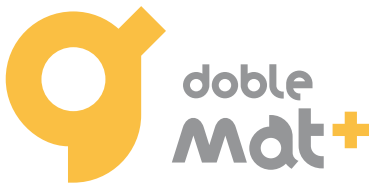


Tema 9: Método de los Elementos Finitos II

Resolución numérica de ecuaciones en derivadas parciales

Alicia Cordero, Juan R. Torregrosa



1 Conceptos básicos

2 Formulación variacional multidimensional

- Funciones base y minimización del funcional
- Triangulación e interpolación

3 Librería pdetool de Matlab

- Problema resuelto

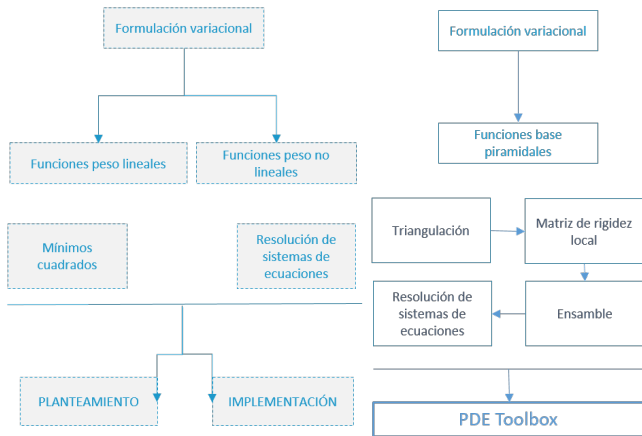
4 Referencias

MÉTODO DE ELEMENTOS FINITOS

INTRODUCCIÓN Y OBJETIVOS

PROBLEMAS DE FRONTERA (EDOS)

PROBLEMAS DE FRONTERA (EDPS)



Objetivos a desarrollar:

- Introducir el principio variacional sobre un problema de frontera multidimensional
- Obtener un procedimiento general, que incluya los siguientes aspectos:
 - Encontrar el funcional que minimiza el problema de frontera a resolver
 - Subdividir el dominio en elementos (mallado)
 - Utilizar funciones de interpolación lineal que actúen sobre cada elemento (funciones base)
 - Minimizar el funcional actuando sobre la relación de interpolación
 - Ensamblar las ecuaciones resultantes del paso anterior y ajustarlas a las condiciones de contorno, con lo que se obtendrá el sistema de ecuaciones a resolver.
- Introducir la herramienta PDE Toolbox de Matlab para resolver problemas modelizados mediante ecuaciones en derivadas parciales utilizando elementos finitos.

Consideramos la ecuación en derivadas parciales

$$\frac{\partial}{\partial x} \left(p(x, y) \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(q(x, y) \frac{\partial u}{\partial y} \right) + r(x, y)u(x, y) = f(x, y), \quad (1)$$

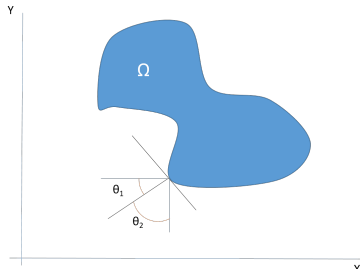
donde $(x, y) \in \Omega \subseteq \mathbb{R}^2$ tal que denotamos por S a la frontera del dominio Ω . Además, la frontera $S = S_1 \cup S_2$ de forma que las condiciones de contorno son

$$u(x, y) = g(x, y), \quad (x, y) \in S_1$$

y

$$p(x, y) \frac{\partial u}{\partial x} \cos(\theta_1) + q(x, y) \frac{\partial u}{\partial y} \cos(\theta_2) + g_1(x, y)u = g_2(x, y), \quad (x, y) \in S_2, \quad (2)$$

donde los ángulos θ_i , $i = 1, 2$ definen la normal exterior a la frontera de Ω .



Teorema

Sean $p, q \in \mathcal{C}^1(\Omega \cup S)$, r, f funciones continuas en $\Omega \cup S$ y g_1, g_2 continuas en S_2 . Supongamos además que

$$p(x, y) > 0, \quad q(x, y) > 0, \quad r(x, y) \leq 0 \text{ y } g_1(x, y) > 0.$$

Entonces una solución $v \in \mathcal{C}^2(\Omega \cup S)$ de la ecuación en derivadas parciales (1) es la única que minimiza el funcional

$$\begin{aligned} I(v) = & \int \int_{\Omega} \left(\frac{1}{2} \left[p(x, y) \left(\frac{\partial v}{\partial x} \right)^2 + q(x, y) \left(\frac{\partial v}{\partial y} \right)^2 - r(x, y) v^2 \right] \right. \\ & \left. + f(x, y) v \right) dx dy + \int_{S_2} \left(-g_2(x, y) v + \frac{1}{2} g_1(x, y) v^2 \right) dS \end{aligned}$$

y cumple la condición

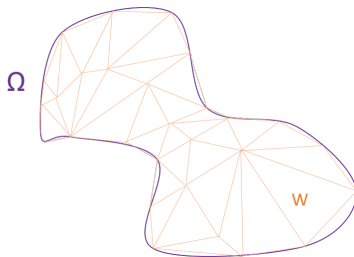
$$v(x, y) = g(x, y),$$

sobre S_1 .

- La EDP (1) es elíptica, por lo que la técnica que describiremos a lo largo de este tema se pueden aplicar de forma directa únicamente a este tipo de problemas.
- Cuando la EDP depende también del tiempo (en particular, cuando es parabólica o hiperbólica), podemos utilizar la técnica de elementos finitos
 - sobre todas las variables, espaciales y temporales,
 - o podemos hacerlo únicamente sobre las variables espaciales, discretizando la variable temporal mediante una técnica de diferencias finitas.

Funciones base y minimización del funcional

Consideraremos en adelante elementos triangulares, cuyos vértices reciben el nombre de nodos, con funciones base lineales, $\phi(x, y) = a + bx + cy$. Llamaremos W al conjunto de elementos triangulares.



El método de elementos finitos proporciona como aproximación al problema de frontera multidimensional una función construida como combinación lineal de las funciones base

$$\phi(x, y) = \sum_{i=1}^m c_i \phi_i(x, y),$$

siendo $\phi_i(x, y)$, $i = 1, 2, \dots, m$ funciones lineales a trozos linealmente independientes definidas sobre W y $c_i \in \mathbb{R}$, $i = 1, 2, \dots, m$.

¿Cómo se calculan los coeficientes c_i , $i = 1, 2, \dots, m$?

- Algunas se obtendrán de la minimización del funcional $I(\phi(x, y))$,

$$c_i, \quad i = 1, 2, \dots, n, \quad n < m.$$

- El resto, $(c_{n+1}, c_{n+2}, \dots, c_m)$ se fijarán al imponer la condición de contorno $\phi(x, y) = g(x, y), \quad (x, y) \in S_1$.

Por tanto,

$$\begin{aligned} I(\phi(x, y)) = & \int \int_{\Omega} \left(\frac{1}{2} \left[p(x, y) \left(\sum_{i=1}^m c_i \frac{\partial \phi_i(x, y)}{\partial x} \right)^2 + q(x, y) \left(\sum_{i=1}^m c_i \frac{\partial \phi_i(x, y)}{\partial y} \right)^2 \right. \right. \\ & \left. \left. - r(x, y) \left(\sum_{i=1}^m c_i \phi_i(x, y) \right)^2 \right] + f(x, y) \sum_{i=1}^m c_i \phi_i(x, y) \right) dx dy \\ & + \int_{S_2} \left(-g_2(x, y) \sum_{i=1}^m c_i \phi_i(x, y) + \frac{1}{2} g_1(x, y) \left(\sum_{i=1}^m c_i \phi_i(x, y) \right)^2 \right) dS \end{aligned}$$

y como condición necesaria para la minimización, exigiremos para cada $j \in \{1, 2, \dots, n\}$, que se satisfaga $\frac{\partial I(\phi(x, y))}{\partial c_j} = 0$.

$$\begin{aligned} \frac{\partial I(\phi(x, y))}{\partial c_j} = & \int \int_{\Omega} \left(p(x, y) \frac{\partial \phi_j(x, y)}{\partial x} \sum_{i=1}^m c_i \frac{\partial \phi_i(x, y)}{\partial x} \right. \\ & + q(x, y) \frac{\partial \phi_j(x, y)}{\partial y} \sum_{i=1}^m c_i \frac{\partial \phi_i(x, y)}{\partial y} \\ & \left. - r(x, y) \phi_j(x, y) \sum_{i=1}^m c_i \phi_i(x, y) + f(x, y) \phi_j(x, y) \right) dx dy \\ & + \int_{S_2} \left(-g_2(x, y) \phi_j(x, y) + g_1(x, y) \phi_j(x, y) \sum_{i=1}^m c_i \phi_i(x, y) \right) dS = 0, \end{aligned}$$

o equivalentemente, para cada $j \in \{1, 2, \dots, n\}$,

$$\begin{aligned} 0 = & \sum_{i=1}^m c_i \left[\int \int_{\Omega} \left(p(x, y) \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} + q(x, y) \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} - r(x, y) \phi_j(x, y) \phi_i(x, y) \right) dx dy \right. \\ & \left. + \int_{S_2} g_1(x, y) \phi_j(x, y) \phi_i(x, y) dS \right] \\ & + \int \int_{\Omega} f(x, y) \phi_j(x, y) dx dy - \int_{S_2} g_2(x, y) \phi_j(x, y) dS. \end{aligned}$$

Expresión matricial:

$$\boxed{Ac = b}, \quad c = (c_1, c_2, \dots, c_m)^T.$$

Expresión matricial:

$$\boxed{Ac = b}, \quad c = (c_1, c_2, \dots, c_m)^T.$$

siendo A la matriz de rigidez, de tamaño $n \times m$, de componentes

$$\begin{aligned} a_{i,j} = & \int \int_{\Omega} \left(p(x, y) \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} + q(x, y) \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} - r(x, y) \phi_j(x, y) \phi_i(x, y) \right) dx dy \\ & + \int_{S_2} g_1(x, y) \phi_j(x, y) \phi_i(x, y) dS, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \end{aligned}$$

y el vector de términos independientes $b = (b_1, b_2, \dots, b_n)^T$

$$b_i = - \int \int_{\Omega} f(x, y) \phi_i(x, y) dx dy + \int_{S_2} g_2(x, y) \phi_i(x, y) dS - \sum_{k=n+1}^m a_{i,k} c_k,$$

para $i = 1, 2, \dots, n$.

Dependiendo del tipo de funciones peso que se utilicen, la matriz A puede ser definida positiva y tener una estructura de bandas, lo que facilita considerablemente la resolución del sistema.

Triangulación e interpolación

Asumiremos que la región del plano $\Omega = W = \bigcup_{i=1}^r T_i$, donde cada T_i , $i = 1, 2, \dots, r$ es una región triangular de nodos $p_j^{(i)} = (x_j^{(i)}, y_j^{(i)})$, para $j = 1, 2, 3$. Además, la frontera S de W es un conjunto de líneas rectas enlazadas.

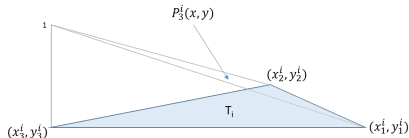
- Estos triángulos no deben estar superpuestos y
- deben definirse de forma que no haya ningún vértice de algún triángulo ubicado sobre el lado de otro.

Asociamos a cada nodo del triángulo T_i una función piramidal

$$P_j^{(i)}(x, y) = \alpha_j^{(i)} + \beta_j^{(i)}x + \gamma_j^{(i)}y,$$

tal que

$$P_j^{(i)}(x_k, y_k) = \begin{cases} 1, & j = k, \\ 0, & j \neq k. \end{cases},$$



Conocidos los nodos $p_j^{(i)}$ de cada triángulo T_i , la resolución de un sencillo sistema lineal $Md = e^{(j)}$ nos proporciona los coeficientes $d = (\alpha_j^{(i)}, \beta_j^{(i)}, \gamma_j^{(i)})^T$, $j = 1, 2, 3$. Notemos que la matriz

$$M = \begin{pmatrix} 1 & x_1^{(i)} & y_1^{(i)} \\ 1 & x_2^{(i)} & y_2^{(i)} \\ 1 & x_3^{(i)} & y_3^{(i)} \end{pmatrix}$$

es no singular por la construcción de la triangulación y el vector $e^{(j)}$ es el vector unitario cuya única componente no nula es la j -ésima.

- La solución del sistema $Md = e^{(j)}$ es única.
- La función piramidal definida sobre cada vértice de un triángulo es única.

Sobre cada triángulo T_i , las funciones peso serán las tres funciones piramidales. Así, el proceso de minimización del funcional se lleva a cabo sobre cada triángulo de W .

Con esta construcción de las funciones piramidales, la matriz de coeficientes A será simétrica, definida positiva y no singular, por lo que el sistema lineal $Ac = b$ tendrá solución única.

Ejemplo

Consideremos el dominio cuadrangular de la Figura, separada en cuatro regiones triangulares T_1 a T_4 . Los nodos o vértices de esta triangularización son $N_1 = (1, 0)$, $N_2 = (0, 1)$, $N_3 = (-1, 0)$, $N_4 = (0, -1)$ y $N_5 = (0, 0)$.

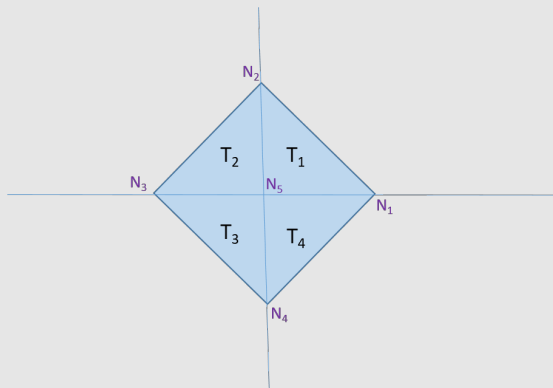
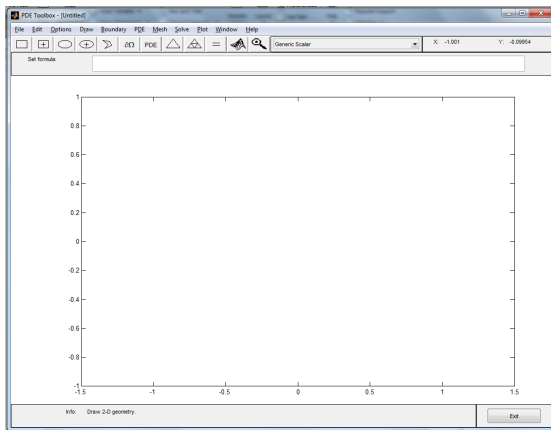


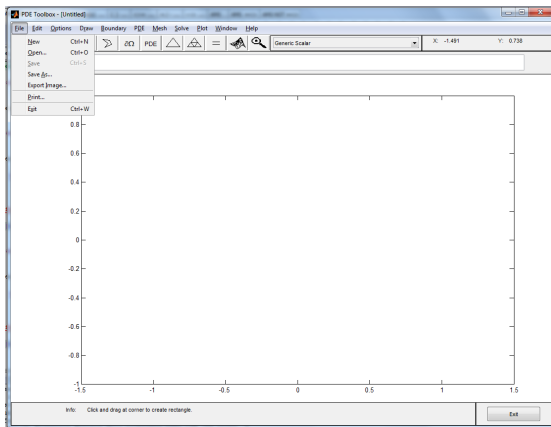
Figure: Dominio cuadrangular y triangularización

Ejecutando en la ventana de comandos de Matlab:

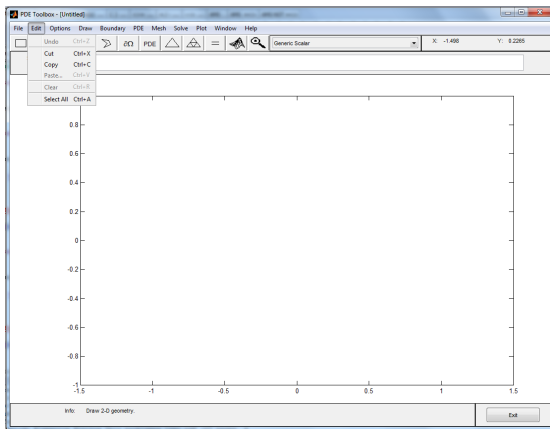
```
>> pdetool
```

aparece la interfaz gráfica:

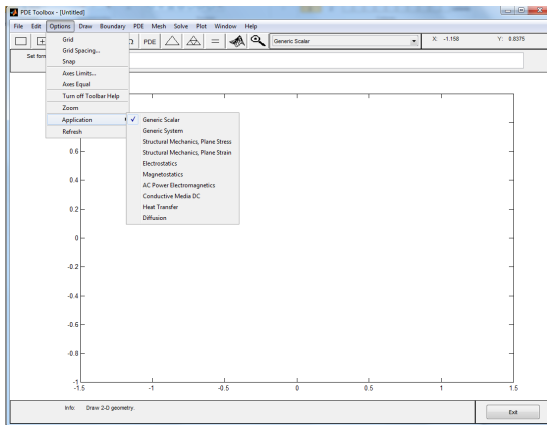




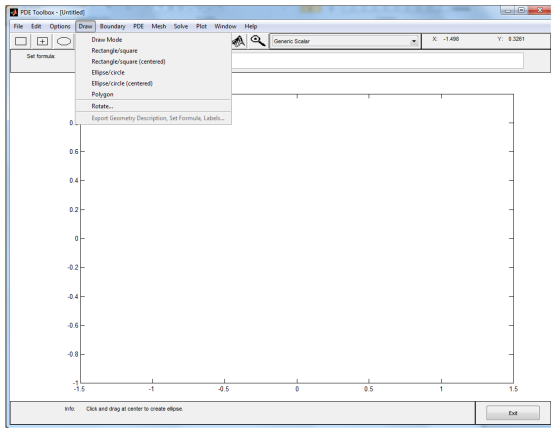
Menú **File**: Gestión de ficheros, impresión de gráficas y salir de la aplicación.



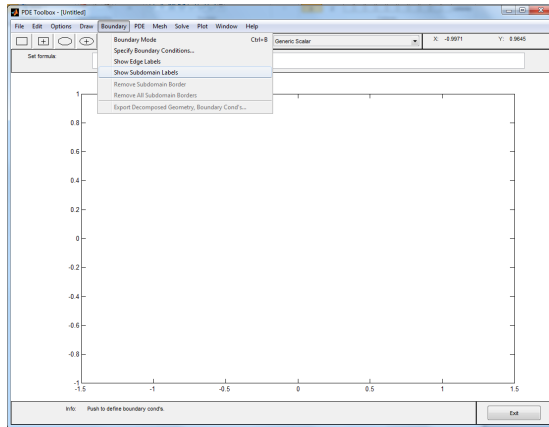
Menú **Edit**: Seleccionar, copiar, cortar, borrar y pegar.



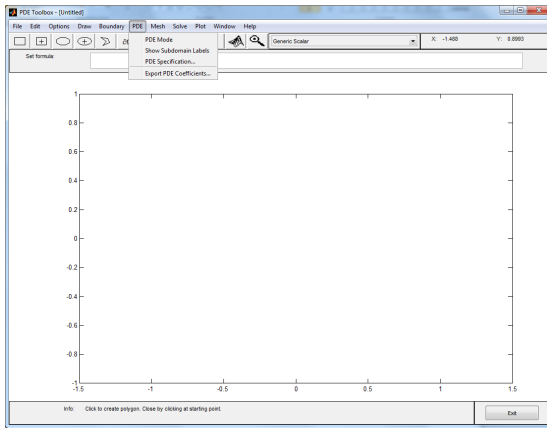
Menú **Options**: Activar/desactivar mallado, definición de los ejes, zoom,...



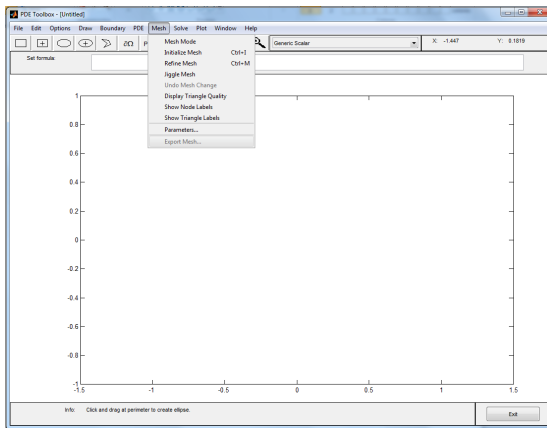
Menú **Draw**: Dibujo de figuras geométricas sobre el mallado (misma función que los botones de la barra de herramientas).



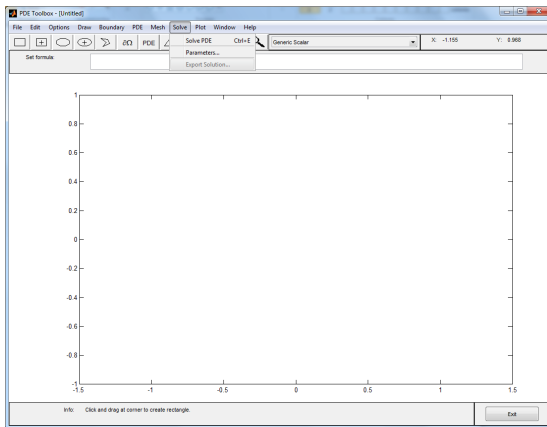
Menú **Boundary**: Permite acceder al cuadro de diálogo donde se definen las condiciones de frontera, poner etiquetas, borrar subdominios,...



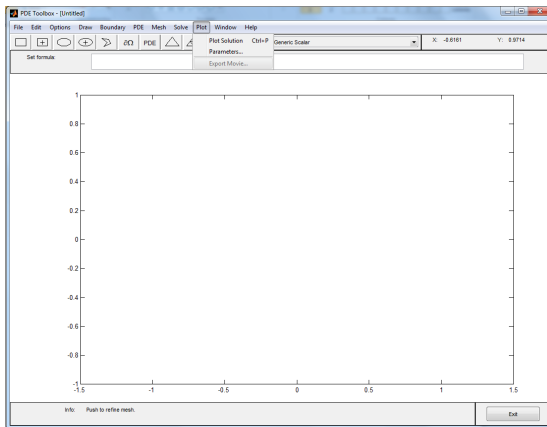
Menú **PDE**: Permite acceder al cuadro de diálogo donde se define la ecuación en derivadas parciales.















Menú **Mesh**: Crea y modifica la malla triangular (etiquetas, refinado,...)



Menú **Solve**: Resuelve la EDP y exporta resultados al espacio de trabajo de Matlab.



Menú **Plot**: Representación gráfica de la solución.

	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.
<hr/>	
	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.

En una región bidimensional P_1 la temperatura $u(x, y)$ satisface la ecuación de Laplace

$$\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) = 0 \text{ en } P_1$$

bajo las condiciones de contorno:

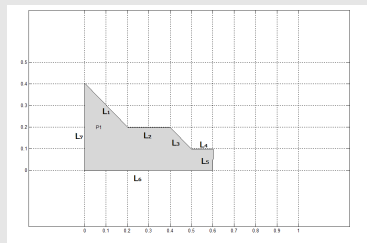
$$u(x, y) = 4, \quad (x, y) \in L_6, (x, y) \in L_7$$

$$\frac{\partial u}{\partial n}(x, y) = x, \quad (x, y) \in L_2, (x, y) \in L_4$$

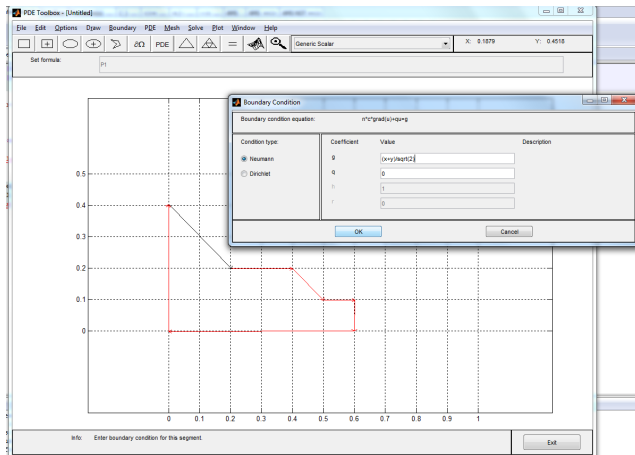
$$\frac{\partial u}{\partial n}(x, y) = y, \quad (x, y) \in L_5$$

$$\frac{\partial u}{\partial n}(x, y) = \frac{x + y}{\sqrt{2}}, \quad (x, y) \in L_1, (x, y) \in L_3$$

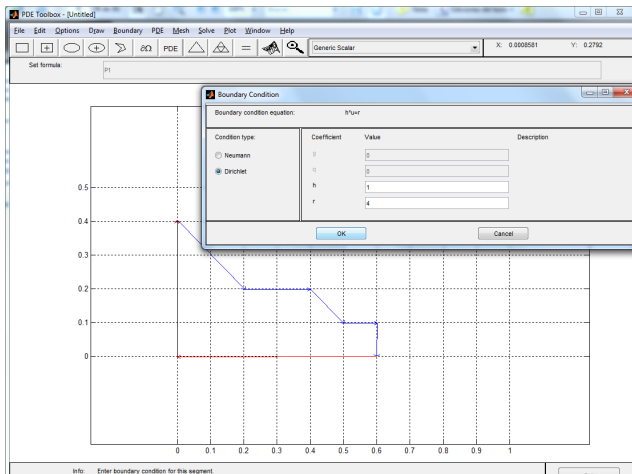
donde $\frac{\partial}{\partial n}$ denota la derivada direccional en la dirección normal a la frontera de la región en el punto (x, y) . Calcula el error exacto en los nodos sabiendo que la solución exacta es $u(x, y) = xy + 4$.



- Dibujamos el dominio poligonal ayudándonos de un mallado de paso 0.1.
- Pulsamos el botón $\partial\Omega$ para especificar las condiciones de contorno del problema. Al pinchar (doble click) sobre cada lado de la frontera se abre un cuadro de diálogo donde especificamos el tipo de condición (en nuestro caso Neumann, en azul; si es de tipo Dirichlet, en rojo) y el valor.

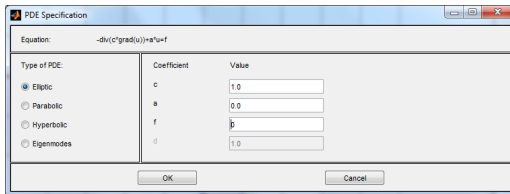


- Dibujamos el dominio poligonal ayudándonos de un mallado de paso 0.1.
- Pulsamos el botón $\partial\Omega$ para especificar las condiciones de contorno del problema. Al pinchar (doble click) sobre cada lado de la frontera se abre un cuadro de diálogo donde especificamos el tipo de condición (en nuestro caso Neumann, en azul; si es de tipo Dirichlet, en rojo) y el valor.

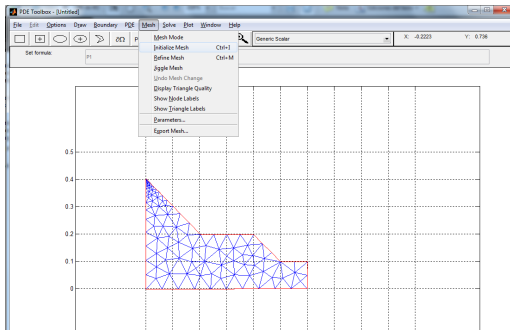


Problema resuelto

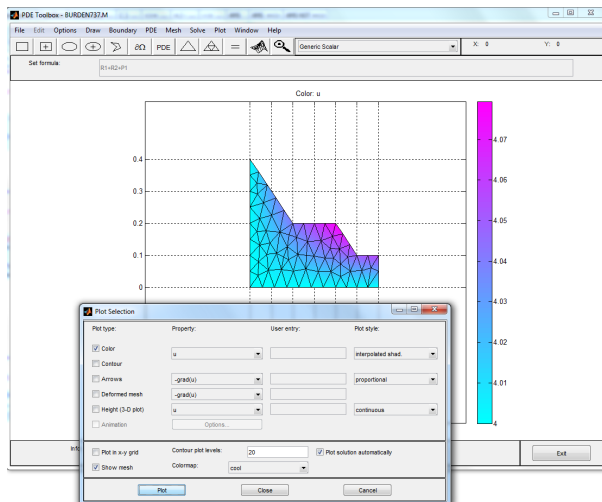
- Pulsamos el botón PDE para introducir el tipo de ecuación en derivadas parciales y su expresión.



- Inicializamos el mallado por defecto (si es necesario, podemos refinarlo).

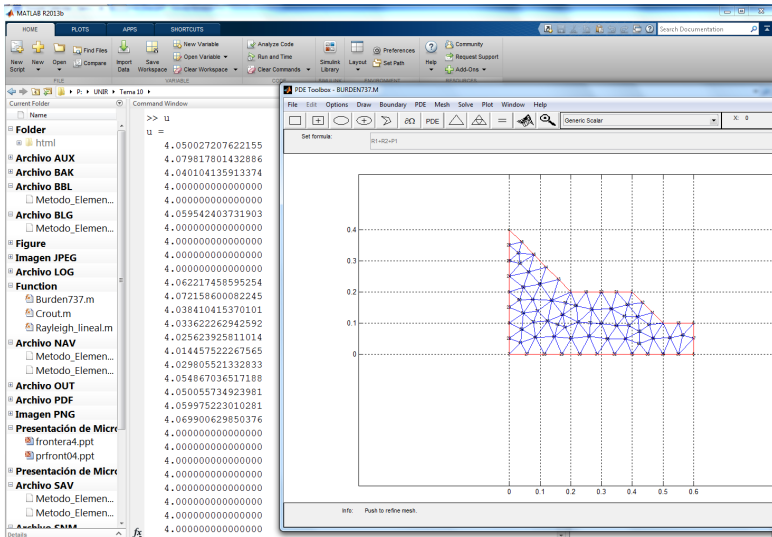


- Resolvemos la EDP con el botón $=$ y las propiedades de visualización con el botón de la gráfica.



Problema resuelto

- Exportamos la solución a Matlab con el menú **Solve** y la opción **Export solution**.



- Si conocemos la solución exacta y queremos calcular el error, exportamos el mallado a Matlab con el menú **Mesh** y la opción **Export Mesh**. En la ventana de comandos podemos calcular el error (con 76 nodos):

```
>> size(u)
ans = 76
>> exact=p(1,:).*p(2,:)+4;
>> error=norm(u-exact')
```

```
error =
0.002031045215266
```

- Para mejorar los resultados refinamos el mallado y exportamos tanto la solución como el mallado. Con esta información, recalculamos el error:

```
>> size(u)
ans = 266 1
>> exact=p(1,:).*p(2,:)+4;
>> error=norm(u-exact')
```

```
error =
8.296409698762747e-04
```




S LARSSON, V THOMÉE, *Partial differential equations with numerical methods*, Springer, Berlin, 2016.



T. MYINT-U, L. DEBNATH, *Partial differential equations for Scientist and engineers*, Ed. North-Holland, New York, 1987.



R. BURDEN, J. FAIRES, *Análisis Numérico*, Ed. Thompson, 2002.



S.C. CHAPRA, R.P. CANALE, *Métodos numméricos para ingenieros*, Ed. McGraw-Hill, México D.F., 2006.



L. LAPIDUS, G. PINDER, *Numerical solution of partial differential equations in science and engineering*, Ed. Wiley Interscience Publication, New York, 1999.



A. CORDERO, J.L. HUESO, E. MARTÍNEZ, J.R. TORREGROSA, *Problemas resueltos de métodos numéricos*, Ed. Thompson, 2006.



J. MATHEWS, K. FINK, *Métodos Numéricos con Matlab*, Ed. Prentice-Hall, 1999.