



DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico I

Discretización de la temperatura de un Alto Horno

Métodos Numéricos
Segundo Cuatrimestre de 2015

Integrante	LU	Correo electrónico
Alvarez, Lautaro Leonel	268/14	lautarolalvarez@gmail.com
Maddonni, Axel Ezequiel	200/14	axel.maddonni@gmail.com
Thibeault, Gabriel Eric	114/13	gabriel.eric.thibeault@gmail.com

En este trabajo discretizaremos el interior de un alto horno para calcular su temperatura interna e implementaremos un algoritmo para determinar si se encuentra en estado de peligro, usando distintos métodos numéricos para resolver sistemas de ecuaciones lineales planteados a partir de ecuaciones matemáticas que describen dicha situación.

eliminación gaussiana, factorización lu



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice

1. Introducción Teórica	3
1.1. Explicación de los métodos utilizados	3
1.1.1. Eliminación Gaussiana	3
1.1.2. Descomposición LU	3
1.1.3. Resolver un sistema triangular	3
1.1.4. Resolver un sistema LU	3
2. Desarrollo	4
2.1. Presentación del problema y explicación del método de discretización	4
2.2. Armado de la Matriz Banda	4
2.3. Demostración: Eliminación Gaussiana sin pivoteo	5
2.3.1. Lema	5
2.3.2. Demostración de la proposición	5
2.4. Planteo de Algoritmos	7
2.5. Planteo de Mediciones Experimentales	8
3. Resultados y Discusión	9
3.1. Resultados con respecto al Tiempo de ejecución	9
3.1.1. Experimento 1	9
3.1.2. Resultados del Experimento 1	9
3.1.3. Experimento 2	9
3.1.4. Resultados del Experimento 2	10
3.1.5. Experimento 3	11
3.1.6. Resultados del Experimento 3	11
3.2. Resultados con respecto a la Isoterma	12
3.2.1. Experimento 4	12
3.2.2. Resultados del experimento 4	12
3.2.3. Experimento 5	13
3.2.4. Resultados del experimento 5	13
3.2.5. Experimento 6	14
3.2.6. Resultados del experimento 6	14
3.2.7. Experimento 7	14
3.2.8. Primera entrada para el experimento 7	15
3.2.9. Resultados para la primera entrada para el experimento 7	15
3.2.10. Segunda entrada para el Experimento 7	15
3.2.11. Resultados para la segunda entrada para el Experimento 7	15
3.2.12. Experimento 8	16
3.2.13. Resultados del Experimento 8	16
3.3. Resultados con respecto a la Peligrosidad	17
3.3.1. Experimento 9	17
3.3.2. Resultados del experimento 9	17
4. Conclusiones	18
5. Apéndices	19
5.1. Apéndice A	19
5.2. Apéndice B	20
6. Referencias	21

1. Introducción Teórica

1.1. Explicación de los métodos utilizados

1.1.1. Eliminación Gaussiana

El método de Eliminación Gaussiana se utiliza para triangular superiormente una matriz. Esto se logra iterando sobre filas, utilizando un pivote diagonal (que llamaremos i en la siguiente explicación por claridad); en cada iteración del ciclo exterior se resta a cada una de las filas inferiores a la i -ésima, la i -ésima multiplicada por un coeficiente tal que el elemento de la i -ésima columna quede en 0. Al iterar sobre todas las filas (excepto la última, ya que no tiene filas inferiores a ella), se triangula la matriz. Al restarle una fila a otra, también se restan los términos independientes correspondientes a cada fila de la misma forma.

Una optimización que incorporamos es no restar las filas si el elemento a dejar en 0 ya está en 0. Como la matriz que representa el problema es Banda (esta proposición se encuentra demostrada en la sección 2.2), y por ende debería tener una significativa cantidad de valores en 0, esta optimización debería ser considerablemente efectiva.

1.1.2. Descomposición LU

La Descomposición LU es similar a la Eliminación Gaussiana, pero al restarle la fila i -ésima multiplicada por el coeficiente (i siendo el pivote diagonal detallado en la sección anterior) a la j -ésima (j siendo el índice de la fila sobre la cual se itera en el ciclo interno), se guarda tal coeficiente como el j, i -ésimo valor de una nueva matriz. Los valores de la diagonal de la nueva matriz se inicializan en 1, y el resto en 0, y por la forma en que se itera para triangular superiormente una matriz, ésta queda triangular inferior.

La diferencia principal de la Descomposición LU respecto de la Eliminación Gaussiana es que no involucra al término independiente, y una vez descompuesta una matriz, se puede resolver el sistema para cualquier término independiente. Ya que la complejidad de resolver un sistema triangular (o dos, en el caso de LU) es $O(n^2)$ (n siendo el tamaño de la matriz), y el de la Descomposición LU o Gauss es $O(n^3)$, si se debe resolver el mismo sistema con distintos términos independientes, LU debería ser el método superior, ya que sólo se debe pagar una vez el coste cúbico.

Utilizamos la misma optimización para matriz banda con LU que mencionamos en la sección anterior.

1.1.3. Resolver un sistema triangular

Para resolver un sistema triangular se itera por filas (comenzando a partir de la fila con un único valor no-nulo), y se despeja el valor de la diagonal, restándole al término independiente todos los otros coeficientes por el valor de sus variables (que, si se itera correctamente, ya deberían haber sido calculadas). Resolver un sistema triangular inferior es similar a resolver un sistema triangular superior, pero se comienza a iterar en distintos puntos y se deben despejar los valores de distintos lados de la diagonal (ya que los sistemas son triangulares no hace falta despejar todos los valores de la fila, ya que se sabe de antemano que los de un lado son todos 0).

1.1.4. Resolver un sistema LU

Resolver un sistema que fue descompuesto mediante LU consiste en encontrar la solución de la matriz triangular inferior con el término independiente original, y luego la matriz triangular superior utilizando como término independiente a la solución recién encontrada para la matriz inferior.

2. Desarrollo

2.1. Presentación del problema y explicación del método de discretización

Para averiguar la posición de la isoterma debemos determinar la temperatura en cada punto de la pared. Al sernos imposible resolver el problema computacionalmente decidimos discretizar el problema, por lo que vamos a buscar las temperaturas cada una cierta distancia (la cual variaremos para experimentar los cambios en los resultados). Identificaremos así un punto de la pared por su distancia al centro del horno (r) y su ángulo (θ) con respecto a un eje fijo.

Para determinar la temperatura en un punto contamos con la siguiente ecuación:

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

Siendo $T(r, \theta)$ la función que nos da como resultado la temperatura en un punto de radio r y ángulo θ .

Discretizamos tomando $r_i = r_0 < r_1 < \dots < r_n = r_e$ (con r_i =radio interno y r_e =radio externo) y $0 = \theta_0 < \theta_1 < \dots < \theta_m = 2\pi$, tomando así $n+1$ radios y m ángulos. Por ende tomamos un t que cumple: $t_{i,j} = T(r_i, \theta_j)$ (temperatura en el punto con distancia r_i al centro y ángulo θ_j).

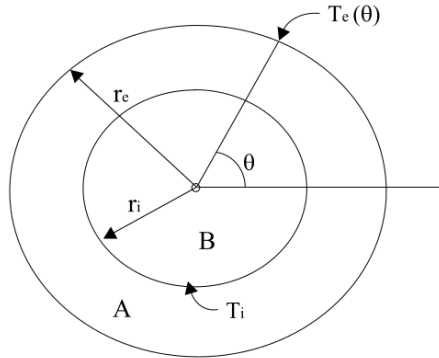
Por ende tomamos la ecuación 1 y discretizando tenemos:

$$\frac{t_{i-1,j} - 2t_{i,j} + t_{i+1,j}}{(\Delta r)^2} + \frac{1}{r} \frac{t_{i+1,j} - t_{i,j}}{\Delta r} + \frac{1}{r^2} \frac{t_{i,j-1} - 2t_{i,j} + t_{i,j+1}}{(\Delta \theta)^2} \quad (2)$$

Con $\Delta r = r_i - r_{i-1}$ para $i=1..n$ y $\Delta \theta = \theta_j - \theta_{j-1}$ para $j=1..m$.

Como vemos en la ecuación 2 la temperatura en un punto i,j ($t_{i,j}$) depende de las temperatura de los 4 puntos mas cercanos ($t_{i-1,j}$, $t_{i+1,j}$, $t_{i,j-1}$, $t_{i,j+1}$).

Si planteamos un sistema de ecuaciones con 1 ecuación para cada i,j de la discretización obtenemos un sistema de $n*m$ ecuaciones donde cada una depende de 5 temperaturas. Al mismo tiempo sabemos de antemano los valores de los puntos externos e internos, teniendo en cuenta que tomamos como parámetros conocidos la temperatura en el borde externo de la pared (T_e) y la temperatura en el borde interno de la pared (T_i).



De esta forma, podemos plantear una matriz que cumple con la propiedad de ser banda, y con la cual además, es posible realizar Eliminación Gaussiana sin pivoteo (esta propiedad está demostrada más adelante).

2.2. Armado de la Matriz Banda

Para generar la matriz es necesario realizar una transformación del espacio polar en el que existen los puntos del horno (cuyas posiciones están dadas por ángulo y radio) a un vector de dimensión $(m + 1) * n$ (este valor corresponde con la cantidad de puntos en la discretización).

La transformación utilizada es bastante simple:

$$f(j, k) = j * n + k \quad (3)$$

donde f es la transformación, j representa que el elemento es el j -ésimo en la componente del radio, k el equivalente para el ángulo, y n es la cantidad de puntos de la discretización con igual radio (la cantidad de puntos en cada círculo concéntrico del horno).

De acuerdo a la ecuación 2, para cada punto $p_{j,k}$ (cuya ecuación se encuentra representada en la fila $f(j,k)$) que no pertenece ni al radio externo ni al interno, los únicos coeficientes no-nulos en su fila corresponden a los puntos $p_{j-1,k}$, $p_{j+1,k}$, $p_{j,k}$, $p_{j,k-1}$, $p_{j,k+1}$; siendo los subíndices la posición de cada punto en la discretización. Utilizando la ecuación 3, estas posiciones corresponden a las siguientes columnas, respectivamente: $(j-1)*n+k$, $(j+1)*n+k$, $j*n+k$, $j*n+k-1$, $j*n+k+1$.

k , como se ha dicho, representa al k -ésimo elemento con radio j ; ya que hay n elementos con igual radio, $0 \leq k \leq n$. Por ende, para $j > 0$ ($j = 0$ corresponde al radio interno, por lo que este caso no nos concierne en este momento), $j*n \geq k$. Por lo tanto, la posición de los valores de la diagonal en cada fila es $j*n+k$, y los valores no-nulos más alejados corresponden a $(j-1)*n+k$ y $(j+1)*n+k$, que entonces equidistan n de la diagonal.

Es decir, se cumple la propiedad de banda para todas las filas de la matriz que no pertenecen al radio interno o externo, y el ancho de la banda es $n+1$. Sin embargo, queda justificar que la propiedad de banda se cumple para todas las demás filas. Esto es simple de probar, ya que los valores de temperatura de cada uno de los puntos de estos radios son conocidos, y su ecuación está simplemente dada por $t_{j,k} = b_{j,k}$; donde el primer término representa la temperatura del punto y el segundo el valor del término independiente determinado por la entrada. Por lo tanto, el único coeficiente no-nulo de las filas de los radios externo e interno son los de la diagonal, cuyo valor es 1. Ya que todos los otros valores son nulos, en particular lo son los que distan más que n de la diagonal, y por ende la matriz es banda. Adicionalmente, hay n puntos en el radio interno, por lo que la altura de la banda también es $n+1$.

El algoritmo que crea la matriz en cuestión se encuentra en el apéndice B con comentarios incluidos, junto con el algoritmo correspondiente al método de Eliminación Gaussiana con aprovechamiento de matriz banda.

2.3. Demostración: Eliminación Gaussiana sin pivoteo

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

2.3.1. Lema

Sea $A \in \mathbb{R}^{n \times n}$ la matriz obtenida para el sistema, banda con ancho y alto $n+1$ (como se demostró previamente). La última diagonal superior que pertenece a la banda, tiene (para los valores correspondientes a los radios no-externo y no-internos) coeficientes positivos. Adicionalmente, al aplicar Eliminación Gaussiana estos coeficientes nunca se modifican.

Demostración del Lema

El hecho que los coeficientes iniciales de la última diagonal (superior) de la banda sean positivos se desprende de las ecuaciones de cada punto, y ya se explicó previamente. El hecho de que éstos no se modifiquen al aplicar Gauss se debe a la estructura banda de la matriz:

$$A_{i,j} = 0 \text{ si } j < i - (n+1) \vee j > i + (n+1) \quad (4)$$

Entonces, si $j = i + (n+1)$ (como es el caso de la última diagonal superior de la banda), para una fila i , para todas las filas anteriores $k < i$, $j > k + (n+1)$, por lo que $A_{k,j} = 0$. Al ser 0, al aplicar Gauss no se alterará el valor del coeficiente.

2.3.2. Demostración de la proposición

Ya detallamos que la ecuación de cada punto indica que en cada fila, la diagonal más la suma de todos los otros coeficientes (no nulos, y luego de todos) es 0. Es decir:

$$A_{i,i} = -\left(\sum_{j \neq i} A_{i,j}\right) \quad (5)$$

Por ende,

$$|A_{i,i}| = |(\sum_{j \neq i} A_{i,j})| = (\sum_{j \neq i} |A_{i,j}|) \geq (\sum_{j \neq i} |A_{i,j}|) \quad (6)$$

El módulo de la suma es igual a la suma de los módulos ya que todos los coeficientes son no-negativos. De acuerdo a la ecuación anterior, la matriz es diagonal dominante (aunque no estrictamente).

Demostraremos por inducción que dadas estas propiedades, se puede realizar Eliminación Gaussiana sin pivoteo.

Caso Base

El caso base consiste en la matriz creada inicialmente, que como ya se demostró, es diagonal dominante, y los valores de su diagonal son no-nulos (para los radios interno y externo valen 1, y para los otros ya se demostró que son no-nulos).

Paso Inductivo

Si A tras realizar k iteraciones de Gauss sin pivotaje es diagonal dominante, con $A_{k,k}$ (es decir, el k -ésimo valor de la diagonal) no-nulo, entonces A tras realizar $k+1$ iteraciones de Gauss sin pivotaje es diagonal dominante, con $A_{k+1,k+1}$ no-nulo.

Demostraremos primero que A tras realizar $k+1$ iteraciones es diagonal dominante. Restamos cada fila inferior a la k por el coeficiente correcto:

$$A_j - (A_{j,k}/A_{k,k}) * A_k \quad (7)$$

Sabemos que es posible hacer esto ya que por Hipótesis Inductiva $A_{k,k} \neq 0$. Sea

$$A_{i,j}^{(2)} = A_{i,j} - (A_{i,k}/A_{k,k}) * A_{k,j} \quad (8)$$

Quiero ver que:

$$\sum_{j \neq i} |A_{i,j}^{(2)}| \leq |A_{i,i}^{(2)}| \quad (9)$$

Veamos que esto sucede:

$$\sum_{j \neq i} |A_{i,j} - (A_{i,k}/A_{k,k}) * A_{k,j}| \quad (10)$$

Por Desigualdad Triangular:

$$\dots \leq \sum_{j \neq i} |A_{i,j}| + \sum_{j \neq i} |(A_{i,k}/A_{k,k})| \quad (11)$$

Por Hipótesis Inductiva:

$$\dots \leq |A_{i,i}| - |A_{i,k}| + (|A_{i,k}|/|A_{k,k}|) * (\sum_{j \neq i} |A_{k,j}|) \quad (12)$$

$$\dots \leq |A_{i,i}| - |A_{i,k}| + (|A_{i,k}|/|A_{k,k}|) * (|A_{k,k}| - |A_{k,i}|) \quad (13)$$

$$\dots \leq |A_{i,i}| - |A_{i,k}| * |A_{k,i}|/|A_{k,k}| \quad (14)$$

$$\dots \leq |A_{i,i}| \quad (15)$$

Finalmente, llegamos a que cada fila donde se restó por Gauss sin pivotaje mantiene su propiedad de diagonalidad dominante (no estricta).

Probamos la primera de las dos propiedades que queríamos demostrar: aún queda justificar que $A_{k+1,k+1}$ es no-nulo luego de la $k+1$ -ésima iteración. +Afortunadamente, esta demostración es más simple. Utilizando la misma notación que en la demostración de la propiedad anterior:

$$A_{k+1,k+1}^{(2)} = A_{k+1,k+1} - (A_{k+1,k}/A_{k,k}) * A_{k,k+1} \quad (16)$$

En particular, queremos ver que $A_{k+1,k+1}^{(2)} \neq 0$:

$$A_{k+1,k+1}^{(2)} \neq 0 \iff A_{k+1,k+1} - (A_{k+1,k}/A_{k,k}) * A_{k,k+1} \neq 0 \quad (17)$$

De esto se desprende:

$$A_{k+1,k+1} \neq (A_{k+1,k}/A_{k,k}) * A_{k,k+1} \quad (18)$$

Aplicando módulo de ambos lados de la inecuación:

$$|A_{k+1,k+1}| \neq |(A_{k+1,k}/A_{k,k})| * |A_{k,k+1}| \quad (19)$$

Por Hipótesis Inductiva, sabemos que

$$|A_{k+1,k+1}| \geq |(A_{k+1,k}| \quad (20)$$

$$|A_{k,k})| \geq |A_{k,k+1}| \quad (21)$$

La única forma en que se podría anular $A_{k+1,k+1}$ es si ambos valores de la diagonal son exactamente iguales a $A_{k,k+1}$ y $A_{k+1,k}$, respectivamente.

Sin embargo, esto no puede suceder, debido al Lema que demostramos al comienzo de esta prueba: los valores de la última diagonal de la banda (que debido a la forma de la matriz, nunca serán $A_{k,k+1}$ o $A_{k+1,k}$) son no-nulos (y en particular positivos, pero con no-nulos alcanza). Si pasara que $|A_{k,k}| = |A_{k,k+1}|$ (o lo mismo para el otro par de valores), entonces:

$$|A_{k,k}| < |A_{k,k+1}| + |A_{k,k+(n+1)}| \quad \text{Addacommenttothisline} \quad (22)$$

Esto es absurdo, ya que por Hipótesis Inductiva, A es diagonal dominante.

QED.

2.4. Planteo de Algoritmos

Para el armado de los algoritmos partimos de una división de los subproblemas a resolver:

- Crear la matriz en base al archivo de entrada, calculando los coeficientes a partir de las ecuaciones y ordenándolos de tal forma que quede banda, como se explicó anteriormente.
- Resolver la matriz con el método de Gauss.
- Calcular la L y la U.
- Resolver en base a una L y una U.
- Calcular la isoterma, con dos métodos diferentes: el primero, calculando el radio más cercano al valor buscado por cada ángulo; y el segundo, aproximando un valor entre los dos radios más cercanos al valor buscado.
- Determinar si es peligroso, de dos maneras diferentes: la primera viendo si algún punto de la isoterma sobrepasa un radio predeterminado cercano al radio externo, y la segunda calculando el porcentaje de puntos que sobrepasan dicho límite.

Algunos algoritmos los planteamos basándonos en sus versiones conocidas (y vistas en clase): método de Gauss y calculo de la L y la U. Luego los modificamos de acuerdo a algunos temas de optimización y detalles para las entradas particulares del problema a resolver.

Pasamos por una etapa de prueba de los algoritmos, tanto individualmente como en conjunto, donde tuvimos que dedicarnos a algunos errores que surgieron en cuanto a índices y casos borde.

Una vez terminados, optimizamos el algoritmo de Eliminación Gaussiana para matrices Banda. Como se puede ver en el apéndice B, el código aprovecha la propiedad de los 0s fuera de la sección diagonal para realizar menos iteraciones.

Teníamos planteado implementar las estructuras para aprovechar la propiedad de banda y reducir la complejidad espacial del algoritmo, pero por cuestiones de tiempo, decidimos enfocarnos en la experimentación y discusión.

Finalmente nos dedicamos a aproximar la isoterma de mejor manera y pasamos a experimentar.

2.5. Planteo de Mediciones Experimentales

Dividiremos los experimentos realizados en tres secciones, dependiendo el aspecto del algoritmo que se quiere medir y comparar.

- Complejidad Temporal: experimentos que apuntan a comparar el costo en tiempo de ejecución de los algoritmos, dependiendo del nivel de la granularidad (Experimento 1), del método escogido con temperaturas dinámicas, Gauss o LU (Experimento 2), y la comparación con el método que aprovecha el hecho de que la matriz sea banda (Experimento 3).
- Isoterma: experimentos que apuntan a comparar cómo varía la isoterma devuelta por el algoritmo modificando la granularidad de los radios (Experimento 4), granularidad de los ángulos (Experimento 5) y granularidad de radios y ángulos a la vez (Experimento 6). Además, se comparan dos métodos para calcular la isoterma: el primero, redondeando el resultado al radio más cercano correspondiente al valor buscado en cada ángulo; y el segundo, que calcula un promedio entre los radios más cercanos al valor buscado (Experimento 7). Un último experimento (Experimento 8) busca corroborar si vale la pena aumentar la granularidad para obtener resultados más precisos, en contraposición con un método menos costoso pero de menor precisión.
- Peligrosidad: experimentos que buscan comparar los dos métodos implementados para calcular la peligrosidad del horno según la isoterma (Experimento 9). El primer método, más brusco, busca solamente si un punto de la isoterma haya traspado el valor de la isoterma 500, mientras que el segundo, realiza un cálculo de porcentaje para calcular si la cantidad de puntos del horno en peligro supera un umbral predeterminado.

3. Resultados y Discusión

3.1. Resultados con respecto al Tiempo de ejecución

3.1.1. Experimento 1

El primer experimento consiste en correr el programa, variando la granularidad (en particular la del parámetro $m + 1$, que determina la cantidad de puntos para cada ángulo), con $ninst = 1$. Se busca comparar el tiempo de ejecución de los distintos algoritmos para resolver un sólo sistema, con un único término independiente (a esto se debe $ninst = 1$). Se espera que la Eliminación Gaussiana optimizada para banda se ejecute más rápido que las otras, y que la Descomposición LU sea la más lenta, ya que se deben resolver dos sistemas en vez de uno sólo en el caso de Gauss. El experimento consiste en correr 25 veces cada algoritmo con una serie de entradas en las cuales se mantienen constantes todos los parámetros excepto $m + 1$ (es decir, variando sólo la granularidad). Se mide el tiempo que tardan en correr estas 25 iteraciones para cada entrada y método de resolución del sistema lineal.

3.1.2. Resultados del Experimento 1

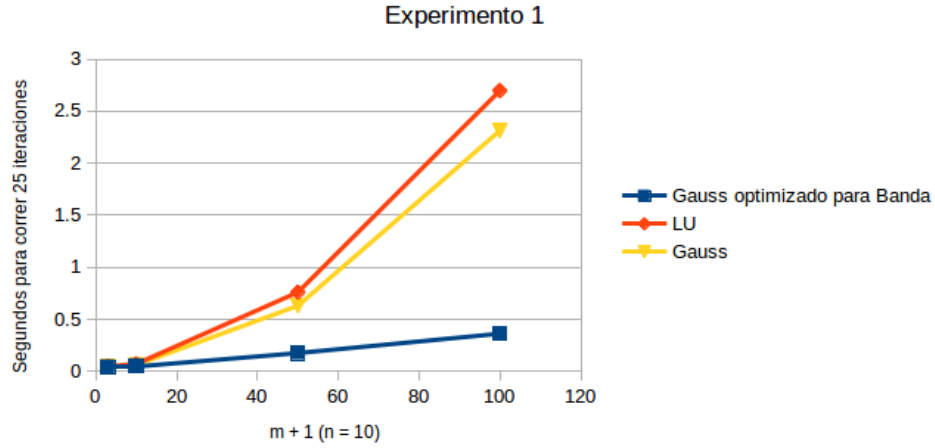


Figura 1: Tiempo de ejecución en función del tamaño de entrada, para $ninst = 1$

Los resultados del experimento 1 se pueden observar en la figura 1. Éstos son los esperados: la Eliminación Gaussiana optimizada es la implementación más rápida, seguida por la Eliminación Gaussiana (no optimizada), y la Descomposición LU siendo la más lenta. Cabe destacar la forma cuadrática de la curva del algoritmo de Gauss optimizado, comparada con la forma cúbica de las otras dos curvas. Ya que n se mantiene constante, el ancho de la banda (que, como se mencionó previamente es $n + 1$) también se mantiene constante, y por ende la complejidad del algoritmo se vuelve $O(tamanodelamatrix^2)$, en vez de $O(tamanodelamatrix^3)$, y el resultado obtenido es lo esperado.

3.1.3. Experimento 2

El segundo experimento consiste en correr el programa, con un mismo tamaño de entrada, pero variando $ninst$. El propósito de este experimento es observar como varía el tiempo de ejecución de los distintos algoritmos para sistemas con varios términos independientes.

Se espera que LU sea más rápido que Gauss, ya que sólo debe descomponer una vez al sistema y luego puede obtener la solución para distintos términos independientes solamente resolviendo dos sistemas triangulares (que se hace en tiempo $O(tamanodelamatrix^2)$).

Sin embargo, la relación entre el tiempo de ejecución del algoritmo LU (que tiene un costo inicial de $O(tamanodelamatrix^3)$, pero que al aumentar $ninst$, debería ser opacado por los costos de resolver múltiples veces el sistema en $O(tamanodelamatrix^2)$) y Gauss optimizado para Banda (que resuelve

una sola vez el sistema, pero debe triangularlo cada vez en $O(\text{tamanodelamatriz}^2)$ no es conocida de antemano; este tiempo es igual desde el punto de vista de la notación O grande, pero las constantes escondidas resultarán en uno de los métodos sobrepasando al otro.

3.1.4. Resultados del Experimento 2

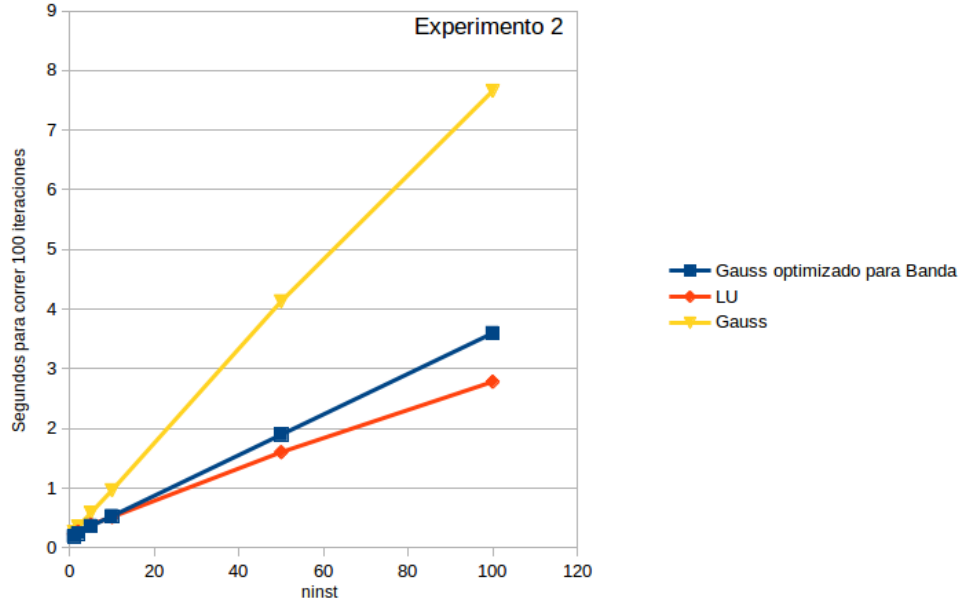


Figura 2: Tiempo de ejecución en función de ninst. Gráfico con todos los resultados del experimento.

Los resultados del experimento 2 se pueden observar en las figuras 2 y 3 (esta última representa los mismos resultados que la anterior, pero sólo para $ninst \leq 10$, para mayor claridad). Si bien los resultados concernientes a la Eliminación Gaussiana son los esperados, la relación entre el tiempo de ejecución de LU y Gauss optimizado es interesante: para valores pequeños de $ninst$, Gauss optimizado es más rápido, pero para valores grandes LU lo es. Esto se debe a que resolver los sistemas de LU tiene un costo menor que triangular y resolver los sistemas utilizando Gauss optimizado; sin embargo, para valores pequeños de $ninst$, el costo cúbico de la Descomposición LU tiene un mayor peso.

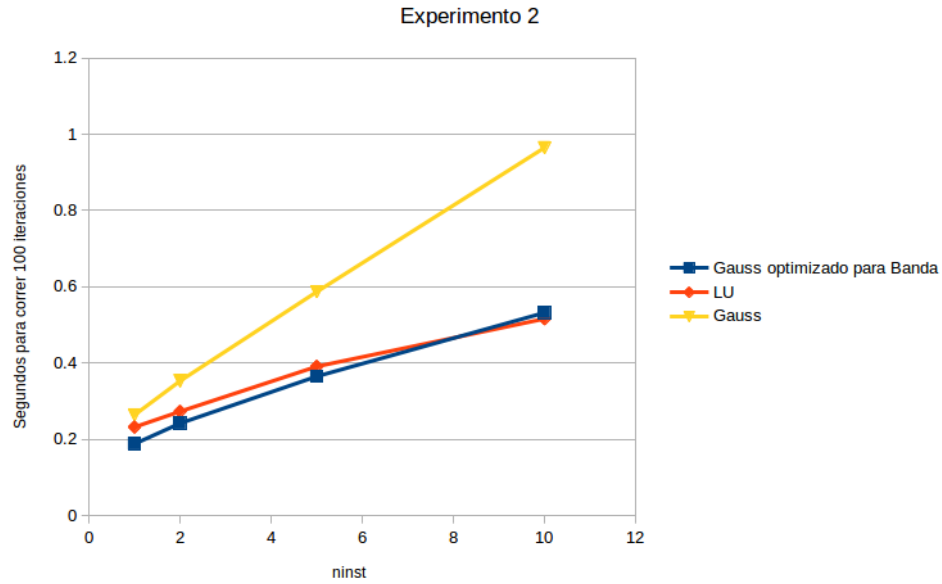


Figura 3: Tiempo de ejecución en función de ninst. Gráfico con los valores hasta ninst = 10

3.1.5. Experimento 3

El experimento 3 consiste en correr los algoritmos, variando $m + 1$ y n , pero manteniendo $(m + 1) * n$ constante. El propósito de este experimento es observar como varía el tiempo de ejecución de los algoritmos al variar sólo el ancho y el alto de la banda (que, como se demostró previamente, equivalen a $n + 1$), pero manteniendo constante la dimensión de la matriz. Se espera ver una clara diferencia en lo concerniente al algoritmo optimizado para banda. Los algoritmos de Gauss y LU, tienen una menor optimización para banda (que fue detallada en la Introducción Teórica), pero no se sabe de antemano cuán efectiva es.

3.1.6. Resultados del Experimento 3

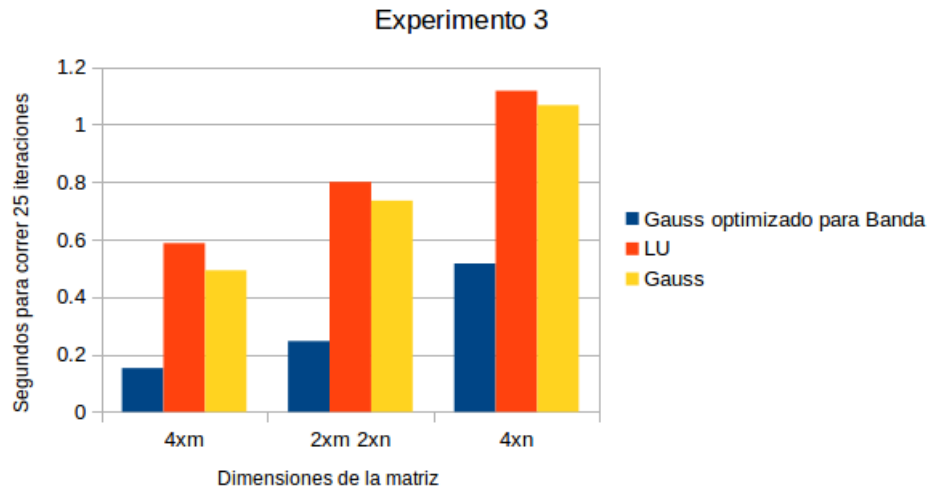


Figura 4: Tiempo de ejecución para banda de distinto tamaño, con dimensión de la matriz constante.

Los resultados del experimento 3 se pueden observar en la figura 4. Como era de esperar, se incrementa el tiempo de ejecución del algoritmo de Gauss optimizado para banda al incrementar el ancho y alto de

ésta; sin embargo, es interesante notar que también se incrementa el tiempo de ejecución de los otros dos algoritmos, y de forma considerable, lo que nos demuestra que la optimización mencionada tiene un efecto significativo sobre la velocidad de los métodos.

3.2. Resultados con respecto a la Isoterma

3.2.1. Experimento 4

En este experimento se busca comparar los resultados obtenidos en las temperaturas del horno y en el cálculo de la isoterma variando la granularidad de los radios. Para ello tomaremos dos inputs de entrada con los mismos valores de temperatura interna y externa (1500 y 0 respectivamente), mismos radio interno e externo (10 y 100) y misma cantidad de ángulos (30). El primero (4a) dividirá el horno en 10 radios, mientras que el segundo (4b) lo hará en 100. Conjeturamos que al tener más granularidad con respecto a los radios, se obtendrá una isoterma con menos imperfecciones, permitiendo evaluar la peligrosidad de manera más precisa.

3.2.2. Resultados del experimento 4

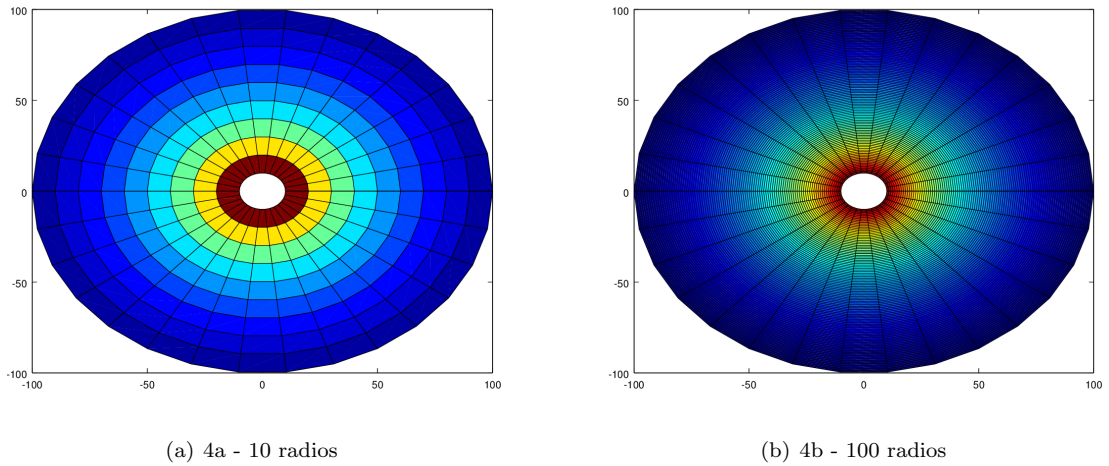


Figura 5: Experimento 4, gráfico de calor

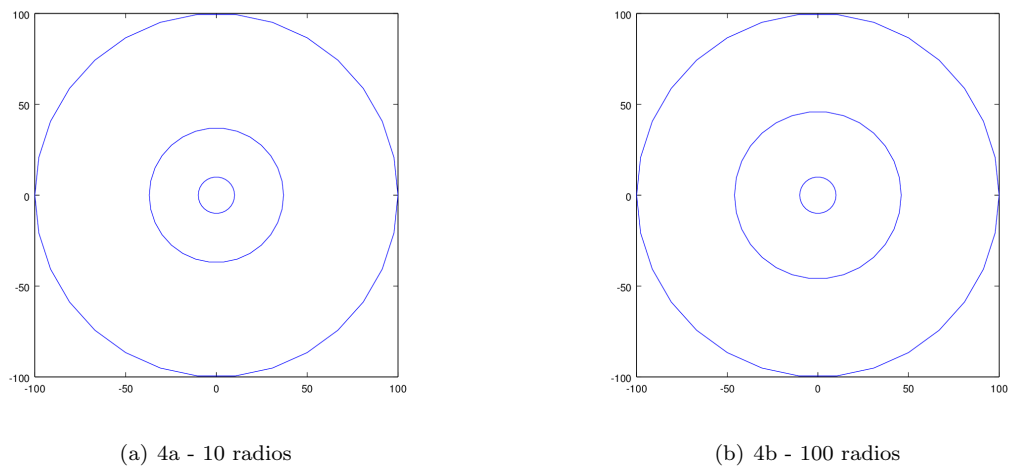


Figura 6: Experimento 4, gráfico de isoterma

Como se ve en los gráficos, se observa mayor precisión al aumentar la granularidad. La isoterma en el caso que toma mayor cantidad de radios se acerca al borde externo, lo que podría indicar que un test de peligrosidad podría llegar a dar un falso negativo si no se toma la cantidad adecuada de radios, provocando una falla grave en el sistema.

3.2.3. Experimento 5

Análogamente, en este experimento se observará como varían las temperaturas y la isoterma al modificar la cantidad de ángulos de entrada, manteniendo las demás variables constantes.

3.2.4. Resultados del experimento 5

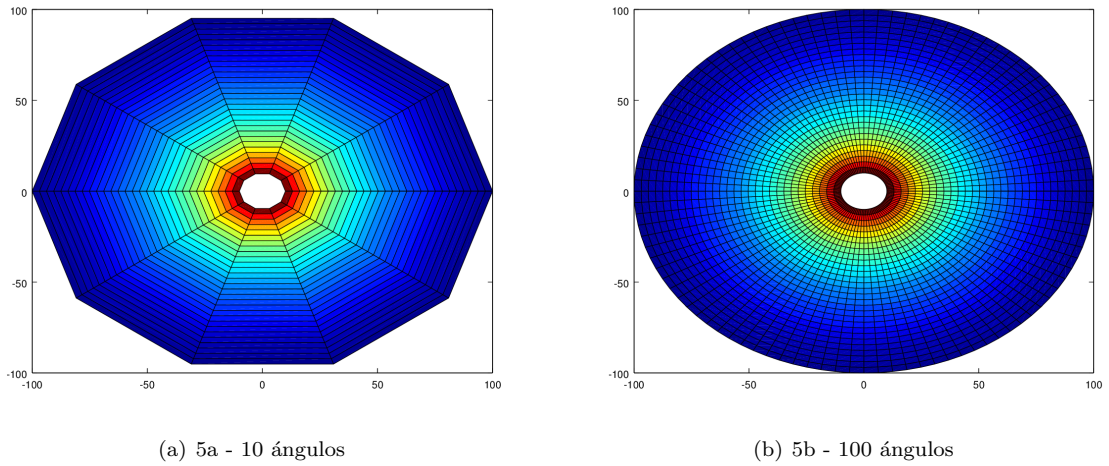


Figura 7: Experimento 5, gráfico de calor

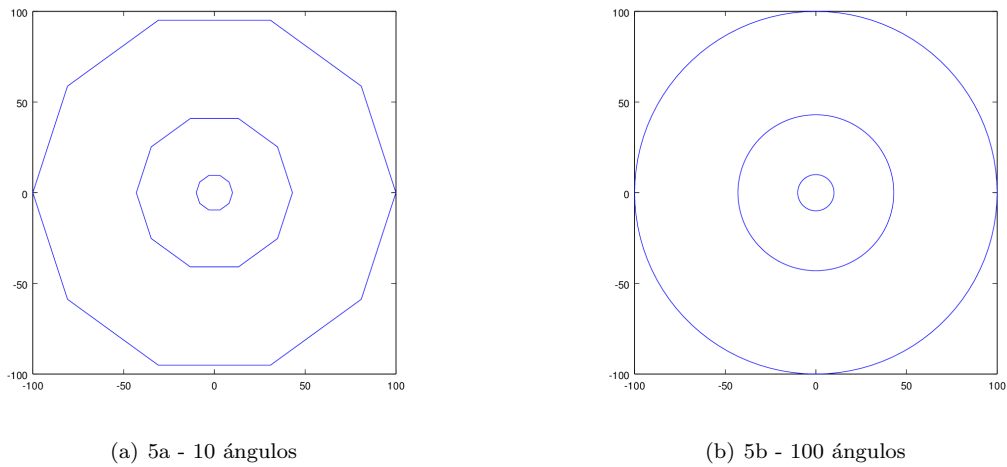


Figura 8: Experimento 5, gráfico de isoterma

Al igual que en el experimento anterior, la precisión aumenta al aumentar la cantidad de ángulos. En este caso la forma de la isoterma se ve que se asemeja a un círculo al aumentar la granularidad en los ángulos, que es la verdadera forma del horno. Esto es importante ya que decidir si se encuentra en condiciones de peligro con una isoterma tan deformada podría dar resultados incorrectos.

3.2.5. Experimento 6

Por último, en este experimento se desea comparar los resultados al aumentar la granularidad en ambas variables (radios y ángulos), con el objetivo de determinar cuál de ellas es más importante, refiriéndonos a su granularidad, a la hora de lograr una isoterma que permita calcular la peligrosidad de manera correcta.

3.2.6. Resultados del experimento 6

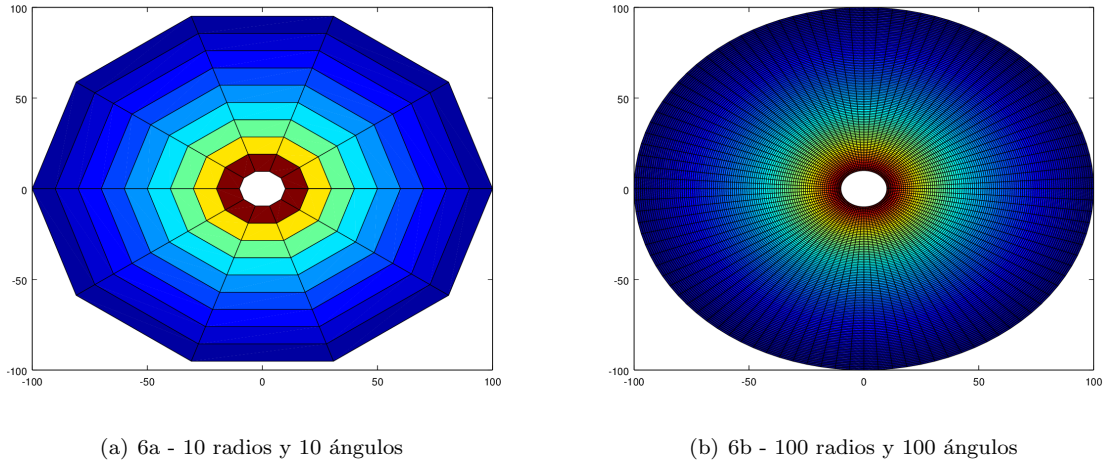


Figura 9: Experimento 6, gráfico de calor

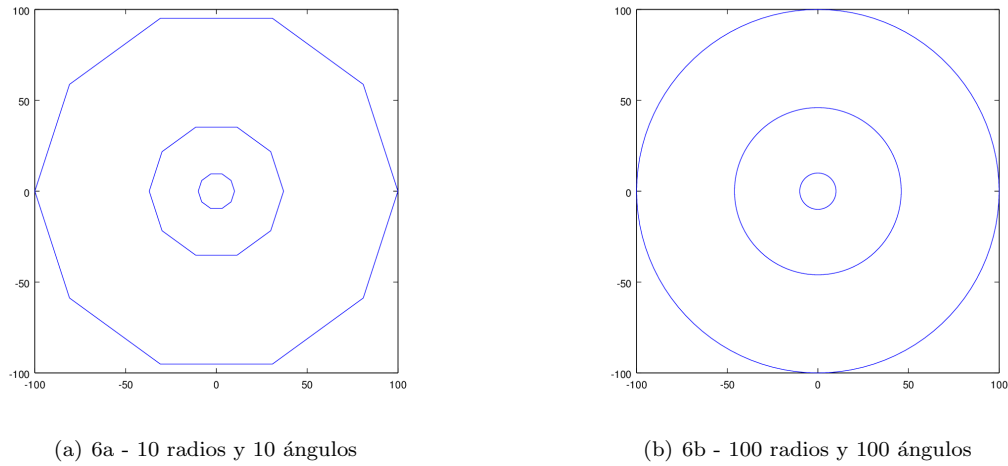


Figura 10: Experimento 6, gráfico de isoterma

Concluimos que la granularidad es una variable muy importante a la hora de determinar las temperaturas del horno, más allá de que aumente la complejidad temporal del algoritmo. Como se pudo ver en estos experimentos, disminuir la granularidad de alguna de las variables puede producir resultados incorrectos a la hora de calcular peligrosidad. Además, vemos que ambas variables (radios y ángulos) tienen igual importancia, a pesar de que el gráfico para poca cantidad de ángulos se vea más deformado a simple vista.

3.2.7. Experimento 7

Contamos con distintas formas de calcular la isoterma:

- Redondear al punto mas cercano.
- Promediar el punto entre los mas cercanos.

Creemos que cuanto mayor distancia entre puntos de discretización, mayor va a ser la distancia entre el punto calculado por el procedimiento 1 y el verdadero valor de radio de la isoterma. En cambio con el procedimiento 2 nos aseguramos acercarnos 'un poco mas' (mas allá de la distancia entre radios).

3.2.8. Primera entrada para el experimento 7

Para simular este caso particular vamos a tomar un estado donde el radio interno es 10mts y el externo es 10000mts. Tomamos tan solo 30 radios, por lo que la distancia entre radios es de 333mts. Tomamos temperaturas internas alternantes entre 5500, 1000 y 550 para lograr una isoterma no circular y variable. Planteamos como hipótesis que en el caso del procedimiento 1 la isoterma va a ser poco variable, y que, en caso de verse un salto, va a ser brusco (de 333mts) entre cada punto. Mientras que con el procedimiento 2 van a poder notarse mas cambios, pero mas leves, ya que al ir promediando va a variar entre puntos intermedios (menores a 333mts).

3.2.9. Resultados para la primera entrada para el experimento 7

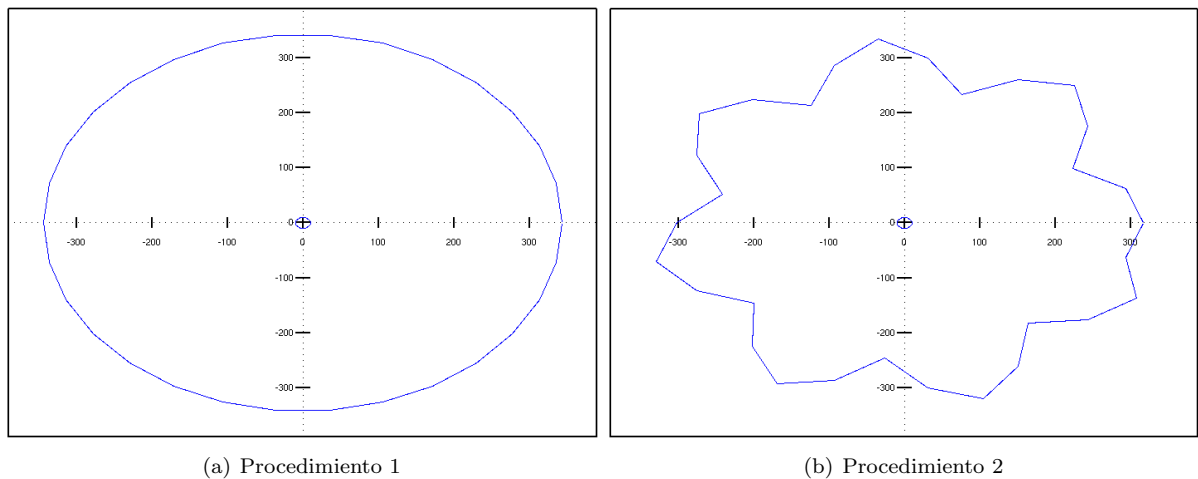


Figura 11: Experimento 7, primera entrada

En los gráficos se puede notar la diferencia entre los resultados del procedimiento 1 y el 2. En el caso del 1, al mantenerse la isoterma cerca del radio 343mts se redondea tomando el punto 343mts. Esto se da en todos los ángulos, por lo que se obtiene una isoterma circular. Mientras que en el caso del procedimiento 2 se pueden distinguir los cambios y variaciones de la isoterma acercándose por valores intermedios de los radios, consiguiéndose así saltos leves.

3.2.10. Segunda entrada para el Experimento 7

Vamos a modificar la entrada dejando solo temperaturas externas variables entre 1500 y 1200. Creemos que esto no va a modificar el gráfico de la isoterma según el procedimiento 1, ya que si disminuimos la diferencia de las temperaturas las isothermas de los diferentes ángulos deberían ocupar radios mas cercanos entre sí. Si creemos que va a cambiar el gráfico de la isoterma del procedimiento 2, ya que si hay poca diferencia entre las temperaturas externas debería acercarse un círculo.

3.2.11. Resultados para la segunda entrada para el Experimento 7

En el gráfico del procedimiento 1, podemos notar que la isoterma tiene forma circular (igual que en A), debido a la poca diferencia entre temperaturas externas y que redondeamos su valor. En cambio en el caso del procedimiento 2 notamos una gran diferencia con respecto al gráfico del A. Como pudimos

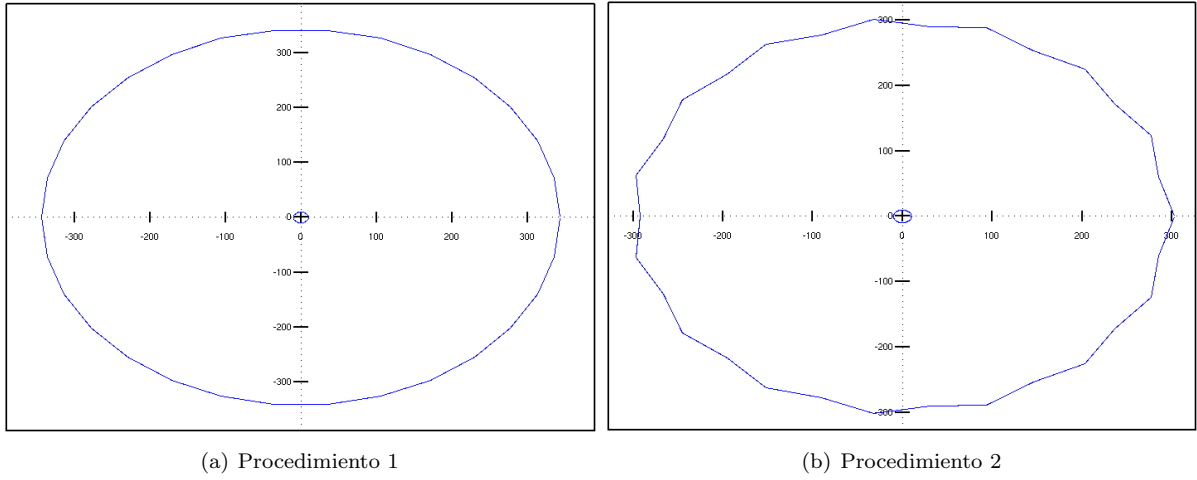


Figura 12: Experimento 7, segunda entrada

predecir, al acercar las temperaturas externas entre sí los puntos de la isoterma se posicionan mas cerca y ya no veos saltos tan bruscos. Lo que si podemos ver es que los valores de la isoterma varían entre 2 puntos (al igual que las temperaturas externas).

3.2.12. Experimento 8

Creemos que podríamos ahorrar tiempo si ejecutamos un estado con pocos radios y usando el procedimiento 2 para calcular la isoterma en vez de usar el procedimiento 1 con mas granularidad de radios. Para esto tomamos un estado con temperaturas externas variables entre 1500 y 1000 grados y vamos a calcular su isoterma 300 utilizando el procedimiento 2 con 10 radios. También calcularemos la misma isoterma utilizando el procedimiento 1 con 20 y 100 radios y compararemos los resultados para ver si podemos aproximar la isoterma utilizando el procedimiento 2 con poca granularidad de radios.

3.2.13. Resultados del Experimento 8

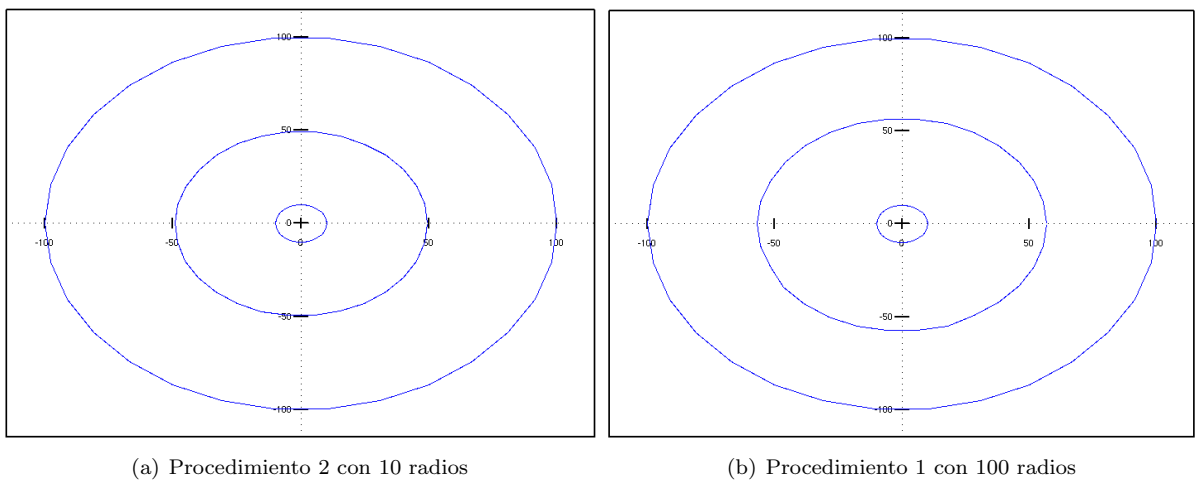


Figura 13: Experimento 8

Como vemos en los gráficos, la isoterma obtenida con el procedimiento 2 nos sirven para aproximar (a grandes rasgos) la posición de la isoterma, pero tomando en cuenta que calcular la isoterma con el procedimiento 1 y 100 radios no nos llevó un tiempo relevante podemos calcularla de esta forma y aproximar mucho mejor la isoterma (sin necesidad de mucho mas tiempo). Así concluimos que nuestro

primer pensamiento de que se podía aproximar la isoterma con rapidéz utilizando el procedimiento 2 era correcta, pero solo a grandes rasgos. Y teniendo en cuenta lo dicho antes podemos utilizar el procedimiento 1 sin necesidad de pagar con gran tiempo de ejecución.

3.3. Resultados con respecto a la Peligrosidad

3.3.1. Experimento 9

En este experimento se buscará comparar dos métodos implementados para decidir la peligrosidad de un horno a partir de la isoterma 500.

El primero, tan solo se fija si alguno de los puntos de la isoterma 500 sobrepasa una distancia predefinida al radio externo:

```
bool estaEnPeligro (vector<double>& isoVec, int ri, int re, double indiceDePeligro)
{
    double sePudrioTodo = ((double) re - ri) * indiceDePeligro;
    for (int i = 0; i < isoVec.size(); i++)
    {
        if (isoVec [i] >= sePudrioTodo) return true;
    }
    return false;
}
```

El segundo algoritmo, en cambio, calcula la cantidad de puntos en peligro y la compara con la cantidad total de puntos en la circunsferencia, calificándolo como en peligro si éste supera el 40%. Seleccionamos uno de los casos bordes para mostrar la diferencia entre los métodos, donde la mitad izquierda del horno es más fría (quizás por algún mecanismo de ventilación o enfriamiento) que la de la derecha.

3.3.2. Resultados del experimento 9

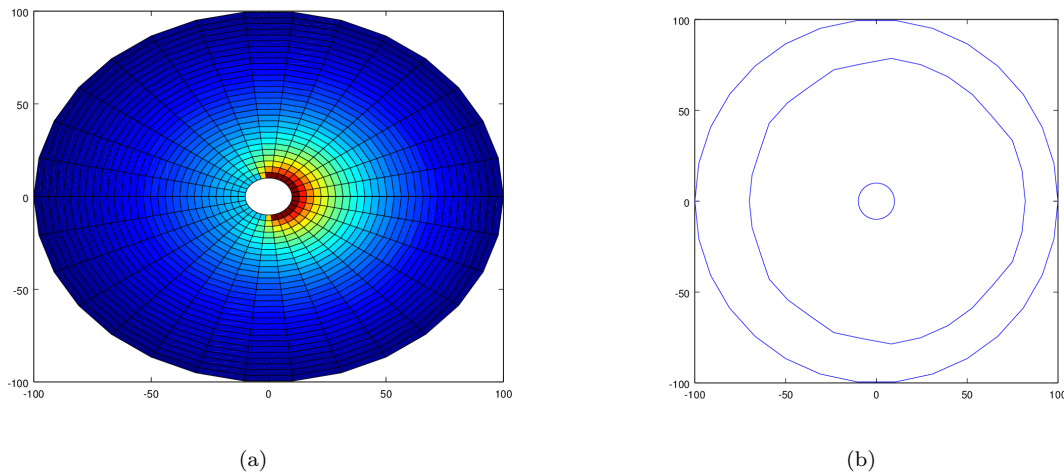


Figura 14: Experimento 9

estaEnPeligro da PELIGRO

estaEnPeligroPromedio da No esta en peligro

Para casos bordes como el mostrado en la figura 15, donde el horno tiene temperaturas más frías en un sector que en otro, los resultados de los métodos difieren. Quedará por decidir cuál de los dos es el más conveniente para cada situación: el primero, si nos encontramos en una situación crítica donde el menor exceso de temperatura pueda significar un grave problema; o el segundo, cuando la situación pueda ser más controlable si sólo una parte del horno haya excedido el límite de cercanía al borde externo.

4. Conclusiones

En primer lugar, concluimos (como se esperaba), que la Eliminación Gaussiana optimizada para banda es el algoritmo más eficiente para resolver sistemas con pocos términos independientes distintos. Gauss resultó ser más eficiente que LU, pero por un factor muy pequeño.

LU resultó ser aún más eficiente que Gauss optimizada para sistemas con muchos términos independientes distintos, a pesar de que ambos tienen complejidad cuadrática (en el tamaño de la matriz) después de la primera solución (en el caso de LU, la descomposición original es cúbica).

La conclusión más interesante que obtuvimos fue respecto al efecto de las optimizaciones de banda: incluso las optimizaciones que originalmente consideramos menores para banda en Gauss y LU tienen un efecto muy significativo en el tiempo de ejecución de los algoritmos.

De estos comentarios se desprende la idea de que contar con cada detalle del sistema a modelar puede permitirnos plantear de mejor manera el problema y la forma en la que lo vamos a resolver, optimizando los resultados que luego se van a obtener. Por ejemplo plantear el sistema de ecuaciones de cierta forma nos permite aprovechar las características de una matriz banda y poder mejorar el rendimiento de nuestro programa.

También pudimos acercarnos mas a la investigación, sus formas y métodos, así como la idea de recibir colaboración por parte de especialistas para plantear las formas de resolver el problema, las ecuaciones y datos extras que pueden ayudarnos para dar con la solución y optimizar el proceso.

5. Apéndices

5.1. Apéndice A

Enunciado

Se debe implementar un programa en C o C++ que tome como entrada los parámetros del problema (r_i , r_e , $m + 1$, n , valor de la isoterma buscada, T_i , $T_e(\theta)$) que calcule la temperatura dentro de la pared del horno utilizando el modelo propuesto en la sección anterior y que encuentre la isoterma buscada en función del resultado obtenido del sistema de ecuaciones. El método para determinar la posición de la isoterma queda a libre elección de cada grupo y debe ser explicado en detalle en el informe.

El programa debe formular el sistema obtenido a partir de las ecuaciones (1) - (6) y considerar dos métodos posibles para su resolución: mediante el algoritmo clásico de Eliminación Gaussiana y la Factorización LU. Finalmente, el programa escribirá en un archivo la solución obtenida con el formato especificado en la siguiente sección.

Como ya se ha visto en la materia, no es posible aplicar los métodos propuestos para la resolución a cualquier sistema de ecuaciones. Sin embargo, la matriz del sistema considerado en el presente trabajo cumple con ser diagonal dominante (no estricto) y que, ordenando las variables y ecuaciones convenientemente, es posible armar un sistema de ecuaciones cuya matriz posee la propiedad de ser *banda*. Luego, se pide demostrar (o al menos dar un esquema de la demostración) el siguiente resultado e incluirlo en el informe:

Proposición 1 *Sea $A \in \mathbb{R}^{n \times n}$ la matriz obtenida para el sistema definido por (1)-(6). Demostrar que es posible aplicar Eliminación Gaussiana sin pivoteo.*¹

La solución del sistema de ecuaciones permitirá saber la temperatura en los puntos de la discretización. Sin embargo, nuestro interés es calcular la isoterma 500, para poder determinar si la estructura se encuentra en peligro. Luego, se pide lo siguiente:

- Dada la solución del sistema de ecuaciones, proponer una forma de estimar en cada ángulo de la discretización la posición de la isoterma 500.
- En función de la aproximación de la isoterma, proponer una forma (o medida) a utilizar para evaluar la peligrosidad de la estructura en función de la distancia a la pared externa del horno.

En función de la experimentación, se busca realizar dos estudios complementarios: por un lado, analizar cómo se comporta el sistema y, por otro, cuáles son los requerimientos computacionales de los métodos. Se pide como mínimo realizar los siguientes experimentos:

1. Comportamiento del sistema.

- Considerar al menos dos instancias de prueba, generando distintas discretizaciones para cada una de ellas y comparando la ubicación de la isoterma buscada respecto de la pared externa del horno. Se sugiere presentar gráficos de temperatura o curvas de nivel para los mismos, ya sea utilizando las herramientas provistas por la cátedra o implementando sus propias herramientas de graficación.
- Estudiar la proximidad de la isoterma buscada respecto de la pared exterior del horno en función de distintas granularidades de discretización y las condiciones de borde.

2. Evaluación de los métodos.

- Analizar el tiempo de cómputo requerido para obtener la solución del sistema en función de la granularidad de la discretización. Se sugiere presentar los resultados mediante gráficos de tiempo de cómputo en función de alguna de las variables del problema.

¹Sugerencia: Notar que la matriz es diagonal dominante (no estrictamente) y analizar qué sucede al aplicar un paso de Eliminación Gaussiana con los elementos de una fila.

- Considerar un escenario similar al propuesto en el experimento 1. pero donde las condiciones de borde (i.e., T_i y $T_e(\theta)$) cambian en distintos instantes de tiempo. En este caso, buscamos obtener la secuencia de estados de la temperatura en la pared del horno, y la respectiva ubicación de la isoterma especificada. Para ello, se considera una secuencia de $ninst$ vectores con las condiciones de borde, y las temperaturas en cada estado es la solución del correspondiente sistema de ecuaciones. Se pide formular al menos un experimento de este tipo, aplicar los métodos de resolución propuestos de forma conveniente y compararlos en términos de tiempo total de cómputo requerido para distintos valores de $ninst$.

De manera opcional, aquellos grupos que quieran ir un poco más allá pueden considerar trabajar y desarrollar alguno(s) de los siguientes puntos extra:

1. Notar que el sistema resultante tiene estructura *banda*. Proponer una estructura para aprovechar este hecho en términos de la *complejidad espacial* y como se adaptarían los algoritmos de Eliminación Gaussiana y Factorización LU para reducir la cantidad de operaciones a realizar.
2. Implementar dicha estructura y las adaptaciones necesarias para el algoritmo de Eliminación Gaussiana.
3. Implementar dicha estructura y las adaptaciones necesarias para el algoritmo de Factorización LU.

Finalmente, se deberá presentar un informe que incluya una descripción detallada de los métodos implementados y las decisiones tomadas, el método propuesto para el cálculo de la isoterma buscada y los experimentos realizados, junto con el correspondiente análisis y siguiendo las pautas definidas en el archivo `pautas.pdf`.

5.2. Apéndice B

Código fuente para la creación de la matriz

```
vector< vector<double> > crearMatriz (double n, double m, double ri, double deltaR,
double deltaA){

    vector< vector<double> > matriz((m+1)*n, vector<double>((m+1)*n));

    //Completo primeras n filas y ultimas n filas con 1s en la diagonal
    for (int i = 0; i < n; i++){
        matriz[i][i] = 1;
    }
    for (int i = (n*(m+1) - n); i < n*(m+1); i++){
        matriz[i][i] = 1;
    }

    //Completo resto de las ecuaciones usando las formulas
    for (int j = 1; j < m; j++){
        double r = ri + j * deltaR;
        for (int k = 0; k < n; k++){
            //uso para calcular en la matriz la posicion j,k = n x j + k

            // t j, k
            matriz[n*j + k][n*j + k] = (-2*pow(r,2)*pow(deltaA, 2) + r*deltaR
            *pow(deltaA, 2) - 2*pow(deltaR, 2))/(pow(deltaR, 2)*pow(r, 2)
            *pow(deltaA, 2));

            // t j-1, k
            matriz[n*j + k][n*(j-1) + k] = (r - deltaR)/(pow(deltaR, 2)* r);

            // t j+1, k
            matriz[n*j + k][n*(j+1) + k] = 1 / pow(deltaR, 2);

            // t j, k-1
            matriz[n*j + k][n*j + (k - 1 > 0? k - 1 : n - 1)] = 1 / (pow(r,
            2)*pow(deltaA, 2));
        }
    }
}
```

```

        //t j, k+1
        matriz[n*j + k][n*j + (k + 1 < n - 1? k + 1 : 0)] = 1 / (pow(r,
            2)*pow(deltaA, 2));
    }
}
return matriz;
}

```

Código fuente para Eliminación Gaussiana con aprovechamiento de banda

```

void EliminacionParaBanda (vector<vector<double>>& mat, int bandaAltura, int bandaAncho,
    vector<double>& adjunta)
{
    int h = mat.size();
    int w = mat[0].size();
    for (int PivoteDiagonal = 0; PivoteDiagonal < w; PivoteDiagonal++)
    {
        //corre desde la fila de abajo de la diagonal hasta la bandaAltura-iesima de
        //abajo de la diagonal
        //el resto son 0 por la propiedad banda
        for (int FilaActual = PivoteDiagonal + 1; FilaActual < PivoteDiagonal +
            bandaAltura && FilaActual < h; FilaActual++)
        {
            if (mat[FilaActual][PivoteDiagonal] != 0)
            {
                double coeficiente = mat[FilaActual][PivoteDiagonal] / mat[
                    PivoteDiagonal][PivoteDiagonal];
                adjunta[FilaActual] -= adjunta[PivoteDiagonal] * coeficiente;
                //corre desde la columna de la diagonal hasta la bandaAncho-iesima de la
                //derecha de la diagonal
                //el resto son 0 por la propiedad banda
                for (int ColumnaEnSuma = PivoteDiagonal; ColumnaEnSuma < FilaActual +
                    bandaAncho && ColumnaEnSuma < w; ColumnaEnSuma++)
                {
                    mat[FilaActual][ColumnaEnSuma] -= mat[PivoteDiagonal][ColumnaEnSuma]
                        * coeficiente;
                }
            }
        }
    }
}

```

6. Referencias

<http://en.cppreference.com/w/>

R. Burden y J.D.Faires, *Análisis numérico*, International Thomson Editors, 1998