

Laboratorio de Métodos Numéricos

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

Trabajo Práctico Número 1

Trabajamos y nos divertimos...

Integrante	LU	Correo electrónico
Zar Abad, Ciro Román	129/15	ciromanzar@gmail.com
Lopez Valiente, Patricio	457/15	patricio454@gmail.com
Romero, Lucas Rafael	440/12	lucasrafael.romero@hotmail.com

Reservado para la catedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. Introducción teórica	4
1.1. Introducción del problema	4
1.2. Eliminación Gaussiana	5
1.3. Factorización LU	5
2. Desarrollo	5
2.1. Métodos utilizados	5
2.2. Implementación	6
2.3. Búsqueda de la isoterma	6
2.4. Evaluación de peligrosidad	7
2.5. Experimentos	8
3. Resultados	10
3.1. Experimentos sobre las discretizaciones.	10
3.1.1. Experimentos 1 a 4	10
3.1.2. Discusión de los resultados	10
3.1.3. Experimentos 5 y 6	12
3.1.4. Discusión de los resultados	13
3.2. Experimentos de Complejidad	17
3.2.1. Experimento 1	17
3.2.2. Discusión de los resultados	18
3.2.3. Experimento 2	18
3.2.4. Experimentos Tres	20
3.2.5. Experimento Cuatro	22
3.2.6. Discusión de los resultados	23
4. Conclusiones	24
5. Apendices	25
5.1. Enunciado	25
5.2. Código fuente	30
5.2.1. Eliminación Gaussiana	30
5.2.2. Factorización LU	30
5.2.3. Backward Substitution	31

5.2.4. Forward Substitution	31
5.3. Proposición 1	31
5.3.1. Demostración	31
6. Referencias	33

Resumen

En este trabajo presentamos el problema de representar computacionalmente el comportamiento de temperatura dentro de las paredes de un alto horno. El objetivo es encontrar los puntos internos de la pared donde la temperatura alcanza los 500°C, a lo que llamamos isoterma 500, en base a las mediciones de temperaturas en el exterior de dicha pared. Luego, esta información se utilizará para analizar el riesgo de derrumbe de la estructura.

Para ello desarrollamos un modelo para discretizar los puntos de la pared, crear el sistema lineal que representa el problema y analizamos distintos métodos numéricos para resolverlo.

A continuación realizamos experimentaciones para comparar la performance de dichos métodos. Corremos las implementaciones para matrices de distintos tamaños y para instancias similares variando tan solo el vector de resultados. Además comparamos los resultados obtenidos contra los proporcionados por matlab, que es la herramienta de la que disponemos cuyos resultados consideramos son los más acercados a la realidad.

Finalmente, exponemos nuestras conclusiones.

1. Introducción teórica

1.1. Introducción del problema

A partir de los datos de un alto horno, lo modelamos como una matriz de puntos discretos tomando una cantidad finita de radios y de ángulos y pretendemos encontrar la isoterma 500. Es decir, los puntos sobre la pared del horno que tienen temperatura de 500 °C.

Para encontrar estos puntos contamos con la ecuación de Laplace:

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

Esto presenta 2 problemas: No es simple encontrar las derivadas de la función y el dominio de esta son los reales. Teniendo eso en cuenta, nos inclinamos a resolver una aproximación más simple:

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

Esta ecuación nos deja con un sistema lineal que podemos resolver para encontrar la temperatura en cada punto del modelo. A partir de eso, interpolamos los valores para obtener la isoterma buscada.

1.2. Eliminación Gaussiana

Uno de los algoritmos posibles para resolver el sistema lineal es la eliminación Gaussiana. Aprovechando el hecho que cambiar las filas de una matriz por combinaciones lineales de estas devuelve un sistema equivalente, es posible operar entre las filas para conseguir una matriz triangular superior equivalente. En definitiva un sistema más simple de resolver.

El algoritmo es el siguiente:

```

procedure ELIMINACIÓN GAUSSIANA( $A \in \mathbb{R}^{n \times n}$ )
  for  $i = 1$  to  $n - 1$  do
    for  $i = 1$  to  $n - 1$  do
       $m_{j,i} \leftarrow \frac{a_{j,i}}{a_{i,i}}$ 
       $F_j \leftarrow F_j - m_{j,i} F_i$ 
    end for
  end for
end procedure

```

$\triangleright F_k = e_k A$

El algoritmo asume que el elemento k -ésimo de la diagonal luego de la iteración k es distinto de 0. Más adelante demostraremos que es una suposición válida para esta clase de sistemas.

1.3. Factorización LU

La factorización LU es similar a la eliminación Gaussiana en el sentido que calcula la misma matriz triangular superior de la misma manera. Sin embargo, la factorización LU se guarda los multiplicadores en una matriz triangular inferior L de manera que $A = LU$.

De esta manera, tenemos una forma simple de expresar la matriz A . Para resolver el sistema $Ax = b$ resolvemos:

$$\begin{cases} Ly = b \\ Ux = y \end{cases}$$

La ventaja que presenta este enfoque frente a la eliminación gaussiana es que, manteniendo la matriz A , es posible resolver el sistema para distintos vectores b sin tener que recalcular la triangulación.

2. Desarrollo

2.1. Métodos utilizados

Los métodos numéricos implementados en este trabajo práctico fueron Eliminación Gaussiana y factorización LU. Estos fueron empleados para la búsqueda de la isoterma de un Alto Horno y resultaron de gran utilidad, ya que se contó con un método para discretizar la temperatura de las paredes del Alto Horno, de forma que

se convertía en un sistema lineal. Esto se logró mediante ecuaciones de calor provistas por la cátedra, y propiedades que cumplen los Altos Hornos. La discretización del sistema fue llevada a cabo tomando una serie de radios y ángulos, donde cada punto marcaba una incógnita.

2.2. Implementación

Para la implementación de los métodos usados, se empezó primero por definir la clase matriz por sobre los cuales iban a operar, la implementación de la misma no fue una tarea difícil. Como siguiente paso se pasó a implementar la Eliminación Gaussiana y por posterior Factorización LU, ya que su implementación una vez que se cuenta con el algoritmo de Eliminación Gaussiana pasa a ser casi trivial. Posteriormente a esto se pasó a implementar los algoritmos para resolver sistemas triangularizados, comúnmente conocidos como "backward substitution" y "forward substitution". Como paso final se implementaron las funciones para la carga, creación y escritura del sistema Matricial.

Durante la primera parte de la implementación, los problemas que surgían, no eran de mayor importancia, ya que eran problemas de tipo o similar, por lo que no eran difíciles de localizar, aunque no eran triviales. Lo mismo ocurrió con la implementación de Eliminación Gaussiana, Factorización LU, "backward substitution" y "forward substitution".

Sin embargo a la hora de implementar los métodos creación y escritura surgieron problemas que demoraron mucho más en ser resueltos, esto se debía a que eran de difícil resolución por su poca notoriedad, ya que se debían a una mala interpretación del sistema en un caso (confundimos radios con ángulos), o problemas de precisión en las funciones nativas de C++. Estas resultaron particularmente difíciles de resolver, ya que requirieron que veamos documentación de C++, y diversos foros como "StackOverflow" y similares hasta dar con la solución, ejemplos de este tipo de problemas se dieron con el operador $<<$ y con el operador de División $/$, que truncaban los resultados al imprimirlos.

2.3. Búsqueda de la isoterma

A la hora de aproximar la ubicación de la isoterma 500, se analizaron distintos tipos de interpolaciones. Finalmente se optó por utilizar interpolación lineal por su simpleza, y porque en caso de no ser lo suficientemente precisa, esto se puede compensar con una discretización de radios más fina.

La ecuación de interpolación lineal, dado un ángulo θ fijo, la función $T(r)$ que determina la temperatura de la pared en el punto (r, θ) y dos radios contiguos r_1 y r_2 tal que $T(r_2) \geq \text{ISO}$ y $T(r_1) < \text{ISO}$ (siendo ISO la temperatura de la isoterma buscada, ej: 500) es la siguiente:

$$ISO = T(r_1) + \frac{T(r_2) - T(r_1)}{r_2 - r_1}(r - r_1) \quad (3)$$

Despejando r , y escribiendo $(r_2 - r_1)$ como Δr queda:

$$r = \frac{(ISO - T(r_1))\Delta r}{T(r_2) - T(r_1)} + r_1 \quad (4)$$

2.4. Evaluación de peligrosidad

A la hora de proponer una forma de evaluar la peligrosidad de la estructura en función de la distancia de la isoterma a la pared externa del horno nos encontramos con problemas y planteos diversos, y aunque se investigó sobre el concepto detrás del Alto Horno al desconocer cuestiones ligadas a la física y ingeniería de la estructura no podíamos encontrar una forma librada de ambigüedad. Algunas de las primeras ideas que surgieron a un principio fueron:

- "Medir la distancia de la media de la isoterma con respecto a la pared exterior, y si es cercana, decir que peligra la estructura". Ésta fue rápidamente descartada ya que al no conocer que como funciona la disipación de calor en los Altos Hornos, no podíamos establecer cuando había o no cercanía.
- "Tomar como medida la cercanía de la isoterma más grande al radio exterior". También fue descartada por razones similares a la anterior, además del hecho de que no es muy expresiva en función de que arrojaría el mismo índice de peligro para una medida en que la isoterma es homogéneamente "grande", que para una que presenta un "outlier".
- "Emitir el índice de peligro en función de si los valores obtenidos se correspondían con respecto a los de una distribución normal dada por la media y la varianza de la isoterma". Ésta fue descartada no trivialmente, ya que aunque en un principio parecía ser una buena opción permitía que isotermas con formas similares a las de una flor (gran varianza y una media promedio) aparenten ningún peligro cuando en realidad lo hacen.

Finalmente se pensó una forma que tenía que ser fiable, y que los conocimientos necesarios para entender su funcionamiento no tenían que depender del conocimiento de la estructura, sino de datos provistos por ella.

Por lo que se ideó el siguiente protocolo para medir el peligro de la estructura:

- 1. Se obtiene a partir de la consulta con un profesional, o del fabricante, datos de la ubicación de la isoterma en un caso controlado borde, donde el peligro es nulo, pero está próximo a no serlo. Si no dispusiéramos de estos medios para conseguir el mismo, como conocemos que la temperatura de la pared exterior se encuentra entre 50 °C y 200 °C, podríamos suponer que un buen candidato para ser nuestro caso es por ejemplo uno donde las temperaturas de la pared exterior varían entre 160 °C y 190 °C, ya que este es un caso que está cercano

a dejar de ser promedio, pero la pared exterior presenta temperaturas mayores a las esperadas. A la media de esta Isoterma la llamaremos ISOCTRL

- 2. Apartir de la isoterma ISOCTR mediremos generaremos tres valores:
 - ISOMD: que consiste en la resta entre la media de la isoterma medida e ISOCTR. En el caso de ser Negativo, se le asigna 0.
 - ISOMAX: que consiste en la resta entre el maximo de la isoterma medida e ISOCTR. En el caso de ser Negativo, se le asigna 0.
 - NRMISO: que consiste en la resta entre el radio exterior e ISOCTR.(esta se usara para normalizar el Indice)

- 3. Ahora de esta forma se obtiene el siguiente indice de peligro:

$$IPREGULAR = \frac{ISOMD}{NRMISO}$$

$$IPLOCAL = \frac{ISOMAX}{NRMISO}$$

donde IPREGULAR e IPLOCAL varían entre 0 y 1, y a medida que estos crecen, el peligro de rotura en la estructura lo hace tambien. De esta forma al disponer de estos dos indices, se puede obtener informacion adicional, por ejemplo si los dos indices son similares se podria inferir que la perdida de calor es homogeneamente alta o normal dependiendo de que valor entre 0 y 1 nos den. De la misma forma en un caso donde IPREGULAR es relativamente mas chico que IPLOCAL, se podria inferir que posiblemente hay un segmento de la pared que no esta disipando el calor de forma adecuada.

2.5. Experimentos

Los experimentos llevados a cabo fueron ideados siguiendo los siguientes lineamientos:

- Para los experimentos referidos a la busqueda de la isoterma 500 y los posibles efectos que podia tener la granulacion del sistema sobre la presicion de la misma se decidieron llevar a cabo cuatro macro experimentos, los cuales consistian en distintas formas de discretizar sistemas en funcion de los radios y angulos, cada uno de ellos divididos en dos casos:
 - Caso A: Presentando un sistema equilibrado con temperaturas dentro del rango de lo esperado.
 - Caso B: Una medición en la pared externa con picos en las temperaturas y que presentan una posible situación de riesgo (igual para todos los experimentos, pero con mayor o menor cantidad de sensores dependiendo del caso).

Para cada uno de estos casos, las discretizaciones elegidas fueron las siguientes:

- Exp. 1: donde la cantidad ángulos y radios tomados era pequeña (10 ángulos y 10 radios).

- Exp. 2: donde la cantidad ángulos y radios tomados era grande (50 ángulos y 50 radios).
- Exp. 3: donde la cantidad ángulos era grande y de radios pequeña (100 ángulos y 20 radios).
- Exp. 4: donde la cantidad ángulos era pequeña y de radios grande (20 ángulos y 100 radios).

Para su realización se construyeron los sistemas y se los resolvio utilizando el método de Eliminación Gaussiana. Luego se realizaron las correspondientes imágenes que grafican los resultados obtenidos.

Dado que los resultados obtenidos difirieron bastante de lo que habíamos esperado encontrarnos en base a nuestra intuición, optamos por realizar un experimento nuevo para reforzar las conclusiones.

Basandonos en el caso B de los experimentos anteriores, dividimos el nuevo experimento en dos etapas:

- Exp. 5: dejando fijo la cantidad de ángulos y variando la de radios.
- Exp. 6: dejando fijo la cantidad de radios y variando los ángulos.

En estos nuevos casos, optamos por superponer los gráficos para poder compararlos mas eficientemente.

- Para los experimentos concernientes a la efectividad en terminos de complejidad temporal de la Eliminacion Gaussiana y la Factorizacion LU, se llevaron a cabo tres experimentos en los que se buscaba ver la efectividad de estos para diferentes aspectos:
 - El experimento numero uno se ideo con el fin de ver como crecia la complejidad de temporal de la Eliminacion Gaussiana y la Factorizacion LU, en funcion del tamaño del sistema a resolver, es decir de la Matriz que definia el sistema, para ello se midio el tiempo que los metodos implementados tardaban en resolver una sola instancia para matrices de tamaños que variaban desde 9x9 hasta 1600x1600.
 - El experimento numero dos se ideo para ver como respondian nuestro metodos numericos ante una medicion de un sistema para un gran numero de instancias, para ellos se midio el tiempo que tardaban en resolver j instancias distintas, para una matriz de 30x30, con $1 \leq j \leq 25$.
 - El experimento numero tres se presento como un experimento posible apartir de observar los primeros resultados del experimento dos, al observar las difencias entre los metodos propuestos para cuando se instanciaba a la misma matriz del sistema con mas de una instacia. Para ello se midio el tiempo que tardaba en dar con el resultado particular de cada instancia, una matriz de 30x30 y 10 instancias.
 - El experimento número 4 pretende comparar los resultados obtenidos por las implementaciones de fact. LU y eliminación gaussiana contra los

resultados arrojados por Matlab para los mismos sistemas. Para eso, se calcularon 2 instancias distintas mediante los distintos métodos (cabe destacar que los resultados de nuestras implementaciones fueron idénticos) y tomamos la diferencia relativa¹ entre ellos.

para la medición de tiempos se empleó la librerías *"chrono"* y *"ctime"* de *c++*, estas mismas cuentan con diversas funciones para la medición de tiempos dentro del código, fueron empleados estos procesos y no otros conocidos como el comando *"time"* de *bash* u otros, ya que al permitir mediciones en el código no solo ofrecían a priori un resultado más fiable y menos permeable a procesos externos, sino que dada su maleabilidad facilitaban enormemente la toma de mediciones.

- Finalmente, con el objetivo de analizar la exactitud de nuestros algoritmos para resolver sistemas lineales, realizamos dos experimentos que consistieron en generar un sistema pequeño, resolverlo usando los dos métodos implementados (EG y LU) y compararlos con los resultados obtenidos resolviendo el mismo sistema con Matlab.

3. Resultados

3.1. Experimentos sobre las discretizaciones.

3.1.1. Experimentos 1 a 4

La experimentación con distintas discretizaciones del espacio dio diversos resultados. Por un lado, se pueden observar en [Figura 1](#) y [Figura 2](#) los gráficos de las temperaturas obtenidas en el experimento con la discretización más fina y equilibrada, con medición de temperaturas equilibrada (A), y con una posible situación de riesgo (B).

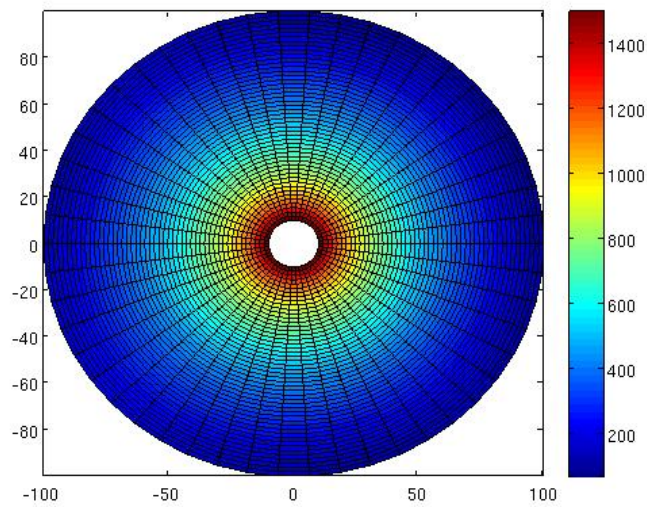
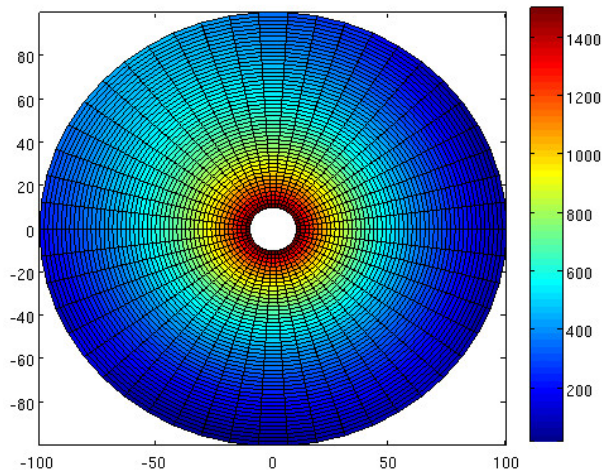
Luego, en base a estas temperaturas obtenidas, se utilizó el método de interpolación lineal para calcular la posición de la isoterma 500. En [Figura 3](#) [Figura 4](#) [Figura 5](#) y [Figura 6](#) se puede observar la ubicación de la isoterma, para cada una de los experimentos en el caso A.

De la misma forma, para el caso B se llevaron a cabo los mismos experimentos como se puede observar en [Figura 7](#) [Figura 8](#) [Figura 9](#) y [Figura 10](#)

3.1.2. Discusión de los resultados

En un principio, antes de realizar los experimentos, pensábamos que llegaríamos a la conclusión de que una discretización pobre llevaría a resultados muy distintos a los obtenidos por una discretización más fina. Por ejemplo, esperábamos encontrarnos con algún caso de riesgo con temperaturas altas que el experimento 1 ignore,

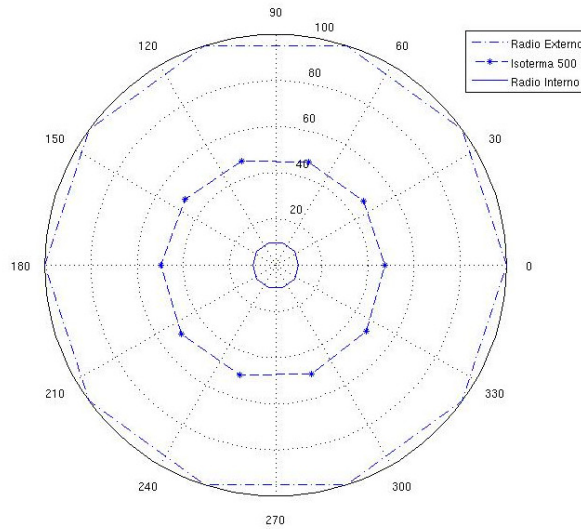
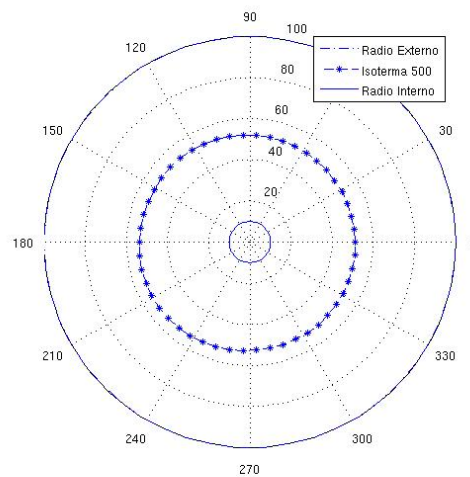
¹restar los resultados de los vectores y dividir el vector por uno de los vectores resultados

Figura 1. Grafico de temperaturas del experimento 2, caso A.**Figura 2.** Grafico de temperaturas del experimento 2, caso B.

pero incluso éste resultó dar una aproximación bastante acertada.

Esto se puede deber a que por la forma en la que se trasmite el calor por la pared, dada por la ecuación de Laplace, no es posible que se registren altas temperaturas en un punto de la pared, sin que se vea reflejado de alguna forma en los puntos cercanos.

Esto se puede observar tambien comparando las [Figura 1](#) y [Figura 2](#), donde claramente se ve el cambio gradual del calor, a pesar de la diferencia extrema de temperaturas registradas en la pared exterior.

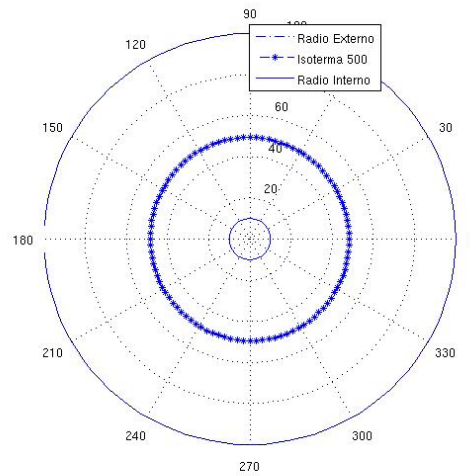
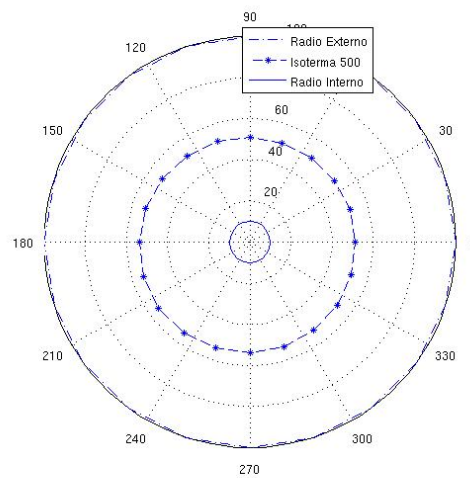
Figura 3. Grafico de la isoterma 500 del experimento 1, caso A.**Figura 4.** Grafico de la isoterma 500 del experimento 2, caso A.

3.1.3. Experimentos 5 y 6

Con el objetivo de corroborar lo observado en los experimentos previos, se realizan dos nuevos experimentos. Además, se espera llegar a alguna conclusión respecto a cual de los rangos de la discretización, la cantidad de ángulos o la cantidad de radios, es más importante para lograr un resultado fiable.

En [Figura 11](#) se pueden observar los resultados superpuestos de variar la cantidad de ángulos (N), dejando los radios fijos en 20.

De la misma forma, en [Figura 12](#) se ven los resultados de variar la cantidad de

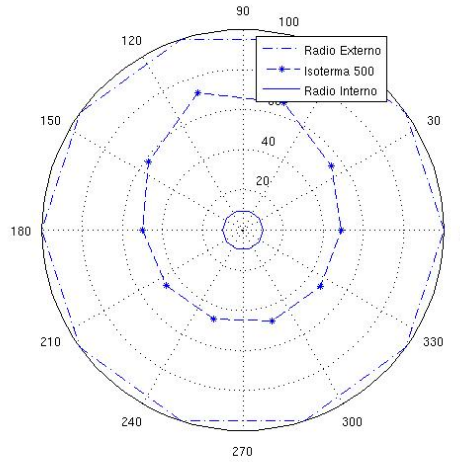
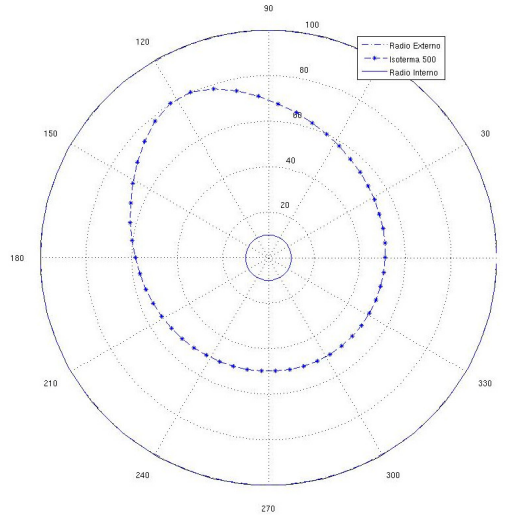
Figura 5. Grafico de la isoterma 500 del experimento 3, caso A.**Figura 6.** Grafico de la isoterma 500 del experimento 4, caso A.

radios ($M + 1$), dejando fijos 20 ángulos.

3.1.4. Discusión de los resultados

Con estos nuevos experimentos, confirmamos lo observado en los experimentos 1 a 4.

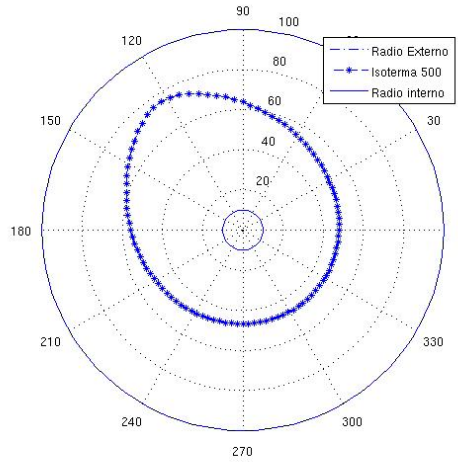
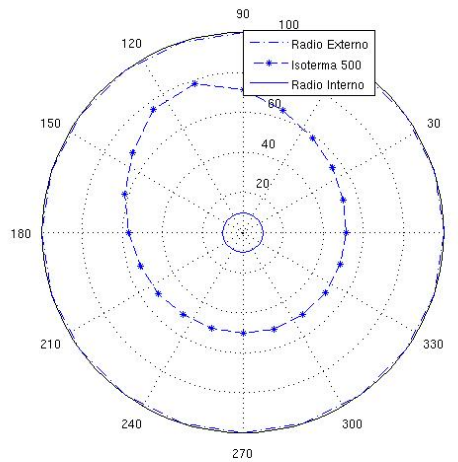
En la [Figura 11](#) vemos que el único experimento que devolvió un resultado alejado de los demás, es en el que usamos solamente 10 radios, pero que a partir de aumentar esta variable a 30 se equilibra el sistema y a partir de allí se consiguen resultados muy similares.

Figura 7. Grafico de la isoterma 500 del experimento 1, caso A.**Figura 8.** Grafico de la isoterma 500 del experimento 2, caso A.

Siendo más exactos, la mayor diferencia entre el punto más cercano de la isoterma a la pared exterior obtenido con 30, 50, 70 y 90 radios se dió entre el resultado de $M+1 = 30$ y $M+1 = 90$, y es tan solo del orden de 1,2 %

$$\frac{ISO_{max}(M+1=90) - ISO_{max}(M+1=30)}{R_e - R_i} = \frac{77,935665 - 76,818577}{100 - 10} = 0,012412089 \quad (5)$$

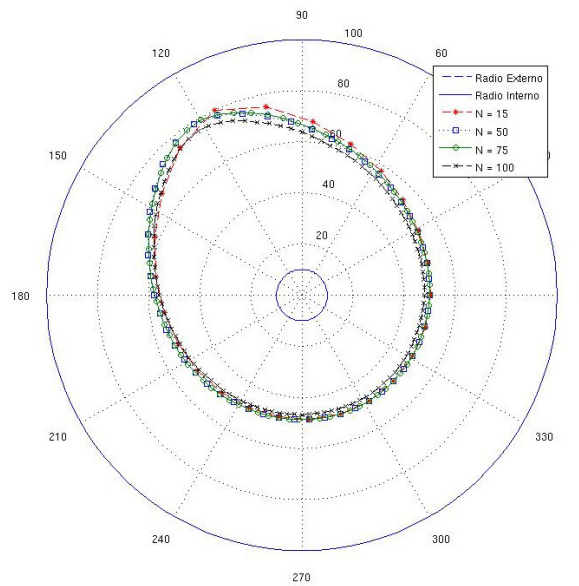
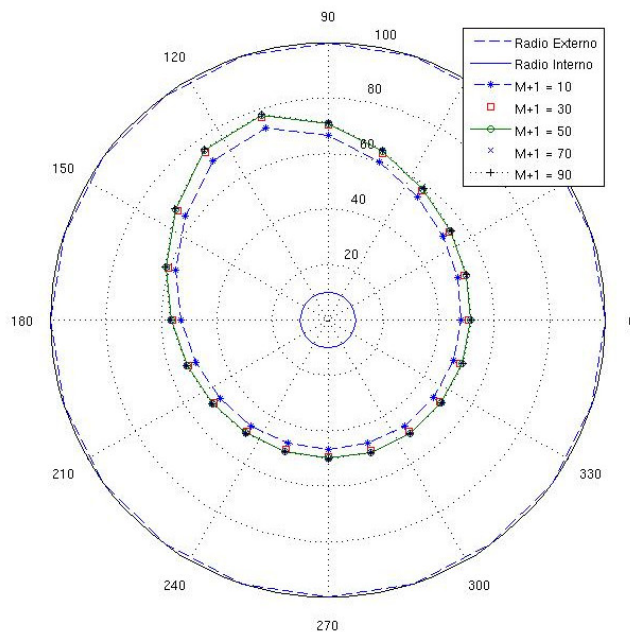
Sin embargo, y ésto es lo que más nos sorprendió, en la [Figura 11](#) observamos que las diferencias entre las distintas aproximaciones de la isoterma es muy pequeña a pesar de la gran diferencia de los valores utilizados. Más aún, no parece haber una relación tan directa entre una discretización de ángulos más fina y un resultado

Figura 9. Grafico de la isoterma 500 del experimento 3, caso A.**Figura 10.** Grafico de la isoterma 500 del experimento 4, caso A.

más preciso. Por ejemplo, el resultado obtenido con 100 y con 15 ángulos tienen una diferencia de apenas 0,69 % entre si, pero de 5,7 % respecto del valor real”(obtenido con una discretización más fina y equilibrada de 100x100, que requirió varios minutos de cómputo)

$$\frac{ISO_{max}(N = 100) - ISO_{max}(N = 15)}{R_e - R_i} = \frac{75,872249 - 75,248781}{100 - 10} = 0,006927422 \quad (6)$$

En conclusión, podemos inferir que no es efectivo tomar discretizaciones con valores de N y M muy diferentes, ya que aumentar una de las variables y no la otra, no necesariamente lleva a mejores resultados. Lo más eficiente es tomar una

Figura 11. Grafico de las isoterma 500 del experimento 5.**Figura 12.** Grafico de las isoterma 500 del experimento 6.

discretización equilibrada en N y M , sin necesidad de que sean valores muy altos, que requerirían mucho tiempo de cómputo.

3.2. Experimentos de Complejidad

3.2.1. Experimento 1

En este experimento se esperaba ver como se comportaban temporalmente nuestras implentaciones de la Factorizacion LU y Eliminacion Gaussiana. Previamente se había demostrado de forma teórica que la complejidad de estos metodos era cúbica, por lo que se esperaba demostrar empíricamente que nuestra implementación cumpliera con dicha complejidad.

Figura 13. Grafico del Tiempo de Resolucion del sistema usando E.G.

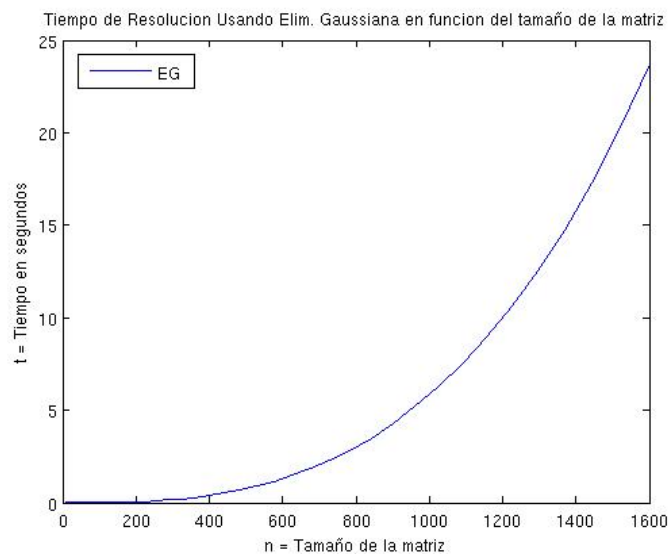
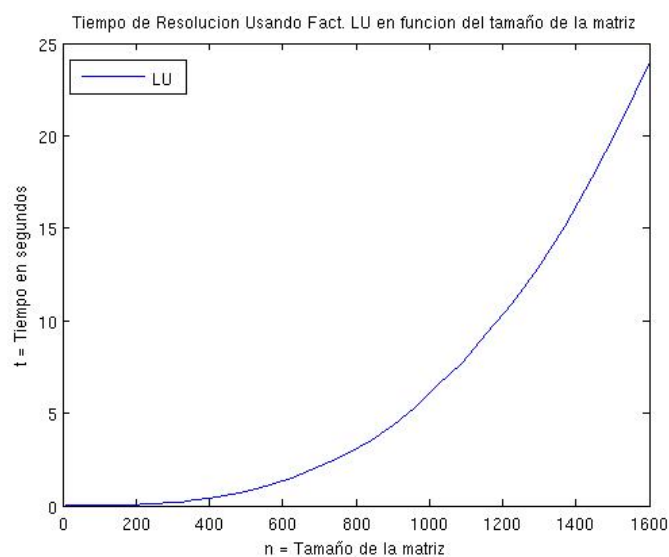


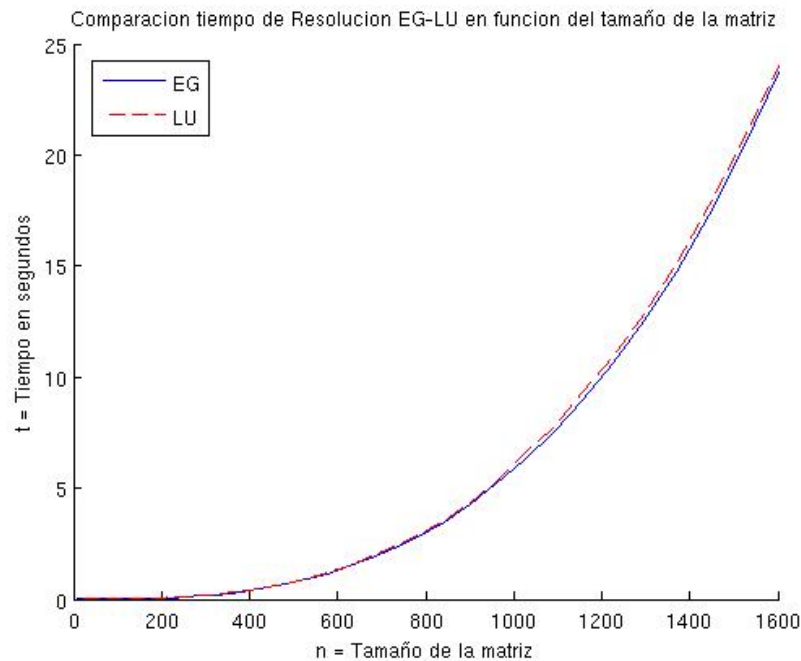
Figura 14. Grafico del Tiempo de Resolucion del sistema usando Fact. LU



Tanto en la [Figura 13](#), como en la [Figura 14](#) se puede observar que la complejidad temporal crece en orden cubico como se habia previsto, esto se debe a que

nuestros metodos son de cota inferior Cubica en peor caso, por lo que apesar de que nuestra implentacion no esta lo suficientemente optimizada para tener una constante minima, es bastante aceptable ya que como se ve en los graficos es de orden cubica.

Figura 15. Comparacion entre Fact.LU y E.G. del Tiempo de Resolucion del sistema



3.2.2. Discusión de los resultados

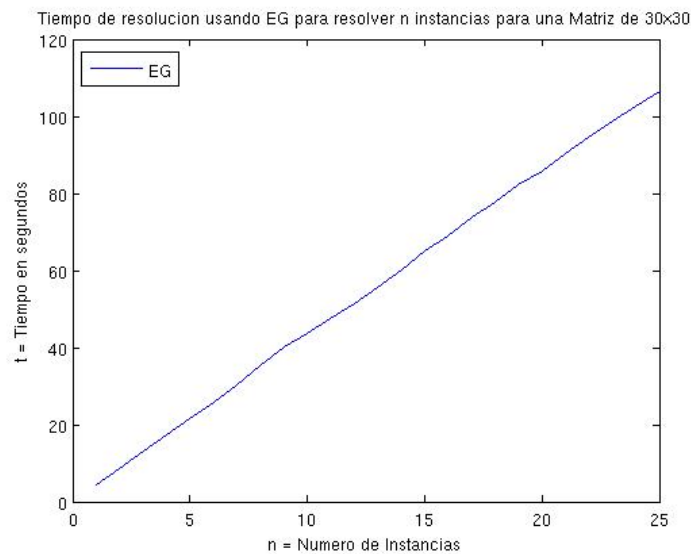
Una de las hipotesis que sosteniamos antes de las experimentacion es que Fact. LU iba a ser mejor que E.G., a pesar de ejecutar una sola instancia, esto como se puede observar en la [Figura 15](#) es falso, aunque por un margen pequeño, suponemos que esto se debe a que mientras la E.G. solo triangula el sistema y resuelve un sistema triangular superior, Fact. LU tambien debe triangular el sistema y resolver un sistema triangular superior, y ademas debe resolver un sistema triangular inferior, por lo que el pequeño margen de tiempo debe encontrarse ahi.

3.2.3. Experimento 2

En este experimento se esperaba ver como variaba la complejidad temporal de nuestros metodos en funcion del numero de instancias a resolver. Esperabamos que Fact. LU fuera mejor que E.G., haciendo uso de las propiedades inertes al proceso de Fact. LU vistas en la teorica.

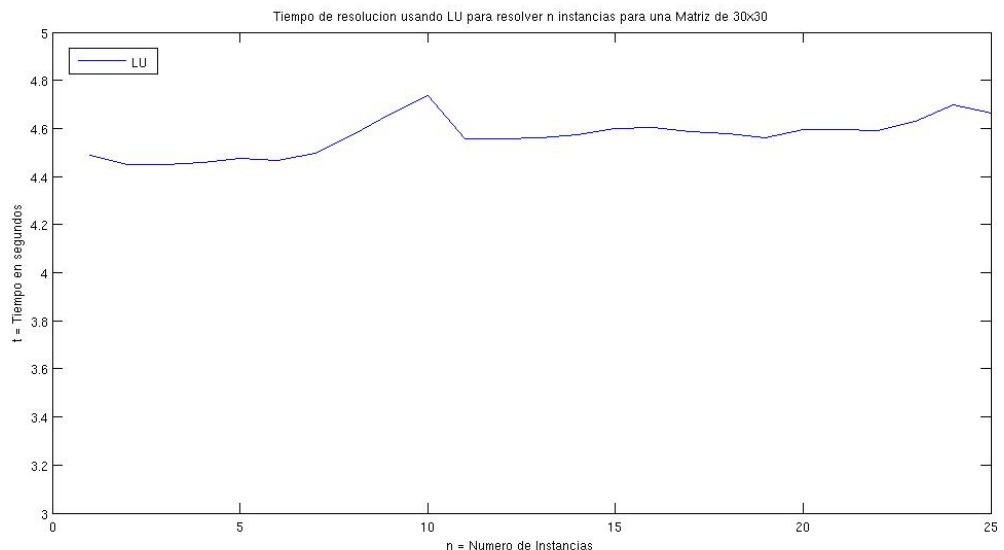
Como se puede ver en la [Figura 16](#), el tiempo que tarda en resolver n instancias distintas usando E.G. es similar a:

Figura 16. Grafico del Tiempo de Resolucion del sistema usando E.G. para n instancias en una Matriz de 30x30



$n \times$ Tiempo para una instancia Esto se debe principalmente a que debe triangular la matriz para cada instancia que ejecuta, por lo que para cada instanciar que resuelva debe volver a gastar tiempo cubico en triangularizar.

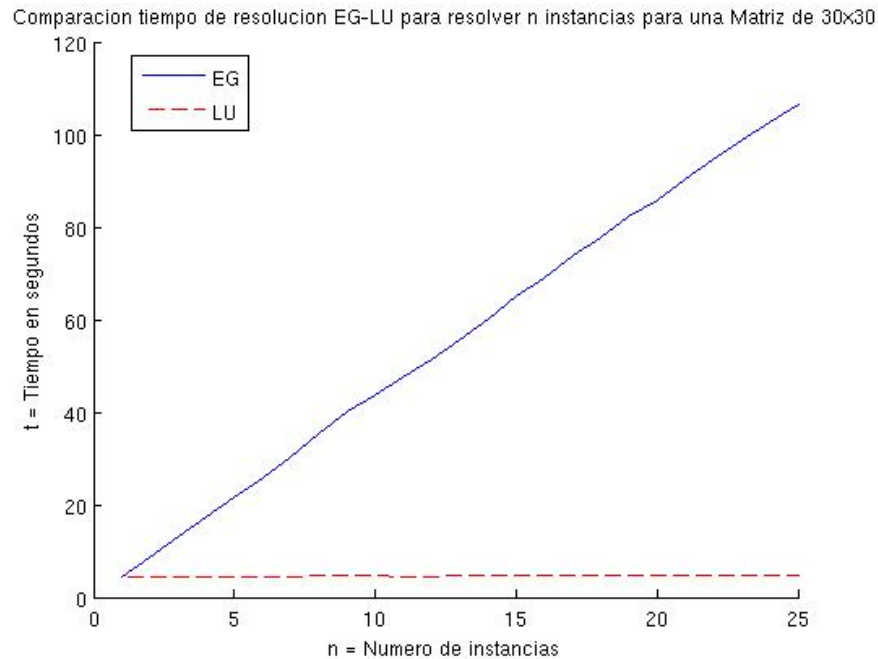
Figura 17. Grafico del Tiempo de Resolucion del sistema usando Fact. LU para n instancias en una Matriz de 30x30



En la [Figura 17](#), en cambio observamos que usando Fact. LU resolver una instancia o numero mucho mayor de estas no repercute perceptiblemente en la complejidad temporal, en este caso al tener la Fact. LU la propiedad que para resolver un sistema: $A \cdot x = b$, se puede resolver los sistemas $A = L \cdot U$, $L \cdot y = b$ y $U \cdot x = y$. Por

lo que de esta forma solo es necesario triangular la matriz en la primera instancia, y en las siguientes solo se debe resolver dos matrices triangular inferior y superior respectivamente, lo que hace que en las siguientes instancias el tiempo sea mínimo.

Figura 18. Comparacion entre Fact.LU y E.G. del Tiempo de Resolucion del sistema para n instancias en una Matriz de 30x30



En la [Figura 18](#) se puede ver claramente que mientras la Fact. LU es practicamente constante con respecto al numero de instancias a resolver, la E.G. es lineal. Este resultado fue una de las cosas que motivaron la realizacion del experimento tres con el fin de terminar de comprobar las hipotesis planteadas en las conclusiones de este experimento.

3.2.4. Experimentos Tres

En este experimento fue motivado apartir de los resultados obtenidos en el experimento dos, se esperaba comprobar dos cosas que la complejidad para resolver la instancia numero uno o la i esima era constante, o practicamente la misma en la E.G. y que en el caso de la Factorizacion LU la instancia numero uno era la que cargaba como el factor temporal dominante y el tiempo de resolucion de las siguientes era casi despreciable. Para esto se ideo un test donde se media el tiempo que tomaba resolver la instancia uno a la diez para una matriz de 30x30.

En la [Figura 19](#) se ve claramente como el tiempo que toma resolver cualquier instancia es practicamente el mismo, esto se produce a que para cada instancia debe resolver el sistema entero, por ser un algoritmo poco eficiente en terminos de reusabilidad.

En la [Figura 20](#) se ve como se confirma la hipotesis inicial de que la complejidad

Figura 19. Grafico del Tiempo de Resolucion de cada instancia particular usando E.G. para 10 instancias en una Matriz de 30x30

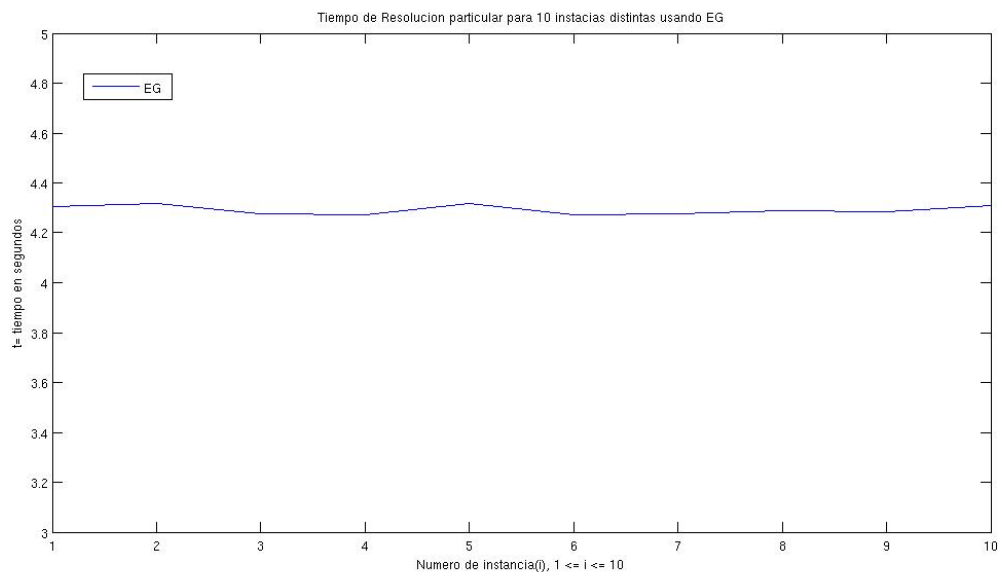
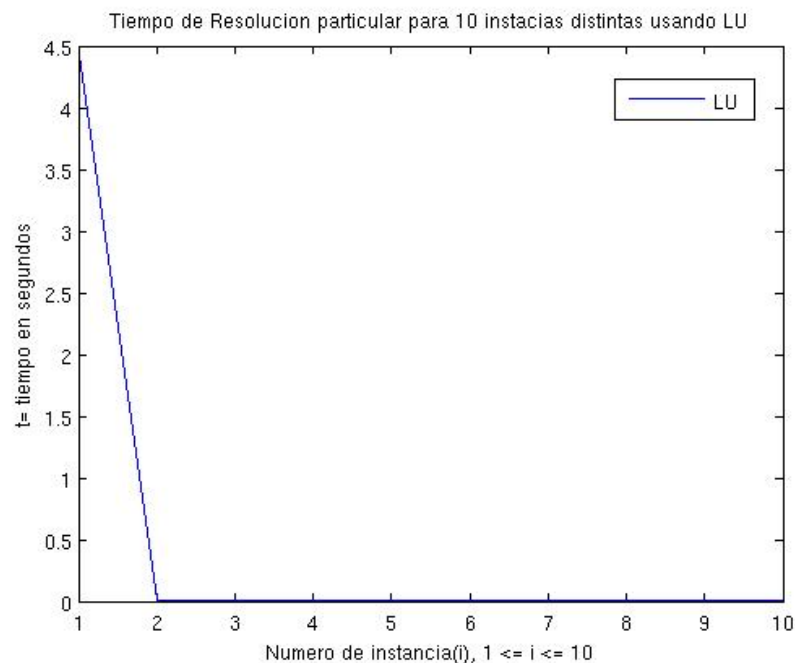


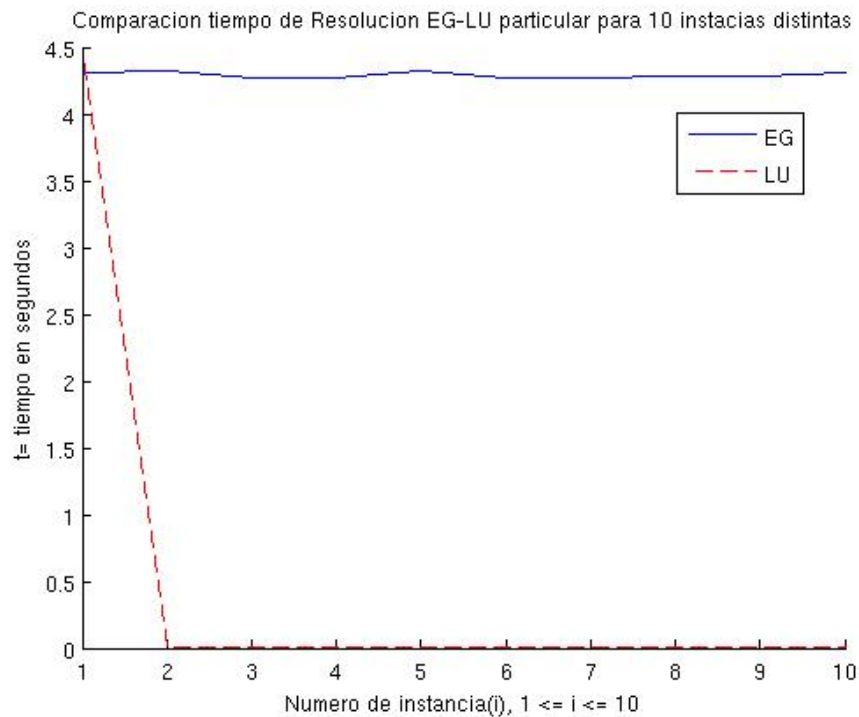
Figura 20. Grafico del Tiempo de Resolucion de cada instancia particular usando Fact. LU para 10 instancias en una Matriz de 30x30



temporal es consumida mayormente en factorizar la matriz y que la resolucion de los sistemas ya triangulados es despreciable temporalmente hablando.

Para representar mas visualmente la diferencia entre Factorizacion LU y Eliminacion Gaussiana se presenta en la [Figura 21](#) una comparacion donde se observa claramen-

Figura 21. Comparacion entre Fact.LU y E.G. del Tiempo de Resolucion de cada instancia particular para 10 instancias en una Matriz de 30x30



te la diferencia de rendimiento entre Factorizacion LU y E.G. luego de ejecutar la primera instancia, esto como se infirio anteriormente tiene que ver con la obligatoriedad en el caso de E.G. de tener que triangular en cada instancia la matriz del sistema.

3.2.5. Experimento Cuatro

El objetivo de este experimento es analizar la exactitud de los métodos implementados.

Elegimos crear dos instancias de un sistema pequeño, con $N = 5$ y $M + 1 = 5$, ya que esto generará dos vectores resultado de tamaño 25, de los cuales 10 son triviales, y consideramos los 15 resultados suficientes para evaluar en promedio cual es el error generado.

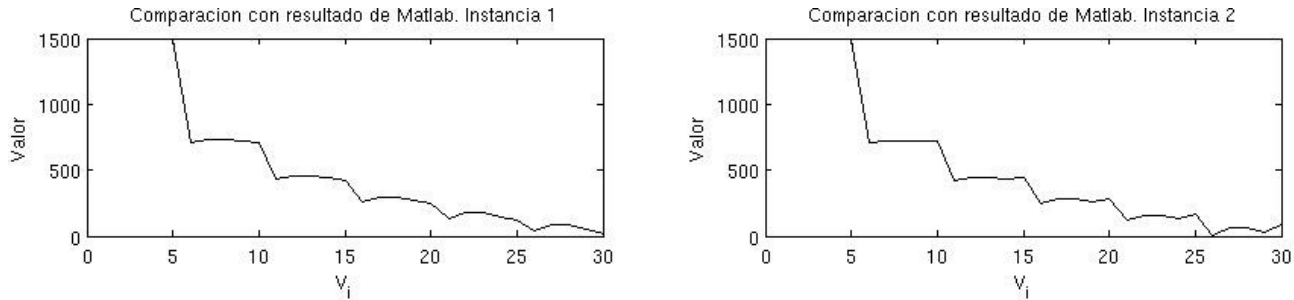
Entendemos que Matlab debe utilizar los algoritmos con las mejores optimizaciones posibles por lo que tomamos estos resultados como representativos de los resultados reales.

El primer paso del experimento consistió en resolver las dos instancias del sistema con los dos metodos implementados, pero nos encontramos con que, tal como esperabamos, conseguimos exactamente los mismos resultados. Por lo tanto, para el resto del experimento no hará falta comparar cada uno de los resultados por separado. Nos referiremos a los resultados obtenidos con cualquiera de los dos metodos

como $v1$ y $v2$

Los resultados obtenidos con Matlab, a los que llamaremos $m1$ y $m2$, se compararon como se puede ver en la [Figura 22](#) en la que, para cada instancia, no se puede notar a simple vista ninguna diferencia ya que los gráficos se superponen.

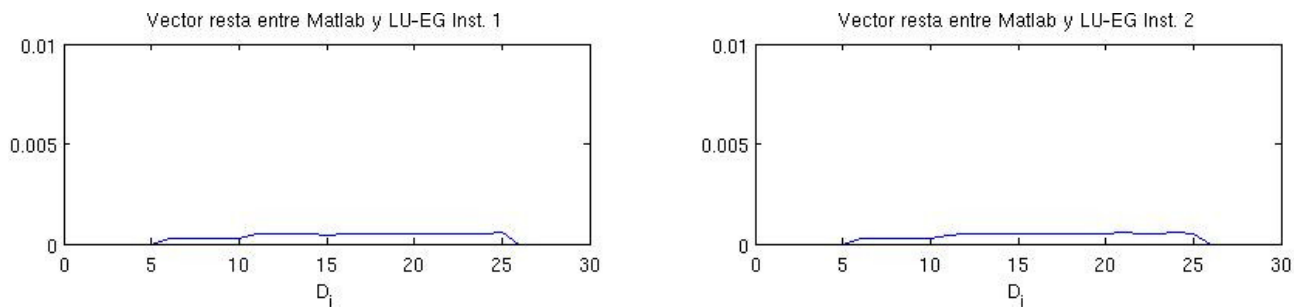
Figura 22. Comparación de los resultados contra Matlab



Para poder sacar conclusiones respecto del experimento, procedimos a restar $m1 - v1$ y a esto dividirlo componente a componente por $m1$. De la misma forma con la segunda instancia $v1$ y $m1$ para obtener dos vectores a los que llamamos Diferencia Relativa, o $d1$ y $d2$.

En la [Figura 23](#) se pueden ver los valores de estas diferencias, o observamos que estan en el orden de las milésimas.

Figura 23. Valores del vector D



3.2.6. Discusión de los resultados

Las conclusiones obtenidas en estos experimentos son que si bien en complejidad teorica los dos metodos empleados se comportan de la misma manera, es decir tienen complejidad cubica en la practica, a la hora de resolver sistemas donde se puedan presentar muchas instancias distintas a resolver en el futuro es altamente recomendable usar Factorizacion LU, por sobre E.G., ya que usar E.G. y no Fact. LU para resolucion de un gran numero de instancias, puede ser enormemente perjudicial en terminos de rendimiento.

Por otra parte, quedó demostrado que aunque nuestros algoritmos parecían perfectos, la sucesión de operaciones sobre números densos en los reales representados de manera discreta generó un error no despreciable. Sin embargo consideramos que en el contexto utilizado, el error obtenido es, aunque no despreciable, al menos aceptable.

4. Conclusiones

Algunas de las conclusiones obtenidas luego de la elaboración de este informe y de haber llevado a cabo diferentes experimentos:

- Con respecto a la relación que hay entre el tamaño de la discretización del sistema y la precisión de los resultados con respecto a la isoterma buscada, concluimos que a partir de una cierta granulación de tamaño medio y equilibrado en sus variables, por ejemplo tomando 30 ángulos y 30 radios, no se encontraron diferencias significativas en la ubicación de la isoterma, con respecto a granulaciones de orden superior, como se puede ver en los experimentos.

Además, notamos que crear discretizaciones dispares, como tomar una amplia cantidad de ángulos y pocos radios o viceversa, no llevaba a resultados satisfactorios en comparación a una distribución equilibrada de ángulos y radios, con matriz resultante de similar tamaño, y por lo tanto tiempo de cómputo equivalente.

- Con respecto a la complejidad de los métodos utilizados para resolver el sistema, se llegó a la conclusión que, como era esperado, éstos se comportan con una complejidad de orden cúbico. Sin embargo, también se pudo observar que al trabajar con un gran número de instancias, la factorización LU es ampliamente superior en términos de complejidad temporal con respecto a la eliminación Gaussiana. Por otro lado, como ya describimos en la discusión de resultados de los experimentos, se puede ver que la implementación de nuestros métodos no está libre de error. Considerando que trabajamos con aritmética finita y algoritmos poco optimizados nos encontramos con errores aceptables. Además, se puede concluir que el gran problema de la resolución de sistemas lineales no pasa por la resolución del sistema en sí, sino por convertir el sistema en uno amigable. Por lo que las líneas de investigación deberían centrarse en ello.
- Algunas de las posibles optimizaciones podrían ser las de aplicar pivoteo parcial al algoritmo de EG, y utilizar Factorización PLU para reducir estos errores de cómputo ya que como se probó en las clases teóricas, usar pivoteo total y/o parcial al posicionar números grandes sobre la diagonal de nuestra matriz, se reduce la posibilidad de dividir por números pequeños lo cual minimiza el error. Por otra parte, otra optimización realizable y que inferiría seguramente de forma positiva con respecto a la exactitud en la búsqueda de la isoterma,

sería usar métodos de interpolación más eficientes, como por ejemplo el método de interpolación de Lagrange.

5. Apendices

5.1. Enunciado

Introducción

Consideremos la sección horizontal de un horno de acero cilíndrico, como en la Figura 1. El sector A es la pared del horno, y el sector B es el horno propiamente dicho, en el cual se funde el acero a temperaturas elevadas. Tanto el borde externo como el borde interno de la pared forman círculos. Suponemos que la temperatura del acero dentro del horno (o sea, dentro de B) es constante e igual a 1500°C .

Tenemos sensores ubicados en la parte externa del horno para medir la temperatura de la pared externa del mismo, que habitualmente se encuentra entre 50°C y 200°C . El problema que debemos resolver consiste en estimar la isoterma de 500°C dentro de la pared del horno, para estimar la resistencia de la misma. Si esta isoterma está demasiado cerca de la pared externa del horno, existe peligro de que la estructura externa de la pared colapse.

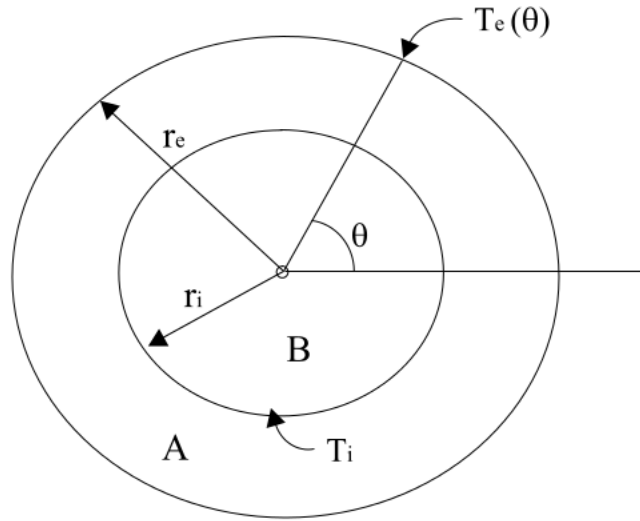


Figura 1: Sección circular del horno

El objetivo del trabajo práctico es implementar un programa que calcule la isoterma solicitada, conociendo las dimensiones del horno y las mediciones de temperatura en la pared exterior.

El Modelo

Sea $r_e \in \mathbb{R}$ el radio exterior de la pared y sea $r_i \in \mathbb{R}$ el radio interior de la pared. Llamemos $T(r, \theta)$ a la temperatura en el punto dado por las coordenadas polares (r, θ) , siendo r el radio y θ el ángulo polar de dicho punto. En el estado estacionario, esta temperatura satisface la ecuación del calor:

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

Si llamamos $T_i \in \mathbb{R}$ a la temperatura en el interior del horno (sector B) y $T_e : [0, 2\pi] \rightarrow \mathbb{R}$ a la función de temperatura en el borde exterior del horno (de modo tal que el punto (r_e, θ) tiene temperatura $T_e(\theta)$), entonces tenemos que

$$T(r, \theta) = T_i \quad \text{para todo punto } (r, \theta) \text{ con } r \leq r_i \quad (2)$$

$$T(r_e, \theta) = T_e(\theta) \quad \text{para todo punto } (r_e, \theta) \quad (3)$$

El problema en derivadas parciales dado por la primera ecuación con las condiciones de contorno presentadas recientemente, permite encontrar la función T de temperatura en el interior del horno (sector A), en función de los datos mencionados en esta sección.

Para resolver este problema computacionalmente, discretizamos el dominio del problema (el sector A) en coordenadas polares. Consideramos una partición $0 = \theta_0 < \theta_1 < \dots < \theta_n = 2\pi$ en n ángulos discretos con $\theta_k - \theta_{k-1} = \Delta\theta$ para $k = 1, \dots, n$, y una partición $r_i = r_0 < r_1 < \dots < r_m = r_e$ en $m + 1$ radios discretos con $r_j - r_{j-1} = \Delta r$ para $j = 1, \dots, m$.

El problema ahora consiste en determinar el valor de la función T en los puntos de la discretización (r_j, θ_k) que se encuentren dentro del sector A. Llamemos $t_{jk} = T(r_j, \theta_k)$ al valor (desconocido) de la función T en el punto (r_j, θ_k) .

Para encontrar estos valores, transformamos la ecuación (1) en un conjunto de ecuaciones lineales sobre las incógnitas t_{jk} , evaluando (1) en todos los puntos de la discretización que se encuentren dentro del sector A. Al hacer esta evaluación, aproximamos las derivadas parciales de T en (1) por medio de las siguientes fórmulas de diferencias finitas:

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

$$\frac{\partial T(r, \theta)}{\partial r}(r_j, \theta_k) \cong \frac{t_{j,k} - t_{j-1,k}}{\Delta r} \quad (5)$$

$$\frac{\partial^2 T(r, \theta)}{\partial \theta^2}(r_j, \theta_k) \cong \frac{t_{j,k-1} - 2t_{jk} + t_{j,k+1}}{(\Delta \theta)^2} \quad (6)$$

Es importante notar que los valores de las incógnitas son conocidos para los puntos que se encuentran sobre el borde exterior de la pared, y para los puntos que se encuentren dentro del sector B. Al realizar este procedimiento, obtenemos un sistema de ecuaciones lineales que modela el problema discretizado. La resolución de este sistema permite obtener una aproximación de los valores de la función T en los puntos de la discretización.

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.
- 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 n_{inst} 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 n_{inst} .

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.

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.

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`.

Programa y formato de archivos

Se deberán entregar los archivos fuentes que contengan la resolución del trabajo práctico. El ejecutable tomará tres parámetros por línea de comando, que serán el archivo de entrada, el archivo de salida, y el método a ejecutar (0 EG, 1 LU).

El archivo de entrada tendrá la siguiente estructura:

- La primera línea contendrá los valores r_i , r_e , $m + 1$, n , iso , $ninst$, donde iso representa el valor de la isoterma buscada y $ninst$ es la cantidad de instancias del problema a resolver para los parámetros dados.
- A continuación, el archivo contendrá $ninst$ líneas, cada una de ellas con $2n$ valores, los primeros n indicando los valores de la temperatura en la pared interna, i.e., $T_i(\theta_0), T_i(\theta_1), \dots, T_i(\theta_{n-1})$, seguidos de n valores de la temperatura en la pared externa, i.e., $T_e(\theta_0), T_e(\theta_1), \dots, T_e(\theta_{n-1})$.

El archivo de salida obligatorio tendrá el vector solución del sistema reportando una componente del mismo por línea. En caso de $ninst > 1$, los vectores serán reportados uno debajo del otro.

Junto con el presente enunciado, se adjunta una serie de scripts hechos en `python` y un conjunto instancias de test que deberán ser utilizados para la compilación y un testeo básico de la implementación. Se recomienda leer el archivo `README.txt` con el detalle sobre su utilización.

Fechas de entrega

- *Formato Electrónico*: Jueves 1 de Septiembre de 2016, hasta las 23:59 hs, enviando el trabajo (informe + código) a la dirección `metnum.lab@gmail.com`. El subject del email debe comenzar con el texto [TP1] seguido de la lista de apellidos de los integrantes del grupo.
- *Formato físico*: Viernes 2 de Septiembre de 2016, de 17:30 a 18:00 hs.

Importante: El horario es estricto. Los correos recibidos después de la hora indicada serán considerados re-entrega. Los grupos deben ser de 3 personas, sin excepción. Es indispensable que los trabajos pasen satisfactoriamente los casos de test provistos por la cátedra.

5.2. Código fuente

A continuación se incluyen los los codigos fuente de las funciones relevantes desde el punto de vista numerico.

5.2.1. Eliminación Gaussiana

```
void Matriz::EG(){
    int c = columnas();
    int f = filas();
    for (int i = 0; i < f-1; i++){
        for(int j = i+1 ; j < f; j++){
            double multiplicador = _matriz[j][i]/(double) _matriz[i][i];
            for (int k = i ; k < c ; k++){
                _matriz[j][k]=_matriz[j][k]-multiplicador*(double)
                _matriz[i][k];
            }
        }
    }
}
```

5.2.2. Factorización LU

(guardando L y U en la misma matriz)

```
void Matriz::LU(){
    int c = columnas();
    int f = filas();
    for (int i = 0; i < f-1; i++){
        for(int j = i+1 ; j < f; j++){
            double multiplicador = _matriz[j][i]/(double) _matriz[i][i];
            for (int k = i ; k < c ; k++){
                if(k != i){
                    _matriz[j][k] = _matriz[j][k] - multiplicador*(
                        double) _matriz[i][k];
                }
                else{
                    _matriz[j][k] = multiplicador;
                }
            }
        }
    }
}
```

5.2.3. Backward Substitution

```
vector<double> Matriz::resolverTS(vector<double> b){
    vector<double> res(_filas, 0);
    for(int i = _filas - 1; i >= 0; i--){
        double acc = 0;
        for(int j = i + 1; j <= _columnas - 1; j++){
            acc += _matriz[i][j] * (double) res[j];
        }
        res[i] = (b[i] - acc) / (double) _matriz[i][i];
    }
    return res;
}
```

5.2.4. Forward Substitution

```
vector<double> Matriz::resolverTI(vector<double> b){
    vector<double> res(_filas, 0);
    for(int i = 0; i < _filas; i++){
        double acc = 0;
        for(int j = 0; j < i; j++){
            acc += _matriz[i][j] * (double) res[j];
        }
        res[i] = (b[i] - acc) / (double) _matriz[i][i];
    }
    return res;
}
```

5.3. Proposición 1

Sea $A \in \mathbb{R}^{n \times n}$ la matriz generada para el sistema del alto horno, es posible aplicar Eliminación Gaussiana sin pivoteo.

5.3.1. Demostración

Inicialmente se sabe que nuestra matriz es diagonal dominante y banda. Por la composición de nuestra matriz sabemos que esta no sólo no es nula sino que ninguna fila lo es.

Además por el armado de la matriz se sabe que el coeficiente de $t_{j,k}$ es:

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

llamemos lo que multiplica a $t_{j,k}$, $I_{j,k}$, se puede ver que $I_{j,k} \neq 0$ entonces sabemos que los elementos de la diagonal son no nulos ($a_{i,i} \neq 0 \forall i$).

Además sobre $t_{j-1,k}$ y $t_{j+1,k}$ se puede inferir lo siguiente:

- $t_{j-1,k}(\frac{1}{(\Delta r)^2} - \frac{1}{r\Delta r})$, llamemos lo que multiplica a $t_{j-1,k}$, $I_{j-1,k}$. $I_{j-1,k} \neq 0$, ya que si $I_{j-1,k} = 0$ entonces $(\Delta r)^2 = r\Delta r \Rightarrow r = \Delta r$ y esto solo pasa si no granulamos el radio.
- $t_{j+1,k}(\frac{1}{(\Delta r)^2})$ llamemos lo que multiplica a $t_{j+1,k}$, $I_{j+1,k}$. Se tiene trivialmente que $I_{j-1,k} > 0$

Además se sabe que el armado de las filas de la matriz es de la forma:

$$A = \begin{pmatrix} I & \dots & \dots & \dots & 0 \\ \ddots & I_{j,k} & \dots & I_{j+1,k} & 0 \\ 0 & \ddots & \ddots & \ddots & \dots \\ 0 & 0 & \ddots & \ddots & I \end{pmatrix}$$

Veamos que luego de aplicar el primer paso de E.G. La matriz $\tilde{A} \in \mathbb{R}^{(n-1) \times (n-1)}$ sigue siendo diagonal dominante con diagonal $\widetilde{a_{i,i}} \neq 0, \forall i$ y banda superior distinta de 0.

Es trivial que puedo aplicar sobre las primeras n filas identidad y también sobre el primer $I_{j,k}$ encontrado también ya que $I_{j,k} \neq 0$. También es trivial ver que los $I_{j+1,k}$ no cambian ya que arriba de ellos solo hay ceros, por lo que los $\widetilde{I_{j+1,k}}$ de \tilde{A} siguen siendo distintos de 0, ya que $\widetilde{I_{j+1,k}} = I_{j+1,k}$.

Veamos que \tilde{A} sigue siendo diagonal dominante con $\widetilde{a_{i,i}} \neq 0$. Por la estructura del algoritmo de EG se realiza el siguiente paso:

$$F_i - \frac{a_{i,1}}{a_{1,1}} F_1 \quad \forall i = 2 \dots n \Rightarrow \widetilde{a_{i,j}} = a_{i,j} - \frac{a_{i,1}}{a_{1,1}} a_{1,j} \quad (7)$$

Sea

$$\sum_{j \neq i}^n |\widetilde{a_{i,j}}| = \sum_{j \neq i}^n \left| a_{i,j} - \frac{a_{i,1}}{a_{1,1}} a_{1,j} \right| \leq \sum_{j \neq i; 2 \leq j}^n |a_{i,j}| + \sum_{j \neq i; 2 \leq j}^n \left| \frac{a_{i,1}}{a_{1,1}} a_{1,j} \right|$$

Como $\sum_{j \neq i; 2 \leq j}^n |a_{i,j}| \leq |a_{i,i}|$, por hipótesis de diagonal dominante:

$$\sum_{j \neq i; 2 \leq j}^n |a_{i,j}| + \sum_{j \neq i; 2 \leq j}^n \left| \frac{a_{i,1}}{a_{1,1}} a_{1,j} \right| \leq |a_{i,i}| - |a_{i,1}| + \left| \frac{a_{i,1}}{a_{1,1}} \right| \sum_{j \neq i; 2 \leq j}^n |a_{1,j}|$$

Como $\sum_{j \neq i; 1 \leq j}^n |a_{1,j}| \leq |a_{1,1}|$ por hipótesis de diagonal dominante:

$$|a_{i,i}| - |a_{i,1}| + \left| \frac{a_{i,1}}{a_{1,1}} \right| \sum_{j \neq i; 2 \leq j}^n |a_{1,j}| \leq |a_{i,i}| - |a_{i,1}| + \left| \frac{a_{i,1}}{a_{1,1}} \right| (|a_{1,1}| - |a_{1,i}|) = |a_{i,i}| - \left| \frac{a_{i,1}}{a_{1,1}} \right| |a_{1,i}|$$

Que por (7)

$$\sum_{j \neq i}^n |\widetilde{a_{i,j}}| \leq |a_{i,i}| - \left| \frac{a_{i,1}}{a_{1,1}} a_{1,i} \right| = |\widetilde{a_{i,i}}| \forall i$$

Además como $I_{j+1,k}$ no es modificado por lo visto anteriormente y $I_{j+1,k}$ esta dentro del conjunto de los $\{\widetilde{a_{i,j}}\}_{j \neq i; i \neq j}$, por lo que se tiene que:

$$0 < |I_{j+1,k}| \leq \sum_{j \neq i} |\widetilde{a_{i,j}}| \leq |\widetilde{a_{i,i}}| \Rightarrow |\widetilde{a_{i,i}}| > 0$$

Entonces \tilde{A} es diagonal dominante, con $\widetilde{a_{i,i}} \neq 0$ y no altera la banda superior. Usando un razonamiento inductivo sobre \tilde{A} , se puede ver fácilmente que se le puede aplicar E.G sin pivoteo ya que al aplicar el primer paso de E.G sobre A obtengo un sistema $\tilde{A} \in \mathbb{R}^{(n-1) \times (n-1)}$ de con las misma propiedades relevantes para esta demostracion. \square

6. Referencias

<http://www.cplusplus.com/>

<http://stackoverflow.com/>

https://en.wikipedia.org/wiki/Blast_furnace

<http://linux.die.net/man/1/time>

<http://stackoverflow.com/questions/556405/what-do-real-user-and-sys-mean-in-the-output-of-time1>

Ademas en temas referidos al armado del sistema y errores de presicion en la implementacion, asi como en la propiedad a demostrar dialogamos con otro grupo que se habia topado con barreras similares a las nuestras.