

Trabajo Práctico 1

Jueves 10 de Abril de 2014

Métodos Numéricos

Discretización de Alto Horno

Integrante	LU	Correo electrónico
Esteban Fernandez	691/12	esteban.pmf@gmail.com
Pedro Rodriguez	197/12	pedrorodriguezsjs@hotmail.com
Martín Caravario	470/12	martin.caravario@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellon I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autonoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
<http://www.fcen.uba.ar>

Índice

1. Introducción	2
2. Desarrollo	2
3. Resultados	9
4. Discusión	15
5. Conclusiones	17
6. Apéndices	19
6.1. Enunciado	19
6.1.1. Introducción	19
6.1.2. El Modelo	19
6.1.3. Algoritmo para despeje de ecuaciones para matrices ya trianguladas	20
6.1.4. Enunciado	20
6.1.5. Programa y formato de archivos	21
6.1.6. <u>Fechas de entrega</u>	22
7. Referencias	23

Resumen

Un alto horno, a grandes rasgos, es una cápsula cilíndrica de acero de unos 30 metros de alto; forrada con un material no metálico y resistente al calor cuya finalidad yace en las bases de la fusión y reducción de minerales de hierro con la intención de elaborar la fundición del mismo.

Este informe explicará como es, mediante el uso de un modelo computacional, el comportamiento del calor en el interior del horno y en la pared que lo rodea y separa del exterior. Otro aspecto importante a destacar de este informe es el uso de métodos de Álgebra Lineal que ayudarán a resolver el problema en cuestión.

Al finalizar este informe se habrá puesto al tanto al lector sobre las razones del modelo planteado; se presentarán experimentaciones junto con sus respectivas reflexiones y se hablará de eficiencia en relación a la implementación y elección particular de los diferentes métodos de Álgebra Lineal para resolver el problema.

Palabras clave: Horno, isoterma, discretización, Gauss

1. Introducción

Las bases teóricas utilizadas para modelar computacionalmente el comportamiento del calor en el horno se centran principalmente en:

- Ecuación de Calor, la cual indicará la temperatura en un punto determinado de la pared.¹
- Discretización ² de la Ecuación de Calor nombrada en el item anterior para poder ser implementada en un lenguaje de programación imperativo, lo cual resultará en la posibilidad de resolverla en una computadora. Esta necesidad de una discretización del problema surge de la imposibilidad de una computadora de trabajar con dominios infinitos. Dicha imposibilidad desembocará en la no terminación efectiva del programa que modelará el problema ya que una computadora no puede calcular infinitos valores en un tiempo finito; es por esto que se fraccionará el dominio original a uno finito (valga la redundancia) al método que calculará las temperaturas. No obstante, este método acarreará errores de aproximación en los resultados efectivos en comparación con los verdaderos respecto de la realidad, por lo que, si bien el programá terminará y retornará valores aproximados, quedará a criterio de uno el tiempo que quiera pagar a cambio de un resultado mas exacto (en este punto cabe aclarar que el programa retornará valores más aproximados mientras más grande sea el dominio a considerar, pero es de esperar que el tiempo de cálculo aumente; depende mucho del problema a tratar).
- Metodo de Eliminación Gaussiana³ y Metodo de Factorización LU⁴: Se plantearán sistemas de ecuaciones lineales⁵ que responderán de forma aproximada a los valores de las temperaturas en la pared del horno en un instante determinado de tiempo. Dichos sistemas serán resueltos mediante los dos métodos de Álgebra Lineal que se nombran a comienzos de este item; estos basan su algoritmia en la multiplicación de matrices por otras particulares, conocidas como "matrices elementales", con el fin de conseguir un nuevo sistema equivalente al original que se estaba resolviendo, pero que, una vez finalizados los mismos, el/los resultante/s (dependiendo de cual de los dos se use) mantendrá/n propiedades(por ejemplo ser triangula superior en el caso de E.Gaussiana) que nos permitirán, sea cual sea la matriz original, resolver de forma générica el sistema de ecuaciones planteado (en tiempo cúbico $O(n^3)$) para el caso de la E.G. y cuadrático para Factorización LU, $O(n^2)$.⁶

2. Desarrollo

Escribir la Ecuación de Calor en terminos de un lenguaje de programación es fundamental para poder modelar computacionalmente el problema en cuestión. Esta escritura implicaría el cálculo y conocimiento de las temperaturas en cualquier punto del horno y, así, ser conscientes de la dispersión del calor dentro del mismo.

Supondremos que el horno es un cilindro que posee un radio interno (del centro a la pared interna) y radio externo (de la pared interna al exterior) determinados. Además daremos por sentado que en un horno, dado un radio y un angulo cualesquiera para calcular una temperatura en un punto, el resultado será el mismo a cualquier altura de este. Por estas razones podremos tomar un corte trasversal del horno cilíndrico, y será suficiente para resolver el problema planteado. Para cada instancia de prueba se fijarán previamente las dimensiones del Alto Horno a considerar. Dicho tamaño se verá reflejado en base al radio interno y radio externo del mismo.

A su vez, para cada instante de tiempo conoceremos la temperatura en donde, efectivamente, se funde el hierro, la misma es homogénea desde el centro del horno hasta el comienzo de la pared interna, determinada por el radio interno. En las paredes externas delimitadas por el radio externo encontraremos diferentes sensores que nos dirán la temperatura en la cara externa de la pared del Alto Horno.

Llamaremos T a la temperatura en el punto dado por las coordenadas polares (r, θ) Esta temperatura responde a la Ecuación de Calor introducida, la cual, ajustada a las dimensiones del problema, es la siguiente:

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

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

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

¹http://es.wikipedia.org/wiki/Ecuacion_de_Calor

²http://es.wikipedia.org/wiki/Metodo_de_los_elementos_finitos

³http://es.wikipedia.org/wiki/Eliminacion_de_Gauss-Jordan

⁴http://es.wikipedia.org/wiki/Factorizacion_LU

⁵http://es.wikipedia.org/wiki/Sistema_de_ecuaciones_lineales

⁶http://es.wikipedia.org/wiki/Teoria_de_la_complejidad_computacional

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

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.

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 $\theta_k = \Delta\theta$ 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 $r_j = \Delta r$ 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:

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

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

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

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.

Finalmente, con estas aclaraciones, redujimos el Alto Horno a un corte transversal de un cilindro que tiene la siguiente forma:

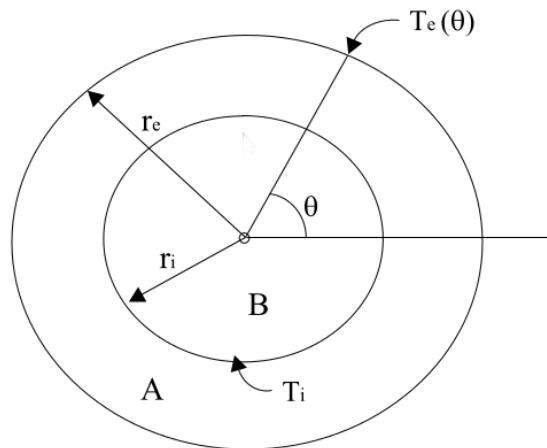


Fig 1: Corte transversal del Alto Horno: r_e hace referencia al radio externo, r_i al interno, **A** y **B** son la pared y el lugar de fundición del horno respectivamente.

A partir de esta discretización de la ecuación de calor, podremos calcular las temperaturas con un dominio finito (Es decir, la computadora calculará en tiempo finito estas), las mismas están ligadas unas con otras por lo que quedará un sistema de ecuaciones lineal del cual se obtendrá una matriz asociada. A partir de esta última podremos (y de otras maneras, que serán vistas mas adelante en este informe) experimentar y evaluar la relación costo/beneficio de los algoritmos que implementan la triangulación gaussiana y la factorización LU. La forma en la cual implementamos el

Método de Eliminación Gaussiana en C++ es la propuesta en el siguiente código:

```
vector<double> gaussElim(vector< vector<double> > A){
double m;
for(int k = 0; k < (A.size()-1); k++){
for(int i = k+1; i < A.size(); i++){ //busca el multiplicador de la fila
if(A[k][k] < 0.00001 && A[k][k] > -0.00001 ){
int toSwap = k;
double maximo = 0;
for(int l = k+1; l < A.size(); l++){
if(abs(A[l][k]) > maximo){
maximo = A[l][k];
toSwap = l;
}
}
if(maximo != 0){
for(int t = 0; t < A[0].size(); t++){
double guardol = A[k][t];
A[k][t] = A[toSwap][t];
A[toSwap][t] = guardol;
}
}
}

m = (A[i][k]/A[k][k]);
for(int j = k; j < A[0].size(); j++){ // asigna valor a cada columna de la fila
dada por i
A[i][j] = A[i][j] - m*A[k][j];
}
}
}
```

La eliminación Gaussiana esta fundamentada en la idea de triangular matrices a partir de la aplicación de las tres operaciones básicas, las cuales pueden ser aplicadas a cualquier matriz A sin que estas tengan efecto sobre sistema de ecuaciones definido por la misma. De esta manera buscamos una matriz A' que sea equivalente a A y que al mismo tiempo A' sea triangular superior. De esta manera, luego podremos aplicar un simple algoritmo que resuelva el sistema $A'x=b'$. Tal algoritmo figura en el apéndice. En cada paso de la iteración en el algoritmos de más arriba, lo que se hace es, a partir de operaciones con matrices elementales, dejar un cero en cada elemento de la columna correspondiente al elemento de la diagonal en el cual estamos parados. Por eso, como queremos llegar a tener una matriz triangular superior, recorreremos todos los elementos de la diagonal (desde $A[0][0]$ hasta $A[tam(A)-1][tam(A)-1]$). En cada paso utilizamos al elemento $A[i][i]$ para calcular el multiplicador m y aplicarlo sobre las filas $i+1..tam(A)$, dejando 0 en los elementos de la columna correspondiente al elemento $A[i][i]$ y modificando el resto de los elementos de cada fila para que nos quede una sistema $(A-b)'$ equivalente al anterior $A-b$. Es importante notar que al resolver un sistema con EG, cada vez que desee resolver más de una instancia, va a ser necesario realizar todo el proceso de EG desde el comienzo, de lo contrario, cuando nos queda la matriz final triangular superior, no vamos a haber actualizado los valores del vector b, los cuales deben ser cambiados por los de b' (y, eventualmente para cada b_i pasados como parámetro, dicha cantidad depende del "numero de instancias") de forma correcta, y al resolver el sistema vamos a obtener resultados erróneos para X.

Luego, la implementación en C++ del método de triangulación LU es:

```
void factorizacionL(vector< vector<double> >& A){
double m;
for(int k = 0; k < (A.size()-1); k++){
for(int i = k+1; i < A.size(); i++){ //busca el multiplicador de la fila
if(A[k][k] < 0.00001 && A[k][k] > -0.00001 ){
int toSwap = k;
double maximo = 0;
for(int l = k+1; l < A.size(); l++){
if(abs(A[l][k]) > maximo){
```

```

        maximo = A[l][k];
        toSwap = l;
    }
} if (maximo != 0) {
    for (int t = 0; t < A[0].size(); t++) {
        double guardol = A[k][t];
        A[k][t] = A[toSwap][t];
        A[toSwap][t] = guardol;
    }
}
}
m = (A[i][k]/A[k][k]);
for (int j = k; j < A[0].size(); j++) { // asigna valor a cada columna de la fila dada
    por i
    A[i][j] = A[i][j] - m*A[k][j];
}
A[i][k] = m;
}
}
diagonalizar(A);
triangularInf(A);
return;
}

void factorizacionU(vector< vector<double> >& A) {
double m;
for (int k = 0; k < (A.size()-1); k++) {

    for (int i = k+1; i < A.size(); i++) { //busca el multiplicador de la fila
        if (A[k][k] < 0.00001 && A[k][k] > -0.00001) {
            int toSwap = k;
            double maximo = 0;
            for (int l = k+1; l < A.size(); l++) {
                if (abs(A[l][k]) > maximo) {
                    maximo = A[l][k];
                    toSwap = l;
                }
            }
            if (maximo != 0) {
                for (int t = 0; t < A[0].size(); t++) {
                    double guardol = A[k][t];
                    A[k][t] = A[toSwap][t];
                    A[toSwap][t] = guardol;
                }
            }
        }
    }
    m = (A[i][k]/A[k][k]);

    for (int j = k; j < A[0].size(); j++) { // asigna valor a cada columna de la fila dada
        por i
        A[i][j] = A[i][j] - m*A[k][j];

    }

}
}
return;
}

```

Por otro lado, está el algoritmo que utilizamos para la construcción de dos matrices: L y U, que verifican la ecuación

$A=LU$ y además L es triangular inferior y U es triangular superior. Luego, vale que para todo vector x , vale $Ax=LUx$. Luego, $Ax=b$ sii $LUx=b$. Entonces, lo que hicimos fue hacer dos funciones distintas, una para crear la matriz L y otra para crear la matriz U a partir de A . En ambas funciones tuvimos en cuenta el caso en que en algún paso de la factorización nos quede un cero en la diagonal. Para eso, tomamos como criterio que un número es igual a cero si está entre 0 y ± 0.00001 . En el caso de `factorizacionL`, lo que la función hace es ir recorriendo iterativamente los elementos de la diagonal de A y en cada paso, primero se fija si $A[i][i]$ es o no igual a cero. Si lo es, intercambiamos esa fila por una de las filas que hay debajo de dicho elemento: la fila que tenga el máximo valor en la columna i y sino, no hace nada. Tras realizar este paso, se busca el multiplicador correspondiente al elemento que quedó en la posición $A[i][i]$ y se hace lo mismo que se hacía en cada paso de la EG excepto que al final se coloca en cada fila de la columna correspondiente a $A[i][i]$ al multiplicador hallado en ese paso. Luego de la finalización del ciclo principal, se colocan 1's en la diagonal y 0's por encima de la diagonal. Así, queda formada una matriz L triangular inferior, con sus columnas cargadas con los multiplicadores que fuimos utilizando en cada paso de la factorización. En el caso de `factorizacionU`, lo que la función hace es exactamente lo mismo que hace la EG, obteniendo una matriz U equivalente a A . Notar que con esta `factorizacionLU`, si tenemos muchos vectores b para los cuales queremos resolver el sistema $Ax=b$, entonces con realizar sólo una vez la `factorizacionLU` alcanza (vamos a poder usar siempre las mismas L y U para resolver el sistema $LUx=b$ (primero vamos a resolver $Ux=y$ y luego $Ly=b$. De esta forma, nos vamos a ahorrar mucho tiempo computacionalmente).

Como compilar los archivos fuente y ejecutar tests: Para compilar los archivos fuente del TP en Linux (Ubuntu), es necesario abrir una terminal (`ctrl + alt + t`), acceder a la carpeta donde se encuentran los archivos fuente `inputoutput.cpp` y `hornear.h` (mediante el comando `'cd ruta de la carpeta'`). Luego, teclear el siguiente comando en la consola:

```
g++ inputoutput.cpp -o ejecutable
```

Una vez hecho esto, nos aparecerá en la misma carpeta donde se encuentra `inputoutput.cpp`, un archivo con nombre `ejecutable`, el cual ejecutaremos desde la línea de comandos para realizar los tests de la siguiente manera (este programa toma 3 parámetros, los cuales se separan por un espacio cuando se los escribe en la línea de comandos, el primero es un archivo con información sobre la entrada, el segundo es donde se guardarán los resultados, y el tercero, Option, es el número que selecciona el método para resolver los sistemas lineales (0 para EG, 1 para F.LU):

```
./ejecutable 'ruta de acceso del archivo .in,' 'ruta de acceso del archivo .out ' Option
```

Una aclaración importante es que, si se desea observar donde figura la isoterma buscada, es necesario cambiar la ruta donde se creará el archivo `isoth.out`, lo cual puede hacerse desde el archivo fuente `inputoutput.cpp`, línea 107. Una vez hecho esto, será necesario volver a compilar como se explicó anteriormente y luego ejecutar nuevamente con los parámetros deseados.

- Hipótesis:

Hipótesis del experimento 1 :

Para este experimento consideramos una temperatura externa de 0°C , un radio interno de 10 metros, un radio externo de 100 metros, 30 radios, 30 ángulos, una única instancia y pedimos la isoterma de 500° .

Al ser un horno grande creemos que se observará una mayor variedad de temperaturas en el gráfico de calor del horno y que los cambios de temperatura no serán tan bruscos. Con respecto a la ubicación de la isoterma consideramos que esta se ubicará más cerca de la pared interna ya que al ser un horno grande habrá un radio externo mayor.

Hipótesis del experimento 2 :

Para este experimento consideramos una temperatura externa de 0°C , un radio interno de 10 metros, un radio externo de 50 metros, 30 radios, 30 ángulos, una única instancia y pedimos la isoterma de 500°C .

Al ser un horno más chico con respecto al utilizado en el experimento 1 creemos que se observarán cambios más bruscos de temperaturas, debido a que la distancia entre el radio externo y el interno es menor al anterior. Con respecto a la isoterma suponemos que esta se encontrará más cerca de la pared interna.

Hipótesis del experimento 3 :

Para este experimento consideramos una temperatura externa de 0°C , un radio interno de 10 metros, un radio externo de 100 metros, 10 radios, 60 ángulos, una única instancia y pedimos la isoterma de 500°C .

Creemos que al realizar la discretización con un número de ángulos mayor al número de radios, obtendremos una mayor precisión del posicionamiento de la isoterma con respecto al centro del horno. Pero consideramos que al tener un número inferior de radios, perderemos precisión en la medida de la distancia de la isoterma con respecto a la pared interna y externa.

Hipótesis del experimento 4 :

Para este experimento consideramos una temperatura externa de 0°C , un radio interno de 10 metros, un radio externo de 100 metros, 60 radios, 10 ángulos, una única instancia y pedimos la isoterma de 500°C .

Al tener una mayor cantidad de radios que de ángulos, consideramos que la posición de la isoterma será más precisa con respecto a las paredes del horno pero que perderemos precisión en cuanto a la forma real que se espera tenga la isoterma.

Hipótesis del experimento 5 :

Para este experimento consideraremos varias instancias de experimentación. Para esto iremos variando el parámetro `ninst`, y dejaremos fijo, un radio interno de 50 metros, un radio externo de 100 metros, 20 radios, 20 ángulos y pedimos la isoterma de 60°C . Luego iremos variando las temperaturas externas según los `ninst` propuestos como se indica en el siguiente gráfico.

ninst	Temperaturas externas utilizadas
1	0°C
2	$0^{\circ}\text{C}, 100^{\circ}\text{C}$
3	$0^{\circ}\text{C}, 100^{\circ}\text{C}, 150^{\circ}\text{C}$
5	$0^{\circ}\text{C}, 100^{\circ}\text{C}, 150^{\circ}\text{C}, 200^{\circ}\text{C}, 250^{\circ}\text{C}$

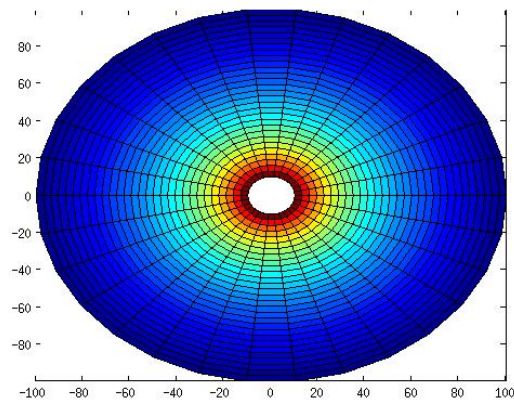
El objetivo de este experimento es descubrir que sucede al tener el mismo horno, pero con distintos números de instancias.

Nuestra hipótesis se basa en que al tener un mayor número de instancias, el método de la factorización LU es el más eficiente y veloz a la hora de resolverlo, ya que al tener precalculada la

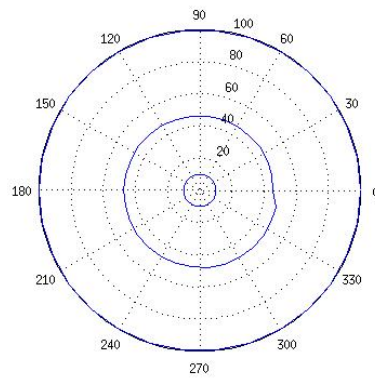
descomposición de la matriz A en L^*U , solo queda resolver los sistemas con dos matrices triangulares, mientras que con la eliminación gaussiana, debo triangular la matriz A , n^2 veces.

3. Resultados

Resultados del experimento 1 :

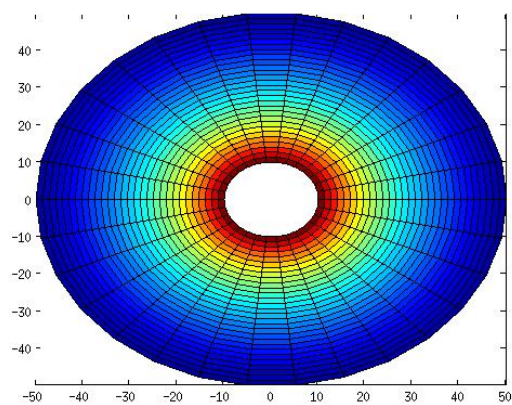


Calor del horno del experimento 1

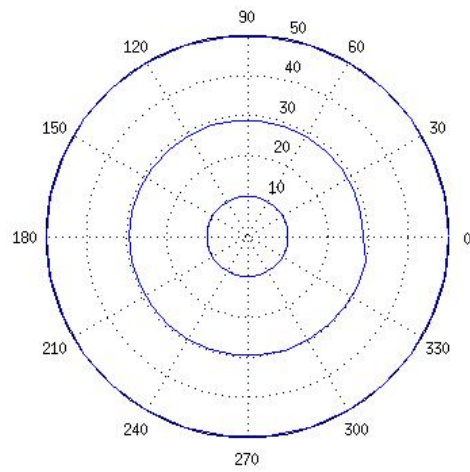


Isoterma del horno del experimento 1

Resultados del experimento 2 :

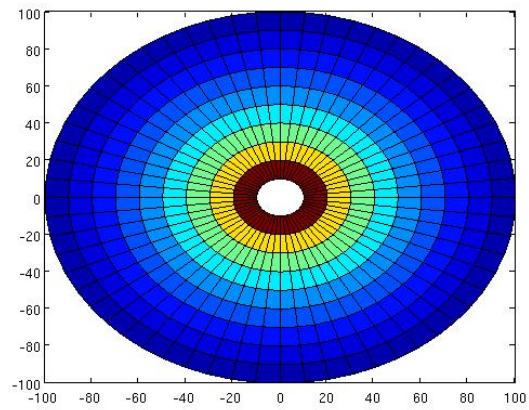


Calor del horno del experimento 2

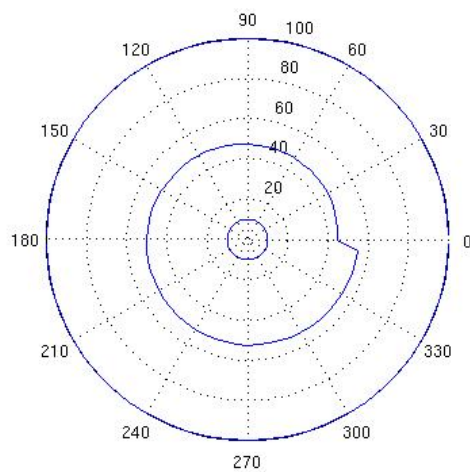


Isoterma del horno del experimento 2

Resultados del experimento 3 :

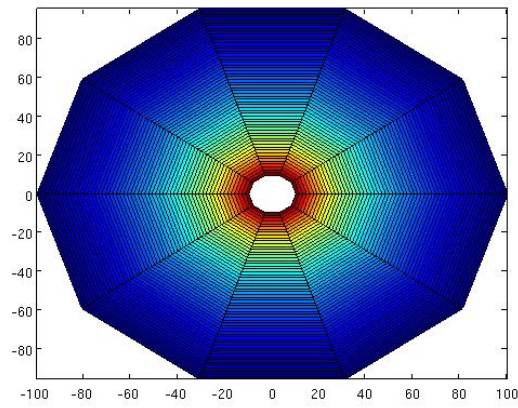


Calor del horno del experimento 3

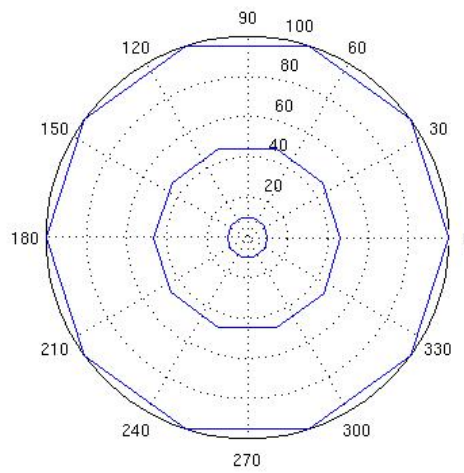


Isoterma del horno del experimento 3

Resultados del experimento 4 :



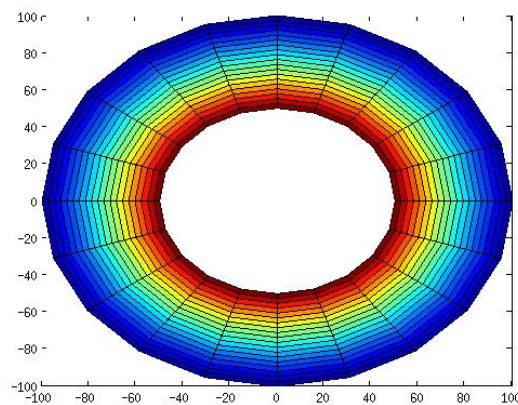
Calor del horno del experimento 4



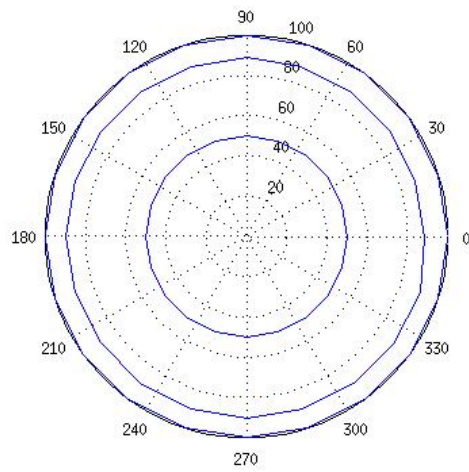
Isoterma del horno del experimento 4

Resultados del experimento 5 :

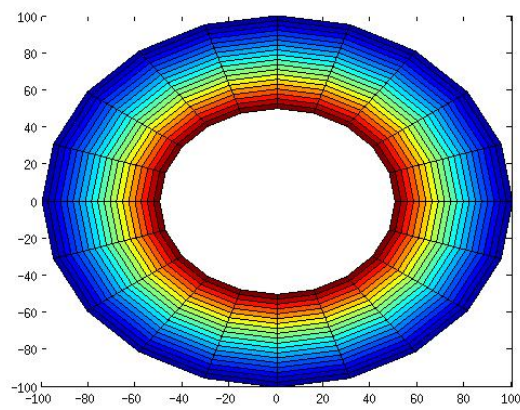
ninst	Eliminación Gaussiana	Factorización LU
1	0.3716 seg	0.7483 seg
2	0.7432 seg	0.7441 seg
3	1.1077 seg	0.7444 seg
5	1.8451 seg	0.7507 seg



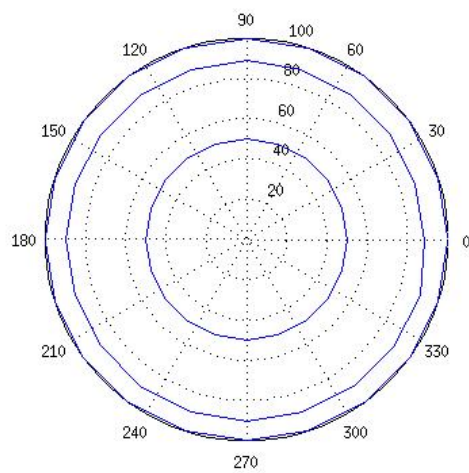
Calor del horno del experimento 5, con ninst=1



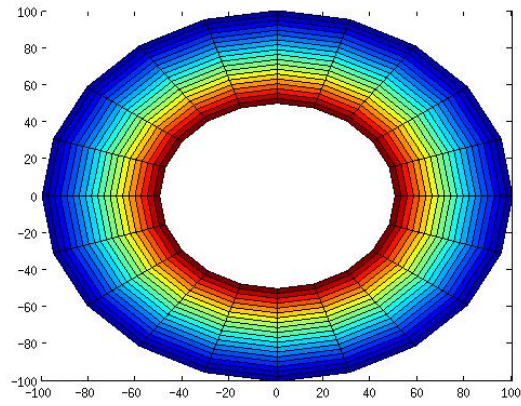
Isoterma del horno del experimento 5, con $n_{inst}=1$



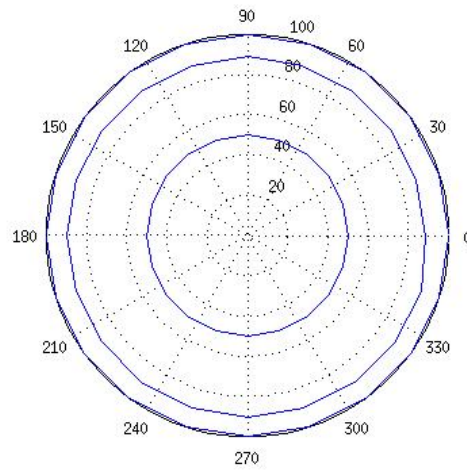
Calor del horno del experimento 5, con $n_{inst}=2$



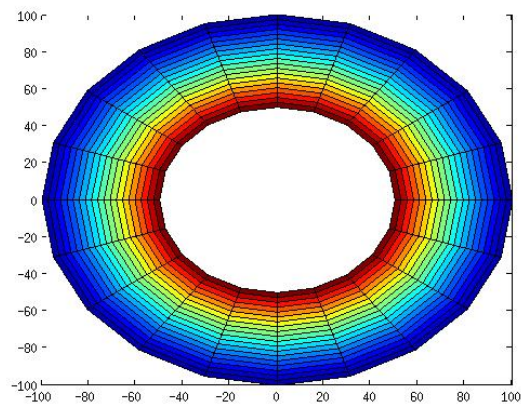
Isoterma del horno del experimento 5, con $n_{inst}=2$



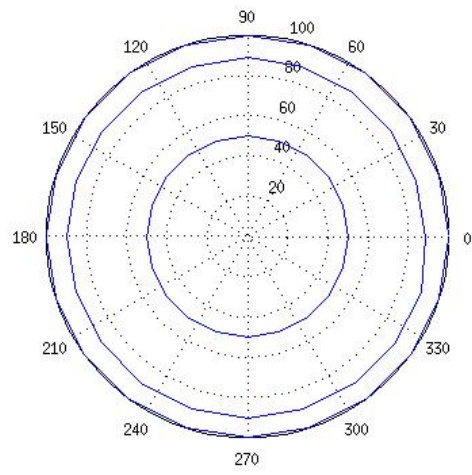
Calor del horno del experimento 5, con $n_{inst}=3$



Isoterma del horno del experimento 5, con $n_{inst}=3$



Calor del horno del experimento 5, con $n_{inst}=5$



Isoterma del horno del experimento 5, con $n_{inst}=5$

4. Discusión

Sobre los experimentos 1 y 2 Como se observa en el gráfico 1, hay una gran variedad de temperaturas dentro del horno y los cambios entre ellas no son bruscos. Esto lo vemos reflejado en la diferencia de colores y la cantidad de matices dentro del horno.

En el gráfico 2, se puede ver una isoterma a casi 40 metros de la pared interna del horno, y casi a 60 metros de la pared externa. Con respecto a los tiempos obtenidos, pudimos notar que al utilizar el método de la eliminación gaussiana el algoritmo tardó 4.12 segundos, mientras que el método de la factorización LU lo hizo en 8.34 segundos. Por otro lado en el experimento 2, como se puede ver en los resultados obtenidos, la isoterma se encuentra a 30 metros de la pared interna y a 20 metros de la externa. La isoterma observada es casi uniforme con pocas irregularidades en su contorno. Se observan menos variedades de temperatura con respecto a la experimentación anterior. Con respecto a los tiempos obtenidos, se obtuvo que al utilizar el método de la eliminación gaussiana, el tiempo cronometrados fue de 4.11 segundos, mientras que el de la factorización LU tardó 8.19 segundos.

Sobre los experimentos 3 y 4 :

En el caso del experimento 3 se puede ver en el gráfico correspondiente que se obtuvo una isoterma con forma en espiral: para los ángulos iniciales se obtuvieron valores iniciales de alrededor de 45 grados, y a medida que aumentan los valores de los ángulos, la isoterma toma valores cada vez mayores hasta casi alcanzar los 60 grados. Con respecto a los tiempos obtenidos, fueron de 1.43 sg y 2.87 sg para la eliminación gaussiana y factorización LU respectivamente.

Contrariamente a lo obtenido en el experimento 3, en el caso del experimento 4 obtuvimos una isoterma de forma decagonal. Para todos los ángulos estudiados se obtuvo una isoterma de valor similar: alrededor de los 45 grados. No se puede discernir a la perfección la forma que debería tener la isoterma (es decir, no se puede asegurar a simple vista que entre el ángulo 0 y el ángulo 30 la línea isotérmica tenga forma cóncava o convexa en lugar de lineal). Esto es debido a la baja cantidad de ángulos que consideramos para la discretización.

Claramente, pudimos observar una gran diferencia entre los gráficos obtenidos para la isoterma en los experimentos 3 y 4. Posibles razones por las cuales los gráficos dieron tan diferentes son: en el experimento 3, al tomar tantos ángulos, se logró representar de forma fidedigna la forma geométrica de la isoterma, pues al construir la matriz A de nuestro modelo, quedaba que cada temperatura puntual del interior del horno dependía de dos puntos de temperatura que estaban muy alejados de este punto (uno que estaba entre el punto y la pared externa del horno y otro que estaba entre el punto y la pared interna del horno) y de dos puntos de temperatura que estaban muy cerca de este punto (que estaban a ± 6 grados del punto en cuestión, respecto del centro del horno). Es decir, se ganó precisión al considerar puntos muy cercanos unos a otros en la discretización y esto seguramente fue porque al tomar puntos más cercanos en la discretización, nos acercamos más a los valores reales que nos daba la fórmula matemática brindada por la cátedra que sabemos cumplían todos los puntos internos del horno (nos acercábamos más a la derivada parcial del ángulo y menos a la derivada parcial del radio, porque estábamos considerando ángulos muy cercanos unos a otros y radios muy lejanos unos a otros). En el caso del experimento 4 sucedió lo mismo, solo que al revés: lo que optimizamos fue la precisión en la posición radial de los puntos de temperatura respecto del centro del horno y no tanto la precisión en la posición angular respecto del mismo.

Sobre el Experimento 5 :

Podemos ver como los gráficos son bastante similares entre sí, y como las isotermas se encuentran muy pegadas a la pared externa. Esto se debe a que las temperaturas externas en los n_{inst} , eran bastante cercanas a las de la isoterma pedida, generando así una isoterma circular uniforme muy cercana a la pared. Con respecto a los tiempos se observa que a partir del $n_{inst} = 3$, la factorización LU comienza a ser más rápida con respecto a la eliminación gaussiana. Tal como se había supuesto en la hipótesis planteada, se puede observar en el cuadro de doble entrada como a medida que voy aumentando el número de instancias, el tiempo que tarda la factorización LU en resolver los sistemas es menor al que tarda la eliminación gaussiana.

Proposición: Sea $A \in \mathbb{R}^{n \times n}$ la matriz obtenida para realizar las mediciones de temperatura del horno, probar que es posible aplicar Eliminación Gaussiana sin pivoteo.

Demostración:

Vamos a demostrar la siguiente propiedad:

Sea $A \in \mathbb{R}^{n \times n}$ una matriz diagonal dominante (no estricta), entonces al aplicar eliminación gaussiana sin pivoteo, la propiedad se mantiene.

Demostración :

Primero que nada, mirando los algoritmos que utilizamos para hacer la triangulación gaussiana se puede ver que en el paso k vale que

i) $|a_{i,j}^{(k)}| = |a_{i,j}^{(k-1)} - \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}} \cdot a_{k,j}^{(k-1)}|$ Además, por construcción podemos asegurar que la matriz A que utilizamos en nuestro modelo es diagonal dominante. Eso se puede demostrar para cualquier discretización de ángulos y radios, haciendo las cuentas. Entonces, quiero ver que:

$$\sum_{i=2i \neq j}^n |a_{i,j}^{(1)}| \leq |a_{j,j}^{(1)}| \text{ para } j = 2..n, \text{ si } |a_{j,j}| \geq \sum_{i=1i \neq j}^n |a_{i,j}| \text{ para } j = 1..n$$

Observación: Como no podemos asegurar que la matriz A sea estrictamente diagonal dominante, podría suceder que el elemento $a_{j,j}$ sea igual a cero para algún valor de $j=1..n$ y todos los $a_{i,j}$ para $i=1..n$ sean también cero. Pero si eso sucediera, entonces la matriz A tendría al menos una columna de ceros. Pero entonces, no sería verdad que todas las columnas de la matriz A son LI. Sin embargo, sabemos, por hipótesis, que todas las columnas de A son LI. Luego, esto último no puede suceder. Usando lo visto en i), tenemos que:

$$|a_{j,j}^{(1)}| \leq |a_{j,j}^{(0)} - \frac{a_{1,j}^{(0)}}{a_{1,1}^{(0)}} \cdot a_{1,j}^{(0)}|$$

Luego, si hago ese reemplazo y distribuyo la sumatoria, vale:

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

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

Además, vale que

$$\sum_{i=2i \neq j}^n |a_{i,j}^{(0)}| \leq |a_{1,1}^{(0)}| \rightarrow \sum_{i=2i \neq j}^n |a_{i,j}^{(0)}| + |a_{j,1}^{(0)}| \leq |a_{1,1}^{(0)}| \rightarrow \sum_{i=2i \neq j}^n |a_{i,j}^{(0)}| \leq |a_{1,1}^{(0)}| - |a_{j,1}^{(0)}|$$

Luego:

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

Y por desigualdad triangular y definición de $a_{i,j}$, vale

$$\leq |a_{j,j}^{(0)} - \frac{a_{1,j}^{(0)}}{a_{1,1}^{(0)}} \cdot a_{j,1}^{(0)}| = |a_{j,j}^{(1)}|$$

Que es lo que queríamos probar

Caso general :

En el caso anterior, demostramos que dada una matriz cuadrada A que verifica ser estrictamente diagonal dominante a la cual se le aplica un paso de la eliminación gaussiana vista en clase da como resultado una nueva matriz A' que también es estrictamente diagonal dominante. Esto quiere decir que para cada $j=2..n$ vale que $\sum_{i=2i \neq j}^n |a'_{i,j}| \leq |a'_{j,j}|$. Pero entonces, es claro que la submatriz $A'[2..n][2..n]$ también es estrictamente diagonal dominante. Es decir, va a valer $\sum_{i=3i \neq j}^n |a''_{i,j}| \leq |a''_{j,j}|$ para $j = 3..n$. Lo cual quiere decir que: i) A'' no tiene ceros en la diagonal y ii) podemos aplicar la misma demostración que hicimos en el caso base a la submatriz $A'[2..n][2..n]$, para ver que al aplicar un nuevo paso de la EG a A' , la submatriz $A''[3..n][3..n]$ también es estrictamente diagonal dominante. Entonces, para cualquier matriz cuadrada y estrictamente diagonal dominante va a valer que tras el paso k de la EG, $A(k)[k+1..n][k+1..n]$ es estrictamente diagonal dominante. Con lo cual $A(k)[k+1][k+1] \neq 0$. Luego, se puede ver que cada paso de la EG se puede realizar sin hacer pivoteo.

5. Conclusiones

Conclusión de los experimentos 1 y 2 :

Como conclusión general de estos dos experimentos, pudimos notar que al variar el tamaño del horno los resultados cambiaban. En el experimento 1, el horno era el doble del tamaño con respecto al del experimento 2. Esto afectó a la ubicación de la isoterma, haciendo que esta se encuentre mas cerca a la pared interna en el caso del horno grande, y mas cerca de la pared externa en el horno chico. Con este resultado podemos concluir que el radio externo influye en la ubicación de la isoterma a calcular, ya que al aumentarlo o disminuirlo vemos como varía la ubicación de esta con respecto a las paredes del horno.

Otra conclusión obtenida de esta experimentación fue con respecto a la variedad de temperaturas dentro del horno. En el horno grande pudimos notar una mayor variedad de temperaturas con cambios graduales a medida que nos acercamos al centro, mientras que en el horno chico los cambios eran mas bruscos y habia menos gamas de colores dentro del horno. Esto se debe a que al haber menor diferencia entre los radios externos e internos el calor no se distribuirá uniformemente en el horno y generará cambios mas bruscos. Finalmente podemos concluir que los tiempos de ambos algoritmos son menores para el horno chico que para el horno grande, y en particular que el método de la eliminación gaussiana es casi el doble de rápido que el de la factorización LU.

Conclusiones de los experimentos 3 y 4 :

De los resultados obtenidos para estos dos experimentos, se puede concluir que al discretizar con una cantidad relativa mayor de ángulos que de radios, se logra obtener resultados que nos hacen dar una buena idea de la forma geométrica que tiene la verdadera isoterma, pero que no es del todo efectiva a la hora de decirnos a que distancia de las paredes del horno se encuentra la isoterma (esto se ve claramente en el salto que pega la linea isotérmica entre el primero y el último de los ángulos considerados). Luego, si tuvieramos que elegir alguna de las dos discretizaciones, la decisión dependería de que es lo que mas nos importa saber sobre la isoterma que obtendremos: ¿nos importa más la distancia de la isoterma a las paredes interna y/o externa del horno o nos interesa la forma geométrica de la isoterma? Si nos importan las dos cosas, una posible solución sería aumentar la cantidad de radios Y ángulos utilizados y otra sería considerar ambos casos e intentar sacar conclusiones acerca de la isoterma real a partir de los mismos, si es que el tiempo computacional de la primera opcion es muy elevado.

Conclusión del experimento 5 Como conclusión final de este experimento, podemos decir que la factorización LU es el método mas conveniente a utilizar, siempre y cuando las instancias a resolver sean mayores o iguales a 3. Este resultado concuerda con la hipótesis que planteamos, pero es aún mas preciso, ya que sabemos a partir de que número de instancia nos conviene utilizar cada método.

Conclusion Final del Informe Para finalizar este informe consideramos importante notar que hay una clara relación entre los resultados esperados, la exactitud de los mismos en base a las medidas tomadas para la precisión y el tiempo de cómputo necesario para resolver los sistemas de ecuaciones para, con ellos, determinar las isotermas buscadas.

Algo que llama mucho la atención es que, dependiendo de los radios interno y externo del horno a considerar, aún calculando una discretización de la Ecuación de Calor, redujendo el dominio infinito de la función a numerables (y muy pocos en comparación) puntos, conseguimos resultados que se aproximan mucho a la realidad. Esto se debe a que, mientras mas puntos conformen el dominio, llega un momento en que la diferencia de los resultados experimentales se ve plasmada en cantidad de decimales que, cabe aclarar, dependiendo del problema en cuestion, pueden llegar a ser irrelevantes para la toma de decisiones en función de los mismos. Esto no quiere decir que podamos resolver cualquier problema con discretizaciones pobres, ya que podría suceder que nos importe mucho un decimal particular que requiera máxima precisión, lo cual obliga al científico a incrementar los puntos que conformarán dicha preimagen de la función asociada al modelo; esto, a por consiguiente, desembocarían en tiempos de cómputo mas elevados.

Para esta experimentación, en particular, hay que tener en cuenta que para obtener resultados aceptables es necesario verificar por lo menos dos decimales de precisión exacta (los cuales podrían

modificar relevantamente los resultados que nos influenciarán para determinar una decisión). Durante la experimentación se logró ver que la aproximación de los resultados y la isoterma está fuertemente ligado a la forma de representar los numeros involucrados(cantidad de decimales, se utilizó "Double-precision floating-point"). También el "delta"(rango entre -0.00001 y 0.00001) utilizado para realizar la comparación de números "doublez determinar únicamente mediante el mayor ($>$) y el menor ($<$) y poder determinar si dos numeros en esta precisión son iguales sin utilizar el operador "==" de C++. Algo que pudimos ver es que es muy importante en el cálculo de la isoterma, la forma en que se determina el radio donde pertenece la misma. Este puede llegar a variar relevantemente si se toman medidas poco rigurosas para su cálculo (en nuestro caso utilizamos el promedio entre los radios entre los cuales, efectivamente, se encontraría la isoterma buscada) lo cual al aumentar o reducir los angulos y radios puede llegar a generar problemas de precisión a nivel decimal. En otros experimentos que requieran mayor precisión a dicho nivel puede llegar a ser un problema grave.

6. Apéndices

6.1. Enunciado

6.1.1. 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.

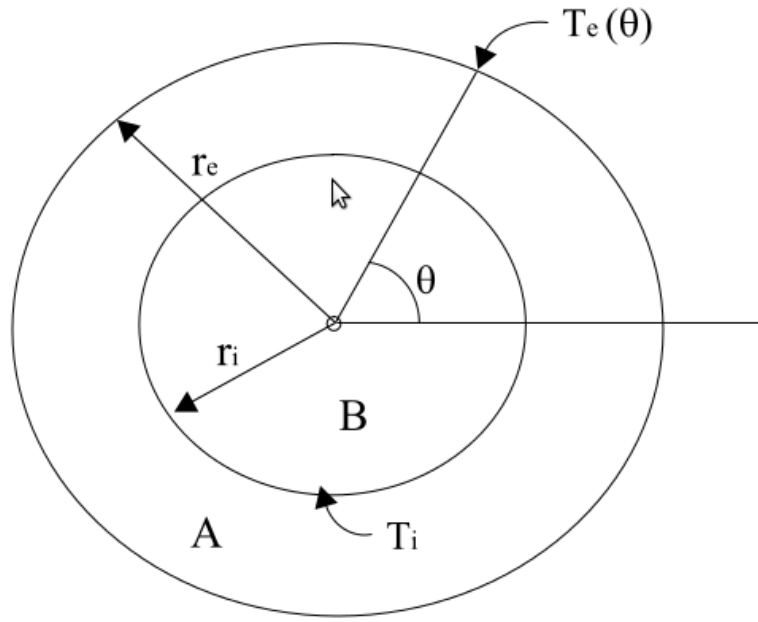


Figura 1: Sección circular del horno

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

6.1.2. 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.

6.1.3. Algoritmo para despeje de ecuaciones para matrices ya trianguladas

En esta sección presentamos el algoritmo utilizado para resolver el sistema $Tx=b$, donde T es una matriz triangular superior. Lo que hace el algoritmo es ir guardandose en cada iteración el valor de los últimos i valores de x_j que resuelven el sistema $T[i..tam(T)].x[i..tam(T)]=b[i..tam(T)]$. Como se ve, si nos movemos desde $i=tam(T)-1$ hasta $i=0$, entonces al finalizar el ciclo, vamos a tener en $X=x[1], \dots, x[tam(T)]$ la solución al sistema $Tx=b$, que va a ser igual a la solución al sistema planteado por la matriz original A : $Ax=b$.

```
for(int i = A.size() - 1; i > -1; i--){
    m1 = (1/A[i][i]);
    double suma = 0;
    for(int j = i+1; j < A[0].size(); j++){
        A[i][j] = ((A[i][j])*m1);
    }
    for(int k = i+1; k < A[0].size() - 1; k++){
        suma = suma + (A[i][k]*solution[k]);
    }
    solution[i] = (A[i][(A[0].size() - 1)] - suma);
    return solution;
}
```

6.1.4. 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 sus filas son linealmente independientes.

Luego, se pide demostrar el siguiente resultado e incluirlo en el informe:

Proposición:

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.⁷

En función de la experimentación, se pide como mínimo realizar los siguientes experimentos:

- 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. Analizar también el tiempo de cómputo requerido en función de la granularidad de la discretización.
- Considerar un escenario similar al propuesto, 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 secuencias 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*.

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`.

6.1.5. 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., $T_e(\theta_0), T_e(\theta_1), \dots, T_e(\theta_{n-1})$.

⁷Sugerencia: Notar que la matriz es diagonal dominante (no estrictamente) y analizar que sucede al aplicar un paso de Eliminación Gaussiana con los elementos de una fila.

El archivo de salida obligatorio tendrá el vector solución del sistema reportando una componente del mismo por línea. En caso de $n_{inst} > 1$, 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.

6.1.6. Fechas de entrega

- *Formato Electrónico:* Jueves 10 de Abril de 2014, 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 11 de Abril de 2014, 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 exactamente 3 personas. Es indispensable que los trabajos pasen satisfactoriamente los casos de test provistos por la cátedra.

7. Referencias

Hubo comunicación y debate sobre la resolución del problema planteado y acerca de la confección del informe presente con el grupo conformado por Erika Kodelia, Martín Mongi y Pedro Tamboridenguy. Páginas web consultadas para la realización del trabajo práctico:

<http://es.wikipedia.org/>