

# Tercer informe de Física Computacional: Ecuaciones con derivadas parciales

Francisco Fernández

Abril del 2020

## Resumen

En este informe se estudió el comportamiento de la ecuación del calor en una barra unidimensional, para ello se desarrollaron tres métodos numéricos que la resuelven. Se realizaron gráficos de superficies 3D y de contorno para controlar el comportamiento de la temperatura, se midieron los tiempos de computo y se los comparó con el cálculo exacto, también se observó como el error cambia en función del tiempo y de la cantidad de divisiones que haya en el espacio.

## 1. Introducción teórica

### 1.1. Generalidades

Algunas cantidades físicas varían continuamente en el espacio y el tiempo, un ejemplo de ellas, que se analizará en este informe, es el de la temperatura. Los cambios en un punto de la función o el *campo*  $U(x, y, z, t)$  que las describe afecta a los puntos vecinos a medida que el tiempo avanza. Esto nos dice que las ecuaciones dinámicas que describen la dependencia de  $U$  con cuatro variables independientes debe ser escrita en términos de derivadas parciales. Por ello tenemos *ecuaciones en derivadas parciales* (EDP).

De forma general, una EDP lineal y de segundo orden, con dos variables independientes, puede escribirse como

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

donde los términos que acompañan a las derivadas son funciones arbitrarias de las variables  $x$  e  $y$ ,  $F$  además puede ser función de  $U$ . De aquí puede definirse el discriminante  $d$

$$d = AC - B^2,$$

que, según su comportamiento, pueden clasificarse EDPs elípticas, parabólica o hiperbólicas.

A diferencial de las ODEs, en las que solo necesitábamos condiciones iniciales para nuestras ecuaciones, para que la solución sea única también hay que especificar condiciones de contorno que pueden ser de:

1. *Dirichlet*: en donde se especifican los valores de la solución en la superficie que rodea al dominio,

2. Neumann: en las cuales se dan los valores de la derivada normal en la superficie que rodea al dominio,
3. Cauchy: donde se tienen ambos valores, la solución y la derivada, en la superficie que rodea al dominio,
4. Mixtas: combinación de las anteriores en distintas regiones.

## 1.2. Ecuación del calor

Si tenemos un material con conductividad térmica  $K$ , calor específico  $C$  y densidad  $\rho$ , entonces la *ecuación del calor* que se obtiene usando la conservación de la energía y suponiendo que la difusión del calor es proporcional al gradiente de la temperatura, para una dimensión, es

$$\frac{\partial T(x, t)}{\partial t} = \frac{K}{C\rho} \frac{\partial^2 T(x, t)}{\partial x^2}, \quad (1)$$

de donde puede definirse la constante de difusión  $D = \frac{K}{C\rho}$ . El discriminante de esta ecuación es igual a cero, por lo tanto es una EDP parabólica.

En este informe se estudiará el caso con las siguientes condiciones iniciales y de contorno de Dirichlet

$$T(x, t = 0) = T_0 = 100^\circ C, \quad T(x = 0, t) = T(x = L, t) = 0^\circ C.$$

### 1.2.1. Solución analítica

Si suponemos separación de variables para el tiempo y el espacio tenemos que

$$T(x, t) = X(x)T(t),$$

y reemplazando en la ecuación (1) se obtienen dos ODEs no acopladas. Las soluciones generales para  $X(x)$  son sumas de  $\sin(x)$  y  $\cos(x)$ , mientras que para  $T(t)$  es una exponencial decreciente. Evaluando las condiciones del problema se tiene que

$$T(x, t) = 4T_0 \sum_{n=1,3,\dots}^{\infty} \frac{\sin(n\pi x)}{n\pi} e^{-(n\pi)^2 Dt}. \quad (2)$$

### 1.2.2. Solución numérica

Para la resolución numérica se convierten las ecuaciones diferenciales en ecuaciones de diferencias finitas. Se discretiza el espacio ( $x_i = i\Delta x$ ) y el tiempo ( $t_j = j\Delta t$ ) en una grilla y se resuelve en cada uno de los sitios.

#### Método explícito

En este método, para la derivada de la temperatura se utiliza diferencia hacia adelante y para la derivada espacial utilizamos diferencia centrada. Reemplazando los resultados correspondientes en la ecuación (1) y reordenando se tiene

$$T_{i,j+1} = T_{i,j} + \eta [T_{i+1,j} + T_{i-1,j} - 2T_{i,j}], \quad (3)$$

donde se introdujo  $\eta = \frac{\Delta t}{\Delta x^2}$ . A este algoritmo se lo llama *explícito* porque provee la solución en términos de valores conocidos de la temperatura.

De la ecuación (3) vemos que la temperatura en el punto espacio-tiempo  $(i, j + 1)$  se computa utilizando los valores del tiempo anterior  $j$  y las posiciones adyacentes  $i - 1$ ,  $i$ ,  $i + 1$ .

Este método es estable sólo cuando  $\eta < \frac{1}{2}$ .

### Método implícito

En este método se aproxima la derivada temporal por diferencias hacia atrás y, al igual que antes, diferencias centradas para la derivada segunda espacial, de donde se obtiene

$$T_{i,j-1} = -\eta T_{i-1,j} + (1 + 2\eta)T_{i,j} - \eta T_{i+1,j}, \quad (4)$$

que es siempre estable. A este algoritmo se lo llama *implícito* porque se tiene que resolver ecuaciones simultaneas para obtener la solución para todo el espacio, el siguiente método también tiene esta propiedad.

### Método de Cranck-Nicolson

Este método utiliza un “*split time step*”, llevando al ecuación a  $t + \Delta t/2$ , para calcular la derivada temporal con una aproximación central y evitar introducir error en el paso temporal inicial. Si de cada lado de la ecuación dejamos los términos que involucran el mismo tiempo, entonces se tiene

$$-T_{i-1,j+1} + \left(\frac{2}{\eta} + 2\right) T_{i,j+1} - T_{i+1,j+1} = T_{i-1,j} + \left(\frac{2}{\eta} - 2\right) T_{i,j} + T_{i+1,j}, \quad (5)$$

que también es siempre estable.

## 1.3. Matrices tridiagonales

Si se realiza un análisis matricial de los últimos dos métodos de la sección anterior se puede ver que la matriz que relaciona las temperaturas anteriores con las siguientes es tridiagonal, y esto es algo que en la implementación numérica se debe explotar.

El problema de la resolución de una matriz,

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b},$$

precisa el almacenamiento de  $N^2$  elementos, pero si solo almacenamos los valores distintos de cero, tres vectores  $\vec{a}$ , de la diagonal inferior,  $\vec{d}$ , de la diagonal principal, y  $\vec{c}$ , de la diagonal superior. De esta manera se reduce el almacenamiento a  $(3N - 2)$  elementos.

Lo primero que se realizan son operaciones de línea para llevar el sistema a triangular superior con la diagonal principal igual a 1. De acá sale que  $h_1 = c_1/d_1$ ,  $p_1 = b_1/d_1$  y, de forma recursiva se puede obtener

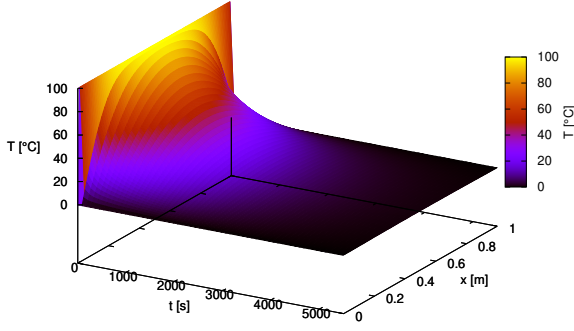
$$h_i = \frac{c_i}{d_i - a_i h_{i-1}}, \quad p_i = \frac{b_i - a_i p_{i-1}}{d_i - a_i h_{i-1}},$$

para luego realizar *back substitution*,

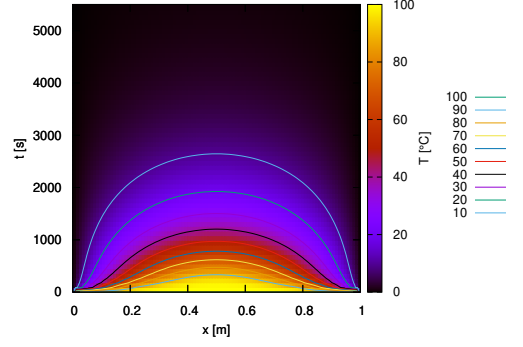
$$x_i = p_i - h_i x_{i+1}, \quad i = n - 1, n - 2, \dots, 1, \quad x_N = p_N.$$

## 2. Resultados y discusiones

En el problema 1 de la guía de Ecuaciones en Derivadas Parciales se nos pide resolver la ecuación del calor (1) mediante diferentes algoritmos para una barra de aluminio de largo  $L = 1m$  y diámetro  $\omega$  despreciable ( $\omega \ll L$ ) alineada a lo largo del eje  $x$ . La misma se encuentra aislada en sus lados pero no en sus extremos, donde la temperatura está fija en  $0^\circ \text{C}$ . Inicialmente la barra



(a) Superficie  $T(x, t)$  versus  $(x, t)$ .



(b) Mapa de contornos con isotermas.

Figura 1: Gráficos de la temperatura en función de la posición y el tiempo para el método explícito.

se encuentra a  $100^\circ \text{C}$ . De esta manera el calor sólo fluye por los extremos. La conductividad térmica,  $K$ , el calor específico,  $C$ , y la densidad,  $\rho$ , del aluminio son:

$$K = 237 \text{W}/(\text{mK}); \quad C = 900 \text{J}/(\text{kgK}); \quad \rho = 2700 \text{kg}/\text{m}^3.$$

Para resolver numéricamente la ecuación del calor es conveniente adimensionalizarla (inciso a), para ello primeros notamos que las unidades de la constante de difusión son  $\text{m}^2/\text{s}$ . Entonces, si tomamos  $\tilde{t} = \frac{Dt}{L^2}$  y  $\tilde{x} = \frac{x}{L}$  y reemplazamos en ambos miembros de la ecuación (1) tenemos

$$\begin{aligned} \frac{\partial T}{\partial t} &= \frac{\partial T}{\partial \tilde{t}} \frac{\partial \tilde{t}}{\partial t} = \frac{D}{L^2} \frac{\partial T}{\partial \tilde{t}}, \\ \frac{\partial^2 T}{\partial x^2} &= \frac{\partial^2 T}{\partial \tilde{x} \partial \tilde{x}} \frac{\partial \tilde{x}}{\partial x} = \frac{\partial^2 T}{\partial \tilde{x}^2} \left( \frac{\partial \tilde{x}}{\partial x} \right)^2 = \frac{1}{L^2} \frac{\partial^2 T}{\partial \tilde{x}^2}, \end{aligned}$$

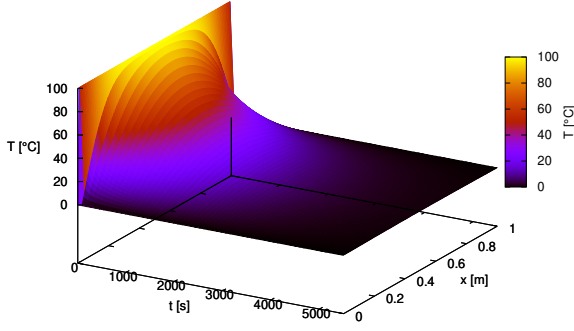
considerando que esta última ecuación va multiplicada por la constante de difusión, nos queda la ecuación del calor adimensionalizada

$$\frac{\partial T(\tilde{x}, \tilde{t})}{\partial \tilde{t}} = \frac{\partial^2 T(\tilde{x}, \tilde{t})}{\partial \tilde{x}^2}.$$

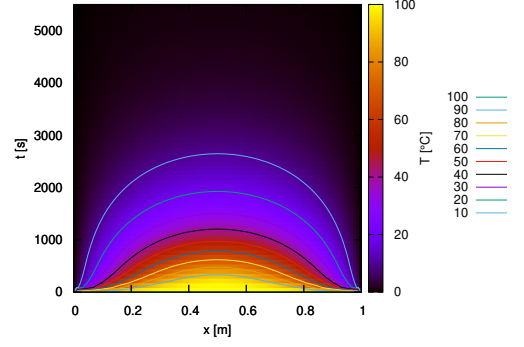
## 2.1. Método explícito

Para este método se escribió el programa principal en Fortran, `main.f90`, un módulo que resuelve el método explícito, `metodos.f90`, tomando como valores de entrada las divisiones en el eje  $x$ , el valor de la temperatura en los extremos y dos vectores, uno para la temperatura del paso temporal siguiente y otro para el actual. El código principal, además de llamar a dicho módulo, también calcula la solución exacta cada 300 pasos temporales, que es cuando se escribe en un archivo de salida y, a pesar de hacer los cálculos adimensionales, imprime los valores en segundos y metros. Hacia el final del código tiene una sección que solo se ejecuta si desde la terminal se pide que se mida el tiempo de cada método.

En la figura (1) podemos observar la superficie de temperatura en función del espacio y el tiempo. Se ve que la distribución de temperatura varía suavemente, tanto a lo largo de la barra como a medida que el tiempo aumenta, que cumple con las condiciones de contorno y que para tiempos largos, mayores a 4000 segundos, la temperatura tiende a cero en toda la barra. En particular, en la figura (1b) se muestran las isotermas cada  $10^\circ \text{C}$ , que muestran como se



(a) Superficie  $T(x, t)$  versus  $(x, t)$ .



(b) Mapa de contornos con isotermas.

Figura 2: Gráficos de la temperatura en función de la posición y el tiempo para el método implícito.

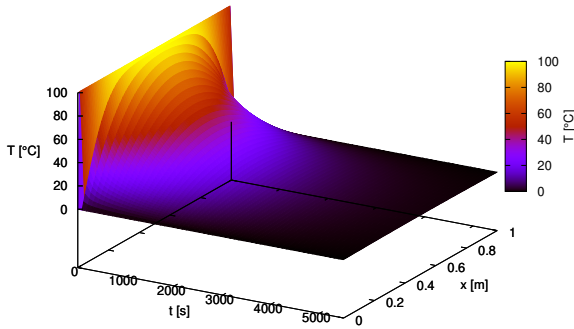
comporta el flujo de calor. Puede observarse como la temperatura decrece rápidamente cerca de los extremos de la barra, a 10cm de cada extremo, tomando para 100s una temperatura menor a 40°C, y como en el centro se demora 2500s para bajar a 10°C.

## 2.2. Método implícito

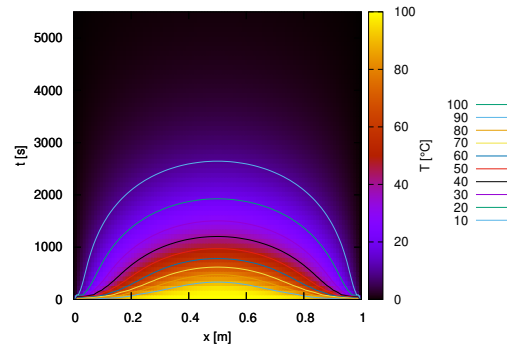
En este caso se cambió ligeramente el `main.f90` para mantener las mismas condiciones iniciales y de borde y mantener la solución exacta, se modificó el *loop* principal para que ahora calculara la temperatura siguiente con el método implícito, para ello se definen los vectores de la matriz tri-diagonal y se llama a la subrutina que la resuelve, que está dentro del módulo, que también se utilizó para el método siguiente, `matrix_solver.f90`.

A simple vista no se ven cambios en los resultados de la figura (2) con respecto a los de la sección anterior, sin embargo, en un análisis posterior de los errores se marcarán las diferencias.

## 2.3. Método de Cranck-Nicolson



(a) Superficie  $T(x, t)$  versus  $(x, t)$ .



(b) Mapa de contornos con isotermas.

Figura 3: Gráficos de la temperatura en función de la posición y el tiempo para el método de Crank-Nicolson.

En este método solo fue necesario modificar los parámetros de entrada a la matriz tri-diagonal para que cumplan la ecuación (5). En la figura (3) se muestran los resultados obtenidos a partir de este algoritmo.

En todos los gráficos referenciados hasta ahora se utilizó  $\Delta t = 0,3s$  y 100 divisiones equiespaciadas en el eje  $x$ . Todos los códigos que han sido mencionados se encuentran adjuntos al informe.

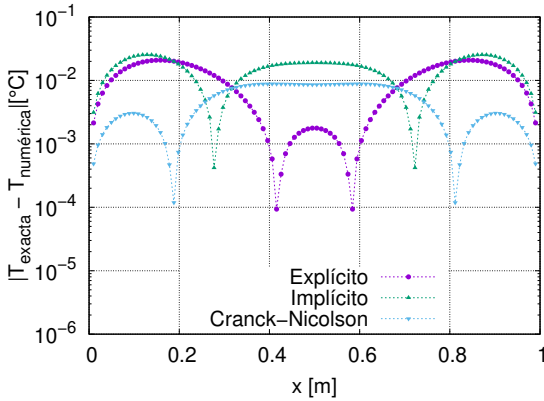
## 2.4. Comparación entre métodos

En la tabla (1) se presentan los tiempos de simulación de los tres métodos y de la solución exacta con 50 términos. Para obtener estos valores se utilizó subrutina `cpu_time` de fortran, llamándola a los tiempos anteriores y posteriores al *loop* principal, sin considerar las definiciones de variable y las asignaciones de memoria, sin que escriba nada para no considerar el tiempo de I/O. Para todas las mediciones se utilizaron 20000 pasos temporales y 100 divisiones en el eje  $x$ . Lo primero que observamos es una diferencia grande entre los métodos numéricos y el método exacto, de tres ordenes de magnitud, que pude deberse a que en los primeros solo se realizan operaciones elementales (sumas y multiplicaciones) mientras que en el segundo se opera con funciones sinusoidales y exponenciales predefinidas en Fortran. Entre el método explícito y los otros dos numéricos, que ambos utilizan la resolución de una matriz tri-diagonal y el código, salvo la definición de los vectores, es el mismo, vemos una diferencia menor a  $3 \times 10^{-2}s$ , por lo que el primer método es ligeramente más rápido.

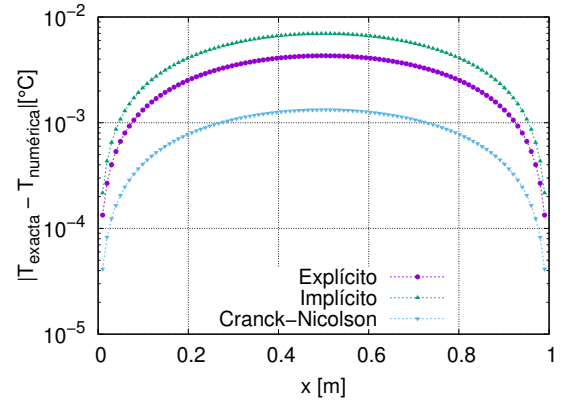
Tabla 1: Tiempos de computo para 20000 pasos temporales según el método.

	Método			
	Explícito	Implícito	Cranck-Nicolson	Exacto
Tiempo de CPU [s]	$7 \times 10^{-3}$	$3 \times 10^{-2}$	$3 \times 10^{-2}$	0,75

En la figura (4) se presenta el error local de la temperatura para 100 divisiones espaciales y dos tiempos específicos. En particular, en la figura (4a), se ve como para un tiempo cercano al inicio de la simulación (600 pasos temporales versus 20000), para los extremos de la barra el método más preciso es el de Crack-Nicolson y para el centro es el explícito. También se observan distintos puntos en los cuales el error decae. Sin embargo, para un tiempo posterior en la simulación (ver figura 4b) los errores, para los tres métodos, son menores cerca de los bordes que en el centro de la barra, con una diferencia de dos ordenes de magnitud. También se ve una diferencia de un orden de magnitud entre el de Cranck-Nicolson, que toma un valor cercano a  $10^{-3}^{\circ}C$ , y el resto para el centro de la barra.



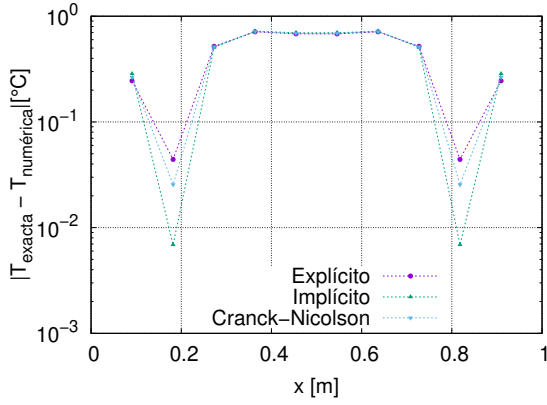
(a)  $t = 180s$



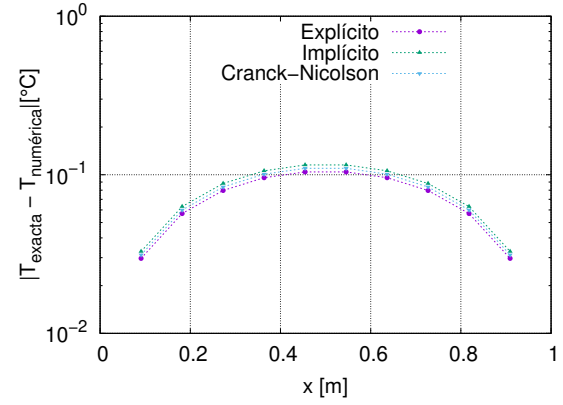
(b)  $t = 1800s$

Figura 4: Error local de la temperatura a lo largo de la barra con 100 divisiones en el eje  $x$  para dos tiempos en particular.

En la figura (5) vemos los mismos errores pero para 10 divisiones en el eje  $x$ . Se observa para tiempos cortos (180s) en la figura (5a) el mismo comportamiento del error para los tres métodos, del orden de  $6 \times 10^{-1}^\circ\text{C}$ , salvo en dos puntos simétricos (0,2m y 0,8m) en los cuales el error decae y el valor mínimo de  $\sim 7 \times 10^{-3}^\circ\text{C}$  es alcanzado por el método implícito. Luego, para un tiempo mayor, de 1800s, se observa que los tres métodos tienen ligeramente el mismo error de  $\sim 10^{-1}^\circ\text{C}$  en el centro de la barra y de  $2 \times 10^{-2}^\circ\text{C}$  en los bordes de la misma.



(a)  $t = 180s$



(b)  $t = 1800s$

Figura 5: Error local de la temperatura a lo largo de la barra con 10 divisiones en el eje  $x$  para dos tiempos en particular.

Como conclusión final se puede decir que los tres métodos resuelven bien el problema físico pero que el de Crank–Nicolson es el más preciso de los tres y, a diferencia del explícito, es siempre estable, para cualquier valor de  $\Delta t$  y  $\Delta x$ .