Homework1

January 28, 2022

1 Astro 404 Assignment #1 Kennedy Robinson

1.1 Calculate bolometric corrections for the U, B and V bands in the Johnson-Cousins photometric system for 3 to 4 main-sequence (MS, luminosity class V) stars of different spectral classes using their Kurucz theoretical spectra

Take the Kurucz theoretical spectra from http://svo2.cab.inta-csic.es/theory/newov/index.php, the UBVRI response functions from http://www.aip.de/en/research/facilities/stella/instruments/data/johnson-ubvri-filter-curves, and parameters ($M/M_\odot$, $R/R_\odot$ and $\log_{10}(L/L_\odot)$) of MS stars (V dwarfs) from internet, e.g. from http://www.pas.rochester.edu/~emamajek/EEM_dwarf_UBVIJHK_colors_Teff.txt. Assume that [Fe/H]=0 for all MS stars, except for Vega that has [Fe/H]=−0.5, $T_{\rm eff} = 9550$ K, $\log_{10} g = 3.95$ (cm/s^2) and $(R/d)^2 = 6.247 \times 10^{-17}$.

[89]:
```
import numpy as np
import matplotlib.pyplot as plt

# physical constants
import scipy.constants as sc

# astronomical constants
from astropy import constants as ac

from scipy.integrate import simps
from scipy import interpolate

fsize=16
plt.rcParams['font.size'] = 16
plt.rcParams['font.family'] = 'serif'

# the following commands allow to produce a nice pdf version of the notebook␣
↪with figures
%matplotlib inline

from IPython.display import set_matplotlib_formats
set_matplotlib_formats('png', 'pdf')
```

```
/tmp/ipykernel_80/754875480.py:21: DeprecationWarning: `set_matplotlib_formats`
is deprecated since IPython 7.23, directly use
`matplotlib_inline.backend_inline.set_matplotlib_formats()`
  set_matplotlib_formats('png', 'pdf')
```

[90]:
```
# minimum and maximum wave lengths in Angstroms used in plots
wl_min = 3000.
wl_max = 10000.
```

A bolometric correction in a photometric band A is

$$BC_{\rm A} = 4.75 - 2.5\log_{10}\left[\frac{4\pi(10{\rm pc})^2\sigma T_{\rm eff}^4}{L_\odot}\right] + 2.5\log_{10}\left[\frac{\int_{\lambda_1}^{\lambda_2} F_\lambda S_\lambda d\lambda}{\int_{\lambda_1}^{\lambda_2} f_\lambda^0 S_\lambda d\lambda}\right] - m_{\rm A}^0,$$

where S_λ is a photometric-band response function, F_λ is the intrinsic (at its surface) monochromatic flux from the star, f_λ^0 is the flux from a standard star (Vega) measured at the top of Earth's atmosphere and $m_{\rm A}^0$ is its apparent magnitude (zero for Vega).

Therefore, we have to first calculate $\int_{\lambda_1}^{\lambda_2} f_\lambda^0 S_\lambda d\lambda$ for Vega. A value of this integral will be assigned to res_0.

1.1.1 NOTE: delete all header and comment lines in your saved text files with spectra and response functions. The read operators below assume that these files contain only numerical data !!!

1.1.2 Also, pay attention to my naming of the spectra files and to the name of the directory in which they are saved. Change these, if you wish!

[91]:
```
# first, read in a Kurucz theoretical spectrum of a star that is as similar to␣
↪Vega
# as possible from http://svo2.cab.inta-csic.es/theory/newov/index.php

star = 'Vega' # almost, with the closest parameters
Teff = 9700 # 9500 # K

# parameters of the star (Vega is an A0V star),
# e.g. from http://www.pas.rochester.edu/~emamajek/
↪EEM_dwarf_UBVIJHK_colors_Teff.txt
M = 2.3
R = 2.09
lgM = np.log10(M)
lgR = np.log10(R)
lgL = 1.54

#print ac.M_sun, ac.R_sun, ac.L_sun
g = sc.G*(10.**lgM*ac.M_sun)/(10.**lgR*ac.R_sun)**2
```

```python
# Note that in expressions one should use g.value, ac.L_sun.value, etc.

print ("g =", g, "and its value in SI is",g.value)
lgg = np.log10(1e2*g.value) # 1e2 transforms m/s**2 to cm/s**2, as used in␣
↪model atmospheres
print ("lgg =", lgg, "(cm/s**2)") # for MS stars lgg should be around 4.
Teff = ((ac.L_sun.value/ac.R_sun.value**2)*10.**lgL/(4*sc.pi*(10.**lgR)**2*sc.
↪sigma))**0.25
print ("Teff =", Teff, "(K)")

lgg = 4.0 # cm/s**2
# [Fe/H] = -0.5
Rd2 = 6.247e-17 # the square of the radius to distance ratio for Vega

mA0 = 0. # for Vega in the Johnson-Cousins system

# these are the names of the directory and file with theoretical stellar spectra
file_theory = open('./data/spectra/'+star+'theoryspectrum.txt',"r")

wl_theory = []
fl_theory = []

for line in file_theory:
    data = line.split()
    wl_theory.append(float(data[0]))
    fl_theory.append(float(data[1]))

file_theory.close()

# normalize the flux
fl_max = max(fl_theory)  # this parameter will be used later

n_theory = len(wl_theory)
for i in range(n_theory):
    fl_theory[i] = fl_theory[i]/fl_max
```

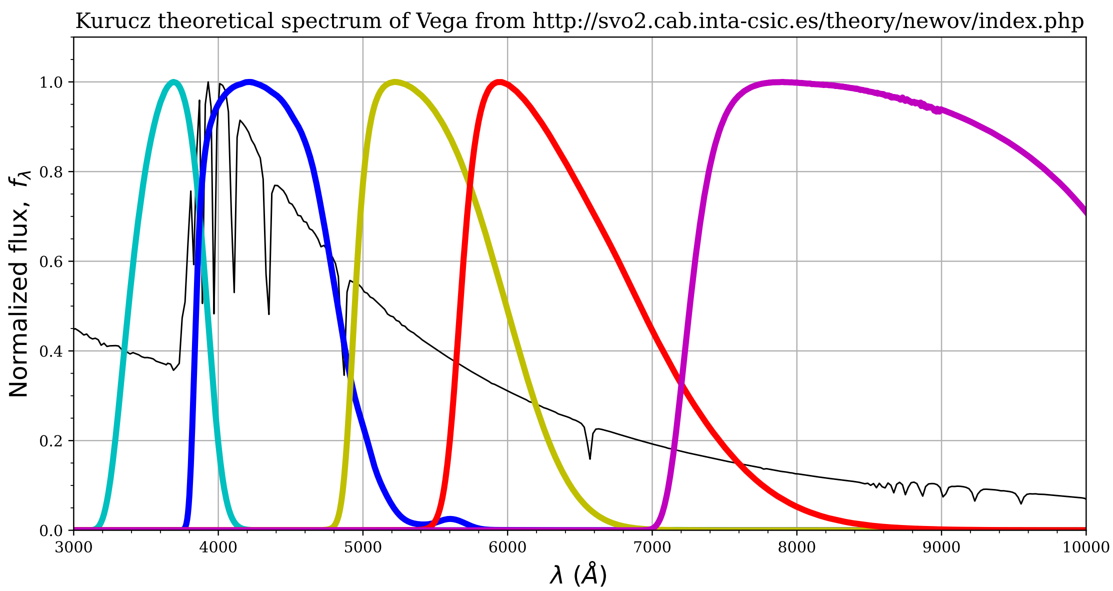
```
g = 144.37862155444202 kg / m2 and its value in SI is 144.37862155444202
lgg = 4.159502891098618 (cm/s**2)
Teff = 9688.430704873494 (K)
```

1.1.3 now, read in the UBVRI response functions that can be downloaded from http://www.aip.de/en/research/facilities/stella/instruments/data/johnson-ubvri-filter-curves

[92]:
```
# now, read in the UBVRI response functions that can be downloaded
# from http://www.aip.de/en/research/facilities/stella/instruments/data/
↪johnson-ubvri-filter-curves

band = ['U', 'B', 'V', 'R', 'I']
col = ['c', 'b', 'y', 'r', 'm']
n_band = len(band)

wl_band = []
sl_band = []

for k in range(n_band):

    wl_band.append([])
    sl_band.append([])
    # these are the names of the directory and file with the response functions␣
↪
    file_band = open('./data/spectra/Bessel_'+band[k]+'-1.txt',"r")


    for line in file_band:
        data = line.split()
        wl_band[k].append(10.*float(data[0])) # multiplied by 10 to transform␣
↪nm to A
        sl_band[k].append(float(data[1]))

    file_band.close()

    # reverse the band arrays, so that wl_band increases along with wl_theory
    wl_band[k][:] = wl_band[k][::-1]
    sl_band[k][:] = sl_band[k][::-1]

    # normalize response functions
    sl_max = max(sl_band[k][:])
    for i in range(len(sl_band[k][:])):
        sl_band[k][i] = sl_band[k][i]/sl_max
```

[93]:
```
# plot the spectrum and response functions
fig=plt.figure()
ax=fig.gca()
fig.set_size_inches(12,6)

plt.plot(wl_theory,fl_theory,'k-',linewidth=1)
```

```
for k in range(n_band):
    plt.plot(wl_band[k][:],sl_band[k][:],col[k],linewidth=4)

plt.xlim(wl_min,wl_max)
plt.ylim(0.,1.1)
plt.xlabel('$\lambda\ (\AA)$',fontsize=fsize)
plt.ylabel('$\mathrm{Normalized\ flux},\ f_\lambda$',fontsize=fsize)
plt.title('Kurucz theoretical spectrum of '+star+\
          ' from http://svo2.cab.inta-csic.es/theory/newov/index.
 ↪php',fontsize=14)
plt.grid()
plt.minorticks_on()
#plt.tight_layout()
plt.show()
```



[94]:
```
# select a photometric band for which the bolometric correction will be␣
 ↪calculated from
for kk in range(n_band):
    print ("k =", kk, "for band", band[kk])

k=2 # this is your selection
band_sel = band[k]
print ("The photometric band", band[k], "is selected")
```

```
k = 0 for band U
k = 1 for band B
k = 2 for band V
```

```
k = 3 for band R
k = 4 for band I
The photometric band V is selected
```

[95]:
```
# isolate a range of wavelengths in which the selected response function is
↪larger than sl_min
sl_min = 1e-4
wl_int = []
sl_int = []
for i in range(len(wl_band[k][:])):
    if sl_band[k][i] >= sl_min:
        wl_int.append(wl_band[k][i])
        sl_int.append(sl_band[k][i])
n_int = len(wl_int)
```

[96]:
```
# select a part of the theoretical spectrum that overlaps with the isolated
↪wavelength range
fl_int = []
x_int = []
wl_int_min = min(wl_int)
wl_int_max = max(wl_int)
for i in range(n_theory):
    if wl_theory[i] >= wl_int_min and wl_theory[i] <= wl_int_max:
        x_int.append(wl_theory[i])
        fl_int.append(fl_theory[i])

n_fl_int = len(x_int)
print (n_int, n_fl_int)
print ("lambda_1 =", wl_int[0], ", lambda_2 =", wl_int[-1])
```

```
523 131
lambda_1 = 4670.0 , lambda_2 = 7285.0
```

[97]:
```
# prepare to interpolate the response function at the points in which the
↪theoretical flux is provided
sl = interpolate.interp1d(wl_int,sl_int)

# this is the integrand that represents the convolution of the flux and
↪response function
f_int = np.linspace(0,0,n_fl_int)
for i in range(n_fl_int):
    f_int[i] = fl_int[i]*sl(x_int[i])

# Simpson's rule is used here for integration
# after the integration don't forget to multiply by the flux normalization
↪factor
# fl_max that was found earlier
```

```
res = fl_max*simps(f_int, x=x_int)
res_0 = res*Rd2 # this is the integral in the denominator in the above␣
 ↪expression for BC_A
```

[98]:
```
# calculate BC for Vega and compare it with the value from the internet table
BC = 4.75-2.5*np.log10(4*sc.pi*(10*ac.pc.value)**2*ac.sigma_sb.value*Teff**4/ac.
 ↪L_sun.value)+\
2.5*np.log10(res/res_0)-mA0

BC_Vega = BC

print ("For the star", star, "the bolometric correction in the band", band_sel,␣
 ↪"is", BC)
```

```
For the star Vega the bolometric correction in the band V is -0.2230900436675327
```

1.1.4 Select a few (3 to 4) MS stars from http://www.pas.rochester.edu/~emamajek/EEM_dwarf_ and use the cells below to calculate bolometric corrections for these stars in the bands U, B and V. Don't forget to repeat the above calculations for Vega when you change the photometric band.

The gravitational acceleration at the surface of the star is

$$g = \frac{GM}{R^2},$$

and its effective temperature can be found from the relation

$$L = 4\pi R^2\sigma T_{\rm eff}^4.$$

[99]:
```
# provide here parameters of the star
star = 'F5V' # its spectral class
print ("A", star, "star")

# parameters of the star,
# e.g. from http://www.pas.rochester.edu/~emamajek/
 ↪EEM_dwarf_UBVIJHK_colors_Teff.txt

M = 1.61
R = 1.728
Teff = 7220.
lgM = np.log10(M)
lgR = np.log10(R)
lgL = 0.86

#print ac.M_sun, ac.R_sun, ac.L_sun
g = sc.G*(10.**lgM*ac.M_sun)/(10.**lgR*ac.R_sun)**2
```

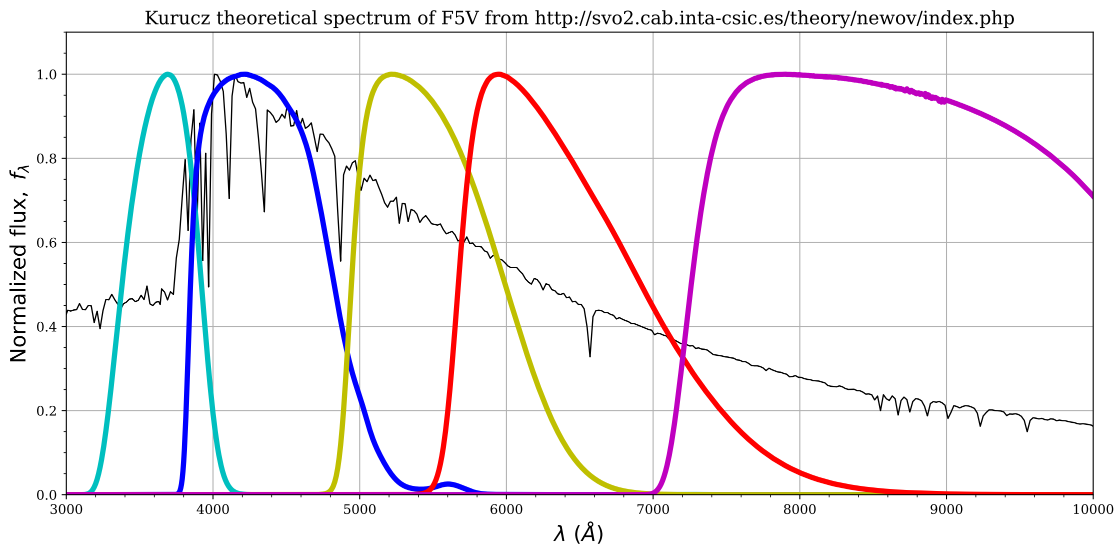
```
# Note that in expressions one should use g.value, ac.L_sun.value, etc.

print ("g =", g, "and its value in SI is",g.value)
lgg = np.log10(1e2*g.value) # 1e2 transforms m/s**2 to cm/s**2, as used in␣
↪model atmospheres
print ("lgg =", lgg, "(cm/s**2)") # for MS stars lgg should be around 4.
Teff = ((ac.L_sun.value/ac.R_sun.value**2)*10.**lgL/(4*sc.pi*(10.**lgR)**2*sc.
↪sigma))**0.25
print ("Teff =", Teff, "(K)")
```

```
A F5V star
g = 147.8447907853394 kg / m2 and its value in SI is 147.8447907853394
lgg = 4.169806027049234 (cm/s**2)
Teff = 7203.684957963438 (K)
```

[100]:
```
# read in a theoretical spectrum of the star
# take the Kurucz spectrum for the above specified parameters
# from http://svo2.cab.inta-csic.es/theory/newov/index.php


file_theory = open('../Spec/'+star+'Spectrum.txt',"r") #CHANGE TO MY FILE NAME

wl_theory = []
fl_theory = []

for line in file_theory:
    data = line.split()
    wl_theory.append(float(data[0]))
    fl_theory.append(float(data[1]))

file_theory.close()

# normalize the flux
fl_max = max(fl_theory)
n_theory = len(wl_theory)
for i in range(n_theory):
    fl_theory[i] = fl_theory[i]/fl_max
```

[101]:
```
# plot the spectrum and response functions
fig=plt.figure()
ax=fig.gca()
fig.set_size_inches(12,6)

plt.plot(wl_theory,fl_theory,'k-',linewidth=1)

for k in range(n_band):
```

```
    plt.plot(wl_band[k][:],sl_band[k][:],col[k],linewidth=4)

plt.xlim(wl_min,wl_max)
plt.ylim(0.,1.1)
plt.xlabel('$\lambda\ (\AA)$',fontsize=fsize)
plt.ylabel('$\mathrm{Normalized\ flux},\ f_\lambda$',fontsize=fsize)
plt.title('Kurucz theoretical spectrum of '+star+\
          ' from http://svo2.cab.inta-csic.es/theory/newov/index.
 ↪php',fontsize=14)
plt.grid()
plt.minorticks_on()
plt.tight_layout()
plt.show()
```



[109]:
```
# select a part of the spectrum that overlaps with the isolated wavelength range
fl_int = []
x_int = []
wl_int_min = min(wl_int)
wl_int_max = max(wl_int)
for i in range(n_theory):
    if wl_theory[i] >= wl_int_min and wl_theory[i] <= wl_int_max:
        x_int.append(wl_theory[i])
        fl_int.append(fl_theory[i])

n_fl_int = len(x_int)
print (n_int, n_fl_int)
print ("lambda_1 =", wl_int[0], ", lambda_2 =", wl_int[-1])
```

523 131

```
lambda_1 = 4670.0 , lambda_2 = 7285.0
```

[110]:
```
# prepare to interpolate the response function at the points in which the␣
↪theoretical flux is provided
sl = interpolate.interp1d(wl_int,sl_int)

# this is the integrand that represents the convolution of the flux and␣
↪response function
f_int = np.linspace(0,0,n_fl_int)
for i in range(n_fl_int):
    f_int[i] = fl_int[i]*sl(x_int[i])

# Simpson's rule is used here for integration
# after the integration don't forget to multiply by the flux normalization␣
↪factor
res = fl_max*simps(f_int, x=x_int)
```

1.1.5 Compare the calculated bolometric corrections BC_V with those listed for MS stars in http://www.pas.rochester.edu/~emamajek/EEM_dwarf_UBVIJHK_colors_Teff.tx

Other bolometric corrections can be calculated as described in Appendix A of the paper http://adsabs.harvard.edu/abs/2014MNRAS.444..392C

[111]:
```
BC = 4.75-2.5*np.log10(4*sc.pi*(10*ac.pc.value)**2*ac.sigma_sb.value*Teff**4/ac.
↪L_sun.value)+\
2.5*np.log10(res/res_0)-mA0

print ("For the star", star, "the bolometric correction in the band", band_sel,␣
↪"is", BC)
```

```
For the star A7V the bolometric correction in the band V is 0.02220196037281852
```

[112]:
```
#starting my selected stars here:
#So I am choosing star:
```

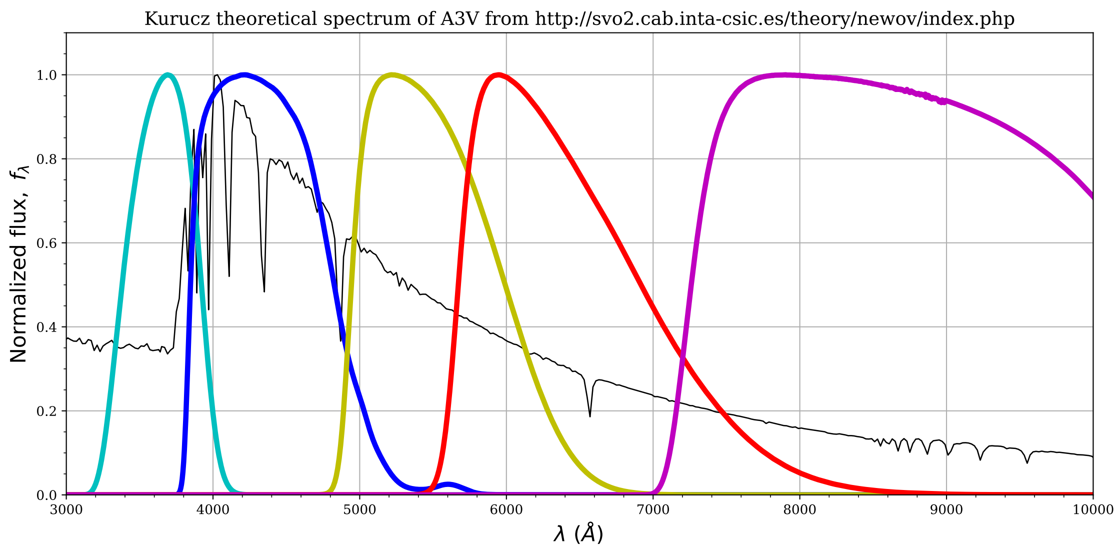
[113]:
```
# provide here parameters of the star
star = 'A3V' # its spectral class
print ("A", star, "star")

# parameters of the star,
# e.g. from http://www.pas.rochester.edu/~emamajek/
↪EEM_dwarf_UBVIJHK_colors_Teff.txt

M = 1.86
R = 1.66
Teff = 8600.
lgM = np.log10(M)
```

```
lgR = np.log10(R)
lgL = 1.09

#print ac.M_sun, ac.R_sun, ac.L_sun
g = sc.G*(10.**lgM*ac.M_sun)/(10.**lgR*ac.R_sun)**2

# Note that in expressions one should use g.value, ac.L_sun.value, etc.

print ("g =", g, "and its value in SI is",g.value)
lgg = np.log10(1e2*g.value) # 1e2 transforms m/s**2 to cm/s**2, as used in␣
↪model atmospheres
print ("lgg =", lgg, "(cm/s**2)") # for MS stars lgg should be around 4.
Teff = ((ac.L_sun.value/ac.R_sun.value**2)*10.**lgL/(4*sc.pi*(10.**lgR)**2*sc.
↪sigma))**0.25
print ("Teff =", Teff, "(K)")

file_theory = open('../Spec/'+star+'Spectrum.txt',"r") #CHANGE TO MY FILE NAME

wl_theory = []
fl_theory = []

for line in file_theory:
    data = line.split()
    wl_theory.append(float(data[0]))
    fl_theory.append(float(data[1]))

file_theory.close()

# normalize the flux
fl_max = max(fl_theory)
n_theory = len(wl_theory)
for i in range(n_theory):
    fl_theory[i] = fl_theory[i]/fl_max

# plot the spectrum and response functions
fig=plt.figure()
ax=fig.gca()
fig.set_size_inches(12,6)

plt.plot(wl_theory,fl_theory,'k-',linewidth=1)

for k in range(n_band):
    plt.plot(wl_band[k][:],sl_band[k][:],col[k],linewidth=4)

plt.xlim(wl_min,wl_max)
plt.ylim(0.,1.1)
plt.xlabel('$\lambda\ (\AA)$',fontsize=fsize)
```

```
plt.ylabel('$\mathrm{Normalized\ flux},\ f_\lambda$',fontsize=fsize)
plt.title('Kurucz theoretical spectrum of '+star+\
          ' from http://svo2.cab.inta-csic.es/theory/newov/index.
 ↪php',fontsize=14)
plt.grid()
plt.minorticks_on()
plt.tight_layout()
plt.show()

# select a part of the spectrum that overlaps with the isolated wavelength range
fl_int = []
x_int = []
wl_int_min = min(wl_int)
wl_int_max = max(wl_int)
for i in range(n_theory):
    if wl_theory[i] >= wl_int_min and wl_theory[i] <= wl_int_max:
        x_int.append(wl_theory[i])
        fl_int.append(fl_theory[i])

n_fl_int = len(x_int)
print (n_int, n_fl_int)
print ("lambda_1 =", wl_int[0], ", lambda_2 =", wl_int[-1])

# prepare to interpolate the response function at the points in which the␣
 ↪theoretical flux is provided
sl = interpolate.interp1d(wl_int,sl_int)

# this is the integrand that represents the convolution of the flux and␣
 ↪response function
f_int = np.linspace(0,0,n_fl_int)
for i in range(n_fl_int):
    f_int[i] = fl_int[i]*sl(x_int[i])

# Simpson's rule is used here for integration
# after the integration don't forget to multiply by the flux normalization␣
 ↪factor
res = fl_max*simps(f_int, x=x_int)

BC = 4.75-2.5*np.log10(4*sc.pi*(10*ac.pc.value)**2*ac.sigma_sb.value*Teff**4/ac.
 ↪L_sun.value)+\
2.5*np.log10(res/res_0)-mA0

print ("For the star", star, "the bolometric correction in the band", band_sel,␣
 ↪"is", BC)
```

```
A A3V star
g = 185.08209017372408 kg / m2 and its value in SI is 185.08209017372408
```

```
lgg = 4.267364395440939 (cm/s**2)
Teff = 8390.204308964097 (K)
```




```
523 131
lambda_1 = 4670.0 , lambda_2 = 7285.0
For the star A3V the bolometric correction in the band V is 0.04436234059657096
```

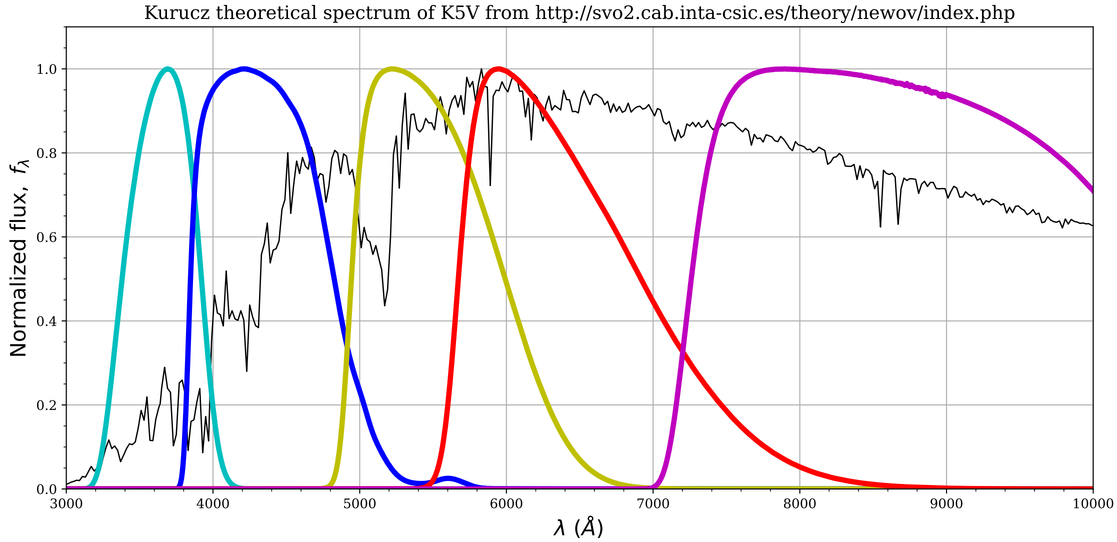
[114]:

```
# provide here parameters of the star
star = 'K5V' # its spectral class
print ("A", star, "star")

# parameters of the star,
# e.g. from http://www.pas.rochester.edu/~emamajek/
 ↪EEM_dwarf_UBVIJHK_colors_Teff.txt

M = 0.68
R = 0.698
Teff = 4410.
lgM = np.log10(M)
lgR = np.log10(R)
lgL = -0.78

#print ac.M_sun, ac.R_sun, ac.L_sun
g = sc.G*(10.**lgM*ac.M_sun)/(10.**lgR*ac.R_sun)**2

# Note that in expressions one should use g.value, ac.L_sun.value, etc.

print ("g =", g, "and its value in SI is",g.value)
```

```python
lgg = np.log10(1e2*g.value) # 1e2 transforms m/s**2 to cm/s**2, as used in␣
 ↪model atmospheres
print ("lgg =", lgg, "(cm/s**2)") # for MS stars lgg should be around 4.
Teff = ((ac.L_sun.value/ac.R_sun.value**2)*10.**lgL/(4*sc.pi*(10.**lgR)**2*sc.
 ↪sigma))**0.25
print ("Teff =", Teff, "(K)")# read in a theoretical spectrum of the star
# take the Kurucz spectrum for the above specified parameters
# from http://svo2.cab.inta-csic.es/theory/newov/index.php

file_theory = open('../Spec/'+star+'Spectrum.txt',"r") #CHANGE TO MY FILE NAME

wl_theory = []
fl_theory = []

for line in file_theory:
    data = line.split()
    wl_theory.append(float(data[0]))
    fl_theory.append(float(data[1]))

file_theory.close()

# normalize the flux
fl_max = max(fl_theory)
n_theory = len(wl_theory)
for i in range(n_theory):
    fl_theory[i] = fl_theory[i]/fl_max

# plot the spectrum and response functions
fig=plt.figure()
ax=fig.gca()
fig.set_size_inches(12,6)

plt.plot(wl_theory,fl_theory,'k-',linewidth=1)

for k in range(n_band):
    plt.plot(wl_band[k][:],sl_band[k][:],col[k],linewidth=4)

plt.xlim(wl_min,wl_max)
plt.ylim(0.,1.1)
plt.xlabel('$\lambda\ (\AA)$',fontsize=fsize)
plt.ylabel('$\mathrm{Normalized\ flux},\ f_\lambda$',fontsize=fsize)
plt.title('Kurucz theoretical spectrum of '+star+\
          ' from http://svo2.cab.inta-csic.es/theory/newov/index.
 ↪php',fontsize=14)
plt.grid()
plt.minorticks_on()
plt.tight_layout()
```

```python
plt.show()

# select a part of the spectrum that overlaps with the isolated wavelength range
fl_int = []
x_int = []
wl_int_min = min(wl_int)
wl_int_max = max(wl_int)
for i in range(n_theory):
    if wl_theory[i] >= wl_int_min and wl_theory[i] <= wl_int_max:
        x_int.append(wl_theory[i])
        fl_int.append(fl_theory[i])

n_fl_int = len(x_int)
print (n_int, n_fl_int)
print ("lambda_1 =", wl_int[0], ", lambda_2 =", wl_int[-1])

# prepare to interpolate the response function at the points in which the␣
 ↪theoretical flux is provided
sl = interpolate.interp1d(wl_int,sl_int)

# this is the integrand that represents the convolution of the flux and␣
 ↪response function
f_int = np.linspace(0,0,n_fl_int)
for i in range(n_fl_int):
    f_int[i] = fl_int[i]*sl(x_int[i])

# Simpson's rule is used here for integration
# after the integration don't forget to multiply by the flux normalization␣
 ↪factor
res = fl_max*simps(f_int, x=x_int)

BC = 4.75-2.5*np.log10(4*sc.pi*(10*ac.pc.value)**2*ac.sigma_sb.value*Teff**4/ac.
 ↪L_sun.value)+\
2.5*np.log10(res/res_0)-mA0

print ("For the star", star, "the bolometric correction in the band", band_sel,␣
 ↪"is", BC)
```

```
A K5V star
g = 382.70637336108445 kg / m2 and its value in SI is 382.70637336108445
lgg = 4.582865694763048 (cm/s**2)
Teff = 4409.596682176123 (K)
```



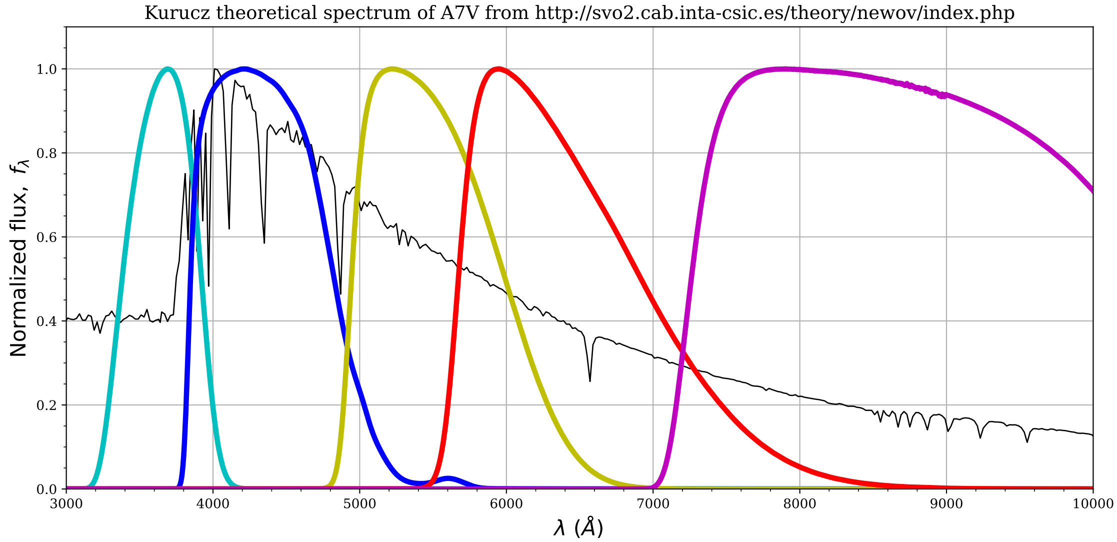

```
523 131
lambda_1 = 4670.0 , lambda_2 = 7285.0
For the star K5V the bolometric correction in the band V is -0.49718986580059976
```

[115]:
```
# provide here parameters of the star
star = 'A7V' # its spectral class
print ("A", star, "star")

# parameters of the star,
# e.g. from http://www.pas.rochester.edu/~emamajek/
 ↪EEM_dwarf_UBVIJHK_colors_Teff.txt

M = 1.77
R = 1.750
Teff = 7760.
lgM = np.log10(M)
lgR = np.log10(R)
lgL = 1.00

#print ac.M_sun, ac.R_sun, ac.L_sun
g = sc.G*(10.**lgM*ac.M_sun)/(10.**lgR*ac.R_sun)**2

# Note that in expressions one should use g.value, ac.L_sun.value, etc.

print ("g =", g, "and its value in SI is",g.value)
lgg = np.log10(1e2*g.value) # 1e2 transforms m/s**2 to cm/s**2, as used in␣
 ↪model atmospheres
print ("lgg =", lgg, "(cm/s**2)") # for MS stars lgg should be around 4.
```

```
Teff = ((ac.L_sun.value/ac.R_sun.value**2)*10.**lgL/(4*sc.pi*(10.**lgR)**2*sc.
 ↪sigma))**0.25
print ("Teff =", Teff, "(K)")# read in a theoretical spectrum of the star
# take the Kurucz spectrum for the above specified parameters
# from http://svo2.cab.inta-csic.es/theory/newov/index.php

file_theory = open('../Spec/'+star+'Spectrum.txt',"r") #CHANGE TO MY FILE NAME

wl_theory = []
fl_theory = []

for line in file_theory:
    data = line.split()
    wl_theory.append(float(data[0]))
    fl_theory.append(float(data[1]))

file_theory.close()

# normalize the flux
fl_max = max(fl_theory)
n_theory = len(wl_theory)
for i in range(n_theory):
    fl_theory[i] = fl_theory[i]/fl_max

# plot the spectrum and response functions
fig=plt.figure()
ax=fig.gca()
fig.set_size_inches(12,6)

plt.plot(wl_theory,fl_theory,'k-',linewidth=1)

for k in range(n_band):
    plt.plot(wl_band[k][:],sl_band[k][:],col[k],linewidth=4)

plt.xlim(wl_min,wl_max)
plt.ylim(0.,1.1)
plt.xlabel('$\lambda\ (\AA)$',fontsize=fsize)
plt.ylabel('$\mathrm{Normalized\ flux},\ f_\lambda$',fontsize=fsize)
plt.title('Kurucz theoretical spectrum of '+star+\
          ' from http://svo2.cab.inta-csic.es/theory/newov/index.
 ↪php',fontsize=14)
plt.grid()
plt.minorticks_on()
plt.tight_layout()
plt.show()

# select a part of the spectrum that overlaps with the isolated wavelength range
```

```
fl_int = []
x_int = []
wl_int_min = min(wl_int)
wl_int_max = max(wl_int)
for i in range(n_theory):
    if wl_theory[i] >= wl_int_min and wl_theory[i] <= wl_int_max:
        x_int.append(wl_theory[i])
        fl_int.append(fl_theory[i])

n_fl_int = len(x_int)
print (n_int, n_fl_int)
print ("lambda_1 =", wl_int[0], ", lambda_2 =", wl_int[-1])

# prepare to interpolate the response function at the points in which the␣
 ↪theoretical flux is provided
sl = interpolate.interp1d(wl_int,sl_int)

# this is the integrand that represents the convolution of the flux and␣
 ↪response function
f_int = np.linspace(0,0,n_fl_int)
for i in range(n_fl_int):
    f_int[i] = fl_int[i]*sl(x_int[i])

# Simpson's rule is used here for integration
# after the integration don't forget to multiply by the flux normalization␣
 ↪factor
res = fl_max*simps(f_int, x=x_int)

BC = 4.75-2.5*np.log10(4*sc.pi*(10*ac.pc.value)**2*ac.sigma_sb.value*Teff**4/ac.
 ↪L_sun.value)+\
2.5*np.log10(res/res_0)-mA0

print ("For the star", star, "the bolometric correction in the band", band_sel,␣
 ↪"is", BC)
```

```
A A7V star
g = 158.47647269666953 kg / m2 and its value in SI is 158.47647269666953
lgg = 4.199964796292351 (cm/s**2)
Teff = 7759.032233905629 (K)
```




```
523 131
lambda_1 = 4670.0 , lambda_2 = 7285.0
For the star A7V the bolometric correction in the band V is 0.02220196037281852
```

[ ]:

[ ]: