

Ecuaciones Lineales

```
Cramer.m x NewtonRaphson.m x Jacobi.m x GaussSeidel.m x +
1 clear
2 A = [ 2 -1 5;
3       -2 -7 -9];
4
5 b = [-3; 0;];
6
7 x = zeros(3,1);
8
9
10 i=1;
11 xi
12 Am = A;
13 Am(:,i) = b;
14 x(i) = det(Am)/det(A)
15
16 i=2;
17
18 % xi
19 Am = A;
20 Am(:,i) = b;
21 x(i) = det(Am)/det(A)
22
23 i=3;
24 % xi
25 Am = A;
26 Am(:,i) = b;
27 x(i) = det(Am)/det(A)
```

```
Cramer.m x NewtonRaphson.m x Jacobi.m x GaussSeidel.m x +
1 for m=1:mmax02
2     error=0; y = x
3     for i=1:n
4         xi = b
5         for j= 1 :i - 1
6             xi =xi - ax
7         end
8         for j = i+ 1 :n
9             xi = xi - ay
10        end
11        xi = xi/a
12        error=error+|yi - xi|
13    end
14    if error<eps1 + eps2*norm(x)
15        return
16    end
17 end
18 disp('numero m áximo de iteraciones alcanzado')
19 return
```

```

Cramer.m x NewtonRaphson.m x Jacobi.m x GaussSeidel.m x +
1  for m = 1: mmax
2      error = 0; y = x
3      for i = 1:n
4          xi = b
5          for j = 1: i - 1
6              xi = xi - xa
7          end
8          for j = i + 1: n
9              xi = xi - ya
10         end
11         xi = xi / a
12         error = error + |yi - xi|
13     end
14     if error < eps1 + eps2*norm(x)
15         return
16     end
17 end
18 disp('numero maximo de iteraciones alcanzado')
19 return

```

Ecuaciones No Lineales

```

Cramer.m x NewtonRaphson.m x Jacobi.m x GaussSeidel.m x +
1  clc
2
3  F= @(x) [
4      x(1)^3 + x(2)^3 - 129;
5      x(1)^2 + x(2)^2 - 9.75;
6      ];
7
8  J= @(x) [
9      2*x(1),    2*x(2),    -2*x(3);
10     1,          1,        -1;
11     ];
12
13  x = [4; 2;];
14
15  error = 1e3;
16  tol = 1e-5;
17  n = 0;
18  |
19  while error > tol
20      dx = -J(x)\F(x);
21      error = norm(dx)/norm(x);
22      x = x + dx;
23      n = n+1;
24  end
25  fprintf('Iteraciones : %d \n',n);
26
27  fprintf('Respuesta : %f \n', x.');
```