

Trabajo Práctico N°1  
Resolución de Sistemas de Ecuaciones Lineales  
Departamento de Computación - FIUBA  
Análisis Numérico (75.12/95.04)

Grupo 1

Camila Bojman - 101055 - camiboj@gmail.com  
Cecilia Hortas - 100687 - ceci.hortas@gmail.com  
Milena Marchese - 100962 - mile.marchese@gmail.com  
Rosario Szuplat - 100798 - szuplat.rosario@gmail.com

2do Cuatrimestre de 2018

Temática a tratar:

En el presente informe se resolverá la ecuación de Laplace en un recinto de dos dimensiones con determinadas condiciones de borde mediante tres métodos numéricos utilizados para resolver sistemas de ecuaciones lineales: Gauss, Jacobi y Gauss-Seidel. También, se hará una comparación de dichos métodos y se analizarán los errores producidos al hallar las soluciones por los métodos iterativos.

Las palabras claves a tener en cuenta durante todo el desarrollo del Trabajo Práctico son:

- Malla
- Error
- Tolerancia
- Sistema

## 1. Introducción

En primer lugar, se describe el problema matemático planteado en este Trabajo Práctico. Se pretende resolver numéricamente la ecuación de Laplace:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (1)$$

donde  $u(x,y)$  es la función potencial en una placa coincidente con el plano  $xy$ .

Se determinará una malla de discretización a fin de calcular el valor del potencial en una cantidad finita de puntos; para hallar el valor en todos los puntos del plano es necesario implementar una resolución analítica. Es decir, será únicamente en los nodos determinados por dicha malla donde se conocerá el valor de la función  $u$ . Luego, se impondrán las condiciones de borde pertinentes, las cuales permitirán obtener una solución única al problema.

Se propone la siguiente aproximación del valor de las derivadas parciales segundas utilizando los valores discretos determinados por la malla mencionada anteriormente:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{(i+1,j)} - 2u_{(i,j)} + u_{(i-1,j)}}{h^2} \quad (2)$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{u_{(i,j+1)} - 2u_{(i,j)} + u_{(i,j-1)}}{h^2} \quad (3)$$

donde  $u_{(i,j)}$  es el valor del potencial en el punto  $(x_i, x_j)$ , el cual se corresponde con un nodo de la malla de de discretización y  $h$  es la distancia entre dos nodos consecutivos. Se consideró que esta distancia es la misma entre un nodo y el siguiente tanto en la dirección  $x$  como en la  $y$ . Por lo tanto, reemplazando con las *Ecuaciones 2 y 3* en la *1* se obtiene:

$$u_{(i+1,j)} + u_{(i-1,j)} + u_{(i,j+1)} + u_{(i,j-1)} - 4u_{(i,j)} = 0 \quad (4)$$

Los valores máximos que pueden tomar  $j$  e  $i$  dependen de la cantidad de nodos que presenta la malla de discretización en la dirección  $x$  e  $y$  respectivamente. Se considera que el valor mínimo es 1. Los puntos  $(x_i, x_j)$  pertenecientes al borde son los que presentan un

índice (el  $i$  y/o el  $j$ ) fuera del rango determinado. Por ejemplo, el potencial en el punto (1,1) se expresa, utilizando la *Ecuación 4*, como:

$$u_{(2,1)} + u_{(0,1)} + u_{(1,2)} + u_{(1,0)} - 4u_{(1,1)} = 0$$

En este caso, los términos  $u_{(0,1)}$  y  $u_{(1,0)}$  corresponden al borde del recinto en el cual se está calculando el potencial. Por lo tanto, se conoce su valor y se lo reemplaza en la expresión, obteniéndose un término independiente no nulo, el cual es indispensable para obtener una solución única.

Se procede con la obtención de un sistema de ecuaciones lineales a partir de la *Ecuación 4* en el cual las incógnitas son el valor del potencial en cada nodo y los términos independientes son los determinados por las condiciones de borde, tal como se mostró en el ejemplo. Cada nodo conlleva una ecuación. Los coeficientes que multiplican a las incógnitas son: -4 para la incógnita correspondiente al valor del potencial en el nodo para el cual se está escribiendo la ecuación; 1 para el resto de las incógnitas que la *Ecuación 4* indica que deben tenerse en cuenta y que no presentan ningún índice fuera de rango; 0 para las incógnitas que no deben tenerse en cuenta y para las que presentan algún índice fuera de rango. Luego, estos coeficientes son los que forman la matriz del sistema de ecuaciones.

Una vez determinado el sistema de ecuaciones lineales tal como se explicó en el párrafo anterior, es posible resolver el problema utilizando diferentes métodos de resolución de sistemas de ecuaciones lineales. Los mismos se introducen teóricamente en el siguiente inciso y se implementan con *Octave*.

En segundo lugar, se pretende extrapolar el razonamiento indicado para la resolución de la ecuación de Laplace en dos dimensiones a tres. Para esto, se debe modificar las *Ecuaciones 2 y 3*, que aproximan discretamente las derivadas parciales segundas en  $x$  e  $y$ , para que incluyan puntos con una dimensión más, es decir,  $(x_i, x_j, x_z)$ :

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{(i+1,j,k)} - 2u_{(i,j,k)} + u_{(i-1,j,k)}}{h^2} \quad (5)$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{u_{(i,j+1,k)} - 2u_{(i,j,k)} + u_{(i,j-1,k)}}{h^2} \quad (6)$$

También, se indica cómo se realizan las derivadas parciales segundas con respecto a  $z$ , suponiendo análogamente a como se lo hizo para dos dimensiones, que la distancia entre dos nodos consecutivos de la malla de discretización en la dirección  $z$  también es  $h$ .

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{(i,j,k+1)} - 2u_{(i,j,k)} + u_{(i,j,k-1)}}{h^2} \quad (7)$$

Luego, sabiendo que la suma de las *Ecuaciones 5, 6 y 7* debe ser nula, se obtiene:

$$u_{(i+1,j,k)} + u_{(i-1,j,k)} + u_{(i,j+1,k)} + u_{(i,j-1,k)} + u_{(i,j,k+1)} + u_{(i,j,k-1)} - 6u_{(i,j,k)} = 0 \quad (8)$$

El resto de los pasos a seguir para la resolución del problema son los mismos que los explicados para el caso de dos dimensiones. El rango de  $k$ , al igual que los de  $i$  y  $j$ , es de 1 al número máximo de nodos que tiene la malla de discretización en la dirección  $z$  antes de llegar al borde del recinto.

## 2. Conceptos teóricos

Uno de los métodos de resolución de sistemas de ecuaciones lineales es el de Gauss, el cual es un método directo y se lo considerará exacto. Este coincide inicialmente en triangular la matriz del sistema de ecuaciones con pivoteo y luego realizar el procedimiento de sustitución hacia atrás. El mismo se llama directo porque se obtiene una única solución durante todo el procedimiento. Los posibles errores que pudiera conllevar este método provienen de las aproximaciones que realiza el software cuando un número no puede expresarse en la grilla que utiliza. Para poder hallar una solución única es necesario que la matriz no sea singular.

A diferencia de Gauss, los métodos de Jacobi y Gauss-Seidel son iterativos, por lo tanto, consisten en ir mejorando una solución inicial propuesta (semilla) hasta que, a pesar de que se sigan haciendo iteraciones, la solución deje de mejorar apreciablemente. Es decir, que la diferencia entre dos soluciones consecutivas sea menor a un parámetro determinado. Este se elegirá durante el desarrollo del programa que implementará cada método. Los mismos pueden no converger, es decir, que, a pesar de realizar infinitas iteraciones, no se llega nunca a la condición de corte explicada. Es por esto, que al implementar estos métodos es necesario agregar una segunda condición de corte: un número máximo de iteraciones a realizar. De esta manera, conociendo la razón por la cual finalizaron las iteraciones, se indica si el resultado hallado es correcto dentro de los parámetros de error propuestos o no.

Dado un sistema de ecuaciones expresado matricialmente como  $A \cdot x = b$ , para utilizar el método de Jacobi se deben expresar las ecuaciones lineales, tomando como ejemplo un sistema de 3x3, como:

$$x_1^{(k)} = [b_1 - (a_{12} \cdot x_2^{(k-1)} + a_{13} \cdot x_3^{(k-1)})] / a_{11}$$

$$x_2^{(k)} = [b_2 - (a_{21}x_1^{(k-1)} + a_{23}x_3^{(k-1)})]/a_{22}$$

$$x_3^{(k)} = [b_3 - (a_{31}x_1^{(k-1)} + a_{32}x_2^{(k-1)})]/a_{33}$$

donde  $k$  indica el número de iteración. Por lo tanto, en el comienzo del procedimiento, es decir, con  $k = 1$ , los  $x_i^{(k-1)}$  son las componentes del vector semilla propuesto y con cada iteración se van obteniendo  $x_i$  más precisos. Es decir, una vez que se hallaron los  $x_i^{(k)}$  para todas las incógnitas, en el ejemplo para  $i = 1$ ,  $i = 2$  e  $i = 3$ , se aumenta  $k$  en una unidad y se hallan nuevos  $x_i^{(k)}$ . La condición de corte explicada se puede expresar con esa notación como:

$$\|x^{(k)} - x^{(k-1)}\| \leq \epsilon \quad (9)$$

donde  $\epsilon$  es el parámetro que indica la máxima diferencia tolerada entre dos soluciones consecutivas.

En cambio, para el método de Gauss-Seidel se utiliza:

$$x_1^{(k)} = [b_1 - (a_{12}x_2^{(k-1)} + a_{13}x_3^{(k-1)})]/a_{11}$$

$$x_2^{(k)} = [b_2 - (a_{21}x_1^{(k)} + a_{23}x_3^{(k-1)})]/a_{22}$$

$$x_3^{(k)} = [b_3 - (a_{31}x_1^{(k)} + a_{32}x_2^{(k)})]/a_{33}$$

Por lo tanto, la diferencia con el método de Jacobi consiste en que, durante una misma iteración, es decir, sin incrementar  $k$ , se utilizan valores de  $x_i^{(k)}$  que fueron calculados en pasos anteriores. Por lo tanto, se espera que Gauss-Seidel converja más rápido, es decir, con menor número de iteraciones. En lo demás, ambos métodos son análogos.

### 3. Desarrollo

En primer lugar, se determina la malla de discretización en el cuadrado  $[0,1] \times [0,1]$  en el plano  $xy$  con una distancia entre nodos ( $h$ ) de 0,1 en las dos direcciones. Las condiciones de borde son las siguientes:

$$\begin{cases} u(x, 0) = u(x, 1) = 0 \\ u(0, y) = u(1, y) = y(1 - y) \end{cases}$$

Se realiza el esquema de la *Figura 1* a fin de observar gráficamente la situación física.

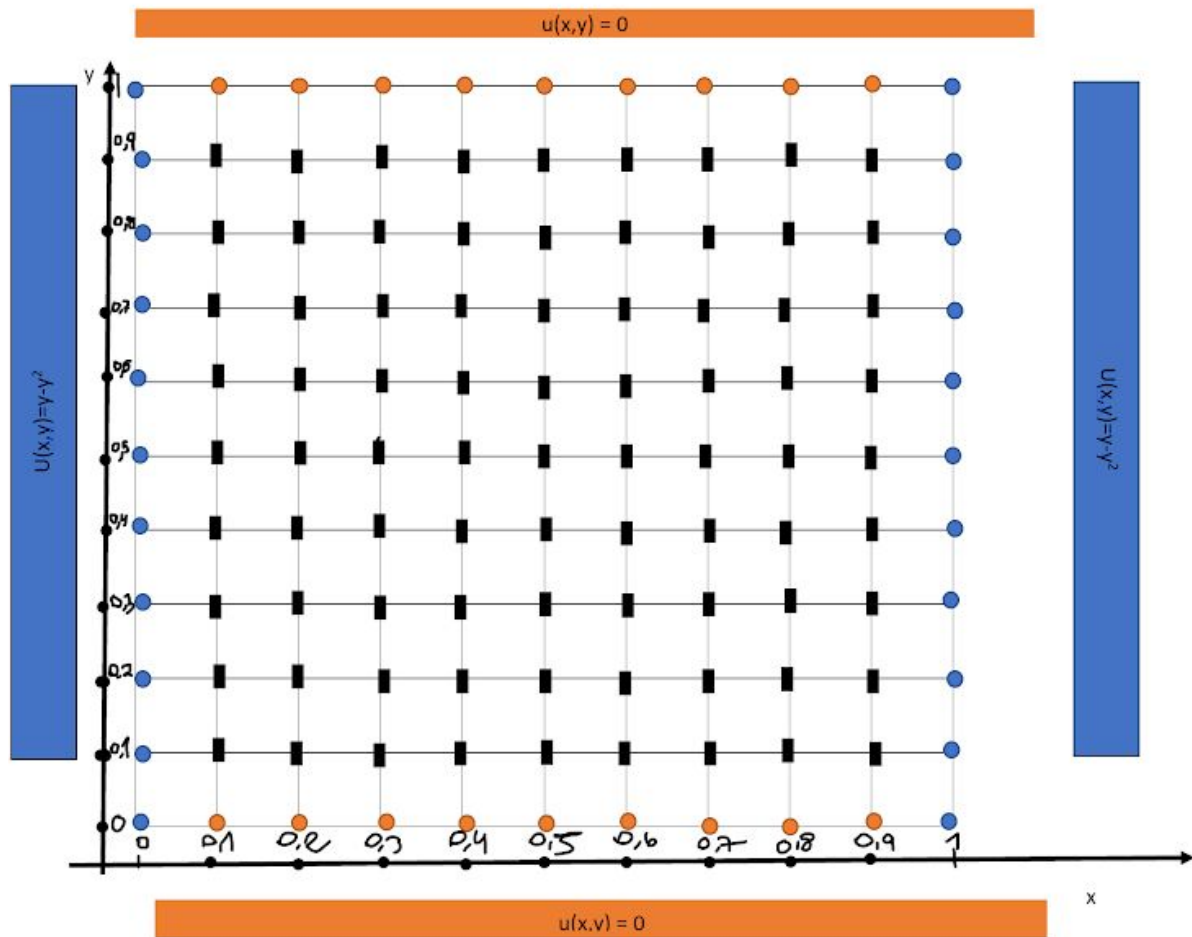


Figura 1: malla de discretización.

Cada cuadrado representa una distancia de  $h = 0,1$ . Los nodos de colores son los pertenecientes al borde, en decir, en los cuales se conocen los potenciales. En el mismo color se indica el valor de cada uno. Los negros son los 81 nodos en los cuales se hallara el potencial. También se grafican los sentidos de referencia  $x$  e  $y$ . Los coeficientes del nodo en la posición  $(0,1;0,9)$  son  $i=1, j=1$ . Al aumentar  $x$  se incrementa  $i$  mientras que al disminuir  $y$  se incrementa  $j$ .

Se plantea una ecuación para cada nodo a partir de la *Ecuación 4* a fin de obtener el sistema de ecuaciones lineales, en el cual los coeficientes de las incógnitas forman la matriz  $A$  y los términos independientes el vector  $b$ .

Luego se procede a trasladar el problema planteado al programa *Octave* para emplear los distintos métodos y realizar el correspondiente análisis. En un primer lugar surge la dificultad de cargar la correspondiente matriz al programa, lo cual se soluciona implementando dos funciones *generador\_vector()* y *generador\_matriz()* que crean respectivamente la matriz y el vector del sistema y los devuelven.

A la hora de implementar los diferentes métodos en *Octave* para obtener las soluciones aproximadas del sistema la primera problemática que surge es la de determinar la cota del error permitido y la cantidad de iteraciones máxima. Se determina que ambos se escogerán en base a probar diferentes opciones y elegir la que resulte más apropiada en base a un criterio adoptado. Se espera que la solución aproximada sea muy próxima a la solución exacta obtenida por el método directo y que la cantidad de iteraciones realizada no sea muy alta. Esto es porque si bien se podría utilizar un error muy chico a fin de obtener la solución más próxima posible, un número muy grande de iteraciones sería una desventaja. Además, se considera que el objetivo de Trabajo Práctico es comparar las distintas soluciones obtenidas por los métodos de Jacobi y el de Gauss-Seidel y eso puede realizarse con el criterio adoptado. Así se determina que el error máximo es de  $10^{-5}$  y la cantidad de iteraciones máxima de 150.

La implementación de los métodos de Jacobi y Gauss-Seidel no presenta ninguna complicación destacable. Se decide realizar dos funciones, una llamada *jacobi()* y otra *gauss\_seidel()* que devuelven la solución aproximada y la cantidad de iteraciones realizadas para obtenerla. En caso de exceder la cantidad máxima de iteraciones y no obtener un resultado con un error menor al deseado, se imprime un mensaje por pantalla que lo indica, además de que sería fácilmente deducible por la cantidad de iteraciones devuelta por la función. Se implementa finalmente la función *tp1* que es la principal del programa y se obtienen los distintos resultados que se requieren para el análisis: la solución obtenida por el método directo, las dos soluciones aproximadas obtenidas por los métodos iterativos y las respectivas cantidades de iteraciones.

Adicionalmente, para el análisis se deciden realizar dos funciones para calcular distintos errores relativos: *error\_relativo\_jacobi()* y *error\_relativo\_gauss\_seidel()*. Ambas siguen el mismo procedimiento con la diferencia que una utiliza el método de Jacobi y la otra el método de Gauss Seidel para hallar la solución aproximada. Se considera que es fructífero para el análisis el cálculo del error relativo de cada solución aproximada con el uso de la siguiente ecuación:

$$\mathcal{E}_{relativo} = \|x_{exacto} - x_{aproximado}\|_{\infty} / \|x_{exacto}\| \quad (10)$$

Además se calcula el error relativo para cada componente de la solución aproximada y se obtienen el máximo y el mínimo de ellos.

$$\mathcal{E}_{relativo\ componente} = (x_{i\ exacto} - x_{i\ aproximado}) / x_{i\ exacto} \quad (11)$$

#### 4. Resultados

Se exponen en este apartado los distintos resultados obtenidos. En primer lugar se deduce que la matriz del sistema (A) y el término independiente (b) son los que siguen:

$$\mathbf{b} = [-0.09, 0, 0, 0, 0, 0, 0, 0, 0, -0.09, 0.16, 0, 0, 0, 0, 0, 0, 0, -0.16, -0.21, 0, 0, 0, 0, 0, 0, 0, -0.21, -0.24, 0, 0, 0, 0, 0, 0, 0, 0, -0.24, -0.25, 0, 0, 0, 0, 0, 0, 0, -0.25, -0.24, 0, 0, 0, 0, 0, 0, 0, -0.24, -0.21, 0, 0, 0, 0, 0, 0, 0, 0, -0.21, -0.16, 0, 0, 0, 0, 0, 0, 0, -0.16, -0.09, 0, 0, 0, 0, 0, 0, 0, -0.09]^T$$

La matriz A es de 81x81 y para su representación se la divide en columnas:

Columna 1 a 19

Columna 20 a 38

[illegible]





9

Columna 77 a 81:

10

[illegible]

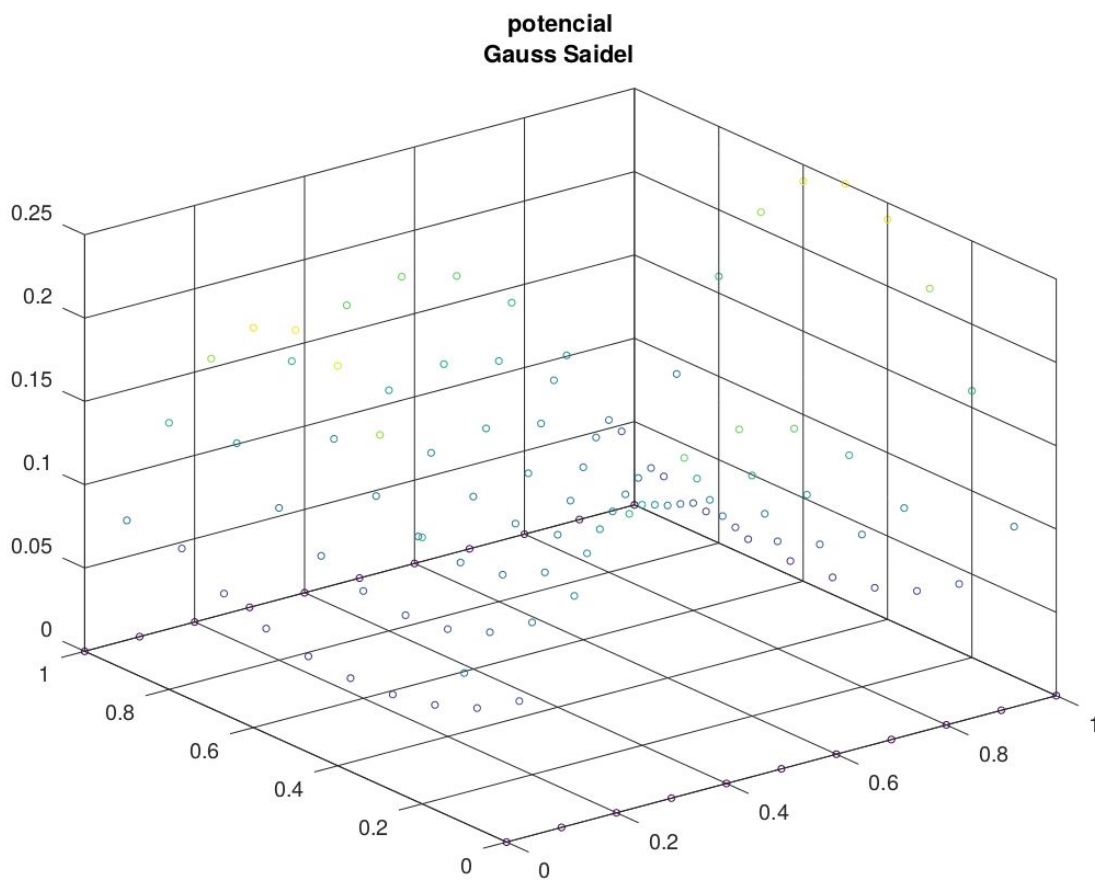
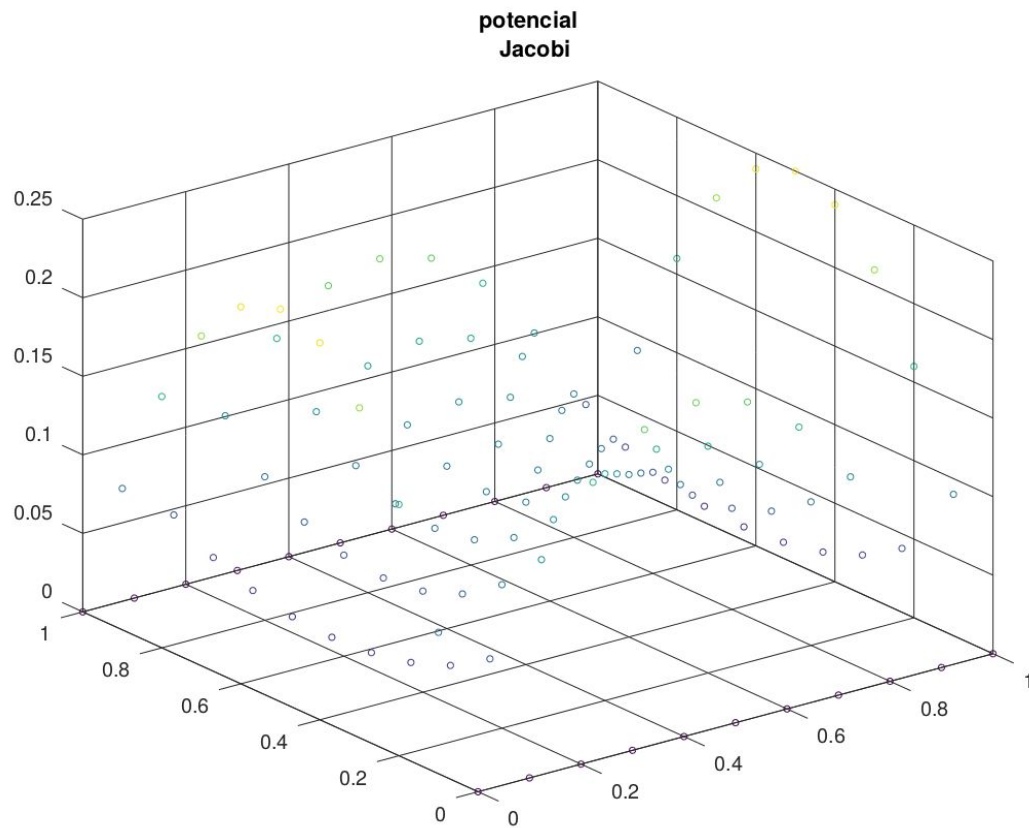
Luego se exponen las soluciones obtenidas por los métodos exactos e iterativos en la *Tabla 1* a fin de que pueda observarse de mejor manera la diferencia obtenida componente a componente.

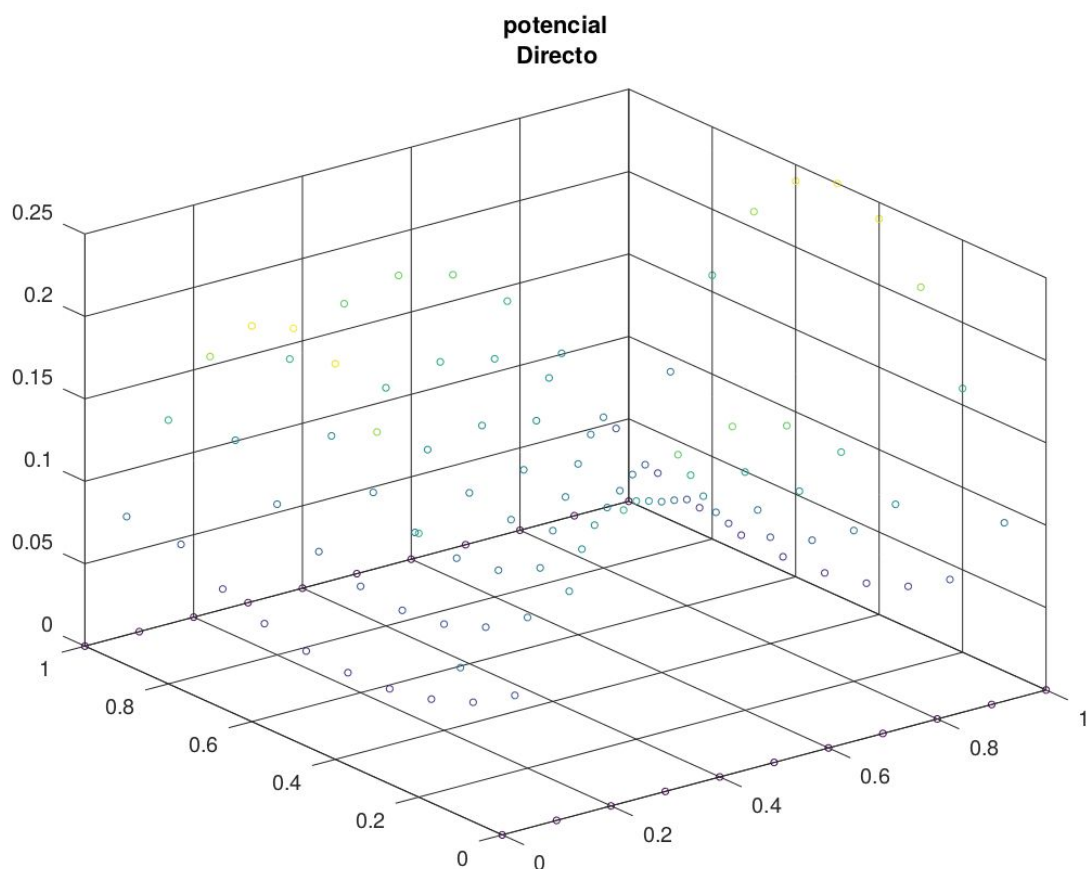
x exacto	x obtenido por Jacobi	x obtenido por Gauss-Seidel
0.064374	0.064356	0.064361
0.048748	0.048714	0.048724
0.039206	0.039159	0.039175
0.034001	0.033946	0.033966
0.032343	0.032286	0.032309
0.034001	0.033946	0.033970
0.039206	0.039159	0.039180
0.048748	0.048714	0.048730
0.064374	0.064356	0.064365
0.118748	0.118714	0.118724
0.091411	0.091347	0.091368
0.074074	0.073986	0.074018
0.064455	0.064351	0.064393
0.061371	0.061262	0.061308
0.064455	0.064351	0.064399
0.074074	0.073986	0.074028
0.091411	0.091347	0.091379
0.118748	0.118714	0.118732
0.159206	0.159159	0.159175
0.124074	0.123986	0.124018
0.101225	0.101103	0.101151
0.088375	0.088232	0.088293
0.084231	0.084081	0.084149
0.088375	0.088232	0.088301
0.101225	0.101103	0.101164
0.124074	0.123986	0.124033
0.159206	0.159159	0.159185
0.184001	0.183946	0.183966
0.144455	0.144351	0.144393
0.118375	0.118232	0.118293
0.103589	0.103421	0.103497
0.098803	0.098626	0.098711
0.103589	0.103421	0.103506
0.118375	0.118232	0.118308
0.144455	0.144351	0.144409
0.184001	0.183946	0.183978
0.192343	0.192286	0.192309
0.151371	0.151262	0.151308
0.124231	0.124081	0.124149
0.108803	0.108626	0.108711

0.103803	0.103617	0.103711
0.108803	0.108626	0.108720
0.124231	0.124081	0.124164
0.151371	0.151262	0.151325
0.192343	0.192286	0.192320
0.184001	0.183946	0.183970
0.144455	0.144351	0.144399
0.118375	0.118232	0.118301
0.103589	0.103421	0.103506
0.098803	0.098626	0.098720
0.103589	0.103421	0.103514
0.118375	0.118232	0.118314
0.144455	0.144351	0.144413
0.184001	0.183946	0.183980
0.159206	0.159159	0.159180
0.124074	0.123986	0.124028
0.101225	0.101103	0.101164
0.088375	0.088232	0.088308
0.084231	0.084081	0.084164
0.088375	0.088232	0.088314
0.101225	0.101103	0.101175
0.124074	0.123986	0.124040
0.159206	0.159159	0.159189
0.118748	0.118714	0.118730
0.091411	0.091347	0.091379
0.074074	0.073986	0.074033
0.064455	0.064351	0.064409
0.061371	0.061262	0.061325
0.064455	0.064351	0.064413
0.074074	0.073986	0.074040
0.091411	0.091347	0.091387
0.118748	0.118714	0.118736
0.064374	0.064356	0.064365
0.048748	0.048714	0.048732
0.039206	0.039159	0.039185
0.034001	0.033946	0.033978
0.032343	0.032286	0.032320
0.034001	0.033946	0.033980
0.039206	0.039159	0.039189
0.048748	0.048714	0.048736
0.064374	0.064356	0.064368

*Tabla 1: comparación de los resultados obtenidos por 3 métodos diferentes.*

Por el mismo motivo se generan los gráficos para cada solución.





Además, para poder comparar los métodos iterativos utilizados se expresan en la *Tabla 2* los distintos errores relativos hallados a partir de lo presentado en el *Desarrollo*:

Método de Jacobi			Método de Gauss-Seidel		
$\epsilon$	$\epsilon_{\text{componente max}}$	$\epsilon_{\text{componente min}}$	$\epsilon$	$\epsilon_{\text{componente max}}$	$\epsilon_{\text{componente min}}$
$1.9126 \times 10^{-4}$	$6.9346 \times 10^{-4}$	$2.7607 \times 10^{-4}$	$9.4672 \times 10^{-5}$	$2.4147 \times 10^{-4}$	$9.1429 \times 10^{-5}$

*Tabla 2: errores porcentuales.*

Finalmente lo último a exponer, en la *Tabla 3*, es la cantidad de iteraciones realizada por cada algoritmo en esta situación particular:

	Método de Jacobi	Método de Gauss-Seidel
Cantidad de iteraciones (k)	135	75

*Tabla 3: cantidad de operaciones para cada método.*



## 5. Conclusiones

A modo de conclusión se afirma que se cumplió el objetivo del Trabajo Práctico ya que se logró hallar soluciones exactas y aproximadas para el sistema planteado. Se halló la matriz del sistema  $A$  y el vector independiente  $b$  a partir de las condiciones de borde planteadas en el enunciado. Así mismo, se implementaron las distintas funciones de los métodos iterativos en el programa *Octave*, pudiendo plasmar los conceptos teóricos en la práctica.

Por otro lado, las soluciones aproximadas obtenidas se acercan de manera significativa a la solución exacta por lo que se considera que los métodos deberían estar planteados e implementados de manera correcta. Es pertinente aclarar que el error máximo tolerado y la cantidad de iteraciones máxima fueron fijados por un criterio personal del grupo que podría ser cambiado para otro tipo de problemática. Aquí se eligió tener un error lo suficientemente pequeño como para que la solución aproximada se acerque a la real pero no muy pequeño para no obtener una gran cantidad de iteraciones para hallar la solución.

Se puede observar que para este tipo de problemática, con las condiciones de borde planteadas y los errores y cantidad de máxima de iteraciones permitidos, el método de Gauss-Seidel utiliza una menor cantidad de iteraciones y presenta un error más pequeño. Esto se condice con la idea de que el método de Jacobi precede al de Gauss-Seidel por lo que en la mayor parte de los casos acelera su convergencia (Gauss-Seidel converge en 75 iteraciones cuando Jacobi lo hace en 135).

Sin embargo, se considera que el mejor método a utilizar es el directo ya que la matriz está compuesta principalmente por ceros, por lo que el mismo es el más eficiente. No es necesario ocupar memoria y tiempo en la implementación de métodos iterativos, en los cuales además se produce mayor error.

## 6. Referencias

Burden, Richard L. y Faires, J. Douglas; “*Análisis Numérico*”; Séptima edición; Thomson Learning.