

To: Distribution

Date: 15 May 1988.

From: Keith Horne

Subject: A SYNTHETIC PHOTOMETRY PACKAGE

The generation of absolute flux distributions for all photometric and spectrophotometric calibration targets requires a set of software tools to manipulate the observational and model data. Of fundamental importance are utilities to compute the quantities defined in the paper 'Synthetic Photometry and the Calibration of the Hubble Space Telescope', but we will also need input/output formatters, scaling/resampling routines, editors, display and plot routines.

Some of these utilities were already available in my proprietary interactive data analysis package and I have chosen to implement additional routines in that environment. The subset relevant for the calibration effort is available on a special ('ATLAS'-) account.

A number of catalogues is directly accessible from the package. The default data-format is described in Appendix A but access to SDAS tables will be provided as soon as the data-format situation stabilizes.

A listing of subroutine and functions presently implemented is attached as Appendix B. Conspicuously lacking at the present time is an interface to the graph tool described elsewhere. This will be provided shortly. Note that these routines can be linked into any VAX/VMS-language routine.

Finally, Appendix C provides proof in graphical form that the package can actually do some useful things. The examples shown involve UBV_R passbands in combination with an A0 and a (reddened) O9 star spectrum from the King-Pruzial catalogue. Pivot-wavelengths are shown as small circles, whereas the 'error-bar' represents the effective width of the passband for the given flux distribution.

[Note: This description is intended as a progress report only and is neither exhaustive nor complete.]

Distribution:

ISB all
CDBS (W. Pence ; 5 copies)
SDAS (B. Hanish ; 5 copies)
PEPSI (M. Johnston ; 5 copies)

Access Subroutines and Format Conventions for Atlases of Flux Distributions

Keith Horne and Roland Buser. May 16, 1986

We are compiling a collection of atlases of flux distributions representing a wide range of stellar and non-stellar types. The collection includes both atlases of spectrophotometric observations, e.g. the Gunn-Stryker and Jacoby *et.al* atlases, and atlases of model atmosphere spectra, e.g. the Kurucz and King/Bruzual grids. These provide building blocks from which we expect to construct hybrid flux distributions that combine both the realism of the spectrophotometric observations and the complete wavelength coverage of the models. The hybrid atlases will be used in the calibration effort to construct complete and realistic energy distributions for the flux-calibration targets to be used by HST. There are many other applications throughout STScI, e.g. instrument simulators, exposure-time estimators, and scientific evaluation of HST passband systems, for which access to the atlases can expand capabilities in the area of stimuli. The use of a common set of stimuli is important to insure that these activities give results that are directly intercomparable.

In section 1 we describe two FORTRAN subroutines LOADATLAS and EVALATLAS that provide for easy access to the spectrum atlases. The first subroutine gives fluxes on the original wavelength set of the atlas, the second interpolates the atlas fluxes to a wavelength set specified by the user. Both are suitable for use in either interactive or batch environments. These subroutines give access to the atlas spectra in a way that is independent of the specific format of the disk files in which the spectra are stored. We would like to encourage the use of these subroutine interfaces throughout the institute, to avoid problems when and if the data formats change.

In section 2 we describe the internal data format we have adopted for the disk files in which the atlas spectra are stored. This is a flexible format that supports non-uniformly spaced wavelength sets, permits either binary or ASCII data formats, and provides on-line assistance for interactive users. The format is suitable for our present purposes, but may be extended to accommodate other applications. The subroutine interfaces will not thereby be affected.

1. Access to Atlas Spectra through the Subroutine Interface

Two FORTRAN subroutines provide access to the atlas spectra.

`LOADATLAS(N WAVE, WAVE, SPEC, ATLAS, NUMBER)`

Loads an atlas spectrum and its wavelength set.

input:

`N WAVE = I4` maximum number of wavelengths

`ATLAS = C*` atlas prefix (e.g. `SHR0:[BUSER.MODELS.KB]KB`)

`NUMBER = I4` desired spectrum number (0 selects interactively)

output:

`N WAVE = I4` number of wavelengths

`WAVE = R4` wavelengths (Angstroms)

`SPEC = R4` atlas spectrum data (mJy)

`NUMBER = I4` atlas spectrum number

`EVALATLAS(N WAVE, WAVE, SPEC, ATLAS, NUMBER)`

Evaluates atlas spectrum on a user-specified wavelength set (by linear interpolation and constant extrapolation).

input:

`N WAVE = I4` number of wavelengths

`WAVE = R4` wavelength set (Angstroms)

`ATLAS = C*` atlas prefix

`NUMBER = I4` desired spectrum number (0 selects interactively)

output:

`SPEC = R4` atlas spectrum data (mJy)

`NUMBER = I4` atlas spectrum number

These subroutines are part of the synthetic photometry subroutine library `DISK$TYCHO:[ATLAS.SYNPHOT]XCAL/LIB`. An interactive test program that can be used to plot selected atlas spectra can be found in `DISK$TYCHO:[ATLAS.SYNPHOT]PLOTATLAS`. We use `PLOTATLAS` to verify that a newly formed atlas has a correct format, and it may also be useful as an example for users wishing to access atlas spectra from their own applications programs. The plots are made with a package called `PGPLOT`, and may be directed to graphics terminals or laser printers. You must `@DISK$TYCHO:[ATLAS.PGPLOT]PGLOGIN.COM` to initialize `PGPLOT` before running `PLOTATLAS`.

2. Internal Data Format for the Spectrum Atlas

This section describes the atlas format we are using at present. We solicit suggestions for extending or modifying this internal format if necessary to support other applications. The access subroutines described above are intended to provide a format-independent interface to the atlas spectra.

Each spectrum atlas is a set of associated disk files located in a single directory. The atlas is identified by a prefix, e.g. GS is the prefix designating the Gunn-Stryker atlas. There are three parts to each atlas:

1. An **index file** describing the format of the atlas data files.
2. A **wavelength file** containing the wavelength set.
3. A set of **spectrum files** containing fluxes for individual stars.

2.1. Index file

The index file describes the spectrum atlas. Its name is the atlas prefix followed by **ATLAS**, e.g. GS**ATLAS**.DAT is the index file for the Gunn-Stryker atlas. The index is a printable ASCII file. Its first few lines contain several parameters that are read by **LOADATLAS** and that define the format of the wavelength and spectrum files. Following this are free-format lines (one line per spectrum) giving enough information about individual stars in the atlas to allow interactive users to locate specific spectra they wish to use.

The first 6 lines (records) of the index file have a (slightly) constrained format as follows:

1. **TITLE**. a comment line with the atlas name and brief description.
2. **NSPEC**. the number of spectra in the atlas.
3. **NWAVE**. the number of wavelengths per spectrum.
4. **NTITLE**, the number of comment records starting each spectrum file.
5. **NENTRY**. the number of wavelength entries per data record.
6. **FORMAT**, the data format for each record, enclosed in quotes, e.g. '(10E8.3)'. If the data are unformatted (binary), the format is given as ' '. Free-format data is specified by the format '* '.

Comments may appear after the required first word of each of these first records. The remainder of the index file contains comments and other useful information, including one-line descriptions for each star so that users can identify and request the stars by spectrum number.

2.2. Wavelength file

The wavelength set used for each atlas spectrum is given in a file called e.g. GS0000.DAT for the Gunn-Stryker atlas. This wavelength file begins with ASCII comment records: their number NTITLE is given in the index file. Following this are records giving the wavelengths in angstroms in the format that is specified by the index file.

2.3. Spectrum files

Each atlas spectrum is contained in a separate spectrum file. The file names are e.g. GS0001.DAT, GS0002.DAT, GS0003.DAT..... etc. Like the wavelength file, the spectrum files begin with ASCII comment records and follow through with data records in the required format. Each data value is the ST magnitude for the corresponding wavelength. ST magnitudes are related to absolute flux densities by

$$ST = -21.10 - 2.5 * \log_{10} f_{\lambda}$$

for f_{λ} in the units $ergcm^{-2}s^{-1}\text{\AA}^{-1}$.

Appendix B: This is a listing of DISK\$SHARE2:[HORNE.SYNPHOT.DOC]XCAL.DOC
as of 19 March 1986.

The synthetic photometry subroutine library XCAL.TLB
is intended as a general purpose tool for FORTRAN programmers
wishing to do synthetic photometry.
The list given below provides a convenient reference
to the names and argument lists of the available subroutines.

The library is being complemented by several interactive programs
to perform specific kinds of investigations, for example :

- (1) reconstructing bandpasses from known spectra and measured count rates
- (2) fitting model spectra to observed count rates
- (3) computing synthetic observations from filter functions and spectra

The function performed by each subroutine is in most cases obvious
from the name and argument list.

Further explanation of inputs and outputs
may be found in comments at the head of each subroutine.

The source code is held in DISK\$SHARE2:[HORNE.SYNPHOT]XCAL.TLB

Persons wishing to use the subroutine library may do so by linking their
FORTRAN programs with DISK\$SHARE2:[HORNE.SYNPHOT]XCAL/L
and DISK\$SHARE2:[HORNE.LIB]KDHLIB/L,
or by contacting Keith Horne (STScI rm614, X4984).

The subroutine library is incomplete at present,
and is in a phase of rapid expansion as new applications are encountered.
Bug reports, verification checks, complaints, comments, and suggestions
are welcome; please transmit via VAX MAIL to HORNE.
I would especially appreciate knowing whenever anyone
has an application that cannot be easily treated by subroutines
from the package.

The following subroutines and functions are presently implemented:

| | |
|------------|---|
| SUBROUTINE | ABSETUP(NSTAR, N WAVE, STARNAME, WV, AB) |
| FUNCTION | ABUNIT(N WAVE, WAVE, P) |
| SUBROUTINE | APERTURE(CM) |
| FUNCTION | AVGLAM(N WAVE, WAVE, P) |
| SUBROUTINE | BANDCOMP(IUNIT, N WAVE, WAVE, NBAND, TRU, DEF, REC, NAME) |
| SUBROUTINE | BANDPLOT(N WAVE, WAVE, NBAND, TRU, DEFAULT, THRU) |
| SUBROUTINE | POSLIM(N, X, I1, I2) |
| FUNCTION | BARLAM(N WAVE, WAVE, P) |
| FUNCTION | BNU(W, T) |
| FUNCTION | DBNUDT(W, T) |
| SUBROUTINE | COMPRESS(NLIST, LIST, N WAVE, SPEC, NAME) |
| SUBROUTINE | DATA PLOT(N WAVE, WAVE, NSTAR, FNU, NOBS, PRED, DATA, SIGMA, IBAND, ISTAR, NBAND, BAND) |
| SUBROUTINE | EDITFILT(N WAVE, WAVE, FILT, NAME) |
| SUBROUTINE | EDITNAME(NAME) |
| SUBROUTINE | EDITSPEC(N WAVE, WAVE, SPEC, NAME) |
| SUBROUTINE | EVALATLAS(N WAVE, WAVE, SPEC, ATLAS, NUMBER) |
| SUBROUTINE | EVALFILT(N WAVE, WAVE, FILT, NAME) |
| SUBROUTINE | EVALSPEC(N WAVE, WAVE, SPEC, NAME) |
| FUNCTION | EXTMAG(WAVE, EBMV) |
| SUBROUTINE | EXTSPEC(N WAVE, WAVE, SPEC, EBMV) |
| SUBROUTINE | EXTSPECGS(N WAVE, WAVE, SPEC, AV) |
| FUNCTION | FLAMAVG(N WAVE, WAVE, THRU, FLAMBDA) |
| SUBROUTINE | FLAMCAL(N WAVE, WAVE, THRU, NSPEC, FLAMBDA, NOBS, ISPEC, OBSN, SIGMA, RATIO, SIGRATIO, CHISQR) |
| SUBROUTINE | FLAMTOFNU(N WAVE, WAVE, SPEC) |
| SUBROUTINE | FLAMTOMJY(N WAVE, WAVE, SPEC) |
| FUNCTION | FNUAVG(N WAVE, WAVE, THRU, FNU) |
| FUNCTION | FNUAVGANAL(N WAVE, WAVE, THRU, FNU, P1, P2, P3, P4, P5) |
| SUBROUTINE | FNUCAL(N WAVE, WAVE, THRU, NSPEC, FNU, NOBS, ISPEC, OBSN, SIGMA, RATIO, SIGRATIO, CHISQR) |
| SUBROUTINE | FNUTOFLAM(N WAVE, WAVE, SPEC) |
| SUBROUTINE | GENDATA(N WAVE, WAVE, NBAND, THRU, NSTAR, FNU, NOBS, DATA, SIGMA, IBAND, ISTAR) |
| SUBROUTINE | GENFILT(N WAVE, WAVE, NFILT, FILT, NAME) |
| SUBROUTINE | GENSPEC(N WAVE, WAVE, NSPEC, SPEC, NAME) |
| SUBROUTINE | GENWAVE(N WAVE, WAVE) |
| SUBROUTINE | GRABFILT(N WAVE, WAVE, FILT, NAME) |
| SUBROUTINE | GRABSPEC(N WAVE, WAVE, SPEC, NAME) |
| FUNCTION | HYDNU(WAVE, TEMP, TAUO, WAVEO) |
| SUBROUTINE | INITFILT(NFILT, N WAVE, NAME, WAVE, FILT) |
| SUBROUTINE | INITSPEC(NSPEC, N WAVE, NAME, WAVE, SPEC) |
| SUBROUTINE | LINTERP(NORG, XORG, YORG, NPIX, XDATA, YDATA) |
| SUBROUTINE | LOADATLAS(N WAVE, WAVE, SPEC, ATLAS, NUMBER) |
| SUBROUTINE | LOADFILT(NPTS, WAVE, SPEC, NAME) |
| SUBROUTINE | LOADSPEC(NPTS, WAVE, SPEC, NAME) |
| SUBROUTINE | LOOKUP(NPIX, DATA, TARGET, ILO, IHI, PART) |
| SUBROUTINE | MJYTOFLAM(N WAVE, WAVE, SPEC) |
| FUNCTION | OPHYD(WAVE, TEMP, Z) |
| SUBROUTINE | PICKLIST(MAXITEM, MAXLIST, NLIST, LIST) |

```
SUBROUTINE PICKONE( MAXITEM, ITEM )
SUBROUTINE PICKRANGE( MAXITEM, MAXSPAN, ITEM1, ITEM2 )
FUNCTION PIVLAM( NWAIVE, WAVE, P )
FUNCTION POWERLAW( WAVE, FNUREF, WREF, ALPHA )
FUNCTION RMSLAM( NWAIVE, WAVE, P )
SUBROUTINE SAVEFILT( NWAIVE, WAVE, NBAND, THRU )
SUBROUTINE SELECT( PROMPT, NLIST, NAME, NSELECT )
SUBROUTINE SHOWBAND( IUNIT, IBAND, NWAIVE, WAVE, THRU, NAME )
SUBROUTINE SHOWNAME( IUNIT, NCOL, NTOTAL, NAME )
SUBROUTINE SIMPLOT( NDATA, XDATA, YDATA, XLABIN, YLABIN, TITLEIN )
FUNCTION STAR( WAVE, TEMP )
FUNCTION DSTARDT( WAVE, TEMP )
FUNCTION STIMCOR( WAVE, TEMP )
SUBROUTINE STMAGTOMJY( NWAIVE, WAVE, SPEC )
FUNCTION STUNIT( NWAIVE, WAVE, P )
SUBROUTINE VMSFILE( FILE, L1, L2 )
SUBROUTINE WORD1( COMMAND, L1, L2 )
```

