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 aqui.

Ejercicio 1

Dada la ecuacion del calor, tenemos el PVIF:

$$u_t = \frac{1}{\pi} u_{xx} \tag{1}$$

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. Ademas, sabemos que la solucion exacta es:

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

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 $h = 1/20$. Podemos escribir el esquema de forma matricial de la siguiente forma:

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

donde:

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

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

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

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 $\rho(A) = max \quad |\lambda_i| \leq 1$ 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 Neuamnn.

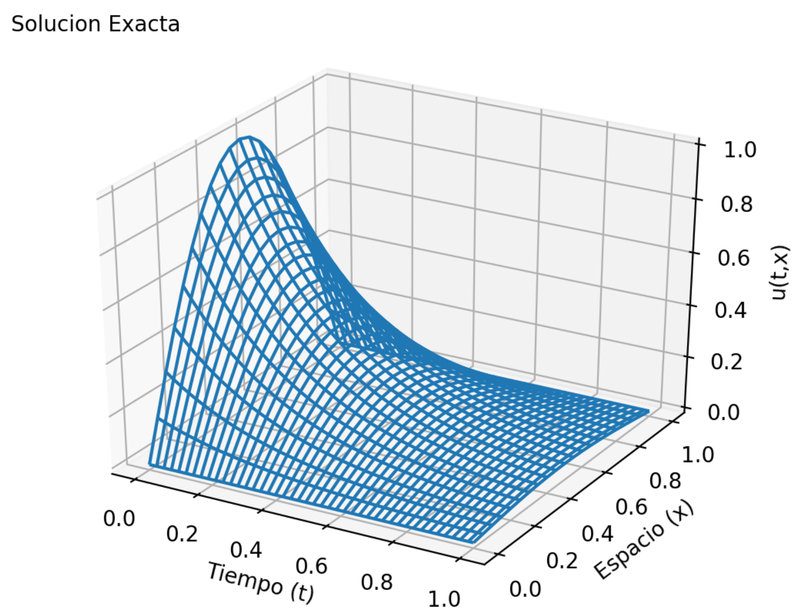
Ahora apliquemos el metodo explicito con 20 pasos en el espacio ($h = 1/20$) y 256 pasos en el tiempo ($k = 1/256$), y veamos el maximo de los errores al tiempo $T = 1$:

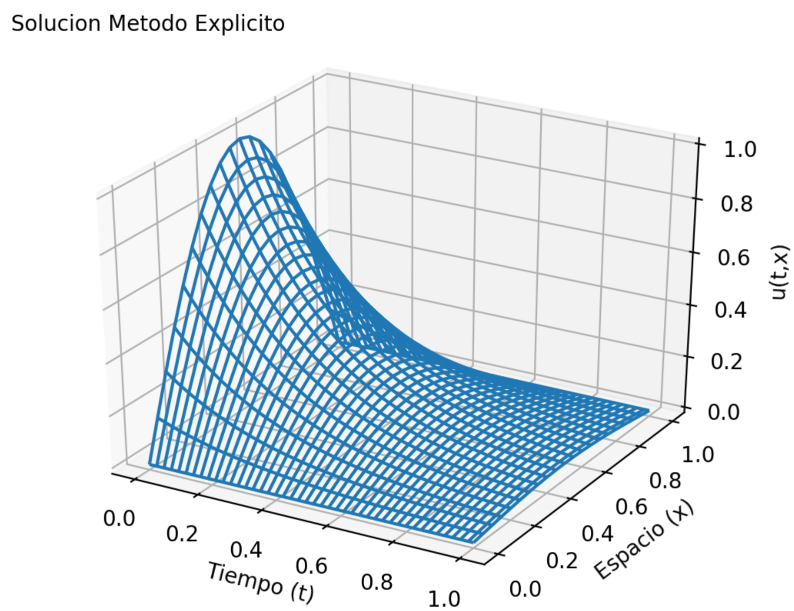
In [2]:
```
M = 20
N = 256

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 Metodo Explicito
1.0
0.8
0.6
0.4
0.2
0.0
u(t,x)
0.0
0.2
0.4
0.6
0.8
1.0
Tiempo (t)
0.0
0.2
0.4
0.6
0.8
1.0
Espacio (x)