



UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Mat. Luis Manuel Rivera Gutiérrez

UN ESQUEMA DE DIFERENCIAS FINITAS PARA LA SOLUCIÓN DE ECUACIONES DIFERENCIALES PARCIALES UTILIZANDO NUBES DE PUNTOS

TESIS

QUE PARA OBTENER EL TÍTULO DE:
Licenciado en Ciencias Físico Matemáticas

PRESENTA:

Nanci Pintor Lázaro

ASESORES:

Dr. José Gerardo Tinoco Ruiz.

Dr. Francisco Javier Domínguez Mota.

MORELIA MICHOACÁN, DICIEMBRE 2015.

Índice

Resumen	V
Abstract	VI
Prefacio	VII
1. Preliminares.	1
1.1. Conceptos.	1
1.2. Ecuaciones diferenciales parciales lineales de 2o. orden.	2
1.2.1. Condiciones asociadas.	3
2. Diferencias finitas.	5
2.1. Diferencias finitas en una dimensión.	5
2.1.1. Primer ejemplo.	9
2.1.2. Error de truncamiento local.	10
2.1.3. Error global.	10
2.1.4. Error de redondeo.	11
2.2. Diferencias finitas en un rectángulo.	11
2.2.1. Ejemplo en dos dimensiones	12
2.2.2. Consistencia.	14
2.2.3. Estabilidad.	15
2.2.4. Convergencia.	16
2.2.5. Segundo ejemplo en dos dimensiones: ecuación del calor.	16
3. Diferencias Finitas en Regiones Irregulares.	19
3.1. Generación de mallas.	19
3.1.1. Generación numérica de mallas estructuradas.	20
3.1.2. Un ejemplo de construcción por interpolación algebraica.	22
3.1.3. Generación Variacional Continua.	24
3.1.4. Generación Variacional Discreta	28
3.1.5. Funcionales de Suavidad y Área adaptativos de Tinoco.	34
3.1.6. Funcional Quasi-Armónico de Domínguez	38
3.2. Diferencias Finitas Generalizadas.	39

4. Nubes.	43
4.1. Criterios de elección de nodos.	49
4.1.1. Variantes del criterio de los cuatro cuadrantes.	50
4.2. Problemas en la elección de nodos.	52
5. Resultados numéricos y conclusiones.	55
5.1. Ecuaciones diferenciales parciales utilizadas.	55
5.2. Resultados.	57
5.2.1. Problemas tomados del trabajo de J. J. Benito. en las nubes del grupo de trabajo.	61
5.2.2. Ecuaciones de prueba del grupo de trabajo en las primeras 7 nubes	67
5.2.3. Nubes y problemas de prueba del equipo de trabajo	69
5.3. Conclusiones	75
A. Programas para la selección de nodos.	79
A.1. Primera variante	79
A.2. Segunda variante	90

Resumen

Mucha de la teoría de aproximación de la solución de ecuaciones diferenciales parciales (EDPs) se lleva a cabo en diferentes tipos de mallas. En muchas ocasiones la generación de la malla conlleva un gran costo computacional y es por eso que se ha comenzado a desarrollar métodos para evitar la generación de una malla al momento de resolver EDPs.

En este trabajo se presenta un esquema de solución de EDPs que utiliza nubes de puntos como base para conseguir la aproximación al resultado de diferentes tipos de EDPs como un primer estudio del método. Para el cual se utilizan dos implementaciones basadas en el método conocido como “criterio de los cuatro cuadrantes” propuesto por T. Lizka y J. Orkisz [13].

Palabras clave: *diferencias finitas, regiones irregulares, nubes de puntos, criterios de elección de nodos, métodos numéricos.*

Abstract

A lot of approximation theory for solution of partial differential equations (PDEs) is carried out in different types of grids. In many cases the grid generation carries with a great computational cost and for this, people have begun to develop methods for avoid the grid generation to the moment to solve PDEs.

This work presents a scheme for solution of PDEs that uses clouds of points as base to obtain the approximation to the result of different types of PDEs as a first study of the method. For it, we used two implementations of the method known as the method "four quadrant criterion" proposed by T. Lizka y J. Orkisz [13].

Prefacio.

El estudio del comportamiento de los fenómenos físicos de la naturaleza aparecen muy frecuentemente las ecuaciones diferenciales, uno de los problemas es que los esquemas que los describen no siempre tienen solución analítica ya que están basados en problemas idealizados. Es por ello que existen los métodos numéricos, que con ayuda de las computadoras se han ido desarrollando esquemas cada vez mejores, que nos ayudan a aproximar mejor las soluciones. Aunque estas no sean exactas nos ayudan a poder describir la evolución del fenómeno en estudio. Para la solución numérica de EDPs pueden emplearse muchos métodos numéricos como son diferencias finitas, elementos finitos, métodos espectrales, volúmenes finitos, elementos de frontera, por mencionar algunos. En este trabajo sólo se utiliza el método de diferencias finitas.

Existe mucha teoría para aproximar dichas soluciones pero la mayoría de los esquemas están desarrollados para mallas, pero adecuar una malla requiere un gran costo computacional cuando se trata de superficies irregulares, por lo que se han desarrollado otros métodos que no requieran una malla.

Este trabajo presenta un esquema de diferencias finitas, que propone aproximar en nubes de puntos, evitando así el uso de un mallado como un primer intento de buscar mejores aproximaciones con menor costo computacional.

El primer capítulo explica brevemente los conceptos básicos que se necesitaran en el desarrollo del mismo, como son: los tipos de ecuaciones diferenciales, ecuaciones diferenciales parciales de segundo orden y las condiciones asociadas para la solución de diferentes problemas.

En el segundo capítulo se hace una ligera introducción del método de diferencias finitas para 1 y 2 dimensiones y algunos ejemplos de cómo aplicar el método.

El tercer capítulo describe el esquema que se utilizó para regiones irregulares, así como una descripción de algunos de los métodos más conocidos en la generación numérica de mallas como una motivación para la búsqueda de otros métodos.

El cuarto capítulo se centra en las nubes de puntos pero sobre todo el desarrollo en los métodos para la elección de los nodos vecinos, describe las variantes del método del criterio de los cuatro cuadrantes que se utilizaron en este trabajo y los problemas que se presentaron en la elección de nodos.

El último capítulo contiene los resultados numéricos obtenidos en cada una de las nubes utilizadas de prueba con cada una de las variantes mencionadas anteriormente. Aquí también se presentan las conclusiones a las que se llegaron y propuestas para un trabajo a futuro.

Capítulo 1

Preliminares.

En este capítulo tiene como objetivo introducir algunos conceptos básicos de ecuaciones diferenciales que se utilizarán a lo largo del presente trabajo, así como describir brevemente el método de diferencias finitas.

1.1. Conceptos.

Iniciamos recordando con lo que es una ecuación diferencial. Una ecuación diferencial es aquella que relaciona las derivadas de una función que depende de una o más variables.

Usualmente las ecuaciones diferenciales se subdividen en *ecuaciones diferenciales parciales* (EDPs), por ejemplo:

$$u_{xx}(x, y) + u_{yy}(x, y) = 0, \quad u_{tt}(x, t) = u_{xx}(x, t) - u^3(x, y) \quad (1.1)$$

y *ecuaciones diferenciales ordinarias* (EDOs), por ejemplo

$$u'(t) = u(t), \quad u''(x) + u'(x) = x^2. \quad (1.2)$$

Las EDPs involucran derivadas parciales mientras que las EDOs solo involucran derivadas con respecto a una variable.

El *orden* de una ecuación diferencial se refiere a la derivada de mayor orden que aparece en la ecuación diferencial.

Otra importante distinción es entre las ecuaciones diferenciales lineales y no lineales. Para poder ver claramente esta diferencia es útil escribir la ecuación en la forma

$$L(u) = f \quad (1.3)$$

diremos que la ecuación de la forma (1.3) es lineal sí

$$L(\alpha u + \beta v) = \alpha L(u) + \beta L(v)$$

para α y β constantes y u, v funciones. Tomemos como ejemplo las ecuaciones de (1.1), empecemos por escribir la primera en la forma (1.3)

$$L(u) = u_{xx} + u_{yy}$$

$$\begin{aligned} L(u+v) &= u_{xx} + v_{xx} + u_{yy} + v_{yy} \\ &= u_{xx} + u_{yy} + v_{xx} + v_{yy} \\ &= L(u) + L(v) \end{aligned}$$

por lo tanto esta ecuación es lineal. Ahora veamos la segunda ecuación

$$L(u) = u_{tt} - u_{xx}(x, t) + u^3$$

$$\begin{aligned} L(u+v) &= u_{tt} + v_{tt} - (u_{xx} + v_{xx}) + (u+v)^3 \\ &= u_{tt} + u_{xx} + v_{tt} + v_{xx} + (u+v)^3 \\ &\neq L(u) + L(v) \end{aligned}$$

ya que $(u+v)^3 \neq u^3 + v^3$ la ecuación es no lineal.

1.2. Ecuaciones diferenciales parciales lineales de 2o. orden.

En la sección pasada se vio que existen diferentes tipos de ecuaciones diferenciales sin embargo en este trabajo sólo nos enfocamos en lo que son EDPs lineales de 2o.orden ya que los problemas clásicos que estaremos estudiando como lo son la ecuación de Laplace/Poisson y la ecuación del calor entre otras son de este tipo de ecuaciones.

Para una función $u(x, y)$ de clase C^2 , la ecuación diferencial parcial (EDP) lineal de segundo orden es de la forma

$$\begin{aligned} L(u) &= Au_{xx} + Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu \\ L(u) &= f \end{aligned} \tag{1.4}$$

donde A, B, C, D, E, F y f son funciones suaves de (x, y) definidas en un conjunto abierto Ω .

Las EDPs de segundo orden se clasifican en tres categorías dependiendo del valor de Q , que es la matriz simétrica asociada a la parte cuadrática de la ecuación (*i.e.* las derivadas de segundo orden de u) y está definida como sigue

$$Q = \begin{pmatrix} A & B \\ B & C \end{pmatrix}.$$

Una EDPs es:

1.2. ECUACIONES DIFERENCIALES PARCIALES LINEALES DE 2O. ORDEN.3

elíptica	si	$\det(Q) > 0$,
parabólica	si	$\det(Q) = 0$,
hiperbólica	si	$\det(Q) < 0$.

■ Ecuaciones elípticas.

Las ecuaciones de este tipo llevan a problemas de procesos a ciclo fijo, cuando la función buscada no depende del tiempo.

El representante típico de éstas ecuaciones es la ecuación de Laplace

$$\Delta u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad u = u(x, t).$$

■ Ecuaciones parabólicas.

Las ecuaciones de este tipo llevan a problemas de conductividad térmica y de difusión.

La ecuación más simple de conductividad térmica tiene la forma

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}, \quad u = u(x, t)$$

donde $a^2 = \frac{K}{c\rho}$, donde ρ es la densidad del medio, c es el calor específico y K es el coeficiente de conductividad térmica.

■ Ecuaciones hiperbólicas.

Estas ecuaciones llevan a problemas de fenómenos de tipo oscilatorio (problemas de la cuerda, oscilaciones electromagnéticas, etc.). La particularidad característica de los procesos descritos por las ecuaciones de tipo hiperbólico es la velocidad finita de propagación de las perturbaciones.

Como ejemplo más concreto de este tipo de ecuaciones es *la ecuación de onda*

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \quad u = u(x, t)$$

donde x es la coordenada espacial, t el tiempo y c es la velocidad de propagación de las perturbaciones.

1.2.1. Condiciones asociadas.

Las EDPs tienen una infinidad de soluciones. Para obtener una solución única de la EDP, debemos agregar condiciones adicionales dependiendo del tipo de EDP que se quiera solucionar.

Condiciones de frontera.

Estas condiciones indican el comportamiento de la solución del problema en la frontera del dominio en consideración. Hay tres principales tipos de problemas con valores en la frontera.

Problema de Dirichlet Este tipo de problema es conocido como de valores en la frontera. En las ecuaciones de segundo orden se necesitan dos condiciones por cada variable sobre la solución $u(x, y)$. Suponiendo que se busca la solución para $a \leq x \leq b$ y $c \leq y \leq d$, las condiciones son

$$\begin{aligned} u(a, y) &= f_1(x, y), & u(b, y) &= f_2(x, y), \\ u(x, c) &= f_3(x, y), & u(x, d) &= f_4(x, y). \end{aligned}$$

De manera general, el problema de Dirichlet nos da el valor de la función incógnita u en toda la frontera de la región de interés.

Problema de Neumann Este es otro tipo de problema de valores en la frontera. Suponiendo que se busca la solución para $a \leq x \leq b$ y $c \leq y \leq d$, las condiciones para este caso son

$$\begin{aligned} u_x(a, y) &= f_1(x, y), & u_x(b, y) &= f_2(x, y), \\ u_y(x, c) &= f_3(x, y), & u_y(x, d) &= f_4(x, y). \end{aligned}$$

De manera general, el problema de Neumann nos da el valor de la derivada normal de la función incógnita en toda la frontera de la región de interés.

Problema de Robin Este problema de valores en la frontera generaliza los dos anteriores. Suponiendo que se busca la solución para $a \leq x \leq b$ y $c \leq y \leq d$, las condiciones son

$$\begin{aligned} \alpha_1 u(x, y) + \beta_1 u_x(a, y) &= f_1(x, y), & \alpha_2 u(x, y) + \beta_2 u_x(b, y) &= f_2(x, y), \\ \alpha_3 u(x, y) + \beta_3 u_y(x, c) &= f_3(x, y), & \alpha_4 u(x, y) + \beta_4 u_y(x, d) &= f_4(x, y). \end{aligned}$$

donde α_i y β_i son constantes dadas. De manera general, el problema de Robin nos da el valor de una combinación lineal de la función incógnita u y su derivada normal en toda la frontera de la región de interés.

Condiciones iniciales.

Las EDPs que modelan procesos en los cuales el tiempo es una de las variables independientes, la solución debe ser especificada por una o varias condiciones iniciales. El número de condiciones iniciales requeridas depende de la derivada de mayor orden en el tiempo que aparece en la ecuación.

La combinación de las condiciones iniciales y las condiciones de frontera en una EDP es lo que se llama un *problema de contorno*.

Capítulo 2

Diferencias finitas.

Las EDPs modelan muchos fenómenos físicos por ello son muy utilizadas, las EDPs que se pueden resolver explícitamente con fórmulas analíticas son muy pocas; es así como surge la necesidad de introducir métodos numéricos para encontrar una aproximación apropiada. El desarrollo de los esquemas de aproximación numérica es una herramienta esencial para obtener información cuantitativa así como la comprensión del posible comportamiento de sus soluciones[14].

El método de diferencias finitas tiene como principal objetivo remplazar los términos de la ecuación diferencial que involucran diferenciación por términos más sencillos que se mostraran más adelante para obtener un sistema algebraico.

2.1. Diferencias finitas en una dimensión.

Iniciaremos el desarrollo de este método con las fórmulas básicas de diferencias finitas para aproximar derivadas de primer orden.

Consideramos una función de una variable $u(x)$, asumimos que es suave y derivable las veces que sean necesarias; cada derivada estará bien definida sobre un intervalo que contendrá un punto de interés particular x_0 [12].

Supongamos que queremos aproximar $u'(x_0)$ por una diferencia finita basada solo en los valores de u en un número finito de puntos cerca de x_0 . Podemos usar

$$D_+u(x_0) \approx \frac{u(x_0 + h) - u(x_0)}{h} \quad (2.1)$$

para algun valor pequeño de $h = \Delta x$. Note que $D_+u(x_0)$ es la pendiente de la línea que interpola a u en los puntos x_0 y $x_0 + h$. La expresión (2.1) es una aproximación unilateral para u' ya que u es evaluada solo en valores de $x \geq x_0$. Otra aproximación unilateral sería

$$D_-u(x_0) \approx \frac{u(x_0) - u(x_0 - h)}{h}. \quad (2.2)$$

En cada una aproximación para $u'(x_0)$, el tamaño del error de aproximación es $\mathcal{O}(h)$ donde la notación se refiere al término que es proporcional a h , o más precisamente cuyo valor absoluto es acotado por una constante múltiplo de $|h|$ cuando $h \rightarrow 0$, más adelante se explicará mejor con el teorema de Taylor (2.1).

Otra posibilidad es usar la aproximación centrada

$$D_0u(x_0) \approx \frac{u(x_0 + h) - u(x_0 - h)}{2h} = \frac{1}{2}(D_+u(x_0) + D_-u(x_0)), \quad (2.3)$$

esta es la pendiente de la línea que interpola u en los puntos $x_0 - h$ y $x_0 + h$. $D_0u(x_0)$ nos da una mejor aproximación que las otras aproximaciones unilaterales ya que el error de aproximación es $\mathcal{O}(h^2)$ es decir proporcional a h^2 .

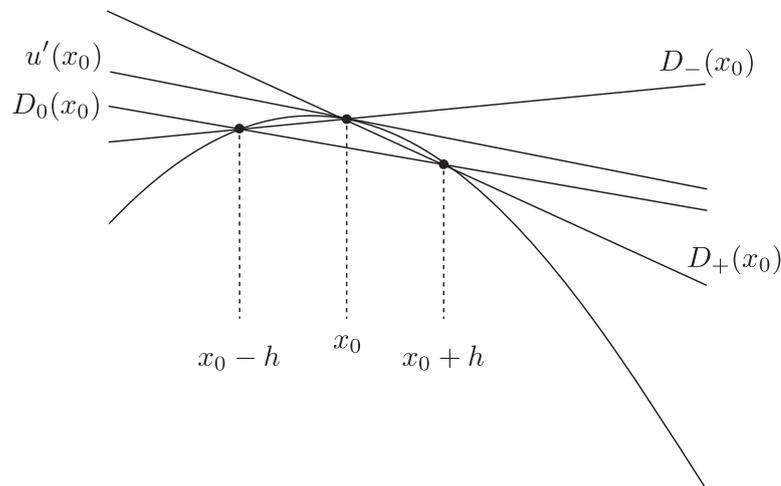
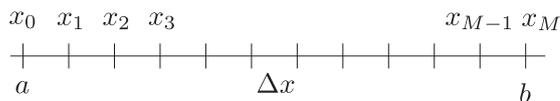


Figura 2.1: Aproximaciones de $u'(x_0)$.

Este método de aproximación numérica se lleva a cabo de la siguiente manera:

1. Generamos un número finito de puntos del dominio, en los cuales buscaremos aproximar la solución de la ecuación diferencial, por ejemplo; tomaremos el intervalo $[a, b]$ de la recta real y dividimos con M subintervalos de ancho $\Delta x = h = \frac{b-a}{M}$. A cada uno de los puntos x_i en los que empieza o termina el subintervalo le llamaremos nodo, los cuales numeramos de forma natural iniciando por la cota inferior del intervalo. Este procedimiento es llamado *mallado del dominio* y es la forma más sencilla de generar un mallado ver figura 2.2.

Figura 2.2: Mallado uniforme sobre el intervalo $[a, b]$.

2. Sustituimos las derivadas correspondientes con alguna de las fórmulas de diferencias finitas.
3. Resolvemos el sistema de ecuaciones para obtener la solución aproximada en cada punto.

Las fórmulas de diferencias finitas solo nos muestran la aproximación de la primera derivada, pero ¿cómo aproximamos las derivadas de orden mayor? Esto es relativamente sencillo. Todo parte del teorema de Taylor para expandir una función continua en términos de las derivadas en puntos dentro de una vecindad. Veamos que dice el teorema de Taylor.

Teorema 2.1 (de Taylor). *Sea $u(x)$ una función continua definida en el intervalo $[a, b]$ que además cumple la propiedad de pertenecer a la clase de funciones infinitamente diferenciables C^∞ sobre (a, b) ; dado un punto $x_0 \in (a, b)$, entonces se tiene que para cualquier otro punto $x \in (a, b)$,*

$$u(x) = u(x_0) + \frac{du(x_0)}{dx}(x-x_0) + \frac{d^2u(x_0)}{dx^2} \frac{(x-x_0)^2}{2!} + \dots + \frac{d^{N-1}u(x_0)}{dx^{N-1}} \frac{(x-x_0)^{N-1}}{(N-1)!} + R_N \quad (2.4)$$

donde

$$R_N = \frac{d^N u(\xi)}{dx^N} \frac{(x-x_0)^N}{(N)!}, \quad \xi \in (a, b)$$

es el residuo [19].

Ahora asumimos que $u(x)$ cumple con las hipótesis del teorema de Taylor, usamos su expansión alrededor del punto x_{i+1} en el intervalo $[a, b]$ para obtener

$$u(x_{i+1}) = u(x_i) + \frac{du}{dx}(x_i)(x_{i+1} - x_i) + \mathcal{O}(\Delta x_i^2),$$

en donde $\mathcal{O}(\Delta x_i^2)$ es el residuo de orden 2 y $\Delta x_i = x_{i+1} - x_i$. De ahí se obtiene la expresión

$$\frac{du}{dx}(x_i) = \frac{u(x_{i+1}) - u(x_i)}{(x_{i+1} - x_i)} + \mathcal{O}(\Delta x_i), \quad (2.5)$$

como u pertenece a C^∞ , podemos aplicar nuevamente el teorema de Taylor y aproximar la segunda derivada,

$$\frac{du}{dx}(x_{i+1}) \approx \frac{du}{dx}(x_i) + \frac{d^2u}{dx^2}(x_i)(x_{i+1} - x_i),$$

sustituimos la primera derivada en los respectivos puntos y obtenemos

$$\frac{d^2u}{dx^2}(x_i) \approx \frac{u(x_{i+2}) - 2u(x_{i+1}) + u(x_i)}{(x_{i+1} - x_i)^2}.$$

Este proceso lo podemos repetir para obtener derivadas de mayor orden. Notemos que la aproximación (2.5) es la llamada diferencia hacia adelante (2.1). El caso de las diferencias centradas es un poco más laborioso, ya que para obtenerlas primero calculamos las expresiones hasta orden 2, para u_{i+1} y u_{i-1}

$$\begin{aligned} u_{i+1} &= u_i + \frac{du}{dx}(x_i)(x_{i+1} - x_i) + \frac{1}{2} + \frac{d^2u}{dx^2}(x_i)(x_{i+1} - x_i)^2 + \mathcal{O}(\Delta x_i^3) \\ u_{i-1} &= u_i + \frac{du}{dx}(x_i)(x_{i-1} - x_i) + \frac{1}{2} \frac{d^2u}{dx^2}(x_i)(x_{i-1} - x_i)^2 + \mathcal{O}(\Delta x_i^3) \end{aligned}$$

después se resta la segunda de la primera

$$u_{i+1} - u_{i-1} = \frac{du}{dx}(x_i)(x_{i+1} - x_{i-1}) + \frac{1}{2} \frac{d^2u}{dx^2}(x_i)[(x_{i+1} - x_i)^2 - (x_{i-1} - x_i)^2] + \mathcal{O}(\Delta x^3),$$

así llegamos a la siguiente expresión de diferencias centradas

$$\frac{du_i}{dx} = \frac{u_{i+1} - u_{i-1}}{x_{i+1} - x_{i-1}} + \mathcal{O}(\Delta x_i^2). \quad (2.6)$$

Para el caso de una malla uniforme donde

$$x_{i+1} - x_i = x_i - x_{i-1}$$

la ecuación (2.6) se reduce a

$$\frac{du}{dx}(x_i) = \frac{u_{i+1} - u_{i-1}}{2\Delta x} + \mathcal{O}(\Delta x^2)$$

la cual obtiene un grado de aproximación mayor.

entonces el vector error E está definido por

$$E = U - \widehat{U}$$

que contiene los errores en cada punto. Nuestro objetivo es ahora es obtener una cota para la magnitud de este vector y mostrar que es $\mathcal{O}(h^2)$ cuando $h \rightarrow 0$. Para medir la magnitud de este vector debemos usar alguna norma, por ejemplo

$$\|E\|_\infty = \max_{1 \leq i \leq m} |E_i| = \max_{1 \leq i \leq m} |U(x_i) - u(x_i)|.$$

Si mostramos que $\|E\|_\infty = \mathcal{O}(h^2)$, entonces el error en cada punto debe ser también $\mathcal{O}(h^2)$. Podemos utilizar otras normas como la norma 1 y la norma 2 que podemos ver en [12].

Regresando al problema de calcular el error en nuestra solución al problema de valores en la frontera. La técnica que usaremos es básica para el análisis del método de diferencias finitas en general. Primero calculamos el *error de truncamiento local* del método y después usamos alguna tipo de *estabilidad* para mostrar que el *error global* puede ser acotado en términos del error de truncamiento local.

2.1.2. Error de truncamiento local.

El error de truncamiento local tomando el ejemplo anterior 2.1.1 está definido por

$$\tau_i = \frac{1}{h^2}(u(x_{i-1}) - 2u(x_i) + u(x_{i+1})) - f(x_i) \quad (2.9)$$

para $i = 1, \dots, m$. Si definimos τ como el vector con componentes τ_j entonces podemos escribirlo de la siguiente manera

$$\tau = A\widehat{U} - F$$

donde \widehat{U} es el vector solución del ejemplo anterior.

Por ejemplo, en la serie de Taylor el *error de truncamiento* se genera cuando se toma un número finito de términos de la serie, como lo hicimos para obtener los esquemas de diferencias, el error de truncamiento en este caso es R_N .

2.1.3. Error global.

El error global lo definimos como

$$E = U - \widehat{U}. \quad (2.10)$$

la relación entre el error de truncamiento local y el error global como

$$AE = -\tau.$$

2.1.4. Error de redondeo.

El *error de redondeo* generalmente ocurre por causa de las limitaciones del equipo de cómputo que estemos utilizando, se reduce el número de cifras significativas de su representación decimal para facilitar uso; sin embargo, dado que el número que estamos usando no es el número real, los cálculos que se hagan con esta representación no serán estrictamente los correctos.

2.2. Diferencias finitas en un rectángulo.

Anteriormente se vieron los esquemas de diferencias finitas en una dimensión por otra parte ¿qué pasa en dos dimensiones? nada inusual, se sigue el procedimiento anterior solo que ahora las derivadas parciales implican una derivada en una dirección.

En una región rectangular como la de la figura 2.2 definida en $[a, b] \times [c, d]$, en la cual se construyen los nodos interiores como

$$\begin{aligned}x_i &= a + i\Delta x, \\y_j &= c + j\Delta y,\end{aligned}$$

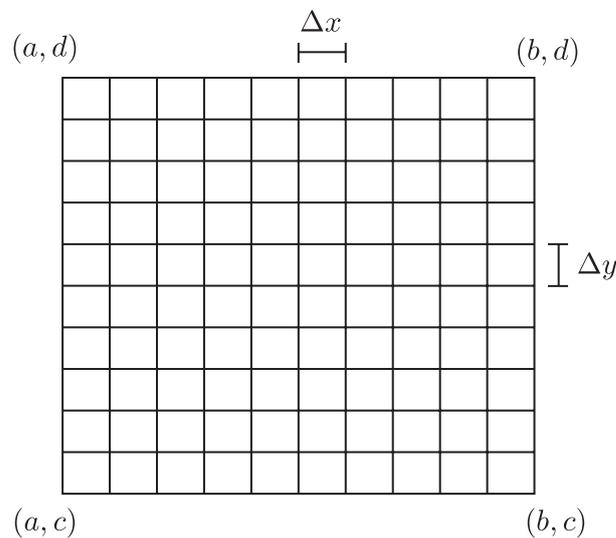


Figura 2.3: Mallado uniforme sobre la región rectangular $[a, b] \times [c, d]$.

donde Δx y Δy son el tamaño de paso horizontal y vertical respectivamente. Siguiendo el procedimiento anterior escribimos las fórmulas para cada una de las variables utilizando diferencias centradas.

$$\frac{\partial u}{\partial x}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - u(x_{i-1}, y_j)}{2\Delta x} + \mathcal{O}(\Delta x^2)$$

$$\frac{\partial u}{\partial y}(x_i, y_j) = \frac{u(x_i, y_{j+1}) - u(x_i, y_{j-1})}{2\Delta y} + \mathcal{O}(\Delta y^2)$$

Las derivadas de 2o. orden siguen el procedimiento antes visto, y sus expresiones son las siguientes:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(x_{i-1}, y_j) - 2u(x_i, y_j) + u(x_{i+1}, y_j)}{(\Delta x)^2},$$

$$\frac{\partial^2 u}{\partial y^2} \approx \frac{u(x_i, y_{i-1}) - 2u(x_i, y_j) + u(x_i, y_{i+1})}{(\Delta x)^2},$$

$$\frac{\partial^2 u}{\partial x \partial y} \approx \frac{u(x_{i+1}, y_{j+1}) - u(x_{i-1}, y_{j+1}) + u(x_{i-1}, y_{j-1}) - u(x_{i+1}, y_{j-1})}{4\Delta x \Delta y}.$$

2.2.1. Ejemplo en dos dimensiones

Para hacer más explícito lo dicho anteriormente, consideremos un caso particular *el problema de Poisson*

$$u_{xx} + u_{yy} = f, \quad (2.11)$$

en el cuadrado unitario $0 \leq x \leq 1$, $0 \leq y \leq 1$ y suponemos que tiene condiciones de frontera de Dirichlet. Usaremos una malla uniforme de puntos (x_i, y_j) , donde $x_i = i\Delta x$ y $y_j = j\Delta y$.

Para simplificar notación usaremos u_{ij} para una aproximación de $u(x_i, y_j)$. Para discretizar (2.11) remplazaremos las derivadas de x y y con las diferencias finitas centradas, y obtenemos

$$\frac{1}{(\Delta x)^2}(u_{i-1,j} - 2u_{ij} + u_{i+1,j}) + \frac{1}{(\Delta y)^2}(u_{i,j-1} - 2u_{ij} + u_{i,j+1}) = f_{ij}. \quad (2.12)$$

Este esquema se representa con el estencil de 5 puntos mostrado en la figura 2.4. Desconocemos u_{ij} para los m^2 puntos de la malla donde $i = 1, 2, \dots, m$ y $j = 1, 2, \dots, m$ y $h = 1/(m+1)$ como en la figura 2.3; entonces tenemos un

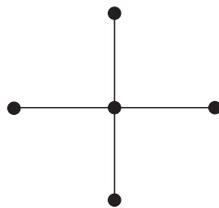


Figura 2.4: Estencil de 5 puntos.

sistema lineal de m^2 variables. Las ecuaciones de diferencias en los puntos cerca de la frontera contienen valores conocidos de la frontera, los cuales pueden moverse de lado derecho de la ecuación.

Si colocamos todas estas ecuaciones juntas en una ecuación matricial, tendremos una matriz de $m^2 \times m^2$, en esta matriz muchos de sus elementos son cero. Ya que cada ecuación tiene a lo más cinco variables, cada renglón de la matriz tiene a lo más cinco elementos distintos de cero y por lo tanto tenemos mínimo $m^2 - 5$ elementos que son cero.

El vector de variables está dividido de la siguiente forma

$$u = \begin{bmatrix} u^{[1]} \\ u^{[2]} \\ \vdots \\ u^{[m]} \end{bmatrix}, \quad \text{donde } u^{[j]} = \begin{bmatrix} u_{1j} \\ u_{2j} \\ \vdots \\ u_{mj} \end{bmatrix}. \quad (2.13)$$

Esto nos da una ecuación matricial donde A tiene la forma

$$A = \frac{1}{h^2} \begin{bmatrix} T & I & & & \\ I & T & I & & \\ & I & T & I & \\ & & \ddots & \ddots & \ddots \\ & & & I & T \end{bmatrix}, \quad (2.14)$$

la cual es una matriz tridiagonal por bloques de $m \times m$ en la cual cada bloque T , I por si mismos son una matriz de $m \times m$,

$$T = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & 1 & -4 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -4 \end{bmatrix}$$

y I es la matriz identidad de $m \times m$. Así tenemos la ecuación matricial

$$Au = \mathcal{F}$$

donde ahora \mathcal{F} es el vector con los valores de f_{ij} .

2.2.2. Consistencia.

La ecuación de diferencias es consistente con una ecuación diferencial parcial si el error de truncamiento local tiende a cero cuando los tamaños de los pasos en la malla de puntos se aproximan a cero, es decir;

$$\tau \longrightarrow 0, \text{ cuando } h \longrightarrow 0.$$

Cuando los errores de truncamiento local de las aproximaciones de las EDPs exactas son conocidos, la comprobación de la consistencia es directa. En cambio, cuando estos errores no se conocen, se debe analizar la ecuación de diferencias para la consistencia. Esto se hace expresando cada término en la ecuación de diferencias por su desarrollo de la serie de Taylor alrededor del nodo (x_i, y_j) . La ecuación resultante, puede ser entonces simplificada para proporcionar la forma del error de truncamiento de la ecuación de diferencias. Por lo tanto, la ecuación diferencial modificada difiere de la ecuación diferencial parcial exacta por el error de truncamiento. De esta manera, se puede concluir que la ecuación diferencial modificada puede ser usada para determinar la consistencia y el orden de un esquema numérico.

El orden de una solución por diferencias de una ecuación diferencial parcial es la razón a la que el error global de la solución por diferencias finitas se aproxima a cero cuando los tamaños de los pasos en la red de puntos tienden a cero. Como ejemplo, se muestra el análisis de consistencia de la aproximación de diferencias finitas de cinco puntos para la ecuación de Poisson

$$u_{xx} + u_{yy} = f$$

cuando $\Delta x = \Delta y = h$.

$$\frac{1}{h^2}(u_{i+1,j} + u_{i,j+1} + u_{i,j-1} + u_{i-1,j} - 4u_{i,j}) = h^2 f \quad (2.15)$$

reordenamos la ecuación de la siguiente manera;

$$(u_{i+1,j} + u_{i-1,j}) + (u_{i,j+1} + u_{i,j-1}) - 4u_{i,j} = h^2 f.$$

Escribiendo la serie de Taylor alrededor del punto (x_i, y_j) para todos los valores de $u(x, y)$ que aparecen en la ecuación anterior, se tiene:

$$u_{|i\pm 1}^j = u_{|i}^j \pm u_x|_i^j h + u_{xx}|_i^j \frac{h^2}{2} \pm u_{xxx}|_i^j \frac{h^3}{6} + u_{xxxx}|_i^j \frac{h^4}{24} \pm \dots$$

$$u|_i^{j\pm 1} = u|_i^j \pm u_y|_i^j h + u_{yy}|_i^j \frac{h^2}{2} \pm u_{yyy}|_i^j \frac{h^3}{6} + u_{yyyy}|_i^j \frac{h^4}{24} \pm \dots$$

Sustituyendo estas expresiones en la ecuación (2.15) y reduciendo la notación tenemos

$$\left(u_{xx} h^2 + u_{xxxx} \frac{h^4}{12} + \dots \right) + \left(u_{yy} h^2 + u_{yyyy} \frac{h^4}{12} + \dots \right) = h^2 f$$

reacomodando términos y dividiendo entre h^2 ambos lados obtenemos

$$u_{xx} + u_{yy} = f - u_{xxxx} \frac{h^2}{12} - u_{yyyy} \frac{h^2}{12}$$

que es la ecuación de Poisson. Por lo tanto, la aproximación de diferencias finitas de cinco puntos es una aproximación consistente de la ecuación de Poisson. El orden de la ecuación es de orden $\mathcal{O}(h^2)$ y podemos ver que es el orden del error de truncamiento local definido anteriormente.

2.2.3. Estabilidad.

Cuando una ecuación diferencial parcial tiene una solución acotada, se dice que la ecuación de diferencias asociada es estable si produce una solución acotada.

El concepto de estabilidad está relacionado con el crecimiento o decrecimiento de los errores que se introducen en la etapa de cómputo. Estos errores no son producidos por una lógica incorrecta sino que se originan porque las computadoras no pueden almacenar un número infinito de cifras decimales, introduciendo de esta manera un error de redondeo.

Un método particular se dice que es estable si el efecto acumulativo de todos los errores de redondeo producidos al aplicar un determinado algoritmo es suficientemente pequeño.

El primer paso en el análisis de la estabilidad de una ecuación de diferencias finitas que aproxima a una ecuación diferencial parcial, es determinar el comportamiento de la solución exacta de la ecuación diferencial parcial. La solución de la mayoría de los problemas físicos es acotada. Por lo tanto, en estos casos, la solución de la ecuación de diferencias finitas también debe ser acotada. Si la solución de la ecuación de diferencias finitas es acotada para cualquier valor de tamaño de paso que se utilice, se dice que la ecuación de diferencias finitas es *incondicionalmente estable*. En cambio, si la ecuación de diferencias finitas es acotada solamente para determinados tamaños de paso, la ecuación de diferencias finitas es *condicionalmente estable*.

2.2.4. Convergencia.

Un método de diferencias finitas es convergente si la solución de la ecuación de diferencias finitas se aproxima a la solución exacta de la ecuación diferencial parcial cuando los tamaños de los pasos en la malla tienden a cero.

La prueba de que una solución aproximada converge a la solución exacta de una ecuación diferencial parcial es generalmente muy difícil, aún en los casos más simples. Por esta razón, se relaciona la convergencia de un método de diferencias finitas con la consistencia y estabilidad de la ecuación de diferencias finitas.

Teorema 2.2 (Equivalencia de Lax). *Considérese un problema lineal de valores en la frontera bien planteado. Un esquema en diferencias finitas para este problema es estable si y sólo si es convergente.*

No se profundiza mucho en estos conceptos debido a que sólo se está dando una introducción además de que no se está trabajando una ecuación en particular, sin embargo en [11] se desarrollan las pruebas de convergencia para algunos de los ejemplos antes presentados.

2.2.5. Segundo ejemplo en dos dimensiones: ecuación del calor.

Anteriormente vimos un ejemplo en dos dimensiones espaciales, ahora veremos un segundo ejemplo en dos dimensiones pero ahora con una dimensión temporal y una espacial.

La ecuación del calor es una EDP dependiente del tiempo, donde las variaciones en el espacio están relacionadas con las variaciones en el tiempo

$$u_t = \nu u_{xx} \tag{2.16}$$

Asumimos $\nu = 1$ por simplicidad, pero ν puede tomar otros valores.

Necesitamos condiciones iniciales en algún tiempo t_0 , donde nuestro $t_0 = 0$

$$u(x, 0) = g(x),$$

y condiciones de frontera (condiciones de Dirichlet)

$$\begin{aligned} u(0, t) &= g_0(t) \quad \text{para } t > 0, \\ u(1, t) &= g_1(t) \quad \text{para } t > 0, \quad 0 \leq x \leq 1. \end{aligned}$$

Anteriormente vimos la discretización de u_{xx} , y ahora veremos la discretización en el tiempo.



Figura 2.5: Esténcil del método de Euler y esténcil de método de Crank-Nicolson.

Aplicamos diferencias finitas a nuestra región discretizada con puntos (x_i, t_n) donde $x_i = i\Delta x$, $t_n = n\Delta t$, Δx es el tamaño de paso en x y Δt es el tamaño de paso en el tiempo. $U_i^n \approx u(x_i, t_n)$ representa la aproximación numérica en el punto (x_i, t_n)

La ecuación del calor se puede resolver utilizando diferencias finitas hacia adelante en el tiempo, determinando los valores de U_i^{n+1} para toda i de los valores U_i^n en el nivel de tiempo anterior. Entonces utilizamos las diferencias centradas en el espacio y diferencias hacia adelante en el tiempo.

Esquema explícito de Euler en la variable temporal

$$\begin{cases} \frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{2u_i^n - u_{j+1}^n - u_{i-1}^n}{(\Delta x)^2} = 0, & i = 1, \dots, N; \quad n \geq 0, \\ u_0^k = u_{N+1}^k = 0, & k \geq 0, \\ u_i^0 = g_i, & i = 1, \dots, N. \end{cases}$$

Calculamos cada U_i^{n+1} explícitamente en términos de los datos anteriores,

$$U_i^{n+1} = U_i^n + \frac{\Delta t}{\Delta x^2}(U_{i-1}^n - 2U_i^n + U_{i+1}^n),$$

la figura 2.5 muestra el esténcil de este método.

Debemos ver que los tamaños de paso en las aproximaciones tengan ordenes de precisión similares lo cual requiere que

$$\Delta t \approx (\Delta x)^2,$$

asumiendo que $\Delta x < 1$ implica que los pasos en el tiempo deben ser mucho más pequeños que los tamaños de paso en el espacio.

Otro método que es muy usado es el método de Crank-Nicolson

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} = \frac{1}{2\Delta x^2}(U_{i-1}^n - 2U_i^n + U_{i+1}^n + U_{i-1}^{n+1} - 2U_i^{n+1} + U_{i+1}^{n+1})$$

Capítulo 3

Diferencias Finitas en Regiones Irregulares.

Hasta el momento solo se había hablado de diferencias finitas en intervalos y regiones rectangulares que son fáciles de discretizar, pero la mayoría de las regiones de interés para trabajar no son rectangulares.

¿Qué pasa cuando queremos resolver una EDP en una región no regular?

A lo largo del tiempo se han ido desarrollando métodos para la discretización de los dominios en que se va a trabajar, algunos de los métodos más desarrollados son los métodos de generación de mallas.

Este trabajo no está centrado en la utilización de mallas para la resolución de las EDPs sin embargo se muestran algunos de los métodos más conocidos como un antecedente y motivación para el desarrollo del mismo.

3.1. Generación de mallas.

Esta sección mostrara un poco de lo que es la generación numérica de mallas, mencionando solo algunos de los métodos más conocidos, los cuales se obtuvieron de [8],[17]; en estos textos podremos ver los métodos más a detalle.

Entenderemos por malla a la discretización del dominio donde se desea modelar un problema mediante la selección de un conjunto finito de puntos que represente la geometría de la región. *i.e.* transformar un dominio físico en una región más simple.

Podemos generar dos tipos de mallas, las *uniformes* y *no uniformes* estas pueden ser en una dimensión en el intervalo $[a, b]$ ver figura 3.1 o dos dimensiones en dominios simples como rectángulos $[0, a] \times [0, b]$ ver figura 3.2. En dos dimensiones podemos encontrar otro tipo de regiones, dominios que no cuentan con una geometría regular donde la generación de mallas no es tan sencilla, este

tipo de dominios serán los de estudio para los diferentes métodos numéricos de generación de mallas.

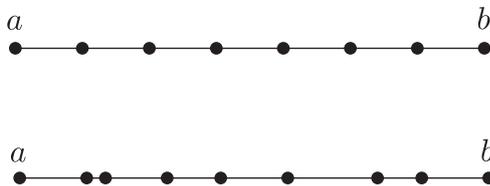


Figura 3.1: Malla uniforme y no uniforme en 1 dimensión.

3.1.1. Generación numérica de mallas estructuradas.

Una malla $\mathbf{x}(\xi, \eta) = (x(\xi, \eta), y(\xi, \eta))^t$ sobre una región $\Omega \in \mathbb{R}^2$ es una función continua $\mathbf{x} : U_2 \rightarrow \Omega$, donde U_2 es el cuadrado unitario $[0, 1] \times [0, 1]$. Si tomamos una línea coordenada ξ o η constante en U_2 su imagen en Ω será una curva, estas curvas son llamadas curvas coordenadas.

Para resolver las ecuaciones diferenciales de la función \mathbf{x} , debe cumplir $\mathbf{x}(\partial U_2) = \partial \Omega$, entonces diremos que se ajustan a la frontera.

Las mallas que nos interesa generar para la región Ω son aquellas que cumplan con las siguientes características.

1. Que todos los puntos diferentes de U_2 tengan como imagen puntos diferentes en Ω ; esto quiere decir que se requiere que \mathbf{x} sea 1-1.
2. Que \mathbf{x} llene a Ω ; es decir, que \mathbf{x} sea un mapeo sobre Ω .
3. Es deseable que la imagen de una curva suave en U_2 sea una curva suave en Ω ; esto es, que las funciones $x(\xi, \eta)$ y $y(\xi, \eta)$ tengan derivadas continuas al ser consideradas como funciones de ξ y η separadamente.

Las mallas que cumplen con las 2 primeras características son llamadas *mallas convexas*, estas se utilizan en trabajos sobre la solución numérica de ecuaciones diferenciales parciales. Si la malla cumple con las 3 características nos dice que existe un difeomorfismo entre U_2 y Ω . Entonces se plantea el problema de generación de mallas como:

Dado un difeomorfismo $\mathbf{x} : \partial(U_2) \rightarrow \partial\Omega$, extenderlo a un difeomorfismo $\mathbf{x} : U_2 \rightarrow \Omega$.

Mencionaremos dos tipos de métodos para generar este tipo de mallas.

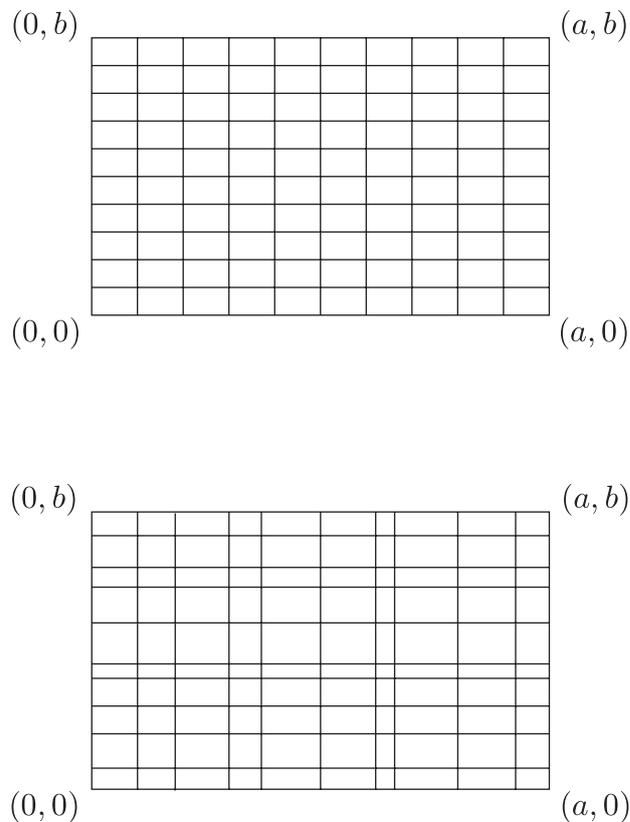


Figura 3.2: Malla uniforme y no uniforme en 2 dimensiones.

- **Construcción por interpolación algebraica.** Estos métodos tienen ventaja de ser de fácil implementación y muy rápidos en comparación con los métodos variacionales contnuos y discretos, por mencionar algunos. Algunas de sus desventajas es que transmiten la falta de suavidad de sus fronteras hacia el interior y frecuentemente generan mallas dobladas, *i.e.* no forman mapeos 1-1. Por su simplicidad estos métodos son muy usados en diferentes áreas y las mallas que generan son utilizadas como punto de partida por otros métodos.
- **Solución numérica de ecuaciones diferenciales parciales.** Las ecuaciones pueden ser elípticas, parabólicas o hiperbólicas, una de sus ventajas es que las mallas obtenidas por este método son más suaves en comparación con las de generación algebraica.

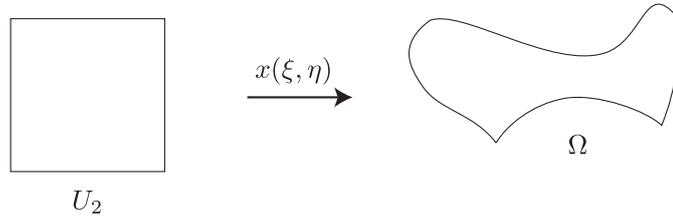


Figura 3.3: $U_2 \longleftrightarrow \Omega$.

3.1.2. Un ejemplo de construcción por interpolación algebraica.

Veremos el método de interpolación transfinita como ejemplo de construcción por interpolación algebraica. Para generar una malla sobre una región Ω se requiere conocer la parametrización de las cuatro partes en las cuales se dividirá la frontera de Ω que son imágenes de cuatro segmentos de la frontera U_2 . La frontera de Ω está dada por cuatro ecuaciones paramétricas

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

donde los subíndices b, r, t, l representan por sus nombres en inglés los lados abajo (bottom), derecha (right), arriba (top), izquierda (left), ver figura ??.

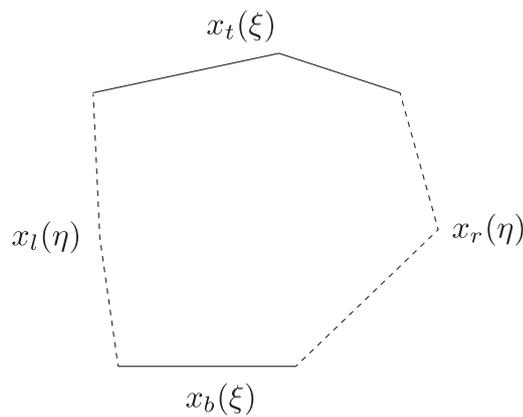


Figura 3.4: Región parametrizada.

Debe conservarse la orientación y además se debe cumplir que en las esquinas

correspondientes los mapeos coincidan para que \mathbf{x} sea continua, *i.e.*

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

La fórmula básica de la interpolación transfinita usa las cuatro ecuaciones paramétricas anteriores para generar una malla sobre Ω de la siguiente manera

$$\mathbf{x}(\xi, \eta) = (1 - \eta)\mathbf{x}_b(\xi) + \eta\mathbf{x}_t(\xi) + (1 - \xi)\mathbf{x}_l(\eta) + \xi\mathbf{x}_r(\eta) - [\xi\eta\mathbf{x}_t(1) + \xi(1 - \eta)\mathbf{x}_b(1) + \eta(1 - \xi)\mathbf{x}_t(0) + (1 - \xi)(1 - \eta)\mathbf{x}_b(0)].$$

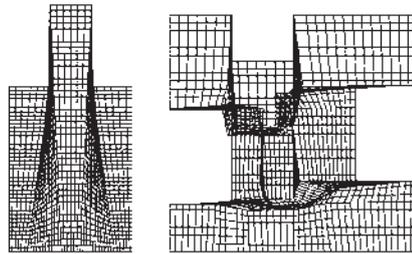


Figura 3.5: Mallas generadas con TFI para las regiones m13 y m19.

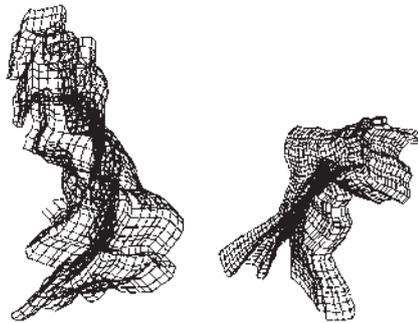


Figura 3.6: Mallas generadas con TFI para las regiones ENG y HAB.

Se muestran mallas generadas por interpolación transfinita para dos regiones no convexas figura 3.5 y para dos regiones geográficas reales 3.6. Las primeras han sido usadas por varios autores para ejemplificar las características de las diferentes técnicas de generación de mallas. Las otras dos son regiones “difíciles”, en el sentido de que no se logra convexidad de las mallas correspondientes cuando se usan métodos clásicos. Estas mallas se obtuvieron tomando la malla uniforme de 41 por 41 puntos por lado del cuadrado unitario, y serán usadas a lo largo del presente trabajo.

3.1.3. Generación Variacional Continua.

Esta sección muestra diferentes métodos de generación de malla mediante la solución numérica de ecuaciones diferenciales parciales.

Entenderemos por un funcional continuo a una función de la forma

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

que a cada malla

$$\mathbf{x} = U_2 \longrightarrow \Omega$$

le asocia un número real. Los funcionales que consideraremos tienen una forma más simple

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

donde L es una función con información acerca de las cantidades geométricas que se desean controlar con el objetivo de generar mallas apropiadas que cumplan con las propiedades especificadas por el funcional.

El problema de la generación de mallas usando métodos variacionales es encontrar \mathbf{x}^* que haga mínimo el valor de $I(\mathbf{x}^*)$; sujeto a las condiciones de frontera. Para esto requerimos algunos resultados de cálculo de variaciones, ver [8]. \mathbf{x}^* debe ser solución del problema de valores en la frontera, también conocido como ecuaciones de Euler–Lagrange

$$\begin{aligned} \frac{d}{d\xi} \left(\frac{\partial L}{\partial x_\xi} \right) + \frac{d}{d\eta} \left(\frac{\partial L}{\partial x_\eta} \right) &= 0 \\ \frac{d}{d\xi} \left(\frac{\partial L}{\partial y_\xi} \right) + \frac{d}{d\eta} \left(\frac{\partial L}{\partial y_\eta} \right) &= 0 \end{aligned}$$

con $\mathbf{x}(\partial U_2)$ dado. Comúnmente este problema de valores a la frontera se resuelve numéricamente tomando una malla generada por interpolación transfinita como una aproximación inicial.

A continuación se muestra un listado de los funcionales más conocidos

- **Funcional de longitud.** El funcional clásico de longitud Φ (en su forma más simple) se obtiene haciendo

$$\lambda L(x_\xi, x_\eta, y_\xi, y_\eta) = x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2.$$

Al obtener el mínimo del funcional se está tratando de obtener una malla que tenga la característica de que las longitudes de las líneas coordenadas sean lo más cercanas posibles a un mismo valor.

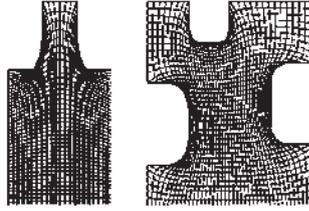


Figura 3.7: Mallas generadas usando el funcional continuo de longitud para las regiones m13 y m19.

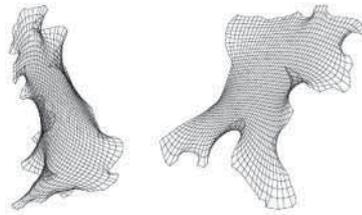


Figura 3.8: Mallas generadas usando el funcional continuo de longitud para las regiones ENG y HAB.

Las mallas generadas por este funcional no son satisfactorias ya que para la mayoría de las regiones no convexas son mallas dobladas, sin embargo son muy suaves, figuras 3.7 3.12.

- **Funcional de Área.** El funcional continuo de área Φ_A en su versión más simple está dado por

$$\begin{aligned}\Phi_A(\mathbf{x}) &= \int_0^1 \int_0^1 (x_\xi y_\eta - x_\eta y_\xi)^2 d\eta d\xi \\ &= \int_0^1 \int_0^1 J^2 d\eta d\xi;\end{aligned}$$

está diseñado para producir mallas tales que la variación del jacobiano sea lo menor posible en el sentido de los cuadrados mínimos. Las mallas generadas no son suaves y, salvo en casos simples, son dobladas, figura 3.9.

- **Funcional de ortogonalidad.** El objetivo al utilizar este funcional es obtener mallas con líneas coordenadas que sean lo más ortogonal posible, es decir, que las tangentes a dichas líneas coordenadas sean ortogonales en cada punto; lo cual podemos describir por la relación

$$O = \mathbf{x}_\xi^t \mathbf{x}_\eta = 0.$$

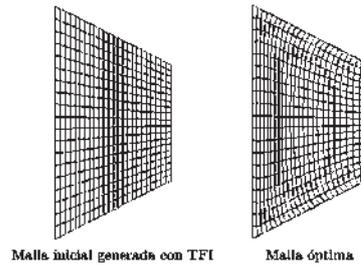


Figura 3.9: Mallas generadas usando el funcional continuo de área para la región CUA.

Se trata de calcular la más cercana posible a la ortogonalidad.

Muy frecuentemente su proceso de solución no converge, y cuando lo hace las mallas generadas son, a menudo, severamente dobladas.

Combinación de funcionales.

Knupp y Steinberg [8] introducen el uso de combinaciones convexas de los funcionales anteriores con el objetivo de lograr mallas que tengan en cierta medida, las ventajas que ofrece cada funcional por separado. La forma general de esta combinación es

$$\Phi(\mathbf{x}) = \sigma_l \Phi_l(\mathbf{x}) + \sigma_A \Phi_A(\mathbf{x}) + \sigma_O \Phi_O(\mathbf{x})$$

en donde $\sigma_l, \sigma_A, \sigma_O \leq 0$ y $\sigma_l + \sigma_A + \sigma_O = 1$. La elección de los parámetros $\sigma_l, \sigma_A, \sigma_O$ depende fuertemente de la experiencia personal, lo cual es una gran limitante para el uso de esta combinación.

- **Funcionales Área–Longitud.** Las combinaciones con $\sigma_O = 0$, que generan los llamados funcionales de área–longitud son muy usados, ya que para regiones no convexas y geoméricamente no muy complicadas producen mallas convexas y aceptablemente suaves; sin embargo en la mayoría de las regiones no simples la convexidad no logra ser alcanzada, figura 3.10.
- **Funcionales Área–Ortogonalidad.** Un generador que sí es automático, es decir, no depende de los parámetros, se obtiene utilizando los valores

$$\sigma_l = 0 \quad \sigma_A = 1/2 \quad \sigma_O = 1/2$$

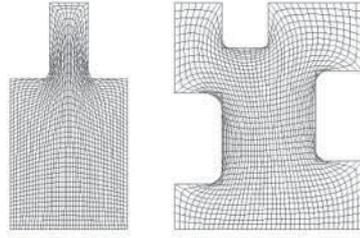


Figura 3.10: Mallas generadas usando el funcional continuo de área-longitud para las regiones m13 y m19.

que da lugar al funcional de área-ortogonalidad de Knupp, que tiene la forma

$$\begin{aligned}
 \Phi_{AO} &= \frac{1}{2} \int_0^1 \int_0^1 [J^2 + O^2] d\eta d\xi \\
 &= \frac{1}{2} \int_0^1 \int_0^1 [(x_\xi y_\eta - x_\eta y_\xi)^2 + (x_\xi x_\eta + y_\xi y_\eta)^2] d\eta d\xi \\
 &= \frac{1}{2} \int_0^1 \int_0^1 (x_\xi^2 y_\eta^2 + x_\xi^2 x_\eta^2 + y_\xi^2 y_\eta^2) d\eta d\xi \\
 &= \frac{1}{2} \int_0^1 \int_0^1 [(x_\xi^2 + y_\xi^2)(x_\eta^2 + y_\eta^2)] d\eta d\xi
 \end{aligned}$$

Las mallas producidas por el funcional de área-ortogonalidad son suaves y convexas, figuras 3.13, 3.13.

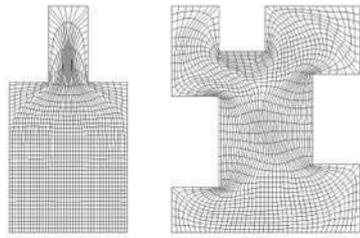


Figura 3.11: Mallas generadas usando el funcional continuo de área-ortogonalidad para las regiones m13 y m19.

- **Funcional de suavidad.** El funcional de longitud definido para la transformación inversa

$$\xi(\mathbf{x}) = \begin{pmatrix} \xi(x, y) \\ \eta(x, y) \end{pmatrix} \quad \xi : \Omega \longrightarrow U_2$$

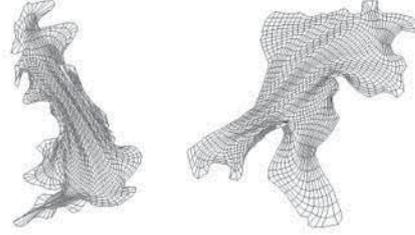


Figura 3.12: Mallas generadas usando el funcional continuo de área–ortogonalidad para las regiones ENG y HAB.

está dado por

$$\int \int_{\Omega} (\xi_x^2 + \xi_y^2 + \eta_x^2 + \eta_y^2) dydx$$

el cual, al hacer un cambio de coordenadas hacia el espacio lógico (integrando sobre el cuadrado unitario) se escribe como

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

Este es el funcional de suavidad de Winslow, también referido como el funcional Thompson-Thames-Mastin homogéneo (TTM).

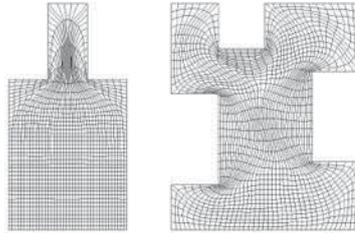


Figura 3.13: Mallas generadas usando el funcional continuo de área–ortogonalidad para las regiones m13 y m19.

Este funcional usualmente se restringe a mejorar, en lo que a suavidad se refiere, una malla que se ha optimizado por otros métodos, figuras 3.13, 3.14.

3.1.4. Generación Variacional Discreta

Nuestro grupo de trabajo ha estado trabajando en la generación de mallas utilizando funcionales discretos, adelante se muestran resumidos algunos de estos

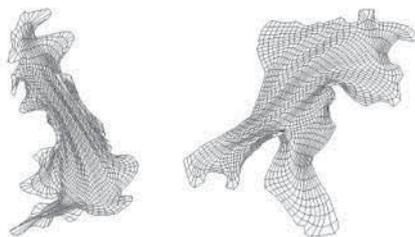


Figura 3.14: Mallas generadas usando el funcional continuo de área–ortogonalidad para las regiones ENG y HAB.

métodos que han servido de motivación para la realización de este trabajo, para más detalle ver [17], [3].

Esta solución se realiza de manera numérica eligiendo una malla sobre el cuadrado unitario; es decir,

$$(\xi_i, \eta_j) \quad i = 1, 2, \dots, m \quad j = 1, 2, \dots, n$$

y aplicando un esquema de diferencias finitas para obtener aproximaciones $P_{i,j}$ a los valores de $\mathbf{x}(\phi_i, \eta_j)$.

Varios autores proponen realizar primeramente una discretización del funcional a minimizar, y después resolver el correspondiente problema de optimización multivariada. En otras palabras, el funcional continuo

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

es reemplazado por una función de $m \times n$ variables

$$\{P_{i,j}\} \quad i = 1, 2, \dots, m \quad j = 1, 2, \dots, n$$

$$F(P_{1,1}, P_{1,2}, \dots, P_{i,n}, P_{2,1}, P_{2,2}, \dots, P_{2,n}, \dots, P_{m,1}, P_{m,2}, \dots, P_{m,n}) = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} f_{i,j}$$

donde $f_{i,j}$ es una aproximación del valor de L en la celda de $[0, 1] \times [0, 1]$ definida por los puntos

$$(\xi_{i-1}, \eta_{j-1}), (\xi_{i-1}, \eta_j), (\xi_i, \eta_j), (\xi_i, \eta_{j-1}).$$

Los puntos

$$\{P_{1,j}\}, \{P_{m,j}\} \quad j = 1, 2, \dots, n \quad \{P_{i,1}\}, \{P_{i,n}\} \quad i = 1, 2, \dots, m$$

son los valores correspondientes del mapeo \mathbf{x} en los puntos de la frontera del cuadrado unitario

$$\begin{aligned} \{\xi_1, \eta_j\}, & \quad \{\xi_m, \eta_j\}, & j = 1, 2, \dots, n \\ \{\xi_i, \eta_1\}, & \quad \{\xi_i, \eta_m\}, & i = 1, 2, \dots, m \end{aligned}$$

que están fijos y los puntos interiores

$$\{P_{i,j}\} \quad i = 2, \dots, m-1 \quad j = 2, \dots, n-1$$

que son las imágenes bajo el mapeo buscado \mathbf{x} de los puntos del cuadrado unitario

$$\{(\xi_i, \eta_j)\} \quad i = 2, \dots, m-1 \quad j = 2, \dots, n-1$$

se calculan de tal forma que hagan mínimo a F .

Esta minimización de F se realiza partiendo de una malla inicial, generalmente generada por TFI. Se usa un método de optimización multivariada de gran escala, por ejemplo *L-BFGS* ó *Newton Truncado*. Cabe mencionar que aunque los métodos discretos clásicos se obtienen a partir de la discretización del funcional continuo correspondiente, también se puede pensar en generar directamente funcionales discretos; o en otras palabras, proponer una cantidad geométrica que se quiera minimizar sobre cada celda de la malla.

- **Funcional Discreto de Longitud.** La forma más simple del funcional discreto de longitud es

$$F_l = \frac{1}{2N\bar{\alpha}} \sum_{q=1}^N l(\Delta q) = \frac{1}{2N\bar{\alpha}} \sum_{q=1}^N l_q$$

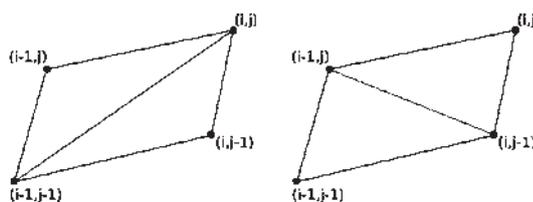


Figura 3.15: Formación de los triángulos relevantes.

donde $N = 4(m-1)(n-1)$ es el número de triángulos relevantes en la malla figura 3.15, m y n representan el tamaño de la malla y $\bar{\alpha}$ es el doble del área promedio de todos los triángulos. Este funcional es usado debido a los buenos resultados que otorga al ser combinado con otros funcionales que comentaremos más adelante figuras . Una versión más general es propuesta en [2] y [?].

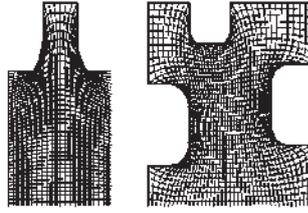


Figura 3.16: Mallas generadas usando el funcional discreto de longitud para las regiones m13 y m19.

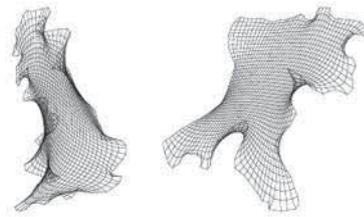


Figura 3.17: Mallas generadas usando el funcional discreto de longitud para las regiones ENG y HAB.

- **Funcional Clásico de Área en su Versión Discreta.** De acuerdo a la formulación anterior, el funcional clásico de área, en su versión discreta es

$$F_{A^2} = \frac{1}{2N\bar{\alpha}} \sum_{q=1}^N \alpha_q^2.$$

Una malla ideal G_* para el funcional de área es aquella en la cual las áreas de todos los triángulos son iguales. El proceso de minimización del funcional F_{A^2} siempre converge, figuras 3.18, 3.19.

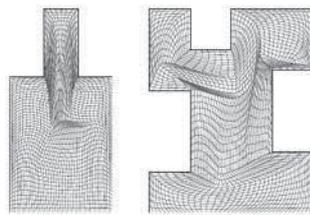


Figura 3.18: Mallas generadas usando el funcional discreto de área para las regiones m13 y m19.

- **Funcional Discreto de Ortogonalidad.** Aunque el uso de la versión discreta del funcional de ortogonalidad no conduce a buenas mallas cuando se

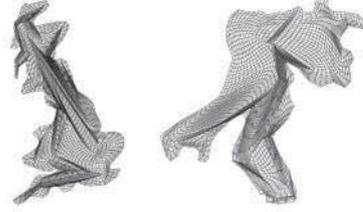


Figura 3.19: Mallas generadas usando el funcional discreto de área para las regiones ENG y HAB.

usa de manera aislada para casi ninguna región, su combinación con otros funcionales sí lo hace. La forma que tiene es

$$F_O = \frac{1}{2N\bar{\alpha}} \sum_{q=1}^N \mathcal{O}(\Delta_q)^2 = \frac{1}{2N\bar{\alpha}} \sum_{q=1}^N \mathcal{O}_q^2.$$

donde \mathcal{O}_q^2 es la discretización del integrando en el funcional continuo de ortogonalidad

$$o^2 = \|c - a\|^2 \|b - a\|^2 \cos^2 \theta.$$

Una malla ideal para el funcional de ortogonalidad es aquella en la que las celdas son rectángulos.

Funcionales Combinados.

También en el enfoque discreto es conveniente usar combinaciones convexas de diferentes funcionales para producir mallas más adecuadas. La forma general de esta combinación es

$$F = \tau_1 F_{A^2} + \tau_2 F_l + (1 - \tau_1 - \tau_2) F_O$$

donde $0 \leq \tau_1, \tau_2 \leq 1$ y $\tau_1 + \tau_2 \leq 1$. Los funcionales se usan en su forma normalizada.

- **Funcional Área–Longitud.** Cuando se hace $\tau_1 = \tau, \tau_2 = 1 - \tau$ obtenemos la familia de funcionales

$$F = \tau F_{A^2} + (1 - \tau) F_l \quad 0 \leq \tau \leq 1$$

que con elecciones adecuadas de τ (usualmente con τ cercano a 1) dan lugar a mallas que son aceptables para la mayoría de las regiones sencillas. Es cuestión de experiencia elegir el valor de τ y lograr buenas mallas en los casos en los que la región es complicada, figuras 3.20, 3.21.

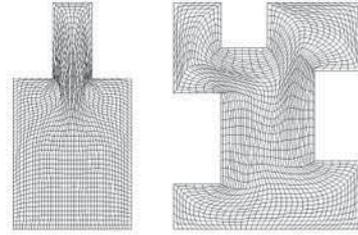


Figura 3.20: Mallas generadas usando el funcional discreto de área-longitud para las regiones m13 y m19.

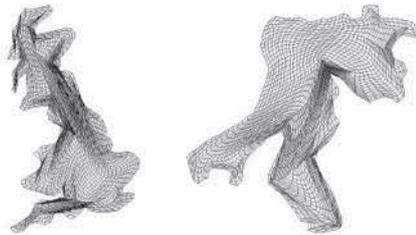


Figura 3.21: Mallas generadas usando el funcional discreto de área-longitud para las regiones ENG y HAB.

- Funcional Discreto de Área-Ortogonalidad.** Al hacer $\tau_2 = 0$ se obtiene la familia conocida como funcionales de Área-Ortogonalidad. La forma de estos funcionales es

$$F_{ao}^\tau = \tau_1 F_{A^2} + (1 - \tau_1) F_O.$$

Cuando se habla del *Funcional de Área-Ortogonalidad*, sin especificar el valor del parámetro τ , se hace referencia al miembro de esta familia correspondiente al valor $\tau = 0.5$, figuras 3.22, 3.23.

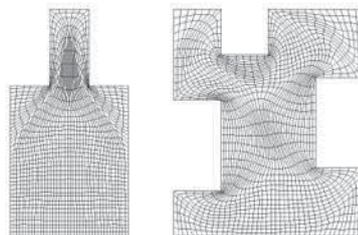


Figura 3.22: Mallas generadas usando el funcional discreto de área-ortogonalidad para las regiones m13 y m19.

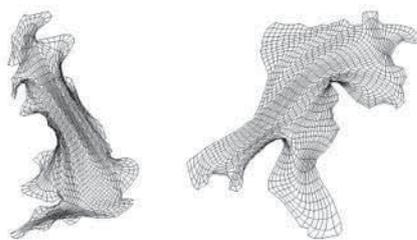


Figura 3.23: Mallas generadas usando el funcional discreto de área–ortogonalidad para las regiones ENG y HAB.

3.1.5. Funcionales de Suavidad y Área adaptativos de Tinoco.

El objetivo principal a conseguir en la generación de mallas sobre regiones planas irregulares es el que la malla sea convexa; una malla que no cumpla con este requisito es completamente inútil para los propósitos que se persiguen. Los métodos revisados hasta ahora no garantizan que el objetivo se logre; en muchos casos la malla óptima es no convexa. Sería deseable contar con un funcional tal que el proceso de minimización conduca siempre a una malla convexa y que se tengan resultados que garanticen que eso siempre se logra, para ver más detalles [17].

- **Funcional de Suavidad Adaptativa** El funcional discreto de suavidad está definido como

$$\sum_{q=1}^N \frac{l_q}{\alpha_q}.$$

Una de las metas deseables es la de obtener funcionales en cuyo proceso de optimización no se requiera de condiciones de arranque muy estrictas y que puedan producir mallas suaves. Supóngase que se tiene una malla no convexa tal que el área de cada triángulo que contribuye al funcional es mayor que $-k/2$.

Entonces la función

$$f_k = \frac{l}{k + \alpha}$$

toma valores positivos para cada triángulo en la malla. Esta función da lugar al funcional

$$F_s = \frac{1}{2N} \sum_{q=1}^N \frac{l}{k + \alpha}.$$

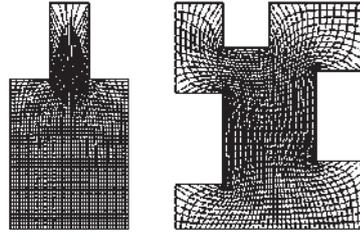


Figura 3.24: Mallas generadas usando el funcional discreto de suavidad para las regiones m13 y m19.

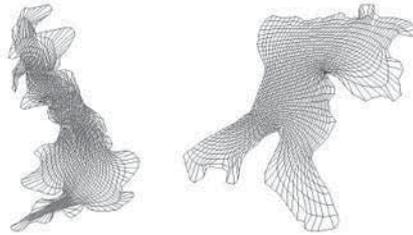


Figura 3.25: Mallas generadas usando el funcional discreto de suavidad para las regiones ENG y HAB.

Minimizando este funcional se obtiene una malla que está más cerca de ser convexa que la original. Una adecuada actualización de k produce, en un número finito de pasos, una malla convexa, figuras 3.24, 3.25, ver [17].

- El funcional de Área Adaptativa. Al hacer una versión discreta del funcional de área continuo explicado en [17] se obtiene el funcional discreto de área para una malla discreta G

$$F_{A^{-1}}(G) = \sum_{\Delta \in G} f_{A^{-1}}(\Delta) = \sum_{\Delta \in G} \frac{1}{\alpha(\Delta)}$$

que, usando una numeración adecuada de triángulos puede ser escrito como

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

Debemos notar que para que el proceso de optimización de $F_{A^{-1}}$ genere una malla convexa, se necesita que la malla inicial ya lo sea. Esta sería limitación para el uso de este funcional la cual, sin embargo, se podrá superar más adelante, figuras 3.26, 3.27.

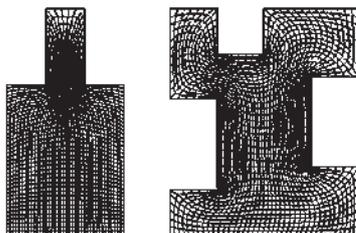


Figura 3.26: Mallas generadas usando el funcional discreto de área para las regiones m13 y m19.

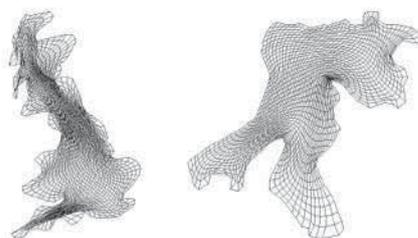


Figura 3.27: Mallas generadas usando el funcional discreto de área para las regiones ENG y HAB.

- **Modificación del Funcional.** Motivados por el buen resultado obtenido en [17] al modificar el funcional de suavidad para ampliar su dominio de definición, se considerará el funcional discreto dado por

$$F_{A^{-1},k} = \frac{k + \bar{\alpha}}{N} \sum_{q=1}^N \frac{1}{\alpha_q + k}$$

el cual se puede optimizar si se parte de una malla G tal que $\alpha_-(G) > -k$ y producirá una malla óptima con la misma propiedad sobre α_- ; de hecho si \hat{G} es la malla óptima y k se elige adecuadamente, $\alpha_-(\hat{G}) > \alpha_-(G)$.

- **Combinaciones de área con longitud.** Una manera de suavisar la mallas producidas con los funcionales de área es seguir la idea de usarlos en combinación con el funcional de longitud. Esto conlleva también a disminuir los tiempos necesarios para obtener la convergencia.

Esta combinación tiene la forma

$$F_{A^{-1}-l}^\tau = \tau F_{A^{-1}} + (1 + \tau) F_l \quad 0 \leq \tau \leq 1.$$

Las figuras 3.28, 3.29, corresponden al valor de $\tau = 0.75$.

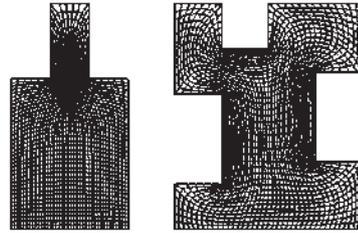


Figura 3.28: Mallas generadas usando el funcional adaptativo de área-longitud para las regiones m13 y m19.

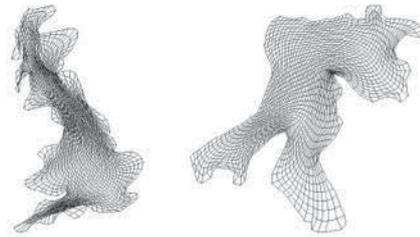


Figura 3.29: Mallas generadas usando el funcional adaptativo de área-longitud para las regiones ENG y HAB.

- Combinaciones de Suavidad con Área.** Una manera de obtener mallas convexas y suaves que tengan una buena distribución del tamaño de las celdas es combinar el esquema adaptativo de los funcionales de suavidad con algún funcional de área. Esta idea da buenos resultados para las combinaciones con el funcional clásico de área

$$F_{S-A}^{\tau} = \tau F_k + (1 - \tau) F_{A^2} \quad 0 \leq \tau \leq 1.$$

Las figuras 3.30, 3.31, corresponden al valor de $\tau = 0.5$.

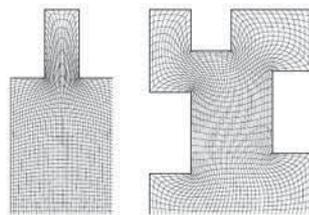


Figura 3.30: Mallas generadas usando el funcional Suavidad-Área para las regiones m13 y m19.

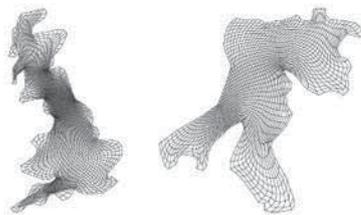


Figura 3.31: Mallas generadas usando el funcional Suavidad-Área para las regiones ENG y HAB.

3.1.6. Funcional Quasi-Armónico de Domínguez

Siguiendo las ideas mostradas hasta el momento Domínguez-Mota y Barrera-Sánchez proponen en [3] un nuevo funcional quasi-armónico que depende de un parámetro $\omega > 0$, basado en el funcional armónico de Ivanenko [7]

$$H(G) = \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^4 \frac{\lambda(\Delta_{i,j}^k)}{\alpha(\Delta_{i,j}^k)} = \sum_{q=1}^N \frac{\lambda(\Delta_q)}{\alpha(\Delta_q)},$$

donde m y n son el número horizontal y vertical de puntos en la malla, y k se usa para denotar los 4 triángulos formados por cada celda de la misma. En este caso N se utiliza para denotar el número total de triángulos de la malla, *i.e.* $N = 4mn$.

La idea principal es reemplazar α^{-1} en el funcional de Ivanenko por una función capaz de actuar como una barrera continua y definiendo $f(\Delta q) = \lambda(\Delta q)\phi_\omega$. Una opción para $\phi_\omega(\alpha)$ es una función continua, convexa y estrictamente decreciente que tienda a $1/\alpha$ para $\alpha \geq \omega$. Entonces, se considera para un número real positivo ω

$$\phi_\omega(\alpha) = \begin{cases} \frac{2\omega - \alpha}{\omega^2} & \text{si } \alpha < \omega \\ \frac{1}{\alpha} & \text{si } \alpha \geq \omega \end{cases}$$

y gracias a esta función se define el funcional

$$H_\omega(G) = \sum_{q=1}^N \lambda(\Delta q)\phi_\omega(\Delta q).$$

Ver figuras 3.32, 3.33.

Un estudio más a fondo de este funcional, así como la implementación del mismo, pueden encontrarse en [3].

Los funcionales discretos antes mencionados se encuentran implementados en UNAMALLA [21].

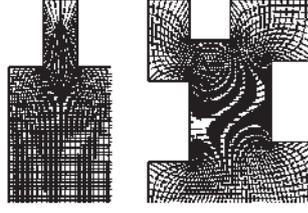


Figura 3.32: Mallas generadas usando el funcional $H_\omega(G)$ para las regiones m13 y m19.

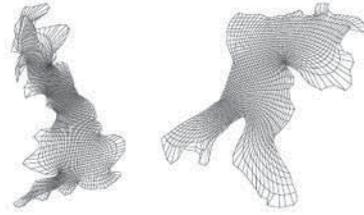


Figura 3.33: Mallas generadas usando el funcional $H_\omega(G)$ para las regiones ENG y HAB.

3.2. Diferencias Finitas Generalizadas.

Nos encontramos ahora en una región irregular en la cual ya no tenemos nuestros nodos uniformemente espaciados y no podemos aplicar los esquemas antes vistos, por ello se necesita un esquema adecuado para estas regiones.

Usualmente para los esquemas de diferencias finitas se parte de un nodos central y sus nodos vecinos $p_0 = (x_0, y_0), \dots, p_q = (x_q, y_q)$; los cuales se requieren para encontrar los coeficientes $\Gamma_0, \Gamma_1, \Gamma_2, \dots, \Gamma_q$ tales que

$$\sum_{i=0}^k \Gamma_i u(p_i) \approx [Au_{xx} + Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu]_{p_0}. \quad (3.1)$$

Queremos que nuestra aproximación se acerque a la solución a medida que los nodos vecinos se aproximan al nodo central p_0 . Entonces el valor del límite de (3.1) está dado por la condición de consistencia

$$\sum_{i=0}^q \Gamma_{i,0} \phi(p_i, 0) - [A\phi_{xx} + B\phi_{xy} + C\phi_{yy} + D\phi_x + E\phi_y]_{p_0} \longrightarrow 0, \quad (3.2)$$

cuando $p_1, \dots, p_q \longrightarrow p_0$, lo anterior lo escribimos haciendo la expansión de Taylor

hasta el segundo orden y reagrupando terminos de la siguiente manera

$$\begin{aligned}
 & [Au_{xx} + Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu]_{p_0} - \sum_{i=0}^q \Gamma_i u(p_i) = \\
 & (F(p_0) - \sum_{i=0}^q \Gamma_i) u(p_0) + (D(p_0) - \sum_{i=1}^q \Gamma_i \Delta x_i) u_x(p_0) + \\
 & (E(p_0) - \sum_{i=1}^q \Gamma_i \Delta y_i) u_y(p_0) + \left(A(p_0) - \sum_{i=1}^q \frac{\Gamma_i (\Delta x_i)^2}{2} \right) u_{xx}(p_0) + \\
 & (B(p_0) - \sum_{i=1}^q \Gamma_i \Delta x_i \Delta y_i) u_{xy}(p_0) + \left(C(p_0) - \sum_{i=1}^q \frac{\Gamma_i (\Delta y_i)^2}{2} \right) u_{yy}(p_0) + \\
 & \mathcal{O}(\max\{\Delta x_i, \Delta y_i\})^3
 \end{aligned} \tag{3.3}$$

donde $\Delta x_i = x_i - x_0$, $\Delta y_i = y_i - y_0$, $1 \leq i \leq q$.

Lo que queremos es que cada uno de los coeficientes de (3.3) sea cero, si logramos esto la diferencia entre nuestro esquema y el operador quedará acotada por la función \mathcal{O} y disminuirá al acercarse los nodos vecinos.

Los q nodos vecinos definidos por p_0, p_1, \dots, p_q , de (3.3) podemos obtener un sistema de ecuaciones (3.4). Queremos ver cuáles son los coeficientes $\Gamma_{i,0}$ tales que al realizar el producto con los respectivos Δx_i y Δy_i nos den los coeficientes correspondientes del operador lineal.

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ 0 & \Delta x_1 & \dots & \Delta x_q \\ 0 & \Delta y_1 & \dots & \Delta y_q \\ 0 & (\Delta x_1)^2 & \dots & (\Delta x_q)^2 \\ 0 & \Delta x_1 \Delta y_1 & \dots & \Delta x_q \Delta y_q \\ 0 & (\Delta y_1)^2 & \dots & (\Delta y_q)^2 \end{pmatrix} \begin{pmatrix} \Gamma_0 \\ \Gamma_1 \\ \Gamma_2 \\ \cdot \\ \cdot \\ \Gamma_q \end{pmatrix} = \begin{pmatrix} F(p_0) \\ D(p_0) \\ E(p_0) \\ 2A(p_0) \\ B(p_0) \\ 2C(p_0) \end{pmatrix} \tag{3.4}$$

el cual se debe satisfacer para los coeficientes Γ_i . Notemos que este sistema (3.4) no está bien determinado, debemos encontrar una solución que nos brinde un esquema consistente.

El problema se tiene que resolver para cada uno de los puntos interiores de la nube. Usando la condición de consistencia, separamos la primera ecuación del sistema (3.4)

$$\sum_{i=1}^q \Gamma_i - F = 0 \tag{3.5}$$

y entonces solucionamos el sistema definido por

$$\begin{pmatrix} \Delta x_1 & \dots & \Delta x_q \\ \Delta y_1 & \dots & \Delta y_q \\ (\Delta x_1)^2 & \dots & (\Delta x_q)^2 \\ \Delta x_1 \Delta y_1 & \dots & \Delta x_q \Delta y_q \\ (\Delta y_1)^2 & \dots & (\Delta y_q)^2 \end{pmatrix} \begin{pmatrix} \Gamma_1 \\ \Gamma_2 \\ \cdot \\ \cdot \\ \Gamma_q \end{pmatrix} = \begin{pmatrix} D(p_0) \\ E(p_0) \\ 2A(p_0) \\ B(p_0) \\ 2C(p_0) \end{pmatrix}. \tag{3.6}$$

Ahora tenemos el sistema

$$\widetilde{M}\Gamma = \widetilde{\beta},$$

del cual calculamos la pseudoinversa de \widetilde{M} para obtener los valores de $\Gamma_1, \dots, \Gamma_q$ y Γ_0 se obtiene de (3.5).

Una vez calculadas las gamas en cada uno de los nodos de la malla, obtenemos una matriz de $M \times M$ donde M es el número total de nodos de nuestra malla. La matriz final a resolver ya no es tan sencilla como las que vimos anteriormente (tridiagonal o tridiagonal por bloques) aunque sigue siendo rala es más difícil de resolver, para esto podemos utilizar el cálculo de la inversa o la pseudoinversa.

Capítulo 4

Nubes.

Sabemos que mucha de la teoría de aproximación a la solución de ecuaciones diferenciales parciales se lleva a cabo en diferentes tipos de mallas, estructuradas o no estructuradas. En el capítulo anterior vimos un panorama de todo lo que conlleva generar una malla adecuada para hacer las aproximaciones de las EDPs y las complicaciones que se presentan en los dominios irregulares. Con la finalidad de buscar alternativas que nos ayuden a ahorrar el costo computacional de la generación de una malla se propone utilizar nubes de puntos. Es decir un conjunto finito de puntos $\mathbf{x} = \{x_1, \dots, x_n\} \subset \Omega$ que describen la geometría de Ω , donde $\Omega \in \mathbb{R}^2$ es la región discretizada. Al conjunto puntos en el control del esquema son llamados “una estrella” de nodos. El número y la posición de nodos de cada estrella dependerán del criterio utilizado, ver figuras 4.10, 4.11, 4.12.

La idea de usar nubes de puntos se ha ido desarrollando a través del tiempo, en los sesentas P. S. Jensen [10] publicó las bases de un método de diferencias finitas el cual hacía uso de nubes de puntos con seis nodos en la estrella. Usando la serie de expansión de Taylor, obtuvo una fórmula de diferencias la cual aproximaba derivadas de segundo orden; sin embargo este método presentaba mal condicionamiento de la estrella. La idea de usar la simplicidad del método de diferencias finitas para nubes de puntos para resolver problemas de ecuaciones diferenciales de segundo orden con condiciones de frontera condujo a muchos autores a concentrarse en cómo evitar estas singularidades de las estrellas. N. Perrone y R. Kao [15] sugirieron la adición de nodos. T. Lizka y J. Orkisz [13] hicieron interesantes contribuciones para desarrollar la selección de nodos de la estrella y evitar los problemas mencionados anteriormente.

Las nubes a diferencia de las mallas que ya vimos anteriormente no tienen sus vecinos bien definidos, por ello es necesario la implementación de métodos que los seleccionen eficientemente. Esto debido a que la mayoría de las implementaciones están hechas para mallas de puntos, como lo es el esquema que utilizaremos. Debido a la geometría de las regiones tiene sus propias complicaciones que más adelante

explicaremos.

Los criterios de elección de nodos buscan formar una estrella de 9 nodos lo más regular posible, similar a los 9 nodos en el esquema de una malla rectangular e ir almacenando la información para posteriormente aplicar el esquema de diferencias finitas.

Las nubes que se están utilizando son nubes irregulares, algunas fueron obtenidas del trabajo de J. J. Benito [4] y de mallas utilizadas en trabajos anteriores por nuestro grupo de trabajo [16] y generadas con el programa UNAMALLA [21], con el fin de poder comparar nuestros resultados. Es por esto que no se modificó la distribución de los nodos de las mallas de las cuales fueron tomadas.

Las nubes a utilizar son las siguientes:

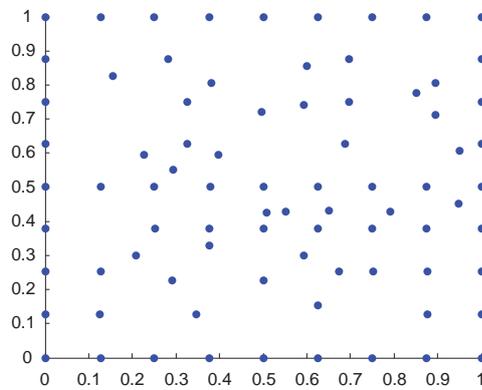


Figura 4.1: Nube 1.

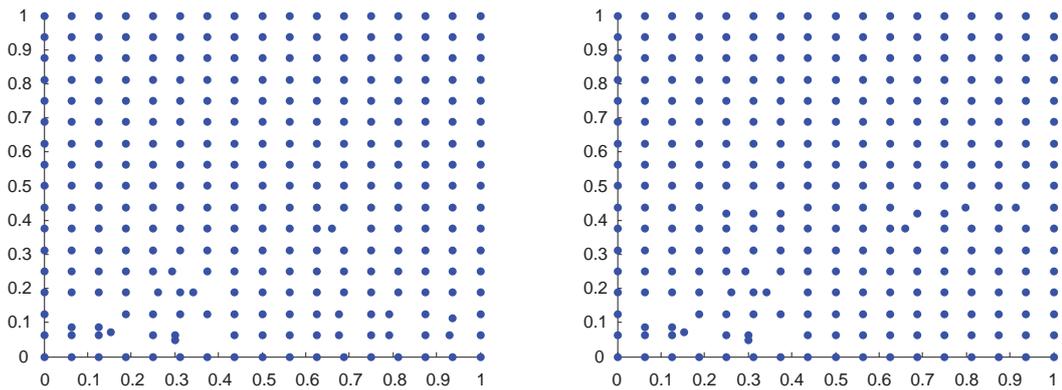


Figura 4.2: Nubes 2 y 3.

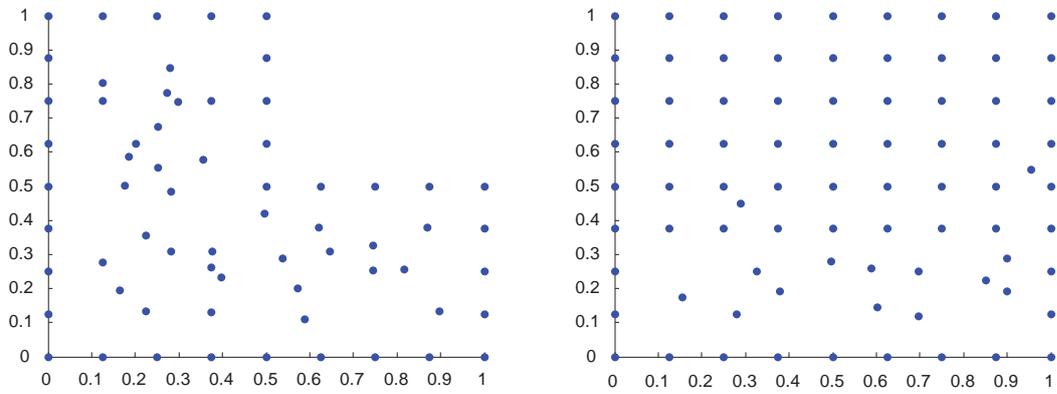


Figura 4.3: Nubes 4 y 5.

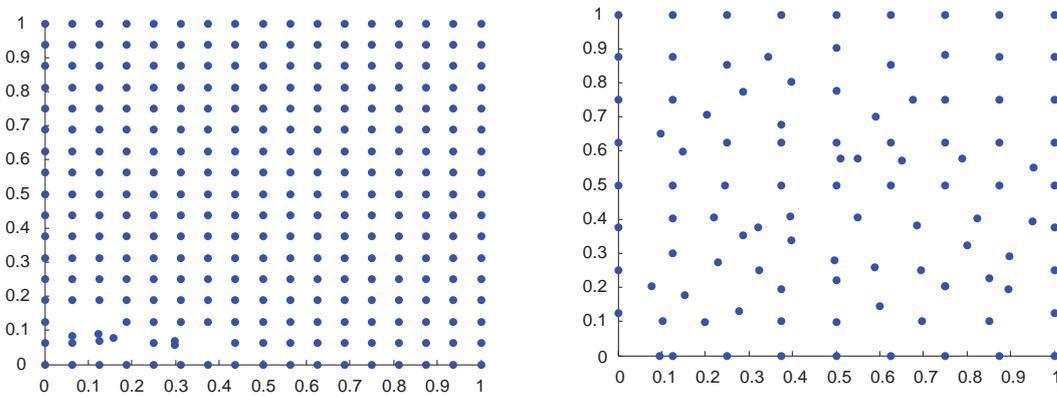


Figura 4.4: Nubes 6 y 7.

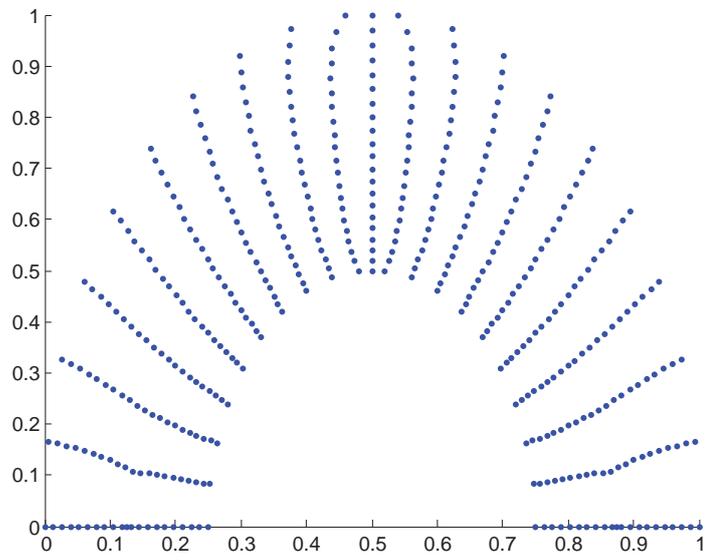


Figura 4.5: Annulus 441 nodos.

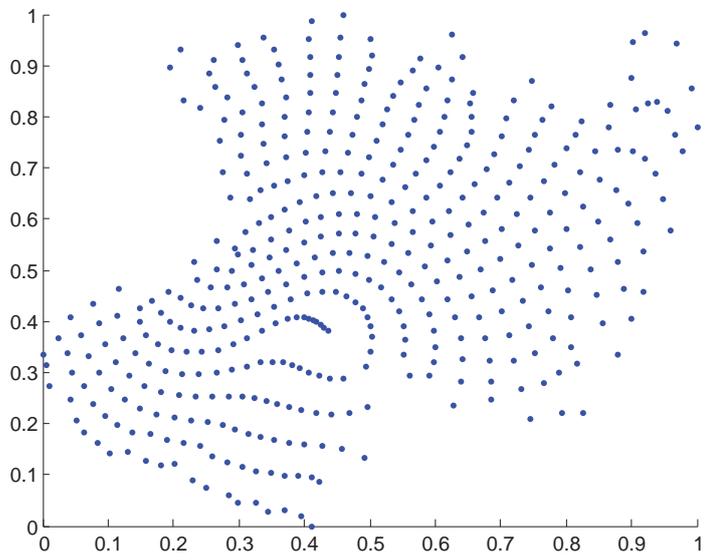


Figura 4.6: Michoacán 441 nodos.

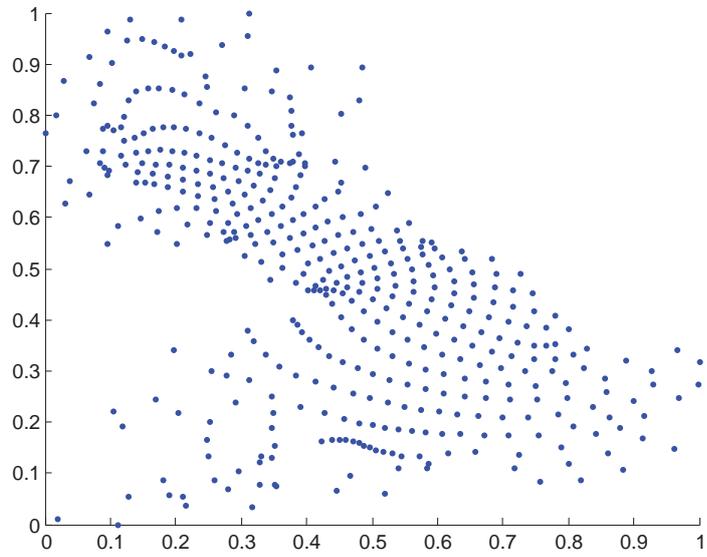


Figura 4.7: Inglaterra 441 nodos.

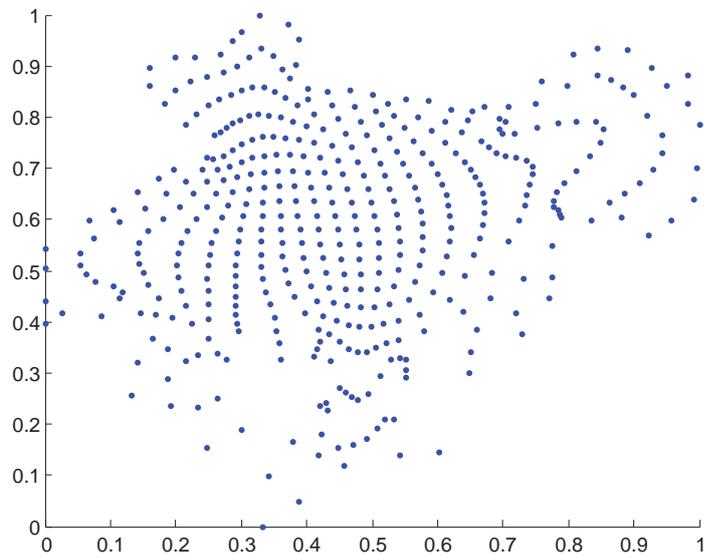


Figura 4.8: Ucha 441 nodos.

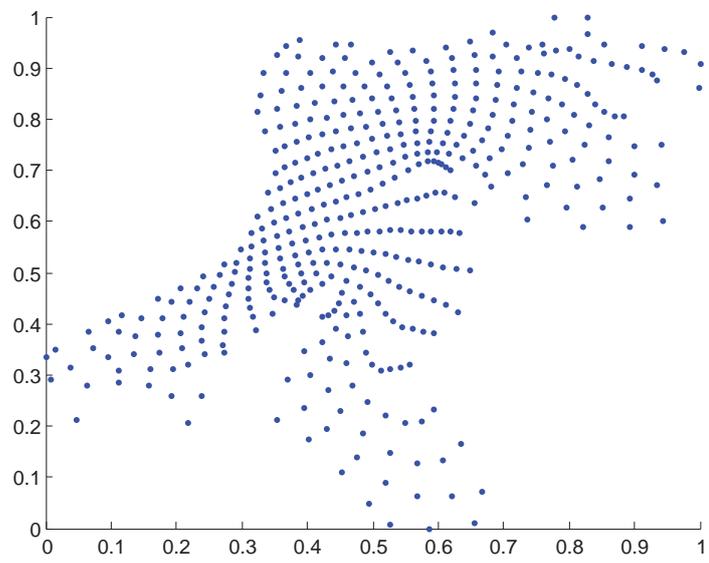


Figura 4.9: Habana 441 nodos.

4.1. Criterios de elección de nodos.

En esta sección nos concentraremos en mostrar el funcionamiento en general de algunos de los métodos mencionados anteriormente. El método utilizado para obtener los resultados obtenidos en el presente trabajo esta basado en el método conocido como *criterio de los cuadrantes*, usando algunos datos de J. J. Benito et al [4].

Criterios de selección de nodos.

- **Criterio de la distancia** (P. S. Jensen [10]).

Selecciona los nodos para la estrella de acuerdo a su distancia del nodo. El criterio es simple pero falla frecuentemente ya que produce estrellas distorsionadas, es decir, distribuciones muy irregulares alrededor del nodo central el cual causa resultados imprecisos, ver Figura 4.10.

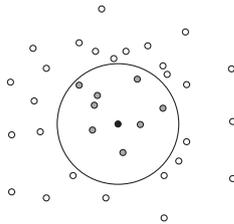


Figura 4.10: Criterio de la distancia.

- **Criterio de los ocho segmentos** (N. Perrone, R. Kao [15]).

Consiste en la selección de un nodo por cada octante en un sistema de ejes cartesianos alrededor del eje central de la estrella. Generan buena selección de estrellas, pero es muy riguroso, demasiado complicada y consume mucho tiempo de computo, ver Figura 4.11.

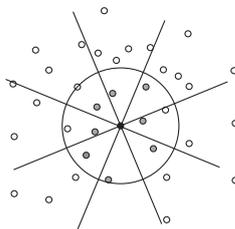


Figura 4.11: Criterio de los ocho segmentos.

■ **Criterio de los cuatro cuadrantes** (T. Lizka, J. Orkisz [13]).

Consiste en la elección de los dos nodos más cercanos por cuadrante, los cuadrantes son establecidos de la misma manera que los ejes cartesianos, ver Figura 4.12.

4.1.1. Variantes del criterio de los cuatro cuadrantes.

El método a utilizar nos sirvió de motivación por los resultados que autores presentaron en sus trabajos como [13], [4]. Sin embargo se hicieron algunas modificaciones del criterio pero se mantuvo la idea original, esto porque se pretendió mejorar la elección de los nodos y utilizarlo en regiones complicadas. Se generaron dos variaciones una más apegada que la otra. Para minimizar el tiempo de cómputo en la elección de los nodos se propone seguir los pasos de cada variante.

• Primera variante:

1. Definimos una vecindad inicial en uno de los cuadrantes con dimensiones basadas en el número de nodos totales de la nube.
2. Aplicamos el criterio de la distancia en la vecindad definida y elegimos los nodos necesarios.
3. Cambiamos de cuadrante y repetimos los pasos anteriores.

Se sigue de la misma manera para cada uno de los cuadrantes hasta elegir los nodos necesarios ver Figura 4.12.

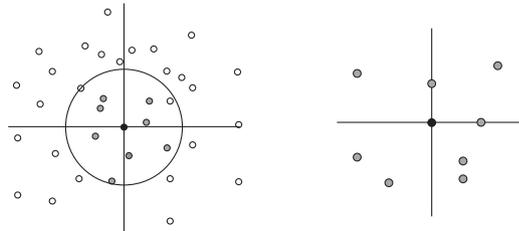


Figura 4.12: Criterio de los cuatro cuadrantes.

• Segunda variante:

1. Seleccionamos 10 nodos por medio del criterio de la distancia.
2. Elegimos un nodo por cuadrante.
3. Los nodos faltantes los elegimos por el criterio de la distancia.

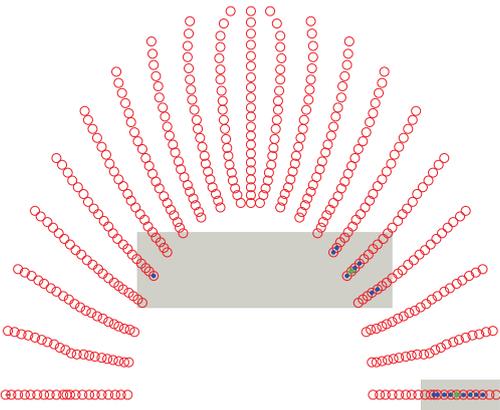
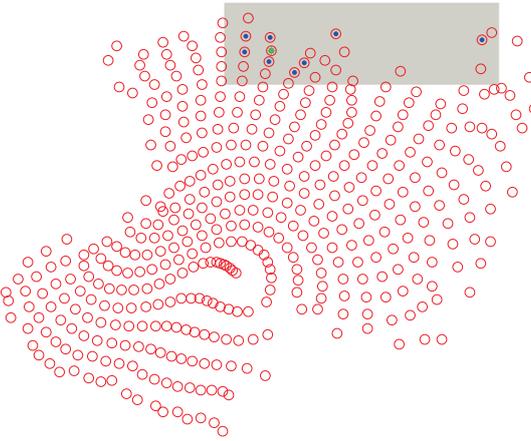
Debemos tener en cuenta que al momento de hacer las implementaciones tenemos que ver los siguientes casos:

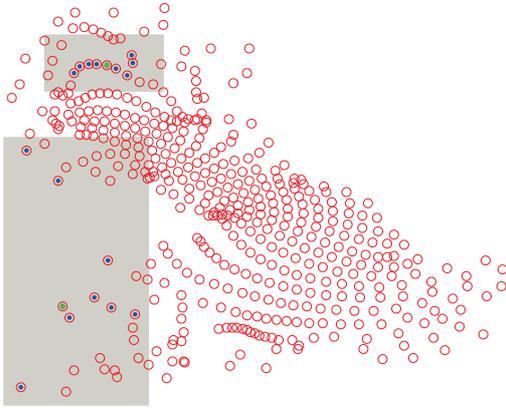
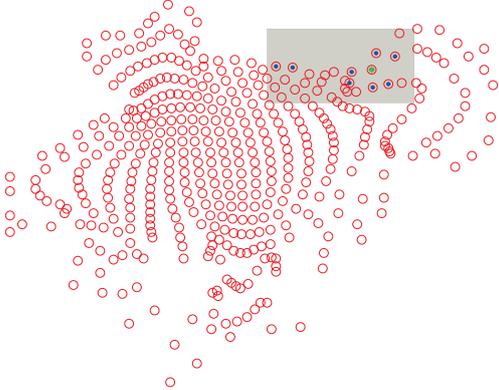
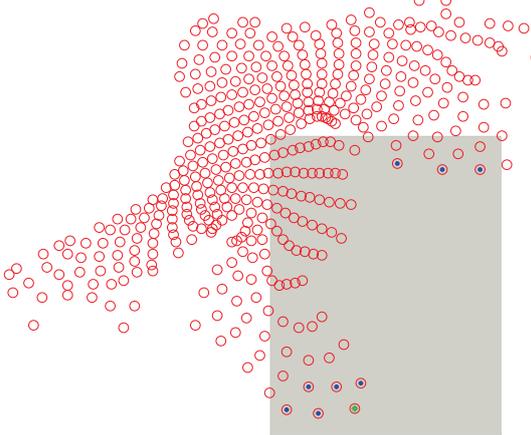
- cuando son nodos de la frontera,
- cuando la vecindad definida no contiene nodos,
- cuando el cuadrante no tiene nodos y
- tamaño de la vecindad,

Las implementaciones propuestas se pueden ver en el apéndice A.

4.2. Problemas en la elección de nodos.

Como ya se mencionó anteriormente la decisión de elegir el criterio de los cuadrantes fue por su simplicidad y resultados del método en trabajos de otros autores [4], [13]; aun así las complicaciones se hacen presentes en la elección de los vecinos debido a la irregularidad, geometría y distribución de nodos en las nubes. Cabe mencionar que la implementación de los métodos de elección de vecinos se hizo buscando la mejora de los problemas a mostrar en las nubes de 441, 1681, 3721, y 6561 nodos, sin modificar las nubes; estos problemas son similares en todas las nubes, sin embargo los mostraremos en las nubes de 441 nodos para ver las estrellas con mayor claridad. Aunque los problemas resultan un tanto repetitivos en todas las nubes queremos mostrar de forma gráfica de ejemplos de estrellas irregulares y como es que el método puede volverse complicado cuando se trabaja con nubes irregulares.

Nube	Problema
	<ul style="list-style-type: none"> ■ El arco interior de la nube genera un hueco, esto hace que el método busque el nodo más cercano del nodo central extendiendo el cuadrante hasta la frontera contraria. ■ Debido a que la distancia del nodo central a los nodos superiores es mayor que la distancia hacia los nodos de los costados, el método elige los vecinos alineados.
	<ul style="list-style-type: none"> ■ Aquí ocurre algo similar a lo anterior, aunque parece que no se tuvieron huecos grandes cerca del nodo central, uno de los cuadrantes no tiene nodos cercanos necesarios y se extiende hasta encontrar el más próximo.

Nube	Problema
	<ul style="list-style-type: none"> ■ Ocurre algo similar a la nube Annulus, el arco genera un hueco pero además se ve afectada en todos los cuadrantes ya que los nodos están muy separados del nodo central. ■ También se generan estrellas con los nodos casi alineados por la misma razón que en la nube Annulus.
	<ul style="list-style-type: none"> ■ Al igual que en la nube Michoacán el cuadrante se extiende hasta el nodo más cercano del nodo central. Este es el problema más frecuente en la elección de nodos cerca de la frontera.
	<ul style="list-style-type: none"> ■ Una vez más se presenta el problema de Annulus, que a su vez es el problemas que hemos estado viendo en todas las nubes.

Capítulo 5

Resultados numéricos y conclusiones.

En este capítulo mostraremos los resultados numéricos utilizando los criterios de selección de nodos variantes del criterio de los cuadrantes junto con el esquema de diferencias finitas mostrado en la sección 3.2.

5.1. Ecuaciones diferenciales parciales utilizadas.

Las siguientes EDPs fueron utilizados de prueba en J. J. Benito et al [4].

- Laplace

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad (5.1)$$

con solución exacta

$$u(x, y) = x^2 - y^2. \quad (5.2)$$

- Poisson

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 12(x^2 - y^2), \quad (5.3)$$

con solución exacta

$$u(x, y) = x^4 - y^4. \quad (5.4)$$

- Hiperbólica

$$\frac{\partial^2 u}{\partial x^2} + 2\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial x \partial y} + \frac{\partial u}{\partial x} - \frac{\partial u}{\partial y} = 0, \quad (5.5)$$

con solución exacta

$$u(x, y) = e^x \sen y. \quad (5.6)$$

- Hiperbólica

$$\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 0, \quad (5.7)$$

con solución exacta

$$u(x, y) = x^2 + y^2. \quad (5.8)$$

También se aproximaron las siguientes EDPs de prueba en nuestro grupo de trabajo.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 10e^{2x+y} \quad (5.9)$$

con solución exacta

$$u(x, y) = 2e^{2x+y}.$$

$$\nabla \cdot (Z \nabla u(x, y)) = f(x, y) \quad (5.10)$$

donde

$$Z \equiv \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} = P^{-1}DP$$

$$P = \begin{bmatrix} \cos(\frac{\pi}{8}) & \text{sen}(\frac{\pi}{8}) \\ -\text{sen}(\frac{\pi}{8}) & \cos(\frac{\pi}{8}) \end{bmatrix} \quad \begin{bmatrix} 1 + 2x^2 + y^2 & 0 \\ 0 & 1 + x^2 + 2y^2 \end{bmatrix}$$

solución exacta

$$u(x, y) = \text{sen}(\pi x) \sin(\pi y).$$

Por último se hizo la aproximación a la ecuación del calor.

$$\frac{\partial u}{\partial t} = \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (5.11)$$

con $\nu = 0.4$, $0 \leq t \leq 1$ y condiciones iniciales

$$u(x, y, t_0) = e^{-2\pi^2\nu t} \cos \pi x \cos \pi y.$$

5.2. Resultados.

En esta primera parte se muestran los errores de aproximación de de las EDPs Laplace 5.1, Poisson 5.3, Hiperbólica 5.5, Hiperbólica 5.7, en las primeras 7 nubes. Se utiliza el siguiente estimador del error global normalizado esto para la comparación de los resultados con el trabajo de J. J. Benito et al [4].

El error en la solución está definido por el siguiente estimador del error global.

$$\text{Error} = \frac{\sqrt{\{\sum_{i=1}^N (\text{aprox}(i) - \text{exac}(i))^2\}/N}}{|\text{exac}_{max}|} \times 100,$$

donde N es el número total de nodos del dominio considerado.

En la siguiente tablas se muestran los errores de aproximación de las EDP's utilizando las dos variantes del criterio de los cuadrantes en la elección de nodos.

	Laplace 5.1	Poisson 5.3	Hiperbólica 5.5	Hiperbólica 5.7
nube 1	9.4223×10^{-15}	0.4173	0.0278	5.1073×10^{-14}
nube 2	1.7678×10^{-14}	0.0069	0.0021	4.1415×10^{-12}
nube 3	1.4930×10^{-14}	0.0120	0.0023	2.6214×10^{-11}
nube 4	3.5832×10^{-15}	0.2643	0.0708	1.0786×10^{-12}
nube 5	8.3693×10^{-15}	0.1247	0.0219	3.3399×10^{-13}
nube 6	1.1421×10^{-14}	0.0045	0.0024	1.1438×10^{-11}
nube 7	5.5312×10^{-15}	0.3868	0.0514	5.4933×10^{-14}

Tabla 5.1: Error global normalizado de las aproximaciones utilizando la primera variante del método de los cuadrantes.

La tabla 5.1 muestra los errores obtenidos con la primera variante del criterio de los cuadrantes, guiados por ellos se ve claramente que las ecuaciones que más se aproximan a su solución exacta son (5.1) y (5.7); se puede decir que para estas ecuaciones en las 7 nubes el método de elección de nodos funciona bien, pues aunque los errores no son del mismo orden magnitud siguen siendo pequeños. Por otro lado las ecuaciones (5.3) y (5.5) son más complicadas y se ven afectadas por las estrellas irregulares que están presentes en las elecciones de nodos, ver figuras 5.1. Sin embargo los resultados son comparables con los que se muestran en J. J. Benito et al [4].

	Laplace 5.1	Poisson 5.3	Hiperbólica 5.5	Hiperbólica 5.7
nube 1	9.1926×10^{-15}	0.2383	0.0246	4.0967×10^{-13}
nube 2	1.3826×10^{-14}	0.0152	0.0051	4.9946×10^{-12}
nube 3	1.5275×10^{-14}	0.0178	0.0061	9.2560×10^{-12}
nube 4	5.1570×10^{-15}	0.2355	0.0538	2.6546×10^{-14}
nube 5	8.1225×10^{-15}	0.1315	0.0166	1.5070×10^{-13}
nube 6	1.5319×10^{-14}	0.0031	0.0027	22.6310
nube 7	7.2126×10^{-15}	0.3192	0.0389	9.6211×10^{-14}

Tabla 5.2: Error global normalizado de las aproximaciones utilizando la segunda variante del método de los cuadrantes.

Los errores de la tabla 5.2 que son con la segunda variante del criterio tienen un comportamiento similar a los de la tabla anterior, se puede decir que los métodos tienen un comportamiento muy parecido a pesar de seguir procedimientos distintos para la elección de nodos, figuras 5.1, 5.2.

Los siguientes resultados se obtuvieron utilizando nubes irregulares tomadas de mallas generadas con UNAMALLA [21] y utilizadas en trabajos anteriores del grupo de matemáticas aplicadas.

Las nubes utilizadas son:

- Annulus 4.5 de 441, 1681, 3721, y 6561 nodos.
- Michoacán 4.6 de 441, 1681, 3721, y 6561 nodos.
- Inglaterra 4.7 de 441, 1681, 3721, y 6561 nodos.
- Ucha 4.8 de 441, 1681, 3721, y 6561 nodos.
- Habana 4.9 de 441, 1681, 3721, y 6561 nodos.

Para las cuales se calculó el error cuadrático medio

$$Error = \sqrt{\sum_{i=1}^n (\text{sol}(i) - \text{exac}(i))^2 \frac{1}{N}}.$$

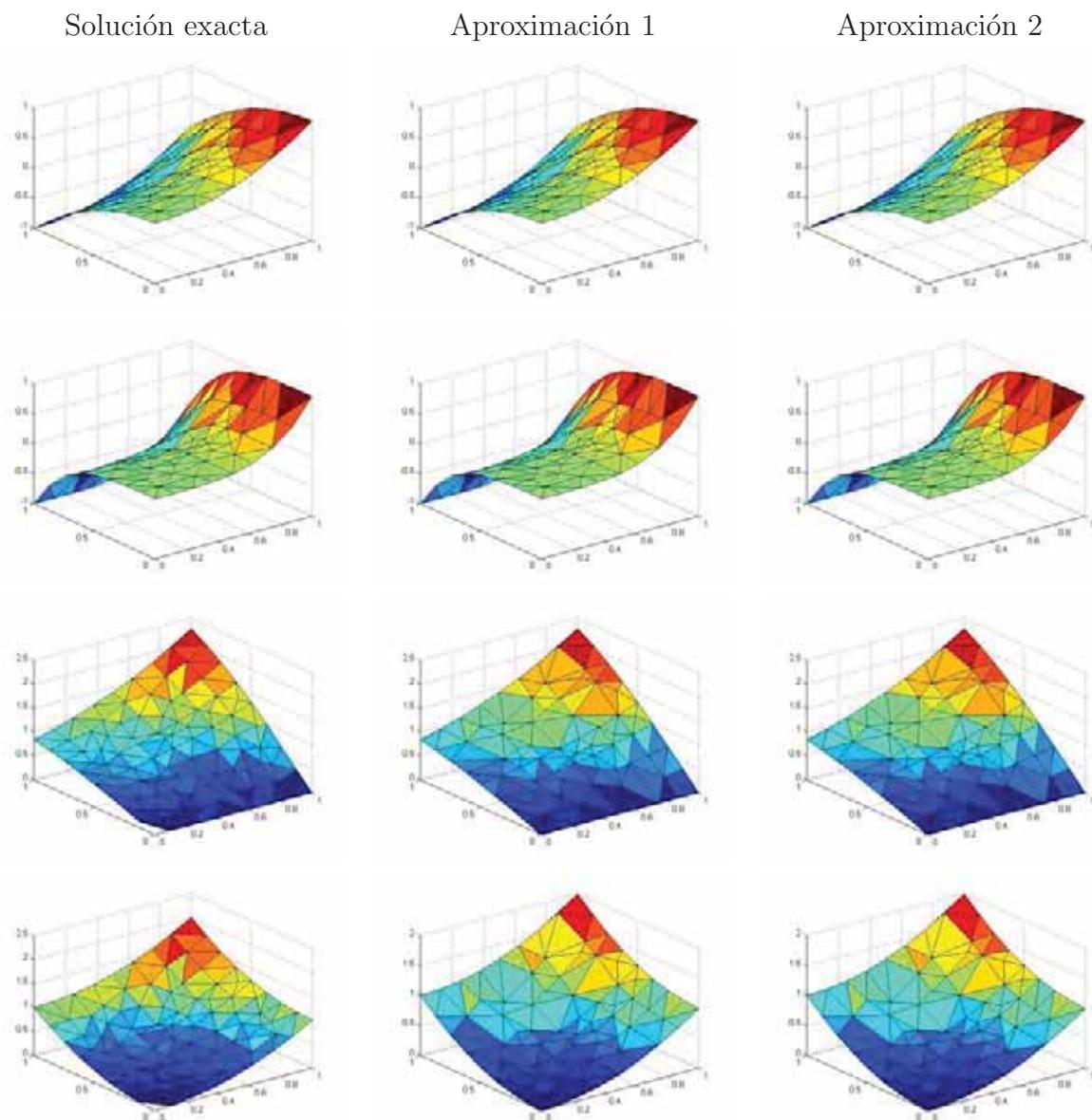


Figura 5.1: En esta figura se muestran las gráficas de los resultados de aproximación de los problemas Laplace 5.1, Poisson 5.3, Hiperbólica 5.5, Hiperbólica 5.7 en la nube 1, en el orden mencionadas. En Aproximación 1 se utilizó la primera variante del criterio de los cuadrantes y en Aproximación la segunda variante.

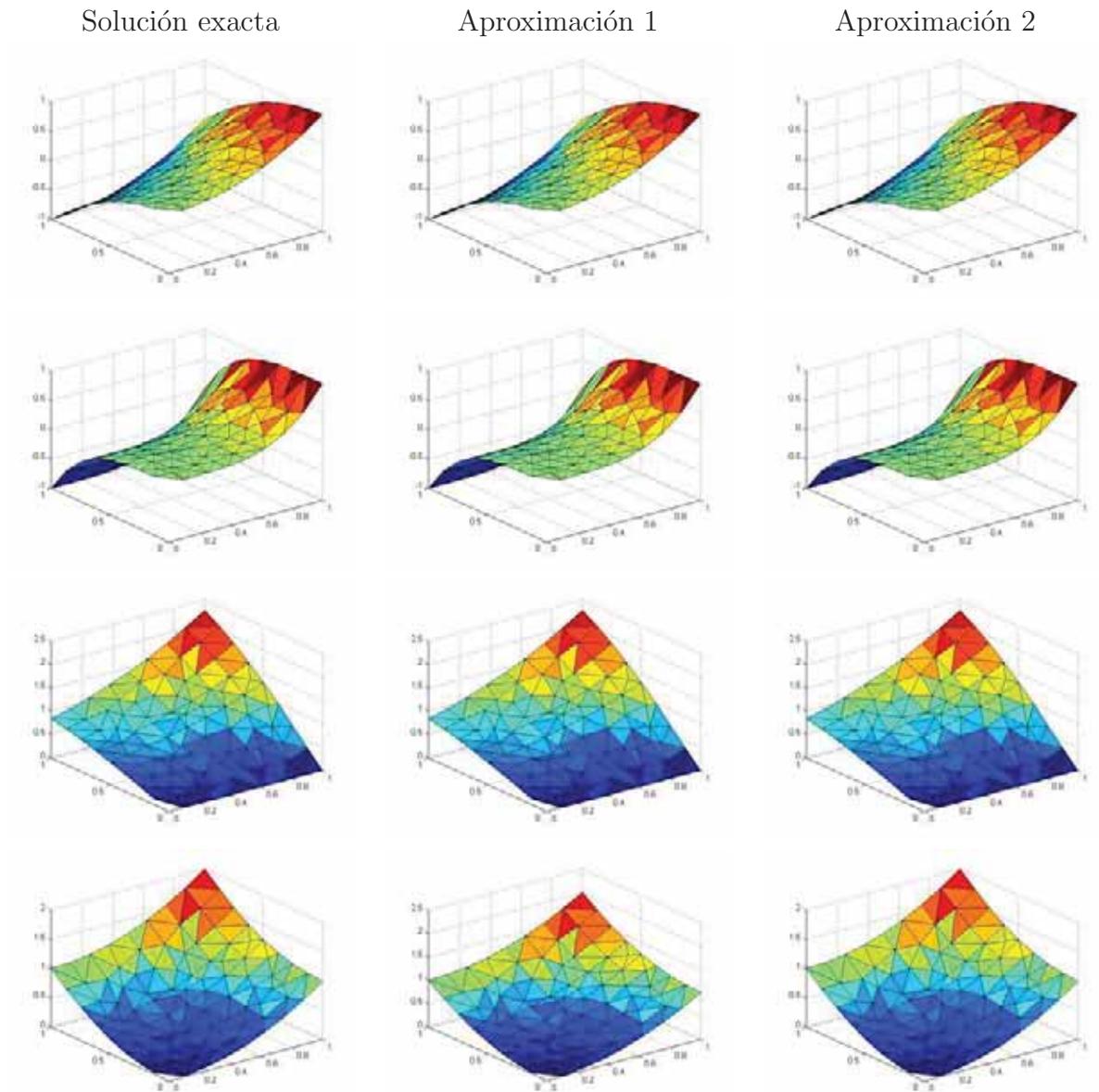


Figura 5.2: En esta figura se muestran las gráficas de los resultados de aproximación de los problemas Laplace 5.1, Poisson 5.3, Hiperbólica 5.5, Hiperbólica 5.7 en la nube 7, en el orden mencionadas. En Aproximación 1 se utilizó la primera variante del criterio de los cuadrantes y en Aproximación la segunda variante.

nube	nodos	Laplace 5.1	Poisson 5.3	Hiperbólica 5.5	Hiperbólica 5.7
ann	441	1.5913×10^{-16}	7.1492×10^{-4}	3.5322×10^{-4}	3.0618×10^{-15}
	1681	3.9618×10^{-16}	3.2665×10^{-4}	7.6524×10^{-5}	2.4890×10^{-13}
	3721	3.9675×10^{-16}	8.9743×10^{-5}	5.3551×10^{-5}	7.1712×10^{-14}
	6561	2.4065×10^{-15}	0.0272	8.0608×10^{-5}	4.0734×10^{-14}
mich	441	2.1879×10^{-15}	0.0443	0.0052	2.5750×10^{-14}
	1681	2.9652×10^{-15}	0.0072	0.0025	2.4429×10^{-13}
	3721	4.6269×10^{-15}	0.0018	0.0027	7.8059×10^{-13}
	6561	1.0341×10^{-14}	0.0040	5.5108×10^{-4}	3.1235×10^{-12}
eng	441	1.6251×10^{-16}	0.0013	1.4792×10^{-4}	6.4313×10^{-15}
	1681	9.6537×10^{-17}	2.0248×10^{-4}	7.1347×10^{-5}	1.3370×10^{-14}
	3721	2.9221×10^{-16}	5.1173×10^{-5}	7.5854×10^{-5}	2.2820×10^{-14}
	6561	4.5054×10^{-16}	1.1283×10^{-4}	1.5587×10^{-5}	4.9879×10^{-14}
uch	441	6.7157×10^{-17}	5.2493×10^{-4}	1.4454×10^{-4}	2.3904×10^{-15}
	1681	1.5166×10^{-16}	5.0920×10^{-4}	4.7444×10^{-5}	2.2738×10^{-13}
	3721	6.3767×10^{-15}	2.0679×10^{-4}	1.9016×10^{-5}	3.5383×10^{-14}
	6561	4.8820×10^{-16}	0.0018	2.7395×10^{-5}	4.2618×10^{-14}
hab	441	8.2981×10^{-17}	2.3454×10^{-4}	1.1263×10^{-4}	2.2243×10^{-14}
	1681	1.5416×10^{-16}	8.4876×10^{-5}	3.5897×10^{-5}	4.4867×10^{-15}
	3721	1.6907×10^{-16}	4.6628×10^{-5}	1.3424×10^{-5}	1.0560×10^{-13}
	6561	2.4664×10^{-16}	3.2642×10^{-5}	2.1223×10^{-5}	1.2476×10^{-14}

Tabla 5.3: Error cuadrático medio de las aproximaciones de las EDPs tomadas del trabajo J. J. Benito, utilizando la primera variante del método de los cuadrantes.

5.2.1. Problemas tomados del trabajo de J. J. Benito. en las nubes del grupo de trabajo.

En las tablas 5.2.2, 5.4 muestran los errores de aproximación de las EDPs (5.1), (5.3), (5.5), (5.7); seguidas por los de las EDPs usadas de prueba en el grupo de trabajo (5.9), (5.10), (5.11), en todas las nubes.

En la tabla 5.2.2 claramente se puede ver que las ecuaciones (5.1) y (5.7) siguen siendo las que mejor se aproximan a sus soluciones exactas y que las ecuaciones (5.3) y (5.5) son las que siguen teniendo problemas. Si vemos el catálogo de nubes nos daremos cuenta de que se complicó la geometría de las nubes y también aumento el número de nodos en cada una de ellas, lo que nos dice que el número de

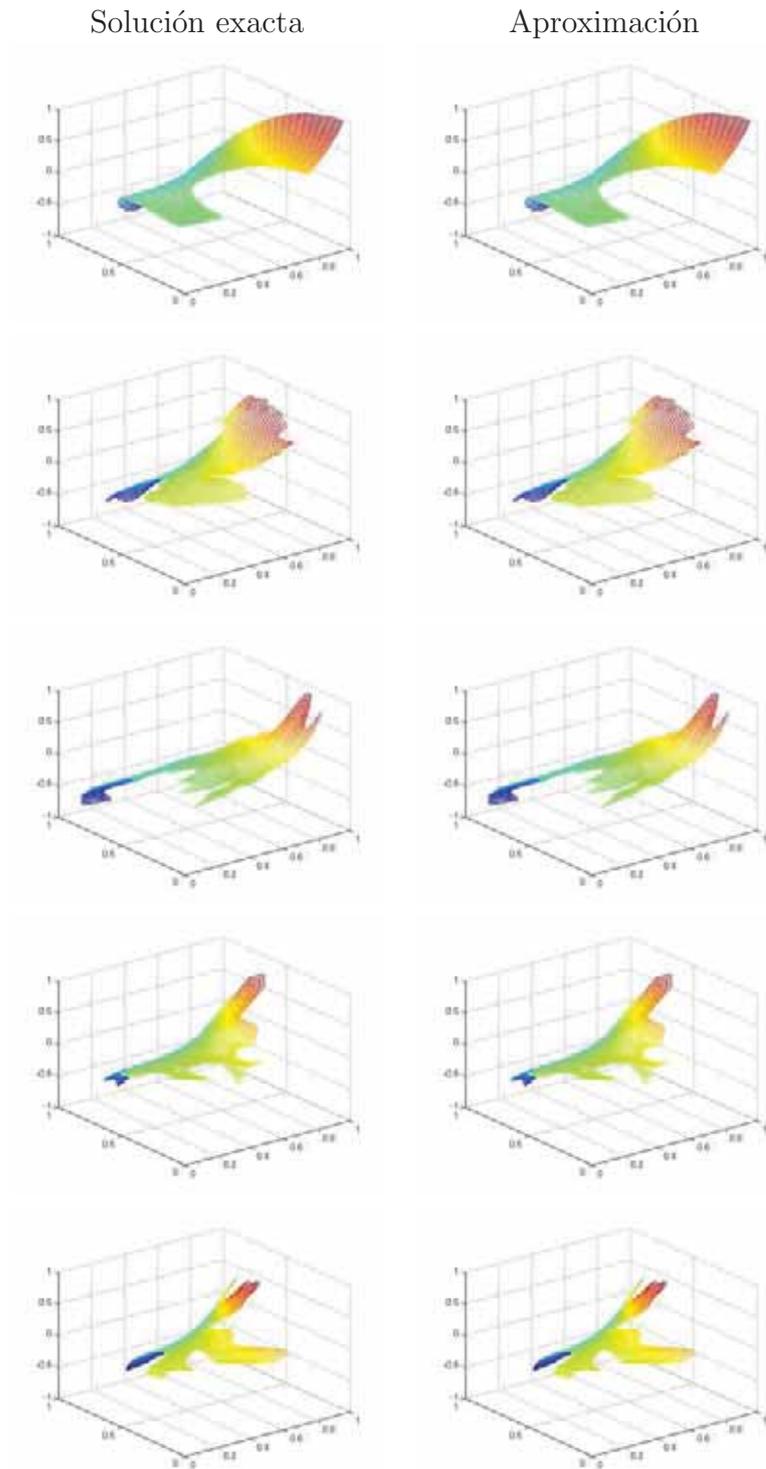


Figura 5.3: En esta figura se muestran las gráficas de los resultados de aproximación del problema Poisson 5.3, en las nubes Annulus 4.5 3721 nodos, Michoacán 4.6 3721 nodos, Inglaterra 4.7 3721 nodos, Ucha 4.8 3721 nodos, Habana 4.9 6561 nodos, con el número de nodos con el cual se obtuvo la mejor aproximación de la ecuación. Aquí se utilizó la primera variante del criterio de los cuadrantes.

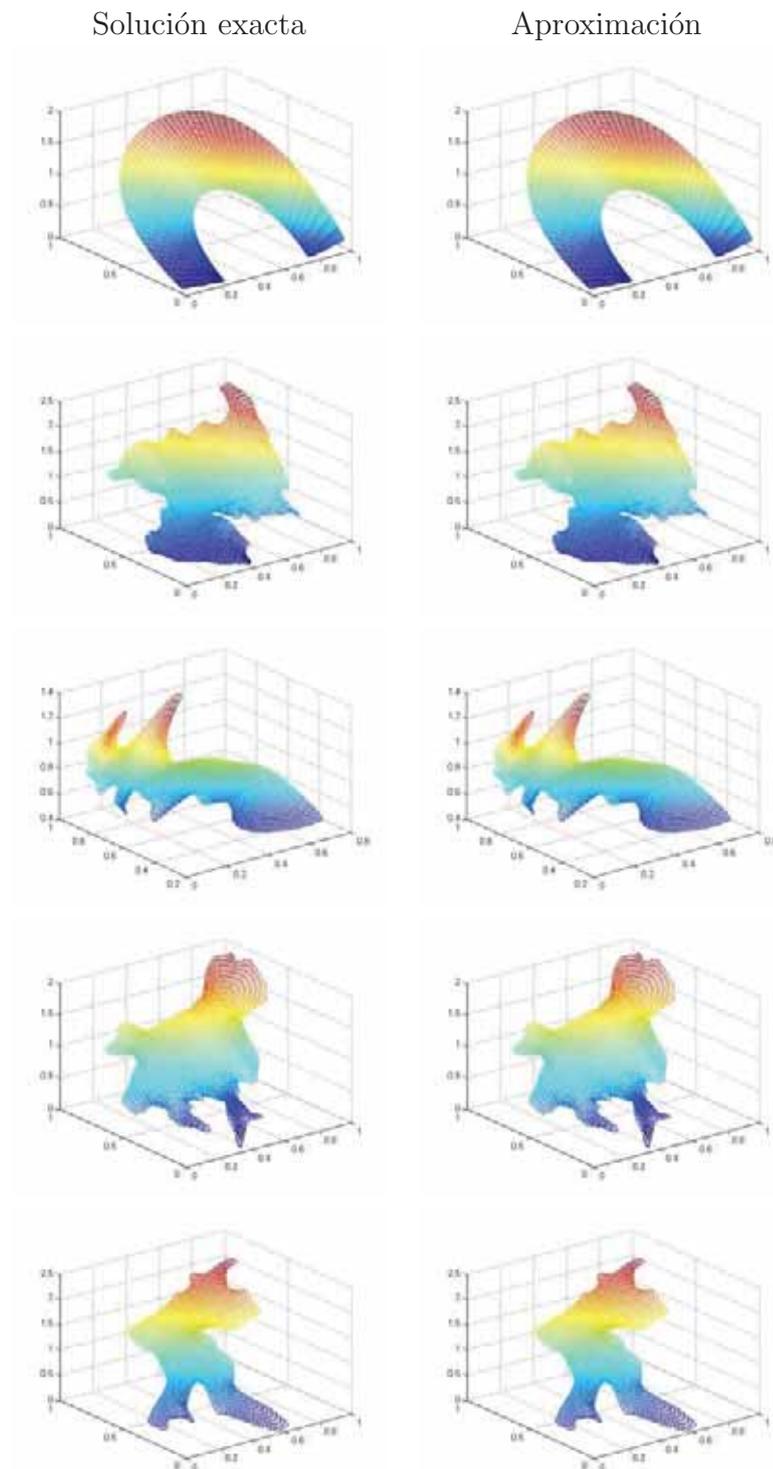


Figura 5.4: En esta figura se muestran las gráficas de los resultados de aproximación del problema Hiperbólica 5.5, en las nubes Annulus 4.5 3721 nodos, Michoacán 4.6 6561 nodos, Inglaterra 4.7 3721 nodos, Ucha 4.8 3721 nodos, Habana 4.9 3721 nodos, con el número de nodos con el cual se obtuvo la mejor aproximación de la ecuación. Aquí se utilizó la primera variante del criterio de los cuadrantes.

nube	nodos	Laplace 5.1	Poisson 5.3	Hiperbólica 5.5	Hiperbólica 5.7
ann	441	1.1580×10^{-16}	0.0015	3.3996×10^{-4}	1.0769×10^{-14}
	1681	2.3944×10^{-16}	7.0294×10^{-4}	9.9322×10^{-5}	3.3839×10^{-13}
	3721	2.0038×10^{-15}	4.2168×10^{-4}	5.1833×10^{-5}	3.7062×10^{-14}
	6561	1.0353×10^{-15}	9.1510×10^{-4}	7.5251×10^{-5}	9.4146×10^{-13}
mich	441	7.5387×10^{-17}	7.9604×10^{-4}	1.6431×10^{-4}	1.5264×10^{-14}
	1681	1.1910×10^{-16}	1.3025×10^{-4}	5.3677×10^{-5}	7.1877×10^{-15}
	3721	1.4109×10^{-16}	4.2451×10^{-5}	2.1101×10^{-4}	1.0704×10^{-14}
	6561	2.1530×10^{-14}	2.0449×10^{-4}	5.6921×10^{-4}	1.3848×10^{-14}
eng	441	7.7496×10^{-17}	3.4185×10^{-4}	7.2173×10^{-5}	1.3438×10^{-15}
	1681	1.2816×10^{-13}	2.8461×10^{-4}	1.6234×10^{-5}	3.5448×10^{-14}
	3721	4.3993×10^{-16}	3.0592×10^{-5}	7.8492×10^{-6}	8.0651×10^{-15}
	6561	1.2721×10^{-15}	1.9837×10^{-4}	7.7045×10^{-5}	5.9641×10^{-14}
uch	441	2.9994×10^{-16}	3.9025×10^{-4}	8.9695×10^{-5}	1.9468×10^{-15}
	1681	5.2514×10^{-16}	1.7684×10^{-4}	8.0022×10^{-6}	2.0550×10^{-14}
	3721	1.4848×10^{-15}	1.3458×10^{-4}	4.1068×10^{-5}	8.7285×10^{-14}
	6561	6.2862×10^{-15}	9.0521×10^{-4}	1.5559×10^{-5}	1.2114×10^{-13}
hab	441	6.8122×10^{-17}	1.4326×10^{-4}	6.4669×10^{-5}	5.9657×10^{-15}
	1681	9.6957×10^{-17}	4.2917×10^{-5}	2.2446×10^{-5}	4.4953×10^{-14}
	3721	3.1815×10^{-16}	1.3243×10^{-4}	6.6223×10^{-6}	7.3440×10^{-14}
	6561	2.9017×10^{-16}	7.9273×10^{-5}	4.4658×10^{-4}	1.3230×10^{-13}

Tabla 5.4: Error cuadrático medio de las aproximaciones de las EDPs tomadas del trabajo de J. J. Benito, utilizando la segunda variante del método de los cuadrantes.

estrellas irregulares también aumento, sin embargo los comportamientos en general fueron muy similares a los anteriores. Se esperaría que al aumentar el número de nodos en las nubes generaría mejores aproximaciones, no fue así, en algunas nubes las mejores aproximaciones son cuando tienen menos puntos y no nos queda más que pensar que las complicaciones en la elección de nodos vistas en la sección 4.2 siguen estando presentes.

Una vez más los errores de la tabla 5.4 que son con la segunda variante del criterio tienen un comportamiento similar a los de la tabla anterior 5.2.2, debido a la complicación de la geometría en las nubes ha aumentado la dificultad en la elección de nodos de este método, aun así hay una mejora de la aproximación en

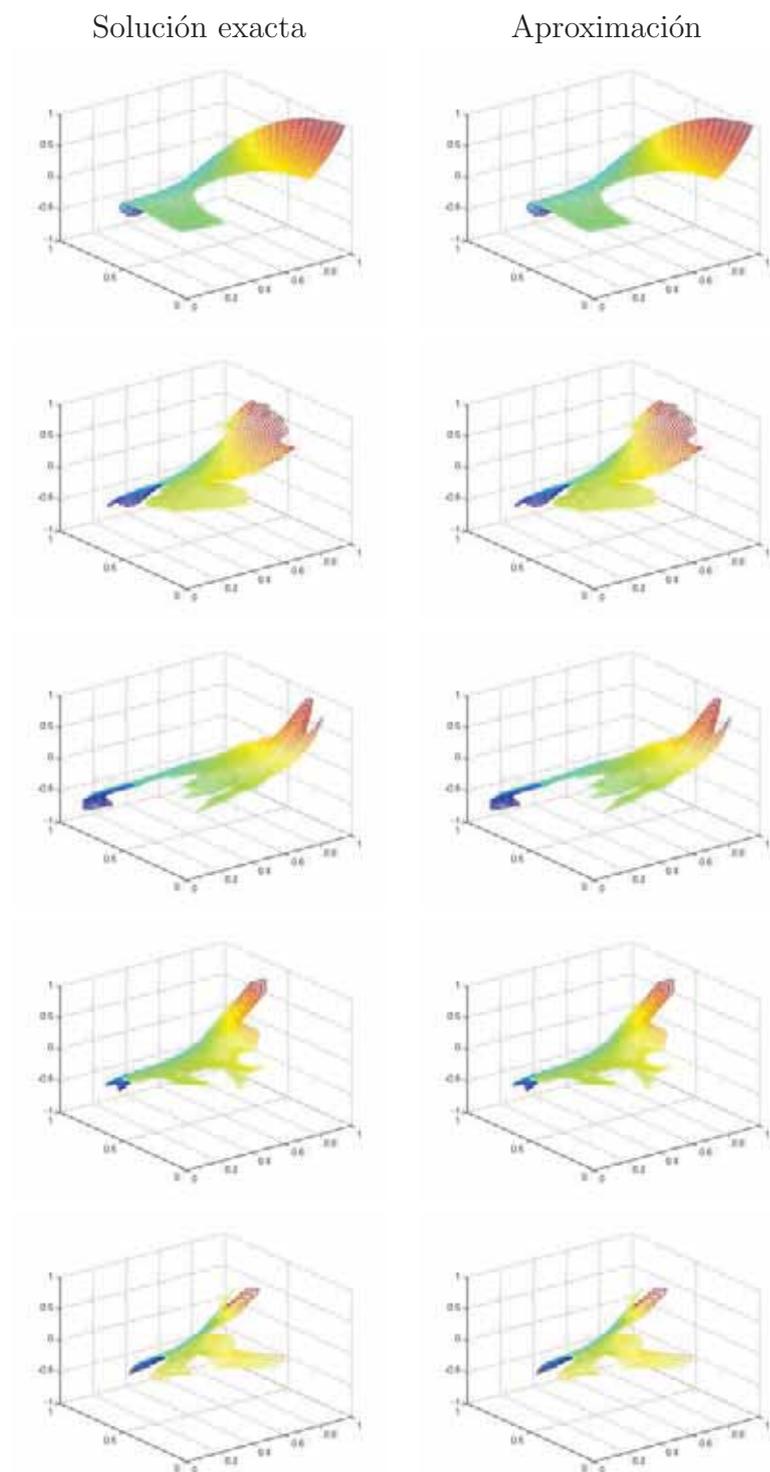


Figura 5.5: En esta figura se muestran las gráficas de los resultados de aproximación del problema Poisson 5.3, en las nubes Annulus 4.5 3721 nodos, Michoacán 4.6 3721 nodos, Inglaterra 4.7 3721 nodos, Ucha 4.8 3721 nodos, Habana 4.9 1681 nodos, con el número de nodos con el cual se obtuvo la mejor aproximación de la ecuación. Aquí se utilizó la segunda variante del criterio de los cuadrantes.

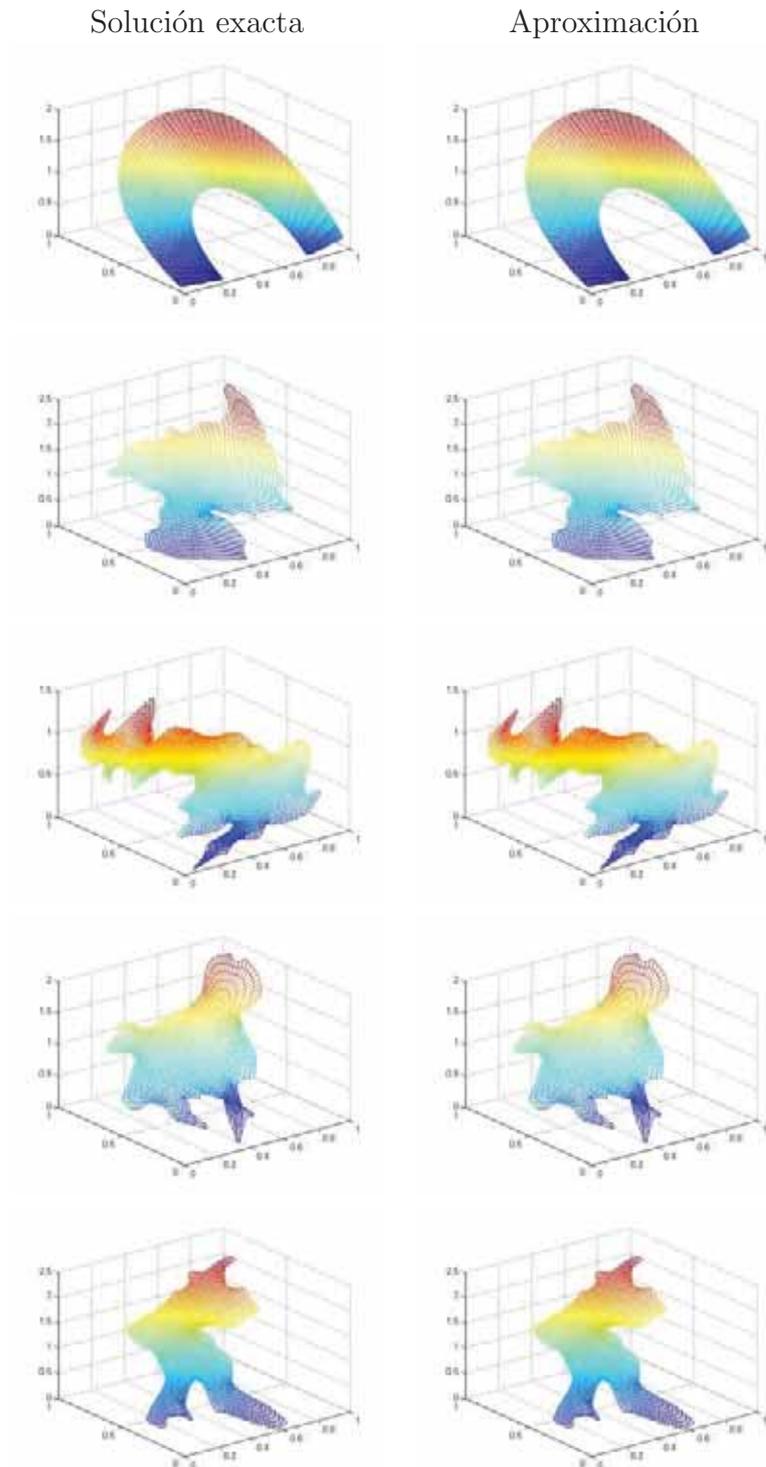


Figura 5.6: En esta figura se muestran las gráficas de los resultados de aproximación del problema Hiperbólica 5.5, en las nubes Annulus 4.5 3721 nodos, Michoacán 4.6 1681 nodos, Inglaterra 4.7 3721 nodos, Ucha 4.8 1681 nodos, Habana 4.9 3721 nodos, con el número de nodos con el cual se obtuvo la mejor aproximación de la ecuación. Aquí se utilizó la segunda variante del criterio de los cuadrantes.

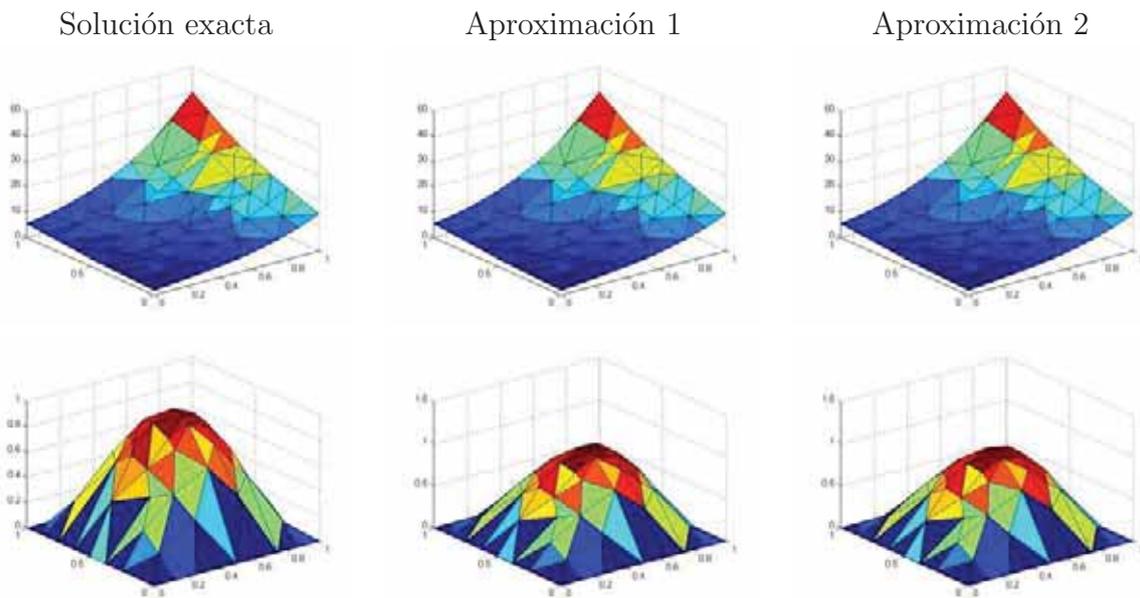


Figura 5.7: En esta figura se muestran las gráficas de los resultados de aproximación de los problemas Poisson 5.3, Hiperbólica 5.5 en la nube 1, en el orden mencionadas. En Aproximación 1 se utilizó la primera variante del criterio de los cuadrantes y en Aproximación la segunda variante.

la nube mich 4.6. Al parecer al aumentar los nodos en las nubes no siempre puede verse una disminución de errores lo que nos dice que la irregularidad de las estrellas afecta las aproximaciones.

5.2.2. Ecuaciones de prueba del grupo de trabajo en las primeras 7 nubes

	nube 1	nube 2	nube 3	nube 4	nube 5	nube 6	nube 7
EDP (5.9)	0.0266	0.0023	0.0033	0.0122	0.0130	0.0048	0.0266
EDP (5.10)	0.0436	0.0429	0.0435	0.0380	0.0367	0.0427	0.0405

Tabla 5.5: Error cuadrático medio de las aproximaciones de las EDPs de prueba en el grupo de trabajo, utilizando la primera variante del método de los cuadrantes.

En las tablas (5.5) y (5.6) regresamos a las primeras 7 nubes pero ahora se complican las EDPs a resolver, en esta tabla se hace visible que la distribución

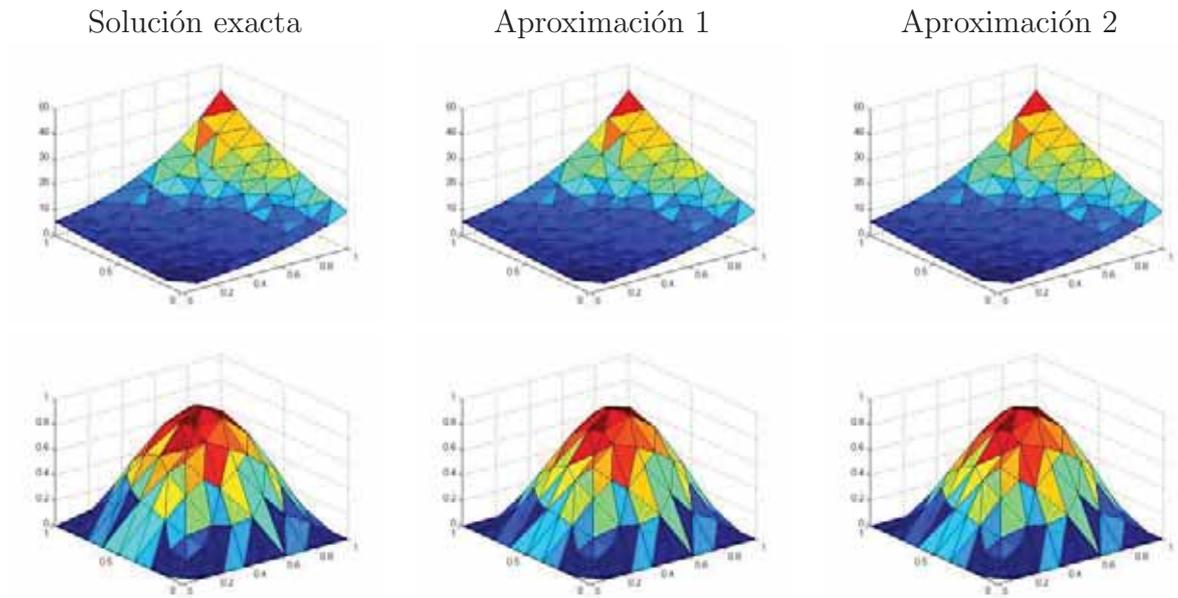


Figura 5.8: En esta figura se muestran las gráficas de los resultados de aproximación de los problemas Poisson 5.3, Hiperbólica 5.5 en la nube 7, en el orden mencionadas. En Aproximación 1 se utilizó la primera variante del criterio de los cuadrantes y en Aproximación la segunda variante.

irregular de los nodos afecta la aproximación de cada una de las EDPs, aunque la mayoría de estas nubes solo son irregulares parcialmente. Aun así los resultados son comparables.

	nube 1	nube 2	nube 3	nube 4	nube 5	nube 6	nube 7
EDP (5.9)	0.0274	0.0051	0.0055	0.0078	0.0233	0.0048	0.0266
EDP (5.10)	0.0371	0.0427	0.0426	0.0381	0.0375	0.0428	0.0380

Tabla 5.6: Error cuadrático medio de las aproximaciones de las EDPs de prueba en el grupo de trabajo, utilizando la primera variante del método de los cuadrantes.

5.2.3. Nubes y problemas de prueba del equipo de trabajo

En las tablas 5.7 y 5.8 muestran los errores de las EDPs utilizadas de prueba en nuestro equipo de trabajo. Es evidente que al complicarse la geometría de las nubes y las EDPs los errores reflejaran las estrellas irregulares aun así los errores obtenidos son comparables.

	nodos	Annulus	Michoacán	Inglaterra	Ucha	Habana
EDP (5.9)	441	0.0046	0.0034	0.0014	0.0029	0.0017
	1681	0.0017	9.2128×10^{-4}	5.1510×10^{-4}	4.8813×10^{-4}	3.4367×10^{-4}
	3721	4.9904×10^{-4}	5.2552×10^{-4}	8.3114×10^{-4}	6.6641×10^{-4}	1.9095×10^{-4}
	6561	0.1373	0.0011	0.1680	0.0042	1.4884×10^{-4}
EDP (5.10)	441	0.0235	0.0138	0.0150	0.0202	0.0106
	1681	0.0233	0.0136	0.0132	0.0197	0.0108
	3721	0.0236	0.0136	0.0128	0.0197	0.0106
	6561	0.0263	0.0134	0.0109	0.0200	0.0108

Tabla 5.7: Error cuadrático medio de las aproximaciones de la EDP (5.9) de prueba en el grupo de trabajo utilizando la primera variante del método de los cuadrantes.

	nodos	Annulus	Michoacán	Inglaterra	Ucha	Habana
EDP (5.9)	441	0.0076	0.0041	0.0013	0.0056	0.0014
	1681	0.0035	0.0014	4.4785×10^{-4}	0.0010	3.0844×10^{-4}
	3721	0.0018	7.4013×10^{-4}	1.2564×10^{-4}	0.0022	7.0754×10^{-4}
	6561	0.0046	0.0018	4.1485×10^{-4}	0.0011	1.1675×10^{-4}
EDP (5.10)	441	0.0233	0.0142	0.0142	0.0199	0.0104
	1681	0.0233	0.0133	0.0120	0.0199	0.0105
	3721	0.0234	0.0138	0.0118	0.0198	0.0104
	6561	0.0241	0.0135	0.0101	0.0193	0.0108

Tabla 5.8: Error cuadrático medio de las aproximaciones de la EDPs de prueba en el grupo de trabajo utilizando la segunda variante del método de los cuadrantes.

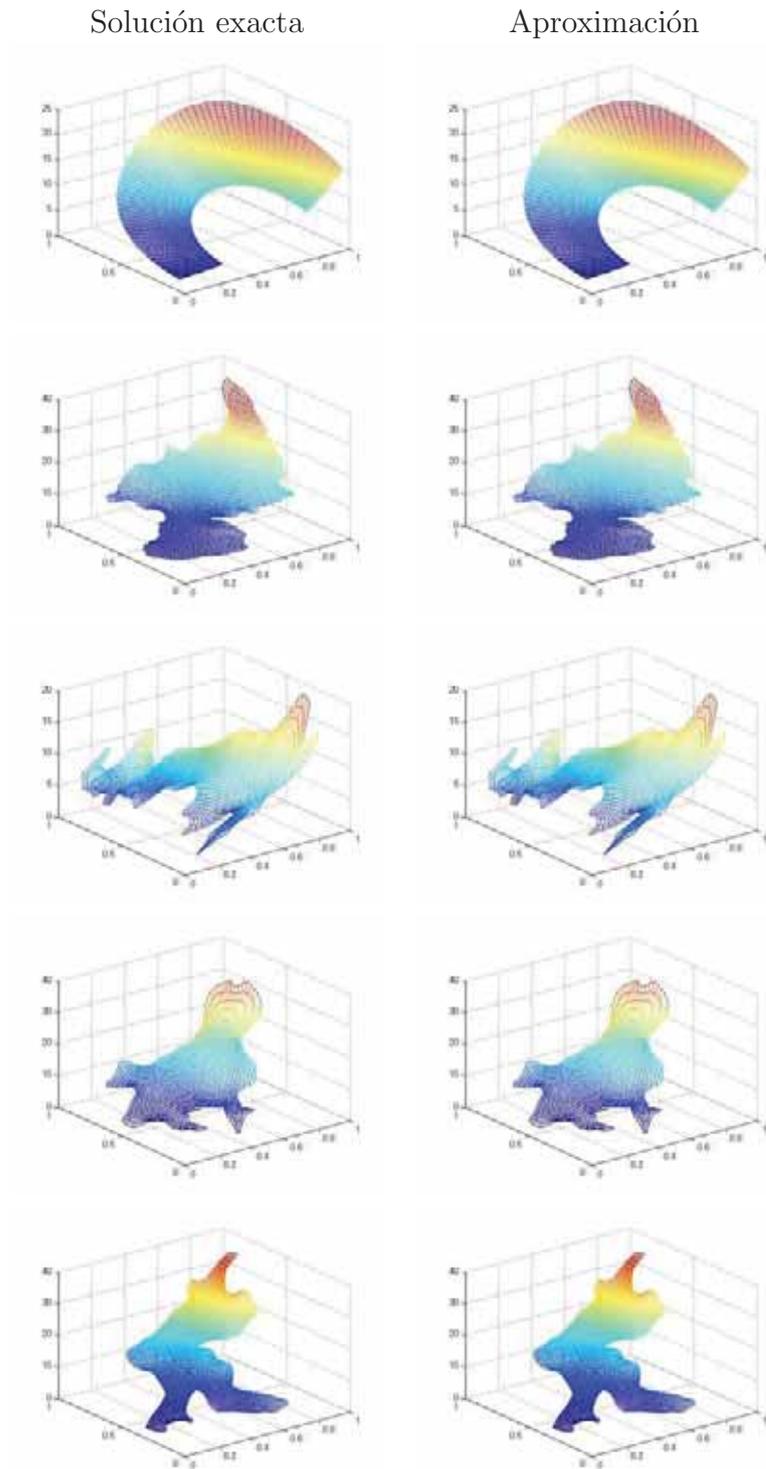


Figura 5.9: En esta figura se muestran las gráficas de los resultados de aproximación de la EDP (5.9), en las nubes Annulus 4.5 3721 nodos, Michoacán 4.6 3721 nodos, Inglaterra 4.7 1681 nodos, Ucha 4.8 1681 nodos, Habana 4.9 6561 nodos, con el número de nodos con el cual se obtuvo la mejor aproximación de la ecuación. Aquí se utilizó la primera variante del criterio de los cuadrantes.

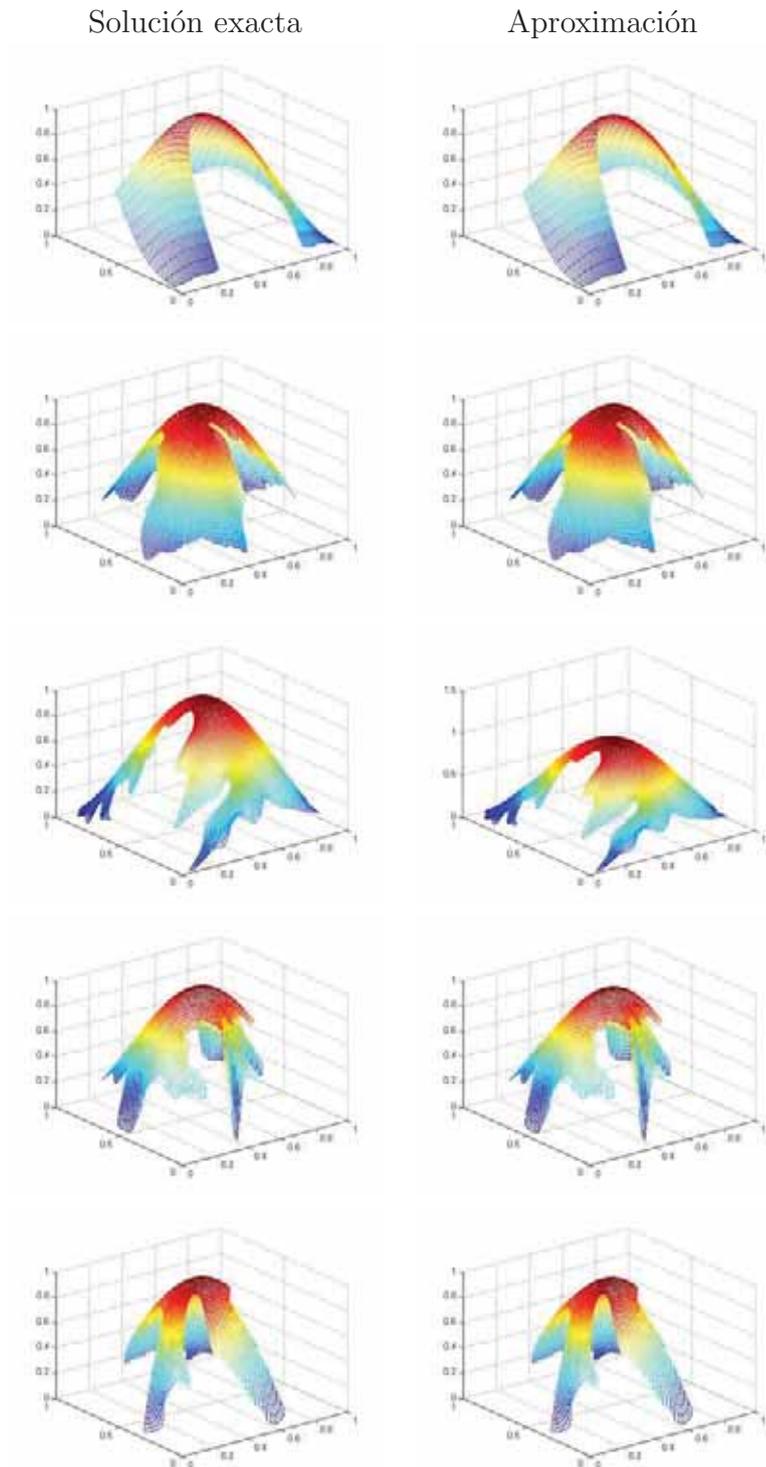


Figura 5.10: En esta figura se muestran las gráficas de los resultados de aproximación de la EDP (5.10), en las nubes Annulus 4.5 1681 nodos, Michoacán 4.6 6561 nodos, Inglaterra 4.7 6561 nodos, Ucha 4.8 3721 nodos, Habana 4.9 3721 nodos, con el número de nodos con el cual se obtuvo la mejor aproximación de la ecuación. Aquí se utilizó la primera variante del criterio de los cuadrantes.

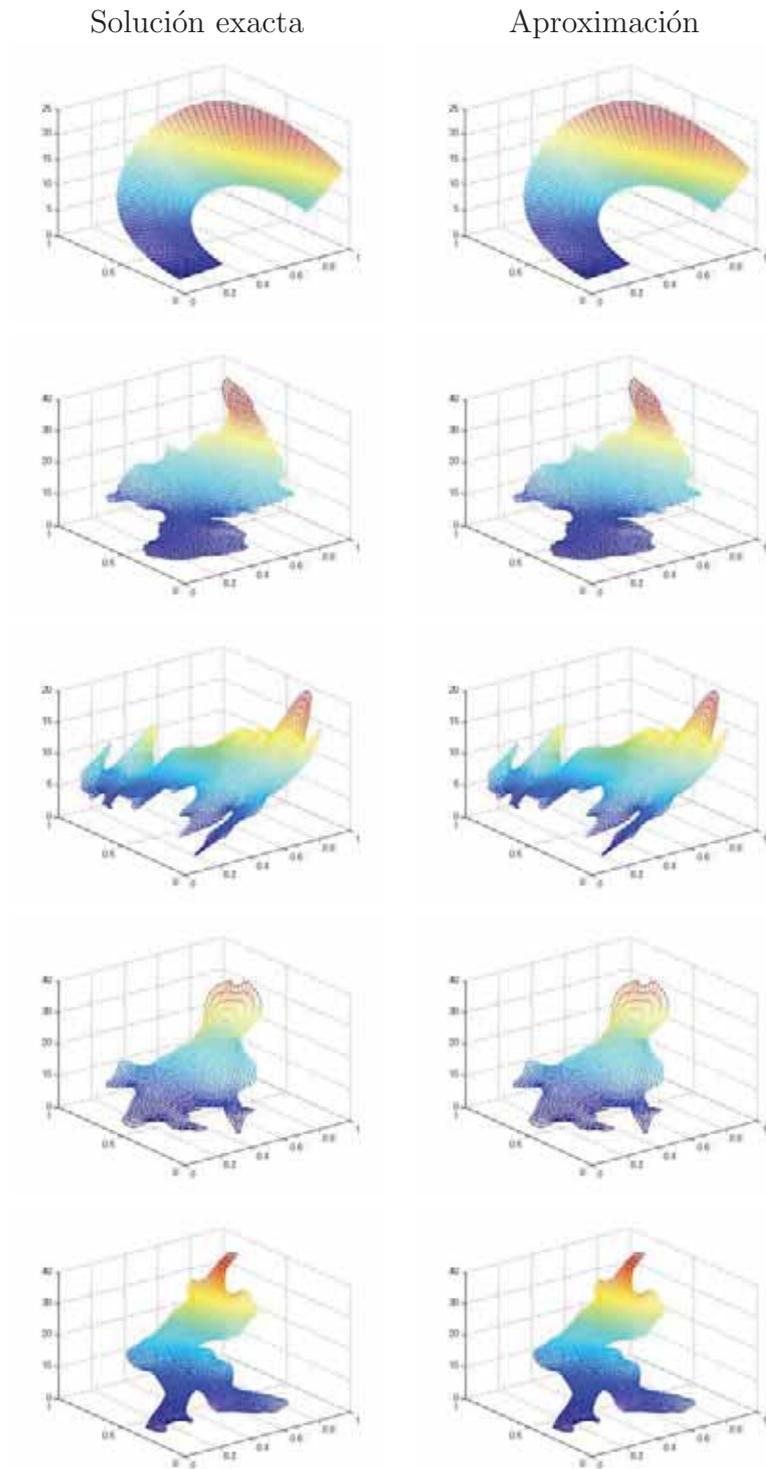


Figura 5.11: En esta figura se muestran las gráficas de los resultados de aproximación de la EDP (5.9), en las nubes Annulus 4.5 3721 nodos, Michoacán 4.6 3721 nodos, Inglaterra 4.7 3721 nodos, Ucha 4.8 1681 nodos, Habana 4.9 6561 nodos, con el número de nodos con el cual se obtuvo la mejor aproximación de la ecuación. Aquí se utilizó la segunda variante del criterio de los cuadrantes.

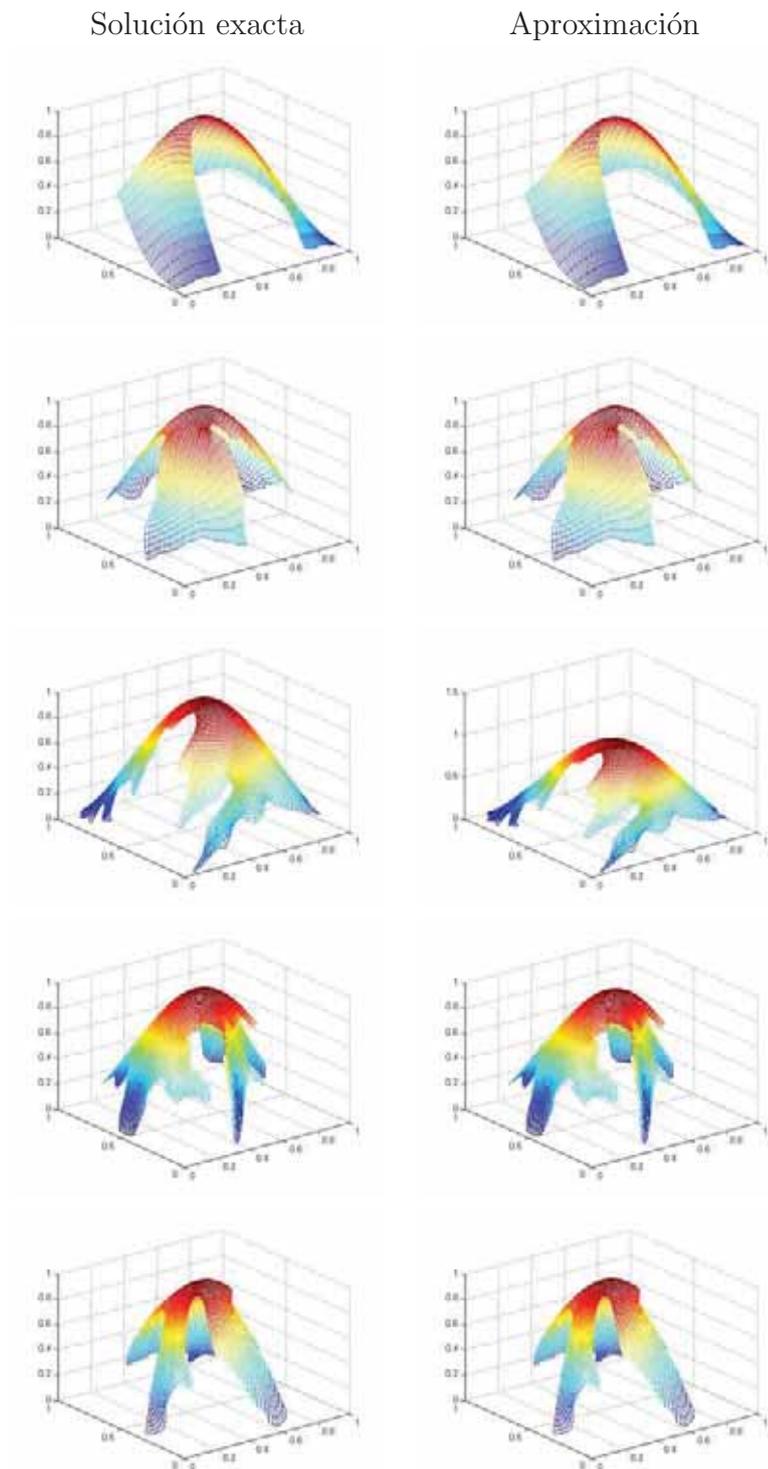


Figura 5.12: En esta figura se muestran las gráficas de los resultados de aproximación de la EDP (5.10), en las nubes Annulus 4.5 1681 nodos, Michoacán 4.6 1681 nodos, Inglaterra 4.7 3721 nodos, Ucha 4.8 6561 nodos, Habana 4.9 3721 nodos, con el número de nodos con el cual se obtuvo la mejor aproximación de la ecuación. Aquí se utilizó la segunda variante del criterio de los cuadrantes.

	Euler $\Delta t = 0.0004$	Crank-Nicolson $\Delta t = 0.002$
nube 1	4.1342×10^{-5}	4.1723×10^{-5}
nube 2	4.6335×10^{-6}	4.8766×10^{-6}
nube 3	4.4279×10^{-6}	4.6616×10^{-6}
nube 4	6.4264×10^{-6}	6.8565×10^{-6}
nube 5	1.8900×10^{-5}	1.1996×10^{-5}
nube 6	1.1462×10^{-6}	1.8537×10^{-6}
nube 7	7.0158×10^{-6}	7.0158×10^{-6}

Tabla 5.9: Error cuadrático medio de las aproximaciones de la ecuación de difusión utilizando la primera variante del método de los cuadrantes.

	Euler $\Delta t = 0.0004$	Crank-Nicolson $\Delta t = 0.002$
nube 1	4.3055×10^{-6}	4.7933×10^{-6}
nube 2	1.0259×10^{-6}	2.0762×10^{-6}
nube 3	9.4777×10^{-7}	2.0282×10^{-6}
nube 4	3.7543×10^{-6}	4.3689×10^{-6}
nube 5	3.9186×10^{-6}	4.5596×10^{-6}
nube 6	1.0182×10^{-6}	2.0883×10^{-6}
nube 7	7.0158×10^{-6}	5.6295×10^{-6}

Tabla 5.10: Error cuadrático medio de las aproximaciones de la ecuación de difusión utilizando la segunda variante del método de los cuadrantes.

En esta última parte de resultados mostramos los errores de aproximación de la ecuación de difusión utilizando el método de Euler y el método de Crank–Nicolson. Para el método de Euler se utilizó un tamaño de paso $\Delta t = 0.0004$, mientras que para el método de Crank–Nicolson se utilizó $\Delta t = 0.002$.

Los resultados a mostrar son los errores de aproximación de la EDP en las primeras 7 nubes. También hicimos el cálculo de errores en las nubes restantes pero debido a que la elección de nodos no es lo suficientemente adecuada para este tipo de nubes, los errores son grandes. Una vez más vemos la necesidad de buscar un nuevo método adecuado de elección de nodos.

5.3. Conclusiones

Se han utilizado dos métodos de elecciones de nodos para la aproximación de las EDPs, anteriormente ya vimos los problemas que se presentan cuando la geometría de las nubes se complica; por tanto podemos concluir lo siguiente.

La geometría de las nubes es fundamental para los métodos de elección de nodos, ya que se presentan estrellas muy irregulares. Una manera de mejorar la elección en este tipo de nubes puede ser la distribución de los nodos en las regiones a trabajar, ya que si las estrellas mejoran también mejoran los resultados.

Los métodos de elección de nodos utilizados son buenos en nubes de poca irregularidad sin embargo es necesario buscar nuevos y mejores métodos que tengan un mejor control en regiones muy irregulares.

Las aproximaciones de las EDPs tomadas del trabajo de J. J. Benito et al [4] en las nubes del mismo, son buenas ya que en algunos de los problemas se superó el orden de magnitud de los errores presentados en el trabajo mencionado.

Las aproximaciones de las EDPs en las nubes de prueba en de nuestro equipo de trabajo son una motivación para la búsqueda de un método de elección de nodos que garantice la convergencia a la solución.

En general los métodos de elección de nodos utilizados son buenos pero falta pulirlos para obtener los resultados deseados en cada región de trabajo que se presente. Este trabajo es un primer intento para el desarrollo de un método más robusto y una introducción para poder entender los problemas que se presentan al momento de su implementación.

Bibliografía

- [1] P. Barrera-Sanchez, L. Castellanos-Noda, A. Pérez-Domínguez, *Métodos variacionales discretos para la generación de las mallas*, DGAPA-UNAM, (1994).
- [2] P. Barrera-Sanchez, A. Pérez-Domínguez, L. Castellanos-Noda, *Curvilinear coordinates system generation over plane irregular regions*, Vínculos Matemáticos, No. 133, Facultad de Ciencias, UNAM, (1992).
- [3] P. Barrera-Sanchez, L. Castellanos-Noda, F. Domínguez-Mota, G. González-Flores, A. Pérez-Domínguez, *Adaptative discrete harmonic grid generation*, Mathematics and Computer Simulations, Vol. 79, (2009), 1802–1809.
- [4] J. J. Benito, F. Ureña, L. Gavete, *Influence of several factors in the generalized finite difference method*, Applied Mathematical Modelling, Vol. 25, (2001), 1039–1053.
- [5] A. Carrillo-Ledesma, O. Mendoza Bernal, *Introducción al Método de Diferencias Finitas y su Implementación Computacional*.
http://mmc.geofisica.unam.mx/femp/Ejemplitos/F_M/FDM
- [6] F. J. Domínguez-Mota, P. M. Fernández-Valdez, E. Ruiz-Díaz, G. Tinoco-Guerrero, J. G. Tinoco-Ruíz, *An heuristic finite difference scheme on irregular plane regions*, Applied Mathematical Sciences, No. 14, Vol. 8, (2014), 671–683.
- [7] Ivanenko S. A., *Adaptative grids and grids on surfaces*, Comput. Maths. Math. Phys., No. 9, Vol. 33, (1993), 8.1–8.41.
- [8] P. Knupp, S. Steinberg, *Fundamentals of Grid Generation*, CRC Press., U. S. A. (1994).
- [9] P. M. Fernández-Valdez, *Un esquema de diferencias finitas para la solución numérica de ecuaciones elípticas en regiones irregulares del plano*, Tesis de licenciatura, Facultad de Ciencias Físico-Matemáticas, UMSNH, 2013.

- [10] P. S. Jensen, *Finite difference technique for variable grids*, Comput. Struct. Vol. 2, (1972), 17–29.
- [11] J. Jost, *Partial Differential Equations*, Springer, New York, 2013.
- [12] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations, Steady-State and Time-Dependent Problems*, SIAM, 2007.
- [13] T. Lizka, J. Orkisz, *The finite difference method at arbitrary irregular grids and its application in applied mechanics*, Comput. Struct. Vol. 11, (1980), 83–95.
- [14] P. J. Olver, *Introduction to Partial Differential Equations*, Springer, Switzerl, 2014.
- [15] N. Perrone, R. Kao, *A general finite difference method for arbitrary meshes*, Comput. Struct. Vol. 5, (1975), 45–58.
- [16] G. Tinoco–Guerrero, *Solución de la ecuación de advección empleando mallas estructuradas sobre regiones planas irregulares utilizando un esquema de diferencias*, Tesis de licenciatura, Facultad de Ciencias Físico–Matemáticas, UMSNH, 2014.
- [17] J. G. Tinoco–Ruiz. *Funcionales Discretos para la generación de mallas suaves y convexas sobre regiones planas irregulares*, Tesis doctoral, CIMAT, (1997).
- [18] A. Tveito, R. Winther, *Introduction to Partial Differential Equations. A Computational Approach*, Springer, Germany, 2005.
- [19] W. Rudin, *Principles of Mathematical Analysis*, Second Edition, 1976.
- [20] E. Zuazua, *Métodos Numéricos de resolución de Ecuaciones en Derivadas Parciales*.
https://www.uam.es/personal_pdi/ciencias/ezuazua/informweb/notas-05_065.pdf
- [21] UNAMALLA Versión 4.0. *An automatic package for numerical grid generation*. Website: http://www.mathmoo.unam.mx/unamalla/home_e.html. Última versión publicada en Junio, 2011.
- [22] MATLAB R2013a. *The language of technical computing*. Website: <http://www.mathworks.com/products/matlab/>

Apéndice A

Programas para la selección de nodos.

Los siguientes implementaciones son las propuestas que se hicieron de las variantes del método de los cuadrantes para la elección de nodos en las nubes, se encuentran programados en el lenguaje de programación Matlab [22].

Como veremos lo único que necesitan de entrada es la nube de puntos, que es una matriz de dimensiones $N \times 3$ donde N es el número total de nodos de la nube. La primera columna de las nubes posee la coordenada en x , la segunda la coordenada en y y la tercera nos dice si es frontera, esto para su respectivo nodo.

A.1. Primera variante

Primera variante del criterio y los subprogramas que utiliza.

```
%%% rutina para buscar vecinos usando el metodo de los cuadrantes
%%% recibe una matriz con los valores de los puntos en x y y en las
%%% primeras dos columnas y en la columna 3; 1 si es frontera,
%%% 0 si no lo es.
```

```
function [vec1] = cuadrantesvariante1(nube)
```

```
%%% el valor de c es el tamaño de lado del cuadrante.
[N,n]=size(nube);
c=3*(1/N);
c1=c;
di=[1 1];%%valor de las distancias iniciales
pos=[0 0];%%valor de las posiciones iniciales
```

```

%%valores de los puntos en x y y
x=nube(:,1);
y=nube(:,2);
vec=zeros(N,9);
for i=1:N

%primer cuadrante

u=1;%%numero de cuadrante
xv1 = [x(i) x(i)+c x(i)+c x(i)]';
yv1 = [y(i) y(i) y(i)+c y(i)+c]';
xv1 = [xv1 ; xv1(1)]; yv1 = [yv1 ; yv1(1)];
vx=x(i);%%valores en x y y del nodo actual.
vy=y(i);
in = inpolygon(x,y,xv1,yv1);
vec(i,1)=i;

[pos]=prueba2n(N,x,y,in,i,vec,pos,di,1);
%%el 1 indica si que pos aun tiene valores
%%iniciales cuando tiene un 2 indica que tiene valor inicial solo
en la segunda entrada

while pos(1)==0
    if(2*c<.7)
        [in,c]=masgrande2(x,y,c,vx,vy,u);
        ui=u;
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,1);
    else
        [in]=sigcuad2(x,y,c,vx,vy,ui);
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,1);
        ui=ui+1;
    end
end
c=c1;
while pos(2)==0
    if(2*c<.7)
        [in,c]=masgrande2(x,y,c,vx,vy,u);
        ui=u;
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,2);
    end
end

```

```

else
    [in]=sigcuad2(x,y,c,vx,vy,ui);
    [pos]=prueba2n(N,x,y,in,i,vec,pos,di,2);
    ui=ui+1;
end
end
vec(i,2:3)=pos;
pos=[0 0];

%segundo cuadrante

u=2;
xv2 = [x(i) x(i)-c x(i)-c x(i)]';
yv2 = [y(i) y(i) y(i)+c y(i)+c]';
xv2 = [xv2 ; xv2(1)]; yv2 = [yv2 ; yv2(1)];
in = inpolygon(x,y,xv2,yv2);
[pos]=prueba2n(N,x,y,in,i,vec,pos,di,1);
c=c1;
while pos(1)==0
    if(2*c<.7)
        [in,c]=masgrande2(x,y,c,vx,vy,u);
        ui=u;
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,1);
    else
        [in]=sigcuad2(x,y,c,vx,vy,ui);
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,1);
        ui=ui+1;
    end
end
end
c=c1;
while pos(2)==0
    if(2*c<.7)
        [in,c]=masgrande2(x,y,c,vx,vy,u);
        ui=u;
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,2);
    else
        [in]=sigcuad2(x,y,c,vx,vy,ui);
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,2);
        ui=ui+1;
    end
end

```

```

    end
end
vec(i,4:5)=pos;
pos=[0 0];

%tercer cuadrante

u=3;
xv3 = [x(i) x(i)-c x(i)-c x(i)]';
yv3 = [y(i) y(i) y(i)-c y(i)-c]';
xv3 = [xv3 ; xv3(1)]; yv3 = [yv3 ; yv3(1)];
in = inpolygon(x,y,xv3,yv3);
[pos]=prueba2n(N,x,y,in,i,vec,pos,di,1);
c=c1;
while pos(1)==0
    if(2*c<.7)
        [in,c]=masgrande2(x,y,c,vx,vy,u);
        ui=u;
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,1);
    else
        [in]=sigcuad2(x,y,c,vx,vy,ui);
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,1);
        ui=ui+1;
    end
end
end
c=c1;
while pos(2)==0
    if(2*c<.7)
        [in,c]=masgrande2(x,y,c,vx,vy,u);
        ui=u;
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,2);
    else
        [in]=sigcuad2(x,y,c,vx,vy,ui);
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,2);
        ui=ui+1;
    end
end
end
vec(i,6:7)=pos;

```

```

pos=[0 0];

%cuarto cuadrante

u=4;
xv4 = [x(i) x(i)+c x(i)+c x(i)]';
yv4 = [y(i) y(i) y(i)-c y(i)-c]';
xv4 = [xv4 ; xv4(1)]; yv4 = [yv4 ; yv4(1)];
in = inpolygon(x,y,xv4,yv4);
[pos]=prueba2n(N,x,y,in,i,vec,pos,di,1);
c=c1;
while pos(1)==0
    if(2*c<.7)
        [in,c]=masgrande2(x,y,c,vx,vy,u);
        ui=u;
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,1);
    else
        [in]=sigcuad2(x,y,c,vx,vy,ui);
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,1);
        ui=ui+1;
    end
end
c=c1;
while pos(2)==0
    if(2*c<.7)
        [in,c]=masgrande2(x,y,c,vx,vy,u);
        ui=u;
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,2);
    else
        [in]=sigcuad2(x,y,c,vx,vy,ui);
        [pos]=prueba2n(N,x,y,in,i,vec,pos,di,2);
        ui=ui+1;
    end
end
vec(i,8:9)=pos;
pos=[0 0];
end
vec1=[vec nube(:,3)];

```

```

function [pos]=prueba2n(N,x,y,in,a,vec,pos,di,v)
    d=zeros(N,2);
    if(v==1)
        p=1;
        for i=1:N;
            if(in(i)==1 && i~=vec(a,1)&& i~=vec(a,2)&& i~=vec(a,3)&& i~=vec(a,4)&&
            i~=vec(a,5) && i~=vec(a,6)&& i~=vec(a,7)&& i~=vec(a,8)&& i~=vec(a,9))
                d(p,1)=sqrt((x(i)-x(a))^2+(y(i)-y(a))^2);
                if (d(p,1)<=di(1))
                    di(1)=d(p,1);
                    pos(1)=i;
                end
                p=p+1;
            end
        end

        p=1;
        for i=1:N

            if(in(i)==1&& i~=pos(1) && i~=vec(a,1) && i~=vec(a,2)&& i~=vec(a,3)&&
            i~=vec(a,4)&& i~=vec(a,5) && i~=vec(a,6)&& i~=vec(a,7)&& i~=vec(a,8)&&
            i~=vec(a,9))
                d(p,2)=sqrt((x(i)-x(a))^2+(y(i)-y(a))^2);
                if (d(p,2)<=di(2))
                    di(2)=d(p,2);
                    pos(2)=i;
                end
                p=p+1;
            end
        end
    end
    if(v==2)
        p=1;

        for i=1:N
            if(in(i)==1&& i~=pos(1) && i~=vec(a,1) && i~=vec(a,2)&& i~=vec(a,3)&&
            i~=vec(a,4)&& i~=vec(a,5) && i~=vec(a,6)&& i~=vec(a,7)&& i~=vec(a,8)&&
            i~=vec(a,9))
                d(p,2)=sqrt((x(i)-x(a))^2+(y(i)-y(a))^2);
                if (d(p,2)<=di(2))

```

```
    di(2)=d(p,2);  
    pos(2)=i;  
end  
    p=p+1;  
end  
end  
end
```

```

function [in,c]=masgrande2(X,Y,c,vx,vy,u)
if(u==1)
    c=2*c;

    xv1 = [vx vx+c vx+c vx]';
    yv1 = [vy vy vy+2*c vy+2*c]';
    xv1 = [xv1 ; xv1(1)]; yv1 = [yv1 ; yv1(1)];
    in1 = inpolygon(X,Y,xv1,yv1);

    lix = [vx vx-.0001 vx-.0001 vx]';
    liy = [vy vy vy+2*c vy+2*c]';
    lix = [lix ; lix(1)]; liy = [liy ; liy(1)];

    in2 = inpolygon(X,Y,lix,liy);
    in=quitar(in1,in2);
end

if(u==2)
    c=2*c;

    xv1 = [vx vx-c vx-c vx]';
    yv1 = [vy vy vy+c vy+c]';
    xv1 = [xv1 ; xv1(1)]; yv1 = [yv1 ; yv1(1)];
    in1 = inpolygon(X,Y,xv1,yv1);

    lix = [vx vx-c vx-c vx]';
    liy = [vy vy vy-.0001 vy-.0001]';
    lix = [lix ; lix(1)]; liy = [liy ; liy(1)];
    in2 = inpolygon(X,Y,lix,liy);
    in=quitar(in1,in2);
end

if(u==3)
    c=2*c;

    xv1 = [vx vx-c vx-c vx]';
    yv1 = [vy vy vy-c vy-c]';
    xv1 = [xv1 ; xv1(1)]; yv1 = [yv1 ; yv1(1)];
    in1 = inpolygon(X,Y,xv1,yv1);

```

```
    lix = [vx vx+.0001 vx+.0001 vx]';
    liy = [vy vy vy-c vy-c]';
    lix = [lix ; lix(1)]; liy = [liy ; liy(1)];
    in2 = inpolygon(X,Y,lix,liy);
    in=quitar(in1,in2);
end

if(u==4)
    c=2*c;

    xv1 = [vx vx+c vx+c vx]';
    yv1 = [vy vy vy-c vy-c]';
    xv1 = [xv1 ; xv1(1)]; yv1 = [yv1 ; yv1(1)];
    in1 = inpolygon(X,Y,xv1,yv1);

    lix = [vx vx+c vx+c vx]';
    liy = [vy vy vy+.0001 vy+.0001]';
    lix = [lix ; lix(1)]; liy = [liy ; liy(1)];
    in2 = inpolygon(X,Y,lix,liy);
    in=quitar(in1,in2);
end
```

```

%%%cambia de cuadrante
function [in]=sigcuad2(X,Y,c,vx,vy,u)

if(u==1)
    xv1 = [vx vx-c vx-c vx]';
    yv1 = [vy vy vy+c vy+c]';
    xv1 = [xv1 ; xv1(1)]; yv1 = [yv1 ; yv1(1)];
    in1 = inpolygon(X,Y,xv1,yv1);

    lix = [vx vx-c vx-c vx]';
    liy = [vy vy vy-.0001 vy-.0001]';
    lix = [lix ; lix(1)]; liy = [liy ; liy(1)];
    in2 = inpolygon(X,Y,lix,liy);
    in=quitar(in1,in2);
end

if(u==2)
    xv1 = [vx vx-c vx-c vx]';
    yv1 = [vy vy vy-c vy-c]';
    xv1 = [xv1 ; xv1(1)]; yv1 = [yv1 ; yv1(1)];
    in1 = inpolygon(X,Y,xv1,yv1);

    lix = [vx vx+.0001 vx+.0001 vx]';
    liy = [vy vy vy-c vy-c]';
    lix = [lix ; lix(1)]; liy = [liy ; liy(1)];
    in2 = inpolygon(X,Y,lix,liy);
    in=quitar(in1,in2);
end

if(u==3)
    xv1 = [vx vx+c vx+c vx]';
    yv1 = [vy vy vy-c vy-c]';
    xv1 = [xv1 ; xv1(1)]; yv1 = [yv1 ; yv1(1)];
    in1 = inpolygon(X,Y,xv1,yv1);

    lix = [vx vx+c vx+c vx]';
    liy = [vy vy vy+.0001 vy+.0001]';
    lix = [lix ; lix(1)]; liy = [liy ; liy(1)];
    in2 = inpolygon(X,Y,lix,liy);
    in=quitar(in1,in2);
end

```

```
end

if(u==4)
    xv1 = [vx vx+c vx+c vx]';
    yv1 = [vy vy vy+c vy+c]';
    xv1 = [xv1 ; xv1(1)]; yv1 = [yv1 ; yv1(1)];
    in1 = inpolygon(X,Y,xv1,yv1);

    lix = [vx vx-.0001 vx-.0001 vx]';
    liy = [vy vy vy+c vy+c]';
    lix = [lix ; lix(1)]; liy = [liy ; liy(1)];
    in2 = inpolygon(X,Y,lix,liy);
    in=quitar(in1,in2);
end

if(u>4)
    u=mod(u,4)+1;
    in=sigcuad(X,Y,c,vx,vy,u);
end
```

```
function [in]=quitar(in1,in2)
    [N,n]=size(in1);
    in=zeros(N,1);
    for i=1:N
        if in1(i)==in2(i)
            in(i)=0;
        else
            in(i)=1;
        end
    end
end
```

A.2. Segunda variante

El siguiente programa es la segunda variante del criterio y este programa fue implementado por el Dr. José Gerardo Tinoco Ruíz.

```

function A = cuadrantesvariante2(nube)
    %La entrada es una nube de puntos en el plano
    %La salida es una matriz A tal que
    %la fila i contiene los vecinos del punto i, usando el criterio de
los
    %cuadrantes.
    %A(i,1)=i
    N=size(nube,1);
    x=nube(:,1); y=nube(:,2);
    A=zeros(N,9);

for i=1:N
    A(i,1)=i;
    %Calculamos vectores de diferencias en ejes y distancias

for j=1:N
    dx(j)=x(j)-x(i);
    dy(j)=y(j)-y(i);
    dist(j)=sqrt(dx(j)^2+dy(j)^2);
end
    %Ordenamos ascendentemente de acuerdo a las distancias
    [d,ii]=sort(dist);
    %asignamos vecinos por cuadrante:
    k=1;
    ncuad(1)=0;
    ncuad(2)=0;
    ncuad(3)=0;
    ncuad(4)=0;
    j=1;
    nousados=[];
    d=d(2:N);ii=ii(2:N);

while ((k<10)&(j<N))
    flag=0;%default suponiendo se usara el punto
    if((dx(ii(j))>0) & (dy(ii(j))>=0))

```

```
if(ncuad(1) <2)
    ncuad(1)=ncuad(1)+1;
    k=k+1;
    A(i,k)=ii(j);
else
    flag=1;
end
elseif((dx(ii(j))<=0) & (dy(ii(j))>0))
if(ncuad(2) <2)
    ncuad(2)=ncuad(2)+1;
    k=k+1;
    A(i,k)=ii(j);
else
    flag=1;
end
elseif((dx(ii(j))<0) & (dy(ii(j))<=0))
if(ncuad(3) <2)
    ncuad(3)=ncuad(3)+1;
    k=k+1;
    A(i,k)=ii(j);
else
    flag=1;
end
else
if(ncuad(4) <2)
    ncuad(4)=ncuad(4)+1;
    k=k+1;
    A(i,k)=ii(j);
else
    flag=1;
end
end
if(flag==1)
    nousados=[nousados ii(j)];
end
    j=j+1;
end
if(k<9)
    A(i,k+1:9)=nousados(1:9-k);
end
```

```
end
```

```
    A=[A nube(:,3)];
```