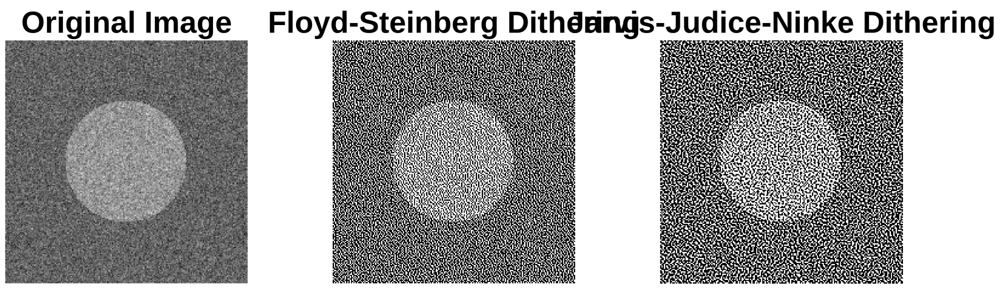
```
% Create a test image
img_size = 200;
[X, Y] = meshgrid(1:img_size, 1:img_size);
center = img_size/2;
radius = img_size/4;
img = double(sqrt((X-center).^2 + (Y-center).^2) <= radius);
img = img + 0.5*randn(img_size);
img = mat2gray(img);  % Normalize to [0, 1]

% Floyd-Steinberg Dithering
function dithered = floyd_steinberg_dither(img)
    [h, w] = size(img);
    dithered = zeros(h, w);
    for y = 1:h
        for x = 1:w
            old_pixel = img(y, x);
            new_pixel = round(old_pixel);
            dithered(y, x) = new_pixel;
            quant_error = old_pixel - new_pixel;

            if x < w
                img(y, x+1) = img(y, x+1) + quant_error * 7/16;
            end
            if x > 1 && y < h
                img(y+1, x-1) = img(y+1, x-1) + quant_error * 3/16;
            end
            if y < h
                img(y+1, x) = img(y+1, x) + quant_error * 5/16;
            end
            if x < w && y < h
                img(y+1, x+1) = img(y+1, x+1) + quant_error * 1/16;
            end
        end
    end
end

% Jarvis-Judice-Ninke Dithering
function dithered = jarvis_judice_ninke_dither(img)
    [h, w] = size(img);
    dithered = zeros(h, w);
    for y = 1:h
        for x = 1:w
            old_pixel = img(y, x);
            new_pixel = round(old_pixel);
            dithered(y, x) = new_pixel;
            quant_error = old_pixel - new_pixel;

            kernel = [
                0 0 0 7 5;
                3 5 7 5 3;
                1 3 5 3 1
            ] / 48;

            for ky = 1:3
                for kx = 1:5
                    ny = y + ky - 1;
                    nx = x + kx - 3;
                    if ny <= h && nx >= 1 && nx <= w
                        img(ny, nx) = img(ny, nx) + quant_error * kernel(ky,
kx);
                    end
                end
            end
        end
    end
end

% Apply dithering algorithms
fs_dithered = floyd_steinberg_dither(img);
jjn_dithered = jarvis_judice_ninke_dither(img);

% Display results
figure;

subplot(1,3,1), imshow(img), title('Original Image');
subplot(1,3,2), imshow(fs_dithered), title('Floyd-Steinberg Dithering');
subplot(1,3,3), imshow(jjn_dithered), title('Jarvis-Judice-Ninke Dithering');
```



```
% Compare results
mse_fs = mean((img(:) - fs_dithered(:)).^2);
mse_jjn = mean((img(:) - jjn_dithered(:)).^2);

fprintf('Mean Squared Error (MSE):\n');
```

```
Mean Squared Error (MSE):
```

```
fprintf('Floyd-Steinberg: %.4f\n', mse_fs);
```

```
Floyd-Steinberg: 0.2109
```

```
fprintf('Jarvis-Judice-Ninke: %.4f\n', mse_jjn);
```

```
Jarvis-Judice-Ninke: 0.1888
```

```
% Structural Similarity Index (SSIM)
ssim_fs = ssim(img, fs_dithered);
ssim_jjn = ssim(img, jjn_dithered);

fprintf('\nStructural Similarity Index (SSIM):\n');
```

```
Structural Similarity Index (SSIM):
```

```
fprintf('Floyd-Steinberg: %.4f\n', ssim_fs);
```

```
Floyd-Steinberg: 0.1525
```

```
fprintf('Jarvis-Judice-Ninke: %.4f\n', ssim_jjn);
```

```
Jarvis-Judice-Ninke: 0.2405
```