Ecuaciones diferenciales parciales de segundo orden

Manuel Carlevaro

Departamento de Ingeniería Mecánica

Grupo de Materiales Granulares - UTN FRLP

manuel.carlevaro@gmail.com

Ecuación diferencial parcial (EDP) lineal de segundo orden:

$$A\frac{\partial^2 \phi}{\partial x^2} + B\frac{\partial^2 \phi}{\partial x \partial y} + C\frac{\partial^2 \phi}{\partial y^2} + D\frac{\partial \phi}{\partial x} + E\frac{\partial \phi}{\partial y} + F\phi = S$$

donde A, B, C, D, E, F y S son funciones de x y y en $D \in \mathbf{R}^2$.

Ecuación diferencial parcial (EDP) lineal de segundo orden:

$$A\frac{\partial^2 \phi}{\partial x^2} + B\frac{\partial^2 \phi}{\partial x \partial y} + C\frac{\partial^2 \phi}{\partial y^2} + D\frac{\partial \phi}{\partial x} + E\frac{\partial \phi}{\partial y} + F\phi = S$$

donde A, B, C, D, E, F y S son funciones de x y y en $D \in \mathbf{R}^2$.

Tipos:

‣ Parabólica: $B^2 - 4AC = 0$

‣ Elíptica: $B^2 - 4AC < 0$

‣ Hiperbólica: $B^2 - 4AC > 0$

para todo $(x, y) \in D$.

Casos:

‣ Conducción de calor en sólidos, flujo de fluidos

‣ Ejemplos:

  › Conducción de calor:
  $$\rho c\frac{\partial T}{\partial t} = k\frac{\partial^2 T(x,t)}{\partial x^2} + Q(x)$$

  › Transporte convectivo:
  $$\frac{\partial \phi}{\partial t} = -\frac{\partial}{\partial x}u(x)\phi + D\frac{\partial^2 \phi}{\partial x^2}$$

Ecuación diferencial parcial (EDP) lineal de segundo orden:

$$A\frac{\partial^2 \phi}{\partial x^2} + B\frac{\partial^2 \phi}{\partial x \partial y} + C\frac{\partial^2 \phi}{\partial y^2} + D\frac{\partial \phi}{\partial x} + E\frac{\partial \phi}{\partial y} + F\phi = S$$

donde A, B, C, D, E, F y S son funciones de x y y en $D \in \mathbf{R}^2$.

Tipos:

▸ Parabólica: $B^2 - 4AC = 0$

▸ Elíptica: $B^2 - 4AC < 0$

▸ Hiperbólica: $B^2 - 4AC > 0$

para todo $(x, y) \in D$.

Casos:

▸ Problemas estacionarios de 2 y 3 dimensiones

▸ Conducción de calor en sólidos, vibración de membranas

▸ Ejemplos:

  › Ecuación de Poisson:

  $$-\nabla^2 \phi(x, y) = S(x, y)$$

  › Ecuación de Laplace:

  $$-\nabla^2 \phi(x, y) = 0$$

Ecuación diferencial parcial (EDP) lineal de segundo orden:

$$A\frac{\partial^2 \phi}{\partial x^2} + B\frac{\partial^2 \phi}{\partial x \partial y} + C\frac{\partial^2 \phi}{\partial y^2} + D\frac{\partial \phi}{\partial x} + E\frac{\partial \phi}{\partial y} + F\phi = S$$

donde A, B, C, D, E, F y S son funciones de x y y en $D \in \mathbf{R}^2$.

Tipos:
- Parabólica: $B^2 - 4AC = 0$
- Elíptica: $B^2 - 4AC < 0$
- Hiperbólica: $B^2 - 4AC > 0$

para todo $(x, y) \in D$.

Casos:
- Problemas oscilatorios, propagación de ondas, fluidos
- Ejemplos:
  - Ecuación de onda:
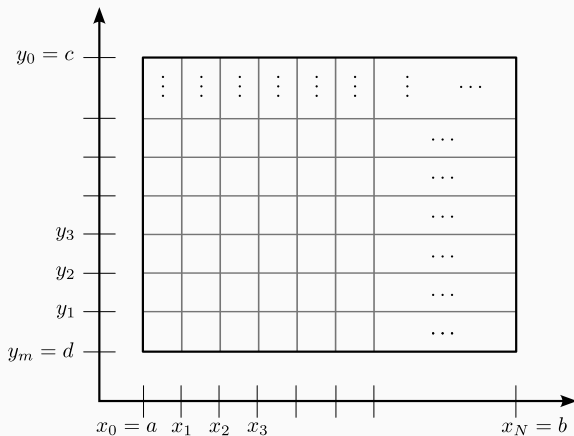 $$\frac{\partial^2 u(x, y, z, t)}{\partial t^2} = c^2 \nabla^2 u(x, y, z, t)$$
  - Navier–Stokes (incompresible):
  $$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu \nabla^2 \boldsymbol{u} = -\nabla\left(\frac{p}{p_0}\right) + \boldsymbol{g}$$

Ecuación de Poisson (elíptica):

$$\nabla^2 u(x,y) = \frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = f(x,y)$$

en $R = \{(x,y)|a < x < b, c < y < d\}$, con $u(x,y) = g(x,y)$ para $(x,y) \in S$, siendo S la frontera de R.

Malla:

- División $[a,b]$ y $[c,d]$ en n y m partes iguales
- $h = (b-a)/n$, $k = (d-c)/m$
- $x_i = a + ih$, $i = 0, 1, \ldots, n$
- $y_j = c + jk$, $j = 0, 1, \ldots, m$

Aproximación en diferencias finitas (serie de Taylor):

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

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

Con $u(x_i, y_j) \mapsto u_{i,j}$:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} = f_{i,j} + \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, y_j) + \frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i, \eta_j)$$
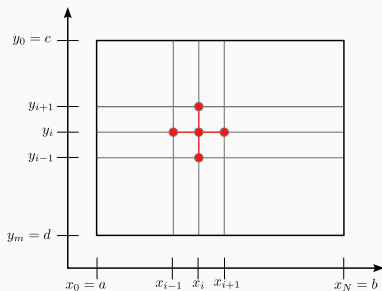
para $i = 1, 2, \ldots, n-1$, $j = 1, 2, \ldots, m-1$ y condiciones de contorno:

$$u_{0,j} = g_{0,j} \quad \text{y} \quad u_{n,j} = g_{n,j}, \quad j = 0, 1, \ldots m;$$
$$u_{i,0} = g_{i,0} \quad \text{y} \quad u_{i,m} = g_{i,m}, \quad i = 0, 1, \ldots n$$

Resulta:

$$2\left[\left(\frac{h}{k}\right)^2 + 1\right] u_{i,j} - (u_{i+1,j} + u_{i-1,j}) - \left(\frac{h}{k}\right)^2 (u_{i,j+1} + u_{i,j-1}) = -h^2 f_{i,j}, \ i \in [1, n-1], \ j \in [1, m-1]$$

Con $u(x_i, y_j) \mapsto u_{i,j}$:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} = f_{i,j} + \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, y_j) + \frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i, \eta_j)$$
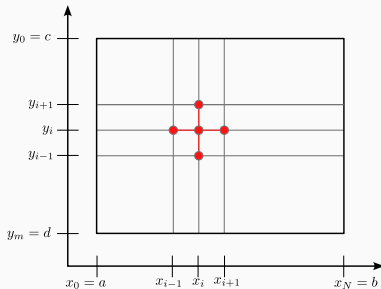
para $i = 1, 2, \ldots, n-1$, $j = 1, 2, \ldots, m-1$ y condiciones de contorno:

$$u_{0,j} = g_{0,j} \quad \text{y} \quad u_{n,j} = g_{n,j}, \quad j = 0, 1, \ldots m;$$
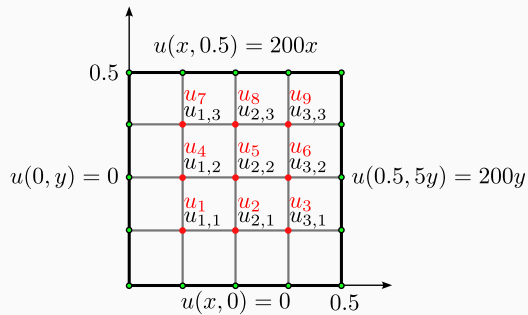$$u_{i,0} = g_{i,0} \quad \text{y} \quad u_{i,m} = g_{i,m}, \quad i = 0, 1, \ldots n$$

Resulta:

$$2\left[\left(\frac{h}{k}\right)^2 + 1\right] u_{i,j} - (u_{i+1,j} + u_{i-1,j}) - \left(\frac{h}{k}\right)^2 (u_{i,j+1} + u_{i,j-1}) = -h^2 f_{i,j}, \; i \in [1, n-1], \; j \in [1, m-1]$$

Ejemplo: Determinar la distribución estacionaria de temperaturas en una placa de 0.5×0.5 m usando $n = m = 4$. Dos bordes adyacentes se mantienen a 0 ℃ y la temperatura se incrementa linealmente en los otros bordes hasta llegar a 100 ℃ en la esquina de unión.

3

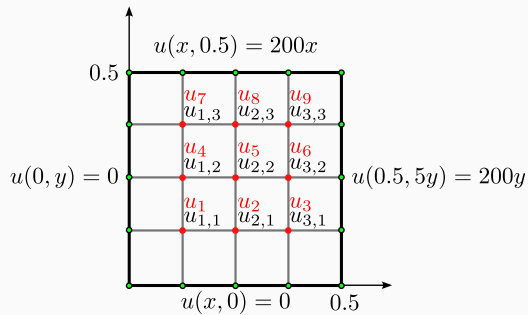$$\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = 0; \ (x,y) \in [0,0.5]^2$$

$$h = k = 1/8$$

$u_{i,j} \mapsto u_l, \ l = i + m(j-1)$

$$u_{1,1} = u_1, \ u_{2,1} = u_2, \ u_{3,1} = u_3$$

$$u_{1,2} = u_4, \ u_{2,2} = u_5, \ u_{3,2} = u_6$$

$$u_{1,3} = u_7, \ u_{2,3} = u_8, \ u_{3,3} = u_9$$

$$\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = 0; \ (x,y) \in [0,0.5]^2$$

$$h = k = 1/8$$

Ecuaciones:

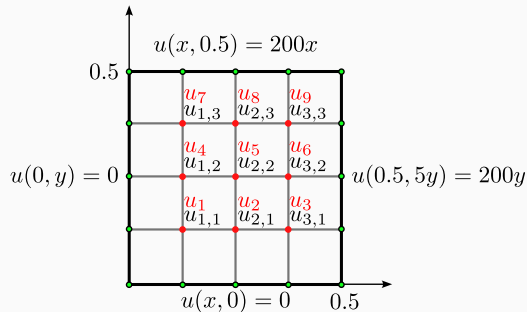$$4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j-1} - u_{i,j+1} = 0$$

para $i = 1,2,3; \ j = 1,2,3$.

$u_{i,j} \mapsto u_l, \ l = i + m(j-1)$

$$u_{1,1} = u_1, \ u_{2,1} = u_2, \ u_{3,1} = u_3$$

$$u_{1,2} = u_4, \ u_{2,2} = u_5, \ u_{3,2} = u_6$$

$$u_{1,3} = u_7, \ u_{2,3} = u_8, \ u_{3,3} = u_9$$

$$\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = 0; \ (x,y) \in [0,0.5]^2$$

$$h = k = 1/8$$

Ecuaciones:

$$4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j-1} - u_{i,j+1} = 0$$

para $i = 1, 2, 3; \ j = 1, 2, 3$.

Condiciones de borde:

$$u_{0,0} = u_{0,1} = u_{0,2} = u_{0,3} = u_{0,4} = 0$$

$$u_{1,0} = u_{2,0} = u_{3,0} = u_{4,0} = 0$$

$$u_{1,4} = u_{4,1} = 25; u_{2,4} = u_{4,2} = 50$$

$$u_{3,4} = u_{4,3} = 75; u_{4,4} = 100$$

$u_{i,j} \mapsto u_l, \ l = i + m(j-1)$

$$u_{1,1} = u_1, \ u_{2,1} = u_2, \ u_{3,1} = u_3$$

$$u_{1,2} = u_4, \ u_{2,2} = u_5, \ u_{3,2} = u_6$$

$$u_{1,3} = u_7, \ u_{2,3} = u_8, \ u_{3,3} = u_9$$

$$
\begin{bmatrix}
4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4
\end{bmatrix}
\begin{bmatrix}
u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9
\end{bmatrix}
=
\begin{bmatrix}
u_{0,1} + u_{1,0} \\
u_{0,2} \\
u_{0,3} + u_{4,1} \\
u_{0,2} \\
0 \\
u_{4,2} \\
u_{0,3} + u_{1,4} \\
u_{2,4} \\
u_{3,4} + u_{3,4}
\end{bmatrix}
\rightarrow
\begin{bmatrix}
u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9
\end{bmatrix}
=
\begin{bmatrix}
6.25 \\
12.5 \\
18.75 \\
12.5 \\
25 \\
37.5 \\
18.75 \\
37.5 \\
56.25
\end{bmatrix}
$$

$$\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{bmatrix} = \begin{bmatrix} u_{0,1} + u_{1,0} \\ u_{0,2} \\ u_{0,3} + u_{4,1} \\ u_{0,2} \\ 0 \\ u_{4,2} \\ u_{0,3} + u_{1,4} \\ u_{2,4} \\ u_{3,4} + u_{3,4} \end{bmatrix} \rightarrow \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{bmatrix} = \begin{bmatrix} 6.25 \\ 12.5 \\ 18.75 \\ 12.5 \\ 25 \\ 37.5 \\ 18.75 \\ 37.5 \\ 56.25 \end{bmatrix}$$

Solución exacta: $u(x, y) = 400xy$ por lo que la aproximación en diferencias finitas no tiene error:

$$\frac{\partial^4 u}{\partial x^4} = \frac{\partial^4 u}{\partial y^4} = 0$$
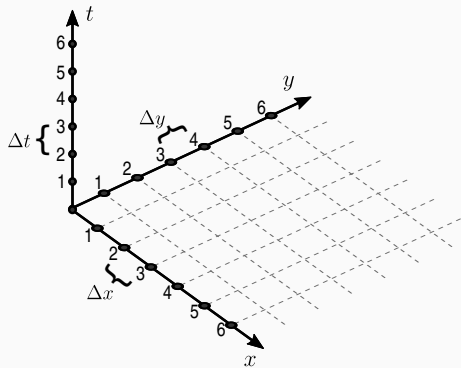
Ecuación de calor (parabólica):

$$\frac{\partial u}{\partial t} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$
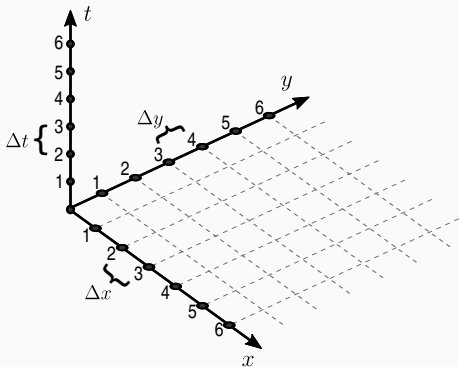
Ecuación de calor (parabólica):

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

Grilla:

$$x_i = i\Delta x, \ y_j = j\Delta y, \ t_k = k\Delta t, \ u(x, y, t) = u_{i,j}^k$$

Ecuación de calor (parabólica):

$$\frac{\partial u}{\partial t} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$
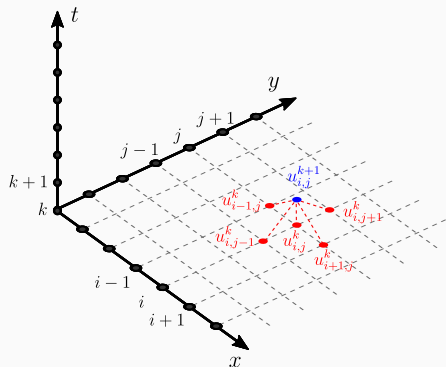
Grilla:

$$x_i = i\Delta x, \ y_j = j\Delta y, \ t_k = k\Delta t, \ u(x,y,t) = u_{i,j}^k$$

Diferencias finitas (hacia adelante – explícito):

$$\frac{u_{i,j}^{k+1} - u_{i,j}^k}{\Delta t} = \alpha \left(\frac{u_{i+1,j}^k - 2u_{i,j}^k + u_{i-1,j}^k}{\Delta x^2} \right)$$
$$+ \left(\frac{u_{i,j+1}^k - 2u_{i,j}^k + u_{i,j-1}^k}{\Delta y^2} \right)$$

Ecuación de calor (parabólica):

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

Grilla:

$$x_i = i\Delta x, \ y_j = j\Delta y, \ t_k = k\Delta t, \ u(x,y,t) = u_{i,j}^k$$

Diferencias finitas (hacia adelante – explícito):

$$\frac{u_{i,j}^{k+1} - u_{i,j}^k}{\Delta t} = \alpha \left(\frac{u_{i+1,j}^k - 2u_{i,j}^k + u_{i-1,j}^k}{\Delta x^2} \right)$$
$$+ \left(\frac{u_{i,j+1}^k - 2u_{i,j}^k + u_{i,j-1}^k}{\Delta y^2} \right)$$

Hacemos $\Delta x = \Delta y$, $\gamma = \alpha \dfrac{\Delta t}{\Delta x^2}$:

$$u_{i,j}^{k+1} = \gamma \left(u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k - 4u_{i,j}^k \right) + u_{i,j}^k$$
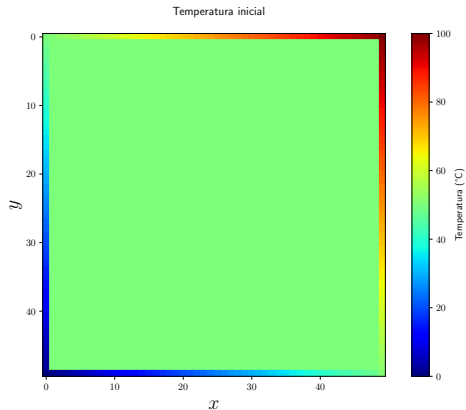
Método explícito: $\Delta t \leq \dfrac{\Delta x^2}{4\alpha} \longleftarrow$ estabilidad numérica.

Stencil:

$$u_{i,j}^{k+1} = \gamma \left(u_{i+1,j}^{k} + u_{i-1,j}^{k} + u_{i,j+1}^{k} + u_{i,j-1}^{k} - 4u_{i,j}^{k} \right) + u_{i,j}^{k}$$

$$\frac{\partial u}{\partial t} - \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = 0$$

$$0 \leq x \leq L_x, \ 0 \leq y \leq L_y$$

Temperatura inicial

Condiciones de borde:

$$u(x, 0) = 50 + \frac{x(100 - 50)}{L_x}$$

$$u(0, y) = 50 - y\frac{50}{L_y}$$

$$u(x, L_y) = 50\frac{x}{L_x}$$

$$u(L_x, y) = 100 - y\frac{50}{L_y}$$

Condición inicial:

$$u(x, y)_{t=0} = 50$$

```python
#!/usr/bin/env python3

import numpy as np

Lx, Ly = 50, 50
max_iter_tiempo = 750
alpha = 5
delta_x = 1
delta_t = (delta_x ** 2)/(4 * alpha)
gamma = (alpha * delta_t) / (delta_x ** 2)

# Condiciones de borde
u_SO = 0.0
u_NO, u_SE = 50.0, 50.0
u_NE = 100.0

# Condición inicial interior
u_inicial = 50
```

```python
def inicializar_u(max_iter_tiempo, ni=Lx, nj=Ly):
    # Inicializar la solución: u(k, i, j)
    u = np.full((max_iter_tiempo, ni, nj), u_inicial)
    # Establecer condiciones de borde
    u[:, 0, :] = u_NO + np.arange(ni) * delta_x \
                 * (u_NE - u_NO) / Lx
    u[:, :, 0] = u_NO - np.arange(nj) * delta_x \
                 * u_NO / Ly
    u[:, -1, :] = np.arange(ni) * delta_x * u_SE / Lx
    u[:, :, -1] = u_NE - np.arange(nj) * delta_x \
                  * u_SE / Ly
    return u

u = inicializar_u(max_iter_tiempo)
```
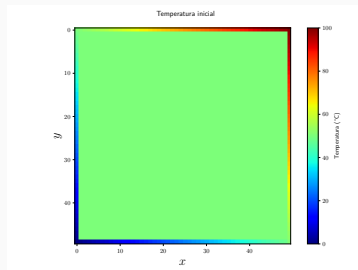

Temperatura inicial
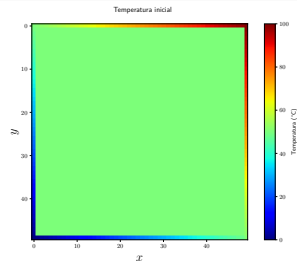
```python
35 import matplotlib
36 import matplotlib.pyplot as plt
37 matplotlib.rcParams.update({"text.usetex": True})
38 fig, ax = plt.subplots(figsize=(8,6))
39 mappable = ax.imshow(u[0], interpolation=None,
40                      cmap=plt.cm.jet)
41 fig.colorbar(mappable, label="Temperatura (°C)", ax=ax)
42 ax.set_xlabel(r"$x$", fontsize=20)
43 ax.set_ylabel(r"$y$", fontsize=20)
44 fig.suptitle("Temperatura inicial")
45 fig.tight_layout()
46 fig.savefig("temp-inicial.pdf")
47 plt.close()
```

```python
49 # Código que aplica el stencil en la grilla (i, j) y en cada t.
50 def calcular(u):
51     nk, ni, nj = u.shape
52     for k in range(0, nk-1):
53         for i in range(1, ni-1):
54             for j in range(1, nj-1):
55                 u[k + 1, i, j] = gamma * (u[k][i+1][j] + u[k][
56                                           + u[k][i][j+1] + u[k
57                                           - 4*u[k][i][j]) + u[
58     return u
59
60 u = calcular(u)
```
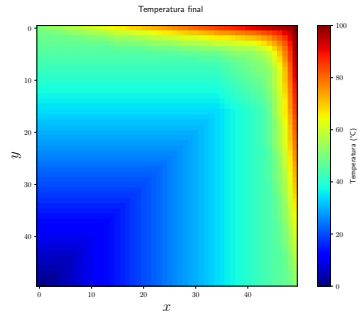
```
49  # Código que aplica el stencil en la grilla (i, j) y en cada tiempo k
50  def calcular(u):
51      nk, ni, nj = u.shape
52      for k in range(0, nk-1):
53          for i in range(1, ni-1):
54              for j in range(1, nj-1):
55                  u[k + 1, i, j] = gamma * (u[k][i+1][j] + u[k][i-1][j]
56                                            + u[k][i][j+1] + u[k][i][j-1]
57                                            - 4*u[k][i][j]) + u[k][i][j]
58      return u
59
60  u = calcular(u)
61  fig, ax = plt.subplots(figsize=(8,6))
62  mappable = ax.imshow(u[-1], interpolation=None,
63                       cmap=plt.cm.jet)
64  fig.colorbar(mappable, label="Temperatura (°C)", ax=ax)
65  ax.set_xlabel(r"$x$", fontsize=20)
66  ax.set_ylabel(r"$y$", fontsize=20)
67  fig.suptitle("Temperatura final")
68  fig.tight_layout()
69  fig.savefig("temp-final.pdf")
70  plt.close()
```
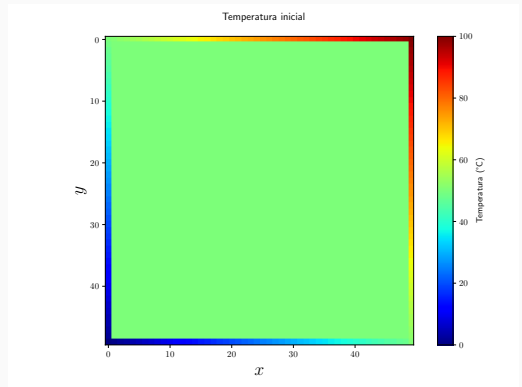
```
72  def plotheatmap(u_k, k):
73      # Limpiamos la figura
74      plt.clf()
75
76      plt.title(f"Temperatura en t = {k*delta_t:.2f} u.t.")
77      plt.xlabel(r"$x$", fontsize=20)
78      plt.ylabel(r"$y$", fontsize=20)
79
80      # Ploteamos u_k (u_{i,j} en `paso de tiempo k)
81      plt.imshow(u_k, cmap=plt.cm.jet,
82                  interpolation="bicubic", vmin=0, vmax=100)
83      plt.colorbar()
84
85      return plt
86
87  import matplotlib.animation as animation
88  from matplotlib.animation import FuncAnimation
89
90  def animate(k):
91      plotheatmap(u[k], k)
92
93  anim = animation.FuncAnimation(plt.figure(),
94      animate, interval=50, frames=max_iter_tiempo,
95      repeat=False)
96  anim.save("solucion_ecuacion_calor_t.mp4")
```
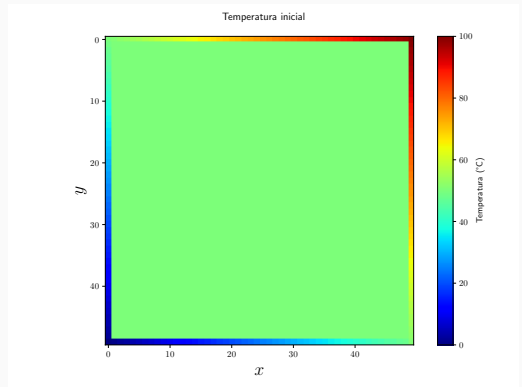
```
72 def plotheatmap(u_k, k):
73     # Limpiamos la figura
74     plt.clf()
75
76     plt.title(f"Temperatura en t = {k*delta_t:.2f} u.t.")
77     plt.xlabel(r"$x$", fontsize=20)
78     plt.ylabel(r"$y$", fontsize=20)
79
80     # Ploteamos u_k (u_{i,j} en `paso de tiempo k)
81     plt.imshow(u_k, cmap=plt.cm.jet,
82                 interpolation="bicubic", vmin=0, vmax=100)
83     plt.colorbar()
84
85     return plt
86
87 import matplotlib.animation as animation
88 from matplotlib.animation import FuncAnimation
89
90 def animate(k):
91     plotheatmap(u[k], k)
92
93 anim = animation.FuncAnimation(plt.figure(),
94     animate, interval=50, frames=max_iter_tiempo,
95     repeat=False)
96 anim.save("solucion_ecuacion_calor_t.mp4")
```

▸ R.L. Burden, D.J. Faires y A.M. Burden. *Análisis numérico.* 10.ª ed. Mexico: Cengage Learning, 2017. Capítulo 12.

▸ E. Kreyszig, H. Kreyszig y E.J. Norminton. *Advanced Engineering Mathematics.* Hoboken, USA: John Wiley & Sons, Inc, 2011. Capítulo 21.

▸ H.P. Langtangen y S. Linge. *Finite Difference Computing with PDEs – A Modern Software Approach.* https://hplgit.github.io/fdm-book/doc/web/index.html. 2016. Capítulo 3.