Appendix D

MATLAB CODE

Implementation of generalized couple wave analysis

9/26/16 12:02 PM /Users/sunitadarbe/Documents/multiwave.m 1 of 3

function [Angle,DETETotOutTE,DETMTotOutTM] = multiwave(phi,l,d,theta,n,n1,w1,modes,varargin)
    % MULTIWAVE gives the diffraction efficiency of 2p+1 output modes for
    % given input wavelength, angle and grating parameters; from R. Magnusson
    % and T. K. Gaylord, "Analysis of multiwave diffraction of thick gratings"
    % (1977)
    %
    % INPUTS
    % d = thickness of grating in microns
    % phi = angle of grating in radians
    % theta = incident angle in radians
    % w1 = incident wavelength in microns
    % L = period of grating in microns
    % n = average index of refraction of grating
    % n1 = amplitude index of refraction modulation
    %
    % OUTPUTS
    % AngOut = output angle for each of the 2p+1 outputs considered in
    % DEGREES!
    % DETE = diffraction efficiency of the output mode
    % exiting at output angle AngOut for polarization perpendicular to
    % the plane of incidence
    % DETM = diffraction efficiency of the output mode
    % exiting at the output AngOut for polarization in the plane of
    % incidence
    % TotOutTE = sum of DETM for all 2p+1 modes
    % TotOutTM = sum of DETE for all 2p+1 modes
    %
    % ASSUMPTIONS
    % Completely lossless material assumed.

    p = modes;
    K = 2*pi/l;
    b0 = 2*pi*w1/l;
    C = -1i*w2*pi*w1/wl;
    D = 1i*w2*pi*w1/wl;
    v=zeros(2p+1,1);
    c=zeros(2p+1,1);
    AngOut = zeros(2p+1,1);
    ModeNum = 1:1:2p+1;
    k = ModeNum-1;
    v = k*K*cos(theta-phi)-k*K^2/2/b0; % dephasing factor
    c = cos(theta)-k*K*cos(phi)/b0; % obliquity factor
    AngOut = atan((b0*sin(theta)-(ModeNum-p-1)*k*K*sin(phi)))
    ./((b0*cos(theta)-(ModeNum-p-1)*k*K*cos(phi)));
    Angle = AngOut*180/pi;

    % Solve Coupled ODEs for a 2p+1 modes (TE and TM)

    BCS = zeros(2p+1,1);
    BCS(1) = 1;
RSof data import into MATLAB and plotting
else
dS(k) = -1/c(k) * (1i*v(k)*S(k)+ 1i/2*( S(k-1)*C+S(k+1)*D ));
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TM, non-perp (E-mode) polarization
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function dS = MWTMSolve(z,S)
global C c D v p AngOut
dS = zeros(2*p+1,1); % a column vector, with p even
dS(1) = -1/c(1)*(1i*v(1)*S(1)+ 1i/2*( S(2)*D*cos(AngOut(1)-AngOut(2))));
dS(2*p+1) = -1/c(2*p+1); %
*(1i*v(2*p+1)*S(2*p+1)+1i/2*( S(2*p)*D*cos(AngOut(2*p)-AngOut(2*p+1))));
for k = 2:1:2*p
  if k == p+1
    dS(k) = -1/c(k)*(1i/2*(S(k-1)*C*cos(AngOut(k-1)-AngOut(k)+ S(k+1)*D*cos(AngOut(k)-AngOut(k+1))));
  else
    dS(k) = -1/c(k)*(1i*v(k)*S(k)+1i/2*(S(k-1)*C*cos(AngOut(k-1) -AngOut(k)))+S(k+1)*D*cos(AngOut(k)-AngOut(k+1))));
  end
end
function [phi, L] = Angle(deswl, ThOut, theta, n)
format compact

%==========================================================================
%ANGLE finds phi and L for a grating that optimizes diffraction efficiency
%of diffraction order +1 for a given wavelength and output angle according
%to the equation in Gaylord
%==========================================================================

% INPUTS
% deswl = free space wavelength of input light in MICRONS
% ThOut = output angle in DEGREES; direction in which high diffraction
% efficiency is desired
% theta = input angle in DEGREES
% n = average index of refraction of the HOE
%==========================================================================

% OUTPUTS
% phi = angle of the grating in DEGREES; defined as the angle between the
% grating normal and the K vector which is in the direction of
% sinusoidal variation. phi range is from -90 to +90 degrees
% L = period of sinusoidal variation of refractive index in the grating
% in MICRONS
%
%==========================================================================

% ASSUMPTIONS
% Simple holographic grating with sinusoidal variation in refractive
% index in a complete lossless material
%
%==========================================================================

% Finding PHI
%--------------------------------------------------------------------------
if ThOut < 0
    phi = 90 - abs((ThOut+theta)/2); % if bending light in negative x direction
    L = double(findL(phi,deswl,ThOut,n));
elseif ThOut > 0
    phi = -90 + abs((ThOut-theta)/2); % bending light in positive x direction
    L = double(findL(phi,deswl,ThOut,n));
else
    phi = NaN;
    L = NaN;
end

% Finding L
%--------------------------------------------------------------------------
function [L] = findL(phi,wl, ThOut, n)
syms period
b0 = 2*pi*n/wl;
K = 2*pi/period;
% solving for the period
S = solve(tand(ThOut) == (-K*sind(phi))./ (b0-K*cosd(phi)), period);
% makes the solution S into a floating point number
L = vpa(S);
%% Hemispherical average with no azimuthal dependence
angle = [(phi(2)-phi(1))/((1:length(phi))) * pi /180];
dtheta = phi(2)-phi(1)*pi/180;

%% Calculate the hemispherical solid angle associated with each angle
Hemis = 2*pi * sin(angle) .* cos(angle) * dtheta;

%% Calculate the hemispherical irradiance (assumed totally diffuse
%% sunlight) weighted R and T and normalize by the total projected
%% solid angle of the hemisphere (should be pi if integrating over angle =
%% 0 to pi)
TotalRef = (Hemis * R_test)/sum(Hemis);
TotalTrans = (Hemis * T_test)/sum(Hemis);

%% Making a well-labeled color plot
figure('units','normalized','outerposition',[0 0 1 1])
for e=1:structSize
    subplot(B,D,e)
    pcolor(eval(xvar),eval(yvar),LSCHCG14(e).Rte)%
    shading flat; colorbar;
    caxis([0 1])
    title(strcat(LSCHCG14(e).name));
    fontsize',fontSize)
    set(gca,'FontSize',fontSize)
    xlabel(Xlabel,'FontSize',fontSize)
    ylabel(Ylabel,'FontSize',fontSize)
end

h=colorbar;
figtitle([ftitle,' Rte'])
title(h,'Rte','FontSize',fontSize);
colormap jet
saveas(gcf,[ftitle,'Rte.jpg']); saveas(gcf,[ftitle,'Rte.fig'])

%% Read data from text files into MATLAB (e.g., RSoft .dat files)

P = dir('*.dat');
fileinfo=dir('*.syms');
R_col=1;
T_col=2;

for ji = 1:Number_of_files
    name = fileinfo(ji).name;  
    sym file name
    runIdj = fopen(name);
    C=textscan(runIdj,'% %f','%delimiter','=',''); % gets variable values
    scaninfo_uns(ji,:)=C(2);  
    % stores them
    fclose(runIdj);  
    % closes sym file
    M=dlmread(files(ji).name,'1,1');  
    % opens .dat file of same name
    R(ji,:)=M(:,R_col);  
    T(ji,:)=M(:,T_col);  
end