```
.pl 60
.mb 3
.po 0
.foGaussian extraction procedure
Page #
```

GEX

An IHAP implementation of the Gaussian Extraction for IUE

Software and documentation by
L.Chiappetti - IFCTR

April 1989

An implementation of the Gaussian Extraction procedure (Urry & Reichert, 1988, enclosed in Appendix) for IUE spectra is now available in the form of an IHAP batch file on the IFCTR HP-1000. The procedure operates starting from a "fourth file" (line-by-line spectrum) and produces a net, one-dimensional spectrum calibrated in flux.

The procedure is alternative to the standard family of extraction batches previously available in IHAP (i.e. IUELOO, IUEZZZ, IUE4TH and DISK4 as described in Belloni & Chiappetti, 1987, "Sistema di analisi IUE. Manuale per l'uso", 3.5.1).

The procedure is similar to the batch DISK4 (Belloni & Chiappetti, 1987, 3.5.3.4), as it always starts from a fourth file, which should have been read onto disk in IHAP from the tape produced by the program IUE (see 3.1 in Belloni & Chiappetti, 1987) using command MRES.

The procedure makes use of two external Fortran programs to perform the bulk of the first and second pass of GEX as described by Urry & Reichert. A number of intermediate files are produced both within IHAP and outside it on disk cartridge X2.

As in the original Urry & Reichert procedure, the only parameters to be chosen are the rebinning factor for the first and second pass. The values recommended there are a rebinning of 10 for the coarse (first) pass, and of 1 (i.e. use the original data) for the fine (second) pass.

The complete batch is called as follows:

BATC,GEX,,file,factor1,factor2

where file is the IHAP file number of the fourth file to be processed, and factor1 and factor2 are the binning factors for the first and second pass (of course factor1 should be more coarse than factor2).

The input file file shall have an identifier of the form **4** CCCnnnnn (where CCC is the camera identifier and nnnnn the image number), as normally generated by the program IUE. It shall also contain the correct exposure time in the user float G37. As in the standard extraction batches, if the exposure info is missing, a valida exposure time in sec is asked to the user.

.he$PAGIFCTR IUE reduction software note
April 1989
.cp4

As a preparation for the first pass, the following actions are performed within IHAP :

.cp5

compute the minimum and maximum of file

generate a new file of the same size, but scaled to 0-1, using the above minimum and maximum. This file has the identifier copied from file but with a "0-1" indication added **4** CCCnnnnn **0-1**.

rebin the scaled file with the requested binning factor factor1. This file has the identifier **COARSE** factor1

transfer the scaled and binned file outside IHAP into work file **PASS1::X2**

The program GEX1 is then scheduled. The only argument passed to this program (which may also be used independently outside IHAP) is the full namr of the input file (here **PASS1::X2**). The program works as follows :

sets the limits for the background and signal regions according to Urry and Reichert (in dependence whether the file is 55- or 110-lines)

for each wavelength chunk, extracts the background data, and fits the data vs scan line number as a straight line, using Bevington's LINFIT routine. Points whose residuals exceed 2 RMS are then removed and the fit repeated according to the prescription of Urry and Reichert.

for each chunk, the best fit background is subtracted from the data in the signal region, and it is verified whether there is enough signal.

For each chunk, and if there is enough signal, a Gaussian is fitted to the net data, using Bevington's CURFIT with no weights. Points whose residuals exceed n RMS (with n adjusted according to the prescription of Urry and Reichert to keep at least half of the points) are removed and the fit eventually repeated.

The values of slope, intercept, Gaussian peak (constrained to the range recommended by Urry and Reichert) and Gaussian width, i.e. sigma, (constrained to 0-3 or 0-6 as in Urry and Reichert) are saved. They are written to file, except for the width.

The width as a function of wavelength is fitted to a parabola, using Bevington's POLFIT. Only points with non-zero width are used (i.e. the points where there was no signal are not used, as the Gaussian parameters were forced to zero).

.cp4

and creates an intermediate file **GEXINT::X2**. This file is overwritten all the times. It is a binary, direct access file, where each record contains four REAL*4 values. The first record contains the number of data record n, and the three parameters of the parabolic fit, the remaining n records contain respectively the wavelength, slope, intercept and peak position.

.cp3

As a preparation for the second pass the following actions are performed within IHAP :

rebin the scaled file with the requested binning factor factor2.
This file has the identifier **FINE** factor2

transfer the scaled and binned file outside IHAP into work file **PASS2::X2**

However in the usual case that no binning is requested for the second pass (i.e. factor2 = 1), the original scaled file is transferred directly into **PASS2::X2**

The program GEX2 is then scheduled. The only argument passed to this program (which may also be used independently outside IHAP) is the full namr of the input file (here **PASS2::X2**). The program works as follows :

for each "fine" wavelength, it finds the applicable wavelength chunk in the intermediate file GEXINT. If the chunk was one with no signal, the Gaussian height is set to zero.

Otherwise the value of slope, intercept, Gaussian peak and width are computed for the "fine" wavelength, either interpolating from values in the intermediate file, or (for the width) using the parabolic fit.

The data at each fine wavelength are fitted versus scan line number over the entire range (55 or 110). The fit used to derive the Gaussian height is a CURFIT to a Gaussian plus linear background, where the height is the only free parameter. The fit is eventually repeated after conditional removal of bad points (same as in the first pass).

The value calculated for each scan line (and put back into a file at the current wavelength) is the best fit of the Gaussian <u>with no</u> <u>background</u>, i.e. the net signal.

and creates an intermediate file **GEXFIN::X2**. This file is overwritten all the times. It is a binary file in IHAP format, which contains a foruth-file-like spectrum, with zero background and signal given by the gaussian fit.

.cp4

The batch is then resumed and performs the following actions within IHAP (in a way similar to the standard extraction):

.cp4

retrieve file **GEXFIN::X2** into a file using the following identifier **4** CCCnnnnn **GEX.**

.cp4

create a new (temporary) file, scaling back from 0-1 to the original uncalibrated fluxes.

.cp4

Determine the camera identifier, read in the appropriate calibration file, do a spline and generate an IHAP 1-d file containing the calibration curve (as in the standard batches).

Add all scan lines in the temporary file in uncalibrated fluxes, creating another temporary (1-d) file.

Multiply the latter by the calibration curve, and divide it by the exposure time. The result is the final extracted spectrum, an 1-d file with identifier **4** CCCnnnnn **GAUSS**

Finally the batch should clear all intermediate files. In the present version the intermediate IHAP files are left on disk (for eventual checks), while the **PASS1** and **PASS2** files are purged.

The intermediate files appear in a DLIS,ALL as follows (note that the **FINE** file (#m = #n+2, k=1) appears only if factor2 is different from unity, ie.e if any rebinning has been requested for the second pass (otherwise k=0; in all cases #p = #m+k+1).

#n	**4** CCCnnnnn	DRES,...	original 4th file
#n+1	**4** CCCnnnnn **0-1**	PFUN,...	4th file scaled to 0-1
#m	**COARSE** factor1	REBIN,...	coarse binned file
#m+k	**FINE** factor2	REBIN,...	fine binned file (if any)
#p	**4** CCCnnnnn **GEX**	DRES,GEXFIN ..	the result of GEX2
#p+1	blank	PFUN,G801,...	rescaled back to "fluxes"
#p+2	blank	TRES,....	the calibration curve
#p+3	blank	SADD,...	1-d uncalibrated spectrum
#p+4	**4** CCCnnnnn **GAUSS**	PFUN,G803,...	final extracted spectrum

.cp7

Possible problems

A number of problems are known from the original paper of Urry and Reichert, and are confirmed.

One concerns the edge of the SW camera, perturbing the background at long-wavelength end. The problem should have been cured, inserting a change in the code of GEX1 as suggested by Urry and Reichert.

Another one concerns saturated lines, namely the geocoronal Lyman a, which has a broad profile and a flat top (because of saturation, actually not even flat, but irregular). In this case the gaussian fit fixes the peak on the top point of the irregular part, and discards the higher parts of the wings. The result is a misplaced centre, and a larger width, which may disturb the parabolic fit. The suggestion is to truncate the input spectrum just longwards of the Lyman a, or blank it out (if you do so remember to copy a suitable IHAP identifier !!).

.cp4

A further known problem is that the gaussian fit does not use any "error" weight on the points (the peak and the wings are weighted equally). However the GEX1 program is consistent with the simplest approach in Urry and Reichert.

The latter point means that the gain using GEX with respect to other extraction methods is likely not to be so large, in terms of better S/N ratio for generic spectra. The improvement for low quality (bad S/N spectra) is actually only due to the fact that a binned spectrum is produced.

An additional minor problem is due to IHAP scaling handling during "transverse" plots. Namely, if one has files with different x-steps (as it is the case for a file extracted via GEX if a binning factor different from one is used in the second pass) they cannot be correctly overtraced. The content of the file is all right, a single file can be plotted with the TRAN command without problems (possibly after issuing a **SCALE,#**n,**X,**factor2 command) : the problem is just with the plotting of more files with the OVER command, and is a known common problem in IHAP (it is a limitation of the OVER command mentioned in the IHAP manual).

A few hints for an esthetic solution, if you want to compare extracted spectra with different x-steps. Consider as example an "original" file #1 with steps of 1.5 A from 1000 to 2000 A, which, when plotted, occupies the entire 2648 screen. Assume also you have two files #2 (binned by a factor 3) and #3 (binned by a factor 7). If you do a default transverse plot, they will occupy about 1/3 and 1/7 of the screen. To use the entire screen you do a SCALE,#2,X,3 and SCALE,#3,X,7 respectively. This applies to the TRAN command, but not to the OVER command.

If you want to compare #2 and #3 on the same plot, you have to do the following:

```
     DLIS,#1             to get the x-step of the original file
     in G33
     REBIN,#2,,G33,,HO   creates #4 with a step 3 times the one
     of #2
     REBIN,#3,,G33,,HO   creates #5 with a step 7 times the one
     of #3
```

Now #4 and #5 will nominally have both a step of 1.5 A, i.e. the one of the original file, however this is fictitious. In the case of file #4 the value of the bin content changes every three bins, in the case of #5 changes every seven bins. When plotted on a transverse they will appear exactly as #2 and #3 respectively. But, bow and behold, when plotted on the same frame with a TRAN followed by an OVER command, they can be compared on the same scale ! Of course the price is that they occupy (actually waste) 3 or 7 times more space than necessary.

.cp55

Notes for programmers and advanced users

The source of the batch GEX is clearly commented. For details concerning the parts in common with the standard batches (e.g. handling camera identifiers etc.), refer to comments in the latter batches.

Also the sources of GEX1 and GEX2 (which are programs **&GEX1::LC** and **&GEX2::LC**) are clearly commented. It has to be noted that they make reference to an include file **&GEXCO::LC** for the definition of the EMA common blocks.

The programs shall be loaded using **TR,*CLK,&GEX**n (n=1,2), using therefore the loader files **#GEX1::LC** and **#GEX2::LC**. Note that the programs are VMA (Virtual Memory) programs. The size of a 110-line fourth file is bigger than any normal EMA area, therefore VMA has been used, with an EMA working set of 100 pages. Therefore the program can run only in the biggest mother partition (if any other program is running in some of its subpartitions, delays are possible).

If the maximum size allowed for IHAP images (which is kept in PARAMETER statements in **&GEXCO**) is altered, it is possible that the dimension of the VMA in the loader files has to be increased accordingly.

The program in the normal version is as quick as possible. To obtain a reasonable speed, IHAP files are transferred from/to disk using direct VMA transfer (this way an entire image is transferred in less than 1 sec, compared with 10-30 sec using either FMP or Fortran calls). The price to pay is that data shall be handled within the program as an 1-d array, instead of a 2-d array (i.e. addressing is explicitly resolved).

.cp4

Furthermore, writing messages to the terminal is kept to a minimum. The time is therefore sheer CPU time, or time spent for paging the VMA in and out of the working set. The first pass is quite quick (13 sec for a 55-line file, with binning factor 10), while the second pass can be up to about 200 sec for an unbinned 55-line file. The slowness of the second pass (contrary to Urry and Reichert) is possibly due to paging (memory arrangement not optimized, forced by IHAP disk arrangement).

.cp2

In any case the time of the complete batch is comparable with the one of the standard extraction batches.

.cp4

A debug version of the programs is available, recompiling them with the D option (**TR,*CLK,&GEX**n,**D**). Of course this way the programs are much slower, because of the numerous messages written to the terminal with intermediate results, and specially because of the plotting.

.cp3

Note the plotting is not optimized at all, it does quite rough things calling the **PABLO** program (schedule with wait). Plotting is arranged as follows (run on a graphic terminal, preferably a colour one):

.cp4

For the first pass GEX1 :

For each chunk, for the background calculation, a frame is plotted to the lower part of the screen, containing the background data (points) and the best linear fit (solid line). If some points are excluded, the points left are redrawn in a different colour, and the new fit also. Finally the signal+background data are plotted for reference (crosses).

For each chunk with sufficient signal to perform the gaussian fit, a similar plot of the background subtracted signal data is made in the upper part of the screen.

Plots like those are done for each chunk. The screen is cleared before each plot. To start plot no action is required, there should be ample wait time to look at one before the next one overwrites it.

At the very end a plot of four frames is made, plotting versus wavelength the following quantities (from bottom to top):

slope of the background
intercept of the background
peak position
gaussian width with parabolic fit

A few temporary files are created for the plots (refer to source listing). An useful one to be replotted in a more elegant way could be file **STUFF::X2** which contains wavelength, slope, intercept, peak and width.

.cp4
For the second pass GEX2:

To speed up things, the plot is generally not done (while the alphanumeric information is displayed to terminal for all chunks). To force a plot for one of the next chunks, press any key while the program is running, and issue a break (reply **BR** to the **S=COMMAND ?** prompt). In this case only the plot is scheduled.

The plot itself is similar to the ones described above, the data and the fit are plotted, then, if the fit is repeated, the points left and the new fit are plotted in a different colour.

.cp4
A few temporary files are created for the plots. Moreover a file GEXPAB::X2 is created in debug mode, which contains the wavelength, the integrated flux under the gaussian, and the gaussian height (all still referring to the 0-1

scaled file !!). This file could be plotted as wished.

.cp4

One note concerning intermediate files. All files created by GEX1 and GEX2 (whether they are the work files for the batch, GEXINT and GEXFIN, or temporary plot or command files in debug mode) are overwritten each time the programs are run. They are anyhow on disk X2, which is cleaned daily, and the user should not be concerned by purging them.

On the other hand the two disk files PASS1 and PASS2 created by IHAP need also to be overwritten each time the batch GEX is called. However they are created by IHAP with a security code (as typed by the user when entering IHAP), which is arbitrary and unknown to the batch. The only way to purge them would be to use a privileged command (implying knowledge of the master security code) to get the file security code, or to access a privileged section of the directory information. To prevent security problems, this is handled by a third program GEX3 (which is run by batch GEX before termination).

This program (source in **&GEX3::LC**, loader file in **#GEX3::LC**, to be loaded with *CLK as it makes reference to unusual libraries) purges only **PASS1::X2** and **PASS2::X2**, using an "unconditional-purge" routine, in a way totally transparent to the user. Advanced users needing a similar functionality for their own batch files, may obtain the code of such routine from me.

.he$PAGIFCTR IUE reduction software note
Appendix A
.pa
.pn 1
.foGaussian extraction procedure (appendix)
Page A-#
.pa
.pa
.pa
.pa
.pa
.pa