



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

# POSGRADO EN CIENCIAS MATEMÁTICAS

## FACULTAD DE CIENCIAS

Un sistema experimental para  
generar mallas no-estructuradas  
en regiones planas irregulares

## TESIS

QUE PARA OBTENER EL GRADO ACADÉMICO DE  
MAESTRO EN CIENCIAS

## PRESENTA

Luis Carlos Velázquez Guerrero

DIRECTOR DE TESIS: Dr. Pablo Barrera Sánchez

MÉXICO, D.F.

Junio, 2009



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Un sistema experimental para generar mallas  
no-estructuradas en regiones planas irregulares

Luis Carlos Velázquez Guerrero

8 de junio de 2009

# Agradecimientos

Aun que estas líneas serán las primeras que un lector puede encontrar, son en realidad las últimas que escribí. Una vez que he concluido con este trabajo quiero agradecer a las personas que lo han hecho posible.

- A mi familia Velázquez Guerrero y a mis amigos. **Si te sientes identificado con mi esfuerzo es por que eres parte de él, lo sabes tú y lo sé yo.**
- A Pablo Barrera por orientar mis estudios y particularmente por la dirección de este trabajo.
- A Guilmer González por su incomparable apoyo.
- A mis sinodales, José Gerardo Tinico Ruiz, Jesús López Estrada y David Romero Vargas por sus valiosas sugerencias y comentarios.
- A Olimpia, Wattie y Violeta por la revisión de este trabajo.
- A Francisco Domínguez por el interés en invertido en mi trabajo.
- A mis compañeros del Laboratorio de Cómputo Científico, por su amistad y por las aportaciones útiles para sacar adelante este trabajo.
- A mis compas del Comité Estudiantil Metropolitano por dejarme formar parte de su lucha a lo largo de 10 años.
- A mis compas de la Smaliyel Ciencias por su buen café y su valiosa amistad.
- A mis compas de Casa Margarita por conformar un colectivo que trasciende la vivienda.

También quiero agradecer a las instituciones que han colaborado con los recursos que han hecho posible que yo me dedique a la ciencia.

- A la Facultad de Ciencias de la UNAM por abrirme una vez más las puertas como alumno y como profesor.
- Al CONACYT por brindarme la beca para el estudio de posgrado.
- A la Fac. de Ingeniería & PEMEX por permitirme participar en uno de sus proyectos.
- Al M.T.U.I.C., por apoyarme con una beca para realizar este trabajo.

# Prólogo

En la actualidad, muchos fenómenos en diversos campos de la Ciencia y la Ingeniería se pueden estudiar realizando simulaciones de los modelos matemáticos correspondientes y esto es posible gracias a la *computación científica*.

La *Geometría Computacional* es la disciplina que se encarga del estudio de los algoritmos para resolver problemas geométricos con el uso de las computadoras. Esta disciplina surge a partir de la tesis doctoral de M. I. Samos [32] en 1978, y ha madurado hasta convertirse en un área bien establecida; en ella convergen de manera natural algunas áreas de la Matemática como: Álgebra, Topología, Ecuaciones Diferenciales, Probabilidad, Análisis Numérico y Geometría con algunas áreas de las Ciencias de la Computación, como Algoritmos de Gráficas, Estructura de Datos y Optimización.

Una de las áreas de la Geometría Computacional es la Generación Numérica de Mallas, que ha despertado gran interés entre ingenieros y físicos. La problemática implica un proceso de descomposición de regiones en celdas geométricas elementales. Esta descomposición es necesaria para la discretización de ecuaciones diferenciales parciales.

El proceso de representación discreta implica una labor matemática no trivial cuyo desarrollo es motivado por las aplicaciones. Hasta el momento, la solución numérica de Ecuaciones Diferenciales Parciales (EDP's) es la aplicación cuyas exigencias cultivan el desarrollo de los procesos de generación de mallas.

Las mallas compuestas de triángulos en el caso plano o tetraedros en el caso tridimensional se utilizan en aplicaciones tales como infografía, interpolación, estudios de suelo y bases de datos. Los algoritmos descritos en este trabajo pueden ser utilizados con éxito para generar mallas para estos y otros fines.

La solución numérica de EDP's por el método de elemento finito es más simple en mallas estructuradas, por la facilidad de la determinación de los vecinos de cada nodo. Mientras que las no estructuradas requieren del almacenamiento de los índices adyacentes, la demanda de espacio de almacenamiento y memoria dinámica son mayores.

Sin embargo, la generación numérica de mallas **no estructuradas** es mucho más sencilla que la generación de mallas estructuradas sobre regiones irregulares, razón por la cual todavía hay mucha demanda de este tipo de generadores. El método desarrollado por Persson [28], para la generación de mallas en regiones de frontera suave a trozos, se ha modificado para poder usarse en regiones planas muy irregulares.

Este trabajo es motivado por el deseo del Grupo Unamalla<sup>1</sup> de satisfacer a una gama muy amplia de usuarios que prefieren trabajar con mallas triangulares, por lo que el sistema *Unamesh* desarrollado en esta tesis podría ser un módulo de una próxima versión del sistema Unamalla.

Un antecedente de este trabajo es el desarrollo del sistema TRIANG [1], el cual es parte del trabajo que Sergio Aguirre realizó durante su Maestría en Ciencias Matemáticas de la Facultad de Ciencias de la UNAM en 1995. Los métodos implementados en TRIANG buscan configuraciones óptimas de triangulaciones para generar mallas sobre regiones en el plano.

La estructura que seguirá el documento es la siguiente:

- **Capítulo 1** Se ilustra la importancia que tiene la generación numérica de mallas, también se exhibe un panorama para la generación numérica de mallas triangulares discutiendo los distintos enfoques en que ha sido abordada la problemática y algunas técnicas representativas.
- **Capítulo 2.** En la primera sección se presenta una somera descripción de la teoría de Triangulaciones de Delaunay en dos dimensiones incluyendo la descripción de algunas técnicas para su cálculo; mientras que la versión de Persson del método de vigas en equilibrio constituye el cuerpo del capítulo. Se hace una descripción detallada del método, se realiza un análisis sobre la convergencia y se muestran algunos ejemplos.
- **Capítulo 3.** Se discute la dificultad para llevar a cabo la técnica de Persson para regiones irregulares, se plantea una solución diferente que involucra algunas modificaciones al método. Se hace una descripción detallada del método modificado, se realiza un nuevo análisis sobre la convergencia y se muestran algunos ejemplos.
- **Capítulo 4.** Se presenta la estructura del sistema Unamesh. Se hace una descripción de los módulos principales, especificando la descripción de los parámetros que cada módulo plantea. También se presentan los formatos usados por el sistema para el manejo de contorno y triangulaciones
- **Apéndice A.** Se ilustra con un ejemplo el uso del Unamesh, útil para guiar a los usuarios.
- **Apéndice B.** Se describe el método de suavizamiento spline cónico para el tratamiento de un contorno.
- **Apéndice C.** Se describe el método de reducción de nodos por criterios de colinealidad para el tratamiento de un contorno.

---

<sup>1</sup>Conformado por estudiantes e investigadores de la Facultad de Ciencias, UNAM y otras dependencias de México y Cuba. <http://www.matematicas.unam.mx/unamalla/>

# Índice general

<b>Agradecimientos</b>	<b>III</b>
<b>Prefacio</b>	<b>v</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Mallas y métodos numéricos . . . . .	2
1.2. Propiedades deseables de las mallas y sus generadores . . . . .	4
1.3. Integración de un generador numérico . . . . .	7
1.4. Clasificación somera de los Métodos de generación de mallas . . . . .	8
1.4.1. Por construcción simultánea de nodos y conectores . . . . .	9
1.4.2. Generación de mallas bajo plantilla . . . . .	12
1.4.3. Nodos Primero . . . . .	17
<b>2. Método de la ecuación de equilibrio</b>	<b>23</b>
2.1. Triangulaciones de Delaunay . . . . .	25
2.1.1. Algoritmos para obtener la triangulación de Delaunay . . . . .	28
2.1.2. Triangulación de Delaunay sujeta a restricción de aristas . . . . .	38
2.2. Método de Persson para triangulaciones en equilibrio de fuerzas . . . . .	42
2.2.1. Ecuación Diferencial de disipación de partículas . . . . .	43
2.2.2. Implementación numérica de la ecuación de equilibrio . . . . .	46
2.2.3. Algoritmo . . . . .	51
2.3. Convergencia . . . . .	52
2.4. Ejemplos . . . . .	54
<b>3. Método de vigas en equilibrio para regiones irregulares</b>	<b>59</b>
3.1. Malla Inicial . . . . .	60
3.2. Tratamiento de Contorno . . . . .	61
3.2.1. Tratamiento spline cónico . . . . .	63
3.2.2. Inserción de nodos . . . . .	64
3.2.3. Reducción de nodos . . . . .	65
3.3. Nuevas especificaciones . . . . .	65
3.3.1. $F_i(\bar{x}) = 0$ . . . . .	66
3.3.2. Función de corrección $C_i$ . . . . .	66

---

3.3.3. $L^*$ Fija . . . . .	67
3.4. Algoritmo modificado . . . . .	68
3.5. Convergencia . . . . .	69
3.6. Ejemplos . . . . .	70
<b>4. Unamesh: Sistema para la generación de mallas de calidad</b>	<b>75</b>
4.1. Diseño e interfase de Usuario . . . . .	75
4.1.1. Sobre el diseño . . . . .	75
4.1.2. Interfase con el usuario . . . . .	77
4.2. Estructura modular . . . . .	78
4.3. Módulo de contornos . . . . .	79
4.3.1. Construir un contorno en pantalla. <b>New Polygon</b> . . . . .	79
4.3.2. Abrir un archivo de datos <b>Open</b> . . . . .	81
4.3.3. Guardar información <b>Save</b> . . . . .	81
4.3.4. Tratamiento por suavizamiento spline cónico . . . . .	82
4.3.5. Tratamiento por adhesión de puntos . . . . .	83
4.3.6. Tratamiento por reducción de puntos . . . . .	85
4.4. Módulo de Mallas <b>Triangulation</b> . . . . .	86
4.4.1. Llamar desde un archivo a una triangulación <b>Open</b> . . . . .	86
4.4.2. Guardar Triangulación <b>Save</b> . . . . .	87
4.4.3. Triangulación Inicial . . . . .	87
4.4.4. Generador de Malla . . . . .	89
4.5. Entrada y salida de datos . . . . .	92
4.5.1. Formato del archivo de datos de los contornos . . . . .	92
4.5.2. Formatos de archivo de Mallas . . . . .	93
<b>5. Conclusiones</b>	<b>95</b>
5.1. Conclusiones . . . . .	95
5.2. Trabajo Futuro . . . . .	96
5.3. Nota final . . . . .	96
<b>A. Guía rápida de Unamesh</b>	<b>97</b>
<b>B. Suavizamiento Spline Cónico</b>	<b>109</b>
<b>C. Reducción de puntos</b>	<b>117</b>
<b>Bibliografía</b>	<b>121</b>

# Índice de figuras

1.1. Malla estructura vs no estructurada. . . . .	3
1.2. Ejemplo una región definida por una poligonal cerrada y simple. . . . .	4
1.3. Como el ángulo mayor es cercano a $180^\circ$ la derivada en dirección vertical, que se calcula a través de la interpolación lineal se hace arbitrariamente grande. . . . .	6
1.4. Elementos no conformes en una triangulación. . . . .	7
1.5. Ejemplo del proceso recursivo de Bykat hasta obtener un triángulo. . . . .	10
1.6. Criterios de eliminación de triángulos de George Alan. . . . .	10
1.7. Proceso de avance hacia adelante por eliminación de nodos. . . . .	11
1.8. Malla en base a rejilla para un ejemplo de frontera suave. . . . .	12
1.9. Ejemplo del método de Heighway y Biddlecombe. . . . .	13
1.10. Ejemplo del método de Yerry y Shephard. . . . .	14
1.11. Interpolación transfinita. . . . .	15
1.12. Malla por Mapeo de elementos. . . . .	16
1.13. Ilustración del método de Brown. . . . .	17
1.14. Descomposición topológica y su refinamiento. . . . .	18
1.15. Ejemplo de la técnica de Lee. . . . .	19
1.16. Ejemplo de generación de elementos por una técnica de avance hacia el frente. . . . .	20
1.17. Ejemplo de la técnica de Lewis y Robinson. . . . .	20
2.1. Relación de adyacencia de nodos. . . . .	23
2.2. Relación de radios. . . . .	24
2.3. Dualidad Voronoi vs Delaunay. . . . .	25
2.4. Caso singular en una triangulación de Delaunay. . . . .	26
2.5. Circunferencias vacías en el sentido de Delaunay. . . . .	27
2.6. Ángulo Máx.-Mín. . . . .	27
2.7. Ordenamiento lexicográfico inducido. . . . .	29
2.8. Lista de adyacencia de $p_i$ y $p_k$ . . . . .	29
2.9. Tapas inferior y superior. . . . .	31
2.10. Hasta no haber caso favorable. . . . .	32
2.11. Caso favorable de tapa inferior. . . . .	32
2.12. Familia de circunferencias vacías con cuerda en común y centro en el segmento de Voronoi. . . . .	34

2.13. Trueque de Aristas 2 a 2. . . . .	35
2.14. Diagrama de convergencia a la triangulación de Delaunay. . . . .	36
2.15. Reemplazamiento de un triángulo por inserción de nodo. . . . .	37
2.16. Se corrige la conectividad para que todos los conectores de la liga sean localmente de Delaunay. . . . .	37
2.17. Triangulación de Delaunay con una restricción. . . . .	39
2.18. “Colage” cruzado en torno a una ranura $l$ . . . . .	40
2.19. Diagrama de Voronoi dual a $T_L^*(S)$ . . . . .	41
2.20. Componentes de fuerza. . . . .	44
2.21. Cambio en conexión de nodos ocasionado por perturbación. . . . .	47
2.22. Principio para la construcción de la Teselación equilátera del plano. . . . .	48
2.23. Primeras cuatro capas del método recursivo. . . . .	49
2.24. Ejemplo de malla inicial para ilustra la técnica. . . . .	50
2.25. Ejemplo 1 del círculo unitario. . . . .	54
2.26. Ejemplo 2 del círculo unitario. . . . .	55
2.27. Ejemplo de cuadrado agujerado. . . . .	55
2.28. Ejemplo de hexágono agujerado. . . . .	56
2.29. Ejemplo de intersección de regiones. . . . .	56
2.30. Triangulación inicial, “Gato”. . . . .	57
2.31. Malla óptima, “Gato”. . . . .	58
3.1. Bahía de la Habana, ejemplo de de región con geometría muy irregular. . . . .	60
3.2. “Avión”, ejemplo de contorno con poca irregularidad geométrica. . . . .	62
3.3. Información de longitudes del contorno Avión. . . . .	62
3.4. Contorno Avión con tratamiento spline cónico. . . . .	63
3.5. Contorno Avión con tratamiento uniforme a trozos. . . . .	64
3.6. Bahía de la Habana con tratamiento por colinealidad. . . . .	65
3.7. Corrección para regiones poligonales. . . . .	67
3.8. Punto de partida y evolución del generador, Avión. . . . .	70
3.9. Malla óptima de Avión con tratamiento spline. . . . .	71
3.10. Punto de partida y evolución del generador, Avión. . . . .	71
3.11. Malla de calidad, Avión. . . . .	72
3.12. Tratamiento de contorno por suavizamiento, Bahía de la Habana. . . . .	72
3.13. Punto de partida y evolución del generador, Bahía de la Habana. . . . .	73
3.14. Malla de calidad, Bahía de la Habana. . . . .	73
4.1. Ventana principal de Unamesh, con una malla inicial en pantalla. . . . .	76
4.2. Estructura del sistema y dependencia de los módulos. . . . .	78
4.3. Ventana de captura de contoro. . . . .	80
4.4. Ventana de tratamiento spline cónico. . . . .	82
4.5. Ventana de tratamiento por adhesión de puntos. . . . .	84
4.6. Ventana de tratamiento por reducción de puntos. . . . .	85
4.7. Ventana del módulo de triangulación inicial. . . . .	88

4.8. Ventana de parámetros módulo generador de malla. . . . .	89
4.9. Ventana de despliegue de datos del generador. . . . .	91
A.1. Ventana principal con el contorno <code>michoacan.con</code> . . . . .	97
A.2. Panel de Triangulación inicial. . . . .	98
A.3. Triangulación inicial, $L_0 = 0.03$ con nodos de frontera incluidos. . . . .	98
A.4. Panel principal, triangulación inicial en memoria. . . . .	99
A.5. Panel de parámetros, 1500 iteraciones. . . . .	99
A.6. Ventana de datos del generador, detenida en la iteración 100. . . . .	100
A.7. Cantidad de nodos por iteración. . . . .	101
A.8. Valor de la función optimal por iteración. . . . .	101
A.9. Medida de calidad topológica por iteración. . . . .	101
A.10. Medida de calidad geométrica por iteración. . . . .	101
A.11. Malla de Michoacán obtenida después de 1500 iteraciones, contorno sin tratamiento y método con eliminación de nodos. . . . .	102
A.12. Malla de Michoacán obtenida en la iteración 1500, sin eliminación de nodos. . . . .	102
A.13. Función optimal por iteración. . . . .	103
A.14. Calidad topológica por iteración. . . . .	103
A.15. Calidad geométrica por iteración. . . . .	103
A.16. Ventana de Tratamiento Spline Cónico, contorno de michoacán con 136 puntos. . . . .	103
A.17. Triangulación inicial, $L_0 = 0.03$ con nodos de frontera incluidos. . . . .	104
A.18. Ventana principal con la triangulación inicial. . . . .	104
A.19. Ventana de datos del generador, al finalizar exitosamente el proceso. . . . .	105
A.20. Ventana principal con malla óptima . . . . .	106
A.21. Cantidad de nodos por iteración. . . . .	106
A.22. Valor de la función optimal por iteración. . . . .	106
A.23. Medida de calidad topológica por iteración. . . . .	107
A.24. Medida de calidad geométrica por iteración. . . . .	107
B.1. Contorno: Gato. . . . .	109
B.2. Construcción del Arco Cónico. . . . .	110
B.3. Segmentos iguales. . . . .	111
B.4. Segmentos desiguales. . . . .	112
B.5. Tipo de Arco según el ángulo. . . . .	112
B.6. Continuidad $C^1$ . . . . .	113
B.7. Contornos suavizados. . . . .	114
B.8. Reparametrización por longitud de arco. . . . .	115
C.1. Criterio de longitud. . . . .	118
C.2. Criterio de error máximo. . . . .	118
C.3. Valor útil en la práctica. . . . .	119
C.4. Contorno: Presa en Valle de Bravo. . . . .	120
C.5. Contorno: Edo. México. . . . .	120



# Capítulo 1

## Introducción

La Geometría Computacional es una parte de la Geometría que se encarga del estudio de los algoritmos para resolver problemas geométricos mediante el uso de las computadoras. Es una disciplina que, se puede decir, nace con la tesis de M. I. Samos [32], en 1978, y que desde entonces ha madurado hasta convertirse en un área bien establecida.

El motivo del crecimiento de esta disciplina, se debe a que diferentes áreas del conocimiento necesitan plantear modelos computacionales en sus problemáticas locales, cada vez que estos modelos sean inherentemente geométricos es necesario abordarlos desde un enfoque desprendido de la aplicación. La necesidad de tratar tales problemas, mediante el estudio sintetizado y unificado, ha sido la razón de ser de la Geometría Computacional. Para ello, ha tenido que crearse una disciplina donde convergen de manera natural algunas áreas de la Matemática, como la Topología, Combinatoria, Álgebra, Probabilidad y, por supuesto, Geometría, con aquellas de las Ciencias de la Computación, como Algoritmos de Gráficas, Estructura de Datos y Optimización.

Es así que la Geometría Computacional se ha avocado a problemas tales como partición de polígonos, envolturas convexas, diagramas de Voronoi, disposición de rectas, búsqueda geométrica y planeación de movimiento, entre otros.

El problema de interés para este trabajo es la generación de triangulaciones sobre regiones irregulares, este problema han sido demandado por la Ingeniería y la Física en solución de ecuaciones diferenciales parciales y para la interpolación spline.

El método que se usará en este trabajo, para la construcción de triangulaciones, está inspirado en el método de vigas en equilibrio propuesto por Persson en su texto *A simple mesh generator in Matlab* [28].

En este capítulo consideramos el problema de la generación de mallas triangulares en regiones poligonales de geometría no simple, y haremos una reseña de como ha sido abordado el problema por diferentes técnicas.

## 1.1. Mallas y métodos numéricos

Muchos fenómenos físicos en la ciencia y la ingeniería pueden ser modelados por ecuaciones diferenciales parciales (EDP's), como la deformación mecánica, la transferencia de calor, dinámica de fluidos, la propagación de ondas electromagnéticas, y la mecánica cuántica. Cuando estas ecuaciones tienen condiciones de frontera complicadas o se plantean sobre regiones de forma irregular, suele no haber una solución analítica y una aproximación numérica de la solución resulta necesaria.

Los métodos de solución numérica de ecuaciones diferenciales parciales transforman la EDP (lineal o no lineal) en un sistema finito de ecuaciones acopladas, correspondientemente lineales o no lineales.

Estos métodos numéricos son clasificados según sea el dominio físico de definición; *el método de elemento finito* (FEM) fue diseñado para dominios planos, mientras que *el método de volumen finito* (FVM, también conocido como *el método de control de volumen*) existe para dominios tridimensionales, y por último *el método de elementos de frontera* (BEM) que es útil para resolver numéricamente ecuaciones diferenciales planteadas sobre la frontera de definición de un dominio físico.

Para poder transformar la EDP en un sistema de ecuaciones es necesario discretizar el dominio físico  $\Omega$ , es decir, es necesario descomponer la región  $\Omega$  como una unión de regiones elementales que en general llamamos celdas; normalmente las celdas son triangulares o cuadrangulares (en caso de un dominio bidimensional), o tetraédricas o hexahédricas (para tres dimensiones). A la salida de este proceso de discretización es a lo que llamamos Malla.

Una malla que sea la discretización de un dominio  $\Omega$  debe ser definida de forma simple, si la malla es una triangulación y se cuenta con las coordenadas de los tres vértices de cada celda, entonces la malla puede ser bien representada. Así, resulta satisfactorio contar con la lista de vértices de la triangulación que define la *geometría de la malla*, y contar también con una estructura que indique como se relacionan los vértices en la triangulación que define la *topología de la malla*.

Como los elementos tienen forma simple, es posible aproximar el comportamiento de una EDP en cada nodo por medio de la relación de adyacencia que guardan, desde este enfoque el problema podría verse como la elección adecuada de vértices y aristas de definición de la malla; este proceso se le llama generación de una malla triangular. Los triángulos serán llamados elementos y los vértices serán nodos en el método de elemento finito.

Si  $\Omega$  es una región plana y  $\{\Delta_i\}$  es una triangulación de  $\Omega$ , entonces usualmente:

$$\Omega_h = \bigcup \Delta_i \subset \Omega \quad (1.1)$$

y lo que nos gustaría es que

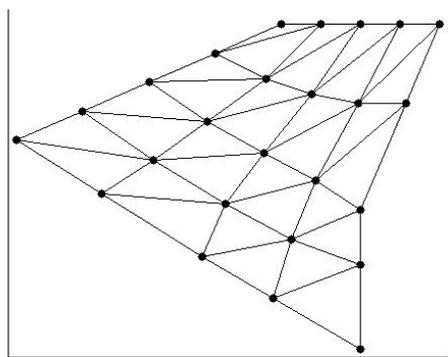
$$\Omega_h = \Omega \quad (1.2)$$

es decir

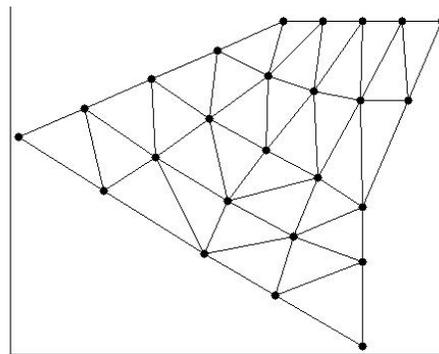
$$\partial \left( \bigcup \Delta_i \right) = \partial \Omega \quad (1.3)$$

en este caso decimos que la malla conforma la región; esto es muy fácil de lograr en regiones poligonales, pero más difícil si la frontera es muy irregular y si contiene “arcos curvos”.

Las mallas pueden clasificarse como estructuradas o no estructuradas, la figura 1.1 ilustra un ejemplo de cada uno. Las mallas estructuradas topológicamente presentan una estructura en la adyacencia que es uniforme mientras que las no estructuradas carecen de tal. En la malla estructurada, los índices de los vecinos a cualquier nodo son fácilmente identificables por que la adyacencia resulta natural, mientras que una malla no estructurada requiere del almacenamiento en una lista de los vecinos de cada nodo.



(a) Malla Estructurada, generada por un mapeo de triángulos del cuadrado unitario a la región



(b) Malla No Estructurada, generada usualmente por algún criterio de ángulo mínimo sobre el conjunto de nodos de la malla estructurada

Figura 1.1: Malla estructura vs no estructurada.

Tanto la generación de mallas estructuradas como las no estructuradas presentan dificultades a resolver, y cada una de ellas abre un campo de estudio interesante. Este documento considera únicamente la tarea de generación de mallas no estructuradas, y además considera sólo mallas formadas por triángulos.

## 1.2. Propiedades deseables de las mallas y sus generadores

El problema de discretización de un dominio para la simulación numérica puede ser un problema más difícil de lo que parece a simple vista. Una malla debe ajustarse a la región dominio a modelar, y también, debe cumplir condiciones sobre el tamaño y la forma de sus elementos, a veces una malla útil debe satisfacer condiciones que de primera intención parecen contradictorias.

El primer objetivo del modelado de la región es mantener o conservar lo mejor posible su forma. Los científicos e ingenieros a menudo necesitan construir modelos sobre regiones con formas complicadas.

A menudo una curva de frontera debe ser aproximada por una frontera lineal a trozos, por lo que los algoritmos de generación de mallas suelen suponer que de entrada la geometría es poligonal. Esta hipótesis será mantenida a lo largo de este documento, aunque aquí se presentará una estrategia para simplificar contornos irregulares y aproximarlos por otros con fronteras más suaves pero conservando la forma.

**Definición 1.** Una región  $\Omega$  es poligonal simplemente conexa si es simplemente conexa y su frontera es un polígono  $\partial\Omega = S_\Omega = \{p_1, p_2, \dots, p_m\}$ .

Trabajaremos con regiones  $\Omega$  poligonales orientadas por lo que supondremos que  $\partial\Omega = S_\Omega$  es un polígono orientado positivamente, vea figura 1.2.

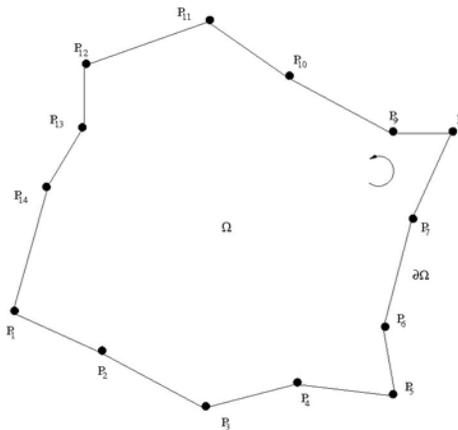


Figura 1.2: Ejemplo una región definida por una poligonal cerrada y simple.

En síntesis el primer objetivo es aproximar la región física  $\tilde{\Omega}$  por una región poligonal  $\Omega$  simplemente conexa.

Un segundo objetivo de generación de mallas es ofrecer el máximo control posible sobre los tamaños de los elementos de la malla. Idealmente, este control incluye la capacidad de pasar de elementos pequeños a grandes a través de una distancia relativamente corta. La razón de este requisito es que el tamaño del elemento tiene dos efectos en una simulación de elemento finito. Los elementos pequeños, que conforman una región densa, ofrecen más precisión que los grandes cuyo espaciamiento es mayor, como lo es el tiempo de cálculo para resolver un problema numérico, ya que es proporcional al número de elementos, y es recomendable plantear la posibilidad de usar distintas distribuciones. Por lo tanto, elegir el tamaño supone un elemento de negociación entre la velocidad de procesamiento y la precisión la aproximación.

Además, el tamaño del elemento necesario para alcanzar una determinada cantidad de precisión depende del comportamiento de los fenómenos físicos en que se basa el modelo, y puede variar a lo largo del dominio. Por ejemplo, una simulación de flujo en tres dimensiones requiere elementos más pequeños en medio de turbulencias que en las zonas de relativa tranquilidad, el elemento ideal en una parte de la malla puede variar en volumen por un factor de un millón o más del ideal en otra parte de la malla. Si en lugar de considerar un cambio sustancial en el tamaño de los elementos en toda la malla para evitar un escollo, se debe elegir un tamaño lo suficientemente pequeño como para garantizar la suficiente precisión en las zonas de mayor exigencia e incurrir posiblemente en la excesiva cantidad de cálculos, un generador de malla debe ofrecer una rápida progresión de pequeño tamaño.

Dada una malla gruesa, con un número relativamente reducido de elementos, no es difícil usarla para producir otra con un número mayor de elementos pequeños en algunas zonas deseadas. El proceso inverso no es tan fácil, por lo tanto, los algoritmos de generación de mallas con frecuencia se fijan el objetivo de poder generar mallas con pocos elementos, y ofrecer la opción de refinar parte de la malla en donde los elementos no sean lo suficientemente pequeños como para obtener la precisión requerida.

Un tercer objetivo de la generación de mallas, y la verdadera dificultad, es que los elementos deberían ser relativamente de buena forma, ya que los elementos con ángulos interiores grandes o pequeños pueden degradar la calidad de la solución numérica, así que se busca que el rango de los ángulos sea corto.

En los métodos numéricos, los elementos con ángulos grandes pueden causar un gran *error de discretización*, es decir, la precisión de la solución dada sobre la malla puede ser mucho menor que si los tres ángulos de cada elemento son parecidos. En principio, la solución calculada por el discreto se aproxima a la solución exacta conforme el tamaño del elemento más grande se acerca a cero. Sin embargo, Babuška y Aziz [2] muestran que en mallas densas la convergencia a la solución exacta puede no ocurrir si los ángulos de la malla son muy abiertos.

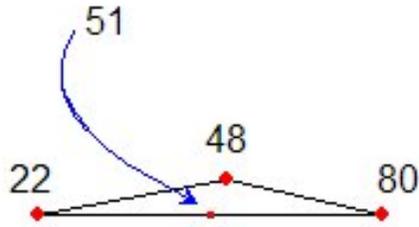


Figura 1.3: Como el ángulo mayor es cercano a  $180^\circ$  la derivada en dirección vertical, que se calcula a través de la interpolación lineal se hace arbitrariamente grande.

Otro problema causado por los grandes ángulos es de errores en las derivadas de la solución, que surgen como efecto de interpolación sobre la malla.

La figura 1.3 muestra un elemento de la malla con los valores asociados a cada nodo que representan una aproximación de la cantidad física de la solución. Si se usa una interpolación lineal para estimar la solución en los puntos no nodales, el valor interpolado en el punto medio del borde inferior es de 51. Este valor interpolado sólo depende de los valores relacionados con los dos nodos de abajo, y es independiente del valor asociado al tercer nodo. Como el ángulo en el tercer elemento es grande, el punto con valor de 51 es arbitrariamente cercano al nodo superior cuyo valor es 48.

Por lo tanto, la derivada parcial de la estimación de la solución en la dirección vertical podrá ser arbitrariamente grande, y el cambio resulta por demás espaciado aun cuando los valores nodales puedan ser exactos. Este efecto se produce porque un valor interpolado linealmente no es suficiente si la solución exacta no es lineal. Este problema puede afectar cualquier aplicación que utilice mallas de interpolación, y no sólo al resolver una EDP. Por ejemplo, en las simulaciones de deformación mecánica, en la que las derivadas de una solución son de interés y no tanto la solución en sí misma, el uso de un interpolante spline de mayor grado en puede mejorar los cambios deseados.

Los triángulos con ángulos pequeños también causan problemas pues el sistema de ecuaciones del método numérico puede ser mal condicionado. Si esto ocurre, la precisión de la solución disminuye si el sistema es resuelto por métodos directos, y en con métodos iterativos la convergencia es lenta.

Muchos algoritmos de generación de mallas optan por obligarse a superar un ángulo mínimo en toda la triangulación, si se decide acotarlo inferiormente entonces el mayor ángulo de la triangulación a su vez se estará acotando superiormente. Por ejemplo, si ningún ángulo es más pequeño que  $\theta$ , entonces ningún ángulo será mayor que  $180 - 2\theta$ .

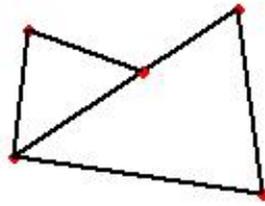


Figura 1.4: Elementos no conformes en una triangulación.

Las limitantes de tamaño del elemento y de forma del elemento son difíciles de conciliar porque los elementos deben compartir la totalidad de sus bordes. En la figura 1.4 se ilustra un ejemplo de adyacencia entre elementos que es considerado como ilegal, pues el borde de un elemento es sólo una parte de un borde de un elemento adyacente. Y aun que hay variantes del método de elemento finito que permiten tales elementos disconformes, tales elementos no son preferidos ya que pueden degradar la convergencia del método. Pudiese ser más fácil crear mallas con elementos de buena calidad disconformes pero los problemas numéricos de error persisten.

### 1.3. Integración de un generador numérico

Los algoritmos para generar una malla útil para los métodos numéricos se conforman por bloques básicos de construcción, cada bloque debe ser independiente y puede ser usado en conjunción con otras técnicas para elaborar distintas estrategias de generación de mallas. Estos bloques son conocidos por módulos.

En general, la salida de un generador de mallas no estructuradas es compuesta por dos arreglos, la Colección de Nodos, que denotaremos  $S$  y constituye la *geometría de la malla* y la Colección de Elementos, que denotaremos  $T(S)$  y constituye la *topología de la malla* pues brinda la conectividad entre nodos.

La generación de una malla puede realizarse ya sea en secuencia (primero la generación de los puntos seguido por una fase de triangulación), o simultáneamente, utilizando una técnica de promoción hacia adelante, o una técnica de triangulación Steiner, (la triangulación de Steiner se refiere a la inclusión de puntos adicionales en una triangulación con el fin de mejorar la calidad de la triangulación).

El problema de la generación de mallas incluye varias fases bien definidas que se pueden resumir como sigue:

- Definición de la frontera del dominio físico.
- Definición de la distribución de tamaño del elemento en función de la ubicación espacial (la distribución de la forma del elemento es también necesaria en el caso de mallas estiradas)

- Construcción de la frontera de la malla utilizando un enfoque de discretización adecuado.
- Construcción de la malla conforme construyendo de forma secuencial o simultánea los nodos y sus conexiones.
- Tratamiento posterior a la malla generada tendiente a mejorar la calidad del elemento.

La frontera del dominio físico normalmente se define a través de algún tipo de base de datos, y a veces implica el uso de curvas spline en 2D, la descripción de la frontera inicial puede ser discretizada como una colección de segmentos de recta. La discretización de la frontera del dominio físico se puede realizar anterior a la generación de la malla pero en ocasiones se hace de forma simultánea.

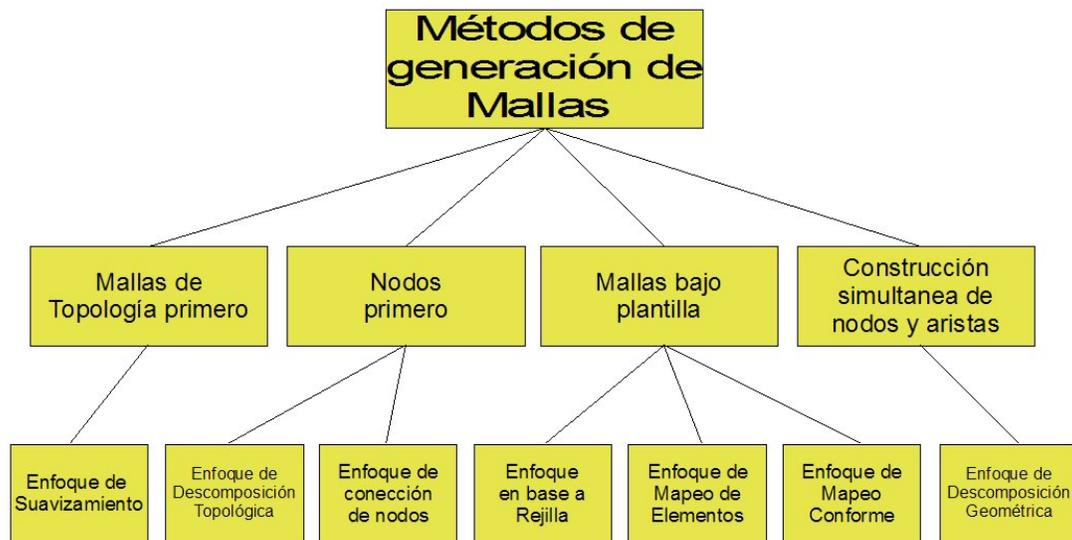
El procedimiento de construcción de malla debe ser capaz de garantizar la integridad de la frontera, en dos dimensiones corresponde a la generación de la triangulación que contengan el subconjunto de los conectores que definen la discretización de la frontera. Dos submódulos constituyen el procedimiento de construcción de la malla en el caso secuencial, uno para la generación de los nodos sujeto a la distribución de tamaño del elemento deseada, el otro para el cálculo de las conexiones donde la teoría de Delaunay, que se verá más adelante, es considerada como la alternativa que genera triangulaciones de mayor calidad.

El tratamiento posterior a la malla es una alternativa opcional aunque a veces necesaria, la necesidad se plantea en algunos casos pues a pesar de establecer una articulación adecuada de los nodos en regiones irregulares los elementos no cuentan con la calidad deseada. Así que el tratamiento posterior busca generar un suavizamiento que tienda a regularizar la distribución deseada en los nodos.

## 1.4. Clasificación somera de los Métodos de generación de mallas

La generación numérica de mallas es un área del campo de la geometría computacional relativamente reciente. Tiene su origen en la generación de mallas para simular soluciones a EDP's por lo que siendo la aplicación con mayores exigencias numéricas, el desarrollo de las teorías para la generación de mallas se ha centrado en cubrir las necesidades de calidad deseada por los métodos numéricos.

La clasificación que aquí se presenta no es exhaustiva y no busca desarrollar completamente ningún método, simplemente se busca describir las cualidades geométricas y topológicas de los distintos enfoques; esta clasificación dada por el siguiente esquema es basada en Ho-Le [19]. Los enfoques en muchos casos son aplicables en mallas 3D.



#### 1.4.1. Por construcción simultánea de nodos y conectores

Esta clase de generación se da siempre que se crean los vértices y las aristas de los triángulos simultáneamente y da lugar al enfoque de **descomposición geométrica**, el cual hace un intento por generar buenos elementos. Para ello se empieza usualmente con la frontera de la región, y determina la ubicación óptima de los nodos y el tamaño de los lados, de manera representativa podemos tomar el método recursivo propuesto por Bykat [8] y el método iterativo presentado por Alan George [14].

El método de Bykat se basa en el proceso recursivo de división de un polígono convexo en dos subregiones convexas que comparten una interfaz común, para ser aplicado a regiones no convexas es necesario hacer una subdivisión previa de la región en partes convexas. A cada subregión definida en la recursividad se le van a insertar nodos en la frontera para satisfacer la distribución de la densidad deseada. Una interfaz se define como un segmento de línea recta que une dos vértices  $p_r$ ,  $p_s$  en la frontera de la poligonal  $\Omega$  y la subdivide en dos poligonales convexas  $\Omega_1$  y  $\Omega_2$  cerca de su eje mayor. Ya que las dos subregiones  $\Omega_1$ ,  $\Omega_2$  son polígonos convexas, es plausible la aplicación recursiva de la operación básica de “bisección” del polígono resultando en la división de  $\Omega$  en un conjunto finito de subregiones convexas, definidas por sus respectivos vértices y almacenadas en un diagrama de árbol. Así es que el proceso recursivo terminará cuando todas las subregiones tengan tres vértices y no haya la necesidad de incluir nuevos nodos, pues la distribución de densidad ha sido alcanzada, es decir, la terminación natural siempre será una triangulación de  $\Omega$ .

<sup>0</sup>Algunas de las imágenes que ilustran las técnicas fueron retomadas de los textos señalados.

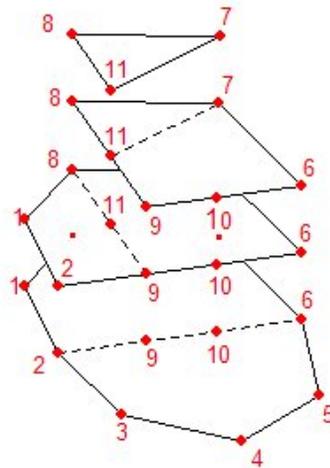


Figura 1.5: Ejemplo del proceso recursivo de Bykat hasta obtener un triángulo.

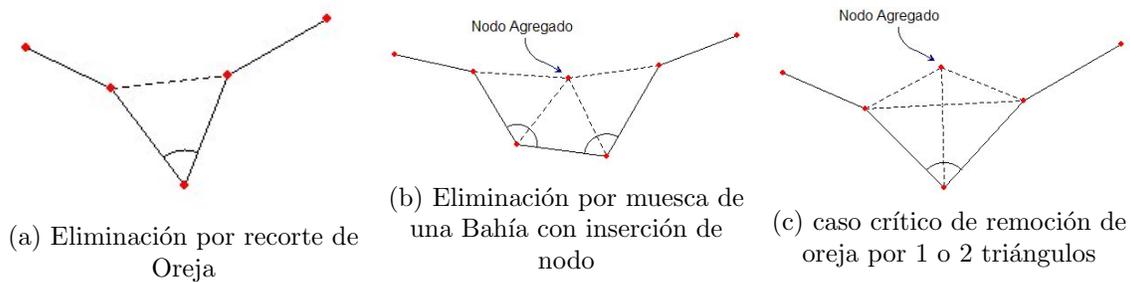


Figura 1.6: Criterios de eliminación de triángulos de George Alan.

El método iterativo de Alan George, no es sólo para regiones poligonales, pues propone una técnica para la colocación de nodos en la frontera de forma que en principio se admiten triángulos curvilíneos. Para el buen uso del método no basta con la frontera de la región de entrada, también es necesario especificar con una sencilla regla la forma en que cada arco será dividido. El método realiza la eliminación de un triángulo por cada iteración, y está catalogado como un proceso de avance hacia enfrente. Para identificar el triángulo a eliminar se hace una distinción en criterios dados por el ángulo interior a la región de cada nodo.

En una primera fase del proceso se detectan las orejas en la frontera y las recorta, en la figura 1.6(a) se muestra una oreja (con ángulo menor a 60 grados) y la forma en que ésta es recortada.

Una vez que la región no cuenta con orejas se pasa a una segunda etapa, en la cual se detectan las bahía (vértices consecutivos que tengan ángulos interiores menores a 150 grados) y se hace la eliminación de un triángulo realizando una muesca para lo cual se utiliza procedimiento de construcción de nodo, figura 1.6(b).

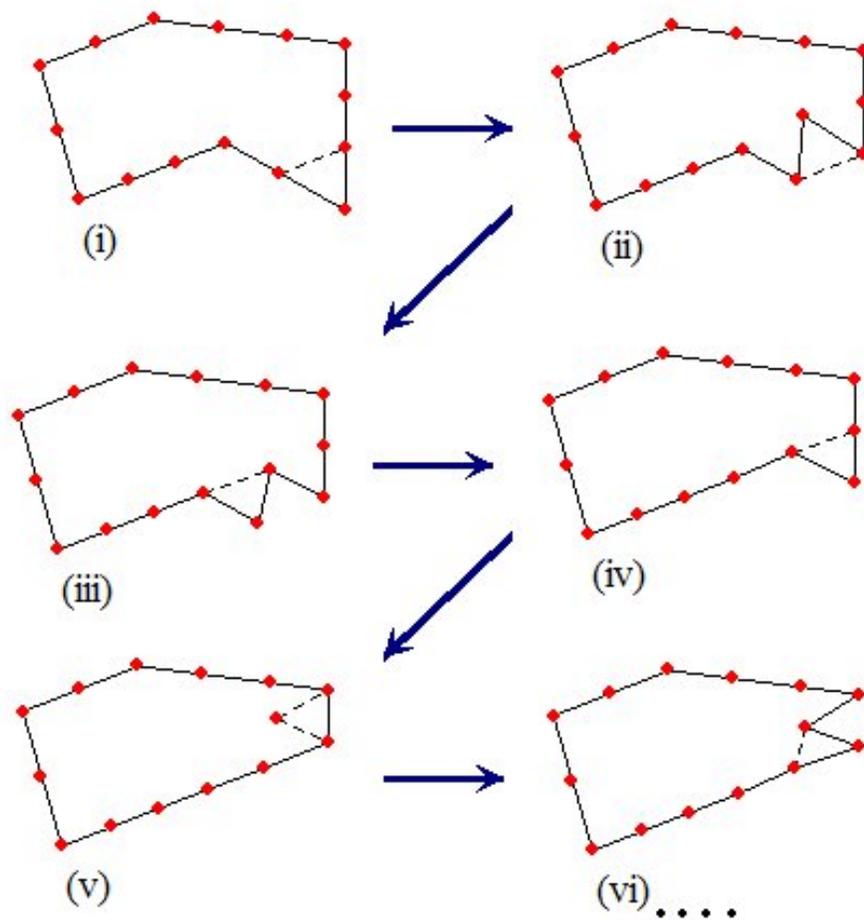


Figura 1.7: Proceso de avance hacia adelante por eliminación de nodos.

En la tercer etapa del proceso (cuando no haya caso de orejas de menos de 60 grados o bahías) se detectan aquellas orejas donde el ángulo interior está entre los 60 y los 90 grados, para este caso se plantean dos posibilidades, la primera es el recorte de la oreja como se propone en la primer etapa y la segunda es la inserción de un nodo para la remoción simultánea de dos triángulos, para decidir que tipo de eliminación se realizará, George propone un criterio que busca que la triangulación sea de la mejor calidad posible, figura 1.6(c).

Finalmente, cuando la región es casi convexa, se repite el proceso de generación de muesca para recaer en la primer etapa del proceso, en la figura 1.4.1 podemos observar algunos pasos en el procedimiento. Cuando la región finalmente sea un triángulo el proceso se da por terminado.

### 1.4.2. Generación de mallas bajo plantilla

Para construir una malla por algoritmos bajo esta clase, una triangulación plantilla es previamente generada en otro lugar, posteriormente la malla es adaptada a la región dominio. Tres enfoques dentro de este caso pueden ser identificados.

#### Enfoque basado en rejilla

El enfoque basado en rejilla surge de la observación de que una rejilla es la malla de alguna región, este enfoque puede hacerse siempre que las celdas que atraviesan la frontera de la región  $\Omega$  puedan convertirse en elementos. Entre más fina sea la rejilla, mejor será la calidad de la malla, ya que los elementos internos son de por sí de calidad y la proporción de elementos de mala calidad tiende a desaparecer. Los métodos actuales para la generación de mallas son más sofisticados (vease los capítulos 2 y 3 de este trabajo), pero los resultados son siempre similares, ya que los elementos internos no varían sustancialmente; la única diferencia se encuentra en los elementos en la frontera.

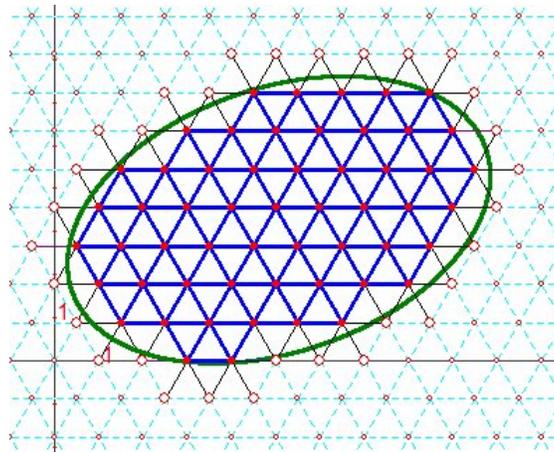


Figura 1.8: Malla en base a rejilla para un ejemplo de frontera suave.

El método propuesto por Thacker, González y Putland [35] es probablemente el primero publicado bajo este enfoque. La región física primero es encerrada en un triángulo equilátero que cuenta con una rejilla triangular. Los puntos de la rejilla que quedan fuera de  $\Omega$  se eliminan, dejando una frontera en zigzag. Los puntos de la frontera en zigzag son entonces recolocados encima de la frontera de  $\Omega$  para crear finalmente la malla.

La variante al método propuesta por Kikuchi [22] consiste en extender el método anterior para crear mallas con cuadriláteros en su mayoría, y con algunos triángulos para rellenar, usando una rejilla rectangular como plantilla.

Uno de los problemas que estos métodos plantean es que las características finas de la forma de  $\Omega$  se pueden perder debido al tamaño de la rejilla.

El método de Heighway y Biddlecombe [18] empieza con la frontera en zigzag totalmente contenida en  $\Omega$  1.9(a), a continuación se reposicionan los nodos más cercanos a la frontera 1.9(b) y se construye una malla fina entre la frontera zigzag y la frontera de  $\Omega$  con un algoritmo de triangulación. Inicialmente las rejilla es rectangular, pero después son diagonalmente quebrados en dos triángulos 1.9(c).

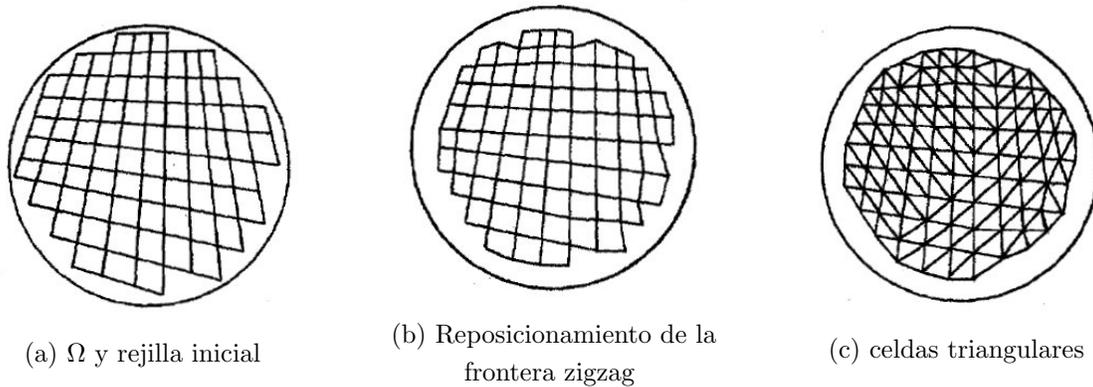


Figura 1.9: Ejemplo del método de Heighway y Biddlecombe.

De manera similar el método de Kela, Perucchio y Voelcker [21] usa una rejilla de rectángulos en el interior, pero hace un esfuerzo por rellenar la franja faltante con cuadriláteros siempre que esto sea posible, y en último caso con triángulos. Inicialmente sólo se generan mallas uniformes, para posteriormente permitir refinamiento en las zonas deseadas. Para el almacenamiento computacional de la malla se usó la *representación constructiva de geometría sólida* que ya era conocida, la CSD se creó para la solidez en las operaciones geométricas booleanas.

El método de Yerry y Shephard [36] comienza con una *codificación quadtree* de la región, vea figura 1.4.2(a). Este quadtree se ha modificado para ser adecuado a la generación de la malla de la siguiente manera:

- El interior de la región es subdividido en cuadrantes cuyo tamaño de malla debe satisfacer la distribución de densidad.
- Los cuadrantes vecinos pueden diferir en más de un nivel de subdivisión.
- Se puede hacer que los cuadrantes en la frontera crucen las puntas.

Cada cuadrante es después dividido en triángulos respetando que la malla resultante sea conforme, vea figura 1.4.2(b). Los triángulos, especialmente en la frontera pueden no ser de buena calidad.

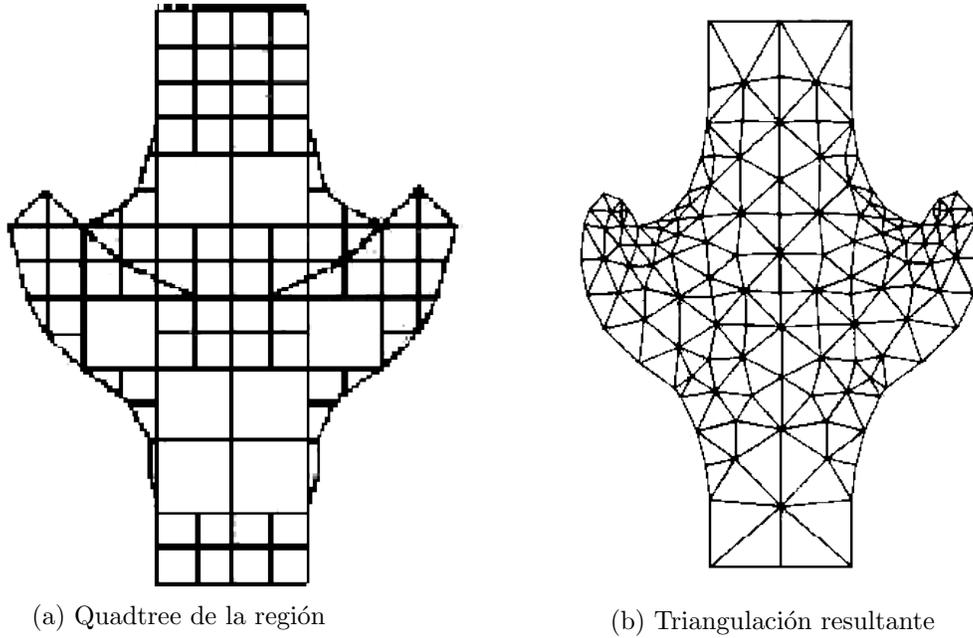


Figura 1.10: Ejemplo del método de Yerry y Shephard.

### Mapeo de elementos

El enfoque de mapeo de elementos en esta clase de generadores es el enfoque favorito para trabajar con mallas estructuradas, se busca usar como plantilla una malla lógica, ya sea triangular o rectangular, en el espacio paramétrico del triángulo equilátero o en el cuadrado unitario según sea el caso. La frontera de  $\Omega$  se subdivide manualmente en tres (o cuatro) caras, que son consideradas elementos macro; por tal motivo no es completamente automático. Posteriormente tomando como dominio la plantilla, se induce una función interpolante sobre la región.

Dada una región de cuatro caras, una malla puede ser inducida por el mapeo de nodos de alguna malla del cuadrado unitario, la metodología de Coons<sup>1</sup> [11] parte de dar las coordenadas cartesianas de un punto  $X(\xi, \eta)$  en el cuadrilátero correspondiente a un punto  $(\xi, \eta)$  de la malla lógica del espacio de parámetros. Es decir:

$$\begin{aligned}
 X(\xi, \eta) = & (1 - \eta)f_1(\xi) + \eta f_3(\xi) + (1 - \xi)f_2(\eta) + \xi f_4(\eta) \\
 & - (1 - \xi)(1 - \eta)X(0, 0) - (1 - \xi)\eta X(0, 1) \\
 & - \xi\eta X(1, 1) - \xi(1 - \eta)X(1, 0)
 \end{aligned}$$

para  $0 \leq \xi \leq 1, 0 \leq \eta \leq 1$ , para las cuatro curvas parametrizadas  $f_1(\xi)$ ,  $f_2(\eta)$ ,  $f_3(\xi)$  y  $f_4(\eta)$  adecuadamente orientadas.

<sup>1</sup>A la metodología se le puede formular en términos de la teoría proyectores como se plantea en *An operator calculus for surface and volume modeling* [15]. El concepto de proyector, permite la generalización de la técnica para el uso de interpolantes de orden superior.

La técnica empleada se le llama la *técnica de interpolación transfinita*<sup>1</sup>. Para poder inducir una distribución de densidad a la malla de  $\Omega$  se debe de usar una plantilla subdividida en un número de piezas diferente por cada región, y puede ser graduada con espaciamiento no uniforme en las líneas de la cuadrícula sobre  $\xi \times \eta$ .

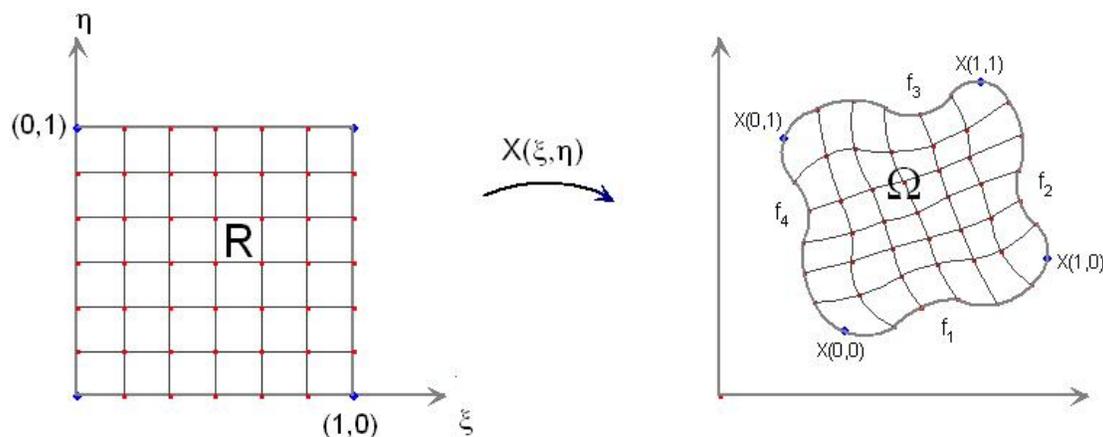


Figura 1.11: Interpolación transfinita.

El *mapeo discreto* de interpolación transfinita **TFI**, es otro caso especial. Para una región poligonal  $\Omega$  que se representa por una lista cerrada de los puntos situados en la curva,

$$\{X(\xi_1, \eta_j), X(\xi_i, \eta_{jl}), X(\xi_d, \eta_j), X(\xi_i, \eta_1) \mid i = 1, \dots, il, j = 1, \dots, jl\},$$

el mapeo se completa asignando  $X(\xi_i, \eta_j)$  para todo  $(i, j)$ , vea figura 1.12.

A menudo, los elementos producidos por un generador automático de mallas por técnicas de mapeo de elementos no satisfacen las necesidades de las aplicaciones, es más, para regiones muy irregulares el mapeo resultante suele no ser considerado como una malla por existir intersección entre los elementos. Para estos casos se acostumbra dar un tratamiento a la malla, usualmente realizado por la clase de generadores de topología primero<sup>3</sup>.

<sup>1</sup>vea *A general two-dimensional graphical finite element preprocessor utilizing discrete transfinite mappings* [16] y *Transfinite interpolation and applications to engineering problems* [17].

<sup>2</sup>El mapeo isoparamétrico, un caso especial de la interpolación transfinita se obtiene cuando las curvas de frontera son descritas por polinomios de Lagrange. La técnica descrita por Zienkiewicz y Phillips [37].

<sup>3</sup>Como las diversas técnicas de estos tratamientos no serán abordadas le sugerimos consultar *Generation of non-degenerate meshes* [20], *Adaptive discrete harmonic grid generation* [4], *Generating Quality Structured Convex Grid On Irregular Regions* [5] y *Robust Discrete Grid Generation on Plane Irregular Regions* [6].

Si la curva de frontera de  $\Omega$  es dividida en tres se puede generar una malla de elementos triangulares mediante el uso de una combinación trilineal descrita por el interpolante de Barnhill, Birkhoff y Gordon [3].

La formulación más simple de la técnica es

$$P(v, \nu, \omega) = \frac{1}{2} \left[ \frac{vg(\nu)}{1-\nu} + \frac{\omega h(1-\nu)}{1-\nu} \frac{\nu h(\omega)}{1-\omega} + \frac{vf(1-\omega)}{1-\omega} + \frac{\omega f(v)}{1-\nu} + \frac{\nu g(1-\nu)}{1-\nu} - \omega f(0) - \nu g(0) - \nu h(0) \right]$$

para  $v + \nu + \omega = 1$ , y  $0 \leq v \leq 1, 0 \leq \nu \leq 1, 0 \leq \omega \leq 1$ .

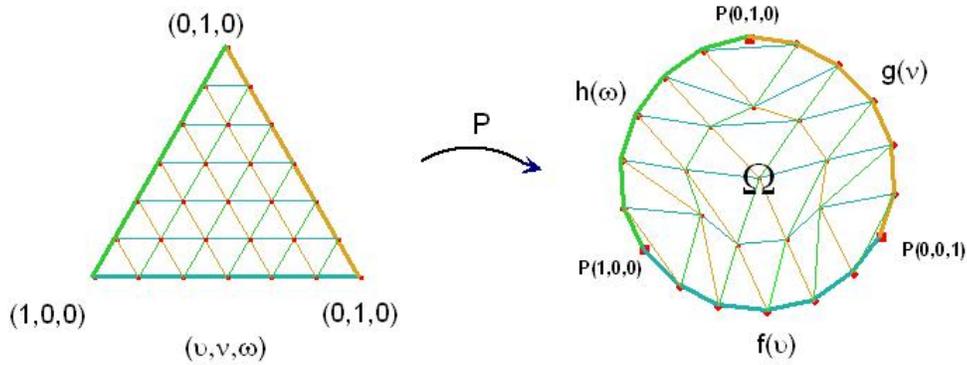


Figura 1.12: Malla por Mapeo de elementos.

### Mapeo Conforme

Este enfoque se puede usar para regiones simplemente conexas que tengan más de cuatro lados definidos, y por lo tanto es más general que el mapeo de elementos, pero la forma y la densidad de la malla son difíciles de controlar. Varios métodos están basados en este enfoque, y el método de Brown se expone aquí, como representante del enfoque. El método consta de cuatro pasos:

- Se parte del supuesto de que la región a mallar es poligonal  $P$  y está trazada en el plano complejo  $\omega$ .
- Se define un polígono ideal  $Q$  en el plano complejo  $z$ , mismo que tiene tantos vértices como  $P$ . Se construye una malla sobre  $Q$  compuesta por elementos de buena calidad y fácilmente generados. La topología de la malla sobre  $P$  será heredada de la topología de malla para  $Q$ .

- Encontrar la transformación de Schwarz Christoffel,  $G$ , que mapea el semi-plano complejo superior  $u$  en el plano interior de  $Q$ , y la transformación  $F$ , del mismo tipo, que mapea el semiplano superior en el interior de  $P$  (véase la figura 1.13)
- La malla generada sobre  $Q$  es mapeada en  $P$  utilizando el mapeo compuesto:

$$\omega = F(G^{-1}(z))$$

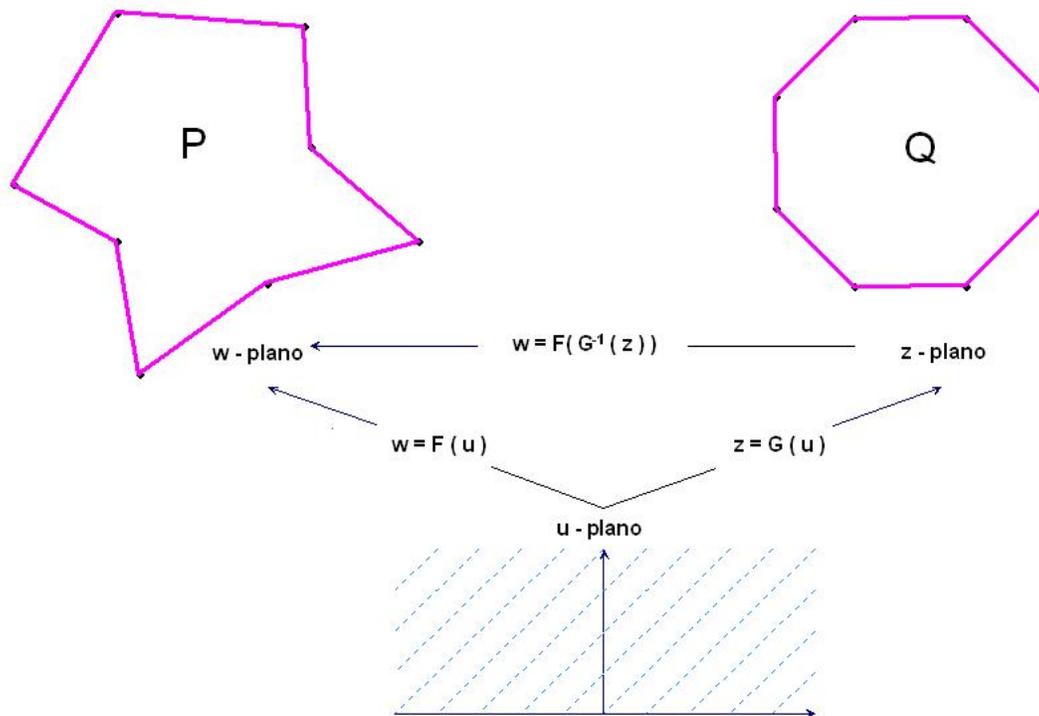


Figura 1.13: Ilustración del método de Brown.

Un problema con este método es que el mapeo inverso  $G^{-1}$  no siempre puede ser hallado.

### 1.4.3. Nodos Primero

Los métodos en esta clase se distinguen por crear en primer lugar los nodos, y luego se conectan para formar elementos triangulares o cuadrangulares, según se desee.

En el enfoque de **Descomposición Topológica** se supone que los vértices de la región poligonal son los únicos vértices de la malla, un algoritmo de triangulación se puede aplicar para crear una malla con el mínimo número de elementos que cubra al objeto.

El nombre de este enfoque es justificado ya que el conjunto mínimo de triángulos está determinado principalmente por la topología de la región, es decir, la complejidad de una región irregular  $\Omega$  se verá reflejada en la incidencia de elementos a cada nodos.

La malla así creada cuenta en su mayoría con elementos *gruesos*, y no se puede utilizar con fines prácticos, así que es necesario refinar la malla de manera selectiva para obtener una malla de buena calidad.

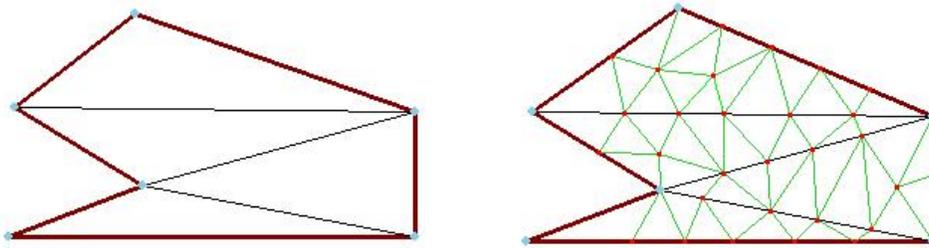


Figura 1.14: Descomposición topológica y su refinamiento.

En el enfoque de descomposición topológica se puede generar mallas cuya densidad esté dada sólo usando refinamiento y de manera semi-automática. El desarrollo de técnicas automáticas que evitan refinamiento para obtener mallas con una densidad deseada representa en todo caso un enfoque diferente.

En este otro enfoque es conocido como enfoque de **Colección de Nodo** se han desarrollado diversas técnicas automáticas que coloquen todos los nodos de la malla final de uno solo, ajustándose a la especificación de una densidad deseada, posteriormente se entra a una segunda fase del generador, donde uno tiene que decidir cómo conectar los nodos de la mejor forma posible.

Este enfoque de generación de malla es muy popular, quizá porque es conceptualmente simple, y abarca la mayor parte de la literatura en la generación de mallas no estructuradas.

Para el mejor entendimiento de este enfoque explicaremos las dos fases, **generación de nodos**, y la **generación de elemento**  $T(S)$ , serán estudiadas de forma separada.

### Generación de Colección de Nodo $S$

Algunos generadores de los nodos requieren la creación manual, pero esto no se acostumbra más que para motivos de investigación; para las aplicaciones, las técnicas automáticas cubren medianamente las expectativas.

Las técnicas de **generación aleatoria** de nodos han sido investigadas y diseñadas; solamente describiremos brevemente la técnica de Cavendish [10].

En la técnica de Cavendish, primeros los nodos de la frontera de  $\Omega$  son añadidos a intervalos regulares. Luego para insertar nodos al interior se usan los requisitos de densidad de malla de la siguiente manera: el objeto se divide en una serie de zonas de diferente tamaño dadas por la densidad, para la  $i$ -ésima zona se sobrepone una cuadrícula de longitud dada  $r(i)$  y para cada celda en la cuadrícula un nodo interior es generado aleatoriamente y es aceptado siempre que este caiga en el interior de  $\Omega$  y que además la distancia a los nodos auxiliares de frontera y a los nodos generados anteriormente sea mayor que  $r(i)$ , si no es así otro nodo aleatorio se genera dentro de la celda para ver si es aceptado, si después de un número fijo de intentos, digamos 5, se realizan sin éxito entonces no se considera un nodo en esa celda.

Para la **generación no aleatoria** de nodos mencionaremos dos opciones.

El método de líneas de Lo [26], plantea tirar líneas paralelas sobre el objeto  $\Omega$  con espaciamiento  $l$ , que es el elemento de longitud deseado para cada zona. Sobre cada línea a distancia se colocan nodos a separación aproximada  $l$  para producir de manera uniforme un conjunto de puntos distribuidos.

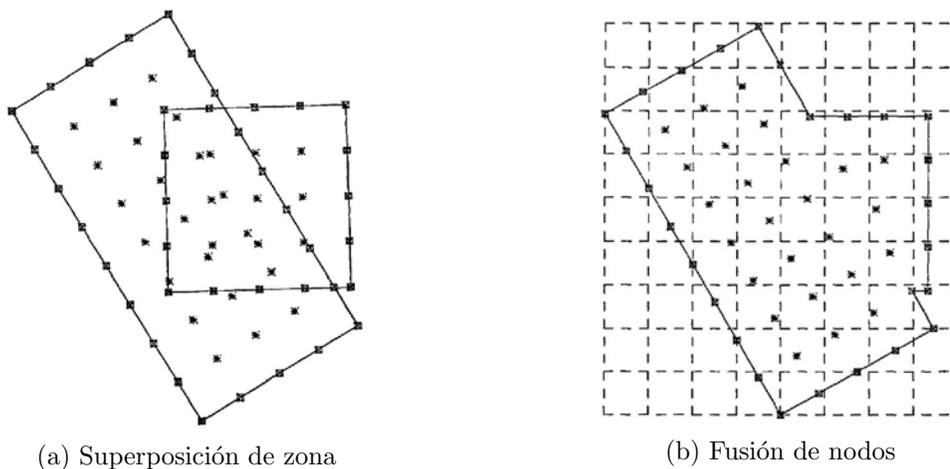


Figura 1.15: Ejemplo de la técnica de Lee.

La técnica de Lee [24], busca la colección de Nodo para cada zona de densidad, basado en una técnica de geometría constructiva sólida para objetos planos, que se creó para la solidez en las operaciones geométricas booleanas. Algunos nodos en la superposición de zonas pueden tener que ser eliminados para mantener la solidez, o puede que sea necesario fusionarlos o perturbarlos, vease figura 1.15.

### Generación de elementos de la triangulación

En esta fase, se busca conectar los nodos para formar los elementos de la triangulación de manera que no se sobrepongan, cubriendo la totalidad del objeto  $\Omega$ . En la mayor parte de la literatura sobre generación de nodos también se encuentran métodos de triangulación. Los métodos que a continuación se indican son representativos y encaminarán la discusión hacia la siguiente sección. Vea figura 1.16

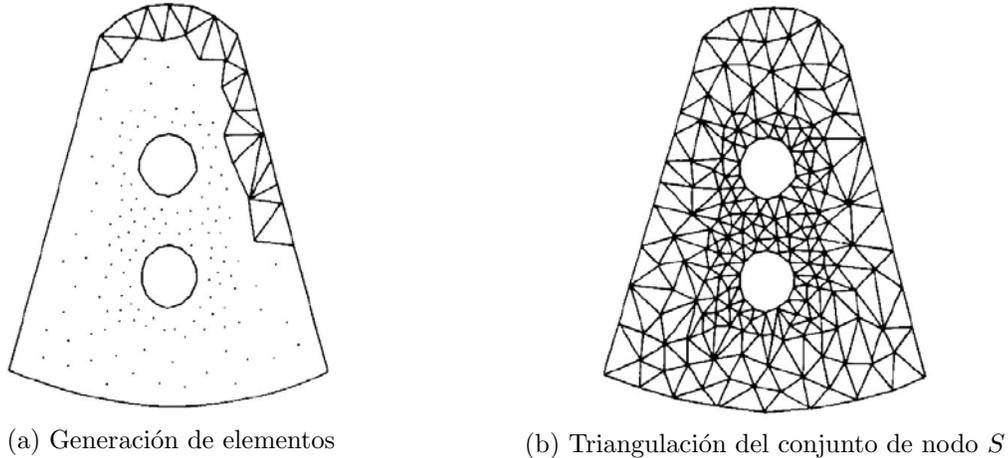


Figura 1.16: Ejemplo de generación de elementos por una técnica de avance hacia el frente.

El método de Lewis y Robinson [25] representa un algoritmo recursivo de tipo “divide y vencerás”, el objeto  $\Omega$  se parte en dos mitades, y se sigue dividiendo cada mitad hasta que sólo se tengan los triángulos. Se hace un intento por encontrar la mejor línea que divida la región en dos mitades de “igual tamaño”, procurando mantener las mitades tan “redondas” como sea posible. Para regiones de geometría elíptica una división a lo largo del eje menor es preferible. Cualquier nodo interior que se encuentre cerca de la línea divisora propuesta se constituye como nodo de frontera de ambas subregiones, dando así un efecto de serrucho, vea figura 1.17.

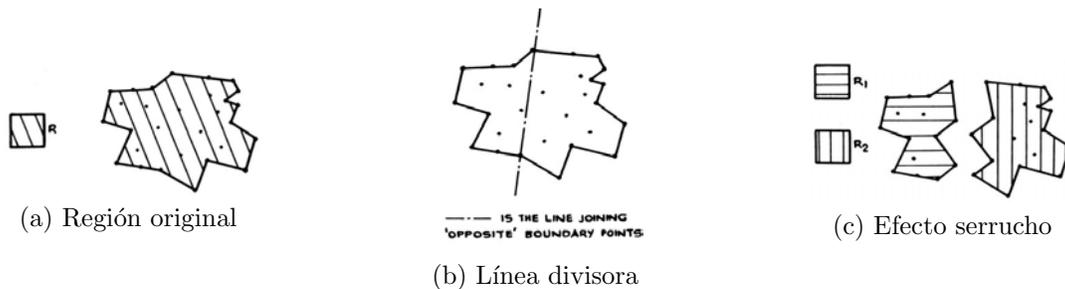


Figura 1.17: Ejemplo de la técnica de Lewis y Robinson.

Para el método de Frederick [13], cada nodo debe ser rodeado con elementos de triángulo. El método comienza por escoger un nodo  $P_i$ , y un nodo más cercano a él,  $P_j$ . El segmento  $\overline{P_i P_j}$  se utiliza entonces para encontrar un tercer nodo  $P_k$  tal que el ángulo  $\angle P_i P_k P_j$  sea un máximo y que el  $\triangle P_i P_j P_k$  esté a la izquierda en una secuencia; el tercer nodo  $P_k$  se convierte en este momento en el primer nodo  $P_i$ , el proceso se repite hasta que el nodo  $P_i$  esté completamente rodeado de triángulos. En cuyo caso se escoge un nuevo nodo  $P_i$  semilla y el proceso se repite hasta haber construido la malla.

Los *métodos de triangulación tipo Delaunay* son quizás los más importantes métodos de triangulación, ya que, una triangulación de Delaunay maximiza el ángulo más pequeño de la malla, creando así la malla de un conjunto determinado de puntos. Dada la importancia y el uso que este trabajo le dará a dichas triangulaciones, dejaremos la explicación de los conceptos y métodos referentes a las triangulaciones de Delaunay para la siguiente Capítulo.

El método de Nelson [27], empieza con cualquier segmento de frontera como una “línea base” y busca un nodo que pueda conectar para formar un primer triángulo, un nodo cuyo elemento resultante no cruce una región fronteriza y de tal forma que todos los demás nodos queden fuera del círculo que circunscribe al elemento resultante, así dos segmentos son creados por la presente relación, uno es útil como nueva “línea base” y el otro se conserva en una pila de espera. Cuando la base de referencia actual es un segmento de frontera o preexiste se escoge una nueva referencia de la pila de espera, cuando la pila está vacía, la rutina ha terminado.



# Capítulo 2

## Método de la ecuación de equilibrio

En este capítulo se describe un sistema de interacción de partículas que se utiliza para distribuir los nodos uniformemente en un dominio  $\Omega$ , con el objetivo es generar una malla con la cualidad de que contar con conectores cuya longitud sea igual a  $h_0$ , valor constante. Como este criterio de optimalidad no es usualmente posible, pues la malla debe corresponder a la geometría de la región  $\Omega$ , Se busca acercarse a esto tanto como sea posible.

Con un esquema de interacción de partículas se busca mejorar la calidad de la triangulación, es decir, la malla resultante debe asemejar la longitud de sus conectores a un solo valor, digamos  $l_0$ .

Para cuantificar la calidad de la malla de la triangulación, hablaremos de dos tipos de irregularidades: *irregularidad topológica de la malla* e *irregularidad geométrica de la malla*.

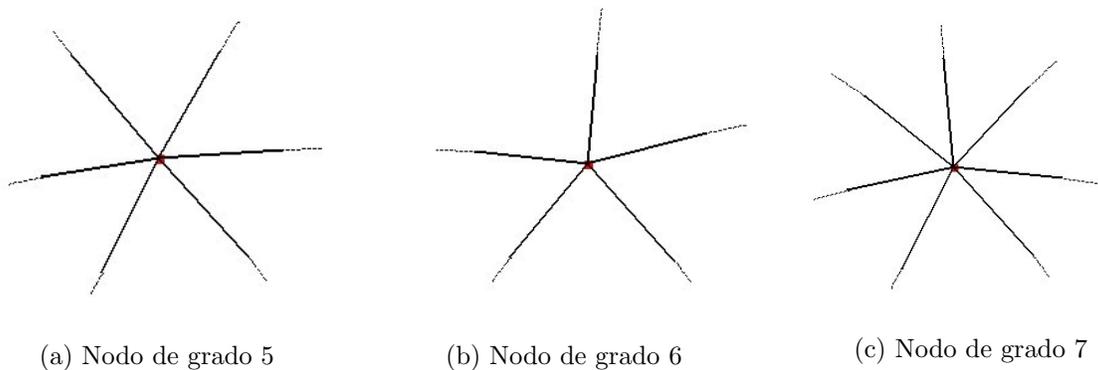


Figura 2.1: Relación de adyacencia de nodos.

Se define la siguiente medida para la irregularidad topológica de la malla:

$$\varepsilon_t = \frac{1}{n} \sum_{i=1}^n |\delta_i - 6|,$$

donde  $n$  representa el número total de nodos interiores de la malla,  $\delta_i$  es el *grado* del  $i$ -ésimo nodo interior, o como se suele decir, número de nodos vecinos, vea figura 2.1. Así que si suele ocurrir que seis elementos comparten un mismo vértice, la malla se acerca topológicamente a una rejilla regular. En el caso general, la medida asigna un valor positivo que refleja que tanto los conectores difieren de ser conectores de una rejilla regular. Si todos los nodos interiores tienen seis vecinos la malla es de la mejor calidad, situación muy particular.

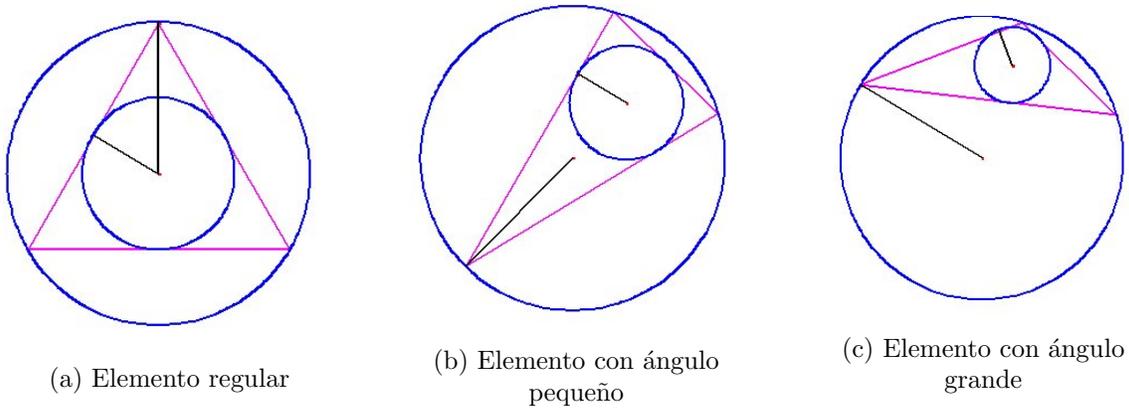


Figura 2.2: Relación de radios.

Para la irregularidad geométrica de la malla, se define la medida,  $\varepsilon_g$ , que es una medida de relación de los radios entre círculos inscritos y círculos circunscritos de los elementos de la triangulación:

$$\varepsilon_g = \frac{1}{m} \sum_{i=1}^m \left( 0,5 - \frac{r_i}{R_i} \right),$$

donde  $m$  representa el número de elementos de la malla,  $R_i$  y  $r_i$  son los radios de los círculos inscritos y circunscritos del  $i$ -ésimo elemento, respectivamente. Para cualquier triángulo la relación  $r_i/R_i \leq 0,5$  con máxima en un triángulo equilátero, como se ilustra en la figura 2.2. Cuanto menor sea el valor de la medida  $\varepsilon_g$ , habrá más regularidad geométrica en la malla. <sup>1</sup>

<sup>1</sup>Consulte Shimada and Grossard [33] si desea conocer extensión de medidas de calidad de una malla 3D.

El capítulo está estructurado de la siguiente manera. En Sección 2.1 se presenta una breve descripción de la teoría de las triangulaciones de Delaunay, donde también se muestran algunos métodos para obtener las triangulaciones de Delaunay. En la Sección 2.2 se describe el método de vigas en equilibrio para el caso de distribución uniforme de los nodos, Persson [28]. Mientras que en la Sección 2.3 presenta un estudio sobre la convergencia del método, para concluir el capítulo con ejemplos del método en la Sección 2.4.

## 2.1. Triangulaciones de Delaunay

Con una Colección de Nudo  $S$  dada en el plano, es posible particionar el espacio en regiones a través de una relación de influencia a cada elemento, es decir cada pedazo del plano es el lugar geométrico de los puntos que se encuentran más cerca del nodo  $p \in S$  que del resto del conjunto,  $S/\{p\}$ , este concepto es conocido como diagrama de Voronoi. Las regiones de influencia son delimitadas por segmentos de mediatriz de un par de nodos y son regiones poligonales convexas no necesariamente acotadas, vea figura 2.3(a).

Este concepto establece una dualidad de diagramas, un diagrama dual se establece al trazar (conectar) los nodos de  $S$  con una arista siempre que en el diagrama de Voronoi exista una cara poligonal (segmento de mediatriz de estos nodos), que en el mejor de los casos las aristas que nombraremos conectores de Delaunay descomponen el casco convexo de  $S$  en regiones triangulares que a su vez nombraremos triángulos de Delaunay como ilustramos en la figura 2.3(b). De ésta manera decimos que la *triangulación de Delaunay*,  $T^*(S)$ , es el diagrama dual al diagrama de Voronoi.

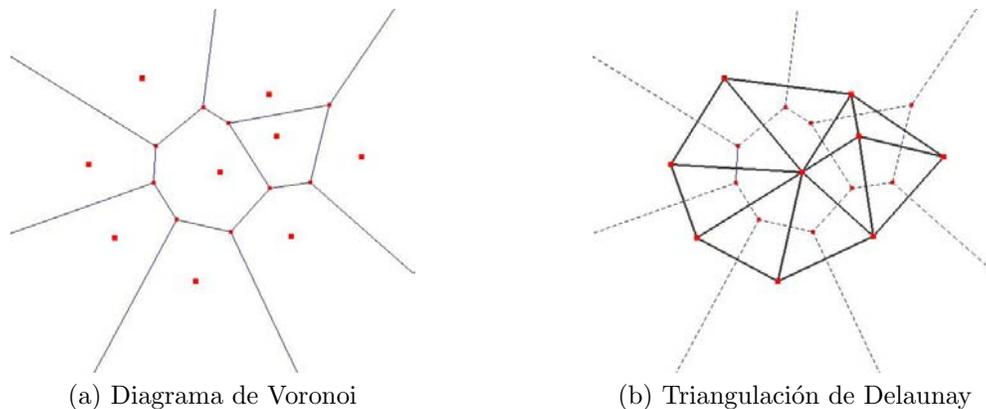


Figura 2.3: Dualidad Voronoi vs Delaunay.

Una característica visible de la dualidad es la ortogonalidad que se preserva entre un lado de Voronoi y una arista de Delaunay, sin que necesariamente dichos segmentos se crucen.

Las triangulaciones de Delaunay tienen un caso especial o singular en el cual sus conectores no encierran regiones triangulares, este caso se presenta si cuatro o mas regiones de Voronoi tienen un punto poligonal en común, es decir si existe un punto que tiene más de 3 nodos de  $S$  con cercanía mínima, situación en la cual la descomposición del casco convexo de  $S$  no estará dada únicamente por triangulaciones. En la figura 2.4 observamos un caso singular donde 4 nodos se encuentran a igual distancia de un punto en el plano. Para este tipo de casos es posible completar la triangulación usando diagonales de la región no triangular y se reconoce la no unicidad de la triangulación de Delaunay.

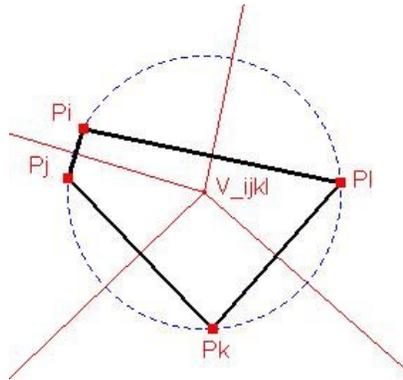


Figura 2.4: Caso singular en una triangulación de Delaunay.

Si descartamos la posibilidad de un caso singular podemos probar dos propiedades importantes de las triangulaciones de Delaunay.

**Propiedad 1.** *Para cualquier triángulo de Delaunay  $\Delta^* \in T^*(S)$  no existe un cuarto nodo de  $S$  en el interior de su círculo circunscrito.*

Una forma sencilla de asimilar esta propiedad es suponiendo la existencia del 4<sup>to</sup> nodo en el interior del círculo circunscrito y observar que bajo la definición de Diagrama de Voronoi no es posible que dicho triángulo sea de Delaunay.

Por medio de este resultado podemos decir que tres nodos están conectados en el sentido de Delaunay y forman un triángulo de Delaunay si y sólo si el círculo circunscrito en él es vacío en el sentido de Delaunay, como se ilustra en la figura 2.5. Este resultado hace una equivalencia con la definición de Triangulación de Delaunay, siempre que ésta sea única, y también es empleada para obtener las triangulaciones, pudiendo construir distintas técnicas con este hecho.

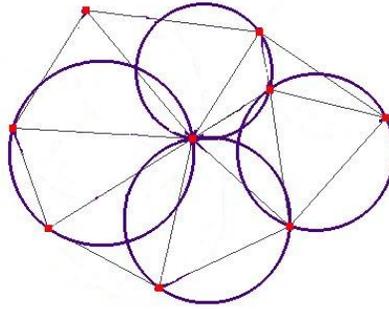
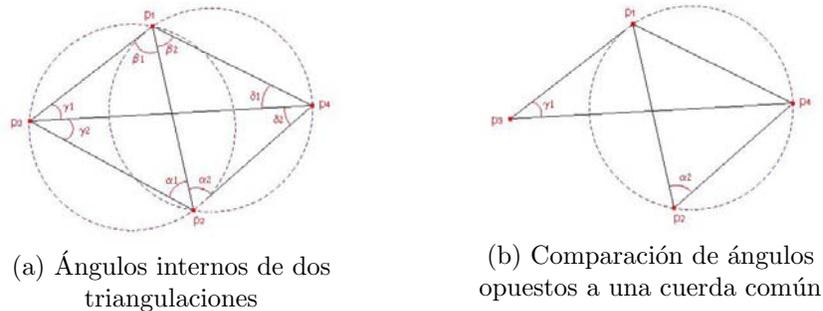


Figura 2.5: Circunferencias vacías en el sentido de Delaunay.

**Propiedad 2.** *El ángulo más agudo de la triangulación de Delaunay es más grande que el ángulo más agudo de cualquier otra triangulación de la lista de nodo  $S$ .*

La demostración formal no es complicada pero sólo haremos un esbozo con una colección de 4 nodos que sólo pueden ser trianguladas de 2 maneras. Esto facilita la comparación y es extensible a una colección de más nodos.



(a) Ángulos internos de dos triangulaciones

(b) Comparación de ángulos opuestos a una cuerda común

Figura 2.6: Ángulo Máx.-Mín.

Comparemos los ángulos de la triangulación de Delaunay  $\{\Delta_{123}, \Delta_{124}\}$  con los ángulos de la otra triangulación posible  $\{\Delta_{134}, \Delta_{234}\}$  de la Fig. 2.6(a), cada triangulación cuenta con 6 ángulos y buscamos saber si el ángulo más pequeño de la triangulación de Delaunay es mayor a algún ángulo de la otra triangulación.

Es posible comparar el ángulo  $\alpha_2$  y  $\gamma_1$  ya que ambos son opuestos al conector  $\overline{p_1p_4}$ , vemos en Fig. 2.6(b) que  $\alpha_2 > \gamma_1$  dado que  $p_3$  está fuera del círculo circunscrito  $\Delta_{124}$ . Con un argumento análogo tenemos las siguientes desigualdades:  $\beta_2 > \gamma_2$ ,  $\beta_1 > \delta_2$  y  $\alpha_1 > \delta_1$ . A los ángulos  $\beta_1 + \beta_2$ ,  $\alpha_1 + \alpha_2$  los podemos acotar con los ángulos  $\beta_1$  y  $\alpha_1$  respectivamente. Lo cual establece que todos los ángulos de la triangulación de Delaunay son más grandes que algún ángulo de la otra triangulación, en particular el ángulo más pequeño no está en la triangulación de Delaunay.

### 2.1.1. Algoritmos para obtener la triangulación de Delaunay

Hay varios algoritmos que sirven para crear una triangulación de Delaunay a partir de un conjunto de puntos. Las técnicas han evolucionado hasta conseguir algoritmos muy veloces para la obtención de la triangulación, presentaremos las técnicas más populares.

#### Divide y vencerás

Los algoritmos de tipo “divide y vencerás” son teóricamente rápidos y muy elegantes, pero de difícil implementación, el algoritmo que aquí se describe es publicado por Lee y Schachter [23]. De lo que se trata este método es de partir un problema en dos subproblemas más pequeños (más o menos del mismo orden cada uno), construir recursivamente las soluciones de cada una de esas dos partes y después combinarlas de forma tal que se obtenga la solución al problema original. Cuando los subproblemas son lo suficientemente pequeños típicamente se pueden resolver mediante métodos triviales, así que, en esencia, lo más importante de esta metodología radica en como realizar efectivamente el proceso de unión.

Sea pues  $S = \{p_1, p_2, \dots, p_n\}$  el conjunto de nodos en cuestión. Para poder aplicar el algoritmo necesitamos ordenar el conjunto en forma lexicográfica ascendente, es decir

$$p_i < p_j \Leftrightarrow \begin{cases} x_i < x_j \\ \text{ó} \\ x_i = x_j, y_i < y_j \end{cases}$$

Si el conjunto de nodo  $S$  no cuenta con dicha estructura, hay que ordenarlo y renombrar los índices de forma que  $p_1 < \dots < p_n$ , de donde denotaremos

NI(S): Nodo más a la izquierda y abajo de la colección  $S$ , o sea  $p_1$ .

ND(S): Nodo más a la derecha y arriba de la colección  $S$ , o sea  $p_n$ .

La separación de  $S$  en dos subconjuntos

$$S_I = \left\{ p_1, \dots, p_{\lfloor \frac{n}{2} \rfloor} \right\} \text{ y } S_D = \left\{ p_{\lfloor \frac{n}{2} \rfloor + 1}, \dots, p_n \right\},$$

implica encontrar  $T^*(S_I)$  y  $T^*(S_D)$  (puede esto hacerse recursivamente) y con ellas encontrar la triangulación  $T^*(S)$  a partir de la unión satisfactoria de las triangulaciones derecha e izquierda.

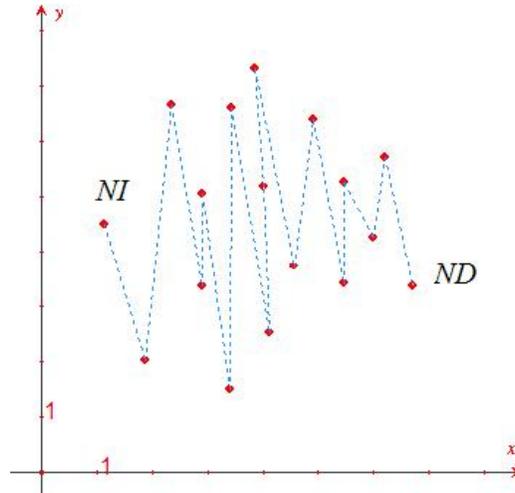


Figura 2.7: Ordenamiento lexicográfico inducido.

Para tener una estructura de la triangulación adecuada a este método es necesario que para cada uno de los nodos conservemos una lista de los nodos con él conectados, diremos así que el nodo  $p_i$  tiene una lista de adyacencia  $Ady(p_i) = \{p_{i_1}, \dots, p_{i_k}\}$  siempre que  $\overline{p_i p_{i_j}}$  para  $j = 1, \dots, k$  sea conector localmente de Delaunay (figura 2.8). La estructura de la triangulación de Delaunay se almacena en una lista por cada nodo. Cada lista se ordena a manera de pivote, es decir de forma circular contra las manecillas del reloj, con un criterio para decidir la posibilidad de empezar con un primer vecino. Esta decisión de inicio de la lista  $Ady(p_i)$  es útil para realizar la unión adecuada de dos triangulaciones, por lo cual, la importancia de inicio recae en los nodos de frontera, ya que posiblemente serán sujetos a conexión entre  $S_I$  y  $S_D$  para la triangulación unión.

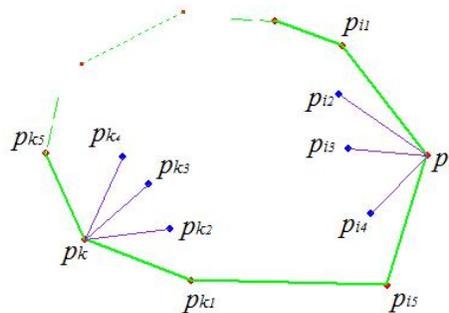


Figura 2.8: Lista de adyacencia de  $p_i$  y  $p_k$ .

Criterio:

- Si el nodo  $p_i$  es interior de la triangulación podemos iniciar la lista de adyacencia con el vecino de orden mínimo o en caso de existir uno o más nodos de frontera adyacentes, se partiría del nodo de frontera con orden mínimo.
- Si el nodo  $p_i$  es de frontera conviene iniciar en el nodo que le siga sobre la frontera, corriendo ésta en el sentido contrario a las manecillas del reloj.

De esta forma se establece la relación de orden cuyo inicio está establecido.

**Primero( $p_i$ ):** Nodo que se encuentra “a la cabeza” de la lista de adyacencia  $Ady(p_i)$ .

Y se establecen las relaciones geométricas entre  $p_i$  y algún vecino, digamos  $p_{i_j}$ .

**Anterior( $p_{i_j}; p_i$ ):** Nodo anterior a  $p_{i_j}$  de la lista circular  $Ady(p_i)$ .

**Posterior( $p_{i_j}; p_i$ ):** Nodo posterior a  $p_{i_j}$  de la lista circular de  $Ady(p_i)$ .

Para realizar la unión adecuada haremos nueva lista de adyacencia con la estructura de cada triangulación, por lo que iremos agregando conexiones y despreciando otras preexistentes, como denotamos a continuación

**Insertar( $p_i, p_j$ ):** añade al nodo  $p_j$  en la lista de  $Ady(p_i)$ , y añade al nodo  $p_i$  en la lista de  $Ady(p_j)$ .

**Borrarr( $p_i, p_j$ ):** elimina al nodo  $p_j$  en la lista de  $Ady(p_i)$ , y viceversa.

Para tomar los criterios con que se va insertar o borrar conectores de la triangulación el método contempla tres problemas geométrico booleanos a resolver.

$$\text{encirculo}(p_l; \triangle p_i p_j p_k): \begin{cases} 1 & \text{Si } p_l \text{ está adentro del circuncírculo de } \triangle_{ijk} . \\ 0 & \text{En caso contrario.} \end{cases}$$

$$\text{DdeL}(p_k; \ell(p_i, p_j)): \begin{cases} 1 & \text{Si } p_k \text{ está la derecha de } \ell(p_i, p_j) = p_i + \lambda \overrightarrow{p_i p_j}. \\ 0 & \text{En caso contrario.} \end{cases}$$

$$\text{IdeL}(p_k; \ell(p_i, p_j)): \begin{cases} 1 & \text{Si } p_k \text{ está la izquierda de } \ell(p_i, p_j) = p_i + \lambda \overrightarrow{p_i p_j}. \\ 0 & \text{En caso contrario.} \end{cases}$$

Para darnos a la tarea de encontrar la Triangulación  $T^*(S)$  dadas las triangulaciones  $T^*(S_I)$  y  $T^*(S_D)$ , debemos primero identificar las *Tapas* inferior y superior comunes, que podemos entender como los dos únicos conectores de frontera a la triangulación objetivo que no figuran en la lista de adyacencia de ambos subconjuntos,  $T^*(S_I) \cup T^*(S_D) = \{Ady(p_i)\}_{i=1, \dots, n}$ . Ver figura 2.9.

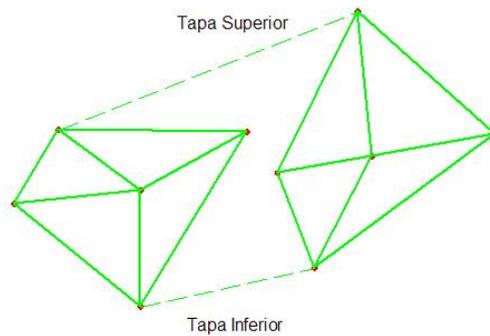


Figura 2.9: Tapas inferior y superior.

Para encontrar la tapa inferior :

Consideremos a los nodos  $p_M = NI(S_D)$  y  $p_{M+1} = ND(S_I)$  como extremos iniciales de un conector dinámico que actualiza sus extremos hasta hallar los extremos izquierdo y derecho del conector de tapa inferior común,  $T_{inf}$ . La idea es partir del conector  $\overline{TD} = \overline{p_M p_{M+1}}$  e ir cambiando los extremos del conector, hasta que IS y DS queden del lado izquierdo de  $\ell(I, D)$  lo cual sólo sucede para un conector de inferior de frontera.

```

I ← ND(SI)
D ← NI(SD)
DS ← Primero(D)
AUX ← Primero (I)
IS ← Anterior (AUX;I)
Repetir hasta caso favorable
.   Si DdeL(DS;ℓ(I,D)) es cierto, hacer
.       AUX ← DS
.       DS ← Posterior(DS;D)
.       D ← AUX
.       ℓ(I,D) ← ℓ(I,D)
.   Si DdeL(IS;ℓ(I,D)) es cierto, hacer
.       AUX ← IS
.       IS ← Anterior(IS;D)
.       I ← AUX
.       ℓ(I,D) ← ℓ(I,D)
.   En otro caso
.       Tinf ←  $\overline{TD}$ 
.       Pare el ciclo

```

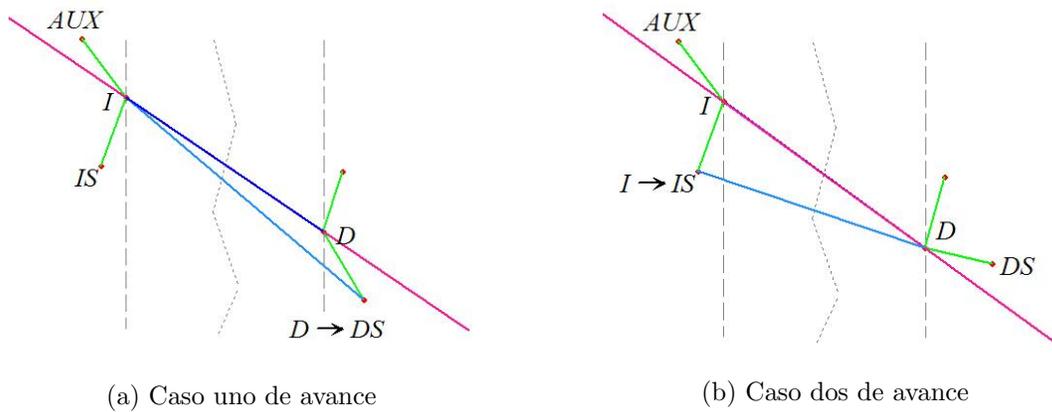


Figura 2.10: Hasta no haber caso favorable.

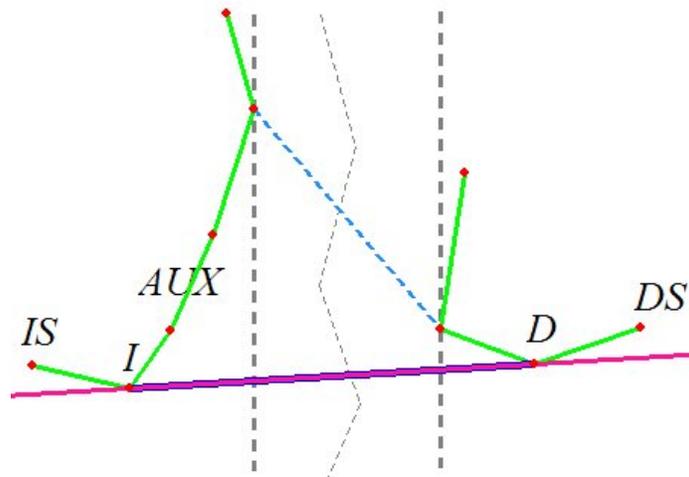


Figura 2.11: Caso favorable de tapa inferior.

Para la Tapa superior ( $T_{sup}$ ) es posible generar un algoritmo análogo que busque convergencia hasta que IS y DS queden del lado derecho de  $\ell(I, D)$  que sólo ocurre para un conector superior de frontera.

Una vez que se tienen ambas tapas, procedemos a realizar la unión de las triangulaciones mediante la eliminación ordenada de aquellos triángulos que no cumplan con el criterio del círculo sobre el conjunto de nodo  $S$ , así como la inserción de nuevos conectores entre  $T^*(S_I) = \{Ady(p_i)\}$  y  $T^*(S_D) = \{Ady(p_i)\}$ . Esto último se va a realizar metódicamente de abajo hacia arriba, es decir, comenzando por la tapa inferior para el valor inicial del conector  $\overline{ID}$  del método, continuando triángulo por triángulo hasta que el conector interno sea la tapa superior; los triángulos candidatos a eliminación se checan antes de la inserción. Como se describe en el siguiente pseudocódigo.

$I \leftarrow$  extremo izquierdo de Tinf

$D \leftarrow$  extremo derecho de Tinf

Repetir mientras  $\ell(ID) \neq \ell(\text{Tsup})$

.  $A \leftarrow$  Falso

.  $B \leftarrow$  Falso

. Insertar(I,D)

.  $D_1 \leftarrow$  anterior(I;D)

. Si  $\text{IdeL}(D_1; \ell(I,D))$  es cierto, hacer

.  $D_2 \leftarrow$  anterior( $D_1; D$ )

. Mientras  $\text{encirculo}(I; \Delta D_2 D_1 D)$  sea cierto, hacer

. Borrar( $D, D_1$ )

.  $D_1 \leftarrow D_2$

.  $D_2 \leftarrow$  anterior ( $D_1; D$ )

. En otro caso, hacer

.  $A \leftarrow$  Verdadero

. Fin de casos

.  $I_1 \leftarrow$  Posterior( $D; I$ )

. Si  $\text{IdeL}(I_1; \ell(I,D))$  es cierto, hacer

.  $I_2 \leftarrow$  posterior( $I_1; I$ )

. Mientras  $\text{encirculo}(D; \Delta I_2 I_1 I)$  sea cierto, hacer

. Borrar( $I, I_1$ )

.  $I_1 \leftarrow I_2$

.  $I_2 \leftarrow$  posterior ( $I_1; I$ )

. En otro caso, hacer

.  $B \leftarrow$  Verdadero

. Fin de casos

. Si  $A$  es cierto, hacer

.  $I \leftarrow I_1$

. Si  $B$  es cierto, hacer

.  $D \leftarrow D_1$

. Si  $\text{encirculo}(I_1; \Delta I D D_1)$  es cierto, hacer

.  $I \leftarrow I_1$

. en otro caso, hacer

.  $D \leftarrow D_1$

. Fin de casos

.  $\ell(I,D) \leftarrow \ell(I,D)$

. Inserta(I,D)

Este algoritmo tiene una complejidad óptima de  $O(n \cdot \log(n))$ . Esto lo hace atractivo para cuestiones de implementación.

Hay que tener en cuenta lo siguiente: la forma en que se aplica la recursión está inducida por el orden impuesto a los nodos, y obliga a que la triangulación se construya por secciones, en donde cada una de éstas se encuentra dentro de una cierta franja vertical. El problema es que mientras más nodos halla, más angostas serán las franjas.

Esto significa que al trabajar en los niveles más bajos de la recursión, es muy probable que se tenga que tratar con nodos cuya coordenada  $x$  es muy parecida y que, por consiguiente, sean, o parezcan, colineales. Debe procurarse que las rutinas que se utilicen en la implementación sean lo suficientemente robustas, como para que esta situación no vaya a producir resultados numéricos adversos o inconsistencia.

### Trueque de Aristas

Una técnica para construir la triangulación de Delaunay, cuando además de tener  $S$  existe una triangulación inicial. En este caso se pueden ir intercambiando o *corrigiendo* las aristas hasta tener todos los conectores de Delaunay.

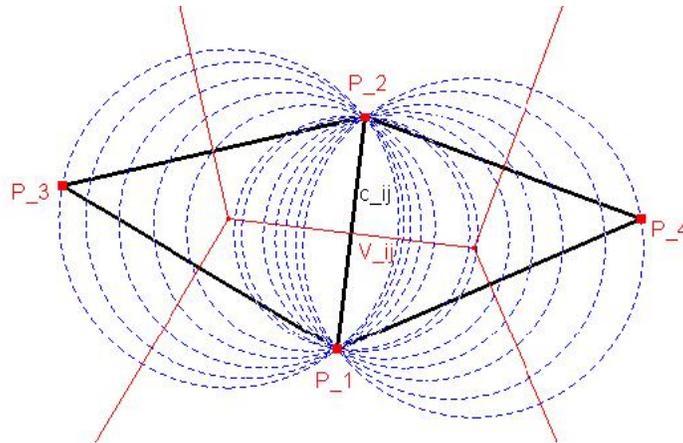


Figura 2.12: Familia de circunferencias vacías con cuerda en común y centro en el segmento de Voronoi.

Para cada conector  $\overline{p_i p_j}$  de Delaunay existe un segmento de Voronoi, es decir, un segmento de la mediatriz y toda circunferencia que pase por  $p_i$  y  $p_j$  con centro sobre el segmento de Voronoi correspondiente no contiene a ningún 3er nodo de  $S$ , es decir, es vacía en el sentido de Delaunay. Dichas circunferencias constituyen una familia de circunferencias vacías con centro en el segmento de Voronoi y con cuerda común  $\overline{p_i p_j}$ , (véase Figura 2.12). Este hecho sólo ocurre para los conectores de Delaunay, localmente de Delaunay, puesto que se desprende de la definición del diagrama de Voronoi. Si encontramos alguna circunferencia vacía con cuerda  $\overline{p_i p_j}$  podemos afirmar que existe el segmento de Voronoi correspondiente lo cual es garantía suficiente para que dicho conector sea localmente de Delaunay.

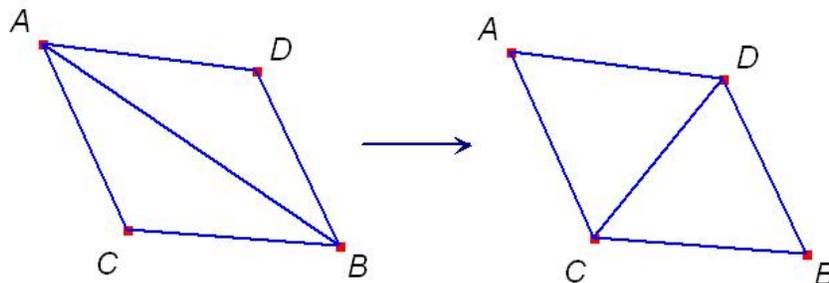


Figura 2.13: Trueque de Aristas 2 a 2.

Así que para realizar el Método de Trueque de Aristas (MTA) hay que tomar una arista de la triangulación y de no ser Delaunay local se corrige haciendo un intercambio 2 a 2, como se muestra en la figura 2.13, siempre que el otro conector sea de Delaunay. En el caso de que ninguna de las 2 diagonales sea conector de Delaunay siempre se puede encontrar un conector localmente Delaunay que sea transversal a la arista a remplazar, se corrige y se completa la triangulación.

El pseudocódigo para MTA es:

Entrada  $T(S)=T_{inicial}$

Mientras no halla convergencia, hacer

- . Tómese una arista  $\overline{p_i p_j}$  interior de  $T(S)$
- . Si  $\overline{p_i p_j}$  no es Delaunay local, hacer
- . Intercambie  $(\overline{p_i p_j}$  por  $\overline{p_h p_k})$  en la triangulación  $T(S)$
- . Para  $\overline{xy} \in \{\overline{p_i p_h}, \overline{p_i p_k}, \overline{p_j p_h}, \overline{p_j p_k}\}$ , hacer
- . Si  $\overline{xy}$  no está en  $T(S)$ , hacer
- . marcar  $\overline{xy}$  en la triangulación  $T(S)$

La única triangulación que tiene todos sus conectores localmente de Delaunay es la triangulación de Delaunay, por lo cual se puede asegurar la convergencia a la triangulación de Delaunay siempre que inter cambiemos cada arista de la triangulación y ésta sea remplazada por un conector de Delaunay.

En la figura 2.14 cada nodo representa una triangulación, si existe un arco del nodo  $\mu$  al nodo  $\nu$  es por que esas dos triangulaciones sólo difieren localmente en una arista, la dirección del arco responde a la exigencia de que el trueque va de una arista que no es localmente de Delaunay a un conector localmente Delaunay. La gráfica de trueques es acíclica e indirectamente todos están conectados.

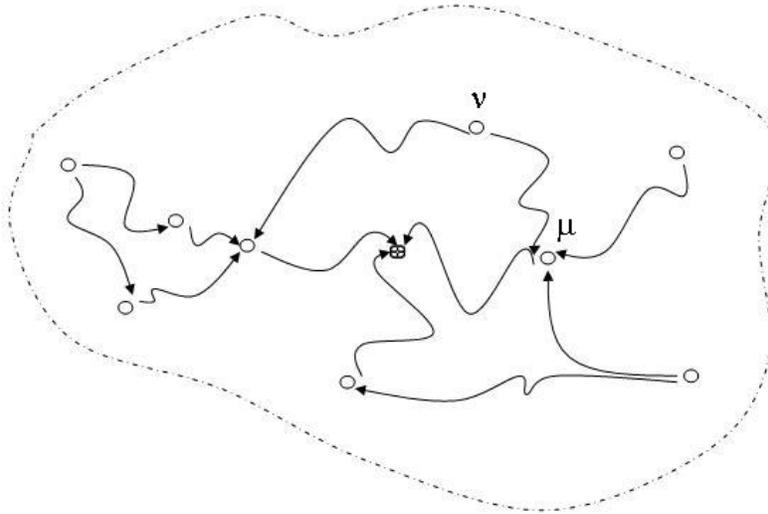


Figura 2.14: Diagrama de convergencia a la triangulación de Delaunay.

### Refinamiento de Delaunay

Cuando sólo se cuenta con la Colección de Nodo  $S$  el método de refinamiento de Delaunay es útil, se puede construir de manera recursiva. Es un algoritmo que combina el trueque de aristas con la adhesión de nodos.

Este método ocupa un orden sobre la Colección de Nodo  $S$ , una vez establecido el índice de orden sobre el conjunto se pretende conocer la triangulación de Delaunay  $T_k^*$  de la subcolección que consta de los primeros  $k$  nodos, como el siguiente pseudocódigo lo explica.

$$S_k \leftarrow S_3$$

$$T_k = \{\triangle p_1 p_2, p_3\}$$

$$k = 4$$

Mientras no halla convergencia, hacer

- .  $S_k = S_k \cup p_k$
- .  $T_k = \text{completar}(T_k, S_k)$
- .  $T_k = \text{MTA}(T_k, S_k)$
- .  $k = k + 1$

Para completar una triangulación al momento de agregar un nodo es importante ver 2 casos, si el nodo agregado se encuentra en el interior de un triángulo se realiza un reemplazamiento 1 a 3, lo cual implica conectar al nodo añadido con los nodos del triángulo existente, véa la figura 2.15. Y si el nuevo nodo se halla en el exterior de la triangulación este nodo se conecta con todo nodo de frontera de  $S_k$  que se encuentre a su alcance, es decir, siempre que no se atravesase un conector existente.

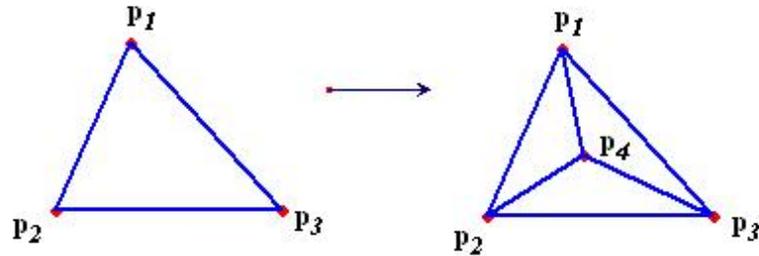
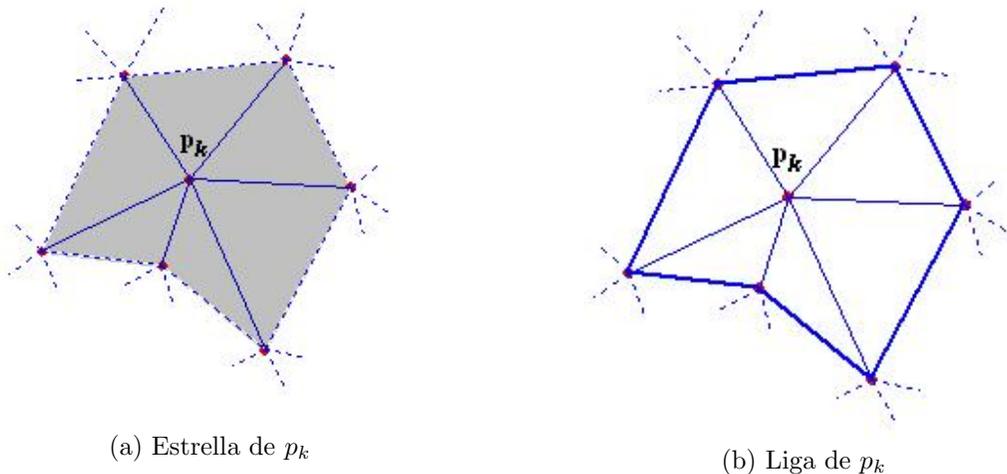


Figura 2.15: Reemplazamiento de un triángulo por inserción de nodo.

Ya que se ha completado la triangulación, el método de trueque de aristas (MTA) se realiza, ya que al haber agregado a  $p_k$  no podemos garantizar que los conectores nuevos sean de Delaunay local, sin embargo no es necesario verificar todos los conectores de la triangulación puesto que la corrección de aristas sólo ocurre alrededor de  $p_k$ . Para evidenciar este hecho nombramos al conjunto de triángulos que tiene a  $p_k$  por vértice como la *estrella de  $p_k$*  y a todo conector de la estrella que sea opuesto a  $p_k$  lo nombramos *liga*, figura 2.16. El trueque de aristas se realiza específicamente sobre aquellos conectores que sean parte de la liga de  $p_k$  y no sean Delaunay local.



(a) Estrella de  $p_k$

(b) Liga de  $p_k$

Figura 2.16: Se corrige la conectividad para que todos los conectores de la liga sean localmente de Delaunay.

### 2.1.2. Triangulación de Delaunay sujeta a restricción de aristas

En esta sección se presenta un estudio de triangulaciones en el plano sujetas a ciertas conexiones especificadas como parte de la entrada. En Chew [12] y Seidel [31] se demuestra que hay una única triangulación con restricciones que es cercana, en cierto sentido, a la triangulación de Delaunay sin restricciones.

Trataremos de identificar aquella triangulación con restricciones que evite ángulos pequeños en la medida de lo posible, usando un concepto de visibilidad entre los puntos.

#### Conectores Fijos

A lo largo de esta sección se han construido triangulaciones para un conjunto determinado de puntos, una nueva variante surge de la necesidad de construir triangulaciones con ciertas restricciones que se manifiestan como conexiones fijas a pesar de no ser estas localmente de Delaunay. La entrada con que se cuenta ahora es, además de la colección de Nudo  $S$ , un conjunto de segmentos  $L$  dada como conexión de dos puntos de  $S$  por segmento. Es necesario que para cuales quiera dos segmentos en  $L$  no se atraviesen y a lo más concurren en sus extremos. Para dichas entradas, se desea construir una triangulación de  $S$  que contenga todos los segmentos de  $L$  como elementos aristas.

#### Triangulación de Delaunay con restricciones

Se dice que dos puntos  $x, y \in \mathbb{R}^2$  son visibles uno respecto al otro si el segmento  $\overline{xy}$  no contiene ningún punto de  $S$  y no atraviesa ningún segmento de  $L$ . Formalmente,  $int(\overline{xy}) \cap S = \emptyset$  y  $int(\overline{xy}) \cap L = \emptyset$ . Asumiendo la posición general de Delaunay diremos que, un segmento  $\overline{ab}$ , con  $a, b \in S$ , pertenece a la triangulación de Delaunay con restricciones dada para  $S$  y  $L$  si:

. (i)  $ab \in L$

ó

. (ii)  $a$  y  $b$  son visibles uno respecto del otro y exista un círculo que pase por ambos tal que cada punto dentro de este círculo sea invisible desde cualquier punto  $x \in int(\overline{ab})$ .

Se dice que el círculo en (ii) es testigo ocular de que el segmento  $\overline{ab}$  es conector de la triangulación de Delaunay con restricciones  $T^*(S, L)$ , notando que el concepto se ajusta a la triangulación de Delaunay siempre que  $L = \emptyset$ .

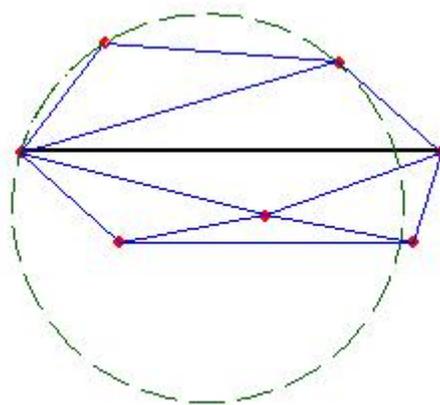


Figura 2.17: Triangulación de Delaunay con una restricción.

### Trueque de aristas

Si se introduce un concepto generalizado de conector localmente Delaunay es posible usarlo para demostrar que la definición anterior tiene sentido. Sea  $T(S, L)$  cualquier triangulación con restricciones dada por  $S$  y  $L$ , un segmento  $\overline{ab}$  en  $T(S, L)$  es localmente Delaunay si  $\overline{ab} \in L$ , o  $\overline{ab}$  es parte del casco convexo de  $S$ , o hay otros dos nodos  $c, d \in S$  tales que el punto  $d$  se encuentra fuera de la visibilidad del punto  $c$ , y además  $\triangle abc, \triangle abd$  fueran elementos de la triangulación  $T(S, L)$

#### **Lema 1.** *Delaunay con restricciones*

*Si cada conector de  $T(S, L)$  es localmente Delaunay, entonces  $T$  es la triangulación de Delaunay.*

La prueba de este lema, se omite por no ser de interés en éste trabajo, para los detalles se pueden consultar las referencias [12] y [31].

Este resultado sugiere que el uso del algoritmo de trueque de aristas visto antes de sujetar la triangulación a restricciones puede ser adaptado para la construcción de la triangulación  $T_L^*(S) = T^*(S, L)$ ; con el ligero cambio de no intercambiar a los conectores  $\overline{uv} \in L$  puesto que por definición ya son aristas localmente de Delaunay. De similar forma, la convergencia es del orden  $O(n^2)$  un poco más veloz que el modelo sin restricción pero sin cambiar de orden. De igual modo el análisis de cambio de ángulos de la triangulación a triangulación se entiende, y el criterio de ángulo Máx-Mín se mantiene en Delaunay sobre cualquier otra triangulación  $T(S, L)$  como se describió en la sección 2.1.1.

### Extensión del diagrama de Voronoi

Así como para  $T^*(S)$ , cada diagrama  $T^*(S, L)$  cuenta con su diagrama dual de Voronoi, pero en este caso resulta un diagrama más elaborado, pues está basado tanto en  $S$  como en  $L$ .

Por la complejidad del problema conviene generalizar el diagrama a partir de un solo segmento de restricción, imagine  $R^2$  como una hoja de papel,  $\Sigma_0$ , con los puntos de  $S$  y el segmentos  $l$  de restricción, para  $l \in L$  haremos una ranura en el papel a lo largo del abierto dado por  $intl$  y pegamos una nueva hoja  $\Sigma_1$ , la cual también es cortada a lo largo de  $l$ .

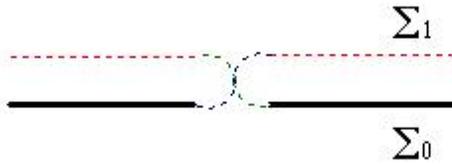


Figura 2.18: “Colage” cruzado en torno a una ranura  $l$ .

Un “colage” cruzado se realiza en torno a  $l$ , de manera que cualquier trayectoria que se trace sobre la hoja  $\Sigma_0$  que se prolonga para atravesar a  $l$  continúe su trazo sobre la hoja  $\Sigma_1$  y viceversa. Para ilustrar el pegado de hojas la figura 2.18 muestra el cruce de hojas por pegado de una sola ranura.

Para el caso de más restricciones llamaremos  $\Sigma_0$  a la hoja principal y en ella pegaremos tantas hojas como restricciones se tenga; cada una de ellas enumerada por un orden dado al conjunto finito de restricciones,  $\Sigma_i$ , ninguna de las hojas secundarias está pegada con otra secundaria. Ahora, para cada punto  $x \in \mathbb{R}^2$ , tenemos una copia del punto  $x_i$  en la hoja  $\Sigma_i$ .

Hemos definido antes el significado de la visibilidad para dos puntos en la hoja primaria, pero es necesario otro concepto para la visibilidad de dos puntos de diferente hoja, generalmente para la visibilidad entre un nodo de la hoja primaria con cualquier nodo en la hoja secundaria  $\Sigma_i$ . De donde surge el concepto de distancia entre los puntos  $x_0$  y  $y_i$ .

$$d(x_0, y_i) = \begin{cases} \|x - y\|, & \text{si } x_0, y_i \text{ son visibles} \\ \infty, & \text{si no son invisibles} \end{cases}$$

La nueva función de distancia se utiliza para definir la extensión del diagrama de Voronoi, que se ilustra en la figura 2.19. Un círculo vacío es testigo por cada conector  $\overline{ab}$  en la triangulación de Delaunay con restricciones que tiene su centro en la hoja primaria o en alguna secundaria. En cualquier casos, aseguramos que el centro de la circunferencia vacía se encuentra más cerca de  $a$  y de  $b$  que de cualquier otro punto en  $S$  con la nueva función distancia. Esto implica que las regiones de influencia de  $a$  y de  $b$  están en contacto a lo largo de una frontera no común. De forma recíproca, todo punto en el segmento de contacto entre dos regiones de influencia (del concepto extendido) es centro de algún círculo vacío testigo del correspondiente conector de la triangulación de Delaunay con restricciones.

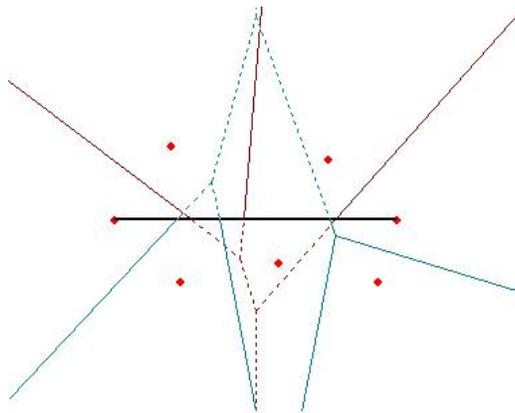


Figura 2.19: Diagrama de Voronoi dual a  $T_L^*(S)$ .

En muchas ocasiones el concepto de Delaunay con restricciones es empleado para mallar regiones poligonales, principalmente para regiones no convexas, para lograr una triangulación adecuada se fijan todos los lados de la región y se descarta todo elemento que quede fuera de la región poligonal. En este documento no se empleará dicha alternativa para las regiones  $\Omega$  que se plantea *mallar*, la alternativa de triangulación de Delaunay que será usada para restringirnos a la región  $T^*|_{\Omega}(S)$  será sin ningún conector fijo, simplemente se descartará a todo elemento que quede fuera de la región  $\Omega$  generando posiblemente un problema de ajuste por permitir que halla una franja o porción de región sin *mallar*, con esto se pretende anunciar que se tomará un rumbo distinto al que sugiere el concepto de triangulaciones de Delaunay con restricción de aristas,  $T_L^*(S)$ .

## 2.2. Método de Persson para triangulaciones en equilibrio de fuerzas

En esta sección se introduce el *método de vigas en equilibrio de fuerzas*, este método es descrito en la tesis doctoral de Persson en [28] en 2004.

Con el método se propone hallar un ajuste con calidad principalmente para regiones de geometría suave a trozos, esto quiere decir que si se tiene una triangulación  $\{\Delta_{ijk} = \Delta(p_i, p_j, p_k)\}$  que esté contenida en la región  $\Omega$  se desea que reposicionando los nodos suceda que la franja entre la malla y la región  $\Omega \setminus \bigcup \Delta(\tilde{p}_i, \tilde{p}_j, \tilde{p}_k)$  sea reducida al mínimo y que la triangulación resultante sea de la mejor calidad posible.

El método de la ecuación de equilibrio de fuerzas propone simular una dinámica de desplazamiento continuo de la posición sobre cada nodo en el conjunto de Nodo  $S(t)$ , se busca que la tendencia en el movimiento de las partículas sea estacionaria,  $S^*$ . Es decir, se pretende hallar una configuración global para  $S$  donde la fuerza que se aplique a cada nodo sea nula. La definición de la fuerza que rige la simulación es de suma importancia para el método, por lo que debemos mencionar que esta definición es dependiente de la estructura de adyacencia de la malla

$$G^*(S; \Omega) = \{\overline{p_i p_j} \mid p_i, p_j \in S \text{ y } \Delta_{ijk} \in T^* \mid_{\Omega} \text{ para algún } k\}$$

a la estructura le llamaremos *estructura de vigas de Delaunay restringida a la región*  $\Omega$ , sugiriendo conocer las componentes finitas de “viga”,  $G^*(S; \Omega) = \{\overline{p_i p_j}\}$  y de “articulación”  $S = \{P_i\}$  de la malla.

Por un momento pensemos que la malla es físicamente elaborada de algún material que sea resistente a contraerse y que tiende a hincharse, como lo hace el concreto.

Imaginemos también que dicha estructura estuviese colocada horizontalmente y encerrada en un espacio que delimita su expansión por medio de una fuerza externa que impide su expansión a través de una frontera rígida.

El problema físico es útil para motivar el método siempre que la estructura de vigas esté bien acoplada, es decir, siempre que la fuerza que se asigne a cada articulación sea dada por la fuerza que aporta cada viga que concurre en él y que a su vez la fuerza que cada viga ejerce sobre sus extremos debe ser dada por el material con que estén hechas.

Si una articulación soporta la fuerza intrínseca de cada viga, este descansará en un estado libre de tensión. Y si todas las articulaciones de una estructura se encuentran en ese estado de relajación diremos que la estructura se haya en equilibrio.

Regresando al método que ha sido motivado geoméricamente, no hablaremos explícitamente de las vigas puesto que los planteamientos serán matemáticos, así que asumiremos las tres siguientes consecuencias de la abstracción del problema.

- Las vigas seguirán siendo nombradas como conectores de la estructura
- Las articulaciones serán nombrados partículas.
- La fuerza que será definida no corresponden a ningún material.

La tercer consecuencia de la abstracción abre la posibilidad de cambiar el tipo de fuerza por un planteamiento matemático, así que será aprovechado para los fines de calidad en que estamos interesados.

### 2.2.1. Ecuación Diferencial de disipación de partículas

Se busca encontrar una malla para  $\Omega$  que sea lo más parecido posible a una rejilla regular, esto a partir de la propiedad de que una rejilla regular es tal que la longitud de todos sus conectores es la misma,  $h_0$ .

A partir de una triangulación  $T = \{\Delta_i\}$  que se encuentre contenida en la región  $\Omega$  se busca obtener una malla  $T^\star = \{\Delta_i^\star\}$ , considerada óptima, realizando un ajuste a la región y pretendiendo que el rango de variación de las longitudes de los conectores,  $[l_{min}, l_{max}]$  sea lo más reducido posible.

El método que se presenta plantea una relación de fuerza-desplazamiento a aplicar a la colección de Nodo  $S_0$ , de la triangulación inicial. Con el desplazamiento de los vértices se busca conseguir el ajuste a la región procurando que el cambio generado en la medida  $\Delta\varepsilon_g$ , de calidad geométrica no sea brusco.

Para la  $i$ -ésima partícula,

$$X_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

se define la fuerza como la suma de fuerzas,

$$F_{total}(X) = F_i(X) + C_i(X) \quad (2.1)$$

donde la fuerza aplicada a la partícula  $i$  puede expresarse como la suma de fuerzas aplicadas a la partícula  $i$  por sus nodos vecinos

$$F_i(X) = \sum_{j \in N^i} F^{ij}, \quad (2.2)$$

la cual sólo depende de la distancia entre las partículas vecinas  $i$  y  $j$ ,

$$F^{ij} = \begin{cases} (l - \|X_i - X_j\|) \frac{X_i - X_j}{\|X_i - X_j\|}, & \text{si } \|X_i - X_j\| < l \\ 0, & \text{si } \|X_i - X_j\| > l \end{cases} \quad (2.3)$$

El valor  $l$  es también un valor dependiente de la estructura, y es calculado para representar un espaciamiento deseado para la dispersión de las partículas y es a este valor al que se compara con cada conector para asignar la fuerza de repulsión, vea figura 2.20(a). Como las interacciones son locales, la partícula  $i$  interactúa sólo con los nodos vecinos  $N_i$ . Note que la fuerza es simétrica en el siguiente sentido,  $F^{ij} = -F^{ji}$ .

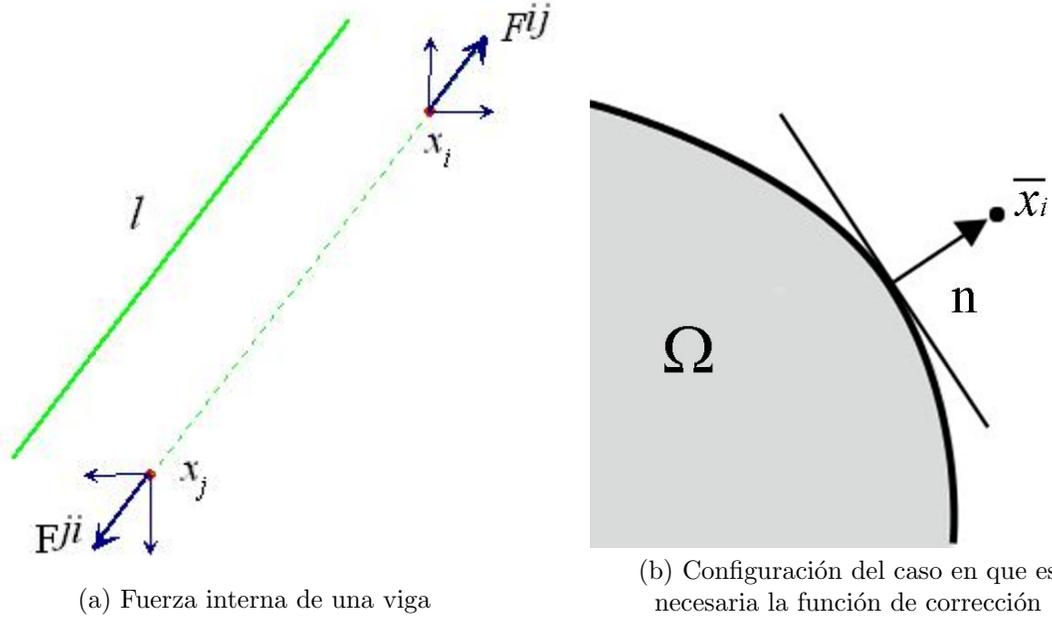


Figura 2.20: Componentes de fuerza.

El valor  $l$  que se busca tener como distancia entre cada pareja de partículas vecinas es calculado para realizar el mejor ajuste. Para controlar el valor de longitud deseado es posible utilizar un promedio de distancias entre vecinos. Siendo la media cuadrática la escogida.

$$l_G^* = F_{scale} \cdot \sqrt{\frac{\sum_{(i,j) \in G} \|p_i - p_j\|^2}{n}} \quad (2.4)$$

La elección de la media cuadrática se da por ser éste valor mayor que la media aritmética, sin embargo es un valor un tanto arbitrario, sólo se busca que  $l$  sea un valor razonable para la aplicación del método por lo que se incorpora un parámetro,  $F_{scale}$ , para aumentar la longitud de  $l$  y con ello incluir a más partículas en la asignación de fuerza. Un valor recomendado para el parámetro  $F_{scale}$  es aquel que incremente la media cuadrática en un 20%.

Ahora bien, la función  $C_i(X)$ , nombrada fuerza de corrección, es impuesta para mantener inmóvil a un nodo o bien para garantizar que ninguna partícula se desplace fuera del dominio acotado  $\Omega$ .

Cuando la partícula  $i$  es considerada como fija,  $C_i(X) = -F_i(X)$  de forma que la fuerza total correspondiente sea  $= 0$ , en el caso en que la partícula  $i$  se halle dentro del dominio, no hay necesidad de aplicar fuerza de corrección, por lo cual  $C_i(X) = 0$ .

Pero siempre que la partícula  $i$  es posicionada fuera de la frontera por la fuerza  $F_i(X)$  hay necesidad de corregir las correspondientes entradas de la fuerza  $f_{total}$  para que la fuerza se mantenga en la región, esto es,  $C_i(X) = -(\hat{n} \cdot F_i(X))\hat{n}$  donde  $\hat{n}$  es un vector unitario normal a la frontera como se indica en figura 2.20(b).

Para ordenar y agrupar la fuerza asociada a los nodos se pensará en los  $2n$ -vectores siguientes, de posición de los  $n$  nodos implicados, de fuerza interna  $F(X)$ , y de fuerza de corrección impuestas por la frontera rígida  $C(X)$

$$X = \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_i \\ y_i \\ \vdots \\ x_n \\ y_n \end{bmatrix} \quad F(X) = \begin{bmatrix} F_1^x(X) \\ F_1^y(X) \\ \vdots \\ F_i^x(X) \\ F_i^y(X) \\ \vdots \\ F_n^x(X) \\ F_n^y(X) \end{bmatrix} \quad C(X) = \begin{bmatrix} C_1^x(X_1; F_1) \\ C_1^y(X_1; F_1) \\ \vdots \\ C_i^x(X_i; F_i) \\ C_i^y(X_i; F_i) \\ \vdots \\ C_n^x(X_n; F_n) \\ C_n^y(X_n; F_n) \end{bmatrix}$$

Cuando una malla se encuentre en equilibrio de fuerzas  $X^\star$ , ocurrirá que  $F_{total}(X) = 0$  y con esto se satisfecerá que  $\|X_i - X_j\| \approx l$  para todo conector.

Considerando que en la naturaleza y en algunos modelos computacionales<sup>2</sup>, el movimiento de las partículas es definido por una ecuación diferencial de segundo orden ( $m\ddot{x} = \sum_i F^i(x, \dot{x})$ ), retomaremos la idea del movimiento de las partículas pero para plantear una ecuación de primer orden, tomando en consideración que la simpleza trae el costo de ser propensos a quedar atrapados en mínimos locales, pero con la ventaja de tener mayor estabilidad.

Para aproximarse a una estructura de partículas en equilibrio de fuerzas, se incorpora artificialmente la variable temporal  $t$  de forma que para las partículas  $S(t)$  se mantenga siempre la topología que brinda la adyacencia de Delaunay, lo que significa que simularemos numéricamente la ecuación de primer orden siguiente

$$\begin{cases} \frac{dX}{dt} = F(X) + C(X) & 0 < t \\ X(t_0) = X_0. \end{cases} \quad (2.5)$$

es decir, se plantea que el campo vectorial definido por el problema de Cauchy es dado por la fuerza total.

---

<sup>2</sup>vea [34] y [7]

### 2.2.2. Implementación numérica de la ecuación de equilibrio

Se desea implementar un generador numérico de mallas que por medio la ecuación diferencial 2.5 realice un ajuste a la región, la relación de fuerza-desplazamiento que artificialmente se ha planteado permite la integración numérica por métodos iterativos, esta posibilidad da lugar a una serie de subproblemas secuenciados que es necesario abordar.

- Ofrecer un generador de triangulaciones iniciales donde todos los elementos estén contenidos en la región  $\Omega$ .
- Definir la estructura  $G^{[K]}$  que regirá la relación de *cercanía* entre las partículas.
- Decidir que método numérico se usará para resolver el problema de Cauchy 2.5.
- Describir hasta que momento se cuenta con una malla optima  $T^*(S^*)$ .

Como la estructura  $G^{[K]}$  regirá la relación de fuerza-desplazamiento puede ser conveniente que la triangulación inicial cuente con la estructura deseada. Es por esta razón que antes de presentar la malla inicial describiremos como se tomará la estructura para el conjunto de Nodos  $S_0$ .

#### Estructura visible y depuración de Delaunay

Para completar la geometría de la región en la estructura de la malla podemos considerar que la topología de la triangulación de Delaunay es adecuada, pero considerando que  $T^*$  es una triangulación de la envolvente convexa de  $S$  y que no todo elemento de la triangulación se encuentra en el interior de  $\Omega$  podemos descartar a algunos elementos para la estructura. Si  $\Omega$  es no convexo habrá algún  $\Delta \in T^*$  tal que  $\Delta \notin \Omega$ . Para resolver este problema hay que elegir una estructura  $G^{[K]}$  que considere que  $\Delta \in \Omega$  para todo  $\Delta \in T^{[K]}$ .

Una primera alternativa sería trabajar con triangulaciones de Delaunay con conectores fijos, de forma que evitemos que halla conectores que atraviesen la frontera de  $\Omega$ , para posteriormente descartar aquellos triángulos que no se hallan en la región  $\Omega$ . Sin embargo la rigidez de los conectores de frontera pueden perjudicar la calidad que se desea obtener en el óptimo y como la distinción entre nodos interiores y nodos de frontera no está dada, habría que hacer una identificación costosa para obtener las restricciones deseadas, así es que consideraremos otra posibilidad más sencilla.

Dada la triangulación  $T^*(S)$  necesitamos dar un criterio simple que nos permita saber si  $\Delta \in \Omega$  para cualquier  $\Delta \in T^*(S)$ .

Como cada triángulo es dado por las coordenadas de sus vértices no resulta tan simple implementar la identificación de aquellos triángulos que no están en  $\Omega$ . Así pues, haremos uso de la siguiente definición.

**Definición 2.** Dada una región simplemente conexa  $\Omega$  y un punto  $p \in \mathbb{R}$ , a la función

$$d_{\Omega}(p) = \begin{cases} \min_{q \in \partial\Omega} \{d(q, p)\} & \text{Si } p \notin \Omega \\ 0 & \text{Si } p \in \partial\Omega \\ -\min_{q \in \partial\Omega} \{d(q, p)\} & \text{Si } p \in \Omega^{\circ} \end{cases}$$

se le nombra distancia a la región  $\Omega$  asociada a la articulación  $p$ .

Ahora, para identificar si un triángulo  $\Delta_{ijk} \in T^*(S)$  se encuentra en el interior de la región tomamos como representante del triángulo al baricentro  $p_{ijk} = (\vec{0p}_i + \vec{0p}_j + \vec{0p}_k) / 3$  y si este se encuentra dentro de la región a por lo menos  $d_{\Omega}(p) > -g_{bnd}$  entonces el triángulo será descartado de la triangulación  $T_{\Omega}^*(S^{[K]})$  y por ende no será considerado en la estructura  $G^{[K]}$ ,  $g_{bnd}$  es un parámetro de *grosor de frontera*  $g_{bnd}$  que por lo general se escoge de forma que

$$\frac{g_{bnd}}{l} \approx 0,01 \quad (2.6)$$

Ahora que hemos establecido como se hará el cálculo de las estructura  $G^{[K]}$  haremos una comparación con la estructura  $G^{[K+1]}$ .

Simplemente, supongamos que los nodos interiores  $a, b, c, d$  de una triangulación de Delaunay formarán una configuración de triángulos adyacentes  $\Delta acd$  y  $\Delta bcd$ . Ahora pensemos que los cuatro nodos serán desplazados de su posición,  $\tilde{a} = a + \delta_a$ ,  $\tilde{b} = b + \delta_b$ ,  $\tilde{c} = c + \delta_c$  y  $\tilde{d} = d + \delta_d$  de forma que la configuración de la triangulación no se conserve, vea figura 2.21.

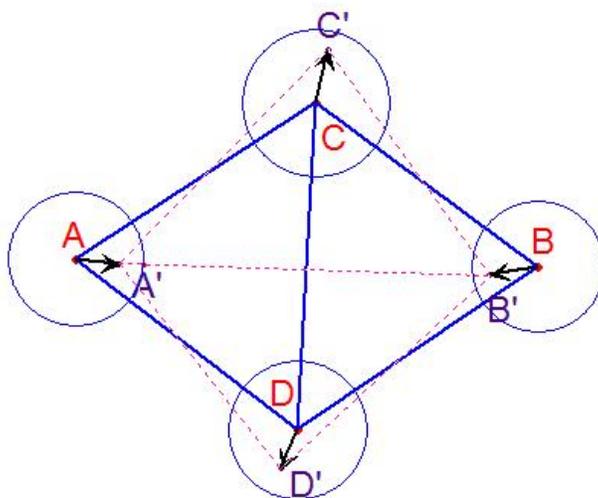


Figura 2.21: Cambio en conexión de nodos ocasionado por perturbación.

Para mantener la conectividad en términos de la triangulación de Delaunay es necesario corregirla cada vez que los nodos sean perturbados, en caso de desear hacer más veloz la obtención de la triangulación de Delaunay es posible considerar útil la conectividad precursora, pues el método de trueque de aristas puede tener una convergencia rápida a la conectividad sucesora siempre que la mayoría de las vigas sean localmente de Delaunay.

La importancia de calcular la fuerza en términos de la conectividad de Delaunay y no mantener fija la conectividad (Topología primero) trasciende a la estabilidad del método. Puesto que si la conectividad se mantiene los elementos se empalman y sucede que todos los nodos son expulsados al exterior de  $\Omega$  y recolocados en la frontera por la función de corrección.

### Malla Inicial

Para ofrecer un generador numérico de triangulación inicial  $\{\Delta_i\}$  sólo se requiere que  $\bigcup_i \Delta_i \subset \Omega$ . Para explicar el generador inicial es posible rescatar la experiencia que se tiene acumulada en la sección 1.4. Retomemos la discusión en el *enfoque basado en rejilla*, las técnicas que se describen resuelven el problema de convertir una rejilla en una malla siempre completando la franja  $\Omega \setminus \cup_i \Delta_i$  ya que  $\{\Delta_i\}$  no es considerada como malla hasta que la frontera de la malla se parezca a la frontera de  $\Omega$ , podemos rescatar de ese enfoque la posibilidad de tener una triangulación de buena calidad que satisfaga todos los elementos de la malla se encuentren en  $\Omega$ .

Partiremos de la construcción de la rejilla regular del plano, con parámetro de longitud de lado  $h_0$ .

Se propone realizar una construcción recursiva para la teselación triangular regular, partiendo del triángulo equilátero de lado  $h_0$  cuyos vértices en coordenadas cartesianas son  $a(0, 0)$ ,  $b(h_0, 0)$ ,  $c\left(\frac{h_0}{2}, \frac{\sqrt{3}}{2}\right)$ , figura 2.22(a).

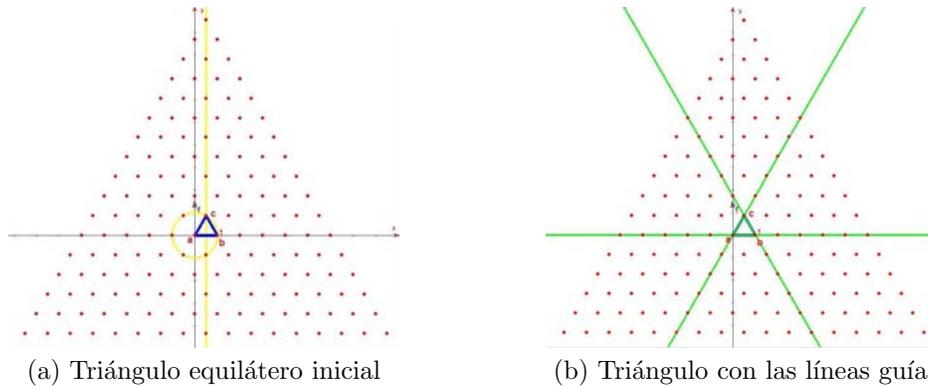


Figura 2.22: Principio para la construcción de la Teselación equilátera del plano.

El primer paso en la construcción es extender las tres aristas del triángulo linealmente, a las tres rectas resultantes les llamamos *guía*, como se muestra en figura 2.22(b). Una vez establecidas las líneas guía procedemos a construir cada capa de la recursividad realizando un trazo recto por cada vértice de intersección de dos líneas, cada una de las nuevas rectas será trazada de forma paralela una línea guía, las nuevas líneas tiradas para cada capa son útiles para definir nuevos nodos siempre que dos rectas se crucen.

Se busca completar los nodos de la capa anterior añadiendo los obtenidos en la recursión. Para ilustrar la construcción presentamos las primeras cuatro capas del método en las siguientes imágenes.

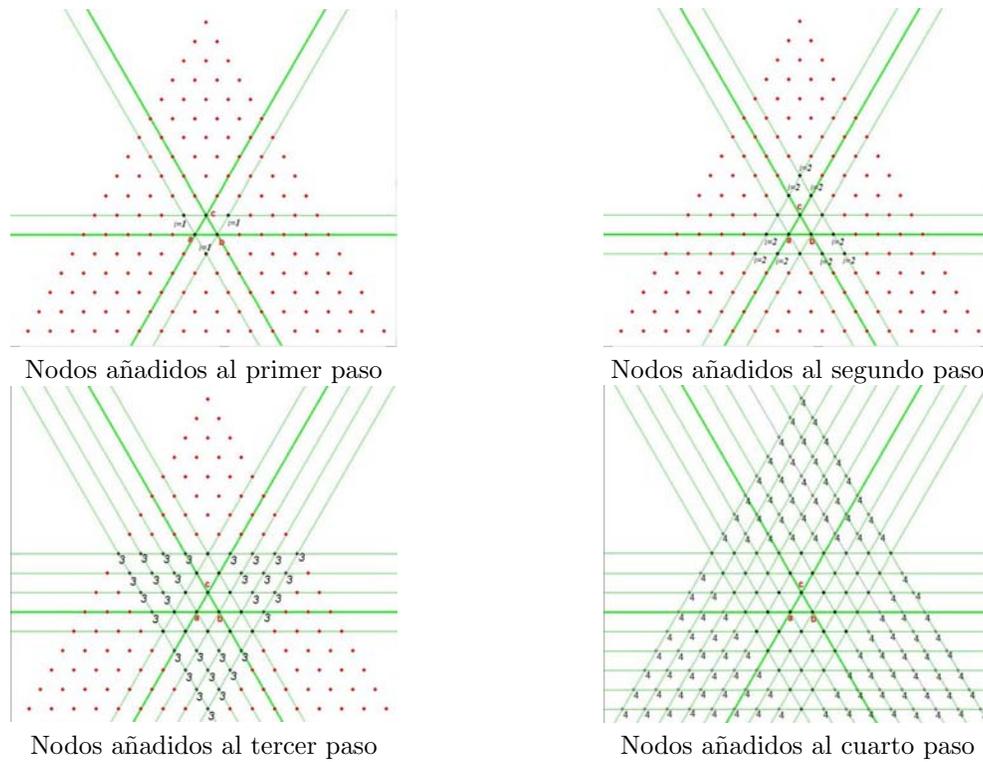


Figura 2.23: Primeras cuatro capas del método recursivo.

Una vez construida la rejilla procedemos a caracterizar a los elementos de la teselación  $\Upsilon$  a partir de los vértices de la rejilla  $\Sigma$  como  $\Upsilon = T(\Sigma)$  y también la estructura  $G(\Sigma)$

Las coordenadas cartesianas de los vértices de la rejilla pueden ser numerados de la siguiente forma.

$$\Sigma = \left\{ p_{i,j} = \left( \frac{h_0}{2}i, \alpha h_0 \left( j + \frac{1 + (-1)^{i-1}}{4} \right) \right) \right\}_{(i,j) \in \mathbb{Z} \times \mathbb{Z}}$$

La estructura se puede describir como

$$G(\Sigma) = \left\{ \begin{array}{l} \overline{p_{i,j}p_{i+1,j}}, \\ \overline{p_{i,j}p_{i+2,j}}, \\ \overline{p_{2i,j}p_{2i+1,j-1}}, \\ \overline{p_{2i+1,j}p_{2i+2,j+1}} \end{array} \right\}_{(i,j) \in \mathbb{Z} \times \mathbb{Z}},$$

y la triangulación como

$$T^*(\Sigma) = \left\{ \begin{array}{l} \triangle p_{2i,j}p_{2i+1,j}p_{2i+2,j}, \\ \triangle p_{2i+1,j}p_{2i+2,j}p_{2i+3,j}, \\ \triangle p_{2i+1,j}p_{2i+3,j}p_{2i+2,j+1}, \\ \triangle p_{2i,j+1}p_{2i+1,j}p_{2i+2,j+1} \end{array} \right\}_{(i,j) \in \mathbb{Z} \times \mathbb{Z}}$$

Para construir la triangulación inicial sólo nos interesan aquellos elementos que se encuentren completamente contenidos en  $\Omega$ , así que hay que descartar todo elemento que quede fuera de  $\Omega$ . Para poder descartar pocos elementos es conveniente que de entrada trabajemos con un subconjunto de  $\Sigma$ .

$$S(a, b, c, d) = \{p_{i,j} \in \Sigma \mid a \leq i \leq b, c \leq j \leq d\} \quad (2.7)$$

A partir del conjunto  $S_0 = S(a, b, c, d) \cap \Omega$  realizamos la triangulación  $T^*_\Omega(S_0)$  como se explicó anteriormente.

Este enfoque para la obtención de nodos se ajusta a las necesidades de calidad para la malla inicial, puesto que la triangulación de Delaunay de este conjunto posee elementos regulares en su interior, notemos que la diferencia primordial con los métodos planteados en 1.4.2 es que no se rellenará con elementos la porción de  $\Omega$  que falta de triangular pues se presenta el problema de ajuste a la región por el método de disipación regular de partículas, dejando del lado el problema de distinguir entre nodos interiores y nodos de frontera.

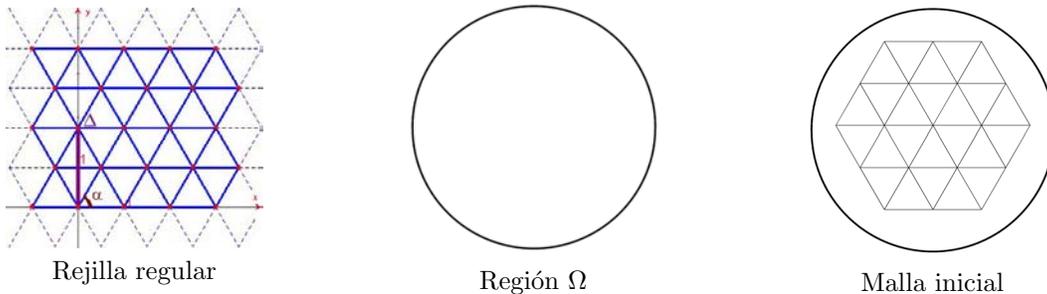


Figura 2.24: Ejemplo de malla inicial para ilustra la técnica.

### Integración numérica

Para resolver la ecuación de equilibrio descrita en la ecuación 2.5, la integración numérica se puede lograr con el método de Euler hacia adelante

$$\bar{x}(t + \Delta t) = \bar{x}(t) + \Delta t \cdot (F(\bar{x}) + C(\bar{x})) \quad (2.8)$$

Donde la notación de tiempo puede ser omitida para dar lugar a la noción iterativa del método, así el proceso de actualización de la posición de las partículas estará dado como

$$\bar{x} \leftarrow \bar{x} + \Delta_t \cdot (F(\bar{x}) + C(\bar{x}))$$

Siempre que una partícula se encuentre lejana a la frontera de  $\Omega$  la fuerza de corrección será nula y no debe resultar problemático actualizar la posición de estas partículas. Tampoco resulta complicado corregir la fuerza si la partícula es inmóvil en todo tiempo.

Si la partícula se halla cerca de  $\partial\Omega$  y es necesario hacer una corrección en la fuerza,  $C_i(\bar{x}) \neq 0$ , resulta más sencillo aplicar la corrección hasta que se ha dado la actualización en la partícula.

Por lo que, la actualización de cada partícula estará dada como

$$\begin{aligned} \bar{x}_i &\leftarrow \bar{x}_i + \Delta_t \cdot F_i(\bar{x}) \\ \text{Si } \bar{x}_i \text{ es fijo, hacer} \\ \cdot \quad \bar{x}_i &\leftarrow \bar{x}_i - \Delta_t \cdot F_i(\bar{x}) \\ \text{Si } \bar{x}_i \notin \Omega, \text{ hacer} \\ \cdot \quad \bar{x}_i &\leftarrow \bar{x}_i + C_i(\bar{x}) \end{aligned}$$

Para realizar la fuerza de corrección cuando es necesario se usa la dirección de máximo crecimiento en la distancia de punto al objeto, de la siguiente manera

$$C(p_i; \Omega) = -d_\Omega(p_i) * \nabla d_\Omega(p_i) \quad (2.9)$$

### 2.2.3. Algoritmo

Partiendo de una estructura  $G_0$  se simula la ecuación diferencial 2.5 introduciendo el parámetro temporal  $\Delta t$  dentro del rango en que el método de Euler hacia adelante es estable.

Parámetros:

$h_0$  Parámetro unico de la rejilla.

$N$  Numero máximo de iteraciones.

$g_{bnd}$  Grosor de la frontera de  $\Omega$

$\Delta t$  Avance temporal de Euler

$F_{scale}$  Escalamiento de la fuerza

Entrada:  $S_\Omega$

$S_0 \leftarrow \Sigma \cap \Omega$

$S \leftarrow S_0$

$K \leftarrow 1$

Mientras  $K \leq N$ , hacer

.  $G \leftarrow G(S, T_\Omega^*(S))$

.  $l \leftarrow l_G^*$

.  $F \leftarrow F(S, G; l)$

. Para  $\bar{x}_i \in S_{fix}$

.  $F_i \leftarrow \bar{0}$

.  $S \leftarrow S + \Delta t \cdot F$

.  $S \leftarrow S + C(S; g_{bnd})$

.  $K \leftarrow K + 1$

## 2.3. Convergencia

Para asegurar la convergencia del método se necesita ver como el método resuelve el problema de mejorar de calidad de una malla.

El método de Liapunov es técnica que usualmente se aplica a una ecuación diferencial ordinaria para poder determinar si hay o no convergencia asintótica, con este resultado analítico se podría garantizar la convergencia en el problema continuo. Aquí supondremos que dicha estabilidad asintótica está garantizada por la forma en que ha sido planteada la ecuación diferencial, así que nos concentraremos en garantizar la estabilidad numérica del algoritmo que ha implementando.

La dinámica simulada por la relación fuerza-desplazamiento tiende a la relajación de fuerzas; toda partícula se desplaza en la dirección en que la suma de fuerzas lo dicta y lo hace con una magnitud suficiente para simular el problema continuo. En ese sentido las partículas, en conjunto, avanzan en pos de acabar con la fuerza que les rige y se aproxima a una estructura  $G^*$  tal que  $F(S^*) = 0$ , considerada como un estado de equilibrio. Dicho estado estacionario cuenta con las características que resuelven el problema central, dado que la fuerza total es nula para cada partícula.

Esto significa que para la partícula  $i$  y su correspondiente conjunto de vecinos  $N^i$ , ocurre alguna de las siguientes dos cosas: Con esto la distancia de cada pareja de partículas vecinas tiende a la longitud media  $l^*$  regularizando los valores de longitud de conectores.

El *método de Euler hacia adelante* es el método numérico más simple para simular una ecuación diferencial, y por lo tanto una sencilla para seguir una trayectoria solución. La estabilidad del algoritmo depende del parámetro  $\Delta t$ , el cual debemos considerar como un valor pequeño, en este documento nos limitaremos a asegurar que el parámetro  $\Delta t = 0,2$  propuesto por Persson, resulta satisfactorio.

Suponemos que siempre que la región  $\Omega$  sea simplemente conexa podremos hacer una búsqueda exhaustiva hacia un punto de equilibrio hipotético, pero no podemos ir más allá sin considerar la geometría de  $\Omega$ . Con esto se quiere decir que la geometría de la región es un factor determinante para la convergencia del método. Por lo cual no es posible garantizar la estabilidad numérica del método, no para cualquier región poligonal ni para cualquier región de frontera implícita.

Por otro lado, es necesario ofrecer un criterio de paro en donde se considere que el método ha tenido éxito, es decir, hasta que punto se cree tener una buena aproximación a un estado estacionario. Para establecer el criterio se plantea una función optimal que permita medir el desplazamiento por cada iteración, así que se rastrea el nodo de mayor suma de fuerzas, y mientras la magnitud de fuerza disminuye entendemos que  $0 \ll \|F(G_{[k]})\|$ , por esto retomamos la función optimal de Persson [28].

$$f_{op}(G) = \max_S \left\{ \frac{1}{h_0} \|\Delta t F_{int}(p_i)\| \right\}. \quad (2.10)$$

Consideramos que el cambio de la estructura  $\Delta G$  es geoméricamente imperceptible si la función optimal es numéricamente menor a  $tol = 0,001$ , es decir, menos del 0,1 por ciento del valor único de la rejilla regular  $h_0$ .

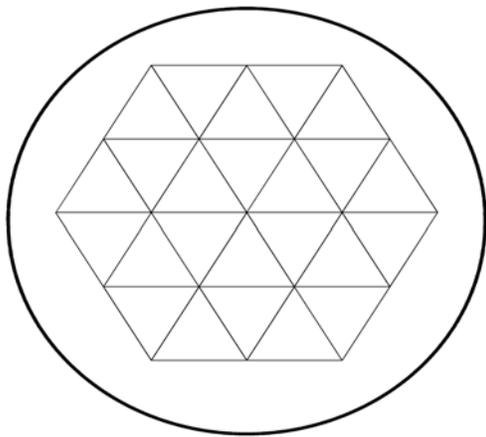
No está establecido un orden de la convergencia de hecho que considera que el método es lento, el número de iteraciones que hay que dar para llegar al minimal depende fuertemente de la geometría de  $\Omega$ , así como de la varianza del tamaño de las vigas de la estructura inicial  $G$ . En algunos casos puede ocurrir que la funcional optimal sea monotonamente creciente sin embargo la mejora en la calidad permite suponer que este efecto sobre el funcional sera momentáneo y se considera al funcional globalmente decreciente.

La estructura inicial  $G_0$  con elementos interiores de calidad (vease 2.2.2), es favorable para obtener lo antes posible una malla en equilibrio de fuerzas, ya que estas triangulaciones requieren menos iteraciones para alcanzar el criterio de paro.

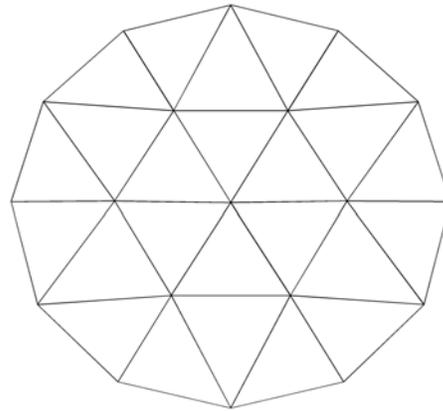
## 2.4. Ejemplos

La siguiente lista de ejemplos ha sido elaborada para ilustrar la convergencia del algoritmo, para cada ejemplo se especificará la región  $\Omega$ , el parametro único de la rejilla  $h_0$ , el numero de iteraciones necesarias para alcanzar la convergencia  $f_{op} < 0,001$ , y el valor  $l_G^*$  que se tiene en el óptimo. El resto de los parámetros se tomarán por default.

- **Círculo unitario.**  $\Omega = \{(x, y) | x^2 + y^2 \leq 1\}$ ,  
 $h_0 = 0.4$ ,  
 31 iteraciones,  
 $l_G^* = 0.64847$



Triangulación inicial



Malla óptima

Figura 2.25: Ejemplo 1 del círculo unitario.

- **Círculo unitario.**  $\Omega = \{(x, y) | x^2 + y^2 \leq 1\}$ ,  
 $h_0 = 0.2$ ,  
 126 iteraciones,  
 $l_G^* = 0.27178$

- **Cuadrado agujerado.**  $\Omega = A \setminus B$ , donde  
 $A = \{(x, y) | -1 \leq x \leq 1, -1 \leq y \leq 1\}$ ,  
 $B = \{(x, y) | x^2 + y^2 \leq 0,4\}$ ,  
 Se añaden los vértices del cuadrado como nodos fijos

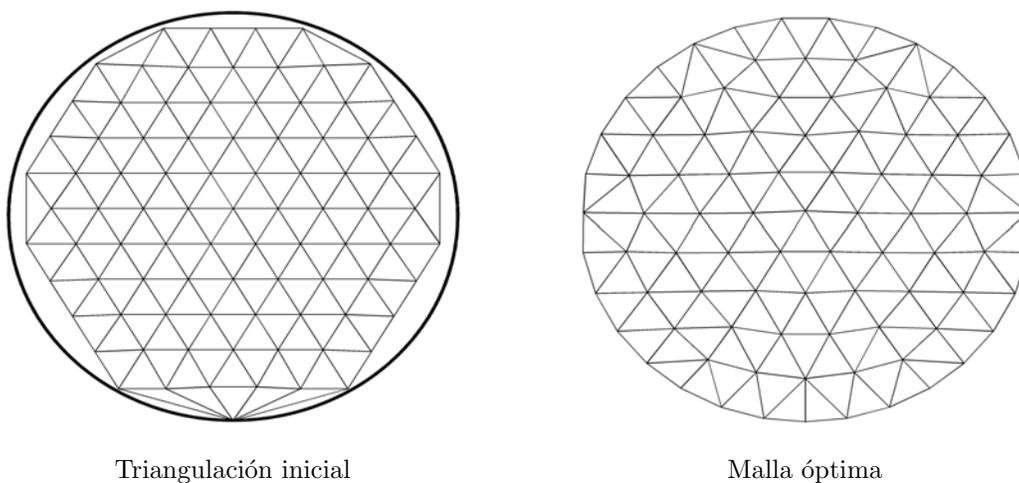


Figura 2.26: Ejemplo 2 del círculo unitario.

$h_0 = 0.15$ ,  
447 iteraciones,  
 $l_G^* = 0.194$

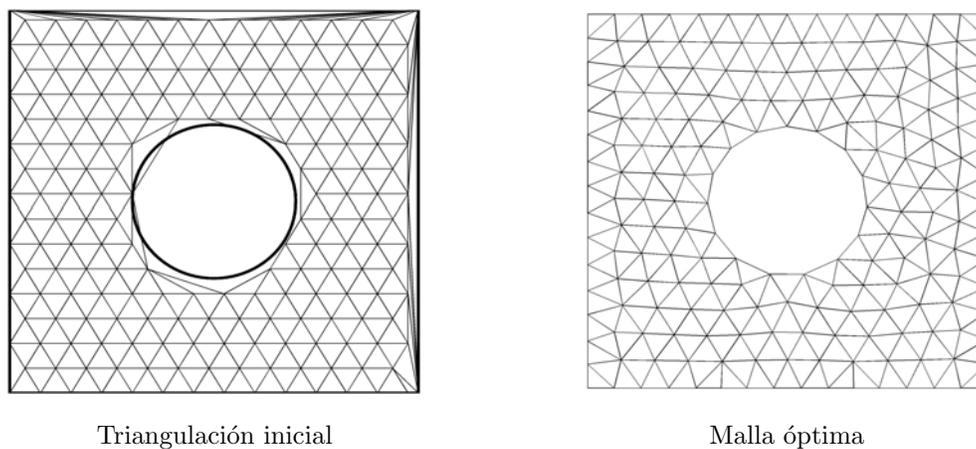


Figura 2.27: Ejemplo de cuadrado agujerado.

- **Hexágono agujerado.**  $\Omega = A \setminus B$ , donde

$$S_A = \left\{ \left( \begin{array}{c} \cos(\theta_i) \\ \sin(\theta_i) \end{array} \right) \mid \theta_i = \frac{i/6}{2\pi}, i = 1, \dots, 6 \right\},$$

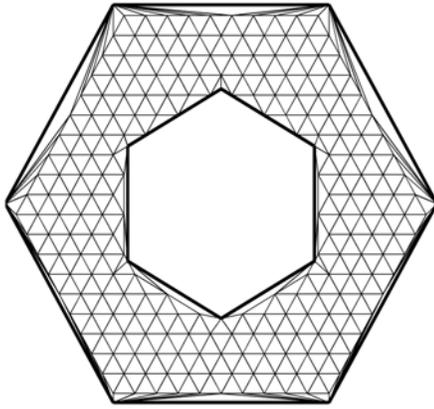
$$S_B = \left\{ \frac{1}{2} \left[ \begin{array}{cc} \cos\left(\frac{\pi}{6}\right) & -\sin\left(\frac{\pi}{6}\right) \\ \sin\left(\frac{\pi}{6}\right) & \cos\left(\frac{\pi}{6}\right) \end{array} \right] \left( \begin{array}{c} x \\ y \end{array} \right) \mid \left( \begin{array}{c} x \\ y \end{array} \right) \in S_A \right\}$$

Se añaden los vértices de ambos hexágonos como nodos fijos

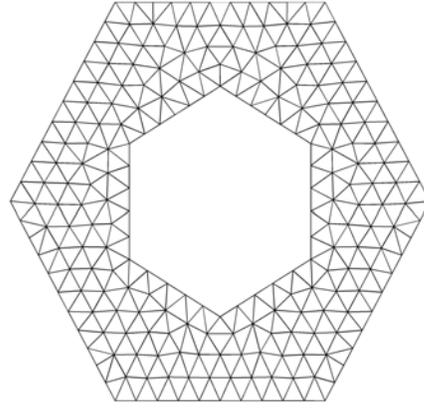
$$h_0 = 0.1,$$

227 iteraciones,

$$l_G^* = 0.13059$$



Triangulación inicial



Malla óptima

Figura 2.28: Ejemplo de hexágono agujerado.

- **Intersección de regiones.**  $\Omega = A \cap B$ , donde  $A = \{(x, y) \mid y \leq \cos(x)\}$ ,

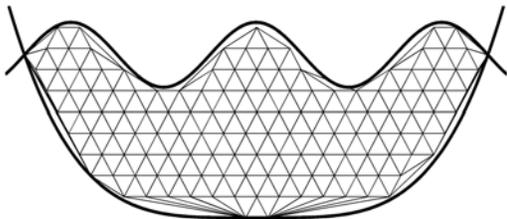
$$B = \left\{ (x, y) \mid 5 \left(\frac{x}{5}\right)^4 - 5 \leq y \right\},$$

Se añaden las intersecciones de las curvas de frontera como nodos fijos

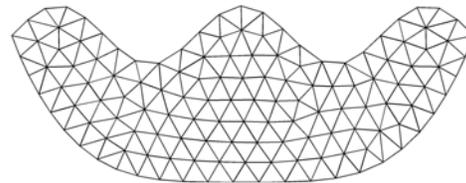
$$h_0 = 0.75,$$

390 iteraciones,

$$l_G^* = 1.00524$$



Triangulación inicial



Malla óptima

Figura 2.29: Ejemplo de intersección de regiones.

- **Región poligonal “Gato”**. La región poligonal será definida por la colección ordenada de vértices

$$S_{\Omega} = \left\{ \begin{array}{l} (0.02,0.79) \\ (0.02,0.50) \\ (0.21,0.50) \\ (0.21,0.31) \\ (0.11,0.31) \\ (0.11,0.02) \\ (0.40,0.02) \\ (0.40,0.21) \\ (0.69,0.21) \\ (0.69,0.02) \\ (0.98,0.02) \\ (0.98,0.31) \\ (0.79,0.31) \\ (0.79,0.60) \\ (0.60,0.60) \\ (0.60,0.40) \\ (0.31,0.40) \\ (0.31,0.50) \\ (0.50,0.50) \\ (0.50,0.69) \\ (0.21,0.69) \\ (0.02,0.79) \end{array} \right\},$$

Que serán considerados también como nodos fijos.

$h_0 = 0.06$ ,

403 iteraciones,

$l_G^* = 0.080789$

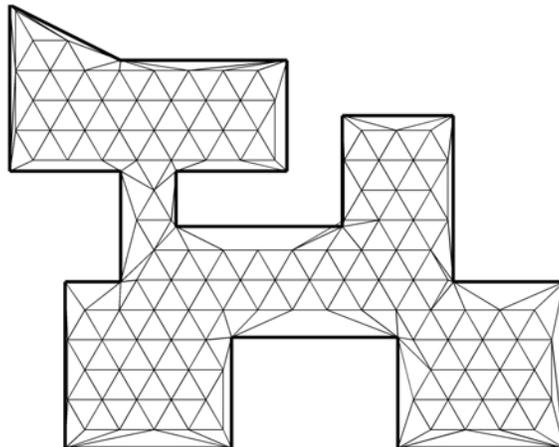


Figura 2.30: Triangulación inicial, “Gato”.

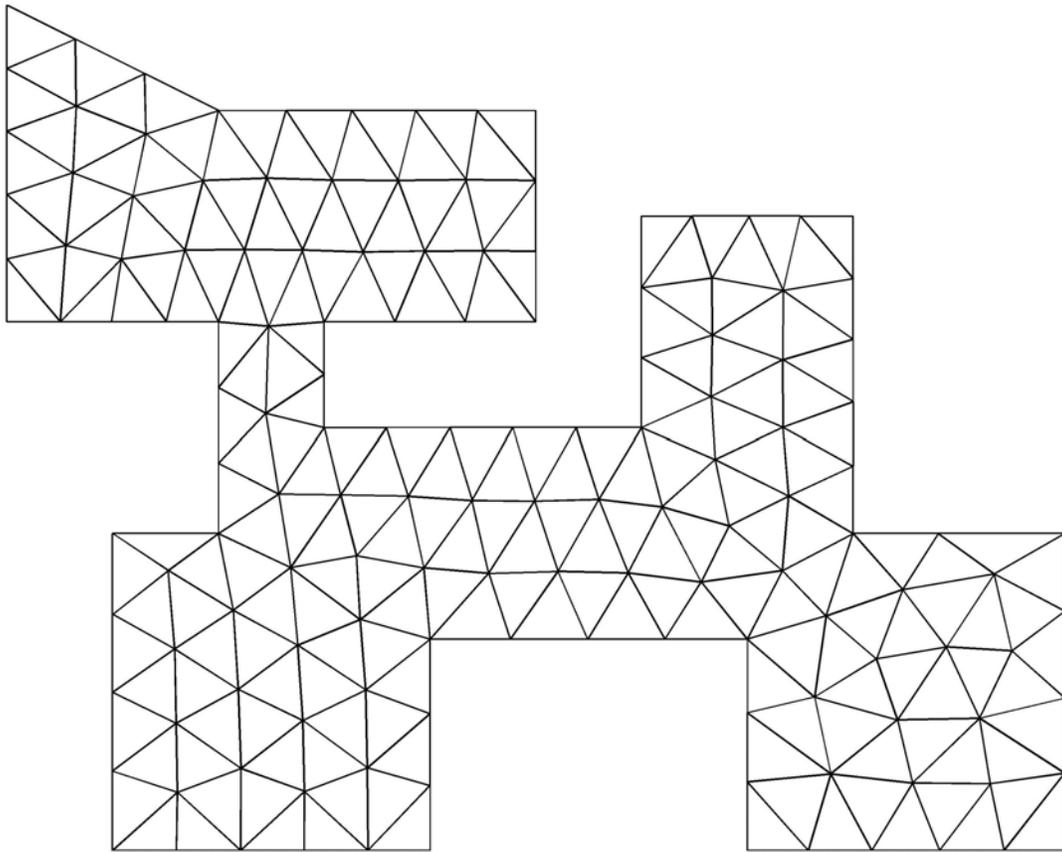


Figura 2.31: Malla óptima, “Gato”.

# Capítulo 3

## Método de vigas en equilibrio para regiones irregulares

La propuesta del capítulo 2 ha sido ideada para la generación de mallas en cualquier tipo de región, y la implementación a la cual se refiere Persson [28] es muestra de que efectivamente es aplicable en regiones con frontera suave. Persson centra el interés en la metodología a seguir para realizar la corrección  $C(x)$  en caso de que la región sea descrita por funciones implícitas como frontera. Para ello muestra cómo simplificar numéricamente el cálculo de la corrección, ecuación 2.9, siempre que la región  $\Omega$  sea el resultado de combinar dos geometrías.

- $\Omega = A \cup B$
- $\Omega = A \setminus B$
- $\Omega = A \cap B$

para dos regiones  $A, B$  descritas por la función implícita de sus fronteras.

El estudio que haremos en este capítulo no seguirá la misma línea pues para el tipo de regiones en se implementa del método de equilibrio se registran otras problemáticas.

La geometría de las regiones de nuestro interés es dada por un conjunto ordenado de vértices como se ha descrito en la definición 1 de la sección 1.2, en la figura 3.1 se puede observar que la irregularidad geométrica que se presenta en una región poligonal simplemente conexa puede ser muy compleja, razón por la cual la implementación a la ecuación de equilibrio plantea retos nuevos.

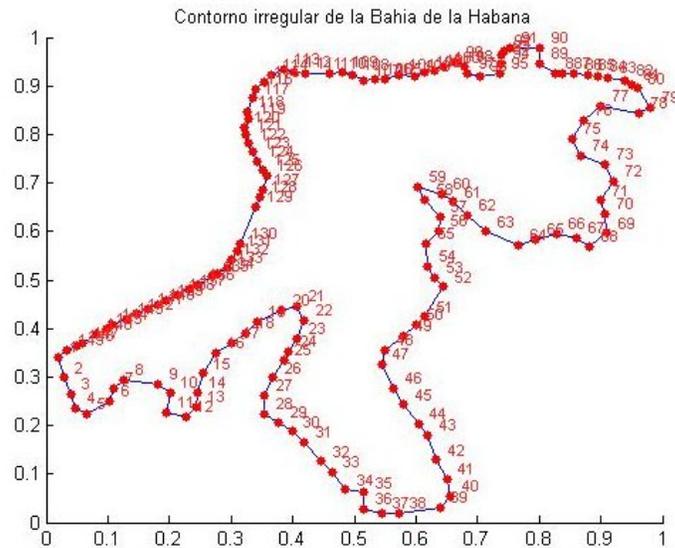


Figura 3.1: Bahía de la Habana, ejemplo de de región con geometría muy irregular.

### 3.1. Malla Inicial

Un primer punto de la discusión es la posibilidad de incluir a los vértices de la poligonal en el conjunto  $S_0$ . Se ha argumentado que, de no hacerlo, el enfoque resultaría ser el anteriormente implementado que hemos visto que resulta favorable para regiones de frontera suave.

Así, se plantea el subproblema de determinar que cualidades geométricas debe tener un vértice de la poligonal para ser incluido como nodo de la triangulación. La opción que no falla para ese cuestionamiento es incluir a todos y mantenerlos fijos a través de la evolución, esta opción ha sido escogida para dejar el cuestionamiento abierto.

La discusión ha dado lugar a la implementación del sistema experimental Unamesh. Teniendo al conjunto de vértices de definición de la región en la colección de nodo  $S$  se plantea la posibilidad de mantener el control de la geometría de la malla a generar, y esto se traduce en conseguir que los lados de la poligonal sean a su vez la frontera de la malla obtenida.

Generalmente el control de la geometría de la región en los métodos iterativos de generación de mallas irregulares optan por mantener la restricción sobre el conjunto de conectores de frontera y mantener la topología en todo el proceso. Sin embargo, como la naturaleza del método de la ecuación de equilibrio no mantiene los conectores, el control de la geometría de la malla plantea el subproblema de definición de nodos de frontera, problema que no ha sido abordado desde la implementación que se maneja en la sección anterior.

La alternativa que aquí se muestra consiste en considerar que toda región poligonal necesita un tratamiento previo al proceso de generación de malla, es decir, se reemplazará el polígono  $P = \partial\Omega$  que describe a la región por un polígono  $Q$  que encierre la misma geometría  $\Omega$  o que tienda a suavizarla, la finalidad del cambio de polígono es tener una nueva colección de nodos que estén distribuidos de mejor manera que los nodos del polígono  $P$ , ya que estos serán nodos fijos de la malla óptima en regularidad.

Se tiene la intención de evidenciar las diferencias entre el conjunto de vértices de una región poligonal  $S_\Omega$  y el conjunto de nodo  $S$  de la triangulación, para poder conjugar de manera adecuada la frontera de la región  $\Omega$  y la frontera de la triangulación. La principal diferencia entre estos conjuntos es que el orden de aparición de  $S_\Omega$  sirve como estructura para que la región simplemente conexa esté bien definida mientras que el ordenamiento de  $S$  no favorece la estructura de la malla.

Una vez que se ha decidido incluir en  $S_0$  a todos los nodos de la poligonal con *tratamiento*, es pertinente adoptar a  $S$  como conjunto con estructura de orden parcial, de manera que las triangulaciones aquí construidas además de ser de calidad, gozarán de la estructura útil para dar las condiciones a la frontera de una ecuación diferencial parcial, la estructura de orden parcial aquí empleada contempla que los primeros  $m$  nodos del arreglo  $S$  sean el conjunto ordenado  $S_\Omega$ , y a partir del nodo  $m+1$  se enlisten los nodos de la rejilla como ha sido descrito en la sección 2.2.2.

## 3.2. Tratamiento de Contorno

El debate trae nuevos planteamientos sobre como operar el método de la ecuación de equilibrio, pues se desea que en el mejor de los casos la estructura de semi-orden de  $S_0$  se mantuviera a lo largo de la aplicación del método. Supongamos que tenemos una región  $\Omega$  irregular en cuyo interior hemos colocado algunos nodos, esto es suficiente para construir una estructura de vigas inicial  $G_0$  para emplear el método de la ecuación de equilibrio; pero ¿quién nos garantiza que  $S(t)$  mantenga la estructura de semi-orden para  $t > 0$ ?, hasta el momento no hay nada que lo haga. Si existe un lado de la poligonal relativamente largo no habría ningún impedimento para que los nodos “interiores” fueran a parar a la frontera de la región y encuentren su equilibrio a la orilla de la región, en estos casos la estructura de semi-orden dado a la colección  $S$  se debilita.

La alternativa es generar primero los nodos de frontera de la malla con una distribución lo más uniforme posible y posteriormente introducir los nodos interiores, para generar los nodos de frontera se hace el tratamiento previo, un tratamiento tal que la varianza en la longitud de los lados de la poligonal sea menor que en la poligonal original. Para ilustrar el tratamiento que se dará, hacemos el tratamiento de los contornos Avión (figura 3.2) y Bahía de la Habana (figura 3.1) a lo largo del capítulo.

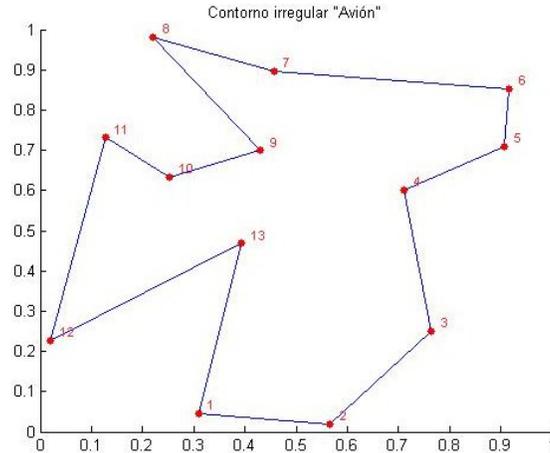


Figura 3.2: “Avión”, ejemplo de contorno con poca irregularidad geométrica.

Para poder identificar como varían las longitudes de los lados de la poligonal podemos recurrir al histograma como un resultado visual de la dispersión de las longitudes 3.3(a), como el histograma no está considerando nunca que las aristas encierran una región resulta ser una visualización carente de información útil para el tratamiento.

Es conveniente ilustrar con otro resultado visual la relación que guardan las longitudes con el orden asociado a los lados del polígono, para este fin se muestra la gráfica de la figura 3.3(b), enumerando los lados conforme el orden en la lista  $S_{\Omega}$  es posible distinguir e identificar cuales son las aristas de longitud similar en la gráfica de un poligonal. En el eje horizontal se usa el orden de los conectores y en el eje vertical su longitud, la visualización simula una gráfica constante a trozos por rescatar que cada viga tiene dos articulaciones extremo. Para completar la gráfica y hacer un estudio más completo se opta por incluir tres rectas, una en la media  $\mu$ , y otras dos altura  $y = \mu \pm \sigma$  que son puntos de inflexión.

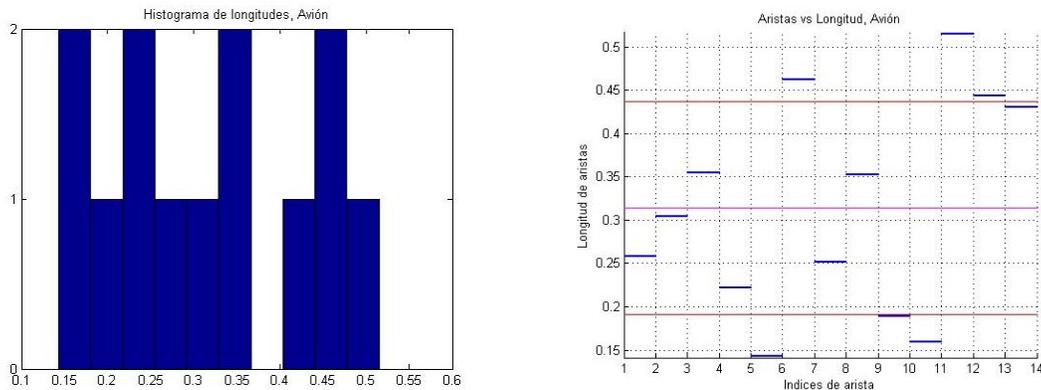


Figura 3.3: Información de longitudes del contorno Avión.

La gran variedad de geometrías que se presenta en las regiones irregulares hace difícil generalizar cualquier tipo de teoría sobre los problemas de la geometría computacional. El tratamiento de contorno de una región irregular ha sido una alternativa para mejorar la calidad de los lados de la poligonal, aquí se trabajó con tres alternativas al problema.

### 3.2.1. Tratamiento spline cónico

La alternativa consiste en encontrar una colección de nodos sobre una curva suave definida por el contorno poligonal, dicha colección de nodos debe ser tal que la longitud de arco entre cualquier par de nodos consecutivos sea constante.

Es decir, al contorno poligonal original que llamaremos *polígono de control* se le asocia una curva suave sustituyendo las esquinas por Arcos Cónicos de tal manera que se conserve lo más posible la geometría del contorno original. Al contorno suavizado se le conoce como spline cónico, y es una curva paramétrica.<sup>1</sup>

La idea de mejorar la calidad de las aristas plantea la generación de nodos con densidades distintas sobre la curva paramétrica, en particular usaremos la distribución uniforme que es resultado de la parametrización por longitud de arco para acotar el problema central a mallas cuasi-equiláteras, figura 3.4.

La técnica depende de dos parámetros, además de ser necesario fijar el número de nodos que se desea en el polígono suavizado ( $N$ ) se solicita saber el *peso*  $w$  asociado al spline cónico, el cual sirve para regular el ajuste de la curva spline al polígono de control, para esto se tiene planteado el parámetro por default  $w=1$  y en caso de desear que la curva se ajuste más al contorno original se usa un peso  $0 < w < 1$ .

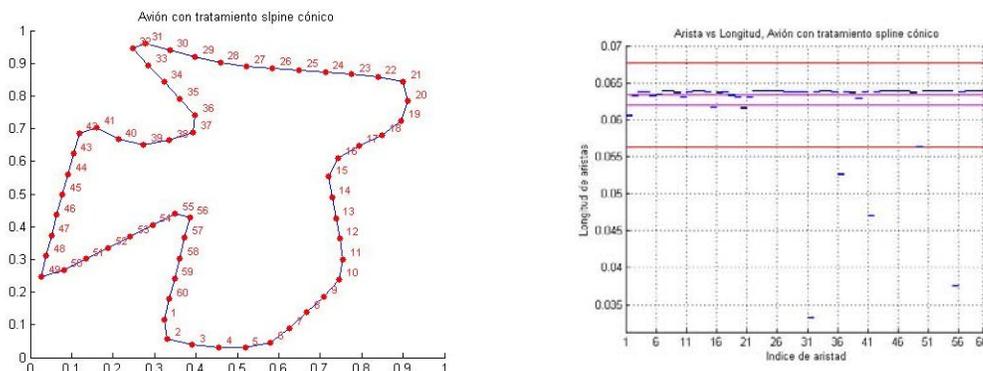


Figura 3.4: Contorno Avión con tratamiento spline cónico.

<sup>1</sup>En el Apéndice B se presenta una breve construcción del spline cónico empleado, para una construcción detallada del Spline y las diferentes parametrizaciones le sugerimos al lector consultar [9] documento en que se basa la implementación del suavizamiento.

Observemos que la poligonal posterior al tratamiento no tiene la misma longitud en cada segmento, esto se debe a que los nodos han sido generados por tener misma longitud de arco sobre el spline. A medida que el ángulo interior del polígono original sea muy cerrado, el tratamiento generará aristas reducidas.

### 3.2.2. Inserción de nodos

La segunda alternativa de tratamiento es una técnica de adhesión de puntos que consiga la conservación de la geometría, la idea es colocar nodos igualmente espaciados en los lados más largos de la poligonal, la idea es útil cuando el polígono original tiene lados largos.

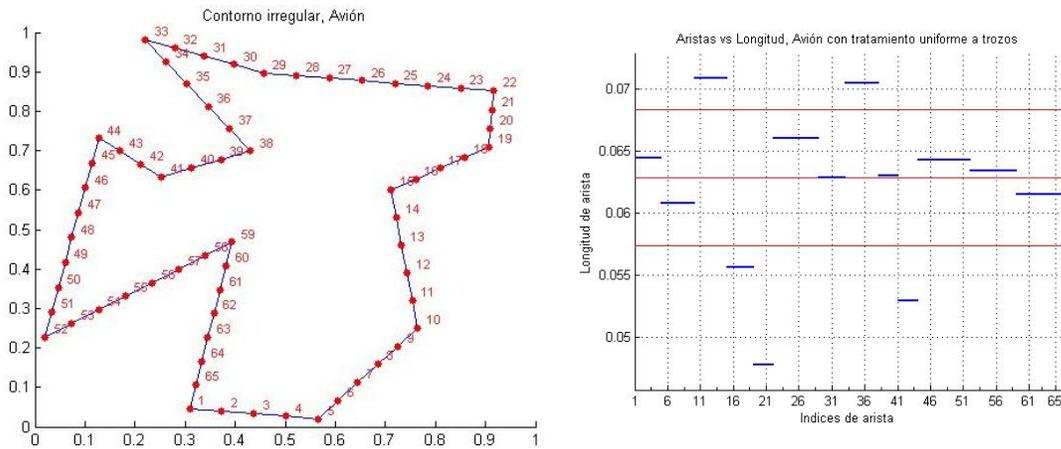


Figura 3.5: Contorno Avión con tratamiento uniforme a trozos.

Esta técnica depende de un parámetro a escoger entre los dos siguientes:

$L_{max}$  Longitud *tope* de segmento, una distancia máxima para nodos consecutivos de la poligonal. Dicho tope estará dado como un porcentaje de perímetro total del contorno.

$N_{min}$  Cantidad de nodos a añadir en el segmento más corto de la poligonal original. (En terminos de este valor se determinará cuantos nodos serán añadidos en el resto de los segmentos).

En el ejemplo, figura 3.5, podemos observar como los segmentos tienen misma longitud siempre que los segmentos son colineales, esto quiere decir que el tratamiento no busca que las aristas no tiendan a un solo valor, sino que el seccionamiento se ha hecho a trozos.

### 3.2.3. Reducción de nodos

Las dos técnicas anteriores son suficientes para dar el tratamiento de contorno previo a la generación de la triangulación, sin embargo un método de tratamiento más es implementado para complementar a los anteriores.

Este enfoque no va dirigido a mejorar la calidad de los conectores de frontera, la finalidad es tener un contorno intermedio entre el original y el contorno de la triangulación, el método es basado en una técnica de eliminación de ruido en tratamiento de imágenes, emplea dos criterios de reducción de nodos por colinealidad a definir con 6 parámetros y busca eliminar los nodos más prescindibles para alterar lo menos posible la geometría de la región<sup>2</sup>.

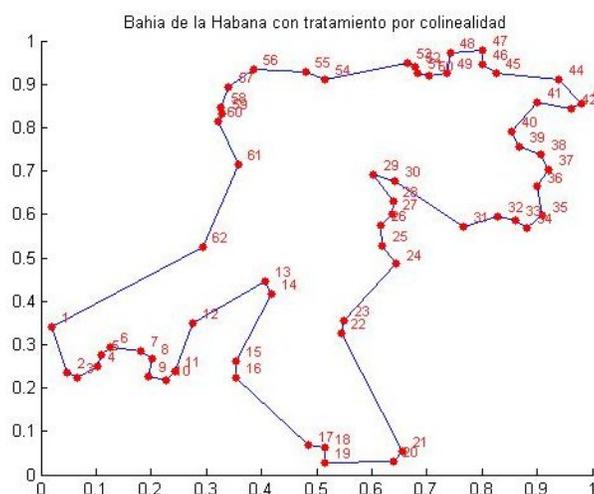


Figura 3.6: Bahía de la Habana con tratamiento por colinealidad.

## 3.3. Nuevas especificaciones

La implementación de la técnica ha sido desarrollada con la modificación del tratamiento previo y así, el sistema funcionaría adecuadamente. Ahora lo que corresponde es replantear la finalidad del método. Si los nodos de frontera se han incluido en la colección de partículas, no se puede pensar que el problema que se tiene es de ajuste a la región puesto que desde la malla inicial se tendría el ajuste.

Si los nodos interiores a la región ya son de calidad, el problema será dotar a los elementos de frontera de la calidad deseada para la solución numérica de una ecuación diferencial parcial. En general la técnica es útil para dar un suavizamiento a la malla.

<sup>2</sup>Para una breve explicación de los criterios de colinealidad consulte el Apéndice C, consulte [29] si desea una explicación más completa.

Seguimos sin poder garantizar que la estructura de semi-orden se preserve al usar el método de la ecuación de equilibrio. Es decir hasta ahora no se ha planteado que hacer si un nodo interior encuentra su equilibrio en la orilla de la región.

Para resolver esta situación se ha optado por replantear que nodos estarán fijos, replantear el uso e implementación del corrector  $C_i(x)$ .

### 3.3.1. $F_i(\bar{x}) = 0$

En general hemos dicho que los nodos de definición de la región serán incluidos en la colección de partículas del método de la ecuación de equilibrio, previo tratamiento de la región. Ahora conviene retomar el cuestionamiento de que nodos mantener fijos para mantener un control de la triangulación.

La necesidad de tener nodos de frontera fijos responde al interés de no perder la geometría de  $\Omega$ ; es decir, para evitar que los nodos de frontera encuentren su equilibrio en el interior de la región. Se ha optado por fijar todos los nodos de frontera si la región no ha recibido ningún tratamiento, o si fue sometida a tratamiento spline.

Si el tratamiento es por adhesión de nodos cabe la posibilidad de no fijar todos los nodos para preservar la geometría ya que la fuerza aplicada es repulsiva. Si solamente se fijan aquellos vértices cuyo ángulo interno sea diferente de  $180^\circ$  estaremos seguros de que la implementación será bien desarrollada.

Lo malo de esta alternativa es que con mayor facilidad un nodo interior puede encontrar su equilibrio en la frontera, lo cual debilita la estructura de semi-orden. Por tal motivo la alternativa no será la opción por default.

### 3.3.2. Función de corrección $C_i$

Para mantener a los nodos de frontera sobre la frontera y a los nodos interiores dentro de la región es necesario hacer algo con aquellos nodos interiores que encuentran su equilibrio en la frontera. Dos opciones tenemos ante esta situación, la primera de ellas es reordenar la colección de partículas  $S$  en cada iteración reestableciendo la estructura de orden parcial y la otra opción es retirar a la partícula conflictiva de la colección  $S$  sin que esto implique reestablecer el semi-orden. La segunda opción resulta favorita para la implementación puesto que se conserva la distribución de nodos de frontera dada por el tratamiento al contorno y la calidad de la malla mejora expulsando a los nodos prescindibles.

Bajo esta alternativa los únicos nodos a los que se aplica la fuerza externa sería a los nodos de frontera, es decir a los primeros  $m$  nodos de  $S$ , y esto sólo en el caso de no ser todos fijos. En estas condiciones es más fácil identificar a que segmento hay que proyectar al nodo saliente por lo cual la implementación que hacemos no usa la distancia  $d(p, \Omega)$  para realizar la corrección, en este caso preferimos hacer la corrección

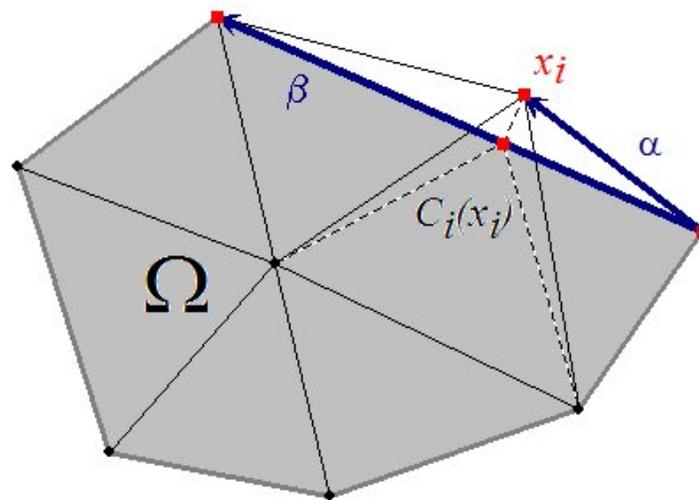


Figura 3.7: Corrección para regiones poligonales.

por medio de la proyección de un vector en otro,  $Pr_{\beta}(\alpha)$ . Vea figura 3.7.

Esta forma de pensar el método, en la que algún(os) nodo(s) interior(es) es(son) prescindible(s) para la malla en equilibrio, en los casos más simples no afecta la estabilidad del mismo, de hecho da facilidades para la convergencia del equilibrio ya que el número de partículas disminuye. La fuga de nodos, así como la entendemos puede generar triangulaciones con menor cantidad de elementos y no hay gran variación en  $l_G^*$  con esta modificación.

### 3.3.3. $L^*$ Fija

Para ampliar el alcance del método de la ecuación de vigas en equilibrio de fuerzas modificado para regiones irregulares, se ha planteado la posibilidad de habilitar al usuario de la oportunidad de fijar el parámetro de la longitud  $l_G^*$  deseada para los conectores en el óptimo. Como el problema de generación de mallas por el método modificado no es propiamente un problema de ajuste a la región dicho parámetro solo es útil si es una magnitud razonable, es decir, la estabilidad del método sólo se da si la longitud promedio del óptimo está dentro de un rango no establecido previamente. El valor default será siendo el calculado para la triangulación inicial como lo indica la ecuación 2.4 sin cambiar uso de escalamiento (reemplazando el parámetro  $F_{scale}$ ), ahora se ofrece la posibilidad de n recalcarlo en cada iteración.

### 3.4. Algoritmo modificado

El método que se propone requiere que la región  $\Omega$  tenga un tratamiento de contorno anterior al proceso de generación de malla. En el proceso de generación por método de vigas en equilibrio se propone que todo nodo de frontera sea partícula fija y que toda partícula interior que se dirija a la frontera sea depurada de la colección  $S$ . Solo aquellos nodos de frontera que no sean útiles para mantener la geometría de la región podran ser sujetos a la corrección en su posición.

Parámetros:

$h_0$ : Parámetro unico de la rejilla.

$N$ : Numero máximo de iteraciones.

$g_{bnd}$ : Grosor de la frontera de  $\Omega$ .

$\Delta t$ : Avance temporal de Euler.

$F_{scale}$ : Escalamiento de la fuerza.

$l_G^*$ : En caso de ser fijo.

Entrada:  $S_\Omega$

$S_\Omega \leftarrow \text{Tratamiento}(S_\Omega)$

$S_0 \leftarrow (\Sigma \cap \Omega) \cup S_\Omega$

$S \leftarrow S_0$

$S_{fix} \leftarrow S_\Omega$

$K \leftarrow 1$

Mientras  $K \leq N$ , hacer

.  $G \leftarrow G(S, T_\Omega^*(S))$

.  $l \leftarrow l_G^*$

.  $F \leftarrow F(S, G; l)$

. Para  $\bar{x}_i \in S_{fix}$ , hacer

.  $F_i \leftarrow \bar{0}$

.  $S \leftarrow S + \Delta_t \cdot F$

. Para  $\bar{x}_i \in S_\Omega \setminus S_{fix}$ , hacer

.  $S \leftarrow S + C(S; g_{bnd})$

.  $S \leftarrow S \cap \Omega$

.  $K \leftarrow K + 1$

### 3.5. Convergencia

En el capítulo anterior dimos la primera discusión sobre la convergencia del método de vigas, ahí vimos que si  $\Delta_t = 0,2$  el método está garantizada, también vimos que aun es posible tomar valores más grandes sin que la estabilidad del método numérico deje ocurrir, lo único que se ha dejado abierto es el valor máximo de  $\Delta_t$  puesto que el valor es dependiente de la geometría de la región  $\Omega$ . Cualquier modificación un método puede afectar la convergencia del mismo, por lo cual toda modificación debe ser debidamente probada.

Para hablar de la convergencia del método modificado podemos tomar como base los resultados de la sección 2.3, ya que el método ha sido probado con distintos ejemplos. La diferencia entre el método original y el método modificado comienza desde el tipo de regiones que se desea triangular.

Por lo que al momento de realizar las modificaciones se tiene la intención de que el método funcione para el tipo de regiones que se desea, garantizando que la convergencia en la mayoría de los casos.

Revisemos por que el método original aplicado a regiones irregulares no produce buenos resultados. Pensemos que se plantea un problema de ajuste sobre una región irregular (como suele suceder cuando los nodos de frontera no son fijos o no son incluidos como partículas) y que se utiliza el método original en busca de convergencia.

Si el ejemplo de región irregular es suficientemente representativo, caeremos en cuenta de que el método se cicla. Esto se debe a que el criterio empleado para determinar si un elemento de la triangulación de Delaunay  $T^*(S_{[k]})$  es también elemento de la malla no estructurada  $T^*|_{\Omega}(S_{[k]})^3$ , no parece responder favorablemente en esas condiciones.

En caso de regiones irregulares el representante (baricentro) de un elemento que se encuentra suficientemente cerca de la franja de frontera puede cambiar su posición entre el interior y la frontera de la región, lo cual provoca que un conector sea visible en algunas ocasiones mientras que en el resto no formará parte de la estructura.

Así que para evitar el problema se ha propuesto dar el tratamiento previo a la región manteniendo los nodos de frontera como nodos fijos.

Aun cuando se ha logrado alcanzar la convergencia del método en la mayoría, existen ejemplos para los cuales aun no se encuentra garantizada, otro caso de ciclado hemos observado. Cuando la malla en equilibrio  $T^{\star}(\Omega, l^*) = \lim_{n \rightarrow \infty} T^*|_{\Omega}(S_{[n]})$  tiene nodos que son suficientemente parecidos a los nodos de una configuración singular de cuatro nodos (en el sentido de Delaunay<sup>4</sup>).

---

<sup>3</sup>vea la sección 2.2.2

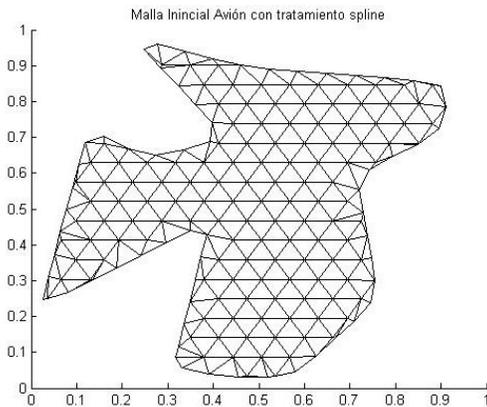
<sup>4</sup>vea la sección 2.1

En estos casos en que el método no converge se sugiere agregar algún nodo para romper la cercanía con la configuración de singularidad.

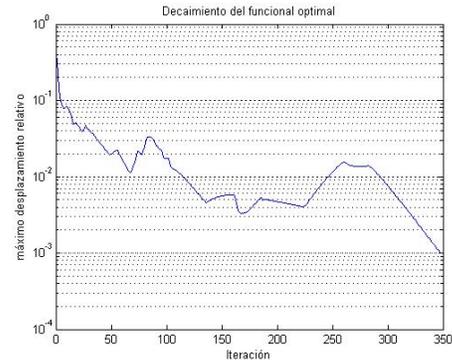
### 3.6. Ejemplos

Considerando que en las secciones anteriores de este capítulo hemos dado tratamiento a la región Avión y Bahía de la Habana retomaremos los ejemplos, tomaremos una la malla inicial y aplicaremos el método de vigas en equilibrio para generar una malla óptima, mostraremos la gráfica de decaimiento de la función optimal para ver como fue la evolución del método.

**Avión con tratamiento Spline** con 60 nodos de frontera.



(a) Malla inicial, Avión con tratamiento spline



(b) Historial del funcional optimal

Figura 3.8: Punto de partida y evolución del generador, Avión.

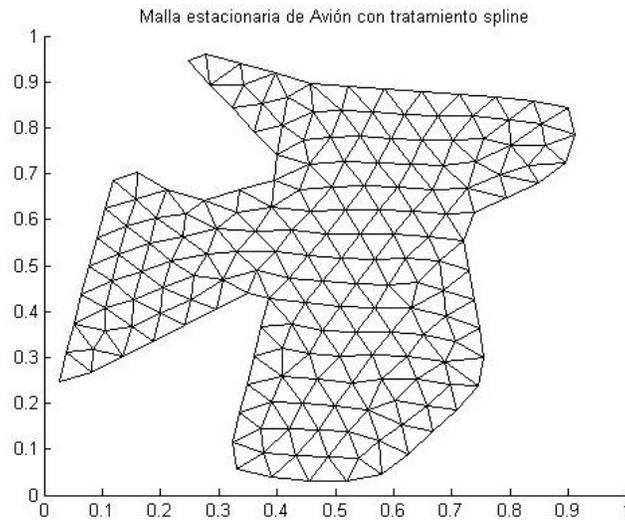
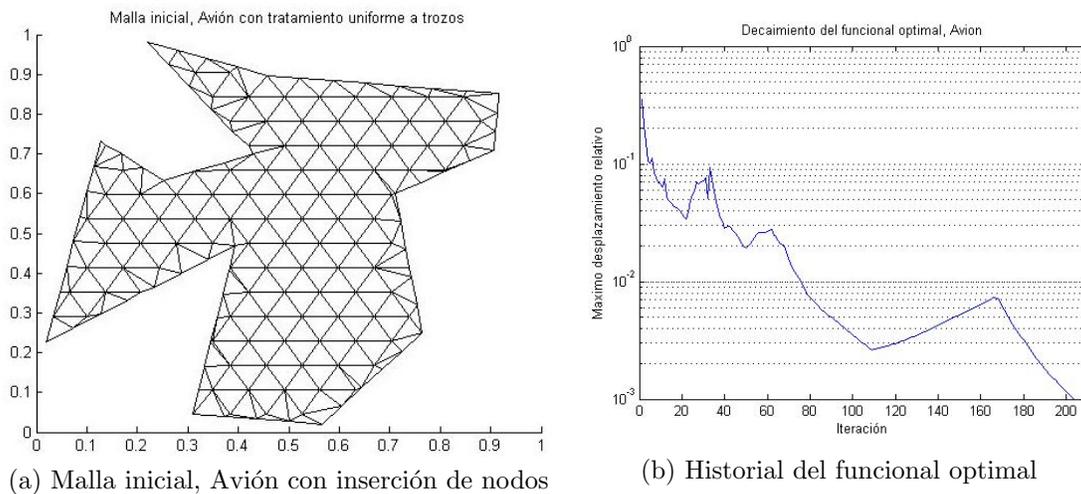


Figura 3.9: Malla óptima de Avión con tratamiento spline.

**Avión con tratamiento por adhesión de nodos con 65 nodos de frontera.** En este caso hubo una fuga de cuatro nodos como se muestra en la gráfica 3.10(a).



(a) Malla inicial, Avión con inserción de nodos

(b) Historial del funcional optimal

Figura 3.10: Punto de partida y evolución del generador, Avión.

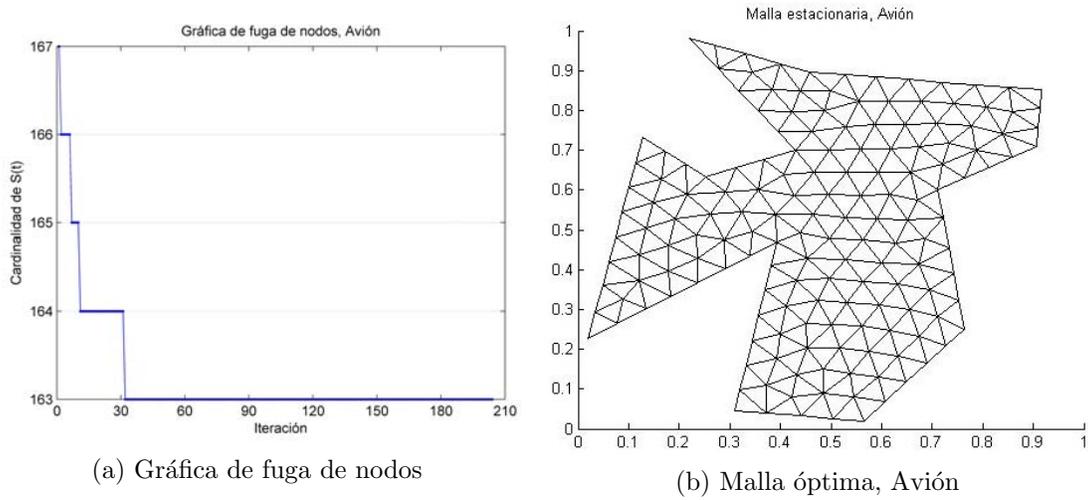


Figura 3.11: Malla de calidad, Avión.

Al contorno de la **Bahía de la Habana** se le ha dado un tratamiento spline con 79 nodos de frontera posterior a la reducción de nodos. En este caso parecería que entre las iteraciones 200 y 400 el método se había ciclado, sin embargo se logró encontrar su equilibrio.

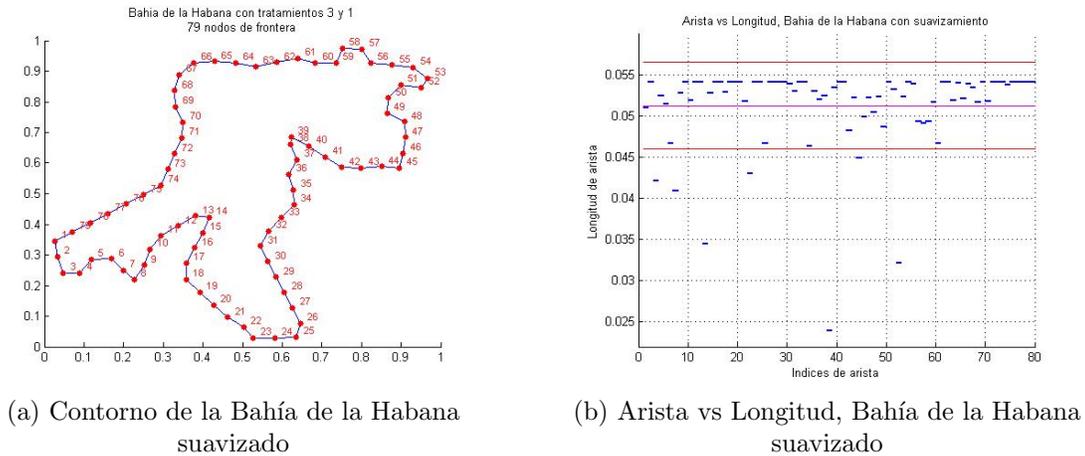


Figura 3.12: Tratamiento de contorno por suavizamiento, Bahía de la Habana.

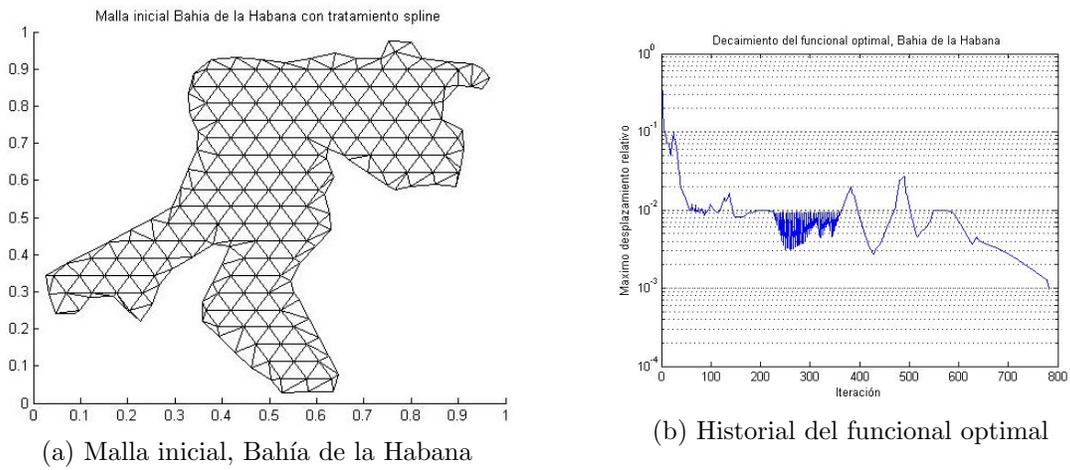


Figura 3.13: Punto de partida y evolución del generador, Bahía de la Habana.

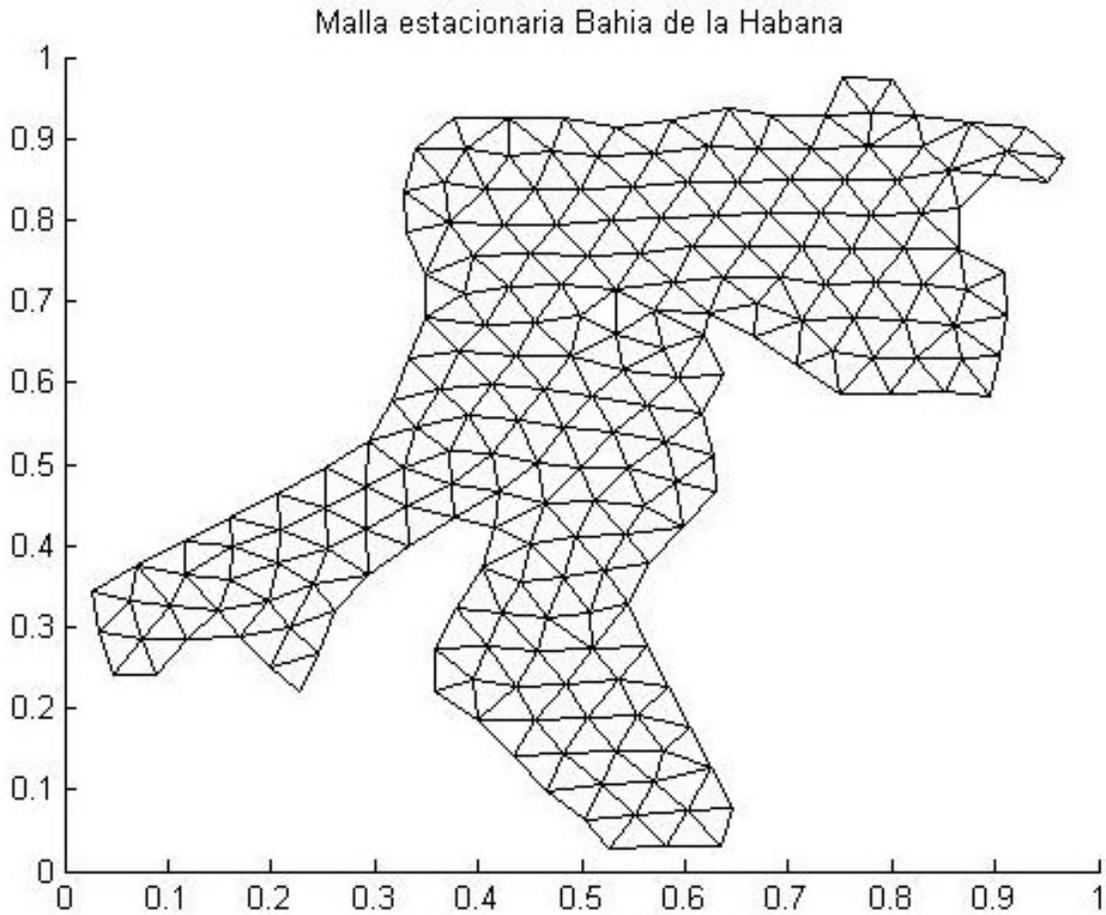


Figura 3.14: Malla de calidad, Bahía de la Habana.



# Capítulo 4

## Unamesh: Sistema para la generación de mallas de calidad

Como se mencionó desde el prefacio, para complementar el trabajo de tesis se ha creado el sistema Unamesh, es decir, un sistema experimental que implementa el método de la ecuación de equilibrio para la generación numérica de mallas no estructuradas en regiones irregulares planas. En este capítulo daremos una explicación básica para el buen uso del sistema.

### 4.1. Diseño e interfase de Usuario

#### 4.1.1. Sobre el diseño

El sistema Unamesh es un paquete computacional, que resuelve de manera eficiente el problema de generar mallas triangulares sobre regiones irregulares planas. Este tipo de mallas es comúnmente utilizado en la solución numérica de ecuaciones diferenciales parciales por el método de elemento finito.

El sistema es totalmente gráfico y cuenta con una interfase gráfica que lo hace amigable al usuario y muy sencillo de usar. El sistema se ha diseñado para que sea semi automático bajo la idea de que con muy pocas opciones de tecla sea posible obtener una malla óptima sobre la región. El sistema ha sido diseñado como un ambiente de ventanas ejecutable sobre MATLAB 7.

El sistema permite trabajar a partir de contornos de regiones planas para obtener sobre ellas una malla de calidad, o bien a partir de una malla triangular obtenida previamente por otros procedimientos y aplicar sobre ella la técnica de vigas en equilibrio implantada en el sistema. Algunos autores se refieren a este enfoque de triangulación como un enfoque de suavizamiento o post-procesamiento de la malla.

El sistema consta de seis Menús de aplicaciones<sup>1</sup>: **Main**, **Contour**, **Triangulation**, **Information**, **View** y **Help**, el uso de cada menú y sus opciones o submenús estarán disponibles siempre que el proceso realizado lo permita. Por ejemplo, si se cuenta con una malla en memoria podremos operar con la opción de suavizamiento, y mientras no se realice este post-procesamiento no estará disponible la información referente a los resultados del método aplicado, En las secciones siguientes haremos una descripción de los módulos **Cotour**, **Triangulation** e **Information** y describiremos cada una de las opciones del menú, cómo y cuándo es posible interactuar con cada una de ellas.

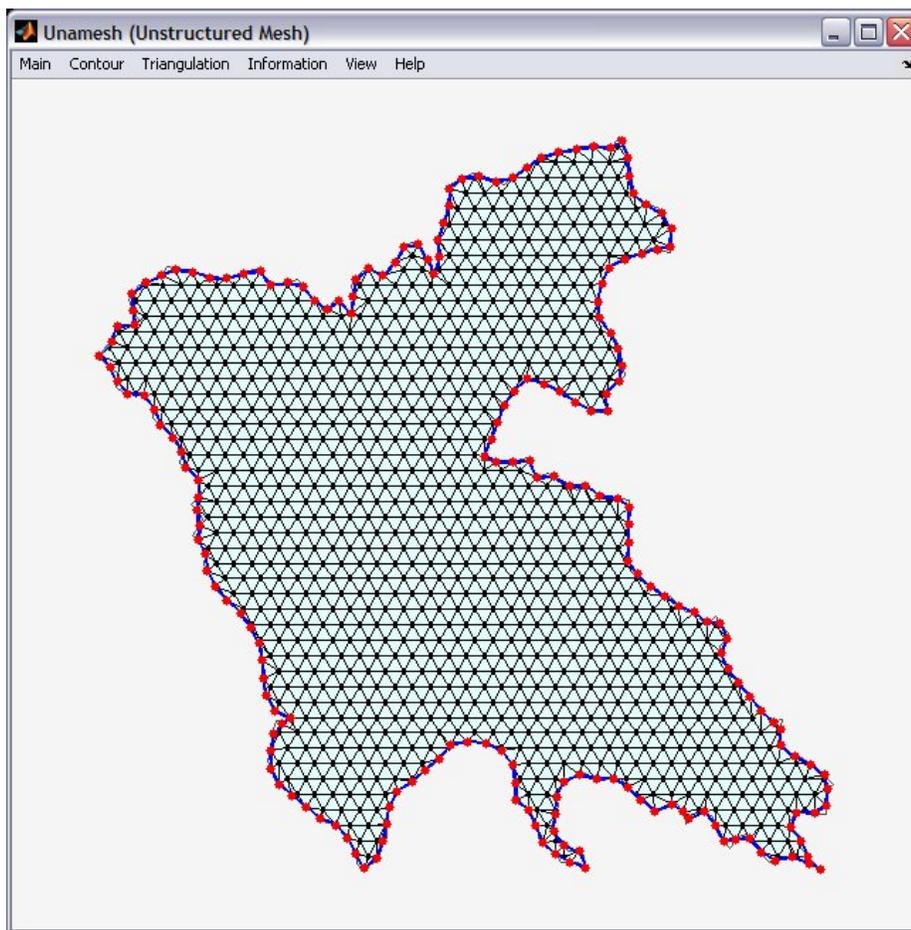


Figura 4.1: Ventana principal de Unamesh, con una malla inicial en pantalla.

---

<sup>1</sup>Las opciones del sistema están escritas en inglés por lo que nos referiremos a las opciones que ha de considerar el usuario tal y como aparecen en pantalla e indicando las palabras clave en tipografía de máquina de escribir.

### 4.1.2. Interfase con el usuario

La interfase con el usuario se encuentra en la barra superior de la ventana de despliegue gráfico y para algunos módulos será desplegado algún sub-panel según la opción elegida. Los paneles consisten en un conjunto de botones, campos de entrada, controles deslizadores y cajas de dialogo.

De acuerdo con el diseño del sistema, el panel de control principal va cediendo el control a otros paneles de acuerdo al trabajo que se desea realizar. Los paneles posteriores son activados para su uso pulsando el área que los describe en el panel principal.

El usuario interactúa con el sistema a través de posicionarse y elegir la opción por golpes *poiting/clicking* de los botones del mouse así como por secuencias de acceso rápido por teclado **ALT+<key>** mientras se encuentra el panel de control activo.

A continuación listamos una serie de operaciones que podemos realizar sobre los paneles.

- Para pulsar un botón u opción del panel, mueva el cursor sobre el botón y pulse uno de los botones del mouse. Puede hacer uso de la secuencia de teclas **ALT+<key>** (donde **key** es la tecla subrayada en el botón) para ejecutar la misma acción.
- Para cambiar el valor de un campo de entrada, coloque el cursor sobre el campo de entrada y pulse alguno de los botones del mouse, o bien haga uso de la tecla **TAB**, tabulador, para obtener la misma operación, e introduzca el valor deseado.
- Para cambiar la posición de un control deslizante, tome el control al posicionar el mouse sobre él y oprimir con el botón izquierdo del mouse sin soltar mientras “barre” el control de un lado a otro. Puede usar los indicadores de inicio y final del control para mover el control de paso en paso.
- Para cambiar un **pop-menu** o menú de opciones, posicione con el mouse sobre el botón de la flecha indicadora y pulse uno de los botones del mouse, al hacerlo será desplegada una pequeña ventana de opciones en la que podrá elegir con el mouse al pulsar con el botón izquierdo la opción deseada. Al hacerlo aparecerá en el letrero del menú el título de la opción elegida.
- Cuando el indicador del mouse se encuentra sobre un botón del panel se desplegará una caja de títulos **tip text** indicando un mensaje relativo a la opción donde se encuentra el indicador.

La ventana principal puede ser modificada en tamaño y mientras no se trabaje en alguna acción se pueden realizar acercamientos en la malla con la acción **Zoom On/Off** en el menú **View**.

## 4.2. Estructura modular

El sistema es modular, esto es, integra una colección de tareas que permiten interactuar entre ellas de manera que el usuario pueda experimentar con los elementos del sistema: la construcción del contorno, tratamiento de contorno, la generación de triangulaciones iniciales y la optimización por el método de la ecuación de equilibrio.

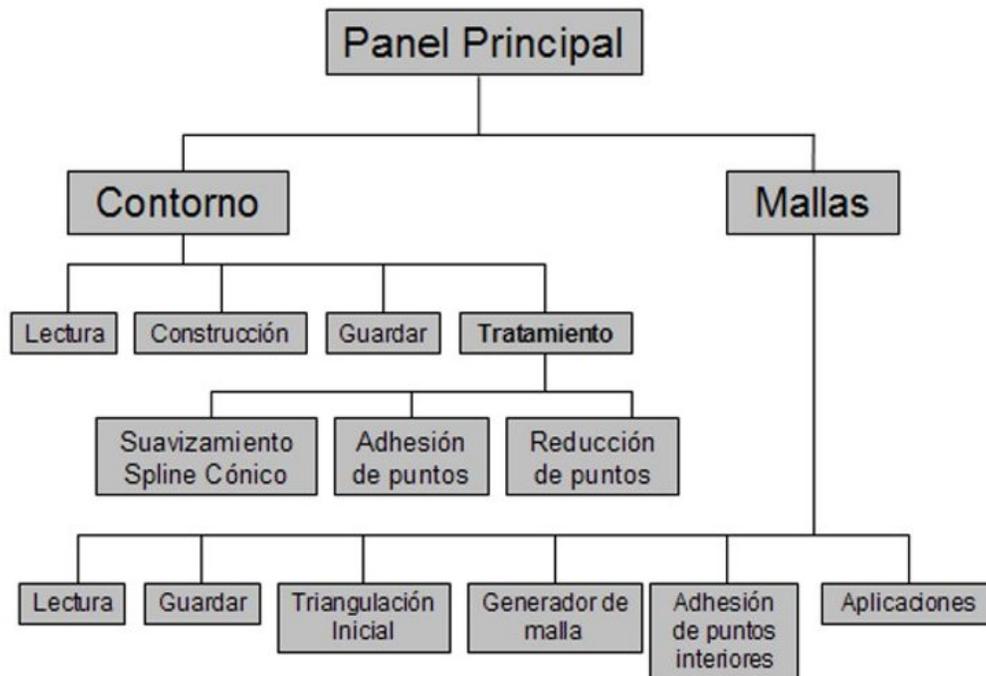


Figura 4.2: Estructura del sistema y dependencia de los módulos.

Como se describe en el diagrama esquemático de los módulos del sistema, las operaciones que pueden realizarse con el sistema son dos: trabajo con contornos y con mallas. Uno puede partir desde la información del contorno de la región previamente obtenida, para operar algún tratamiento y generar una triangulación inicial. De igual manera puede considerar la información de una malla, en archivo o en memoria y sobre ella operar el suavizamiento o realizar nuevas triangulaciones.

En los siguientes apartados describiremos los módulos principales del sistema y daremos la manera de interactuar con ellos.

## 4.3. Módulo de contornos

El conjunto de operaciones que podemos realizar con un contorno son: obtener la información del contorno a partir de un archivo de datos (con el formato convenido), construir un contorno poligonal en pantalla, guardar el contorno en uso y dar tratamiento al contorno ya sea por medio de un suavizamiento spline cónico, o por una adhesión uniforme de nodos o por una reducción de nodos por dos criterios de colinealidad.

El sistema contempla que en cada una de las opciones antes señaladas se cuente con un panel apropiado. En el caso de tratamientos de contorno cada uno cuenta con su propia ventana de visualización y sólo se visualizará el nuevo contorno en la ventana principal hasta ser concluido el tratamiento.

Al solicitar la opción **Contour** del menú principal aparecerá en pantalla el submenú con las opciones descritas.

Las opciones de **Open** y **New Polygon** son independientes de que exista o no algún contorno en memoria de ahí que éstas opciones esén siempre disponibles.

Describamos cada una de las opciones disponibles del módulo **Contour**.

### 4.3.1. Construir un contorno en pantalla. **New Polygon**

Bajo esta opción el usuario puede construir un contorno poligonal para ser usado durante el resto de su trabajo en el sistema. Esta opción abre una ventana de control con un espacio de graficación de dimensiones  $[0, 1] \times [0, 1]$  desde el cual podrá capturar el contorno poligonal a partir de la colocación de vértices pulsando el botón izquierdo del mouse y pulsando el botón derecho del mouse para cerrar el contorno.

Describamos cada una de las opciones integradas en éste panel.

<b>New</b>	Volver a iniciar la captura de un contorno. Con esta opción se eliminará de la memoria cualquier trabajo previo realizado en el área de trabajo. Con ello será posible iniciar una nueva captura de datos.
<b>Remove Point</b>	Con esta opción el usuario podrá deshacerse del último vértice colocado en el panel actual siempre que el contorno no se haya cerrado.
<b>Accept</b>	Una vez obtenido el contorno cerrado esta opción finaliza la captura de los datos y nos regresa el módulo principal. Al pulsar la opción se llevará a memoria los datos del contorno cerrado definido y sobre éste se trabajará a partir de ese momento. Si el contorno no fue orientado positivamente el sistema invertirá el orden para que sea positivo.

- Help** Es una pequeña ayuda, se abre el help de Matlab y desde ahí se asesora en la creación de un contorno.
- Quit** Esta opción cancela todo trabajo hecho con este módulo y nos lleva de vuelta al módulo principal. Al salir no se conservarán los datos de un contorno previo en memoria ya que fueron borrados al optar por la opción New Polygon.

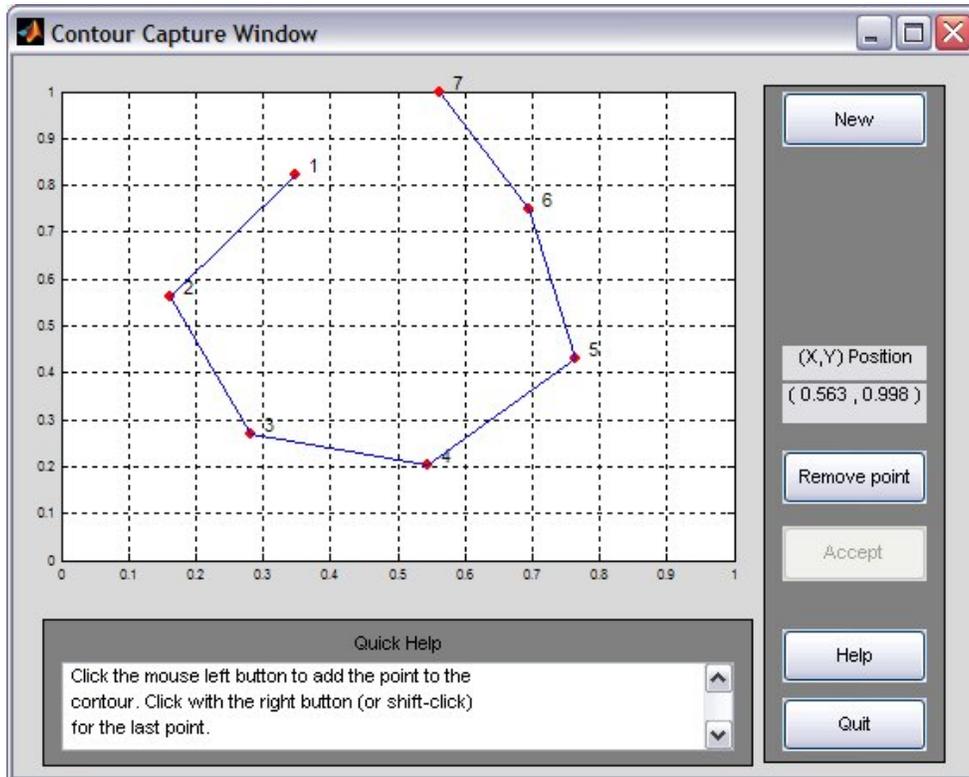


Figura 4.3: Ventana de captura de contorno.

Dentro de este panel puede observarse una caja de diálogo. En ella se desplegará una serie de breves señalamientos sobre el estado que guardan las acciones llevadas a cabo, sea que no se haya cerrado el contorno, que hagan falta puntos para ello, que exista un contorno en memoria o bien, que la ejecución haya sido satisfactoria y se cuente con el contorno en memoria.

De igual manera se observa una caja donde se despliega la posición del mouse dentro del área de trabajo, esto con la finalidad de conocer la posición del mouse y del punto a elegir del contorno dentro del sistema coordenado.

### 4.3.2. Abrir un archivo de datos Open

La información de un contorno puede estar contenida en un archivo de documentos tipo ASCII en formato convenido y podemos tomar esta información para trabajar con ella a lo largo del sistema.

Estos archivos tienen una extensión de archivo `.con` y a través de un **browser** podemos tomar su información desde algún directorio. Al pulsar la opción **Open** del menú de contornos será desplegada la ventana desde la cual podrá desplazarse por el sistema de archivos hasta localizar el archivo de contornos deseado. Esta caja de diálogo es la que se conoce como **browser** y consta de botones propios para aceptar el archivo elegido, reescanear el directorio, y presentar información de la localización del mismo.

Cada vez que el archivo sea elegido los datos pasarán a la memoria del sistema, es decir será posible trabajar en la región y esta será graficada en la ventana principal.

### 4.3.3. Guardar información Save

Una vez que el sistema cuenta con un contorno en memoria es posible almacenar su información en un nuevo archivo o reemplazar un archivo ya existente. En cualquier momento se puede salvar el contorno aún cuando se cuente con una malla en memoria. Con ello queremos decir que la información de un contorno y de una malla son independientes.

Estando en el módulo contorno podemos pulsar la opción **Save** con lo cual aparecerá en pantalla un **browser** bajo el cual podemos indicar la ruta por búsqueda de directorios donde se hospedarán los archivos de datos del contorno. En la línea de entrada deberemos escribir el nombre del archivo que contendrá dichos datos. Si en el nombre del archivo omitimos la extensión de archivo `.con` el sistema lo integra automáticamente.

El **browser**, al igual que en la opción **Open**, permanece en primer plano mientras que el panel principal se encuentra inactivo hasta que la operación haya sido concluida. La información del contorno en memoria es guardada en el formato convenido `.con`. En la sección correspondiente a los formatos de archivos se detalla dicho formato.

### 4.3.4. Tratamiento por suavizamiento spline cónico

El módulo de tratamiento spline cónico se encuentra disponible una vez que se tiene cargado un contorno.

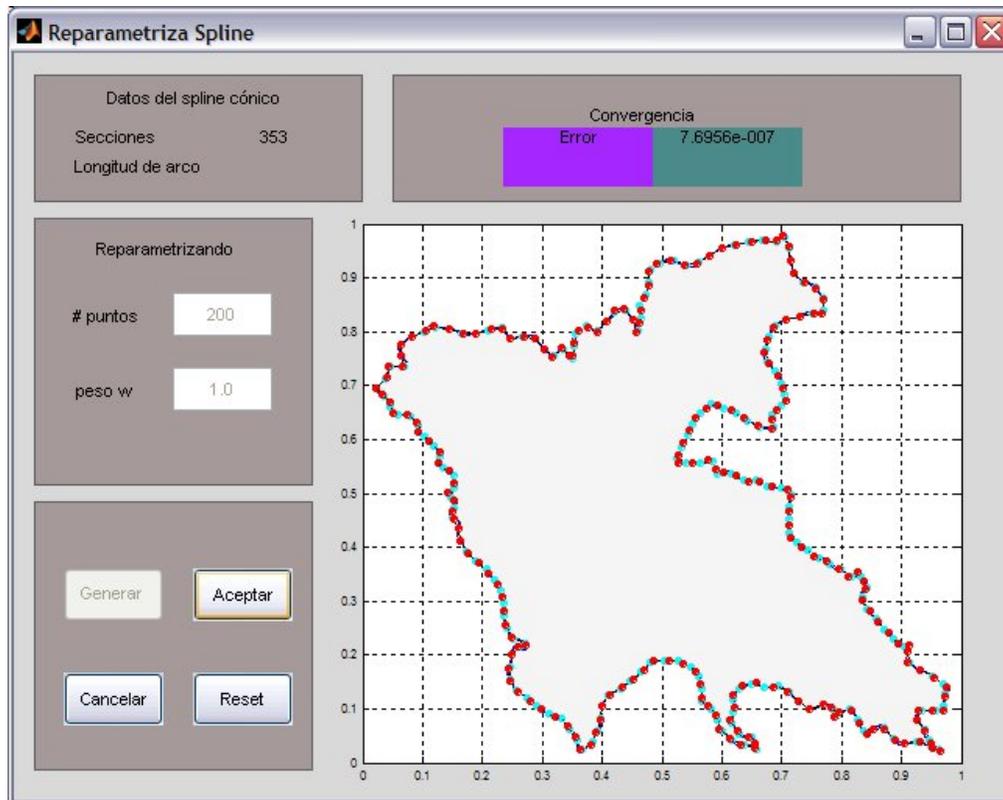


Figura 4.4: Ventana de tratamiento spline cónico.

#### Parámetros.

**# puntos** Cantidad de puntos deseable para construir la poligonal suavizada. Recuerde que el sistema carga el nodo final y el nodo inicial, por lo que a su cuenta debe añadir una unidad.

**peso w** Valor de ajuste a la región, el parámetro por default es = 1.0, para mayor ajuste utilice  $0 < w < 1$  y para mayor ajuste utilice  $1 < w$ .

**Botones.**

Generar	Una vez establecidos los parámetros con este botón se podrá iniciar el proceso iterativo de suavizamiento, es cada paso del método conoceremos el error obtenido por el método de Newton al aproximar la inversa de la longitud de arco, [9]. El método converge hasta que el error sea menor que $10^{-5}$ y en ese momento se activan los botones posteriores.
Aceptar	Este botón permite escoger el contorno construido para trabajar en la triangulación, se cerrará la ventana del tratamiento spline y con ello se cederá el control al panel principal.
Cancelar	Esta opción cancela todo trabajo hecho con este módulo y nos lleva de vuelta al módulo principal. Al salir no se conservarán los datos del tratamiento.
Reset	Esta opción rechaza el tratamiento dado al contorno, y permite cambiar los parámetros para realizar un nuevo tratamiento.

**4.3.5. Tratamiento por adhesión de puntos**

El módulo de tratamiento por adhesión de puntos es activado una vez que se tiene cargado un contorno, y puede ser usado en cualquier momento que se desee pues se mantiene activo a partir de ese momento.

**Parámetros** A escoger uno del menú de opciones.

$h_{max}$	Máximo sobre las longitudes de conector del contorno con tratamiento, se mide en términos del perímetro de la región. El parámetro default indica que cada conector mide menos del 10% del perímetro de $\Omega$ .
$N_1$	Número de puntos a insertar en el segmento más corto del contorno original, la cantidad de nodos adheridos en el resto de los segmentos se determina por medio del parámetro. Si se usa el parámetro por default no se agregará ningún nodo en el segmento más corto.

**Botones.**

- Apply** Una vez establecido el parámetro con este botón realiza la adhesión de puntos, se activan los botones posteriores.
- Aceptar** Este botón permite escoger el contorno construido para trabajar en la triangulación, se cerrará la ventana de tratamiento por adhesión de puntos y con ello se cederá el control al panel principal.
- Cancel** Esta opción cancela todo trabajo hecho con este módulo y nos lleva de vuelta al módulo principal. Al salir no se conservarán los datos del tratamiento.
- Reset** Esta opción rechaza el tratamiento dado al contorno, y permite cambiar los parámetros para realizar un nuevo tratamiento.

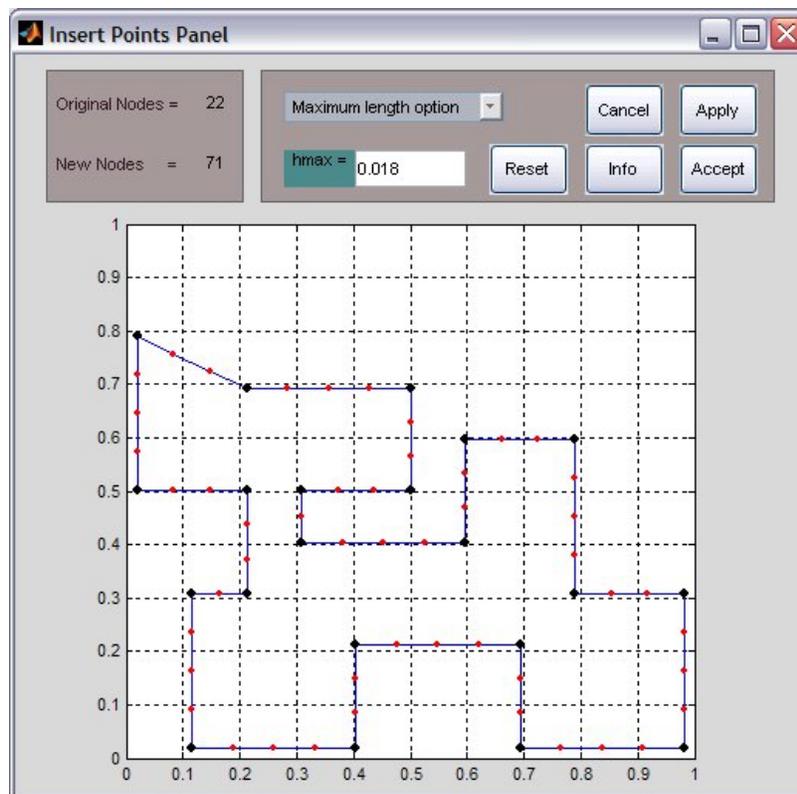


Figura 4.5: Ventana de tratamiento por adhesión de puntos.

### 4.3.6. Tratamiento por reducción de puntos

El módulo de tratamiento por adhesión de puntos es activado una vez que se tiene cargado un contorno, y se mantiene activo a partir de ese momento.

**Controles** Campos y barras deslizables.

- N Número de puntos a tomar para aplicar los criterios de colinealidad.
- R0 Parámetro 1 del criterio de longitud, para la manipulación de los parámetros utilice las barras deslizables.
- R1 Parámetro 2 del criterio de longitud, recuerde que  $1 < R0 < R1$ .
- E0 Parámetro 1 del criterio de error máximo.
- E1 Parámetro 2 del criterio de error máximo, recuerde que  $1 < R0 < R1$ .

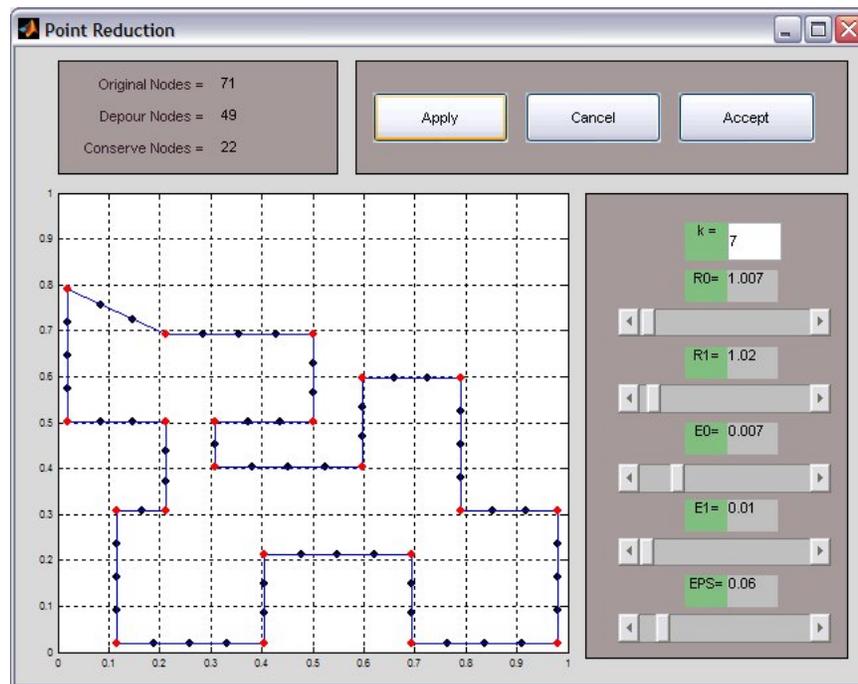


Figura 4.6: Ventana de tratamiento por reducción de puntos.

**Botones.**

<b>Apply</b>	Una vez establecido el parámetro con este botón realiza la adhesión de puntos, se activan los botones posteriores.
<b>Aceptar</b>	Este botón permite escoger el contorno construido para trabajar en la triangulación, se cerrará la ventana de tratamiento por adhesión de puntos y con ello se cederá el control al panel principal.
<b>Cancel</b>	Esta opción cancela todo trabajo hecho con este módulo y nos lleva de vuelta al módulo principal. Al salir no se conservarán los datos del tratamiento.

## 4.4. Módulo de Mallas *Triangulation*

El módulo de mallas ofrece la posibilidad de crear y trabajar una malla o bien llevar a memoria una que se encuentre en archivo de formato convenido. Las tareas que podemos realizar con una malla son: llamar una malla desde un archivo, guardar los resultados obtenidos, crear una triangulación inicial a partir de un contorno, dar tratamiento por método de la ecuación de equilibrio a una triangulación, insertar manualmente nodos en una triangulación y realizar alguna aplicación sobre la triangulación.

Al pulsar la opción *Triangulation* del panel principal aparecerá la barra de opciones de tareas que podemos ejecutar con el sistema.

Si contamos con una malla en memoria podremos ejecutar las opciones de operación con ella. En cada caso los botones relativos a las operaciones deberán ser activados. En el caso de no contar con un contorno en memoria sólo estará activada la opción de abrir triangulación.

Describamos las operaciones que podemos obtener con éste módulo.

### 4.4.1. Llamar desde un archivo a una triangulación *Open*

Contando o no con una malla en memoria esta opción siempre estará disponible. Con ello podemos llevar a memoria una malla en archivo, bajo el formato convenido.

Al pulsar la opción *Open* aparecerá en pantalla en primer plano un **browser** desde el cual podremos indicar el archivo a leer de entre la lista que aparecerá en el espacio destinado a tal efecto. De igual manera podremos movernos por el sistema de archivos hasta localizar el archivo que contiene los datos de la triangulación deseada.

Los archivos de datos cuentan con una estructura muy simple y se encuentran en formato `.tri`. El sistema no cuenta con un análisis de datos de los archivos previo a su lectura, lo único que verifica es la cantidad de nodos y de elementos y pasa a leer los datos tipeados por esas cantidades. Si existen problemas en la lectura de archivos, que no se pueda abrir el documento la operación será cancelada.

Al leer satisfactoriamente los datos en la ventana principal se dibujará la malla indicando en color cyan los segmentos de frontera, en rojo los nodos de frontera, en verde los conectores interiores que conforman los elementos y en color el fondo de la región.

Si existiera un contorno en memoria con ésta opción será eliminado. Sin embargo contaremos con el contorno de la región de la malla así leída. Esta opción es transparente al usuario.

#### 4.4.2. Guardar Triangulación Save

Esta opción del módulo malla nos permite guardar los datos de la malla en curso en un archivo de datos.

Al situarnos en la opción **Save** del panel principal se despliega un menú con dos opciones de formato para salvar la triangulación: **Unamesh format** y **Triangle format**. La opción de formato **Unamesh** es con la cual trabaja el sistema, sin embargo se ofrece el formato **Triangle** por si el usuario de **Unamesh** también es usuario de **Triangle**. Si se escoge cualquiera de las dos opciones aparecerá en primer plano de pantalla un **browser** con el cual podemos indicar el directorio donde deseamos depositar dicha información.

En la parte superior del **browser** se indicará el directorio en uso y en el cuerpo del **browser** observaremos un listado de archivos que ahí se encuentran. Al pulsar la opción **Save** los datos serán depositados en el disco y el control del sistema volverá al panel principal. Pulsando la opción **Cancel** anularemos la operación de guardar información de igual forma retornaremos el control al panel principal.

#### 4.4.3. Triangulación Inicial

Este módulo podrá usarse siempre que se cuente con un contorno aun en caso de no haber recibido tratamiento. Para poder usarlo se recomienda que no haya muchos nodos de grado mayor que 7.

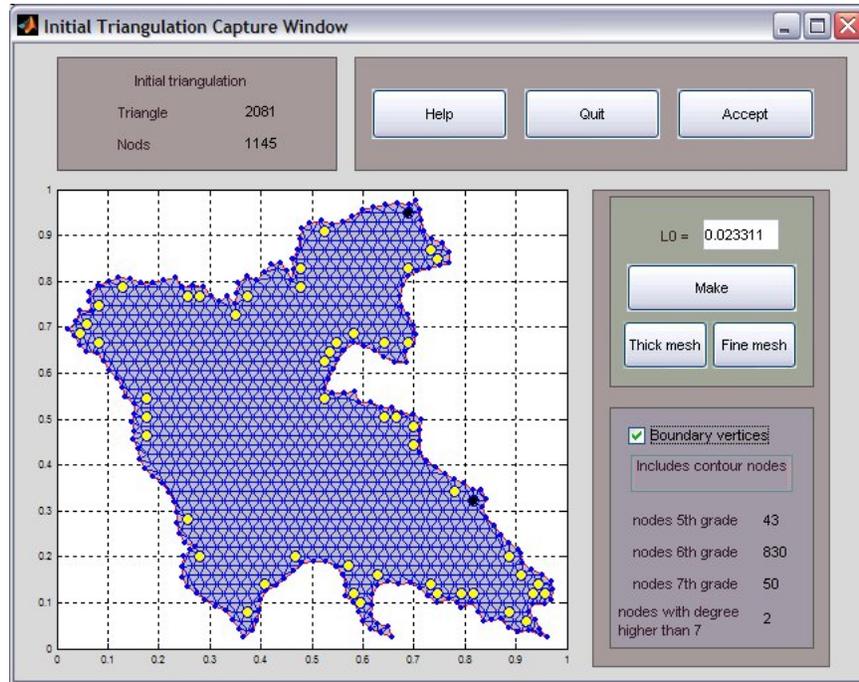


Figura 4.7: Ventana del módulo de triangulación inicial.

### Parámetros.

$L_0$  Parámetro de la rejilla regular, este valor puede ser escrito directamente en el campo bajo su nombre pulsando el botón **Make** o bien puede aumentarse o disminuirse usando los botones **Thick mesh** y **Fine mesh**.

**Check box.** Si la opción no es activada no se contará con los vértices de frontera en la triangulación, y se da paso al problema de ajuste.

### Botones.

**Aceptar** Este botón permite escoger la triangulación inicial construida para trabajar el tratamiento posterior, se cerrará la ventana en uso y con ello se cederá el control al panel principal.

**Quit** Esta opción cancela todo trabajo hecho con este módulo y nos lleva de vuelta al módulo principal. Al salir no se cargará ninguna triangulación realizada dentro del módulo.

**Help** Esta opción ofrece la ayuda útil para la generación de la triangulación inicial.

#### 4.4.4. Generador de Malla

En este módulo se lleva a cabo la integración numérica del método de la ecuación de equilibrio.

##### Panel de parámetros

Lo primero que se deberá hacer para aplicar el método es fijar los parámetros con los que funcionará el método, para iniciar el tratamiento de la malla deberá pulsar el botón **Start**.

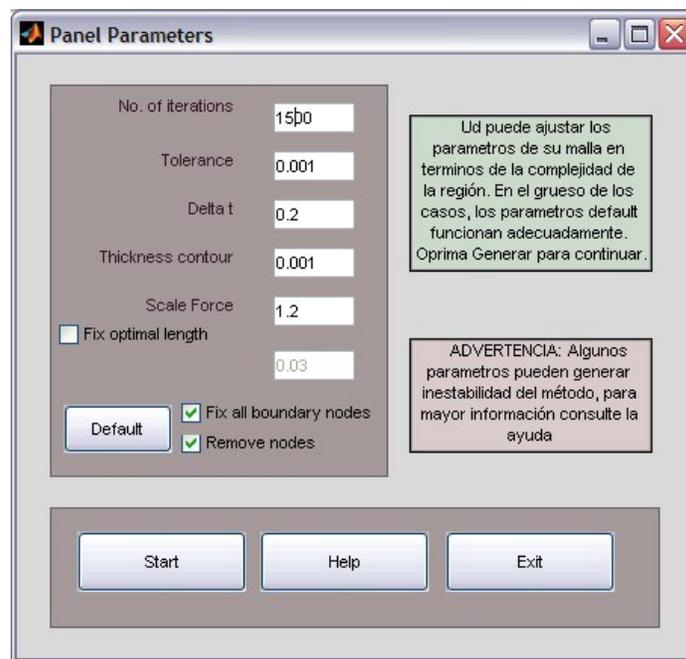


Figura 4.8: Ventana de parámetros módulo generador de malla.

##### Parámetros.

No. Iter	Criterio de paro simple, número de iteraciones máximo para la aplicación del método, se realizarán todas las iteraciones si no se minimiza la función optimal.
Tolerance	Criterio de optimalidad, el sistema se detiene cuando la tolerancia se haya revasado por la funcion optimal.
Delta t	Parámetro de paso del método de Euler.

Thickness contour	Valor para la definición del ancho de banda de frontera.
Scale Force	Escalamiento de la longitud $L^*$ .
Check box 1.	<b>Fix optimal length</b> , al activar esta opción aparecerá un campo en el cual se puede modificar la longitud optima de conector.
Check box 2.	<b>Fix boundary nodes</b> , esta función se haya activa por default, pero si la triangulación recibió tratamiento por adhesión de nodos es posible querer que algunos nodos de frontera puedan cambiar de posición.
Check box 3.	<b>Remove nodes</b> , En caso de no ser activada esta función, todo nodo interior será mantenido en la región con la aplicación del corrector.

### Botones.

Default	Al pulsar el botón los parámetros serán ajustados a los valores por default con los cuales se ofrece mayor facilidad en la convergencia.
Start	Este botón iniciar el método de vigas en equilibrio, se cerrará la ventana de parámetros y con ello se cederá el control al panel de datos del generador.
Exit	Esta opción cancela todo trabajo hecho con este módulo y nos lleva de vuelta al módulo principal.
Help	Esta opción ofrece la ayuda útil para el manejo de los parámetros.

### Panel de datos

Este panel aparecerá una vez que se haya pulsado **Start** en el panel de parámetros y mantendrá el control del sistema hasta que el usuario no pulse **Exit** o cierre la ventana.

#### Botones.

Change parameter	Conserva la malla manejada en la última iteración del método y regresa al panel de parámetros.
Initial Mesh	Se regresa a la malla inicial por si se desean cambiar los parámetros.
Pause/Continue	Para detener el proceso por un lapso de tiempo.

- Continue** Para reanudar el método una vez que se haya alcanzado el número de iteraciones deseado.
- Stop** Para detener el método por decisión del usuario.
- Quit** Si se permite concluir exitosamente el proceso, se cerrará la ventana en uso y con ello se cederá el control al panel principal.



Figura 4.9: Ventana de despliegue de datos del generador.

## 4.5. Entrada y salida de datos

La especificación de los datos que emplea el sistema ayudará a intercambiar la información hacia y desde el sistema. Pasamos a describir el formato de los datos que utiliza el sistema.

### 4.5.1. Formato del archivo de datos de los contornos

Los archivos de un contorno cuentan con un formato que no es originario de UNAMESH, como se busca generar una compatibilidad con el sistema UNAMALLA se ha decidido optar por el formato de archivo `.con`, entiéndase entonces que UNAMESH no hará uso de todos los valores que el formato almacena, los valores marcados con (\*) no serán usados.

Hemos de enfatizar que los contornos son cerrados por lo que la descripción debe considerar que el punto inicial y el punto final serán los mismos; de igual manera para fines prácticos los consideramos con un orden: el sentido contrario al de las manecillas del reloj. Los archivos de datos tienen el formato siguiente:

```

m      ibflag  nb1  nb2  nb3  nb4
x1    y1
x2    y2
x3    y3
⋮      ⋮
xm-1  ym-1
xm    ym

```

Bajo este formato  $(x_m, y_m) = (x_1, y_1)$ . Explicaremos cada una de las entradas.

m Es el número total de nodos de frontera. El primer y último deben repetirse para considerar el contorno cerrado.

ibflag Indicador del tipo de contorno.

= 1. Los segmentos de frontera están definidos.

= 0. No hay segmentos de frontera determinados para el mapeo.

nb1\* Número de nodos que describen el segmento de frontera 1.

nb2\* Número de nodos que describen el segmento de frontera 2.

nb3\* Número de nodos que describen el segmento de frontera 3.

nb4\* Número de nodos que describen el segmento de frontera 4.

$x_i, y_i$  Coordenadas  $x$  y  $y$  del contorno.

### 4.5.2. Formatos de archivo de Mallas

El formato que UNAMESH utiliza para las triangulaciones plantea una distinción entre nodos de frontera y nodos interiores, pues se conserva una orientación en los nodos de frontera, a diferencia con el formato de contorno en este caso no habrá repetición de nodo principio-fin pues no resulta necesario. En la primer línea se presentan los valores enteros de nodos de frontera, nodos interiores y cantidad de triángulos en la malla, posteriormente se cuenta con el listado de nodos a dos columnas empezando con los nodos de frontera y continuando con los nodos interiores, posterior a la posición de los nodos se describen la colección de triángulos dada por el índice de orden de los tres vértices de cada triángulo. En este caso no mostraremos un ejemplo por no ocupar mucho espacio, pero en general el formato seria de la siguiente manera.

<b>nb</b>	<b>ni</b>	<b>nt</b>
$x_1$	$y_1$	
$x_2$	$y_2$	
⋮	⋮	
$x_{nb}$	$y_{nb}$	
$x_{nb+1}$	$y_{nb+1}$	
⋮	⋮	
$x_{nb+ni}$	$y_{nb+ni}$	
$i_{1,1}$	$i_{2,1}$	$i_{3,1}$
$i_{1,2}$	$i_{2,2}$	$i_{3,2}$
⋮	⋮	⋮
$i_{1,nt}$	$i_{2,nt}$	$i_{3,nt}$

El sistema cuenta también con una alternativa de ficheros para ser trabajado con TRIANGLE, este sistema para la generación de mallas lee tres ficheros, uno con la colección de nodos (.node), otro con los índices de conectores de aristas de frontera y el tercero con los índices de los tres vértices de cada elemento de la malla. Vease <http://www.cs.cmu.edu/quake/triangle.html>.



# Capítulo 5

## Conclusiones

### 5.1. Conclusiones

En este trabajo se abordó el problema de la generación de malla casi-equiláteras sobre regiones planas irregulares, los puntos más relevantes son los siguientes:

- La generación de mallas casi-equiláteras por el método de vigas en equilibrio sólo se había desarrollado para regiones planas relativamente simples y con fines de ajuste de calidad.
- Se mostró que es posible usar el método de equilibrio en regiones irregulares planas, siempre que los nodos de frontera se mantengan fijos en cada iteración del método.
- La implantación del método modificado es eficiente con los parámetros establecidos, para la mayoría de las regiones irregulares.
- Se implementó una técnica numérica diferente a la usada por Persson para el cálculo de la función de retención de nodos  $C(X)$ .
- Se implementó una técnica innovadora de depuración de nodos interiores.
- Se ascendió el valor de *longitud de viga promedio*  $l^*$  a nivel de parámetro de entrada para el cálculo de la fuerza total  $F_{total}$  del método.
- Se creó el sistema Unamesh para la implementación del método de vigas en equilibrio de fuerzas en regiones irregulares, con un módulo de refinamiento local.
- Se consiguió mejorar el método incluyendo dos técnicas de tratamiento de contorno previo a la generación de malla.

## 5.2. Trabajo Futuro

Varias líneas de trabajo pueden seguirse de manera directa a partir de los desarrollos que hemos realizado. Específicamente, estamos interesados en:

1. Utilizar las mallas generadas por Unamesh para resolver numéricamente algunas ecuaciones diferenciales parciales.
2. Utilizar las mallas generadas por Unamesh para interpolar datos dispersos en regiones planas acotadas.
3. Implementar un módulo para la generación de mallas sobre regiones irregulares agujeradas.
4. Implementar un módulo para determinar los nodos fijos interiores.
5. Implementar un módulo para el uso local del método.
6. Aplicar nuestros resultados para generar mallas con una densidad deseada en regiones irregulares.
7. Buscar las modificaciones pertinentes para generar mallas adaptivas.
8. Extender el método de equilibrio en regiones irregulares para generar mallas 3D de calidad.

## 5.3. Nota final

El sistema Unamesh para generar mallas casi-equiláteras está disponible para su uso o modificación, este puede ser descargado libremente en la dirección:

<http://www.matematicas.unam.mx/computocientifico/>

# Apéndice A

## Guía rápida de Unameash

Para facilitar el uso del sistema Unameash, en este apéndice se ilustra con un ejemplo como elaborar una malla de buena calidad de una región irregular. Recordemos que el sistema es experimental y nos permitirá generar mallas empleando el método en diferentes modalidades.

Para empezar a usar Unameash, inicie Matlab 7 en su computadora y sitúe el directorio actual (Current Directory) en la carpeta `..\Unameash`, que contiene al sistema. Posteriormente escriba `unameash` en la ventana de comandos (Command Window). Con esto usted debe estar visualizando la ventana principal del sistema, donde en la parte superior verá el menú principal, vea figura A.1.

Para iniciar con el ejemplo, tomaremos la región definida por el contorno `michoacan.con` que usted puede encontrar en la carpeta `...\Unameash\contornos\mapas`. Para cargar este contorno en el sistema use la opción `Open` del submenú `Cotour`.

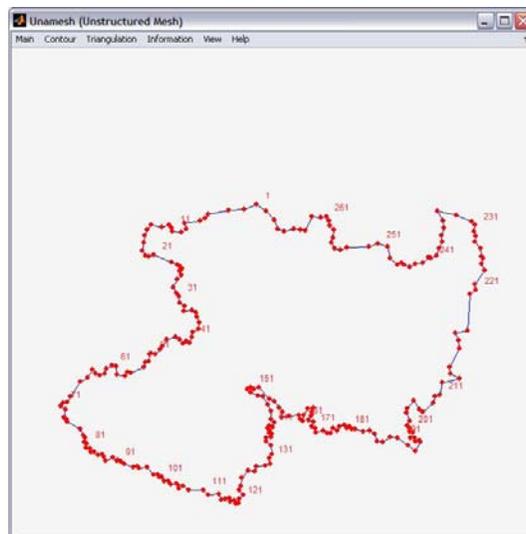


Figura A.1: Ventana principal con el contorno `michoacan.con`

Antes de realizar un ejemplo donde se aplique un tratamiento de contorno previo a la generación de malla, realicemos un primer ensayo sin darle tratamiento a la región. Vaya directamente a la opción **Initial Triangulation** del submenú **Triangulation**, si se realiza adecuadamente usted visualizará la siguiente ventana.

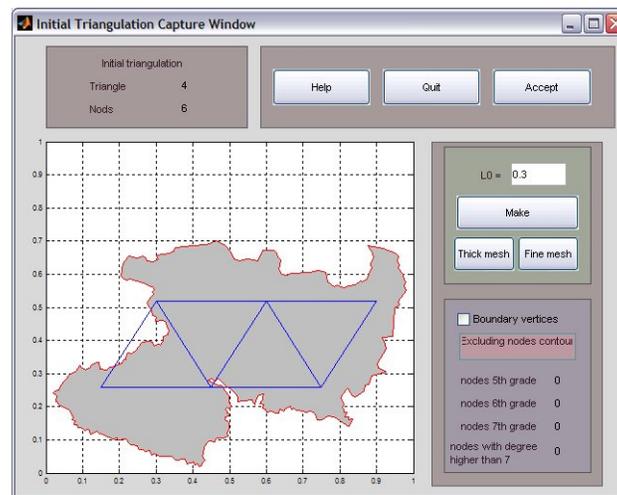


Figura A.2: Panel de Triangulación inicial.

Como se alcanza a apreciar en la figura A.2, la triangulación inicial es muy gruesa para ante la irregularidad de la región, por lo cual es necesario modificar el parámetro  $L0$  que corresponde al valor de longitud de lado de cada triángulo de la rejilla inicial. Reemplace la cantidad del campo por 0.03 y pulse el botón **Make** para aplicar la modificación. Antes de aceptar la triangulación inicial incluya los nodos de frontera con el check box de título **Boundary Vertices**.

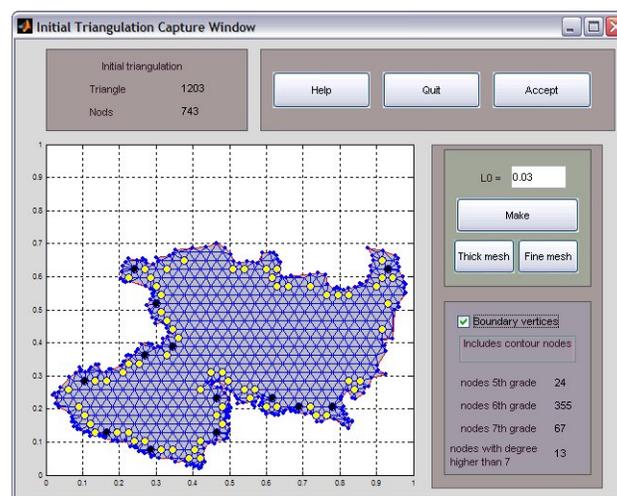


Figura A.3: Triangulación inicial,  $L0 = 0.03$  con nodos de frontera incluidos.

Al momento de aceptar la malla inicial, el subpanel de triangulación inicial se cerrará cediendo el control al panel principal, usted deberá estar viendo el panel principal con la triangulación inicial.

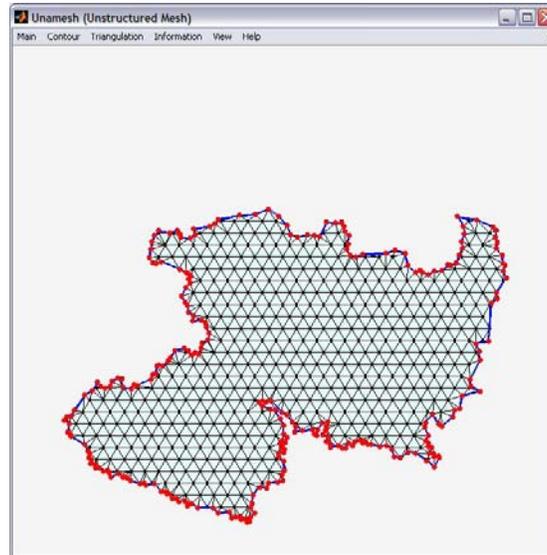


Figura A.4: Panel principal, triangulación inicial en memoria.

Para iniciar el proceso de suavizamiento por el método de vigas en equilibrio, usted tendrá que aplicar la opción **Generation Panel** del submenú **Triangulation**. Primero se tiene la opción de modificar los parámetros del método, en el panel de parámetros, figura A.5. Para efectos del ejemplo únicamente cambiaremos el número de iteraciones por 1500<sup>1</sup>.

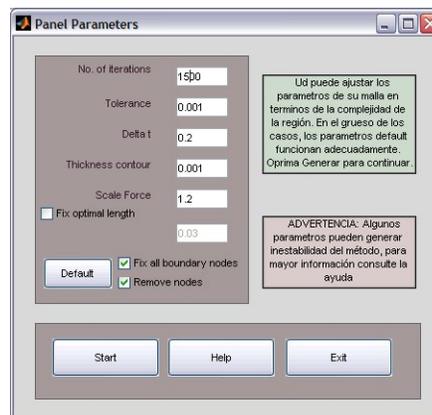


Figura A.5: Panel de parámetros, 1500 iteraciones.

<sup>1</sup>Vea que la opción **Remove nodes** se encuentre activada, puesto que en esta primera corrida se pretende que los nodos interiores sean removidos siempre que se acerquen a la frontera.

Cuando pulse **start** el control del sistema será cedido a la ventana de datos del generador, ahí podrá conocer la información en tiempo real del método. La visualización de la malla durante el suavizamiento será en el panel principal, verá que los nodos interiores que se aproximan al segmento más largo de la frontera desaparecen.

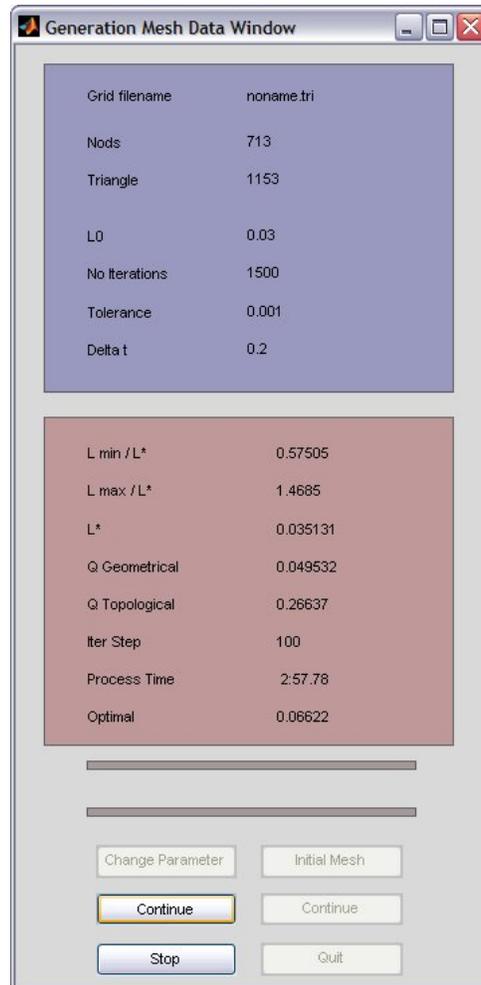


Figura A.6: Ventana de datos del generador, detenida en la iteración 100.

Continuando con la guía rápida, se sugiere pulsar **Stop** en la iteración 99, para abortar la búsqueda del equilibrio, pues el proceso terminaría después de 45 minutos sin hallar convergencia.

Le presentaremos los resultados de las 1500 iteraciones para que usted pueda conocer cuales son, no es necesario que invierta su tiempo en cotejar los resultados; para que usted sienta la seguridad de que se está realizando de forma correcta sólo compare la ventana de datos del generador que (debe estar en la iteración 100) con la ventana que se ilustra en la figura A.6. Posteriormente cierre la ventana de datos pulsando el botón **Quit**.

En esta corrida, el proceso de remoción de nodos interiores continua durante las 1500 iteraciones, en la figura A.7 se ve que a lo largo del método se han depurado más de 50 nodos; este hecho nos dice que la región no está debidamente encerrada por puntos fijos. En la figura A.8 podemos ver que la función a minimizar sufre cambios bruscos y aunque los valores de la función se acercan a la tolerancia en algunas iteraciones no podríamos garantizar que continúe así para futuras iteraciones. También podemos notar en las figuras A.9 y A.10 que la calidad topológica y geométrica de la malla es peor conforme el método avanza.

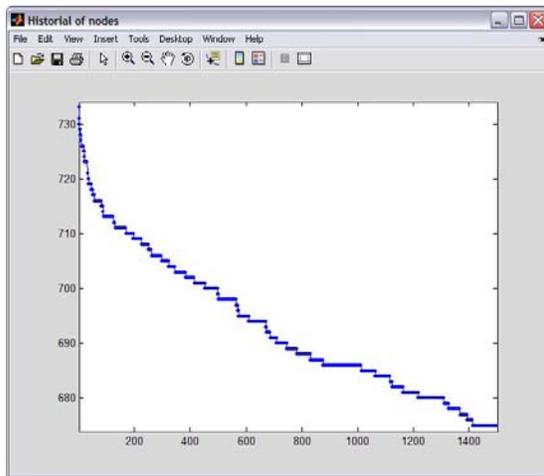


Figura A.7: Cantidad de nodos por iteración.

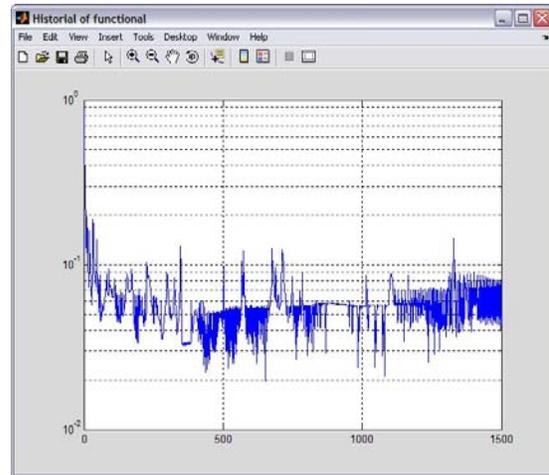


Figura A.8: Valor de la función optimal por iteración.

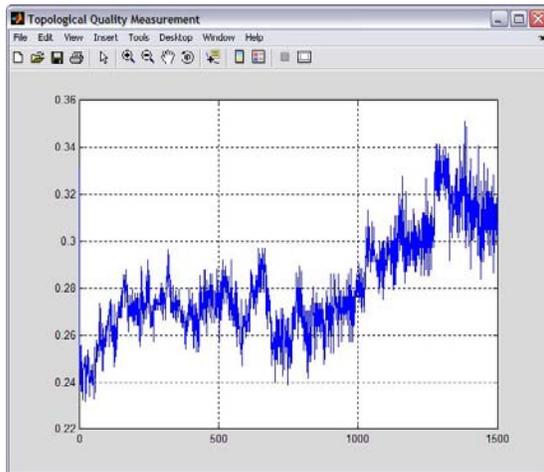


Figura A.9: Medida de calidad topológica por iteración.

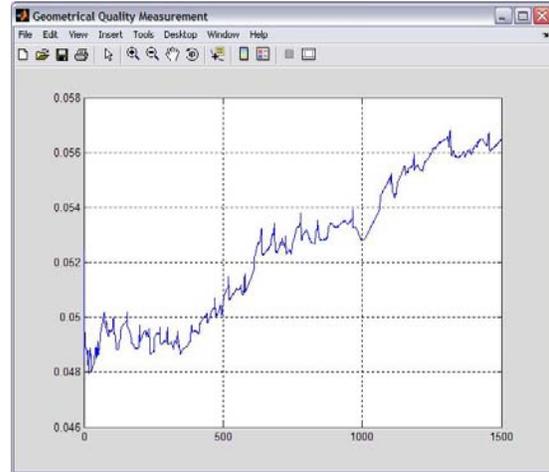


Figura A.10: Medida de calidad geométrica por iteración.

Al cabo de las 1500 iteraciones la malla generada es la que se ilustra en A.11

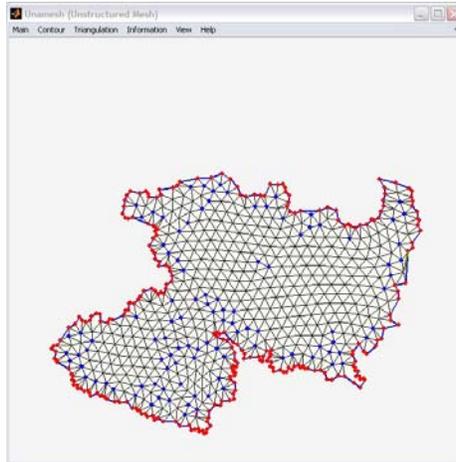


Figura A.11: Malla de Michoacán obtenida después de 1500 iteraciones, contorno sin tratamiento y método con eliminación de nodos.

Ahora hagamos un segundo ensayo, nos preguntamos si el método converge si no se realiza eliminación de nodos interiores. Vuelva a cargar la malla inicial como se indica en la figura A.3 y nuevamente escoja la opción **Generation Panel** del menú **Triangulation**. También aplicaremos 1500 iteraciones, y desactivaremos la opción **Remove nodes**.

El proceso tarda aproximadamente 1 hora con 52 minutos, sin hallar convergencia. Una vez más, no será necesario que espere a que termine el proceso, usted puede frenarlo en el momento que guste (observe que el cambio de conectores es frecuente en el proceso). Al cabo de las 1500 iteraciones los resultados son los siguientes:

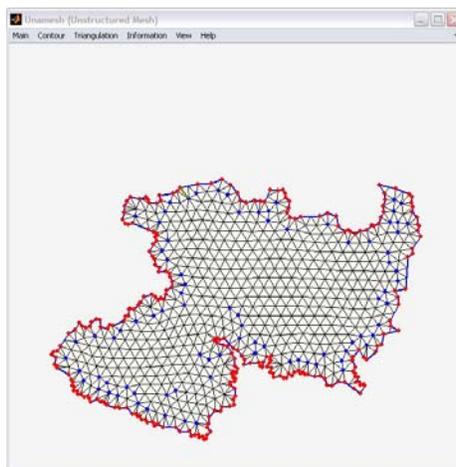


Figura A.12: Malla de Michoacán obtenida en la iteración 1500, sin eliminación de nodos.

De la figura A.13 podemos deducir que no basta con que el método no realice eliminación de nodos interiores para que la función a minimizar no sufra cambios bruscos. También con la figura A.14 podemos notar que frecuentemente hay cambio de conectores, mientras que en la figura A.10 vemos que la medida de calidad geométrica de la malla no cambia notablemente de una iteración a otra, (no mejora ni empeora).

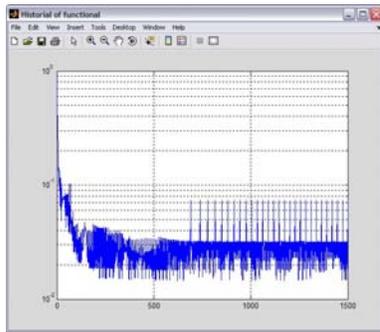


Figura A.13: Función optimal por iteración.

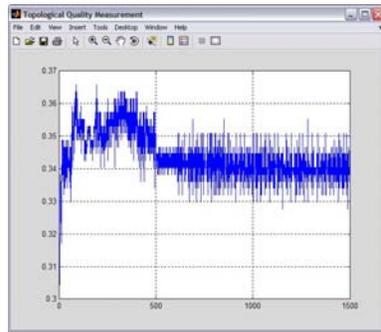


Figura A.14: Calidad topológica por iteración.

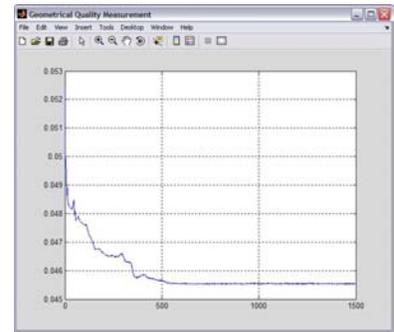


Figura A.15: Calidad geométrica por iteración.

Una vez que hemos ilustrado que la convergencia no se presenta si no hay un tratamiento previo al contorno, procedamos a realizar el ejemplo completo del método. Como contamos con el contorno de michoacán en memoria, las opciones de tratamiento de contorno se encuentran disponibles, aplique la opción de **Conic Spline** que puede encontrar en el submenú **Contour\Treatment**. Coloque la cantidad 136 en el campo **# puntos** y posteriormente pulse el botón generar. Cuando el método haya terminado usted debe estar viendo la ventana que se ilustra en la figura A.16.

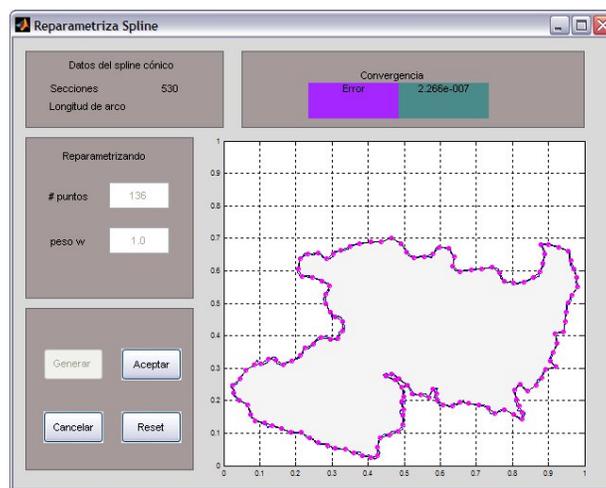


Figura A.16: Ventana de Tratamiento Spline Cónico, contorno de michoacán con 136 puntos.

Acepte el tratamiento que se le ha dado al contorno y proceda a generar la triangulación inicial, utilice el valor  $L0.03$  y no olvide incluir los vértices del contorno activando el check box de nombre **Boundary Vertices**. La ventana de malla inicial en dichas condiciones se visualizará como la ventana de la figura A.17.

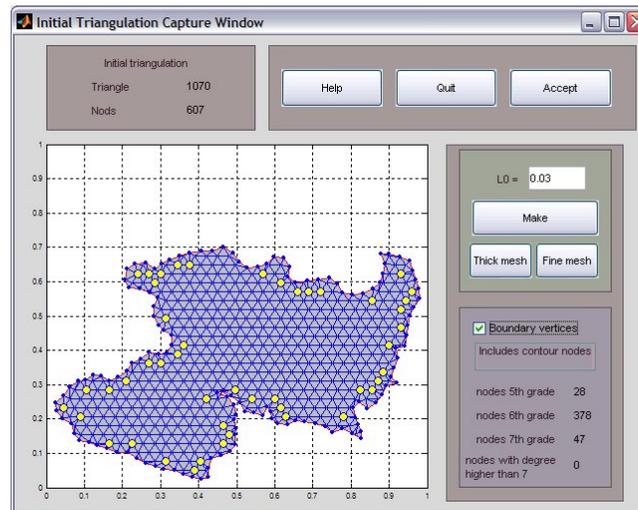


Figura A.17: Triangulación inicial,  $L0 = 0.03$  con nodos de frontera incluidos.

Pulse el botón **Accept** para cargar la malla inicial en memoria, con esta acción se visualizará la malla inicial en la ventana principal, figura A.18.

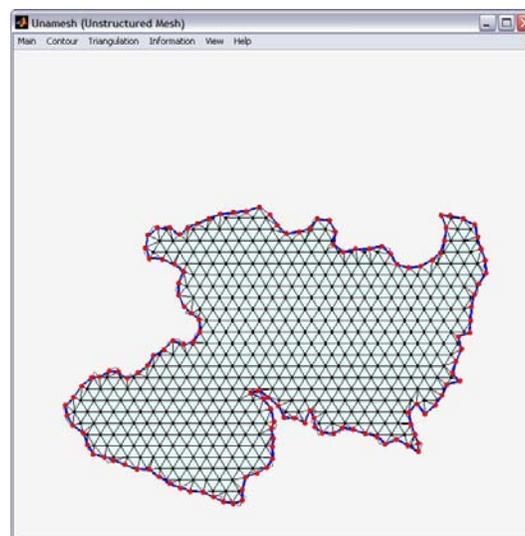


Figura A.18: Ventana principal con la triangulación inicial.

Para iniciar el proceso de suavizamiento, aplique la opción **Generation Panel** del submenú **triangulation**. Antes de iniciar el proceso de cómputo cambie, en el Panel de parámetros<sup>2</sup>, el número de iteraciones por 1500, figura A.5.

En esta ocasión el método deberá tener éxito, el proceso terminará antes de concluir con iteraciones estipuladas (exactamente en la iteración 625), después de 14 minutos de haber iniciado, como se ilustra en figura A.19.

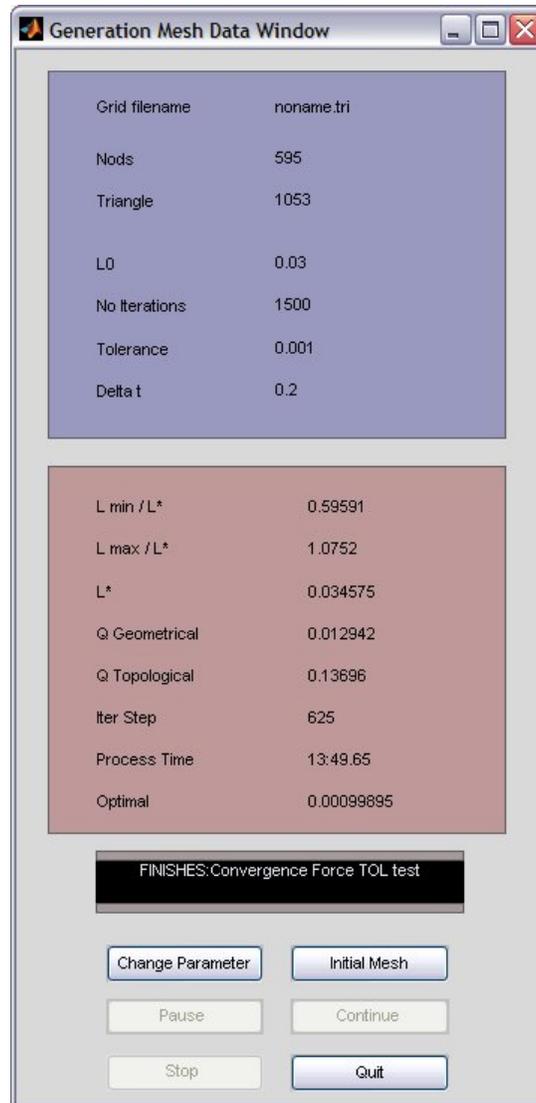


Figura A.19: Ventana de datos del generador, al finalizar exitosamente el proceso.

<sup>2</sup>Vea que la opción **Remove nodes** se encuentre activada, ya que pretendemos emplear el método completo.

Pulse **Quit** en el panel de datos para ceder el control al panel principal. La malla óptima se visualiza en la ventana principal, figura A.20

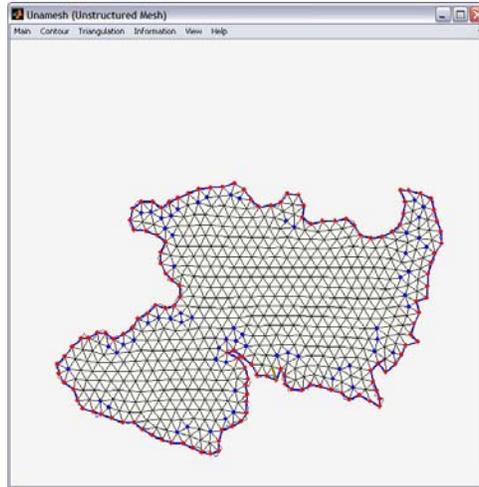


Figura A.20: Ventana principal con malla óptima

La información que está disponible en menú **Information\Generator** muestra que en efecto hay convergencia en este ejemplo. Como se muestra en la figura A.21 la fuga de nodos solo se presenta en las primeras iteraciones. En la figura A.22 se puede ver que no hay cambios bruscos en el valor del funcional, y que el método alcanza el criterio de paro en la iteración 625.

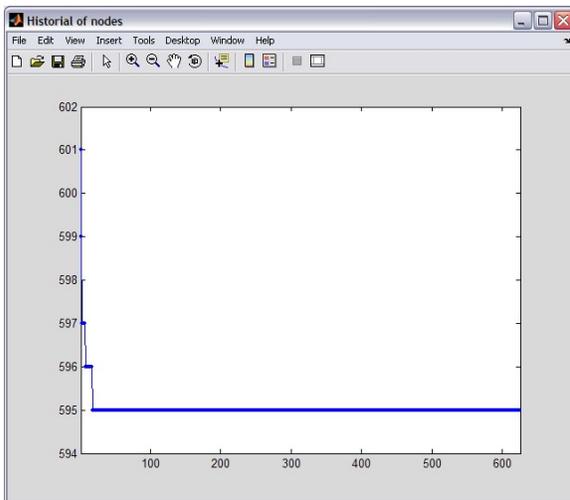


Figura A.21: Cantidad de nodos por iteración.

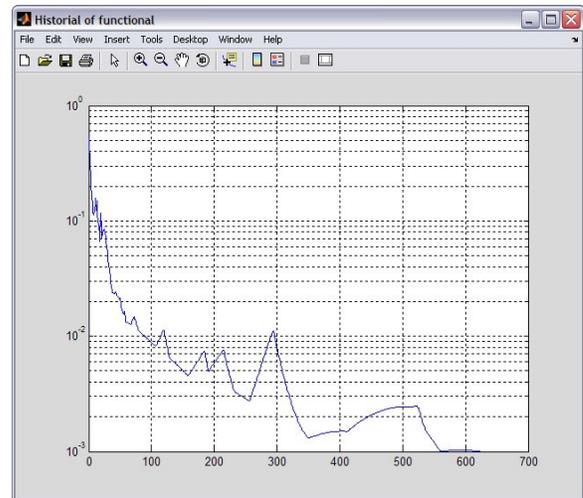


Figura A.22: Valor de la función óptima por iteración.

La estabilidad del método también es evidente desde los resultados de las medidas de calidad, figuras A.23 y A.24. El hecho de que no haya cambio en la medida de calidad topológica en las últimas iteraciones es reflejo de que los conectores se mantienen lo cual es condición para búsqueda de un estado estacionario. El poco cambio en la medida de calidad geométrica en las últimas iteraciones nos dice que el refinamiento de la malla está llegando a su fin, alcanzado un mínimo local.

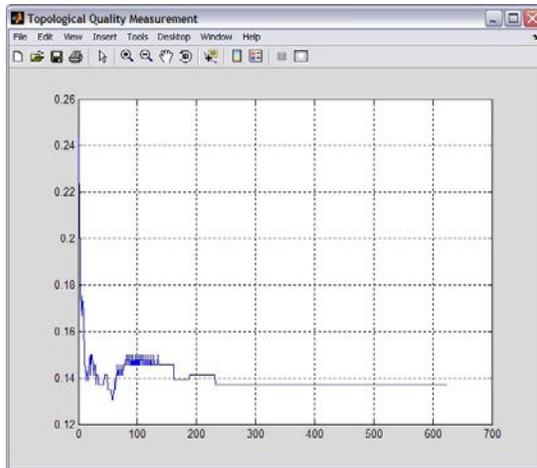


Figura A.23: Medida de calidad topológica por iteración.

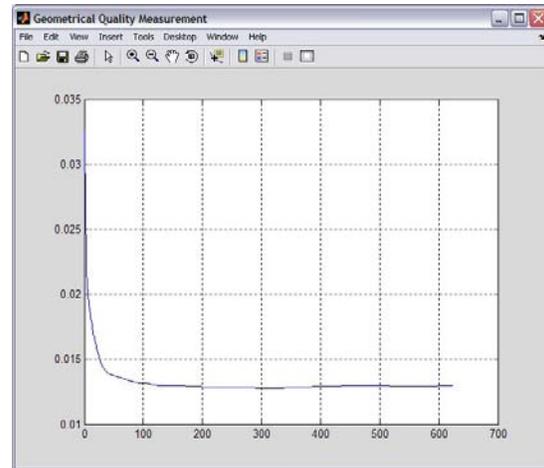


Figura A.24: Medida de calidad geométrica por iteración.

Con este ejemplo el usuario tuvo un primer acercamiento al sistema. En general el procedimiento que se ha descrito puede ser aplicado para cualquier otro contorno, en la mayoría de estos ejemplos la convergencia del método está garantizada siempre que sean usados los parámetros por default.



# Apéndice B

## Suavizamiento Spline Cónico

El siguiente tratamiento que se le puede dar a un contorno, independientemente si se reducen puntos en él o no, es de suavizamiento. Este se lleva a cabo por medio de Spline Cónicos con la premisa de conservar lo más que se pueda la forma del contorno original.

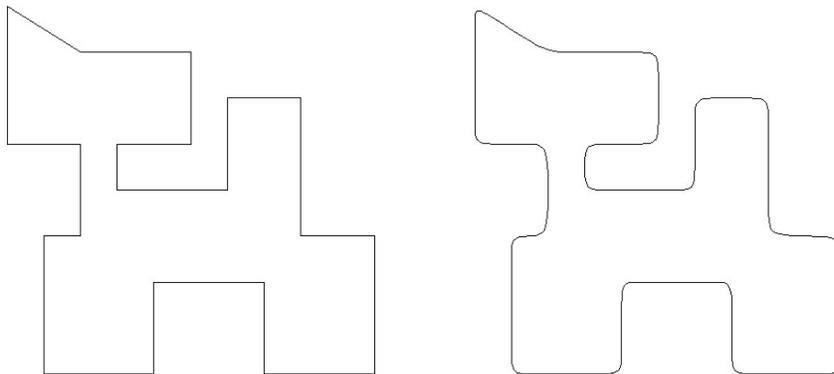


Figura B.1: Contorno: Gato.

La figura B.1 muestra un contorno (izquierda) y su suavizado (derecha), donde no se modifica sustancialmente.

La idea geométrica que hay detrás es, sustituir los picos de la poligonal por arcos simples. Estos son arcos cónicos y están definidos por tres puntos  $\mathbf{b}_0$ ,  $\mathbf{b}_1$  y  $\mathbf{b}_2$ , los cuales son llamados puntos de control, la figura B.2 muestra como se construye el arco.

Para poder tener una expresión manejable del arco cónico se recurre al siguiente resultado que ayuda a simplificar las expresiones.

**Teorema 2.** *Todo Arco Cónico definido de  $\mathbf{b}_0$  a  $\mathbf{b}_2$  puede expresarse por medio de una Curva Racional de Bézier en la siguiente forma:*

$$\mathbf{p}(t) = \frac{B_0^2(t)\mathbf{b}_0 + \delta B_1^2(t)\mathbf{b}_1 + B_2^2(t)\mathbf{b}_2}{B_0^2(t) + \delta B_1^2(t) + B_2^2(t)} \quad t \in [0, 1] \quad (\text{B.1})$$

donde  $\delta$  es llamado peso y  $B_i(t)$  son los polinomios de Bernstein.

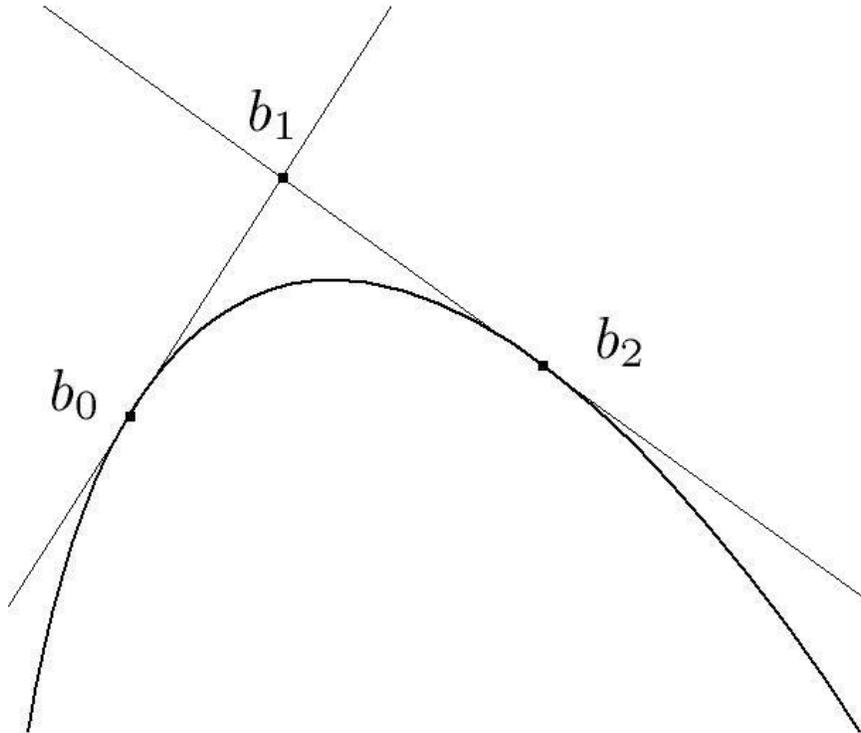


Figura B.2: Construcción del Arco Cónico.

## Puntos de Control

Dentro de la metodología empleada se tienen que considerar criterios de donde se colocarán los *puntos de control*, que definen a la cónica.

Esencialmente se han trabajado dos técnicas que han mostrado ser buenas en la práctica<sup>1</sup>. Se describirá en que consisten ambos métodos.

<sup>1</sup>véa *Un punto de vista sobre las cónicas y su uso en el suavizamiento de polígonos* [30]

### Los lados de la poligonal son de longitudes iguales.

Se procede de la siguiente manera: cada lado de la poligonal se divide en tres partes iguales, el arco será colocado en cada esquina, teniendo como descanso a cada costado (lado de la poligonal) un punto de control.

Cabe mencionar que por cada lado de la poligonal, se conserva un segmento del contorno original. La figura B.3 muestra algunos ejemplos.

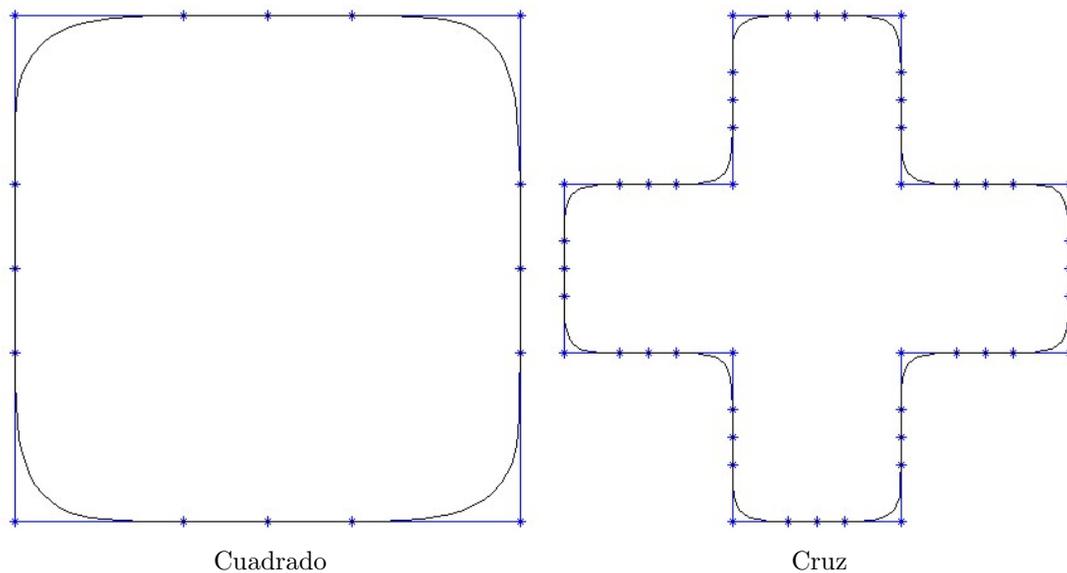


Figura B.3: Segmentos iguales.

### Los lados de la poligonal son de longitudes diferentes.

Si los lados del polígono son desiguales, se procede como sigue: búsquese la longitud menor  $L_{min}$  y con ello se realiza el siguiente criterio de asignación.

Llámesse  $L_i$  a la longitud del lado  $i$ -ésimo de la poligonal, así cuando  $L_i > 2L_{min}$  entonces, el punto, supóngase  $\mathbf{b}_0$ , se pone a distancia  $L_{min}/2$  de  $\mathbf{b}_1$  (*vértice a suavizar*). Si sucede lo contrario, es decir,  $L_i < 2L_{min}$  entonces  $\mathbf{b}_0$ , es el punto medio del mismo segmento.

Esta misma regla se sigue para el otro punto de control  $\mathbf{b}_2$ , hasta recorrer toda la poligonal, véase figura B.4.

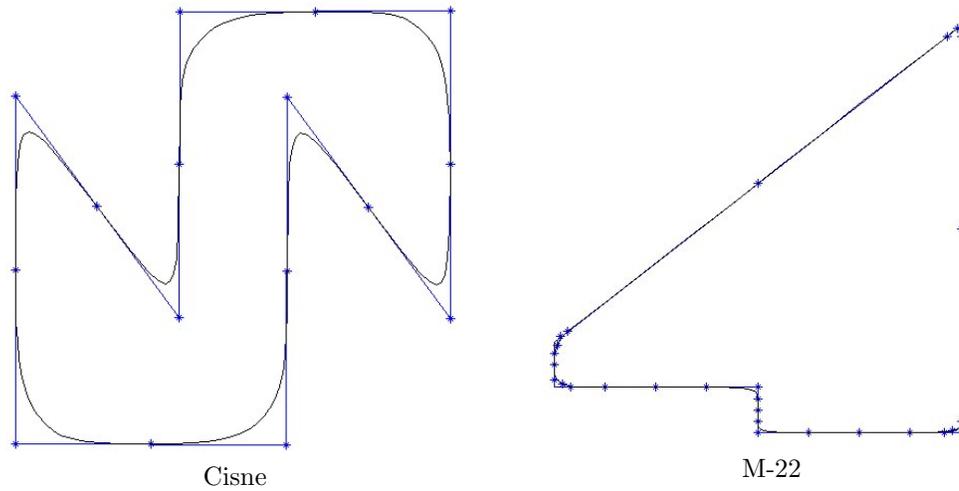


Figura B.4: Segmentos desiguales.

## Pesos

Luego de colocar los puntos de control, resta ver que tipo de cónica se asignará al vértice de la poligonal. Para ello utilizaremos el peso  $\delta$  que se muestra en [29], bajo el criterio que se mostrará, se elige una parábola, una elipse o una hipérbola.

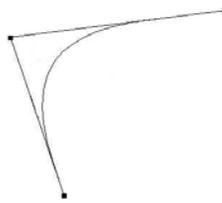
Si  $\theta$  es el ángulo formado por los segmentos  $\mathbf{b}_0\mathbf{b}_1$  y  $\mathbf{b}_1\mathbf{b}_2$ , entonces:

- Si  $0^\circ < \theta < 120^\circ$  se utiliza un *Arco Hiperbólico*.
- Si  $\theta = 120^\circ$  se utiliza un *Arco Parabólico*.
- Si  $120^\circ < \theta < 180^\circ$  se utiliza un *Arco Elíptico*.

*Arco Hiperbólico*



*Arco Parabólico*



*Arco Elíptico*



Figura B.5: Tipo de Arco según el ángulo.

## Continuidad $\mathcal{C}^1$

Una vez que se explicó como construir la nueva curva, que sustituirá a la poligonal original, se tiene la dificultad de tener un solo parámetro para describirla en su totalidad.

Para ello se aprovecha la propiedad geométrica que se tiene al pedir que las tangentes de los arcos cónicos sean los lados de la poligonal, a lo cual se llama continuidad geométrica.

Las condiciones de continuidad están dadas si se tienen los Arcos Cónicos

$$\mathbf{p}_0(t) = \frac{B_0^2(t)\mathbf{b}_0 + \delta_1 B_1^2(t)\mathbf{b}_1 + B_2^2(t)\mathbf{b}_2}{B_0^2(t) + \delta_1 B_1^2(t) + B_2^2(t)}$$

y

$$\mathbf{p}_1(t) = \frac{B_0^2(t)\mathbf{b}_2 + \delta_3 B_1^2(t)\mathbf{b}_3 + B_2^2(t)\mathbf{b}_4}{B_0^2(t) + \delta_3 B_1^2(t) + B_2^2(t)}$$

Entonces el Spline Cónico definido por:

$$s(u) = \begin{cases} \mathbf{p}_0(t(u)) : \text{con } t(u) = \frac{u-u_0}{u_1-u_0} \text{ y } u \in [u_0, u_1] \\ \mathbf{p}_1(t(u)) : \text{con } t(u) = \frac{u-u_1}{u_2-u_1} \text{ y } u \in [u_1, u_2] \end{cases}$$

es  $\mathcal{C}^1$  si y sólo si:

$$\frac{\delta_1(\mathbf{b}_2 - \mathbf{b}_1)}{\Delta_0} = \frac{\delta_3(\mathbf{b}_3 - \mathbf{b}_2)}{\Delta_1} \quad (\text{B.2})$$

donde  $\Delta_0 = u_1 - u_0$  y  $\Delta_1 = u_2 - u_1$ .

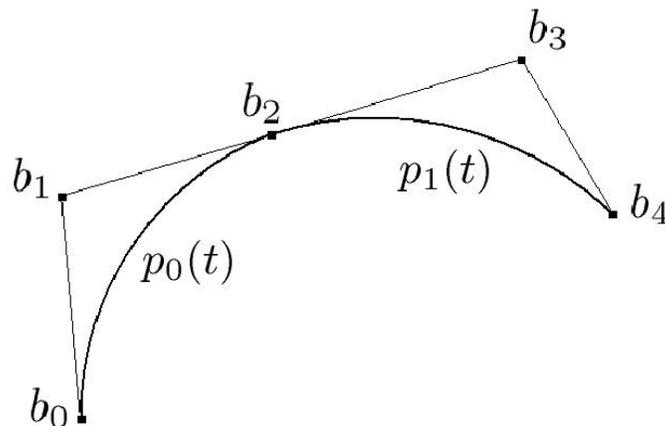


Figura B.6: Continuidad  $\mathcal{C}^1$ .

La figura B.6 muestra dos arcos. La idea que hay detrás es la de garantizar la continuidad en los puntos de unión y ello se logra en base a los nodos sobre los cuales está definida.

El algoritmo empleado para obtener la siguiente relación<sup>2</sup>, deja dos parámetros libres, por lo que  $u_0$  es tomado como 0 y  $u_1 = \|b_2 b_0\|$ . De esta forma la parametrización obtenida es  $\mathcal{C}^1$  y el *Spline Cónico* expresado como:

$$\begin{aligned} \cdot \quad & u_0 = 0 \\ \cdot \quad & u_1 = \|b_2 b_0\| \\ \cdot \quad & u_i = u_{i-1} + \Delta_{i-2} \frac{\delta_{2(i-1)+1} \|\Delta \mathbf{b}_{2(i-1)}\|}{\delta_{2(i-1)-1} \|\Delta \mathbf{b}_{2(i-1)-1}\|}, \quad \text{con } i = 2, \dots, n \end{aligned}$$

Entonces la continuidad de la curva se da en términos de los puntos de control que la componen, poniendo los intervalos de definición para cada arco  $[u_i, u_{i+1}]$ , en función de ellos.

## Reparametrización

Cuando se genera el Spline Cónico, se comienza con una distribución uniforme en el dominio de definición de los arcos cónicos y al realizar el suavizamiento los puntos resultantes no conservan dicha distribución, como se muestra en las graficas de la Figura B.7.

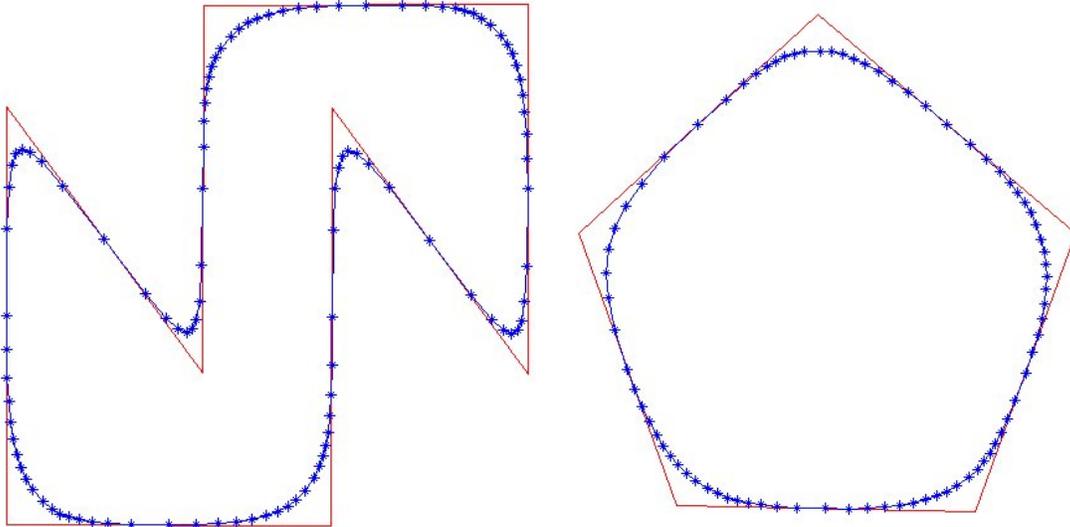


Figura B.7: Contornos suavizados.

Para obtener una distribución uniforme se realiza una reparametrización, ello con ayuda de conceptos de curvas que brevemente se recordarán.

<sup>2</sup>vea *Un módulo para el tratamiento de contornos en el sistema Unamalla* [9]

La función longitud de arco de una curva  $\beta(t) : I \rightarrow \mathbb{R}^2$ ,  $t \in [a, b]$  dada como:

$$s(t) = \int_a^t \|\beta'(\tau)\| d\tau$$

para  $a$  fija y  $t \in [a, b]$ .

**Teorema 3.** Si  $\beta$  es una curva regular en  $\mathbb{R}^2$  entonces, la reparametrización  $\tilde{\beta}$  de  $\beta$  por la longitud de arco, tiene rapidez unitaria.

Al reparametrizar por longitud de arco, se requiere de calcular:

$$\alpha(\tilde{s}) = s^{-1}(t) \quad \text{y} \quad \alpha'(\tilde{s}) = \frac{1}{\|\beta'(\alpha(\tilde{s}))\|}$$

La complicación numérica que se genera en este proceso radica en el cálculo de la inversa de la longitud de arco. Ya que la función es no lineal:

$$s(t) = \int_a^t \|\beta'(\tau)\| d\tau$$

Se necesita construir una buena aproximación. Desde un punto de vista numérico, ella se obtiene por medio del método de Newton:

$$t_i^{k+1} = t_i^k - \frac{s(t_i^k) - \tilde{s}_i}{s'(t_i^k)}, \quad \tilde{s}_i \in [0, l]$$

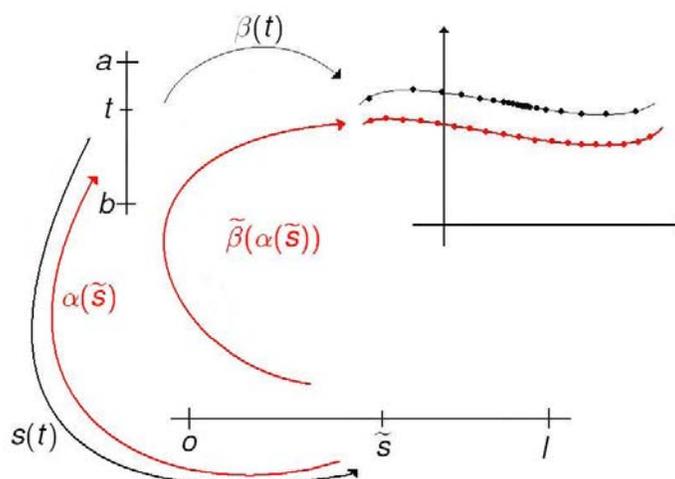


Figura B.8: Reparametrización por longitud de arco.

La figura B.8, muestra en un esquema como se utiliza la inversa de la función longitud de arco para obtener una distribución uniforme.

Una vez exploradas las ideas se tiene el siguiente algoritmo, que esté en términos de los intervalos de definición de los arcos cónicos.

Si el Spline  $\alpha$  está definido en el intervalo  $[u_0, u_n]$  de la siguiente manera:

$$\alpha(u) = \gamma_i(u) \quad u_i \leq u \leq u_{i+1} \quad i = 0, 1, \dots, n-1$$

y

$$\gamma_i(u) = \begin{cases} \gamma_{i1}(u) & u_i \leq u \leq u_{i+1/2} \\ \gamma_{i2}(u) & u_{i+1/2} \leq u \leq u_{i+1} \end{cases}$$

con  $u_{i+1/2} = (1 - \lambda)u_i + \lambda u_{i+1}$ ,  $\lambda \in [0, 1]$

$$\gamma_{i1}(u) = \frac{\omega_i t_i (u_{i+1/2} - u) + \omega_{i+1/2} t_{i+1/2} (u - u_i)}{\omega_i (u_{i+1/2} - u) + \omega_{i+1/2} (u - u_i)}$$

$$\gamma_{i2}(u) = \frac{\omega_{i+1/2} t_{i+1/2} (u_{i+1} - u) + \omega_{i+1} t_{i+1} (u - u_{i+1/2})}{\omega_{i+1/2} (u_{i+1} - u) + \omega_{i+1} (u - u_{i+1/2})}$$

$$\omega_{i+1} = \left( \frac{\alpha'(u_i)}{\alpha'(u_{i+1})} \right)^{1/2} \omega_i$$

$$t_{i+1/2} = [\lambda \omega_i \alpha'(u_i) + (1 - \lambda) \omega_{i+1} \alpha'(u_{i+1})] \frac{u_{i+1} - u_i}{t_{i+1} - t_i}$$

Donde  $\lambda$  y  $\omega$  son elegidos de manera adecuada para comenzar las iteraciones [9].

# Apéndice C

## Reducción de puntos

Dado un conjunto de puntos que definen una poligonal, se desea eliminar el “exceso” de puntos en ella y construir una nueva que cumpla con:

- Aproxime al contorno original.
- Respete la forma de la curva, reduciendo lo más que se pueda el número de puntos.

La idea central en todo el procedimiento es eliminar los puntos que estén “casi alineados”. La definición de colinealidad que se usará es la que dan los siguientes conceptos.

### Criterio de Longitud

Se tiene un conjunto de puntos  $\{p_i\}_{i=1}^k$  que determinan los vértices de una poligonal, definida por los segmentos  $\overline{p_i p_{i+1}}$  para  $i = 1, \dots, k$ .

Sea  $D$  la longitud del segmento formado por el conjunto  $\{p_i\}_{i=1}^k$  dado como:

$$D = \sum_{i=1}^{k-1} l_i$$

con  $l_i$  la longitud del segmento  $\overline{p_i p_{i+1}}$  y  $S$  a la longitud del segmento  $\overline{p_1 p_k}$ .

El criterio a seguir se observa cuando las longitudes  $D$  y  $S$  son parecidas, ello se refleja en el cociente, si este es cercano a 1 o no.

Sean  $R_0$  y  $R_1$  constantes tal que  $1 < R_0 < R_1$  tenemos:

- Si  $\frac{S}{D} \leq R_0$ . Los puntos son “colineales”.
- Si  $R_1 \leq \frac{S}{D}$ . Los puntos son no “colineales”.
- Si  $R_0 < \frac{S}{D} < R_1$ . Se aplica otra prueba de colinealidad.

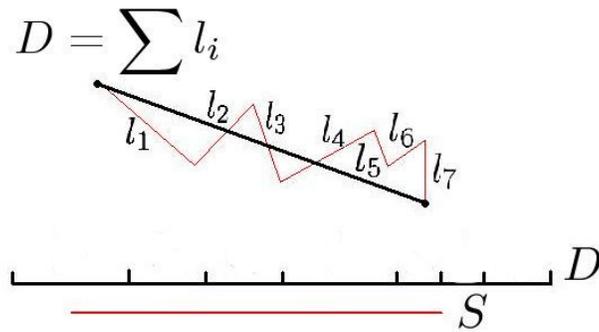


Figura C.1: Criterio de longitud.

## Criterio de Error Máximo

Este criterio está basado en la siguiente idea, al igual que en el caso anterior, sea el conjunto de puntos dado por  $\{p_i\}_{i=1}^k$  y  $E0$  un número positivo.

Diremos que el conjunto de puntos está alineado si éste cae en una vecindad de radio  $E0$  con centro en el segmento  $\overline{p_i p_k}$ , figura C.2.

Aquí también se tendrá un intervalo  $(E0, E1)$  de confianza donde se decidirá si aceptamos por “colineales” los puntos o no.

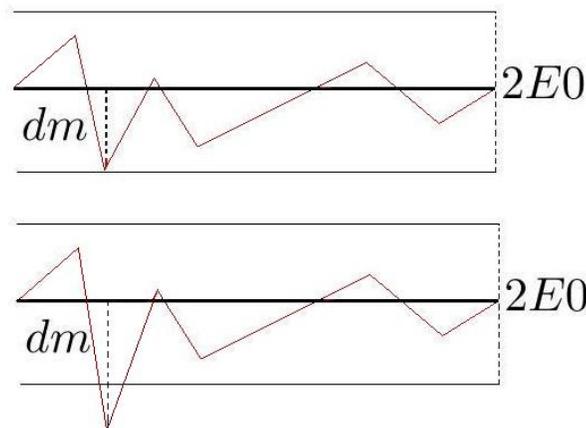


Figura C.2: Criterio de error máximo.

Si  $d_m$  es la distancia del punto más alejado del segmento y  $j$  es el índice del punto que alcanza esta distancia  $d_m = d(p_j, \overline{p_1 p_k})$ . Entonces se procede como sigue:

- Si  $d_m \leq E0$ . Los puntos son “colineales”.
- Si  $E0 < d_m < E1$ . Se aplica otra prueba de colinealidad.
- Si  $d_m \geq E1$ . Los puntos son no “colineales”.

## Parámetros

Una decisión importante es considerar, ¿cuántos puntos se deben tomar para aplicar los criterios de colinealidad?.

La práctica ha mostrado que con 7 puntos son bastante eficientes los métodos.

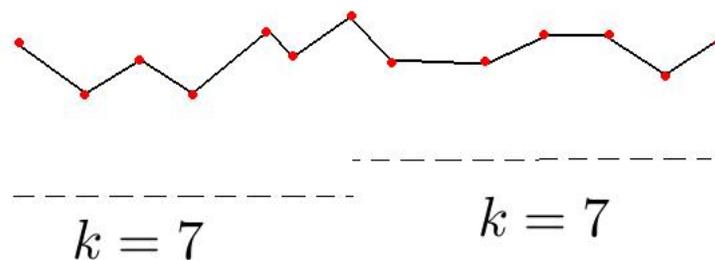


Figura C.3: Valor útil en la práctica.

## Ejemplos

Algunos ejemplos de contornos donde se aplicó la reducción de puntos se muestran a continuación.

El primer ejemplo (figura C.4) corresponde al contorno de la presa de Valle de Bravo, a la izquierda se encuentra el contorno original con 191 puntos y a la derecha el contorno reducido con 139 puntos.

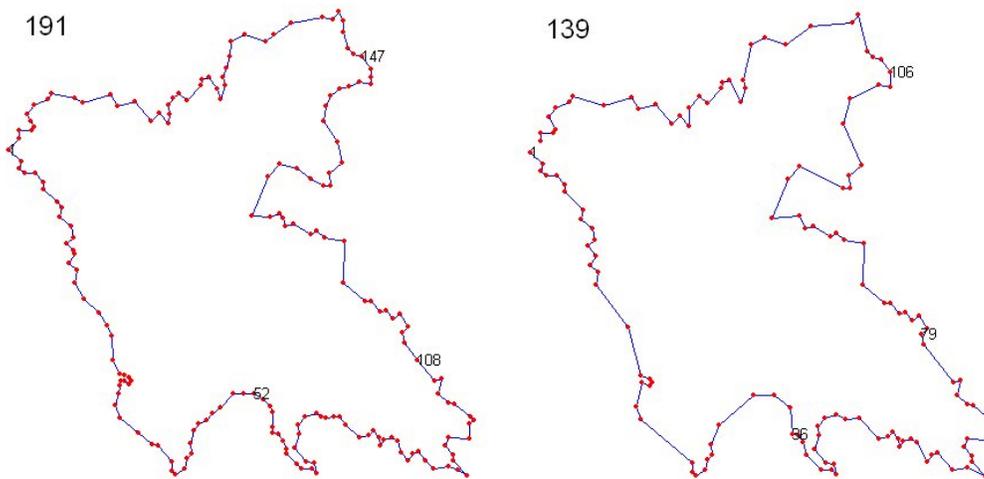


Figura C.4: Contorno: Presa en Valle de Bravo.

El segundo ejemplo (figura C.5) corresponde al contorno del Estado de México, a la izquierda se encuentra el contorno original con 397 puntos y a la derecha el contorno reducido con 218 puntos.

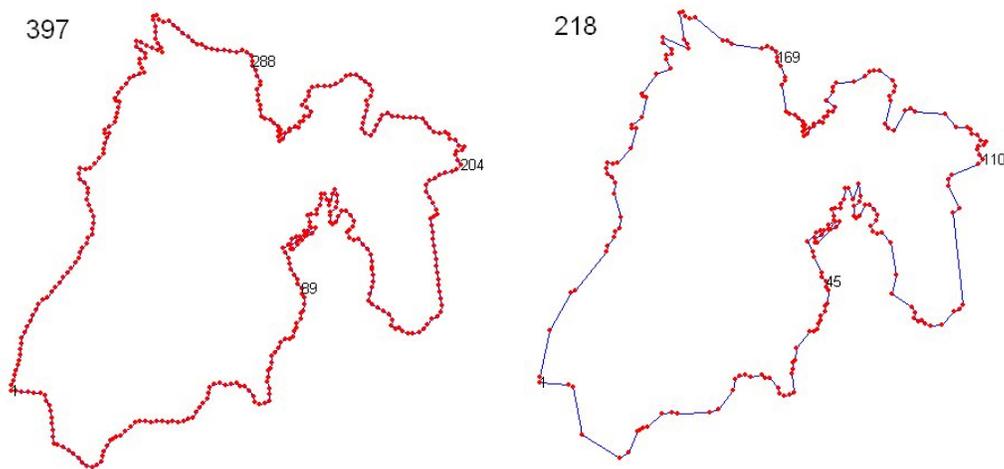


Figura C.5: Contorno: Edo. México.

# Bibliografía

- [1] Aguirre, S. *Introducción a las triangulaciones de Delaunay*. Reporte interno del Departamento de Matemáticas, Facultad de Ciencias, UNAM. 1995
- [2] Babuška, I and Aziz, A.K. *On the angle condition in the finite element method*. SIAM Journal on Numerical Analysis Vol 13(2), 1976, pp 214–226
- [3] Barnhill, R.E., Birkhoff, G., and Gordon, W.J. *Smooth interpolation in triangles*. J. Approx. Theory Vol. 8, 1973, pp 114–128
- [4] Barrera, P., Castellanos N., Domínguez-Mota, F.J., González, G., and Pérez D. *Adaptive discrete harmonic grid generation*. Mathematics and Computers in Simulation, Vol 79, Issue 6, 2009, pp 1792–1809
- [5] Barrera, P. Domínguez-Mota, F.J., González, G., and Tinoco R. *Generating quality structured convex grid on irregular regions*. Submitted to Electronic Transactions on Numerical Analysis. 2008
- [6] Barrera, P. Domínguez-Mota F., and González G. *Robust discrete grid generation on plane irregular regions*. Computational Mathematics and Mathematical Physics, 43, No.6, 2003, pp 845–853
- [7] Brossen, F.J. *Anisotropic mesh generation with particles*. Technical Report CMU-CS-96-134, CS Dept., Camegie Mellon University, 1996
- [8] Bykat, A. *Desing of a recursive, shape controlling mesh generator*. Int. J. Numer. Meth. Eng. Vol 19, 1983, pp 1375–1390
- [9] Carreón, C. *Un módulo para el tratamiento de contornos en el sistema Unamalla*. Tesis de Licenciatura, Facultad de Ciencias, UNAM, México 2008
- [10] Cavendish, J.C. *Automatic mesh generation for finite element analysis*. Int. J. Numer. Meth. Eng. Vol 8, 1974, pp 679–696
- [11] Coons, S.A. *Surfaces for computer-aided design of space forms*. Technical Report MAC-TR 44 MIT, Cambridge, MA, USA, 1967

- [12] Chew, L.P. *Constrained Delaunay triangulations*. In "Proc. 3rd Ann. Sympos. Comput. Geom.", 1987, pp 215–222
- [13] Frederick, C.O., Wong, Y.C., and Edge, F.W. *Two-dimensional automatic mesh generation for structural analysis*. Int. J. Numer. Meth. Eng. Vol 2, 1970, pp 133–144
- [14] George, J.A. *Computer implementation of the finite element method*. Ph. D. thesis, AAI7205916, Stanford University, Stanford, CA, USA. 1971
- [15] Gordon, W.J. *An operator calculus for surface and volume modeling*. IEEE Comput. Graph. & Appl., 1983
- [16] Haber, R., Shephard, M.S., Abel, J.F., Gallagher, R.H., and Greenberg, D.P. *A general two-dimensional graphical finite element preprocessor utilizing discrete transfinite mappings*. Int. J. Numer. Meth. Eng. Vol 17, 1981, pp 1015–1044
- [17] Hall, C.A. *Transfinite interpolation and applications to engineering problems*. in Law and Sahney (eds) Theory of approximations Academic Press, 1976, pp 308–331
- [18] Heighway, E.A. and Biddlecombe, C.S. *Two-dimensional automatic triangular mesh generation for the finite element electromagnetics package PE2D*. IEEE Trans, on Mag. Vol MAG-18 No. 2, 1982, pp 594–598
- [19] Ho-Le, K. *Finite element mesh generation methods: a review and classification*. Computer Aided Design, Vol 20, No. 1, 1988, pp 27–38
- [20] Ivanenko, S.A. *Generation of non-degenerate meshes*. USSR Comput. Maths. Math. Phys. Vol. 28, No.2, 1988, pp 141–146
- [21] Kela, A., Perucchio, R., and Voelcker, H.B. *Toward automatic finite element analysis*. Comput. Mech. Eng. Vol 5 No. 1, 1986, pp 57–71
- [22] Kikuchi, N. *Adaptive grid-design methods for finite element analysis*. Comput. Methods in Applied Mech. Eng. Vol. 55, 1986, pp 129–160
- [23] Lee, D.T. and Schachter, B.J. *Two algorithms for constructing a Delaunay triangulation*. Internat. J. Comput. Inform. Sci. 9, No. 3, 1980, pp 219–242
- [24] Lee, Y.T. *Automatic finite element generation based on constructive solid geometry*. Ph. D. thesis Mechanical Engineering Dept, University of Leeds, Leeds, UK. 1983
- [25] Lewis, B.A. and Pobinson, J.S. *Triangulation of planar regions with applications*. The Computer Journal Vol 21 No. 4, 1978, pp 324–332

- 
- [26] Lo, S.H. *A new mesh generation scheme for arbitrary planar domains*. Int. J. Numer. Meth. Eng. Vol 21, 1985, pp 1403–1426
- [27] Nelson, J.M. *A triangulation algorithm for arbitrary planar domains*. Appl. Math. Modeling, Vol 2, 1978, pp 151–159
- [28] Persson, P., and Strang, G. *A simple mesh generator in Matlab*. SIAM Rev. Vol 46, No. 2, 2004, pp 329–345
- [29] Rivera, A. *Fundamentos de curvas de bézier racionales y cálculo numérico de la reparametrización de curvas*. Tesis de Maestría, IIMAS, UNAM, México, 2005
- [30] Rivera, A. *Un punto de vista sobre las cónicas y su uso en el suavizamiento de polígonos*. Tesis de Licenciatura, Facultad de Ciencias, UNAM, México, 1999
- [31] Seidel, R. *Constrained Delaunay triangulations and Voronoi diagrams with obstacles*. In “1978-1988 Ten Years IIG”, 1988, pp 178–191
- [32] Shamos, M.I. *Computational Geometry*. Ph.D. thesis, UMI 7819047, Yale University, New Haven, Conn. 1978
- [33] Shimada K. and Grossard D. *Bubble mesh: Automated triangular meshing of non-manifold geometry by sphere packing*. In Third Symp. on Solid Modeling and Appls., 1995, pp 409-419
- [34] Shimada K. and Grossard D. *Computational methods for physically-based fe mesh generation*. In PROLAMAT. IFIP TC5/WG5.3, 1992, pp 411–420
- [35] Thacker, W.C., Gonzalez, A. and Putland, G.E. *A method for automating the construction of irregular computational grids for strom surge forecast models*. J. of Computational Physics. Vol 37, 1980, pp 317–387
- [36] Yerry, M.A. and Shephard, M.S. *A modified quadtree approach to finite element mesh generatior*. IEEE Comput. Graph. & Appl., 1983, pp 39–46
- [37] Zienkiewicz, O.C. and Phillips, D.V. *An automatic mesh generation scheme for plane and curved surfaces by “isoparametric” co-ordinates*. Int. J. Numer. Meth. Eng. Vol 3, 1971, pp 519–528