



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS MATEMÁTICAS Y
DE LA ESPECIALIZACIÓN EN ESTADÍSTICA APLICADA

PARAMETRIZACIÓN DE REGIONES POLIGONALES COMPLICADAS

TESIS
QUE PARA OPTAR POR EL GRADO DE:
DOCTOR EN CIENCIAS

PRESENTA:
IVÁN MÉNDEZ CRUZ

DIRECTOR DE TESIS:
DR. PABLO BARRERA SÁNCHEZ
FACULTAD DE CIENCIAS UNAM

COMITÉ TUTOR:
DR. JESÚS LÓPEZ ESTRADA
FACULTAD DE CIENCIAS UNAM
DR. FRANCISCO JAVIER DOMÍNGUEZ MOTA
FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS UMSNH

CIUDAD DE MÉXICO, MAYO 2023.



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.

A mis padres, Artemia y Bulmaro.

Para mis tíos, padrinos y primos.

AGRADECIMIENTOS

Mi más sincero agradecimiento a mi director de tesis, el profesor Pablo Barrera Sánchez, su tutoría fue indispensable en mi formación académica.

Quiero expresar mi gratitud a mis sinodales, Jesús López Estrada, Francisco Javier Domínguez Mota, José Gerardo Tinoco, Irma Delia García y Julián Tercero por sus sugerencias y observaciones que enriquecen este trabajo.

Deseo agradecer a los profesores Guilmer Ferdinand y Luis Alberto Vazquez, y a mis compañeros del proyecto *UNAMALLA*: Gustavo, Jorge, Isidro, Javier de Jesús, Leticia, Adriana Rivera y César, sus consejos y trabajos fueron de gran ayuda.

Agradezco a la Facultad de Ciencias UNAM por permitirme realizar mis estudios de posgrado y al CONACYT por el apoyo económico brindado durante la realización de este proyecto.

Me siento agradecido con mis padres por su enorme apoyo en mis estudios.

Muchas gracias a Gloria Olimpia, esposa del profesor Pablo, por atenderme en su hogar y brindarme atención médica, además de ayudarme a revisar el texto.

A Mario y Valeria, con quienes he trabajado durante los últimos años, a Miguel Raz y Raymundo, gracias por su apoyo.

A mis amigos y compañeros de la UNAM, gracias.

Algunos problemas de dinámica de fluidos, aeronáutica e ingeniería asistida por computadora necesitan representar cuerpos de agua, máquinas y materiales de geometría complicada mediante *mallas*, que consisten en ensamblajes de figuras elementales llamadas *celdas*. En este proyecto generamos mallas sobre regiones poligonales de modo que sus celdas sean cuadriláteros que empaten lado a lado y respeten las fronteras de las regiones.

Las celdas de una malla de cuadriláteros se pueden alinear con direcciones dadas y adaptarse a animaciones y simulaciones, para ello necesitamos que tengan el mínimo número de *singularidades interiores*, a saber, los puntos interiores que no se conectan exactamente con otros cuatro puntos.

Deseamos generar mallas sin singularidades interiores, en particular nos interesan las *mallas estructuradas de cuadriláteros*, que pueden verse como un arreglo rectangular mediante un mapeo. Estas mallas son más sencillas de manipular, se pueden generar sobre regiones con diferente grado de dificultad, e incluso producen matrices en banda en algunos métodos numéricos de las ecuaciones diferenciales, sin embargo pueden tener cuadriláteros distorsionados.

Algunas aplicaciones necesitan que los cuadriláteros de la malla cumplan propiedades geométricas deseables, nosotros queremos una *malla de alta calidad* en el sentido de que la mayoría de sus cuadriláteros sean aproximadamente rectángulos, por lo que separamos regiones complicadas en bloques admisibles para generar mallas con la calidad deseada.

En este proyecto generamos mallas de alta calidad sin singularidades interiores como *mallas estructuradas por bloques*, que son familias de mapeos del cuadrado unitario a los bloques de la región. En nuestra metodología abordamos problemas de aproximación de contornos, descomposición de regiones, programación entera y optimización de mallas.

Generar mallas sobre regiones con alto nivel de detalle o ruido en la frontera puede ser computacionalmente costoso, más aún, si queremos mallas de alta calidad, por lo que simplificamos las fronteras de las regiones. Motivados por ideas de la Cartografía, desarrollamos un nuevo método para simplificar contornos poligonales.

Una vez que aproximamos la frontera, descomponemos la región en bloques sin agujeros, que a su vez separamos en bloques admisibles. Podemos separar la región de diferentes maneras, en nuestro caso realizamos cortes para no introducir singularidades interiores. Cabe mencionar que la calidad de la malla depende de la elección de los cortes, para seleccionarlos usamos criterios geométricos como la concavidad.

Luego de descomponer la región en bloques, separamos sus fronteras en cuatro lados delimitados por esquinas y construimos mapeos compatibles entre los lados del cuadrado y de los bloques. Distribuimos el mismo número de puntos en lados opuestos cuidando que coincidan en los cortes para construir los mapeos y resolvemos un problema de programación lineal entera con restricciones para obtener tamaños compatibles que dependan de las longitudes de los lados.

Extendemos los mapeos compatibles para generar mallas estructuradas sobre los bloques, unimos estas mallas en una malla global y mejoramos su calidad al mover sus puntos interiores mediante el enfoque variacional, en el cual minimizamos una función que mide propiedades geométricas de sus celdas.

Nuestro método reduce la generación de mallas estructuradas por bloques a la elección adecuada de cortes y esquinas, su implementación en **Julia** dió lugar al lanzamiento de la nueva versión del sistema **UNAMalla 6**, que le permite al usuario generar de manera interactiva mallas de alta calidad sobre regiones poligonales complicadas.

Con base en lo anterior, organizamos este trabajo como sigue:

Capítulo 1. Damos una introducción a la generación de mallas de cuadriláteros, describimos nuestra metodología y mencionamos las principales contribuciones del proyecto.

Capítulo 2. Hacemos hincapié en la calidad de la malla y formulamos la generación de mallas de alta calidad como la construcción de parametrizaciones admisibles y compatibles.

Capítulo 3. Revisamos antecedentes sobre aproximación de contornos con énfasis en la simplificación de polígonos en cartografía y reconocimiento de patrones, y presentamos un nuevo método de simplificación.

Capítulo 4. Revisamos la literatura sobre descomposición de regiones, indicamos las características que debe reunir una región admisible para generar mallas de alta calidad y describimos nuestra descomposición por cortes.

Capítulo 5. Explicamos la construcción de mapeos compatibles para los lados de los bloques, indicamos nuestra selección de esquinas y la formulamos el problema de programación entera para determinar los tamaños compatibles.

Capítulo 6. Describimos cómo generar, unir y suavizar las mallas estructuradas sobre los bloques, además revisamos la generación variacional de mallas.

Apéndice A. Damos una breve descripción del generador de mallas **UNAMalla 6**.

Apéndice B. Planteamos cómo construir mapeos de regiones rectilíneas a regiones poligonales mediante mallas estructuradas por bloques.

En la página web:

<http://tikhonov.fciencias.unam.mx/unamalla/>

se puede descargar las rutinas del generador de mallas **UNAMalla 6**, señalamos los requisitos y ponemos a disposición una guía rápida para su instalación y uso, además de una galería de mallas que pueden visualizarse de manera interactiva.

ÍNDICE GENERAL

Prefacio	I
1. Introducción	1
1.1. Planteamiento del Problema	1
1.2. Mallas Estructuradas por Bloques	6
1.3. Logros y Contribuciones	11
1.4. Método para Generación de Mallas	12
1.5. Resumen del Capítulo	16
2. Parametrizaciones por Mallas Estructuradas	19
2.1. Parametrización del Cuadrado Unitario a la Región	20
2.1.1. Parametrización de Regiones Simplemente Conexas	20
2.1.2. Parametrización de Regiones Múltiplemente Conexas	25
2.2. Parametrizaciones Admisibles	27
2.3. Parametrización con Mallas por Bloques	30
2.4. Resumen del Capítulo	35
3. Aproximación de Contornos	37
3.1. Simplificación de Polígonos	42
3.1.1. Simplificación de Polígonos en Cartografía	44
3.2. Reducción de Ruido	45
3.2.1. Detección de Ruido	45
3.2.2. Método para Reducción de Ruido	47
3.3. Nuevos Métodos para Aproximación de Polígonos	49
3.3.1. Criterio de las Áreas	50
3.3.2. Criterio de los Radios	51
3.3.3. Simplificación con Reducción de Ruido	52
3.3.4. Métodos Modificados	53
3.4. Resumen del Capítulo	60
4. Descomposición de Regiones	63
4.1. Regiones y Descomposiciones Admisibles	63
4.2. Métodos para Descomposición de Regiones	67
4.2.1. Método para Descomposición Admisible	71
4.2.2. Descomposición Simplemente Conexas	76
4.2.3. Descomposición Aproximadamente Convexa	77
4.2.4. Descomposición para Ángulos Críticos	79
4.3. Gráficas de la Descomposición Admisible	80

4.4. Medidas y Criterios de Concavidad para Descomposición Admisible	82
4.5. Resumen del Capítulo	86
5. Construcción de Parametrizaciones Compatibles	87
5.1. Descomposición de la Frontera	88
5.1.1. Elección de Esquinas	89
5.2. Generación de Tamaños Compatibles	92
5.2.1. Ecuaciones de Compatibilidad	92
5.2.2. Compatibilidad por Programación Entera	95
5.3. Reparametrización de la Frontera	99
5.4. Resumen del Capítulo	101
6. Generación de Mallas Estructuradas por Bloques	103
6.1. Interpolación Transfinita	104
6.2. Unión de Mallas	105
6.2.1. El orden para la unión de mallas	106
6.2.2. Unión recursiva de mallas	108
6.2.3. Numeración para la unión de mallas	109
6.3. Suavizamiento de Mallas por Bloques	110
6.4. Generación Variacional de Mallas	115
6.4.1. Generación Variacional Continua	115
6.4.2. Generación Variacional Discreta	117
6.5. Resumen del Capítulo	125
Conclusiones y Trabajo Futuro	127
A. Generador de Mallas Estructuradas UNAMalla 6	129
B. Parametrizaciones sobre Regiones Rectilíneas	135
Bibliografía	139

1.1. Planteamiento del Problema

Varios problemas de Computación Científica en 2D, requieren separar regiones planas en subregiones elementales llamadas *celdas* como son triángulos, cuadriláteros, hexágonos, entre otras. Métodos para resolver estos problemas dependen de las propiedades geométricas de las celdas. La idea es representar las regiones por ensamblajes de celdas conocidos como *mallas*.

Las mallas sobre regiones planas pueden verse como mosaicos decorativos sobre pisos o paredes, donde las baldosas del mosaico corresponden a las celdas de la malla. En este contexto, generar mallas puede pensarse como colocar baldosas sobre la fachada de una casa, Fig. 1.1. Un ejemplo más complicado es cubrir la silueta de un animal, Fig. 1.2.

Las mallas tienen diversas aplicaciones en la Ciencia e Ingeniería, no sólo se generan con fines recreativos, sino que son herramientas indispensables en la modelación computacional. Mencionamos algunas aplicaciones:

- En el Diseño Asistido por Computadora se usan mallas para crear personajes animados [14], Fig. 1.3.
- En las simulaciones computacionales de turbomáquinas y aeronaves se necesitan mallas sobre ventiladores y alas de aviones [77, 122, 125], Fig. 1.4.
- En el Procesamiento de Imágenes Médicas se generan mallas sobre superficies extraídas de las imágenes [67, 150], Fig. 1.5.
- En la Industria Automotriz se generan mallas sobre piezas de autos para realizar simulaciones de choques, Fig. 1.6.
- Simulaciones de flujo en cuerpos de agua requieren mallas sobre las regiones de estudio [85, 87, 123], Fig. 1.7.

En la literatura predominan las mallas con celdas de triángulos. No obstante, en algunas simulaciones computacionales se prefieren que todas las celdas sean cuadriláteros, pues estas celdas se pueden acomodar en una estructura adecuada para realizar las simulaciones de manera eficiente, más aún, en algunos problemas de la Dinámica Computacional de Fluidos se han reportado errores de discretización más pequeños usando cuadriláteros en comparación con las mallas de triángulos [99]. Otra ventaja es que se pueden alinear con la geometría de la región, esta característica es de utilidad para simulaciones de fluidos y generar personajes animados, Fig. 1.3.

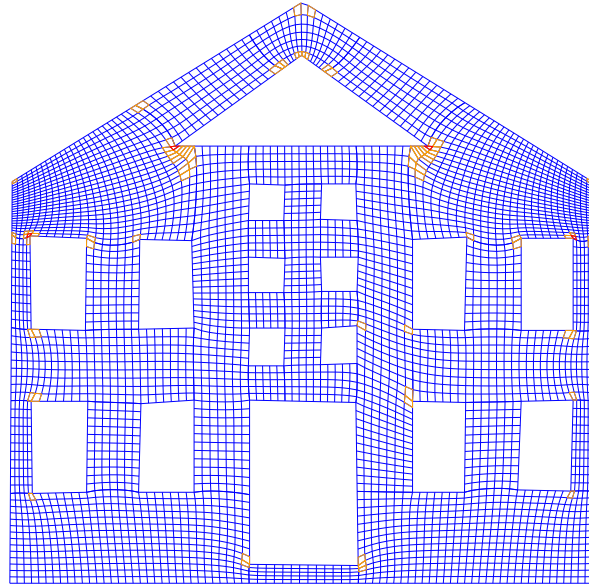


Figura 1.1: Malla de cuadriláteros sobre fachada de una casa.

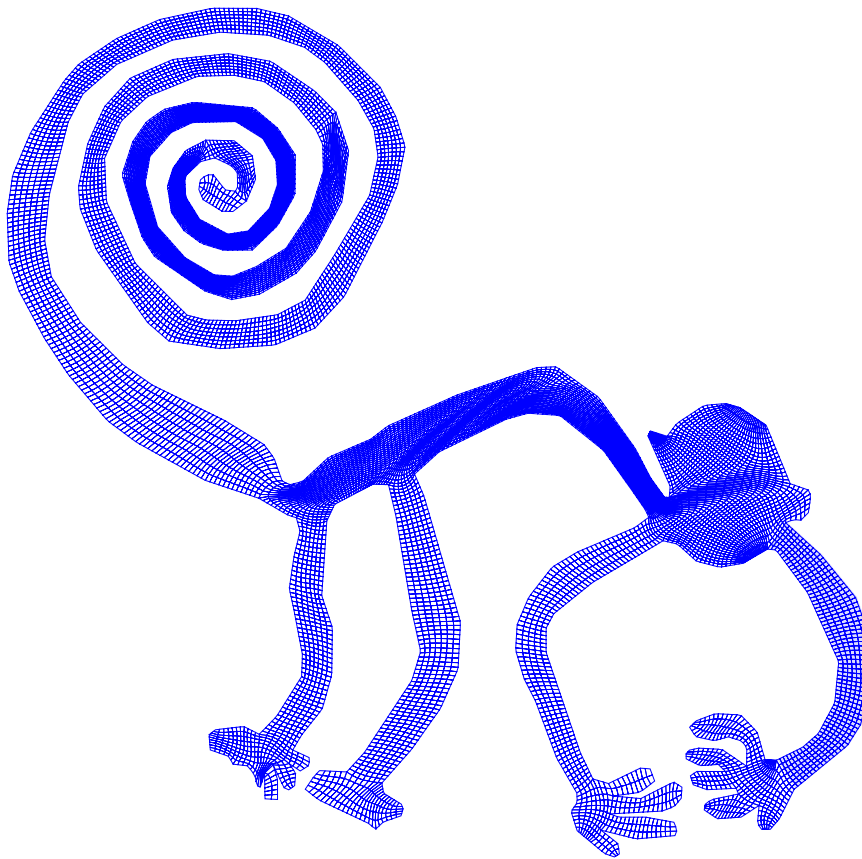


Figura 1.2: Malla de cuadriláteros sobre la silueta del mono de las Líneas de Nazca.

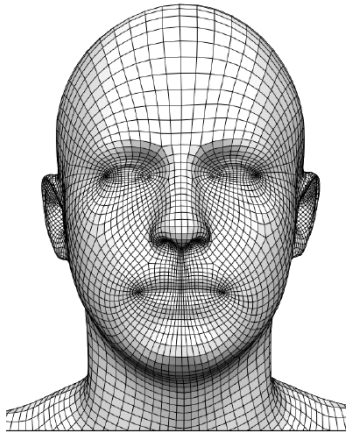


Figura 1.3: Malla de cuadriláteros sobre un rostro [32]. Las celdas se alinean con los rasgos faciales.

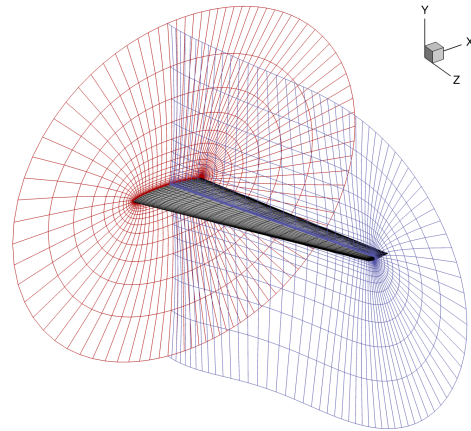


Figura 1.4: Mallas de cuadriláteros para el ala de un avión. [122].

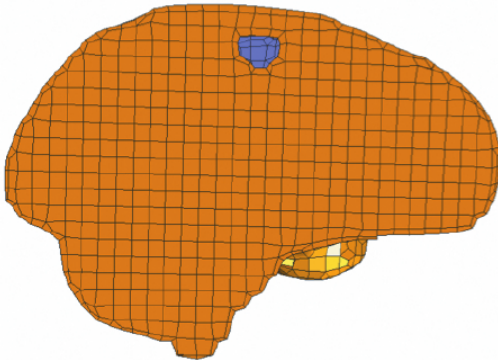


Figura 1.5: Malla de cuadriláteros sobre una sección transversal del cerebro. Las celdas azules corresponden a un tumor [150].

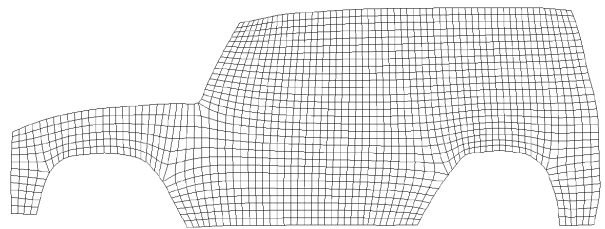


Figura 1.6: Malla de cuadriláteros sobre un perfil de la carrocería de una camioneta [73].

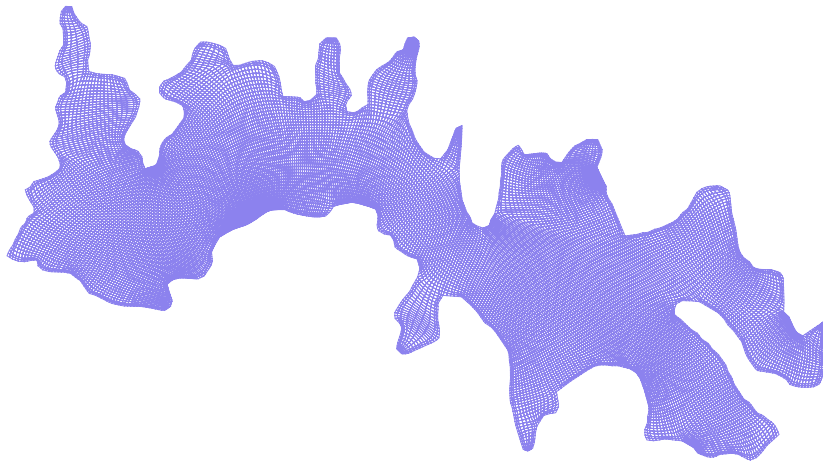


Figura 1.7: Malla de cuadriláteros sobre la Presa Villa Victoria en el Estado de México.

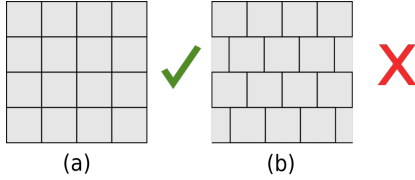


Figura 1.8: Comparación de mallas de cuadriláteros. En (a) los cuadriláteros empatan lado a lado a diferencia de la malla en (b).

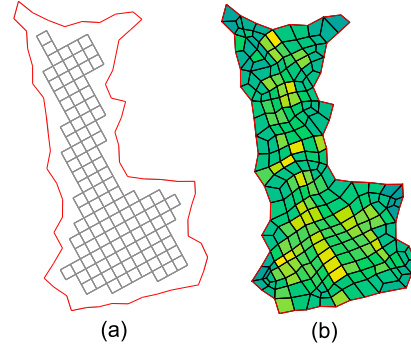
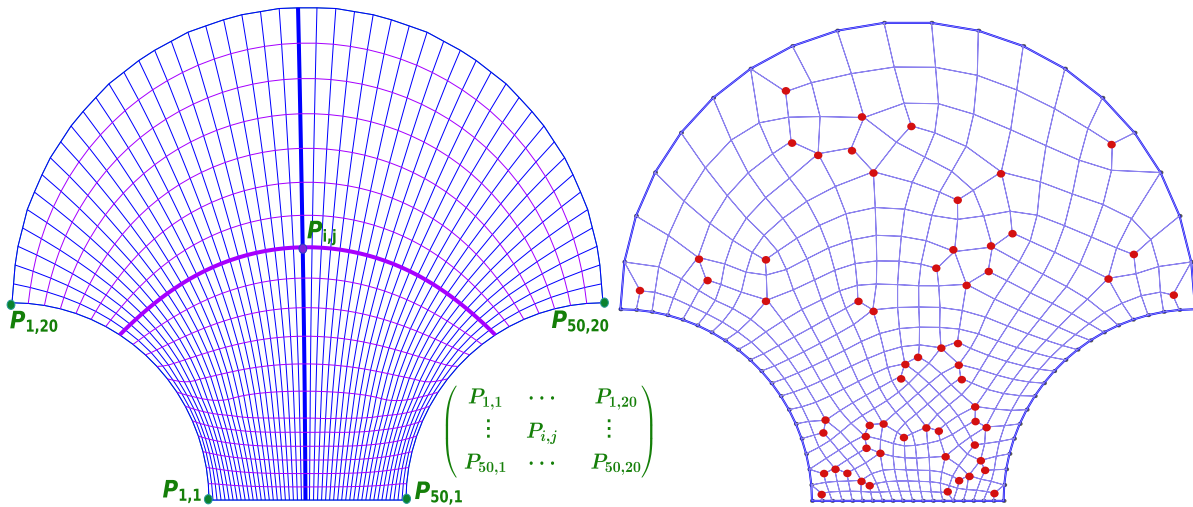


Figura 1.9: Mallas de cuadriláteros sobre una misma región [123]. La malla en (a) no respeta la frontera a diferencia de la malla en (b)



(a) Malla estructurada de tamaño 50×20 se visualiza como un arreglo rectangular de puntos y sus líneas atraviesan la región respetando la geometría.

(b) Malla no estructurada tiene singularidades interiores de color rojo, donde se conectan tres o cinco líneas en vez de cuatro.

Figura 1.10: Mallas de cuadriláteros sobre una concha.

Nuestro problema es generar mallas de cuadriláteros sobre regiones poligonales que pueden tener agujeros, por ejemplo una laguna con islas, Fig. 2.2. Nos interesa que las celdas empaten lado a lado, Fig. 1.8, y que respeten la frontera de la región, Fig. 1.9.

La *conectividad de la malla* es un aspecto importante, que consiste en identificar las conexiones de sus puntos. Lo ideal es generar mallas de cuadriláteros con la propiedad de que cada punto interior se conecta exactamente con cuatro puntos. Las *mallas estructuradas de cuadriláteros* tienen esta propiedad y cada uno de sus puntos frontera se conecta exactamente con tres puntos, con excepción de las esquinas que se conectan con dos puntos. Estas mallas pueden verse como un arreglo rectangular de celdas, su tamaño está dado por el arreglo, más aún, sus celdas se ordenan por parejas de índices y se alinean siguiendo las direcciones de las columnas y los renglones, Fig. 1.10(a).

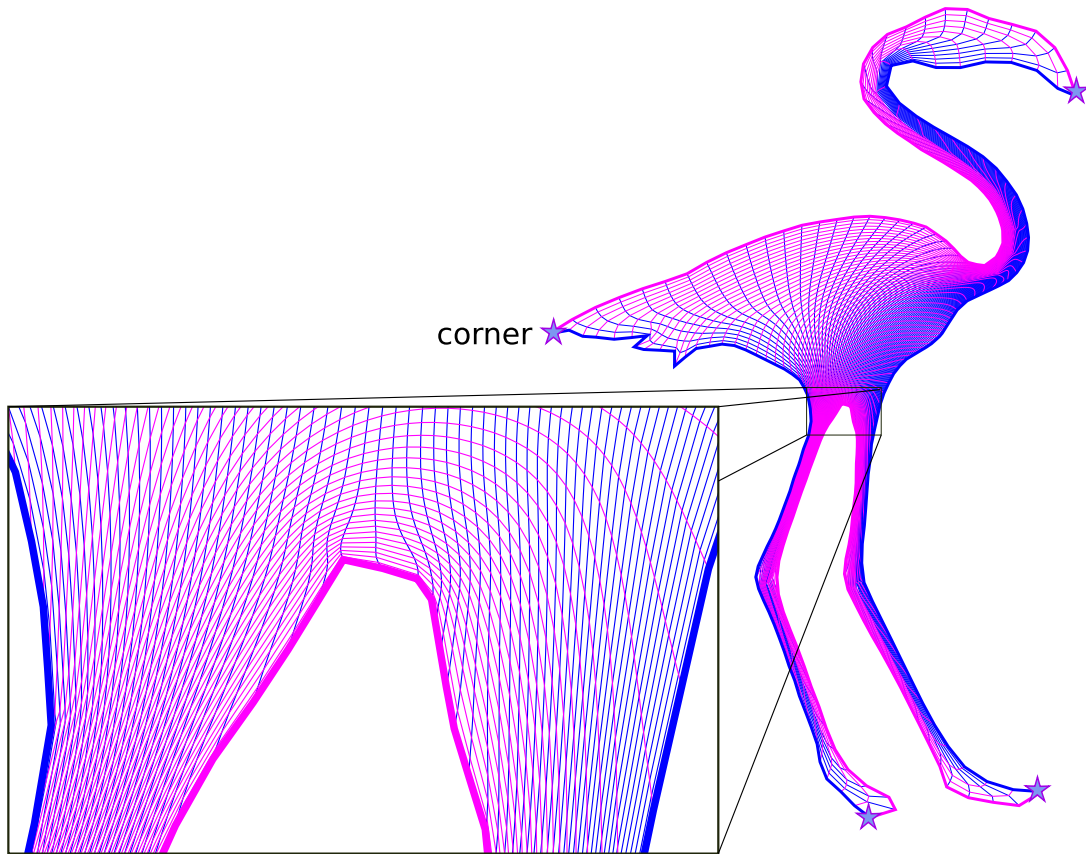


Figura 1.11: Malla estructurada de cuadriláteros de tamaño 80×80 sobre un flamenco.

Los puntos interiores de una malla de cuadriláteros que no se conectan exactamente con cuatro puntos se llaman *singularidades interiores*, Fig. 1.10(b). En este trabajo generamos mallas sin singularidades interiores, como se muestra en las figuras 1.1 y 1.2.

Varios avances en la generación de mallas se han enfocado en la generación de mallas con un alto porcentaje de singularidades. No obstante, en algunas simulaciones computacionales es recomendable generar mallas de cuadriláteros con el mínimo número de singularidades interiores, ya que pueden ocasionar comportamientos inesperados.

Además de la conectividad de la malla, examinamos su calidad. Algunas aplicaciones requieren que las celdas posean propiedades geométricas específicas, en particular es indispensable que todas las celdas sean convexas, más aún, en algunos modelos computacionales es conveniente que las celdas formen ángulos rectos, tengan lados paralelos, o aproximadamente la misma área, entre otras propiedades [5].

Queremos generar *mallas de cuadriláteros de alta calidad* en el sentido de que un alto porcentaje de los cuadriláteros sean aproximadamente rectángulos. Así que formulamos nuestro problema como sigue:

Problema: Dada una región poligonal, generar una malla de cuadriláteros sin singularidades interiores y de alta calidad sobre esa región de modo que las celdas empaten lado a lado y se preserve la forma de la región.



Figura 1.12: Un panel solar visto como una malla estructurada por bloques.

En principio, deseamos generar mallas estructuradas de cuadriláteros, al respecto Barreira et al. [11, 10] han generado estas mallas sobre regiones complicadas usando el software UNAMalla 4.0 [138], sin embargo no siempre tienen la calidad deseada, pues en algunas ocasiones las celdas alrededor de la frontera presentan distorsiones, Fig. 1.11. Más aún, generar mallas estructuradas sobre regiones con múltiples ramificaciones, cuellos de botella o vueltas puede ser casi imposible en la práctica.

1.2. Mallas Estructuradas por Bloques

En lugar de generar mallas estructuradas sobre una región complicada, las generamos sobre subregiones, representamos la región por parches que esencialmente son cuadriláteros. Una malla de cuadriláteros que se puede separar en submallas estructuradas se llama *malla estructurada por bloques*, ésta puede verse como la unión de arreglos rectangulares de cuadriláteros, sus bloques forman una malla de polígonos con celdas que a su vez se separan en cuadriláteros. Un ejemplo intuitivo es el panel solar, Fig. 1.12, mientras que un ejemplo más complicado es la malla sobre el flamenco, Fig. 1.13.

Nos interesa generar mallas de alta calidad sobre regiones poligonales complicadas, su conectividad depende de las submallas sobre las subregiones, por un lado debemos evitar introducir singularidades interiores cuando separamos la región, y por otra parte las celdas deben empatar lado a lado entre los bloques. Así que reformulamos nuestro problema como sigue:

Reformulación del Problema: Dada una región poligonal, descomponerla en una colección finita de subregiones poligonales que no se traslapen y generar mallas estructuradas de cuadriláteros de alta calidad sobre estas subregiones de modo que sus cuadriláteros empaten lado a lado sin singularidades interiores en las intersecciones de las subregiones.

Algunos modelos computacionales requieren mallas estructuradas por bloques, pues estas mallas poseen una conectividad que brinda mayor control sobre la estructura del modelo. Métodos numéricos para resolver ecuaciones diferenciales parciales sobre regiones complicadas como el Análisis de Elemento Finito y el Análisis Isogeométrico [53] pueden beneficiarse al usar estas mallas [58, 63], pues los cálculos se realizan de manera más eficiente y se pueden obtener resultados más precisos [5], más aún, podemos generarlas con alta calidad sobre regiones complicadas a comparación de las mallas estructuradas de un solo bloque.

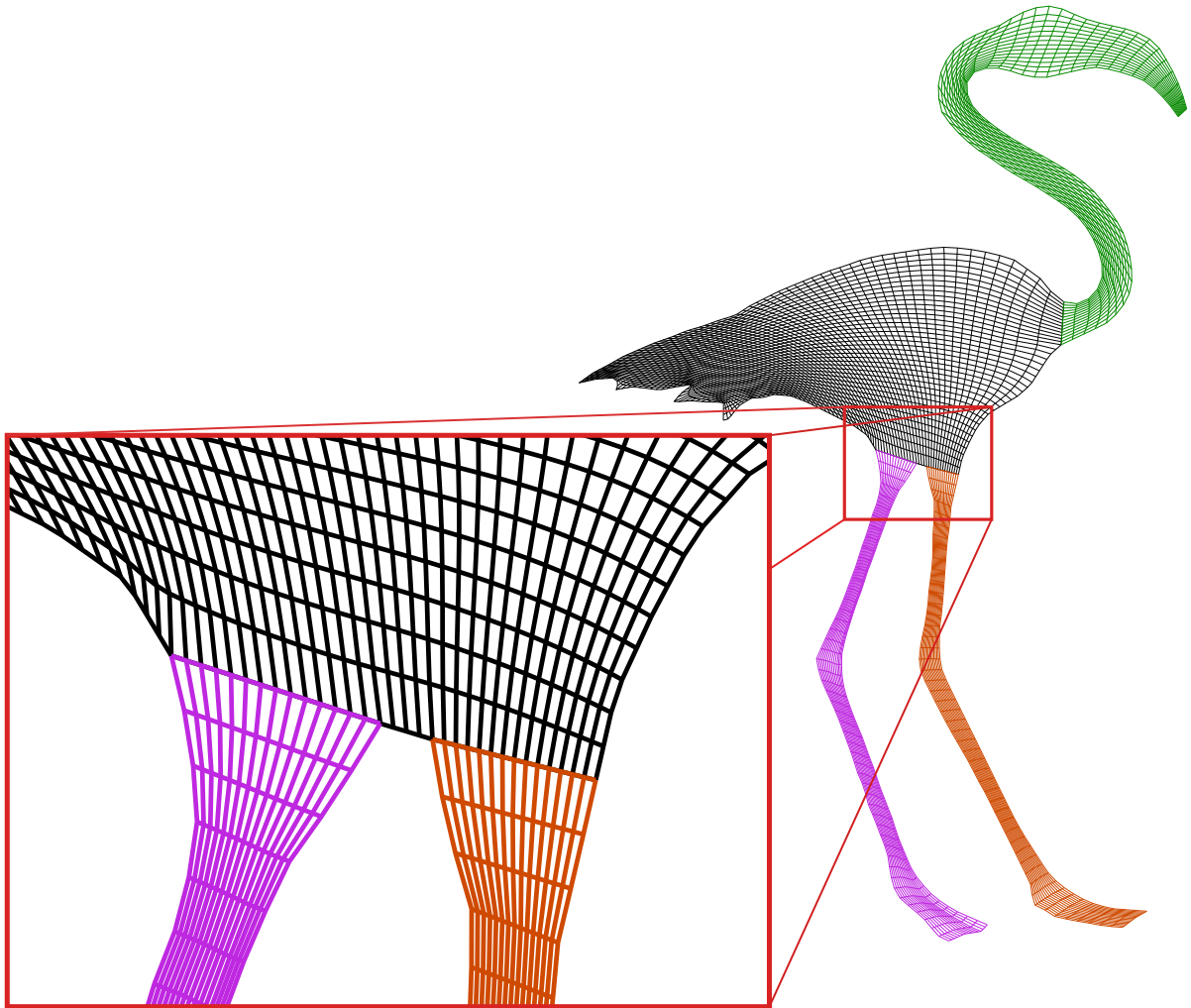


Figura 1.13: Malla estructurada de cuadriláteros por bloques para región flamingo.

Un esquema convencional para generar mallas de cuadriláteros estructuradas por bloques es como sigue:

1. Descomponer la región en subregiones adecuadas.
2. Generar mallas estructuradas compatibles sobre las subregiones.
3. Mejorar la calidad de la malla global.

Un problema indispensable en la generación de mallas estructuradas por bloques es la descomposición de la región en subregiones adecuadas para generar mallas estructuradas. Algunas descomposiciones introducen singularidades en la malla por bloques en puntos interiores donde se intersectan más de dos subregiones [7], el problema es determinar dónde colocarlas de modo que la malla tenga la calidad deseada, más aún, se recomienda generar mallas con el mínimo número de singularidades.

Varios métodos de la literatura generan mallas estructuradas por bloques con singularidades interiores sobre regiones planas, estos métodos usan principalmente las siguientes ideas:

- *Eje Medial*. Separamos la región por una aproximación del eje medial y generamos mallas estructuradas en cada subregión con celdas que empatan en las intersecciones, introduciendo singularidades en puntos del eje medial [6, 130], Fig. 1.14. La dificultad es construir una aproximación adecuada del eje medial de regiones complicadas.

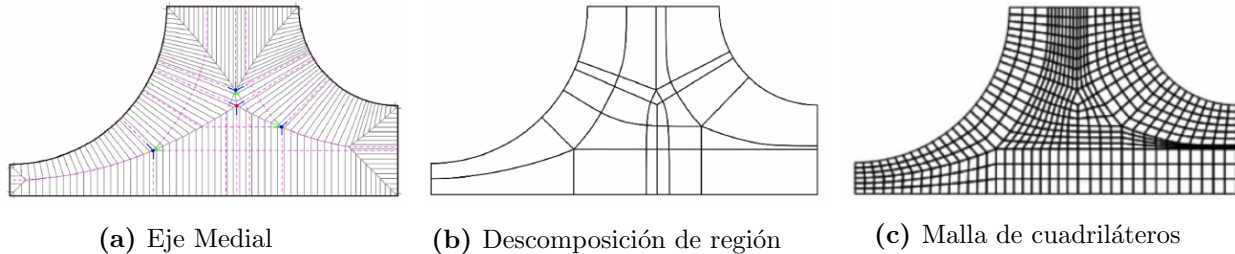


Figura 1.14: Generación de una malla estructurada por bloques mediante el eje medial [128].

- *Campo de Direcciones*. La familia de segmentos de recta que pasan por una colección de puntos es un campo de direcciones si las pendientes de los segmentos satisfacen una ecuación diferencial dada en esos puntos. Construimos un campo de direcciones que se alinea con la geometría de la región para separar la región en subregiones, donde generamos mallas estructuradas con celdas que empatan en las intersecciones. El campo de direcciones se puede obtener resolviendo ecuaciones diferenciales parciales [144] y el método se puede automatizar para generar mallas de alta calidad [32, 65], Fig. 1.15, sin embargo se pueden generar algunas singularidades interiores y puede ser difícil generar el campo de direcciones sobre regiones complicadas.

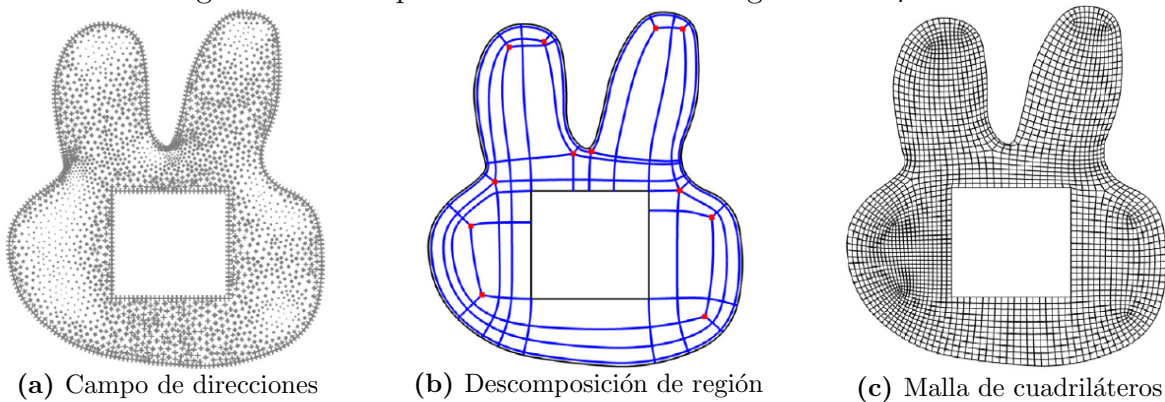
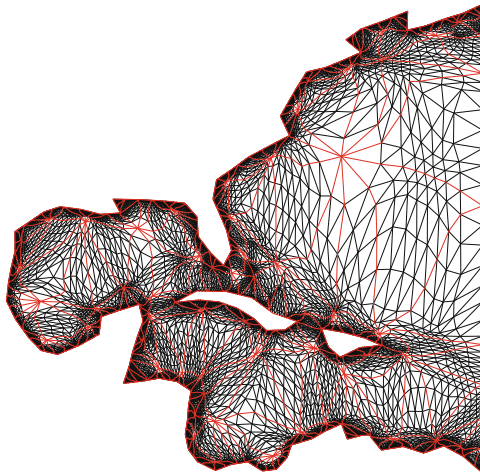
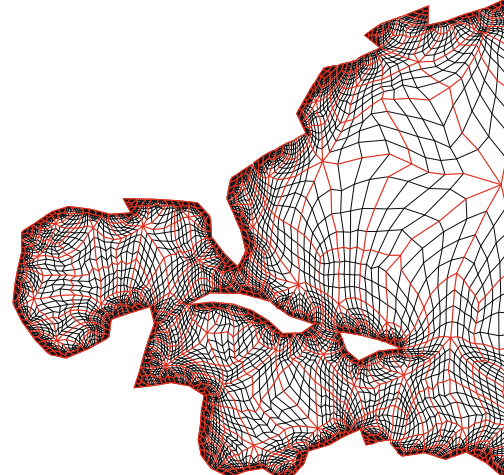


Figura 1.15: Generación de una malla de cuadriláteros estructurada por bloques sobre región con un agujero mediante un campo de direcciones [144].

- *Conversión de triángulos en cuadriláteros*. Generamos mallas de triángulos, que simplificamos para separar la región en bloques y las recombina mos para generar cuadriláteros [149, 148]. Este procedimiento es automático, pero puede introducir varias singularidades, Fig. 1.16, para remediar esto, Zhang et al. [149] separan la región en bloques que son aproximadamente rectángulos mediante triangulaciones de Delaunay con adaptividad en la frontera, lo que a su vez permite generar mallas de alta calidad, Fig. 1.17.



(a) Malla de triángulos por bloques.



(b) Malla de cuadriláteros.

Figura 1.16: Conversión de una malla de triángulos a malla de cuadriláteros estructurada por bloques para una región del Océano Atlántico [148].

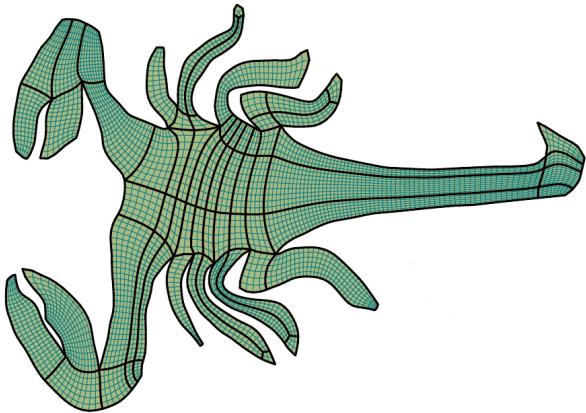


Figura 1.17: Malla estructurada por bloques generada con método de Zhang et al. [149].

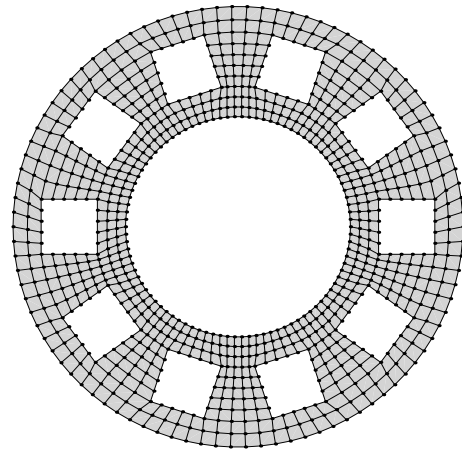
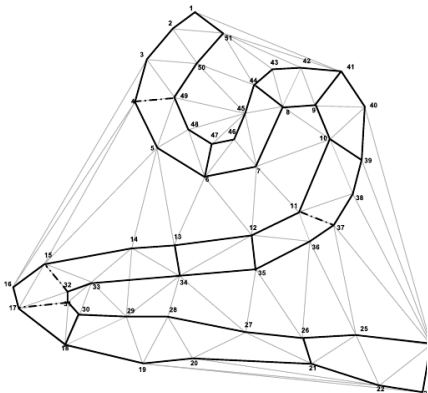
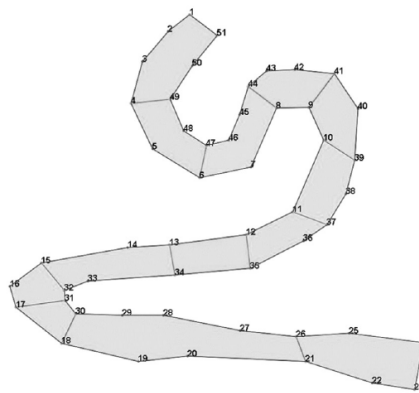


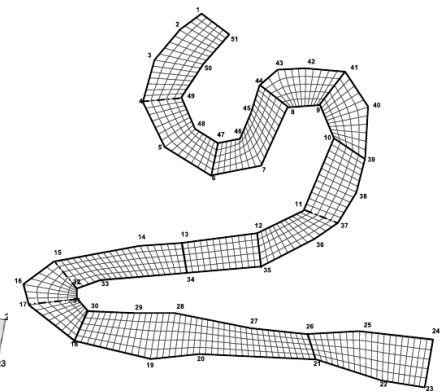
Figura 1.18: Malla estructurada por bloques generada por submapping [116].



(a) Triangulación de Delaunay



(b) Descomposición por cortes



(c) Malla estructurada

Figura 1.19: Generación de una malla de cuadriláteros estructurada por bloques sin singularidades interiores por método de extracción y avance [153].

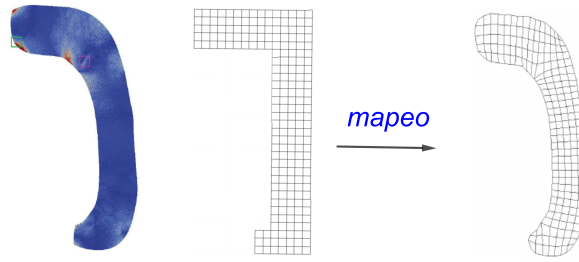


Figura 1.20: Mapeo de *región rectilínea* a región poligonal en azul. La malla de rectángulos se transforma en una malla de cuadriláteros sin singularidades interiores [72, 73].

Armstrong et al. [7] plantean que métodos para generación de mallas como el *eje medial*, el campo de direcciones y *Paving* [17] emplean una misma filosofía que se desarrolla de diferentes maneras, en cualquier caso pueden generar singularidades interiores. Hay otros métodos que pueden generar mallas de cuadriláteros sin estas singularidades:

- *Submapping*. En algunas regiones poligonales podemos clasificar los vértices por su ángulo y conectarlos por cortes para separar la región en subregiones, donde generamos mallas estructuradas. Resolvemos un problema de programación entera con base a la clasificación de vértices para garantizar que las celdas empaten lado a lado entre los bloques [116]. Este método se puede extender a regiones con agujeros, sin embargo su uso se limita a regiones sencillas, Fig. 1.18.
- *Avance y Extracción*. Regiones que poseen una estructura de árbol se separan por cortes en un bloque principal y en bloques que representan las ramas mediante una triangulación de Delaunay, luego mallas estructuradas compatibles se generan en cada bloque usando una rejilla que cubre la región. Este método puede generar mallas de alta calidad y se puede implementar de manera automática [153], pero su alcance se limita sólo a algunas regiones sin agujeros, Fig. 1.19.
- *Mapeos sobre Regiones Rectilíneas*. Representamos regiones poligonales complicadas por *regiones rectilíneas*, donde la frontera es la unión de segmentos horizontales y verticales. Los mapeos de regiones rectilíneas a poligonales pueden verse como una malla estructurada por bloques, Fig. 1.20, para construirlos podemos separar la región en subregiones y simplificar las subregiones en rectilíneas para construir sus parametrizaciones mediante mallas estructuradas. Liu et al. [72] resuelven un problema de coloración de gráficas para garantizar que las celdas empaten lado a lado entre los bloques. Más detalles se muestran en el Apéndice B.

Usualmente, la generación de mallas puede necesitar intervención manual del usuario sobre regiones complicadas [26]. Generadores de mallas interactivos de propósito general como **GridPro** [100], **Gmsh** [109] y **Pointwise** [97] le permiten al usuario generar mallas de alta calidad con algunas singularidades interiores, Fig.1.21. En algunas aplicaciones específicas se han desarrollado generadores de mallas estructuradas por bloques: **PADRAM** [125] para diseño de turbomaquinaria, **NNW-GridStar** [77] para simulaciones de flujo alrededor de aeronaves. Estos programas pueden generar mallas con singularidades interiores.

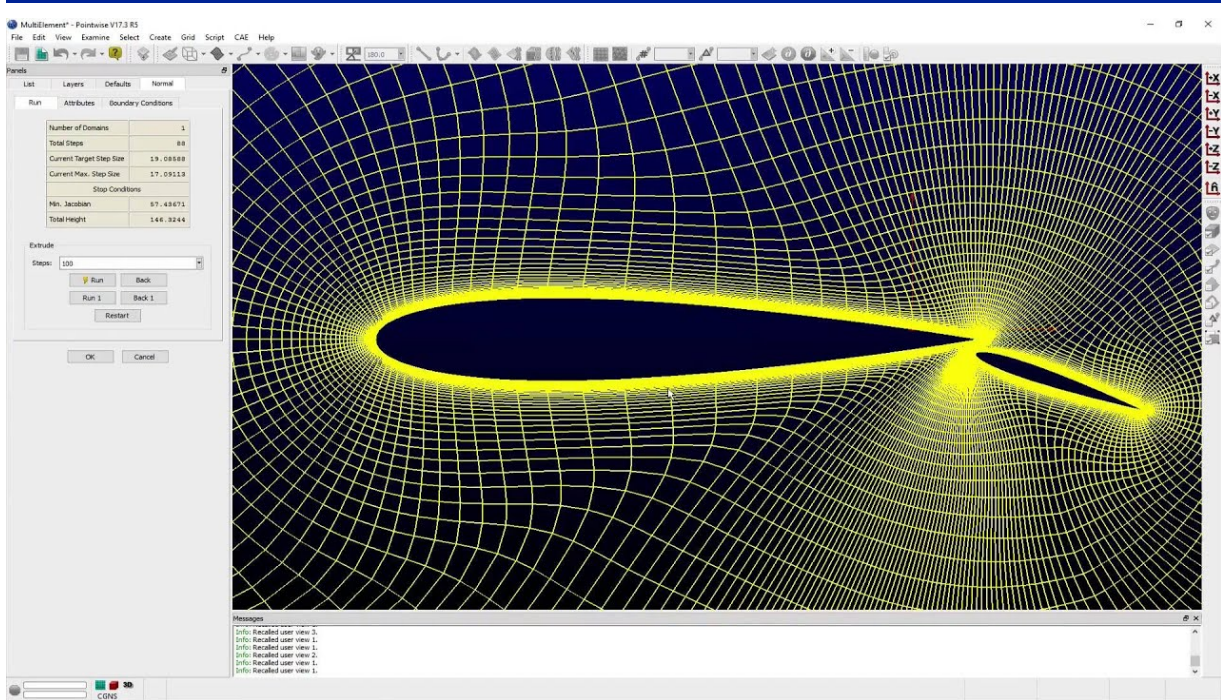


Figura 1.21: Interfaz del generador de mallas Pointwise [97].

Por otra parte, software como CCHE-MESH [151, 152], GridGen [118] y LaCàN [114] pueden generar mallas sin singularidades interiores sobre algunas regiones poligonales que pueden tener agujeros, no obstante tienen limitaciones sobre regiones complicadas y requieren usuarios con experiencia, por lo que es de interés desarrollar un generador de mallas de propósito general que pueda generar mallas de cuadriláteros estructuradas por bloques de alta calidad sin singularidades interiores sobre regiones complicadas.

1.3. Logros y Contribuciones

Nos enfocamos en aspectos prácticos de la generación de mallas, proporcionamos herramientas útiles para que científicos e ingenieros puedan generar mallas con conectividad y propiedades geométricas deseables que requieren algunos problemas de cómputo científico sobre regiones complicadas. Las principales contribuciones de este proyecto son las siguientes:

- Desarrollamos un método para la generación de mallas de cuadriláteros de alta calidad sin singularidades interiores, que se publicó en nuestro artículo *Parametrization of Plane Irregular Regions: A Semi-Automatic Approach I* [12] para regiones sin agujeros. En el presente escrito extendemos la metodología a regiones con agujeros.
- Elaboramos métodos robustos para simplificación de polígonos y reducción de ruido, a saber, los criterios modificados de áreas y radios, así como el método modificado de tres pasos que describimos en §3.3.4. En particular, nuestro criterio de los radios y su modificación son nuevos en la literatura.

- Probamos en §5.2 que podemos construir mallas estructuradas compatibles sobre las subregiones de una descomposición por cortes. La idea es formular la elección de tamaños compatibles como un problema de programación entera.
- Creamos una nueva versión de nuestro software, UNAMalla 6, para generación interactiva de mallas estructuradas por bloques, que puede descargarse en:

<http://tikhonov.fciencias.unam.mx/unamalla/>

En la página se muestran ejemplos y se indican las instrucciones para el uso de nuestro software, de esta manera podemos ver nuestras ideas en acción y aplicarlas en la práctica. Más detalles del software se muestran en el Apéndice A.

Varios métodos de la literatura generan mallas de cuadriláteros con varias singularidades interiores. Por lo que no se tiene control sobre la estructura de la malla. En cambio, nosotros generamos mallas de cuadriláteros con una conectividad uniforme. Nuestras mallas se separan en bloques de mallas estructuradas. A diferencia de otros trabajos que generan mallas estructuradas por bloques [128, 144, 149, 148], generamos mallas sin singularidades interiores. Si bien encontramos algunos métodos para generar estas mallas [72, 116, 153], tienen limitaciones sobre regiones complicadas con agujeros.

Nuestra metodología extiende los métodos de nuestro grupo UNAMALLA [138] e incorporamos otros nuevos, para mostrar que es robusta, generamos mallas de alta calidad sobre una colección de regiones poligonales con diferentes grados de dificultad.

Generamos mallas usando nuestro propio software, a saber, UNAMalla 6. Cabe señalar que este software es la implementación de nuestra metodología. La mayoría de los generadores de mallas construyen triangulaciones. Algunos generadores de mallas pueden generar todas las celdas de cuadriláteros, y entre estos programas solamente unos pocos pueden generar mallas estructuradas por bloques [100, 109, 97]. Incluso el software conocido para generar mallas sin singularidades interiores tiene limitaciones [151, 118, 114]. En cambio, UNAMalla 6 le permite al usuario generar mallas de alta calidad sin singularidades interiores sobre regiones complicadas.

1.4. Método para Generación de Mallas

Queremos mallas de cuadriláteros sin singularidades interiores, una manera es generar mallas estructuradas por bloques. La idea principal es separar por cortes una región e identificar los puntos de las subregiones que correspondan a las esquinas del cuadrado unitario. La elección adecuada de los cortes y las esquinas es clave para generar mallas de alta calidad.

Nuestro método para generar mallas estructuradas por bloques se desarrolla por etapas consecutivas. Las regiones complicadas se simplifican preservando la forma original y se descomponen en subregiones adecuadas para generar mallas estructuradas, luego construimos mapeos del cuadrado unitario a la frontera de cada subregión que sean compatibles entre las subregiones. Estos mapeos se extienden al interior de las subregiones como mallas

estructuradas que unimos y suavizamos para obtener la calidad deseada. Para ilustrar este método usamos la región del Golfo de México que mostramos en la Fig. 1.22.

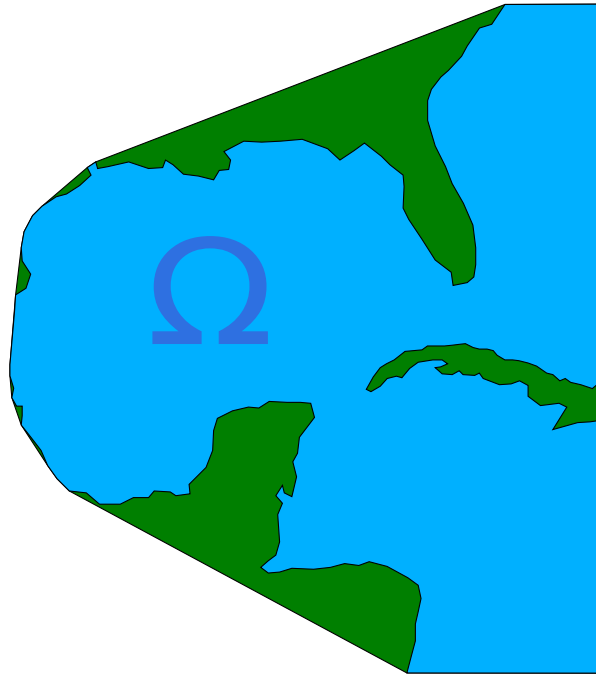
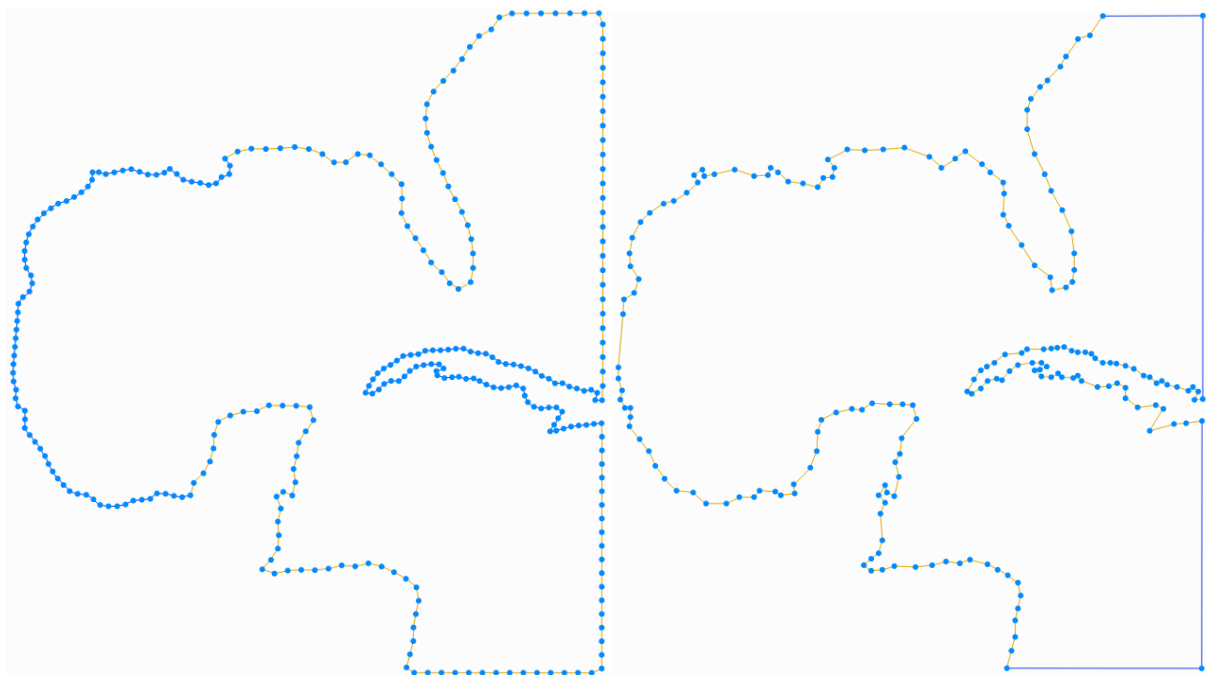


Figura 1.22: Región: Golfo de México



(a) Frontera Original

(b) Aproximación

Figura 1.23: Reducción de 316 puntos a 200 puntos del contorno del Golfo de México.

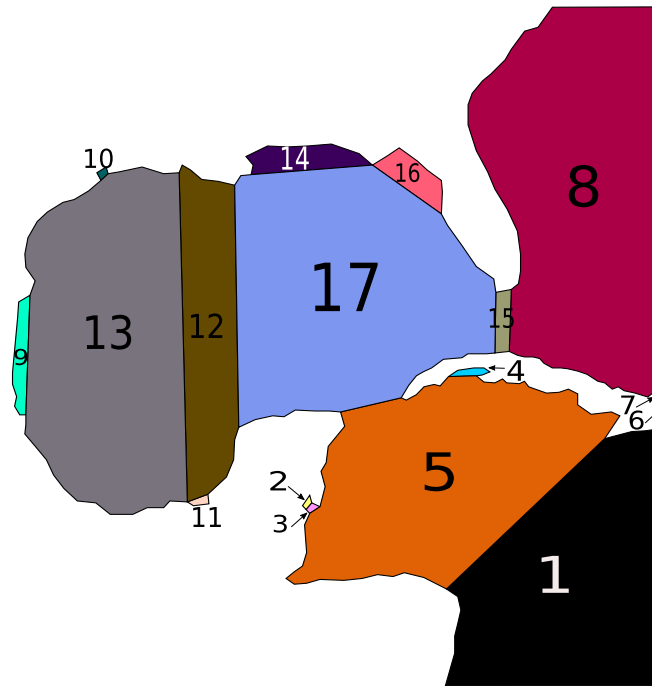


Figura 1.24: *Descomposición admisible* del Golfo de México en 17 subregiones.

1. **Aproximación de la región.** La frontera de la región se aproxima por curvas poligonales que preservan la forma de la región con una menor cantidad de puntos, Fig. 1.23.
2. **Descomposición de la región.** La región se separa por cortes en subregiones sin agujeros ni ángulos pequeños de modo que sean aproximadamente convexas, Fig. 1.24.
3. **Construcción de parametrizaciones compatibles.** En cada subregión identificamos dos parejas de curvas opuestas y los cortes de la región. Las curvas opuestas se parametrizan con el mismo número de puntos de modo que coincidan en los cortes, Fig. 1.25.
4. **Extensión por generación de mallas.** Generamos mallas estructuradas sobre las subregiones con puntos frontera dados por las curvas compatibles, unimos y suavizamos estas mallas para mejorar su calidad, Fig. 1.26.

En los siguientes capítulos reformulamos nuestro problema y desarrollamos las etapas de nuestro método, a continuación damos una breve descripción de su contenido:

- En el Capítulo 2 formulamos la generación de mallas estructuradas por bloques como la construcción de parametrizaciones, presentamos algunos trabajos previos de nuestro grupo UNAMALLA y resaltamos la calidad de nuestras mallas.

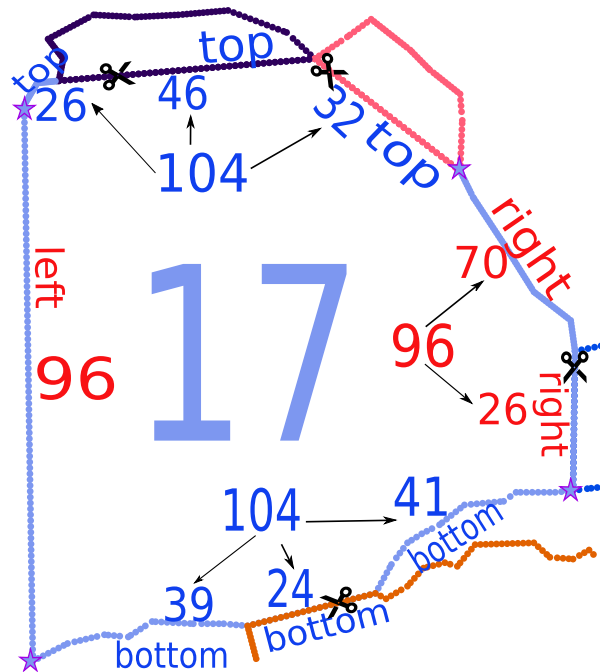


Figura 1.25: La frontera de la región 17 del Golfo de México se separa en cuatro curvas: top y bottom tienen 104 puntos, mientras que las curvas left y right tienen 96 puntos. Marcamos los cortes con tijeras y las esquinas donde se intersectan las curvas por estrellas.

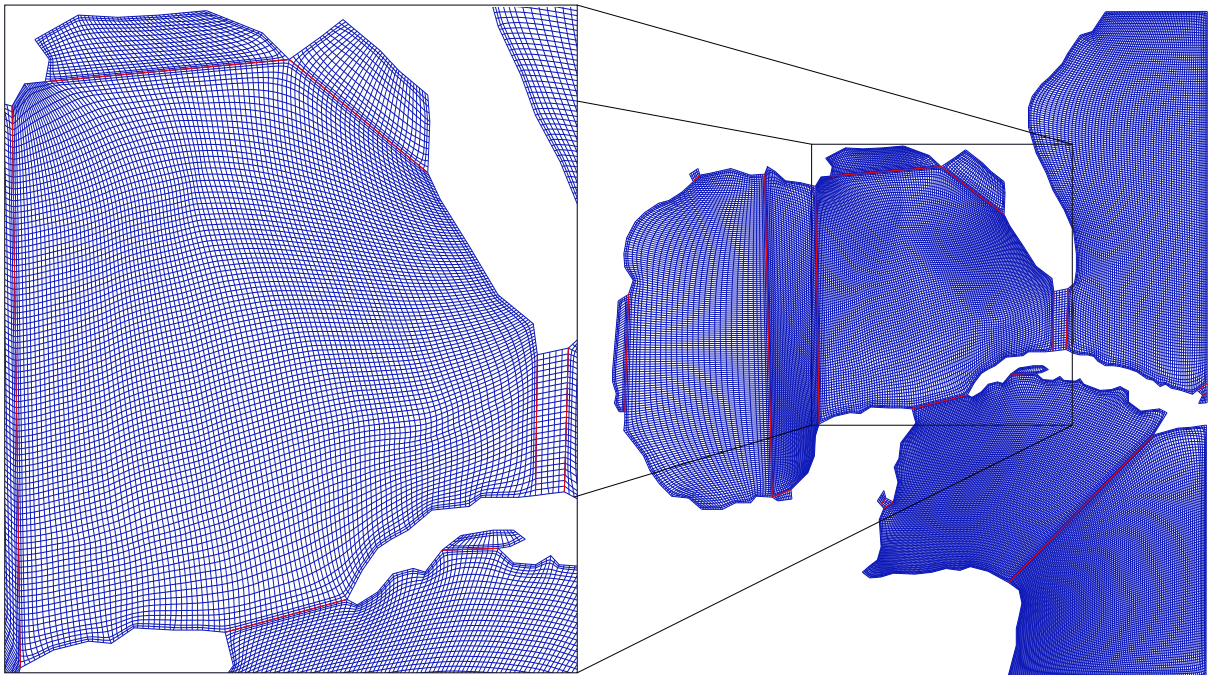


Figura 1.26: Malla de cuadriláteros estructurada por bloques sobre el Golfo de México.

- En el Capítulo 3 aproximamos la frontera de la región (Paso 1). Revisamos algunos antecedentes y presentamos nuevos métodos para simplificación y reducción de ruido de polígonos.
- En el Capítulo 4 desarrollamos un método para separar las regiones en subregiones admisibles para la generación de mallas estructuradas (Paso 2).
- En el Capítulo 5 construimos parametrizaciones compatibles para las fronteras de las subregiones (Paso 3), presentamos criterios para la elección de esquinas y formulamos la elección de tamaños compatibles como un problema factible de programación entera.
- En el Capítulo 6 generamos mallas estructuradas para extender las parametrizaciones de las fronteras, las unimos en una malla global, que suavizamos siguiendo el enfoque variacional discreto [11, 55] (Paso 4).

En los apéndices damos una breve descripción del sistema computacional UNAMalla 6 y planteamos la construcción de parametrizaciones sobre *regiones rectilíneas*.

1.5. Resumen del Capítulo

Una malla de cuadriláteros sobre una región es un ensamblaje de cuadriláteros que empatan lado a lado y respetan la frontera de la región, sus puntos interiores son singularidades si no se conectan exactamente con cuatro puntos. La malla es estructurada si no tiene singularidades, de otro modo es estructurada por bloques si se puede separar en submallas estructuradas. Cuando un alto porcentaje de sus cuadriláteros son aproximadamente rectángulos, decimos que la malla tiene alta calidad. Una revisión somera de la generación de mallas de cuadriláteros se puede consultar en Boomes et al [14].

Nuestro problema es generar mallas de cuadriláteros sobre regiones poligonales complicadas de modo que sean de alta calidad y no tengan singularidades interiores, reformulamos este problema como la generación de mallas estructuradas por bloques. Revisamos algunos métodos de la literatura, no obstante, pueden generar singularidades interiores. Por lo que desarrollamos nuestro método, que describimos de manera breve en los siguientes pasos:

1. Simplificar la frontera de la región complicada
2. Descomponer por cortes la región en *subregiones admisibles*
3. Construir *parametrizaciones compatibles* de los lados del cuadrado unitario a la fronteras de las subregiones.
4. Extender las parametrizaciones de las fronteras mediante generación de mallas estructuradas de alta calidad

El aporte principal de este proyecto es la implementación del método anterior en la nueva versión de nuestro generador de mallas, llamado **UNAMalla 6**, que le permite a los usuarios generar sus propias mallas y ver nuestro método en acción.

Antes de describir cada paso de nuestro método, en el siguiente capítulo veremos que las mallas estructuradas por bloques pueden pensarse como familias de mapeos del cuadrado unitario a los bloques, por lo que la generación de estas mallas es la construcción de parametrizaciones para las regiones.

PARAMETRIZACIONES POR MALLAS ESTRUCTURADAS

Las mallas que tienen la estructura y geometría adecuadas pueden pensarse como mapeos que representan regiones complicadas como regiones sencillas, conocidos en la literatura como parametrizaciones de regiones [47, 126], Fig. 2.1.

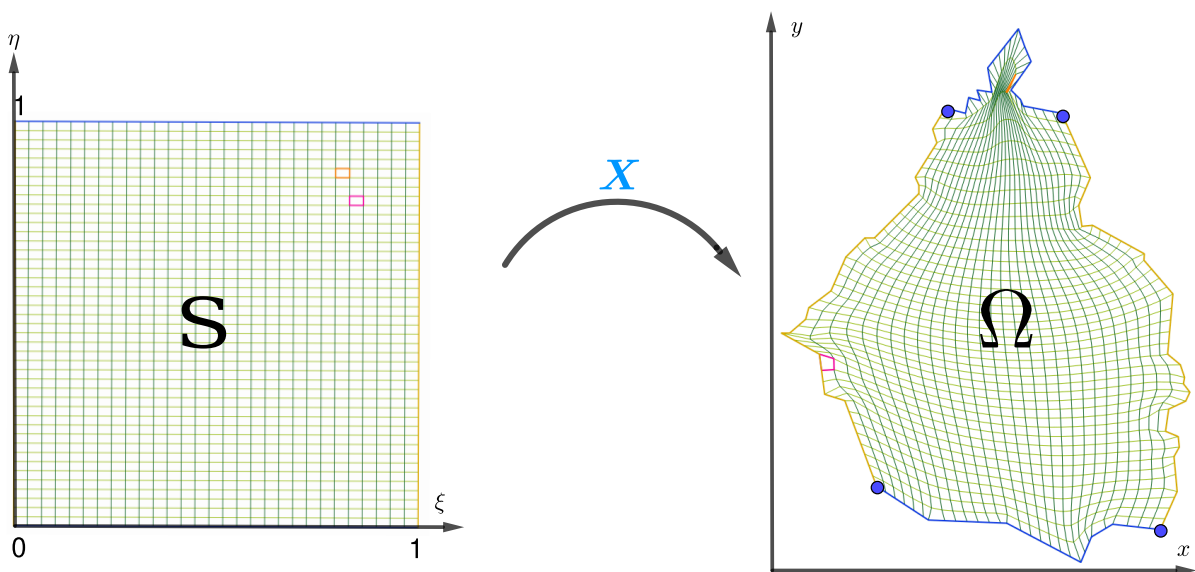


Figura 2.1: Parametrización \mathbf{X} del cuadrado unitario S en el plano $\xi\eta$ a la Ciudad de México Ω en el plano xy . La malla sobre la Ciudad de México es la imagen de la rejilla sobre un cuadrado.

Las regiones planas que usamos son regiones poligonales, a saber, conjuntos cerrados y acotados Ω de \mathbb{R}^2 con fronteras dadas por una colección de polígonos simples mutuamente ajenos P_0, \dots, P_m de modo que P_0 encierra a P_1, \dots, P_m . La región Ω es *simplemente conexa* si $m = 0$. De otro modo, Ω es una *región múltiplemente conexa* ($m + 1$ conexa). La *frontera exterior* de Ω es P_0 . Las *fronteras interiores* de Ω son P_1, \dots, P_m . Los *agujeros* de Ω son las regiones simplemente conexas acotadas por las fronteras interiores, Fig. 2.2.

Una parametrización de la región Ω es un *homeomorfismo* de una región plana R a la región Ω , esto es, una función continua e invertible $\mathbf{X} : R \rightarrow \Omega$ tal que su función inversa $\mathbf{X}^{-1} : \Omega \rightarrow R$ es a su vez una función continua. Si además, \mathbf{X} y su función inversa son continuamente diferenciables, la parametrización \mathbf{X} se llama *difeomorfismo*. El dominio R de \mathbf{X} se llama *región lógica* y la imagen Ω de \mathbf{X} se llama *región física*.

La parametrización \mathbf{X} y su función inversa son cambios de coordenadas de la región lógica a la región física, y viceversa. Las líneas curvilíneas que atraviesan la región lógica se transforman en líneas que atraviesan la región física, generando mallas sobre ambas subregiones, por lo que mallas sobre la región lógica se transforman en mallas sobre la región física, Fig. 2.1. De esta manera, la generación de mallas se puede pensar como la construcción de parametrizaciones.

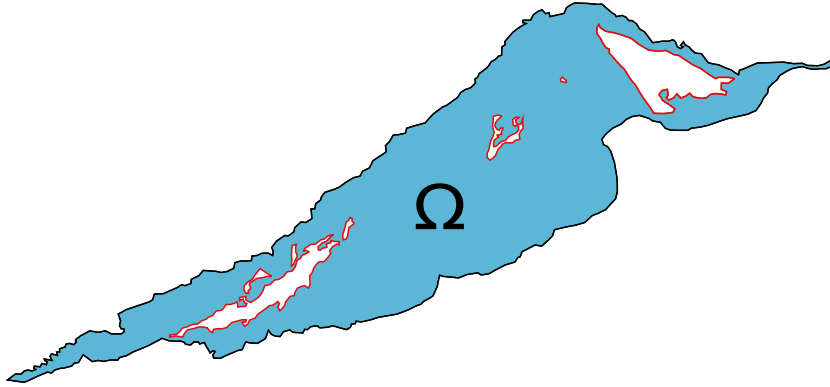


Figura 2.2: La Laguna de Tamiahua en Veracruz es una región con siete agujeros en color rojo.

Para construir una parametrización de la región Ω necesitamos elegir una región lógica R homeomorfa a Ω y hallar un mapeo \mathbf{X} que respete la frontera, en el sentido de que el mapeo \mathbf{X} transforma ∂R en $\partial\Omega$, Fig. 2.1, más aún, si la región Ω es múltiplemente conexa, el mapeo \mathbf{X} debe transformar agujeros de R en agujeros de Ω . Así que formulamos la construcción de parametrizaciones como sigue:

Problema 2.1 (Parametrización Global). Sean Ω una región poligonal $m + 1$ conexa. Hallar una región $m + 1$ conexa de \mathbb{R}^2 , denotada por R , y un homeomorfismo $\mathbf{X} : R \rightarrow \Omega$ tal que $X(\partial R) = \partial\Omega$.

2.1. Parametrización del Cuadrado Unitario a la Región

Una manera para construir parametrizaciones de regiones es generar mallas que posean una conectividad uniforme, a saber, las mallas estructuradas, pues se pueden visualizar como una rejilla uniforme sobre el cuadrado unitario $S = [0, 1] \times [0, 1]$. Esto nos sugiere elegir S como la región lógica y construir mapeos de S a regiones poligonales mediante mallas estructuradas. Primero, construimos las parametrizaciones de regiones simplemente conexas.

2.1.1. Parametrización de Regiones Simplemente Conexas

Sea Ω una región poligonal acotada sin agujeros, extendemos una parametrización de su frontera como un mapeo de S a Ω .

Extensión de la parametrización. Separamos la frontera de la región Ω en cuatro curvas consecutivas que no se traslapan $\Omega_b, \Omega_r, \Omega_t$ y Ω_l , y construimos parametrizaciones de los lados del cuadrado unitario S a estas curvas:

$$\begin{aligned} \mathbf{b}_b : \{(\xi, 0) : 0 \leq \xi \leq 1\} &\rightarrow \Omega_b, & \mathbf{b}_l : \{(0, \eta) : 0 \leq \eta \leq 1\} &\rightarrow \Omega_l, \\ \mathbf{b}_t : \{(\xi, 1) : 0 \leq \xi \leq 1\} &\rightarrow \Omega_t, & \mathbf{b}_r : \{(1, \eta) : 0 \leq \eta \leq 1\} &\rightarrow \Omega_r. \end{aligned}$$

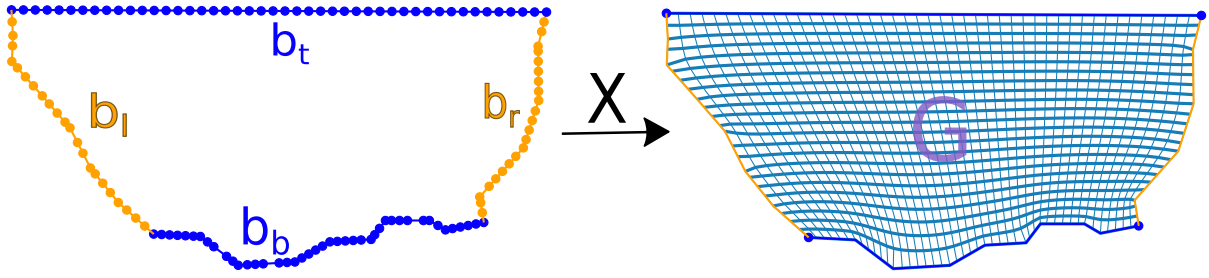


Figura 2.3: Extensión X de las parametrizaciones b_b , b_r , b_t y b_l al interior de la región mediante una malla estructurada de cuadriláteros G .

Las curvas Ω_b y Ω_t son fronteras opuestas, en el sentido de que tienen parametrizaciones sobre lados opuestos del cuadrado unitario, lo mismo decimos de las curvas Ω_l y Ω_r . Nuestro problema es extender estas parametrizaciones al interior de la región, Fig. 2.3.

Problema 2.2. (Extensión sobre el cuadrado unitario). Sea Ω una región poligonal acotada con frontera parametrizada por las curvas b_b , b_t , b_l y b_r , hallar un homeomorfismo $X : (\xi, \eta) \in S \mapsto (x, y) \in \Omega$ tal que

$$\begin{aligned} X|_{\eta=0} &= b_b, & X|_{\eta=1} &= b_t, \\ X|_{\xi=0} &= b_l, & X|_{\xi=1} &= b_r. \end{aligned} \quad (2.1)$$

Parametrizamos las fronteras opuestas con el mismo número de puntos para extenderlas al interior de la región, las curvas b_b y b_t son reparametrizaciones de los lados Ω_b y Ω_t por longitud de arco con m puntos; mientras que las curvas b_l y b_r son reparametrizaciones de los lados Ω_l y Ω_r por longitud de arco con n puntos, respectivamente.

Mallas Estructuradas. Extendemos las parametrizaciones de las fronteras mediante una malla estructurada de cuadriláteros G de tamaño $m \times n$ sobre la región Ω , que consiste en una tripleta (V, E, F) , donde:

- V es el conjunto de puntos $P_{i,j} \in \Omega$ para $i = 1, \dots, m, j = 1, \dots, n$,
- E es el conjunto de segmentos de recta $P_{i,j}P_{i+1,j}$ y $P_{i,j}P_{i,j+1}$,
- F es el conjunto de cuadriláteros $Q_{i,j} = P_{i,j}P_{i+1,j}P_{i+1,j+1}P_{i,j+1}$.

Los puntos frontera de la malla G son los puntos $P_{i,j}$ con $i \in \{1, m\}$ o $j \in \{1, n\}$, que recorreremos en el siguiente orden:

1. $P_{i,1}, i = 1, \dots, m.$
2. $P_{m,j}, j = 1, \dots, n.$
3. $P_{m-i+1,n}, i = 1, \dots, m.$
4. $P_{m,n-j+1}, j = 1, \dots, n.$

Así, formamos un polígono que preserve la forma y orientación de la región Ω , y que a su vez determina la orientación de los cuadriláteros de la malla.

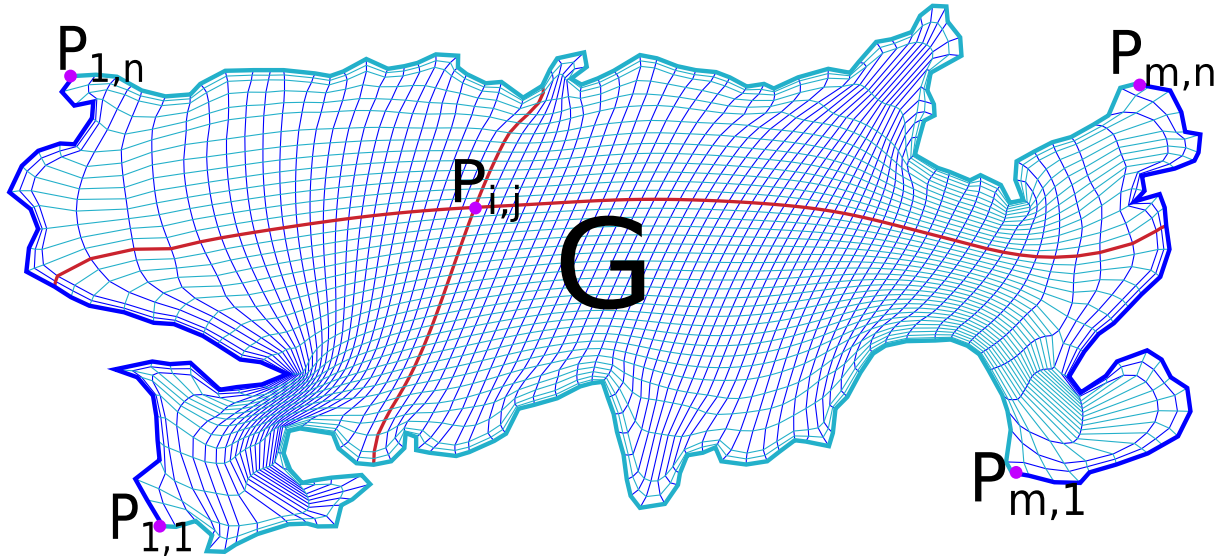


Figura 2.4: Malla estructurada de cuadriláteros de tamaño 80×40 sobre el Lago Titicaca.

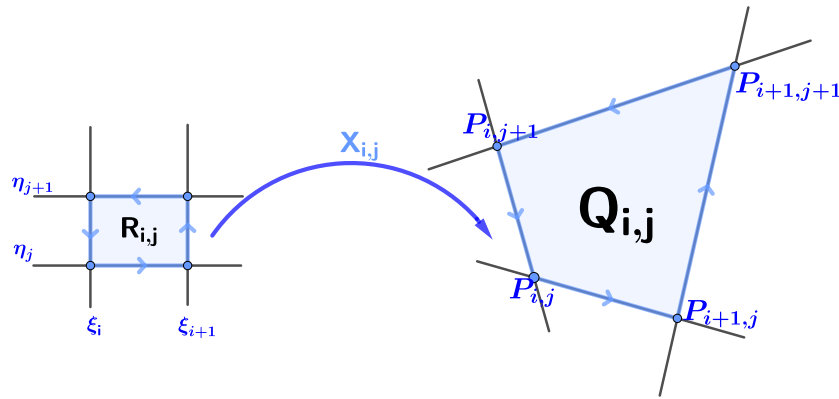


Figura 2.5: Mapeo bilineal $X_{i,j}$ del rectángulo $R_{i,j}$ al cuadrilátero $Q_{i,j}$.

La conectividad de un punto de la malla G es el número de sus segmentos incidentes, los puntos interiores de G tienen conectividad cuatro, mientras que sus puntos frontera tienen conectividad tres, con excepción de los cuatro puntos $P_{1,1}, P_{1,m}, P_{n,1}, P_{m,n}$, llamados *esquinas*, que tienen conectividad dos. Cuando fijamos uno de los índices, formamos una curva poligonal que atraviesa la región conectando lados opuestos, Fig. 2.4.

Mapeos Bilineales. La malla G se puede visualizar como una malla de rectángulos que tiene la misma conectividad sobre el cuadrado unitario S con puntos (ξ_i, η_j) y rectángulos $R_{i,j} = [\xi_i, \xi_{i+1}] \times [\eta_j, \eta_{j+1}]$, donde $\{\xi_i\}_{i=1}^m$ y $\{\eta_j\}_{j=1}^n$ son particiones del intervalo $[0, 1]$, Fig. 2.5. Podemos transformar los rectángulos $R_{i,j}$ en los cuadriláteros $Q_{i,j}$ de la malla G mediante mapeos bilineales, por lo que construimos mapeos de S a Ω de modo que su restricción al rectángulo $R_{i,j}$ sea el mapeo bilineal $X_{i,j} : R_{i,j} \rightarrow Q_{i,j}$ tal que

$$\begin{aligned} X_{i,j}(\xi_i, \eta_j) &= P_{i,j}, & X_{i,j}(\xi_{i+1}, \eta_j) &= P_{i+1,j}, \\ X_{i,j}(\xi_i, \eta_{j+1}) &= P_{i,j+1}, & X_{i,j}(\xi_{i+1}, \eta_{j+1}) &= P_{i+1,j+1}. \end{aligned}$$

Podemos obtener una expresión analítica de la parametrización de Ω , lo que hacemos es parametrizar su frontera en términos de la i -ésima función *B-spline lineal* con sucesión de nodos

$$t_1^\xi = 0, \quad t_2^\xi = \xi_1, \quad \dots, \quad t_{m+1}^\xi = \xi_m, \quad t_{m+2}^\xi = 1$$

definida por

$$B_{i,m}^2(t) = \frac{t - t_i^\xi}{t_{i+1}^\xi - t_i^\xi} B_{i,m}^1(t) + \frac{t_{i+2}^\xi - t}{t_{i+2}^\xi - t_{i+1}^\xi} B_{i+1,m}^1(t), \quad (2.2)$$

donde

$$B_{i,m}^1(t) = \begin{cases} 1, & \text{si } t_i^\xi \leq t < t_{i+1}^\xi, \\ 0, & \text{de otro modo,} \end{cases} \quad i = 1, \dots, m. \quad (2.3)$$

La j -ésima función *B-spline lineal* con sucesión de nodos

$$t_1^\eta = 0, \quad t_2^\eta = \eta_1, \quad \dots, \quad t_{n+1}^\eta = \eta_n, \quad t_{n+2}^\eta = 1,$$

denotada por $B_{j,n}^2$, se define análogamente reemplazando i por j , m por n , y t_i^ξ por t_j^η , respectivamente, para $j = 1, \dots, n$.

Las parametrizaciones de los cuatro lados de Ω pueden expresarse como *curvas B-spline lineales* con puntos de control dados por los puntos frontera de la malla G :

$$\begin{aligned} \mathbf{b}_b(\xi) &= \sum_{i=1}^m P_{i,1} B_{i,m}^2(\xi), & \mathbf{b}_t(\xi) &= \sum_{i=1}^m P_{i,n} B_{i,m}^2(\xi), & 0 \leq \xi \leq 1, \\ \mathbf{b}_l(\eta) &= \sum_{j=1}^n P_{1,j} B_{j,n}^2(\eta), & \mathbf{b}_r(\eta) &= \sum_{j=1}^n P_{m,j} B_{j,n}^2(\eta), & 0 \leq \eta \leq 1. \end{aligned} \quad (2.4)$$

Usamos todos los puntos $P_{i,j}$ de la malla G para extender las curvas B-spline lineales al interior de la región Ω como el *producto tensorial B-spline bilineal* $\mathbf{X} : S \rightarrow \Omega$ dado por:

$$\mathbf{X}(\xi, \eta) = \sum_{i=1}^m \sum_{j=1}^n P_{i,j} B_{i,m}^2(\xi) B_{j,n}^2(\eta), \quad \xi, \eta \in [0, 1]. \quad (2.5)$$

Con base a lo anterior, la malla estructurada de cuadriláteros G determina una parametrización bilineal de la región Ω . En la literatura [47], G se conoce como la malla de control del mapeo \mathbf{X} . Consulte el trabajo de Carl De Boor [30] para una introducción detallada a los *B-splines*.

El mapeo bilineal \mathbf{X} debe ser inyectivo para que sea una parametrización de la región Ω , una condición suficiente es que la malla G sea convexa, esto es, que todos sus cuadriláteros sean convexos.

Teorema 2.1 ([1]). *Sea $\mathbf{X} : R \rightarrow \Omega$ el mapeo bilineal dado por (2.5) con malla de control G . Si la malla G es convexa, entonces el mapeo \mathbf{X} es inyectivo.*

En resumen, realizamos los siguientes pasos para construir un mapeo del cuadrado unitario S a la región poligonal simplemente conexa Ω :

1. Separar la frontera de Ω en cuatro curvas.
2. Parametrizar curvas de fronteras opuestas con el mismo número de puntos.
3. Generar una malla estructurada de cuadriláteros sobre Ω tal que sus puntos frontera están dados por la parametrización de la frontera y todas sus celdas sean convexas.

Ejemplo 2.1. Construimos una parametrización bilineal del cuadrado unitario al estado de Zacatecas, Fig. 2.6, para ello separamos la frontera en dos curvas azules y dos curvas naranja, cada una con 50 puntos y las extendemos al interior como una malla estructurada convexa de cuadriláteros de tamaño 50×50 .

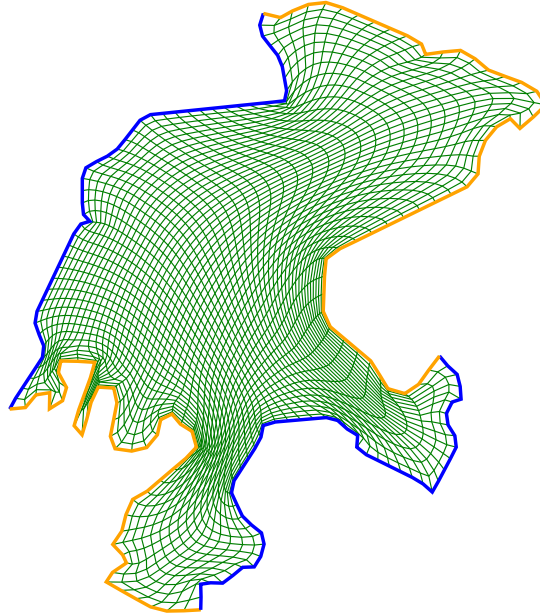


Figura 2.6: Parametrización del estado de Zacatecas por malla estructurada de tamaño 50×50 .

Observaciones 2.1:

☞ El mapeo bilineal entre las regiones se reduce al mapeo correspondiente entre las celdas:

$$\mathbf{X}(\xi, \eta) = \mathbf{X}_{i,j}(\xi, \eta), \quad (\xi, \eta) \in R_{i,j}, \quad i = 1, \dots, m, \quad j = 1, \dots, n,$$

más aún,

$$\mathbf{X}(\xi_i, \eta_j) = P_{i,j}, \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

☞ Además de mapeos bilineales, se pueden construir mapeos bicuadráticos del cuadrado unitario a regiones simplemente conexas delimitadas por *curvas B-spline cuadráticas*, al respecto Abelló et al. [2] desarrollan un método para construirlos y dan condiciones suficientes para garantizar que sean inyectivos.

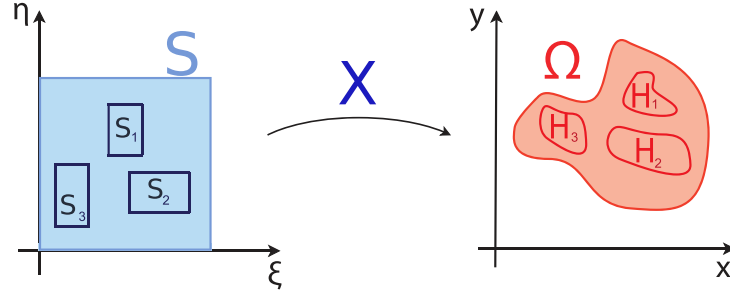


Figura 2.7: Mapeo \mathbf{X} del cuadrado unitario S con agujeros S_i a la región Ω con agujeros H_i .

2.1.2. Parametrización de Regiones Múltiplemente Conexas

En trabajos previos [44] nuestro grupo UNAMALLA ha construido parametrizaciones de regiones con agujeros como mapeos del cuadrado unitario agujerado a las regiones de modo que los agujeros del cuadrado se transformen en agujeros de la región, Fig. 2.7. Para construir estos mapeos extendemos las parametrizaciones de la frontera exterior y las fronteras interiores.

Sea Ω una región poligonal con agujeros H_1, \dots, H_k , y sea S el cuadrado unitario agujerado por rectángulos S_1, \dots, S_k , denotamos por $\partial\Omega_{\text{ext}}$ y ∂S_{ext} a sus respectivas fronteras exteriores. La construcción de una parametrización de Ω se formula como sigue:

Problema 2.3 (Extensión sobre regiones con agujeros). Dados los mapeos:

$$\mathbf{b}_0 : \partial S_{\text{ext}} \rightarrow \partial\Omega_{\text{ext}} \quad \mathbf{b}_i : \partial S_i \rightarrow \partial H_i, \quad i = 1, \dots, k,$$

hallar un homeomorfismo $\mathbf{X} : S \rightarrow \Omega$ tal que

$$\mathbf{X}|_{\partial S_{\text{ext}}} = \mathbf{b}_0, \quad \mathbf{X}|_{\partial S_i} = \mathbf{b}_i, \quad i = 1, \dots, k.$$

En García [44] se reduce el Problema 2.3 al caso de regiones sin agujeros. La idea es separar la región Ω mediante *líneas de corte*, que son segmentos curvilíneos en el interior de la región con extremos en puntos de la frontera. Para separar la región Ω conectamos las fronteras interiores con la frontera exterior por líneas de corte como sigue:

1. Conectamos $\partial\Omega_{\text{ext}}$ con el agujero H_1 más cercano mediante una línea de corte c_1 .
2. Si hay otros agujeros, conectamos H_1 con el agujero H_2 más cercano por otra línea de corte c_2 , hasta conectar todos los agujeros H_1, \dots, H_k con líneas de cortes c_1, \dots, c_k .
3. Conectamos el agujero H_k con $\partial\Omega_{\text{ext}}$ por otra línea de corte c_{k+1} .

Así, Ω se separa en subregiones Ω_1 y Ω_2 , que se intersectan en las líneas de corte, Fig. 2.8.

Generamos mallas estructuradas convexas G_1 y G_2 de tamaño $m \times n$ sobre Ω_1 y Ω_2 , de modo que sus puntos frontera coincidan en cada línea de corte c_1, \dots, c_k , sus esquinas sean los extremos de c_1 y c_{k+1} , tengan m puntos sobre c_1 y c_{k+1} , y que la suma del número de puntos sobre c_2, \dots, c_{k-1} sea menor que n . Así, la unión de G_1 y G_2 es una malla estructurada convexa G de tamaño $m \times 2n - 1$ sobre la región Ω , y por consiguiente el mapeo deseado de S a Ω se puede obtener como el mapeo bilineal (2.5) con malla de control dada por G .

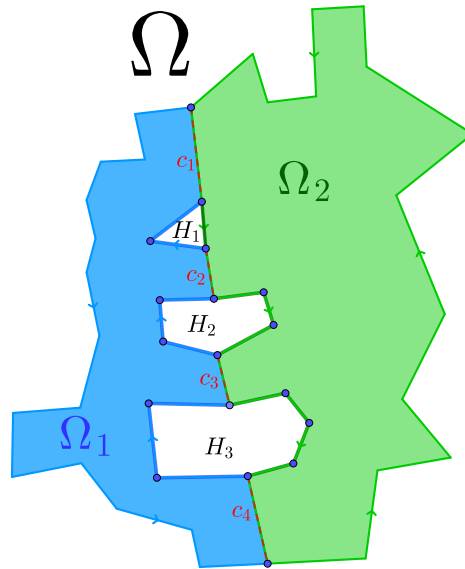


Figura 2.8: La región Ω con tres agujeros H_1, H_2 y H_3 se separa en dos subregiones Ω_1 y Ω_2 conectando la frontera exterior con los agujeros mediante líneas de corte c_1, c_2, c_3 y c_4 .

El método anterior puede generar mallas estructuradas sobre regiones complicadas con agujeros como se muestra en el Ejemplo 2.2, sin embargo las mallas generadas pueden no tener la calidad deseada como hacemos notar en el Ejemplo 2.8, por lo que investigamos otro método para generar mallas con mejor calidad.

Ejemplo 2.2. En García [44] se construye un mapeo bilineal del cuadrado unitario S con dos agujeros S_1 y S_2 al Lago Toba Ω con dos agujeros H_1 y H_2 , para ello separa Ω en dos subregiones sin agujeros mediante tres líneas de cortes y genera mallas estructuradas de cuadriláteros sobre cada subregión que se unen en una malla global, Fig. 2.9.

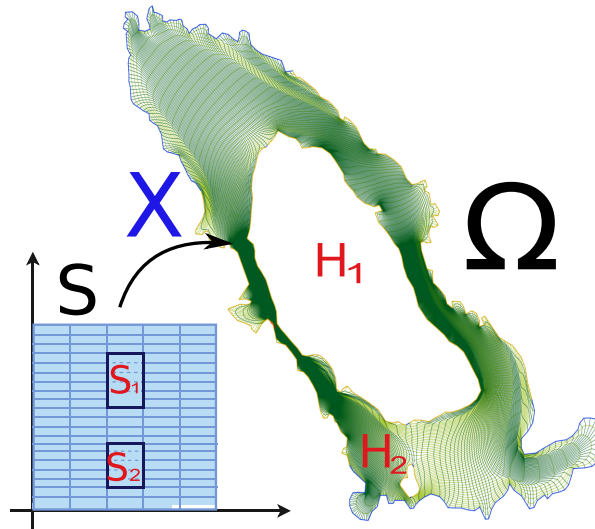


Figura 2.9: Mapeo X del cuadrado S con dos agujeros S_i al Lago Toba Ω con dos agujeros H_i .

2.2. Parametrizaciones Admisibles

En la literatura se han generado mallas estructuradas convexas de cuadriláteros sobre regiones poligonales complicadas [8, 11, 10, 33, 44, 55, 54, 62], sin embargo varias de sus celdas pueden estar elongadas, e incluso distorsionadas, como se muestra en las figuras 1.11, 2.4 y 2.9, por lo que estas mallas no son adecuadas para las simulaciones computacionales.

En cambio, algunas aplicaciones requieren *mallas de cuadriláteros de alta calidad*, en el sentido de que un alto porcentaje de sus cuadriláteros tengan la geometría deseada. Hay diferentes maneras para medir la calidad de una malla, dependiendo de las propiedades geométricas sus cuadriláteros, a nosotros nos interesa que un alto porcentaje de los cuadriláteros sean aproximadamente rectángulos. González [46] hace una revisión somera de las medidas de calidad para mallas de cuadriláteros.

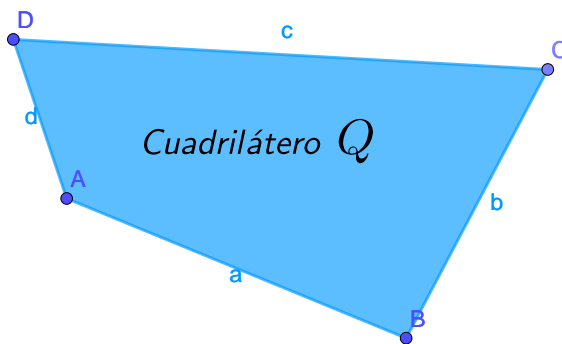


Figura 2.10: Cuadrilátero Q con vértices A, B, C, D y lados de longitudes a, b, c, d .

Para formalizar el concepto de la calidad para una malla de cuadriláteros empleamos una medida que nos indica cuando un cuadrilátero es aproximadamente un rectángulo. Sea Q un cuadrilátero con vértices A, B, C, D y lados de longitud a, b, c, d como se muestra en la Fig. 2.10, usamos la medida de calidad *rectangles 2015* propuesta por González [46], que está dada por

$$\mu(Q) = \frac{2 \min(a_1, a_2, a_3, a_4)}{\sqrt{(a^2 + c^2)(b^2 + d^2)}} \in (0, 1],$$

donde

$$\begin{aligned} a_1 &= \text{área}(ABC), & a_2 &= \text{área}(CDA), \\ a_3 &= \text{área}(ABD), & a_4 &= \text{área}(BCD). \end{aligned}$$

Observaciones 2.2: $\mu(Q) = 1$ si y sólo si Q es un rectángulo.

Sea G una malla de cuadriláteros sobre una región poligonal Ω , decimos que G es una *malla de alta calidad* si al menos un porcentaje p de sus cuadriláteros tiene medida $\mu \geq \tau$, donde $0 < p \leq 100$ y $0 < \tau \leq 1$; de otro modo decimos que la malla G tiene *baja calidad*.

Observaciones 2.3: Generar mallas donde más del 90 % de sus cuadriláteros tengan medida $\mu \geq 0.9$ puede requerir una descomposición en varias subregiones y más tiempo computacional. En cambio, si menos del 80 % de los cuadriláteros tienen medida $\mu \geq 0.8$,

los métodos que requieren cuadriláteros con ángulos aproximadamente rectos pueden verse afectados. Así que en nuestras mallas de alta calidad pedimos $p = 85\%$ de los cuadriláteros con tolerancia $\tau = 0.8$ para la medida *rectangles 2015*.

Podemos inspeccionar visualmente la calidad de una malla de cuadriláteros G mediante un *mapa de calidad*, esto es, coloreamos las celdas de la malla con un gradiente de colores basado en los valores de la medida μ , en nuestro estudio empleamos el siguiente esquema de colores:

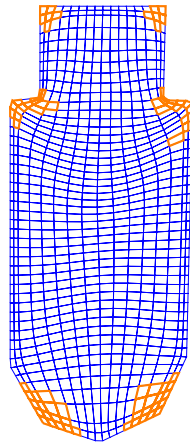
En celdas azules: $\mu \geq 0.8$.

En las celdas naranjas: $0.4 \leq \mu < 0.8$.

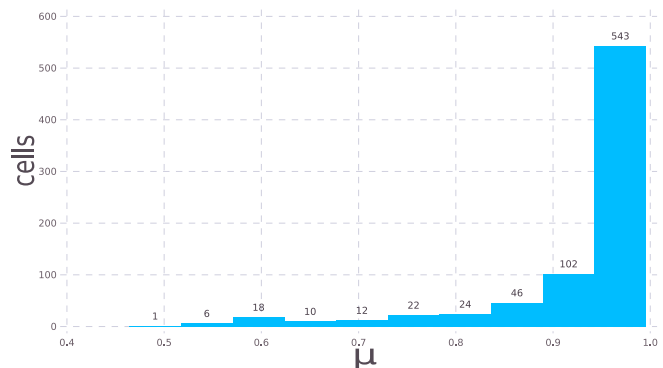
En las celdas rojas: $\mu < 0.4$.

Además del mapa de calidad examinamos un *histograma de calidad*, que nos indica la distribución de cuadriláteros con base al valor de la medida de calidad μ como mostramos en la Fig. 2.11(b). Contamos el número de cuadriláteros que tienen medida de calidad en subintervalos de $[0, 1]$.

Ejemplo 2.3. Generamos una malla estructurada convexa de cuadriláteros de tamaño 17×50 sobre la región USB, su mapa de calidad se muestra en la Fig. 2.11(a), donde observamos que la mayoría de sus celdas son azules, de hecho, la malla es de alta calidad porque el 90% de sus cuadriláteros tienen calidad $\mu \geq 0.8$. Además, mostramos el histograma de calidad correspondiente en la Fig. 2.11(b), donde notamos que la mayoría de los cuadriláteros tienen medida de calidad en el subintervalo $0.95 \leq \mu \leq 1$.



(a) Mapa de calidad



(b) Histograma de calidad

Figura 2.11: Mapa e Histograma de calidad para malla estructurada de cuadriláteros 17×50 sobre la región USB.

La medida de calidad μ nos permite formular la construcción de parametrizaciones como la generación de mallas estructuradas de calidad. Sea \mathbf{X} la parametrización bilineal del cuadrado unitario a Ω dada por (2.5) asociada a la malla estructurada G , decimos que \mathbf{X} es una *parametrización admisible* si la malla G es de alta calidad.

Queremos construir parametrizaciones admisibles de regiones poligonales, sin embargo esta tarea no es fácil, incluso las mallas estructuradas sobre algunas regiones convexas pueden tener una baja calidad.

Ejemplo 2.4. Mostramos dos regiones donde no podemos generar mallas estructuradas de un solo bloque con alta calidad debido a los bordes puntiagudos. Primero, generamos una malla estructurada 25×50 sobre el triángulo ABC con ángulos 102° , 50° y 28° , para ello tuvimos que agregar el punto medio P del segmento BC , su mapa de calidad se muestra en la Fig. 2.12. La región es convexa, pero la malla es de baja calidad, pues sólo el 60% de las celdas tienen calidad $\mu \geq 0.8$.

Ahora, generamos una malla estructurada 50×6 para un cuadrilátero $ABCD$ con ángulos de 26° , 152° , 22° y 160° y mostramos su mapa de calidad en la Fig. 2.13. Aunque la región sea sencilla, no hay celdas de alta calidad en color azul, donde $\mu \geq 0.8$, en cambio, el 85% de las celdas tienen calidad intermedia en color naranja, donde $0.4 \leq \mu < 0.8$.

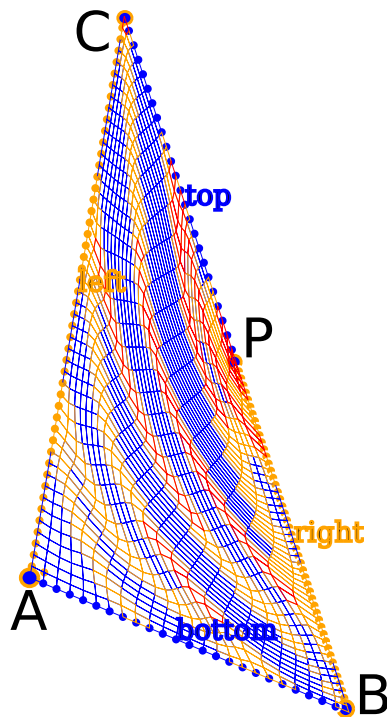


Figura 2.12: Mapa de calidad para malla estructurada del triángulo ABC .

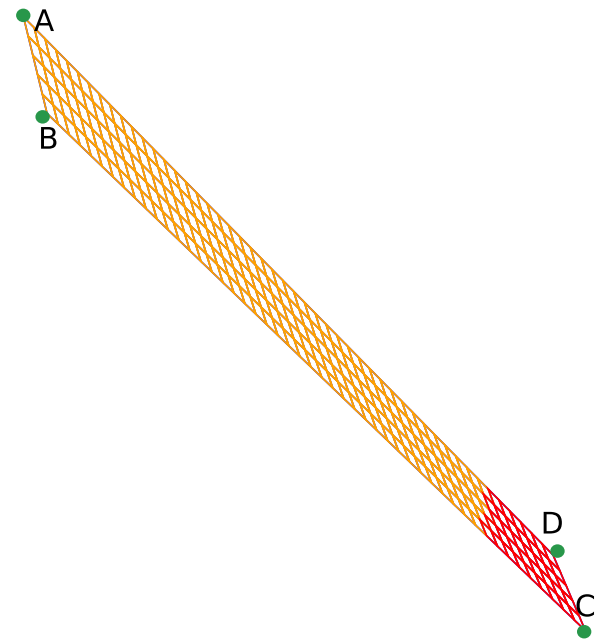


Figura 2.13: Mapa de calidad para malla estructurada 50×6 sobre cuadrilátero $ABCD$.

Generar una malla estructurada de alta calidad es casi imposible cuando la geometría de la región es complicada. Un aspecto a considerar es la concavidad de la región, pues mallas estructuradas sobre regiones que no sean convexas pueden tener una baja calidad. Además de los bordes puntiagudos con ángulos pequeños, tenemos que tener cuidado con los vértices que forman ángulos interiores mayores que 180° , llamados *vértices reflex*.

Ejemplo 2.5. Generamos mallas estructuradas 25×17 sobre dos regiones no convexas A y B con la misma selección de esquinas marcadas con estrellas rojas, la diferencia es que la región B tiene un pico más pronunciado que la región A en el vértice reflex, mostramos sus mapas de calidad en las figuras 2.14 y 2.15, respectivamente. La malla sobre la región A es de alta calidad, mientras que la malla sobre la región B es de baja calidad.

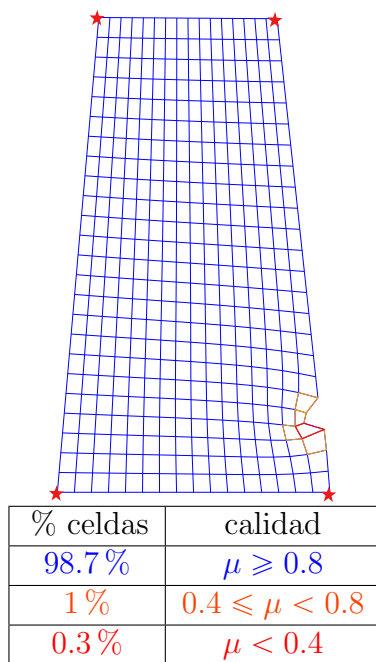


Figura 2.14: Mapa de calidad para malla estructurada sobre región A.

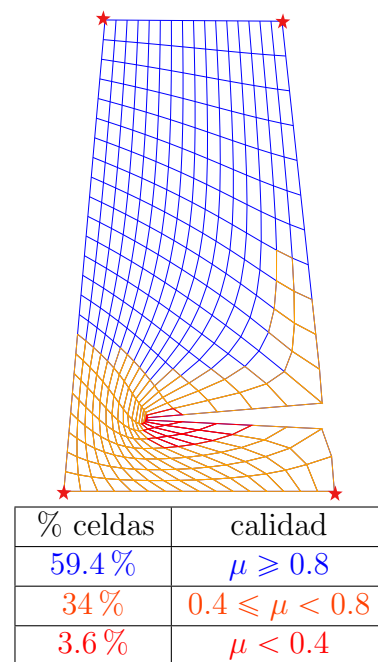


Figura 2.15: Mapa de calidad para malla estructurada sobre región B.

2.3. Parametrización con Mallas por Bloques

La clave para construir *parametrizaciones admisibles* de regiones complicadas es descomponerlas en subregiones más sencillas, separamos la región Ω en una colección finita de subregiones Ω_k para $k = 1, \dots, n$. Construimos una familia de parametrizaciones X_k del cuadrado unitario S a las subregiones Ω_k , de ese modo la región Ω vista por bloques es esencialmente un cuadrado, en el sentido de que tenemos cambios de coordenadas de S a los bloques, Fig. 2.16. Esto nos permite generar mallas de mejor calidad en comparación con las mallas de un solo bloque.

Deseamos obtener una *descomposición admisible* $\{\Omega_k\}$ de la región Ω , en el sentido de que las subregiones Ω_k sean adecuadas para generar mallas estructuradas de alta calidad sin singularidades interiores en las intersecciones. Condiciones suficientes para obtener esta descomposición de regiones se describen en el Capítulo 4.

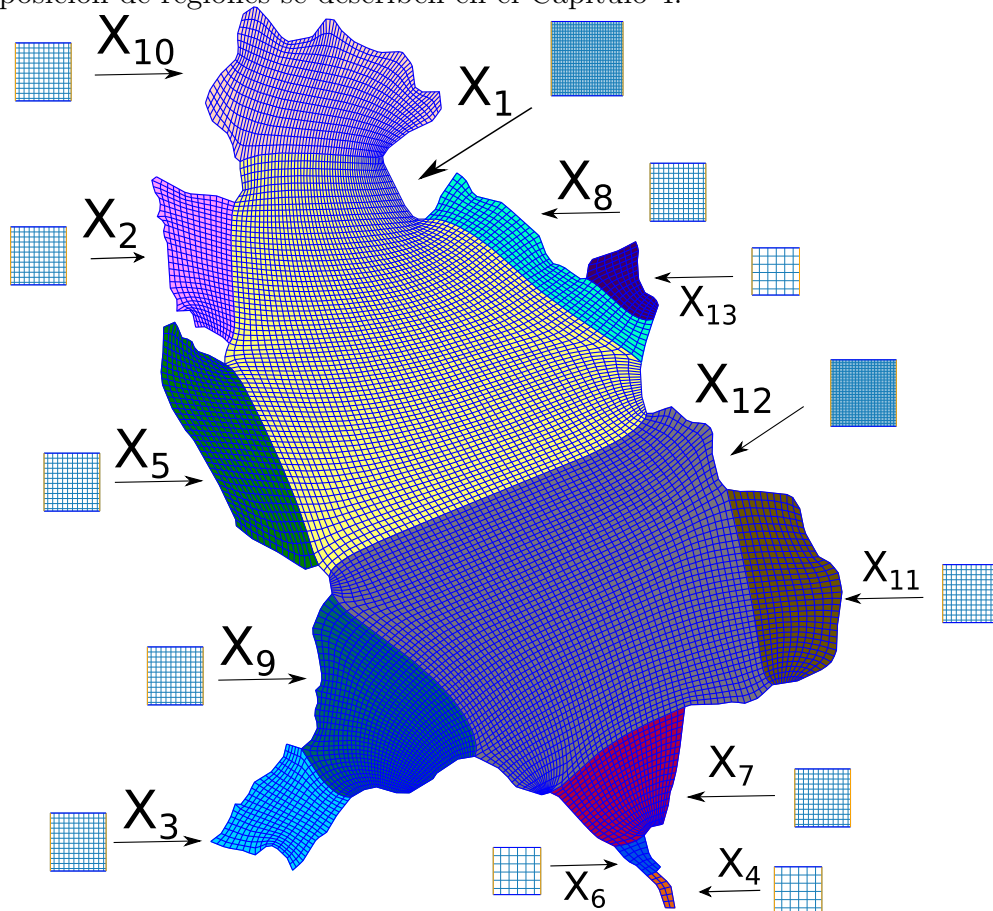


Figura 2.16: *Parametrizaciones compatibles y admisibles* de subregiones del Estado de Nayarit, su numeración depende del orden en que realizamos los cortes.

Una vez que separamos la región en subregiones, construimos cambios de coordenadas del cuadrado unitario a las subregiones. Para que los puntos en las intersecciones no tengan diferentes representaciones con estos cambios de coordenadas, necesitamos que sean compatibles, en el sentido de que induzcan mapeos entre las preimágenes de las intersecciones.

Supongamos que las subregiones Ω_i y Ω_j de la región Ω se intersectan, decimos que los mapeos $X_i : S \rightarrow \Omega_i$ y $X_j : S \rightarrow \Omega_j$ son *parametrizaciones compatibles* si la composición de funciones $X_i^{-1} \circ X_j$ es un homeomorfismo de la preimagen $X_j^{-1}(\Omega_i \cap \Omega_j)$ a la preimagen $X_i^{-1}(\Omega_i \cap \Omega_j)$, Fig. 2.17.

Nuestro problema es construir una familia de *mapeos admisibles y compatibles* del cuadrado unitario a subregiones de una *descomposición admisible*, a saber, la generación de mallas estructuradas por bloques de alta calidad.

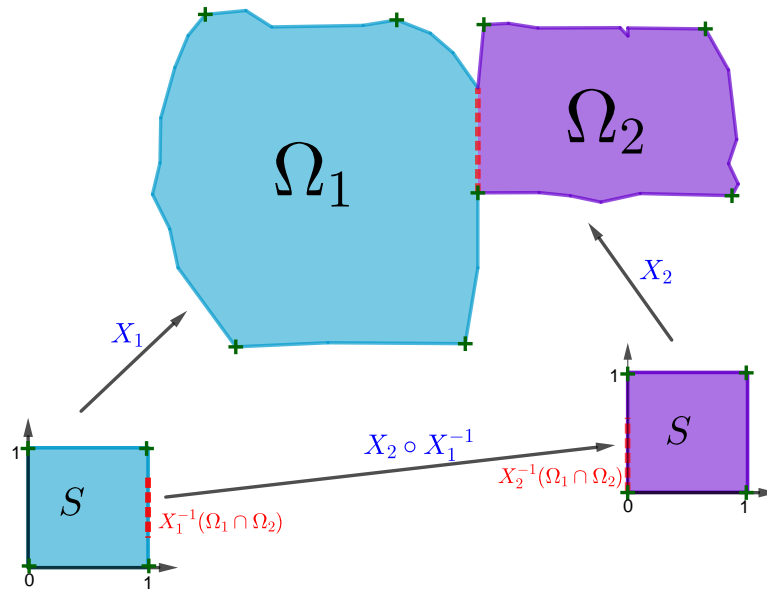


Figura 2.17: Parametrizaciones compatibles $X_1 : S \rightarrow \Omega_1$ y $X_2 : S \rightarrow \Omega_2$ de las subregiones Ω_1 y Ω_2 inducen el mapeo $X_2 \circ X_1^{-1}$ entre lados del cuadrado S .

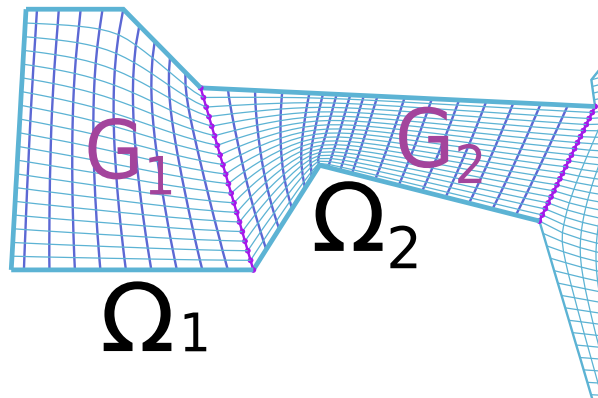


Figura 2.18: Parametrizaciones compatibles de las subregiones Ω_1 y Ω_2 tienen mallas con los mismos puntos frontera en $\Omega_1 \cap \Omega_2$.

Problema 2.4. (Generación de Mallas Estructuradas por Bloques) Sea Ω una región poligonal. Hallar una colección de parejas (Ω_i, \mathbf{X}_i) , $i = 1, \dots, n$, donde

1. $\{\Omega_i\}_{i=1}^n$ es una *descomposición admisible* de Ω ,
2. $\mathbf{X}_i : S \rightarrow \Omega_i$, $i = 1, \dots, n$ son *parametrizaciones admisibles y compatibles*.

Observaciones 2.4: Construimos las parametrizaciones X_k como los mapeos bilineales (2.5) dados por mallas estructuradas G_k sobre las subregiones Ω_k , por lo que si las mallas G_i y G_j tienen los mismos puntos frontera sobre $\Omega_i \cap \Omega_j$, entonces los mapeos X_i y X_j son compatibles, Fig. 2.18.

Para mostrar las ventajas de las *parametrizaciones compatibles y admisibles*, generamos mallas estructuradas por bloques en algunas regiones donde habíamos generado mallas de baja calidad, notamos que las nuevas mallas son de mejor calidad.

Ejemplo 2.6. Separamos el cuadrilátero del Ejemplo 2.4 en un rectángulo y dos triángulos para generar la malla estructurada de tres bloques que mostramos en la Fig. 2.19, junto con su mapa de calidad. En comparación con la malla de la Fig. 2.13, la calidad mejoró del 3.94 % al 94.13 %.

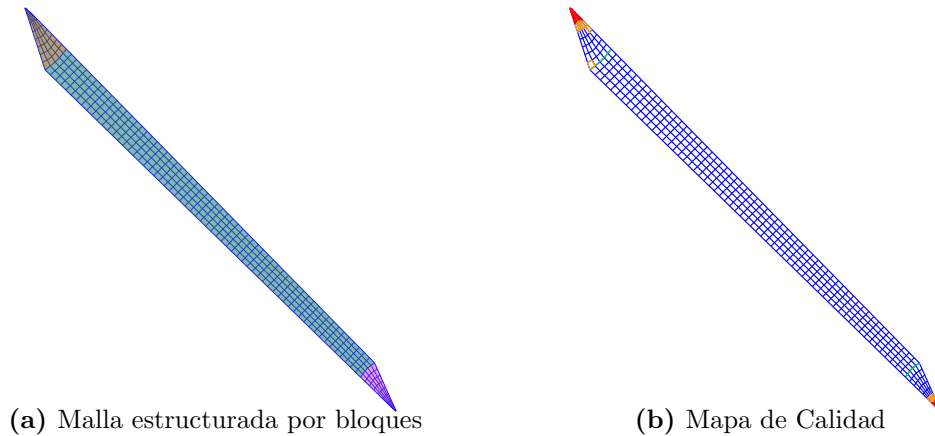


Figura 2.19: Nueva malla sobre el cuadrilátero de Fig. 2.13.

Ejemplo 2.7. Separamos la región B del Ejemplo 2.5 en dos bloques para generar la malla por bloques que mostramos en la Fig. 2.20, junto con su mapa de calidad. En comparación con la malla de la Fig. 2.15, la calidad mejoró del 59.4 % al 97.22 %.

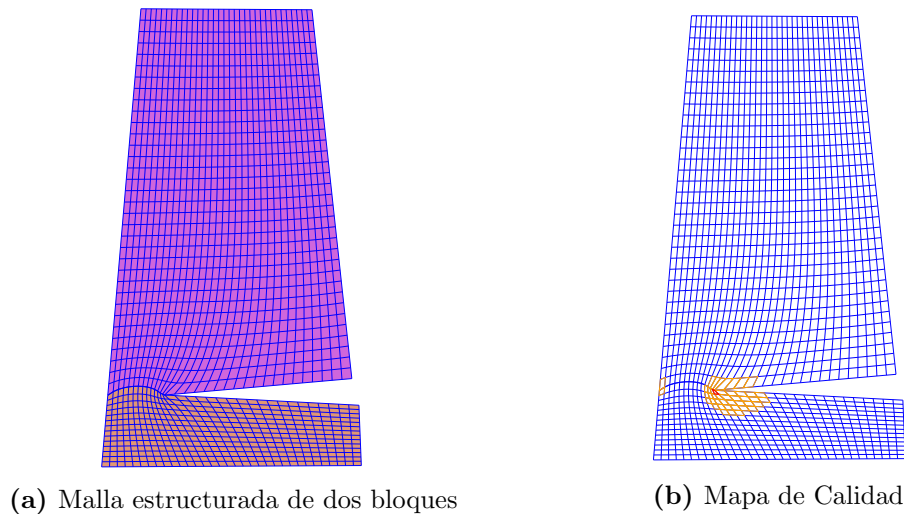


Figura 2.20: Nueva malla sobre región no convexa de la Fig. 2.15.

Ejemplo 2.8. Generamos una malla estructurada de tamaño 9×233 sobre un cuadrado agujerado con el logotipo de Batman siguiendo el esquema de García [44]. Esta malla tiene varias celdas elongadas alrededor de la frontera, de hecho sólo el 67 % de las celdas tienen calidad $\mu \geq 0.8$. En la Fig. 2.21 mostramos su mapa de calidad y marcamos las esquinas por estrellas rojas.

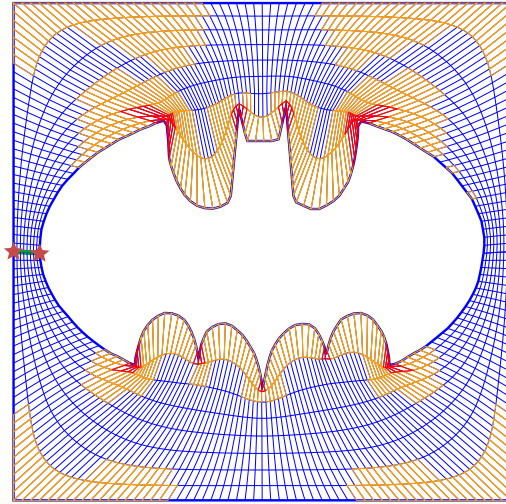
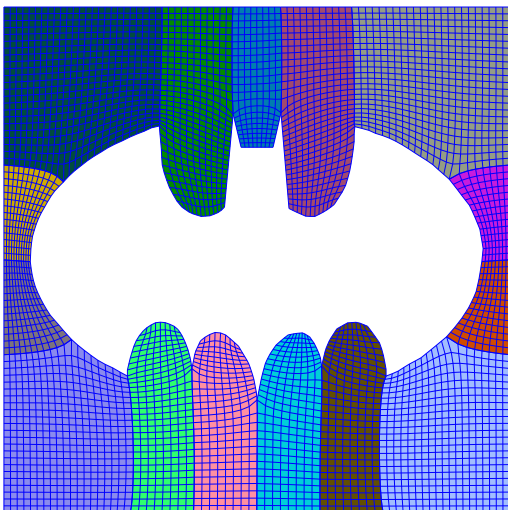
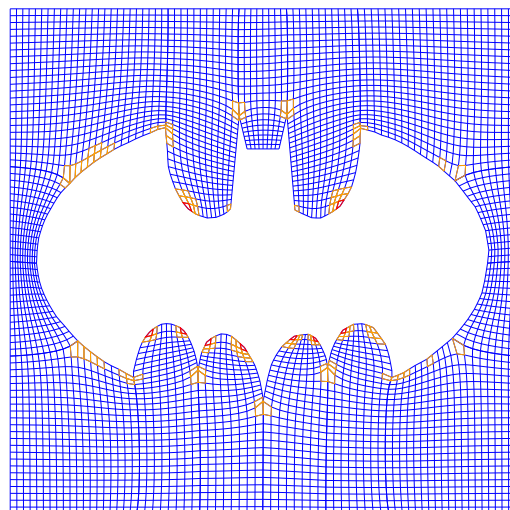


Figura 2.21: Mapa de calidad para malla estructurada sobre un cuadrado con un agujero.

Ahora, generamos una malla estructurada de 15 bloques sobre la misma región, que mostramos en la Fig. 2.22, junto con su mapa de calidad. En comparación con la malla de la Fig. 2.21, la calidad aumentó al 97.19 %.



(a) Malla estructurada por bloques



(b) Mapa de Calidad

Figura 2.22: Nueva malla sobre cuadrado con agujero de batman.

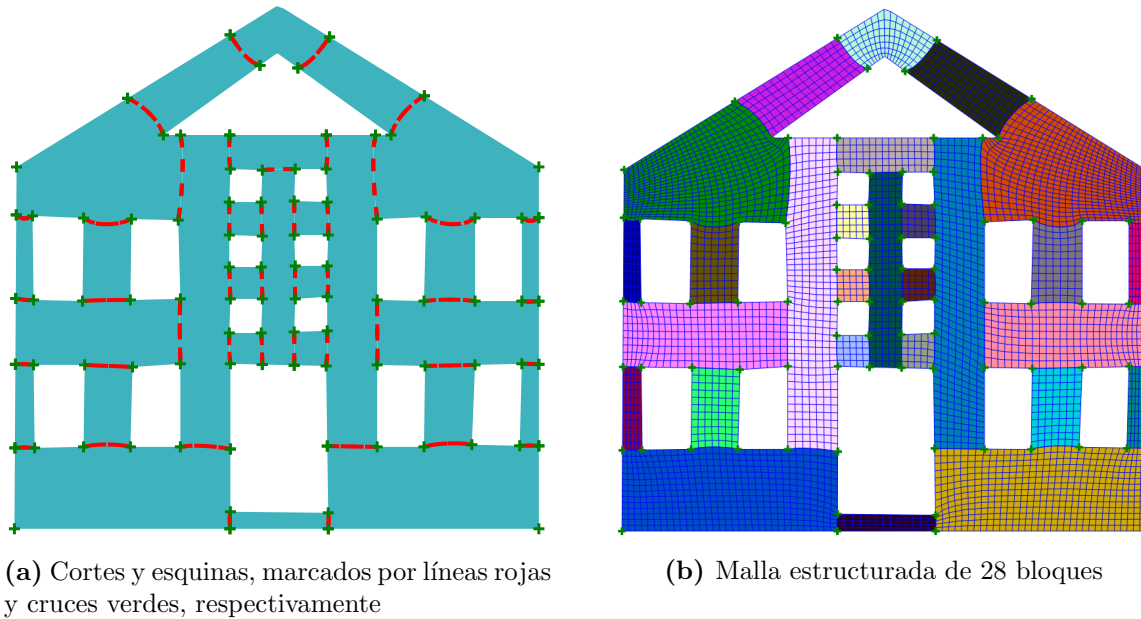


Figura 2.23: Generación de una malla estructurada por bloques a partir de la selección de cortes y esquinas.

La construcción de nuestras parametrizaciones requiere una *descomposición admisible* de la región y la generación de mallas estructuradas sobre las subregiones de modo que sus celdas empaten lado a lado en las intersecciones. Para evitar introducir singularidades interiores separamos la región Ω por cortes, de ese modo el problema de hallar una *descomposición admisible* se puede reducir a la elección adecuada de cortes que describimos en el Capítulo 4.

En cada subregión Ω_k elegimos cuatro puntos en su frontera, llamados *esquinas*, que corresponden a las esquinas del cuadrado S . Estos puntos separan la frontera de Ω_k en cuatro lados que parametrizamos sobre los lados de S con un número compatible de puntos como describimos en el Capítulo 5.

Generamos mallas estructuradas en cada subregión con puntos frontera dados por los lados compatibles. Podemos unir estas mallas y mejorar su calidad de manera automática como describimos en el Capítulo 6, Fig. 2.23. En consecuencia:

La generación de mallas estructuradas por bloques de alta calidad sin singularidades interiores, y por consiguiente, la construcción de parametrizaciones se pueden reducir a la elección adecuada de cortes y esquinas.

2.4. Resumen del Capítulo

Las mallas estructuradas de cuadriláteros sobre regiones poligonales pueden pensarse como una colección de mapeos del cuadrado unitario a subregiones, por lo que generar estas mallas es construir parametrizaciones de regiones.

Sea Ω una región poligonal acotada sin agujeros, y sea S el cuadrado unitario, elegimos cuatro puntos sobre $\partial\Omega$, llamados esquinas, que separan la frontera de Ω en cuatro curvas consecutivas $\mathbf{b}_b, \mathbf{b}_r, \mathbf{b}_t$ y \mathbf{b}_l parametrizadas sobre los lados de S . Extendemos estas curvas como un homomorfismo \mathbf{X} que transforma puntos (ξ, η) de S en puntos (x, y) de Ω de modo que

$$\begin{aligned} \mathbf{X}|_{\eta=0} &= \mathbf{b}_b, & \mathbf{X}|_{\eta=1} &= \mathbf{b}_t, \\ \mathbf{X}|_{\xi=0} &= \mathbf{b}_l, & \mathbf{X}|_{\xi=1} &= \mathbf{b}_r. \end{aligned}$$

Para extender las parametrizaciones de la frontera, generamos una malla estructurada de cuadriláteros G de tamaño $m \times n$ sobre Ω con puntos frontera dados por las curvas $\mathbf{b}_b, \mathbf{b}_r, \mathbf{b}_t$ y \mathbf{b}_l . De esta manera, el mapeo \mathbf{X} se puede obtener por el *producto tensorial B-spline bilineal* (2.5).

Para medir la calidad de las mallas de cuadriláteros usamos la medida *rectangles 2015* de González [46], denotada por μ , que identifica un cuadrilátero Q como rectángulo si y sólo si $\mu(Q) = 1$. Decimos que una malla es de alta calidad si al menos el 85% de sus celdas tienen medida $\mu \geq 0.8$, y que la parametrización \mathbf{X} es *admisibile* si su malla de control G es de alta calidad. Podemos inspeccionar la calidad de una malla de manera visual mediante un mapa de calidad, que consiste en colorear las celdas con base a los valores de la medida μ .

Las mallas estructuradas de un solo bloque sobre regiones complicadas pueden tener una baja calidad, por lo que descomponemos la región Ω en subregiones Ω_k que sean admisibles para generar mallas estructuradas de alta calidad, y por consiguiente que tengan *parametrizaciones admisibles* $\mathbf{X}_k : S \rightarrow \Omega_k$. Cuando las subregiones Ω_i y Ω_j se intersectan, sus parametrizaciones \mathbf{X}_i y \mathbf{X}_j son compatibles si inducen un mapeo entre las preimágenes de $\Omega_i \cap \Omega_j$. Así que reformulamos nuestro problema como la generación de mallas estructuradas por bloques:

Hallar una descomposición $\{\Omega_k\}$ de la región Ω en *subregiones admisibles* y una familia $\{\mathbf{X}_k : S \rightarrow \Omega_k\}$ de *parametrizaciones admisibles y compatibles*.

Separamos la región Ω por cortes para evitar singularidades interiores y separamos la frontera de cada subregión por cuatro esquinas para construir mapeos compatibles entre los lados del cuadrado unitario y de las subregiones, que a su vez podemos extender como mapeos bilineales. Así que nuestro problema se puede reducir a la elección adecuada de cortes y esquinas.

En el siguiente capítulo describimos el primer paso para construir nuestras parametrizaciones, a saber, simplificar la frontera de regiones complicadas. Este paso es indispensable para reducir el costo computacional que requiere la generación de mallas estructuradas por bloques sobre regiones complicadas.

Deseamos generar mallas de alta calidad sobre regiones poligonales, sin embargo sus fronteras pueden tener alto nivel de detalle, bordes puntiagudos o ruido, y en el peor de los casos la generación de mallas puede no ser viable, así que aproximamos sus fronteras por curvas más suaves que preserven la forma de las regiones con menos puntos.

Problema 3.1. (Aproximación de Contornos Poligonales) Sea P un polígono simple. Hallar una curva cerrada simple y suave C que preserve la forma de P con la mínima cantidad de puntos.

Antecedentes. Aproximamos contornos poligonales por curvas suaves, en particular hemos usado curvas parametrizadas por polinomios definidos por tramos que cumplen condiciones de suavidad, a saber, las *curvas spline*. El problema de minimizar el error de aproximación entre el contorno y una *curva spline* en la norma L^2 está *mal planteado* en el sentido de que el error se reduce al refinar el spline, esto es, se genera una sucesión de *curvas spline* que minimizan el error, pero su límite puede no ser una *curva spline*. Así que regularizamos este problema para suavizar contornos.

Una manera de regularizar el problema es minimizar el funcional dado por la norma L^2 de la segunda derivada con una penalización sobre el error en la misma norma. El problema regularizado es factible y una solución del mismo es un *spline cúbico natural*, ya que minimiza este funcional entre todos los interpolantes con segunda derivada continua [9, 30]. Otra regularización del problema es minimizar el error en norma L^2 con una penalización sobre el número de nodos del spline. Brandt et al. [21] usan *spline cúbicos* y en la penalización detectan los nodos que pueden eliminarse con una condición sobre la tercera derivada.

En trabajos previos [9, 49, 112] nuestro grupo UNAMALLA ha suavizado contornos con *spline cúbicos* y *spline cónicos*, no obstante preferimos las *curvas B-spline cuadráticas*, Fig. 3.1, ya que hay métodos para extender estas curvas como *mapeos bicuadráticos* [1, 2], la dificultad es generar *curvas B-spline* que preserven la forma del contorno.

Algunos bordes pueden quedar más redondeados de lo necesario en el suavizamiento, por lo que simplificamos contornos en lugar de suavizarlos, esto es, eliminamos puntos que no contribuyan a preservar la forma, Nuestro grupo UNAMALLA desarrolló métodos para reducción de puntos en trabajos previos [111], donde se emplean criterios de colinealidad para reemplazar curvas poligonales por segmentos, siguiendo la idea de Pavlidis [96], y por otra parte se eliminan puntos por detección de puntos dominantes, usando una modificación del algoritmo de Ray [107]. En un principio utilizábamos estos métodos, pero tienen dificultades para simplificar polígonos con alto nivel de detalle. Ambos métodos pueden remover más puntos de los necesarios, e incluso generar bordes puntiagudos y autointersecciones, Fig. 3.2.

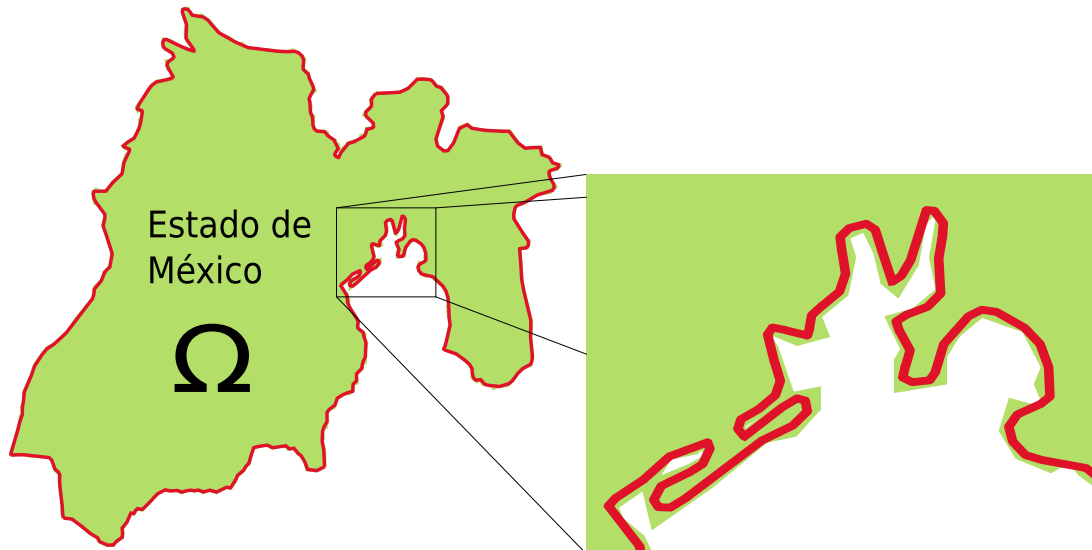


Figura 3.1: La *curva B-spline cuadrática* de color rojo aproxima la frontera del estado de México.

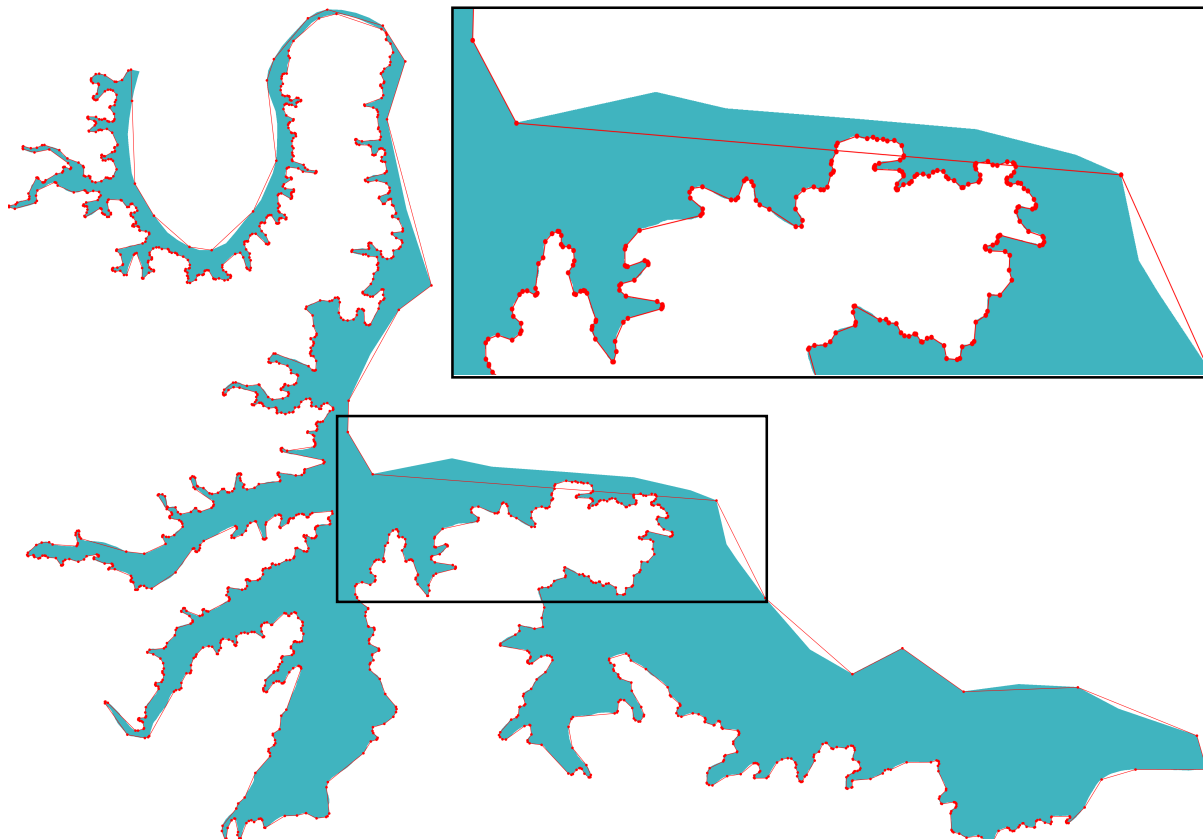
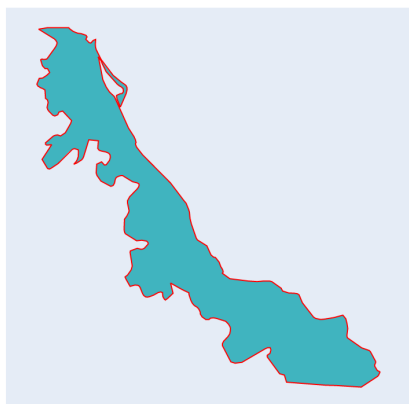


Figura 3.2: Simplificación del contorno *Presas La Amistad* de 10445 a 1823 puntos por método de colinealidad de Pavlidis [96] con modificación de Rivera [111]. El contorno simplificado de color rojo tiene autointersecciones.

Nuevos Métodos. En vista de lo anterior, en este trabajo introducimos dos criterios para la simplificación de polígonos, ambos usan los triángulos formados por cada tres vértices consecutivos. El primer criterio compara las áreas de los triángulos en relación con su área promedio, mientras que el otro criterio compara los *inradios* y *circunradios* de los triángulos. Ambos criterios son independientes de la escala de la región y nos permiten simplificar polígonos preservando su forma, como mostramos en el Ejemplo 3.1.

Ejemplo 3.1. Usamos el contorno poligonal del estado de Veracruz con 389 puntos para ilustrar los criterios de las áreas y de los radios. Simplificamos el contorno por criterio de las áreas removiendo 95 puntos con factor $\epsilon = 0.01$, Fig. 3.3. Por otra parte, simplificamos por el criterio de los radios removiendo 96 puntos con umbral $\rho = 0.002$. En cambio si usamos $\rho = 0.01$, removemos 190 puntos, pero la forma comienza a perderse, Fig. 3.4.



(a) Polígono original y su simplificación.



(b) Acercamiento.

Figura 3.3: Simplificación del contorno del Estado de Veracruz por criterio de área con factor $\epsilon = 0.01$. El polígono original tiene relleno azul y el polígono simplificado es el contorno rojo.



(a) $\rho = 0.002$.



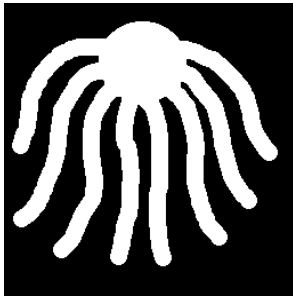
(b) $\rho = 0.01$.

Figura 3.4: Simplificación del contorno del Estado de Veracruz por criterio de los radios con distintos valores para el umbral ρ . El polígono simplificado es el contorno rojo.

Además del nivel de detalle, los contornos pueden tener ruido. Los criterios de las áreas y los radios no son adecuados para reducir ruido, por lo que incorporamos un método para suavizar picos [90]. Sin embargo, se presentan dificultades aún con esta modificación, pues no se preserva la forma de algunos *contornos digitales* como mostramos en el Ejemplo 3.2.

Ejemplo 3.2. Queremos reducir el ruido del contorno digital *Octopus12* de 2287 puntos, mediante nuestros criterios de áreas y radios. Primero, empleamos el criterio de las áreas con factor $\epsilon = 0.19$ del área promedio. Eliminamos 1627 puntos, pero no reducimos el ruido. Cuando aumentamos el factor a $\epsilon = 0.2$, solo nos quedamos con 33 puntos, sin embargo ya no conservamos la forma de la región.

Ahora, empleamos el criterio de los radios con umbral $\rho = 0.01$, removemos 1329 puntos, pero no reducimos el ruido. Si $\rho = 0.03$, eliminamos 1808 puntos, sin embargo perdimos la forma del contorno.



(a) Imagen digital.

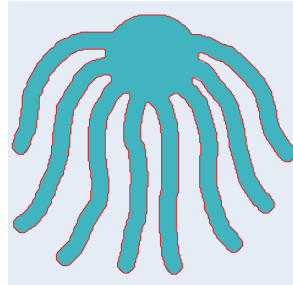
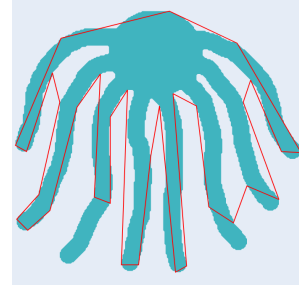
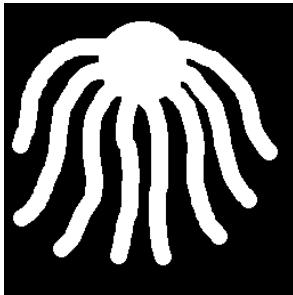
(b) $\epsilon = 0.19$.(c) $\epsilon = 0.2$.

Figura 3.5: Simplificación de curva digital *Octopus12* por criterio de áreas con distintos valores del factor ϵ . El polígono simplificado es el contorno rojo.



(a) Imagen digital.

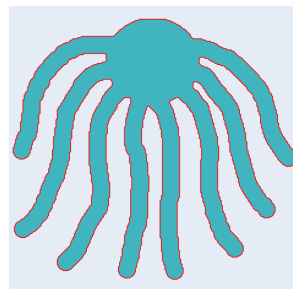
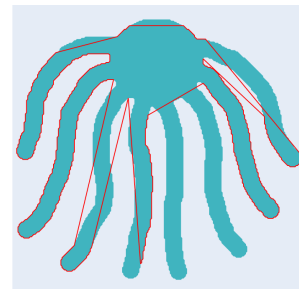
(b) $\rho = 0.01$.(c) $\rho = 0.03$.

Figura 3.6: Simplificación de curva digital *Octopus12* por criterio de los radios con distintos valores del umbral ρ . El polígono simplificado es el contorno rojo.

Motivados por la simplificación de polígonos en Cartografía [135], modificamos nuestros criterios de áreas y radios, lo que hacemos es eliminar de manera secuencial los triángulos más pequeños en ambos criterios. Esto nos llevó a desarrollar un *método robusto de tres pasos* para reducir ruido y simplificar polígonos con alto nivel de detalle preservando la forma de la región como mostramos en el Ejemplo 3.3. En §3.3.4 explicamos los criterios modificados y describimos nuestro *método de tres pasos*.

Ejemplo 3.3. Usamos nuestros métodos modificados para reducción de ruido de la curva digital *octopus12* de 2287 puntos y los comparamos con el Algoritmo de Visvalingam [135]. En la Tabla 3.1 se muestran los puntos eliminados así como los valores de los umbrales correspondientes. Mostramos el contorno simplificado que generamos con nuestro *método de tres pasos modificado* en la Fig. 3.7. Comparamos con los otros métodos en un acercamiento que realizamos en la Fig. 3.8 para apreciar mejor las diferencias.

Método	Pts. Removidos	Factor
Visvalingam con escalamiento	2011	$\epsilon = 20$
Áreas Modificado	2015	$\epsilon = 20$
Radios Modificado	2023	$\epsilon = 20$
Tres Pasos Modificado	2098	$\delta = 10^{-4}, \epsilon = 3$

Tabla 3.1: Puntos eliminados y factores empleados en los métodos modificados para reducción de ruido del contorno *octopus12*.

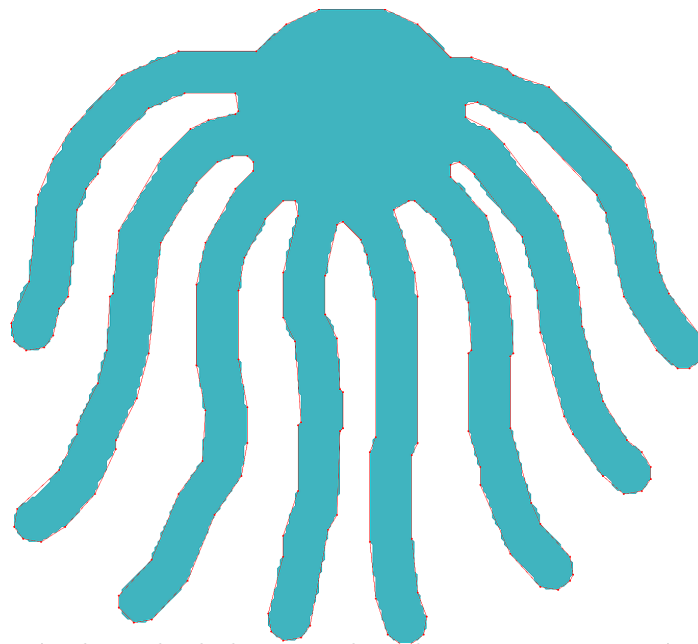
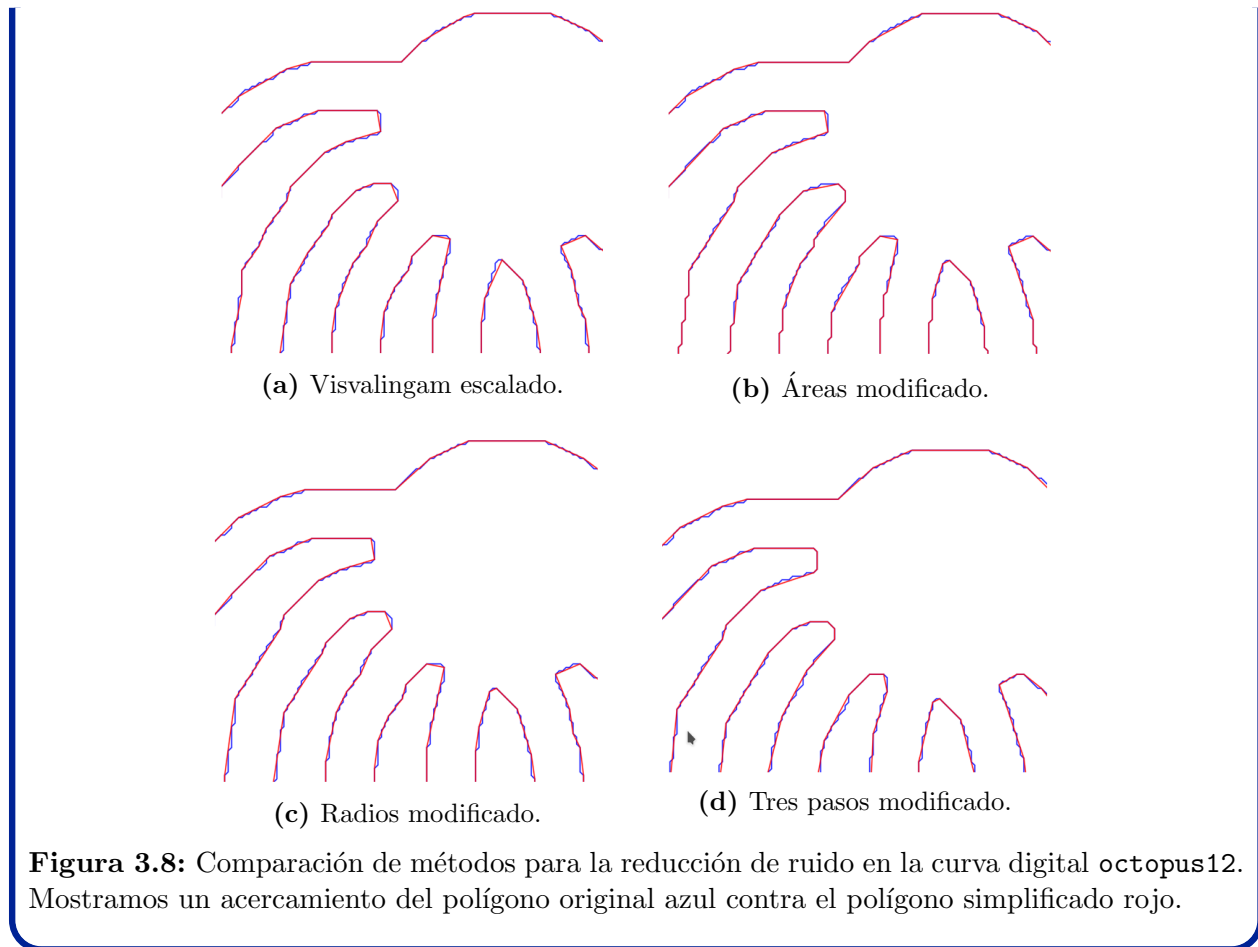


Figura 3.7: Reducción de ruido de la curva digital *octopus12* por método modificado de tres pasos. El contorno rojo es el polígono simplificado.



3.1. Simplificación de Polígonos

La simplificación de polígonos complicados consiste en construir un polígono que preserve la forma del polígono original usando la menor cantidad posible de vértices de modo que los vértices del polígono aproximado sean vértices del polígono original. Eliminamos puntos sin perder la forma, los puntos no se mueven ni se agregan otros puntos nuevos.

Problema (Simplificación de Polígonos): Sea P un polígono simple. Hallar un polígono simple P' tal que:

- los vértices de P' son vértices de P ,
- P' preserva la forma de P ,
- un polígono que se obtiene al remover vértices de P' no conserva la forma de P .

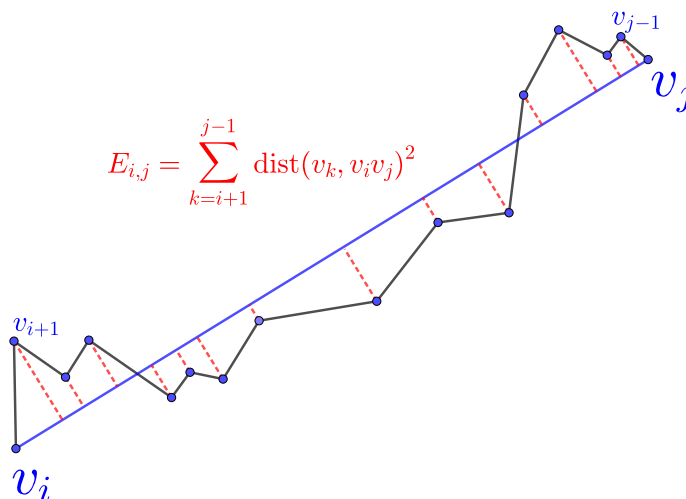


Figura 3.9: Distancia perpendicular entre poligonal $v_i v_{i+1} \dots v_{j-1} v_j$ y segmento $v_i v_j$ medida como el error cuadrado integral $E_{i,j}$.

Simplificación por Criterios de Colinealidad. Reemplazamos curvas poligonales del contorno por segmentos de recta que conectan sus extremos. Al respecto, Pavlidis [96] desarrolló un método para hallar el mínimo número de segmentos con base a criterios de colinealidad, que emplean cotas para el error entre las curvas poligonales y los segmentos candidatos. En cada paso escogemos grupos de curvas poligonales con el mismo número de vértices y las reemplazamos por segmentos cuando su error está acotado. Rivera [111] modifica estos criterios de colinealidad y sugiere valores experimentales para las cotas de los errores.

Simplificación por Detección de Puntos Dominantes. En el reconocimiento de patrones se han simplificado polígonos por detección de *puntos dominantes*, que son esenciales para preservar la forma y usualmente son puntos de alta curvatura. Métodos iterativos para detectar puntos dominantes [82, 101] emplean la distancia perpendicular entre poligonales y los segmentos que unen los extremos, Fig. 3.9.

Ray [107] detecta puntos dominantes de curvas digitales mediante una función que compensa la longitud de los segmentos candidatos con su distancia perpendicular a la poligonal correspondiente, los puntos dominantes son los puntos donde esta función alcanza sus valores máximos. Rivera [111] introduce un parámetro de regularización en la función de Ray para extender el método a curvas poligonales que no sean digitales, el valor del parámetro se elige experimentalmente.

El problema de detectar puntos dominantes puede plantearse como encontrar el mínimo número de puntos de modo que el error entre el polígono original y su aproximación no rebese un umbral dado. Siguiendo esta idea, Prasad [98] construye un esquema independiente de parámetros y heurísticas para la detección de puntos dominantes.

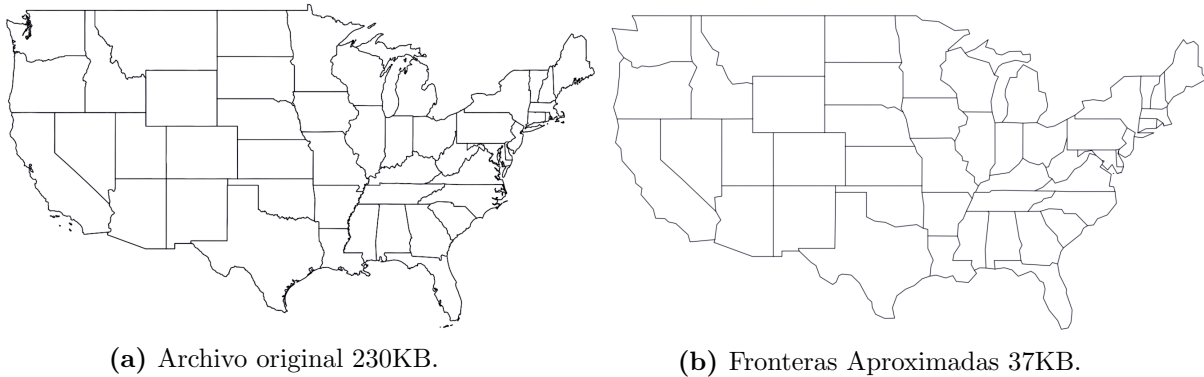


Figura 3.10: Simplificación de polígonos para la división política de EUA [34].

3.1.1. Simplificación de Polígonos en Cartografía

En el diseño de mapas interactivos se requiere aproximar contornos de territorios y cuerpos de agua con diferentes niveles de detalle. Fig. 3.10. Usualmente los polígonos que se manejan tienen un alto nivel de detalle en el sentido de que presentan varios cambios de concavidad y una cantidad considerable de vértices. Varios de estos vértices no son necesarios para mantener la forma. Describimos dos métodos para simplificación de polígonos que se emplean en Cartografía:

- **Algoritmo de Ramer-Douglas-Peucker** [35]. Reemplazamos poligonales por segmentos que unen sus extremos de manera recursiva. En el primer paso, la poligonal tiene todos los vértices. Luego, en cada paso la reemplazamos si las distancias perpendiculares de sus vértices al segmento están acotadas por un umbral dado, Fig. 3.11. De otro modo, se consideran dos nuevas poligonales que comparten el vértice más alejado del segmento. El algoritmo procede de manera recursiva usando el mismo valor del umbral hasta que las distancias entre los segmentos y las poligonales rebasen el umbral dado.
- **Algoritmo de Visvalingam** [135]. Identificamos y removemos de manera iterativa vértices usando las áreas de los triángulos formados por tres vértices consecutivos. En cada paso se calculan las áreas de los triángulos asociados a vértices consecutivos del polígono actual y se identifica el vértice asociado al área más pequeña. Este vértice se elimina si el área correspondiente no cumple con el criterio deseado. En ese caso se genera un nuevo polígono con un vértice menos. El criterio para remover el vértice es que el área del triángulo correspondiente esté por debajo de un umbral dado. Fig. 3.12.

Observaciones 3.1: El algoritmo de Visvalingam [137] tiene diferentes implementaciones y heurísticas. Nosotros escalamos las áreas de los triángulos entre su área promedio para que sea independiente de la escala del polígono. En cada paso eliminamos solo el vértice asociado al triángulo de menor área.

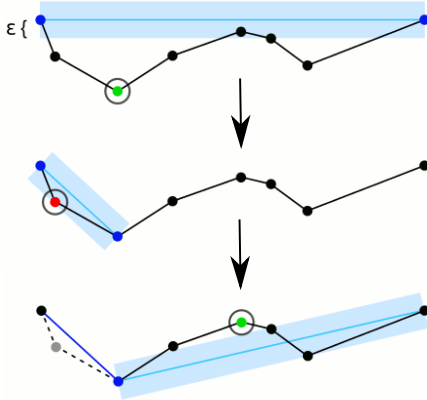


Figura 3.11: Simplificación por algoritmo de Ramer-Douglas-Peucker [35], el doble del umbral ϵ es el ancho de la banda, el punto verde es el más alejado del segmento, y el punto rojo se elimina

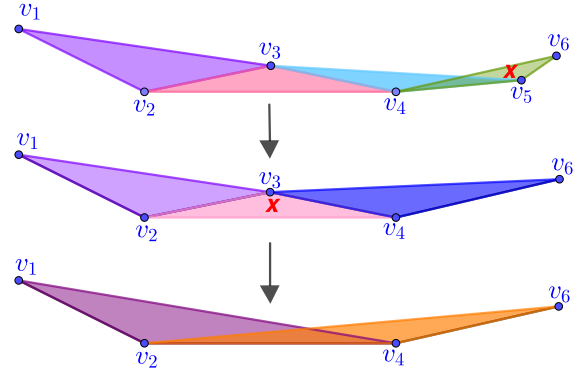


Figura 3.12: Simplificación por algoritmo de Visvalingam [135], donde en cada paso se elimina el triángulo de menor área que marcamos con una cruz.

3.2. Reducción de Ruido

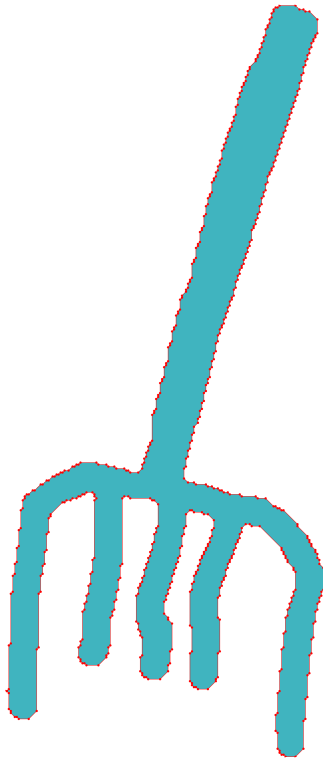
Algunas regiones tienen ruido en la frontera. Podemos generar mallas de cuadriláteros sobre estas regiones, sin embargo su calidad se ve comprometida, pues los cuadriláteros alrededor de los picos pueden tener una baja calidad. Usualmente estos picos no son indispensables y se pueden eliminar. Más aún, mallas de mejor calidad se pueden generar cuando disminuimos el ruido de la región. La reducción de ruido es parte del tratamiento de polígonos complicados.

3.2.1. Detección de Ruido

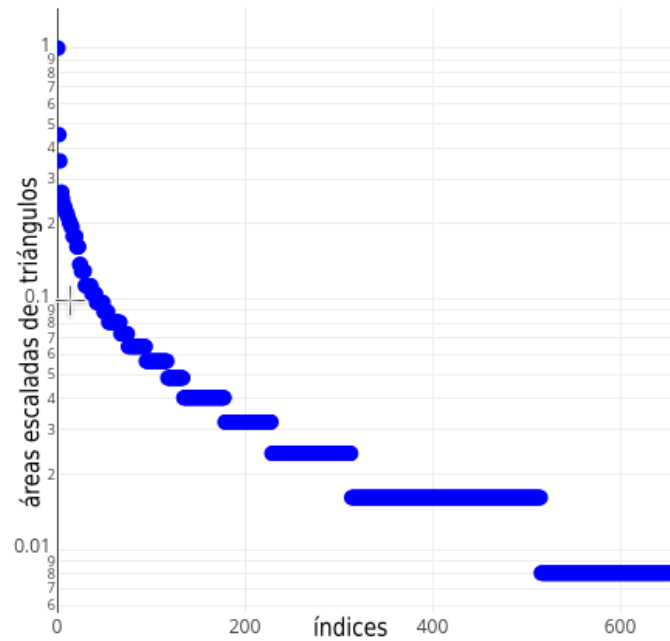
Si sabemos que un polígono tiene ruido, podemos seleccionar un método adecuado para reducir el ruido. La primera pregunta es cómo detectar el ruido. Para ello proponemos un método gráfico. La clave es examinar las áreas de los triángulos formados por tres vértices consecutivos. En concreto examinamos la gráfica en escala logarítmica de sus áreas escaladas y ordenadas. Identificamos que el polígono tiene ruido cuando observamos saltos en el decaimiento de las áreas.

Sea P un polígono con vértices no colineales $v_i, i = 1, \dots, n$. Denotamos por a_i el área del triángulo $v_{i-1}v_iv_{i+1}$ y por \bar{a} a su área promedio. Escalamos las áreas como:

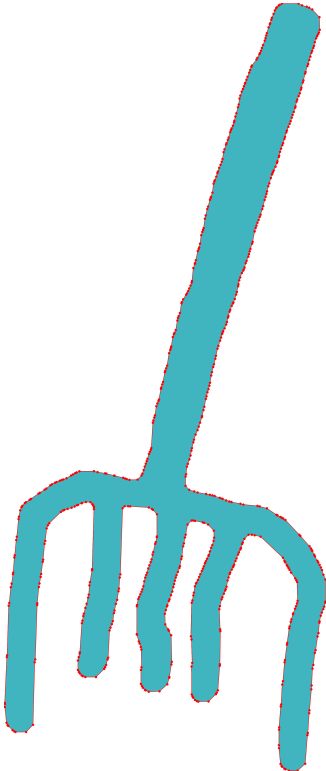
$$\log_{10} \left(\frac{a_i}{\max_i a_i} \right).$$



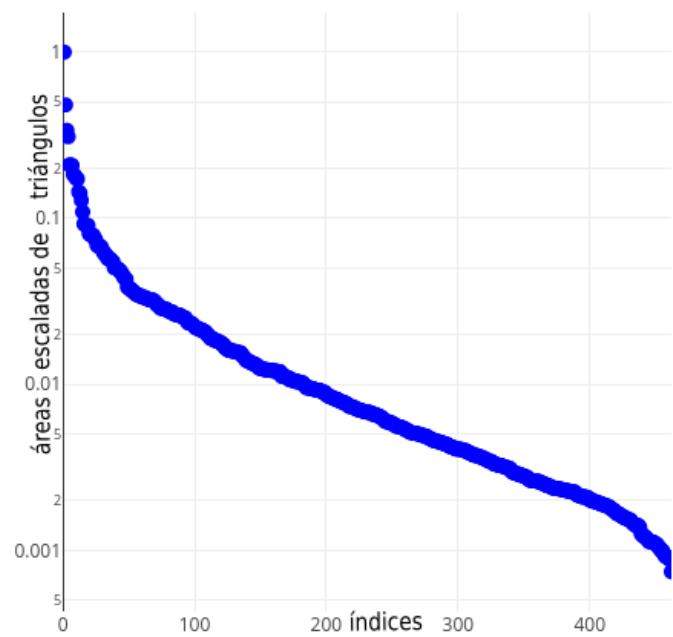
(a) Polígono con ruido.



(b) Gráfica para el polígono del inciso (a).



(c) Polígono sin ruido.



(d) Gráfica para el polígono del inciso (c).

Figura 3.13: Gráficas en escala logarítmica de las áreas de los triángulos dadas por (3.1) para dos contornos de la región Fork10: (a) con ruido, (c) sin ruido. Ambos contornos tienen 656 vértices.

Reordenamos las áreas escaladas en orden decreciente con una permutación σ del conjunto $\{1, \dots, n\}$. Examinamos la gráfica de los puntos:

$$\left(i, \log_{10} \left(\frac{a_{\sigma(i)}}{a_{\sigma(1)}} \right) \right), \quad i = 1, \dots, n. \quad (3.1)$$

La gráfica de estos puntos es escalonada y el valor máximo de las áreas escaladas es 1. De manera experimental hemos encontrado que cuando el polígono tiene ruido, existen algunos saltos considerables entre estos puntos en vez de un decaimiento gradual que se observa en polígonos sin ruido, Fig. 3.13.

3.2.2. Método para Reducción de Ruido

Una manera para suavizar los picos del polígono sin eliminar puntos es el método del polígono de perímetro mínimo [90], que consiste en hallar el polígono de perímetro más pequeño en una vecindad del polígono original de modo que tenga el mismo número de puntos. A continuación, formulamos el problema de optimización asociado.

Sea P el polígono v_1, \dots, v_n . Una vecindad de P de radio δ puede verse como la unión de las bolas de radio δ con centro en puntos densamente equiespaciados sobre el contorno de P . La restricción de que un polígono u_1, \dots, u_n se encuentre en una vecindad de P consiste en que cada u_i se encuentre en un cuadrado de lado 2δ centrado en el vértice v_i :

$$\|v_i - u_i\|_\infty \leq \delta, \quad i = 1, \dots, n$$

El método se puede modificar para que sea independiente de la escala del contorno [59]. La idea es que el radio de las bolas dependa del perímetro p_0 de P como:

$$\|v_i - u_i\|_\infty \leq p_0 \cdot \delta.$$

Así, el método del polígono de perímetro mínimo en la vecindad de P se formula como:

$$\min_{\substack{u_i \in \mathbb{R}^2 \\ \|v_i - u_i\|_\infty \leq p_0 \cdot \delta}} \text{perímetro}(u_1, \dots, u_n). \quad (3.2)$$

Observaciones 3.2:

- ☞ El problema 3.2 es una optimización cuadrática con restricciones de caja. Las incógnitas son las coordenadas de los vértices del polígono óptimo. Este problema tiene solución única si las bolas centradas en los vértices son ajenas [59]. Optimizadores como LBFGS-B pueden usarse para resolver este problema.
- ☞ Suprimimos más ruido conforme aumentamos el valor del factor δ . No obstante, la forma del contorno se puede perder cuando el valor excede un umbral. En nuestras pruebas usamos un factor $10^{-5} \leq \delta \leq 10^{-3}$ para conservar la forma del polígono.
- ☞ Podemos resolver el problema del polígono de perímetro mínimo para suprimir algunos picos pequeños que no necesariamente representen ruido.

Ejemplo 3.4. Reducimos el ruido de la curva digital `ray10` mediante el método del polígono de perímetro mínimo tomando el factor $\delta = 3 \times 10^{-4}$ para el radio de la vecindad. En la Fig. 3.14 mostramos la curva suavizada, la cual preserva la forma con menos ruido y el mismo número de puntos. Las gráficas de las áreas escaladas de sus triángulos se muestran en la Fig. 3.15, la gráfica correspondiente a la curva suavizada ya no es escalonada.

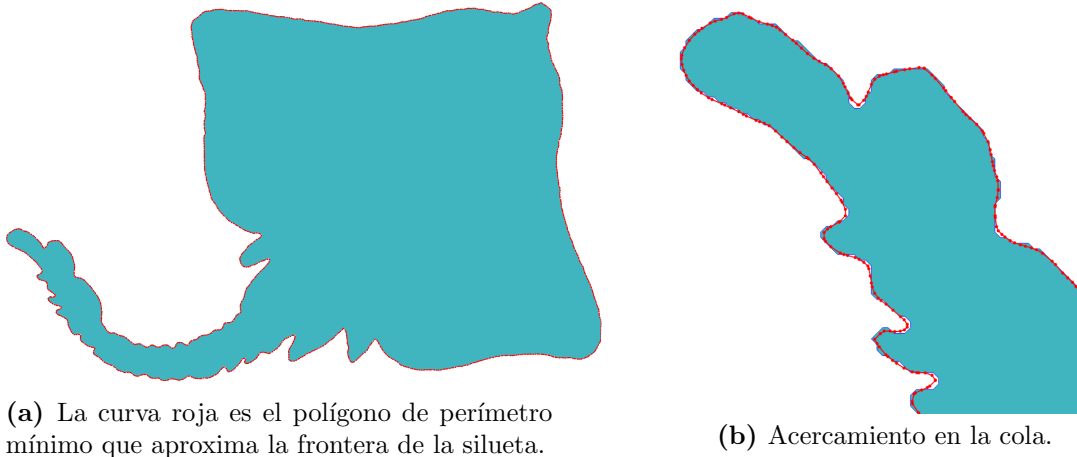


Figura 3.14: Reducción de ruido para la curva digital `ray10` con factor $\delta = 3 \times 10^{-4}$.

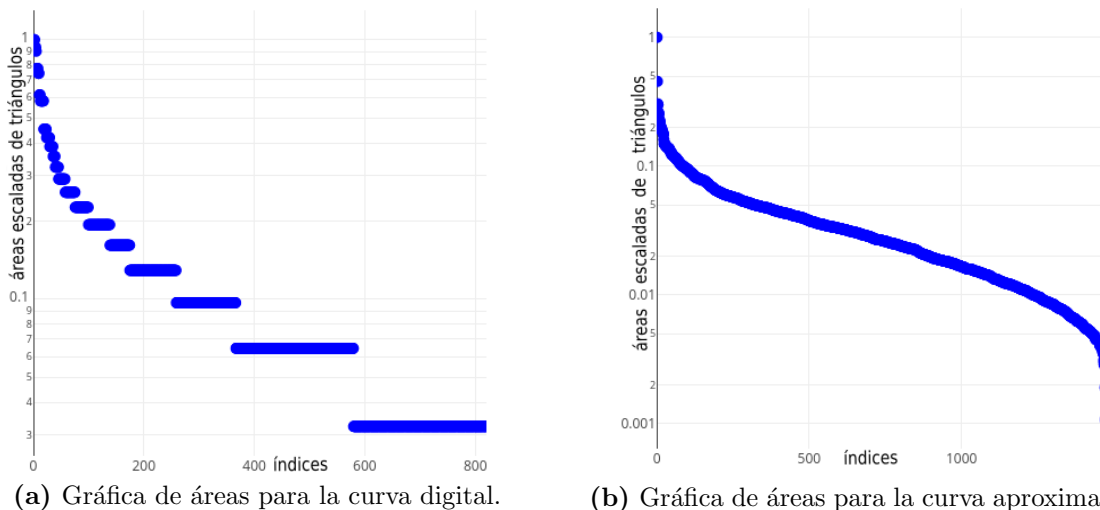
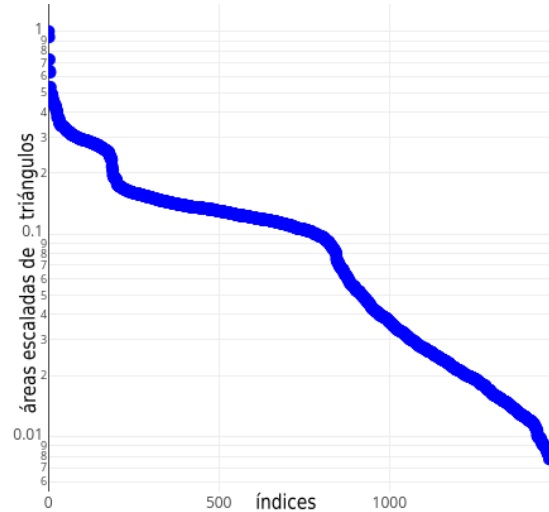


Figura 3.15: Gráficas en escala logarítmica de las áreas escaladas y ordenadas de los triángulos del contorno `ray10` y su polígono de perímetro mínimo con factor $\delta = 3 \times 10^{-4}$.

Generamos polígonos de perímetro mínimo para el contorno de la mantarraya con los factores $\delta = 5 \times 10^{-5}$ y $\delta = 0.001$. En las Figs. 3.16 y 3.17 mostramos un acercamiento a estos polígonos en la cola de la mantarraya así como sus gráficas de áreas correspondientes. Observamos que para $\delta = 5 \times 10^{-5}$ se preserva la forma, pero falta reducir ruido; mientras que para $\delta = 0.001$ se reduce el ruido pero no se preserva la forma. Notamos un cambio en las gráficas de las áreas.



(a) Acercamiento al polígono de perímetro mínimo en color rojo.

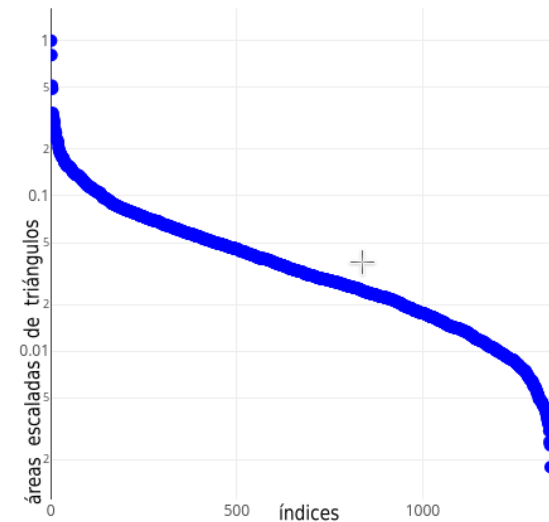


(b) Gráfica de áreas del polígono de perímetro mínimo.

Figura 3.16: Reducción de ruido para la curva digital ray10 con factor $\delta = 5 \times 10^{-5}$.



(a) Acercamiento al polígono de perímetro mínimo en color rojo.



(b) Gráfica de áreas del polígono de perímetro mínimo.

Figura 3.17: Reducción de ruido para la curva digital ray10 con factor $\delta = 0.001$.

3.3. Nuevos Métodos para Aproximación de Polígonos

Sea P un polígono simple orientado con vértices $v_i, i = 1, \dots, n$. Formamos los triángulos orientados T_i que se forman por tres vértices consecutivos v_{i-1}, v_i, v_{i+1} , donde $v_0 = v_n$ y $v_{n+1} = v_1$. Vértices aproximadamente colineales de P se identifican y remueven usando medidas geométricas de los triángulos T_i .

3.3.1. Criterio de las Áreas

Motivados por el concepto de ϵ -convexidad para mallas [11], empleamos las áreas a_i de los triángulos T_i . Denotamos por $\bar{\alpha}$ al área promedio de los triángulos de P . Una sugerencia para detectar vértices aproximadamente colineales de P es identificar los triángulos con área más pequeña que $\bar{\alpha}$.

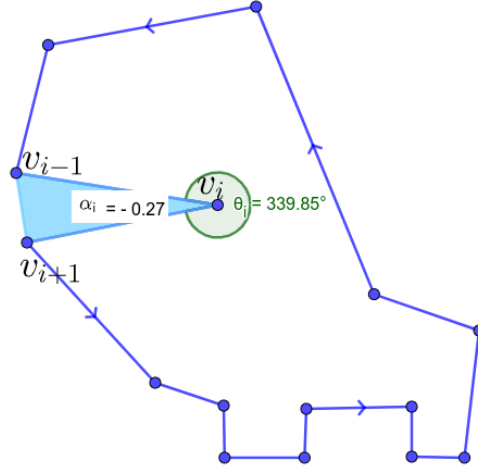


Figura 3.18: Área α_i del triángulo $v_{i-1}v_iv_{i+1}$ y ángulo interior θ_i de v_i en polígono orientado.

Escalamos las áreas de los triángulos entre su área promedio para tener un criterio de colinealidad independiente de la escala. Además, introducimos un factor $\epsilon > 0$ en el área promedio para conservar mejor la forma del polígono original.

Detectamos que el vértice v_i es aproximadamente colineal si se cumple lo siguiente:

$$\frac{a_i}{\bar{\alpha}} < \epsilon. \quad (3.3)$$

Podemos eliminar más vértices ponderando las áreas de los triángulos. La elección adecuada de los pesos nos permite simplificar más el contorno sin perder la forma. En nuestro caso ponderamos por el *seno* del ángulo interior que forman los vértices del polígono. Esto nos permite resaltar los triángulos pequeños con ángulos críticos.

Sea α_i el doble del área del triángulo T_i , y sea θ_i el ángulo interior del vértice v_i en el polígono P , Fig. 3.18. Denotamos:

$$\zeta_i = \alpha_i \cdot |\sin(\theta_i)|, \quad i = 1, \dots, n. \quad (3.4)$$

Para obtener un criterio de colinealidad independiente de la escala del contorno, reescalamos las áreas ponderadas por del área promedio de los triángulos.

Criterio de áreas. Dado un umbral $0 < \epsilon \ll 1$, eliminar los vértices v_i que cumplan la siguiente desigualdad:

$$\frac{\zeta_i}{\bar{\alpha}} < 2\epsilon. \quad (3.5)$$

3.3.2. Criterio de los Radios

Identificamos puntos colineales mediante el cociente del *inradio* r_i entre el *circunradio* R_i de los triángulos $v_{i-1}v_iv_{i+1}$. El valor máximo del cociente es $1/2$ y se alcanza cuando el triángulo es equilátero, mientras que su valor mínimo es cero cuando los vértices son colineales, por lo que este cociente es una medida independiente de la escala del contorno.

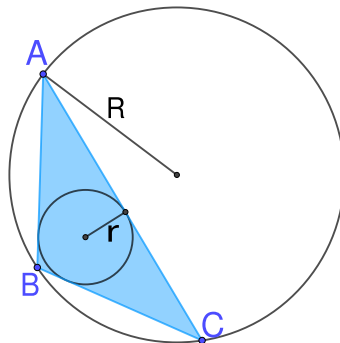


Figura 3.19: Inradio r y circunradio R del triángulo ABC .

Criterio de radios. Dado un umbral $0 < \rho \ll 1$, eliminar los vértices v_i que cumplan la siguiente desigualdad:

$$\frac{r_i}{R_i} < \rho.$$

El Teorema de Carnot nos provee de una manera eficiente para calcular el cociente del inradio entre el circunradio.

Teorema 3.1 (Carnot). *La suma de las distancias del circuncentro de un triángulo ABC a sus tres lados es igual a la suma del circunradio R con el inradio r , esto es,*

$$R \cos(A) + R \cos(B) + R \cos(C) = r + R.$$

La identidad del Teorema de Carnot se puede escribir como:

$$\frac{r}{R} = \cos(A) + \cos(B) + \cos(C) - 1.$$

Observaciones 3.3:

- ☞ De manera experimental elegimos los valores de los umbrales de modo que preservemos la forma. Recomendamos usar valores del umbral en el intervalo $10^{-4} \leq \epsilon \leq 10^{-2}$ tanto para el *criterio de áreas* como en el *criterio de los radios*.
- ☞ Ambos criterios se emplean de manera recursiva en un bucle con el mismo valor del umbral. En cada paso eliminamos todos los vértices que cumplen el criterio correspondiente y aplicamos el mismo criterio al nuevo polígono hasta que ningún vértice lo cumpla.
- ☞ Ambos criterios se pueden extender a regiones múltiplemente conexas eliminando puntos en cada agujero por separado.

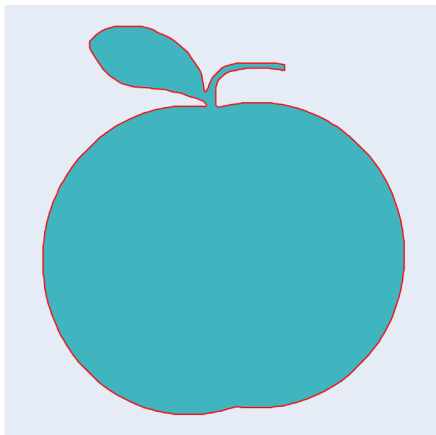
3.3.3. Simplificación con Reducción de Ruido

Podemos simplificar y reducir el ruido de contornos poligonales si aplicamos métodos de simplificación al polígono de perímetro mínimo. Para ello usamos los criterios de las áreas y de los radios. Fijamos las tolerancias de estos criterios y sólo ajustamos el factor para hallar el polígono de perímetro mínimo. Elegimos los valores de los umbrales de manera experimental para preservar la forma. De este modo, el método que usamos para simplificar y reducir el ruido de polígonos es como sigue:

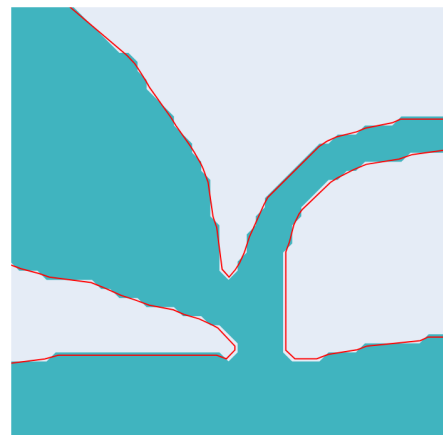
1. Eliminar puntos colineales con el criterio de las áreas con tolerancia $\epsilon = 10^{-4}$.
2. Hallar el polígono de perímetro mínimo con factor $10^{-4} \leq \delta \leq 10^{-3}$.
3. Eliminar puntos usando el criterio de los radios con tolerancia $\rho = 10^{-4}$.

Este método es independiente de la escala. Si el polígono original no tiene ruido, sugerimos usar el factor $\delta = 10^{-4}$. Mientras que si el ruido predomina, sugerimos aumentar el factor a $\delta = 10^{-3}$. Aunque este método no conserva los vértices originales, puede preservar la forma de polígonos con alto nivel de detalle como hemos notado en nuestras pruebas.

Ejemplo 3.5. Consideramos la curva digital `apple03` de 864 puntos de la colección MPEG7CS. Usamos el método de tres pasos con factor $\rho = 10^{-3}$ en el polígono de perímetro mínimo para reducir el ruido y simplificar este contorno a 343 puntos, Fig. 3.20. Aunque no conservamos los puntos originales, suprimimos los picos pequeños y eliminamos ruido.



(a) Polígono original y simplificado.



(b) Acercamiento.

Figura 3.20: Reducción de ruido para el contorno `apple03` usando el *método de tres pasos* con factor $\rho = 10^{-3}$. El polígono simplificado es el contorno rojo.

Ejemplo 3.6. Simplificamos el contorno poligonal del Lago de Chapala con 7275 puntos usando nuestro método de tres pasos. Removimos 1365 puntos preservando la forma usando el factor $\delta = 3 \times 10^{-4}$ al generar el polígono de perímetro mínimo. Mostramos el contorno simplificado en las Figuras 3.21 y 3.22.



Figura 3.21: Contorno simplificado en color rojo del Lago de Chapala por nuestro *método de tres pasos*.



(a) Primer acercamiento.



(b) Segundo acercamiento.

Figura 3.22: Acercamiento al contorno simplificado del Lago de Chapala.

3.3.4. Métodos Modificados

Si bien podemos simplificar y reducir el ruido de varios polígonos con nuestro método de tres pasos, en algunos casos se requiere que los vértices del polígono con ruido suprimido sean vértices del polígono original. Hemos encontrado algunos casos donde podemos perder la forma cuando no movemos los vértices del polígono. En particular, esta dificultad se presenta en algunas curvas digitales con ruido.

Las curvas extraídas de imágenes digitales se llaman *curvas digitales*. Estas curvas son contornos poligonales orientados con vértices de coordenadas enteras. Tanto el *criterio de las áreas* como el *criterio de los radios* pueden eliminar puntos relevantes de curvas

digitales. La eliminación de puntos debe ser gradual conforme aumenta la tolerancia, en cambio varios puntos se eliminan con pequeños cambios en las tolerancias de estos criterios. En consecuencia, la forma de la curva digital se puede perder. Así que modificamos nuestros métodos para simplificar polígonos.

Modificamos los *criterios de áreas y radios* para presentar un método robusto que pueda simplificar polígonos con alto nivel de detalle y reducir el ruido, a saber, el *método modificado de tres pasos*.

Criterio modificado de las áreas. Visvalingam & Whyatt [136] han revisado las implicaciones y ventajas de ponderar las áreas de los triángulos en el *Algoritmo de Visvalingam*. Polígonos con alto nivel de detalle e incluso con ruido se pueden simplificar considerablemente con la elección adecuada de los pesos. Nuestro criterio de las áreas tiene este enfoque, pero la eliminación de puntos no es secuencial.

Ahora, vamos a remover sólo el vértice con el triángulo más pequeño en cada paso. Esto nos permite eliminar vértices de manera gradual al aumentar el valor del umbral. En cambio, si eliminamos varios vértices, podemos perder la forma de la figura con cambios pequeños en el valor del umbral. Más aún, en cada paso basta recalcular las áreas de los triángulos asociados a los dos vértices vecinos, lo que nos permite reducir el costo computacional.

Criterio modificado. Dado un umbral $\epsilon > 0$, remover el vértice v_j con índice

$$j = \arg \min_{i \in J} \zeta_i$$

si se cumple la siguiente desigualdad:

$$\zeta_j < \bar{\alpha} \cdot \epsilon, \quad (3.6)$$

donde

ζ_i es el área ponderada de un triángulo dada por (3.4),

J es el conjunto de índices de los vértices del paso actual,

$\bar{\alpha}$ es el área promedio de los triángulos del polígono inicial.

Determinamos el valor del umbral ϵ de manera experimental. En nuestras pruebas hemos simplificado polígonos con $0.1 \leq \epsilon \leq 1$. Más aún, podemos suprimir el ruido sin perder la forma con $1 \leq \epsilon \leq 20$, incluso en curvas digitales.

Criterio modificado de los radios. El criterio de los radios se puede modificar para que sólo el vértice con el cociente más pequeño se elimine en cada paso. No obstante, cuando queremos reducir el ruido de curvas digitales, hemos observado que la forma se puede perder aún con esta modificación. En vez de usar el cociente de los radios, usamos su producto para superar las dificultades que se presentan en la reducción de ruido.

A partir del producto del *inradio* con el *circunradio* obtenemos una *métrica* para simplificar polígonos complicados. Usamos el producto de estos radios porque es una cota para el área del triángulo correspondiente. La cota que usamos está basada en la *desigualdad de Blundon*, la cual nos dice que el *semiperímetro* de un triángulo está acotado por la suma del doble de su *circunradio* con su *inradio*.

Teorema 3.2 (Desigualdad de Blundon [36, 142]). *Para cualquier triángulo con inradio r , circunradio R y perímetro p se cumple la siguiente desigualdad:*

$$\frac{p}{2} \leq 2R + (3\sqrt{3} - 4)r.$$

La igualdad se obtiene si y sólo si el triángulo es equilátero. Si además el triángulo no tiene ángulos obtusos, se cumple la siguiente desigualdad:

$$2R + r \leq \frac{p}{2}.$$

La igualdad se obtiene si y sólo si el triángulo forma un ángulo recto.

El área a , el perímetro p y el inradio r de un triángulo se relacionan por la identidad

$$a = r \cdot \frac{p}{2}.$$

Entonces por la *desigualdad de Blundon* tenemos la siguiente cota superior para el área:

$$a < r \cdot (2R + 1.2r)$$

Más aún, para triángulos sin ángulos obtusos tenemos la siguiente cota inferior:

$$a \geq r \cdot (2R + r).$$

Lo anterior nos sugiere emplear el producto $2rR$ como una aproximación del área a . Sin embargo, $rR \gg a$ en triángulos obtusos. Para compensar esto, ponderamos el producto de los radios con el *seno* del ángulo interior. Así en cada triángulo formado por tres vértices consecutivos con *inradio* r_i y *circunradio* R_i calculamos la siguiente *métrica*:

$$\rho_i = 2r_i R_i \sin(v_i). \tag{3.7}$$

Además de ponderar el producto de los radios, necesitamos que su valor sea independiente de la escala del polígono. Como este producto es una aproximación del área de los triángulos, escalamos por el área promedio \bar{a} de los triángulos.

Ponderamos y escalamos el producto de los radios. Sin embargo, esto no es suficiente para preservar la forma de polígonos con ruido. Al igual que el *criterio de las áreas*, eliminamos puntos secuencialmente. En cada paso quitamos el vértice con la menor *métrica* ρ_i si su valor está por debajo de un umbral.

Criterio modificado. Dado un umbral $\epsilon > 0$, removemos el vértice v_j con índice

$$j = \arg \min_{i \in J} \rho_i$$

si se cumple la siguiente desigualdad:

$$\rho_j < \bar{\alpha} \cdot \epsilon, \quad (3.8)$$


donde


ρ_i es el producto ponderado de los radios dado por (3.7),

J es el conjunto de índices de los vértices del paso actual,

$\bar{\alpha}$ es el área promedio de los triángulos del polígono inicial.

Observaciones 3.4:

 Como el producto ponderado de los radios es una aproximación de las áreas de los triángulos, podemos emplear el mismo rango de valores para el umbral en los *criterios modificados de áreas y radios*.

 Podemos calcular de manera eficiente el producto del *inradio* r por el *circuncentro* R de un triángulo con lados $\ell_i, i = 1, 2, 3$ mediante la siguiente identidad [57]:

$$2r \cdot R = \frac{\ell_1 \cdot \ell_2 \cdot \ell_3}{\ell_1 + \ell_2 + \ell_3}.$$

Método modificado de tres pasos. En nuestro *método de tres pasos* para aproximación de polígonos reemplazamos el *criterio de los radios* por el *criterio modificado de áreas*, y realizamos la iteración de manera secuencial. De este modo el método modificado es como sigue:

1. Eliminar puntos colineales por el criterio de áreas modificado con umbral $\epsilon = 0.1$.
2. Generar polígono de perímetro mínimo.
3. Simplificar polígono por el criterio de áreas modificado.

Podemos simplificar polígonos con alto nivel de detalle y reducir el ruido de curvas digitales sin perder la forma. Elegimos los valores de los umbrales correspondientes de manera experimental. Para simplificar polígonos sin ruido:

- tomamos el umbral $\delta = 10^{-5}$ en el problema del polígono de perímetro mínimo,
- tomamos el umbral $\epsilon = 0.1$ en el *criterio modificado de áreas*.

Por otra parte, para reducción de ruido:

- tomamos el umbral $\delta = 10^{-4}$ en el problema del polígono de perímetro mínimo,
- tomamos el umbral del *criterio modificado de áreas* en el intervalo $3 \leq \epsilon \leq 20$.

Observaciones 3.5:

- ☞ En los métodos modificados de áreas, radios y tres pasos notamos que la reducción de ruido ocurre de manera gradual a diferencia de los métodos sin las modificaciones.
- ☞ Los métodos anteriores pueden aproximar la frontera de regiones con agujeros. Se aplican a la frontera exterior y a cada agujero por separado.
- ☞ Para saber si la curva aproximada preserva la forma del polígono original realizamos una comparación por inspección visual. Adicionalmente, podemos verificar que se conserven puntos distintivos y que el cambio relativo de las áreas no rebase un umbral dado. Sin embargo, un criterio para decidir cuando se preserva la forma con el nivel de detalle deseado dependerá de la aplicación específica.

Ejemplo 3.7. Usamos nuestros métodos modificados para simplificar el contorno poligonal Presa La Amistad de 10445 puntos y los comparamos con el *Algoritmo de Visvalingam* con escalamiento. En la Tabla 3.2 se muestran los puntos que removimos así como los valores de los factores correspondientes. Mostramos el contorno simplificado que generamos con nuestro método de tres pasos modificado en la Fig. 3.23. Comparamos con los otros métodos en un acercamiento que realizamos en la Fig. 3.24 para apreciar mejor las diferencias.

Método	Pts. Removidos	Factor
Visvalingam con escalamiento	8690	$\epsilon = 1$
Áreas Modificado	8904	$\epsilon = 1$
Radios Modificado	8702	$\epsilon = 1$
Tres Pasos Modificado	8684	$\delta = 10^{-5}, \epsilon = 0.1$

Tabla 3.2: Puntos eliminados y factores empleados en los métodos modificados para simplificar el contorno Presa La Amistad.

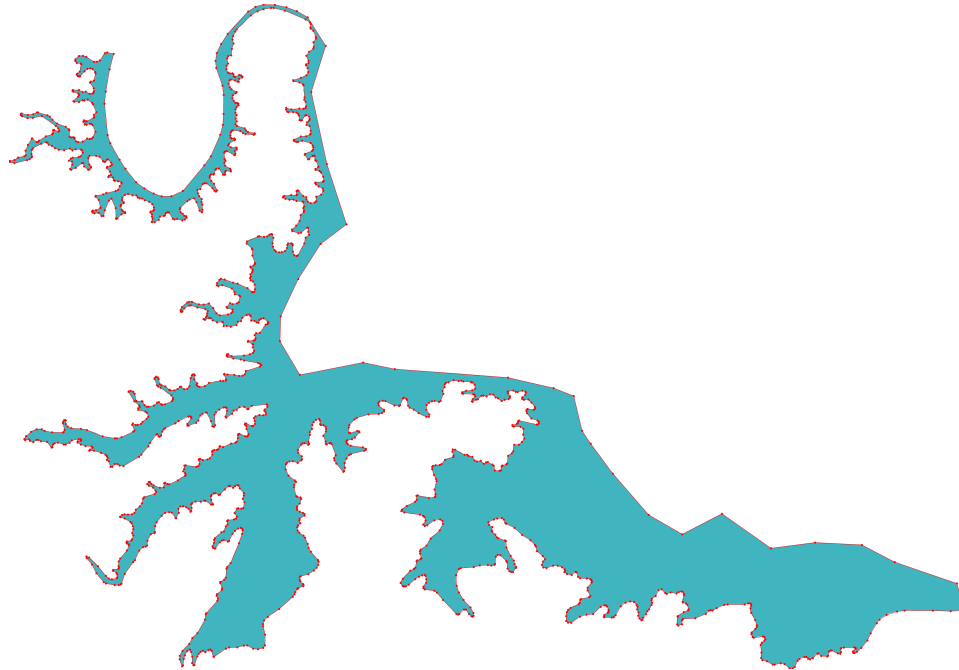
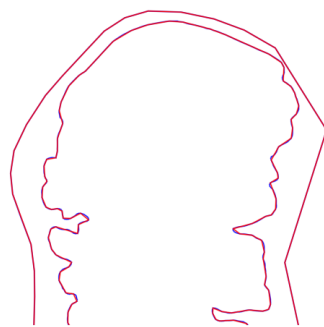
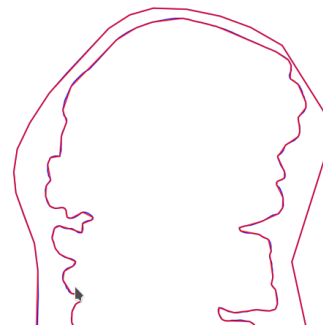


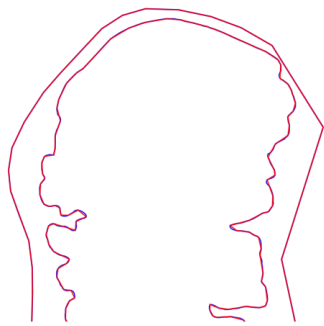
Figura 3.23: Simplificación del contorno Presa La Amistad por el *método modificado de tres pasos*. El contorno en rojo es el polígono simplificado.



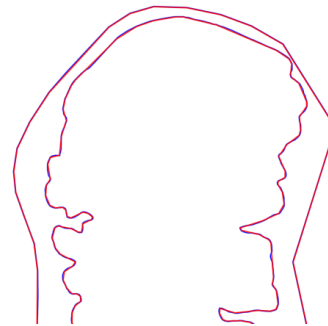
(a) Visvalingam escalado.



(b) Áreas modificado.



(c) Radios modificado.



(d) Tres pasos modificado.

Figura 3.24: Comparación de métodos para la simplificación del contorno Presa La Amistad. Mostramos un acercamiento del polígono original en azul contra el polígono simplificado en rojo.

Ejemplo 3.8. Usamos nuestros métodos modificados para simplificar el contorno poligonal Laguna Honda de 3492 puntos en la frontera exterior con 8 agujeros que en total tienen 227 puntos. Comparamos los métodos con el *Algoritmo de Visvalingam* con escalamiento. En la Tabla 3.3 se muestran los puntos que removimos así como los valores de los factores correspondientes. Mostramos el contorno simplificado que generamos con nuestro método de tres pasos modificado en la Fig. 3.26. Comparamos con los otros métodos en un acercamiento que realizamos en la Fig. 3.25 para apreciar mejor las diferencias.

Método	Pts. Removidos	Factor
Visvalingam con escalamiento	1609	$\epsilon = 1$
Áreas Modificado	2327	$\epsilon = 1$
Radios Modificado	1610	$\epsilon = 1$
Tres Pasos Modificado	2067	$\delta = 10^{-5}, \epsilon = 0.1$

Tabla 3.3: Puntos eliminados y umbrales en los métodos para simplificar la Laguna Honda.

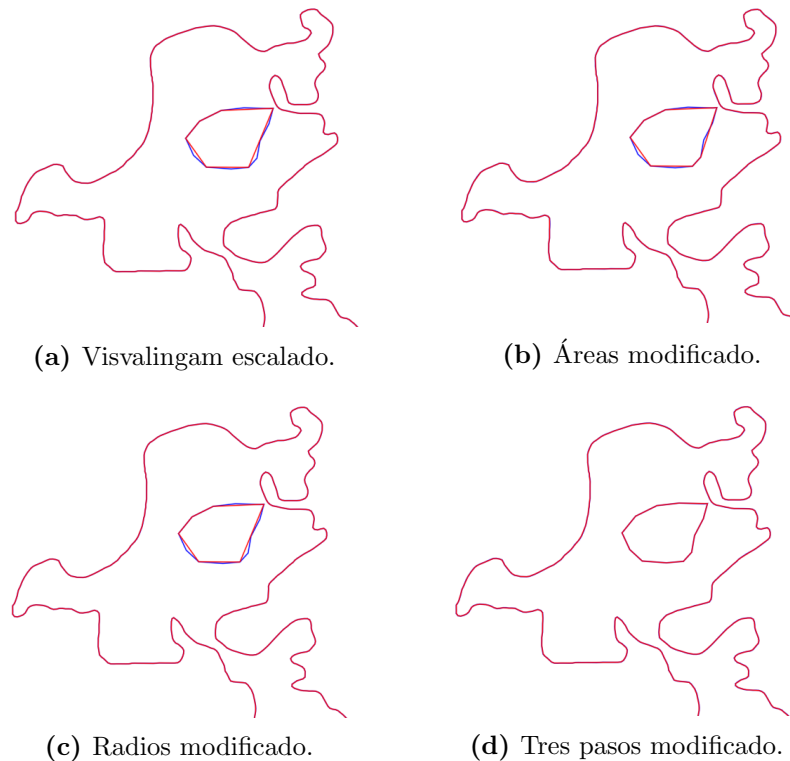


Figura 3.25: Comparación de los métodos para simplificar la Laguna Honda. Mostramos un acercamiento del polígono original en azul contra el polígono simplificado en rojo.



3.4. Resumen del Capítulo

La generación de mallas sobre una región poligonal puede no ser viable cuando su frontera tiene alto nivel de detalle, picos, e incluso ruido, por lo que aproximamos su frontera por curvas más suaves que preserven la forma de la región con una menor cantidad de puntos, para ello abordamos los problemas de simplificación de polígonos y reducción de ruido.

Nuestro grupo UNAMALLA ha empleado el suavizamiento por spline cónicos y simplificación por criterios de colinealidad y detección de puntos dominantes, no obstante estos métodos tienen limitaciones para preservar la forma de polígonos complicados, por lo que

desarrollamos nuevos métodos iterativos que emplean medidas geométricas de los triángulos formados por tres vértices consecutivos, a saber, sus áreas, inradios y circunradios.

Además de simplificar polígonos, podemos detectar ruido con las áreas de sus triángulos, lo que hacemos es inspeccionar la gráfica en escala logarítmica de sus áreas ordenadas en orden decreciente y escaladas por su área promedio contra el conjunto de índices, cuando esta gráfica tiene saltos, el polígono correspondiente tiene ruido. Para reducir el ruido de un polígono v_1, \dots, v_n con perímetro p_0 buscamos el polígono de perímetro mínimo u_1, \dots, u_n en una vecindad de radio δ dada por $\|v_i - u_i\|_\infty \leq p_0 \cdot \delta$ para $i = 1, \dots, n$.

Nuestros métodos pueden simplificar polígonos con alto nivel de detalle y reducir ruido de algunos polígonos, pero tienen dificultades para preservar la forma de *curvas digitales*, así que los modificamos. Motivados por la simplificación de polígonos en Cartografía, eliminamos vértices de manera secuencial, de este modo nuestros métodos son:

- **Criterio modificado de áreas.** En cada paso eliminamos el vértice asociado al triángulo con área ponderada más pequeña si está por debajo de un umbral fijo.
- **Criterio modificado de radios.** En cada paso removemos el vértice asociado al producto ponderado más pequeño de los *inradios* por los *circunradios* si este producto está por debajo de un umbral fijo.
- **Método modificado de tres pasos.** Primero, ocupamos el criterio modificado de áreas con umbral $\epsilon = 0.1$ para eliminar puntos colineales, luego, buscamos el polígono de perímetro mínimo en una vecindad de radio δ para suavizar picos y volvemos a usar el *criterio modificado de áreas*, pero con otro valor del umbral ϵ .

Nuestros métodos son independientes de la escala de la región y pueden emplearse en regiones múltiplemente conexas aproximando cada agujero por separado. En particular, en el método modificado de tres pasos simplificamos polígonos sin ruido con la elección de umbrales $\delta = 10^{-5}$ y $\epsilon = 0.1$, mientras que para reducir ruido tomamos $\delta = 10^{-4}$ y $3 \leq \epsilon \leq 20$.

La descomposición de regiones con alto nivel de detalle o ruido en la frontera puede producir un número excesivo de subregiones, y algunas pueden no ser admisibles para generar mallas estructuradas, por lo que primero simplificamos la frontera con los métodos anteriores, y luego separamos la región simplificada. En el siguiente capítulo precisamos cómo deben ser las *subregiones admisibles* y desarrollamos un método para descomponer regiones complicadas en estas subregiones.

En algunas aplicaciones se requiere generar mallas de cuadriláteros con alta calidad sobre regiones poligonales que poseen una geometría complicada. Para este fin generamos mallas estructuradas por bloques. Descomponemos las regiones en subregiones (bloques) y generamos mallas estructuradas en cada bloque. Un paso fundamental es la descomposición de regiones. Necesitamos separar regiones complicadas en *subregiones admisibles* para la generación de mallas estructuradas de alta calidad.

En §4.1 identificamos *regiones admisibles* y mostramos las características de una *descomposición admisible*. Luego, en §4.2 describimos nuestro método para separar regiones complicadas en *subregiones admisibles*. Posteriormente, en §4.4 damos los detalles de nuestra descomposición.

4.1. Regiones y Descomposiciones Admisibles

Queremos identificar regiones poligonales adecuadas para generar mallas estructuradas de un solo bloque con alta calidad. Recordamos que nosotros medimos la calidad de las mallas con el porcentaje de celdas que son aproximadamente rectángulos. Por lo que mallas de alta calidad se pueden generar sobre regiones que sean aproximadamente rectángulos, no obstante estas mallas pueden generarse sobre más regiones, es conveniente examinar algunas características que presentan las regiones donde es complicado generarlas.

Hemos observado que las regiones con ruido en la frontera, bordes puntiagudos o con alta concavidad no son adecuadas para generar mallas de alta calidad. Regiones con ruido en la frontera se aproximan por otras que preservan la forma con menos ruido como vimos en el capítulo anterior. Una vez que reducimos el ruido, procedemos a examinar otras características de las regiones.

La concavidad de las regiones es otro aspecto a considerar en la calidad de las mallas. Podemos generar mallas estructuradas de un solo bloque sobre varias regiones cóncavas, sin embargo estas mallas pueden no tener la calidad deseada, pues un alto porcentaje de las celdas pueden ser elongadas o distorsionadas. Mallas de baja calidad se pueden generar sobre regiones cóncavas que tienen cuellos de botella, ramificaciones o la forma de una espiral. Para distinguir estas regiones, medimos su concavidad.

Sea Ω una región poligonal. La *envolvente convexa* de Ω , denotada por \mathcal{C} , es el polígono convexo más pequeño que contiene a Ω . Una manera de medir la concavidad de la región Ω es comparar su perímetro respecto al de su *envolvente convexa* \mathcal{C} como:

$$\kappa_1(\Omega) = \frac{\text{perímetro}(\Omega) - \text{perímetro}(\mathcal{C})}{\text{perímetro}(\Omega)}. \quad (4.1)$$

En las regiones de alta concavidad tenemos que κ_1 es mayor que un umbral τ_1^+ , mientras que en regiones aproximadamente convexas tenemos que κ_1 es menor que un umbral τ_1^- . Buscamos nuestras *regiones admisibles* entre las *regiones aproximadamente convexas*.

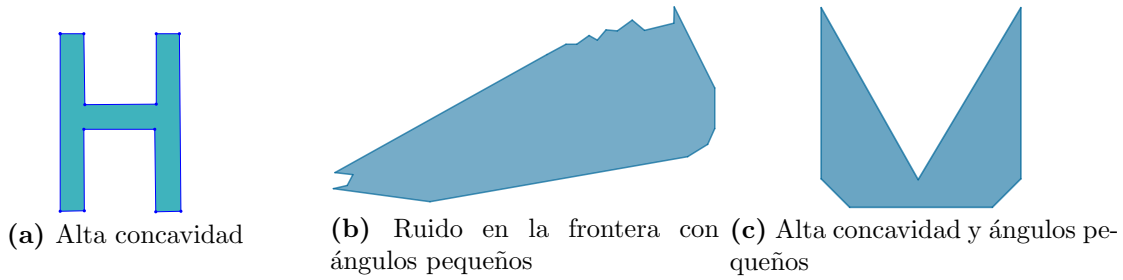


Figura 4.1: Regiones poligonales no admisibles.

Observaciones 4.1:

- ☞ La medida κ_1 es independiente de la escala de la región: $0 \leq \kappa_1(\Omega) < 1$.
- ☞ Sugerimos los valores experimentales $\tau_1^+ = 0.5$ para regiones de alta concavidad y $\tau_1^- = 0.1$ para regiones aproximadamente convexas.
- ☞ En §4.4 introducimos otras *medidas de concavidad*.

En un principio pensábamos que las regiones convexas eran ideales para generar mallas estructuradas de un solo bloque. Sin embargo, incluso las mallas sobre regiones convexas pueden ser de baja calidad. Estas regiones pueden tener bordes puntiagudos donde varias celdas no tienen la calidad deseada. Los bordes puntiagudos ocurren en ángulos interiores críticos. Dados los ángulos θ_- y θ_+ , un ángulo interior θ de un polígono es un *ángulo crítico* si $\theta < \theta_-$ o bien $\theta > \theta_+$. Los ángulos críticos pueden ser agudos o mayores que 180° . Los vértices con ángulos interiores mayores que 180° se llaman *vértices reflex*.

Pocas celdas de baja calidad se pueden generar alrededor de vértices reflex. En cambio, un alto porcentaje de celdas con baja calidad se pueden generar alrededor de ángulos agudos críticos.

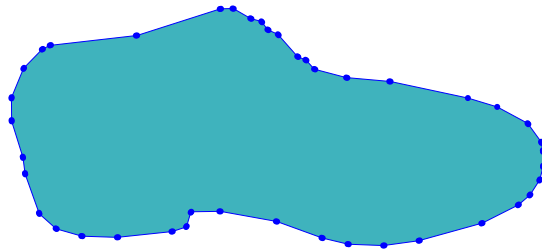
Observaciones 4.2: Sugerimos tomar $\theta_- = 40^\circ$ para identificar ángulos agudos críticos.

Los *ángulos críticos* y la *concavidad* tanto de la región delimitada por la frontera exterior como en los agujeros influyen en la calidad de las mallas. Generar mallas con alta calidad sobre regiones con agujeros puede ser difícil, incluso sobre regiones con frontera exterior sencilla y un agujero.

En vista de lo anterior, proponemos condiciones suficientes para identificar las regiones donde podemos generar mallas estructuradas de un solo bloque con alta calidad. Decimos que una región poligonal \mathcal{P} es una *región admisible* si cumple las siguientes características:

- * \mathcal{P} es simplemente conexa,
- * \mathcal{P} es aproximadamente convexa,
- * la frontera de \mathcal{P} no tiene ruido,
- * \mathcal{P} no tiene *ángulos críticos* agudos.

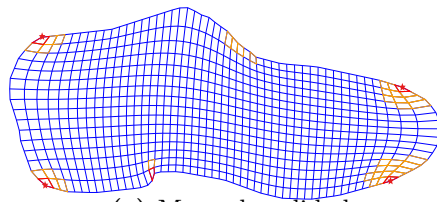
Ejemplo 4.1. Generamos tres regiones poligonales admisibles C_1, C_2 y C_3 con base a las tres propiedades que pedimos, así como sus mallas estructuradas de alta calidad. Mostramos sus mapas de calidad y medidas de la región en las Figs. 4.2, 4.3 y 4.4.



(a) Región

θ mín	143°
$\frac{\text{núm. reflex}}{\text{núm. vértices}}$	0.24
κ_1	2.06×10^{-2}

(b) Medidas de la Región

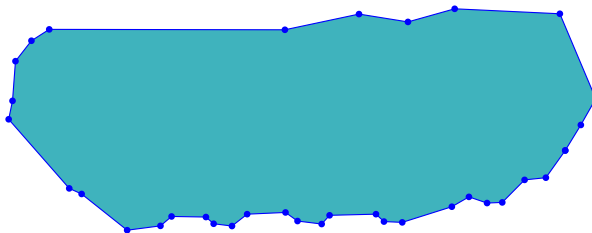


(c) Mapa de calidad

% celdas	calidad
94.8 %	$\mu \geq 0.8$
4.6 %	$0.4 \leq \mu < 0.8$
0.6 %	$\mu < 0.4$

(d) Medidas de Calidad

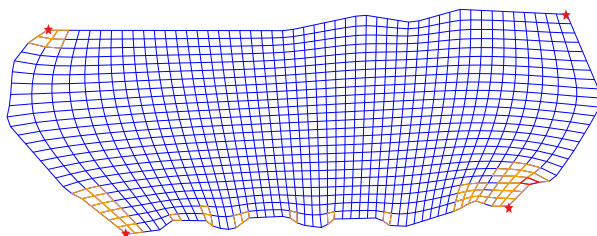
Figura 4.2: Mapa de calidad para malla estructurada 50×17 sobre región C_1 con medidas de la región.



(a) Región

θ mín	115°
$\frac{\text{núm. reflex}}{\text{núm. vértices}}$	0.34
κ_1	4.84×10^{-2}

(b) Medidas de la Región

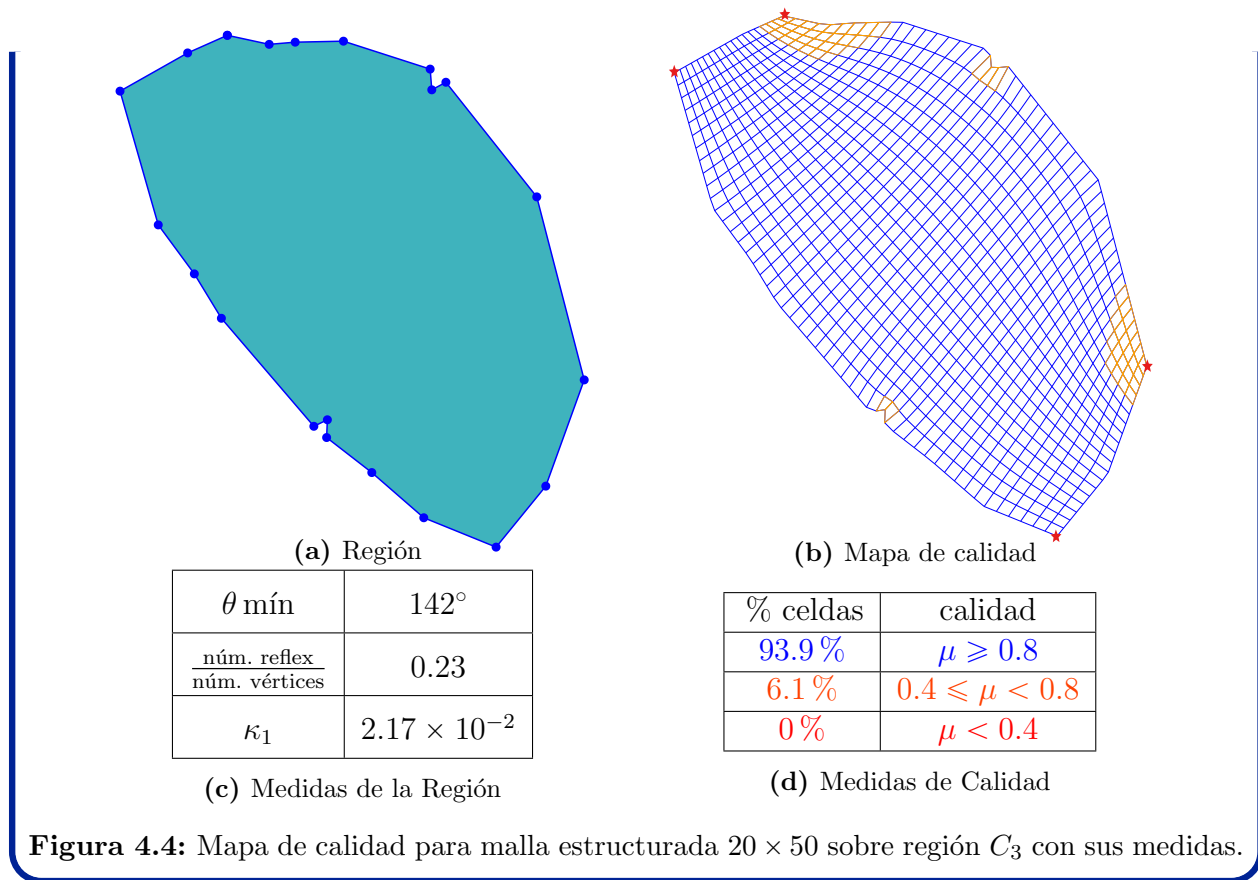


(c) Mapa de calidad

% celdas	calidad
95.6 %	$\mu \geq 0.8$
4.3 %	$0.4 \leq \mu < 0.8$
0.1 %	$\mu < 0.4$

(d) Medidas de Calidad

Figura 4.3: Mapa de calidad para malla estructurada 50×25 sobre región C_2 con medidas de la región.



Las regiones complicadas pueden descomponerse de diferentes maneras. A nosotros nos interesa descomponerlas en *subregiones admisibles* para generar mallas de alta calidad. Describimos las características que debe satisfacer una descomposición de regiones para generar nuestras mallas.

Las celdas de las mallas deben empatar en las intersecciones de las subregiones. Si las subregiones se traslapan, se requiere que tanto puntos frontera como puntos interiores de las mallas coincidan en las intersecciones. Queremos subregiones que no se traslapan, excepto en sus fronteras, de ese modo basta que los puntos frontera de las mallas coincidan en las intersecciones. Así que si dos subregiones se intersectan, necesitamos que su intersección esté contenida en sus fronteras.

Además de que las celdas empaten, queremos mallas sin singularidades interiores. Sin embargo, no cualquier descomposición de la región es adecuada para generar estas mallas. Las subregiones se pueden intersectar en puntos interiores de la región que representan singularidades interiores de la malla global. Para evitar estas singularidades incorporamos algunas condiciones a la descomposición de la región. Empleamos curvas simples contenidas en el interior de la región que unen dos puntos de la frontera. Estas curvas se llaman líneas de corte. Una condición suficiente para generar mallas sin singularidades interiores es que las intersecciones de las subregiones sean líneas de corte de la región.

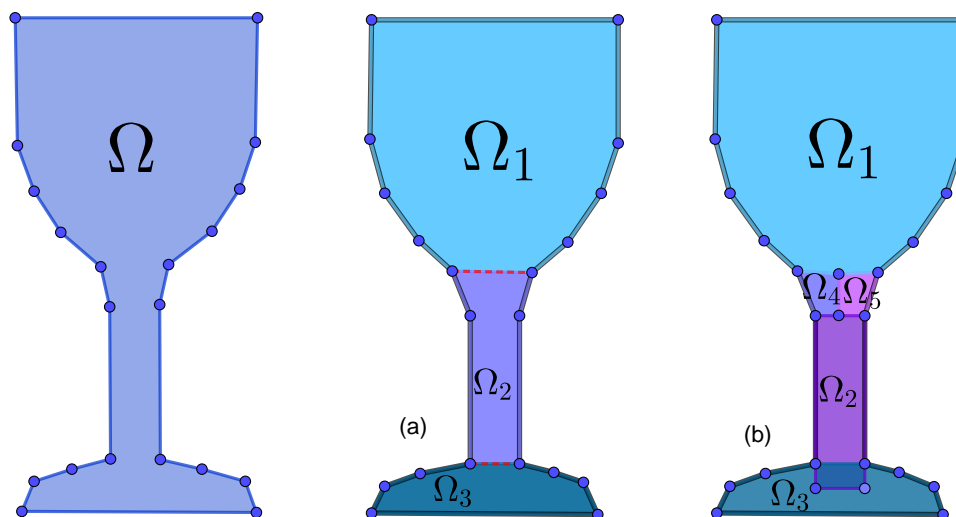


Figura 4.5: Descomposición de una región poligonal Ω en subregiones Ω_i : (a) *descomposición admisible*; (b) descomposición no admisible porque se agregan puntos interiores y las subregiones se traslapan.

Descomponemos regiones para generar mallas sin singularidades interiores. No obstante, algunas singularidades en la frontera pueden afectar la calidad de la malla. Varias subregiones se pueden intersectar en un solo punto de la frontera. Las mallas sobre esas subregiones pueden tener varias celdas distorsionadas alrededor de ese punto. Necesitamos que las subregiones no se intersecten en puntos aislados. Para ello es suficiente que sus intersecciones sean exactamente líneas de cortes de la región.

Con base a lo anterior, precisamos cuando una descomposición de regiones es *admisible* para generar mallas estructuradas por bloques de alta calidad sin singularidades interiores. Sea Ω una región poligonal, decimos que una colección finita $\{\Omega_k\}_{k=1}^n$ de subregiones es una *descomposición admisible* de Ω si cumple lo siguiente:

- Ω_k es una región admisible para $k = 1, \dots, n$,
- $\Omega = \cup_{k=1}^n \Omega_k$,
- si Ω_i y Ω_j no son ajenas, $\Omega_i \cap \Omega_j$ es una *línea de corte* de Ω .

Tenemos condiciones suficientes para identificar *descomposiciones admisibles* de regiones respecto a la generación de mallas con alta calidad. En la siguiente sección procedemos a obtener estas descomposiciones de regiones.

4.2. Métodos para Descomposición de Regiones

La descomposición de una región es un problema usual en visión artificial y reconocimiento de patrones. Revisamos algunas descomposiciones de la literatura:

- * La descomposición de una región puede basarse en medidas geométricas. Scanlan [119] introduce medidas para detectar bordes puntiagudos de un polígono. La región se separa por cortes en subregiones con base a las medidas propuestas, no obstante estas subregiones pueden ser elongadas y puntiagudas. Más aún, estas medidas no se han extendido a polígonos con agujeros, Fig. 4.6.

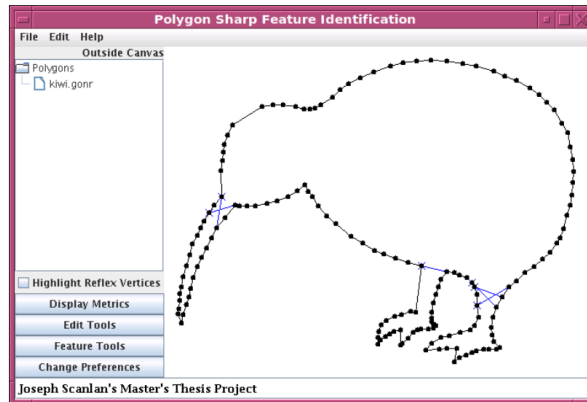


Figura 4.6: Descomposición del polígono Kiwi por detección de bordes puntiagudos [119].

- * La descomposición de una región plana puede basarse en la concavidad. Lien [68] genera una *descomposición aproximadamente convexa*, esto es, la región se separa en subregiones aproximadamente convexas con base a medidas de concavidad. Esta descomposición depende de las medidas y sus tolerancias, Fig. 4.7. Más aún, esta descomposición se puede extender a regiones con agujeros.

Liu et al. [71] modifican la descomposición aproximadamente convexa en la *descomposición dual*, que usa el complemento de la región respecto a su *envolvente convexa*. Esta descomposición es robusta al ruido de la frontera, no obstante ambas descomposiciones pueden no ser adecuadas para generar mallas de calidad, pues las subregiones pueden tener bordes puntiagudos e incluso generar triángulos que no son necesarios.

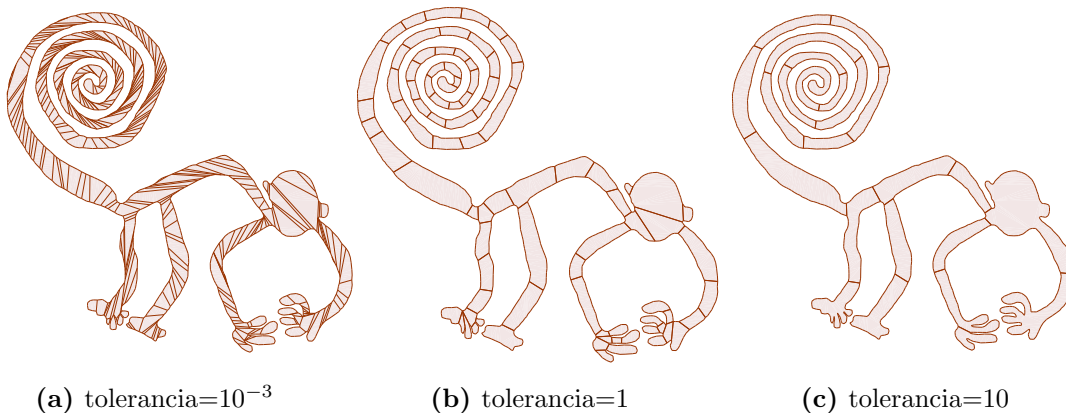


Figura 4.7: Descomposición aproximadamente convexa del mono de las líneas de Nazca [68].

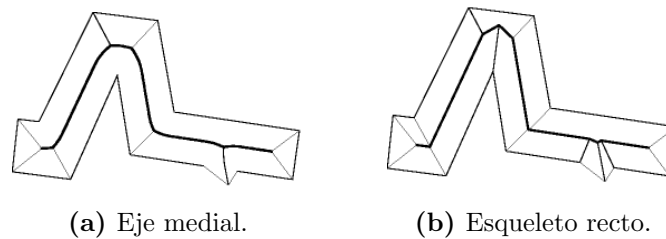


Figura 4.8: Diferentes esqueletos de polígonos [48].



Figura 4.9: Esqueleto recto generado con biblioteca CGAL.

- * La colección de puntos de un polígono que son cercanos a más de una arista forman el *eje medial*. Este lugar geométrico induce una descomposición de la región, sin embargo las fronteras de las subregiones no necesariamente son curvas poligonales, ya que el *eje medial* de un polígono puede tener *arcos parabólicos*, Fig. 4.8(a).

Una aproximación del eje medial es el *esqueleto recto* introducido por Aichholzer y Aurenhammer [4]. Al igual que el *eje medial*, el esqueleto recto es un *árbol*, ver §4.3, sus raíces son los vértices de la región poligonal y todas sus aristas son segmentos de recta [133], Fig. 4.8(b). Sin embargo, el *esqueleto recto* de un polígono complicado puede dar lugar a una descomposición con más subregiones de las necesarias, Fig. 4.9, y por consiguiente, la generación de mallas puede ser computacionalmente costosa. Más aún, el *esqueleto recto* pueden introducir singularidades interiores.

- * Zeng et al. [154] emplean los cruces relevantes del eje medial así como la curvatura de la frontera para separar la región por cortes, que eligen en las intersecciones de la frontera con círculos centrados en los cruces, Fig. 4.10.

Por otra parte, Papanelopoulos [95] desarrolla un esquema que emplea el eje medial junto con criterios de concavidad para elegir los cortes. No obstante, ambos métodos [95, 154] sólo se han usado para algunas regiones sencillas y no se han extendido a regiones con agujeros.

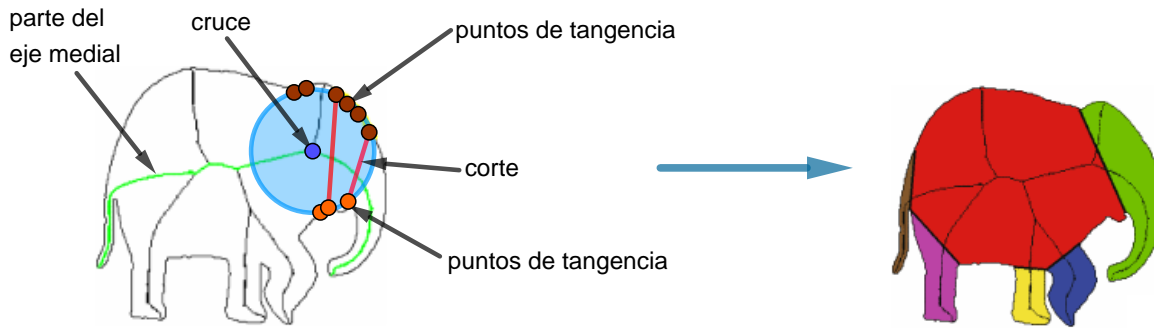


Figura 4.10: Descomposición por cortes de una región plana que requiere del eje central [154].

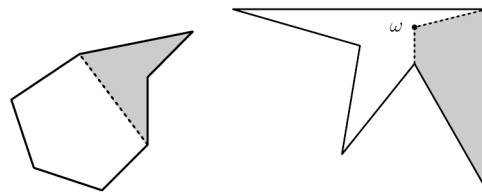


Figura 4.11: Descomposición en cuadriláteros por extracción secuencial de cuadriláteros [105].

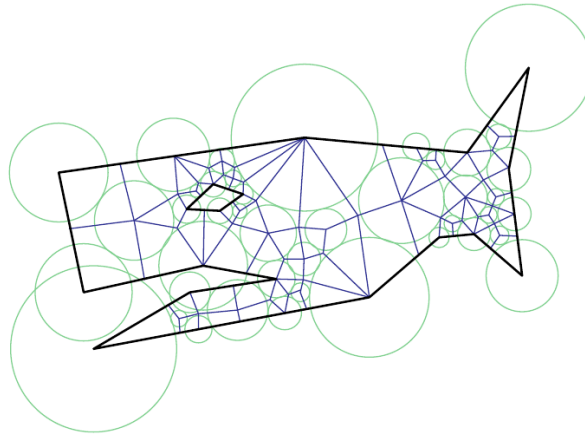


Figura 4.12: Descomposición en cuadriláteros por empaquetamiento de círculos [15].

- * Randrianarivory [105] diseña un algoritmo para extraer recursivamente cuadriláteros de un polígono usando dos operaciones, la primera usa cortes, y la otra agrega puntos interiores, Fig. 4.11. Por otra parte, Bern y Eppstein [15] usan empaquetamiento de círculos junto con diagramas de Voronoi para separar regiones poligonales en cuadriláteros, Fig. 4.12. Ambos métodos [15, 105] se pueden extender a regiones con agujeros, pero generan mallas no estructuradas y se han empleado sobre regiones sencillas. Más aún, los cuadriláteros pueden formar bordes puntiagudos y singularidades interiores.

4.2.1. Método para Descomposición Admisible

Los métodos de descomposición anteriores pueden generar bordes puntiagudos, singularidades interiores, un número excesivo de subregiones, o bien sólo se han probado en regiones sencillas. Nosotros empleamos algunas de las ideas expuestas como realizar cortes y medir la concavidad.

La concavidad juega un papel importante en la *descomposición admisible*. Recordamos que las *subregiones admisibles* son *aproximadamente convexas* en el sentido de que tienen una medida de concavidad pequeña. Métodos como la *Descomposición Aproximadamente Convexa* [68] y la *Descomposición Dual* [71] usan *medidas de concavidad*. Nuestra *descomposición admisible* está motivada por estos métodos.

Las subregiones de una *descomposición admisible* se intersectan en líneas de corte. Por lo que es conveniente separar las regiones por cortes. Nosotros realizamos una descomposición por cortes interactiva y recursiva. Separamos regiones poligonales en *subregiones admisibles* como sigue:

1. Separar regiones con agujeros en subregiones sin agujeros, Fig. 4.13.
2. Separar regiones sin agujeros en *subregiones aproximadamente convexas*, Fig. 4.14.
3. Separar regiones aprox. convexas en subregiones sin bordes puntiagudos, Fig. 4.15.

En el Algoritmo 1 resumimos nuestra *descomposición admisible* de regiones poligonales. En cada paso la región se separa por cortes en dos subregiones. Primero, describimos el método para separar regiones sin agujeros ni bordes puntiagudos. La descomposición de regiones con agujeros se puede reducir a este caso. Posteriormente, extendemos la descomposición para manejar bordes puntiagudos.

Algoritmo 1 Descomposición Admisible

```

Rutina DESCOMPOSICIÓN( $\Omega$ )
  if  $\Omega$  tiene agujeros then
     $\Omega_1, \Omega_2 \leftarrow$  DESCOMPOSICIÓN I( $\Omega$ )
    DESCOMPOSICIÓN( $\Omega_1$ )
    DESCOMPOSICIÓN( $\Omega_2$ )
  if  $\Omega$  no cumple criterio de concavidad then
     $\Omega_1, \Omega_2 \leftarrow$  DESCOMPOSICIÓN II( $\Omega$ )
    DESCOMPOSICIÓN( $\Omega_1$ )
    DESCOMPOSICIÓN( $\Omega_2$ )
  if  $\Omega$  tiene más de tres lados y ángulos agudos críticos then
     $\Omega_1, \Omega_2 \leftarrow$  DESCOMPOSICIÓN III( $\Omega$ )
    DESCOMPOSICIÓN( $\Omega_2$ )

```

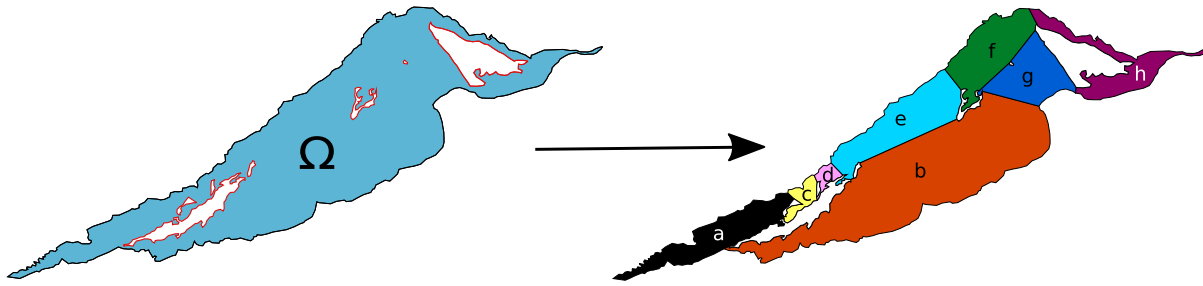


Figura 4.13: Descomposición de la Laguna de Tamiahua con siete agujeros en ocho subregiones simplemente conexas.

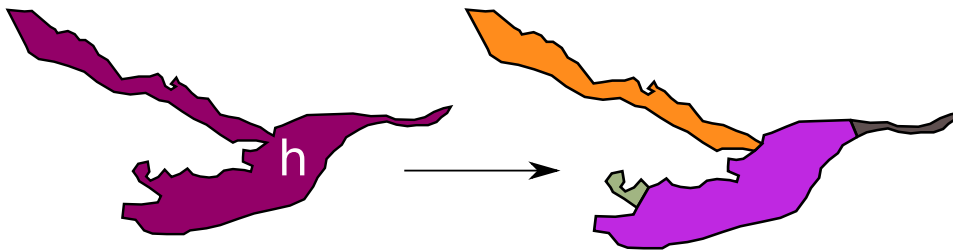


Figura 4.14: Descomposición de una subregión simplemente conexa de la Laguna de Tamiahua en cuatro subregiones aproximadamente convexas.

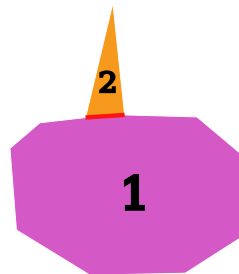
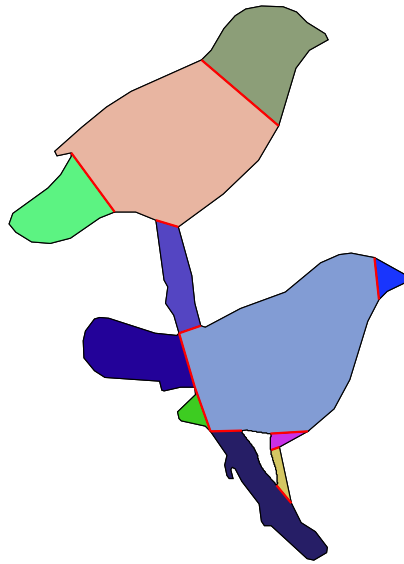


Figura 4.15: Descomposición de región con borde puntiagudo en un triángulo y una subregión sin ese borde.

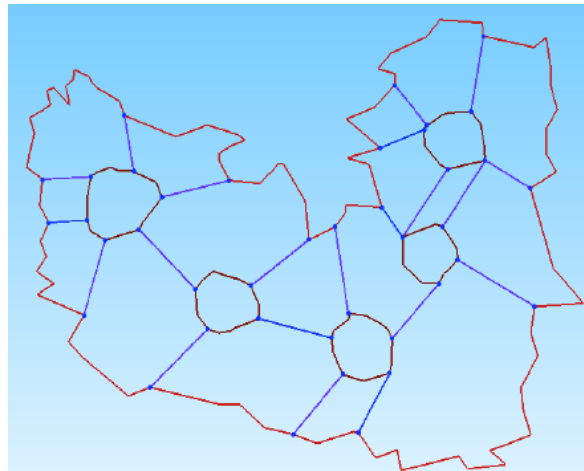
Observaciones 4.3:

☞ DUDE2D [71] es un algoritmo para una descomposición dual y aproximadamente convexa de regiones poligonales. Este algoritmo es automático e incluso puede separar regiones con agujeros Fig. 4.16(a). Sin embargo, las subregiones de la descomposición pueden tener bordes puntiagudos.

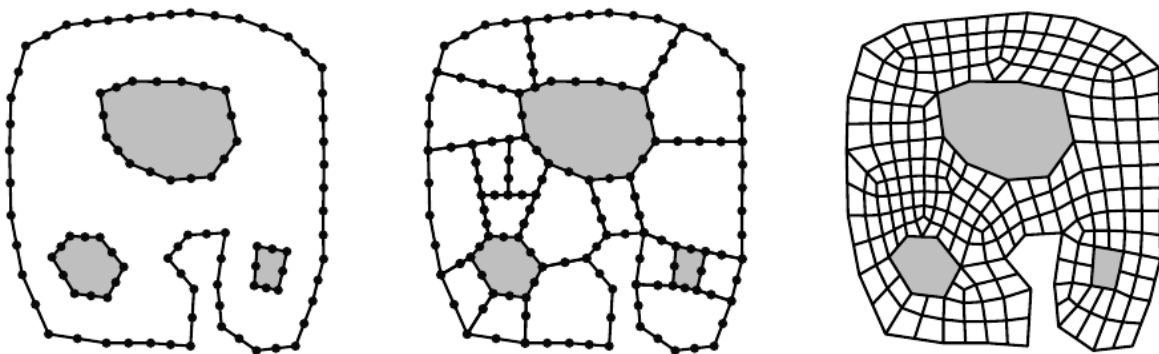
☞ Hongguang et al. [50] y Nowottny [92] emplean otros métodos para descomponer regiones por cortes, que usan medidas de concavidad y se pueden extender a regiones con agujeros, pero requieren algunas modificaciones para obtener *descomposiciones admisibles*. En particular, Nowottny ha generado mallas por bloques a partir de una descomposición por cortes, no obstante sus mallas pueden tener singularidades interiores, Fig. 4.16(c). Necesitamos realizar más pruebas y modificaciones pertinentes.



(a) Descomposición Dual DUDE2D [71].



(b) Descomposición de Hongguang et al. [50].



(c) Malla generada por la descomposición de Nowotny en [92].

Figura 4.16: Otros métodos para descomposición por cortes.

Ejemplo 4.2. La región `bird` no tiene agujeros, pero tiene cuellos de botella y varios bordes puntiagudos. Generamos de manera interactiva una *descomposición admisible* de 32 subregiones, de las cuales seis son triángulos. Empleamos la medida (4.3) para el criterio de concavidad y la elección de los cortes, Fig. 4.17. Usamos esta descomposición para generar una malla estructurada por bloques con una calidad del 97.77%, su mapa de calidad se muestra en la Fig. 4.18. Aunque algunas celdas alrededor de los bordes puntiagudos tienen baja calidad, la malla global es de alta calidad.

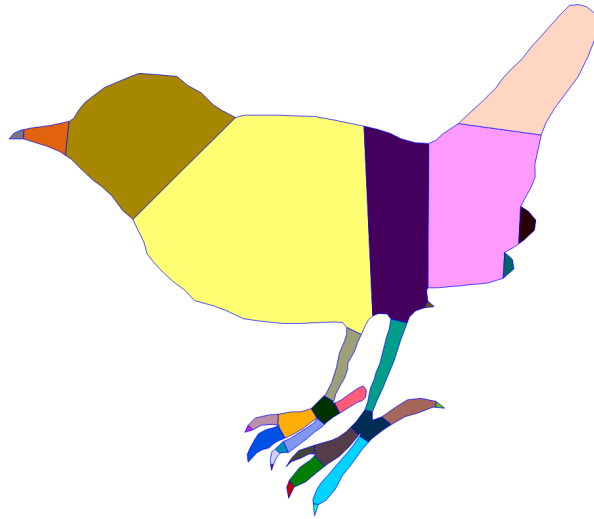


Figura 4.17: *Descomposición admisible* de un pajarito en 32 subregiones.

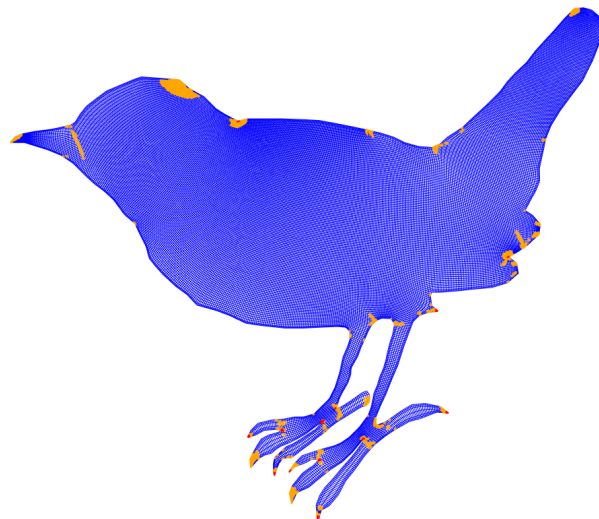
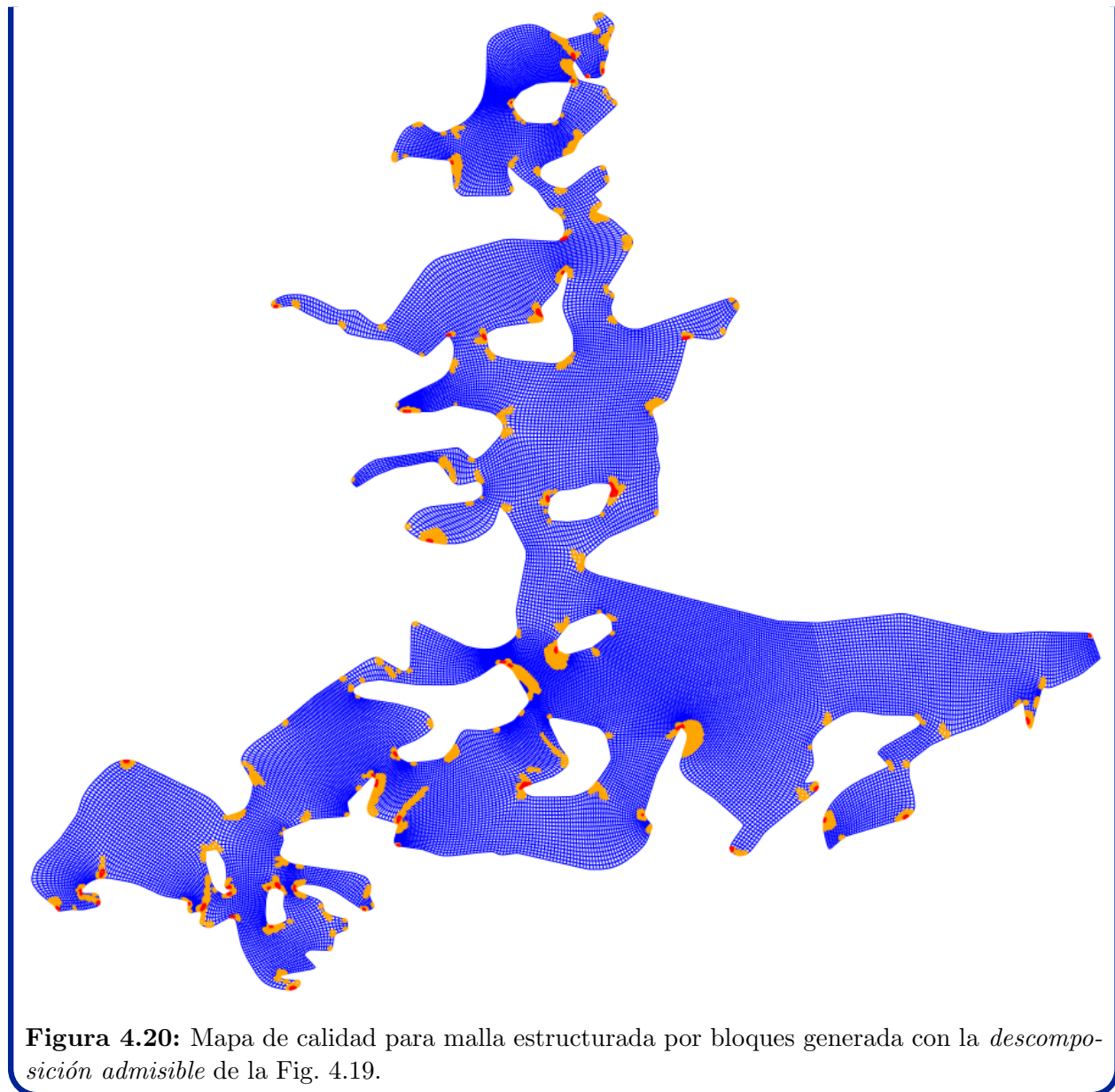


Figura 4.18: Mapa de calidad para malla estructurada por bloques generada con la *descomposición admisible* de Fig. 4.17.

Ejemplo 4.3. En el estado de Tamauliapas se ubica la Laguna Honda con ocho islas. Generamos de manera interactiva una *descomposición admisible* de 69 subregiones aproximadamente convexas, donde empleamos la medida (4.3) para el criterio de concavidad y la elección de los cortes, Fig. 4.19. Usamos esta descomposición para generar una malla estructurada por bloques con una calidad del 95.39%. Su mapa de calidad se muestra en la Fig. 4.20.



Figura 4.19: *Descomposición admisible* de la Laguna Honda en 69 subregiones.



4.2.2. Descomposición Simplemente Conexa

Las regiones con agujeros se separan en subregiones sin agujeros. La idea es conectar los agujeros con la frontera exterior mediante cortes. Esta descomposición por cortes es recursiva. En cada paso removemos un agujero mediante dos cortes. Cada corte conecta un punto del agujero con otro punto de la frontera, ya sea un punto sobre la frontera exterior o de otro agujero. La descomposición termina hasta que ninguna de las subregiones tenga agujeros. A continuación describimos un paso de esta descomposición por cortes.

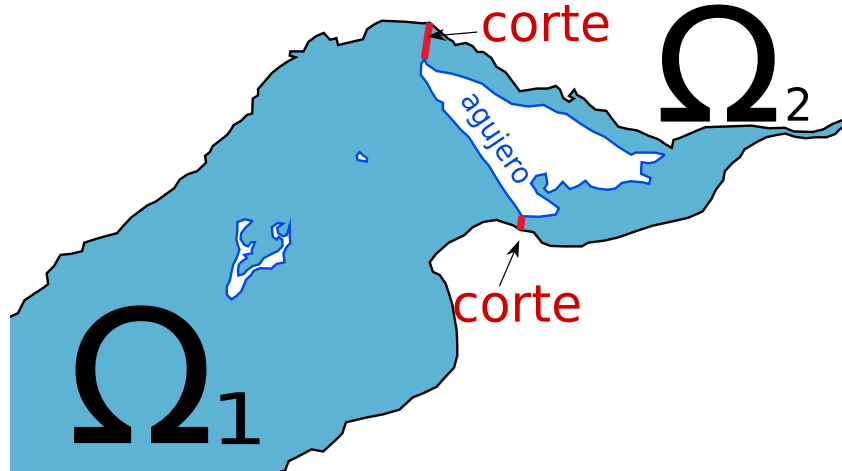


Figura 4.21: La frontera exterior se conecta con el agujero más cercano por dos cortes.

Algoritmo 2 Descomposición Simplemente Conexa

Rutina DESCOMPOSICIÓN I(Ω)

Elegir el agujero H más cercano a la frontera exterior E .

Hallar $\mathbf{p}_1 \in E$ y $\mathbf{q}_1 \in H$ que minimicen la distancia entre E y H .

Elegir $\mathbf{q}_2 \in \partial H$ con $\mathbf{q}_2 \neq \mathbf{q}_1$.

$\mathbf{p}_2 = \arg \min \{ \|\mathbf{p} - \mathbf{q}_2\| : \mathbf{p} \in \partial E \}$.

Separar Ω mediante los cortes $\overline{\mathbf{p}_1\mathbf{q}_1}$ y $\overline{\mathbf{p}_2\mathbf{q}_2}$ en Ω_1 y Ω_2 .

Sea Ω una región poligonal con agujeros. Denotamos por E a la frontera exterior de Ω . Entre todos los agujeros de Ω , removemos el agujero H con la menor distancia a E . Para ello realizamos dos cortes $\overline{\mathbf{p}_1\mathbf{q}_1}$ y $\overline{\mathbf{p}_2\mathbf{q}_2}$, donde $\mathbf{p}_1, \mathbf{p}_2 \in E$ y $\mathbf{q}_1, \mathbf{q}_2 \in \partial H$. Seleccionamos estos cortes como se muestra en el Algoritmo 2.

La región Ω se separa en dos subregiones Ω_1 y Ω_2 , que no contienen al agujero H como se muestra en las Figuras 4.13 y 4.21. Por construcción, la frontera del agujero H está contenida en las fronteras de las subregiones Ω_1 y Ω_2 .

4.2.3. Descomposición Aproximadamente Convexa

Separamos regiones simplemente conexas sin bordes puntiagudos usando medidas de concavidad [68, 71] con las modificaciones de Barrera et al. [12].

Sea Ω una región poligonal sin agujeros ni ángulos agudos críticos. Verificamos que Ω cumpla con un criterio de concavidad deseado. Por ejemplo, que el cambio relativo de su perímetro respecto al de su envolvente convexa sea pequeño. Si Ω no cumple el criterio de concavidad, la separamos por un corte en dos subregiones Ω_1 y Ω_2 , y repetimos este procedimiento en ambas subregiones. Continuamos de esta manera hasta que todas las subregiones cumplan el criterio de concavidad con una tolerancia fija. Ver Algoritmo 3.

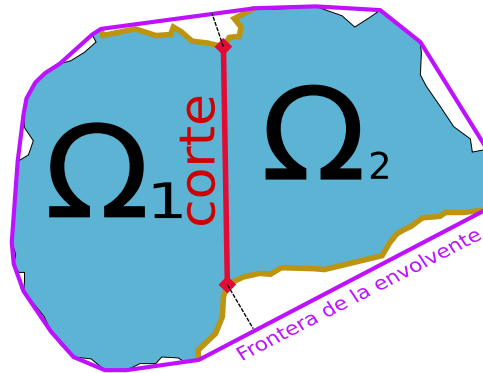


Figura 4.22: Elección del corte basada en la distancia a la envolvente convexa.

Algoritmo 3 Descomposición Aproximadamente Convexa

Rutina DESCOMPOSICIÓN II(Ω)

Elegir un vértice \mathbf{p} de alta concavidad.

Elegir otro punto \mathbf{q} sobre $\partial\Omega$.

Separar Ω por el corte $\overline{\mathbf{pq}}$ en Ω_1 y Ω_2 .

La elección de los cortes es un paso fundamental de nuestra *descomposición admisible*. Para escogerlos medimos la concavidad de los puntos en la frontera, de manera intuitiva, los puntos de baja concavidad están cerca de la frontera de la envolvente convexa, mientras que los puntos de alta concavidad se encuentran lejos, por lo que son candidatos a puntos de corte, Fig. 4.22.

Observaciones 4.4: En la elección del corte no es necesario escoger el punto de mayor concavidad, basta elegir un punto frontera de alta concavidad.

Las ramificaciones y cuellos de botella pueden removerse por cortes que conecten dos puntos de alta concavidad. No obstante, estos cortes pueden no ser adecuados en algunas regiones. Incluso los segmentos que unen puntos de alta concavidad pueden atravesar la frontera. En ese caso conectamos un punto de alta concavidad con otro de baja concavidad.

En vista de lo anterior, sugerimos realizar un corte $\overline{\mathbf{pq}}$ de Ω como sigue:

1. Seleccionar un punto frontera \mathbf{p} de alta concavidad.
2. De todos los puntos frontera que se pueden unir con \mathbf{p} por un corte, elegir el punto \mathbf{q} tal que las subregiones separadas por \mathbf{pq} tengan la menor medida de concavidad.

Aún con la elección de cortes sugerida, la descomposición aproximadamente convexa puede no ser admisible. Para evitar esto damos las siguientes recomendaciones:

- Las subregiones no deben tener ángulos críticos en los cortes.
- Los cortes deben ser ajenos.
- Evitar subregiones de área despreciable respecto a la región completa.

4.2.4. Descomposición para Ángulos Críticos

En algunas aplicaciones se requieren generar mallas de calidad aceptable sobre regiones con bordes puntiagudos. Aunque algunas celdas puedan tener baja calidad alrededor de esos bordes, es suficiente que un alto porcentaje tengan una alta calidad. La dificultad se presenta en los bordes con ángulos agudos críticos.

Deseamos generar mallas de calidad aceptable sobre regiones con estos ángulos sin introducir singularidades interiores. Para ello removemos los triángulos que forman los ángulos críticos. La región se descompone en subregiones sin estos ángulos y triángulos, Fig. 4.23.

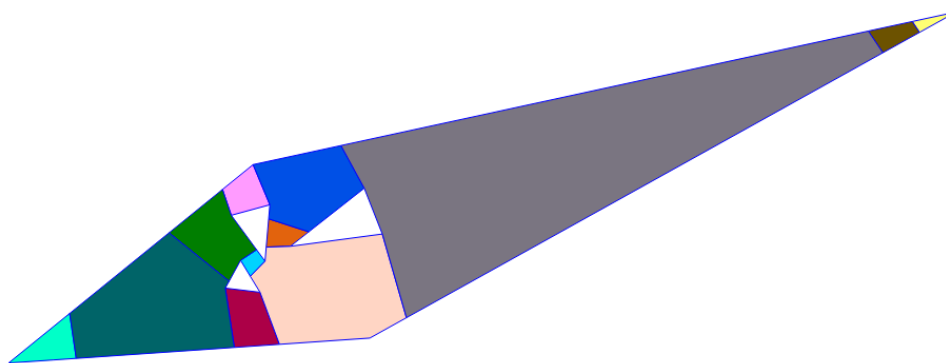


Figura 4.23: Descomposición por cortes de un rombo con tres agujeros y dos bordes puntiagudos en doce subregiones. Las dos subregiones con ángulos agudos críticos son triángulos.

Algoritmo 4 Descomposición para Ángulos Críticos

Rutina DECOMPOSICIÓN III(Ω)

Elegir puntos \mathbf{p} y \mathbf{q} en los dos lados que forman el ángulo más pequeño.

Separar Ω por el corte $\overline{\mathbf{pq}}$ en un triángulo Ω_1 y una subregión Ω_2 .

Sea Ω una región poligonal con vértices $\mathbf{v}_1, \dots, \mathbf{v}_n$. Identificamos un ángulo agudo crítico θ_i en el vértice \mathbf{v}_i . Sea \mathbf{p} es un punto del lado $\overline{\mathbf{v}_{i-1}\mathbf{v}_i}$, y sea \mathbf{q} un punto del lado $\overline{\mathbf{v}_i\mathbf{v}_{i+1}}$. La región Ω se separa por el corte $\overline{\mathbf{pq}}$ en el triángulo \mathbf{pqv}_i , denotado por Ω_1 , y en una subregión Ω_2 , Fig. 4.15. Repetimos este procedimiento en Ω_2 hasta extraer todos los triángulos con ángulos agudos críticos. En el Algoritmo 4 se muestra el paso principal de esta descomposición.

Observaciones 4.5:

- ☞ Bordes puntiagudos se pueden formar al realizar cortes en los vértices. Para evitar esto elegimos de manera interactiva otros puntos de corte sobre la frontera.
- ☞ Evitar remover triángulos con áreas pequeñas comparadas con la región completa para que las mallas no tengan tamaños enormes.

4.3. Gráficas de la Descomposición Admisible

En la descomposición de regiones se generan estructuras de datos que puede examinarse en el contexto de la teoría de gráficas. En nuestro caso generamos dos gráficas con nuestro método de descomposición.

Recordamos que una gráfica es una pareja de conjuntos, donde el segundo conjunto es una colección de parejas de elementos del primer conjunto. Los elementos del primer conjunto son los *nodos de la gráfica*, mientras que los elementos del segundo conjunto son las *aristas de la gráfica*. Los *nodos* y *aristas* de una gráfica usualmente se dibujan por puntos y segmentos que los unen.

Las gráficas que generamos son finitas y simples, esto es, gráficas con un número finito de nodos y aristas tales que todas las aristas unen dos nodos distintos y cada pareja de nodos se une a lo más con una arista.

Empleamos una estructura de datos conocida como *árbol*, esto es, una gráfica simple y conexa sin ciclos. En una *gráfica conexa* todos los nodos inciden con al menos una arista. Un *ciclo* en una gráfica es una sucesión consecutiva de nodos y aristas tal que el primer nodo es a la vez el último de la sucesión. Los nodos del árbol que inciden exactamente con una arista se llaman *hojas*.


Cuando a un solo nodo del árbol se le asocia una etiqueta, decimos que el árbol tiene raíz en ese nodo. En ese caso, los nodos que se conectan por una arista tienen una relación Padre-Hijo respecto a la raíz. Un árbol enraizado en el que cada nodo tiene a lo más dos hijos se llama *árbol binario*, Fig. 4.24.

Generamos un árbol binario T con nuestro método de descomposición, pues en cada paso separamos una región en dos subregiones de manera recursiva. Los nodos de T son las subregiones que generamos en cada paso. Las aristas de T son las parejas de subregiones que se obtienen una a partir de la otra en un solo paso del método, Fig. 4.24. La raíz de T es la región completa, y sus hojas son las subregiones de la *descomposición admisible*.

Además del árbol, asociamos otra gráfica \mathcal{G} a la *descomposición admisible*. Sus nodos son las subregiones de esta descomposición y sus aristas son las parejas de subregiones que comparten un corte. Esta gráfica es simple porque si dos subregiones de una *descomposición admisible* se intersectan, entonces su intersección es un corte, Fig. 4.25.

Observaciones 4.6:

 Los nodos de \mathcal{G} son las hojas del árbol binario T , y se conectan por una arista si tienen el mismo padre en el árbol T .

 Si la región es simplemente conexa, la gráfica \mathcal{G} es un árbol. En cambio, si la región tiene agujeros, la gráfica \mathcal{G} tiene un ciclo por cada agujero.

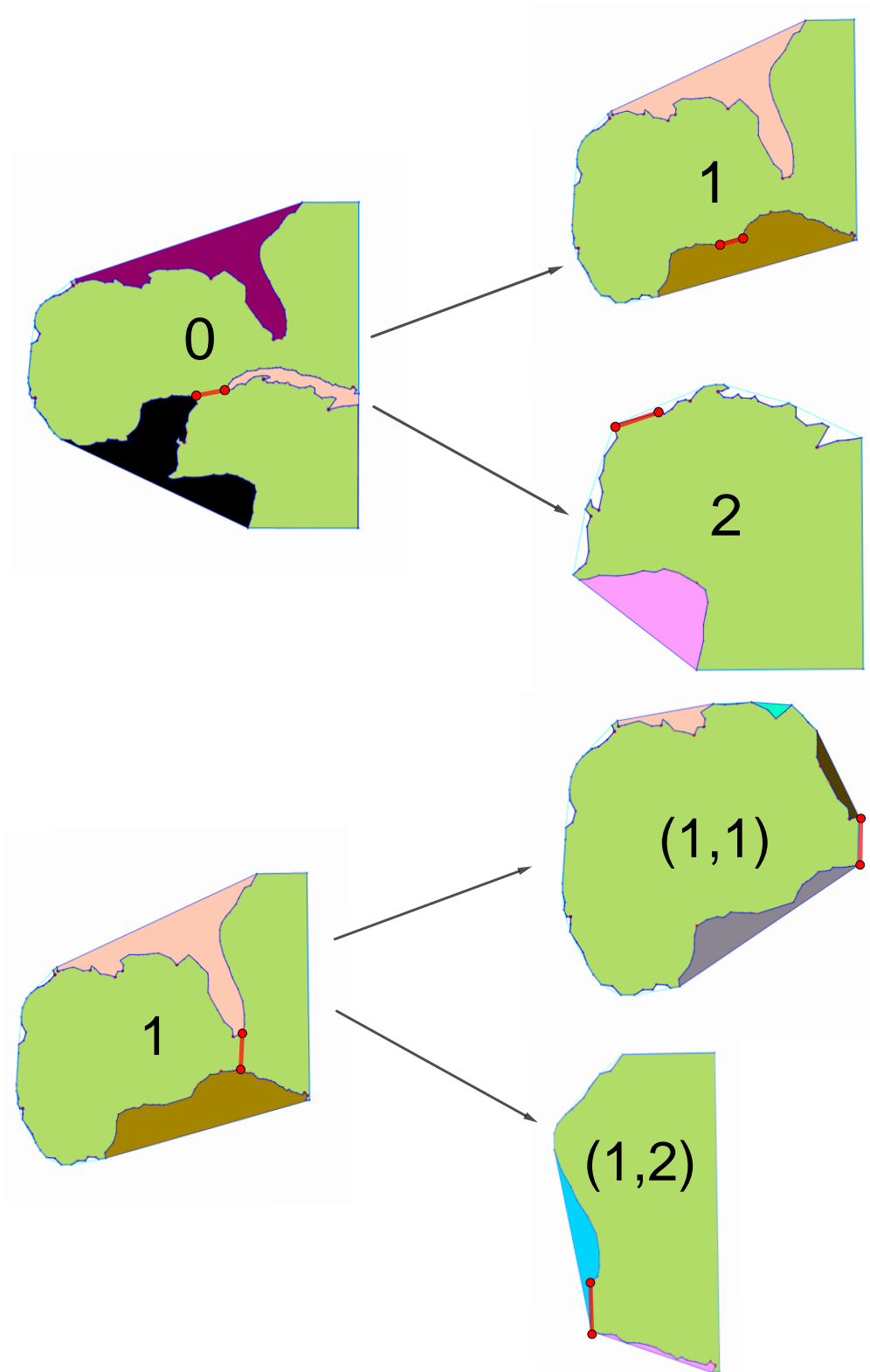
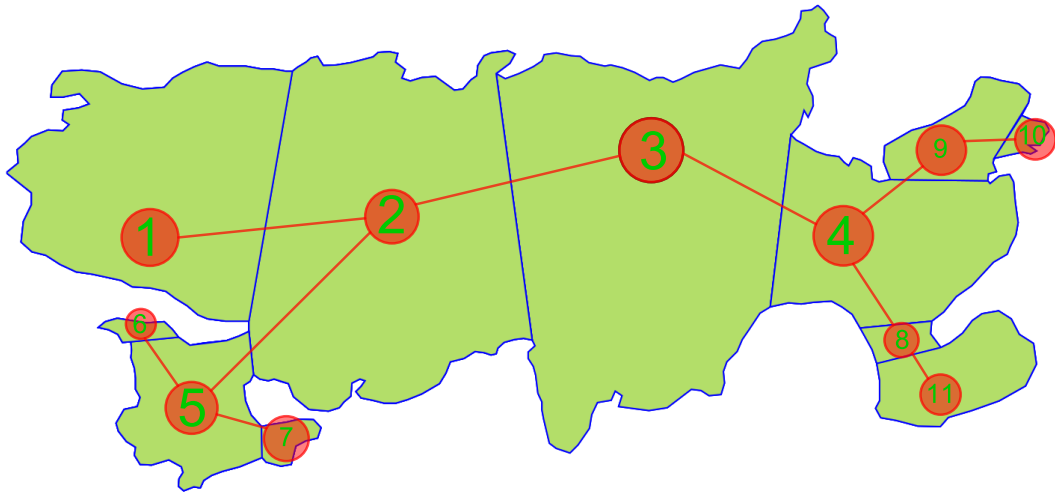
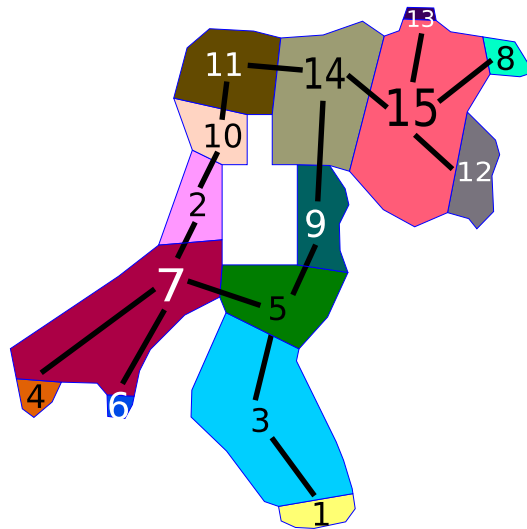


Figura 4.24: Descomposición recursiva por cortes del Golfo de México en verde. Las subregiones generadas se etiquetan con tuplas de 1's y 2's para indicar la estructura de árbol binario.



(a) La gráfica para una *descomposición admisible* del Lago Titicaca es un *árbol*.



(b) La gráfica para una *descomposición admisible* de la Bahía de la Habana con un agujero tiene *ciclos*.

Figura 4.25: Gráfica \mathcal{G} asociada a una *descomposición admisible* de una región poligonal.

4.4. Medidas y Criterios de Concavidad para Descomposición Admisible

La concavidad es clave para identificar *regiones admisibles* y descomponer regiones complicadas en *subregiones admisibles*. En esta sección indicamos las medidas y *criterios de concavidad* que emplea nuestro método de descomposición. Para este fin usamos los conceptos introducidos por Brinkhoff et al. [23], Lien J. [68] y Liu et al. [71]. Medimos la concavidad de la región en relación a su *envolvente convexa*.



Figura 4.26: Complemento convexo de una región del Golfo de México. Mostramos un *punte* y su *bolsillo* correspondiente.

Sea Ω una región poligonal no convexa sin agujeros. Para medir su concavidad, comparamos su área y perímetro con los de su *envolvente convexa* \mathcal{C} como sigue:

$$\kappa_1(\Omega) = \frac{\text{perímetro}(\Omega) - \text{perímetro}(\mathcal{C})}{\text{perímetro}(\Omega)}, \quad (4.2)$$

$$\kappa_2(\Omega) = \frac{\text{área}(\mathcal{C}) - \text{área}(\Omega)}{\text{área}(\mathcal{C})}. \quad (4.3)$$

Ambas medidas son independientes de la escala de la región:

$$0 \leq \kappa_i(\Omega) < 1, \quad i = 1, 2.$$

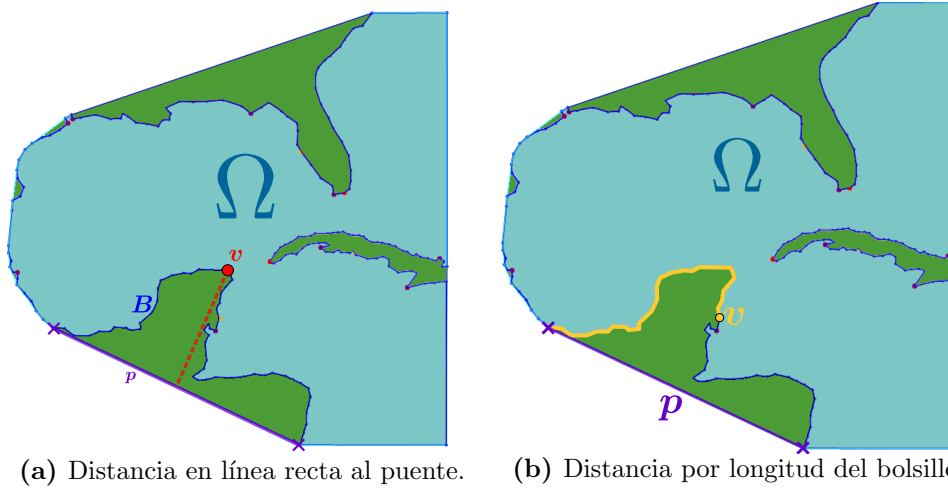
Esto nos permite identificar regiones aprox. convexas con los siguientes criterios:

- 1° criterio de concavidad. El perímetro de la envolvente \mathcal{C} es pequeño respecto al de la región Ω , esto es,

$$\kappa_1(\Omega) \leq \tau_1^-, \quad \tau_1^- \in (0, 1). \quad (4.4)$$

- 2° criterio de concavidad. El área de la región Ω es pequeña respecto al de su *envolvente* \mathcal{C} , es decir,

$$\kappa_2(\Omega) \leq \tau_2^-, \quad \tau_2^- \in (0, 1). \quad (4.5)$$



(a) Distancia en línea recta al puente. (b) Distancia por longitud del bolsillo.

Figura 4.27: La concavidad de un punto v en bolsillo B es su distancia al respectivo puente p .

Otros medidas y criterios de concavidad se pueden obtener al remover la región de la envolvente convexa. El *complemento convexo* de Ω es la diferencia de conjuntos $\mathcal{C} \setminus \Omega$. La concavidad se puede medir con las curvas poligonales sobre la frontera y los segmentos en el complemento convexo.

Los *puentes* de Ω son segmentos de recta en $\mathcal{C} \setminus \Omega$ que unen dos puntos sobre $\partial\Omega$. Cada puente de Ω tiene asociada una curva poligonal sobre $\partial\Omega \setminus \partial\mathcal{C}$ que conecta los extremos del puente. Esta curva se llama *bolsillo*. Los puentes que usamos son las componentes de $\partial\mathcal{C} \setminus \partial\Omega$. Sus respectivos bolsillos son las componentes de $\partial\Omega \setminus \partial\mathcal{C}$, Fig. 4.26.

Observaciones 4.7:

- ☞ La unión de un puente con su bolsillo es una curva cerrada. La región encerrada por esta curva se encuentra en el complemento convexo.
- ☞ El complemento convexo puede ser un conjunto disconexo. Más aún, las fronteras de sus componentes son uniones de puentes con sus bolsillos correspondientes.

Medimos la concavidad de los puntos frontera usando puentes y bolsillos. Sea p un punto sobre la frontera de Ω . La concavidad de p es su distancia a la frontera de la envolvente convexa \mathcal{C} . Si $p \in \partial\mathcal{C}$, la concavidad de p es cero. De otro modo, p se encuentra en algún bolsillo ρ con puente β . Como los puentes que usamos se encuentran sobre la frontera de \mathcal{C} , la concavidad de p se puede medir como su distancia al puente β . Esta distancia puede medirse en línea recta o por longitud de arco, Fig. 4.27. Así que tenemos dos medidas de concavidad:

- Denotamos por $\kappa_3(p)$ a la distancia ortogonal del punto p al segmento β .
- Denotamos por $\kappa_4(p)$ a la longitud de la curva más pequeña sobre el bolsillo ρ que une p con el puente β .

Necesitamos que las medidas κ_3 y κ_4 sean independientes de la escala de la región para distinguir los puntos de alta concavidad de los puntos que tienen baja concavidad, para ello reescalamos la región Ω como sigue:

- Ω se reescala dentro de un círculo de radio uno para medir κ_3 .
- Ω se reescala de modo que su perímetro sea uno para medir κ_4 .

Sea \mathbf{q} un punto sobre un bolsillo de Ω . Decimos que \mathbf{q} es un *punto de baja concavidad* respecto a la medida κ_i si $\kappa_i(\mathbf{p}) \leq \tau_i^-$. En cambio, si $\kappa_i(\mathbf{p}) \geq \tau_i^+$, decimos que \mathbf{p} es un *punto de alta concavidad* respecto a la medida κ_i , donde $0 < \tau_i^- < \tau_i^+ < 1$ para $i = 3, 4$.

La concavidad de los puntos en los bolsillos se usa para medir la concavidad de los puentes correspondientes. Sea β un puente de Ω con bolsillo ρ . La concavidad de β es la máxima concavidad de los puntos sobre ρ . Por lo que medimos la concavidad de los puentes como:

$$\kappa_i(\beta) = \max\{\kappa_i(\mathbf{q}) : \mathbf{q} \in \rho\}, \quad i = 3, 4.$$

Esto a su vez nos permite definir medidas de concavidad para las regiones. La concavidad de la región Ω puede medirse como la máxima concavidad de sus puentes:

$$\kappa_i(\Omega) = \max\{\kappa_i(\beta) : \beta \text{ es puente de } \Omega\}, \quad i = 3, 4.$$

De este modo podemos emplear las medidas anteriores para identificar regiones aproximadamente convexas. Para ello usamos los siguientes criterios de concavidad:

- 3° criterio de concavidad: La distancia en línea recta a los puentes es pequeña.

$$\kappa_3(\Omega) \leq \tau_3^-, \quad \tau_3^- \in (0, 1). \quad (4.6)$$

- 4° criterio de concavidad: La distancia por longitud de arco a los puentes es pequeña.

$$\kappa_4(\Omega) \leq \tau_4^-, \quad \tau_4^- \in (0, 1). \quad (4.7)$$

Los criterios de concavidad que usamos dependen tanto de la medida de concavidad elegida y de la tolerancia empleada. Las regiones que cumplen alguno de estos criterios son aprox. convexas. De manera más precisa, decimos que Ω es una *región aproximadamente convexa* respecto a la medida κ_i si

$$\kappa_i(\Omega) \leq \tau_i^-.$$

En cambio, Ω es una región es una *región altamente cóncava* respecto a la medida κ_i si

$$\kappa_i(\Omega) \geq \tau_i^+.$$

Observaciones 4.8: Las tolerancias τ_i^- y τ_i^+ son independientes de la elección de los puntos y las regiones. Sugerimos los siguientes valores experimentales: $\tau_1^+ = 0.5$, $\tau_1^- = 0.1$, $\tau_2^- = 0.45$, $\tau_3^- = 0.1$.

4.5. Resumen del Capítulo

Para generar una malla de alta calidad sin singularidades interiores sobre una región poligonal complicada Ω , la descomponemos en una colección finita $\{\Omega_k\}$ de subregiones que no tienen agujeros ni ángulos agudos críticos, y que satisfacen criterios de concavidad, estas subregiones son *admisibles* para generar mallas estructuradas de alta calidad, si además, sus intersecciones no vacías son cortes de Ω y cada agujero se conecta con al menos cuatro cortes, decimos que la colección $\{\Omega_k\}$ es una *descomposición admisible* de Ω .

Revisamos algunos métodos de la literatura para descomposición de regiones, de los cuales empleamos algunas ideas como realizar cortes y medir la concavidad para desarrollar nuestro método de descomposición. De manera recursiva generamos una *descomposición admisible* por cortes de la región Ω en tres etapas:

1. **Descomposición en regiones simplemente conexas.** Conectamos el agujero más cercano a la frontera exterior mediante dos cortes para separar la región en dos subregiones sin ese agujero, repetimos el proceso hasta obtener subregiones sin agujeros.
2. **Descomposición en regiones aproximadamente convexas.** Descomponemos regiones que no satisfacen criterios de concavidad en dos subregiones mediante un corte, que eligimos con base a medidas de concavidad, repetimos el proceso hasta obtener subregiones que cumplan los criterios de concavidad.
3. **Descomposición para ángulos críticos.** Separamos una región con un *ángulo agudo crítico* en un triángulo y una subregión sin ese ángulo mediante un corte, repetimos el proceso hasta que todas las subregiones con estos ángulos sean triángulos.

Para identificar *subregiones admisibles* y elegir cortes medimos la concavidad de la región Ω con las medidas κ_i introducidas en §4.4, que comparan la región con su envolvente convexa \mathcal{C} , si alguna de estas medidas está por debajo de un umbral dado, decimos que la región Ω cumple con un criterio de concavidad; de otro modo, recomendamos elegir puntos sobre $\partial\Omega$ alejados de $\partial\mathcal{C}$ como puntos de corte y separar la región Ω .

En nuestro proceso de descomposición generamos un *árbol binario*, sus nodos son las subregiones de cada paso, que conectamos por una arista si una subregión se obtiene de otra en un solo paso. Por otra parte, asociamos otra gráfica a la *descomposición admisible*, sus nodos son las *subregiones admisibles*, que conectamos por una arista si comparten un corte de la región.

Una vez que separamos la región en *subregiones admisibles*, procedemos a construir las parametrizaciones de cada subregión, la construcción se realiza primero en las fronteras, y luego se extiende al interior. En el siguiente capítulo construimos mapeos compatibles del los lados del cuadrado unitario las fronteras de las subregiones.

Construimos parametrizaciones de regiones poligonales, lo que hacemos es hallar una colección de *mapeos compatibles* del cuadrado unitario a las subregiones de una *descomposición admisible*. En el capítulo 4 vimos cómo separar regiones. Ahora, construimos sus parametrizaciones de modo que sean compatibles en sus intersecciones. Dado que las subregiones de una *descomposición admisible* no se traslapan, podemos construir sus *parametrizaciones compatibles* en dos etapas:

1. Parametrizar las fronteras de las subregiones por curvas compatibles.
2. Extender las parametrizaciones de las fronteras.

En este capítulo parametrizamos las fronteras de las subregiones, mientras que en el siguiente capítulo las extendemos al interior de la región. Nuestras parametrizaciones de las fronteras son mapeos de los lados del cuadrado unitario a los lados de las subregiones que son biyecciones entre los lados opuestos y compatibles en las intersecciones, que en nuestro caso son cortes de la región. Así que formulamos la construcción de sus *parametrizaciones compatibles* como sigue:

Problema 5.1. (Parametrizaciones compatibles) Dada una descomposición admisible de una región, encontrar mapeos de los lados del cuadrado unitario a los lados de las fronteras de las subregiones que sean biyecciones entre los puntos de lados opuestos y compatibles en los cortes.

En regiones sencillas los mapeos de los lados del cuadrado unitario a la frontera se pueden extender como parametrizaciones globales. Para construir estos mapeos separamos la frontera en cuatro curvas que corresponden a los lados del cuadrado. Reparametrizamos estas curvas de modo que los puntos de lados opuestos estén en correspondencia biunívoca. Generalizamos esta idea para construir *parametrizaciones compatibles*.

Una manera para construir *parametrizaciones compatibles* de las fronteras es separar las fronteras de las subregiones en curvas que reparametrizamos con un número compatible de puntos. Abordamos el Problema 5.1 como tres subproblemas:

1. **Descomposición de la frontera.** Separar las fronteras de las subregiones de modo que en cada una tenga cuatro *lados* formados por cortes o curvas de la frontera de la región completa.
2. **Compatibilidad de la frontera.** Hallar el número de puntos que requieren las curvas de la frontera para que coincidan en los cortes y que los lados opuestos de cada subregión tengan una correspondencia biunívoca entre sus puntos.

3. **Reparametrización de la frontera.** Reparametrizar cada curva de la descomposición de la frontera con el número compatible de puntos.

En las siguientes secciones describimos cómo resolver estos problemas.

5.1. Descomposición de la Frontera

Sea Ω una región poligonal, y sea $\{\Omega_k\}$ una *descomposición admisible* de Ω en n subregiones. Las fronteras de las subregiones se descomponen en una colección de curvas poligonales que las separan en cuatro lados y contienen a los cortes de Ω . A continuación construimos esta descomposición de las fronteras.


En cada bloque Ω_k seleccionamos cuatro puntos sobre la frontera, a saber, sus *esquinas*, de ese modo la frontera de Ω_k se separa en cuatro curvas consecutivas que denotamos por $\Omega_{k,b}$, $\Omega_{k,r}$, $\Omega_{k,t}$ y $\Omega_{k,l}$. Las parejas $(\Omega_{k,b}, \Omega_{k,t})$ y $(\Omega_{k,r}, \Omega_{k,l})$ son lados opuestos de la frontera de Ω_k . Construimos mapeos del cuadrado unitario a la frontera de la subregión Ω_k que transforman las *esquinas* del cuadrado en esquinas de Ω_k y lados del cuadrado en lados de la subregión Ω_k , así que los lados opuestos de a subregión corresponden a lados opuestos del cuadrado.


Los cuatro lados de cada subregión Ω_k se descomponen a su vez en curvas poligonales, denotadas por Q_j , que son:

- cortes de Ω .
- curvas contenidas en la frontera exterior de Ω o en sus fronteras interiores.

Cada una de las curvas Q_j conecta un par de puntos en la frontera de Ω que son *esquinas* de las subregiones Ω_k o puntos de corte de la región Ω . De esta manera obtenemos una descomposición $\{Q_j\}$ para las fronteras de las subregiones en cortes de Ω y curvas contenidas en $\partial\Omega$, Fig. 5.1.

Observaciones 5.1:

 Si la región Ω tiene agujeros, las curvas Q_j sobre la frontera de Ω están contenidas en la frontera exterior de Ω o bien en la frontera de alguno de los agujeros de Ω , Fig. 5.2.

 Por cada corte de Ω tenemos dos curvas de la descomposición $\{Q_j\}$. Aunque ambas curvas representen el mismo corte, consideramos que pertenecen a dos subregiones distintas.

La descomposición de las fronteras se reduce a la elección de *esquinas*. Una vez que separamos la frontera de cada subregión en cuatro lados, identificamos los cortes de la región para obtener las curvas de la descomposición. La elección de las *esquinas* es el paso clave de la descomposición de las fronteras.

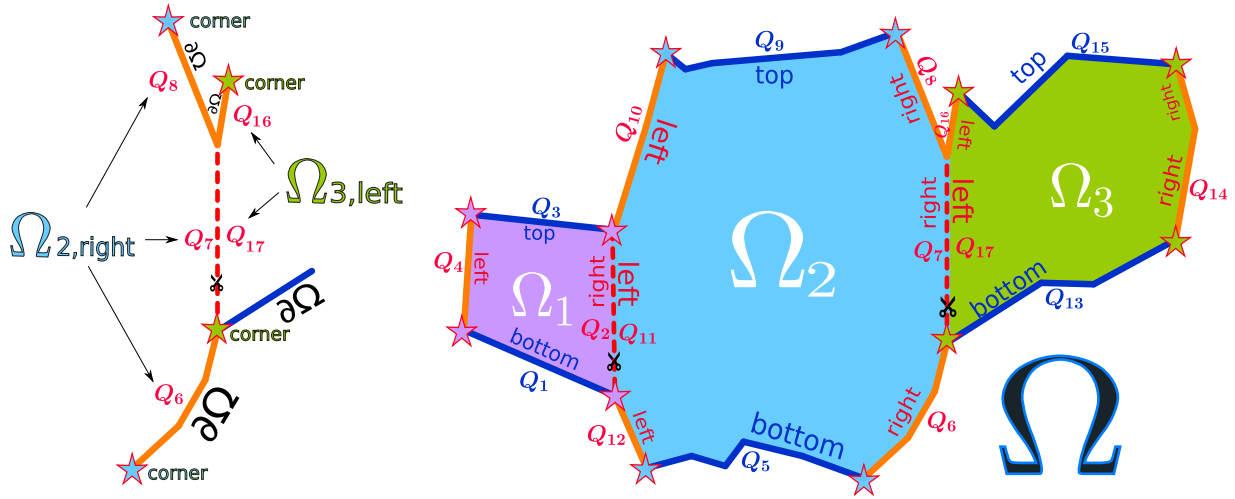


Figura 5.1: La región Ω se descompone en tres bloques Ω_1 , Ω_2 y Ω_3 . Separamos la unión de sus fronteras de estos en 17 curvas Q_p . Las curvas Q_2 , Q_7 , Q_{11} y Q_{17} son cortes de Ω mientras que las otras curvas se encuentran en $\partial\Omega$. El lado $\Omega_{2,\text{right}}$ se separa en tres curvas, mientras que el lado $\Omega_{3,\text{left}}$ se separa en dos curvas.

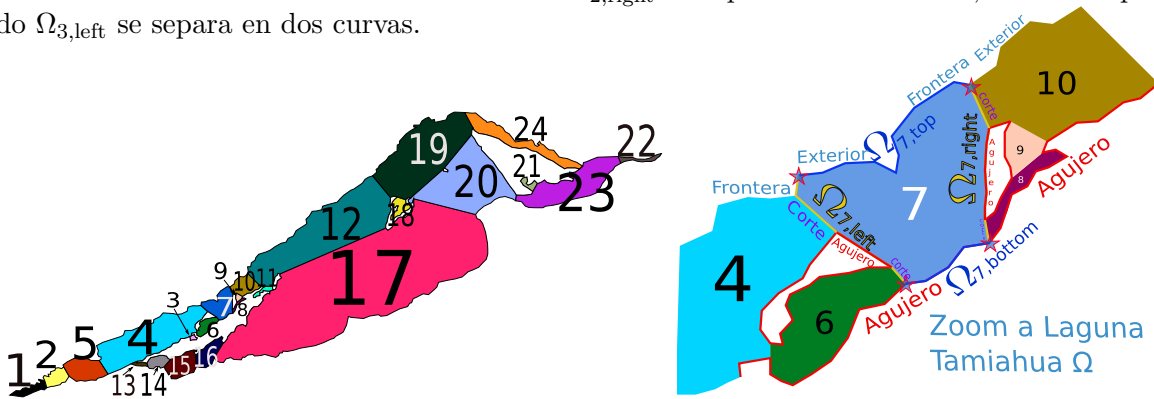


Figura 5.2: *Descomposición admisible* de la Laguna de Tamiahua Ω en 24 bloques. La frontera del bloque Ω_7 se separa en cuatro lados que conectan las estrellas. A su vez cada lado se separa en cortes, curvas contenidas en la frontera exterior o en las fronteras de agujeros.

5.1.1. Elección de Esquinas

Las *esquinas* de una región son puntos de la frontera con alta curvatura, que pueden pensarse como las esquinas de un cuadrilátero que preserva la forma de la región. La calidad de nuestras mallas, y por consiguiente de nuestras parametrizaciones depende de la elección de adecuada de las *esquinas*.

En cada subregión seleccionamos cuatro *esquinas* de modo que los mapeos entre los lados del cuadrado unitario y los lados de la subregión se puedan extender como mallas estructuradas de alta calidad. Como las subregiones se obtienen de una *descomposición admisible*, basta elegir *esquinas* de *regiones admisibles*.

Una condición necesaria para que los vértices de una región poligonal sean esquinas es que sus ángulos interiores sean menores que 180° . Las esquinas que son puntos colineales de la frontera generan celdas distorsionadas y las esquinas que son *vértices reflex* generan celdas no convexas.

Las cuatro esquinas separan la frontera de la región en dos parejas de lados opuestos. Si un lado de la frontera es largo en comparación con el lado opuesto, las mallas que generamos tienen celdas elongadas alrededor del lado más corto, más aún, la calidad de la malla puede verse comprometida.

Con base a lo anterior, empleamos la siguiente heurística para seleccionar esquinas:

Criterio para elección de *esquinas* en *regiones admisibles*. Seleccionar cuatro vértices que no sean puntos colineales ni vértices reflex de modo que los lados opuestos tengan aproximadamente la misma longitud.


Para implementar el criterio anterior buscamos vértices candidatos y seleccionamos cuatro con base a la proporción de las longitudes de los lados opuestos. Sugerimos elegir dos puntos diametralmente opuestos de la *envolvente convexa* y otros dos puntos sobre la frontera de la envolvente tales que los lados opuestos tengan la proporción deseada.


Elección de esquinas para regiones con ángulos críticos. En algunos casos se requiere construir parametrizaciones de regiones con ángulos agudos críticos. El problema es generar mallas de alta calidad sin singularidades interiores sobre estas regiones.

Recordamos que separamos regiones con ángulos críticos en *subregiones admisibles* y triángulos. Como ya tenemos una heurística para seleccionar las esquinas de *regiones admisibles*, basta examinar la elección de esquinas para triángulos. La clave es agregar puntos colineales cercanos a los vértices con ángulos críticos. Distinguimos dos casos:

- Si el triángulo tiene un solo vértice con ángulo crítico, agregamos dos puntos sobre las aristas que inciden con ese vértice de modo que ambos equidisten del vértice. Los nuevos puntos y los otros dos vértices son las esquinas que elegimos, Fig. 5.3.
- Si el triángulo tiene dos vértices con *ángulos agudos críticos*, agregamos cuatro puntos de modo que ambos vértices tengan dos puntos cercanos y equidistantes sobre las respectivas aristas incidentes. Elegimos los nuevos puntos como esquinas, Fig. 5.4.

Observaciones 5.2:

 Nuestro generador de mallas UNAMalla 6 implementa la heurística anterior para seleccionar automáticamente las *esquinas*. Si la elección de esquinas no es adecuada, se puede realizar una selección interactiva.

 La calidad de muestras mallas se ve comprometida en regiones con *ángulos agudos críticos*. Las mallas estructuradas sobre triángulos tienen celdas distorsionadas alrededor de los bordes puntiagudos y las *esquinas* que son puntos colineales.

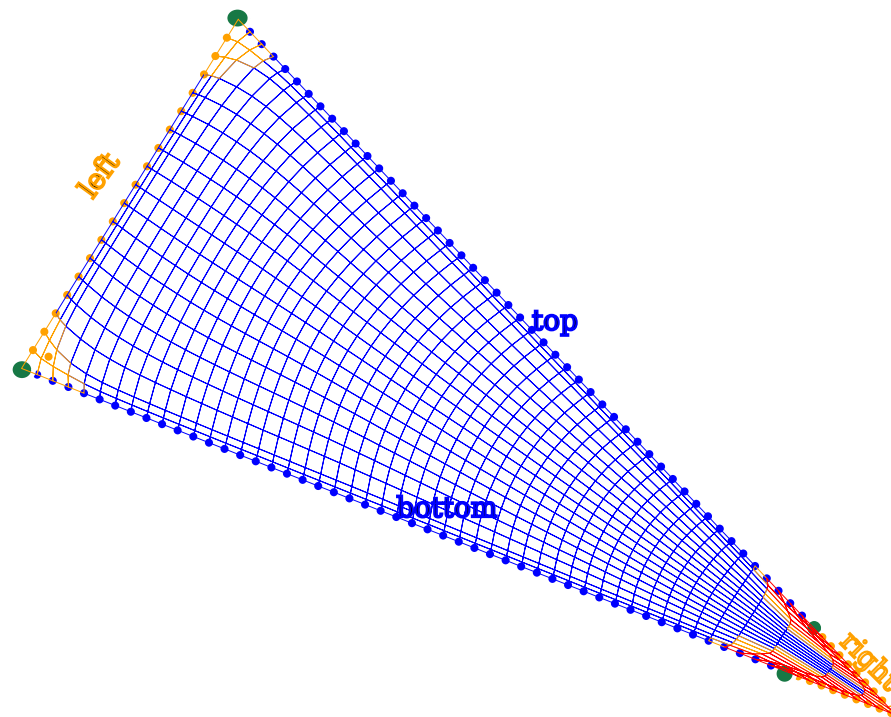


Figura 5.3: Mapa de calidad para malla estructurada 50×20 de un triángulo isósceles con un ángulo de 24° y otro de 78° . Agregamos dos puntos colineales en verde. Las esquinas son los vértices con mismo ángulo junto con los nuevos puntos.

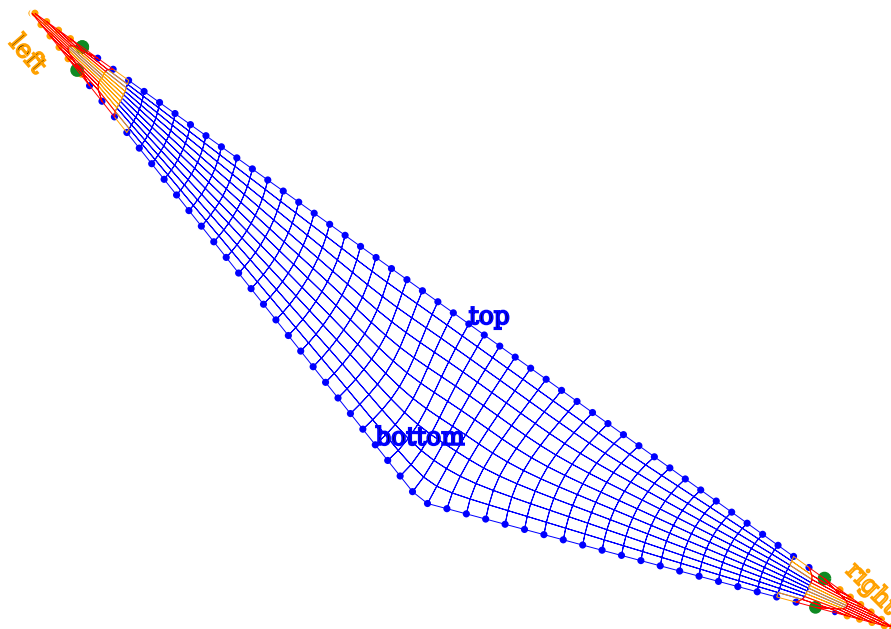


Figura 5.4: Mapa de calidad para malla estructurada de un triángulo escaleno con un ángulo de 15° y otro de 20° . Agregamos cuatro puntos colineales en color verde que elegimos como las *esquinas*.

5.2. Generación de Tamaños Compatibles

La construcción de parametrizaciones compatibles se formula como la generación de mallas estructuradas con tamaños compatibles. La región se descompone en subregiones y su vez las fronteras de las subregiones se separan en curvas para generar estas mallas.

Las curvas de la descomposición de la frontera se reparametrizan con un número compatible de puntos para obtener los puntos frontera de las mallas. De ese modo el problema de construir parametrizaciones compatibles se reduce a determinar el número compatible de puntos que requieren las curvas de la frontera. Formulamos las condiciones de compatibilidad que deben satisfacer los tamaños de las mallas y derivamos las ecuaciones correspondientes.

5.2.1. Ecuaciones de Compatibilidad

Sea $\{\Omega_k\}$ una *descomposición admisible* de la región Ω en n bloques Ω_k que requiere q cortes. Denotamos por $\Omega_{k,b}$, $\Omega_{k,r}$, $\Omega_{k,t}$ y $\Omega_{k,l}$ a los lados (abajo, derecha, arriba, izquierda) del bloque Ω_k . Sea D la descomposición de $\cup_k \partial\Omega_k$ en p curvas poligonales Q_j que son cortes de Ω o curvas contenidas en $\partial\Omega$ y en alguno de los lados. Extendemos las curvas de D como mallas compatibles. Para ello estas curvas deben cumplir las siguientes condiciones de compatibilidad:

1. Las curvas de D que son cortes tienen los mismos puntos.
2. En cada bloque hay biyecciones entre los puntos de lados opuestos.

Sea m_j el número de puntos de la curva Q_j . Expresamos las condiciones de compatibilidad como ecuaciones en las variables m_j .

1. Para cada corte de Ω hay exactamente dos curvas en la descomposición de la frontera que son iguales al corte. Así que el corte i es igual a dos curvas de D con índices ι_i y ς_i que tienen m_{ι_i} y m_{ς_i} vértices, respectivamente. Como ambas curvas deben tener los mismos puntos, tenemos las siguientes ecuaciones:

$$m_{\iota_i} - m_{\varsigma_i} = 0, \quad \iota_i, \varsigma_i \in \{1, \dots, p\}, \quad i = 1, \dots, q. \quad (5.1)$$

2. Los lados opuestos deben tener el mismo número de puntos para que haya biyecciones entre sus puntos. Así que la suma del número m_j de puntos de las curvas Q_j de D en el lado $\Omega_{k,b}$ debe ser igual a la suma del número m_j de puntos de las curvas Q_j de D en el lado opuesto $\Omega_{k,t}$, sin contar los puntos repetidos donde las curvas se unen.

Para contar los puntos repetidos usamos la diferencia $s_{k,b} - s_{k,t}$, donde $s_{k,b}$ y $s_{k,t}$ son el número de curvas en D contenidas en $\Omega_{k,b}$ y $\Omega_{k,t}$, respectivamente. Entonces la condición de compatibilidad para los lados $\Omega_{k,b}$ y $\Omega_{k,t}$ se expresa como las siguientes ecuaciones:

$$\sum \{m_j : Q_j \subset \Omega_{k,b}\} - \sum \{m_j : Q_j \subset \Omega_{k,t}\} = s_{k,b} - s_{k,t}, \quad k = 1, \dots, n, \quad (5.2)$$

Las curvas Q_j de los lados opuestos $\Omega_{k,r}$ y $\Omega_{k,l}$ satisfacen la misma condición de compatibilidad. Así que tenemos las siguientes ecuaciones:

$$\sum\{m_j : Q_j \subset \Omega_{k,r}\} - \sum\{m_j : Q_j \subset \Omega_{k,l}\} = s_{k,r} - s_{k,l}, \quad k = 1, \dots, n, \quad (5.3)$$

donde $s_{k,r}$ y $s_{k,l}$ son el número de curvas de D contenidas en $\Omega_{k,r}$ y $\Omega_{k,l}$.

Las ecuaciones (5.1)-(5.3) son lineales en las variables m_j . Para expresar estas ecuaciones en forma matricial recorreremos las curvas de la descomposición D bloque por bloque y lado por lado. Recorreremos los bloques en el orden $\Omega_1, \dots, \Omega_n$, y sus lados en el orden $\Omega_{k,b}, \Omega_{k,r}, \Omega_{k,t}, \Omega_{k,l}$, que corresponde a recorrer los lados del cuadrado en el siguiente orden: abajo, derecha, arriba, izquierda. Introducimos conjuntos de índices para identificar la posición de las curvas de D en este orden.

El número de curvas de D en la frontera del bloque Ω_k es

$$s_k = s_{k,b} + s_{k,r} + s_{k,t} + s_{k,l}.$$

La suma acumulativa del número de curvas de D hasta el bloque Ω_k es:

$$\sigma_k = \begin{cases} 0, & \text{if } k = 1; \\ \sum_{\ell=1}^{k-1} s_\ell, & \text{if } k > 1; \end{cases} \quad k = 1, \dots, n$$

Ubicamos las curvas de D en el orden deseado con los siguientes conjuntos de índices:

$$\begin{aligned} J_{k,b} &= \sigma_k + \{1, \dots, s_{k,b}\}, \\ J_{k,r} &= \sigma_k + s_{k,b} + \{1, \dots, s_{k,r}\}, \\ J_{k,t} &= \sigma_k + s_{k,b} + s_{k,r} + \{1, \dots, s_{k,t}\}, \\ J_{k,l} &= \sigma_k + s_{k,b} + s_{k,r} + s_{k,t} + \{1, \dots, s_{k,l}\}, \end{aligned}$$

donde la suma de un número con un conjunto de índices es sumar el número a cada índice. De esta manera:

$$Q_j \subset \Omega_{k,\text{side}} \iff j \in J_{k,\text{side}}, \quad k = 1, \dots, n, \quad \text{side} = b, t, r, l.$$

Esto nos permite expresar las ecuaciones (5.1)-(5.3) como el siguiente sistema de ecuaciones lineales con coeficientes enteros:

$$A\mathbf{m} = \mathbf{b}, \quad (5.4)$$

donde

- A es una matriz entera de tamaño $(2n + q) \times p$ con entradas dadas por

$$a_{2k-1,j} = \begin{cases} 1, & \text{si } j \in J_{k,b}; \\ -1, & \text{si } j \in J_{k,t}; \\ 0, & \text{de otro modo;} \end{cases} \quad \begin{matrix} k = 1, \dots, n, \\ j = 1, \dots, p. \end{matrix}$$

$$a_{2k,j} = \begin{cases} 1, & \text{si } j \in J_{k,r}; \\ -1, & \text{si } j \in J_{k,l}; \\ 0, & \text{de otro modo} \end{cases} \quad \begin{matrix} k = 1, \dots, n, \\ j = 1, \dots, p. \end{matrix}$$


$$a_{2n+i,j} = \begin{cases} 1, & \text{si } j = \iota_i; \\ -1, & \text{si } j = \varsigma_i; \\ 0, & \text{de otro modo} \end{cases} \quad \begin{matrix} i = 1, \dots, q, \\ j = 1, \dots, p. \end{matrix}$$


- \mathbf{b} es un vector de $2n + q$ componentes enteras dadas por:


$$\begin{aligned} b_{2j-1} &= s_{j,b} - s_{j,t}, & j &= 1, \dots, n; \\ b_{2j} &= s_{j,r} - s_{j,l}, & j &= 1, \dots, n; \\ b_{2n+j} &= 0, & j &= 1, \dots, q. \end{aligned}$$

- \mathbf{m} es un vector de p variables enteras que son los números compatibles de puntos.

Observaciones 5.3:

 El número de ecuaciones (número de renglones) es el doble de subregiones más el número de cortes. El número de incógnitas (número de columnas) es el número de curvas de la descomposición de la frontera.

 El sistema de ecuaciones lineales (5.4) tiene renglones linealmente independientes y más incógnitas que ecuaciones ($2n + q < p$) debido a que separamos la frontera de cada bloque en al menos cuatro curvas y tenemos exactamente dos curvas por cada corte. Así que este sistema es *compatible indeterminado*, esto es, admite un conjunto infinito de soluciones.

 Si la región Ω es simplemente conexa, entonces la *descomposición admisible* de Ω en n bloques requiere exactamente $q = n - 1$ cortes, y por consiguiente la matriz A tiene tamaño $(3n - 1) \times p$.

 El número de puntos para los lados de abajo y arriba del bloque k es

$$M_k = \sum \{m_j : j \in J_{k,b}\} - s_{k,b} + 1, \quad (5.5)$$

mientras que el número de puntos para los lados derecho e izquierdo del mismo bloque es

$$N_k = \sum \{m_j : j \in J_{k,r}\} - s_{k,r} + 1. \quad (5.6)$$

5.2.2. Compatibilidad por Programación Entera

Como el sistema de ecuaciones lineales (5.4) es compatible indeterminado, podemos elegir tamaños compatibles de diferentes maneras. Sin embargo, no cualquier elección de tamaños es recomendable. Las mallas pueden tener tamaños enormes, las celdas pueden ser elongadas e incluso la calidad de las mallas puede verse comprometida. Una elección pertinente para los tamaños compatibles de las mallas considera las longitudes de las curvas de la frontera.

Elegimos tamaños compatibles de modo que el número total de puntos de las curvas de D sea mínimo y que el número de puntos de las curvas de D dependa de la longitud de las curvas. De manera más precisa, resolvemos el siguiente problema de programación entera lineal:

$$\min_{\mathbf{m} \in \mathbb{Z}^p} \sum_{j=1}^p m_j \quad (5.7)$$

sujeto al sistema de ecuaciones lineales (5.4) y la siguiente restricción:

El número m_j de puntos de la curva Q_j es más grande que una proporción de la longitud de Q_j respecto a la curva más corta de D .

Sea ℓ_j la longitud de la curva Q_j . Denotamos por ℓ_{\min} a la longitud de la curva más corta de D . Entonces la restricción anterior es:

$$m_j \geq \lceil \ell_j / \ell_{\min} \rceil, \quad j = 1, \dots, p. \quad (5.8)$$

Sean

$$\begin{aligned} \mathbf{m} &= [m_1 \ \dots \ m_p]^t, & \boldsymbol{\ell} &= [\ell_1 \ \dots \ \ell_p]^t, \\ \mathbf{0} &= \underbrace{[0 \ \dots \ 0]}_p^t, & \mathbf{1} &= \underbrace{[1 \ \dots \ 1]}_p^t. \end{aligned}$$

Entonces el Problema 5.7 se escribe como:

$$\min_{\mathbf{m} \in \mathbb{Z}^p} \mathbf{1}^t \mathbf{m},$$

y las restricciones (5.8) se escriben como:

$$\mathbf{m} \geq \lceil \boldsymbol{\ell} / \ell_{\min} \rceil,$$

donde la desigualdad y las operaciones son componente a componente. Adicionalmente, agregamos una constante $d \in \mathbb{N}$ a la restricción para controlar la densidad de la malla como:

$$\mathbf{m} \geq d \lceil \boldsymbol{\ell} / \ell_{\min} \rceil. \quad (5.9)$$

De este modo, la elección de tamaños compatibles se formula como un problema lineal de programación entera sujeto a restricciones lineales que consiste en minimizar el número total de puntos de las curvas sujeto a las ecuaciones de compatibilidad (5.4) y las desigualdades (5.9) que relacionan el número de puntos con las longitudes de las curvas.

Problema de la elección de tamaños compatibles. Dado $d \in \mathbb{N}$, resolver

$$\min\{\mathbf{1}^t \mathbf{m} : \mathbf{m} \in \mathbb{Z}^p, A\mathbf{m} = \mathbf{b}, \mathbf{m} \geq d\lfloor \ell/\ell_{\min} \rfloor\}. \quad (5.10)$$

Condiciones suficientes para un problema factible. Queremos que el Problema 5.10 sea factible en el sentido de que tenga soluciones enteras. Para verificar esto, lo formulamos como el siguiente problema equivalente:

$$\min\{\mathbf{1}^t \widehat{\mathbf{m}} : \widehat{\mathbf{m}} \in \mathbb{Z}^p, A\widehat{\mathbf{m}} = \widehat{\mathbf{b}}, \widehat{\mathbf{m}} \geq \mathbf{0}\}, \quad (5.11)$$

donde usamos el siguiente cambio de variable:

$$\widehat{\ell} = d\lfloor \ell/\ell_{\min} \rfloor, \quad \widehat{\mathbf{m}} = \mathbf{m} - \widehat{\ell}, \quad \widehat{\mathbf{b}} = \mathbf{b} - A\widehat{\ell}.$$

Consideramos el poliedro

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^p : A\mathbf{x} \leq \widehat{\mathbf{b}}, \mathbf{x} \geq \mathbf{0}\}.$$

El Problema 5.11 es factible si los vértices del poliedro \mathcal{P} son enteros. Más aún, todos los vértices del poliedro \mathcal{P} son enteros si y sólo si el problema

$$\min_{\mathbf{x} \in \mathcal{P}} \mathbf{c}^t \mathbf{x}$$

tiene solución entera para cualquier vector \mathbf{c} de p componentes enteras.

Una condición suficiente para que los vértices de \mathcal{P} sean enteros es que la matriz A sea una *matriz totalmente unimodular*, esto es, que todas sus submatrices cuadradas tengan determinante igual a 0, 1 o -1 .

Teorema 5.1 (Hoffman-Kruskal [121]). *Sea $A \in \mathbb{Z}^{s \times p}$. Entonces A es una matriz totalmente unimodular si y sólo si los vértices del poliedro \mathcal{P} son enteros para cualquier vector $\widehat{\mathbf{b}} \in \mathbb{Z}^s$.*

A consecuencia del Teorema de Hoffman-Kruskal, la elección de tamaños compatibles es un problema factible si las ecuaciones de compatibilidad poseen una estructura adecuada. Tenemos condiciones suficientes para que estas ecuaciones formen una matriz totalmente unimodular.

Proposición 5.2 ([121]). *Sea $A \in \{0, -1, 1\}^{s \times p}$. Si cada columna de A tiene a lo más un 1 y lo más un -1 , entonces A es una matriz totalmente unimodular.*

A continuación mostramos que la matriz A del sistema de ecuaciones lineales (5.4) cumple las hipótesis de la Proposición 5.2, y por consiguiente es una matriz totalmente unimodular.

La matriz A tiene sus entradas en $\{0, -1, 1\}$. Sea A_j la j -ésima columna de A . Afirmamos que A_j tiene a lo más un 1 y lo más un -1 . En efecto, por construcción, A_j está asociada a una sola curva Q_j de la descomposición D de la frontera. Esta curva está contenida en la frontera de la región Ω o es un corte de Ω .

- Si Q_j está contenida en la frontera de Ω , entonces el número m_j de puntos de Q_j tiene asociado una sola ecuación de compatibilidad en (5.2) o (5.3), y por consiguiente la columna A_j tiene exactamente una componente distinta de cero que es 1 o -1 .
- Supongamos que Q_j es un corte de Ω . Entonces existe otra curva $Q_{j'}$ de D que coincide con Q_j de modo que Q_j y $Q_{j'}$ se encuentran en las fronteras de bloques distintos.

Sea $A_{j'}$ la columna de A asociada a $Q_{j'}$. Como las curvas Q_j y $Q_{j'}$ son cortes, entonces cada curva satisface una ecuación de compatibilidad para los lados opuestos y otra por ser corte. Así que A_j y $A_{j'}$ tienen exactamente dos componentes distintas de cero que son 1 o -1 , la primera corresponde a la compatibilidad de lados opuestos y la segunda corresponde a la compatibilidad de los cortes. Basta probar que las componentes distintas de cero tanto de A_j como de $A_{j'}$ tienen signos contrarios.

Reordenamos los lados de los bloques de modo que Q_j y $Q_{j'}$ se encuentren en lados opuestos de sus respectivos bloques. De este modo, las primeras componentes distintas de cero de A_j y $A_{j'}$ tienen signos opuestos. Las segundas componentes distintas de cero se pueden escoger con signo contrario respecto a las primeras, ya que están asociadas a las ecuaciones (5.1) para compatibilidad de cortes.

Con base a lo anterior, la elección de tamaños compatibles es un problema factible debido a que la matriz del sistema de ecuaciones (5.4) es una matriz totalmente unimodular. Ahora, procedemos a resolver el Problema 5.10.

Método para resolver la programación entera. Necesitamos resolver un problema lineal de programación entera sujeto a restricciones lineales. El *método de ramificación y corte* es un método de la literatura en Optimización para resolver esta clase de problemas. Se basa en la idea de que el conjunto factible de soluciones se puede descomponer en subconjuntos con las soluciones del problema original con menos restricciones.

El *método de ramificación y corte* usa un diagrama de *árbol* para organizar el esquema de descomposición. Los nodos son subconjuntos de soluciones y las *ramas* representan restricciones, a saber, cotas sobre las variables que representan planos de corte.

En la *raíz* del *árbol* se resuelve el problema sin la restricción a soluciones enteras. Esto puede realizarse con el *método simplex* para programación lineal [91]. Luego, en cada paso se agregan restricciones y la solución correspondiente se redondea. La solución entera factible de un problema de minimización se alcanza en la *hoja* del *árbol* con la cota inferior más pequeña.

Para más detalles sobre el *método de ramificación y corte* consulte [141]. El optimizador *COIN-OR Branch and Cut (Cbc)* [40] tiene una implementación eficiente de este método. Nosotros usamos este optimizador para resolver el Problema 5.10.

Ejemplo 5.1. Mostramos el sistema de ecuaciones lineales compatibles (5.10) y una solución al Problema de programación entera 5.10 asociados a las fronteras compatibles de la región poligonal Ω de la Fig. 5.1.

La región Ω se separa en tres bloques Ω_1, Ω_2 y Ω_3 mediante dos cortes. La unión de las fronteras de los bloques se separa en 17 curvas poligonales Q_j que forman los lados $\Omega_{k,b}, \Omega_{k,r}, \Omega_{k,t}, \Omega_{k,l}$ de los bloques como se muestra en la Fig. 5.5.

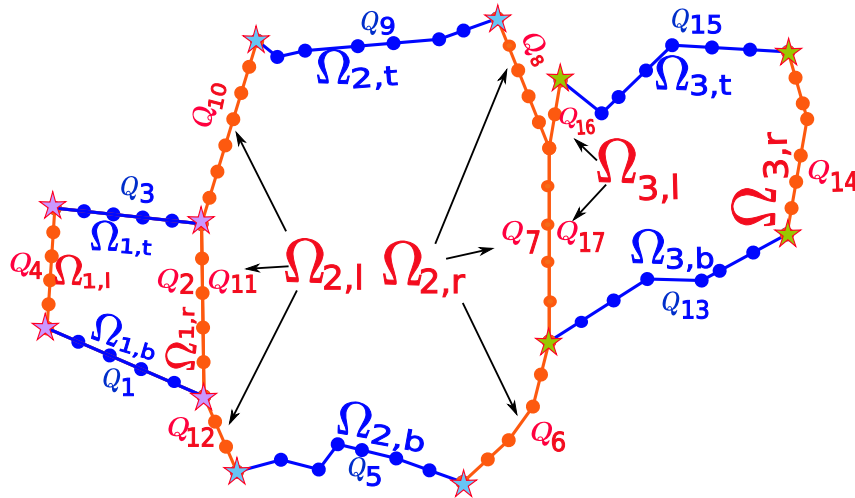


Figura 5.5: Las 17 curvas Q_j y los cuatro lados $\Omega_{k,b}, \Omega_{k,r}, \Omega_{k,t}, \Omega_{k,l}$ de los tres bloques Ω_k de la Fig. 5.1 se reparametrizan con un número compatible de puntos.

El número m_j de puntos para las curvas Q_j debe cumplir las siguientes ecuaciones de compatibilidad:

$$\begin{array}{ll}
 m_1 - m_3 = 0, & \Omega_{1,b} \text{ y } \Omega_{1,t} \text{ compatibles} \\
 m_2 - m_4 = 0, & \Omega_{1,r} \text{ y } \Omega_{1,l} \text{ compatibles} \\
 m_5 - m_9 = 0, & \Omega_{2,b} \text{ y } \Omega_{2,t} \text{ compatibles} \\
 m_6 + m_7 + m_8 - m_{10} - m_{11} - m_{12} = 0, & \Omega_{2,r} \text{ y } \Omega_{2,l} \text{ compatibles} \\
 m_{14} - m_{16} - m_{17} = -1, & \Omega_{3,b} \text{ y } \Omega_{3,t} \text{ compatibles} \\
 m_{13} - m_{15} = 0, & \Omega_{3,r} \text{ y } \Omega_{3,l} \text{ compatibles} \\
 -m_2 + m_{11} = 0, & \Omega_1 \text{ y } \Omega_2 \text{ se intersectan en corte} \\
 -m_7 + m_{17} = 0. & \Omega_2 \text{ y } \Omega_3 \text{ se intersectan en corte}
 \end{array}$$

Una solución para el Problema 5.10 asociado a estas ecuaciones con factor de densidad

$d = 1$ está dada como sigue:

$$\begin{aligned} \text{(bloque 1)} \quad & m_1 = m_2 = m_3 = m_4 = 6, \\ \text{(bloque 2)} \quad & m_5 = 8, \quad m_6 = m_7 = m_8 = 6, \\ & m_9 = m_{10} = 8, \quad m_{11} = 6, \quad m_{12} = 4, \\ \text{(bloque 3)} \quad & m_{13} = m_{14} = m_{15} = 8, \quad m_{16} = 3, \quad m_{17} = 6. \end{aligned}$$

Reparametrizamos las curvas Q_j con este número compatible de puntos. Los puntos sobre la frontera no están distribuidos de manera uniforme para preservar los puntos de la frontera original.

5.3. Reparametrización de la Frontera

Completamos la construcción de las parametrizaciones compatibles de las fronteras. Una vez que tenemos el número compatible de puntos para las fronteras, las reparametrizamos con ese número de puntos, y luego parametrizamos los lados de los bloques por curvas spline lineales.


En secciones anteriores separamos las fronteras de los bloques en curvas poligonales y determinamos su número compatible de puntos resolviendo un problema de programación entera. Ahora, reparametrizamos estas curvas con el número compatible de puntos. Sea Q_j una curva poligonal de la descomposición D . Sea m_j la j -ésima componente de la solución \mathbf{m} del Problema 5.10. Reparametrizamos Q_j por su longitud de arco con m_j puntos. A continuación describimos esta reparametrización.

Sea γ una curva poligonal abierta de longitud L con vértices $v_i = (x_i, y_i)$, $i = 1, \dots, s$. Reparametrizamos γ como una curva poligonal con m nodos equidistantes. Denotamos por L_i a la longitud de γ del vértice v_1 hasta el vértice v_i . Construimos una *curva B-spline lineal* β_x que interpola los puntos (L_i, x_i) , $i = 1, \dots, s$ y una *curva B-spline lineal* β_y que interpola los puntos (L_i, y_i) , $i = 1, \dots, s$. Formamos la *curva spline lineal parametrizada*

$$\beta(t) = (\beta_x(t), \beta_y(t)) : [0, L] \rightarrow \mathbb{R}^2$$

Evaluamos la curva β en una partición uniforme $0 = t_1 < \dots < t_m = L$ de m puntos del intervalo $[0, L]$ para obtener la curva poligonal $\hat{\gamma}$ con vértices $\beta(t_1), \dots, \beta(t_m)$. La curva $\hat{\gamma}$ es una reparametrización de γ con m puntos equidistantes.

Observaciones 5.4:

 Nosotros reparametrizamos las curvas con puntos equidistantes, sin embargo esto no es una condición necesaria, basta que los puntos coincidan en los cortes. La distribución de puntos sobre una curva puede cambiarse reparametrizando la curva, más aún, la calidad de las mallas puede mejorar si sus puntos frontera se distribuyen adecuadamente. La dificultad es determinar automáticamente distribuciones adecuadas para los puntos.

☞ Se pueden perder algunos detalles de la región en la reparametrización de la frontera. Por lo que cuidamos que las curvas reparametrizadas preserven la forma de la región en la medida de lo posible.

Reparametrizamos cada curva Q_j de la descomposición D con el número compatible de puntos. Denotamos por \widehat{Q}_j a la reparametrización de la curva Q_j con m_j puntos. A su vez reparametrizamos los lados de cada bloque. Los lados reparametrizados son la unión de las curvas reparametrizadas que conectan esquinas:

$$\widehat{\Omega}_{k,\text{side}} = \bigcup_{j \in J_{k,b}} \widehat{Q}_j, \quad \text{side} = b, t, r, l, \quad k = 1, \dots, n.$$

Por construcción, los lados opuestos de cada bloque tienen el mismo número de puntos:

- El número M_k de puntos para $\widehat{\Omega}_{k,b}$ y $\widehat{\Omega}_{k,t}$ está dado por (5.5).
- El número N_k de puntos para $\widehat{\Omega}_{k,r}$ y $\widehat{\Omega}_{k,l}$ está dado por (5.6).

Así que en cada bloque Ω_k podemos ordenar los puntos de los lados en correspondencia con los índices de una matriz de $M_k \times N_k$ como sigue:

puntos del lado inferior	→	primera columna de la matriz
puntos del lado derecho	→	último renglón de la matriz
puntos del lado superior	→	última columna invertida de la matriz
puntos del lado izquierdo	→	primer renglón invertido de la matriz

Denotamos a los puntos sobre la frontera del bloque Ω_k por $P_{i,j}^{(k)}$ con índices (i, j) en el siguiente conjunto:

$$\{1, M_k\} \times \{1, \dots, N_k\} \cup \{1, N_k\} \times \{1, \dots, M_k\}$$

de modo que

$$\begin{aligned} \widehat{\Omega}_{k,b} &= \left\{ P_{i,1}^{(k)} : i = 1, \dots, M_k \right\}, \\ \widehat{\Omega}_{k,r} &= \left\{ P_{M_k,j}^{(k)} : j = 1, \dots, N_k \right\}, \\ \widehat{\Omega}_{k,t} &= \left\{ P_{i,N_k}^{(k)} : i = 1, \dots, M_k \right\}, \\ \widehat{\Omega}_{k,l} &= \left\{ P_{1,j}^{(k)} : j = 1, \dots, N_k \right\}. \end{aligned}$$

De esta manera, cada lado de los bloques queda reparametrizado con puntos compatibles.

Ahora, parametrizamos los lados de los bloques por *curvas B-spline lineales*

$$\begin{aligned} \psi_{k,b} : [0, 1] &\rightarrow \widehat{\Omega}_{k,b}, & \psi_{k,t} : [0, 1] &\rightarrow \widehat{\Omega}_{k,t}, \\ \psi_{k,r} : [0, 1] &\rightarrow \widehat{\Omega}_{k,r}, & \psi_{k,l} : [0, 1] &\rightarrow \widehat{\Omega}_{k,l}. \end{aligned}$$

dadas por

$$\begin{aligned} \psi_{k,b}(\xi) &= \sum_{i=1}^{M_k} P_{i,1}^{(k)} B_{i,M_k}^2(\xi), & \psi_{k,t}(\xi) &= \sum_{i=1}^{M_k} P_{i,N_k}^{(k)} B_{i,N_k}^2(\xi), & 0 \leq \xi \leq 1, \\ \psi_{k,r}(\eta) &= \sum_{j=1}^{N_k} P_{M_k,j}^{(k)} B_{j,N_k}^2(\eta), & \psi_{k,l}(\eta) &= \sum_{j=1}^{N_k} P_{1,j}^{(k)} B_{j,N_k}^2(\eta), & 0 \leq \eta \leq 1, \end{aligned} \quad (5.12)$$

donde

B_{i,M_k} el i -ésimo B -spline lineal (2.2) con sucesión de nodos dada por una partición de M_k puntos del intervalo $[0, 1]$ para $i = 1, \dots, M_k$,

B_{j,N_k} el j -ésimo B -spline lineal (2.2) con sucesión de nodos dada por una partición de N_k puntos del intervalo $[0, 1]$ para $j = 1, \dots, N_k$.

En cada bloque combinamos las parametrizaciones de los lados para conseguir una parametrización ψ_k de la frontera del cuadrado unitario $[0, 1] \times [0, 1]$ a la frontera del bloque Ω_k dada por *curvas B-spline lineales*:

$$\psi_k(\xi, \eta) = \begin{cases} \psi_{k,b}(\xi), & \text{si } \eta = 0; \\ \psi_{k,r}(\eta), & \text{si } \xi = 1; \\ \psi_{k,t}(\xi), & \text{si } \eta = 1; \\ \psi_{k,l}(\eta), & \text{si } \xi = 0, \end{cases} \quad 0 \leq \xi, \eta \leq 1, \quad k = 1, \dots, n. \quad (5.13)$$

Por construcción, nuestras parametrizaciones ψ_k de las fronteras de los bloques Ω_k son compatibles.

5.4. Resumen del Capítulo

El primer paso para generar una malla estructurada por bloques sobre una región poligonal Ω a partir de una *descomposición admisible* $\{\Omega_k\}$ es generar sus puntos compatibles en la frontera, esto puede verse como construir una familia de mapeos compatibles de los lados del cuadrado unitario S a las fronteras de las subregiones Ω_k . Construimos estas parametrizaciones de las fronteras como sigue:

1. **Descomposición de la frontera.** Separamos la frontera de cada subregión Ω_k en cuatro lados eligiendo las esquinas con base a una heurística, luego separamos cada lado en cortes y curvas sobre la frontera de Ω para obtener una colección $\{Q_j\}$ de curvas poligonales.
2. **Generación de tamaños compatibles.** Determinamos el número de puntos de las curvas Q_j de modo que por cada corte tengamos dos curvas con los mismos puntos y que cada subregión tenga lados opuestos con mismo número de puntos, formulamos

estas condiciones como el sistema de ecuaciones lineales enteras indeterminado (5.4). Buscamos la solución que minimiza el número total de puntos y acota el número de puntos de cada curva por una proporción de su longitud, por lo que resolvemos el Problema 5.10 de programación lineal entera para hallar el número compatible de puntos.

3. **Reparametrización de la frontera.** Reparametrizamos cada curva Q_j como una *curva B-spline lineal* con el número compatible nodos y unimos las curvas reparametrizadas para obtener por *curvas B-spline lineales compatibles* $\psi_k : S \rightarrow \Omega_k$.

Nuestra construcción es viable porque el Problema 5.10 es factible, para probarlo verificamos que la matriz de coeficientes del sistema de ecuaciones (5.4) cumple las hipótesis del Teorema de Hoffman-Kruskal, que garantiza soluciones enteras para una colección de problemas de optimización lineal entera.

Resolvemos el Problema 5.10 con el *método de ramificación y corte* para generar automáticamente los tamaños compatibles, así la construcción de *parametrizaciones compatibles* de las fronteras se puede reducir a la *elección de esquinas*.

En el siguiente capítulo generamos mallas estructuradas con puntos frontera fijos para extender las *parametrizaciones compatibles* de las fronteras al interior de la región, a saber, extendemos las fronteras parametrizadas por *curvas B-spline lineales* como los *productos tensoriales B-spline bilineales* (2.5).

Una vez que separamos la región en subregiones y construimos parametrizaciones compatibles para las fronteras de las subregiones, extendemos las parametrizaciones al interior de la región.

Problema 6.1. Sea $\{\Omega_k\}$ una descomposición de una región poligonal Ω en n subregiones mediante líneas de corte. Sea ψ_k una familia de mapeos compatibles de ∂S a $\partial\Omega_k$, donde $S = [0, 1] \times [0, 1]$. Construir una familia $\{X_k\}$ de mapeos admisibles de S a Ω_k tales que

$$X_k|_{\partial S} = \psi_k, \quad k = 1, \dots, n.$$

Extendemos las parametrizaciones de las fronteras como una familia de mapeos bilineales *admisibles* y *compatibles* dados por (2.5) que transforman el cuadrado unitario en las subregiones. Estos mapeos requieren *mallas de control* estructuradas y compatibles. Por lo que nuestro problema es generar mallas estructuradas por bloques a partir de las fronteras compatibles.

Problema 6.2. Sea $\{\Omega_k\}$ una descomposición de una región poligonal Ω en n subregiones mediante líneas de corte tal que $\{\partial\Omega_k\}$ es una colección de curvas compatibles. Generar una malla estructurada de cuadriláteros G_Ω de alta calidad sobre Ω con n bloques G_k tales que

$$\partial G_k = \partial\Omega_k, \quad k = 1, \dots, n.$$

Para abordar el Problema 6.2 realizamos los siguientes pasos:

1. Generar mallas estructuradas sobre cada bloque
2. Unir las mallas en una malla global
3. Mejorar la calidad de la malla global

Generamos mallas iniciales sobre los bloques por un método algebraico en §6.1. Explicamos cómo unir estas mallas en §6.2. Posteriormente, en §6.3 suavizamos las mallas para mejorar su calidad. Finalmente, en §6.4 damos una breve revisión del método variacional que empleamos para suavizar mallas estructuradas.

Observaciones 6.1: La calidad de la malla global puede mejorar si no fijamos los puntos sobre los cortes. Por eso en los problemas 6.1 y 6.2 empleamos una descomposición por líneas de corte. Esta descomposición se puede obtener a partir de una *descomposición admisible* cambiando las fronteras de las subregiones. Los cortes (segmentos de recta) se suavizan en *líneas de corte* (curvas poligonales).

6.1. Interpolación Transfinita

En cada bloque generamos una malla estructurada de cuadriláteros por el método algebraico de interpolación transfinita (TFI). El interpolante es un mapeo que extiende las curvas parametrizadas de las fronteras al interior de la región. Este mapeo deforma continuamente los lados opuestos de los bloques. Para construirlo es necesario que los lados opuestos de los bloques tengan el mismo número de puntos. Esta condición se cumple porque construimos lados compatibles. Basta examinar la TFI de un bloque, pues la construcción es análoga para los otros bloques.

Supongamos que Ω es una *región admisible*. Sean ψ_b , ψ_r , ψ_t y ψ_l los lados compatibles de Ω parametrizados por curvas B-spline lineales. Usamos el mapeo $\mathbf{X}_{TFI} : [0, 1] \times [0, 1] \rightarrow \Omega$ dado por

$$\mathbf{X}_{TFI}(\xi, \eta) = H_{bt}(\xi, \eta) + H_{rl}(\xi, \eta) - H_c(\xi, \eta), \quad 0 \leq \xi, \eta \leq 1,$$

donde

- $H_{bt}(\xi, \eta) = (1 - \eta)\psi_b(\xi) + \eta\psi_t(\xi)$ deforma continuamente el lado inferior al superior.
- $H_{rl}(\xi, \eta) = (1 - \xi)\psi_l(\eta) + \xi\psi_r(\eta)$ deforma continuamente el lado izquierdo al derecho.
- $H_c(\xi, \eta) = (1 - \xi)\lambda_l(\eta) + \xi\lambda_r(\eta)$ deforma continuamente los segmentos

$$\begin{aligned} \lambda_l(\eta) &= (1 - \eta)\psi_b(0) + \eta\psi_t(0), \\ \lambda_r(\eta) &= (1 - \eta)\psi_b(1) + \eta\psi_t(1). \end{aligned}$$

Podemos visualizar \mathbf{X}_{TFI} como una sucesión de curvas entre los lados opuestos, Fig. 6.1. Los puntos de la malla se obtienen al evaluar este mapeo en una rejilla uniforme del cuadrado unitario.

Supongamos que el lado Ω_b tiene M puntos y que el lado Ω_r tiene N puntos. Sean $\{\xi_i\}$ y $\{\eta_j\}$ particiones uniformes de M y N puntos del intervalo $[0, 1]$, respectivamente. Generamos una malla estructurada G de tamaño $M \times N$ sobre Ω con puntos dados por

$$\mathbf{p}_{i,j} = \mathbf{X}_{TFI}(\xi_i, \eta_j), \quad i = 1, \dots, M, \quad j = 1, \dots, N.$$

De esta manera podemos mallas estructuradas G_k de tamaños $M_K \times N_k$ por TFI en cada bloque Ω_k de la *descomposición admisible*.

Observaciones 6.2:

- ☞ Por construcción las celdas de las mallas generadas por TFI empatan en los cortes. Los tamaños compatibles $M_K \times N_k$ de las mallas están dados por (5.5) y (5.6).
- ☞ El mapeo \mathbf{X}_{TFI} puede escribirse como el *producto tensorial B-spline bilineal* (2.5).
- ☞ Las mallas generadas por TFI pueden estar dobladas o tener baja calidad, esto es, el mapeo \mathbf{X}_{TFI} puede no ser inyectivo, Fig. 6.2

En la siguiente sección describimos como unir las mallas TFI de cada bloque en una malla global.

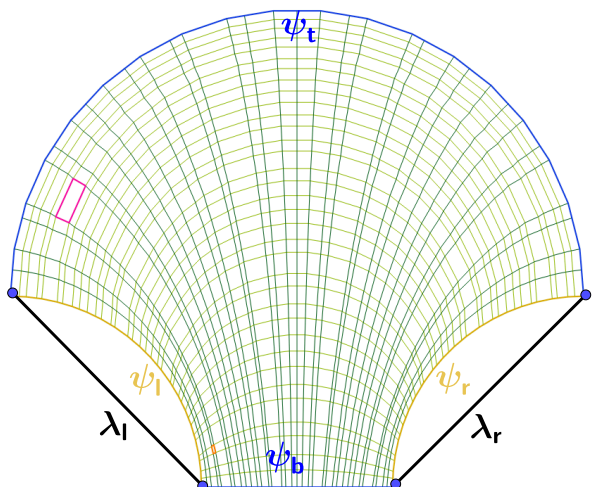


Figura 6.1: Malla estructurada generada por el mapeo X_{TFI} .

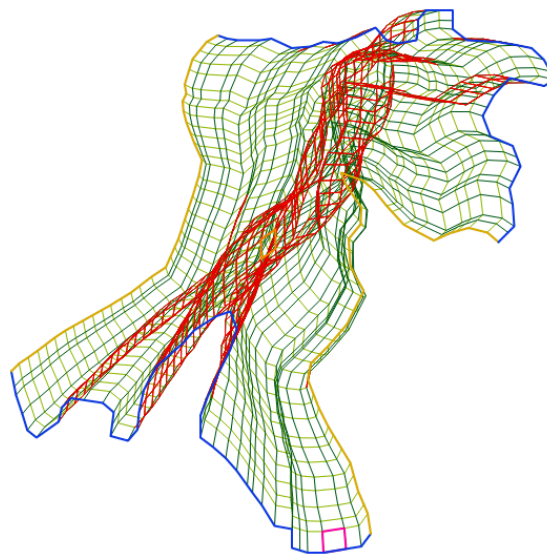


Figura 6.2: Malla doblada sobre la Bahía de la Habana con celdas no convexas de color rojo.

6.2. Unión de Mallas

En esta sección unimos las mallas estructuradas G_k sobre los bloques Ω_k en una sola malla global G_Ω sobre la región Ω , a saber, una malla estructurada por bloques. Primero, precisamos cómo es la unión de mallas compatibles.

Sea $\{G_k\}$ una colección de n mallas compatibles de cuadriláteros sobre *subregiones admisibles* de Ω . La unión de las mallas G_k , es una malla de cuadriláteros sobre Ω , denotada por

$$G_\Omega = \bigcup_{k=1}^n G_k, \quad (6.1)$$

tal que

- Los puntos frontera de G_Ω son los puntos fronteras de cada malla G_k sin los puntos donde las celdas empatan.
- Los puntos interiores de G_Ω son los puntos interiores de cada malla G_k junto con los puntos donde las celdas empatan.
- Las celdas de G_Ω son las celdas de cada malla G_k .

Para formar la malla G_Ω necesitamos distinguir los puntos interiores de los puntos frontera. Más aún, en algunas aplicaciones es importante evitar repetir puntos y enumerar los puntos de la frontera en orden secuencial.

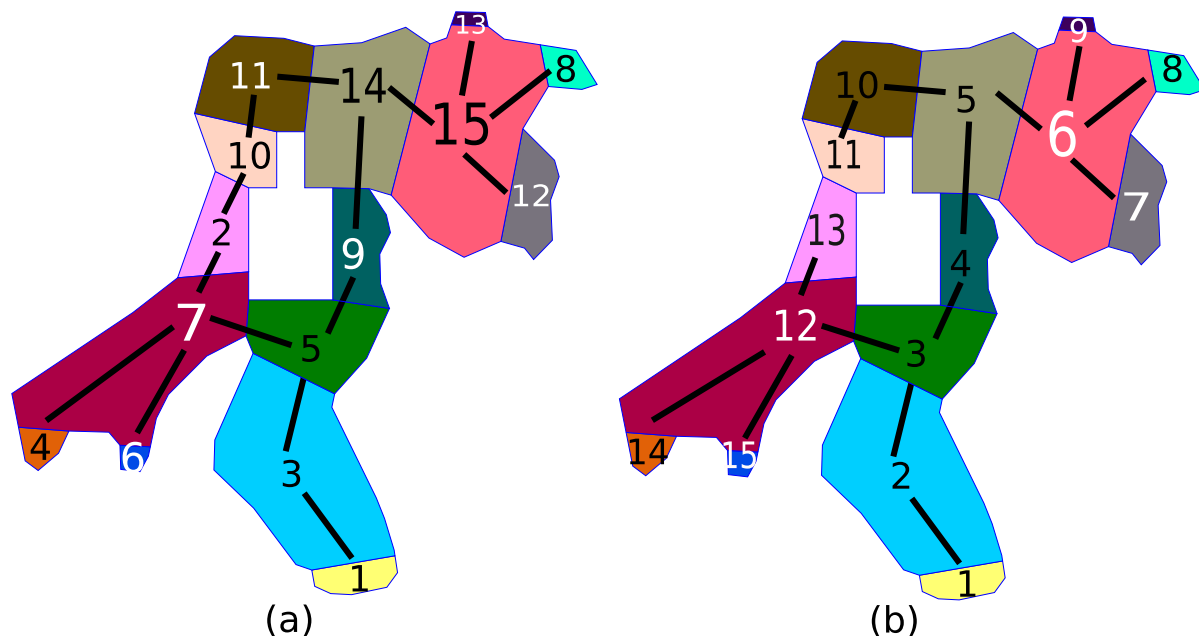


Figura 6.3: *Descomposición admisible* de la Bahía de la Habana con un agujero en 15 bloques: (a) Gráfica asociada a la *descomposición admisible*, (b) un *árbol generador* para la gráfica.

Problema 6.3 (Unión de Mallas). Sea $\{\Omega_k\}$ una *descomposición admisible* de una región poligonal Ω en n bloques. Sea $\{G_k\}$ una colección de mallas estructuradas compatibles para $\{\Omega_k\}$. Generar la malla de cuadriláteros G_Ω sobre Ω dada por la unión de mallas (6.1) sin puntos repetidos y con puntos frontera ordenados secuencialmente.

Unimos las mallas de manera recursiva. En cada paso unimos una malla por bloques con una malla de un solo bloque siguiendo un orden dado.

6.2.1. El orden para la unión de mallas

El orden en que unimos las mallas es una permutación β del conjunto $\{1, \dots, n\}$, donde n es el número de bloques. Para obtener esta permutación empleamos la gráfica \mathcal{G} de §4.2.1 que asociamos a la *descomposición admisible*. Recordamos que los *nodos* de \mathcal{G} son los bloques Ω_k y se conectan por una *arista* si se intersectan en un corte.

Para obtener el orden de la unión construimos un *árbol generador* para la gráfica \mathcal{G} . La permutación β está dada por el orden en que construimos el *árbol generador*, Fig. 6.3.

Métodos para construir *árboles generadores* pueden emplearse para obtener el orden de la unión. Uno de estos métodos es la *Búsqueda en anchura* (*Breadth First Search*) [27]: Elegimos un nodo de la gráfica \mathcal{G} como la raíz de árbol y se exploran todos los vecinos de este nodo. Luego, para cada uno de los vecinos se exploran su vez sus respectivos vecinos adyacentes, así hasta que recorramos toda la gráfica \mathcal{G} .

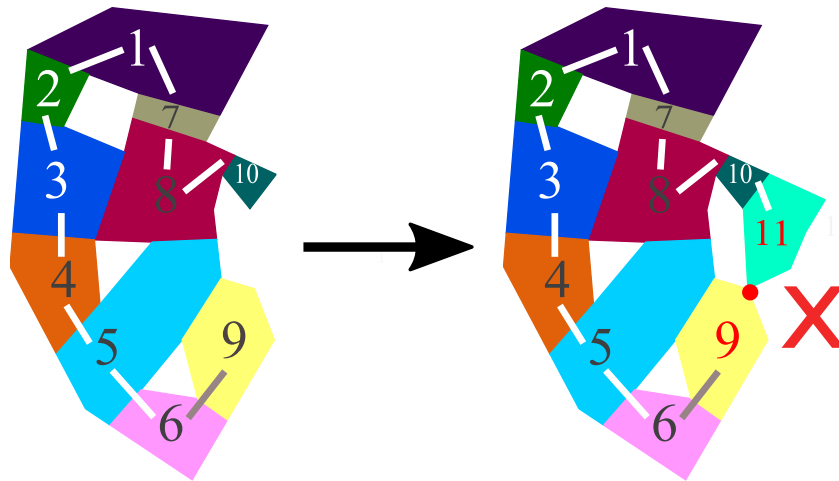


Figura 6.4: Construcción de un árbol generador para unir bloques de una descomposición por cortes de una región con agujeros. Los bloques 9 y 11 se intersectan en un punto de la frontera exterior que a su vez está en la frontera interior.

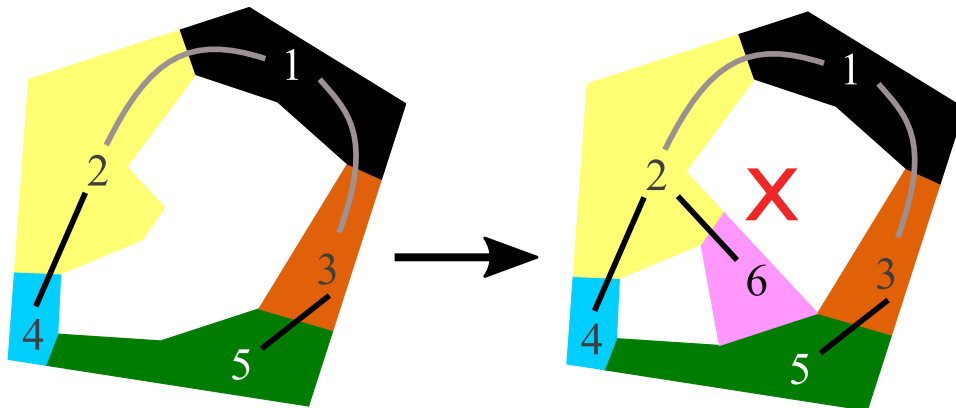


Figura 6.5: Construcción de un árbol generador para una descomposición por cortes de una región con agujeros. Al unir el bloque 6 separamos un agujero en dos agujeros.

Unimos las mallas en el orden con que construimos un *árbol generador*. No obstante, algunos *árboles generadores* pueden no ser adecuados para unir bloques de regiones con agujeros. Cuando unimos un nuevo bloque la frontera exterior se puede intersectar con las fronteras interiores como se muestra en la Fig. 6.4 o bien un agujero se puede separar en más de dos agujeros como se muestra en la Fig. 6.5. En ambos casos es complicado distinguir las fronteras de la región. Así que construimos el *árbol generador* de modo que la unión de los bloques correspondientes cumpla con las siguientes condiciones:

- En cada paso la frontera no tiene autointersecciones.
- Los agujeros del paso anterior son a su vez agujeros del paso actual.

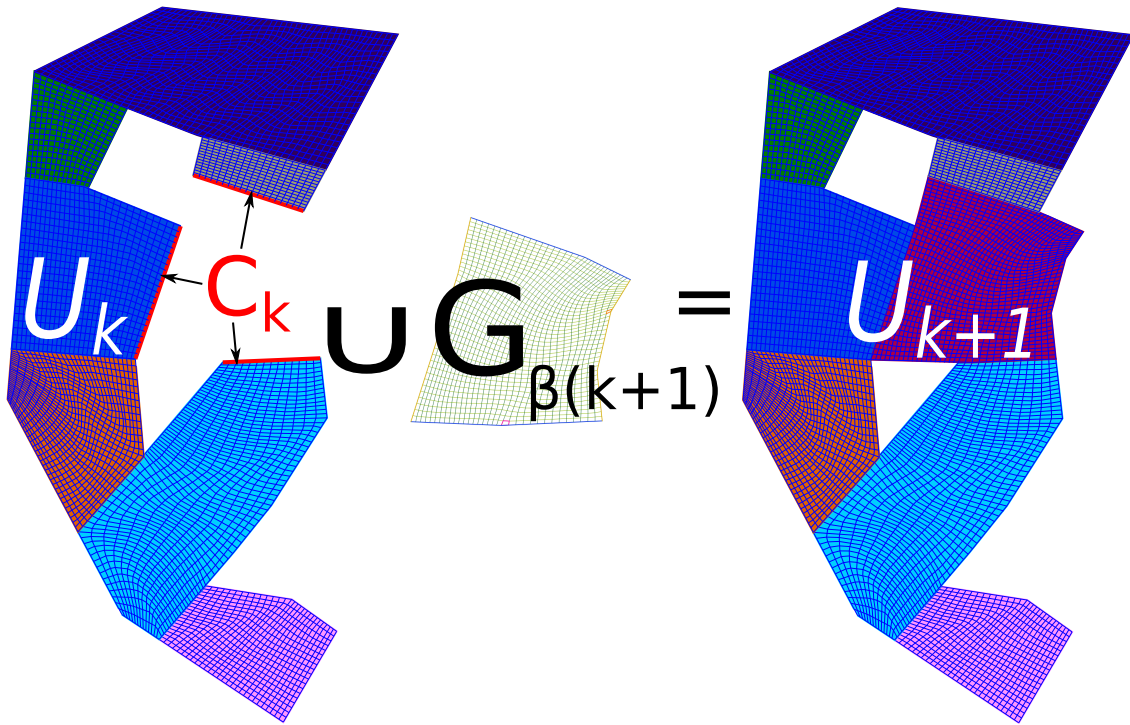


Figura 6.6: La malla estructurada por bloques U_k del paso anterior se une con la malla de un solo bloque $G_{\beta(k+1)}$ para formar la malla U_{k+1} del paso actual, su intersección es el conjunto C_k dado por la unión de cortes en color rojo.

6.2.2. Unión recursiva de mallas

Unimos de manera recursiva una malla por bloques con otra de un solo bloque siguiendo el orden β que obtuvimos en §6.2.1. Iniciamos con la malla

$$U_1 = G_{\beta(1)},$$

luego unimos esta malla con $G_{\beta(2)}$ para obtener la malla

$$U_2 = U_1 \cup G_{\beta(2)}.$$

Ahora, supongamos que en el paso k unimos las mallas de los primeros k bloques respecto al orden β , esto es, formamos la malla

$$U_k = G_{\beta(1)} \cup \dots \cup G_{\beta(k)}.$$

Entonces, en el paso $k + 1$ unimos la malla del paso anterior con $G_{\beta(k+1)}$ para obtener la malla

$$U_{k+1} = U_k \cup G_{\beta(k+1)}.$$

La unión de las mallas U_k y $G_{\beta(k+1)}$ está bien definida debido a que sus celdas empujan en cortes de la región Ω , mostramos su unión en la Fig. 6.6.

Observaciones 6.3:

☞ $U_{k+1} = G_{\beta(1)} \cup \dots \cup G_{\beta(k+1)}.$

☞ Los puntos de la malla U_{k+1} tienen la misma conectividad en las mallas U_k y $G_{\beta(k+1)}$, excepto en el conjunto de cortes C_k .

☞ Cuando unimos la frontera exterior de U_k con la frontera de $G_{\beta(k+1)}$ sin el conjunto de cortes C_k , obtenemos la frontera exterior de la malla U_{k+1} y fronteras interiores que no se encuentran en U_k .

6.2.3. Numeración para la unión de mallas

En cada paso de la unión recursiva de mallas enumeramos sus puntos sin repetirlos de modo que sus puntos frontera estén ordenados secuencialmente, para ello actualizamos la numeración de las mallas U_k y $G_{\beta(k+1)}$. Los índices de la malla U_{k+1} se separan en dos conjuntos en correspondencia uno a uno con los puntos de las mallas U_k y $G_{\beta(k+1)}$, respectivamente. La clave para obtener estas correspondencias es identificar los puntos sobre los cortes.

Sea C_k el conjunto de puntos de la malla U_k sobre los cortes que la conectan con el bloque $G_{\beta(k+1)}$, excepto los extremos de los cortes. Entonces:

- Los puntos interiores de la malla U_{k+1} son los puntos interiores de las mallas U_k y $G_{\beta(k+1)}$ junto con los puntos de C_k ,
- Los puntos frontera de la malla U_{k+1} son los puntos frontera de las mallas U_k y $G_{\beta(k+1)}$ sin los puntos de C_k .

Una vez que identificamos los puntos interiores y frontera de la nueva malla U_{k+1} , los enumeramos en el siguiente orden:

1. Frontera exterior de la nueva malla U_{k+1} .
2. Fronteras interiores de la malla anterior U_k .
3. Fronteras interiores de la nueva malla U_{k+1} que no son fronteras interiores de la malla anterior U_k .
4. Interior de la malla anterior U_k .
5. Interior del bloque $G_{\beta(k+1)}$.
6. Conjunto de cortes C_k .

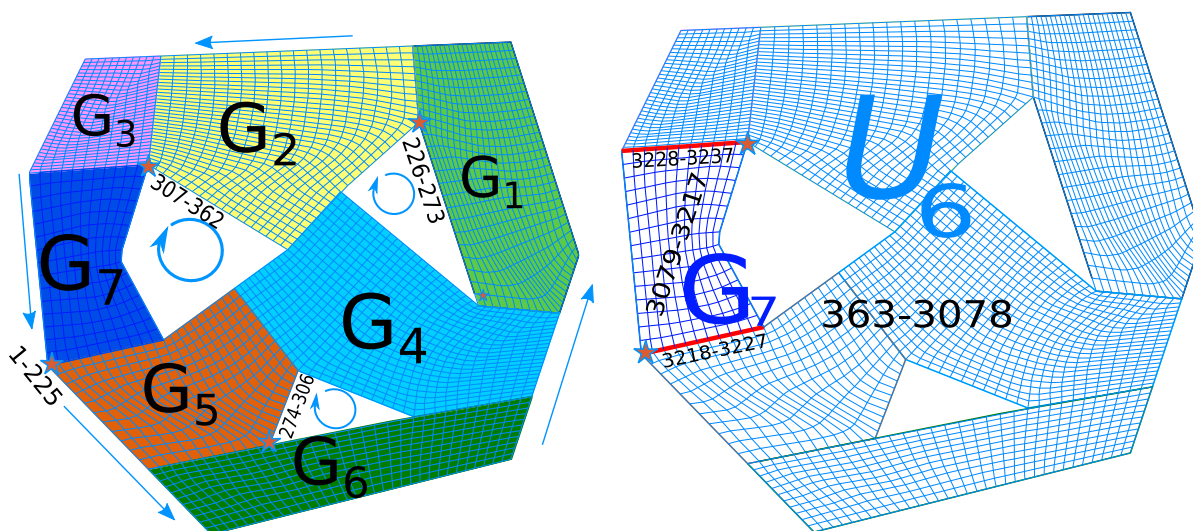
(a) Bloques G_k y numeración de la frontera(b) Unión de mallas $U_7 = U_6 \cup G_7$ y numeración de puntos interiores

Figura 6.7: Unión de siete mallas estructuradas. En el último paso unimos la malla por bloques $U_6 = G_1 \cup \dots \cup G_6$ con el bloque G_7 . En este caso el orden β es la permutación identidad.

Generamos correspondencias uno a uno de los índices de la nueva malla U_{k+1} a los índices de la malla anterior U_k y del bloque $G_{\beta(k+1)}$ siguiendo la numeración anterior. Si los puntos de U_k y $G_{\beta(k+1)}$ se pueden recorrer en orden secuencial con sus fronteras orientadas en sentido antihorario, entonces podemos recorrer la nueva malla U_{k+1} en orden secuencial, comenzando con los puntos de la malla anterior U_k , y luego con los puntos del bloque $G_{\beta(k+1)}$, más aún, conservamos la orientación de su frontera como mostramos en la Fig. 6.7.

Aplicamos inducción para obtener una malla de cuadriláteros sobre Ω sin puntos repetidos y con puntos frontera ordenados secuencialmente como la unión de mallas:

$$G_\Omega = \bigcup_{k=1}^n G_{\beta(k)}.$$

Así, las mallas G_k que generamos por TFI sobre los bloques se unen en la malla global G_Ω , no obstante esta malla puede no tener la calidad deseada e incluso estar doblada, por lo que en la siguiente sección veremos cómo mejorar su calidad.

6.3. Suavizamiento de Mallas por Bloques

En el suavizamiento de mallas fijamos los puntos de la frontera y movemos los puntos interiores para generar celdas con líneas coordenadas más suaves. Cambiamos la calidad de la malla sin modificar su conectividad. Nosotros suavizamos mallas estructuradas por bloques que generamos por TFI para mejorar su calidad.

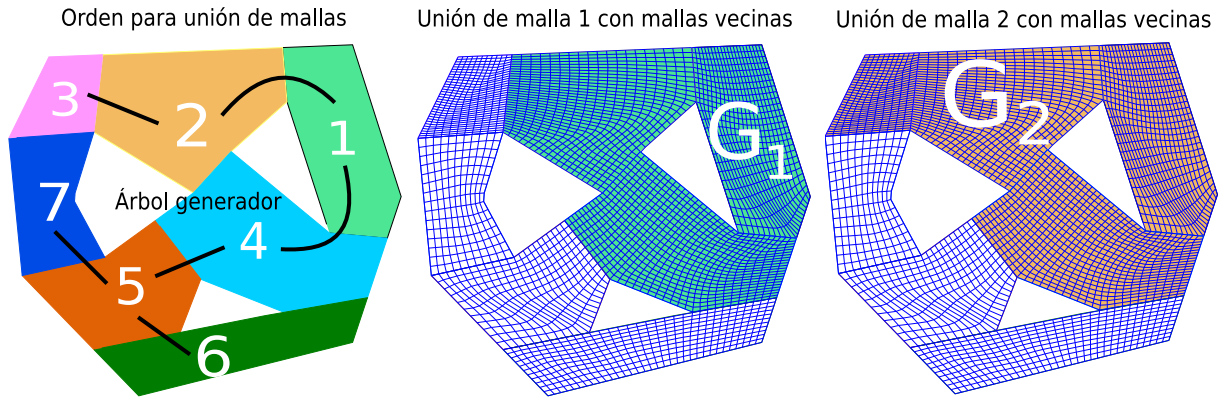


Figura 6.8: Suavizamiento global de una malla estructurada: suavizamos mallas formadas por bloques vecinos siguiendo el orden en que unimos los bloques.

Problema 6.4 (Suavizamiento de Mallas). Sea G_Ω una malla de cuadriláteros estructurada por bloques sobre una región poligonal Ω . Reacomodar los puntos interiores de G_Ω de modo que G_Ω sea de alta calidad.

La malla por bloques G_Ω es la unión de n mallas estructuradas G_k sobre las subregiones Ω_k de la región Ω . Por lo que suavizar G es suavizar cada bloque de modo que las líneas coordenadas que atraviesan bloques adyacentes sean suaves. Esta tarea puede no ser viable si queremos suavizar todas las líneas coordenadas de una sola vez, por lo que realizamos el suavizamiento por etapas donde suavizamos submallas de menor tamaño:

Suavizamiento Local. Suavizamos cada malla estructurada G_k que generamos por TFI por el enfoque variacional discreto. Elegimos un funcional F que mide propiedades geométricas de las celdas y hallamos la malla óptima para este funcional:

$$G_k^* = \arg \min \{F(G) : G \in \mathcal{M}_k\}, \quad (6.2)$$

donde \mathcal{M}_k el conjunto de mallas estructuradas de cuadriláteros con la misma conectividad y puntos frontera que G_k .

Suavizamiento Global. El suavizamiento se lleva a cabo de manera secuencial por grupos de bloques contiguos hasta que recorramos todos los bloques. El orden del suavizamiento es el orden β con el que unimos las mallas. En cada paso la malla $G_{\beta(k)}$ se une con las mallas que comparten los mismos cortes de Ω en una malla estructurada por bloques, denotada por \widehat{G}_k , que suavizamos con el enfoque variacional discreto. Buscamos la malla óptima:

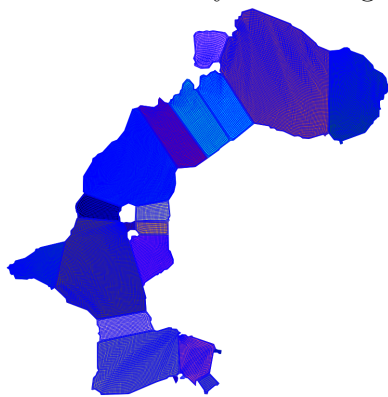
$$\widehat{G}_k^* = \arg \min \{F(G) : G \in \widehat{\mathcal{M}}_k\}, \quad (6.3)$$

donde $\widehat{\mathcal{M}}_k$ es el conjunto de mallas de cuadriláteros con la misma conectividad y puntos frontera que \widehat{G}_k , Fig. 6.8.

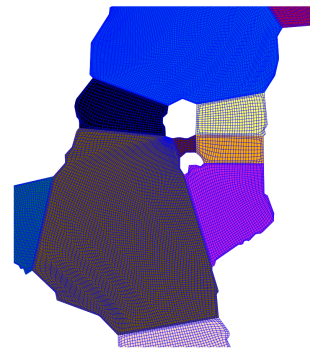
Observaciones 6.4:

- ☞ En el suavizamiento local generamos las mallas óptimas G_k^* a partir de las mallas TFI. Suavizamos cada bloque, pero las líneas coordenadas entre bloques contiguos no son suaves, por lo que se lleva a cabo la segunda etapa de suavizamiento.
- ☞ En el suavizamiento global resolvemos n problemas de optimización. Las fronteras de los bloques cambian en esta etapa debido a que suavizamos los cortes.
- ☞ La elección del funcional adecuado es clave para generar mallas de alta calidad. En ambas etapas empleamos la misma combinación convexa de funcionales dada por (6.9).

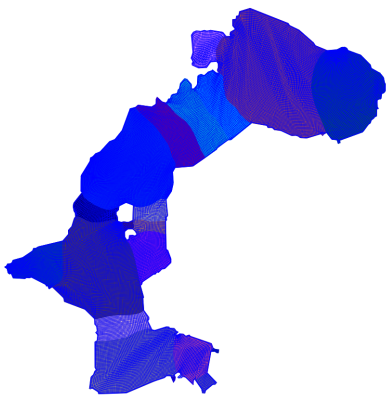
Ejemplo 6.1. El Lago de Pátzcuaro con dos agujeros se descompone por cortes en 21 subregiones, donde generamos mallas estructuradas compatibles por TFI. Suavizamos cada bloque por separado, Figuras 6.9(a)-(b). Luego, suavizamos bloques vecinos para suavizar los cortes y la malla global, Figuras 6.9(c)-(d).



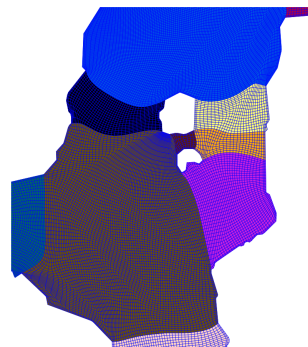
(a) Malla suavizada en cada bloque



(b) Acercamiento a la malla de (a)



(c) Malla suavizada en toda la región



(d) Acercamiento a la malla de (c)

Figura 6.9: Suavizamos una malla estructurada de 21 bloques sobre el Lago de Pátzcuaro.

Ejemplo 6.2. Generamos una malla estructurada suave de 17 bloques sobre el Golfo de México a partir de la descomposición por cortes que vimos en la Fig. 1.24. Mostramos los mapas de calidad para la mallas que generamos en el suavizamiento local y global en las Figuras 6.10 y 6.11, respectivamente. El suavizamiento global permite que las líneas coordenadas sigan la geometría de la región, lo que contribuye a mejorar la calidad de la malla global.

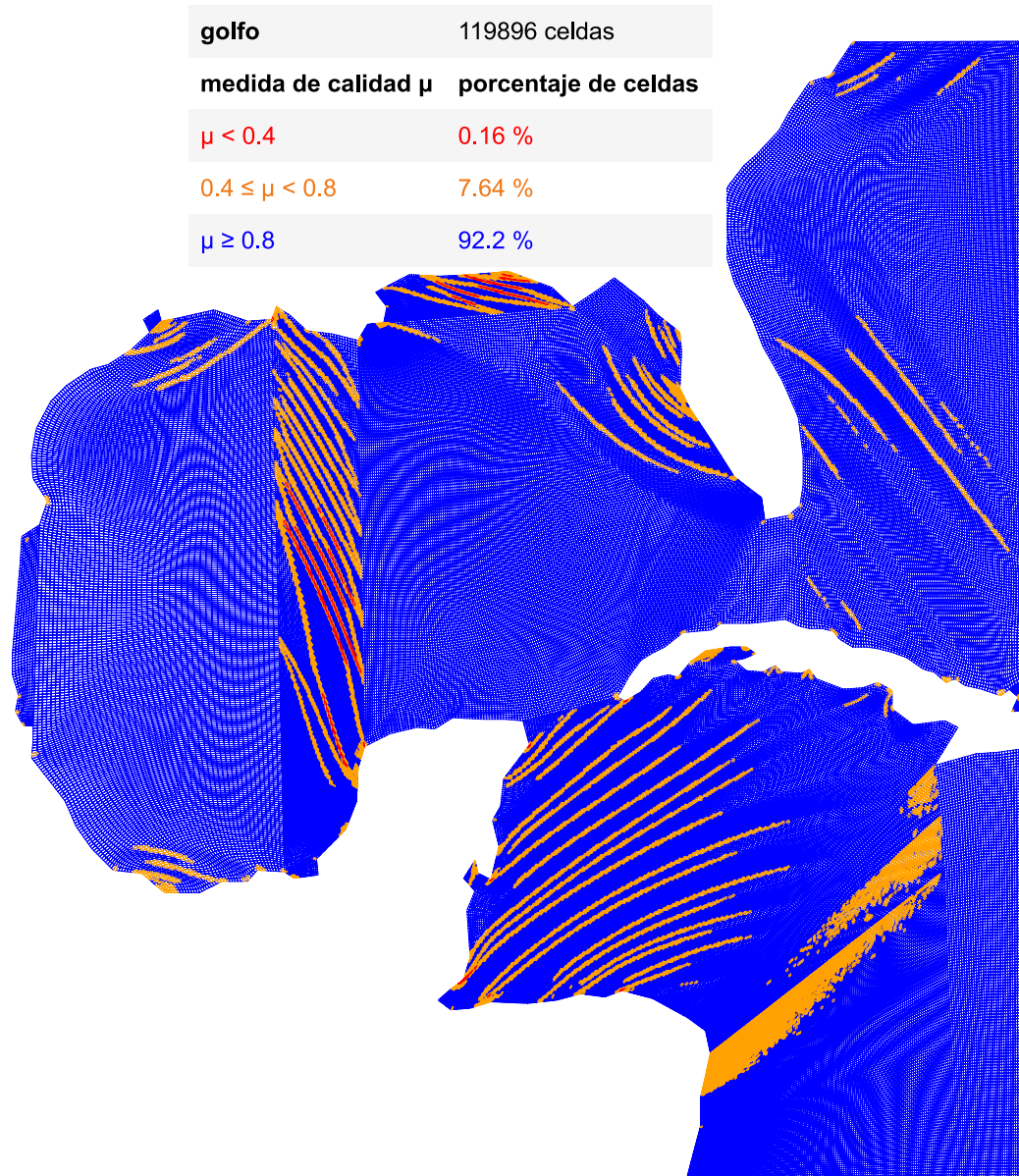


Figura 6.10: Mapa de calidad para malla del Golfo de México con suavizamiento local.

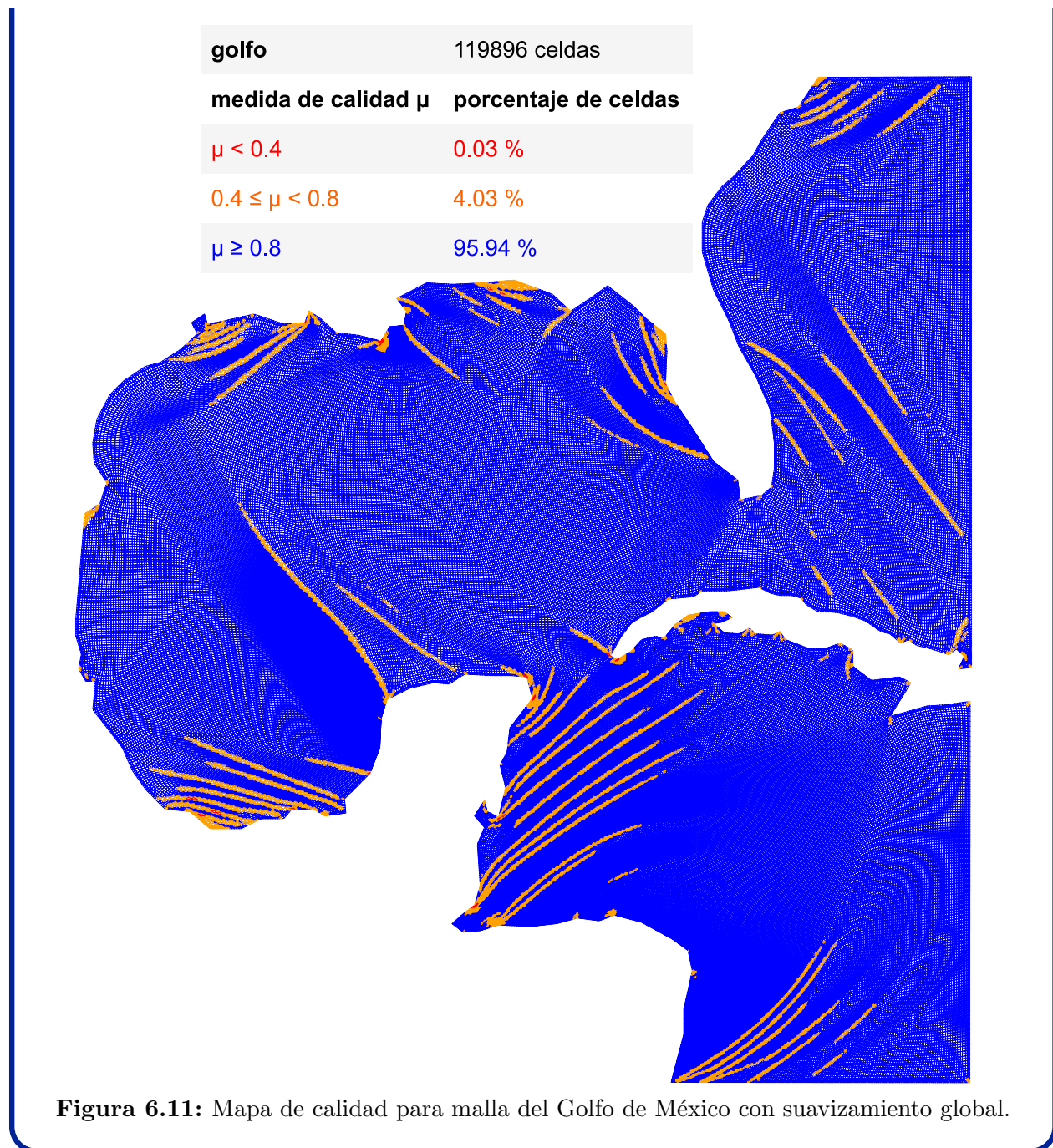


Figura 6.11: Mapa de calidad para malla del Golfo de México con suavizamiento global.

Hemos formulado el suavizamiento de mallas como la generación variacional de mallas. En la siguiente sección haremos una breve revisión de este enfoque variacional. Explicaremos el problema de optimización correspondiente así como la elección de los funcionales.

6.4. Generación Variacional de Mallas

Queremos controlar propiedades geométricas de las mallas como las longitudes y los ángulos de sus líneas, y las áreas y convexidad de sus celdas para generar mallas con la calidad deseada. Para ello empleamos el enfoque variacional [55, 113], que se originó con Brackbill y Saltzman [22] como una extensión del trabajo de Winslow [139]. La idea es encontrar funcionales adecuados que al minimizarlos se obtenga alguna propiedad geométrica deseable de las celdas. Damos una breve revisión de la generación variacional de mallas.

6.4.1. Generación Variacional Continua

Sea Ω una región poligonal y $S = [0, 1] \times [0, 1]$, y sea $\psi : \partial S \rightarrow \partial\Omega$ una parametrización entre las fronteras, podemos generar una malla suave sobre Ω con el método de Winslow [139], que consiste en encontrar un mapeo

$$\mathbf{X} : \boldsymbol{\xi} = (\xi, \eta) \in S \mapsto \mathbf{x} = (x, y) \in \Omega$$

que cumpla la *ecuación de Laplace*:

$$\begin{aligned} x_{\xi\xi} + x_{\eta\eta} &= 0, \\ y_{\xi\xi} + y_{\eta\eta} &= 0, \end{aligned}$$

sujeto a la condición de Dirichlet:

$$\mathbf{X}|_{\partial S} = \psi. \quad (6.4)$$

El mapeo \mathbf{X} se llama *mapeo armónico*. Como consecuencia del Teorema de Radó-Kneser-Choquet [38], tenemos que el *mapeo armónico* es único y suave, más aún, es inyectivo si la región Ω es convexa, y su inversa es a su vez un *mapeo armónico*.

Podemos usar el *Principio de Dirichlet* [39] para obtener el *mapeo armónico* como el mínimo del *funcional de longitud* dado por:

$$\mathcal{F}_L(\mathbf{X}) = \int_S (x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2) d\xi d\eta,$$

sobre los mapeos suaves de S a Ω sujetos a la condición de Dirichlet (6.4). No obstante, las mallas generadas con el funcional \mathcal{F}_L pueden quedar dobladas sobre regiones no convexas [8, 131], para remediar esto, Brackbill y Saltzman [22] emplean el *Principio de Dirichlet* sobre el mapeo inverso, obtienen \mathbf{X}^{-1} como el mínimo del *funcional de suavidad* dado por:

$$\mathcal{F}_S(\mathbf{X}) = \int_\Omega (\xi_{xx}^2 + \xi_{yy}^2 + \eta_{xx}^2 + \eta_{yy}^2) dx dy = \int_S \frac{x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2}{J_{\mathbf{X}}} d\xi d\eta,$$

donde $J_{\mathbf{X}}$ es el jacobiano de \mathbf{X} .

Podemos generar mallas suaves y no dobladas sobre regiones no convexas minimizando el funcional \mathcal{F}_S , más aún, estas mallas son *mapeos armónicos* [22]. Una revisión más detallada puede consultarse en Ivanenko [56].

En la generación variacional de mallas elegimos un funcional adecuado y buscamos el mapeo óptimo para ese funcional. Siguiendo a Knupp y Steinberg [62], buscamos un funcional \mathcal{F} de la forma:

$$\mathcal{F}(\mathbf{X}) = \int_S f(\mathbf{X}_\xi, \mathbf{X}_\eta, \mathbf{X}, \boldsymbol{\xi}) d\xi d\eta, \quad (6.5)$$

donde $f : \mathbb{R}^8 \rightarrow \mathbb{R}$ es una función suave y escribimos de manera compacta:

$$f(\mathbf{X}_\xi, \mathbf{X}_\eta, \mathbf{X}, \boldsymbol{\xi}) = f(x_\xi(\xi, \eta), x_\eta(\xi, \eta), y_\xi(\xi, \eta), y_\eta(\xi, \eta), x(\xi, \eta), y(\xi, \eta), \xi, \eta).$$

El funcional \mathcal{F} se define sobre el *espacio de Sobolev* $H^1(S, \Omega)$ [39], que consiste de funciones de S a Ω con primeras derivadas parciales débiles tales que la función y sus derivadas son cuadrado integrables. Este espacio tiene las propiedades algebraicas y topológicas de un *espacio de Hilbert* [39] cuando lo dotamos con un producto interior de las funciones cuadrado integrables. En la generación variacional de mallas nos interesa los mapeos en $H^1(S, \Omega)$ sujetos a la condición de Dirichlet (6.4).

Problema 6.5 (Generación Variacional de Mallas). Dado un mapeo $\psi : \partial S \rightarrow \partial\Omega$, elegir un funcional \mathcal{F} de la forma (6.5) y hallar el mapeo óptimo:

$$\mathbf{X}^* = \arg \min_{\mathbf{X}} \mathcal{F}(\mathbf{X}),$$

donde el mínimo se toma sobre los mapeos \mathbf{X} en $H^1(S, \Omega)$ sujetos a la condición de Dirichlet (6.4).

Observaciones 6.5: En la literatura, tenemos condiciones suficientes para garantizar que el funcional \mathcal{F} tenga un mínimo en $H^1(S, \Omega)$, de manera informal, es suficiente que el funcional crezca rápidamente cuando $\|\mathbf{J}_X\| \rightarrow \infty$ y que la función f sea convexa en las variables correspondientes a las derivadas del mapeo [52]. Definiciones y enunciados más precisos pueden consultarse en Evans [39].

En la Tabla 6.1 mostramos algunos funcionales clásicos que se han empleado en la literatura de generación de mallas:

- El funcional de área \mathcal{F}_A genera celdas con áreas aproximadamente iguales.
- El funcional de ortogonalidad \mathcal{F}_O genera líneas aproximadamente perpendiculares.
- El funcional de longitud \mathcal{F}_L genera líneas que tengan aproximadamente la misma longitud.
- El funcional de suavidad \mathcal{F}_S , conocido también como el funcional de Winslow, genera líneas más suaves.

Funcional	$f(\mathbf{X}_\xi, \mathbf{X}_\eta)$
Área	$J_{\mathbf{X}}^2$
Ortogonalidad	$(\mathbf{X}_\xi^T \mathbf{X}_\eta)^2$
Área Ortogonalidad	$\ \mathbf{X}_\xi\ _2^2 \ \mathbf{X}_\eta\ _2^2$
Longitud	$\ \mathbf{X}_\xi\ _2^2 + \ \mathbf{X}_\eta\ _2^2$
Suavidad	$(\ \mathbf{X}_\xi\ _2^2 + \ \mathbf{X}_\eta\ _2^2)/J_{\mathbf{X}}$

Tabla 6.1: Funcionales Clásicos en Generación de Mallas.

Podemos realizar combinaciones convexas de los funcionales. Por ejemplo:

$$\mathcal{F} = \sigma_1 \mathcal{F}_A + \sigma_2 \mathcal{F}_O + \sigma_3 \mathcal{F}_L,$$


donde

$$\sigma_1 + \sigma_2 + \sigma_3 = 1, \quad \sigma_i \geq 0, \quad i = 1, 2, 3.$$

Observaciones 6.6:

 El funcional de área ortogonalidad se puede obtener como

$$\mathcal{F}_{AO} = \frac{1}{2}(\mathcal{F}_A + \mathcal{F}_O).$$

 Los mapeos óptimos para los funcionales de área, ortogonalidad y longitud son inyectivos sobre regiones convexas, sin embargo pueden no serlo sobre regiones no convexas, las mallas correspondientes usualmente quedan dobladas [131].

En la siguiente sección formulamos el problema discreto de la generación variacional de mallas y elegimos funcionales para generar mallas convexas.

6.4.2. Generación Variacional Discreta

Ivanenko [55] desarrolló un método para generar mallas convexas mediante la minimización de funcionales. Lo que hacemos es discretizar el Problema 6.5 como un problema de optimización en varias variables. Los funcionales se aproximan por funciones de las celdas. En nuestro caso usamos este método para generar mallas estructuradas convexas con la calidad deseada.

Discretización del Problema Variacional. Sea Ω una región poligonal orientada en sentido antihorario, y sea G una malla estructurada de tamaño $M \times N$ sobre Ω . Denotamos por $\mathbf{p}_{i,j}$ a los puntos de la malla G . El número de puntos interiores de G es

$$N_I = MN - 2M - 2N + 4.$$

Las celdas de G son los cuadriláteros orientados

$$Q_{i,j} = \mathbf{p}q\mathbf{r}\mathbf{s}, \quad \begin{array}{l} i = 1, \dots, M-1, \\ j = 1, \dots, N-1. \end{array}$$

donde

$$\mathbf{p} = \mathbf{p}_{i,j}, \quad \mathbf{q} = \mathbf{p}_{i,j+1}, \quad \mathbf{r} = \mathbf{p}_{i+1,j+1}, \quad \mathbf{s} = \mathbf{p}_{i+1,j}.$$

Medimos las propiedades geométricas de cada cuadrilátero $Q_{i,j}$ mediante una función $f_{quad} : \mathbb{R}^8 \rightarrow \mathbb{R}$ que evaluamos en las coordenadas de los vértices de $Q_{i,j}$, denotamos por $f_{quad}(Q_{i,j})$ al valor correspondiente. Sumamos las contribuciones de todos los cuadriláteros de la malla G mediante la función $F : \mathbb{R}^{2N_I} \rightarrow \mathbb{R}$ tal que su valor $F(G)$ en las coordenadas de los puntos interiores de la malla G está dado por:

$$F(G) = \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} f_{quad}(Q_{i,j}). \quad (6.6)$$

A continuación mostramos que F es una aproximación del funcional \mathcal{F} dado por (6.5), para ello empleamos una rejilla uniforme U sobre S del mismo tamaño que G . Sean $\{\xi_i\}$ y $\{\eta_j\}$ particiones uniformes de M y N puntos del intervalo $[0, 1]$, respectivamente. Los puntos de la malla U son $\boldsymbol{\xi}_{i,j} = (\xi_i, \eta_j)$. Las celdas de U son los rectángulos orientados

$$R_{i,j} = \mathbf{u}\mathbf{v}\mathbf{w}\mathbf{z},$$

donde

$$\mathbf{u} = \boldsymbol{\xi}_{i,j}, \quad \mathbf{v} = \boldsymbol{\xi}_{i+1,j}, \quad \mathbf{w} = \boldsymbol{\xi}_{i+1,j+1}, \quad \mathbf{z} = \boldsymbol{\xi}_{i,j+1}.$$

Sea $\mathbf{X} \in \text{Homeo}(S, \Omega)$. Denotamos por $\mathbf{X}^{i,j}$ a la restricción del mapeo \mathbf{X} al rectángulo $R_{i,j}$. Entonces el funcional (6.5) es

$$\mathcal{F}(\mathbf{X}) = \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} \int_{R_{i,j}} f(\mathbf{X}_\xi^{i,j}, \mathbf{X}_\eta^{i,j}) d\xi.$$

Aproximamos $\mathbf{X}^{i,j}$ por el mapeo bilineal $\mathbf{B}^{i,j} : R_{i,j} \rightarrow Q_{i,j}$ tal que:

$$\mathbf{B}^{i,j}(\mathbf{u}) = \mathbf{p}, \quad \mathbf{B}^{i,j}(\mathbf{v}) = \mathbf{q}, \quad \mathbf{B}^{i,j}(\mathbf{w}) = \mathbf{r}, \quad \mathbf{B}^{i,j}(\mathbf{z}) = \mathbf{s}.$$

Por lo que

$$\mathcal{F}(\mathbf{X}) \approx \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} \int_{R_{i,j}} f(\mathbf{B}_\xi^{i,j}, \mathbf{B}_\eta^{i,j}) d\xi.$$

Luego, aproximamos la integral por el valor promedio de f en los vértices de $R_{i,j}$:

$$\mathcal{F}(\mathbf{X}) \approx \frac{1}{4} \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} \sum_{\mathbf{u} \in \text{vértices}(R_{i,j})} f(\mathbf{B}_\xi^{i,j}(\mathbf{u}), \mathbf{B}_\eta^{i,j}(\mathbf{u})).$$

Aproximamos los valores de f en cada vértice \mathbf{u} de $R_{i,j}$ mediante la función f_{quad} como:

$$f(\mathbf{B}_\xi^{i,j}(\mathbf{u}), \mathbf{B}_\eta^{i,j}(\mathbf{u})) \approx f_{quad}(Q_{i,j}).$$

Esto se debe a que las derivadas parciales $\mathbf{B}_\xi^{i,j}$ y $\mathbf{B}_\eta^{i,j}$ en los vértices de $R_{i,j}$ son los lados escalados de $Q_{i,j}$:

$$\begin{aligned} \mathbf{B}_\xi^{i,j}(\mathbf{v}) &= \mathbf{B}_\xi^{i,j}(\mathbf{u}) = (M-1)(\mathbf{q} - \mathbf{p}), \\ \mathbf{B}_\xi^{i,j}(\mathbf{w}) &= \mathbf{B}_\xi^{i,j}(\mathbf{z}) = (M-1)(\mathbf{r} - \mathbf{s}), \\ \mathbf{B}_\eta^{i,j}(\mathbf{v}) &= \mathbf{B}_\eta^{i,j}(\mathbf{z}) = (N-1)(\mathbf{s} - \mathbf{p}), \\ \mathbf{B}_\eta^{i,j}(\mathbf{u}) &= \mathbf{B}_\eta^{i,j}(\mathbf{w}) = (N-1)(\mathbf{r} - \mathbf{q}). \end{aligned}$$

Por lo tanto,

$$\mathcal{F}(\mathbf{X}) \approx \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} f_{quad}(Q_{i,j}) = F(G).$$

Con base a lo anterior, el Problema 6.5 de construir el mapeo óptimo para el funcional \mathcal{F} se discretiza como el problema de generar la malla óptima para la función F .

Una vez que generamos la malla óptima, podemos construir mapeos bilineales de la forma (2.5) usando la malla óptima como malla de control. Una condición suficiente para que estos mapeos sean inyectivos es que la malla óptima sea convexa [1]. Así que formulamos el problema discreto como sigue:

Problema 6.6. Sea Ω una región poligonal. Sea $M(\Omega)$ el conjunto de mallas estructuradas sobre Ω con los mismos puntos frontera. Elegir un funcional F de la forma (6.6) y hallar la malla óptima:

$$G^* = \arg \min_{G \in M(\Omega)} F(G),$$

de modo que G^* sea convexa.

Observaciones 6.7: Mapeos suaves de S a Ω pueden construirse generando mallas de alto orden a partir de mallas estructuradas de cuadriláteros. Al respecto, Abelló et al. [2] han construido mapeos bicuadráticos a partir de mallas estructuradas.

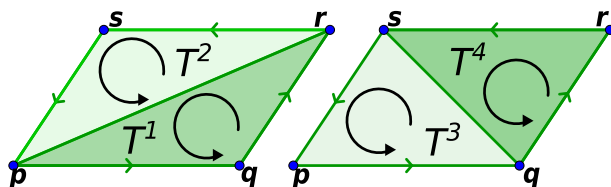


Figura 6.12: Triángulos T_k , $k = 1, 2, 3, 4$ asociados al cuadrilátero orientado $pqrs$.

Condiciones suficientes para generar mallas convexas. La idea para identificar cuándo una malla es convexa es orientar cada celda respetando la orientación de la frontera. Asociamos triángulos a cada celda orientada y examinamos sus áreas con signo.

Orientamos la frontera de la malla G en sentido antihorario. Separamos cada cuadrilátero orientado $Q = pqrs$ de la malla en cuatro triángulos orientados:

$$T_1 = pqr, \quad T_2 = rsp, \quad T_3 = spq, \quad T_4 = qrs.$$

En la Fig. 6.12 mostramos estos cuatro triángulos. Sea $\{T_q\}$ el conjunto de triángulos de la malla G . El número total de triángulos para una malla estructurada de cuadriláteros de tamaño $M \times N$ es

$$N_T = 4(M - 1)(N - 1).$$

El área con signo de un triángulo orientado $T = pqr$ es

$$\alpha(T) = (\mathbf{q} - \mathbf{p})^t J_2 (\mathbf{r} - \mathbf{p}), \quad J_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

Si T está orientado en sentido antihorario, $\alpha(T) \geq 0$. De otro modo, $\alpha(T) \leq 0$.

Sea α_q el área con signo del triángulo T_q de G . Asociamos dos medidas a la malla G :

$$\begin{aligned} \alpha_-(G) &= \min_q \alpha_q. \\ \bar{\alpha}(G) &= \frac{1}{N_T} \sum_q \alpha_q. \end{aligned}$$

Una condición suficiente para generar mallas convexas de cuadriláteros es que todos los triángulos asociados a las celdas tengan área con signo positiva.

Proposición 6.1 ([10]). *Sea G una malla de cuadriláteros sobre una región Ω orientada en sentido antihorario. La malla G es convexa si $\alpha_-(G) > 0$.*

Mallas ϵ -convexas. La generación variacional discreta de una malla convexa puede estar mal condicionada en el sentido de que la convexidad de la mallas óptima es sensible a valores pequeños del área mínima α_- . Más aún, α_- depende de la escala de la región. Para remediar esto Barrera et al. [11] introducen el siguiente concepto: Dado $\epsilon > 0$, una malla G es ϵ -convexa si

$$\alpha_-(G) > \epsilon \cdot \bar{\alpha}(G).$$

La clave es que el área promedio $\bar{\alpha}(G)$ es el área de la región Ω por el número de celdas.

Reformulación del Problema 6.6. Formulamos la generación variacional discreta de mallas usando los cuatro triángulos asociados a cada cuadrilátero, para ello usamos una función $f_{tri} : \mathbb{R}^6 \rightarrow \mathbb{R}$, y denotamos por $f_{tri}(T_q)$ a su evaluación en las coordenadas de los vértices del triángulo T_q . La función f_{tri} mide propiedades geométricas de los triángulos como las que mostramos en la Tabla 6.2.

La función f_{quad} evaluada en las coordenadas de los vértices del cuadrilátero Q puede obtenerse como:

$$f_{quad}(Q) = f_{tri}(T_1) + f_{tri}(T_2) + f_{tri}(T_3) + f_{tri}(T_4).$$

Así, la funcional discreta (6.6) se puede expresar en términos de los N_T triángulos T_q de la malla G como:

$$F(G) = \sum_{q=1}^{N_T} f_{tri}(T_q). \quad (6.7)$$

De este modo el Problema 6.6 se formula como sigue:

Problema 6.7 (Generación Variacional Discreta de Mallas). Sea Ω una región poligonal orientada en sentido antihorario. Elegir un funcional F de la forma (6.7) y hallar la malla óptima:

$$G^* = \arg \min_{G \in M(\Omega)} F(G),$$

de modo que G^* sea ϵ -convexa.

Resolvemos el Problema 6.7 para mejorar la calidad de nuestras mallas en las dos etapas del suavizamiento:

- El Problema 6.2 del suavizamiento local se formula como el Problema 6.7, para ello reemplazamos la región Ω por el bloque Ω_k y el conjunto $M(\Omega)$ por la colección de mallas de cuadriláteros con misma conectividad y puntos frontera que la malla estructurada sobre Ω_k generada por TFI.
- El Problema 6.3 del suavizamiento global se puede formular como el Problema 6.7, para ello en cada paso reemplazamos Ω por la unión de los bloques contiguos y el conjunto $M(\Omega)$ por la colección de mallas con la misma conectividad y puntos frontera que la unión de las mallas correspondientes generadas por TFI.

Elección de los Funcionales discretos. En la Tabla 6.2 mostramos algunos funcionales para generación variacional discreta de mallas. Estos funcionales pueden pensarse como la discretización de los correspondientes funcionales continuos, no obstante tienen dificultades para generar mallas óptimas convexas sobre regiones complicadas.

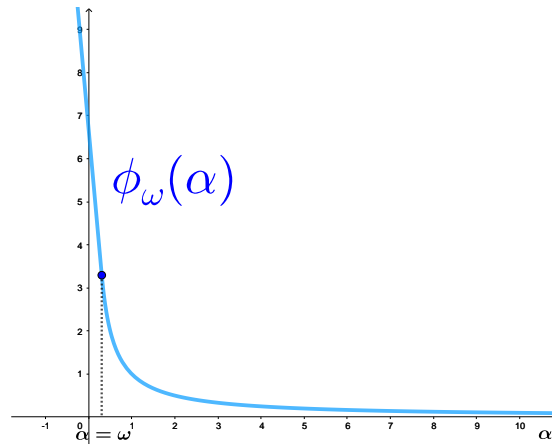


Figura 6.13: Función ϕ_ω para regularización del polo $1/\alpha$

Funcional	f_{tri} en el triángulo $T = pqr$
Área F_A	$\alpha^2(T)$, $\alpha(T) = (\mathbf{r} - \mathbf{q})^T J_2(\mathbf{p} - \mathbf{q})$
Ortogonalidad F_O	$o^2(T)$, $o(T) = (\mathbf{p} - \mathbf{q})^T (\mathbf{r} - \mathbf{q})$
Área Ortogonalidad F_{AO}	$\frac{o^2(T) + \alpha^2(T)}{2} = \frac{1}{2} \ \mathbf{r} - \mathbf{p}\ _2^2 \ \mathbf{q} - \mathbf{p}\ _2^2$
Longitud F_L	$\lambda(T) = \ \mathbf{p} - \mathbf{q}\ _2^2 + \ \mathbf{r} - \mathbf{q}\ _2^2$
Suavidad F_S	$\frac{\lambda(T)}{\alpha(T)}$

Tabla 6.2: Funcionales discretos clásicos para generación de mallas.

La dificultad del Problema 6.7 es asegurar que la malla óptima sea convexa. Una manera para generar y preservar celdas convexas es mediante una *regularización* de los funcionales de área recíproca y suavidad, lo que hacemos es reemplazar el polo $\frac{1}{\alpha}$ por funciones suaves convexas no decrecientes, por ejemplo:

$$\phi(\alpha) = \begin{cases} (\alpha - 1)(\alpha - 2) + 1, & \text{si } \alpha < 1; \\ \frac{1}{\alpha}, & \text{si } \alpha \geq 1, \end{cases}$$

$$\phi_\omega(\alpha) = \begin{cases} \frac{2(\omega - \alpha)}{\omega^2}, & \text{si } \alpha < \omega; \\ \frac{1}{\alpha}, & \text{si } \alpha \geq \omega. \end{cases}$$

donde $\omega > 0$ es un *parámetro de regularización*, Fig. 6.13.

Barrera et al. [11, 10] prueban que si la malla inicial es convexa, los funcionales $S_{\omega, \epsilon}$ y H_ω generan mallas óptimas convexas con la elección adecuada de parámetros ϵ y ω , donde:

$$S_{\omega, \epsilon}(G) = \sum_q \phi(\omega \cdot (\alpha_q - \epsilon)),$$

$$H_\omega(G) = \sum_q \lambda(T_q) \phi_\omega(\alpha_q).$$

Los funcionales $S_{\omega, \epsilon}$ y H_ω generan y preservan celdas convexas, sin embargo no contribuyen a la calidad de la malla, para compensar esto pueden combinarse con otros funcionales. Se requiere del funcional de ortogonalidad para generar celdas que sean aproximadamente rectángulos, y por otra parte se ocupa el funcional de área para que tengamos celdas de áreas aproximadamente iguales, así que para generar mallas ϵ -convexas de alta calidad usamos la siguiente combinación convexa de funcionales:

$$F(G) = (1 - \sigma) \cdot F_{AO}(G) + \sigma \cdot H_\omega(G), \quad 0 < \sigma < 1. \quad (6.8)$$

Observaciones 6.8: Los funcionales que combinamos deben tener el mismo orden de magnitud para que la geometría de las celdas cambie como se desea, así que los escalamos por una *constante de normalización* K , que depende del valor del funcional en la malla óptima con celdas ideales [131]:

- en el funcional de longitud F_L tenemos que las celdas de la malla óptima debe ser rectangulares y $K = 8 \cdot \text{área}(\Omega)$,
- en el funcional de área F_A tenemos que las celdas de la malla óptima deben tener la misma área y $K = \text{área}^2(\Omega)/N_T$,
- el funcional de ortogonalidad F_O tiene el mismo orden de magnitud que F_A , por lo que usamos la misma *constante de normalización*.

Adaptabilidad geométrica. Podemos acumular las celdas de la malla alrededor de un objeto geométrico contenido en la región Ω mediante una función no negativa, conocida como *función de densidad* [46, 52], que desempeña el papel de una función de pesos para el funcional de la generación variacional.

En nuestro caso empleamos una *función de densidad* w que asigna un entero menor que 10 a las celdas alrededor de la frontera de Ω y uno al resto de las celdas para generar adaptabilidad a la frontera. Generamos mallas de alta calidad con celdas que se acumulan alrededor de $\partial\Omega$ mediante el funcional ponderado:

$$F(G) = (1 - \sigma) \cdot w \cdot F_{AO}(G) + \sigma \cdot H_\omega(G), \quad (6.9)$$

como mostramos en el Ejemplo 6.3.

González [46] emplea otras *funciones de densidad* y proporciona más detalles sobre la generación de mallas con adaptabilidad geométrica.

Ejemplo 6.3. Generamos una malla de cuadriláteros sin singularidades interiores sobre la Laguna Honda mediante la *descomposición admisible* del Ejemplo 4.3, para ello generamos mallas estructuradas en cada bloque por TFI y resolvemos el problema de programación entera 5.10 para determinar sus tamaños compatibles, luego las unimos en una malla global, que suavizamos minimizando el funcional

$$F = 0.9w \cdot F_{AO} + 0.1H_w.$$

A diferencia de la malla de la Fig. 4.20, acumulamos las celdas alrededor de la frontera como se muestra en el mapa de calidad de la Fig. 6.14.

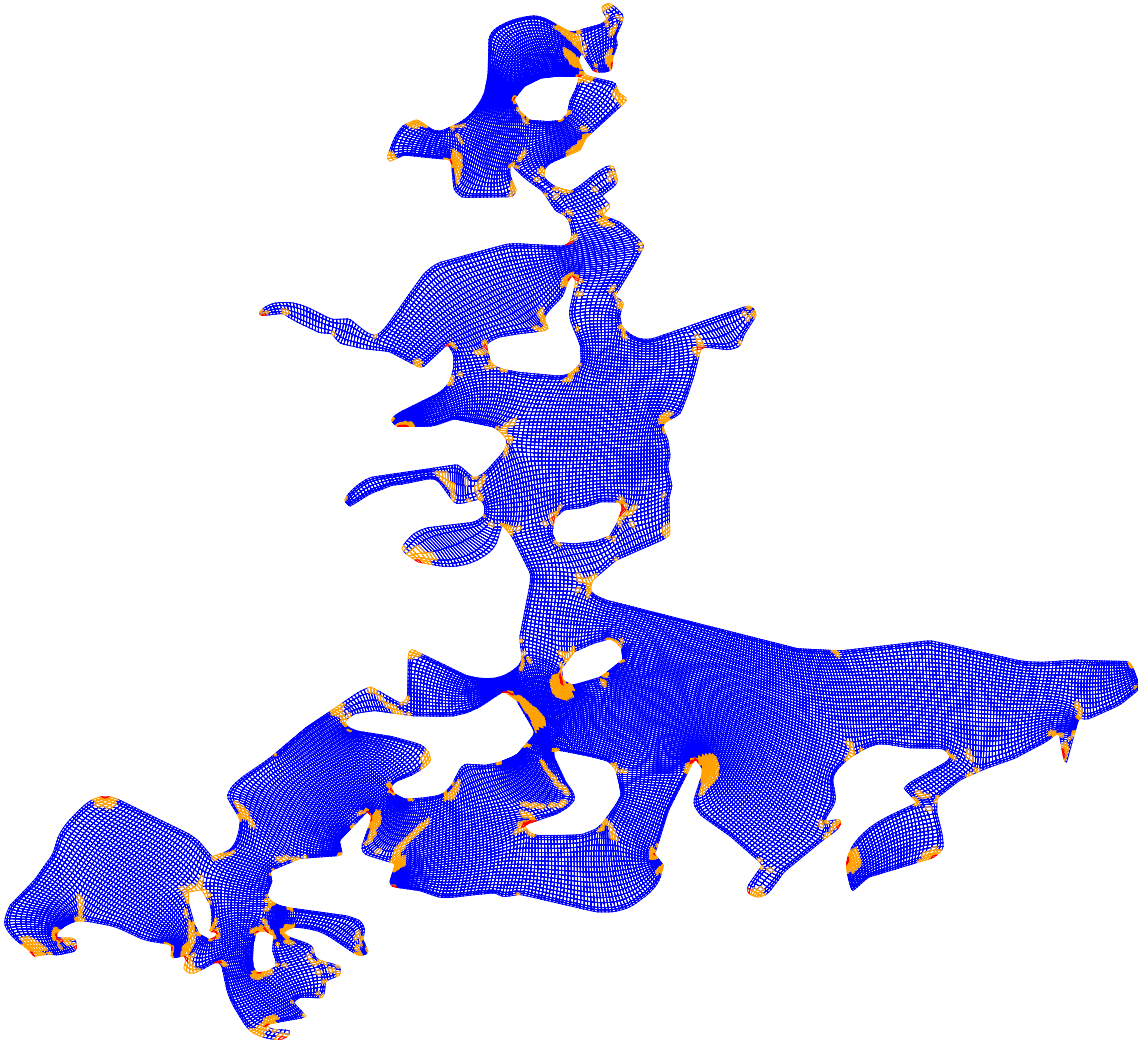


Figura 6.14: Mapa de calidad para malla sin singularidades interiores de la Laguna Honda con adaptabilidad a la frontera.

Optimización. El Problema 6.7 es una optimización de varias variables, donde las variables son las coordenadas de los puntos interiores, mientras que los puntos frontera están fijos. Para resolver este problema empleamos el *método de Newton con región de confianza* implementado en el optimizador TRON [70], que requiere los gradientes de los funcionales así como el patrón de esparcidad de las matrices hessianas correspondientes.

Castellanos [93] proporciona expresiones analíticas para los gradientes de los funcionales y detalles sobre su implementación. Por otra parte, el patrón de esparcidad de las matrices hessianas de los funcionales se puede obtener a partir de la conectividad de la malla.

6.5. Resumen del Capítulo

Construimos *parametrizaciones admisibles y compatibles* \mathbf{X}_k del cuadrado unitario S a las subregiones Ω_k de una *descomposición admisible* de una región Ω mediante la generación de mallas estructuradas por bloques de alta calidad con puntos frontera dados por parametrizaciones compatibles $\psi_k : \partial S \rightarrow \partial\Omega_k$, para generar estas mallas realizamos los siguientes pasos:

1. Extendemos las parametrizaciones ψ_k por interpolación transfinita (TFI) para generar mallas estructuradas G_k sobre las subregiones Ω_k .
2. Unimos recursivamente las mallas G_k en una malla global G_Ω enumerando los puntos frontera en orden secuencial, en cada paso unimos una malla por bloques con una malla de un solo bloque y actualizamos su numeración, las unimos en el orden que construimos un *árbol* generador para la gráfica de la *descomposición admisible*. Como separamos la región Ω por cortes, la malla G_Ω no tiene singularidades interiores.
3. Suavizamos la malla global G_Ω mediante el enfoque variacional discreto para mejorar su calidad, el suavizamiento se realiza en dos etapas:

Suavizamiento local. Para cada bloque G_k generamos la malla óptima G_k^* del funcional (6.8) sobre el conjunto de mallas con los mismos puntos frontera y conectividad.

Suavizamiento global. De manera secuencial suavizamos grupos de bloques contiguos en el orden en que los unimos hasta que cubramos toda la región, esto tiene la ventaja de reducir el costo computacional de la optimización. En el k -ésimo paso generamos la malla óptima del funcional (6.8) sobre el conjunto de mallas con mismos puntos frontera y conectividad que la unión de G_k^* con sus mallas vecinas.

El funcional (6.8) puede generar mallas óptimas convexas de alta calidad sobre los bloques, así que los *productos tensoriales B-spline bilineales* (2.5) con estas mallas de control son mapeos inyectivos y *admisibles*, y por consiguiente son las parametrizaciones

deseadas de las subregiones. La minimización de este funcional es una optimización de varias variables sin restricciones que puede ser de gran escala, obtenemos la malla óptima por un *método de Newton con región de confianza* [70].

En nuestro proyecto generamos mallas estructuradas de cuadriláteros con alta calidad sobre regiones poligonales complicadas. La clave es separar regiones por cortes en *subregiones admisibles*. Basándonos en esta idea y en trabajos previos [11, 10] desarrollamos un método secuencial para generar las mallas deseadas.

En nuestro método resolvemos problemas de aproximación de contornos, descomposición de regiones, programación entera y optimización de gran escala sin restricciones. Además mostramos que es robusto, pues generamos mallas de alta calidad sobre regiones con diferentes grados de dificultad.

Resaltamos que la contribución principal del proyecto es la implementación de la metodología en la nueva versión de nuestro generador de mallas, UNAMalla 6. Los usuarios interesados puedan generar mallas de alta calidad de manera interactiva con este software.

Con base a las pruebas realizadas y las mallas generadas obtuvimos las siguientes conclusiones:

- * Reducimos el problema de generar mallas estructuradas por bloques de alta calidad a la elección de cortes y esquinas.
- * A diferencia de otros métodos, podemos generar mallas sin singularidades interiores de alta calidad sobre regiones complicadas con la limitación de que algunas celdas de baja calidad en regiones con ángulos agudos críticos.
- * UNAMalla 6 es una de las pocas herramientas gratuitas y de código abierto que le permite a los usuarios generar mallas de cuadriláteros sin singularidades interiores.

Como trabajo futuro planeamos realizar las siguientes tareas:

- * Encontrar condiciones suficientes para la elección de cortes y esquinas de modo que nuestras mallas estructuradas por bloques sean parametrizaciones de *regiones rectilíneas* a regiones poligonales.
- * Implementar un algoritmo para descomposición por cortes que pueda generar *descomposiciones admisibles* para una colección de regiones de prueba.
- * Extender nuestra metodología para generar mallas de cuadriláteros sobre regiones con agujeros dados por curvas poligonales.
- * Resolver algunas ecuaciones diferenciales parciales sobre regiones complicadas usando nuestras mallas.

UNAMalla es un software desarrollado por nuestro grupo de trabajo [138] para la generación interactiva de mallas estructuradas de cuadriláteros sobre regiones planas. En este proyecto desarrollamos una nueva versión, UNAMalla 6, que es la implementación de nuestra metodología en el lenguaje de programación Julia [16]. En comparación con versiones previas de UNAMalla y otros generadores de mallas como HOHQMesh [64], Gmsh [109] y CCHE-MESH [151], el usuario puede generar mallas de alta calidad sin singularidades interiores sobre regiones complicadas como se muestran en las Figuras 1.2, 6.11 y 6.14.

UNAMalla 6 consiste en una colección de módulos y rutinas escritos en Julia que se ejecutan secuencialmente en una libreta de Jupyter [61] para generar mallas estructuradas por bloques con los gráficos interactivos de Makie [29] y PlotlyJS [79] y los optimizadores Cbc [40], JuMP [37], TRON [70] y Optim [89] como se muestra en la Fig. A.1.

La generación de mallas se lleva a cabo mediante los siguientes módulos:

- * **Contour Creator.** El usuario crea sus polígonos de manera interactiva, e incluso puede agregar agujeros, Fig. A.2.
- * **Edit Boundary.** Tratamiento de la frontera de regiones complicadas mediante los métodos de la subsección 3.3.4 y la edición interactiva de polígonos, Fig. A.3.
- * **Region Decomposition.** Descomposición recursiva e interactiva de regiones por cortes, donde en cada paso se elige una subregión y puntos de corte para separarla en dos nuevas subregiones, además se pueden cargar y modificar descomposiciones por cortes, Fig. A.4.
- * **Mesh Compatibility.** En cada subregión de una descomposición por cortes se generan mallas estructuradas de modo que sus celdas empaten lado a lado en los cortes como sigue:
 - Selección automática de esquinas con base a la heurística de la subsección 5.1.1, y que el usuario puede cambiar interactivamente, Fig. A.5.
 - Solución del problema de programación entera 5.10 mediante los optimizadores JuMP y Cbc para obtener tamaños compatibles de las mallas con un factor de densidad dado por el usuario.
 - Interpolación transfinita para extender las fronteras compatibles de las subregiones como mallas estructuradas.
- * **Mesh Merging.** Las mallas sobre las subregiones se unen en una malla global con sus puntos frontera enumerados secuencialmente como se describe en la sección 6.2. La unión de las mallas es automática y recursiva, y el orden en que unen se puede cambiar interactivamente, Fig. A.6.

- * **Mesh Smoothing.** Suavizamiento de la malla global mediante el enfoque variacional discreto en dos etapas, primero en el suavizamiento local se minimiza el funcional (6.8) sobre las mallas de cada subregión, luego en el suavizamiento global se recorren los bloques en el orden que los unimos y se minimiza el mismo funcional, pero sobre mallas de bloques vecinos, Fig. A.7. En ambas etapas, las mallas óptimas se generan con el optimizador TRON.

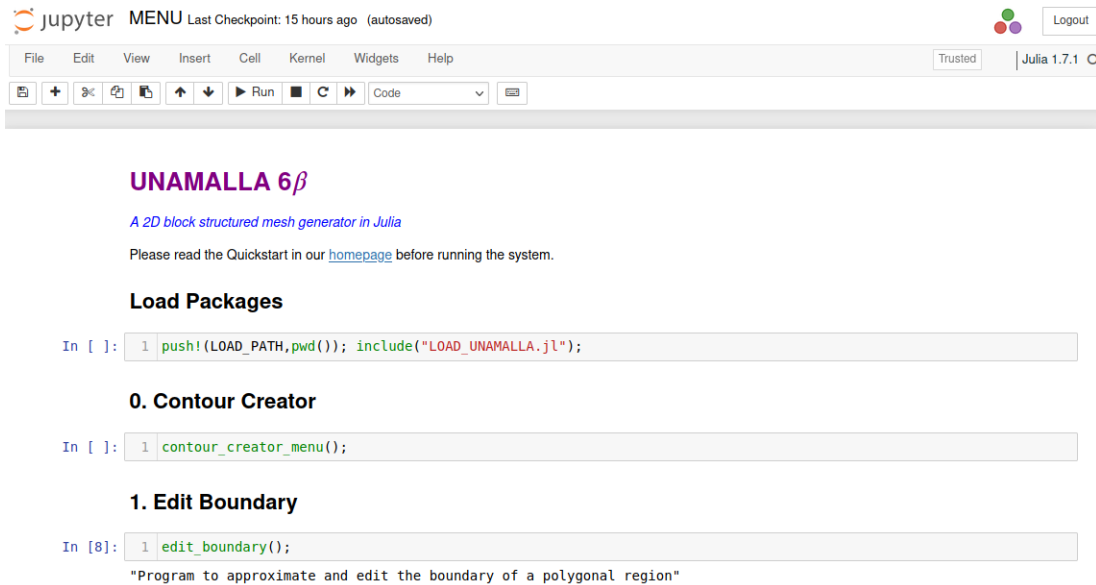


Figura A.1: Libreta de Jupyter corriendo el sistema UNAMalla 6.

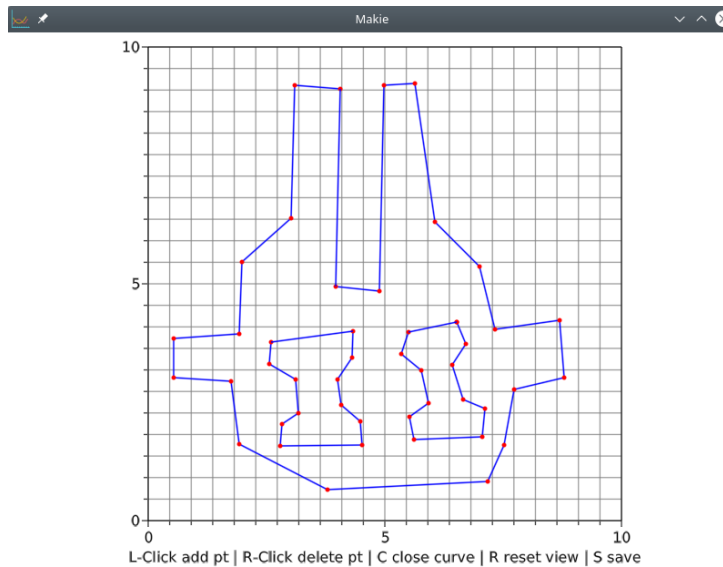


Figura A.2: Ventana interactiva del módulo **Contour Creator**. El usuario crea un polígono mediante clicks del ratón.

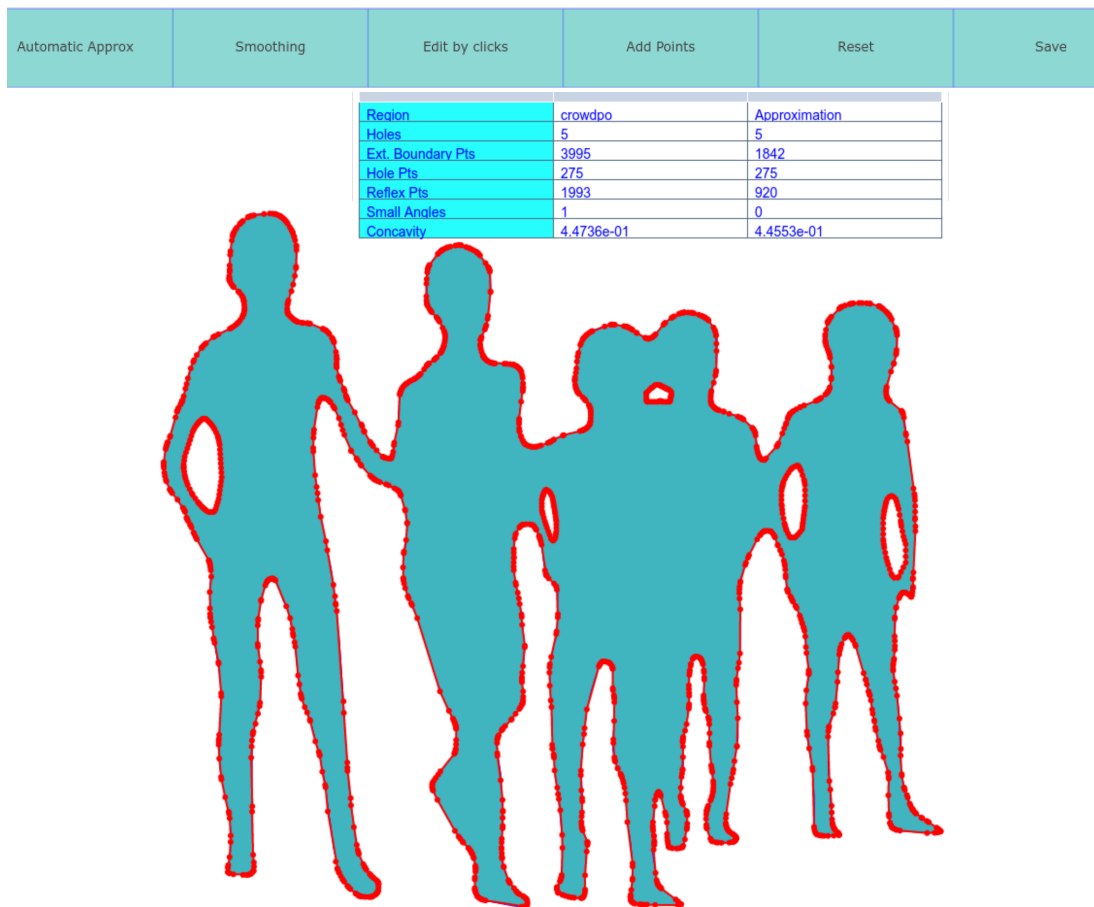


Figura A.3: Menú del módulo **Edit Boundary**. Comparamos la región contra la aproximación de su frontera en color rojo.

Códigos, ejemplos y una guía rápida de UNAMalla 6 pueden encontrarse en el sitio web:

<http://tikhonov.fciencias.unam.mx/unamalla/>

Observaciones A.1:

- ☞ La visualización y optimización de mallas tiene limitaciones, por lo que se recomienda eliminar agujeros despreciables y cambiar algunos cortes y esquinas para reducir los tamaños de las mallas.
- ☞ El módulo **Region Decomposition** tiene la opción de llamar al programa externo DUDE2D [71] para generar automáticamente una descomposición por cortes, no obstante las subregiones obtenidas pueden no ser adecuadas para generar mallas de alta calidad, en ese caso se recomienda modificar la descomposición.
- ☞ La unión automática de mallas puede fallar en algunas regiones con agujeros, esto usualmente puede remediarse cambiando el orden en que las unimos.
- ☞ Las regiones y mallas generadas con UNAMalla 6 son compatibles con el generador de mallas Gmsh, pero no viceversa.



Figura A.4: Menú del módulo **Region Decomposition**. Separamos recursivamente una región por cortes con el botón *Show/Split Region*. Si queremos deshacer el corte, presionamos el botón *Merge Regions*.

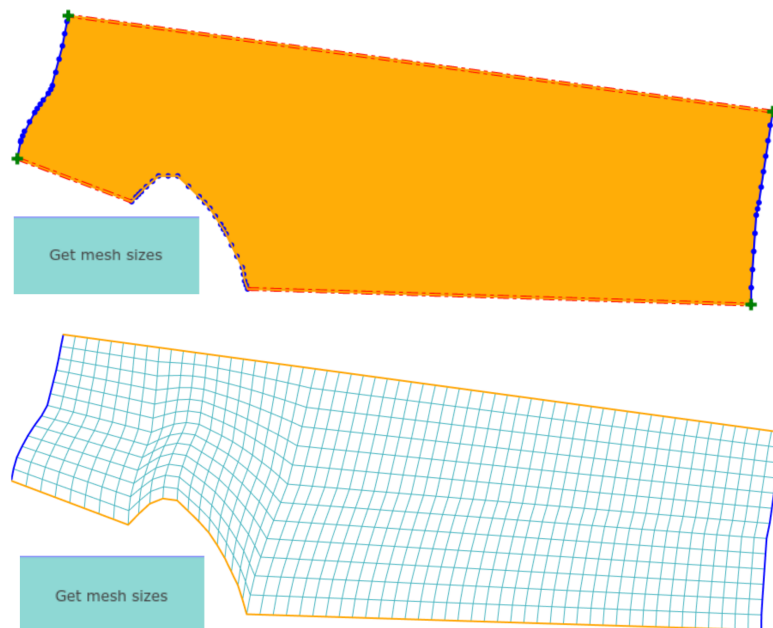


Figura A.5: Interfaz del módulo **Mesh Compatibility**. En cada bloque seleccionamos cuatro esquinas, indicadas por cruces verdes, y luego presionamos el botón *Get mesh sizes* para obtener los tamaños compatibles y generar las mallas correspondientes.

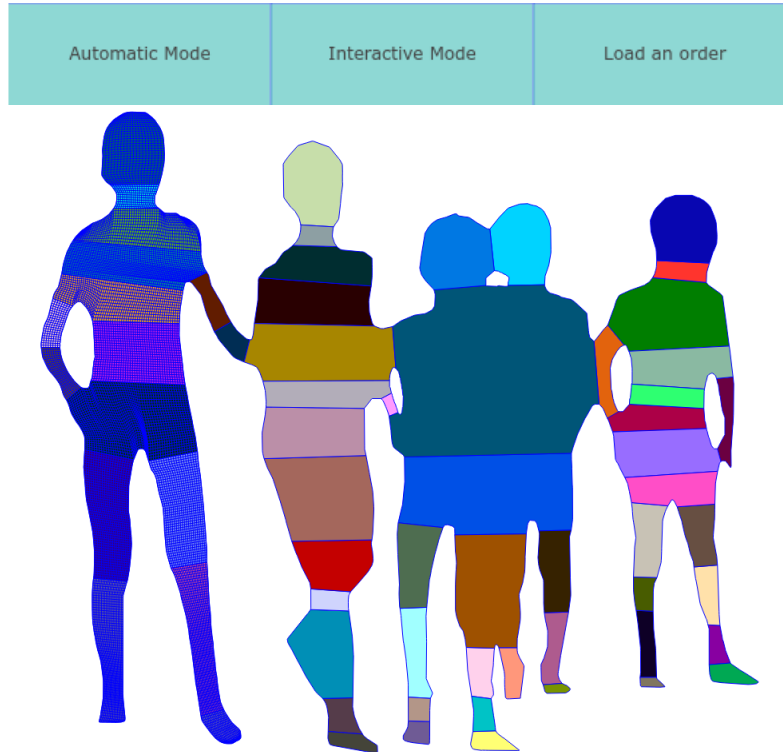


Figura A.6: Menú del módulo **Mesh Merging**. Las mallas se pueden unir de manera automática o interactiva dando clicks sobre bloques consecutivos.

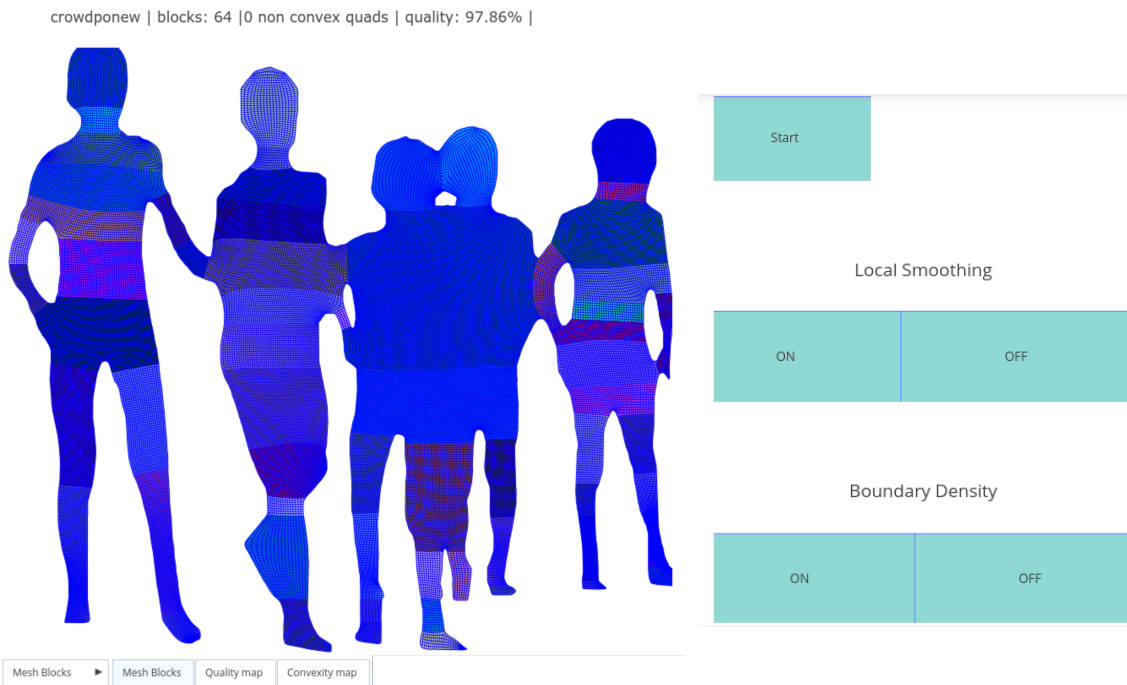


Figura A.7: Menú del módulo **Mesh Smoothing**. Al presionar el botón *Start* suavizamos la malla sobre cada bloque, y luego sobre la región completa, además podemos acumular las celdas alrededor de la frontera en el submenú *Boundary Density*.

Parametrizaciones definidas sobre discos o cuadrados pueden no ser adecuadas para representar regiones complicadas con la calidad deseada, por lo que necesitamos encontrar un dominio adecuado. El dominio de parametrizaciones asociadas a mallas de alta calidad se puede ver como una región simplificada, en las pruebas que realizamos hemos notado que su frontera puede ser la unión de segmentos de recta horizontales y verticales, a saber, una *región rectilínea*. Queremos construir mapeos de *regiones rectilíneas* a regiones poligonales complicadas que respeten la frontera y transformen mallas de rectángulos en mallas de cuadriláteros con alta calidad, Fig. B.1.

Problema B.1. Sea Ω una región poligonal acotada. Hallar una *región rectilínea* R homeomorfa a Ω y un homomorfismo $\mathbf{X} : R \rightarrow \Omega$ tal que $\mathbf{X}(\partial R) = \partial\Omega$, y si G_R es una malla de rectángulos sobre la región R , entonces su imagen $\mathbf{X}(G_R)$ es una malla de cuadriláteros de alta calidad sobre la región Ω .

Liu et al. [72] desarrollaron un método para construir mapeos de regiones rectilíneas a poligonales como mallas de alta calidad sin singularidades interiores a partir de una de una teselación de Voronoi de la región poligonal, no obstante su descomposición de regiones es más complicada que nuestra descomposición por cortes, y las regiones de sus ejemplos son sencillas.

Deseamos hallar la *región rectilínea* y el homomorfismo del Problema B a partir de la descomposición de regiones y las parametrizaciones que construimos con nuestro método, nuestra idea es encontrar una *descomposición admisible* de la región Ω que pueda dibujarse como una descomposición en rectángulos. Una manera es simplificar las subregiones eliminando los puntos que no sean cortes o *esquinas*, y luego deformarlas en rectángulos preservando las *esquinas*. De esta manera, el Problema B se puede reducir a la elección de cortes y *esquinas*.

No cualquier elección de cortes y esquinas es adecuada, pues algunas descomposiciones por cortes de la región poligonal no pueden dibujarse como descomposiciones en rectángulos que preservan las *esquinas*. Requerimos condiciones suficientes sobre los cortes y *esquinas* para obtener la descomposición deseada.

En el contexto de la teoría de gráficas, Rahman et al. [102] proporcionan condiciones suficientes para hallar un dibujo rectilíneo de una *gráfica plana*, que enunciamos como sigue:

Teorema B.1. *Sea G una gráfica plana biconexa con nodos de grado máximo a lo más tres y que tiene al menos cuatro nodos de grado dos sobre su cara exterior. Entonces G se pueden dibujar usando solamente segmentos horizontales y verticales.*

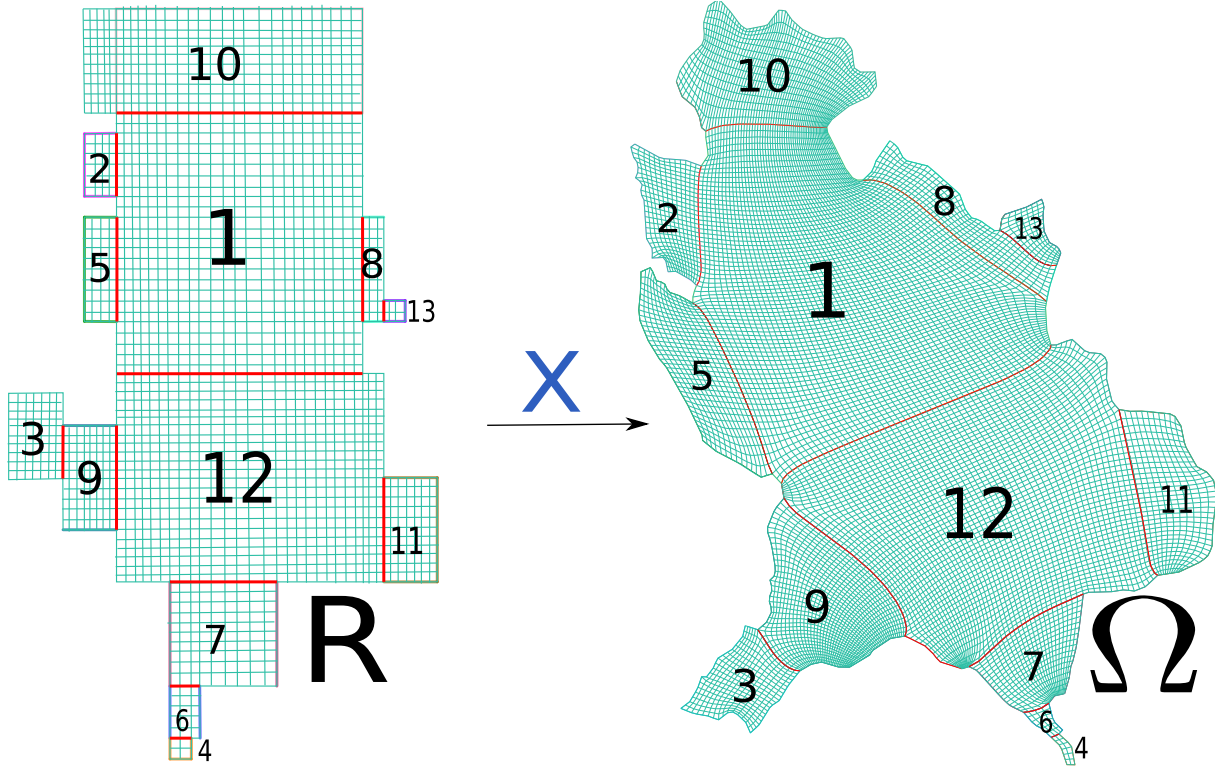


Figura B.1: Parametrización \mathbf{X} que transforma una malla estructurada de 13 bloques sobre una *región rectilínea* R en una malla con la misma conectividad sobre el Estado de Nayarit Ω .

La gráfica con nodos dados por los puntos de corte y esquinas que se unen con base a una *descomposición admisible* cumple con las hipótesis del Teorema B.1 por construcción, por lo que esta descomposición tiene un dibujo rectilíneo. En un trabajo futuro planeamos encontrar condiciones sobre los cortes y esquinas de modo que las subregiones se puedan dibujar como rectángulos que formen una *región rectilínea*.

En nuestro proyecto construimos una *descomposición admisible* de la región poligonal Ω en n subregiones Ω_k y una familia de *parametrizaciones admisibles y compatibles* \mathbf{X}_k del cuadrado unitario S a las subregiones Ω_k . Una vez que encontramos rectángulos R_k homeomorfos a las subregiones Ω_k de modo que su unión sea una *región rectilínea* R , podemos extender las parametrizaciones \mathbf{X}_k como el mapeo deseado, lo que hacemos es transformar el cuadrado unitario S a los rectángulos R_k mediante mapeos bilineales $\phi_k : R_k \rightarrow S$ y construimos el mapeo $\mathbf{X} : R \rightarrow \Omega$ como la función por tramos dada por

$$\mathbf{X}|_{R_k} = \mathbf{X}_k \circ \phi_k, \quad k = 1, \dots, n. \quad (\text{B.1})$$

La función (B.1) está bien definida debido a que los mapeos \mathbf{X}_k son compatibles, más aún, se puede probar que es un homeomorfismo de R a Ω , a saber:

Teorema B.2 ([19]). *Sean R y Ω dos regiones acotadas y homeomorfas de \mathbb{R}^2 tales que R tiene una descomposición finita $\{R_k\}$ en subregiones convexas. Sea $\mathbf{X} \in C^1(R, \Omega)$ tal que*

∂R es homeomorfo a $\partial\Omega$ bajo la función \mathbf{X} ,
la frontera de cada subregión R_k es homeomorfa a su imagen bajo la función \mathbf{X} ,
el jacobiano de \mathbf{X} es positivo en cada punto de cada subregión R_k .

Entonces $\mathbf{X} \in \text{Homeo}(R, \Omega)$.

Observaciones B.1:

- ☞ La malla G_R no tiene singularidades interiores, y por continuidad la malla $\mathbf{X}(G_R)$ tiene la misma conectividad.
- ☞ La función por tramos (B.1) satisface las condiciones del Teorema B.2, ya que transforma una descomposición en rectángulos de la *región rectilínea* R en una *descomposición admisible* de la región Ω mediante *parametrizaciones compatibles*.

BIBLIOGRAFÍA

- [1] Abelló, I. A., Hernández, V., Barrera, P., & González, G. F. (2017) Parametrización B-spline de regiones planas con frontera irregular. *Ciencias Matemáticas* 31(2), 95–107.
- [2] Abelló, I. A., Hernández, V., Barrera, P. & González, G. F. (2018) Injectivity of B-spline biquadratic maps. *Comput. Methods Appl. Mech. Engrg.* 341, 586–608.
- [3] Ai, T., Li, Z., & Liu, Y. (2005) Progressive Transmission of Vector Data Based on Changes Accumulation Model. In: *Developments in Spatial Data Handling*, Springer, 85-96. https://doi.org/10.1007/3-540-26772-7_7
- [4] Aichholzer, O., Aurenhammer, F., Alberts, D., & Gärtner, B. (1995) A Novel Type of Skeleton for Polygon. *Journal of Universal Computer Science* 1(12), 752–761.
- [5] Allahyari, M., Esfahanian, V., & Yousefi, K. (2020) The Effects of Grid Accuracy on Flow Simulations: A Numerical Assessment. *Fluids* 5(3), 110. <https://doi.org/10.3390/fluids5030110>
- [6] Ali, Z., Tyacke, J., Tucker, P. G., & Shahparb, S. (2016) Block topology generation for structured multi-block meshing with hierarchical geometry handling. *Procedia Engineering* 163, 212 – 224
- [7] Armstrong, C. G., Fogg, H. J., Tierneya, C. M., & Robinson T. T. (2015) Common themes in multi-block structured quad/hex mesh generation. *Procedia Engineering*. 124, 70–82.
- [8] Azarenok, B. N. (2010) On 2D Structured Mesh Generation using Mappings. *Numer. Methods Partial Differential Eq.* 27, 1072–1091. <https://doi.org/10.1002/num.20570>
- [9] Barrera, P., Hernández, V., Durán., C. (1996) El ABC de los Splines. *Aportaciones Matemáticas, Sociedad Matemática Mexicana*.
- [10] Barrera, P., González G. F., & Domínguez, F.J. (2003) Robust Discrete Grid Generation on Plane Irregular Regions. *USSR Comput. Math. Math. Phys.* 43(6), 884–892.
- [11] Barrera, P., Cortés, J.J., González, G. F., Domínguez, F.J., & Tinoco, J. G. (2010) Smoothness and Convex Area Functionals Revisted. *SIAM J. Sci. Comput.* 32(4), 1913–1928.

-
- [12] Barrera, P., & Méndez, I. (2021) Parametrization of Plane Irregular Regions: A Semi-Automatic Approach I In: Garanzha, V.A., Kamenski, L., Si, H. (eds). *Numerical Geometry, Grid Generation and Scientific Computing. Lecture Notes in Computational Science and Engineering* 143, 263–279. https://doi.org/10.1007/978-3-030-76798-3_17
- [13] Bommers, D., Campen, M., Ebke, H. C., Alliez, P., & Kobbelt, L. (2013) Integer-Grid Maps for Reliable Quad Meshing. *ACM Trans. Graphics* 32(4): 98, 1–12. <https://doi.org/10.1145/2461912.2462014>
- [14] Bommers, D., Levy, B., Pietroni, N., Puppo, E., Silva, C., & Zorin, D. (2013) Quad-Mesh Generation and Processing: A Survey. *Computer Graphics Forum* 32(6), 51–76. <https://doi.org/10.1111/cgf.12014>
- [15] Bern, M. & Eppstein, D. (2000) Quadrilateral Meshing by Circle Packing *International Journal of Computational Geometry & Applications* 10(4), 347–360.
- [16] Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017) Julia: A Fresh Approach to Numerical Computing. *SIAM Review* 2017 59(1) 65–98.
- [17] Blacker, T. D., & Stephenson, M. B. (1991) Paving: A new approach to automated quadrilateral mesh generation. *Int. J. Numer. Methods Eng.*, 32(4), 811–847
- [18] Bloch, M (accedido en 2022) Mapshaper v. 0.6.6 <http://mapshaper.org/>
- [19] Bobylëv, N. A., Ivanenko, S. A., & Kazunin A. V. (2003) On piecewise-smooth homeomorphic mappings of bounded domains and its applications to the theory of grids. *Comput. Math. Math. Phys.* 43(6), 772–781.
- [20] Bose, P., & Toussaint, G. (1995) No quadrangulation is extremely odd. In: J. Staples, et al. (eds). *Algorithms and Computations. ISAAC 1995*. 372–381
- [21] Brandt, C., Seidel, H. P., & Hildebrandt K. (2015) Optimal Spline Approximation via ℓ_0 -Minimization. In: Sorkine-Homung, Wimmer, M. (eds) *Eurographics 2015* 2(34).
- [22] Brackbill, J. U., & Saltzman, J. S. (1982) Adaptive Zoning for Singular Problems in Two Dimensions *Journal of Computational Physics* 46(3), 342–368.
- [23] Brinkhoff, T., Kriegel, H. P. Schneider, R., & Braun, A. (1995) Measuring the Complexity of Polygonal Objects. *ACM-GIS* 1995
- [24] Carreón, C. (2008) Un Módulo para el Tratamiento de Contornos en el Sistema UNAMALLA. *Tesis de Licenciatura*, Universidad Nacional Autónoma de México.
- [25] Castillo, O. (1999) A practical guide to direct optimization for planar grid generation. *Computers & Mathematics with Applications* 37(9), 123–156.

-
- [26] Cai, W. H., Zhan, J. M., & Luo Y. Y. (2021) User-intervened structured meshing methods and applications for complex flow fields based on multiblock partitioning. *J. Comput. Des. Eng.* 8(1), 225–238. <https://doi.org/10.1093/jcde/qwaa074>
- [27] Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein C. (2001) Introduction to Algorithms. *The MIT Press*, Second Edition, pp. 531–539.
- [28] Couplet, M., Reberol, M., & Remacle, J. F. (2021) Generation of High-Order Coarse Quad Meshes on CAD Models via Integer Linear Programming. <https://doi.org/10.48550/arXiv.2108.02635>
- [29] Danisch, S., & Krumbiegel J. (2021) Makie.jl: Flexible high-performance data visualization for Julia. *Journal of Open Source Software* 6(65), 3349. <https://doi.org/10.21105/joss.03349>
- [30] De Boor, C. (1986) B(asic)-spline Basics. *Technical Summary Report, Mathematics Research Center, University of Wisconsin-Madison.*
- [31] De Santis, G. (2011) Novel mesh generation method for accurate image-based computational modelling of blood vessels. *PhD Thesis. Ghent University. Faculty of Engineering and Architecture, Belgium*
- [32] Dielen, A., Lim, I., Lyon, M., & Kobbelt, L. (2021) Learning Direction Fields for Quad Mesh Generation. *Computer Graphics Forum 2021*, 40, 181–191. <https://doi.org/10.1111/cgf.14366>
- [33] Domínguez, F. J. (2005) Sobre la Generación Variacional Discreta de Mallas Casior-togonales en el Plano. *Tesis Doctoral, Universidad Nacional Autónoma de México.*
- [34] Dougherty, J. & Ilyankou, I. (2022) Hands-On Data Visualization: Interactive Storytelling from Spreadsheets to Code. *O'Reilly Open-access web edition.* <https://handsondataviz.org/>
- [35] Douglas, D. H. & Peucker, T. K. (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica* 1973, 10(2), 112–122.
- [36] Dospinescu, G., Lascu, M., Pohoata, C., & Tetiva, M. (2008) An elementary proof of Blundon's inequality. *J. Inequal. Pure and Appl. Math.* 2008, 9(4), 100.
- [37] Dunning, I., Huchette, J., & Lubin, M. (2017) JuMP: A Modeling Language for Mathematical Optimization. *SIAM Review* 59(2), 295–320. <https://doi.org/10.1137/15M1020575>.
- [38] Duren, P. (2004) Harmonic mappings in the plane. *Cambridge Tracts in Mathematics* 156

-
- [39] Evans, L. C. (2010) Partial Differential Equations. *American Mathematical Society Graduate Studies in Mathematics* 19.
- [40] Forrest, J., Ralph, T., & Vigerske, S. (Access on 2022) coin-or/Cbc: Version 2.9.9. <https://projects.coin-or.org/Cbc>.
- [41] Freeman, H. (1961) On the Encoding of Arbitrary Geometric Configurations. *IRE Transactions on Electronic Computers* EC-10(2), 260–268. doi:10.1109/TEC.1961.5219197
- [42] Frey, P. J., George, P. L. (2008) Mesh generation: application to finite elements. *IST Ltd and John Wiley & Sons, Inc.*
- [43] Garanzha, V. A. (2000) Barrier method for quasi-isometric grid generation. *Comput. Math. Math. Phys.* 40(11), 1617–1637 .
- [44] García, G. A. (2011) Un Sistema para la Generación Numérica de Mallas Estructuradas en Regiones con Agujeros. *Tesis de Licenciatura*, Universidad Nacional Autónoma de México.
- [45] García, G. A. (2016) Un Algoritmo de Partición de Regiones Planas en Regiones Simplemente Conexas para Generar Mallas Estructuradas por Bloques. *Tesis de Maestría*, Universidad Nacional Autónoma de México.
- [46] González, G. F. (2018) Generación Numérica de Mallas Estructuradas de Calidad y Adaptativas en Regiones Planas Irregulares. *Tesis Doctoral*, Universidad Nacional Autónoma de México.
- [47] Gravesen, A., Egrafov, J., Nguyen, D. M., & Nortoft, P. (2014) Planar Parametrization in Isogeometric Analysis In: M. Floater et al. Editors. *Mathematical Methods for Curves and Surfaces*, Springer, 189–212.
- [48] Haunert, J. H., & Sester, M. (2008) Area Collapse and Road Centerlines based on Straight Skeletons. *Geoinformatica 2008* 12, 169–191. <https://doi.org/10.1007/s10707-007-0028-x>
- [49] Hernández, V. (1996) Ajuste de datos por medio de un spline cónico. *Tesis de Maestría*, Instituto de Cibernética, Matemática y Física, Cuba.
- [50] Hongguang, Z., & Guozhao, W. (2015) New Framework for Decomposing a Polygon with Zero or More Holes. *The Open Cybernetics & Systemics Journal*, 9, 390–405.
- [51] Huang, W., Kamenski, L., & Russell, R. D. (2015) A comparative numerical study of meshing functionals for variational mesh adaptation. *Journal of Mathematical Study* 48(2), 168–186.

- [52] Huang, W. & Russell, R. D (2011) Adaptive Moving Mesh Methods. *Springer*.
- [53] Hughes, T. J. R., Cottrell, J. A., & Bazilevs, Y. (2005) Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement *Comput. Meth. Appl. Mech. Engrg.* 194, 4135–4195.
- [54] Ivanenko, S. A., & Charakhch'yan, A. A. (1987) An Algorithm for Constructing Curvilinear Meshes of Convex Quadrilaterals. *Dokl. Akad. Nauk. SSSR.* 295(2), 280–283.
- [55] Ivanenko, S. A. (1988) Generation of Non-Degenerate Meshes. *USSR Comput. Math. Math. Phys.* 28(5), 141–146.
- [56] Ivanenko, S. A. (1999) Harmonic Mappings. In: Thompson, J. F., Soni, B. K., Weatherill, N. P. (eds) *Handbook of Grid Generation*. CRC Press, Chap. 8.
- [57] Johnson, R. A. (1929) Modern Geometry: An Elementary Treatise on the Geometry of the Triangle and the Circle. *Houghton Mifflin Company, The Riverside Press, USA*.
online free version: <https://hdl.handle.net/2027/wu.89043163211>
- [58] Kapl, M., Buchegger, F., Bercovier, M., Jüttler, B. (2017) Isogeometric analysis with geometrically continuous functions on planar multi-patch geometries. *Comput. Methods Appl. Mech. Engrg.* 316, 209–234. <http://dx.doi.org/10.1016/j.cma.2016.06.002>
- [59] Kashyap, R. L., & Oommen, B. J. (1983) Scale preserving-smoothing of polygons. *IEEE Trans. Pattern Anal. Machine Intell.* 5(6), 667–671.
- [60] Kestelman, H. (1971) Mappings with non-vanishing Jacobian. *Amer. Math. Monthly* 78, 662–663.
- [61] Kluyver T. et al. (2016) Jupyter Notebooks: A Publishing Format for Reproducible Computational Workflows. In: Loizides F. and Schmidt B. (eds). *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press, 87–90.
- [62] Knupp, P., & Steinberg, S. (1994) Fundamentals of Grid Generation *CRC Press*.
- [63] Koch, M. D. (2021) Generating Block-Structured Kernels for Low Order Finite Element Methods: A High-Performance Oriented View. *PhD. Dissertation*,. *Westfälische Wilhelms Universität, Münster, Germany*.
- [64] Kopriva, D. A., Schoonover, J., & Winters, A. (Access 2022) HOHQMesh: the High Order Hex-Quad Mesher <https://github.com/trixi-framework/HOHQMesh>
- [65] Kowalski, N., Ledoux, F., & Frey. P. (2015) Automatic domain partitioning for quadrilateral meshing with line constraints. *Engineering with Computers* 31, Springer-Verlag, 405–421.

- [66] Lee, A., Dobkin, D., Sweldens, W., & Schröder, P. (1999) Multiresolution Mesh Morphing. In: *Computer Graphics Proceedings (SIGGRAPH 99)*, 343–350.
- [67] Levine, J. A., Paulsen, R. R., & Zhang, Y. (eds) (2012) *Mesh Processing in Medical Image Analysis 2012: MICCAI 2012 International Workshop, MeshMed 2012 Proceedings. Lecture Notes in Computer Science* 7599.
- [68] Lien, J. M. (2006) Approximate Convex Decomposition and its Applications. *PhD. Thesis. Texas A&M University, USA*
- [69] Lien, J. M., Keyser, J., & Amato, N. M. (2006) Simultaneous Shape Decomposition and Skeletonization. *Solid and Physical Modeling Symp., Association for Computing Machinery*, 219–228.
- [70] Lin, C., & More, J. J. (1999) Newton’s method for large bound-constrained optimization problems. *SIAM Journal on Optimization* 9, 1100–1127.
- [71] Liu, G., Xi, Z., & Lien, J. M. (2014) Dual-Space Decomposition of 2D Complex Shapes. *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4154–4161.
- [72] Liu, C., Chen, Z., & Li, X. (2017) 2D Quad Mesh generation using Divide and Conquer Poly-square Maps. *12th International Conference on Computer Science and Education (ICCSE)*, 263–268. <http://doi.org/10.1109/ICCSE.2017.8085500>.
- [73] Liu, C., Yu, W., Chen, Z., & Li, X. (2017) Distributed poly-square mapping for large-scale semi-structured quad mesh generation. *Computer-Aided Design* 90, 5–17. <http://dx.doi.org/10.1016/j.cad.2017.05.005>
- [74] Lo, S. H. (1985) A new mesh generation scheme for arbitrary planar domains. *Numerical Methods in Engineering* 21, 1403–1426.
- [75] Lo, S. H. (2015) Finite Element Mesh Generation. *CRC Press*.
- [76] Longuet-Higgins M. S. (2003) On the ratio of inradius to the circumradius of a triangle. *The Mathematical Gazette* 87(508), 119–120.
- [77] Lu, F., Qi, L., Jiang, X., Liu, G., Liu, Y., Chen, B., Pang, Y., & Hu, X. (2020) NNW-GridStar: Interactive Structured Mesh Generation Software for Aircrafts. *Adv. Eng. Softw.* 145: 102803. <https://doi.org/10.1016/j.advengsoft.2020.102803>.
- [78] Lyche, T. & Mørken, K. (2018) Spline methods. *Draft. University of Oslo*
- [79] Lyon, S. et al. (citado en Diciembre 2022) PlotlyJS.jl: Julia interface to plotly.js visualization library <https://github.com/JuliaPlots/PlotlyJS.jl>

-
- [80] Ma, D., Zhao, Z., Zheng, Y., Gou, R., & Zhu, W. (2020) PolySimp: A Tool for Polygon Simplification Based on the Underlying Scaling Hierarchy *ISPRS Int. J. Geo-Inf* 9(10), 594. <https://doi.org/10.3390/ijgi9100594>
- [81] Ma, X. W., Zhao, G. Q., & Sun, L. (2011) AUTOMESH-2D/3D: robust automatic mesh generator for metal forming simulation. *Materials Research Innovations* 15(1), W. S. Maney & Son Ltd.
- [82] Masood, A., & Haq, S. A. (2007) A novel approach to polygonal approximation of digital curves. *Journal of Visual Communication and Image Representation* 18(3), 264–274.
- [83] Masood, A. (2008) Dominant point detection by reverse polygonization of digital curves. *Image and Vision Computing* 26(5), 702–715.
- [84] Marshall, B., & Eppstein, D. (2000) Quadrilateral Meshing by Circle Packing. *International Journal of Computational Geometry & Applications* 4(10), 347–360.
- [85] Mattioli, D. D. (2017) QUADMESH+: A Quadrangular Advanced Mesh Generator for Hydrodynamic Models. *Master Thesis. The Ohio State University, USA*.
- [86] McAllister, D. F., & Roulier, J. R. (1981) An Algorithm for Computing a Shape Preserving Osculatory Quadratic Spline. *ACM Transactions on Mathematical Software* 7(3), 331–347.
- [87] McDonald, C. L. (2006) Automatic, Unstructured Mesh Generation for 2D Shelf Based Tidal Models. *Master Thesis. Brigham Young University, USA* 2006
- [88] Miller, K., & Miller, R. N. (1981) Moving Finite Elements I. *SIAM J. Numer. Anal.* 18(19), 1019–1032.
- [89] Mogensen, P. K., & Nilsen, R. A. Optim: A mathematical optimization package for Julia. *Journal of Open Source Software* 3(24): 615 (2018). <https://github.com/JuliaNLSolvers/Optim.jl/>
- [90] Montanari, U. (1970) A note on minimal length polygonal approximation to a digitized contour. *Commun. ACM* 13(1), 41–47.
- [91] Nash, J. C. (2000) The (Dantzig) simplex method for linear programming. In: *Computing in Science & Engineering* 2(1), 29–31.
- [92] Nowotny, D. (1997) Quadrilateral Mesh Generation via Geometrically Optimized Domain Decomposition. In: *Proceedings of 6th International Meshing Roundtable 1*, 309–320.

- [93] Castellanos, J. L. (1995) Generación de Redes usando Newton Truncado. *Tesis de Doctorado*, Instituto de Cibernética, Matemática y Física, Cuba.
- [94] Owen, S. J., Staten, M. L., Canann, S. A. & Saigal, S. (1998) Advancing Front Quad Meshing Using Local Triangle Transformation *Proceedings, 7th International Meshing Roundtable Conference*.
- [95] Papanelopoulos, N., & Avrithis, Y. (2015) Planar shape decomposition made simple. In: *British Machine Vision Conference 2015*.
- [96] Pavlidis, T. (1982) Curve Fitting as a Pattern Recognition Problem. *Proc. 6th Conf. Pattern Recognition in IEEE Computer Society Press*, 853–859.
- [97] Pointwise, Inc. (citado en 2022) Quality and Control – Two Reasons Why Structured Grids Aren’t Going Away. <https://www.pointwise.com/>
- [98] Prasad, D. K., Leung, M. K. H., Quek, C., & Cho, S.Y. (2012) A novel framework for making dominant point detection methods non-parametric *Image and Vision Computing* 30(11), 843–859.
- [99] Probst, A., Mazlum, E. & Radespiel, R. (2007) Investigation of computational uncertainties of airfoil flow phenomena close to trailing edges. *RTO-Symp. Computational Uncertainty in Military Vehicle Design*. RTO-MP-AVT-147.
- [100] Program Development Corporation. (citado en January 2022) GridPro 8.1. www.gridpro.com
- [101] Ramaiah, M., & Prasad, D. K. (2020) Polygonal Approximation of Digital Planar Curve Using Novel Significant Measure. In: C. Volosencu, S. Küçük, J. Guerrero, & O. Valero (eds), *Automation and Control. IntechOpen*. <https://doi.org/10.5772/intechopen.92145>
- [102] Rahman, S., Nishizeki, T., & Naznin, M. (2003) Orthogonal Drawings of Plane Graphs without Bends. *Journal of Graph Algorithms and Applications* 7(4), 335–362.
- [103] Ramírez, L. A. (2011) Un Módulo para el Suavizamiento de Contornos Usando Spline en Tensión para el Sistema Editboundary. *Tesis de Licenciatura*, Universidad Nacional Autónoma de México.
- [104] Randrianarivory, M. (2006) Geometric Processing of CAD Data and Meshes as Input of Integral Equation Solvers. *PhD. Thesis, Chemnitz University of Technology, Germany*.
- [105] Randrianarivory, M. (2008) Quadrilateral Decomposition by Two-Ear Property Resulting in CAD Segmentation. *International Journal of Physical and Mathematical Sciences, World Academy of Science, Engineering and Technology* 8(2), 523–529.

- [106] Raposo, P. (2013) Scale-specific automated line simplification by vertex clustering on a hexagonal tessellation. *Cartography and Geographic Information Science* 40(5), 427–443. <http://dx.doi.org/10.1080/15230406.2013.803707>
- [107] Ray, B. K., & Ray, K. S. (1994) A non-parametric sequential method for polygonal approximation of digital curves. *Pattern Recognition Letters* 15, 161–167.
- [108] Ray, K. S., & Ray, B. K. (2013) Polygonal Approximation and Scale-Space Analysis of Closed Digital Curves *Apple Academic Press, Inc.*
- [109] Reberol, M., Georgiadis, C., & Remacle J. F. (2021) Quasi-structured quadrilateral meshing in Gmsh – a robust pipeline for complex CAD models. arXiv:2103.04652.
- [110] Ren, Z., Yuan, J., & Liu, W. (2013). Minimum Near-Convex Shape Decomposition. *IEEE transactions on pattern analysis and machine intelligence* 35(10), 2546–2552. <https://doi.org/10.1109/TPAMI.2013.67>
- [111] Rivera, A. (1999) Un punto de vista sobre las cónicas y su uso en el suavizamiento de polígonos. *Tesis de Licenciatura*, Universidad Nacional Autónoma de México.
- [112] Rivera, A. (2005) Fundamentos de curvas de Bézier racionales y cálculo numérico de la reparametrización de curvas. *Tesis de Maestría*, Universidad Nacional Autónoma de México.
- [113] Roache, P. J., Steinberg, S. (1985) A new Approach to Grid Generation Using a Variational Formulation. AIAA 1985-1527. *7th Computational Physics Conference*, 360–370.
- [114] Roca, X., Sarrate, J. (2007) An Interactive Mesh Generation Environment for Geometry-Based Simulations. *6th Workshop Numer. Meth. Appl. Sci. Eng.* (NMASE 07).
- [115] Rosin, P. L. (1997) Techniques for Assessing Polygonal Approximations of Curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(6), 659–666. <https://doi.org/10.1109/34.601253>
- [116] Ruiz, E., Sarrate, J. (2008) Discretización de superficies múltiplemente conexas mediante submapping. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 24(2), 163–181.
- [117] Sadrehaghghi, I. (2022) Structure Meshing for CFD *CFD Open Series*
- [118] Sakov, P. (accedido en 2022) Gridgen. <https://github.com/sakov/gridgen-c>
- [119] Scanlan, J. P. (2011) Sharp feature identification in a polygon. *UNLV Theses, Dissertations, Professional Papers, and Capstones* 965.

-
- [120] Schneider, T., Panozzo, D., & Zhou, X. (2021) Isogeometric high order mesh generation. *Computer Methods in Applied Mechanics and Engineering* 386. <https://doi.org/10.1016/j.cma.2021.114104>
- [121] Schrijver, A. (1998) Theory of Linear and Integer Programming. *John Wiley & Sons Ltd.*
- [122] Secco, N. R., Kenway, G. K. W., He, P., Mader, C., & Martins J. R. R. A. (2021) Efficient Mesh Generation and Deformation for Aerodynamic Shape Optimization. *AIAA Journal* 59(4), 1151-1168. <https://doi.org/10.2514/1.J059491>
- [123] Simons, L., & Amenta, N. (2017) All-Quad Meshing for Geographic Data via Templated Boundary Optimization. *Procedia Engineering* 203, 388–400.
- [124] Skjermo, J., & Eidheim, O. C. (2005) Polygon Mesh Generation of Branching Structures. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds), *Image Analysis. SCIA 2005. Lecture Notes in Computer Science 3540*, 750–759. https://doi.org/10.1007/11499145_76
- [125] Shahpar, S., & Lapworth, L. (2012) PADRAM: Parametric Design and Rapid Meshing System for Turbomachinery Optimisation. *Proceedings of the ASME Turbo Expo 2012: Turbine Technical Conference and Exposition. Volume 8: Turbomachinery*, 2135–2148. <https://doi.org/10.1115/GT2012-69030>
- [126] Sheffer, A., Praun, E., & Rose, K. (2006) Mesh Parameterization Methods and Their Applications. *Found. Trends Comput. Graphics Vision* 2(2), 105–171.
- [127] Steinberg, S., & Roache, P. J. (1986) Variational Grid Generation. *Numerical Methods for Partial Differential Equations 2*, 71–96.
- [128] Sun, L., Armstrong, C. G., Robinson, T. T., & Papadimitrakis, D. (2021) Quadrilateral multiblock decomposition via auxiliary subdivision. *Journal of Computational Design and Engineering* 8(3), 871–893. <https://doi.org/10.1093/jcde/qwab020>
- [129] Takayama, K., Panozzo, D., & Sorkine-Hornung, O. (2014) Pattern-Based Quadrangulation for N -Sided Patches. In: *Computer Graphics Forum* 33, 177–184.
- [130] Tam, T. K. H., & Armstrong, C. G. (1991) 2D Finite Element Mesh Generation By Medial Axis Subdivision. *Adv. Eng. Softw. Work.* 13, 313–324.
- [131] Tinoco, J. G. (1997) Funcionales Discretos para la Generación de Mallas Suaves y Convexas sobre Regiones Planas Irregulares. *Tesis Doctoral*, Centro de Investigación en Matemáticas A. C., Guanajuato, México.
- [132] Touya, G. (accedido en 2023) CartAGen: a plugin for GeOxygene dedicated to cartographic generalisation. <https://ignf.github.io/CartAGen/>.

- [133] Tănase, M., & Veltkamp, R. C. (2003) Polygon Decomposition based on the Straight Line Skeleton. *Proceedings of the 19th Annual Symposium on Computational Geometry, Association for Computing Machinery*, 58–67.
- [134] Thompson, J. F. & Warsi, Z. U. A. & Mastin, C. W. (1985) Numerical Grid Generation: Foundations and Applications *North-Holland*, Amsterdam.
- [135] Visvalingam, M., & Whyatt, J. D. (1993) Line generalisation by repeated elimination of points. *Cartographic Journal* 30(1), 46–51. <http://dx.doi.org/10.1117/000870493786962263>
- [136] Visvalingam, M., & Whyatt, J. D. (2014) Implications of weighting metrics for line generalisation with Visvalingam’s algorithm. *The Cartographic Journal* 53(3), 253–267. <http://dx.doi.org/10.1080/00087041.2016.1149906>
- [137] Visvalingam, M. (2016) The Visvalingam Algorithm: Metrics, Measures and Heuristics. *The Cartographic Journal* 53(3), 242–252. <http://dx.doi.org/10.1080/00087041.2016.1151097>
- [138] UNAMalla Workgroup. (accedido en 2023) UNAMalla v4.0 <http://www.mathmoo.unam.mx/unamalla/>
- [139] Wislow, A. M. (1966) Numerical solution of quasilinear Poisson Equation in nonuniform triangle mesh. *J. Comput. Phys.* 1(2), 149–172.
- [140] Wislow, A. M. (1981) Adaptive mesh zoning by the equipotential method. *Technical Report UCID-19062*, Lawrence Livermore Laboratory.
- [141] Wolsey, L. A. (1998) Integer Programming *John Wiley & Sons*.
- [142] Wu, S. H., Chu, Y. M. (2014) Geometric interpretation of Blundon’s inequality and Ciamberlini’s inequality. *J. Inequal. Appl.* 2014, 381.
- [143] Wu, M., Chen, T., Zhang, K., Jing, Z., Han, Y., Chen, M., Wang, H., & Lv, G. (2018) An Efficient Visualization Method for Polygonal Data with Dynamic Simplification. *SPRS Int. J. Geo-Inf.* 2018 7(4), 138. <https://doi.org/10.3390/ijgi7040138>
- [144] Xiao, Z., He, S., Xu, G., Chen, J., & Wu, Q. (2020) A Boundary Element-Based Automatic Domain Partitioning Approach for semi-structured quad mesh generation. *Engineering Analysis with Boundary Elements* 11, 133–144.
- [145] Xu, G., Murrain, B., Duvigneau, R., & Galligo, A. (2011) Parametrization of Computational Domain in Isogeometric Analysis. methods and comparison. *Comput. Methods Appl. Mech. Engrg.* 200(23–24), 2021–2031. <https://doi.org/10.1016/j.cma.2011.03.005>

- [146] Xu, G., Li, M., Mourrain, B., Rabczuk, T., Xu, J., & Bordas, S. P. A. (2018) Constructing IGA-suitable planar parametrization from complex CAD boundary by domain partition and global/local optimization. *Comput. Methods Appl. Mech. Engrg.* 328, 175–200.
- [147] Zaib, A., Dhanasekaran, P. C., P. G. Tucker, R. Watson, & S. Shahpar. (2017) Optimal multi-block mesh generation for CFD. *International Journal of Computational Fluid Dynamics* 3(4-5) 195–213.
- [148] Zint, D., Grosso, r., Aizinger, V. & Köstler, H. (2019) Generation of Block Structured Grids on Complex Domains for High Performance Simulation. *Comput. Math. and Math. Phys.* 59, 2108–2123. <https://doi.org/10.1134/S0965542519120182>
- [149] Zhang, C., Chai, S., Liu, L., & Fu, X. M. (2021) Quad Meshing with Coarse Layouts for Planar Domains. *Computer-Aided Design* 140, 103084 <https://doi.org/10.1016/j.cad.2021.103084>
- [150] Zhang, Y. J. (2016) Geometric Modeling and Mesh Generation from Scanned Images. *CRC Press. Taylor & Francis Group.*
- [151] Zhang Y., & Jia, Y. (2009) CCHE-MESH 2D Structured Mesh Generator Users Manual—version 3.x. <https://www.ncche.olemiss.edu/cche-mesh-2/>
- [152] Zhang, Y., Jia, Y., Altinakar, M. S., Ding, Y., Ramalingam, V., & Kuiry, S. N. (2012) Structured Mesh Generation along Louisiana-Mississippi. In: Hagen, S., Chopra, M., Madani, K., Medeiros, S., Wang, D. (eds) *ICHE 2012. Proceedings of the 10th International Conference on Hydrosience & Engineering.* <https://hdl.handle.net/20.500.11970/109741>
- [153] Zhang, Y., & Jia, Y. (2018) 2D automatic body-fitted structured mesh generation using advancing extraction method. *Journal of Computational Physics* 353, 316–335.
- [154] Zeng, J. T., Lakaemper, R., Yang, X. W., & X. Li. (2008) 2D Shape Decomposition Based on Combined Skeleton-Boundary Features. In Bebis, G. (ed) *ISVC 2008 Part II, Lecture Notes in Computer Science* 5359, 682–691.