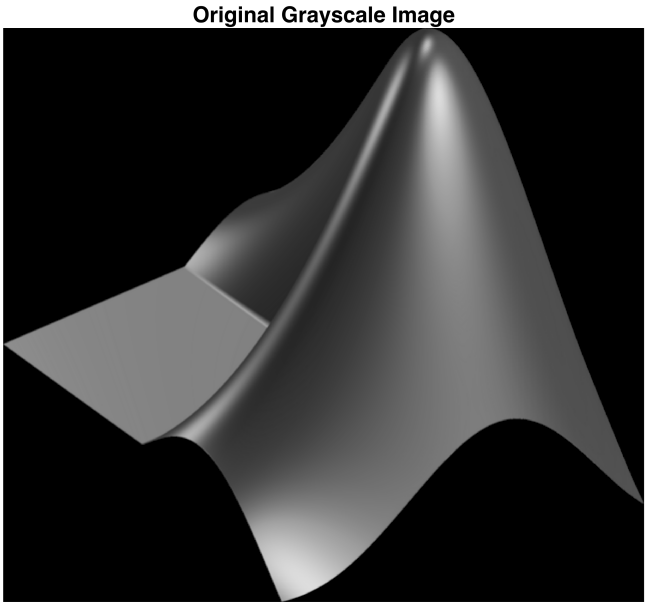
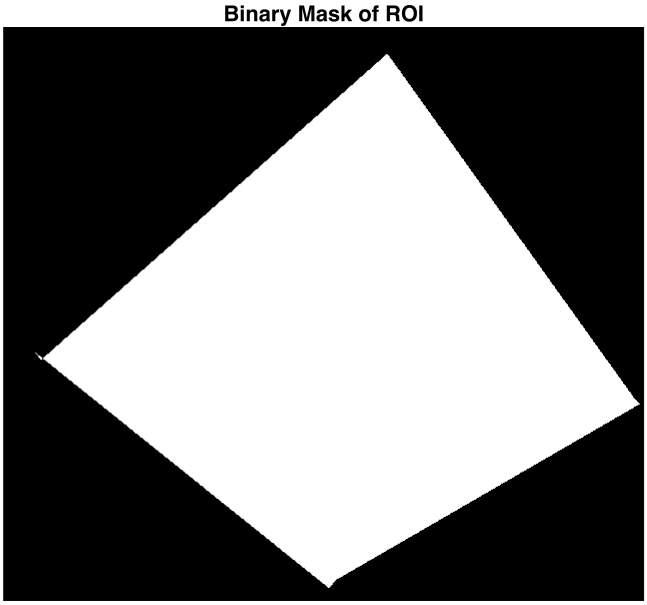
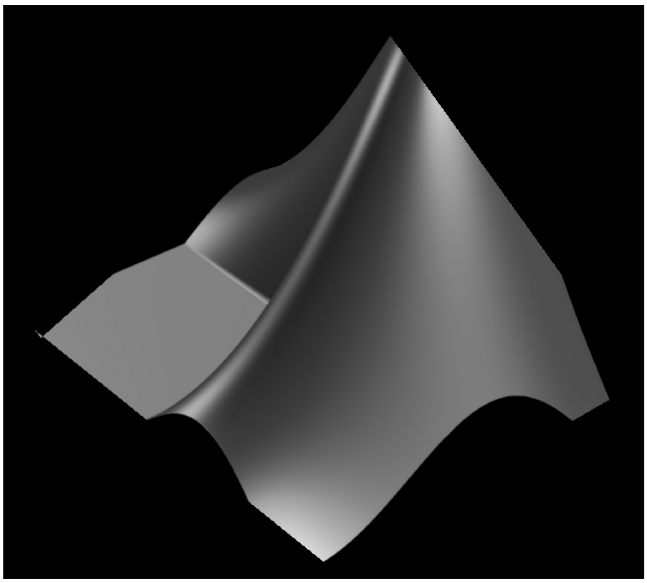
```
% Load and display the image
I = imread('image1.png');  % Replace with your actual image file

% Convert to grayscale if the image is in color
if size(I, 3) == 3
    I_gray = rgb2gray(I);
else
    I_gray = I;
end
figure, imshow(I_gray), title('Original Grayscale Image');
```

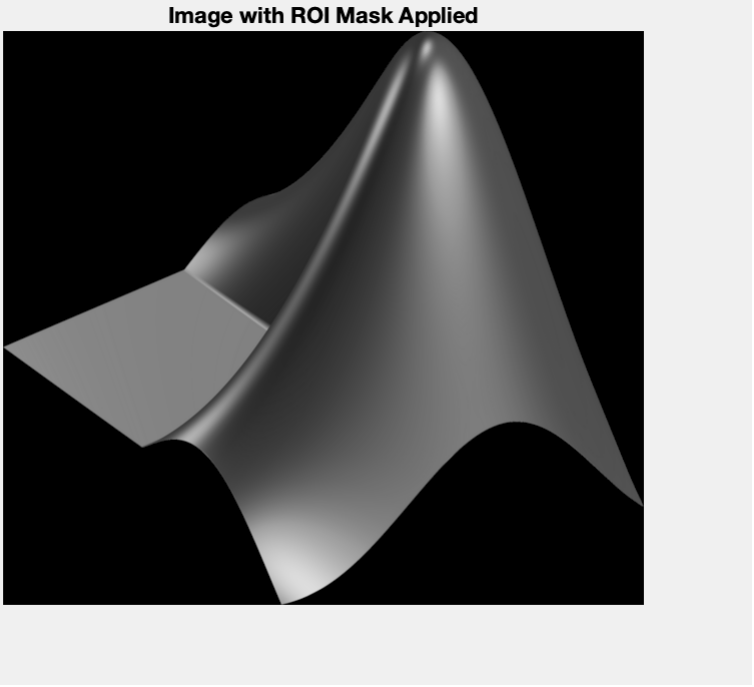



```
% Create a binary mask using roipoly
mask = roipoly(I_gray);  % Manually select the ROI
figure, imshow(mask), title('Binary Mask of ROI');
```

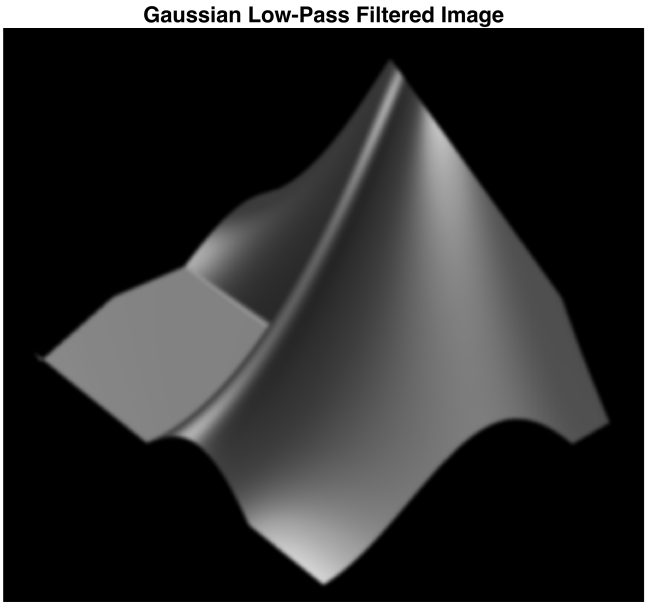



```
% Apply the binary mask to the original image
I_roi = I_gray .* uint8(mask);
figure, imshow(I_roi), title('Image with ROI Mask Applied');
```

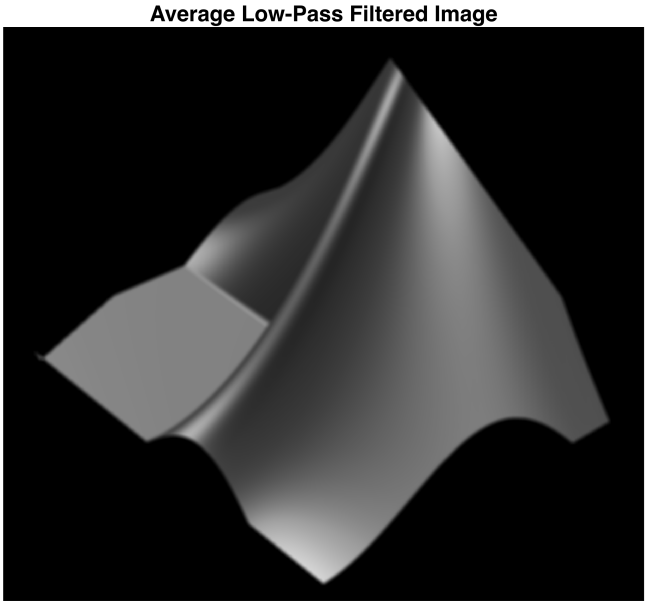




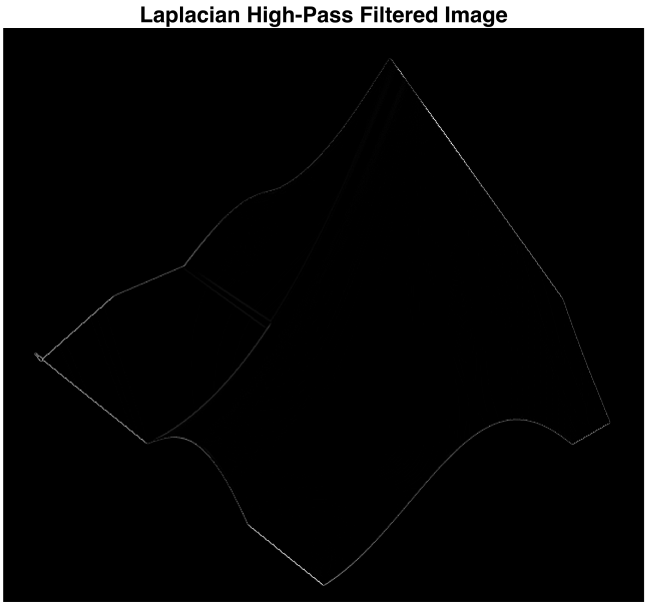

```
% Apply Gaussian Low-Pass Filter
I_gaussian = imgaussfilt(I_roi, 2);  % Apply Gaussian filter with sigma = 2
figure, imshow(I_gaussian), title('Gaussian Low-Pass Filtered Image');
```




```
%  Apply Average Low-Pass Filter
h_avg = fspecial('average', [5 5]);  % Create a 5x5 averaging filter
I_avg = imfilter(I_roi, h_avg);      % Apply the filter
figure, imshow(I_avg), title('Average Low-Pass Filtered Image');
```




```
%  Apply Laplacian High-Pass Filter
h_lap = fspecial('laplacian', 0.2);  % Create a Laplacian filter with alpha
= 0.2
I_laplacian = imfilter(I_roi, h_lap); % Apply the Laplacian filter
figure, imshow(I_laplacian, []), title('Laplacian High-Pass Filtered
Image');
```




```
% Apply Prewitt High-Pass Filter
h_prewitt_h = fspecial('prewitt');  % Horizontal Prewitt filter
```

```
h_prewitt_v = h_prewitt_h';          % Vertical Prewitt filter (transpose)
I_prewitt_h = imfilter(double(I_roi), h_prewitt_h);  % Convert to double
before applying filter
I_prewitt_v = imfilter(double(I_roi), h_prewitt_v);  % Convert to double
before applying filter

% Combine horizontal and vertical edges
I_prewitt = sqrt(I_prewitt_h.^2 + I_prewitt_v.^2);  % Use sqrt on numeric
arrays

% Display the Prewitt filtered image
figure, imshow(I_prewitt, []), title('Prewitt High-Pass Filtered Image');
```