

**ITESO**

**MÉTODOS NUMÉRICOS**

**LÓPEZ LAZARENO DIEGO ALBERTO IF722100**

**PRÁCTICA 06**

1. Desarrolla un programa que resuelva sistemas de ecuaciones no lineales mediante el método de Newton-Raphson.

% Función que contiene a la matriz de Jacobi

function matriz\_jacobi=matriz\_jacobi(x,y)

a(1,1)=2\*x+y;

a(1,2)=x;

a(2,1)=4\*(y^2);

a(2,2)=1+8\*x\*y;

matriz\_jacobi=a;

end

% Función que evalúa las dos funciones en los puntos (x,y) dados

function funciones\_vectoriales=funciones\_vectoriales(x,y)

a(1,1)=(x^2)+(x\*y)-5;

a(2,1)=y+(4\*x\*(y^2))-39;

funciones\_vectoriales=a;

end

% Aproximaciones iniciales para las soluciones (x,y) del sistema de ecuaciones

x=input("Ingrese su valor inicial de x para empezar la aproximación");

y=input("Ingrese su valor inicial de y para empezar la aproximación");

% Matriz que guarda las aproximaciones

aproximaciones=1;

% Ciclo para aproximar las soluciones al sistema de ecuaciones

for i=1:10

operacion\_matricial=inv(matriz\_jacobi(x,y))\*funciones\_vectoriales(x,y);

aproximaciones(1,1)=x-operacion\_matricial(1,1);

aproximaciones(2,1)=y-operacion\_matricial(2,1);

x=aproximaciones(1,1);

y=aproximaciones(2,1);

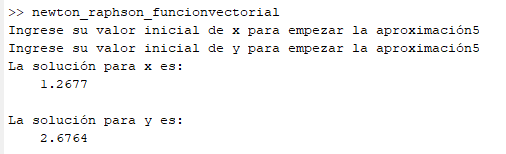
end

disp("La solución para x es:")

disp(x)

disp("La solución para y es:")

disp(y)



**Conclusión**

En esta práctica de laboratorio se abordó al Método de Newton-Raphson en su forma vectorial para la resolución de sistemas de ecuaciones no lineales. Dicho algoritmo converge rápidamente a un vector solución, sin embargo, su programación puede ser un tanto tediosa, sobre todo si se tienen sistemas de ecuaciones no lineales muy grandes y de numerosas variables; pues se necesitarán calcular muchas derivadas parciales para obtener la matriz de Jacobi. Por último, me parece interesante cómo un problema de sistemas de ecuaciones se convierte en uno de encontrar raíces, ya que ambas funciones vectoriales se dejan igualadas a 0.