

UNIVERSIDAD AUTÓNOMA DE MADRID

Métodos numéricos para EDP

Comparación de diversos métodos
aplicados a la ecuación del calor

GUILLERMO RUIZ ÁLVAREZ

19 DE OCTUBRE DE 2015

Índice

1. Definición del problema	2
2. Programación del θ-método	2
3. Programación de la solución real	4
4. Ejercicio 1	6
4.1. Convergencia del θ -método	6
4.1.1. Consistencia	6
4.1.2. Estabilidad	8
4.1.3. Convergencia	9
4.2. Cálculo del orden a partir de los resultados experimentales	10
4.3. Análisis de los resultados	10
4.3.1. Método explícito	10
4.3.2. Método de Crank-Nicolson	12
4.3.3. Método implícito	13
5. Ejercicio 2	15
5.1. Método explícito	16
5.2. Método de Crank-Nicolson	17
5.3. Método implícito	17

1. Definición del problema

Consideramos el siguiente problema para la ecuación del calor

$$\begin{cases} u_t = u_{xx} & (x, t) \in (0, 1) \times (0, t_F) \\ u(0, t) = u(1, t) = 0 & t > 0 \\ u(x, 0) = u_0(x) = x(1 - x) & 0 \leq x \leq 1 \end{cases}$$

Vamos a obtener aproximaciones a la solución de este problema utilizando el θ -método para los siguientes valores de θ :

- $\theta = 0$: método explícito.
- $\theta = 1$: método implícito.
- $\theta = \frac{1}{2}$: método de Crank-Nicolson.

En todo lo que sigue se utilizará la notación y los datos que siguen:

$$\boxed{t_F = 0,6} \quad \boxed{\nu = \frac{\Delta t}{(\Delta x)^2}} \quad \boxed{\mu = \frac{\Delta t}{\Delta x}}$$

2. Programación del θ -método

El θ -método tiene la forma siguiente, con $j = 0, 1, \dots, J$ y $n = 0, 1, \dots, N$:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \theta \left[\frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{(\Delta x)^2} \right] + (1 - \theta) \left[\frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta x)^2} \right]$$

Matricialmente, el método tiene la expresión:

$$AU^{n+1} = BU^n$$

Y se basa en resolver la ecuación matricial y obtener:

$$U^{n+1} = A^{-1}BU^n$$

Donde:

$$U^n = \begin{bmatrix} U_1^n \\ U_2^n \\ \vdots \\ U_{J-1}^n \end{bmatrix} \quad U^{n+1} = \begin{bmatrix} U_1^{n+1} \\ U_2^{n+1} \\ \vdots \\ U_{J-1}^{n+1} \end{bmatrix} \quad A = \begin{bmatrix} 1 + 2\nu\theta & -\nu\theta & & \\ -\nu\theta & \ddots & \ddots & \\ & \ddots & \ddots & -\nu\theta \\ & & -\nu\theta & 1 + 2\nu\theta \end{bmatrix}$$

$$B = \begin{bmatrix} 1 - (1 - \theta)2\nu & (1 - \theta)\nu & & \\ (1 - \theta)\nu & \ddots & \ddots & \\ & \ddots & \ddots & (1 - \theta)\nu \\ & & (1 - \theta)\nu & 1 - (1 - \theta)2\nu \end{bmatrix}$$

A continuación se muestra el código en MATLAB del θ -método para la ecuación del calor con las condiciones iniciales y de contorno descritas.

```

1  % theta-method function for the heat equation problem with:
2  %   - Initial value:  $x(0) = x(1) = 0$ 
3  %   - Boundary value:  $u_0 = x(1-x)$ 
4  %
5  % Parameters:
6  %   - theta: theta value in  $[0,1]$ . (1 x 1)
7  %   - J: number of nodes for the variable x. (1 x 1)
8  %   - tF: final time. (1 x 1)
9  %   - nu: value of  $\Delta t / (\Delta x^2)$ . If nu is > 0
10 %       mu is ignored, otherwise nu is ignored. (1 x 1)
11 %   - mu: value of  $\Delta t / \Delta x$ . (1 x 1)
12 %
13 % Return:
14 %   - x: mesh for the variable x. (J+1 x 1)
15 %   - U: numerical approximation at time tF. (J+1 x 1)
16 %   - t_cpu: execution time. (1 x 1)
17 function [x, U, t_cpu] = thetamet(theta, J, tF, nu, mu)
18 % Initialize N (number of nodes for the variable t)
19 if nu > 0
20     N = round(tF*J*J/nu);
21 elseif mu > 0
22     N = round(tF*J/mu);
23     nu = mu*J;
24 end
25
26 % Initialize step sizes and variables
27 dx = 1/(J); % Step size for x
28 x = (0:dx:1)'; % J+1 nodes
29 dt = tF/(N); % Step size for t
30 t = (0:dt:tF)'; % N+1 nodes
31
32 % Initialize U
33 U = zeros(length(x), 1); % Initial value
34 U(:, 1) = (x.*(1-x)); % Boundary value
35
36 % Tridiagonal matrices
37 A = tridiag(J-1, -nu*theta, 1+2*nu*theta, -nu*theta);
38 B = tridiag(J-1, (1-theta)*nu, 1-(1-theta)*2*nu, (1-theta)*nu);
39 AinvB = A\B;
40
41 % Start stopwatch timer
42 tic;
43
44 % theta-method
45 for t_n = 1 : N
46     U(2:end-1, 1) = AinvB * U(2:end-1, 1);
47 end
48
49 % Read elapsed time from stopwatch
50 t_cpu = toc;
51 end

```

Para crear las matrices A y B se ha programado una función que acepta como parámetros el tamaño de la matriz y los valores de las tres diagonales y devuelve la matriz tridiagonal optimizada.

```

1  % Returns a tridiagonal matrix.
2  %
3  % Parameters:
4  %   -n: number of rows and columns.
5  %   -da: value for the diagonal above.
6  %   -d: value for the main diagonal.
7  %   -db: value for the diagonal below.
8  %
9  % Retorno:
10 %   -tridiag: tridiagonal matrix.
11 function [tridiag] = tridiag(n, da, d, db)
12
13     % Main diagonal.
14     diag = repmat(d, n, 1);
15
16     % Diagonal above and below.
17     diag_a = repmat(da, n-1, 1);
18     diag_b = repmat(db, n-1, 1);
19
20     % Set the correct length.
21     diag_a = [0; diag_a];
22     diag_b = [diag_b; 0];
23
24     % Return the tridiagonal matrix.
25     tridiag = spdiags ([diag_b diag diag_a], -1 : 1, n, n);
26 end

```

3. Programación de la solución real

La solución real del problema es:

$$u(x, t) = \sum_{m=1}^{\infty} B_m e^{-(m\pi)^2 t} \sin(m\pi x)$$

Los coeficientes B_m tienen la expresión:

$$B_m = 2 \int_0^1 x(1-x) \sin(m\pi x)$$

Realizando la integral por partes se obtiene:

$$B_m = \frac{4 [(-1)^{m+1} + 1]}{\pi^3 m^3}$$

Para programar el método se ha escrito una función que realiza la suma y la trunca cuando llega a cierto número de iteraciones. El código de dicha función se muestra a continuación.

```
1  % Solution for the heat equation problem with:
2  %   - Initial value:  $x(0) = x(1) = 0$ 
3  %   - Boundary value:  $u_0 = x(1-x)$ 
4  %
5  % Parameters:
6  %   - x: vector of nodes. (J+1 x 1)
7  %   - t: time value. (1 x 1)
8  %   - iter: number of iterations for the truncated sum. (1 x 1)
9  %
10 % Return:
11 %   - Usol: vector of solution values. (J+1 x 1)
12 %
13 function [Usol] = heat_sol(x, t, iter)
14     % Solution initialization
15     Usol = zeros(length(x), 1);
16
17     % Truncated sum
18     for m = 1 : iter
19         % B_m value.
20         B_m = (-4*(-1)^(m+4))/((pi*m)^3);
21
22         % Sum term.
23         sum_term = B_m * exp(-(m*pi)^2*t) * sin(m*pi*x);
24         Usol(:,1) = Usol(:,1) + sum_term;
25     end
26 end
```

4. Ejercicio 1

En la figura 1 se muestra un gráfico en escala doblemente logarítmica representando J frente al error en norma infinito en tiempo $t_F = 0,6$ usando distintos valores de ν y μ para los tres métodos.

Se han utilizado valores para J desde 10 hasta 100 con un paso de 5 y 100 iteraciones para la suma truncada.

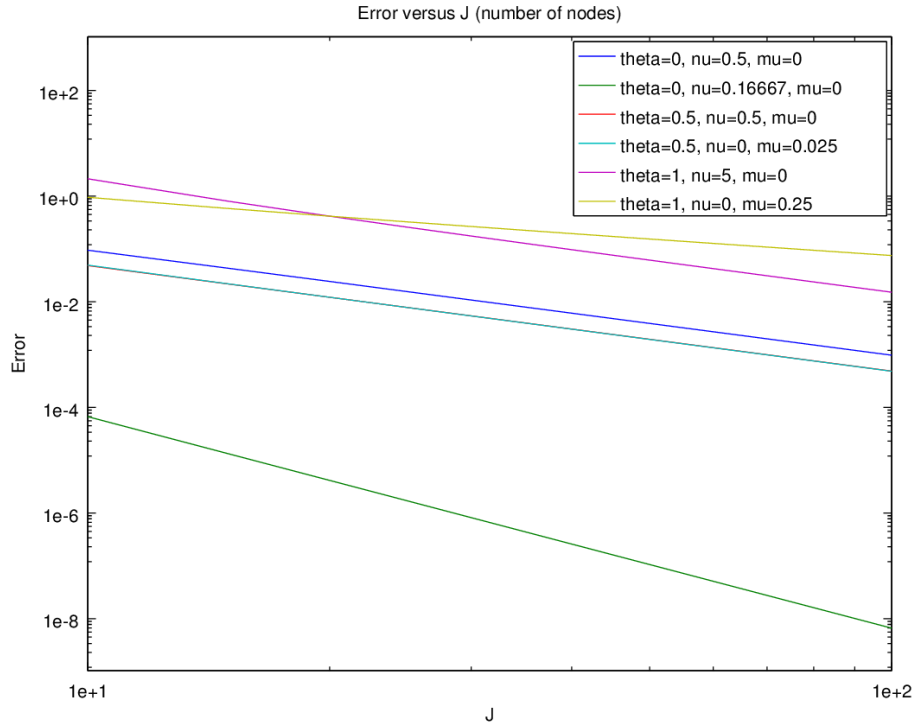


Figura 1: J frente al error en norma infinito (Escala logarítmica).

4.1. Convergencia del θ -método

Vamos a dividir el estudio de la convergencia del θ -método en dos fases. En primer lugar estudiaremos el control del residuo del esquema y seguidamente se estudiará la estabilidad del mismo.

4.1.1. Consistencia

El error de truncación puede escribirse de la siguiente forma:

$$T_j^n = \frac{u_j^{n+1} - u_j^n}{\Delta t} - \theta \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\Delta x)^2} - (1 - \theta) \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}$$

Vamos a realizar el desarrollo de Taylor en el punto $(x_j, t_{n+\frac{1}{2}})$.

- **Paso 1:** Realizamos el desarrollo de Taylor en el punto $(x_j, t_{n+\frac{1}{2}})$ del primer término.

$$u_j^{n+1} = \left[u + \left(\frac{1}{2} \Delta t \right) u_t + \frac{1}{2} \left(\frac{1}{2} \Delta t \right)^2 u_{tt} + \frac{1}{6} \left(\frac{1}{2} \Delta t \right)^3 u_{ttt} + \dots \right]_j^{n+\frac{1}{2}}$$

$$u_j^n = \left[u - \left(\frac{1}{2} \Delta t \right) u_t + \frac{1}{2} \left(\frac{1}{2} \Delta t \right)^2 u_{tt} - \frac{1}{6} \left(\frac{1}{2} \Delta t \right)^3 u_{ttt} + \dots \right]_j^{n+\frac{1}{2}}$$

Obtenemos:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \left[u_t + \frac{1}{24} (\Delta t)^2 u_{ttt} + \dots \right]_j^{n+\frac{1}{2}}$$

- **Paso 2:** Realizamos el desarrollo de Taylor en el punto (x_j, t_n) y (x_j, t_{n+1}) respectivamente de los siguientes términos:

$$u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1} = \left[\Delta x^2 u_{xx} + \frac{1}{12} \Delta x^4 u_{xxxx} + \frac{2}{6!} \Delta x^6 u_{xxxxxx} + \dots \right]_j^{n+1}$$

$$u_{j+1}^n - 2u_j^n + u_{j-1}^n = \left[\Delta x^2 u_{xx} + \frac{1}{12} \Delta x^4 u_{xxxx} + \frac{2}{6!} \Delta x^6 u_{xxxxxx} + \dots \right]_j^n$$

- **Paso 3:** Obtenemos, en el punto $(x_j, t_{n+\frac{1}{2}})$, los desarrollos de los términos anteriores, partiendo de los pasos 1 y 2:

$$\begin{aligned} u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1} &= \left[\Delta x^2 u_{xx} + \frac{1}{12} \Delta x^4 u_{xxxx} + \frac{2}{6!} \Delta x^6 u_{xxxxxx} + \dots \right]_j^{n+\frac{1}{2}} \\ &\quad + \left(\frac{1}{2} \Delta t \right) \left[\Delta x^2 u_{xxt} + \frac{1}{12} \Delta x^4 u_{xxxxt} + \dots \right]_j^{n+\frac{1}{2}} \\ &\quad + \frac{1}{2} \left(\frac{1}{2} \Delta t \right)^2 \left[\Delta x^2 u_{xxtt} + \dots \right]_j^{n+\frac{1}{2}} \end{aligned}$$

$$\begin{aligned} u_{j+1}^n - 2u_j^n + u_{j-1}^n &= \left[\Delta x^2 u_{xx} + \frac{1}{12} \Delta x^4 u_{xxxx} + \frac{2}{6!} \Delta x^6 u_{xxxxxx} + \dots \right]_j^{n+\frac{1}{2}} \\ &\quad - \left(\frac{1}{2} \Delta t \right) \left[\Delta x^2 u_{xxt} + \frac{1}{12} \Delta x^4 u_{xxxxt} + \dots \right]_j^{n+\frac{1}{2}} \\ &\quad + \frac{1}{2} \left(\frac{1}{2} \Delta t \right)^2 \left[\Delta x^2 u_{xxtt} + \dots \right]_j^{n+\frac{1}{2}} \end{aligned}$$

- **Paso 4:** Realizamos el desarrollo de Taylor en el punto $(x_j, t_{n+\frac{1}{2}})$ del segundo término.

$$\begin{aligned} \theta \left[\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2} \right] + (1 - \theta) \left[\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} \right] = \\ \left[u_{xx} + \frac{1}{12} \Delta x^2 u_{xxxx} + \frac{2}{6!} \Delta x^4 u_{xxxxxx} + \dots \right]_j^{n+\frac{1}{2}} \\ + \left(\theta - \frac{1}{2} \right) \Delta t \left[u_{xxt} + \frac{1}{12} \Delta x^2 u_{xxxxt} \right]_j^{n+\frac{1}{2}} \\ + \frac{1}{8} \Delta t^2 [u_{xxtt}]_j^{n+\frac{1}{2}} \end{aligned}$$

- **Paso 5:** Reordenamos los términos de error

$$\begin{aligned} T_j^n = & \left[\left(\frac{1}{2} - \theta \right) \Delta t u_{xxt} - \frac{1}{12} (\Delta x)^2 u_{xxxx} \right] + \left[\frac{1}{24} \Delta t^2 u_{ttt} - \frac{1}{8} \Delta t^2 u_{xxtt} \right] \\ & + \left[\frac{1}{12} \left(\frac{1}{2} - \theta \right) \Delta t \Delta x^2 u_{xxxxt} \right] - \frac{2}{6!} \Delta x^4 u_{xxxxxx} \end{aligned}$$

Escrito de otro modo, tenemos que

$$\begin{aligned} T_j^n = & \left[\left(\frac{1}{2} - \theta \right) \Delta t u_{xxt} - \frac{1}{12} (\Delta x)^2 u_{xxxx} \right] + \left[\frac{1}{24} \Delta t^2 u_{ttt} - \frac{1}{8} \Delta t^2 u_{xxtt} \right] \\ & + \left[\frac{1}{12} \left(\frac{1}{2} - \theta \right) \Delta t \Delta x^2 u_{xxxxt} \right] - \frac{2}{6!} \Delta x^4 u_{xxxxxx} \end{aligned}$$

En general tenemos que $T_j^n = \mathcal{O}(\Delta t) + \mathcal{O}((\Delta x)^2)$. En conclusión, el método es incondicionalmente consistente ya que

$$T_j^n \xrightarrow{\Delta t \rightarrow 0, \Delta x \rightarrow 0} 0$$

4.1.2. Estabilidad

Para estudiar la estabilidad, vamos a realizar el análisis de Fourier del método. La solución general del problema para la ecuación del calor tiene la forma siguiente, con B_m con la expresión que ya se ha definido:

$$u(x, t) = \sum_{m=-\infty}^{\infty} B_m e^{-(m\pi)^2 t} e^{i(m\pi)x}$$

Para realizar el análisis de Fourier del método explícito vamos a buscar el factor de amplificación $\lambda(k)$ de forma que $U_j^n = \lambda^n(k) e^{ikj\Delta x}$, con $k = m\pi$. Partimos escribiendo el método de la siguiente forma:

$$U_j^{n+1} = U_j^n + \nu \theta [U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}] + \nu(1 - \theta) [U_{j+1}^n - 2U_j^n + U_{j-1}^n]$$

- **Paso 1:** Sustituimos U_j^n por $\lambda^n(k)e^{ikj\Delta x}$ en el método obteniendo

$$\begin{aligned}\lambda^{n+1}e^{ikj\Delta x} &= \lambda^n e^{ikj\Delta x} \\ &\quad + \nu\theta\lambda^{n+1} \left[e^{ik(j+1)\Delta x} - 2e^{ik(j)\Delta x} + e^{ik(j-1)\Delta x} \right] \\ &\quad + \nu(1-\theta)\lambda^n \left[e^{ik(j+1)\Delta x} - 2e^{ik(j)\Delta x} + e^{ik(j-1)\Delta x} \right]\end{aligned}$$

- Dividimos por $\lambda^n e^{ikj\Delta x}$:

$$\begin{aligned}\lambda &= 1 + \nu\theta\lambda \left[e^{ik\Delta x} - 2 + e^{-ik\Delta x} \right] + \nu(1-\theta) \left[e^{ik\Delta x} - 2 + e^{-ik\Delta x} \right] \\ &= 1 + \nu\theta\lambda \left[-4\sin^2 \left(\frac{k\Delta x}{2} \right) \right] + \nu(1-\theta) \left[-4\sin^2 \left(\frac{k\Delta x}{2} \right) \right]\end{aligned}$$

- Despejamos λ :

$$\lambda(k) = \frac{1 - 4\nu(1-\theta) \left[\sin^2 \left(\frac{k\Delta x}{2} \right) \right]}{1 + 4\nu\theta \left[\sin^2 \left(\frac{k\Delta x}{2} \right) \right]}$$

Tenemos inestabilidad si $|\lambda| > 1$, ya que implicaría que la serie infinita diverge. Siempre tenemos que $\lambda < 1$, por tanto esta condición es equivalente a $\lambda < -1$, luego tendríamos:

$$1 - 4\nu(1-\theta) \left[\sin^2 \left(\frac{k\Delta x}{2} \right) \right] < -1 - 4\nu\theta \left[\sin^2 \left(\frac{k\Delta x}{2} \right) \right]$$

Que es equivalente a la siguiente condición:

$$2 < 4\nu(1-2\theta) \left[\sin^2 \left(\frac{k\Delta x}{2} \right) \right] < 4\nu(1-2\theta)$$

Como conclusión podemos resumir la estabilidad del método, en función del valor de θ , como sigue:

- $0 < \theta < \frac{1}{2}$: estabilidad si $\nu < \frac{1}{2(1-2\theta)}$.
- $\frac{1}{2} \leq \theta \leq 1$: estabilidad incondicional.

4.1.3. Convergencia

En cuanto a la convergencia, concluimos que:

- Método explícito ($\theta = 0$): Convergente si $\nu < \frac{1}{2}$.
- Método de Crank-Nicolson ($\theta = \frac{1}{2}$): Convergente siempre.
- Método implícito ($\theta = 1$): Convergente siempre.

4.2. Cálculo del orden a partir de los resultados experimentales

Para calcular el orden de convergencia del θ -método hay que hallar las pendientes de las rectas representadas en escala logarítmica. La razón es la siguiente, supongamos que el error del método es $E = \mathcal{O}((\Delta x)^s)$ es decir:

$$E = C \cdot (\Delta x)^s = \frac{C}{J^s} = C \cdot J^{-s}$$

para una constante C . Tomando logaritmos tenemos que:

$$\log(E) = -s \cdot \log(J) + \log(C) = -s \cdot \log(J) + K$$

Obteniendo que $-s$ es la pendiente de la recta representada en escala logarítmica, siendo s el orden del método.

Para calcular las pendientes de las rectas se ha utilizado la función `polyfit` de MATLAB que utiliza mínimos cuadrados para calcular los coeficientes del polinomio que mejor se aproxime a los datos. De esta manera, se han calculado las pendientes de cada recta de la siguiente forma. Si J es el vector de los pasos utilizados y err es el vector de errores, la pendiente de la recta se calcula como sigue:

```
1 % Degree 1.
2 poly = polyfit(log(J), log(err), 1);
3 slope = poly(1)
```

4.3. Análisis de los resultados

4.3.1. Método explícito

En esta sección se analizan los resultados para el método explícito ($\theta = 0$) con los valores $\nu = \frac{1}{2}$ y $\nu = \frac{1}{6}$. En la figura 2 se muestra un gráfico comparando los resultados de ambos casos.

- Caso $\nu = \frac{1}{2}$: Para este valor de ν , el método es estable. En este caso vemos que el orden del error es

$$E = \mathcal{O}(\Delta t) + \mathcal{O}((\Delta x)^2)$$

Como tenemos un valor fijo para $\nu = \frac{\Delta t}{(\Delta x)^2} = \frac{1}{2}$, tenemos que

$$E = \mathcal{O}((\Delta x)^2)$$

luego

$$E = C \cdot (\Delta x)^2 = C \cdot \frac{1}{J^2}$$

con C una constante. Tomando logaritmos obtenemos que la pendiente teórica ha de ser -2 :

$$\log(E) = -2\log(J) + K$$

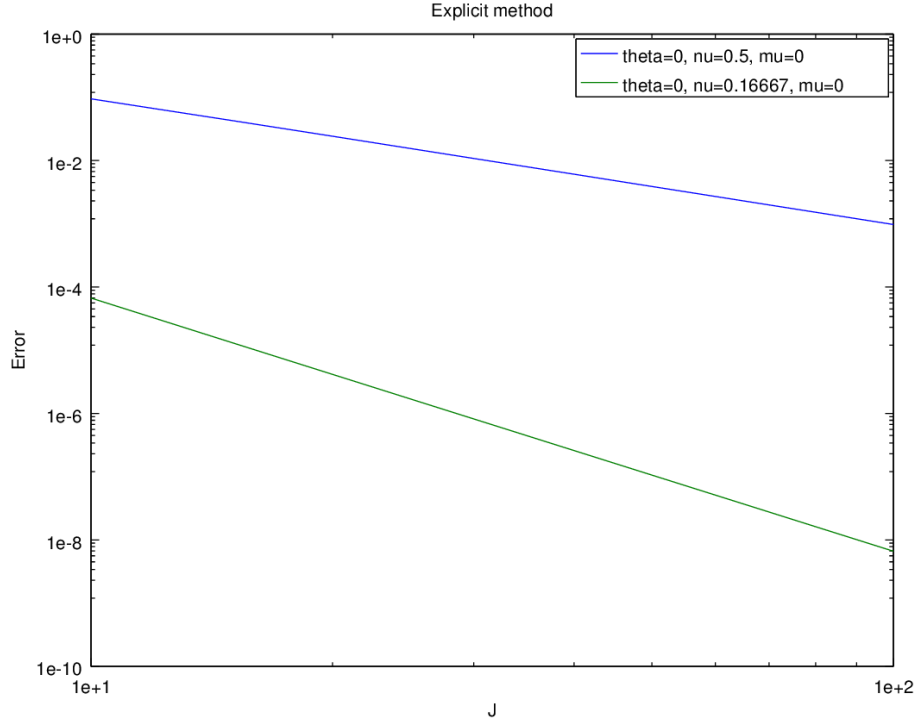


Figura 2: Resultados del método explícito.

Experimentalmente se ha obtenido que la pendiente de la recta es $-1,9948 \approx -2$. El error es lineal sobre la variable temporal y cuadrático sobre la variable espacial.

- Caso $\nu = \frac{1}{6}$: Para este valor de ν , el método es estable. Este es un caso particular en el que el error tiene el orden

$$E = \mathcal{O}((\Delta t)^2) + \mathcal{O}((\Delta x)^4)$$

Como tenemos un valor fijo para $\nu = \frac{\Delta t}{(\Delta x)^2} = \frac{1}{6}$, tenemos que

$$E = \mathcal{O}((\Delta x)^4)$$

luego

$$E = C \cdot (\Delta x)^4 = C \cdot \frac{1}{J^4}$$

con C una constante. Tomando logaritmos obtenemos que la pendiente teórica ha de ser -4 :

$$\log(E) = -4\log(J) + K$$

Experimentalmente se ha obtenido que la pendiente de la recta es $-4,0027 \approx -4$. El error es de orden 2 sobre la derivada temporal y de orden 4 sobre la derivada espacial.

4.3.2. Método de Crank-Nicolson

En esta sección se analizan los resultados para el método de Crank-Nicolson ($\theta = \frac{1}{2}$) con los valores $\nu = \frac{1}{2}$ y $\mu = \frac{1}{40}$. En la figura 3 se muestra un gráfico comparando los resultados de ambos casos.

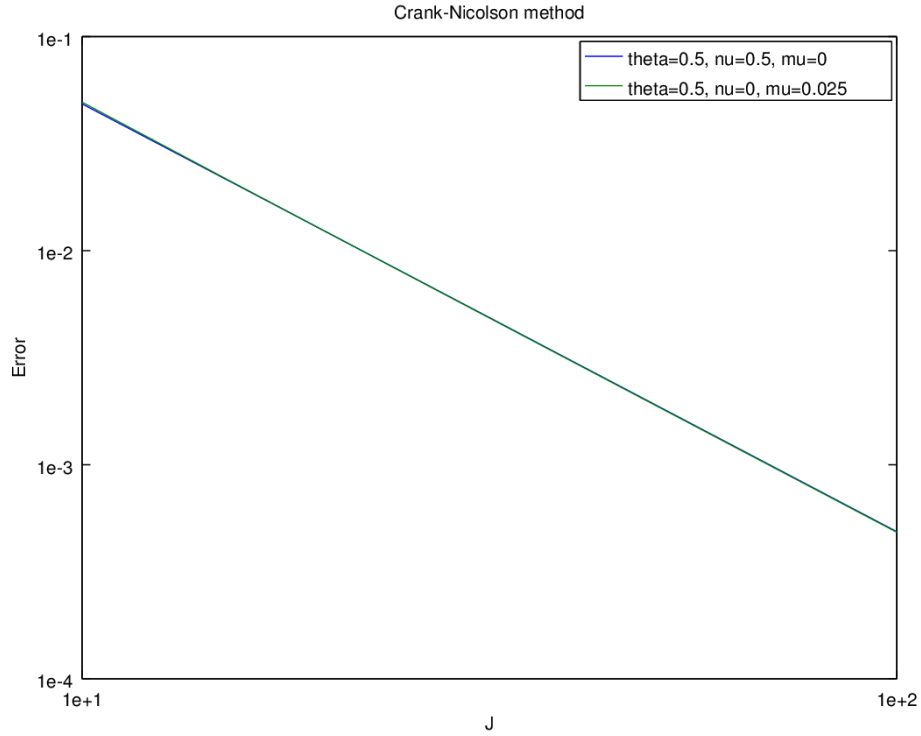


Figura 3: Resultados del método de Crank Nicolson.

- Caso $\nu = \frac{1}{2}$: En este caso el orden del error es

$$E = \mathcal{O}((\Delta t)^2) + \mathcal{O}((\Delta x)^2)$$

Como tenemos un valor fijo para $\nu = \frac{\Delta t}{(\Delta x)^2} = \frac{1}{2}$, tenemos que

$$E = \mathcal{O}((\Delta x)^2)$$

luego

$$E = C \cdot (\Delta x)^2 = C \cdot \frac{1}{J^2}$$

con C una constante. Tomando logaritmos obtenemos que la pendiente teórica ha de ser -2 :

$$\log(E) = -2\log(J) + K$$

Experimentalmente se ha obtenido que la pendiente de la recta es $-1,9986 \approx -2$, luego el método tiene orden 2. El error es cuadrático sobre ambas variables, sin embargo, al tener un valor fijo para ν y por tanto un valor fijo para la relación entre Δt y $(\Delta x)^2$, tenemos que $E = \mathcal{O}((\Delta x)^2) = \mathcal{O}(\Delta t)$.

- Caso $\mu = \frac{1}{40}$: En este caso el orden del error es

$$E = \mathcal{O}((\Delta t)^2) + \mathcal{O}((\Delta x)^2)$$

Como tenemos un valor fijo para $\mu = \frac{\Delta t}{\Delta x} = \frac{1}{40}$, tenemos que

$$E = \mathcal{O}((\Delta x)^2)$$

luego

$$E = C \cdot (\Delta x)^2 = C \cdot \frac{1}{J^2}$$

con C una constante. Tomando logaritmos obtenemos que la pendiente teórica ha de ser -2 :

$$\log(E) = -2\log(J) + K$$

Experimentalmente se ha obtenido que la pendiente de la recta es $-1,9986 \approx -2$, luego el método tiene orden 2. Este método presenta un cambio con respecto al anterior, al fijar el valor de μ se quita la condición de que la relación entre el paso temporal y el cuadrado del paso espacial sea fija, obteniendo así orden cuadrático en ambas variables para el error.

4.3.3. Método implícito

En esta sección se analizan los resultados para el método implícito ($\theta = 1$) con los valores $\nu = 5$ y $\mu = \frac{1}{4}$. En la figura 4 se muestra un gráfico comparando los resultados de ambos casos.

- Caso $\nu = 5$: En este caso el orden del error es

$$E = \mathcal{O}(\Delta t) + \mathcal{O}((\Delta x)^2)$$

Como tenemos un valor fijo para $\nu = \frac{\Delta t}{(\Delta x)^2} = 5$, tenemos que

$$E = \mathcal{O}((\Delta x)^2)$$

luego

$$E = C \cdot (\Delta x)^2 = C \cdot \frac{1}{J^2}$$

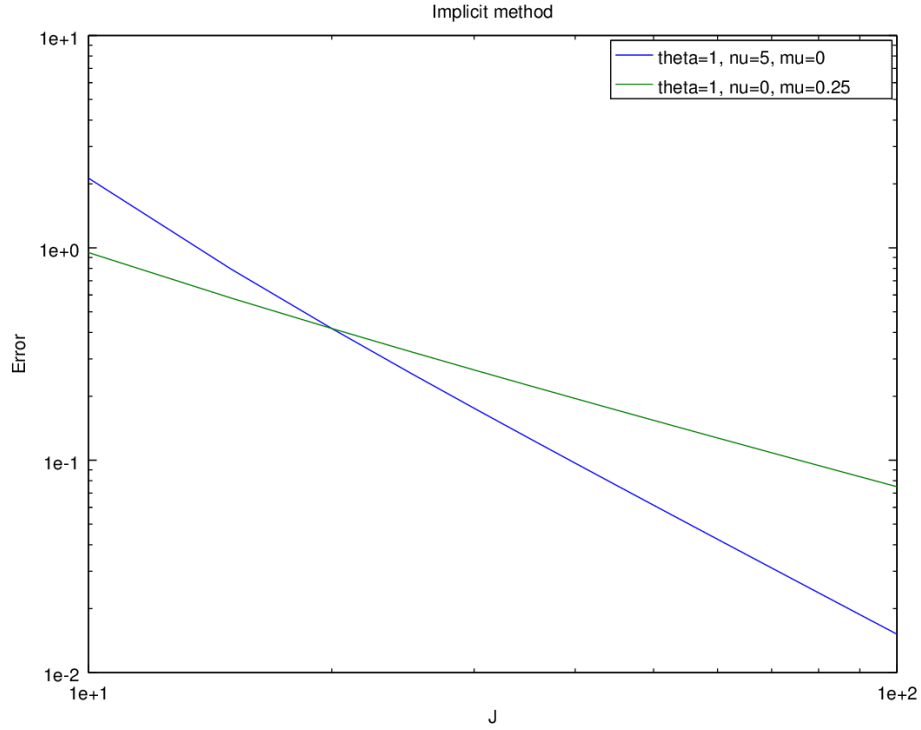


Figura 4: Resultados del método implícito.

con C una constante. Tomando logaritmos obtenemos que la pendiente teórica ha de ser -2 :

$$\log(E) = -2\log(J) + K$$

Vemos que en este caso da igual que $\nu > \frac{1}{2}$ pues el método implícito es siempre convergente y funciona bien para ν grande.

Experimentalmente se ha obtenido que la pendiente de la recta es $-2,1079 \approx -2$. El error es lineal en la variable temporal y cuadrático en la variable espacial.

- Caso $\mu = \frac{1}{4}$: En este caso el orden del error es

$$E = \mathcal{O}(\Delta t) + \mathcal{O}((\Delta x)^2)$$

Como tenemos un valor fijo para $\mu = \frac{\Delta t}{\Delta x} = \frac{1}{4}$, tenemos que

$$E = \mathcal{O}(\Delta t) = \mathcal{O}(\Delta x)$$

luego

$$E = C \cdot \Delta t = C \cdot \frac{1}{J}$$

con C una constante. Tomando logaritmos obtenemos que la pendiente teórica ha de ser -1 :

$$\log(E) = -\log(J) + K$$

Experimentalmente se ha obtenido que la pendiente de la recta es $-1,0874 \approx -1$. El orden del error es lineal para ambas variables, ya que están relacionadas linealmente por el valor fijo de μ .

Se puede observar que este método aproxima el resultado de peor forma que en el caso anterior. Esto se debe a que, al fijar el valor de μ , el orden del error del método para la variable espacial se reduce a 1.

5. Ejercicio 2

En la figura 5 se muestra un gráfico en escala doblemente logarítmica representando t_{cpu} frente al error en norma infinito en tiempo $t_F = 0,6$ usando distintos valores de ν y μ para los tres métodos.

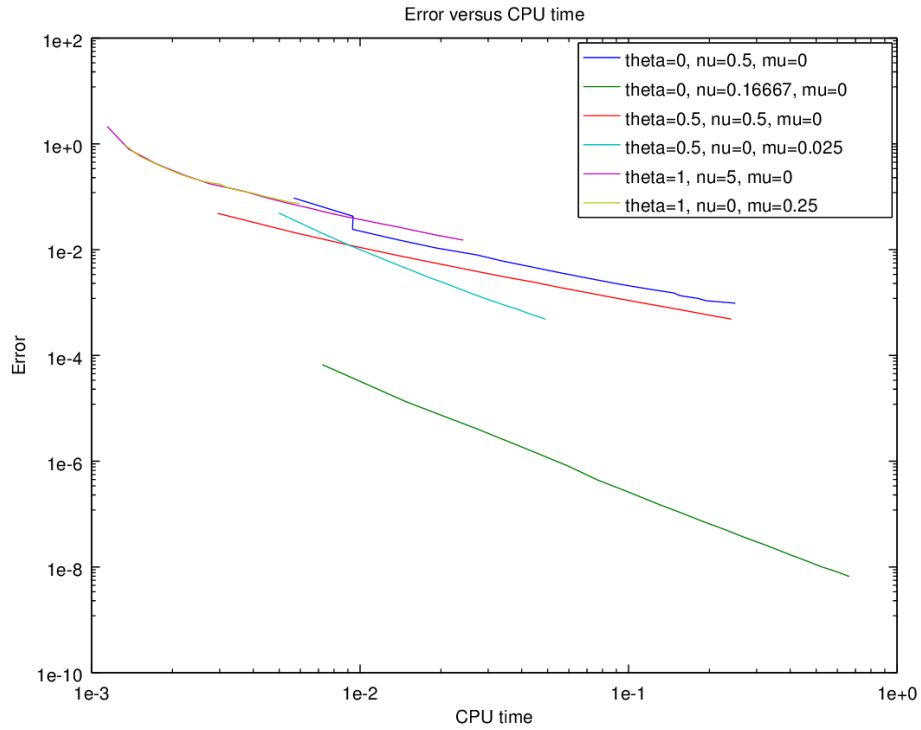


Figura 5: t_{cpu} frente al error en norma infinito (Escala logarítmica).

Dado que el tiempo de CPU es función tanto de el número de nodos para

la variable temporal como el número de nodos de la variable espacial, tenemos que, si C y K son constantes:

- Si ν es constante:

$$t_{cpu} = C \cdot N \cdot J = C \cdot \frac{J^2}{\nu} \cdot J = K J^3$$

- Si μ es constante:

$$t_{cpu} = C \cdot N \cdot J = C \cdot \frac{J}{\mu} \cdot J = K J^2$$

A partir de estos valores y de los errores obtenidos en el ejercicio anterior, se podrán calcular las pendientes teóricas de las rectas representadas en escala logarítmica.

5.1. Método explícito

- Caso $\nu = \frac{1}{2}$: En este caso hemos visto que el error es de orden 2 para la variable espacial.

Tenemos

$$\left. \begin{array}{l} E = C_1 \cdot J^{-2} \\ t_{cpu} = C_2 \cdot J^3 \end{array} \right\} \implies E = K \cdot t_{cpu}^{-\frac{2}{3}}$$

Luego tomando logaritmos

$$\log(E) = -\frac{2}{3} \log(t_{cpu})$$

Teóricamente, la pendiente de la recta ha de ser $-\frac{2}{3} \approx -0,66667$. El resultado experimental es que la pendiente obtenida es $-1,15814$.

- Caso $\nu = \frac{1}{6}$: En este caso hemos visto que el error es de orden 4 para la variable espacial.

Tenemos

$$\left. \begin{array}{l} E = C_1 \cdot J^{-4} \\ t_{cpu} = C_2 \cdot J^3 \end{array} \right\} \implies E = K \cdot t_{cpu}^{-\frac{4}{3}}$$

Luego tomando logaritmos

$$\log(E) = -\frac{4}{3} \log(t_{cpu})$$

Teóricamente, la pendiente de la recta ha de ser $-\frac{4}{3} \approx -1,3333$. El resultado experimental es que la pendiente obtenida es $-1,94958$.

Se puede observar que la pendiente en este caso es muy negativa, lo que implica que el método aproximará con menos error en menos tiempo.

El error en el cálculo de los resultados experimentales puede deberse a que el método explícito es computacionalmente menos costoso que el resto de métodos, hecho que puede provocar que la medida sea menos precisa, al tratarse de valores muy pequeños para el tiempo.

5.2. Método de Crank-Nicolson

- Caso $\nu = \frac{1}{2}$: En este caso hemos visto que el error es de orden 2 para la variable espacial.

Tenemos

$$\left. \begin{array}{lcl} E & = & C_1 \cdot J^{-2} \\ t_{cpu} & = & C_2 \cdot J^3 \end{array} \right\} \implies E = K \cdot t_{cpu}^{-\frac{2}{3}}$$

Luego tomando logaritmos

$$\log(E) = -\frac{2}{3}\log(t_{cpu})$$

Teóricamente, la pendiente de la recta ha de ser $-\frac{2}{3} \approx -0,66667$. El resultado experimental es que la pendiente obtenida es $-0,76695$.

- Caso $\mu = \frac{1}{40}$: En este caso hemos visto que el error es de orden 2 para la variable espacial.

Tenemos

$$\left. \begin{array}{lcl} E & = & C_1 \cdot J^{-2} \\ t_{cpu} & = & C_2 \cdot J^2 \end{array} \right\} \implies E = K \cdot t_{cpu}^{-1}$$

Luego tomando logaritmos

$$\log(E) = -\log(t_{cpu})$$

Teóricamente, la pendiente de la recta ha de ser -1 . El resultado experimental es que la pendiente obtenida es $-1,23301$.

En este caso también se puede observar que la pendiente es muy negativa. Hay que tener en cuenta que, dado que $E = \mathcal{O}((\Delta t)^2) + \mathcal{O}((\Delta x)^2)$, se puede elegir μ fija y por tanto Δt del mismo orden que Δx , lo que implica que no hay que reducir tanto el paso temporal.

5.3. Método implícito

- Caso $\nu = 5$: En este caso hemos visto que el error es de orden 2 para la variable espacial.

Tenemos

$$\left. \begin{array}{lcl} E & = & C_1 \cdot J^{-2} \\ t_{cpu} & = & C_2 \cdot J^3 \end{array} \right\} \implies E = K \cdot t_{cpu}^{-\frac{2}{3}}$$

Luego tomando logaritmos

$$\log(E) = -\frac{2}{3}\log(t_{cpu})$$

Teóricamente, la pendiente de la recta ha de ser $-\frac{2}{3} \approx -0,66667$. El resultado experimental es que la pendiente obtenida es $-0,81102$.

- Caso $\mu = \frac{1}{4}$: En este caso hemos visto que el error es de orden 1 para la variable espacial.

Tenemos

$$\left. \begin{array}{lcl} E & = & C_1 \cdot J^{-1} \\ t_{cpu} & = & C_2 \cdot J^2 \end{array} \right\} \Rightarrow E = K \cdot t_{cpu}^{-\frac{1}{2}}$$

Luego tomando logaritmos

$$\log(E) = -\frac{1}{2}\log(t_{cpu})$$

Teóricamente, la pendiente de la recta ha de ser $-\frac{1}{2}$. El resultado experimental es que la pendiente obtenida es $-0,67017$.

Los métodos que ofrecen una pendiente más negativa, es decir, que aproximan la solución con menos error en un tiempo menor, son el método explícito con $\nu = \frac{1}{6}$ y el método de Crank-Nicolson con $\nu = \frac{1}{40}$. El método de Crank-Nicolson tiene la ventaja de que no obliga a tomar pasos temporales muy pequeños, sino que se pueden tomar del mismo orden que el paso para la variable espacial.