White Paper: Holograms

Optical Setup

SLM-H06 is used as spatial phase modulator. At the output of the device, on the support a lens of 2 inches (for example with a focal length of 250mm) is directly fixed. The observation plane is the focal plane.

In this configuration the intensity observed is related to the Fourier transform of the light transmitted by the SLM.

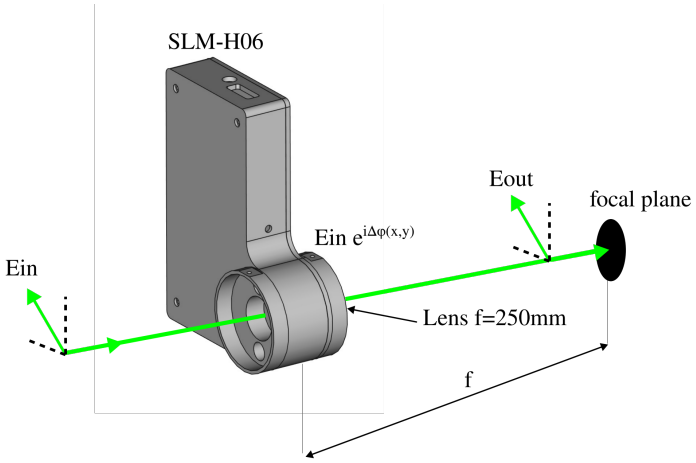


Fig.1: Optical setup

The input light is linearly polarized at 45 deg. The phase shift Δφ(x,y) is then controlled by sending an image to the SLM-H06. The output light intensity is recorded with a camera.

The position, rotation, scale, amplitude, general features of the image can be precisely tuned directly by using the software.

Example 1: Optical Vortex

An optical vortex is generated by creating its phase hologram. Basically, some fringes with a dislocation, or defect. The dislocation corresponds to the hologram of the phase singularity that characterizes the vortex. The topological charge of the dislocation gives the vortex charge.

In the appendix examples of code to generate the hologram are reported.

For example if a phase hologram of charge 1 is loaded on the SLM-H06 control software, the results on the focal plane is a vortex of charge 1 in the first diffraction order.

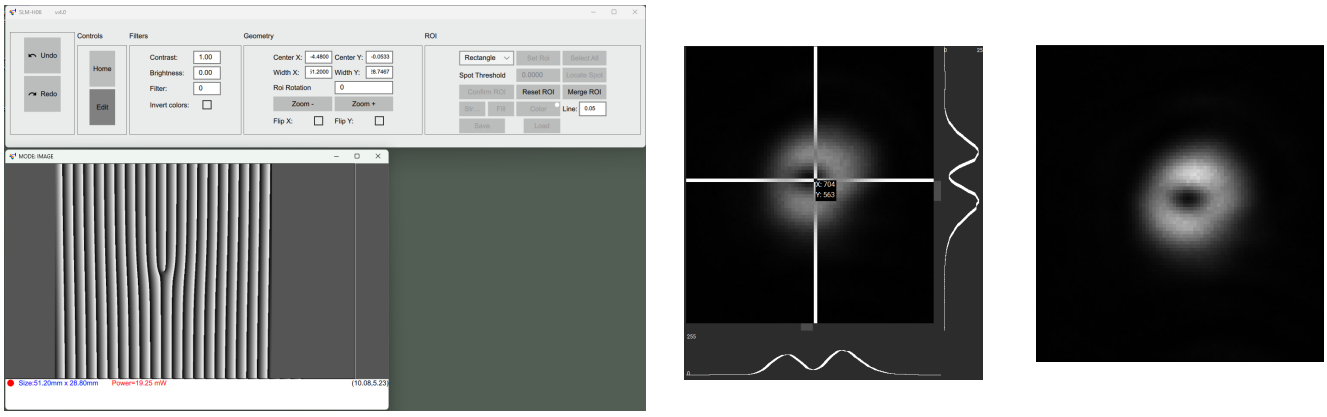


Fig.2: Optical vortex generation. Left: software window with phase hologram. Center and right: generated vortices on the focal plane.

Example 2: Flat top profile

A flat top profile is generated in the focal plane by addressing the SLM with the corresponding phase hologram. As an example, the *Gerchberg–Saxton* algorithm is used the create the corresponding image for circular or square flat top.

Below an example in the case of a square flat top

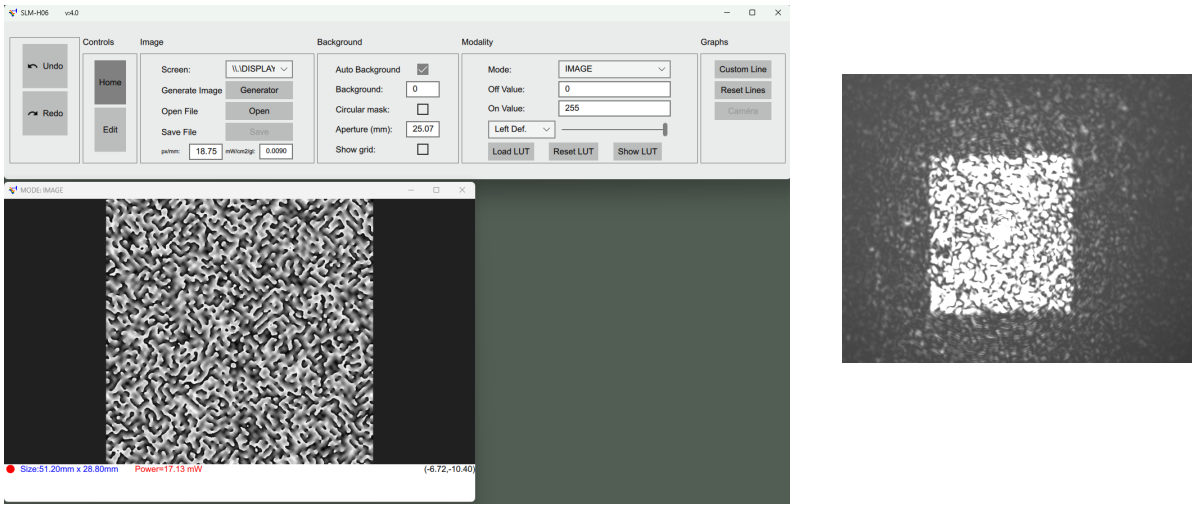
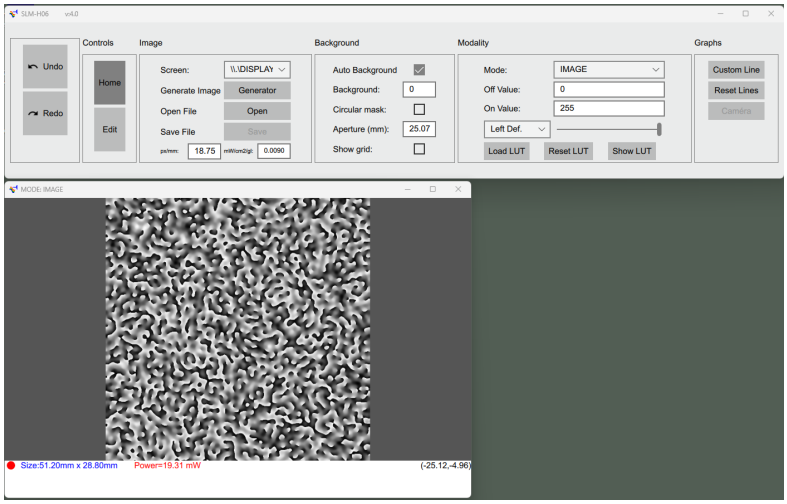


Fig.3 Generation of square flat top.

And for circular flat top



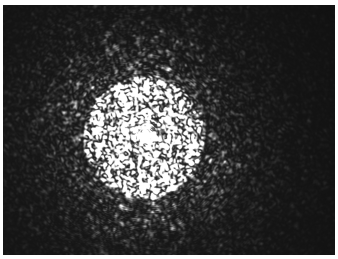
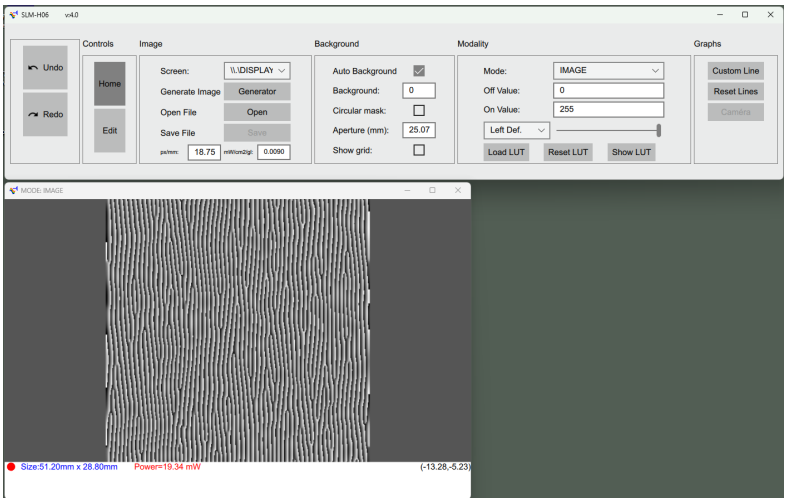


Fig.4 Generation of circular flat top.

Another example is the generation of an off-axis hologram, by incorporating a grating over the previous phase hologram.



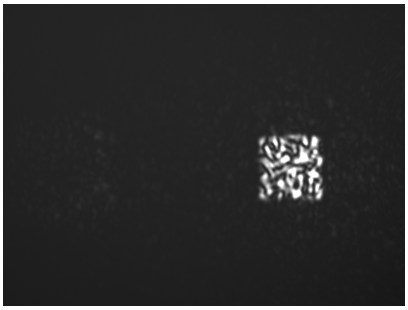


Fig.5 Generation of off-axis square flat top.

And for circular flat top

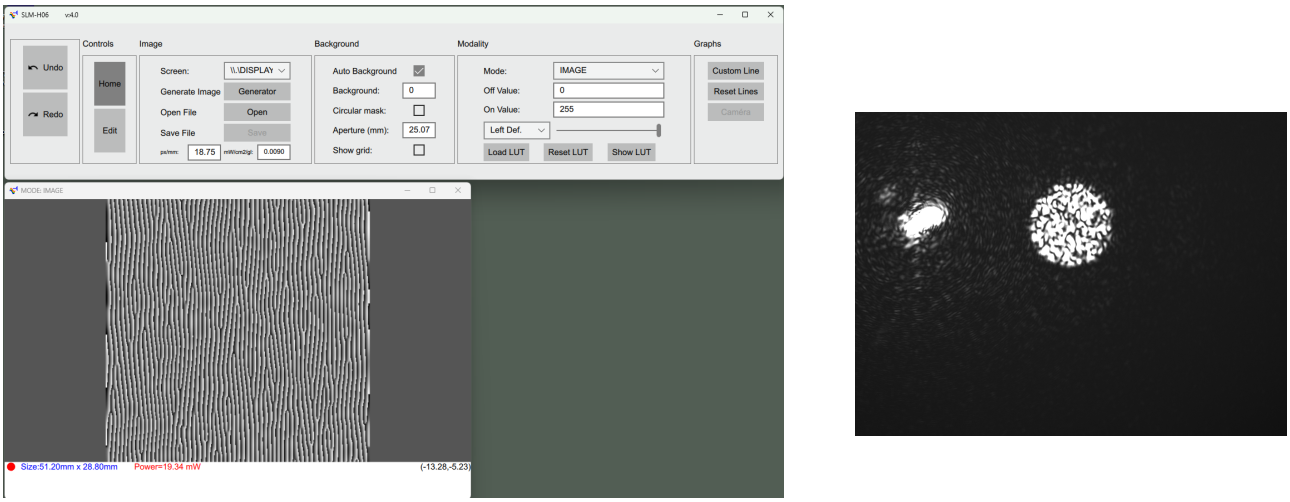


Fig.6 Generation of off-axis circular flat top.

Appendix
Examples of Octave code to generate the phase plates shown in the examples.

Vortex phase plate

```
% ----------------------------------------------------------
% Forked Diffraction Hologram (Optical Vortex Fork Grating)
% ----------------------------------------------------------

clear all; close all; clc;

% ------------------------------
% Parameters
% ------------------------------
N = 1024;           % grid size (pixels)
l = 1;              % vortex topological charge l=1,2,....
pixel_size = 8e-6;  % pixel pitch (m)

grating_period = 100e-6;  % grating period
kx = 2*pi / grating_period;  % grating spatial frequency

% ------------------------------
% Coordinates
% ------------------------------
[x, y] = meshgrid( (-N/2:N/2-1)*pixel_size );
theta  = atan2(y, x);

% ------------------------------
% Fork hologram phase
% ------------------------------
% Phase = vortex + grating
phase = l * theta + kx * x;

% Wrap to [0 , 2π)
phase_wrapped = mod(phase, 2*pi);

% ------------------------------
% Visualization
% ------------------------------
figure(1);
imagesc(phase_wrapped);
axis image off;
```

```
colormap gray;

% ------------------------------
% Save as image (optional)
% ------------------------------
phase8 = uint8(phase_wrapped/(2*pi)*255);
imwrite(phase8, "forked_hologram.png");
```

Circular flat-top

```
% ================================================================
% Square flat-top hologram with a blazed grating
% ================================================================

clear; close all; clc;

% ---------- Parameters ----------
N = 1024;              % grid size
iters = 80;            % GS iterations

% ---------- Circular flat-top (target) ----------
[x,y] = meshgrid(linspace(-1,1,N));
r = sqrt(x.^2 + y.^2);
radius = 0.08;     % flat-top radius
I_target = double(r <= radius);
A_target = sqrt(I_target);

% ---------- Random initial phase ----------
field = exp(1i * rand(N)*2*pi);

% ---------- Gerchberg–Saxton ----------
for k = 1:iters
    F = fftshift(fft2(ifftshift(field)));
    F = A_target .* exp(1i * angle(F));
    field = fftshift(ifft2(ifftshift(F)));
    field = exp(1i * angle(field));  % enforce phase-only at SLM
end

% ---------- Final phase (0 → 2π) ----------
phase = mod(angle(field) + 2*pi, 2*pi);

% ================================================================
%              ADDING A STRONG BLAZED GRATING
% ================================================================

% direction: 0 = horizontal, 1 = vertical
direction = 0;

% frequency: number of blaze periods across image
blaze_periods = 60;   % increase for more tilt

[ix, iy] = meshgrid(0:N-1, 0:N-1);

if direction == 0
    blaze = 2*pi * blaze_periods * (ix / N);    % horizontal
else
    blaze = 2*pi * blaze_periods * (iy / N);    % vertical
end

% Add blaze (wrap to 0–2π)
phase_blazed = mod(phase + blaze, 2*pi);

% ================================================================
%               Export 8-bit PNG (0–255)
% ================================================================

img8 = uint8(round(phase_blazed / (2*pi) * 255));       % replace  phase_blazed → phase to remove grating
imwrite(img8, "SLM_circular_flat_top_with_blaze.png");

fprintf("\nSaved: SLM_circular_flat_top_with_blaze.png\n");

% ================================================================
%               Display result
% ================================================================

figure; imagesc(phase_blazed); axis image;              % replace  phase_blazed → phase to remove grating
colormap(jet); colorbar;
title("Phase mask WITH STRONG BLAZED GRATING (0 → 2π)");

figure; imagesc(I_target); axis image;
colormap(gray); colorbar;
title("Square flat-top target");
```

Square flat-top

```
% ============================================================
% Square flat-top hologram with a blazed grating
% ============================================================

clear; close all; clc;

% ---------- Parameters ----------
N = 1024;              % grid size
iters = 200;           % GS iterations

% ---------- Square flat-top (target) ----------
[x,y] = meshgrid(linspace(-1,1,N));

half = 0.08;   % square half-width (normalized units)
I_target = double(abs(x) <= half & abs(y) <= half);
A_target = sqrt(I_target);

% ---------- Random initial phase ----------
field = exp(1i * rand(N)*2*pi);

% ---------- Gerchberg–Saxton ----------
for k = 1:iters
    F = fftshift(fft2(ifftshift(field)));
    F = A_target .* exp(1i * angle(F));
    field = fftshift(ifft2(ifftshift(F)));
    field = exp(1i * angle(field));  % enforce phase-only at SLM
end

% ---------- Final phase (0 → 2π) ----------
phase = mod(angle(field) + 2*pi, 2*pi);

% ============================================================
%              ADDING A BLAZED GRATING
% ============================================================

% direction: 0 = horizontal, 1 = vertical
direction = 0;

% frequency: number of blaze periods across image
blaze_periods = 60;   % increase for more tilt

[ix, iy] = meshgrid(0:N-1, 0:N-1);

if direction == 0
    blaze = 2*pi * blaze_periods * (ix / N);    % horizontal
else
    blaze = 2*pi * blaze_periods * (iy / N);    % vertical
end

% Add blaze (wrap to 0–2π)
phase_blazed = mod(phase + blaze, 2*pi);

% ============================================================
%              Export 8-bit PNG (0–255)
% ============================================================
img8 = uint8(round(phase_blazed / (2*pi) * 255));       % replace  phase_blazed → phase to remove grating
imwrite(img8, "SLM_square_flat_top_with_blaze.png");
fprintf("\nSaved: SLM_square_flat_top_with_blaze.png\n");

% ============================================================
%              Display result
% ============================================================
figure; imagesc(phase_blazed); axis image;                % replace  phase_blazed → phase to remove grating
colormap(jet); colorbar;
title("Phase mask WITH STRONG BLAZED GRATING (0 → 2π)");

figure; imagesc(I_target); axis image;
colormap(gray); colorbar;
title("Square flat-top target");
```