



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Métodos Numéricos

Trabajo Práctico 1
Primer Cuatrimestre de 2022

Integrante	LU	Correo electrónico
Valeria Arrosio	223/20	vale.arrosio@gmail.com
Camila Rosario Camaño	310/17	camicam26@gmail.com
Lucas Fabrizio Di Salvo	446/18	lucasdisalvo@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (+ +54 +11) 4576-3300

<https://exactas.uba.ar>

Resumen

En el siguiente informe nos proponemos encontrar una forma de encontrar una isoterma de 500°C dentro de las paredes de un alto horno con el objetivo de determinar el nivel de peligrosidad de la estructura de las paredes.

Utilizaremos una ecuación de calor (ecuación de Laplace), la cual todos los puntos del horno deben cumplir, pero para facilitar los cálculos, construiremos una discretización de la pared del horno y de la propia ecuación. Finalizado este proceso, organizaremos las ecuaciones resultantes en un sistema de ecuaciones para reescribirlo como una matriz. Luego, utilizando eliminación Gaussiana y factorización LU, resolveremos el sistema para obtener el valor de todas las temperaturas internas y poder estimar la posición de la isoterma buscada.

Generaremos 3 instancias de datasets con distintos objetivos: conseguir una buena discretización que no tenga gran costo computacional, decidir qué método es el más eficiente, y observar qué sucede en un sistema en peligro.

Habiendo encontrado una discretización satisfactoria y el método más eficiente, nos disponemos a ver cómo se comporta la isoterma cuando en ciertos puntos de la pared externa se registran temperaturas más elevadas que en otros.

Palabras clave: Análisis Numérico, Eliminación Gaussiana, Factorización LU

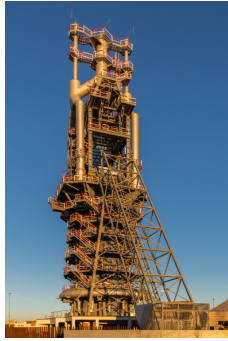
Índice

1. Introducción	2
1.1. Modelo	2
1.2. Demostración	5
1.3. Métodos numéricos utilizados	7
1.3.1. Eliminación Gaussiana	7
1.3.2. Factorización LU	8
2. Desarrollo	9
2.1. Implementaciones en código	9
2.1.1. Resolución de sistemas	9
2.1.2. Resolución Gauss	9
2.1.3. Resolución LU	9
2.1.4. Isotherma y peligrosidad	9
2.2. Experimentos	9
2.2.1. Experimento 1: Prueba de modelo	10
2.2.2. Experimento 2: Prueba de cómputo	10
2.2.3. Experimento 3A: Prueba de riesgo	11
2.2.4. Experimento 3B: Prueba de sensado	12
3. Resultados	13
3.1. Análisis del Experimento 1	13
3.2. Análisis del Experimento 2	14
3.3. Análisis del Experimento 3	15
3.3.1. Experimento 3.A	15
3.3.2. Experimento 3.B	17
4. Conclusiones	18
5. Referencias	18

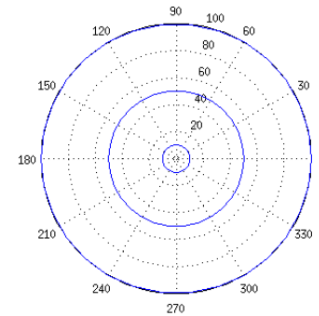
1. Introducción

Al trabajar con maquinaria pesada, es de suma importancia realizar constantes chequeos de control para evitar accidentes y catástrofes. En el siguiente informe implementaremos dos formas de observar y catalogar el estado de un alto horno.

Un **alto horno** consiste en una estructura cilíndrica donde en su interior se lleva a cabo la fusión de distintos materiales. A su vez, la **isoterma** consiste en una indicación de puntos con un mismo valor. Dispondremos de sensores en la pared externa del horno con el objetivo de estimar la posición de la isoterma de 500°C a lo largo del sistema para chequear si la pared externa del horno podría colapsar o no.



(a) Alto horno en la vida real



(b) Gráfico de una isoterma

Figura 1: Imágenes de un alto horno y una isoterma.

1.1. Modelo

Modelaremos el problema de la siguiente manera:

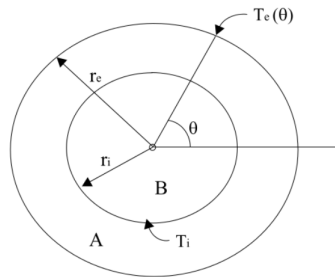


Figura 2: Diagrama del horno

Donde:

- $T(r, \theta)$ es la temperatura en el punto dado por las coordenadas polares (r, θ) , con r el radio y θ el ángulo polar de dicho punto,
- $r_e \in \mathbb{R}$ es el radio exterior de la pared,
- $r_i \in \mathbb{R}$ es el radio interior de la pared,
- $T_e(\theta) = T(r_e, \theta)$ para todo punto (r_e, θ) ,
- $T_i = T(r, \theta)$ para todo punto (r, θ) con $r \leq r_i$.

Luego, tendremos en cuenta que las temperaturas satisfacen la siguiente ecuación de calor de Laplace:

$$\frac{\partial^2 T(r, \theta)}{\partial r^2} + \frac{1}{r} \frac{\partial T(r, \theta)}{\partial r} + \frac{1}{r^2} \frac{\partial^2 T(r, \theta)}{\partial \theta^2} = 0 \quad (1)$$

Utilizaremos las fórmulas propuestas en el enunciado para armar un conjunto de ecuaciones lineales sobre las incógnitas $t_{j,k}$, donde j = cantidad de radios, k = cantidad de ángulos y $t_{j,k} = T(r, \theta)$. Entonces, utilizaremos las siguientes ecuaciones:

$$\frac{\partial^2 T(r, \theta)}{\partial r^2}(r_j, \theta_k) \approx \frac{t_{j-1,k} - 2t_{j,k} + t_{j+1,k}}{\Delta r^2} \quad (2)$$

$$\frac{\partial T(r, \theta)}{\partial r}(r_j, \theta_k) \approx \frac{t_{j,k} - t_{j-1,k}}{\Delta r} \quad (3)$$

$$\frac{\partial T(r, \theta)}{\partial \theta^2}(r_j, \theta_k) \approx \frac{t_{j,k-1} - 2t_{j,k} + t_{j,k+1}}{\Delta \theta^2} \quad (4)$$

Realizando los reemplazos correspondientes en la ecuación 1 con los términos de 2, 3 y 4, obtenemos la siguiente ecuación:

$$\left(\frac{1}{\Delta r^2} - \frac{1}{r_j \Delta r}\right)t_{(j-1,k)} + \left(\frac{-2}{\Delta r^2} + \frac{1}{r_j \Delta r} - \frac{2}{r_j^2 \Delta \theta^2}\right)t_{(j,k)} + \left(\frac{1}{\Delta r^2}\right)t_{(j+1,k)} + \left(\frac{1}{r_j^2 \Delta \theta^2}\right)t_{(j,k-1)} + \left(\frac{1}{r_j^2 \Delta \theta^2}\right)t_{(j,k+1)} \quad (5)$$

Observando con cuidado los términos que se repiten, podemos reescribir los coeficientes como:

$$\mathbf{a} = \frac{1}{\Delta r^2}, \mathbf{b} = \frac{1}{r_j \Delta r}, \mathbf{c} = \frac{1}{r_j^2 \Delta \theta^2}$$

Teniendo esto en cuenta, podemos reescribir la ecuación 1 como:

$$(a - b)t_{(j-1,k)} + (-2ab + b - 2c)t_{(j,k)} + at_{(j+1,k)} + ct_{(j,k-1)} + ct_{(j,k+1)} = 0 \quad (6)$$

Con todo esto, ya tenemos lo necesario para armar el sistema de ecuaciones a resolver: conocemos los valores de las temperaturas de las paredes externas e internas, y a su vez sabemos que las temperaturas interiores cumplen con la ecuación 1.

Recopilando toda la información hasta ahora, si tomamos n cantidad de ángulos y $m + 1$ cantidad de radios, conseguimos la siguiente discretización del horno como una matriz de $\mathbb{R}^{(m+1) \times n}$:

r_0, θ_0	r_0, θ_1	...	r_0, θ_n
r_1, θ_0	r_1, θ_1	...	r_1, θ_n
...
r_{m+1}, θ_0	r_{m+1}, θ_1	...	r_{m+1}, θ_n

En donde la primera columna representa las temperaturas de las paredes internas, y la última las temperaturas de las paredes externas. Teniendo en cuenta que estos valores son conocidos, las incógnitas a resolver son "las del medio".

r_0, θ_0	r_0, θ_1	...	r_0, θ_{n-1}	r_0, θ_n
r_1, θ_0	r_1, θ_1	...	r_1, θ_{n-1}	r_1, θ_n
...
r_{m+1}, θ_0	r_{m+1}, θ_1	...	r_{m+1}, θ_{n-1}	r_{m+1}, θ_n

En total, tendríamos $(m + 1) * n$ elementos en la matriz, donde $m + 1$ es la cantidad de columnas y n la cantidad de filas. Si quitamos la primera y última columna, tendríamos en total $(m + 1 - 2) * n = (m - 1) * n$ incógnitas a resolver, de las cuales ya sabemos que cumplen con la ecuación 1 y las podríamos calcular.

Ahora reescribimos tomando $t_{j,k} = T(r_j, \theta_k)$, con $j = 1, \dots, m+1$ y $k = 1, \dots, n-1$ en la ecuación 6, obteniendo un sistema de la siguiente forma:

$$\begin{array}{cccccc}
 (a-b) \cdot t_{01} & (-2a+b-2c) \cdot t_{11} & a \cdot t_{21} & c \cdot t_{13} & c \cdot t_{12} & = 0 \\
 (a-b) \cdot t_{02} & (-2a+b-2c) \cdot t_{12} & a \cdot t_{22} & c \cdot t_{11} & c \cdot t_{13} & = 0 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & = \vdots \\
 (a-b) \cdot t_{j-1,k} & (-2a+b-2c) \cdot t_{j,k} & a \cdot t_{j+1,k} & c \cdot t_{j,k-1} & c \cdot t_{j,k+1} & = 0
 \end{array}$$

con el que tendremos la salvedad de que cuando nos encontremos con que alguna incógnita tiene cómo índice la primer fila del horno (la cual queremos omitir del sistema por tener los mismos valores que la última fila del horno), o los índices exceden los ángulos disponibles, realizamos el siguiente mapeo:

- t_{jk} tal que $k = 0 \mapsto t_{jn}$,
- t_{jk} tal que $k > n \mapsto t_{j1}$,

Ahora tomaremos un ejemplo en donde trabajaremos con $n = 3$ y $m = 3$, lo cual permitirá realizar la siguiente reescritura del sistema, para obtener el sistema de ecuaciones que queremos resolver:

$$\begin{array}{ccccccccc}
 (-2a+b-2c) \cdot t_{11} & c \cdot t_{12} & c \cdot t_{13} & a \cdot t_{21} & 0 & 0 & = & -(a-b) \cdot t_{01} \\
 c \cdot t_{11} & (-2a+b-2c) \cdot t_{12} & c \cdot t_{13} & 0 & a \cdot t_{22} & 0 & = & -(a-b) \cdot t_{02} \\
 c \cdot t_{11} & c \cdot t_{12} & (-2a+b-2c) \cdot t_{13} & 0 & 0 & a \cdot t_{23} & = & -(a-b) \cdot t_{03} \\
 (a-b) \cdot t_{11} & 0 & 0 & (-2a+b-2c) \cdot t_{21} & c \cdot t_{22} & c \cdot t_{23} & = & -a \cdot t_{32} \\
 0 & (a-b) \cdot t_{12} & 0 & c \cdot t_{21} & (-2a+b-2c) \cdot t_{22} & c \cdot t_{23} & = & -a \cdot t_{32} \\
 0 & 0 & (a-b) \cdot t_{13} & c \cdot t_{21} & c \cdot t_{22} & (-2a+b-2c) \cdot t_{23} & = & -a \cdot t_{33}
 \end{array}$$

Resultando en un sistema de tantas filas y columnas como *incógnitas* tengamos en el horno, ubicando las mismas sobre la diagonal, y dejando como termino independiente de cada ecuación los valores que sean *datos*.

En la mayoría de las ecuaciones, el término independiente será cero, pero habrá $2m$ *incógnitas* que se encontrarán al lado de una *pared interior* o de una *pared exterior*, donde dicha información es sabida para la ecuación de calor que respeta ese punto del horno, y se pasa hacia el otro lado de la ecuación multiplicado por el coeficiente que le corresponda, los mismos serán aquellos que tengan un índice para los radios igual a cero (pues se trata de la pared interior), o igual a m (pues se trata de la pared exterior).

Adicionalmente, es importante notar que la forma en la que se planteo este nuevo sistema presenta un orden donde primero se presentan los ángulos para el primer radio, luego para el segundo, hasta el último radio previo a la pared exterior. Dicho esto, se puede observar que la matriz presenta una forma de *banda*.

1.2. Demostración

Luego de este procedimiento, obtenemos un sistema de ecuaciones lineales, con la propiedad de ser una matriz banda. Para su resolución, primero demostraremos que es posible aplicar eliminación Gaussiana sin pivoteo a la matriz. Antes de llegar esa demostración, primero vamos a ver si la matriz resultante es una matriz estrictamente diagonal dominante.

Sea A una matriz banda como la del sistema propuesto, queremos probar que es EDD , es decir, estrictamente diagonal dominante. Si lo fuese, tendría que cumplirse la siguiente condición en cada fila de A :

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}| \quad (7)$$

Sabemos que los coeficientes de la diagonal serán $-2a + b - 2c$, entonces:

$$\begin{aligned} |-2a + b - 2c| &> \sum_{j=1, j \neq i}^n |a_{ij}| \\ |-2(a + c) + b| &> \sum_{j=1, j \neq i}^n |a_{ij}| \end{aligned}$$

Y por desigualdad triangular tenemos:

$$|-2(a + c) + b| \leq |-2(a + c)| + |b| \leq -2|a + c| + |b| \leq 2|a| + 2|c| + |b|$$

Ahora analicemos la sumatoria de los coeficientes restantes de los valores de la matriz: si están cerca de la **pared interna** van a cumplir con

$$|c + c + a| = |2c + a| \leq |2c| + |a|,$$

si están en la pared externa van a cumplir con

$$|(a - b) + c + c| = |a - b + 2c| \leq |a + (-1)b| + |2c| = |a| + |-1||b| + |2c| \leq |a| + |b| + |2c|,$$

y si están en el interior del horno:

$$|(a - b) + a + c + c| = |2(a + c) - b| \leq |2(a + c)| + |-1||b| \leq 2|a| + 2|c| + |b|.$$

Volviendo a la ecuación 7, reemplacemos los valores para cada caso:

- Pared interna:

$$\begin{aligned} |-2(a + c) + b| &> |2c + a| \\ 2|a| + 2|c| + |b| &\geq |-2(a + c) + b| > |2c| + |a| \geq |2c + a| \\ 2|a| + 2|c| + |b| &> 2|c| + |a| \end{aligned}$$

- Pared externa:

$$\begin{aligned} |-2(a + c) + b| &> |a - b + 2c| \\ 2|a| + 2|c| + |b| &\geq |-2(a + c) + b| > |a| + |b| + 2|c| \geq |a - b + 2c| \\ 2|a| + 2|c| + |b| &> |a| + |b| + 2|c| \end{aligned}$$

- Dentro del horno:

$$\begin{aligned} |-2(a + c) + b| &> |a - b + 2c + a| \\ 2|a| + 2|c| + |b| &\geq |-2(a + c) + b| > 2|a| + |b| + 2|c| \geq |a - b + 2c + a| \\ 2|a| + 2|c| + |b| &\geq 2|a| + |b| + 2|c| \end{aligned}$$

Teniendo en cuenta el desarrollo, nos encontramos con que la matriz del sistema es **diagonal dominante (no estrictamente)**. A su vez, sabemos que es una matriz inversible porque el determinante en los $a_{ii} \neq 0$ ya que eso implicaría que las temperaturas a las cuales se está realizando este producto, serían cero, pero dichas temperaturas son la incógnita en cuestión de cada fila del sistema, y las mismas no pueden ser 0.

Ahora, queremos ver si es posible aplicar Gauss sin pivoteo a una matriz *diagonal dominante*. Para ello, vamos a aplicar un paso de Gauss a la matriz y ver cómo quedan los coeficientes. Tomemos una matriz banda $A \in \mathbb{R}^{n \times n}$:

$$\begin{array}{ccccccc} a_{1,1} & a_{1,2} & 0 & \dots & 0 & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & 0 & 0 & 0 \\ 0 & a_{3,2} & a_{3,3} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{n-2,n-2} & a_{n-2,n-1} & 0 \\ 0 & 0 & 0 & \dots & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & 0 & \dots & 0 & a_{n,n-1} & a_{n,n} \end{array}$$

Si aplicamos el primer paso de Gauss, cada valor $a_{i,j}$ va a tener la forma:

$$a_{i,j}^1 = a_{i,j}^0 - \frac{a_{i,1}^0}{a_{1,1}^0} \cdot a_{1,j}^0$$

Queremos ver si se va a mantener la propiedad de matriz diagonal dominante. Si pasa eso, podemos afirmar que se puede realizar Gauss sin pivoteo.

Teniendo en cuenta la ecuación 7, queremos llegar a algo del estilo:

$$\sum_{j=2, j \neq i}^n |a_{i,j}^1| \leq |a_{i,i}^1|$$

Si desarrollamos la primera parte:

$$\begin{aligned} \sum_{j=2, j \neq i}^n |a_{i,j}^1| &= \sum_{j=2, j \neq i}^n \left| a_{i,j}^0 - \frac{a_{i,1}^0}{a_{1,1}^0} \cdot a_{1,j}^0 \right| \leq \\ &\sum_{j=2, j \neq i}^n |a_{i,j}^0| + \sum_{j=2, j \neq i}^n \frac{|a_{i,1}^0|}{|a_{1,1}^0|} \cdot |a_{1,j}^0| \leq \\ &|a_{i,i}^0| - |a_{i,1}^0| + \frac{|a_{i,1}^0|}{|a_{1,1}^0|} \cdot \sum_{j=2, j \neq i}^n |a_{1,j}^0| \leq \\ &|a_{i,i}^0| - |a_{i,1}^0| + \frac{|a_{i,1}^0|}{|a_{1,1}^0|} \cdot (|a_{1,1}^0| - |a_{1,i}^0|) = \\ &|a_{i,i}^0| - \frac{|a_{i,1}^0| \cdot |a_{1,i}^0|}{|a_{1,1}^0|}, \end{aligned}$$

y como tenemos que

$$|a_{i,i}^0| - \frac{|a_{i,1}^0| \cdot |a_{1,i}^0|}{|a_{1,1}^0|} \leq |a_{i,i}^0| - \frac{|a_{i,1}^0| \cdot |a_{1,i}^0|}{|a_{1,1}^0|},$$

entonces

$$\sum_{j=2, j \neq i}^n |a_{i,j}^1| \leq |a_{i,i}^0| - \frac{|a_{i,1}^0| \cdot |a_{1,i}^0|}{|a_{1,1}^0|}.$$

Si miramos el último término del resultado

$$|a_{i,i}^0 - \frac{|a_{i,1}^0| * |a_{1,i}^0|}{|a_{1,1}^0|}| = |a_{i,i}^1|,$$

con esto se concluye

$$\sum_{j=2, j \neq i}^n |a_{i,j}^1| \leq |a_{i,i}^1|$$

Por lo que la matriz mantiene la propiedad de ser diagonal dominante y se podrá realizar EG sin pivoteo.

1.3. Métodos numéricos utilizados

Vamos aplicar eliminación Gaussiana y factorización LU para triangular matrices, resolverlas y luego calcular la isoterma de un horno. A continuación explicaremos los métodos numéricos utilizados, la manera en la que fueron implementados y las consideraciones que tuvimos al armar los datasets. Los métodos mencionados se encuentran en el archivo *algoritmos.cpp* dentro de la carpeta *src*.

1.3.1. Eliminación Gaussiana

Utilizaremos el siguiente pseudocódigo dado en la clase teórica 1 para implementar eliminación Gaussiana:

Algorithm 1 Algoritmo de eliminación Gaussiana

```
1: function GAUSS(m)
2:   for i = 1...n-1 do
3:     for j = i+1 ... n do
4:        $m_{ji} = a_{ji}^{i-1} / a_{ii}^{i-1}$ 
5:       for k = i ... n+1 do
6:          $a_{jk}^i = a_{jk}^{i-1} - m_{ji} a_{ik}^{i-1}$ 
7:       end for
8:     end for
9:   end for
10: end function
```

Su implementación es el método **Gauss**. La función toma como parámetros una matriz m de tamaño $n \times n$ y un vector solución b . Tuvimos la idea de acoplar b a la matriz como última columna con el objetivo de ir modificando los valores de b a la par que se triangula la matriz. Esta matriz extendida resultante la utilizaremos luego para resolver el sistema en el método **resolver_gauss** explicado más adelante.

Su tiempo de cómputo pertenece a $O(n^3)$ debido a que tiene que debe recalcular los coeficientes una y otra vez, triangulando y resolviendo el sistema para cada instancia.

1.3.2. Factorización LU

Para factorización LU, nos guiamos con el pseudocódigo del libro *Análisis Numérico*, páginas 392-393 y con el pseudocódigo de Gauss, modificándolo para que guarde los coeficientes calculados en la matriz L.

Algorithm 2 Algoritmo de factorización LU

```
1: function LU(U,L)
2:   for i = 1...n-1 do
3:     for j= i+1 ... n do
4:        $U_{ji} = a_{ji}^{i-1} / a_{ii}^{i-1}$ 
5:       for k = i ... n+1 do
6:          $a_{jk}^i = a_{jk}^{i-1} - U_{ji}a_{ik}^{i-1}$ 
7:       end for
8:        $L_{ji} = U_{ji}$ 
9:     end for
10:  end for
11: end function
```

Su implementación es el método **LU**. Tomamos como parámetros dos matrices m , L . La matriz m es la matriz a triangular, mientras que la matriz L es una matriz llena de ceros en donde pondremos unos en su diagonal. El algoritmo triangula m y guarda en L los coeficientes que fueron usados. Para resolver el sistema, utilizaremos la función **resolver_gauss** detallada en la siguiente parte.

La gracia de este algoritmo es que puede hacer uso de un sistema de coeficientes con un tiempo perteneciente a $O(n^3)$ computado una única vez, y luego resolver dos sistemas en un tiempo perteneciente a $O(n^2)$ para cada instancia sucesiva que haga uso del mismo sistema inicial.

2. Desarrollo

2.1. Implementaciones en código

2.1.1. Resolución de sistemas

Para resolver los sistemas, armamos el método **resolver_sistema**, el cual toma una matriz m , un booleano *triangular_i* indicando si es triangular superior o inferior, y un vector b . El objetivo es que retorne el vector solución.

2.1.2. Resolución Gauss

resolver_gauss recibe la matriz m a triangular y el vector b . Ejecuta el método de Gauss y después se le pasa la matriz triangulada extendida resultante a **resolver_sistema** para recibir finalmente el vector solución.

2.1.3. Resolución LU

A diferencia de **resolver_gauss**, podemos aprovechar que solo hace falta una única corrida del método LU a un sistema para poder resolver cualquier otro. Por ende, para situaciones donde haya muchas instancias a resolver, realizamos una única vez LU para la primera instancia (por fuera del **resolver_lu**, está en la función **main**). Luego utilizamos las matrices resultantes L y U como parámetros para **resolver_LU**.

2.1.4. Isoterma y peligrosidad

Pensamos en algo sencillo para **buscar la isoterma**: que busque el valor más cercano en todos los radios. Dicho de otra manera, dada una isoterma iso , r_i radios con $i = 0, \dots, m$ y $temp(r_i, \theta)$ la temperatura en el punto (r_i, θ) , vamos a proponer que la isoterma en r_i sea aquel número que tenga la menor diferencia con el valor buscado:

$$isoterma(r_i) = \min(|iso - temperatura(r_i, \theta)|) \quad (8)$$

Para la **peligrosidad**, en una primera instancia pensamos que el sistema se considerará peligroso si algún punto de la isoterma se encuentra dentro del tercio de radios más cercanos a la pared exterior.

Analizando posibles fallas, notamos lo siguiente: si pensamos en un sistema donde uno de los sensores falla y detecta la isoterma cerca a la pared exterior, este sistema sería equivalente a uno en donde toda la isoterma esté cerca de la pared exterior. Por ello, reconsideramos algo más realista, que verifique si un sector de la isoterma está en el tercio más cercano; es decir, si un ángulo de la isoterma se encuentra en el tercio más cercano, y *también* sus ángulos aledaños lo están. En este caso, se puede considerar el que no se trata de un fallo de un sensor, sino una posible situación de riesgo.

2.2. Experimentos

A continuación vamos a detallar los experimentos realizados y los datasets creados para cada uno. Los experimentos fueron corridos en una computadora con las siguientes especificaciones:

- Ubuntu 20.04.4 LTS 64-bit, 14,0 GiB Memory, Intel Core i5-3570,
- Ubuntu 20.04.4 LTS 64-bit, 15,6 GiB Memory, AMD Ryzen 3 2200g.

Tomamos las siguientes decisiones a la hora de crear los datasets para los experimentos:

1. Para los valores de los radios r_i y r_e decidimos tomar los mismos valores constantes a lo largo de los experimentos debido a que es posible armar sistemas equivalentes para distintos valores de los radios,
2. El valor de la isoterma será de 500, tal como se menciona en el enunciado,
3. Los valores que iremos modificando serán la cantidad de radios y cantidad de ángulos, los valores de las temperaturas de las paredes internas y externas, y la cantidad de instancias.

2.2.1. Experimento 1: Prueba de modelo

La idea del *primer experimento* consiste en encontrar los valores de m y n para obtener la "mejor" discretización: que muestre un gráfico lo suficientemente fino y que conseguirlo no sea muy costoso temporalmente.

Para este experimento solo vamos a utilizar Gauss dado que al tener una sola instancia por archivo, no podemos aprovechar las ventajas del método LU. Una sola instancia de LU sería lo mismo que resolver el sistema con Gauss (con el adicional de algunas operaciones extra).

Nuestro objetivo será encontrar una discretización tal que la posición calculada de la isoterma sea lo más cercana posible a la real, sin excedernos computando instancias cuyo rendimiento sea decreciente considerando el tiempo de computo de la misma y la precisión que se obtenga. Por ello, para este dataset dejaremos fijo el valor de n (ángulos) y modificaremos los valores de m , incrementando la cantidad de radios. Nuestra **hipótesis** será que con $n = 40$, podremos encontrar un valor de m tal que la discretización es lo suficientemente fina como para estar cerca de la isoterma real, y que a su vez calcularla no tarda demasiado.

Para decidir qué m elegiremos, calcularemos las posiciones de las isotermas de los hornos resultantes, iremos comparando los resultados con la instancia anterior y analizaremos a partir de qué momento la diferencia se hace despreciable.

- Este primer dataset contempla una discretización del horno con temperaturas fijas $T_i = 1500$ y $T_e = 200$,
- $n = 40$,
- $m \in \{x : x = i \cdot 2, i = 2 \dots 20\}$,
- $ninst = 1$, con tantos archivos como las instancias que surgen de los pares (m, n) ,
- $isoterma = 500$,
- $r_i = 10$,
- $r_e = 15$.

2.2.2. Experimento 2: Prueba de cómputo

Una vez conseguida la discretización buscada, nos propusimos armar un *segundo experimento* con el objetivo de decidir cuál método de resolución es el óptimo entre Gauss y LU. Para ello, armamos un dataset en donde el horno va cambiando con el tiempo, es decir, el archivo en cuestión tiene 30 instancias del mismo horno.

Adicionalmente, las temperaturas de las paredes respetan una distribución normal para simular un posible error de medición en los sensores con $\sigma = \frac{\mu}{50}$.

Con esto, creamos el **Dataset 2: Temperaturas que cambian con el tiempo**, con las siguientes características:

- $n = 40$,
- $m = 30$,
- $ninst = 30$,
- $isoterma = 500$,
- $r_i = 15$,
- $r_e = 5$,
- $T_i \sim N(1500, 30)$,
- $T_e \sim N(200, 4)$,

Para este dataset aplicaremos Gauss y LU, con la **hipótesis** de que LU va a tardar mucho menos que Gauss.

2.2.3. Experimento 3A: Prueba de riesgo

Investigando sobre el funcionamiento de altos hornos, encontramos que los materiales que constituyen sus paredes son resistentes al calor. Teniendo esto en cuenta, pensamos en distintos motivos por los cuales podría haber un accidente, y pensamos un *tercer experimento* en donde tenemos un alto horno con paredes construidas de materiales de baja calidad, los cuales tienen un límite de resistencia y utilizan un sistema de refrigeración pero el mismo *falla*, y mitad del horno comienza a incrementar su temperatura con el pasar del tiempo.

Nuestro objetivo es ver cómo cambia la isoterma cuando hay una diferencia de temperaturas irregular en una zona considerable del horno a través del tiempo y observar cómo afecta este cambio en el sistema.

Vamos a utilizar la discretización hallada en el primer experimento, con el método LU, dado que vamos a usar el mismo sistema con los mismos coeficientes aprovechando lo hallado en los dos experimentos anteriores.

Utilizar LU será más eficiente, dado que solamente la primera instancia se ejecuta con tiempo $O(n^3)$, y todas las posteriores se ejecutaran en $O(n^2)$.

Nuestra **hipótesis** es que creemos que la isoterma va a perder su forma circular, y empieza a deformarse hacia la zona de temperaturas más altas, acercándose al borde con temperaturas más altas con el paso del tiempo de la siguiente forma:

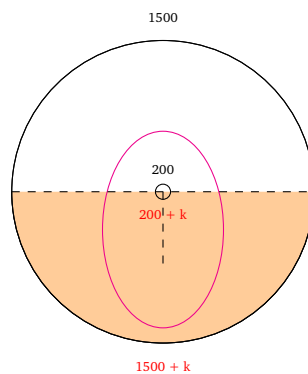


Figura 3: Esbozo gráfico de la hipótesis

donde k es el incremento de temperatura que se percibe. Para corroborar esto creamos el **Dataset 3: Horno con sistema de refrigeración no funcional** de las siguientes características:

- $n = 40$,
- $m = 30$,
- $n_{inst} = 50$,
- $isoterma = 500$,
- $r_i = 10$,
- $r_e = 15$,
- $T_i = 1500$,
- $T_e = 200$,
- $k = 10$, este k es el incremento de temperatura que sufre la mitad del horno en cada instancia.

2.2.4. Experimento 3B: Prueba de sensado

En este experimento queremos atestiguar cómo influye en el sensado de la isoterma el que haya sensores defectuosos en el horno. Para lograr esta situación, contemplamos que en sensores tomados al azar, el desvío que manejen sea mucho mayor que el de los sensores en correcto funcionamiento.

Creamos el **Dataset 4: Horno con sensores fallados** con las siguientes características, donde SF es la temperatura que manejan los sensores fallados:

- $n = 40$,
- $m = 30$,
- $ninst = 30$,
- $isoterma = 1000$,
- $r_i = 15$,
- $r_e = 5$,
- $T_i \sim N(1500, 30)$,
- $T_e \sim N(500, 4)$,
- $SF \sim N(500, 100)$,

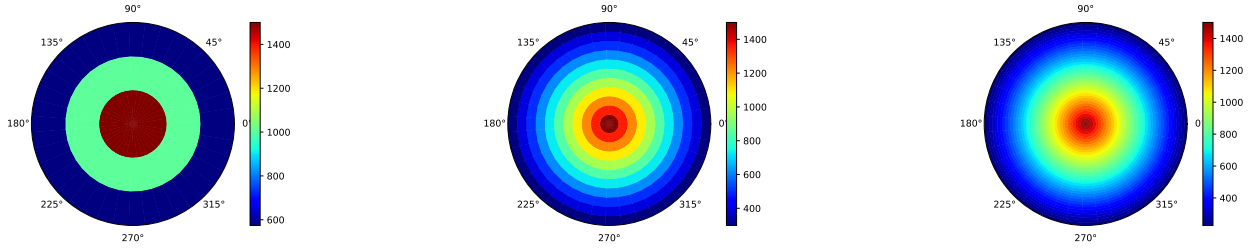
Aclararemos que para cada generación del dataset, dichos sensores defectuosos son elegidos al azar, y no cambian a lo largo de las instancias.

Para este dataset aplicaremos LU, con la **hipótesis** de que se podrán distinguir perturbaciones en la forma de la isoterma en los sensores defectuosos.

3. Resultados

3.1. Análisis del Experimento 1

Observemos los gráficos de calor de los hornos resultantes:



(a) Gráfico de calor con $m = 4$.

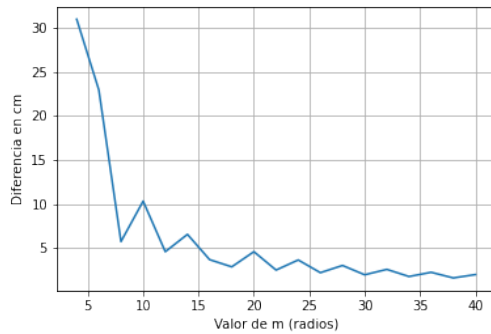
(b) Gráfico de calor con $m = 12$

(c) Gráfico de calor con $m = 40$

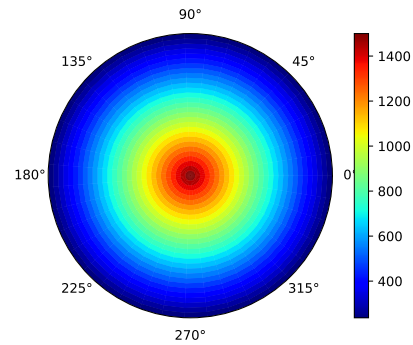
Figura 4: Gráficos de nitidez de la discretización (en grados Celsius).

Notamos que a medida que incrementamos el valor de m los gráficos se hacen cada vez más detallados, pero ello aún no es lo suficientemente convincente para seleccionar el m adecuado y dichos gráficos no son los indicados para ilustrar la precisión que ofrece cada discretización.

Para ver a partir de qué momento hacer una discretización más fina no vale la pena fuimos comparando la norma 2 de los radios donde se ubicaba la isoterma hallada en cada instancia con la de la instancia anterior.



(a) Diferencias de de las normas entre isotermas



(b) Gráfico de calor con $m = 30$

Figura 5: Gráfico de diferencias entre radios y gráfico de calor del radio elegido.

Se puede notar que a partir de un valor $20 \leq m \leq 30$, ya no se logra un incremento considerable de la precisión. Con esto concluimos que a partir de $m = 30$ con $n = 40$ la diferencia entre las posiciones de las isotermas se vuelve lo suficientemente pequeña como para poder tener una discretización lo suficientemente certera y hacer experimentos cómodamente.

Dicho esto, podríamos pensar en situaciones en donde esta propuesta falla. Por ejemplo, ¿qué pasaría si hay una explosión no muy grande en un punto específico del horno? ¿Sería visible en la discretización propuesta? Tal vez un cambio no muy significativo en las temperaturas no sea detectado. Luego, ¿cómo se arreglaría este problema? Si aumentáramos la cantidad de radios en vez de ángulos quizás se solucione. Esta línea de pensamiento no la continuaremos en este informe, pero podría ser algo a incluir en un futuro trabajo.

3.2. Análisis del Experimento 2

Haciendo uso de la discretización encontrada en el experimento previo, es de interés distinguir qué tanto tiempo consumen los algoritmos de eliminación Gaussiana y factorización LU. Cada instancia del horno fue calculada 5 veces, por lo que graficaremos las medias de los tiempos que tardó cada una con sus desvíos.

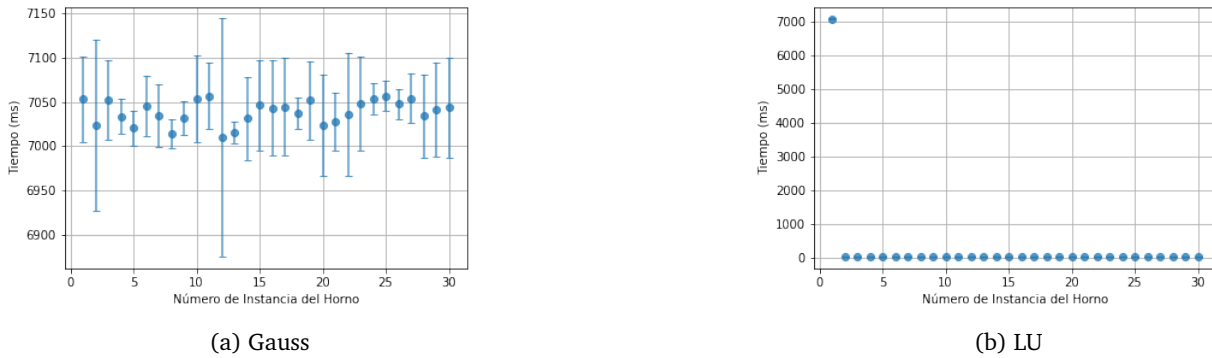


Figura 6: Scatterplots de las medias de los tiempos para cada método.

Analizaremos con más detalle los tiempos de cómputo de factorización LU, quitando la primera instancia:

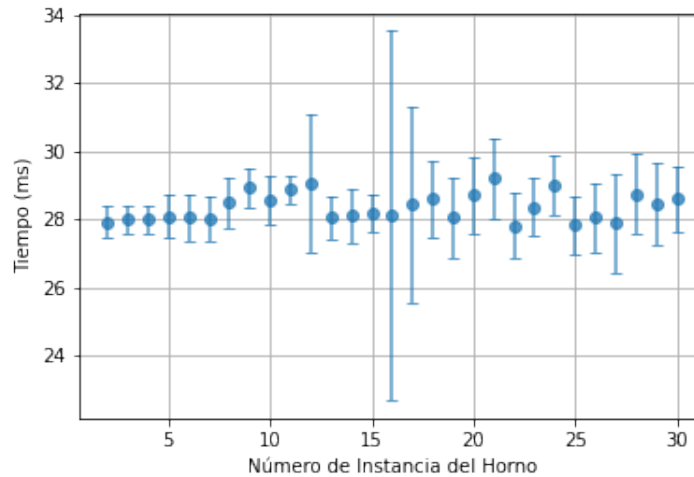


Figura 7: Scatterplot de factorización LU sin la primera instancia

Es posible observar que una única primer instancia del algoritmo de factorización LU tiene un tiempo de cómputo elevado, mientras que las sucesivas instancias requieren un tiempo de cómputo despreciable en comparación. Por otro lado, se puede ver que el algoritmo de eliminación Gaussiana requiere alrededor de 7 segundos para cada instancia luego de la primera. Contrastándolo con LU, este último tarda al rededor de 20 milisegundos, una vez computada la factorización a través de Gauss.

Analizando ambos métodos en un mismo gráfico:

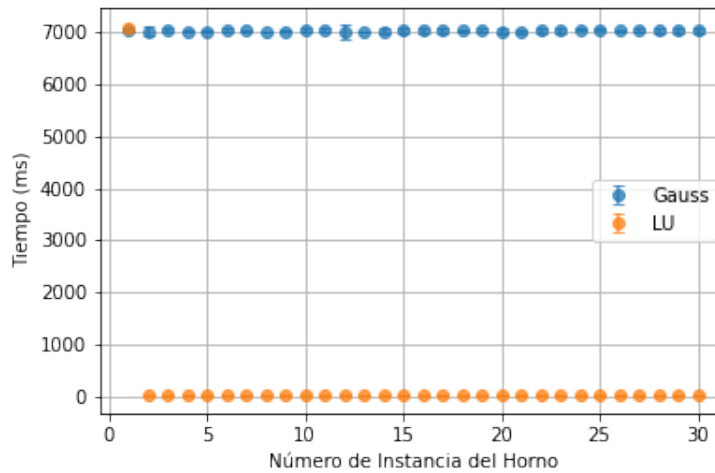


Figura 8: Comparación de tiempo entre Gauss y LU

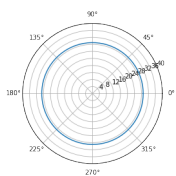
Aquí se puede observar con más claridad la gran diferencia entre Gauss y LU. Con esto, es claro que cuando se requiere trabajar sobre más de una instancia con un mismo sistema, la *Factorización LU* resulta mucho menos costosa. Mientras que si sólo se utiliza una única instancia (es decir, se requiere resolver un sistema una única vez), es recomendable utilizar la *Eliminación Gaussiana*.

Reflexionando sobre los resultados, no es sorprendente ver que LU es mucho más rápido que Gauss. Conocíamos el tiempo que iba a tardar cada uno en términos computacionales, por lo que no esperábamos resultados distintos. Lo que sí nos impactó fue la gran diferencia de tiempo entre un método y otro a la hora de probarlos. No pensamos que el cambio sería tan grande: Gauss tarda alrededor de 7 segundos para cada instancia, mientras que LU tarda menos de un segundo. Esto da cuenta de lo importante que es analizar los tiempos de cómputo de un algoritmo y tener en cuenta los recursos disponibles.

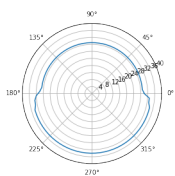
3.3. Análisis del Experimento 3

3.3.1. Experimento 3.A

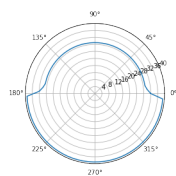
Observemos los gráficos de la isoterma para los resultados del experimento 3 a medida que las temperaturas van aumentando hacia una mitad del horno:



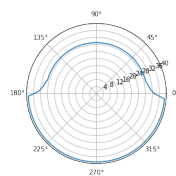
(a) Gráfico de la isoterma en la instancia 1.



(b) Gráfico de la isoterma en la instancia 16.



(c) Gráfico de la isoterma en la instancia 32.



(d) Gráfico de la isoterma en la instancia 50.

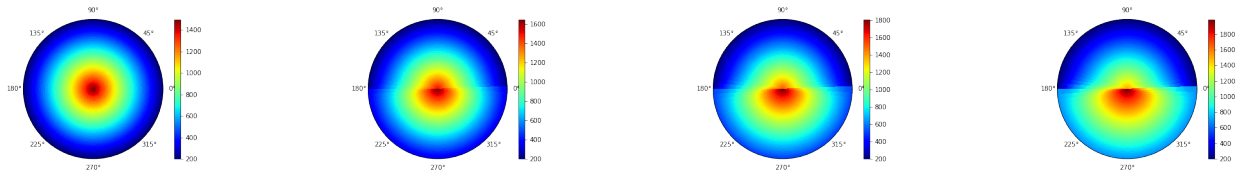
Figura 9: Gráficos de las isotermas.

Podemos observar que hay una deformación de la isoterma, sin embargo, esta es diferente a la que se esperaba. La isoterma avanza hacia la mitad inferior del horno y mantiene una forma circular.^a diferencia de la forma ovalada que esperábamos. La razón por la cual mantiene su silueta es porque armamos el dataset de manera tal que todas las temperaturas externas de una mitad del horno incrementan uniformemente, por lo que la isoterma mantiene en general su estructura.

Además, podemos notar que la posición de la isoterma en el segundo gráfico es casi igual a su posición en el tercero. Con esta observación, podríamos concluir que a partir de cierto punto, aunque las temperaturas continúen aumentando, la isoterma no cambia de forma.

Por otra parte, si tomamos en cuenta nuestra definición de peligrosidad, nos encontramos frente a un sistema altamente peligroso, en donde la isoterma se encuentra en todo el borde de una mitad de la pared exterior.

Ahora analicemos los gráficos de calor correspondientes:



(a) Gráfico de la calor en la instancia 1. (b) Gráfico de la calor en la instancia 16. (c) Gráfico de la calor en la instancia 32. (d) Gráfico de la calor en la instancia 50

Figura 10: Gráficos de calor en grados Celcius.

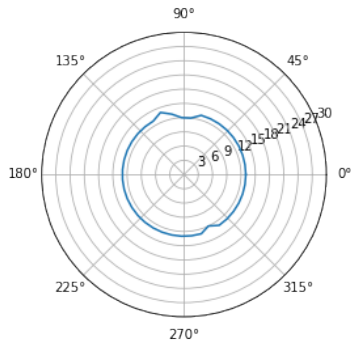
Aquí podemos detallar lo comentado anteriormente: el incremento de calor es parejo a lo largo de la mitad inferior. A su vez, podemos notar con facilidad las dos mitades: los bordes de una mitad presentan temperaturas de 200°C y la otra 800°C. En la vida real, esto no sería posible, habría un gradiente de temperaturas entre las mitades.

Si nos detenemos por un minuto y pensamos en este experimento más detalladamente, los gráficos muestran una situación en donde toda una mitad del horno estaría presentando temperaturas de alrededor de 500°C. El límite de resistencia al calor de algunos materiales no supera los 450°C, mientras que otros resisten hasta 1315°C, por lo que si pensamos que los materiales no son los mejores, no estaríamos equivocados en asumir que la pared se esta destruyendo. En tiempos modernos, teniendo en cuenta el avance tecnológico y los controles actuales en mega construcciones, es algo casi imposible. El único motivo por el cual podría darse esta situación sería si estuviésemos frente a un horno muy primitivo, y aún así sería complicado sabiendo que los altos hornos más antiguos se datan desde 1000-500 antes de Cristo, y estamos en el 2022. Dicho esto, una situación más realista sería un fallo en un pequeño sector del horno y no toda una mitad.

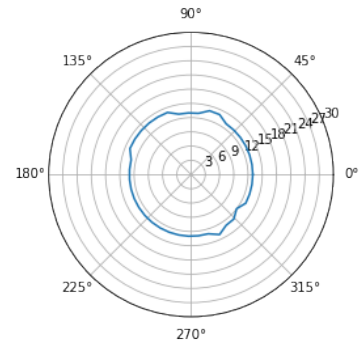
Teniendo en cuenta lo analizado anteriormente, concluimos que la forma de la isoterma depende de las temperaturas del sistema. A su vez, el dataset que planteamos nos presenta con resultados poco realistas. En un alto horno en la vida real, el cambio de temperaturas se daría con un gradiente mucho más suave que teniendo, por ejemplo, un sensor que registra 200°C al lado de uno que registra 500°C.

3.3.2. Experimento 3.B

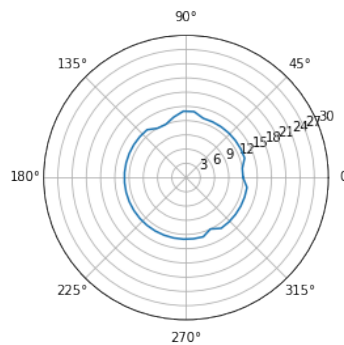
Utilizando el dataset 4, analizaremos las isothermas resultantes:



(a) Isotherma de la instancia 7



(b) Isotherma de la instancia 17



(c) Isotherma de la instancia 29

Figura 11: Isothermas de las instancias del horno con sensores defectuosos.

Podemos observar que en determinados puntos de cada isoterma se perciben deformaciones de la misma. Dichos puntos coinciden con las posiciones de los sensores defectuosos que contemplan un error más amplio que el esperado de un sensor que funciona correctamente.

De esta forma, se perciben estos cambios de la isoterma erróneos a través de las distintas instancias, cuando en verdad la isoterma no estaría sufriendo dichas perturbaciones.

A diferencia del experimento 3A, los resultados de este experimento 3B se asemejarían más a lo que puede ocurrir en la realidad. En las instancias que generamos nosotros, ninguno de los hornos entraría dentro de la circunstancia de peligrosidad que habíamos formulado, dado que puede verse que la isoterma obtenida no difiere mucho de la idealisoterma circular.

De todos modos, estos gráficos de la isoterma podrían ayudar a determinar cuál es el problema que se tiene en el alto horno: si los sensores efectivamente están fallados y a eso se deben las detecciones de temperaturas mayores/menores a las que realmente son y el horno se encuentra en buenas condiciones, o si se confirma que los sensores funcionan correctamente, entonces puede ser que haya problemas con la integridad estructural de la pared del alto horno en los lugares donde la isoterma se difiere del círculo que debería ser la isoterma.

4. Conclusiones

En este trabajo, presentamos dos formas de resolver un sistema de ecuaciones para aplicarlos como métodos de control de un alto horno y realizamos distintos experimentos para ver su comportamiento.

Para la discretización del horno, encontramos que a partir de cierto valor para m y n ya no es conveniente seguir aumentándolos. A su vez, notamos que si tenemos una única instancia de un horno, aplicar Gauss y LU tardan casi lo mismo computacionalmente, pero si queremos analizar varias instancias, LU es muchísimo más óptimo. Por último, el cambio en la posición de la isoterma implica un incremento de las temperaturas hacia las paredes externas, lo cual indicaría que el sistema se encuentra comprometido.

Como último punto, a lo largo del informe reflexionamos sobre algunas ideas que podríamos implementar y analizar en un futuro:

- Testear si nuestra idea de que r_i y r_e no afectan mucho en la discretización,
- Encontrar alguna optimización para la búsqueda de la isoterma,
- En una de las clases teóricas, vimos que las matrices bandas tienen una propiedad muy particular: Una matriz banda cumple que en su factorización LU, la L es exactamente la banda inferior con 1s en su diagonal principal, y la U es igual a la banda superior con los valores originales en la diagonal principal. Dicho esto, podríamos pensar en alguna forma de aprovecharlo en los algoritmos,
- Pensar en datasets con incrementos de temperaturas diferentes al propuesto en el experimento 3 y analizar qué sucede,
- Agregar a los notebooks de análisis y ejecución una forma de indicar qué instancia está ejecutando.

5. Referencias

1. Alto horno: [Link Wikipedia](#)
2. Isoterma: [Link Wikipedia](#)
3. Algoritmos y propiedades: Análisis Numérico, Richard L. Burden, et.al., Cenage Learning.