

# Universidad de Buenos Aires



## Trabajo práctico

*Sistema de ecuaciones lineales- Métodos iterativos*

### Alumnos

María Fernanda Pont Tovar # 104229

Romualdo Eva Edwin Osvaldo # 110131

**Fecha de entrega:** 04/10/2023

# Índice

|  |           |
|--|-----------|
| <b>1. Objetivos .....</b>                                      | <b>2</b>  |
| <b>2. Introducción .....</b>                                   | <b>3</b>  |
| <b>3. Desarrollo.....</b>                                      | <b>4</b>  |
| <b>3.1 Planteo del problema .....</b>                          | <b>4</b>  |
| <b>3.2 Método SOR .....</b>                                    | <b>6</b>  |
| <b>3.3 Parte I.....</b>  | <b>7</b>  |
| <b>3.3.a Numeración y estructura de la matriz .....</b>        | <b>7</b>  |
| <b>3.3.c Aplicaciones con <math>w</math> variable .....</b>    | <b>9</b>  |
| <b>3.3.d Evaluando el caso <math>N = 32</math>.....</b>        | <b>10</b> |
| <b>3.3.e Representación de <math>w</math>-iteraciones.....</b> | <b>11</b> |
| <b>3.3.f Análisis de datos.....</b>                            | <b>14</b> |
| <b>3.4 Parte II .....</b>                                      | <b>15</b> |
| <b>Anexo 1: Código .....</b>                                   | <b>17</b> |

## 1. Objetivos

- Experimentar con el uso de métodos iterativos para la resolución de sistemas de ecuaciones lineales resultantes de discretizaciones mediante diferencias finitas de ecuaciones diferenciales.
- Verificar experimentalmente los resultados teóricos y las estimaciones empíricas respecto de la velocidad de convergencia del proceso iterativo.
- Analizar el efecto de incrementar la discretización del problema.
- Efectuar un cálculo asociado a una eventual aplicación práctica.

## 2. Introducción

En el presente informe se presenta el abordaje de un problema de ingeniería habitual, en el cual se plantea encontrar la diferencia de temperaturas ubicadas en distintos puntos de una motherboard, generando un sistema de ecuaciones lineales. Dicho sistema es transformado a su forma matricial gracias al uso de librerías como *numpy* y matrices tipo *sparse*.

Muchos de estos métodos iterativos resultan lentos por la cantidad de operaciones que se deben realizar, presentando un gran tiempo de cómputo para la resolución del problema, por ello se busca el uso de un código que reduce la cantidad de procesos que una maquina debe realizar.

El uso del método *SOR* resulta ágil para resolver el problema planteado, donde se presentan las diferencias que existen en el método a medida que se escogen distintos *factores de sobre-relajación* ( $w$ ).

### 3. Desarrollo

#### 3.1 Planteo del problema

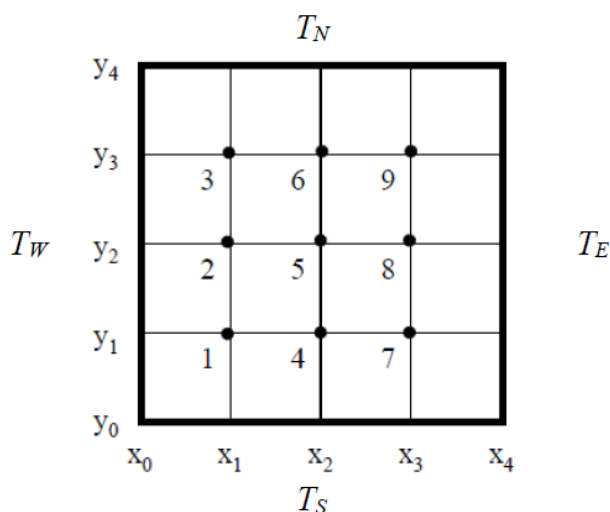
El problema se basa en el cálculo de las temperaturas de una *motherboard*, donde se asume que las temperaturas de los bordes son una constante conocida sin margen de error, se busca entonces calcular distintas temperaturas que sufre la tarjeta madre en ubicaciones distintas, a estas posiciones las llamamos nodos. Las temperaturas de los nodos pueden calcularse según

$$4T_{ij} - T_{(i-1)j} - T_{(i+1)j} - T_{i(j-1)} - T_{i(j+1)} = 0$$

La resolución de la ecuación, evaluada en distintas coordenadas, muestra las distintas temperaturas de la *motherboard*, formando un sistema de ecuaciones lineales cuyo tamaño depende de la cantidad de puntos, dicho de otro modo, de la cantidad de mediciones de temperatura en la que se quiera dividir la placa.

**Figura 1**

Representación de la *motherboard* para  $N = 4$



*Nota.* Sea  $A$  la matriz que se forma a partir del sistema de ecuaciones, entonces decimos que la matriz  $A \in (N - 1) \times (N - 1)$ , siendo esta una matriz cuadrada.

Para identificar los nodos, se plantea la existencia de dos variables finitas  $(i, j)$  donde  $i$  representa las posiciones de los nodos sobre el eje  $x$  de derecha a izquierda, por su parte  $j$  representa las posiciones en el eje  $y$  de abajo hacia arriba.

## Figura 2

*Representación de las posiciones de los distintos nodos.*

$$\begin{bmatrix} (1, \mu) & (2, \mu) & \cdots & (\mu, \mu) \\ \vdots & \vdots & \cdots & \vdots \\ (1, 2) & (2, 2) & \ddots & (2, \mu) \\ (1, 1) & (2, 1) & \cdots & (1, \mu) \end{bmatrix}$$

*Nota.* Denominamos  $\mu = N - 1$

A cada uno de los  $(i, j)$  representados se les aplica la ecuación de temperaturas presentada anteriormente, se nota que existen posiciones donde los nodos pueden resultar cero o iguales a  $N$ , en estos casos a esa posición de temperatura se le asigna la temperatura del lado que corresponde la posición, así se asigna  $T_N$  cuando la posición de  $j=N$ ,  $T_S$  cuando la posición de  $j=0$ ,  $T_E$  cuando la posición de  $i=N$ , y se usa  $T_W$  cuando  $i=0$ .

Se construye de esta forma el sistema de ecuaciones lineales, el cual puede ser representado de forma matricial por  $A$ . Se nota a su vez que cada ecuación dependerá de cuatro temperaturas, que son las que rodean a un nodo.

El vector  $b$  está formado por las temperaturas conocidas  $(T_N, T_S, T_E, T_W)$  de aquellas ecuaciones en las que aparezcan.

### Figura 3

*Forma matricial de la matriz.*

$$A_N = \begin{bmatrix} T & -I & O & \dots & O \\ -I & T & -I & \dots & O \\ O & -I & T & \dots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & O & -I & T \end{bmatrix} \quad T = \begin{bmatrix} 4 & -1 & O & \dots & O \\ -1 & 4 & -1 & \dots & O \\ O & -1 & 4 & \dots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & O & -1 & 4 \end{bmatrix}, \quad -I^{n \times n}, \quad O^{n \times n}$$

*Nota.* Se denota que la matriz  $A$  está compuesta por submatrices  $T, I, O$ , las cuales siguen un patrón, entonces se puede construir una matriz  $A$  para cualquier  $N$  con esta agrupación.

### 3.2 Método SOR

La resolución del problema presentado se realiza con el método iterativo de SOR (Successive Over Relaxation), para ello se plantea el uso de distintos factores de sobre-relajación ( $w$ ) y una semilla para iniciar el proceso iterativo.

Se denota entonces que la matriz una descomposición tal que  $A = D + L + U$ , donde  $D$  es una matriz diagonal,  $L$  es la matriz triangular inferior (Lower) y  $U$  es la matriz diagonal superior (Upper). se puede resolver el sistema de ecuaciones lineales que se obtiene al multiplicar cada lado de la iteración por  $(D + wL)$  a la izquierda:

$$(D + wL)x = wb - [wU + (w - 1)D]x \quad \forall w, 1 < w < 2$$

Recordemos que  $(D + wL)$  es triangular inferior, entonces podemos tomar provecho de la sustitución hacia adelante. Obtenemos una expresión de índice de  $x$  en la iteración  $k+1$  a partir de valores de la iteración  $k$  y de la iteración  $k+1$  ya calculados.

$$x_i^{(k+1)} = (1 - w)x_i^{(k)} + \frac{w}{a_{ii}}(b_i - \sum_{j < i} a_{ij}x_j^{(k+1)} - \sum_{j > i} a_{ij}x_j^{(k)})$$

Siendo este el proceso que utiliza el programa para calcular los distintos valores de temperatura en los nodos de la *motherboard*.

### 3.3 Parte I

#### 3.3.a Numeración y estructura de la matriz

Se propone entonces construir la forma que adquiere la matriz para un  $N$  genérico, sabiendo que la matriz construida siempre presenta un tamaño de  $(N - 1) \times (N - 1)$

De esta forma se evidencia que la estructura de la matriz  $A$  está dada por un patrón de submatrices repetidas de manera uniforme, siendo producto de las ecuaciones  $E_{ij}$  cuya estructura se refleja en cada fila.

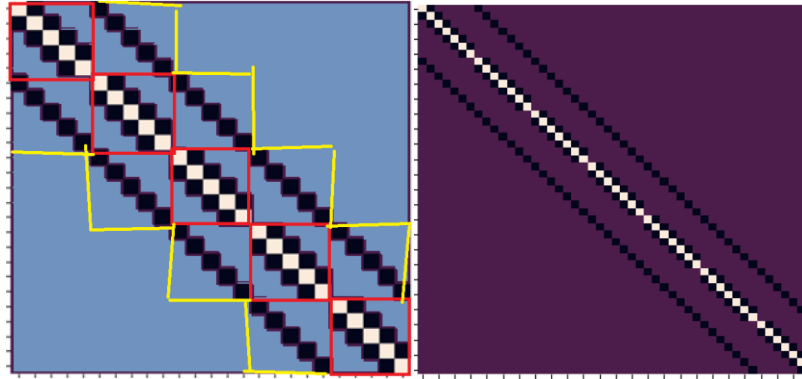
$$\left\{ \begin{array}{l} 4T_{1,1} - T_{2,1} - T_{1,2} = T_W + T_S \\ \vdots \\ 4T_{1,3} - T_{2,3} - T_{1,2} = T_W + T_N \\ \vdots \\ 4T_{N-1,N-1} - T_{N-2,N-1} - T_{N-1,N-2} = T_N + T_E \end{array} \right.$$

Cada ecuación corresponde a un nodo de la grilla y por cada nodo vecino, el valor menos uno ( $-1$ ) va en la columna correspondiente. La diagonal obtiene siempre el valor de cuatro ( $4$ ), reflejando que en todas las ecuaciones se multiplica el nodo actual por cuatro para calcular el promedio. (Ver figura 4)



## Figura 4

*Representación compacta de la estructura matricial para  $N = 5$  y  $N = 7$ .*



*Nota.* De color rojo se blanco equivale al valor cuatro (4), el negro menos uno (-1) y cero (0) para los demás

Se menciona que un cambio en la numeración de los nodos afecta la estructura de la matriz resultante, pero las temperaturas en cada posición no se ven alteradas. Esto sucede pues el orden de las ecuaciones en el sistema lineal cambia de orden

### 3.3.b Programa

Se escribió el programa utilizando el lenguaje Python y la plataforma Colab de Google. Se muestran las principales funciones del programa.

```
def solve_for_x_until(b, A, seed, until=1, omega=1.5)
def generate_matrix(N)
def generate_b(N, cardinal_temperatures)
```

`solve_for_x_until` Dado un vector  $b$ , una matriz  $A$ , una tolerancia “Until” y un valor  $w$  mayor a uno, devuelve un *tuple* con la última iteración, las iteraciones realizadas hasta cumplir con la tolerancia, y un vector de los errores entre cada par de iteraciones. La tolerancia *until* se mide calculando la distancia euclídea entre dos iteraciones (*numpy.norm*)

`generate_matrix` y `generate_b` Generan la matriz y el vector  $b$  en el sistema  $Ax = b$  para el problema dado utilizando la matriz *sparse* tipo “*lil\_matrix*”

### 3.3.c Aplicaciones con $w$ variable

Se realizó con parámetros de temperaturas de los bordes iguales a 1 y semilla (0 0 0 0 0 0 0 0), mientras que  $w$  se evaluó en distintos valores con el fin de encontrar el  $w_{optimo}$ .

Se opta por usar un error relativo, realizadas las iteraciones se demuestra que para el valor de tolerancia  $R = 0.01$ , los resultados resultan muy similares al variar  $w$ , por ello trabajamos también con una tolerancia  $R' = 0.0001$  con el fin de representar de manera adecuada los datos.

| <b><math>R=0.01</math></b> |                           |                             |
|----------------------------|---------------------------|-----------------------------|
| <b><math>w</math></b>      | <b><i>Iteraciones</i></b> | <b><math>p(Tsor)</math></b> |
| 1                          | 5                         | 0.3981072                   |
| 1.05                       | 5                         | 0.3981072                   |
| 1.1                        | 5                         | 0.3981072                   |
| 1.15                       | 5                         | 0.3981072                   |
| 1.2                        | 4                         | 0.3162278                   |
| 1.25                       | 4                         | 0.3162278                   |
| 1.3                        | 4                         | 0.3162278                   |
| 1.35                       | 4                         | 0.3162278                   |
| 1.4                        | 4                         | 0.3162278                   |
| 1.45                       | 4                         | 0.3162278                   |
| 1.5                        | 4                         | 0.3162278                   |
| 1.55                       | 5                         | 0.3981072                   |
| 1.6                        | 6                         | 0.4641589                   |
| 1.65                       | 7                         | 0.5179475                   |
| 1.7                        | 8                         | 0.5623413                   |
| 1.75                       | 11                        | 0.6579332                   |
| 1.8                        | 12                        | 0.6812921                   |
| 1.85                       | 13                        | 0.7017038                   |
| 1.9                        | 23                        | 0.8185467                   |
| 1.95                       | 44                        | 0.900628                    |

| <b><math>R=0.00001</math></b> |                           |                             |
|-------------------------------|---------------------------|-----------------------------|
| <b><math>w</math></b>         | <b><i>Iteraciones</i></b> | <b><math>p(Tsor)</math></b> |
| 1                             | 13                        | 0.70170383                  |
| 1.05                          | 12                        | 0.68129207                  |
| 1.1                           | 11                        | 0.65793322                  |
| 1.15                          | 10                        | 0.63095734                  |
| 1.2                           | 9                         | 0.59948425                  |
| 1.25                          | 8                         | 0.56234133                  |
| 1.3                           | 8                         | 0.56234133                  |
| 1.35                          | 9                         | 0.59948425                  |
| 1.4                           | 9                         | 0.59948425                  |
| 1.45                          | 10                        | 0.63095734                  |
| 1.5                           | 11                        | 0.65793322                  |
| 1.55                          | 11                        | 0.65793322                  |
| 1.6                           | 12                        | 0.68129207                  |
| 1.65                          | 15                        | 0.73564225                  |
| 1.7                           | 18                        | 0.77426368                  |
| 1.75                          | 22                        | 0.81113083                  |
| 1.8                           | 27                        | 0.84319093                  |
| 1.85                          | 36                        | 0.87992254                  |
| 1.9                           | 56                        | 0.92105532                  |
| 1.95                          | 113                       | 0.96006555                  |

### 3.3.d Evaluando el caso $N = 32$

Con estas consideraciones da como resultado una matriz  $A^{31 \times 31}$ , siendo esta una rala, por ello es necesario evitar que se realice un trabajo de cómputo con los ceros contenidos en la matriz. Como se utiliza una matriz *sparse* en la etapa de generación de  $A$ , no se recurre al uso excesivo de memoria.

Se optaron por usar las mismas condiciones de semilla que en el caso anterior, siendo necesario también modificar la tolerancia para lograr evidenciar de manera correcta las diferencias en las iteraciones cuando varía el  $w$ .

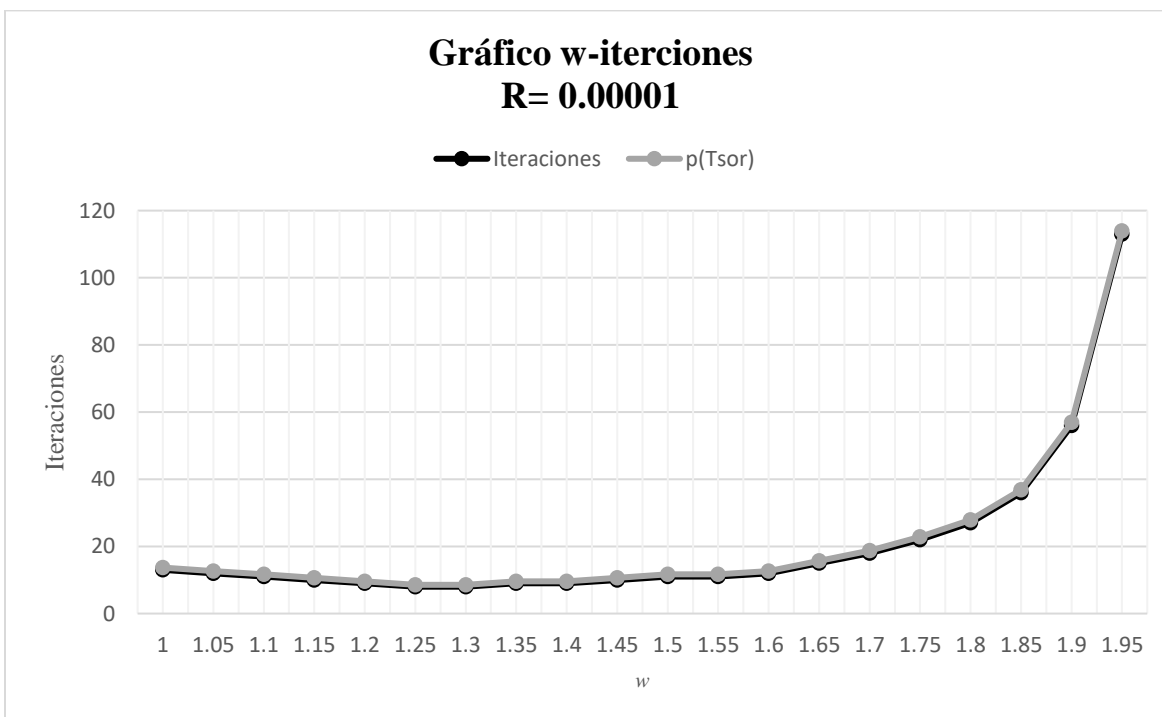
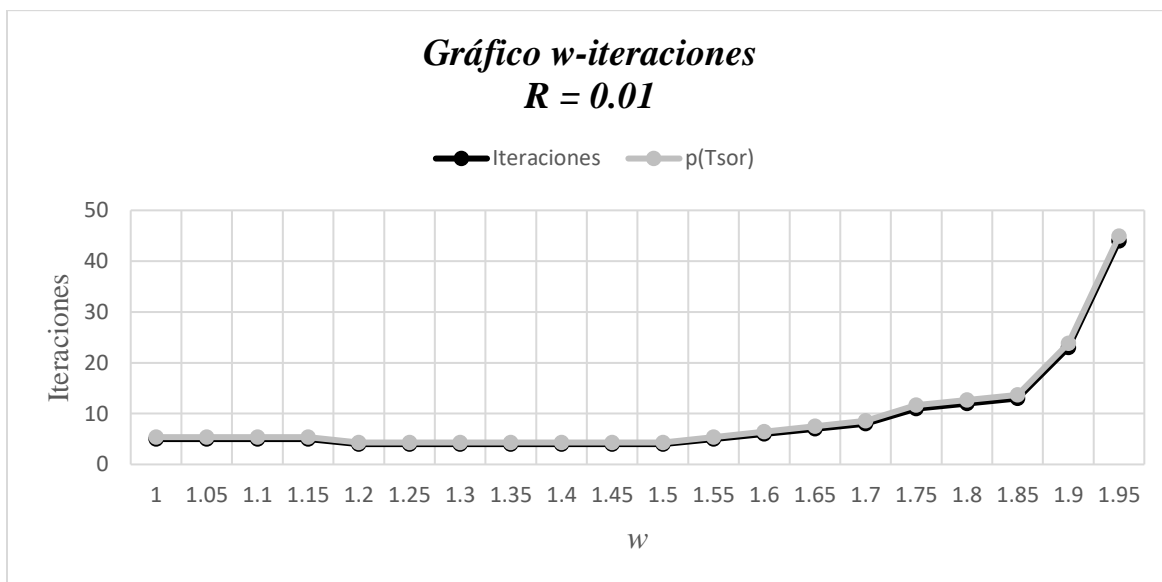
| <b><math>R=0.01</math></b> |                           |                             |
|----------------------------|---------------------------|-----------------------------|
| <b><math>w</math></b>      | <b><i>Iteraciones</i></b> | <b><math>p(Tsor)</math></b> |
| 1                          | 6                         | 0.4641589                   |
| 1.05                       | 6                         | 0.4641589                   |
| 1.1                        | 6                         | 0.4641589                   |
| 1.15                       | 6                         | 0.4641589                   |
| 1.2                        | 6                         | 0.4641589                   |
| 1.25                       | 6                         | 0.4641589                   |
| 1.3                        | 6                         | 0.4641589                   |
| 1.35                       | 6                         | 0.4641589                   |
| 1.4                        | 6                         | 0.4641589                   |
| 1.45                       | 6                         | 0.4641589                   |
| 1.5                        | 6                         | 0.4641589                   |
| 1.55                       | 6                         | 0.4641589                   |
| 1.6                        | 6                         | 0.4641589                   |
| 1.65                       | 6                         | 0.4641589                   |
| 1.7                        | 7                         | 0.5179475                   |
| 1.75                       | 7                         | 0.5179475                   |
| 1.8                        | 8                         | 0.5623413                   |
| 1.85                       | 9                         | 0.5994843                   |
| 1.9                        | 11                        | 0.6579332                   |
| 1.95                       | 20                        | 0.7943282                   |

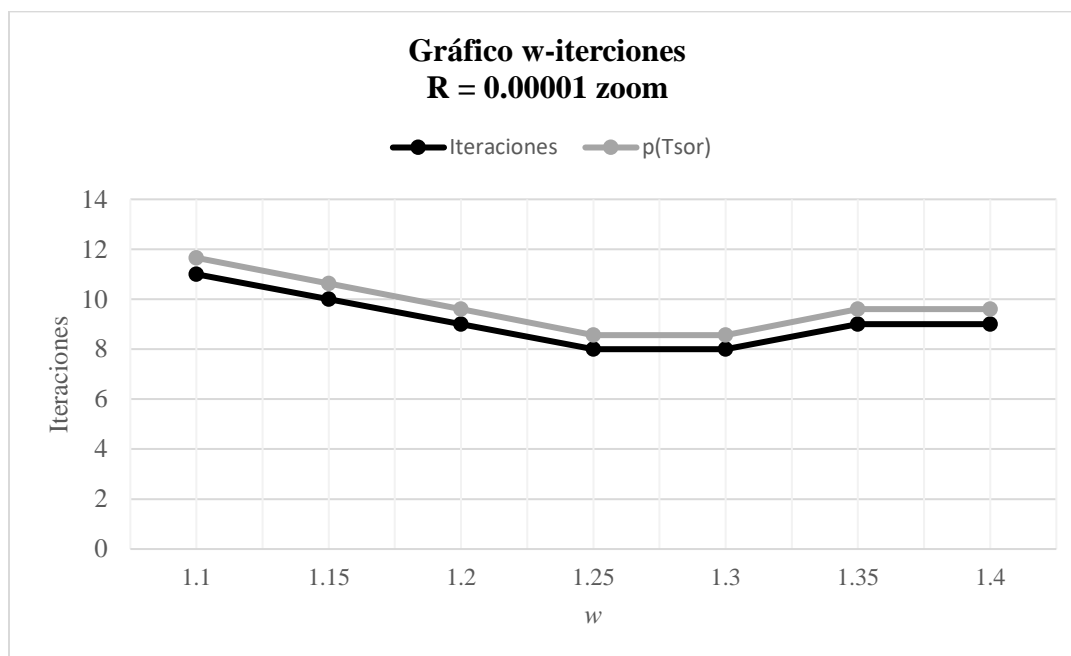
| <b><math>R=0.00001</math></b> |                           |                             |
|-------------------------------|---------------------------|-----------------------------|
| <b><math>w</math></b>         | <b><i>Iteraciones</i></b> | <b><math>p(Tsor)</math></b> |
| 1                             | 44                        | 0.900628                    |
| 1.05                          | 44                        | 0.900628                    |
| 1.1                           | 44                        | 0.900628                    |
| 1.15                          | 44                        | 0.900628                    |
| 1.2                           | 44                        | 0.900628                    |
| 1.25                          | 44                        | 0.900628                    |
| 1.3                           | 43                        | 0.8984385                   |
| 1.35                          | 43                        | 0.8984385                   |
| 1.4                           | 42                        | 0.8961505                   |
| 1.45                          | 41                        | 0.8937571                   |
| 1.5                           | 40                        | 0.8912509                   |
| 1.55                          | 38                        | 0.8858668                   |
| 1.6                           | 37                        | 0.88297                     |
| 1.65                          | 35                        | 0.8767124                   |
| 1.7                           | 33                        | 0.869749                    |
| 1.75                          | 30                        | 0.8576959                   |
| 1.8                           | 28                        | 0.8483429                   |
| 1.85                          | 30                        | 0.8576959                   |
| 1.9                           | 44                        | 0.900628                    |
| 1.95                          | 69                        | 0.9354369                   |

### 3.3.e Representación de $w$ -iteraciones

**Figura 5**

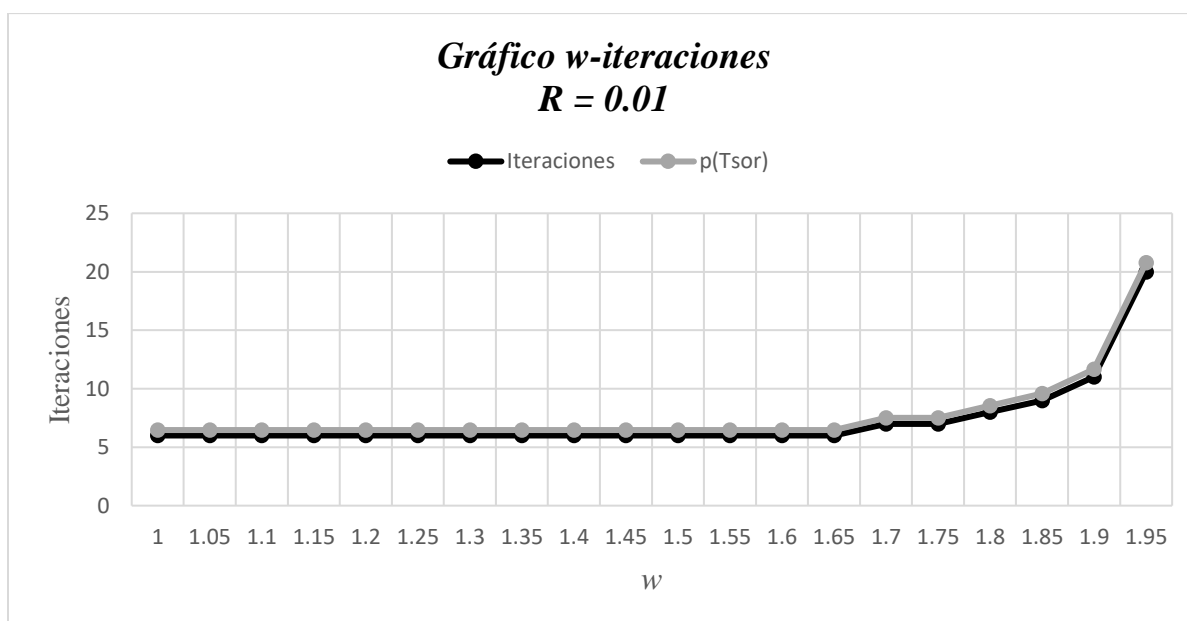
Gráficos para  $N = 4$

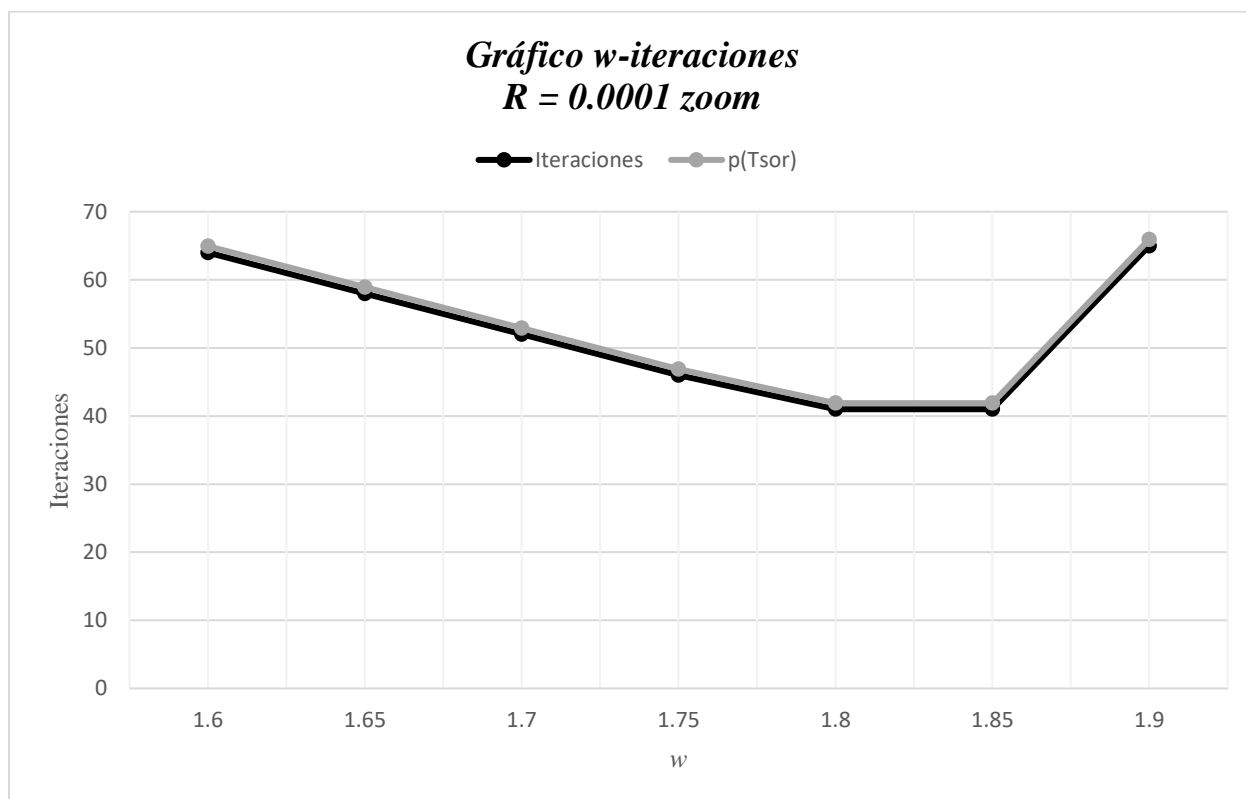
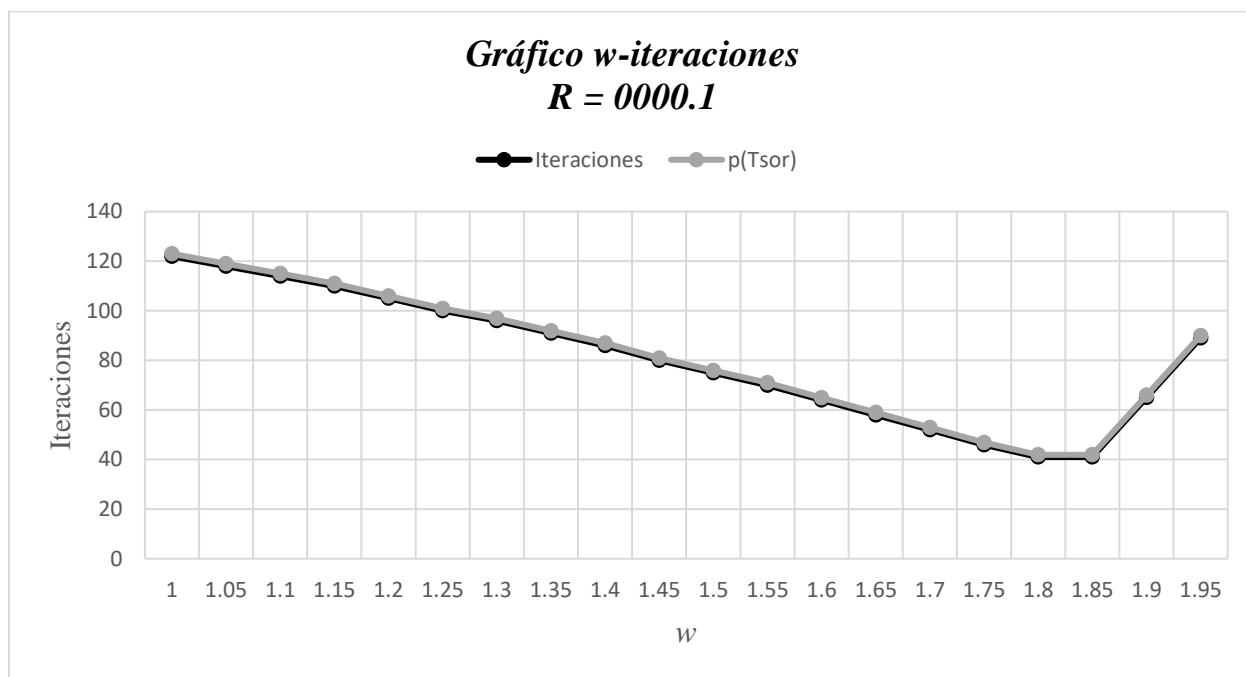




**Figura 6**

*Gráficos para  $N = 32$*





### 3.3.f Análisis de datos

Veamos que el  $w_{optimo}$  teórico puede ser calculado según  $w_{optimo}^t = \frac{2}{1+\sin(\pi/N)}$ , dando como resultado para  $N = 4$  un  $w_{optimo}^t = 1.17$ , mientras que el experimental se encuentra en un rango de  $1.25 < w_{optimo}^{exp} < 1.30$ .

Considerando  $N = 32$  se obtiene un resultado de  $w_{optimo}^t = 1.82$ , resultado coherente pues el experimental se encuentra en un rango de  $1.80 < w_{optimo}^{exp} < 1.85$ .

Si el  $\rho(T_{SOR}) = w_{optimo} - 1$ , los resultados para  $N = 4$  resultan con una desviación grande pues el valor teórico es 0.17 y la aproximación calculada mediante  $\rho(T_{SOR}) \approx R^{1/n}$  resulta ser 0.56, siendo menores que  $\rho(T_{GS}) = \cos^2\left(\frac{\pi}{N}\right) = 0.5$ .

Para el caso de  $N = 32$  se obtuvieron datos experimentales de 0.89, mientras que con el teórico resulta 0.82, valor que se asemeja, siendo además menores que  $\rho(T_{GS})$ .

Se concluye de esta forma que los valores de  $w_{optimo}$  y  $\rho(T_{SOR})$  se asemejan a los valores teóricos mientras más crece de tamaño el número  $N$ , siendo menores que el  $\rho(T_{GS})$  sin importar el tamaño de la matriz.

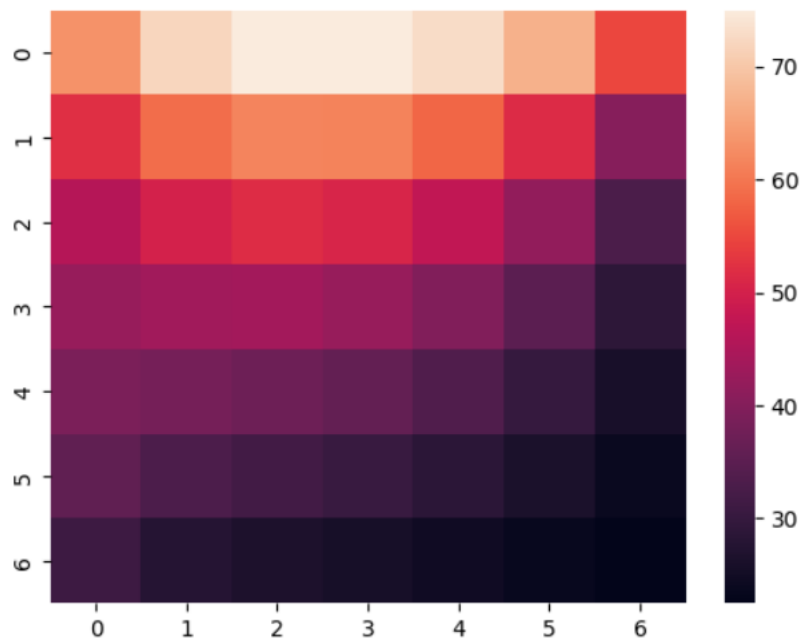
### 3.4 Parte II

Se procede a realizar una situación donde las temperaturas se corresponden a  $T_N = 40$ ,  $T_S = 20$ ,  $T_W = 20$ ,  $T_E = 90$ . La se asignó a  $R = 0.001$  y  $w = 1.8$ . El método de SOR necesitó de 17 iteraciones para la realización del problema, devolviendo un vector de temperaturas de 49 dimensiones.

La *motherboard* queda distribuida en una matriz  $A^{7 \times 7}$  donde la temperatura media corresponde al valor  $a_{44}$ , en esta posición la temperatura registrada fue de  $42.28 \pm 0.01$ .

#### Figura 7

*Distribución de temperatura de la motherboard para  $N = 8$*



*Nota.* Se aprecia un rango cualitativo de colores vs temperatura, donde el blanco es la temperatura mayor y el negro la menor.



#### 4. Conclusiones

Se demuestra que algunos problemas ingeniería pueden conllevar un análisis exhaustivo para una persona, sea de ejemplo el caso de separar la grilla para  $N = 32$ , problemas que pueden ser resueltos de una manera práctica mediante procesos iterativos como el método de SOR.

Se resalta que las estimaciones experimentales del  $w_{optimo}$  se acercan mucho a la idealidad presentada. Es de suma importancia encontrar el valor de  $w_{optimo}$ , pues nos ayuda a converger más rápidamente, siendo necesarias un menor número de iteraciones, por ende, un menor costo computacional y menor tiempo para la resolución efectiva del problema.

El tiempo necesario para encontrar el  $w_{optimo}$  puede ser tardado, pero es muy útil pues se acerca a aproximación ideal.

La optimización del código es también necesario, pues gracias a bibliotecas como *numpy* las matrices se construyen de una manera fácil, dado las condiciones del problema también es necesario reducir la matriz rala con algún método como *sparse row*.

Concluimos de esta forma que la utilización del método de SOR sobresale sobre otros métodos como Gauss-Seidel ( $w = 1$ ), siendo útil en distintos problemas de ingeniería, donde es necesaria la tolerancia ( $R$ ) como criterio de corte, produciendo errores de truncamiento en los resultados que se obtienen al aplicar el código.

## **Anexo 1: Código**