Análisis Numérico

Segundo Cuatrimestre 2025

Entrega N°2

Esta entrega está motivada por los ejercicios 6 y 7 de la Guía de Laboratorio 2.
El objetivo es resolver el siguiente problema

$$\begin{cases} U_t &= U_{xx} + U_{yy} & \Omega = [0,2]^2,\ t \in (0, T_f) \\ U(x,y,t) &= 0 & (x,y) \in \partial\Omega,\ t \in (0, T_f). \\ U(x,y,0) &= e^{-10((x-0.5)^2+(y-0.5)^2)} & (x,y) \in \Omega \end{cases}$$

Consideramos una grilla del interior de Ω:

$$\{x_i = ih_x :\ 1 \le i \le I-1\} \times \{y_j = jh_y :\ 1 \le j \le J-1\}.$$

Para simplificar, puede tomarse $h_x = h_y = h$ e $I = J$. A esto agregamos la grilla temporal $\{t_n = n\Delta t :\ 0 \le T\}$, con $T = T_f/\Delta t$. Sobre esta grilla se define u^n_{ij}, una aproximación de $U(x_i, y_j, t_n)$. Notamos u^n al arreglo (u^n_{ij}).

Método Explícito:

$$u^{n+1} = u^n + r(\delta_x^2(u^n) + \delta_y^2(u^n))$$

Método de Crank-Nicholson:
Este método consiste en implementar

$$\left(1 - \frac{1}{2}r_x\delta_x^2 - \frac{1}{2}r_y\delta_y^2\right)u^{n+1} = \left(1 + \frac{1}{2}r_x\delta_x^2 + \frac{1}{2}r_y\delta_y^2\right)u^n$$

Método ADI:
El método de direcciones alternadas (ADI) consiste en resolver dos problemas en cada iteración temporal, cada uno de ellos implícito en una única coordenada, y explícito en la otra. Definiendo la solución a un paso intermedio $u^{n+\frac{1}{2}}$, calculamos

$$(1 - \frac{1}{2}r_x\delta_x^2)u^{n+\frac{1}{2}} = (1 + \frac{1}{2}r_y\delta_y^2)u^n$$
$$(1 - \frac{1}{2}r_y\delta_y^2)u^{n+1} = (1 + \frac{1}{2}r_x\delta_x^2)u^{n+\frac{1}{2}}$$

Ejercicio 1 Implementar un método explícito, el método de C-N y el método ADI para la ecuación dada. Realizar el gráfico de la solución a tiempo final. Y estudiar numéricamente la estabilidad de los métodos.

Ejercicio 2 Comparar los tiempos de ejecución con el macro `@time` entre el método de C-N y el método ADI.

IMPORTANTE: La entrega debe ser a través de la creación de una librería propia `DiferenciasFinitas.jl`

Sugerencias:

1. Puede ser útil implementar la función `meshgrid`

```
function meshgrid(x,y)
    X = [xi for yi in y,xi in x]
    Y = [yi for yi in y,xi in x]
    return X,Y
end
```

   Esta función devuelve dos matrices `X,Y` tal que `(X[i,j],Y[i,j])` representa un punto de la grilla generada por x e y.

2. Para el caso del Método de C-N tener en cuenta que deberán invertir una matriz de $(J-1)\times(J-1)$. Es recomendable pasar la matriz u^n a un vector con la función `vec()` y al finalizar usar la función `reshape()`. Para el armado de la matriz pueden usar lo visto en la teórica o usar la función `kron()` para armarse la matriz del lado derecho. Esta función utiliza el producto de Kronecker de tal forma que si A es la matriz tridiagonal habitual, el producto $A \otimes I$ realiza las derivadas en la dirección x mientras que $I \otimes A$ las correspondientes a y, donde I es la matriz identidad de tamaño $(J-1)\times(J-1)$. `https://es.wikipedia.org/wiki/Producto_de_Kronecker`