Diferencias Finitas

Problemas No estacionarios

Modelación computacional en las ciencias y las ingenierías como apoyo en el proceso enseñanza-aprendizaje (PAPIME-PE101019)

Instituto de Geofísica
Universidad Nacional Autónoma de México

Contenido

MODELO

MATEMÁTICO

Ecuación "general" de transferencia de calor (véase [4]):

$$\boxed{c_p \rho \frac{\partial T}{\partial t} + c_p \rho \frac{\partial}{\partial x_j}(u_j T) - \frac{\partial}{\partial x_j}\left(\kappa \frac{\partial T}{\partial x_j}\right) = S}$$

(índices repetidos se suman)

donde se define lo siguiente:

Símbolo		**Unidades**
	Parámetros físicos	
c_p	Capacidad calorífica específica.	[J / Kg ^{o}K]
ρ	Densidad.	[Kg / m^3]
κ	Conductividad térmica.	[W / m ^{o}K]
S	Ganancia (fuente) o pérdida (sumidero) de calor	[J/m^3 s]
$\alpha = \dfrac{\kappa}{c_p \rho}$	Difusividad térmica.	[m^2/s]
	Variables independientes	
x_j	Coordenadas cartesianas: $(x_1, x_2, x_3) \equiv (x, y, z)$.	[m]
t	Tiempo.	[s]
	Variables dependientes	
T	Temperatura.	[^{o}K]
u_j	Componentes de la velocidad: $(u_1, u_2, u_3) \equiv (u_x, u_y, u_z)$.	[m/s]

Dado que estamos interesados en la conducción de calor **NO estacionaria** eliminamos el término de advección de la ecuación general:

$$c_p\rho\frac{\partial T}{\partial t} + \cancel{c_p\rho\frac{\partial}{\partial x_j}(u_jT)} - \frac{\partial}{\partial x_j}\left(\kappa\frac{\partial T}{\partial x_j}\right) = S$$

Entonces, el modelo matemático para este problema en 1D, con $c_p\rho = constante$, es:

$$\boxed{\begin{aligned} \frac{\partial T}{\partial t} - \frac{\partial}{\partial x}\left(\alpha\frac{\partial T}{\partial x}\right) &= Q \\ T(x=0) &= T_A \\ T(x=L) &= T_B \end{aligned}} \tag{1}$$

donde $Q = \frac{S}{c_p\rho}$. Obsérvese que se tienen condiciones de tipo Dirichlet: la variable dependiente, T, está dada en las fronteras. Estas condiciones también se conocen como de primer tipo.

Problema de Valor Inicial

Problema de Valor Inicial

La ecuación (1) se puede escribir en términos de un PVI:

Encontrar $u(x,t)$ que cumpla:

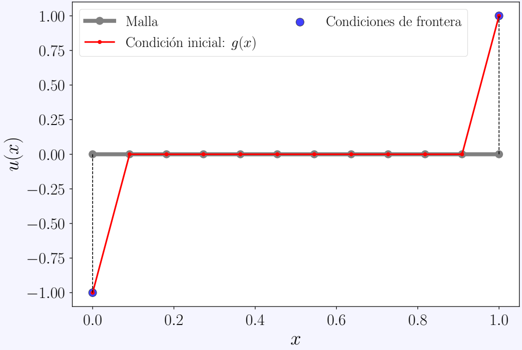
$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\alpha \frac{\partial u}{\partial x}\right) + Q, \quad \text{para} \quad a \leq x \leq b, \quad \text{y} \quad 0 \leq t \leq T_{max}. \tag{2}$$

Condiciones iniciales

$$u(x,0) = g(x)$$

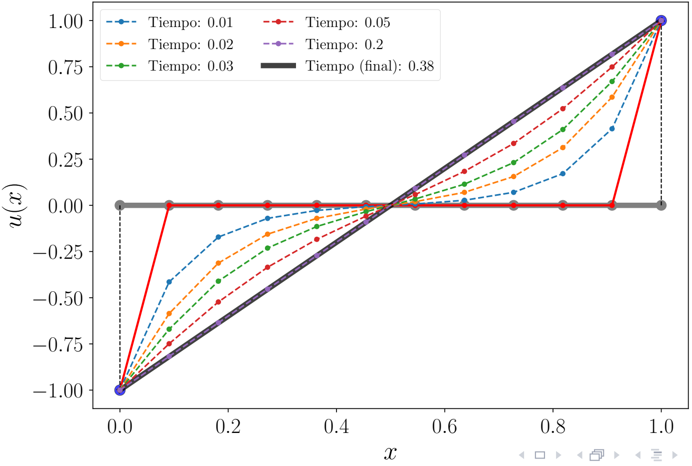
Condiciones de frontera

$$\begin{aligned} u(a,t) &= A(t) \\ u(b,t) &= B(t) \end{aligned}$$



Problema de Valor Inicial

Resolver el problema de valor inicial descrito antes, implica encontrar soluciones en cada paso de tiempo:



Contenido

MODELO

NUMÉRICO

Problema no estacionario

Cuando $\alpha = constante$, la ecuación (1) se transforma en:

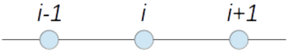
$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} + Q, \quad \text{para} \quad a \leq x \leq b, \quad \text{y} \quad 0 \leq t \leq T_{max}.$$

Definimos lo siguiente:

- N_x es el número de incógnitas en x,
- N_t es el número de pasos en el tiempo,
- $\Delta x = \dfrac{b-a}{N_x+1} = h$ es el tamaño de la malla
- $\Delta t = \dfrac{T_{max}}{N_t} = h_t$, es el paso de tiempo.

Para cada punto i de la malla espacial podemos aproximar:

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_i \approx \frac{(u_{i+1} - 2u_i + u_{i-1})}{h^2}$$



Problema no estacionario: semidiscretización

Entonces, el problema (2) se transforma en N_x PVI's, uno por cada punto de la malla espacial:

$$\begin{aligned} \left.\frac{\partial u}{\partial t}\right|_i &= f(t, u_{i+1}, u_i, u_{i-1}, Q_i), \quad 0 \leq t \leq T_{max} \\ u(x_i, 0) &= g(x_i) \equiv g_i \end{aligned} \tag{3}$$

para $i = 1, \ldots, N_x$.

Tenemos entonces una ecuación del tipo (3) para cada i, donde:

$$f(t, u_{i+1}, u_i, u_{i-1}, Q_i) = \frac{\alpha}{h^2}(u_{i+1} - 2u_i + u_{i-1}) + Q_i \tag{4}$$

Observe que para los nodos en los extremos, se deben aplicar las condiciones de frontera correspondientes.

Método de Euler hacia adelante

Problema no estacionario: Euler hacia adelante (explícito)

La ecuación

$$\left.\frac{\partial u}{\partial t}\right|_i = f(t, u_{i+1}, u_i, u_{i-1}, Q_i) \tag{5}$$

para $0 \leq t \leq T_{max}$, $i = 1, \ldots, N_x$, con condición inicial $u(x_i, 0) = g(x_i) \equiv g_i$ se puede resolver aproximando la derivada temporal usando **diferencias finitas hacia adelante**:

$$\left.\frac{\partial u}{\partial t}\right|_i \approx \frac{u_i^{n+1} - u_i^n}{h_t} \tag{6}$$

donde $u_i^n \equiv u(x_i, t_n)$ y $u_i^{n+1} \equiv u(x_i, t_n + h_t)$.

Sustituyendo (6) en (5) obtenemos:

$$\boxed{u_i^{n+1} = u_i^n + h_t\, f(t_n, u_{i+1}^n, u_i^n, u_{i-1}^n, Q_i^n)} \tag{7}$$

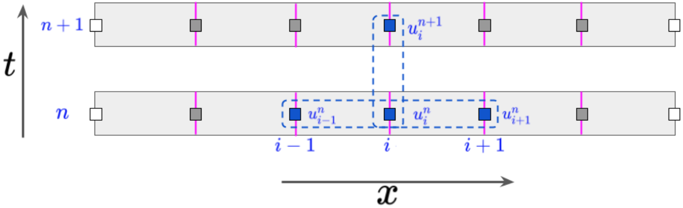
Problema no estacionario: Euler hacia adelante (explícito)

Sustituyendo la ecuación (4) en (7) obtenemos el **Método de Euler hacia adelante** para la ecuación de calor (1):

$$\boxed{u_i^{n+1} = u_i^n + r\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)} \tag{8}$$

donde $r = \dfrac{h_t}{h^2}\alpha$ y hemos supuesto por ahora que $Q_i = 0$ para $i = 1, \dots, N_x$.

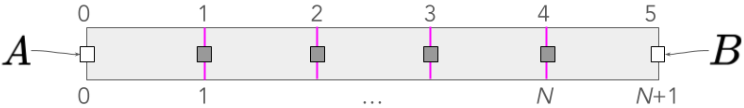
La fórmula (8) proporciona una forma **explícita** para encontrar u_i^{n+1} usando los valores conocidos u_{i-1}^n, u_i^n y u_{i+1}^n, la figura siguiente esquematiza este hecho.



Este es un método **condicionalmente estable**.

Problema no estacionario: Euler hacia adelante (explícito)

Para los nodos 1 y N_x se deben aplicar las condiciones de frontera correspondientes.



Supongamos las siguientes condiciones de frontera:

- Condición Dirichlet en $i = 0$: $u_0^n = A^n$

$$
\begin{aligned}
u_1^{n+1} &= u_1^n + r\,(u_2^n - 2u_1^n + u_0^n) \\
u_1^{n+1} &= u_1^n + r\,(u_2^n - 2u_1^n + A^n) \qquad \text{(eq. para } i = 1)
\end{aligned}
$$

- Condición Neumman en $i = N_x + 1$: $\left.\dfrac{\partial u}{\partial n}\right|_{N_x+1}^{n} \approx \dfrac{u_{N_x+1}^n - u_{N_x}^n}{h} = B^n$

$\Longrightarrow u_{N_x+1}^n = h * B^n + u_{N_x}^n$.

$$
\begin{aligned}
u_{N_x}^{n+1} &= u_{N_x}^n + r\,(u_{N_x+1}^n - 2u_{N_x}^n + u_{N_x-1}^n) \\
u_{N_x}^{n+1} &= u_{N_x}^n + r\,(h * B^n - u_{N_x}^n + u_{N_x-1}^n) \qquad \text{(eq. para } i = N_x)
\end{aligned}
$$

Ejercicio 10: Método de Euler hacia adelante

Encontrar $u(x,t)$ que cumpla

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\alpha \frac{\partial u}{\partial x}\right) + Q, \quad \text{para} \quad a \leq x \leq b, \quad \text{y} \quad 0 \leq t \leq T_{max}.$$

Condición inicial:

$$u(x,0) = 0, \quad \text{para} \quad a < x < b,$$

Condiciones de frontera:

$$\begin{aligned} u(a,t) &= -1.0 \quad \text{para} \quad 0 \leq t \leq T_{max}. \\ u(b,t) &= 1.0 \quad \text{para} \quad 0 \leq t \leq T_{max}. \end{aligned}$$

para los siguientes valores numéricos: $L = 1.0$, $\alpha = 1.0$, $Q = 0.0$, $T_{max} = 1.0$ y $h_t = 0.0001$ ($N_t = 10000$).

1. Abra el notebook `E10_ForwardEuler.ipynb`, del repositorio `Mixbaal` y complete el código de la celda 4 implementando la solución al problema anterior usando el **método de Euler hacia adelante**, use como base el algoritmo 1 (*pag.* 19).
2. Reproduzca la figura 1 (*pag.* 20) en donde se usó $h_t = 0.0001$.
3. Explique el comportamiento del error conforme avanza el tiempo.
4. ¿Qué pasa cuando $h_t = 0.0002$, 0.0003 y 0.0004? Explique el comportamiento.

Ejercicio 10: Método de Euler hacia adelante

Algoritmo 1 Pseudo-código del Método de Euler hacia adelante

for $n = 1$ to N_t **do**
  Inicializar el error para el paso n (`e = 0.0`)
  Iniciar el cronómetro (`t1=time.perf_counter()`)
  for $i = 1$ to N_x **do**
    $u_i^{n+1} = u_i^n + r\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)$
    Calcular: $E_i = (u_i^{n+1} - u_i^n)^2$ y acumularlo (`e +=` E_i)
    Actualizar u_i^n $(u_i^n \leftarrow u_i^{n+1})$
  end for
  Detener el cronómetro (`t2=time.perf_counter()`)
  Sumar el tiempo al total (`suma_tiempos += (t2-t1)`)
  Completar el cálculo del error (`e = np.sqrt(h*e)`)
  Agregar el error a la lista (`error.append(e)`)
  $\ldots$
end for

Observe que en el código, u_{i+1}^n, u_i^n y u_{i-1}^n están representadas por `u[i+1]`, `u[i]` y `u[i-1]` respectivamente. Puede usar la etiqueta `unew` para representar a u_i^{n+1} (no requiere de un arreglo).

Ejercicio 10: Método de Euler hacia adelante

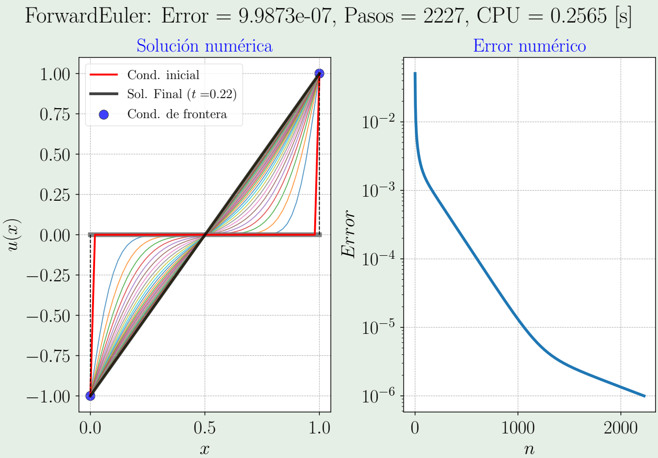


Figura 1. Resultado obtenido con $h_t = 0.0001$ y `tolerancia =` 1×10^{-6}. En la gráfica de la izquierda, las líneas delgadas de colores muestran el avance de la solución, desde la cond. inicial (línea roja) hasta la sol. final (línea negra).

Método de Euler hacia atrás

Problema no estacionario: Euler hacia atrás (implícito)

La ecuación

$$\left.\frac{\partial u}{\partial t}\right|_i = f(t, u_{i+1}, u_i, u_{i-1}, Q_i) \tag{9}$$

para $0 \leq t \leq T_{max}$, $i = 0, \dots, N_x$, con condición inicial $u(x_i, 0) = g(x_i) \equiv g_i$ se puede resolver aproximando la derivada temporal usando **diferencias finitas hacia atrás**:

$$\left.\frac{\partial u}{\partial t}\right|_i \approx \frac{u_i^n - u_i^{n-1}}{h_t} \tag{10}$$

donde $u_i^n \equiv u(x_i, t_n)$ y $u_i^{n-1} \equiv u(x_i, t_n - h_t)$.

Sustituyendo (10) en (9) obtenemos:

$$\boxed{u_i^n = u_i^{n-1} + h_t\, f(t_n, u_{i+1}^n, u_i^n, u_{i-1}^n, Q_i^n)} \tag{11}$$

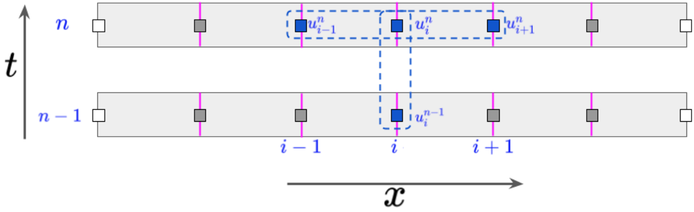
Problema no estacionario: Euler hacia adelante (explícito)

Sustituyendo la ecuación (4) en (11) obtenemos el **Método de Euler hacia atrás** para la ecuación de calor (1):

$$\boxed{u_i^n = u_i^{n-1} + r\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)} \tag{12}$$

donde $r = \dfrac{h_t}{h^2}\alpha$ y hemos supuesto por ahora que $Q_i = 0$ para $i = 1, \ldots, N_x$.

En la fórmula (12) observamos que tenemos tres incógnitas u_i^n, u_{i+1}^n y u_{i-1}^n; y un valor conocido en el tiempo anterior u_i^{n-1}, la figura siguiente esquematiza este hecho.



Por lo tanto la fórmula es **implícita** y se debe resolver un sistema de ecuaciones. Este método es **incondicionalmente estable**.

Problema no estacionario: Euler hacia atrás (implícito)

Este método genera el siguiente sistema de ecuaciones

$$
\begin{array}{rcl}
u_1^n & = & u_1^{n-1} + r\left(u_2^n - 2u_1^n + u_0^n\right) \\
u_2^n & = & u_2^{n-1} + r\left(u_3^n - 2u_2^n + u_1^n\right) \\
\ldots & \ldots & \ldots \\
u_i^n & = & u_i^{n-1} + r\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \\
\ldots & \ldots & \ldots \\
u_{N_x}^n & = & u_{N_x}^{n-1} + r\left(u_{N_x+1}^n - 2u_{N_x}^n + u_{N_x-1}^n\right)
\end{array}
$$

Considerando una condición de tipo Dirichlet en $i = 0$ y una Neumman en $i = N_x$ tenemos:

$$
u_1^n = u_1^{n-1} + r\left(u_2^n - 2u_1^n + A^n\right) \quad \text{(Dirichlet)}
$$

$$
u_{N_x}^n = u_{N_x}^{n-1} + r\left(hB^n - u_{N_x}^n + u_{N_x-1}^n\right) \quad \text{(Neumman)}
$$

Problema no estacionario: Euler hacia atrás (implícito)

Escribimos el sistema de ecuaciones anterior como sigue:

$$
\begin{aligned}
-ru_2^n + (1+2r)u_1^n &= u_1^{n-1} + rA^n \\
-ru_3^n + (1+2r)u_2^n - ru_1^n &= u_2^{n-1} \\
\cdots \quad \cdots &\quad \cdots \\
-ru_{i+1}^n + (1+2r)u_i^n - ru_{i-1}^n &= u_i^{n-1} \\
\cdots \quad \cdots &\quad \cdots \\
(1+r)u_{N_x}^n - ru_{N_x-1}^n &= u_{N_x}^{n-1} + rhB^n
\end{aligned}
$$

En forma matricial:

$$
\begin{bmatrix}
(1+2r) & -r & 0 & 0 & 0 & 0 \\
-r & (1+2r) & -r & 0 & 0 & 0 \\
0 & -r & (1+2r) & -r & 0 & 0 \\
0 & 0 & -r & (1+2r) & -r & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & -r & (1+r)
\end{bmatrix}
\begin{bmatrix}
u_1^n \\ u_2^n \\ u_3^n \\ u_4^n \\ \vdots \\ u_{N_x}^n
\end{bmatrix}
=
\begin{bmatrix}
u_1^{n-1} + rA^n \\ u_2^{n-1} \\ u_3^{n-1} \\ u_4^{n-1} \\ \vdots \\ u_{N_x}^{n-1} + rhB^n
\end{bmatrix}
$$

Ejercicio 11: Método de Euler hacia atrás

Encontrar $u(x,t)$ que cumpla

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\alpha \frac{\partial u}{\partial x}\right) + Q, \quad \text{para} \quad a \leq x \leq b, \quad \text{y} \quad 0 \leq t \leq T_{max}.$$

Condición inicial:

$$u(x,0) = 0, \quad \text{para} \quad a < x < b,$$

Condiciones de frontera:

$$\begin{aligned} u(a,t) &= -1.0 \quad \text{para} \quad 0 \leq t \leq T_{max}. \\ u(b,t) &= 1.0 \quad \text{para} \quad 0 \leq t \leq T_{max}. \end{aligned}$$

para los siguientes valores numéricos: $L = 1.0$, $\alpha = 1.0$, $Q = 0.0$, $T_{max} = 1.0$ y $h_t = 0.0001$ ($N_t = 10000$).

1. Abra el notebook `E11_BackwardEuler.ipynb`, del repositorio `Mixbaal` y complete el código de la celda 5 implementando la solución al problema anterior usando el **método de Euler hacia atrás**, use como base el algoritmo 2 (*pag.* 27).
2. Reproduzca la figura 2 (*pag.* 28) en donde se usó $h_t = 0.0001$.
3. Explique el comportamiento del error conforme avanza el tiempo.
4. ¿Se obtiene un buen resultado con $h_t = 0.1$ y $h_t = 0.5$? ¿Por qué?

Ejercicio 11: Método de Euler hacia atrás

Algoritmo 2 Pseudo-código del Método de Euler hacia atrás

for $n = 1$ to N_t **do**
    Iniciar el cronómetro (`t1 = time.perf_counter()`)
    Aplicar las cond de frontera (`f[0] += r * bA`; `f[N-1] += r * bB`)
    Resolver el sistema lineal (`u[1:N+1] = np.linalg.solve(A,f)`)
    Detener el cronómetro (`t2 = time.perf_counter()`)
    Sumar el tiempo al total (`suma_tiempos += (t2-t1)`)
    Calcular el error (`e = np.sqrt(h) * np.linalg.norm(uold-u)`)
    Agregar el error a la lista (`error.append(e)`)
    $\dots$
end for

Observe que en el código, u^n está representada por el arreglo `u`. En este caso necesitamos un arreglo adicional para almacenar la solución en el paso previo u^{n-1}, este arreglo es `uold`.

Ejercicio 11: Método de Euler hacia atrás

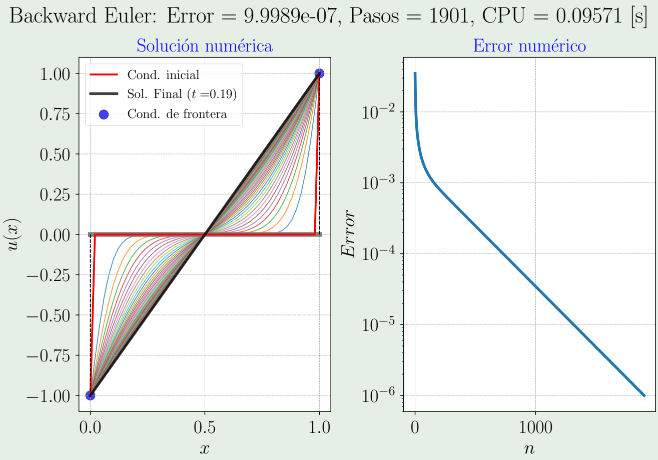


Figura 2. Resultado obtenido con $h_t = 0.0001$ y `tolerancia =` 1×10^{-6}. En la gráfica de la izquierda, las líneas delgadas de colores muestran el avance de la solución, desde la cond. inicial (línea roja) hasta la sol. final (línea negra).

Convergencia, consistencia y estabilidad

Estabilidad: Euler Forward vs. Euler Backward

Euler Forward (Explícito)

- La evaluación de u_i^{n+1} se realiza mediante la evaluación de una fórmula explícita.
- Cada una de estas evaluaciones es independiente de las otras, por lo que este esquema es paralelizable directamente.
- Es condicionalmente estable: $\frac{h_t}{h^2}\alpha < \frac{1}{2} \implies h_t < \frac{h^2}{2\alpha}$.
- La condición anterior obliga a realizar muchos pasos de tiempo para llegar a T_{max}.

Euler Backward (Implícito)

- La solución en el paso $n+1$ se encuentra resolviendo un sistema lineal de ecuaciones.
- Es posible paralelizar la solución del sistema lineal.
- Es incondicionalmente estable, por lo que h_t puede ser grande.
- Si la matriz del sistema está mal condicionada, la solución de un paso de tiempo puede tardar mucho.

- En ambos casos el orden de la aproximación es $\mathcal{O}(h^2) + \mathcal{O}(h_t)$.

Convergencia, consistencia y estabilidad

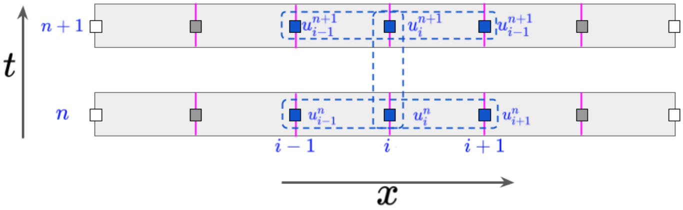
- **Convergencia** : Es la propiedad de un método numérico de producir una solución que aproxima a la solución exacta conforme la distancia entre los puntos de la malla tiende a cero.
- **Consistencia** : Un esquema numérico consistente produce sistemas de ecuaciones algebraicas equivalentes a las ecuaciones gobernantes originales cuando el espaciamiento de los nodos de la malla tiende a cero.
- **Estabilidad** : La estabilidad está asociada con la amortiguación del error conforme el método numérico avanza.

Teorema de equivalencia de Lax

Para problemas lineales, una condición necesaria y suficiente para obtener convergencia es que el método sea consistente y estable.

Esquema Crank-Nicholson

Problema no estacionario: Crank-Nicholson (implícito)



Esquema de orden $\mathcal{O}(h^2 + h_t^2)$. Aproximamos en $(x_i, t^{n+\frac{1}{2}})$:

$$
\begin{aligned}
\left.\frac{\partial u}{\partial t}\right|_i^{n+\frac{1}{2}} &= \frac{u_i^{n+1} - u_i^n}{h_t} \\
\left.\frac{\partial^2 u}{\partial x^2}\right|_i^{n+\frac{1}{2}} &= \frac{\kappa}{2}\left[\frac{(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1})}{h^2} + \frac{(u_{i+1}^{n} - 2u_i^{n} + u_{i-1}^{n})}{h^2}\right]
\end{aligned}
$$

Método de Crank-Nicholson:

$$
\boxed{u_i^{n+1} - \frac{r}{2}\left(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right) = u_i^n + \frac{r}{2}\left(u_{i+1}^{n} - 2u_i^{n} + u_{i-1}^{n}\right)} \tag{13}
$$

La fórmula (13) es implícita y condicionalmente estable.

Contenido

- [1] R.J. Leveque,
  Finite Difference Method for Ordinary and Partial Differential Equations: Steady State and Time-Dependent Problems ,
  Society for Industrial and Applied Mathematics (SIAM), Philadelphia, **2007**.
- [2] Y. Saad
  Iterative Methods for Sparse Linear Systems.
  PWS/ITP 1996.
  Online: http://www-users.cs.umn.edu/~saad/books.html, **2000**
- [3] Richard Burden and J. Douglas Faires
  Numerical Analysis
  Cengage Learning; 9 edition (August 9, **2010**)
- [4] I. Herrera & G. F. Pinder,
  Mathematical Modeling in Science and Engineering: An Axiomatic Approach,
  John Wiley **2012**.

Contenido

Créditos

Dr. Luis M. de la Cruz Salas

Departamento de Recursos Naturales

Instituto de Geofísica

Universidad Nacional Autónoma de México

Trabajo realizado con el apoyo del Programa UNAM-DGAPA-PAPIME PE101019