

# Constrained Optimization of Light Or Radiation (COLOR)

```
%%%% Assumptions
% 1) user spds are at 100%, so we divide by 100 to get spd/percent. Then
% we can multiply by 50 or 100 to get the real percent values.
% 2) for xyTarg, we assume we want the Y value to be 100. I don't see this
% being a problem. The spds are scaled to that later during initial percent
% establishment
```

```
% Reset everything
close all; clear all; clc;
format shortG;
%directory handling
currentDir = pwd;
if currentDir(end-12:end) ~= "MetaMesmerize"
    error("Set address bar to lowest folder of MetaMesmerize")
end
addpath(genpath(pwd));
%show version
matlabVersion = ver().Release
```

```
matlabVersion =
'(R2021a)'
```

```
disp("Your Matlab is " + string(matlabVersion) + ": I Used 2021a To Create This");
```

```
Your Matlab is (R2021a): I Used 2021a To Create This
```

## User Inputs And Choices

Choose what spds you are trying to optimize.

1. If you choose synthetic, you will need to input the peaks and fwhms of gaussian spds.
2. If you choose UserCSV, ensure it is all numbers and it starts at A1 and is 380:780. Do not have a wavelength column. It will know based on length, as long as the range is 380 to 780nm
3. The UserSPDs are assumed to be at 100%.

## Source of SPD (Synthetic or userSPDs file)

```
spdSource = "SYNTHETIC";
if spdSource == "USERSPDS"
    disp("You chose to use the UserSPDs.xlsx")
    userFileName = "RealSPDs/UserSPDs.xlsx";
    [userSPDs, wInt_user] = importUserSPDs(userFileName, "UserSPDs");
elseif spdSource == "SYNTHETIC"
    disp("You will use the synthetic SPDs that you chose")
end
```

```
You chose to use the UserSPDs.xlsx
```

No negative values found  
All the intervals were the same:5 nm

## Necessary Decisions

These are choices you must make regardless of whether you choose synthetic SPDs or user SPDs

If you are loading spd's, choose which file and what column ids. If empty, it loads the entire sheet

```
userChannels = str2num("");
```

This just changes the chromaticity diagram you can look at during the optimization. Currently only 2 degree.

```
Observer = 2;  
switch Observer  
    case 2  
        disp("Your graph will be in 2 degree standard observers")  
    case 10  
        error("10 degree evaluation not implemented yet. It can be easily done though")  
end
```

Your graph will be in 2 degree standard observers

## How Do You Want To Choose The Initial Channel Percentages?

Once the SPDs are generated or loaded, you need to give the optimization an initial starting point.

1. Equal Percentages, 50%. **Each channel will be set to 50%** of their value that they were loaded as or created as.
2. Percentages such that they reach an **xy chromaticity** in the observer chosen above. Ytarget is assumed to be 100. This might cause problems with user spd's, so it is recommended for now to choose equal percents

```
initialGuessMode = "All_50_Percent"
```

```
initialGuessMode =  
"All_50_Percent"
```

```
switch initialGuessMode  
    case "All_50_Percent"  
        disp("Your initial mix percent will have 50% on for each channel")  
    case "xyTarget"  
        disp("Your initial mix percent will be one of perhaps many to get to xTarg, yTarg")  
        xyTarg = str2num("0.33, 0.45")  
        XYZTarg = 100*[xyTarg(1:2), 1-sum(xyTarg(1:2))] ./ xyTarg(2);  
        disp("This will be in " + Observer + " Degree")  
    otherwise  
        error("initialGuessMode error")  
end
```

Your initial mix will have 50% on for each channel

Currently, the code has a universal wavelength range from 380 to 830 nanometers. If you are loading SPDs, you don't need to enter anything here, as the program will find the interval and interpolate all alpha opics and cmfs to that interval.

If you are making synthetic spds, then you can generate spds with 1nm (380, 381, 382...) , 2nm, or 5 nm. Going from 1nm to 5nm only speeds up the code by maybe 10%.

```
if spdSource == "SYNTHETIC"
    wlInt = 2; %decide what interval to do calculations in
elseif spdSource == "USERSPDS"
    wlInt = wlInt_user;
end
wl = 380:wlInt:780;
switch wlInt
    case 1
        disp("You will optimize in 1nm increments of wavelengths")
    case 2
        disp("You will optimize in 2nm increments of wavelengths")
    case 5
        disp("You will optimize in 5nm increments of wavelengths")
    otherwise
        error("wavelength interval is not accepted. They must be from 380 to 780 in 1,2, or 5nm")
end
```

You will optimize in 5nm increments of wavelengths

## Load Color Matching Functions And Plot

The program loads these tables inside the functions, but they will be plotted as an extra precaution for error checking and debugging. Commented out for test branch

```
%%%%%%%%%%%% Load Color Matching Functions and Maybe Plot
%%%%%%%%%%%% %%%%%%%%%%%%%
%%%%%%%%%%%% %%%%%%%%%%%%%

% load the standards and make a table out of CMFs and an array out of alpha
% opics. Interpolate them such that they match the wlInt input above
load('Standards\TM-30-18_tools_etc\TM30_V204.mat');
load('Standards\CIES026_E2018\CIE_S026_AlphaOpic_v1049a.mat','T_Alpha_Opic_Radiometric');
A_Alpha_Opics = table2array( T_Alpha_Opic_Radiometric(1:wlInt:end, 2:end) ); %first col was wl

A_CMFS2deg = [A_CMF(1:wlInt:end, 2:4)];
A_CMFS10deg = [A_CMF(1:wlInt:end, 5:7)];
switch Observer
    case 2
        A_CMFS_Eval = A_CMFS2deg;
    case 10
        error("10 degree diagram not yet implemented")
```

end

## Synthetic SPD Generation

### SPD Peak Wavelength Input (List or Equal Interval)

List Example: " 420 540 450 670 455 " (no commas)

Equal Interval: "450:50:600" --> 450, 500, 550, 600

To use interval mode, remove text until you see "Enter Text" in the list input, otherwise it will assume you want the list input. You can still have values on the sliders for the interval mode, but it won't use them unless the list input says "Enter Text"

```
if spdSource == "SYNTHETIC"
    peakEntry = "List";
    switch peakEntry
        case "List"
            spdPkNm = str2num(""); %let them list out the peaks
            if isempty(spdPkNm)
                error("Your SPD Peak List Is Empty Or Incorrect")
            end
        case "Interval"
            intervalMin_spdPkNm = 450;
            intervalInt_spdPkNm = 20;
            intervalMax_spdPkNm = 650;

            spdPkNm = intervalMin_spdPkNm : intervalInt_spdPkNm : intervalMax_spdPkNm;
            if isempty(spdPkNm)
                error("Your SPD Peak Interval Is Incorrect")
            end
        end
    end
    disp("Your SPD peak wavelengths are")
    disp(spdPkNm)

    nChannels = length(spdPkNm);
    disp("You have " + string(nChannels) + " Independent SPDs")

    fwhmEntry = "Equal";

    switch fwhmEntry
        case "List"
            spdFwhms = str2num("");
            if isempty(spdFwhms)
                error("Input a list of fwhms or change your entry type")
            end
        case "Equal"
            equal_spdFwhms = str2num("30");
            if isempty(equal_spdFwhms)
                error("Input a number for fwhms or change entry type")
            end
            spdFwhms = equal_spdFwhms*ones(nChannels,1);
        end
    end
end
```

```
end
%end of synthetic code block
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SCALE SCALE SCALE INITIAL VALUES spds, whichever you are
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%using%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if spdSource == "SYNTHETIC"
    switch initialGuessMode
        case "All_50_Percent"
            synthSpdChannels = makeSpds(spdPkJm, spdFwhms, ones(size(spdFwhms)), wl);
            [~,XYZChBeforeScale] = spdsToXyXYZ(synthSpdChannels, A_CMFS_Eval) %XYZ for sum of s

        case "xyTarget"
            synthSpdChannels = makeSpds(spdPkJm, spdFwhms, ones(size(spdFwhms)), wl);

            [~,XYZChBeforeScale] = spdsToXyXYZ(synthSpdChannels, A_CMFS_Eval) %XYZ for sum of s
            scaleAmount = 100/( sum(XYZChBeforeScale(:,2) )); %how much to scale s.t t
            synthSpdChannels = scaleAmount.*synthSpdChannels; % Now they should sum to 100

    end

    [~,XYZCh] = spdsToXyXYZ(synthSpdChannels, A_CMFS_Eval) %for debug

    figure
    plot(wl, synthSpdChannels); xlabel('Wavelength(nm)'); ylabel('Radiant Watt/nm'); title('SP

elseif spdSource == "USERSPDS"
    disp("You chose to load from UserSPDs.xlsx channels")
    disp("These are assumed to be 100 percent powered")
    if isempty(userChannels)
        disp("You chose to load all channels, because you didn't enter a list above")
        userSPDs = userSPDs;
    else
        userSPDs = userSPDs(:,userChannels);
        disp("Loading The following channels: ");disp(string(userChannels))
    end
    nChannels = width(userSPDs);
    userSPDs = userSPDs./100; %make them as a per percent now.

%
%    [~,XYZChBeforeScale] = spdsToXyXYZ(userSPDs, A_CMFS_Eval); %XYZ for sum of spds with peak
%    [~,XYZCh] = spdsToXyXYZ(userSPDs, A_CMFS_Eval); %for debug

    disp("Plotting userSPDs")
    figure
    plot(wl, userSPDs); xlabel('Wavelength(nm)'); ylabel('Radiant Watt/nm');
    text( 400, 1.3*mean(max(userSPDs)), string(sum(XYZCh(:,2),2) ));
    text( 500, 1.2*mean(max(userSPDs)), "Y For Each Loaded Channel at 100 Percent" );
    text( 500, 0.8*mean(max(userSPDs)), "Of what was loaded" );
elseif spdSource == "LIBRARY"
    error("Library mode not yet implemented. Please Paste into userSPDs.xlsx")
else
    error("Error while setting initial mix percent. Source type doesn't exist")
end
```

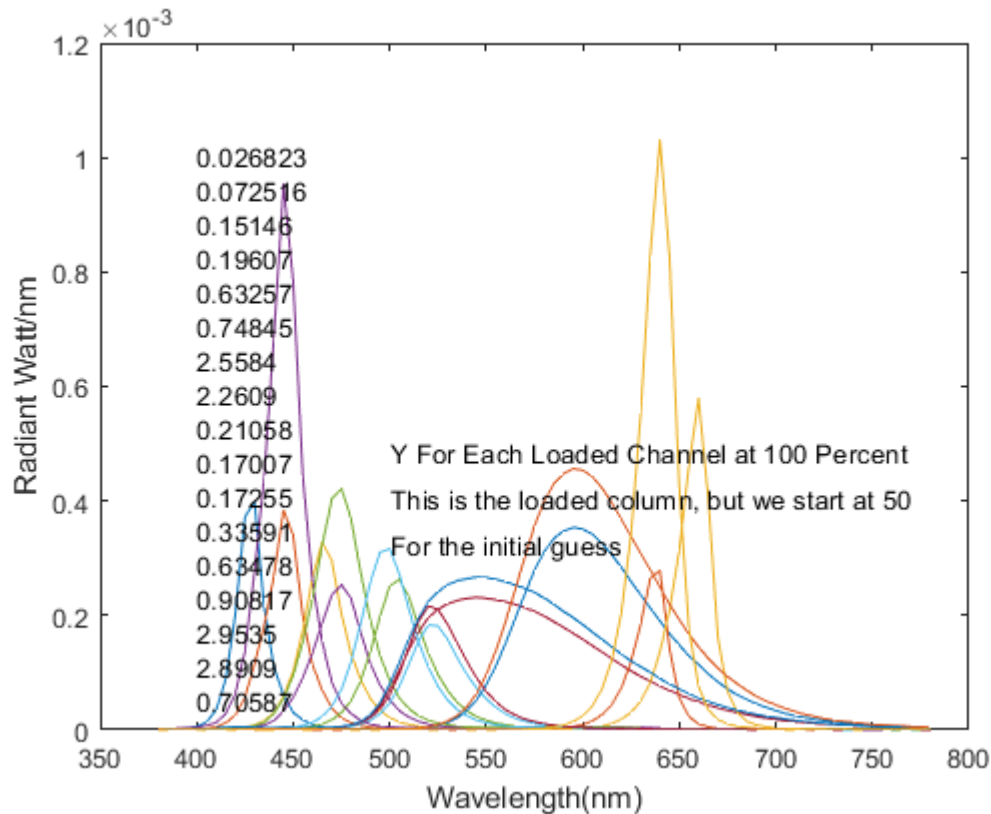
end

You chose to load from UserSPDs.xlsx channels

These are assumed to be 100 percent powered

You chose to load all channels, because you didn't enter a list above

Plotting userSPDs



```
%rename spdChannels to whatever source you used
```

```
switch spdSource
```

```
case "SYNTHETIC"
```

```
    spdChannels = synthSpdChannels;
```

```
case "USERSPDS"
```

```
    spdChannels = userSPDs;
```

```
end
```

```
%spdChannels is what is modified later in code. rename based on choice and  
%continue on
```

## Setup Initial Guess Ratios

Get chromaticity and Tristim for each channel.

```
[xy_Eval_Channels, XYZ_Eval_Channels] = spdsToXyXYZ(spdChannels, A_CMFS_Eval);
```

```

% %%%%%%%%% Find spdPercent_0 %%%%%%%%%
% %%%%%%%%%
% %%%%%%%%%
switch initialGuessMode
    case "xyTarget"
        % If you want an xyY
        spdPercents_0 = lsqnonneg(XYZ_Eval_Channels',XYZTarg'); %these ratios give us a fit
        spdMix_0 = spdChannels*spdPercents_0; %initial guess based on the percents
    case "All_50_Percent"
        %need to multiply by 50%. they are in per percent mode if it was
        %user

        spdPercents_0 = 50.*ones([nChannels, 1]); %set equal to 50%
        spdMix_0 = spdChannels*spdPercents_0; %initial guess based on the percents
    otherwise
        error("Error in establishing spdPercent0 with chosen initialGuessMode. The mode does not exist")
end
disp("Your spd channel percentages are")

```

Your spd channel percentages are

```
disp(spdPercents_0')
```

Columns 1 through 16

5000      5000      5000      5000      5000      5000      5000      5000      5000

Column 17

5000

```

currentSpdFig = figure; %name the current mix so I can pass handle to optimizer
title('Scaled SPDs For Optimization');
xlabel('Wavelength (nm)');
ylabel('Radiant Watt/nm');
plot(wl, [spdMix_0, spdPercents_0'.*spdChannels]);

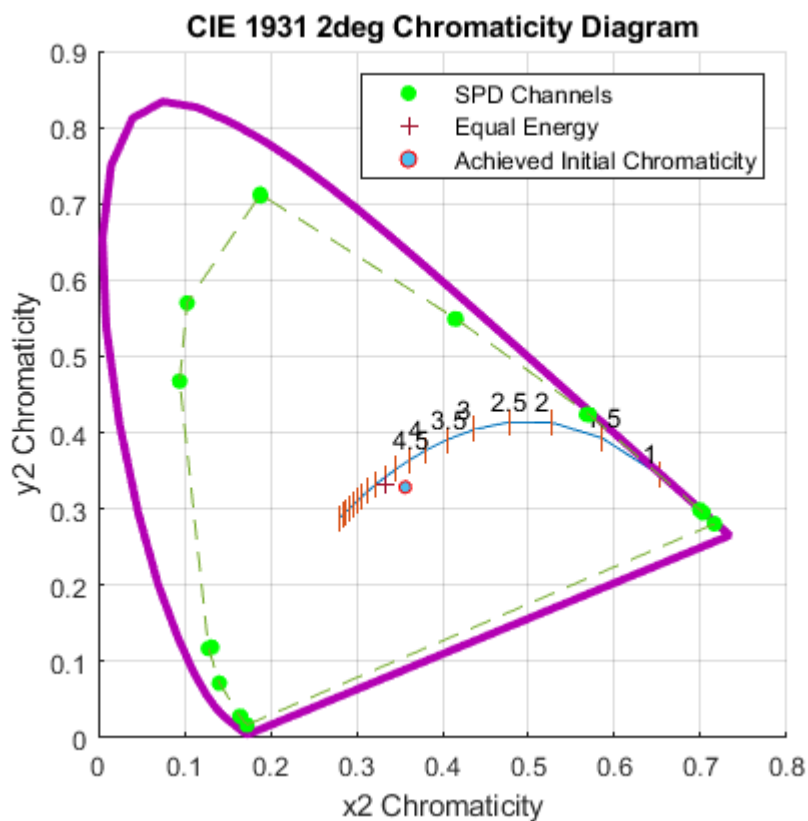
```





```
% %%%%%%%%% Put the SPD Channel chromaticities, their gamut, %%%%%%%%%%
% %%%%%%%%%the desired guess, and the actual mix %%%%%%%%%%
% %%%%%%%%%(should be same as the desired guess) %%%%%%%%%%
```

```
chromDiagram; %just make sure the figure is the same one
xySpdsPlot      = scatter(xy_Eval_Channels(:,1), xy_Eval_Channels(:,2), 'filled', 'g');
xyGamut         = plot( xy_Eval_Channels(convhull(xy_Eval_Channels),1),...
                        xy_Eval_Channels(convhull(xy_Eval_Channels),2) , '--'); %make a contour
xyMix_0         = scatter( xy_Eval_Mix_0(1), xy_Eval_Mix_0(2), 'filled', 'SizeData', 20, 'MarkerEd
xyEqualEnergy   = scatter(0.333,0.333, '+', 'sizedata', 30);
axis square; grid on;
legend([xySpdsPlot, xyEqualEnergy, xyMix_0],...
      {'SPD Channels', 'Equal Energy',...
       'Achieved Initial Chromaticity'});
```



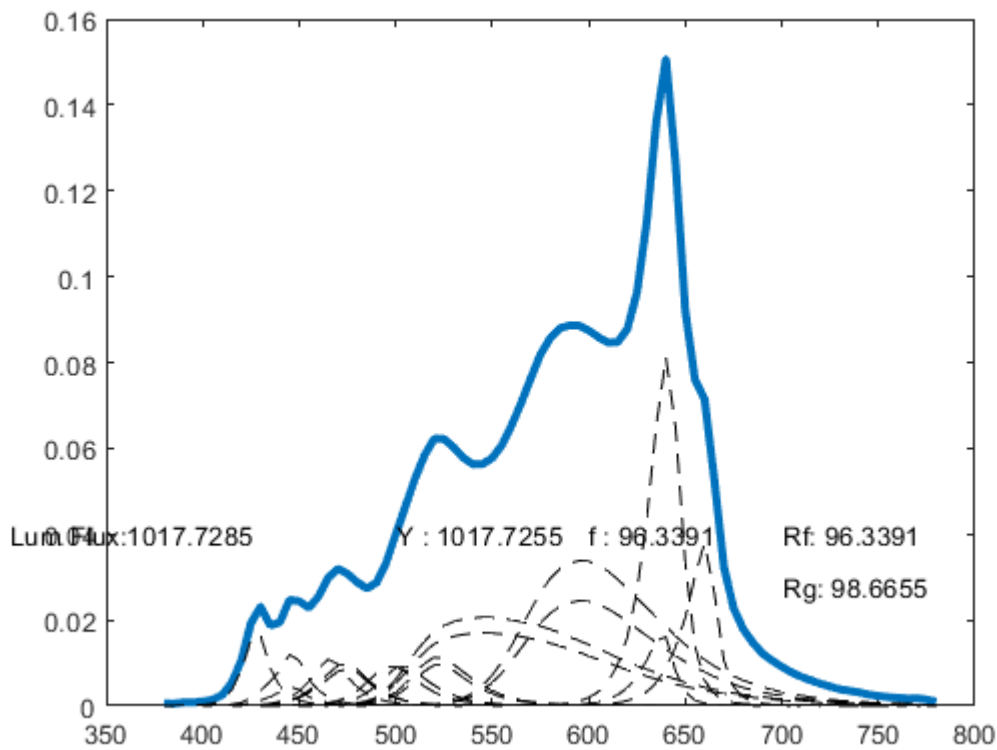
```
% %%%%%%%%% Data to Compare %%%%%%%%%%
% %%%%%%%%%%
spdCompare.x     = [];
spdCompare.y     = [];
spdCompare.sOpic = [];
spdCompare.mOpic = [];
spdCompare.lOpic = [];
spdCompare.melOpic = [];
```

```
spdCompare.rOpic = [];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
dataToPass = {currentSpdFig, chromDiagram, w1, XYZ_Eval_Mix_0, spdCompare, A_CMFS_Eval};
```

## Optimizer Notes

```
% Pass fixed parameters to objfun  
objfun = @(spdPercents)myObjFun(spdPercents,spdChannels,dataToPass);  
  
% Pass fixed parameters to confun  
confun = @(spdPercents)myFunConstraint(spdPercents,spdChannels,dataToPass);  
  
% Set nondefault solver options  
options = optimoptions('fmincon','PlotFcns',{'optimplotx','optimplotfval',...  
    'optimplotconstrviolation','optimplotstepsize'});  
  
% Solve  
[solution,objectiveValue] = fmincon(objfun,spdPercents_0,[],[],[],[],...  
    zeros(size(spdPercents_0)),[],confun,options);
```



fmincon stopped by the output or plot function.

```
% Clear variables
clearvars objfun confun options
```

```
function [f] = myObjFun(spdPercents, spdChannels, dataToPass) % pg 1-39 of their optimization o
    %% Main Ingredient
    spdMix(:,1) = spdChannels*spdPercents;
    %% Metrics From the SPD
    % TM30-18
    [TM30_Struct]= ANSI_IES_TM_30_20_Coulter_Edit_V3(spdMix);
    %CIE S026:E2018 Alpha Opic LER
    [opic_Struct,luminousFlux] = spdToAlphaOpics(spdMix);
    %% Other
    radWatts = trapz(dataToPass{3},spdMix);
    % ler = ;%
    A_CMFS_Eval = dataToPass{6};
    [xyEval_0,XYZEval] = spdsToXyXYZ(spdMix, A_CMFS_Eval);

    %Describe Obj Function. It minimizes, so if you want to maximize A,
    %then use f = -A
```

```

%      f = -1* mel/XYZ2XYZ10_mix(2); %max melanopic to Y2
%      fMinimize = opic_Struct.melOpic/opic_Struct.rOpic;
%      fMaximize = -fMinimize;
%      f = fMaximize;
%      f = -opic_Struct.melOpic;
f = -TM30_Struct.rf;

%      %% Don't Look%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(dataToPass{1}.Number); %currentSpdFig
plot( dataToPass{3},spdMix,'-', 'linewidth',3);
hold on
plot(dataToPass{3}, spdChannels.*spdPercents','--k');
text(500, mean(spdMix), "Y : " +string(XYZEval(2) ));
text(600, mean(spdMix), "f : " + string(-f) );
text( 700, mean(spdMix), "Rf: " + string(TM30_Struct.rf) );
text( 700, 0.7*mean(spdMix), "Rg: " + string(TM30_Struct.rg) );
text(300, mean(spdMix), "Lum Flux:" + string(luminousFlux));
hold off

%
%      figure(dataToPass{2}.Number) %chrom dia
%      axList = get(gca, 'Children');
%      axList(length(axList)+1) = scatter(xy2xy10_mix(1), xy2xy10_mix(2) );
%      textVal = Rf_mix;
%      legend( axList(end) , { strcat("Rf: ", string(Rf_mix), ",MLE: ",string(-f) ) });
%      axList(end) = [];
%      xlim([0.27 0.54]);
%      ylim([0.27 0.54]);
%      axis square;
end

```

```

function [c,ceq] = myFunConstraint(spdPercents, spdChannels,dataToPass) % pg 1-39 of their opti
%%Main Ingredient
spdMix = spdChannels*spdPercents;

% TM30-18
[TM30_Struct]= ANSI_IES_TM_30_20_Coulter_Edit_V3(spdMix);
%CIE S026:E2018 Alpha Opic LER
[opic_Struct] = spdToAlphaOpics(spdMix);

A_CMFS_Eval = dataToPass{6};
[xyEval,XYZEval] = spdsToXyXYZ(spdMix, A_CMFS_Eval);

%Other
radWatts = trapz(dataToPass{3},spdMix);
%      ler =

```

```

c    = []; ceq = [];

%% inequalities
% A < B --->  c = [c ; 1*( A - B )];
% A > B --->  c = [c ; -1*( A - B )];
%   c = [c ; -1*(TM30_Struct.rf - 85)];
%   c = [c ; -1*(Duv_mix      - 0.08*2)];
%
%   c = [c; -(TM30_Struct.rg - 95)];
c = [c; spdPercents - 100*ones(size(spdPercents) )];
c = [c; -(spdPercents - zeros(size(spdPercents) ))];

%%Equalities%
% A = B --->  ceq = [ceq ; A - B ];
% A = 0 --->  ceq = [ceq ; A      ];

ceq = [ceq; (TM30_Struct.cct - 3000)];

end

```

Evaluate

Unsure what to do but I have to uncomment (drag select and press "Ctrl + T" to remove comment and "Ctrl + R" to comment