The IUE High Dispersion Spectra Processing in IRAF

Petr Škoda
Astronomical Institute, Academy of Sciences of the Czech Republic

December 1, 1998

Abstract

The IUE high dispersion spectra available in NEWSIPS MXHI format of IUE final archive can not be directly analyzed in IRAF. A number of different packages like IUEDR (part of Starlink) or IUERDAF (written in IDL) has usually been used for this purpose. However, with the combination of external IRAF package `iuetools` and several home-written `cl` scripts a batch analysis of a large number of spectra can be performed. We describe in detail the whole process of obtaining the data, conversion to IRAF format, echelle orders merging and continuum fitting up to the measurements of radial velocity of selected resonance lines. We also give a hints for rapid preparing of a convenient plot of line profile series ordered by time and orbital phase. In conclusions, a brief discussion of advantages and bottlenecks of IRAF approach in regards of other packages is given.

1 Introduction

The spectra obtained by IUE satellite during its long life mission are one of the important sources of information about hot stars. So any astronomer working in the field of Be and WR stars should be able to work with the data in the simple manner without necessary detailed study of instrumental setup, image distortions, camera aging effects etc. For that purposes the best way is taking the NEWSIPS absolutely calibrated data available as MXHI (high dispersion) and MXLO (low dispersion) files. The complete description of the structure of these files is given in the Chapter 12 of the *IUE NEWSIPS Information Manual v 2.0* [1]

2 Obtaining data

The NEWSIPS data may be found at two places - the data processed at Goddard on the NASA server NDADS and some of them on ESA INES server[2] - there is, however, only a part of all IUE spectra available - e.g. for CX Dra there were only 7 from 102 available at NASA. The spectra in ESA archive are reprocessed in a new way (improved extraction, echelle order splicing etc.). The spectra of a particular star (not only from IUE but other missions too) can be easily found using WISARD engine[3] For the already known image number it is better to use direct browser[4] of MXLO and MXHI files. The data retrieval itself is done by filling in the form for jukebox server including e-mail address. After some time the relevant data are copied to the staging disk area and a e-mail notification is sent. You may also fill in the information about access to your ftp server and the data from NDADS are automatically downloaded there. The other possibilities how to contact NDADS may be found on this web page[5].

3 Conversion to IRAF formats

As the most important and most complicated data files are the high dispersion spectra, we will describe only MXHI files handling and only SWP camera data. Basic options for IUE data processing can be found here:[6]. For data reduction in IRAF the *iuetools* package[7] must be installed (it requires STSDAS and TABLES packages v 2.0 or later). It consists of two tasks: `mxexpand` takes the MXHI file and produces FITS with the binary table extension. The trick is to take the MXHI information about the starting pixel number and corresponding wavelength and wavelength increment for each echelle order and create from this information another table column called `Wave_Array`. This is a one dimensional array containing explicit wavelength for each valid data

[1] `http://archive.stsci.edu/iue/manual/newsips/`
[2] `http://ines.vilspa.esa.es/ines/`
[3] `http://hypatia.gsfc.nasa.gov/wisard/wisard.html`
[4] `http://hypatia.gsfc.nasa.gov/iue/browser_select.html`
[5] `http://archive.stsci.edu/iue/retrievals.html`
[6] `http://archive.stsci.edu/iue/da_sw.html`
[7] `http://ra.stsci.edu/newsips/`

pixel. Information for each order is stored in one row of binary table. The new FITS file may be already used for data processing in STSDAS (e.g plotting with `sgplot` or `igi`), but the majority of users will probably know much better the basic IRAF packages for CCD data analysis in multispec format (`splot, continuum, specplot`). Thus the FITS file has to be converted by the `mxtomultispec` task to the `*.imh` multispec file with each order in separate aperture line. Only the absolutely calibrated fluxes are extracted. Each aperture has the dispersion polynomial fitted and written into WCS header. This file may be already analyzed in `splot` looking at each echelle order as different aperture/beam — the line profiles may be fitted, equivalent width and line centroid measured etc.

Several remarks to the `iuetools` tasks: `mxexpand` allows bulk file processing using @-lists or image templates (asking for the output file directory), however the wrong behavior was observed after several images were converted. It probably occurs after internal buffers are exhausted — then the task begins producing incorrect files (errors with HDU closing are reported). The `mxtomultispec` does not support either image templates, or @-lists. So we used following simple script for bulk image processing. The script called `mxconvert` will take the image template (for MXHI files) and for each file it calls `mxexpand` with output redirected to the current directory. Than it calls `mxtomultispec`. The temporary files (for each echelle order) are sent to /tmp directory. The script must be called as a IRAF task defining

```
task mxconvert=/path/mxconvert.cl
```

Then you may simply call `mxconvert *` and the converted files will be created in the current directory with extension `*.fits` and `*.imh` respectively . For correct run of the script the input MXHI files must have the `.mxhi` extension (e.g. `swp51041.mxhi`), so that the output files are `swp51041mxhi.fits` and `swp51041.imh` respectively.

```
procedure mxconvert(inputfiles)
#converts MXHI data to TABLES readable FITS and multispec format
# v 1.0    15.6.1998
# Petr Skoda - skoda@sunstel.asu.cas.cz
file inputfiles {"", prompt="input MXHI files"}
struct *inpfilstruc
begin
int sl   #string length
file infile,outfile,rootfile,inpfileslist #working files
#
inpfileslist=mktemp("tmp$imlist")
sections(inputfiles,option="fullname",>inpfileslist)
inpfilstruc=inpfileslist
while (fscan(inpfilstruc,infile)!=EOF)
{
sl=strlen(infile)
if (substr(infile,sl-4,sl)==".mxhi")
```

```
   rootfile=substr(infile,1,sl-5)
else
   {
   print("input filename error")
   break
   }
mxexpand (infile, directory=" ", wvarr_col="Wave_Array")
infile=rootfile//"mxhi.fits"
outfile=rootfile
mxtomultispec (infile,outfile,
 flux_col="default", wave_col="Wave_Array", verbose=no, tempdir="tmp$",
 dout_list="")
}
#end of file processing
delete(inpfileslist,ver-,>&"dev$null")
end
```

4 Merging echelle orders

The analysis of each order separately with `splot` may be useful for single line investigation, however the wavelength coverage of one order is very small (about 20 Å) and the increased noise at both ends (caused by ripple correction) will degrade the useful part even more. The order splicing is thus necessary. We decided for the sake of simplicity to splice all orders in the SWP range in one long spectrum. The catch is that the data in MXHI files are linearized in wavelength but each echelle order is rebinned with different wavelength difference interval. As the IRAF world coordinate system (WCS) requires only one wavelength increment for the whole spectrum, the flux data must be resampled — it may introduce some minor error in the position of sharp spectral features. We have, however, come to the conclusion that the influence of resampling errors on the radial velocity measurements is negligible in comparison with other effects like the uncertainity of line profile fitting due to the strong noise, aperture pointing errors etc. The Fig. 1 shows the zoomed area in the core of SiIV line.

For order merging we have adopted the method by Solano[8]. He has shown that in the overlapping region of orders m and $m-1$ of SWP camera the S/N ratio is is higher in one third of it. So the data are cut at wavelength λ_c

$$\lambda_c = \lambda_s + (\lambda_e - \lambda_s)/3 \qquad ,$$

where λ_s is the starting wavelength of order $m-1$, λ_e the ending wavelength of the order m. Problematic are the orders with $m \leq 72$ which do not overlap and the gap between data is thus prominent.

We have developed the cl task `mxmerge` which takes the multispec files `*.imh`, copies the selected range of wavelengths (about 0.03 Å smaller at each side to avoid the overlapping uncertainity) for each order into separate files

[8] `http://ines.vilspa.esa.es/ines/docs/esm.pdf`

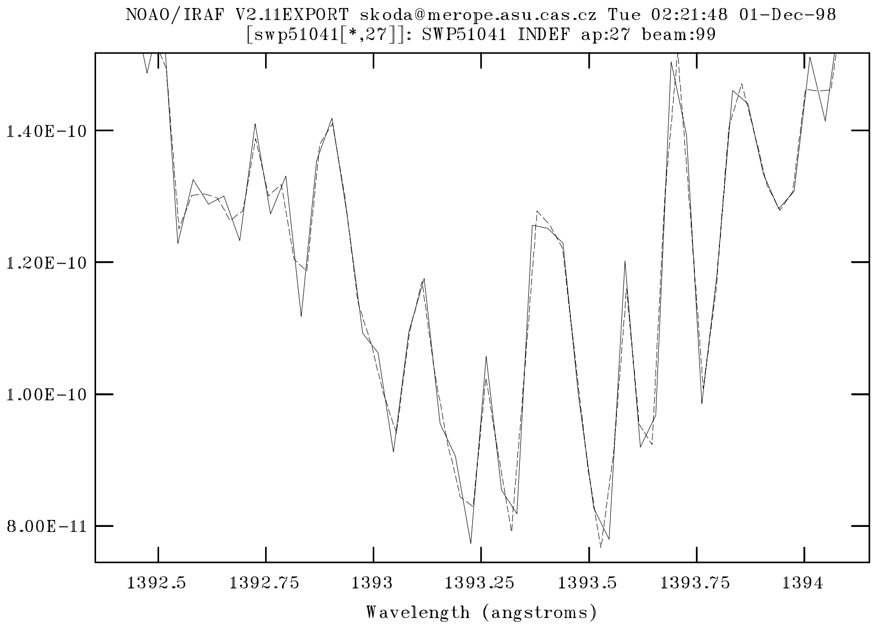

Figure 1: Zoomed area of SiIV line - dashed line shows the efect of linearization in spliced spectrum

using `scopy` and them merges these separate files into one long spectrum using `scombine`. The output prefix for the merged filename is asked. Suppose that the multispec file is called `swp51041.imh`. Using the `''all''` prefix the output name of merged file file will be `allswp51041.imh`.

```
procedure mxmerge(inputfiles,outputprefix)
# merges multispec IUE spectra converted by mxexpand
# and mkmultispec by the
# order-cutting and catenating  method of Cassatela
#
# v 1.0  20.5.1998  Petr Skoda   skoda@sunstel.asu.cas.cz
#
#
file inputfiles {"", prompt="input multispec files"}
string outputprefix {"all",prompt="merged file prefix"}
struct *inpfilstruc
begin
int m,n  #order numbers
int sl   #string length
real lcut1,lcut2  #cut wavelengths
file infile,outfile,orderfile,inpfileslist #working files
string opref  #output file prefix
#
```

```
inpfileslist=mktemp("tmp$imlist")
sections(inputfiles,option="fullname",>inpfileslist)
inpfilstruc=inpfileslist
opref=outputprefix
#
while (fscan(inpfilstruc,infile)!=EOF)
{
sl=strlen(infile)
if (substr(infile,sl-3,sl)==".imh")
   infile=substr(infile,1,sl-4)
orderfile="order"//infile
outfile=opref//infile
print(infile,"  ",orderfile,"  ",outfile)
#order  cuts calculation
lcut2=1150.0
for (m=120; m >= 73; m -= 1) {
lcut1=lcut2
lcut2=24.3952+132875.4838/m+325840.9715/m/m
print (m,lcut1,lcut2)
scopy (infile, orderfile,w1=lcut1+0.03, w2=lcut2-0.03,\
  apertures="",beams=m, apmodulus=0, format="onedspec",\
  renumber=no, offset=-5, clobber=yes,\
  merge=yes, rebin=no, verbose=yes)
}
# non overlapping regions
for (n=72; n>=70; n -= 1)
{
if (n==72)
{
lcut1=lcut2
lcut2=1931.6
}
else if (n==71) {
lcut1=1932.8
lcut2=1958.0
}
else if (n==70) {
lcut1=1961.4
lcut2=1979
}
print (n,lcut1,lcut2)
scopy (infile, orderfile,w1=lcut1, w2=lcut2,  apertures="",\
 beams=n, apmodulus=0, format="onedspec", renumber=no,\
 offset=-5, clobber=yes, merge=yes,\
 rebin=no, verbose=yes)
}
# spliced file preparation
```

```
scombine (orderfile//"*",
outfile, noutput="", logfile="STDOUT", apertures=" ", group="all",
combine="average", reject="none", first=no, w1=INDEF, w2=INDEF, dw=INDEF,
nw=INDEF, , scale="none", zero="none", weight="none", sample="",
lthreshold=INDEF, hthreshold=INDEF, nlow=1, nhigh=1, nkeep=1, mclip=no,
lsigma=3., hsigma=3., rdnoise="0.", gain="1.", snoise="0.", sigscale=0.1,
pclip=-0.5, grow=0, blank=0.)
imdel (orderfile//"*", yes, verify=no, default_acti=yes)
}
#end of file processing
delete(inpfileslist,ver-,>&"dev$null")
end
```

5 Fitting continuum and spikes removal

For some purposes (e.g. study of line profile changes) it is convenient to normalize the spectra to the common continuum level. As it is quite difficult to achive this in some spectral regions due to the line blanketing, we have to estimate some smooth curve along the whole spliced spectra. The IRAF task `continuum` fits well. We have found an optimal setting with spline3 curve of order 5, high_rej=0 and low_rej=2. The continuum sample regions have to be selected manually panning through the zoomed view, however the correct continuum fiting is a little bit magic playing with the spline parameters and sampling region selection. For shorter spectral regions, where the change of the slope of the continuum is small, the normalizing may not be so useful.

The removal of the bad points is little complicated, fortunately the most annoying features like fiducial marks and cosmic ray hits have the shape of a sharp and deep or high spike. So we may use the `continuum` task again, now the order of spline must be very high (about a 30) and the sigma rejection boundaries about 4 (both low and high rej.). It is convenient to copy first just selected region with interesting lines by `scopy` to the new file (i.e. `scopy` with `w1` and `w2` set to boundaries) and then use the spline fitting on this much shorter and smoother part of spectra.

6 Examining interesting lines

Standard `splot` fitting routines like Gaussian, Lorentzian or Voigt may be used for fitting line profiles, estimating the position of the line centers etc. Some boxcar smoothing is also helpful.

7 Plotting sets of spectra

The nice plot of stacked spectra labeled e.g. with the binary phase can be achieved easily using the powerful IRAF task `specplot`. Several plots may be placed at one paper sheet using `gkimosaic`. We'll show an example of batch processing of 64 SWP spectra of Be star CX Dra. Each 8 spectra will be stacked in one plot and labeled with corresponding phase. Finally 4 plots are placed at one page. For this we first have to prepare several lists with image numbers and corresponding phases. We may use the `catfits` task from TABLES package to extract relevant information from FITS file (produced by `mxexpand`) headers. We may use the following format to produce the exposure list:

```
IMAGE  -14.14
LDATEOBS -10.10
LTIMEOBS -10.10
LEXPTIME -10.10
LJD-MID  -15.15
LHELCORR -9.9
LRADVELO -5.5
```

Then we get the ASCII list like this (only part shown):

```
IMAGE  LDATEOBS  LTIMEOBS LEXPTIME LJD-MID        LHELCORR LRADV
8599   30/03/80  10:58:20 434.865  2444328.95969  -.00045  6.92
8634   02/04/80  19:03:02 419.710  2444332.29620  -.00037  4.59
9981   01/09/80  15:02:17 329.598  2444484.12849  0.00111  -2.29
```

From this the phases can be calculated using the `LJD-MID` and `LHELCORR` column by e.g. `awk` script and sorted according to phases. Then a simple file list (say `imbyphase`) of image numbers and phases value (say `phasebyphase`) — each value on one line — is produced. This is then cut into several files per 8 numbers using e.g. UNIX `csplit` command. After such a preparation we have (in our example): 8 files `imbyphase0`–`imbyphase7` and 8 files `phasebyphase0` – `phasebyphase7`. Now comes the powerful `specplot` task. Let us have the normalized cleaned spectral region of the SiIV dublet stored by `scopy` in files `clnsi4swp*`. Then the following `epar` settings of `specplot` will produce the desired stacked plot for first 8 spectra:

```
PACKAGE = onedspec
   TASK = specplot
spectra = cleansi4swp//@imbyphase0  List of spectra to plot
(apertur=                 ) Apertures to plot
(bands  =                1) Bands of 3D images to plot
(autolay=               no) Use automatic layout algorithm?
(autosca=               no) Scale to common mean for automatic layout?
(fractio=               1.) Fraction of automatic minimum separation step
(units  =                 ) Coordinate units
(scale  =               1.) Intensity scale (value, @file, keyword)
(offset =               0.) Intensity offset (value, @file, keyword)
(step   =             -0.7) Default separation step
(ptype  =                1) Plotting type
(labels =             user) Type of labels
(ulabels=phasebyphase0    ) User labels (file)
(xlpos  =             1.02) X label position (fraction of range)
(ylpos  =               0.) Y label position (fraction of mean)
```

```
(sysid  =              no) Include system banner and step value?
(yscale =              no) Draw Y axis scale?
(title  = CX Dra: Si IV line 1393.775 A ) Plot title
(xlabel =                ) X axis label
(ylabel =           phase) Y axis label
(xmin   =           INDEF) X axis left limit
(xmax   =           INDEF) X axis right limit
(ymin   =           INDEF) Y axis bottom limit
(ymax   =           INDEF) Y axis top limit
(logfile=                ) Logfile
(graphic=        stdgraph) Graphics output device
(cursor =                ) Cursor input
(mode   =              ql)
```

The similar plots may be prepared for other spectral regions (using another file prefix) and using the graphics redirection operator `>G` the graphics metafiles `*.mc` containing each picture are produced.

```
string imroot
specplot.step=-0.6
specplot.title="CX Dra: Si IV lines 1393.755 A and 1402.769 A"
set imroot="clnsi4swp"
specplot imroot$//@imp0 ulabel=php0 >G si4_0.mc
specplot imroot$//@imp1 ulabel=php1 >G si4_1.mc
specplot imroot$//@imp2 ulabel=php2 >G si4_2.mc
specplot imroot$//@imp3 ulabel=php3 >G si4_3.mc
specplot imroot$//@imp4 ulabel=php4 >G si4_4.mc
specplot imroot$//@imp5 ulabel=php5 >G si4_5.mc
specplot imroot$//@imp6 ulabel=php6 >G si4_6.mc
specplot imroot$//@imp7 ulabel=php7 >G si4_7.mc
#
specplot.step=-0.8
set imroot="clnal3swp"
specplot.title="CX Dra: Al III lines 1854.716 A and 1862.790 A"
specplot imroot$//@imp0 ulabel=php0 >G al3_0.mc
specplot imroot$//@imp1 ulabel=php1 >G al3_1.mc
specplot imroot$//@imp2 ulabel=php2 >G al3_2.mc
specplot imroot$//@imp3 ulabel=php3 >G al3_3.mc
specplot imroot$//@imp4 ulabel=php4 >G al3_4.mc
specplot imroot$//@imp5 ulabel=php5 >G al3_5.mc
specplot imroot$//@imp6 ulabel=php6 >G al3_6.mc
specplot imroot$//@imp7 ulabel=php7 >G al3_7.mc
```

Now we can fit several pictures to one page using the `gkimosaic` task and send this to the default postscript printer:

```
gkimosaic si4_[0-3].mc nx=2 ny=2 device=stdplot
```

The result is in Fig. 2.

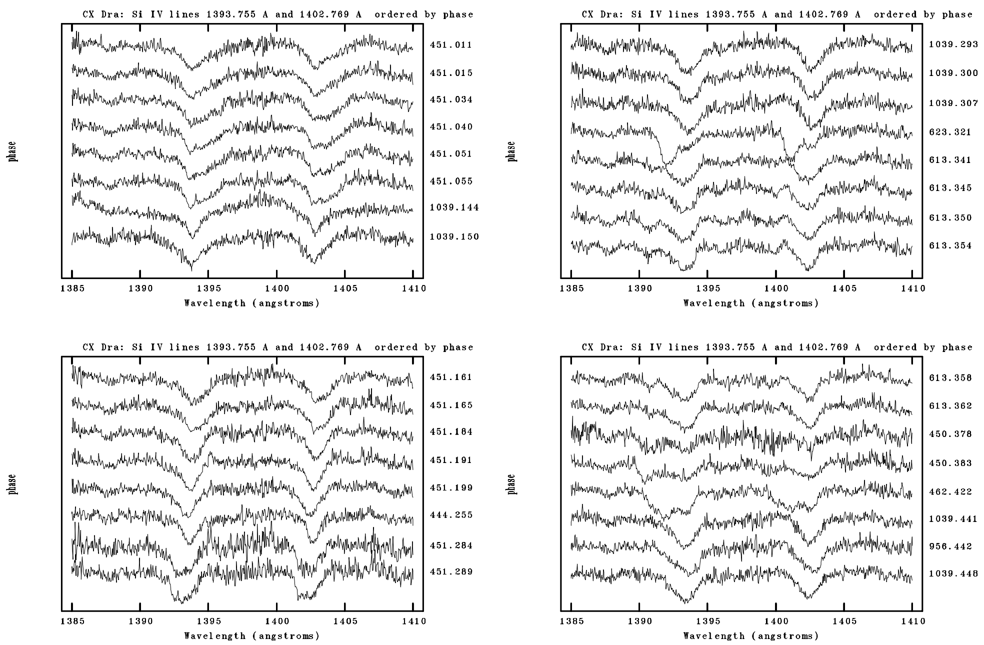

Figure 2: Stacked plot of 4 graphs using gkimosaic

8 Transformation to RV scale

The last interesting operation may be the transformation of spectral region to the radial velocity scale with the origin at particular laboratory wavelength. The transformation will be done by the `mkv.cl` script written by Frank Valdes and taken from the *adass.iraf.application* newsgroup. Here it is little modified to take only the desired part of spectra between wavelengths $wx1$ and $wx2$ which are together with the laboratory wavelength w stored in file `silist`

```
procedure mkv (input,lines,output)
file input {prompt="input image"}
file lines {prompt="file of wavelengths"}
file output {prompt="output name"}
begin
 int i
 string in,out,ap,w,v0,dv
 real wx1,wx2
 in=input
 out=output
 list=lines
 i=0
 while(fscan(list,w,wx1,wx2)!=EOF) {
 i=i+1
 scopy(in,out,w1=(wx1),w2=(wx2),aperture="",beams="",
```

```
 format="multispec",renumber=yes,offset=0,clobber=yes,
 merge=no,rebin=no,verbose=yes)
 print out//i
 v0="((crval1/"//w//"-1)*2.998e5)"
 dv="(cdelt1/"//w//"*2.998e5)"
 hedit(out,"crval1",v0,update=yes,show=yes,verify=no)
 hedit(out,"cd1_1",dv,update=yes,show=yes,verify=no)
 hedit(out,"cdelt1",dv,update=yes,show=yes,verify=no)
 hedit(out,"WAT1_001", 'wtype=linear label="radial \
      velocity"  units=km/s''',\
 update=yes,show=yes,verify=no)
 }
 list=""
end
```

Using the simple rest wavelenght list `silist` as such :

```
1393.755 1388.5 1398.5
```

and calling it by

```
mkv clnsi4swp//@imbyphase0 %clnsi4swp%rv1si4swp%//@imbyphase0 silist
```

we finally get after processing by `specplot` and `gkimosaic` (as shown above) the stacked plots in radial velocities as shown in Fig. 3.

9 Conclusions

There are indications, that the NEWSIPS extraction routine introduced into final spectra some systematic errors[9] — mainly for high dispersion SWP data, so for the precise measurements (especially changes in the line depth of order of several percent) the reextraction of the spectra should be done from SIHI files. (They may be previewed also through SI_browser[10]. This is, however, out of scope of this article . We were not able to find any routine capable of such a task in Starlink IUEDR. In IUERDAF software there is only a beta version of a procedure `hgex.pro`, but it unfortunately didn't work for our particular data. Thus the only hope for reprocessing the IUE data from scratch remains the IRAF port of NEWSIPS routines which was announced for the Spring 1998[11] , but it is still not here.

We have shown that using the combination of IRAF scripts and its powerful tasks the customized data files and nice overview plots can be produced for large number of spectra easily. In comparison to the IUERDAF software, there ramain some doubts about the precision of linear resampling and the most frustrating is the lack of a quality flags handling in present IRAF which would enable the correct removal of all problematic data marked by some quality flag.

[9] `http://hypatia.gsfc.nasa.gov/iue/seville.html`

[10] `http://hypatia.gsfc.nasa.gov/iue/sibrowse/SI_browse_info.html`

[11] `ftp://ra.stsci.edu/newsips/newsips.ps.gz`

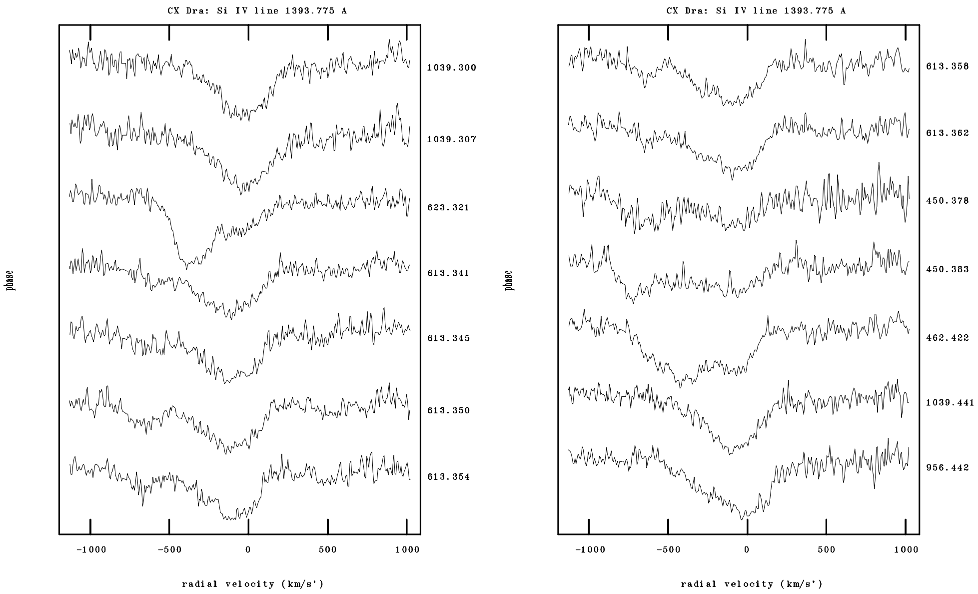

Figure 3: Stacked plot in radial velocity scale

Such a task can be done in IUERDAF by `patch` routine. On the other side the processing using IRAF can be done quickly with the strong interactive controll (e.g. `specplot` task) using only software available free of charge in contrast to the quite expensive IDL system. This article was meant as a quick cookbook for a simple processing of IUE data. Any astronomer with basic IRAF experiece should be then able to produce necessary outputs for further spectral analysis very easily.

If you find this document and accompanying scripts useful, please report your experience with IUE data reduction as well as other suggestions, remarks and/or criticism at author's address `skoda@sunstel.asu.cas.cz`.