```
%% ADAGRAD OPTIMIZATION
clear; clc;

% -------------------- PARAMETERS --------------------
eta = 0.1;           % Initial learning rate
eps = 1e-8;          % Small constant to avoid division by zero
iterations = 3;      % First 3 iterations only
tol = 1e-6;          % Termination condition

% -------------------- INITIAL POINT --------------------
x = 1; y = 1;        % Starting point

% Accumulators for squared gradients
Gx = 0;
Gy = 0;

% -------------------- STORAGE --------------------
X = zeros(iterations+1,1);
Y = zeros(iterations+1,1);

X(1) = x;
Y(1) = y;

fprintf('ADAGRAD\n');
```

```
ADAGRAD
```

```
fprintf('Iter\t x\t\t y\n');
```

```
Iter     x        y
```

```
fprintf('0\t %.4f\t %.4f\n', x, y);
```

```
0     1.0000     1.0000
```

```

% -------------------- ITERATIONS --------------------
for k = 1:iterations

    % Gradient calculation
    dfx = 2*x + y + cos(x+y);
    dfy = x + 4*y + cos(x+y);

    % Accumulate squared gradients
    Gx = Gx + dfx^2;
    Gy = Gy + dfy^2;

    % Adaptive learning rate update
    x = x - (eta / sqrt(Gx + eps)) * dfx;
    y = y - (eta / sqrt(Gy + eps)) * dfy;

    % Store updated values
```

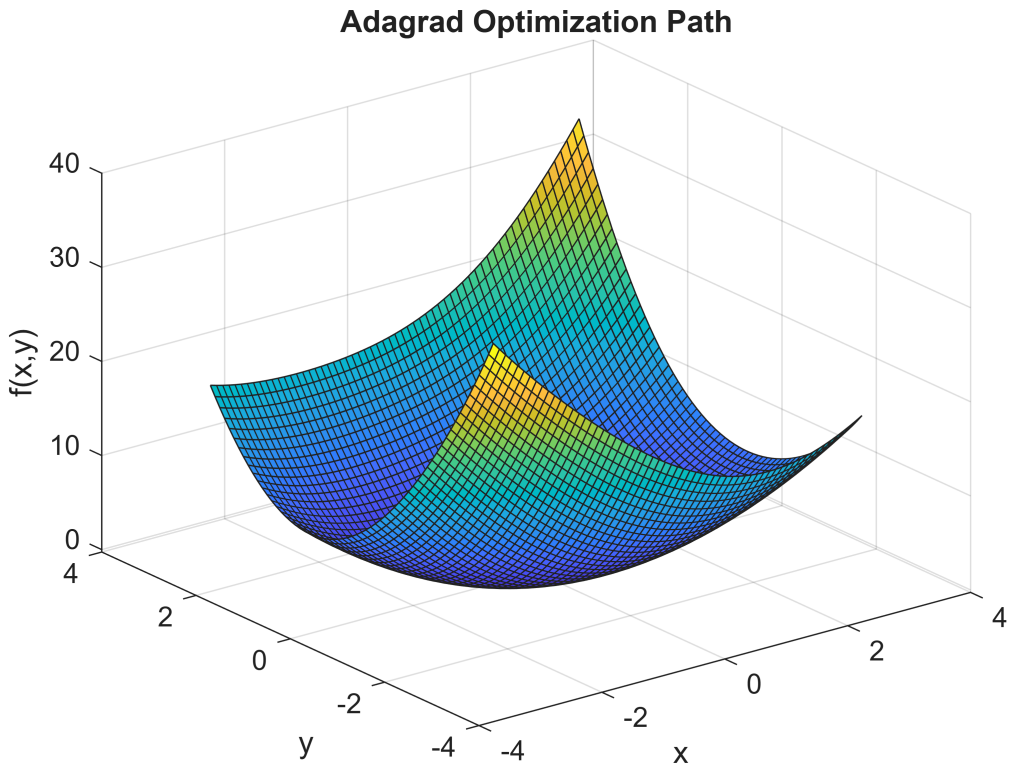
```
    X(k+1) = x;
    Y(k+1) = y;

    fprintf('%d\t %.4f\t %.4f\n', k, x, y);
end
```

```
1     0.9000     0.9000
2     0.8309     0.8318
3     0.7751     0.7774
```

```
% -------------------- VISUALIZATION --------------------
[xs, ys] = meshgrid(-3:0.1:3, -3:0.1:3);
fs = xs.^2 + xs.*ys + 2*ys.^2 + sin(xs + ys);

figure;
surf(xs, ys, fs); hold on;
plot3(X, Y, X.^2 + X.*Y + 2*Y.^2 + sin(X+Y), 'g-o', 'LineWidth', 2);
xlabel('x'); ylabel('y'); zlabel('f(x,y)');
title('Adagrad Optimization Path');
grid on;
```




```
figure;
contour(xs, ys, fs, 30); hold on;
plot(X, Y, 'r-o', 'LineWidth', 2);
xlabel('x'); ylabel('y');
title('Contour Plot – Adagrad');
```

```
grid on;
```