```
% This script is an more developed version of Spice_Simulation_8_0 where it
% is attempted to simulate the spacial coherence aspect

%Speckle_Sim_2 Speckle simulation with spherical waves

clc; clear all; close all;

load('waveOrigin_2018-07-18');
%% GLOBAL: parameters

lambda = 0.00053;   %mm
ff = 100;           %mm

sourceDensity = 0.1;  %fraction of pixels that act as sources

surfaceVariance = 0.00003; %mm
zPlanes = -10;      %mm
radiusPlane_l = 1;  %mm
radiusPlane_r = 1;  %mm

apertureSize = 0.1;   %mm
Dp_slm = 0.008;     %mm
zSLM = 0;          %mm

%% SPECKLES: parameters

res = 1024;
dx = lambda*ff/(Dp_slm*res);       %pixel size (theses are chosen so that the sampling↙
of the SLM and the fft overlap)
du_4f = 1/(res*dx);    %pixel size in fourier domain in 1/mm for propagation of the↙
right plane



%% SPECKLES: Wavefield Calculation
NN = 100; %Number of Waves

%waveOriginX = ((rand(NN,1)*(res+1)-0.5)-res/2)*dx;
%waveOriginY = ((rand(NN,1)*(res+1)-0.5)-res/2)*dx;

%dz = randn(NN,1)*surfaceVariance;

[screenX, screenY] = meshgrid(dx*(-res/2+1:res/2), dx*(-res/2+1:res/2));

waveField = zeros(res);
intensityField = zeros(res);

u_z = zeros(res);

gamma = 0;
for ii = 1:NN
    sphericalWave = exp(1i*2*pi/lambda*sqrt((zPlanes+dz(ii)).^2+(screenX+waveOriginX↙
(ii)).^2 + (screenY+waveOriginY(ii)).^2));
    waveField = waveField + sphericalWave;
```

```
    imgPlaneWavefield = sphericalWave;

    %%↙
=========================================================================================↙
================
    %% SPICE: parameters

    res = length(imgPlaneWavefield);

    du_4f = lambda * ff / (dx * res);

    %% initialization
    fourierAperture = zeros(res);

    waveField = imgPlaneWavefield;

    %% SPICE: first lens
    %DFT_il = ifftshift(fft2(double(imread('einstein.bmp'))));
    FwaveField = fft2(double(waveField));

    %% SPICE: aperture
    fourierAperture(ceil(res/2),ceil(res/2)) = 1;
    fourierAperture = (bwdist(fourierAperture) <= apertureSize/2/du_4f);       %↙
aperture in fourier domain

    FapertureWaveField = FwaveField .* fftshift(fourierAperture);

    %% SPICE: SLM propagatiom

    [uu,vv] = meshgrid(-res/2+1:res/2, -res/2+1:res/2);
    uu = du_4f*uu;
    vv = du_4f*vv;
    transferFunction = ((exp(-1i * 2*pi/lambda * zSLM * sqrt( 1 -  (uu.^2 + vv.^2) /↙
ff^2))));
    if du_4f == Dp_slm
        U_r = fftshift(transferFunction).*FapertureWaveField;
    else
        display('Error: du_4f must be equal to Dp_slm')
        return
    end


    intensityField = intensityField + abs(ifft2(U_r)).^2 + gamma*2*abs(ifft2(U_r)).↙
*sqrt(intensityField).*cos(angle(u_z)-angle(ifft2(U_r)));
    u_z = u_z + ifft2(U_r);
    ii
end
figure, imshow(angle(u_z),[])
figure, imshow(abs(u_z),[])
figure, imshow(sqrt(intensityField),[])
histogram = imhist(abs(u_z));
figure, plot(histogram)
```