

Flux Adjustment by Self-Modeling of IUE Images:

An accurate method to extract IUE high resolution spectras

Introduction

It is well known that the fluxes given in the 3rd file of IUE high resolution standard software (IUESIPS) are affected by overlapping of orders that tends to overestimate the background. Ramella et al. (1983) proposed a solution. They used a bi-dimensional deconvolution with filtering and gained that way both a better effective resolution in λ , and, with the narrowing of the orders, a better determination of the background. The choice of the filter used in such a method is very important and the precise knowledge of the Point Spread Function (PSF) is required to derive unambiguous results. Another way is chosen here since the PSF may vary from one image to another, upon parameters such as the exposure time. The method described here yields, as a by-product, a good estimate of the flux in not too heavily saturated spectral regions.

In the following, the reader is assumed to be familiar with IUE high resolution images and the principles of the method employed by standard IUESIPS software (Bohlin et al. 1982), discussed in a preliminary paper by Borsenberger (1983).

Figure 1 illustrates an idealised overlapping situation. Hereafter (figure 2) the x-direction is inclined at $\pi/4$ from the scanning direction (this is for LWR Images, whereas LWP and SWP are inverted left to right to follow the same pattern).

Some other drawbacks may arise with the use of IUESIPS:

- The predicted position of the orders may differ from the actual one, thus increasing the effect of overlap on net fluxes.
- The PSF may vary from the center of an order to the ends, and the fixed-length pseudo-slit used in IUESIPS ignores more flux as it approaches the target edges.

The software proposed here starts from the 2nd file (photometrically corrected image), and does not perform any software rotation so as not to degrade the original information. It uses non linear fitting with precise modeling for pixel illumination.

Let I , a monochromatic signal at wavelength λ be spread along x , following a gaussian law :

$$S_{\lambda}(x) = \frac{I(\lambda)}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-x_0)^2}{2\sigma^2}\right) \quad (1)$$

where x_0 is the position of the center of the order, σ the order width. The unit used is the pixel-diagonal. The contribution of $I(\lambda)$ to the illumination of a pixel lying between j and $j+1$ will be, taking into account the shape of the pixel:

$$C_j = I(\lambda) F_j(\sigma, x_0) = E I(\lambda) \left\{ \frac{\sigma}{2\pi} \left[\exp\left(\frac{-(j-x_0)^2}{2\sigma^2}\right) + \exp\left(\frac{-(j+1-x_0)^2}{2\sigma^2}\right) - 2 \exp\left(\frac{-(j+1/2-x_0)^2}{2\sigma^2}\right) \right] + \frac{1}{2} \left[(j-x_0) \left\{ \operatorname{erf}\left(\frac{j-x_0}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{j+1/2-x_0}{\sqrt{2}\sigma}\right) \right\} + (j+1-x_0) \left\{ \operatorname{erf}\left(\frac{j+1-x_0}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{j+1/2-x_0}{\sqrt{2}\sigma}\right) \right\} \right] \right\} \quad (2)$$

where the constant E stands for pixel efficiency. One could see Borsenberger (1983) for more details here.

Considering now all orders, superscripted (n) affecting the pixel j , its level of illumination L_j should be, if the noise is not taken into account:

$$L_j = \sum_n I_{\lambda}^{(n)} F_j(\sigma^{(n)}, x_0^{(n)}) + C B_k \quad (3)$$

where b_k is the background level, supposed locally constant, and C a constant equal to $E/4$.

On the image we measure the actual illuminations M_j , let G be the sum of squares of the differences.

$$G(I_{\lambda}(n=1\dots n2), \sigma(n=1\dots n2), x_0(n=1\dots n2), b_k(n=1\dots n2)) = \sum_j (L_j - M_j)^2 \quad (4)$$

We will search for the set of parameters (I, b_k, σ, x_0) minimizing G . We will take advantage of the fact that, in practice only 3 orders may be considered at a time, and that σ , b_k , and x_0 vary smoothly. Moreover, I have checked that one can proceed by iteration, treating one order at a time, subtracting from M_j the estimated contributions of adjacent orders.

As the problem is non-linear, we have to search by iteration in a 4-

dimensional space, but two of the variables act linearly (b_k and I). This implies that, once the two non-linear variables are fixed, the two others are completely determined by the least-square condition. That way, we can make a search in a 2-dimensional space, which is done using a gradient algorithm, with parabolic approximation and analytic calculation of partial derivatives.

2/ Strategy

The software proceeds in 2, 3 or 4 phases explained below. The flow-chart is shown in figure 3.

2.1 Find position

As x_0 is a nonlinear parameter, it is necessary to have an estimate of it. A first guess of positions is made visually on a cross-cut located in the center of the image, is smoothed, and used to start minimization. Then the result is used, knowing the mean inclination of orders on the y -direction, to determine the starting positions for the next cross-cut.

As the presence of spikes, microphonics, or other undesirable things (such as broad deep lines, for this particular aim) may lead to misleading position, a weighted polynomial fit of the relation between the order number and its location is made so that rejection of unreliable positions is performed.

Then positions are collected order by order and a weighted polynomial fit with rejection is performed along y for x_0 , σ , and b_k . The presence of "wiggles" on order position imposes the use of high order polynomials (usually 16 terms), while σ is conveniently fitted with 4 terms, and b_k with 6 terms.

In order to reduce the effects of noise this phase uses cross-cuts averaged 4 by 4, projected along y . This affects the position only by second order terms (i.e. local curvature of the order, as first order terms cancel each other, leading to completely negligible error. But as dispersion does not follow exactly the y direction this manipulation tends to slightly overestimate σ . This is usually still negligible, but in the case of very narrow order width, it has to be corrected with the often bypassed second phase described below.

2.2 Determining the precise width

We will now consider the position of orders as fixed, and determine just the width σ . The method is almost the same, except the search is made

in a one-dimensional space and cross-cuts are no longer averaged. As above the results are fitted order by order with a polynomial.

2.3 Determining background

At this time we have a linear problem, as only I and b_k remain unfixed, consequently a direct method is used, and then results are smoothed as above.

2.4 Adjusting fluxes

Up to now, we have implicitly supposed that light of a particular λ spreads perpendicularly to the order, a convenient over-simplification. The two-dimensional pattern involved (PSF) cannot be derived from the data. As the order width varies from an image to another, we may infer that this PSF, determined for some particular observations, is not strictly constant. The less dramatic hypothesis one can make on this point is to assume the PSF to be circularly symmetric. Let us call y_0 the abscissae in the dispersion direction associated with λ_0 , we have for the PSF:

$$S_{\lambda}^{(n)} = \frac{I_{\lambda_0}^{(n)}}{2\pi\sigma} \exp [-(x-x_0^{(n)})^2 + (y-y_0^{(n)})^2] / 2\sigma^2] \quad (5)$$

If we suppose, for the sake of simplicity, that σ is locally constant, and that the orders are perpendicular to cross-cuts, the global influence of an order may be evaluated as follows:

$$B^n(x, y_0) = \frac{1}{\sqrt{2\pi}} \exp [-(x-x_0)^2 / 2\sigma^2] \sum_j \frac{I_j^{(n)}}{\sqrt{2\pi}} \exp [-(y_0 - y_j)^2 / 2\sigma^2] \quad (6)$$

Which is very similar to (1), $I(\lambda)$ being replaced by a sum along the order. We can then proceed the same way as in the preceding phases, apart from the fact we have to wholly compute an order before subtracting its influence on nearby pixels.

Theoretically, this may be applied to the spread of the signal along dispersion, but here the distances are much shorter, the numerical behaviour is completely different, the results are vulnerable to noise (in fact this would be like a de-convolution).

2.5 Wavelength

The software presented here does not re-calibrate λ , it uses the fact that its final step is strictly the same as that of the 3rd file of IUESIPS. It performs correlations with the "Net Not Ripple Corrected" data

of IUESIPS, determines the best fit and subjects it to an eye check and interactive correction (only necessary in particular cases). The corresponding wavelength of IUESIPS are then adopted.

3/ Results

The software has been applied to a lot of IUE images which have permitted us to define more clearly its possibilities.

Figure 4 is an example of a display plotted by SEEFIT, showing the results of phase 1 (MINIUE), with the corresponding polynomial fit (dashed lines). To clarify the figure the position is corrected from the linear term, so that the complexity of the position of an order can be appreciated. This particular case here is far from being extreme, as fluctuations in position increase as one get closer to face plate edges, that is for low order number.

Figures 5 (IUESIPS) and 6 (This software) illustrate the improvement for an intermediate order, on the background determination. The effect would be more important on higher orders, but we do not have up to now sufficiently exposed images at our disposal.

Figures 7, 8, and 9 illustrate the possibility of restoring the fluxes in presence of saturation. Figure 7 is a spectra of HD60178 overexposed (3 min), the saturated regions are hatched on the top of the picture, the use of extrapolated ITF is marked by a square. Figure 8 is the spectrum obtained by the present software with the same image, which coincide with the preceding one in heavy line cores. Figure 9 is a well exposed spectrum (30 s) of the same star, which shows excellent agreement with Figure 8.

This software does not work on too faint orders; these are lost during order tracking if nearby orders are also faint. As a consequence it cannot work on emission line objects, because if peaks are well exposed, orders are too weak. It is moreover necessary, for the order tracking to work well to use a sufficient quantity of orders (minimum 15) and this leads to a significant CPU time (30 min of VAX 11-780). These drawbacks may be dropped in a further version of MINIUE... if I get the time.

4/Machine dependancy

The commented FORTRAN code, written for VAX is available on tape, and should work on any computer with virtual memory (there is no special call to VMS). A complete description in English will be available soon.

Since this software has been built up first on a 16-bit mini-computer (read-only virtual memory is then simulated in FORTRAN to store the image) it is possible to adapt it back to such a machine.

5/Acknowledgements

I am indebted for many fruitful discussions on the project philosophy to C. Catala who presented with patience and cleverness the User point of view. I have to thank Dr F. Praderie and P. Couturier who have made this work possible. It is a pleasure here to thank Dr C. Harvey for help in the correction of the original pidgin English and Dr A. Talavera for the final revision.

J. Borsenberger

Dep. Recherches Spatiales U.A. 264

Observatoire de Meudon 92195 Meudon Cedex FRANCE

References

- Bohlin, R.C., Turnrose, B.E. 1982 "IUE Image Processing News, ESA IUE Newsletter N° 13, p3.
- Borsenberger, J. 1983 "The Origin of Extra Noise", ESA IUE Newsletter N°18, p81.
- Ramella, M. Morossi, C., Allocchio, C., Beckman, J.E., Crivellari, L., Franco, M.L., Molaro, P., Vladilo, G., 1983 "The Procedure in Use at the Trieste Observatory to Extract the Maximum Information from IUE High Resolution Images" IUE ESA Newsletter N°18, p70.

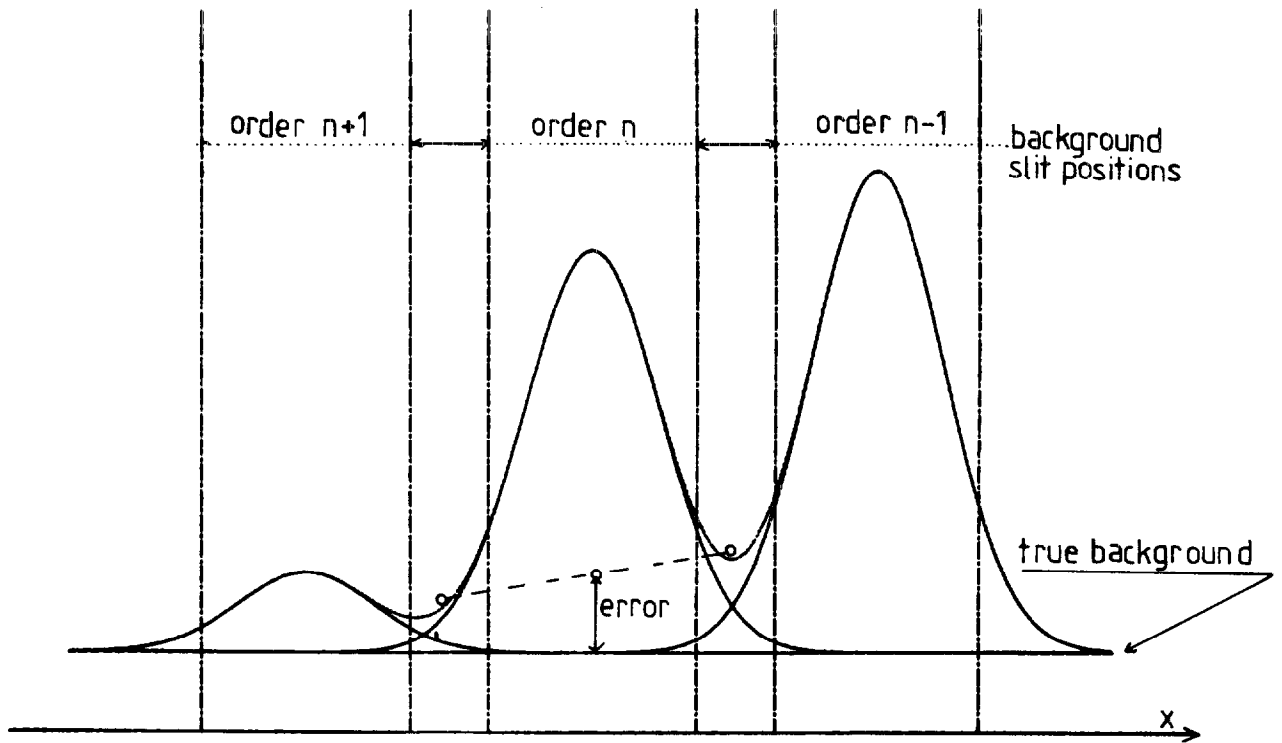


figure 1

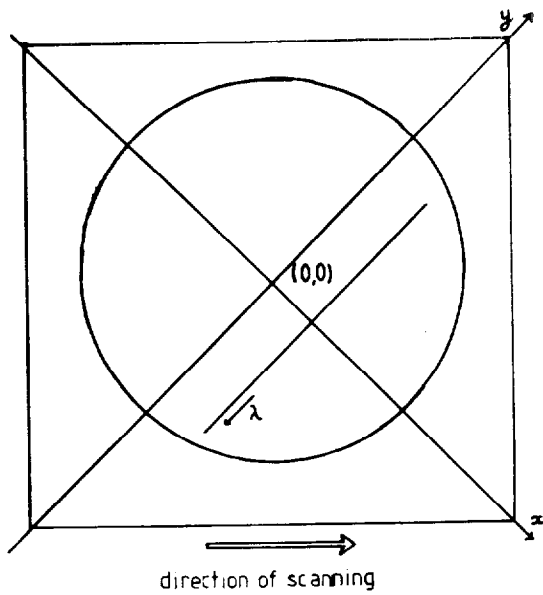


figure 2

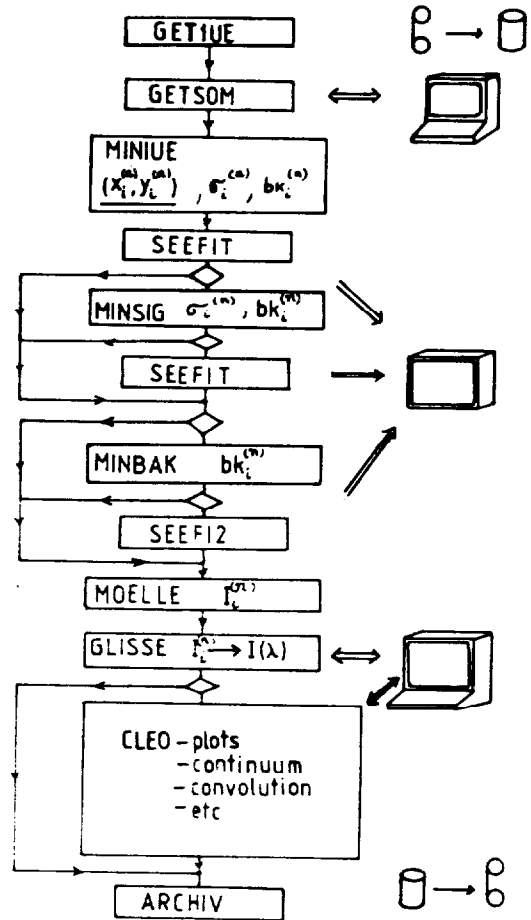


figure 3

Figure 4

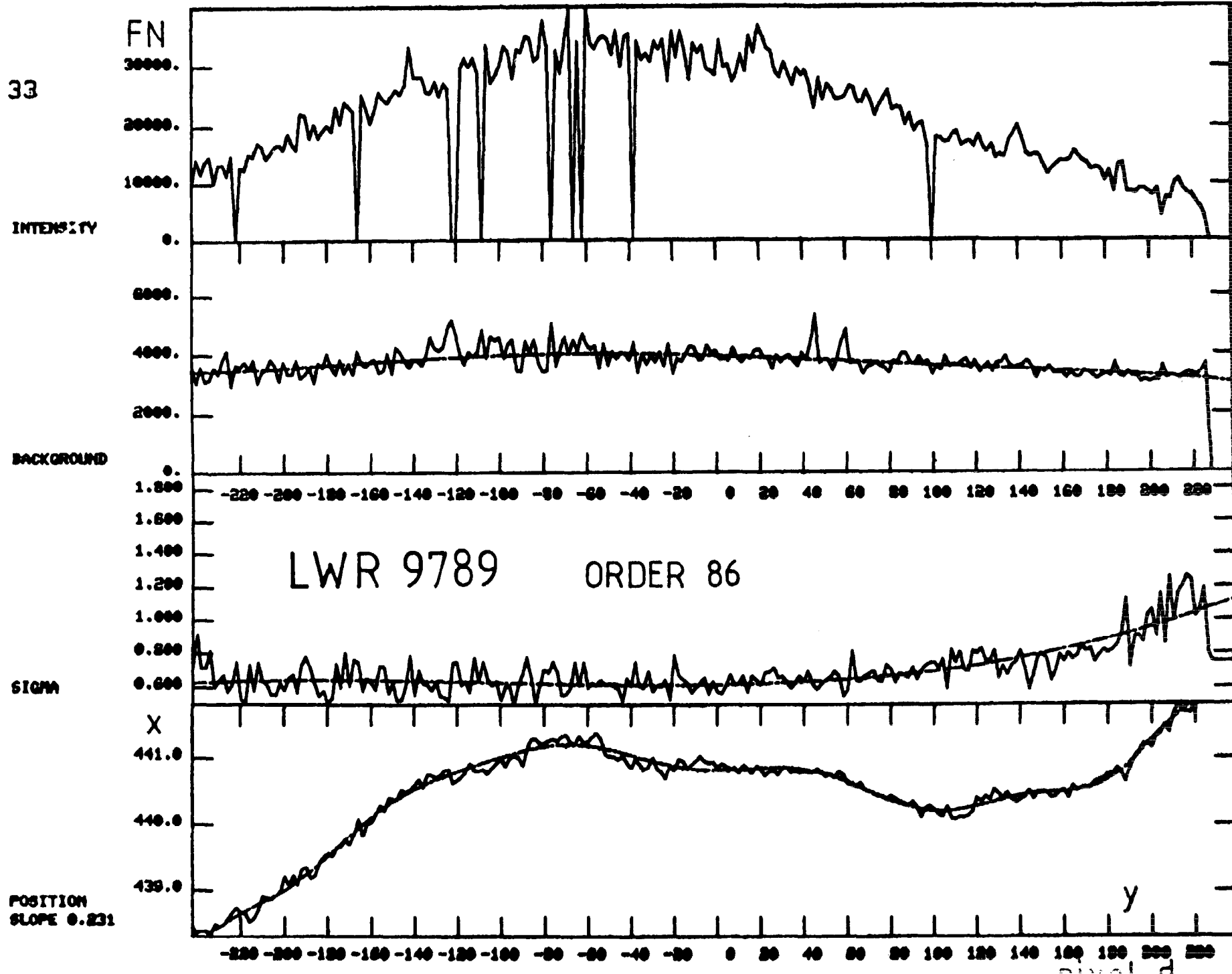
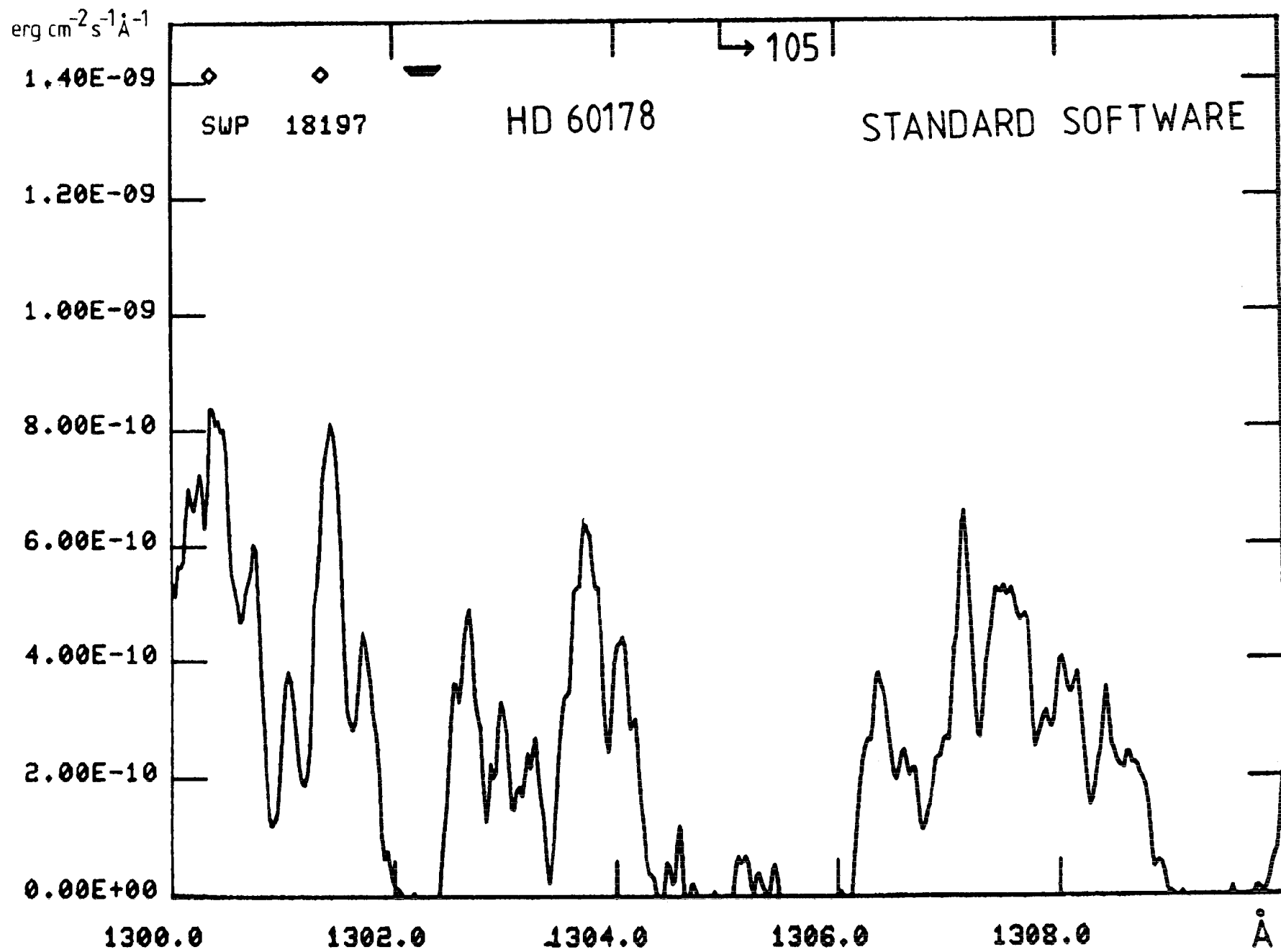


FIGURE 5



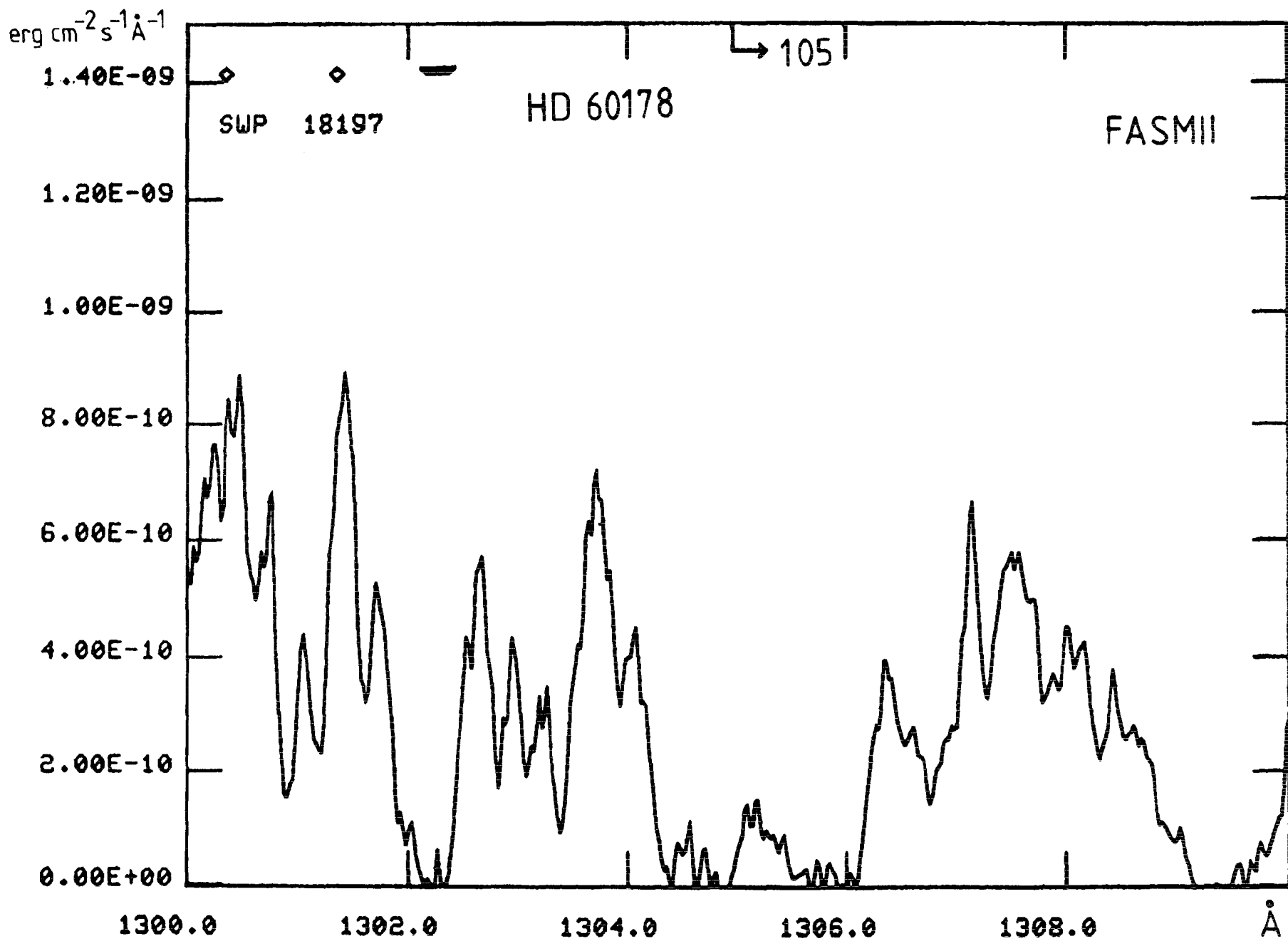


FIGURE 6

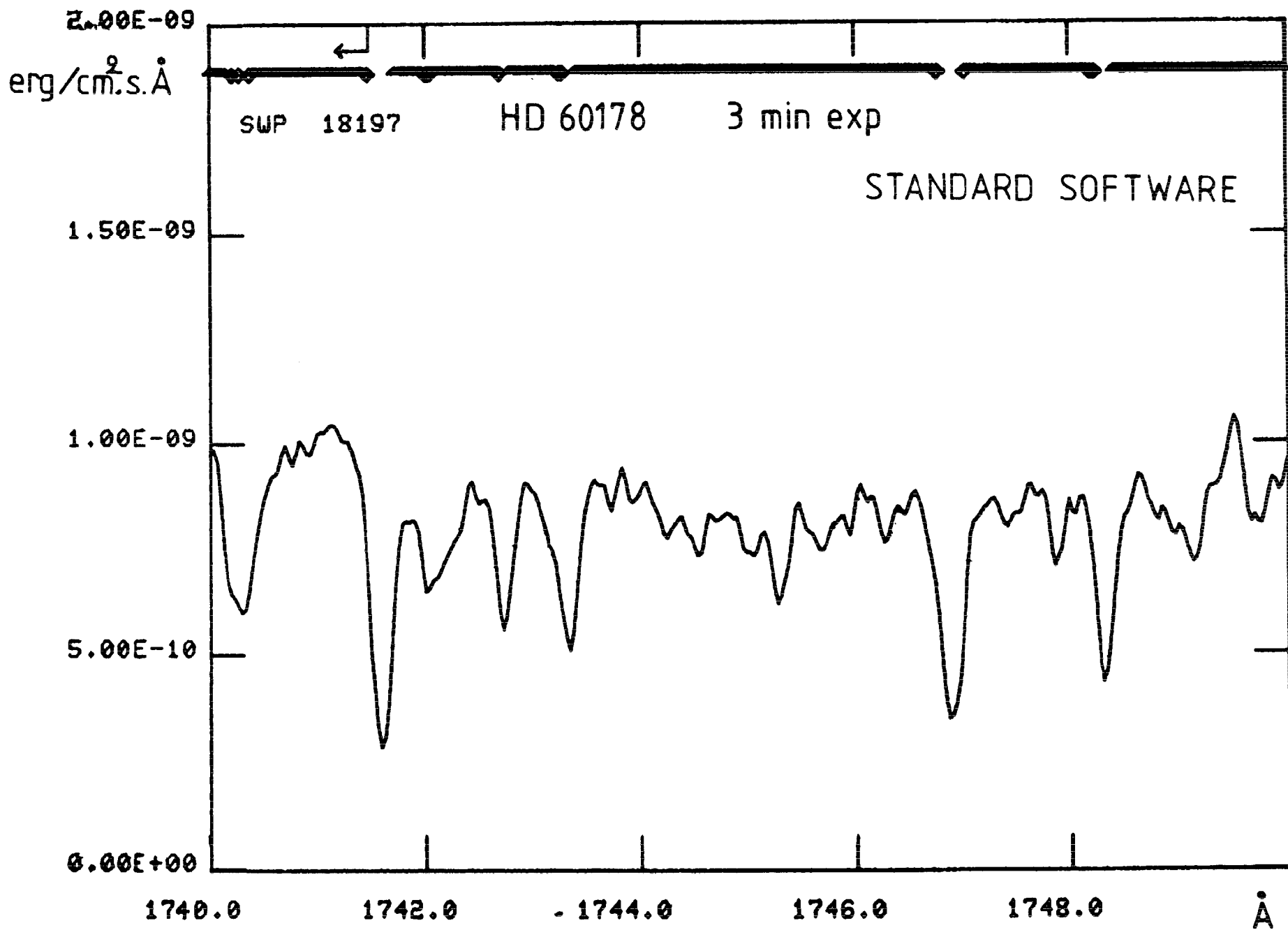


FIGURE 7

FIGURE 8

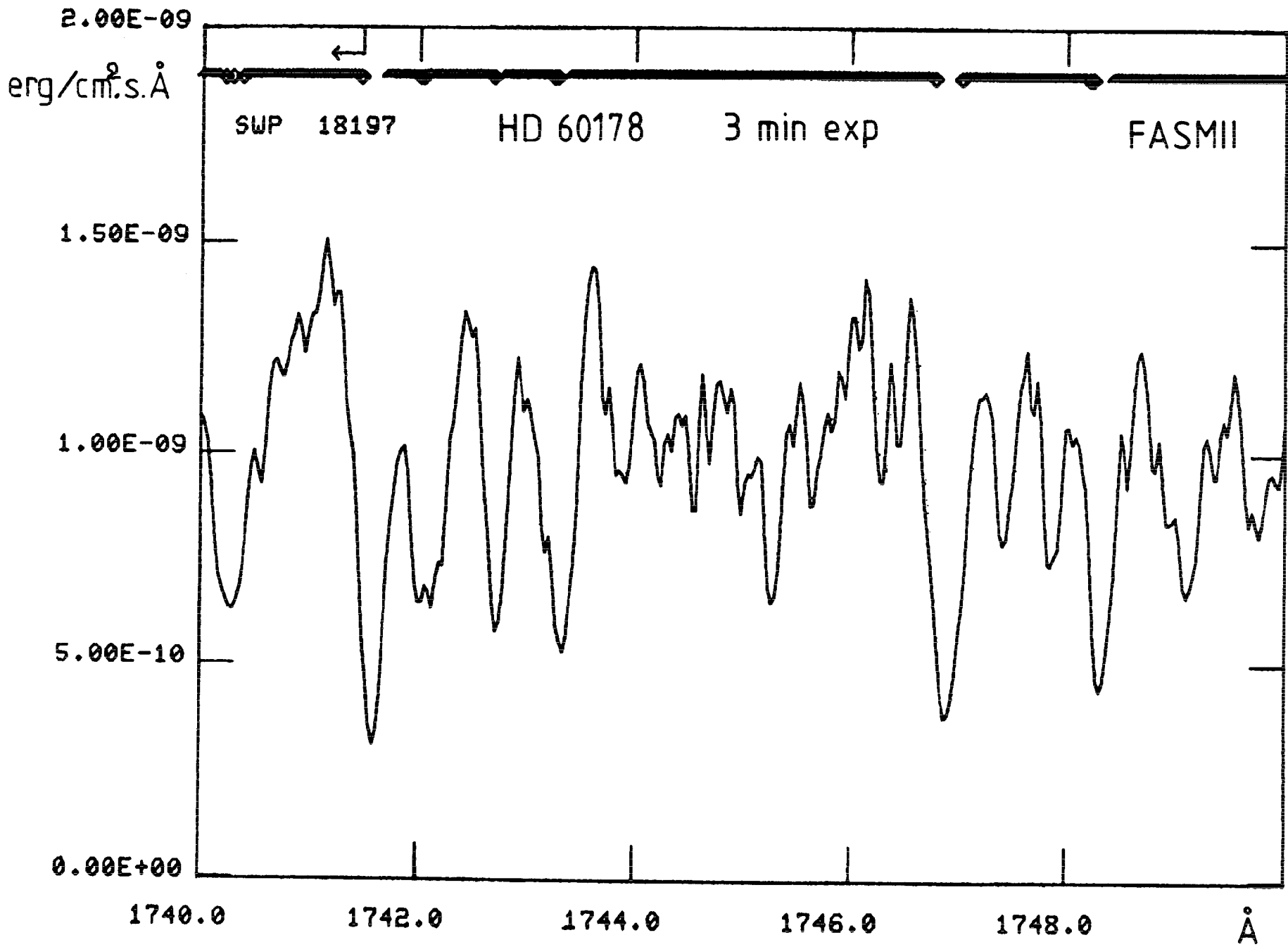


FIGURE 9

