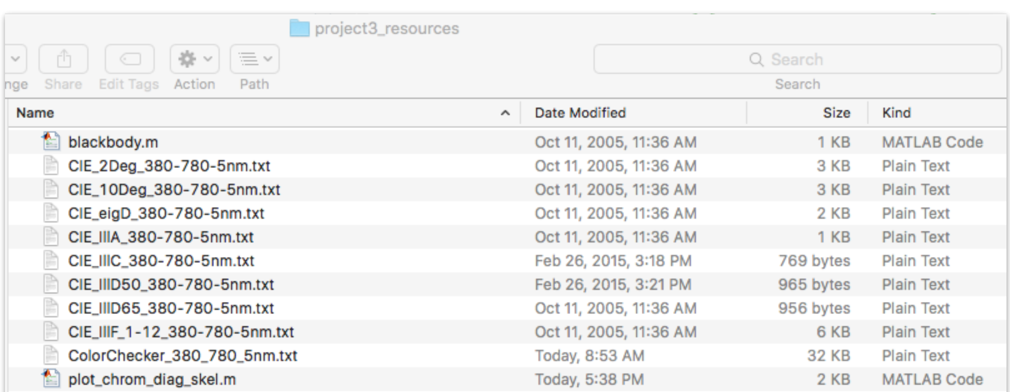
Project 3: Colorimetry

In this project you will develop functions to load a structure of CIE observer and illuminant data, calculate CIE XYZ tristimulus values from reflectance spectra, and calculate x,y chromaticity coordinates from the XYZ values. You will test your functions by calculating the tristimulus values and chromaticity coordinates of the patches in the ColorChecker chart. You will then use your functions to calculate the tristimulus values and chromaticity coordinates of your real, imaged, and matching patches from the spectral measurements you made using the ColorMunki in Project 2, and compare the ColorMunki measured values with your calculated values.

1) Download and unpack the **project3_resources.zip** file provided in myCourses to your working directory.



2) Create a function **cie = loadCIEdata** that returns a structure of CIE observer and illuminant data by performing the following steps.

a. Create a new function in the editor with the header **function [cie] = loadCIEdata**

b. Inside the function define a structure with the following fields (lambda, cmf2deg, cmf10deg, illA, illC, illD50, illD65, illE, illF, eigD).

c. Fill the structure by reading in the text files listed below and loading the data into the appropriate fields of the structure.

```
CIE_2Deg_380-780-5nm.txt
CIE_10Deg_380-780-5nm.txt
CIE_eigD_380-780-5nm.txt
CIE_IllA_380-780-5nm.txt
CIE_IllC_380-780-5nm.txt
CIE_IllD50_380-780-5nm.txt
CIE_IllD65_380-780-5nm.txt
CIE_IllF_1-12_380-780-5nm.txt
ColorChecker_380_780_5nm.txt
```

First store the wavelength information from the file CIE_2Deg_380-780-5nm.txt as a 1x81 row vector in the "lambda" field. Store the remaining color matching function data as an 81x3 column array in the "cmf2deg" field. Then remove the wavelength information from each of the remaining datasets by deleting the first column of each loaded matrix and store the remaining data in the corresponding field. (Note that that CIE_IllF_1-12_380-780-5nm.txt contains 12 illuminants)

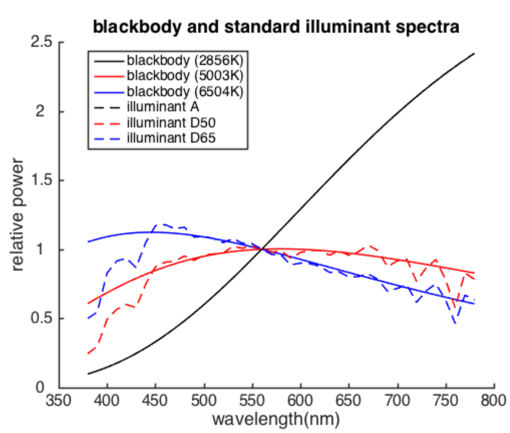
d. Include the CIE equal energy illuminant E in your structure by creating a vector in the "illE" field with a length equal to the wavelength range and a constant value of 100.

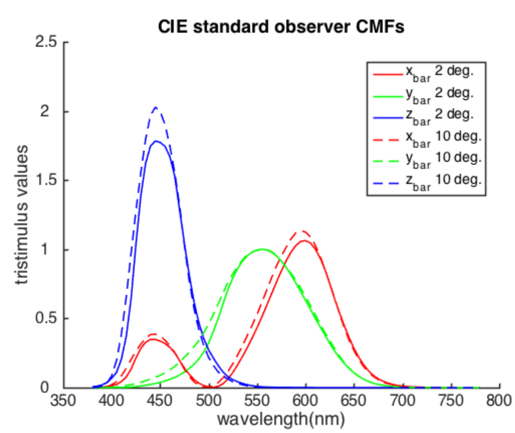
e. When called as below, your function should produce the following output:

```
>> cie = loadCIEdata

cie =

      lambda: [1x81 double]
     cmf2deg: [81x3 double]
    cmf10deg: [81x3 double]
        illA: [81x1 double]
        illC: [81x1 double]
      illD50: [81x1 double]
      illD65: [81x1 double]
        illE: [81x1 double]
        illF: [81x12 double]
        eigD: [81x3 double]
```

f. Confirm that the returned structure contains the correct data in the correct formats.

3) Test your loadCIEdata function by recreating the graphs shown below (including all text and styles). The function **blackbody.m** has been provided to produce the spectral data for plotting the blackbody curves.





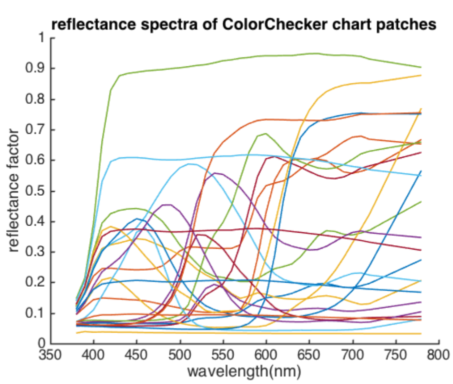
4) The equations for calculating XYZ tristimulus values from surface reflectance R(λ), color matching function $\bar{x}(\lambda)$,$\bar{y}(\lambda)$,$\bar{z}(\lambda)$, and illumination S(λ) data are shown below. Create a function **XYZ = ref2XYZ(refs, cmfs, illum)** that takes as input: 'refs' an (nx1) vector of reflectance factor data; 'cmfs' an (nx3) set of CIE color matching functions; and 'illum' an (nx1) spectral power distribution of a light source, and returns 'XYZ' a (3x1) vector of CIE XYZ tristimulus values.

$$X = k\sum_{\lambda}\bar{x}(\lambda)S(\lambda)R(\lambda)\Delta\lambda,$$

$$Y = k\sum_{\lambda}\bar{y}(\lambda)S(\lambda)R(\lambda)\Delta\lambda, \qquad k = \frac{100}{\sum_{\lambda}\bar{y}(\lambda)S(\lambda)\Delta\lambda.}$$

$$Z = k\sum_{\lambda}\bar{z}(\lambda)S(\lambda)R(\lambda)\Delta\lambda$$

5) Test your ref2XYZ function by calculating the XYZ values for the patches in the ColorChecker chart. The spectral data is provided in the file **ColorChecker_380_780_5nm.txt** (see plot). Use the CMFs for the 2 degree observer and illuminant D65 in your calculations. The Matlab code for the test is given below. Confirm that your values match the values shown to rounding error.




```
CC_spectra = importdata('ColorChecker_380_780_5nm.txt');
CC_XYZs = ref2XYZ(CC_spectra(:,2:25),cie.cmf2deg,cie.illD65)

CC_XYZs =

  Columns 1 through 7

   11.5145   39.1346   18.3488   11.1492   25.8437   31.7110   37.1457
   10.3819   36.5981   19.6332   13.8551   24.3868   43.8600   29.5592
    7.1502   27.0564   35.6470    7.4267   45.6142   44.8778    6.5006

  Columns 8 through 14

   13.8627   29.1328    8.5889   33.9174   46.1864    8.9183   15.0353
   12.3179   19.8475    6.4569   44.1533   42.4957    6.4177   24.1079
   39.3093   14.9941   15.4745   11.4297    8.6771   32.2736    9.6379

  Columns 15 through 21

   19.3447   55.8457   29.6768   14.4138   87.8402   57.9621   35.2286
   11.3576   58.9726   19.3515   19.9750   92.3781   61.0426   37.0414
    5.5526    9.6411   32.2626   39.0008   95.6125   65.4909   40.2256

  Columns 22 through 24

   19.3492    8.7646    3.2111
   20.4708    9.2915    3.3763
   22.1545   10.3188    3.9312
```

6) The equations for calculating x,y chromaticity coordinates from XYZ tristimulus values are shown below. Create a function **xyY = XYZ2xyY(XYZ)** that takes as input 'XYZ' a (3xn) vector of XYZ tristimulus values and returns 'xyY' a (3xn) vector of chromaticity coordinates (x,y) and luminance factor (Y).

$$x = \frac{X}{X+Y+Z}$$
$$y = \frac{Y}{X+Y+Z}$$

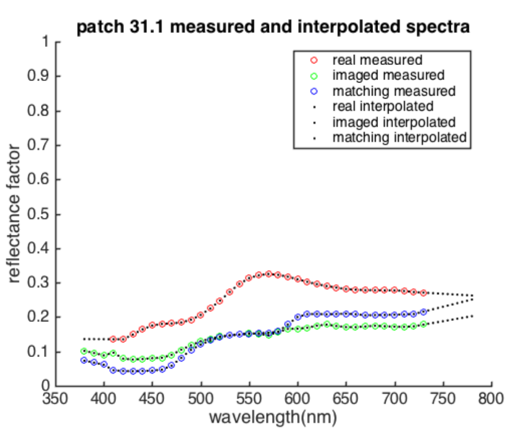
7) Test your **XYZ2xyY** function by calculating the xyY values for the ColorChecker chart from the XYZ values you calculated in step 5. The Matlab code for the test is given below. Confirm that your values match the values shown to rounding error.

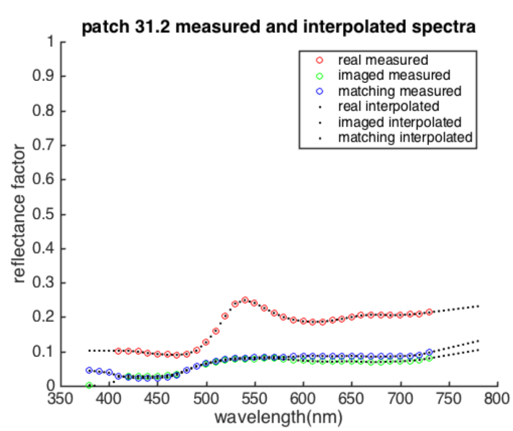
```
CC_xyYs = XYZ2xyY(CC_XYZs)

CC_xyYs =

  Columns 1 through 7

    0.3964    0.3807    0.2492    0.3438    0.2696    0.2633    0.5074
    0.3574    0.3561    0.2667    0.4272    0.2544    0.3641    0.4038
   10.3819   36.5981   19.6332   13.8551   24.3868   43.8600   29.5592

  Columns 8 through 14

    0.2117    0.4554    0.2814    0.3790    0.4744    0.1873    0.3082
    0.1881    0.3102    0.2116    0.4933    0.4365    0.1348    0.4942
   12.3179   19.8475    6.4569   44.1533   42.4957    6.4177   24.1079

  Columns 15 through 21

    0.5336    0.4487    0.3651    0.1964    0.3185    0.3142    0.3132
    0.3133    0.4738    0.2381    0.2722    0.3349    0.3309    0.3293
   11.3576   58.9726   19.3515   19.9750   92.3781   61.0426   37.0414

  Columns 22 through 24

    0.3122    0.3089    0.3053
    0.3303    0.3275    0.3210
   20.4708    9.2915    3.3763
```

8) Now use the functions you've just created to calculate the tristimulus values and chromaticity coordinates of your real, imaged, and matching color patches from the spectral data you measured in Project 2. To do this first adapt the script below to define some parameters and load the measured spectral data.

```
% load the CIE observer and illuminant data

% define ColorMunki/Argyll/spotread measurement wavelengths for the
% normal and transmissive (-t) modes
cm_lams_norm = 410:10:730;
cm_lams_trans = 380:10:730;

% define header offsets for reading the .sp files
cm_h_offset_norm = 54;
cm_h_offset_trans = 57;

% load and normalize the measured spectral data for the patch #1
data = importdata('31.1_real.sp', ' ', cm_h_offset_norm);
real_311 = data.data/100;

data = importdata('31.1_imaged.sp', ' ', cm_h_offset_trans);
imaged_311 = data.data/100;

data = importdata('31.1_matching.sp', ' ', cm_h_offset_trans);
matching_311 = data.data/100;

% repeat the section above for patch #2
```

9) Now extend the script to interpolate your data to the correct sampling interval and range for your ref2XYZ function. Your ref2XYZ function calculates XYZ values from spectra defined at 5nm intervals over a 380-780nm range. The data provided by the ColorMunki is defined at 10nm intervals over a 410-730nm or 380-730nm range. To use ref2XYZ you need to interpolate/extrapolate the data using the "interp1" function in Matlab. Use the correct "cm_lams…" for the input range, the patch spectral data as the input values, cie.lambda(:) as the output range, 'linear' as the 'method', and 'extrap' as an option. Create graphs like the ones shown below that visualize and confirm the process.





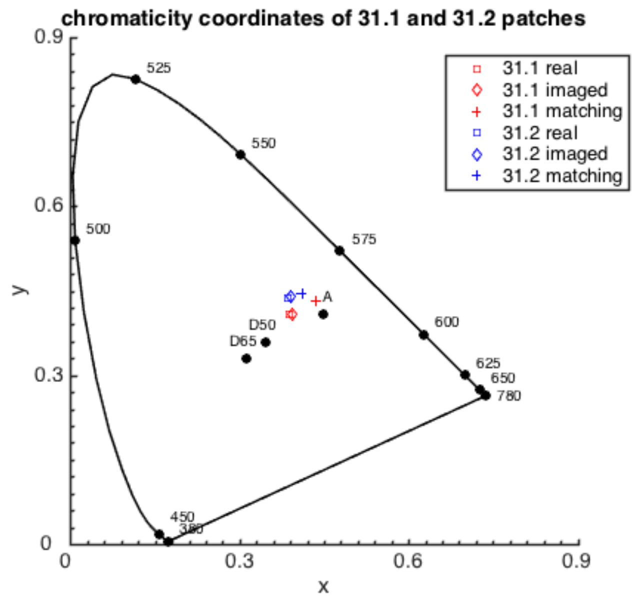
10) Now use your ref2XYZ function to calculate XYZ tristimulus values for each of your patches from the interpolated patch spectra. Use the CIE 2^o standard observer CMFs and the D50 illuminant as input parameters to the function. Create a clearly formatted table like the one shown below that summarizes and compares the XYZ values measured directly by the ColorMunki and the ones you calculated using your ref2XYZ function (hint: I pasted my data into Excel for the formatting and printed the table as a pdf). Confirm that the measured and calculated values are nominally the same.

tristimulus values	patch (#).1						patch (#).2					
	measured			calculated			measured			calculated		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
real	27.6317	28.9911	14.4655	27.6207	28.978	14.4682	17.9585	20.3373	8.1219	17.9607	20.322	8.1352
imaged	14.6797	15.1442	7.35259	14.6796	15.142	7.36597	6.5926	7.39427	2.8582	6.59273	7.3908	2.86388
matched	15.933	15.8489	4.81373	15.9317	15.846	4.83058	7.42336	8.01709	2.5965	7.42389	8.0139	2.60672

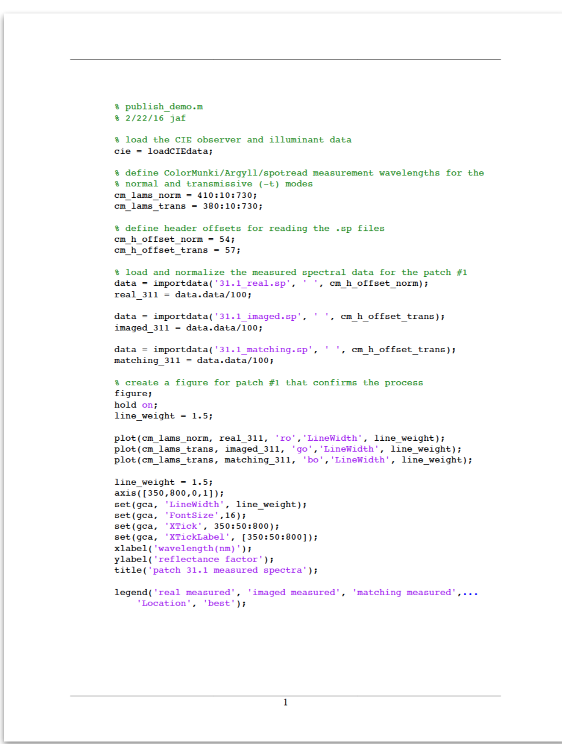
11) Now use your XYZ2xyY function to calculate x,y chromaticity coordinates from both the measured and calculated XYZ values you tabulated in the previous step. Create a clearly formatted table like the one shown below that compares the measured and calculated x,y,Y values. Confirm that the measured and calculated values are nominally the same

chromaticity coordinates	patch (#).1						patch (#).2					
	measured			calculated			measured			calculated		
	x	y	Y	x	y	Y	x	y	Y	x	y	Y
real	0.3887	0.40782	28.9911	0.38866	0.4078	28.9776	0.38689	0.43814	20.337	0.38694	0.4378	20.3217
imaged	0.39487	0.40736	15.1442	0.39475	0.4072	15.1416	0.39137	0.43896	7.3943	0.39132	0.4387	7.39079
matched	0.43538	0.43308	15.8489	0.43519	0.4329	15.8463	0.41156	0.44448	8.0171	0.41142	0.4441	8.01393

12) Visualize the measured and calculated patch chromaticities by plotting their values on a clearly formatted chromaticity diagram as like the one shown below. The function **plot_chrom_diag_skel.m** has been provided to plot the skeleton of the diagram. Write a script to add your data and the legend.



13) Use the "publish" menu/function in Matlab to produce a PDF file that contains all the (commented) code, results, and figures you generated in steps 2-9 and 12. Add an additional page to the PDF that shows the two tables you generated in steps 10 and 11. Submit this PDF as "teamX_project3_report.pdf" to myCourses.



```
% publish_demo.m
% 2/22/16 jaf

% load the CIE observer and illuminant data
cie = loadCIEdata;

% define ColorMunki/Argyll/spotread measurement wavelengths for the
% normal and transmissive (-t) modes
cm_lams_norm = 410:10:730;
cm_lams_trans = 380:10:730;

% define header offsets for reading the .sp files
cm_h_offset_norm = 54;
cm_h_offset_trans = 57;

% load and normalize the measured spectral data for the patch #1
data = importdata('31.1_real.sp', ' ', cm_h_offset_norm);
real_311 = data.data/100;

data = importdata('31.1_imaged.sp', ' ', cm_h_offset_trans);
imaged_311 = data.data/100;

data = importdata('31.1_matching.sp', ' ', cm_h_offset_trans);
matching_311 = data.data/100;

% create a figure for patch #1 that confirms the process
figure;
hold on;
line_weight = 1.5;

plot(cm_lams_norm, real_311, 'ro','LineWidth', line_weight);
plot(cm_lams_trans, imaged_311, 'go','LineWidth', line_weight);
plot(cm_lams_trans, matching_311, 'bo','LineWidth', line_weight);

line_weight = 1.5;
axis([350,800,0,1]);
set(gca, 'LineWidth', line_weight);
set(gca, 'FontSize',16);
set(gca, 'XTick', 350:50:800);
set(gca, 'XTickLabel', [350:50:800]);
xlabel('wavelength(nm)');
ylabel('reflectance factor');
title('patch 31.1 measured spectra');

legend('real measured', 'imaged measured', 'matching measured',...
    'Location', 'best');
```

1