



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Métodos Numéricos

Trabajo Práctico 1

No creo que a él le gustará eso

Resumen

En este trabajo estudiaremos algoritmos para resolver sistemas de ecuaciones para problemas reales mediante una discretización. Se expondrán técnicas para obtener la temperatura en un punto crítico de un parabrisa usando eliminación Gaussiana y Factorización LU, aprovechando las matrices Banda, y la formula Sherman-Morrison Al final del trabajo, se llegarán a conclusiones sobre lo descubierto.

Integrante	LU	Correo electrónico
Armagno, Julián	377/12	julian.armagno@gmail.com
Balbachan, Alexis		
More, Ángel		
Pinzón, Germán	475/13	pinzon.german.94@gmail.com

Palabras claves:

Eliminación Gaussiana. Factorización LU. Punto Crítico. Matrices Banda. Sistemas de ecuaciones. Sherman-Morrison.

Índice

1. Introducción Teórica	4
2. Introducción:	4
3. Desarrollo	5
3.1. Planteo del Problema	5
3.2. Nuestro sistema de ecuaciones.	5
3.3. ¿Por qué una matriz Banda?	6
3.4. Estructura de la matriz bandas	7
3.5. Eliminación Gaussiana para matriz Banda	8
3.6. BackWard Substitution para matriz Banda	9
3.7. Matriz LU para matriz Banda	9
3.8. Experimento 1	10
3.8.1. Hipótesis	10
3.9. Experimento 2	11
3.9.1. Eliminación Gaussiana vs Factorización LU	11
3.9.2. Experimento: calidad/tiempo de cómputo	11
3.10. Cálculo de la temperatura en el punto crítico	12
3.11. Experimento 3	13
3.11.1. Eliminar sanguijuela	17
4. Resultados	18
4.1. Experimento 1	18
4.1.1. Primer caso	18
4.1.2. Segundo caso	19
4.1.3. Tercer caso	21
4.1.4. Cuarto caso	23
4.2. Experimento 2	24
4.2.1. Experimento relación tiempo-calidad de cómputo	24
4.3. Experimento 3	28
4.3.1. Experimento para Hipótesis 1	28
4.3.2. Experimento para Hipótesis 2	30
5. Discusión	32
5.1. Experimento 1	32
5.2. Experimento 2	33
5.2.1. Hipótesis planteadas sobre calidad/tiempo de cómputo	33
5.3. Experimento 3	34
6. Conclusiones	35

7. Apéndices	37
7.1. A	37
7.2. B	40

1. Introducción Teórica

2. Introducción:

Se plantea el problema de que el parabrisas de una nave está siendo atacado por sanguijuelas mutantes. Dicho ataque consiste en aplicar altas temperaturas constantes sobre la superficie del mismo, con el objetivo de lograr romperlo, para poder lograr un ataque más mortífero. La superficie del parabrisas donde se aplica el calor es circular (la sopapa de ataque es circular).

Para defenderse de estos ataques Guybrush cuenta con un sistema de refrigeración que aplica temperaturas de -100°C a los bordes del parabrisas. El parabrisas se romperá si alcanza una temperatura mayor o igual a los 235°C en el punto central (llamaremos a este punto, *punto crítico*). Si el sistema de refrigeración no es suficiente para salvar el parabrisas, se puede utilizar un arma para destruir una sanguijuela que de ser posible salve el parabrisas y reduzca lo mayor posible la temperatura en el punto crítico (llamaremos a dicha sanguijuela *la mejor*). En principio deberemos representar al parabrisas, el cual está compuesto por infinitos puntos. Sabemos que no es posible representar en una computadora los infinitos puntos del mismo, por lo que se utilizará cierta discretización. Si bien la temperatura en los puntos en el estado estacionario satisface la siguiente ecuación.

$$\frac{\delta^2 T(x, y)}{\delta x^2} + \frac{\delta^2 T(x, y)}{\delta y^2} = 0 \quad (1)$$

Al discretizar los puntos, la ecuación anterior para obtener temperaturas en cada punto del parabrisas continuo, se transforma en la siguiente ecuación por diferencias finitas.

$$t_{ij} = \frac{t_{i-1,j} + t_{i+1,j} + t_{i,j-1} + t_{i,j+1}}{4} \quad (2)$$

Es decir que en el parabrisas discretizado, la temperatura en cada punto se calcula como el promedio de la temperatura de los puntos vecinos (los puntos que estan arriba, abajo, izquierda y derecha).

Esto planteara nuevos problemas que seran analizados por ejemplo como manejar la aritmética finita de las computadoras para obtener las temperaturas mas precisas. O, si es que influye, como lo hace el utilizar distintas granularidades.

Como ademas debemos ser capaces de dar la temperatura de todos los puntos de la discretización. Para hacerlo, plantearemos el parabrisas discretizado como un sistema de ecuaciones lineales. Haciendo uso de la características que va a tener el mismo, ser un sistema Bandas. Y para poder resolver el mismo, se utilizaran tanto la técnica de Eliminación Gaussian, como la factorización LU (siempre tratando de aprovechar las características del sistema).

Como otro de los problemas es salvar el parabrisas (si es posible) eliminando *la mejor* sanguijuela. Se probarara eliminando una por una, recalculando todo el sistema y obteniendo las nuevas temperaturas en el punto crítico. Pero, como computacionalmente puede resultar costoso rearmar todo el sistema. Experimentaremos calcular, las nuevas temperaturas, aplicando la fórmula de Sherman–Morrison cuando sea posible. Evaluando si mediante esta técnica obtenemos un mejor rendimiento temporal que cuando no se la aplica: (*)

$$(A + uv^t)^{-1} = A^{-1} - \frac{A^{-1}uv^t A^{-1}}{1 + v^t A^{-1}u}. \quad (3)$$

Siendo A una matriz inversible, y u, v^t dos vectores.

3. Desarrollo

3.1. Planteo del Problema

Como dijimos anteriormente en nuestro problema teníamos un parabrisas con infinitos puntos, por ser una superficie continua, una forma de pensar el problema es discretizar estos puntos, y trabajar sobre ello. Una vez hecha esta operación, podemos modelar los puntos resultantes con una matriz, donde cada posición de esta matriz represente un punto en el parabrisas. Para simplificar como los bordes del parabrisas tienen una temperatura constante de -100. Estos puntos no se representan en el parabrisas discretizado. Por lo que tampoco en la matriz asociada. Al discretizar los puntos, para obtener las temperaturas en cada punto restante del parabrisas continuo utilizaremos la formula:

$$t_{ij} = \frac{t_{i-1,j} + t_{i+1,j} + t_{i,j-1} + t_{i,j+1}}{4}$$

En nuestro problema nos interesa conocer el punto crítico (el centro del parabrisas), esto además significa conocer la temperatura de sus vecinos, y a su vez los vecinos necesitaran conocer la temperatura de sus otros vecinos. Es decir que en un principio es necesario calcular la temperatura de todos los puntos del parabrisas discretizado. Este problema es modelado mediante un sistema de ecuaciones, en el cual cada ecuación se corresponde con un solo punto del parabrisas. Dicho sistema de ecuaciones lo representamos en una matriz cuyo tamaño es $\#puntos \times \#puntos$. Finalmente podemos calcular la temperatura en cada punto aplicando *eliminación gaussiana* sobre la matriz.

Una vez que sabemos qué temperatura hay en el punto crítico, podremos decidir que criterio utilizar para matar sanguijuelas, mediante distintas experimentaciones.

3.2. Nuestro sistema de ecuaciones.

Con el objetivo de calcular la temperatura de cada punto de nuestra matriz de $n \times m$, la cual es una representación discreta del parabrisas de largo b , ancho a y granularidad h . En nuestro problema inicial, planteamos un sistema de ecuaciones compuesto por $n * m$ incógnitas (una por cada punto con $1 \leq i \leq (a/h - 1)$, $1 \leq m \leq (b/h - 1)$). Con este motivo, numeramos los puntos (i, j) del parabrisas mediante la función:

$$F(i, j) = i * m + j, \quad F(i, j) \in [1, n * m] \quad (4)$$

Además, como se mencionó anteriormente, para cada punto $\alpha_{i,j}$ en el parabrisas se le corresponde una de las siguientes ecuaciones:

- $T_{\alpha} = -100$ si: $i = 0 \vee i = a/h \vee j = b/h \vee j = b/h$
- $T_{\alpha} = T_{Sanguijuela}$ si: $(i, j) \in Sanguijuela$
- $4T_{\alpha} - T_{\alpha+1} - T_{\alpha-1} - T_{\alpha+m} - T_{\alpha-m} = 0$ Demás casos

Finalmente, representamos el sistema de las ecuaciones $(1, \dots, n * m)$ como la matriz extendida de $Ax = B$. A modo de ejemplo la representación del sistema de ecuaciones de un parabrisas discretizado de 4×4 :

$$\begin{pmatrix}
 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -200 \\
 -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -100 \\
 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -200 \\
 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -100 \\
 -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -100 \\
 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & T_{Sang1} \\
 0 & 0 & 0 & 0 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -100 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & -100 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & T_{Sang2} \\
 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & -100 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -200 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & -100 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & -100 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -200
 \end{pmatrix}$$

En este ejemplo se representa un parabrisas discretizado con dos sangüuelas

3.3. ¿Por qué una matriz Banda?

La obtención de la matriz Banda esta ligada directamente con nuestro sistema de ecuaciones, con la ecuación de calor discretizada y con el orden de numeración de las variables. Por ejemplo, la siguiente figura muestra un ejemplo de instancia de un parabrisa:

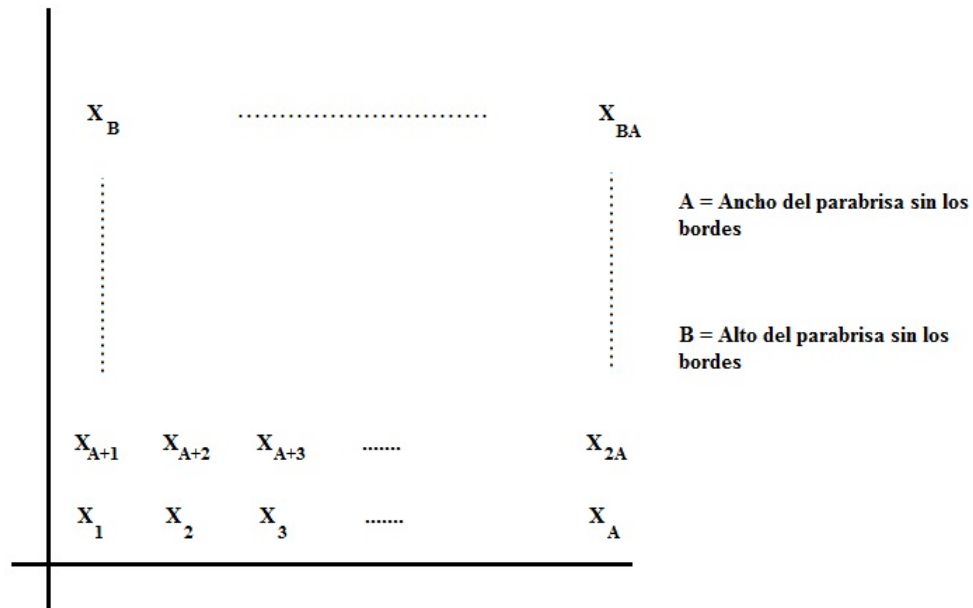


Figura 1: Cada X_i representa un punto de la discretización de este parabrisa

Empezemos a despejar las variables y armar nuestro sistema de ecuaciones:

- $X_1 = (X_2 + X_{a+1} - 100 - 100)/4$
- $4 \cdot X_1 - X_2 - X_{a+1} = -200$

$$\begin{aligned} \blacksquare X_2 &= (X_1 + X_{a+2} + X_3 - 100)/4 \\ 4X_2 - X_1 - X_{a+2} - X_3 &= -100 \end{aligned}$$

Así, sucesivamente hasta despejar todas las incógnitas y completar el sistema. Como consecuencia del uso de la ecuación de calor, cada incógnita va a depender de una incógnita que este a la izquierda, derecha, arriba, y abajo de la misma. Esto se interpreta como que cada una de las variables, se relaciona con a lo sumo dos variable pegadas y con dos variable que están a A de distancia de la variable en estudio (Decimos a lo sumo, para especificar el caso de las variables que estás pegadas a los bordes, o sanguijuelas). Esto, y que empezamos despejando desde el extremo inferior izquierdo del parabrisa, son las causas de que la matriz quede establecida como una matriz banda.

Armando la matriz asociada al sistema de ecuaciones, nos da como resultado la siguiente figura, que muestra la relación de las variables con la distancia A.

$$A = \begin{pmatrix} 4 & -1 & \dots & \dots & -1 & 0 & \dots & \dots & 0 & -200 \\ -1 & 4 & -1 & \dots & \dots & -1 & 0 & \dots & 0 & -100 \\ \vdots & \vdots & \text{A de distancia} & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \text{A de distancia} & \ddots & \vdots & \text{A de distancia} & \vdots & \vdots & \vdots & \vdots \\ 0 & -1 & \dots & -1 & 4 & -1 & \dots & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \text{A de distancia} & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & -1 & \dots & -1 & 4 & -1 & -100 \\ 0 & \dots & \dots & \dots & 0 & -1 & \dots & -1 & 4 & -200 \end{pmatrix}$$

A de distancia

3.4. Estructura de la matriz bandas

Para representar el sistema de ecuaciones, vamos a implementar una estructura que trabaja internamente con una matriz, la cual guarda únicamente los elementos del sistema que nos interesan. Recordemos primero, que dados a, b y h , $n = a/h$, $m = b/h$ entonces estaremos discretizando $(n+1)x(m+1)$ puntos. Dado que hay $(n+1)x(m+1)$ puntos a los cuales debemos asociarles sus respectivas temperaturas, como sabemos a priori que los bordes tienen una temperatura constante de -100, en realidad nos van a interesar las temperaturas de los demás puntos, es decir de $(n-1)x(m-1)$ puntos (ya que hay dos bordes horizontales y dos verticales). Sabemos entonces, que si tenemos $(n-1)x(m-1)$ puntos a los cuales debemos asociarles temperaturas, vamos a tener $(n-1)x(m-1)$ incógnitas y también $(n-1)x(m-1)$ ecuaciones lineales. La idea es aprovechar que la matriz asociada al sistema de ecuaciones tiene la forma de una matriz bandas, usando esto a nuestro favor para reducir la complejidad espacial que requiere el problema.

La idea es utilizar como dijimos antes, una matriz que represente el sistema de ecuaciones. Para hacer esto, no vamos a poder evitar tener $(n-1)x(m-1)$ filas debido a que cada fila va a representar cada punto de la discretización. Lo que sí se puede evitar, es tener $(n-1)x(m-1)$ columnas ya que, al tratarse de una matriz bandas, la ecuación i que representa al punto X_i va

a usar a lo sumo a los puntos $X_{i-(m-1)}$, $X_{i+(m-1)}$, X_{i+1} y X_{i-1} . Entonces como sucede esto, con tener $2x(m-1) + 2$ columnas es suficiente. Las dos columnas adicionales que contamos se deben a que necesitamos en cada fila poder representar el valor correspondiente al coeficiente del punto X_i y, dado que vamos a guardar la matriz asociada al sistema, ésta va a ser de la forma $A|b$, por lo tanto necesitamos una columna más para los elementos de b . Utilizando una matriz de estas dimensiones, nuestra complejidad espacial para representar el sistema pasa de ser $O((n-1)^2(m-1)^2)$ a ser $O((n-1)(m-1)^2)$.

Hasta ahora lo que explicamos es simplemente que es posible almacenar la matriz asociada al sistema de ecuaciones de una manera más eficiente que guardándola todos sus valores. En nuestra implementación, lo que hicimos es que la estructura que representa el sistema de ecuaciones, mantenga privada esta matriz de $(n-1) \times (2(m-1)+2)$ de manera tal que cuando queramos acceder a un elemento de la matriz que representa lo hagamos como si estuviéramos tratando con la matriz entera. Para esto, es necesario poder "mapear" de alguna manera cada elemento de la matriz asociada al sistema de ecuaciones, a la matriz que utiliza internamente nuestra estructura para representarla. Explicaremos a continuación, la lógica detrás del mapeo este:

Sea A_{ij} $1 \leq i \leq n-1$, $1 \leq j \leq m-1$ un elemento de la matriz asociada al sistema de ecuaciones que queremos resolver $A \in \mathbb{R}^{(n-1) \times (m-1)}$ y sea $A' \in \mathbb{R}^{(n-1) \times (2(m-1)+2)}$ la matriz interna que utiliza nuestra estructura para representar a A :

- Si $i = j$, entonces $A_{ij} = A'_{i(m+1)}$ debido a que estamos tratando de acceder al coeficiente de la incógnita correspondiente a la fila i y, por una cuestión de implementación decidimos situarlo en el medio de cada fila.
- Si $j = m-1$ entonces $A_{ij} = A'_{ij}$ ya que estamos tratando de acceder a un b_i y el vector b de resultados de las ecuaciones queda almacenado de manera intacta.
- Si $j > i$ estamos tratando de acceder a un elemento que está a la derecha (en nuestra matriz interna) del elemento que almacena el coeficiente de la incógnita correspondiente a esta fila.
 - Si $j - i \leq m-1$ entonces quiere decir que tenemos almacenado a este elemento y por lo tanto $A_{ij} = A'_{i, m+j-i}$
 - Caso contrario $A_{ij} = 0$ ya que estamos tratando de acceder al coeficiente (en la ecuación i) de un punto que seguro esta ecuación no utiliza.
- Si $j < i$ estamos tratando de acceder a un elemento que está a la izquierda (en nuestra matriz interna) del elemento que almacena el coeficiente de la incógnita correspondiente a esta fila.
 - Si $j - i \leq m-1$ entonces quiere decir que tenemos almacenado a este elemento y por lo tanto $A_{ij} = A'_{i, m-(i-j)}$.
 - Caso contrario $A_{ij} = 0$ ya que estamos tratando de acceder al coeficiente de un punto que seguro esta ecuación no utiliza.

3.5. Eliminacion Gaussiana para matriz Banda

Algorítmicamente, la idea es similar a la Eliminacion Gaussiana clásica. En este caso, como consecuencia de nuestro sistema de ecuaciones, la matriz resultante del sistema asociado se compone por una diagonal integrada:

- Por números 4, si se trata de una posicion sin sanguijuela.
- Por números 1, si se trata de una posición con sanguijuela.

Las demás posiciones de la matriz se llenan con -1 o 0, depende el caso. Aprovechando las características de la diagonal y los posibles números, no es necesario intercambiar filas debido a que, al triangular, la diagonal nunca va a tener números 0. Además, ya que estas matrices tienen la cualidad de tener diagonales de 0 en los extremos superior derecho e inferior izquierdo nos simplifica las siguiente operacion en relación al algoritmo clásico:

- No es necesario restar la fila completa, por la presencia de números 0 en el extremo superior derecho.
- Por cada paso que avanza en la diagonal, no es necesario realizar la resta de filas hasta abajo, por la presencia de 0 en el extremo inferior izquierdo

Además, en el momento de ir consiguiendo los número 0 de la triangulación, se los asigna directamente para no tener problemas de precisión. Resumiendo, el código quedaría así:

Algorithm 1 EliminacionGaussiana0()

```

1: for  $j = 0; \text{filas}; j++$  do
2:   for  $(i = j + 1; (i \leq j + \text{ancho}) \&\& (i < \text{filas}); i++)$  do
3:     coeficiente = (Obtener(i,j)/Obtener(j,j));
4:     RestarFila0(coeficiente, j, i, j+1);
5:   end for
6:   cerosAIzquierda(j,j);
7: end for

```

Siendo ancho, el ancho del parabrisa, es decir el ancho de la banda, filas la cantidad de filas de la matriz y restarFila0 es la función para restar filas con la particularidad de lo expresado anteriormente

3.6. BackWard Substitution para matriz Banda

Aprovechando la estructura de las matrices bandas, al aplicar Eliminación Gaussiana, la matriz triangular superior resultante hereda las bandas superiores de 0. Ante esto el algoritmo clásico de BackWard Substitution se puede modificar para aprovechando que algunos valores son 0. El caso de ForWard Substitution es análogo

3.7. Matriz LU para matriz Banda

Consideremos A una matriz banda, $A \in \mathbb{R}^{n \times n}$, con b_s y b_i las bandas superiores e inferiores correspondientemente y $b_s = d$, $b_i = d$.

Al momento de representar la matriz LU de A, se aprovechó que la misma fuese bandas. Ya que entonces, la matriz LU hereda el número de bandas de A. Para la matriz L, esto se puede ver al momento de aplicar la Eliminación Gaussiana en una matriz de este tipo. Si desarrollamos el primer paso de la eliminación Gaussiana lo que se realiza es la diferencia entre $F_i - (a_{1,i}/a_{1,1})1 * F_1, \forall 2 \leq i \leq d+1$. Notemos que i solo tiene que llegar hasta $d+1$ debido a que se trata de una matriz banda (el por qué de este hecho se explicó en la pág 8, *Eliminacion Gaussiana para matriz Banda*). A su vez, en el primer paso podemos obtener los multiplicando: $\{l_{2,1} = a_{1,1}/a_{2,1}, l_{3,1} = a_{1,1}/a_{3,1}, \dots, l_{d+1,1} = a_{1,1}/a_{d+1,1}\}$; en el siguiente paso haremos $F_i - (a_{2,i}/a_{2,2})1 * F_2, \forall 3 \leq i \leq d+2$ y obtendremos: $\{l_{3,2} = a_{2,2}/a_{3,2}, l_{4,2} = a_{2,2}/a_{4,2}, \dots, l_{d+2,2} = a_{2,2}/a_{d+2,2}\}$, así hasta calcular $F_{n-1} - (a_{n-1,n-1}/a_{n,n-1})1 * F_{n-1}$ y obteniendo el multiplicando $l_{n,n-1} = (a_{n-1,n-1}/a_{n,n-1})$. Notemos los últimos multiplicando de cada paso, y obtenemos: $\{l_{d+1,1}, l_{d+2,2}, \dots, l_{d+i,i}, \dots, l_{n,n-1}\}$. Es decir que los elementos de las filas fila_i $d+2 \leq i \leq n$ son ceros hasta la columna i, lo cual es el comportamiento de una matriz banda con $b_i = d$. Si lo representamos obtenemos la matriz L =

$$\begin{pmatrix} 1 & & & & & & & & & \\ l_{2,1} & 1 & & & & & & & & \\ l_{3,1} & l_{3,2} & 1 & & & & & & & \\ \vdots & \vdots & \vdots & & & & & & & \\ l_{d+1,1} & l_{d+1,2} & \cdots & l_{d+1,d} & 1 & & & & & \\ 0 & l_{d+2,2} & l_{d+2,3} & \cdots & l_{d+2,d+1} & 1 & & & & \\ \vdots & \ddots & & & & \vdots & & & & \\ \vdots & & 0 & l_{n-1,n-1-d} & \cdots & \cdots & l_{n-1,n-2} & 1 & & \\ 0 & \cdots & \cdots & 0 & l_{n,n-1-d} & \cdots & \cdots & \cdots & l_{n,n-1} & 1 \end{pmatrix}$$

representación de los elementos ubicados antes de la diagonal, quienes conforman la matriz L . Como se puede observar la matriz L tiene d bandas inferiores, al igual que la matriz A .

Mientras que para la matriz U , si $d_s = d$ significa que heredó la misma cantidad de ceros que no conformaban las bandas superiores del triangulante superior. Esto se observa cuando realizamos las restas entre filas en la Eliminación Gaussiana tomemos el paso k . La misma establece que debemos realizar $F_i - (a_{i,k}/a_{k,k}) * F_k, \forall k+1 \leq i \leq \min(n, k+d)$. como A tiene d bandas superiores, luego de la posición $a_{k,k+d}$ la fila_k tiene ceros. Cuando se realiza $F_i - (a_{i,k}/a_{k,k}) * F_k$, luego de $a_{i,i+d}$ F_i tiene ceros y como $k+d < i+d$. Todas las posiciones luego de $i+d$ van a seguir siendo cero ya que se estan restando solo estos valores. Es por esto que en la Eliminación Gaussiana implementada, cuando calculamos la diferencia entre dos fila sólo lo hacemos entre d de sus elementos.

Al conocer las propiedades que va a tener la matriz LU (la misma cantidad de bandas que la matriz A a la cual se aplicó dicha factorización). Para optimizar en costo espacial. Lo que se implementó fue, que a medida que se factorizaba la matriz A guardabamos los multiplicando en la misma matriz. Y luego, mediante dos funciones, pedir la matriz L y la matriz U . Cada una representada con un vector de vectores. Pero, como cada fila de las mismas tiene a lo sumo $d+1$ elementos distintos de cero (con d la cantidad de bandas superiores e inferiores y el uno que se adiciona es por contar el elemento de la diagonal) cada uno de los vectores interiores era de a lo sumo de tamaño $d+1$. Representando de esta manera solo los elementos que se ubican dentro de las bandas correspondientes para cada matriz.

3.8. Experimento 1

Para el experimento número uno, comportamiento de la distribución de la temperatura, se evaluaron distintos casos. Como el cálculo de la temperatura del punto crítico difiere según sea el tamaño del parabrisas y su posterior discretización. Se trabajó con distintos tamaños. Más aún ingresamos otros factores entre los distintos casos, como el número de sanguijuelas y el radio de acción de cada una de estas. Además, vamos a comparar como varía la temperatura en el punto crítico (si es que lo hace) para un mismo parabrisas pero, variando la granularidad del mismo. De esta forma, aunque estamos representando a un mismo parabrisas, obtenemos discretizaciones de diferentes dimensiones.

3.8.1. Hipótesis

Hipótesis1: A medida que la granularidad baja, la temperatura en el punto crítico aumenta;
 Hipótesis2: Para granularidades de mayor valor las temperaturas se van distorsionando aún más;
 Hipótesis3: Las diferencias de temperaturas van a ser mayores si se trabaja con un parabrisas de tamaño IMPARxIMPAR;

Hipótesis4: Si en un sistema predominan sanguijuelas unitarias entonces, se van a presentar mayores diferencias entre una discretización y otra.

3.9. Experimento 2

3.9.1. Eliminación Gaussiana vs Factorización LU

La idea en esta sección es analizar ambos métodos y realizar una suerte de comparación para ver cuando es que conviene utilizar uno y cuando el otro. Si bien los dos nos ayudan a resolver un sistema de ecuaciones lineales y están estrechamente vinculados, existen ocasiones donde utilizar LU en lugar de Eliminación Gaussiana, puede ahorrarnos mucho tiempo de cómputo. Para empezar, es importante recordar que *no siempre* vamos a poder factorizar una matriz y por ende, cuando esto suceda no va a quedar otra opción que utilizar Eliminación Gaussiana. Para poder hacer la comparación entonces, vamos a asumir que en los casos que nombremos se va a poder utilizar el método de factorización.

A priori, la complejidad teórica de ambos algoritmos es $O(n^3)$ así que eso no nos dice cuál es mejor en la práctica, por lo tanto habrá que probar ambos para el caso particular con el cual estemos trabajando y de ahí decidir cuál se comporta mejor. Llamemos $A|b$ al sistema de ecuaciones lineales asociado al problema que queremos resolver. Si nuestro objetivo es simplemente resolver ese sistema y de ahí en más trabajar únicamente con esa solución, entonces sí la mejor idea es probar ambos algoritmos y ver cuál se comporta mejor en la práctica. Ahora, podría suceder que el problema no sea solo eso, si no que necesitemos resolver $A|b_1, A|b_2, \dots, A|b_t$ donde lo único que varía es el b_i . En estos casos el método de Factorización LU es claramente superior al método de Eliminación Gaussiana en términos de tiempo de cómputo. Esto se debe a que, si bien encontrar a L (triangular inferior) y a U (triangular superior) tales que $A = LU$ es $O(n^3)$ al igual que el algoritmo de Eliminación Gaussiana, una vez que ya hicimos este paso, resolver cada uno de los $A|b_i$ tiene complejidad $O(n^2)$. Si utilizáramos Eliminación Gaussiana, estaríamos pagando un costo de $O(n^3)$ por cada sistema $A|b_i$ que resolvamos. Esto quiere decir que en este caso, resolver el problema completo con el método de factorización es $O(n^3 + t.n^2)$ mientras que por Eliminación Gaussiana es $O(t.n^3)$. Es fácil ver que si t es grande, entonces la brecha entre el tiempo de cómputo consumido por resolver el problema utilizando Eliminación Gaussiana y utilizando Factorización LU va a ser grande también. Incluso si t no es grande puede haber una diferencia notable en la práctica, ya que cuando hablamos de complejidad teórica estamos hablando en términos asintóticos.

En el contexto del trabajo práctico, el método de Factorización LU juega un papel importante a la hora de utilizar la fórmula de Sherman-Morrison. Esto se debe a que hay que computar $A^{-1}b$ y $A^{-1}u$, lo que equivale a resolver los sistemas $Ax_1 = b$ y $Ax_2 = u$ y ahí es donde podemos reusar la L y la U . En el caso de tener una cantidad considerable de sanguijuelas unitarias, el tiempo de cómputo ahorrado es grande, ya que por cada sanguijuela unitaria que haya se va a estar pagando un costo de $O(n^2)$ utilizando LU contra uno de $O(n^3)$ utilizando Eliminación Gaussiana.

3.9.2. Experimento: calidad/tiempo de cómputo

La idea del siguiente experimento, es tratar de ver como se relaciona el tiempo de cómputo necesario para resolver el sistema de ecuaciones planteado, con la calidad de la solución encontrada. Es importante para esto, decir a qué nos referimos cuando hablamos de calidad. Es necesario para responder esta pregunta, recordar que nuestro objetivo es calcular la temperatura en cierto punto del parabrisas. Para lograr esto, lo que hacemos es discretizar la superficie (porque estamos trabajando con aritmética finita) y pasar de una superficie que tiene infinitos puntos a una representación de dicha superficie con una cantidad finita de puntos. Ahora bien, cuando realizamos la discretización y pasamos de un problema en el espacio continuo a uno en el espacio discreto estamos perdiendo información. Entonces cuando hablamos de calidad, nos referimos justamente a cuanta información estamos perdiendo, como costo de modelar en problema de manera discreta.

El experimento se planteó para trabajar con instancias de 20x20 y 40x40, a los fines del mismo, la temperatura y posición de las sanguijuelas no tienen mucha relevancia como sí la tienen la granularidad y los radios de dichas sanguijuelas. Básicamente, se tomó una instancia de 20x20 generada de manera manual y se fueron modificando los valores de los radios y granularidad. En la elección de la instancia de 40x40 lo que importa pasa a ser la cantidad de sanguijuelas, por eso es que se generó de manera distinta (ver epígrafe del resultado del experimento).

Una vez que fijamos la instancia que vamos a utilizar y a ir modificando para realizar las mediciones, lo que se va a medir es como varía la temperatura en el punto crítico y tiempo de cómputo en función de la granularidad, la cantidad de sanguijuelas y el tamaño de los radios. Utilizaremos el algoritmo de Eliminación Gaussiana (modo 0). Los tiempos fueron tomados en segundos.

Ahora que sabemos qué es lo que se intenta determinar en el experimento, vamos a tratar de pensar qué es lo que debería pasar (y luego contrastarlo con los resultados para ver si efectivamente es lo que sucede) a grandes rasgos, planteando así una serie de hipótesis.

1. En cuanto al tiempo de cómputo, sabemos que como vamos a estar utilizando el algoritmo de Eliminación Gaussiana, la cota teórica va a ser $O(n^3)$. El sistema de ecuaciones con el que trabajaremos va a ser de $n' \times n'$ donde $n' = (b/h - 1) \times (a/h - 1)$ siendo a (ancho del parabrisas), b (alto del parabrisas), h (granularidad) parámetros del problema. Podemos ver como el tamaño del sistema (y por lo tanto el tiempo de cómputo ya que es una función del tamaño) depende directamente de h , de manera tal que si h es chico, entonces el tamaño es grande. Esto nos dice entonces que, dado una instancia del parabrisas (con sus dimensiones y sanguijuelas), cuanto más chico sea h , más tiempo se va a tardar en resolver el problema.
2. Dado que para poder computar la temperatura de los puntos de nuestra discretización es necesario saber cuáles de ellos son afectados por alguna sanguijuela, una cantidad grande de sanguijuelas impacta negativamente en el tiempo de cómputo.
3. Debido a que perdemos información al discretizar el problema porque estamos trabajando con un espacio discreto, cuando en realidad el problema es de naturaleza continua, parece razonable pensar que cuanto más chico sea h , la solución obtenida va a estar más cerca de la real.
4. Nuestra última hipótesis se desprende de alguna manera de la que enunciamos en el punto anterior. Si al aumentar el tamaño de nuestra discretización estamos acercándonos más a la verdadera solución del problema y las temperaturas de los puntos dependen de las sanguijuelas, parece razonable pensar que haciendo esto tenemos menos chances de descartar sanguijuelas, ya que nuestra discretización es "más adecuada".

3.10. Cálculo de la temperatura en el punto crítico

En esta sección, explicaremos brevemente como se calcula la temperatura en el punto crítico. Como explicaremos más adelante, dado que estamos discretizando el problema, puede llegar a ocurrir que, dependiendo de las dimensiones de nuestra discretización, no exista **el** punto crítico. Por lo tanto esto va a ser una explicación de las decisiones que tomamos en estos casos, más que una explicación sobre cómo se "debe calcular" la temperatura en el punto crítico ya que podrían existir otros criterios.

Existen entonces cuatro casos distintos a la hora de calcular esta temperatura y, todos ellos dependen de las dimensiones de nuestra discretización. Sea n' el alto de nuestra discretización y m' el ancho

- Si tanto n' como m' son pares quiere decir que no vamos a poder encontrar el punto del medio de nuestra discretización. Lo que haremos en este caso será calcular un promedio de los cuatro puntos que estarían "rodeando" de alguna manera al punto crítico. Estos puntos

serían $(n'/2, m'/2)$, $(n'/2+1, m'/2)$, $(n'/2, m'/2+1)$ y $(n'/2+1, m'/2+1)$ (estamos indexando desde 1).

- Si n' y m' son ambos impares, entonces efectivamente vamos a poder encontrar el punto crítico de nuestra discretización. Dicho punto será $(n'/2 + 1, m'/2 + 1)$
- Si n' es impar y m' par, no vamos a poder encontrar el punto crítico de nuestra discretización porque no hay un punto medio a lo ancho, pero sí a lo largo. Con lo cual la temperatura del punto crítico será estimada con el promedio de las temperaturas de los dos puntos que estarían rodeándolo, o sea $(n'/2 + 1, m'/2)$ y $(n'/2 + 1, m'/2 + 1)$.
- Si n' es par y m' impar ahora no podemos encontrar el punto crítico porque no hay un punto medio a lo largo, con lo cual la temperatura se estima con el promedio de los puntos a lo largo que lo rodean, dejando fija la coordenada correspondiente al ancho: $(n'/2, m'/2 + 1)$ y $(n'/2 + 1, m'/2 + 1)$.

Una vez que ubicamos él o los puntos requeridos para realizar el cálculo, resta ver como hacemos para obtener sus respectivas temperaturas. Dado el vector b de temperaturas, si queremos el valor de la temperatura del punto (i, j) de nuestra discretización, ese valor estará en el elemento $m'i + j$ de b . Esto es así porque el vector b guarda las temperaturas de los puntos de manera tal que los primeros m' elementos corresponden a la primer fila de la discretización, los segundos m' elementos a la segunda y así con lo cual al multiplicar por m' nos estamos moviendo i filas (que es lo que queremos) y al sumar j nos estamos posicionando en la columna correspondiente, obteniendo así el elemento del vector correspondiente a la temperatura del punto (i, j) .

Es importante aclarar que, en la implementación los índices difieren levemente debido a que tenemos que indexar desde 0, sin embargo, estos son los conceptos que subyacen a lo que está implementado.

3.11. Experimento 3

Uno de los objetivos del presente trabajo es determinar si se puede salvar el parabrisas eliminando una sola sanguijuela. Y de ser posible que eliminemos la mejor. Para esto se plantean dos posibles formas. La segunda es utilizar la fórmula Sherman–Morrison* (CITAR FUENTE CON LA EXPLICACION) la misma plantea que:

$$(A + uv^t)^{-1} = A^{-1} - \frac{A^{-1}uv^tA^{-1}}{1 + v^tA^{-1}u}. \quad (5)$$

Siendo A una matriz inversible, y u, v^t dos vectores.

Aplicandola siempre que sea posible su uso. Para esto analicemos el caso de un sistema que posee al menos una sanguijuela unitaria, luego de discretizar y platear el sistema de ecuaciones para el problema obtendremos una matriz como la siguiente: $A \in \mathbb{R}^{n \times n}$; $A =$

$$\begin{pmatrix} 4_{1,1} & -1_{1,2} & \cdots & -1_{1,1+d} & \cdots & \cdots & \cdots & 0_{1,n}| & -200 \\ -1_{2,1} & 4_{2,2} & -1_{2,3} & \cdots & -1_{1,2+d} & \cdots & \cdots & 0_{2,n}| & -100 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0_{i,1} & \cdots & 0_{i,i-1} & 1_{i,i} & 0_{i,i+1} & \cdots & \cdots & 0| & T_{sang_i} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0_{n-1,1} & \cdots & \cdots & -1_{n-1,n-1-d} & \cdots & -1_{n-1,n-2} & 4_{n-1,n-1} & -1_{n-1,n}| & -100 \\ 0_{n,1} & \cdots & \cdots & \cdots & -1_{n,n-d} & \cdots & -1_{n,n-1} & 4_{n,n}| & -200 \end{pmatrix}$$

la fila i , representa a la sanguijuela unitaria $sang_i$, con T_{sang_i} su correspondiente temperatura y d el ancho del parabrisas discretizado.

Con el sistema a resolver $Ax = b$.

Pero, al eliminar $sang_i$ ese punto dejara de tener el valor T_{Sang_i} . Para pasar a ser el promedio de los puntos $(x+1,y)$, $(x-1,y)$, $(x,y+ancho)$, $(x,y-ancho)$ (con ancho referenciamos al ancho del parabrisas discretizado y sin contar los bordes, y (x,y) coordenadas en la discretización de la $Sang_i$). De esta forma obtenemos el siguiente sistema: $A'x = b'$.

Con la matriz asociada:

$$A' \in \mathbb{R}^{n \times n}; A' =$$

$$\begin{pmatrix} 4_{1,1} & -1_{1,2} & \cdots & -1_{1,1+d} & \cdots & \cdots & \cdots & 0_{1,n}| & -200 \\ -1_{2,1} & 4_{2,2} & -1_{2,3} & \cdots & -1_{1,2+d} & \cdots & \cdots & 0_{2,n}| & -100 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -1_{i,i-d} & \cdots & -1_{i,i-1} & 4_{i,i} & -1_{i,i+1} & \cdots & -1_{i,d+i} & 0| & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0_{n-1,1} & \cdots & \cdots & -1_{n-1,n-1-d} & \cdots & -1_{n-1,n-2} & 4_{n-1,n-1} & -1_{n-1,n}| & -100 \\ 0_{n,1} & \cdots & \cdots & \cdots & -1_{n,n-d} & \cdots & -1_{n,n-1} & 4_{n,n}| & -200 \end{pmatrix}$$

Notemos que la única modificación se produjo en la fila i . El resto de la matriz sigue siendo la misma. Entonces podríamos reescribir a A' como:

$$A' = A + B \quad (1);$$

Para algún B conveniente

Por lo que $B = A' - A$

Denotemos a cada matriz de la siguiente forma:

$$A = \begin{pmatrix} A_1^t \\ A_2^t \\ \vdots \\ A_i^t \\ \vdots \\ A_n^t \end{pmatrix}; A' = \begin{pmatrix} A_1'^t \\ A_2'^t \\ \vdots \\ A_i'^t \\ \vdots \\ A_n'^t \end{pmatrix}; B = A - A' = \begin{pmatrix} A_1^t - A_1'^t \\ A_2^t - A_2'^t \\ \vdots \\ A_i^t - A_i'^t \\ \vdots \\ A_n^t - A_n'^t \end{pmatrix} \quad (2)$$

con $A_j, B_j, A'_j \in \mathbb{R}^n, \forall j, 1 \leq j \leq n$

Como ya dijimos $\forall j, j \neq i, A_j = A'_j$ Entonces a partir de (2) podemos concluir que $\forall j, j \neq i, B_j = 0$

Mientras que:

$$B_i = A'_i - A_i$$

$$B_i = (0, \dots, -1_{i,i-d}, \dots, -1, 4_{i,i}, -1, \dots, -1_{i,i+d}, \dots, 0) - (0, \dots, 0, 1_{i,i}, 0, \dots, 0)$$

$$B_i = (0, \dots, -1_{i,i-d}, \dots, -1, 3_{i,i}, -1, \dots, -1_{i,i+d}, \dots, 0). \quad (\text{Siendo } d \text{ el ancho del parabrisas discretizado})$$

De esta manera ya conocemos nuestra matriz B . dado que $B \in \mathbb{R}^{n \times n}$. Podemos pensar a B como el producto de dos vectores u y v^t , con $u, v \in \mathbb{R}^n$

1) Primero empezemos por analizar el producto en general de estos dos vectores:

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_i \\ \vdots \\ u_n \end{pmatrix} ; v^t = (v_1, \dots, v_2, \dots, v_n) ; u * v^t = \begin{pmatrix} u_1 * v_1 \cdots u_1 * v_k \cdots u_1 * v_n \\ u_2 * v_1 \cdots u_2 * v_k \cdots u_2 * v_n \\ \vdots \\ u_i * v_1 \cdots u_i * v_k \cdots u_i * v_n \\ \vdots \\ u_n * v_1 \cdots u_n * v_k \cdots u_n * v_n \end{pmatrix} = \begin{pmatrix} u_1 * v^t \\ u_2 * v^t \\ \vdots \\ u_i * v^t \\ \vdots \\ u_n * v^t \end{pmatrix}$$

Como queremos que $B = u * v^t$

$$\begin{pmatrix} 0 \\ 0 \\ \vdots \\ B_i \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} u_1 * v^t \\ u_2 * v^t \\ \vdots \\ u_i * v^t \\ \vdots \\ u_n * v^t \end{pmatrix}$$

Por lo que si pensamos a u como el vector canónico e_i y a v^t como la fila que queremos sumar, es decir como B_i tenemos que:

$$u * v^t = \begin{pmatrix} 0 * B_i \\ 0 * B_i \\ \vdots \\ 1 * B_i \\ \vdots \\ 0 * B_i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ B_i \\ \vdots \\ 0 \end{pmatrix}$$

El objetivo es resolver:

$$A'x = b'$$

$$(A + B)x = b' \text{ por (1)}$$

$$(A + uv^t)x = b'$$

$$\text{Como } A' \text{ es inversible } (A + u * v^t)^{-1}b' = x$$

Pero ahora podemos aplicar la formula de Sherman–Morrison, de esta manera tenemos que:

$$x = A^{-1} * b' - \frac{A^{-1}uv^t A^{-1} * b'}{1 + v^t A^{-1}u}. \quad (3) \quad (6)$$

De esta fórmula lo único que resta calcular es: $A^{-1} * u$ (4) y $A^{-1} * b'$ (5).

Pero, podemos reescribir a (4) y a (5) como:

$A * z = u$ y $A * y = b'$. Y resolver ambos sistemas. Pudiendo concluir que la fórmula es aplicable cada vez que queremos eliminar una sanguijuela unitaria.

Análisis de la complejidad y algoritmo:

A continuación se presenta un pseudo-código del algoritmo implementado para la fórmula Sherman–Morrison:

Algorithm 2 Sherman–Morrison(matriz L, matriz U, vector u, coordenadas (x,y))

```

1:  $z' \leftarrow \text{ForWardSubstitution}(u, L) \ O(n^2)$ 
2:  $A^{-1} * u \leftarrow \text{BackWardSubstitution}(z', U) \ O(n^2)$ 
3:  $y \leftarrow \text{ForWardSubstitution}(b', L) \ O(n^2)$ 
4:  $A^{-1} * b' \leftarrow \text{BackWardSubstitution}(y, U) \ O(n^2)$ 
5:  $v^t \leftarrow \text{vector de longitud } 5 \ (*1) \ O(1)$ 
6: if (x no esta en el borde superior) then
7:    $v^t[1] \leftarrow -1 \ O(1)$ 
8: end if
9: if (x no esta en el borde derecho) then
10:   $v^t[2] \leftarrow -1 \ O(1)$ 
11: end if
12:  $v^t[3] \leftarrow 3 \ O(1)$ 
13: if (x no esta en el borde izquierdo) then
14:   $v^t[4] \leftarrow -1 \ O(1)$ 
15: end if
16: if (x no esta en el borde inferior) then
17:   $v^t[5] \leftarrow -1 \ O(1)$ 
18: end if (En los casos contrarios completar la posición de  $v^t$  con cero)  $O(1)$ 
19:  $k \leftarrow 0 \ O(1)$ 
20:  $l \leftarrow 1 \ O(1)$ 
21: for  $i = 1$  to  $5$  do
22:   if ( $v^t[i] \neq 0$ ) then
23:     $k \leftarrow (\text{producto de } v^t[i] \text{ con } A^{-1} * b' \text{ segun la fila a la cual hace referencia}) + k; \ (*2)$ 
     $O(1)*5$ 
24:     $l \leftarrow (\text{producto de } v^t[i] \text{ con } A^{-1} * u \text{ segun la fila a la cual hace referencia}) + l; \ O(1)*5$ 
25:   end if
26: end for
27: Return devolver un vector con las restas entre  $(A^{-1} * b)[i]$  y  $((k/l)A^{-1} * u)[i] \ O(n)$ 

```

Siendo L y U las correspondientes matrices de la factorización LU de A ; u es el vector canónico e_i , con i la fila que representa en la matriz la sanguijuela a eliminar; y las coordenadas (x,y) del parabrisas discretizado (recordemos que en el parabrisas discretizado no contamos los bordes del parabrisas original. Por lo que al referirnos a bordes en el pseudo-código hacemos referencia a la nueva representación)

Explicación:

Empezando por las operaciones $A * z = u$ y $A * y = b'$:

Si ya contamos con la factorización LU de A , reescribimos:

$LU * z = u$;

llamamos $z' = U * z$ (6) y $L * z' = u$ (7);

Resolvemos (7) mediante la técnica Forward Substitution. Y realizado esto, podemos calcular (6) aplicando Backward Substitución. De esta forma la complejidad de ambas técnicas cuestan $O(n^2)$. Siguiendo la misma idea resolvemos $A * y = b'$ en $O(n^2)$. Luego, procedemos a efectuar los productos $v^t A^{-1} * b'$ (8) y $v^t A^{-1} u$ (9). En (*1) se crea el vector v^t pero con un tamaño igual a 5, ya que como se explico en la pág 15, el mismo se utiliza como la nueva fila que hay que sumarle a A para llegar a A' . Por lo que su tamaño debería ser igual al largo de la fila de ambas matrices. Sin embargo, como a lo sumo solo va a poseer 5 valores distintos de cero, optamos por acotar su tamaño. Siguiendo con el producto (8), en (*2) efectuamos esta operación. Teniendo en cuenta la idea del vector v^t , $v^t[1]$ se corresponde con la posición f - ancho del vector solución de $A^{-1} * b'$ (f fila en la que se representan las coordenadas (x,y) en la matriz asociada al parabrisas discretizado. La misma fue calculada como $f = \text{ancho} * y + x$); $v^t[2]$ con $f-1$ de dicho vector, etc. De esta manera obtenemos el escalar solución en $O(1)$. Con la misma logica obtenemos el escalar $1 + v^t A^{-1} u$ en $O(1)$. Lo que resta hacer es la diferencia entre ambos vectores soluciones (linea 27), costo $O(n)$. Por lo que el costo temporal total, para eliminar k sanguijuelas unitarias, es $k.O(n^2) + O(n^3)$ (costo

de obtener LU).

3.11.1. Eliminar sanguijuela

Ante lo observado y desarrollado, se llevará a cabo una cierta cantidad de experimentaciones para poder contrastar las siguiente hipótesis:

1. Al utilizar factorización LU y la formula de Sherman-Morrison, el algoritmo de EliminarSanguijuela logra una mejor performance en los casos de que existieran sanguijuelas unitarias.
2. Por otra parte quisimos ver que pasaba al modificar la granularidad. En otras palabras, verificar si al disminuir la misma, la posibilidad de encontrar sanguijuelas unitarias también disminuía, y en consecuencia el algoritmos modificado con Sherman-Morrison bajaba en su rendimiento, asemejándose en complejidad al algoritmo original

Para verificar la hipótesis (1), se llevará a cabo una experimentación tomando 3 instancias, cada una describiendo el caso mejor, caso promedio y caso peor de la cantidad de sanguijuelas unitarias. Para concluir se tomarán los tiempos de ejecución del algoritmo original y el algoritmo modificado por Sherman-Morrison por cada instancia y se los comparará. Para verificar la hipótesis (2), se llevará a cabo otra experimentación tomando 3 instancias exactamente iguales, a excepción de la granularidad, que se la irá cambiando.

4. Resultados

4.1. Experimento 1

4.1.1. Primer caso

Para el primer caso lo que se hizo fue trabajar con un parabrisas de tamaño $PAR \times PAR$ (a par y b par). En el caso de la función `dameTempPtoCritico` ésta representación entra como el caso `IMPARxIMPAR` (ya que recordemos que se resta una unidad a las medidas de ancho y alto para no contar los bordes). De esta forma, la temperatura en el punto crítico está dada por un solo punto. El radio de las sanguijuelas puede variar hasta los 15 metros, para tener una mayor diversidad de tamaños, y tratar de verificar lo planteado en la hipótesis 5.

Se trabajó con 3 discretizaciones distintas 2, 5 y 10. Los resultados se pueden observar en los gráficos: *figure 1.a*, *figura 1.b* y *figure 1.c*

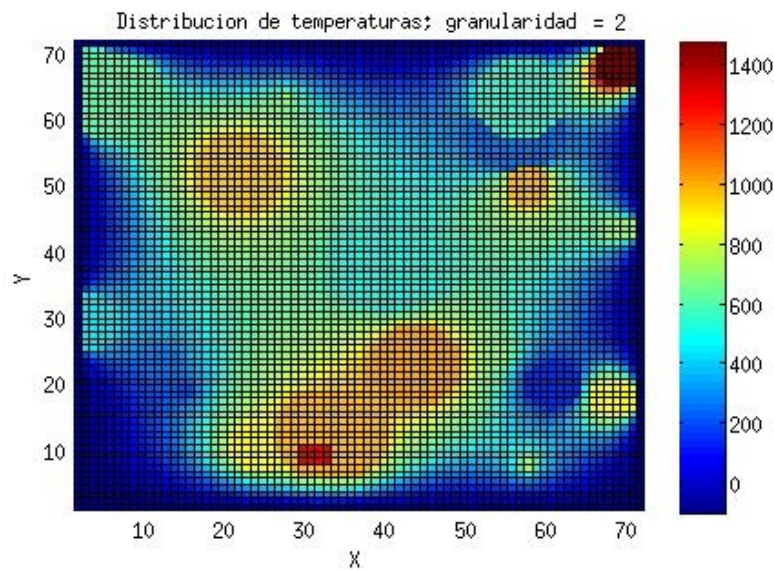


Figure 1.a: Distribución de temperaturas para un parabrisas con $a = 140$ metros; $b = 140$ metros; $h = 2$; $n = 25$. Donde el radio máximo de cada sanguijuela es 15 metros. Y la temperatura en el punto crítico es de 499.00000°C .

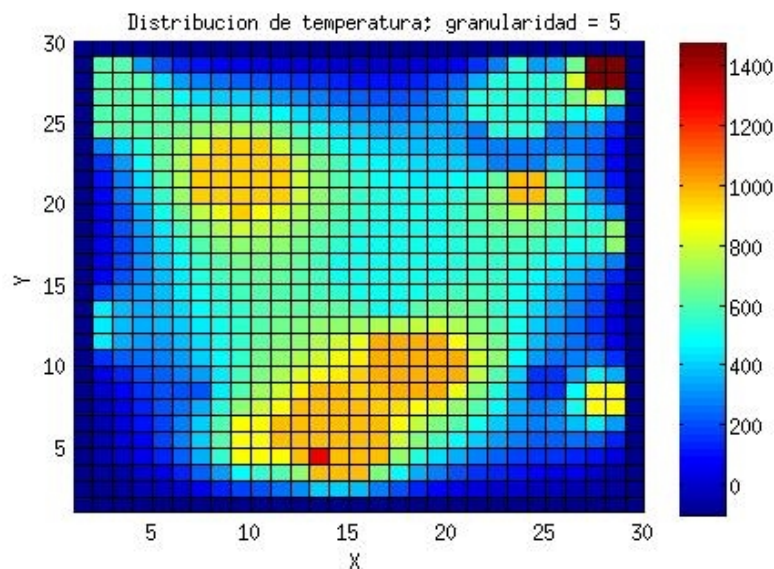


Figure 1.b: Mismas características que para el parabrisas representado en la *figure 1.a*. Pero, con granularidad igual a 5. La temperatura en el punto crítico es de 499.00000 °c.

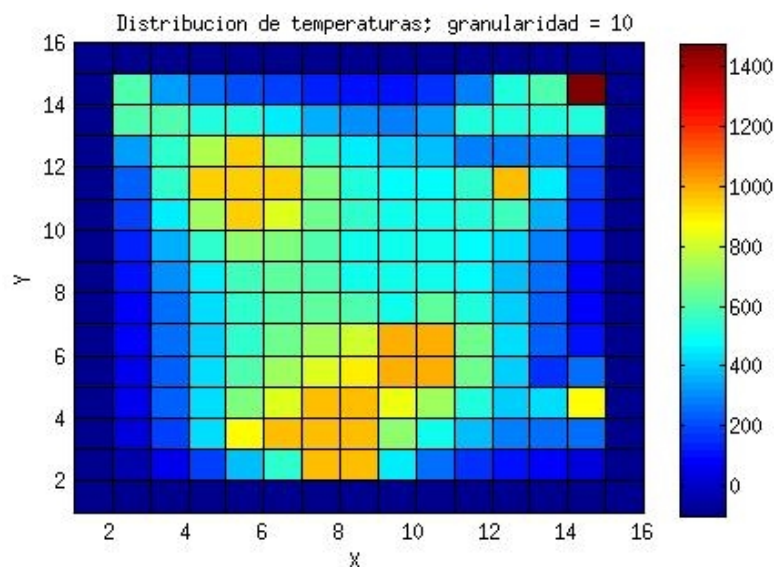


Figure 1.c: Mismas características que para el parabrisas representado en la *figure 1.a*. Pero, con granularidad igual a 10 y cuya temperatura en el punto crítico es de 499.00000 °c .

4.1.2. Segundo caso

Otro caso que analizamos fue trabajar con un parabrisas de tamaño IMPARxIMPAR (lo que se corresponde al caso PARxPAR, al momento de obtener la temperatura en el punto crítico). El motivo de trabajar con este caso, fue que como consecuencias de las dimensiones del parabrisas la temperatura en el punto crítico ahora depende de 4 valores en total. Basta con que al menos uno de los mismos se modifique, entre las distintas discretizaciones, para que la temperatura en el punto crítico cambie. Esta idea es la que se refleja en la hipótesis3. Además, se trabajó con una granularidad menor a uno en este caso. Ya que a diferencia de los

restantes las dimensiones del parabrisas son menores y dado que al momento de discretizar. Los parametros ancho y alto son divididos por la granularidad, el sistema cuenta con mas puntos que para los valores mayores e iguales a 1. Y para parabrisas de mayor tamaño el tiempo de computo sería excesivamente caro. Una característica de esta experimentación es que predominan las denominadas Sanguijuelas unitarias. Dicha elección se realizo para tratar de verificar la hipótesis4. Los resultados obtenidos son los presentados en los gráficos *figure 2.a*, *figure 2.b*, *figure 2.c* y *figure 2.d*

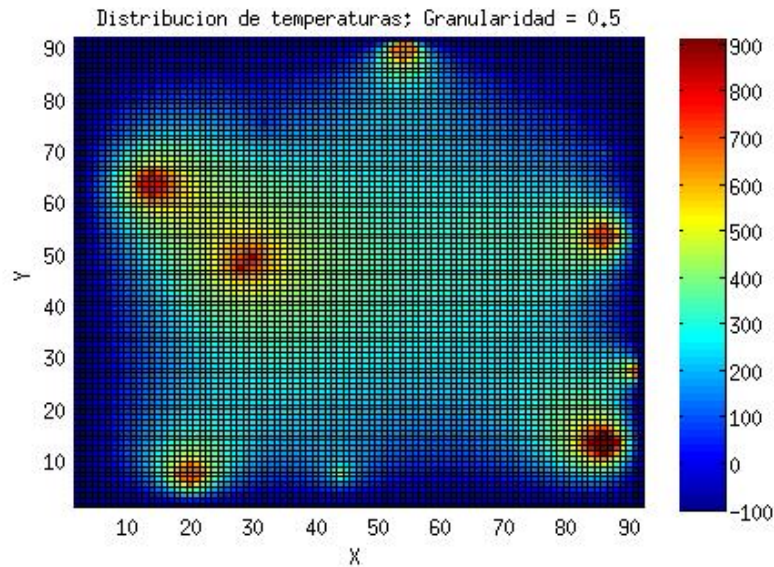


Figure 2.a: Distribución de temperaturas para un parabrisas con $a = 45$ metros; $b = 45$ metros; $h = 0.5$; $n = 15$. Donde el radio máximo de cada sanguijuela es de 2 metros. Y la temperatura en el punto crítico es de $350.69248^{\circ}\text{C}$.

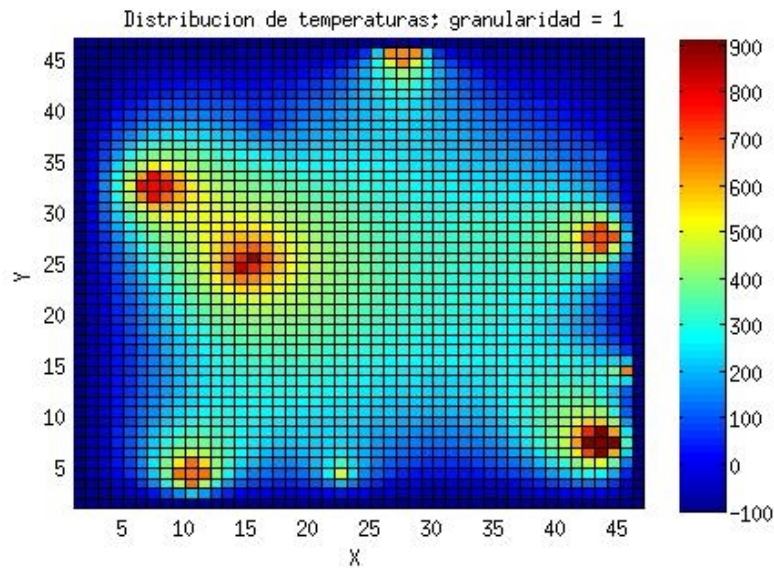


Figure 2.b: Mismas características que para el parabrisas representado en la *figure 2.a*. Pero, con granularidad igual a 5 y cuya temperatura en el punto crítico de $370.11472^{\circ}\text{C}$.

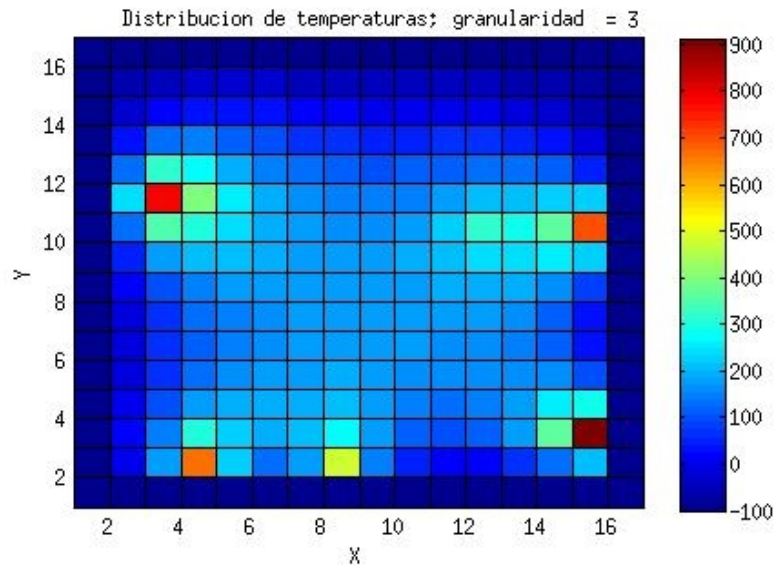


Figure 2.c: Se varía la granularidad del parabrisas representado por la *figure 2.a* a 3. La temperatura en el punto crítico, de esta nueva representación, es de $175.58285^{\circ}\text{C}$.

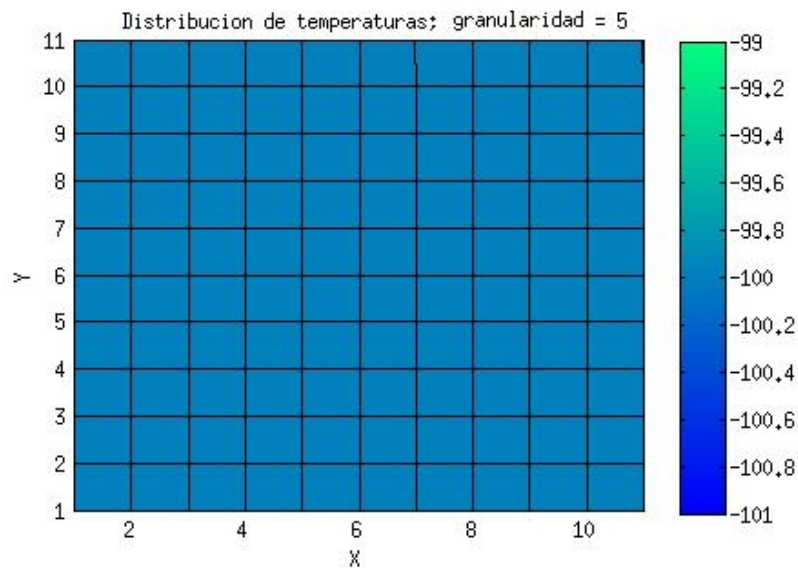


Figure 2.d: Mismas características que para el parabrisas representado en la *figure 2.a*. Pero, con granularidad igual a 5 y cuya temperatura en el punto crítico es de $-100.00000^{\circ}\text{C}$.

4.1.3. Tercer caso

Para continuar con la idea de ver cuanto difiere las temperaturas en un parabrisas de tamaño IMPARxIMPAR. Pero, sobre todo para corroborar la hipótesis4. Se trabajó con el mismo parabrisas que en el caso anterior. Por lo que ahora se aumentó el radio de acción de las sanguijuelas de manera que haya una mayor diversidad de sanguijuelas. Además, que de esta forma vamos a poder concluir como afecta el radio de una sanguijuela sobre un mismo sistema. Los datos obtenidos se muestran en las figuras: *figure 3.a*, *figure 3.b*, *figure 3.c*

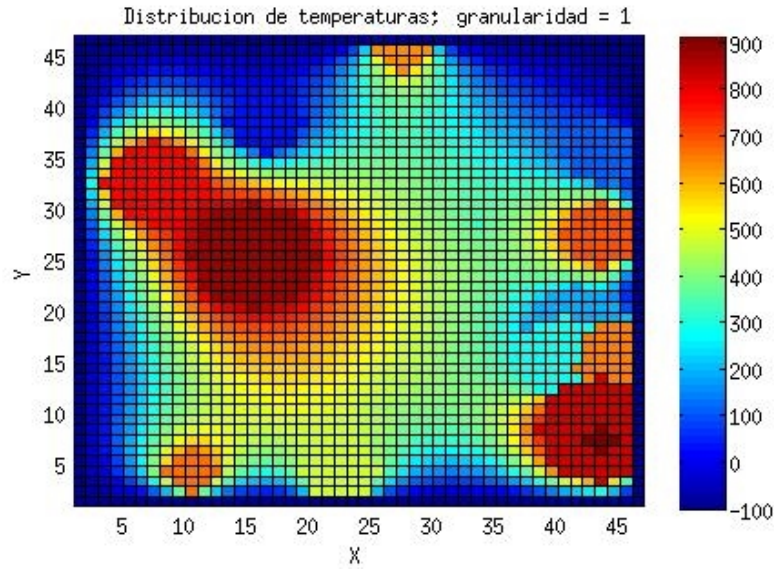


Figure 3.a: Representación de la distribución de temperaturas para un parabrisas con $a = 45$ metros; $b = 45$ metros; $h = 1$; $n = 15$. Donde el radio máximo de cada sanguijuela es 8 metros. Y la temperatura alcanzada en el punto crítico es de $644.83827^{\circ}\text{C}$.

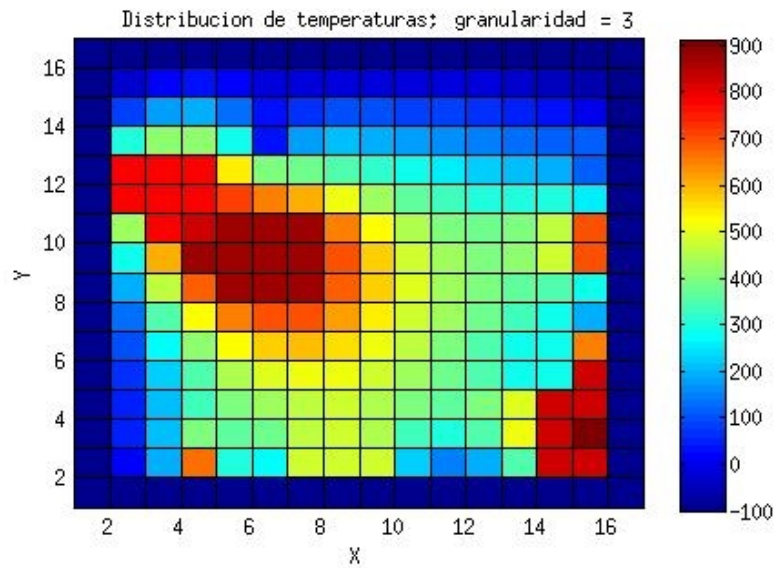


Figure 3.b: Mismas características que para el parabrisas representado en la *figure 3.a*. Pero, con granularidad igual a 3. Y la temperatura alcanzada en el punto crítico es de $633.08219^{\circ}\text{C}$.

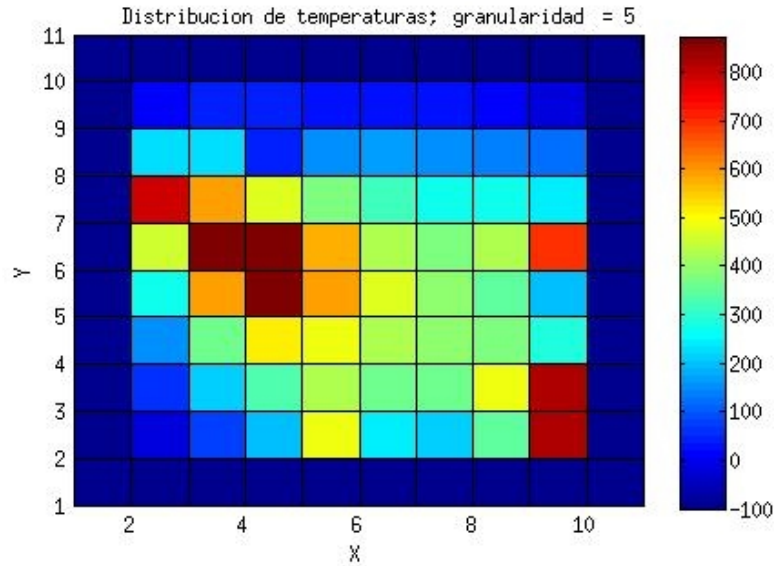


Figure 3.c: Mismas características que para el parabrisas representado en la *figure 3.a*. Pero, con granularidad igual a 5 y cuya temperatura en el punto crítico es igual a 515.79566°C .

4.1.4. Cuarto caso

Debido a que en los anteriores caso el valor del punto crítico dependía de uno o cuatro valores. Se decidió ver la última variante que existe, que es cuando un parabrisas tiene un lado par y el otro impar. De esta forma, el punto crítico depende del promedio entre dos valores. Este caso nos resulta útil para tratar de corroborar la veracidad de la hipótesis3. Para acercarnos a un caso promedio en el que no solo van a aparecer sanguijuelas con radio acción bajo, el mismo puede variar hasta los 10 metros. Los gráficos presentados a continuación reflejan los datos obtenidos.

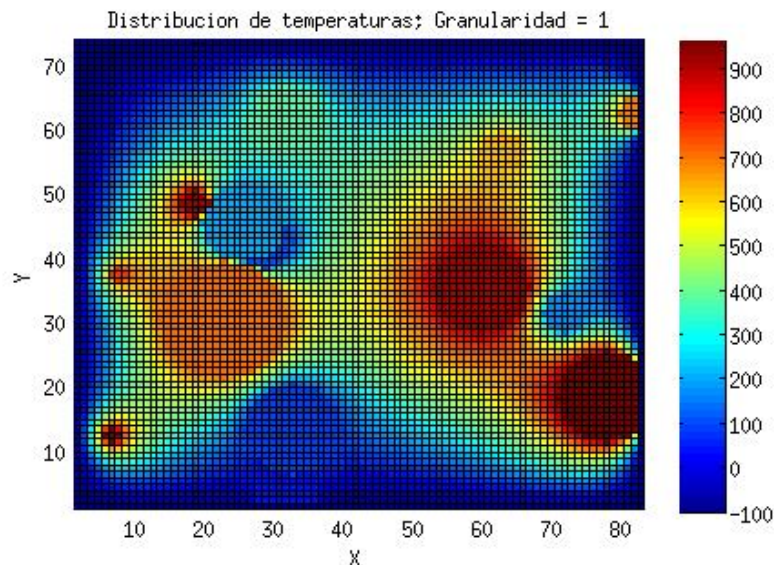


Figure 4.a: Representación de la distribución de temperaturas para un parabrisas con $a = 81$ metros; $b = 71$ metros; $h = 1$; $n = 15$. Donde el radio máximo de cada sanguijuela es de 10 metros. La temperatura en el punto crítico es de 533.86579°C .

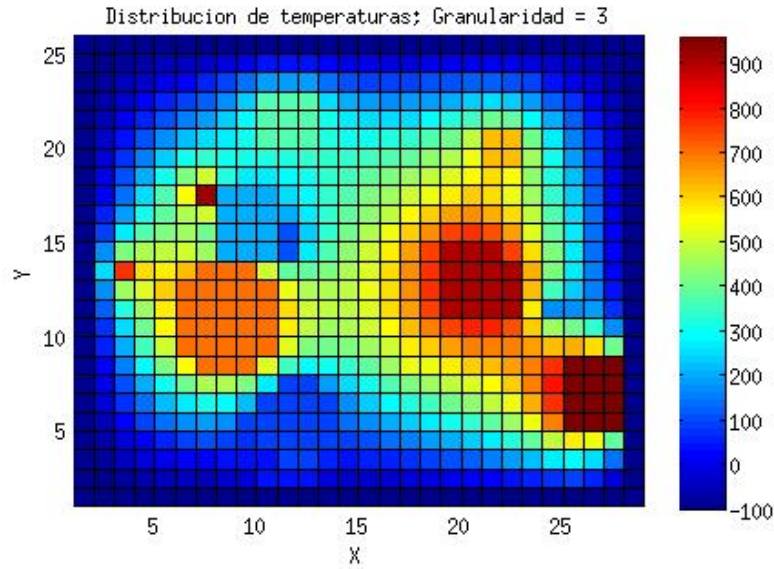


Figure 4.b: Mismas características que para el parabrisas representado en la *figure 4.a*. Pero, con granularidad igual a 3. Y la temperatura alcanzada en el punto crítico es de $495.18948^{\circ}\text{C}$.

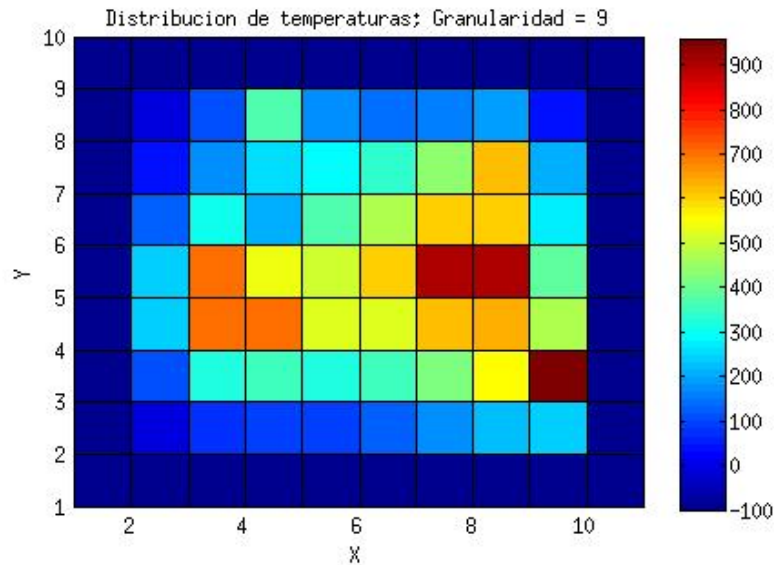


Figure 4.c: Mismas características que para el parabrisas representado en la *figure 4.a*. Pero, con granularidad igual a 5 y cuya temperatura en el punto crítico es de $552.85704^{\circ}\text{C}$.

4.2. Experimento 2

4.2.1. Experimento relación tiempo-calidad de cómputo

Las siguientes tres tablas representan los datos tomados de tres instancias del problema. Lo único que vamos a variar en dichas instancias son los radios de las sanguijuelas. Para las tres instancias $a = 20$, $b = 20$, la cantidad de sanguijuelas es 8 y las ubicaciones y temperaturas (variando en un rango de 50 a 730) de cada sanguijuela son las mismas (pueden encontrarse en los archivos

de Experimentos/Experimento2/instancias20x20).

Granularidad	Tiempo	Temp. pto crítico
5	3.60E-005	166.425
4	0.000105	177.267
2	0.002355	248.066
1	0.012867	230.499
0.8	0.035207	218.562
0.5	0.224236	227.963
0.4	0.542077	227.558
0.2	8.84272	224.521
0.1	145.142	223.948

Figura 2: En este conjunto de instancias los radios de las sanguijuelas están en el conjunto $\{2, 3, 4\}$. Podemos observar como efectivamente el tiempo crece cuando el parámetro h de granularidad disminuye. En cuanto a la temperatura del punto crítico, se puede ver como a medida que aumentamos el tamaño de nuestras discretizaciones, se va acercando a un valor cercano a 224 grados.

Granularidad	Tiempo	Temp. pto crítico
5	0.000107	-100
4	0.000246	-100
2	0.004086	-100
1	0.027411	89.5291
0.8	0.066003	53.1171
0.5	0.445484	83.7256
0.4	1.08866	68.3881
0.2	17.9335	96.5599
0.1	289.115	94.5064

Figura 3: En este otro experimento, hemos decidido achicar considerablemente los radios de las sanguijuelas de manera que estén en el rango $[0.03, 1]$. Por un lado, podemos ver que el tiempo sigue aumentando a medida que h disminuye al igual que en el experimento anterior. Sin embargo, se puede ver también que el algoritmo toma más tiempo que antes en resolver cada instancia. De todas formas, lo más llamativo parece encontrarse en las primeras tres entradas de la tabla, en los valores de temperatura del punto crítico. Pareciera como si las primeras tres discretizaciones de nuestro modelo del problema, aproximan tan mal al caso real que quedan descartadas todas las sanguijuelas. Sin embargo, al igual que en el experimento anterior, se cumple que a medida que aumenta el tamaño de la discretización, la temperatura en el punto crítico se acerca a un valor.

Granularidad	Tiempo	Temp. pto crítico
5	0.000111	-100
4	0.000239	-100
2	0.003144	-100
1	0.028953	89.5291
0.8	0.073902	-100
0.5	0.455657	67.3394
0.4	1.13656	-100
0.2	18.4805	45.0856
0.1	297.293	31.8193

Figura 4: Una vez más, volvemos a achicar los radios de las sanguijuelas. Vemos que el tiempo se sigue comportando de manera similar y con las primeras tres instancias del problema ocurre lo mismo que con el experimento anterior. Sin embargo, ahora sí está sucediendo algo que contradice una de las hipótesis (la tercera) planteada en el desarrollo ya que vemos que ahora, no pareciera que nos estemos acercando a un valor (el valor de la solución real). Esto se ve en las entradas 6 y 7 de la tabla, cuyas granularidades son 0.8 y 0.4, donde a pesar de que son más chicas que 1 (y por lo tanto las discretizaciones poseen tamaños más grandes), se está obteniendo -100 como temperatura del punto crítico, lo cual no parece estar bien.

Granularidad	Instancia con 100 sanguijuelas		Instancia con 1 sanguijuela	
	Tiempo	Temp. pto crítico	Tiempo	Temp. pto crítico
5	0.000156	443.358	0.000526	-56.6723
4	0.004226	443.358	0.00394	-100
2	0.013969	443.358	0.030393	-100
1	0.083721	443.358	0.4493	-74.7124
0.8	0.18614	443.358	1.11816	-100
0.5	1.02154	443.358	7.38949	-78.5975
0.4	2.51406	443.358	18.2381	-100
0.2	38.3409	443.358	299.151	-82.2174
0.1	577.083	443.358	4810.44	-84.2356

Figura 5: En este experimento se trabajó con dos instancias, esta vez ambas de 40x40 y con distintas sanguijuelas asociadas a una y a otra. Más específicamente y como indica la tabla, en una instancia hay 100 sanguijuelas y en la otra solo 1. Los radios de las 100 sanguijuelas se generaron con distribución uniforme en el rango $[25, 35]$ mientras que radio de la única sanguijuela de la segunda instancia es de 0.002. Vemos como estos resultados entran en conflicto con la hipótesis 2 ya que los tiempos medidos de la instancia con 1 sanguijuela son mucho más largos que con la instancia de 100 sanguijuelas. También podemos observar algo interesante acerca de la instancia con 100 sanguijuelas y esto es que, la temperatura en el punto crítico siempre fue la misma.

Granularidad	Instancia con 100 sanguijuelas		Instancia con 1 sanguijuela	
	Tiempo	Temp. pto crítico	Tiempo	Temp. pto crítico
5	0.000156	443.358	0.000369	350
4	0.004226	443.358	0.000782	350
2	0.013969	443.358	0.0107	350
1	0.083721	443.358	0.064883	350
0.8	0.18614	443.358	0.144703	350
0.5	1.02154	443.358	0.954458	350
0.4	2.51406	443.358	2.31257	350
0.2	38.3409	443.358	37.2852	350
0.1	577.083	443.358	596.912	350

Figura 6: Para hacer esta última medición, decidimos dejar intacta la instancia con 100 sanguijuelas y modificar la instancia que tiene solo una sanguijuela. La modificación que efectuamos fue aumentar considerablemente el radio de la sanguijuela a un valor de 40, siendo antes 0.002. Podemos observar que los tiempos son significativamente menores, parecidos a los que tomamos al trabajar con la instancia de 100 sanguijuelas e incluso en algunos casos menores. Es importante observar también, que la temperatura en el punto crítico ahora es siempre la misma (350) y antes (cuando el radio era 0.002) variaba y a veces la sanguijuela era descartada (para las granularidades 4, 2, 0.8 y 0.4 de la tabla anterior).

4.3. Experimento 3

4.3.1. Experimento para Hipótesis 1

Se modelaron 3 instancias, todas con los mismos valores de altura, ancho, granularidad, y cantidad de sanguijuelas. Lo único que se modificó en cada instancia fue el radio de estas últimas, principalmente con la idea de que:

- 1° Instancia = Todas sanguijuelas unitarias.
- 2° Instancia = Mitad sanguijuelas unitarias, mitad no unitarias.
- 3° Instancia = Todas sanguijuelas no unitarias.

La temperatura de las sanguijuelas no es relevante al experimento, ya que no nos agrega nada saber la temperatura del punto crítico al mismo, y por simplicidad las mismas fueron ubicadas en posiciones múltiplos de la granularidad

Los resultados obtenidos fueron los siguientes:

Algoritmo original			
Datos	Instancia 1	Instancia 2	Instancia 3
Entrada	"exp1.in"	"exp2.in"	"exp3.in"
N° de ejecuciones	10	10	10
Tiempo de ejecuciones	"MedicionesMetodo2.txt"	"MedicionesMetodo2b.txt"	"MedicionesMetodo2c.txt"
Media (en segundos)	38.78939	29.31231	26.0657
Desvío Standard	0.051020926100612	0.066167974881499	0.0047303277050597

Algoritmo modificado con Sherman-Morrison			
Datos	Instancia 1	Instancia 2	Instancia 3
Entrada	"exp1.in"	"exp2.in"	"exp3.in"
N° de ejecuciones	10	10	10
Tiempo de ejecuciones	"MedicionesMetodo3.txt"	"MedicionesMetodo3b.txt"	"MedicionesMetodo3c.txt"
Media (en segundos)	1.85237	15.23841	27.12823
Desvío Standard	0.0038183347157167	0.026859465744926	0.034835443151916

(Los archivos de entrada y los tiempos de corridas se puede encontrar en Experimentos/Experimento3).

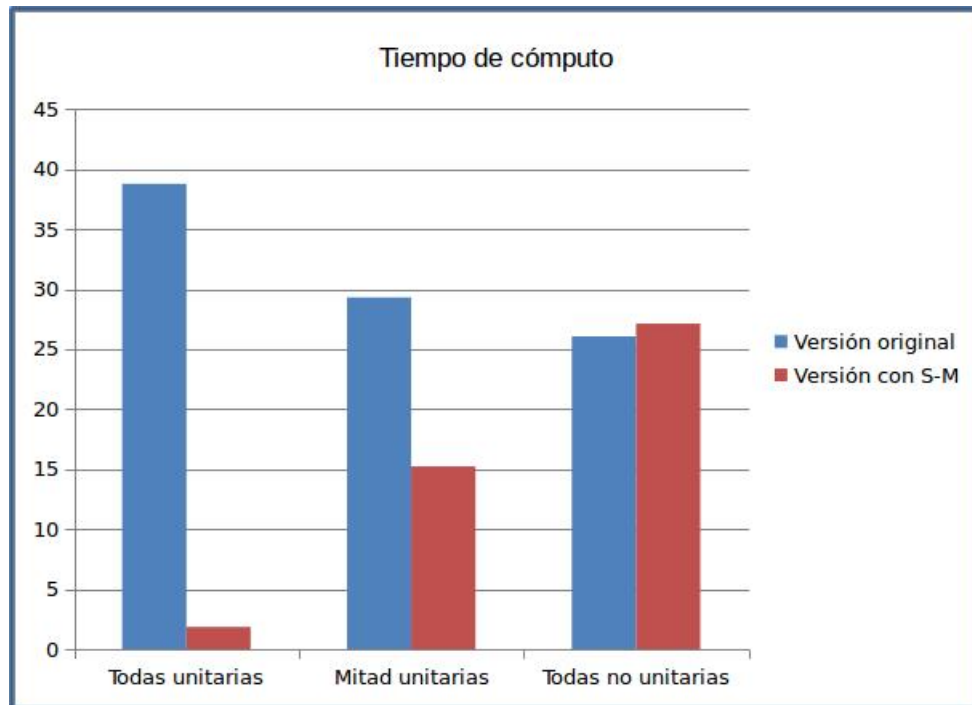


Figura 7: Gráfico obtenido de los valores de la media del tiempo de cada ejecución

4.3.2. Experimento para Hipótesis 2

Se modelaron 3 instancias, todas con los mismos valores de altura, ancho, cantidad de sanguijuelas, modificando la granularidad en cada una. El radio de las sanguijuelas se generó pseudo-aleatoriamente con MATLAB en un rango posible de 0.1 a 16. Se eligió este rango para que el experimento sea más representativo con las granularidades elegidas para experimentar. Por simplicidad, las mismas fueron ubicadas en posiciones múltiplos de 16. Las instancias son las siguientes:

- 1° Instancia = Granularidad 8
- 2° Instancia = Granularidad 4
- 3° Instancia = Granularidad 2

La temperatura de las sanguijuelas no es relevante al experimento, ya que no nos agrega nada saber la temperatura del punto crítico al mismo.

Los resultados obtenidos fueron los siguientes:

Datos	Instancia 1	Instancia 2	Instancia 3
Entrada	"exp4.in"	"exp5.in"	"exp6.in"
Nº de ejecuciones	10	10	10
Media Algoritmo original (en segundos)	0.0593982	0.9738903	17.79381
Media Algoritmo modificado con Sherman-Morrison (en segundos)	0.0409079	0.7462364	15.92311
Porcentaje de eficiencia de Sherman-Morrison	45,19%	30,50%	11,74%

(Los archivos de entrada y los tiempos de corridas se puede encontrar en Experimentos/Experimento3).

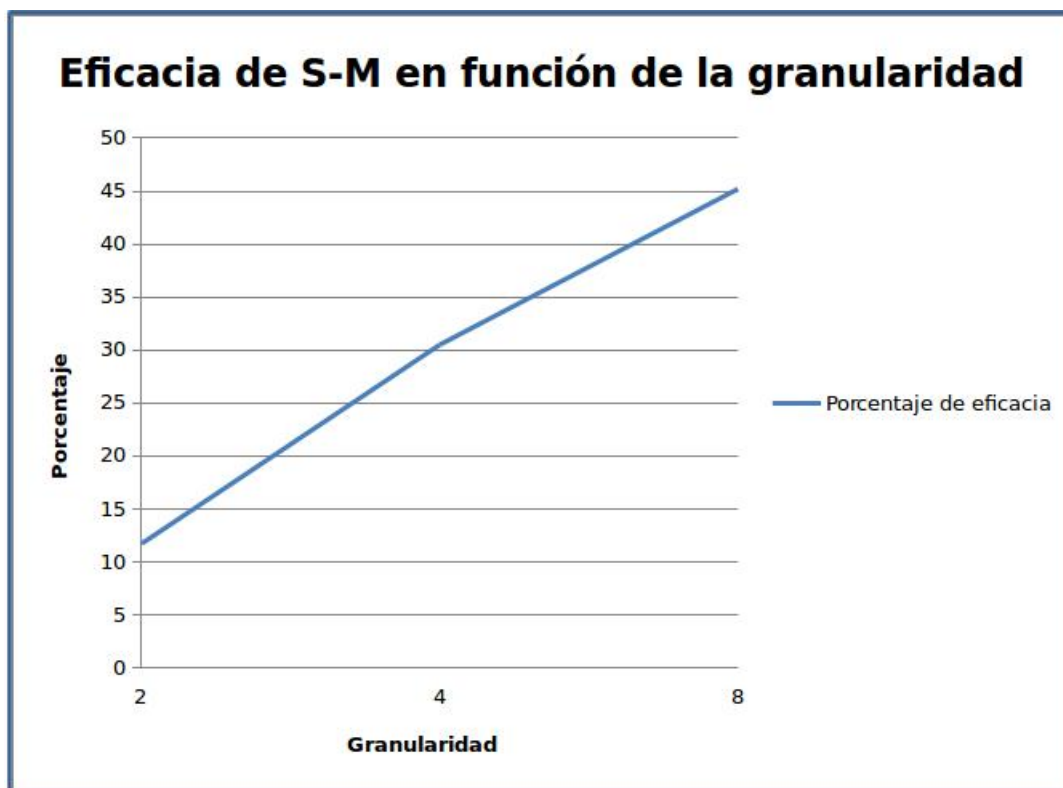


Figura 8: Gráfico obtenido de los valores del porcentaje de eficiencia del algoritmo con Sherman-Morrison comparándolo con el algoritmo original en función de la granularidad.

5. Discusión

5.1. Experimento 1

Hipótesis1: Apartir de los experimentos realizados podemos concluir que la Hipótesis1 es falsa. Debido a diversos motivos. En el caso número 1, variamos notablemente el valor de las granularidades con las cuales trabajamos, yendo desde el valor 2 hasta el 10. Y en todos los casos obtuvimos la misma temperatura en el punto crítico. Además como se puede observar en *figure 1.a*, *figure 1.b* y *figure 1.c*, *pág 18 experimento 1*, los puntos restantes poseen una distribución de temperaturas muy similar entre cada Gráfico. En este caso, la temperatura en el punto crítico se mantuvo constante entre cada representación. En el caso número 2, se varió la granularidad y los valores se modificaron (si bien representa un caso particular, donde las sanguijuelas son mayormente unitarias. sigue siendo un caso posible). Podemos observar que el valor del punto crítico en la *figure 2.d* es menor que en la de 2.c, la cual es menor a su vez que en la obtenida en la *figure 2.b* (entre cada gráfico fuimos disminuyendo el valor de la granularidad). Sin embargo, cuando probamos con $h = 0.5$, la temperatura en el punto crítico para este caso fue de $350.69248^{\circ}\text{C}$ (ver *figure 2.a*) siendo menor que la obtenida para una granularidad de valor 1 (*figure 2.b*). Falseando nuestra hipótesis. El origen de pensar en esta hipótesis se basó en la idea de que al discretizar el sistema por valores de h mas grandes obtenemos menos puntos y la distancia entre ellos es mayor. Así, si entre dos puntos A_{ij} , $A_{i+1,j}$ (de un parabrisas A, discretizado), existía una distancia de valor h' (valor de la granularidad). Y una sanguijuela actúa sobre la posición $i' j'$ con radio r (en el parabrisas sin discretizar) con $j = h' * j'$ e $i * h' < i' + r < (i + 1) * h'$. Entonces esa sanguijuela se omitiría (ya que no afecta a ningún punto de la discretización) obteniendo un sistema con una sanguijuela menos y por lo tanto un punto de calor menos. Que si se trabajase con un h'' , $h'' < h'$ e $i' * h'' \geq i' * r \vee i' * h'' \leq i' + r$. Al observar este comportamiento ahora fue necesario analizar el por qué de este hecho (agregar si ya hicieron este caso sino explicar).

Hipótesis2: Si bien apartir del experimento anterior podemos ver que no se cumple dicha hipótesis. No podemos dar por totalmente errónea a la misma. Sino, que llegamos a la conclusión de que depende el caso con el que se trabaje. Ya que a diferencia del caso 1. En los gráficos de los casos restantes (págs 19 - 24) se puede observar que varían las temperaturas para distintas granularidades. Aún más en la *figure 2.d* se detalla que el punto crítico (y se puede observar) tiene una temperatura de -100°C mientras que para el mismo sistema pero con otra discretización alcanza una temperatura de $350.69248^{\circ}\text{C}$ (ver *figure 2.a*). Siendo la mayor diferencia de temperatura que se registró entre los distintos casos (el por qué de este hecho lo desarrollaremos a lo largo de la hipótesis3 e hipótesis4). Dado que el único caso en el que no se produce un cambio en las temperaturas es el uno. Podemos concluir que se debe a las características propias que presenta el primer caso y que los restantes no. Como tratarse de sistema con sanguijuelas de distintos tamaños. También ser de la forma PARxPAR. Ya que de esta forma el punto crítico solo depende un solo punto de la discretización y es el central. Por lo que si hay una sanguijuela, cuyo radio afecta en este punto y es lo suficientemente grande para permanecer con las granularidades que se utilicen entonces siempre vamos a obtener la misma temperatura en el punto crítico.

Hipótesis3: en las páginas 19 - 23, Trabajamos con una parabrisas de forma IMPARxIMPAR, (segundo y tercer caso del experimento número uno) Por hipótesis, estos deberían ser los casos donde la temperatura del punto crítico varíe mas entre cada discretización utilizada. Y para el primer caso lo es, como ya se mencionó en este sistema se presentó la mayor diferencia. Sin embargo, para el mismo sistema con sanguijuelas de radio mayor (vease caso Pág 21) la temperatura cambió pero, no tan bruscamente, de hecho en porcentaje es muy similar al caso número 4 (pág 23). Donde las temperaturas en el punto crítico y en el resto, como puede verse, se mantienen muy similares. Entonces hay otro factor que influye además de que la matriz sea IMPARxIMPAR. Por lo que la hipótesis se podría mejorar diciendo que si se trabaja con una matrix IMPARxIMPAR o un lado

PAR y el otro IMPAR (y sanguijuelas de radio variado, hecho analizado en la hipótesis 4). Se van a obtener mayores diferencias que utilizando otras dimensiones con sanguijuelas de radio variado.

Hipótesis4: Como ya hicimos referencia anteriormente los cambios de temperatura más notable se presentaron en el caso 2. Y fue donde trabajamos con un predominio de sanguijuelas unitarias. Este cambio no solo se nota en el punto crítico sino que como se puede ver entre la *figure 2.a* y la *figure 2.c* varios puntos afectados por sanguijuelas se pierden entre cada experimentación, aquellos con temperaturas no mayor a los 400 °C, se pierden en la siguiente experimentación. Solo permanecen aquellos en los que el radio de acción es mayor a 0. Luego en la *figura 2.d* se perdieron todos los puntos sobre los cuales actuaban las sanguijuelas. De esta forma, y pese a que la granularidad no varió tanto entre los distintos experimentos; Como por ejemplo entre la *figure 4.a* y la *figure 4.d* donde la granularidad va de 1 a 9 o incluso el mismo sistema pero con sanguijuelas de mayor tamaño caso 3. Podemos concluir que la hipótesis (para estos casos) es verdadera. Más aún, podemos concluir en general que variar las granularidades puede llegar a afectar el resultado de la distribución de las temperaturas, según el tipo de dimensiones del parabrisas. Pero, que este no es el único factor de distorsión, sino que también está dado por el radio de las sanguijuelas, afectando en mayor medida si el mismo es de tamaño muy bajo.

5.2. Experimento 2

5.2.1. Hipótesis planteadas sobre calidad/tiempo de cómputo

En los experimentos realizados, vimos como efectivamente el tiempo de cómputo aumenta a medida que el h se achica, ya que el tamaño de la discretización es más grande y, como cada punto de la discretización representa una incógnita en nuestro sistema de ecuaciones, vamos a tener que resolver un sistema más grande. Por lo tanto, nuestra hipótesis 1. planteada en el Desarrollo de que el valor de h impacta de manera directa en el tiempo de cómputo es correcta y bien reflejada en nuestros experimentos.

Lo que no se ve reflejado en los experimentos, es lo que dicen las hipótesis 3 y 4. Como ambas están relacionadas, vamos a tratarlas de forma conjunta. En la sección de Resultados, podemos ver que las mediciones efectuadas en el primer conjunto de instancias (primera tabla) no entran en conflicto con lo que dicen estas hipótesis, pero sí las que se encuentran en la segunda y tercer tabla. Esto no es casualidad. La diferencia entre el primer conjunto de instancias y los otros dos, radica esencialmente en los radios de las sanguijuelas ya que como explicamos en los resultados, los radios se van reduciendo considerablemente. Sea S_i una sanguijuela cuyo radio es r_i , cuanto más chico sea r_i más chances existen de que S_i sea una sanguijuela unitaria o, en el peor caso, que sea descartada. Esto se debe a que si S_i cae justo en un punto de nuestra discretización, a menos que $r_i = 0$, van a haber más chances de que abarque solo ese punto. Pero si S_i **no cae** en un punto de nuestra discretización, como r_i es chico podría suceder que entonces r_i no alcance a **ningún** punto de nuestra discretización, descartando así a S_i . Y, como es de esperarse, el hecho de que S_i caiga o no en un punto de nuestra discretización, no depende de que tan chico o grande es h , si no de que las coordenadas de la posición de S_i sean múltiplos o no de h .

Con lo dicho anteriormente podemos afirmar entonces que, por un lado es cierto que cuanto más grande es nuestra discretización más nos acercamos a poder modelar el problema con mayor precisión en cuanto a la propagación del calor (una forma de convencerse de esto matemáticamente, es observar que en la ecuación del calor, $h \rightarrow 0$). Pero por otro lado, dado que nuestro problema tiene sanguijuelas con un radio variable que puede ser arbitrariamente chico (dentro de los límites de la computadora) y coordenadas reales, para llegar a una solución razonable es necesario tener en cuenta todas las sanguijuelas y esto no necesariamente depende del tamaño de h .

Faltaría hablar sobre lo que dice la hipótesis 2. Por los experimentos realizados con instancias de 40x40 (cuyos resultados se encuentran en las últimas dos tablas del experimento relación tiempo-calidad de cómputo), viendo la primera tabla del experimento podemos ver claramente que

la hipótesis no se cumple. Esto es así porque los tiempos tomados de la instancia con 100 sanguijuelas son significativamente menores a los tiempos tomados de la instancia con una sanguijuela. Pero esto tampoco significa que valga lo contrario, es decir que tampoco podemos afirmar que "si hay una menor cantidad de sanguijuelas entonces el tiempo de cómputo es mayor". De hecho, en la segunda tabla del experimento con instancias de 40x40, podemos observar que esto tampoco se cumple ya que ahora, los tiempos son muy similares para la instancia de 100 sanguijuelas y la de una sanguijuela. A este punto, es fácil ver que lo que está sucediendo es que, el tiempo de cómputo en realidad está variando en función de el radio de las sanguijuelas. Cuanto mayor es el radio de la sanguijuela, menor es el tiempo de cómputo. Es por esto que también, cuando el radio de una sanguijuela es grande y toca al punto crítico, como es poco probable que la sanguijuela sea descartada al variar la granularidad (debido al tamaño del radio), la temperatura en el punto crítico va a ser igual al valor de la temperatura de la sanguijuela y difícilmente va a variar. En cuanto al tiempo de cómputo, básicamente lo que sucede es que al tener sanguijuelas con radio grande, van a haber muchos puntos de nuestra discretización con una temperatura constante (proveniente de alguna sanguijuela actuando sobre ellos) y por lo tanto, los puntos cuya temperatura sea desconocida y tengamos que calcular van a ser unos pocos.

5.3. Experimento 3

Mediante la experimentación realizada, se logro verificar la hipótesis (1). Realmente, el tiempo de computo disminuye notablemente al usar el algoritmo modificado con Sherman-Morrison en casos donde haya sanguijuelas unitarias. Los resultados mostraron que cuanto más sanguijuelas unitaria haya, menor va a ser el tiempo de computo del mismo, comparado con el algoritmo original.

Esto significó un acercamiento práctico a lo expuesto teóricamente en la sección del desarrollo, es decir, que se pudo constatar la justificación teórica por la cual es mejor aplicar Sherman-Morrison. Un caso para resaltar, es cuando no hay ninguna sanguijuela unitaria, los 2 algoritmos tienen un tiempo de computo similar, hasta en algunos caso el algoritmo modificado con Sherman-Morrison tarda unos segundos más que el original, debido a que tiene más saltos condicionales y evalúa más expresiones.

En cuanto a la hipótesis (2), aunque asumimos predicados muy fuertes para que la experimentación sea representativa y no librada al azar (como que las sanguijuelas estén ubicadas en puntos múltiplos de la granularidad por ejemplo), se logro verificar lo que intuitivamente pensamos. Al disminuir la granularidad, la distancia entre 2 puntos cualesquiera se achica y la cantidad de puntos de la discretización aumenta. Esto implica que la probabilidad de encontrar sanguijuelas unitarias también disminuye, ya que es más probable, que al haber más puntos, cada sanguijuelas toque más de un punto del parabrisa.

6. Conclusiones

Muchos de los problemas de la vida real que quisieramos resolver por computadora son de naturaleza continua, esto significa que las herramientas matemáticas que se utilizan para tratarlos, trabajan con números reales. La computadora, al trabajar con aritmética finita, necesita que planteemos el problema a resolver de una manera discreta. En este trabajo, utilizar el método de aproximación por diferencias finitas, nos permitió modelar un problema continuo de manera discreta. De esta manera, pudimos adaptar la ecuación del calor a nuestra discretización del parabrisas, fijando distintos valores para el parámetro h .

Al realizar variaciones en el h pudimos observar como impacta en la calidad de los resultados. Pero, por si solo esta variación no actuaba como un factor determinante. Como se pudo observar en la experimentación 1. En algunos casos, aunque se variaba la granularidad de un sistema, la temperatura en el punto crítico seguía siendo la mismo. Concluyendo que un factor importante recaía además en las dimensiones del parabrisas. Ya que como se mencionó cuando el punto crítico dependía de más de un punto. Se presentaban variaciones entre la temperatura del punto crítico para distintas discretizaciones. Si bien, no podemos concluir que sí el punto crítico depende de solo uno entonces, no variará su temperatura (dada las limitaciones en la cantidad de experimentos desarrollados). Si podemos pensar que existe una mayor probabilidad de que esto suceda, que si trabajamos con dimensiones que requieren que nuestro punto crítico dependa de la temperatura de un promedio de puntos. A su vez, cuando se producían cambios, un factor influyente fue el radio de las sanguijuelas. Estas producían un cambio mas o menos bruscos, en la distribución de las temperaturas, a medida que se variaba su radio. Atribuyendole a granularidades mayor o igual que los radios promedios de las sanguijuelas, resultados peores en comparación a los casos contrarios. Además experimentamos obtener una relación adecuada entre el tiempo de cómputo y la calidad de la solución obtenida. Esta búsqueda no suele ser fácil, sobre todo cuando queremos evaluar la calidad de una solución, dado que desconocemos la solución real al problema, puede llegar a suceder que las soluciones varíen mucho según el tipo de granularidad elegida. Una de las cosas que pudimos averiguar mediante la experimentación, es que para las instancias del problema cuyos radios de las sanguijuelas sean muy pequeños, es probable que las soluciones varíen mucho dependiendo de la granularidad elegida. Esto no sucede si los radios de las sanguijuelas son grandes y, mucho menos si hay alguna sanguijuela que actúa de manera directa en el punto crítico. También pudimos ver que el tiempo de cómputo de los algoritmos propuestos para la resolución del problema, no está dominado por la cantidad de sanguijuelas de una instancia del mismo. Experimentos que quedaron pendientes que hubiera servido para aumentar la validez de estas afirmaciones consistirían en realizar comparaciones en los tiempos de cómputo de resolver instancias sin sanguijuelas contra instancias con sanguijuelas, variando nuevamente el radio y granularidad. Por ejemplo, una instancia sin sanguijuelas debería tardar más en resolverse que una instancia con sanguijuelas de radios grandes (en el contexto del tamaño de la discretización). Sin embargo, no deberían existir grandes diferencias entre el tiempo de cómputo consumido para resolver una instancia sin sanguijuelas y otra instancia con sanguijuelas de radios muy pequeños.

En materia a lo referido en las experimentaciones de eliminarSanguijuela, aprendimos lo importante que es abordar un tema con una base teórica fuerte, como por ejemplo con la utilización de la formula Sherman-Morrison. Dicha formula nos permitió aumentar la eficiencia del algoritmo eliminarSanguijuela más de lo que nos imaginábamos. Como consecuencia de esto, nos trajo curiosidad pensar ideas y variantes a este ecuación que sean aplicables al caso general. En un futuro nos gustaría poder desarrollarlas y experimentar para ver si es posible llevar la resolución del sistema al mínimo posible de complejidad temporal. La curiosidad sobre la modificación de dicha ecuación, también vino por parte de la experimentación que realizamos cambiando la granularidad, verificamos que al disminuir la misma, la distancia de los puntos de la discretización disminuye, haciendo bajar la probabilidad de encontrar sanguijuelas unitarias, en estos casos el algoritmo con la formula propuesta en el enunciado no era mucho más eficiente que el algoritmo original.

```
void SistemaBandas::EliminacionGaussiana0(char metodo)
{
    for(unsigned int j = 0; j < filas; j++)
    {
```

```
for (unsigned int i = j+1; (i <= j + ancho) && (i < filas); i++)
{
    if(Obtener(i,j) !=0)
    {
        double coeficiente = (Obtener(i,j)/Obtener(j,j));
        RestarFila(coeficiente, j, i, j+1, metodo);
    }
}
CerosAIzquierda(j,j);
}
```

7. Apéndices

7.1. A

Laboratorio de Métodos Numéricos - Primer Cuatrimestre 2015 Trabaja Práctico Número 1: “No creo que a él le gustará eso”

Introducción

El afamado Capitán Guybrush Threepwood se encuentra nuevamente en problemas. El parabrisas de su nave El Pepino Marino está siendo atacado simultáneamente por varios dispositivos hostiles vulgarmente conocidos como *sanguijuelas mutantes*. Estos artefactos se adhieren a los parabrisas de las naves y llevan a cabo su ataque aplicando altas temperaturas sobre la superficie, con el objetivo de debilitar la resistencia del mismo y permitir así un ataque más mortífero. Cada sanguijuela consta de una *sopapa de ataque* circular, que se adhiere al parabrisas y aplica una temperatura constante sobre todos los puntos del parabrisas en contacto con la sopapa.

Para contrarrestar estas acciones hostiles, el Capitán Guybrush Threepwood cuenta con el sistema de refrigeración de la nave, que puede aplicar una temperatura constante de -100°C a los bordes del parabrisas. El manual del usuario de la nave dice que si el punto central del parabrisas alcanza una temperatura de 235°C , el parabrisas no resiste la temperatura y se destruye. Llamamos a este punto el *punto crítico* del parabrisas.

En caso de que el sistema de refrigeración no sea suficiente para salvar el punto crítico, nuestro Capitán Guybrush Threepwood tiene todavía una posibilidad adicional: puede destruir alguna de las sanguijuelas. Es importante destacar que solo puede destruir una de ellas, ya que la eliminación de muchas sanguijuelas consumiría energía vital para la finalización de la misión, y llevaría únicamente al fracaso de la misma. La situación es desesperante, y nuestro héroe debe tomar una rápida determinación: debe decidir, cuando sea posible, qué sanguijuela eliminar de modo tal que el parabrisas resista hasta alcanzar la base más cercana.

El modelo

Suponemos que el parabrisas es una placa rectangular de a metros de ancho y b metros de altura. Llamemos $T(x, y)$ a la temperatura en el punto dado por las coordenadas (x, y) . En el estado estacionario, esta temperatura satisface la ecuación del calor:

$$\frac{\partial^2 T(x, y)}{\partial x^2} + \frac{\partial^2 T(x, y)}{\partial y^2} = 0. \quad (7)$$

La temperatura constante en los bordes queda definida por la siguiente ecuación:

$$T(x, y) = -100^{\circ}\text{C} \quad \text{si } x = 0, a \text{ ó } y = 0, b. \quad (8)$$

De forma análoga es posible fijar la temperatura en aquellos puntos cubiertos por una sanguijuela, considerando T_s a la temperatura ejercida por las mismas.

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 parabrisas, en función de los datos mencionados en esta sección.

Para estimar la temperatura computacionalmente, consideramos la siguiente discretización del parabrisas: sea $h \in \mathbb{R}$ la granularidad de la discretización, de forma tal que $a = m \times h$ y $b = n \times h$, con $n, m \in \mathbb{N}$, obteniendo así una grilla de $(n + 1) \times (m + 1)$ puntos. Luego, para $i = 0, 1, \dots, n$ y $j = 0, 1, \dots, m$, llamemos $t_{ij} = T(x_j, y_i)$ al valor (desconocido) de la función T en el punto $(x_j, y_i) = (jh, ih)$, donde el punto $(0, 0)$ se corresponde con el extremo inferior izquierdo del parabrisas. La aproximación por *diferencias finitas* de la ecuación del calor afirma que:

$$t_{ij} = \frac{t_{i-1,j} + t_{i+1,j} + t_{i,j-1} + t_{i,j+1}}{4}. \quad (9)$$

Es decir, la temperatura de cada punto de la grilla debe ser igual al promedio de las temperaturas de sus puntos vecinos en la grilla. Adicionalmente, conocemos la temperatura en los bordes, y los datos del problema permiten conocer la temperatura en los puntos que están en contacto con las sanguijuelas.

Enunciado

Se debe implementar un programa en C o C++ que tome como entrada los parámetros del problema (a , b , h , junto con las posiciones, radio y temperatura de las sanguijuelas) y calcule la temperatura en el parabrisas utilizando el modelo propuesto en la sección anterior.

Para resolver este problema, se deberá formular un sistema de ecuaciones lineales que permita calcular la temperatura en todos los puntos de la grilla que discretiza el parabrisas, e implementar el método de Eliminación Gaussiana (EG) para resolver este sistema particular. Dependiendo de la granularidad utilizada en la discretización, el sistema de ecuaciones resultante para este problema puede ser muy grande. Luego, es importante plantear el sistema de ecuaciones de forma tal que posea cierta estructura (i.e., una matriz banda), con el objetivo de explotar esta característica tanto desde la *complejidad espacial* como *temporal* del algoritmo. Además de la estructura banda, en determinados casos es posible utilizar la descomposición LU del sistema para acelerar el cómputo de evaluar qué sucedería si una sanguijuela fuese eliminada.

En función de la implementación, como mínimo se pide:

1. *Explotando la estructura*: Representar la matriz del sistema aprovechando la estructura banda de la misma, haciendo hincapié en la complejidad espacial. Realizar las modificaciones necesarias del algoritmo de EG clásico para que aproveche la estructura banda de la matriz, e implementarlo.
2. *Factorización LU*: Implementar un algoritmo que permita calcular la descomposición LU de la matriz, aprovechando la estructura banda de la misma. Además, se deben implementar los algoritmos de *forward/back substitution* para resolver el sistema utilizando la factorización LU. Discutir alternativas de implementación buscando minimizar el espacio utilizado.
3. *Última Esperanza*: Implementar un algoritmo para decidir si es posible salvar el parabrisas de la destrucción mediante la eliminación de una sanguijuela. En caso de ser posible, el algoritmo debe determinar la *mejor* sanguijuela a eliminar, (i.e., aquella que al ser removida genera la menor temperatura en el punto crítico).
4. *No hay tiempo que perder*: Existen casos donde la eliminación de una sanguijuela modifica *levemente* el sistema, alterando solo una *fila* de este. En estos casos, es posible utilizar la factorización LU y lo que se conoce como fórmula de Sherman–Morrison [1]:

$$(A + uv^t)^{-1} = A^{-1} - \frac{A^{-1}uv^tA^{-1}}{1 + v^tA^{-1}u}. \quad (10)$$

para acelerar el cómputo de evaluar la temperatura de eliminar una sanguijuela. Identificar estos casos, analizar cómo se modifica el sistema, e implementar una versión del algoritmo propuesto en 3 que explote esta propiedad cuando sea posible.

En función de la experimentación, como mínimo debe realizarse lo siguiente:

- Considerar al menos tres instancias originales de prueba, generando discretizaciones variando la granularidad para cada una de ellas y comparando el valor de la temperatura en el punto crítico. Se sugiere presentar gráficos de temperatura para los mismos, ya sea utilizando las herramientas provistas por la cátedra o implementando sus propias herramientas de visualización.
- Analizar el tiempo de cómputo requerido en función de la granularidad de la discretización, buscando un compromiso entre la calidad de la solución obtenida y el tiempo de cómputo requerido. Comparar los resultados obtenidos para alguna de las variantes propuestas en 1 y 2, y analizar ventajas y desventajas de ambos esquemas. ¿Cómo impacta en los resultados la elección del esquema 1 y 2? ¿Por qué?
- Estudiar el comportamiento del método propuesto para la estimación de la temperatura en el punto crítico y para la eliminación de sanguijuelas, comparando las variantes propuestas en 3 y 4.

Finalmente, se deberá presentar un informe que incluya una descripción detallada de los desarrollos teóricos propuestos, métodos implementados y las decisiones tomadas, incluyendo las estructuras utilizadas para representar la matriz banda y los experimentos realizados, junto con el correspondiente análisis y siguiendo las pautas definidas en el archivo `pautas.pdf`.

Opcional:

- Al aplicar Eliminación Gaussiana sobre el sistema asociado a este problema en particular, ¿Es necesario pivotar? ¿Por qué?
- ¿Cómo modificaría el algoritmo propuesto en 3 para aprovechar la fórmula de Sherman–Morrison [1] para cualquier tipo de sanguijuela?

Programa y formato de archivos

El programa debe tomar tres parámetros (y en ese orden): el archivo de entrada, el archivo de salida y el método a ejecutar, (*0*: Eliminación Gaussiana banda, *1*: Factorización LU explotando la propiedad banda y utilizando forward/back substitution, *2*: Algoritmo de eliminación de sanguijuela simple, *3*: Algoritmo de eliminación de sanguijuela mejorado, usando la fórmula de Sherman–Morrison cuando sea posible).

El archivo de entrada contiene los datos del problema (tamaño del parabrisas, ubicación, radio y temperatura de las sanguijuelas) desde un archivo de texto con el siguiente formato:

```
(a) (b) (h) (k)
(x1) (y1) (r1) (t1)
(x2) (y2) (r2) (t2)
...
(xk) (yk) (rk) (tk)
```

En esta descripción, $a \in \mathbb{R}_+$ y $b \in \mathbb{R}_+$ representan el ancho y largo en metros del parabrisas, respectivamente. De acuerdo con la descripción de la discretización del parabrisas, h es la longitud de cada intervalo de discretización, obteniendo como resultado una grilla de $n + 1 \in \mathbb{N}$ filas y $m + 1 \in \mathbb{N}$ columnas. Además, $k \in \mathbb{N}$ es la cantidad de sanguijuelas. Finalmente, para $i = 1, \dots, k$, el par (x_i, y_i) representa la ubicación de la i -ésima sanguijuela en el parabrisas, suponiendo que el punto $(0, 0)$ corresponde al extremo inferior izquierdo del mismo, mientras que $r_i \in \mathbb{R}_+$ representa el radio de la sopapa de ataque de la sanguijuela (en metros), y $t_i \in \mathbb{R}$ es la temperatura de dicha sopapa.

El archivo de salida contendrá los valores de la temperatura en cada punto de la discretización utilizando la información original del problema (es decir, antes de aplicar el método de remoción de sanguijuela), y será utilizado para realizar un testeo parcial de correctitud de la implementación. El formato del archivo de salida contendrá, una por línea, el indicador de cada posición de la grilla i, j junto con el correspondiente valor de temperatura. A modo de ejemplo, a continuación se muestran cómo se reportan los valores de temperatura para las posiciones $(3, 19)$, $(3, 20)$, $(4, 0)$ y $(4, 1)$.

```
...
3 19 -92.90878
3 20 -100.00000
4 0 -100.00000
4 1 60.03427
...
```

El programa debe ser compilado, ejecutado y testado utilizando los *scripts* de *python* que acompañan este informe. Estos permiten ejecutar los tests provistos por la cátedra, incluyendo la evaluación de los resultados obtenidos e informando si los mismos son correctos o no. Es requisito que el código entregado pase satisfactoriamente los casos de tests provistos para su posterior corrección. Junto con los archivos podrán encontrar un archivo `README` que explica la utilización de los mismos.

Sobre la entrega

- FORMATO ELECTRÓNICO: Jueves 09 de Abril de 2015, hasta las 23:59 hs., enviando el trabajo (informe + código) a `metnum.lab@gmail.com`. El asunto del email debe comenzar con el

texto [TP1] seguido de la lista de apellidos de los integrantes del grupo. Ejemplo: [TP1] Acevedo, Miranda, Montero

- FORMATO FÍSICO: Viernes 05 de Abril de 2015, de 17:30 a 18:00 hs.

Referencias

- [1] Golub, G. H. and Van Loan, C. F. Matrix Computations, 3rd ed. Baltimore, MD: Johns Hopkins, p. 51, 1996.

7.2. B

```
void SistemaBandas::EliminacionGaussiana0(char metodo)
{
    for(unsigned int j = 0; j < filas; j++)
    {
        for (unsigned int i = j+1; (i <= j + ancho) && (i < filas); i++)
        {
            if(Obtener(i,j) !=0)
            {
                double coeficiente = (Obtener(i,j)/Obtener(j,j));
                RestarFila(coeficiente, j, i, j+1, metodo);
            }
        }
        CerosAIzquierda(j,j);
    }
}

////////////////////////////////////

void SistemaBandas::RestarFila(double coeficiente, unsigned int primera,
unsigned int segunda, unsigned int desdeColumna, char metodo)
{
    int maxRestasPos = 2*ancho + 1;
    for(unsigned int i = desdeColumna, j = 0 ; i < filas && j <= maxRestasPos; i++, j++)
    {
        Modificar(segunda,i, Obtener(segunda,i) - (coeficiente*Obtener(primer,i)));
    }
    if(metodo == '0' || metodo == '2' )
    {
        Modificar(segunda, filas, Obtener(segunda,filas)
        - (coeficiente*Obtener(primer,filas)));
    }
}

////////////////////////////////////

void SistemaBandas::CerosAIzquierda(unsigned int fila, unsigned int hastaColumna)
{
    for(unsigned int i = 0; i < hastaColumna; i++)
        Modificar(fila,i, 0);
}

////////////////////////////////////

vector<double> SistemaBandas::BackWardSubstitution()
```



```

{
vector<double> result = vector<double>(filas);
    for (int i = filas -1; i>=0; i--)
    {
        result[i] = Obtener(i,filas);
        for (int k=i+1; k< filas && (k <= ( i + ancho)); k++)
        {
            result[i] = result[i] - Obtener(i,k)*result[k];
        }
        result[i] = result[i]/Obtener(i,i);
    }
    return result;
}

////////////////////////////////////

void SistemaBandas::LU1(char metodo)
{
    for(unsigned int j = 0; j < filas; j++)
    {
        for (unsigned int i = j+1; (i<= j + ancho) && (i < filas); i++)
        {
            double coeficiente = (Obtener(i,j)/Obtener(j,j));
            Modificar(i, j, coeficiente);
            if(Obtener(i,j) !=0) //para no hacer restas de mas
            {
                RestarFila(coeficiente, j, i, j+1, metodo);
            }
        }
    }
}

////////////////////////////////////

vector<double> Sherman_Morrison(vector<double> &nuevoB, vector<vector<double>
> &L, vector<vector<double> > &U, vector<double> &vt, vector<double> &u,
int f_v, unsigned int discrWidth)
{
    vector<double> y = ForWardSubstitution(nuevoB, L, discrWidth -1);

    vector<double> invAxB = BackWardSubstitution2(y, U);

    vector<double> z = ForWardSubstitution(u, L, discrWidth -1);

    vector<double> invAxu = BackWardSubstitution2(z, U);

    double k = 0;

    if (vt[0]!=0) {k += invAxB[(int)(f_v - (discrWidth -1))]*vt[0];}
    if(vt[1]!=0) {k += invAxB[f_v - 1]*vt[1];}
    k += invAxB[f_v]*vt[2];
    if(vt[3]!=0) {k += invAxB[f_v + 1]*vt[3];}
    if(vt[4]!=0) {k += invAxB[(int)(f_v + discrWidth -1)]*vt[4];};

    double l = 1;
    if (vt[0]!=0) {l += invAxu[(int)(f_v - (discrWidth -1))]*vt[0];}
    if(vt[1]!=0) {l +=invAxu[f_v - 1]*vt[1];}
    l += invAxu[f_v]*vt[2];

```

```
    if(vt[3]!=0) {l += invAxi[f_v +1 ]*vt[3];}
    if(vt[4]!=0) {l += invAxi[(int)(f_v + discrWidth -1)]*vt[4];};

    for(int i = 0; i < invAxi.size(); i++)
    {
        invAxi[i] = (k/l)*invAxi[i];
    }

    for(int i = 0; i < invAxi.size(); i++)
    {
        invAxB[i] = invAxB[i] - invAxi[i];
    }

    return invAxB;
}
```