Influence of Noice on Fitted Position and Intensity of Spectral Line-Shapes

Blom, Anders

2000

Influence of noise on fitted position and intensity of spectral line-shapes

Anders Blom

Lund Reports on Atomic Physics

LRAP–262
(2000)

Influence of noise on fitted position and intensity of spectral line-shapes

Anders Blom

Department of Physics, Lund University, S–223 62 Lund, Sweden

Abstract

The expected statistical error (standard deviation) of the area and position of a spectral line containing noise, obtained from fitting a Gaussian or a Lorentzian to the line shape, is modelled as a function of the signal-to-noise ratio, the width of the line and the sampling distance. A simple analytical expression – of the same form for both Gaussian and Lorentizan shapes, and with a common numerical coefficient for both area and position – is found to describe the errors accurately for a wide range of parameters.

1 Introduction

When determining the position and intensity of a spectral line, the standard method is to fit a line profile to the experimental data. The algorithm used for the fitting usually supplies some estimate of the error in the fitted parameters. In contrast, this report will provide means for *predicting* the statistical error in the same kind of fit, when three basic parameters are known: the signal-to-noise ratio (SNR), the width (FWHM) of the line, and the sampling density.

The difference between these two ways of estimating the error in the fitted parameters is fundamental, but perhaps not obvious. The error estimate supplied by the fitting routine is only based on one single observation of the line shape, and deals only with the statistical properties of that particular measurement. This will take into account statistical fluctuations of the samples (noise), the fact that our model perhaps does not agree perfectly with the observation (the line shape may not be perfectly Gaussian), and also errors caused by the fitting algorithm itself. However it does not provide any means to distinguish between these different sources of errors, and can in some cases be misleading, either over- or under-estimating the errors, due to spurious data.

The method described in this report instead focuses entirely on the aspect of noise. To do this, a synthetic spectrum, consisting of a perfect line shape (Gaussian or Lorentzian) plus added random noise, was generated. To this was then fitted a line profile of the same shape. By comparing the difference between the results obtained from the fit and the original parameters used for generating the line, and repeating this for the same parameter combination but with different noise, the desired error may be estimated as the standard deviation of the individual differences.

By performing this for a large number of different combinations of parameters, it was finally possible to parametrize the error as an analytical function of the signal-to-noise ratio, the width of the line, and the sampling density.

Thus there is an important difference between the two methods which can be put to good use. While the error estimate supplied by the fit routine is only known *after* the fit is performed, the expressions provided in this report allow the errors to be *predicted* beforehand. This means that given say the sampling distance and the line width (parameters usually known at least approximately before a measurement is carried out), it is possible to determine the required signal-to-noise ratio (which in practice controls the time required for the data acquisition) such that the errors will be acceptably low.

In this report is first presented the model used, then the analysis and the thereby obtained results, and finally as an Appendix a User's Guide to the computer program used for the analysis.

2 Definitions and method

- *Definitions*

  Noise is *white (random) noise*, normally distributed with zero mean and standard deviation 0.5. This makes approximately 95% of the noise fall between ±1.

  The spectral line is assumed to be either *Gaussian*

  $$y(\sigma) = A\sqrt{\frac{4\ln 2}{\pi}}\,\exp\left(-4\ln 2\left(\frac{\sigma-\sigma_0}{w}\right)^2\right) \tag{1}$$

  or *Lorentzian*

  $$y(\sigma) = \frac{A}{2\pi}\frac{w}{(\sigma-\sigma_0)^2+(w/2)^2}, \tag{2}$$

  where in both cases σ_0 is the center wavenumber and w the FWHM (both in cm^{-1}). For the Lorentzian the area under the curve is A; for the Gaussian it is Aw.

  Signal-to-noise ratio (SNR) is defined as the ratio of the maximum value of the spectral feature to the rms (root-mean-square) value of the noise. Given the above stated definition of the noise amplitude the rms value is 0.5, so for a peak amplitude of 10, the SNR is 20.

  Resolution (dx) is taken as the distance between two data sampling points, in units of cm^{-1}.

- *Method*

  1. Generate a spectral line with a Gaussian or Lorentzian shape. Its center position is chosen to lie *between* two data sampling points[1].
  2. Generate noise distributed as stated, and add noise and spectral line to make a synthetic spectrum.
  3. Fit a Gaussian/Lorentzian to the synthetic spectrum. Initial fit parameter values are found from a very rough analysis (simply locate the sample of highest amplitude), which is adequate since the spectrum has such a simple appearance and only three parameters are fitted. The fit tolerance can be set to provide very high accuracy.
  4. Calculate the intensity (area) of the original line and the fitted one and compare. Do the same for the original and fitted center position.
  5. Do this many times, for the same synthetic line but with different noise, and make statistics.

[1] This is introduced just in case the fit algorithm is influenced by whether the peak value is sampled or not. Most likely this is not important.

3 Analysis

Data was generated using the program described in Appendix A, covering an extensive parameter range. The total calculation time was around 50 hours on a moderately fast PC. For each parameter combination, a large number of iterations ($N = 1000$) were made, each with regenerated noise, for both Gaussian and Lorentzian profiles. For each iteration the fitted position σ_i and area A_i of the spectral line was compared to the true, known, values σ and A. Estimates of the standard deviations[2]

$$(\Delta\sigma)^* = \sqrt{\frac{\sum(\sigma_i - \sigma^*)^2}{N-1}}, \qquad (\Delta A)^* = \sqrt{\frac{\sum(A_i - A^*)^2}{N-1}}, \tag{3}$$

were calculated and recorded. These are the quantities of interest, as they give an estimate of the error in the position and area determination.

To avoid repeating the same information, only one of the two cases (Gaussian and Lorentzian) will be shown in this report for each type of plot, but the same dependencies apply to both, as was verified explicitly during the analysis.

An important feature of the results is that the fitted positions and areas within the same parameter combinations are close to normally distributed, as shown in Fig. 1, which justifies the use of the normal approximation for the estimation of the standard deviation. Possibly the statistical arguments for using the normal approximation are valid even for much fewer iterations, but by using such a large number of iterations, the errors in the fitted parameters could be reduced substantially, compared to a similar analysis made for $N = 200$.

To decompose the problem, the dependence on the signal-to-noise ratio was considered first. When plotting the inverse of the standard deviation[3] against the SNR (denoted by S in the formulas) for a fixed width and dx, a linear trend is obvious; see Fig. 2. Therefore direct proportionalities

$$\frac{A}{\Delta A} = c_A S, \qquad \frac{1}{\Delta\sigma} = c_\sigma S, \tag{4}$$

were assumed. Note that for the area we need to normalize by studying $A/\Delta A$, since the absolute error of the fit increases as the area itself increases, whereas for the position the accuracy is independent of the actual wavenumber, as it should be[4]. The exclusion of a constant term is reasonable; when the SNR is zero, there is no line and therefore the error is infinite.

This fit is performed for many different values of the width for a fixed sampling distance ($dx = 0.01$ cm^{-1}), and the results (the different c's) are recorded. The errors of the fits are very small, indicating good agreement with the assumed relations (4).

To study the dependence on the width we plot the error against w (not shown in this report). From this plot one finds it reasonable to assume a power-law relation $a\,w^b$, and when this is fitted to the data, the exponents come out as $b_A = 0.515$ and $b_\sigma = -0.472$. These both values are close to ±0.5, so it seems reasonable to use simple square-root dependencies:

$$\Delta\sigma \propto \sqrt{w}, \qquad \Delta A \propto \frac{1}{\sqrt{w}}. \tag{5}$$

These relations are further motivated[5] by plotting the c's (from Eq. (4)) obtained from the SNR fit mentioned above against w; fitting a power law to these curves gives exponents 0.498 and -0.501 (these are example numbers only, but they are representative).

[2] Since the symbol σ is used for the wavenumber, we use Δ for the standard deviation.

[3] In the following we drop the stars indicating estimates to simplify the notation.

[4] There are however other factors in spectroscopy which *are* wavenumber dependent, such as the sampling frequency, and so may enter implicitly. The independence of the actual wavenumber was also verified within the framework of this analysis.

Finally we turn to the sampling distance dx, defined as the distance in wavenumbers between the individual samples. For obvious reasons this parameter must be constrained to satisfy

$$dx \lesssim w \tag{6}$$

if the analysis is to be physically meaningful. It is however not clear from the outset whether the dependence on dx is as simple as $\Delta\sigma = f(dx)$, or of the form $\Delta\sigma = f(w - dx)$, which could seem reasonable because of the constraint (6). This will have to be answered within the analysis.

To study the effect of dx, the errors were calculated for the same number of iterations ($N = 1000$) for a fixed width and a fixed SNR. However, to be able to verify that the expressions (4) and (5) are still valid, this was done for a couple of different combinations of SNR and w.

When plotting the errors $\Delta\sigma$ and $\Delta A/A$ against dx (keeping $dx < w$ in mind), the result again resembles a power law, and when fitting, the exponents again come out very close to 0.5; c.f. Fig. 3, where the two lines are fitted square-root curves.

To resolve the issue whether there is any dependence in the error on $|dx - w|$, the two curves in Fig. 3 were re-scaled to remove the dependence of w. Since $w = 0.05$ and 0.10 for the two curves, from (5) the scaling factor becomes $\sqrt{2}$. The result is shown in Fig. 4, where the error is plotted against $\sqrt{dx}$. From this we conclude that the dependence on dx may be taken as the simple square-root proportionalities (5), as long as the constraint (6) is fulfilled. Similar plots for different SNR show that also (4) is valid independently of dx.

So, the final dependencies can be written

$$\frac{\Delta A}{A} = \alpha_A \frac{\sqrt{dx}}{S} \frac{1}{\sqrt{w}}; \qquad \Delta\sigma = \alpha_\sigma \frac{\sqrt{dx}}{S} \sqrt{w}. \tag{7}$$

The proportionality factors α may easily be found from the extensive amount of data already generated for the SNR and w analyses, as long as dx is eliminated. Using Eqs. (4) and (7) we immediately see that the proportionality factors are obtained by fitting a straight line (with intercept zero) from to following relations (Figs. 5 and 6):

$$c_A^2 \, dx = \frac{1}{\alpha_A^2} \, w, \qquad \frac{1}{c_\sigma^2 \, dx} = \alpha_\sigma^2 \, w. \tag{8}$$

The results from the fitting (traditional linear regression) are given below.

To provide an error estimate of the fitted parameters, one could use standard statistical methods for error propagation through the steps of the analysis. It is however judged that this will *underestimate* the errors, due to the following facts. The fitting of the final parameters has very small errors, as seen from Figs. 5 and 6; even when magnified by a factor 10, the residuals are still one order of magnitude smaller than the data itself, and the errors in the fitted parameters as estimated by the fit algorithm are in all cases smaller than 1%. However, the scatter in the plot against dx (Fig. 3) will not be properly accounted for, since we *assumed* the square-root dependencies (albeit after finding the fitted exponents to be very close to 0.5), so any errors associated with non-compliance to exact square-root dependence are swept under the rug. Also, it is difficult to estimate the statistical errors in the original results, i.e. the errors in the calculated standard deviations (3).

Therefore, to estimate the errors of the fitted numerical coefficients, each *individual* measurement was used. The corresponding coefficient was calculated from Eqs. (7) (since for each measurement the three

[5] To be even more statistically correct, the proper procedure should probably be to perform a hypothesis test to check the probability that the data is consistent with an exponent of 0.5 at a confidence level of say 95%. This has not been done yet. Still, the results themselves are not modified by this, only their justification, which anyway is reasonable from the arguments presented already.

parameters w, dx and SNR are known), and a traditional estimate of the standard deviation was taken as an estimate of the error in the fitted parameters.

The results were:

$$\begin{array}{rll} \text{Gaussian:} & \alpha_A = 1.41 \pm 0.04 & \alpha_\sigma = 0.69 \pm 0.02 \\ \text{Lorentzian:} & \alpha_A = 1.60 \pm 0.04 & \alpha_\sigma = 0.80 \pm 0.13 \end{array} \tag{9}$$

These numbers show an astonishing result: it seems likely to conclude that $\alpha_A = 2\alpha_\sigma$, valid for both Gaussian and Lorentzian shapes.

4 Summary and conclusions

We have found from a careful numerical analysis that the final expressions for the expected statistical error (standard deviation) of the fitted position and area can be written

$$\boxed{\frac{\Delta A}{A} = 2\,\alpha \frac{\sqrt{dx}}{S}\frac{1}{\sqrt{w}}, \qquad \Delta\sigma = \alpha \frac{\sqrt{dx}}{S}\sqrt{w}.} \tag{10}$$

The errors depend on a single numerical coefficient,

$$\boxed{\alpha = \begin{cases} 0.7 & \text{Gaussian,} \\ 0.8 & \text{Lorentzian.} \end{cases}} \tag{11}$$

The type of dependencies found are physically very reasonable: increasing the SNR naturally makes it easier to determine the position of the line. An increased width, however, will make the uncertainties of the position determination larger. It also becomes easier to determine the area with an increased SNR and with an increased width, even when the error is taken relative to the area, which of course itself increases with the width and the SNR. The fact that an increased sampling density (decreased dx) reduces the errors is also to be expected.

The most interesting point of the results is that all dependencies follow linear or square-root functions, and furthermore that the numerical coefficients for the two cases, area and position, are so simply related. None of these relations were initially assumed, but came out as results of the analysis.

These findings seem rather unlikely to occur by coincidence; however, to derive theoretically this relation would probably require a formidable effort, considering the large number of steps in the analysis - the addition of random noise, the linear fitting to the SNR, and then the subsequent fitting for w to the *results* of the SNR-fitting. However, for applications, whatever way the results were obtained is immaterial, as long as they provide a correct description.

As for extrapolating the results to even wider lines (or higher resolution) or stronger SNR, there are indications that the analytical expressions given above will fail to provide correct results for very large SNR, say over 500. But for these extreme cases, the errors described here are so small anyway that probably other errors involved in the measurement process will dominate.

An final interesting point to notice is that expressions like (10) have occurred earlier in the literature [3], but there is nowhere given any indication of how they were obtained. They also include a "constant of order unity". In this report we have given a clear justification of these expressions, and also determined the constant.

References

[1] J.J. More, "The Levenberg-Marquardt algorithm: Implementation and theory", in *Lecture Notes in Mathematics, 630, Proceedings, Biennial Conference, Dundee 1977* (A. Dold, B. Eckmann, and G.A. Watson, eds.) pp. 105–116, Springer-Verlag, 1978

[2] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C.* Cambridge: Cambridge University Press, 1992 (section 15.5)

[3] J.W. Brault, *Mikrochimica Acta*, 1987, III, 215–227

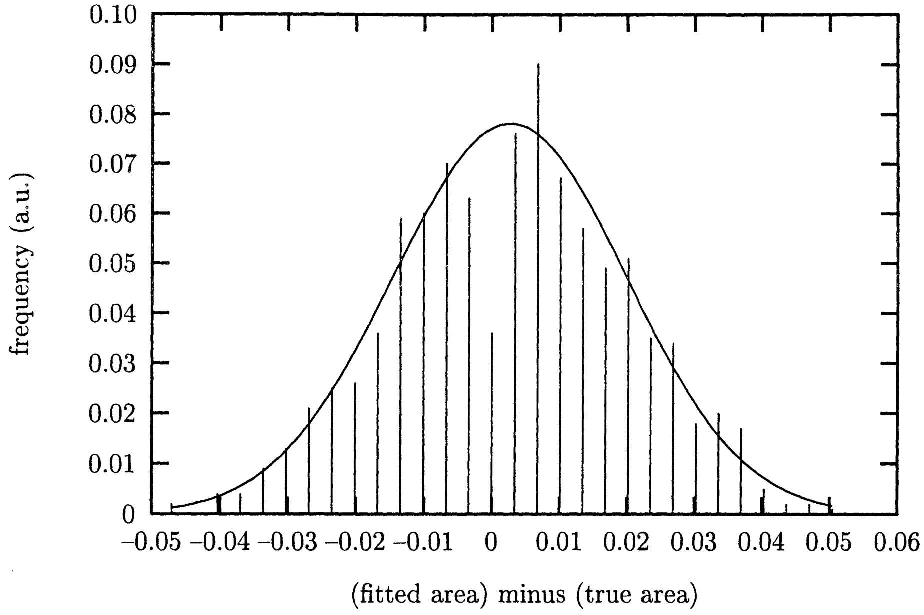

Figure 1: A histogram of the difference between the fitted and the true area for a Gaussian with SNR=20, w = 0.05 cm^{-1} and dx = 0.01 cm^{-1}, based on 1000 iterations. The figure indicates that the iterations are normally distributed around the true area, which is to be expected for a large number of iterations from the central limit theorem. The solid curve is a Gaussian fitted to the distribution.

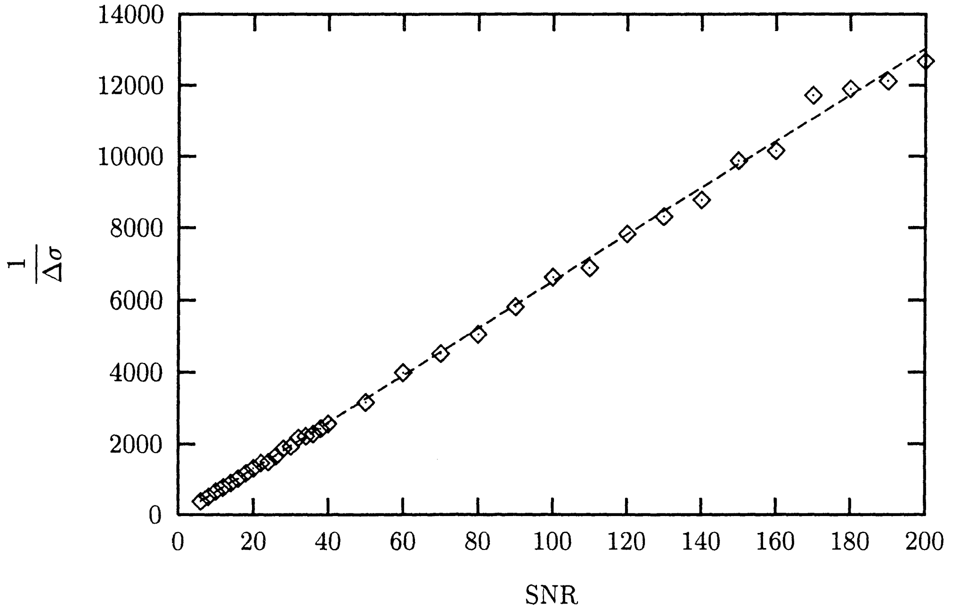

Figure 2: The inverse of the error $\Delta\sigma$ of the position is directly proportional to the SNR; c.f. Eq. (4). The plot shows the Gaussian case for w = 0.05 cm^{-1} and dx = 0.01 cm^{-1}. For large SNR the errors become very small, and so any small fluctuation in $\Delta\sigma$ will give a large change in $1/\Delta\sigma$, hence the increased scatter in the right hand part of the plot.

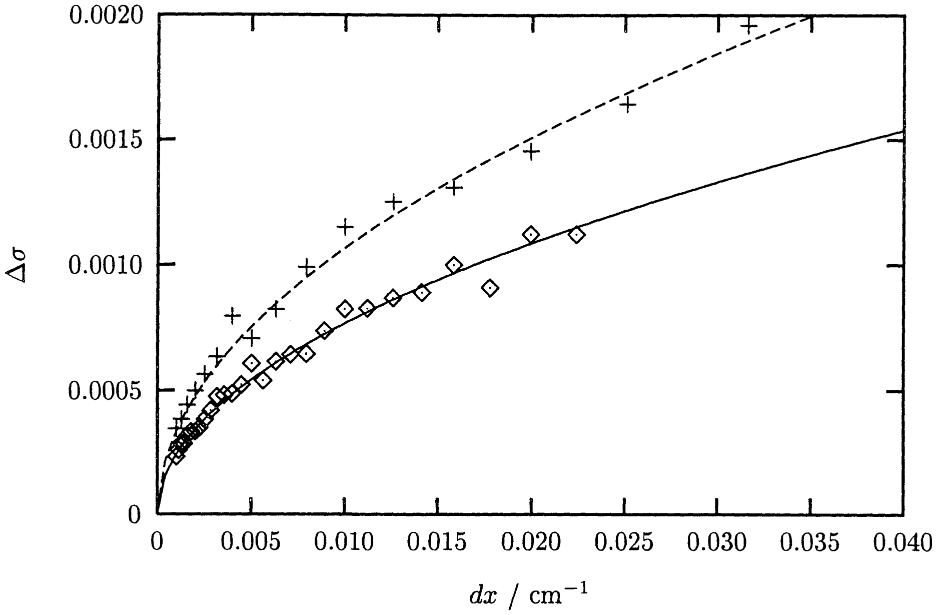

Figure 3: The error plotted against sampling distance dx for the Gaussian case of SNR=20 and $w = 0.05$ and $w = 0.10$ cm^{-1}, respectively. For both widths a fitted square-root curve interpolates the data well. Also see Fig. 4.

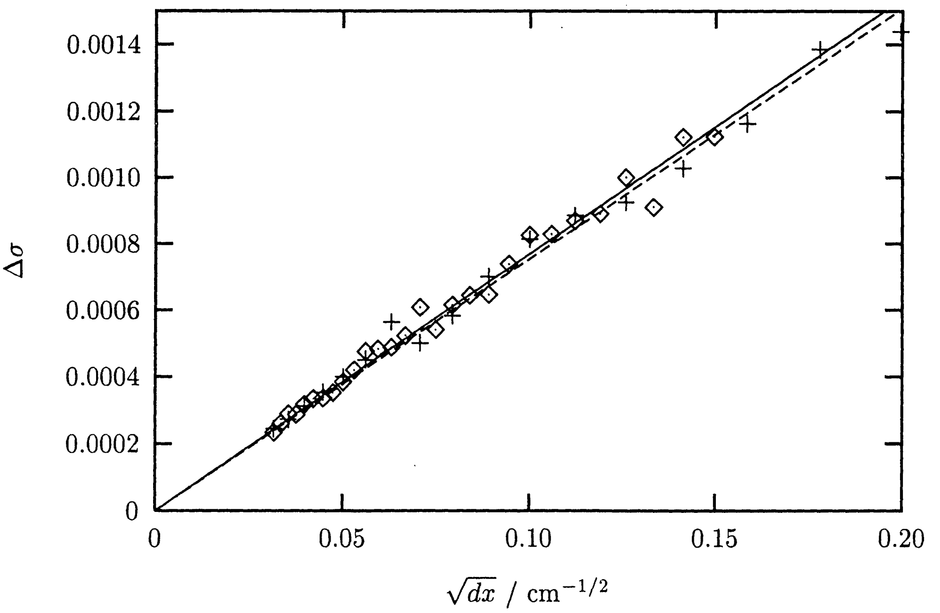

Figure 4: To verify that there is no dependence on $|dx - w|$, the errors for $w = 0.10$ were re-scaled by a factor $\sqrt{2}$, and compared to $w = 0.05$. As seen from this figure, the two cases overlap almost exactly. The straight lines correspond to the same fitted square-root curves as in Fig. 3, again with the case $w = 0.10$ re-scaled by $\sqrt{2}$. The fit to the data is very good indeed, and the slopes of the lines, after the re-scaling, coincide to within 1.5%.

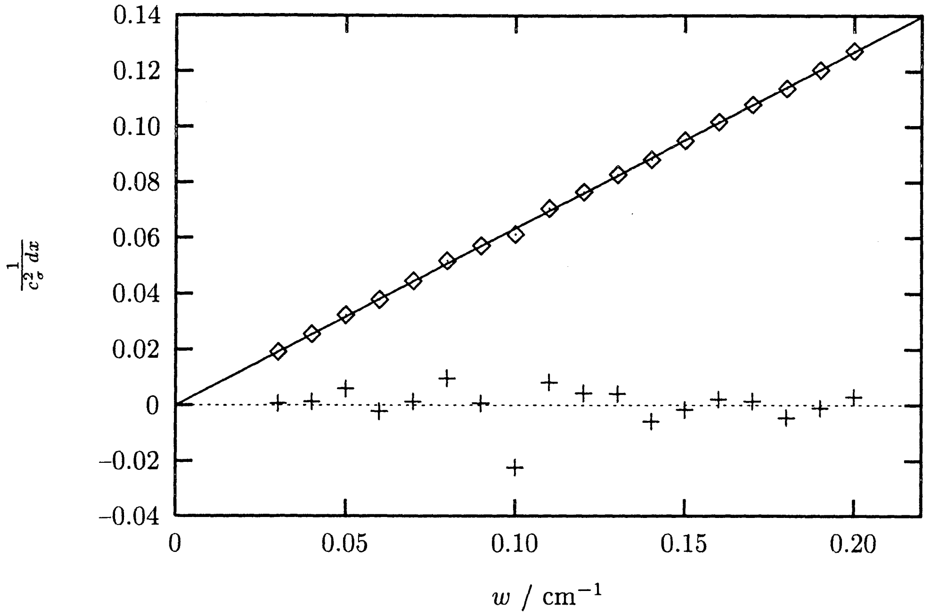

Figure 5: When plotting according to Eq. (8), the points fall very closely on a straight line, as predicted. This shows the Lorentzian case for the error in the position, and as before the data is based on 1000 iterations. At the bottom are shown the residuals, magnified by a factor 10.

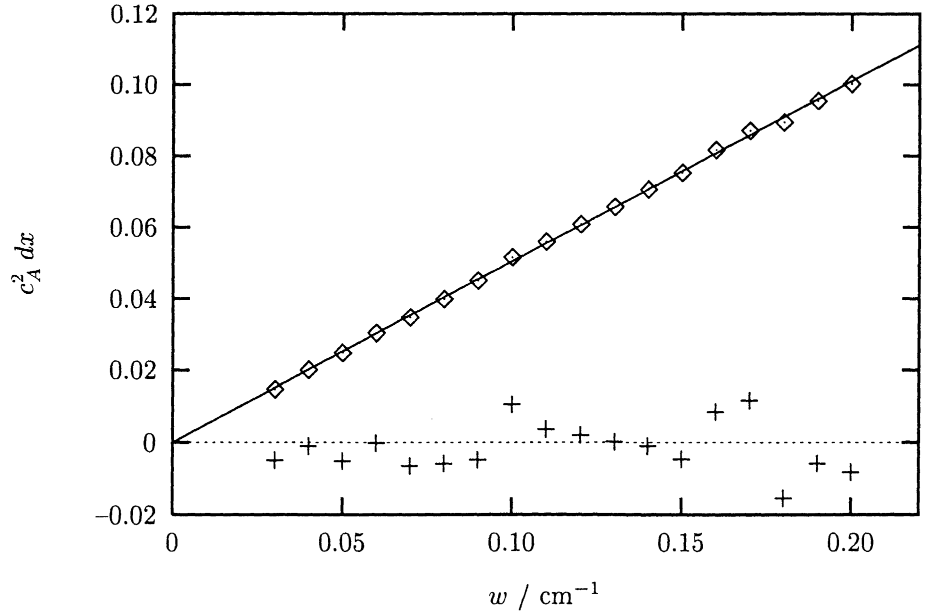

Figure 6: Again according to Eq. (8), the error in the area for the Gaussian case shown here, the points fall on a straight line. The residuals are also here magnified by a factor 10.

Appendix A: User's guide to the program

To run the program, simply launch the SNERR.EXE executable file. When running under Windows NT or 2000 the file RTM.EXE is also required in the program directory. The program is very easy to use and mostly self-explaining. No special hardware or software is needed to run the program, except for some basic requirements:

- 486 processor or above (Pentium or above strongly recommended)
- Microsoft operating system, DOS version 5.0 or above (Windows works fine, both NT, 95/98 and 2000[6])
- VGA color graphics

Even if a single calculation will execute very quickly, iterations over a large parameter set may take hours to complete. As an example, the data used for the results presented in this report took over 9 hours to compute on a Pentium 133 MHz. Therefore methods for increasing the computation speed have been implemented, involving turning off unnecessary program parts – most notably the graphical presentation.

The fit algorithm used in the program is the Levenberg–Marquardt method for non-linear least squares fitting [1, 2]. The code was translated into Pascal from a translation by Steve Moshier into C from the original Fortran code of the MINPACK package from Argonne National Laboratories.

To obtain a copy of the program, please contact the author by email at `blom@teorfys.lu.se`.

A.1 Program parameters

The program parameters which can be changed are (here presented as they appear in the menu for changing them; for the basic definitions see page 2):

- `SNR`
  Signal-to-noise ratio
- `WIDTH`
  FWHM of the spectral line (in cm^{-1})
- `DX`
  Sampling distance or resolution interval (in cm^{-1})
- `PROFILE  Lorentz / Gauss`
- `POINTS`
  Number of sample points – sets the desired wavenumber interval. Example: `POINTS`=100, `SIGMA0` = 16000 and `DX`=0.010 will make the wavenumber interval [15999.50,16000.50]. A high value will *not* increase the accuracy of the calculations but only slow the program down. However, unless a high enough value is set, erratic results may be produced if the spectral line does not fit completely inside the interval. As a rule of thumb, the interval should be at least ten times the FWHM of a Lorentzian, perhaps slightly less if the SNR is very large. For a Gaussian, five times is enough.

[6] There is a problem when running Windows 2000 in terminal mode, since this does not give access to full-screen text mode. The program will not be able to run under this special circumstance.

- `SIGMAO`
  Center wavenumber of the interval (in cm^{-1}). This parameter is not really used in the program, but is included for completeness of the physical picture.

- `SHOW    yes / no`
  If set to `yes`, then each iteration will present the spectra graphically on the screen when running in batch or multi mode (see section A.2). If set to `no`, only an iteration counter is showed. Setting this parameter to `no` will increase calculation speed by a large amount, especially if the machine has slow graphics.

- `MULTI-ITER`
  How many iterations to run in multi mode for each SNR.

- `STEP   DX / SNR`
  In multi mode, step in dx or SNR.

- `STEP_FROM`
  In multi mode, start dx or SNR at this value.

- `STEP_TO`
  In multi mode, end at this value.

- `STEP_STEP`
  In multi mode, increase by this amount in each step.

- `STEP_MODE   LIN / LOG`
  In multi mode, increase linearly or logarithmically in each step. If linear, then STEP_STEP is simply added to the previous value in each step; if logarithmic then the value x ($=dx$ or SNR) is updated in the following way: $\log x_{N+1} = \log x_N +$STEP_STEP. Logarithmic example: STEP_STEP=0.5 will step through the values 0.01, 0.05, 0.1, 0.5, 1, 5, 10, etc.

- `TOL`
  Fit tolerance level. The actual fit tolerance is 10^{-x} if x is the tolerance level. A higher number means convergence criteria are stricter. A value of 5–9 is acceptable for testing, making the program speed up, but for sharp data runs the maximum value 14 is recommended.

- `FILENAME`
  Name of output file. If a non-valid filename is entered, such as a non-existing drive, the program will not detect it until it tries to create the file. Then an error message will be shown, and the program returns to normal running. *Note that the program overwrites, without prompting, any existing file with this name.*

A.2 Commands and features

Available commands and keys to press:

F2 **Change parameters**
This invokes a menu where the parameters of section A.1 can be changed. Simply walk around in the menu using the arrow keys, and edit any entry. Upon entrance, each entry will be inverted, and any typing will erase the previous value and replace it with the input. To edit the value without erasing the contents, first press the left or right arrow key. Other available keys are:

- END / HOME: bring the cursor to the end/beginning of the line
- CTRL+END / CTRL+HOME: bring the cursor to the last/first entry in the menu
- CTRL+DEL: delete the entire active entry
- INS: toggle insert mode on/off (default is off)
- F1: display/undisplay help regarding each menu entry
- F10: save changes and exit
- ESC: discard changes and exit

Some entries are toggle fields, which means that only two values are possible (such as yes or no). In these cases the desired value is entered by pressing the first letter of it (such as G for Gaussian) or use the left and right arrow keys to flip between the two possibilities.

Any typing errors will be detected by the program (such as the entering of a negative value for SNR, or a non-numeric value for a numeric parameter) when F10 is pressed. If an error is detected, an error message is displayed and the cursor placed on the offending entry. It is not possible to exit the menu (and save the changes) unless all inputs are valid. Pressing ESC will always exit the menu, no questions asked, but changes are lost.

F7 **Batch mode**

In this mode the program generates a spectral line once, and then iterates – virtually endlessly[7] – the generation of random noise and fitting to the synthetic spectrum. If SHOW is set to yes then each iteration shows the generated and fitted spectra on the screen and displays an iteration counter plus statistics. The iterations are interrupted by pressing ESC. When this is done, the accumulated data will be exported to a file, with the name given by the parameter FILENAME.

The output file is an ASCII text file which includes a header, parameters used to generate the data and the following statistics:

- True area of the spectral line
- Mean area of the fits
- Standard deviation of the fitted areas
- True center position of the spectral line
- Mean center position of the fits
- Standard deviation of the fitted center positions
- Number of performed iterations

Thereafter follows for each iteration the actual value of the fitted area, and later on in the file a similar list of the fitted positions.

This mode is primarily intended for single calculations, when data is desired for a specific combination of SNR, FWHM and resolution.

F8 **Multi mode**

This mode is similar to the previous, with the exception that it runs not only several iterations for the same generated spectral line (each time with different noise), but further does this for several values of SNR or dx – which one is changed is determined by the parameter STEP.

[7] There is an upper limit of 5000 iterations, but to reach this the program will have to run for a very long time.

The number of iterations is set by the parameter MULTI-ITER; the parameters STEP_MODE, STEP_FROM, STEP_TO and STEP_STEP guide the value range to use. Example: STEP_FROM=3, STEP_TO=10 and STEP_STEP=2 will generate data for SNR=3, 5, 7 and 9. The output file format is also similar to the one from batch mode, except that the individual values of fitted areas and positions are omitted, and instead the data is presented in labelled columns (same as listed above, plus SNR or dx), one line for each value of SNR or dx.

ENTER **Iterate**
This is the simplest feature of the program; it generates noise and fits a profile to the synthetic spectrum, i.e. the same which is done in one iteration in batch mode. The same spectral line is used over and over. To change the spectral line, press S.

F5 The option SHOW (see above) is also available through this keyboard shortcut. Press F5 to toggle the option on/off.

S Pressing S will cause the program to re-initialize all parameters.

Q To quit the program, press Q.

A.3 Initilization files and default parameters

If the program is called with a command line parameter, this parameter is assumed to be the name of an initialization file. This file (which need not reside in the same folder as the program) contains values for the different parameters. Not all parameters need to be given, but only the ones wanted to be changed from the default values (see below how to change these). The program will read the file and set the parameters accordingly, and start as normal.

The format of each line of the file must be as follows:

```
PARAMETER = value
```

Only one parameter may be specified per line, and they must be entered exactly as given in section A.1 – note the underscores – although upper or lower case is optional. Blank lines and spaces and tabs are ignored, as are lines starting with a semicolon (;), which is useful for temporarily leaving out a parameter without deleting that line. If the file contains unrecognized commands or the given values are unallowed (such as entering a negative FWHM), the program will give an error message, stating the offending line number in the file.

There is one additional command for the initilization file which is not available from the program menu. It is

- AUTO yes / no
  See section A.4 for details.

To change the default program parameters, simply put a parameter file with the above given format (the AUTO command is not available for this file) in the directory from which the program is run (need not be the same directory as the executable itself resides in) and name it DEFAULT.INI. If the program is then called with a command line parameter, parameters specified in that initilization file will be used instead of the ones in the default-file. Note that not all parameters need to be specified in DEFAULT.INI; those omitted will take program basic default values.

A.4 Auto mode

The parameter AUTO is used to enter auto mode, which is only accessible from an initialization file. If this parameter is set to yes[8], the program will not produce any graphics on the screen, but run in multi mode as specified by the parameters MULTI-ITER, STEP_FROM etc., save the results to the file specified by FILENAME and then quit the program. This way a powerful multi data run can be set up by creating a batch-file, e.g. with the following contents:

```
ftserror run1.ini
ftserror run2.ini
ftserror run3.ini
```

where the ini-files each contain specific parameter settings and, very importantly, a *unique filename* for each output file, since the program always overwrites existing files.

While running in auto mode, the program will present an iteration counter and show the SNR or dx interval. Upon completion, a message is printed to what file the data was saved. In auto mode, the program may be interrupted by pressing ESC, which will terminate the present program run. Normally the program will exit with the DOS environment variable errorlevel set to 0, but if ESC is pressed in auto mode, it will be set to 1. This way the whole batch-file may be terminated by testing for this case:

```
ftserror run1.ini
if errorlevel=1 goto end
ftserror run2.ini
if errorlevel=1 goto end
ftserror run3.ini
:end
```

[8] Setting it to "no" has no effect, i.e. is equal to omitting the AUTO parameter from the file altogether.