```
% 3-D Beam Plot
tic
j=sqrt(-1);

vel=1540;                                  % Speed of sound - all units MKS
num_elems = 32;                              % Number of elements
fc=10e6;                                    % Center frequency
pitch= vel/(2*fc);                             % Array element pitch
z_foc=50e-3;                               % Range direction focal distance
theta_steer=0*pi/180;                     % Steer angle

fs=fc/64;                                  % Define a sampling frequency (not critical)
f=[fs:fs:8*fc];                            % Define an adequate frequency range
w=2*pi*f;                                  % Angular frequency radians
ns=length(f);                              % Number of samples

tdel=1.0e-6;                               % Use a fixed time offset so that base↙
wavefor is 0 for t<0 (not critical)

bw=30;                                     % Fractional bandwidth as percent
sig=bw*fc/100;                             % Width of Gausian
gauss_pulse=exp(-pi*((f-fc)/sig).^2);      % Generate Gaussian pulse (frequency domain)

gauss_pulse=gauss_pulse.*exp(-j*w*tdel);    % Apply timed delay so 0 for t<0


gauss_t=real(ifft(gauss_pulse));           % Time domain of base waveform for reference
env_gauss_t=abs(hilbert(gauss_t));         % Envelope calculation
tstep=1./max(f);                           % Time steps after using Inverst FFT
t=[1:ns].*tstep;                           % Define time axis

weight=ones(num_elems);                 % Define the weighting function (you can change↙
this)

xres = 2;
zres = 2;
yres = 2;
% Increase resolution for calculation in the x field.
x_pts=[-50:xres:50].*1e-3;                  % Define X-direction field locations
y_pts=[-50:yres:50].*1e-3;                  % Define Y-direction field locations
z_pts=[0.01:zres:50.01].*1e-3;              % Define Z-direction field locations


% Create focal delays
for i=1:num_elems

    x_elem(i)=((i-1)-(num_elems-1)./2).*pitch;  % Calculate locations of each array↙
element in x

    for j=1:num_elems
```

```
        y_elem(j)=((j-1)-(num_elems-1)./2).*pitch;  % Calculate locations of each array
element in y

        foc_del(i,j)=(sqrt(x_elem(i).^2 + y_elem(j).^2+ z_foc.^2)-z_foc)./vel; %
Calculate focusing delays

        steer_del(i,j) = i*pitch*sin(theta_steer)/vel;

        foc_del(i,j)=foc_del(i)+steer_del(i);     % You can included steering and
focusing in one line - in fact it works better that way

    end
end

for j=length(z_pts):length(z_pts)                              % Just do z = 50 mm.

    for i=1:length(x_pts)
        sum_pulse=zeros(size(gauss_pulse)); % Initialize sum of waveforms to zero before
starting loop


        for k = 1:length(y_pts)

            for index_x=1:num_elems

                for index_y = 1:num_elems

                    prop_del = (sqrt((x_elem(index_x)-x_pts(i)).^2 + (y_elem(index_y)-
y_pts(k)).^2 + z_pts(j)^2))./vel;

                    sum_pulse = sum_pulse+weight(index_x,index_y).*gauss_pulse.*exp(-sqrt
(-1)*w.*(prop_del-foc_del(index_x,index_y)));

                end
            end

            sum_pulse_t=real(ifft(sum_pulse));  % Convert to time domain
            field_val(j,i,k)= max(abs(hilbert(sum_pulse_t)));

        end

        field_val(j,i,:)=field_val(j,i,:)./max(field_val(j,i,:));

    end

end

field_db=20*log10(field_val);

f = zeros(51);
```

```
for i = 1:51
    for j = 1:51
        f(i,j) = (field_db(length(z_pts), i ,j));
    end
end
% Plot it for z = 50 mm.
view(3)
mesh(y_pts, x_pts, f)
view(3)
toc
beep
beep
beep
beep
```