



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

<b>1. Introducción Teórica</b>	<b>3</b>
1.1. Temperatura del Parabrisas . . . . .	3
1.2. Problemas . . . . .	3
1.3. Planteo del Problema . . . . .	4
<b>2. Desarrollo</b>	<b>4</b>
2.1. Estructura de la matriz bandas . . . . .	4
2.2. Eliminación Gaussiana vs Factorización LU . . . . .	5
2.3. Experimento: calidad/tiempo de cómputo . . . . .	6
<b>3. Resultados</b>	<b>7</b>
3.1. Experimento relación tiempo-calidad de cómputo . . . . .	7
<b>4. Discusión</b>	<b>10</b>
4.1. Hipótesis planteadas sobre calidad/tiempo de cómputo . . . . .	10
<b>5. Conclusiones</b>	<b>11</b>
<b>6. Apéndices</b>	<b>12</b>

## 1. Introducción Teórica

El parabrisas la nave del capitán Guybrush Threepwood 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 solamente con un sistema de refrigeración que aplica temperaturas de  $-100^{\circ}\text{C}$  a los bordes del parabrisas. El parabrisas se romperá si alcanza una temperatura mayor o igual a los  $235^{\circ}\text{C}$  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 algunas sanguijuelas, pero se desea que sea la menor cantidad posible, siempre y cuando el parabrisas siga en pie, pues dicha arma consume energía que es de vital importancia.

### 1.1. Temperatura del Parabrisas

Para calcular las temperaturas en el parabrisas se aplicará el siguiente criterio:

En los bordes, como se explicó anteriormente, la temperatura será de  $-100^{\circ}\text{C}$ , es decir sean  $x$  e  $y$  las coordenadas del parabrisas, y  $T(x, y)$  la función que devuelve la temperatura en un determinado punto, sea  $b$  el ancho y  $a$  el alto:

$$T(x, y) = -100^{\circ}\text{C} \quad \text{si} \quad x = 0 \quad \vee \quad x = b \quad \vee \quad y = 0 \quad \vee \quad y = a \quad (1)$$

La temperatura de los puntos que se encuentren dentro del perímetro de la sopapa circular de una sanguijuela será igual a la temperatura aplicada por dicha sanguijuela ( $T_s$ ).

$$T(x, y) = T_s \quad \text{si} \quad (x, y) \in \text{PuntosSanguijuela} \quad (2)$$

La temperatura en el resto de los puntos en el estado estacionario satisface la siguiente ecuación.

$$\frac{\delta^2 T(x, y)}{\delta x^2} + \frac{\delta^2 T(x, y)}{\delta y^2} = 0 \quad (3)$$

### 1.2. Problemas

En el siguiente trabajo veremos como la aritmética finita de las computadoras puede generar distintos problemas.

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, la cual haremos variar con el motivo de estudiar el comportamiento de nuestro sistema.

Otro problema que deberemos afrontar es que al trabajar en la búsqueda de soluciones de problemas que conllevan a la utilización de gran cantidad de operaciones matemáticas de punto flotante, en cada operación se puede perder cierta precisión, y la acumulación de estos errores puede escalar hasta llegar a una solución no satisfactoria. Esta pérdida de precisión se debe nuevamente a la limitación de las computadoras para representar números infinitos.

### 1.3. 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.

Al discretizar los puntos, la ecuación (3) para obtener temperaturas en cada punto del parabrisas continuo, se transforma en la siguiente ecuación por diferencias finitas.

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

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 están arriba, abajo, izquierda y derecha).

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 necesitarán 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 qué criterio utilizar para matar sanguijuelas, mediante distintas experimentaciones.

## 2. Desarrollo

### 2.1. 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  $(n+1) \times (m+1)$  puntos. Dado que hay  $(n+1) \times (m+1)$  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  $(n-1) \times (m-1)$  puntos (ya que hay dos bordes horizontales y dos verticales). Sabemos entonces, que si tenemos  $(n-1) \times (m-1)$  puntos a los cuales debemos asociarles temperaturas, vamos a tener  $(n-1) \times (m-1)$  incógnitas y también  $(n-1) \times (m-1)$  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  $(n-1) \times (m-1)$  filas debido a que cada fila va a representar cada punto de la discretización. Lo que sí se puede evitar, es tener  $(n-1) \times (m-1)$  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  $(n-1)x(2(m-1)+2)$  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  $A \in \mathbb{R}^{(n-1)x(m-1)}$  y sea  $A' \in \mathbb{R}^{(n-1)x(2(m-1)+2)}$  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.

## 2.2. 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  $O(t.n^3)$ . 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.

### 2.3. 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. Resultados

#### 3.1. 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 1: 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 2: 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 3: 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 4: 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 5: 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. Discusión

### 4.1. 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. 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 misma. 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 esta 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.

## 6. Apéndices

### 6.1. A

#### Laboratorio de Métodos Numéricos - Primer Cuatrimestre 2015 Trabajo 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  $-100^{\circ}\text{C}$  a los bordes del parabrisas. El manual del usuario de la nave dice que si el punto central del parabrisas alcanza una temperatura de  $235^{\circ}\text{C}$ , 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:

$$\frac{\partial^2 T(x, y)}{\partial x^2} + \frac{\partial^2 T(x, y)}{\partial y^2} = 0. \quad (5)$$

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

$$T(x, y) = -100^{\circ}\text{C} \quad \text{si } x = 0, a \text{ ó } y = 0, b. \quad (6)$$

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:

$$t_{ij} = \frac{t_{i-1,j} + t_{i+1,j} + t_{i,j-1} + t_{i,j+1}}{4}. \quad (7)$$

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]:

$$(A + uv^t)^{-1} = A^{-1} - \frac{A^{-1}uv^tA^{-1}}{1 + v^tA^{-1}u}. \quad (8)$$

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](mailto: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.