



Trabajo Práctico 1

Alto Horno

Métodos Numéricos

| Integrante | LU | Correo Electrónico |
|--------------------|--------|--------------------------------|
| Teo Kohan | 385/20 | teo.kohan@gmail.com |
| Martin Santesteban | 397/20 | martin.p.santesteban@gmail.com |

Abstract

En este informe se trata el problema de determinar la posición de la isoterma en la pared de un alto horno. Para ello se calculó la temperatura en cada punto de una discretización del mismo mediante la resolución de un sistema de ecuaciones lineales. Estas fueron resueltas aplicando los algoritmos de Eliminación Gaussiana y Factorización LU . Además se idearon distintos algoritmos para lograr aproximar la isoterma en el modelo propuesto. Adicionalmente, se estudiaron distintas formas de medir el riesgo que presenta la proximidad de la misma a la pared externa del horno. Finalmente, se contemplo la complejidad computacional de los métodos de resolución del sistema a resolver en relación a la granularidad del modelo y cantidad de subinstancias. Como resultado se observó la diferencia entre la interpolación lineal y bilineal, además de los tiempos de ejecución notablemente mejores de la Factorización LU con respecto a la Eliminación Gaussiana para muchas subinstancias.

Keywords— Métodos numéricos, Sistemas de ecuaciones, Factorización LU , Eliminación Gaussiana.

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>



Índice

| | | |
|----------|---|-----------|
| 1 | Introducción | 3 |
| 1.1 | Motivación | 3 |
| 1.2 | Modelo | 3 |
| 1.3 | Objetivo | 4 |
| 1.4 | Eliminación Gaussiana y Factorización LU | 4 |
| 1.5 | Supuestos | 4 |
| 1.6 | Modelo Computacional | 4 |
| 2 | Descripcion general del sistema | 6 |
| 2.1 | Generalidades | 6 |
| 2.2 | Propiedades | 8 |
| 2.2.1 | Dominancia Diagonal | 8 |
| 2.2.2 | Eliminación Gaussiana | 10 |
| 3 | Métodos numéricos utilizados | 12 |
| 3.1 | Cálculo de isoterma. | 12 |
| 3.1.1 | Vecino más Cercano | 13 |
| 3.1.2 | Interpolación Lineal | 13 |
| 3.1.3 | Interpolación Bilineal | 13 |
| 3.2 | Calculo de riesgo | 14 |
| 3.2.1 | Riesgo Promedio | 14 |
| 3.2.2 | Riesgo Distancia Mínima | 14 |
| 3.3 | Análisis de ambas mediciones | 15 |
| 4 | Experimentación | 15 |
| 4.1 | Hipótesis generales | 15 |
| 4.2 | Instancias | 16 |
| 4.2.1 | Caso Uniforme (k) | 16 |
| 4.2.2 | Caso Pico (p, k) | 16 |
| 4.2.3 | Caso Distribución Normal (μ, σ^2) | 16 |
| 4.2.4 | Caso Oscilante | 16 |
| 4.3 | Experimentos | 16 |
| 4.3.1 | EG vs. LU , número de subinstancias creciente | 17 |
| 4.3.2 | EG vs. LU , $n \cdot m$ creciente. | 18 |
| 4.3.3 | Riesgo Caso Pico | 18 |
| 4.3.4 | Interpolaciones isoterma | 20 |
| 5 | Conclusiones | 21 |

1 Introducción

1.1 Motivación

Un *alto horno* es un tipo de horno que hace uso de un caudal superior de oxígeno para descomponer hierro entre otros minerales en distintos estratos y subproductos. Para su operación es necesario alcanzar temperaturas muy elevadas y por lo tanto es requerido recubrirlo con algún material extremadamente aislante.

Uno de los peligros a controlar en la correcta operación es la identificación de *hot spots* en la coraza para evitar desgastes en la estructura o potencialmente accidentes.

Para esto se plantea monitorear la línea isotérmica de 500°C como indicador de la efectividad de la aislación actual de la coraza.

1.2 Modelo

Sean $R_e \in \mathbb{R}$ el radio exterior y $r_i \in \mathbb{R}$ el radio interior de la pared aislante. Sea

$$T(r, \theta) : [0, R_e] \times [0, 2\pi) \rightarrow \mathbb{R}$$

la función que da la temperatura en grados celsius en el punto dado por las coordenadas polares (r, θ) , siendo r el radio y θ el ángulo de dicho punto como muestra la Figura 1.

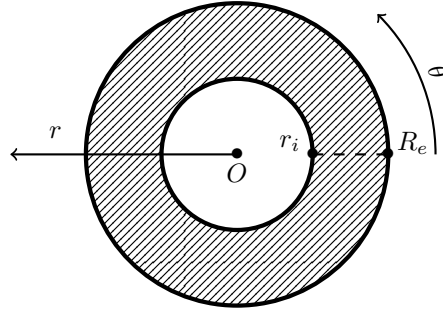


FIGURA 1: Representación esquemática de un alto horno.

Sea la temperatura interna homogénea $T_i \in \mathbb{R}$ (Sector O en la Figura 1) y $T_e : [0, 2\pi) \rightarrow \mathbb{R}$ la función de temperatura en la coraza exterior del horno luego se tienen las ecuaciones (2) y (3) del enunciado.

Además cuando el sistema se encuentra en reposo también se satisface la ecuación diferencial (1) del enunciado.

Esta ecuación, la ecuación del calor de Laplace, adaptada a coordenadas polares, permite resolver el sistema para cualquier par (r, θ) , tal que $r_i \leq r \leq R_e$ y $0 \leq \theta < 2\pi$.

1.3 Objetivo

En este trabajo esperamos resolver el problema de encontrar la línea isotérmica de k grados para un horno de radio interno r_i y radio externo R_e teniendo el valor T_i y la función T_e y utilizando métodos matriciales como eliminación Gaussiana, y factorización LU . Además queremos hallar una buena norma para el peligro que supone la isoterma obtenida para la integridad del horno.

1.4 Eliminación Gaussiana y Factorización LU

Para la resolución de sistemas lineales de ecuaciones de la forma $Ax = b$, donde $A \in \mathbb{R}^{n \times n}$ y $b \in \mathbb{R}^n$ se utilizará el método de Eliminación Gaussiana (sin pivoteo), y, dadas las características de las matrices con las que tratamos también haremos uso de la factorización LU de A .

El método de Eliminación Gaussiana[1, pp. 358–364] calcula el sistema equivalente $A^*x = b^*$, donde A^* es una matriz triangular superior. Se opera sobre la matriz aumentada $(A | b)$ secuencialmente utilizando la fila i -ésima como pivote con el objetivo de anular la i -ésima columna restándole a las demás filas un múltiplo del pivote (multiplicación por matrices elementales). Finalmente se obtiene la matriz triangular superior A^* equivalente a A y su vector aumentado b^* . Notar que la Eliminación Gaussiana puede llevarse a cabo si y solo si nunca se anula el elemento de la diagonal de la fila pivote (la fila que estamos utilizando para anular una columna).

La Factorización LU [1, p. 406] de A , es tal que $A = LU$ donde L es triangular inferior con unos en la diagonal y U es triangular superior. El proceso de factorización es muy similar a la Eliminación Gaussiana, donde la matriz resultante es U y L almacena los multiplicadores necesarios para triangular a A . Luego $Ax = b$ es equivalente a $LUx = b$, entonces si $Ux = y$ tenemos que $Ly = b$. Como beneficio si quisieramos resolver otro sistema $Ax = b'$ el grueso del cómputo ya ha sido realizado en la factorización $LUx = b'$ y únicamente nos queda resolver dos sistemas triangulares.

El costo computacional de ambos algoritmos pertenece a $\mathcal{O}(n^3)$, sin embargo la constante multiplicativa de la Factorización LU es $\frac{2}{3}$ mientras la de Eliminación Gaussiana es $\frac{3}{3}$.

1.5 Supuestos

- $0 < r_i < R_e$.
- Para todo ángulo θ se tiene que $T_e(\theta) \leq T_i$.

1.6 Modelo Computacional

Para resolver este problema computacionalmente, necesitamos discretizar el dominio del problema (el sector rayado de la figura 1), esto es, realizar una partición del área total en puntos discretos. Contamos con sensores equiespaciados cada $\Delta\theta$ radianes sobre la pared interna y externa del horno que reportan la temperatura.

Entonces sea \mathcal{R} una partición en $m - 1$ radios $r_j = r_i + j \cdot \Delta r$ para todo $1 \leq j \leq m - 1$ donde $\Delta r = \frac{R_e - r_i}{m}$ y Θ una partición en n ángulos $\theta_k = k \cdot \Delta\theta$ para todo $0 \leq k < n$ donde $\Delta\theta = \frac{2\pi}{n}$.

Y sean $I_r = \{1, \dots, m-1\}$ e $I_\theta = \{0, \dots, n-1\}$ los conjuntos índice de las particiones \mathcal{R} y Θ .

Por simplicidad definamos

$$t_{jk} := T(r_j, \theta_k) \quad (7)$$

para todo $j \in I_r$, $k \in I_\theta$ la temperatura en el punto dado por las coordenadas polares (r_j, θ_k) donde r_j es el radio y θ_k es el ángulo, adicionalmente $t_{0k} := T_i$ y $t_{mk} := T_e(\theta_k)$ para todo $k \in I_\theta$. Por lo tanto facilitemos la siguiente definición

$$t_j := (t_{j0}, t_{j1}, \dots, t_{j,n-1}) \quad (8)$$

para todo $0 \leq j \leq m$.¹ Debemos despejar todo t_j para $j \in I_r$, tomando la ecuación (1) del enunciado y convirtiéndola en un conjunto de ecuaciones lineales dependientes de los t_{jk} .

Aproximamos las derivadas parciales de T en la ecuación (1) del enunciado utilizando las siguientes fórmulas de diferencias finitas derivadas de las ecuaciones (4), (5) y (6) del enunciado

$$\frac{\partial^2 T(r, \theta)}{\partial r^2}(r_j, \theta_k) \approx \frac{t_{j-1,k} - 2t_{jk} + t_{j+1,k}}{(\Delta r)^2} \quad (9)$$

$$\frac{\partial T(r, \theta)}{r \cdot \partial r}(r_j, \theta_k) \approx \frac{t_{jk} - t_{j-1,k}}{r \cdot \Delta r} \quad (10)$$

$$\frac{\partial^2 T(r, \theta)}{r^2 \cdot \partial \theta^2}(r_j, \theta_k) \approx \frac{t_{j,k-1} - 2t_{jk} + t_{j,k+1}}{(r \cdot \Delta \theta)^2}. \quad (11)$$

Donde debido a la topología del horno $t_{j,-1} := t_{j,n-1}$ y $t_{jn} := t_{j0}$ para todo $j \in I_r$.

Por lo tanto reinterpretando la ecuación (1) del enunciado de forma finita

$$E(j, k) := \frac{t_{j-1,k} - 2t_{jk} + t_{j+1,k}}{(\Delta r)^2} + \frac{t_{jk} - t_{j-1,k}}{r \cdot \Delta r} + \frac{t_{j,k-1} - 2t_{jk} + t_{j,k+1}}{(r \cdot \Delta \theta)^2} \approx 0. \quad (12)$$

A partir de este punto se intentará armar un sistema de ecuaciones lineales del tipo $Ax = b$, donde el vector solución x hará referencia a las temperaturas de la discretización.

¹Notar que $t_{jk} = (t_j)_k$ por lo tanto la notación es consistente.

2 Descripción general del sistema

2.1 Generalidades

Para encontrar la temperatura en cada punto de la discretización, recordemos que para cada par $(j, k) \in I_r \times I_\theta$ se cumple la ecuación (12) que puede ser reescrita de la siguiente forma

$$\begin{aligned}
 & t_{j-1,k} \cdot \left(\frac{1}{(\Delta r)^2} - \frac{1}{r_j \cdot \Delta r} \right) + \\
 & t_{j,k-1} \cdot \left(\frac{1}{r_j^2 \cdot (\Delta \theta)^2} \right) + \\
 & t_{jk} \cdot \left(\frac{-2}{(\Delta r)^2} + \frac{1}{r_j \cdot \Delta r} - \frac{2}{r_j^2 \cdot (\Delta \theta)^2} \right) + \\
 & t_{j,k+1} \cdot \left(\frac{1}{r_j^2 \cdot (\Delta \theta)^2} \right) + \\
 & t_{j+1,k} \cdot \left(\frac{1}{(\Delta r)^2} \right) \approx 0.
 \end{aligned} \tag{13}$$

Esta ecuación se cumple para todo punto de la discretización con $r_i < r_j < R_e$, por lo tanto se tienen $n \cdot (m-1)$ puntos donde esta ecuación debe ser válida. Pretendemos averiguar el valor de t_k para $0 \leq k \leq m$ por lo tanto tomemos en consideración al siguiente vector $x \in \mathbb{R}^{n \cdot (m+1)}$

$$x^t = [t_{00} \quad t_{01} \quad \dots \quad t_{0,n-1} \quad t_{11} \quad \dots \quad t_{m,n-1}] \tag{14}$$

El vector x está ordenado primero en función al valor del radio, y luego en función al ángulo, entonces sabemos que $x_{j \cdot n + k} = t_{jk}$. Los primeros n valores están asociados a las temperaturas de t_0 , mientras que los últimos n valores están asociados a las temperaturas de t_m . Donde por definición $t_0 = [T_i \quad T_i \quad \dots \quad T_i]$ y $t_m = [T_e(\theta_0) \quad T_e(\theta_1) \quad \dots \quad T_e(\theta_{n-1})]$.

Consideremos la siguiente partición de x

$$x^t = \left[\begin{array}{c|c|c} t_0 & x^* & t_m \end{array} \right]$$

donde $t_0, t_m \in \mathbb{R}^n$ son datos y $x^* \in \mathbb{R}^{n \cdot (m-1)}$ son las temperaturas incógnita.

Debemos armar la matriz de coeficientes $A \in \mathbb{R}^{n(m+1) \times n(m+1)}$ adecuada para que se cumpla la ecuación (12) para cada incógnita. Empezamos por definir a cada fila tal que el producto de la i -ésima fila de A y x tengamos como resultado la ecuación $E(\lfloor i/n \rfloor, \text{resto}_n(i))$. Notemos que para que esto se cumpla, la i -ésima fila de A estará compuesta solo por ceros, excepto por cinco elementos que serán los coeficientes respectivos a la ecuación correspondiente. Esto podemos lograrlo teniendo el mismo orden de x descripto en (14) con los coeficientes en las filas de A . Por lo tanto, si

$t_{jk} = x_{j \cdot n + k} = x_i$, entonces se cumple que

$$\begin{aligned}
 t_{j-1,k} = x_{i-n} &\Rightarrow a_{i,i-n} = \frac{1}{(\Delta r)^2} - \frac{1}{r_j \cdot \Delta r} \\
 t_{j,k-1} = x_{j \cdot n + \text{resto}_n(k-1)} &\Rightarrow a_{i,j \cdot n + \text{resto}_n(k-1)} = \frac{1}{r_j^2 \cdot (\Delta \theta)^2} \\
 t_{jk} = x_i &\Rightarrow a_{ii} = \frac{-2}{(\Delta r)^2} + \frac{1}{r_j \cdot \Delta r} - \frac{2}{r_j^2 \cdot (\Delta \theta)^2} \\
 t_{j,k+1} = x_{j \cdot n + \text{resto}_n(k+1)} &\Rightarrow a_{i,j \cdot n + \text{resto}_n(k+1)} = \frac{1}{r_j^2 \cdot (\Delta \theta)^2} \\
 t_{j+1,k} = x_{i+n} &\Rightarrow a_{i,i+n} = \frac{1}{(\Delta r)^2}
 \end{aligned}$$

Luego, el resto de los elementos para la i -ésima fila son cero.

Con respecto a las matriz de coeficientes para radios igual a 0 y m , la temperatura ya es dato consecuentemente las primeras y últimas n filas tendrán coeficientes nulos. Finalmente, la matriz A tendrá sus coeficientes ubicados de una forma similar a la Figura (2), formando una banda.

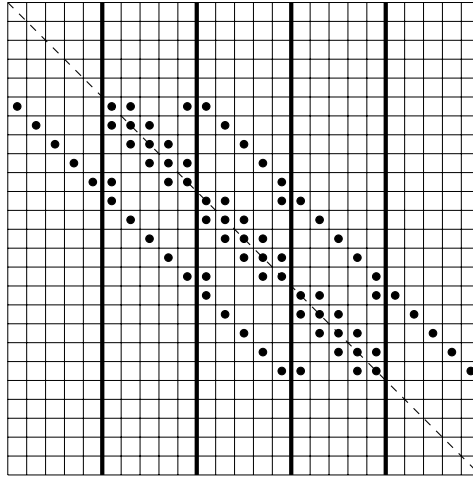


FIGURA 2: Ubicación de coeficientes no nulos en una matriz con $n = 5$, $m = 4$ y $n \cdot (m + 1) = 25$.

Sacamos las primeras y últimas n filas, debido que todos los coeficientes correspondientes son nulos.

Teniendo la matriz de coeficientes, procedemos a resolver el sistema. El objetivo es averiguar las temperaturas desconocidas, x^* , planteamos un sistema de ecuaciones lineales donde x^* es el vector solución. Para despejar x^* particionamos A de la siguiente manera

$$A = \left[\begin{array}{c|c|c} C & A^* & D \end{array} \right]$$

Con $C, D \in \mathbb{R}^{n(m-1) \times n}$ y $A^* \in \mathbb{R}^{n(m-1) \times n(m-1)}$ luego

$$Ax = \begin{bmatrix} C \\ A^* \\ D \end{bmatrix} \begin{bmatrix} t_0 \\ x^* \\ t_m \end{bmatrix} = C \cdot t_0 + A^* x^* + D \cdot t_m \approx 0$$

$$A^* x^* \approx -C \cdot t_0 - D \cdot t_m$$

Se puede observar en la Figura (3) la ubicación de los coeficientes en A^* .

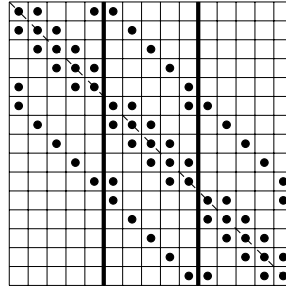


FIGURA 3: Ubicación de coeficientes no nulos en la matriz de la Figura (2) luego de recortar las primeras y últimas n filas y columnas.

Para resolver el sistema $A^* x^* = -C \cdot t_0 - D \cdot t_m$ aplicaremos el algoritmo de eliminación Gaussiana sobre A^* . Para garantizar su funcionamiento y consecuentemente el de la Factorización LU demostraremos que, sabiendo que A^* es una matriz inversible, A^* es diagonal dominante y Eliminación Gaussiana es aplicable.

2.2 Propiedades

Vamos a querer demostrar la siguiente proposición

Proposición 1. Sea $A \in \mathbb{R}^{n \times n}$ inversible es posible aplicar Eliminación Gaussiana sin pivoteo sobre la matriz obtenida para el sistema definido.

Para ello vamos a demostrar la siguiente propiedad

2.2.1 Dominancia Diagonal

Definición 1 (Matriz diagonal dominante). Sea $A \in \mathbb{R}^{n \times n}$ A se dice diagonal dominante si para toda fila i

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|.$$

Proposición 2. A^* es diagonal dominante.

Demostración. A^* es diagonal dominante (no estrictamente).

Para toda fila i se tiene que $|a_{ii}^*| \geq \sum_{j \neq i} |a_{ij}^*|$ si solo si

$$\left| \frac{-2}{(\Delta r)^2} + \frac{1}{r_j \cdot \Delta r} + \frac{-2}{r_j^2 \cdot (\Delta \theta)^2} \right| \geq \left| \frac{1}{(\Delta r)^2} - \frac{1}{r_j \cdot \Delta r} \right| + \left| \frac{1}{r_j^2 \cdot (\Delta \theta)^2} \right| + \left| \frac{1}{r_j^2 \cdot (\Delta \theta)^2} \right| + \left| \frac{1}{(\Delta r)^2} \right|$$

Analizemos la parte izquierda de la igualdad primero

$$\left| \frac{-2}{(\Delta r)^2} + \frac{1}{r_j \cdot \Delta r} + \frac{-2}{r_j^2 \cdot (\Delta \theta)^2} \right|$$

Notar que $r_j = r_i + \Delta r \cdot j$ y que $j \in I_r$ donde el menor valor posible de $j = 1$ entonces $r_j \geq r_i + \Delta r$ y dado que $r_i > 0$ entonces $r_j > \Delta r$ para todo $j \in I_r$. Sigue que $r_j \Delta r > (\Delta r)^2$. Por lo tanto la suma de los dos primeros términos es negativa y el tercero también lo es ya que r_j y $\Delta \theta$ son positivos, por lo tanto la suma de los tres es negativa.

$$\left| \frac{-2}{(\Delta r)^2} + \frac{1}{r_j \cdot \Delta r} + \frac{-2}{r_j^2 \cdot (\Delta \theta)^2} \right| = \frac{2}{(\Delta r)^2} - \frac{1}{r_j \cdot \Delta r} + \frac{2}{r_j^2 \cdot (\Delta \theta)^2}$$

Ahora, observemos la parte derecha de la igualdad

$$\left| \frac{1}{(\Delta r)^2} - \frac{1}{r_j \cdot \Delta r} \right| + \left| \frac{1}{r_j^2 \cdot (\Delta \theta)^2} \right| + \left| \frac{1}{r_j^2 \cdot (\Delta \theta)^2} \right| + \left| \frac{1}{(\Delta r)^2} \right|$$

Dado que r_j , Δr y $\Delta \theta$ son números positivos entonces los últimos tres términos son positivos por ser cociente de números positivos.

$$\left| \frac{1}{(\Delta r)^2} - \frac{1}{r_j \cdot \Delta r} \right| + \frac{1}{r_j^2 \cdot (\Delta \theta)^2} + \frac{1}{r_j^2 \cdot (\Delta \theta)^2} + \frac{1}{(\Delta r)^2}$$

De la misma forma que en la parte izquierda tenemos que $r_j \Delta r > (\Delta r)^2$ por lo tanto en este caso la suma del primer término es positiva.

$$\frac{1}{(\Delta r)^2} - \frac{1}{r_j \cdot \Delta r} + \frac{1}{r_j^2 \cdot (\Delta \theta)^2} + \frac{1}{r_j^2 \cdot (\Delta \theta)^2} + \frac{1}{(\Delta r)^2}$$

Luego, tenemos la siguiente inecuación

$$\frac{2}{(\Delta r)^2} - \frac{1}{r_j \cdot \Delta r} + \frac{2}{r_j^2 \cdot (\Delta \theta)^2} \geq \frac{1}{(\Delta r)^2} - \frac{1}{r_j \cdot \Delta r} + \frac{1}{r_j^2 \cdot (\Delta \theta)^2} + \frac{1}{r_j^2 \cdot (\Delta \theta)^2} + \frac{1}{(\Delta r)^2}$$

Simplificando tenemos que

$$\frac{2}{(\Delta r)^2} - \frac{1}{r_j \cdot \Delta r} + \frac{2}{r_j^2 \cdot (\Delta \theta)^2} \geq \frac{2}{(\Delta r)^2} - \frac{1}{r_j \cdot \Delta r} + \frac{2}{r_j^2 \cdot (\Delta \theta)^2}$$

□

2.2.2 Eliminación Gaussiana

Demostración. De la **Proposición 1.** es posible aplicar Eliminación Gaussiana sin pivoteo.²

Sea $A \in \mathbb{R}^{n \times n}$ inversible y diagonal dominante, demostremos por inducción que en todo paso de eliminación Gaussiana $A^{(k)}$ se mantienen las propiedades de dominancia diagonal y que $a_{kk} \neq 0$.

Sea P_k para todo $k \in \mathbb{N}$, $k < n$ tal que

$$P_k := \text{La matriz } A^{(k)} \text{ es diagonal dominante, inversible y } a_{k+1,k+1} \neq 0.$$

CASO BASE P_0 :

Dado que A es inversible sabemos que su rango fila es n , por lo tanto la primer fila no es nula y existe a_{1j} , $1 \leq j \leq n$ distinto de cero, luego por la Propiedad (2) sabemos que A es diagonal dominante y entonces a_{11} será mayor o igual en módulo a este valor.

PASO INDUCTIVO $P_{k-1} \Rightarrow P_k$:

$$P_k := \text{La matriz } A^{(k-1)} \text{ es diagonal dominante, inversible y } a_{kk} \neq 0.$$

Debemos demostrar

1. La dominancia diagonal de $A^{(k)}$.
2. La inversibilidad de $A^{(k)}$.
3. $(A^{(k)})_{kk} \neq 0$.

1. Para el paso k -ésimo de Eliminación Gaussiana se tiene que

$$a_{ij}^{(k)} = \begin{cases} a_{ij}^{(k-1)} & \text{si } i \leq k \\ a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} \cdot a_{kj} & \text{si } i > k \end{cases}$$

Queremos probar que

$$|a_{ii}^k| \geq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}^k|$$

para todo $1 \leq i \leq n$.

Dado que las primeras k filas de $A^{(k)}$ son iguales a las de $A^{(k-1)}$ y que $A^{(k-1)}$ es diagonal dominante por hipótesis inductiva, entonces solo resta demostrar que las filas $k+1$ hasta n son diagonal dominante.

$$\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}^{(k)}|$$

²En esta demostración por simplicidad la primer fila y columna de A será la número 1, siendo a_{11} el elemento en la esquina superior izquierda.

Sea $i \geq k$ y $j \leq k$ entonces $a_{ij} = 0$, por lo tanto

$$\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}^{(k)}| = \sum_{\substack{j=k+1 \\ j \neq i}}^n |a_{ij}^{(k)}|$$

Ya que $i \geq k$ por la definición de $A^{(k)}$ se tiene que

$$= \sum_{\substack{j=k+1 \\ j \neq i}}^n \left| a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} \cdot a_{kj}^{(k-1)} \right|$$

Por desigualdad triangular

$$\leq \left[\sum_{\substack{j=k+1 \\ j \neq i}}^n |a_{ij}^{(k-1)}| + \left| \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} \cdot a_{kj}^{(k-1)} \right| \right]$$

Separamos la sumatoria

$$= \left[\sum_{\substack{j=k+1 \\ j \neq i}}^n |a_{ij}^{(k-1)}| \right] + \left[\sum_{\substack{j=k+1 \\ j \neq i}}^n \left| \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} \cdot a_{kj}^{(k-1)} \right| \right]$$

Sacamos factor común

$$= \left[\sum_{\substack{j=k+1 \\ j \neq i}}^n |a_{ij}^{(k-1)}| \right] + \frac{|a_{ik}^{(k-1)}|}{|a_{kk}^{(k-1)}|} \left[\sum_{\substack{j=k+1 \\ j \neq i}}^n |a_{kj}^{(k-1)}| \right]$$

Sumamos y restamos los terminos $|a_{ik}^{(k-1)}|$ a la primer sumatoria y $|a_{ki}^{(k-1)}|$ a la segunda.

$$= \left[\sum_{\substack{j=k \\ j \neq i}}^n |a_{ij}^{(k-1)}| \right] - |a_{ik}^{(k-1)}| + \frac{|a_{ik}^{(k-1)}|}{|a_{kk}^{(k-1)}|} \left(\left[\sum_{j=k+1}^n |a_{kj}^{(k-1)}| \right] - |a_{ki}^{(k-1)}| \right)$$

Dado que $A^{(k-1)}$ es diagonal dominante acotamos las sumatorias por sus elementos diagonales.

$$\leq |a_{ii}^{(k-1)}| - |a_{ik}^{(k-1)}| + \frac{|a_{ik}^{(k-1)}|}{|a_{kk}^{(k-1)}|} \left(|a_{kk}^{(k-1)}| - |a_{ki}^{(k-1)}| \right)$$

Simplificamos.

$$\begin{aligned} &= |a_{ii}^{(k-1)}| - |a_{ik}^{(k-1)}| + \frac{|a_{ik}^{(k-1)}|}{|a_{kk}^{(k-1)}|} |a_{kk}^{(k-1)}| - \frac{|a_{ik}^{(k-1)}|}{|a_{kk}^{(k-1)}|} |a_{ki}^{(k-1)}| \\ &= |a_{ii}^{(k-1)}| - \frac{|a_{ik}^{(k-1)}|}{|a_{kk}^{(k-1)}|} |a_{ki}^{(k-1)}| \end{aligned}$$

Ya que $|a| - |b| \leq |a - b|$

$$\leq \left| a_{ii}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{ki}^{(k-1)} \right|$$

Que es

$$= |a_{ii}^{(k)}|$$

2. Dado que $A^{(k-1)}$ es inversible tenemos que $\det(A^{(k-1)}) \neq 0$, cada paso del algoritmo de Eliminación Gaussiana se puede ver como la multiplicación de $A^{(k-1)}$ por $M^{(k-1)}$, la k -ésima matriz de transformación de Gauss[1, p. 402]. Las matrices de transformación de Gauss son matrices triangular inferiores con unos en la diagonal por lo tanto su determinante es 1. Luego

$$\det(M^{(k-1)}A^{(k-1)}) = \det(M^{(k-1)}) \cdot \det(A^{(k-1)})$$

Ya que $\det(M^{(k-1)}) = 1$.

$$= \det(A^{(k-1)})$$

Por otro lado ya que $M^{(k-1)}A^{(k-1)} = A^{(k)}$

$$\det(M^{(k-1)}A^{(k-1)}) = \det(A^{(k)})$$

entonces

$$\det(A^{(k-1)}) = \det(A^{(k)})$$

Ya que $A^{(k-1)}$ era inversible entonces $\det(A^{(k-1)}) \neq 0$ por lo tanto $\det(A^{(k)}) \neq 0$ y eso implica que $A^{(k)}$ es inversible

3. Dado que $A^{(k)}$ es inversible por (2.) entonces las filas son L.I. y sigue que ninguna es nula. Luego

$$|a_{k+1,k+1}^{(k)}| \geq \sum_{j \neq k+1} |a_{k+1,j}^{(k)}|$$

ya que A es diagonal dominante por (1.).

Como resultado si $|a_{k+1,k+1}^{(k)}|$ fuera igual a 0 entonces

$$0 \geq \sum_{j \neq k+1} |a_{k+1,j}^{(k)}|$$

Lo cual implica que

$$\sum_{j \neq k+1} |a_{k+1,j}^{(k)}| = 0$$

Lo cual implica que la fila es nula. Pero esto es absurdo ya que las filas son L.I..

Por lo tanto, se preserva la dominancia diagonal y nunca se anula la diagonal, como conclusión el algoritmo de Eliminación Gaussiana puede finalizar, ya que todas las operaciones entre las filas estarán bien definidas. \square

3 Métodos numéricos utilizados

3.1 Cálculo de isoterma.

Por simplicidad vamos a asumir que una isoterma es una función $I_{r_i, R_e} : [0, 2\pi) \rightarrow [r_i, R_e]$ tal que para cada ángulo retorna el radio de la isoterma en ese ángulo.

3.1.1 Vecino más Cercano

Consiste en seleccionar para cada ángulo el radio cuya temperatura se aproxime más, en valor absoluto al valor de la isoterma.

3.1.2 Interpolación Lineal

Como un refinamiento de este algoritmo se propone utilizar interpolación lineal en los radios. Se busca el punto de la discretización cuya temperatura esté más cerca de la temperatura de la isoterma, y se interpola linealmente con alguno de sus vecinos en el mismo ángulo.

Sea ι la temperatura de la isoterma buscada, θ_k un ángulo fijo y r_j, r_{j+1} dos radios adyacentes en la discretización y r el radio tal que $T(r, \theta_k) = \iota$ y $r_1 < r < r_2$

$$\iota = t_{jk} + (r - r_j) \cdot \frac{t_{j+1,k} - t_{jk}}{r_{j+1} - r_j}$$

Dejando r .

$$r = (r_{j+1} - r_j) \frac{\iota - t_{jk}}{t_{j+1,k} - t_{jk}} + r_j$$

3.1.3 Interpolación Bilineal

Una segunda sofisticación consiste en interpolar tanto por ángulo como por radio, utilizando interpolación bilineal. Ya que no es trivial encontrar la inversa de la interpolación bilineal se construye una nueva discretización con mayor resolución a partir de la discretización original, y luego se rellenan los puntos no conocidos de esta nueva discretización interpolando los ya conocidos.

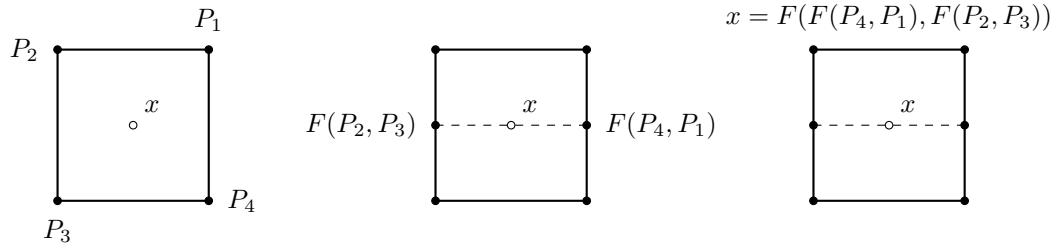


FIGURA 4: Funcionamiento de la interpolación bilineal, primero se interpola en un eje (en este caso el vertical) para obtener dos puntos, que interpolados linealmente dan el valor de la incógnita.

El método de interpolación bilineal consiste en tomar un cuadrilátero de puntos conocidos, e interpolar linealmente en alguna de sus dimensiones para luego hacerlo en la otra dimensión con los valores previamente interpolados como muestra la Figura (4).

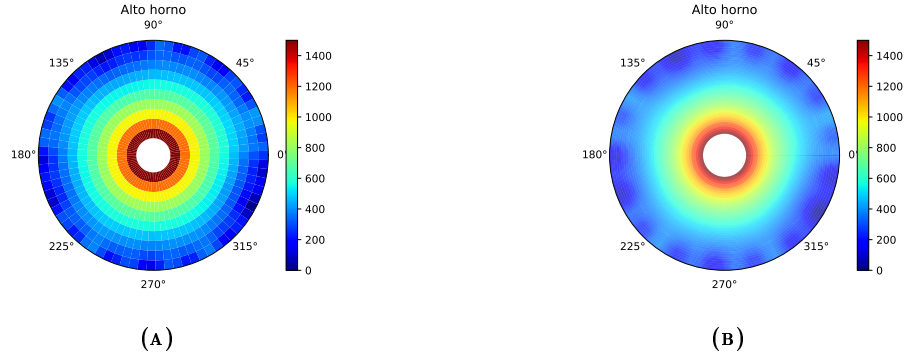


FIGURA 5: *Discretización original en (A) contra su interpolación bilineal en (B).*

El resultado como exhibe la Figura (5) preserva los puntos originales y “suaviza” los puntos intermedios; luego se procede a utilizar alguno de los algoritmos anteriores de cálculo de isoterma sobre esta nueva discretización.

3.2 Cálculo de riesgo

Luego de obtener la línea isotérmica es necesario dar un estimado de cuanto riesgo esta presenta al horno.

Definiremos funciones de riesgo $\phi: \mathbb{R}^{\mathbb{R}} \rightarrow [0, 1]$ que dada una isoterma devuelva su valor de riesgo, donde un valor próximo a 1 implica un riesgo elevado y un valor cercano a 0 implica un riesgo bajo.

Introducimos las siguientes dos nociones de riesgo.

3.2.1 Riesgo Promedio

Un algoritmo que dado $v \in \mathbb{R}^n$ una discretización de la isoterma retorna

$$\frac{\bar{v} - r_i}{R_e - r_i}.$$

El algoritmo RIESGO PROMEDIO calcula el radio promedio de la isoterma, cuando este se aproxima a R_e entonces la isoterma está en promedio cerca de la pared externa y el riesgo es alto. Por otro lado si el radio promedio es cercano a r_i el riesgo es bajo. Se puede pensar que lo que ocurre es que estamos aproximando la isoterma por el círculo que mejor se ajusta. Cuanto menos circular sea la isoterma, peor será la aproximación.

3.2.2 Riesgo Distancia Mínima

Un algoritmo que dado $v \in \mathbb{R}^n$ una discretización de la isoterma retorna

$$\frac{\|v\|_{\infty} - r_i}{R_e - r_i}.$$

El algoritmo RIESGO DISTANCIA mide el riesgo en función de la distancia entre la pared externa y el punto más cercano de la isoterma. A diferencia del riesgo promedio está función de riesgo es especialmente susceptible a “picos” en la isoterma. Este riesgo acota la isoterma por arriba con el círculo más pequeño posible.

3.3 Análisis de ambas mediciones

Medir la distancia entre la pared y el punto más cercano de la isoterma c puede resultar volátil. Un cambio rotundo de temperatura en un solo punto de la pared eleva el riesgo enormemente. Además, este método retorna las mismas mediciones que en el caso en el que todos los puntos presentan está temperatura máxima. No solo RIESGO DISTANCIA puede devolver mediciones iguales para casos que presentan riesgos reales muy distintos, sino que además puede variar mucho en función del tiempo.

Sin embargo medir el riesgo en función de la temperatura promedio no presenta este problema: Es más difícil cambiar la medición drásticamente en poco tiempo. Para que el riesgo sea alto una gran parte de la pared externa debe presentar temperaturas elevadas. Sin embargo esta medición, en contraparte, no diferencia entre picos de temperatura y casos donde la temperatura es uniforme.

De todas formas, sigue siendo preferible poder determinar picos de temperatura en regiones específicas para poder distinguir imperfecciones en la pared. Lo ideal sería poder hacer mediciones utilizando ambos algoritmos a la vez, o medir el riesgo en función a la mínima distancia en distintas regiones del horno (contar picos).

4 Experimentación

Para tener un mayor entendimiento del modelo, se llevó a cabo una serie de experimentos. Luego se ha buscado comparar la eficiencia de la resolución del sistema con respecto a la Eliminación Gaussiana y la factorización LU . Además buscamos comparar los algoritmos de cálculo de riesgo en función a los casos borde y experimentar con los algoritmos de interpolación para el cálculo de la isoterma.

4.1 Hipótesis generales

- i. Utilizar el método de factorización LU tiene una menor constante multiplicativa que EG por lo tanto será más eficiente; esto es especialmente verdad cuando el número de instancias sea mayor.
- ii. El método de riesgo de distancia mínima (3.2.2) es un mejor predictor del riesgo que el de distancia promedio (3.2.1).
- iii. La interpolación linear y bilinear son suficientemente exactas y útiles como alternativas baratas a resolver el sistema con una discretización con mayor granularidad.

4.2 Instancias

4.2.1 Caso Uniforme (k)

Esta instancia plantea un caso estándar donde todas las temperaturas externas son las mismas.

$$r_i = 1, Re = 2, n = 30, m + 1 = 30, Ti = 1500, iso = 500$$

$$Te(\theta) = k$$

4.2.2 Caso Pico (p, k)

Esta instancia busca exponer un caso en el que la isoterma presenta un pico.

$$r_i = 1, Re = 2, n = 30, m + 1 = 30, Ti = 1500, iso = 500$$

$$Te(\theta) = \left(\frac{\sin(\theta) + 1}{2} \right)^p \cdot k$$

4.2.3 Caso Distribución Normal (μ, σ^2)

Esta instancia modela una situación tipo, con fluctuaciones y una temperatura media estable.

$$r_i = 10, Re = 100, n = 30, m + 1 = 30, Ti = 1500, iso = 500$$

$$Te(\theta) \sim \mathcal{N}(\mu, \sigma^2)$$

4.2.4 Caso Oscilante

Esta instancia busca una matriz de temperaturas irregular, que produzca una isoterma apta para experimentación. Se tiene en cuenta el experimento para poder demostrar la eficiencia de los algoritmos de cálculo de isoterma con interpolación.

$$r_i = 1, Re = 2, n = 30, m + 1 = 30, Ti = 1500, iso = 500$$

$$Te = \sum_{i=1}^k (\sin(\theta + \mu_i) + 1) \cdot \sigma_i$$

La temperatura externa es una suma de funciones sinusoidales con distintas fases y amplitudes.

4.3 Experimentos

Todos los experimentos fueron ejecutados sobre un procesador Intel(R) Core(TM) i7-3770K CPU @ 3.50GHz, memoria 16GB RAM @ 1600mhz, y un disco Samsung 960 evo SSD 1TB en un sistema operativo Manjaro Linux.

4.3.1 EG vs. LU, número de subinstancias creciente

Para este experimento, se ejecutaron instancias de Caso Normal ($\mu = 250$, $\sigma^2 = 125$). El objetivo del experimento es exponer la velocidad de cómputo de los distintos métodos, en relación a la cantidad de subinstancias. Por lo tanto, todos los parámetros menos la cantidad de *ninst* se han mantenido constantes.

Para cada valor de *ninst* se han llevado a cabo ≈ 50 mediciones (para instancias con *ninst* más grande dado que el tiempo de computo es mayor son necesarias menos muestras para reducir la varianza a un nivel aceptable) de cincuenta mediciones, y se ha graficado el promedio para cada método.

Hipótesis

Al aumentar la cantidad de subinstancias, lo esperado es ver que el tiempo de cómputo requerido para la resolución de ambos métodos crezca. Sin embargo, la Factorización *LU* debería tardar notablemente menos a medida que *ninst* toma valores mayores ya que la resolución de las subinstancias subsiguientes es de costo $\mathcal{O}(n^2)$ contra el costo de Eliminación Gaussiana que es de $\mathcal{O}(n^3)$ para todas las subinstancias.

Resultados Se puede observar en la Figura (6) como el costo de cómputo de ambos métodos crece en relación al numero de subinstancias. Donde el costo de ejecutar más subinstancias con Eliminación Gaussiana crece exponencialmente más rápido que la misma cantidad utilizando Factorización *LU*.

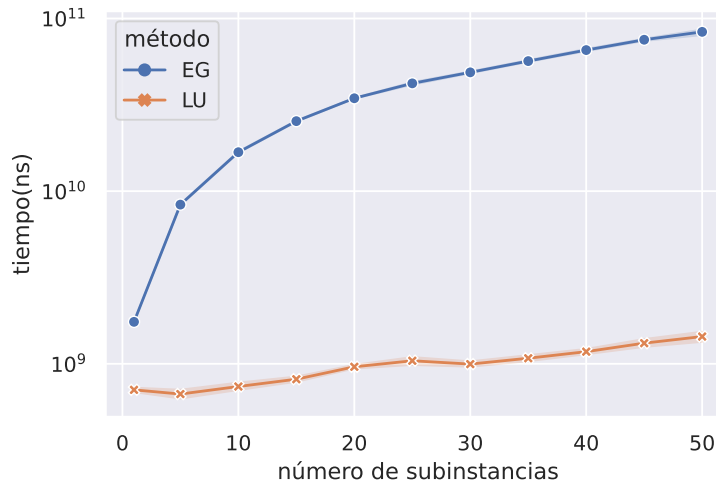


FIGURA 6: Velocidad de computo en función a la cantidad de subinstancias (escala logarítmica) con intervalo de confianza $p = 0.95$.

4.3.2 EG vs. LU, $n \cdot m$ creciente.

Este experimento busca exponer la eficiencia de ambos métodos, con respecto a una sola subinstancia en función al tamaño de la matriz. Se han ejecutado instancias de Caso Distribución Normal(250, 125) donde los parámetros n y m aumentan constantemente.

Hipótesis

Al tener que resolver una sola subinstancia del sistema, lo esperado es ver que ambos métodos tardan prácticamente lo mismo. Sin embargo, al ser la constante multiplicativa de LU $\frac{2}{3}$ comparada con la de Eliminación Gaussiana $\frac{3}{3}$, donde n es la cantidad de filas de la matriz, lo esperable es que al aplicar ambos algoritmos a matrices cada vez más grandes se acentúe la diferencia. Aun así para valores de $n \cdot (m + 1)$ bajos la diferencia debería ser despreciable.

Resultados

Los tiempos de cómputo de ambos algoritmos son similares (esencialmente realizan las mismas operaciones), sin una vez realizada la factorización LU subsiguientes instancias de este problema tienen una complejidad perteneciente a $\mathcal{O}(n^2)$ [1, p. 401] por lo tanto cuando crece el tamaño de la instancia su tiempo de cómputo es mucho menor. Se puede observar que el orden de crecimiento es el mismo, y que hay una constante multiplicativa que hace una leve diferencia. Esta diferencia puede apreciarse en la Figura (7), donde se exponen los tiempos de ejecución en función al tamaño de la muestra en escala logarítmica.

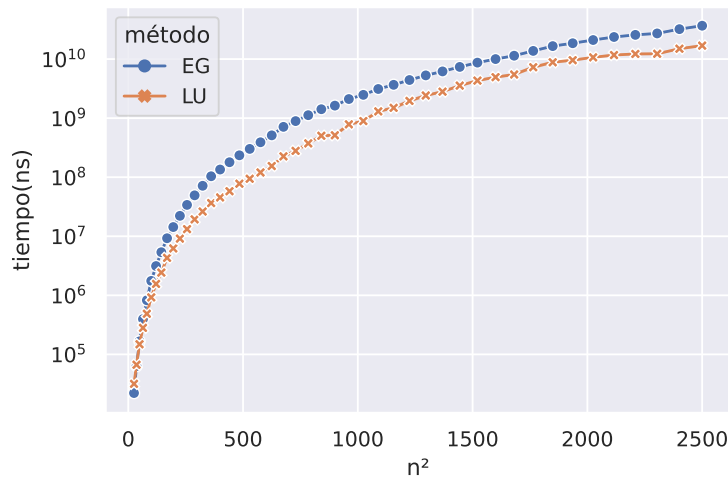


FIGURA 7: Velocidad de cómputo en función a la granularidad de la matriz en escala logarítmica.

4.3.3 Riesgo Caso Pico

El objetivo es poder apreciar las mediciones de riesgo con respecto a la intensidad del pico. Para esto se han ejecutado muchas instancias del caso Pico(p, k), donde se grafican las mediciones de

riesgo de ambos algoritmos (RIESGO PROMEDIO y RIESGO DISTANCIA) en relación al aumento de p , con una temperatura k . Notar que para valores mas grandes, se genera un pico en la isoterma cada vez delgado.

Hipótesis

Lo esperable es poder ver que para valores bajos de k , la isoterma presenta un pico mas pronunciado. Por lo tanto, en promedio las temperaturas serán mayores. Los puntos adyacentes al valor máximo de la isoterma tendrán temperaturas no demasiado menores. Sin embargo, cuando crece p , el pico se vuelve mas pronunciado para un solo punto. Por lo tanto es esperado apreciar que en promedio la temperatura es menor. Es por esto que esperamos ver que las mediciones de RIESGO DISTANCIA se mantengan constantes a medida que crece p (ya que la distancia entre la pared externa y el punto máximo es la misma), mientras que el RIESGO PROMEDIO baja y converge.

Resultados

La Figura (8) remarca el cambio que tiene el aumento de p sobre la isoterma, para una temperatura máxima k . Se puede observar en la misma como la temperatura total (puede pensarse como el área bajo la isoterma) es menor a medida que aumenta p , por más que k se mantenga constante y se forme un pico mas pronunciado. En la Figura (9) se puede observar como los valores de RIESGO DISTANCIA se mantienen constantes, ya que el punto con el máximo de la isoterma se mantiene siempre a misma distancia (siempre la isoterma toca la pared externa), mientras que los valores de RIESGO PROMEDIO bajan a medida que crece p . Esto se debe a que el valor de RIESGO PROMEDIO es proporcional al área debajo de la isoterma, mientras que RIESGO DISTANCIA devuelve el riesgo en función a la temperatura máxima.

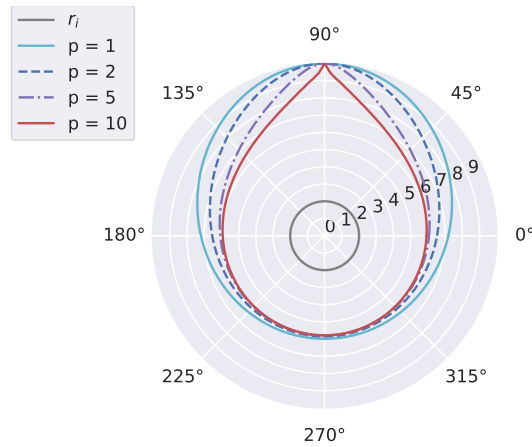


FIGURA 8: Gráfico de las isotermas para distintas instancias del caso pico.

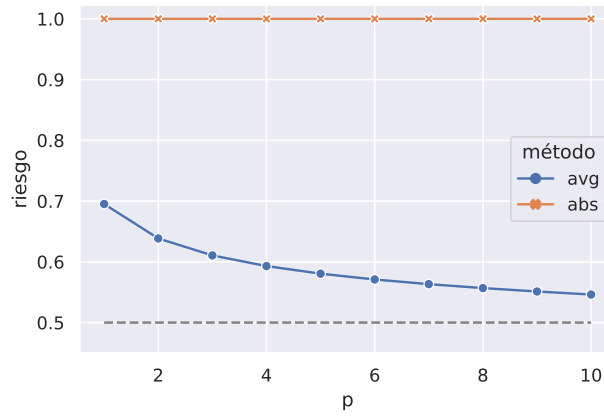


FIGURA 9: Comparación de mediciones de riesgo en función al parámetro p .

4.3.4 Interpolaciones isoterma

Este experimento se lleva a cabo para comparar distintos métodos para el cálculo de la isoterma. Se ejecuto una única instancia del Caso Ondulado con n y m bajos para luego graficar la isoterma interpolada linealmente y bilinealmente, adicionalmente, para n y m crecientes y para cada método se gráfico la norma dos, de la isoterma real menos la aproximada.

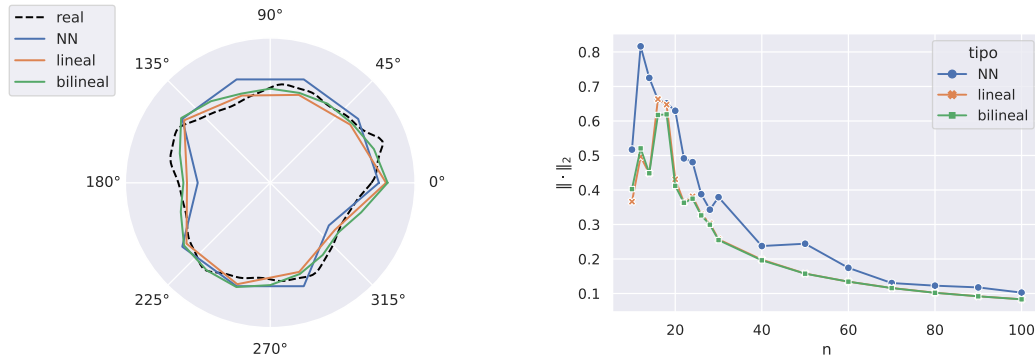
Hipotesis

Lo esperado es que el mejor predictor de la isoterma real sea la interpolación bilinear seguida de la interpolación lineal. Esto se debe a que la primera hace la predicción usando interpolación lineal en función a cuatro puntos, en contraste a la segunda que solo utiliza 2, mientras que vecino más cercano es el más burdo de todos.

Resultados

La Figura (10a) muestra un ejemplo de las isothermas aproximadas de los tres métodos a partir de una granularidad baja. Vecino más Cercano (3.1.1), Interpolación Lineal (3.1.2) e Interpolación Bilineal (3.1.3). Además la figura (10b) contrasta la norma 2 de la diferencia entre la isoterma “real” (una aproximación con mucha granularidad) y la isoterma aproximada por el método, a medida que crece n con $m = \frac{n}{2}$.

El resultado no fue exactamente el esperado, esperábamos una mayor mejora de la interpolación bilinear respecto a la interpolación lineal, pero rápidamente ambas se estabilizan y se comportan de igual manera. Si fue esperado que la isoterma absoluta presente grandes diferencias con la isoterma real. Con n suficientemente grande todas convergen.



(A) Proyecciones de los métodos de aproximación de isothermas.

(B) Comparación en función a la norma 2.

FIGURA 10: En (A) se pueden ver los distintos métodos utilizando una discretización de baja granularidad intentando aproxima la isoterma real (línea punteada) en (B) tenemos la norma 2 de la diferencia entre la isoterma real y la aproximación para los distintos métodos.

5 Conclusiones

El trabajo presente abordó el cálculo de temperaturas del modelo discretizado de un alto horno. Para la resolución del mismo se comparó la eficiencia de dos métodos: la Eliminación Gaussiana y la Factorización LU . Mediante experimentación se pudo concluir que efectivamente la Factorización LU requiere menos tiempo de cómputo al resolver un sistema. Cuando la cantidad de subinstancias era baja se pudo contemplar que la diferencia en tiempo de cómputo no era significativa, mientras que al aumentar la cantidad de subinstancias la diferencia se vuelve más notable. Además, al aumentar el tamaño de las matrices se pudo ver una leve mejora en los tiempos de ejecución, exponiendo la diferencia de las constantes multiplicativas de ambos métodos.

Adicionalmente, se pudo apreciar la diferencia entre el algoritmo de Vecino más Cercano, Interpolación Lineal e Interpolación Bilineal, demostrando la mejoría de las interpolaciones lineal y bilineal ante vecino más cercano.

Se pudo ver como en casos de baja granularidad la interpolación lineal y bilineal resulta efectiva, mientras que la interpolación de vecino más cercano requiere mayor granularidad para converger.

También se pudo comparar el cálculo del riesgo que presenta la proximidad de la isoterma con respecto a la pared externa en función a dos algoritmos distintos.

Referencias

- [1] Burden R.L. and Faires J.D. *Numerical Analysis*. 9th ed. Brooks/Cole, 2010.