clc

clear all

Max\_iterations=50; % Maximum Number of Iterations

correction\_factor = 2.0; % Correction factor

inertia = 1.0; % Ineritia Coeffecient

swarm\_size = 5; % Number of particles

LB=[-5.12 -5.12]; % Lower Boundaries

UB=[5.12 5.12]; % Upper Boundaries

xrange=UB(1)-LB(1);

yrange=UB(2)-LB(2);

% Initial Positions

swarm(:, 1, 1)=rand(1,swarm\_size)\*xrange+LB(1);

swarm(:, 1, 2)=rand(1,swarm\_size)\*yrange+LB(2);

% Initial best value so far

swarm(:, 4, 1) = 1000;

% Initial velocity

swarm(:, 2, :) = 0;

for iter = 1 : Max\_iterations

% Calculating fitness value for all particles

for i = 1 : swarm\_size

swarm(i, 1, 1) = swarm(i, 1, 1) + swarm(i, 2, 1)/1.3; %update x position

swarm(i, 1, 2) = swarm(i, 1, 2) + swarm(i, 2, 2)/1.3; %update y position

% The fitness function (DeJong) F(x,y)=x^2+y^2

Fval = (swarm(i, 1, 1))^2 + (swarm(i, 1, 2))^2; % fitness evaluation (you may replace this objective function with any function having a global minima)

% If the fitness value for this particle is better than the

% best fitness value of that particle exchange both values

if Fval < swarm(i, 4, 1) % if new position is better

swarm(i, 3, 1) = swarm(i, 1, 1); % Update the position of the first dimension

swarm(i, 3, 2) = swarm(i, 1, 2); % Update the position of the second dimension

swarm(i, 4, 1) = Fval; % Update best value

end

end

% Search for the global best solution

[temp, gbest] = min(swarm(:, 4, 1)); % global best position

% Updating velocity vectors

for i = 1 : swarm\_size

swarm(i, 2, 1) = rand\*inertia\*swarm(i, 2, 1) + correction\_factor\*rand\*(swarm(i, 3, 1) - swarm(i, 1, 1)) + correction\_factor\*rand\*(swarm(gbest, 3, 1) - swarm(i, 1, 1)); %x velocity component

swarm(i, 2, 2) = rand\*inertia\*swarm(i, 2, 2) + correction\_factor\*rand\*(swarm(i, 3, 2) - swarm(i, 1, 2)) + correction\_factor\*rand\*(swarm(gbest, 3, 2) - swarm(i, 1, 2)); %y velocity component

end

% Store the best fitness valuye in the convergence curve

ConvergenceCurve(iter,1)=swarm(gbest,4,1);

disp(['Iterations No. ' int2str(iter) ' , the best fitness value is ' num2str(swarm(gbest,4,1))]);

end

% Plot convergence curve

plot(ConvergenceCurve,'r-')

title('Convergence Curve')

xlabel('Iterations')

ylabel('Fitness Value')