

Capítulo 10

Ecuaciones en derivadas parciales I: MATLAB PDE toolbox

Contenidos del capítulo

10.1. Bases del método de los elementos finitos	142
10.2. Usando la interfaz gráfica de la PDE toolbox	143
10.2.1. Menús	144
10.2.2. Barra de herramientas	145
10.3. Definición de geometrías complicadas	146
10.3.1. Creando esquinas redondeadas	147
10.4. Método sugerido de modelado	148
10.5. Ejemplos de problemas elípticos: la ecuación de Poisson .	149
10.6. Ejemplos de problemas parabólicos: la ecuación del calor	149
10.7. Ejemplos de problemas hiperbólicos: la ecuación de ondas	151
10.8. Ejemplos de problemas de autovalores: modos de vibración	152

Muchos de los problemas estudiados por diversas disciplinas de la ingeniería como la mecánica de medios continuos, el electromagnetismo, la transferencia de calor . . . , involucran el estudio de magnitudes que evolucionan no solamente en tiempo sino también en las variables espaciales. Por esta razón, la formación de un ingeniero no debe cubrir únicamente el campo de las ecuaciones diferenciales ordinarias, sino también el modelado de sistemas mediante ecuaciones en derivadas parciales. Sin embargo, en muy pocas ocasiones puede obtenerse una solución analítica a estos problemas, especialmente cuando el dominio computacional presenta una geometría complicada. Aunque existen varios enfoques distintos para la resolución numérica de estos problemas, quizá el más extendido en la comunidad ingenieril es el método de los *elementos finitos*. En este capítulo analizamos las funcionalidades que MATLAB incorpora en este campo mediante su toolbox para ecuaciones en derivadas parciales.

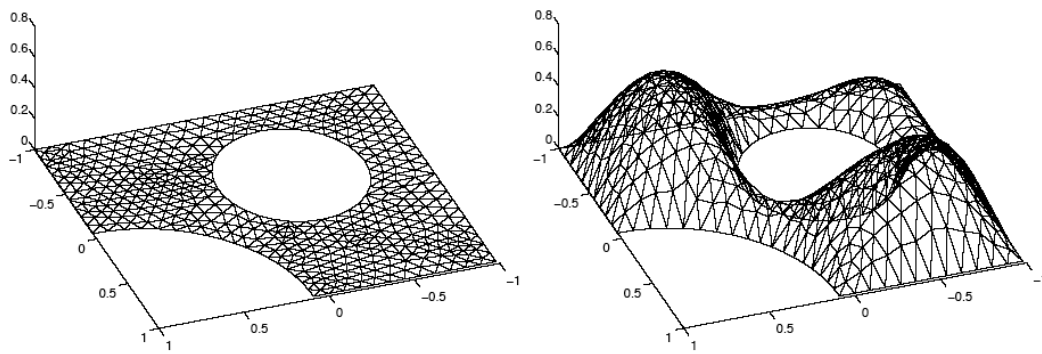


Figura 10.1: (Izquierda) Malla generada con elementos triangulares. (Derecha) Función lineal continua a trozos sobre esa malla.

10.1. Bases del método de los elementos finitos

Incluso la solución de ecuaciones en derivadas parciales sencillas en geometrías complicadas raramente puede expresarse en términos de funciones elementales. De este modo, nos enfrentamos a dos problemas: en primer lugar, la necesidad de describir una geometría complicada y generar una malla en ella. Una vez que se ha realizado el mallado, discretizar la ecuación en derivadas parciales en esta malla y resolver la ecuación para la aproximación discreta de la solución. La interfaz gráfica de MATLAB `pdetool`, parte de la “PDE toolbox”, proporciona una herramienta gráfica de fácil manejo para la descripción de estas geometrías complicadas, generación de mallas, resolución de la ecuación discretizada y representación de resultados. A continuación se presenta un pequeño resumen de las bases en la que se fundamenta la misma, el método de los elementos finitos (FEM).

El punto de inicio es aproximar el dominio computacional Ω como la unión de objetos geométricos sencillos, en este caso triángulos. Seguidamente se impone que la solución debería ser simple en cada uno de los triángulos. Los polinomios son una buena elección: son fáciles de evaluar y tienen buenas propiedades de aproximación en dominios pequeños, pudiéndose conseguir conectar las soluciones sobre los lados de triángulos vecinos de modo que se consiga una solución aproximada continua. Para ello debemos elegir el orden de los polinomios que vamos a emplear. Si empleásemos constantes su evaluación en cada triángulo sería muy sencilla, pero no podríamos emparejar los valores de triángulos vecinos. El mínimo orden que podemos pedir a los polinomios para cumplir los requisitos es que sean funciones lineales (Fig. 10.1).

Para continuar con la descripción del método empleemos la ecuación elíptica básica (expresada sobre el dominio computacional Ω)

$$-\nabla \cdot (c\nabla u) + au = f$$

Si u_h es una aproximación lineal a trozos de la solución exacta u , entonces no está claro qué significa el término de la segunda derivada. Dentro de cada triángulo ∇u_h es una constante, y por lo tanto el término de la derivada de segundo orden se anula. En los bordes de los triángulos $c\nabla u_h$ es en general discontinua, y una derivada de orden superior carece de sentido.

En realidad lo que estamos buscando es la mejor aproximación de u en la clase de polinomios continuos a trozos. Por lo tanto lo que tenemos que hacer es probar la ecuación para u_h frente a todas las posibles funciones v que pertenezcan a esa clase. Probar significa formalmente el multiplicar el residuo por cualquier función e integrar, es decir, determinar u_h de modo que

$$\int_{\Omega} (-\nabla \cdot (c\nabla u_h) + au_h - f)v \, dx = 0$$

para todas las posibles v . Estas funciones v son llamadas normalmente *funciones de prueba*.

El integrar por partes (fórmula de Green) nos conduce a que u_h debe satisfacer

$$\int_{\Omega} ((c\nabla u_h)\nabla v + au_h v) \, dx - \int_{\partial\Omega} \vec{n} \cdot (c\nabla u_h)v \, ds = \int_{\Omega} f v \, dx \quad \forall v$$

donde $\partial\Omega$ es la frontera de Ω y \vec{n} es la normal apuntando hacia afuera de $\partial\Omega$. Véase que esta nueva formulación del problema ahora se encuentra bien definida incluso en el caso en el que u_h y v son funciones lineales a trozos.

Las condiciones de frontera se incluyen de la siguiente manera. Si u_h se conoce en algunos de los puntos de la frontera (condiciones de frontera Dirichlet), imponemos que las funciones de prueba sean $v = 0$ en esos puntos y que u_h posea el valor deseado en ese punto. Para el resto de puntos que posean condiciones de frontera Neumann, $(c\nabla u_h) \cdot \vec{n} + qu_h = g$, la formulación en FEM nos conduce a encontrar un u_h de modo que

$$\int_{\Omega} ((c\nabla u_h)\nabla v + au_h v) \, dx - \int_{\partial\Omega_1} qu_h v \, ds = \int_{\Omega} f v \, dx + \int_{\partial\Omega_1} g v \, ds \quad \forall v$$

donde $\partial\Omega_1$ es la parte de la frontera con condiciones de frontera Neumann. Las funciones de prueba v deben ser cero sobre $\partial\Omega - \partial\Omega_1$.

Cualquier función continua a trozos u_h puede ser representada como una combinación

$$u_h(x) = \sum_{i=1}^N U_i \phi_i(x)$$

donde los $\{\phi_i\}$ son una base de funciones lineales a trozos y U_i son coeficientes escalares. Escojamos los ϕ_i como si fuera una “tienda de campaña”, es decir, una función con altura 1 en el nodo i y altura 0 en el resto de los nodos. Para cada v que escojamos ahora, la formulación FEM conduce a una ecuación algebraica en las incógnitas U_i . Dado que queremos determinar N incógnitas, necesitamos N funciones de prueba v distintas. ¿Qué mejores candidatos que escoger $v = \phi_j$, $j = 1, 2, \dots, N$? De esta manera puede escribirse un sistema de ecuaciones $KU = F$ donde la matriz K y el lado derecho F contiene integrales en términos de las funciones de prueba ϕ_i , ϕ_j , y los coeficientes que definen el problema: c , a , f , q y g . El vector solución U contiene los coeficientes de la expansión de u_h , que coinciden con los valores de u_h en cada nodo x_i puesto que $u_h(x_i) = U_i$.

10.2. Usando la interfaz gráfica de la PDE toolbox

La “PDE toolbox” de MATLAB incluye una interfaz gráfica de usuario (GUI) completa, que cubre todos los aspectos del proceso de solución de una EDP. La interfaz se inicializa escribiendo

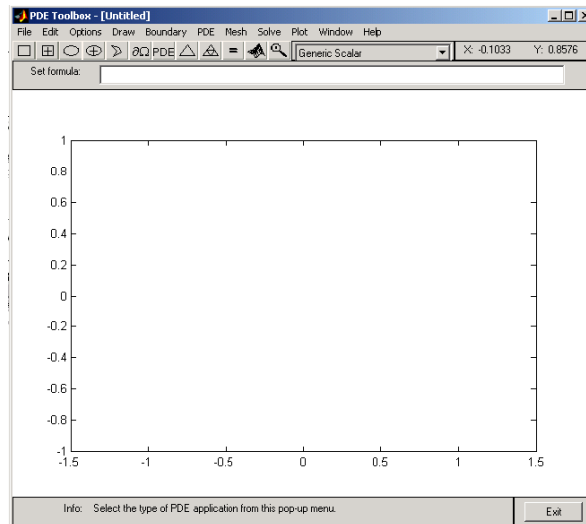


Figura 10.2: Interfaz gráfica `pdetool` de la toolbox de EDPs de MATLAB

`pdetool`

en la línea de comandos de MATLAB, siempre y cuando la toolbox mencionada se encuentre instalada en tu sistema. La Fig. 10.2 muestra el aspecto que presenta la `pdetool` GUI nada más ser inicializada. El uso de la misma es muy sencillo: iremos familiarizándonos con la misma a lo largo del capítulo.

10.2.1. Menús

Existen un total de 11 menús desplegables en la interfaz gráfica. Brevemente, la funcionalidad de cada uno de ellos es la siguiente:

- Menú **File**. Como es habitual, desde este menú pueden abrirse y salvarse ficheros `.m` que contienen los modelos en ecuaciones en derivadas parciales en los que se esté trabajando. También pueden imprimirse las gráficas activas en ese momento y salir de la interfaz.
- Menú **Edit**. Capacidades de edición habituales: copiar, cortar, borrar y pegar objetos, así como opciones de seleccionar todo.
- Menú **Options**. Contiene opciones como cambiar el rango y espaciado de los ejes, obligar a que en la fase de dibujo las formas se anclen a los puntos de la rejilla, zoom, etc.
- Menú **Draw**. Desde este menú se pueden seleccionar los objetos sólidos básicos como círculos o polígonos que se emplearán en la definición de la geometría, y a continuación dibujarlos en el área de trabajo mediante el uso del ratón. Se recomienda el uso de la barra de herramientas para este fin.
- Menú **Boundary**. Desde este menú se accede al cuadro de diálogo donde se definen las condiciones de frontera. Adicionalmente se pueden poner etiquetas

a los bordes y a los subdominios, borrar bordes entre subdominios y exportar la geometría descompuesta y las condiciones de fronteras al espacio de trabajo de MATLAB.

- Menú **PDE**. Este menú proporciona un cuadro de diálogo para especificar la EDP, y opciones para etiquetar subdominios y exportar los coeficientes de la ecuación al espacio de trabajo.
- Menú **Mesh**. Desde este menú se crea y se modifica la malla triangular. Puede inicializarse la malla, refinarla, reorganizarla, deshacer cambios previos en la malla, etiquetar los nodos y los triángulos, visualizar la calidad de la malla y exportar la misma al espacio de trabajo.
- Menú **Solve**. Para resolver la EDP. También abre un cuadro de diálogo donde pueden ajustarse los parámetros involucrados en la resolución, y exportar la solución al espacio de trabajo.
- Menú **Plot**. Desde este menú se puede dibujar correctamente la solución a la EDP. Un cuadro de diálogo permite seleccionar que propiedad va a visualizarse, en qué estilo y otros tipos de parámetros. Si se ha generado una animación en tiempo de la solución, también puede exportarse al espacio de trabajo.
- Menú **Window**. Básicamente para seleccionar cuál de las ventanas de figuras de MATLAB es la activa entre las abiertas en ese momento.
- Menú **Help**. Breve ayuda sobre ciertos comandos y funcionalidades.

10.2.2. Barra de herramientas

La barra de herramientas colocada debajo del menú principal (Fig. 10.3) en la parte superior de la interfaz gráfica contiene botones con iconos que proporcionan un acceso fácil y rápido a algunas de las funcionalidades más importantes de la `pdeTool`.










Figura 10.3: Barra de herramientas de la `pdeTool` de MATLAB

Los cinco botones más a la izquierda son los botones del modo de dibujo, representando cada uno, de izquierda a derecha:

	Dibuja un rectángulo/cuadrado comenzando en una esquina.
	Dibuja un rectángulo/cuadrado comenzando en el centro.
	Dibuja una elipse/círculo comenzando en el perímetro.
	Dibuja una elipse/círculo comenzando en el centro.
	Dibuja un polígono. Pincha y arrastra para crear los lados del polígono. El polígono puede cerrarse haciendo click con el botón derecho del ratón o pinchando sobre el vértice inicial.

Como en gran número de programas informáticos, sólo puede haber uno de estos botones de dibujo activado en cada momento. El hacer doble-click sobre uno de ellos fija como esa herramienta como activa, pudiendo seguir dibujando objetos del mismo tipo hasta que vuelva a pulsarse el botón. Usando el botón derecho del ratón, o bien Control+click, se restringen las herramientas a dibujar cuadrados o círculos en vez de rectángulos o elipses.

El segundo grupo de botones contiene las siguientes herramientas de análisis:

	Entra en el modo para especificar condiciones de frontera.
	Abre el cuadro de diálogo para especificar la EDP a resolver.
	Inicializa la malla triangular.
	Refina la malla triangular.
	Resuelve la EDP.
	Abre el cuadro de diálogo para representar los resultados.
	Zoom on/off.

10.3. Definición de geometrías complicadas

Las geometrías complicadas pueden generarse a partir de dibujar objetos sólidos básicos (rectángulos/cuadrados, elipses/círculos y polígonos) que se solapen, parcial o totalmente. La interfaz gráfica asigna automáticamente un nombre a cada objeto sólido que se cree: R1, R2, ... para los rectángulos; SQ1, SQ2, ... para cuadrados; E1, E2, ... para elipses; C1, C2, ... en el caso de los círculos; y P1, P2, ... para nombrar los polígonos. Obviamente, estos nombres pueden ser modificados por el usuario haciendo doble-click sobre los mismos, lo que abre el cuadro de diálogo de las propiedades del objeto (Fig. 10.4). Este cuadro de diálogo también nos posibilita el modificar otras características de la forma básica, como la posición de su centro, dimensiones, etc.

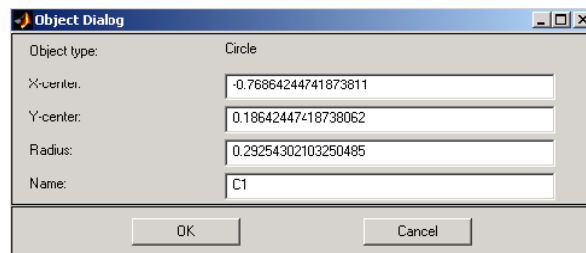


Figura 10.4: Cuadro de diálogo con las propiedades de un círculo

Una vez que los objetos básicos han sido dibujados, la geometría final se crea mediante la introducción, en la línea situada debajo de la barra de herramientas, de una fórmula que emplee operaciones del álgebra de conjuntos, +, * y -. De todos ellos, el operador de mayor precedencia es el operador diferencia, -, mientras que

los operadores unión e intersección, $+$ y $*$, poseen igual prioridad. Sin embargo, este orden de precedencia puede controlarse mediante el uso de paréntesis. El modelo geométrico final, Ω , es el conjunto de todos los puntos para los cuales la fórmula introducida puede evaluarse como verdadera. El proceso general puede entenderse más fácilmente a partir del siguiente ejemplo, la creación de una placa con esquinas redondeadas.

10.3.1. Creando esquinas redondeadas

Inicia la interfaz gráfica y activa la propiedad de la rejilla “snap-to-grid” localizada dentro del menú Options. Además, cambia el espaciado a $-1.5:0.1:1.5$ para el eje- x y $-1:0.1:1$ para el eje- y .

Selecciona el icono para crear rectángulos y usando el ratón dibuja uno de anchura 2 y altura 1, comenzando en el punto $(-1,0.5)$. Para crear las esquinas redondeadas añade círculos, uno en cada esquina. Los círculos deben tener radio 0.2 y centros a una distancia de 0.2 unidades de las fronteras izquierda/derecha y superior/inferior del rectángulo $((-0.8,-0.3), (-0.8,0.3), (0.8,-0.3)$ y $(0.8,0.3)$). Para dibujar círculos en vez de elipses, recuerda usar el botón derecho del ratón o mantener la tecla Ctrl pulsada mientras realizas el dibujo. Para finalizar, dibuja en cada una de las esquinas un pequeño cuadrado de lado 0.2. Los objetos dibujados deberían presentar el aspecto mostrado en la parte izquierda de la Fig. 10.5.

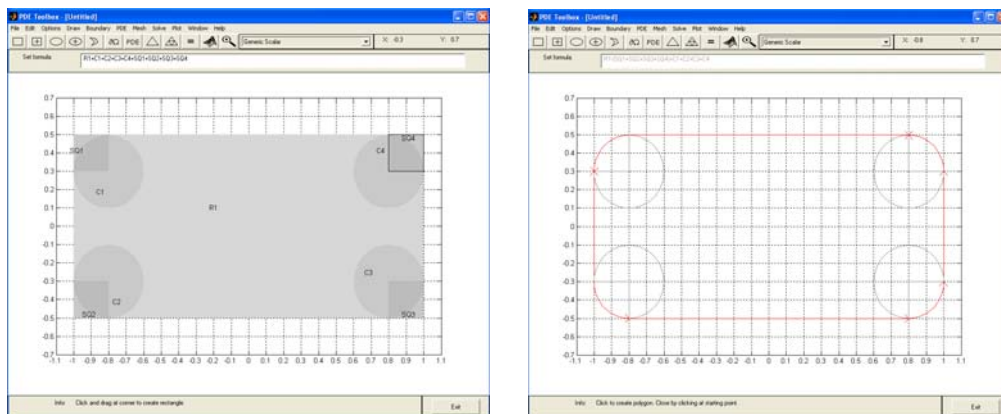



Figura 10.5: (Izquierda) Objetos básicos a partir de los cuales será definida la geometría definitiva. (Derecha) Modelo geométrico final, Ω .

Ahora debemos editar la fórmula que define la geometría. Para conseguir las esquinas redondeadas, restemos los cuadrados pequeños del rectángulo y sumemos a continuación los círculos. En forma de expresión de conjuntos como:







$$R1 - (SQ1 + SQ2 + SQ3 + SQ4) + C1 + C2 + C3 + C4$$

Presionando el botón  podemos entrar en el modo Boundary y ver las fronteras de la geometría final (Fig. 10.5, derecha). Puede observarse que aún existen dentro de la placa algunas de las fronteras provenientes de los subdominios originales. Si se supone que la placa es homogénea, entonces podemos borrarlos. Para ello, selecciona

la opción “Remove All Subdomain Borders” del menú Boundary. Ahora el modelo de la placa está completo.

10.4. Método sugerido de modelado

El flujo básico de acciones al emplear la interfaz gráfica de la toolbox de ecuaciones en derivadas parciales es el indicado de izquierda a derecha por los menús y los botones de la barra de herramientas, trabajando en este sentido a lo largo del proceso de modelado, definición y resolución del problema. La siguiente secuencia de acciones cubre todos los pasos de una sesión normal empleando la `pdetool`:

1. Usa la `pdetool` como herramienta de dibujo para realizar el dibujo de la geometría 2-D en la que se quiere resolver la EDP, haciendo uso de los objetos básicos y de la característica de “fijar a rejilla”. Combina los objetos sólidos mediante las fórmulas de álgebra de conjuntos para crear la geometría definitiva.
2. Salva la geometría a un fichero de modelo (un fichero `.m`), de manera que puedas seguir empleándola en futuras sesiones de trabajo. Si salvas el fichero más adelante a lo largo del proceso de resolución, el fichero del modelo también incluirá ciertos comandos para recrear las condiciones de frontera, los coeficientes de la EDP y la malla.
3. Pasa a especificar las condiciones de frontera presionando el botón . Si las fronteras no son las correctas, puedes volver a editar la geometría volviendo al modo de dibujo (Draw mode). Si durante la definición de la geometría han quedado bordes de subdominios no deseados puedes borrarlos mediante las opciones del menú Boundary (“Remove Subdomain Border” o “Remove All Subdomain Borders”). A continuación puedes fijar las condiciones de cada una de las fronteras haciendo doble-click sobre cada una de ellas.
4. Usa el botón  para especificar la EDP a resolver. En el caso en el que los coeficientes de la EDP dependan del material, estos son introducidos entrando en el modo PDE y haciendo doble-click en cada uno de los subdominios.
5. Inicializa la malla triangular mediante el botón . Normalmente, los parámetros por defecto del algoritmo de generación de la malla producen buenos resultados, aunque en caso necesario pueden modificarse desde la opción “Parameters” del menú Mesh.
6. Si es necesario, refina la malla mediante el botón . En cada refinamiento, el número de triángulos aumenta en un factor cuatro. Ten presente que cuando más fina sea la malla mayor será el tiempo requerido para calcular la solución. Otra opción es reordenar la triangulación de la malla para mejorar su calidad mediante la opción “Jiggle Mesh” del menú Mesh.
7. Resuelve la EDP presionando el botón .
8. Visualiza las propiedades de la solución en las que estés interesado mediante el botón . También puedes exportar la solución y/o la malla al espacio de trabajo principal de MATLAB para un análisis en mayor detalle.







10.5. Ejemplos de problemas elípticos: la ecuación de Poisson

Como ejemplo básico de un problema elíptico resolvamos la *ecuación de Poisson* en un disco unitario con condiciones de frontera Dirichlet homogéneas. La formulación del problema es la siguiente:

$$-\Delta u = 1 \text{ en } \Omega, \quad u = 0 \text{ en } \delta\Omega$$

donde Ω es el disco unitario y $\delta\Omega$ su frontera.

Tanto para éste como para el resto de ejemplos del capítulo seleccionaremos el modo **Generic Scalar** de la lista desplegable de modos disponible que se encuentra localizada a la derecha de la barra de herramientas. A continuación se listan los pasos a realizar con la `pdeTool` para resolver el problema descrito.

1. Dibuja un círculo de radio unidad centrado en el origen usando las herramientas de dibujo. Si dibujas una elipse en vez de un círculo, o el mismo no está perfectamente centrado, haz doble-click en el mismo y cambia sus propiedades en el cuadro de diálogo.
2. Imposición las condiciones de frontera. Para ello primero pulsa el botón  y a continuación haz doble-click sobre las fronteras de la región. Selecciona para todas ellas condiciones de frontera Dirichlet iguales a cero (aunque éstas son las condiciones de frontera por defecto).
3. Define la EDP presionando el botón . Esto abre un cuadro de diálogo donde pueden introducirse los coeficientes de la ecuación, c , a y f . En este caso tan sencillo todos son constantes: $c = 1$, $f = 1$ y $a = 0$. En el caso de que dependan de la posición, también pueden introducirse con la notación habitual de MATLAB en forma de productos de vectores (por ejemplo, $c = x.^2 + y.^2$).
4. Inicializa la malla haciendo uso del botón . Refínala con .
5. Resuelve la ecuación presionando . Emplea también  si quieres cambiar las propiedades del gráfico que la `pdeTool` devuelve por defecto, como por ejemplo el mapa de colores.

10.6. Ejemplos de problemas parabólicos: la ecuación del calor

Un problema parabólico típico es la *ecuación del calor*, que describe la difusión de calor en un cuerpo:

$$d\frac{\partial u}{\partial t} - \Delta u = 0$$

Como ejemplo estudiemos el comportamiento de calentar un bloque de metal que posee una cavidad rectangular en el medio. El lado izquierdo del bloque se calienta a 100°C. En la parte derecha del bloque, el calor fluye al aire circundante en una tasa

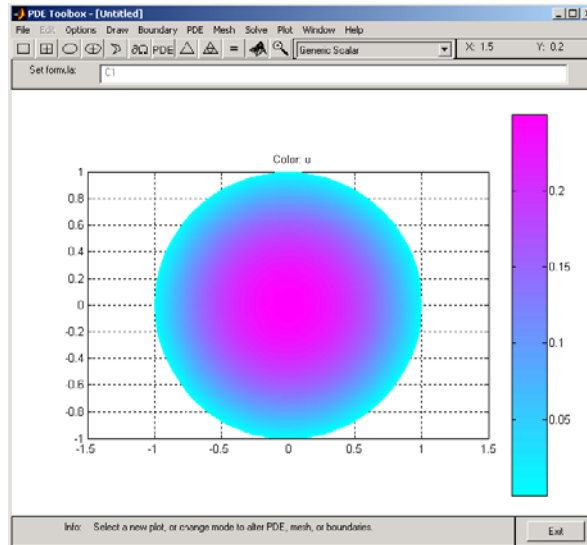



Figura 10.6: Solución a la ecuación de Poisson en el círculo unidad.

constante. Todas las otras fronteras del bloque se hayan aisladas térmicamente. Todo esto nos conduce al siguiente conjunto de condiciones de contorno:

- $u = 100$ en el lado izquierdo (condición Dirichlet)
- $\frac{\partial u}{\partial n} = -10$ en el lado derecho (condición Neumann)
- $\frac{\partial u}{\partial n} = 0$ en todas las restantes fronteras (condición Neumann)

Además, para la ecuación del calor necesitamos una condición inicial para t_0 . Para este caso supondremos que la temperatura del bloque es de 0°C en el momento en el que empezamos a aplicar el calor. Finalmente, para completar la formulación del problema, especificaremos que el tiempo inicial es 0 y que queremos estudiar la distribución de calor durante los cinco primeros segundos.

Resolvamos el problema utilizando el `pdeTool` de MATLAB. Los pasos a dar para resolver el problema son los siguientes:

1. En primer lugar, asegúrate de que se encuentra seleccionado el modo **Generic Scalar** en la lista desplegable que se encuentra a la derecha de la barra de herramientas.
2. Dibuja un rectángulo (R1) con las esquinas opuestas localizadas en $(-0.5, 0.8)$ y $(0.5, -0.8)$. Para crear la cavidad, dibuja otro rectángulo (R2) con esquinas opuestas situadas en $(-0.05, 0.4)$ y $(0.05, -0.4)$. Realiza la diferencia de ambas geometrías, introduciendo en la línea de debajo de la barra de herramientas la fórmula $R1 - R2$.
3. Impón las condiciones de frontera. Para ello primero pulsa el botón  y a continuación haz doble-click sobre las fronteras de la región. Para la frontera izquierda introduce la condición Dirichlet $u = 100$, para la derecha la condición Neumann $\partial u / \partial n = -10$, y para el resto condiciones Neumann homogéneas $\partial u / \partial n = 0$.

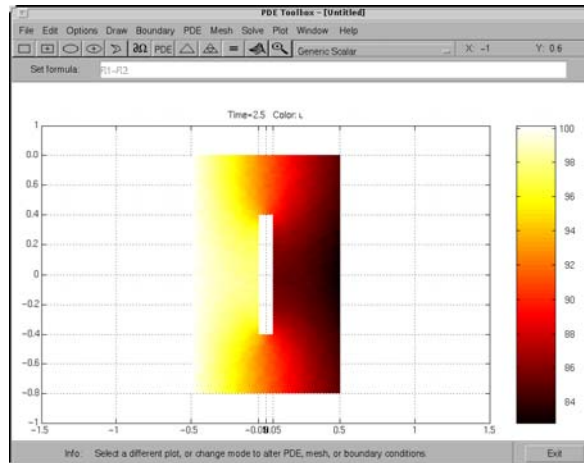





Figura 10.7: Solución a la ecuación del calor en la geometría estudiada.

4. Introduce los coeficientes que definen la EDP presionando el botón **PDE**. Para este caso, $d = 1$, $c = 1$, $a = 0$ y $f = 0$. Al ser un problema con evolución temporal, también debemos introducir la condición inicial $u_0 = 0$ y el rango de tiempo en el que queremos resolver el problema $[0 : 0.5 : 5]$. Estos parámetros son introducidos dentro del menú Solve seleccionando la opción “Parameters”.
5. Inicializa la malla haciendo uso del botón . Refínala con .
6. Resuelve la ecuación presionando **=**. Por defecto, la `pdeTool` muestra la distribución de calor para el instante final de la simulación. Presionando  puedes cambiar las propiedades del gráfico, por ejemplo, seleccionar un `colormap` más apropiado, como el `hot`. También es interesante poder ver cómo se mueve el calor a lo largo de la simulación. Para ello, activa la casilla “Animation” dentro del cuadro de diálogo de las propiedades de visualización. Por defecto, la animación se ejecuta cinco veces, aunque este número de veces también puede modificarse.

10.7. Ejemplos de problemas hiperbólicos: la ecuación de ondas

Como ejemplo de una EDP hiperbólica, resolvamos la *ecuación de ondas*

$$\frac{\partial^2 u}{\partial t^2} - \Delta u = 0$$

para las vibraciones transversales de una membrana en un cuadrado con esquinas en $(-1, -1)$, $(-1, 1)$, $(1, -1)$ y $(1, 1)$. La membrana está anclada ($u = 0$) en los lados izquierdo y derecho, y se encuentra libre ($\partial u / \partial n = 0$) en los lados superior e inferior. Adicionalmente necesitamos especificar los valores iniciales para $u(t_0)$ y $\partial u(t_0) / \partial t$. Para este ejemplo emplearemos como condiciones iniciales







$$u(0) = \arctan\left(\cos\left(\frac{\pi}{2}x\right)\right)$$

y

$$\frac{\partial u(0)}{\partial t} = 3 \operatorname{sen}(\pi x) e^{\sin(\frac{\pi}{2} y)}$$

que son valores iniciales que satisfacen las condiciones de frontera. La razón de escoger las funciones arcotangente y exponencial es tan sólo para introducir más modos en la solución y hacerla de esta forma más atractiva.

La secuencia de pasos a realizar para resolver el problema por medio de la `pdetool` de MATLAB son los que se listan a continuación:

1. Asegúrate de que se encuentra seleccionado el modo **Generic Scalar** en la lista de los posibles modos de solución.
2. Dibuja la geometría en la que estamos interesados, es decir, el cuadrado con esquinas en $(-1, -1)$, $(-1, 1)$, $(1, -1)$ y $(1, 1)$.
3. Impón las condiciones de frontera. Para ello primero pulsa el botón  y a continuación haz doble-click sobre las fronteras de la región. Para las frontera izquierda y derecha introduce la condición Dirichlet $u = 0$ y para las superior e inferior condiciones Neumann homogéneas $\partial u / \partial n = 0$.
4. Introduce los coeficientes que definen la EDP presionando el botón . Para este caso, $d = 1$, $c = 1$, $a = 0$ y $f = 0$. También debemos introducir la condiciones iniciales y el rango de tiempo en el que queremos resolver el problema $[0 : 0.5 : 5]$. Para ello, selecciona dentro del menú Solve la opción "Parameters". En el cuadro de diálogo emergente introduce `linspace(0, 5, 31)` como tiempos en los que resolver el problema, `atan(cos(pi/2*x))` como condición inicial para u y para $\partial u / \partial t$ introduce `3*sin(pi*x) .* exp(sin(pi/2*y))`
5. Inicializa la malla haciendo uso del botón . Refínala con .
6. Resuelve la ecuación presionando . Presionando  puedes cambiar las propiedades de visualización. Como sugerencia, la mejor forma de ver el movimiento de las ondas es en forma de una animación, aunque la misma puede ser relativamente costosa en términos de tiempo y memoria.

10.8. Ejemplos de problemas de autovalores: modos de vibración

El encontrar los autovalores y las correspondientes autofunciones de una membrana con forma de L es un problema de interés a todos los usuarios de MATLAB, dado que el logo del producto es precisamente la primera de estas autofunciones. Formalmente, lo que intentamos es calcular la solución al siguiente problema de autovalores

$$-\Delta u = \lambda u$$

en una geometría con forma de membrana en L, con condiciones de frontera Dirichlet homogéneas ($u = 0$ en todas las fronteras).

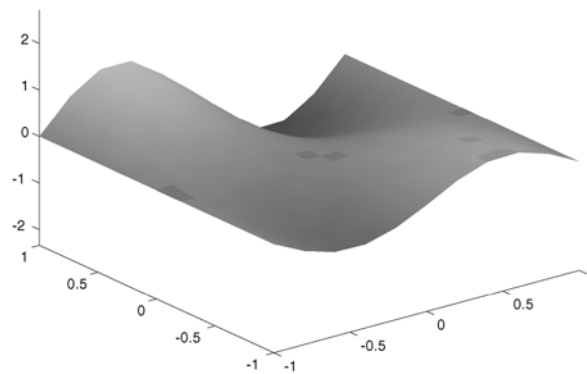








Figura 10.8: Solución a la ecuación de ondas.

1. Asegúrate de que se encuentra seleccionado el modo **Generic Scalar** en la lista de los posibles modos de solución.
2. Dibuja la membrana en forma de L como un polígono con esquinas en $(0, 0)$, $(-1, 0)$, $(-1, -1)$, $(1, -1)$, $(1, 1)$ y $(0, 1)$.
3. Impón las condiciones de frontera. Pulsa el botón  e introduce la condición Dirichlet homogénea, $u = 0$, para todas las fronteras de la membrana (condición por defecto).
4. Introduce los coeficientes que definen la EDP presionando el botón . Para este caso, $d = 1$, $c = 1$ y $a = 0$. También debemos introducir el rango de autovalores en el que estamos interesados. Para ello, selecciona dentro del menú Solve seleccionando la opción "Parameters". En el cuadro de diálogo emergente introduce $[0 \ 100]$.
5. Inicializa la malla haciendo uso del botón . Refínala con .
6. Resuelve la ecuación presionando . Presionando  puedes cambiar las propiedades de visualización, y seleccionar de un menú desplegable las distintas autofunciones que han sido calculadas como solución al problema.

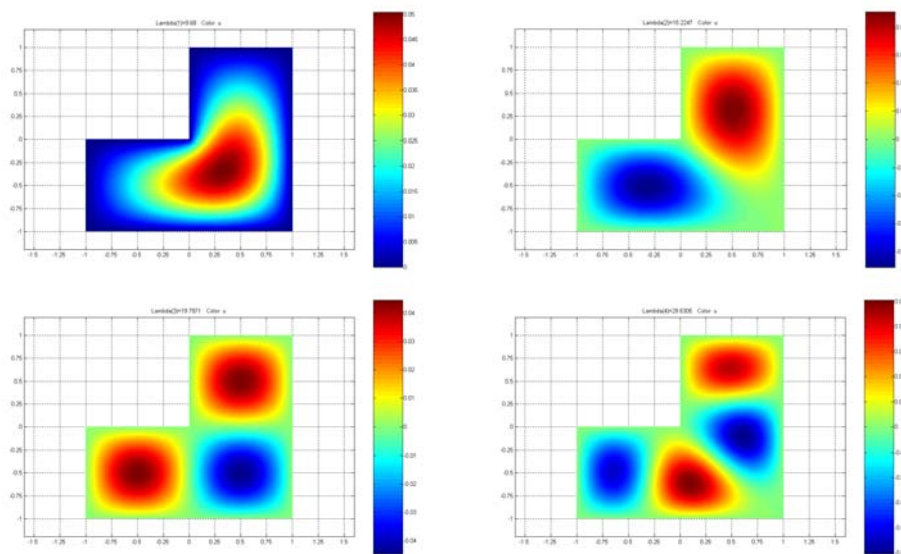


Figura 10.9: Cuatro primeras autofunciones al problema de autovalores en la membrana con forma de L, con $\lambda_1 = 9.68$, $\lambda_2 = 15.22$, $\lambda_3 = 19.79$ y $\lambda_4 = 29.63$.