

Contenido

Prefacio	v
1 Planteamiento del problema	1
1.1 Notación	1
1.2 Descripción del problema continuo	2
1.3 Generación de mallas como un problema variacional	5
1.3.1 Una interpretación de los funcionales clásicos	6
1.4 Descripción del problema discreto	9
2 Funcionales Armónicos continuos y discretos	19
2.1 El sistema elíptico más simple	19
2.2 Funcionales armónicos discretos	27
2.3 Una discretización del funcional de Winslow	31
2.4 Funcional de Suavidad discreto regularizado	40
2.4.1 Propiedades del funcional discreto de Suavidad	42
3 Funcionales de Área continuos y discretos	55
3.1 Funcional de Área continuo clásico	55
3.2 Funcionales de Área continuos	58
3.3 Funcional discreto de Área de Castillo	61
3.4 Funcional de Área discreto, t -Área	64
3.5 Funcionales discretos de Área	69

3.6	Una forma general de los funcionales discretos de Área	70
3.6.1	Solución al problema de optimización	73
3.7	Ejemplos de funcionales discretos de Área	74
3.7.1	Funcional discreto inverso de Área	77
4	Funcionales para la generación automática de mallas	79
4.1	Funcional Modificado de Suavidad	79
4.2	Funcional Adaptivo de Suavidad	87
4.3	Funcional Inverso modificado de Área	90
4.4	Homotopía o trayectoria de adaptividad	94
4.5	Funcional Adaptivo de Área	95
4.6	Una descripción práctica de los funcionales Adaptivos	99
4.6.1	Sobre la actualización de k	100
4.6.2	Sobre la implantación del funcional de suavidad	104
4.6.3	Sobre la regularización	105
5	Combinaciones interesantes entre funcionales Nuevos y Clásicos	107
5.1	Normalización de Funcionales	108
5.2	Combinaciones interesantes entre funcionales Adaptivos y Clásicos . .	111
5.3	Homotopía entre funcionales	114
5.4	Homotopía y Adaptividad, un esquema práctico	116
6	Requerimientos de un sistema automático generador de mallas y su diseño	121
6.1	Análisis General	121
6.1.1	Objetivo	121
6.1.2	Panorama	121
6.1.3	Descripción del proyecto	122
6.1.4	Rango de alcance	122
6.1.5	Características del Usuario	123

6.1.6	Restricciones generales para el usuario	123
6.1.7	Requerimientos específicos	124
6.1.8	Requerimientos de Interfase externa	128
6.1.9	Requerimientos no funcionales	129
6.2	Diseño del sistema	129
6.2.1	Modularidad	130
6.2.2	Lenguajes de Programación	132
7	Conclusiones y trabajo a futuro	135
7.1	Conclusiones	136
7.2	Trabajo a futuro	137
A	Manual Operativo del Sistema Unamalla v. 2.0 para X Windows	139
A.1	Diseño e Interface del Usuario	139
A.1.1	Sobre el diseño	139
A.1.2	Interfase para el Usuario	141
A.2	Módulos principales	143
A.3	Módulo de Contornos	144
A.3.1	Construir un contorno en pantalla. <u>C</u> reate	145
A.3.2	Abrir un archivo de datos <u>L</u> oad.	147
A.3.3	Definir los segmentos de frontera. <u>D</u> efine Bnds.	148
A.3.4	Guardar información <u>S</u> ave	151
A.3.5	Generar una malla inicial por TFI <u>G</u> enerate	152
A.4	Módulo de Mallas	154
A.4.1	Llamar desde un archivo a una malla <u>L</u> oad	155
A.4.2	Herramientas de visualización de la malla <u>Z</u> oom	156
A.4.3	Optimización de malla <u>O</u> ptimize	160
A.4.4	Guardar información <u>S</u> ave	169
A.4.5	Aplicaciones con mallas convexas <u>A</u> pplications	169

A.5	Entrada y salida de datos	178
A.5.1	Formato del archivo de datos de los contornos	178
A.5.2	Formatos de archivo de mallas	180
A.5.3	Formato de archivos MESH	183
A.6	Sobre el sistema UNAMALLA y su nombre	184
A.6.1	Sobre el nombre UNAMALLA	184
A.6.2	Sobre el grupo de trabajo UNAMALLA	184
A.7	Requerimientos técnicos del sistema	185
A.7.1	Plataformas soportadas	185
A.7.2	Instalación de las distribuciones binarias	186
A.7.3	Soporte Técnico para el Sistema.	186
B	Métodos de Optimización de Gran Escala	187
B.1	Introducción	187
B.1.1	Búsqueda lineal y región de confianza	188
B.2	Método de Newton	189
B.3	Métodos Quasi-Newton	190
B.4	Problemas de gran escala	191
B.4.1	Métodos de Newton truncado	192
B.4.2	Descripción básica del método	192
B.5	Métodos Quasi-Newton para problemas de gran escala	193
B.5.1	Memoria Limitada BFGS	194
B.5.2	Método de Newton punto a punto	196
C	Galería de Mallas	201
	Bibliografía	216

Prefacio

El sistema UNAMALLA es el resultado de un enorme esfuerzo de un grupo de trabajo al construir pequeños programas o módulos que realizan tareas específicas para la generación de mallas razonables en regiones planas. Veamos una breve reseña de la evolución del sistema a lo largo de los últimos años.

GRID

El origen del sistema UNAMALLA se remonta al año de 1989 cuando fue diseñado el programa GRID. Este sistema fue construido en la ciudad de la Habana por los profesores Ángel Pérez y Pablo Barrera [2] quienes abordaron el problema de la generación numérica de mallas para resolver un problema sobre la bahía de la Habana, ver Figura 1.

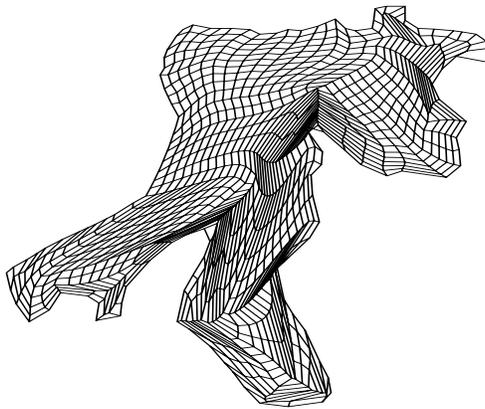


Figura 1: Una malla sobre la bahía de la Habana.

Aportaciones

Analizando con detenimiento la forma de atacar el problema discreto por Castillo [13] consideraron la malla como una colección de triángulos estructurados y no de

cuadriláteros. Propusieron el problema de minimizar la suma de cuadrados del área para generar mallas convexas.

El principal aporte de este programa fue verificar mediante la computadora que los resultados obtenidos por el funcional Área de Barrera-Pérez son superiores al funcional de Área de Castillo. La malla inicial considerada por el sistema se genera por interpolación lineal entre fronteras. Dado que el problema de optimización involucrado es de gran escala, la generación discreta de mallas resultó ser muy atractivo, ya que permite analizar los métodos de optimización existentes y plantea la búsqueda de un algoritmo eficiente para su resolución.

Detalles técnicos y plataforma

Con este sistema experimentaron con tres diferentes métodos de optimización de gran escala: Gradiente Conjugado de Fletcher-Reeves (1964), Memoria Limitada de Nocedal (1980) y Newton Truncado con búsqueda en la línea (Dembo 1983). Observaron los resultados obtenidos con cada uno de ellos. El programa estuvo escrito en lenguaje FORTRAN 77, estando escrito en lenguaje C el programa principal y teniendo al graficador como programa independiente, escrito en QUICKBASIC, todo esto desde luego para PC DOS.

Malla 1.0

Al año siguiente y a través de un acuerdo de colaboración entre la Academia de Ciencias de Cuba y la Facultad de Ciencias de la UNAM, patrocinado por el CONA-CyT en México se desarrolló la versión del sistema MALLA 1.0 donde es mejorada la interacción del sistema con el usuario.

Aportaciones

Definieren los archivos de entrada del contorno de una región poligonal así como la posibilidad de trabajar regiones con agujeros. De igual forma se define el archivo de mallas logrando con ello contar con un mecanismo de entrada y salida de datos del sistema, formato de uso directo para el usuario.

En el sistema MALLA 1.0 se elimina el uso del funcional de Castillo y sólo se consideran los funcionales de Área de Barrera-Pérez y el funcional de Longitud con los que Castillo ya había trabajado. Son incorporados distintos tipos de topologías de regiones de interés como triángulos y trapecios sobre las que se experimentan las mallas.



Figura 2: Meú principal del sistema MALLA v. 1.0.

Detalles técnicos y plataforma

Este sistema integra los módulos de generación inicial de la malla y de optimización, de esta manera queda establecida la modalidad de generación automática de mallas. El ambiente en que interactúan los módulos fue escrito en PASCAL y el graficador de mallas en lenguaje C haciendo uso de la biblioteca gráfica de Microsoft.

SEGRED

En 1990, Rina Ojeda en sus tesis de maestría [37] continúa estas ideas y construye el sistema SEGRED que permite darle un tratamiento a la frontera del contorno de la región. Este tratamiento consiste en generar los puntos sobre la frontera que permanecerán fijos. Esto lo logra usando interpolación lineal, la interpolación cúbica paramétrica o bien, a través del suavizamiento de datos por splines cúbicos paramétricos. De esta manera logra controlar y suavizar los datos de una región poligonal antes de construir una malla sobre ella.

Aportaciones

Haciendo uso de éstas herramientas queda constituida la modalidad automática de particionamiento del contorno de la región en subfronteras mediante un algoritmo que involucra la longitud del contorno y la admisibilidad de la región poligonal. La malla inicial es obtenida al considerar aquella que por interpolación entre fronteras opuestas obtenga el menor número de celdas no convexas. Otro aporte de esta tesis es el extender el funcional de Longitud como una suma ponderada entre el funcional que mide la tensión de las líneas verticales y el que mide la tensión de las horizontales; de ésta manera y mediante una elección adecuada del peso son tensadas unas líneas más que las otras.

Detalles técnicos y plataforma

Los programas siguen estando escritos en lenguaje FORTRAN 77 y el despliegue gráfico se integra como una opción del menú principal lo que permite observar la malla tanto al final del proceso de generación inicial como al final de la optimización. La interacción de los módulos se logra con un programa en Pascal. Todo esto en plataforma para PC DOS.

Programas y módulos experimentales

Entre 1991 y 1994 se experimenta con los programas del sistema como módulos separados; son mejorados los métodos de optimización tenidos a prueba [12] y son estudiados las propiedades sobre regiones muy irregulares. Son comparados los resultados obtenidos tanto para PC como con la SuperComputadora Cray.

Aportaciones

En estos sistemas experimentales se incorpora el funcional de Area-Ortogonalidad. Se incorpora una regularización al funcional de discreto de Suavidad [5] logrando con ello generar mallas suaves y convexas en algunas regiones irregulares pero con la restricción de iniciar el proceso de optimización con mallas de pocas celdas no convexas. La generación de mallas sobre regiones muy irregulares como la bahía de la habana se torna un proceso laborioso al obtener una malla con pocas celdas no convexas para tomarlo como partida de otro.

Detalles técnicos y plataforma

Los programas continúan escritos en lenguaje FORTRAN pero ahora se experimenta con el compilador de Lahey que permite trabajar con PC y con Estación de Trabajo de manera transparente, excepto claro está en las particularidades del sistema de cada plataforma. Uno de los estudiantes de licenciatura de Barrera diseña una biblioteca genérica de graficación para este compilador [25] para ser usado fácilmente en los cursos de Análisis Numérico que se imparte en la Facultad de Ciencias. En 1993 me incorporo al grupo de trabajo del profesor Barrera e integro esta biblioteca gráfica al programa experimental de mallas logrando con ello sea posible observar la evolución del proceso de generación de mallas sobre la pantalla gráfica; ya no analizamos números a la salida, ahora observamos su evolución durante el proceso.

UNAMALLA 1.0

En 1994 mediante un convenio entre Cuba y México se desarrolla una versión profesional del sistema en lenguaje C facilitando al usuario la interacción con el sistema mediante opciones de menú integradas. Este sería el sistema UNAMALLA v. 1.0, ver Figura 3.



Figura 3: Presentación del sistema UNAMALLA v. 1.0.

Aportaciones

En esta versión se incorporan todas las ideas obtenidas de los experimentos realizados hasta entonces; se incorpora la modalidad automática con las opciones de combinación convexa de funcionales que mejores resultados presentaron en la práctica. Son incorporadas utilerías para la creación y edición gráfica de los contornos de regiones mediante el teclado así como la definición de los segmentos de fronteras. Es posible observar gráficamente el comportamiento del proceso de optimización y detenerlo por instrucciones del teclado. La generación inicial de la malla se logra mediante el método algebraico TFI.

Detalles técnicos y plataforma

En esta versión sólo se incorpora el método de optimización de Memoria Limitada de Nocedal, siendo el principal interés lograr configuraciones razonables mediante la optimización de funcionales discretos, se cambia la estructura empleada para el almacenamiento interno de la malla, ahorrando tiempo de ejecución. De igual forma se propone un nuevo formato de archivos de intercambio de información en disco. Sin embargo, este sistema presentó serias dificultades de operación. El diseño del sistema resultó ser muy cerrado en el intercambio de información de las mallas entre los módulos. Lo cual no permitió hacer experimentos con nuevos funcionales, con ideas geométricas interesantes a observar en las mallas. Tampoco permitió incorporar

los algoritmos de optimización que se estudiaban. Por otra parte, la interacción del sistema a archivos de información fue a través de proyectos de trabajo, su almacenamiento contó con un formato que considera el contorno y la malla generada tanto la inicial como aquella obtenida al final de la optimización. Debido a esta modalidad el sistema resultó muy difícil de incorporar en aplicaciones específicas. El formato de archivos fue deshechado en posteriores sistemas.

Sistemas experimentales

Posteriormente, se regresó a los sistemas experimentales programados en **FORTTRAN**. Para ello, usamos el compilador de **Microsoft 5.1** el cual emplea de manera transparente al usuario las bibliotecas gráficas de **C Microsoft 6.0**. Se experimenta con versiones de **Newton Truncado con Región de Confianza** y con **Búsqueda en la Línea**. En forma paralela, se continúa con los programas desarrollados para **Lahey** y se hace uso de la memoria dinámica para generar mallas de dimensión grande. En la Figura 4 se muestra la pantalla de uno de los sistemas.

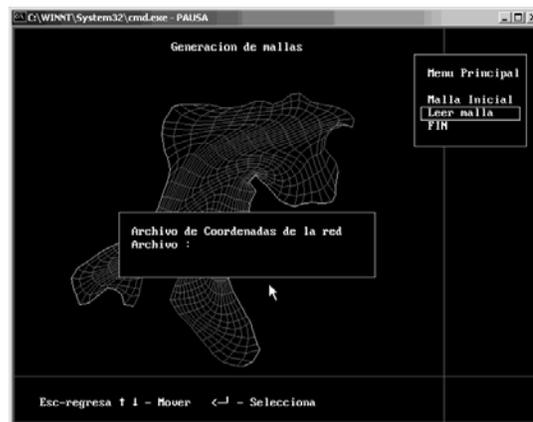


Figura 4: Un sistema experimental para la generación de mallas haciendo uso de los funcionales discretos clásicos, escrito en **Fortran Lahey**.

Se construye una utilería que permite suavizar contornos mediante splines cónicos, ver Figura 5. Esta herramienta facilita la obtención de mallas suaves y convexas al eliminar la limitante de picos sobre todo en los segmentos cóncavos en la frontera, pudiendo así generar mallas suaves y convexas sobre algunas regiones usando los funcionales de Área-Ortogonalidad y de Suavidad Regularizado [7].

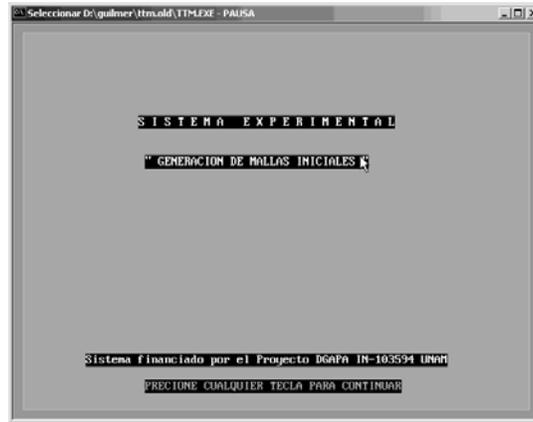


Figura 5: Un sistema experimental para la generación de mallas iniciales, con suavizado NURBS en las fronteras de la región.

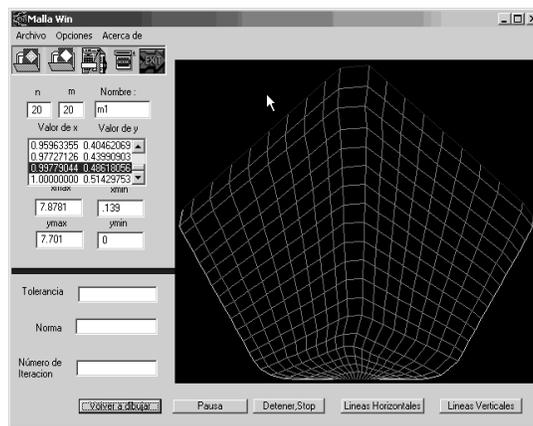


Figura 6: Un sistema experimental para Windows.

TRIANG

La idea que llevó al profesor Barrera a considerar la malla como una colección de triángulos estructurados le condujo a obtener resultados muy interesantes sobre mallas convexas. Esta inquietud le hace proponer a Sergio Aguirre en 1995 como trabajo de tesis el estudio de configuraciones óptimas de triangulaciones sobre regiones planas. Trabajaron con las triangulaciones de Delaunay y compararon resultados con los funcionales discretos de Barrera-Pérez y de Suavidad formulados ambos sobre triángulos. Construyeron un sistema TRIANG para observar el comportamiento de los funcionales. En la Figura 7 se observa el sistema.

Aportaciones

Son comparando las configuraciones obtenidas con las triangulaciones de Delaunay y de otros tipos obteniendo un resultado muy valioso: un óptimo del funcional de Suavidad es una triangulación de Delaunay.

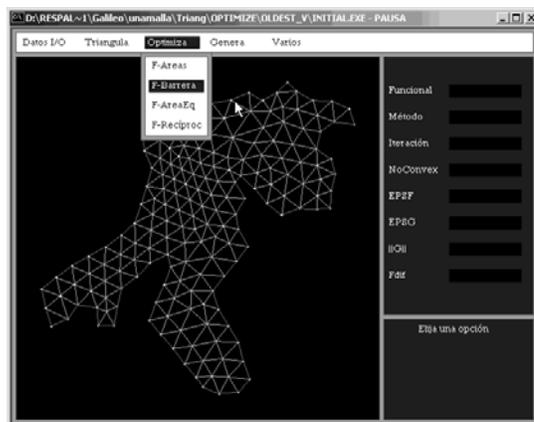


Figura 7: Sistema de triangulaciones óptimas.

De igual forma son aplicadas las mallas obtenidas en la resolución de algunas ecuaciones elípticas y se comparan los resultados sobre puntos específicos usando el Toolbox PDETOOLS de MATLAB.

Detalles técnicos y plataforma

El sistema se escribió en lenguaje C ANSI haciendo uso de las bibliotecas gráficas de Microsoft 6.0. Se consideró un formato sencillo para el intercambio de archivos.

UMALLA

Una de las dificultades para la operación del funcional de Suavidad Regularizado sobre regiones planas irregulares, es iniciar con mallas casi-convexas. En 1996 Gerardo Tinoco se incorpora al grupo de trabajo del profesor Barrera quien le propone para su tesis doctoral el estudio del funcional de Suavidad para formular una variante que conserve sus características y pueda operar sin la limitante de contar con una malla casi-convexa como inicio. Desarrolla métodos que amplían el dominio de trabajo con mallas no convexas y que a través de un esquema adaptivo conduce a mallas convexas de existir éstas. Este trabajo sería parte esencial para posteriores estudios y desarrollos de nuevos funcionales discretos. Los funcionales que Tinoco propuso se encuentran programados en el sistema UMALLA.

Aportaciones

No es necesario iniciar el proceso de mallas con pocas celdas no convexas. Los métodos que desarrolla son adaptivos y con ellos es posible controlar incluso el tamaño del área de las celdas. Los funcionales son llamados Funcionales Adaptivos de Área y de Suavidad.

Detalles técnicos y plataforma

Se considera una modalidad de optimización puntual que permite manejar mallas de grandes dimensiones. El sistema fue escrito en lenguaje FORTRAN de Lahey, haciendo uso de las bibliotecas gráficas construidas hasta entonces.

UNAMALLA v.2.0 para PC

A inicios de 1998 es desarrollado un sistema para PC que incorpora los nuevos funcionales de Área y de Suavidad en combinación con funcionales clásicos.

Aportaciones

De igual forma se implementa una modalidad de optimización puntual de Newton. Se incorpora una modalidad de homotopía para lograr la combinación convexa de los funcionales adaptivos de Área con Longitud y el adaptivo de Suavidad con Área clásica.

Detalles técnicos y plataforma

Se implanta una interfase con el usuario amigable al considerar menús de control con accesos directo desde el teclado. Es posible editar el contorno de manera gráfica con el uso del ratón. Se incorpora al sistema una modalidad de suavizamiento previo que consiste en aplicarle a la malla algunos pasos del optimizador usando el funcional de Longitud para eliminar muchas celdas no convexas en mallas dobladas. Este es el Sistema UNAMALLA v. 2.0 para PC, ver Figura 8.

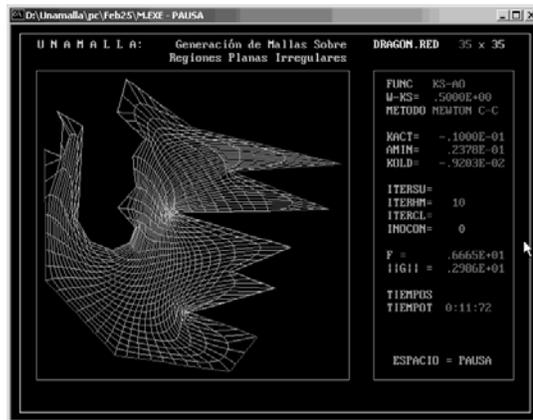


Figura 8: Sistema UNAMALLA v. 2.0 para PC, DOS.

UNAMALLA v.2.0 for X Windows

El objetivo de la tesis es desarrollar un sistema multiplataforma para UNIX que haga uso de bibliotecas gráficas libres y portables. Primeramente, y en base a la experiencia adquirida de la participación en el desarrollo de los sistemas y programas experimentales, describimos los requerimientos necesarios para el desarrollo de un sistema totalmente automático. Esto es definir cada una de las tareas que ha de realizar el sistema y su interacción a través de una interfase amigable al usuario. Otro de los puntos fundamentales será el describir de manera puntual el proceso de discretización de los funcionales continuos y discutir las propiedades geométricas de la malla óptima.

En lo que sigue describiremos la filosofía de trabajo del sistema, su estructura de programación así como la teoría básica que sustenta los funcionales incorporados.

Introducción: Se define el problema de la generación de mallas sobre regiones planas tanto en su forma continua como en su forma discreta. Se discute sobre algunas técnicas para resolver el problema de forma variacional continua, así como una interpretación de los funcionales.

Capítulo 2: Se hace énfasis en la construcción de mallas suaves y convexas a través de mapeos armónicos continuos y discretos entendiéndose en primer lugar, el funcional elíptico más simple. Se estudia el funcional de Winslow en su formulación continua y se plantea una discretización del mismo al restringirlo sobre cada celda de la malla transformando el problema continuo a un problema discreto que involucra un proceso de optimización de gran escala.

Capítulo 3: Siguiendo la idea de la discretización del funcional de Winslow, se discretiza el funcional de Área y se plantea una forma general de obtener funcionales discretos de Área, donde una malla uniforme en área sea punto crítico de ellos. En forma particular se estudia al funcional inverso de área.

Capítulo 4: Los funcionales que se estudian únicamente operan sobre mallas casi-convexas. En este capítulo se describe una forma de extender los dominios de las funciones de suavidad e inverso de área. Se obtiene una colección de funcionales llamados k -funcionales que a través de un procedimiento adaptivo se logra obtener mallas convexas. Se estudia el proceso adaptivo y se propone una forma práctica de llevarlo a cabo.

Capítulo 5: Se combinan los k -funcionales de suavidad y de área con los funcionales clásicos de área y de longitud, a fin de contar con una malla convexa suave. Se propone un procedimiento homotópico para combinar las propiedades de cada funcional.

Capítulo 6: Se describe los requerimientos para el posterior desarrollo de un sistema automático generador de mallas que hace uso de los nuevos funcionales. Se hace hincapié en el rango de alcance del sistema, las características del usuario, los requerimientos de interfase externa; para entonces proponer un diseño modular basado en las distintas tareas aquí descritas.

Apéndice A: Manual del sistema multiplataforma desarrollado donde hemos implantado los nuevos funcionales de área y de suavidad. Este sistema está basado en los requerimientos propuestos en el capítulo 6.

Apéndice B: Son descritos los algoritmos de optimización de gran escala que se emplean en el sistema multiplataforma.

Apéndice C: Colección de mallas generadas con el sistema, donde es posible apreciar las distintas configuraciones de mallas convexas acordes a la combinación elegida entre funcionales.

Capítulo 1

Planteamiento del problema

La generación de mallas es un proceso que consiste en partir un dominio físico en otros subdominios elementales para observar algún fenómeno medible sobre cada uno de los subdominios, ver Figura 1.1. Una malla estructurada es aquella bajo la cual podemos identificar en forma rápida los nodos de la malla, interiores y exteriores y cada nodo interior cuenta con el mismo número de elementos adyacentes. Este tipo de mallas son muy socorridas en aplicaciones de EDP, pero presentan muchas dificultades en la aproximación sobre la frontera cuando el dominio no es simple. Existen muchos métodos para obtener este tipo de mallas estructuradas y uno de ellos es el método del mapeo que consiste en obtener un mapeo que transforme el dominio físico en uno mas sencillo y entonces lograr de forma directa una malla, ver Figura 1.2.

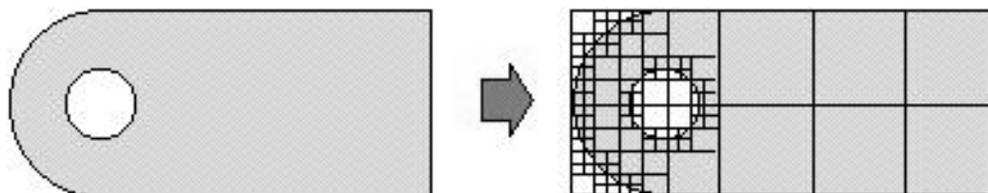


Figura 1.1: Una malla por un proceso *quadtree*, tomado de [38].

1.1 Notación

La terminología empleada en la generación de mallas es muy flexible. Por ejemplo, a la región de estudio se conoce como **región física**, así como **dominio físico** o **espacio físico**; a la región elemental le llamaremos **región lógica** y **espacio lógico**, aunque estos últimos términos espacio lógico y físico son adecuados emplearlos para el espacio vectorial en el cual está inmersa la región referida. Es importante señalar

que no se hace diferencia alguna entre mapeo y transformación en cuanto a la función que cada cual desempeña.

En el espacio lógico consideraremos el sistema coordenado usual, donde los ejes horizontal y vertical los denotaremos por las letras griegas ξ y η ; con esto, un punto cualquiera en el espacio lógico estará perfectamente determinado por la pareja $(\xi, \eta)^t$, con $\xi, \eta \in \mathbb{R}$, en caso de que el punto pertenezca a la región elemental, tendremos que sus componentes serán tales que $0 \leq \xi \leq 1$ y $0 \leq \eta \leq 1$.

Denotaremos B_2 a la región plana elemental (cuadrado unitario). También denotaremos por Ω a la región física donde $\Omega \subset \mathbb{R}^2$ y es **simplemente conexa**. Aunque como se verá posteriormente, haciendo algunos cortes podemos considerar a cualquier región en una simplemente conexa por secciones.

1.2 Descripción del problema continuo

El problema de la generación de mallas planas puede ser planteado de la siguiente forma. Sobre de una región Ω plana acotada y simplemente conexa, consideremos una transformación continua del cuadrado unitario en la región Ω

$$\mathbf{x} : B_2 \longrightarrow \Omega, \quad (1.1)$$

una malla sobre Ω es una transformación continua $\mathbf{x}(\xi, \eta) = (x(\xi, \eta), y(\xi, \eta))$

$$\mathbf{x} : B_2 \longrightarrow \Omega \quad (1.2)$$

donde B_2 es el cuadrado unitario $[0, 1] \times [0, 1]$.

Si sobre del cuadrado unitario generamos una retícula de tal manera que una los lados opuestos dos a dos; al mapear estas líneas bajo el mapeo \mathbf{x} obtendremos una malla sobre la región, ver Figura 1.3.

De manera particular, estamos interesados en aquellos mapeos \mathbf{x} que conformen la frontera de Ω ,

$$\mathbf{x}(\partial B_2) = \partial \Omega \quad (1.3)$$

esto es, que la frontera de B_2 sea mapeada a la frontera de Ω . Bajo esta idea, el problema de la generación de mallas puede plantearse como

dada una transformación biyectiva y continua entre la frontera de B_2 y de Ω , extenderla a una transformación continua entre B_2 y Ω

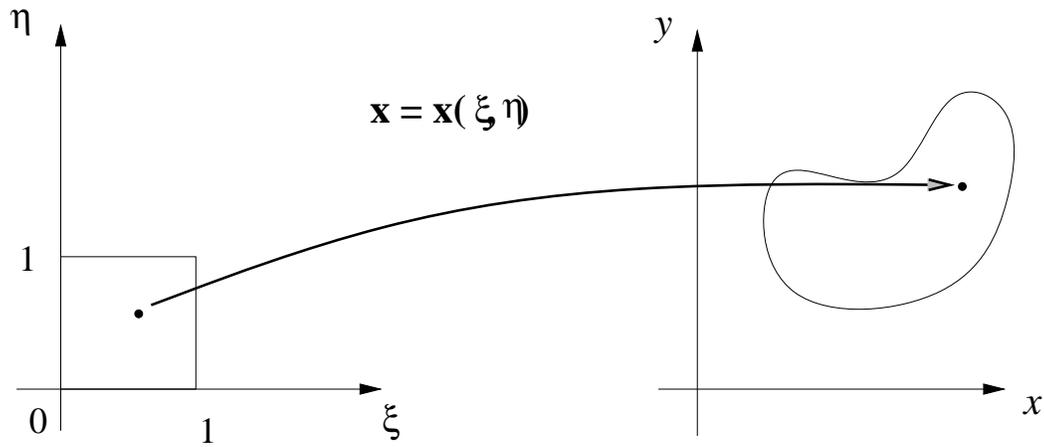


Figura 1.2: Mapeo entre regiones.

en otras palabras, extender un mapeo continuo y biyectivo entre fronteras

$$\mathbf{x} : \partial B_2 \longrightarrow \partial \Omega \quad (1.4)$$

a un mapeo continuo entre regiones

$$\mathbf{x} : B_2 \longrightarrow \Omega. \quad (1.5)$$

Sin embargo, no cualquier mapeo es útil. Para generar una malla razonable sobre Ω debe considerarse que las líneas no se entrelacen y que los lados del cuadrado sean mapeados a líneas sobre la frontera. La característica de que el mapeo sea continuo garantiza que el orden de los lados del cuadrado bajo el mapeo sea conservado. Ver Figura 1.4.

Contar con un mapeo continuo entre fronteras nos lleva a considerar la frontera de la región en cuatro segmentos, segmentos curvilíneos a decir verdad o sub-fronteras, conservando el orden impuesto por el cuadrado unitario, ver Figura 1.4.

Con este orden, la idea gráfica detrás de la generación de una malla sobre Ω es unir las fronteras 1-2 con 4-3 y 1-4 con 2-3 por medio de segmentos curvilíneos de manera tal que no se intersecten. La idea es “tirar” líneas entre segmentos opuestos.

Más adelante explicaremos cómo construir estas líneas de manera discreta, es decir, cómo generar los puntos que describen de manera discreta a los segmentos curvilíneos.

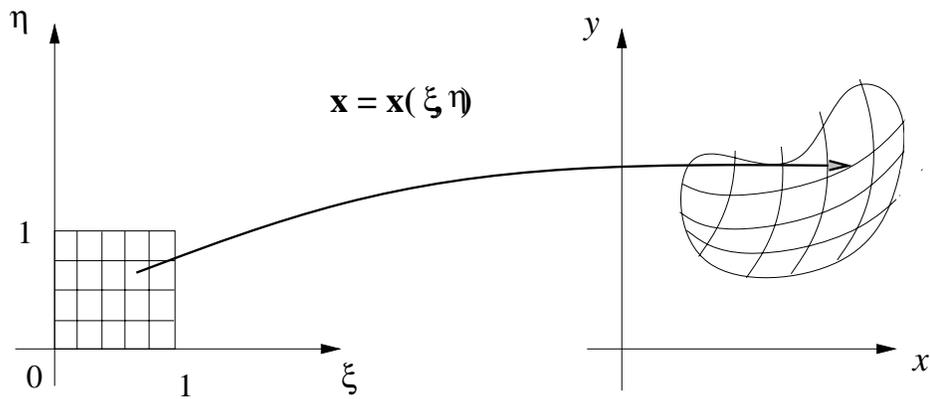


Figura 1.3: Una malla sobre Ω .

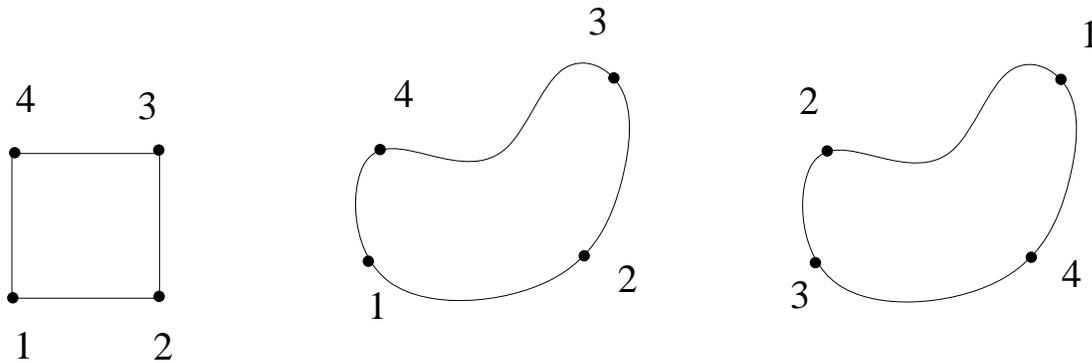


Figura 1.4: Conservación del orden del mapeo de las fronteras.

Una contracción de la frontera hacia el interior sería un ejemplo de obtención de una malla a partir de la frontera de la región, ver Figura 1.5.

Regresando a la construcción del mapeo como extensión de un homeomorfismo entre fronteras debemos tener presente que la solución no es única por lo que es necesario imponer condiciones que garanticen que la construcción esté bien definida. Por una parte resulta no deseable que:

1. Un punto dentro del cuadrado unitario sea mapeado a un punto fuera de la región Ω , ver Figura 1.6(a).
2. Que más de un punto del cuadrado unitario sea mapeado al mismo punto sobre la región física, ver Figura 1.6(b).

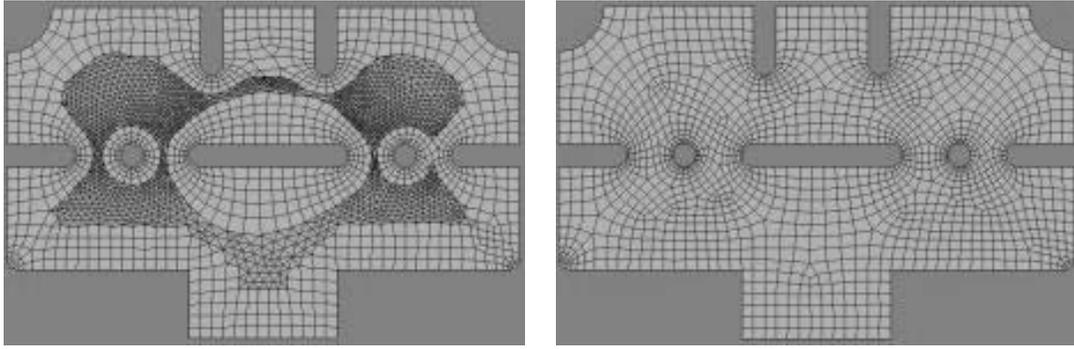


Figura 1.5: Construcción de una malla a través de la contracción de la frontera hacia el interior de la región *front end*. Tomado de [38].

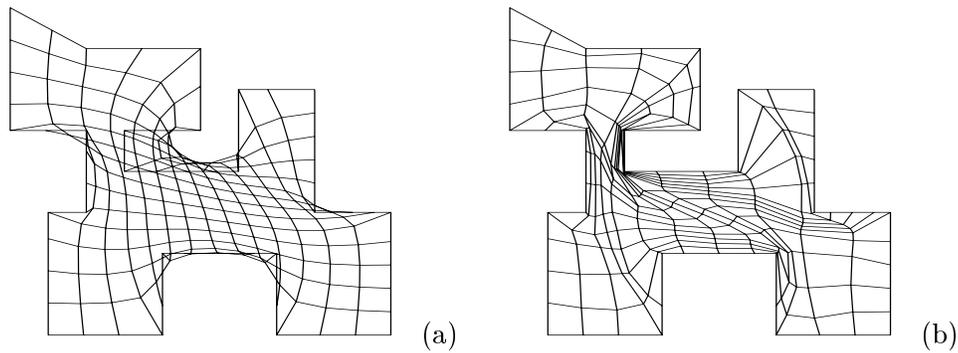


Figura 1.6: Mallas dobladas sobre la región del gato.

En cualquiera de los casos, diremos que la malla sufre un doblado y por consiguiente, el mapeo no es útil ya que el jacobiano de la transformación se anula y el cambio de coordenadas no es válido.

Teniendo en cuenta que el jacobiano de la transformación no se anule para obtener convexidad de las celdas, es deseable encontrar mapeos en los que el jacobiano sea positivo. Podemos pedir más que eso, por ejemplo, un comportamiento particular del jacobiano en zonas de la región y que las líneas curvilíneas sean suaves.

1.3 Generación de mallas como un problema variacional

Bajo la idea de imponer condiciones al homeomorfismo resultante podemos considerar al problema de la generación de mallas como un problema variacional continuo:

encontrar un mapeo de entre aquellos de frontera conforme, es decir, que cubra la frontera, que sea no doblado y con algunas características geométricas a observar.

Típicamente los funcionales considerados en un problema variacional para la generación de mallas que involucran medidas de suavidad y de tensión de los segmentos curvilíneos, así como el comportamiento del jacobiano, tienen la forma:

$$I(\mathbf{x}) = \int_0^1 \int_0^1 L(x_\xi, x_\eta, y_\xi, y_\eta) d\xi d\eta \quad (1.6)$$

donde L , al que llamaremos Lagrangiano, mide las características geométricas de los segmentos curvilíneos y es nuestro objetivo estudiar aquellos que describan un mapeo acorde a nuestros propósitos.

Así, considerando un Lagrangiano continuo con las características a observar sobre la malla, el problema de la generación de mallas puede plantearse como *hallar el mapeo entre B_2 y Ω de entre todos aquellos donde la frontera de B_2 es mapeada a $\partial\Omega$ y que minimice a $I(\mathbf{x})$* , lo que escribimos como

$$\mathbf{x}^* = \arg \min_{\mathbf{x}(\partial B_2) = \partial\Omega} I(\mathbf{x}). \quad (1.7)$$

Por el momento sólo estamos interesados en considerar propiedades que hacen que L sea continuamente diferenciable sobre Ω . Bajo esta observación, y siguiendo la teoría básica del cálculo de variaciones [20] tenemos que la solución a (1.7) debe satisfacer

$$\frac{d}{d\xi} \left(\frac{\partial L}{\partial x_\xi} \right) + \frac{d}{d\eta} \left(\frac{\partial L}{\partial x_\eta} \right) = 0 \quad (1.8)$$

$$\frac{d}{d\xi} \left(\frac{\partial L}{\partial y_\xi} \right) + \frac{d}{d\eta} \left(\frac{\partial L}{\partial y_\eta} \right) = 0 \quad (1.9)$$

Estas son las condiciones de Euler–Lagrange. Por todo esto, la solución de (1.7) está ligada a cómo resolver el sistema (1.8)–(1.9) con condiciones a la frontera

$$\mathbf{x}(\partial B_2) = \partial\Omega.$$

1.3.1 Una interpretación de los funcionales clásicos

Como hemos señalado, estamos interesados en medir ciertas propiedades geométricas en las mallas. Por ejemplo, si nuestro interés es generar mallas cuyos segmentos

curvilíneos sean suaves una manera de lograrlo es medir la razón de cambio de los segmentos sobre la región [54], una representación de tal medida es

$$I(\mathbf{x}) = \int_0^1 \int_0^1 (x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2) d\xi d\eta \quad (1.10)$$

este es el funcional de Longitud y mide la tensión de los segmentos curvilíneos en las direcciones $\xi - \eta$.

Como se observa de este funcional, la medida involucrada actúa tanto en la línea ξ como en la η , luego si la región de estudio es muy irregular, necesitaremos tenzar unas líneas más que otras [3] usando para esto un peso $0 \leq \omega \leq 1$ entre cada medida y de la forma

$$I(\mathbf{x}) = \omega \int_0^1 \int_0^1 (x_\xi^2 + y_\xi^2) d\xi d\eta + (1 - \omega) \int_0^1 \int_0^1 (x_\eta^2 + y_\eta^2) d\xi d\eta \quad (1.11)$$

Ahora bien, siendo el Jacobiano de la transformación \mathbf{x} una medida del área local, estamos interesados en su comportamiento a lo largo de la región de tal forma que al considerar

$$I(\mathbf{x}) = \int_0^1 \int_0^1 J(\xi, \eta)^2 d\xi d\eta \quad (1.12)$$

en el sentido de mínimo de cuadrados deseamos encontrar un mapeo que transforme la retícula uniforme en B_2 en una malla de área uniforme en Ω .

Sin embargo, no siempre es posible obtenerlo, ya que las ecuaciones de Euler-Lagrange son no lineales y no elípticas [19]; y si añadimos la irregularidad de una región, pudiera no existir su solución. Por otra parte, numéricamente este tipo de sistemas son muy difíciles y costosos de resolver.

Regresando a las propiedades que deseamos observar en la malla, podemos combinar los dos últimos funcionales de manera que el comportamiento de la tensión de las líneas y entonces su suavidad se encuentren en correspondencia al área local de la región de tal manera que donde el área sea pequeña las líneas no se tencen tanto y donde el área sea grande sí ocurra esto. Bajo esta característica, un funcional que mide esta propiedad es

$$I(\mathbf{x}) = \int_0^1 \int_0^1 \frac{x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2}{J(\xi, \eta)} d\xi d\eta. \quad (1.13)$$

Acorde a su propiedad, las líneas de mallas resultantes por este funcional se aproximan a fronteras cóncavas y se alejan de las convexas [30]. Este es el funcional de Suavidad.

Otra medida muy importante es el ángulo que forman los segmentos curvilíneos que se intersectan. Por ejemplo, si la malla deseada sobre la región es aplicada a la solución numérica de ecuaciones diferenciales, y los segmentos de la malla son ortogonales, los términos $\xi\eta$ se anulan por lo que el sistema a resolver es más sencillo. En el caso de que los segmentos estén cercanos a la ortogonalidad, esos términos influyen poco en la solución. El funcional de Ortogonalidad

$$I(\mathbf{x}) = \int_0^1 \int_0^1 (x_\xi y_\xi + x_\eta y_\eta)^2 d\xi d\eta. \quad (1.14)$$

cuenta con esta propiedad. Sin embargo, no siempre es posible obtener una malla sobre cualquier región ya que sus ecuaciones de Euler–Lagrange suelen ser no elípticas [47] lo que dificulta el proceso de obtención de una malla.

Hasta aquí, la solución a (1.7) usando estos funcionales no siempre existe y de existir podrían no generar mallas convexas. Sin embargo, la idea de observar las propiedades de convexidad y suavidad de las líneas nos permite experimentar con combinaciones de los funcionales ponderados, dependiendo del comportamiento global o incluso local, que deseamos cuente la malla.

Un ejemplo lo encontramos al considerar el Lagrangiano de área en suma ponderada con el de longitud, digamos en la forma

$$L(\mathbf{x}) = \omega J(\xi, \eta)^2 + (1 - \omega)[x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2] \quad (1.15)$$

donde ω es un escalar entre 0 y 1, que se elige dependiendo de si deseamos observar mayor área de las celdas o bien, tenzar más las líneas.

La importancia de esta combinación radica en que si ω no es cero, las ecuaciones de Euler–Lagrange asociadas a (1.7) son acopladas y elípticas, la solución existe y es fácil de encontrar. Sin embargo, la misma combinación no garantiza mallas convexas ni pidiendo uniformidad en celdas, esto es eligiendo ω cercana a 1.0 [4].

Otra combinación atractiva viene al considerar el Lagrangiano de área y el de ortogonalidad, en combinación

$$L(\mathbf{x}) = J(\xi, \eta)^2 + (x_\xi x_\eta + y_\xi y_\eta)^2 \quad (1.16)$$

expandiendo a J y agrupando términos tenemos que

$$L(\mathbf{x}) = (x_\xi^2 + x_\eta^2)(y_\xi^2 + y_\eta^2) \quad (1.17)$$

dando a lugar un funcional muy sencillo de evaluar y que en la práctica produce mallas con celdas uniformes, o cerca de lograrlo y de líneas cercanas a la ortogonalidad. Las ecuaciones de Euler–Lagrange para este funcional son elípticas [29]. Por esta última propiedad tenemos que las mallas generadas tienen líneas suaves, es decir, este funcional es muy útil a nuestro propósito. Sin embargo, por la influencia de la ortogonalidad, las mallas así obtenidas pueden resultar dobladas y entonces no convexas, esto puede ocurrir en muchas regiones, pero con una adecuada reparametrización de la frontera alguna vez podemos hallar una malla con estas tres características [29].

Hasta aquí, hemos discutido algunos funcionales que miden propiedades geométricas deseables a observar en una malla y que mediante un proceso variacional es posible encontrar el mapeo que transforma la retícula sobre del cuadrado unitario en una malla sobre Ω . Cabe señalar que no es la única manera de lograr el mapeo (véase por ejemplo las transformaciones conformes) pero nos permite entender los métodos directos que serán discutidos a lo largo de este trabajo.

1.4 Descripción del problema discreto

El problema discreto de la generación de mallas puede plantearse como la necesidad de construir una representación de la región Ω como una colección de partes más simples para su posterior análisis, por ejemplo los cuadriláteros, cuya unión resulte la región.

Definición 1.1 *Una malla para Ω es una colección de elementos c_i simples llamados celdas de la malla, con las propiedades siguientes*

1. $\bigcup c_i = \Omega$
2. $Int(c_i) \cap \partial\Omega = \emptyset$

Las celdas c_i pueden ser triángulos o cuadriláteros. Veamos algunos ejemplos en la Figura 1.7.

En la Figura 1.7 (a) la colección de triángulos cubre a Ω ; sin embargo, el orden de la colección hace deshechar ese tipo de mallas para resolver EDP por diferencias finitas, ya que resulta imprescindible contar con un mecanismo que nos permita tener fácil y rápido acceso a cada uno los elementos de la discretización, hecho que la colección en (a) no nos ofrece.

El orden de la colección en la Figura 1.7 (b) es directa debido a que las líneas que forman la colección de elementos han sido elegidos paralelos a los ejes coordenados. La dificultad es que la colección abarca más allá de la región Ω y resulta

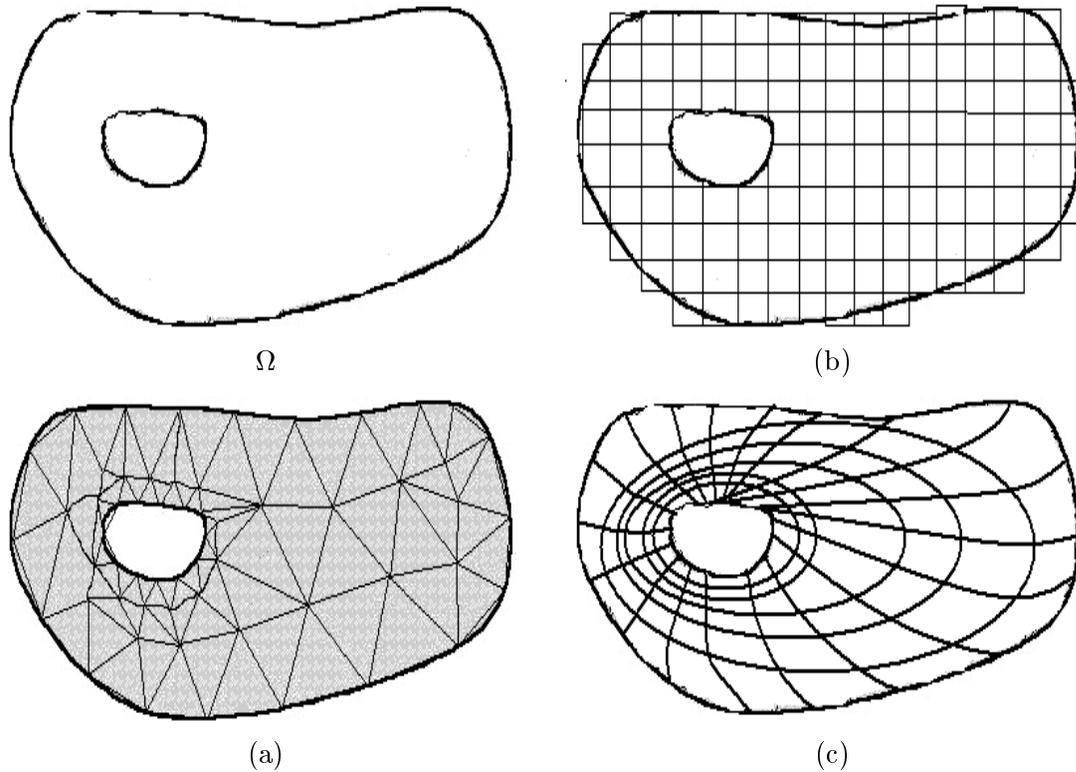


Figura 1.7: Algunas mallas sobre Ω .

indispensable resolver esta dificultad numérica con el fin de que la aproximación se realice únicamente sobre la región. En el caso particular de la solución numérica de ecuaciones diferenciales parciales, las condiciones a la frontera deben ser aproximadas y entonces surge la necesidad de definir de qué forma haremos la aproximación en la frontera y cual será la precisión de esta y desde luego, la manera en que será implantada en forma automática o semi-automática dentro de un sistema computacional lo que dificulta procesar problemas reales. La Figura 1.7 (c) es una forma idónea de discretizar la región con el fin de contar con una forma directa de numerar los nodos, así como cubrir toda la región de estudio. Este tipo de mallas reciben el nombre de *sistemas coordenados curvilíneos* ya que la idea es sumergir en un sistema curvilíneo la región para entonces y de forma directa, construir la malla. Serán este tipo de mallas las que nos interese construir, por otra parte, la forma discreta vendrá dada por la unión de celdas formadas por cuadriláteros.

Como se ha comentado al inicio del trabajo, la discretización de la región de estudio juega un papel importante cuando se desea resolver numéricamente un problema continuo sobre de ella, como lo es la solución numérica de alguna ecuación diferencial parcial. La discretización es un paso del trabajo a desarrollar sobre un problema

específico y es esencial cuando se desea evitar errores innecesarios de aproximación.

En muchas aplicaciones la región de trabajo Ω es solución de un problema diferencial o algebraico; en otros es posible contar con la descripción de la región en forma continua a través de una parametrización de las componentes o bien, del contorno mismo. En muchas más, sólo se dispone de puntos sobre el contorno obtenidos tras alguna observación.

Consideremos que el contorno de la región Ω está formado por un conjunto finito de puntos en el plano, con esto, el dominio a tratar es una región poligonal. Si (x_i, y_i) , para $i = 1, \dots, n$ es una colección de puntos en el plano, bajo una orientación, estamos interesados en estudiar la región poligonal que describe esos puntos

$$\partial\Omega = \text{polígono}((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), (x_1, y_1)). \quad (1.18)$$

La orientación del contorno es muy importante ya que nos describirá por una parte la forma de la región y por otra tendremos una referencia cuando señalemos las celdas dobladas. Los puntos del polígono bajo una orientación, la orientación positiva (el sentido contrario a las manecillas del reloj) será nuestra referencia de Ω en adelante, en cuyo caso Ω es una *región poligonal* formada por los segmentos generados por la unión orientada de los puntos en la frontera. Dos polígonos se muestran en la Figura 1.8.

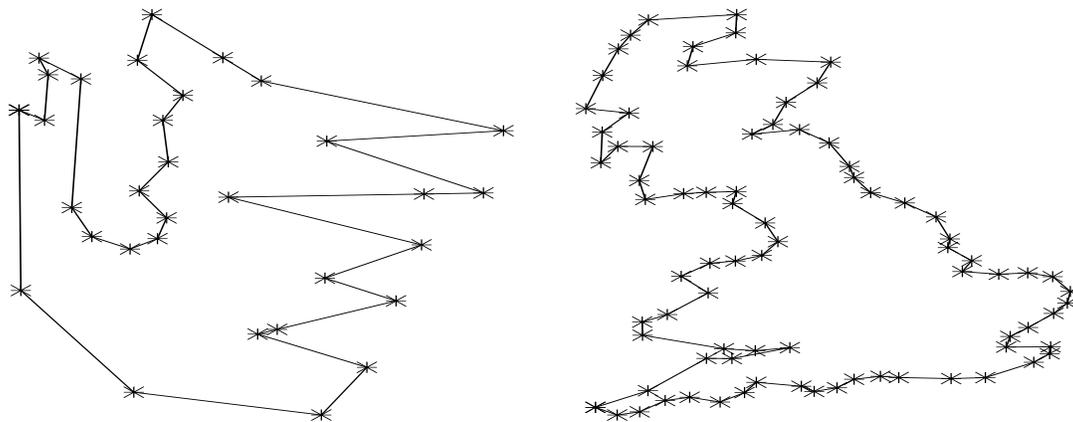


Figura 1.8: Ejemplo de regiones poligonales en el plano.

Una vez definida nuestra región de estudio planteamos de nueva cuenta el interés de este trabajo: **generar mallas suaves y convexas sobre regiones poligonales**.

La idea de la discretización se sigue del planteamiento continuo (1.7), lograr un mapeo entre el cuadrado unitario y ésta poligonal. Para lograr un mapeo con esta

condición es indispensable saber a dónde irán a ser mapeados los vértices del cuadrado unitario.

Un punto que no debemos pasar por alto es la elección de esos nodos de mapeo o de referencia del mapeo ya que de esto depende la obtención de la malla y desde luego esa elección dependerá de la observación requerida sobre la región; veamos lo siguiente. Consideremos por región poligonal al conjunto de puntos en el plano descrita en la Tabla 1.1

n	(x, y)
1	(0.0, 0.0)
2	(1.0, 0.0)
3	(1.0, 1.0)
4	(0.0, 1.0)
5	(0.0, 0.0)

Tabla 1.1: Puntos sobre la frontera del cuadrado unitario.

que corresponden al cuadrado unitario. Hemos repetido el punto inicial para reafirmar que trabajamos polígonos cerrados y acotados. La gráfica de esta región puede observarse a la izquierda de la Figura 1.9.

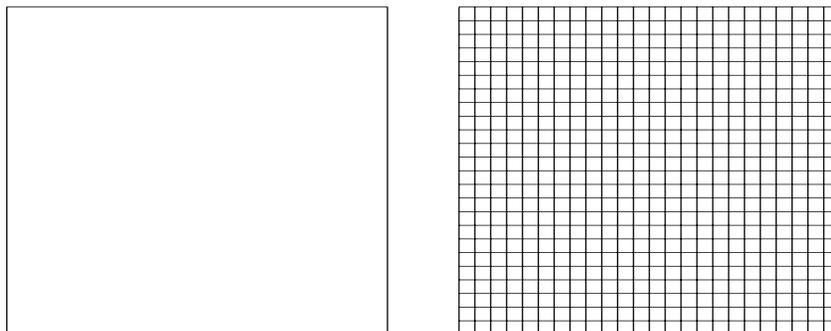


Figura 1.9: Cuadrado unitario y una malla sobre de él.

Una malla sugerente para esta región vendría dada al unir los segmentos opuestos, esto es, "tirar" línea entre los segmentos o subfronteras del cuadrado. La representación continua del mapeo es la interpolación lineal entre los segmentos.

Sin embargo, no es la única forma de obtener un mapeo entre regiones, ya que no es la única forma de construir una malla sobre el cuadrado unitario. Consideremos los puntos sobre la frontera del cuadrado unitario dado por la colección señalada en la Tabla 1.2.

n	(x, y)
1	(0, 0.0)
2	(0.25, 0.0)
3	(1.0, 0.0)
4	(1.0, 0.75)
5	(1.0, 1.0)
6	(0.0, 1.0)
7	(0.0, 0.0)

Tabla 1.2: Puntos sobre la frontera del cuadrado unitario. Otra distribución.

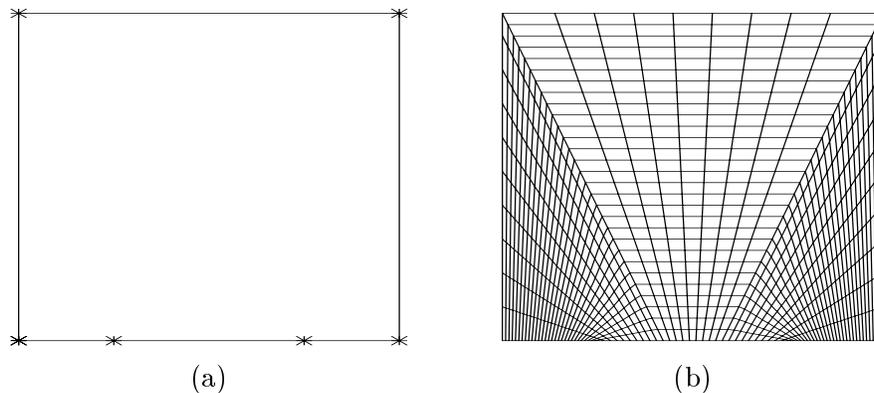


Figura 1.10: (a) Asignación de los segmentos de frontera. (b) Malla obtenida por interpolación bilineal.

Esta colección representa de nueva cuenta al cuadrado unitario. Consideremos ahora que el mapeo de \mathbf{B}_2 hacia Ω lo construimos en la forma siguiente: que el segmento “de abajo” de \mathbf{B}_2 es mapeado al segmento entre $(0.25, 0)$ y $(0.75, 0.0)$, que el segmento “derecho” de \mathbf{B}_2 es mapeado al “segmento” de frontera compuesto por los puntos $(0.75, 0.0)$ y $(1.0, 0.0)$; de la misma forma, ahora consideremos que el segmento “de arriba” de \mathbf{B}_2 sea mapeado al segmento entre $(1.0, 0.0)$, $(1.0, 1.0)$, $(0.0, 1.0)$ y $(0.0, 0.0)$; y por último que el segmento “izquierdo” de \mathbf{B}_2 sea mapeado al “segmento” de frontera comprendido por los puntos $(0.0, 0.0)$ y $(0.25, 0.0)$. Esta asignación la representamos en la Figura 1.10(a).

Una interpolación entre las líneas nos lleva a considerar la malla que tras ésta elección de puntos hemos hecho, dicha malla la visualizamos en la Figura 1.10(b).

La malla de la Figura 1.10(b) surge de la interpolación bilineal y de la asignación entre segmentos elegida. Lo importante de este ejemplo es mostrar que sobre una misma región es posible considerar multiples mapeos y lograr mallas dependiendo de la distribución elegida de los segmentos de fronteras que hemos de considerar

para la malla estructurada. Una elección adecuada es esencial para el problema que necesitamos resolver. Por ejemplo, si es un problema de flujo y nos interesa medir con precisión las fuentes, debe tenerse en cuenta este hecho para construir la malla.

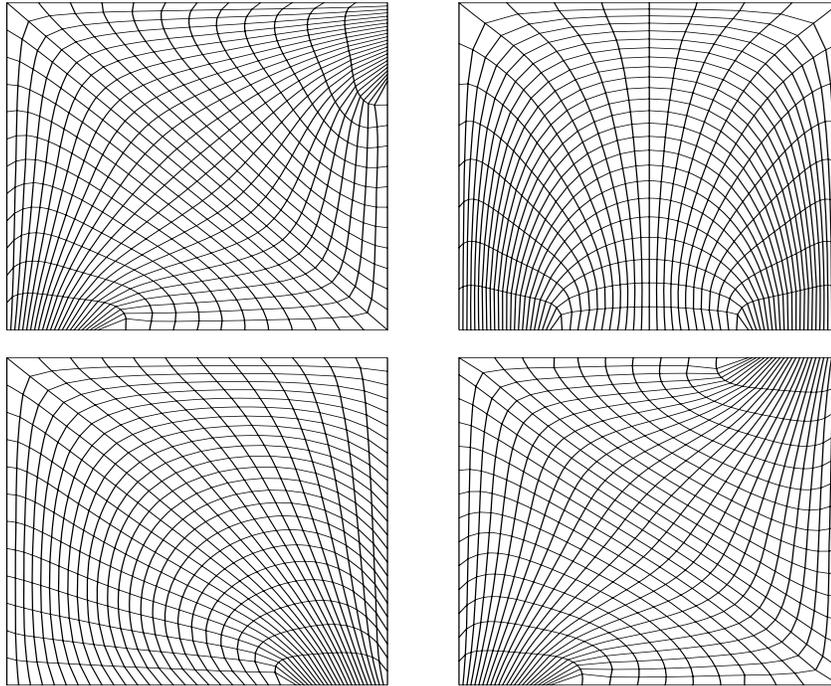


Figura 1.11: Distintas configuraciones de mallas sobre el cuadrado unitario.

En la Figura 1.11 mostramos otras mallas obtenidas de distintas distribuciones de los segmentos de frontera.

Como se observa de la Figura 1.11, sobre una misma región podemos obtener distintas configuraciones de mallas. En muchas aplicaciones, dependiendo del problema a resolver deberá considerarse el tipo de configuración necesaria sobre la región.

Otra forma directa de obtener una malla, es considerar un método algebraico. Los métodos algebraicos se basan en considerar la parametrización de los segmentos de frontera, sea discretos o continuos y sobre ellos considerar una interpolación entre fronteras, ver [47] y [19]. Un ejemplo típico de un mapeo algebraico para la generación de mallas está dado por el método TFI [40]. A continuación comentaremos muy brevemente sobre él.

Supongamos que contamos con una descripción de la frontera Ω a través de los cuatro segmentos de frontera, véase la Figura 1.12, y todas ellas parametrizadas en la forma

$$\begin{aligned} \mathbf{x}_b(\xi), \quad \mathbf{x}_t(\xi), \quad 0 \leq \xi \leq 1 \\ \mathbf{x}_l(\eta), \quad \mathbf{x}_r(\eta) \quad 0 \leq \eta \leq 1 \end{aligned}$$

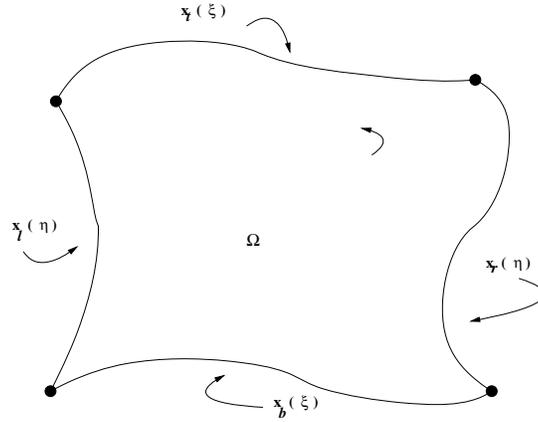


Figura 1.12: Segmentos de frontera de $\partial\Omega$.

con la propiedad de continuidad del mapeo; es decir, que las parametrizaciones se empaten

$$\mathbf{x}_b(0) = \mathbf{x}_l(0) \quad \mathbf{x}_b(1) = \mathbf{x}_r(0) \quad \mathbf{x}_r(1) = \mathbf{x}_t(1) \quad \mathbf{x}_t(0) = \mathbf{x}_l(1) \quad (1.19)$$

El mapeo proveniente de la interpolación transfinita TFI se escribe como

$$\begin{aligned} \mathbf{x}(\xi, \eta) = & (1 - \eta)\mathbf{x}_b(\xi) + \eta\mathbf{x}_t(\xi) + (1 - \xi)\mathbf{x}_l(\eta) + \xi\mathbf{x}_r(\eta) - \\ & [\xi\eta\mathbf{x}_t(1) + \xi(1 - \eta)\mathbf{x}_b(1) + \eta(1 - \xi)\mathbf{x}_l(0) + (1 - \xi)(1 - \eta)\mathbf{x}_b(0)] \end{aligned} \quad (1.20) \quad (1.21)$$

El cual es un mapeo muy elemental descrito a partir de las fronteras. Pero por un lado, es un mapeo que propaga las discontinuidades diferenciales de la frontera hacia el interior de la región [47] y por consiguiente, al usarlo sobre regiones irregulares el resultado es una malla muy “doblada”.

En la práctica, cuando contamos con una región plana $\Omega \subset \mathbb{R}^2$, por lo general únicamente contamos con una discretización de su frontera $\partial\Omega$ a través de una colección orientada de puntos, ver Figura 1.14(a) y sobre esa región poligonal cerrada hemos de construir el mapeo TFI. El paso siguiente necesariamente es parametrizar los cuatro segmentos de frontera $\mathbf{x}_b(\xi)$, $\mathbf{x}_l(\eta)$, $\mathbf{x}_t(\xi)$ y $\mathbf{x}_r(\eta)$.

La forma usual de lograr la parametrización es teniendo por parámetro la longitud de arco. Contando con la parametrización de los segmentos de frontera se procede

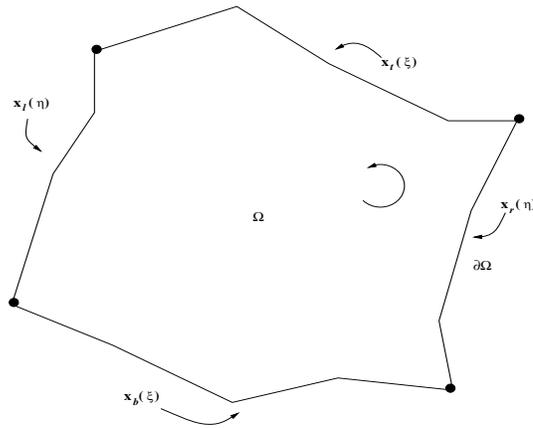


Figura 1.13: Construcción de una malla a partir de los puntos sobre la frontera.

a obtener una colección de puntos tratando de que sean igualmente espaciados pero respetando la forma del contorno. En el trabajo de Ojeda [37] se describe un criterio para respetar la forma del contorno, y entonces aplicar el mapeo TFI sobre esa colección de puntos, ver Figura 1.14(c).

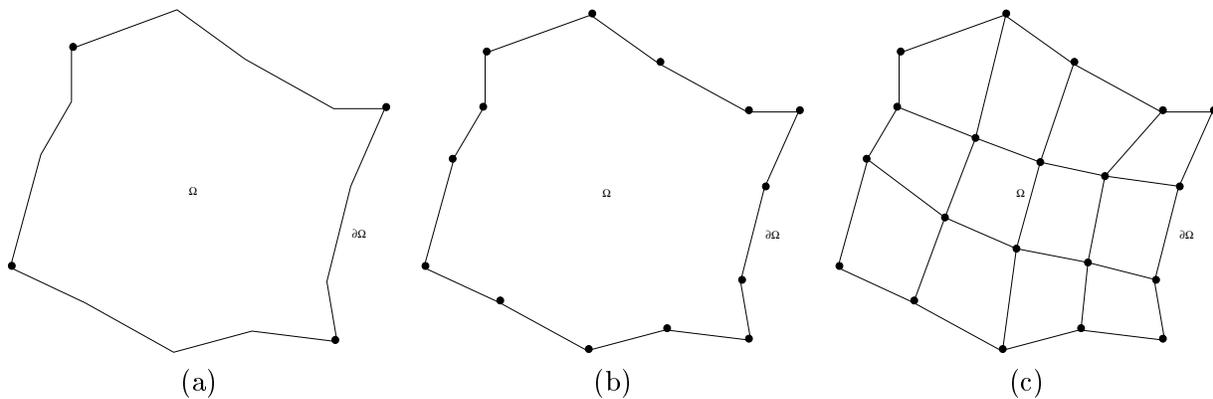


Figura 1.14: Construcción de una malla a partir de los puntos sobre la frontera.

En la Figura 1.15 se muestran algunas mallas provenientes del método TFI sobre distintas regiones. En cada una de ellas se observa la propagación de las singularidades de la frontera hacia el interior.

Como se ha apreciado con los ejemplos anteriores, la construcción de una malla sobre una región no es única y depende en mucho de la distribución de los segmentos de frontera, y desde luego la propiedad o propiedades geométricas a observar dentro de la malla.

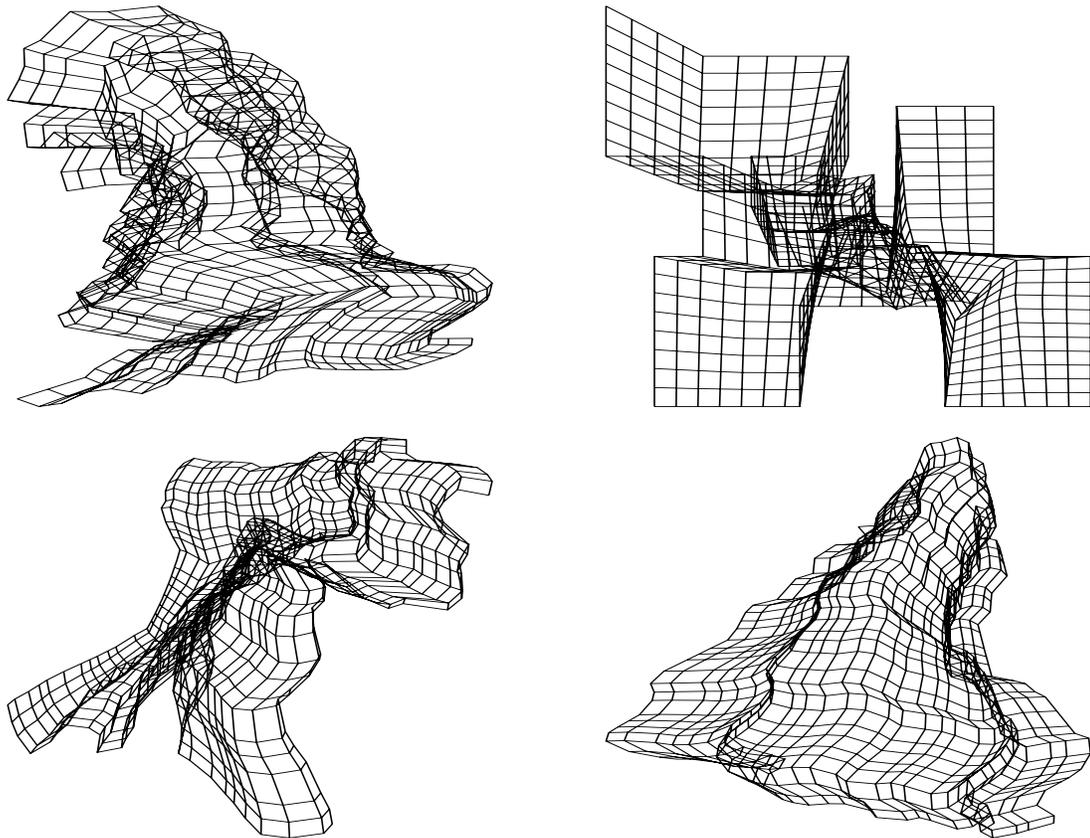


Figura 1.15: Algunas mallas construidas vía el método algebraico TFI.

Capítulo 2

Funcionales Armónicos continuos y discretos

La construcción de mallas a través de mapeos armónicos se inició con el estudio del operador elíptico más sencillo que sobre una región puede definirse. Describiremos cómo ha evolucionado su uso tanto en el caso continuo como en el discreto.

2.1 El sistema elíptico más simple

Consideremos una parametrización de la frontera de la región de estudio Ω definida por cuatro segmentos de forma que: $\mathbf{x}_b = \mathbf{x}_b(\xi)$ representa el segmento de frontera de “abajo”; $\mathbf{x}_r = \mathbf{x}_r(\eta)$ representa el segmento de frontera de la “derecha”; $\mathbf{x}_t = \mathbf{x}_t(\xi)$ representa el segmento de frontera de “arriba” y $\mathbf{x}_l = \mathbf{x}_l(\eta)$ el segmento de la frontera de la “izquierda”, ver Figura 2.1. La notación de los segmentos de frontera seguirá la abreviatura $b=bottom$, $t=top$, $l=left$ y $r=right$.

Dada la región Ω y el cuadrado unitario B_2 estamos interesados en construir mapeos directos entre ellos, ver Figura 2.2.

Una de las primeras formas que se eligieron para la construcción del mapeo directo fue pedir que las componentes del mapeo satisfagan la ecuación elíptica más simple que es

$$\begin{aligned}x_{\xi\xi} + x_{\eta\eta} &= 0 \\y_{\xi\xi} + y_{\eta\eta} &= 0\end{aligned}\tag{2.1}$$

sujeta a que las fronteras se correspondan; es decir,

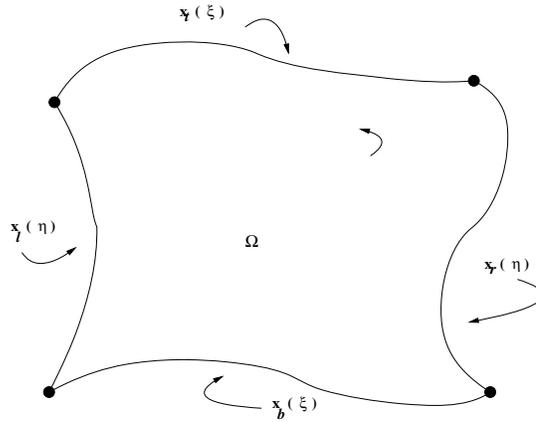


Figura 2.1: Segmentos de frontera de $\partial\Omega$.

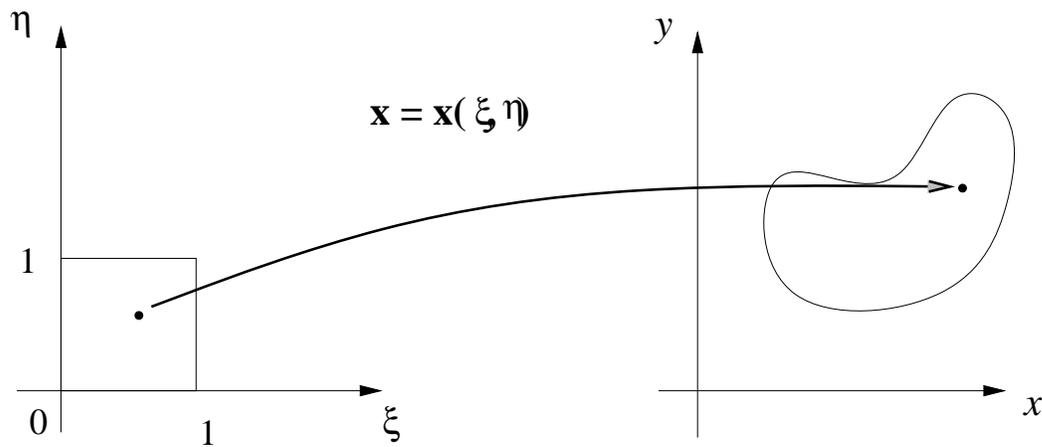


Figura 2.2: Mapeo directo entre B_2 y Ω .

$$\begin{aligned}
 (x(\xi, 0), y(\xi, 0)) &= \mathbf{x}_b(\xi), \\
 (x(0, \eta), y(0, \eta)) &= \mathbf{x}_l(\eta), \\
 (x(\xi, 1), y(\xi, 1)) &= \mathbf{x}_t(\xi), \\
 (x(1, \eta), y(1, \eta)) &= \mathbf{x}_r(\eta).
 \end{aligned} \tag{2.2}$$

El problema tiene una única solución infinitamente diferenciable hacia el interior de Ω siempre que el mapeo entre fronteras sea continuo; lo que es cierto en nuestro caso. Para regiones convexas no hay nada que decir, la malla así generada es convexa.

Por una parte, el mapeo solución de (2.1) es un mapeo suave en Ω pero sobre regiones no convexas la malla puede resultar doblada; esto es, con muchas celdas no convexas. Tomemos la transformación descrita por Ivanenko[28]:

$$x(\xi, \eta) = \frac{1}{2}(\xi^2 - \eta^2) - \frac{2}{3}\xi \quad (2.3)$$

$$y(\xi, \eta) = \xi\eta + \frac{1}{2}\xi - \frac{1}{3}\eta \quad (2.4)$$

La representación gráfica del mapeo sobre el cuadrado unitario puede observarse en la Figura 2.3.

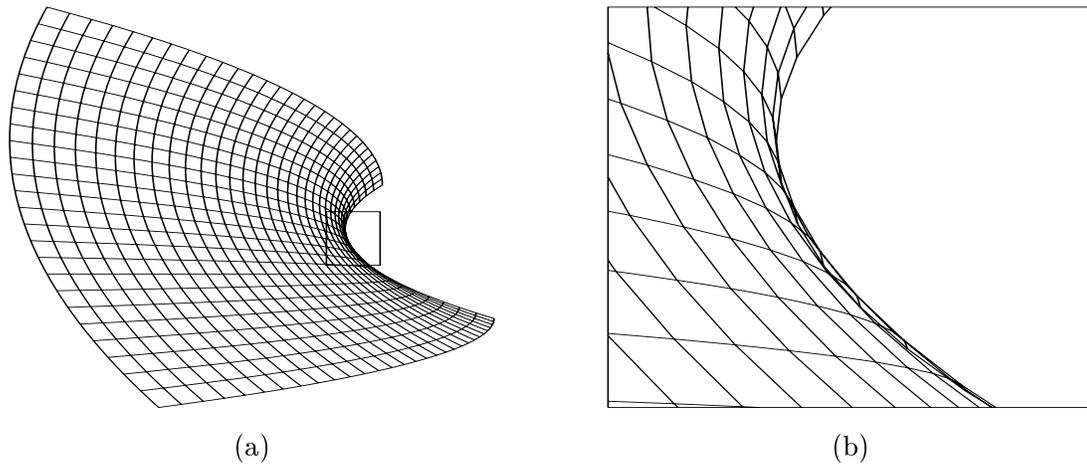


Figura 2.3: Ejemplo de un sistema coordenado curvilíneo doblado.

Inmediatamente podemos comprobar que el mapeo satisface la ecuación de Laplace

$$\begin{aligned} x_{\xi\xi} + x_{\eta\eta} &= 0 \\ y_{\xi\xi} + y_{\eta\eta} &= 0. \end{aligned} \quad (2.5)$$

Sin embargo, al analizar el jacobiano de la transformación tenemos que

$$\begin{aligned} J(\xi, \eta) &= x_{\xi}y_{\eta} - x_{\eta}y_{\xi} \\ &= \left(\xi - \frac{2}{3}\right)\left(\xi - \frac{1}{3}\right) + \eta\left(\eta + \frac{1}{2}\right) \end{aligned} \quad (2.6)$$

con lo que $J(\xi, \eta) < 0$ para $\eta = 0$ y $1/3 < \xi < 2/3$, así que un segmento de las líneas cercanas a la frontera de abajo de B_2 al ser mapeadas bajo \mathbf{x} resultan ser segmentos curvilíneos que se doblan, ver Figura 2.3(b). Lo interesante de este ejemplo es que el

mapeo solución a la ecuación de Laplace produce mallas dobladas en ésta región que es muy simple; los segmentos de frontera son parábolas.

En el caso de regiones un poco irregulares, usando solamente este generador, el operador de Laplace, las líneas resultan doblarse debido a la curvatura o picos de la región, véase la Figura 2.4. En el caso de regiones poligonales es más fácil ver este problema si la región no es convexa.

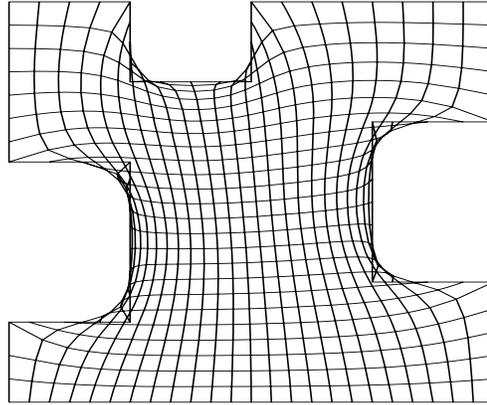


Figura 2.4: Operador de Laplace resuelto sobre la región M19

Por su simplicidad para generar líneas suaves, la ecuación de Laplace da la pauta para que, mediante condiciones adecuadas a la frontera, pueda emplearse para construir sistemas curvilíneos con un comportamiento deseado. Por ejemplo, que las líneas que parten de una de las fronteras sean ortogonales y que hacia el interior se satisfaga la ecuación de Laplace, véase la Figura 2.5.

En [41], Khamaryseh, Kuprat y Mastin, tratan el mismo problema del operador elíptico pero considerando condiciones de control en la frontera y condiciones de ortogonalidad. Spekrijse [45] describe algunas transformaciones, mapeos de control, para medir y controlar los segmentos curvilíneos y el espaciamiento de los puntos de la red. Esto lo hace mediante la idea de introducir un sistema de referencia y sobre de él generar una distribución de las líneas de manera que la composición entre mapeos genere la propiedad que desea: ortogonalidad en la frontera.

Uno de los atractivos de la obtención de mallas a través de la ecuación de Laplace, es que se pueden interpretar en forma variacional ya que las ecuaciones de Laplace son las ecuaciones de Euler–Lagrange del funcional

$$I_L = \int_0^1 \int_0^1 (x_\xi^2 + y_\xi^2 + x_\eta^2 + y_\eta^2) d\xi d\eta \quad (2.7)$$

sujeto a las correspondientes condiciones de frontera. De aquí nació el enfoque varia-

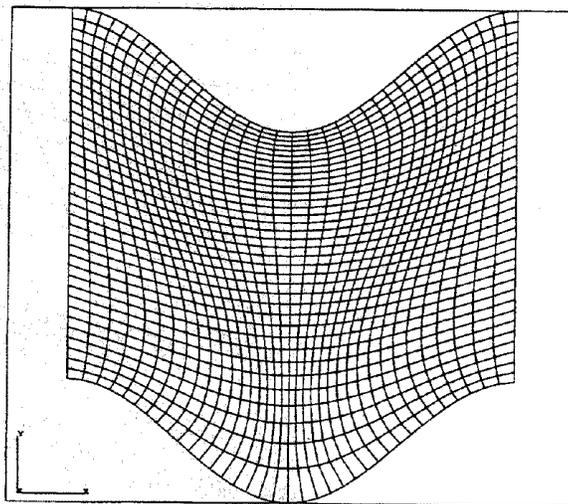


Figura 2.5: Los segmentos curviíneos son casi-ortogonales en las fronteras de arriba y abajo, tomado de Spekreijse [45].

cional para la generación de mallas pero como veremos, la construcción numérica de las mallas resulta diferente.

Una de las propiedades geométricas que nos interesa es la convexidad, ya que aquellas transformaciones que doblan las líneas no son adecuadas en la mayoría de las aplicaciones prácticas. Como se sabe la convexidad de las celdas está ligada al jacobiano de la transformación, ver González [24].

Veamos cómo se ha usado el operador de Laplace y sus variantes.

Consideremos el mapeo inverso, ver Figura 2.6, para el cual pediremos que las componentes

$$\xi = \xi(x, y), \quad \eta = \eta(x, y) \tag{2.8}$$

minimicen el funcional

$$I = \int_{\Omega} (\xi_x^2 + \eta_x^2 + \xi_y^2 + \eta_y^2) \, dx dy \tag{2.9}$$

otra vez, el mínimo se encuentra en los mapeos armónicos $\xi(x, y), \eta(x, y)$ de Ω :

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= 0, \\ \eta_{xx} + \eta_{yy} &= 0 \end{aligned} \tag{2.10}$$

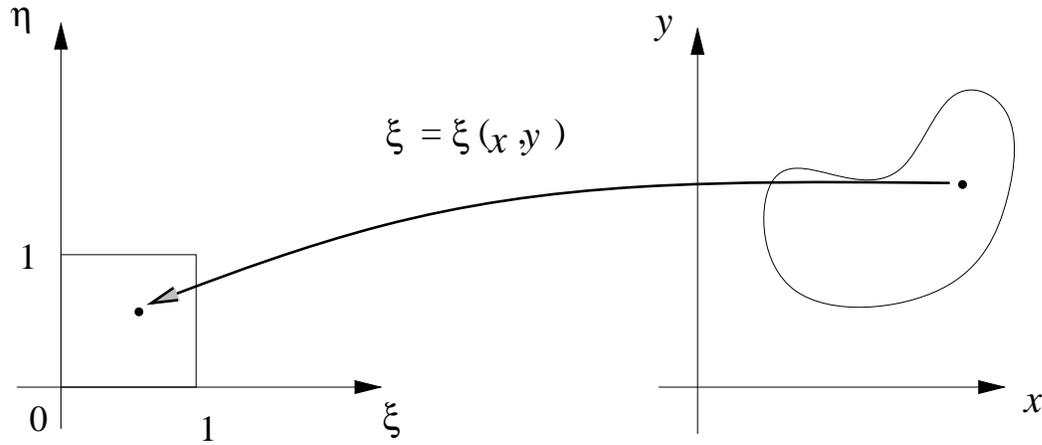


Figura 2.6: Mapeo inverso entre B_2 y Ω .

sobre el cuadrado unitario. Las condiciones de Dirichlet para esta ecuación están dadas por el mapeo inverso de las fronteras. Con la propiedad de que el mapeo inverso sea armónico es posible demostrar que es uno a uno y sobre, Ivanenko [28]. Por lo tanto, la transformación de Winslow produce mallas no dobladas.

Las funciones que satisfagan las ecuaciones de Laplace se les llama funciones armónicas; y al mapeo obtenido con el funcional de Winslow se le llama *mapeo armónico*.

Definición 2.1 *Un mapeo armónico es una transformación uno a uno de Ω sobre el cuadrado unitario B_2 de manera que cumple las condiciones*

- a) *La frontera de Ω es mapeada sobre la frontera de B_2*

$$\mathbf{x}^{-1}(\partial\Omega) = \partial B_2$$

- b) *Las esquinas de Ω , empates de las secciones de frontera en un sentido (por ejemplo; el sentido positivo, contrario a las manecillas del reloj), son mapeadas sobre las esquinas correspondientes de B_2*

$$\mathbf{x}_l(1) = \mathbf{x}_b(0) = (0, 0)^t$$

$$\mathbf{x}_b(1) = \mathbf{x}_r(0) = (1, 0)^t$$

$$\mathbf{x}_r(1) = \mathbf{x}_t(0) = (1, 1)^t$$

$$\mathbf{x}_t(1) = \mathbf{x}_l(0) = (0, 1)^t$$

- c) *Las dos componentes del mapeo son funciones armónicas hacia en interior de Ω*

Debido a que las ecuaciones de Winslow se definen sobre la región Ω ; tendremos que para regiones arbitrarias aún con frontera suave resulta muy costoso resolverlas.

Para lograr un procedimiento confiable, Roache y Steinberg [42] proponen usar la idea de Brackbill y Saltzman[11] de transformar las ecuaciones (2.10) al espacio lógico y resolverlas sobre el cuadrado unitario B_2 .

Si hacemos el cambio de variables al espacio lógico en la forma

$$\begin{aligned} \xi_x &= \frac{y_\eta}{J} & \xi_y &= \frac{x_\eta}{J} \\ \eta_x &= \frac{y_\xi}{J} & \eta_y &= \frac{x_\xi}{J} \end{aligned} \quad (2.11)$$

con $J \neq 0$. Tenemos ahora el problema de encontrar el mapeo directo $\mathbf{x} : B_2 \rightarrow \Omega$ que minimiza a la funcional

$$I_S(x) = \int_{B_2} \frac{x_\xi^2 + y_\xi^2 + x_\eta^2 + y_\eta^2}{J} d\xi d\eta \quad (2.12)$$

sujeto a las condiciones de frontera correspondientes, y que cumple la condición

$$J = (x_\xi y_\eta - x_\eta y_\xi) > 0 \quad 0 < \xi, \eta < 1. \quad (2.13)$$

Las ecuaciones de Euler–Lagrange para este caso son:

$$\begin{aligned} g_{22}x_{\xi\xi} - 2g_{12}x_{\xi\eta} + g_{11}x_{\eta\eta} &= 0 \\ g_{22}y_{\xi\xi} - 2g_{12}y_{\xi\eta} + g_{11}y_{\eta\eta} &= 0 \end{aligned} \quad (2.14)$$

donde

$$\begin{aligned} g_{11} &= x_\xi^2 + y_\xi^2, \\ g_{12} &= x_\xi x_\eta + y_\xi y_\eta, \\ g_{22} &= x_\eta^2 + y_\eta^2 \end{aligned} \quad (2.15)$$

y por otra parte, el Jacobiano de la transformación es

$$\mathcal{G} = g_{11}g_{22} - g_{12}^2. \quad (2.16)$$

Roache y Steiberg emplean un esquema en diferencias finitas centradas y, a través de un algoritmo iterativo SOR con relajación resuelven numéricamente las ecuaciones (2.14). A éste esquema lo llamaron TTM

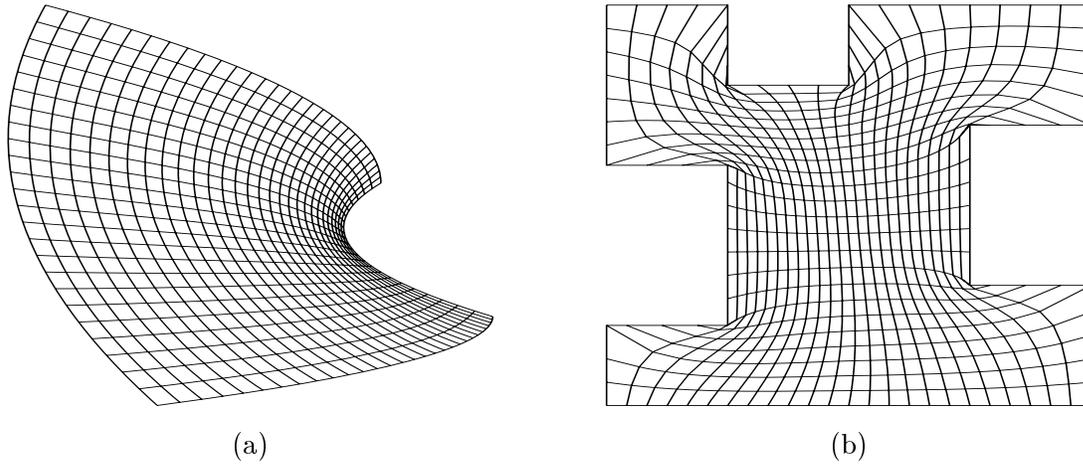


Figura 2.7: (a) Malla convexa generada por TTM. b) Malla no convexa sobre M19 generada por TTM.

Sin embargo, como bien hacen notar Roache y Steinberg [43], la solución numérica de estas ecuaciones están regidas por errores de truncación. Así, la solución obtenida de la discretización puede ser una malla doblada aún para regiones no difíciles, ver Figura 2.7(b). En la Figura 2.8, la región de la bahía de la Habana, una de las regiones más irregulares en su frontera que las que hemos estudiado, se observa cómo la solución obtenida con la discretización de Steinberg falla.

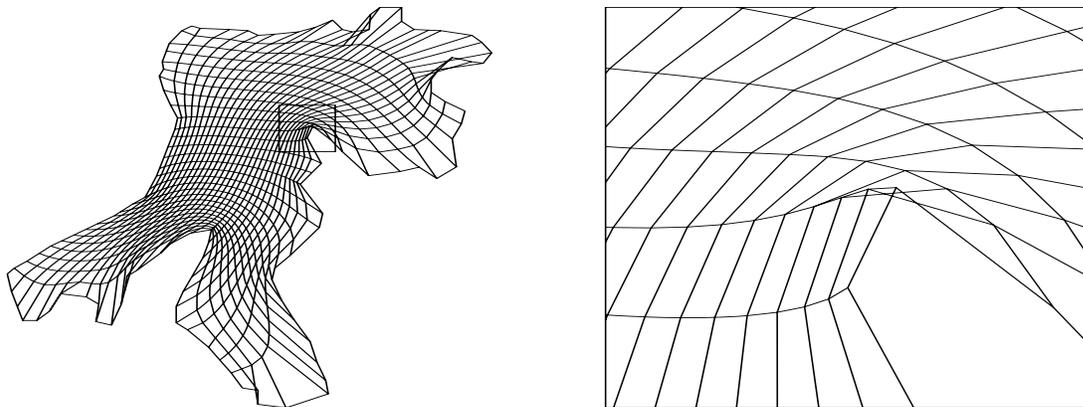


Figura 2.8: Malla doblada sobre la región de la bahía de la Habana obtenida por TTM.

Las regiones no regulares, como la región de la bahía de la Habana son de especial interés. Nos abocaremos a construir mallas no dobladas sobre la región, con propiedades interesantes entre sus celdas y segmentos de línea. Para lograrlo, seguiremos la idea de Ivanenko de usar la forma variacional discreta del problema de Winslow.

2.2 Funcionales armónicos discretos

Describiremos una forma general para construir funcionales discretos, usando como modelo al funcional de Winslow. Consideremos la región Ω definida a través de una colección de puntos sobre su frontera $\partial\Omega$ que la definen completamente; es decir

$$\partial\Omega = \text{polígono}(P_1, P_2, \dots, P_r, P_1) \tag{2.17}$$

Con base a esta suposición, nuestra región es un polígono cerrado de ahí, que el primer punto para su descripción sea considerado en dos ocasiones. En la Figura 2.9, se muestra esta numeración.

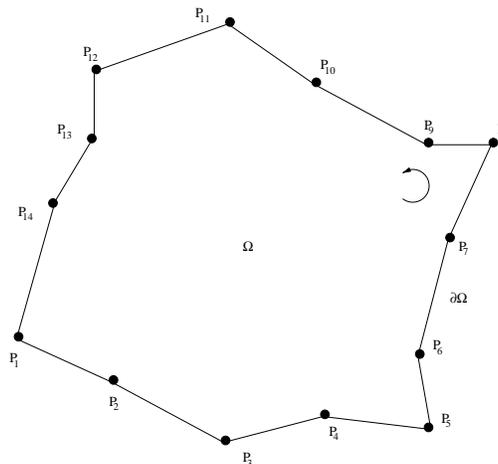


Figura 2.9: Ejemplo de región poligonal de trabajo y sus puntos a la frontera.

Ahora bien, podemos definir a una malla sobre Ω como una colección de puntos en la forma:

Definición 2.2 Una malla G de dimensión $m \times n$ sobre la región poligonal Ω es una colección

$$G = \{P_{i,j} | i = 1, \dots, m, j = 1, \dots, n\} \tag{2.18}$$

de puntos en el plano tales que $\partial\Omega \subset G$. Tomemos en cuenta que $m, n \in \mathbb{N} - \{1, 2\}$.

Los puntos $P_{i,j}$ son los *puntos o nodos* de la malla, incluidos los de la frontera de Ω . En la Figura 2.10 se muestra un ejemplo de ésta numeración que se precisará a continuación.

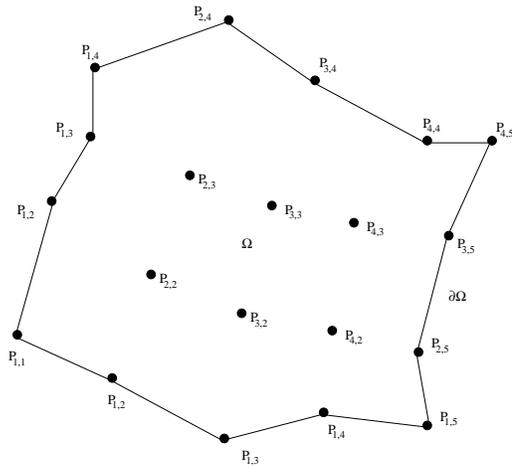


Figura 2.10: Una malla para $\partial\Omega$ de 5×4 .

Bajo la idea de construir un mapeo entre el cuadrado unitario y Ω , estamos interesados en denotar gráfica y analíticamente la correspondencia del mapeo. Para esto, debemos indicar cuales representarán a los lados del cuadrado que son mapeados a $\partial\Omega$.

La forma práctica es darle un sentido de orientación, el sentido positivo de una curva (contrario a las manecillas del reloj) e indicar quienes serían los lados hacia “abajo”, “derecha”, “arriba” e “izquierda” de $\partial\Omega$ numerándolos de manera que podamos identificar los segmentos curvilíneos que unen a cada uno de los lados opuestos por “verticales” y “horizontales” atendiendo a la orientación.

Bajo estas ideas definiremos la parte de frontera de Ω de “abajo”, la correspondiente a los puntos $\{(\xi, 0) \mid 0 \leq \xi \leq 1\}$ bajo el mapeo directo, como la colección

$$\{P_{1,1}, P_{2,1}, \dots, P_{m,1}\} \quad (2.19)$$

y la parte de “arriba” de la frontera de Ω , la correspondiente a los puntos $\{(\xi, 1) \mid 0 \leq \xi \leq 1\}$ bajo el mapeo directo, como

$$\{P_{m,n}, P_{m-1,n}, \dots, P_{1,n}\} \quad (2.20)$$

En la misma forma, la parte de la “izquierda” de $\partial\Omega$, la correspondiente a los puntos $\{(0, \eta) \mid 0 \leq \eta \leq 1\}$ bajo el mapeo directo, sería

$$\{P_{1,1}, P_{1,2}, \dots, P_{1,n}\} \tag{2.21}$$

y la parte de la “derecha”, la correspondiente a los puntos $\{(1, \eta) | 0 \leq \eta \leq 1\}$ bajo el mapeo directo, sería

$$\{P_{m,1}, P_{m,2}, \dots, P_{m,n}\} \tag{2.22}$$

quedando plenamente identificadas las “esquinas” de la frontera por los puntos

$$P_{1,1}, P_{m,1}, P_{m,n}, P_{1,n} \tag{2.23}$$

En la Figura 2.11 puede observarse gráficamente éstas consideraciones.

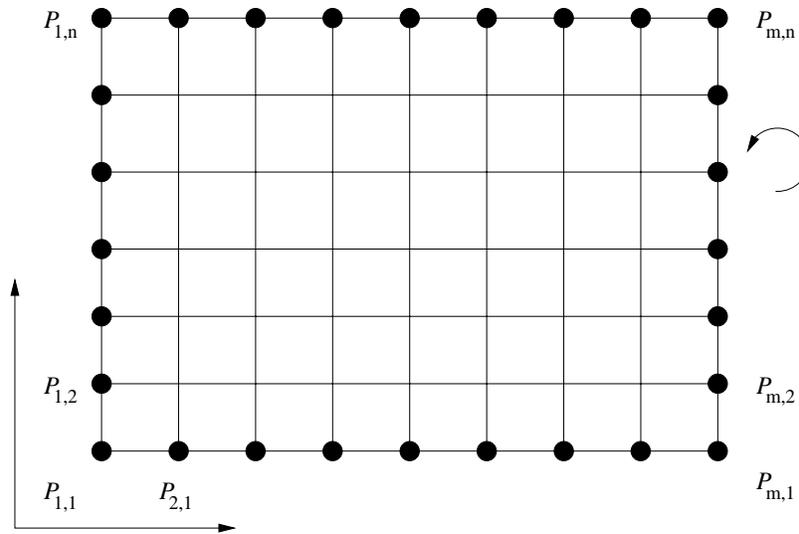


Figura 2.11: Numeración de los segmentos de frontera.

Así, las líneas que conectan los segmentos de frontera “abajo” con “arriba” las llamaremos líneas “verticales” de la malla, con esto, la i -ésima línea será

$$\{P_{i,1}, P_{i,2}, \dots, P_{i,n}\} \tag{2.24}$$

y la j -ésima línea “horizontal” que une los segmentos de frontera “izquierda” con “derecha” será

$$\{P_{1,j}, P_{2,j}, \dots, P_{m,j}\} \tag{2.25}$$

En la Figura 2.12 se han representado tales segmentos curvilíneos así como los puntos hacia el interior.

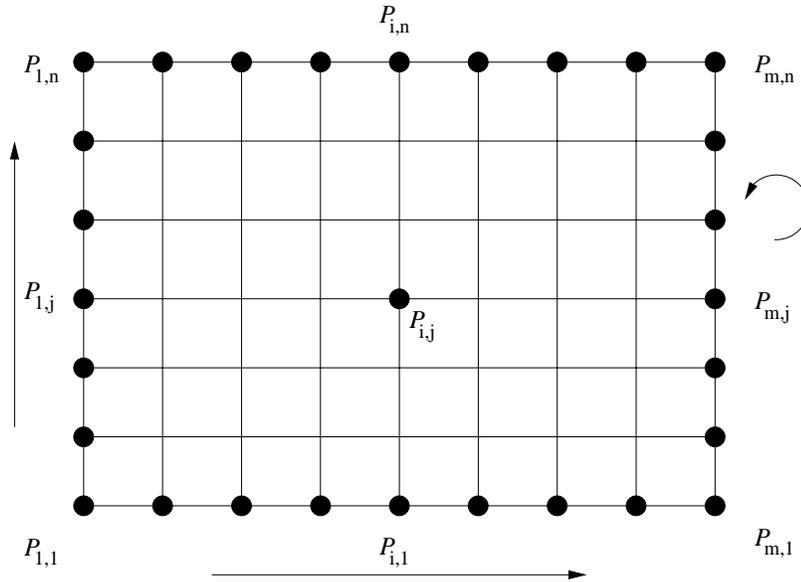


Figura 2.12: Segmentos curvilíneos “horizontales” y “verticales”

El número de puntos o nodos interiores de la malla son $(m - 2)(n - 2)$

$$P_{i,j}, \quad 1 < i < m; \quad 1 < j < n, \quad (2.26)$$

cada uno de los cuales tiene cuatro celdas alrededor de él, y por consiguiente una fuerte dependencia entre ellas. Véase la Figura 2.13.

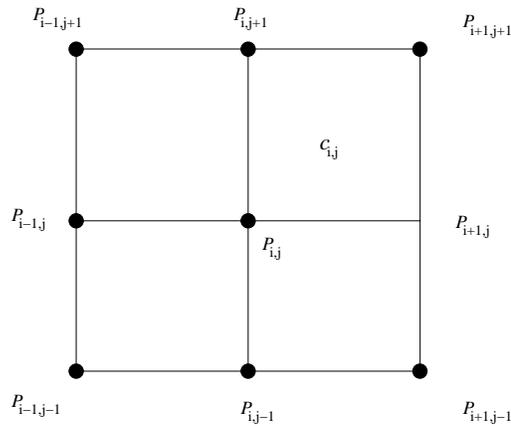


Figura 2.13: Celdas alrededor del punto $P_{i,j}$

Obsérvese que la celda $c_{i,j}$ esta formada por los puntos:

$$P_{i,j}, P_{i+1,j}, P_{i+1,j+1}, P_{i,j+1} \quad (2.27)$$

y hay $(m-1)(n-1)$ celdas en la malla. Cada punto interior $P_{i,j}$ cuenta con 8 puntos vecinos, ver Figura 2.13. Esta observación será considerada posteriormente cuando veamos cómo describir los funcionales y algunas de sus propiedades relacionados con los puntos interiores.

Una vez definida una malla discreta sobre una región poligonal retomemos el funcional de Winslow y veamos una discretización de él sobre la malla.

2.3 Una discretización del funcional de Winslow

Consideremos al funcional continuo de Winslow tal como lo hemos escrito o bien en su forma vectorial

$$I(\mathbf{x}) = \int_{B_2} \frac{x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2}{J} d\xi d\eta = \int_{B_2} \frac{\|\mathbf{x}_\xi\|^2 + \|\mathbf{x}_\eta\|^2}{\mathbf{x}_\xi^t J_2 \mathbf{x}_\eta} d\xi d\eta \quad (2.28)$$

con

$$J_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad (2.29)$$

el funcional está definido sobre mapeos que correspondan la frontera de B_2 con $\partial\Omega$, por lo que podemos ver la malla que se va a construir en Ω como la determinada en B_2 y transformada por el mapeo solución al funcional de Winslow, ver Figura 2.14.

Ahora, restrinjamos el mapeo \mathbf{x} a una de las celdas de B_2 . Si $B_{i,j}$ es la celda en B_2 que será mapeada a $c_{i,j}$, tendremos que

$$\mathbf{x}(B_{i,j}) = c_{i,j} \quad (2.30)$$

con esto, podremos reescribir al funcional como una suma de funcionales sobre cada una de las celdas: la restricción del funcional de Winslow sobre cada celda:

$$I(\mathbf{x}) = \sum \int_{B_{i,j}} \frac{\|\mathbf{x}_\xi\|^2 + \|\mathbf{x}_\eta\|^2}{\mathbf{x}_\xi^t J_2 \mathbf{x}_\eta} d\xi d\eta \quad (2.31)$$

así, la idea ahora es aproximar cada integral por medio de una cuadratura para obtener un función que sólo dependa de los puntos interiores. El proceso de aproximación se realizará de la manera siguiente:

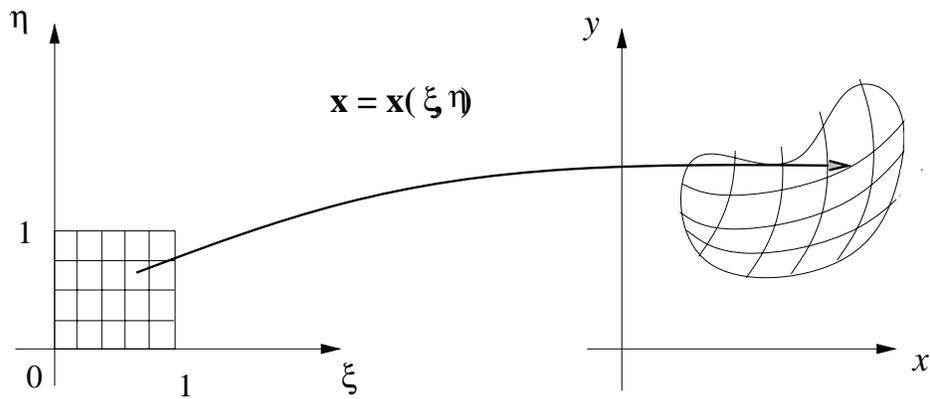


Figura 2.14: Construcción de una malla en Ω al transformar una malla en B_2 a través del mapeo directo \mathbf{x} .

En cada celda $B_{i,j}$ el mapeo de \mathbf{x} de esta celda hacia el cuadrilátero $c_{i,j}$, se aproxima por la transformación más simple que es el mapeo bilineal

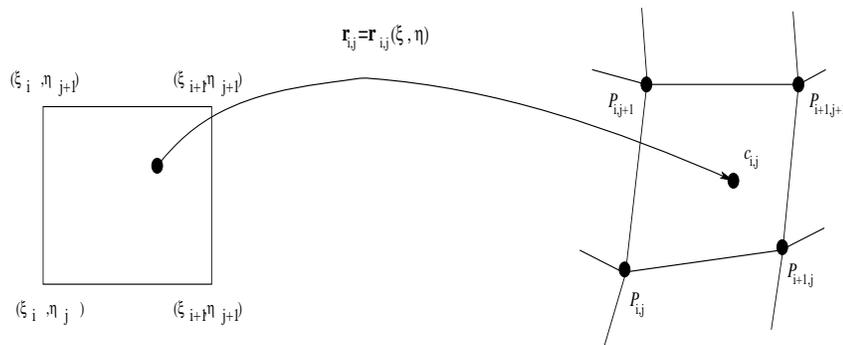


Figura 2.15: Mapeo bilineal entre $B_{i,j}$ y $c_{i,j}$

$$\mathbf{r}_{i,j}(\xi, \eta) = \mathbf{A} + \mathbf{B}\xi + \mathbf{C}\eta + \mathbf{D}\xi\eta, \quad (2.32)$$

ver Figura 2.15, sujeto a las condiciones de interpolación entre los cuadriláteros (2.30). Esto es,

$$\mathbf{x}|_{B_{i,j}} \approx \mathbf{r}_{i,j} \quad (2.33)$$

entonces la funcional de Winslow se aproxima en la forma

$$I_S(\mathbf{x}) \approx \sum \int_{B_{i,j}} \frac{\|\mathbf{r}_\xi\|^2 + \|\mathbf{r}_\eta\|^2}{\mathbf{r}_\xi^t J_2 \mathbf{r}_\eta} d\xi d\eta \quad (2.34)$$

(hemos omitido los índices en \mathbf{r} para no hacer pesada la notación). Veamos la forma de ésta aproximación. Si consideramos una malla de $m \times n$ en B_2 uniforme en ξ y η , tenemos que los puntos de la celda $B_{i,j}$ están determinados por

$$\begin{aligned} (\xi_i, \eta_j) &= \left(\frac{i}{m}, \frac{j}{n}\right) \rightarrow P_{i,j} & (\xi_{i+1}, \eta_j) &= \left(\frac{i+1}{m}, \frac{j}{n}\right) \rightarrow P_{i+1,j} \\ (\xi_{i+1}, \eta_{j+1}) &= \left(\frac{i+1}{m}, \frac{j+1}{n}\right) \rightarrow P_{i+1,j+1} & (\xi_i, \eta_{j+1}) &= \left(\frac{i}{m}, \frac{j+1}{n}\right) \rightarrow P_{i,j+1} \end{aligned} \quad (2.35)$$

Ahora bien, para no arrastrar la notación de los puntos de la malla en Ω , representemos a la celda $c_{i,j}$ como el cuadrilátero formado por los puntos P, Q, R, S en el mismo sentido positivo de orientación de la celda, ver Figura 2.16. Observemos la forma del mapeo bilineal \mathbf{r} para ésta celda y la aproximación al funcional de Winslow en $B_{i,j}$.

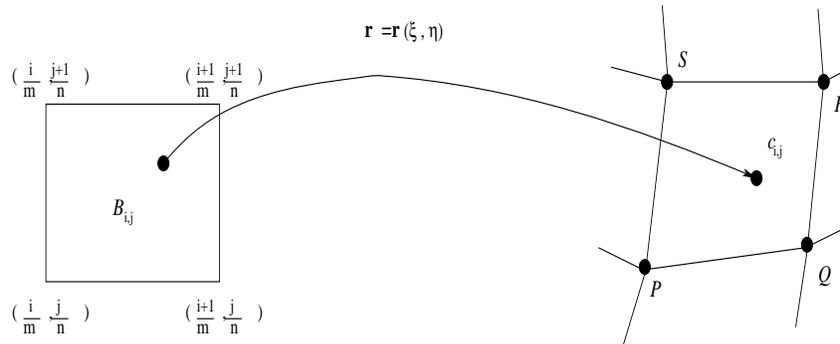


Figura 2.16: Mapeo bilineal entre $B_{i,j}$ y $c_{i,j}$

El mapeo bilineal entre $B_{i,j}$ y el cuadrilátero P, Q, R, S , lo podemos escribir en la forma

$$\mathbf{r}(\xi, \eta) = \mathbf{A} + \mathbf{B}\left(\xi - \frac{i}{m}\right) + \mathbf{C}\left(\eta - \frac{j}{n}\right) + \mathbf{D}\left(\xi - \frac{i}{m}\right)\left(\eta - \frac{j}{n}\right) \quad (2.36)$$

los coeficientes del mapeo son determinados por la correspondencia entre los puntos, así tendremos que

$$\begin{aligned} \mathbf{A} &= P \\ \mathbf{B} &= m(Q - P) \end{aligned}$$

$$\begin{aligned}\mathbf{C} &= n(S - P) \\ \mathbf{D} &= mn(R - Q + P - S)\end{aligned}\quad (2.37)$$

Consideremos la aproximación de la funcional de Winslow restringida a la celda $B_{i,j}$ usando el mapeo bilineal \mathbf{r} y usemos una regla de cuadratura para aproximar la integral sobre $B_{i,j}$

$$\begin{aligned}\int_{B_{i,j}} \frac{\|\mathbf{r}_\xi\|^2 + \|\mathbf{r}_\eta\|^2}{\mathbf{r}_\xi^t J_2 \mathbf{r}_\eta} &\approx \frac{1}{4} \left[\frac{\|\mathbf{r}_\xi(\frac{i}{m}, \frac{j}{n})\|^2 + \|\mathbf{r}_\eta(\frac{i}{m}, \frac{j}{n})\|^2}{\mathbf{r}_\xi^t(\frac{i}{m}, \frac{j}{n}) J_2 \mathbf{r}_\eta(\frac{i}{m}, \frac{j}{n})} \right. \\ &+ \frac{\|\mathbf{r}_\xi(\frac{i+1}{m}, \frac{j}{n})\|^2 + \|\mathbf{r}_\eta(\frac{i+1}{m}, \frac{j}{n})\|^2}{\mathbf{r}_\xi^t(\frac{i+1}{m}, \frac{j}{n}) J_2 \mathbf{r}_\eta(\frac{i+1}{m}, \frac{j}{n})} \\ &+ \frac{\|\mathbf{r}_\xi(\frac{i+1}{m}, \frac{j+1}{n})\|^2 + \|\mathbf{r}_\eta(\frac{i+1}{m}, \frac{j+1}{n})\|^2}{\mathbf{r}_\xi^t(\frac{i+1}{m}, \frac{j+1}{n}) J_2 \mathbf{r}_\eta(\frac{i+1}{m}, \frac{j+1}{n})} \\ &\left. + \frac{\|\mathbf{r}_\xi(\frac{i}{m}, \frac{j+1}{n})\|^2 + \|\mathbf{r}_\eta(\frac{i}{m}, \frac{j+1}{n})\|^2}{\mathbf{r}_\xi^t(\frac{i}{m}, \frac{j+1}{n}) J_2 \mathbf{r}_\eta(\frac{i}{m}, \frac{j+1}{n})} \right]\end{aligned}\quad (2.38)$$

Por una parte, las parciales de \mathbf{r} son

$$\begin{aligned}\mathbf{r}_\xi &= m(Q - P) + mn(R + P - Q - S)(\eta - \frac{j}{n}) \\ \mathbf{r}_\eta &= n(S - P) + mn(R + P - Q - S)(\xi - \frac{i}{m})\end{aligned}\quad (2.39)$$

y sus valores en las esquinas de $B_{i,j}$

$$\begin{aligned}\mathbf{r}_\xi(\frac{i}{m}, \frac{j}{n}) &= m(Q - P) & \mathbf{r}_\eta(\frac{i}{m}, \frac{j}{n}) &= n(S - P) \\ \mathbf{r}_\xi(\frac{i+1}{m}, \frac{j}{n}) &= m(Q - P) & \mathbf{r}_\eta(\frac{i+1}{m}, \frac{j}{n}) &= n(R - Q) \\ \mathbf{r}_\xi(\frac{i+1}{m}, \frac{j+1}{n}) &= m(R - S) & \mathbf{r}_\eta(\frac{i+1}{m}, \frac{j+1}{n}) &= n(R - Q) \\ \mathbf{r}_\xi(\frac{i}{m}, \frac{j+1}{n}) &= m(R - S) & \mathbf{r}_\eta(\frac{i}{m}, \frac{j+1}{n}) &= n(S - P)\end{aligned}\quad (2.40)$$

El primer sumando de (2.38) sería

$$\frac{\|\mathbf{r}_\xi(\frac{i}{m}, \frac{j}{n})\|^2 + \|\mathbf{r}_\eta(\frac{i}{m}, \frac{j}{n})\|^2}{\mathbf{r}_\xi^t(\frac{i}{m}, \frac{j}{n}) J_2 \mathbf{r}_\eta(\frac{i}{m}, \frac{j}{n})} = \frac{m^2 \|Q - P\|^2 + n^2 \|S - P\|^2}{mn(Q - P)^t J_2(S - P)}\quad (2.41)$$

de manera similar tendríamos los restantes sumandos

$$\frac{\|\mathbf{r}_\xi(\frac{i+1}{m}, \frac{j}{n})\|^2 + \|\mathbf{r}_\eta(\frac{i+1}{m}, \frac{j}{n})\|^2}{\mathbf{r}_\xi^t(\frac{i+1}{m}, \frac{j}{n})J_2\mathbf{r}_\eta(\frac{i+1}{m}, \frac{j}{n})} = \frac{m^2\|Q - P\|^2 + n^2\|R - Q\|^2}{mn(Q - P)^tJ_2(R - Q)} \quad (2.42)$$

$$\frac{\|\mathbf{r}_\xi(\frac{i+1}{m}, \frac{j+1}{n})\|^2 + \|\mathbf{r}_\eta(\frac{i+1}{m}, \frac{j+1}{n})\|^2}{\mathbf{r}_\xi^t(\frac{i+1}{m}, \frac{j+1}{n})J_2\mathbf{r}_\eta(\frac{i+1}{m}, \frac{j+1}{n})} = \frac{m^2\|R - S\|^2 + n^2\|R - Q\|^2}{mn(R - S)^tJ_2(R - Q)} \quad (2.43)$$

$$\frac{\|\mathbf{r}_\xi(\frac{i}{m}, \frac{j+1}{n})\|^2 + \|\mathbf{r}_\eta(\frac{i}{m}, \frac{j+1}{n})\|^2}{\mathbf{r}_\xi^t(\frac{i}{m}, \frac{j+1}{n})J_2\mathbf{r}_\eta(\frac{i}{m}, \frac{j+1}{n})} = \frac{m^2\|R - S\|^2 + n^2\|S - P\|^2}{mn(R - S)^tJ_2(S - P)} \quad (2.44)$$

con esto, nuestra aproximación al funcional de Winslow sobre $B_{i,j}$ la escribimos en términos de los puntos P, Q, R, S en la forma:

$$\int_{B_{i,j}} \frac{\|\mathbf{r}_\xi\|^2 + \|\mathbf{r}_\eta\|^2}{\mathbf{r}_\xi^tJ_2\mathbf{r}_\eta} \approx \frac{1}{4} \left[\frac{m^2\|Q - P\|^2 + n^2\|S - P\|^2}{mn(Q - P)^tJ_2(S - P)} + \frac{m^2\|Q - P\|^2 + n^2\|R - Q\|^2}{mn(Q - P)^tJ_2(R - Q)} + \frac{m^2\|R - S\|^2 + n^2\|R - Q\|^2}{mn(R - S)^tJ_2(R - Q)} + \frac{m^2\|R - S\|^2 + n^2\|S - P\|^2}{mn(R - S)^tJ_2(S - P)} \right] \quad (2.45)$$

con lo cual tendremos que dicha expresión es una suma de funciones que dependen únicamente de tres puntos de un triángulo; es decir una función de triángulos. En el caso particular en que $m = n$, la expresión anterior se simplifica aún más

$$\int_{B_{i,j}} \frac{\|\mathbf{r}_\xi\|^2 + \|\mathbf{r}_\eta\|^2}{\mathbf{r}_\xi^tJ_2\mathbf{r}_\eta} \approx \frac{1}{4} \left[\frac{\|Q - P\|^2 + \|S - P\|^2}{(Q - P)^tJ_2(S - P)} + \frac{\|Q - P\|^2 + \|R - Q\|^2}{(Q - P)^tJ_2(R - Q)} + \frac{\|R - S\|^2 + \|R - Q\|^2}{(R - S)^tJ_2(R - Q)} + \frac{\|R - S\|^2 + \|S - P\|^2}{(R - S)^tJ_2(S - P)} \right] \quad (2.46)$$

Con esto, la integral de Winslow la hemos aproximado por una función o colección de funciones, que dependen únicamente de los nodos interiores de la malla. En síntesis,

$$I_S \approx I_{\mathcal{D}_S} = \sum_{i=2}^{m-1} \sum_{j=2}^{n-1} \psi(P_{i,j}, P_{i+1,j}, P_{i+1,j+1}, P_{i,j+1}) \quad (2.47)$$

Ahora bien, hemos transformado el problema continuo de Winslow en un problema de minimización sobre los nodos interiores de la malla. El número de nodos interiores es $(m - 2)(n - 2)$ por lo que para mallas de dimensión grande $m, n > 30$ contaremos con un problema de optimización de gran escala. Por otra parte, este es un problema sin restricciones, ya que los nodos en la frontera de Ω están fijos.

Ivanenko [26] observa esta discretización del funcional de Winslow y se interesa por lograr los nodos interiores de la malla para conformar la configuración, véase la Figura 2.17. Procede a emplear un método Quasi-Newton para resolver el sistema de ecuaciones provenientes de hacer cero al gradiente en cada uno de los puntos de $P_{i,j}$, es decir:

$$R_x = \frac{\partial I}{\partial x_{ij}} = 0 \quad R_y = \frac{\partial I}{\partial y_{ij}} = 0 \quad (2.48)$$

o en la forma explícita

$$\begin{aligned} \tau R_x + \frac{\partial R_x}{\partial x}(x_{ij}^{l+1} - x_{ij}^l) + \frac{\partial R_x}{\partial y}(y_{ij}^{l+1} - y_{ij}^l) &= 0 \\ \tau R_y + \frac{\partial R_y}{\partial x}(x_{ij}^{l+1} - x_{ij}^l) + \frac{\partial R_y}{\partial y}(y_{ij}^{l+1} - y_{ij}^l) &= 0. \end{aligned} \quad (2.49)$$

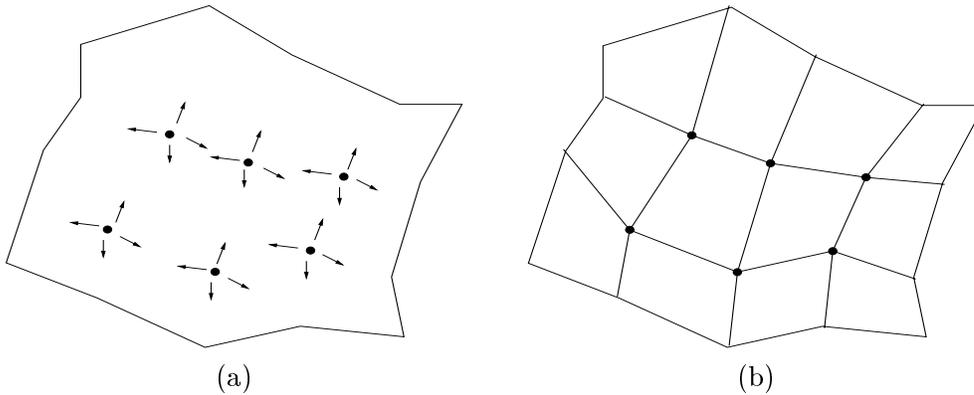


Figura 2.17: La malla como configuración de puntos.

Para resolver este sistema, Ivanenko construye una malla inicial convexa y aplica un método Quasi-Newton con τ eligiéndola de manera que la solución de (2.49) sea una malla convexa, este parámetro lo va actualizando mediante un esquema de interpolación entre valores de τ [26], y así garantiza que trabajará siempre con mallas convexas dándole sentido a la discretización. La elección de la malla inicial para el método de optimización es un problema más complejo, él la obtiene mediante

un proceso de optimización de mínimos cuadrados entre $\epsilon > 0$ y el jacobiano de la discretización

$$I_{I_0} = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \sum_{k=1}^4 (\epsilon - J(\Delta_{ij}^{(k)})_+)^2 \tag{2.50}$$

con $(f)_+ = \max(0, f)$.

Como se observa, el procedimiento descrito por Ivanenko es muy costoso y artesanal. Construir una malla convexa inicial es muy difícil. A lo largo del documento hemos comentado lo difícil es que lograr una malla convexa en una región Ω muy irregular y agregando las condiciones para garantizar que en cada paso de (2.49) se obtengan mallas convexas resultaría muy costoso, todo esto porque el funcional discreto (2.47) únicamente opera sobre mallas convexas.

Barrera en 1992 [4] consideró el proceso de discretizar el funcional de suavidad, no sólo como un problema de optimización de gran escala sin restricciones. Observó la configuración de la malla como una colección de triángulos orientados y entonces el funcional de suavidad como un proceso de optimización sobre todos esos triángulos formados por las celdas de la malla, ver Figura 2.18. Veamos cómo precedió.

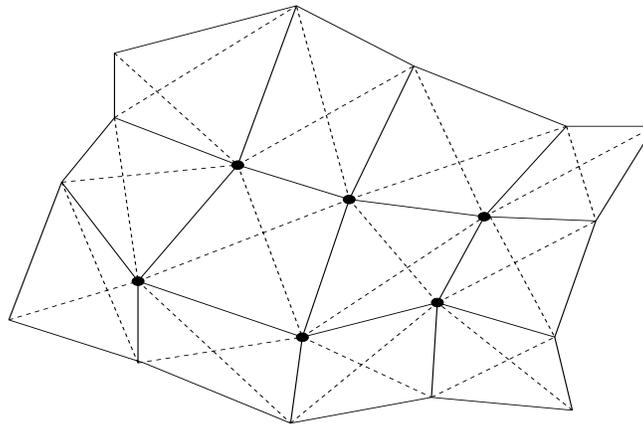


Figura 2.18: Una malla como colección de triángulos.

Consideremos los cuatro triángulos de la celda $c_{i,j}$, con la orientación positiva, contraria a las manecillas del reloj, como en la Figura 2.19

Por una parte, los denominadores de (2.46) (el integral de Winslow restringido a la celda $B_{i,j}$) bajo la orientación positiva, pueden ser escritos a través del área del triángulo en curso $\Delta^{(k)}$, si este lo denotamos por P, Q, R tenemos lo siguiente

$$2 \text{ área}(P, Q, S) = \det(Q - P, S - P) = (Q - P)^t J_2(S - P) \tag{2.51}$$

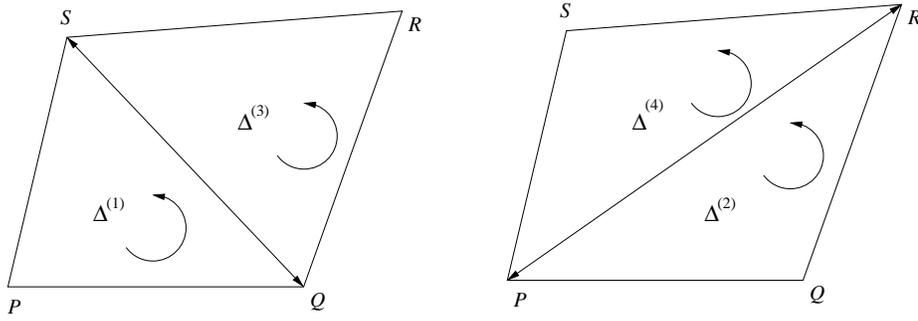


Figura 2.19: Triángulos de una celda.

$$\int_{B_{i,j}} \frac{\|\mathbf{r}_\xi\|^2 + \|\mathbf{r}_\eta\|^2}{\mathbf{r}_\xi^t J_2 \mathbf{r}_\eta} \approx \frac{1}{4} \left[\frac{\|Q - P\|^2 + \|S - P\|^2}{2 \text{área}(P, Q, S)} + \frac{\|Q - P\|^2 + \|R - Q\|^2}{2 \text{área}(Q, R, P)} + \frac{\|R - S\|^2 + \|R - Q\|^2}{2 \text{área}(R, S, Q)} + \frac{\|R - S\|^2 + \|S - P\|^2}{2 \text{área}(S, P, R)} \right] \quad (2.52)$$

Siguiendo la notación sugerida en la Figura 2.19, tenemos que

$$\begin{aligned} \Delta^{(1)} &= \Delta(P, Q, S) \\ \Delta^{(2)} &= \Delta(Q, R, P) \\ \Delta^{(3)} &= \Delta(R, S, Q) \\ \Delta^{(4)} &= \Delta(S, P, R) \end{aligned} \quad (2.53)$$

y podemos simplificar la expresión (2.52) escribiéndola como función de triángulos en la forma

$$\int_{B_{i,j}} \frac{\|\mathbf{r}_\xi\|^2 + \|\mathbf{r}_\eta\|^2}{\mathbf{r}_\xi^t J_2 \mathbf{r}_\eta} \approx \frac{1}{4} \sum_{k=1}^4 f(\Delta^{(k)}) \quad (2.54)$$

donde

$$f(\Delta) = \frac{\|P - Q\|^2 + \|R - P\|^2}{2\alpha(\Delta)} \quad (2.55)$$

donde P, Q, R es el triángulo $\Delta^{(k)}$ en curso, ver Figura 2.20. Por otra parte

$$\alpha(\Delta) = \frac{1}{2} \det(Q - P, R - P) = \text{área}(P, Q, R). \quad (2.56)$$

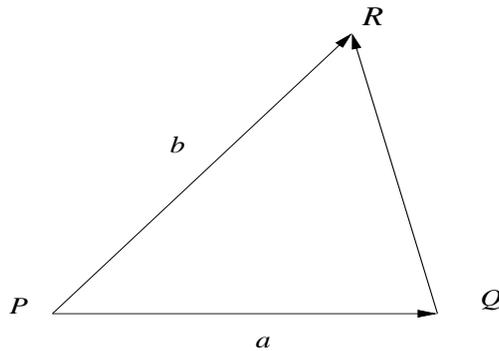


Figura 2.20: Triángulo genérico de una celda.

Con esto, la integral de Winslow la hemos aproximado por una función o colección de funciones, que dependen únicamente de los nodos interiores de la malla.

$$\begin{aligned}
 I_S \approx I_{\mathcal{D}_S} &= \frac{1}{4} \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \sum_{k=1}^4 f(\Delta_{ij}^{(k)}) \\
 &= \frac{1}{4} \sum_{q=1}^N f(\Delta_q)
 \end{aligned} \tag{2.57}$$

donde N es el número de triángulos de la malla. Así, el problema continuo de Winslow se transforma en un problema de minimización sobre los triángulos de la malla

$$\min_{G \in \mathcal{D}} I_{\mathcal{D}_S} \tag{2.58}$$

donde \mathcal{D} es el conjunto de mallas sobre Ω de dimensión $m \times n$. Un problema que está exento de restricciones.

Por la forma de la funcional discreta $I_{\mathcal{D}_S}$, Barrera propuso una regularización del funcional para que sea el optimizador quien se encargue de modificar aquellos triángulos con área problemática, los triángulos que hagan inestable a f . Para de esta forma ampliar el conjunto de mallas sobre las cuales operar con el funcional discretizado; conjunto de mallas que llamó *cuasi-convexas*. A continuación describiremos el procedimiento que empleó.

2.4 Funcional de Suavidad discreto regularizado

Si el problema de minimización (2.58) lo atacamos como un problema de minimización de gran escala, podemos considerar las partes elementales de la función $I_{\mathcal{D}_s}$ y observar su comportamiento sobre los triángulos.

Consideramos la función sobre cada triángulo

$$f(P, Q, R) = \frac{l(P, Q, R)}{2\alpha(P, Q, R)} \quad (2.59)$$

con

$$l(P, Q, R) = \|Q - P\|^2 + \|R - P\|^2 \quad (2.60)$$

Por una parte, se observa que para celdas casi convexas, el numerador l para los triángulos de la celda, no sufren cambios considerables por lo que puede considerarse constante en esa gama de celdas y tendría la función un comportamiento semejante a $1/t$. Por otra parte para esa gama de celdas el área de los triángulos hace inestable la funcional f ya que para valores suficientemente pequeños de α se podría provocar un *overflow*. La idea de Barrera es controlar esta inestabilidad y ampliar el conjunto de mallas a trabajar considerando mallas casi-convexas, mallas con pocas celdas no convexas; para esto, propone correr el polo en f , veamos cómo lograrlo.

Si aproximamos el efecto

$$\varphi(t) = \frac{1}{t} \quad (2.61)$$

por

$$\tilde{\varphi}(t) = \begin{cases} \frac{1}{t} & \text{si } t \geq \epsilon_c \\ \frac{a}{(t+\epsilon)^2} & \text{si } -\epsilon_c < t \leq \epsilon_c \end{cases} \quad (2.62)$$

estaremos desplazando el polo de 0 a $-\epsilon_c$ y con ello trabajando con triángulo de área pequeña, positiva o negativa. Pidiendo diferenciabilidad continua a $\tilde{\varphi}$, tendremos que:

$$\tilde{\varphi}(t) = \begin{cases} \frac{1}{t} & \text{si } t \geq \epsilon_c \\ \frac{4\epsilon_c}{(t+\epsilon)^2} & \text{si } -\epsilon_c < t \leq \epsilon_c \end{cases} \quad (2.63)$$

Así, bajo ésta idea de correr el polo de f para aceptar triángulos de área pequeña la regularización que propone Barrera al funcional sobre triángulos tiene la forma

$$\tilde{f}(\Delta) = \begin{cases} \frac{l(\Delta)}{2\alpha(\Delta)} & \text{si } \alpha(\Delta) \geq \epsilon_c \\ \frac{4\epsilon_c l(\Delta)}{(2\alpha(\Delta)+\epsilon_c)^2} & \text{si } -\epsilon_c < \alpha(\Delta) \leq \epsilon_c \end{cases} \quad (2.64)$$

en la Figura 2.21, se observa el corrimiento de los polos.

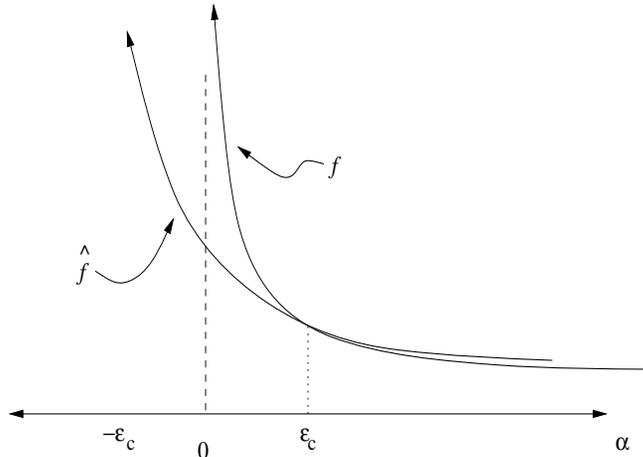


Figura 2.21: Regularización de f .

La condición de continuidad de primer orden es suficiente para que la función \hat{f} que regulariza a f lo haga sin problemas de discontinuidad en los gradientes y entonces puedan operar los métodos de optimización, con ésta regularización y con se van corrigiendo los triángulos con área pequeña positiva o negativa, o bien cero. Ver la Figura 2.22 para observar el efecto deseado por la regularización.

El valor ϵ_c que aparece en la regularización de f y que representa hacia dónde se ha corrido el polo, depende indiscutiblemente de la complejidad de la región en que se genera la malla. De igual manera éste valor depende de la noción ϵ -convexidad a considerar; esto es, qué tan no convexa es nuestra malla inicial.

Barrera propuso que, de considerar regiones donde es posible trabajar con mallas convexas y que en el óptimo sea de área uniforme, el valor de ϵ_c sería:

$$\epsilon_c = \frac{\text{área}(\Omega)}{2(m-1)(n-1)}. \quad (2.65)$$

El trabajo de Barrera dió la pauta para analizar ésta forma de regularizar f y construir posteriormente funcionales que llevaran consigo la idea de ampliar el dominio de trabajo de las mallas casi-convexas.

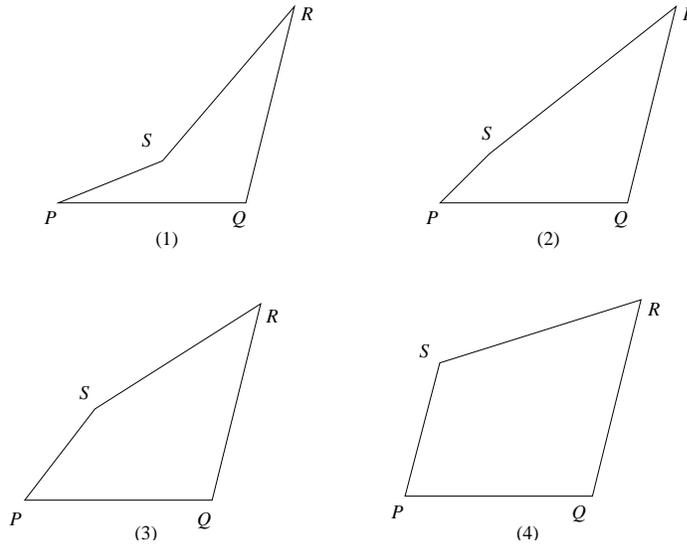


Figura 2.22: Efecto de la regularización sobre una celda. Hasta el paso 3 vendría actuando la regularización \hat{f} .

Para generar un malla convexa usando el funcional regularizado debe iniciarse con una malla de pocas celdas no convexas (Barrera [4]); iniciando con una malla con muchas celdas no convexas el proceso se vuelve lento y los “ruidos” provocados por la regularización nos llevarán a resultados no deseados. En ese sentido este funcional falla para lograr mallas suaves y convexas de manera automática. Sin embargo, la idea de regularizar los funcionales es muy interesante y atractiva a los métodos de optimización de gran escala en sus distintas estrategias. Nos será útil toda vez que trabajemos con funcionales discretos de la forma $1/t$.

Ahora veamos algunas propiedades atractivas detrás del funcional de suavidad observadas sobre los triángulos de la malla como lo describió Barrera, razón por la cual se siguió trabajando sobre esa línea: analizar las propiedades geométricas de los triángulos de la malla.

2.4.1 Propiedades del funcional discreto de Suavidad

Sobre un triángulo PQR , orientado positivamente (contrario a las manecillas del reloj, ver Figura 2.23) el funcional discreto de suavidad tiene la representación

$$f(P, Q, R) = \frac{l(P, Q, R)}{\alpha(P, Q, R)} \quad (2.66)$$

donde $l = l(P, Q, R)$ viene dado por

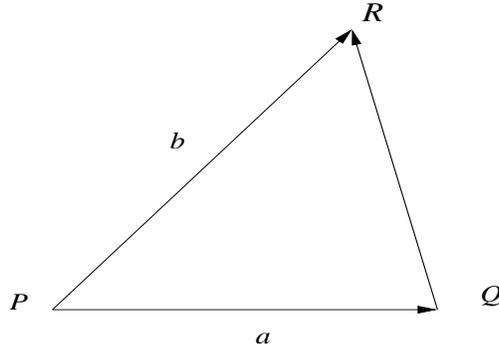


Figura 2.23: Triángulo genérico.

$$l(P, Q, R) = \|P - Q\|^2 + \|P - R\|^2 \tag{2.67}$$

y $\alpha = \alpha(P, Q, R)$ por

$$\alpha(P, Q, R) = (Q - P)^t J_2 (R - P) \tag{2.68}$$

con

$$J_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \tag{2.69}$$

Ahora fijemos dos puntos, digamos P y Q y dejemos libre a R ; el valor del funcional sobre P, Q, R lo podemos escribir en forma vectorial como $\mathbf{a} = Q - P$, $\mathbf{b} = R - P$, escribiendo la función de la forma

$$\bar{f}(\mathbf{a}, \mathbf{b}) = \frac{\bar{l}(\mathbf{a}, \mathbf{b})}{\bar{\alpha}(\mathbf{a}, \mathbf{b})} = \frac{\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2}{\mathbf{a}^t J_2 \mathbf{b}}. \tag{2.70}$$

Veamos como es la forma que toman \mathbf{a} y \mathbf{b} en un punto crítico de \bar{f} . Por un lado tendremos que

$$\begin{aligned} \frac{\partial \bar{l}}{\partial \mathbf{a}}(\mathbf{a}, \mathbf{b}) &= 2\mathbf{a} \\ \frac{\partial \bar{l}}{\partial \mathbf{b}}(\mathbf{a}, \mathbf{b}) &= 2\mathbf{b} \\ \frac{\partial \bar{\alpha}}{\partial \mathbf{a}}(\mathbf{a}, \mathbf{b}) &= J_2 \mathbf{b} \\ \frac{\partial \bar{\alpha}}{\partial \mathbf{b}}(\mathbf{a}, \mathbf{b}) &= -J_2 \mathbf{a} \end{aligned}$$

y por el otro

$$\begin{aligned}\frac{\partial \bar{f}}{\partial \mathbf{a}} &= \frac{1}{\alpha} \left[\frac{\partial \bar{l}}{\partial \mathbf{a}} - \frac{\bar{l}}{\bar{\alpha}} - \frac{\partial \bar{\alpha}}{\partial \mathbf{a}} \right] \\ &= \frac{\bar{l}}{\bar{\alpha}} \left[2\mathbf{a} - \frac{\bar{l}}{\bar{\alpha}} J_2 \mathbf{b} \right]\end{aligned}\quad (2.71)$$

y

$$\begin{aligned}\frac{\partial \bar{f}}{\partial \mathbf{b}} &= \frac{1}{\alpha} \left[\frac{\partial \bar{l}}{\partial \mathbf{b}} - \frac{\bar{l}}{\bar{\alpha}} - \frac{\partial \bar{\alpha}}{\partial \mathbf{b}} \right] \\ &= \frac{\bar{l}}{\bar{\alpha}} \left[2\mathbf{b} + \frac{\bar{l}}{\bar{\alpha}} J_2 \mathbf{a} \right]\end{aligned}\quad (2.72)$$

Si consideramos que $\alpha \neq 0$, el punto crítico de \bar{f} satisface que

$$2\mathbf{a} - \frac{\bar{l}}{\bar{\alpha}} J_2 \mathbf{b} = 0 \quad (2.73)$$

$$2\mathbf{b} + \frac{\bar{l}}{\bar{\alpha}} J_2 \mathbf{a} = 0, \quad (2.74)$$

resolviendo el sistema anterior tendremos que

$$\mathbf{a} = \pm J_2 \mathbf{b} \quad (2.75)$$

por una parte

$$\|\mathbf{a}\| = \|\mathbf{b}\| \quad (2.76)$$

y por la otra

$$\mathbf{a}^t \mathbf{b} = 0, \quad (2.77)$$

es decir, el triángulo P, Q, R que es punto crítico de \bar{f} es un rectángulo isósceles; esto es, los triángulos de las celdas de la mallas tratan de lograr esta propiedad geométrica y al lograrlo la configuración así obtenida puede ser preferida sobre otras configuraciones para resolver tanto ecuaciones diferenciales en diferencias como por el

método de elemento finito. Por otra parte la orientación introducida en los triángulos nos lleva buscar el mínimo para (2.58) y es precisamente ésta consideración la que influye para que, a través del optimizador, busquemos la propiedad de la convexidad por medio de los triángulos de las celdas. Las celdas no se doblan en el extremo si hay una orientación de los triángulos.

Esta propiedad de fijar una orientación a las celdas y a los triángulos de las celdas la observó Barrera en trabajos anteriores sobre los funcionales discretos de área, Barrera 1989 [3], logrando propiedades muy interesantes del mínimo de las funcionales y extendiendo resultados a otros estudios, como lo es el caso del funcional discreto de suavidad.

La propiedad de lograr triángulos isósceles bajo esta funcional es local, es sobre triángulos aislados; sin embargo, y debido a la forma parcialmente separable de la funcional: suma de elementos simples dependientes de pocas variables, podemos poner atención en un comportamiento por subregiones a lo largo de la región de interés Ω .

Hasta aquí, hemos probado que el funcional discreto de suavidad trata de lograr que los triángulos sean rectángulos isósceles. Si la región es suficientemente regular obtendremos sobre ella mallas con ésta característica y si la región es irregular y nos encontramos cerca de una malla convexa, podemos lograr una malla convexa donde por subregiones observemos triángulos cercanos a ser rectángulos isósceles.

Antes de concluir con las propiedades del funcional discreto de suavidad, recordemos que para lograr (2.52) supusimos que la malla en B_2 es uniforme y de dimensión m . Pero ésto no es la regla, las mallas en B_2 pueden contar con distinta distribución en sus fronteras y la malla en B_2 generada por la interpolación entre las fronteras puede ser usada para discretizar el funcional de Winslow, observese la Figura 2.24.

Veamos el caso particular en que la malla en B_2 no es uniforme. Consideremos que las funciones $g_b(\xi)$, $g_r(\eta)$, $g_t(\xi)$ y $g_l(\eta)$ son parametrizaciones de $[0, 1]$ en $[0, 1]$ y representan los lados correspondientes de B_2 con la particularidad de que

$$\begin{aligned} g_b(\xi) &= g_t(\xi) \\ g_l(\eta) &= g_r(\eta) \end{aligned} \tag{2.78}$$

es decir, que los segmentos de abajo y arriba, cuentan con la misma parametrización. De igual forma, los segmentos de la izquierda y de la derecha.

Ahora bien, tomemos en cuenta el mapeo bilineal que transforme cada uno de los rectángulos de B_2 en las celdas de la malla en Ω . Si P, Q, R, S representan los puntos de una celda $c_{i,j}$ en Ω , la correspondencia con $B_{i,j}$ la escribimos como

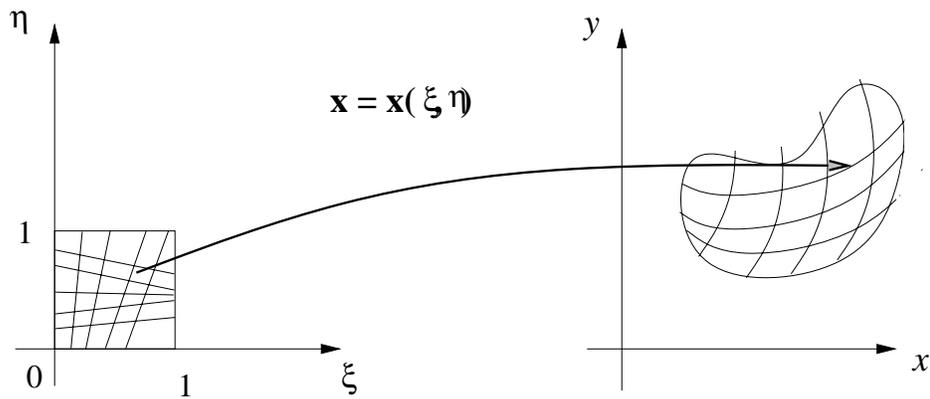


Figura 2.24: Mapeo de una malla en B_2 no uniforme.

$$\begin{aligned}
 (\xi_i, \eta_j) &= (g_b(\frac{i}{m}), g_l(\frac{j}{n})) \rightarrow P_{i,j} & (\xi_{i+1}, \eta_j) &= (g_b(\frac{i+1}{m}), g_l(\frac{j}{n})) \rightarrow P_{i+1,j} \\
 (\xi_{i+1}, \eta_{j+1}) &= (g_b(\frac{i+1}{m}), g_l(\frac{j+1}{n})) \rightarrow P_{i+1,j+1} & (\xi_i, \eta_{j+1}) &= (g_b(\frac{i}{m}), g_l(\frac{j+1}{n})) \rightarrow P_{i,j+1}
 \end{aligned} \tag{2.79}$$

El mapeo bilineal entre $B_{i,j}$ y el cuadrilátero P, Q, R, S lo podemos expresar en la forma

$$\mathbf{r}(\xi, \eta) = \mathbf{A} + \mathbf{B}(\xi - g_b(\frac{i}{m})) + \mathbf{C}(\eta - g_l(\frac{j}{n})) + \mathbf{D}(\xi - g_b(\frac{i}{m}))(\eta - g_l(\frac{j}{n})) \tag{2.80}$$

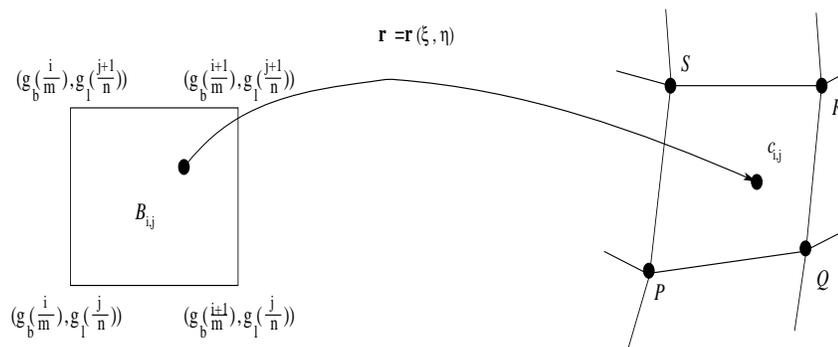


Figura 2.25: Mapeo bilineal entre $B_{i,j}$ y $c_{i,j}$

Determinando los coeficientes por la correspondencia de los puntos tenemos que

$$\begin{aligned}
 \mathbf{A} &= P \\
 \mathbf{B} &= \frac{1}{g_b(\frac{i+1}{m}) - g_b(\frac{i}{m})}(Q - P) \\
 \mathbf{C} &= \frac{1}{g_l(\frac{j+1}{n}) - g_l(\frac{j}{n})}(S - P) \\
 \mathbf{D} &= \frac{R - Q + P - S}{(g_b(\frac{i+1}{m}) - g_b(\frac{i}{m})) (g_l(\frac{j+1}{n}) - g_l(\frac{j}{n}))} \tag{2.81}
 \end{aligned}$$

Consideremos la aproximación de la funcional de Winslow restringida a la celda $B_{i,j}$, para esto tomemos en cuenta el mapeo bilineal \mathbf{r} y usemos una regla de cuadratura para aproximar la integral sobre $B_{i,j}$, así

$$\begin{aligned}
 \int_{B_{i,j}} \frac{\|\mathbf{r}_\xi\|^2 + \|\mathbf{r}_\eta\|^2}{\mathbf{r}_\xi^t J_2 \mathbf{r}_\eta} \approx & \frac{1}{4} \left[\frac{\|\mathbf{r}_\xi(g_b(\frac{i}{m}), g_l(\frac{j}{n}))\|^2 + \|\mathbf{r}_\eta(g_b(\frac{i}{m}), g_l(\frac{j}{n}))\|^2}{\mathbf{r}_\xi^t(g_b(\frac{i}{m}), g_l(\frac{j}{n})) J_2 \mathbf{r}_\eta(g_b(\frac{i}{m}), g_l(\frac{j}{n}))} \right. \\
 & + \frac{\|\mathbf{r}_\xi(g_b(\frac{i+1}{m}), g_l(\frac{j}{n}))\|^2 + \|\mathbf{r}_\eta(g_b(\frac{i+1}{m}), g_l(\frac{j}{n}))\|^2}{\mathbf{r}_\xi^t(g_b(\frac{i+1}{m}), g_l(\frac{j}{n})) J_2 \mathbf{r}_\eta(g_b(\frac{i+1}{m}), g_l(\frac{j}{n}))} \\
 & + \frac{\|\mathbf{r}_\xi(g_b(\frac{i+1}{m}), g_l(\frac{j+1}{n}))\|^2 + \|\mathbf{r}_\eta(g_b(\frac{i+1}{m}), g_l(\frac{j+1}{n}))\|^2}{\mathbf{r}_\xi^t(g_b(\frac{i+1}{m}), g_l(\frac{j+1}{n})) J_2 \mathbf{r}_\eta(g_b(\frac{i+1}{m}), g_l(\frac{j+1}{n}))} \\
 & \left. + \frac{\|\mathbf{r}_\xi(g_b(\frac{i}{m}), g_l(\frac{j+1}{n}))\|^2 + \|\mathbf{r}_\eta(g_b(\frac{i}{m}), g_l(\frac{j+1}{n}))\|^2}{\mathbf{r}_\xi^t(g_b(\frac{i}{m}), g_l(\frac{j+1}{n})) J_2 \mathbf{r}_\eta(g_b(\frac{i}{m}), g_l(\frac{j+1}{n}))} \right] \tag{2.82}
 \end{aligned}$$

Por una parte, las parciales de \mathbf{r} son

$$\begin{aligned}
 \mathbf{r}_\xi &= \frac{Q - P}{g_b(\frac{i+1}{m}) - g_b(\frac{i}{m})} + \frac{R + P - Q - S}{(g_b(\frac{i+1}{m}) - g_b(\frac{i}{m})) (g_l(\frac{j+1}{n}) - g_l(\frac{j}{n}))} (\eta - g_l(\frac{j}{n})) \\
 \mathbf{r}_\eta &= \frac{S - P}{g_l(\frac{j+1}{n}) - g_l(\frac{j}{n})} + \frac{R + P - Q - S}{(g_b(\frac{i+1}{m}) - g_b(\frac{i}{m})) (g_l(\frac{j+1}{n}) - g_l(\frac{j}{n}))} (\xi - g_b(\frac{i}{m})) \tag{2.83}
 \end{aligned}$$

y sus valores en las esquinas de $B_{i,j}$

$$\begin{aligned}
\mathbf{r}_\xi(g_b(\frac{i}{m}), g_l(\frac{j}{n})) &= \Delta_i^{-1} g_b(Q - P) & \mathbf{r}_\eta(g_b(\frac{i}{m}), g_l(\frac{j}{n})) &= \Delta_j^{-1} g_l(S - P) \\
\mathbf{r}_\xi(g_b(\frac{i+1}{m}), g_l(\frac{j}{n})) &= \Delta_i^{-1} g_b(Q - P) & \mathbf{r}_\eta(g_b(\frac{i+1}{m}), g_l(\frac{j}{n})) &= \Delta_j^{-1} g_l(R - Q) \\
\mathbf{r}_\xi(g_b(\frac{i+1}{m}), g_l(\frac{j+1}{n})) &= \Delta_i^{-1} g_b(R - S) & \mathbf{r}_\eta(g_b(\frac{i+1}{m}), g_l(\frac{j+1}{n})) &= \Delta_j^{-1} g_l(R - Q) \\
\mathbf{r}_\xi(g_b(\frac{i}{m}), g_l(\frac{j+1}{n})) &= \Delta_i^{-1} g_b(R - S) & \mathbf{r}_\xi(g_b(\frac{i}{m}), g_l(\frac{j+1}{n})) &= \Delta_j^{-1} g_l(S - P)
\end{aligned} \tag{2.84}$$

con

$$\Delta_i^{-1} g_b = \frac{1}{g_b(\frac{i+1}{m}) - g_b(\frac{i}{m})} \tag{2.85}$$

$$\Delta_j^{-1} g_l = \frac{1}{g_l(\frac{j+1}{n}) - g_l(\frac{j}{n})} \tag{2.86}$$

El primer sumando de (2.38) sería

$$\frac{\|\mathbf{r}_\xi(g_b(\frac{i}{m}), g_l(\frac{j}{n}))\|^2 + \|\mathbf{r}_\eta(g_b(\frac{i}{m}), g_l(\frac{j}{n}))\|^2}{\mathbf{r}_\xi^t(g_b(\frac{i}{m}), g_l(\frac{j}{n})) J_2 \mathbf{r}_\eta(g_b(\frac{i}{m}), g_l(\frac{j}{n}))} = \frac{(\Delta_i^{-1} g_b)^2 \|Q - P\|^2 + (\Delta_j^{-1} g_l)^2 \|S - P\|^2}{\Delta_i^{-1} g_b \Delta_j^{-1} g_l (Q - P)^t J_2 (S - P)} \tag{2.87}$$

de manera semejante tendríamos los restantes sumandos

$$\frac{\|\mathbf{r}_\xi(g_b(\frac{i+1}{m}), g_l(\frac{j}{n}))\|^2 + \|\mathbf{r}_\eta(g_b(\frac{i+1}{m}), g_l(\frac{j}{n}))\|^2}{\mathbf{r}_\xi^t(g_b(\frac{i+1}{m}), g_l(\frac{j}{n})) J_2 \mathbf{r}_\eta(g_b(\frac{i+1}{m}), g_l(\frac{j}{n}))} = \frac{(\Delta_i^{-1} g_b)^2 \|Q - P\|^2 + (\Delta_j^{-1} g_l)^2 \|R - Q\|^2}{\Delta_i^{-1} g_b \Delta_j^{-1} g_l (Q - P)^t J_2 (R - Q)} \tag{2.88}$$

$$\frac{\|\mathbf{r}_\xi(g_b(\frac{i+1}{m}), g_l(\frac{j+1}{n}))\|^2 + \|\mathbf{r}_\eta(g_b(\frac{i+1}{m}), g_l(\frac{j+1}{n}))\|^2}{\mathbf{r}_\xi^t(g_b(\frac{i+1}{m}), g_l(\frac{j+1}{n})) J_2 \mathbf{r}_\eta(g_b(\frac{i+1}{m}), g_l(\frac{j+1}{n}))} = \frac{(\Delta_i^{-1} g_b)^2 \|R - S\|^2 + (\Delta_j^{-1} g_l)^2 \|R - Q\|^2}{\Delta_i^{-1} g_b \Delta_j^{-1} g_l (R - S)^t J_2 (R - Q)} \tag{2.89}$$

$$\frac{\|\mathbf{r}_\xi(g_b(\frac{i}{m}), g_l(\frac{j+1}{n}))\|^2 + \|\mathbf{r}_\eta(g_b(\frac{i}{m}), g_l(\frac{j+1}{n}))\|^2}{\mathbf{r}_\xi^t(g_b(\frac{i}{m}), g_l(\frac{j+1}{n})) J_2 \mathbf{r}_\eta(g_b(\frac{i}{m}), g_l(\frac{j+1}{n}))} = \frac{(\Delta_i^{-1} g_b)^2 \|R - S\|^2 + (\Delta_j^{-1} g_l)^2 \|S - P\|^2}{\Delta_i^{-1} g_b \Delta_j^{-1} g_l (R - S)^t J_2 (S - P)} \tag{2.90}$$

y entonces, nuestra aproximación al funcional de Winslow sobre $B_{i,j}$ la escribimos en términos de los puntos P, Q, R, S en la forma:

$$\begin{aligned}
 \int_{B_{i,j}} \frac{\|\mathbf{r}_\xi\|^2 + \|\mathbf{r}_\eta\|^2}{\mathbf{r}_\xi^t J_2 \mathbf{r}_\eta} &\approx \frac{1}{4} \left[\frac{(\Delta_i^{-1} g_b)^2 \|Q - P\|^2 + (\Delta_j^{-1} g_l)^2 \|S - P\|^2}{\Delta_i^{-1} g_b \Delta_j^{-1} g_l (Q - P)^t J_2 (S - P)} \right. \\
 &+ \frac{(\Delta_i^{-1} g_b)^2 \|Q - P\|^2 + (\Delta_j^{-1} g_l)^2 \|R - Q\|^2}{\Delta_i^{-1} g_b \Delta_j^{-1} g_l (Q - P)^t J_2 (R - Q)} \\
 &+ \frac{(\Delta_i^{-1} g_b)^2 \|R - S\|^2 + (\Delta_j^{-1} g_l)^2 \|R - Q\|^2}{\Delta_i^{-1} g_b \Delta_j^{-1} g_l (R - S)^t J_2 (R - Q)} \\
 &\left. + \frac{(\Delta_i^{-1} g_b)^2 \|R - S\|^2 + (\Delta_j^{-1} g_l)^2 \|S - P\|^2}{\Delta_i^{-1} g_b \Delta_j^{-1} g_l (R - S)^t J_2 (S - P)} \right] \quad (2.91)
 \end{aligned}$$

En forma compacta, la discretización del funcional de Winslow la expresamos como

$$I_{\mathcal{D}_S} = \frac{1}{4} \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \sum_{k=1}^4 \bar{f}(\Delta_k) \quad (2.92)$$

donde

$$\bar{f}(\mathbf{a}, \mathbf{b}) = \frac{\alpha_k^2 \|\mathbf{a}\|^2 + \beta_k^2 \|\mathbf{b}\|}{2\alpha_k \beta_k \mathbf{a}^t J_2 \mathbf{b}} \quad (2.93)$$

donde los escalares α_k y β_k representan el recíproco de la diferencia entre el tamaño de los lados del rectángulo $B_{i,j}$.

Ahora bien observemos que

$$\begin{aligned}
 \bar{f}(\mathbf{a}, \mathbf{b}) - 1 &= \frac{\alpha_k^2 \|\mathbf{a}\|^2 + \beta_k^2 \|\mathbf{b}\|}{2\alpha_k \beta_k \mathbf{a}^t J_2 \mathbf{b}} - 1 \\
 &= \frac{\alpha_k^2 \|\mathbf{a}\|^2 + \beta_k^2 \|\mathbf{b}\| - 2\alpha_k \beta_k \mathbf{a}^t J_2 \mathbf{b}}{2\alpha_k \beta_k \mathbf{a}^t J_2 \mathbf{b}} \quad (2.94)
 \end{aligned}$$

$$= \frac{\|\alpha_k \mathbf{a} - \beta_k J_2 \mathbf{b}\|^2}{2\alpha_k \beta_k \mathbf{a}^t J_2 \mathbf{b}} \quad (2.95)$$

lo que nos llevará a tomar en cuenta que para triángulos de área positiva

$$\alpha_k \mathbf{a} - \beta_k J_2 \mathbf{b} = \mathbf{0} \quad (2.96)$$

y con esto obtener una relación entre las longitudes de los segmentos \mathbf{a} y \mathbf{b}

$$\|\mathbf{a}\| = \frac{\beta_k}{\alpha_k} \|\mathbf{b}\| \quad (2.97)$$

lo que nos lleva a la igualdad siguiente

$$2\alpha_k\beta_k\|\mathbf{a}\| \cdot \|\mathbf{b}\| = 2\beta_k^2\|\mathbf{b}\|^2 \quad (2.98)$$

Entonces, para mallas de dimensión $m \times n$, con $m \neq n$ y con distribución no uniforme a la frontera, usual en mallas sobre regiones muy irregulares, el funcional sobre triángulos busca que éstos sean cercanos a isósceles bajo la idea del *aspect ratio* que provoca la malla sobre la región.

Enfatizemos que para llegar a ésta representación supusimos que dos a dos los segmentos de frontera B_2 cuentan con la misma parametrización. Si usáramos parametrizaciones distintas de los lados de B_2 , lograríamos controlar ésta propiedad de los triángulos y obtener configuraciones interesantes a lo largo de la región en Ω .

Por otra parte, hemos escrito el funcional sobre triángulos de una forma muy interesante. Al considerar $\bar{f}(\mathbf{a}, \mathbf{b}) - 1$, (la cual dista de la original por una constante y desde luego cuentan ambas con los mismos puntos críticos) se obtiene una representación en la forma

$$\bar{f}(\mathbf{a}, \mathbf{b}) - 1 = \frac{\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\mathbf{a}^t J_2 \mathbf{b}}{2\mathbf{a}^t J_2 \mathbf{b}} \quad (2.99)$$

$$= \frac{\|\mathbf{a} - J_2 \mathbf{b}\|^2}{2\mathbf{a}^t J_2 \mathbf{b}} \quad (2.100)$$

para $\alpha_k = \beta_k$. Esta es la funcional a la que haremos referencia cuando la retomemos en los capítulos siguientes para ampliar su dominio de trabajo, y de igual forma, es la funcional implantada en el sistema UNAMALLA.

Reflexionemos sobre algunos aspectos de la regularización. En los sistemas previos a UNAMALLA la elección de ϵ_c para la regularización es experimental, fue con la que nos obtuvimos buenos resultados toda vez que nos encontrábamos cerca de mallas convexas o bién, con pocas celdas. Por ejemplo, en el caso de la región M19 la malla inicial generada por Interpolación Transfinita no es atractiva para el uso de la regularización del funcional de suavidad ya que cuenta con muchas celdas no convexas, ver Figura 2.26.

Lo que hace necesario aplicarle un proceso previo, digamos aplicarle algún otro funcional o bién obtenida por otro método para que el número de celdas no convexas sea pequeño y entonces operar con la regularización, veáse la Figura 2.27(a).

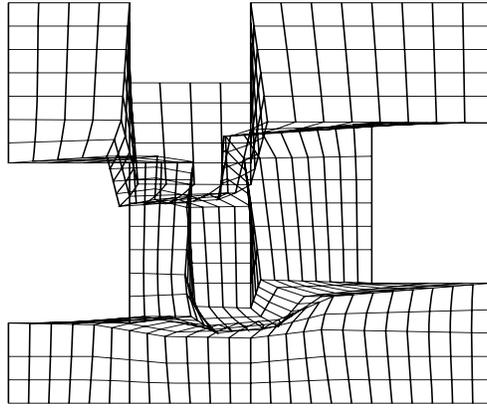


Figura 2.26: Malla sobre M19 obtenida por interpolación transfinita.

Por otra parte, no basta que las celdas no convexas sean pocas, si no que se cuente con un “espacio” donde los puntos puedan moverse y entonces la regularización pueda operar. En un caso se usó el funcional AO para luego operar el funcional regularizado; sin embargo, la regularización no trabajó muy bien, véase la Figura 2.27(b), luego de algunos pasos el optimizador no logró encontrar dirección de descenso y concluyó.

Sin embargo, usando el funcional de Área, obtuvimos mejores resultados. Para el ejemplo, puede apreciarse cómo ésta elección nos condujo a una malla convexa y suave. En la Figura 2.28(b) se puede apreciar unos pasos de la optimización del funcional de Área y en la Figura 2.29 la malla suave y convexa del funcional de Suavidad.

Como se aprecia el proceso de lograr una malla suave y convexa tanto por Ivanenko como por la estrategia de Barrera es muy laborioso, casi manual. Sin embargo la diferencia entre un procedimiento y el otro radica precisamente en la concepción del proceso discreto. No es la forma en que cada uno ataca el problema de optimización sino el concepto geométrico involucrado. Por un lado, Ivanenko observa la configuración de los puntos de la malla óptima,

$$\min_{P_{i,j} \in G} I_D \tag{2.101}$$

véase la Figura 2.17(a). Por otro lado, Barrera observa el problema de la generación de mallas como un problema de optimización de gran escala donde se encuentra involucrada la geometría de los triángulos de la malla, véase la Figura 2.18,

$$\min_{G \in \mathcal{D}} \sum_{\Delta \in G} f(\Delta) \tag{2.102}$$

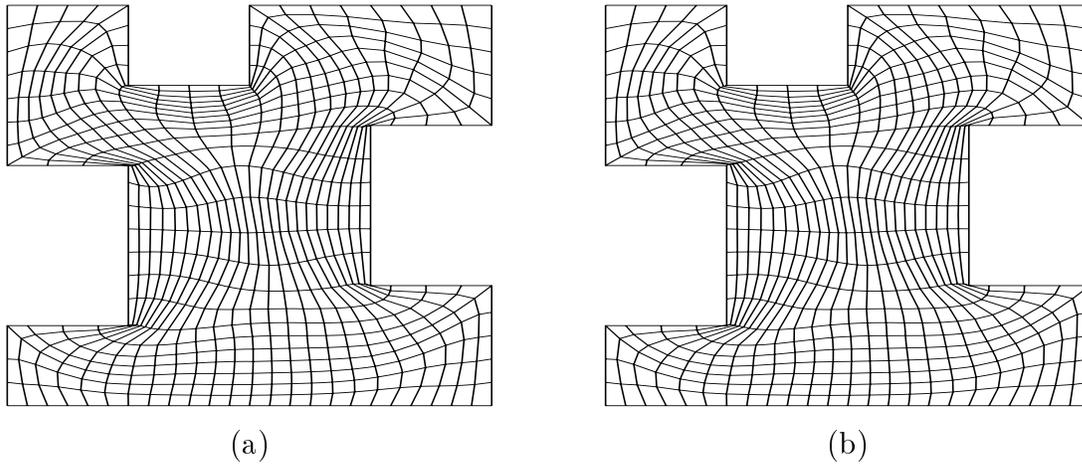


Figura 2.27: (a) Malla obtenida por el funcional AO. (b) Malla obtenida por la regularización del funcional de suavidad al usar como inicial al primero, el optimizador no encontró dirección de descenso.

donde el funcional trabaja sobre los triángulos y, a través de esa geometría da una explicación de la configuración de una malla, suave y convexa. Por otra parte, el problema de optimización así planteado, con triángulos orientados, es de gran escala, sin restricciones y sobre funciones parcialmente separables: $f(\Delta)$ sólo depende de pocos puntos, por lo que ésta forma de atacar el problema es muy interesante ya que puede ser explotada toda la estructura de la función. García [23] presenta la estructura parcialmente separable de los funcionales y propone una forma económica de resolverlo.

La necesidad de encontrar un procedimiento discreto que haga uso de la potencialidad de los mapeos armónicos nos ha llevado a desarrollar funcionales discretos del tipo de suavidad, funcionales basados en los triángulos para lograr dicho objetivo. En el siguiente capítulo daremos algunas bases que nos permita entender la influencia del área en la convexidad y uniformidad de las celdas para de esta forma reescribir y reinterpretar el funcional de suavidad a fin de desarrollar funcionales o procedimientos discretos eficientes.

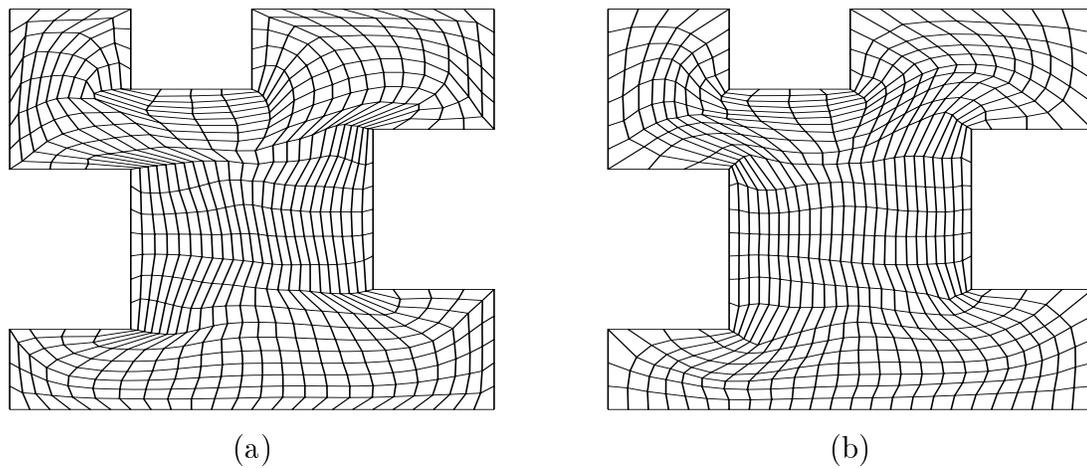


Figura 2.28: (a) Malla obtenida por el funcional Área. (b) Malla obtenida tras unos pasos de la optimización de la regularización del funcional de suavidad.

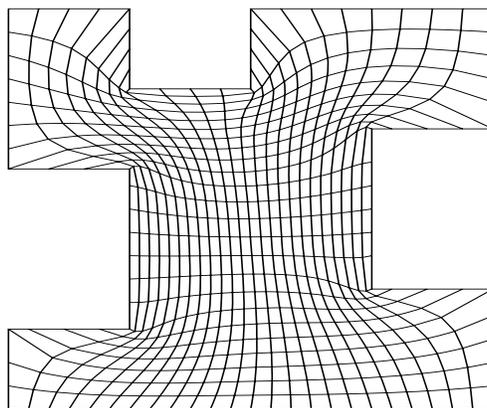


Figura 2.29: Malla obtenida por la regularización del funcional de suavidad.

Capítulo 3

Funcionales de Área continuos y discretos

Durante el presente capítulo discutiremos una forma general de construir funcionales continuos de Área con las mismas propiedades del funcional de Área clásico; de igual forma veremos una discretización de él para entonces introducir los nuevos funcionales discretos para la generación de mallas.

3.1 Funcional de Área continuo clásico

El funcional de Área continuo clásico se ha planteado como una medida del área de las celdas sobre las coordenadas del mapeo directo en Ω para uniformizar el tamaño de las celdas, siempre que ésto pueda lograrse.

Una de las propiedades geométricas que deseamos medir en una malla es la uniformidad del área de las celdas para de esta forma controlar su tamaño y no se presenten cambios inesperados entre celda y celda acorde a la medición a realizar sobre la región. Por una parte, el jacobiano del mapeo directo \mathbf{x} mide localmente el área sobre la región de estudio Ω ; por otra, la relación que guardan las coordenadas físicas y lógicas viene dada por la

$$dxdy = J d\xi d\eta \quad (3.1)$$

de ésta manera, el área de la región Ω viene dado por

$$A(\Omega) = \int \int_{\Omega} dxdx = \int_0^1 \int_0^1 J d\xi d\eta. \quad (3.2)$$

Esta relación es válida para todo mapeo directo \mathbf{x} entre B_2 y Ω que conforme la frontera, aún para aquellos mapeos cuyos segmentos curvilíneos se entrelacen.

Brackbill y Saltzman [11] propusieron en 1982 minimizar el jacobiano de la transformación controlando la variación de J a lo largo de Ω por medio de una función de peso $w = w(\xi, \eta)$. Para lograrlo sugirieron minimizar el funcional

$$I(\mathbf{x}) = \int_0^1 \int_0^1 w(\xi, \eta) J(\xi, \eta) d\xi d\eta \quad (3.3)$$

al que llamaron funcional de **Área ponderada**. La función de peso w adecúa la malla dándole más peso a aquellas subregiones de Ω donde se desea observar con mayor precisión la propiedad del área. Esta función de peso es muy socorrida si de antemano sabemos cuál es ese comportamiento a lograr a lo largo de Ω .

Nuestro interés es construir mallas convexas sobre regiones muy irregulares de manera automática, por lo que contar con una función de peso de acuerdo a una región involucra un enorme esfuerzo en cómputo y tiempo.

Steinberg y Roache [42], propusieron que la función de peso w fuese precisamente el jacobiano de la transformación, con lo cual el problema variacional que atacaron fue

$$I_a(\mathbf{x}) = \frac{1}{2} \int_0^1 \int_0^1 J^2(\xi, \eta) d\xi d\eta \quad (3.4)$$

$$= \frac{1}{2} \int_0^1 \int_0^1 (x_\xi y_\eta - x_\eta y_\xi)^2 d\xi d\eta \quad (3.5)$$

A diferencia de Brackbill y Saltzman, Steinberg y Roache formularon éste funcional de manera independiente y se basaron en un enfoque geométrico a observar en las mallas: la uniformidad de las celdas. Analicemos las ecuaciones de Euler–Lagrange para el funcional de Área.

Las ecuaciones de Euler–Lagrange para el funcional I_a son

$$\frac{d}{d\xi} \left(\frac{\partial J^2}{\partial x_\xi} \right) + \frac{d}{d\eta} \left(\frac{\partial J^2}{\partial x_\eta} \right) = 0 \quad (3.6)$$

$$\frac{d}{d\xi} \left(\frac{\partial J^2}{\partial y_\xi} \right) + \frac{d}{d\eta} \left(\frac{\partial J^2}{\partial y_\eta} \right) = 0 \quad (3.7)$$

por una parte tenemos

$$\begin{aligned} \frac{\partial J^2}{\partial x_\xi} &= 2Jy_\eta & \frac{\partial J^2}{\partial x_\eta} &= -2Jy_\xi \\ \frac{\partial J^2}{\partial y_\xi} &= -2Jx_\eta & \frac{\partial J^2}{\partial y_\eta} &= 2Jx_\xi \end{aligned} \quad (3.8)$$

desarrollando cada expresión

$$\begin{aligned} \frac{d}{d\xi} \frac{\partial J^2}{\partial x_\xi} &= \frac{d(2Jy_\eta)}{d\xi} \\ &= 2Jy_{\xi\eta} + 2y_\eta \frac{dJ}{d\xi} \end{aligned} \quad (3.9)$$

$$\begin{aligned} \frac{d}{d\eta} \frac{\partial J^2}{\partial x_\eta} &= \frac{d(-2Jy_\xi)}{d\eta} \\ &= -2Jy_{\xi\eta} - 2y_\xi \frac{dJ}{d\eta} \end{aligned} \quad (3.10)$$

$$\begin{aligned} \frac{d}{d\xi} \frac{\partial J^2}{\partial y_\xi} &= \frac{d(-2Jx_\eta)}{d\xi} \\ &= -2Jx_{\xi\eta} - 2x_\eta \frac{dJ}{d\xi} \end{aligned} \quad (3.11)$$

$$\begin{aligned} \frac{d}{d\eta} \frac{\partial J^2}{\partial y_\eta} &= \frac{d(2Jx_\xi)}{d\eta} \\ &= 2Jx_{\xi\eta} + 2x_\xi \frac{dJ}{d\eta} \end{aligned} \quad (3.12)$$

con esto, las ecuaciones de Euler–Lagrange para el funcional de Área se escriben como

$$2 \left(y_\eta \frac{dJ}{d\xi} - y_\xi \frac{dJ}{d\eta} \right) = 0 \quad (3.13)$$

$$2 \left(-x_\eta \frac{dJ}{d\xi} + x_\xi \frac{dJ}{d\eta} \right) = 0 \quad (3.14)$$

o bien, en forma matricial por

$$2 \begin{pmatrix} y_\xi & y_\eta \\ x_\xi & x_\eta \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \frac{dJ}{d\xi} \\ \frac{dJ}{d\eta} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (3.15)$$

el cual es un sistema cuasi-lineal y no elíptico. Sobre éste sistema no se cuenta con prueba de existencia y de unicidad sobre una región Ω arbitraria. El hecho de que el sistema es no elíptico se ve reflejado en las mallas al no ser éstas suaves.

Steinberg y Knupp [30] observaron sobre un conjunto de regiones de prueba (la Galería de Roache) que no es posible obtener mallas con la característica de que cuenten con área uniforme y sean convexas. Resolvieron el sistema de Euler–Lagrange por medio del método iterativo SOR [10] con un *stencil* de nueve puntos y observaron que el proceso diverge en pocos pasos presumiblemente porque en esas regiones no era posible encontrar una malla uniforme en tamaño de celda.

3.2 Funcionales de Área continuos

El funcional de Área continuo puede ser generalizado de manera que conservemos la propiedad del funcional de Área clásico: área uniforme.

Consideremos a $L = L(x)$ una función escalar diferenciable dos veces y J el jacobiano de la transformación, . Nuestro interés se centra en observar las propiedades que pediremos a L de manera que la expresión

$$\int_{B_2} L(J) d\xi d\eta \quad (3.16)$$

represente una funcional con propiedades geométricas semejantes al funcional de Área.

Consideremos las ecuaciones de Euler–Lagrange del problema variacional asociado

$$\frac{d}{d\xi} \left(\frac{\partial L(J)}{\partial x_\xi} \right) + \frac{d}{d\eta} \left(\frac{\partial L(J)}{\partial x_\eta} \right) = 0 \quad (3.17)$$

$$\frac{d}{d\xi} \left(\frac{\partial L(J)}{\partial y_\xi} \right) + \frac{d}{d\eta} \left(\frac{\partial L(J)}{\partial y_\eta} \right) = 0 \quad (3.18)$$

Por claridad denotemos

$$L' = \frac{dL}{dJ}, \quad L'' = \frac{d^2L}{dJ^2} \quad (3.19)$$

con esto, cada elemento de las ecuaciones las expresamos como

$$\begin{aligned} \frac{d}{d\xi} \left(\frac{\partial L}{\partial x_\xi} \right) &= \frac{d}{d\xi} \left(L' \frac{\partial J}{\partial x_\xi} \right) \\ &= \frac{d}{d\xi} (y_\eta L') \end{aligned}$$

$$= y_{\xi\eta}L' + y_{\eta}L''\frac{dJ}{d\xi} \quad (3.20)$$

$$\begin{aligned} \frac{d}{d\eta} \left(\frac{\partial L}{\partial x_{\eta}} \right) &= \frac{d}{d\eta} \left(L' \frac{\partial J}{\partial x_{\eta}} \right) \\ &= -\frac{d}{d\eta}(y_{\xi}L') \\ &= -y_{\xi\eta}L' - y_{\xi}L''\frac{dJ}{d\eta} \end{aligned} \quad (3.21)$$

$$\begin{aligned} \frac{d}{d\xi} \left(\frac{\partial J}{\partial y_{\xi}} \right) &= \frac{d}{d\xi} \left(L' \frac{\partial J}{\partial y_{\xi}} \right) \\ &= -\frac{d}{d\xi}(x_{\eta}L') \\ &= -x_{\xi\eta}L' - x_{\eta}L''\frac{dJ}{d\xi} \end{aligned} \quad (3.22)$$

$$\begin{aligned} \frac{d}{d\eta} \left(\frac{\partial L}{\partial y_{\eta}} \right) &= \frac{d}{d\eta} \left(L' \frac{\partial J}{\partial y_{\eta}} \right) \\ &= \frac{d}{d\eta}(x_{\xi}L') \\ &= x_{\xi\eta}L' + x_{\xi}L''\frac{dJ}{d\eta} \end{aligned} \quad (3.23)$$

con esto, las ecuaciones Euler–Lagrange de (3.18) son

$$\begin{aligned} \frac{d}{d\xi} \left(\frac{\partial L(J)}{\partial x_{\xi}} \right) + \frac{d}{d\eta} \left(\frac{\partial L(J)}{\partial x_{\eta}} \right) &= \\ &= L''(y_{\eta}\frac{dJ}{d\xi} - y_{\xi}\frac{dJ}{d\eta}) = 0 \end{aligned} \quad (3.24)$$

$$\begin{aligned} \frac{d}{d\xi} \left(\frac{\partial L(J)}{\partial y_{\xi}} \right) + \frac{d}{d\eta} \left(\frac{\partial L(J)}{\partial y_{\eta}} \right) &= \\ &= L''(x_{\eta}\frac{dJ}{d\xi} - x_{\xi}\frac{dJ}{d\eta}) = 0 \end{aligned} \quad (3.25)$$

y en forma matricial

$$L'' \begin{pmatrix} y_\eta \frac{dJ}{d\xi} - y_\xi \frac{dJ}{d\eta} \\ x_\eta \frac{dJ}{d\xi} - x_\xi \frac{dJ}{d\eta} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (3.26)$$

las cuales son idénticas al funcional de área clásico excepto por el factor L'' . De aquí, se concluye que una solución para el funcional de área clásico será solución para cualquier otro funcional continuo de área de la forma (3.16). De igual manera, si G^* es una malla solución de un funcional continuo de área con la propiedad de que $L'' \neq 0$ entonces G^* será solución del funcional de área clásico.

Haciendo un pequeño cambio en el orden de las ecuaciones últimas podemos reescribirlas como

$$L'' \begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \frac{dJ}{d\xi} \\ \frac{dJ}{d\eta} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (3.27)$$

Si consideramos mapeos no singulares, es decir con

$$J = \det \begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix} \neq 0, \quad (3.28)$$

las ecuaciones de Euler–Lagrange resultan

$$L'' \begin{pmatrix} \frac{dJ}{d\xi} \\ \frac{dJ}{d\eta} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (3.29)$$

Ahora bien, si para estos mapeos nuestra función L es de tal forma que $L'' \neq 0$, el jacobiano J de la transformación será constante en Ω ; esto es, la malla así generada por el funcional continuo será uniforme en Ω .

Cabe señalar que la forma de la función L es muy importante para lograr la propiedad de área uniforme ya que nos indica la aportación de J a lo largo de Ω .

Algunos ejemplos de funcionales de Área del tipo señalado son

1. $L(J) = (J - \bar{\alpha})^4$
2. $L(J) = J^{-1}$
3. $L(J) = \frac{1}{J-\omega}$, con $\omega \in \mathbb{R}$ adecuado.
4. $L(J) = \frac{1}{J+\omega}$, con $\omega \in \mathbb{R}$ adecuado.

En el primer ejemplo $\bar{\alpha}$ representa al valor promedio de α de un triángulo (dos veces el área, ver Capítulo 1 y 2). En el ejemplo 2, el funcional tiene sentido sobre mapeos no singulares, $J \neq 0$. En el ejemplo 3 el valor de ω debe ser tal que $J > \omega$ sobre las mallas del conjunto que puede ser descrito con esta propiedad. De igual forma usar el Lagrangiano en el ejemplo 4 nos obliga a contemplar mallas tales que $J > -\omega$, siempre que ésto pueda ocurrir. En el Capítulo siguiente abordaremos las funciones construidas por la discretización de estos funcionales y veremos para qué valores de ω es posible definirlos.

En el funcional clásico de Área su Lagrangiano es J^2 , su formulación geométrica es lograr uniformidad en área de las celdas de la malla [42], ésta misma idea intuitiva es usada para describir de manera cualitativa la propiedad geométrica que desea ser medida usando los anteriores ejemplos. En la sección siguiente veremos, a través de los funcionales discretos, ésta aseveración.

La forma discreta de los funcionales por medio de la discretización de Barrera-Ivanenko nos permiten lograr mallas cercanas a la uniformidad sobre regiones poligonales irregulares, veremos la discretización lograda por Barrera y señalaremos algunas de sus propiedades. Para lograrlo veamos un funcional discreto que dió pie a este estudio.

3.3 Funcional discreto de Área de Castillo

Castillo en 1987 introduce lo que llamamos hoy en día los funcionales discretos al interpretar los funcionales continuos de Área, Longitud y de Ortogonalidad de manera directa sobre las celdas y líneas de la malla dando lugar a una función suma de funciones sobre las celdas o líneas de la malla. Barrera y Castillo propusieron resolver el problema por métodos de optimización de gran escala y hacer uso de la estructura de la función, el gradiente y la hessiana e implantarlo en un sistema computacional.

Veamos en qué consiste el método y mencionemos ligeramente su dificultad. Consideremos la celda c_{ij} de una malla de $m \times n$ sobre una región poligonal Ω , ver Figura 3.1.

Consideremos el área del cuadrilátero c_{ij} , esto es

$$A_{i,j} = \text{área}(c_{i,j}) \quad (3.30)$$

El funcional discreto de Área lo define Castillo como la suma del cuadrado del área de todas las celdas de la malla

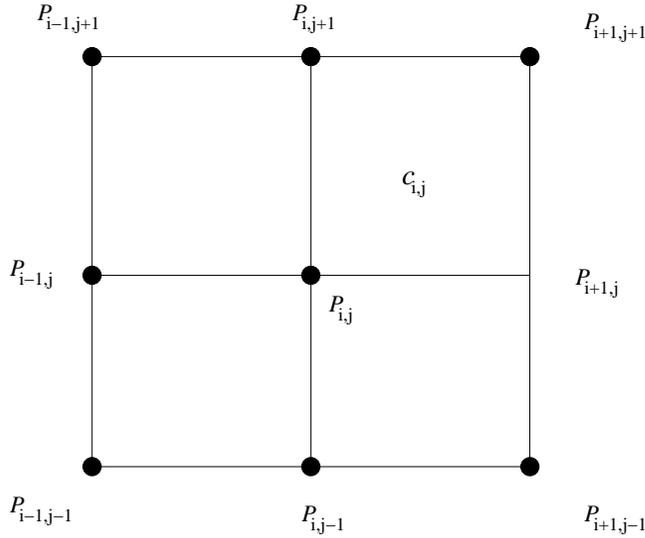


Figura 3.1: Celdas alrededor del punto $P_{i,j}$

$$F_A = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} A_{i,j}^2 \quad (3.31)$$

y el problema de gran escala surge al minimizar la función sobre los puntos interiores de la malla

$$\min_{P_{i,j}} F_A \quad (3.32)$$

Por una parte, esta función es muy sencilla y el número de puntos interiores son las incógnitas del problema de optimización, un problema de gran escala cuando la malla es de dimensión grande.

La estructura de la funcional discreta está descrita en [37] y [5], donde se señala a detalle la estructura del gradiente y la matriz hessiana para su implantación en un sistema computacional. Interesado por el problema de gran escala, Barrera apoyó a Castillo en el estudio de la función a minimizar atacándolo a través de Gradiente Conjugado y otros métodos de optimización.

Una de las propiedades que encontró Castillo del funcional, es que una malla con la misma área en todas sus celdas es óptimo del funcional [13], siempre que esta pueda construirse. Esto es, que en principio éste funcional pudiera conducirnos a mallas uniformes. Sin embargo, esto no es así.

Tiempo después de su trabajo doctoral, Castillo considerando un problema modelo [15], construyó configuraciones que al ser optimizadas le llevaba a puntos no

adecuados, a configuraciones donde la malla es no convexa.

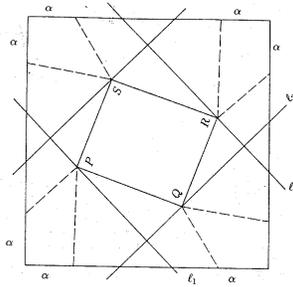


Figura 3.2: Problema modelo atacado por Castillo [16].

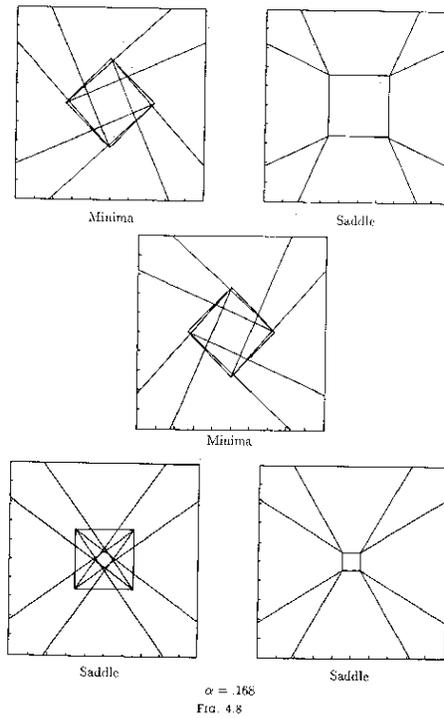


Figura 3.3: Configuraciones obtenidas por el problema modelo de Castillo [16].

Las conclusiones de Castillo fueron que se debía combinar área con longitud para obtener configuraciones aceptables.

Barrera y Pérez [4] retoman las ideas de los funcionales discretos, pero a través de una discretización de los continuos y demuestran, en forma independiente a Castillo, que su funcional falla para producir mallas convexas y muestran un ejemplo de una malla con área uniforme en celdas pero que es no convexa, ver Figura 3.4.

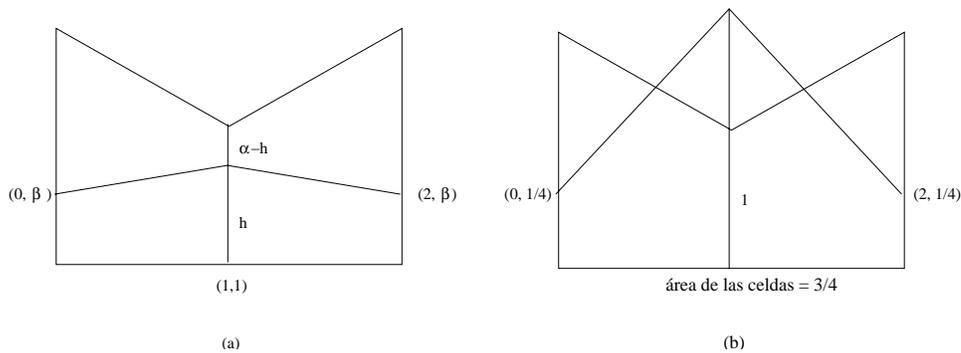


Figura 3.4: (a) Problema modelo. (b) Solución por el funcional de Castillo.

En 1988 Barrera y Pérez trabajan en conjunto para resolver un problema de flujo sobre la bahía de la Habana y consideran una variante del funcional de área pero con una observación interesante: los elementos deben ser considerados con orientación positiva. Por otra parte ya no trabajan sobre el área de la celda sino a través de los triángulos que ésta describe [3] y con esto, la formulación del funcional discreto de área se vuelve muy interesante.

$$F_t = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \sum_{k=1}^4 (\alpha_{i,j}^{(k)})^2 \quad (3.33)$$

De igual forma, se interesan por distintos métodos de optimización de gran escala como lo son el BFGS con Memoria Limitada y métodos de Newton Truncado para aprovechar la estructura del gradiente y de la matriz hessiana para de esta forma, hacer eficiente el proceso de optimización.

En el siguiente apartado describiremos la discretización del funcional de Área clásico, siguiendo la idea de la discretización de Winslow para obtener el funcional discreto t -Área de Barrera-Pérez.

3.4 Funcional de Área discreto, t -Área

De la misma forma en que procedimos a discretizar el funcional continuo de Suavidad, hagámoslo ahora con el funcional de Área clásico.

El funcional de Área lo podemos escribir como una suma del funcional restringido a cada una de las celdas de B_2

$$\begin{aligned}
 I_a(\mathbf{x}) &= \sum_{i,j} \int_{B_{i,j}} J^2 d\xi d\eta \\
 &= \sum_{i,j} \int_{B_{i,j}} (\mathbf{x}_\xi^t J_2 \mathbf{x}_\eta)^2 d\xi d\eta
 \end{aligned}
 \tag{3.34}$$

y para cada celda $B_{i,j}$ el mapeo restringido aproximararlo por la función bilineal $\mathbf{r} = \mathbf{r}(\xi, \eta)$, es decir, con la forma más simple de mapear dos cuadriláteros, obteniendo

$$I_a(\mathbf{x}) \approx \sum_{i,j} \int_{B_{i,j}} (\mathbf{r}_\xi^t J_2 \mathbf{r}_\eta)^2 d\xi d\eta.
 \tag{3.35}$$

Siguiendo las ideas del capítulo anterior sea G una malla de $m \times n$ en B_2 uniforme en ξ y η y por claridad consideremos la celda $c_{i,j}$ como el cuadrilátero compuesto por los puntos P, Q, R, S , ver Figura 3.5.

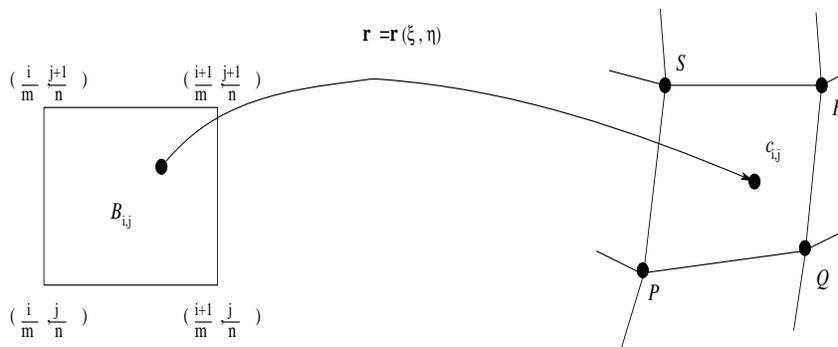


Figura 3.5: Mapeo bilineal entre $B_{i,j}$ y $c_{i,j}$

Escribiendo el mapeo bilineal entre $B_{i,j}$ y el cuadrilátero P, Q, R, S en la forma

$$\mathbf{r}(\xi, \eta) = \mathbf{A} + \mathbf{B}(\xi - \frac{i}{m}) + \mathbf{C}(\eta - \frac{j}{n}) + \mathbf{D}(\xi - \frac{i}{m})(\eta - \frac{j}{n})
 \tag{3.36}$$

los coeficientes del mapeo son determinados por la correspondencia entre los puntos, así tendremos que

$$\mathbf{A} = P$$

$$\begin{aligned}
\mathbf{B} &= m(Q - P) \\
\mathbf{C} &= n(S - P) \\
\mathbf{D} &= mn(R - Q + P - S)
\end{aligned} \tag{3.37}$$

Una vez hecho esto, usemos una regla de cuadratura para aproximar la integral sobre $B_{i,j}$ obteniendo

$$\begin{aligned}
\int_{B_{i,j}} (\mathbf{r}_\xi^t J_2 \mathbf{r}_\eta)^2 &\approx \frac{1}{4} \left[\left(\mathbf{r}_\xi^t \left(\frac{i}{m}, \frac{j}{n} \right) J_2 \mathbf{r}_\eta \left(\frac{i}{m}, \frac{j}{n} \right) \right)^2 \right. \\
&\quad + \left(\mathbf{r}_\xi^t \left(\frac{i+1}{m}, \frac{j}{n} \right) J_2 \mathbf{r}_\eta \left(\frac{i+1}{m}, \frac{j}{n} \right) \right)^2 \\
&\quad + \left(\mathbf{r}_\xi^t \left(\frac{i+1}{m}, \frac{j+1}{n} \right) J_2 \mathbf{r}_\eta \left(\frac{i+1}{m}, \frac{j+1}{n} \right) \right)^2 \\
&\quad \left. + \left(\mathbf{r}_\xi^t \left(\frac{i}{m}, \frac{j+1}{n} \right) J_2 \mathbf{r}_\eta \left(\frac{i}{m}, \frac{j+1}{n} \right) \right)^2 \right]
\end{aligned} \tag{3.38}$$

Por una parte, las parciales de \mathbf{r} son

$$\begin{aligned}
\mathbf{r}_\xi &= m(Q - P) + mn(R + P - Q - S)\left(\eta - \frac{j}{n}\right) \\
\mathbf{r}_\eta &= n(S - P) + mn(R + P - Q - S)\left(\xi - \frac{i}{m}\right)
\end{aligned} \tag{3.39}$$

y sus valores en las esquinas de $B_{i,j}$ son

$$\begin{aligned}
\mathbf{r}_\xi \left(\frac{i}{m}, \frac{j}{n} \right) &= m(Q - P) & \mathbf{r}_\eta \left(\frac{i}{m}, \frac{j}{n} \right) &= n(S - P) \\
\mathbf{r}_\xi \left(\frac{i+1}{m}, \frac{j}{n} \right) &= m(Q - P) & \mathbf{r}_\eta \left(\frac{i+1}{m}, \frac{j}{n} \right) &= n(R - Q) \\
\mathbf{r}_\xi \left(\frac{i+1}{m}, \frac{j+1}{n} \right) &= m(R - S) & \mathbf{r}_\eta \left(\frac{i+1}{m}, \frac{j+1}{n} \right) &= n(R - Q) \\
\mathbf{r}_\xi \left(\frac{i}{m}, \frac{j+1}{n} \right) &= m(R - S) & \mathbf{r}_\eta \left(\frac{i}{m}, \frac{j+1}{n} \right) &= n(S - P)
\end{aligned} \tag{3.40}$$

Con esto, reescribimos el primer elemento de (3.38)

$$\mathbf{r}_\xi^t \left(\frac{i}{m}, \frac{j}{n} \right) J_2 \mathbf{r}_\eta \left(\frac{i}{m}, \frac{j}{n} \right) = mn(Q - P)^t J_2 (S - P) \tag{3.41}$$

y de manera semejante tendremos los restantes términos

$$\mathbf{r}_\xi^t\left(\frac{i+1}{m}, \frac{j}{n}\right) J_2 \mathbf{r}_\eta\left(\frac{i+1}{m}, \frac{j}{n}\right) = mn(Q-P)^t J_2(R-Q) \quad (3.42)$$

$$\mathbf{r}_\xi^t\left(\frac{i+1}{m}, \frac{j+1}{n}\right) J_2 \mathbf{r}_\eta\left(\frac{i+1}{m}, \frac{j+1}{n}\right) = mn(R-S)^t J_2(R-Q) \quad (3.43)$$

$$\mathbf{r}_\xi^t\left(\frac{i}{m}, \frac{j+1}{n}\right) J_2 \mathbf{r}_\eta\left(\frac{i}{m}, \frac{j+1}{n}\right) = mn(R-S)^t J_2(S-P) \quad (3.44)$$

con esto, la aproximación al funcional de Área sobre $B_{i,j}$ la escribimos en términos de los puntos P, Q, R, S de la forma:

$$\int_{B_{i,j}} (\mathbf{r}_\xi^t J_2 \mathbf{r}_\eta)^2 \approx \frac{1}{4} m^2 n^2 \left[((Q-P)^t J_2(S-P))^2 + ((Q-P)^t J_2(R-Q))^2 + ((R-S)^t J_2(R-Q))^2 + ((R-S)^t J_2(S-P))^2 \right] \quad (3.45)$$

De la misma forma en que procedimos en el capítulo anterior, hagamos un examen atento de los cuatro triángulos que pueden ser descritos sobre un rectángulo. En nuestro caso será sobre cada celda $c_{i,j}$, con orientación positiva, tal como se muestra en la Figura 3.6.

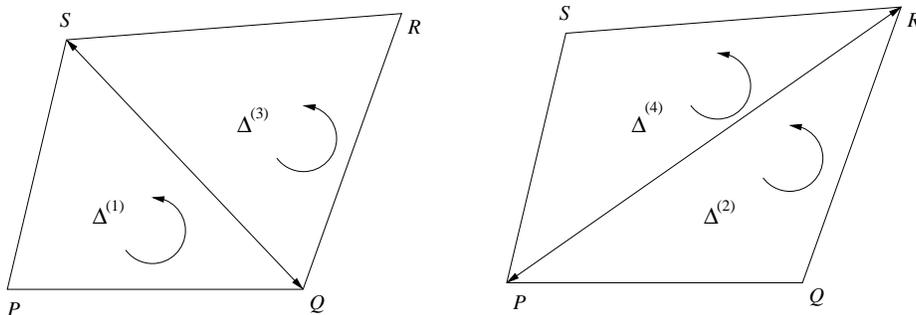


Figura 3.6: Triángulos de una celda.

Dado que los triángulos cuenta con orientación positiva, el área de cada uno de ellos está representarlo por la expresión

$$2 \text{ área}(P, Q, S) = \det(Q - P, S - P) = (Q - P)^t J_2(S - P) \quad (3.46)$$

lo que nos lleva a escribir la integral de Área restringida a la celda $B_{i,j}$ en forma compacta como

$$\int_{B_{i,j}} \mathbf{r}_\xi^t J_2 \mathbf{r}_\eta \approx \frac{1}{4} m^2 n^2 [(2 \text{ área}(P, Q, S))^2 + (2 \text{ área}(Q, R, P))^2 + (2 \text{ área}(R, S, Q))^2 + (2 \text{ área}(S, P, R))^2] \quad (3.47)$$

Podemos simplificar aún más la expresión anterior escribiéndola como función de triángulos en la forma

$$\int_{B_{i,j}} \mathbf{r}_\xi^t J_2 \mathbf{r}_\eta \approx \sum_{k=1}^4 f(\Delta^{(k)}) \quad (3.48)$$

donde

$$f(\Delta) = \alpha^2(\Delta) \quad (3.49)$$

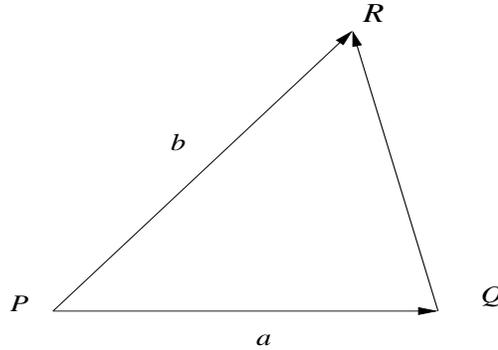


Figura 3.7: Triángulo genérico de una celda.

y

$$\alpha(\Delta) = \frac{1}{2} \det(Q - P, R - P) = \text{área}(P, Q, R). \quad (3.50)$$

no perdamos de vista que P, Q, R representa el triángulo $\Delta^{(k)}$ en curso, ver Figura 3.7.

Nótese que en esta forma de escribir el funcional hemos omitido la cantidad $m^2 n^2$ de (3.47) ya que sólo es un factor. Este factor lo tomaremos en cuenta cuando veamos la necesidad de normalizar los funcionales para usarlos en combinaciones entre ellos.

Ahora bien, la integral de Área, al igual que la integral de Winslow, la hemos aproximado por una función o colección de funciones, que dependen únicamente de los nodos interiores de la malla.

$$\begin{aligned} I_a \approx I_{\mathcal{D}_A} &= \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \sum_{k=1}^4 f(\Delta_{ij}^{(k)}) \\ &= \sum_{q=1}^N f(\Delta_q) \end{aligned} \tag{3.51}$$

donde N es el número de triángulos de la malla. Con esto, hemos transformado el problema continuo de Área en un problema de minimización sobre los triángulos de la malla identificando aquellas configuraciones de mallas cuyos triángulos satisfagan el problema de gran escala

$$\min_{G \in \mathcal{D}} I_{\mathcal{D}_A} \tag{3.52}$$

donde \mathcal{D} es el conjunto de mallas sobre Ω de dimensión $m \times n$. Un problema exento de restricciones.

Este funcional coincide con la propuesta de Barrera-Pérez para el funcional de área midiendo los triángulos orientados de la malla. Por una parte el funcional de Barrera-Pérez es una discretización simple del funcional continuo de área y por la otra, cuenta con propiedades muy atractivas descritas por sus autores en [4].

Siguiendo las ideas de la discretización del funcional de Área y atendiendo a la forma de los funcionales discretos: sumas de funciones simples, veamos una manera de representar funcionales discretos que miden el área de los triángulos.

3.5 Funcionales discretos de Área

En lo que resta del capítulo mostraremos una forma de construir funcionales discretos para la generación de mallas basados en las propiedades del funcional de Área clásico discretizado, t -Área. Para lograrlo extendamos la noción del área uniforme a través de una función sobre el área de los triángulos y veamos qué propiedades debe tener la función para que el problema óptimo cuente entre sus puntos críticos a la malla uniforme.

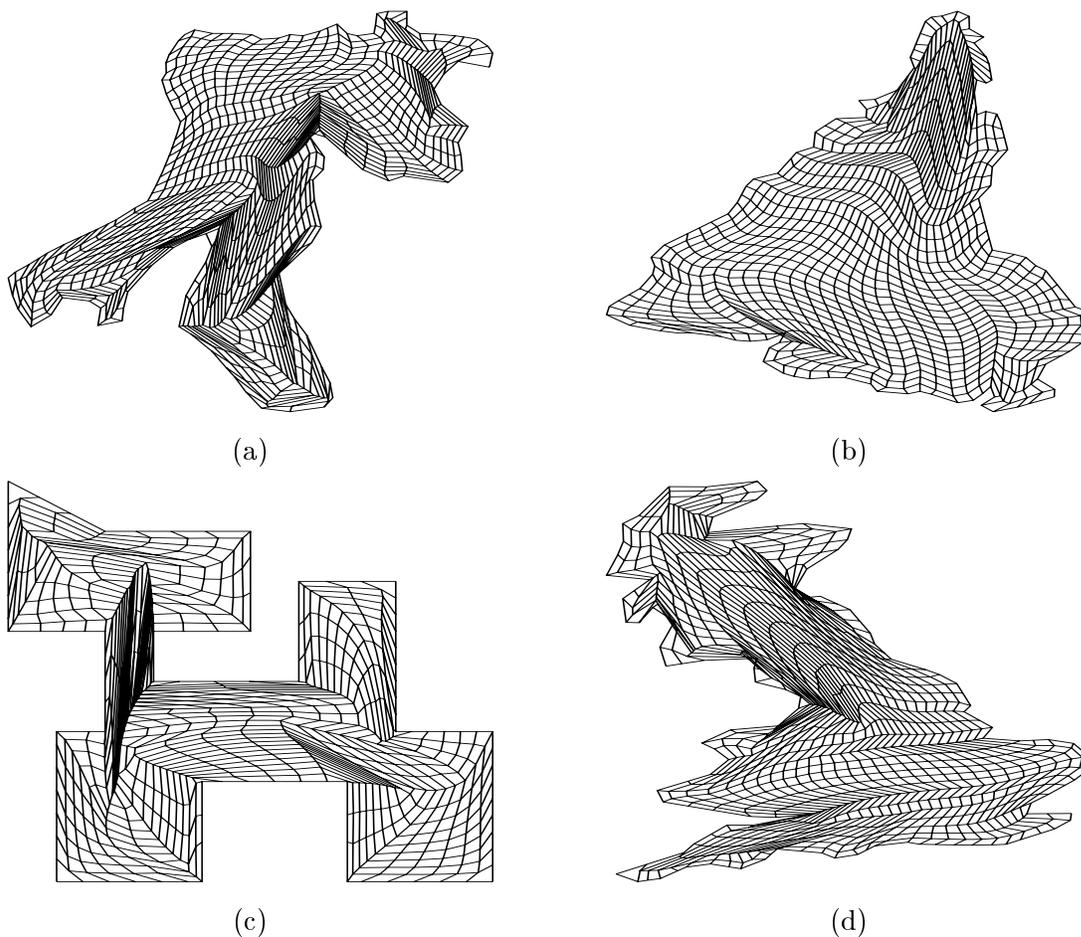


Figura 3.8: Mallas generadas por el funcional de t -Área. (a) región de la bahía de la Habana, (b) región de Sudamérica, (c) región del Gato y (d) región del Reino Unido. Ninguna es convexa.

3.6 Una forma general de los funcionales discretos de Área

La cantidad α_q (dos veces el área) de los triángulos de una malla en Ω con orientación positiva tiene la propiedad de que

$$\sum_{q=1}^N \alpha_q = 4 \cdot \text{área}(\Omega) \quad (3.53)$$

ver [4]. Esta cantidad es independiente del grado de convexidad o “irregularidad”

de la región de estudio; así un problema de optimización que involucre el área de los triángulos de la malla puede ser tratado como un problema de optimización sin restricciones, en la siguiente sección enfatizaremos estas ideas.

Si $f : \mathbb{R} \rightarrow \mathbb{R}$ es una función continua, consideremos la expresión

$$F = \sum_{q=1}^N f(\alpha_q) \tag{3.54}$$

con α_q dos veces el área del triángulo. Esta expresión para F es una función del área de los triángulos de la malla que pondera el área de los triángulos a lo largo de la región. Habiendo definido f deseamos resolver el problema de optimización

$$\min_{G \in \mathcal{D}} F = \sum_{q=1}^N f(\alpha_q). \tag{3.55}$$

Nótese que no se pide más que continuidad en la función f ; en la práctica nuestras funciones de estudio son continuamente diferenciables.

En principio la expresión (3.54) podría considerar celdas y no solamente triángulos. Por ejemplo, el funcional de Castillo [13], es un funcional de área por celdas, pero como de él han demostrado Barrera y Pérez [3] el funcional de Castillo no es uno a uno por lo que un punto crítico podría ser una malla doblada; de esto hemos platicado en apartados anteriores.

La proposición siguiente nos da una idea de cómo construir funcionales del tipo (3.54) y a partir de las propiedades que se espera observar.

Proposición 3.1 *Si G^* es una malla tal que*

$$\alpha_q = \bar{\alpha}, \quad \text{para } q = 1, 2, \dots, N \tag{3.56}$$

y F es un funcional de área como en (3.54), entonces

- a) G^* es un punto crítico de F .*
- b) Si f es tal que $f'' \geq 0$ para toda malla G , G^* es mínimo de F .*

Demostración

Si z_r es un punto interior de una malla de dimensión $m \times n$ sobre Ω , la derivada parcial de F con respecto a z_r es

$$\frac{\partial F}{\partial z_r} = \sum_{q=1}^N f'(\alpha_q) \frac{\partial \alpha_q}{\partial z_r} \quad (3.57)$$

y los elementos rs de la matriz Hessiana de F son

$$\frac{\partial^2 F}{\partial z_r \partial z_s} = \sum_{q=1}^N \left[f''(\alpha_q) \frac{\partial \alpha_q}{\partial z_r} \frac{\partial \alpha_q}{\partial z_s} + f'(\alpha_q) \frac{\partial^2 \alpha_q}{\partial z_r \partial z_s} \right] \quad (3.58)$$

de manera compacta la matriz Hessiana de F la representamos por

$$H = A^t \Gamma A + \sum_{q=1}^N f'(\alpha_q) B_q \quad (3.59)$$

donde

$$\begin{aligned} \Gamma &= \text{diag}(f''(\alpha_1), f''(\alpha_2), \dots, f''(\alpha_N)) \\ A_{i,j} &= \frac{\partial \alpha_j}{\partial z_i} \\ B_q &= \text{matriz Hessiana de } \alpha_q \end{aligned} \quad (3.60)$$

Habiendo descrito la matriz Hessiana de F consideremos la suma de los triángulos de una malla G sobre Ω

$$B = \sum_{q=1}^N \alpha_q = 4 \cdot \text{área}(\Omega) \quad (3.61)$$

la cual es constante para cualquier malla sobre Ω con triángulos orientados positivamente, dando a lugar que las derivadas parciales de esta cantidad con respecto a los puntos interiores son cero

$$\frac{\partial}{\partial z_r} \left(\sum_{q=1}^N \alpha_q \right) = 0, \quad \text{para } r = 1, 2, \dots, (m-2)(n-2) \quad (3.62)$$

válido para cualquier malla y de manera particular para G^* , la malla de triángulos con área uniforme. De (3.57) tenemos que

$$\frac{\partial F}{\partial z_r}(G^*) = \left(\sum_{q=1}^N f'(\alpha_q) \frac{\partial \alpha_q}{\partial z_r} \right) (G^*)$$

$$\begin{aligned}
 &= \sum_{q=1}^N f'(\hat{\alpha}) \frac{\partial \alpha_q}{\partial z_r}(G^*) \\
 &= f'(\hat{\alpha}) \left(\sum_{q=1}^N \frac{\partial \alpha_q}{\partial z_r} \right) (G^*) \\
 &= 0
 \end{aligned} \tag{3.63}$$

Con esta argumentación hemos probado la primera parte de la proposición, que G^* la malla ideal, la de los triángulos de área uniforme, es punto crítico de un funcional de Área (3.54). Sobre esta malla, la matriz Hessiana de α_q es cero y entonces lo es la Hessiana de la suma

$$\frac{\partial^2}{\partial z_r \partial z_s} \left(\sum_{q=1}^N \alpha_q \right) (G^*) = 0, \quad \text{para } r, s = 1, 2, \dots, (m-2)(n-2) \tag{3.64}$$

por lo que la matriz Hessiana de F en G^* toma la forma

$$H(G^*) = A^t \Gamma A(G^*) \tag{3.65}$$

ya que Γ en G^* es una matriz semi-definida positiva (es una matriz diagonal con elementos no negativos) también lo es H . Así, queda demostrado que G^* es un mínimo de F .

l.q.q.d.

Lo interesante de ésta proposición es relacionar la malla ideal G_A^* , la malla con uniformidad en área de las celdas y el valor óptimo de los funcionales de área, de manera que puedan ser construidos funcionales discretos F con la propiedad de que si existe una malla uniforme en Ω ésta sea punto crítico y tal vez mínimo de F .

Por supuesto habrán funcionales de Área más sencillos y útiles que otros y para cada uno de ellos con la característica (3.54) podría el optimizador en principio conducirnos al óptimo o estar cerca; algunos de estos ejemplos funcionarán mejor que otros y en algunos tendremos dificultad para implantarlo. Concluiremos el capítulo discutiendo dos ejemplos de funcionales de Área de sumo interés.

En la sección siguiente veremos dos funcionales de este tipo con ciertas particularidades.

3.6.1 Solución al problema de optimización

En los trabajos [14] y [4] se desarrolló una manera de interpretar el funcional discreto de longitud, veamos ahora un modelo análogo de optimización que nos lleve a trabajar los funcionales de Área.

Ahora consideremos el problema de optimización expresado como

$$\begin{aligned} \min F &= \sum_{q=1}^N f(\omega_q) \\ \text{sujeto a } \sum_{q=1}^N \omega_q &= B \end{aligned} \quad (3.66)$$

donde B es una constante y f es una función uno a uno dos veces continuamente diferenciable.

Este problema de optimización tiene por punto crítico a

$$\omega_q = \frac{B}{N}, \quad q = 1, \dots, N \quad (3.67)$$

para los detalles de ésta afirmación referimos el trabajo [24].

Lo interesante de esta formulación es relacionarla con el área de los triángulos. Por una parte, los triángulos sobre una malla con orientación positiva tiene la propiedad de que

$$\sum_{q=1}^N \alpha_q = 4 \cdot \text{área}(\Omega) \quad (3.68)$$

así, los funcionales de área sobre los triángulos orientados de una malla tienen por punto crítico a aquella configuración de puntos para los cuales se satisface que

$$\alpha_q = \frac{4 \cdot \text{área}(\Omega)}{N}, \quad q = 1, \dots, N \quad (3.69)$$

esto es, una malla orientada con área uniforme en los triángulos es un punto crítico de los funcionales discretos de área. El haber pedido que los triángulos cuenten con orientación positiva es fundamental para este resultado.

Veamos algunos ejemplos de funcionales discretos de área.

3.7 Ejemplos de funcionales discretos de Área

Dos funcionales discretos de Área muy útiles son el funcional de Área cuarto promedio A4P y el funcional discreto de Área inversa. El funcional A4P (Área cuarto promedio)

[48] tiene por representación

$$F_{AAP} = \sum_{q=1}^N (\alpha_q - \bar{\alpha})^4 \quad (3.70)$$

éste funcional discreto intenta que la variación de los triángulos con respecto al valor del área promedio (valor del área de un triángulo genérico de la malla ideal, la malla de área uniforme) sea pequeño. Tinoco reporta (Tabla 3, Capítulo 4 [48]) que éste funcional discreto produce mallas convexas o bien cercanas a la convexidad en algunos casos en que el funcional discreto t -Área no lo logra; sin embargo, aún sobre regiones muy irregulares éste funcional falla para lograr una malla convexa y óptima en el sentido (3.54).

La elección del exponente en (3.70) fue experimental, representa una medida fuerte a observar sobre la malla, y ha sido con el cual se han logrado buenos resultados al magnificar las desviaciones del área de los triángulos para entonces controlar el área esperada de una mejor manera y claro, sobre toda la región.

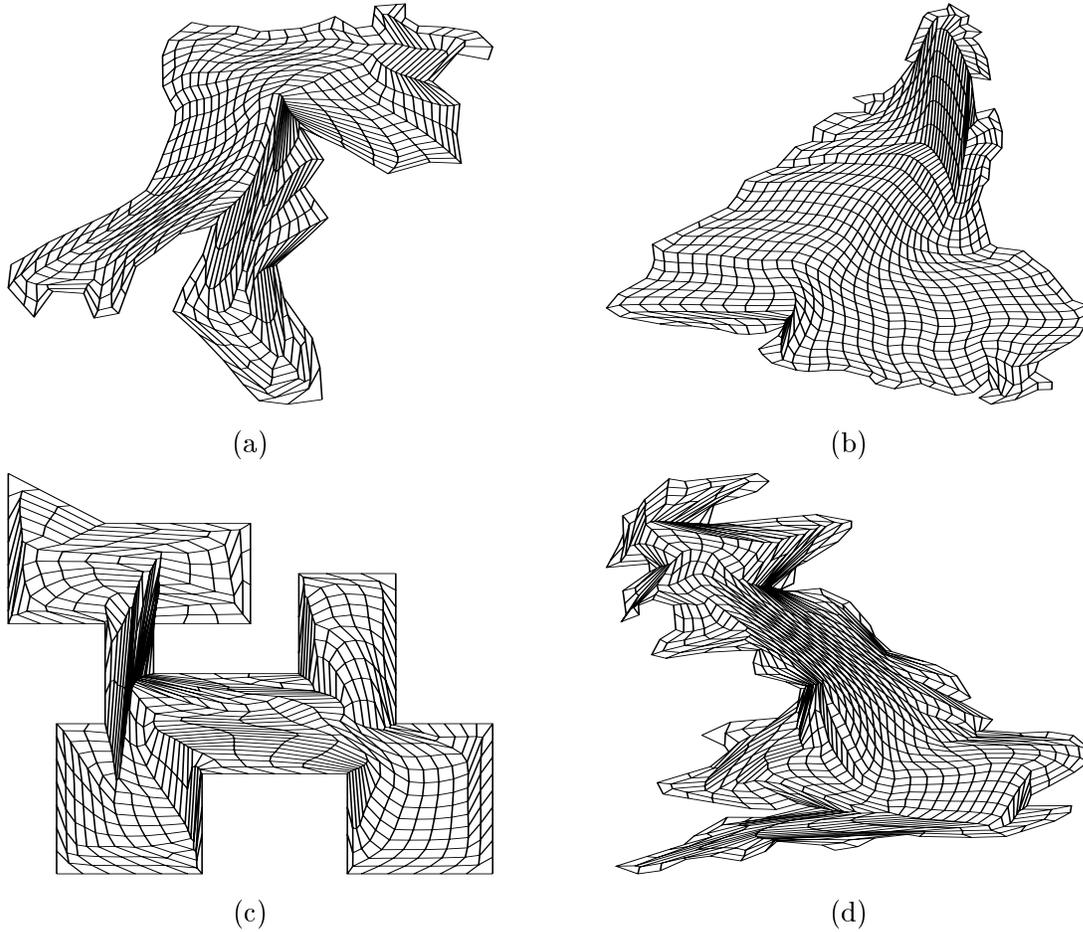


Figura 3.9: Mallas generadas por el funcional A4P. (a) región de la bahía de la Habana, (b) región de Sudamérica, (c) región del Gato y (d) región del Reino Unido. Ninguna es convexa.

3.7.1 Funcional discreto inverso de Área

Otro ejemplo del funcional discreto de Área lo representa el funcional discreto de área inversa

$$F_{A^-} = \sum_{q=1}^N \frac{1}{\alpha_q} \tag{3.71}$$

Éste funcional discreto de Área tiene la particularidad de estar definido sobre mallas con triángulos de Área positiva. Como ya fue comentado, cuando se aplica un método iterativo de optimización cabe la posibilidad de que el optimizador nos lleve a considerar como candidatos al óptimo a mallas no convexas o bien, con celdas degeneradas, en cuyo caso la expresión (3.71) puede no estar definida (alguna $\alpha_q \leq 0$). Si $\alpha_q < 0$ el funcional carece de significado y el optimizador podría conducirnos a puntos o mallas que nada tienen que ver con la propiedad del funcional discreto y continuo de Área.

Por esta razón debe hacerse una regularización del funcional inverso de Área de manera semejante que en con el funcional de suavidad discreto, para permitir al optimizador rechazar puntos (configuraciones de puntos, las mallas) de prueba no aceptables durante su proceso.

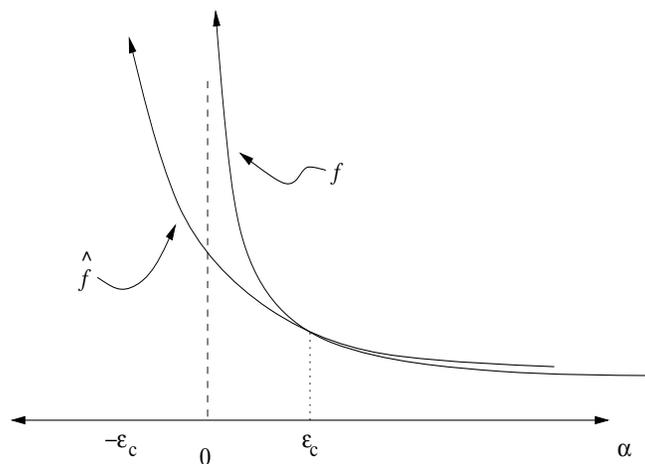


Figura 3.10: Regularización del funcional.

La misma propuesta para regularizar el funcional de Suavidad hecha por Barrera ha dado buenos resultados a éste funcional discreto inverso de Área; por lo que la idea de contar con un conjunto de mallas cuasi-convexas y sobre ése conjunto operar la regularización del funcional continúa resultando atractivo y da la pauta para construir funcionales ampliando el concepto de cuasi-convexidad.

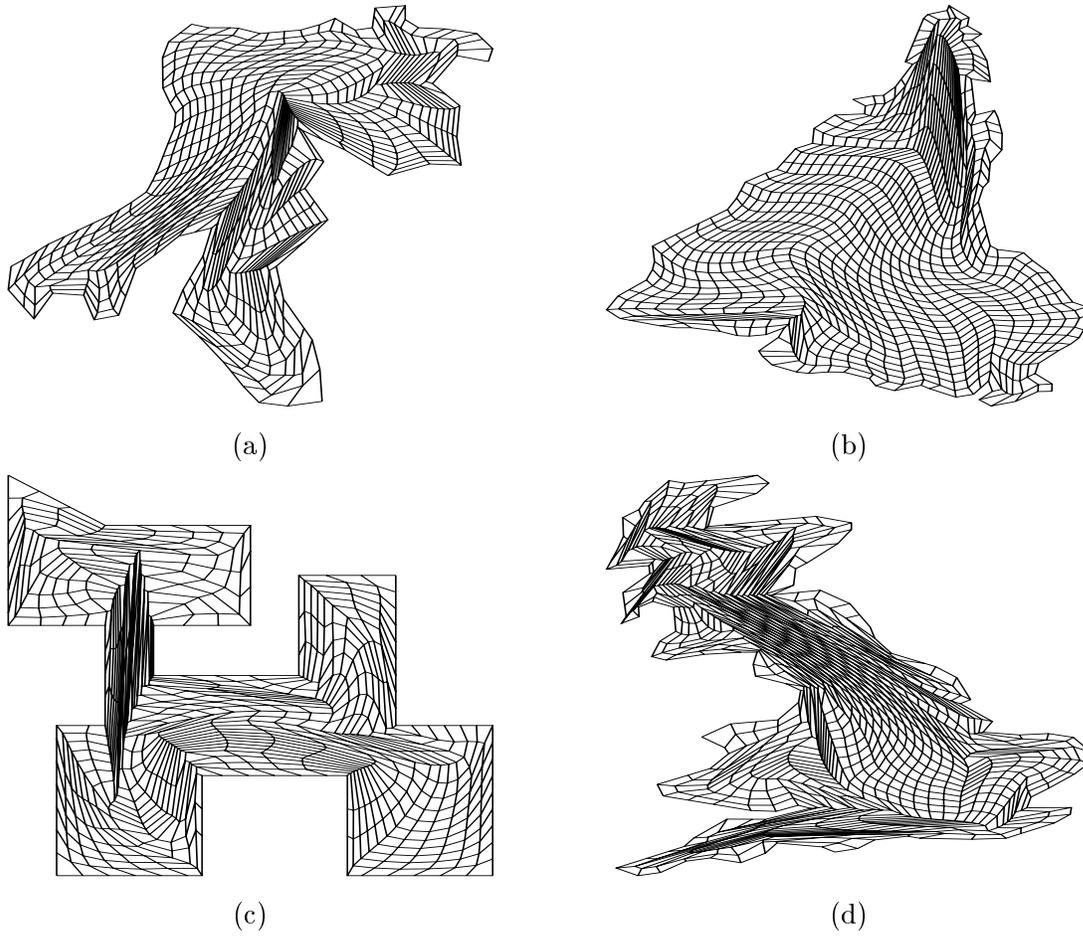


Figura 3.11: Mallas generadas por el funcional de Área inverso regularizado. (a) región de la bahía de la Habana, (b) región de Sudamérica, (c) región del Gato y (d) región del Reino Unido. Todas son convexas.

Capítulo 4

Funcionales para la generación automática de mallas

Como hemos visto, con los funcionales descritos hasta aquí no podemos generar mallas convexas y suaves en regiones atípicas y además, para alguno de ellos es indispensable contar con mallas casi-convexas para iniciar el proceso para esperar obtener una malla con las propiedades deseadas.

En el caso concreto de los funcionales discretos de Suavidad y discreto inverso de Área es necesario contar con mallas cercanas a la convexidad para usar una regularización de manera que el optimizador nos lleve a una solución. Este mecanismo de operación no es práctico cuando se desea construir una malla sobre regiones irregulares de manera que la información necesaria sea lo mínimo indispensable para lograr un malla con las características de suavidad y por supuesto, de convexidad.

Pasaremos a describir algunos de los nuevos funcionales que para la generación automática de mallas se han desarrollado, sus propiedades geométricas así como describir la implantación llevada a cabo en el sistema UNAMALLA.

4.1 Funcional Modificado de Suavidad

La idea central de este funcional surge al ampliar el dominio de operación del funcional discreto de suavidad y mediante una trayectoria Γ obtener una malla convexa con la propiedad del funcional de suavidad. Describamos esta idea en su conjunto.

Una cantidad que nos es útil para medir el “grado de convexidad” de una malla G es el valor mínimo de α sobre los triángulos de la malla

$$\alpha_-(G) = \min_{\Delta \in G} \alpha(\Delta)$$

bajo este concepto trabajar sobre un conjunto de mallas ϵ -convexas,

$$\mathcal{D}_\epsilon = \{G | \alpha_-(G) > -\epsilon\} \quad (4.1)$$

La idea inmediata es expandir el dominio de definición de la funcional de manera que podamos trabajar con mallas ϵ -convexas, véase la Figura 4.1

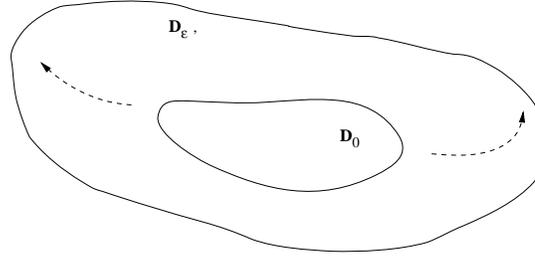


Figura 4.1: Ampliación de dominios.

Sobre cada uno de los triángulos de la malla G observemos el valor de la funcional de Suavidad

$$\bar{f}(\mathbf{a}, \mathbf{b}) - 1 = \frac{\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\mathbf{a}^t J_2 \mathbf{b}}{2\mathbf{a}^t J_2 \mathbf{b}} \quad (4.2)$$

$$= \frac{l(\Delta_q) - 2\alpha(\Delta_q)}{\alpha(\Delta_q)} \quad (4.3)$$

donde $\Delta_q = \Delta(P, Q, R)$ es un triángulo genérico orientado positivamente. Esta expresión como comentamos en el capítulo 2, tiene exactamente las mismas propiedades del funcional de Winslow y nos es muy útil en la implantación dentro del sistema UNAMALLA.

La idea inmediata para ampliar el dominio de acción de este funcional es modificar el cociente de la expresión por una cantidad, digamos k en la forma

$$f_k(\Delta) = \frac{l(\Delta) - 2\alpha(\Delta)}{k + \alpha(\Delta)}, \quad (4.4)$$

la cual tiene sentido sobre el conjunto \mathcal{D}_k en donde la cantidad $k + \alpha$ no se anula para todo triángulo interior de $G \in \mathcal{D}_k$. Bajo esta idea el funcional sugerente, al que llamaremos funcional Modificado de Suavidad, tiene la forma

$$\begin{aligned}
 F_S[k] &= \sum_{q=1}^N f_k(\Delta_q) \\
 &= \sum_{q=1}^N \frac{l(\Delta_q) - 2\alpha(\Delta_q)}{k + \alpha(\Delta_q)}
 \end{aligned}
 \tag{4.5}$$

Veamos cómo operar éste funcional para una malla inicial G_0 ; básicamente cómo construir el conjunto de mallas donde definir el dominio del funcional.

Para poder operar con este funcional tomemos una malla inicial G_0 , para esta encontremos una $k_0 > 0$ y construyamos \mathcal{D}_{k_0} de forma que $G_0 \in \mathcal{D}_{k_0}$, conjunto en el que estará bien definido $F_S[k_0]$ y sobre el cual podremos optimizarlo. En la Figura 4.2 se muestra la idea de construcción de la región \mathcal{D}_{k_0}

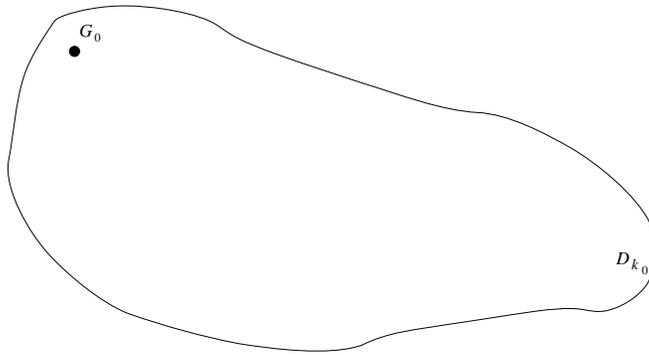


Figura 4.2: Dada una malla inicial construimos un dominio para operar $F_S[k]$.

Una vez construido ese dominio, procedemos a optimizar $F_S[k_0]$

$$G_1 = \arg \min_{G \in \mathcal{D}_{k_0}} F_S[k_0](G)
 \tag{4.6}$$

malla que se encuentra dentro de \mathcal{D}_{k_0} , recalquemoslo. En la Figura 4.3 se muestra gráficamente el óptimo.

Una vez obtenida esta malla óptima, procedemos de manera semejante a encontrar k_1 de manera que G_1 se encuentre dentro del conjunto \mathcal{D}_{k_1} , más aún, cerca de la frontera de éste conjunto, ver Figura 4.4 y sobre ese conjunto optimizar de tal forma que encontremos una malla óptima G_2

$$G_2 = \arg \min_{G \in \mathcal{D}_{k_1}} F_S[k_1](G)
 \tag{4.7}$$

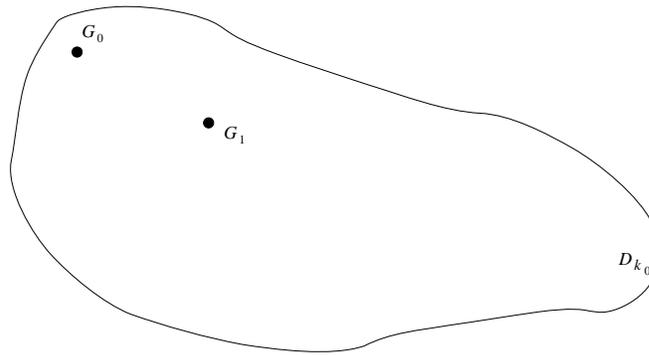


Figura 4.3: El óptimo se encuentra dentro de \mathcal{D}_{k_0} .

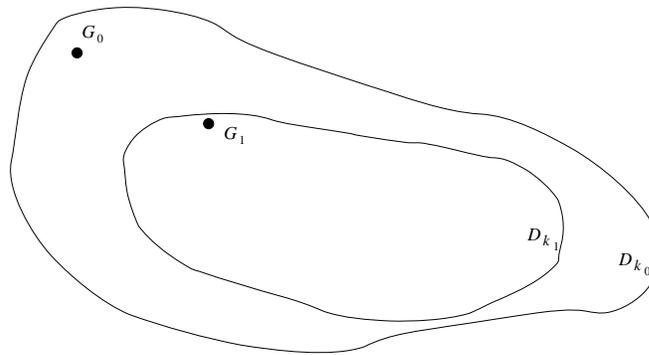


Figura 4.4: Dada la malla G_1 , construimos un dominio donde operar $F_S[k]$.

la que se encontrará dentro de \mathcal{D}_{k_1} , ver Figura 4.5.

Contando con esta malla, procedemos a definir k_2 de tal manera que dentro de \mathcal{D}_{k_2} se encuentre G_2 muy cerca de la frontera, ver Figura 4.6.

Este procedimiento genera una sucesión de números k_0, k_1, \dots, k_r , todos ellos positivos y bajo la idea de que el óptimo G_i se encuentra dentro de \mathcal{D}_{k_i} y cerca de su frontera garantizamos que la sucesión es decreciente

$$k_0 > k_1 > k_2 > \dots > k_r \quad (4.8)$$

de esta forma, si existe una malla convexa bajo este procedimiento nos acercaremos a ella a través de esa sucesión. Lo demostraremos en breve.

Gráficamente el proceso lo describimos así: si G_0 es una malla inicial sobre Ω y G^* una malla convexa y óptima bajo el funcional $F_S[k]$, la trayectoria Γ a seguir para llegar a G^* viene representado graficamente en la Figura 4.7.

Nótese que la forma en que hemos planteado lograr ése camino es debilitar primero la condición de trabajar exclusivamente con mallas convexas o casi-convexas y tra-

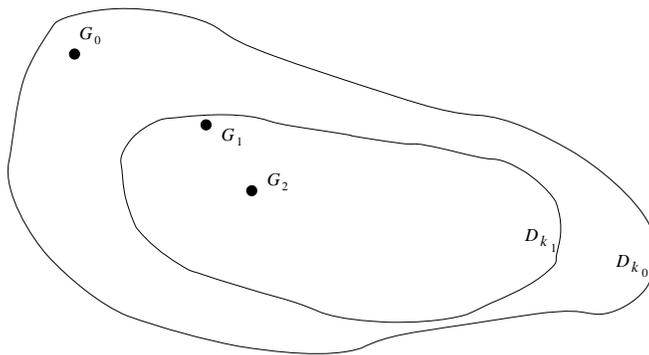


Figura 4.5: El óptimo se encuentra dentro de \mathcal{D}_{k_1} .

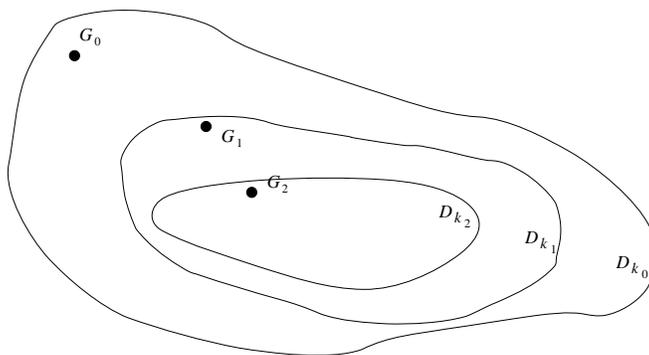


Figura 4.6: Dada la malla G_2 , construimos un dominio donde operar $F_S[k_1]$.

bajar con mallas con $\alpha_- > k$, para k adecuada, y poco a poco ir condicionando hasta obtener mallas convexas. A final de cuentas, este procedimiento tiene sentido siempre y cuando sobre Ω puedan construirse mallas con $\alpha_- > 0$, es decir, mallas convexas. Este es un proceso adaptivo, la definición de k nos conduce a reducir el conjunto de mallas en cada paso de manera que nos acerquemos a un subconjunto de mallas convexas.

Este procedimiento resulta atractivo si podemos de antemano saber que sobre la región existe una malla convexa de dimensión $m \times n$ con los puntos prefijados sobre la frontera. La región de Ivanenko [26] es un claro ejemplo de una región donde no es posible construir mallas convexas de 3×3 pero sí de otra dimensión.

Nuestra propuesta de trabajo es observar esta clase de funcionales $F_S[k]$ de manera que la sucesión en k nos conduzca a una sucesión de conjuntos \mathcal{D}_k cada vez más cercanos al conjunto de mallas convexas \mathcal{D}_0 . Esta idea resulta atractiva, veamos su justificación matemática para después hacer una descripción de uso de éste esquema adaptivo propuesto.

A partir de la definición de $F_S[k]$ en \mathcal{D}_k demostremos que $F_S[k]$ tiene un mínimo

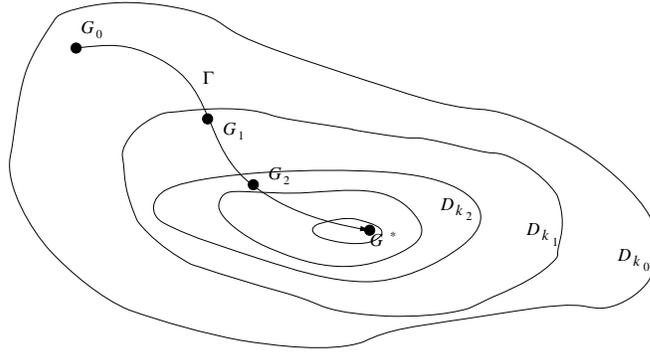


Figura 4.7: Trayectoria.

local en \mathcal{D}_k . Para las siguientes proposiciones, citamos Ivanenko [26], y Tinoco [48].

Proposición 4.1 *Sea k un número real positivo. Sea $\{G_r\}$ una sucesión de mallas en \mathcal{D}_k tal que $\{G_r\} \rightarrow \partial\mathcal{D}_k$, es decir, $\{\alpha_-(G_r)\} \rightarrow -k$. Entonces $\{F_S[k](G_r)\} \rightarrow \infty$.*

Demostración

Supongamos lo contrario, es decir, que $\{F_S[k](G)\}_r$ está acotado en $\partial\mathcal{D}_k$ y bajo la hipótesis tenemos que

$$\{l - 2\alpha_-(G)\} \longrightarrow 0$$

como $F_S[k]$ está acotado, la hipótesis nos lleva a que para algún triángulo Δ_q tenemos que

$$k + \alpha(\Delta_q) \longrightarrow 0$$

es decir que la sucesión $\{(\Delta_q)_r\}$ converge a un rectángulo isósceles o bien a uno con lados nulos. Ya que ninguna de los dos caminos es posible, la proposición queda probada.

l.q.q.d.

Esta proposición nos conduce a la afirmación siguiente:

Corolario 4.2 *Sea k un número real positivo. Si $\mathcal{D}_k \neq \emptyset$, entonces $F_S[k]$ tiene un mínimo local en \mathcal{D}_k*

Hasta aquí, hemos probado que en \mathcal{D}_k para k apropiado, es posible encontrar un mínimo local de $F_S[k]$. Nuestro interés ahora se centra en describir las propiedades de la malla G_k mínimo de $F_S[k]$.

Proposición 4.3 *Sea $G_1 \in \mathcal{D}_k$ para $k > 0$ dado. Sea $G_k \in \mathcal{D}_k$ tal que*

$$F_S[k](G_k) = \min \{F_S[k](G) : G \in \mathcal{D}_k\} \quad (4.9)$$

Entonces, $\alpha_-(G_k) \geq -k'$ con $k' = k - \frac{2k}{\lambda_k+2}$ con $\lambda_k = F_S[k](G_1)$.

Demostración

Sea $G \in \mathcal{D}_k$ con $\alpha_-(G) < -k'$, luego

$$F_S[k](G) \geq \frac{l(\Delta) - 2\alpha_-(G)}{k + \alpha_-(G)}$$

donde Δ es el triángulo de área mínima de G , así

$$\begin{aligned} F_S[k](G) &= \frac{l(\Delta) + 2k}{k + \alpha_-(G)} - 2 \\ &\geq \frac{2k}{k + \alpha_-(G)} - 2 \\ &> \frac{2k}{\frac{2k}{\lambda_k+2}} - 2 \\ &= \lambda_k \\ &= F_S[k](G_1), \end{aligned} \quad (4.10)$$

entonces $F_S[k]$ no tiene mínimo en \mathcal{D}_k . Con lo que hemos probado la proposición.

l.q.q.d.

La última proposición nos lleva a una relación entre el mínimo de $F_S[k]$ en \mathcal{D}_k y una malla en \mathcal{D} . Veamos ahora cómo poder calcular k de manera que $F_S[k]$ tenga por mínimo a una malla más cercana a la convexidad, esto ayudará acelerar el proceso. Precisemos esto último en la siguiente proposición.

Proposición 4.4 *Supongamos que existen mallas convexas \bar{G} y G_0 en \mathcal{D}_0 tales que $\beta = \alpha_-(G_0) < 0$. Sea $k = -\beta + \epsilon$ con $0 < \epsilon < -\beta/\lambda$ y con $\lambda = F_S[0](\bar{G})$. Sea G_k una malla en \mathcal{D}_k en donde se alcanza un mínimo de $F_S[k]$. Entonces $\alpha_-(G_k) > \beta$.*

Demostración

De la proposición anterior se desprende que para la malla $G \in \mathcal{D}_k$ con $\alpha_-(G) < -k + \frac{2k}{\lambda_k + 2}$ donde $\lambda_k = F_S[k](\bar{G})$ se tiene que $F_S[k](G) > \lambda_k$, entonces

$$\begin{aligned}
 \alpha_- &\geq -k + \frac{2k}{\lambda_k + 2} \\
 &= \beta - \epsilon + \frac{2(-\beta + \epsilon)}{\lambda_k + 2} \\
 &= \frac{\lambda_k}{\lambda_k + 2}(\beta - \epsilon) \\
 &> \frac{\lambda_k}{\lambda_k + 2}\left(\beta + \frac{\beta}{\lambda}\right) \\
 &= \frac{\lambda_k}{\lambda} \left(\frac{\lambda + 2}{\lambda_k + 2}\right) \left(\frac{\lambda + 1}{\lambda + 2}\right) \beta \tag{4.11}
 \end{aligned}$$

por una parte, $\lambda_k \leq \lambda$ entonces

$$\frac{\lambda_k}{\lambda_k + 2} \frac{\lambda + 2}{\lambda} \leq 1$$

y siendo $\beta < 0$ tenemos que

$$\frac{\lambda_k}{\lambda} \frac{\lambda + 2}{\lambda_k + 2} \beta \geq \beta$$

entonces de (4.11) se concluye que

$$\alpha_-(G_k) > \frac{\lambda + 1}{\lambda + 2} \beta > \beta.$$

l. q. q. d.

Hasta aquí, hemos podido lograr un óptimo (mínimo local) en \mathcal{D}_k con la propiedad de que sea “más convexa” que la malla inicial. Sin embargo, la elección de k depende del valor del funcional en dicha malla óptima, y ésta bien pudiera no existir sobre Ω . Pero ésta proposición nos da una pauta del dominio de valores en los que se espera encontrar k , básicamente cercano a $-\alpha_-(G_k)$ y mayor que éste; si $\epsilon > 0$, sería de la forma

$$k_s = -\alpha_-(G_s) + \epsilon \tag{4.12}$$

Pasemos ahora a describir un algoritmo adaptivo en k para el funcional de Suavidad.

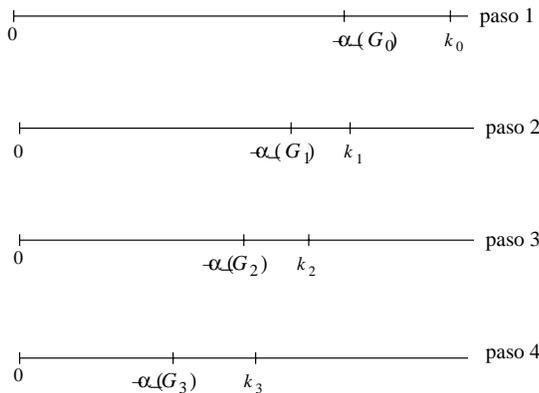


Figura 4.8: En cada paso k debe estar a la derecha de $-\alpha_-(G)$

4.2 Funcional Adaptivo de Suavidad

La última proposición nos conduce a desarrollar un mecanismo para lograr una trayectoria en k y obtener una malla con las características geométricas deseadas en $F_S[k]$. Por ejemplo, de iniciar con una malla G_1 , calcular k_1 para esta malla en la forma (4.12) y optimizar para lograr una malla G_2 , calcular k_2 para esta malla y repetir el proceso; ese sería nuestro mecanismo.

Algoritmo 4.1

1. Sea G_1 una malla sobre Ω .
2. Calcúlese

$$k = -\alpha(G_1) + \epsilon$$

con $\epsilon > 0$ de manera que k satisfaga la Proposición 4.4.

3. Calcular G_k , mínimo local de $F_S[k]$,

$$G_k = \arg \min_{G \in \mathcal{D}_k} F_S[k](G)$$

4. Para G_k , repetir los pasos 2 y 3 anteriores hasta convexidad.

Este mecanismo en principio nos llevaría a una malla convexa, siempre que exista, o cercana a ella y con la propiedad de los funcionales $F_S[k]$, la propiedad del funcional discreto de suavidad. La siguiente proposición confirma esta idea.

Proposición 4.5 *Si el conjunto \mathcal{D}_0 de mallas convexas de dimensión $m \times n$ sobre Ω , es no vacío, entonces el Algoritmo 4.1 logra obtener una malla convexa en un número finito de actualizaciones de la k .*

Demostración

Supongamos que el proceso no se logra en un número finito de pasos. Observemos la sucesión de $\{k_q\}$ y $\{G_q\}$ obtenidas del Algoritmo 4.1. De acuerdo con la proposición anterior, tenemos que

$$0 > \alpha_-(G_q) > \left(\frac{\lambda + 1}{\lambda + 2}\right)^q \alpha_-(G_0) \quad (4.13)$$

obteniendo

$$\lim_{q \rightarrow \infty} \alpha_-(G_q) = 0 \quad (4.14)$$

por otra parte, siendo $k_q > -\alpha_-(G_q)(1 + 1/\lambda)$, se tiene

$$\lim_{q \rightarrow \infty} k_q = 0. \quad (4.15)$$

Ahora bien, si

$$D^* = \lim_{q \rightarrow \infty} D_{k_q} \quad (4.16)$$

entonces las mallas en este conjunto deben ser tales que ninguna de sus celdas son estrictamente convexas; es decir, el área de todos los triángulos de las mallas en D^* es no negativas.

El proceso genera mallas G_{k_q} con la propiedad

$$F_S[k_q](G_{k_q}) \leq F_S[k_q](G) \leq F_S[0](G) \quad (4.17)$$

donde G es una malla convexa.

Ahora bien, si H es una malla en D^* con $\alpha_-(H) = 0$ y Δ un triángulo de H de área mínima, esto es $\alpha(\Delta) = 0$. Entonces

$$F_S[k_q](H) \geq \frac{l(\Delta) - 2\alpha(\Delta)}{k_q + \alpha(\Delta)} = \frac{l(\Delta)}{k_q} \quad (4.18)$$

Si H se obtiene como límite de tal proceso

$$\lim_{q \rightarrow \infty} F_S[k_q](H) \leq F_S[0](G) \quad (4.19)$$

esto es,

$$\lim_{k_q \rightarrow 0} \frac{l(\Delta)}{k_q} \quad (4.20)$$

es finito. Pero entonces $l(\Delta)$ debe ser cero y por lo tanto los lados de Δ deben ser de longitud nula. Esto a su vez lleva a que los triángulos de H que tengan un lado en común con Δ deben ser de área nula.

El razonamiento reiterativo conducirá a que todos los lados de las celdas de H sean nulos, incluso de la frontera de la región, lo cual es imposible. Así queda demostrado el teorema. El Algoritmo 4.1 nos conduce a una malla convexa en un número finito de pasos si ésta existe.

l.q.q.d.

El procedimiento descrito, el Algoritmo 4.1, representa un esquema adaptivo en k , esto es, k varía hasta 0, de manera que éste procedimiento nos conduce a una malla cada vez más cercana a la convexidad. A este esquema o procedimiento y al conjunto de funcionales descritos para cada k lo llamaremos *Funcional Adaptivo Discreto de Suavidad*. En realidad, el algoritmo 4.1 no necesariamente debe detenerse cuando se logra una malla convexa, podemos seguir operando hasta lograr un valor $k^* > 0$ y entonces obtener una malla que además de ser convexa tenga la propiedad de que el área de los triángulos sea mayor que k^* .

Una vez descrita la potencialidad de este procedimiento adaptivo, ampliaremos este esquema a otro tipo de funcional.

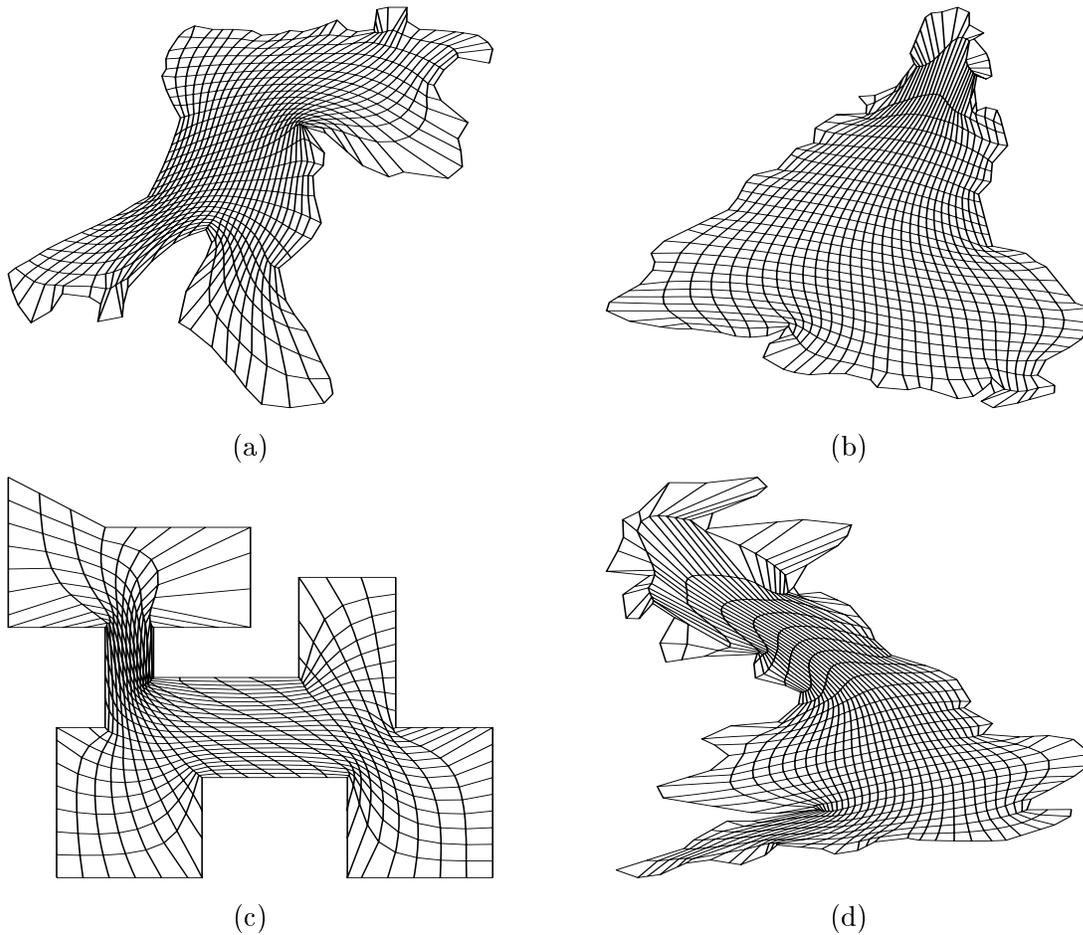


Figura 4.9: Mallas convexas generadas por el funcional de k -Suavidad. (a) región de la bahía de la Habana, (b) región de Sudamérica, (c) región del Gato y (d) región del Reino Unido.

4.3 Funcional Inverso modificado de Área

Retomemos el funcional discreto inverso de Área descrito en el capítulo anterior,

$$F_{A^-}(G) = \sum_{q=1}^N \frac{1}{\alpha_q} \quad (4.21)$$

tal como se mencionó para operar este tipo de funcionales debemos tener en cuenta mallas que estén cercanas a la convexidad y bajo una regularización poder operar con el funcional. Procedamos como en el apartado anterior y observemos dominios extendidos donde trabajar este funcional.

Esta idea nos lleva a trabajar con un conjunto de mallas cercanas a la convexidad. Una forma de lograrlo es considerar una medida de “admisibilidad” o de convexidad para su operación. Esta medida bien puede ser la siguiente:

$$\alpha_-(G) = \min_{\alpha \in G} \alpha$$

bajo este concepto trabajar sobre un conjunto de mallas ϵ -convexas,

$$\mathcal{D}_\epsilon = \{G | \alpha_-(G) > -\epsilon\} \tag{4.22}$$

con $\epsilon > 0$ pequeña para lograr una malla adecuada, con las propiedades del funcional o cercana a ellas. Esto es lo mínimo indispensable para lograrlo. Sin embargo, recordemos que el funcional inverso de Área trabaja sobre mallas convexas.

En la práctica, en no pocas ocasiones, contamos con mallas cuya admisibilidad o medida de convexidad es muy grande, pero para que esta idea funcione debemos trabajar con mallas ϵ -cercanas a la convexidad. Una manera de tratar este problema es extender el radio de acción de dicho funcional de manera que podamos abarcar a ese dominio de mallas no convexas y entonces operar con el funcional o bien uno con las mismas propiedades.

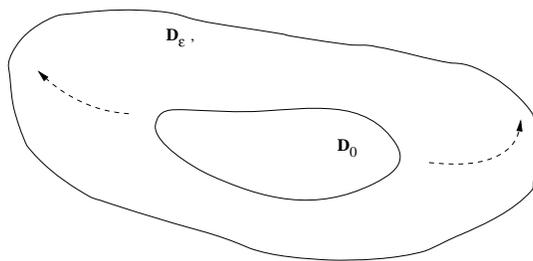


Figura 4.10: Expansión del dominio \mathcal{D}_ϵ a $\mathcal{D}_{\epsilon'}$

Para lograrlo se propone la expresión

$$f(\alpha) = \frac{1}{\alpha + k} \tag{4.23}$$

la cual tiene sentido sobre el conjunto de mallas \mathcal{D}_k con

$$\mathcal{D}_k = \{G | \alpha_-(G) > -k\}$$

válido para $k > 0$ dada. Tiene sentido porque el funcional es convexo y es del tipo de los funcionales de Área descritos en el capítulo anterior.

El funcional

$$F_A[k](G) = \sum_{q=1}^N \frac{1}{k + \alpha_q} \quad (4.24)$$

es un funcional discreto de Área, tal y como se mostró en el Capítulo 3, actúa sobre el conjunto de mallas \mathcal{D}_k de dimensión $m \times n$ que sobre Ω pueden ser construidos. A éste funcional le llamaremos funcional *discreto inverso modificado de Área*.

Para poder operar con este funcional partamos de una malla inicial G_{k_0} para la cual debmos encontrar una $k > 0$ y construir \mathcal{D}_k , con $G_{k_0} \in \mathcal{D}_k$ para que $F_A[k]$ este bien definido y poder optimizarlo.

Por supuesto mientras más grande sea k mayor flexibilidad tendremos en operar (4.24) pero de igual manera estaremos tomando como admisibles (en operación) mallas con medida alta de no convexidad. Veamos algunas propiedades del funcional modificado.

Proposición 4.6 *Sea $k > 0$, si $\mathcal{D}_k \neq \emptyset$, entonces $F_A[k]$ tiene al menos un mínimo local en \mathcal{D}_k .*

Proposición 4.7 *Supóngase que existen G_1 y $G_2 \in \mathcal{D}_k$ con*

$$G_2 = \arg \min_{G \in \mathcal{D}_k} F_A[k](G)$$

entonces $\alpha_-(G_2) \geq -k'$ con $-k' = -k - \frac{1}{\lambda_k}$ y $\lambda_k = F_A[k](G_0)$.

Demostración

Supongamos que $\alpha(G_1) < -k'$. Sea Δ un triángulo de G_1 de área mínima ($\alpha(\Delta) = \alpha_-(G_1)$), entonces

$$\begin{aligned} F_A[k](G_1) &> \frac{1}{k + \alpha(\Delta)} \\ &= \frac{1}{k + \alpha_-(G_1)} \\ &> \frac{1}{k - k'} \\ &= \lambda_k = F_k(G_0) \end{aligned}$$

de ser cierto, no se cumple el hecho de que $F_A[k](G_1)$ es el mínimo.

l.q.q.d.

Proposición 4.8 Sean $\bar{G} \in \mathcal{D}_{k_0}$ y G_1 tal que $\beta = \alpha_-(G_1) < k_0$. Sea $k = -\beta + \epsilon$ con $0 < \epsilon < \frac{1}{2\lambda}$ y $\lambda = F_A[k_0](\bar{G})$. Sea $G_2 \in \mathcal{D}_k$ tal que

$$G_2 = \arg \min_{G \in \mathcal{D}_k} F_A[k](G)$$

entonces $\alpha_-(G_2) > \beta$.

Demostración

De la proposición anterior se desprende que para las mallas $G \in \mathcal{D}_k$ con $\alpha_-(G) < -k + 1/\lambda_k$ y $\lambda_k = F_A[k](G)$, entonces se cumple $F_A[k](G) > \lambda_k$. Por lo tanto

$$\begin{aligned} \alpha_-(G_1) &\geq -k + \frac{1}{\lambda} \\ &> \beta - \epsilon + \frac{1}{\lambda_k} \\ &> \beta - \epsilon + \frac{1}{\lambda} \end{aligned}$$

dado que $\lambda = F_A[k_0](\bar{G}) > F_A[k](\bar{G}) = \lambda_k$, obtendremos que

$$\begin{aligned} \alpha_-(G_1) &> \beta - \frac{1}{2\lambda} + \frac{1}{\lambda} \\ &= \beta + \frac{1}{2\lambda} \\ &> \beta \end{aligned}$$

l.q.q.d.

La proposición última va más allá y nos permite comparar en forma distinta los funcionales y los valores de $\alpha_-(G_k)$ que se van obteniendo y así poder afirmar que con una adecuada actualización de k disminuirá α_- .

De igual manera se sigue que un mínimo de $F_A[k]$ en \mathcal{D}_k será una malla “más cercana a la convexidad” que alguna otra. Ahora bien, si

$$G_k^* = \arg \min_{G \in \mathcal{D}_k} F_A[k](G) \tag{4.25}$$

entonces $G_k^* \in \mathcal{D}_k$ y $\alpha_-(G_k^*) > \alpha_-(G_k)$, además que de existir una malla convexa, con un esquema adaptivo de actualización en k , se obtiene una malla convexa con la propiedad de uniformidad en área.

Si controláramos k de manera que se garantice la propiedad de que el mínimo de la funcional vaya disminuyendo poco a poco, nos aproximaríamos a mallas convexas con la propiedad geométrica del funcional, siempre que exista una malla convexa en la región. Esta última proposición así nos lo hace saber al poder comparar dos mallas entre dos funcionales.

Describamos ampliamente este procedimiento de “homotopía” o adaptividad.

4.4 Homotopía o trayectoria de adaptividad

El procedimiento que a *grosso modo* se ha planteado es, que si existe una malla convexa en Ω podemos construir una sucesión de mallas óptimas en \mathcal{D}_k de manera que la sucesión en k nos conduzca a obtener una malla convexa y óptima en el sentido de la funcional inverso de Área.

Esta idea enfocada como un procedimiento iterativo: “en cada paso vamos obteniendo una malla cada vez cercana a la convexidad” describe con ello una “trayectoria” que parte de \mathcal{D}_k hasta lograr una malla convexa en \mathcal{D}_0 bajo el supuesto que éste conjunto no sea vacío (que exista alguna malla convexa en Ω). La región de Ivanenko [26] es un claro ejemplo donde no es posible construir una malla convexa de 3×3 .

En términos formales una trayectoria Γ sobre \mathcal{D} que nos conduzca de \mathcal{D}_k hacia el conjunto de mallas convexas \mathcal{D}_0 la describimos como

$$\Gamma[-k_0, 0] \longrightarrow \mathcal{D} \quad (4.26)$$

donde

$$\Gamma(-k_0) = G_{k_0} \quad (4.27)$$

$$\Gamma(0) = G^* \in \mathcal{D}_0 \quad (4.28)$$

Revisemos esta idea de la trayectoria, cómo construirla y desde luego, su justificación matemática.

Sea \mathcal{D} el conjunto de mallas de dimensión $m \times n$ que sobre Ω pueden ser construidas. Sea $G \in \mathcal{D}$ una malla sobre Ω , sea

$$\alpha_-(G) = \min_{\Delta \in G} \alpha(\Delta) \quad (4.29)$$

sobre los triángulos Δ interiores en G , definamos ahora el conjunto

$$\mathcal{D}_k = \{G : G \in \mathcal{D} \text{ con } \alpha_-(G) > -k\} \quad (4.30)$$

con $k \in \mathbb{R}$. Veamos algunas propiedades de este conjunto.

Por una parte, k está acotado inferiormente por

$$-\bar{\alpha} = -\frac{\text{área}(\Omega)}{(m-1)(n-1)}, \quad (4.31)$$

ya que para $k \leq -\bar{\alpha}$, el conjunto $\mathcal{D}_k = \emptyset$, esto es, no existe malla G sobre Ω con la propiedad que $\alpha_-(G) > \bar{\alpha}$. Bajo esta observación consideremos la cota inferior de k

$$k_- = \inf \{k : \mathcal{D}_k \neq \emptyset\}, \quad (4.32)$$

si éste valor, la cota inferior de k , es no positivo y mayor que $-\bar{\alpha}$, entonces el conjunto de mallas convexas sobre Ω es no vacío.

Otra propiedad del conjunto \mathcal{D}_k es la inclusión de conjuntos: si k_1 y k_2 son números tales que $k_1 < k_2$, entonces

$$\mathcal{D}_{k_1} \subset \mathcal{D}_{k_2}; \quad (4.33)$$

así pues, la trayectoria Γ deseada la podemos representar en la búsqueda de mallas G_k^* , óptimo de $F_A[k]$ dentro de las regiones \mathcal{D}_k hasta lograr una malla en \mathcal{D}_0 con la propiedad del funcional deseado. Desde luego estaremos interesados en que la trayectoria fuese continuamente diferenciable. Esto último no será objeto de estudio en este trabajo.

Veamos de qué forma aplicar esta idea al funcional de Área.

4.5 Funcional Adaptivo de Área

Los anteriores resultados nos garantizan que de existir una malla convexa con las propiedades del funcional inverso de Área se puede obtener bajo un esquema adaptivo de actualización de k . Observemos el algoritmo siguiente:

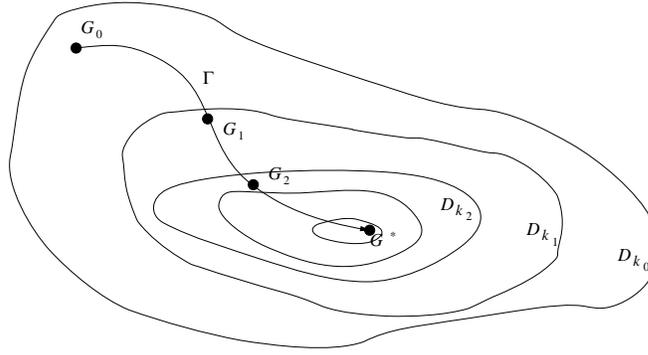


Figura 4.11: Trayectoria hacia \mathcal{D}_0

Algoritmo 4.2

1. Obtener una malla inicial G_1 sobre Ω .
2. Calcular k para G_1 con $\alpha_-(G_1)$ con $\alpha_-(G_1) > -k$.
3. Optimizar $F_A[k]$

$$G_k = \arg \min_{G \in \mathcal{D}_k} F_A[k](G)$$

4. Repetir los pasos 2–3 hasta obtener una malla convexa con las propiedades de $F_A[k]$ o una cercana a la convexidad.

La siguiente proposición da validez al Algoritmo 4.2 en la obtención de mallas convexas, siempre que estas existan.

Proposición 4.9 Si $\mathcal{D}_0 \neq \emptyset$, el Algoritmo 4.2 nos permite obtener una malla $G_k \in \mathcal{D}_0$ en un número finito de pasos.

Demostración

Sea $\{G_q\}$ una sucesión de mallas óptimas generadas con el proceso anterior donde G_0 es la malla inicial, observemos que

$$\alpha_-(G_1) > \alpha_-(G_0) + \frac{1}{2\lambda} \quad (4.34)$$

y entonces tendremos que la sucesión

$$\begin{aligned}
 \alpha_-(G_1) &> \alpha_-(G_0) + \frac{1}{2\lambda} \\
 \alpha_-(G_2) &> \alpha_-(G_1) + \frac{1}{2\lambda} \\
 &> \alpha_-(G_0) + 2\frac{1}{2\lambda} \\
 \alpha_-(G_3) &> \alpha_-(G_2) + \frac{1}{2\lambda} \\
 &> \alpha_-(G_0) + 3\frac{1}{2\lambda} \\
 &\vdots \\
 \alpha_-(G_q) &> \alpha_-(G_0) + q\frac{1}{2\lambda}
 \end{aligned}$$

expresión de la que se concluye que para alguna q , $\alpha_-(G_q) > 0$.

l.q.q.d.

Profundicemos en el problema de la actualización de k . En la proposición 4.8 se observa que la actualización de k no tiene por qué estar tan cerca de $\alpha_-(G)$ ya que $f_k = \frac{1}{k+\alpha}$ evitará llevarnos a triángulos con área cercana a k , mientras esto sea posible. Así, para evitar problemas con el optimizador en la búsqueda de Δ_q , k debe ser de tal forma que permita lograr una malla con $\alpha_-(G_k^*) > -k + \epsilon$. Aquí se aprecia con mejor claridad nuestra propuesta de no optimizar $F_A[k]$ completamente, tan sólo lograr unos pasos del optimizador y luego proceder a actualizar el valor de k .

Otra propiedad que se aprecia de este “funcional adaptivo” es poder lograr, en caso de que exista (recalquemos este punto una y otra vez) una malla con la característica de que el área de los triángulos de las celdas esté acotado inferiormente por $k^*/2$; y es bajo ésta propiedad, la de poder lograr esta medida de calidad por la que hemos llamado a estos funcionales adaptivos.

El procedimiento descrito, el Algoritmo 4.2, representa un esquema adaptivo en k ; esto es k varía hasta 0 de manera que las mallas generadas por este procedimiento nos lleva a una malla cada vez más cerca de la convexidad. A este esquema o procedimiento y al conjunto de funcionales descritos para cada k lo llamaremos *Funcional Adaptivo Discreto de Área*. En realidad, el Algoritmo 4.2 no necesariamente debe detenerse cuando se logra una malla convexa, podemos seguir operando hasta lograr un valor $k^* > 0$ y entonces obtener una malla que además de ser convexa tenga la propiedad de que el área de los triángulos sea superior a k^* , una medida de calidad de la malla.

Como se observa el procedimiento 4.2 puede producir valores de k negativos y este pudiera no decrecer o hacerlo lentamente. En el sistema UNAMALLA se ha implementado

una homotopía de la trayectoria de k de manera que pueda agilizarze y protegerse el algoritmo, para lograrlo planteamos la actualización de k en la forma

$$\begin{aligned}\tilde{k} &= -1.0015 * \alpha_- + 0.01 * \bar{\alpha} \\ k &= \max\{k^* \bar{\alpha}, \tilde{k}\}\end{aligned}\tag{4.35}$$

esta elección de k nos ha dado buenos resultados y de existir una malla convexa en Ω con la propiedad de que

$$\alpha_-(G) > k^* \quad \text{con} \quad k^* = 0.01\bar{\alpha}\tag{4.36}$$

el Algoritmo 4.2 con la actualización (4.35) nos conduce a una malla convexa con la propiedad del funcional de discreto de área.

La constante k^* involucrada en la actualización (4.35) es un número positivo experimental y es una cota de la actualización a efectuar en el Algoritmo 4.2 para de esa forma detenerlo y entonces lograr una malla con la propiedad de que $\alpha_- > k^*$; nuestros experimentos avalan estos argumentos.

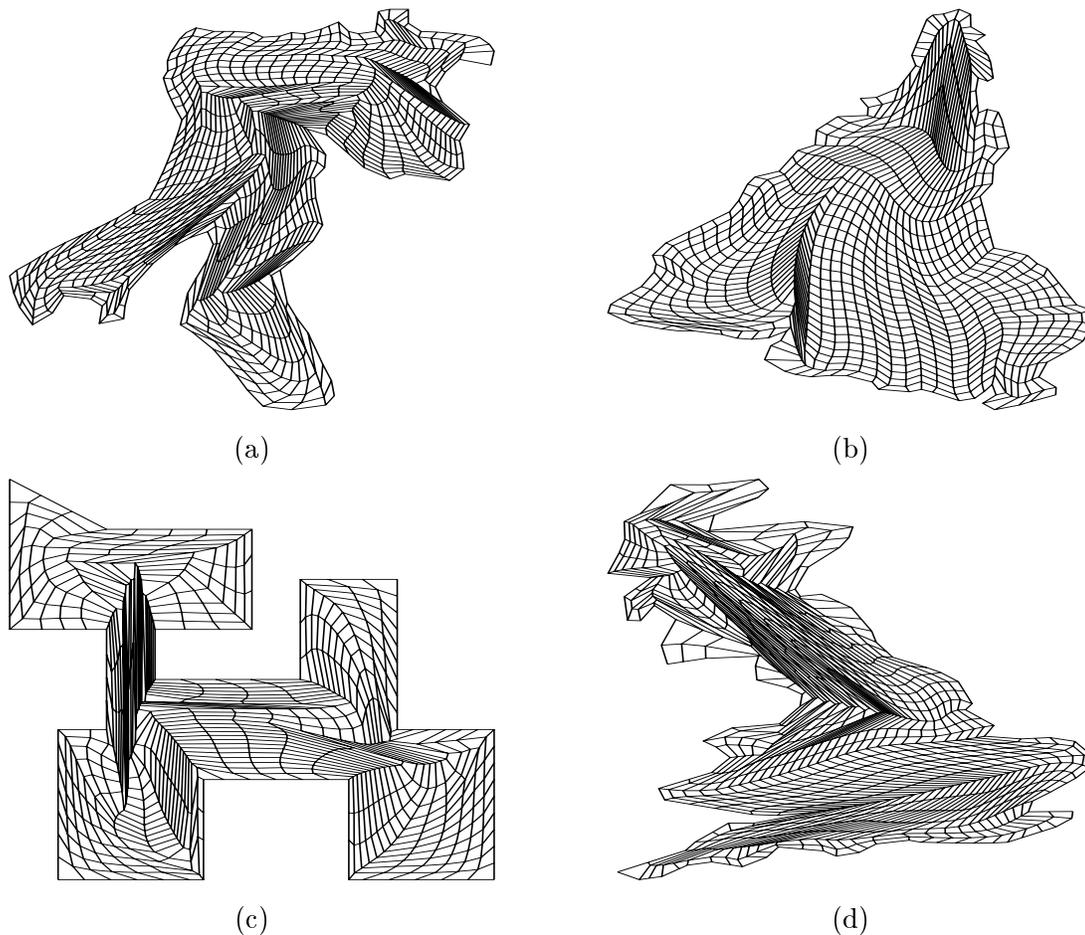


Figura 4.12: Mallas convexas generadas por el funcional de k -Área. (a) región de la bahía de la Habana, (b) región de Sudamérica, (c) región del Gato y (d) región del Reino Unido.

4.6 Una descripción práctica de los funcionales Adaptivos

La automatización es un proceso que siempre ha estado presente en la descripción y desarrollo de nuestros funcionales con la finalidad de que con poca información podamos obtener sobre la región de estudio una malla convexa con ciertas propiedades geométricas en un número finito de pasos.

Los métodos anteriormente descritos proponen una forma de llevar a cabo la adaptividad que nos conduzca a una malla convexa siguiendo k pero no cómo llevarlo a cabo. Nuestra experiencia nos ha hecho proponer un mecanismo práctico para lograr

este proceso de manera aceptable. Describiremos dicho proceso así como algunas consideraciones y observaciones de la implantación de los funcionales.

4.6.1 Sobre la actualización de k

Los algoritmos 4.1 y 4.2 nos dicen que debe llevarse a cabo la actualización de k pero no cómo hacerlo. La observación de que el valor de k esté cercano a α_- nos conduce a proponer algunos valores.

Los mismos algoritmos nos piden optimizar cada paso para k previamente elegida y luego actualizar k y entonces repetir el proceso. Cada ciclo de este algoritmo representa un paso para lograr la trayectoria. Sin embargo no es necesario lograr el óptimo en cada ciclo del algoritmo ya que tan sólo requerimos seguir la trayectoria lo más cerca posible y tal vez al final entonces sí dedicarle todo el esfuerzo al optimizar.

En el sistema UNAMALLA se tiene implementado la actualización (4.35) y la optimización de F_k , el paso 3 del Algoritmo 4.1, no se realiza completamente, es decir, no se llega el mínimo local de F_k sólo se realizan algunas iteraciones 5, 10 dependiendo del método de optimización empleado. Esta idea, la de no llevar a cabo por completo la optimización de F_k , la justificamos bajo el supuesto que la malla G_k que resulta podría estar al inicio alejada de la solución, por ejemplo la malla inicial obtenida por el método algebraico TFI en mallas irregulares tiene muchas celdas dobladas y las líneas no son suaves por lo que es inútil obtener una malla óptima en los primeros pasos dentro del proceso adaptivo. Se ha visto que esta idea la de emplear solamente algunas iteraciones es útil en la práctica ya que en pocos pasos del algoritmo la k va logrando acercarse a k^* nos vamos acercando cada vez más a \mathcal{D}_k y por consiguiente nos aproximamos a las propiedades de F_k .

Para la región de la bahía de la Habana podemos observar en la Figura 4.13 la evolución de k a lo largo de las iteraciones y en la Figura 4.14 la evolución de α_- a lo largo de las iteraciones.

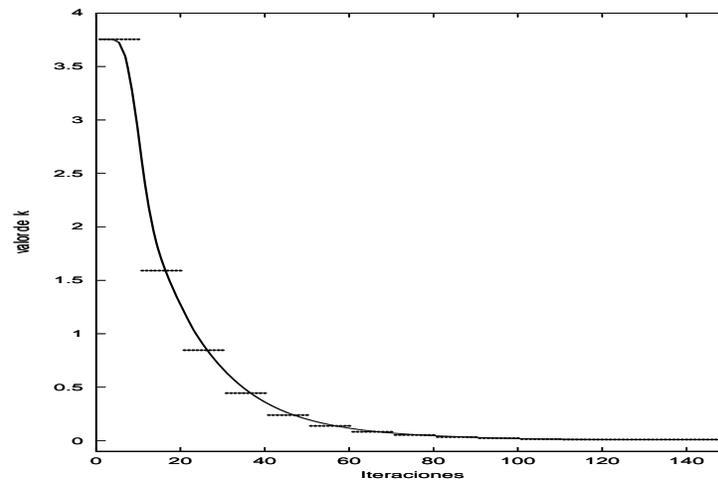


Figura 4.13: Cambio de k a través de las iteraciones del Algoritmo 4.1 y sobre la región de la bahía de la habana.

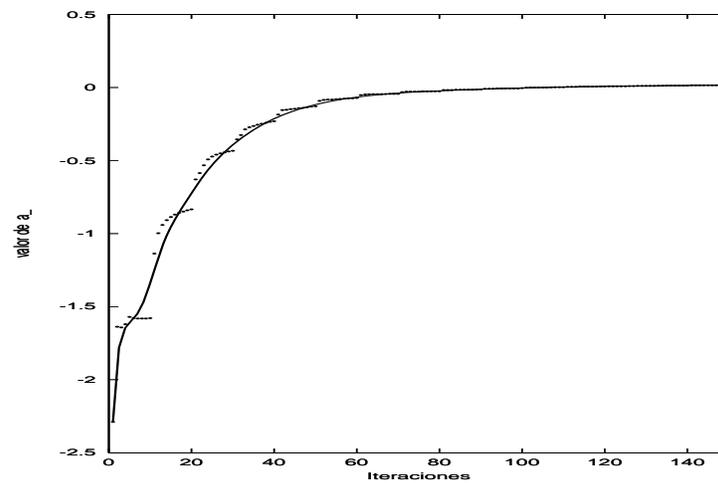


Figura 4.14: Cambio de $\alpha_$ a través de las iteraciones del Algoritmo 4.1 y sobre la región de la bahía de la habana.

De igual forma podemos observar el comportamiento de k y de α_- a lo largo de las iteraciones por el Algoritmo 4.2, el funcional adaptivo de Área en las Figuras 4.15 y 4.16.

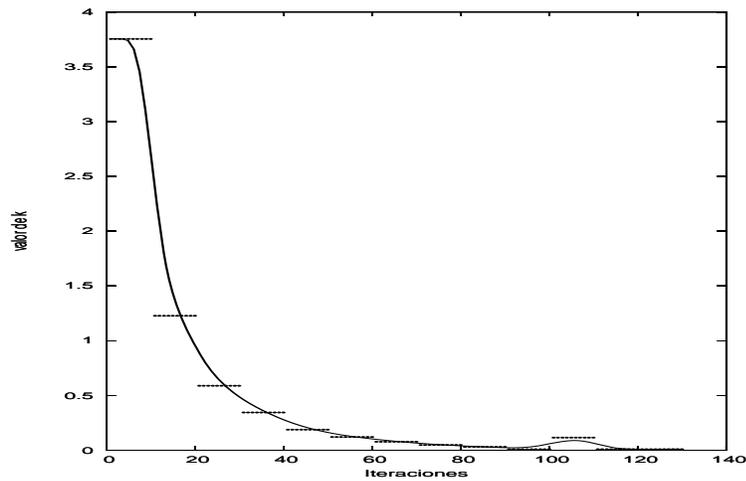


Figura 4.15: Cambio de k a través de las iteraciones del Algoritmo 4.2 y sobre la región de la bahía de la habana.

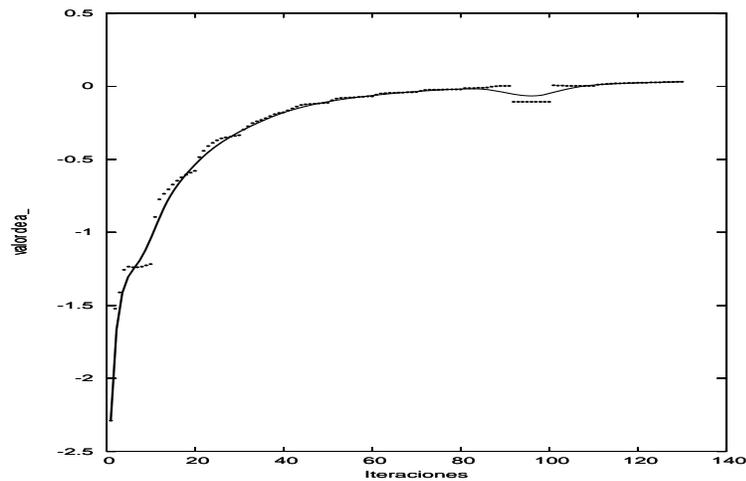


Figura 4.16: Cambio de α_- a través de las iteraciones del Algoritmo 4.2 y sobre la región de la bahía de la habana.

Dentro del sistema UNAMALLA, hemos desarrollado el proceso en dos partes. La primera es propiamente la Adaptividad hasta lograr una malla convexa y la siguiente es teniendo la malla convexa de la primera, refinar nuestro proceso de optimización.

Optimización de funcionales adaptivos

1. Paso de adaptividad hasta lograr una malla convexa
2. Paso de refinamiento de la optimización

El primer paso consiste en seguir el Algoritmo 4.1 o 4.2 hasta lograr una malla convexa y hasta que $k = k^*$ el segundo es ya un refinamiento de la optimización con k^* fija. Para el primer paso describiremos de manera puntal su implantación.

Paso de adaptividad

1. Contando con una malla inicial G_0 calcular k

$$\begin{aligned}\tilde{k} &= -1.0015 \cdot \alpha_- + 0.01 \cdot \tilde{\alpha} \\ k &= \max\{k^* \cdot \tilde{\alpha}, \tilde{k}\}\end{aligned}$$

2. Si $k = k^*$ y la malla es convexa, salir, de lo contrario continuar.
3. Obtener G

$$G = \arg \min F[k]$$

luego de m pasos con un optimizador de gran escala o bien, un criterio débil en el gradiente.

4. Calcular k como en 1

$$\begin{aligned}\tilde{k} &= -1.0015 \cdot \alpha_- + 0.01 \cdot \tilde{\alpha} \\ k &= \max\{k^* \cdot \tilde{\alpha}, \tilde{k}\}\end{aligned}$$

5. Si $k < k^*$ o la malla no es convexa, ir al paso 3
6. de lo contrario salir

El valor de k^* que hemos probado con buenos resultados es -0.01 , éste valor va ligado al porcentaje de $\tilde{\alpha}$ esperado como “tope” mínimo para el área de los triángulos de la malla. Es posible probar con otros valores sin mayor condición que el de la existencia en la región de una malla de tales características.

El segundo paso del proceso es optimizar $F[k^*]$ para k^* tope hasta obtener convergencia o un criterio de paro.

Los métodos de optimización aquí utilizados son:

- L-BFGS: BFGS de Memoria Limitada
- Newton Truncado con Búsqueda en la Línea
- Newton Truncado con Región de Confianza
- Newton Puntual

Cada uno de los algoritmos son discutidos en un apartado final. Lo importante es señalar que en cada uno de ellos el número de pasos m requeridos del optimizador es distinto debido a la misma naturaleza del optimizador. Por ejemplo en el caso del Optimizador de Newton Puntual (por componente y simultánea) su convergencia es rápida cuando nos encontramos lejos de la solución pero su convergencia es lenta al final. En el caso de Newton Truncado, éste funciona muy rápido en funciones cuadráticas y es muy adecuado para funciones convexas. El método de Memoria Limitada lo hemos empleado con buenos resultados para un número de 3 vectores de reserva para la construcción de la solución haciendo uso de la información recabada en esos últimos pasos. El método de Newton Puntual es una modalidad interesante del método de Newton aplicado a los puntos interiores de la malla. Esta idea viene a raíz de la estructura de los elementos, donde un punto depende fuertemente de sus “vecinos” pero no así de los elementos más alejados de él. Su implementación será discutida posteriormente.

4.6.2 Sobre la implantación del funcional de suavidad

En el sistema UNAMALLA no está implementado la modificación del funcional de Suavidad representado en (4.5), se ha optado por una reescritura del mismo para facilitar su implantación y así agilizar los cálculos de la función y el gradiente.

De la expresión

$$\begin{aligned} f &= \frac{l - 2\alpha}{k + \alpha} \\ &= \frac{l + 2k}{k + \alpha} - 2 \end{aligned} \tag{4.37}$$

el funcional implantado dentro del sistema UNAMALLA es

$$f = \frac{l + 2k}{k + \alpha} \tag{4.38}$$

Toda vez que hagamos referencia a la implantación del funcional Adaptivo de Suavidad tendremos en cuenta esta forma de escribirlo, la cual es mucho más económica en el cálculo de los gradientes y Hessiana que la forma original.

4.6.3 Sobre la regularización

Por la forma que presenta los funcionales modificado de Suavidad y de Área, ambos requieren ser regularizados para que el optimizador opere adecuadamente y rechace puntos de prueba que genera a lo largo de su ejecución y por otra parte, aunque una malla G_k pueda ser óptimo de F_k puede ocurrir que alguno de sus triángulos sea de tal manera que α_q esté cerca de k y entonces presentar dificultades numéricas que se acarrearán al elegir esa configuración G_k .

El funcional implementado en el sistema UNAMALLA es

$$f_k = \frac{l + 2k}{k + \alpha} \quad (4.39)$$

la idea de la regularización es “empatar” en un punto $\hat{\alpha}$ ésta curva con una función cuadrática

$$q(\alpha) = a(\alpha - \hat{\alpha})^2 + b(\alpha - \hat{\alpha}) + c \quad (4.40)$$

sólo cuando así se considere. Para lograr que el empate en $\hat{\alpha}$ sea dos veces continuamente diferenciable los coeficientes de la cuadrática deberán cumplir que

$$\begin{aligned} a &= \frac{l + 2k}{(k + \hat{\alpha})^3} \\ b &= -\frac{l + 2k}{(k + \hat{\alpha})^2} \\ c &= \frac{l + 2k}{k + \hat{\alpha}} \end{aligned} \quad (4.41)$$

Para el punto de empate se ha considerado

$$\hat{\alpha} = -k + 0.015\alpha_-$$

y la regularización sólo se aplica cuando

$$\hat{\alpha} < 0.15\alpha_- \quad (4.42)$$

éste criterio experimental ha resultado ser adecuado y el que mejores resultados tiene, cualquier otra posibilidad en la consideración de $\hat{\alpha}$ y el momento de hacer uso de la regularización debe de guardar consistencia.

Para el caso del funcional modificado de Área el tipo de regularización empleada es idéntica; aquí los coeficientes son

$$\begin{aligned} a &= \frac{1}{(k + \hat{\alpha})^3} \\ b &= -\frac{1}{(k + \hat{\alpha})^2} \\ c &= \frac{1}{k + \hat{\alpha}} \end{aligned} \quad (4.43)$$

y al igual que en el funcional modificado de Suavidad, el valor del punto de empate $\hat{\alpha}$ y su criterio de uso son los mismos.

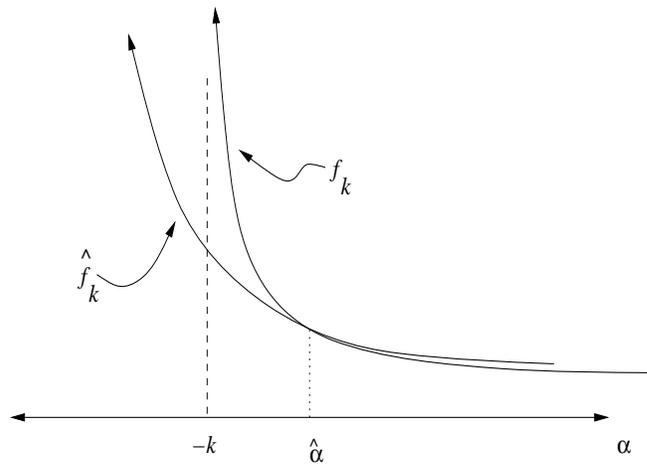


Figura 4.17: Regularización de f_k .

En el siguiente capítulo discutiremos combinaciones de éstos funcionales para obtener diversas propiedades geométricas en las mallas. En el apéndice C mostramos una colección de mallas sobre regiones muy irregulares donde hemos experimentado los resultados obtenidos.

Capítulo 5

Combinaciones interesantes entre funcionales Nuevos y Clásicos

Cuando describimos los funcionales clásicos hicimos hincapié en la necesidad de realizar combinaciones que nos conduzcan a resultados adecuados a las propiedades geométricas a observar. En el transcurso del capítulo anterior describimos funcionales que de manera automática nos conducen a mallas convexas, tal es el caso de los funcionales adaptivo de Suavidad y adaptivo de Área; ahora combinaremos sus propiedades con la de los funcionales clásicos.

La combinación de funcionales logra generar mallas con características deseables acordes a la ponderación de ellos. Para contar con una buena combinación es imprescindible lograr una buena normalización de los funcionales con el fin de que los elementos involucrados se encuentren en un mismo plano de valores [5].

Castillo y Barrera observaron la necesidad de combinaciones convexas entre funcionales de valores semejantes

$$F_c = \sigma F_1 + (1 - \sigma)F_2 \quad 0 \leq \sigma \leq 1 \quad (5.1)$$

Cada uno de los funcionales, aún contando con el mismo dominio de definición miden propiedades geométricas de magnitudes reales muy distintas; si deseamos combinar propiedades para trabajar con diversos funcionales, las magnitudes de los valores deben ser semejantes, ver Barrera [4].

Utilizando una adecuada normalización \tilde{F}_1, \tilde{F}_2 de los funcionales involucrados estamos en condiciones de llevar a cabo una combinación objetiva de las propiedades geométricas a observar. Para contar con una combinación en porcentajes, usaremos la combinación convexa descrita anteriormente

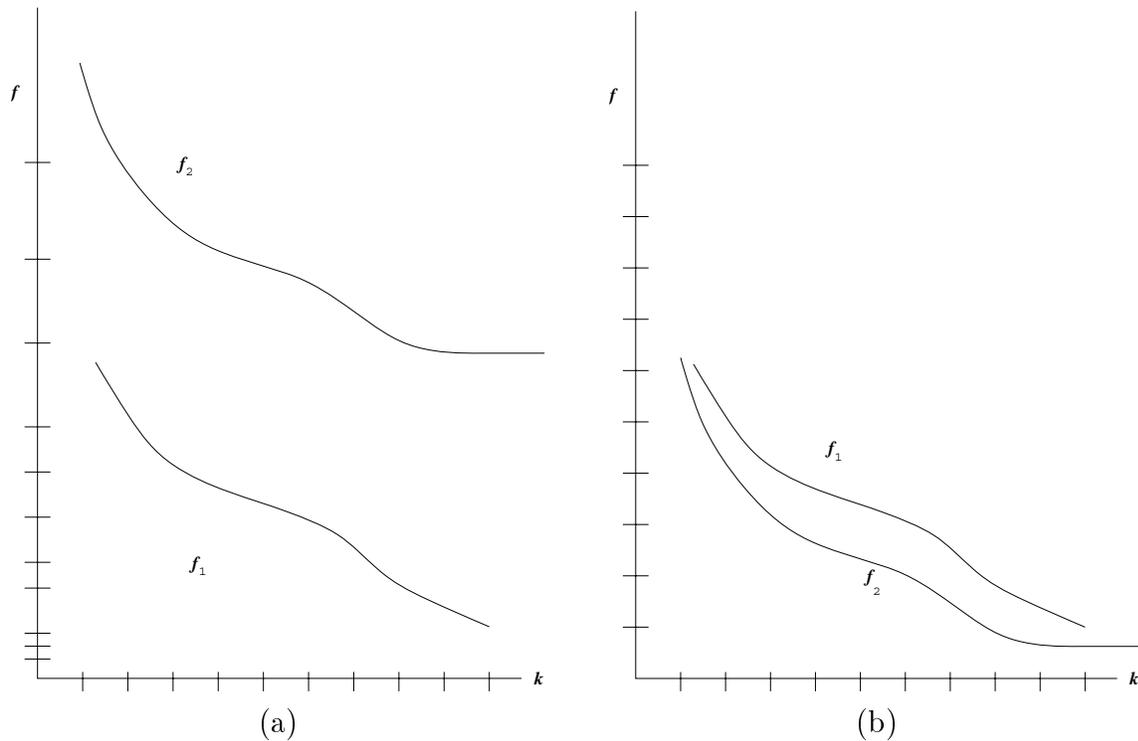


Figura 5.1: (a) Funcionales con orden de magnitud distinta. (b) Funcionales con orden de magnitud igual.

$$\tilde{F}_c = \sigma \tilde{F}_1 + (1 - \sigma) \tilde{F}_2 \quad 0 \leq \sigma \leq 1 \quad (5.2)$$

y de esta forma ponderar los funcionales \tilde{F}_1 y \tilde{F}_2 normalizados obteniendo la combinación promedio deseada. Un problema interesante es cómo encontrar una normalización lo más adecuada posible.

5.1 Normalización de Funcionales

La forma más sencilla de normalizar valores de los funcionales es hacerlo con respecto al valor esperado o deseado que se alcance en el mínimo. En el caso del funcional de Área es clave para entender esta idea. En [3] se mostró que el óptimo de éste funcional tiende a construir configuraciones donde los triángulos cuentan con área uniforme. De ésta manera, en regiones regulares y en configuraciones donde es posible conseguirlo. El valor del funcional en el óptimo es

$$\begin{aligned}
F_A^* &= \sum_{q=1}^N \alpha_q^2 \\
&= N\bar{\alpha}^2
\end{aligned} \tag{5.3}$$

por lo que el funcional normalizado de área vendría dado por

$$\tilde{F} = \frac{1}{N\bar{\alpha}^2} F_A \tag{5.4}$$

En combinación con otros funcionales, como por ejemplo Longitud o k -Suavidad, debemos ser muy cuidadosos al medir el rango de valores donde se espera encontrar cada funcional.

Para el caso del funcional de Longitud, en un trabajo [5] Barrera y Pérez proponen por normalización a una constante,

$$F_L = \frac{(n-2)(m-1)}{m^2} + \frac{(n-1)(m-2)}{n^2}$$

suponen que la malla puede ser inscrita en el cuadrado unitario con la cual, experimentalmente, observan que la normalización del funcional bajo el criterio anterior varía entre 1 y 10, claro está, en regiones con *aspect radio* pequeño.

Barrera y Tinoco [48] analizan los valores normalizados y proponen uno distinto que se encuentra más cercano a la realidad e invariante a la escala de la región. Sugieren observar al funcional de Longitud de la siguiente forma

$$F_L = \sum_{q=1}^N l_q \tag{5.5}$$

$$= \sum_{q=1}^N l_q - 2 \sum_{q=1}^N \alpha_q + 2 \sum_{q=1}^N \alpha_q \tag{5.6}$$

$$= \sum_{q=1}^N (l_q - 2\alpha_q) + 8 \cdot \text{área}(\Omega) \tag{5.7}$$

para de esta forma relacionarlo con el área de la región. Por una parte, $l_q - 2\alpha_q \geq 0$ (ver el capítulo anterior) luego el valor mínimo posible para f_l vendría cuando $l_q - 2\alpha_q = 0$ y en consecuencia el valor óptimo esperado sería $8 \cdot \text{área}(\Omega)$, por lo que la normalización del funcional queda

$$\begin{aligned}
 F_L &= \frac{1}{8 \cdot \text{área}(\Omega)} \sum_{q=1}^N l_q \\
 F_L &= \frac{1}{2N\hat{\alpha}} \sum_{q=1}^N l_q
 \end{aligned} \tag{5.8}$$

Para el caso del funcional de k -Suavidad, observemos el su valor sobre un triángulo

$$f_S[k](\Delta) = \frac{l - 2\alpha(\Delta)}{k + \alpha(\Delta)} \tag{5.9}$$

en capítulos anteriores se analizó que éste funcional tiene su óptimo sobre triángulos rectángulos isósceles, por lo tanto, en el óptimo su valor es de 2 y por consiguiente tenemos que el funcional de suavidad modificado normalizado se escribe como

$$F_S[k] = \frac{1}{2N} \sum_{k=1}^N \frac{l + 2k}{k + \alpha} \tag{5.10}$$

Para el funcional modificado de Área con k fija, independientemente del valor de k el funcional tiene un punto crítico en los triángulos de área uniforme. Con esto, para k fija la normalización del funcional modificado de área resulta

$$F_A[k] = \frac{k + \bar{\alpha}}{N} \sum_{q=1}^N \frac{1}{k + \alpha}. \tag{5.11}$$

En lo sucesivo cuando hablemos de estos funcionales tendremos en mente a los ya normalizados, a menos que explícitamente se comente lo contrario.

5.2 Combinaciones interesantes entre funcionales Adaptivos y Clásicos

A lo largo del trabajo, hemos comentado que algunos funcionales ofrecen propiedades geométricas interesantes y hemos visto la necesidad de su normalización cuando se desea combinar con otros funcionales a fin de lograr una malla que en principio presente las propiedades de la combinación.

En los trabajos [3], [4] y [5] con los funcionales desarrollados y estudiados hasta ese momento, comentamos la necesidad de combinar funcionales con el mismo propósito que ahora. Esos funcionales clásicos y modernos en su formulación nos permite trabajar con algunas combinaciones estables y observar buenas propiedades en su conjunto.

Nombre	Funcional 1	Funcional 2	Peso
t -Área – Longitud	t -Área	Longitud	.95, .7, .5
Área – Ortogonalidad	t -Área	Ortogonalidad	.5

Tabla 5.1: Combinaciones usuales entre funcionales clásicos y modernos.

En los experimentos realizados, se observa que un valor que produce buenos resultados y que podría tenerse en cuenta como un valor en un sistema automático es el de .9 en la combinación del funcional de Área y de Longitud; y es que con dicho valor se busca más la convexidad y poca suavidad de las líneas. Con éste parámetro y sobre un número importante de regiones poligonales complicadas se obtuvieron buenos resultados. En la Tabla 5.1 se puede consultar algunos valores para la combinación entre funcionales clásicos.

Con los nuevos funcionales la idea de combinar propiedades y explotar la estructura que presenta nos conduce a funcionales atractivos. En las siguientes secciones profundizaremos al respecto y revisará una modalidad interesante para llevar a cabo combinaciones.

Siguiendo la idea de los funcionales clásicos podemos combinar los funcionales de k -Área con el clásico de longitud. De igual manera podemos combinar el funcional de k -Suavidad con Área clásico acorde a la propiedad que deseamos observar.

Observemos que siendo uno de los funcionales adaptivo, la combinación con un funcional clásico da lugar a un funcional adaptivo, el de la combinación. Por consiguiente el esquema adaptivo seguirá presente.

Los valores indicados en la tabla anterior son aquellos que mejores resultados han registrado acorde a la propiedad geométrica deseada a observar. Los valores que mejores resultados han presentados y que consideramos como “automáticos” para

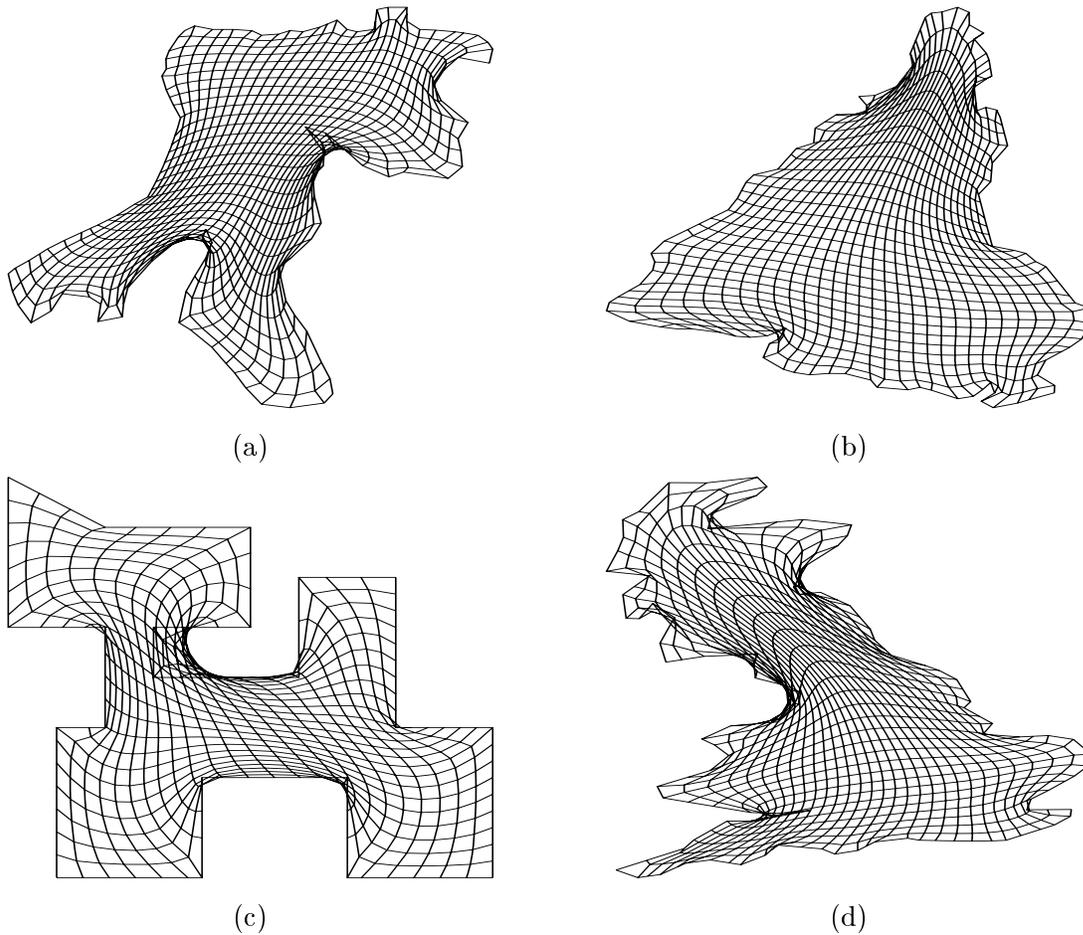


Figura 5.2: Mallas generadas por la combinación entre los funcionales t -Área y Longitud con un peso de $\sigma = .5$. (a) región de la bahía de la Habana, (b) región de Sudamérica, (c) región del Gato y (d) región del Reino Unido. Ninguna es convexa.

la obtención de mallas convexas y suaves son $.5$ tanto para la combinación k -Área-Longitud como para k -Suavidad-Área. Siguiendo el teorema de existencia de mallas convexas para los funcionales adaptivos, siempre que exista una malla convexa en Ω podemos obtener una malla convexa cercana a las propiedades geométricas deseables. Los valores “automáticos” son usados en el sistema UNAMALLA.

Sin embargo, existe una dificultad al combinar los funcionales. Esta se encuentra en el proceso adaptivo. Por ejemplo, para el caso de la combinación k -Área-Longitud se observa que el funcional de longitud pesa mucho en el proceso ya que éste funcional “estira” las líneas observando que el proceso adaptivo de la combinación se torna lento en lograr una malla convexa. Aspecto semejante encontramos en la combinación k -Suavidad-Área y es que el funcional de Área nos conduce a mallas uniformes pero

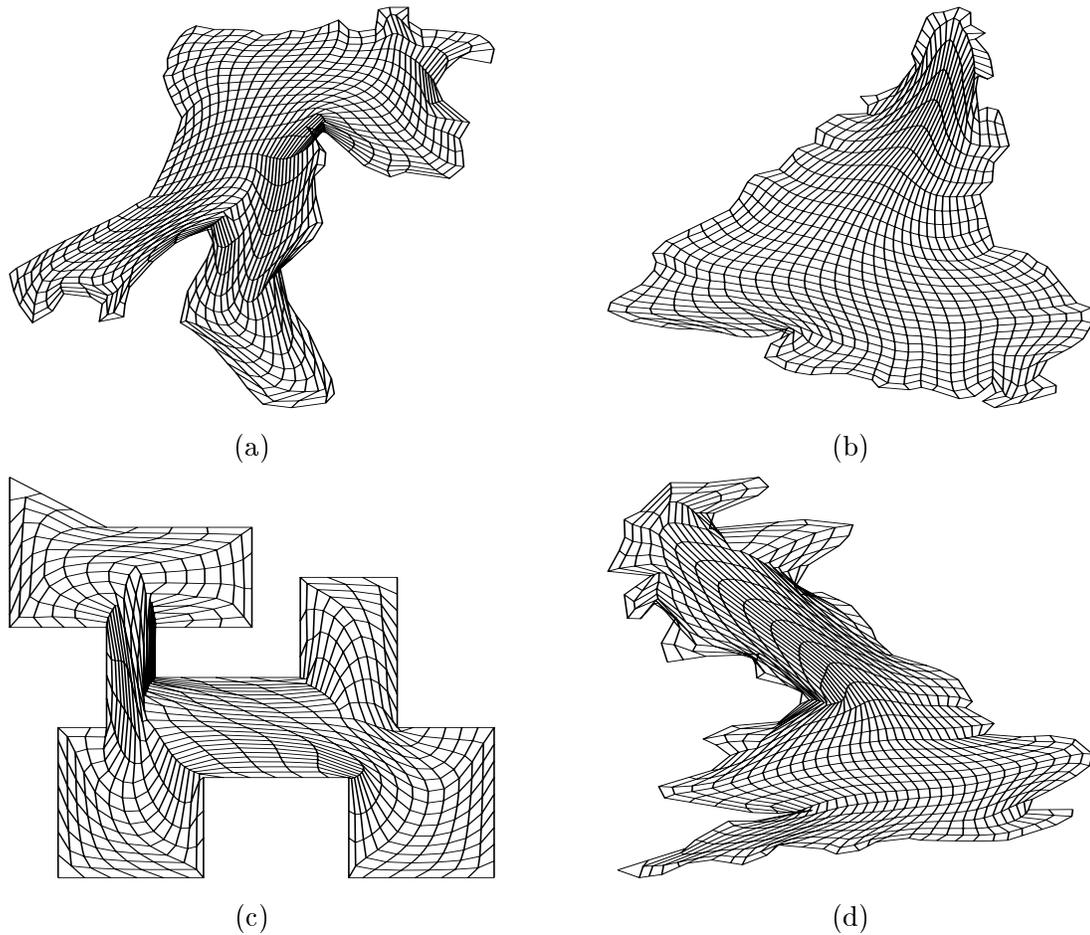


Figura 5.3: Mallas generadas por la combinación entre los funcionales t -Área y Longitud con un peso de $\sigma = .95$. (a) región de la bahía de la Habana, (b) región de Sudamérica, (c) región del Gato y (d) región del Reino Unido. Ninguna es convexa.

con líneas no suaves, muy “quebradas” y entonces el funcional de k -Suavidad intenta suavizar las líneas dando como resultado que el proceso se vuelva lento ya que lo estamos forzando al trabajar con ambas propiedades geométricas simultáneamente.

La idea inmediata es combinar esfuerzos, tomar lo mejor que tenemos de los funcionales adaptivos, esto es, nos conducen a mallas convexas; y combinarlo de alguna forma para la obtención de una malla con la característica deseada. Describamos el procedimiento directo a que nos conduce los funcionales adaptivos.

Elijamos un funcional discreto f_c como combinación lineal de otros dos, f_1 y f_2 en la forma

Nombre	Funcional 1	Funcional 2	Peso
k -Área – Longitud	k -Área	Longitud	.5, .75, .9
k -Suavidad – Área	k -Suavidad	Área	.25, .5, .95

Tabla 5.2: Tabla de combinaciones usuales entre los nuevos funcionales.

$$f_c = (1 - \sigma)f_1 + \sigma f_2, \quad 0 \leq \sigma \leq 1, \quad (5.12)$$

este funcional es el que deseamos optimizar,

$$\min_{G \in \mathcal{D}} f_c \quad (5.13)$$

Ahora bien, dentro del proceso para la optimización de f_c está inmerso una adaptividad en k cuyo objetivo es acercarnos a k^* para el funcional f_1 ; por ejemplo, para el funcional de k -Suavidad o bien k -Área. Bajo éste hecho sería deseable observar la adaptividad de f_1 para generar una homotopía de f_1 hacia f_c conforme $k \rightarrow k^*$, ya que el primer funcional nos acerca a mallas convexas de forma rápida.

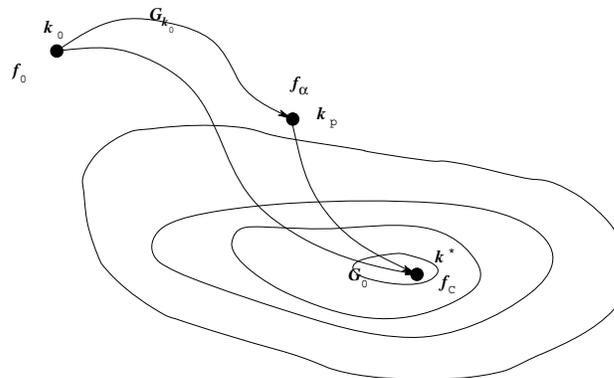


Figura 5.4: Doble homotopía entre funcionales.

Bajo esta idea, veamos cómo definir una homotopía entre funcionales. Primero veamos alguna definición y ejemplos para entender la homotopía en f antes de combinarlo con la adaptividad de los k -funcionales.

5.3 Homotopía entre funcionales

Nuestro objetivo es lograr una malla óptima de la combinación entre dos funcionales f_1 y f_2 , esto lo podemos conseguir si partimos de uno de los funcionales y describimos

una trayectoria hacia el punto de combinación entre los funcionales f_c . De manera gráfica vemos la Figura 5.5

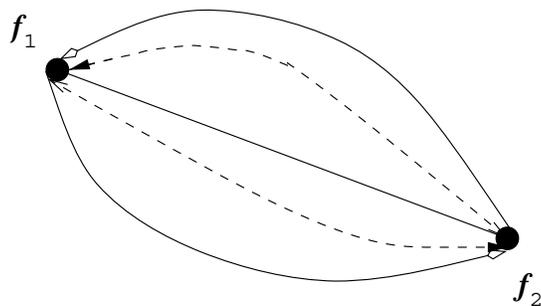


Figura 5.5: Trayectorias entre funcionales.

Necesitamos que la trayectoria sea suave y que nos conduzca rápidamente hacia el punto de la combinación f_c . La homotopía desde un funcional f_1 hacia la combinación con el funcional clásico puede ser una combinación lineal convexa de la forma

$$\tilde{f}_c = (1 - \tilde{\sigma})f_1 + \tilde{\sigma}f_c, \quad 0 \leq \tilde{\sigma}(t) \leq 1 \quad (5.14)$$

con $\tilde{\sigma}$ continuamente diferenciable y creciente.

Bajo esta idea, necesitamos construir una función continuamente diferenciable $\tilde{\sigma} = \tilde{\sigma}(t)$ con

$$0 \leq t \leq 1, \quad 0 \leq \tilde{\sigma}(t) \leq 1 \quad (5.15)$$

Una parametrización del intervalo $0 \leq t \leq 1$ funciona.

Algunas parametrizaciones del intervalo $0 \leq t \leq 1$ son

$$0 \leq t \leq 1, \quad \tilde{\sigma}(t) = t^2 \quad (5.16)$$

$$\tilde{\sigma}(t) = t^3 \quad (5.17)$$

$$\tilde{\sigma}(t) = \text{sen}\left(\frac{\pi}{2}t\right) \quad (5.18)$$

Haciendo una reparametrización de $\tilde{\sigma}$, \hat{f}_c queda como

$$\hat{f}_c = (1 - \hat{\sigma})f_1 + \hat{\sigma}f_2, \quad \text{con } 0 \leq \hat{\sigma}(t) \leq 1 - \sigma \quad (5.19)$$

Aquí la reparametrización $\hat{\sigma} = \hat{\sigma}(t)$ resulta una función continuamente diferenciable y creciente. Esta expresión una homotopía entre funcionales que nos conduce desde el funcional f_1 hacia la combinación entre funcionales f_c .

La homotopía entre funcionales la usaremos para la combinación entre funcionales adaptivos y clásicos de tal manera que iniciemos el proceso con el funcional adaptivo, por ejemplo k -Suavidad, hasta lograr la combinación deseada con el funcional de Área. En el siguiente apartado describiremos esta idea y cómo llevarla a cabo teniendo en cuenta el proceso de adaptividad presente.

5.4 Homotopía y Adaptividad, un esquema práctico

La idea de llevar a cabo la homotopía no sólo se justifica en el uso de los funcionales adaptivos, la razón esencial es la de obtener de manera rápida la propiedad de la combinación. Para esto lo natural es iniciando con el funcional adaptivo que nos lleva a mallas convexas y poco a poco acercandonos a la combinación deseada. Esto hará que la influencia del funcional clásico se refleje al final de la trayectoria modificando poco a poco la convexidad en su entrada en funcionamiento e impulsando con ello la rapidez del proceso.

Surgen varios problemas prácticos, la parametrización $\hat{\sigma}$ varía en t . ¿Cuál sería la interpretación de t o su uso en la descripción de un algoritmo de homotopía entre funcionales? Otro más: tenemos dos procesos, uno que va con el método adaptivo, la adaptividad en k u homotopía hacia k^* , y ahora la homotopía entre funcionales. ¿Cómo llevar a cabo esta doble homotopía? A continuación describiremos estos problemas, la implantación práctica de una homotopía entre funcionales y la implantación práctica de la adaptividad y la homotopía entre funcionales, lo que en adelante conoceremos como doble homotopía.

Observemos que, en la homotopía del proceso adaptivo, partimos de k_0 hasta k^* , valor tope elegido y donde se estima la malla es convexa y con las propiedades geométricas del funcional. Consideremos este conjunto de valores de k que se irán generando en el intervalo $[k_0, k^*]$ como punto de partida para definir una homotopía lineal que nos conduzca a f_c cuando k se aproxime a k^* .

Siguiendo la idea anterior, la homotopía lineal que tenemos en mente la escribimos por

$$\hat{\sigma}(k) = \frac{1 - \sigma}{k_0 - k_*}(k_0 - k) \quad (5.20)$$

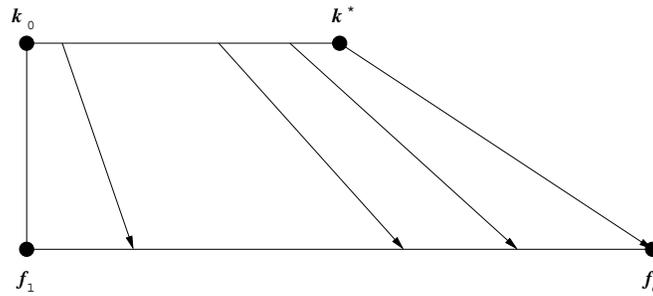


Figura 5.6: Homotopía de los funcionales a partir de la homotopía en k .

esta función es creciente, es lineal y define perfectamente la homotopía en f deseada.

Esta idea es interesante ya que las propiedades de la homotopía de los funcionales van influyendo en la búsqueda de la convexidad por el funcional adaptivo ver Figura 5.7.

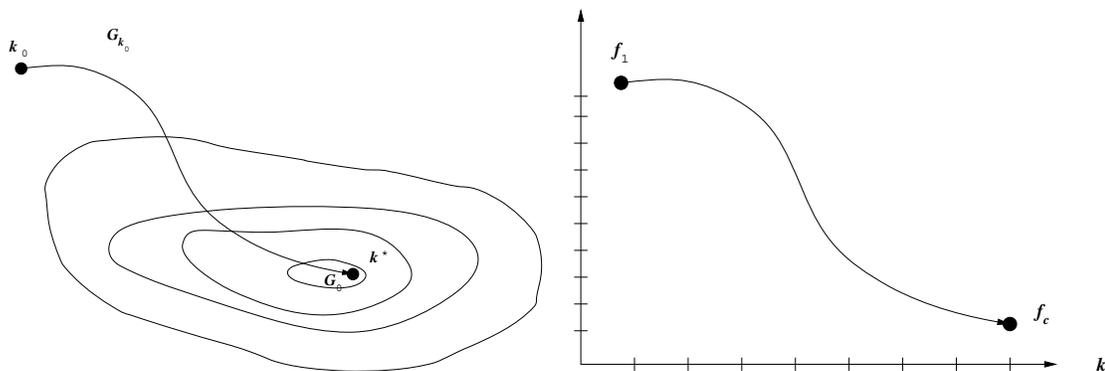


Figura 5.7: Homotopía de los funcionales a partir de la homotopía en k .

Es de tomar en cuenta que ésta homotopía ofrece muchas ventajas en su implantación. En la búsqueda de la doble homotopía ocurre que k va cambiando y entonces el funcional adaptivo cambia y luego f_c varía acorde a k .

Recordemos que los funcionales adaptivos en su implantación al sistema se encuentran “protegidos” para que el optimizador rechaze puntos de prueba computacionalmente inestables; dicha protección podría resultar no ser la conveniente para la malla en curso y entonces aceptar mallas con triángulos problemáticos.

Esta posibilidad hace que el cambio de k no sea monótono y eso nos hace retornar a un conjunto G_k que nada tiene que ver con $G_{k_{anterior}}$, entonces el peso de la combinación debe regresar en la misma proporción.

La dependencia de la homotopía de los funcionales con la adaptividad hace flexible

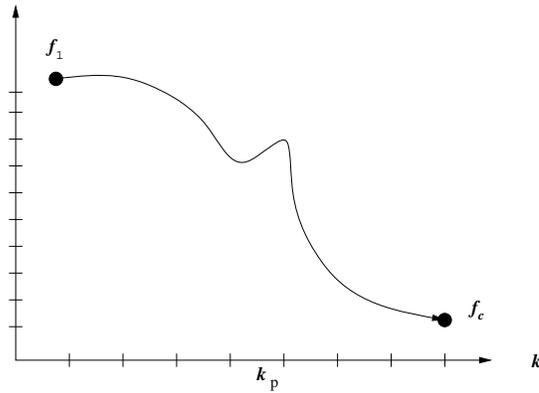


Figura 5.8: Automatización del reinicio de la homotopía al depender de k .

el proceso haciendo que automáticamente “reinicie” al encontrar un problema en k seguir un camino distinto al primero pero buscando de nuevo la convergencia a f_c . Esta técnica intrínseca de la homotopía elegida para los funcionales la hace muy atractivo.

Finalmente, en el siguiente capítulo describiremos un sistema automático para la generación de mallas planas en regiones irregulares haciendo uso de los nuevos funcionales adaptivos de Área y de Suavidad. Describiremos los requerimientos del sistema, las tareas deseadas que contempla, su funcionalidad, y por último presentar el sistema y su manual operativo.

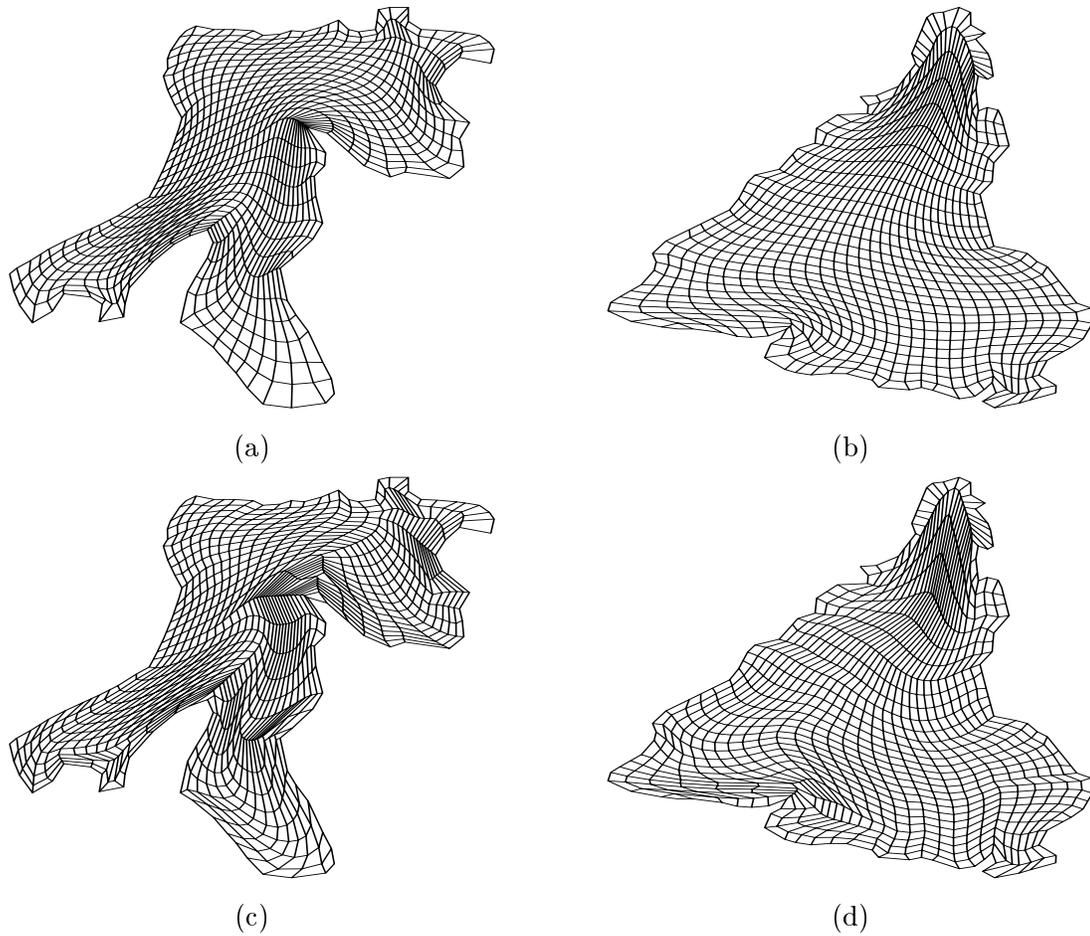


Figura 5.9: Mallas convexas generadas por la combinación entre los funcionales k -Área y Longitud con un peso de $\sigma = .5$. (a) región de la bahía de la Habana, (b) región de Sudamérica, y con un peso de $\sigma = .9$ (c) región de la bahía de la Habana y (d) región de Sudamérica.

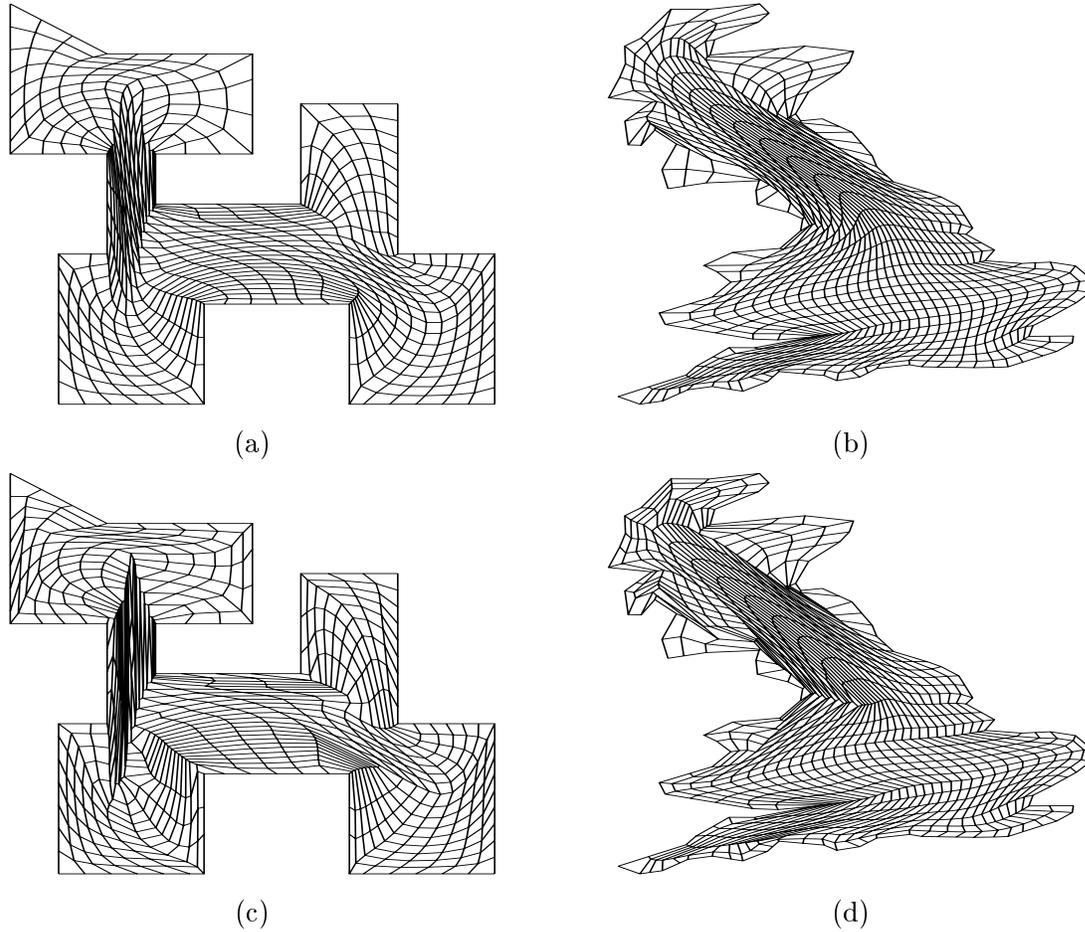


Figura 5.10: Mallas convexas generadas por la combinación entre los funcionales k -Área y Longitud con un peso de $\sigma = .5$. (a) región del Gato, (b) región del Reino Unido, y con $\sigma = .9$ (c) región del Gato y (d) región del Reino Unido.

Capítulo 6

Requerimientos de un sistema automático generador de mallas y su diseño

A lo largo del capítulo describiremos los requerimientos necesarios para el desarrollo de un sistema basado en los nuevos funcionales discretos de Área y Suavidad. Con base a los requerimientos propondremos un diseño para el sistema.

6.1 Análisis General

6.1.1 Objetivo

Construir un sistema automático para la generación de mallas rectangulares en regiones planas poligonales usando los nuevos funcionales de área y suavidad. Un sistema multiplataforma `Unix` orientado a `WorkStation` con sistema de ventanas `X Windows`.

6.1.2 Panorama

Como fué comentado en el prefacio de este trabajo, la primera referencia al sistema se encuentra en el programa `Gridgen` [2] y en el trabajo de investigación que se realizó [3]. Parte de las rutinas descritas en este sistema para la construcción de la malla inicial han sido adaptadas al sistema. A lo largo de los años según hemos podido experimentar con nuevas modalidades y construir versiones preliminares de sistemas que generan mallas; con esto hemos ganado la experiencia necesaria para diseñar un

sistema profesional que retome nuestra inquietud de un sistema automático [6] con el que podamos obtener mallas planas convexas. Los trabajos recientes de Tinoco–Barrera [49] nos han permitido lograr que ésto ocurra. En García [23] se muestra un sistema experimental basado en PC DOS limitado a la memoria base existente 640Kb y con gráficos de la biblioteca del compilador. Ahora nos interesa especificar un sistema mucho más versátil, totalmente gráfico y multiplataforma.

A continuación describiremos las características deseables de un sistema y su configuración por módulo de tareas.

6.1.3 Descripción del proyecto

El sistema deberá contar con dos funciones principales: el trabajo con contornos y el de las mallas. Es importante que algunas aplicaciones tipo estuvieran consideradas. Con detalle se explica cada una de las funciones que se debe contemplar:

- **Trabajo con contornos.** Para obtener una malla previa a la óptima el usuario debe trabajar sobre un contorno, construirlo, definir las cuatro subfronteras que deberán ser mapeadas al cuadrado unitario; es decir, desde las que deseamos “tirar” líneas curvilíneas. Tomando un contorno en memoria el sistema debe ser capaz de crear una malla inicial o previa al proceso de optimización.
- **Trabajo con mallas.** Contando con una malla inicial podemos trabajar con ella para obtener una óptima en el sentido de los funcionales discretos. El proceso de optimización deberá ser gráfico con el fin de que el usuario visualice sus resultados según se vayan obteniendo. También se pueden considerar algunas aplicaciones de las mallas suaves y convexas como lo son el refinamiento y la solución numéricas de algunas EDP.

6.1.4 Rango de alcance

La idea del sistema es automatizar la obtención de mallas planas haciendo uso de los nuevos funcionales discretos descritos en capítulos anteriores y en [48]. Para contar con una herramienta capaz de generar de manera efectiva mallas convexas y suaves sobre una región plana arbitraria. La automatización de un sistema de estas características permitirá al usuario poco entendido en la materia obtener resultados inmediatos en un problema específico: la discretización de una región para su posterior análisis.

El sistema deberá estar orientado a plataformas Unix con despliegue gráfico X Windows. Será multiplataforma, explotando las ventajas del sistema de ventanas X Windows logrando un sistema totalmente gráfico. El sistema será automático en el

sentido de que con sólo unas pocas pulsaciones de teclas será posible lograr una malla adecuada.

Es deseable que se tome en cuenta un esquema modular de interacción con fin de que el crecimiento y mantenimiento del sistema quede garantizado.

6.1.5 Características del Usuario

Los usuarios potenciales del sistema más que profesionales en la construcción y obtención de mallas óptimas son expertos en las áreas donde desean aplicar los resultados por lo que solamente desean obtener una discretización óptima o bien, una razonable de la región de estudio. Para estos usuarios es necesario que el sistema cuente con una modalidad automática para que puedan obtener una malla óptima o aceptable; es decir, suave y convexa.

Otro de los usuarios tipo para el sistema es el “experto” en la generación de mallas y aquel que desea experimentar en la construcción de la malla y la obtención de la malla óptima por los funcionales discretos, así como con diversos métodos de optimización. Para estos usuarios el sistema deberá contemplar opciones de botón y de menú para que puedan utilizar las herramientas de edición de contorno, la dimensión de la malla, las combinaciones de funcionales discretos y poder experimentar con los métodos de optimización.

En cualesquiera de los casos del usuario tipo, éstos deben contar con un mínimo de conocimientos en el uso del sistema de ventanas X `Windows` y así poder interactuar con el sistema satisfactoriamente.

6.1.6 Restricciones generales para el usuario

El usuario trabajará sobre segmentos de línea recta para construir y operar los contornos.

En la construcción de la malla inicial los puntos sobre la frontera se distribuyen uniformemente buscando conservar la forma de la frontera [37] y son sobre éstas partes fijas con las que se trabaja para obtener la malla. La construcción inicial de la malla dependerá fuertemente de esta distribución, parametrización que influye en los métodos discretos, consultar [29]. A corde con la teoría: si es posible construir una malla convexa sobre la región de estudio será entonces posible hallarla con el sistema, a través de los funcionales discretos implantados. Para las aplicaciones EDP, solamente deberán ser resueltas algunas ecuaciones diferenciales elípticas con condiciones a la frontera fijas a los cuatro segmentos de frontera de la malla.

6.1.7 Requerimientos específicos

Requerimiento funcional 1: Menú principal

1. **Resumen:** Se requerirá de dos ventanas, una donde se visualicen los gráficos y la otra desde la que se realice el control de las acciones con los contornos, las mallas, un acceso para la ayuda en línea y uno para la salida del sistema.
2. **Entrada:** Para la ejecución del sistema el usuario no deberá requerir parámetro alguno de entrada.
3. **Procesamiento:** Controla y distribuye las tareas principales del sistema.
4. **Salida:** Al finalizar el sistema no se devuelve información. Toda ésta se realiza en curso y sobre el sistema gráfico.

Requerimiento funcional 2: Construcción de un contorno

1. **Resumen:** Es necesario contar con una herramienta que permita la construcción de un contorno. La herramienta de captura o creación del contorno es muy útil en áreas de experimentación con datos.
2. **Entrada:** No hay elementos de entrada ya que se construirá la región de trabajo.
3. **Procesamiento:** Se define los segmentos de contornos introduciendo puntos al ir pulsando botones del mouse. Conforme se introduce el contorno los datos del mismo se van añadiendo a una lista en memoria y éstos datos son desplegados en pantalla y enumerados. No se debe permitir que se introduzcan puntos repetidos.
4. **Salida:** Al finalizar la captura, la información del contorno permanece en memoria y será sobre la que trabajará el usuario durante la ejecución.

Requerimiento funcional 3: Lectura de Contornos

1. **Resumen:** El sistema debe contemplar mecanismos para tomar la información de un contorno en archivo con un formato específico y llevarlo a memoria.
2. **Entrada:** Nombre de archivo en disco.
3. **Procedimiento:** Mediante un **browser** el usuario debe elegir el nombre del archivo y señalar su ubicación.

4. **Salida:** El contorno en memoria. Si el formato del contorno cuenta con una asignación de las fronteras éstas serán señaladas en pantalla de no ser así de igual forma podrá observarse un letrero en la caja de estado indicando éste hecho.

Requerimiento funcional : Definición de fronteras

1. **Resumen:** Es muy importante asignar los segmentos de frontera de la región que serán mapeadas a los correspondientes del cuadrado unitario.
2. **Entrada:** Un contorno previamente definido, sea en pantalla o por fichero. Si las fronteras estan ya asignadas y podemos entonces modificarlas.
3. **Procesamiento:** Se debe tomar en cuenta una modalidad para elegir automáticamente una distribución de segmentos de frontera, buscando uniformidad a 4 partes del contorno. Usando opciones de control y/o campos de entrada se modificará la elección permitiendo elegir una que sea aceptable para su posterior uso.
4. **Salida:** Al finalizar se contará con un contorno donde no haya pico hacia adentro en las esquinas y en principio pueda obtenerne una malla convexa; es decir, que no se doble.

Requerimiento funcional 5: Guardar información del contorno

1. **Resumen:** Una vez creado el contorno es útil contar con la información en disco bajo formato convenido.
2. **Entrada:** La información del contorno en memoria y el nombre del archivo a guardar los datos.
3. **Procedimiento:** Al elegir la opción de salvar la información debe ser desplegada en pantalla una caja de diálogo, un **browser**, donde se podrá señalar la ubicación donde quedará el archivo y su nombre.
4. **Salida:** El archivo del contorno en disco.

Requerimiento funcional 6: Generación de la malla inicial

1. **Resumen:** Una vez que se cuenta con un contorno aceptable, podemos generar una malla por interpolación que nos permita usarla como punto de partida o inicial para el procedimiento de optimización involucrado en la generación de la malla por los métodos discretos. El método TFI de interpolación que

comunmente usamos conserva la frontera de la región poligonal y propaga las singularidades de la frontera hacia el interior de la región. Para los detalles puede consultar [47].

2. **Entrada:** Un contorno aceptable en la asignación de frontera. La dimensión de la malla.
3. **Procesamiento:** A través de campo de entrada se introducirá la dimensión de la malla y al pulsar la opción diseñada se construirá la malla por el método de Interpolación Transfinita TFI.
4. **Salida:** Si la elección de dimensión de la malla es “adecuada” para lograr una malla que conserve la forma de la región poligonal sin que el contorno deje de ser aceptable, obtendremos una malla inicial. En caso contrario no podremos trabajar con el resultado. El concepto de aceptabilidad de un contorno viene definido en [37].

Requerimiento funcional 7: Lectura de Mallas

1. **Resumen:** Necesitamos trabajar con mallas previamente generadas o bien continuar el trabajo que realizamos sobre ellas. Para esto, es indispensable poder llevar a la memoria una malla en disco.
2. **Entrada:** Un archivo de mallas en formato convenido.
3. **Procedimiento:** A través de una caja de diálogo el usuario proveerá el nombre del archivo o bien mediante el cursor y las opciones del **browser** navegará por los directorios del sistema hasta señalar el archivo deseado.
4. **Salida:** Al elegir el archivo de mallas los datos se cargaran en memoria y deben ser desplegados en pantalla.

Requerimiento funcional 8: Herramientas de despliegue gráfico

1. **Resumen:** Una herramienta que ha abierto mucho el estudio posterior de los métodos discretos y potencia la obtención de los resultados ha sido el desplegar la malla conforme se va obteniendo en pantalla, observarla en puntos discretos y observar cómo va esta evolucionando hasta su convergencia, así como para el análisis de los resultados.
2. **Entrada:** Una malla en memoria.

3. **Procedimiento:** Con la malla en memoria mediante el uso de controles deslizadores y botones de opción se cambiará la caja de despliegue para poder hacer un zoom hacia adentro o hacia afuera y de un lado a otro. De igual forma la caja se podrá mover con el fin de observar parte de la malla.
4. **Salida:** Despliegue en pantalla.

Requerimiento funcional 9: Optimización de una malla

1. **Resumen:** La optimización de una malla va en el sentido de lograr propiedades óptimas acorde a los funcionales discretos, ésto es que la malla obtenida esté cercana a las propiedades del funcional. Aquí se deberá emplear los nuevos funcionales discretos para la obtención de mallas suaves y convexas.
2. **Entrada:** Una malla en memoria.
3. **Procedimiento:** A través de una serie de opciones y menús el usuario podrá elegir la opción que desee o bien usar las predeterminadas. Al ejecutar la opción de realizar el procedimiento deberá aparecer en pantalla una ventana donde se muestre el desarrollo de las iteraciones así como en pantalla podrá verse gráficamente cómo evoluciona la malla hacia el óptimo.
4. **Salida:** Una malla optimizada.

Requerimiento funcional 10: Guardar información de la malla a disco

1. **Resumen:** Una vez obtenida la malla es necesario guardar la información en disco para su posterior uso.
2. **Entrada:** Una malla en memoria y el nombre de archivo donde se guardarán los datos.
3. **Procedimiento:** Al igual que en el procedimiento de lectura podemos a través de un **browser** depositar la información de la malla en el directorio deseado así como indicar el tipo de formato en que será guardada la información.
4. **Salida:** La malla en archivo de disco.

Requerimiento funcional 11: Refinamiento uniforme

1. **Resumen:** El refinamiento de la malla es una herramienta útil en muchas aplicaciones. Contamos con una malla sobre la región en algunas aplicaciones deseamos hacer algún cálculo con precisión en subregiones o partes de la región.

2. **Entrada:** Una malla convexa.
3. **Procedimiento:** Por refinamiento uniforme entenderemos una interpolación entre las líneas originales. En la opción de refinamiento uniforme deberemos elegir el número de líneas que serán interpoladas entre dos ya existentes y realizar entonces la interpolación.
4. **Salida:** La malla en pantalla con el refinamiento uniforme deseado.

Requerimiento funcional 12: Solución numérica de algunas EDP

1. **Resumen:** Una aplicación de sumo interés es resolver algunas ecuaciones diferenciales parciales sobre las mallas convexas logradas. Nuestro interés es experimentar con algunas EDP elípticas.
2. **Entrada:** Una malla convexa. Los parámetros de la ecuación elíptica y las condiciones a la frontera.
3. **Procedimiento:** A través de un esquema de diferencias finitas se resolverá el sistema y será desplegada en pantalla la superficie solución.
4. **Salida:** Solución obtenida sea en pantalla, o bien, en archivo con formato MESH de Geomview.

6.1.8 Requerimientos de Interfase externa

User Interfase

La interfase con el usuario debe ser amigable, debe usar recursos comunes como **mouse**, teclado y contar con la posibilidad de usar alguno de ellos cuando el otro no exista. La idea es contar con un sistema totalmente gráfico por lo que la interfase debe ser “natural” al usuario en el manejo de las opciones. Conectar algunas tareas con otras de manera que el proceso sobre los contornos y las mallas se encuentren plenamente identificadas.

Hardware Interface

El sistema no deberá depender de ningún hardware. No deberá estar orientado a plataforma de sistemas específicos, por el contrario ser lo más amplio posible. No dependerá el despliegue gráfico de los **pixeles** de la pantalla gráfica o de especificaciones de tarjeta gráfica alguna.

Software Interface

Para el despliegue gráfico se usará instrucciones basadas en `Xlib` de manera que sea lo más independiente de cualquier otra biblioteca gráfica. Sólo deberá emplearse rutinas de distribución libre y en su caso bibliotecas gráficas soportadas en casi cualquier sistema.

6.1.9 Requerimientos no funcionales

El sistema debe tener la capacidad de ser llamado desde una `WorkStation` sin límite de usuarios, excepto quizá las limitaciones de recursos en la memoria del sistema permita.

El sistema debe ser rápido en ejecución y todo en pantalla. Es preferible que no se generen archivos en disco evitando así que permisos no declarados puedan detener el proceso de ejecución.

Una buena colección de mensajes desplegados apropiadamente ayudará a que los usuarios puedan saber las razones por las que no es posible realizar una operación o bien si ésta fue llevada a cabo satisfactoriamente.

El sistema debe contar con los siguientes atributos:

- **Disponibilidad** En todo momento el usuario podrá ejecutar el sistema siempre que se encuentre trabajando en el sistema de ventanas de la computadora. El número de usuarios no deberá ser impedimento para su ejecución, sólo dependerá de los recursos de memoria.
- **Mantenimiento** El mantenimiento del sistema jugará un papel importante en futuros desarrollos. Contemplar que el sistema cuente con una infraestructura que permita actualizar, sustituir o eliminar sus partes permitirá el crecimiento del mismo con costo mínimo. Si la construcción del sistema se encuentra muy bien documentada en sus partes permitirá que un mayor número de personas puedan participar en futuros desarrollos o aplicaciones del mismo.

6.2 Diseño del sistema

Con base en la especificación de los requerimientos podemos considerar el diseño del sistema identificando las tareas que se han de realizar y la forma de interactuar entre ellas. De igual forma describiremos los lenguajes de programación requeridos de acuerdo a las bibliotecas o rutinas creadas en proyectos anteriores y de uso específico.

6.2.1 Modularidad

El sistema será modular. De acuerdo a lo descrito en la especificación de requerimientos, el sistema debe ser diseñado siguiendo la interacción de pequeñas tareas o módulos para su integración en el sistema.

Por una parte estamos interesados con el trabajo exclusivo del contorno, desde su creación hasta la edición de las sub-fronteras para la posterior generación de la malla por TFI. Por otro lado estamos interesados en darle un tratamiento a las mallas generadas previamente y obtener mallas óptimas en el sentido de los funcionales discretos que trabajamos. De esta manera podemos distinguir dos módulos principales: Contornos y Mallas

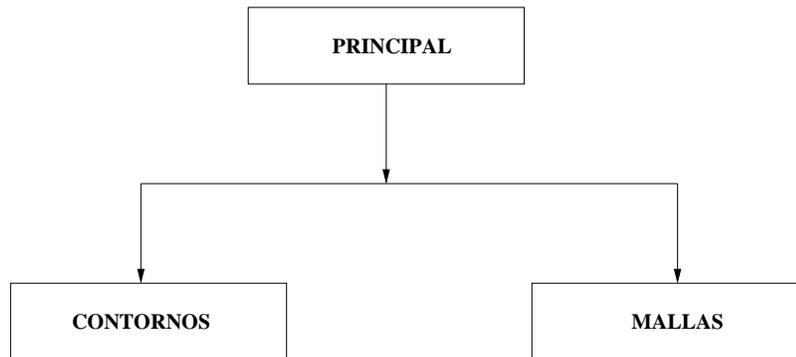


Figura 6.1: Módulos principales del sistema

Trataremos cada una de las tareas integradas a los módulos para formar el diagrama de flujo de las operaciones contempladas en el sistema.

Módulo Contornos

El trabajo con contornos abarca tanto su construcción como la asignación de los segmentos de frontera para generar la malla rectangular. Por una parte nos interesa construir los contornos así como considerar aquellos obtenidos previamente por métodos o mecanismos diversos y en archivos de disco.

Uno de los primeros pasos será la obtención de un contorno. Una forma es leer la información del contorno desde un archivo y la otra construirla en pantalla. La primera debe seguir un formato establecido que contemple los puntos del contorno poligonal y una descripción de los segmentos de frontera a los que será mapeado el cuadrado unitario. Si los segmentos de frontera no están establecidos, no podrá construirse una malla inicial.

La construcción de un contorno en pantalla debe ser muy claro. Nuestros dominios

son polígonos cerrados en el plano así que debe considerarse un mecanismo que cierre el contorno. De igual manera debe considerarse evitar introducir puntos repetidos. Luego de la entrada de los datos se pasará inmediatamente a la asignación de los segmentos de frontera.

Contando con un contorno en memoria, podrá optarse por guardar la información o bien generar la malla inicial para su uso posterior en el módulo de mallas.

Bajo la descripción del orden de las tareas que hemos planteado, el módulo de contornos seguirá el diagrama de flujo de la Figura 6.2.

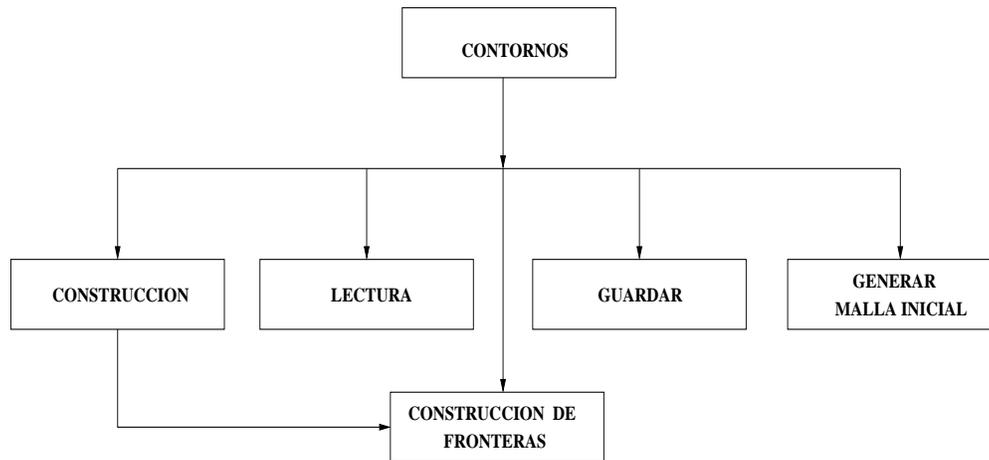


Figura 6.2: Diagrama de flujo de las tareas de un contorno.

Módulo de Mallas

El trabajo sobre mallas nos permite obtener una malla optimizada bajo los nuevos funcionales así como realizar algunas aplicaciones y observar en detalle las mallas.

Entre las principales tareas se encuentra la de llevar a memoria una malla en disco para su posterior procesamiento. Con la malla en memoria podemos emplear los diversos funcionales discretos a fin de optimizarla usando para dicho efecto diferentes métodos de optimización de gran escala. Una vez obtenida la malla óptima podemos tomarla como entrada para el proceso siguiente y es que la malla deberá estar en memoria y el sistema nos deberá llevar al menú anterior con la finalidad de poder seguir con ese módulo hasta que así lo considere el usuario.

Si la malla en uso es convexa podremos trabajar con las aplicaciones definitas para tal fin. Por ejemplo podemos resolver alguna EDP sobre la malla y si los resultados no son adecuados o simplemente deseamos experimentar con un malla más fina deberemos contemplar la posibilidad de rápidamente acceder al refinamiento uniforme y luego volver a ejecutar el módulo EDP y obtener lo deseado. Esto nos llevará a

considerar las aplicaciones sobre mallas convexas como una opción dentro del módulo principal de mallas.

En cualquier momento del proceso podemos guardar los resultados bajo el formato convenido.

Por otra parte, una herramienta muy poderosa es la visualización a detalle de la malla, sea obtener un acercamiento de zonas complicadas o ver algunas características geométricas y visuales de los segmentos curvilíneos. Esta tarea deberá estar disponible en todo momento desde el módulo de las mallas..

Bajo este esquema de trabajo, el diagrama de flujo de las tareas agrupadas en el módulo de mallas se presenta en la Figura 6.3.

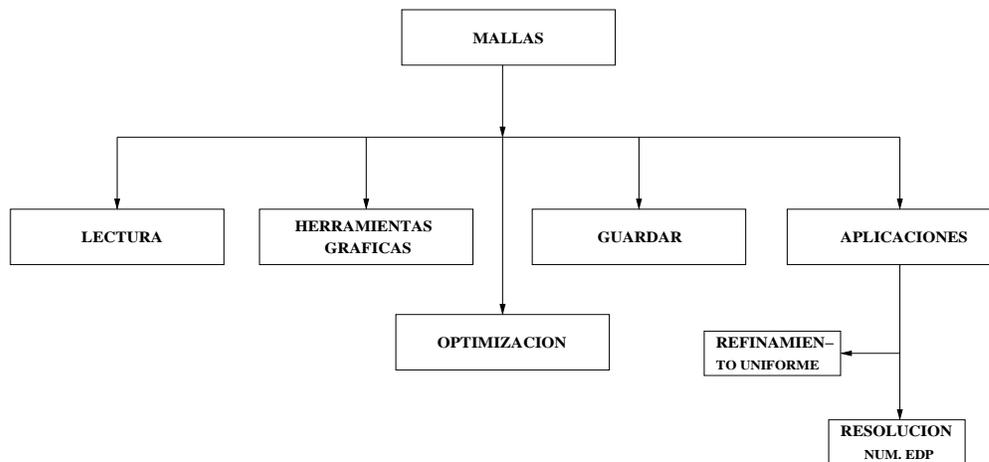


Figura 6.3: Diagrama de flujo de las tareas en el módulo de mallas.

Una vez descritas las tareas que agrupan cada módulo del sistema, se tendrá un diagrama esquemático por módulos y tareas del sistema. Este se muestra en la Figura 6.4.

Bajo este esquema de integración el sistema crecerá y podrá contar con un mantenimiento adecuado al estar las tareas vinculadas con los módulos correspondientes y así poder ser éstos sustituidos o alterados fácilmente.

6.2.2 Lenguajes de Programación

La construcción de la malla inicial, la implantación de los funcionales discretos y los métodos de optimización están escritos en **Fortran 77** probados con gran éxito en sistemas anteriores y muchos de ellos son piezas de fina ingeniería computacional en problemas de optimización y de generación de mallas.

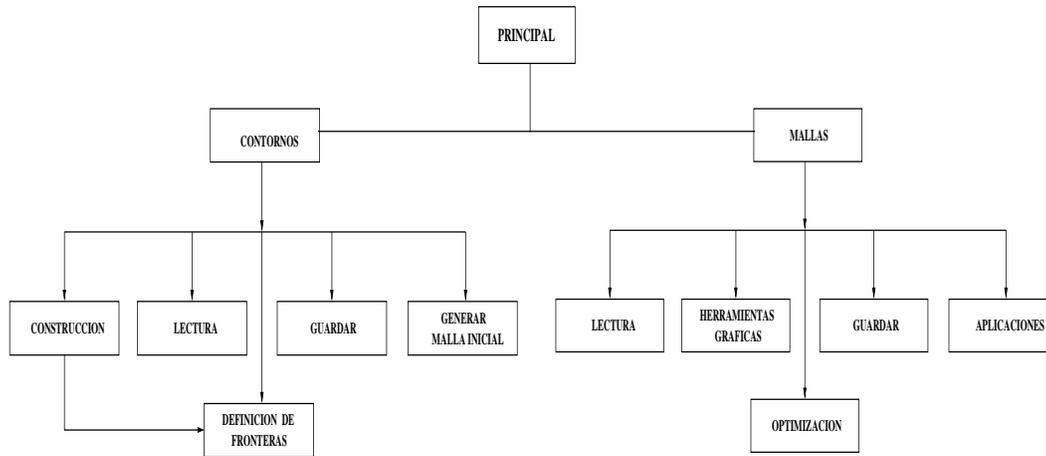


Figura 6.4: Diagrama esquemático de los módulos del sistema y las tareas que contempla.

La implantación de los funcionales discretos representa el poder manipular una malla y sus elementos: celdas y triángulos, de manera que sean perfectamente identificables así como el realizar cálculos de las medidas geométricas que cada uno de ellos representa. La implantación de estos esquemas de representación y los funcionales mismos se encuentra muy bien trabajados en **Fortran 77** y la idea es crecer en lo elaborado tomando lo usable. De esta manera se sugiere continuar con este plan de crecimiento, programando nuevos funcionales y esquemas de conteo de las propiedades sobre los elementos de la malla en lenguaje **Fortran 77**.

Los **drives** para el problema de optimización de gran escala han sido perfectamente probados, usados por la comunidad científica y muchos de ellos representan una joya de programación en **Fotran 77** por lo que se aconseja no hacer una traducción al lenguaje **C** u otro para su utilización sino lograr una interfase que los llame cuando sean requeridos y desde luego sea transparente al usuario.

Otro de los lenguajes que es de gran utilidad es el **C**. Con este lenguaje es posible apartar espacio de memoria para su posterior uso en la construcción y optimización de las mallas. Los sistemas **Unix** proveen de una herramienta muy eficaz para el manejo de la memoria a través del lenguaje **C**. De igual manera las instrucciones de las bibliotecas gráficas **Xlib** pueden ser empleadas indistintamente entre sistemas **Unix** desde el lenguaje **C ANSI**.

Por esta razón se seguiere trabajar con dos lenguajes de programación **Fortran 77** y **C ANSI** de manera que puedan interactuar con los **drives** programados y las bibliotecas de uso amplio. Para realizar la interfase de programación se sugiere que en lenguaje **C** sean programados el manejo de la memoria y las llamadas a las rutinas de **Fortran**. Así mismo se recomienda usar **C** para el intercambio de información entre

ficheros de disco (llamar y guardar).

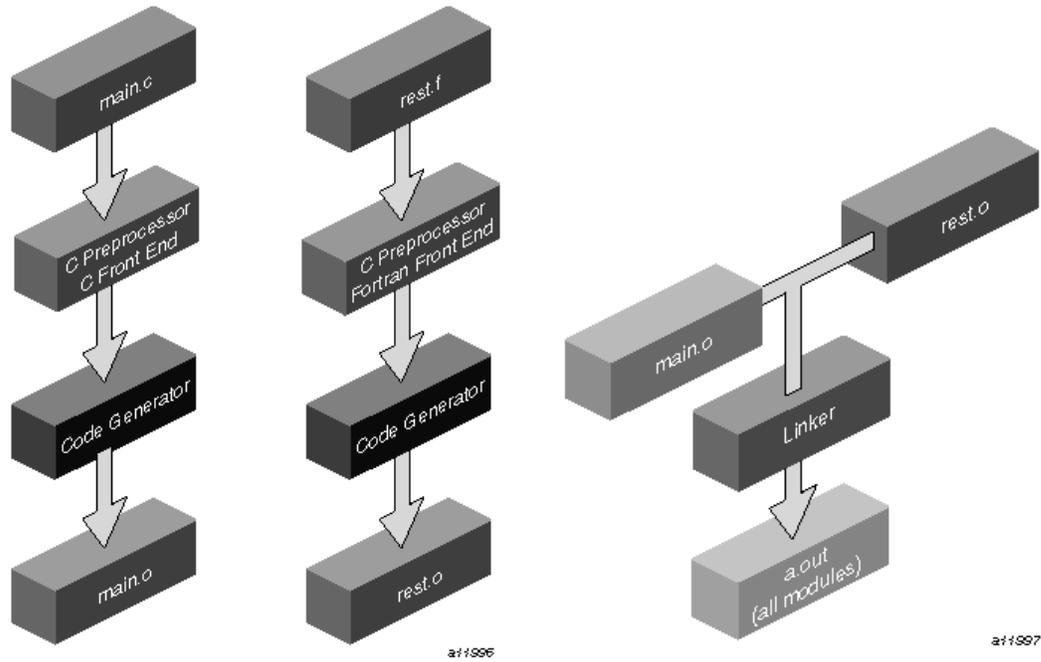


Figura 6.5: Esquema para compilar y ligar drives y programas de Fortran y C.

La integración de los módulos en un sistema seguirá el esquema de programación de los módulos y tareas particulares. En la Figura 6.5 se muestra un esquema adoptado para ligar drives y subrutinas en FORTRAN y C. El programa principal deberá estar escrito en lenguaje C para así contar con un mejor control de las tareas, el espacio de memoria a usar, los gráficos y el manejo de archivos de entrada y salida.

Capítulo 7

Conclusiones y trabajo a futuro

Durante el transcurso de la última década, una herramienta que ha sido fundamental en el desarrollo de investigación y experimentación con distintas ideas para lograr resolver de manera eficiente el problema de construir mallas suaves y convexas sobre regiones atípicas ha sido la visualización gráfica de la malla.

En los primeros años sólo contábamos con la evolución del proceso de optimización a través de los valores de la función así como el decrecimiento del número de celdas no convexas. Tratabamos de aprender sobre el efecto de los funcionales al imaginarnos su comportamiento en cada paso de optimización y sólo observando la malla óptima al final del proceso.

El contar con un sistema gráfico que nos permita observar y aprender del fenómeno que sobre la malla producen los funcionales ha sido una pieza clave para proponer otras formas de atacar el problema a través de funcionales muy robustos.

La experimentación llevada a cabo con los sistemas gráficos nos permitió proponer una regularización al funcional de suavidad discreto y entender cómo actúa sobre una celda. Esta idea dió pie a proponer una forma de extender dominios donde operar satisfactoriamente dicho funcional llevándonos al desarrollo de los llamados k -funcionales o funcionales adaptivos.

Una de las inquietudes del grupo de trabajo UNAMALLA es contar con un sistema profesional, versátil, con el que poder llevar a cabo una colección de experimentos, que pueda ser empleado en aplicaciones concretas y a la vez usarlo en cursos sobre solución numérica de ecuaciones diferenciales parciales. Por otra parte deseamos que el sistema cuente con una estructura que garantice su crecimiento y fácil mantenimiento.

Con los sistemas previos fuimos aprendiendo primero a desarrollar programas eficientes, que lleven a cabo los procesos de optimización y permitiendo entender la geometría detrás de cada método. Luego, fue nuestro interés interactuar con esa

colección de programas y módulos para permitirnos la fácil experimentación de las ideas sobre regiones singulares y de esta forma calibrar los funcionales y los métodos de optimización de gran escala.

Sin embargo, durante los primeros años estuvimos restringidos a la plataforma PC-DOS donde trabajamos con distintos tipos de compiladores y sus bibliotecas gráficas desarrollando algunas herramientas que no proveían estos compiladores y ganando experiencia sobre todo en el desarrollo de sistemas computacionales, planeando en cada uno mejorar la forma de interactuar con el usuario.

La forma de interactuar con el usuario permite que un sistema sea funcional para diversas aplicaciones, así como facilita su uso y experimentación. Los *Graphical User Interface* GUI nos proveen herramientas gráficas como botones y opciones de menús que al ser pulsados o elegidos las tareas asociadas son ejecutadas. Hoy en día el uso de los GUI juega una parte esencial dentro de cualquier sistema computacional profesional.

El sistema que se presenta hace uso de bibliotecas gráficas libres disponibles para ambientes Unix, así como de primitivas de graficación que hoy en día son un standard para ambientes gráficos. Las rutinas principales están programadas en lenguaje Fortran y el manejo de la memoria, los gráficos y la interacción con los módulos en lenguaje C. Este es un sistema modular multiplataforma para Unix.

7.1 Conclusiones

Con el presente trabajo se desarrolló un sistema que hace uso de los nuevos funcionales de área y suavidad que para la generación de mallas se cuenta actualmente, los llamados k -funcionales o funcionales adaptivos.

El resultado es un sistema independiente de la plataforma sea PC-DOS, Macintosh o de algún ambiente de programación como lo es Matlab, este último, cuya dependencia resultaría muy costoso para su posterior crecimiento.

Se describió a detalle una discretización muy interesante del funcional de Winslow que nos permite entender el efecto de la dimensión de la malla así como de la parametrización de la frontera de la región sobre el funcional.

Se especificaron los requerimientos y el diseño del sistema para la construcción de un contorno, la obtención de una malla inicial y una malla óptima. De tal fortuna que fueran definidos de forma puntual cada módulo, la interacción entre ellos y cada una de las tareas que integra.

El sistema se basó en bibliotecas gráficas libres. Se utilizó la biblioteca XForms para los GUI y se empleó primitivas de OpenGL; en lo particular se utilizó la biblioteca

Mesa para la graficación.

La estructura modular del sistema hace que su crecimiento y reuso para los próximos años se encuentre plenamente garantizado.

7.2 Trabajo a futuro

El sistema computacional aquí desarrollado es una pieza clave dentro del proyecto UNAMALLA. Nuestros objetivos a mediano y largo alcance son los siguientes:

- Elaborar un libro sobre la Generación Discreta de Mallas a través de los funcionales que hemos venido desarrollando. Un libro que pueda ser usado tanto como consulta dentro de la investigación en generación de mallas, y de igual forma ser usado en un curso sobre solución numérica de EDP's en el que se interactúe con el sistema.
- Implantar los nuevos funcionales α -convexos que se han desarrollado y que nos ha permitido robustecer la teoría de la generación discreta de mallas.
- Implantar al sistema un tratamiento de la frontera de forma tal que podamos suavizarla por curvas NURBS y controlar los puntos a fijar a través de una parametrización por splines racionales.
- Con este sistema podemos construir mallas estructuradas rectangulares. Nuestro interés es construir triangulaciones óptimas sobre regiones muy irregulares. El sistema permite crecer en ésta dirección al implantarle un módulo sobre triangulaciones a obtener a partir de la malla rectangular, así como considerar aquella obtenida por algún otro método y optimizarla a través de nuestros funcionales. Para lograrlo, necesitaremos que los archivos de salida puedan ser compatibles con los standares avalados por la ISGG (International Society of Grid Generation), el IGES y STL, para que entonces puedan ser usados los resultados en alguna aplicación de elemento finito.
- El sistema cuenta con un módulo para resolver algunas EDP's. Nuestro objetivo es implantarle algunos métodos por diferencias finitas novedosos para resolver sobre regiones atípicas algunas ecuaciones diferenciales parciales muy interesantes. De esta forma, el sistema podrá ser usado en algún curso de esta naturaleza y desde luego, en aplicaciones concretas.
- Nuestras mallas estructuradas son construidas a partir de puntos sobre la frontera fijos, y únicamente se busca la propiedad del funcional a lo largo de la región. Nuestro interés es incursionar en la generación de mallas adaptivas, mallas que sigan un patrón particular del fenómeno interesado a medir por la

aplicación deseada sobre la región, y desde luego, observen las propiedades de nuestros funcionales.

- Los funcionales discretos trabajan sobre triángulos, por lo que podemos definirlos de forma natural sobre superficies. Habiendo ganado experiencia con las primitivas de `OpenGL`, así como en el diseño de software y el intercambio de información con el usuario, en nuestros planes está elaborar un sistema para construir mallas en superficies.

Hemos de comentar que el sistema `UNAMALLA v. 2.0 for X Windows` es referencia dentro de tres importantes portales donde puede uno informarse sobre la investigación y desarrollo que actualmente se lleva a cabo dentro del área de la generación de mallas:

- Mesh Generation & Grid Generation on the Web.
Administrado por el Dr. Robert Schneiders, especialista en generación de mallas y su post/pre procesamiento.

`http://www.users.informatik.rwth-aachen.de/~roberts/meshgeneration.html`

- International Society of Grid Generation.

`http://www.isgg.org/soft_pub.html`

- Mesh Research Corner.
Administrado por el Dr. Steven J. Owen, jefe de un grupo de investigación en generación de mallas en Sandia National Laboratory.

`http://www.andrew.cmu.edu/user/sowen/mesh.html`

El sistema desarrollado se encuentra a la altura de los actuales sistemas para la generación de mallas disponibles tanto públicos como comerciales.

Para finalizar, hemos de comentar que la forma de tratar el problema de la generación de mallas por medio de funcionales discretos es novedosa entre los sistemas disponibles y aún quedan muchas ideas por llevar a cabo por nuestra parte.

Apéndice A

Manual Operativo del Sistema Unamalla v. 2.0 para X Windows

Basados en los *Requerimientos para un Sistema Automático Generador de Mallas* hemos desarrollado el Sistema Unamalla v. 2.0 para X Windows. A continuación describimos sus componentes, la filosofía detrás de su desarrollo y un modo de uso.

A.1 Diseño e Interface del Usuario

A.1.1 Sobre el diseño

El sistema UNAMALLA para X Windows es un paquete computacional, que resuelve de manera eficiente el problema de generar mallas rectangulares sobre regiones irregulares planas y acotadas. El tipo de mallas con que trabaja son las estructuradas y rectangulares. Este tipo de mallas es comúnmente utilizadas en la solución numérica de ecuaciones diferenciales parciales por el método de diferencias finitas.

El sistema es totalmente gráfico y hace uso de los GUI (Graphical User Interface) de la biblioteca XForms, por lo que es amigable al usuario y muy sencillo de usar. El sistema se ha diseñado para que sea automático bajo la idea de que con muy pocas opciones de tecla sea posible obtener una malla óptima sobre una región, siempre que esta exista. Para el despliegue gráfico se hace uso de primitivas de OpenGL por lo que el sistema es portable a casi cualquier plataforma Unix donde ambas bibliotecas puedan ser usadas. Los lenguajes de programación empleados son Fortran y C ANSI, el sistema puede estar disponible en casi cualquier sistema Unix excepto quizás por algunas consideraciones de bibliotecas.

Este sistema permite trabajar contornos poligonales de regiones planas y acotadas

para obtener sobre ellas una malla óptima, o bien a partir de una malla rectangular obtenida previamente por otros procedimientos y aplicar sobre ella las técnicas de optimización implantadas en el sistema. Algunos autores se refieren en términos generales a esta técnica como suavizamiento o post-procesamiento de la malla.

El sistema fue diseñado para el ambiente de trabajo de X Windows, y ha sido probado con éxito en los sistemas Linux RedHat para PC 6.2, 7.0 y 7.1, de igual manera en WorkStations SGI con sistema operativo IRIX 6.2 y 6.4 en los apartados últimos señalamos las necesidades técnicas del sistema. En la sección de instalación y ejecución indicamos los pasos a seguir para ejecutar el sistema. La forma de trabajo con el sistema es a través de dos ventanas, una de despliegue gráfico Display Window y la otra con menús de opciones y teclas de acceso rápido Control Panel y, de acuerdo al desarrollo del trabajo, con ventanas adicionales y opciones de elección que por botones integrados se van presentando.

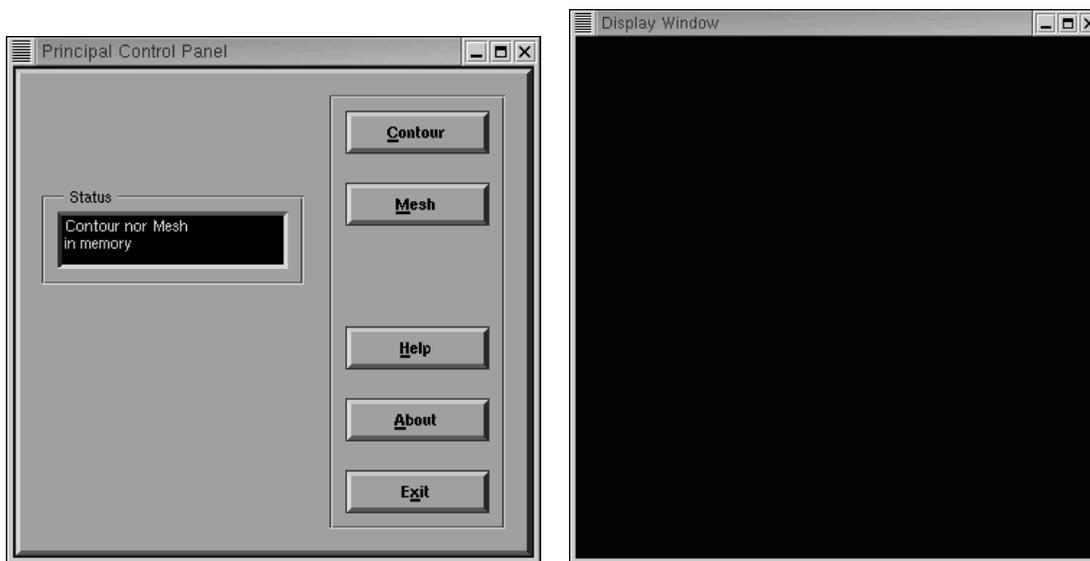


Figura A.1: Ventanas iniciales del sistema.

Cabe señalar que las opciones del sistema están escritas en inglés por lo que nos refiremos a las opciones que ha de considerar el usuario tal y como aparecen en pantalla e indicando las palabras clave en tipografía de máquina de escribir.

El sistema consta de cuatro menús de aplicaciones: **Main**, **Contour**, **Mesh**, **Information** y **Help**, el uso de cada menú y sus opciones o submenús estarán disponibles siempre que el proceso realizado lo permita. Por ejemplo, si se cuenta con una malla convexa en memoria podremos operar con todas las opciones de **Mesh**, pero si la malla no es convexa no estará disponible la opción de submenú correspon-

diente a las aplicaciones: **Applications**. En las secciones siguientes haremos una descripción de los módulos **Contour** y **Mesh**, comentaremos cada una de las opciones del menú, cómo y cuándo es posible interactuar con cada una de ellas.

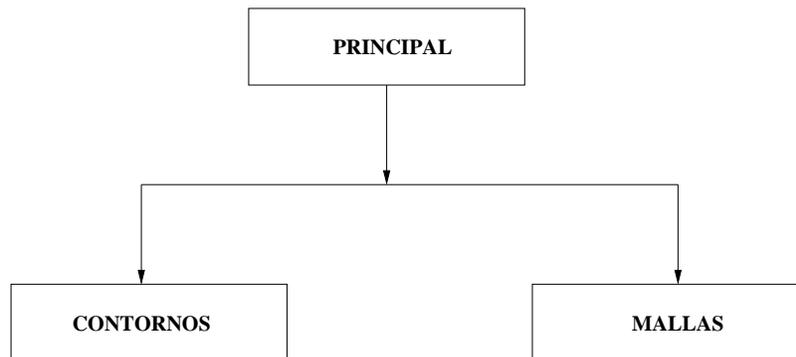


Figura A.2: Diagrama de flujo de los procesos del sistema.

A.1.2 Interfase para el Usuario

La interfase con el usuario consiste en un panel principal y algunos sub-páneos que serán desplegados en sustitución al primero dependiendo de la opción elegida. Los páneos consisten en un conjunto de botones, campos de entrada, controles deslizadores, cajas de diálogo así como buscadores y cajas de mensajes.

De acuerdo al diseño del sistema, el panel de control principal va cediendo a otros páneos de control de acuerdo al trabajo que se desea realizar; de igual manera en la ventana de despliegue gráfico observamos la acción con la malla o el contorno a realizar. La idea de diseño del sistema para los sistemas **X Windows** contempla la posibilidad de manipular la ventana de despliegue y de control de manera independiente para así facilitar el trabajo del despliegue gráfico, los páneos de control son activados para su uso posterior pulsando el área que los describe.

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

Debe tenerse en cuenta que se trabaja sobre el sistema de despliegue de ventanas **X Windows**, por el uso de la interfase y su presentación se limita a la forma en que las ventanas son desplegadas y de qué manera éstas están activas así como el diseño en color y forma de las mismas. Los páneos de control pueden ser activados al pulsar uno de los botones del mouse sobre la ventana.

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

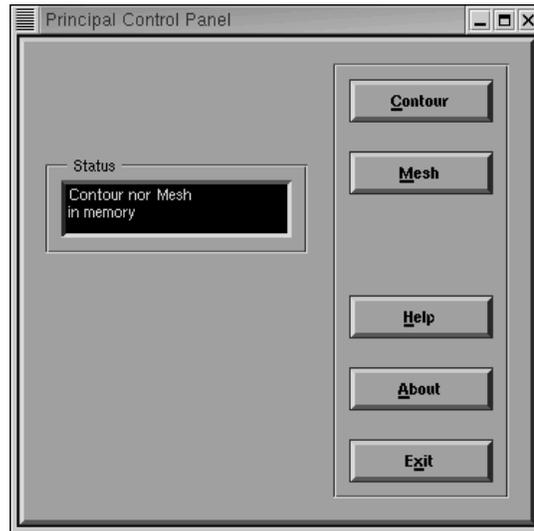


Figura A.3: Panel de control principal del sistema.

- Para pulsar un botón u opción del panel, mueva el cursor sobre el botón y pulse uno de los botones del mouse. Puede hacer uso de la secuencia de teclas **ALT+<key>** (donde **key** es la letra señalada en el botón) para ejecutar la misma acción.
- Para cambiar el valor de un campo de entrada, coloque el cursor sobre del campo y pulse alguno de los botones del mouse, o bien haga uso de la tecla **TAB**, tabulador, para obtener la misma operación, e introduzca el valor deseado.
- Para cambiar la posición de un control deslizable, tome el control al posicionar el mouse sobre él y oprima con el botón izquierdo del mouse sin soltar mientras “barre” el control de un lado a otro. Puede usar los indicadores de inicio y final del control para mover el control de paso en paso.
- Para elegir un elemento dentro de un buscador, un **browser**, posicione el indicador del mouse sobre el objeto dentro del browser y pulse el botón izquierdo. El color de fondo indica que el objeto se encuentra señalado y su valor se podrá observar en el campo de entrada del **browser** asignado para tal efecto.
- Para cambiar un **pop-menu** o menú de opciones, posicíonese con el mouse sobre el botón de la flecha indicadora y pulse uno de los botones del mouse, al hacerlo será desplegado una pequeña venta de opciones en la que podrá elegir con el mouse al pulsar con el botón izquierdo la opción deseada. Al hacerlo aparecerá en el letrero del menú el título de la opción elegida.
- Cuando el indicador del mouse se encuentra sobre un botón se desplegará una caja de títulos **tip text** desplegando un mensaje relativo a la opción.

La ventana del despliegue gráfico puede ser modificada en tamaño. Cuando se trabaja sobre la opción de despliegue de los resultados del módulo EDP puede usar el botón derecho del mouse para obtener algunas propiedades del despliegue de la solución, así como usar las flechas del teclado para rotar el objeto.

El diseño de UNAMALLA para los sistemas X Windows contempla la posibilidad de manipular la ventana del despliegue y de control de manera independiente para con esto facilitar el trabajo del despliegue gráfico; los paneles de control son activados para su uso posterior pulsando los botones en la caja que los describe.

A.2 Módulos principales

El sistema es modular. Esto es, integra una colección de tareas que permiten interactuar entre ellas de manera que el usuario pueda experimentar con los elementos del sistema: la construcción del contorno, la generación de mallas iniciales, la optimización de las mallas por métodos discretos.

La estructura de la programación y dependencia de los módulos la describimos esquemáticamente en la Figura A.4.

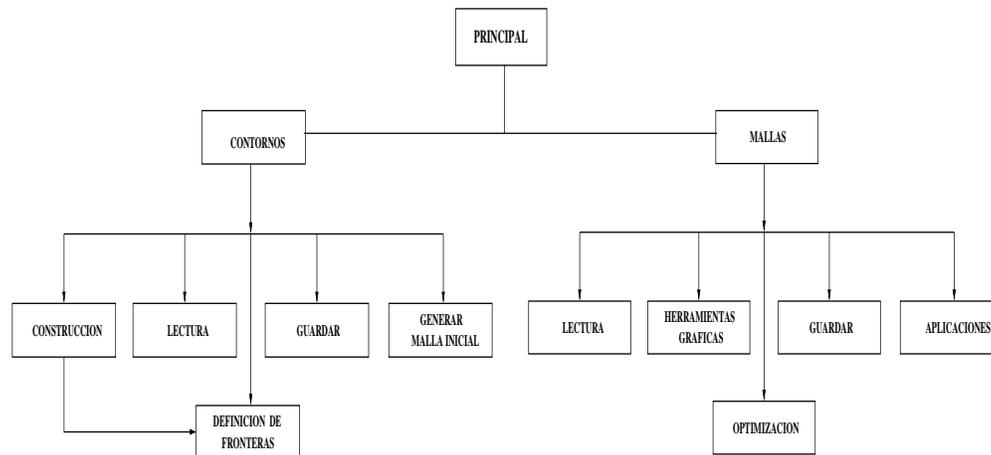


Figura A.4: Diagrama esquemático de los módulos del sistema

Como se describe en este diagrama esquemático de los módulos del sistema, las operaciones que pueden realizarse con el sistema son dos: trabajo con contornos y con mallas. Uno puede partir desde la información previa obtenida de alguna forma del contorno de la región, operar con la distribución de los segmentos de frontera y generar una malla por interpolación. De igual manera puede considerar la información de una malla, en archivo o en memoria y sobre ella operar los funcionales discretos, experimentar con el módulo de optimización y hacer aplicaciones sobre

mallas convexas: refinamiento uniforme y solución numérica de algunas EDP.

En los apartados siguientes describimos en rigor los módulos principales del sistema y cómo interactuar entre ellos.

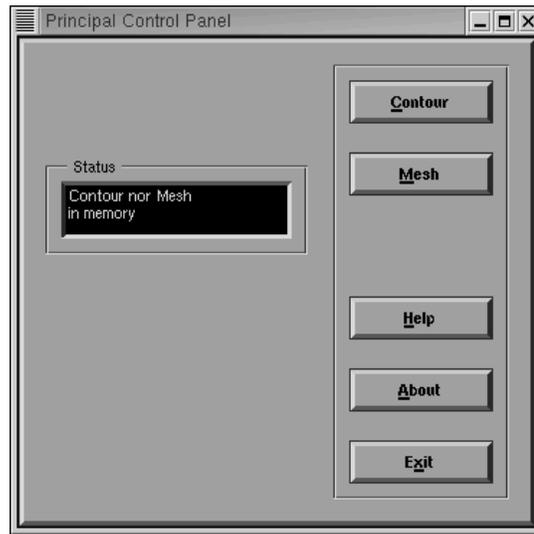


Figura A.5: Panel de control principal del sistema.

A.3 Módulo de Contornos

El conjunto de operaciones que podemos realizar con un contorno son: obtener la información del contorno a partir de un archivo de datos (con el formato convenido), construir un contorno poligonal en pantalla, editar los segmentos de frontera del mapeo, guardar el contorno en uso y generar una malla inicial por interpolación TFI con el contorno en memoria. Estas tareas están disponibles en el panel de control del módulo.

El sistema sobre el módulo de contornos contempla que en cada una de la opciones antes señaladas se cuente en pantalla con un panel de control apropiado para cada una. En la misma ventana del despliegue gráfico se visualizará y en su caso se hará la captura del contorno. Los paneles de control sustituyen al principal o a la última en cada una de las operaciones en uso.

Al solicitar la opción Contour en el panel de control principal aparecerá en pantalla un panel con las opciones descritas anteriormente

Las opciones de Create y Load son independientes de que exista o no algún contorno en memoria de ahí que éstas opciones estén siempre disponibles.

Desarrollemos cada una de las opciones disponibles del módulo Contour

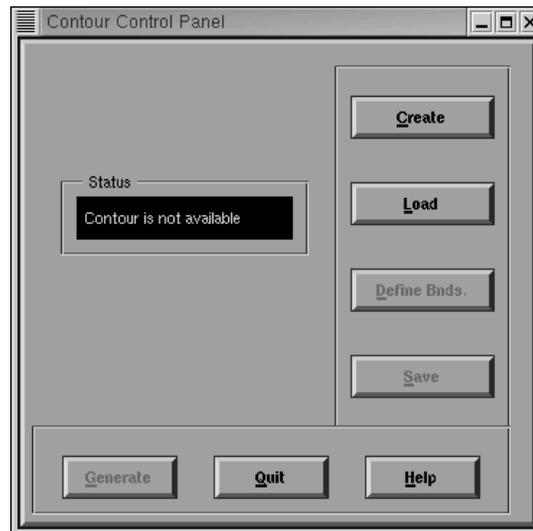


Figura A.6: Panel del Control de la opción Contour.

A.3.1 Construir un contorno en pantalla. Create

Bajo esta opción el usuario puede construir un contorno poligonal para ser usado durante el desarrollo de su trabajo en el sistema. Esta opción abre una ventana de control, en lugar de la anterior, desde la cual y siguiendo las intrucciones podrá aceptar el contorno generado así como eliminar los puntos no requeridos durante la construcción. En la Figura A.7 podemos observar el panel de control correspondiente a la opción Create.

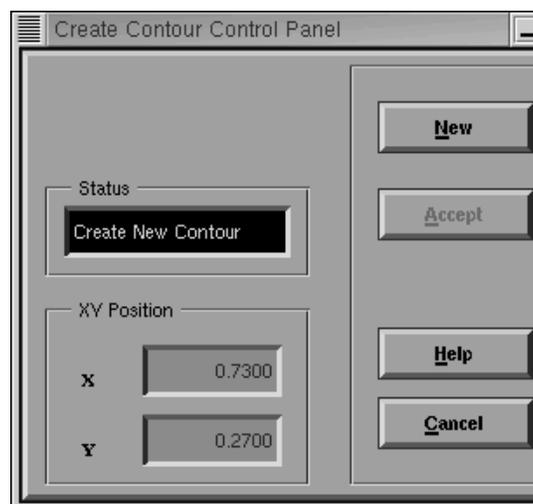


Figura A.7: Panel de control de la opción Create.

Como se observa, la ventana de despliegue gráfico cambia para mostrar una

cuadrícula definida en $[-1, 1] \times [-1, 1]$ en la que se podrá construir el contorno deseado sobre la ventana gráfica.

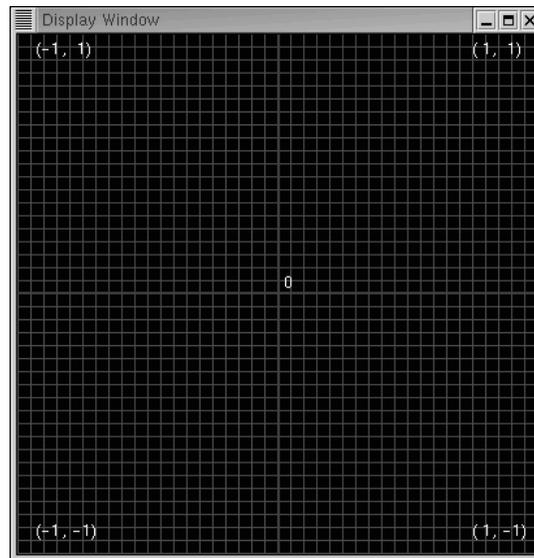


Figura A.8: Ventana de despliegue gráfico para la captura de contornos.

El trabajo de la captura de contorno es a través de los botones izquierdo del mouse para considerar el punto de pantalla como punto del polígono y el botón derecho se usará para cerrar el contorno. El botón medio del mouse se ha asignado para eliminar el punto último capturado del contorno y removerlo así de la lista de puntos que constituyen el contorno cerrado. Mientras no esté cerrado el contorno no podrá ser considerado para su uso.

Describamos cada una de las opciones integradas en éste panel.

- New** Volver a iniciar la captura de un contorno. Con esta opción se eliminará de la memoria cualquier trabajo previo realizado en el área de trabajo de la ventana de despliegue. Así, será posible iniciar una nueva captura de datos.
- Acept** Una vez obtenido el contorno cerrado la opción finaliza la captura de los datos y nos regresa al módulo anterior. Al pulsar la opción se llevará a la memoria los datos del contorno cerrado definido y sobre el cual se trabajará de ahora en adelante. Esta opción activa **Define Bnds.** en módulo **Countour** y nos permitirá posteriormente definir los segmentos de frontera del contorno.
- Help** Es una pequeña ayuda en línea. Se activa el **browser Netscape** con la página **web** asignada a tal propósito.
- Cancel** Esta opción cancela todo trabajo hecho con este módulo y nos lleva al módulo anterior. Al salir no se conservarán los datos de un contorno previo en memoria ya que fueron borrados al optar por la opción **Create**.

Dentro del panel de control puede observarse una caja de diálogo. En ella se desplegará una serie de breves señalamientos sobre el estado que guardan las acciones llevadas a cabo, sea que no se haya cerrado el contorno, que haga falta puntos para ello, que exista un contorno en memoria o bien, que la ejecución haya sido satisfactoria y se cuente con el contorno en memoria. De igual manera se observan un par cajas donde es desplegada la posición del mouse dentro del área de trabajo, esto con la finalidad de tener una posición del mouse y del punto a elegir del contorno dentro del sistema coordinado.

A.3.2 Abrir un archivo de datos **Load**.

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

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

Cuando el archivo sea elegido los datos pasarán a memoria del sistema, es decir, será sobre el cual se trabajará y la región será graficada en la ventana de despliegue. Si el contorno cuenta con la definición de los segmentos de frontera del mapeo, serán

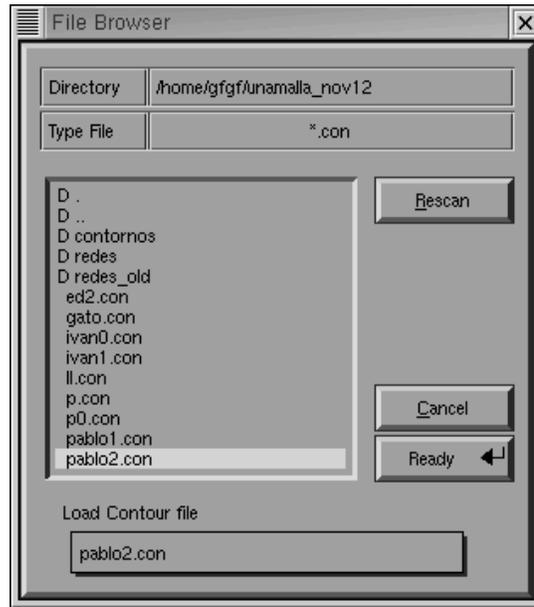


Figura A.9: browser para señalar el archivo de datos de un contorno.

éstas dibujadas en colores azul y amarillo dependiendo si representan los segmentos de línea verticales y horizontales del cuadrado unitario al que serán mapeados. En caso contrario el color azul bajo representará que no han sido definidos. En cualquier caso se observará un mensaje en la caja correspondiente al panel del contorno indicando el estado que presenta.

A.3.3 Definir los segmentos de frontera. Define Bnds.

Como se ha comentado, para la generación de una malla rectangular estructurada sobre una región plana, se requiere dejar bien claro los segmentos de frontera que serán mapeados a los segmentos del cuadrado unitario o viceversa.

Si contamos con un contorno en memoria es a través de ésta opción mediante la cual podemos o bien, definir los cuatro segmentos de frontera del contorno poligonal o bien modificar la elección previa. Para ésta operación se ha dispuesto que a través de controles deslizables y campos de entrada sean modificados o definidos los valores de los puntos que delimitan cada segmento.

Ahora bien, si contamos con un contorno en memoria, podemos pulsar la opción Define Bnds., no importando si se cuenta o no con la definición de segmentos de frontera del mapeo. Al dar acceso a la opción Define Bnds. el panel de control cambiará dando lugar a un panel que contenga opciones de entrada y controles deslizables. En la Figura A.11 se muestra dicho panel.

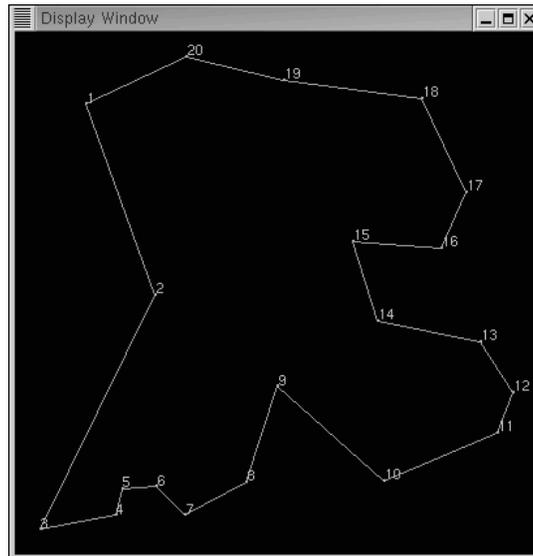


Figura A.10: Ventana de despliegue con el contorno que se encuentra en memoria.

El panel cuenta con las opciones de botón **A**utomatic, **A**cept, **H**elp, **C**ancel así como una caja de estado **S**tatus, una caja de entrada y de controles deslizables **s**liders.

En este módulo de asignación de los segmentos de frontera del mapeo o de edición de los mismos, podrá hacerse uno de una distribución de los segmentos de manera predeterminada o bien modificar la existente u obtenida por la opción **A**utomatic.

Opción Automatic.

Esta opción genera una distribución de los cuatro segmentos de frontera para mapeo del cuadrado unitario de manera que la longitud de los segmentos sea lo más cercano a ser iguales. Siempre y cuando la construcción sea buena para la generación de mallas convexas, es decir no se presenten picos hacia adentro en los vértices de unión de los segmentos de frontera [37]. Si la partición entregada por el sistema es aceptable, un mensaje correspondiente será desplegado en la caja de mensajes y la opción **A**cept será activada.

Opción Manual

Si se cuenta con una asignación previa de los segmentos de frontera del mapeo, podrá modificarse ésta para lograr una distribución de los mismos acorde al experimento o el deseo de construir la malla por el mapeo. Si no se cuenta con una asignación previa, puede emplearse la opción automática y a través de los controles deslizables

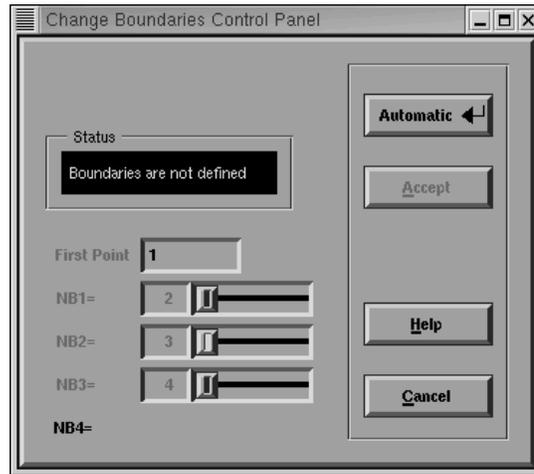


Figura A.11: Panel del control para la definición y edición de los segmentos de frontera del mapeo.

cambiar la distribución. La caja de entrada para el primer punto **First point** es muy importante ya que es un punto de referencia para fijar el inicio de la partición. Uno puede cambiar el dato contenido en ésta caja de entrada y aplicar la opción **Automatic** para luego y con la ayuda de los controles deslizables modificar la asignación entregada por ésta primera acción.

Los números **NB1**, **NB2**, **NB3** y **NB4** corresponden al número de puntos que incluye cada uno de los cuatro segmentos de frontera a partir del punto de referencia. Al modificar el punto de referencia **First point** el sistema renumera los puntos del contorno conservando el orden para de esta manera sea fácil de indentificar los segmentos de frontera. Cada uno de los **slider** involucrados cuenta con un color que identifica al tipo de segmento de frontera del mapeo, azul, amarillo, azul y amarillo que corresponden a los segmentos de “abajo”, “derecha”, “arriba” e “izquierda” respectivamente del cuadrado unitario teniendo por referencia al punto 1 y siguiendo la orientación positiva: el orden contrario a las manecillas del reloj.

En la caja de mensajes se despliega aquel correspondiente como resultado de la acción realizada: **Boundaries are not defined**, **Good definition**, **Bad definition** y dependiendo de dicha acción es posible considerar las asignaciones de los segmentos de frontera al encontrar activa la opción **A**cept.

Cuando la asignación de los segmentos ha sido satisfactoria, la opción **A**cept queda activa y entonces podemos considerar ya en memoria al contorno con las definiciones de los segmentos de frontera. El trabajo realizado con los segmentos puede ser anulado cuando pulsemos en éste panel la opción **C**ancel; en este caso continuaremos trabajando con la asignación previa, de existir ésta.

En cualquiera de los casos **A**cept o bién **C**ancel el panel de control relativo a la

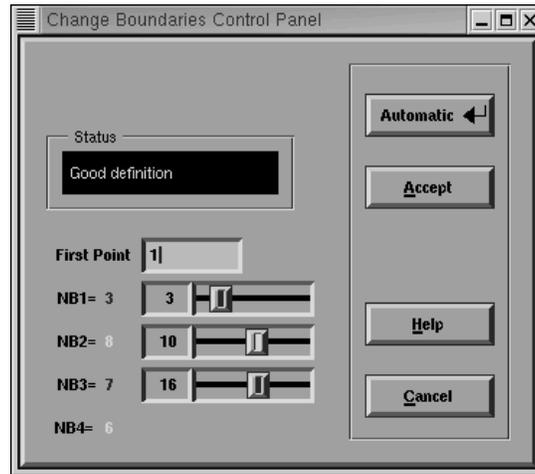


Figura A.12: Elección de los segmentos de frontera.

asignación de fronteras desaparecerá y volvemos al panel de control del contorno.

A.3.4 Guardar información Save

Contando con un contorno en memoria podemos en cualquier momento guardar su información para un posterior uso.

Aunque el contorno lo hayamos obtenido por archivo y luego modificado o no en la asignación de segmentos de contorno, su información puede ser almacenada en cualquier momento incluso cuando ya estemos trabajando con una malla en memoria podemos seguir contando con un contorno en memoria sea de la malla en curso o distinto a ésta. Con ello queremos enfatizar que la información de un contorno y de una malla son independientes. En la sección de las mallas se profundizan este punto para entenderlo a pleno.

Partiendo del módulo contorno, al pulsar la opción Save aparecerá en pantalla un **browser** (ver Figura A.13) bajo el cual podemos indicar la ruta por búsqueda de directorios donde se hospedará el archivo de datos. En la línea de entrada deberemos escribir el nombre del archivo que contendrá dichos datos. Si omitimos la extensión .con el sistema lo integra automáticamente.

El **browser**, al igual que en la opción Load, permanece en primer plano mientras que el panel de control del módulo para contornos se encontrará inactivo hasta que la operación haya sido concluida. La información del contorno en memoria es guardada en formato convenido. En la sección correspondiente a los formatos de archivo se detalla al respecto.

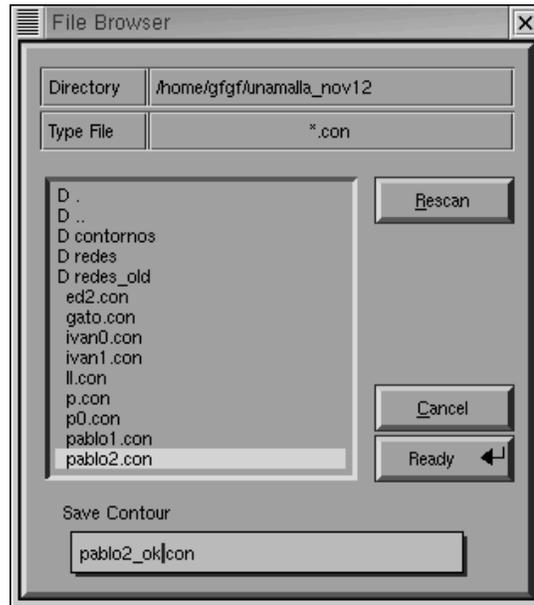


Figura A.13: browser para guardar información de un contorno.

A.3.5 Generar una malla inicial por TFI Generate

Como una opción adicional al módulo contornos se ha integrado la opción de generar una malla por interpolación TFI a partir de los datos del contorno y asignación de los cuatro segmentos de frontera. Esta opción no forma parte del módulo malla ya que se trabajará a partir de un contorno, de ahí la filosofía del diseño del sistema.

Teniendo un contorno con los cuatro segmentos de frontera asignados la opción Generate se encontrará activa y al elegirla el panel de control que le reemplaza tiene la presentación que se observa en la Figura A.14.

Con esta opción se podrá tener en memoria y en pantalla una malla generada por el método TFI. Para lograrlo basta indicar en los campos de entrada la dimensión de la malla por construir. La etiqueta `il` en el panel corresponde al número de líneas deseadas entre los segmentos de frontera de “arriba” y “abajo” indicadas por el color azul fuerte. La etiqueta `j1` corresponde al número de líneas deseadas entre los segmentos de frontera “izquierda” y “derecha” indicadas por el color amarillo.

De manera predeterminada el sistema contempla en los campos de entrada la opción de generar mallas de 20×20 . Los campos de entrada fueron diseñados para aceptar solamente números de tres cifras, con esto la malla más grande que se puede construir sobre la región es de 999×999 siempre y cuando se cuente con la memoria suficiente para lograrlo, en caso contrario aparecerá en la caja de mensajes el correspondiente a éste inconveniente.

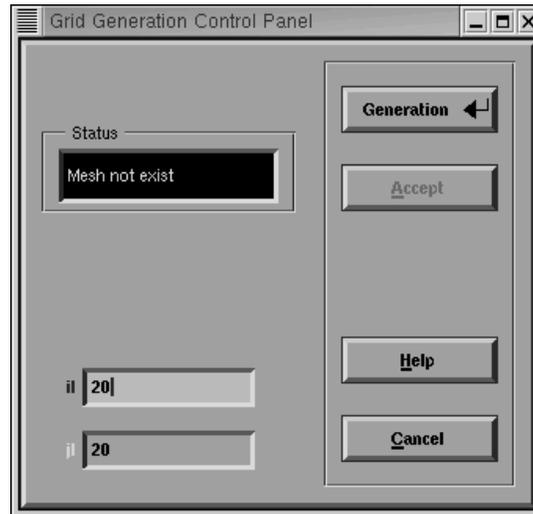


Figura A.14: Panel de control de la opción Generate.

Al pulsar la opción **Generation** si la construcción de la malla fue satisfactoria ésta aparecerá en pantalla sustituyendo en la ventana de despliegue gráfico **Display Window** la malla de la dimensión solicitada. De igual manera que en los señalamientos del contorno aquí podrá apreciarse los segmentos de frontera por los mismos colores que el caso del contorno y las líneas intermedias por un color azul débil. Si la malla presenta celdas no convexas éstas serán identificadas en pantalla por un color rojo y por disposición de despliegue de las mallas solamente serán señaladas las celdas problemáticas que representen menos del 20% del número total de celdas.

Si ya hemos generado la malla y no estamos convencidos del número de líneas en alguna de las fronteras podemos cambiar la dimensión editando los campos de entrada para esto dispuesto en el panel **Grid Generation Control Panel** y de nueva cuenta habrá que pulsar la opción **Generation**. Este proceso puede repetirse hasta quedar satisfecho con la dimensión de la malla. Al pulsar la opción **Accept** del panel de control la malla será dispuesta en memoria y será sobre la cual se trabajará. El panel de control se cierra y en su lugar aparecerá el panel de control principal donde podemos elegir trabajar con el menú malla.

La opción **Cancel** anulará el trabajo con la malla de interpolación o inicial y nos quedaremos con el contorno en memoria. Esta opción nos regresará al menú contorno.

Hemos de comentar que en el panel de control podrá observarse dentro de la caja de mensajes el correspondiente estado que guarda la acción última, sea que no se cuente con una malla **Mesh not exist**, sea la operación de la construcción fue un éxito: **Mesh defined** o bien tuvo un problema en la construcción al ser la malla no admisible: **Mesh not defined**.

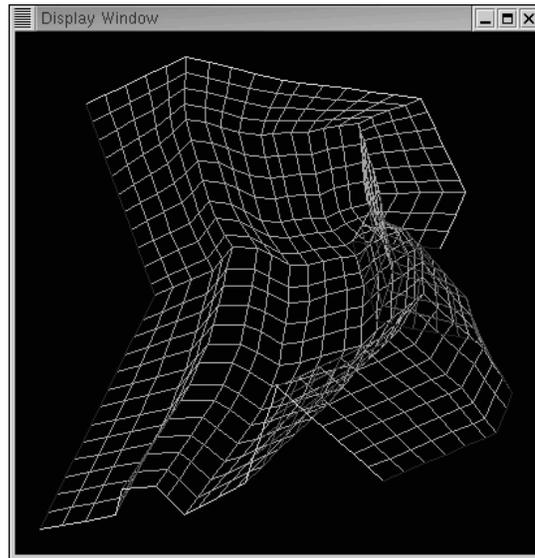


Figura A.15: Malla generada por TFI a partir del contorno.

Hasta aquí hemos descrito el módulo de contornos del sistema y cada una de las partes, ahora pasaremos a la descripción del módulo de mallas.

A.4 Módulo de Mallas

El módulo de mallas ofrece la posibilidad de trabajar con la malla en curso o bien llevar a memoria una que se encuentre en archivo con alguno de los formatos convenidos. Las tareas que podemos realizar con una malla son: llamar una malla desde un archivo, propiedades del despliegue de la malla, optimizar usando métodos discretos, guardar los resultados obtenidos y realizar alguna aplicación sobre mallas convexas.

Al pulsar la opción **M**esh del panel de control **Principal Control Panel** (ver Figura A.3) aparecerá en pantalla un panel de control conteniendo las opciones de tareas que podemos ejecutar con el sistema, ver Figura A.16.

Si contamos con una malla en memoria podremos ejecutar las opciones dispuestas para ella y si la malla es convexa podremos realizar algunas aplicaciones **A**pplication. En cada caso los botones relativos a las operaciones deberán estar activados. En la caja de mensajes se apreciará el relativo a si se cuenta una malla en memoria o no, **M**esh in memory y **N**o mesh in memory respectivamente. En el primer caso se desplegará la malla en la ventana de despliegue **D**isplay Window y en el segundo ésta estará vacía.

Describamos las operaciones que podemos obtener con éste módulo.

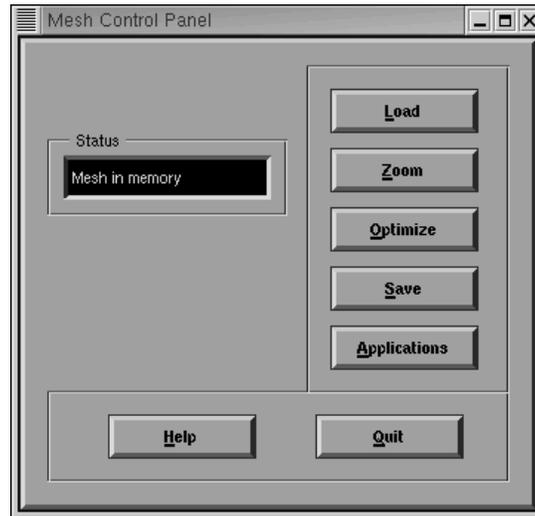


Figura A.16: Panel de control del módulo mesh.

A.4.1 Llamar desde un archivo a una malla Load

Contando o no con una malla en memoria ésta opción siempre estará disponible. Con esto, podemos llevar a la memoria una malla en archivo, bajo uno de los formatos convenidos.

Al pulsar la opción Load aparecerá en pantalla y en primer plano un **browser** desde el cual podremos indicar el archivo a leer de entre la lista que aparecerá en el espacio destinado a tal efecto. De igual manera podremos movernos por el sistema de archivos hasta localizar el archivo de datos. En la caja de texto **Filename** aparecerá el nombre del archivo a elegir.

Existen dos formatos que son manejados en el sistema, el ya clásico formato **XY** y el reciente formato **RED**. El **browser** cuenta con una opción de botón que nos informa sobre el formato de archivo de mallas esperado, pulsando el botón cambiaremos la modalidad y el sistema esperará entonces que el archivo cuente con el formato señalado.

Los archivos de datos cuentan con una estructura muy simple y se encuentran en formato **ASCII**. El sistema no cuenta con un análisis de datos de los archivos previo a su lectura, lo único que verifica es la dimensión de la malla y pasa a leer los datos tipeados para esa dimensión. Si existen problemas en la lectura de archivos, que no se pueda abrir el documento o que éste no se encuentre en la caja de mensajes aparecerá el que corresponda y la operación será cancelada.

Al leer satisfactoriamente los datos en la ventana **Display Window** se dibujará la malla indicando en colores azul y amarillo los segmentos de frontera “abajo”, “arriba”

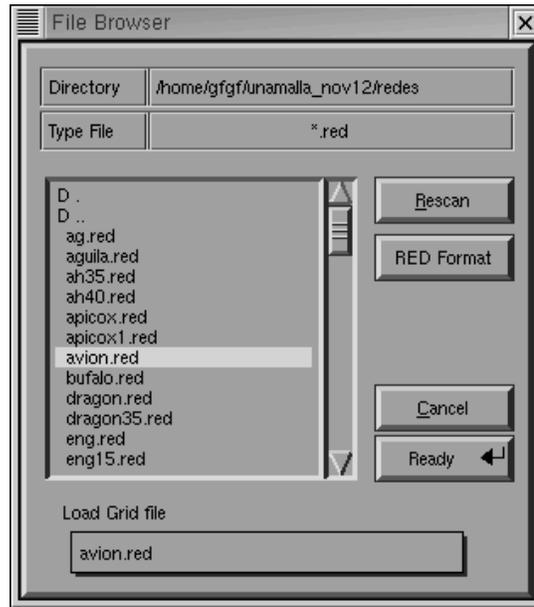


Figura A.17: browser para la localización de un archivo de malla.

e “izquierda” y “derecha” respectivamente. De la misma manera se podrá observar en pantalla las celdas no convexas o problemáticas y las líneas curvilíneas en color azul bajo.

Si existiera un contorno en memoria con ésta opción será eliminado. Sin embargo, a través un pequeño mecanismo de eliminación de puntos, consultar [7], contaremos en memoria con el contorno de la región de la malla así leída. Esta operación es transparente al usuario.

A.4.2 Herramientas de visualización de la malla Zoom

El sistema ofrece una herramienta para visualizar detalles de la malla. Esta herramienta permite ver la malla en su totalidad y en pequeñas partes zoom tanto la rejilla como los puntos de la malla y el contorno de la región.

Al pulsar la opción Zoom aparecerá en pantalla el panel de control Zoom Control Panel el cual tiene disponible sliders, botones agrupados y botones de reinicio, salida y la ayuda en línea (ver Figura A.19). Describamos cada una de sus partes.

Zoom Parameters Aquí se agrupan tres controles deslizadores para el manejo de la caja de visualización a lo largo de la malla: X y Y y un parámetro de profundidad zoom. A través de esos sliders podemos cambiar la profundidad de la visión de la malla y cambiar la caja de visualización para de ésta manera tener en pantalla la sección de la malla que estemos interesados en observar a detalle.

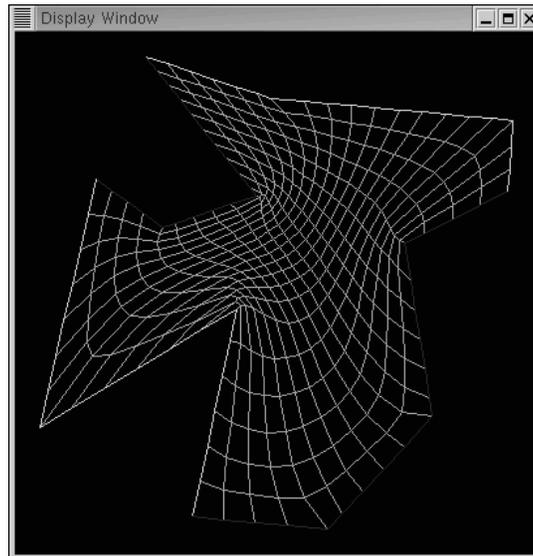


Figura A.18: Despliegue de la malla tomada de un archivo de datos.

Shape En esta sección del panel de control se agrupan tres modalidades de despliegue de los datos: la malla en su conjunto **G**rid, los nodos de la malla **G**rid **P**oints y únicamente el contorno **C**ontour. Esta agrupación de botones iluminados hace que solamente una de las opciones sea empleada en todo momento.

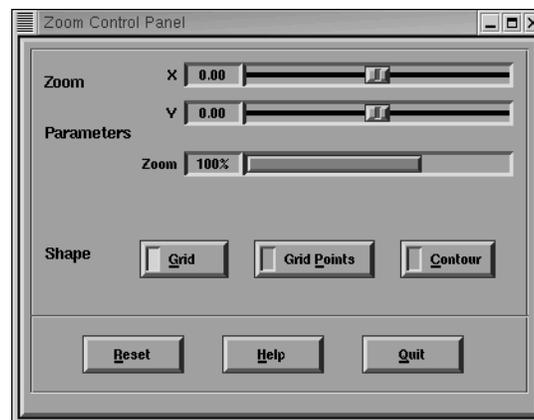


Figura A.19: Panel de control de las herramientas de visualización.

- Grid** Esta opción permite desplegar la malla en memoria por composición de celdas. Es el despliegue usual que el sistema contempla para la malla. La composición del despliegue gráfico viene dado por el trazo del contorno diferenciando los segmentos opuestos de frontera por colores azul y amarillo y las segmentos curvilíneos en azul bajo. Si la malla contiene celdas no convexas éstas son dibujadas en color rojo. El sistema contempla un número máximo como tope para graficar las celdas no convexas, éste número fue calculado esperando que a lo más el 20% de celdas de la malla sean no convexas, si el número total es mayor, el sistema únicamente señala las primeras.
- Grid Points** Otra de las herramientas para el despliegue gráfico de la malla lo representa el despliegue de los nodos de la red. Esta opción describe en forma semejante la anterior, el contorno de la región, sólo que los nodos de la red o puntos de intersección de los segmentos curvilíneos son dibujados ahora con un pequeño círculo de color azul bajo (ver Figura A.20).
- Contour** Esta opción únicamente despliega en colores azul y amarillo, los segmentos de frontera opuestos dos a dos que conforman la malla. No describe información alguna del orden de los segmentos. Los segmentos en color azul representan los segmentos “abajo” y “arriba” y el color amarillo a los segmentos “izquierda” y “derecha”

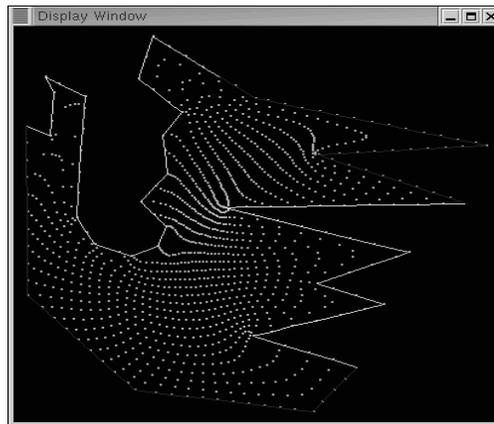


Figura A.20: Despliegue de los nodos interiores de la malla.

Las opciones agrupadas **Zoom parameters** y **Shape** son independientes. Con esto, queremos decir que moviendo la caja de visualización y el parámetro de profundidad podemos observar la malla en su conjunto como líneas coordenadas o bien, si es el caso, los nodos interiores o el contorno de la región.

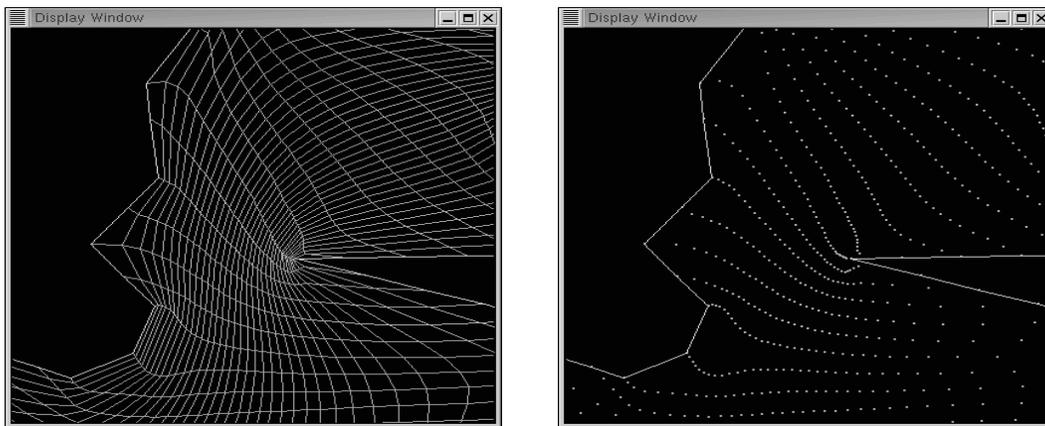


Figura A.21: Distintos despliegues de la malla en un acercamiento.

A.4.3 Optimización de malla Optimize

Esta opción del módulo Mesh nos permite trabajar con los funcionales discretos, aquí implantados para obtener una malla óptima.

La malla de trabajo es la que se encuentra en memoria obtenida por TFI o bien, desde un archivo de datos. Al pulsar la opción Optimize aparecerá en pantalla un panel de control conteniendo opciones pop-menu, un control deslizable slider, opciones radio, botones agrupados y no agrupados según se muestra en la Figura A.22.

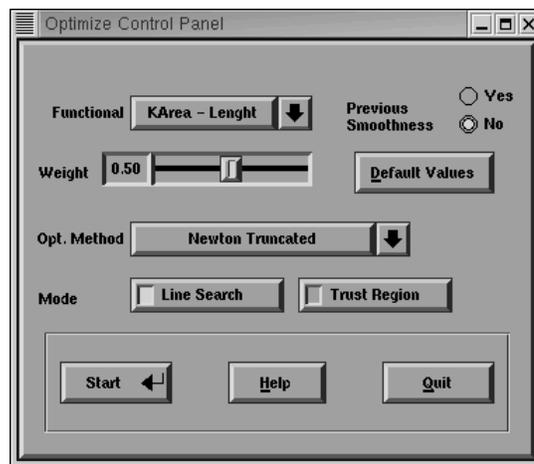


Figura A.22: Módulo de optimización Optimize

Partamos de un ejemplo concreto, la malla `cuad2pp.red`, y describamos cada una de las opciones y cómo actúa éste módulo sobre la malla en memoria para obtener una malla óptima. La malla `cuad2pp.red` se obtuvo alterando parámetros de los métodos obteniendo una malla bastante no convexa sobre una región de trabajo muy simple, el cuadrado unitario pero con una distribución de los segmentos de frontera muy interesante. Ver Figura A.23.

En el panel de control de optimización Optimize Control Panel, elegimos el tipo de funcional a trabajar: opción Functional, el peso de la combinación convexa deseada: slider Weight; así como el método de optimización a usar: opción Opt. Method y la modalidad de optimización, opción Mode. Todas esas opciones las describiremos a continuación. De igual manera se cuenta con una opción de botón Default values que permite retornar a las opciones que por defecto se contempla en el sistema y son las que consideramos al entrar a éste módulo.

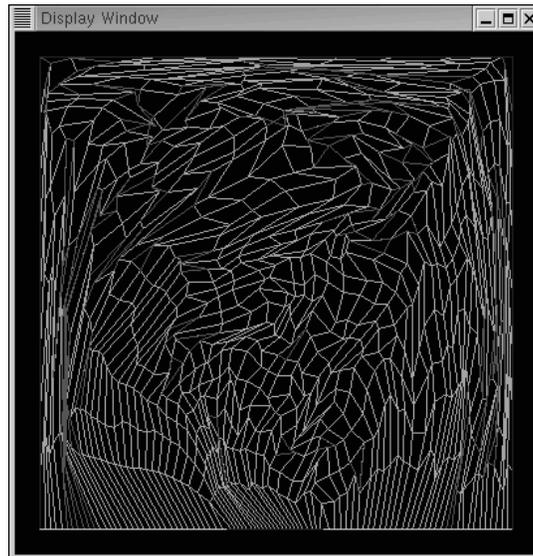


Figura A.23: Malla de trabajo `cuad3pp.red`

Elección del funcional y del peso

Para la elección del funcional de trabajo se cuenta con una opción `pop-menu` en la que se encuentra una lista de los funcionales implantados en el sistema:

Suavidad adaptivo – Área	<code>kSmooth-Area</code>
Área adaptivo – Longitud	<code>kArea-Lenght</code>
Área – Longitud	<code>Area-Lenght</code>
Área – Ortogonalidad	<code>Area-Ortho.</code>

Al pulsar la flecha indicadora hacia abajo del `pop-menu` se desplegará una pequeña caja donde se podrá elegir con ayuda del mouse cualesquiera de los funcionales antes señalados. Al hacerlo en el título del control aparecerá el relativo al funcional que se trabajará.

Una vez elegido el funcional, o combinación de funcionales, se considerará el grado de combinación. Esto lo hacemos a través del control deslizador `slider` para tal efecto. Moviendo el `slider` observaremos que el peso, o grado de la combinación varía desde 0 hasta 1 con incrementos de .01 dependiendo de cómo deslicemos el `slider`. Excepto para el funcional de Área–Ortogonalidad, el valor de la combinación es independiente del funcional. Para éste caso específico del funcional la combinación es .5 y la opción del peso `Weight` se inhabilita. En uno de los extremos del control puede observarse (en una caja de texto) el valor elegido al mover el `slider`.

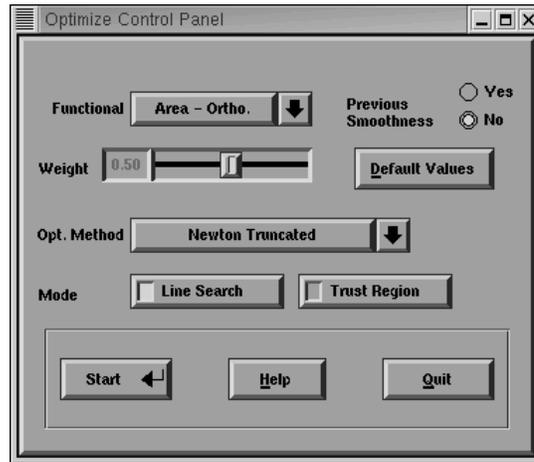


Figura A.24: Inhabilitación del slider al elegir la opción Area-Orto.

Elección del método de optimización y su modalidad.

Independientemente de la elección del funcional y su peso, podemos elegir el método operar o experimentar en la optimización.

En el sistema se han implantado tres métodos de optimización de gran escala: Métodos de Newton Truncado: **Newton Truncated**, Método de Memoria Limitada L-BFGS de Nocedal: **Limited Memory L-BFGS** y una opción experimental con la cual se han obtenido buenos resultados para el Método de Newton puntual: **Newton Point to Point**.

Cada una de las anteriores opciones son elegidas a través del **pop-menu Opt. Method**. Para cada método se ha implantado una modalidad. Por ejemplo en el caso del **Newton Truncado** podemos desear realizarlo con **Búsqueda en la Línea** o bien con **Región de Confianza**. Para el caso de **Memoria Limitada** se lleva a cabo la optimización almacenando 3 variables en cada paso del método o bien 5; y para el método de **Newton Puntual** resolver la ecuación de Newton por componentes o simultáneo, ver García [23].

Suavizamiento previo

Otra de las opciones contempladas es realizar un suavizamiento de la malla previo a su optimización **Previous Smoothness**. Este pre-procesamiento va en el sentido de desdoblar la malla un poco antes de emplear el funcional elegido y así reducir el tiempo y esfuerzo de obtención de una malla óptima. El usuario puede elegir con los controles **radio** para tal efecto si desea que dicho proceso sea realizado. En la parte superior del panel de control **Optimize Control Panel** puede elegir la opción

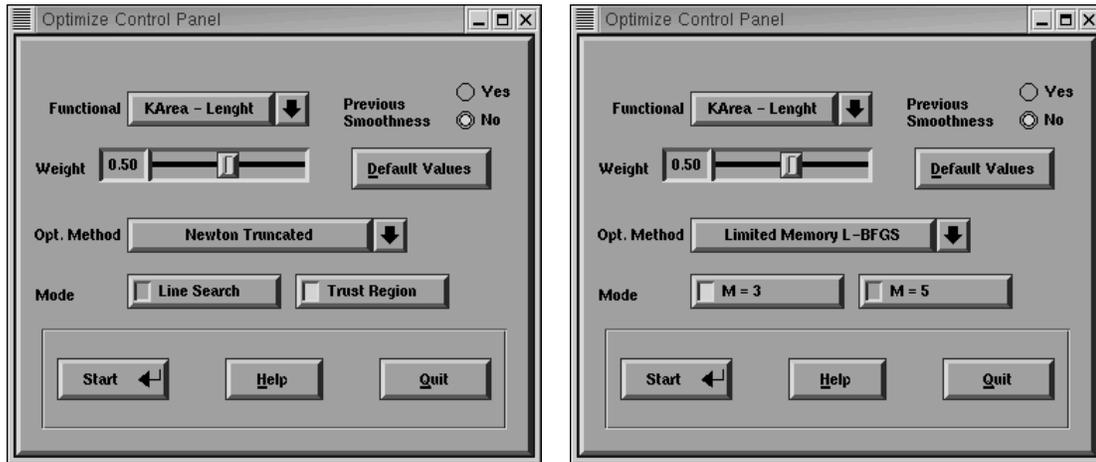


Figura A.25: Distintos métodos de optimización y el modo o parámetro para llevarlo a cabo.

No si no desea que se lleve a cabo o bien la opción Yes si desea que esto ocurra. Por defecto tenemos la disposición de que este procesamiento no sea solicitado. El suavizamiento de las líneas se obtiene usando del funcional de longitud con solamente algunas iteraciones y usando el método de Newton Truncado con Búsqueda en la Línea, este suavizamiento previo es transparente al usuario sólo se informa en el panel de despliegue del proceso de optimización las iteraciones y el tiempo transcurrido.

Valores predeterminados

En cualquier momento el usuario puede pulsar el botón `Default values` y los valores predeterminados se usarán para la optimización. Los valores predeterminados son: Funcional de k -Área-Longitud, con un peso de .5, haciendo uso del optimizador Newton Truncado con Región de Confianza y sin suavizamiento previo.

Una vez que hemos elegidos los parámetros para buscar la optimización, al pulsar el botón `Start` de `Optimize Control Panel` el proceso iniciará ocultando la ventana de control de la elección de la optimización y en su lugar presentará una ventana de control donde se desplegará la información relevante del proceso.

En dicha ventana se observan los datos seleccionados para la optimización en el costado izquierdo, en fondo negro y letras de color amarillo. La información que sale es la siguiente

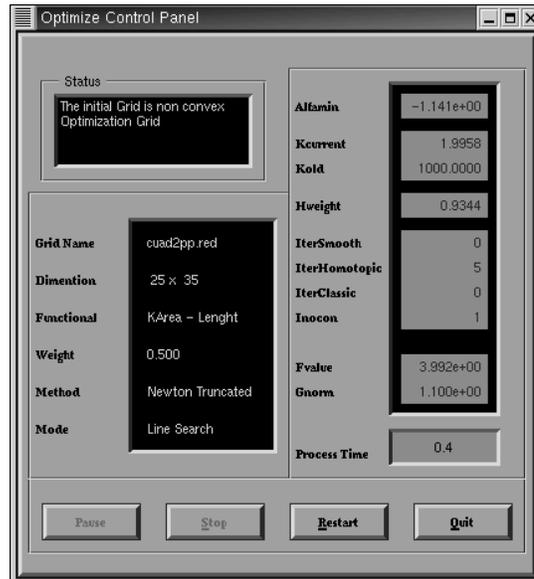


Figura A.26: Ventana de despliegue de información de la optimización.

Grid name	Nombre de la malla
Dimention	Dimensión de la malla
Functional	Funcional elegido
Weight	Peso de la combinación del funcional
Method	Método de optimización a gran escala
Mode	Modalidad del método de optimización.

Al costado derecho de la ventana se observa información de los datos que el sistema va desplegando en cada paso de la optimización. Algunos de estos valores son muy técnicos pero nos describen el desarrollo del proceso. La información que ahí se observa en la Tabla A.1.

En la parte superior, en la caja de mensajes **Status**, se ve el estado que guarda el proceso, si al inicio la malla es convexa o no, si el proceso de optimización fue satisfactorio, si éste fue detenido por el usuario o bien si hubo algún problema con la optimización y de qué tipo. Los mensajes usuales estan la Tabla A.2.

Esos mensajes nos aportan información de lo ocurrido en el proceso de optimización. Naturalmente los mensajes idóneos serían **Convex grid**, **Stop by FTOL condition** o bien **Convex grid**, **Stop by GTOL condition**, es decir, que el proceso se ha detenido satisfactoriamente cuando se ha logrado la condición de paro por diferencia relativa en los valores de la función, o bien por haber logrado la condición de paro por el gradiente.

El panel de control contempla tres botones importantes: **Pause/Continue**, **Stop** y **Restart**, ver Figura A.26. El botón **Pause** permite poner en pausa el proceso y

Alfamin	α_- de la malla actual
Kcurrent	$k/\bar{\alpha}$
Kold	$k_{\text{anterior}}/\bar{\alpha}$
IterSmooth	Número de iteraciones de suavizamiento previo
IterHomotopic	Número de iteraciones del paso de homotopía
IterClassic	Número de iteraciones del paso clásico de optimización
Inocon	Número de celdas no convexas de la malla actual
Fvalue	Valor del funcional
Gnorm	Valor de la norma del gradiente
Process Time	Tiempo transcurrido del proceso de optimización.

Tabla A.1: Información del proceso de optimización.

se continua al pulsar el mismo botón cuya leyenda ha cambiado a **Continue**. Esta herramienta permite poner en pausa el proceso a fin de detenernos a observar el cambio de los valores de la optimización y la malla misma. El botón **Stop** detiene el proceso. El tercer botón **Restart** permite reiniciar el proceso cuando éste haya concluido o bien cuando lo hayamos detenido. Esta herramienta es muy importante ya que dada las condiciones de paro para la optimización podríamos estar interesados en reiniciar el proceso para obtener algunas iteraciones más con la malla de salida con el objetivo de tener una “mejor” malla. Esta opción permanecerá inactiva en tanto no se concluya el proceso de optimización o bien no lo detengamos pulsando la opción **Stop**.

Concluimos esta opción de **Optimize** del módulo malla mostrando las ventanas correspondientes a la información de la malla cuando han transcurrido 5, 10, 15 y 25 iteraciones así como la óptima obtenida luego de optimizarla un poco más.

```

The initial grid is convex, optimizing
The initial grid is non convex, optimizing
Convex grid, the initial grid is good
Convex grid, the homotopic step is ok
Convex grid, STOP by FTOL condition
Convex grid, STOP by GTOL condition
Convex grid, STOP by MAXITER condition
Non convex grid, STOP by MAXITER condition
Convex grid, STOP by user
Non convex grid, STOP by user
The interval of uncertainty is at mos xtol
Convex grid, Rounding errors prevent furthet progress
Non convex grid, Rounding errors prevent further progress
Convex grid, the step is at the lower bound STPMIN
Non convex grid, the step is at the lower bound STPMIN
Convex grid, the step is at the upper bound STPMAX
Non convex grid, the step is at the upper bound STPMAX
Convex grid, the number calls of FCN has reached
Non convex grid, the number calls of FCN has reached

```

Tabla A.2: Mensajes de salida y ejecución en el proceso de optimización.

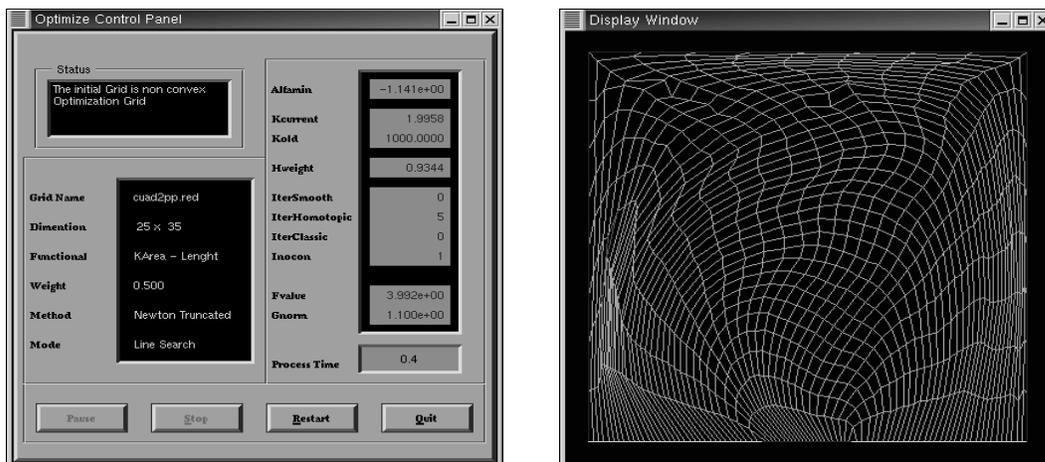


Figura A.27: Proceso luego de 5 iteraciones

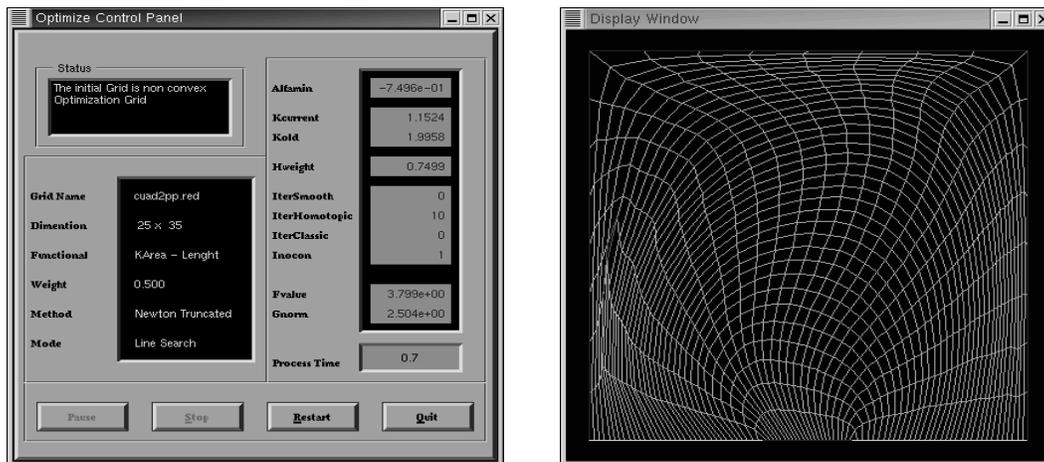
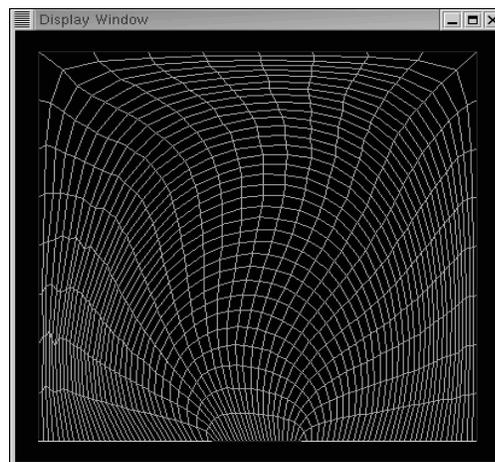
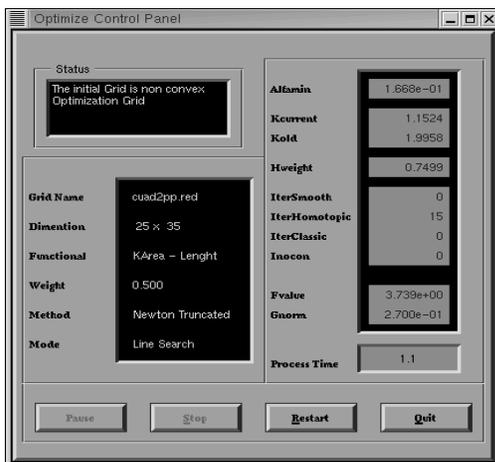
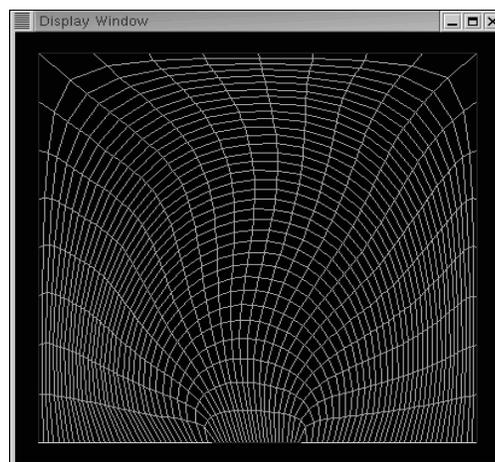
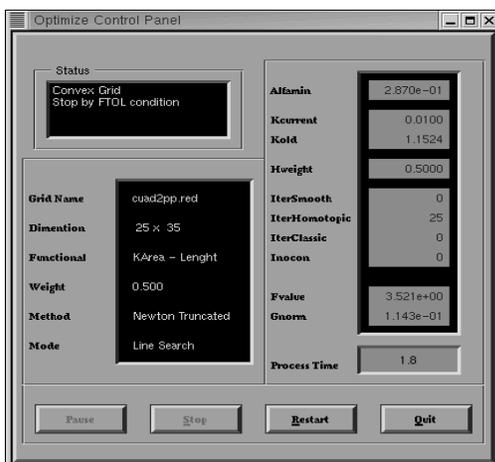


Figura A.28: Proceso luego de 10 iteraciones



Proceso luego de 15 iteraciones



Proceso luego de 25 iteraciones

Figura A.29: Ventanas durante la ejecución.

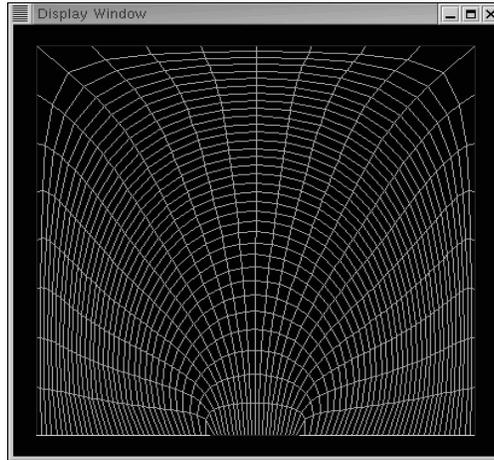


Figura A.30: Malla óptima luego de algunas interacciones

A.4.4 Guardar información Save

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

Al situarnos en el panel de control **Mesh Control Panel** y contar con una malla en memoria se guarda la información de esta malla en cualquier momento al pulsar la opción **S**ave. Al hacerlo aparecerá en primer plano de pantalla un **browser** con el cual podemos indicar el directorio donde se deposita la información así como elegir o indicar en el campo de entrada el nombre del archivo y el formato **XY** o **RED** donde los datos se guardan.

En la parte superior del **browser** se indica el directorio en uso y en el cuerpo del **browser** observaremos un listado de los archivos que ahí se encuentran. Al pulsar la opción **Ready** los datos se depositan en disco y se regresa el control de funciones a **Mesh Control Panel**. Pulsando la opción **Cancel** se anula la operación de guardar información y de igual forma regresamos el control a **Mesh Control Panel**.

A.4.5 Aplicaciones con mallas convexas Applications

Esta opción es exclusiva para mallas convexas. Contando con una malla en memoria podemos llevar a cabo operaciones de refinamiento uniforme y resolver algunas ecuaciones diferenciales elípticas sobre la región.

Este módulo de aplicaciones se ha integrado al sistema de manera que el usuario pueda experimentar o resolver algunas ecuaciones diferenciales y también pueda construir una malla más fina a partir de la obtenida.

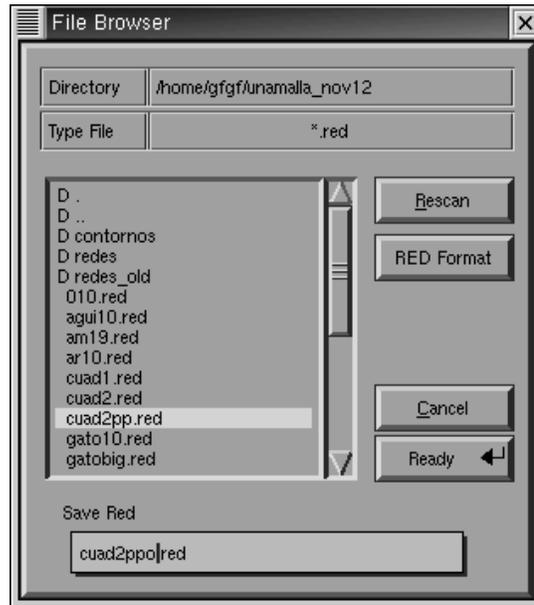


Figura A.31: browser para guardar la malla en curso.

Al pulsar la opción Applications del módulo de mallas aparecerá un panel de control Applications Control Panel donde hay dos opciones de botón Refinement y PDE solve así como las ya conocidas Help y Cancel.

Refinamiento uniforme Refinement

Una herramienta muy útil para algunas aplicaciones es contar con una malla más fina que la actual es cuando estamos interesados en hacer mediciones con mayor precisión o con mayor detenimiento en secciones o área particulares de la región. Muchos generadores de mallas o programas que refinan mallas lo hacen de manera automática siguiendo un patrón de comportamiento de la solución *adaptiva*. En otros el refinamiento es manual y específico sobre la sección de interés.

En el sistema hemos considerado que el refinamiento sea uniforme en toda la malla ya que nos interesa hacer mediciones globales y proveer al usuario diversas mallas refinadas que luego pueda optimizar y entonces utilizar en sus mediciones. La técnica que empleamos para lograr este refinamiento es a través de interpolación lineal, interpolación entre líneas.

Al pulsar la opción Refinement se deja paso a Refinement Control Panel en el cual se introduce el número de líneas que van a ser interpoladas entre dos segmentos de línea. El panel de control consta de dos counters con los que introducir por golpe de botón de mouse el número m de líneas a insertar entre las líneas curvilíneas que

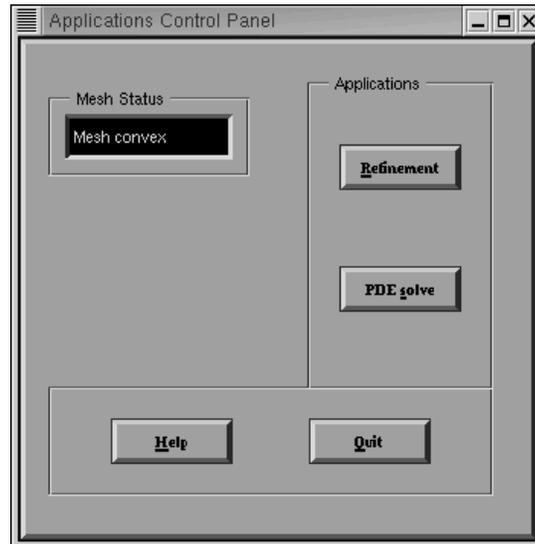


Figura A.32: Panel de control de las aplicaciones con una malla convexa.

unen los segmentos de frontera “azul” y el número n de líneas entre los segmentos curvilíneos que unen los segmentos de frontera “amarillo”. Ver Figura A.33.

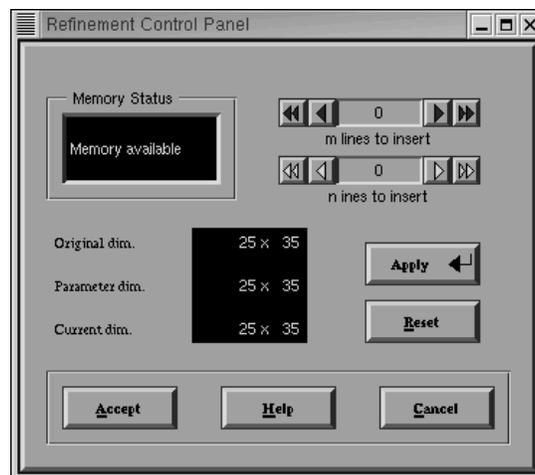


Figura A.33: Refinement Control Panel

El proceso consiste en insertar líneas entre las ya existentes. Al finalizar el proceso, o bien tendremos la malla con la nueva dimensión, o bien la original antes del proceso al cancelar éste.

Como se ha comentado con los contadores se indica el número de líneas a insertar (los puntos a interpolar entre líneas). En la caja de información **Original dim.** se observa la dimensión original de la malla antes del proceso de refinamiento. En la caja **Parameter dim.** se ve cuál es la nueva dimensión de la malla si pedimos que

ésta se construya (al pulsar el botón **Apply**). En la caja **Current dim.** se presenta la dimensión de la malla que se encuentra en pantalla.

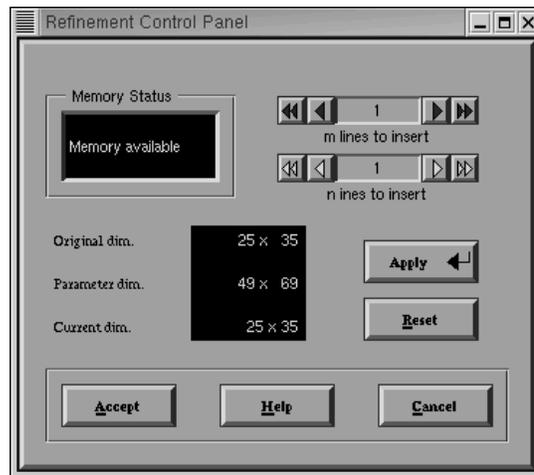


Figura A.34: Cambio de dimensión de la malla al refinarla

Al pulsar el botón **Apply** se llevará a cabo el refinamiento y obten en pantalla la malla resultante, ver Figura A.35

La opción **Reset** cancelará el refinamiento llevándonos a los parámetros originales y a la malla primera. Con el botón **Accept** se obtiene la malla resultante y con **Cancel** se anula toda operación hecha teniendo en pantalla la malla primera antes de éste proceso.

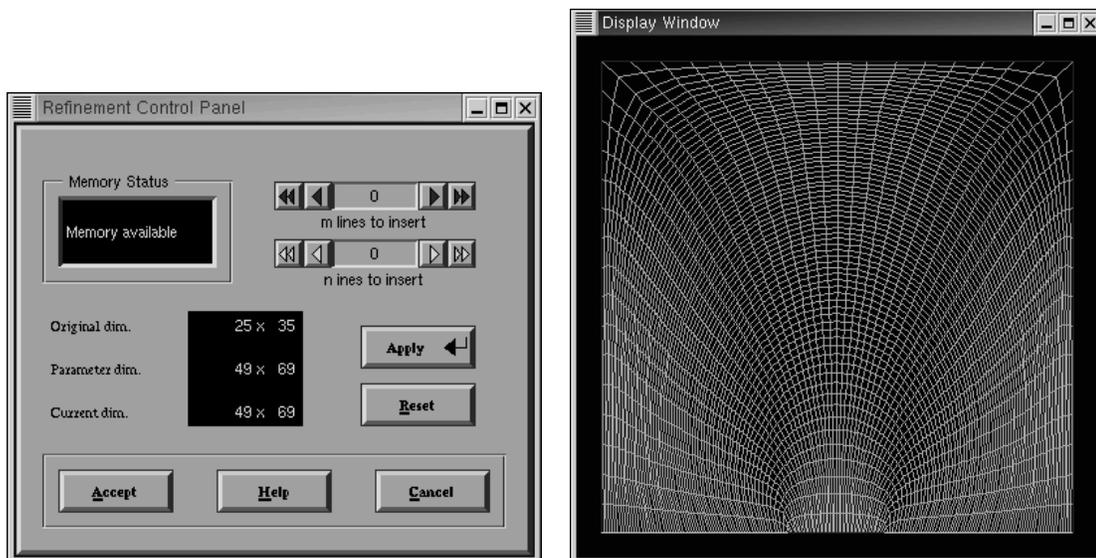


Figura A.35: Aplicación del refinamiento y malla resultante.

Solución numérica de algunas EDP PDE solve

Otra de las aplicaciones que se ha dispuesto en el sistema es la resolver algunas ecuaciones diferenciales parciales elípticas sobre la región.

Este módulo fue construido para experimentar las diferentes mallas óptimas del sistema obtenidas con los funcionales discretos. La idea es verificar qué tan adecuadas resultan nuestras mallas sobre un problema específico de EDP. Con esta aplicación el usuario puede experimentar con la malla convexa y óptima comparar resultados tanto con diferentes funcionales empleados como con distintos refinamientos sobre la malla. El sistema es versátil al conjuntar una serie de herramientas integradas y conectadas para su uso en todo momento.

Contando con una malla convexa y trabajando en el panel de control **Applications Control Panel** al pulsar la opción de botón **PDE solve** el panel anterior desaparecerá dando lugar a **PDE Solver Control Panel** y una nueva ventana de despliegue gráfico: **Surface Solution** se tendrá en pantalla junto con la típica ventana de despliegue de la malla; así por una parte tendremos la malla en la región de trabajo y por otra la superficie solución cuando necesitemos graficarla.

El panel de control consta de un **pop-menu** para elegir el tipo de ecuación a resolver. Por el momento sólo se dispone de ecuaciones diferenciales elípticas escritas en la forma:

$$a \frac{\partial^2 U}{\partial x^2} + 2b \frac{\partial^2 U}{\partial x \partial y} + c \frac{\partial^2 U}{\partial y^2} + dU = g(x, y)$$

donde $a, b, c, d \in \mathbb{R}$ son constantes y g es una función escalar sobre la región Ω . El sistema cuenta con campos de entrada para introducir los valores de las constantes y un **pop-menu** donde el usuario puede elegir la función g entre constante o bien una sinusoidal. Las condiciones de frontera de la EDP son tratadas sobre cada uno de los cuatro segmentos de frontera y para cada una de ellas **Bnd1**, **Bnd2**, **Bnd3** y **Bnd4** el usuario puede elegir entre ser constantes, una función parabólica o bien sinusoidal; a un costado de cada **pop-menu** se ha dispuesto de un campo de entrada para introducir el parámetro en cada una de las funciones, éste parámetro básicamente es en valor máximo que sobre cada segmento de frontera tiene la función partiendo ésta de los vértices correspondientes.

Una vez elegida la ecuación diferencial y los valores a la frontera, podemos intentar resolverla y en su caso graficar los resultados. Se dispone de botones de opción **Solve**, **Plot results**, **Save results** y los convencionales **H**elp y **Q**uit.

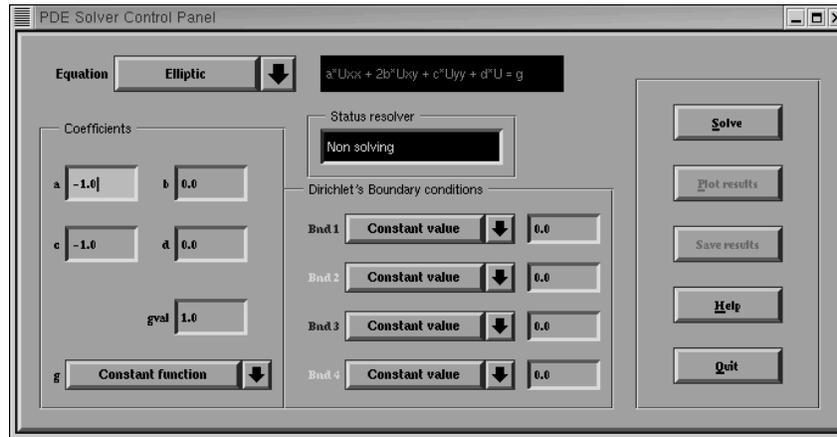


Figura A.36: Panel de control del resolvidor de EDP.

- Solve** Al pulsar esta opción se realizará la llamada al procedimiento que resuelve la EDP y, dependiendo si el resultado ha sido exitoso o no en la caja de mensajes **Status solver** se desplegará información al respecto. El **drive** usa un esquema iterativo de SOR de 9 puntos.
- Plot results** Habiendo obtenido un resultado exitoso se tendrá activa ésta opción con la cual se podrá graficar la superficie solución. Esta opción actualiza la superficie existente en la ventana de despliegue **Surface solution**.
- Save results** El resultado solución obtenido se guarda en el archivo de datos bajo el formato **MESH**. Con esta opción tendremos un **browser** con el cual indicamos la trayectoria donde estará el archivo y el nombre de éste.
- Help** Proporciona ayuda en línea a través del **browser** Netscape.
- Quit** Salida del módulo EDP.

Una vez obtenido el resultado se grafica con la opción **Plot results** en la ventana **Surface solution** y mediante las teclas de flechas podemos rotar sobre su eje la superficie, ver Figura A.38.

Sobre la ventana de despliegue de la superficie y con el botón derecho del mouse, elegimos en un menú elegimos algunas características del despliegue de superficie como son: **Model view** y **View type**. En **Model view** podemos elegir el modo **Smooth** o **Flat** si queremos un suavizamiento de los arcos de las líneas o sean éstas simplemente desplegadas. En **View type** podemos elegir ver en superficie o en malla la solución. En la Figura A.39 mostramos la forma de despliegue por **Surface** y por **Mesh**.

El formato **MESH** elegido para guardar los resultados es compatible con **Geomview**, el visualizador interactivo de imágenes 3D, con él se puede desplegar los resultados desde diversos ángulos, sombras y distintos focos de visión.

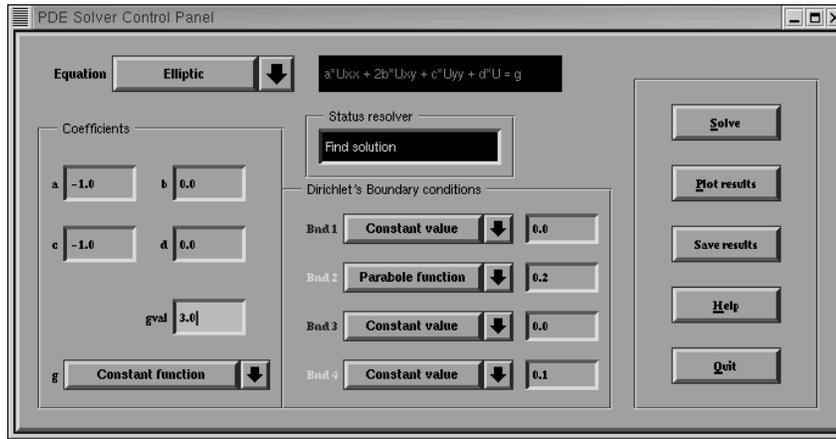


Figura A.37: Elección de una EDP y mensaje del resultado obtenido por SOR

En algunos casos puede ser que no obtengamos una solución por diversas razones, si entre ellas se encuentra que la malla no es lo suficientemente fina, podemos volver al módulo de aplicaciones para hacer un refinamiento de la malla e intentar resolverlo de nueva cuenta. El módulo implantado solamente resuelve algunas ecuaciones diferenciales particulares del tipo elíptico.

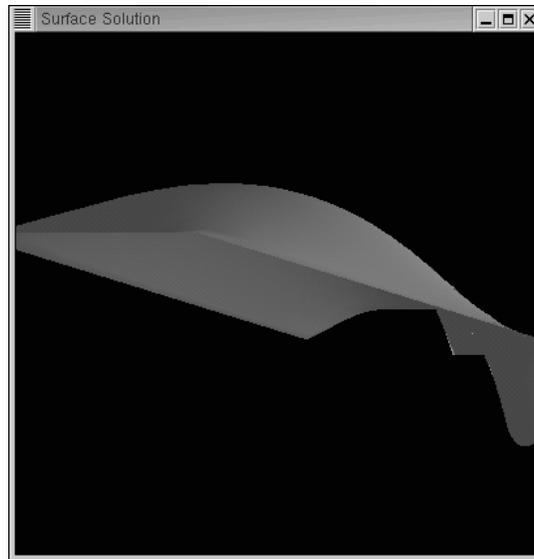


Figura A.38: Graficación de la superficie solución.

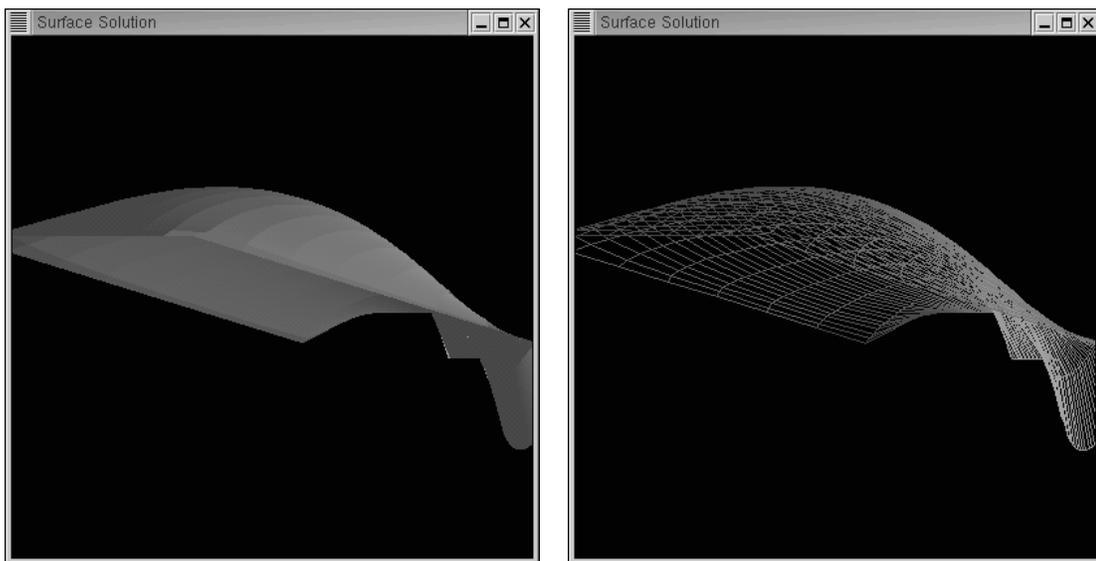


Figura A.39: Despliegue de la solución tanto por superficie como por la malla.

A.5 Entrada y salida de datos

La especificación de los datos que emplea el sistema ayuda a conocer cómo intercambiar la información hacia y desde el sistema con nuestra aplicación. Pasemos a describir el formato de datos que maneja el sistema.

Los datos que se manejan son tres: contorno, malla y superficie. Cada uno ello tiene un formato diferente todos en formato ASCII.

A.5.1 Formato del archivo de datos de los contornos

Los archivos de un contorno cuentan con un formato muy simple: el número de puntos, una identificación de los segmentos de frontera que cuenta y las coordenadas del contorno. Enfatizamos que los contornos son cerrados por lo que la descripción debe considerar que el punto de inicio y el punto final sean los iguales; de igual manera para fines prácticos los consideramos con un orden: el sentido contrario a las manecillas del reloj. Otro punto más es que el sistema acepta regiones con agujeros siempre que se señale.

El archivo de datos tiene el formato:

```
m ibflag nb1 nb2 nb3 nb4
x_1 y_2
x_2 y_2
x_3 y_3
. .
. .
. .
x_m y_m
```

bajo este formato $x_m=x_1$ y $y_m=y_1$. Se explica cada una de las entradas.

<code>m</code>	Es el número total de puntos a la frontera. El primero y el último deben repetirse para considerar el contorno cerrado.
<code>ibflag</code>	Indicador del tipo de contorno. =1. Los segmentos de frontera irán definidos. =13. Los segmentos de frontera 1 y 3 son iguales. En una región con agujeros es necesario hacer un corte y entonces dos segmentos de frontera son iguales. =24. Los segmentos de frontera 2 y 4 son iguales. Similar al caso anterior. =0. No hay segmentos de frontera determinados para el mapeo. El sistema considerará una elección automática.
<code>nb1</code>	Número de puntos que describen el segmento de frontera 1.
<code>nb2</code>	Número de puntos que describen el segmento de frontera 2.
<code>nb3</code>	Número de puntos que describen el segmento de frontera 3.
<code>nb4</code>	Número de puntos que describen el segmento de frontera 4.
<code>x_i y_i</code>	Coordenadas x y y del contorno.

Un ejemplo de archivo donde la frontera 1 y la frontera 3 son iguales es el siguiente:

```
13 13 2 7 2 5
 0.0  0.0
 9.0  0.0
12.0  4.0
16.0  4.0
19.0  1.0
15.0 -3.0
11.0 -3.0
 9.0  0.0
 0.0  0.0
20.0 -15.0
30.0  1.0
12.0 15.0
 0.0  0.0
```

Cuando la región consta de más de un agujero necesitamos realizar más de un “corte”, tratando de unir fronteras exteriores con la interior y entonces “desdoblar” la región.

En el caso de la malla `cuad2pp.red` que hasta ahora nos ha servido para ejemplificar los módulos del sistema tiene por contorno al archivo

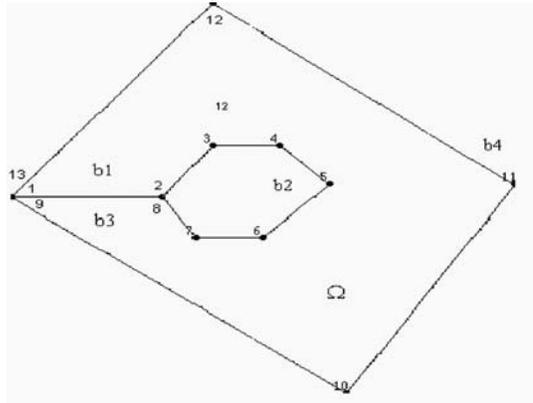


Figura A.40: Contorno de una región con un agujero

```

7      1      2      2      4      2
0.4000    0.0000
0.6000    0.0000
1.0000    0.0000
1.0000    1.0000
0.0000    1.0000
0.0000    0.0000
0.4000    0.0000

```

Cabe destacar en los datos del contorno que el sistema guarda en archivos son impresos en formato F11.4 de FORTRAN. Sin embargo, la lectura el formato es libre, sólo debemos respetar, con un espacio en blanco las distintas entradas así como el salto de línea al finalizar una colección de datos.

La orientación de los contornos es importante para la posterior obtención de mallas óptimas por medios discretos. Si el contorno en cuestión presenta orden negativo (sentido de las manecillas del reloj) el sistema realiza el cambio a sentido positivo (contrario a las manecillas del reloj) de manera transparente para el usuario.

A.5.2 Formatos de archivo de mallas

El sistema maneja dos tipos de formato para los archivos de mallas, formato XY y formato RED. El primero ha sido empleado en sistemas anteriores y su formato sigue la estructura de una matriz. El segundo es un formato de reciente diseño donde los nodos interiores son fácilmente localizables y ha resultado ser éste último muy práctico en la optimización de los funcionales discretos. Pasemos a describir cada uno de ellos.

Tomando en cuenta los nodos interiores y exteriores de la malla como un gran

matriz, numeramos los nodos y su posición por filas y columnas. Fijando los segmentos de frontera “abajo”, “derecha”, “arriba” e “izquierda”, nuestra matriz de nodos partiría de abajo hacia arriba y de izquierda a derecha, ver Figura A.41.

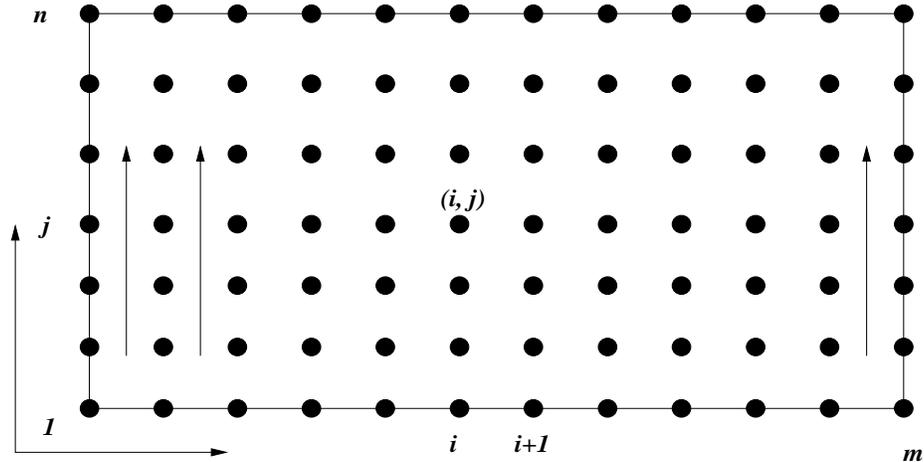


Figura A.41: Matriz de nodos y su orientación para la numeración de los mismos.

Siguiendo esta idea, un nodo r tendría coordenadas i y j ; es decir, en la columna i y fila j de nodos. Una celda $c_{i,j}$ tendrá por vértices a los nodos (i, j) , $(i + 1, j)$, $(i + 1, j + 1)$, $(i, j + 1)$. Bajo estas suposiciones, veamos cada uno de los formatos dispuestos para la malla.

Formato XY

El formato XY consiste en imprimir en archivo la matriz de nodos por columnas, iniciando desde la izquierda y hasta la última columna imprimiendo 7 valores por línea de archivo bajo el siguiente esquema

```
m n
filename
x_11 y_11 x_12 y_12 x_13 y_13 x_14
y_14 x_15 y_15 x_16 y_16 x_17 y_17
. . .
. . .
. . .
x_21 y_21 x_22 y_22 x_23 y_23 x_24
y_24 x_25 y_25 x_26 y_26 x_27 y_27
. . .
. . .
. . .
```

```

x_m1 y_m1 x_m2 y_m2 x_m3 y_m3 x_m4
y_m4 x_m5 y_m5 x_m6 y_m6 x_m7 y_m7
. . .
. . .
. . .
. . . x_mn y_mn

```

En este formato $m \times n$ es la dimensión de la malla. Expliquemos cada una de las entradas

m	Número de puntos en cada línea horizontal de la malla. Tengamos en cuenta que la dimensión de la malla es mn .
n	Número de puntos en cada línea vertical de la malla. Tengamos en cuenta que la dimensión de la malla es mn .
filename	Nombre del archivo de datos.
$x_{i,j}, y_{i,j}$	Coordenadas de los nodos de la malla. Las coordenadas se imprimen 7 por cada línea de archivo luego viene un salto de línea y se prosigue hasta finalizar todos los nodos.

Este formato como bien hemos señalado es natural en cuanto al orden empleado para nombrar los nodos como elementos de una matriz. El formato de impresión sigue siendo F11.4 aunque para su lectura basta que los datos se encuentren separados por un espacio siguiendo la distribución de sus elementos.

Formato RED

El formato RED surge como una necesidad para delimitar los nodos a la frontera y los nodos interiores para de ésta forma, manipular fácilmente los nodos internos de la malla en el módulo de optimización. Los nodos interiores son las incógnitas al problema de optimización. Bajo ésta idea el formato RED describe primero los nodos a la frontera y seguidamente los nodos interiores, describamos el formato de los archivos. Un archivo típico de mallas bajo el formato RED tiene la forma

```

m n
filename
x_11 y_11 x_21 y_21 x_31 y_31 x_41
y_41 x_51 y_51 x_61 y_61 x_71 y_71
. . .
. . .
. . .
. . . x_m1 y_m1 x_m2 y_m2
x_m3 y_m3 x_m4 y_m4 x_m5 y_m5 x_m6

```

```

. . .
. . .
. . .
. . . x_12 y_12 x_22 y_22
y_23 x_23 y_24 x_24 y_25 x_26 y_26
. . .
. . .
. . .
. . . x_2m-1 y_2m-1 x_32 y_32
x_33 y_33 x_34 y_34 x_35 y_35 x_36
. . .
. . .
. . .
. . . x_mn y_mn

```

donde

m	Número de puntos en cada línea horizontal de la malla. Tengamos en cuenta que la dimensión de la malla es mn .
n	Número de puntos en cada línea vertical de la malla. Tengamos en cuenta que la dimensión de la malla es mn .
filename	Nombre del archivo de datos.
x_r, y_r	Coordenadas de los nodos de la malla. Los datos se imprimen 7 por cada línea de archivo luego viene un salto de línea y se prosigue hasta finalizar todos los nodos. Primero se imprimen los nodos de la frontera 1, luego los que se encuentran en la frontera 2 sin repetir, luego los nodos de la frontera 3 sin repetir y finalmente los de la frontera 4 sin repetir ni uno. Luego vienen los nodos interiores imprimiéndolos por líneas verticales inmediatamente después de la frontera 4 hasta antes de la frontera 2.

Este formato permite contar en un arreglo a todos los nodos de la malla y a partir de un número $ibo = 2(2m + 2n - 4)$ todos los nodos interiores. El formato de impresión de los datos es F11.4, es decir, con cuatro cifras decimales.

Si el formato a leer desde un archivo no coincide con el esperado, en el despliegue gráfico observaremos información que no corresponde a la malla.

A.5.3 Formato de archivos MESH

Los archivos MESH son resultados de la superficie solución de una EDP obtenida por el sistema. El formato es muy simple: describe los puntos x, y, z de la superficie a partir de la numeración por columnas de la malla. El formato es el siguiente:

```
MESH
m n
x_11 y_11 z_11
x_12 y_12 z_12
. . .
. . .
. . .
x_1n y_1n z_1n
x_21 y_21 z_21
x_22 y_22 z_22
. . .
. . .
. . .
x_mn y_mn z_mn
```

Para el formato de impresión seguimos empleando F11.4. Los archivos con éste formato pueden ser leídos con el programa *Geomview* para visualizar algunas características de la superficie desde distintos puntos de observación y sombras.

A.6 Sobre el sistema UNAMALLA y su nombre

A.6.1 Sobre el nombre UNAMALLA

El nombre UNAMALLA es la unión de dos siglas: UNAM y MALLA. Puede ser leído como UNA-MA-LLA y pronunciado de diversas formas: UNA-MALLA, UNAM-MALLA. La idea del nombre es connotar el hecho de que en la UNAM se realizan investigaciones sobre la generación de mallas.

A.6.2 Sobre el grupo de trabajo UNAMALLA

El grupo de trabajo UNAMALLA consta de estudiantes y profesores de distintas universidades del país interesados en el área de generación de mallas y sus aplicaciones.

Las universidades participantes son: Universidad Michoacana de San Nicolás de Hidalgo, Universidad Autónoma de Saltillo, Coahuila y la Universidad Nacional Autónoma de México.

Los responsables del proyecto son: Dr. Pablo Barrera (UNAM) y el Dr. J. Gerardo Tinoco Ruíz (UMICH). Los participantes en el proyecto son: Fco. Domínguez Mota (UMICH), Irma D. García Calvillo (UAdeC), Adriana Hernández Rivera (UNAM), Guilmer González Flores (UNAM) y Luis Alberto Vásquez (UNAM).

A.7 Requerimientos técnicos del sistema

El sistema está escrito en lenguaje `Fortran 77` y `C ANSI` por lo que los módulos y los `drives` de tareas específicas pueden ser empleados directamente en otras aplicaciones. Para la interfase con el usuario, las `GUI` (Graphical User Interface), se empleó la biblioteca libre `XForms Library` basada exclusivamente en la biblioteca `Xlib`, lo cual hace que el sistema pueda ser empleado en casi cualquier computadora que cuente con el sistema de ventanas `X Windows` haciendo con esto portable el sistema hacia los sistemas donde pueda trabajar correctamente la biblioteca `Xlib`.

Para el despliegue de datos del contorno y la malla se emplearon primitivas de `OpenGL` haciendo eficiente el manejo y despliegue de los gráficos en sistemas que lo soporten. `OpenGL` fue diseñado para Workstation `SGI` por lo que el sistema funciona muy bien en este tipo de plataformas. Las primitivas de `OpenGL` ha tenido gran aceptación entre los diseñadores de sistemas gráficos a tal punto que existe software y hardware basados en ellas. Uno de estas herramientas es la biblioteca `Mesa` que interpreta primitivas de `OpenGL` y es portable a casi cualquier plataforma con sistemas `X Windows`.

El sistema `UNAMALLA` se carga en memoria y hace uso de los recursos existentes por lo que el único límite en usuarios simultáneos es el que pueda soportar el sistema. Cualquier falta de memoria es reportada en la caja de estado del panel de control. Con el sistema ha sido posible generar y optimizar mallas de 999×999 en computadoras PC teniendo por referencia el sistema básico Linux RedHat 6.2 con 32Mb y una tarjeta de video de 4Mb.

A.7.1 Plataformas soportadas

El sistema `UNAMALLA` trabaja sobre una gran cantidad de sistemas Unix. El sistema ha sido probado satisfactoriamente en los sistemas: Linux, SGI y Sun (Sun-OS 4.1x y Solaris). La distribución precompilada se encuentra disponibles en tales sistemas.

El sistema está escrito en `Fortran 77` y `C ANSI` usa las bibliotecas `XForms Library` y las primitivas de `OpenGL` para el despliegue gráfico. La creación y manejo de ventanas se logra con la biblioteca `GLUT`. Así pues, el sistema puede ser compilado en casi cualquier plataforma Unix con sistema de ventanas `X Windows` donde éstas bibliotecas se encuentren disponibles: Por ejemplo: FreeBSD, HP, RISC, Dec Alpha, RS/6000, NeXT, por mencionar algunas.

En algunas de las versiones precompiladas, las bibliotecas gráficas son ligadas de manera estática por lo que no es necesario contar con tales bibliotecas instaladas en la máquina.

Cabe señalar que existe una versión para PC en el sistema operativo DOS. Esta versión es más pequeña y hace uso de las bibliotecas gráficas de Microsoft C 6.0A. En éste tipo de compiladores la representación por 3D es muy complicada por esto no se encuentra implantada el módulo PDE.

A.7.2 Instalación de las distribuciones binarias

Si se ha hecho de una copia de la distribución binaria para un sistema Unix (Linux, SGI, Solaris, etc.) puede ejecutar el sistema luego de desempacarlo al introducirse al directorio `unamalla` y tipeando en la línea de comandos `mall`.

El programa binario se encuentra dentro del directorio principal de distribución el cual contiene algunos subdirectorios donde el usuario cuenta con contornos y mallas de ejemplos así como depositar ahí sus resultados.

Al ejecutar el sistema se tendrá en pantalla dos ventanas `Principal Control Panel` y `Display Window` que podrá moverlas dentro del escritorio del sistema de ventanas `X Windows` preferido para su uso. Si existe algún problema para el despliegue gráfico, contacte con el soporte técnico del sistema.

A.7.3 Soporte Técnico para el Sistema.

Existen muchas formas de obtener apoyo para el uso e instalación del sistema.

1. Visitando la página UNAMALLA, <http://tycho.fciencias.unam.mx/unamalla>. El cual contiene los documentos más relevantes del sistema y los funcionales discretos empleados así como versiones no definitivas de nuevos desarrollos y sistemas experimentales.
2. Contactando con el grupo UNAMALLA para el soporte. Puede enviar un correo a la dirección `unamalla@athena.fciencias.unam.mx` para recibir ayuda sobre el uso del sistema así como puede reportar en ésta dirección cualquier problema que el diseñador y programador no haya previsto para su funcionamiento.
3. Contactando con la persona que diseñó y programó el sistema y uno de quienes se encargan del mantenimiento del mismo: Guilmer González en la dirección de e-mail: `gfgf@athena.fciencias.unam.mx`

Apéndice B

Métodos de Optimización de Gran Escala

B.1 Introducción

Comenzando con x_0 los algoritmos de optimización generan una sucesión $\{x_k\}_{k=0}^N$ hasta que no avance o cuando algún iterando satisfaga las condiciones de mínimo local con alguna precisión, es decir, que converge. Para decidir cómo moverse de una iteración a la siguiente, los algoritmos utilizan información de la función en x_k y posiblemente de iteraciones anteriores x_0, x_1, \dots, x_{k-1} , con esta información se encuentra un nuevo punto x_{k+1} donde el valor de la función es menor que en x_k .

Los métodos de descenso generan una sucesión $\{x_k\}$ de aproximaciones a la solución de manera que

$$f(x_{k+1}) < f(x_k)$$

para toda k , esto es, se acepta la nueva iteración x_{k+1} si produce una reducción en el valor de la función objetivo, de tal forma que el valor de la función en la iteración actual es menor que el valor de la función en iteraciones anteriores. Una de las características de los métodos de descenso es que en la iteración k se debe avanzar sobre una dirección p la cual tiene que ser una dirección de descenso, esto es, una dirección sobre la que la función, localmente, decrezca.

Definición. Una dirección p se dice de descenso en el punto x_k si la derivada direccional de f en x_k sobre la dirección p es negativa, esto es, si satisface

$$p^t \nabla f(x_k) < 0$$

La condición de descenso por sí sola no es suficiente para garantizar que la iteración $\{x_k\}$ se aproxime a un mínimo local. Se requieren condiciones más fuertes para forzar a la sucesión a entrar en una vecindad del mínimo local. Una vez que la iteración entra en tal vecindad, los métodos de descenso usualmente convergen rápidamente al mínimo local.

Existen dos estrategias fundamentales para llevar la iteración a una vecindad del mínimo local: búsqueda lineal y región de confianza. La mayoría de los algoritmos utilizan una de ellas.

B.1.1 Búsqueda lineal y región de confianza

En la estrategia de búsqueda lineal el algoritmo escoge una dirección de descenso, p_k , llamada dirección de búsqueda, y busca a lo largo de esa dirección, partiendo de x_k , un nuevo elemento de la iteración donde el valor de la función decrezca.

La distancia a recorrer sobre la dirección p_k puede encontrarse resolviendo aproximadamente el problema unidimensional de encontrar un $\alpha^* \in \mathbb{R}$ tal que

$$\alpha^* = \arg \min_{\alpha > 0} f(x_k + \alpha p_k)$$

Si este problema se resuelve exactamente, se tendría el máximo beneficio sobre la dirección p_k , sin embargo resolver el problema exactamente es muy costoso en la práctica, es por esto que se utilizan soluciones aproximadas al mínimo de esta función por medio de un algoritmo que genera un número limitado de puntos de prueba, hasta que se encuentra un α cercano al mínimo exacto. Una vez que se tiene la nueva iteración x_{k+1} se genera una nueva dirección de descenso y se repite el proceso.

En la segunda estrategia, conocida como región de confianza, se utiliza información local de f para construir un modelo de la función, Ψ_k , de manera que el comportamiento de Ψ_k cerca de x_k sea semejante al comportamiento de f . Entonces minimizando Ψ_k nos acercaremos al mínimo de f . Como el modelo Ψ_k puede no ser una muy buena aproximación de f cuando no se está cerca de x_k , se restringe la búsqueda del mínimo de Ψ_k en alguna región alrededor de x_k , usualmente a una bola con centro en x_k y radio Δ_k , a Δ_k se le conoce como el radio de la región de confianza. En otras palabras, generamos el paso p_k como

$$p_k = \arg \min_p \Psi_k(x_k + p) \tag{B.1}$$

donde $x_k + p$ está dentro de la región de confianza. Si con p_k no se produce una reducción suficiente en f , es porque que la región de confianza es muy grande, se disminuye Δ_k y se vuelve a resolver (B.1).

El modelo Ψ_k usualmente es una función cuadrática de la forma

$$\Psi_k(x_k + p) = f(x_k) + p^t \nabla f(x_k) + \frac{1}{2} p^t B_k p$$

donde $\nabla f(x_k)$ es el gradiente de la función y B_k es una matriz que aproxima la matriz hessiana en x_k .

La estrategia de región de confianza, determina primero cuánto debe moverse y después la dirección en la que hay que hacerlo, mientras que la búsqueda lineal primero se determina la dirección sobre la que hay que avanzar y después cuánto moverse sobre esa dirección.

B.2 Método de Newton

Supongamos que tenemos $f : \mathbb{R}^n \rightarrow \mathbb{R}$ una función dos veces continuamente diferenciable. El método de Newton para minimizar una función puede derivarse suponiendo que tenemos una aproximación x_k al mínimo local de f y que en una vecindad de x_k se satisface

$$f(x_k + w) \approx f(x_k) + \Psi(w)$$

donde

$$\Psi(w) = \nabla f(x_k)^t w + \frac{1}{2} w^t \nabla^2 f(x_k) w$$

es el modelo local cuadrático de f en x_k . Si esta aproximación es buena, entonces una mejor aproximación se obtiene con

$$x_{k+1} = x_k + s_k$$

donde el paso s_k es el mínimo de Ψ_k . Debido a las condiciones de un mínimo local, s_k debe ser tal que

$$\nabla \Psi_k(s_k) = \nabla f(x_k) + \nabla^2 f(x_k) s_k = 0$$

entonces el método de Newton parte de una aproximación inicial x_0 y genera una sucesión de la forma

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k) \tag{B.2}$$

Uno de los principales resultados del método de Newton es su convergencia local para cualquier punto inicial que se encuentre en una vecindad del mínimo local, esto es, que se encuentre en el llamado *dominio de atracción*, la región donde se garantiza que la iteración de Newton converge. Desafortunadamente para muchos problemas este dominio de atracción es muy pequeño y es necesario utilizar estrategias de globalización de manera que hagan que la iteración en algún momento entre en él, de forma que una vez ahí, la convergencia está garantizada.

Una de las principales características del método de Newton es su velocidad de convergencia. Si la función es Lipschitz continua entonces en una vecindad del mínimo local, el método converge cuadráticamente, esto es, existe $\beta > 0$ tal que

$$\|x_{k+1} - x^*\| \leq \beta \|x_k - x^*\|^2$$

La convergencia cuadrática implica que el número de cifras significativas de x_k al aproximar a x^* se duplica en cada iteración.

El método de Newton con alguna estrategia de globalización como búsqueda lineal o región de confianza opera muy bien en la práctica, pero tiene la desventaja de ser muy costoso. Como una alternativa recomendable, se proponen los métodos Quasi-Newton, los cuales tienen una velocidad de convergencia menor que la de Newton, superlineal, pero con la ventaja de ser económicos y confiables. Se recomiendan sobre todo en problemas densos.

B.3 Métodos Quasi-Newton

Muchos de los métodos de optimización están basados en modelos cuadráticos locales de la función objetivo, la forma más conocida de este tipo de métodos es el método de Newton.

La k -ésima iteración del método de Newton está dada por

- i) Resolver $\nabla^2 f(x_k) s_k = -\nabla f(x_k)$
- ii) $x_{k+1} = x_k + s_k$

La principal desventaja del método de Newton es el conocimiento de la segunda derivada de la función en cualquier punto, en muchas aplicaciones lo más que puede tenerse es la primera derivada. Es por esto que se generaron métodos relacionados con el método de Newton donde solo se hace uso de la información de la primera derivada de la función. La primera idea es aproximar la matriz hessiana de f por medio de diferencias de los vectores gradientes, pero esto tiene el inconveniente de que la matriz resultante puede no ser positiva definida, además se requieren n evaluaciones del gradiente para cada iteración, el proceso puede resultar muy costoso.

Para superar estas desventajas Davidon en 1958, introdujo una nueva clase de métodos conocidos como Quasi-Newton, éstos son métodos tipo Newton en donde la inversa de la matriz hessiana es aproximada por una matriz simétrica positiva definida, la cual es corregida o actualizada en cada iteración. La estructura básica de estos métodos en la iteración k , partiendo que se cuenta con H_k , una aproximación a la inversa de la matriz hessiana en x_k , es la siguiente:

- i) $s_k = -H_k \nabla f(x_k)$
- ii) $x_{k+1} = x_k + s_k$
- iii) Actualizar H_k para obtener H_{k+1} .

Este método presenta algunas ventajas respecto al método de Newton:

- a) Solo requiere información de la primera derivada de la función.
- b) H_k positiva definida, implica que la dirección encontrada es siempre una dirección de descenso.
- c) Reduce el número de flops por iteración.

El punto c) se alcanza ya que se puede aproximar la inversa de la matriz hessiana, en lugar de la matriz hessiana directamente, lo cual evita la resolución de un sistema lineal en cada iteración. Debido a estas propiedades es que los métodos Quasi-Newton son bastante utilizados en la práctica. Su desempeño es muy satisfactorio, el orden de convergencia ya no es cuadrático como en el caso de Newton, se reduce a superlineal.

Existen varias formulaciones para la actualización de la matriz hessiana, lo cual da lugar a varios métodos tipo Quasi-Newton, de las mas eficientes podemos mencionar la actualización tipo BFGS, en donde dada B aproximación a la inversa de la matriz hessiana en x , se obtiene la formulación

$$B_+ = B + \frac{(s - By)s^t + s(s - By)^t}{s^t y} - \frac{y^t (s - By) s s^t}{(s^t y)^2}$$

que es una aproximación a la inversa de la matriz hessiana en el punto x_+ , donde

$$s = x_+ - x, \quad y = \nabla f(x_+) - \nabla f(x)$$

Este tipo de métodos también admite estrategias de globalización como búsqueda lineal y región de confianza.

B.4 Problemas de gran escala

En la actualidad se han desarrollado algoritmos muy eficientes para tratar problemas de optimización con un gran número de variables, llamados problemas de gran escala. La adaptación del método de Newton a este tipo de problemas, da como resultado un método muy efectivo al trabajar con problemas de grandes dimensiones: el método de Newton truncado, éste consiste en obtener una aproximación a la dirección de Newton, resolviendo las ecuaciones de Newton, con algún método iterativo. También se tienen adaptaciones para los métodos Quasi-Newton que dan lugar a los llamados métodos de memoria limitada.

B.4.1 Métodos de Newton truncado

El método de Newton, como se ha mencionado, es la base de muchos de los métodos de optimización, esto es, métodos que aproximan f localmente por una función cuadrática Ψ . Si el número de variables n , es grande, entonces el método de Newton puede ser problemático ya que requiere el cálculo y almacenamiento de la matriz Hessiana de segundas derivadas parciales. Para funciones de gran escala esto representa grandes costos.

En esta sección presentamos una clase de métodos adecuados para problemas de gran escala llamados métodos de Newton Truncado, éstos métodos están basados en minimizar aproximadamente la función cuadrática Ψ usando un esquema iterativo tal como el algoritmo de Gradiente Conjugado lineal. Un algoritmo de Newton Truncado se compone de dos subalgoritmos: un algoritmo externo no lineal que controla la minimización y un algoritmo interno lineal para minimizar aproximadamente Ψ .

Recordemos que la minimización de la cuadrática Ψ , que determina el paso básico de Newton p_k^N , se obtiene resolviendo el sistema lineal simétrico de $n \times n$

$$\nabla^2 f(x_k)p_k = -\nabla f(x_k) \quad (\text{B.3})$$

conocido como las ecuaciones de Newton.

B.4.2 Descripción básica del método

Como el método de Newton está basado en la expansión en serie de Taylor cerca de la solución del problema de minimización, no hay garantía de que la dirección de búsqueda calculada sea crucial lejos de x^* . De hecho, al inicio del proceso una aproximación razonable a la dirección de Newton puede ser casi tan efectiva como la dirección de Newton misma. Gradualmente, conforme nos acerquemos a la solución, la dirección de Newton tiene mas significado.

Esto sugiere utilizar un método iterativo para resolver las ecuaciones de Newton. Además, debe ser un método iterativo con tolerancia variable tal que lejos de la solución (B.3) no se resuelva con mucha precisión. Solo cuando nos acerquemos a la solución consideraremos la solución de (B.3) con precisión. También conforme nos acerquemos a la solución x^* , el sistema de ecuaciones a resolver será mas similar, consecuentemente es posible que una aproximación cercana a la solución del sistema lineal pueda encontrarse sin mucho esfuerzo utilizando información anterior.

Como el sistema lineal no se resuelve exactamente, los pasos que resultan al resolver las ecuaciones de Newton aproximadamente, se conocen como pasos de Newton inexactos.

Se han propuesto varios métodos para la iteración interna, el más recomendable para esta aplicación es el algoritmo de gradiente conjugado lineal, ya que no requiere el almacenamiento explícito de la matriz del sistema, solo es necesaria una rutina que realice el producto matriz–vector. Un requerimiento de este método es que la matriz de coeficientes debe ser positiva definida, se sabe que la matriz Hessiana tiene garantía de ser positiva semi–definida en la solución del problema de minimización, pero puede ser indefinida en cualquier otro lugar; entonces, el método iterativo para resolver (B.3) debe ser capaz de detectar los sistemas indefinidos.

Enseguida presentamos el algoritmo del Método de Newton truncado.

Algoritmo: Newton Truncado

Iteración principal

for $k = 0, 1, \dots$

Calcular $f(x_k), \nabla f(x_k), \nabla^2 f(x_k)$

Iteración menor

Calcular p_k tal que $B_k p_k = -\nabla f(x_k)$

donde B_k es alguna aproximación definida positiva
a la hessiana en x_k

end

$x_{k+1} = x_k + p_k$

end

end

Este algoritmo, al igual que el método de Newton estándar, converge localmente y además se pueden aplicar estrategias de globalización como búsqueda lineal y región de confianza para obtener convergencia global.

B.5 Métodos Quasi–Newton para problemas de gran escala

Los métodos Quasi–Newton descritos anteriormente no se aplican directamente a problemas de optimización grandes, ya que los requerimientos de almacenamiento y cómputo son excesivos. Las aproximaciones a la matriz hessiana o a su inversa son usualmente densas y aún si las matrices se almacenan en forma descompuesta (por ejemplo, almacenando los vectores s_k y y_k para todas las iteraciones $k = 0, 1, \dots$) los requerimientos de memoria crecen en cada iteración y eventualmente sobrepasarán los recursos disponibles.

Sin embargo, hay varias formas de extender los métodos Quasi–Newton para problemas grandes. En la primera aproximación, los métodos Quasi–Newton de memoria

limitada, nos enfocaremos al método de BFGS con memoria limitada.

B.5.1 Memoria Limitada BFGS

Los métodos Quasi-Newton de Memoria Limitada son útiles para resolver problemas grandes donde la matriz hessiana no puede ser calculada a un costo razonable, o son densas. Estos métodos mantienen aproximaciones simples y compactas de las matrices hessianas: en lugar de almacenar las aproximaciones completas de $n \times n$ se guardan solo una cuantos vectores de longitud n que representan las aproximaciones implícitamente. Aún con los modestos requerimientos de almacenamiento estos métodos frecuentemente tienen una aceptable razón de convergencia, casi siempre lineal. Se han propuesto varios métodos de memoria limitada, nos enfocaremos principalmente en el algoritmo conocido como L-BFGS, introducido por Nocedal, y que está basado en la fórmula de actualización BFGS. La idea básica del método es usar información solo de las mas recientes iteraciones para construir la aproximación a la hessiana. La información de la curvatura de iteraciones no recientes, las cuales no aportan información relevante del comportamiento de la hessiana en la iteración actual, son eliminadas para ahorrar espacio en memoria.

Para empezar la descripción del método L-BFGS, recordemos el método BFGS estándar. Cada paso de este método tiene la forma

$$x_{k+1} = x_k - \alpha_k H_k g_k, \quad k = 0, 1, 2, \dots$$

donde α_k es la longitud del paso y H_k se actualiza en cada iteración por la fórmula

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k) s_k^t + s_k (s_k - H_k y_k)^t}{s_k^t y_k} - \frac{y_k^t (s_k - H_k y_k) s_k s_k^t}{(s_k^t y_k)^2}$$

Tenemos una representación mas compacta de las matrices BFGS

$$H_{k+1} = V_k^t H_k V_k + \rho_k s_k s_k^t \tag{B.4}$$

donde

$$\rho_k = \frac{1}{y_k^t s_k}, \quad V_k = I - \rho_k y_k s_k^t \tag{B.5}$$

y

$$s_k = x_{k+1} - x_k, \quad y_k = g_{k+1} - g_k \tag{B.6}$$

decimos que la matriz H_{k+1} se obtiene actualizando H_k usando el par $\{s_k, y_k\}$.

La aproximación H_k generalmente va a ser densa, por lo que el costo de almacenaje y manipulación es bastante alto cuando el número de variables es grande, usualmente H_{k+1} se reescribe sobre H_k y por ser matrices simétricas necesitamos $n(n+1)/2$ lugares

en memoria para almacenarla. Una forma de superar este problema es no formar y almacenar la matriz H_k de $n \times n$ explícitamente, en su lugar almacenamos una versión modificada de H_k implícitamente, almacenando un cierto número, digamos m de los pares de vectores $\{s_i, y_i\}$ que se utilizan en la formulación (B.4)–(B.6).

En general, para $k \leq m$ tenemos la aproximación estándar BFGS

$$\begin{aligned} H_k &= (V_{k-1}^t \cdots V_0^t) H_0 (V_0 \cdots V_{k-1}) \\ &\quad + \rho_0 (V_{k-1}^t \cdots V_1^t) s_0 s_0^t (V_1 \cdots V_{k-1}) \\ &\quad + \rho_1 (V_{k-1}^t \cdots V_2^t) s_1 s_1^t (V_2 \cdots V_{k-1}) \\ &\quad + \cdots \\ &\quad + \rho_{k-1} s_{k-1} s_{k-1}^t \end{aligned} \tag{B.7}$$

Y para $k > m$ tenemos la actualización especial BFGS

$$\begin{aligned} H_k &= (V_{k-1}^t \cdots V_{k-m}^t) H_0 (V_{k-m} \cdots V_{k-1}) \\ &\quad + \rho_{k-m} (V_{k-1}^t \cdots V_{k-m+1}^t) s_{k-m} s_{k-m}^t (V_{k-m+1} \cdots V_{k-1}) \\ &\quad + \rho_{k-m+1} (V_{k-1}^t \cdots V_{k-m+2}^t) s_{k-m+1} s_{k-m+1}^t (V_{k-m+2} \cdots V_{k-1}) \\ &\quad + \cdots \\ &\quad + \rho_{k-1} s_{k-1} s_{k-1}^t \end{aligned} \tag{B.8}$$

A las matrices (B.7) y (B.8) Nocedal las denomina matrices especiales BFGS. Esta aproximación es adecuada para problemas grandes ya que la experiencia práctica ha mostrado que valores de m entre 3 y 20 producen resultados satisfactorios. Con esta definición de H_k el algoritmo L-BFGS puede establecerse como sigue.

Algoritmo L-BFGS

Escoger un punto inicial x_0 y un entero $m > 0$

for $k = 0, 1, \dots$

 Escoger $H_0^{(k)}$

 Calcular $p_k = -H_k g_k$

 Calcular $x_{k+1} = x_k + \alpha_k p_k$, donde α_k se escoge con algún algoritmo de búsqueda lineal

 Si $k > m$

 Eliminar el par de vectores $\{s_{k-m}, y_{k-m}\}$ de memoria

 end

 Calcular $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$

$k = k + 1$

end

Durante las primeras $m - 1$ iteraciones el algoritmo L-BFGS es equivalente al algoritmo BFGS estándar, y al igual que el método estándar, se admiten estrategias de globalización.

B.5.2 Método de Newton punto a punto

Cuando se trabajan funciones con un gran número de variables, debemos tratar de explotar cualquier estructura que presente la función para tener algoritmos más económicos. Muchas de las funciones de gran escala que aparecen en la práctica poseen una propiedad conocida como separabilidad parcial y en los últimos años se han desarrollado métodos que explotan esta estructura en la función objetivo.

En un problema de optimización separable la función objetivo puede descomponerse en una suma de funciones simples que pueden optimizarse de forma independiente, si tenemos

$$f(x) = f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n)$$

entonces podemos encontrar el óptimo de $f(x)$ minimizando cada función f_i , $i = 1, \dots, n$, de manera independiente. En muchos casos, realizar n minimizaciones unidimensionales es más económico que el realizar una minimización n -dimensional.

En muchos problemas de gran escala, la función objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$ no es separable, pero puede escribirse como suma de funciones simples conocidas como *funciones elementales*. Para cada función elemental es posible definir una base ortogonal en \mathbb{R}^n de tal forma que el valor de la función no varíe al movernos sobre estas direcciones base. Si esta propiedad se satisface para todas las funciones elementales decimos que f es *parcialmente separable*. Todas las funciones cuyas hessianas son *sparse* son parcialmente separables, pero también existe la separabilidad parcial en funciones cuyas hessianas no son *sparse*.

La forma más simple de separabilidad parcial aparece cuando la función objetivo puede escribirse como

$$f(x) = \sum_{i=1}^{ne} f_i(x)$$

donde ne es el número de funciones elementales y cada $f_i(x)$ depende solo de unos cuantos componentes de la variable x , entonces los gradientes ∇f_i y las matrices hessianas $\nabla^2 f_i$ de cada función elemental típicamente contendrán solo unos cuantos elementos distintos de cero. El gradiente y la matriz hessiana de la función f están dados por

$$\nabla f = \sum_{i=1}^{ne} \nabla f_i(x), \quad \nabla^2 f = \sum_{i=1}^{ne} \nabla^2 f_i(x)$$

y si se aplica un método tipo Quasi-Newton para resolver el problema de optimización es más económico actualizar por el método Quasi-Newton cada matriz hessiana elemental $\nabla^2 f_i(x)$ en lugar de actualizar toda la matriz hessiana total.

En el problema que tenemos de generación de mallas, podemos observar que el funcional total a optimizar se describe como la suma de un funcional aplicado a cada

celda o a cada triángulo de la malla, esto es, el funcional total a optimizar, trabajando por celdas, es de la forma

$$F(x) = \sum_{i=1}^{N_c} f_i(x)$$

con

$$f_i(x) = f(c_i)$$

donde c_i es una celda representativa de la malla y N_c es el número total de celdas, esto es, el funcional es parcialmente separable.

Los funcionales a optimizar para generar las mallas óptimas, como ya hemos mencionado, son funciones parcialmente separables, tienen un subespacio invariante grande, entonces se pensó en realizar una optimización considerando la función como una función separable, donde cada función elemental dependiera solo de dos variables: las coordenadas de un nodo interior. Esto es, se propone una optimización nodo a nodo.

La idea es la siguiente: al optimizar una malla los nodos a la frontera permanecen fijos y solo nos interesa realizar la optimización para los nodos interiores de la malla. Si consideramos una malla de 3×3 como en la figura B.1, entonces al realizar la optimización resulta un problema de dos variables: la posición del nodo $z = (z_1, z_2) \in \mathbb{R}^2$.

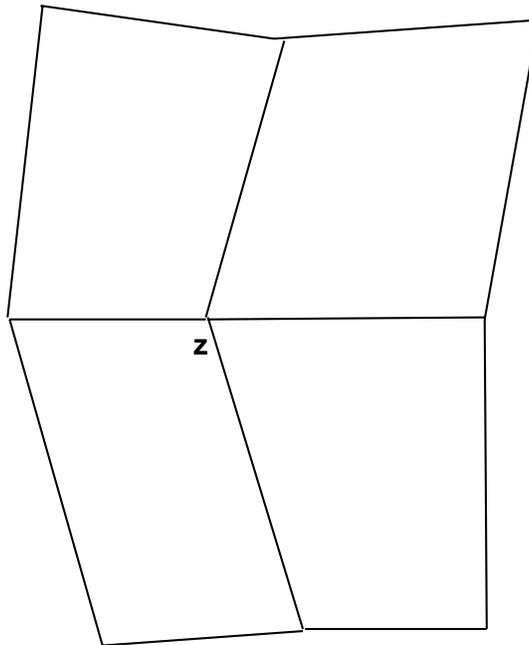


Figura B.1: Una malla de 3×3

Si tenemos una malla de $m \times n$ el valor de un funcional sobre una submalla de 3×3 no varía si movemos un nodo interior de la malla ajeno a esta submalla de 3×3 , entonces resulta razonable realizar la optimización del funcional solo de manera local, esto es, para cada nodo interior P_{ij} extraemos la submalla de 3×3 formada por sus nodos vecinos, como se muestra en la figura B.2, y optimizamos el funcional restringido a esta submalla de 3×3 como ya mencionamos se realiza una optimización de solo dos variables.

Una vez realizada la optimización sobre este nodo se reinserta la submalla de 3×3 ya optimizada a su posición en la malla original, de esta forma solo se alterará el nodo P_{ij} en toda la malla.

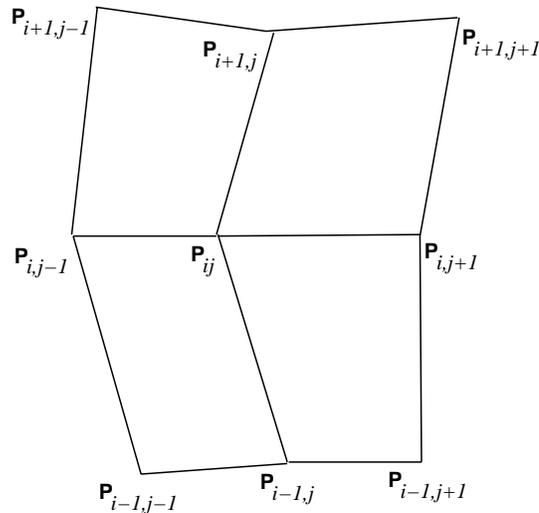


Figura B.2: Malla local del 3×3 con los nodos vecinos del punto P_{ij}

Este proceso se hace para cada nodo interior de la malla, con lo que los requerimientos de memoria son muy bajos, no necesitamos almacenar matrices de grandes dimensiones, solo vectores y matrices de 2×2 que se utilizan en la optimización local.

Optimización local

Veamos ahora cómo es el proceso de optimización local. Tenemos una malla de 3×3 y un funcional definido sobre esta malla. Trabajaremos con funcionales por triángulos,

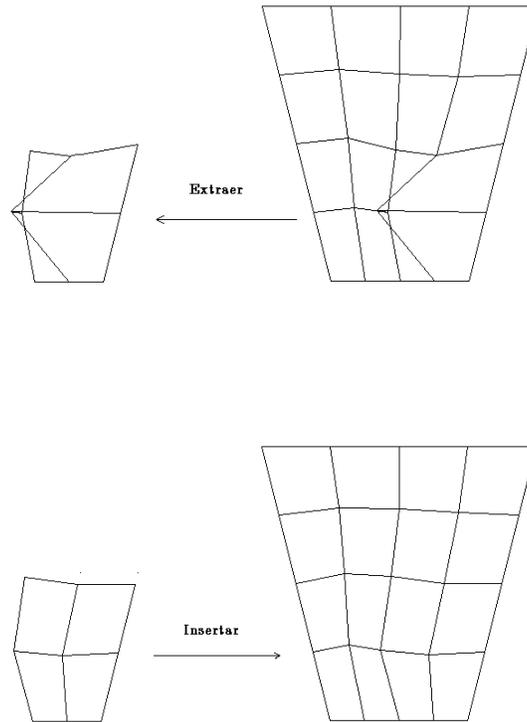


Figura B.3: Extracción e Inserción de la submalla de 3×3

la idea para celdas es semejante. Hay 12 triángulos en la malla de 3×3 en los que interviene el nodo interior P_{ij} , éstos se ilustran en la figura B.4.

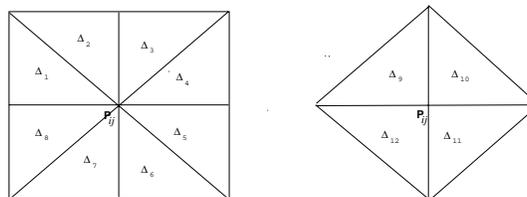


Figura B.4: Los doce triángulos asociados al punto P_{ij}

Entonces el funcional local por optimizar es de la forma

$$f = \sum_{i=1}^{12} f(\Delta_i)$$

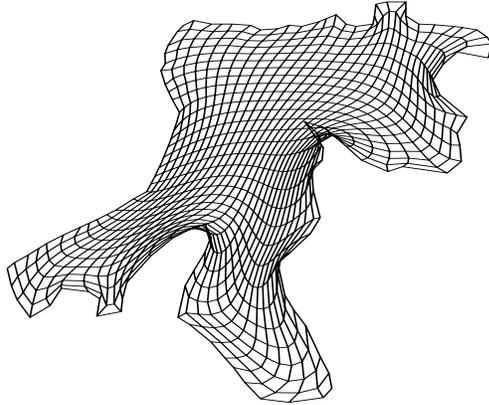
con $f(\Delta_i)$ el funcional sobre el triángulo Δ_i . Calculando el gradiente y la matriz hessiana locales de 2×2 analíticamente, el sistema hace la optimización local utilizando algunas iteraciones del método de Newton en dos dimensiones, por lo que como ya mencionamos, no necesitamos almacenar mas que un vector de grandes dimensiones:

el vector de variables totales, localmente solo se trabajan vectores y matrices de 2×2 , por lo que el sistema puede operar con pocos recursos.

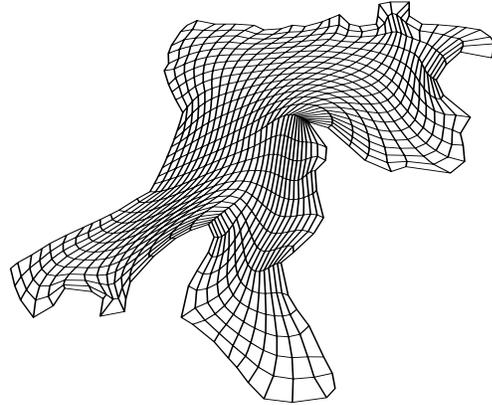
Los resultados obtenidos han sido muy satisfactorios, hemos generado mallas óptimas de grandes dimensiones con este proceso de optimización punto a punto.

Apéndice C

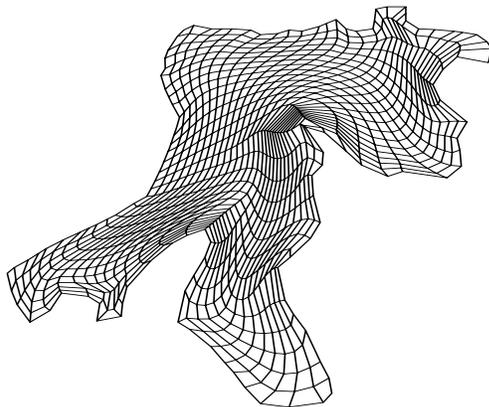
Galería de Mallas



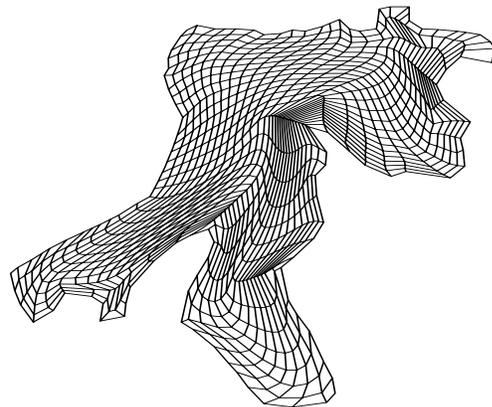
t-Área-Longitud
 $\sigma = 0.75$ no convexa



k-Área-Longitud
 $\sigma = 0.5$ convexa

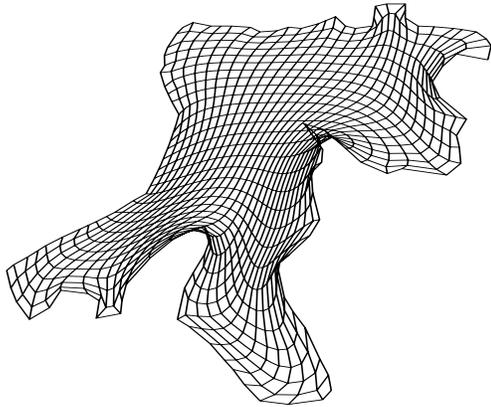


k-Área-Longitud
 $\sigma = 0.75$ convexa

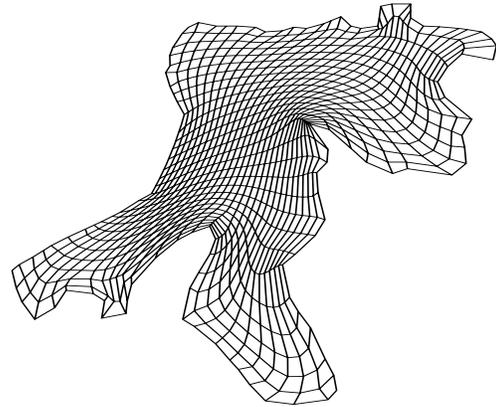


k-Área-Longitud
 $\sigma = 0.9$ convexa

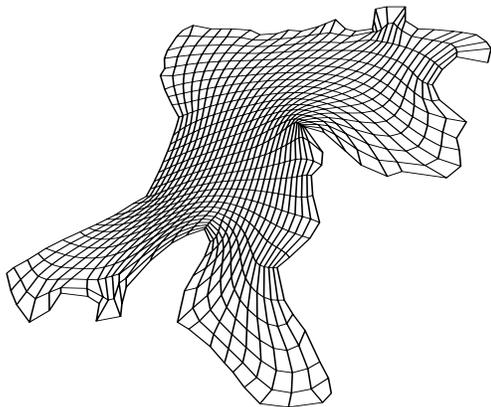
Figura C.1: Mallas generadas sobre la bahía de la habana.



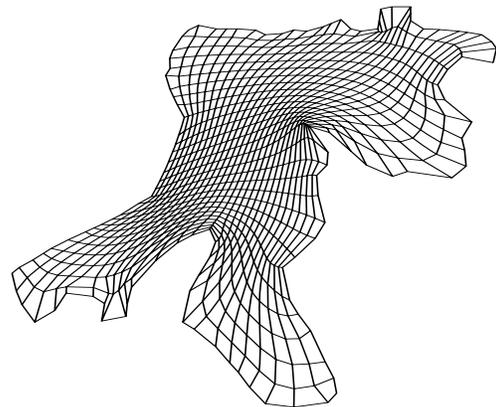
t -Área-Longitud
 $\sigma = 0.75$ no convexa



k -Suavidad-Área
 $\sigma = 0.25$ convexa

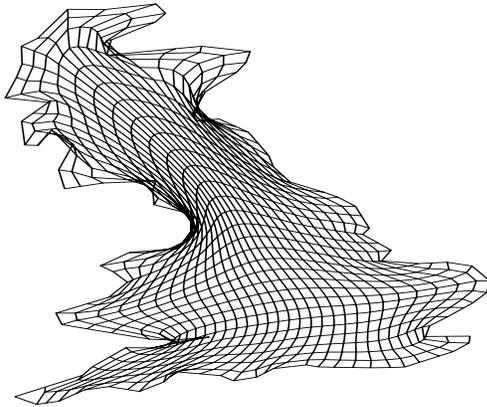


k -Suavidad-Área
 $\sigma = 0.5$ convexa

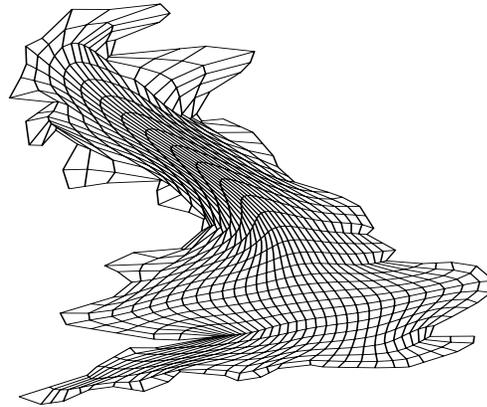


k -Suavidad-Área
 $\sigma = 0.75$ convexa

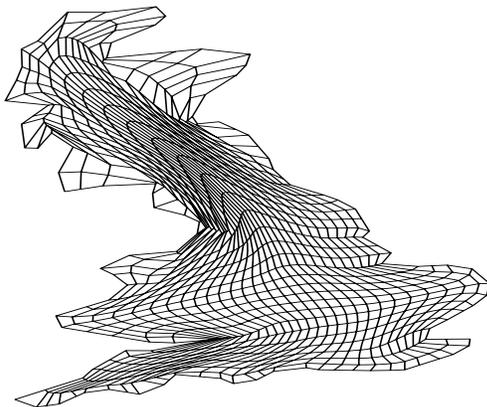
Figura C.2: Mallas generadas sobre la bahía de la habana.



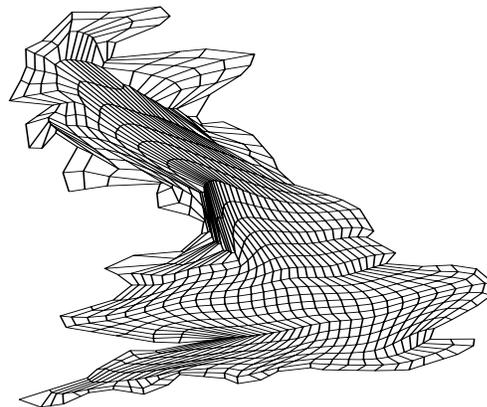
t-Área-Longitud
 $\sigma = 0.75$ no convexa



k-Área-Longitud
 $\sigma = 0.5$ convexa

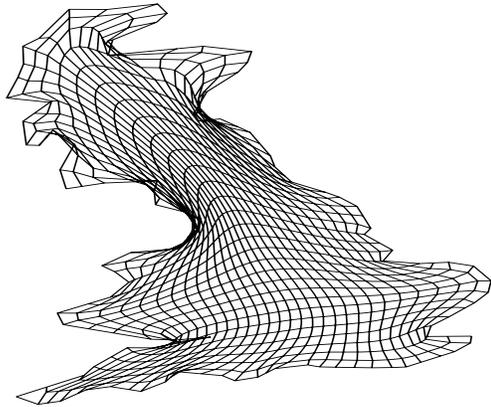


k-Área-Longitud
 $\sigma = 0.75$ convexa

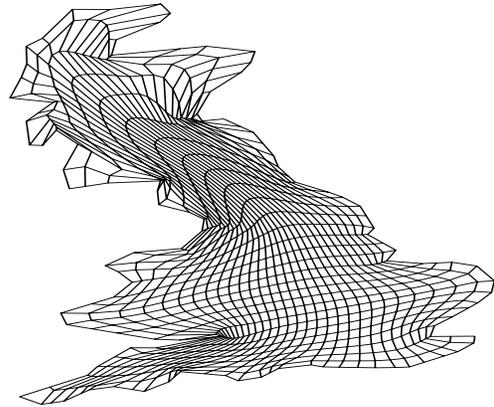


k-Área-Longitud
 $\sigma = 0.9$ convexa

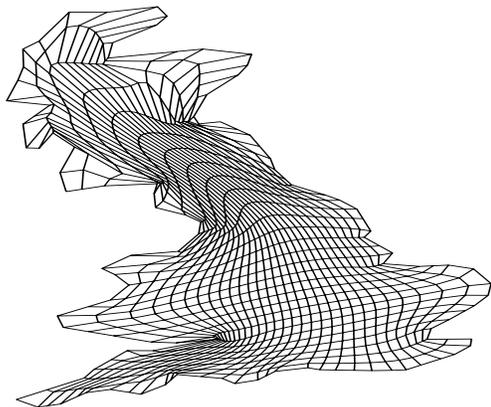
Figura C.3: Mallas generadas sobre el Reino unido.



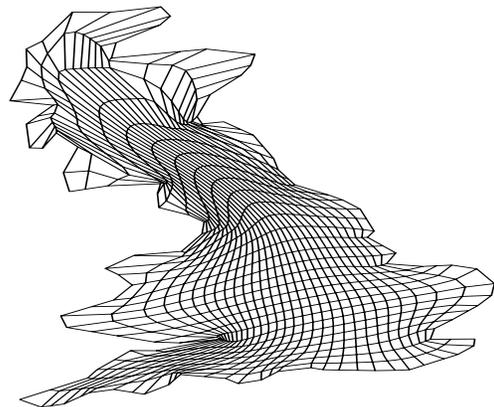
t-Área-Longitud
 $\sigma = 0.75$ no convexa



k-Suavidad-Área
 $\sigma = 0.25$ convexa

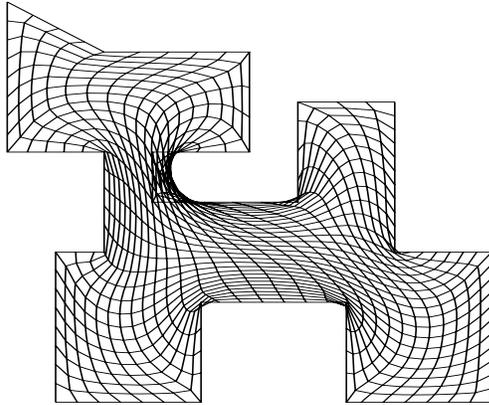


k-Suavidad-Área
 $\sigma = 0.5$ convexa

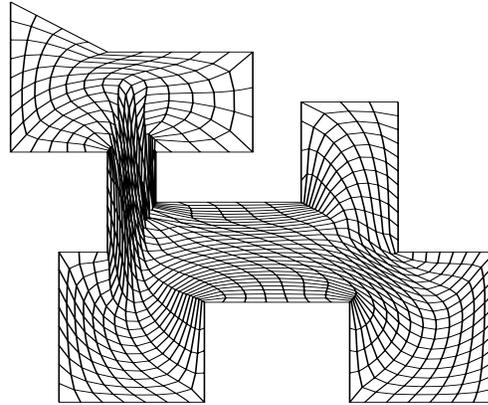


k-Suavidad-Área
 $\sigma = 0.75$ convexa

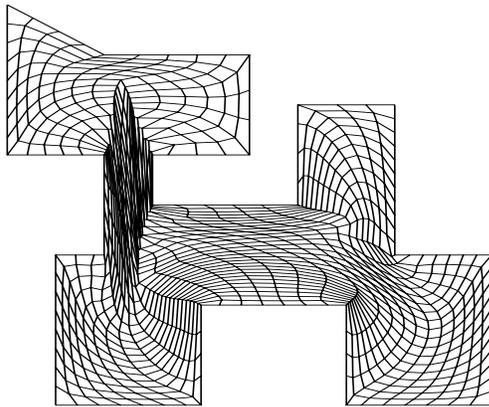
Figura C.4: Mallas generadas sobre el Reino unido.



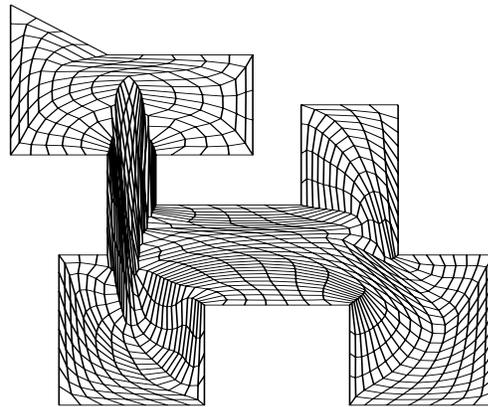
t-Área-Longitud
 $\sigma = 0.75$ no convexa



k-Área-Longitud
 $\sigma = 0.5$ convexa

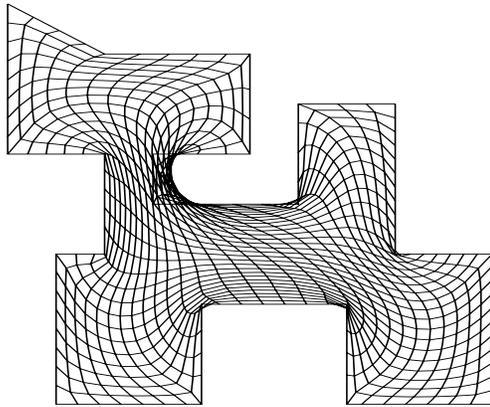


k-Área-Longitud
 $\sigma = 0.75$ convexa

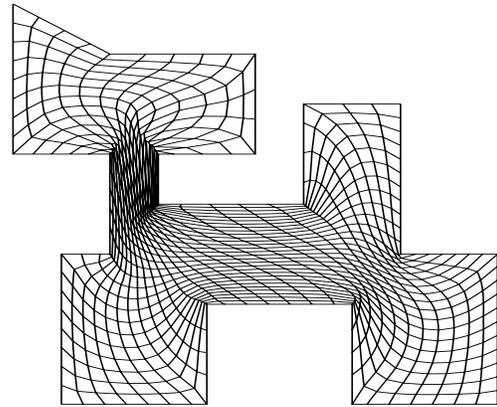


k-Área-Longitud
 $\sigma = 0.9$ convexa

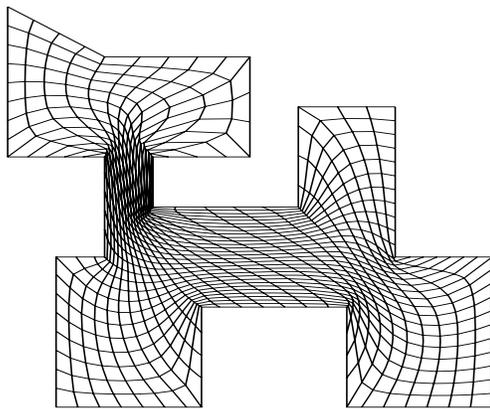
Figura C.5: Mallas generadas sobre la región del Gato.



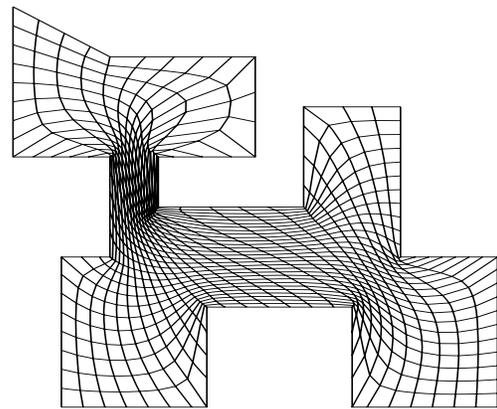
t -Área-Longitud
 $\sigma = 0.75$ no convexa



k -Suavidad-Área
 $\sigma = 0.25$ convexa

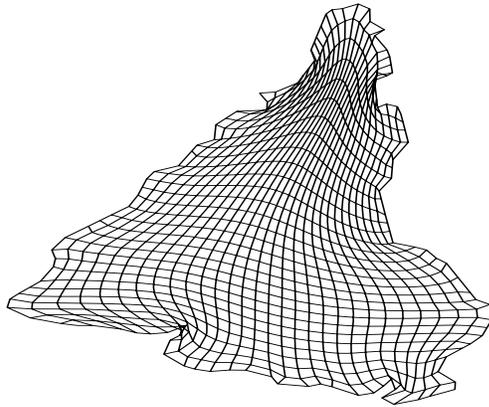


k -Suavidad-Área
 $\sigma = 0.5$ convexa

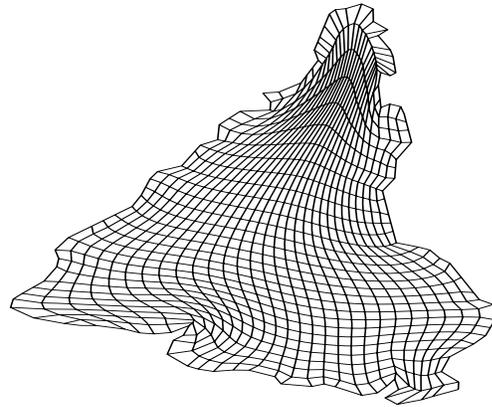


k -Suavidad-Área
 $\sigma = 0.75$ convexa

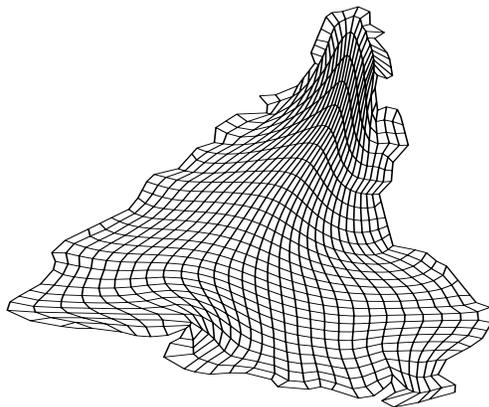
Figura C.6: Mallas generadas sobre la región del Gato.



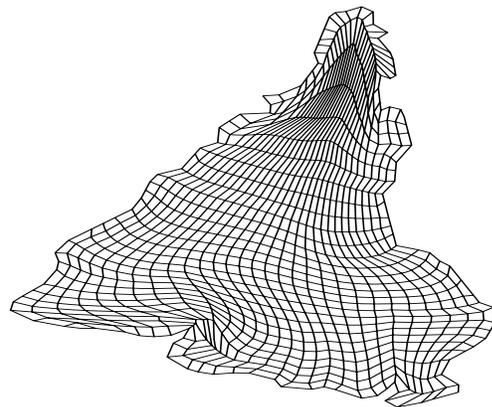
t -Área-Longitud
 $\sigma = 0.75$ no convexa



k -Área-Longitud
 $\sigma = 0.5$ convexa

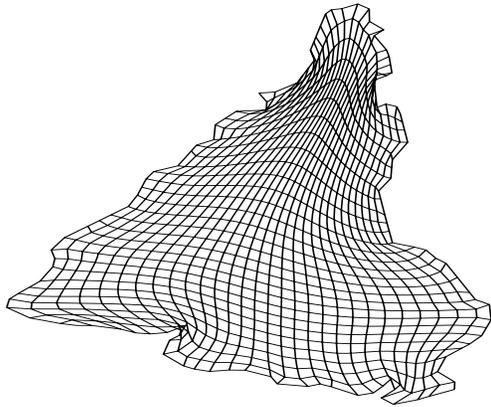


k -Área-Longitud
 $\sigma = 0.75$ convexa

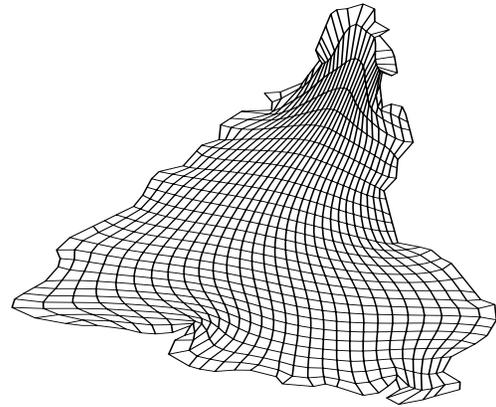


k -Área-Longitud
 $\sigma = 0.9$ convexa

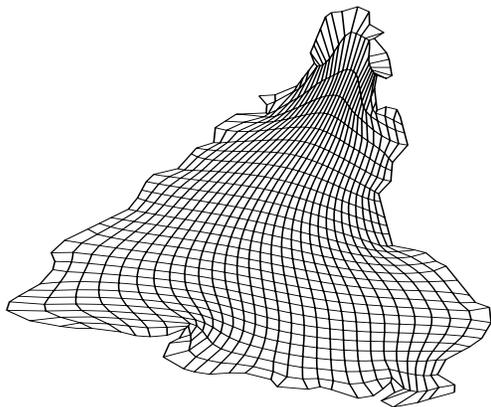
Figura C.7: Mallas generadas sobre la región de Sudamérica.



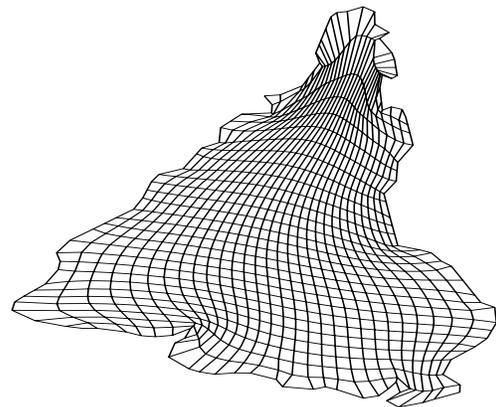
t -Área-Longitud
 $\sigma = 0.75$ no convexa



k -Suavidad-Área
 $\sigma = 0.25$ convexa

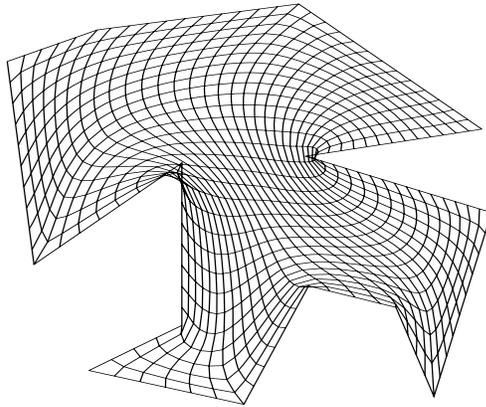


k -Suavidad-Área
 $\sigma = 0.5$ convexa

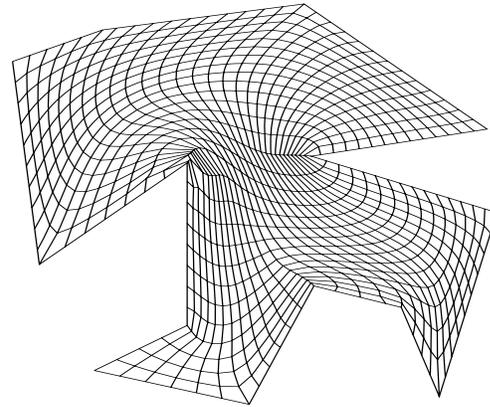


k -Suavidad-Área
 $\sigma = 0.75$ convexa

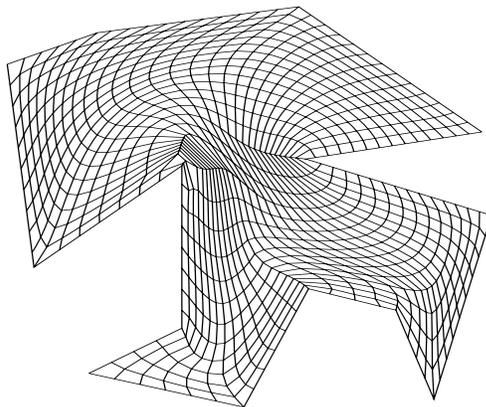
Figura C.8: Mallas generadas sobre la región de Sudamérica.



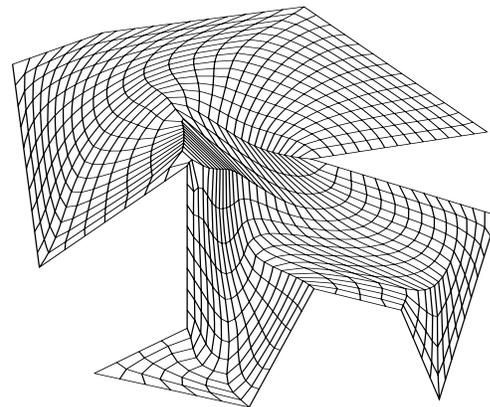
t -Área-Longitud
 $\sigma = 0.75$ no convexa



k -Área-Longitud
 $\sigma = 0.5$ convexa

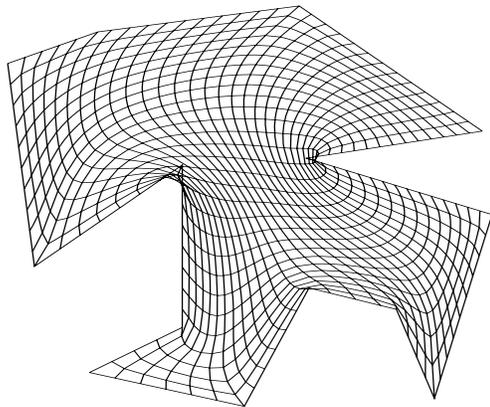


k -Área-Longitud
 $\sigma = 0.75$ convexa

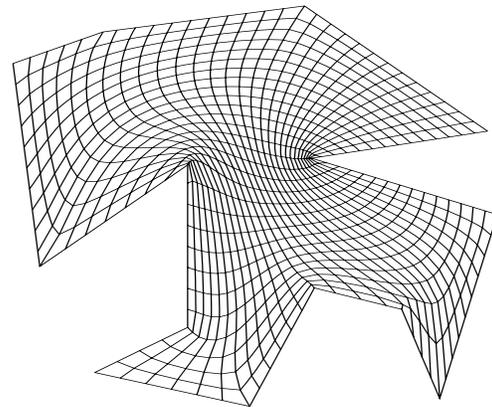


k -Área-Longitud
 $\sigma = 0.9$ convexa

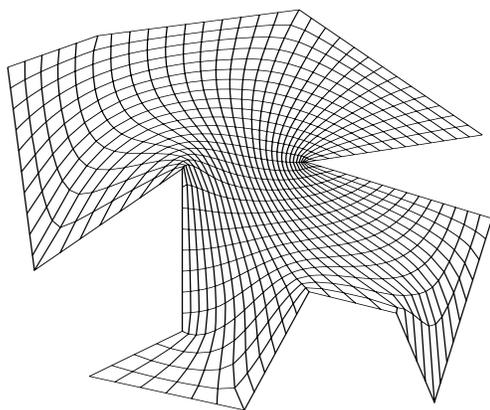
Figura C.9: Mallas generadas sobre la región del Búfalo.



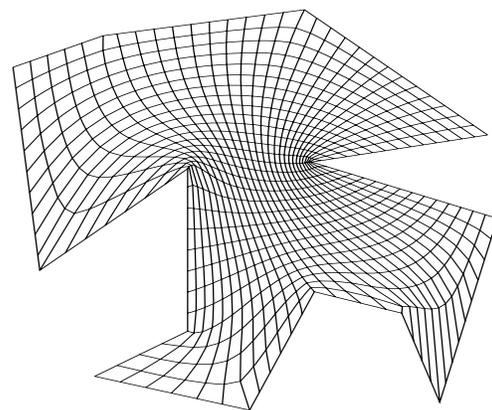
t -Área-Longitud
 $\sigma = 0.75$ no convexa



k -Suavidad-Área
 $\sigma = 0.25$ convexa

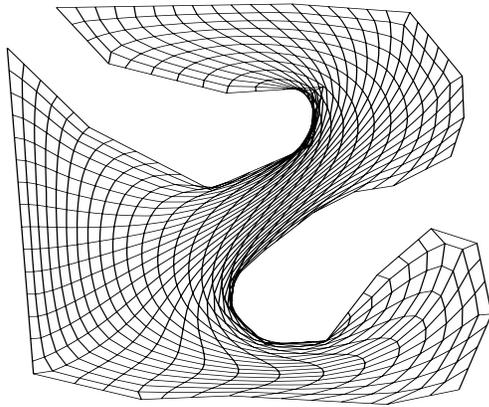


k -Suavidad-Área
 $\sigma = 0.5$ convexa

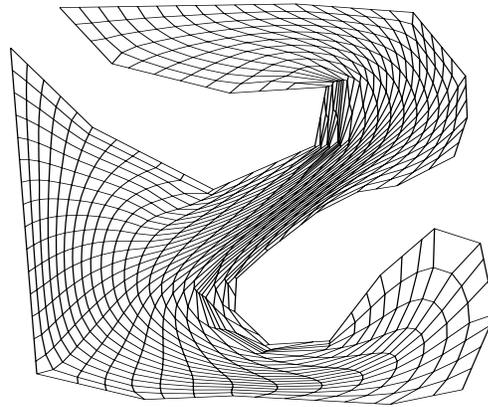


k -Suavidad-Área
 $\sigma = 0.75$ convexa

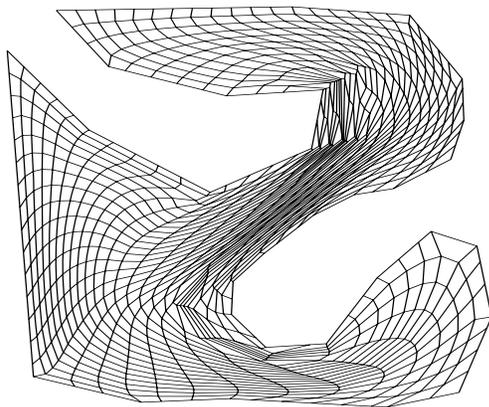
Figura C.10: Mallas generadas sobre la región del Búfalo.



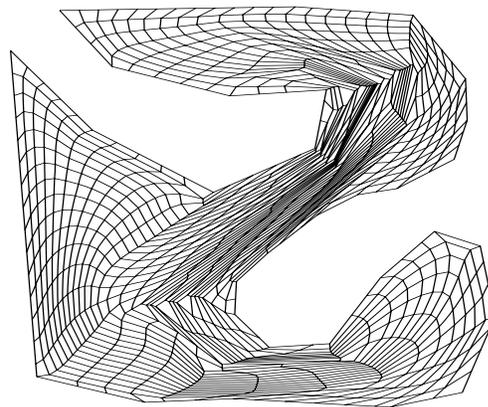
t-Área-Longitud
 $\sigma = 0.75$ no convexa



k-Área-Longitud
 $\sigma = 0.5$ convexa

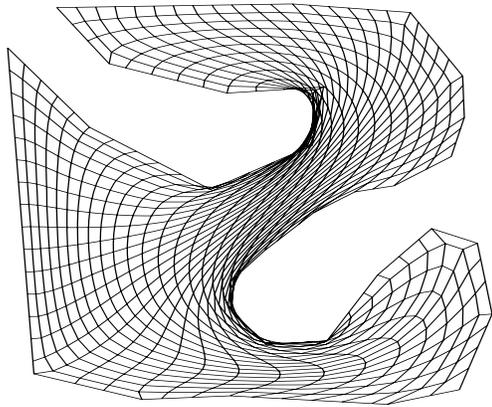


k-Área-Longitud
 $\sigma = 0.75$ convexa

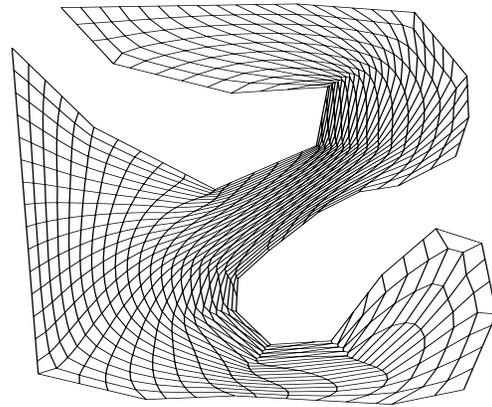


k-Área-Longitud
 $\sigma = 0.9$ convexa

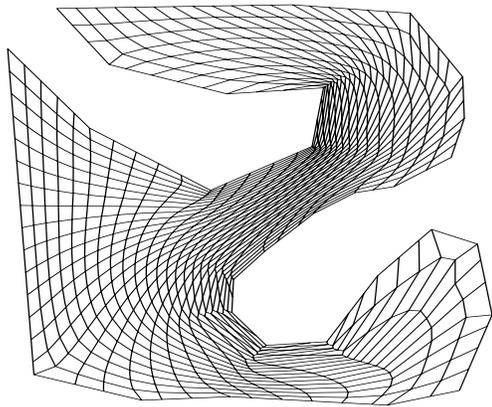
Figura C.11: Mallas generadas sobre la región del Cisne.



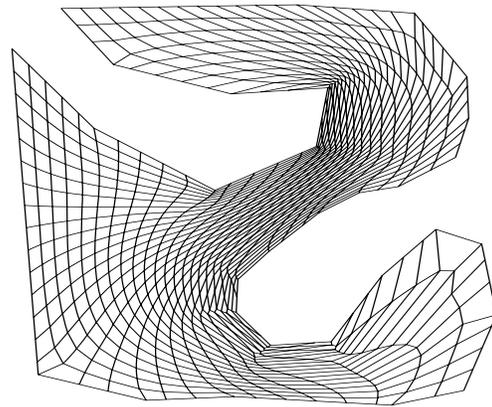
t -Área-Longitud
 $\sigma = 0.75$ no convexa



k -Suavidad-Área
 $\sigma = 0.25$ convexa

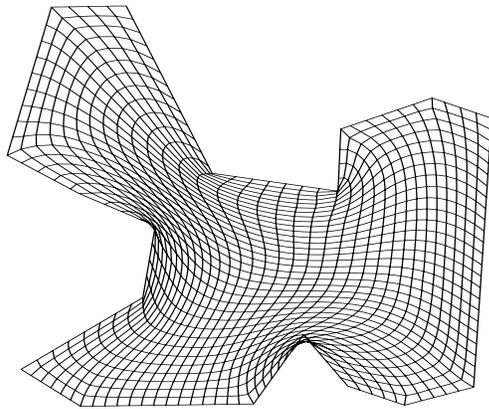


k -Suavidad-Área
 $\sigma = 0.5$ convexa

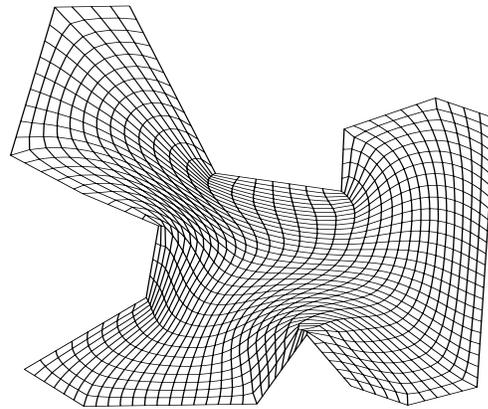


k -Suavidad-Área
 $\sigma = 0.75$ convexa

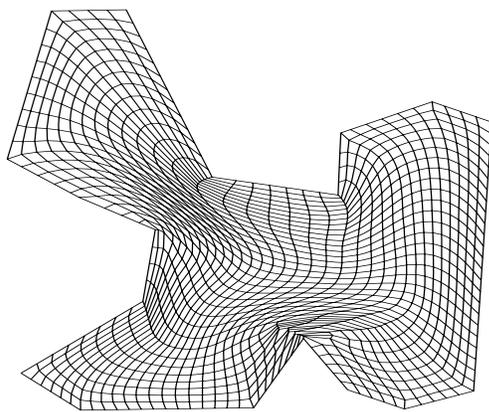
Figura C.12: Mallas generadas sobre la región del Cisne.



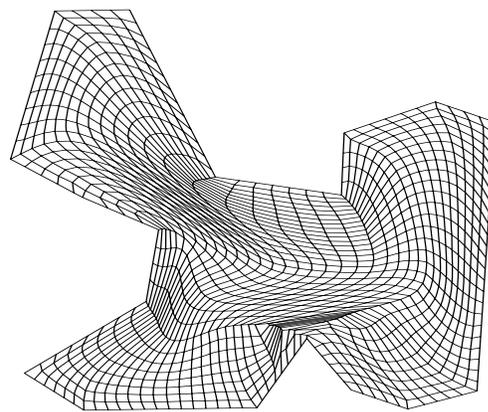
t-Área-Longitud
 $\sigma = 0.75$ no convexa



k-Área-Longitud
 $\sigma = 0.5$ convexa

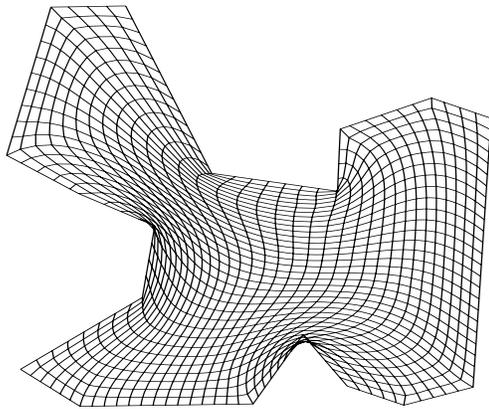


k-Área-Longitud
 $\sigma = 0.75$ convexa

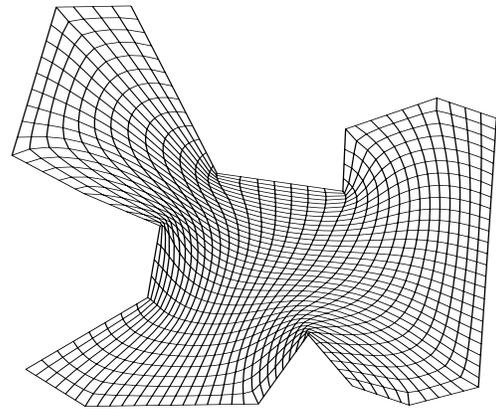


k-Área-Longitud
 $\sigma = 0.9$ convexa

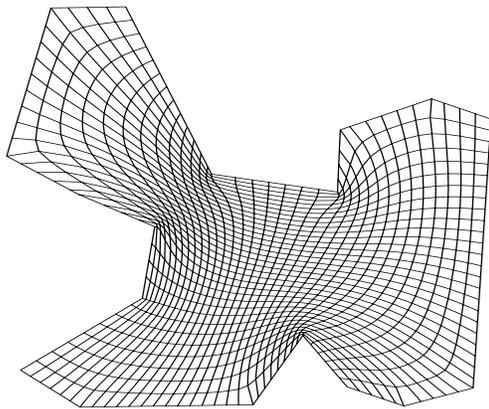
Figura C.13: Mallas generadas sobre la región del Ratón.



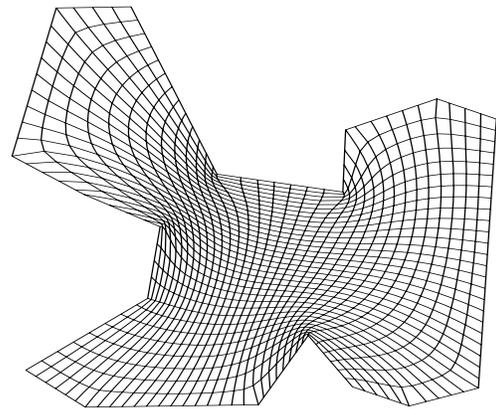
t-Área-Longitud
 $\sigma = 0.75$ no convexa



k-Suavidad-Área
 $\sigma = 0.25$ convexa



k-Suavidad-Área
 $\sigma = 0.5$ convexa



k-Suavidad-Área
 $\sigma = 0.75$ convexa

Figura C.14: Mallas generadas sobre la región del Ratón.

Bibliografía

- [1] Barrera, P. and J.E. Castillo. 1987. *A Large Scale Optimization problem arising from Numerical Grid Generation*, Tech. Rep. Dept. of Math. and Stat. Univ. of New Mexico.
- [2] Barrera, P. y Pérez, A. 1988. *Numerical Grid Generation* presentado en el III Congreso de la Sociedad Cubana de Matemáticas.
- [3] Barrera, P., Castellanos, N. and Pérez, A. 1989. *A New Method for Grid Generation*, 4th Workshop of Numerical Analysis, Mérida, Yuc.
- [4] Barrera, P., Castellanos, N. and Pérez, A. 1993. *Curvilinear Coordinate System Generation over Plane Irregular Regions*, Vínculos Matemáticos No. 133. Dept. de Matemáticas, Facultad de Ciencias, UNAM, México.
- [5] Barrera, P., Castellanos, L., y Pérez, A. 1994. *Métodos Variacionales Discretos para la Generación de Mallas*, DGAPA-UNAM, México.
- [6] Barrera, P., González, G., Pérez, A. y Castellanos, L. 1994. *Manual de Usuarios del Sistema UNAMALLA v. 1.0: Generación de Mallas Planas sobre Regiones Irregulares* DGAPA-UNAM, México.
- [7] Barrera, P., González, G., Rivera, A. y Ugalde, I. 1995. *Generación de Mallas en Regiones Planas Irregulares Usando Métodos Variacionales Continuos*, artículo auspiciado por el proyecto Triangulaciones Óptimas, No. IN103594, DGAPA, UNAM.
- [8] Barrera, P. and Tinoco J.G. 1997. *Smooth and Convex Grid Generation over General Plane regions* in Mathematics and Computer in Simulation.
- [9] Barrera, P., García, I. y González, G. 2000. *Manual Operativo del Sistema UNAMALLA v. 2.0 para PC*, Cuadernos de Investigación, **21** Área I, Física-Matemáticas e Ingeniería, Universidad Autónoma de Coahuila, México.
- [10] Birkhoff, G., and Lynch, R.E. 1984. *Numerical Solution of Elliptic Problems*, SIAM, Philadelphia.

- [11] Brackbill, J.U., and Saltzman, J.S. 1982. *Application on Generalizations of Variational Methodos for Generating Adaptive Meshes*, in *Numerical Grid Generation*, J.F. Thompson, ed. North-Holland, New York, 865–884.
- [12] Castellanos, J.L. *Generación Numérica de Redes usando Newton Truncado*. Tesis Doctoral, ICIMA, Ministerio de la Ciencia, Tecnología y el Medio Ambiente. La Habana, Cuba, 1994.
- [13] Castillo, J.E., 1986. *Mathematical Aspects of Numerical Grid Generation I. on Numerical Grid Generation in Computational Fluid Dynamics*. J.Hauser and C. Taylor eds., pp. 35–43.
- [14] Castillo, J.E., 1987. *On Variational Grid Generation*. Ph.D. thesis, University of Mexico, Alburquerque, New Mexico.
- [15] Castillo, J.E., Steinberg, S., and Roach, P.J. 1997. *Mathematical aspects of variational grid generation II*. *J. Comp. and Appl. Math.* **20**, pp. 127–135.
- [16] Castillo, J.E., 1991. *Mathematical aspects on Numerical grid generation*. SIAM Press.
- [17] Castillo, J.E., 1991. *Discrete Variational Grid Generation* in *Mathematical aspects on Numerical grid generation*. J.E. Castillo, ed. SIAM Press, pp. 35–58.
- [18] Dembo, R.S., and Steihaug, T. 1983. *Truncated Newton Algorithms for Large-Scale Unconstrained Optimization*, *Math. Prog.*, **26**, pp. 190–212.
- [19] Eiseman, P.R. 1982. *Numerical Grid Generation*, in *Numerical Grid Generation*, J.F. Thompson, ed., North-Holland, New York, pp. 193-226.
- [20] Gelfand, I.M. and Fomin, S.V. 1963. *Calculus of Variations*, Prentices–Hall, Englewood, Cliffs, N.J.
- [21] Fletcher, R. 1980. *Practical Methods of Optimization 1: Unconstrained Optimization*, John wiley, New York.
- [22] Forsythe, G.E., and Wasow, W.R. *Finite Difference Methods for Partial Differential Equations*, Wiley, New York.
- [23] García, I.D., 1999. *Métodos de Optimización de Gran Escala y algunas aplicaciones a Funciones Parcialmente Separables*. Tesis de Maestría, Facultad de Ciencias, UNAM, México.
- [24] González Flores, G.F., 1994. *Generación de Mallas en Regiones Planas Irregulares*. Tesis de Licenciatura. Universidad Autónoma de Yucatán, Yucatán, Méx. 1994.

- [25] García, I., 1994. *Mapea: Una colección de rutinas para la graficación usando el compilador de Fortran Lahey*. Notas de Clase. Facultad de Ciencias - UNAM.
- [26] Ivanenko, S.A., and Charkhch'yan, A.A., 1998. *Curvilinear Grids of Convex Aquadrilaterals*, J. Comp. Physics.
- [27] Ivanenko, S.A. *Adaptive-harmonic grid generation and its application for numerical solution of the problems with boundary and interior layers*, Comput. Math. Math. Phys., **35**(10), pp. 1203-1220.
- [28] Ivanenko, S.A., 1999. *Harmonic maps in Handbook of Grid Generation*, CRC Press Inc. 8.1-8.43.
- [29] Knupp, P., 1992. *A Robust Elliptic Grid Generator*. J. Comp. Phys. **100**, 409-418.
- [30] Knupp, P. and Steinberg, S. 1993. *Fundamentals of Grid Generation*. CRC Press, Inc.
- [31] Liao, G. 1991. On harmonic maps. In *Mathematical Aspects of Numerical Grid Generation*, J.E. Castillo, ed. SIAM, Philadelphia.
- [32] Liao, G. 1992. *Variational approach to grid generation*. Num. P.D.E's, **8**, pp. 143-147.
- [33] Mastin, C.W., and Thompson, J.F. 1978. *Elliptic Systems and Numerical Transformations*. J. Math. Anal. App., 62, pp. 52-62.
- [34] Mastin, C.W., 1982. *Error Induced by Coordinate System on Numerical Grid Generation*. J.F. Thompson, ed., pp. 31-40. North Holland, New York.
- [35] Moré, J.J. and Thuente D.J. 1990. *On line search algorithms with guaranteed sufficient decrease*. Math. and Computer Science Division Preprint MCS-P153-0590, Argonne National Laboratory, Argonne, Ill. USA.
- [36] Moretti, G. 1980. *Grid Generation using classical techniques*. on *Numerical Grid Generation Techniques*. R.E. Smith. ed., pp. 1-35. NASA CP 2166, NASA Langley Research Center, Hampton VA.
- [37] Ojeda, R.B. 1991. *Métodos Directos para la Generación de Redes en Regiones Planas*, Tesis de Maestría, Facultad de Ciencias, UNAM, México.
- [38] Owen, S. 2001. *An Introduction to Unstructured Mesh Generation. Part I: Meshing Algorithms* proceedings on 10th International Meshing Roundtable. Sandia National Laboratories. <http://www.andrew.cmu.edu/user/sowen/>

- [39] Saunders, B.V., 1985. *Algebraic Grid Generation using Tensor Product B-splines*. NASA Contractor Report 177968. NASA Langley Research Center, Hampton VA.
- [40] Smith, R.E. 1982. *Algebraic Grid Generation on Numerical Grid Generation*, J.F. Thompson, ed., pp. 137–170. North Holland. New York.
- [41] Soni, B.K., Thompson, J.F. and Weatherbill, N.P. 1999. *Handbook of Grid generation*, CRC Press, Inc.
- [42] Steinberg, S., and Roache, P.J., 1986. *Variational Grid Generation*, Num. Meth. for P.D.E.s., **2**, 71–96.
- [43] Steinberg, S., and Roache, P.J., 1990. *Anomalies in Grid Generation on Curves*, *J. Comp. Physics*, **91**, 255–277.
- [44] Steinberg, S., and Roache, P.J., 1992. *Variational Curve and Surface Grid Generation*, *J. Comp. Phys.*, **100**, 163–178.
- [45] Spekreijse, S.P., [1999]. *Elliptic Generation Systems in Handbook of Grid Generation*, Thompson, J.F., Soni, B.K. and Weatherill N.P. eds., CRC Press, Inc., 4.1–4.49.
- [46] Owen, S. *A survey of Unstructured Mesh Generation Technology*, Department of Civil and Environmental Engineering, Ansys, Inc. <http://www.andrew.cmu.edu/sowen/survey>.
- [47] Thompson, J.E., Warsi, Z.U.A., and Mastin, C.W., *Numerical Grid Generation: Foundations and Applications*, Elsevier, New York.
- [48] Tinoco, J.G. 1997. *Funcionales Discretos para la Generación de Mallas Suaves y Convexas sobre Regiones Planas Irregulares*, Tesis de Doctorado, CIMAT, México.
- [49] Tinoco, J.G. and Barrera, P. 1998. *Area functionals in plane Grid Generation in International conference on Numerical Grid Generation in Computational Field Simulation*. International Society of Grid Generation.
- [50] Tinoco, J.G., Barrera, P. and Cortés, A. 2001. *Some properties of Area in Numerical Grid Generation* Proceedings, 10th International Meshing Roundtable, Sandia National Laboratories, pp. 43-54, October 7-10.
- [51] Toledo, F. 1995. *Métodos Numéricos para Mapeo Conforme con Aplicación a la generación de Mallas*. Tesis de Maestría. Facultad de Ciencias, UNAM, México.
- [52] Van Vliet, H. *Software Engineering: Principles and practice*. John Wiley & Sons, England.

-
- [53] Winslow, Z.U.A., 1967. *Numerical Solution of the quasilinear poisson equations in a nonuniform triangle mesh*, J. Comput. Phys., **2**, pp. 149–172.
- [54] Warzi, Z.U.A., 1982. *Basic Differential Models for Coordinate Generation on Numerical Grid Generation*. J.F. Thompson, ed., pp. 41–78. North Holland.