M2 CompuPhys
Practical Work« Astronomical data processing »

Photometric analysis of a Kuiper-Belt Object

This practical work has for objective of computing the light curve (and, consequently, the rotation period) of a small body (in that cas a Kuiper Belt Object, 29981 (1999 TD_{10})). To do that a series of preprocessed images are available, obtained during two nights in a row.

Observational data

The data provided are images in the fits format, the standard for astronomical images (uncompressed intensities), obtained during the night from October 8th to 9th and 9th to 10th, 2001 with a 1.54-m telescope located in the La Silla Observatory (ESO, Chile).

The CCD sensor has 2048x4096 pixels but it was not completely illuminated and, after preprocessing, the images have 2048x2063 pixels. The pixel size corresponds to 0.37'' on the sky, i.e. the overall field of view corresponds to 12,6x12,7'. Different broad-band filters were used during the observations but only the ones obtained with the R filter are available for this work, because they are numerous and they have the best signal-to-noise ratio. These images are oriented with North up and East at left.

All the images are preprocessed. They correspond to raw images with the BIAS subtracted and divided by a flat field (normalized to 1, computed with a combination of different images obtained on the sky during the twilight). They are, consequently, corrected for fixed-pattern noise and defaults of the optical system (such as vignetting or filters defaults), and their intensity starts to zero.

Each of the images has a header containing different descriptors providing information about different parameters (image size, time, integration time, filter, MJD...)

They are called « td10r_n2_N.fits », N being a four digit number.

Software to use

The images can be visualized with a tool called SAOimage DS9, developped by the Smithsonian Astrophysical Observatory (USA). It is possible to load different images (for example images 102 and 106) and to display them by blinking to find the moving object. Pixel intensities and positions can also be measured with this tool.

DS9 permits to detect the target or a star and to measure, approximately, its position. A more complete photometric analysis must be performed with codes written in Python (astropy).

Magnitude determination of the KBO

From the preliminary analysis conducted with DS9 it is possible to open the fits images and to measure the flux for the target and another star by writing a Python code, based on astropy.

A file providing a basic code can be used and improved to perform this analysis. The flux can be measured with disk centered on the target, from an accurate centroid position, based on the preliminary position measured with DS9. The total flux measured must be corrected from the sky background (see the two other disks visualized on the cropped image provided by this code).

The analysis will consists in :

- measuring the instrumental magnitude of 1999 TD_{10} in the different images and the one of a star appearing in the same field of view.

- from these magnitudes it is possible to computed the real magnitude of 1999 TD_{10} if the one of the star is known. This can be done thanks to other calibrated images obtained by Pan-STARRS. These images are aso available, for both nights of observation (images called « PanSTARRS_8_oct_r.fits » and « PanSTARRS_9_oct_r.fits), they have been downloaded on this website: http://ps1images.stsci.edu/cgi-bin/ps1cutouts. These images have been obtained with a r filter, having a spectral width and central wavelength very close to the R filter use to observe 1999 TD_{10}. We have R-r=0.25. In the Pan-STARRS image the real magnitude of a star can be computed with the following formula: r=mag+25+2,5Log10(t) where r is the real magnitude in the r filter, mag the instrumental magnitude provided by the Python code on the star and t the total exposure time. Because they correspond to the coaddition of different images this exposure time is long (it is given in the header, see the Python code).

- A light curve implies also to know the time of observation. This time can be expressed with the MJD parameter (« Modified Julian Day »). This parameter is also given in the header (see Python code).

From these measurements it is possible to compute a data file containing the magnitude of the target in function of time (one column with MJD and another one with the R magnitude). From these data it is possible to compute, at least approximately, a lightcurve based on a rotation period. Ideally an analysis with a periodogram can provide a good idea of the rotation period but a « visual estimate » can also be used.

Report

1. Briefly summarize the principles of a photometric analysis, with the coordinates of the target and the stars used in both field of view (8th and 9th of October).
2. Give an ascii file containing the MJD and the R magnitude of 1999 TD_{10} (with, possibly an uncertainty for R)
3. Provide a figure representing these data (magnitude = f(MJD)) with errorbars, if available.
4. Compute the rotation period of 1999 TD10 and the corresponding light curve
5. Provide also the codes used to perform this analysis

Pyhton code for starting the photometric analysis :

```
#  File M2_CompuPhys_Data_Processing.py for Astronomical data processing course
#
#  Example of Python code for measuring magnitudes in a fits image... Needs to be adapted !

import matplotlib.pyplot as plt

from astropy.io import fits
hdulist = fits.open('td10r_n2_0133.fits')      # Define the image name to open
hdulist.info()                                 # To get information on the image


header = hdulist[0].header


scidata = hdulist[0].data                # transfer the image (pixels values) in scidata
print(scidata.shape)
mjd = header['MJD-OBS']
print(mjd)
exptime = header['EXPTIME']
print(exptime)

#print(repr(header))              # to print all the descriptors in the header, if necessary
#print(repr(header['MJD-OBS']))     # to print a given descriptor, if necessary
#scidata /= exptime               # divide the image by its exposure time, if necessary

r=8        # radius for measuring target flux
r_in=15     # inner radius for the annulus used for measuring sky background
r_out=25    # outer radius for the annulus used for measuring sky background


import numpy as np

from photutils import CircularAperture, CircularAnnulus, aperture_photometry
from astropy.visualization import simple_norm

Xcent=1083     # approximate target coordinates (x)
Ycent=660      # approximate target coordinates (y)


from photutils.centroids import centroid_com, centroid_1dg, centroid_2dg, centroid_quadratic
data = scidata[Ycent-40:Ycent+40,Xcent-40:Xcent+40]  # subimage centered on the target

x1, y1 = centroid_quadratic(data)     # for computing accurately the position of maximum of brightness
print('resultats centroid =%.3f'%(x1),',%.3f'%(y1))


norm = simple_norm(data, 'sqrt', percent=99)
plt.imshow(data, norm=norm, origin='lower')
positions = [(x1, y1)]
aperture = CircularAperture(positions, r)
annulus_aperture = CircularAnnulus(positions, r_in, r_out)

aperture.plot(color='white', lw=2)
annulus_aperture.plot(color='white', lw=2)


phot_table = aperture_photometry(data, aperture)
phot_table['aperture_sum'].info.format = '%.8g'  # for consistent table output
print(phot_table)
Sum_target_raw=(phot_table['aperture_sum'])

print('\nSum_target_raw=%.3f\n'%(Sum_target_raw))

phot_table = aperture_photometry(data, annulus_aperture)
phot_table['aperture_sum'].info.format = '%.8g'  # for consistent table output
print(phot_table)
Sky_background=(phot_table['aperture_sum'])

print('\nSky_background=%.3f\n'%(Sky_background))




from math import log10
Sum_target=Sum_target_raw-Sky_background/(r_out*r_out-r_in*r_in)*(r*r)
print('Target flux without sky=%.3f\n'%(Sum_target))
Magnitude=-2.5*log10(Sum_target)
print('instrumental magnitude=%.3f\n'%(Magnitude))



hdulist.close()

plt.show()
```