**21070122154 Shaurya Gupta Assignment 10 OTAL**

**Q1) Briefly describe the properties of chaotic sequences.**

Chaotic functions exhibit certain key properties that set them apart from other mathematical functions. Here are the essential characteristics:

* A small change in the starting point of a chaotic system can lead to significantly different outcomes. This is often referred to as the "butterfly effect." In chaotic systems, predicting long-term behavior becomes almost impossible due to this sensitivity.
* Despite being governed by deterministic rules, chaotic functions generate outcomes that appear random and unpredictable over time. Their unpredictability stems from the system’s complexity and sensitivity rather than randomness.
* Chaotic systems are typically nonlinear, meaning the output is not directly proportional to the input. Nonlinear equations and interactions between variables contribute to the chaotic behavior, as small variations can cause significant changes in system behavior.
* Chaotic functions do not repeat their states in a predictable, periodic cycle. While they may exhibit patterns or attractors, these patterns do not loop back into a precise repetition over time.
* Chaotic systems often produce fractals, structures that are self-similar across different scales. The intricate and seemingly random patterns at different levels are a hallmark of chaotic behavior.
* Chaotic systems exhibit a "mixing" behavior where, over time, different parts of the system become intertwined. This means that neighboring points in the system become increasingly separated and mixed throughout the phase space.
* In chaotic functions, orbits (trajectories of the system) are dense in a specific space, meaning that the system can come arbitrarily close to any point within that space. This behavior contributes to the unpredictability of chaotic systems.

These properties of chaotic functions are fundamental in fields like weather prediction, stock market analysis, and nature-inspired algorithms, where chaotic systems can model real-world unpredictable phenomena effectively.

**Q2) Implement the following chaotic sequences to replace random numbers used at some of the steps of PSO.**

**a. Logistic Map**

**b. Circle Map**

**c. Kent Map**

**d. Piecewise Map**

**e. Sine Map**

**f. Sinusoidal Map**

**Compare the results of the basic PSO and all the six chaotic variants for any 5 test functions on the following parameters for 1000 iterations:**

**a) Mean function values for 25 runs**

**b) Best function values for 25 runs**

**c) Standard deviation values for 25 runs Compare the iteration-wise mean function values plots for all the variants.**

**Code:**

clc;

clear;

function [Cha] = LogisticMap(n)

x = ones(1, n);

x(1) = rand;

A = [0.00 0.25 0.50 0.75 1.00];

while ismember(x(1), A)

x(1) = rand;

end

for i = 1:n-1

x(i+1) = 4 \* x(i) \* (1 - x(i));

end

Cha = x;

end

function [Cha] = CircleMap(n)

x = ones(1, n);

x(1) = rand;

for i = 1:n-1

x(i+1) = mod(x(i) + 0.1, 1);

end

Cha = x;

end

function [Cha] = KentMap(n)

x = ones(1, n);

m = rand;

x(1) = rand;

for i = 1:n-1

if x(i) <= m

x(i+1) = x(i) / m;

else

x(i+1) = (1 - x(i)) / (1 - m);

end

end

Cha = x;

end

function [Cha] = PiecewiseMap(n)

x = ones(1, n);

x(1) = rand;

for i = 1:n-1

if x(i) <= 0.7

x(i+1) = x(i) / 0.7;

else

x(i+1) = (1 - x(i)) / (1 - 0.7);

end

end

Cha = x;

end

function [Cha] = SineMap(n)

x = ones(1, n);

x(1) = rand;

for i = 1:n-1

x(i+1) = sin(pi \* x(i));

end

Cha = x;

end

function [Cha] = SinosoidalMap(n)

x = ones(1, n);

x(1) = 0.55;

for i = 1:n-1

x(i+1) = 2.3 \* x(i)^2 \* sin(pi \* x(i));

end

Cha = x;

end

MaxIter = 1000;

str = '-20\*exp(-0.2\*sqrt((x1^2 + x2^2 + x3^2 + x4^2 + x5^2)/5)) - exp((cos(2\*pi\*x1) + cos(2\*pi\*x2) + cos(2\*pi\*x3) + cos(2\*pi\*x4) + cos(2\*pi\*x5))/5) + 20 + exp(1)';

D = 5;

Npop = 50;

nRuns = 25;

LB = [-10, -10, -10, -10, -10];

UB = [10, 10, 10, 10, 10];

syms x [1 D];

fitnessFunction = str2sym(str);

f = matlabFunction(fitnessFunction, 'vars', {x});

PSO\_bestFitness = zeros(MaxIter, 6);

PSO\_meanFitness = zeros(MaxIter, 6);

PSO\_stdFitness = zeros(1, 6);

PSO\_globalBest = zeros(1, 6);

PSO\_meanLast = zeros(1, 6);

PSO\_stdLast = zeros(1, 6);

PSO\_globalBestLast = zeros(1, 6);

% Chaotic map functions

chaoticMaps = {@LogisticMap, @CircleMap, @KentMap, @PiecewiseMap, @SineMap, @SinosoidalMap};

mapNames = {'LogisticMap', 'CircleMap', 'KentMap', 'PiecewiseMap', 'SineMap', 'SinosoidalMap'}; % Function names

for mapIdx = 1:length(chaoticMaps)

chaoticMapFunction = chaoticMaps{mapIdx};

bestFitnessOverAllRuns = zeros(MaxIter, nRuns);

for run = 1:nRuns

[bestFitnessOverTime, globalBestFitness] = particleSwarmOptimization(f, Npop, MaxIter, D, LB, UB, chaoticMapFunction);

bestFitnessOverAllRuns(:, run) = bestFitnessOverTime;

end

PSO\_bestFitness(:, mapIdx) = min(bestFitnessOverAllRuns, [], 2);

PSO\_meanFitness(:, mapIdx) = mean(bestFitnessOverAllRuns, 2);

PSO\_stdFitness(mapIdx) = std(bestFitnessOverAllRuns(:));

PSO\_globalBest(mapIdx) = min(PSO\_bestFitness(:, mapIdx));

% For the final iteration (last iteration)

PSO\_meanLast(mapIdx) = PSO\_meanFitness(MaxIter, mapIdx);

PSO\_stdLast(mapIdx) = PSO\_stdFitness(mapIdx);

PSO\_globalBestLast(mapIdx) = PSO\_bestFitness(MaxIter, mapIdx);

disp(['Standard Deviation of Global Best Fitness for ', func2str(chaoticMaps{mapIdx}), ': ', num2str(PSO\_stdFitness(mapIdx))]);

end

% Create the table for SD, Mean, and Global Best Fitness at the last iteration

T = table(mapNames', PSO\_stdLast', PSO\_meanLast', PSO\_globalBestLast', ...

'VariableNames', {'Map', 'StandardDeviation', 'MeanFitness', 'GlobalBestFitness'});

disp(T);

% Plotting

figure;

hold on;

colors = lines(length(chaoticMaps));

for i = 1:length(chaoticMaps)

plot(1:MaxIter, PSO\_bestFitness(:, i), 'Color', colors(i, :), 'LineWidth', 2, 'DisplayName', mapNames{i});

end

xlabel('Iteration');

ylabel('Best Fitness Value');

title('Best Fitness Value Over Iterations for Different Chaotic Maps');

legend('show');

grid on;

figure;

hold on;

for i = 1:length(chaoticMaps)

plot(1:MaxIter, PSO\_meanFitness(:, i), 'Color', colors(i, :), 'LineWidth', 2, 'DisplayName', mapNames{i});

end

xlabel('Iteration');

ylabel('Mean Fitness Value');

title('Mean Fitness Value Over Iterations for Different Chaotic Maps');

legend('show');

grid on;

function [bestFitnessOverTime, globalBestFitness] = particleSwarmOptimization(fitnessFunction, nParticles, maxIterations, nDimensions, LB, UB, chaoticMapFunction)

cognitiveCoeff = 2;

socialCoeff = 2;

inertiaWeight = 0.5;

positions = LB + (UB - LB) .\* rand(nParticles, nDimensions);

velocities = rand(nParticles, nDimensions);

fitness = arrayfun(@(i) fitnessFunction(positions(i,:)), 1:nParticles, 'UniformOutput', false);

fitness = cell2mat(fitness);

pBestPositions = positions;

pBestFitness = fitness;

[globalBestFitness, globalBestIndex] = min(fitness);

gBestPosition = positions(globalBestIndex, :);

bestFitnessOverTime = zeros(maxIterations, 1);

for iteration = 1:maxIterations

r1 = chaoticMapFunction(nDimensions);

r2 = chaoticMapFunction(nDimensions);

velocities = inertiaWeight \* velocities + ...

(cognitiveCoeff \* r1) .\* (pBestPositions - positions) + ...

(socialCoeff \* r2) .\* (gBestPosition - positions);

positions = positions + velocities;

positions = max(min(positions, UB), LB);

fitness = arrayfun(@(i) fitnessFunction(positions(i,:)), 1:nParticles, 'UniformOutput', false);

fitness = cell2mat(fitness);

improved = fitness < pBestFitness;

pBestPositions(improved, :) = positions(improved, :);

pBestFitness(improved) = fitness(improved);

[currentGlobalBestFitness, globalBestIndex] = min(fitness);

if currentGlobalBestFitness < globalBestFitness

globalBestFitness = currentGlobalBestFitness;

gBestPosition = positions(globalBestIndex, :);

end

bestFitnessOverTime(iteration) = globalBestFitness;

end

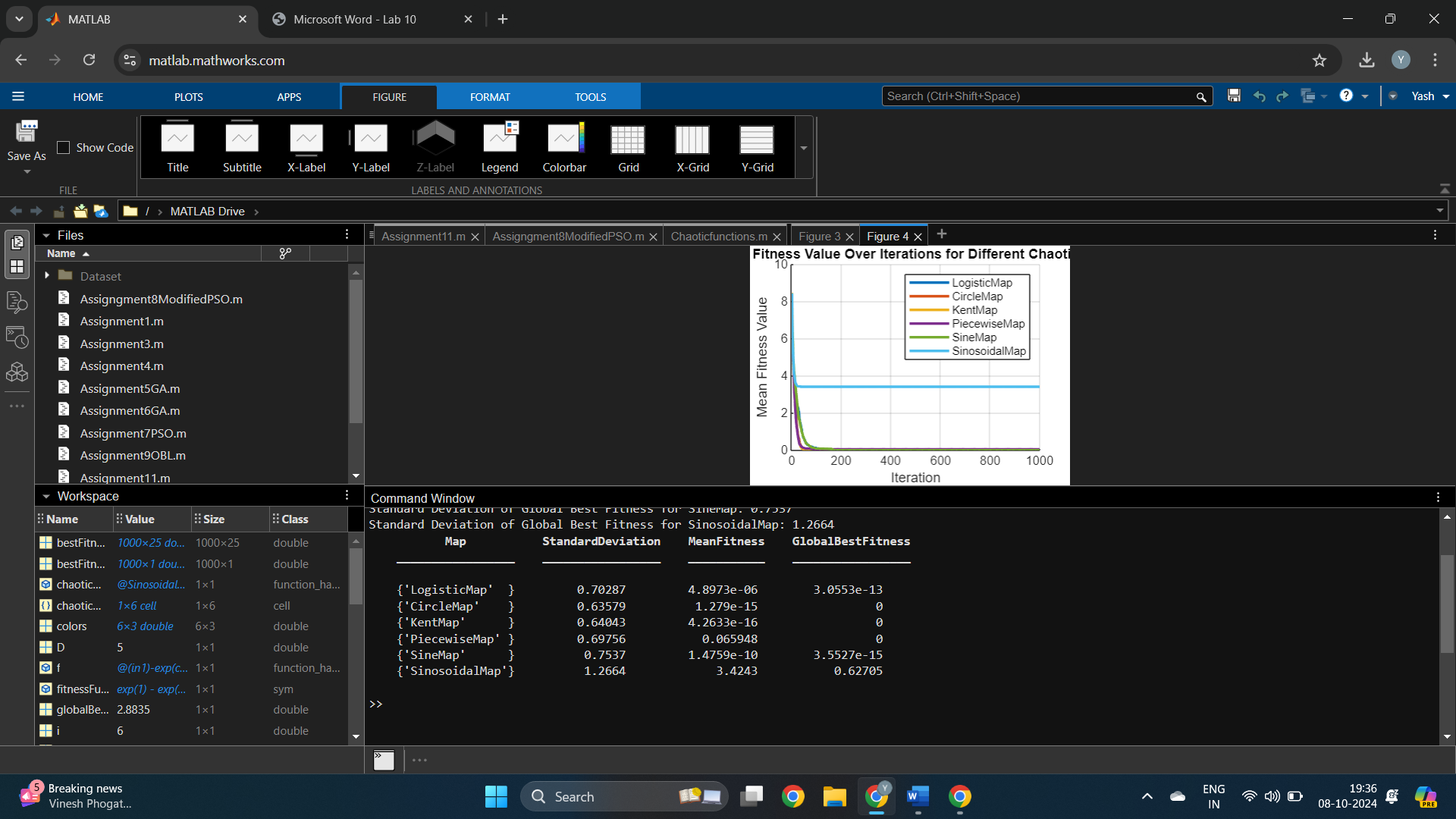
end

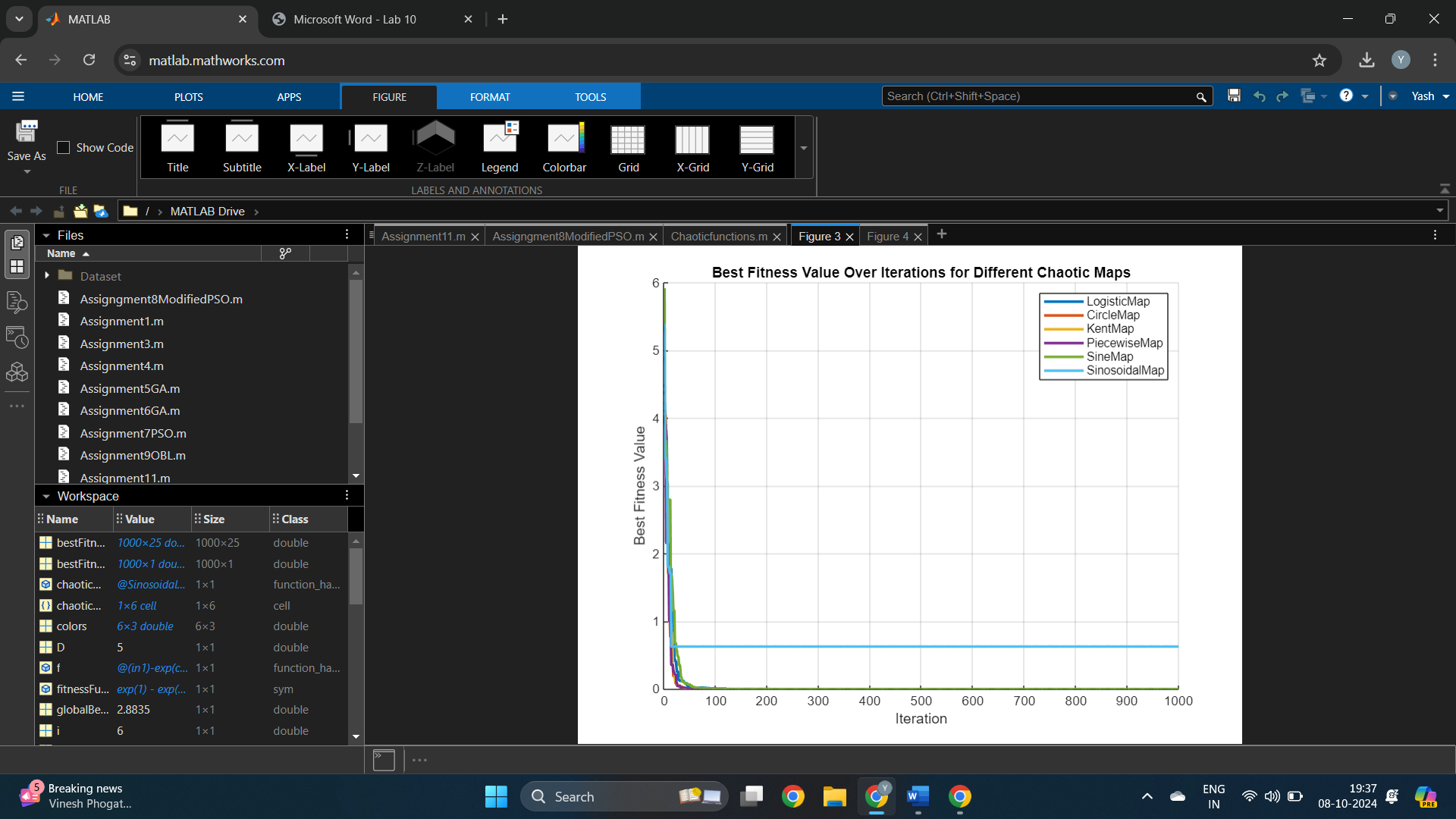
**Output:**

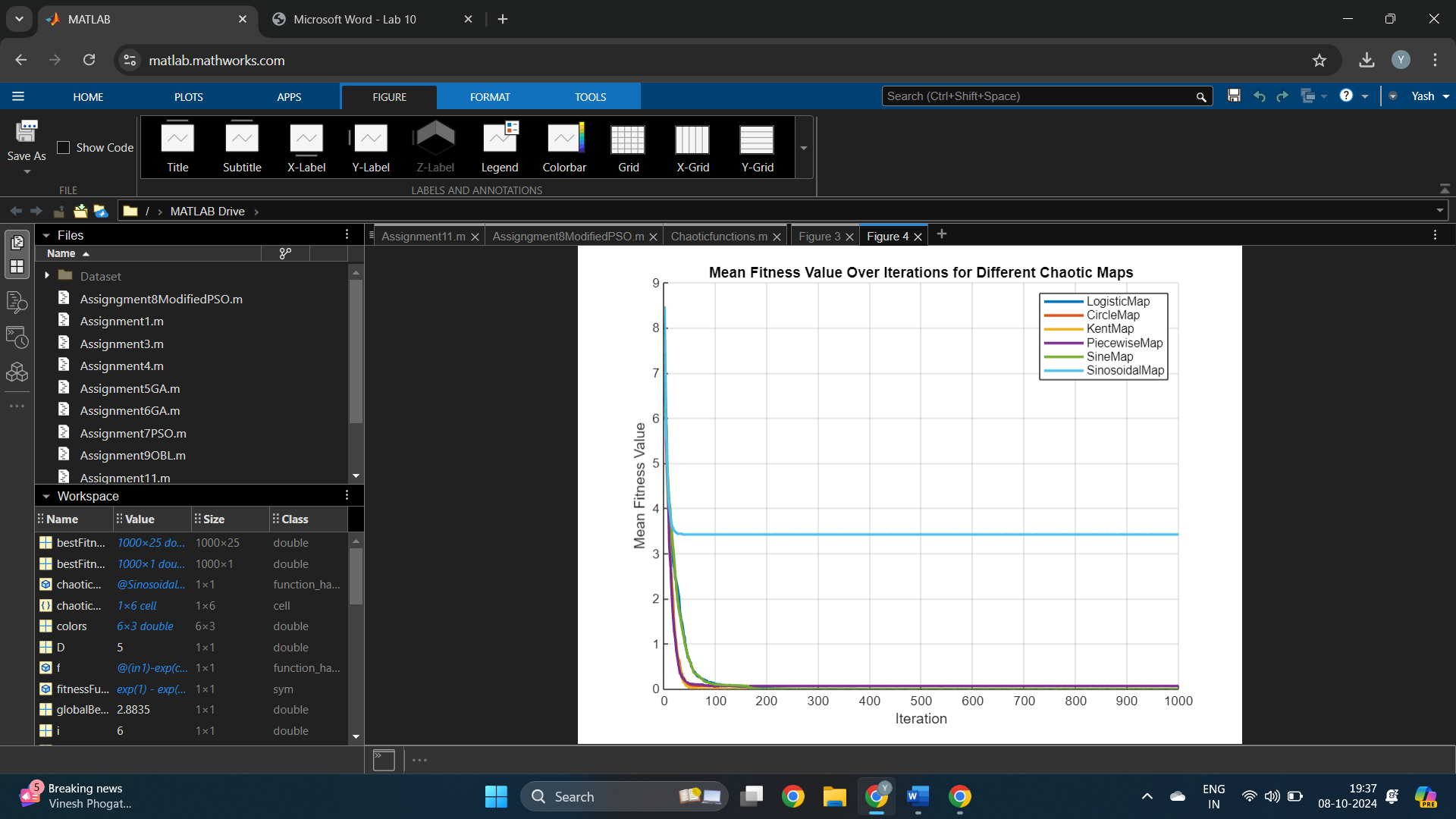
1. Ackley

Dimensions=5

F='-20\*exp(-0.2\*sqrt((x1^2 + x2^2 + x3^2 + x4^2 + x5^2)/5)) - exp((cos(2\*pi\*x1) + cos(2\*pi\*x2) + cos(2\*pi\*x3) + cos(2\*pi\*x4) + cos(2\*pi\*x5))/5) + 20 + exp(1)'



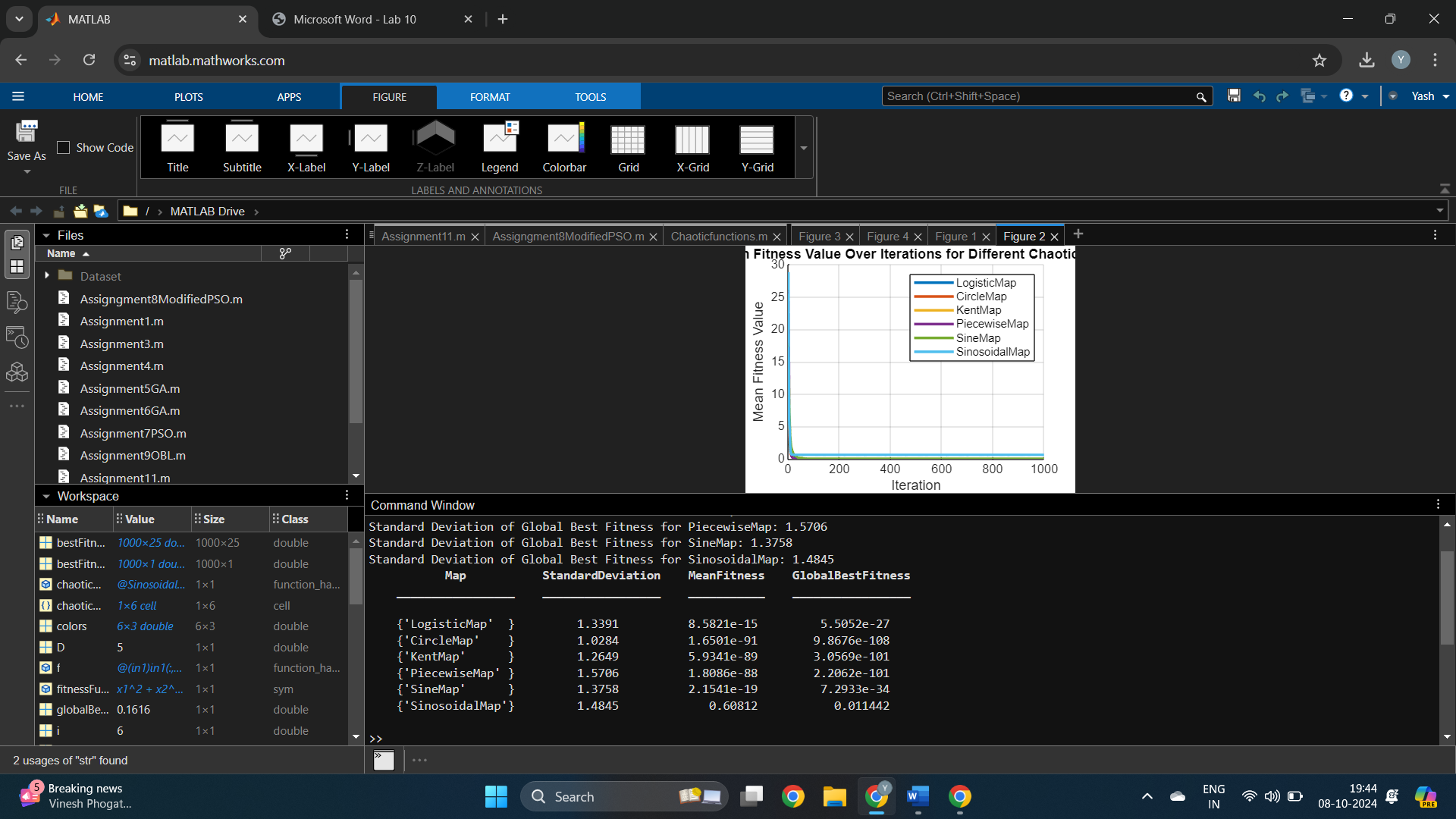


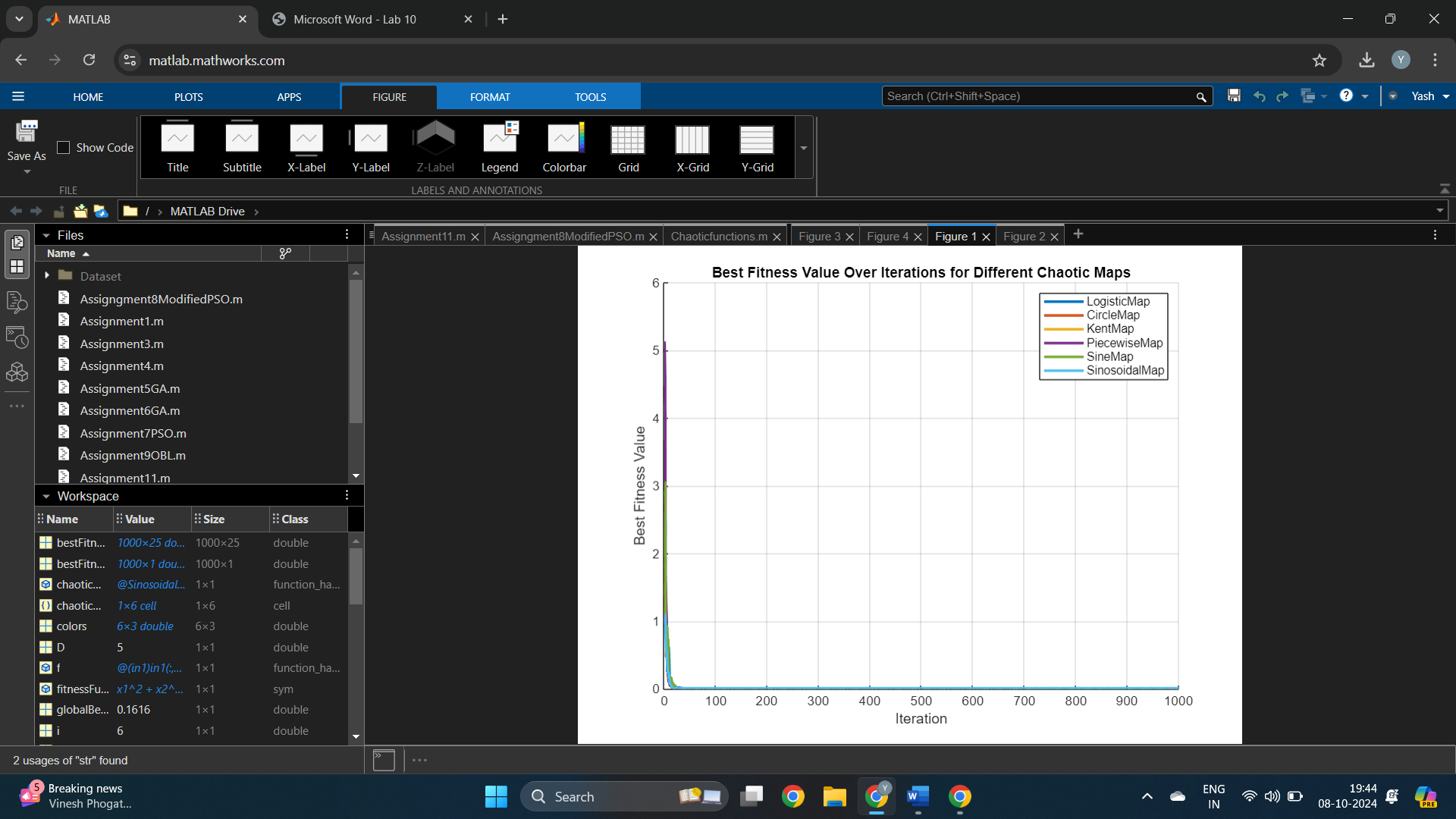


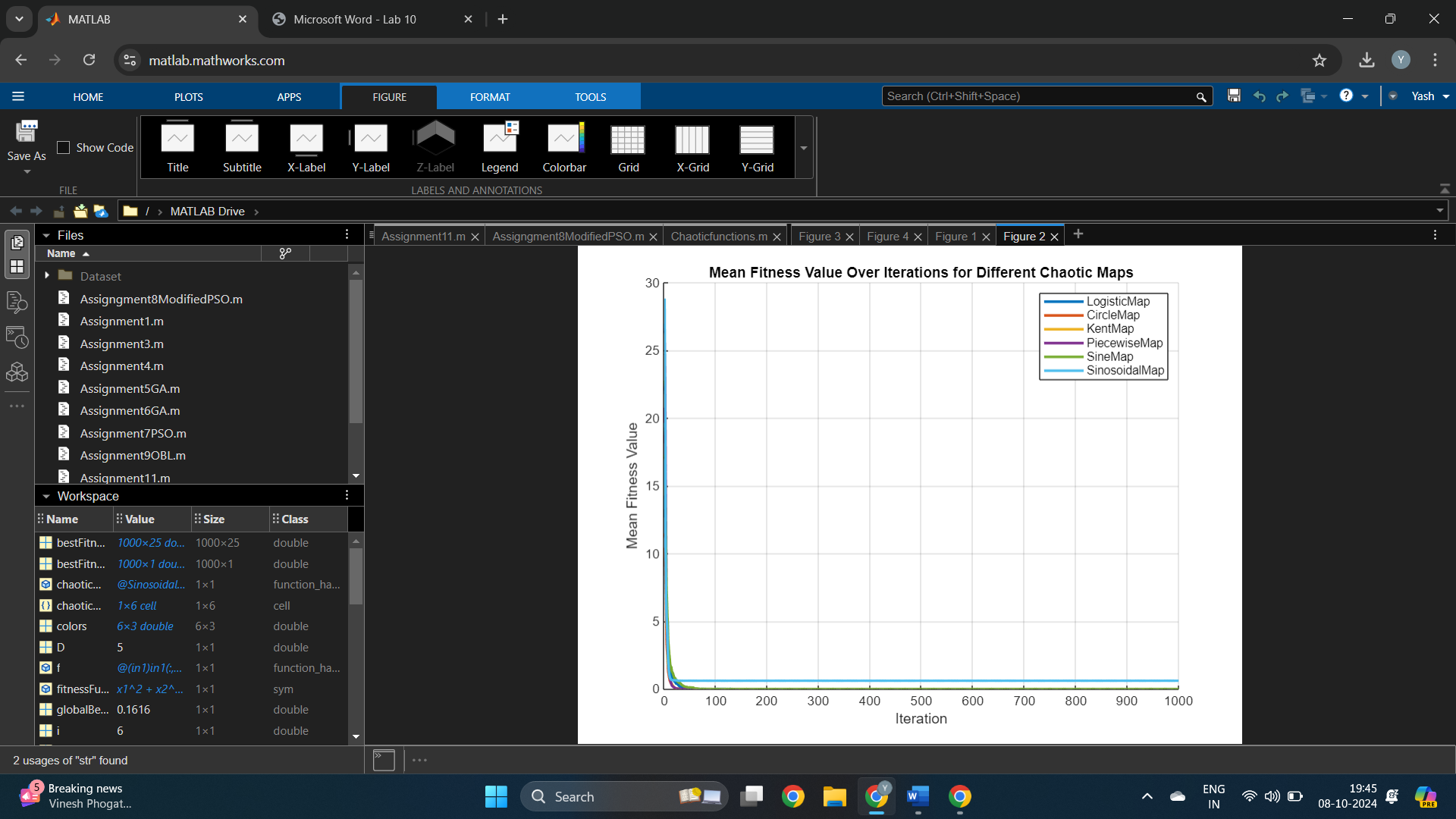
1. Sphere Function

Dimensions=5

F='x1^2 + x2^2 + x3^2 + x4^2 + x5^2'



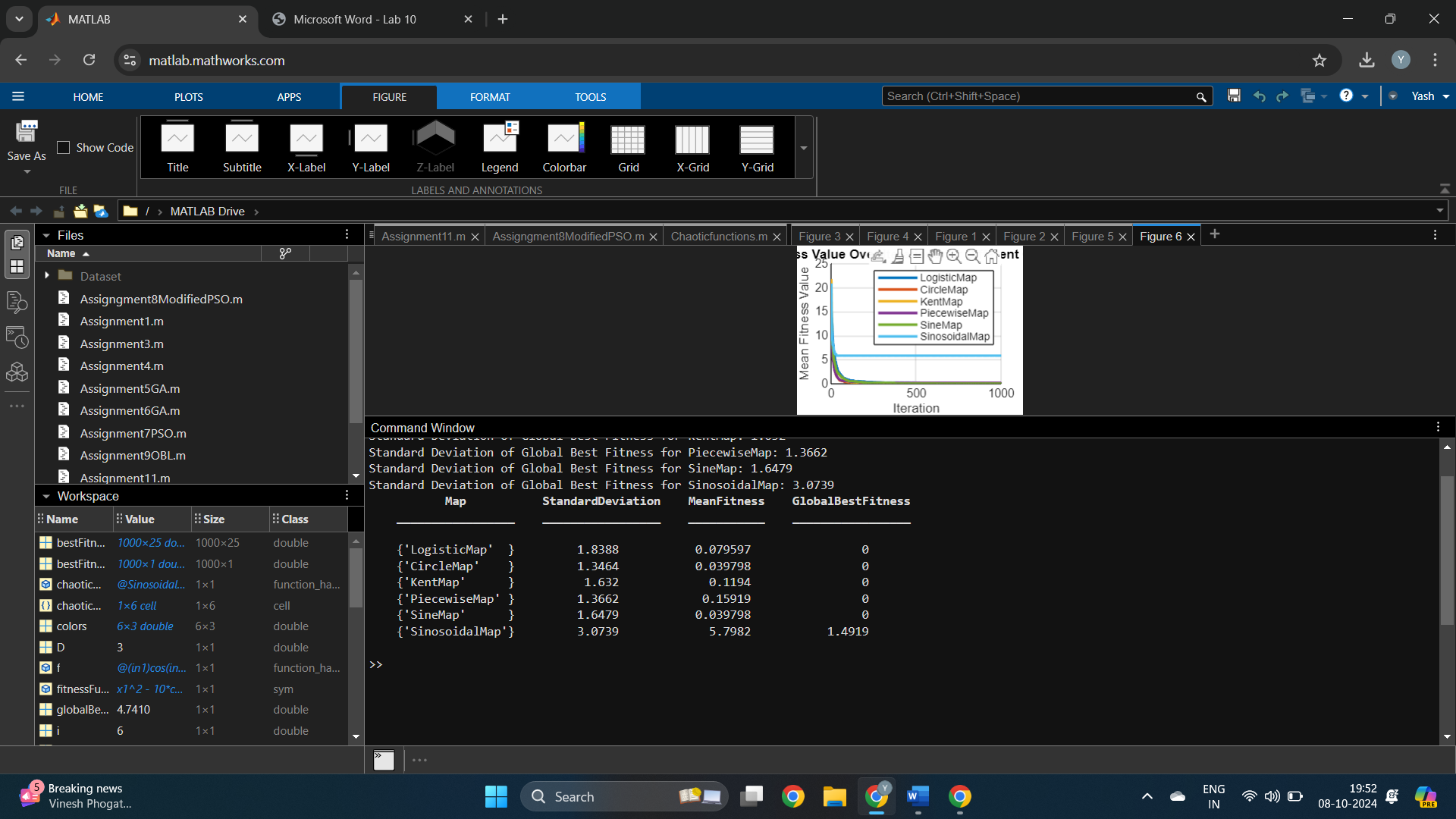


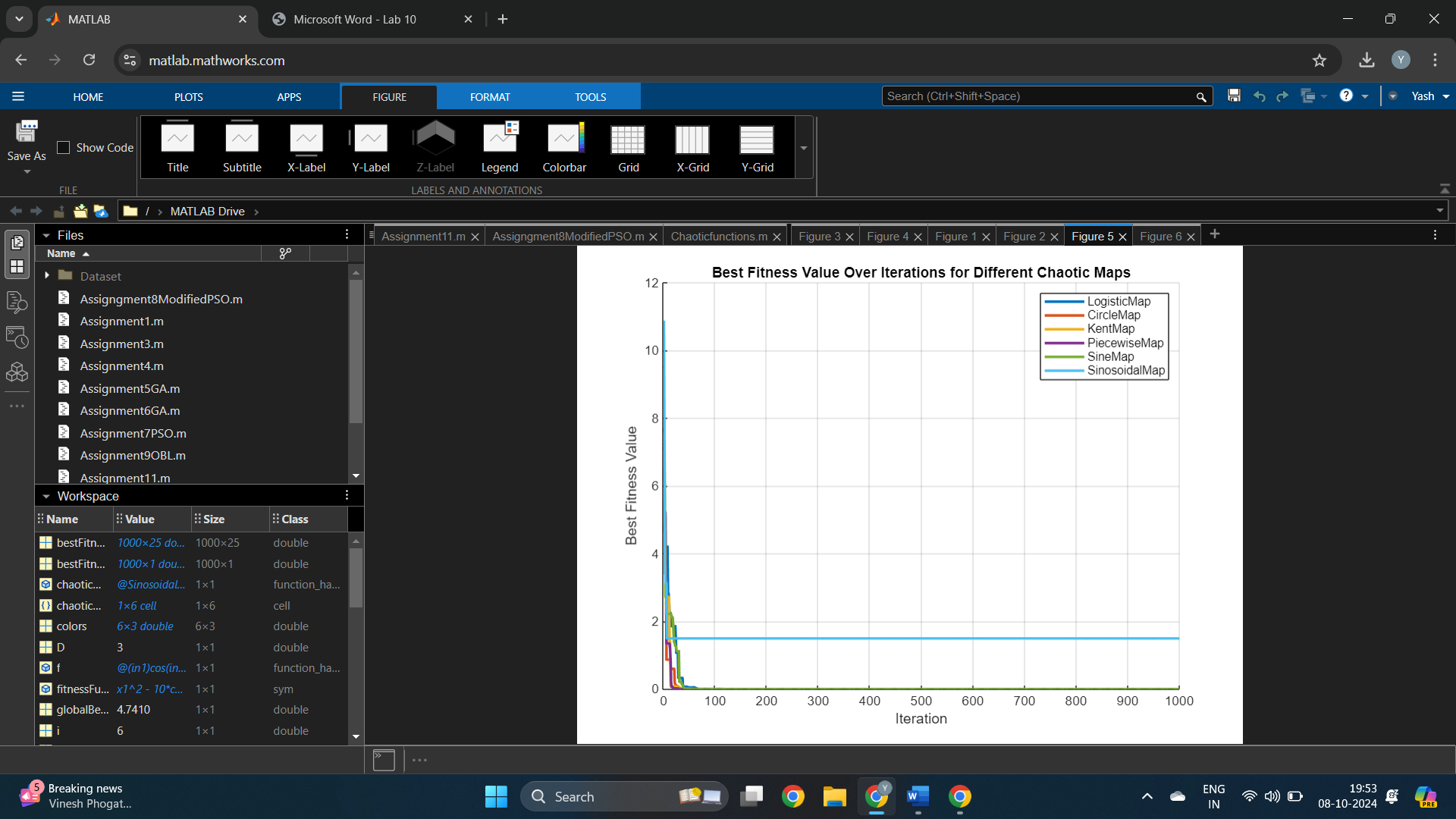


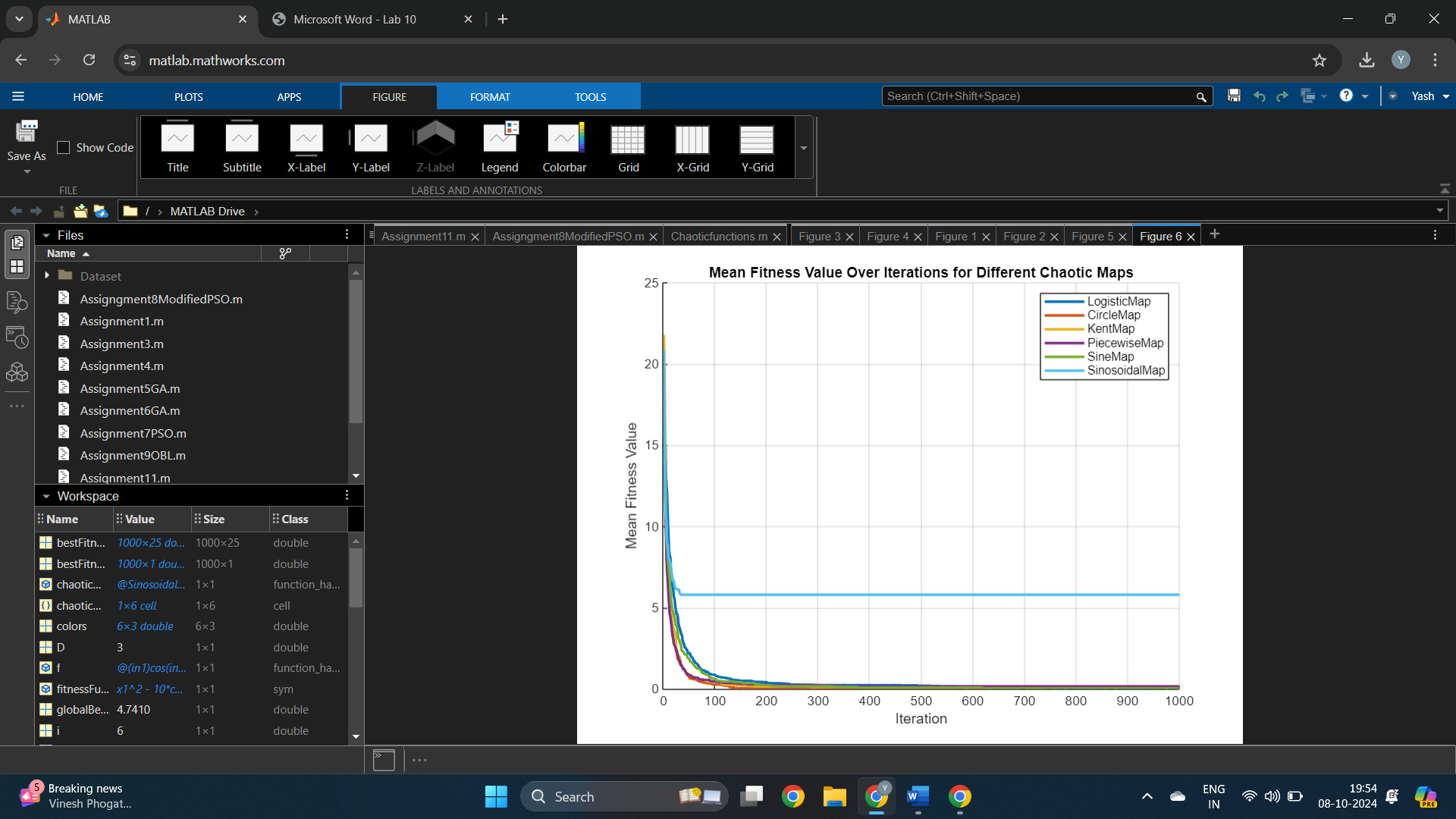
1. Rastrigin Function:

Dimensions=3

F='10\*3 + (x1^2 - 10\*cos(2\*pi\*x1)) + (x2^2 - 10\*cos(2\*pi\*x2)) + (x3^2 - 10\*cos(2\*pi\*x3))'



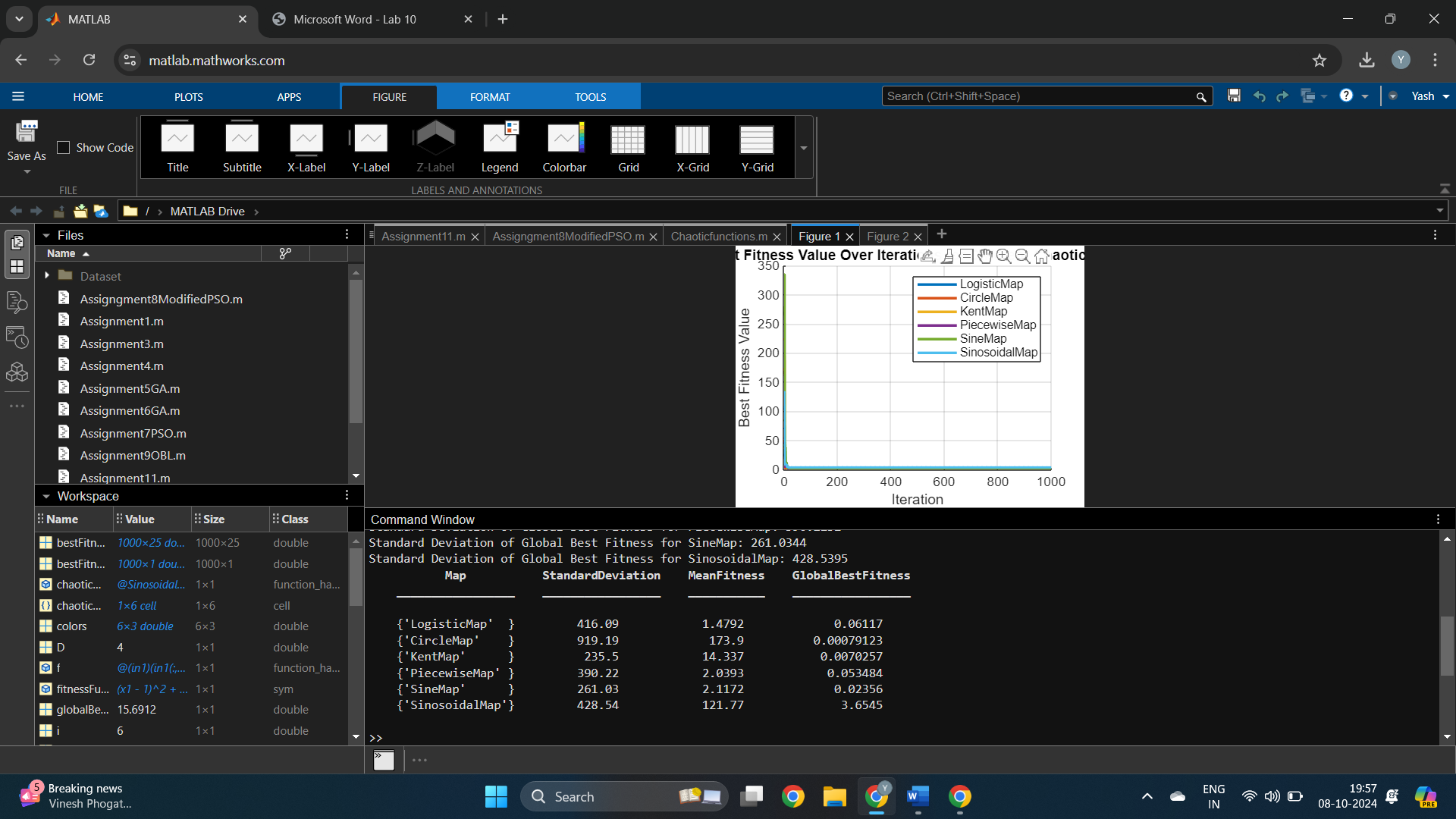


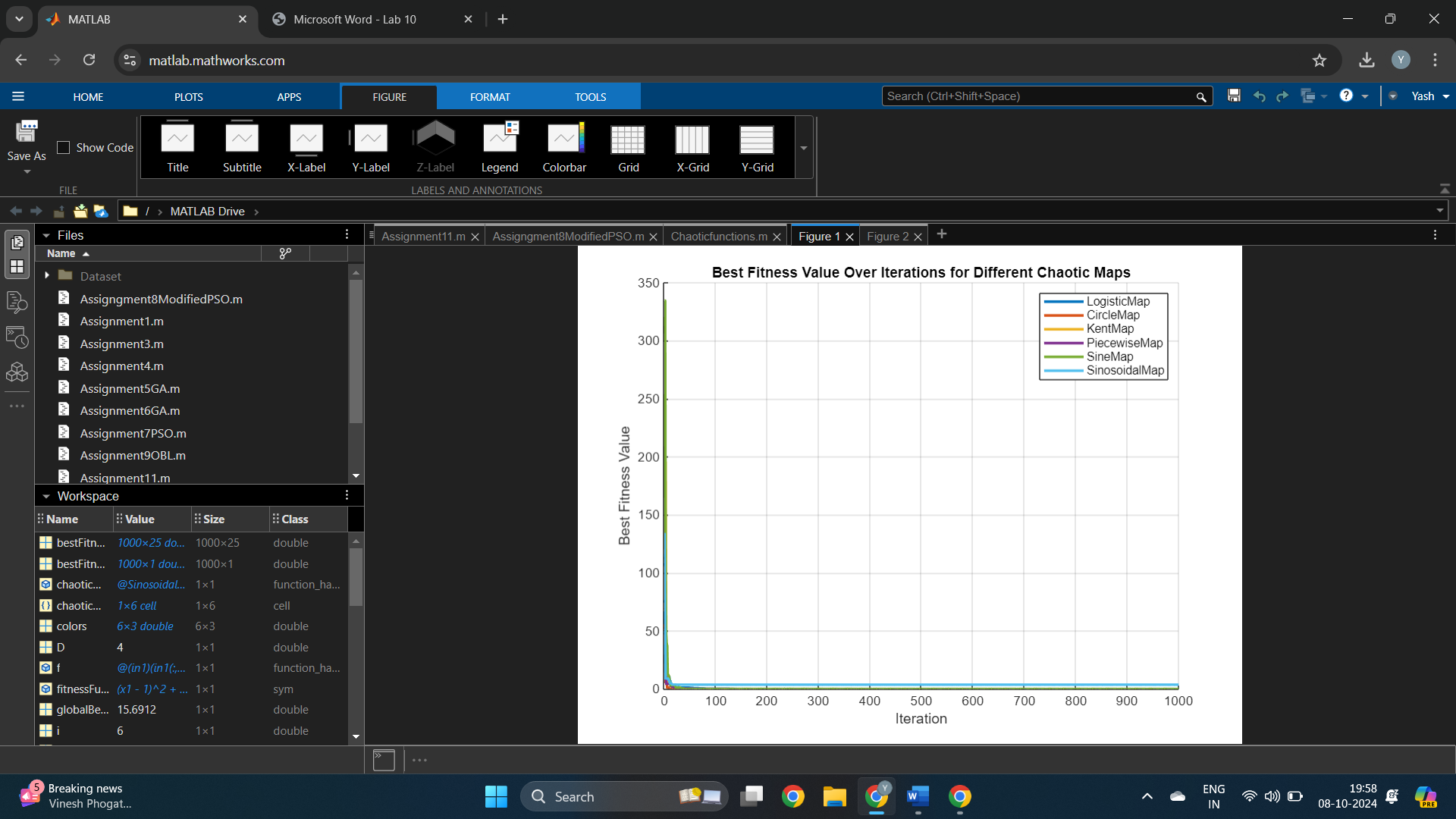


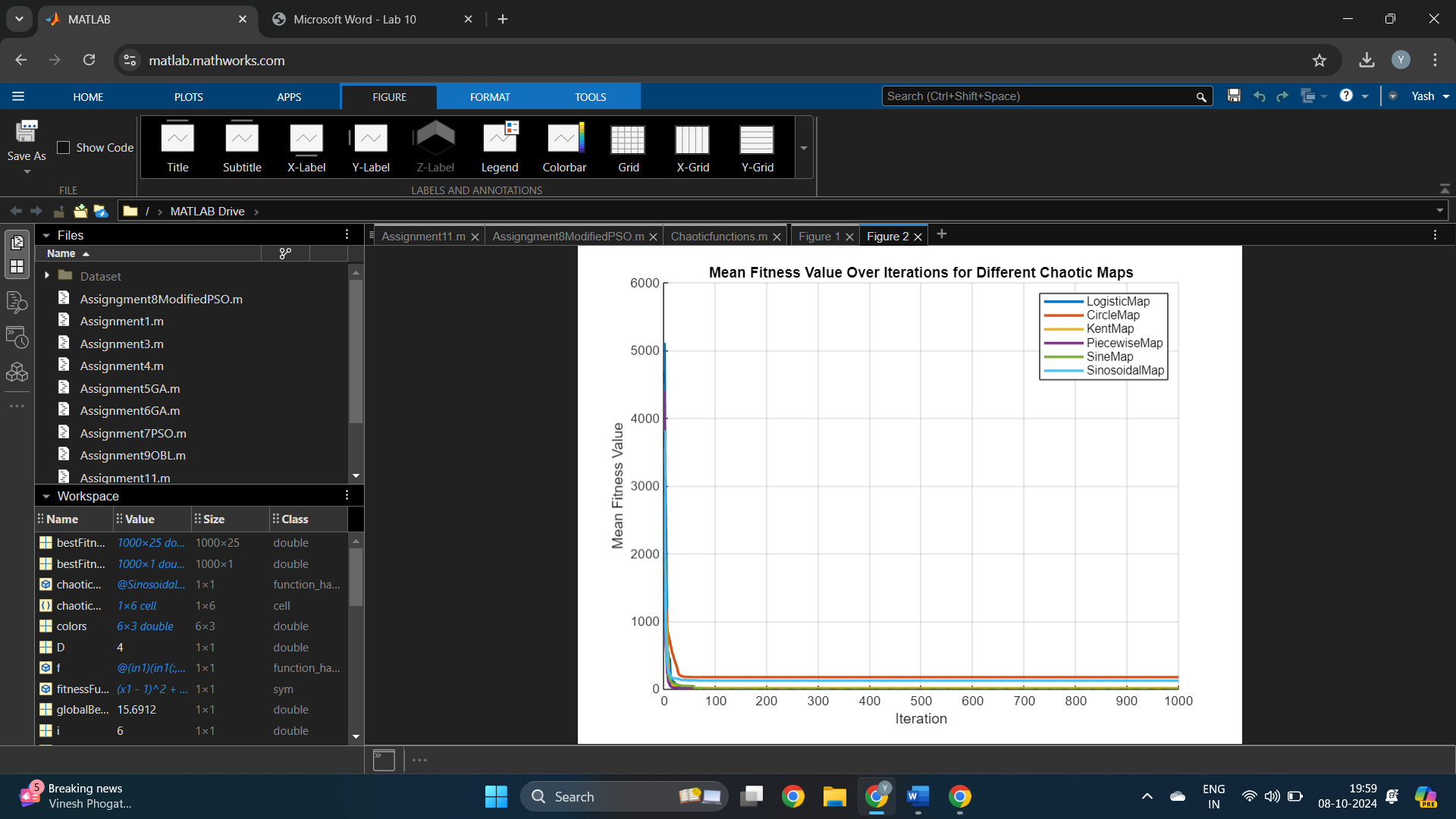
1. Rosenbrock Function

Dimension=4

F='100\*(x2 - x1^2)^2 + (x1 - 1)^2 + 100\*(x3 - x2^2)^2 + (x2 - 1)^2 + 100\*(x4 - x3^2)^2 + (x3 - 1)^2'







1. Griewank Function

Dimensions=6

F='1 + (x1^2 + x2^2 + x3^2 + x4^2 + x5^2 + x6^2)/4000 - cos(x1/sqrt(1))\*cos(x2/sqrt(2))\*cos(x3/sqrt(3))\*cos(x4/sqrt(4))\*cos(x5/sqrt(5))\*cos(x6/sqrt(6))'

