INSTITUTO POLITÉCNICO NACIONAL

Unidad Profesional Interdisciplinaria de Biotecnología

**Unidad de Aprendizaje**: Métodos Numéricos

**Tarea No 4.**

*“ Métodos de Gauss-Jordan y Matriz inversa con pivoteo parcial”*

**Profesores:**

Marin Albino María del Carmen

Rosas Mendoza Jorge Luis

**Alumnos:**

Escalante Villalba Alexa

Minajas Carbajal Francisco Javier

Mireles Pérez María Caridad

Salmerón Ramírez Amanda

**Grupo:** 4FV3

**Fecha de entrega:** 3/10/2017

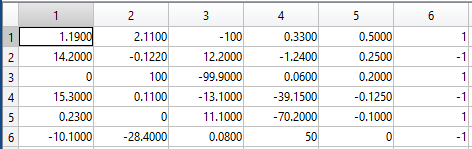
Equipo 9

**Ciclo escolar:** 2018/1

**Ejercicio 2**

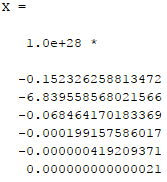
Utilizando el método de la matriz inversa evaluar:

Se introdujo la matriz como sigue en el programa:



Generando las variables A y B respectivamente conteniendo la matriz de coeficientes y la matriz de resultados.

Después de procesar la solución en el programa este dio lo siguiente:



**Código Utilizado**

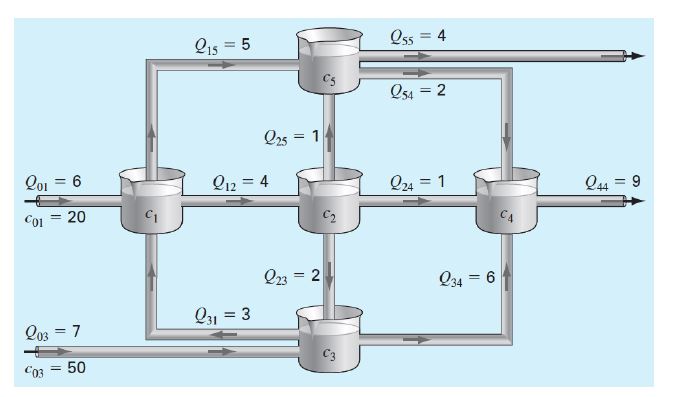
1. **%Matriz inversa**
2. **clear all;**
3. **clc;**
4. **A=[1.19000000000000,2.11000000000000,-100,0.330000000000000,0.500000000000000,1;14.2000000000000,-0.122000000000000,12.2000000000000,-1.24000000000000,0.250000000000000,-1;0,100,-99.9000000000000,0.0600000000000000,0.200000000000000,1;15.3000000000000,0.110000000000000,-13.1000000000000,-39.1500000000000,-0.125000000000000,-1;0.230000000000000,0,11.1000000000000,-70.2000000000000,-0.100000000000000,1;-10.1000000000000,-28.4000000000000,0.0800000000000000,50,0,-1];**
5. **b=[-1.12000000000000;-3.44000000000000;-2.15000000000000;-4.16000000000000;-8.75000000000000;9.01000000000000];**
6. **%matriz aumentada a=[A,b]**
7. **format long**
8. **%a=[A b]**
9. **a=[A eye(size(A))]**
10. **tam=size(a,1);**
11. **pos=1;**
13. **for ii=1:tam**
14. **pos=0;**
15. **aux=a(:,ii);**
16. **aux=transpose(aux);**
17. **pos=posicion(aux,ii);**
18. **aux2=a(pos,:);**
19. **a(pos,:)=a(ii,:);**
20. **a(ii,:)=aux2;**
21. **%pivoteamos**
22. **for iii=1:tam**
23. **if a(ii,ii)~=0**
24. **a(ii,iii)=a(ii,iii)/a(ii,ii);**
25. **else**
26. **a(ii,iii)=0;**
27. **end**
28. **end**
30. **for i=1:tam**
31. **if i~=ii**
32. **a(i,:)=a(i,:)-a(ii,:)\*a(i,ii);**
33. **end**
34. **end**
35. **%a**
37. **end**
38. **format long**
39. **A\_inv=a(:,tam+1:tam\*2);**
40. **X=A\_inv\*b**
41. **inv(A);**
43. **function y=posicion(a,b)**
44. **format rat**
45. **[m n]=size(a);**
46. **for c=1:b**
47. **if c<b**
48. **a(1,c)=0;**
49. **end**
50. **end**
51. **for c=1:n**
52. **a(1,c)=abs(a(1,c));**
53. **end**
55. **numeromayor=a(1,1);**
56. **iposicion=1;**
58. **for x=1:n**
59. **if a(1,x)>numeromayor**
60. **numeromayor=a(1,x);**
61. **iposicion=x;**
62. **end**
63. **end**
64. **y=iposicion;**
65. **end**

**Ejercicio 4**

En la figura se muestran cinco reactores unidos por tuberías. Como puede ver, la tasa de transferencia de químicos a través de cada conducto es igual a la tasa de flujo Q (con unidades de multiplicado por la concentración c del reactor desde el cual el flujo se origina (con unidades de . Si el sistema está en estado estacionario, el flujo de masa de entrada y de salida debe ser igual, por ejemplo, para el primer reactor el balance de masa es



Escriba las ecuaciones de equilibrio-masa para los reactores y resuelva el sistema para las concentraciones aplicando el Método de Matriz Inversa



Obteniendo las ecuaciones de parte de cada uno de los reactores tenemos que:

Para el reactor 1

Para el reactor 2

por lo tanto:

Para el reactor 3

por lo tanto :

Para el reactor 4

Y para el reactor 5

Por lo tanto:

Sustituyendo los valores:

ecuación (1)

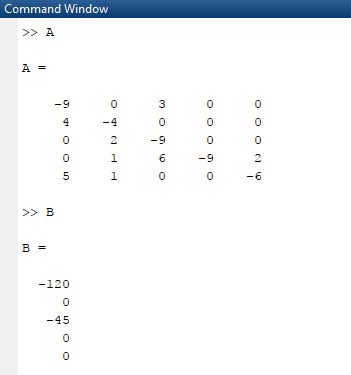
ecuación (2)

ecuación (3)

ecuación (4)

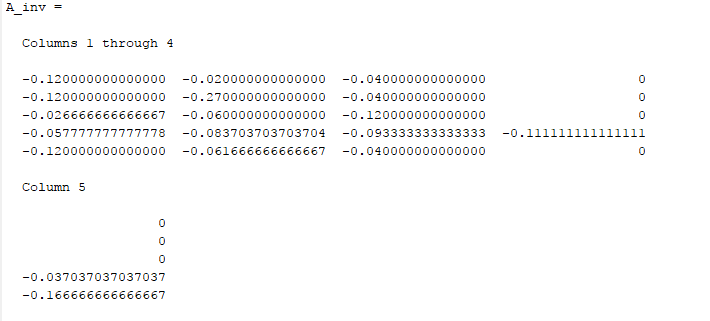
ecuación (5)

Introduciendo las matrices en la ventana de comando

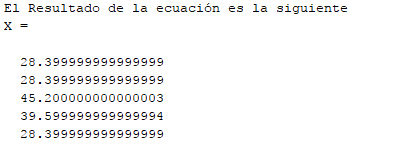


Para resolverlo por matriz inversa se necesita una matriz identidad para obtener una matriz expandida es decir

Se obtiene como resultado de la matriz inversa



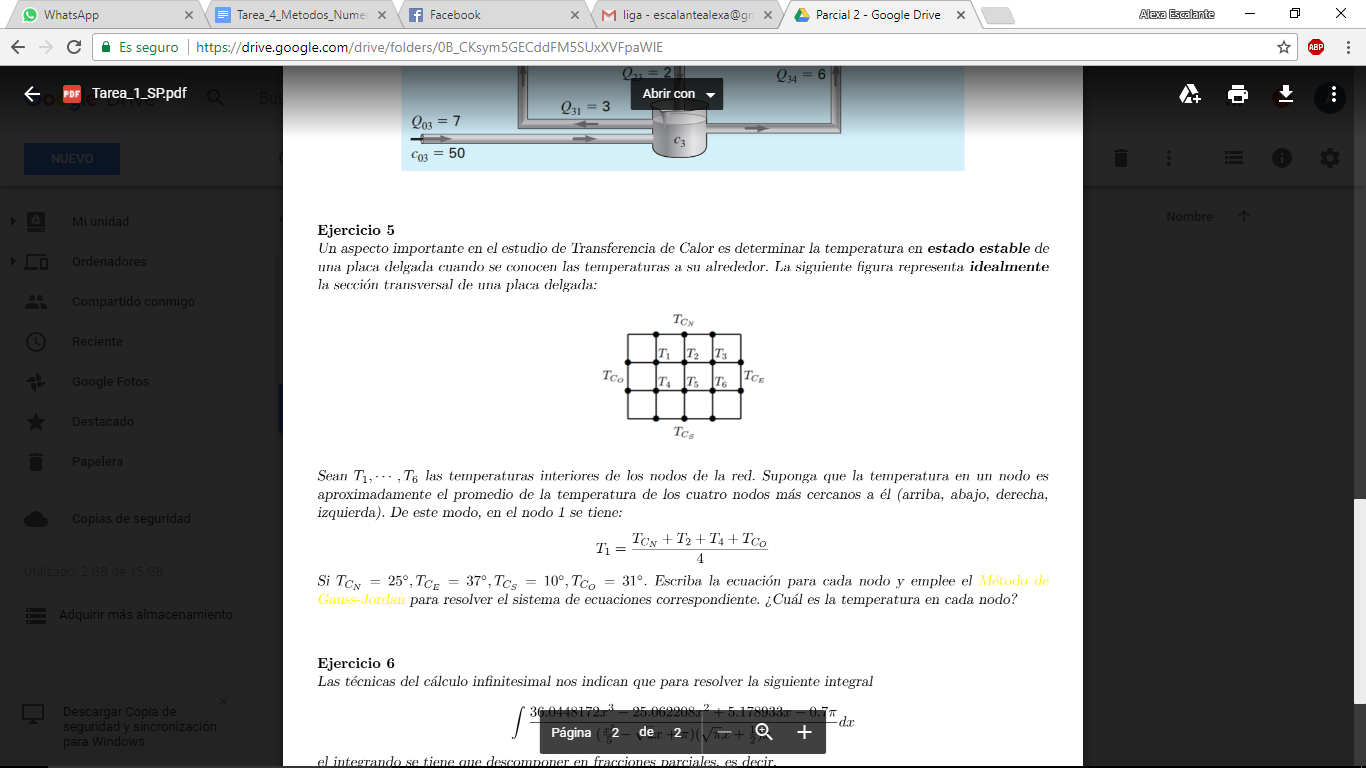
Y el resultado a la ecuación (X) es :



El código empleado para la realización del programa es:

1. A=[-9,0,3,0,0;4,-4,0,0,0;0,2,-9,0,0;0,1,6,-9,2;5,1,0,0,-6];
2. B=[-120;0;-350;0;0];
3. I=[1,0,0,0,0;0,1,0,0,0;0,0,1,0,0;0,0,0,1,0;0,0,0,0,1];
4. format long
5. a=[A I];
6. a=[A eye(size(a))];

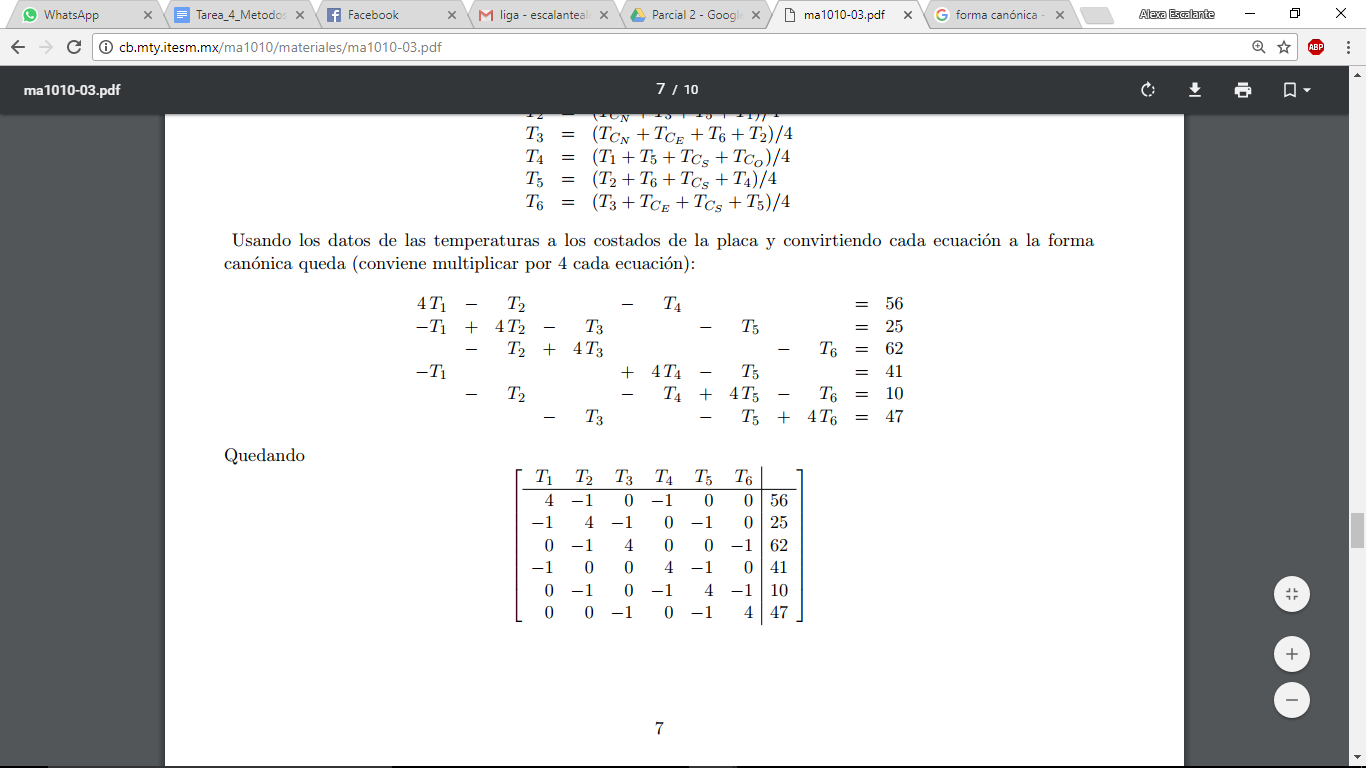
9. a(1,:)=a(1,:)/a(1,1);% ya buscado el mejor candidator, el reglon donde se quiere tener el "1" se divide entre el pivote(el mejor pivoteo ) ah esto de le llama pivoteo
10. for i=1:5
11. if i~=1
12. a(i,:)=a(i,:)-a(1,:)\*a(i,1);
13. end
14. end
15. a(2,:)=a(2,:)/a(2,2);
16. for i=1:5
17. if i~=2
18. a(i,:)=a(i,:)-a(2,:)\*a(i,2);
19. end
20. end
21. a(3,:)=a(3,:)/a(3,3);
22. for i=1:5
23. if i~=3
24. a(i,:)=a(i,:)-a(3,:)\*a(i,3);
25. end
26. end
27. a(4,:)=a(4,:)/a(4,4);
29. a(5,:)=a(5,:)/a(5,5);
30. for i=1:5
31. if i~=5
32. a(i,:)=a(i,:)-a(5,:)\*a(i,5);
33. end
34. end
35. % a
36. % inv(A)
37. A\_inv=a(:,6:10)
38. X=A\_inv\*B;
39. fprintf('El Resultado de la ecuación es la siguiente',X);
40. X



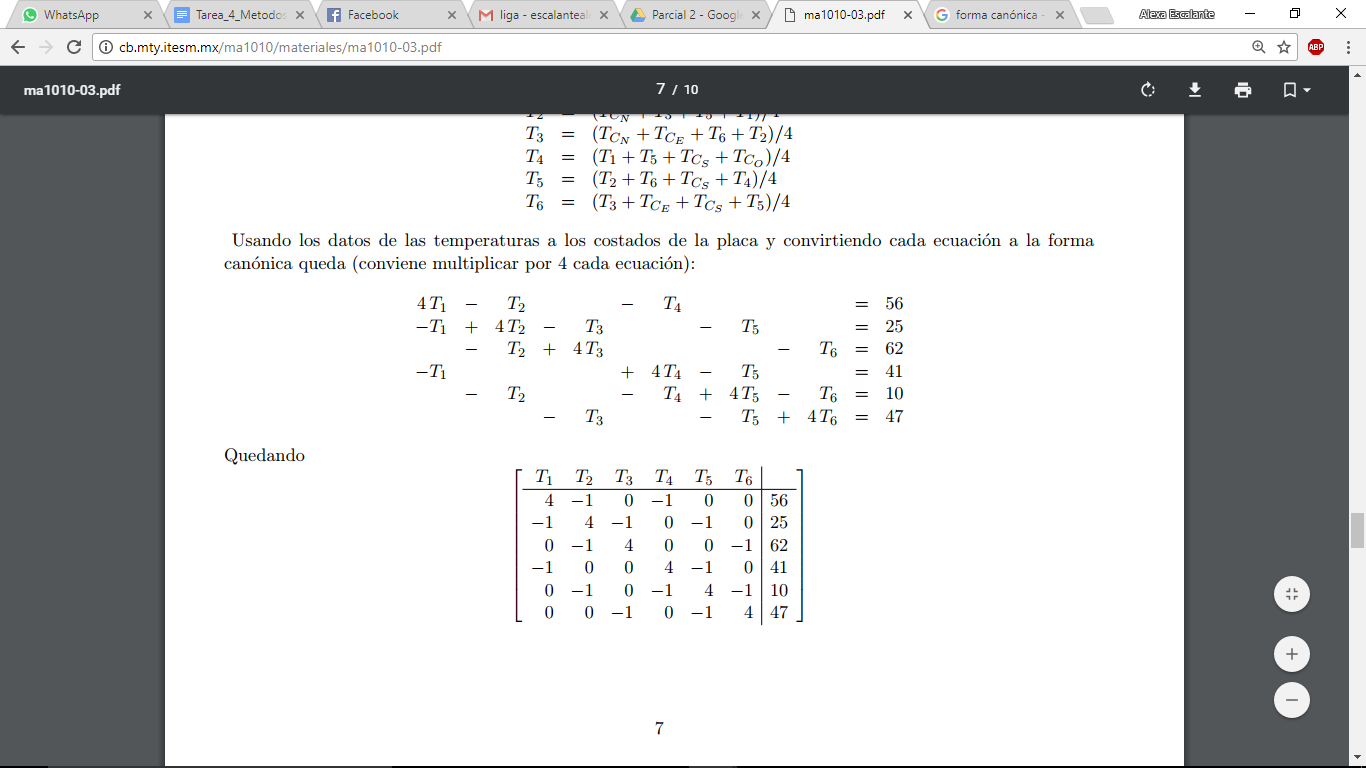
Teniendo en cuenta que:

De esta manera, sustituyendo los valores de las temperaturas dadas por el ejercicio:

Ahora se convierte la ecuación en su forma canónica,(es decir, que todas las ecuaciones contengan TODAS las variables involucradas) para eliminar nuestro divisor, multiplicamos todos los términos por 4, obteniendo que:



De esta manera ya podemos establecer nuestra matriz a resolver:



Código utilizado:

1. %EJERCICIO 5
2. %Método de GAUSS JORDAN
3. clear all;
4. %matriz de los coeficientes
5. A=[4,-1,0,-1,0,0;-1,4,-1,0,-1,0;0,-1,4,0,0,-1;-1,0,0,4,-1,0;0,-1,0,-1,4,-1;0,0,-1,0,-1,4];
6. %Matriz b
7. b=[56;25;62;41;10;47];
8. %Matriz aumentada
9. format long
10. a=[A b]
11. aux=a(6,:); %se guarda el renglón 6 en la variable auxiliar
12. a(6,:)=a(1,:);
13. a(1,:)=aux;% se vacía lo que había en la variable auxiliar
14. a(1,:)=a(1,:)/a(1,1);% ya buscado el mejor candidato, el renglón donde se quiere tener el "1" se divide entre el pivote(el mejor pivote)
15. for i=1:6
16. if i~=1
17. a(i,:)=a(i,:)-a(1,:)\*a(i,1);
18. end
19. end
20. a(2,:)=a(2,:)/a(2,2);
21. for i=1:6
22. if i~=2
23. a(i,:)=a(i,:)-a(2,:)\*a(i,2);
24. end
25. end
26. aux=a(3,:);
27. a(3,:)=a(4,:);
28. a(4,:)=aux;
29. a(3,:)=a(3,:)/a(3,3);
30. for i=1:6
31. if i~=3
32. a(i,:)=a(i,:)-a(3,:)\*a(i,3);
33. end
34. end
35. a(4,:)=a(4,:)/a(4,4);
36. for i=1:6
37. if i~=4
38. a(i,:)=a(i,:)-a(4,:)\*a(i,4);
39. end
40. end
41. a(5,:)=a(5,:)/a(5,5);
42. for i=1:6
43. if i~=5
44. a(i,:)=a(i,:)-a(5,:)\*a(i,5);
45. end
46. end
47. a(6,:)=a(6,:)/a(6,6);
48. for i=1:6
49. if i~=6
50. a(i,:)=a(i,:)-a(6,:)\*a(i,6);
51. end
52. end
53. x=a(:,7)
54. rref ([A b])

Entonces se obtiene como resultado las siguientes matrices:

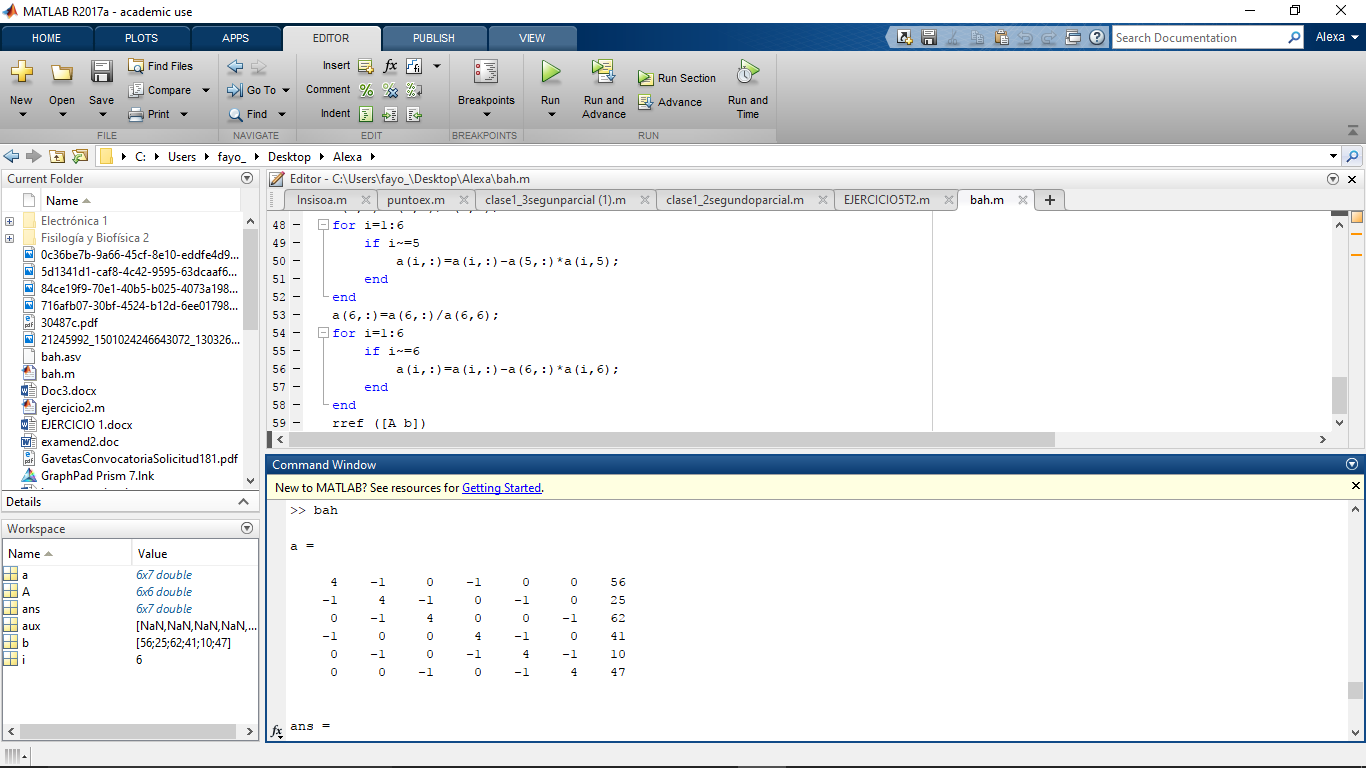
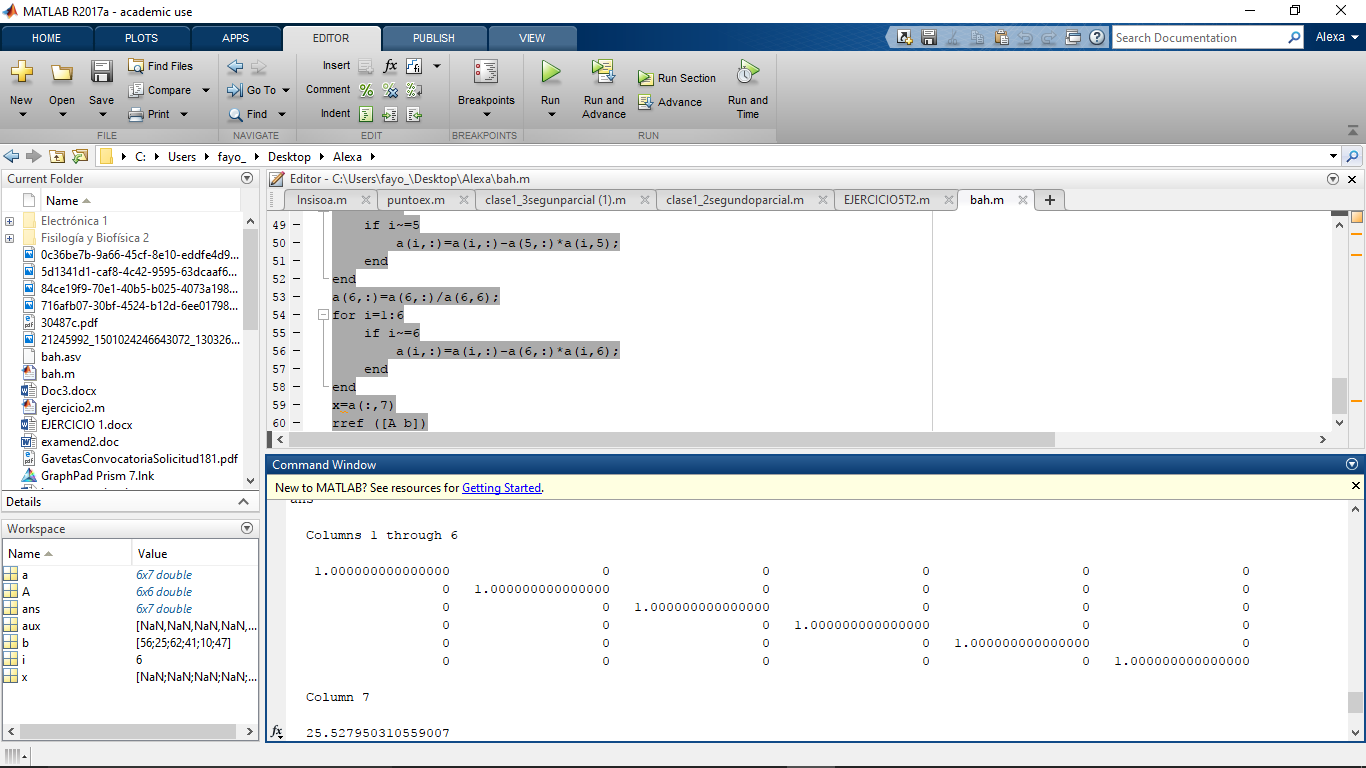


Fig 5.0 : Matriz inicial

Fig 5.1: Matriz de Gauss-Jordan

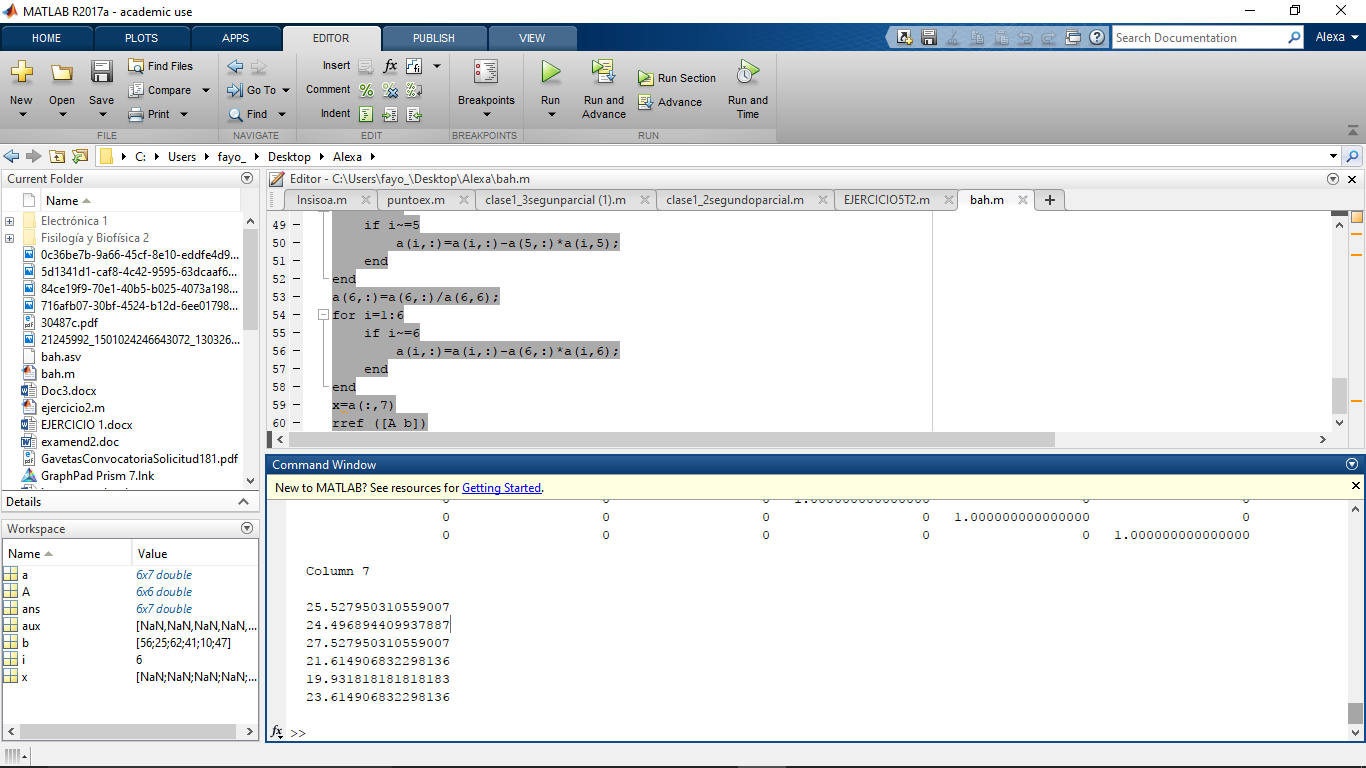


Fig 5.2: Matriz de resultados de las temperaturas.

Solución:

T1 = 25.527°C

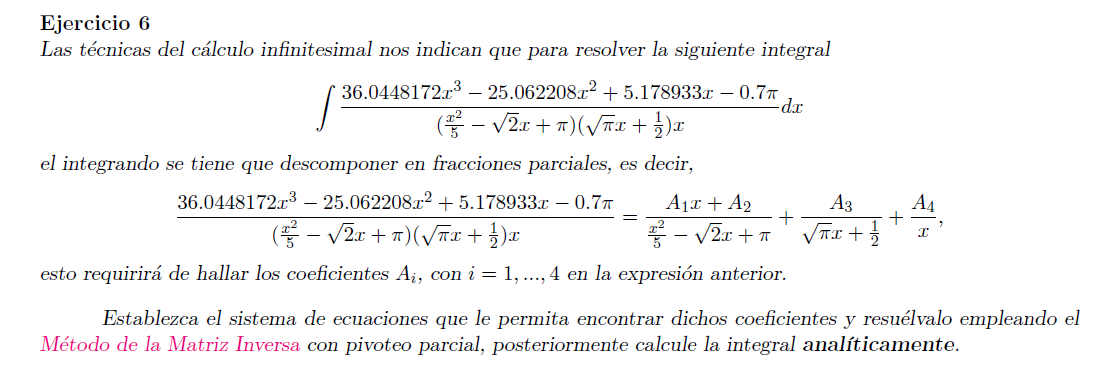
T2 = 24.496 °C

T3 = 27.527 °C

T4 = 21.614 °C

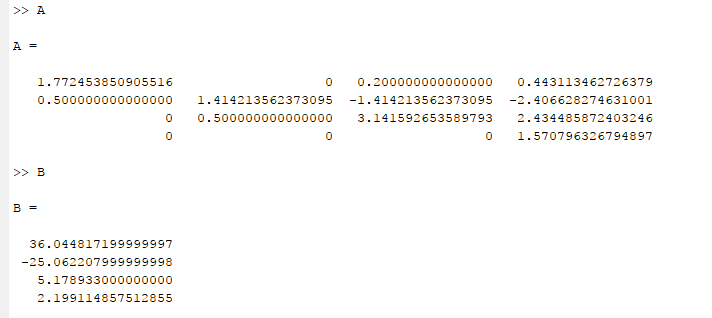
T5 = 19.931°C

T6 = 23.614°C

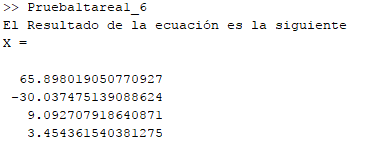


Obteniendo las ecuaciones por fracciones parciales, tenemos que:

Introduciendo las matrices obtenidas, la matriz A y la matriz solución:



Los coeficientes son los siguientes



1. B=[36.0448172000000;-25.0622080000000;5.17893300000000;2.19911485751286];
2. I=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1]
3. format long
4. a=[A I];
5. a=[A eye(size(a))];
6. a(1,:)=a(1,:)/a(1,1);% ya buscado el mejor candidato, el renglón donde se
7. %quiere tener el "1" se divide entre el pivote(el
8. %mejor pivoteo ) ah esto se le llama pivoteo
9. for i=1:4
10. if i~=1
11. a(i,:)=a(i,:)-a(1,:)\*a(i,1);
12. end
13. end
14. a(2,:)=a(2,:)/a(2,2);
15. for i=1:4
16. if i~=2
17. a(i,:)=a(i,:)-a(2,:)\*a(i,2);
18. end
19. end
20. a(3,:)=a(3,:)/a(3,3);
21. for i=1:4
22. if i~=3
23. a(i,:)=a(i,:)-a(3,:)\*a(i,3);
24. end
25. end
26. a(4,:)=a(4,:)/a(4,4);
27. for i=1:4
28. if i~=4
29. a(i,:)=a(i,:)-a(4,:)\*a(i,4);
30. end
31. end
32. % % inv(A)
33. A\_inv=a(:,5:8);
34. X=A\_inv\*B;
35. fprintf('El Resultado de la ecuación es la siguiente',X);
36. X