~/Downloads/lab11b.m

```
% Read the input image
image = imread('rgb.jpeg');  % Replace with your image file
gray_image = rgb2gray(image);      % Convert to grayscale if necessary

% Convert to double precision
image_double = double(gray_image);

% Step 1: Compute the 2D Fourier Transform of the image
F = fft2(image_double);

% Step 2: Shift the zero-frequency component to the center
F_shifted = fftshift(F);

% Step 3: Create the High-Pass Filter
[M, N] = size(F_shifted);  % Size of the image
[U, V] = meshgrid(-floor(N/2):floor(N/2)-1, -floor(M/2):floor(M/2)-1);  % Frequency coordinates
D0 = 5;  % Cutoff frequency, you can adjust this value
H = double(sqrt(U.^2 + V.^2) > D0);  % High-pass filter (1 for high frequencies, 0 for low frequencies)

if size(H) ~= size(F_shifted)
    H = imresize(H, size(F_shifted));
end

% Step 4: Apply the high-pass filter in the frequency domain
F_filtered = F_shifted .* H;

% Step 5: Inverse Fourier Transform to get the filtered image
F_inverse = ifftshift(F_filtered);  % Shift back the zero-frequency component
image_filtered = real(ifft2(F_inverse));  % Compute the inverse 2D Fourier transform and take the real part

% Step 6: Display the original and filtered images
figure;
subplot(1, 2, 1);
imshow(gray_image, []);
title('Original Image');

subplot(1, 2, 2);
imshow(image_filtered, []);
title('Filtered Image (High Pass)');

```