Problem #1: star() function

The Redheffer star product can occur frequently in different parts of a code, particularly when complicated devices are being simulated. It is cumbersome to repeat the code which implements the star product. Instead, write a generic function in MATLAB using structures that performs the star product procedure. Do not use the inv() function in MATLAB. Be sure that your function works with any size scattering matrix, not just 2×2 or 4×4. Use the following header for your function:

```
function S = star(SA,SB)
% STAR      Redheffer Star Product
%
% S = star(SA,SB)
%
% INPUT ARGUMENTS
% ================
% SA        First Scattering Matrix
%   .S11    S11 scattering parameter
%   .S12    S12 scattering parameter
%   .S21    S21 scattering parameter
%   .S22    S22 scattering parameter
%
% SB        Second Scattering Matrix
%   .S11    S11 scattering parameter
%   .S12    S12 scattering parameter
%   .S21    S21 scattering parameter
%   .S22    S22 scattering parameter
%
% OUTPUT ARGUMENTS
% ================
% S         Combined Scattering Matrix
%   .S11    S11 scattering parameter
%   .S12    S12 scattering parameter
%   .S21    S21 scattering parameter
%   .S22    S22 scattering parameter
%
```

Verify that your star() function works correctly, by running the test_star.p script from the course website. Report this in your homework.

Problem #2: cascn() function

Scattering matrices provide a very efficient way to simulate longitudinally periodic structures composed of hundreds, thousands, and even millions of periods. Write a MATLAB function that uses the doubling algorithm to cascade any integer number of periods, not just powers of two. Do not use the inv() function in MATLAB. Be sure that your function works with any size scattering matrix, not just 2×2 or 4×4. Use the following header for your function:

```
function SC = cascn(S,N)
% CASC      Cascading and Doubling Algorithm
%
% SC = cascn(S,N)
%
% This MATLAB function uses an efficient doubling algorithm
% to cascade N periods.
%
% INPUT ARGUMENTS
% ================
% S         Scattering Matrix for One Period
%   .S11    S11 scattering parameter
%   .S12    S12 scattering parameter
%   .S21    S21 scattering parameter
%   .S22    S22 scattering parameter
%
% N         Number of scattering matrices to cascade
%
% OUTPUT ARGUMENTS
% ================
% SC        Overall Scattering Matrix for Cascade
%   .S11    S11 scattering parameter
%   .S12    S12 scattering parameter
%   .S21    S21 scattering parameter
%   .S22    S22 scattering parameter
```

Verify that your cascn() function works correctly, by running the test_cascn.p script from the course website. Report this in your homework.

Problem #3: Revise TMM1D Algorithm

Revise your previous `tmm1d` code to use `star()` to perform all of the star products and `cascn()` to efficiently cascade any number of unit cells. Interpret the layers described by `ER`, `UR`, and `L` as a single unit cell of a longitudinally periodic device. The new parameter `NP` in the dashboard quantifies how many repetitions of this unit cell comprises the overall device.

```
% Homework #4, Problem #3
% EE 5337 - COMPUTATIONAL ELECTROMAGNETICS
%
% This MATLAB program implements the transfer matrix method using
% star() and cascn().

% INITIALIZE MATLAB
close all;
clc;
clear all;

% UNITS
degrees = pi/180;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% DEFINE SIMULATION PARAMETERS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% SOURCE PARAMETERS
lam0  = 1;                        %free space wavelength
theta = 0 * degrees;              %elevation angle
phi   = 0 * degrees;              %azimuthal angle
pte   = 0;                        %amplitude of TE polarization
ptm   = 1;                        %amplitude of TM polarization

% EXTERNAL MATERIALS
ur1 = 1.0;                        %permeability in the reflection region
er1 = 1.0;                        %permittivity in the reflection region
ur2 = 1.0;                        %permeability in the transmission region
er2 = 1.0;                        %permittivity in the transmission region

% DEFINE LAYERS
UR = [ 1 3 1 ];             %array of permeabilities in each layer of unit cell
ER = [ 3 3 3 ];             %array of permittivities in each layer of unit cell
L  = [ 0.4 0.8 1.3 ];       %array of thickness of each layer of the unit cell
NP = 78536124;              %number of times to cascade the unit cell

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% IMPLEMENT TRANSFER MATRIX METHOD
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

It is recommended to benchmark your revised code by reproducing at least some of the results from Homework #3.

Problem #4: Test the Revised Code

Perform the following simulations and record the reflectance and transmittance for two cases using circular polarization.

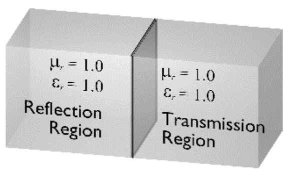
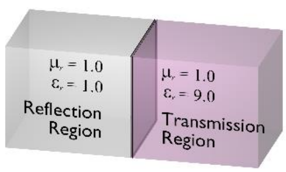
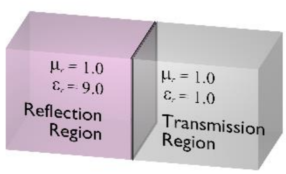
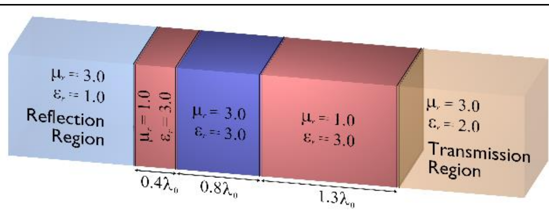
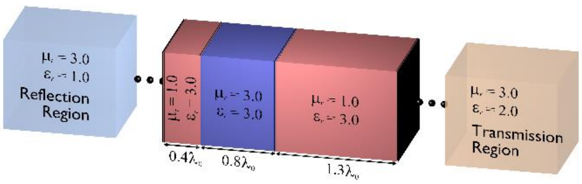
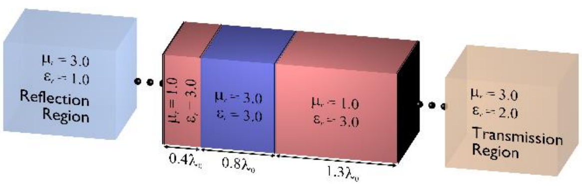
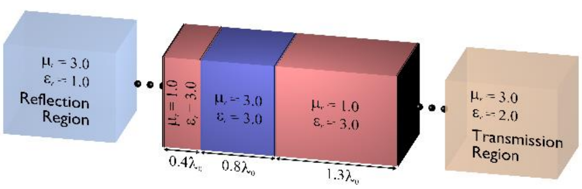
Device	Case 1			Case 2		
	Normal Incidence			CP, θ=55° and ϕ=125°		
	R	T	R+T	R	T	R+T
						
						
						
						

Table continued on next page.

Table continued from previous page.

Device	Case 1			Case 2		
	Normal Incidence			CP, θ=55° and ϕ=125°		
	R	T	R+T	R	T	R+T
`NP = 1`  						
`NP = 6`  						
`NP = 89`  						
`NP = 78,536,124` This is a single number: Seventy-eight million, five hundred thirty-six thousand, one hundred twenty-four