```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%% LOOKUP TABLE GENERATION  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mua_lut=[];            % Range of absorption coefficients
musp_lut=[];           % Range of scattering coefficients
freq_LUT=[];           % Range of spatial frequencies used by imaging device


for imua=1:size(mua_lut,2)
    for imusp=1:size(musp_lut,2)
        LUT_MC(imua,imusp,:) = MCref_DM([mua_lut(imua) musp_lut(imusp)],n,freq_LUT,2);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function R = fresnel(n_in,n_out,theta);
% Fresnel reflectance function required for Monte Carlo simulation
theta_prime = (n_in/n_out)*sin(theta);
cos_theta = cos(theta);
cos_theta_prime = cos(theta_prime);
R = 0.5*((n_in*cos_theta_prime - n_out*cos_theta)./(n_in*cos_theta_prime+n_out*cos_theta)).^2 +
0.5*((n_in*cos_theta - n_out*cos_theta_prime)./(n_in*cos_theta+n_out*cos_theta_prime)).^2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function y = MCref_DM(p,f,n,sim_type)
%=======================================================================================
% Inputs:
% p = [mua musp] - array of optical coeffecients, dimensionally 1/mm
% f = spatial frequencies (Mx1 column vector, dimensionally 1/mm)
% n = index of refraction; note this is usually either 1.33 or 1.41
% sim_type = generate your own reflectance grid or borrowed from literature
%
% Output:
% y = Reflectance in frequency vs rho at given optical properties and s-d separations
% R for each frequency and each pair of optical properties [mua musp]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%persistent place sim_types here;
load('sim_type1')
f=reshape(f,length(f),1);
c=300; % speed of light in vacuum
v=c/n;  %  speed through medium in mm/ns
y=zeros([length(mua_lut) length(musp_lut) length(f)]);
y_at_rho=zeros([length(sim_Rho) 1]);
    max_step=length(mua_lut)*length(mua_lut);
    for j=1:length(mua_lut)
        for k=1:length(musp_lut)
        mua= mua_lut (j);
        musp= musp_lut (k);
        for r_idx = 1:length(sim_Rho)
            y_at_rho(r_idx) = sum( sim_Rho(r_idx,:).*exp(-
mua*v*t_MC/musp).*t_interval/musp)*musp^3/(1-fresnel(1,n,0)); %Reflectance curve
        end;
        for f_idx = 1:size(y_at_fx,3)
            y(j,k,f_idx) = sum( y_at_rho.* besselj( 0, 2*pi*f(f_idx).*sim_Rho/musp
).*(2*pi*sim_Rho) .*sim_R_int/musp^2);
        end
        end
    end
```