

## Problema 2

### Ecuación de Calor: condiciones de contorno de Neumann

Resolveremos, mediante diferentes algoritmos, la ecuación de calor ya adimensionalizada

$$\frac{\partial[T(\tilde{x}, \tilde{t})]}{\partial \tilde{t}} = \frac{\partial^2[T(\tilde{x}, \tilde{t})]}{\partial^2 \tilde{x}} \quad (1)$$

con las siguientes condiciones de contorno e iniciales

$$T(x, 0) = \cos(\pi x), \quad \underbrace{\frac{\partial[T(0, t)]}{\partial x}}_{g_0} = \underbrace{\frac{\partial[T(L, t)]}{\partial x}}_{g_L} = 0 \quad (2)$$

- Modifique los códigos escritos en el problema anterior para que consideren condiciones de contorno de Neumann.
- Resuelva la ecuación utilizando los diferentes métodos. Utilice  $\Delta\tilde{x} = 0.05$  y  $\Delta\tilde{t} = 0.001$ . Al menos para alguno de ellos, haga un gráfico de superficie mostrando  $T(x, t)$  versus  $(x, t)$  y un mapa de temperatura incluyendo curvas de nivel.
- La solución analítica es

$$T(x, t) = \exp(-\pi^2 t) \cos(\pi x) \equiv T(x, 0) \exp(-\pi^2 t) \quad (3)$$

Compare en un gráfico de  $T(x, 1)$  versus  $x$  los tres métodos y la solución exacta. Grafique, además, el "máximo error absoluto" de la solución numérica a lo largo de la barra, para cada tiempo, versus tiempo. Compare los tres métodos (en la misma figura).

- Si disminuye a  $\Delta\tilde{t} = 0.00025$ , ¿qué espera obtener? ¿habría que cambiar/variar algo más? Repita las gráficas del inciso anterior, compare con resultados previos y discuta la teoría.

## Introducción

Aquí para calcular la cantidad de elementos finitos tuvimos en cuenta que

$$\Delta\tilde{x} = \frac{1}{(n+1)} \Rightarrow n = \text{int}\left(\frac{1}{\Delta\tilde{x}} - 1\right)$$

sin embargo, el calculo anterior puede no ser correcto pues, como  $n$  es entero y  $\Delta\tilde{x}$  es real, al hacer la conversión anterior pude darnos el entero inmediatamente anterior al correcto para obtener un  $\Delta\tilde{x}_{approx}$  lo más cercano al propuesto para ello, se utilizó la función módulo y se implementó el siguiente

```

1 [...]
2 integer(sp)      :: n           ! numero de elementos finitos (FE)
3 integer(sp)      :: a,b         ! variables auxiliares
4 real(dp),parameter :: x_step_adim=0.05 ! deltaX adimensional
5 ! calculo parte entera del modulo y del resto de x_step_adim^(-1)
6 a=int(mod(1._dp,x_step_adim),sp); b=int(1._dp/x_step_adim,sp)
7 ! calculamos el número de FE
8 n=(a+b-1_sp)*(1_sp/(1_sp-a))
9 [...]
```

esto nos asegura que el número entero  $n$  obtenido siempre nos asegura obtener el  $\Delta\tilde{x}_{approx}$  más cercano al real. Y además, como el valor de  $\Delta\tilde{x}$  debe estar entre cero y uno, la formula de la línea (10) nos asegura que no exista un error aritmético.

**Inciso a)**

Se tuvieron en cuenta las siguientes ecuaciones que modifican que nos permite modificar los códigos del problema 1 para utilizar las condiciones de contorno de Von-Neumann, en resumidas cuentas, los que hacemos es no sólo evolucionar los  $n$  puntos internos de la barra (de 2 a  $n + 1$  puntos) sino que también se evolucionan los puntos extremos, y en total se estarían evolucionando de 1 a  $n + 2$  puntos de la barra (todos los puntos físicos). Esta es la principal diferencia respecto a las condiciones de contorno de Dirichlet, es decir, el problema ahora se modifica agregando dos puntos "fantasmas" (los puntos ficticios 0 y  $n + 3$ ) y se encuentra una expresión de estos puntos en términos de los puntos reales de la grilla pero con el agregado de que ahora evolucionan todos los puntos de la barra (esto se hace aproximando la derivada, condiciones de borde de V-N, por su expresión en el método de diferencias centradas).

**Método explícito (diferencia hacia adelante para derivada temporal)**

$$\begin{aligned} T_{1,(j+1)} &= 2\eta T_{1,j} + (1 - 2\eta)T_{0,j} - 2\eta\Delta x g_0; i = 1 \\ T_{i,(j+1)} &= (1 - 2\eta)T_{i,j} + \eta[T_{(i-1),j} + T_{(i+1),j}] \quad \forall 2 \leq i \leq (n + 1) \\ T_{n+2,(j+1)} &= 2\eta T_{(n+2),j} + (1 - 2\eta)T_{(n+2),j} + 2\eta\Delta x g_{(n+2)}; i = (n + 2) \end{aligned}$$

**Método implícito (diferencia hacia atrás para derivada temporal)**

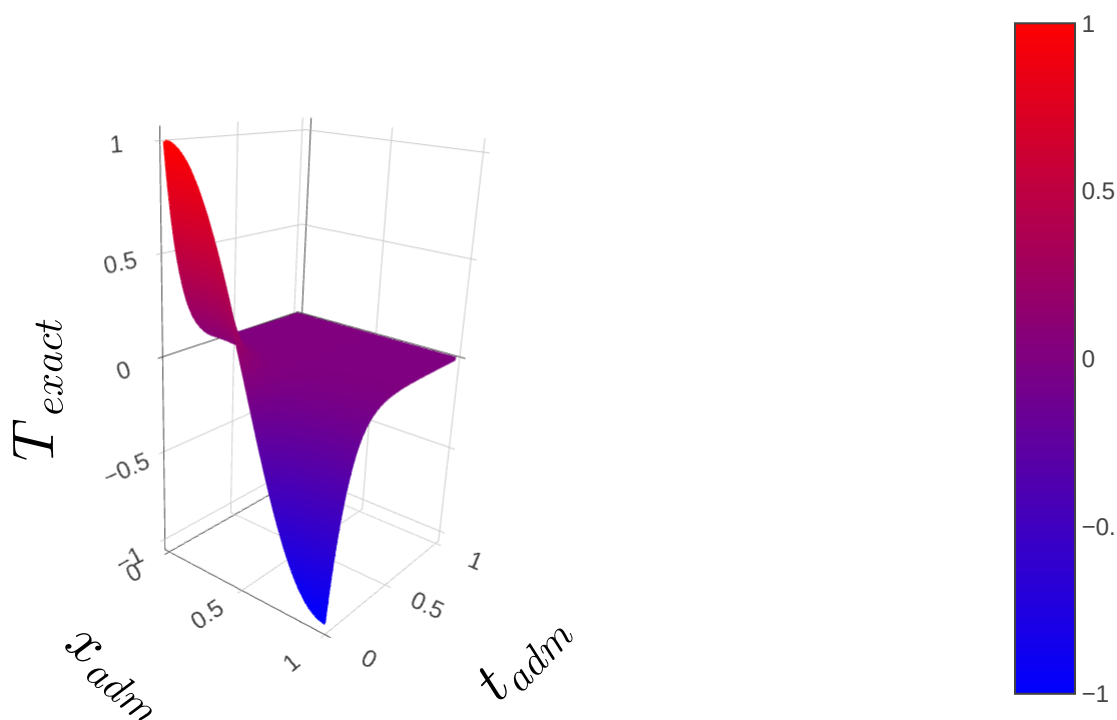
$$\begin{aligned} T_{1,(j+1)} &= -2\eta T_{1,j} + (1 - 2\eta)T_{0,j} + 2\eta\Delta x g_0; i = 1 \\ T_{i,(j+1)} &= (1 + 2\eta)T_{i,j} - \eta[T_{(i-1),j} + T_{(i+1),j}] \quad \forall 2 \leq i \leq (n + 1) \\ T_{n+2,(j+1)} &= -2\eta T_{(n+2),j} + (1 + 2\eta)T_{(n+2),j} - 2\eta\Delta x g_{(n+2)}; i = (n + 2) \end{aligned}$$

**Método de Crank-Nicolson (diferencia centrada para derivada temporal - split lineal)**

$$\begin{bmatrix} \frac{2}{\eta} + 2 & -2 & 0 & \cdots & \cdots 0 \\ -1 & \frac{2}{\eta} + 2 & -1 & \cdots & \cdots 0 \\ 0 & \cdots & \cdots & \cdots & \vdots \\ \vdots & 0 \cdots & -1 & \frac{2}{\eta} + 2 & -1 \\ 0 & 0 \cdots & 0 & -2 & \frac{2}{\eta} + 2 \end{bmatrix} \cdot \begin{bmatrix} T_{0,j+1} \\ T_{1,j+1} \\ \vdots \\ T_{n,j+1} \\ T_{n+1,j+1} \end{bmatrix} = \begin{bmatrix} \frac{2}{\eta} - 2 & -2 & 0 & \cdots & \cdots 0 \\ -1 & \frac{2}{\eta} - 2 & -1 & \cdots & \cdots 0 \\ 0 & \cdots & \cdots & \cdots & \vdots \\ \vdots & 0 \cdots & -1 & \frac{2}{\eta} - 2 & -1 \\ 0 & 0 \cdots & 0 & -2 & \frac{2}{\eta} - 2 \end{bmatrix} \cdot \begin{bmatrix} T_{0,j} \\ T_{1,j} \\ \vdots \\ T_{n,j} \\ T_{n+1,j} \end{bmatrix} + \begin{bmatrix} 2\eta\Delta x (g_{0,j} - g_{0,j+1}) \\ 0 \\ \vdots \\ 0 \\ -2\eta\Delta x (g_{n+1,j} - g_{n+1,j+1}) \end{bmatrix}$$

**inciso c)**

Si graficamos la solución exacta obtenemos el siguiente resultado



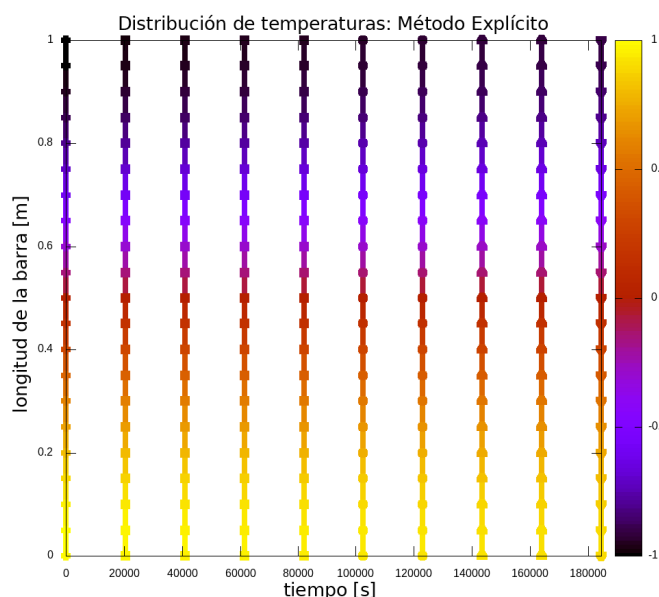
esto nos va a permitir comparar fielmente los resultados numéricos. Además, de la ecuación exacta y las condiciones de contorno vemos que las fórmulas consideran a todas las variables como adimensionales es decir, coordenada espacial ( $x$ ), coordenada temporal ( $t$ ) y temperatura ( $T$ ). Ahora bien, como la temperatura, para cualquier tiempo, oscila entre  $-1$  y  $1$  se deduce que estamos pensando en temperaturas adimensionales obtenidas a partir de temperaturas en grados centígrados pues, es posible obtener temperaturas negativas que sabemos que no puede ser así si consideramos temperaturas en Kelvins.

Por otro lado, como la temperatura de termalización es cero y las temperaturas máximas que puede alcanzar la barra son de  $\pm 1$  (valor no muy lejano al cero) los tiempos de relajación al equilibrio son relativamente cortos, comparados, por ejemplo, con el problema 1 de la guía (que requería  $\approx 15000$  pasos temporales de evolución). Aquí, con solo 1000 pasos temporales, la temperatura se encuentra a valores del  $\mathcal{O}(10^{-5})$ .

## Resultados y Discusiones

### Inciso b)

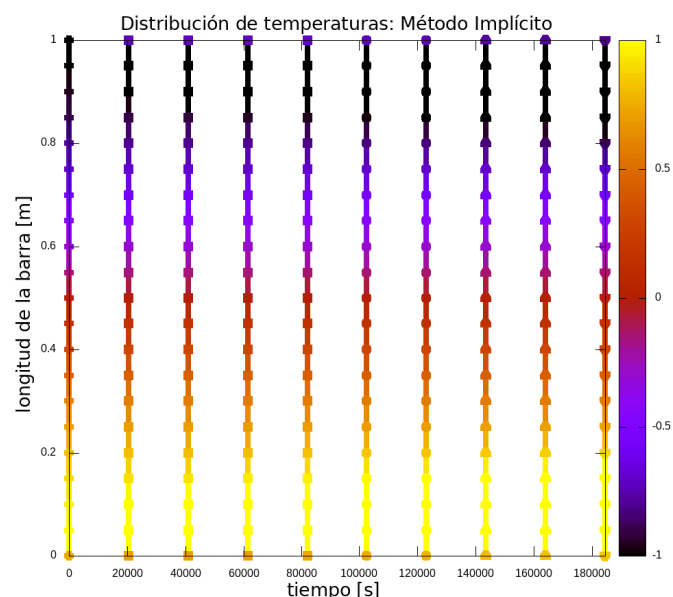
Los gráficos que muestran las distribuciones de temperatura vs el tiempo para los métodos explícito e implícito se muestran a continuación



**Figure 1:**

Si bien las coordenadas temporales y espaciales están adimensionalizadas se supuso que se trataba de una barra de  $1[m]$  de longitud (tal como fue definido en el problema 1 de la guía) y se dimensionalizó el tiempo con el tiempo característico de  $0.1025E+05[s]$ , además, se evolucionó la barra durante 20 pasos temporales y se imprimieron los datos cada 2 pasos temporales obteniendo 10 barras con su distribución de temperatura correspondiente.

Ahora bien, para el método de C-N se realizó un gráfico de superficie obteniendo,



**Figure 2:**

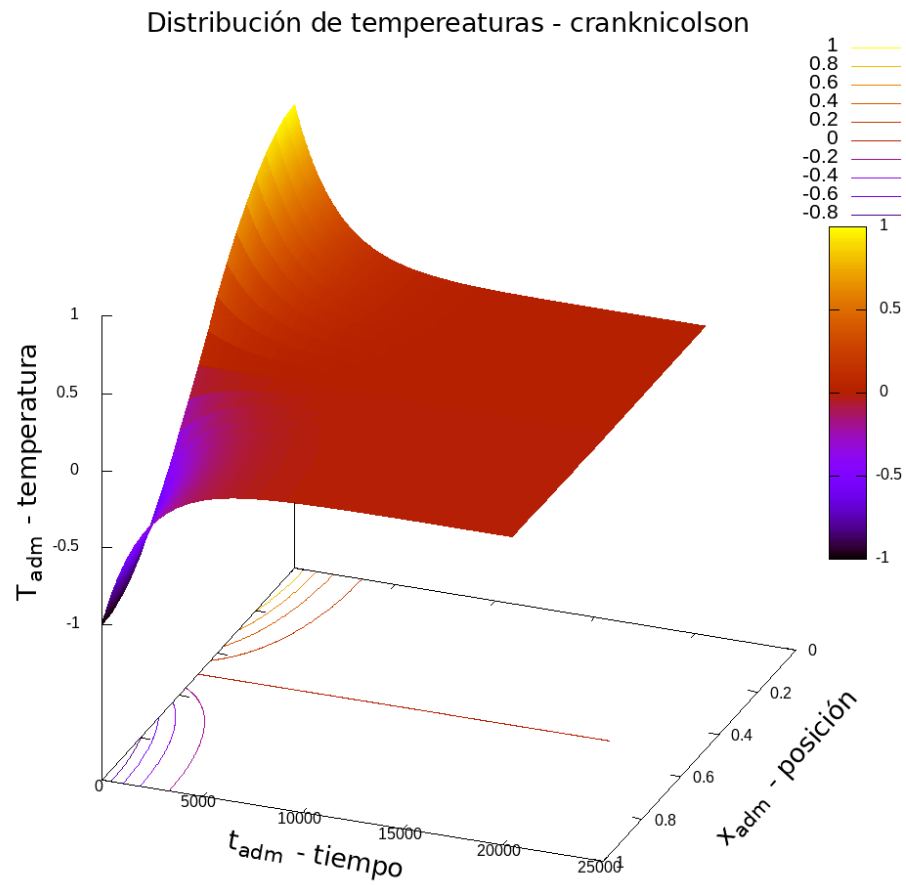


Figure 3:

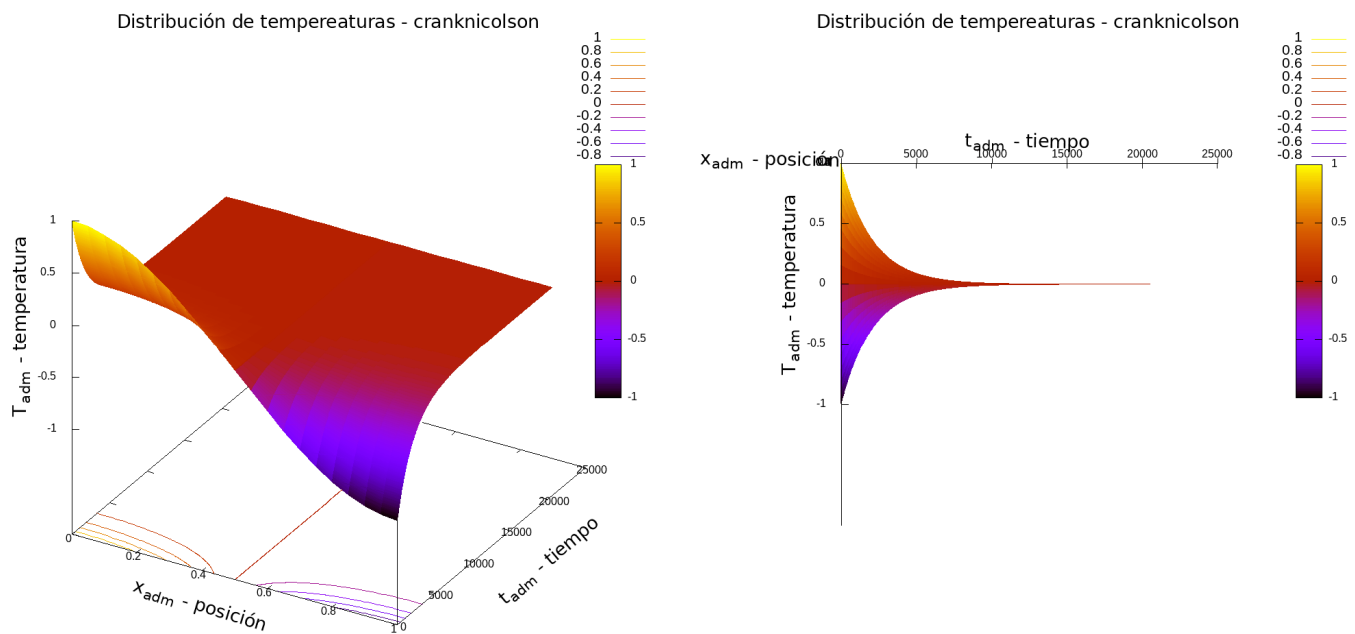


Figure 4:

Figure 5:

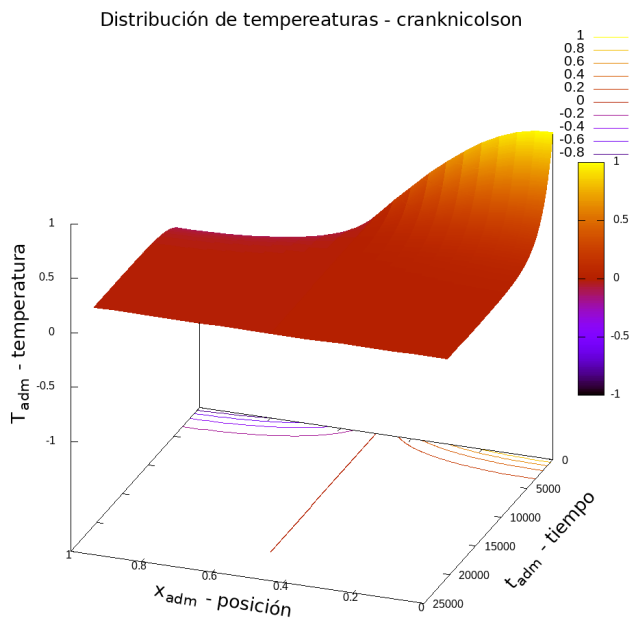


Figure 6:

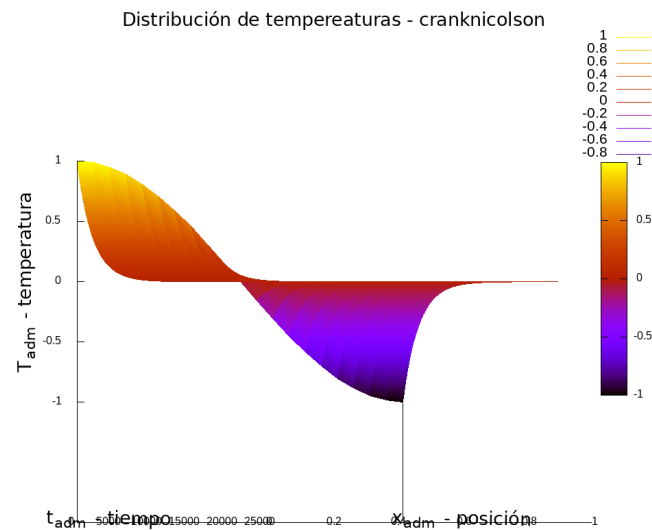


Figure 7:

Comparando esta gráfica de superficie con la superficie de la solución exacta vemos que son muy similares, en cada una de las dimensiones (por ello se han hecho vistas superior y lateral). Teniendo una distribución de temperaturas cosenoidal en el instante inicial y una distribución cosenoidal en el tiempo, modulada con una exponencial decreciente, lo cual nos muestra el rápido decaimiento al equilibrio térmico. En la base del gráfico se pueden apreciar algunas curvas de nivel.

**Inciso c)**

A continuación se muestran los errores numéricos obtenidos,

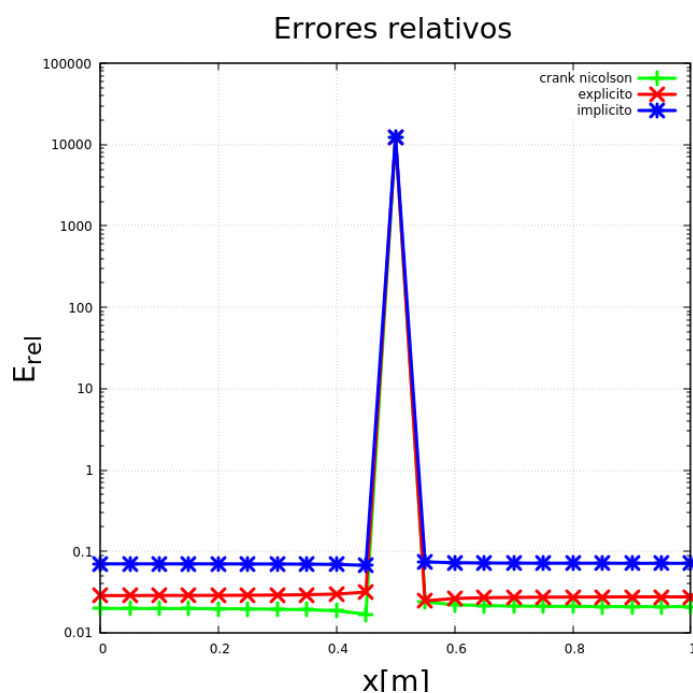


Figure 8:

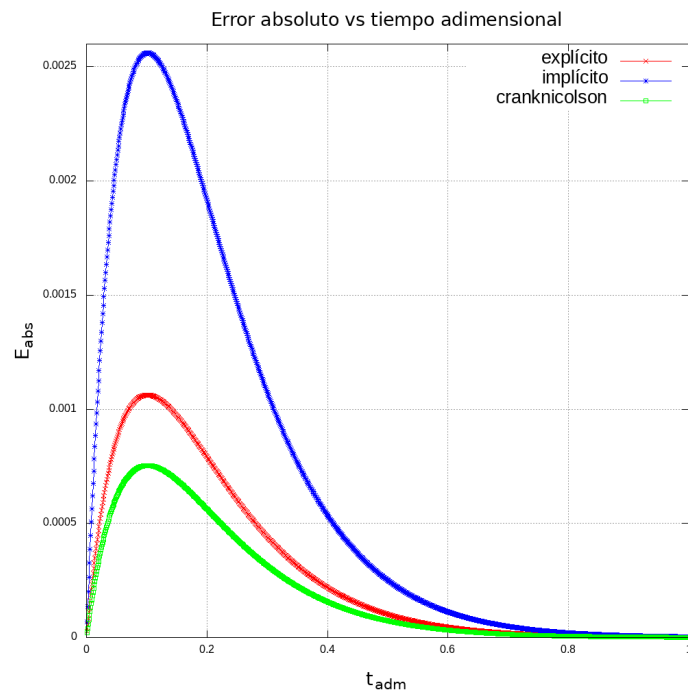
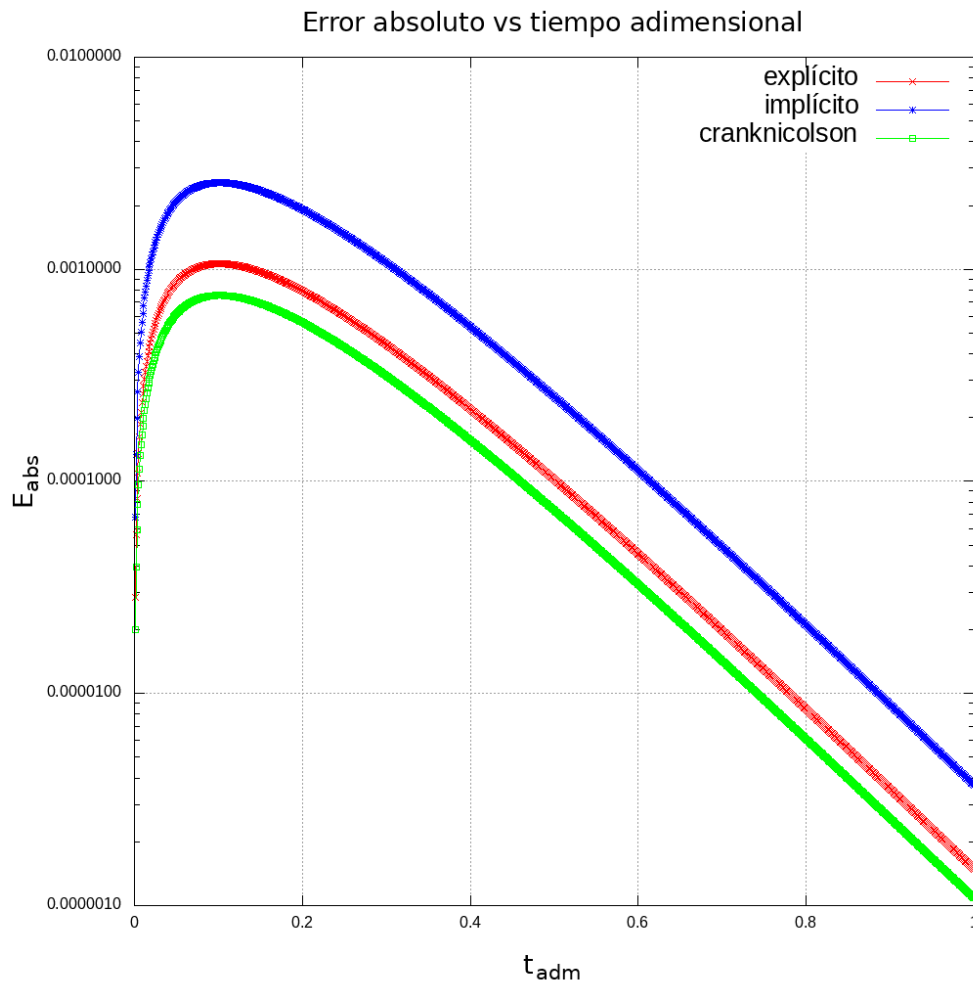


Figure 9:

En la primer figura, se observa el error relativo  $\epsilon_{rel} = \left| (\epsilon_{eacto} - \epsilon_{aprox.}) / \epsilon_{eacto} \right|$  para la distribución de temperaturas al tiempo fijo de  $t_{adm} = 1$  para cada posición de la barra. El mayor error se obtuvo

para el centro de la barra y para el método implícito, seguido por el método explícito y finalmente el método de C-N. Como el eje vertical se graficó en escala logarítmica notamos que los órdenes de magnitud son similares para cada uno de los métodos.

En la segunda figura, se observa el mayor error absoluto definido como  $\epsilon_{rel} = \max |(\epsilon_{exacto} - \epsilon_{aprox.})|$  para la distribución de temperaturas para todo tiempo (en el rango de  $0 \leq t_{adm} \leq 1$ ), al igual que en el caso anterior se observa que el método más preciso es el de C-N, seguido por el método implícito y finalmente por el método explícito. Además, en vemos que esta gráfica es útil para observar a que tiempo los errores son mayores, y a medida que incluimos más pasos temporales el error disminuye convergiendo al valor exacto. Si cambiamos la escala del eje vertical a logarítmica podremos apreciar la ley de potencia del error,



**Figure 10:**

al tener los tres métodos la misma pendiente siguen la misma ley de potencia.

**Inciso d)**

Si disminuimos la discretización temporal a un valor de  $\Delta\tilde{t} \equiv \Delta t_{adim} = 0.00025$  entonces tendremos que el parámetro para controlar estabilidad será (sin modificar la discretización espacial  $\Delta\tilde{x}$ )

$$\alpha = \frac{\Delta\tilde{t}}{(\Delta\tilde{x})^2} = \frac{0.00025}{(0.05)^2} = 0.1 < 0.5$$

los cual nos asegura que el método explícito es convergente y (como se ha discutido en el problema 1) si no modificamos la discretización en la coordenada espacial el error quizás disminuya pero no apreciablemente, pues la discretización de la barra seguirá siendo la misma que en los casos anteriores y el error mínimo estaría condicionado a esta discretización, lo que si se esperaría es que los tiempos de CPU sean mayores, pues para un tiempo final de termalización fijo, al achicar el

paso temporal nos tomaría más pasos evolucionar para llegar a dicho tiempo.

## Códigos

### Repositorio GitHub

<https://github.com/mendzmartin/fiscomp2022.git>

### Repositorio GitHub del problema

<https://github.com/mendzmartin/fiscomp2022/tree/main/lab02/prob02>

### Programas principales

[https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/heateq\\_explicit\\_von\\_neumann.f90](https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/heateq_explicit_von_neumann.f90)

[https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/heateq\\_implicit\\_von\\_neumann.f90](https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/heateq_implicit_von_neumann.f90)

[https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/heateq\\_crank\\_nicolson\\_von\\_neumann.f90](https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/heateq_crank_nicolson_von_neumann.f90)

[https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/heateq\\_comparison\\_01.f90](https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/heateq_comparison_01.f90)

[https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/heateq\\_comparison\\_02.f90](https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/heateq_comparison_02.f90)

[https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/heateq\\_comparison\\_03.f90](https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/heateq_comparison_03.f90)

### Bash script para correr los códigos

[https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/script\\_run.sh](https://github.com/mendzmartin/fiscomp2022/blob/main/lab02/prob02/code/script_run.sh)

## Referencias

Chapman, S. (2007). *Fortran 95/2003 for Scientists & Engineers*. McGraw-Hill Education. [https://books.google.com/books/about/Fortran\\_95\\_2003\\_for\\_Scientists\\_Engineers.html?hl=&id=c8cLDQEACAAJ](https://books.google.com/books/about/Fortran_95_2003_for_Scientists_Engineers.html?hl=&id=c8cLDQEACAAJ)

Chapra, S. C., & Canale, R. P. (2007). *Métodos numéricos para ingenieros*. [https://books.google.com/books/about/M%C3%A9todos\\_num%C3%A9ricos\\_para\\_ingenieros.html?hl=&id=hoH0MAAACAAJ](https://books.google.com/books/about/M%C3%A9todos_num%C3%A9ricos_para_ingenieros.html?hl=&id=hoH0MAAACAAJ)

Landau, R. H., Mejía, M. J. P., Páñez, M. J., Kowallik, H., & Jansen, H. (1997). *Computational Physics*. Wiley-VCH. [https://books.google.com/books/about/Computational\\_Physics.html?hl=&id=MJ3vAAAAMAAJ](https://books.google.com/books/about/Computational_Physics.html?hl=&id=MJ3vAAAAMAAJ)