

SISTEMA DE ECUACIONES LINEALES



LUIS CARLOS SANCHEZ

JULIAN DAVID LEDEZMA

ANDRES FELIPE ESCALLON

Ingeniero

UNIVERSIDAD COOPERATIVA DE COLOMBIA

FACULTAD DE INGENIERÍA

VI INGENIERIA DE SISTEMAS

AREA DE ANALISIS NÚMÉRICO

POPAYÁN

2010

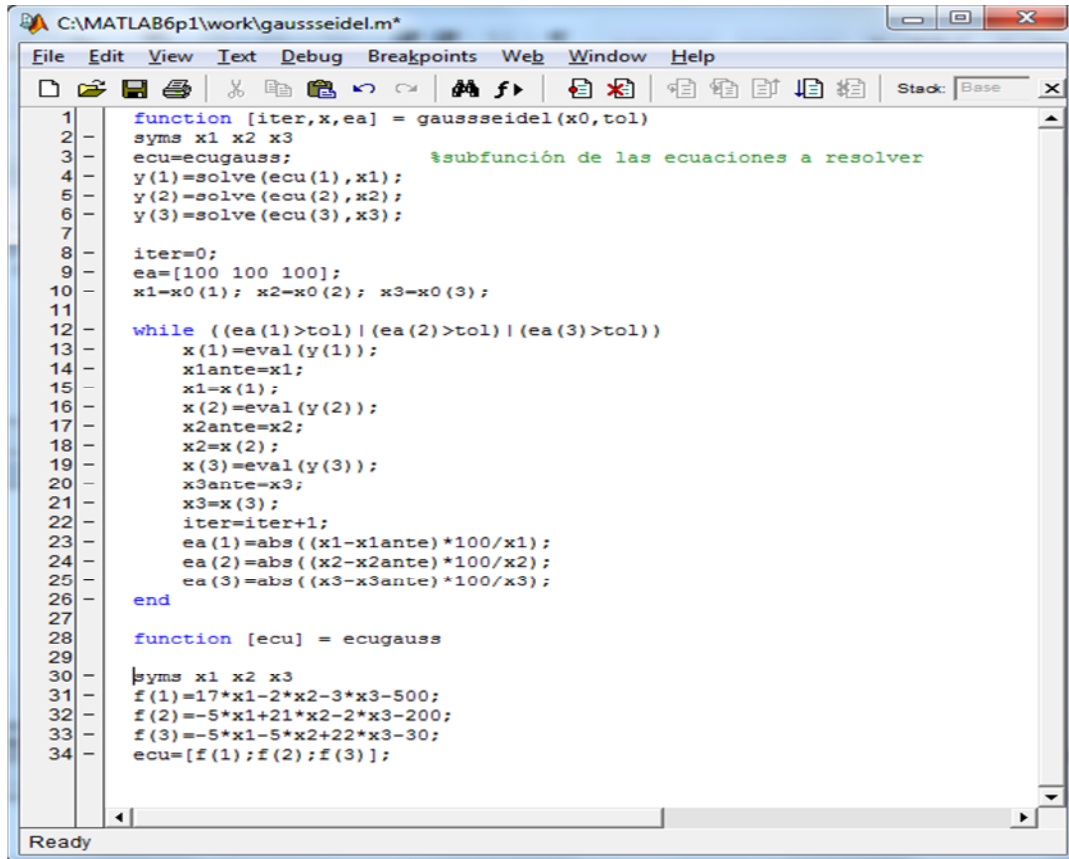
Método Gauss Seidel

Se utilizó el respectivo código del Método de Gauss Seidel implementado en MATLAB para la solución del siguiente sistema de ecuaciones lineales propuestas.

$$\begin{aligned}17c_1 - 2c_2 - 3c_3 &= 500 \\ -5c_1 - 5c_2 - 22c_3 &= 30 \\ -5c_1 + 21c_2 - 2c_3 &= 200\end{aligned}$$

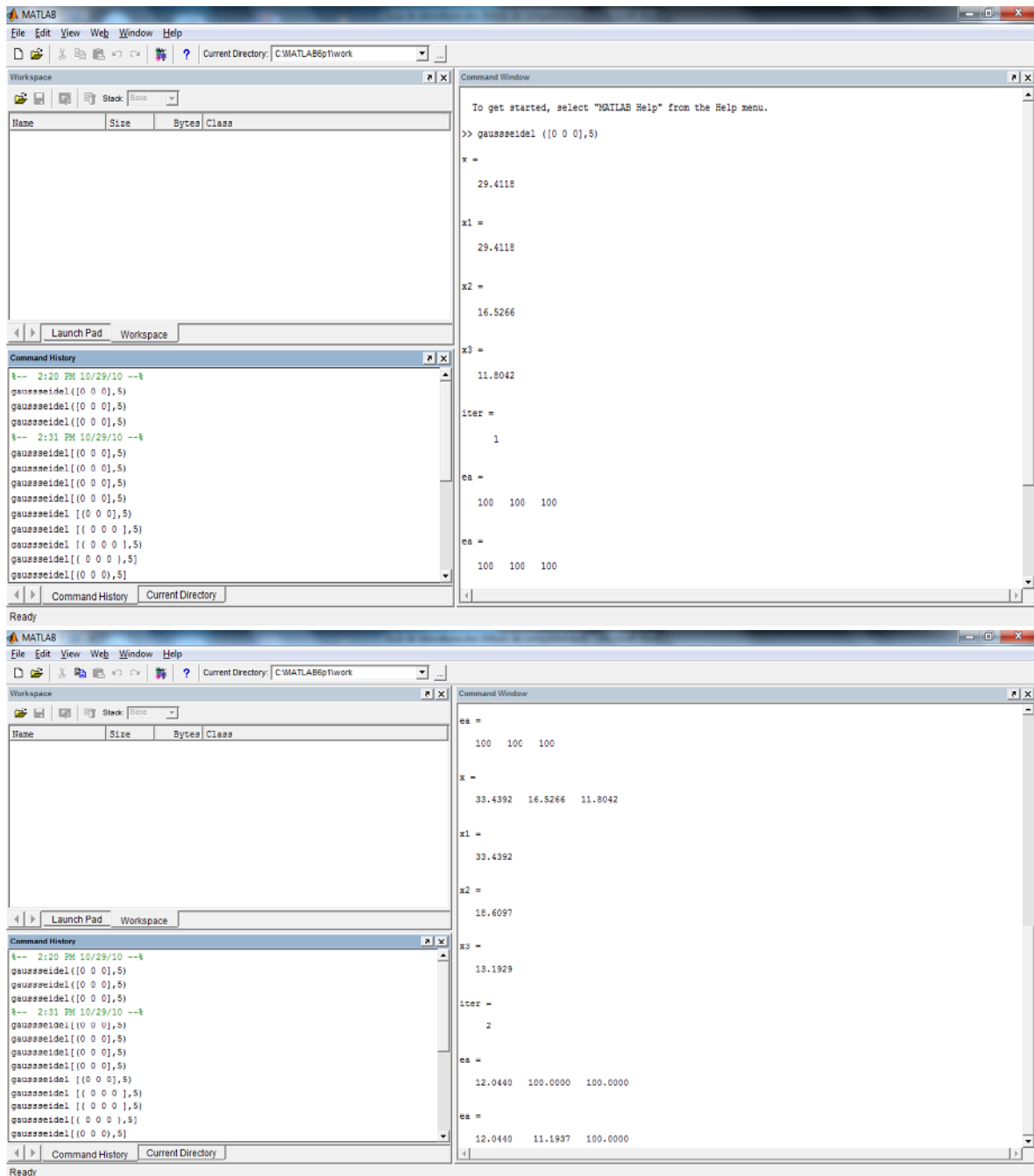
Como las dos ultimas no cumplen se re organizaron para poder encontrar la solución correcta de la siguiente manera:

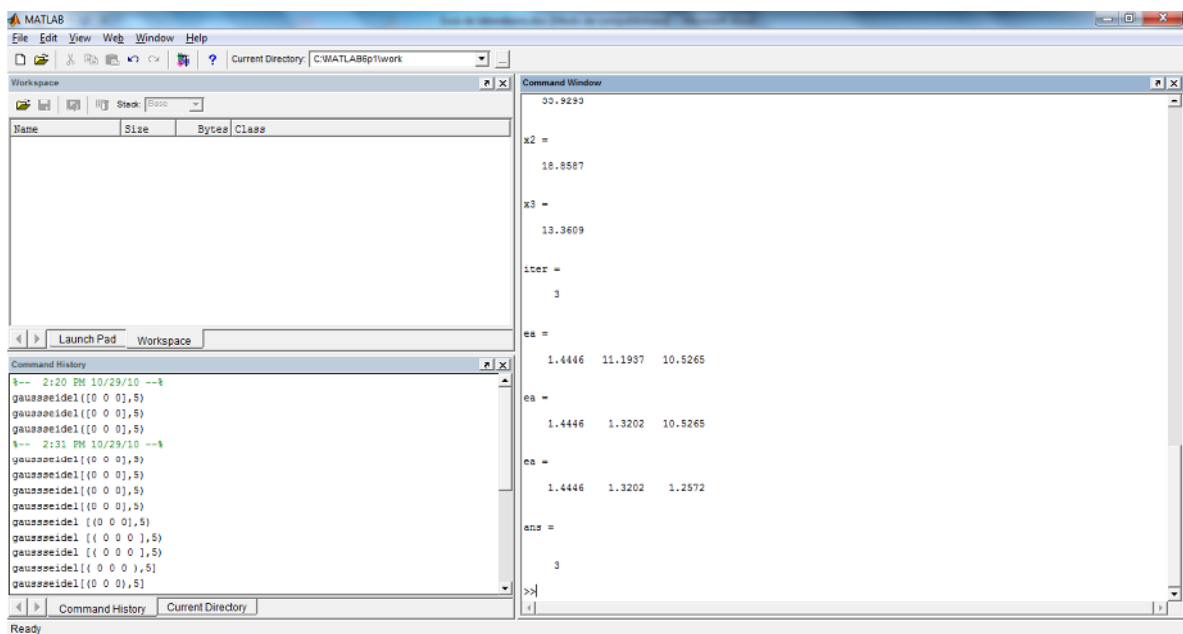
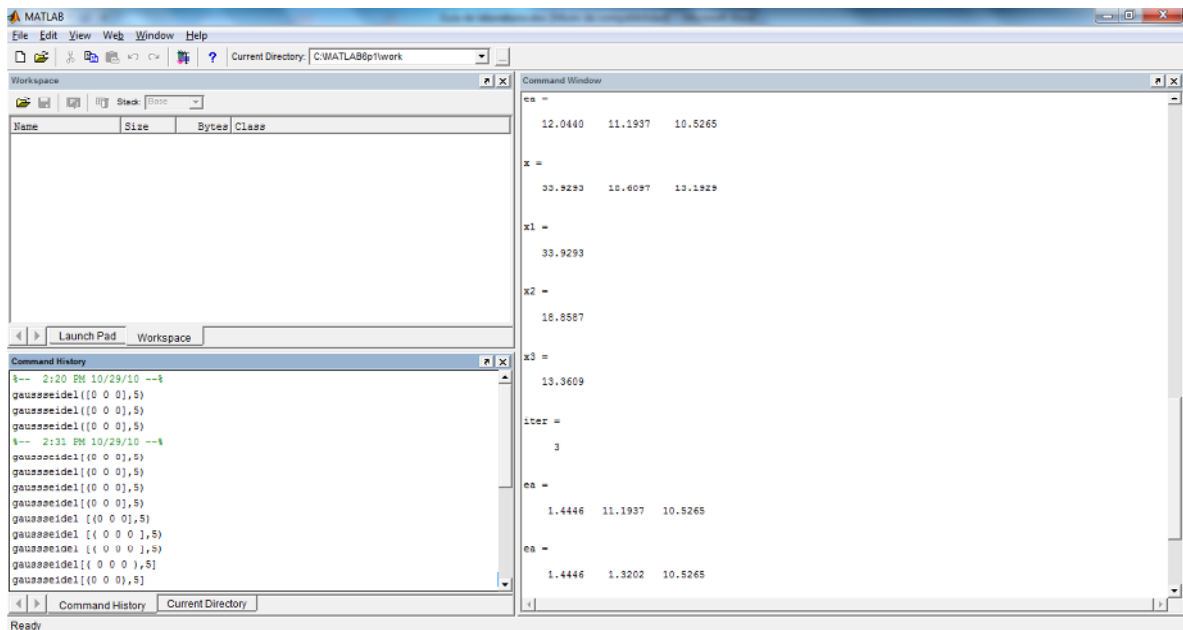
$$\begin{aligned}17c_1 - 2c_2 - 3c_3 &= 500 \\ -5c_1 + 21c_2 - 2c_3 &= 200 \\ -5c_1 - 5c_2 - 22c_3 &= 30\end{aligned}$$



```
1 function [iter,x,ea] = gaussseidel(x0,tol)
2 syms x1 x2 x3
3 ecu=ecugauss; %subfunción de las ecuaciones a resolver
4 y(1)=solve(ecu(1),x1);
5 y(2)=solve(ecu(2),x2);
6 y(3)=solve(ecu(3),x3);
7
8 iter=0;
9 ea=[100 100 100];
10 x1=x0(1); x2=x0(2); x3=x0(3);
11
12 while ((ea(1)>tol) | (ea(2)>tol) | (ea(3)>tol))
13 x(1)=eval(y(1));
14 x1ante=x1;
15 x1=x(1);
16 x(2)=eval(y(2));
17 x2ante=x2;
18 x2=x(2);
19 x(3)=eval(y(3));
20 x3ante=x3;
21 x3=x(3);
22 iter=iter+1;
23 ea(1)=abs((x1-x1ante)*100/x1);
24 ea(2)=abs((x2-x2ante)*100/x2);
25 ea(3)=abs((x3-x3ante)*100/x3);
26 end
27
28 function [ecu] = ecugauss
29
30 syms x1 x2 x3
31 f(1)=17*x1-2*x2-3*x3-500;
32 f(2)=-5*x1+21*x2-2*x3-200;
33 f(3)=-5*x1-5*x2-22*x3-30;
34 ecu=[f(1);f(2);f(3)];
```

Ejecución del programa





Al ejecutar el programa en la 3ra iteración se hallan los valores de c_1 , c_2 y c_3 así:

$c_1 = 33.9293$

$c_2 = 18.8587$

$c_3 = 13.3609$

Método de Jacobi

Se utilizó el respectivo código del Método de Jacobi implementado en MATLAB para la solución del siguiente sistema de ecuaciones lineales propuestas. (Ejercicio 4)

$$-5x_1 + 12x_2 = 80$$

$$4x_1 - x_2 - x_3 = -2$$

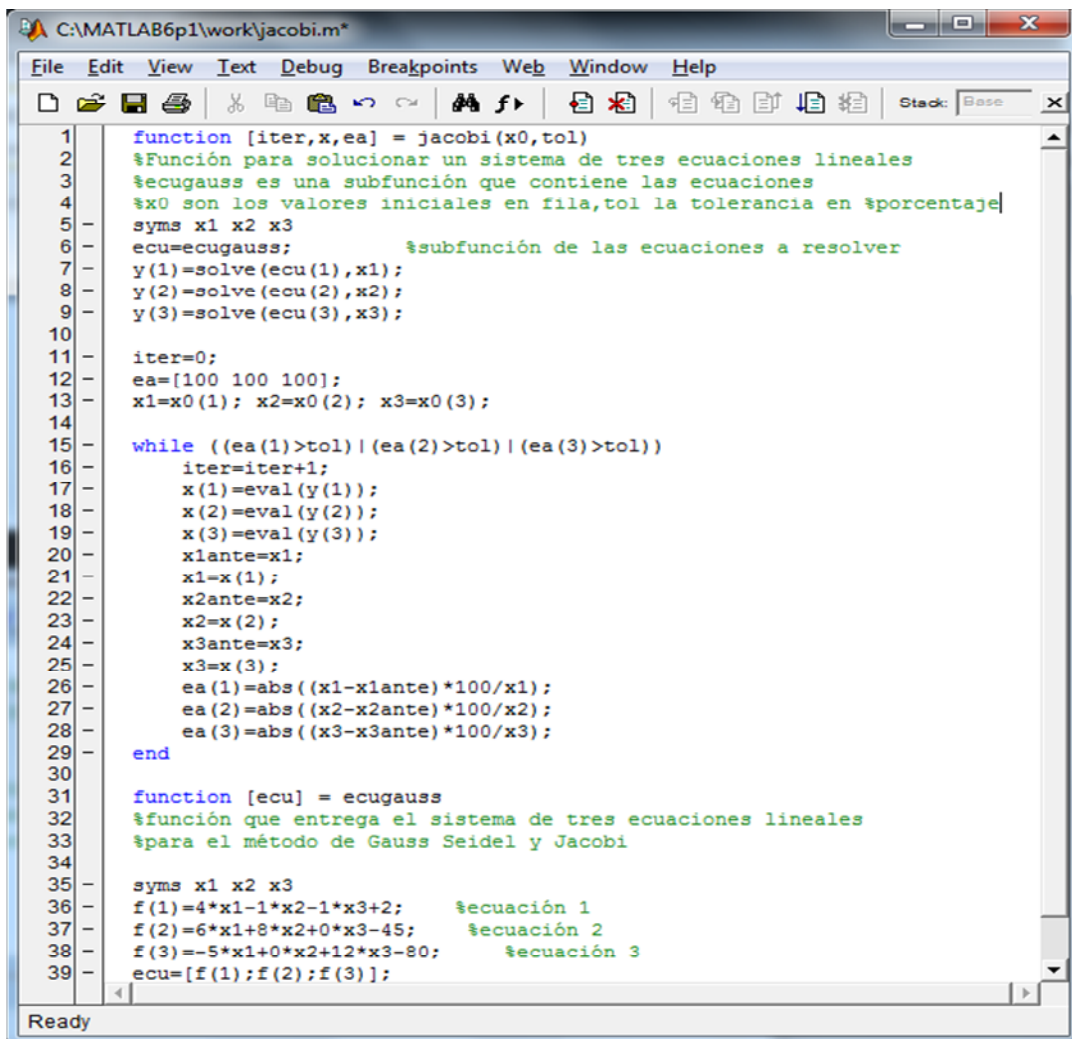
$$6x_1 + 8x_2 = 45$$

Las ecuaciones se organizaron de la siguiente manera para aplicar el respectivo código:

$$4x_1 - x_2 - x_3 = -2$$

$$6x_1 + 8x_2 = 45$$

$$-5x_1 + 12x_2 = 80$$



```
C:\MATLAB6p1\work\jacobi.m*
File Edit View Text Debug Breakpoints Web Window Help
[Icons] Stack: Base
1 function [iter,x,ea] = jacobi(x0,tol)
2 %Función para solucionar un sistema de tres ecuaciones lineales
3 %ecugauss es una subfunción que contiene las ecuaciones
4 %x0 son los valores iniciales en fila,tol la tolerancia en %porcentaje|
5 syms x1 x2 x3
6 ecu=ecugauss; %subfunción de las ecuaciones a resolver
7 y(1)=solve(ecu(1),x1);
8 y(2)=solve(ecu(2),x2);
9 y(3)=solve(ecu(3),x3);
10
11 iter=0;
12 ea=[100 100 100];
13 x1=x0(1); x2=x0(2); x3=x0(3);
14
15 while ((ea(1)>tol) | (ea(2)>tol) | (ea(3)>tol))
16 iter=iter+1;
17 x(1)=eval(y(1));
18 x(2)=eval(y(2));
19 x(3)=eval(y(3));
20 x1ante=x1;
21 x1=x(1);
22 x2ante=x2;
23 x2=x(2);
24 x3ante=x3;
25 x3=x(3);
26 ea(1)=abs((x1-x1ante)*100/x1);
27 ea(2)=abs((x2-x2ante)*100/x2);
28 ea(3)=abs((x3-x3ante)*100/x3);
29 end
30
31 function [ecu] = ecugauss
32 %función que entrega el sistema de tres ecuaciones lineales
33 %para el método de Gauss Seidel y Jacobi
34
35 syms x1 x2 x3
36 f(1)=4*x1-1*x2-1*x3+2; %ecuación 1
37 f(2)=6*x1+8*x2+0*x3-45; %ecuación 2
38 f(3)=-5*x1+0*x2+12*x3-80; %ecuación 3
39 ecu=[f(1);f(2);f(3)];
Ready
```

Ejecución del programa

The screenshot shows the MATLAB Command Window with the following output:

```
>> jacobi([0 0 0],5)

iter =

     1

x1 =

    -0.5000

x2 =

     5.6280

x3 =

     6.6667

ea =

    100    100    100

iter =

     2

x1 =

     2.5729

x2 =

     -
```

The Workspace window shows a variable `ans` of size `1x1` and class `double array`. The Command History window shows the command `jacobi([0 0 0],5)`.

The screenshot shows the MATLAB Command Window with the following output:

```
x2 =

     6

x3 =

     6.4583

ea =

    119.4332    100.0000    100.0000

ea =

    119.4332     6.2500     3.2258

iter =

     3

x1 =

     2.6146

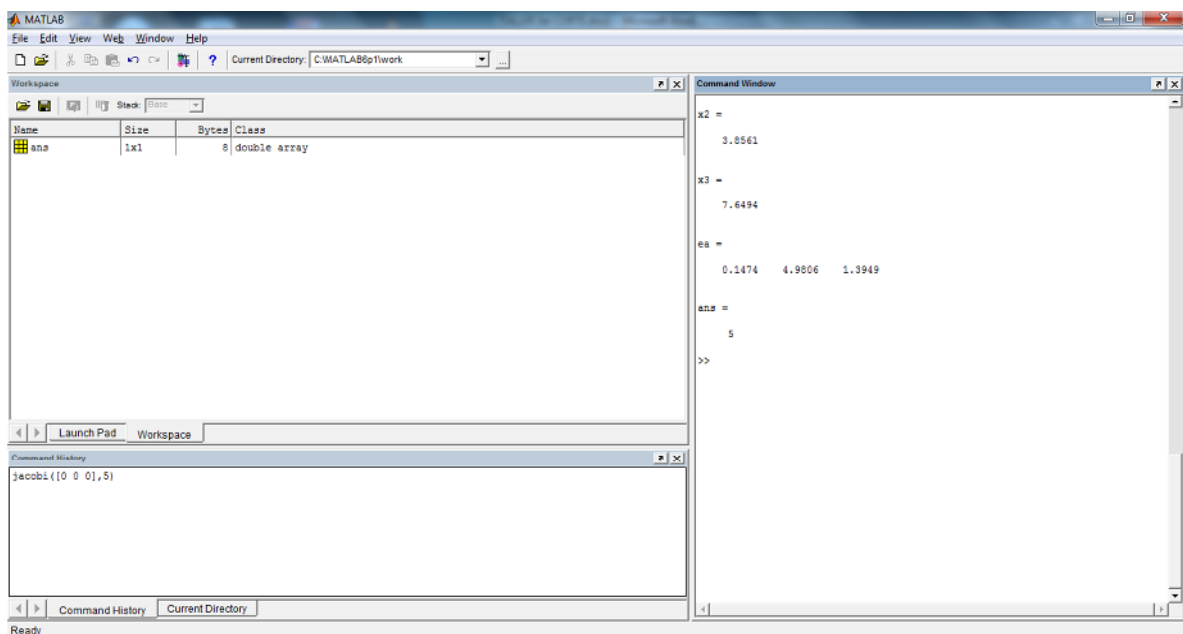
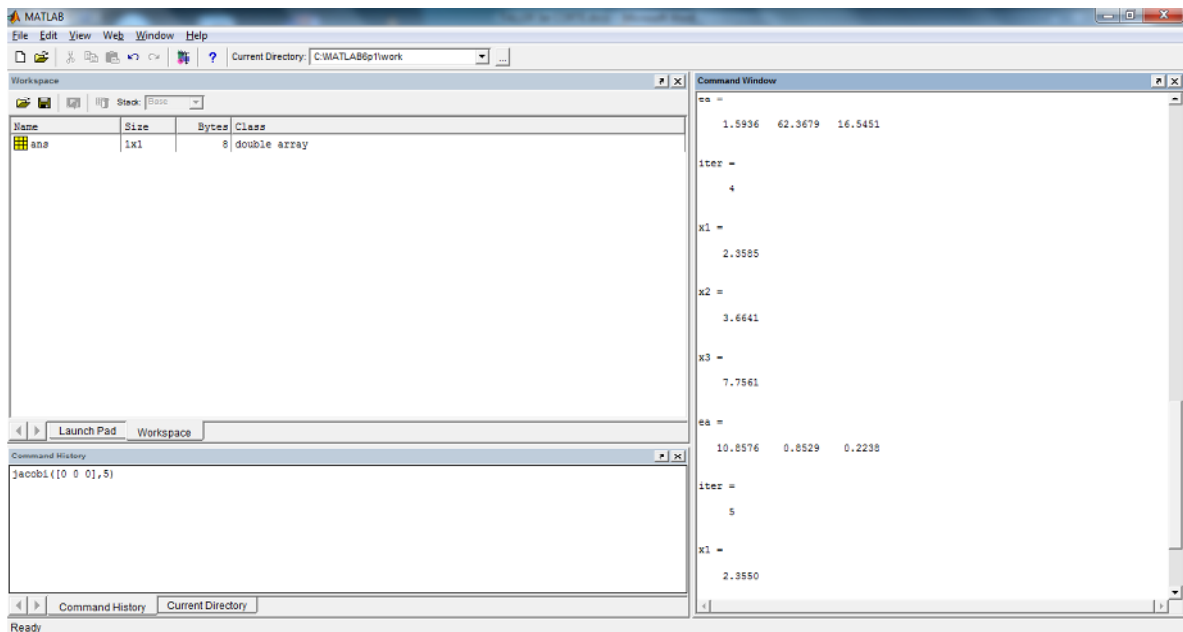
x2 =

     3.6953

x3 =

     7.7381
```

The Workspace window shows a variable `ans` of size `1x1` and class `double array`. The Command History window shows the command `jacobi([0 0 0],5)`.



Al ejecutar el programa en la 3ra iteración se hallan los valores de x_1 , x_2 y x_3 así:

$$x_1 = 2.3550 \quad x_2 = 3.8561 \quad x_3 = 7.6494$$