Métodos Computacionales - Trabajo Práctico 1

Resoluciones Numéricas de Ecuaciones Diferenciales

Mariño Martina, Martinez Kiara
Universidad Torcuato Di Tella

24 de septiembre de 2025

Índice

Parte I

Ecuación del Calor

1. Formulación matemática - Derivación de los métodos explícito e implícito

En esta sección vamos a ver cómo se llega a las fórmulas de los métodos explícito e implícito para resolver la ecuación del calor usando diferencias finitas. La idea es empezar de la ecuación original, discretizarla, y después aproximar las derivadas.

1.1. La ecuación que queremos resolver

La ecuación del calor en una dimensión es:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \quad x \in (0,1), \quad t > 0$$

donde:

- $u(x,t)$ representa la temperatura en el punto x y tiempo t.
- $\alpha > 0$ es la constante de difusión térmica.

Además, tenemos condiciones de frontera de Dirichlet:

$$u(0,t) = 0, \quad u(1,t) = 0,$$

y una condición inicial que nos da la distribución de temperatura al inicio:

$$u(x,0) = f(x).$$

En palabras, se trata de una barra de longitud 1 donde los extremos se mantienen a temperatura cero. Sabemos cómo estaba la temperatura al comienzo y queremos ver cómo cambia con el tiempo.

1.2. Discretización: malla de espacio y tiempo

Para resolver el problema de forma numérica dividimos el espacio y el tiempo en puntos separados por pasos fijos:

- Dividimos el intervalo espacial $[0,1]$ en N puntos con separación Δx. Así, las posiciones son:

  $$x_j = j\Delta x, \quad j = 0, 1, \ldots, N$$

  donde $\Delta x = \frac{1}{N}$.
- El tiempo se divide en pasos de tamaño Δt. Los instantes de tiempo quedan como:

  $$t_n = n\Delta t, \quad n = 0, 1, 2, \ldots, M$$

  con $M\Delta t = T$ siendo el tiempo final de simulación.

Notación: llamamos u_j^n a la aproximación numérica de $u(x_j, t_n)$. Los nodos de frontera son $j = 0$ y $j = N$ (donde ya conocemos u gracias a las condiciones de borde). Los nodos internos son $j = 1, \ldots, N-1$, que son los que vamos a actualizar.

1.3. Aproximación de las derivadas con diferencias finitas

Reemplazamos las derivadas por aproximaciones usando diferencias finitas:

- Para la derivada temporal usamos una **diferencia hacia adelante**:

$$\frac{\partial u}{\partial t}(x_j, t_n) \approx \frac{u_j^{n+1} - u_j^n}{\Delta t}.$$

- Para la segunda derivada espacial usamos **diferencias centradas**:

$$\frac{\partial^2 u}{\partial x^2}(x_j, t_n) \approx \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}.$$

Estas aproximaciones introducen un error, pero mientras Δx y Δt sean pequeños, la aproximación es bastante buena.

1.4. Método explícito

Partimos de la ecuación del calor discretizada:

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

Definición de r:

Definimos un parámetro muy útil:

$$r = \frac{\alpha \Delta t}{(\Delta x)^2}.$$

Este parámetro combina la constante física α con el paso de tiempo Δt y el paso espacial Δx.

Paso 1. Sustituir r

$$u_j^{n+1} - u_j^n = r\big(u_{j+1}^n - 2u_j^n + u_{j-1}^n\big).$$

Paso 2. Aislar u_j^{n+1}

$$u_j^{n+1} = u_j^n + r\big(u_{j+1}^n - 2u_j^n + u_{j-1}^n\big).$$

Paso 3. Expandir el paréntesis

$$u_j^{n+1} = u_j^n + r\,u_{j+1}^n - 2r\,u_j^n + r\,u_{j-1}^n.$$

Paso 4. Agrupar términos

$$u_j^{n+1} = (1 - 2r)u_j^n + r\,u_{j-1}^n + r\,u_{j+1}^n.$$

El nuevo valor de temperatura en j depende del valor previo en el mismo punto y de sus dos vecinos. Es una combinación lineal:

- $(1 - 2r)u_j^n$ (el valor previo en j),
- $r\,u_{j-1}^n$ (vecino izquierdo),
- $r\,u_{j+1}^n$ (vecino derecho).

Esquema explícito

$$\boxed{u_j^{n+1} = (1-2r)u_j^n + r\,u_{j-1}^n + r\,u_{j+1}^n}$$

Condición de estabilidad:

Este método solo funciona bien si se cumple:

$$r = \frac{\alpha\Delta t}{(\Delta x)^2} \leq \frac{1}{2}.$$

De lo contrario, la solución numérica se vuelve inestable, con oscilaciones crecientes que no reflejan el fenómeno físico real.

Interpretación:

La temperatura en j tiende a "suavizarse" dependiendo de cómo están los vecinos $j-1$ y $j+1$. Es como un promedio ponderado que se va corrigiendo paso a paso.

1.5. Método implícito

Partimos de la misma aproximación temporal, pero evaluando la derivada espacial en $n+1$:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} \approx \alpha\,\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\Delta x)^2}.$$

Paso 1. Multiplicar por Δt

$$u_j^{n+1} - u_j^n = \alpha\,\frac{\Delta t}{(\Delta x)^2}\big(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}\big).$$

Paso 2. Definir r y sustituir

$$r := \frac{\alpha\,\Delta t}{(\Delta x)^2},$$

$$u_j^{n+1} - u_j^n = r\big(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}\big).$$

Paso 3. Reorganizar términos

$$u_j^{n+1} - r\big(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}\big) = u_j^n.$$

Paso 4. Expandir el paréntesis

$$u_j^{n+1} - r\,u_{j+1}^{n+1} + 2r\,u_j^{n+1} - r\,u_{j-1}^{n+1} = u_j^n.$$

Paso 5. Agrupar coeficientes

$$-r\,u_{j-1}^{n+1} + (1+2r)\,u_j^{n+1} - r\,u_{j+1}^{n+1} = u_j^n.$$

Esquema implícito

$$\boxed{-r\,u_{j-1}^{n+1} + (1+2r)\,u_j^{n+1} - r\,u_{j+1}^{n+1} = u_j^n}$$

Interpretación:

En este caso cada valor u_j^{n+1} está acoplado con sus vecinos en el mismo nivel temporal $n+1$. Por eso, no hay una fórmula directa: es necesario resolver un sistema lineal en cada paso. La gran ventaja es que el método es **incondicionalmente estable**, sin restricciones sobre r.

1.6. Resumen comparativo

- **Explícito:** sencillo de implementar, no requiere resolver sistemas, pero exige la condición $r \leq \frac{1}{2}$.
- **Implícito:** más costoso porque requiere resolver un sistema en cada paso, pero estable para cualquier Δt y Δx.

2. Sistemas lineales y forma matricial

2.1. Recordemos el dominio

Tenemos $N+1$ puntos en el espacio, numerados de $j=0$ a $j=N$. Los extremos ($j=0$ y $j=N$) son conocidos gracias a las condiciones de frontera $u(0,t)=u(1,t)=0$. Los valores desconocidos corresponden a los nodos internos $j=1,2,\ldots,N-1$.
Esto significa que en cada instante de tiempo debemos trabajar con un vector de tamaño $N-1$, que contiene únicamente los valores de los nodos internos.

2.2. Método explícito como sistema lineal

El método explícito en su forma final es:

$$u_j^{n+1} = (1-2r)u_j^n + r\,u_{j-1}^n + r\,u_{j+1}^n, \quad j = 1,2,\ldots,N-1.$$

Forma matricial:

Podemos escribir esta relación como:

$$U^{n+1} = B\,U^n,$$

donde:

$$U^n = \begin{bmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{N-2}^n \\ u_{N-1}^n \end{bmatrix}$$

es el vector con los valores de los nodos internos en el tiempo t_n.

La matriz B tiene estructura tridiagonal:

$$B = \begin{bmatrix} 1-2r & r & 0 & \cdots & 0 \\ r & 1-2r & r & \cdots & 0 \\ 0 & r & 1-2r & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & r \\ 0 & 0 & 0 & r & 1-2r \end{bmatrix}.$$

Interpretación:

Para pasar del tiempo t_n al tiempo t_{n+1}, basta con multiplicar el vector U^n por la matriz B. El método explícito se reduce, por lo tanto, a una simple multiplicación matricial.

2.3. Método implícito como sistema lineal

En el método implícito, la ecuación para cada nodo es:

$$-r\, u_{j-1}^{n+1} + (1+2r)\, u_j^{n+1} - r\, u_{j+1}^{n+1} = u_j^n, \quad j = 1, 2, \ldots, N-1.$$

Forma matricial:

Esto se puede escribir como un sistema lineal:

$$A\, U^{n+1} = U^n,$$

donde U^{n+1} es el vector de incógnitas en el nuevo tiempo:

$$U^{n+1} = \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ \vdots \\ u_{N-2}^{n+1} \\ u_{N-1}^{n+1} \end{bmatrix}.$$

La matriz A también es tridiagonal, pero con coeficientes distintos:

$$A = \begin{bmatrix} 1+2r & -r & 0 & \cdots & 0 \\ -r & 1+2r & -r & \cdots & 0 \\ 0 & -r & 1+2r & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & -r \\ 0 & 0 & 0 & -r & 1+2r \end{bmatrix}.$$

Interpretación:

En este caso, el nuevo vector U^{n+1} no se puede calcular de forma directa, ya que está involucrado en ambos lados de la ecuación. Para avanzar un paso en el tiempo es necesario resolver el sistema lineal:

$$A\, U^{n+1} = U^n.$$

Esto se puede hacer en Python con:

```
U_next = numpy.linalg.solve(A, U)
```

2.4. Resumen comparativo

- **Explícito:**

$$U^{n+1} = B\,U^n$$

  Evoluciona mediante una multiplicación matricial sencilla.

- **Implícito:**

$$A\,U^{n+1} = U^n$$

  Requiere resolver un sistema lineal en cada paso temporal.

3. Evaluación con parámetros dados

Se evaluarán las funciones implementadas anteriormente con los siguientes parámetros:

$$\alpha = 1, \quad \Delta t = 0{,}001, \quad \Delta x = 0{,}05, \quad T = 0{,}1$$

La condición inicial $f(x)$ utilizada es:

$$f(x) = \begin{cases} 4x & 0 \leq x < \frac{1}{4}, \\ -2x + \frac{3}{2} & \frac{1}{4} \leq x < \frac{1}{2}, \\ 2x - \frac{1}{2} & \frac{1}{2} \leq x < \frac{3}{4}, \\ -4x + 4 & \frac{3}{4} \leq x \leq 1. \end{cases}$$

4. Visualización y análisis de resultados

Con los valores obtenidos en las simulaciones se construirá un GIF para visualizar cómo varía la solución a medida que evoluciona el tiempo. Se analizarán las diferencias entre los métodos explícito e implícito y se discutirá cómo el parámetro r afecta la estabilidad y precisión de los resultados.

Parte II

Ecuación de Transporte

5. Formulación matemática - Derivación de los métodos explícito e implícito

En esta sección vamos a aplicar el mismo procedimiento que en la parte 1 - ecuación del calor, pero ahora sobre la ecuación de transporte con velocidad constante α:

$$\frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} = 0, \quad x \in (0,1), \quad t > 0$$

con condición inicial sugerida:

$$u(x,0) = f(x) = \sin(\pi x).$$

Aquí $u(x,t)$ describe cómo una cantidad se desplaza en el tiempo a lo largo del eje x con velocidad α.

Al igual que antes, trabajamos con condiciones de frontera de Dirichlet homogéneas:

$$u(0,t) = u(1,t) = 0$$

5.1. Discretización: malla de espacio y tiempo

La malla espacial y temporal es exactamente la misma que usamos en la ecuación del calor:

- Puntos espaciales:
$$x_j = j\Delta x, \quad j = 0, 1, \ldots, N, \quad \Delta x = \tfrac{1}{N}.$$
- Puntos temporales:
$$t_n = n\Delta t, \quad n = 0, 1, 2, \ldots, M.$$

Denotamos la aproximación como $u_j^n \approx u(x_j, t_n)$.

5.2. Aproximación de las derivadas con diferencias finitas

Como en la ecuación del calor, aproximamos las derivadas con fórmulas en diferencias finitas. La diferencia es que aquí sólo necesitamos la derivada espacial de primer orden.

- Derivada temporal: diferencia hacia adelante
$$\frac{\partial u}{\partial t}(x_j, t_n) \approx \frac{u_j^{n+1} - u_j^n}{\Delta t}.$$
- Derivada espacial: diferencia hacia atrás (esquema para $\alpha > 0$)
$$\frac{\partial u}{\partial x}(x_j, t_n) \approx \frac{u_j^n - u_{j-1}^n}{\Delta x}.$$

5.3. Método explícito

Partimos de la ecuación discretizada:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + \alpha \frac{u_j^n - u_{j-1}^n}{\Delta x} = 0.$$

Definición de r:

Sustituimos $r = \frac{\alpha \Delta t}{\Delta x}$:

$$u_j^{n+1} - u_j^n = -r\,(u_j^n - u_{j-1}^n).$$

Paso 1. Expandir

$$u_j^{n+1} = u_j^n - r\,u_j^n + r\,u_{j-1}^n.$$

Paso 2. Agrupar términos

$$u_j^{n+1} = (1-r)\,u_j^n + r\,u_{j-1}^n.$$

Esquema explícito

$$\boxed{u_j^{n+1} = (1-r)\,u_j^n + r\,u_{j-1}^n}$$

Condición de estabilidad:

El esquema explícito es estable si y sólo si:

$$0 \le r \le 1.$$

Interpretación:

El nuevo valor en j es un promedio ponderado entre el valor anterior en j y el vecino izquierdo $j-1$. Esto refleja el transporte de la información en la dirección de la velocidad α.

5.4. Método implícito

Ahora evaluamos la derivada espacial en el tiempo $n+1$:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + \alpha \frac{u_j^{n+1} - u_{j-1}^{n+1}}{\Delta x} = 0.$$

Paso 1. Sustituir r

$$u_j^{n+1} - u_j^n = -r\,(u_j^{n+1} - u_{j-1}^{n+1}).$$

Paso 2. Reorganizar

$$(1-r)\,u_j^{n+1} + r\,u_{j-1}^{n+1} = u_j^n.$$

Esquema implícito

$$\boxed{(1-r)\,u_j^{n+1} + r\,u_{j-1}^{n+1} = u_j^n}$$

Interpretación:

En este caso, los valores de u^{n+1} aparecen acoplados entre sí. Esto obliga a resolver un sistema lineal en cada paso temporal. La ventaja es que el método implícito es **incondicionalmente estable**, sin restricciones sobre r.

5.5. Resumen comparativo

- **Explícito:**
$$u_j^{n+1} = (1-r)\,u_j^n + r\,u_{j-1}^n,$$
sencillo de implementar, pero estable sólo si $0 \leq r \leq 1$.
- **Implícito:**
$$(1-r)\,u_j^{n+1} + r\,u_{j-1}^{n+1} = u_j^n,$$
más costoso computacionalmente, porque requiere resolver un sistema lineal en cada paso, pero estable para cualquier r.

6. Sistemas lineales y forma matricial

6.1. Recordemos el dominio

Como ya vimos en la parte 1, trabajamos con $N+1$ puntos espaciales. Los valores desconocidos corresponden a los nodos internos $j = 1, \ldots, N-1$. En cada instante de tiempo usamos un vector de tamaño $N-1$.

6.2. Método explícito como sistema lineal

El método explícito se escribe como:

$$u_j^{n+1} = (1-r)\,u_j^n + r\,u_{j-1}^n.$$

Forma matricial:

$$U^{n+1} = B\,U^n,$$

con

$$U^n = \begin{bmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{N-2}^n \\ u_{N-1}^n \end{bmatrix}.$$

La matriz B en este caso es bidiagonal:

$$B = \begin{bmatrix} 1-r & 0 & 0 & \cdots & 0 \\ r & 1-r & 0 & \cdots & 0 \\ 0 & r & 1-r & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & r & 1-r \end{bmatrix}.$$

Interpretación:

La evolución temporal se obtiene simplemente multiplicando U^n por B. Cada fila refleja la dependencia entre un nodo y su vecino izquierdo.

6.3. Método implícito como sistema lineal

El método implícito es:

$$(1-r)\,u_j^{n+1} + r\,u_{j-1}^{n+1} = u_j^n.$$

Forma matricial:

$$A\,U^{n+1} = U^n,$$

donde

$$A = \begin{bmatrix} 1-r & 0 & 0 & \cdots & 0 \\ r & 1-r & 0 & \cdots & 0 \\ 0 & r & 1-r & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & r & 1-r \end{bmatrix}.$$

Interpretación:

Aquí U^{n+1} está en ambos lados de la ecuación, por lo que en cada paso debemos resolver:

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

6.4. Resumen comparativo

- **Explícito:**

  $$U^{n+1} = B\,U^n,$$

  evolución mediante una multiplicación matricial (bidiagonal).

- **Implícito:**

  $$A\,U^{n+1} = U^n,$$

  requiere resolver un sistema lineal en cada paso temporal (también con matriz bidiagonal).

7. Evaluación con parámetros dados

Se evaluarán las funciones implementadas anteriormente con los siguientes parámetros:

$$\alpha = 1, \quad \Delta t = 0{,}001, \quad \Delta x = 0{,}05, \quad T = 0{,}1$$

La condición inicial $f(x)$ utilizada es:

$$f(x) = \begin{cases} 4x & 0 \leq x < \frac{1}{4}, \\ -2x + \frac{3}{2} & \frac{1}{4} \leq x < \frac{1}{2}, \\ 2x - \frac{1}{2} & \frac{1}{2} \leq x < \frac{3}{4}, \\ -4x + 4 & \frac{3}{4} \leq x \leq 1. \end{cases}$$

8. Visualización y análisis de resultados

Con los valores obtenidos en las simulaciones se construirá un GIF para visualizar cómo varía la solución a medida que evoluciona el tiempo. Se analizarán las diferencias entre los métodos explícito e implícito y se discutirá cómo el parámetro r afecta la estabilidad y precisión de los resultados.