

# Laboratorio de Métodos Numéricos

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

## Trabajo Práctico Número 1

Con 15  $\theta$ 's discretizo alto horno

Integrante	LU	Correo electrónico
Ciruelos Rodríguez, Gonzalo	063/14	gonzalo.ciruelos@gmail.com
Costa, Manuel José Joaquín	035/14	manuc94@hotmail.com
Gatti, Mathias Nicolás	477/14	mathigatti@gmail.com

Descripcion del Tp

Eliminación gaussiana    factorización LU    discretización    key4

## 1. Introducción teórica

## 2. Desarrollo

### 2.1. Convenciones

De aquí en adelante, si no se aclara otra cosa, se asumen las siguientes convenciones:

- $r_i$  es el radio que va del centro del horno al borde interno de la pared, mientras que  $r_e$  es el radio considerando el borde externo;
- $n$  es la cantidad de ángulos discretos ( $0 = \theta_0 < \dots < \theta_{n-1} = 2\pi - \Delta\theta$ ) en los que se particiona la pared;
- $m + 1$  es el total de radios discretos ( $r_i = r_0 < \dots < r_m = r_e$ );
- $t_{k,j} = T(r_k, \theta_j)$ , donde  $T$  es la función de temperatura de la pared (que desconocemos);
- $b \in \mathbb{R}^{n \times (m+1)} : b = (b_0, b_1, \dots, b_{(n \times (m+1)) - 1})$  será el vector de términos independientes del sistema que plantearemos luego (veremos que la dimensión escogida es correcta).

### 2.2. Métodos numéricos usados

A partir de la ecuación del calor de Laplace y las discretizaciones de las derivadas parciales dadas en el enunciado del presente trabajo práctico, se puede obtener un sistema de ecuaciones donde las soluciones son las temperaturas en los puntos de la discretización. Es decir que el problema de hallar la isoterma se reduce a dos sub-problemas: el primero es resolver efectivamente el sistema planteado, mientras que el segundo consiste en usar las temperaturas halladas para estimar la posición de la isoterma.

Para resolver el sistema haremos uso de los métodos de eliminación gaussiana y la factorización LU, para luego poder contrastar su eficiencia ante distintas situaciones.

Para hallar la posición de la isoterma (de temperatura  $t_{iso}$ ), dado un ángulo  $\theta_j$  de la discretización, lo que haremos es buscar dos radios,  $r_k$  y  $r_{k+1}$ , tales que  $t_{k+1,j} \leq t_{iso} \leq t_{k,j}$ . A partir de estos dos valores, fácilmente podemos realizar un ajuste lineal considerando la recta de pendiente  $m = \frac{t_{k,j} - t_{k+1,j}}{\Delta r}$  que pasa por  $(r_k, t_{k,j})$  y  $(r_{k+1}, t_{k+1,j})$ . Para una explicación más detallada ver el apéndice ?????.

### 2.3. Armado del sistema de ecuaciones y su matriz asociada

Tenemos un sistema con  $n \times (m + 1)$  incógnitas.

Primero contamos con  $n$  variables, los  $t_{0,j}$  con  $j = 0, 1, \dots, n - 1$  (es decir, las temperaturas interiores), cuyos valores conocemos pues nos son dados como inputs. Lo mismo sucede con  $t_{m,j}$  para  $j = 0, 1, \dots, n - 1$  (temperaturas externas). Luego, para las temperaturas interiores sabemos gracias a la función de Laplace y a la discretización de las derivadas que vale que

$$\frac{t_{k-1,j} - 2t_{k,j} + t_{k+1,j}}{(\Delta r)^2} + \frac{1}{r_k} \times \frac{t_{k,j} - t_{k-1,j}}{\Delta r} + \frac{1}{r_k^2} \times \frac{t_{k,j-1} - 2t_{k,j} + t_{k,j+1}}{(\Delta \theta)^2} = 0$$

Reescribiendo convenientemente la ecuación de arriba nos queda que

$$\frac{r_k - \Delta r}{r_k(\Delta r)^2} t_{k-1,j} + \frac{r_k \Delta r (\Delta \theta)^2 - 2(\Delta r)^2}{r_k^2 (\Delta r)^2 (\Delta \theta)^2} t_{k,j} + \frac{1}{(\Delta r)^2} t_{k+1,j} + \frac{1}{r_k^2 (\Delta \theta)^2} t_{k,j-1} + \frac{1}{r_k^2 (\Delta \theta)^2} t_{k,j+1} = 0 \quad (1)$$

Vale destacar que si  $j = 0$  entonces en lugar de  $t_{k,j-1}$  se usa  $t_{k,n-1}$ , mientras que si  $j = n - 1$  en lugar de  $t_{k,j+1}$  va  $t_{k,0}$ .

De esto se desprende inmediatamente que el valor de cada  $t_{k,j}$  ( $1 \leq k < m$ ) depende exclusivamente de sus cuatro vecinos. Cabe mencionar también que si bien todos los coeficientes dependen de la distancia al centro del horno ( $r_k$ ), ninguno lo hace respecto del ángulo concreto en que se encuentra el punto. Esto resulta muy razonable si tenemos en cuenta que el valor del ángulo depende exclusivamente del sistema de referencia escogido (dónde ubicamos el ángulo 0), mientras que dado un punto de la pared es lógico que su temperatura no dependa del sistema de referencia escogido para su medición.

Para facilitar la lectura, de ahora en adelante llamaremos a los coeficientes de la ecuación (1) (respetando el orden en que aparecen):  $a_k$  (el coeficiente que acompaña a  $t_{k-1,j}$ ),  $b_k$ ,  $c_k$ ,  $d_k$  (el coeficiente que acompaña a  $t_{k,j-1}$  y  $t_{k,j+1}$ ). Notar que por la observación anterior estos coeficientes sólo dependen del radio.

$$\left\{ \begin{array}{l} t_{0,0} = b_0 \\ t_{0,1} = b_1 \\ \vdots \\ t_{0,n-1} = b_{n-1} \\ a_1 t_{0,0} + b_1 t_{1,0} + c_1 t_{2,0} + d_1 t_{1,n-1} + d_1 t_{1,1} = 0 \\ \vdots \\ a_1 t_{0,n-1} + b_1 t_{1,n-1} + c_1 t_{2,n-1} + d_1 t_{1,n-2} + d_1 t_{1,0} = 0 \\ a_2 t_{1,0} + b_2 t_{2,0} + c_2 t_{3,0} + d_2 t_{2,n-1} + d_2 t_{2,1} = 0 \\ \vdots \\ a_{m-1} t_{m-2,n-1} + b_{m-1} t_{m-1,n-1} + c_{m-1} t_{m,n-1} + d_{m-1} t_{m-1,n-2} + d_{m-1} t_{m-1,0} = 0 \\ t_{m,0} = b_{n \times m} \\ \vdots \\ t_{m,n-1} = b_{n \times (m+1) - 1} \end{array} \right. \quad (2)$$

## 2.4. Estructuración del código

Para el modelado del problema diseñamos dos módulos: Matriz y Sistema.

El primero provee una representación para matrices con una interfaz conveniente. Utilizamos como representación interna un vector de vectores, junto con la cantidad de filas y columnas de la matriz. Las operaciones de Matriz permiten, dado un vector de términos independientes, obtener la solución de un sistema triangular superior sin ceros en la diagonal (o triangular inferior sin ceros en la diagonal) mediante *backward substitution* (respectivamente *forward substitution*), realizar eliminación gaussiana sin pivoteo (en caso de que sea posible), y, de existir, obtener la factorización LU de una matriz. Adicionalmente, hicimos otras versiones de eliminación gaussiana y factorización LU que requieren que la matriz en la que se aplican sea banda, y aprovechan este hecho para reducir la cantidad de operaciones básicas a realizar.

### 3. Resultados y discusión

#### 3.1. Evaluación de los métodos

##### 3.1.1. Único $b$

Nuestro primer análisis de los métodos utilizados para la resolución del problema va a consistir en una teórica.

La resolución mediante el método de eliminación Gaussiana tiene dos partes, la primera es la eliminación gaussiana propiamente dicha, que tiene un costo de  $O(k^3)$  flops, donde  $k$  es la cantidad de filas (y columnas) de la matriz y la segunda es el algoritmo llamado *backwards substitution*, que tiene un costo de  $O(k^2)$  flops.

La resolución mediante el método de factorización LU tiene tres partes, la primera es obtener la factorización LU de la matriz en cuestión, que tiene un costo de  $O(k^3)$ , y luego aplicar *forward substitution* y *backward substitution*, cada uno con un costo de  $O(n^2)$  flops.

Para medir los tiempos de ambas implementaciones decidimos hacer dos experimentos separados, sin embargo similares. En uno fijamos una cantidad de radios y movemos la cantidad de ángulos, y en el otro al revés. Lo hicimos así para que las dimensiones de la matriz (ancho o alto, dado que son iguales porque es cuadrada) crezca linealmente, ya que si hacemos variar ángulos y radios al mismo tiempo, las dimensiones dejan de crecer linealmente. Queremos que las dimensiones crezcan linealmente para que los resultados se entiendan mejor, dado que una escala lineal es, generalmente, más naturales a la vista.

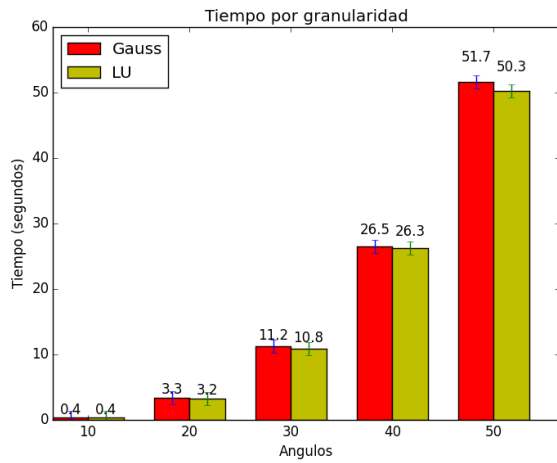


Figura 1: Tiempo tomado por la nuestra implementación de eliminación gaussiana y de factorización LU para resolver el problema. La granularidad de radios está fija en 40 y la de ángulos se indica en el eje  $x$ . La barra principal indica el promedio, y el segmento indica la desviación standard.

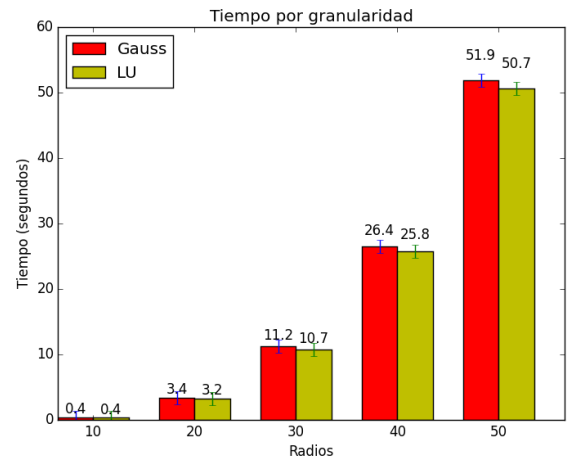


Figura 2: Tiempo tomado por la nuestra implementación de eliminación gaussiana y de factorización LU para resolver el problema. La granularidad de ángulos está fija en 40 y la de radios se indica en el eje  $x$ . La barra principal indica el promedio, y el segmento indica la desviación standard.

Como se ve en las figuras 1 y 2, las performances entre las implementaciones son realmente similares. También puede observarse algo que nos llamó la atención, que es que el método por eliminación gaussiana tuvo consistentemente peor performance que LU, contradiciendo parcialmente el análisis a priori. Sin embargo, la diferencia es mínima: siempre menor al 3%.

Por esta razón no creemos que sea algo importante a tener en cuenta, y podríamos atribuirlo a optimizaciones del compilador, dado que los códigos son realmente parecidos (de hecho hicimos nuestra implementación de la factorización LU sobre nuestra implementación de eliminación gaussiana), y las partes que son diferentes le agregan complejidad a la factorización LU.

Por eso lo atribuimos a cuestiones relacionadas con la optimización en tiempo de compilación y no a errores en las mediciones, dado que los tests fueron corridos varias veces por el hecho de que estos resultados eran llamativos.

Por otro lado, vemos que dividir entre ángulos y radios no hace diferencia en la performance, dado que como esperábamos, el runtime solamente depende del tamaño de la matriz, es decir  $n(m+1)$ , que en ambos gráficos es igual columna a columna.

Ahora es el turno de las implementaciones que aprovechan que la matriz es banda. La diferencia es enorme. Para comparar rápidamente, podemos usar una tabla:

Implementación	Gauss	Gauss Banda	LU	LU Banda
Tiempo (segundos)	26.24	0.98	25.53	1.04

Figura 3: Tiempo promedio tomado por las implementaciones para resolver un sistema con  $n = m + 1 = 40$ .

Con la Figura 3 se observa como la diferencia entre las implementaciones *vanilla* y las optimizadas es abismal, superando el 2600 %.

Esta tabla no pretende analizar las implementaciones optimizadas, simplemente probar la diferencia de rendimiento que se obtiene.

A continuación analizaremos las implementaciones optimizadas.

[ACA ANALIZAMOS LA COMPLEJIDAD O LO REPETIMOS CORTITO]

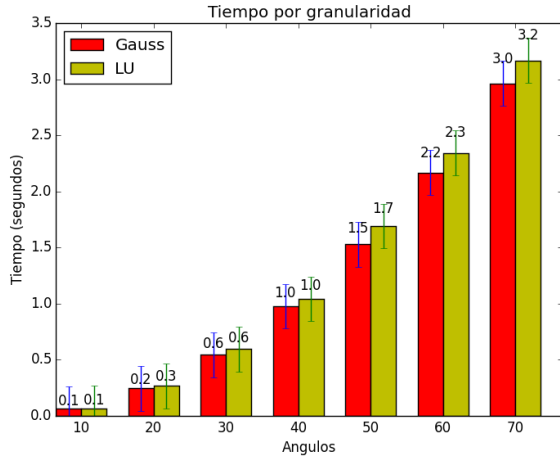


Figura 4: Tiempo tomado por la nuestra implementación optimizada de eliminación gaussiana y de factorización LU para resolver el problema. La granularidad de radios está fija en 40 y la de ángulos se indica en el eje  $x$ . La barra principal indica el promedio, y el segmento indica la desviación standard.

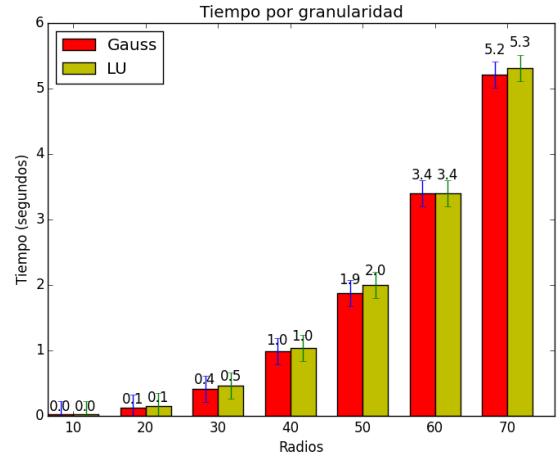


Figura 5: Tiempo tomado por la nuestra implementación optimizada de eliminación gaussiana y de factorización LU para resolver el problema. La granularidad de ángulos está fija en 40 y la de radios se indica en el eje  $x$ . La barra principal indica el promedio, y el segmento indica la desviación standard.

Los resultados que se ven en las Figuras 4 y 5 son muy interesantes. Primero notemos que como los procesos de eliminación gaussiana y factorización LU ambos consumen menos tiempo, se nota más la ventaja que le saca gauss a LU con un solo  $b$ , dado que los procesos de sustitución empiezan a pesar asintóticamente.

Como la complejidad de los algoritmos optimizados es  $O(\text{radios}^3 \text{angulos})$ , en los experimentos se refleja que es mucho más caro aumentar la granularidad con respecto a los radios que con respecto a los ángulos.

### 3.1.2. Múltiples $b$ 's

Cuando buscamos simular un escenario similar al experimento anterior, pero donde las condiciones de borde (temperaturas interiores y exteriores) cambian en distintos instantes de tiempo, la situación es muy distinta. Aquí esperaríamos ver el verdadero poder de la factorización LU.

Por esta razón, esperamos que la implementación de la factorización LU supere ampliamente a la de la eliminación Gaussiana, dado que en la Gaussiana se paga un costo cúbico cada vez que se quiere resolver el sistema, mientras que en con la factorización LU el costo cúbico se paga solo una vez.

Por esta razón se puede decir que, para muchas instancias, el costo de resolver  $Ax = b$  es cúbico para la eliminación gaussiana y cuadrático amortizado para la factorización LU.

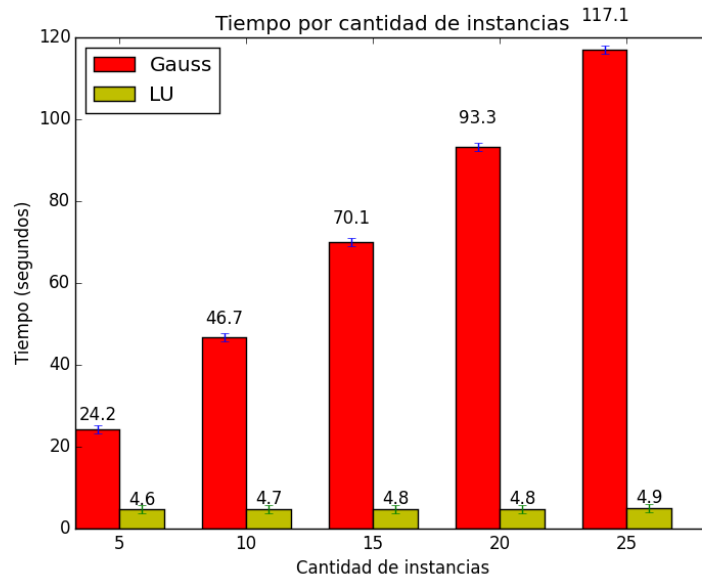


Figura 6: Tiempo tomado por la nuestra implementación de eliminación gaussiana y de factorización LU para resolver el problema para varias instancias de  $n = m + 1 = 30$ . La barra principal indica el promedio, y el segmento indica la desviación standard.

En los resultados se refleja lo que esperabamos. Como se ve en la figura 6, luego de aplicar la factorización LU, el costo que hay que pagar para resolver cada sistema es muy poco, lo cual permite una excelente performance.

Por otra parte, se observa que el tiempo que le toma a la eliminación gaussiana resolver  $n$  instancias, es también es lineal en la cantidad de instancias, pero con una pendiente mucho mayor, lo cual confirma más aún nuestras expectativas derivadas de la teoría.

En cuanto a las implementaciones optimizadas, esperamos que su comportamiento sea similar a los resultados anteriores.

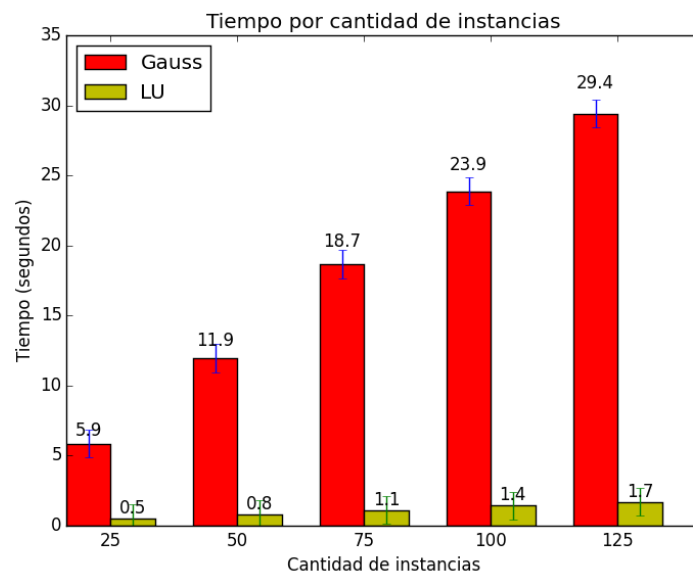


Figura 7: Tiempo tomado por la nuestra implementación optimizada de eliminación gaussiana y de factorización LU para resolver el problema para varias instancias de  $n = m + 1 = 30$ . La barra principal indica el promedio, y el segmento indica la desviación standard.

Como se observa en la Figura 7, confirmamos nuestras expectativas. Se observa, al igual que antes, que la implementación optimizada de Gauss crece linealmente con respecto a la cantidad de instancias del problema entradas, al igual que la implementación de la factorización LU, sólo que esta ultima lo hace con una pendiente mucho menor.

## 4. Conclusiones

## 5. Apéndices