Universidad Nacional del Litoral

Mecánica Computacional

Docentes:
Norberto Marcelo Nigro (nnigro@intec.unl.edu.ar)
Gerardo Franck (gerardofranck@yahoo.com.ar)
Diego Sklar (diegosklar@gmail.com)
Carlos Gentile (csgentile@gmail.com)

GUIA DE TRABAJOS PRACTICOS Nº 1

INTRODUCCIÓN A MODELOS DE ECUACIONES
MÉTODO DE DIFERENCIAS FINITAS

Problema Estacionario: Ecuación de Poisson

Recordemos el problema de difusión en una dimensión

$$c\rho u_t - K_0 u_{xx} = Q, \quad u = u(x,t), \qquad 0 \leq x \leq L, \ t \geq 0.$$

Al considerar el *estado estacionario* en que la variable $u(x,t)$ no cambia con respecto al tiempo t, tenemos $u_t = 0$ y obtenemos la ecuación de Poisson

$$-K_0 u_{xx} = Q, \quad u = u(x), \qquad 0 \leq x \leq L.$$

En la siguiente sección vemos cómo se discretiza el problema de Poisson unidimensional con condiciones de borde de tipo Dirichlet.

Condiciones de borde Dirichlet

Consideremos el problema

$$\begin{cases} -K_0 u_{xx} = Q, & 0 \leq x \leq L, \\ u(0) = a, \\ u(L) = b. \end{cases} \tag{1}$$

Idea: La idea principal del método de diferencias finitas consiste en reemplazar derivadas por cocientes de diferencias.

Recordando que $\dfrac{dg}{dx}(x) = \lim_{h\to 0} \dfrac{g(x+h)-g(x)}{h}$, la idea es tomar h pequeño y aproximar $\frac{dg}{dx}(x)$ por $\frac{g(x+h)-g(x)}{h}$, o alguna variante:

Diferencia	Fórmula	Error: $\left\lvert \text{fórmula} - \dfrac{dg}{dx} \right\rvert$
Adelantada:	$\dfrac{dg}{dx}(x) \approx \dfrac{g(x+h)-g(x)}{h}$	$O(h)$ si $g \in C^2$
Atrasada:	$\dfrac{dg}{dx}(x) \approx \dfrac{g(x)-g(x-h)}{h}$	$O(h)$ si $g \in C^2$
Centrada:	$\dfrac{dg}{dx}(x) \approx \dfrac{g(x+\frac{h}{2})-g(x-\frac{h}{2})}{h}$	$O(h^2)$ si $g \in C^3$

Las acotaciones del error se obtienen utilizando el Teorema de Taylor. Recordemos que en la ecuación de Poisson aparece $u_{xx} = \frac{d^2u}{dx^2}$. Para derivadas segundas hacemos lo siguiente, utilizando diferencias centradas:

$$
\begin{aligned}
\frac{d^2u}{dx^2}(x) &= \frac{d}{dx}\left(\frac{du}{dx}\right)(x) \approx \frac{\frac{du}{dx}(x+\frac{h}{2}) - \frac{du}{dx}(x-\frac{h}{2})}{h} \\
&\approx \frac{\frac{u(x+\frac{h}{2}+\frac{h}{2})-u(x+\frac{h}{2}-\frac{h}{2})}{h} - \frac{u(x-\frac{h}{2}+\frac{h}{2})-u(x-\frac{h}{2}-\frac{h}{2})}{h}}{h} \\
&= \frac{\frac{u(x+h)-u(x)}{h} - \frac{u(x)-u(x-h)}{h}}{h} \\
&= \frac{u(x+h)-2u(x)+u(x-h)}{h^2}
\end{aligned}
$$

Si $u \in C^4$ *entonces*

$$
\frac{d^2u}{dx^2}(x) = \frac{u(x+h)-2u(x)+u(x-h)}{h^2} + O(h^2).
$$

$$\frac{d^2u}{dx^2}(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} + O(h^2).$$

Más precisamente, si $M_4 = \max_{[x-h,x+h]} |u^{IV}|$, *entonces*

$$\left|\frac{d^2u}{dx^2}(x) - \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}\right| \leq \frac{M_4}{12}\, h^2.$$

Demostración. Observemos que por el teorema de Taylor, como $u \in C^4$, resulta

$$\begin{aligned}
u(x+h) &= u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{3!}u'''(x) + \frac{h^4}{4!}u^{IV}(\xi_1) \\
u(x-h) &= u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{3!}u'''(x) + \frac{h^4}{4!}u^{IV}(\xi_2),
\end{aligned}$$

con $x - h < \xi_2 < x < \xi_1 < x + h$. Luego, sumando ambas igualdades, obtenemos

$$u(x+h) - 2u(x) + u(x-h) = h^2 u''(x) + \frac{h^4}{12}\frac{u^{IV}(\xi_1) + u^{IV}(\xi_2)}{2}.$$

Como u^{IV} es continua, $\frac{u^{IV}(\xi_1)+u^{IV}(\xi_2)}{2} = u^{IV}(\eta)$ para algún $x - h < \eta < x + h$, y luego

$$\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = u''(x) + \frac{h^2}{12}u^{IV}(\eta),$$

que implica la afirmación de la proposición.

En vista de esta proposición, si u es la solución de (1) , resulta

$$-K_0 \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} \approx Q(x),$$

siempre que $x-h, x, x+h \in [0, L]$, es decir, si $h \leq x \leq L-h$. El error de aproximación en esta fórmula es $O(h^2)$.

Para obtener un método numérico, tomamos $N \in \mathbb{N}$ y $h = \dfrac{L}{N}$, y definimos la partición

$$0 = x_1 < x_2 = h < x_3 = 2h < \ldots < x_N = L - h < x_{N+1} = L.$$

Si observamos que $x_i - h = x_{i-1}$, y que $x_i + h = x_{i+1}$, resulta

$$-K_0 \frac{u(x_i+h) - 2u(x_i) + u(x_i-h)}{h^2} = -K_0 \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} \approx Q(x_i)$$

para $i = 2, 3, 4, \ldots, L$. Además, $u(x_1) = a$ y $u(x_{N+1}) = b$. Resumiendo,

$$\begin{cases} u(x_1) = a, & \\ -K_0 \dfrac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} \approx Q(x_i), & i = 2, 3, \ldots, N, \\ u(x_{N+1}) = b. & \end{cases}$$

La idea es entonces definir aproximaciones $U_i \approx u(x_i)$ que cumplan las fórmulas anteriores *con igualdad*, es decir

$$\begin{cases} U_1 = a, \\ -K_0 \dfrac{U_{i+1} - 2U_i + U_{i-1}}{h^2} = Q(x_i), \qquad i = 2, 3, \ldots, N, \\ U_{N+1} = b. \end{cases}$$

Multiplicando por $\frac{h^2}{K_0}$ llegamos al siguiente sistema de ecuaciones:

$$\begin{cases} U_1 = a, \\ -U_{i-1} + 2U_i - U_{i+1} = \dfrac{h^2}{K_0} Q(x_i), \qquad i = 2, 3, \ldots, N, \\ U_{N+1} = b. \end{cases}$$

Este sistema de ecuaciones tiene $N+1$ ecuaciones y $N+1$ incógnitas $U_1, U_2, \ldots, U_{N+1}$. El mismo, puede reescribirse de la siguiente manera:

$$\begin{cases} U_1 & = a \\ -U_1 + 2U_2 - U_3 & = \dfrac{h^2}{K_0} Q(x_2) \\ \quad - U_2 + 2U_3 - U_4 & = \dfrac{h^2}{K_0} Q(x_3) \\ \qquad \ddots & \vdots \\ \qquad -U_{N-1} + 2U_N - U_{N+1} & = \dfrac{h^2}{K_0} Q(x_N) \\ \qquad U_{N+1} & = b. \end{cases}$$

En forma matricial resulta

$$\begin{bmatrix} 1 & 0 & 0 & \ldots & \ldots & 0 \\ -1 & 2 & -1 & 0 & \ldots & 0 \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & -1 & 2 & -1 \\ 0 & 0 & \ldots & \ldots & 0 & 1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_N \\ U_{N+1} \end{bmatrix} = \begin{bmatrix} a \\ \frac{h^2}{K_0} Q(x_2) \\ \frac{h^2}{K_0} Q(x_3) \\ \vdots \\ \frac{h^2}{K_0} Q(x_N) \\ b \end{bmatrix}.$$

He aquí una implementación computacional, que consiste en escribir un programa que ensamble el sistema y lo resuelva:

ESTO ES LO QUE DEBERIAN PROGRAMAR PARA EL CASO ESTACIONARIO

Teorema 53. *Si la solución exacta $u(x)$ de* (1) *es $C^4[0, L]$ entonces*

$$\max_{i=1,2,\ldots,N+1} |u(x_i) - U_i| \leq CM_4h^2 = CM_4L^2\frac{1}{N^2},$$

donde $M_4 = \max_{[0,L]} |u^{IV}|$ y C es una constante que depende de los parámetros K_0, L, de la ecuación, pero es independiente de la función $Q(x)$, y de los datos de borde a, b y del parámetro de discretización h (o N).

Demostración. La demostración de este teorema queda fuera del alcance de este curso. El lector interesado puede encontrarla en [Larsson-Thomée 2009, Ch. 4] $\square$

Observación 54. Cuando uno escribe un programa como el anterior para resolver un problema, debe verificar que funcione bien. Para ello, se busca una solución exacta. Esto parece difícil, pero es cuestión de elegir una $u(x)$ cualquiera, que sea $C^4[0, L]$ aunque no sea muy obvia, como lineal o cuadrática y luego, dado un k elegido, se calcula $f(x)$, $u(0)$ y $u(L)$.

Por ejemplo: Tomemos $L = 1$, $K_0 = 2$ y $u(x) = x^2 + e^{-(x-0.5)^2}$. Para que esta u sea solución de (7.1), calculamos $Q = -K_0 u_{xx}$:

$$u_x(x) = 2x - 2(x - 0.5)e^{-(x-0.5)^2}, \qquad u_{xx}(x) = 2 - 2[1 - 2(x - 0.5)^2]e^{-(x-0.5)^2}$$

Luego, resolvemos el problema con

$$\begin{aligned} Q(x) &= -4 + 4[1 - 2(x - 0.5)^2]e^{-(x-0.5)^2} && \Big(= -K_0 u_{xx}(x) \quad \text{con} \quad K_0 = 2\Big) \\ a &= e^{-0.5^2} && \Big(= u(0)\Big) \\ b &= 1 + e^{-0.5^2} && \Big(= u(1) \quad \text{con} \quad L = 1\Big) \end{aligned}$$

y deberíamos obtener una solución aproximada a $u(x) = x^2 + e^{-(x-0.5)^2}$.

Para estar seguros que el código está bien programado, debemos verificar que cuando tomamos h más pequeño, el error máximo se reduce como Ch^2. Es decir, cada vez que tomamos h igual a la mitad de un h anterior, el error debe reducirse por un factor 1/4.

Otras Condiciones de Borde

En esta sección consideramos en el extremo derecho $x = L$ una condición de borde diferente, que contiene a la de tipo Neumann y de tipo Robin.

Consideremos el problema

$$\begin{cases} -K_0 u_{xx} = Q, \quad 0 \leq x \leq L, \\ u(0) = a, \\ K_0 u'(L) + H_1 u(L) = H_2 u_E. \end{cases} \tag{2}$$

Aquí los datos del problema son: K_0, L, $Q(x)$, a, H_1, H_2 y u_E. Notar que la condición de borde que aparece aquí sirve para resolver problemas con *flujo prescripto* y también con *ley de enfriamiento de Newton*.

La diferencia esencial entre este problema y el tratado en la sección anterior radica en la derivada $u'(L)$ que aparece en la condición de borde.

Para no perder precisión en el método, y mantener un error de orden $O(h^2)$, usaremos una diferencia centrada para la derivada:

$$u'(L) \approx \frac{u(L+h) - u(L-h)}{2h}.$$

Notamos que $x_{N+2} = L + h$ es un punto que cae fuera del dominio $[0, L]$ y por eso a x_{N+2} lo llamamos *nodo ficticio*. La condición de borde se reemplaza entonces por la ecuación

$$K_0 \frac{U_{N+2} - U_N}{2h} + H_1 U_{N+1} = H_2 u_E.$$

Esta ecuación reemplaza a la ecuación $U_{N+1} = b$ en el sistema del problema anterior. Observamos que seguimos teniendo $N+1$ ecuaciones, pero ahora tenemos $N+2$ incógnitas, por lo que nos falta una ecuación más. Lo que hacemos es considerar $x_{N+1} = L$ como un *nodo interior*: imponemos la ecuación diferencial $-K_0 u'' = f$ en $x = x_{N+1} = L$. Es decir, agregamos al sistema la ecuación

$$k\frac{-U_N + 2U_{N+1} - U_{N+2}}{h^2} = f_{N+1}.$$

De esta manera, el sistema resulta

$$\begin{cases} U_1 & = a \\ -U_1 + 2U_2 - U_3 & = \dfrac{h^2}{K_0} Q(x_2) \\ \quad - U_2 + 2U_3 - U_4 & = \dfrac{h^2}{K_0} Q(x_3) \\ \qquad \ddots & \vdots \\ \qquad -U_{N-1} + 2U_N - U_{N+1} & = \dfrac{h^2}{K_0} Q(x_N) \\ \qquad\quad -U_N + 2U_{N+1} - U_{N+2} & = \dfrac{h^2}{K_0} Q(x_{N+1}) \\ \qquad\quad -U_N + \dfrac{2hH_1}{K_0} U_{N+1} + U_{N+2} & = \dfrac{2hH_2}{K_0} u_E, \end{cases}$$

De forma matricial se escribe de la siguiente manera:

$$\begin{bmatrix} 1 & 0 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & \frac{2hH_1}{k} & 1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_N \\ U_{N+1} \\ U_{N+2} \end{bmatrix} = \begin{bmatrix} a \\ \frac{h^2}{k} f_2 \\ \frac{h^2}{k} f_3 \\ \vdots \\ \frac{h^2}{k} f_N \\ \frac{h^2}{k} f_{N+1} \\ \frac{2hH_2}{k} u_E \end{bmatrix}.$$

La implementación computacional consiste en escribir un programa que ensamble el sistema y lo resuelva.

También para este problema se tiene un resultado de estimación del error, cuya demostración escapa al alcance de este curso:

Teorema 55. *Si la solución exacta $u(x)$ de* (2) *es $C^4[0, L]$ entonces*

$$\max_{i=1,2,\dots,N+1} |u(x_i) - U_i| \leq CM_4h^2 = CM_4L^2\frac{1}{N^2},$$

donde $M_4 = \max_{[0,L]} |u^{IV}|$ y C es una constante que depende de los parámetros k, L, H_1, H_2 de la ecuación, pero es independiente de la función $f(x)$, del dato u_E y del parámetro de discretización h.

Problema No Estacionario: (Difusión no estacionaria)

Consideremos ahora el problema de difusión no estacionaria siguiente

$$\begin{cases} c\rho u_t - K_0 u_{xx} = Q(x,t), & 0 \leq x \leq L, \quad 0 \leq t \leq T, \\ u(0,t) = a(t), & 0 \leq t \leq T, \\ K_0 u_x(L,t) + H_1 u(L,t) = H_2 u_E(t), & 0 \leq t \leq T, \\ u(x,0) = u_0(x), & 0 \leq x \leq L. \end{cases}$$

Aquí los datos del problema son: c, ρ, K_0, L, T, $Q(x,t)$, $a(t)$, H_1, H_2, $u_E(t)$ y $u_0(x)$. Definiendo $k = \frac{K_0}{c\rho}$ y $q(x,t) = \frac{Q(x,t)}{c\rho}$ resulta

$$\begin{cases} u_t - k u_{xx} = q(x,t), & 0 \leq x \leq L, \quad 0 \leq t \leq T, \\ u(0,t) = a(t), & 0 \leq t \leq T, \\ K_0 u_x(L,t) + H_1 u(L,t) = H_2 u_E(t), & 0 \leq t \leq T, \\ u(x,0) = u_0(x), & 0 \leq x \leq L. \end{cases} \tag{3}$$

Para resolver este problema consideramos la grilla de puntos (x_i, t_j) con $x_i = (i-1)h$ como antes y $t_j = j\Delta t$, donde Δt es el *parámetro de discretización temporal.*

Método explícito

Si aproximamos u_t con una diferencia *adelantada*

$$u_t(x,t) \approx \frac{u(x,t+\Delta t) - u(x,t)}{\Delta t},$$

y u_{xx} como antes

$$u_{xx}(x,t) \approx \frac{u(x-h,t) - 2u(x,t) + u(x+h,t)}{h^2},$$

obtenemos que en cada punto del dominio se cumple

$$\frac{u(x,t+\Delta t) - u(x,t)}{\Delta t} - k\frac{u(x-h,t) - 2u(x,t) + u(x+h,t)}{h^2} \approx q(x,t).$$

Si nos centramos en $(x,t) = (x_i, t_j)$, observando que $x_i \pm h = x_{i\pm 1}$, $t_j + \Delta t = t_{j+1}$, tenemos que

$$\frac{u(x_i,t_{j+1}) - u(x_i,t_j)}{\Delta t} - k\frac{u(x_{i-1},t_j) - 2u(x_i,t_j) + u(x_{i+1},t_j)}{h^2} \approx q(x_i,t_j).$$

Definimos las aproximaciones U_i^j de $u(x_i,t_j)$ de la siguiente manera:

$$\frac{U_i^{j+1} - U_i^j}{\Delta t} - k\frac{U_{i-1}^j - 2U_i^j + U_{i+1}^j}{h^2} = f_i^j,$$

donde denotamos $q_i^j = q(x_i, t_j)$. Observamos que solo hay un supra-índice $j+1$ en la fórmula anterior. Si despejamos el término U_i^{j+1} que lo contiene obtenemos, llamando $\lambda = \frac{k\Delta t}{h^2}$, la siguiente fórmula:

$$U_i^{j+1} = \Delta t q_i^j + \left[\lambda U_{i-1}^j + (1-2\lambda)U_i^j + \lambda U_{i+1}^j\right].$$

Esta fórmula será útil para calcular U en el interior de $[0, L]$, para $t > 0$.

En $t = 0$ ($j = 0$) usaremos la condición inicial $u(x, 0) = u_0(x)$, por lo que tomaremos $U_i^0 = u_0(x_i)$, $i = 1, 2, \ldots, N+1$.

En $x = 0$ usaremos la condición de borde $u(0, t) = a(t)$ por lo que tomaremos $U_1^j = a(t_j)$.

En $x = L$ tenemos la condición de borde $K_0 u_x(L, t) + H_1 u(L, t) = H_2 u_E(t)$. Aproximando $u_x(L, t) \approx \frac{u(L+h,t)-u(L-h,t)}{h}$ llegamos a la fórmula

$$K_0 \frac{U_{N+2}^j - U_N^j}{2h} + H_1 U_{N+1}^j = H_2 u_E(t_j) \quad \rightsquigarrow \quad U_{N+2}^j = \frac{2h}{K_0}(H_2 u_E(t_j) - H_1 U_{N+1}^j) + U_N^j.$$

A partir de estas fórmula, diseñamos el siguiente método numérico para la ecuación de difusión (3) Sean $N, M \in \mathbb{N}$ y definamos $h = L/N$, $\Delta t = T/M$, $\lambda = k\Delta t/h^2$

- $(j=0)\ U_i^0 = u_0(x_i)$, $i = 1, 2, \ldots, N+1$;
- $U_{N+2}^0 = (H_2 u_E(0) - H_1 U_{N+1}^0)\frac{2h}{K_0} + U_N^0$;
- Para $j = 0, 2, \ldots M-1$
  - $U_1^{j+1} = a(t_{j+1})$;
  - $U_i^{j+1} = \Delta t\, q_i^j + \left[\lambda U_{i-1}^j + (1-2\lambda)U_i^j + \lambda U_{i+1}^j\right]$, $i = 2, \ldots, N+1$;
  - $U_{N+2}^{j+1} = \frac{2h}{k}(H_2 u_E(t_{j+1}) - H_1 U_{N+1}^{j+1}) + U_N^{j+1}$;

Este método tiene una restricción para resultar *estable*. Si observamos el término que multiplica U_i^j en la fórmula para U_i^{j+1} vemos que es $1-2\lambda$ con $\lambda = k\Delta t/h^2$. Resulta que si $\lambda \geq 1/2$, de manera que $1-2\lambda \leq 0$ se observa un fenómeno de *inestabilidad* (verificarlo con el código) que hace que las soluciones oscilen en cada paso de tiempo. Lo que ocurre es que pequeños errores de redondeo se magnifican exponencialmente.

Se recomienda, una vez elegido h, elegir Δt de manera que $\lambda \leq 1/4$, para que el algoritmo funcione correctamente. Más aún, en el código que mostramos, elegimos λ y luego Δt se calcula para que $\lambda = \frac{k\Delta t}{h^2}$, es decir $\Delta t = \lambda\frac{h^2}{k}$.

También hay estimaciones del error, que resumimos en el siguiente Teorema.

Teorema 56. *Si la solución exacta $u(x,t)$ de* (3) *es $C^4[0,L]$ para la variable x y $C^2[0,T]$ para la variable T, entonces, si $\lambda = k\Delta t/h^2 < 1/2$, resulta*

$$\max_{\substack{i=1,2,\ldots,N+1\\ j=1,2,\ldots,M}} |u(x_i,t_j) - U_i^j| \leq C(h^2 + \Delta t)M_{4,2},$$

donde $M_{4,2} = \max_{[0,L]\times[0,T]} |u_{xxxx}| + |u_{tt}|$ y C es una constante que depende de los parámetros c, ρ, K_0, L, H_1, H_2 de la ecuación, pero es independiente de la función $f(x,t)$, del dato inicial $u_0(x)$, de los datos de borde $a(t)$, $u_E(t)$ y de los parámetros de discretización h, Δt.

Método implícito

Cuando los coeficientes son variables, o cuando queremos resolver problemas en dos o tres dimensiones espaciales, la condición de estabilidad puede ser más difícil de determinar. En esta sección veremos un método que es *incondicionalmente estable*, es decir, resulta estable sin importar la relación entre Δt y h.

La diferencia principal con el método obtenido anteriormente está en aproximar la derivada temporal u_t por una diferencia *atrasada* en lugar de *adelantada*. Es decir, consideraremos que

$$\frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} \approx u_t(x_i, t_{j+1}).$$

Por lo tanto, vemos que

$$\frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} - k\frac{u(x_{i-1}, t_{j+1}) - 2u(x_i, t_{j+1}) + u(x_{i+1}, t_{j+1})}{h^2} \approx q(x_i, t_{j+1}).$$

Definimos entonces las aproximaciones U_i^j de $u(x_i, t_j)$ de la siguiente manera:

$$\frac{U_i^{j+1} - U_i^j}{\Delta t} - k\frac{U_{i-1}^{j+1} - 2U_i^{j+1} + U_{i+1}^{j+1}}{h^2} = q_i^{j+1}.$$

Si ahora *despejamos* las incógnitas con supra-índice $j + 1$ a la izquierda, nos queda

$$-\lambda U_{i-1}^{j+1} + (1 + 2\lambda)U_i^{j+1} - \lambda U_{i+1}^{j+1} = \Delta t\, q_i^{j+1} + U_i^j.$$

Vemos que no queda U_i^{j+1} *despejada* sola a la izquierda, sino que queda relacionada con U_{i-1}^{j+1} y con U_{i+1}^{j+1}. Por eso se dice que el método es *implícito*.

Por lo tanto, en cada paso de tiempo deberemos resolver un *sistema de ecuaciones*. Esto es un poco más costoso computacionalmente que lo que se hace en el caso *explícito*, pero se gana en estabilidad.

El método implícito resulta estable para cualquier $\Delta t > 0$.

Si incorporamos las condiciones de borde, vemos que, conocido U_i^j, para un j dado y para $i = 1, 2, \ldots, N+2$, los valores U_i^{j+1}, $i = 1, 2, \ldots, N+2$ deben satisfacer el siguiente sistema de $N+2$ ecuaciones

$$\begin{cases} U_1^{j+1} = a(t_{j+1}), \\ -\lambda U_{i-1}^{j+1} + (1+2\lambda)U_i^{j+1} - \lambda U_{i+1}^{j+1} = \Delta t\, q_i^{j+1} + U_i^j, \qquad i = 2, 3, \ldots, N+1, \\ -U_N^{j+1} + \dfrac{2hH_1}{K_0} U_{N+1}^{j+1} + U_{N+2}^{j+1} = \dfrac{2hH_2}{K_0} u_E(t_{j+1}). \end{cases} \tag{4}$$

En forma matricial, el sistema a resolver en cada paso de tiempo resulta:

$$\begin{bmatrix} 1 & 0 & 0 & \dots & \dots & 0 \\ -\lambda & 1+2\lambda & -\lambda & 0 & \dots & 0 \\ 0 & -\lambda & 1+2\lambda & -\lambda & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -\lambda & 1+2\lambda & -\lambda \\ 0 & \dots & 0 & -1 & \frac{2hH_1}{k} & 1 \end{bmatrix} \begin{bmatrix} U_1^{j+1} \\ U_2^{j+1} \\ U_3^{j+1} \\ \vdots \\ U_N^{j+1} \\ U_{N+1}^{j+1} \\ U_{N+2}^{j+1} \end{bmatrix} = \begin{bmatrix} a(t_{j+1}) \\ U_2^j \\ U_3^j \\ \vdots \\ U_N^j \\ U_{N+1}^j \\ \frac{2hH_2}{k} u_E \end{bmatrix} + \Delta t \begin{bmatrix} 0 \\ f_2^{j+1} \\ f_3^{j+1} \\ \vdots \\ f_N^{j+1} \\ f_{N+1}^{j+1} \\ 0 \end{bmatrix}.$$

Esto debe repetirse para $j = 0, 1, 2, \dots, M$ luego de haber definido

$$U_i^0 = u_0(x_i), \quad i = 1, 2, \dots, N+1 \qquad \text{y} \qquad U_{N+2}^0 = (H_2 u_E(0) - H_1 U_{N+1}^0)\frac{2h}{k} + U_N^0.$$

También hay estimaciones del error, que resumimos en el siguiente Teorema.

Teorema 57. *Si la solución exacta $u(x,t)$ de* (3) *es $C^4[0,L]$ para la variable x y $C^2[0,T]$ para la variable T, entonces resulta*

$$\max_{\substack{i=1,2,\dots,N+1\\ j=1,2,\dots,M}} |u(x_i,t_j) - U_i^j| \leq C(h^2 + \Delta t)M,$$

donde $M = \max_{[0,L]\times[0,T]} |u_{xxxx}| + |u_{tt}|$ y C es una constante que depende de los parámetros c, ρ, K_0, L, H_1, H_2 de la ecuación, pero es independiente de la función $Q(x,t)$, del dato inicial $u_0(x)$, de los datos de borde $a(t)$, $u_E(t)$ y de los parámetros de discretización h, Δt.