



UNIVERSIDAD DE
GUANAJUATO

Universidad de Guanajuato

División de Ciencias e Ingenierías

Ingeniería Química Sustentable

“Métodos Numéricos”

**Proyecto Sistemas Lineales – Determinación de
temperaturas en sistema de calentadores en serie**

José Emilio Sierra Magaña - 827118

Docente: Dra. Alma Xóchitl González Morales

Fecha de entrega: 16/11/2025

OBJETIVO

Solucionar un problema relacionado a la ingeniería química mediante métodos de solución de sistemas lineales de 4 dimensiones.

INTRODUCCIÓN

En la industria, los sistemas de calentamiento son esenciales para llevar a cabo transformaciones físicas y químicas de forma controlada.

Para el calentamiento de diversas sustancias se tiene una amplia variedad de equipos y métodos, que transfieren energía a la sustancia objetivo haciendo uso de conceptos como la diferencia de calor, el efecto Joule, entre otros.

En este proyecto, se plantea un sistema de 4 calentadores conectados en serie, los cuales calientan un fluido mediante resistencias sumergidas. Este calentamiento está sujeto al calor aportado por cada calentador, por los calentadores vecinos y por la pérdida de calor al ambiente.

PLANTEAMIENTO DEL SISTEMA

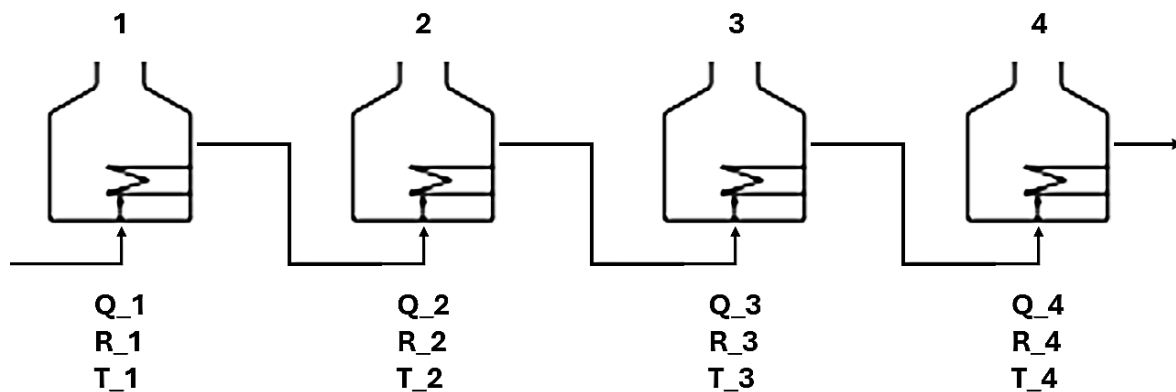


Figura 1. Diagrama ilustrativo de 4 calentadores dispuestos en serie.

El sistema en cuestión consiste en una serie de 4 calentadores, por los cuales fluye un fluido el cual recibe calor en diferente magnitud en cada nodo. La temperatura en cada nodo depende del calor que cada resistencia ejerce, el calor

que gana o pierde en relación con los nodos vecinos y el calor que pierde por la temperatura del medio.

La ecuación que rige el calor en cada nodo es la siguiente. [1]

$$Q_i = hA(T_i - T_\infty) + \frac{T_i - T_{i-1}}{R_i} + \frac{T_i - T_{i+1}}{R_i} \quad (1)$$

Donde el primer término representa la pérdida de calor al ambiente, y el segundo y tercero representa la aportación de los nodos vecinos. En el caso de que solo tenga un nodo vecino (para el primero y el último), uno de los términos de nodo vecino se elimina.

Por comodidad, definimos al producto del coeficiente de convección por el área superficial como K.

$$K = hA \quad (2)$$

Sustituyendo en la ecuación 1.

$$Q_i = K(T_i - T_\infty) + \frac{T_i - T_{i-1}}{R_i} + \frac{T_i - T_{i+1}}{R_i} \quad (3)$$

Partiendo de esta ecuación para cada nodo, se establecen valores arbitrarios de K, Q_i y R_i para cada nodo, por lo que las variables a determinar son las temperaturas dentro de cada nodo.

CÁLCULOS E IMPLEMENTACIÓN DE CÓDIGO

Se definen como valores arbitrarios:

$$R_1 = 4 \frac{^\circ\text{C}}{\text{W}}$$

$$R_4 = 1 \frac{^\circ\text{C}}{\text{W}}$$

$$Q_4 = 30 \text{ W}$$

$$R_2 = 2 \frac{^\circ\text{C}}{\text{W}}$$

$$Q_1 = 90 \text{ W}$$

$$K = 0.5 \frac{\text{W}}{^\circ\text{C}}$$

$$R_3 = 3 \frac{^\circ\text{C}}{\text{W}}$$

$$Q_2 = 50 \text{ W}$$

$$T_\infty = 23^\circ\text{C}$$

$$Q_3 = 10 \text{ W}$$

Antes de asignar los valores a nuestras expresiones, es conveniente definir las ecuaciones de modo que se despejen los coeficientes que se agregarán a la matriz final.

Partiendo de la ecuación (3), despejamos todos los términos de temperatura.

$$(K + R_1^{-1})T_1 - R_1^{-1}T_2 = Q_1 + KT_\infty \quad (4)$$

$$-R_2^{-1}T_1 + (K + R_2^{-1})T_2 - R_2^{-1}T_3 = Q_2 + KT_\infty \quad (5)$$

$$-R_3^{-1}T_2 + (K + R_3^{-1})T_3 - R_3^{-1}T_4 = Q_3 + KT_\infty \quad (6)$$

$$-R_4^{-1}T_3 + (K + R_4^{-1})T_4 = Q_4 + KT_\infty \quad (7)$$

A partir de este sistema de ecuaciones, se pueden escribir los vectores $Ax=b$.

$$\begin{pmatrix} K + R_1^{-1} & -R_1^{-1} & 0 & 0 \\ -R_2^{-1} & K + R_2^{-1} & -R_2^{-1} & 0 \\ 0 & -R_3^{-1} & K + R_3^{-1} & -R_3^{-1} \\ 0 & 0 & -R_4^{-1} & K + R_4^{-1} \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{pmatrix} = \begin{pmatrix} Q_1 + KT_\infty \\ Q_2 + KT_\infty \\ Q_3 + KT_\infty \\ Q_4 + KT_\infty \end{pmatrix} \quad (8)$$

Así, sustituyendo los valores definidos al principio en la ecuación (8).

$$\begin{pmatrix} 0.75 & -0.25 & 0 & 0 \\ -0.5 & 1 & -0.5 & 0 \\ 0 & -0.33 & 0.83 & -0.33 \\ 0 & 0 & -1 & 1.5 \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{pmatrix} = \begin{pmatrix} 101.5 \\ 61.5 \\ 21.5 \\ 41.5 \end{pmatrix} \quad (9)$$

Este sistema es el que se escribe en un archivo de texto, el cual será leído por los códigos para el método de Gauss-Jordan, Gauss-Seidel y factorización LU.

Estos códigos se encuentran en el repositorio de GitHub.[2]

RESULTADOS Y DISCUSIÓN DE RESULTADOS

Gauss – Jordan

A continuación, se muestra el mensaje de la terminal luego de ejecutar el código con los datos de la ecuación (9).

Primero, el programa pide ingresar el número de filas y columnas, así el programa define límites para la lectura del archivo de texto.

Posteriormente despliega la matriz, y, de ser necesario, la modifica para evitar divisiones entre cero.

Se muestran los factores de multiplicación para la transformación con el método de Gauss simple, y luego el de Gauss – Jordan.

Ambas matrices se muestran para verificar que el método está avanzando de forma correcta.

Finalmente, cuando tenemos la matriz linealizada, se evalúa el determinante para así definir si el sistema está bien condicionado.

Finalmente, se muestran los resultados, en este caso las temperaturas dentro de cada calentador.

```
Ingresar numero de filas y columnas:
4
4
```

```
Matriz leída exitosamente desde: matriz.txt
```

```
Matriz inicial:
```

0.7500	-0.2500	0.0000	0.0000		101.5000
-0.5000	1.0000	-0.5000	0.0000		61.5000
0.0000	-0.3333	0.8333	-0.3333		21.5000
0.0000	0.0000	-1.0000	1.5000		41.5000

```
Matriz modificada:
```

0.7500	-0.2500	0.0000	0.0000		101.5000
-0.5000	1.0000	-0.5000	0.0000		61.5000
0.0000	-0.3333	0.8333	-0.3333		21.5000
0.0000	0.0000	-1.0000	1.5000		41.5000

```
Factores de multiplicación:
```

```
-0.666667
0.000000
0.000000
-0.399960
0.000000
-1.578981
```

```
Matriz diagonal resultante (Gauss):
```

0.7500	-0.2500	0.0000	0.0000		101.5000
0.0000	0.8333	-0.5000	0.0000		129.1667
0.0000	0.0000	0.6333	-0.3333		73.1615
0.0000	0.0000	0.0000	0.9737		157.0206

```
Factores de multiplicación:
```

```
-0.342294
0.000000
0.000000
-0.789490
0.000000
-0.300000
```

```
Matriz diagonal resultante (Gauss-Jordan):
```

0.7500	0.0000	0.0000	0.0000		170.3080
0.0000	0.8333	0.0000	0.0000		229.3598
0.0000	0.0000	0.6333	0.0000		126.9086
0.0000	0.0000	0.0000	0.9737		157.0206

```
El sistema está bien condicionado, det(A) = 0.385
```

```
Resultados:
```

```
x[1] = 227.0773
x[2] = 275.2318
x[3] = 200.3863
x[4] = 161.2575
```

Gauss – Seidel

El inicio del programa es similar al de Gauss – Jordan, se solicita el número de filas y columnas, y se muestra la matriz inicial y modificada en caso de que los primeros coeficientes sean cero. En este caso no es necesaria una modificación.

Posteriormente se definen las matrices N y P.

Finalmente, se llevan a cabo las iteraciones del vector x, las cuales convergen después de 23 iteraciones, cuando alcanza un error menor a 0.0001.

Ingresa número de filas y columnas:

4

4

Matriz leída exitosamente desde: matriz.txt

Matriz inicial:

0.7500	-0.2500	0.0000	0.0000		101.5000
-0.5000	1.0000	-0.5000	0.0000		61.5000
0.0000	-0.3333	0.8333	-0.3333		21.5000
0.0000	0.0000	-1.0000	1.5000		41.5000

Matriz modificada:

0.7500	-0.2500	0.0000	0.0000		101.5000
-0.5000	1.0000	-0.5000	0.0000		61.5000
0.0000	-0.3333	0.8333	-0.3333		21.5000
0.0000	0.0000	-1.0000	1.5000		41.5000

Matriz N:

0.7500	-0.2500	0.0000	0.0000
0.0000	1.0000	-0.5000	0.0000
0.0000	0.0000	0.8333	-0.3333
0.0000	0.0000	0.0000	1.5000

Matriz P:

0.0000	0.0000	0.0000	0.0000
0.5000	0.0000	0.0000	0.0000
-0.0000	0.3333	0.0000	0.0000
-0.0000	-0.0000	1.0000	0.0000

Iteraciones Gauss-Seidel

Iter	Error	x1	x2	x3	x4
0	79.309731	135.333328	129.166656	77.464592	79.309731
1	37.216370	178.388870	189.426727	133.289154	116.526100
2	20.044685	198.475586	227.382370	163.356171	136.570786
3	11.040436	211.127457	248.741821	179.916824	147.611221
4	6.101135	218.247269	260.582031	189.068542	153.712357
5	3.373230	222.194016	267.131287	194.128387	157.085587
6	1.865143	224.377090	270.752747	196.926086	158.950729
7	1.031281	225.584244	272.755157	198.473022	159.982010
8	0.570236	226.251709	273.862366	199.328369	160.552246
9	0.315308	226.620789	274.474579	199.801331	160.867554
10	0.174332	226.824875	274.813110	200.062836	161.041885
11	0.096405	226.937698	275.000275	200.207443	161.138290
12	0.053299	227.000076	275.103760	200.287384	161.191589
13	0.029465	227.034592	275.160980	200.331589	161.221054
14	0.016296	227.053665	275.192627	200.356033	161.237350
15	0.009018	227.064209	275.210114	200.369553	161.246368
16	0.004974	227.070023	275.219788	200.377014	161.251343
17	0.002747	227.073257	275.225128	200.381134	161.254089
18	0.001526	227.075027	275.228088	200.383423	161.255615
19	0.000839	227.076035	275.229736	200.384689	161.256454
20	0.000488	227.076584	275.230652	200.385406	161.256943
21	0.000244	227.076889	275.231140	200.385788	161.257187
22	0.000137	227.077042	275.231415	200.385986	161.257324
23	0.000092	227.077148	275.231567	200.386124	161.257416

Factorización LU

Este código inicia de forma similar a los anteriores hasta la normalización de la matriz. Es entonces cuando define y despliega los vectores L y U, los cuales son el resultado de la descomposición de la matriz original.

Debido a que se define que $Ux=y$, se despliega el vector y, con el cual se obtienen los valores del vector x.

```
Ingresa numero de filas y columnas:
4
4
Matriz leída exitosamente desde: matriz.txt

Matriz inicial:
  0.7500 -0.2500  0.0000  0.0000 | 101.5000
-0.5000  1.0000 -0.5000  0.0000 |  61.5000
  0.0000 -0.3333  0.8333 -0.3333 |  21.5000
  0.0000  0.0000 -1.0000  1.5000 |  41.5000

Matriz modificada:
  0.7500 -0.2500  0.0000  0.0000 | 101.5000
-0.5000  1.0000 -0.5000  0.0000 |  61.5000
  0.0000 -0.3333  0.8333 -0.3333 |  21.5000
  0.0000  0.0000 -1.0000  1.5000 |  41.5000

Matriz U resultante:
  0.7500 -0.2500  0.0000  0.0000
  0.0000  0.8333 -0.5000  0.0000
  0.0000  0.0000  0.6333 -0.3333
  0.0000  0.0000  0.0000  0.9737

Matriz L resultante:
  1.0000  0.0000  0.0000  0.0000
-0.6667  1.0000  0.0000  0.0000
  0.0000 -0.4000  1.0000  0.0000
  0.0000  0.0000 -1.5790  1.0000

Vector y:
y[1] = 101.5000
y[2] = 129.1667
y[3] = 73.1615
y[4] = 157.0206

Vector x:
x[1] = 227.0773
x[2] = 275.2318
x[3] = 200.3863
x[4] = 161.2575
```

Comparación de los resultados en cada método

Nodo	Gauss - Jordan	Gauss - Seidel	Factorización LU
1	227.0773	227.0771	227.0773
2	275.2318	275.2316	275.2318
3	200.3863	200.3861	200.3863
4	161.2575	161.2574	161.2527

Tabla 1. Comparación de los resultados para cada método.

Como se muestra en la tabla 1, los resultados apenas cambiaron entre métodos, lo que sugiere que el sistema es relativamente sencillo. Es por esto por lo que el determinante indica que el sistema está bien condicionado.

La variación entre resultados es de menos del 0.02%.

En cuanto a la interpretación, estos valores representan la temperatura en cada calentador expresada en grados centígrados, y son congruentes con el valor de calor suministrado (directamente proporcional) y resistencia térmica (inversamente proporcional), tanto del nodo en cuestión como con los nodos vecinos.

CONCLUSIÓN

El sistema lineal propuesto mostró ser numéricamente estable y bien condicionado, lo cual permitió que los tres métodos empleados —Gauss-Jordan, Gauss-Seidel y factorización LU— generaran soluciones prácticamente idénticas. Esto confirma que las características del modelo, como las diferencias suficientemente grandes entre magnitudes y la estructura casi diagonal dominante, favorecen la precisión y la robustez de los métodos.

Además, el comportamiento físico de la solución resultó completamente coherente con los principios de transferencia de calor aplicados: las temperaturas obtenidas siguen patrones esperados y reflejan adecuadamente la influencia del calor suministrado y de las resistencias térmicas del sistema.

BIBLIOGRAFIAS

[1] Çengel, Yunus. (2007) Transferencia de Calor y Masa, un enfoque práctico.

[2] <https://github.com/DCI-alxogm/me2025-clase-joseemiliosierra123/tree/main/Proyecto%20Sistemas%20Lineales>