

# RESOLUCIÓN DE ECUACIONES DIFERENCIALES MEDIANTE INCREMENTOS FINITOS

PAUL ALEJANDRO PEÑAHERRERA VASQUEZ

Semestre 2023-B

Consideremos el siguiente problema

$$\begin{cases} \dot{y}(t, x) - y''(t, x) = f(t, x), & x \in (a, b), a, b \in \mathbb{R}, t \in [0, T], T > 0, \\ y(t, a) = y(t, b) = 0, & t \in [0, T], \\ y(0, x) = y_0(x), & x \in (a, b). \end{cases}$$

Aquí  $\dot{y}$  representa la derivada respecto de  $t$  y  $y''$  las derivadas respecto de  $x$ .

## 1. Escriba un esquema de diferencias finitas en $x$ para este problema.

Vamos a considerar que  $y_i = y(x_i)$  y en el caso de dos varibales consideraremos que  $y_i(t) = y(t, x_i)$ . Recordemos que la derivada de una función se puede aproximar mediante las siguientes ecuaciones:

$$y'(x) \approx \frac{y(x+h) - y(x)}{h} \approx \frac{y(x) - y(x-h)}{h}$$

Por tanto, la segunda derivada de una función se puede escribir como

$$\begin{aligned} y''(x) &\approx \frac{y'(x+h) - y'(x)}{h} \\ &\approx \frac{y(x+h) - y(x)}{h^2} - \frac{y(x) - y(x-h)}{h^2} \\ &\approx \frac{1}{h^2}(y_{i+1} - 2y_i + y_{i-1}) \end{aligned}$$

En nuestro caso, dado que la función  $y$  es de dos variables, tendríamos que

$$y''(t, x) \approx \frac{y(t, x+h) - 2y(t, x) + y(t, x-h)}{h^2}$$

con  $h = \frac{b-a}{n_x}$ , donde  $n_x$  representa el número de particiones que deseamos hacer. Remplazando la ecuación obtenida en nuestra EDP se tiene que

$$\dot{y}(t, x) - \frac{y(t, x+h) - 2y(t, x) + y(t, x-h)}{h^2} \approx f(t, x)$$

Así,

$$\dot{y}(t, x_k) - \frac{y(t, x_{k+1}) - 2y(t, x_k) + y(t, x_{k-1}))}{h^2} \approx f(t, x_k)$$

$$\dot{y}_k(t) - \frac{y_{k+1}(t)}{h^2} + \frac{2}{h^2}y_k(t) - \frac{y_{k-1}(t)}{h^2} \approx f_k(t)$$

Gracias a las condiciones de frontera y la ecuación anterior, vemos que si

$k=1$

$$\dot{y}_1(t) - \frac{y_2(t)}{h^2} + \frac{2}{h^2}y_1(t) \approx f_1(t)$$

$k=2$

$$\dot{y}_2(t) - \frac{y_3(t)}{h^2} + \frac{2}{h^2}y_2(t) - \frac{y_1(t)}{h^2} \approx f_2(t)$$

$\vdots$   
k=n-1

$$\dot{y}_{n-1}(t) + \frac{2}{h^2}y_{n-1}(t) - \frac{y_{n-2}(t)}{h^2} \approx f_{n-1}(t)$$

Por tanto, el sistema anteriore se puede reescribir como

$$\begin{cases} \dot{\vec{y}}_h(t) + A^h \vec{y}(t) = \vec{f}_h(t), \\ \vec{y}_h(0) = \vec{y}_{h,0}. \end{cases} \quad (1)$$

donde

$$\begin{aligned} \vec{y}_h(t) &= (y_1(t), y_2(t), \dots, y_{N-1}(t))^T \\ \vec{f}_h(t) &= (f_1(t), f_2(t), \dots, f_{N-1}(t))^T \\ \dot{\vec{y}}_h(t) &= (\dot{y}_1(t), \dot{y}_2(t), \dots, \dot{y}_{N-1}(t))^T \\ \vec{y}_{h,0} &= (y_1(0), y_2(0), \dots, y_{N-1}(0))^T \end{aligned}$$

y

$$A^h = \begin{pmatrix} \frac{2}{h^2} & -\frac{1}{h^2} & 0 & \cdots & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -\frac{1}{h^2} & \frac{2}{h^2} \end{pmatrix}$$

2. Utilice el esquema explícito de Euler para el problema de Cauchy que se obtiene luego de la discretización espacial.

Despejando  $\dot{\vec{y}}_h(t)$  y tomando  $\vec{F}(t) = \vec{f}_h(t) - A^h \vec{y}(t)$  podemos reescribir el sistema (1) como

$$\begin{cases} \dot{\vec{y}}_h(t) = \vec{F}(t) \\ \vec{y}_h(0) = \vec{y}_{h,0}. \end{cases} \quad (2)$$

el cual tiene la forma de un problema de Cauchy. Además, nuevamente de la definición de derivada, para un  $\alpha$  suficientemente vemos que

$$\frac{\vec{y}_h(t_j + \alpha) - \vec{y}_h(t_j)}{\alpha} = \vec{F}(t_j), \quad \forall j = 0, \dots, n_\alpha$$

por tanto

$$\vec{y}_h(t_{l+1}) = \vec{y}_h(t_j) + \alpha \left( \vec{f}_h(t_j) - A^h \vec{y}(t_j) \right) \quad \forall j = 0, \dots, n_\alpha$$

de donde cada componente de  $\vec{y}_h(t_{j+1})$  se puede ver como

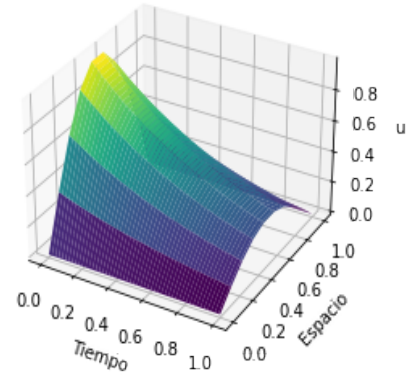
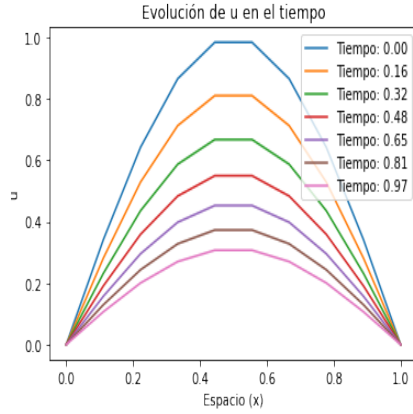
$$\begin{aligned} y_{i+1}(x_j) &= \frac{\alpha}{h^2} (y_i(x_{j+1}) - 2y_i(x_j) + y_i(x_{j-2})) + y_i(x_j) + \alpha f_i(x_j) \\ &= \frac{\alpha}{h^2} (y_i(x_{j+1}) + y_i(x_{j-1})) + \left( 1 - \frac{2\alpha}{h^2} \right) y_i(x_j) + \alpha f_i(x_j) \end{aligned}$$

para cada  $i \in [1, \dots, n_\alpha]$  y  $j \in [1, \dots, n_x]$ .

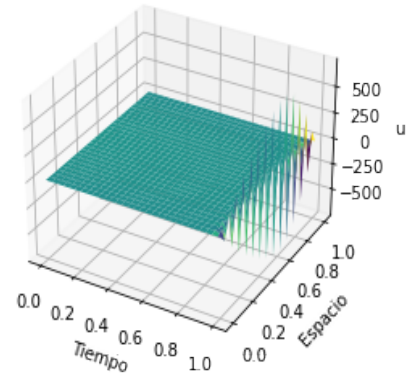
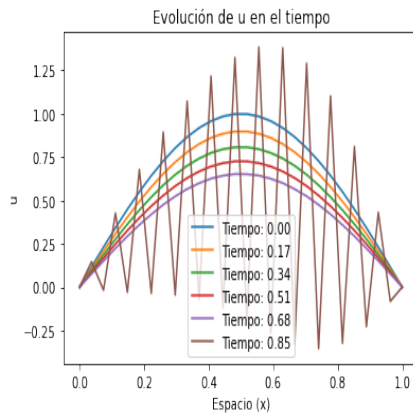
3. Implemente computacionalmente el problema discretizado que se obtiene luego de efectuados los puntos 1 y 2. Realice una experimentación numérica con:

a)  $a = 0, b = 1, T = 1, f(t, x) = 0$  y  $y_0(x) = \sin(\pi x)$ .

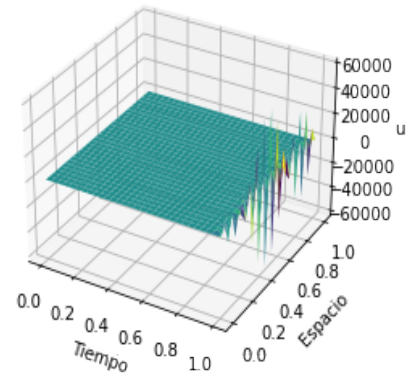
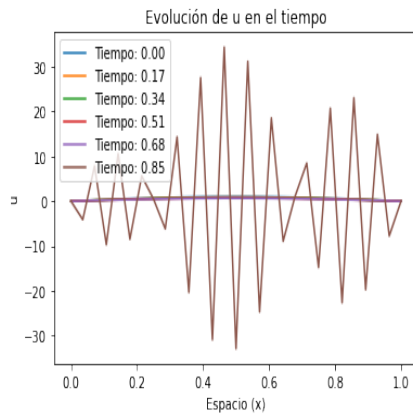
Para  $n_x = 10$  y  $n_\alpha = 100$  se tiene que el algoritmo nos da una buena aproximación y funciona correctamente.



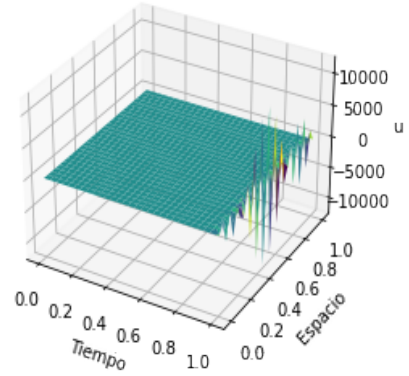
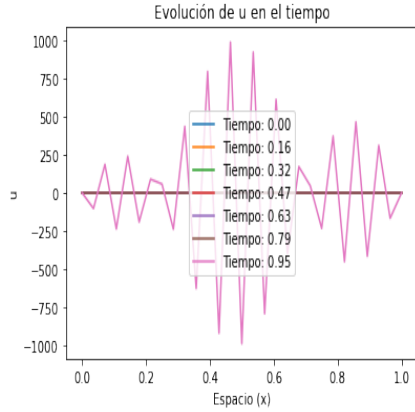
Para  $n_x = 28$  y  $n_\alpha = 60$  se aprecia que el algoritmo comienza a dar signos de inestabilidad.



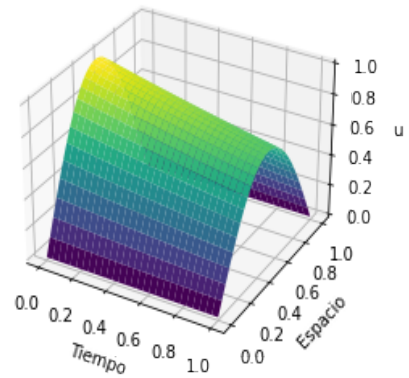
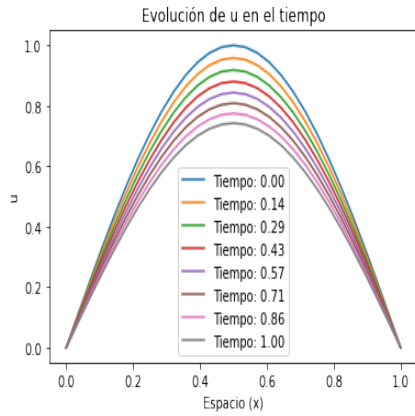
Si solamente aumentamos una unidad al eje espacial, es decir si  $n_x = 29$  y  $n_\alpha = 60$  el algoritmo es totalmente inestable.



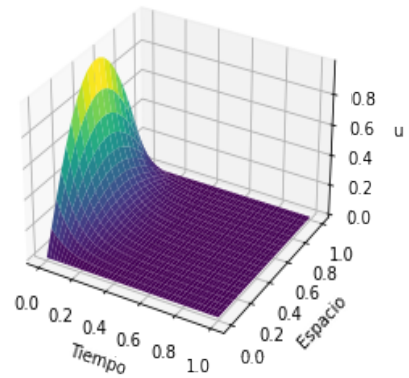
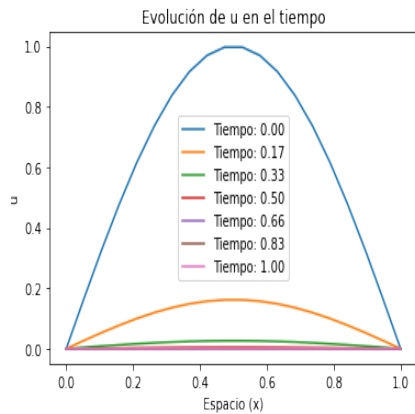
Lo cual no hace preguntarnos, que pasará si aumentamos ahora variamos el parámetro de tiempo, para ello veamos que si  $n_x = 28$  y  $n_\alpha = 58$  el algoritmo sigue siendo totalmente inestable.



Ahora si seguimos disminuyendo las particiones del eje temporal, es decir  $n_x = 28$  y  $n_\alpha = 28$  el algoritmo recupera su estabilidad.



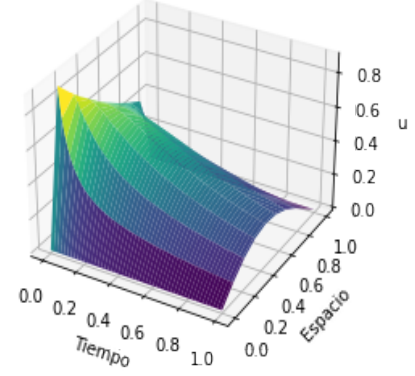
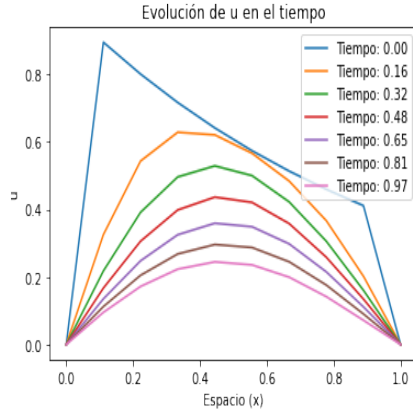
Finalmente si aumentamos bastante las particiones del eje temporal, es decir  $n_x = 29$  y  $n_\alpha = 1000$  el algoritmo sigue dandonos una respuesta aceptable y se nota claramente que alcanza la estabilidad.



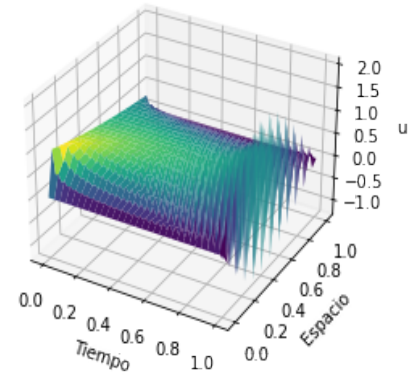
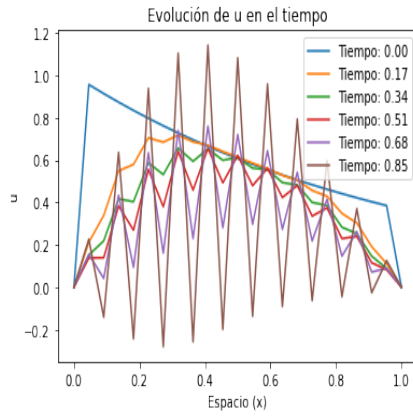
**Observación.** Si bien la estabilidad depende de ambas variables, depende mucho más de  $n_x$ , pues leves variaciones causan grandes irregularidades, mientras que al variar  $n_\alpha$  el algoritmo permite más flexibilidad para mantener su estabilidad.

b)  $a, b \in \mathbb{R} (a = 0, b = 1), T = 1, f(t, x) = t \cos(\pi x)$  y  $y_0(x) = e^{-x}$

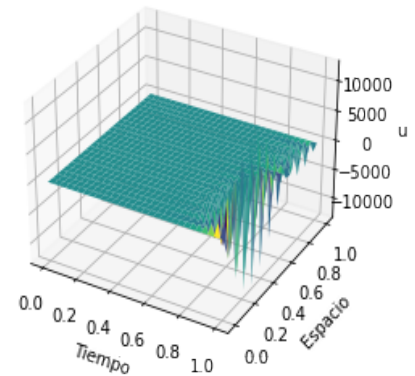
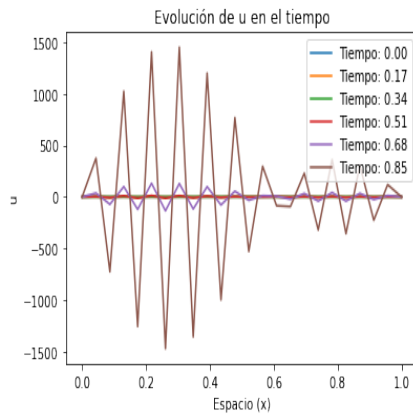
Para  $n_x = 10$  y  $n_\alpha = 100$  se tiene que el algoritmo nuevamente nos da una buena aproximación y funciona correctamente.



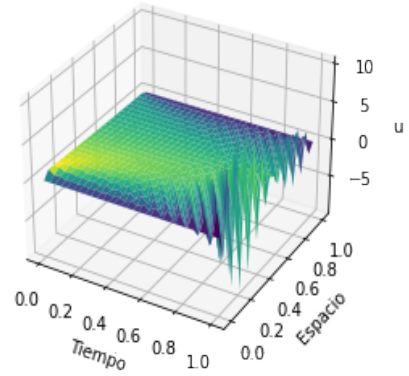
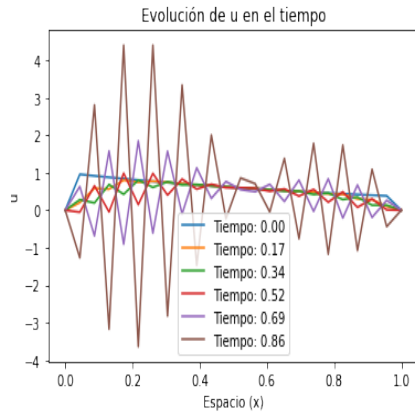
Para  $n_x = 23$  y  $n_\alpha = 60$  se aprecia que el algoritmo comienza a dar signos de inestabilidad, sin embargo todavía es apreciable su solución.



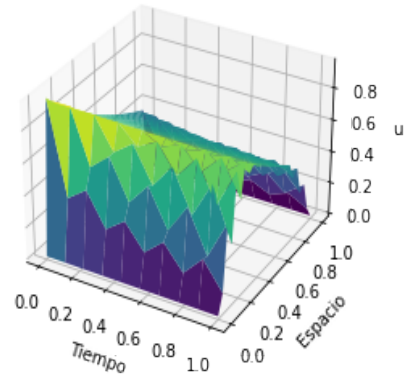
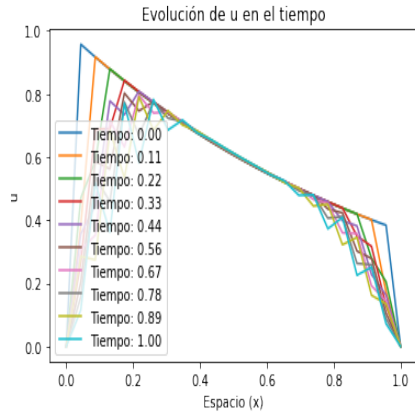
Otra vez, notamos que si aumentamos una unidad al eje espacial, es decir si  $n_x = 24$  y  $n_\alpha = 60$  el algoritmo es totalmente inestable y se pierde la buena aproximación.



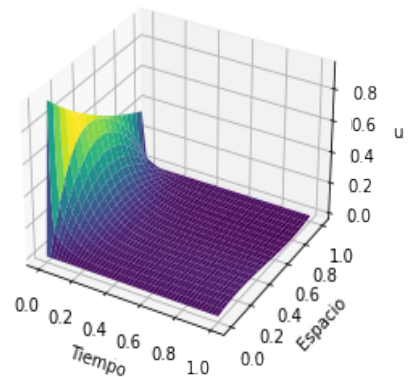
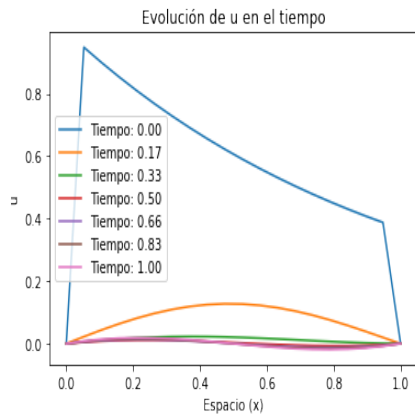
Análogo al anterior ejemplos si disminuimos las particiones temporales,  $n_x = 24$  y  $n_\alpha = 30$ , el algoritmo empieza a mejorar su estabilidad, sin embargo el resultado sigue siendo ineficiente.



Ahora si seguimos disminuyendo las particiones temporales de tal forma que sea menor al eje espacial, es decir  $n_x = 24$  y  $n_\alpha = 10$ , el algoritmo recupera su estabilidad y da buenas soluciones aproximadas.



Finalmente, si aumentamos nuevamente las particiones temporales de manera "exagerada", es decir  $n_x = 20$  y  $n_\alpha = 1000$ , el algoritmo sigue dandonos una respuesta aceptable y se nota claramente es estable.



**Observación.** Para el buen funcionamiento del algoritmo, nuevamente se llega a las observaciones anteriores en cuanto a la relación de las particiones temporales y espaciales, por lo que podemos deducir finalmente, es que si deseamos una buena solución y estabilidad, las variables  $n_x$  y  $n_\alpha$  deben ser bastante cercanas o en todo caso iguales.

4. **Repita los numerales 2 y 3, pero considerando un esquema absolutamente convergente para el problema de Cauchy. Tome  $T = 1, 10, 100$ .**

Vamos a ocupar el método euler implícito el cual sabemos que es  $\mathcal{A}$ -estable, el cual está dado por

$$y_{i+1}(x_j) = y_i(x_j) + \alpha \dot{y}_{i+1}(x_j).$$

Pero recordemos que del literal 1 deducimos que

$$\dot{y}_i(x_j) - \frac{y_i(x_{j+1})}{h^2} + \frac{2}{h^2}y_i(x_j) - \frac{y_i(x_{j-1}))}{h^2} = f_i(x_j)$$

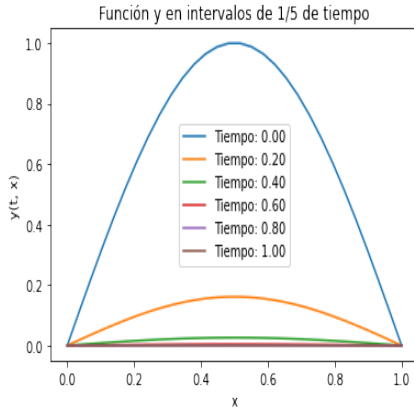
Por tanto, combinando ambas ecuaciones, obtenemos el siguiente método

$$y_{i+1}(x_j) = y_i(x_j) + \alpha \left( \frac{(y_{i+1}(x_{j+1}) + y_{i+1}(x_{j-1})))}{h^2} - \frac{2}{h^2}y_{i+1}(x_j) + f_{i+1}(x_j) \right)$$

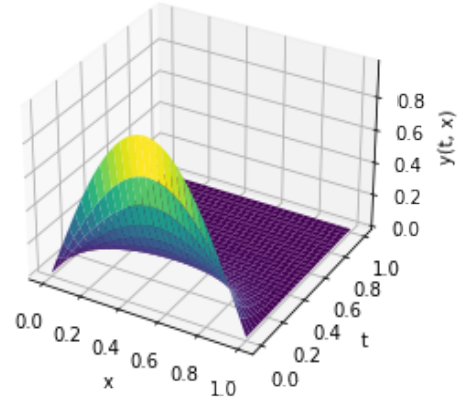
Dado que el método es  $\mathcal{A}$ -estable, entonces solo analizaremos el problema en los puntos donde el método anterior tenía dificultades de estabilidad. Consideremos

- a)  $a = 0, b = 1, f(t, x) = 0$  y  $y_0(x) = \sin(\pi x)$ .

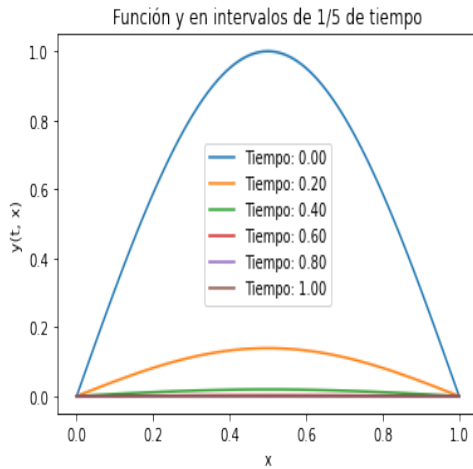
Cuando  $T = 1, n_x = 29$  y  $n_\alpha = 60$  se tiene que



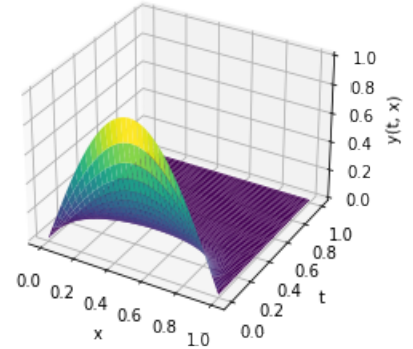
Solución de la ecuación diferencial



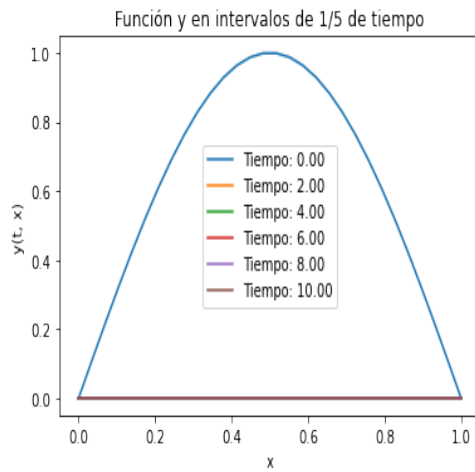
Cuando  $T = 1, n_x = 100$  y  $n_\alpha = 9000$  se tiene que



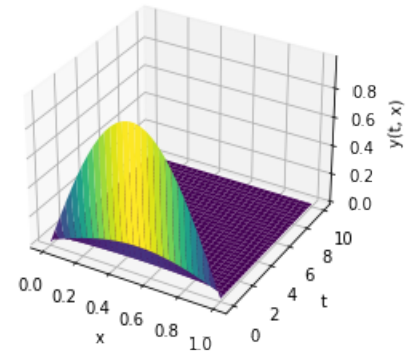
Solución de la ecuación diferencial



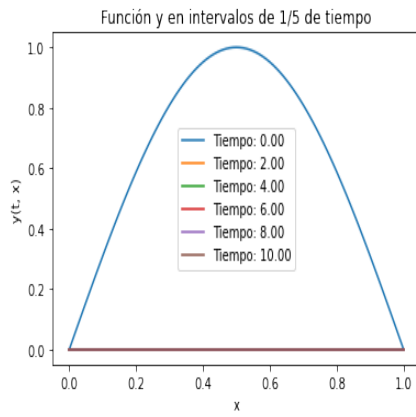
Cuando  $T = 10$ ,  $n_x = 29$  y  $n_\alpha = 60$  se tiene que



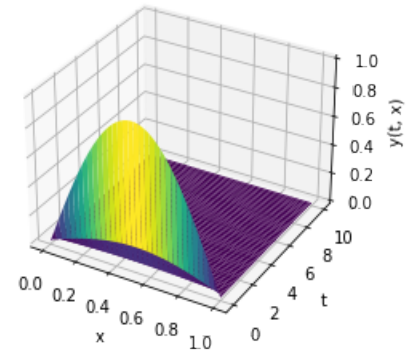
Solución de la ecuación diferencial



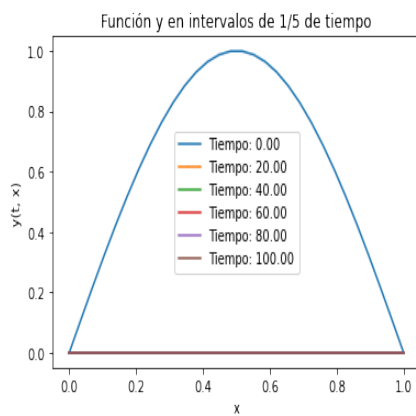
Cuando  $T = 10$ ,  $n_x = 100$  y  $n_\alpha = 9000$  se tiene que



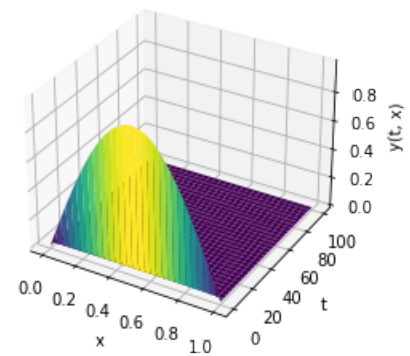
Solución de la ecuación diferencial



Cuando  $T = 100$ ,  $n_x = 29$  y  $n_\alpha = 60$  se tiene que

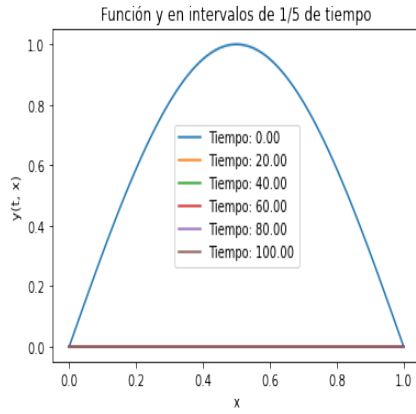


Solución de la ecuación diferencial

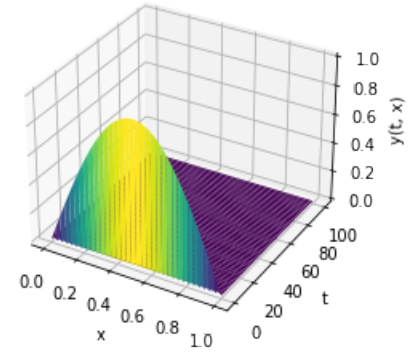




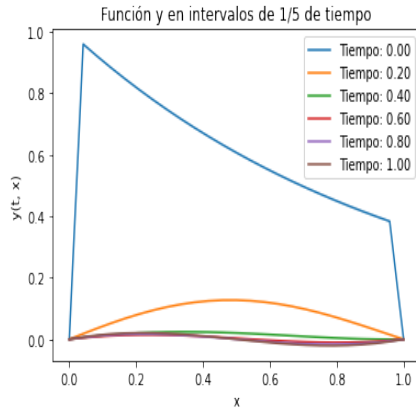
Cuando  $T = 100$ ,  $n_x = 100$  y  $n_\alpha = 9000$  se tiene que



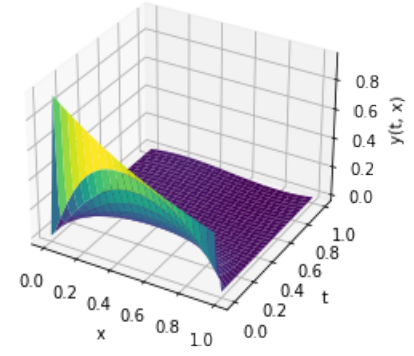
Solución de la ecuación diferencial



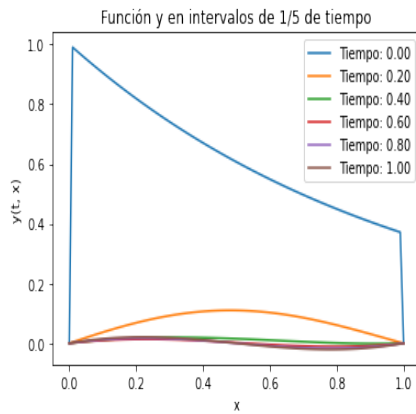
b)  $a, b \in \mathbb{R}(a = 0, b = 1), T = 1, f(t, x) = t \cos(\pi x)$  y  $y_0(x) = e^{-x}$  Cuando  $T = 1$ ,  $n_x = 24$  y  $n_\alpha = 60$  se tiene que



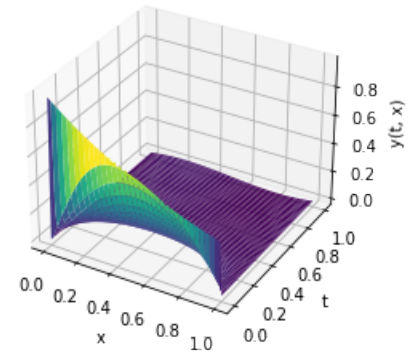
Solución de la ecuación diferencial



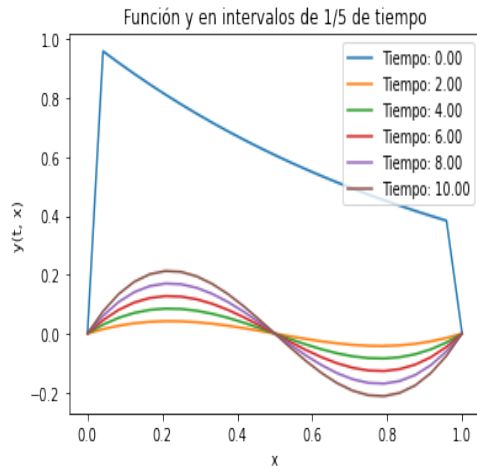
Cuando  $T = 1$ ,  $n_x = 100$  y  $n_\alpha = 9000$  se tiene que



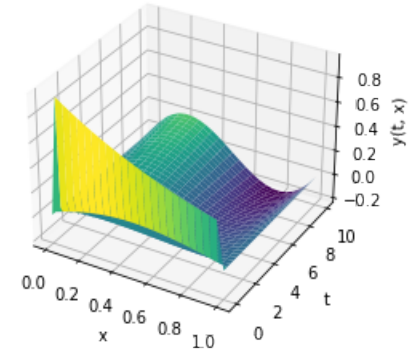
Solución de la ecuación diferencial



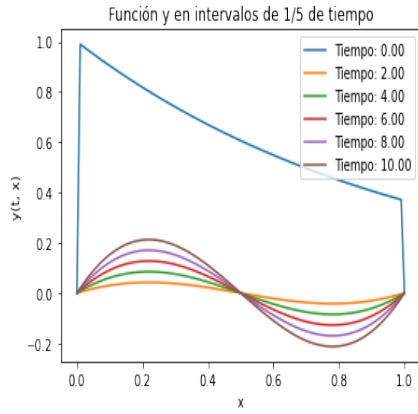
Cuando  $T = 10$ ,  $n_x = 24$  y  $n_\alpha = 60$  se tiene que



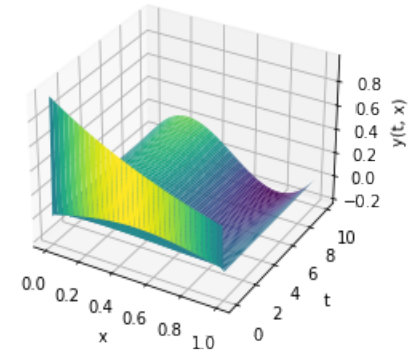
Solución de la ecuación diferencial



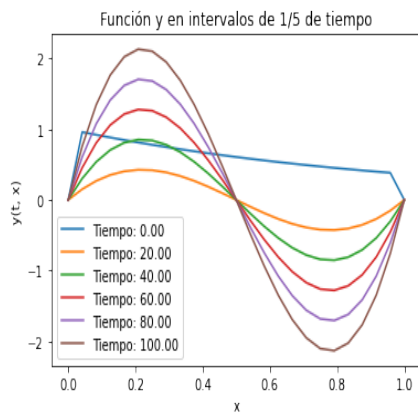
Cuando  $T = 10$ ,  $n_x = 100$  y  $n_\alpha = 9000$  se tiene que



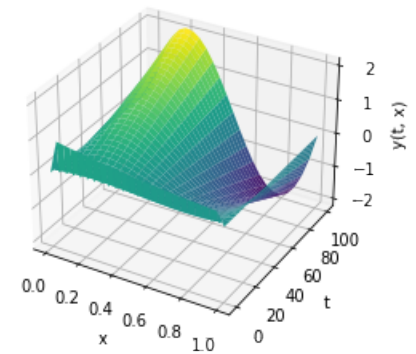
Solución de la ecuación diferencial



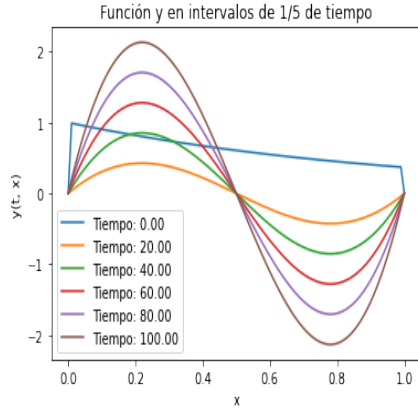
Cuando  $T = 100$ ,  $n_x = 24$  y  $n_\alpha = 60$  se tiene que



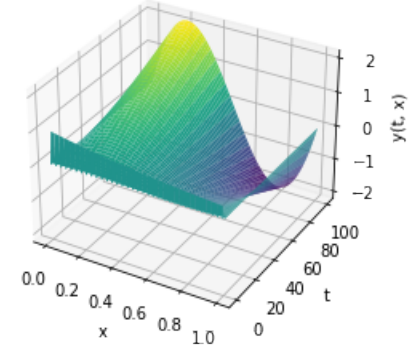
Solución de la ecuación diferencial



Cuando  $T = 100$ ,  $n_x = 100$  y  $n_\alpha = 9000$  se tiene que



Solución de la ecuación diferencial



**Observación.** Notamos que en ambos casos, en los puntos en los cuales el algoritmo inicial fallaba, este no tiene ningún problema de estabilidad. Además, también podemos apreciar que si incrementamos en gran medida la distancia entre  $n_x$  y  $n_\alpha$ , el algoritmo no tiene problemas de rendimiento. Es más, la solución aproximada se vuelve cada vez más fina, es decir, se aproxima de mejor manera a la solución real.

5. Finalmente, repita los numerales 1,3 y 4, para el problema siguiente

$$\begin{cases} \dot{y}(t, x) - y''(t, x) = f(t, x), & x \in (a, b), a, b \in \mathbb{R}, t \in [0, T], T > 0, \\ y'(t, a) = y'(t, b) = 0, & t \in [0, T], \\ y(0, x) = y_0(x), & x \in (a, b). \end{cases}$$

Recordemos que inicialmente llegamos a la siguiente expresión

$$y_{i+1}(x_j) = \frac{\alpha}{h^2} (y_i(x_{j+1}) + y_i(x_{j-1})) + \left(1 - \frac{2\alpha}{h^2}\right) y_i(x_j) + \alpha f_i(x_j) \quad (3)$$

para cada  $i \in [1, \dots, n_\alpha]$  y  $j \in [1, \dots, n_x]$ . Ahora veamos que sucede si aplicamos nuestra condición de frontera, sabemos que

$$y'_i(x_j) = \frac{y_i(x_{j+1}) - y_i(x_{j-1}))}{2h} = 0$$

por tanto, vemos que

$$y_i(x_{j+1}) = y_i(x_{j-1})$$

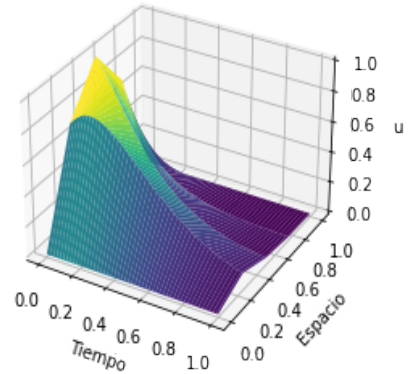
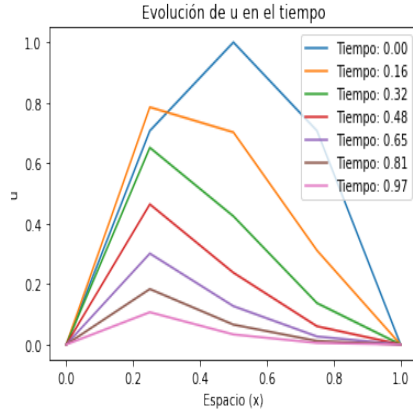
y reemplazando en (3), obtenemos que

$$y_{i+1}(x_j) = \frac{2\alpha}{h^2} y_i(x_{j+1}) + \left(1 - \frac{2\alpha}{h^2}\right) y_i(x_j) + \alpha f_i(x_j)$$

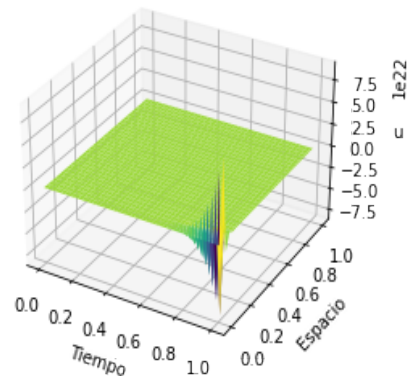
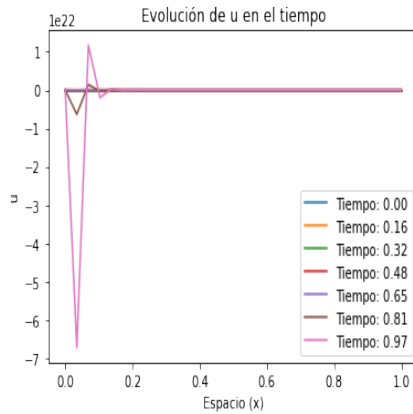
**Implemente computacionalmente el problema discretizado y realice una experimentación numérica con:**

a)  $a = 0, b = 1, T = 1, f(t, x) = 0$  y  $y_0(x) = \sin(\pi x)$ .

Para  $n_x = 5$  y  $n_\alpha = 100$  se tiene que el algoritmo funciona correctamente, sin embargo al ser tan bajas las particiones espaciales, no nos da una buena aproximación

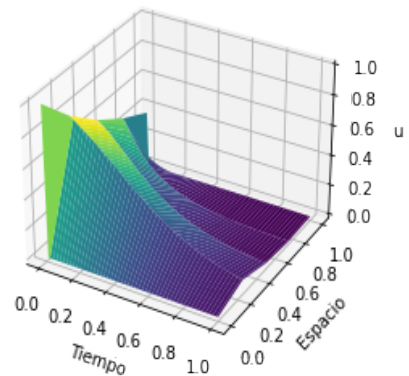
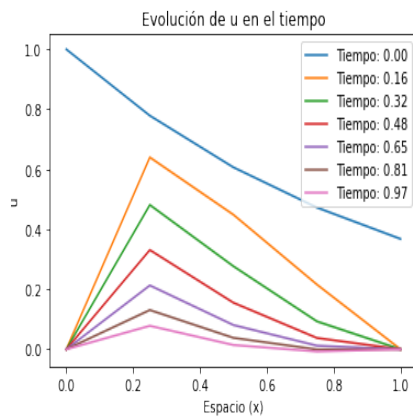


Para  $n_x = 30$  y  $n_\alpha = 100$  se tiene que el algoritmo funciona da problemas de estabilidad muy severos.

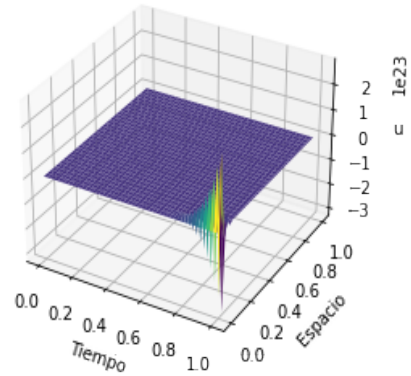
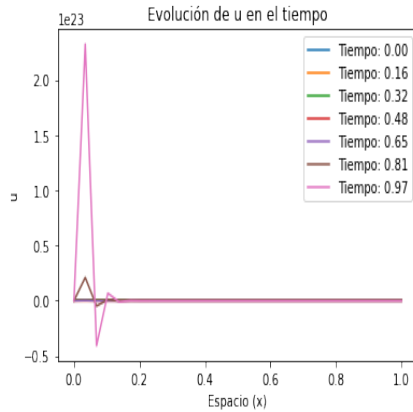


b)  $a, b \in \mathbb{R}(a = 0, b = 1), T = 1, f(t, x) = t \cos(\pi x)$  y  $y_0(x) = e^{-x}$

Para  $n_x = 5$  y  $n_\alpha = 100$  nuevamente el algoritmo nos da una aproximación "burda" del la solución debido a la baja cantidad de particiones.



Para  $n_x = 30$  y  $n_\alpha = 100$  vemos que nuevamente el algoritmo da problemas de estabilidad irresolubles.

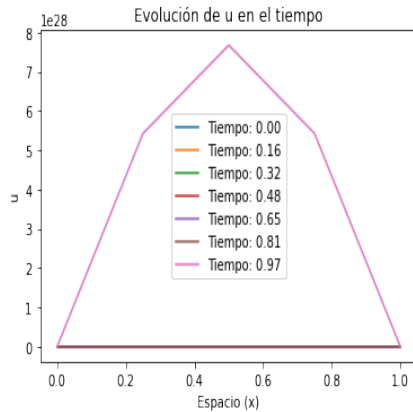


**Observación.** Notamos que el funcionamiento del algoritmo no es muy eficiente y la estabilidad aumenta considerablemente debido a las condiciones de frontera de Neumann.

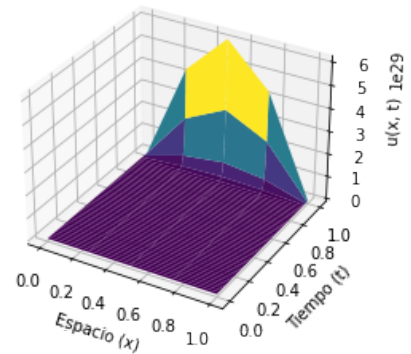
Considerando un esquema absolutamente convergente para el problema de Cauchy. Tome  $T = 1, 10, 100$ . Igual que antes, consideraremos el método de euler implícito, el cual sabemos que es  $\mathcal{A}$ -estable.

a)  $a = 0, b = 1, f(t, x) = 0$  y  $y_0(x) = \sin(\pi x)$ .

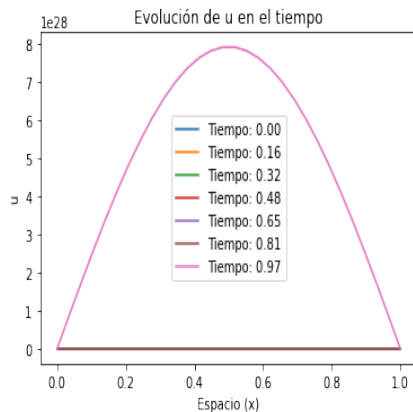
Cuando  $T = 1$ ,  $n_x = 5$  y  $n_\alpha = 100$  se tiene que el algoritmo funciona correctamente. Cuando



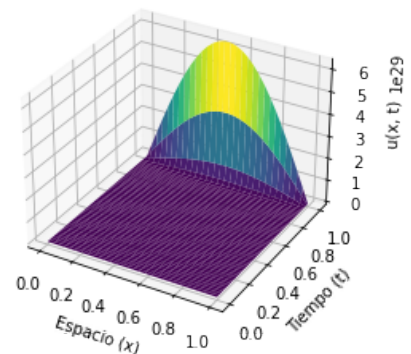
Evolución de la ecuación de difusión



$T = 1$ ,  $n_x = 30$  y  $n_\alpha = 100$ , vemos que en el punto en el cual antes se presentaba inestabilidad, ahora ya no existe ese problema

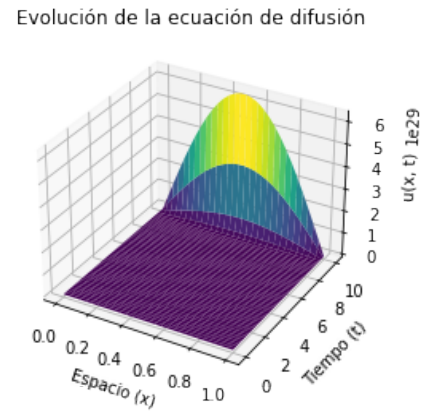
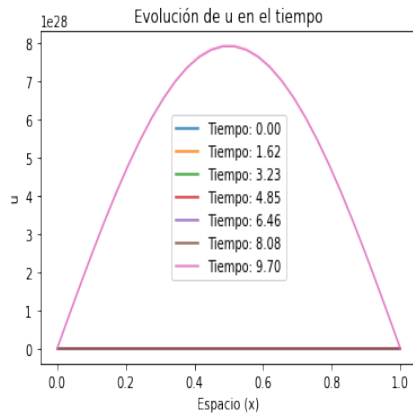


Evolución de la ecuación de difusión

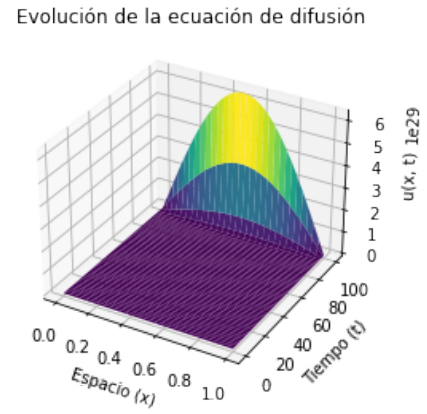
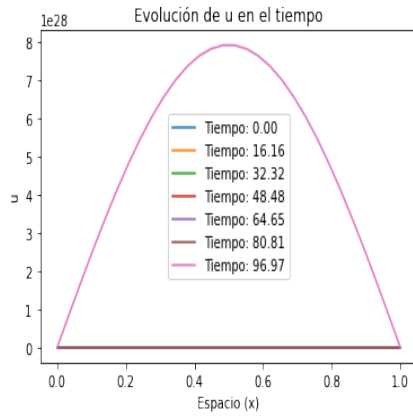


Ahora, analicemos solamente en los puntos donde antes presentaba inestabilidad el método y aumentemos el tiempo.

Cuando  $T = 10$ ,  $n_x = 30$  y  $n_\alpha = 100$

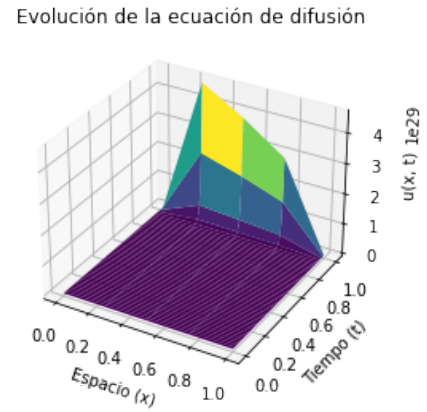
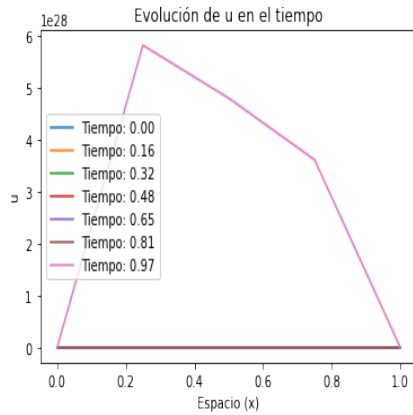


Cuando  $T = 100$ ,  $n_x = 30$  y  $n_\alpha = 100$

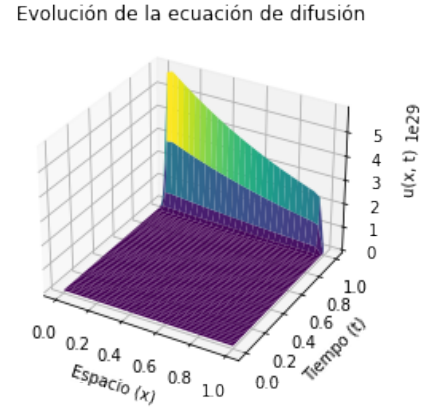
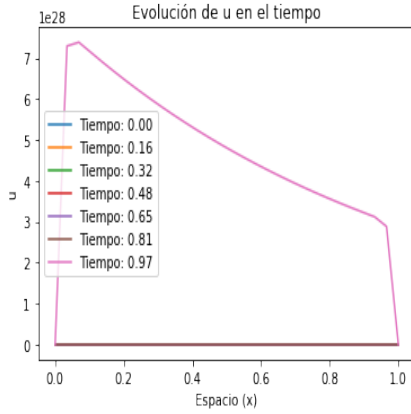


b)  $a, b \in \mathbb{R} (a = 0, b = 1), T = 1, f(t, x) = t \cos(\pi x)$  y  $y_0(x) = e^{-x}$

Cuando  $T = 1$ ,  $n_x = 5$  y  $n_\alpha = 100$  se tiene que el algoritmo funciona correctamente, pero aun se puede mejorar la aproximación.

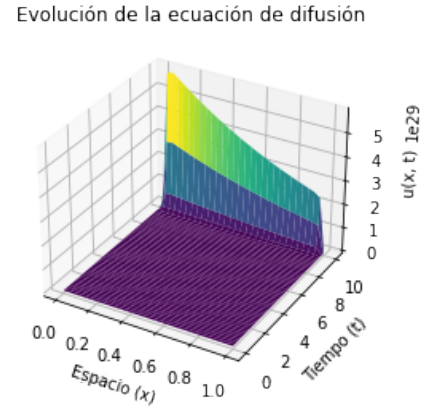
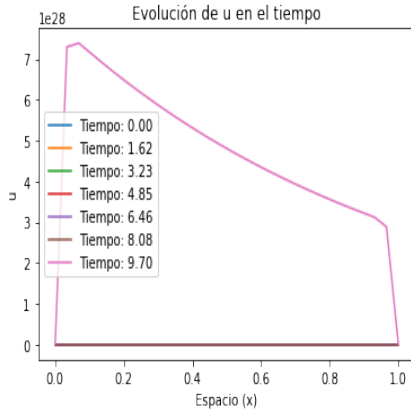


Cuando  $T = 1$ ,  $n_x = 30$  y  $n_\alpha = 100$ , vemos que en el punto en el cual antes se presentaba inestabilidad, ahora ya no existe ese problema

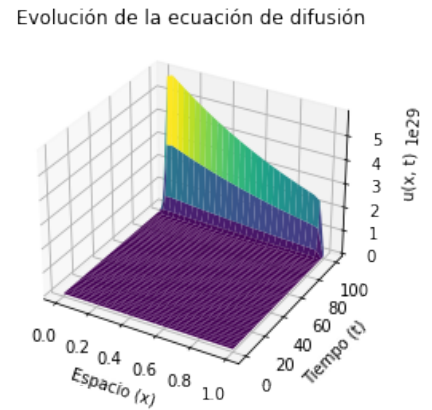
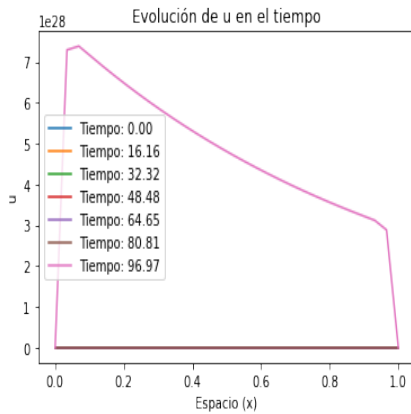


Nuevamente, analizaremos solamente los puntos donde antes presentaba inestabilidad el método y aumentaremos el tiempo.

Cuando  $T = 10$ ,  $n_x = 30$  y  $n_\alpha = 100$



Cuando  $T = 100$ ,  $n_x = 30$  y  $n_\alpha = 100$



**Observación.** Los métodos de diferencias finitas son muy útiles para resolver ecuaciones diferenciales parciales, sin embargo presentan inconvenientes de estabilidad en algunos casos. Estos problemas pueden ser resueltos combinando métodos  $\mathcal{A}$ -estables como el método de euler implícito o el BDF2.