```
function [x_vals, y_vals, X, delta_x, delta_y] = Radiator_fin(x_iter, y_iter, TOL,↙
max_iter)

    L = pi/8;
    H = pi/16;
    beta=10000;
    X = zeros(x_iter, y_iter, 1);
    x_vals = linspace(0,L,x_iter);
    y_vals = linspace(0,H,y_iter);

    delta_x = x_vals(2) - x_vals(1);
    delta_y = y_vals(2) - y_vals(1);

    for i = 2:max_iter

        X(:,1,i) = 0;
        X(1,:,i) = 0;
        X(:,end,i) = beta*x_vals.*(L-x_vals);
        X(end,:,i) = 0;
        X(2:x_iter-1,2:y_iter-1,i) = 0.25*(X(3:x_iter,2:y_iter-1,i-1) + X(1:x_iter-2,2:↙
y_iter-1,i-1) + X(2:x_iter-1,3:y_iter,i-1) + X(2:x_iter-1,1:y_iter-2,i-1));
        if max(abs(X(:,:,i) - X(:,:,i-1)),[],"all") <= TOL
            break
        end
    end

    str = "";
    decide = input(str + "Press enter to exit. Type 1 for Finite Difference Solution",↙
"s");
    switch decide
        case "1"

            % Create meshgrid for surface plot
            [X_grid, Y_grid] = meshgrid(x_vals, y_vals(1:2:end));  % Assuming x and y are↙
1D vectors

            figure;
            hold on
            h = surf(X_grid, Y_grid, X(:,1:2:end,end)', "FaceAlpha", 0.25, 'FaceColor',↙
'b');
            xlabel('x');
            ylabel('y');
            zlabel('Temperature');
            title('2D Heat Equation Solution');
            % Fix z-axis limits to avoid recalculating each frame
            zlim([min(X(:)), max(X(:))]);
            view(3);
        case isempty(decide)
            return
    end
```

```
end
```