# Tarea 3
# Análisis de Fourier

**IMT2113-1**

Entregar hasta el viernes 6 de noviembre 2020, 23:59.

**Nombre: Lucas Buvinic Meira**

## Problemas

### Problema 1.

*Solución.* Elegimos $(N_x - 1)(N_y - 1)$ nodos (en $x$ desde 0 a $N_x$ y en $y$ desde 0 hasta $N_y$). Sabemos que $A$ es simétrica. También podemos ver que, como $0 < ih_x, jh_y < 1$ para $i \in \{1, \dots, 1/h_x - 1\}$ y $j \in \{1, \dots, 1/h_y - 1\}$, y como $i \pm \frac{1}{2} > 0$ y $j \pm \frac{1}{2} > 0$, $\alpha_{i\pm\frac{1}{2},j}, \alpha_{i,j\pm\frac{1}{2}} > 0$. Entonces es claro que el centro del stencil que define la diagonal $A_{k,k}$, $k \in \{1, \dots, ((N_x - 1)(N_y - 1))^2\}$ y $k = (i, j)$.

Ahora vemos que:

$$\sum_{j=1\,,j\neq k} |A_{k,j}| = E_k + W_k + S_k + N_k \leq A_{k,k}$$

Entonces por el Teorema de Gershorin tenemos que $A$ debe todos los valores propios $\lambda$ de $A$ son positivos. Esto implica que $A$ es positiva definida por lo siguiente. Asumamos que $A$ es efectivamente definida positiva. Tenemos dos casos: Digamos que algún $\lambda = 0$. Entonces tenemos que existe $x$ tal que $Ax = 0 \implies x^T Ax = 0$, una contradicción. Luego digamos que $\lambda < 0$. Entonces tenemos que existe $x$ tal que $Ax = \lambda x \implies x^T Ax = x^T x\lambda = \lambda|x|^2 < 0$, una contradicción. Entonces $A$ debe ser definida positiva, lo que es una condición suficiente para aplicar el algoritmo CG. ■

### Problema 2.

*Solución.* a) Elegimos una partición en bloques horizontales de los nodos.

b) Los arrays se guardan por secciones en cada procesador. Particionamos cada vector respecto a la partición por bloques de los nodos. Los stenciles se generan en cada procesador para sus nodos respectivos. En particular, si tenemos $(N_x-1)(N_y-1)$ nodos, particionamos primero el dominio horizontal de manera semi-equitativa, dejando los nodos del resto de la división entera al último procesador. Cada stencil tiene tamaño $3 \times 3$ y tenemos un total de $(N_x - 1)(N_y - 1)$ de nodos. Cada procesador guarda, entonces,

$\frac{9(N_x-1)(N_y-1)}{P}$ datos, donde $P$ es el número de procesadores. Los vectores son de tamaño $1 \times (N_x - 1)(N_y - 1)$ y necesitamos tener siempre disponible el espacio equivalente a 3 de estos (sin contar los 2 extra que ocupamos para la comunicación entre procesadores en el matvec), por lo que cada procesador guarda $\frac{3(N_x-1)(N_y-1)}{P}$ datos ∎

**Problema 3.**

*Solución.* c) Elegimos los vectores el vector $b = (1, 0, 0, \cdots)$ y $x_0 = 0$. e) Para las operaciones de álgebra lineal básicas ocupé las operaciones colectivas que ofrece MPI. Por el otro lado, para la comunicación entre procesos en el matvec decidí hacer una comunicación más simple pero que exige un poco de overhead. Agrupé los datos fronterizos de cada sección de la malla correspondientes a cada procesador y los envié en orden ascendente y luego descendente (porque cada procesador requiere los valore fronterizos de la izqueirda y la derecha para hacer el matvec), dando especificaciones para los distintos tipos de nodos (nodos de ezquina, abajo, arriba, a la izquierda a la derehca, en el medio). El overhead aquí aparece por tener que almacenar 2 arrays de tamaño $1 \times (N_y - 1)$ extra en cada procesador. Para esta comunicación ocupé las funciones MPIReciv y MPISend que ofrece MPI. ∎