# Conversion To JWST Magnitudes From Other Magnitudes

## Introduction

The code described here is used to convert other magnitudes into approximate JWST filter magnitudes for any of the instruments (NIRISS, Guider, NIRCam, MIRI, and NIRSpec). This conversion is based upon the results of simulating magnitudes in many filters, including the JWST imaging filters, for different spectral shapes. Given input colour-magnitude data in two filters that are simulated, the code can produce approximate conversions to JWST filter magnitudes.

The program, jwst\_magnitude\_converter.py, uses as input files that contain simulated magnitudes for different sets of spectra produced by the colcor.py program. It is designed to read in pairs of magnitude values, or magnitude/colour pairs, from an input file that can be either an ascii file with regular columns that is readable by the numpy.loadtxt routine or from a FITS table file. Once the input data values have been read in, one can define which JWST imaging filter magnitudes one wants to calculate, and produce a new colour/magnitude diagram in the NIRISS filters based on the colour relations between the input simulated magnitudes.

The code needs to be used with caution; not all combinations of possible input magnitudes produce colour-magnitude relations that transform well into NIRISS magnitudes. If, for example, the simulated colour-magnitude relation is double valued in the magnitude for a given colour value , as is the case in the Johnson B and V filters for example, then the code will produce a bad result. One also needs to bear in mind that the values are calculated for spectral shapes with no interstellar (or circumstellar) extinction effects. If the observed colours for a source that is reddened are used as inputs to the code, the outputs will in general be incorrect. In such instances one would need to de-redden the input photometry, transform it, and the redden the output photometry. The code here has no such capabilities.

The code is written in Python 2 and requires a number of packages: matplotlib for producing plots, Tkinter for making the interface, numpy for reading ascii files and carrying out the calculations, astropy for reading FITS tables, and the math package. The code uses configobj for parameters. The code should also work in Python 3 as well as Python 2.

## The Simulated Magnitudes

The program requires output magnitude listings from the magnitudes.py program. These magnitudes are based upon a comparison of either the integrated total photon count rate (for the NIRISS, NIRCam, Guider, MIRI, and NIRSpec filters plus the GAIA filters) or the integrated total flux (for the other filters). The standard Vega model spectrum used for Hubble Space Telescope and Spitzer Space Telescope absolute calibration (Bohlin, 2014) is used as a reference spectrum for the calculations in most cases. To calculate relative magnitudes the input spectra are scaled to match the Vega spectrum flux density for magnitude 15.0 at 2.159 μm, the effective wavelength of the 2MASS Ks filter. This normalization is arbitrary, but it serves to anchor the spectra at a wavelength close to the middle of the NIRISS wavelength range. The reference Vega spectrum is assumed to define a magnitude of +0.03 unless there is explicit statement in the literature that the Vega magnitude is taken to be 0.00 for a set of filters; the latter is the case for the Hubble Space Telescope filters and the 2MASS filters, for example.

For each filter that is simulated the Vega spectrum is multiplied by the filter throughput and integrated to produce a total flux. For the JWST instrument the integration is done in photon flux density rather than the wavelength flux density Fλ because in these cases the filter response profiles being used include the full telescope and instrument throughput including the detector quantum efficiency. In the photon counting case the telescope mirror area is used in the calculation, and a total observed count-rate is produced. For the non-JWST filters it is generally assumed that the filter profiles do not include effects such as the telescope throughput and the detector response, although this is known to be wrong in some instances. For most of the non-JWST filters the flux density is used in the integration. In either case, the integrated signal from the scaled Vega spectrum is assumed to define magnitude 15.0, and the integrated signal from the input spectrum is compared to the scaled Vega spectrum value to produce a simulated magnitude. This use of a scaled Vega spectrum as a standard signal is used to produce an approximate standardization of the model magnitudes.

For the GAIA filters the reference spectrum is a Castelli-Kurucz stellar model with a specific normalization in brightness. The reference spectrum was reproduced for the magnitude simulation. In this case, since the throughput curves include the detector response and since there is no issue of atmospheric absorption the photon flux density is used in the calculation

The Sloan and Pan-STARRS magnitudes are calculated differently because they are defined on an AB-type system. While the code simulation code follows the prescription for the calculation of the Sloan magnitudes as published, comparison of the model colours with Sloan data suggests that these simulations are less accurate for these magnitudes than they are for the Vega-magnitude systems. The Pan-STARRS magnitudes also are calculated as outlined in the documentation, in that case by comparing the flux from the source spectrum to the flux from a flat spectrum with Fν = 3631 Jy.

Calculations have been made comparing the results of the flux ratios to the photon flux ratios for the different non-JWST filters, and it is found that this does not introduce a large effect on the output magnitudes. The uncertainties in the exact filter profiles and in the correction for atmospheric effects are significantly larger than any uncertainties introduced by the using power flux density rather than photon flux density.

There are five sets of simulated magnitudes available for use with the program: first, a set of simulated magnitudes from the Kurucz (1992) Atlas9 stellar atmosphere model spectra, using the 412 models in solar metallicity set. These models cover effective temperatures from 3500 K to 50000 K and surface gravity log10(g) values between +0.0 and +5.0. In addition to the 410 model spectra in the main grid, there is a solar model spectrum and a Vega model spectrum in the set. This is the default set of values to use with the magnitude conversion. The second set of models are the Phoenix models from Husser et al. (2013). These models have a much more detailed wavelength coverage than the Kurucz model set, but the wavelengths only go to 5.5 μm so calculations cannot be carried out for various longer wavelength filters. The Kurucz models have sparse wavelength coverage at long wavelengths, but they extend far enough in wavelength to allow an approximate simulation of mid-infrared filters. The Phoenix models cover effective temperatures from 2300 K to 12000 K and log10(g) values from +0.0 to +6.0, and are for solar metallicity.

The third set of models are a sub-set of the BOSZ models described by Bohlin et al. (2017). That set of models uses the Kurucz Atlas9 code with an expanded wavelength coverage to allow simulation of the JWST instrument magnitudes. The library of models is very large, and for the current usage only the sub-set of BOSZ models with unaltered solar abundances is used, denoted the “normal composition” sub-set. Although these models cover the wavelength range to 32 μm, to model such things are the IRAS and Spitzer magnitudes the models were extrapolated in wavelength to 200 μm in the same way as was done for the Kurucz models.

The fourth set of model magnitudes are from pure blackbody spectra with temperatures between 100000 K and 1000 K. One can if needed use a version of the blackbody magnitudes file that extends down to 100 K, but in general it is better to use the more restricted temperature range. While the blackbody spectral shape is unrealistic for most stars, these magnitudes are certain to produce a smooth transformation between the different magnitudes. This is not certain for either of the stellar atmosphere model sets.

## Filters Available for the Transformations

The filters for which the simulated colours are calculated include a variety of optical, infrared, and mid-infrared filters. The longer wavelength filters are generally from space missions (IRAS, Spitzer, WISE) rather than ground-based instruments. The shorter wavelength filters are a mixture of ground-based and Hubble Space Telescope filters.

The following lists the available filters. Those shown in red are longer wavelength filters that are not simulated for the Phoenix models because of the limited wavelength coverage of those models. The Kurucz and blackbody models have simulated magnitudes for all filters while the Phoenix models only have those printed in black.

1. NIRISS filters: F090W, F115W, F140M, F150W, F158M, F200W, F277W, F356W, F380M, F430M, F444W, F480M
2. NIRCam filters: F070W, F090W, F115W, F140M , F150W, F150W2 , F162M, F164N, F182M , F187M , F200W , F210M , F212N , F250M , F277W , F300M, F322W2 , F323N , F335M , F356W, F360M , F405N , F410M , F430M , F444W, F460M , F466N, F470N, F480M
3. Guider “filters”: Guider 1 and Guider 2 for the two detectors
4. NIRSpec filters: Clear, F110W, F140X, F070LP, F100LP, F170LP, F290LP
5. Hubble Space Telescope WFC3 filters: F218W, F225W, F275W, F336W, F390W, F438W, F475W, F555W, F606W, F625W, F775W, F814W, F105W, F110W, F125W, F140W, F160WF218W, F225W, F275W, F336W, F390W, F438W, F475W, F555W, F606W, F625W, F775W, F814W, F105W, F110W, F125W, F140W, F160W
6. Hubble Space Telescope ACS filters: F220W, F250W, F330W, F435W, F475W, F555W, F606W, F625W, F775W, F814W
7. Optical and near-infrared filters: Sloan u, Sloan g, Sloan r, Sloan i, Sloan z, Bessel U, Bessel B, Bessel V, Bessel/Cousins R, Bessel/Cousins I, Johnson U, Johnson B, Johnson V, Johnson R, Johnson I, Johnson J, Johnson H, Johnson K, Johnson L, Johnson M, Johnson N, 2MASS J, 2MASS H, 2MASS Ks, DENIS i, DENIS J, DENIS K, UKIDSS Z, UKIDSS Y, UKIDSS J, UKIDSS H, UKIDSS K, Pan-STARRS1 g, Pan-STARRS1 r, Pan-STARRS1 i, Pan-STARRS1 z, Pan-STARRS1 y, Pan-STARRS1 w
8. Space-based infrared filters: IRAS [12], IRAS [25], IRAS [60], IRAS [100], IRAC [3.6], IRAC [4.4], IRAC [5.7], IRAC [8.0], MIPS [24], MIPS [70], MIPS [160], WISE W1, WISE W2, WISE W3, WISE W4

Note that even when some filters are simulated these are bad choices for transformation to NIRISS magnitudes, at least for stars. The stellar colours become very small for most normal stars at wavelengths beyond H-band. Use of IRAS or MIPS filters for a transformation to NIRISS magnitudes is likely to produce bad results.

Using short wavelength filters also may cause issues depending upon the exact filter pair used. The Johnson U, B, and V filters are known to produce double-valued colour relations for normal stars and thus are less likely to give a good result when a transformation is attempted. More generally, the quality of the transformations depends on the accuracy of the filter profiles that are used in the simulations. The filter profiles used to make the simulated magnitudes come from a variety of sources and may have more sparse sampling in some cases. The user needs to be cautious about transforming from optical wavelengths or from wavelengths longer than 5 μm into the NIRISS filters. The transformation works best if one or both of the filters overlaps with the NIRISS wavelength range, and if one filter is at a wavelength shorter than 2 μm.

## Use of the Program

The program is intrinsically interactive and needs to be invoked from the command line to start. Upon issuing the niriss\_magnitude\_converter.py command, the code brings up the user interface and attempts to read in the simulated colours for the Kurucz, Phoenix, and blackbody model sets. The appearance of the widget upon its initial start is shown in Figure 1 below. If the reading of the four files magslist\_old\_kurucz.new, phoenix\_grid\_magslist.new, magslist\_bosz\_normal.new, and blackbody\_magslist.new succeeds, then a message is printed in the text box at the top of the widget as shown in the Figure. If not, then a message appears stating that the magnitudes could not be read in and that transformations cannot be derived.

The four required files of simulated magnitudes need to be kept in a directory defined by the path variable. This value is defined as the path the program itself, so as long as the program and the three magnitudes files are kept together the program should execute properly. If the four magslist files cannot be read, the code can still be used to read in magnitudes and plot colour-magnitude diagrams, but transformations to JWST magnitudes will not be possible.

The main window has several sections from top to bottom, as shown in Figure 1 below. The first area at the top is a message area. Below that is a region where one defines the magnitudes to be read in and gets data from a file. This needs to be done first before any of the other functions of the widget can be used. There are two pull-down menus that allow one to select which magnitudes are to be read in from a file. The filters that are available depend on whether the Kurucz/blackbody or Phoenix model sets are selected. Initially the Kurucz model set is selected and the full set of filters (aside from the NIRISS ones) are shown in the menus. One needs to select two different filters in the two menus.

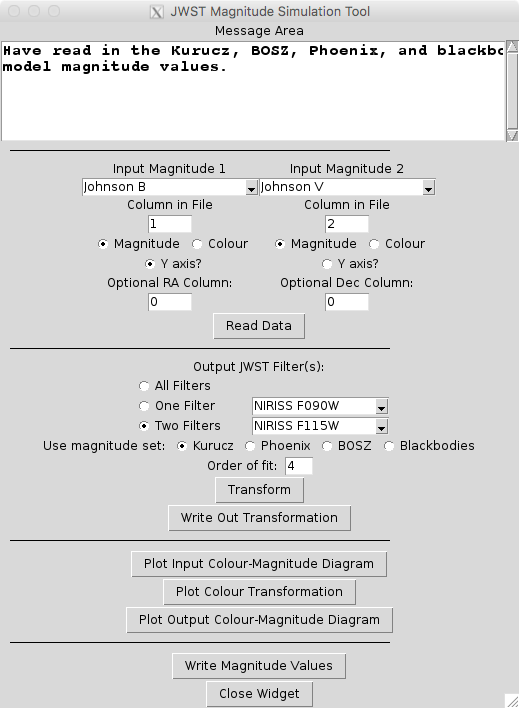


Figure 1: The appearance of the jwst\_magnitude\_converter.py main window upon start-up, if the "standard" magnitude values have been read in correctly in this case.

The code can read in either two magnitude values or one magnitude and one colour formed from the two filters that are specified. One needs to tell the code whether the input values are magnitudes, in which case the colour is formed from the magnitude selected in menu 1 at left minus the magnitude selected in menu 2 at right, or whether one value is a colour value. In the latter case, one needs to set the radio button below the menu from “magnitude” to “colour”. Either the first or second input value can be a colour, whereupon the other input value is taken to be a magnitude.

Directly below the magnitude selection menus are fields that let one define which columns in the input data file correspond to the selected magnitudes or colours. The column values count from 1. It is assumed that the input data values are in regular columns. The file can be a FITS table whereupon the indicated columns are extracted, or it can be an ascii file. In the latter case columns with ‘#’, ‘\’, and ‘|’ are assumed to be comment columns (as with the IPAC GATOR data tables) and are ignored in reading in the data.

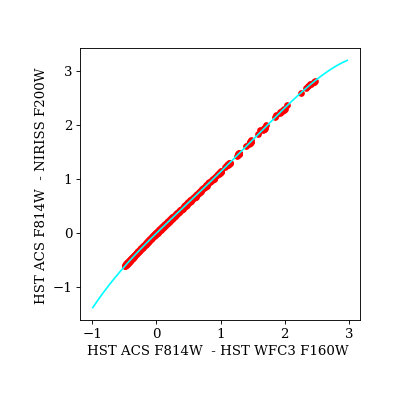


Figure 2: The simulated colour relation between ACS F814W − WFC3 F160W and ACS F814W − NIRISS F200W, as predicted from the Kurucz model set. The red points show the simulated colours for different stellar temperatures in the models. The cyan line shows the polynomial fit relation. The fit is used to convert the F814W − F160W colour to F814W − F200W colour, and then with the F814W magnitude the estimated F200W magnitude is obtained.

When two magnitude values are read in, one of the two needs to be selected as the y axis quantity in a colour-magnitude plot. For example, if the input magnitudes are the Johnson B and V magnitudes, the colour will be B − V but one can plot either V with B ­− V or B with B − V. The “Y axis?” radio button selects one of the two magnitudes to be the y axis value if a colour-magnitude plot is produced.

One can also optionally read columns for the star positions (α,δ) in the file. These are assumed to be string fields since the positions may be specified in sexagestimal form (i.e. 05:42:08.00 –69:12:42.0 for example). If the column values in the “Optional RA column” and “Optional Dec column” entry fields are greater than zero, the code tries to read the columns in and, if successful and if the number of values equals the number of data values, the code will write these values out with the magnitudes if ascii output is requested. If any issues are encountered, these values will not be written out. They are not used internally in the program.

Below the menus and the option buttons is the “Read data” button. One uses this button to select an input file and attempt to read the data in. If the reading is successful, the number of data points read in is given in the text message area. Otherwise a message is put there saying that reading in the data failed.

The next section defines which NIRISS magnitudes are to be simulated. One can simulate all 12 magnitudes, two magnitudes, or just one magnitude. Normally it is assumed that two magnitudes will be simulated to make a colour-magnitude diagram for comparison with the input colour-magnitude data. In the central part of the widget there are menus for which JWST magnitudes are to be used, along with a radio button to select one of the three possible options. In this section one can also select which set of magnitudes to base the transformation on.

Once one or more JWST magnitudes are selected in the menus, the program is ready to calculate transformations from the input colour/magnitude pairs to simulated NIRISS magnitudes. The transformation is made from input magnitudes M1 and M2 to NIRISS magnitude N1 by looking at the relation between M1 − N1 and M1 − M2. To take a concrete example, consider input magnitudes ACS F814W and WFC3 F160W and NIRISS output magnitude F200W. The input colour ACS F814W − WFC3 F160W is plotted against the predicted ACS F814W − NIRISS F200W colour as in Figure 2. In this case there is a smooth relation between the two colours. A fourth order polynomial fit is made to the points, as shown by the cyan line in the Figure.

When fits are made, the root mean square deviation value of the output colour from the input model points is calculated and the values are written to the message area, along the range of absolute deviations. These values allow the user to have some idea of whether the transformation is accurate or not. If the root mean square deviation is large, or if the range of deviations is large, these would show that the transformation is of limited utility. It is possible that the transformation will still be good for a limited range of input colour (i.e., for early type stars, for example) and that it can still be used if the input magnitudes are within the useful range. This judgment is left to the user. There is no mechanism within the program as it is written to limit the input colour range for the transformation in this manner.

From the fit one can transform F814W − F160W to a predicted F814W − F200W for any object with input F814W − F160W colour within the fit range to a F814W − F200W colour. Once this value is found, the F200W magnitude is estimated as F814W − (F814W − F200W). The transformation is carried out for all the input points. The same process can be done for all of the NIRISS filters to produce a full set of simulated magnitudes. Objects with F814W − F160W colours beyond the fit range are given the colour at the nearest end of the range for the transformation. These will have inaccurate output magnitudes.

Exactly what output is created depends on which JWST filters are selected. If all filters are selected then the above type of transformation is carried out 59 times and all the output magnitudes are stored. If two filters are selected then each one is calculated individually. If only one filter is selected then just the one transformation is calculated.

Another thing to note is that the JWST magnitudes are also included in the input magnitude selection menus. This allows one to try to transform, say, NIRCam F090W − F115W colours to NIRISS F090W − F115W colours. If one is doing that type of transformation one may need to pay special attention to the transformation fit, as one may get a trivial transformation depending on how one selects the input and output magnitudes. One also may get unusual extrapolations of the fit depending on the input colour values and the order of the fit. Finally, it is found that the Phoenix grid models tend to show a large variation in the values in the two-colour plots that are being fit when one considers the low temperature objects. This can in some cases produce poor quality fits for the very red objects. In general, the Kurucz and BOSZ models are much better behaved in this regard. If the Phoenix models are being used for a transformation more caution about the fitting is needed.

The fitting is done by the “Transform” button. A message is posted to indicate either success or failure of the fitting. The transformation area has an entry field for the order of the fitting. The fitting is done using the numpy.polynomial.legendre package. Hence it uses lgendre polynomials for the fitting. The default order is 4, but this can be changed by entering some other positive integer value in the entry field. Once the fit coefficients have been calculated, these are applied to a range that is 0.5 magnitudes larger than the input data range on each end. The fit is calculated at 1000 points within this range for the plotting of the fit. These values can be written out to an output file, along with the coefficients, using the “Write Out Transform” button. In the code the transformation is applied directly to the input colour values using the numpy.legendre.legval function, and then the values beyond the fit minimum and maximum colour are assigned the end point values of the transformation.

Below the transformation area are the plot buttons. One can plot the input colour-magnitude values, the transformation from the input colour value to the NIRISS colour value, or the output NIRISS colour-magnitude diagram. All three plots share a common interface and have a general appearance as in **Figure 3**. The input data here are the absolute B and V magnitudes for all stars with Hipparchos parallax values of better than 2% relative accuracy. This is a plot of input colour/magnitude values from the input B and V absolute magnitudes.

The layout of the window is the same for all plots. The plot area is in the upper part of the window. When the cursor is in the plot area the position is written to the status line just above the plot. Below the plot are entry fields for the range of the plot plus some function buttons. One can change the range by changing the values in the entry fields for the x and y ranges and clicking on the “Apply Range” button. One can revert to the original range by clicking on the “Default Range” button.

When the plot range is changed and the points are re-plotted the y axis may end up inverted compared to what is wanted. Hence there is a button to invert the y axis values. It is not clear to the author why this happens, hence it has not been fixed in the code.



Figure 3: An example of the plot window. All plots share the same basic widget functions.

Finally, at the bottom of the window are buttons to produce output files. One can export the plot as a postscript file, export the plot as a PNG file, write out the (x,y) values to an ascii file. As in this area is a button to close the window.

In the case of the plot of the transformation, there is an extra button allowing one to switch to see the transformation of the second NIRISS filter when two such are specified. When the plot is switched, the range field are not updated. To go back to the original transformation plot one needs to close the window and bring it up again.

## Command-line Usage

The code is intended to be interactive because generally one needs to look at the quality of the fitting before carrying out a transformation. However, there is a command line option allowing the code to use a parameter file to specify the filters and some other options, and the code will attempt an automatic fitting and conversion. The parameter file is assumed to be a configobj file, so one needs to have this package installed for the code to work. Even if the option is not going to be used, the code imports the configobj package to the code will not run unless it is installed. The installation instructions can be found at <https://pypi.org/project/configobj/>.

An example of the parameter file structure is shown immediately below. The file needs to have the two sections and the specific “parameter=value” entries. The first section defines the input magnitudes and the input data file name. The second section defines the output two NIRISS magnitudes, the model set, the fitting order, and the output file name.

[Input\_Magnitude\_Parameters]

filter1=HST\_ACS\_F814W

filter2=HST\_WFC3\_F160W

column1=3

column2=4

column1type=magnitude

column2type=magnitude

yvalue=1

datafile=m31\_f814\_f160\_subset.data

racolumn= -1

deccolumn= -1

[Output\_Filter\_Values]

niriss1=F115W

niriss2=F200W

modelset=Kurucz

fitorder=4

outfilename=niriss\_transformed.txt

The filter names used in the program have spaces in them, in general. For the parameter file one needs to use underscore characters, “\_”, in place of spaces. The code will replace the underscores with spaces in the strings before comparing these with the available filter list for whichever model set is selected.

The racolumn and deccolumn parameters can be set as in the widget, to columns to read for the position values of the stars. If these values are zero or negative this is not attempted.

The input magnitude parameters define the two input filter names, the columns for these values in the input file, the type of input value either magnitude of colour, which one is to be used as the magnitude value in a colour-magnitude diagram (1 or 2), and finally the data file name. The file can be in the local directory or have the path defined. The program will try to read in the data as indicated.

The second part defines the output JWST magnitudes to calculate. The two filters names need to be given. Both magnitudes will be calculated from the input colour and magnitude values. The model set needs to be specified as “Kurucz”, “Phoenix”, “BOSZ” or “blackbody”. The fit order needs to be given as some integer value 2 or larger, and then the output magnitude file name needs to be specified.

If all the parameters are read in properly and the filter names match up to the allowable ones for the given model set, then the program will attempt the transformation and if it works will write out the values to the specified file. The program if run non-interactively will print out a message at the end of the processing to indicate success, otherwise an error at any point in the process will produce an message that the program failed to produce the output file. In case of failure of the process of getting the data and transforming the magnitudes there is no way to indicate at what point in the calculations there was an error. Hence it is best to attempt a trial calculation interactively before going to an automatic transformation.

In addition to the specified output file, the code will produce two PNG files, showing the colour transformation plots for the two specified NIRISS filters. The file names for these plots include the three filter names used in the transformation along with the magnitude set name. Hence the resulting file names are somewhat unwieldy. An example file name is “transformation\_HST\_ACS\_F814W\_minus\_HST\_WFC3\_F160W\_to\_HST\_ACS\_F814W\_minus\_F115W\_kurucz.png”. The name indicates the input colour is ACS F814W – WFC3 F160W and that the output colour is ACS F814W – NIRISS F115W, and that the Kurucz model set is used in the transformation.

One can get single JWST magnitude output by having the same name in the two JWST filter name fields in the parameter file. If this is the case, the result is the same as in the interactive usage when the “One Filter” radio button is selected. Both filter name parameters still need to be defined in the configuration file.

## Contents of the Package

The package includes the program, the required model magnitude files, and an example file set that can be used for testing the program. The example can be run either interactively or from the command line. The minimum set of files needed to run the program are:

Magslist\_bosz\_normal.new

magslist\_blackbody.new

magslist\_old\_kurucz.new

magslist\_phoenix\_grid.new

niriss\_magnitude\_converter.py

The four “magslist” input files need to be stored in the same directory as the python code for the program to work properly.

The file magslist\_blackbody.new contains simulated magnitude values for blackbody models with temperatures between 100000 K and 1000 K. If one needs a larger range of temperature for some magnitude conversion the alternate file magslist\_blackbody\_fullrange.new can be used in the place the “regular” blackbody colours file. The simplest way to do this is to uncomment line 262 in the program and comment out line 263 in the program. That would mean that the program reads in the alternate file rather than the usual one. As it is intended, the smaller blackbody temperature range should be used. If the larger range is used, the quality of the transformation is likely to suffer.

Other files are provided along with the code. These are:

hst\_to\_niriss.cfg

m31\_f814\_f160\_subset.data

magnitude\_conversion.docx

magnitude\_transform.py

magslist\_blackbody\_fullrange.out

niriss\_transformed.txt

readme.txt

transformation\_HST\_ACS\_F814W\_minus\_HST\_WFC3\_F160W\_to\_HST\_ACS\_F814W\_minus\_F115W\_kurucz.png

transformation\_HST\_ACS\_F814W\_minus\_HST\_WFC3\_F160W\_to\_HST\_ACS\_F814W\_minus\_F200W\_kurucz.png

The file hst\_to\_niriss.cfg gives an example of the parameter file needed to run the code non-interactively. It contains a random subset of the photometry data from the Panchromatic Hubble Andromeda Treasury survey (see Dalcanton et al., 2012) in position brick 23, using all stars with photometry in the Hubble ACS F814W filter and the Hubble WFC F160W filter. There are about 2.4 million stars in this data sub-set with photometry in the two filters, and so a random number generator was used to select 1% of the objects into the sample data file. Columns 1 and 2 of the file give the star positions (RA,Dec) in degrees. Column 3 gives the F814W magnitude values and column 4 gives the F160W magnitude values. These magnitudes overlap with the NIRISS wavelength range and this helps to produce a smooth transformation to the NIRISS filters as predicted by theory. When these magnitudes are converted to NIRISS F115W and F200W filter magnitudes the values listed in file niriss\_transformed.txt result. Running the code with the .cfg file should reproduce the output file provided. It will also produce the two transformed\*.png files. These are automatically generated plots of the transformations to the two NIRISS filters, so that the user can examine these and see if there are issues.

Also provided in the package is documentation: this document and the readme file that gives a short description of the files in the package and how to run the code.

## The magnitude\_transform.py Program

Along with the main program, a utility code named magnitude\_transform.py is provided. This code can be used to look at the colour-colour plots for the input magnitudes from the different available magnitude sets: BOSZ, Kurucz, Phoenix, and the blackbody models. One can also see the fits made with the Legendre polynomial routine on the plots if one wishes. This code is intended to allow users to look at the magnitude transformations and determine whether they are useful for a given data set. The main code can show the transformation but only for one set of magnitudes at a time, and it requires input data values. The utility code has no such limitations.

To use the code one selects four filters, not all of which need to be distinct, from the filter list menus. One specifies filters 1 and for 2 for the x value, magnitude 1 – magnitude 2, and filters 3 and 4 for the y value, magnitude 3 – magnitude 4. Some filters are not available for the Phoenix model set as described earlier. If any of the filters are not available for the model set then no plot is produced. Otherwise, once the filters are selected one can use the button to plot the colour-colour diagram. There are selector buttons for each of the 4 model sets. One can plot all of these or any selection of one or more sets. The plot of the simulated colours uses black, blue, forest green, and red circles for the Kurucz, BOSZ, Phoenix, and blackbody model sets respectively. If one asks for the model fits these are plotted as lines of colours slate grey, cyan, green, and orange respectively.

The other difference between the code here and the main magnitude conversion code is that any filters may be used for the plot. One can plot the simulated NIRISS F090W – F115W and the simulated NIRCam F090W – F115W colour-colour plot, for example. The only cases that are not allowed are where the two filters are the same for the x or y values, or having the same values plotted in both x and y.

Examination of the results for a number of filters suggests that the Phoenix grid models need to be used with caution, because the colours generally are double-valued for the cool objects. This may not be an issue if one has an input set of magnitudes only for K-type and earlier stars. For any given filter pair one may need to look explicitly at the simulated magnitude values to see at what temperature the colours become double-valued. These comparisons also show a very strong similarly in the Kurucz and BOSZ model sets, as expected.

The magnitude\_transform.py code looks for the input “magslist” files in a directory defined by an environment variable $SIMULATED\_MAGNITUDES\_PATH. If this variable is not defined, then the current directory path is used. Hence one either needs to run the code in the directory where the magslist files are located or define the path in the environment variable.

## References

Bohlin, R. C., 2014, AJ, 147, 127B.

Dalcanton, J. J., Williams, B. F., Lang, D., Lauer, T. R., et al., 2012, ApJS, 200, 18.

Husser, T.-O., Wende-von Berg, S., Dreizler, S., Homeier, D., et al., 2013, A&A, 553, 6.

Kurucz, R. L., 1992, IAUS 149 “The Sellar Populations of Galaxies” (Beatriz Barbuy and Alvio Renzini ed.), Kluwer, Dordrecht, p. 225.