

Practica 3

González Borja, Miguel
Illescas Arizti, Rodrigo
Meyer Mañón, Juan Carlos
Rodríguez Orozco, Alejandro

18 / Mayo / 2018

Introducción

A continuación los ejercicios de la práctica. Todos fueron realizados en Python 3.6 utilizando los paquetes *numpy* y *scipy.optimize*. También se incluye un cuaderno de IPython para cada ejercicio dentro de la carpeta Ejercicios, junto con los scripts *.py* para obtener los resultados. Cada uno de estos utiliza los siguientes imports:

```
In [1]: import sys
        sys.path.append('../')
        import latexStrings as ls
        from IPython.display import Latex
        import numpy as np
        import matplotlib.pyplot as plt
        from mpl_toolkits.mplot3d import Axes3D
        import pdesolver
```

Se puede encontrar un repositorio del proyecto [aquí](#).

Ejercicio 1

Dada la ecuación del calor, tenemos el PVIF:

$$u_t = \frac{1}{\pi} u_{xx} \quad (1)$$

Con las condiciones:

$$\begin{cases} u(x, 0) = \sin(\pi x) & \text{para } 0 \leq x \leq 1 & (2) \\ u(0, t) = 0 & \text{para } 0 \leq t \leq 1 & (3) \\ u_x(1, t) = -\pi e^{-\pi x} & \text{para } 0 \leq t \leq 1 & (4) \end{cases}$$

Notamos que (2) nos da las condiciones iniciales para x , mientras que para t tenemos condiciones de frontera (3) Dirichlet en la izquierda y (4) Neumann por la derecha. Además, sabemos que la solución exacta es:

$$u(t, x) = e^{-\pi t} \sin(\pi x) \quad (5)$$

Entonces podemos escribir nuestro problema como:

```
In [1]: u = lambda x, t: np.exp(-np.pi*t)*np.sin(np.pi*x)
eq= {}
eq['D'] = 1/np.pi
eq['ic'] = lambda x : np.sin(np.pi*x)
eq['bcL'] = lambda t : 0
eq['bcR'] = lambda t : -np.pi*np.exp(-np.pi*t)
Ix = [0, 1]
It = [0, 1]
```

Ahora verifiquemos el maximo paso del tiempo que podemos dar dado $h = 1/20$. Tomando únicamente las condiciones iniciales y de frontera tipo Dirichlet, podemos escribir el esquema de forma matricial de la siguiente forma:

$$W^{j+1} = AW^j + S^j$$

donde:

$$A = (1 - 2\sigma)I_{m-1} + \sigma T \quad \sigma = \frac{1}{\pi} \frac{k}{h^2}$$

$$T = \begin{pmatrix} 0 & 1 & \dots & 0 \\ 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \dots & 1 & 0 \end{pmatrix} \in \mathbb{R}^{(M-1) \times (M-1)}$$

$$S^j = \begin{pmatrix} \sigma w_0^j \\ 0 \\ \vdots \\ 0 \\ \sigma w_M^j \end{pmatrix} \in \mathbb{R}^{M-1}$$

Tomando únicamente las condiciones iniciales y de frontera tipo Dirichlet, para que el esquema sea estable se debe cumplir la condición :

$$\sigma = \frac{1}{\pi} \frac{k}{h^2} \leq \frac{1}{2}$$

En este caso $h = 1/20$ entonces :

$$\sigma = \frac{1}{\pi} \frac{k}{h^2} = \frac{400k}{\pi} \leq \frac{1}{2}$$

$$\Rightarrow k \leq \frac{\pi}{800}$$

Por lo tanto el máximo paso estable teórico en el tiempo es $k = \frac{\pi}{800}$. Esto se debe a que $\rho(A) = \max |\lambda_i| \leq 1$ para que sea estable, donde λ son los eigenvalores de A . Por lo tanto se cumple la definición de estabilidad en el sentido de Von Neumann.

Ahora apliquemos el metodo explicito con 20 pasos en el espacio ($h = 1/20$) y 256 pasos en el tiempo ($k = 1/256$), y veamos el maximo de los errores al tiempo $T = 1$:

```

In [2]: M = 20
        N = 256

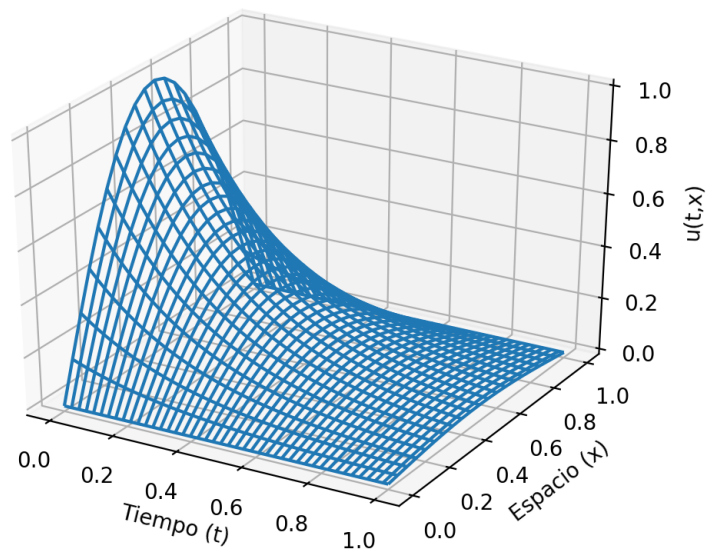
        W, X, T = pdesolver.explicitHeat(eq, Ix, It, M, N)
        U = np.array([[u(x,t) for t in T] for x in X])
        Error = max(abs(U[:,256]-W[:,256]))

```

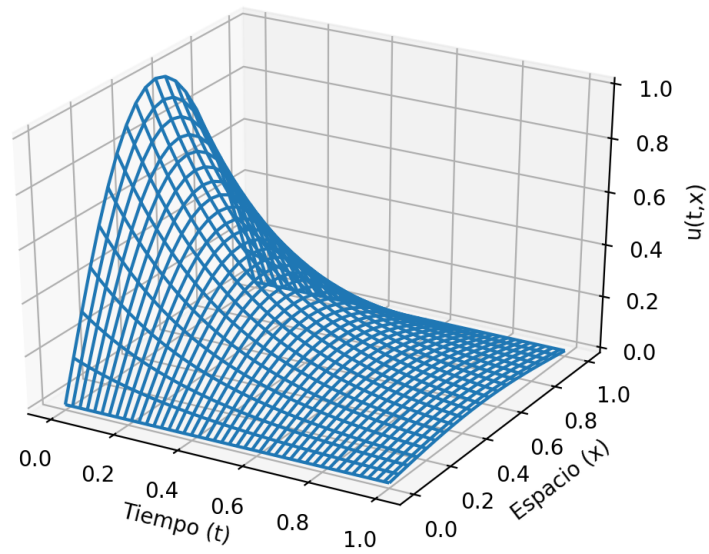
Maximo Error en $T = 1$: 0.00128585109347

Finalmente, veamos como se ven graficamente la solucion exacta, asi como la aproximada:

Solucion Exacta



Solucion Metodo Explicito

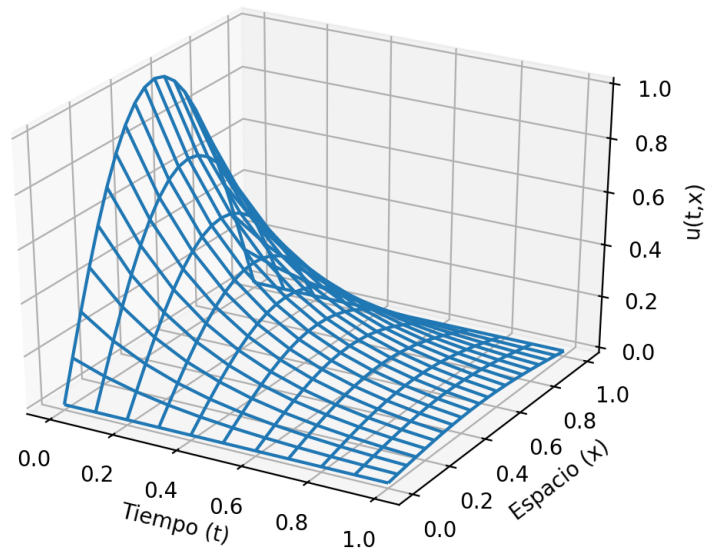


Ejercicio 2

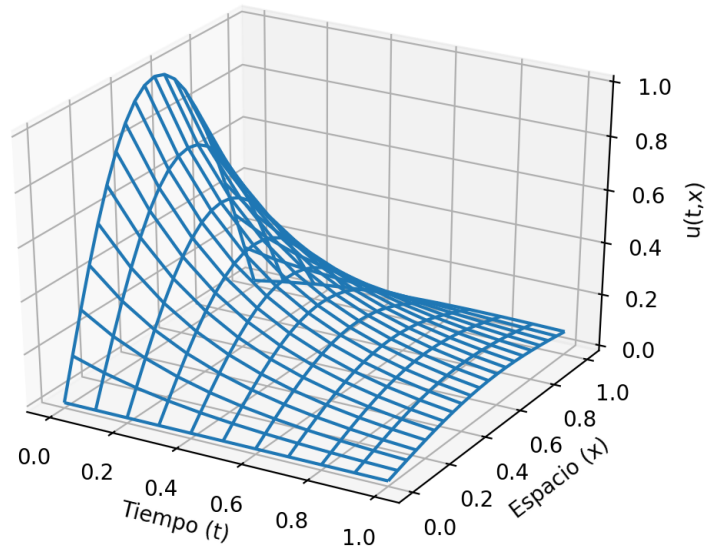
Dado el PVIF (1) planteado anteriormente, veamos el comportamiento del metodo implicito para su aproximacion. Primero observemos una grafica de la aproximacion usando 20 pasos en el espacio ($h = 1/20$) y 10 pasos en el tiempo ($k = 1/10$):

```
In [1]: M = 20  
        N = 10  
        W, X, T = pdesolver.implicitHeat(eq, Ix, It, M, N)
```

Solucion Exacta



Solucion Metodo Implicito



Vemos que con los mismos pasos en el tiempo, pero mas de 20 veces menos los pasos en el espacio, el metodo implicito obtiene una aproximacion comparable a la del metodo explicito. Ahora veamos como varia el error al considerar los pasos $k = \{1/16, 1/64, 1/256\}$ y $h = \{1/10, 1/20, 1/40\}$:

h	k	Error	erc
1/10	1/16	0.05783108269993153	NaN
1/20	1/64	0.014939536443073914	1.952709728546003
1/40	1/256	0.0037665127468114214	1.987834066736966

En la teoria, el error se comporta como:

$$Error(h, k) = O(h^2 + k)$$

Podemos ver a k como función de h de la siguiente forma:

$$k(h) = \frac{25}{4}h^2$$

Sustituyendo la expresión anterior en el error:

$$Error(h, k) = O(h^2 + k) = O(h^2 + \frac{25}{4}h^2) \approx O(h^2)$$

Por otro lado, si calculamos el “erc” (*estimated rate of convergence*) dado por :

$$erc_i := \frac{\ln(Error(h_i)/Error(h_{i+1}))}{\ln(h_i/h_{i+1})}$$

con $i \in \{1, 2\}$ debe ser $erc \approx 2$ dado que $Error(h) = O(h^2)$ es de orden 2. En la tabla observamos que en efecto erc es aproximadamente 2. Por lo tanto, en este caso el error en función de h , se reduce de manera cuadrática, entonces nuestros resultados son consistentes con la teoría.

Ahora consideremos los pasos $k = h = \{1/10, 1/20, 1/40\}$:

h	k	Error	erc
1/10	1/10	0.08096946676799946	NaN
1/20	1/20	0.03789343824335091	1.0954299285102875
1/40	1/40	0.018281192075090623	1.0515878997074606

Podemos observar que:

$$Error(h_2, k_2) = 0.03789343824335091 \approx \left(\frac{1}{2}\right) 0.08096946676799946 = \left(\frac{1}{2}\right) Error(h_1, k_1)$$

$$Error(h_3, k_3) = 0.018281192075090623 \approx \left(\frac{1}{2}\right) 0.03789343824335091 = \left(\frac{1}{2}\right) Error(h_2, k_2)$$

Como se mencionó anteriormente, el error se comporta de la siguiente manera:

$$Error(h, k) = O(h^2 + k)$$

En este caso, como $h_i = k_i$ entonces:

$$Error(h, k) = O(h^2 + k) = O(h^2 + h) \approx O(h) := Error(k)$$

Por lo tanto, para $h = k$, teóricamente el error se comporta de manera lineal dado que $Error(h) = O(h)$ es de orden 1. En efecto, en la tabla observamos que el *erc* se acerca a 1.

En conclusión, el error en función de h dado $h = k$, $Error(h) = O(h)$, se reduce de manera lineal, entonces nuestros resultados son consistentes con la teoría.

Finalmente tomemos los pasos $h = 1/10, k = \{1/25, 1/50, 1/100, 1/200\}$:

k	Error	E_i/E_{i-1}	erc
1/25	0.043252578458086034	NaN	NaN
1/50	0.02983931506749922	0.6898852306901205	0.5355717197880779
1/100	0.022968343027896996	0.7697342575035833	0.37756763811094635
1/200	0.019491123414336453	0.8486081643182896	0.23682953626242717

Recordemos que si, $h^2 < k$:

$$Error(h, k) = O(h^2 + k) \approx O(k)$$

Esto indica que en teoría el error debería decrecer linealmente con k , o bien, al duplicar el número de pasos en el tiempo esperamos que el error se reduzca a la mitad.

Sin embargo, en este ejemplo notamos que el error no parece decrecer linealmente. En los primeros pasos, la reducción del error es consistente con la teoría, pero a partir de $k = 1/100$ esto no es cierto. Aún más, parece que $Error(h, k_i) / Error(h, k_{i-1})$ se acerca a 1 cada iteración. ¿A qué se debe esto? ¿Acaso la teoría está mal?

Todo se explica al considerar el valor de h . Si se tomara un paso en el espacio extremadamente pequeño, como $h = \frac{1}{100}$, se puede comprobar que, efectivamente, el error se reduce aproximadamente a la mitad en cada iteración, ya que h^2 es insignificante comparado con k .

k	Error	E_i/E_{i-1}	erc
1/25	0.027446399165031332	NaN	NaN
1/50	0.014027218685576865	0.5110768301966745	0.9683879070281181
1/100	0.007151869447247056	0.5098565587061664	0.9718366735889642
1/200	0.0036721007159067343	0.513446273452352	0.9617147727334843

Vemos que en efecto, el erc está alrededor de 1, lo cual implica $Error(h, k_i) \approx O(k_i)$

Ahora, regresando al ejemplo original, observamos que $h^2 = \frac{1}{100}$, y, aun que inicialmente $k_1 = \frac{1}{25} \ll h^2$, después de dos iteraciones tenemos que $k_3 = \frac{1}{100} = h^2$, por lo que ya no es cierto que $Error(h, k) = O(h^2 + k) \approx O(k)$

Para las iteraciones consecuentes se cumple que $k_i < h^2$, por lo que seguir disminuyendo la magnitud de el paso en el tiempo tiene poco efecto sobre el error máximo, ya que en estos casos $h^2 = 1/100$ es el factor que afecta al error en mayor magnitud y al mantener $h = 1/10$ constante, el error disminuye poco ante cambios en k . Por esto, la razón de los errores consecutivos se acerca a 1 cada paso. Por esto podemos concluir que el ejemplo, a pesar de que parezca contradecir la teoría, en efecto es consistente con la teoría.