**SOURCE PROGRAMS FOR THE THREE DIFFERENT ERAS PLOT**

1. **FOR Noachian Era**

% Parameters

gridSize = 200; % Size of the grid (higher = finer resolution)

craterDensity = 0.02; % Density of craters (higher = more craters)

maxCraterDepth = -2; % Maximum depth of craters (negative for depressions)

volcanicHeight = 5; % Height of volcanic features

smoothness = 2; % Smoothness of the terrain (higher = smoother)

% Generate base terrain (random noise)

terrain = randn(gridSize); % Random noise

terrain = imgaussfilt(terrain, smoothness); % Smooth the terrain

% Add craters (circular depressions)

numCraters = round(craterDensity \* gridSize^2);

for i = 1:numCraters

% Random crater center and radius

centerX = randi([1, gridSize]);

centerY = randi([1, gridSize]);

radius = randi([5, 20]); % Random radius

% Create a crater mask

[X, Y] = meshgrid(1:gridSize, 1:gridSize);

craterMask = ((X - centerX).^2 + (Y - centerY).^2) <= radius^2;

% Add the crater to the terrain

terrain(craterMask) = terrain(craterMask) + maxCraterDepth \* (1 - ((X(craterMask) - centerX).^2 + (Y(craterMask) - centerY).^2) / radius^2);

end

% Add volcanic features (random peaks)

numVolcanoes = 5; % Number of volcanic peaks

for i = 1:numVolcanoes

% Random volcano center

centerX = randi([1, gridSize]);

centerY = randi([1, gridSize]);

% Create a volcano mask

[X, Y] = meshgrid(1:gridSize, 1:gridSize);

volcanoMask = ((X - centerX).^2 + (Y - centerY).^2) <= 100; % Broad base

% Add the volcano to the terrain

terrain(volcanoMask) = terrain(volcanoMask) + volcanicHeight \* exp(-((X(volcanoMask) - centerX).^2 + (Y(volcanoMask) - centerY).^2) / 50);

end

% Plot the terrain

figure;

mesh(terrain);

colormap('turbo'); % Use a colormap to represent elevation

title('Noachian Era Terrain on Ancient Mars');

xlabel('X (km)');

ylabel('Y (km)');

zlabel('Elevation (km)');

colorbar; % Add a colorbar to show elevation

axis tight;

view(3); % 3D view

grid on;

% Observations

disp('Observations from the graph:');

disp('1. The terrain is heavily cratered, reflecting the intense bombardment during the Noachian era.');

disp('2. Volcanic features (peaks) are visible, indicating volcanic activity.');

disp('3. The surface is uneven, with both depressions (craters) and elevated regions (volcanoes).');

disp('4. The presence of liquid water could be inferred from the smoother regions between craters.');

**2**) **FOR HESPERIAN ERA**

% Parameters for Hesperian Era (Water Loss)

gridSize = 200; % Size of the grid (higher = finer resolution)

craterDensity = 0.01; % Reduced crater density compared to Noachian

maxCraterDepth = -1.5; % Less deep craters due to erosion and filling

volcanicHeight = 7; % Higher volcanic peaks due to extensive volcanic activity

smoothness = 3; % Smoother terrain due to lava flows

% Generate base terrain (random noise)

terrain = randn(gridSize);

terrain = imgaussfilt(terrain, smoothness);

% Add craters (circular depressions)

numCraters = round(craterDensity \* gridSize^2);

for i = 1:numCraters

centerX = randi([1, gridSize]);

centerY = randi([1, gridSize]);

radius = randi([5, 20]);

[X, Y] = meshgrid(1:gridSize, 1:gridSize);

craterMask = ((X - centerX).^2 + (Y - centerY).^2) <= radius^2;

terrain(craterMask) = terrain(craterMask) + maxCraterDepth \* (1 - ((X(craterMask) - centerX).^2 + (Y(craterMask) - centerY).^2) / radius^2);

end

% Add volcanic features (random peaks)

numVolcanoes = 10;

for i = 1:numVolcanoes

centerX = randi([1, gridSize]);

centerY = randi([1, gridSize]);

[X, Y] = meshgrid(1:gridSize, 1:gridSize);

volcanoMask = ((X - centerX).^2 + (Y - centerY).^2) <= 120;

terrain(volcanoMask) = terrain(volcanoMask) + volcanicHeight \* exp(-((X(volcanoMask) - centerX).^2 + (Y(volcanoMask) - centerY).^2) / 60);

end

% Add drying water channels

numChannels = 5;

for i = 1:numChannels

channelLength = randi([30, 80]);

xStart = randi([1, gridSize - channelLength]);

yStart = randi([1, gridSize - channelLength]);

for j = 1:channelLength

x = round(xStart + j \* randn());

y = round(yStart + j \* randn());

if x > 0 && x <= gridSize && y > 0 && y <= gridSize

terrain(y, x) = terrain(y, x) - 1.2; % Create a shallow depression

end

end

end

% Plot the terrain

figure;

mesh(terrain);

colormap('copper'); % Colormap representing dry, dusty, and volcanic terrain

title('Hesperian Era Terrain on Mars ');

xlabel('X (km)');

ylabel('Y (km)');

zlabel('Elevation (km)');

colorbar;

axis tight;

view(3);

grid on;

% Observations

disp('Observations from the graph:');

disp('1. Terrain shows reduced cratering but more volcanic activity with prominent peaks.');

disp('2. Presence of dried water channels indicating past flowing water now mostly gone.');

disp('3. Lava plains are evident, creating smoother, flatter regions.');

disp('4. The landscape depicts a transition from a wetter environment to a colder, drier Mars.');

3) FOR AMAOZOIN ERA

% Parameters for Amazonian Era (Dry and Arid Conditions)

gridSize = 200; % Size of the grid (higher = finer resolution)

craterDensity = 0.005; % Very low crater density due to resurfacing

maxCraterDepth = -1; % Shallow craters due to erosion and filling

volcanicHeight = 4; % Lower volcanic peaks due to reduced activity

smoothness = 4; % Smoother terrain due to erosion, dust, and ice deposits

% Generate base terrain (random noise)

terrain = randn(gridSize);

terrain = imgaussfilt(terrain, smoothness);

% Add craters (circular depressions)

numCraters = round(craterDensity \* gridSize^2);

[X, Y] = meshgrid(1:gridSize, 1:gridSize); % Precompute meshgrid for efficiency

for i = 1:numCraters

centerX = randi([1, gridSize]);

centerY = randi([1, gridSize]);

radius = randi([5, 15]);

craterMask = ((X - centerX).^2 + (Y - centerY).^2) <= radius^2;

terrain(craterMask) = terrain(craterMask) + maxCraterDepth \* (1 - ((X(craterMask) - centerX).^2 + (Y(craterMask) - centerY).^2) / radius^2);

end

% Add volcanic features (random peaks)

numVolcanoes = 5;

for i = 1:numVolcanoes

centerX = randi([1, gridSize]);

centerY = randi([1, gridSize]);

volcanoMask = ((X - centerX).^2 + (Y - centerY).^2) <= 80;

terrain(volcanoMask) = terrain(volcanoMask) + volcanicHeight \* exp(-((X(volcanoMask) - centerX).^2 + (Y(volcanoMask) - centerY).^2) / 50);

end

% Add aeolian erosion and sand dunes

numDunes = 8;

for i = 1:numDunes

xStart = randi([1, gridSize - 20]);

yStart = randi([1, gridSize - 20]);

duneWidth = randi([5, 10]);

duneHeight = 0.5; % Small, subtle sand dune height

for j = 1:20

x = xStart + j;

y = yStart + round(3 \* sin(j / 2));

if x > 0 && x <= gridSize && y > 0 && y <= gridSize

terrain(y, max(1, x-duneWidth):min(gridSize, x+duneWidth)) = terrain(y, max(1, x-duneWidth):min(gridSize, x+duneWidth)) + duneHeight;

end

end

end

% Add glaciation evidence (smooth, scoured regions)

numGlacialRegions = 3;

for i = 1:numGlacialRegions

centerX = randi([1, gridSize]);

centerY = randi([1, gridSize]);

radius = randi([15, 30]);

glacialMask = ((X - centerX).^2 + (Y - centerY).^2) <= radius^2;

terrain(glacialMask) = terrain(glacialMask) - 0.5 \* exp(-((X(glacialMask) - centerX).^2 + (Y(glacialMask) - centerY).^2) / (2 \* radius));

end

% Plot the terrain

figure;

mesh(terrain);

colormap('autumn'); % Reddish, dusty appearance due to ferric oxide alteration

title('Amazonian Era Terrain on Mars (Dry and Arid Conditions)');

xlabel('X (km)');

ylabel('Y (km)');

zlabel('Elevation (km)');

colorbar;

axis tight;

view(3);

grid on;

% Observations

disp('Observations from the graph:');

disp('1. Terrain shows very few craters due to extensive resurfacing.');

disp('2. Volcanic activity is present but reduced, with lower peaks.');

disp('3. Aeolian erosion and sand dunes are evident.');

disp('4. Signs of glaciation and scoured regions are present.');

disp('5. The reddish appearance reflects the ferric oxide alteration of the surface.');

4) **Program to Calculate and Plot Maximum Potential Survival Ratio on Mars**

% MATLAB Program to Calculate and Plot Maximum Potential Survival Ratio on Mars

% Define survival factors and their respective weights

factors = {'Water', 'Oxygen', 'Food', 'Energy', 'Radiation Protection', ...

'Temperature Regulation', 'Building Materials', 'Atmosphere Modification', ...

'Waste Management', 'Health Support'};

weights = [0.2, 0.15, 0.15, 0.1, 0.1, 0.1, 0.05, 0.05, 0.05, 0.05]; % Relative importance

% Define current survival scores for each factor (values between 0 and 1)

scores\_current = [0.6, 0.4, 0.5, 0.7, 0.3, 0.4, 0.2, 0.1, 0.6, 0.5];

% Define maximum possible survival scores assuming full problem resolution

scores\_max = ones(size(scores\_current)); % All factors set to 1 (fully sustainable)

% Compute the weighted survival ratio for current and maximum scenarios

survival\_ratio\_current = sum(weights .\* scores\_current);

survival\_ratio\_max = sum(weights .\* scores\_max);

% Display results

fprintf('Current Survival Ratio on Mars: %.2f\n', survival\_ratio\_current);

fprintf('Maximum Potential Survival Ratio on Mars: %.2f\n', survival\_ratio\_max);

% Interpretation

if survival\_ratio\_max >= 0.8

disp('Mars could become highly habitable with full problem resolution.');

survival\_ratio\_max >= 0.5

disp('Mars could support life with significant improvements.');

else

disp('Even with all improvements, survival on Mars remains highly challenging.');

end

% Plot the survival factors and their scores

figure;

bar([scores\_current; scores\_max]', 'grouped');

set(gca, 'XTickLabel', factors, 'XTickLabelRotation', 45);

ylabel('Survival Score');

title('Current vs Maximum Survival Factor Scores on Mars');

legend('Current', 'Max Potential');

grid on;

% Plot the weighted contribution of each factor

figure;

bar([weights .\* scores\_current; weights .\* scores\_max]', 'grouped');

set(gca, 'XTickLabel', factors, 'XTickLabelRotation', 45);

ylabel('Weighted Contribution');

title('Current vs Maximum Weighted Contribution of Survival Factors');

legend('Current', 'Max Potential');

grid on;