

## Guía complementaria Tarea 5:

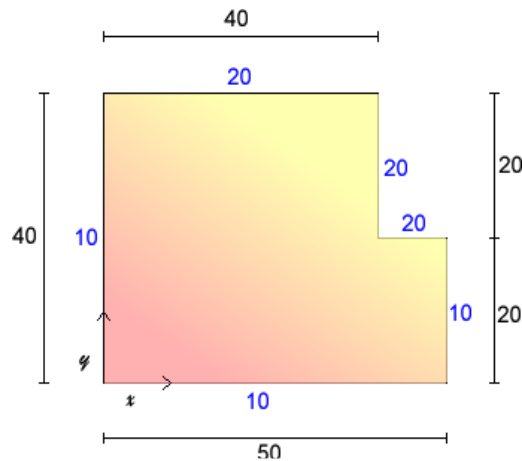
### Ejemplo de Diferencias Finitas en un dominio no rectangular.

#### Visualización en Matlab y matrices sparse.

A continuación se presenta un ejemplo de aplicación del método de diferencias finitas. En particular, se resuelve la ecuación de Laplace en un dominio no rectangular con condiciones de borde tipo Dirichlet.

$$\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2} = 0 \quad (1)$$

Dominio y condiciones de borde:

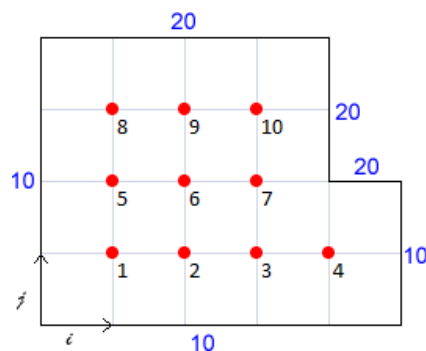


Aplicando la aproximación de diferencias finitas y considerando un mismo paso de discretización para las variables x e y, se obtiene la siguiente expresión:

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = 0 \quad (2)$$

Donde se utiliza el subíndice i asociado a la variable x y j asociado a la variable y.

El dominio se discretiza en una malla regular con 10 puntos interiores, los que se numeran con el orden mostrado en la figura:



Cada punto interior del dominio constituye una incógnita. Escribiendo la ecuación (2) para cada incógnita:

Punto	i	j	Ecuación
1	1	1	$u_{2,1} + 10 + u_{1,2} + 10 - 4u_{1,1} = 0$
2	2	1	$u_{3,1} + u_{1,1} + u_{2,2} + 10 - 4u_{2,1} = 0$
3	3	1	$u_{4,1} + u_{2,1} + u_{3,2} + 10 - 4u_{3,1} = 0$
4	4	1	$10 + u_{3,1} + 20 + 10 - 4u_{4,1} = 0$
5	1	2	$u_{2,2} + 10 + u_{1,3} + u_{1,1} - 4u_{1,2} = 0$
6	2	2	$u_{3,2} + u_{1,2} + u_{2,3} + u_{2,1} - 4u_{2,2} = 0$
7	3	2	$20 + u_{2,2} + u_{3,3} + u_{3,1} - 4u_{3,2} = 0$
8	1	3	$u_{2,3} + 10 + 20 + u_{1,2} - 4u_{1,3} = 0$
9	2	3	$u_{3,3} + u_{1,3} + 20 + u_{2,2} - 4u_{2,3} = 0$
10	3	3	$20 + u_{2,3} + 20 + u_{3,2} - 4u_{3,3} = 0$

Escribiendo la incógnita  $u_{i,j}$  como  $U_k$  con el orden introducido en la figura, el sistema de ecuaciones se puede escribir matricialmente como:

$$\begin{bmatrix} -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \\ U_7 \\ U_8 \\ U_9 \\ U_{10} \end{bmatrix} = \begin{bmatrix} -20 \\ -10 \\ -10 \\ -40 \\ -10 \\ 0 \\ -20 \\ -30 \\ -20 \\ -40 \end{bmatrix}$$

$A \qquad \qquad \qquad U \qquad = \qquad b$

Resolviendo este sistema, y volviendo a las variables originales, se obtiene:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \\ U_7 \\ U_8 \\ U_9 \\ U_{10} \end{bmatrix} = \begin{bmatrix} 11,2175 \\ 12,1937 \\ 12,9795 \\ 13,2449 \\ 12,6762 \\ 14,5778 \\ 16,4794 \\ 14,9095 \\ 16,9619 \\ 18,3603 \end{bmatrix} = \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{4,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ u_{1,3} \\ u_{2,3} \\ u_{3,3} \end{bmatrix}$$

Poniendo estas soluciones sobre el dominio, y agregando las condiciones de borde, se obtiene la siguiente matriz solución:

$$u = \begin{bmatrix} 10 & 20 & 20 & 20 & 20 \\ 10 & 14,9095 & 16,9619 & 18,3603 & 20 \\ 10 & 12,6762 & 14,5778 & 16,4794 & 20 & 20 \\ 10 & 11,2175 & 12,1937 & 12,9795 & 13,2449 & 10 \\ 10 & 10 & 10 & 10 & 10 & 10 \end{bmatrix}$$

## Visualización en Matlab

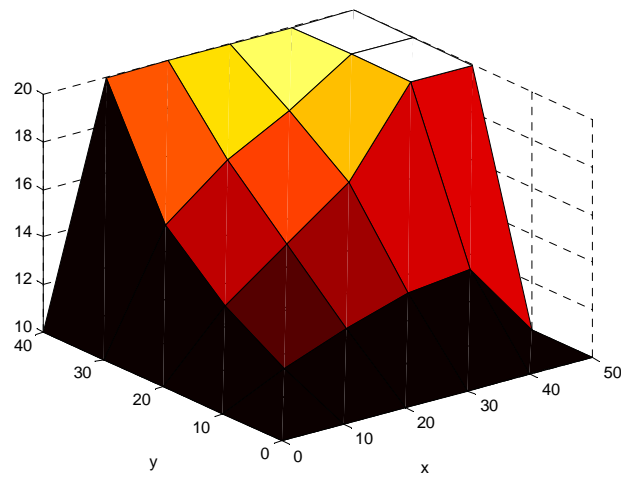
Es posible visualizar la superficie de solución fácilmente utilizando Matlab.

```
u2=u(5:-1:1,:);  
ex=linspace(0,50,6);  
ey=linspace(0,40,5);
```

La primera línea invierte la matriz verticalmente, ya que Matlab considera el eje x hacia la derecha y el eje y hacia abajo, comenzando desde la posición (1,1). Las 2 líneas siguientes se encargan de generar los ejes.

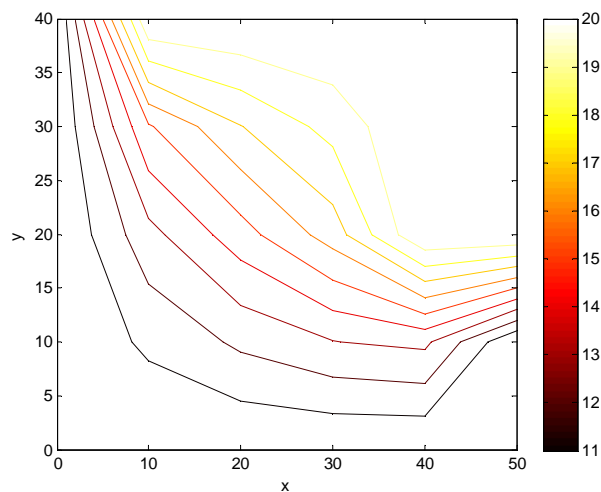
La función surf permite visualizar la superficie:

```
surf(ex,ey,u2); colormap hot; xlabel('x'); ylabel('y');
```



Por otro lado, contour, permite visualizar las curvas de nivel:

```
contour(ex,ey,u2); colormap hot; xlabel('x'); ylabel('y');
```



## Limitación de memoria y matrices sparse

Considere un dominio rectangular con 100 puntos tanto horizontal como verticalmente, se tiene entonces, un total de  $100 * 100 = 10^4$  incógnitas. De esta forma, para la matriz de coeficientes, se necesita almacenar  $10^4 * 10^4 = 10^8$  números. Matlab trabaja con precisión double, luego, se necesita un total de  $10^8 * 8 \text{ bytes} \approx 800 \text{ Mb}$ . Luego, para grandes cantidades de incógnitas, el problema se hace inmanejable por limitaciones de memoria.

Sin embargo, se observa que la matriz de coeficientes resultante posee una gran cantidad de ceros. Para estos casos, Matlab facilita las matrices sparse. En estas matrices se guardan solo los valores no nulos y su posición en la matriz, optimizándose el uso de memoria.

### Ejemplo:

```
sA=sparse(100,100);  
sA(1,10)=30;  
sA(40,40)=50;  
A=full(sA);
```

La primera línea genera una matriz sparse nula. Las 2 siguientes líneas ubican los valores 30 y 50 en las posiciones especificadas. Para trabajar con estas matrices se utilizan las mismas expresiones que para las matrices normales, esto es:  $A^2$ ,  $A*B$ ,  $A+B$ , etc... Incluso se puede operar entre matrices sparse y normales. Finalmente, con la función full, se transforma una matriz sparse a una normal.

*Documento elaborado por Daniel Calderon S.  
Revisado por M.C. Rivara*