Radial Velocity Measurements with IDENTIFY

Francisco Valdes

IRAF Group - Central Computer Services
National Optical Astronomy Observatories††
P.O. Box 26732, Tucson, Arizona 85726
August 1986
Revised August 1990

ABSTRACT

The IRAF task **identify** may be used to measure radial velocities. This is done using the classical method of determining the doppler shifted wavelengths of emission and absorption lines. This paper covers many of the features and techniques available through this powerful and versatile task which are not immediately evident to a new user.

August 8, 1990

Radial Velocity Measurements with IDENTIFY

Francisco Valdes

IRAF Group - Central Computer Services
National Optical Astronomy Observatories††
P.O. Box 26732, Tucson, Arizona 85726
August 1986
Revised August 1990

1. Introduction

The task **identify** is very powerful and versatile. It can be used to measure wavelengths and wavelength shifts for doing radial velocity measurements from emission and absorption lines. When combined with the CL's ability to redirect input and output both from the standard text streams and the cursor and graphics streams virtually anything may be accomplished either interactively or automatically. This, of course, requires quite a bit of expertise and experience with **identify** and with the CL which a new user is not expected to be aware of initially. This paper attempts to convey some of the possibilities. There are many variations on these methods which the user will learn through experience.

I want to make a caveat about the suggestions made in this paper. I wrote the **identify** task and so I am an expert in its use. However, I am not a spectroscopist, I have not been directly involved in the science of measuring astronomical radial velocities, and I am not very familiar with the literature. Thus, the suggestions contained in this paper are based on my understanding of the basic principles and the abilities of the **identify** task.

The task **identify** is used to measure radial velocities by determining the wavelengths of individual emission and absorption lines. The user must compute the radial velocities separately by relating the observed wavelengths to the known rest wavelengths via the Doppler formula. This is a good method when the lines are strong, when there are only one or two features, and when there are many, possibly, weaker lines. The accuracy of this method is determined by the accuracy of the line centering algorithm.

The alternative method is to compare an observed spectrum to a template spectrum of known radial velocity. This is done by correlation or fourier ratio methods. These methods have the advantage of using all of the spectrum and are good when there are many very weak and possibly broad features. Their disadvantages are confusion with telluric lines, they don't work well with just a few real features, and they require a fair amount of preliminary manipulation of the spectrum to remove continuum and interpolate the spectrum in logarithmic wavelength intervals. IRAF tasks for correlation and fourier ratio methods are under development at this time. Many people assume that these more abstract methods are inherently better than the classical method. This is not true, it depends on the quality and type of data.

Wavelength measurements are best done on the original data rather than after linearizing the wavelength intervals. This is because 1) it is not necessary as will be shown below and 2) the interpolation used to linearize the wavelength scale can change the shape of the lines, particularly strong, narrow emission lines which are the best ones for determining radial velocities.

This paper is specifically about **identify** but one should be aware of the task **splot** which also may be used to measure radial velocities. It differs in several respects from **identify**.

---

††Operated by the Association of Universities for Research in Astronomy, Inc. under cooperative agreement with the National Science Foundation.

Splot works only on linearized data; the wavelength and pixel coordinates are related by a zero point and wavelength interval. The line centering algorithms are different; the line centering is generally less robust (tolerant of error) and often less accurate. It has many nice features but is not designed for the specific purpose of measuring positions of lines and, thus, is not as easy to use for this purpose.

There are a number of sources of additional information relating to the use of the task **identify**. The primary source is the manual pages for the task. As with all manual pages it is available online with the **help** command and in the *IRAF User Handbook*. The NOAO reduction guides or cookbooks for the echelle and IIDS/IRS include additional examples and discussion. The line centering algorithm is the most critical factor in determining dispersion solutions and radial velocities. It is described in more detail under the help topic **center1d** online or in the handbook.

2. Method 1

In this method, arc calibration images are used to determine a wavelength scale. The dispersion solution is then transferred to the object spectrum and the wavelengths of emission and absorption lines are measured and recorded. This is relatively straightforward but some tricks will make this easier and more accurate.

2.1. Transferring Dispersion Solutions

There are several ways to transfer the dispersion solution from an arc spectrum to an object spectrum differing in the order in which things are done.

(1) One way is to determine the dispersion solution for all the arc images first. To do this interactively specify all the arc images as the input to **identify**. After determining the dispersion solution for the first arc and quitting (*q* key) the next arc will be displayed with the previous dispersion solution and lines retained. Then use the cursor commands *a* and *c* (all center) to recenter and *f* (fit) to recompute the dispersion solution. If large shifts are present use *s* (shift) or *x* (correlate peaks) to shift, recenter, and compute a wavelength zero point shift to the dispersion function. A new dispersion function should then be fit with *f*. These commands are relatively fast and simple.

An important reason for doing all the arc images first is that the same procedure can be done mostly noninteractively with the task **reidentify**. After determining a dispersion solution for one arc image **reidentify** does the recenter (*a* and *c*), shift and recenter (*s*), or correlation features, shift, and recenter (*x*) to transfer the dispersion solutions between arcs. This is usually done as a background task.

To transfer the solution to the object spectra specify the list of object spectra as input to **identify**. For each image begin by entering the colon command *:read arc* where arc is the name of the arc image whose dispersion solution is to be applied; normally the one taken at the same time and telescope position as the object. This will read the dispersion solution and arc line positions. Delete the arc line positions with the *a* and *d* (all delete) cursor keys. You can now measure the wavelengths of lines in the spectrum.

(2) An alternative method is to interactively alternate between arc and object spectra either in the input image list or with the *:image name* colon command.

2.2. Measuring Wavelengths

(1) To record the feature positions at any time use the *:features file* colon command where *file* is where the feature information will be written. Repeating this with the same file appends to the file. Writing to the database with the *:write* colon command also records this information. Without an argument the results are put in a file with the same name as the image and a prefix of "id". You can use any name you like, however, with *:write name*. The *:features* command is probably preferable because it only records the line information while the database format includes the dispersion solution and other information not

needed for computing radial velocities.

(2) Remember that when shifting between emission and absorption lines the parameter *ftype* must be changed. This may be done interactively with the *:ftype emission* and *:ftype absorption* commands. This parameter does not need to be set except when changing between types of lines.

(3) Since the centering of the emission or absorption line is the most critical factor, one should experiment with the parameter *fwidth*. To change this parameter type *:fwidth value*. The positions of the marked features are not changed until a center command (*c*) command is given.

A narrow *fwidth* is less influenced by blends and wings but has a larger uncertainty. A broad *fwidth* uses all of the line profile and is thus stable but may be systematically influenced by blending and wings. One possible approach is to measure the positions at several values of *fwidth* and decide which value to use or use some weighting of the various measurements. You can record each set of measurements with the *:fe file* command.

(4) For calibration of systematic effects from the centering one should obtain the spectrum of a similar object with a known radial velocity. The systematic effect is due to the fact that the centering algorithm is measuring a weighted function of the line profile which may not be the true center of the line as tabulated in the laboratory or in a velocity standard. By using the same centering method on an object with the same line profiles and known velocity this effect can be eliminated.

(5) Since the arcs are not obtained at precisely the same time as the object exposures, there may be a wavelength shift relative to the arc dispersion solution. This may be calibrated from night sky lines in the object itself (the night sky lines are "good" in this case and should not be subtracted away). There are generally not enough night sky lines to act as the primary dispersion calibrator but just one can determine a possible wavelength zero point shift. Measure the night sky line positions at the same time the object lines are measured. Determine a zero point shift from the night sky to be taken out of the object lines.

3. Method 2

This method is similar to the correlation method in that a template spectrum is used and the average shift relative to the template measures the radial velocity. This has the advantage of not requiring the user to do a lot of calculations (the averaging of the line shifts is done by identify) but is otherwise no better than method 1. The template spectrum must have the same features as the object spectrum.

(1) Determine a dispersion solution for the template spectrum either from the lines in the spectrum or from an arc calibration.

(2) Mark the features to be correlated in the template spectrum.

(3) Transfer the template dispersion solution and line positions to an object spectrum using one of the methods described earlier. Then, for the current feature, point the cursor near the same feature in the object spectrum and type *s*. The mean shift in pixels, wavelength, and fractional wavelength (like a radial velocity without the factor of the speed of light) for the object is determined and printed. A new dispersion solution is determined but you may ignore this.

(4) When doing additional object spectra, remember to start over again with the template spectrum (using *:read template*) and not the solution from the last object spectrum.

(5) This procedure assumes that the dispersion solution between the template and object are the same. Checks for zero point shifts with night sky lines, as discussed earlier, should be made if possible. The systematic centering bias, however, is accounted for by using the same lines from the template radial velocity standard.

(6) One possible source of error is attempting to use very weak lines. The recentering may find the wrong lines and affect the results. The protections against this are the *threshold* parameter and setting the centering error radius to be relatively small.

4. Method 3

This method uses only strong emission lines and works with linearized data without an **identify** dispersion solution; though remember the caveats about rebinning the spectra. The recipe involves measuring the positions of emission lines. The strongest emission lines may be found automatically using the *y* cursor key. The number of emission lines to be identified is set by the *maxfeatures* parameter. The emission line positions are then written to a data file using the *:features file* colon command. This may be done interactively and takes only a few moments per spectrum. If done interactively, the images may be chained by specifying an image template. The only trick required is that when proceeding to the next spectrum the previous features are deleted using the cursor key combination *a* and *d* (all delete).

For a large number of images, on the order of hundreds, this may be automated as follows. A file containing the cursor commands is prepared. The cursor command format consists of the x and y positions, the window (usually window 1), and the key stroke or colon command. Because each new image from an image template does not restart the cursor command file, the commands would have to be repeated for each image in the list. Thus, a CL loop calling the task each time with only one image is preferable. Besides redirecting the cursor input from a command file, we must also redirect the standard input for the response to the database save query, the standard output to discard the status line information, and , possibly, the graphics to a metacode file which can then be reviewed later. The following steps indicate what is to be done.

(1) Prepare a file containing the images to be measured (one per line). This can usually be done using the sections command to expand a template and directing the output into a file.

(2) Prepare a cursor command file (let's call it cmdfile) containing the following two lines.

```
1 1 1 y
1 1 1 :fe positions.dat
```

(3) Enter the following commands.

```
list="file"
while (fscan (list,s1) !=EOF){
print ("no") | identify (sl,maxfeatures=2, cursor="cmdfile",
>"dev$null", >G "plotfile")
}
```

Note that these commands could be put in a CL script and executed using the command

```
on> cl <script.cl
```

The commands do the following. The first command initializes the image list for the loop. The second command is the loop to be run until the end of the image file is reached. The command in the loop directs the string "no" to the standard input of identify which will be the response to the database save query. The identify command uses the image name obtained from the list by the fscan procedure, sets the maximum number of features to be found to be 2 (this can be set using **eparam** instead), the cursor input is taken from the cursor command file, the standard output is discarded to the null device, and the STDGRAPH output is redirected to a plot file. If the plot file redirection is not used, the graphs will appear on the specified graphics device (usually the graphics terminal). The plot file can then be disposed of using the **gkimosaic** task to either the graphics terminal or a hardcopy device.