Tr. J. of Physics
23 (1999) , 271 – 287.

International Workshop:
"Magnetic Activity in Cool Stars"
(ed. O. Demircan)

Spectral Broadening Functions

Slavek RUCINSKI
Canada-France-Hawaii Telescope
Co, P.O.Box 1597, Kamuela, HI 96743, USA
e-mail: rucinski@cfht.hawaii.edu

Received 21 September 1998

Abstract

Many astrophysical phenomena involving velocity fields produce broadening of spectral lines. Frequently, the cross-correlation function (CCF) is used to extract information about this broadening from segments of high resolution spectra. CCF has become the standard tool for handing extraction of radial-velocity and broadening information from high resolution spectra as it permits integration of information which is common to many spectral lines into one function which is easy to calculate, visualize and interpret. However, it can be argued that this is not the best tool for applications such as determinations of metallicities ([Fe/H]), for finding locations of star spots on active stars or to study projected shapes of such distorted stars as contact binaries. For such applications, the proper broadening functions (BF) should be used. Properties of the BF's are discussed in this paper with a stress on the fact that the CCF's are not broadening functions. This note concentrates on the advantages of determining the BF's through the process of linear inversion, preferably accomplished using the Singular Value Decomposition (SVD).

1. Convolution and cross-correlation

This review attempts to have some practical value. Therefore, several examples of practical applications will be given. Because the IDL programming language contains many relevant routines and offers one of the most concise notations, the examples will be given in this language. They will be marked by a symbolic prompt: `IDL>` . We should note that stellar spectra will be treated as simple, one-dimensional vectors.

Convolution is an operation that the nature does for us. We seldom see 'naked" functions; they are usually accessible as convolutions. These could be a convolution with the spectrograph's instrumental profile or with the radial component of the microturbulence velocity field in the stellar atmosphere or a rotational broadening function making spectrum of one star different relative to that of another star. Thus, instead of a function $f(u)$, we observe a function $h(x)$ which is a convolution with a braodening function:

$$h(x) = \int_{-\infty}^{+\infty} f(u)\; g(x-u)\; du = f(x) * g(x)$$

This natural process can be simulated in numerical packages by operations which in IDL take the shape either of a special operator:

```
IDL> h = convol(f,h)
```

or can be obtained by the Fourier-transform multiplication and its inverse:

```
IDL> h = float(fft(fft(f,-1)*fft(g,-1),+1))
```

Cross-correlation is an operation which for real functions differs from the convolution really only in the symmetry of the arguments. For complex functions things are slightly different (real and imaginary parts have different symmetries), but astronomers observe real spectra so we do not have to worry about the mathematical nuances.

$$c(x) = \int_{-\infty}^{+\infty} f(u)\; g(u+x)\; du = f(x) \star g(x)$$

The cross-correlation function (CCF or $c(x)$; note the different asterisk in the symbolic notation) can be computed easily and many software packages provide routines for its calculation:

```
IDL> lag = findgen(201) - 100
IDL> c = c_correlate(f,g,lag)
```

2. Re-sampling of spectra

In most cases, the broadening of spectral lines is due to the Doppler effect. This effect produces broadening which is linear in the $\ln\lambda$ scale. This is because of the equality $\Delta\lambda/\lambda = \Delta\ln\lambda = V/c$. Thus, it is equivalent to say that a spectrum is re-sampled to equal velocity intervals or to the $\ln\lambda$ scale. Re-sampling of the spectrum `Sp`, with its original wavelength scale in the vector `w`, can be accomplished by specifying the starting (initial) wavelength `wSt` and the velocity interval `delV`. If one wants to have `n` elements of the new vector `Sp1` then the commands are:

```
IDL> r = delV/2.997924d+5
IDL> w1 = wSt * (1.d0 + r)^double(findgen(n))
IDL> sp1 = interpol(sp, w, w1)
```

One normally selects `wSt` to coincide with the short wavelength end of the vector `w`. that is at `w(0)`, whereas – not to lose any information – `delV` and `n` can be selected to mildly over-sample the original spectrum.

3. Broadening functions

Suppose we observe a sharp-line ($S(\lambda)$) and a broad-line ($P(\lambda)$) spectra and we want to determine the broadening and other differences which make the latter spectrum more

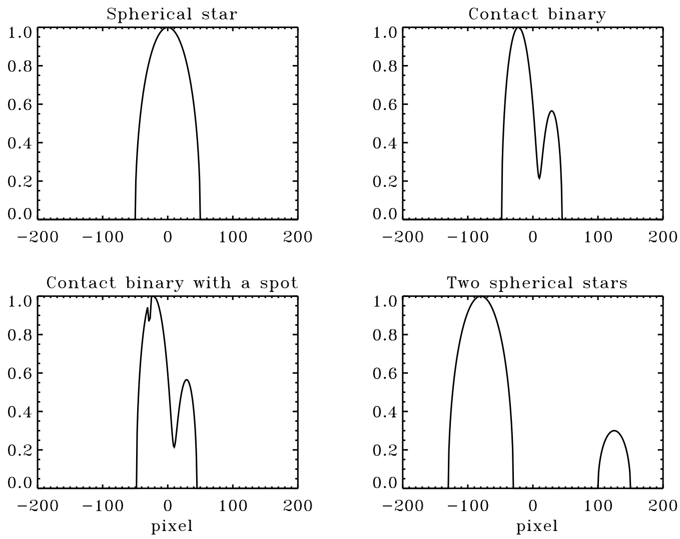


Figure 1. Four schematic examples of broadening functions. The first case (upper left) is the case of a single star rotating uniformly. The case of a contact binary (upper right) illustrates how the shape of the binary is projected into the velocity space. For two detached stars (lower right) the two peaks are disjoint but we still can define a single BF. The star spots correspond to missing flux at certain velocities. Their manifestation would be indentations in the BF (lower left).

interesting than the former. However, even the sharp-line spectrum is not free of some broadening. This can be a thermal broadening of lines or micro-turbulence effects; we call them jointly $T(\lambda)$. Thus, schematically, the sharp-line spectrum can be written as:

$$S(\lambda) = (\sum_i a_i \delta(\lambda_i)) * T(\lambda)$$

while the broad-line spectrum, broadened additionally by $B(\lambda)$, can be written as:

$$P(\lambda) = S(\lambda) * B(\lambda) = (\sum_i a_i \delta(\lambda_i)) * T(\lambda) * B(\lambda)$$

Function $B(\lambda)$ is the broadening function which contains important radial velocity information. Our goal will be to extract this information from the spectrum $P(\lambda)$, utilizing a sharp-line spectrum $S(\lambda)$ for a star selected to be the most similar in spectral type and other charactristics (except rotation) to that of $P(\lambda)$. Representative examples of expected broadening functions are shown in Figure 1.

The most common operation at this stage leading to an estimate of $B(\lambda)$ would be to compute the cross-correlation function (CCF) of both spectra (note the different asterisks for the correlation, $\star$, and the convolution,$*$):

$$\begin{aligned} C(\lambda) &= S(\lambda) \star P(\lambda) \\ &= S(\lambda) \star (S(\lambda) * B(\lambda)) \\ &= T(\lambda) * T(\lambda) * B(\lambda)) \\ &= \mathcal{B}(\lambda) \end{aligned}$$

The result of the cross-correlation operation, a new function $\mathcal{B}(\lambda)$, is not identical to $B(\lambda)$, as it inherits the common natural (thermal, micro-turbulence, instrumental) broadening components from both spectra. Tonry & Davis [1] showed that for those additional components represented by Gaussians, the addition is quadratical, which for these functions really means repeated convolutions.

Thus, CCF cannot give us the same information as the broadening function. But it can give us some approximation of the BF and will remain a useful tool to have some preliminary estimate on the degree of the line broadening. For symmetrical broadening functions, it will remain the simplest tool to determine the radial velocities simultaneously from many spectral lines.

The differences between the BF and the CCF can be seen when the convolution operation is applied to a sharp-line spectrum (Figure 2) and then the resulting spectrum (Figure 3) is subject to the CCF operation. The result of the CCF is obviously different from the BF (Figure 4): The CCF shows negative baseline excursions and, most worryingly, it shows the 'peak-pulling" effect which would lead to an under-estimate of the individual component velocities. While this last problem can be overcome in the case of binary systems by application of the TODCOR technique [2], we clearly see that the CCF is not the BF.

4. Fourier transform de-convolution

Some attempts to determine the broadening functions [3] utilized the well known property of the Fourier transforms of a correspondence between convolutions and multiplications in the two relevant domains. Thus, a convolution:

$$P(\lambda) = S(\lambda) * B(\lambda)$$

transformed with the Fourier transform $\mathcal{F}$ changes into a product of the transforms:

$$\mathcal{F}\{P(\lambda)\} = \mathcal{F}\{S(\lambda)\} \cdot \mathcal{F}\{B(\lambda)\}$$

Therefore, the broadening function can be obtained from:

$$B(\lambda) \simeq \mathcal{F}^{-1}\{\mathcal{F}\{P(\lambda)\}/\mathcal{F}\{S(\lambda)\}\}$$

In practice, this can be done easily as:

```
IDL> b = float(fft(fft(p,-1)/fft(s,-1),+1))
```

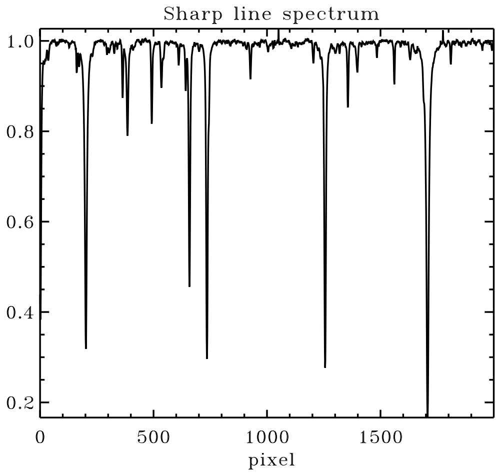


Figure 2. A typical case of a high resolution spectrum of high quality. Such spectra must be re-binned to equal velocity spacing, but – for practical purposes – one uses spectra as one-dimensional functions or vectors defined over a certain number of pixels. Typical spectra have now lengths of 1000 to 4000 pixels.

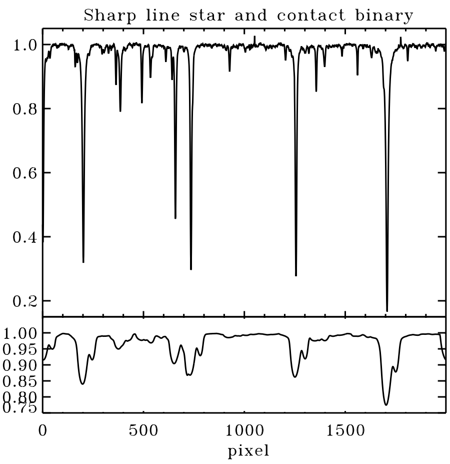


Figure 3. Comparison of a sharp-line spectrum (same as in Figure 2) with the spectrum broadened by a BF for a contact binary (upper right in Figure 1). The broadened spectrum has been obtained here by application of the convolution operator, but normally this would be a result of observations. The examples of various stages of the de-convolution shown later have been obtained by adding Gaussian noise to the broadened function so that the resulting spectrum $P(\lambda)$ would have the signal-to-noise ratio of 100.

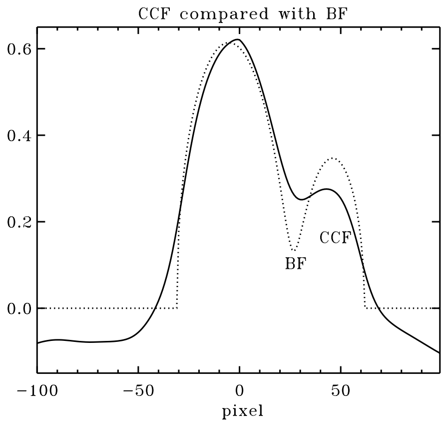


Figure 4. Comparison of the BF used to generate the artificial broadened spectrum in Figure 3 (dotted line) with the cross-correlation function of the sharp and the broadened spectra. Notice that the blending of peaks in the CCF reduces their separation and that the baseline of the CCF is entirely wrong and goes to negative values. The first effect would affect radial velocity measurements and thus determination of stellar masses while the second effect would make the total strength of the CCF wrong, which would be of importance, for example, in line-strength (metallicity) determinations.

While the mathematical background is simple and easy, and the derived function is indeed a broadening function, and not its proxy, the practice of the derivation is not simple at all. The problems are as follows: First, the resulting $B(\lambda)$ spans the whole spectral window, so that one determines a lot of zeroes; there is no 'compression" of information whatsoever. But, more importantly, the division operation usually produces poor results because the high frequency noise becomes amplified by small errors in the denominator in the last expression above. Things may become particularly unpleasant when one or a few frequencies in the sharp-line spectrum are for some reason measured too weak or too strong as this can produce ripples spoiling the final result (Figure 5). Usually, some sort of frequency filtering is applied at this stage to suppress the high spatial frequencies which are most likely to represent the noise. But removal of the high spatial frequencies means a loss of spectral resolution so that the final result may actually depend on the applied filter.

5. Convolution in the formalism of linear equations

There are two main issues that re-casting convolution into a set of linear equations can resolve. These are: (1) How to channel information over the whole spectrum (say 2000 pixel long) into the BF window (say 200 pixels long)? (2) How to utilize all information contained in sharp-line spectra and remove the influence of the noise in the continuum?

The convolution can be written as an over-determined system of linear equations which

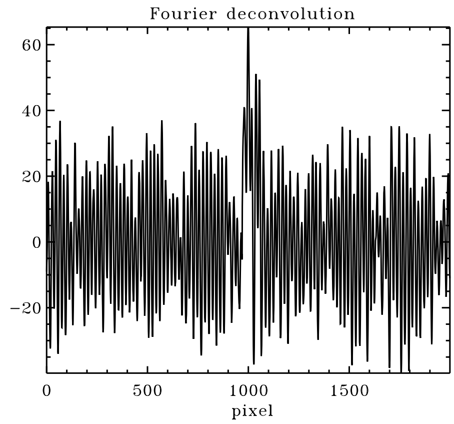


Figure 5. A rather typical case of the Fourier de-convolution. We know that the BF differs from zero over a small interval of some 100 – 200 pixels, yet it is determined over the whole span of the original spectra. In addition, small errors in the sharp-line spectra frequently become amplified in the Fourier-transform division stage. This results in strong fringing of one or several spatial frequencies invalidating the whole process.

link a sharp-line spectrum $\vec{S}$ (n elements), via the broadening function $\vec{B}$ (m elements), with the broadened spectrum $\vec{P}$ (n elements). The mapping is through the 'design matrix" $\widehat{Des}(m,n)$ which is actually formed from the sharp line spectrum $\vec{S}$ by consecutive vertical shifts by one element. This can be done by a proper indexing as in the following IDL routine:

```
function map4,s,m
; shifts vectors s vertically within m
; m - must be odd, n must be even
n = n_elements(s) & t = fltarr(m) # fltarr(n-m+1)
; t(m,n-m) = t(small,large-small) dimensions
for j = 0,m-1 do for i = m/2,n-m/2-1 do t(j,i-m/2)=s(i-j+m/2)
return,t
end
```

An example of using this routine for a 201-pixel long window would be:
`IDL> des = map4(s,201)`
The program spectra must accordingly be trimmed to `n-m+1` with:
`IDL> p = p(m/2:n-m/2-1)`

The system of equations has a familiar form of the over-determined linear set:

$$\boxed{\widehat{Des}}\ \boxed{\vec{B}} = \boxed{\vec{P}}$$

6. The least-squares solution (normal equations)

One of the traditional ways of solving the system of equations above would be to transform it into a system of 'normal" equations. One achieves this by multiplication of both sides by the transpose of the design matrix: $\widehat{Des}^T\,\widehat{Des} = \widehat{Des}^T\,\vec{P}$.

$$\boxed{\widehat{Des}^T}$$

The system shrinks in size from $m \times (n - m)$ to $m \times m$:

$$\boxed{\widehat{Des}^2}\ \boxed{\vec{B}} = \boxed{\widehat{Des}^T\,\vec{P}}$$

and then is solved using any method of linear algebra. As desired, the result is the best one in the least-squares sense and one does achieve compression of the equation set, usually several times, from the size $n - m$ to m. However, this does not mean that the results will be well defined, because of the poor conditioning of the set of normal equations. A excellent approach, permitting to utilize the original rectangular system of mapping equations $\widehat{Des}$, and yet with controlled conditioning, utilizes the method of singular value decomposition. We describe this approach below.

7. Singular value decomposition (SVD)

The SVD technique is beautifully described in the 'numerical techniques Bible" of Press et al. [4]. They describe it as a somewhat magic black box and for most users it is just fine. If you want to learn how the technique *really* works, then the books of Golub & Van Loan [5] and Craig & Brown [6] are probably the best references.

The essence of the SVD is the property that one can represent any matrix by a product of 3 matrices; in our case: $\widehat{Des} = \widehat{U}\,\widehat{W}\,\widehat{V}^T$. These matrices are, the column ortho-normal $\widehat{U}$ and $\widehat{V}$ and the diagonal matrix $\widehat{W}$ (this is really a vector containing the diagonal elements). The ortho-normal property of the columns in $\widehat{U}$ and $\widehat{V}^T$ is that the following products, $\widehat{U}^T\widehat{U} = \widehat{I}$ and $\widehat{V}^T\widehat{V} = \widehat{I}$, give the unity array $\widehat{I}$ (1 on the diagonal). As we will see later, these columns can be interpretted as basis vectors in the vector spaces of the spectra and of the restorable broadening functions (see Figures 6 and 7).

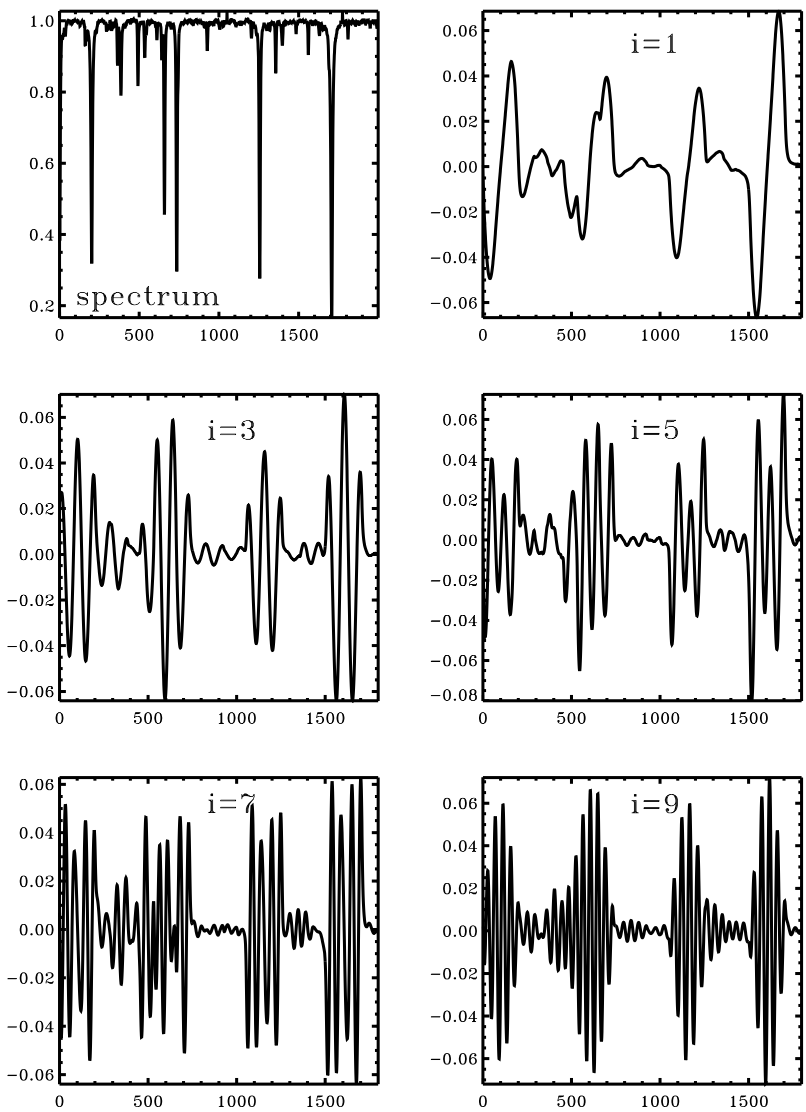


Figure 6. The SVD process involves decomposition of the original spectra into basis vectors which are stored in columns of the matrix $\widehat{Des}$. The same sharp-line spectrum as in the previous graphs is shown here in the upper left panel together with five typical basis vectors (every second one, as indicated).

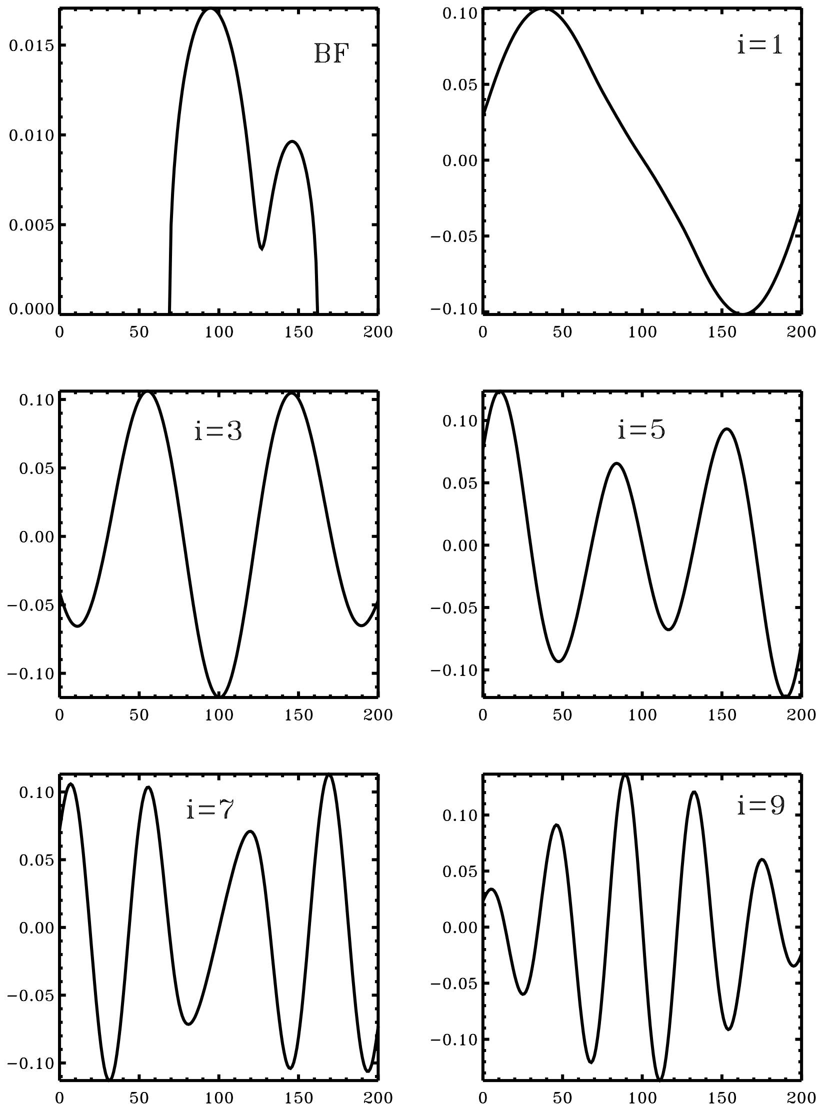
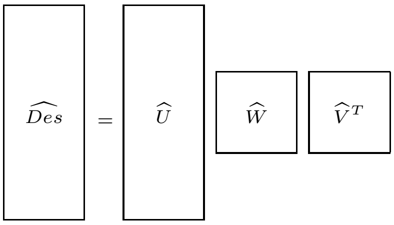


Figure 7. The basis vectors in the space of BF's which can be derived from a given sharp-line spectrum are shown here in the same format as in the previous picture. The first panel shows the BF that we would like to derive. It will be a linear combination of the respective vectors with weights defined by the elements of the diagonal array W.



In IDL, the above operation is performed using:

```
IDL> svdc,des,w,u,v,/double
```

Here, `des` is the input, and the arrays `w,u,v` are produced by the routine as its output. The keyword `/double` is for higher precision and is optional.

One can check the correctness of the operations by the following commands:

```
IDL> wf = fltarr(m,m)
IDL> for i = 0,m-1 do wf(i,i) = w(i)
IDL> des_check = u ## wf ## transpose(v)
```

Because of the properties of the new three matrices, each can be easily inverted. Since $\widehat{U}$ and $\widehat{V}$ are ortho-normal arrays their inverses are just transposes, while the diagonal array $\widehat{W}$ is then replaced by a diagonal array $\widehat{W1}$, with all its elements containing the inverses, $w1_i = 1/w_i$:

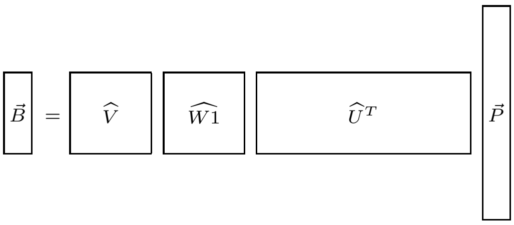
```
IDL> w1 = fltarr(m,m)
IDL> for i = 0,m-1 do w1(i,i) = 1./w(i)
```

The solution is:

$$\vec{B} = \widehat{V}\widehat{W_1}(\widehat{U}^T\vec{P})$$

or

`IDL> b = reform(v ## w1 ## (transpose(u) ## p))` and can be represented by:



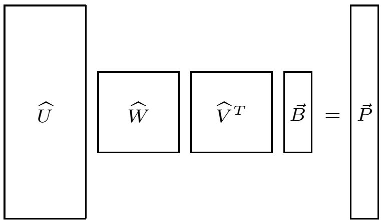
The solution operation is simplified in IDL-4 (and higher versions):

```
IDL> b = svsol(u,w,v,p,/double)
```

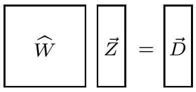
The elements of $\vec{B}$ are all independent, so any – even strange or discontinuous – broadening functions can be restored as no condition imposed on the smoothness of the result. Note that, if only one sharp-line template is used, the decomposition operation svdc is done only once, for possibly many broad-line spectra p, each giving a separate solution b.

8. Columns of the arrays $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$ as vector spaces

Suppose one replaces the original system of equations $\widehat{U}\widehat{W}\widehat{V}^T\vec{B} = \vec{P}$:



by a system in which the diagonal matrix $\widehat{W}$ is the same, but both sides are multiplied by $\widehat{U}^T$:



In such a diagonalized system, the two vectors are given by $\vec{Z} = \widehat{V}^T\vec{B}$ and $\vec{D} = \widehat{U}^T\vec{P}$. One sees that the columns of the arrays U and V form the respective basis vectors in the spaces of spectra and of restorable broadening functions, respectively (see Figures 6 and 7). The solution of the diagonalized system would be then: $\vec{Z} = \vec{D}/\vec{W}$ (in practice, the diagonal of $\widehat{W}$ is a vector, hence $z_i = d_i/w_i$).

9. Advantages and disadvantages of the SVD approach

Although the SVD approach offers mainly advantages over other methods, one should be aware of the positives and negatives of the application. We list them below:

+ The problem can be treated as a set of linear equations; many methods exist for solving them, but the SVD is probably the best here.

+ An 'inverse" of the *rectangular* array $\widehat{Des}$ is possible; normally only square arrays are invertible.

+ The solution of $\vec{B}$ is defined in the least-squares sense (shortest modulus) which is an established method in sciences.

+ The result is a real broadening function and not its substitute.

− One must solve a large system of, say, 2000 equations for 200 unknowns; this may take some time even on large computers.

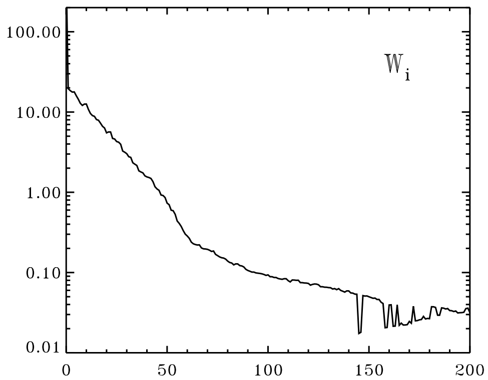


Figure 8. Elements of the diagonal array W for the SVD of the sharp-line spectrum in Figure 2. The solution of the BF determination process that we propose involves use of only a few first elements of this array. This corresponds to discarding of a large amount of basis vectors so that the BF shapes are represented with some loss of resolution. The next figure shows the BF determinations based on only the first 5, 10, 20 or 30 singular values.

– One must know a priori how many unknowns (say 200, or maybe more); the solver must decide this using other methods.

– The results may turn out quite poor, because of the presence of plenty of linearly-dependent equations in the system (parts of spectra where the featureless continuum provides no broadening information); this is related to the limitations in the conditioning of the solution.

The SVD approach offers a simple resolution of the linear dependency problem as *it permits removal of the effects of the continuum in an objective way.* The key element here are the singular values contained in the diagonal of $\widehat{W}$ (Figure 8). Since the solution involves $1/w_i$, small values in w_i spoil the solution. These are exactly those problematic values that one wants to avoid. Thus, by rejecting of small values of w_i, one can (i) remove the linearly dependent equations, (ii) diminish the influence of the noise from the continuum, (iii) reduce the influence of the computer round-off errors (which enter multiplied by the order of the problem) and (iv) reduce the number of the unknowns (because the system is usually not over-determined at all). All these properties are related to the 'conditioning" of the array $\widehat{Des}$. The reader is directed to the source texts on this subject for further reading.

The important factor is $\max(w_i)/\min(w_i)$ which provides information on how many of the singular will be included. In practice, one can keep on adding more w_i and see the successive solutions. The diagonal arrays $\widehat{W}$ and $\widehat{W1}$ will have then elements: $w_i = w_0, w_1, w_2, w_k, ..., w_{m-1}$ and $w1_i = 1/w_0, 1/w_1, 1/w_{k-1}, 0, ..., 0$ with k (call it order of solution) spanning the whole range 0 to $m-1$. In IDL, this can be done by forming a square matrix of solutions:

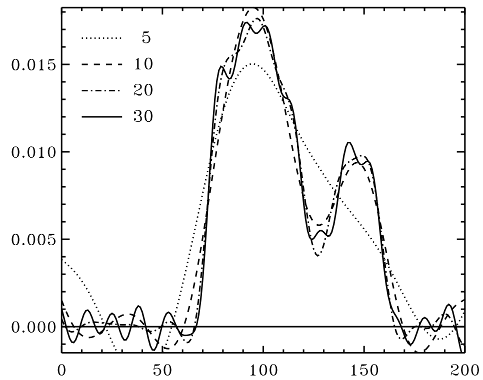


Figure 9. The BF (the same as the dotted line in Figure 3) restored through the SVD process from the broadened spectrum (as in the lower panel of Figure 3, but with an added noise at $S/N = 100$). The power of the SVD is demonstrated by the fact that as few as 10 or 20 first singular values are sufficient for an excellent determination, which is much superior to the cross-correlation function (compare this figure with Figure 4).

```
b = fltarr(m,m)
for i = 0,m-1 do begin
wb = fltarr(m)
wb(0:i) = w(0:i)
b(*,i) = svsol(u,wb,v,p,/double)
end
```

Note that *svsol* does not use the vector of inverses of elements in `w`, but `wb` which is identical to `w` but with the small values replaced by zero.

The array of vectors $\vec{B}$, (`b(m,m)`), contains as its rows the progressively better solutions. We show solutions numbered $k = 5, 10, 20, 30$ for our example case, in which $m = 200$, in Figure 9. As one can see, solutions very quickly become very close to the original broadening function used in creation of the spectrum $\vec{P}$. The full 2-D array `b(m,m)` is shown in Figure 10. Horizontal sections through this contour plot are the individual solutions. The range of good solutions is in our case somewhere between $k \simeq 8$ and $k \simeq 30$; but where is it exactly?

10. How far in k should one go and where to stop?

The essential operation is to plot (usually in log units) the vector $\vec{W}$. There are 3 parts of it: (1) the good, large singular values, (2) the part corresponding to the noise in $S(\lambda)$ and (3) the numerical errors (Figure 8). You may want to stop no further than at the kink below the good part. But the real 'quality control" is the fit. If the error of the fit stops decreasing, you have found the right point. Beyond that point, you will start fitting the

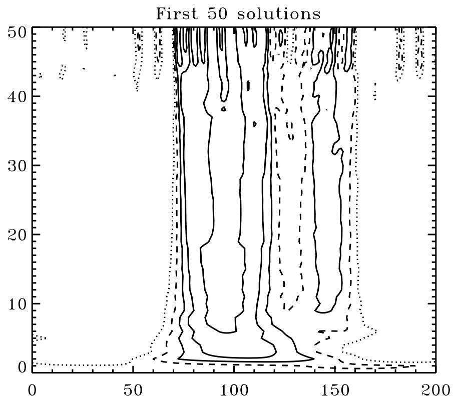


Figure 10. The solutions which use progressively more singular values are shown as horizontal lines in this contour plot (i.e. hhorizontal sections through this 2-D plot would give functions as in the previous figure). Note that the two peaks start appearing at $k \simeq 8$ and that noise becomes strong for $k > 40$. The question is: How many singular values should be used? When should one stop?

noise! However, it is useful to analyze the solutions for different k and see how they first improve and then get worse. Sometimes the fit will remain poor, in spite of the leveling error (versus order of solution i); this means a wrong choice of the sharp-line spectrum or a more complex physics which is not accounted for in the mathematical description of the broadening.

The standard error of the fit can be calculated from:

```
sig = fltarr(m) ; error
pred = des ## transpose(b) ; predicted fits
for i=0,m-1 do sig(i) = sqrt(total((pred(i,*)-p)^2)/m)
```

One should note that stopping too early with a small k is not advisable, as this leads to a loss of resolution, as then not all basis vectors contribute to the solution. Thus, a solution which – in principle – has all elements in $\vec{B}$ independent suddenly acquires inter-element correlations. Thus, it may be advantageous to go to a high i and thus, at first, insure high resolution, and then decrease the noise by smoothing (Figure 12).

One should be aware that the errors may be under-estimated for the case of truncated solutions ($k < m$). While the prescriptions of Rix & White [7] and Rucinski, Lu & Shi [8] are based on the theory of the full SVD, the error analysis for the truncated case has not yet been done. In this situation, it may be advantageous to utilize techniques of the external estimates, such as the bootstrap or Monte Carlo. But this area certainly requires more work ...

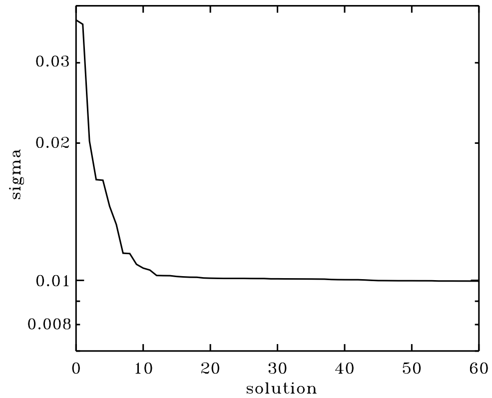


Figure 11. The answer to the question posed in the caption to Figure 10 is in the plot of the quality of fit. For $k \simeq 12 - 15$ the standard error of the fit stops decreasing and this is the place to fix the number of the singular values. Addition of more singular values does increase the resolution, but one then models the noise in the broadened spectrum. In this case the S/N was 100 and the asymptotic error level is indeed 0.01.

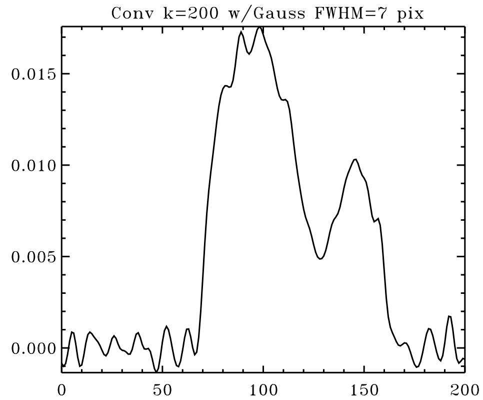


Figure 12. Instead of truncating the solutions at some singular value, one can determine the solution with full possible resolution (i.e. uutilizing all basis vectors in the SVD), and then reduce the large (but uncorrelated!) noise by convolving the result with some smoothing function. This approach has an advantage of providing control over the smoothing process, rather than leaving it to the properties of the basis vectors themselves.

11. Conclusions

The broadening functions can be extracted from broadened spectra using various methods. While the most common approach of deriving the broadening information is through the cross correlation function (CCF), this function is really just a proxy for the proper BF [9]. The CCF does integrate the geometrical information from a spectrum, but retains the natural broadening from the sharp-line template and has undesirable properties in terms of a poor baseline definition and peak-pulling for two components (as in binary stars). The broadening function restoration based on linear equations gives well-defined baselines and is free of the peak-pulling effects. The SVD technique is particularly well suited to solving the linear problem as it offers an elegant way to remove the effects of the featureless continuum though the full control over the conditioning of the system of equations.

References

[1] Tonry, J. & Davis, M. 1979, AJ, 84, 1511

[2] Zucker, S. & Mazeh, T. 1994, ApJ, 420, 806

[3] Anderson, L., Stanford, D. & Leininger, D. 1983, ApJ, 270, 200

[4] Press, W.H., Teukolsky, S.A., Vetterling, W.T. & Flannery, B.P. 1986, Numerical Recipes, Cambridge University Press (this and all other editions)

[5] Golub, G.H. & Van Loan, C.F. 1989, Matrix Computations, 2nd ed. (Baltimore: Johns Hopkins Univ. Press)

[6] Craig, I,J.D. & Brown, J.C. 1986, Inverse Problems in Astronomy, (Bristol and Boston: Adam Hilger Ltd)

[7] Rix, H.-W. & White, S.D.M. 1992, MNRAS, 254, 384

[8] Rucinski, S.M., Lu, W.-X. & Shi, J. 1993, AJ, 106, 1174;

[9] Rucinski, S.M. 1992, AJ, 104, 1968