
3.2. Herramientas a desarrollar . . . . .	15
3.3. Estructura de Clases . . . . .	16
<b>4. Configuraciones 1-irregulares generadas por Bisección</b>	<b>18</b>
4.1. Consideraciones adicionales . . . . .	18
4.1.1. Puntos extra en las configuraciones . . . . .	18
4.1.2. Convenciones a usar y nomenclatura . . . . .	19
4.2. Generación de Datos . . . . .	21
4.2.1. Construyendo una configuración . . . . .	22
4.2.2. Algoritmo para encontrar la teselación de una configuración . . . . .	22
4.2.3. Identificando a los elementos de la partición . . . . .	26
4.3. Generación de Datos: Un caso de ejemplo . . . . .	27
4.3.1. Identificación de puntos de Steiner . . . . .	27

4.3.2.	Aplicación del Algoritmo para encontrar la teselación . . . . .	27
4.3.3.	Identificación de elementos finales . . . . .	29
4.4.	Visualizador . . . . .	30
4.4.1.	Implementación . . . . .	31
4.4.2.	Componentes . . . . .	31
4.5.	Validación . . . . .	35
4.5.1.	Determinación de Clases de Equivalencia . . . . .	35
4.5.2.	Otras pruebas y desempeño . . . . .	35
<b>5.</b>	<b>Configuraciones 1-irregulares generadas por Intersección</b>	<b>36</b>
5.1.	Consideraciones adicionales . . . . .	36
5.1.1.	Puntos extra en las configuraciones . . . . .	36
5.1.2.	Igualdad entre configuraciones . . . . .	37
5.2.	Revisión de la cantidad de configuraciones distintas . . . . .	38
5.3.	Generación de datos . . . . .	39
5.3.1.	Construyendo una configuración . . . . .	39
5.3.2.	Algoritmo para encontrar la teselación de una configuración . . . . .	39
5.3.3.	Identificando a los elementos de la partición . . . . .	40
5.4.	Desempeño . . . . .	40
<b>6.</b>	<b>Resultados</b>	<b>40</b>
6.1.	Elementos coesféricos adicionales identificados en configuraciones 1-irregulares generadas por bisección . . . . .	40
6.2.	Análisis de configuraciones 1-irregulares generadas por bisección . . . . .	46
6.2.1.	Caso A: Cuboide de dimensiones $a = b = c$ . . . . .	49
6.2.2.	Caso B: Cuboide de dimensiones $4a = 2b = c$ . . . . .	51
6.2.3.	Caso C: Cuboide de dimensiones $a\sqrt{2} = b = c$ . . . . .	52
6.3.	Calidad de las particiones generadas por bisección . . . . .	53
6.3.1.	Caso A: Cuboide de dimensiones $a = b = c$ . . . . .	53
6.3.2.	Caso B: Cuboide de dimensiones $4a = 2b = c$ . . . . .	54
6.3.3.	Caso C: Cuboide de dimensiones $a\sqrt{2} = b = c$ . . . . .	54
6.4.	Elementos coesféricos adicionales identificados en configuraciones 1-irregulares generadas por intersección . . . . .	54
6.5.	Resumen de configuraciones 1-irregulares generadas por intersección . . . . .	59
<b>7.</b>	<b>Conclusiones y Trabajo futuro</b>	<b>61</b>
	<b>Referencias</b>	<b>63</b>
<b>A.</b>	<b>Anexo: Diagrama de Clases</b>	<b>65</b>

<b>B. Anexo: Clases de equivalencia de configuraciones 1-irregulares generadas por bisección</b>	<b>66</b>
<b>C. Anexo: Ejemplos del Visualizador en funcionamiento</b>	<b>76</b>

## Índice de figuras

1.	Ejemplo de malla mixta sobre un cuboide y su diagrama de Voronoi . . . . .	2
2.	Ejemplo de configuraciones n-irregulares . . . . .	3
3.	División de un cuboide mediante un refinado por bisección de aristas . . . . .	3
4.	Cuboide 1-irregular generado mediante un refinado por bisección de aristas . . . . .	3
5.	División de un cuboide mediante un refinado por intersección . . . . .	4
6.	Cuboide 1-irregular generado mediante un refinado por intersección . . . . .	4
7.	Ejemplo de configuraciones pertenecientes a una misma clase de equivalencia . . . . .	7
8.	Descripción de un Cuboide . . . . .	9
9.	Descripción de un Tetraedro . . . . .	10
10.	Descripción de una Pirámide de base cuadrilátera . . . . .	10
11.	Descripción de un Prisma . . . . .	11
12.	Descripción de un Complemento de Tetraedro . . . . .	11
13.	Descripción de un Prisma deformado . . . . .	12
14.	Descripción de un Complemento de Tetraedro deformado . . . . .	12
15.	Diagrama de Clases simplificado . . . . .	17
16.	Distintas teselaciones posibles para un rectángulo en 2D con 4 puntos de Steiner . . . . .	18
17.	Criterios para agregar puntos adicionales en configuraciones 1-irregulares (bisección) . . . . .	19
18.	Identificación de las 19 posiciones posibles para ubicar puntos en configuraciones 1-irregulares de tipo bisección . . . . .	20
19.	Ejemplo de la relación entre el identificador de una configuración, su representación en binario y los puntos que la conforman . . . . .	21
20.	La configuración $C_{2780}$ y su representación . . . . .	28
21.	Visualizador de teselaciones y sus componentes . . . . .	32
22.	Opciones generales del Visualizador . . . . .	32
23.	Visualizador en funcionamiento . . . . .	34
24.	Criterios para agregar puntos adicionales en configuraciones 1-irregulares (intersección) . . . . .	37
25.	Igualdad entre configuraciones 1-irregulares (intersección) . . . . .	37
26.	Distintas formas de ubicar puntos de Steiner en ejes paralelos . . . . .	38
27.	Pirámide Pentagonal . . . . .	41
28.	Pirámide Hexagonal . . . . .	41
29.	Bipirámide Triangular . . . . .	42
30.	Bipirámide Cuadrilátera . . . . .	42
31.	Bipirámide Pentagonal . . . . .	42
32.	Bipirámide Hexagonal . . . . .	43
33.	Biprisma Triangular . . . . .	43
34.	Elemento genérico #1 . . . . .	43
35.	Elemento genérico #2 . . . . .	44

36.	Elemento genérico #3 . . . . .	44
37.	Elemento genérico #4 . . . . .	44
38.	Elemento genérico #5 . . . . .	45
39.	Elemento genérico #6 . . . . .	45
40.	Elemento genérico #7 . . . . .	45
41.	Elemento genérico #8 . . . . .	46
42.	Elemento genérico #9 . . . . .	46
43.	Elemento genérico #10 . . . . .	46
44.	Ejemplo de elemento coesférico separable directamente . . . . .	47
45.	Ejemplo de elemento coesférico separable agregando aristas internas . . . . .	48
46.	Elemento genérico #11 . . . . .	54
47.	Elemento genérico #12 . . . . .	55
48.	Elemento genérico #13 . . . . .	55
49.	Elemento genérico #14 . . . . .	55
50.	Elemento genérico #15 . . . . .	56
51.	Elemento genérico #16 . . . . .	56
52.	Elemento genérico #17 . . . . .	56
53.	Elemento genérico #18 . . . . .	57
54.	Elemento genérico #19 . . . . .	57
55.	Elemento genérico #20 . . . . .	57
56.	Elemento genérico #21 . . . . .	58
57.	Elemento genérico #22 . . . . .	58
58.	Elemento genérico #23 . . . . .	58
59.	Elemento genérico #24 . . . . .	59
60.	Diagrama de clases extendido . . . . .	65
61.	Visualizador con opciones por defecto . . . . .	76
62.	Visualizador en modo Wireframe . . . . .	76
63.	Visualizador en modo Wireframe, con centroides y circuncentros visibles . . . . .	77
64.	Visualizador en modo Wireframe, con centroides visibles, elementos desplazados y proporciones modificadas . . . . .	77
65.	Visualizador en modo Wireframe, con circuncentros visibles y proporciones modificadas . . . . .	78



# 1. Introducción

## 1.1. Aspectos Generales

En el contexto del modelamiento de problemas que requieren una solución específica aplicando métodos numéricos, como el método de volúmenes finitos, surge la necesidad de generar una discretización compuesta de celdas simples (elementos básicos) que represente de la mejor forma posible el dominio a estudiar. Esta discretización es conocida como malla geométrica y permite simular el comportamiento del objeto representado. Específicamente, para el método de volúmenes finitos se necesita una malla conformada por elementos convexos básicos: polígonos (en un dominio 2D) o poliedros (en el caso de un dominio 3D), los cuales deben satisfacer la condición de Delaunay<sup>1</sup> para así usar su estructura dual, el diagrama de Voronoi, cuyas regiones son utilizadas para calcular una solución aproximada. Los elementos básicos usados hasta ahora son triángulos y cuadriláteros en 2D y tetraedros, cuboides, prismas y pirámides en 3D.

Estas mallas se construyen sobre dominios en 2D o 3D descritos por conjuntos de puntos, polígonos o poliedros (dependiendo del caso). Uno de los métodos más usados para la generación de una malla en 2D está basado en quadrees, mientras que, análogamente, en 3D se encuentra el enfoque basado en octrees: el dominio es ajustado por un cuboide, el cual es repetidamente refinado dividiéndolo en cuboides de menor tamaño insertando puntos en sus aristas sucesivamente (puntos conocidos como puntos de Steiner), hasta lograr una representación lo suficientemente aproximada. Originalmente, este proceso se usaba para generar una malla de tetraedros, pero es posible permitir la inclusión de nuevos elementos (como prismas, pirámides y cuboides) durante el proceso de refinado, produciendo una malla denominada mixta[1][2]. La ventaja de usar mallas mixtas es que usando estos elementos se disminuye la cantidad de arcos necesarios, pues por ejemplo, un cuboide no es necesario dividirlo en tetraedros. Por otro lado, una desventaja es que las ecuaciones deben ser discretizadas usando distintos elementos.

La generación de mallas mixtas basadas en octrees genera naturalmente conjuntos de puntos co-esféricos. Para generar una malla mixta que satisface la condición de Delaunay, no es necesario dividir los cuboides en tetraedros; más aún, se pueden aceptar como elementos terminales todos los elementos co-esféricos que se generen, como se ve en la Figura 1, donde los elementos terminales en este ejemplo son 5 pirámides y 4 tetraedros. En general se restringe el número de elementos distintos para no complicar la solución numérica de las ecuaciones diferenciales parciales que modelan el problema. En particular, cuando se usa el método de volúmenes finitos, se requiere generar sólo los arcos de la malla que tengan una cara de Voronoi de área distinta de cero. Esto se logra si ninguna configuración de puntos co-esféricos, se divide en elementos más pequeños.

---

<sup>1</sup>Esto es, el circuncírculo (o circun-esfera en 3D) de cada elemento no contiene a ningún otro punto de la malla.

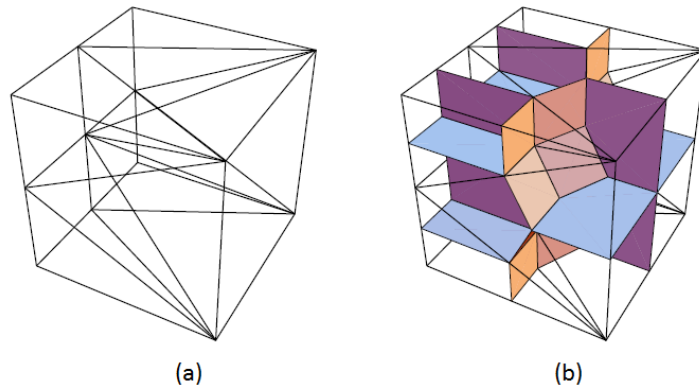


Figura 1: (a) Malla mixta sobre un cuboide que cumple la condición de Delaunay, (b) malla mixta y su diagrama de Voronoi asociado.

## 1.2. Motivación

El presente trabajo está inmerso en el contexto del proyecto Fondecyt 1120495[4], el cual se centra en el estudio y diseño de algoritmos y estructuras de datos para generaciones de mallas en  $2 \frac{1}{2}$  D y mallas mixtas de elementos en tres dimensiones. Estas mallas son de utilidad en varias aplicaciones de ingeniería, por ejemplo la construcción de mallas mixtas es apropiada para aplicar el método de volúmenes finitos. Uno de sus objetivos plantea mejorar el desempeño y alcance de un generador de mallas mixtas desarrollado en el Departamento de Ciencias de la Computación, Universidad de Chile. Este generador de mallas ha sido usado para la simulación de semiconductores y se quiere usar a futuro para modelar ciudades. Éste usa una estrategia basada en octrees modificados para modelar el dominio en donde las aristas de los cuboides contenidos pueden ser divididas en cualquier posición. Una vez representado el dominio, la mayor parte de los elementos se refinan bisectando sus aristas para alcanzar la densidad de puntos deseada. Después se refinan o insertan puntos de tal manera que todos los elementos hojas del octree sean de tipo 1-irregular (es decir, cada arista puede tener a lo más un punto de Steiner) o bien, elementos finales (no presentan puntos de Steiner). Algunos ejemplos de configuraciones n-irregulares se encuentran en la Figura 2.

Dado que el dominio a estudiar se representa por medio de un octree, sus elementos 1-irregulares serán siempre cuboides, dentro de los cuales se distinguen dos tipos básicos de configuración dependiendo de la posición de los puntos de Steiner, los cuales se describen a continuación.

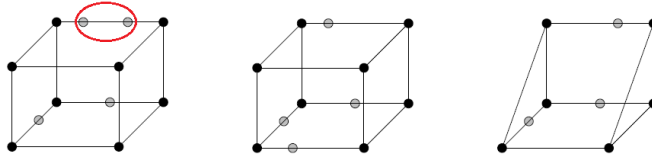


Figura 2: A la izquierda: cuboide 2-irregular, pues presenta dos puntos de Steiner en una de sus aristas (marcados en rojo). Al centro y a la derecha: elementos 1-irregulares (cuboide y prisma, respectivamente).

### 1.2.1. Configuraciones 1-irregulares generadas por bisección

En este tipo de configuración, los puntos de Steiner insertados en el refinamiento siempre estarán ubicados en los puntos medios de cada figura, lo que generará divisiones entre dos, cuatro u ocho cuboides de menor tamaño como se ve en la Figura 3.

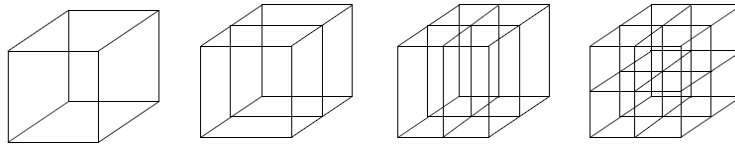


Figura 3: Cuboide y divisiones en dos, cuatro y ocho elementos mediante un refinado por bisección de aristas.

El refinamiento anterior producirá cuboides vecinos con puntos de Steiner ubicados en los puntos medios de las aristas. Los cuboides 1-irregulares son cuboides de mayor tamaño vecinos a los ya refinados, como se aprecia en la Figura 4.

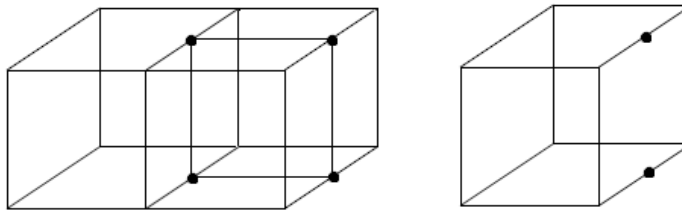


Figura 4: A la izquierda, el refinado por bisección de uno de los cuboides producirá que su vecino se vuelva un elemento 1-irregular como el cuboide de la derecha.

Existen  $2^{12}$  configuraciones posibles generadas de esta forma. Notar un cuboide tiene 12 aristas,

y en cada una de ellas puede o no existir un punto de Steiner (dos posibilidades). Como la existencia de cada punto de Steiner es completamente independiente de los demás se obtiene ese resultado.

### 1.2.2. Configuraciones 1-irregulares generadas por intersección

Este caso es similar al anterior, pero los puntos de Steiner no necesariamente se ubicarán en los puntos medios de las aristas. En general no hay restricciones sobre la ubicación de los puntos, excepto el hecho de que aristas paralelas deben ser divididas en la misma posición relativa a sus extremos para garantizar la generación de cuboides y no cualquier otro poliedro. La Figura 5 muestra un ejemplo de refinamiento por intersección.

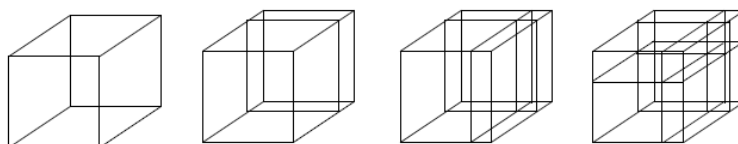


Figura 5: Cuboide y ejemplo de divisiones posibles mediante un refinado por intersección.

El refinamiento anterior producirá cuboides vecinos con puntos de Steiner ubicados en alguna sección de las aristas. Los cuboides 1-irregulares son cuboides de mayor tamaño vecinos a los ya refinados, como se aprecia en la Figura 6.

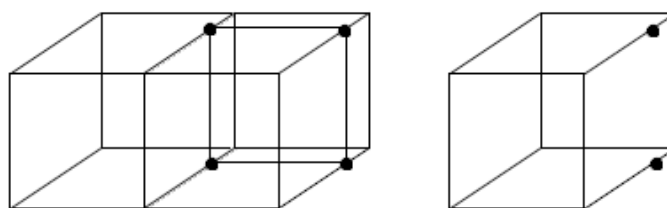


Figura 6: A la izquierda, el refinado por intersección de uno de los cuboides producirá que su vecino se vuelva un elemento 1-irregular como el cuboide de la derecha.

## 1.3. Objetivos

### 1.3.1. Objetivo General

- Desarrollar un conjunto de herramientas que permitan generar, visualizar, contar y analizar las teselaciones de Delaunay óptimas para cualquier cuboide 1-irregular tanto para refinamiento por bisección como intersección.

### 1.3.2. Objetivos Específicos

- Diseñar e implementar un algoritmo para encontrar la partición óptima de todos los elementos 1-irregular.
- Visualizar las teselaciones generadas por bisección interactivamente, permitiendo examinar tanto en conjunto como separadamente los elementos que la componen, visualizando centroides y puntos de Voronoi (circumcentros de cada elemento).
- Identificar el número de elementos co-esféricos de formas distintas necesarios para teselar todos los elementos 1-irregular generados por bisección, tanto para un cubo como para un cuboide, bajo distintos criterios.
- Identificar el número de elementos co-esféricos de formas distintas necesarios para teselar todos los elementos 1-irregular generados por intersección, tanto para un cubo como para un cuboide.
- Generar estadísticas asociadas a la partición, como cuántas veces se utilizan ciertos elementos específicos, y cuantificar cuántas de las teselaciones pueden ser usadas para el método de volúmenes finitos sin tener que agregar nuevos vértices. Además, analizar las teselaciones de elementos 1-irregulares generadas por bisección bajo distintos criterios que afectan la cantidad de elementos coesféricos finales distintos.

## 1.4. Contenido

El contenido de este trabajo se agrupa como se especifica a continuación:

- El Capítulo 2 contiene una revisión bibliográfica de dos publicaciones y se presenta una serie de antecedentes relativos a configuraciones 1-irregulares y generación de mallas geométricas.

- En el Capítulo 3 se presenta el diseño de la solución, describiendo el ambiente de desarrollo, las herramientas a implementar y la estructura de clases.
- El Capítulo 4 está dedicado al estudio de las configuraciones 1-irregulares generadas por bisección, explicando el proceso de generación de datos y la implementación de una herramienta de visualización.
- Asimismo, el Capítulo 5 trata sobre las configuraciones 1-irregulares generadas por intersección, mostrando una forma de calcular la cantidad de configuraciones distintas y el proceso de generación de datos.
- El Capítulo 6 contiene los resultados obtenidos en el contexto de este trabajo: el detalle de elementos coesféricos encontrados, resumen de los distintos tipos de configuraciones 1-irregulares y un análisis de la calidad de las teselaciones generadas por bisección.
- Finalmente, el Capítulo 7 presenta las conclusiones del trabajo realizado y propuestas sobre el posible trabajo futuro.

## 2. Revisión Bibliográfica y Antecedentes

En esta sección se presentan definiciones de conceptos clave para entender el contexto de este trabajo, además de análisis y resultados de trabajos previos relacionados.

### 2.1. Teselación de cuboides con puntos de Steiner

En [3] se presenta un estudio de los diferentes cuboides 1-irregulares generados en mallas construidas usando un enfoque de octrees modificados, haciendo distinción entre configuraciones generadas por bisección o intersección.

#### 2.1.1. Antecedentes sobre configuraciones 1-irregulares generadas por bisección

Dentro de los antecedentes relevantes sobre las configuraciones generadas por bisección presentados en esta publicación se encuentran:

- La determinación del número de clases de equivalencia para las configuraciones dentro de un cuboide 1-irregular. Dos configuraciones se consideran equivalentes si una puede ser transformada en la otra, usando operaciones de rotación y reflexión<sup>2</sup>. Las configuraciones de la Figura 7 son equivalentes bajo este criterio y pertenecen a la misma clase. Se identifican 144 patrones<sup>3</sup> distintos a partir de las 4096 configuraciones posibles.

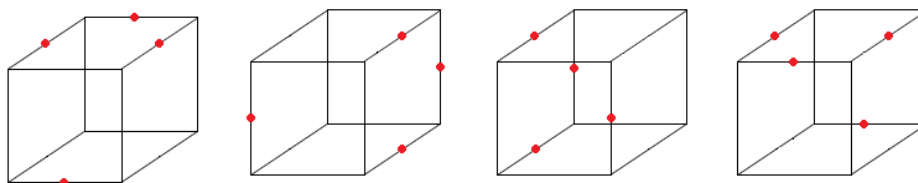


Figura 7: Ejemplo de configuraciones pertenecientes a una misma clase de equivalencia.

- Para cada patrón, existe una cota superior de 15 posibles teselaciones, las cuales varían dependiendo de las proporciones del cuboide dadas por el “edge length ratio”, es decir, el valor resultante de la división de la arista más larga con la arista más corta de cada cara.

---

<sup>2</sup>Esta relación de equivalencia (reflexiva, simétrica y transitiva) permite identificar las distintas clases de equivalencia.

<sup>3</sup>Un patrón se define como un cuboide 1-irregular representante de una clase de equivalencia.

### 2.1.2. Antecedentes sobre configuraciones 1-irregulares generadas por intersección

Dentro de los antecedentes relevantes sobre las configuraciones generadas por intersección presentados en esta publicación se encuentran:

- El establecimiento de una noción de igualdad entre configuraciones 1-irregulares bajo este enfoque, que considera que una configuración  $i_1$  es igual a una configuración  $i_2$  si la posición relativa de los puntos de Steiner ubicados en los ejes paralelos de  $i_1$  e  $i_2$  es la misma con respecto a una configuración normalizada.
- Una formula explícita para calcular el número de configuraciones 1-irregulares en  $d$  dimensiones diferentes, considerando la noción de igualdad definida en el punto anterior. Esta fórmula está dada por la expresión  $N = n^d$ , donde  $n$  representa la cantidad de formas de ubicar hasta  $2^{d-1}$  puntos de Steiner en uno de los conjuntos de  $2^{d-1}$  aristas paralelas del elemento 1-irregular y se define por la expresión:

$$n = 1 + \sum_{i=1}^{2^{d-1}} \binom{2^{d-1}}{i} \frac{(2i-1)!}{2^{i-1}(i-1)!}$$

En 3 dimensiones, el total de configuraciones 1-irregulares generadas por intersección está dada por<sup>4</sup>  $N = 188^3 = 6,644,672$ .

### 2.2. Generalización de octrees modificados para modelamiento geométrico

En [2], se estudian alternativas en la generación de mallas usando octrees modificados. Una implementación de esta técnica es incluida en el Generador de Mallas  $\Omega$ . Este generador usa distintos tipos de “macro-elementos”<sup>5</sup> usados para representar la geometría a refinar y se analiza cuál es el conjunto de elementos finales encontrados en la malla resultante. El algoritmo básico para generar este tipo de mallas es el siguiente:

1. Generar una macro-malla que cubre a la geometría del objeto a modelar exactamente.
2. Hacer un proceso de refinamiento debido a variaciones en algunos valores internos y ciertos parámetros geométricos.
3. Generar una malla final de Delaunay sobre este resultado:
  - Hacer que la malla sea 1-irregular.

---

<sup>4</sup>En el paper se menciona que el total es  $187^3$ . Esto corresponde a un error.

<sup>5</sup>En el contexto de este trabajo el único macro-elemento estudiado es un cuboide.



- Teselar los elementos 1-irregulares si es posible, de lo contrario refinarlos o insertar puntos en posiciones estratégicas.

4. Guardar información requerida.

Actualmente, el Generador de Mallas  $\Omega$  tiene teselaciones prealmacenadas para 25 clases de equivalencia, lo cual permite encontrar en  $O(1)$  la teselación correspondiente. Cuando se encuentra un elemento 1-irregular que no tiene su partición almacenada, se insertan puntos estratégicamente de tal manera de lograr generar sólo configuraciones 1-irregulares conocidas. En algunos casos, este proceso no converge tan rápidamente como se desea.

**2.2.1. Elementos coesféricos base**

Uno de los resultados presentados de importancia para este trabajo consiste en la determinación de un conjunto de elementos finales que resuelve aproximadamente el 80 % de las configuraciones 1-irregulares cuyo macro-elemento es un cuboide. Dicho conjunto de 7 elementos finales corresponde a los posibles elementos geométricos resultantes de cada teselación, los cuales se utilizan como conjunto base de elementos coesféricos reconocibles. La descripción de estos elementos se detalla a continuación:

- **Cuboide:** Elemento formado por 6 caras rectangulares, 12 aristas y 8 vértices, como se muestra en la Figura 8.

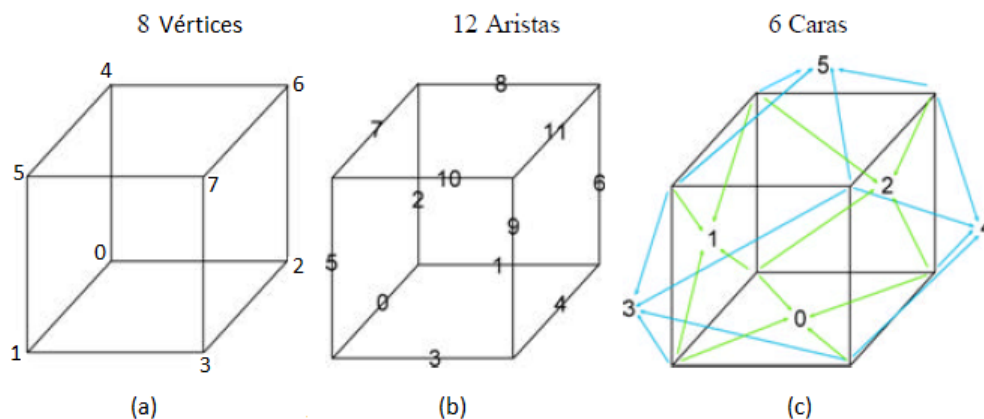


Figura 8: Descripción de un Cuboide: (a) identificación de vértices, (b) identificación de aristas, (c) identificación de caras.

- **Tetraedro:** Elemento formado por 4 caras triangulares, 6 aristas y 4 vértices, como se muestra en la Figura 9.

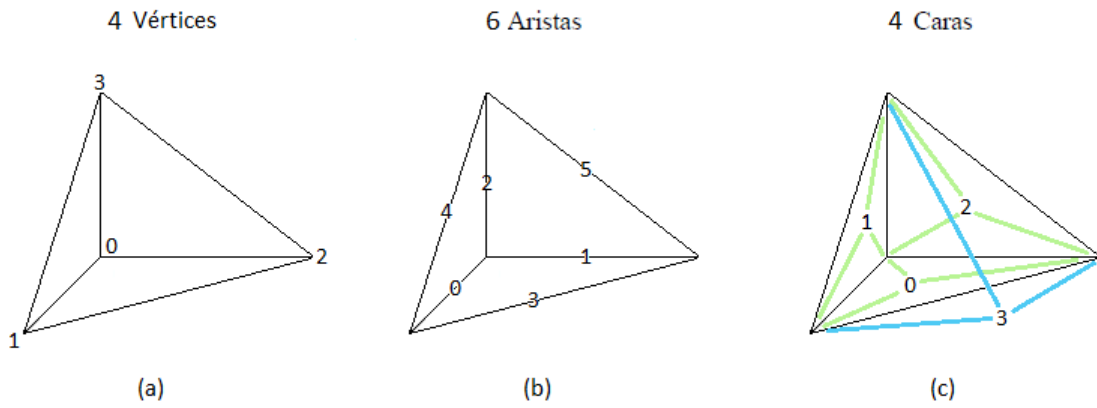


Figura 9: Descripción de un Tetraedro: (a) identificación de vértices, (b) identificación de aristas, (c) identificación de caras.

- **Pirámide de base cuadrilátera:** Elemento formado por 3 caras triangulares y 1 cara cuadrilátera, 8 aristas y 5 vértices, como se muestra en la Figura 10.

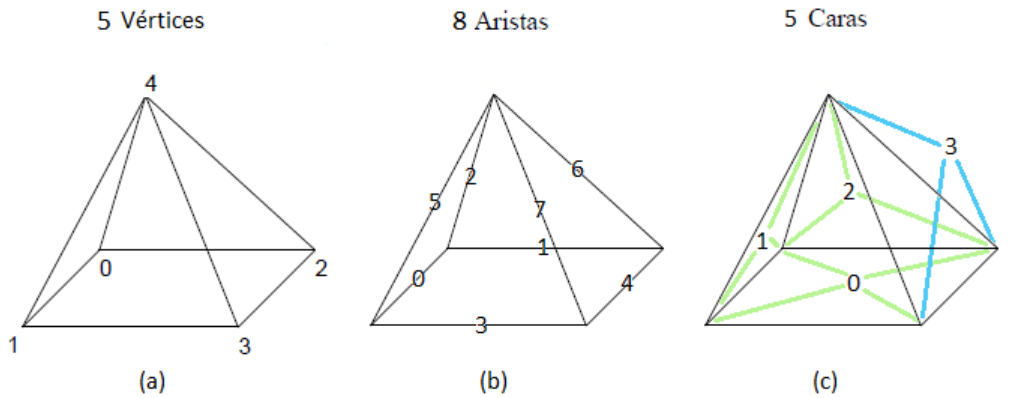


Figura 10: Descripción de una Pirámide de base cuadrilátera: (a) identificación de vértices, (b) identificación de aristas, (c) identificación de caras.

- **Prisma:** Elemento formado por 2 caras triangulares y 3 caras rectangulares, 9 aristas y 6 vértices, como se muestra en la Figura 11. Específicamente, corresponde a un prisma triangular.

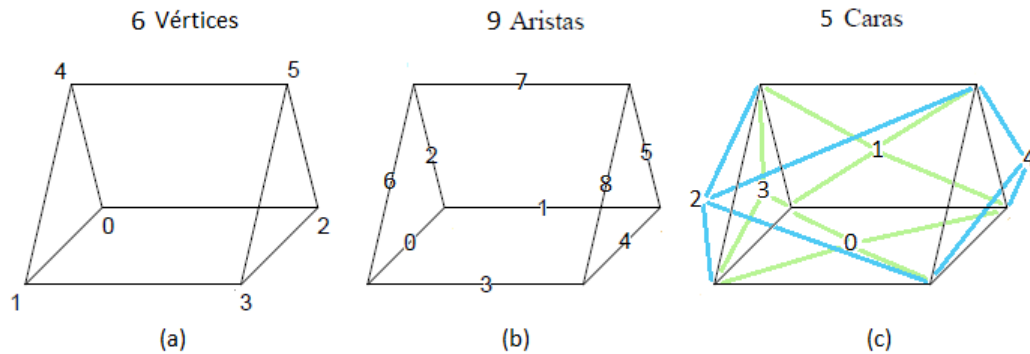


Figura 11: Descripción de un Prisma: (a) identificación de vértices, (b) identificación de aristas, (c) identificación de caras.

- **Complemento de Tetraedro:** Elemento formado por 4 caras triangulares (3 de ellas triángulos rectángulos) y 3 caras rectangulares, 12 aristas y 7 vértices, como se muestra en la Figura 12. Recibe su nombre ya que este elemento es el complemento de un tetraedro trirectangular, considerando un cuboide como universo.

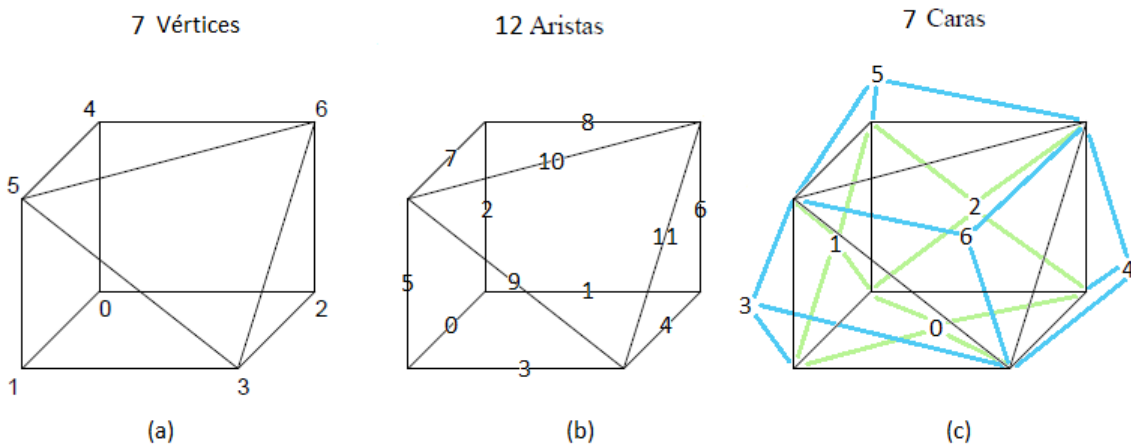


Figura 12: Descripción de un Complemento de Tetraedro: (a) identificación de vértices, (b) identificación de aristas, (c) identificación de caras.

- **Prisma deformado:** Elemento formado por 6 caras triangulares y una rectangular, 11 aristas y 6 vértices, los cuales quedan identificados como se muestra en la Figura 13. Recibe su nombre debido al hecho de que posee la misma cantidad de vértices que el elemento Prisma.

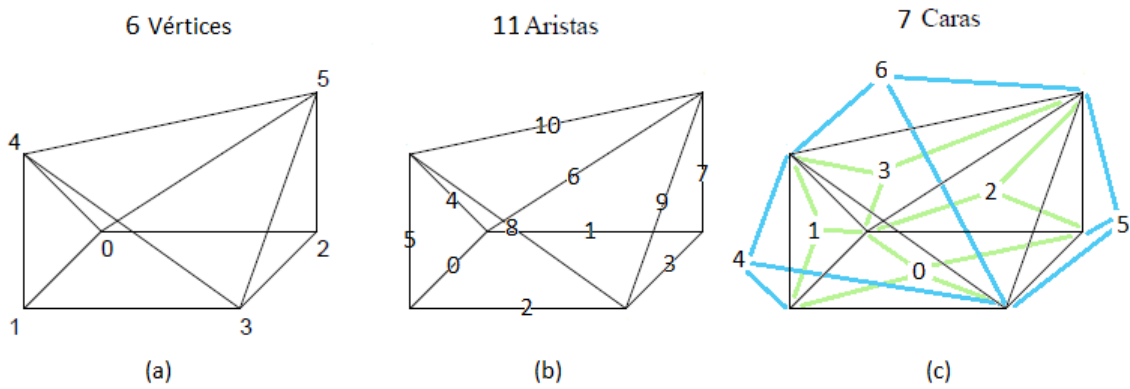


Figura 13: Descripción de un Prisma deformado: (a) identificación de vértices, (b) identificación de aristas, (c) identificación de caras.

- **Complemento de Tetraedro deformado:** Elemento formado por 8 caras triangulares y una rectangular, 14 aristas y 7 vértices, como se muestra en la Figura 14. Recibe su nombre por el hecho de tener la misma cantidad de vértices que el elemento Complemento de Tetraedro.

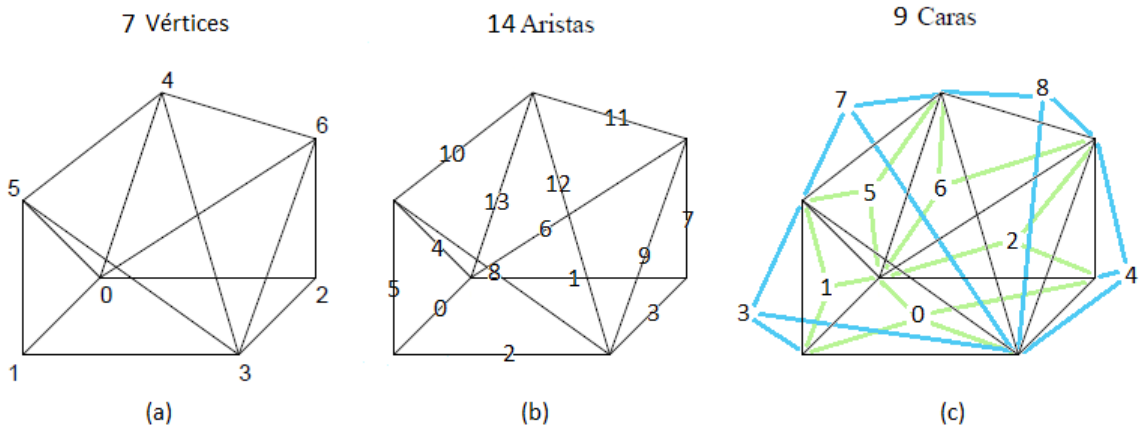


Figura 14: Descripción de un Complemento de Tetraedro deformado: (a) identificación de vértices, (b) identificación de aristas, (c) identificación de caras.

## 3. Diseño de la Solución

En esta sección, se presenta el diseño de la solución a construir, describiendo el ambiente de desarrollo, definiendo las herramientas a implementar y la estructura de clases que corresponde al esqueleto de la solución.

### 3.1. Ambiente de Desarrollo

La solución a implementar se construye en el lenguaje C++, siendo el motivo principal para esto la disponibilidad de librerías y herramientas útiles para desarrollar, las cuales se describen en las secciones siguientes.

#### 3.1.1. Qhull

Qhull[5] es una herramienta que permite computar, entre otros, la cerradura convexa, el diagrama de Voronoi y la teselación de Delaunay de un conjunto de puntos en 2D, 3D o superior. En particular para este trabajo, esta herramienta se utiliza para calcular teselaciones de Delaunay y cerraduras convexas de conjuntos de puntos en 3D (correspondientes a configuraciones 1-irregulares, y subconjuntos de éstas).

Entre sus características se encuentran el hecho de que Qhull puede ser ejecutado por medio de un comando, lo cual permite dentro de un entorno C/C++ interactuar con esta herramienta con, por ejemplo, una llamada a `system`, o `popen/pclose`.

Qhull debe recibir como input, ya sea por entrada estándar o especificando un archivo a leer como argumento, la descripción de los puntos en el siguiente formato:

- La primera línea indica la dimensión  $D$  del espacio en donde se trabaja (3 en este caso).
- La segunda línea especifica el número  $N$  de puntos del conjunto de entrada.
- Las  $N$  líneas restantes describen las  $D$  coordenadas de cada punto.

Adicionalmente, se deben especificar como argumentos de Qhull las opciones aplicables a la llamada. Las opciones utilizadas en el contexto de esta memoria se describen a continuación:

- `d`: Especifica que sobre los puntos de entrada se debe calcular la teselación de Delaunay (por defecto se calcula la cerradura convexa).

- Qz: Agrega puntos adicionales (ubicados en “el infinito”) al conjunto de entrada, permitiendo el cálculo de una teselación de Delaunay única en el caso de existir subconjuntos de 5 o más puntos coesféricos.
- i: Imprime en la salida estándar los vértices correspondientes a cada elemento encontrado (caras en el caso de la cerradura convexa y tetraedros en el caso de la teselación de Delaunay).

### 3.1.2. PStreams

PStreams[6] permite, dentro de una aplicación C++, la transferencia de datos entre programas de forma similar a como se realiza con las funciones de C, `popen` y `pclose`. La principal diferencia es que PStreams utiliza los `iostreams` propios de C++ para manejar redirecciones de entrada y salida estándar en vez de la librería `stdio` de C. Esto permite expresar las interacciones entre programas con exactamente la misma sintáxis que utilizan las funciones pertenecientes a la librería `iostream`, lo que ayuda a que el código sea más sencillo de leer y entender, y se estandariza el código en estilo C++, en vez de mezclarlo con funcionalidad de C.

Para ejemplificar lo anterior, se muestra a continuación un segmento de código que expresa el modo de uso de esta librería:

```
redi::pstream command("qhull d Qz i"); // Inicializa un stream bidireccional
                                        // con el comando de qhull especificado

command << "datos" << std::endl;      // Agrega datos vía entrada estándar

command << redi::eof;                 // Envía un carácter de fin de archivo

string var;
command >> var;                       // Obtiene datos desde la salida estándar

getline(command, var);                // Obtiene una línea completa desde la
                                        // salida estándar
```

### 3.1.3. Qt

Qt[7] es un framework de desarrollo multiplataforma orientado a la creación de aplicaciones con interfaz gráfica de usuario.

Incluye un entorno de desarrollo (IDE) llamado Qt Creator, el cual permite el desarrollo de aplicaciones integrando fácilmente las prestaciones ofrecidas por Qt.

Además dentro del framework se incluye la herramienta Qt Designer, la cual permite diseñar la estructura general de la interfaz de una aplicación gráfica en un formato WYSIWYG (“What you see is what you get” - “lo que ves es lo que obtienes”).

Con respecto a este trabajo, la implementación fue desarrollada íntegramente dentro de proyectos de Qt Creator.

#### **3.1.4. QGLViewer**

QGLViewer[8] es una librería implementada en C++ y basada en Qt que permite la creación de visualizadores 3D de forma sencilla. Esta herramienta es multiplataforma, pudiendo ser utilizada en ambientes Unix, Windows y MacOS.

Una de sus componentes principales es el objeto del mismo nombre QGLViewer, el cual representa a un visualizador que incluye funcionalidades como el control de cámara mediante el mouse (rotación, traslación y zoom in/out) implementadas por defecto, lo que facilita el trabajo a la hora de diseñar una forma de interactuar con los elementos representados. Además es fácil de extender, lo cual resulta ser una característica muy importante en el contexto de este trabajo.

Otra de sus componentes relevantes para la implementación de este trabajo son los objetos ManipulatedFrame, los cuales permiten definir distintos “sistemas de coordenadas” dentro de un entorno de dibujo, permitiendo la asociación entre estos sistemas de coordenadas y conjuntos de elementos dibujados para interactuar con ellos independientemente del resto la escena.

### **3.2. Herramientas a desarrollar**

Para lograr los objetivos propuestos se plantea una solución que consta de dos herramientas principales: un generador de datos y un visualizador. En términos generales, los roles y funcionalidades de cada uno se presentan a continuación:

- El módulo encargado de generación de datos contiene funciones necesarias para construir y analizar configuraciones 1-irregulares generadas tanto por bisección como intersección. Este módulo es construido como un objeto, el cual puede ser instanciado posteriormente por el

visualizador. Adicionalmente, puede ejecutarse en forma individual como una aplicación de consola para obtener resultados particulares predefinidos:

- Procesar una configuración 1-irregular generada por bisección especificando el tamaño del cuboide base y el identificador que la representa. Retorna la descripción de los elementos que componen la teselación de la configuración indicada.
  - Procesar todas las configuraciones 1-irregulares generadas por bisección, especificando el tamaño del cuboide base. Los resultados de estas configuraciones (teselaciones), pueden ser almacenadas en archivos de texto (indicando que así se desea). Además, se genera un reporte con la especificación de los elementos finales y la frecuencia de éstos.
  - Procesar todas las configuraciones 1-irregulares generadas por intersección dentro de un cuboide de lados iguales. Dada la cantidad de configuraciones posibles, no es factible almacenar los resultados para cada configuración en particular en archivos de texto, por lo que este proceso sólo genera un reporte con la especificación de los elementos finales y la frecuencia de éstos. La ejecución de este proceso puede tomar varias horas.
- El módulo visualizador corresponde a una aplicación con interfaz gráfica de usuario (GUI) que permite una visualización interactiva de configuraciones 1-irregulares generadas por bisección. Más detalles de este módulo se encuentran en la sección 4.4.

En conjunto con estas herramientas se cuenta con una estructura de clases que funciona como el esqueleto de la implementación, estructura que se describe en la siguiente sección.

### 3.3. Estructura de Clases

En esta sección se analiza la estructura de clases a utilizar para complementar a las herramientas que componen la implementación correspondiente a este trabajo. La Figura 15 muestra un diagrama de clases simplificado donde sólo se despliega información relacionada al nombre de las clases y las relaciones entre ellas. Un diagrama más detallado se encuentra en el Anexo A. La funcionalidad general de cada una de las clases se describe a continuación:

- **Point:** Representa a un vértice 3D cuyas coordenadas son valores de punto flotante.
- **Points:** Un contenedor para elementos de tipo Point.
- **Face:** Un polígono representado por un conjunto de puntos ordenados, lo cual define implícitamente a sus aristas.
- **Faces:** Análogo al objeto Points, corresponde a un contenedor para elementos de tipo Face.



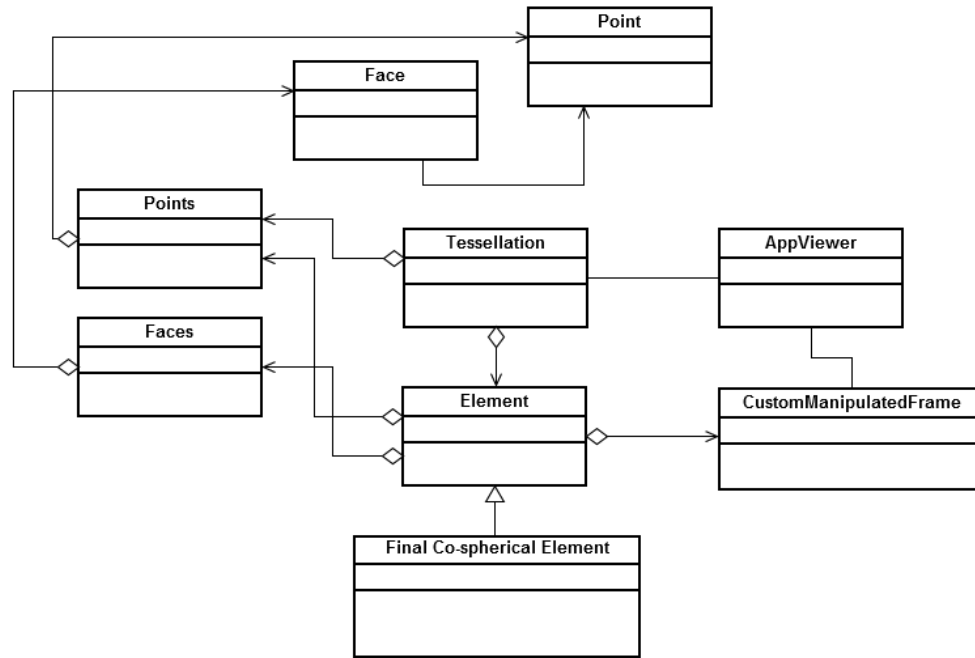


Figura 15: Diagrama de Clases simplificado.

- **CustomManipulatedFrame:** Subclase del objeto ManipulatedFrame de la librería QGLViewer. En esta clase se redefine el método que permite detectar bajo qué condiciones se debe activar el contexto asociado a este objeto.
- **Element:** Representa a un elemento geométrico genérico. Definido por una representación de sus caras y vértices (e implícitamente sus aristas). Contiene además un objeto CustomManipulatedFrame, lo que permite asociar este elemento a un contexto de dibujo independiente.
- **Tessellation:** Representa a una configuración 1-irregular y su teselación. Contiene un conjunto de objetos Element que representa a una teselación particular, además de un puntero a un objeto Points con los puntos que describen a esta configuración.
- **AppViewer:** Subclase del objeto QGLViewer de la librería del mismo nombre que personaliza el comportamiento de la visualización. Para detalles de su implementación consultar la sección 4.4.1.
- **Final Co-spherical Element:** Representa a un elemento geométrico definido. Subclase de Element. Esta clase en sí no existe, si no que más bien se presenta en el diagrama para ejemplificar el hecho de que los elementos definidos están representados por subclases de Element. Inicialmente existen 7 de estas clases, asociadas a los elementos coesféricos finales presentados en la sección 2.2.1.

## 4. Configuraciones 1-irregulares generadas por Bisección

Esta sección está enfocada al estudio de las configuraciones 1-irregulares generadas por bisección, explicando algunas consideraciones adicionales a tener en cuenta, el proceso de generación de datos y la implementación de una herramienta de visualización.

### 4.1. Consideraciones adicionales

#### 4.1.1. Puntos extra en las configuraciones

Como ya se mencionó anteriormente, las configuraciones 1-irregulares generadas por bisección presentan sus puntos de Steiner en los puntos medios de las aristas respectivas, dando como resultado 12 posiciones posibles en un cuboide. Una consideración adicional es la de agregar puntos extra a las configuraciones bajo dos criterios especiales con el fin de mejorar la teselación resultante. En la Figura 16 se aprecia un ejemplo en 2D de por qué el resultado mejora: sin agregar un punto extra, la teselación resultante depende de la relación de aspecto entre los lados del rectángulo. Sin embargo, agregando un punto adicional en el centro de la cara se asegura la condición de Delaunay con una teselación única, independiente de la forma del rectángulo, que además disminuye la cantidad de aristas y polígonos resultantes al costo de un vértice adicional.

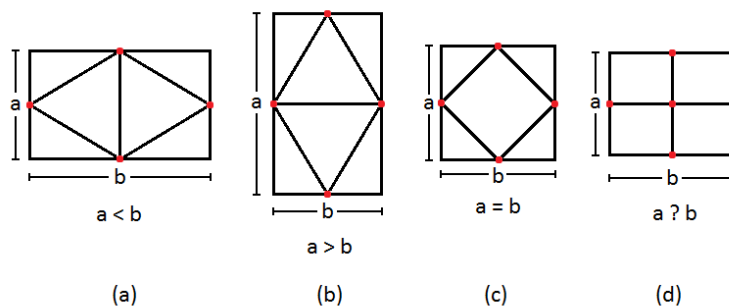


Figura 16: Distintas teselaciones posibles para un rectángulo en 2D con 4 puntos de Steiner: (a), (b) y (c) muestran las distintas formas de teselar un rectángulo, las cuales dependen de la relación entre sus lados  $a$  y  $b$ . (d) Agregando un punto al centro del rectángulo se obtiene esta única partición independiente de la relación entre  $a$  y  $b$ .

Es posible extrapolar esta idea a un espacio en 3D, lo cual da como resultado los siguientes criterios para agregar puntos adicionales:

- Si las 4 aristas que conforman una de las caras del cuboide tienen un punto de Steiner se agrega un punto adicional en el centro de la cara.

- Si se ha agregado un punto extra en todas sus caras, entonces se incluye otro punto adicional al centro del cuboide. Notar que la existencia de un punto en cada una de las aristas del cuboide ocurre en sólo un caso.

En la Figura 17 se encuentran ejemplos correspondientes a los dos criterios anteriormente descritos.

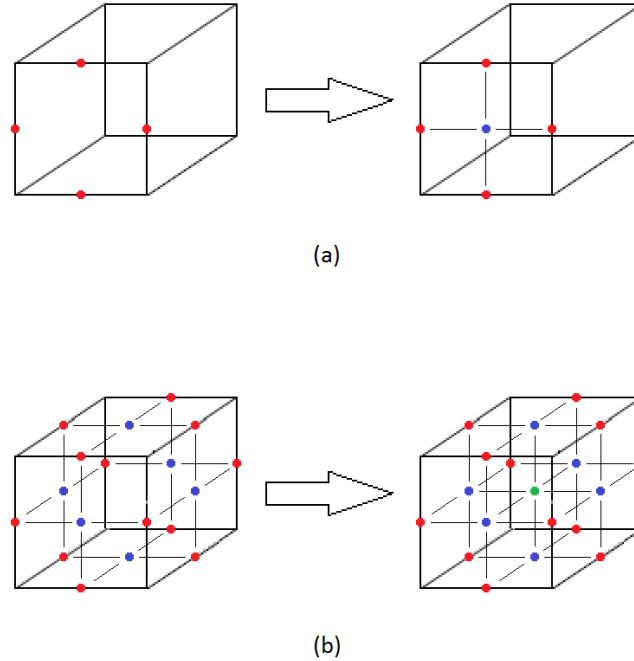


Figura 17: Criterios para agregar puntos adicionales en configuraciones 1-irregulares: (a) Punto extra en una cara. (b) Punto extra al centro del cuboide.

Estos criterios aumentan la cantidad de posiciones posibles de los puntos a 19 (12 en las aristas, 6 en las caras y 1 en el centro), sin embargo, la cantidad de configuraciones totales no se ve afectada, ya que estas posiciones extra son dependientes de las 12 originales.

#### 4.1.2. Convenciones a usar y nomenclatura

Las 19 posiciones posibles para ubicar puntos pueden ser identificadas por un número  $i \in [0, 18]$  de la forma como muestra la Figura 18

Asimismo, cada una de las  $2^{12}$  configuraciones puede ser identificada por un número  $k \in [0, 4095]$ , el cual está fuertemente relacionado con los identificadores  $i$  de los puntos de Steiner que la confor-

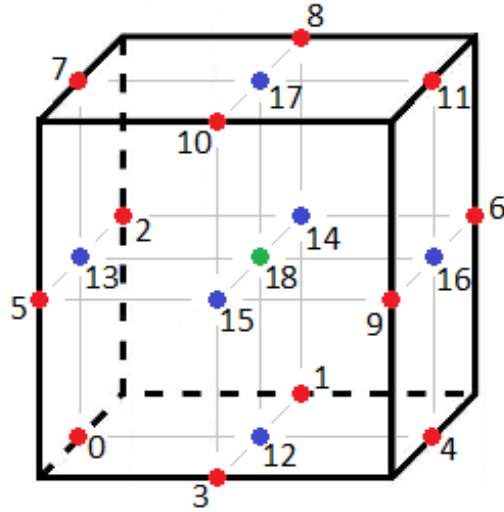


Figura 18: Las 19 posiciones posibles para ubicar puntos en configuraciones 1-irregulares generadas por bisección.

man. La forma de saber qué número  $k$  le corresponde a una configuración dada se hace mediante el siguiente proceso:

- Dada una configuración, se toman los identificadores  $i$  de todos los puntos de Steiner presentes en sus aristas, sin considerar ninguno de los puntos en posiciones extra, en caso de existir.
- Para cada valor  $i$ , se calcula el valor de  $2^i$ . La suma de todos estas potencias corresponde al valor de  $k$  que identifica a esta configuración.

El proceso inverso, es decir, dada una configuración identificada por un número  $k$  encontrar los valores  $i$  que la conforman, es igualmente sencillo:

- Expresar  $k$  en su representación binaria y luego determinar las posiciones de todos los valores 1 en esta representación, contando de derecha a izquierda y comenzando de 0 como la primera posición.
- Las posiciones encontradas corresponden a los valores  $i$  buscados. Notar que esto es cierto pues la representación en binario entrega las potencias de 2 cuya suma es  $k$ .

En la Figura 19 se muestran dos ejemplos de la relación entre los identificadores de una configuración y los identificadores de los puntos de Steiner que la conforman.

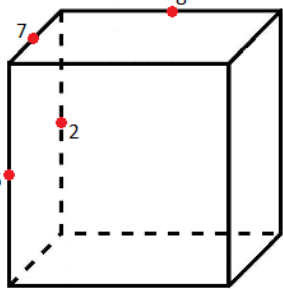
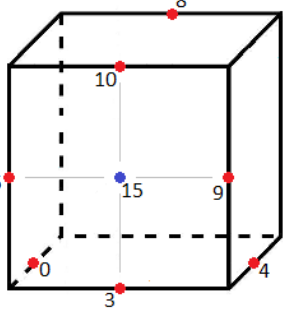
Identificador	Representaciones	Configuración
420	$2^2 + 2^5 + 2^7 + 2^8$ $000110100100_2$	
1849	$2^0 + 2^3 + 2^4 + 2^5 + 2^8 + 2^9 + 2^{10}$ $011100111001_2$	

Figura 19: Relación entre el identificador de una configuración, su representación en binario y los puntos que la conforman. Notar que en el segundo caso, el punto extra (15) no tiene influencia en el identificador que representa esta configuración.

En lo que sigue, se referirá a una configuración identificada por un número  $k$  como  $C_k$  y a un punto de Steiner identificado por un número  $i$  como  $P_i$ .

#### 4.2. Generación de Datos

Para obtener datos asociados a este tipo de configuraciones se implementa un conjunto de funciones dentro de la herramienta de generación de datos, enfocadas en procesar configuraciones y obtener resultados específicos, como la descripción de los poliedros que conforman cada configuración.

El procesamiento de una configuración en específico consta de los siguientes pasos:

- Construir una representación de la configuración, indicando las coordenadas de los vértices del cuboide base y de los puntos de Steiner.
- Encontrar una teselación de Delaunay compuesta por tetraedros para esta configuración.
- Determinar cuáles de los tetraedros encontrados se pueden unir para formar elementos coesféricos finales.
- Identificar a qué poliedro corresponde cada elemento coesférico final.

En las secciones siguientes se explica detalladamente los pasos de este proceso.

#### 4.2.1. Construyendo una configuración

Para encontrar la teselación de una configuración primero se debe encontrar el conjunto de puntos que la representa, para esto se crea una función que, dado un identificador de configuración (valor entero entre 0 y 4095) y el tamaño del cuboide base (alto, largo y ancho), retorna la descripción de la configuración contenida en un string que sigue el formato específico para ser utilizado como input de Qhull<sup>6</sup>. Como ya se mencionó en la sección 4.1.2, es posible obtener los puntos de Steiner que componen una configuración a partir de su identificador, exceptuando puntos extra, los cuales se determinan con un proceso sencillo usando operaciones de bits: dados dos números  $x$  e  $y$ , la expresión  $x \& y == y$  es verdadera si y sólo si los bits con valores 1 de la representación de  $y$  aparecen en la representación de  $x$ . Basta entonces usar como valor de  $x$  el identificador de la configuración y como  $y$  un valor que represente a los puntos en las aristas de cada cara, el cual corresponde a la suma de las potencias de 2 de los identificadores de sus puntos. A modo de ejemplo, en la configuración  $C_{1849}$  mostrada en la Figura 19 se aprecia un punto extra en la cara que contiene a los puntos de Steiner  $P_3, P_5, P_9$  y  $P_{10}$ , usando  $x = 1849$  e  $y = 2^3 + 2^5 + 2^9 + 2^{10} = 1576$  se tiene que, efectivamente, la expresión  $1849 \& 1576 == 1576$  es verdadera, por lo que corresponde agregar como punto extra el punto correspondiente a esa cara. Los valores de  $y$  contra los cuales se compara el identificador de cada configuración se calculan con anterioridad, siendo estos 27, 165, 326, 1576, 2640 y 3456, los cuales corresponden a los puntos extra  $P_{12}, P_{13}, P_{14}, P_{15}, P_{16}$  y  $P_{17}$ , respectivamente. El punto extra  $P_{18}$  sólo se incluye si la configuración tiene todos los puntos de Steiner posibles, es decir, la configuración  $C_{4095}$ .

#### 4.2.2. Algoritmo para encontrar la teselación de una configuración

Con el conjunto de puntos ya definido se procede a aplicar el algoritmo que dado estos datos como entrada, retorna el conjunto de elementos que componen la teselación. Este algoritmo se en-

---

<sup>6</sup>Este formato se encuentra descrito en la sección 3.1.1

cuenta (en pseudocódigo) en Algoritmo 1 y es implementado como una función que recibe como entrada el string construido con la función descrita en la sección 4.2.1, y un objeto Vector (vacío), cuyo contenido serán objetos de tipo Element que representarán a los elementos finales de la teselación.

En detalle, el algoritmo procede como sigue:

- Se crea un objeto  $E = \text{map}\langle \text{int}, \text{set}\langle \text{int}\rangle\rangle$ , para almacenar los índices de los vértices de cada elemento. La estructura map funciona con pares (llave, valor), que permite acceso directo a los valores almacenados por medio de su llave de la forma  $\text{map}[\text{key}]$ . En este caso las llaves son números enteros y los valores almacenados son objetos set (estructuras que aseguran que todos sus elementos son distintos) cuyos contenidos son números enteros.
- Utilizando PStreams, se crea un nuevo proceso que ejecuta Qhull con las opciones  $d$ ,  $Qz$  e  $i^7$ , el cual recibe como entrada el string con la descripción de la configuración.
- A continuación se procede a leer la salida del comando creado en el paso anterior. Dado que la opción  $d$  construye una teselación de Delaunay, el resultado obtenido es un conjunto de tetraedros, descrito cada uno por cuatro números que representan sus vértices. Estos números (índices) están asociados al orden en que los vértices se definieron en la entrada, es decir, un tetraedro descrito por la secuencia 0 1 5 6 está formado por el primer, segundo, sexto y séptimo vértice de la entrada.
- El objetivo ahora consiste en determinar cuáles tetraedros definen a cada elemento de la teselación, lo cual se consigue uniendo tetraedros cuyos vértices sean coesféricos, dado que esta es la característica principal que define a los elementos finales. Esto se consigue como se detalla a continuación:
  - Dado que la opción  $Qz$  agrega puntos extra al conjunto de entrada para lidiar con el caso de más de 4 vértices coesféricos, es directo deducir que todos los tetraedros que es posible unir presentarán como vértice un punto que no pertenece al conjunto original, ya que los vértices que si pertenecen al conjunto de entrada son coesféricos (de lo contrario, no habría un punto adicional).
  - Si un tetraedro contiene un vértice agregado, Qhull garantiza que éste siempre es el primero de los cuatro números que lo representa. De esta forma es posible identificar a cada tetraedro por su primer número.
  - Si el identificador del tetraedro es mayor que el número de puntos de entrada, el resto de sus puntos se almacena en la estructura  $E$ , dentro del set cuya llave es precisamente este identificador (así, todos los puntos coesféricos se almacenarán juntos). En caso contrario, el tetraedro representado no se une con ningun otro formando parte de los elementos

---

<sup>7</sup>El significado de estas opciones se encuentra en la sección 3.1.1

finales, por lo que sus cuatro valores se almacenan directamente en la estructura  $E$  indexados por una llave de valor negativo y decreciente, para asegurar unicidad.

- Una vez examinados todos los tetraedros se termina el proceso creado con PStreams. Luego de este paso, la estructura  $E$  contiene una descripción parcial de los elementos finales (se sabe cuáles son sus vértices, pero se desconoce la información sobre sus caras y aristas).
- Una forma sencilla de obtener la información restante se deduce a partir del siguiente hecho: dado que los vértices de los elementos finales son coesféricos, necesariamente estos elementos deben ser convexos. Luego basta con encontrar la cerradura convexa (usando Qhull) de estos vértices para obtener las caras y aristas de cada elemento final. Para esto se procede como sigue:
  - El primer paso es construir la entrada para Qhull. Recordando que la salida contiene índices a los vértices de entrada, es deseable que estos índices también se correspondan con los vértices originales para obtener una identificación directa. Para esto se construyen los strings de entrada manteniendo el número de vértices originales y reemplazando los que no correspondan al elemento final por vértices `dummy`. Estos vértices `dummy` no deben pertenecer a la cerradura convexa, es decir, deben estar al interior del elemento. Dado que, como se dedujo anteriormente, los elementos finales son convexos, necesariamente su centroide estará en el interior, siendo este punto el candidato perfecto para representar a los vértices `dummy`.
  - Con el string ya construído, se abre un nuevo proceso con PStreams que ejecuta Qhull con la opción `i`, al cual se le pasa el string generado como entrada para calcular la cerradura convexa y obtener los datos necesarios que permiten construir a los objetos `Element` que se almacenan en el `Vector` dado como argumento.
- Finalmente, el `Vector` argumento contendrá a todos los elementos finales que forman parte de esta teselación.



---

**Algoritmo 1:** Algoritmo de generación de teselaciones

---

**Input:**  $S$  = Conjunto de puntos que define el cuboide 1-irregular

**Output:**  $P$  = Conjunto de elementos que componen la teselación

**begin**

$P \leftarrow \emptyset$ ;

$E \leftarrow \text{new map}\langle \text{int}, \text{set} \langle \text{int} \rangle \rangle$  ;

$P_T \leftarrow \text{qhull d Qz } i < S$ ;

$nid \leftarrow -1$ ;

**for each**  $T$  **in**  $P_T$  **do**

$id \leftarrow T[0]$ ;

$a \leftarrow T[1]$ ;

$b \leftarrow T[2]$ ;

$c \leftarrow T[3]$ ;

**if**  $E[id] \neq \emptyset$  **then**

$E[id] = E[id] \cup \{a, b, c\}$ ;

**else**

**if**  $id < |S|$  **then**

$E[nid] = \{id, a, b, c\}$ ;

$nid = nid - 1$ ;

**else**

$E[id] = \{a, b, c\}$ ;

**end**

**end**

**end**

**for each**  $SET$  **in**  $E$  **do**

$S_E \leftarrow \emptyset$ ;

**for each**  $i$  **in**  $SET$  **do**

$S_E = S_E \cup S[i]$ ;

**end**

$P_E \leftarrow \text{qhull } i < S_E$ ;

$P = P \cup P_E$ ;

**end**

**end**

---

Para determinar la complejidad de este algoritmo se examina la complejidad de sus partes. Si  $n$  corresponde a la cantidad de puntos de entrada, se tiene que:

- El cálculo de una teselación de Delaunay tiene complejidad  $O(n \log n)$ .
- Procesar el output de la teselación de Delaunay toma una complejidad lineal con respecto al número de tetraedros obtenidos.
- El centroide de un elemento de  $k$  vértices se determina con una complejidad de  $O(k)$ .
- La implementación interna de Qhull para una cerradura convexa de un elemento de  $k$  vértices tiene complejidad  $O(n \log k)$ .

Es decir, una configuración cuya teselación de Delaunay tiene  $t$  tetraedros y produce  $t_f$  elementos finales de  $k_i$  vértices ( $i = 1, 2, \dots, t_f$ ) tiene una complejidad dada por la expresión:

$$O(n \log n) + O(t) + \sum_{i=1}^{t_f} (O(k_i) + O(n \log k_i))$$

### 4.2.3. Identificando a los elementos de la partición

El algoritmo presentado en la sección 4.2.2 obtiene un conjunto de elementos finales, sin embargo estos sólo están descritos por algunas de sus propiedades geométricas (vértices, aristas, caras, centroide y circuncentro) y en este paso aún se desconoce el poliedro que cada uno representa.

Dentro de la implementación del generador de datos, existe un método que es capaz de clasificar un elemento encontrado identificándolo con cualquiera de los elementos finales conocidos. Esto se hace reconociendo el número de vértices, número de caras, número de aristas y la identificación de la forma de sus caras<sup>8</sup>. En caso de no poder realizar una identificación exitosa, el clasificador retorna una descripción del elemento desconocido con información útil para que éste sea incluido en el conjunto de elementos finales. Esta descripción contiene el número de vértices, número de caras, número de aristas, la forma de sus caras y un ejemplo de una configuración donde este elemento aparece.

El conjunto de elementos finales base descrito en la sección 2.2.1 es extendido a medida que al analizar configuraciones aparecen elementos que el clasificador no es capaz de reconocer, y se mejora

---

<sup>8</sup>Por ejemplo, un Complemento de Tetraedro está representado como un elemento de 7 vértices, 7 caras, 12 aristas, 4 triángulos y 3 rectángulos.

el mismo para que tenga la capacidad de hacerlo. Para encontrar la totalidad de los elementos que pueden aparecer en todas las configuraciones se utiliza un cuboide de lados iguales (cubo) como estructura inicial, ya que estas proporciones generan la mayor cantidad de elementos coesféricos distintos, pues presenta la mayor cantidad de subconjuntos de vértices coesféricos, dando como resultado la identificación de 17 elementos nuevos, cuya descripción detallada se encuentra en la sección 6.1.

### 4.3. Generación de Datos: Un caso de ejemplo

Para ejemplificar el proceso de generación de datos mostrado en la sección 4.2, se presentan los pasos de éste aplicados a la configuración  $C_{2780}$  sobre un cuboide de aristas iguales de tamaño 2.

#### 4.3.1. Identificación de puntos de Steiner

El primer paso del proceso consiste en determinar cuáles son los puntos de Steiner que aparecen en la configuración. Como se estableció en la sección 4.1.2, se debe analizar la representación del identificador de la configuración. Se tienen entonces las siguientes igualdades:

$$2780 = 101011011100_2 = 2^2 + 2^3 + 2^4 + 2^6 + 2^7 + 2^9 + 2^{11}$$

Con esto, los puntos de Steiner de la configuración son  $P_2, P_3, P_4, P_6, P_7, P_9$  y  $P_{11}$ . Se hace la pregunta, ¿es necesario agregar puntos extra?. De acuerdo a lo mostrado en la sección 4.2.1, basta con hacer las operaciones de bits respectivas para obtener la respuesta. Se tiene que sólo la expresión  $2780 \& 2640 == 2640$  es verdadera, por lo que se agrega el punto  $P_{16}$ .

Con estos datos se construye el input para ser utilizado por el Algoritmo de la sección 4.2.2. Debido a que el cuboide base utilizado se extiende por las coordenadas  $\{-1, 1\}$  la descripción de la configuración queda como se aprecia en la Figura 20.

#### 4.3.2. Aplicación del Algoritmo para encontrar la teselación

Con los datos encontrados en la sección anterior se aplica el Algoritmo de generación de teselaciones. En esta sección se abordaran los puntos más relevantes.

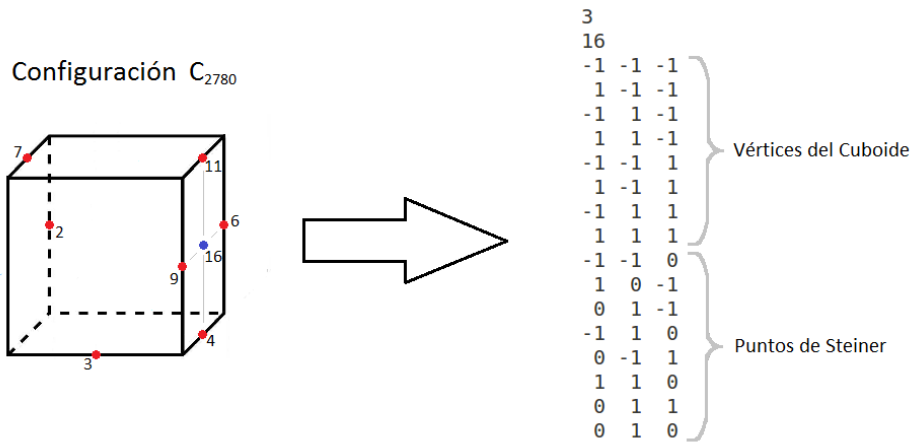


Figura 20: La configuración  $C_{2780}$  y su representación como input para el algoritmo.

Ejecutar el comando `qhull d Qz i`, entregando el input correspondiente (Figura 20), produce el siguiente output de 48 tetraedros<sup>9</sup>:

```

48
 8 9 1 0    18 10 9 0    18 15 10 0    18 8 9 15    18 9 10 15
18 8 15 0    18 9 8 0    19 11 15 2    19 11 2 0    19 15 10 2
19 2 10 0    19 10 15 0    19 15 8 0    19 8 11 0    19 11 8 15
20 10 9 15   20 13 10 15   20 13 3 10   20 9 13 15   20 3 13 9
20 10 3 9    8 12 9 15    5 12 9 1    12 8 9 1    21 8 11 15
21 11 8 4    21 11 14 15   21 6 14 11   21 6 11 4    21 14 6 4
21 12 14 4   21 14 12 15   21 8 12 4    21 12 8 15   22 14 7 15
22 14 5 7    22 5 13 7    22 13 5 15   22 7 13 15   22 12 14 15
22 5 12 15   22 12 5 14    23 13 5 9    23 5 13 15   23 13 9 15
23 12 5 15   23 9 12 15    23 5 12 9

```

Luego de examinar este output, la estructura mapa que contiene los índices de los vértices de cada elemento final se presenta a continuación:

<sup>9</sup>Se omiten saltos de línea por motivos de espacio, cada una de las 48 4-tuplas representa una línea de output.

ID	Conjunto de Índices
-4	1 8 9 12
-3	1 5 9 12
-2	8 9 12 15
-1	0 1 8 9
18	0 8 9 10 15
19	0 2 8 10 11 15
20	3 9 10 13 15
21	4 6 8 11 12 14 15
22	5 7 12 13 14 15
23	5 9 12 13 15

Cada uno de los 10 conjuntos encontrados conforman un elemento coesférico final. Aplicando a cada uno de ellos el comando `qhull` se obtiene la definición de sus caras (y por extensión, la de sus aristas) que permite construir objetos genéricos de tipo `Element` que los representan<sup>10</sup>. En esta fase es directo notar la existencia de 4 tetraedros (conjuntos indexados por un valor negativo, compuestos por 4 vértices cada uno), pero resta reconocer a los otros 6 elementos. Se obtienen 10 elementos genéricos con las siguientes características:

ID	N° Vértices	N° Caras	N° Aristas	Detalle de sus caras
-4	4	4	6	4 Triángulos
-3	4	4	6	4 Triángulos
-2	4	4	6	4 Triángulos
-1	4	4	6	4 Triángulos
18	5	5	8	4 Triángulos, 1 Cuadrilátero
19	6	5	9	2 Triángulos, 3 Cuadriláteros
20	5	5	8	4 Triángulos, 1 Cuadrilátero
21	7	7	12	4 Triángulos, 3 Cuadriláteros
22	6	5	9	2 Triángulos, 3 Cuadriláteros
23	5	5	8	4 Triángulos, 1 Cuadrilátero

### 4.3.3. Identificación de elementos finales

Ya se ha determinado la existencia de 10 elementos coesféricos finales en la configuración  $C_{2780}$  dentro de un cuboide de lados iguales. Falta entonces realizar su identificación: el clasificador toma cada elemento genérico `Element` y retorna cuál es el elemento final respectivo. Como ya se han

<sup>10</sup>Este paso se omite pues no aporta mucha información sobre el proceso.

identificado todos los elementos finales en configuraciones 1-irregulares generadas por bisección, no apareceran elementos nuevos. El conjunto final de elementos reconocidos por el clasificador resulta ser:

ID	Elemento coesférico final
-4	Tetraedro
-3	Tetraedro
-2	Tetraedro
-1	Tetraedro
18	Pirámide de base cuadrilátera
19	Prisma triangular
20	Pirámide de base cuadrilátera
21	Complemento de tetraedro
22	Prisma triangular
23	Pirámide de base cuadrilátera

#### 4.4. Visualizador

Para apoyar y facilitar el análisis de la teselaciones se construye una herramienta que permite visualizar interactivamente los resultados de la teselación de todas las configuraciones bajo distintos tamaños del cuboide usado como estructura inicial, indicando ademas cuáles son los poliedros que la conforman. Para desarrollar esta aplicación gráfica se utilizan las herramientas que facilita el framework Qt y la librería QGLViewer.

En particular, esta herramienta consta de las siguientes funcionalidades:

- Visualización interactiva de la teselación de una configuración especificada, la cual permite examinar tanto en conjunto como separadamente los elementos que la componen.
- Visualización de los centroides de cada elemento coesférico final.
- Visualización de los puntos de Voronoi (circuncentros) de cada elemento coesférico final.
- Controles para modificar el tamaño del cuboide base.
- Identificación del poliedro que representa a cada elemento coesférico final.

#### 4.4.1. Implementación

La implementación se realiza utilizando QtCreator, para definir clases y métodos, y QtDesigner, para crear la estructura general de la aplicación. En su mayoría, los componentes de la aplicación corresponden a elementos estándar como checkboxes, o sliders. Sin embargo, un componente que requiere especial mención es el que representa a un visor<sup>11</sup>, para el cual se utiliza un objeto AppViewer (subclase del objeto QGLViewer), con el fin de personalizar su comportamiento para adecuarlo a los requerimientos de la aplicación. En particular, se sobrescribe el método `init()` para especificar valores relevantes a la aplicación (como el estado del `mouseTracking`, que activo permite hacer un seguimiento interno de la posición del cursor dentro del visor) y asignar parámetros especiales de OpenGL (valores de iluminación y tamaños de puntos y líneas), y el método `draw()`, donde se personalizan los parámetros de dibujo de acuerdo a las opciones incluídas en el visualizador (como por ejemplo, seleccionar qué dibujar). Además, en este objeto se almacena el conjunto de colores utilizado para dibujar, junto con nombres que los describen, con el fin de identificar a cada elemento con su color.

#### 4.4.2. Componentes

Es posible dividir la estructura del visualizador en distintas componentes de acuerdo a su funcionalidad. La Figura 21 muestra una visión general de la aplicación con sus componentes numeradas, las cuales se describen a continuación.

### Componente 1. Opciones generales

En esta componente se puede seleccionar entre un conjunto de opciones aplicables a la configuración a dibujar. Se encuentra dividida en dos grupos de opciones separadas por pestañas, las cuales se pueden ver en la Figura 22.

#### Componente 1.1. Opciones de visualización

Permite especificar opciones generales de visualización:

- Activar/desactivar modo Wireframe. Estando desactivado (valor por defecto), los elementos dibujados son sólidos y la iluminación está activa. En caso contrario, las caras de cada elemento son transparentes y la iluminación se desactiva. Activar el modo Wireframe permite

---

<sup>11</sup>Se usa el termino visor para distinguirlo de “visualizador”, el cual en el contexto de este trabajo se refiere a la aplicación en general.

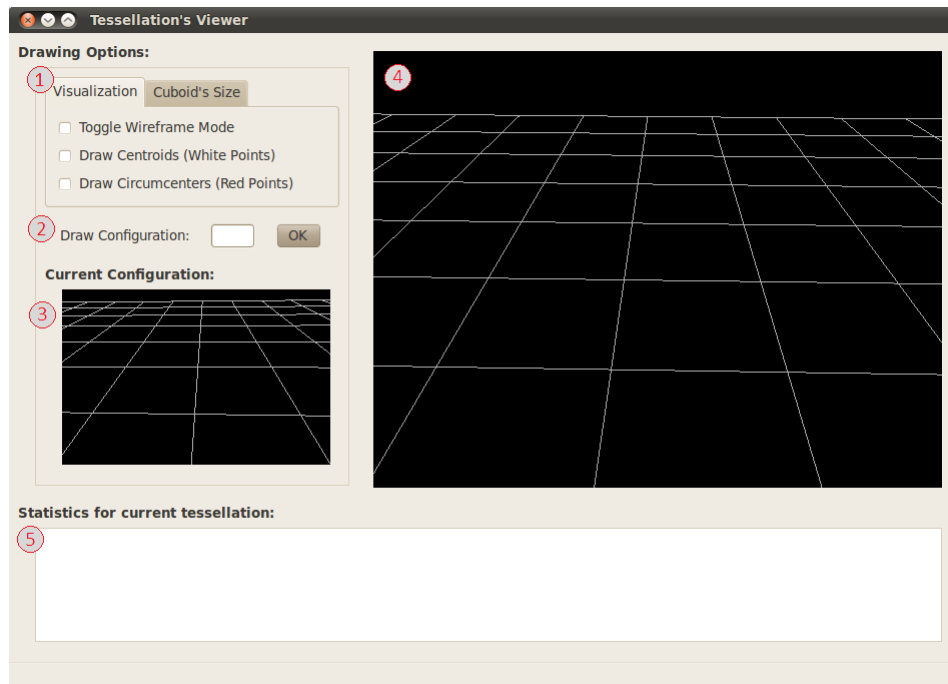


Figura 21: Visualizador de teselaciones y sus componentes.



Figura 22: Opciones generales.

identificar la ubicación de los centroides y circuncentros (asumiendo que dichas opciones se encuentran activas).

- Dibujar/ocultar los centroides de cada elemento, los cuales aparecen como puntos blancos. Activar esta opción resulta útil para identificar cuál es el área de acción que permite seleccionar un elemento en particular, pues esta área es dependiente de la posición del centroide. Si el modo Wireframe está desactivado, activar esta opción no tiene efecto, pues todos los centroides están en el interior de los elementos, siendo estos invisibles si los elementos son sólidos.

- Dibujar/ocultar los circuncentros de cada elemento, los cuales aparecen como puntos rojos. Activar esta opción muestra la posición del circuncentro de cada el elemento, siendo esto útil para examinar si alguno de estos puntos queda en el exterior de la teselación.

## Componente 1.2. Selector de tamaño del cuboide base



Permite especificar el tamaño del cuboide base mediante tres controladores (sliders) que varían el ancho, la profundidad y el alto entre un rango de 1.00 y 5.00, siendo 1.00 el valor por defecto en cada controlador. El visualizador se actualiza automáticamente con los cambios en esta sección. Modificar estos valores resulta de utilidad para identificar, dada una configuración determinada, los valores límite para los cuales los circuncentros se mantienen dentro de la teselación, o para encontrar los valores límite para los cuales el conjunto de elementos de la teselación varía.

## **Componente 2. Selector de configuración**

Esta componente contiene un campo de texto el cual permite ingresar el identificador de la configuración a visualizar. Ignora cualquier valor que sea distinto de un número entero entre 0 y 4095. El visualizador sólo se actualiza luego de presionar el botón OK, modificando la configuración desplegada, pero manteniendo el estado de las opciones.

## **Componente 3. Visor de configuración actual**

Este pequeño visor contiene una descripción interactiva de la configuración activa, mostrando el cuboide base (representado por sus aristas) y los vértices usados en la teselación. Permite un control de cámara mediante comandos del mouse de tipo “click and drag”: el botón izquierdo permite una rotación con respecto al origen de la vista, el botón derecho permite trasladar la posición de la cámara y el botón central (rueda) permite modificar el nivel de zoom. Este visor no cuenta con la opción de mouseTracking activa y no es afectado por los cambios en las opciones del visualizador, a excepción del cambio de dimensiones, que lo actualiza acorde a los valores especificados.

## **Componente 4. Visor de teselación actual**

La componente principal del visualizador. Este visor despliega a los elementos que componen la teselación correspondiente de la configuración seleccionada. Posee comandos de control de cámara idénticos a los del visor de configuraciones. Además, este visor tiene la opción de mouseTracking activa, lo que proporciona comandos adicionales. En particular, esto permite que en conjunto con el objeto CustomManipulatedFrame perteneciente a cada objeto Element, se pueda utilizar la funcionalidad del método `checkIfGrabsMouse` que otorga la posibilidad de seleccionar un elemento dibujado (el cual cambia de color para indicar que está siendo seleccionado) con sólo posar el puntero del mouse sobre el área de efecto (región en torno al centroide). Un elemento seleccionado activa

comandos especiales del mouse de tipo “click and drag”<sup>12</sup>: el botón izquierdo permite una rotación del elemento en torno a su “origen” (asignado como su centroide), mientras que el botón derecho permite trasladar al elemento por la escena (esto permite aislarlos y examinarlos por separado), por último, el botón central (rueda) permite desplazar al elemento a través de un eje específico.

## Componente 5. Descripción de configuración actual

Esta componente contiene un cuadro de texto no editable en el que se despliega información relacionada con la configuración representada, como el número de elementos que la componen y la descripción de éstos. Cada elemento es asociado al color con el que es dibujado (gracias a que la clase que representa al visor tiene la definición de los colores usados), con el objetivo de poder identificarlo en el visor de teselaciones.

La Figura 23 muestra un ejemplo del visualizador en funcionamiento. En el Anexo C se encuentran imágenes adicionales, con el visualizador bajo distintas configuraciones y opciones.

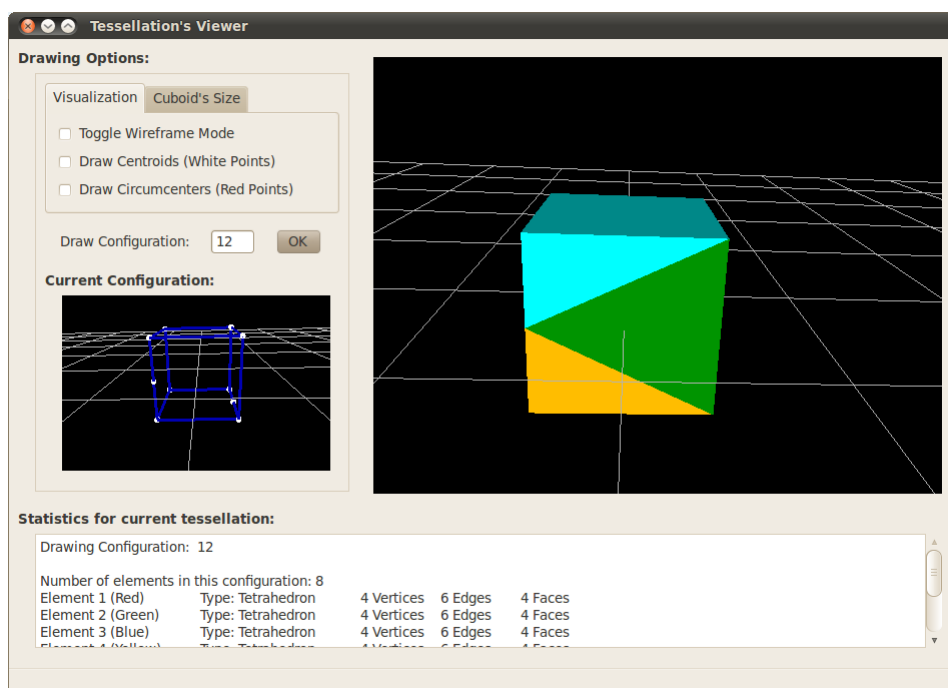


Figura 23: Visualizador en funcionamiento.

<sup>12</sup>Esto se traduce en que, mientras un elemento esté seleccionado, los controles de cámara con el mouse no estarán disponibles.

## 4.5. Validación

### 4.5.1. Determinación de Clases de Equivalencia

En la sección 2.1.1 se menciona que uno de los resultados recopilados como antecedente es la determinación de la existencia de 144 clases de equivalencia distintas. En [3], se presentan los conceptos de clases de equivalencia mediante la siguiente definición: Dadas dos configuraciones 1-irregulares  $c_1$  y  $c_2$ , se dice que pertenecen a la misma clase de equivalencia si  $c_1$  puede ser transformado en  $c_2$  mediante rotaciones o reflexiones. En la misma publicación, y mediante una examinación exhaustiva (fuerza bruta) se identificaron 144 clases de equivalencia, pero no hay un registro de cómo están conformadas. Por esto, se implementa una solución que permita validar este resultado y encontrar el conjunto de configuraciones que compone cada clase de equivalencia.

Se extiende el generador de datos con funciones que permiten rotar una configuración y reflejarla en torno a algún eje. Esto permite que para cualquier configuración sea posible generar todas sus equivalencias. Finalmente, se procede mediante un enfoque exhaustivo descrito a continuación:

- Se crea un arreglo de valores booleanos para marcar cuales configuraciones ya fueron procesadas.
- Se revisa cada configuración secuencialmente: si la configuración ya fue marcada como procesada, se continúa con la siguiente. En caso contrario, se generan todas sus configuraciones equivalentes, se almacenan y se marcan como procesadas.

Al terminar este proceso se obtienen 144 conjuntos de configuraciones correspondientes a las clases de equivalencia. El detalle completo de estos conjuntos se encuentra en el Anexo B.

Considerando el hecho de que luego de este proceso se han determinado completamente a las 144 clases de equivalencia surge naturalmente la pregunta de por qué no se utilizaron estos resultados en la implementación de este trabajo. La respuesta es que tanto el algoritmo para encontrar teselaciones como el visualizador fueron desarrollados considerando el enfoque más general posible, pensando en un eventual trabajo futuro. Sin ir más lejos, como se verá en la sección 5 de este trabajo el algoritmo implementado es perfectamente compatible con configuraciones 1-irregulares generadas por intersección.

### 4.5.2. Otras pruebas y desempeño

Adicionalmente, una forma de validar que las teselaciones generadas esten correctas es comparándolas con las teselaciones pre-calculadas usadas por el Generador  $\Omega$ [2], para esto, se examina

la descripción de las configuraciones almacenadas y sus particiones respectivas en el código fuente de la aplicación verificando que éstas sean exactamente iguales a las teselaciones obtenidas por el algoritmo.

Con respecto al tiempo computacional de procesamiento, generar las 4096 configuraciones distintas en una máquina virtual con un procesador de 2.4GHz y 1MB de RAM, toma cerca de 13 segundos.

## 5. Configuraciones 1-irregulares generadas por Intersección

En esta sección se analizan los aspectos relacionados a las configuraciones 1-irregulares generadas por intersección, explicando algunas consideraciones adicionales a tener en cuenta, mostrando la revisión de uno de los resultados presentados como antecedente y definiendo el proceso de generación de datos.

### 5.1. Consideraciones adicionales

#### 5.1.1. Puntos extra en las configuraciones

Análogamente al caso de bisección (sección 4.1.1), también se tienen criterios para agregar puntos adicionales a las configuraciones generadas por intersección. Estos criterios son muy similares a los ya presentados existiendo sólo pequeñas diferencias, como se verá a continuación:

- Si las 4 aristas que conforman una de las caras del cuboide base presentan puntos de Steiner, se agrega un punto adicional solamente si el segmento que une a cada par de puntos en aristas opuestas es perpendicular a dichas aristas. De agregarse, el punto extra se ubica en la intersección de los segmentos que unen a cada par de puntos.
- Si se ha agregado un punto extra en todas las caras, se agrega un punto adicional en la intersección de los segmentos que unen a cada par de puntos en caras opuestas.

En la Figura 24 se encuentran ejemplos correspondientes a los dos criterios recientemente descritos.

Al igual que en las configuraciones por bisección, la inclusión de estos puntos adicionales no afecta a la cantidad total de configuraciones posibles.

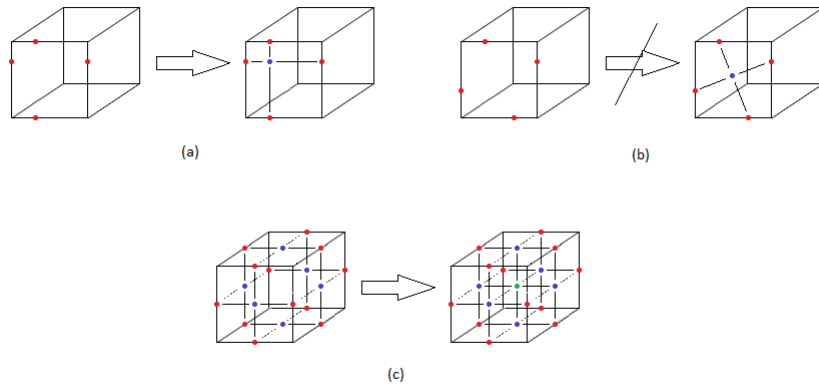


Figura 24: Criterios para agregar puntos extra a una configuración: (a) Punto extra en una cara. (b) A pesar de que las 4 aristas que conforman una cara tienen un punto de Steiner, el punto extra no se agrega pues no se cumple el criterio asociado. (c) Punto extra al interior del cuboide.

### 5.1.2. Igualdad entre configuraciones

Dado que en este tipo de configuraciones los puntos de Steiner pueden aparecer en cualquier parte de una arista, la cantidad de posiciones disponibles para estos puntos es infinita. Para acotar el número de configuraciones a estudiar, se hace uso de la noción de igualdad entre configuraciones definida en [3] y mencionada en la sección 2.1.2. Esto permite enfocarse solamente en las posiciones relativas de los puntos de Steiner con respecto a sus aristas paralelas. En la Figura 25 se muestran cuboides 1-irregulares que ejemplifican esta definición.

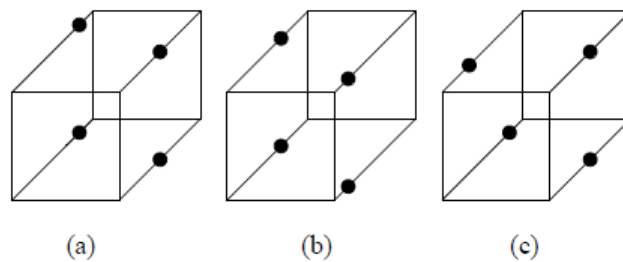


Figura 25: Igualdad entre configuraciones: Los cuboides 1-irregulares (a) y (b) se consideran iguales, ya que las posiciones relativas de sus puntos de Steiner coinciden. El cuboide (c) se considera distinto, pues la posición relativa del punto en su arista superior izquierda difiere con respecto a la de los cuboides anteriores.

## 5.2. Revisión de la cantidad de configuraciones distintas

Como se mencionó en la sección 2.1.2, en [3] se presenta una forma de obtener el total de configuraciones 1-irregulares distintas, calculando la cantidad de formas posibles de ubicar puntos en un conjunto de ejes paralelos y posteriormente elevando el valor obtenido a la cantidad de conjuntos de ejes paralelos, correspondiente al número de dimensiones del espacio estudiado, en este caso 3. Sin embargo, esta forma de contar es demasiado general, pues considera el hecho de que dados  $n$  ejes paralelos y  $k$  puntos ya fijados ( $n < k$ ), se tienen  $2k + 1$  posiciones relativas para fijar a un  $(k + 1)$ -ésimo punto: 1 posición a la izquierda de los  $k$  puntos, 1 posición a la derecha de los  $k$  puntos,  $k - 1$  posiciones intermedias y  $k$  posiciones sobre los  $k$  puntos. Este resultado no es siempre cierto, como se ve en la Figura 26, con  $k = 0$  se tiene sólo una posición relativa ( $2 \times 0 + 1$ ). Asignada esta posición ( $k = 1$ ), se tienen 3 ( $2 \times 1 + 1$ ) posibles posiciones. Posteriormente, con cada una de estas posibles asignaciones de dos puntos ( $k = 2$ ) se esperarían 5 ( $2 \times 2 + 1$ ) posiciones relativas por cada caso, lo cual no se cumple: si los dos primeros puntos se ubican en la misma posición relativa, para un tercero sólo habrán 3 posibilidades. Las 5 formas esperadas serían: una a la derecha, una a la izquierda, una entre ambos puntos, otra en la misma posición del primer punto y la última en la misma posición del segundo punto. Resulta evidente que en este caso las 3 últimas son exactamente iguales. Esto permite concluir que la cantidad efectiva de configuraciones distintas es menor al valor presentado.

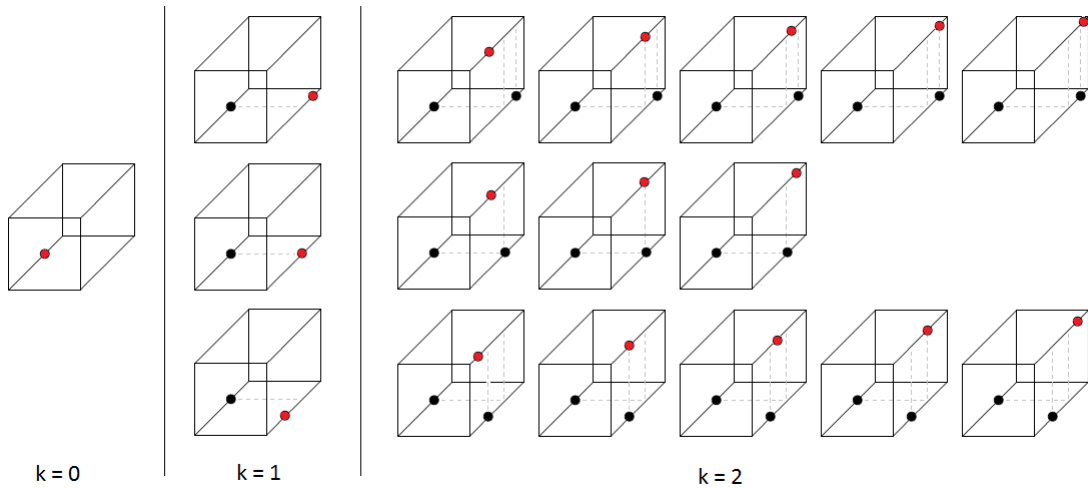


Figura 26: Distintas formas de ubicar puntos de Steiner. El valor de  $k$  indica la cantidad de puntos que ya se han agregado y los puntos rojos indican posibles posiciones en cada caso.

Para encontrar la cantidad efectiva de configuraciones se generan las 188 posibles formas de posicionar puntos en aristas paralelas, siguiendo la idea presentada en [3]. Naturalmente esto genera repeticiones, las cuales se eliminan para conseguir finalmente un total de 137 formas, dando como

resultado la existencia de  $137^3 = 2.571.353$  configuraciones 1-irregulares distintas.

### 5.3. Generación de datos

El proceso de obtención de datos es bastante similar al proceso utilizado para las configuraciones generadas por bisección (sección 4.2), la principal diferencia es que en este caso el tamaño del cuboide base está fijado con anterioridad como un cuboide de lados iguales. Este proceso se describe en las secciones siguientes.

#### 5.3.1. Construyendo una configuración

Para este tipo de configuraciones no se cuenta con una forma directa de identificar a una en específico. Como fue mencionado en la sección 5.2, se sabe cuáles son las 137 formas de ubicar un conjunto de puntos en aristas paralelas, por lo que para determinar una configuración basta con escoger una de estas formas para cada conjunto de aristas paralelas. Sin embargo como ésto sólo indica una posición relativa, se debe definir las posiciones reales de cada uno de los puntos. La convención a usar es la siguiente:

- El primer punto siempre estará ubicado en el punto medio de una arista.
- Si el  $k$ -ésimo punto se ubica a la izquierda de los  $k - 1$  anteriores, éste quedará en el punto medio entre el segmento que une al vértice izquierdo de la arista con el punto ya asignado que esté más a la izquierda. Análogamente, la posición a la derecha de los  $k - 1$  puntos anteriores queda determinada por el punto medio del segmento que une el punto ya asignado más a la derecha con el vértice derecho de la arista.
- Si la posición relativa del  $k$ -ésimo punto está entre dos puntos ya asignados, su posición real queda determinada por el punto medio del segmento que une a los dos puntos.

Con las posiciones reales de los puntos ya definidas, y con tres formas distintas de escoger las posiciones relativas (una para cada conjunto de aristas paralelas) se puede construir la definición de la configuración para ser procesada por el algoritmo que permite encontrar su teselación.

#### 5.3.2. Algoritmo para encontrar la teselación de una configuración

Observando el algoritmo definido en 4.2.2, se puede notar que en ningún momento hace uso de alguna propiedad exclusiva presente en sólo configuraciones generadas por bisección, por lo que es

perfectamente aplicable en el contexto de configuraciones generadas por intersección sin necesidad de hacer modificación alguna. Recordar que este algoritmo sólo retorna un conjunto de elementos aún no identificados, siendo necesario un proceso posterior de clasificación.

### **5.3.3. Identificando a los elementos de la partición**

El clasificador opera de la misma forma a como lo hace en el caso de configuraciones generadas por bisección (sección 4.2.3). Además de los 24 elementos finales ya identificados (7 elementos base y 17 adicionales encontrados en el contexto de este trabajo), aparecen 14 elementos adicionales exclusivos a configuraciones generadas por intersección, cuya descripción se encuentra en la sección 6.4.

## **5.4. Desempeño**

La generación y procesamiento de las  $137^3$  configuraciones de este tipo toma cerca de 81700 segundos en la misma máquina descrita en 4.5.2, como consecuencia de esto, una implementación en paralelo resulta no ser imprescindible.

## **6. Resultados**

En este capítulo se presentan los nuevos elementos co-esféricos encontrados luego de realizar la experimentación sobre ambos tipos de configuración 1-irregular, además de un análisis sobre los resultados obtenidos variando el tamaño de los lados del cuboide para el caso de configuraciones generadas por bisección y un resumen sobre la frecuencia de los elementos encontrados en configuraciones generadas por intersección. Todas las figuras presentadas correspondientes a elementos coesféricos finales se obtuvieron desde el Visualizador, modificando la paleta de colores de la aplicación.

### **6.1. Elementos coesféricos adicionales identificados en configuraciones 1-irregulares generadas por bisección**

Como se mencionó en la sección 4.2.3, se han identificado 17 elementos adicionales a los 7 ya existentes para las teselaciones de un cubo 1-irregular. De ser posible, se les asigna un nombre idóneo, en caso contrario reciben un nombre genérico. A continuación se presenta una descripción de cada uno de ellos:



- **Pirámide Pentagonal:** Elemento de 6 vértices, 6 caras y 10 aristas. Se compone de 5 caras triangulares y 1 cara pentagonal. Un ejemplo de este elemento se puede apreciar en la Figura 27.

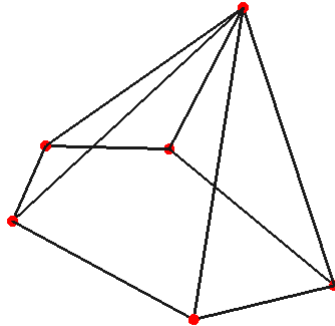


Figura 27: Pirámide Pentagonal.

- **Pirámide Hexagonal:** Elemento de 7 vértices, 7 caras y 12 aristas. Se compone de 6 caras triangulares y 1 cara hexagonal. Un ejemplo de este elemento se puede apreciar en la Figura 28.

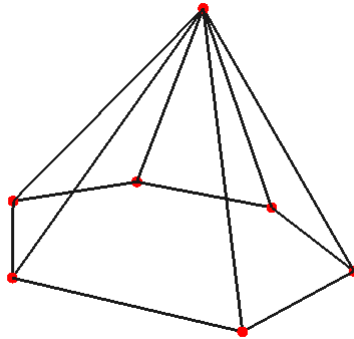


Figura 28: Pirámide Hexagonal.

- **Bipirámide Triangular:** Elemento de 5 vértices, 6 caras y 9 aristas. Se compone de 6 caras triangulares. Como su nombre sugiere, corresponde a la unión de dos pirámides triangulares (tetraedros). Un ejemplo de este elemento se puede apreciar en la Figura 29.

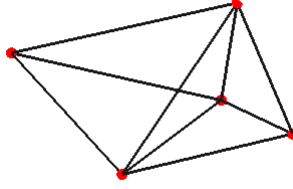


Figura 29: Bipirámide Triangular.

- **Bipirámide Cuadrilátera:** Elemento de 6 vértices, 8 caras y 12 aristas. Se compone de 8 caras triangulares. Como su nombre sugiere, corresponde a la unión de dos pirámides de base cuadrilátera. Un ejemplo de este elemento se puede apreciar en la Figura 30.

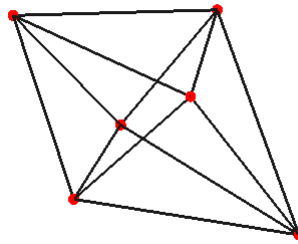


Figura 30: Bipirámide Cuadrilátera.

- **Bipirámide Pentagonal:** Elemento de 7 vértices, 10 caras y 15 aristas. Se compone de 10 caras triangulares. Como su nombre sugiere, corresponde a la unión de dos pirámides pentagonales. Un ejemplo de este elemento se puede apreciar en la Figura 31.

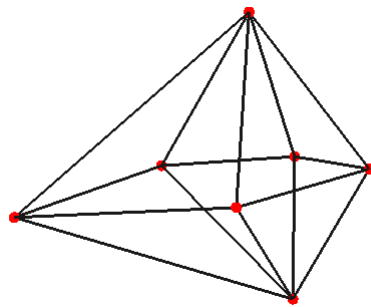


Figura 31: Bipirámide Pentagonal.

- **Bipirámide Hexagonal:** Elemento de 8 vértices, 12 caras y 18 aristas. Se compone de 12 caras triangulares. Como su nombre sugiere, corresponde a la unión de dos pirámides hexagonales. Un ejemplo de este elemento se puede apreciar en la Figura 32.

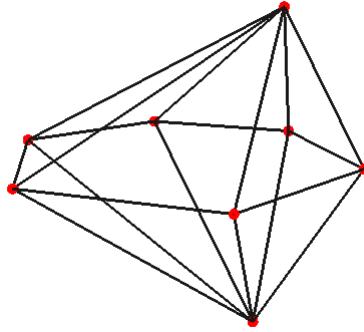


Figura 32: Bipirámide Hexagonal.

- **Biprisma:** Elemento de 8 vértices, 8 caras y 14 aristas. Se compone de 4 caras triangulares y 4 cuadriláteros. Se puede expresar como dos prismas triangulares unidos de forma cruzada. Un ejemplo de este elemento se puede apreciar en la Figura 33.

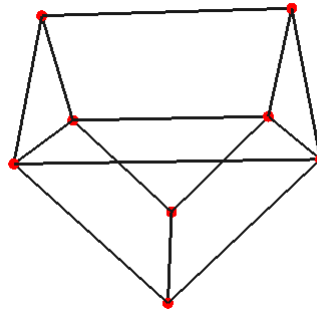


Figura 33: Biprisma Triangular.

- **Elemento genérico #1:** Elemento de 6 vértices, 5 caras y 9 aristas. Se compone de 2 caras triangulares y 3 cuadriláteros. Un ejemplo de este elemento se puede apreciar en la Figura 34.

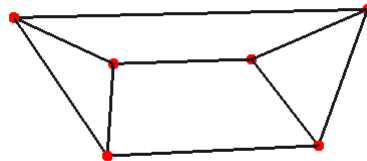


Figura 34: Elemento genérico #1.

- **Elemento genérico #2:** Elemento de 6 vértices, 6 caras y 10 aristas. Se compone de 4 caras triangulares y 2 cuadriláteros. Un ejemplo de este elemento se puede apreciar en la Figura 35.

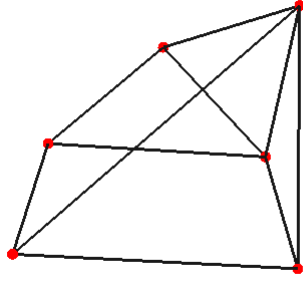


Figura 35: Elemento genérico #2.

- **Elemento genérico #3:** Elemento de 6 vértices, 7 caras y 11 aristas. Se compone de 6 caras triangulares y 1 cuadrilátero. Un ejemplo de este elemento se puede apreciar en la Figura 36.

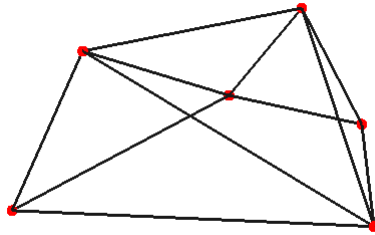


Figura 36: Elemento genérico #3.

- **Elemento genérico #4:** Elemento de 7 vértices, 7 caras y 12 aristas. Se compone de 5 caras triangulares, 1 cuadrilátero y 1 pentágono. Un ejemplo de este elemento se puede apreciar en la Figura 37.

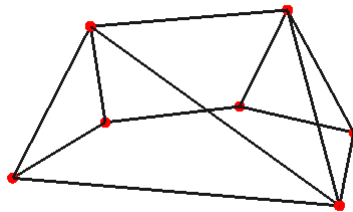


Figura 37: Elemento genérico #4.

- **Elemento genérico #5:** Elemento de 7 vértices, 8 caras y 13 aristas. Se compone de 6 caras triangulares y 2 cuadriláteros. Un ejemplo de este elemento se puede apreciar en la Figura 38.

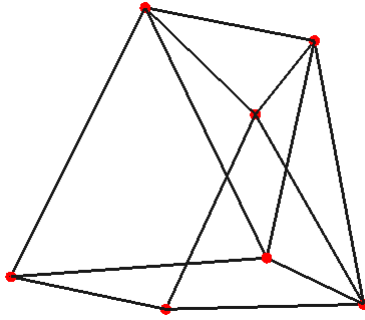


Figura 38: Elemento genérico #5.

- **Elemento genérico #6:** Elemento de 8 vértices, 9 caras y 15 aristas. Se compone de 6 caras triangulares y 3 cuadriláteros. Un ejemplo de este elemento se puede apreciar en la Figura 39.

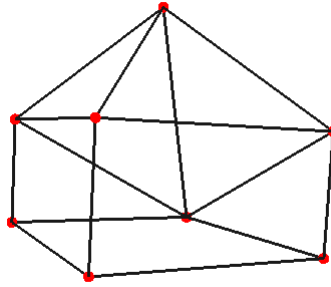


Figura 39: Elemento genérico #6.

- **Elemento genérico #7:** Elemento de 8 vértices, 10 caras y 16 aristas. Se compone de 8 caras triangulares y 2 cuadriláteros. Un ejemplo de este elemento se puede apreciar en la Figura 40.

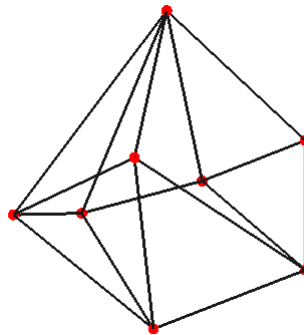


Figura 40: Elemento genérico #7.

- **Elemento genérico #8:** Elemento de 8 vértices, 11 caras y 17 aristas. Se compone de 10 caras triangulares y 1 cuadrilátero. Un ejemplo de este elemento se puede apreciar en la Figura 41.

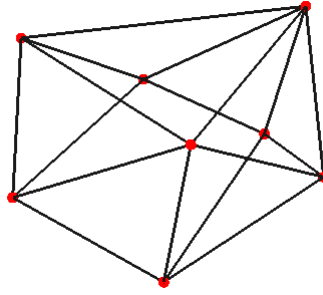


Figura 41: Elemento genérico #8.

- **Elemento genérico #9:** Elemento de 9 vértices, 9 caras y 16 aristas. Se compone de 4 caras triangulares y 5 cuadriláteros. Un ejemplo de este elemento se puede apreciar en la Figura 42.

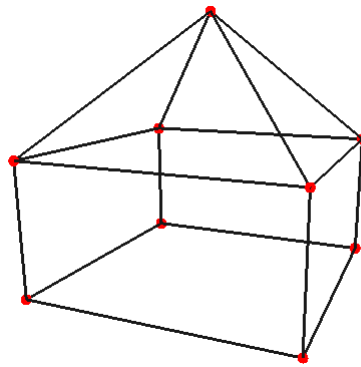


Figura 42: Elemento genérico #9.

- **Elemento genérico #10:** Elemento de 9 vértices, 11 caras y 8 aristas. Se compone de 8 caras triangulares y 3 cuadriláteros. Un ejemplo de este elemento se puede apreciar en la Figura 43.

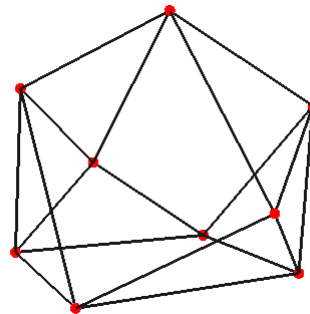


Figura 43: Elemento genérico #10.

## 6.2. Análisis de configuraciones 1-irregulares generadas por bisección

A continuación se presenta un resumen de los resultados obtenidos al generar las teselaciones de todas las configuraciones 1-irregulares y su análisis bajo distintos criterios. Los criterios a considerar

para construir las teselaciones son:

- **Encontrar las teselaciones óptimas:** Este es simplemente el proceso que se ha realizado hasta ahora. La teselación óptima encuentra la menor cantidad de elementos coesféricos para cada teselación, maximizando la cantidad de elementos distintos en total.
- **Minimizar la cantidad de elementos coesféricos distintos sin agregar aristas internas:** Bajo este criterio se busca reducir la cantidad de elementos finales distintos (pues puede resultar complicado para un generador de mallas trabajar con una cantidad extensa de elementos finales), con la condición de no agregar aristas adicionales a los elementos finales. Para esto se busca cuáles son los elementos finales que es posible separar directamente en dos o más. Examinando el conjunto de elementos finales (sección 6.1) se observa que las bipirámides y el biprisma son naturalmente divisibles en dos elementos, y además también lo son los elementos genéricos #5 (separable en un prisma y una pirámide de base cuadrilátera), #8 (separable en un prisma y dos pirámides de base cuadrilátera) y #9 (separable en un cuboide y una pirámide de base cuadrilátera). Un ejemplo de este tipo de separación se encuentra en la Figura 44.

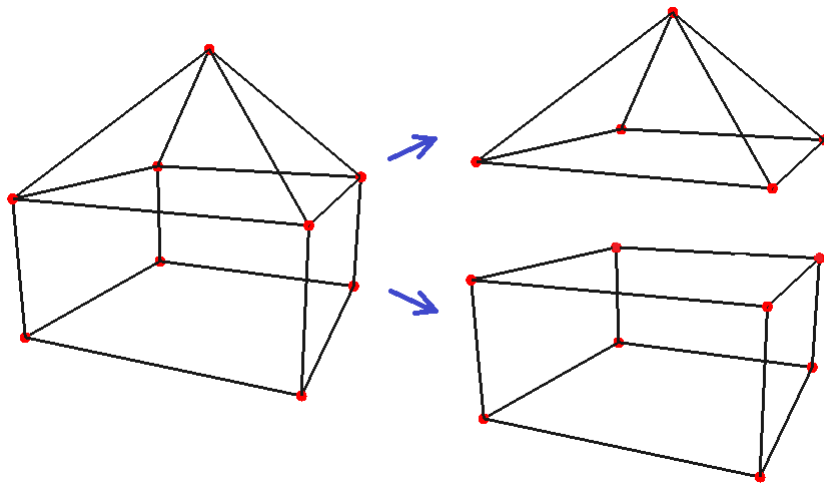


Figura 44: Separación del elemento genérico #9 en dos elementos.

- **Minimizar la cantidad de elementos coesféricos distintos agregando aristas internas:** Este criterio incluye las condiciones del criterio anterior y además agrega la condición de que es posible agregar aristas siempre y cuando estén en el interior de los elementos finales. La razón de sólo permitir aristas internas es que al agregar aristas externas se puede modificar la partición exterior de las caras del cuboide base, lo cual no es posible considerando que estos cuboides están inmersos en un contexto correspondiente a una malla geométrica más grande (modificar la partición de una de las caras del cuboide provocará una inconsistencia con el cuboide vecino, forzando a que éste deba ser modificado también). Bajo este criterio,

son separables los elementos genéricos #3 (una arista interna produce dos tetraedros y una pirámide de base cuadrilátera), #6 (una arista interna produce dos tetraedros y un complemento de tetraedro) y #7 (dos aristas internas producen dos tetraedros, una pirámide de base cuadrilátera y un prisma deformado). Un ejemplo de este tipo de separación se encuentra en la Figura 45.

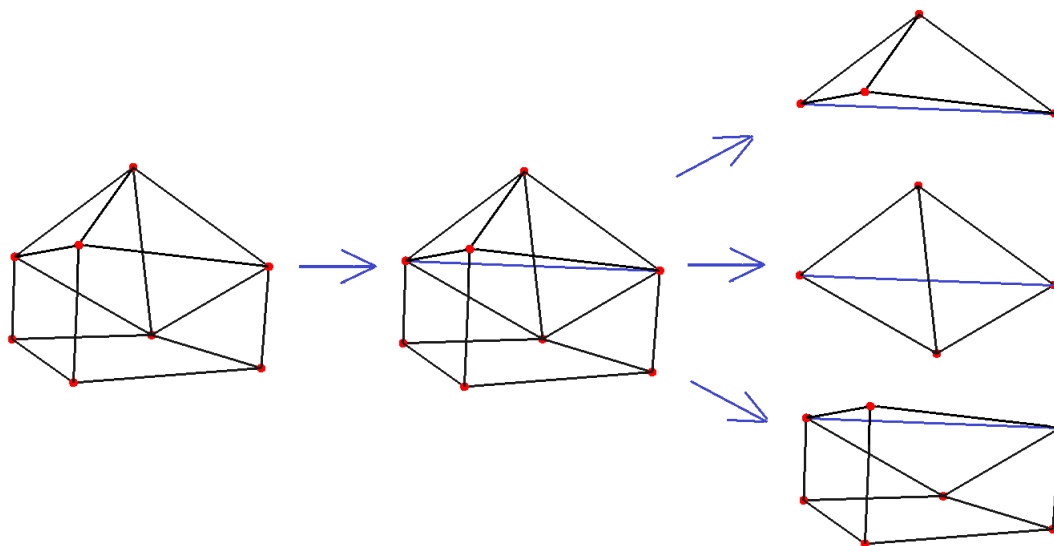


Figura 45: Separación del elemento genérico #6 en tres elementos, agregando una arista interna (azul).

Estos criterios se aplican sobre tres casos de estudio correspondientes a distintos tamaños del cuboide base:

- **Cuboide de dimensiones  $a = b = c$ :** El cubo 1-irregular aparece naturalmente en los octree estándar, siendo este método utilizado por la mayor parte de los generadores de mallas.
- **Cuboide de dimensiones  $4a = 2b = c$ :** Este corresponde a un ejemplo de un cuboide representativo de generadores de malla que no usan sólo cubos, como el Generador  $\Omega$ .
- **Cuboide de dimensiones  $a\sqrt{2} = b = c$ :** Se ha demostrado en [9] que cualquier cuboide 1-irregular dentro de estas dimensiones puede ser teselado sin problemas para el método de volúmenes finitos.

Para cada caso se obtiene la frecuencia con que aparece cada elemento distinto.



### 6.2.1. Caso A: Cuboide de dimensiones $a = b = c$

**Primer criterio, teselaciones óptimas:** Se observa un total de 24 elementos coesféricos finales distintos, cuyas frecuencias se distribuyen de la siguiente forma:

Elemento	Frecuencia
Cuboide	195
Tetraedro	18.450
Pirámide de base cuadrilátera	11.718
Prisma Triangular	3.720
Complemento de Tetraedro	992
Prisma Deformado	396
Complemento de Tetraedro deformado	144
Pirámide Pentagonal	384
Pirámide Hexagonal	56
Bipirámide Triangular	240
Bipirámide Cuadrilátera	272
Bipirámide Pentagonal	192
Bipirámide Hexagonal	36
Biprisma Triangular	6
Genérico #1	12
Genérico #2	96
Genérico #3	48
Genérico #4	48
Genérico #5	120
Genérico #6	24
Genérico #7	48
Genérico #8	48
Genérico #9	6
Genérico #10	8
Total	37.259

De estos datos se observa que los elementos más representados corresponden a tetraedros y pirámides de base cuadrilátera ( $\sim 49,5\%$  y  $\sim 31,5\%$  del total de elementos, respectivamente). Además, el conjunto de 7 elementos coesféricos inicial representa un  $\sim 95,6\%$  del total.

**Segundo criterio, minimización sin adición de aristas internas:** La cantidad de elementos coesféricos finales distintos se reduce a 16, cuyas frecuencias se distribuyen de la siguiente forma:

Elemento	Frecuencia
Cuboide	201
Tetraedro	18.930
Pirámide de base cuadrilátera	12.484
Prisma Triangular	3.900
Complemento de Tetraedro	992
Prisma Deformado	396
Complemento de Tetraedro deformado	144
Pirámide Pentagonal	768
Pirámide Hexagonal	128
Genérico #1	12
Genérico #2	96
Genérico #3	48
Genérico #4	48
Genérico #6	24
Genérico #7	48
Genérico #10	8
Total	38.227

Nuevamente, los elementos más representados corresponden a tetraedros y pirámides de base cuadrilátera ( $\sim 49,5\%$  y  $\sim 32,7\%$  del total de elementos, respectivamente). El conjunto de 7 elementos coesféricos inicial representa un  $\sim 96,9\%$  del total.

**Tercer criterio, minimización permitiendo adición de aristas internas:** Finalmente, la cantidad de elementos coesféricos finales distintos se reduce a 13, cuyas frecuencias se distribuyen de la siguiente forma:

Elemento	Frecuencia
Cuboide	201
Tetraedro	19.170

Pirámide de base cuadrilátera	12.580
Prisma Triangular	3.900
Complemento de Tetraedro	1.016
Prisma Deformado	444
Complemento de Tetraedro deformado	144
Pirámide Pentagonal	768
Pirámide Hexagonal	128
Genérico #1	12
Genérico #2	96
Genérico #4	48
Genérico #10	8
Total	38.515

Los elementos más representados corresponden a tetraedros y pirámides de base cuadrilátera ( $\sim 49,8\%$  y  $\sim 32,7\%$  del total de elementos, respectivamente). El conjunto de 7 elementos coesféricos inicial representa un  $\sim 97,2\%$  del total.

### 6.2.2. Caso B: Cuboide de dimensiones $4a = 2b = c$

**Primer criterio, teselaciones óptimas:** Bajo estas condiciones, aparecen sólo 6 elementos coesféricos finales distintos, cuyas frecuencias se distribuyen de la siguiente forma:

Elemento	Frecuencia
Cuboide	103
Tetraedro	29.118
Pirámide de base cuadrilátera	12.620
Prisma Triangular	3.120
Complemento de Tetraedro	536
Prisma Deformado	84
Total	45.581

De estos datos se observa que los elementos más representados corresponden a tetraedros y pirámides de base cuadrilátera, sumando entre ambos más del  $90\%$  del total de los elementos

(~63,9% de tetraedros y ~27,7% de pirámides de base cuadrilátera).

Dado que no aparecen elementos separables bajo ninguno de los dos criterios restantes la distribución de elementos no varía, por lo que no se muestran los resultados asociados.

### 6.2.3. Caso C: Cuboide de dimensiones $a\sqrt{2} = b = c$

**Primer criterio, teselaciones óptimas:** En este caso de estudio se identifica un total de 10 elementos coesféricos finales distintos, cuyas frecuencias se distribuyen de la siguiente forma:

Elemento	Frecuencia
Cuboide	199
Tetraedro	25.252
Pirámide de base cuadrilátera	12.300
Prisma Triangular	3.780
Complemento de Tetraedro	1,008
Prisma Deformado	284
Bipirámide Triangular	128
Bipirámide Cuadrilátera	52
Genérico #2	16
Genérico #5	16
Total	43.035

Los elementos más representados corresponden a tetraedros y pirámides de base cuadrilátera (~58,7% y ~28,6% del total de elementos, respectivamente). Además, el conjunto de 7 elementos coesféricos inicial representa un ~99,5% del total.

**Segundo criterio, minimización sin adición de aristas internas:** La cantidad de elementos coesféricos finales distintos se reduce a 7, cuyas frecuencias se distribuyen de la siguiente forma:

Elemento	Frecuencia
----------	------------

Cuboide	199
Tetraedro	25.508
Pirámide de base cuadrilátera	12.420
Prisma Triangular	3.796
Complemento de Tetraedro	1,008
Prisma Deformado	284
Genérico #2	16
Total	43.231

Nuevamente, los elementos más representados corresponden a tetraedros y pirámides de base cuadrilátera ( $\sim 59,0\%$  y  $\sim 28,7\%$  del total de elementos, respectivamente). Sólo aparecen 6 de los 7 elementos coesféricos inicial, representados por un  $\sim 99,96\%$  del total.

Dado que este conjunto de 7 elementos no es separable bajo el tercer criterio se omiten estos resultados, pues la distribución de elementos no varía.

### 6.3. Calidad de las particiones generadas por bisección

Como parte de los resultados se estudia si las particiones generadas cumplen con el requerimiento para ser utilizado en el contexto del método de volúmenes finitos. Para esto se requiere que el circuncentro de cada elemento final se encuentre contenido dentro del cuboide base de la configuración 1-irregular (se acepta que el circuncentro esté fuera del elemento correspondiente, siempre que no salga del cuboide). Esto es para poder resolver el problema localmente para cada cuboide 1-irregular.

El análisis de calidad se realiza sobre los mismos tamaños base estudiados en la sección 6.2, obteniendo los siguientes resultados:

#### 6.3.1. Caso A: Cuboide de dimensiones $a = b = c$

En este caso se observa que los circuncentros de todos los elementos se encuentran al interior del cuboide base para las 4096 configuraciones, por lo que todas las configuraciones producen teselaciones válidas para usar con el método de volúmenes finitos.

### 6.3.2. Caso B: Cuboide de dimensiones $4a = 2b = c$

Observando las proporciones de este cuboide, se puede intuir que no corresponden a las de un caso ideal (el cuboide es demasiado “estirado”). Efectivamente, sólo 132 configuraciones producen teselaciones válidas. De las 3964 restantes: 347 tienen un circuncentro fuera, 739 tienen dos, 657 tienen 3, 1208 tienen cuatro, 599 tienen cinco y 414 tienen seis.

### 6.3.3. Caso C: Cuboide de dimensiones $a\sqrt{2} = b = c$

Al igual que en el Caso A, se encuentra que todas las configuraciones producen teselaciones válidas para el método de volúmenes finitos, verificando el resultado presentado en [9]. Sin embargo, en este caso existen configuraciones que contienen elementos cuyo circuncentro se ubica sobre el borde del cuboide base.

## 6.4. Elementos coesféricos adicionales identificados en configuraciones 1-irregulares generadas por intersección

A continuación se presentan las descripciones de los 14 elementos coesféricos finales encontrados en configuraciones generadas por intersección. Para obtener las figuras correspondientes, se modifica el Visualizador para dibujar exclusivamente una configuración en particular.

- **Elemento genérico #11:** Elemento de 7 vértices, 6 caras y 11 aristas. Se compone de 2 caras triangulares y 4 caras cuadriláteras. Un ejemplo de este elemento se puede apreciar en la Figura 46.

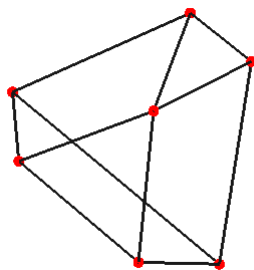


Figura 46: Elemento genérico #11.

- **Elemento genérico #12:** Elemento de 7 vértices, 6 caras y 11 aristas. Se compone de 3 caras triangulares, 2 caras cuadriláteras y una cara pentagonal. Un ejemplo de este elemento se puede apreciar en la Figura 47.

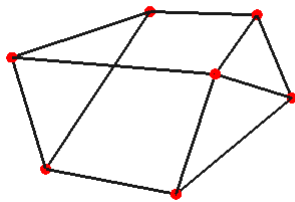


Figura 47: Elemento genérico #12.

- **Elemento genérico #13:** Elemento de 7 vértices, 7 caras y 12 aristas. Se compone de 4 caras triangulares y 3 caras cuadriláteras. Un Complemento de Cuboide cumple con esta descripción, pero se considera un elemento distinto ya que se explicita el hecho de que sus caras cuadriláteras son rectangulares. Un ejemplo de este elemento se puede apreciar en la Figura 48.

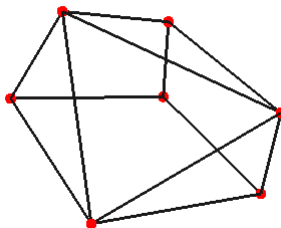


Figura 48: Elemento genérico #13.

- **Elemento genérico #14:** Elemento de 7 vértices, 8 caras y 13 aristas. Se compone de 7 caras triangulares y 1 cara pentagonal. Un ejemplo de este elemento se puede apreciar en la Figura 49.

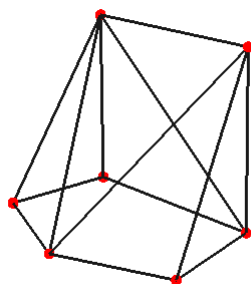


Figura 49: Elemento genérico #14.

- **Elemento genérico #15:** Elemento de 7 vértices, 9 caras y 14 aristas. Se compone de 8 caras triangulares y 1 cara cuadrilátera. Un ejemplo de este elemento se puede apreciar en la Figura 50.

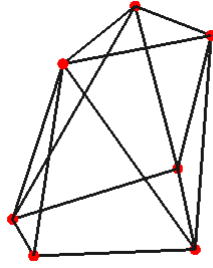


Figura 50: Elemento genérico #15.

- **Elemento genérico #16:** Elemento de 8 vértices, 6 caras y 12 aristas. Se compone de 2 caras triangulares, 2 caras cuadriláteras y 2 caras pentagonales. Un ejemplo de este elemento se puede apreciar en la Figura 51.

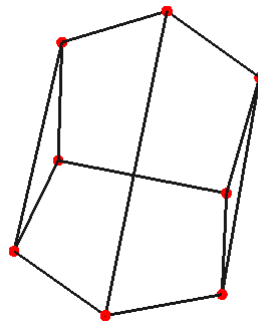


Figura 51: Elemento genérico #16.

- **Elemento genérico #17:** Elemento de 8 vértices, 6 caras y 12 aristas. Se compone de 6 caras cuadriláteras. Si bien un Cuboide cumple con esta descripción, se considera un elemento distinto ya que se hace explícito el hecho de que sus caras cuadriláteras son rectángulos. Un ejemplo de este elemento se puede apreciar en la Figura 52.

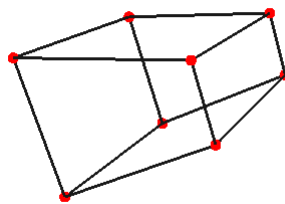


Figura 52: Elemento genérico #17.

- **Elemento genérico #18:** Elemento de 8 vértices, 7 caras y 13 aristas. Se compone de 4 caras triangulares, 2 caras cuadriláteras y 1 cara hexagonal. Un ejemplo de este elemento se puede apreciar en la Figura 53.



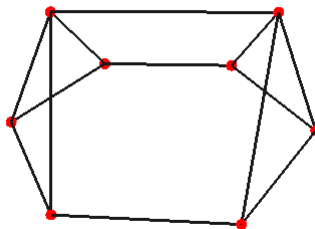


Figura 53: Elemento genérico #18.

- **Elemento genérico #19:** Elemento de 8 vértices, 8 caras y 14 aristas. Se compone de 5 caras triangulares, 2 caras cuadriláteras y 1 cara pentagonal. Un ejemplo de este elemento se puede apreciar en la Figura 54.

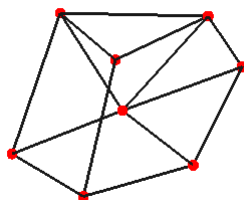


Figura 54: Elemento genérico #19.

- **Elemento genérico #20:** Elemento de 8 vértices, 9 caras y 15 aristas. Se compone de 7 caras triangulares, 1 cara cuadrilátera y 1 cara pentagonal. Un ejemplo de este elemento se puede apreciar en la Figura 55.

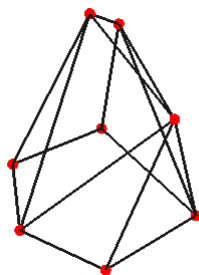


Figura 55: Elemento genérico #20.

- **Elemento genérico #21:** Elemento de 9 vértices, 8 caras y 15 aristas. Se compone de 3 caras triangulares, 4 caras cuadriláteras y 1 cara pentagonal. Un ejemplo de este elemento se puede apreciar en la Figura 56.

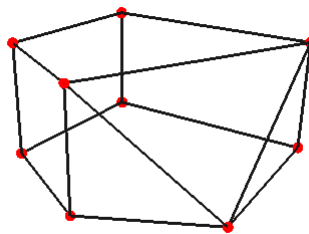


Figura 56: Elemento genérico #21.

- **Elemento genérico #22:** Elemento de 9 vértices, 8 caras y 15 aristas. Se compone de 4 caras triangulares, 3 caras cuadriláteras y 1 cara hexagonal. En la Figura 57 se puede apreciar un ejemplo de este elemento, el cual corresponde al poliedro conocido como Cúpula Triangular. Sin embargo, como no se asegura que todos los elementos encontrados de este tipo cumplen con el hecho de que la cara hexagonal es paralela a una de las caras triangulares, se decide mantener la denominación de este elemento como genérico.

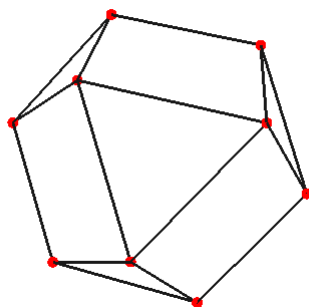


Figura 57: Elemento genérico #22.

- **Elemento genérico #23:** Elemento de 9 vértices, 9 caras y 13 aristas. Se compone de 5 caras triangulares, 3 caras cuadriláteras y 1 cara pentagonal. Un ejemplo de este elemento se puede apreciar en la Figura 58.

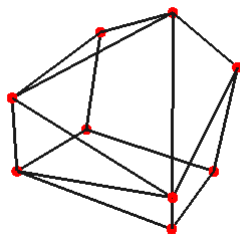


Figura 58: Elemento genérico #23.

- **Elemento genérico #24:** Elemento de 6 vértices, 6 caras y 10 aristas. Se compone de 6 caras triangulares, 1 cara cuadrilátera y 2 caras pentagonales. Un ejemplo de este elemento se puede apreciar en la Figura 59.

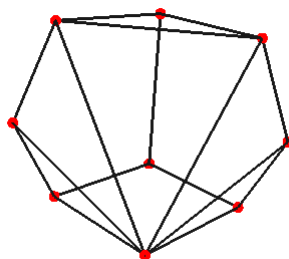


Figura 59: Elemento genérico #24.

## 6.5. Resumen de configuraciones 1-irregulares generadas por intersección

A continuación se presenta un resumen de los resultados obtenidos al generar las teselaciones de todas las configuraciones 1-irregulares sobre un cuboide de lados iguales:

Elemento	Frecuencia
Cuboide	531
Tetraedro	39.590.100
Pirámide de base cuadrilátera	5.200.926
Prisma Triangular	184.374
Complemento de Tetraedro	11.220
Prisma Deformado	84.200
Complemento de Tetraedro deformado	14.701
Pirámide Pentagonal	171.838
Pirámide Hexagonal	7.353
Bipirámide Triangular	625.447
Bipirámide Cuadrilátera	139.851
Bipirámide Pentagonal	25.686
Bipirámide Hexagonal	1.755
Biprisma	148
Genérico #1	61.044
Genérico #2	186.594
Genérico #3	94.020
Genérico #4	28.218
Genérico #5	28.028

Genérico #6	58
Genérico #7	1.881
Genérico #8	2.340
Genérico #9	6
Genérico #10	108
Genérico #11	6.236
Genérico #12	9.288
Genérico #13	3.972
Genérico #14	874
Genérico #15	4.966
Genérico #16	146
Genérico #17	204
Genérico #18	1.361
Genérico #19	4.033
Genérico #20	197
Genérico #21	42
Genérico #22	214
Genérico #23	4
Genérico #24	6
Total	46.491.970

Siguiendo la tendencia observada en las configuraciones 1-irregulares generadas por bisección, se tiene que los elementos más representados corresponden a tetraedros y pirámides de base cuadrilátera ( $\sim 85,15\%$  y  $\sim 11,19\%$  del total de elementos respectivamente, correspondientes a más del  $96\%$ ). El conjunto de 7 elementos inicial representa un  $\sim 96,98\%$ , mientras que el conjunto de 24 elementos encontrados en configuraciones generadas por bisección abarca un  $\sim 99,93\%$ . Finalmente, se tiene que los elementos que aparecen exclusivamente en configuraciones generadas por intersección corresponden sólo a un  $\sim 0,07\%$  del total.

## 7. Conclusiones y Trabajo futuro

En este trabajo, se desarrollaron dos herramientas, un generador de datos y un visualizador, que permiten el estudio de teselaciones de Delanay sobre cuboides 1-irregulares generados tanto por refinamiento basado en bisección como intersección.

Para el generador de datos, se diseñó y construyó un algoritmo que permite encontrar la partición en elementos finales coesféricos de cualquier configuración, especificando sus dimensiones y las posiciones de sus puntos de Steiner. Esto permite la identificación de todos los elementos coesféricos finales que pueden aparecer: 24 elementos finales en configuraciones generadas por bisección, número que es extendido a 38 en configuraciones generadas por intersección.

El visualizador implementado permite un estudio en detalle de las configuraciones 1-irregulares generadas por bisección, permitiendo especificar una configuración en específico bajo distintas dimensiones del cuboide base y distintas opciones de visualización, como examinar la partición resultante en modo wireframe y las posiciones de los centroides y circuncentros de cada elemento. Para esto se desarrolló una interfaz que incluye cinco componentes principales que permiten controlar el comportamiento de la aplicación y el despliegue de las configuraciones y sus teselaciones.

Usando el generador de datos, obtuvieron estadísticas asociadas a las particiones de todas las configuraciones bajo los distintos enfoques (bisección e intersección), encontrándose algunos resultados interesantes:

- Dos de las proporciones estudiadas (cuboide de lados iguales y cuboide de lados  $a\sqrt{2} = b = c$ ) en configuraciones 1-irregulares de tipo bisección generan teselaciones óptimas para ser utilizadas por el método de volúmenes finitos. En particular los resultados encontrados para un cubo hacen que un generador de mallas basado en octrees sea un candidato perfecto para ser integrado con el algoritmo desarrollado.
- Considerando los criterios establecidos para reducir el número de elementos coesféricos finales sobre configuraciones 1-irregulares generadas por bisección se tiene que, para un cuboide de lados iguales, los 24 elementos finales distintos se reducen a 16 si se permite la descomposición directa de elementos coesféricos en dos o más, y a 13 si además se permite la agregación de aristas internas. Para las otras proporciones estudiadas el impacto es menor: el conjunto de elementos finales para un cuboide de proporciones  $a\sqrt{2} = b = c$  se reduce de 10 a 7 al permitir descomposición directa, cantidad que no se ve afectada al incluir aristas internas, mientras que el conjunto de 6 elementos finales para un cuboide de proporciones  $4a = 2b = c$  es imposible de reducir.

- Considerando la totalidad de las configuraciones 1-irregulares sobre un cuboide de lados iguales, se tiene que los 7 elementos originales representan más del 95 % de los elementos en el caso de configuraciones generadas por bisección, y cerca del 97 % de los elementos en el caso de configuraciones generadas por intersección.
- Si bien existen 14 elementos finales adicionales en configuraciones de tipo intersección, estos sólo representan menos del 0,07 % del total de elementos, haciendo que la inclusión de estos elementos como finales en el contexto de un generador de mallas sea un esfuerzo que puede no valer la pena, considerando que se aumenta el conjunto de elementos finales en un  $\sim 58\%$  (de 24 a 38).

En cuanto al trabajo a futuro, existen al menos tres líneas de acción:

- Incorporar las configuraciones 1-irregulares generadas por intersección al visualizador. Dado que el algoritmo de generación de teselaciones y los métodos de dibujo fueron implementados de forma que sean independientes del tipo de configuración 1-irregular, incorporar esta extensión no requiere un rediseño general de la aplicación, tan sólo basta con definir una forma de identificar a cada configuración. Una propuesta es indicar los puntos de Steiner directamente sobre el visor de configuraciones del Visualizador.
- Incorporar el algoritmo desarrollado al Generador de Mallas  $\Omega$ . Dado que el algoritmo es independiente de la especificación de una configuración irregular, es posible aplicarlo incluso sobre un conjunto de puntos arbitrarios. En particular, este conjunto puede estar compuesto por la descripción de la geometría del objeto a modelar, diferenciándose de un enfoque basado en octrees. Sin embargo, este proceso no puede ser aplicado directamente, pues existen consideraciones adicionales a tener en cuenta, como la regiones que delimitan las zonas de distinto material de la geometría, pues allí si es necesario que los circuncentros de los elementos ubicados al borde de estas regiones estén dentro de la zona que los contiene.
- Realizar un estudio exhaustivo sobre las teselaciones posibles de las configuraciones 1-irregulares generadas por bisección: se tiene una cota superior para la cantidad de teselaciones de una configuración (recordando que una configuración se puede teselar de distintas formas dependiendo de las proporciones del cuboide base), y una forma de generar la teselación de cualquier configuración bajo distintos tamaños de las aristas del cuboide base. Pero se desconoce cuáles son las proporciones que generan distintas teselaciones. Encontrar estas proporciones permitirá caracterizar completamente a este tipo de configuraciones.

## Referencias

- [1] N. Hitschfeld, Generation 3D of mixed element meshes using a flexible refinement approach. *Engineering with Computers*, 21(2):101-114, 2005.
- [2] N. Hitschfeld, Generalization of Modified Octrees for Geometric Modeling. *Geometric Modeling: Theory and Practice. The state of the art*, Springer-Verlag, pp 260-272, 1997 (Blaubeuren II'96)
- [3] N. Hitschfeld, G. Navarro, R. Farías. 1-irregular configurations in cuboids. *Proceedings of the 9th Annual International Meshing Roundtable. 2-5 Octubre 2000, New Orleans, U.S.A*, pp: 275-282.
- [4] Proyecto Fondecyt N°1120495: Improving the functionality and performance of meshing tools. 2012-2015. Investigador responsable: Nancy Hitschfeld K.
- [5] Software Qhull.  
Página principal: <http://www.qhull.org/>
- [6] Librería PStreams.  
Página principal: <http://pstreams.sourceforge.net/>
- [7] Framework Qt.  
Página principal: <http://qt.nokia.com/products/>
- [8] Librería libQGLViewer.  
Página principal: <http://www.libqglviewer.com/>
- [9] P. Conti, Grid Generation for Three-dimensional Semiconductor Device Simulations. *Series in Microelectronics, Volume 12*. Hartung-Gorre. 1991





# A. Anexo: Diagrama de Clases

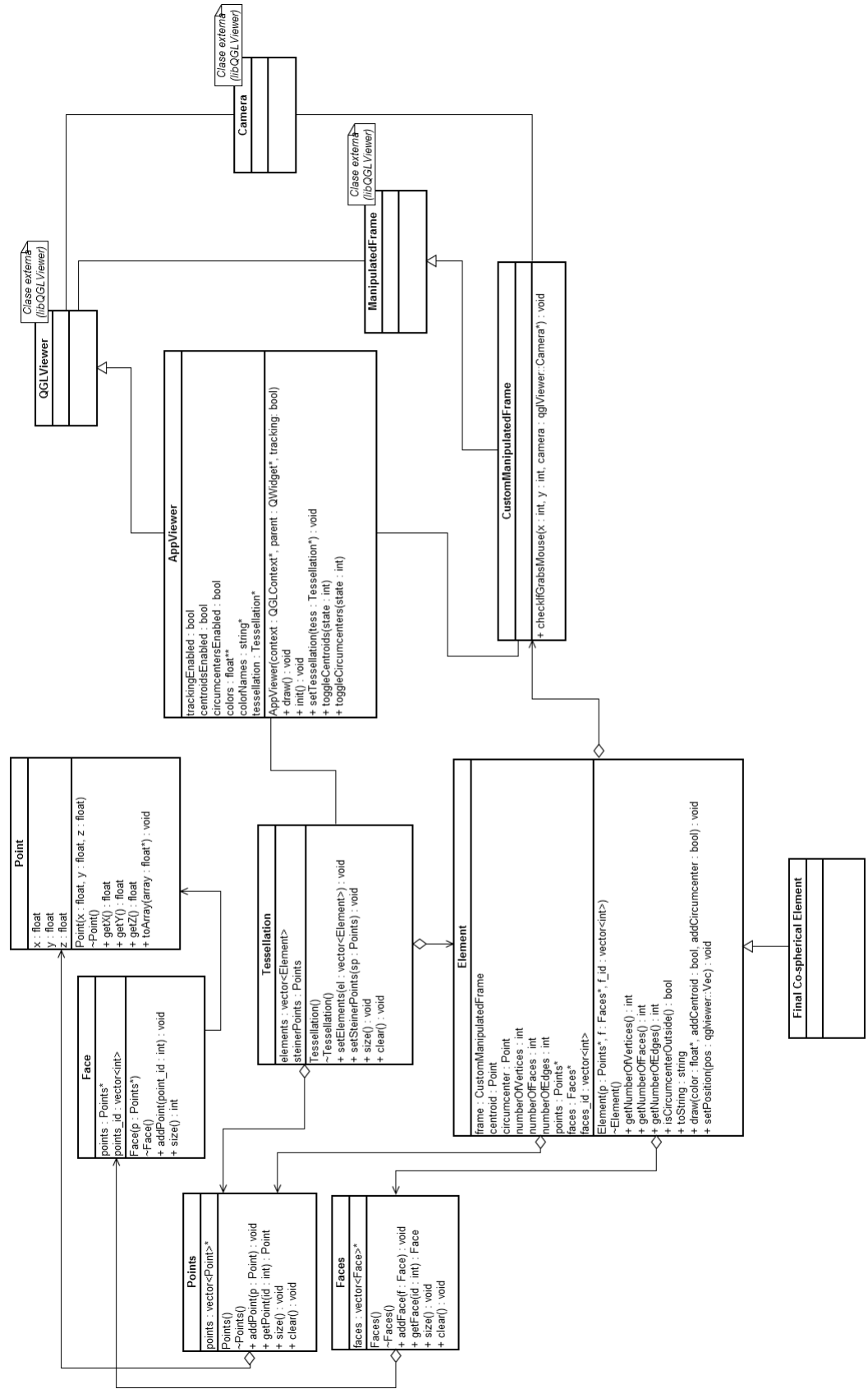


Figura 60: Diagrama de clases extendido

## B. Anexo: Clases de equivalencia de configuraciones 1-irregulares generadas por bisección

ID	Configuraciones en la clase de equivalencia
001	$C_0$
002	$C_1 C_8 C_{16} C_2 C_{64} C_4 C_{32} C_{512} C_{128} C_{1024} C_{2048} C_{256}$
003	$C_3 C_9 C_{24} C_{18} C_{80} C_6 C_{33} C_{520} C_{66} C_5 C_{40} C_{528} C_{1152} C_{3072} C_{2304} C_{384} C_{2112} C_{260} C_{160}$ $C_{1536} C_{320} C_{132} C_{1056} C_{2560}$
004	$C_7 C_{41} C_{536} C_{82} C_{1184} C_{3584} C_{2368} C_{388}$
005	$C_{10} C_{17} C_{576} C_{68} C_{36} C_{544} C_{258} C_{129} C_{1032} C_{2064} C_{1280} C_{2176}$
006	$C_{11} C_{25} C_{26} C_{19} C_{592} C_{70} C_{37} C_{552} C_{322} C_{133} C_{1064} C_{2576} C_{1408} C_{3200} C_{3328} C_{2432} C_{2624}$ $C_{324} C_{164} C_{1568} C_{262} C_{161} C_{1544} C_{2128}$
007	$C_{12} C_{48} C_{514} C_{65} C_{272} C_{130} C_{1025} C_{2056} C_{288} C_{640} C_{1088} C_{2052} C_{72} C_{20} C_{34} C_{513} C_{768} C_{192}$ $C_{1028} C_{2080} C_{257} C_{136} C_{1040} C_{2050}$
008	$C_{13} C_{56} C_{530} C_{67} C_{336} C_{134} C_{1057} C_{2568} C_{416} C_{1664} C_{3136} C_{2308} C_{88} C_{22} C_{35} C_{521} C_{2816}$ $C_{448} C_{1156} C_{3104} C_{261} C_{168} C_{1552} C_{2114}$
009	$C_{14} C_{49} C_{522} C_{81} C_{578} C_{69} C_{44} C_{560} C_{274} C_{131} C_{1033} C_{2072} C_{1312} C_{2688} C_{1344} C_{2180} C_{74}$ $C_{21} C_{42} C_{529} C_{832} C_{196} C_{1060} C_{2592} C_{259} C_{137} C_{1048} C_{2066} C_{1792} C_{2240} C_{1284} C_{2208} C_{584}$ $C_{84} C_{38} C_{545} C_{2306} C_{385} C_{1160} C_{3088} C_{1600} C_{2116} C_{292} C_{672} C_{386} C_{1153} C_{3080} C_{2320}$
010	$C_{15} C_{57} C_{538} C_{83} C_{594} C_{71} C_{45} C_{568} C_{338} C_{135} C_{1065} C_{2584} C_{1440} C_{3712} C_{3392} C_{2436} C_{90}$ $C_{23} C_{43} C_{537} C_{2880} C_{452} C_{1188} C_{3616} C_{263} C_{169} C_{1560} C_{2130} C_{3840} C_{2496} C_{1412} C_{3232} C_{600}$ $C_{86} C_{39} C_{553} C_{2370} C_{389} C_{1192} C_{3600} C_{3648} C_{2372} C_{420} C_{1696} C_{390} C_{1185} C_{3592} C_{2384}$
011	$C_{27} C_{2640} C_{326} C_{165} C_{1576} C_{3456}$
012	$C_{28} C_{50} C_{515} C_{73} C_{2562} C_{321} C_{140} C_{1072} C_{276} C_{162} C_{1537} C_{2120} C_{2336} C_{896} C_{1216} C_{3076} C_{784}$ $C_{194} C_{1029} C_{2088} C_{2054} C_{289} C_{648} C_{1104}$
013	$C_{29} C_{58} C_{531} C_{75} C_{2578} C_{323} C_{141} C_{1080} C_{340} C_{166} C_{1569} C_{2632} C_{2464} C_{1920} C_{3264} C_{3332} C_{89}$ $C_{30} C_{51} C_{523} C_{2832} C_{450} C_{1157} C_{3112} C_{325} C_{172} C_{1584} C_{2626} C_{2944} C_{1472} C_{3204} C_{3360} C_{2136}$ $C_{278} C_{163} C_{1545} C_{2118} C_{293} C_{680} C_{1616} C_{3152} C_{2310} C_{417} C_{1672} C_{198} C_{1061} C_{2600} C_{848}$
014	$C_{31} C_{59} C_{539} C_{91} C_{2642} C_{327} C_{173} C_{1592} C_{342} C_{167} C_{1577} C_{2648} C_{3488} C_{3968} C_{3520} C_{3460}$ $C_{2896} C_{454} C_{1189} C_{3624} C_{2374} C_{421} C_{1704} C_{3664}$
015	$C_{46} C_{561} C_{586} C_{85} C_{2322} C_{387} C_{1161} C_{3096} C_{1316} C_{2720} C_{1856} C_{2244}$
016	$C_{47} C_{569} C_{602} C_{87} C_{2386} C_{391} C_{1193} C_{3608} C_{1444} C_{3744} C_{3904} C_{2500}$
017	$C_{52} C_{546} C_{577} C_{76} C_{2058} C_{273} C_{138} C_{1041} C_{2068} C_{290} C_{641} C_{1096} C_{2084} C_{800} C_{704} C_{1092} C_{1296}$ $C_{2178} C_{1281} C_{2184} C_{193} C_{1036} C_{2096} C_{770}$
018	$C_{53} C_{554} C_{593} C_{78} C_{2074} C_{275} C_{139} C_{1049} C_{2132} C_{294} C_{673} C_{1608} C_{2212} C_{1824} C_{2752} C_{1348}$ $C_{3344} C_{2434} C_{1409} C_{3208} C_{197} C_{1068} C_{2608} C_{834}$

019	$C_{54} C_{547} C_{585} C_{92} C_{2122} C_{277} C_{170} C_{1553} C_{2070} C_{291} C_{649} C_{1112} C_{3108} C_{2848} C_{960} C_{1220} C_{579}$ $C_{77} C_{60} C_{562} C_{1360} C_{2182} C_{1313} C_{2696} C_{195} C_{1037} C_{2104} C_{786} C_{1728} C_{3140} C_{2340} C_{928} C_{2570}$ $C_{337} C_{142} C_{1073} C_{2324} C_{418} C_{1665} C_{3144} C_{1808} C_{2242} C_{1285} C_{2216} C_{449} C_{1164} C_{3120} C_{2818}$
020	$C_{55} C_{555} C_{601} C_{94} C_{2138} C_{279} C_{171} C_{1561} C_{2134} C_{295} C_{681} C_{1624} C_{3236} C_{3872} C_{3008} C_{1476}$ $C_{595} C_{79} C_{61} C_{570} C_{3408} C_{2438} C_{1441} C_{3720} C_{199} C_{1069} C_{2616} C_{850} C_{3776} C_{3396} C_{2468} C_{1952}$ $C_{2586} C_{339} C_{143} C_{1081} C_{2388} C_{422} C_{1697} C_{3656} C_{3856} C_{2498} C_{1413} C_{3240} C_{453} C_{1196} C_{3632}$ $C_{2882}$
021	$C_{62} C_{563} C_{587} C_{93} C_{2634} C_{341} C_{174} C_{1585} C_{2326} C_{419} C_{1673} C_{3160} C_{3364} C_{2976} C_{1984} C_{3268}$ $C_{1872} C_{2246} C_{1317} C_{2728} C_{451} C_{1165} C_{3128} C_{2834}$
022	$C_{63} C_{571} C_{603} C_{95} C_{2650} C_{343} C_{175} C_{1593} C_{2390} C_{423} C_{1705} C_{3672} C_{3492} C_{4000} C_{4032} C_{3524}$ $C_{3920} C_{2502} C_{1445} C_{3752} C_{455} C_{1197} C_{3640} C_{2898}$
023	$C_{96} C_{516} C_{264} C_{144} C_{1026} C_{2049}$
024	$C_{97} C_{524} C_{112} C_{518} C_{280} C_{146} C_{1027} C_{2057} C_{2113} C_{268} C_{176} C_{1538} C_{644} C_{1120} C_{2564} C_{352}$ $C_{532} C_{98} C_{517} C_{104} C_{3074} C_{2305} C_{392} C_{1168} C_{148} C_{1058} C_{2561} C_{328} C_{2144} C_{772} C_{224} C_{1540}$ $C_{1042} C_{2051} C_{265} C_{152} C_{208} C_{1030} C_{2081} C_{776} C_{2312} C_{400} C_{1154} C_{3073} C_{2053} C_{296} C_{656} C_{1090}$
025	$C_{99} C_{525} C_{120} C_{534} C_{344} C_{150} C_{1059} C_{2569} C_{2115} C_{269} C_{184} C_{1554} C_{1668} C_{3168} C_{2820} C_{480}$ $C_{3138} C_{2309} C_{424} C_{1680} C_{464} C_{1158} C_{3105} C_{2824}$
026	$C_{100} C_{548} C_{608} C_{580} C_{266} C_{145} C_{1034} C_{2065} C_{1282} C_{2177} C_{1288} C_{2192}$
027	$C_{101} C_{556} C_{624} C_{582} C_{282} C_{147} C_{1035} C_{2073} C_{2129} C_{270} C_{177} C_{1546} C_{676} C_{1632} C_{2628} C_{356}$ $C_{596} C_{102} C_{549} C_{616} C_{3330} C_{2433} C_{1416} C_{3216} C_{149} C_{1066} C_{2577} C_{330} C_{2656} C_{836} C_{228} C_{1572}$ $C_{1050} C_{2067} C_{267} C_{153} C_{2256} C_{1286} C_{2209} C_{1800} C_{3336} C_{2448} C_{1410} C_{3201} C_{2181} C_{1320} C_{2704}$ $C_{1346}$
028	$C_{103} C_{557} C_{632} C_{598} C_{346} C_{151} C_{1067} C_{2585} C_{2131} C_{271} C_{185} C_{1562} C_{1700} C_{3680} C_{2884} C_{484}$ $C_{3394} C_{2437} C_{1448} C_{3728} C_{2512} C_{1414} C_{3233} C_{3848}$
029	$C_{105} C_{540} C_{114} C_{519} C_{792} C_{210} C_{1031} C_{2089} C_{2369} C_{396} C_{1200} C_{3586} C_{900} C_{1248} C_{3588} C_{2400}$ $C_{404} C_{1186} C_{3585} C_{2376} C_{1106} C_{2055} C_{297} C_{664}$
030	$C_{106} C_{533} C_{840} C_{212} C_{1062} C_{2593} C_{2307} C_{393} C_{1176} C_{3090} C_{1796} C_{2272} C_{526} C_{113} C_{1602} C_{2117}$ $C_{300} C_{688} C_{402} C_{1155} C_{3081} C_{2328} C_{1376} C_{2692}$
031	$C_{107} C_{541} C_{122} C_{535} C_{856} C_{214} C_{1063} C_{2601} C_{2371} C_{397} C_{1208} C_{3602} C_{1924} C_{3296} C_{3844} C_{2528}$ $C_{542} C_{115} C_{527} C_{121} C_{3650} C_{2373} C_{428} C_{1712} C_{406} C_{1187} C_{3593} C_{2392} C_{3424} C_{2948} C_{1504} C_{3716}$ $C_{1618} C_{2119} C_{301} C_{696} C_{466} C_{1159} C_{3113} C_{2840} C_{2888} C_{468} C_{1190} C_{3617} C_{2311} C_{425} C_{1688} C_{3154}$
032	$C_{108} C_{564} C_{610} C_{581} C_{778} C_{209} C_{1038} C_{2097} C_{2321} C_{394} C_{1169} C_{3082} C_{804} C_{736} C_{1604} C_{2148}$ $C_{588} C_{116} C_{550} C_{609} C_{1794} C_{2241} C_{1292} C_{2224} C_{401} C_{1162} C_{3089} C_{2314} C_{864} C_{708} C_{1124} C_{2596}$ $C_{1098} C_{2069} C_{298} C_{657} C_{2194} C_{1283} C_{2185} C_{1304} C_{1352} C_{2196} C_{1314} C_{2689} C_{2179} C_{1289} C_{2200}$ $C_{1298}$

033	$C_{109} C_{572} C_{626} C_{583} C_{794} C_{211} C_{1039} C_{2105} C_{2385} C_{398} C_{1201} C_{3594} C_{932} C_{1760} C_{3652} C_{2404}$ $C_{604} C_{118} C_{551} C_{617} C_{3842} C_{2497} C_{1420} C_{3248} C_{405} C_{1194} C_{3601} C_{2378} C_{2912} C_{964} C_{1252} C_{3620}$ $C_{1114} C_{2071} C_{299} C_{665} C_{2258} C_{1287} C_{2217} C_{1816} C_{3400} C_{2452} C_{1442} C_{3713} C_{2183} C_{1321} C_{2712}$ $C_{1362}$
034	$C_{110} C_{565} C_{618} C_{597} C_{842} C_{213} C_{1070} C_{2609} C_{2323} C_{395} C_{1177} C_{3098} C_{1828} C_{2784} C_{1860} C_{2276}$ $C_{590} C_{117} C_{558} C_{625} C_{1858} C_{2245} C_{1324} C_{2736} C_{403} C_{1163} C_{3097} C_{2330} C_{1888} C_{2756} C_{1380} C_{2724}$ $C_{1610} C_{2133} C_{302} C_{689} C_{2450} C_{1411} C_{3209} C_{3352} C_{1864} C_{2260} C_{1318} C_{2721} C_{2435} C_{1417} C_{3224}$ $C_{3346}$
035	$C_{111} C_{573} C_{634} C_{599} C_{858} C_{215} C_{1071} C_{2617} C_{2387} C_{399} C_{1209} C_{3610} C_{1956} C_{3808} C_{3908} C_{2532}$ $C_{606} C_{119} C_{559} C_{633} C_{3906} C_{2501} C_{1452} C_{3760} C_{407} C_{1195} C_{3609} C_{2394} C_{3936} C_{3012} C_{1508} C_{3748}$ $C_{1626} C_{2135} C_{303} C_{697} C_{2514} C_{1415} C_{3241} C_{3864} C_{3912} C_{2516} C_{1446} C_{3745} C_{2439} C_{1449} C_{3736}$ $C_{3410}$
036	$C_{123} C_{543} C_{2904} C_{470} C_{1191} C_{3625} C_{2375} C_{429} C_{1720} C_{3666} C_{3972} C_{3552}$
037	$C_{124} C_{566} C_{611} C_{589} C_{2826} C_{465} C_{1166} C_{3121} C_{2325} C_{426} C_{1681} C_{3146} C_{2852} C_{992} C_{1732} C_{3172}$ $C_{1810} C_{2243} C_{1293} C_{2232} C_{2198} C_{1315} C_{2697} C_{1368}$
038	$C_{125} C_{574} C_{627} C_{591} C_{2842} C_{467} C_{1167} C_{3129} C_{2389} C_{430} C_{1713} C_{3658} C_{2980} C_{2016} C_{3780} C_{3428}$ $C_{605} C_{126} C_{567} C_{619} C_{3858} C_{2499} C_{1421} C_{3256} C_{469} C_{1198} C_{3633} C_{2890} C_{3040} C_{1988} C_{3300} C_{3876}$ $C_{3162} C_{2327} C_{427} C_{1689} C_{2262} C_{1319} C_{2729} C_{1880} C_{3416} C_{2454} C_{1443} C_{3721} C_{2247} C_{1325} C_{2744}$ $C_{1874}$
039	$C_{127} C_{575} C_{635} C_{607} C_{2906} C_{471} C_{1199} C_{3641} C_{2391} C_{431} C_{1721} C_{3674} C_{4004} C_{4064} C_{4036} C_{3556}$ $C_{3922} C_{2503} C_{1453} C_{3768} C_{2518} C_{1447} C_{3753} C_{3928}$
040	$C_{154} C_{1043} C_{2059} C_{281} C_{2625} C_{332} C_{180} C_{1570} C_{774} C_{225} C_{1548} C_{2160} C_{3329} C_{2440} C_{1424} C_{3202}$ $C_{720} C_{1094} C_{2085} C_{808} C_{354} C_{645} C_{1128} C_{2580}$
041	$C_{155} C_{1051} C_{2075} C_{283} C_{2641} C_{334} C_{181} C_{1578} C_{838} C_{229} C_{1580} C_{2672} C_{3457} C_{3464} C_{3472} C_{3458}$ $C_{2768} C_{1350} C_{2213} C_{1832} C_{358} C_{677} C_{1640} C_{2644}$
042	$C_{156} C_{1074} C_{2563} C_{329} C_{788} C_{226} C_{1541} C_{2152} C_{2337} C_{904} C_{1232} C_{3078} C_{2121} C_{284} C_{178} C_{1539}$ $C_{912} C_{1218} C_{3077} C_{2344} C_{353} C_{652} C_{1136} C_{2566}$
043	$C_{157} C_{1082} C_{2579} C_{331} C_{852} C_{230} C_{1573} C_{2664} C_{2465} C_{1928} C_{3280} C_{3334} C_{2137} C_{286} C_{179} C_{1547}$ $C_{2960} C_{1474} C_{3205} C_{3368} C_{357} C_{684} C_{1648} C_{2630}$
044	$C_{158} C_{1075} C_{2571} C_{345} C_{2627} C_{333} C_{188} C_{1586} C_{790} C_{227} C_{1549} C_{2168} C_{3361} C_{2952} C_{1488} C_{3206}$ $C_{2123} C_{285} C_{186} C_{1555} C_{976} C_{1222} C_{3109} C_{2856} C_{355} C_{653} C_{1144} C_{2582} C_{1936} C_{3266} C_{3333} C_{2472}$ $C_{2633} C_{348} C_{182} C_{1571} C_{2822} C_{481} C_{1676} C_{3184} C_{1744} C_{3142} C_{2341} C_{936} C_{482} C_{1669} C_{3176} C_{2836}$
045	$C_{159} C_{1083} C_{2587} C_{347} C_{2643} C_{335} C_{189} C_{1594} C_{854} C_{231} C_{1581} C_{2680} C_{3489} C_{3976} C_{3536} C_{3462}$ $C_{2139} C_{287} C_{187} C_{1563} C_{3024} C_{1478} C_{3237} C_{3880} C_{359} C_{685} C_{1656} C_{2646} C_{3984} C_{3522} C_{3461}$ $C_{3496} C_{2649} C_{350} C_{183} C_{1579} C_{2886} C_{485} C_{1708} C_{3696} C_{3792} C_{3398} C_{2469} C_{1960} C_{486} C_{1701}$ $C_{3688} C_{2900}$
046	$C_{190} C_{1587} C_{2635} C_{349} C_{2838} C_{483} C_{1677} C_{3192} C_{3365} C_{2984} C_{2000} C_{3270}$

047	$C_{191} C_{1595} C_{2651} C_{351} C_{2902} C_{487} C_{1709} C_{3704} C_{3493} C_{4008} C_{4048} C_{3526}$
048	$C_{200} C_{1044} C_{2082} C_{769} C_{2060} C_{304} C_{642} C_{1089}$
049	$C_{201} C_{1052} C_{2098} C_{771} C_{785} C_{202} C_{1045} C_{2090} C_{833} C_{204} C_{1076} C_{2594} C_{897} C_{1224} C_{3092} C_{2338}$ $C_{2076} C_{306} C_{643} C_{1097} C_{2690} C_{1345} C_{2188} C_{1328} C_{308} C_{674} C_{1601} C_{2124} C_{2352} C_{898} C_{1217} C_{3084}$ $C_{1105} C_{2062} C_{305} C_{650} C_{706} C_{1093} C_{2092} C_{816} C_{2248} C_{1300} C_{2210} C_{1793} C_{2086} C_{801} C_{712} C_{1108}$
050	$C_{203} C_{1053} C_{2106} C_{787} C_{849} C_{206} C_{1077} C_{2602} C_{835} C_{205} C_{1084} C_{2610} C_{1921} C_{3272} C_{3348} C_{2466}$ $C_{2078} C_{307} C_{651} C_{1113} C_{2754} C_{1349} C_{2220} C_{1840} C_{310} C_{675} C_{1609} C_{2140} C_{3376} C_{2946} C_{1473}$ $C_{3212} C_{1617} C_{2126} C_{309} C_{682} C_{962} C_{1221} C_{3116} C_{2864} C_{2760} C_{1364} C_{2214} C_{1825} C_{2342} C_{929}$ $C_{1736} C_{3156}$
051	$C_{207} C_{1085} C_{2618} C_{851} C_{1953} C_{3784} C_{3412} C_{2470} C_{2142} C_{311} C_{683} C_{1625} C_{3010} C_{1477} C_{3244}$ $C_{3888}$
052	$C_{216} C_{1046} C_{2083} C_{777} C_{2817} C_{456} C_{1172} C_{3106} C_{773} C_{232} C_{1556} C_{2146} C_{2061} C_{312} C_{658} C_{1091}$ $C_{368} C_{646} C_{1121} C_{2572} C_{432} C_{1666} C_{3137} C_{2316}$
053	$C_{217} C_{1054} C_{2099} C_{779} C_{2833} C_{458} C_{1173} C_{3114} C_{837} C_{236} C_{1588} C_{2658} C_{2945} C_{1480} C_{3220}$ $C_{3362} C_{2077} C_{314} C_{659} C_{1099} C_{2706} C_{1347} C_{2189} C_{1336} C_{372} C_{678} C_{1633} C_{2636} C_{2480} C_{1922}$ $C_{3265} C_{3340} C_{3153} C_{2318} C_{433} C_{1674} C_{710} C_{1125} C_{2604} C_{880} C_{2264} C_{1302} C_{2211} C_{1801} C_{2150}$ $C_{805} C_{744} C_{1620}$
054	$C_{218} C_{1047} C_{2091} C_{793} C_{2881} C_{460} C_{1204} C_{3618} C_{775} C_{233} C_{1564} C_{2162} C_{3841} C_{2504} C_{1428} C_{3234}$ $C_{2063} C_{313} C_{666} C_{1107} C_{722} C_{1095} C_{2093} C_{824} C_{370} C_{647} C_{1129} C_{2588} C_{1456} C_{3714} C_{3393} C_{2444}$ $C_{3649} C_{2380} C_{436} C_{1698} C_{902} C_{1249} C_{3596} C_{2416} C_{728} C_{1110} C_{2087} C_{809} C_{2402} C_{901} C_{1256} C_{3604}$
055	$C_{219} C_{1055} C_{2107} C_{795} C_{2897} C_{462} C_{1205} C_{3626} C_{839} C_{237} C_{1596} C_{2674} C_{3969} C_{3528} C_{3476} C_{3490}$ $C_{2079} C_{315} C_{667} C_{1115} C_{2770} C_{1351} C_{2221} C_{1848} C_{374} C_{679} C_{1641} C_{2652} C_{3504} C_{3970} C_{3521}$ $C_{3468} C_{3665} C_{2382} C_{437} C_{1706} C_{966} C_{1253} C_{3628} C_{2928} C_{2776} C_{1366} C_{2215} C_{1833} C_{2406} C_{933}$ $C_{1768} C_{3668}$
056	$C_{220} C_{1078} C_{2595} C_{841} C_{2819} C_{457} C_{1180} C_{3122} C_{789} C_{234} C_{1557} C_{2154} C_{2849} C_{968} C_{1236} C_{3110}$ $C_{2125} C_{316} C_{690} C_{1603} C_{914} C_{1219} C_{3085} C_{2360} C_{369} C_{654} C_{1137} C_{2574} C_{944} C_{1730} C_{3141} C_{2348}$ $C_{3145} C_{2332} C_{434} C_{1667} C_{2694} C_{1377} C_{2700} C_{1392} C_{1240} C_{3094} C_{2339} C_{905} C_{2274} C_{1797} C_{2280}$ $C_{1812}$
057	$C_{221} C_{1086} C_{2611} C_{843} C_{2835} C_{459} C_{1181} C_{3130} C_{853} C_{238} C_{1589} C_{2666} C_{2977} C_{1992} C_{3284} C_{3366}$ $C_{2141} C_{318} C_{691} C_{1611} C_{2962} C_{1475} C_{3213} C_{3384} C_{373} C_{686} C_{1649} C_{2638} C_{2992} C_{1986} C_{3269}$ $C_{3372} C_{3161} C_{2334} C_{435} C_{1675} C_{2758} C_{1381} C_{2732} C_{1904} C_{3288} C_{3350} C_{2467} C_{1929} C_{2278} C_{1829}$ $C_{2792} C_{1876}$
058	$C_{222} C_{1079} C_{2603} C_{857} C_{2883} C_{461} C_{1212} C_{3634} C_{791} C_{235} C_{1565} C_{2170} C_{3873} C_{3016} C_{1492} C_{3238}$ $C_{2127} C_{317} C_{698} C_{1619} C_{978} C_{1223} C_{3117} C_{2872} C_{371} C_{655} C_{1145} C_{2590} C_{1968} C_{3778} C_{3397} C_{2476}$ $C_{3657} C_{2396} C_{438} C_{1699} C_{2950} C_{1505} C_{3724} C_{3440} C_{1752} C_{3158} C_{2343} C_{937} C_{2530} C_{1925} C_{3304}$ $C_{3860}$

059	$C_{223} C_{1087} C_{2619} C_{859} C_{2899} C_{463} C_{1213} C_{3642} C_{855} C_{239} C_{1597} C_{2682} C_{4001} C_{4040} C_{3540} C_{3494}$ $C_{2143} C_{319} C_{699} C_{1627} C_{3026} C_{1479} C_{3245} C_{3896} C_{375} C_{687} C_{1657} C_{2654} C_{4016} C_{4034} C_{3525}$ $C_{3500} C_{3673} C_{2398} C_{439} C_{1707} C_{3014} C_{1509} C_{3756} C_{3952} C_{3800} C_{3414} C_{2471} C_{1961} C_{2534} C_{1957}$ $C_{3816} C_{3924}$
060	$C_{240} C_{1542} C_{2145} C_{780} C_{2313} C_{408} C_{1170} C_{3075} C_{2565} C_{360} C_{660} C_{1122}$
061	$C_{241} C_{1550} C_{2161} C_{782} C_{2329} C_{410} C_{1171} C_{3083} C_{2629} C_{364} C_{692} C_{1634} C_{2693} C_{1384} C_{2708}$ $C_{1378} C_{2581} C_{362} C_{661} C_{1130} C_{3218} C_{3331} C_{2441} C_{1432} C_{244} C_{1574} C_{2657} C_{844} C_{2288} C_{1798}$ $C_{2273} C_{1804} C_{3091} C_{2315} C_{409} C_{1178} C_{724} C_{1126} C_{2597} C_{872} C_{2456} C_{1426} C_{3203} C_{3337} C_{2149}$ $C_{812} C_{752} C_{1606}$
062	$C_{242} C_{1543} C_{2153} C_{796} C_{2377} C_{412} C_{1202} C_{3587} C_{2567} C_{361} C_{668} C_{1138} C_{3589} C_{2408} C_{916} C_{1250}$ $C_{1234} C_{3079} C_{2345} C_{920} C_{1264} C_{3590} C_{2401} C_{908}$
063	$C_{243} C_{1551} C_{2169} C_{798} C_{2393} C_{414} C_{1203} C_{3595} C_{2631} C_{365} C_{700} C_{1650} C_{3717} C_{3432} C_{2964} C_{1506}$ $C_{2583} C_{363} C_{669} C_{1146} C_{3282} C_{3335} C_{2473} C_{1944} C_{246} C_{1575} C_{2665} C_{860} C_{3312} C_{3846} C_{2529}$ $C_{1932} C_{3603} C_{2379} C_{413} C_{1210} C_{980} C_{1254} C_{3621} C_{2920} C_{2968} C_{1490} C_{3207} C_{3369} C_{2405} C_{940}$ $C_{1776} C_{3654}$
064	$C_{245} C_{1582} C_{2673} C_{846} C_{2331} C_{411} C_{1179} C_{3099} C_{2645} C_{366} C_{693} C_{1642} C_{2725} C_{1896} C_{2772} C_{1382}$ $C_{3474} C_{3459} C_{3465} C_{3480} C_{2800} C_{1862} C_{2277} C_{1836}$
065	$C_{247} C_{1583} C_{2681} C_{862} C_{2395} C_{415} C_{1211} C_{3611} C_{2647} C_{367} C_{701} C_{1658} C_{3749} C_{3944} C_{3028} C_{1510}$ $C_{3538} C_{3463} C_{3497} C_{3992} C_{3824} C_{3910} C_{2533} C_{1964}$
066	$C_{248} C_{1558} C_{2147} C_{781} C_{2825} C_{472} C_{1174} C_{3107} C_{2821} C_{488} C_{1684} C_{3170} C_{2573} C_{376} C_{662} C_{1123}$ $C_{1682} C_{3139} C_{2317} C_{440} C_{496} C_{1670} C_{3169} C_{2828}$
067	$C_{249} C_{1566} C_{2163} C_{783} C_{2841} C_{474} C_{1175} C_{3115} C_{2885} C_{492} C_{1716} C_{3682} C_{2949} C_{1512} C_{3732}$ $C_{3426} C_{2589} C_{378} C_{663} C_{1131} C_{3730} C_{3395} C_{2445} C_{1464} C_{500} C_{1702} C_{3681} C_{2892} C_{2544} C_{1926}$ $C_{3297} C_{3852} C_{3155} C_{2319} C_{441} C_{1690} C_{726} C_{1127} C_{2605} C_{888} C_{2520} C_{1430} C_{3235} C_{3849} C_{2151}$ $C_{813} C_{760} C_{1622}$
068	$C_{250} C_{1559} C_{2155} C_{797} C_{2889} C_{476} C_{1206} C_{3619} C_{2823} C_{489} C_{1692} C_{3186} C_{3845} C_{2536} C_{1940}$ $C_{3298} C_{2575} C_{377} C_{670} C_{1139} C_{1746} C_{3143} C_{2349} C_{952} C_{498} C_{1671} C_{3177} C_{2844} C_{1520} C_{3718}$ $C_{3425} C_{2956} C_{3651} C_{2381} C_{444} C_{1714} C_{918} C_{1251} C_{3597} C_{2424} C_{984} C_{1238} C_{3111} C_{2857} C_{2403}$ $C_{909} C_{1272} C_{3606}$
069	$C_{251} C_{1567} C_{2171} C_{799} C_{2905} C_{478} C_{1207} C_{3627} C_{2887} C_{493} C_{1724} C_{3698} C_{3973} C_{3560} C_{3988}$ $C_{3554} C_{2591} C_{379} C_{671} C_{1147} C_{3794} C_{3399} C_{2477} C_{1976} C_{502} C_{1703} C_{3689} C_{2908} C_{3568} C_{3974}$ $C_{3553} C_{3980} C_{3667} C_{2383} C_{445} C_{1722} C_{982} C_{1255} C_{3629} C_{2936} C_{3032} C_{1494} C_{3239} C_{3881} C_{2407}$ $C_{941} C_{1784} C_{3670}$
070	$C_{252} C_{1590} C_{2659} C_{845} C_{2827} C_{473} C_{1182} C_{3123} C_{2837} C_{490} C_{1685} C_{3178} C_{2853} C_{1000} C_{1748}$ $C_{3174} C_{2637} C_{380} C_{694} C_{1635} C_{1938} C_{3267} C_{3341} C_{2488} C_{497} C_{1678} C_{3185} C_{2830} C_{1008} C_{1734}$ $C_{3173} C_{2860} C_{3147} C_{2333} C_{442} C_{1683} C_{2710} C_{1379} C_{2701} C_{1400} C_{1496} C_{3222} C_{3363} C_{2953} C_{2275}$ $C_{1805} C_{2296} C_{1814}$

071	$C_{253} C_{1598} C_{2675} C_{847} C_{2843} C_{475} C_{1183} C_{3131} C_{2901} C_{494} C_{1717} C_{3690} C_{2981} C_{2024} C_{3796}$ $C_{3430} C_{2653} C_{382} C_{695} C_{1643} C_{3986} C_{3523} C_{3469} C_{3512} C_{501} C_{1710} C_{3697} C_{2894} C_{3056} C_{1990}$ $C_{3301} C_{3884} C_{3163} C_{2335} C_{443} C_{1691} C_{2774} C_{1383} C_{2733} C_{1912} C_{3544} C_{3478} C_{3491} C_{3977} C_{2279}$ $C_{1837} C_{2808} C_{1878}$
072	$C_{254} C_{1591} C_{2667} C_{861} C_{2891} C_{477} C_{1214} C_{3635} C_{2839} C_{491} C_{1693} C_{3194} C_{3877} C_{3048} C_{2004}$ $C_{3302} C_{2639} C_{381} C_{702} C_{1651} C_{2002} C_{3271} C_{3373} C_{3000} C_{499} C_{1679} C_{3193} C_{2846} C_{2032} C_{3782}$ $C_{3429} C_{2988} C_{3659} C_{2397} C_{446} C_{1715} C_{2966} C_{1507} C_{3725} C_{3448} C_{2008} C_{3286} C_{3367} C_{2985} C_{2531}$ $C_{1933} C_{3320} C_{3862}$
073	$C_{255} C_{1599} C_{2683} C_{863} C_{2907} C_{479} C_{1215} C_{3643} C_{2903} C_{495} C_{1725} C_{3706} C_{4005} C_{4072} C_{4052}$ $C_{3558} C_{2655} C_{383} C_{703} C_{1659} C_{4050} C_{3527} C_{3501} C_{4024} C_{503} C_{1711} C_{3705} C_{2910} C_{4080} C_{4038}$ $C_{3557} C_{4012} C_{3675} C_{2399} C_{447} C_{1723} C_{3030} C_{1511} C_{3757} C_{3960} C_{4056} C_{3542} C_{3495} C_{4009} C_{2535}$ $C_{1965} C_{3832} C_{3926}$
074	$C_{504} C_{1686} C_{3171} C_{2829}$
075	$C_{505} C_{1694} C_{3187} C_{2831} C_{2845} C_{506} C_{1687} C_{3179} C_{2893} C_{508} C_{1718} C_{3683} C_{2957} C_{1528} C_{3734}$ $C_{3427} C_{2552} C_{1942} C_{3299} C_{3853} C_{1750} C_{3175} C_{2861} C_{1016}$
076	$C_{507} C_{1695} C_{3195} C_{2847} C_{2909} C_{510} C_{1719} C_{3691} C_{2895} C_{509} C_{1726} C_{3699} C_{3981} C_{3576} C_{3990}$ $C_{3555} C_{3798} C_{3431} C_{2989} C_{2040} C_{2006} C_{3303} C_{3885} C_{3064}$
077	$C_{511} C_{1727} C_{3707} C_{2911} C_{4013} C_{4088} C_{4054} C_{3559}$
078	$C_{612} C_{1290} C_{2193}$
079	$C_{613} C_{620} C_{628} C_{614} C_{1306} C_{2195} C_{1291} C_{2201} C_{2257} C_{1294} C_{2225} C_{1802} C_{740} C_{1636} C_{2660} C_{868}$ $C_{3338} C_{2449} C_{1418} C_{3217} C_{2197} C_{1322} C_{2705} C_{1354}$
080	$C_{615} C_{621} C_{636} C_{630} C_{1370} C_{2199} C_{1323} C_{2713} C_{2259} C_{1295} C_{2233} C_{1818} C_{1764} C_{3684} C_{2916}$ $C_{996} C_{3402} C_{2453} C_{1450} C_{3729} C_{2513} C_{1422} C_{3249} C_{3850}$
081	$C_{622} C_{629} C_{1866} C_{2261} C_{1326} C_{2737} C_{2451} C_{1419} C_{3225} C_{3354} C_{1892} C_{2788}$
082	$C_{623} C_{637} C_{638} C_{631} C_{1882} C_{2263} C_{1327} C_{2745} C_{2515} C_{1423} C_{3257} C_{3866} C_{2020} C_{3812} C_{3940}$ $C_{3044} C_{3914} C_{2517} C_{1454} C_{3761} C_{2455} C_{1451} C_{3737} C_{3418}$
083	$C_{639} C_{3930} C_{2519} C_{1455} C_{3769} C_{4068}$
084	$C_{705} C_{1100} C_{2100} C_{802} C_{1297} C_{2186}$
085	$C_{707} C_{1101} C_{2108} C_{818} C_{1361} C_{2190} C_{1329} C_{2698} C_{1729} C_{3148} C_{2356} C_{930} C_{2102} C_{803} C_{713} C_{1116}$ $C_{2250} C_{1301} C_{2218} C_{1809} C_{3124} C_{2850} C_{961} C_{1228}$
086	$C_{709} C_{1132} C_{2612} C_{866} C_{1299} C_{2187} C_{1305} C_{2202} C_{721} C_{1102} C_{2101} C_{810} C_{737} C_{1612} C_{2164} C_{806}$ $C_{2442} C_{1425} C_{3210} C_{3345} C_{2753} C_{1356} C_{2228} C_{1826}$
087	$C_{711} C_{1133} C_{2620} C_{882} C_{1363} C_{2191} C_{1337} C_{2714} C_{723} C_{1103} C_{2109} C_{826} C_{1761} C_{3660} C_{2420}$ $C_{934} C_{2166} C_{807} C_{745} C_{1628} C_{2506} C_{1429} C_{3242} C_{3857} C_{2103} C_{811} C_{729} C_{1118} C_{3636} C_{2914} C_{965}$ $C_{1260} C_{1817} C_{2266} C_{1303} C_{2219} C_{3009} C_{1484} C_{3252} C_{3874} C_{3722} C_{3409} C_{2446} C_{1457} C_{2484} C_{1954}$ $C_{3777} C_{3404}$
088	$C_{714} C_{1109} C_{2094} C_{817} C_{1857} C_{2252} C_{1332} C_{2722} C_{899} C_{1225} C_{3100} C_{2354}$

089	$C_{715} C_{1117} C_{2110} C_{819} C_{1873} C_{2254} C_{1333} C_{2730} C_{963} C_{1229} C_{3132} C_{2866} C_{1985} C_{3276} C_{3380}$ $C_{2978} C_{2762} C_{1365} C_{2222} C_{1841} C_{2358} C_{931} C_{1737} C_{3164}$
090	$C_{716} C_{1140} C_{2598} C_{865} C_{1795} C_{2249} C_{1308} C_{2226} C_{913} C_{1226} C_{3093} C_{2346} C_{2156} C_{820} C_{738} C_{1605}$ $C_{906} C_{1233} C_{3086} C_{2353} C_{1353} C_{2204} C_{1330} C_{2691}$
091	$C_{717} C_{1148} C_{2614} C_{867} C_{1811} C_{2251} C_{1309} C_{2234} C_{977} C_{1230} C_{3125} C_{2858} C_{993} C_{1740} C_{3188}$ $C_{2854} C_{2172} C_{822} C_{739} C_{1613} C_{2954} C_{1489} C_{3214} C_{3377} C_{2357} C_{938} C_{1745} C_{3150} C_{2868} C_{994}$ $C_{1733} C_{3180} C_{1369} C_{2206} C_{1331} C_{2699} C_{2755} C_{1357} C_{2236} C_{1842} C_{3274} C_{3349} C_{2474} C_{1937} C_{2230}$ $C_{1827} C_{2761} C_{1372}$
092	$C_{718} C_{1141} C_{2606} C_{881} C_{1859} C_{2253} C_{1340} C_{2738} C_{915} C_{1227} C_{3101} C_{2362} C_{1889} C_{2764} C_{1396}$ $C_{2726} C_{2158} C_{821} C_{746} C_{1621} C_{970} C_{1237} C_{3118} C_{2865} C_{2355} C_{907} C_{1241} C_{3102} C_{1844} C_{2786}$ $C_{1861} C_{2284} C_{1865} C_{2268} C_{1334} C_{2723} C_{2947} C_{1481} C_{3228} C_{3378} C_{1738} C_{3157} C_{2350} C_{945} C_{2482}$ $C_{1923} C_{3273} C_{3356}$
093	$C_{719} C_{1149} C_{2622} C_{883} C_{1875} C_{2255} C_{1341} C_{2746} C_{979} C_{1231} C_{3133} C_{2874} C_{2017} C_{3788} C_{3444}$ $C_{2982} C_{2174} C_{823} C_{747} C_{1629} C_{3018} C_{1493} C_{3246} C_{3889} C_{2359} C_{939} C_{1753} C_{3166} C_{3892} C_{3042}$ $C_{1989} C_{3308} C_{1881} C_{2270} C_{1335} C_{2731} C_{3011} C_{1485} C_{3260} C_{3890} C_{3786} C_{3413} C_{2478} C_{1969} C_{2486}$ $C_{1955} C_{3785} C_{3420}$
094	$C_{725} C_{1134} C_{2613} C_{874} C_{3347} C_{2443} C_{1433} C_{3226} C_{2785} C_{1868} C_{2292} C_{1830} C_{2165} C_{814} C_{753}$ $C_{1614} C_{2458} C_{1427} C_{3211} C_{3353} C_{2740} C_{1890} C_{2757} C_{1388}$
095	$C_{727} C_{1135} C_{2621} C_{890} C_{3411} C_{2447} C_{1465} C_{3738} C_{3809} C_{3916} C_{2548} C_{1958} C_{2167} C_{815} C_{761}$ $C_{1630} C_{2522} C_{1431} C_{3243} C_{3865} C_{3764} C_{3938} C_{3013} C_{1516}$
096	$C_{730} C_{1111} C_{2095} C_{825} C_{3905} C_{2508} C_{1460} C_{3746} C_{903} C_{1257} C_{3612} C_{2418}$
097	$C_{731} C_{1119} C_{2111} C_{827} C_{3921} C_{2510} C_{1461} C_{3754} C_{967} C_{1261} C_{3644} C_{2930} C_{4033} C_{3532} C_{3508}$ $C_{4002} C_{2778} C_{1367} C_{2223} C_{1849} C_{2422} C_{935} C_{1769} C_{3676}$
098	$C_{732} C_{1142} C_{2599} C_{873} C_{3843} C_{2505} C_{1436} C_{3250} C_{917} C_{1258} C_{3605} C_{2410} C_{2913} C_{972} C_{1268}$ $C_{3622} C_{2157} C_{828} C_{754} C_{1607} C_{922} C_{1235} C_{3087} C_{2361} C_{2417} C_{910} C_{1265} C_{3598} C_{948} C_{1762} C_{3653}$ $C_{2412} C_{3401} C_{2460} C_{1458} C_{3715} C_{2695} C_{1385} C_{2716} C_{1394} C_{1242} C_{3095} C_{2347} C_{921} C_{2290} C_{1799}$ $C_{2281} C_{1820}$
099	$C_{733} C_{1150} C_{2615} C_{875} C_{3859} C_{2507} C_{1437} C_{3258} C_{981} C_{1262} C_{3637} C_{2922} C_{3041} C_{1996} C_{3316}$ $C_{3878} C_{2173} C_{830} C_{755} C_{1615} C_{2970} C_{1491} C_{3215} C_{3385} C_{2421} C_{942} C_{1777} C_{3662} C_{2996} C_{2018}$ $C_{3781} C_{3436} C_{3417} C_{2462} C_{1459} C_{3723} C_{2759} C_{1389} C_{2748} C_{1906} C_{3290} C_{3351} C_{2475} C_{1945} C_{2294}$ $C_{1831} C_{2793} C_{1884}$
100	$C_{734} C_{1143} C_{2607} C_{889} C_{3907} C_{2509} C_{1468} C_{3762} C_{919} C_{1259} C_{3613} C_{2426} C_{3937} C_{3020} C_{1524}$ $C_{3750} C_{2159} C_{829} C_{762} C_{1623} C_{986} C_{1239} C_{3119} C_{2873} C_{2419} C_{911} C_{1273} C_{3614} C_{1972} C_{3810}$ $C_{3909} C_{2540} C_{3913} C_{2524} C_{1462} C_{3747} C_{2951} C_{1513} C_{3740} C_{3442} C_{1754} C_{3159} C_{2351} C_{953} C_{2546}$ $C_{1927} C_{3305} C_{3868}$



101	$C_{735} C_{1151} C_{2623} C_{891} C_{3923} C_{2511} C_{1469} C_{3770} C_{983} C_{1263} C_{3645} C_{2938} C_{4065} C_{4044} C_{3572}$ $C_{4006} C_{2175} C_{831} C_{763} C_{1631} C_{3034} C_{1495} C_{3247} C_{3897} C_{2423} C_{943} C_{1785} C_{3678} C_{4020} C_{4066}$ $C_{4037} C_{3564} C_{3929} C_{2526} C_{1463} C_{3755} C_{3015} C_{1517} C_{3772} C_{3954} C_{3802} C_{3415} C_{2479} C_{1977} C_{2550}$ $C_{1959} C_{3817} C_{3932}$
102	$C_{741} C_{1644} C_{2676} C_{870} C_{1307} C_{2203} C_{2769} C_{1358} C_{2229} C_{1834} C_{3466} C_{3473}$
103	$C_{742} C_{1637} C_{2668} C_{884} C_{1355} C_{2205} C_{1338} C_{2707} C_{2662} C_{869} C_{748} C_{1652} C_{1482} C_{3221} C_{3370}$ $C_{2961} C_{2227} C_{1803} C_{2265} C_{1310} C_{1930} C_{3281} C_{3342} C_{2481}$
104	$C_{743} C_{1645} C_{2684} C_{886} C_{1371} C_{2207} C_{1339} C_{2715} C_{2771} C_{1359} C_{2237} C_{1850} C_{1765} C_{3692} C_{2932}$ $C_{998} C_{2678} C_{871} C_{749} C_{1660} C_{3530} C_{3477} C_{3498} C_{3985} C_{2231} C_{1835} C_{2777} C_{1374} C_{3700} C_{2918}$ $C_{997} C_{1772} C_{1819} C_{2267} C_{1311} C_{2235} C_{3025} C_{1486} C_{3253} C_{3882} C_{3978} C_{3537} C_{3470} C_{3505} C_{2485}$ $C_{1962} C_{3793} C_{3406}$
105	$C_{750} C_{1653} C_{2670} C_{885} C_{1867} C_{2269} C_{1342} C_{2739} C_{2963} C_{1483} C_{3229} C_{3386} C_{1893} C_{2796} C_{1908}$ $C_{2790} C_{1994} C_{3285} C_{3374} C_{2993} C_{2483} C_{1931} C_{3289} C_{3358}$
106	$C_{751} C_{1661} C_{2686} C_{887} C_{1883} C_{2271} C_{1343} C_{2747} C_{3027} C_{1487} C_{3261} C_{3898} C_{2021} C_{3820} C_{3956}$ $C_{3046} C_{4042} C_{3541} C_{3502} C_{4017} C_{2487} C_{1963} C_{3801} C_{3422}$
107	$C_{756} C_{1638} C_{2661} C_{876} C_{3339} C_{2457} C_{1434} C_{3219} C_{2709} C_{1386} C_{2289} C_{1806}$
108	$C_{757} C_{1646} C_{2677} C_{878} C_{3355} C_{2459} C_{1435} C_{3227} C_{2773} C_{1390} C_{2741} C_{1898} C_{2789} C_{1900} C_{2804}$ $C_{1894} C_{3482} C_{3475} C_{3467} C_{3481} C_{2293} C_{1838} C_{2801} C_{1870}$
109	$C_{758} C_{1639} C_{2669} C_{892} C_{3403} C_{2461} C_{1466} C_{3731} C_{2711} C_{1387} C_{2717} C_{1402} C_{3685} C_{2924} C_{1012}$ $C_{1766} C_{2663} C_{877} C_{764} C_{1654} C_{1498} C_{3223} C_{3371} C_{2969} C_{2291} C_{1807} C_{2297} C_{1822} C_{1780} C_{3686}$ $C_{2917} C_{1004} C_{3851} C_{2521} C_{1438} C_{3251} C_{2965} C_{1514} C_{3733} C_{3434} C_{1946} C_{3283} C_{3343} C_{2489} C_{2545}$ $C_{1934} C_{3313} C_{3854}$
110	$C_{759} C_{1647} C_{2685} C_{894} C_{3419} C_{2463} C_{1467} C_{3739} C_{2775} C_{1391} C_{2749} C_{1914} C_{3813} C_{3948} C_{3060}$ $C_{2022} C_{2679} C_{879} C_{765} C_{1662} C_{3546} C_{3479} C_{3499} C_{3993} C_{2295} C_{1839} C_{2809} C_{1886} C_{3828} C_{3942}$ $C_{3045} C_{2028} C_{3867} C_{2523} C_{1439} C_{3259} C_{3029} C_{1518} C_{3765} C_{3946} C_{3994} C_{3539} C_{3471} C_{3513} C_{2549}$ $C_{1966} C_{3825} C_{3918}$
111	$C_{766} C_{1655} C_{2671} C_{893} C_{3915} C_{2525} C_{1470} C_{3763} C_{2967} C_{1515} C_{3741} C_{3450} C_{3941} C_{3052} C_{2036}$ $C_{3814} C_{2010} C_{3287} C_{3375} C_{3001} C_{2547} C_{1935} C_{3321} C_{3870}$
112	$C_{767} C_{1663} C_{2687} C_{895} C_{3931} C_{2527} C_{1471} C_{3771} C_{3031} C_{1519} C_{3773} C_{3962} C_{4069} C_{4076} C_{4084}$ $C_{4070} C_{4058} C_{3543} C_{3503} C_{4025} C_{2551} C_{1967} C_{3833} C_{3934}$
113	$C_{923} C_{1243} C_{3103} C_{2363} C_{3669} C_{2414} C_{949} C_{1770} C_{974} C_{1269} C_{3630} C_{2929} C_{3529} C_{3484} C_{3506}$ $C_{3971} C_{2780} C_{1398} C_{2727} C_{1897} C_{1863} C_{2285} C_{1852} C_{2802}$
114	$C_{924} C_{1266} C_{3591} C_{2409}$
115	$C_{925} C_{1274} C_{3607} C_{2411} C_{988} C_{1270} C_{3623} C_{2921} C_{2537} C_{1948} C_{3314} C_{3847} C_{2425} C_{926} C_{1267}$ $C_{3599} C_{2972} C_{1522} C_{3719} C_{3433} C_{2413} C_{956} C_{1778} C_{3655}$
116	$C_{927} C_{1275} C_{3615} C_{2427} C_{3671} C_{2415} C_{957} C_{1786} C_{990} C_{1271} C_{3631} C_{2937} C_{3561} C_{3996} C_{3570}$ $C_{3975} C_{3036} C_{1526} C_{3751} C_{3945} C_{3911} C_{2541} C_{1980} C_{3826}$

117	$C_{946} C_{1731} C_{3149} C_{2364} C_{2702} C_{1393} C_{2851} C_{969} C_{1244} C_{3126} C_{2282} C_{1813}$
118	$C_{947} C_{1739} C_{3165} C_{2366} C_{2766} C_{1397} C_{2734} C_{1905} C_{3277} C_{3388} C_{2994} C_{1987} C_{2867} C_{971} C_{1245}$ $C_{3134} C_{3292} C_{3382} C_{2979} C_{1993} C_{2286} C_{1845} C_{2794} C_{1877}$
119	$C_{950} C_{1763} C_{3661} C_{2428} C_{3151} C_{2365} C_{954} C_{1747} C_{2718} C_{1395} C_{2703} C_{1401} C_{3181} C_{2876} C_{1010}$ $C_{1735} C_{2915} C_{973} C_{1276} C_{3638} C_{1500} C_{3254} C_{3875} C_{3017} C_{2283} C_{1821} C_{2298} C_{1815} C_{1756} C_{3190}$ $C_{2855} C_{1001} C_{2859} C_{985} C_{1246} C_{3127} C_{3861} C_{2538} C_{1941} C_{3306} C_{1970} C_{3779} C_{3405} C_{2492} C_{1521}$ $C_{3726} C_{3441} C_{2958}$
120	$C_{951} C_{1771} C_{3677} C_{2430} C_{3167} C_{2367} C_{955} C_{1755} C_{2782} C_{1399} C_{2735} C_{1913} C_{3309} C_{3900} C_{3058}$ $C_{1991} C_{2931} C_{975} C_{1277} C_{3646} C_{3548} C_{3510} C_{4003} C_{4041} C_{2287} C_{1853} C_{2810} C_{1879} C_{3804} C_{3446}$ $C_{2983} C_{2025} C_{2875} C_{987} C_{1247} C_{3135} C_{3925} C_{2542} C_{1973} C_{3818} C_{4018} C_{4035} C_{3533} C_{3516} C_{1525}$ $C_{3758} C_{3953} C_{3022}$
121	$C_{958} C_{1779} C_{3663} C_{2429} C_{2974} C_{1523} C_{3727} C_{3449} C_{3437} C_{3004} C_{2034} C_{3783} C_{2923} C_{989} C_{1278}$ $C_{3639} C_{2012} C_{3318} C_{3879} C_{3049} C_{2539} C_{1949} C_{3322} C_{3863}$
122	$C_{959} C_{1787} C_{3679} C_{2431} C_{3038} C_{1527} C_{3759} C_{3961} C_{3565} C_{4028} C_{4082} C_{4039} C_{2939} C_{991} C_{1279}$ $C_{3647} C_{4060} C_{3574} C_{4007} C_{4073} C_{2543} C_{1981} C_{3834} C_{3927}$
123	$C_{995} C_{1741} C_{3196} C_{2870} C_{1373} C_{2238} C_{1843} C_{2763} C_{3278} C_{3381} C_{2986} C_{2001}$
124	$C_{999} C_{1773} C_{3708} C_{2934} C_{1375} C_{2239} C_{1851} C_{2779} C_{3534} C_{3509} C_{4010} C_{4049}$
125	$C_{1002} C_{1749} C_{3182} C_{2869} C_{1869} C_{2300} C_{1846} C_{2787} C_{2955} C_{1497} C_{3230} C_{3379} C_{2862} C_{1009} C_{1742}$ $C_{3189} C_{2490} C_{1939} C_{3275} C_{3357} C_{1404} C_{2742} C_{1891} C_{2765}$
126	$C_{1003} C_{1757} C_{3198} C_{2871} C_{1885} C_{2302} C_{1847} C_{2795} C_{3019} C_{1501} C_{3262} C_{3891} C_{1997} C_{3324} C_{3894}$ $C_{3043} C_{2878} C_{1011} C_{1743} C_{3197} C_{3790} C_{3445} C_{2990} C_{2033} C_{2494} C_{1971} C_{3787} C_{3421} C_{3452} C_{2998}$ $C_{2019} C_{3789} C_{1907} C_{2767} C_{1405} C_{2750} C_{2003} C_{3279} C_{3389} C_{3002} C_{3050} C_{2005} C_{3310} C_{3893} C_{3383}$ $C_{2987} C_{2009} C_{3294}$
127	$C_{1005} C_{1788} C_{3702} C_{2919} C_{1823} C_{2299} C_{3033} C_{1502} C_{3255} C_{3883} C_{2940} C_{1014} C_{1767} C_{3693} C_{3982}$ $C_{3569} C_{2493} C_{1978} C_{3795} C_{3407} C_{1403} C_{2719} C_{3562} C_{3989}$
128	$C_{1006} C_{1781} C_{3694} C_{2933} C_{1871} C_{2301} C_{1854} C_{2803} C_{2971} C_{1499} C_{3231} C_{3387} C_{1901} C_{2812} C_{1910}$ $C_{2791} C_{2926} C_{1013} C_{1774} C_{3701} C_{1998} C_{3317} C_{3886} C_{3057} C_{2491} C_{1947} C_{3291} C_{3359} C_{1916} C_{2806}$ $C_{1895} C_{2797} C_{1899} C_{2781} C_{1406} C_{2743} C_{3987} C_{3531} C_{3485} C_{3514} C_{2026} C_{3797} C_{3438} C_{2997} C_{3507}$ $C_{3979} C_{3545} C_{3486}$
129	$C_{1007} C_{1789} C_{3710} C_{2935} C_{1887} C_{2303} C_{1855} C_{2811} C_{3035} C_{1503} C_{3263} C_{3899} C_{2029} C_{3836} C_{3958}$ $C_{3047} C_{2942} C_{1015} C_{1775} C_{3709} C_{4046} C_{3573} C_{4014} C_{4081} C_{2495} C_{1979} C_{3803} C_{3423} C_{3964} C_{3062}$ $C_{2023} C_{3821} C_{1915} C_{2783} C_{1407} C_{2751} C_{4051} C_{3535} C_{3517} C_{4026} C_{4074} C_{4053} C_{3566} C_{4021} C_{3511}$ $C_{4011} C_{4057} C_{3550}$
130	$C_{1017} C_{1758} C_{3191} C_{2863} C_{3869} C_{2554} C_{1943} C_{3307} C_{3021} C_{1532} C_{3766} C_{3939} C_{2877} C_{1018} C_{1751}$ $C_{3183} C_{3742} C_{3443} C_{2959} C_{1529} C_{2556} C_{1974} C_{3811} C_{3917}$
131	$C_{1019} C_{1759} C_{3199} C_{2879} C_{3933} C_{2558} C_{1975} C_{3819} C_{3023} C_{1533} C_{3774} C_{3955} C_{4045} C_{3580} C_{4022}$ $C_{4067} C_{3806} C_{3447} C_{2991} C_{2041} C_{2007} C_{3311} C_{3901} C_{3066}$

132	$C_{1020} C_{1782} C_{3687} C_{2925} C_{3855} C_{2553} C_{1950} C_{3315} C_{2973} C_{1530} C_{3735} C_{3435}$
133	$C_{1021} C_{1790} C_{3703} C_{2927} C_{3871} C_{2555} C_{1951} C_{3323} C_{3037} C_{1534} C_{3767} C_{3947} C_{3053} C_{2044} C_{3830}$ $C_{3943} C_{2941} C_{1022} C_{1783} C_{3695} C_{3998} C_{3571} C_{3983} C_{3577} C_{2557} C_{1982} C_{3827} C_{3919} C_{3068} C_{2038}$ $C_{3815} C_{3949} C_{3451} C_{2975} C_{1531} C_{3743} C_{3799} C_{3439} C_{3005} C_{2042} C_{3578} C_{3991} C_{3563} C_{3997} C_{3319}$ $C_{3887} C_{3065} C_{2014}$
134	$C_{1023} C_{1791} C_{3711} C_{2943} C_{3935} C_{2559} C_{1983} C_{3835} C_{3039} C_{1535} C_{3775} C_{3963} C_{4077} C_{4092} C_{4086}$ $C_{4071} C_{4062} C_{3575} C_{4015} C_{4089} C_{4055} C_{3567} C_{4029} C_{4090}$
135	$C_{1902} C_{2805} C_{3483}$
136	$C_{1903} C_{2813} C_{1918} C_{2807} C_{3547} C_{3487} C_{3515} C_{3995} C_{2030} C_{3829} C_{3950} C_{3061}$
137	$C_{1909} C_{2798} C_{3390} C_{2995} C_{1995} C_{3293}$
138	$C_{1911} C_{2799} C_{1917} C_{2814} C_{3454} C_{2999} C_{2027} C_{3805} C_{3295} C_{3391} C_{3003} C_{2011} C_{3822} C_{3957} C_{3054}$ $C_{2037} C_{3902} C_{3059} C_{1999} C_{3325} C_{3549} C_{3518} C_{4019} C_{4043}$
139	$C_{1919} C_{2815} C_{3966} C_{3063} C_{2031} C_{3837} C_{3551} C_{3519} C_{4027} C_{4059} C_{4078} C_{4085}$
140	$C_{2013} C_{3326} C_{3895} C_{3051} C_{3453} C_{3006} C_{2035} C_{3791}$
141	$C_{2015} C_{3327} C_{3903} C_{3067} C_{3959} C_{3055} C_{2045} C_{3838} C_{4075} C_{4061} C_{3582} C_{4023} C_{3455} C_{3007} C_{2043}$ $C_{3807} C_{3070} C_{2039} C_{3823} C_{3965} C_{4030} C_{4083} C_{4047} C_{3581}$
142	$C_{2046} C_{3831} C_{3951} C_{3069} C_{3999} C_{3579}$
143	$C_{2047} C_{3839} C_{3967} C_{3071} C_{4063} C_{3583} C_{4031} C_{4091} C_{4079} C_{4093} C_{4094} C_{4087}$
144	$C_{4095}$

## C. Anexo: Ejemplos del Visualizador en funcionamiento

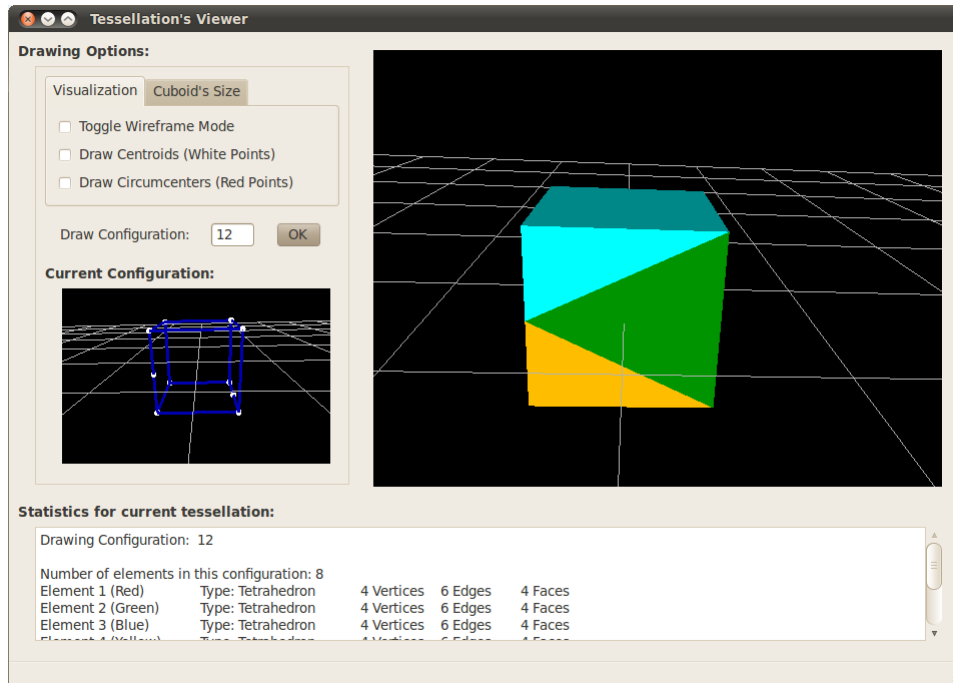


Figura 61: Visualizador con opciones por defecto

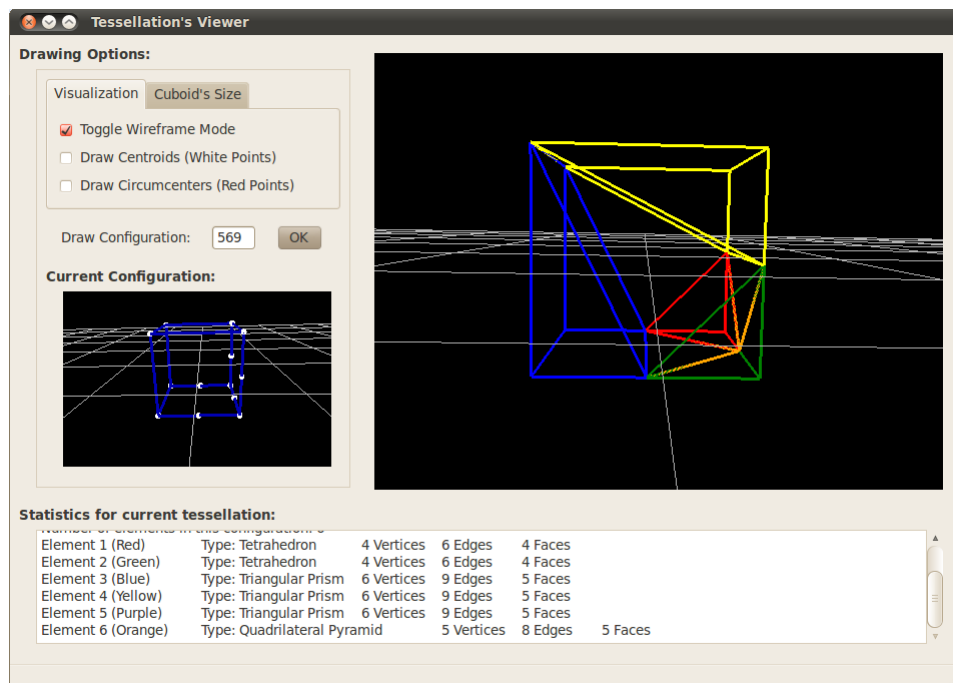


Figura 62: Visualizador en modo Wireframe

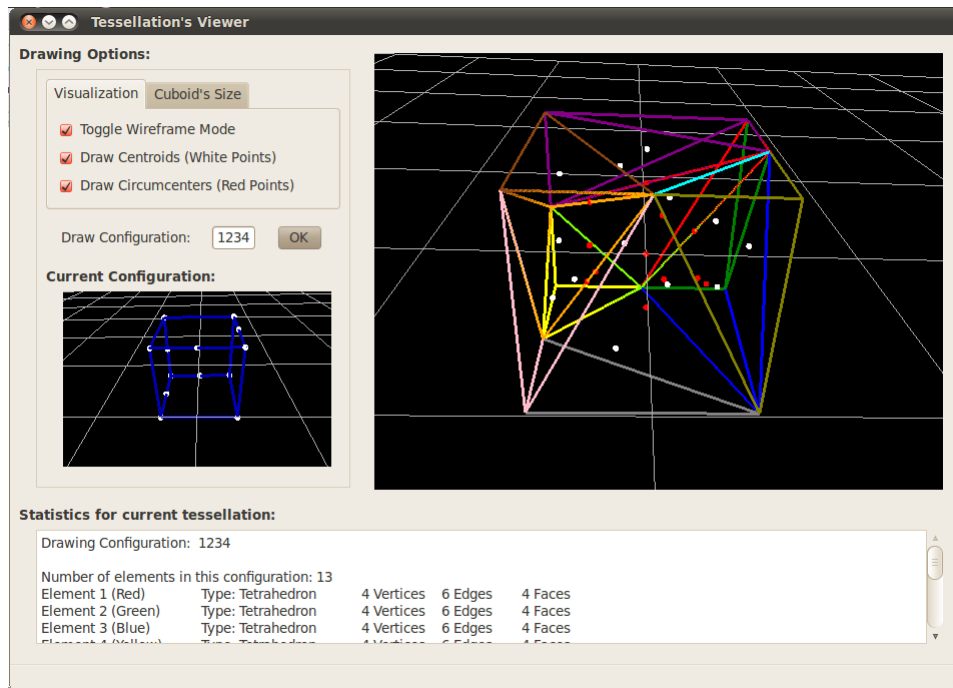


Figura 63: Visualizador en modo Wireframe, con centroides y circuncentros visibles

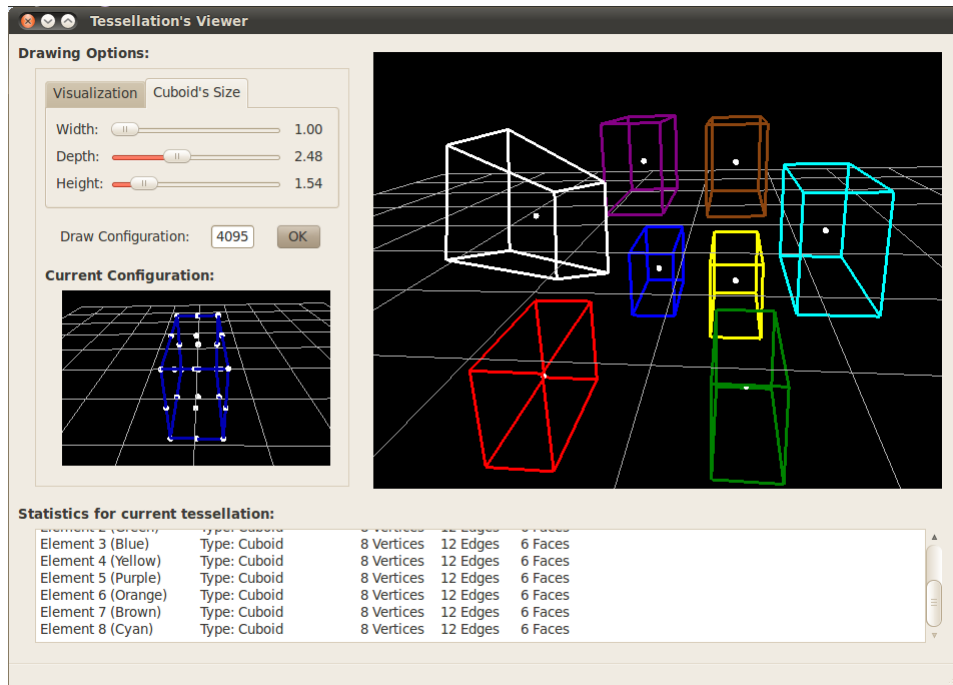


Figura 64: Visualizador en modo Wireframe, con centroides visibles, elementos desplazados y proporciones modificadas

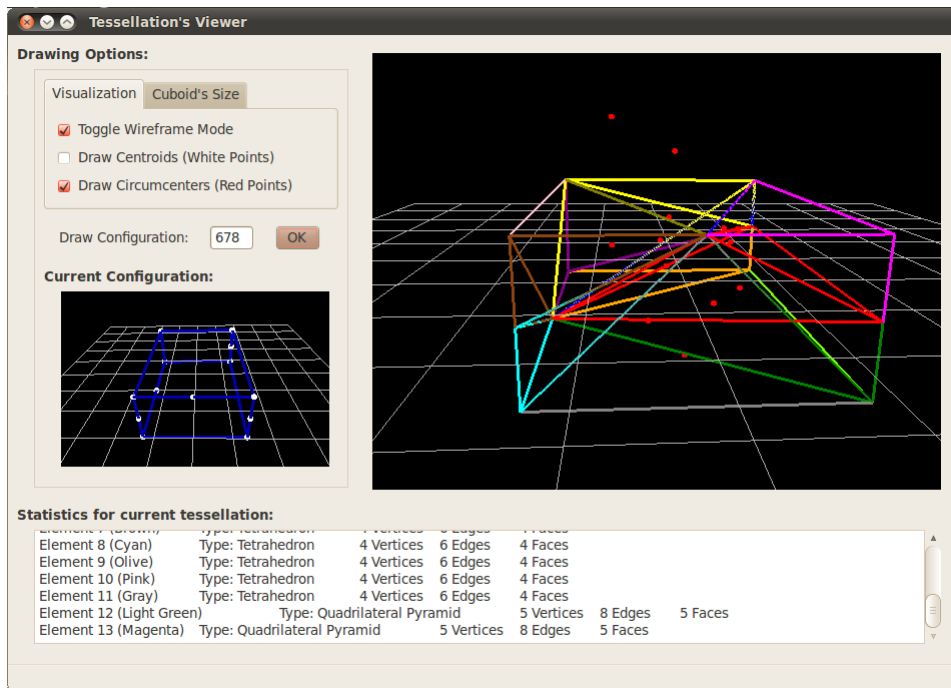


Figura 65: Visualizador en modo Wireframe, con circuncentros visibles y proporciones modificadas