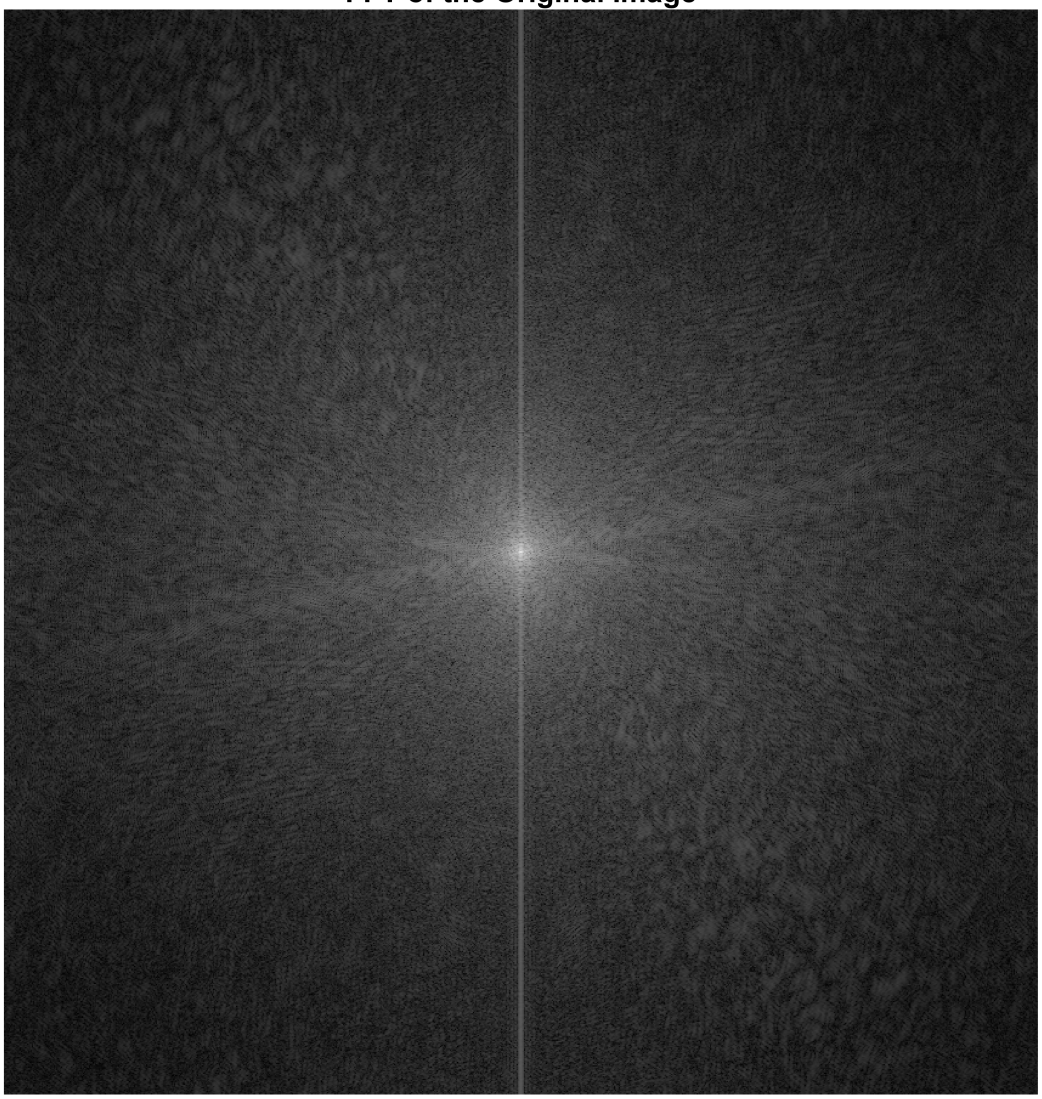
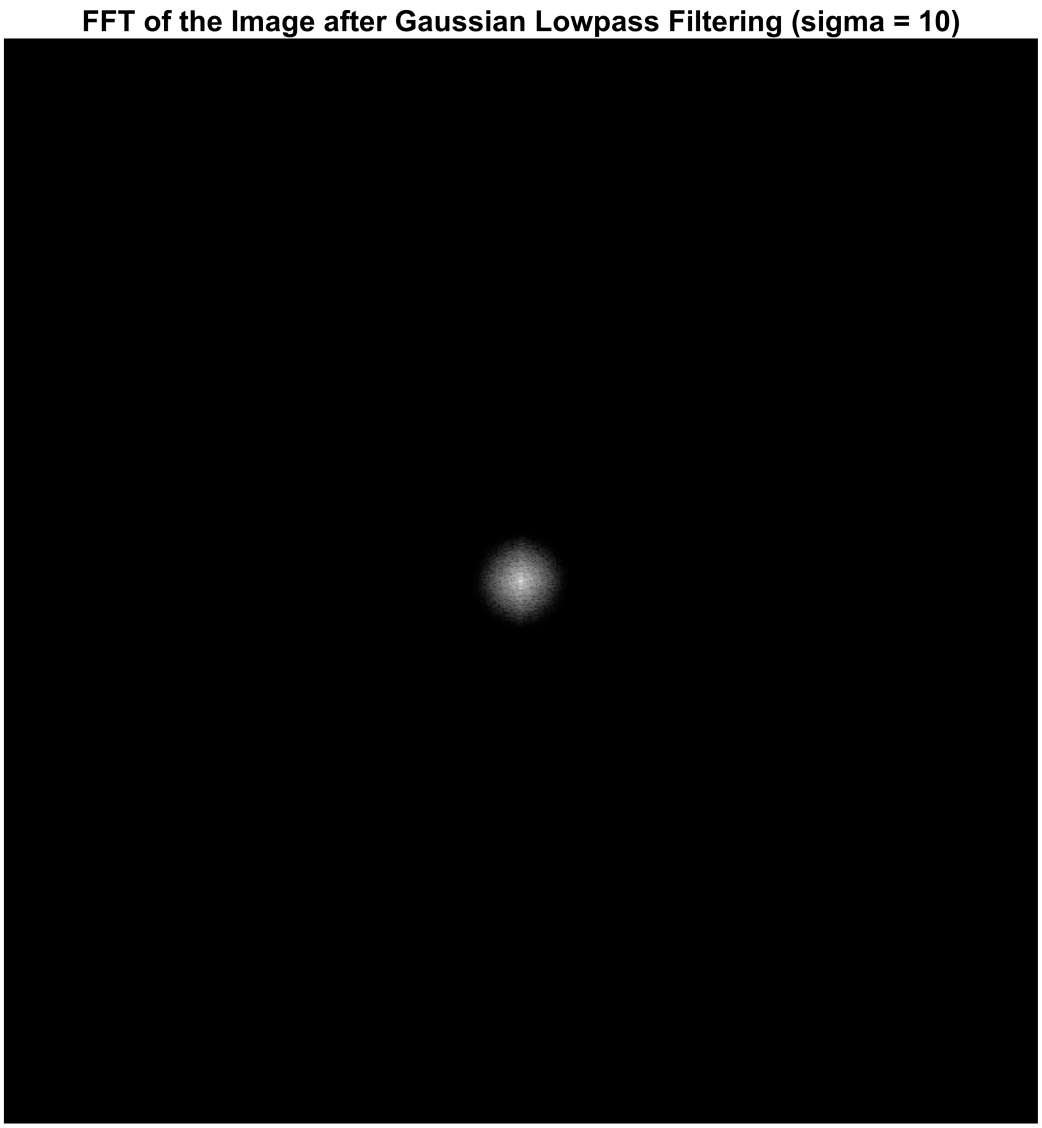
```
input_img = imread('Newton.png');
input_image = rgb2gray(input_img);
input_image = im2double(input_image);
[rows, cols] = size(input_image);

% Fourier transform to the image
cf = fft2(input_image);
cf_shifted = fftshift(cf);

% Original FFT magnitude (log scale)
figure;
imshow(mat2gray(log(1 + abs(cf_shifted))));
title('FFT of the Original Image');
```



```
%Gaussian Filter Implementation
% Low-pass Gaussian filters
g1 = mat2gray(fspecial('gaussian', [rows, cols], 10));  % (sigma=10)
g2 = mat2gray(fspecial('gaussian', [rows, cols], 30));  % (sigma=30)

% High-pass Gaussian filters
h1 = 1 - g1;
h2 = 1 - g2;

% Gaussian low-pass filters in the frequency domain
cg1 = cf_shifted .* g1;
cg2 = cf_shifted .* g2;

% Gaussian high-pass filters in the frequency domain
ch1 = cf_shifted .* h1;
ch2 = cf_shifted .* h2;

% Inverse FFT to return to the spatial domain for Gaussian filters
cgi1 = ifft2(ifftshift(cg1));
cgi2 = ifft2(ifftshift(cg2));
chi1 = ifft2(ifftshift(ch1));
chi2 = ifft2(ifftshift(ch2));

% Butterworth Filter Implementation
% Meshgrid matching the image dimensions
[x, y] = meshgrid(-floor(cols/2):floor((cols-1)/2), -floor(rows/2):floor((rows-1)/
2));

% Parameters for Butterworth filters
D = 15;

% Butterworth lowpass filter
bl = 1 ./ (1 + ((x.^2 + y.^2) / D).^2);

% Butterworth highpass filter
bh = 1 - 1 ./ (1 + ((x.^2 + y.^2) / D).^2);

% Butterworth lowpass filter in the frequency domain
cfbl = cf_shifted .* bl;
% Inverse FFT to get the lowpass filtered image
cfli = ifft2(ifftshift(cfbl));

% Butterworth highpass filter in the frequency domain
cfbh = cf_shifted .* bh;
% Inverse FFT to get the highpass filtered image
cfhi = ifft2(ifftshift(cfbh));



figure, imshow(mat2gray(log(1 + abs(cg1))));
```

```
title('FFT of the Image after Gaussian Lowpass Filtering (sigma = 10)');
```

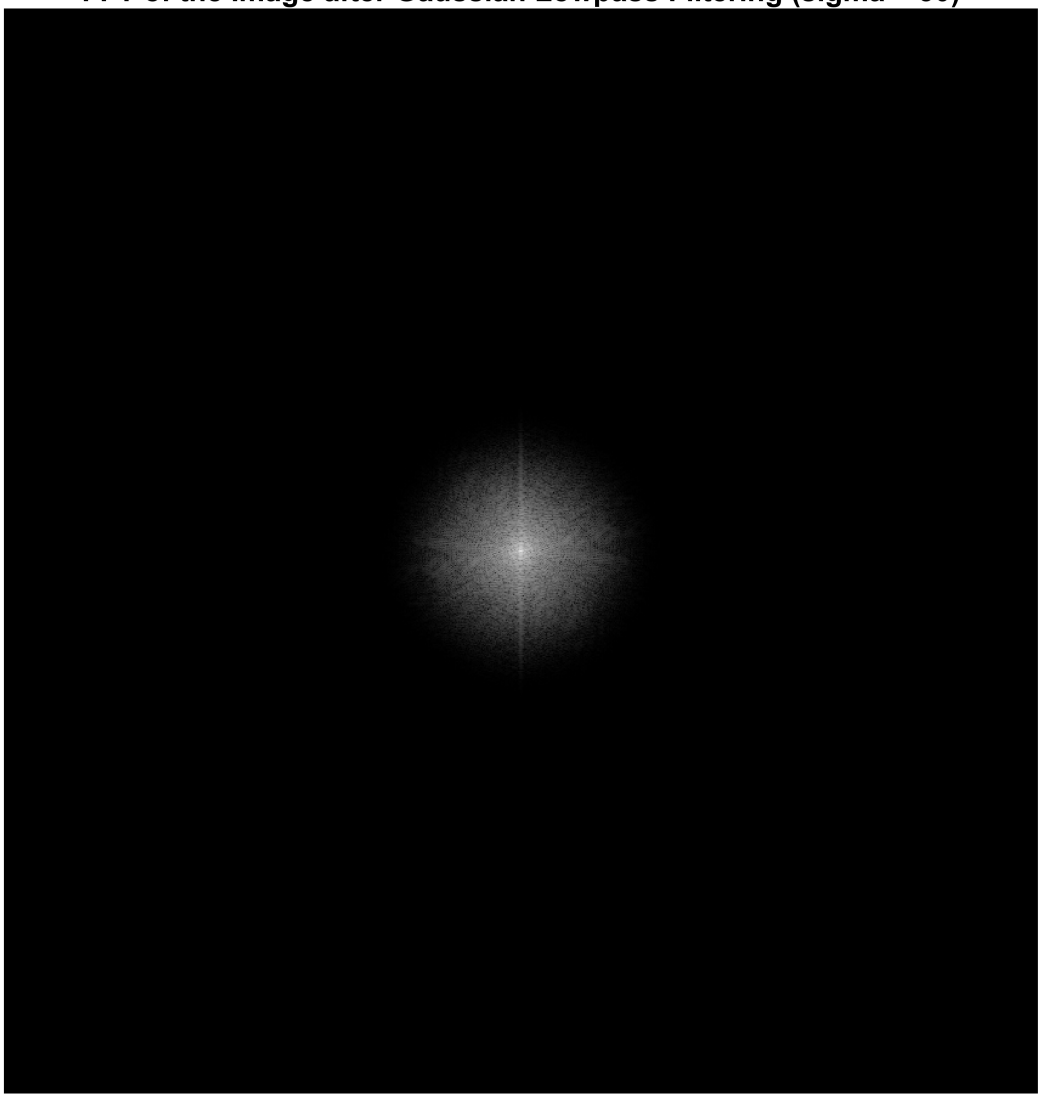



```
figure, imshow(mat2gray(abs(cgi1)));
title('Gaussian Lowpass Filtered Image (sigma = 10)');
```

Gaussian Lowpass Filtered Image (sigma = 10)

```
figure, imshow(mat2gray(log(1 + abs(cg2))));
title('FFT of the Image after Gaussian Lowpass Filtering (sigma = 30)');
```

FFT of the Image after Gaussian Lowpass Filtering (sigma = 30)

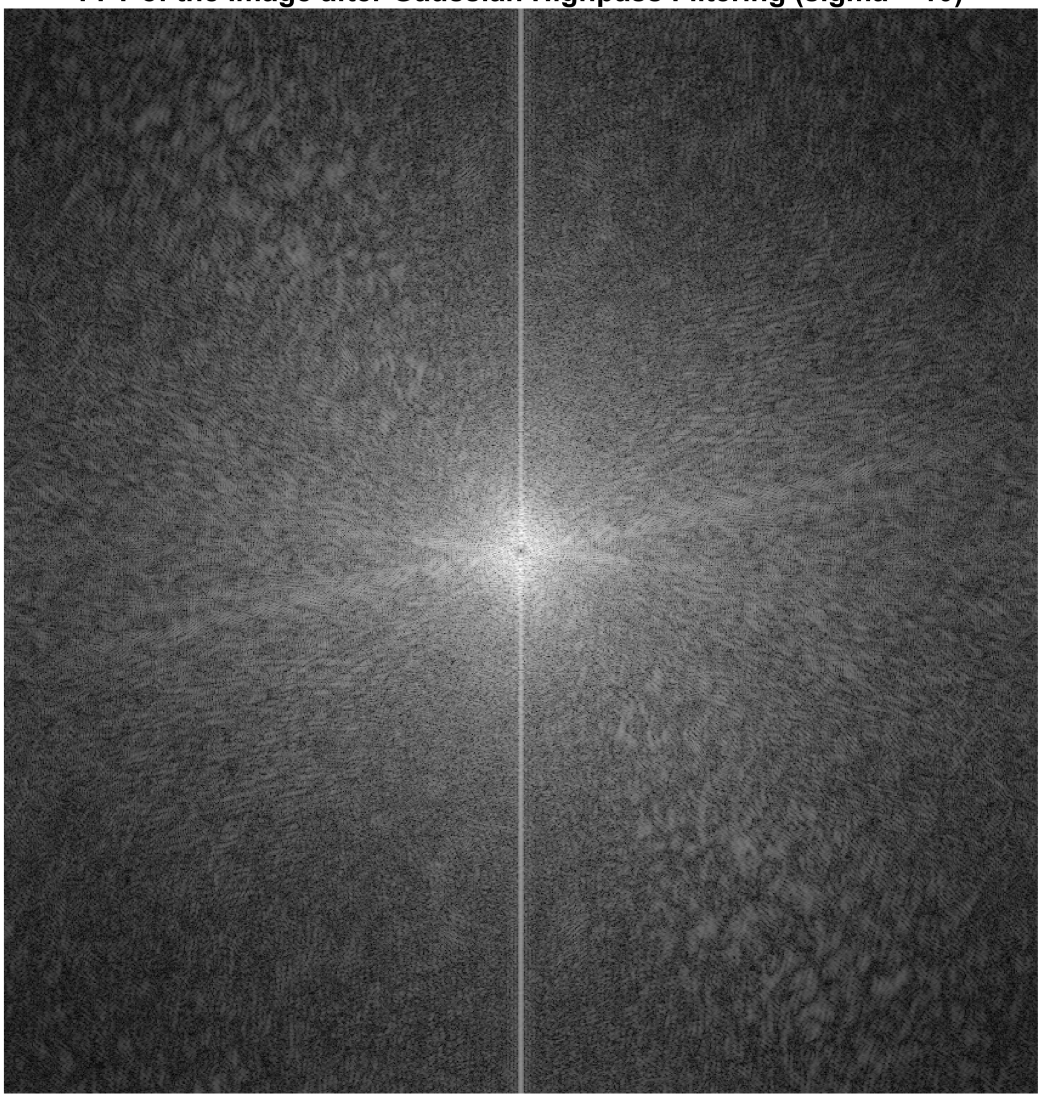


```
figure, imshow(mat2gray(abs(cgi2)));
title('Gaussian Lowpass Filtered Image (sigma = 30)');
```

Gaussian Lowpass Filtered Image (sigma = 30)

```
figure, imshow(mat2gray(log(1 + abs(ch1))));
title('FFT of the Image after Gaussian Highpass Filtering (sigma = 10)');
```

FFT of the Image after Gaussian Highpass Filtering (sigma = 10)

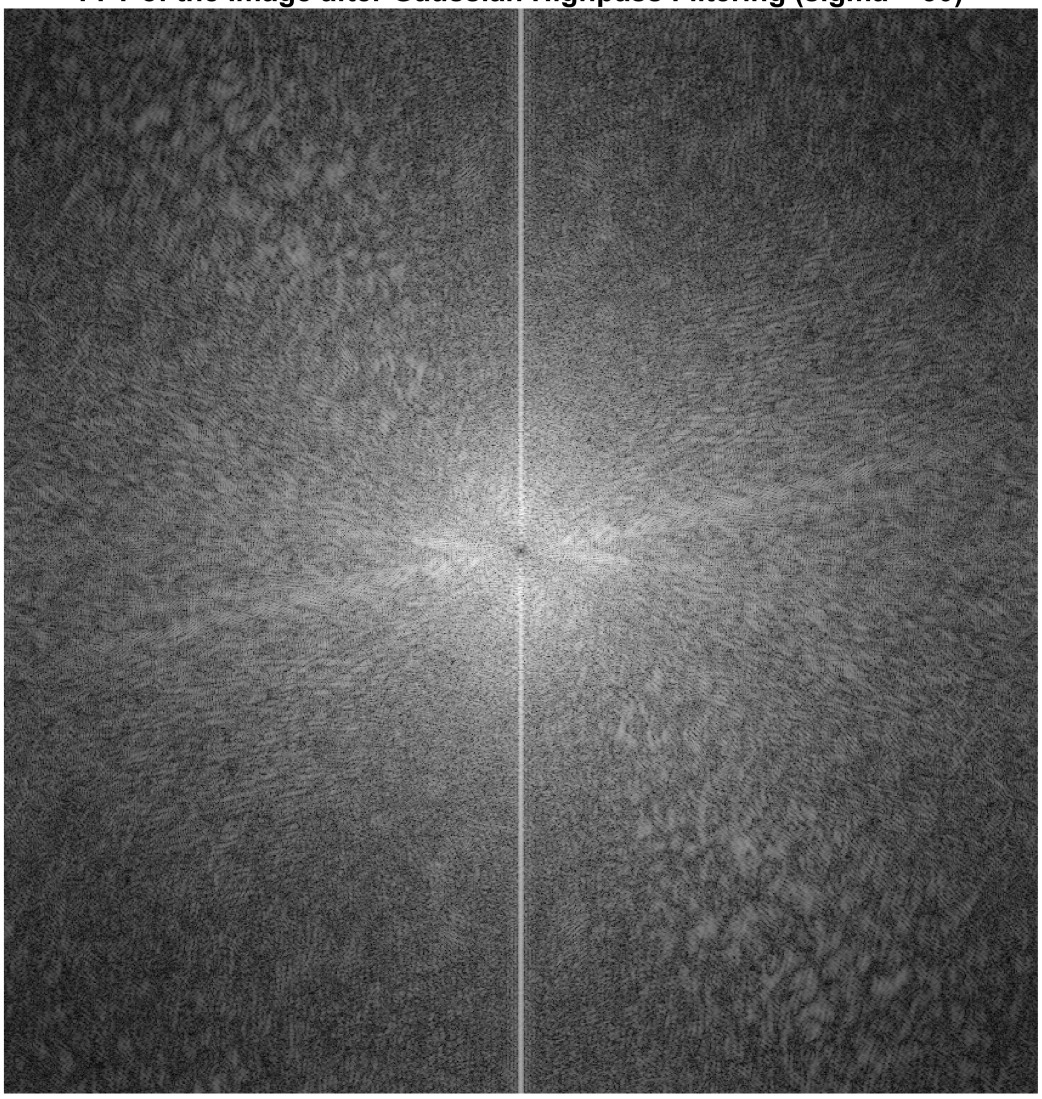


```
figure, imshow(mat2gray(abs(chi1)));
title('Gaussian Highpass Filtered Image (sigma = 10)');
```

Gaussian Highpass Filtered Image (sigma = 10)

```
figure, imshow(mat2gray(log(1 + abs(ch2))));
title('FFT of the Image after Gaussian Highpass Filtering (sigma = 30)');
```

FFT of the Image after Gaussian Highpass Filtering (sigma = 30)

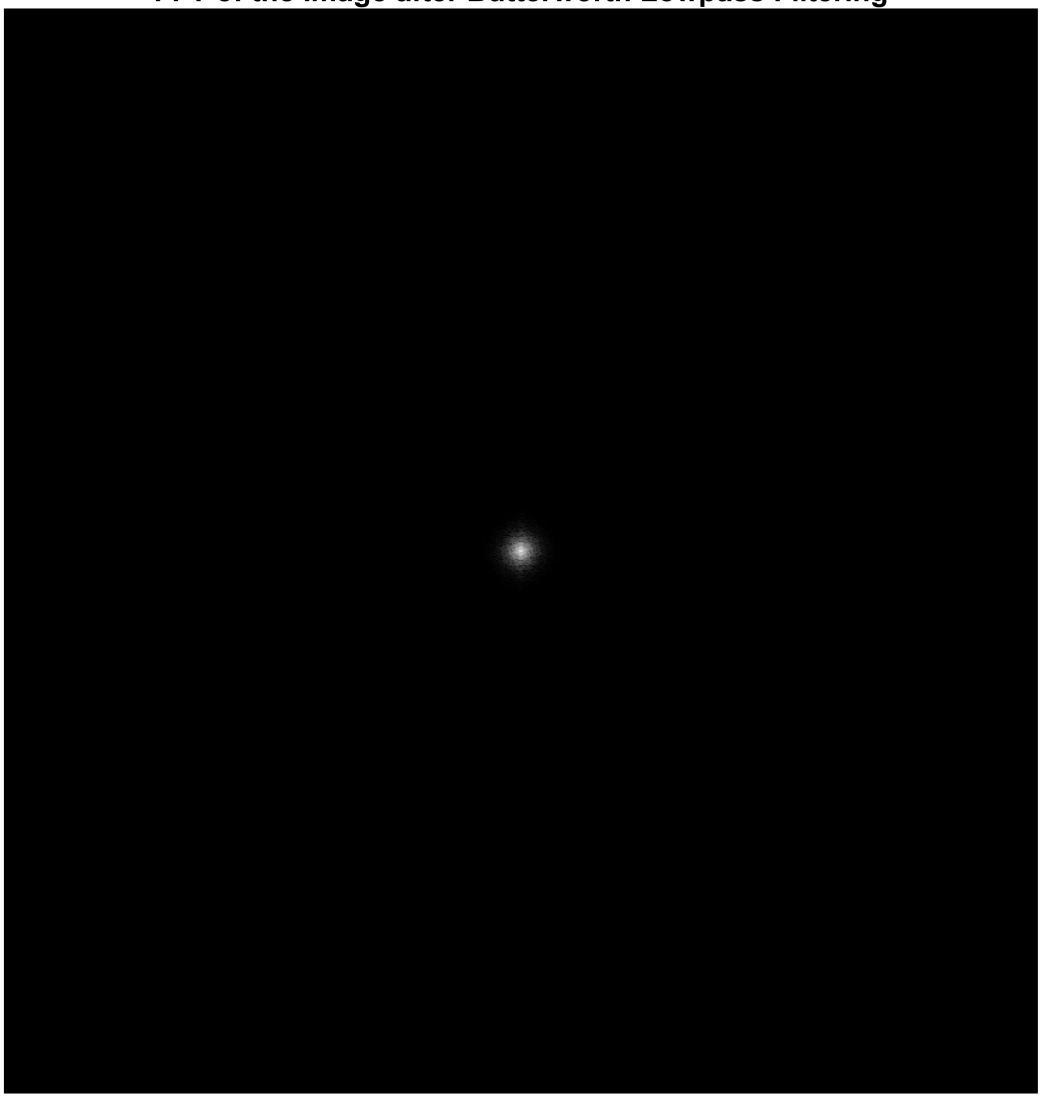


```
figure, imshow(mat2gray(abs(chi2)));
title('Gaussian Highpass Filtered Image (sigma = 30)');
```

Gaussian Highpass Filtered Image (sigma = 30)

```
figure, imshow(mat2gray(log(1 + abs(cfbl))));
title('FFT of the Image after Butterworth Lowpass Filtering');
```

FFT of the Image after Butterworth Lowpass Filtering

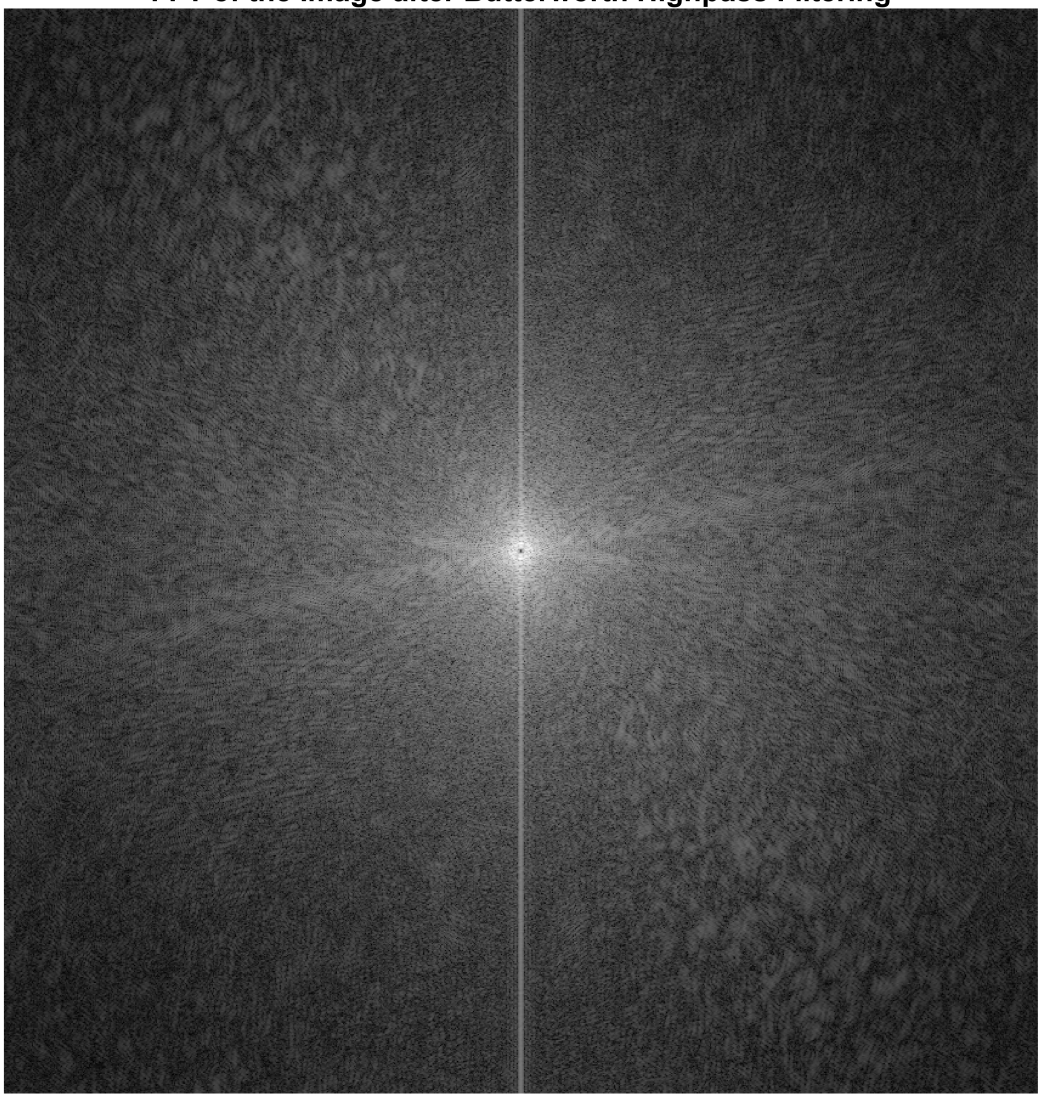


```
figure, imshow(mat2gray(abs(cfli)));
title('Butterworth Lowpass Filtered Image');
```

Butterworth Lowpass Filtered Image

```
figure, imshow(mat2gray(log(1 + abs(cfbh))));
title('FFT of the Image after Butterworth Highpass Filtering');
```

FFT of the Image after Butterworth Highpass Filtering



```
figure, imshow(mat2gray(abs(cfhi)));
title('Butterworth Highpass Filtered Image');
```

Butterworth Highpass Filtered Image