

# Tarea 2

## IMT2112 - Algoritmos Paralelos en Computación Científica

Elwin van 't Wout

9 de septiembre de 2019

### Introducción

Las factorizaciones de matrices sirven, por ejemplo, para resolver sistemas lineales de ecuaciones. La factorización  $A = LU$  es una descomposición en una matriz triangular inferior  $L$  y una matriz triangular superior  $U$ . La solución de una ecuación  $A\mathbf{x} = \mathbf{b}$  es eficiente cuando la factorización  $LU$  es conocida.

### Actividades

Para la tarea, responden las preguntas y hagan las actividades siguientes.

1. Explica brevemente como funciona la descomposición LU y como usarla para resolver un sistema lineal. Usa el algoritmo visto en clase, que resulta en una matriz  $U$  con elementos de valor uno en el diagonal principal.
2. Implementa el algoritmo en C++, con las condiciones siguientes.
  - a) Sintetiza la matriz  $A$  con un generador de números aleatorios.  
*Sugerencia:* se puede usar la biblioteca `<cstdlib>` que cuenta con la función `rand()` que genera un número aleatorio entre cero y un número máximo dado por `RAND_MAX`. Tenga cuidado con la asignación de los variables como número entero o número de punto flotante.
  - b) El algoritmo debe funcionar para cualquier dimensión de la matriz.
  - c) Se debe usar los arreglos estándares de C++.
  - d) Se debe implementar todas las sumas con bucles. Entonces, al final hay tres niveles de bucles.
3. Verifica el resultado de la descomposición, es decir, calcula  $LU$  y el error  $\|A - LU\|$ .
4. Usa la biblioteca OpenMP para paralelizar el código en dos maneras.
  - a) El primer nivel de bucles no se puede paralelizar, explica por qué.
  - b) Paraleliza el segundo y el tercer nivel de bucles de forma separado. Es decir, un (parte de) código que paraleliza el segundo nivel de bucles y otro código para paralelizar el tercero.
  - c) Explica qué pasará si se paraleliza ambos niveles de bucles al mismo tiempo.

5. Cambia la forma de programación de carga, es decir, usa la función `scheduling` de OpenMP para elegir una programación distinta. Explica si se espera un mejor rendimiento con una programación distinta, por ejemplo cuales son las ventajas y desventajas de programación estática y dinámica. Verifica el rendimiento de tu código para distintas programaciones y discute el resultado.

## Reglamento

La tarea debe cumplir con las reglas siguientes.

- Mientras se puede trabajar junto a compañeros en la preparación de las tareas, el código y el informe deben ser elaborados de forma individual.
- No es permitido copiar código de fuentes externos, tampoco sacar código de internet. El código entregado debe ser programado por tu mismo.
- Entrega un archivo comprimido con el código de C++ y un informe corto, elaborado en  $\text{\LaTeX}$ , con las respuestas a las preguntas. Suben el archivo al cuestionario en Siding y usa como nombre del archivo `Tarea2_apellido.zip` en lo cual se tiene que reemplazar `apellido` con tu propio apellido.
- El plazo de entrega es detallado en el cuestionario correspondiente a este tarea en Siding. Es posible subir la tarea múltiples veces hasta el plazo del cuestionario.