In our “LaserPropagation” function we pass the phase coefficient as a parameter.

Each coefficient is multiplied by a space variable on the SLM to produce a single “term” in our polynomial phase.

Each term in our polynomial expression is a “gene”.

Together, these genes create a single potential solution called a “chromosome.”

The variable our coefficient is multiplied by is x; a space variable on the SLM (512 pixels for 4 mm).

% Compute the propagation of the laser up to focus  
% Syntax: [I1,I2,I3] = LaserPropagation(PhaseCoeff,Npad)  
% PhaseCoeff = vector of size 1\*NumVar  
function [I1,I2,I3, phase] = LaserPropagation(PhaseCoeff, Npad, option)

A2=13; % Factor for defocusing in closer and more distant z plane  
 FWHM=0.4; % FHWM of laser amplitude in near field (in pixels)  
  
 NumVar=length(PhaseCoeff); % Number of polynomial coefficients  
  
 % x is space variable on the SLM (512 pixels for 4 mm).  
 % It is centered at x=0 and goes from -1 to +1  
 Nx=512; x=1:Nx; x=x-Nx/2; x=x/(Nx/2);  
  
 amp0=exp(-2\*log(2)\*(x.^2)/(FWHM^2)); % Near field amplitude

## Create the phase of member

if option % OPTION 1 : Polynomial phase  
 phase = polyval(PhaseCoeff(1:NumVar),x);  
 else % OPTION 2 : Polynomial phase on each side of x=0, with symmetry  
 x2=x(1+0.5\*Nx:Nx);  
 phase2 = polyval(PhaseCoeff(1:NumVar),x2);  
 phase1 = fliplr(phase2);  
 phase = [phase1 phase2];  
 end  
  
 amp1=amp0.\*exp(1i\*phase).\*exp(-1i\*A2\*x.^2);  
 amp2=amp0.\*exp(1i\*phase);  
 amp3=amp0.\*exp(1i\*phase).\*exp(1i\*A2\*x.^2);  
  
 amp0FF=fftshift(fft(amp0,Npad));  
 amp1FF=fftshift(fft(amp1,Npad));  
 amp2FF=fftshift(fft(amp2,Npad));  
 amp3FF=fftshift(fft(amp3,Npad));  
  
 I0=abs(amp0FF).^2;  
 I1=abs(amp1FF).^2;  
 I2=abs(amp2FF).^2;  
 I3=abs(amp3FF).^2;  
  
 I1=I1/max(I0);  
 I2=I2/max(I0);  
 I3=I3/max(I0);

end