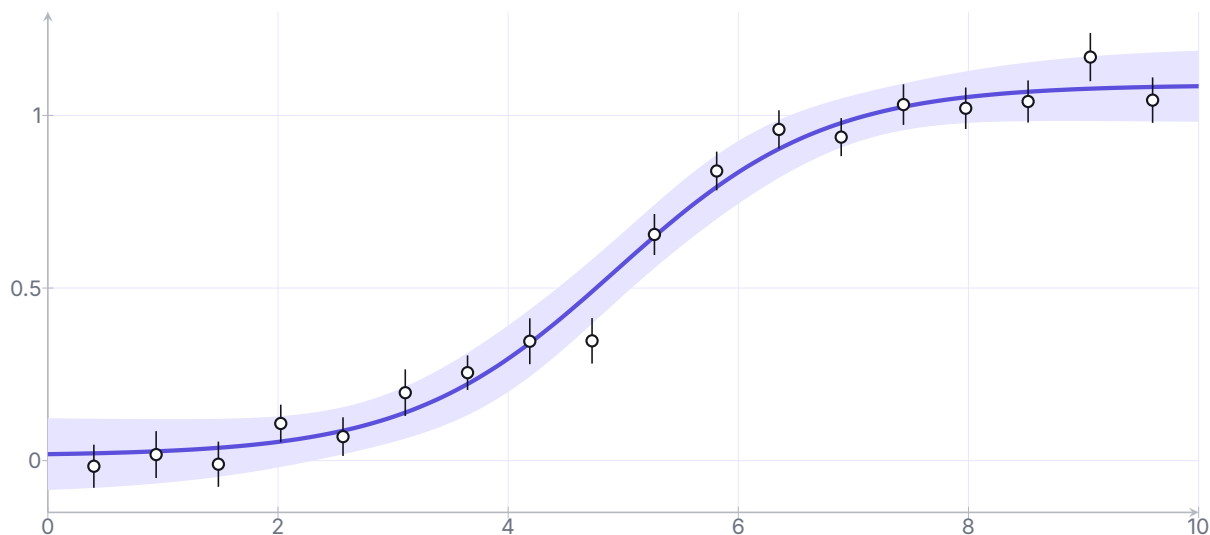


REFERENCE MANUAL

TrollFit

Methods, models and numerics behind every result

A complete mathematical description of how TrollFit fits data, estimates uncertainties, tests its fits and reads data from images — written so that every number the app shows can be checked by hand.



About this manual

TrollFit is a tool for scientific data analysis, and results that end up in a thesis, a paper or a laboratory report have to be traceable. This manual therefore describes *exactly* what the program computes: every model, every estimator, every test statistic and every numerical safeguard, with the formulas that the code implements and the constants it uses. Nothing in TrollFit is a black box — if a number appears on the screen, this document says where it comes from.

The manual is a reference, not a tutorial. It assumes familiarity with basic statistics (standard deviation, the normal and Student distributions, least squares). Short **In the app** boxes connect the mathematics to the controls in the user interface.

What this manual covers

- data input, measurement uncertainties in y and in x , and data transformations (chapter 2);
- the weighted least-squares framework, the Levenberg–Marquardt algorithm and the estimation of parameter uncertainties (chapter 3);
- all built-in models, the Fourier series and custom equations (chapter 4);
- fixed parameters and parameter limits (chapter 5), orthogonal distance regression (chapter 6) and global fits (chapter 7);
- residual diagnostics, confidence and prediction bands, inverse prediction, integrals and derivatives (chapters 8–9);
- model comparison with AICc (chapter 10);
- the graph digitizer (chapter 11);
- numerical methods and the validation of TrollFit against reference software (chapter 12).

Version. This edition describes TrollFit 0.5.0; the version shown in the header of the app tells you which edition applies.

Citing TrollFit. If TrollFit contributed to published work, please cite it — recommended forms, including a BibTeX entry, are given in appendix A. Citation is appreciated, never required.

Feedback. If you believe a formula here, or a result of the program, is wrong, please write to info@trollsoftworks.com. Reports of this kind are treated as the most important bug reports there are.

© 2026 Šimon Svoboda (Troll Softworks). This manual may be freely copied in unmodified form. TrollFit is proprietary software provided without warranty; it is not intended for safety-critical use. See trollsoftworks.com/terms.

Contents

About this manual	i
1. Principles and notation	1
1.1. Design principles	1
1.2. Notation	2
1.3. How a fit proceeds	2
2. Data and measurement uncertainties	3
2.1. Datasets and points used in a fit	3
2.2. Uncertainties in y	3
2.3. Uncertainties in x	4
2.4. Transforming data	4
2.5. Swapping axes	5
3. Least squares and parameter uncertainties	6
3.1. The estimation problem	6
3.2. Models linear in the parameters	6
3.3. The Levenberg–Marquardt algorithm	7
3.4. Starting values and multiple starts	8
3.5. Covariance matrix and standard uncertainties	8
3.6. Goodness of fit	9
3.7. Parameter correlations	9
3.8. Presentation of results	9
4. Models	10
4.1. Polynomial models	10
4.2. Exponential, power and saturation models	11
4.3. Peak models	12
4.4. Fourier series	13
4.5. Custom equations	15
5. Fixed parameters and parameter limits	17
5.1. Formulation	17
5.2. Algorithm	17
5.3. Uncertainties and degrees of freedom	18
6. Uncertainties in x: orthogonal distance regression	19
6.1. The problem	19
6.2. Algorithm	20
6.3. Uncertainties	20
7. Global fits of several datasets	22
7.1. Formulation	22
7.2. Is sharing consistent with the data?	22
8. Checking a fit: residual diagnostics	24
8.1. Standardised residuals	24

8.2. Outliers	24
8.3. Runs test: are the residuals random?	25
8.4. Normality of residuals: Anderson–Darling test	25
9. Bands and derived quantities	26
9.1. Confidence band	26
9.2. Prediction band	26
9.3. Evaluating the fit and inverse prediction	26
9.4. Integral	27
9.5. Derivative	27
9.6. Combining fitted functions	28
10. Comparing models	29
10.1. Definition	29
10.2. Interpretation	29
11. Digitizing graphs	30
11.1. Axis calibration	30
11.2. Ink detection	31
11.3. Snapping a click	31
11.4. Finding similar markers	31
11.5. Tracing a curve	32
11.6. Accuracy	32
12. Numerical methods and validation	34
12.1. Special functions	34
12.2. Linear algebra	34
12.3. Reproducibility	34
12.4. Validation	35
12.5. Limitations	35
A. How to cite TrollFit	36
A.1. Citing the software	36
A.2. Citing this manual	36
A.3. Describing the analysis	37
Bibliography	38

Principles and notation

1.1 Design principles

TrollFit runs entirely in the web browser: data never leave the user's computer and all computation is done locally in double-precision (IEEE 754 binary64) arithmetic. Four principles shape the numerical core; they are referred to throughout the manual.

Scale invariance.

Results must not depend on the units of the data. A fit of x in nanometres and y in microvolts must give the same answer as the same data in metres and volts, up to the unit conversion. TrollFit therefore never compares a quantity with an *absolute* threshold such as $\det(\mathbf{A}) < 10^{-14}$. Linear problems are solved by a QR decomposition with scaled columns, nonlinear problems with a Jacobian-scaled Levenberg–Marquardt method, and covariance matrices are inverted after scaling their diagonal to one. All tolerances are relative.

Proper nonlinear fitting.

Models that are nonlinear in their parameters (exponential, power law, Gaussian, ...) are fitted by nonlinear least squares on the original scale of y . Linearising transformations such as $\ln y$ are used only to obtain starting values, because they distort the error distribution and give biased parameters and incorrect uncertainties.

Honest uncertainties.

Every parameter is reported with its standard uncertainty. When the uncertainty cannot be determined (for example when the number of points equals the number of parameters) it is shown as n/a , never as a misleading ± 0 .

Determinism.

The same input always produces the same output. Where randomness is used (random starting values for custom equations) it comes from a generator with a fixed seed.

1.2 Notation

Symbol	Meaning
n	number of data points used in the fit (checked rows with valid x and y)
p	number of <i>free</i> fitted parameters
$\nu = n - p$	degrees of freedom
x_i, y_i	data point i , $i = 1, \dots, n$
σ_i	standard uncertainty of y_i (written $\sigma_{y,i}$ when x -uncertainties are also present)
$\sigma_{x,i}$	standard uncertainty of x_i
$w_i = 1/\sigma_i^2$	statistical weight of point i ; $w_i = 1$ for an unweighted fit
$\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^\top$	parameter vector; $\hat{\boldsymbol{\beta}}$ its estimate
$f(x; \boldsymbol{\beta})$	model function
$\hat{y}_i = f(x_i; \hat{\boldsymbol{\beta}})$	fitted value
$r_i = y_i - \hat{y}_i$	residual
$\mathbf{J}$	Jacobian, $J_{ij} = \partial f(x_i; \boldsymbol{\beta}) / \partial \beta_j$
$\mathbf{W} = \text{diag}(w_1, \dots, w_n)$	weight matrix
$S(\boldsymbol{\beta}) = \sum_i w_i r_i^2$	weighted sum of squared residuals; χ^2 when the σ_i are absolute
s^2	residual variance factor (section 3.5)
$\mathbf{C}$	covariance matrix of $\hat{\boldsymbol{\beta}}$
$u(\cdot)$	standard uncertainty
$t_q(\nu)$	q -quantile of Student's t distribution with ν degrees of freedom
$z_q = \Phi^{-1}(q)$	q -quantile of the standard normal distribution

Logarithms written $\ln$ are natural; $\log_{10}$ is used explicitly where needed. Vectors are bold, matrices are sans-serif.

1.3 How a fit proceeds

For every dataset the same pipeline is followed:

1. The checked rows of the table are collected; their uncertainties σ_y and σ_x are converted to weights (chapter 2).
2. The chosen model is fitted without restrictions (chapters 3 and 4).
3. If parameters are fixed or limited, the unrestricted solution is used as a start for a constrained fit (chapter 5).
4. If x -uncertainties are present, the result is refined by orthogonal distance regression (chapter 6).
5. Uncertainties, goodness-of-fit statistics and residual diagnostics are computed (sections 3.5–3.6, chapter 8).

If a global fit is active, steps 2–5 are replaced by the simultaneous fit of chapter 7.

Data and measurement uncertainties

2.1 Datasets and points used in a fit

A project contains up to five datasets. Each dataset has its own table with columns x , y and, optionally, the uncertainties σ_y and σ_x ; each has its own model and fit. A row takes part in the fit only if (i) its checkbox is ticked and (ii) both x and y parse as finite numbers. Unticked rows remain visible in the chart as hollow markers. Points can also be (de)selected by dragging a rectangle in the chart. The number of points used, n , is shown with the results as *Points used: n / total*.

Number format. Numbers are read with `parseFloat` after replacing a decimal comma by a decimal point, so 2,5 and 2.5 are equivalent; scientific notation (1.6e-19) is accepted. When text is pasted or a text file is read, each line is split into columns by the first rule that applies: tab; semicolon; whitespace; comma. Hence in a line that contains whitespace, a comma is always taken as a decimal separator. Empty lines and lines starting with `#` are ignored; if the first line is not numeric, it is used for axis titles. Text files from instruments (`.xy`, `.xye`, `.asc`, `.prn`, `.chi`, `.dpt`, ...) often begin with lines of metadata; the data start at the first place where at least three consecutive lines (or all remaining ones) consist of ≥ 2 numeric columns. The line directly above is kept as the axis titles only if it is entirely non-numeric text; everything before it is skipped and the number of skipped lines is reported. Spreadsheets (`.xlsx`, `.xlsm`, `.ods`) are read as tables in the same way. Data with more than two columns open an import wizard in which the columns for x , y , σ_y and σ_x are chosen explicitly.

2.2 Uncertainties in y

When the σ_y column is switched on, each point carries an uncertainty that is entered either as an absolute value (in the units of y) or as a percentage q_i of the value:

$$\sigma_i = \sigma_{y,i} \quad (\text{absolute}), \quad \sigma_i = \frac{q_i}{100} |y_i| \quad (\text{percent}). \quad (2.1)$$

The weights are $w_i = 1/\sigma_i^2$. A weighted fit is performed only if *every* point used in the fit has a valid, strictly positive σ_i . If even one value is missing or not positive, TrollFit does not guess: the fit is carried out without weights and the results panel says so.

2.2.1. Absolute uncertainties and relative weights

The σ_i can be interpreted in two ways, selected by the option **sy as relative weights**:

Absolute (default).

The σ_i are true standard deviations of the measurements. The parameter covariance follows from them directly ($s^2 = 1$ in eq. (3.4)), the statistic $S(\hat{\beta})$ is a χ^2 with ν degrees of freedom if the model is correct, and critical values for intervals come from the normal distribution.

Relative.

The σ_i are known only up to a common factor (for example rough estimates). They determine the *relative* weights of the points, while the overall scale of the uncertainties is estimated from the scatter of the residuals ($s^2 = S/\nu$), and critical values come from Student's t distribution. This is the behaviour of `scipy.optimize.curve_fit` with `absolute_sigma=False`.

Without σ_y all weights are one and the fit is treated like the relative case.

2.3 Uncertainties in x

When the σ_x column is switched on (which requires σ_y as well), each point may carry an uncertainty $\sigma_{x,i} \geq 0$ of its abscissa. An empty cell or zero means that x_i is exact. With at least one positive $\sigma_{x,i}$ the fit is computed by orthogonal distance regression (chapter 6). The σ_x values are always absolute (in the units of x); the option *relative weights* affects the scale of the covariance in the same way as above.

In the app. σ_x is not used by the Fourier series and by global fits; the results panel then states that the fit was done with σ_y only.

2.4 Transforming data

The **Transform data** dialog computes new columns from formulas $x' = g(x, y)$ and $y' = h(x, y)$ (the variables **x**, **y**, **sy**, **sx** may be used; **ln** is accepted as a synonym of the natural logarithm). Uncertainties are propagated to first order (the law of propagation of uncertainty, [11]). With σ_y only,

$$\sigma_{y'} = \left| \frac{\partial h}{\partial y} \right| \sigma_y, \quad (2.2)$$

and with both σ_x and σ_y each new coordinate receives the contributions of both original uncertainties,

$$\sigma_{y'} = \sqrt{\left(\frac{\partial h}{\partial x} \sigma_x \right)^2 + \left(\frac{\partial h}{\partial y} \sigma_y \right)^2}, \quad \sigma_{x'} = \sqrt{\left(\frac{\partial g}{\partial x} \sigma_x \right)^2 + \left(\frac{\partial g}{\partial y} \sigma_y \right)^2}. \quad (2.3)$$

Percentage uncertainties are first converted to absolute ones. The derivatives are evaluated by central differences with step $h = \max(10^{-6}|v|, 10^{-3}\sigma_v)$, where v is the variable being differentiated. The correlation between x' and y' that arises when both depend on the same original variable is neglected. Results are stored with ten significant digits, propagated uncertainties with six.

Note. A transformation such as $y' = \ln y$ is useful for exploring data, but fitting a straight line to transformed data is *not* equivalent to fitting the original model. For quantitative results, fit the nonlinear model directly (section 4.2.1).

2.5 Swapping axes

Swap $x \leftrightarrow y$ exchanges the two columns. If both σ_x and absolute σ_y are in use, the two uncertainty columns are exchanged as well; in all other cases the uncertainty columns stay unchanged, so check them after swapping.

Least squares and parameter uncertainties

3.1 The estimation problem

Except for orthogonal distance regression (chapter 6), every fit in TrollFit minimises the weighted sum of squared residuals

$$S(\beta) = \sum_{i=1}^n w_i (y_i - f(x_i; \beta))^2, \quad \hat{\beta} = \arg \min_{\beta} S(\beta), \quad (3.1)$$

with $w_i = 1/\sigma_i^2$ (weighted fit) or $w_i = 1$ (unweighted fit). If the errors of y_i are independent and normally distributed with standard deviations σ_i , $\hat{\beta}$ is the maximum-likelihood estimate [4, 24].

Two cases are distinguished. Models that are *linear in the parameters* (straight line, polynomials, logarithmic model, Fourier series with a known period) have a closed-form solution, computed by a QR decomposition. All other models are fitted iteratively by the Levenberg–Marquardt algorithm.

3.2 Models linear in the parameters

If $f(x; \beta) = \sum_j \beta_j \phi_j(x)$, the problem is $\min \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{X}\beta)\|^2$ with the design matrix $X_{ij} = \phi_j(x_i)$. TrollFit solves it without forming the normal equations, whose condition number is the square of that of $\mathbf{X}$:

1. Form $\mathbf{A} = \mathbf{W}^{1/2}\mathbf{X}$ and $\mathbf{b} = \mathbf{W}^{1/2}\mathbf{y}$.
2. Scale the columns to unit length: $c_j = \|\mathbf{A}_{:,j}\|_2$, $\mathbf{A} \leftarrow \mathbf{A}\mathbf{D}^{-1}$ with $\mathbf{D} = \text{diag}(c_1, \dots, c_p)$. This makes the problem independent of the units of the parameters.
3. Compute the Householder QR decomposition $\mathbf{A} = \mathbf{Q}\mathbf{R}$ and $\mathbf{z}$ from $\mathbf{R}\mathbf{z} = (\mathbf{Q}^T\mathbf{b})_{1:p}$ by back substitution.
4. Unscale: $\hat{\beta} = \mathbf{D}^{-1}\mathbf{z}$, and

$$(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1} = \mathbf{D}^{-1}\mathbf{R}^{-1}\mathbf{R}^{-T}\mathbf{D}^{-1}. \quad (3.2)$$

The design is declared singular (*the data do not determine all parameters of this model*) if a column has zero norm or if $|R_{kk}| \leq 10^{-12} \max_j |R_{jj}|$.

Centring. For polynomial-type models the basis functions are evaluated in a centred (and for higher-degree polynomials also scaled) variable t , which keeps the problem well conditioned even for data such as $x \approx 2000$ (years). The coefficients and their covariance are then transformed back to powers of x , see section 3.5.1.

3.3 The Levenberg–Marquardt algorithm

Nonlinear models are fitted by the Levenberg–Marquardt (LM) method [16, 18, 20]. At the current estimate β the model is linearised, $f(x_i; \beta + \delta) \approx f(x_i; \beta) + \mathbf{J}_{i,:} \delta$, and the step δ solves the damped normal equations

$$(\mathbf{A} + \lambda \text{diag}(\mathbf{A})) \delta = \mathbf{g}, \quad \mathbf{A} = \mathbf{J}^T \mathbf{W} \mathbf{J}, \quad \mathbf{g} = \mathbf{J}^T \mathbf{W} \mathbf{r}. \quad (3.3)$$

Scaled solution. Equation (3.3) is solved in scaled form: with $d_j = A_{jj}^{-1/2}$ (or $d_j = 1$ if $A_{jj} = 0$) the matrix $\tilde{\mathbf{A}} = \mathbf{D} \mathbf{A} \mathbf{D}$ has a unit diagonal, which is replaced by $1 + \lambda$ (or by λ for a parameter without influence on the model, $A_{jj} = 0$). Then $\tilde{\mathbf{A}} \mathbf{z} = \mathbf{D} \mathbf{g}$ is solved by Gauss–Jordan elimination with partial pivoting and $\delta = \mathbf{D} \mathbf{z}$. The scaling makes the method invariant to the units of the parameters.

Step control. The damping starts at $\lambda = 10^{-3}$. A trial step is accepted only if it decreases S ; then $\lambda \leftarrow \max(\lambda/10, 10^{-12})$. Otherwise $\lambda \leftarrow 10\lambda$ and the step is recomputed; when λ reaches 10^{12} without an improvement, the iteration stops.

Stopping rules. The iteration ends when any of the following holds:

- the relative decrease of S in an accepted step is below 10^{-12} ;
- the largest relative parameter change $\max_j |\delta_j|/|\beta_j|$ is below 10^{-10} ;
- *stagnation*: over the last 25 iterations S decreased by no more than $10^{-8} S$ (this ends the slow creeping typical of degenerate models);
- *exact data*: $S \leq 10^{-26} S_{\text{ref}}$, where $S_{\text{ref}} = \sum_i w_i (y_i - \bar{y}_w)^2$ and $\bar{y}_w$ is the weighted mean (a relative floor, so noise-free test data do not run to the iteration limit);
- a non-finite Jacobian, or the iteration limit of the model (400–1000 iterations).

Large datasets. For $n > 6000$ the iteration first runs on an evenly spaced subsample of every k -th point, $k = \lceil n/2500 \rceil$ (at most 300 iterations), and is then continued on all points for at most 40 iterations, starting from the subsample solution if that is better than the original start. The final result is a minimum of S on the full data; the subsample only speeds up the approach to it.

3.4 Starting values and multiple starts

Nonlinear least squares can have several local minima. For every nonlinear model TrollFit derives one or more starting points from the data (they are listed with each model in chapter 4), runs LM from each, and keeps the solution with the smallest S . Where a model has equivalent parameterisations (for example σ and $-\sigma$ of a Gaussian), the solution is normalised to a canonical form after each run. For large datasets ($n > 6000$) with several starts, every start is first tried for 40 iterations on the subsample and only the most promising one is iterated to convergence on all data.

3.5 Covariance matrix and standard uncertainties

At the solution, the covariance matrix of the parameters is estimated as

$$\mathbf{C} = s^2 (\mathbf{J}^T \mathbf{W} \mathbf{J})^{-1}, \quad s^2 = \begin{cases} 1 & \text{absolute } \sigma_y, \\ S(\hat{\beta})/\nu & \text{relative } \sigma_y \text{ or no } \sigma_y, \end{cases} \quad (3.4)$$

with $\nu = n - p$. For linear models $\mathbf{J} = \mathbf{X}$ and the inverse comes from eq. (3.2). The standard uncertainties of the parameters are $u(\hat{\beta}_j) = \sqrt{C_{jj}}$. This is the same estimate as `pcov` returned by `scipy.optimize.curve_fit` (with `absolute_sigma` set accordingly).

Numerical safeguards. The matrix $\mathbf{J}^T \mathbf{W} \mathbf{J}$ is inverted after scaling its diagonal to one, by Gauss–Jordan elimination with partial pivoting and a *relative* pivot threshold of 10^{-13} . If the matrix is singular, if a diagonal element is zero (a parameter does not influence the model) or if $\nu \leq 0$ in the relative case, the uncertainties are reported as n/a .

3.5.1. Reparameterisation

When a model is fitted in an internal parameterisation γ and reported in $\beta = \mathbf{T}\gamma$ (a linear map), the covariance is transformed exactly:

$$\mathbf{C}_\beta = \mathbf{T} \mathbf{C}_\gamma \mathbf{T}^T. \quad (3.5)$$

For example, the straight line is fitted as $y = A(x - \bar{x}) + B$ and reported as $y = ax + b$ with $a = A$, $b = B - A\bar{x}$, i.e. $\mathbf{T} = \begin{pmatrix} 1 & 0 \\ -\bar{x} & 1 \end{pmatrix}$. Bands and derived quantities are computed in the internal parameterisation, which is numerically more stable and mathematically equivalent.

3.5.2. Coverage factors

Parameters are reported as value $\pm$ standard uncertainty (coverage factor $k = 1$). Where TrollFit shows 95 % intervals — confidence and prediction bands, the evaluation, integral and derivative tools — the half-width is $k u$ with

$$k = z_{0.975} = 1.959964 \text{ (absolute } \sigma_y), \quad k = t_{0.975}(\nu) \text{ (otherwise)}. \quad (3.6)$$

If $\nu \leq 0$ the interval is not available.

3.6 Goodness of fit

Coefficient of determination.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}. \quad (3.7)$$

R^2 is always computed from *unweighted* residuals. For models other than the straight line it is a descriptive measure only (it is not the square of a correlation coefficient and can be negative for a poor fit). If all y_i are equal, i.e. $\sum (y_i - \bar{y})^2 \leq 10^{-28} \sum y_i^2$, R^2 is set to 1 for a perfect fit and to 0 otherwise. Values above 0.99 are displayed with enough digits to distinguish fits (0.9999987 rather than 1.0000).

Chi-square. For a weighted fit TrollFit shows χ^2/ν with $\chi^2 = S(\hat{\beta}) = \sum_i w_i r_i^2$. With absolute uncertainties and a correct model, χ^2 follows a χ^2 distribution with ν degrees of freedom, so $\chi^2/\nu \approx 1$ is expected; values much larger indicate an inadequate model or underestimated uncertainties, values much smaller overestimated uncertainties.

3.7 Parameter correlations

The correlation matrix of the parameters,

$$\rho_{jk} = \frac{C_{jk}}{\sqrt{C_{jj}C_{kk}}}, \quad (3.8)$$

is shown as a colour map. Pairs with $|\rho_{jk}| \geq 0.95$ are highlighted: such parameters can largely compensate each other, so the individual uncertainties are large while certain combinations are well determined. Fixed parameters have zero variance and no correlation.

3.8 Presentation of results

A value with uncertainty is rounded according to the uncertainty [11]: the uncertainty is rounded to two significant digits and the value to the same decimal place. For $\hat{\beta} = 2.34567$, $u = 0.01234$ the display is 2.346 ± 0.012 . A common power of ten is factored out when $\max(|\hat{\beta}|, u) < 10^{-3}$ or $\geq 10^5$, e.g. $(1.602 \pm 0.011) \cdot 10^{-19}$. Numbers without uncertainty are shown with six significant digits. Unrounded values (all digits of the double-precision result) are available through **Copy results** in the *Excel table* format.

Models

Table 4.1 lists the built-in models. For each of them this chapter gives the model function, the fitting method, the minimum number of points, the starting values of nonlinear fits and any derived quantities. All nonlinear models use analytic Jacobians, which are stated explicitly.

Table 4.1.: Built-in models. *Method:* L = linear least squares (QR, section 3.2), N = Levenberg–Marquardt (section 3.3).

Model	Function	Method	Min. n
Linear	$y = ax + b$	L	2
Quadratic	$y = ax^2 + bx + c$	L	3
Polynomial (degree m , $1 \leq m \leq 8$)	$y = a_0 + a_1x + \dots + a_mx^m$	L	$m + 1$
Logarithmic	$y = a \ln x + b, \quad x > 0$	L	2
Exponential	$y = a e^{bx}$	N	2
Power	$y = a x^b, \quad x \geq 0$	N	2
Rational	$y = (ax + b)/(cx + 1)$	N	3
Michaelis–Menten	$y = V_{\max} x/(K_m + x)$	N	2
Sigmoid (logistic)	$y = y_0 + A/(1 + e^{-k(x-x_0)})$	N	4
Gauss ($N = 1, 2, 3$ peaks) [†]	$y = \sum_{k=1}^N A_k e^{-(x-\mu_k)^2/(2\sigma_k^2)} + B(x)$	N	5, 8, 11
Lorentz [†]	$y = A\gamma^2/((x-x_0)^2 + \gamma^2) + B(x)$	N	4
Fourier series	$y = a_0 + \sum_{k=1}^H (a_k \cos k\omega x + b_k \sin k\omega x)$	L / N	$2H + 3$
Custom equation	any expression $f(x; \beta)$	N	p

[†] Peak models: background $B(x) = c$ (constant, default) or $B(x) = c + mx$ (linear, section 4.3.3); the linear background needs one point more.

4.1 Polynomial models

4.1.1. Linear

The straight line is fitted as $y = A(x - \bar{x}) + B$ with $\bar{x} = \frac{1}{n} \sum x_i$ (the unweighted mean) by the QR method, and reported as $a = A$, $b = B - A\bar{x}$ with the covariance transformed as in section 3.5.1. With fixed or limited parameters, the line is fitted by the general method of chapter 5; fixing $b = 0$ gives a line through the origin.

4.1.2. Quadratic

Fitted as $y = At^2 + Bt + C$ with $t = x - \bar{x}$; then

$$a = A, \quad b = B - 2A\bar{x}, \quad c = C - B\bar{x} + A\bar{x}^2, \quad \mathbf{T} = \begin{pmatrix} 1 & 0 & 0 \\ -2\bar{x} & 1 & 0 \\ \bar{x}^2 & -\bar{x} & 1 \end{pmatrix}. \quad (4.1)$$

4.1.3. Polynomial of degree m

To keep the design matrix well conditioned for any degree, the basis is formed in the standardised variable

$$t = \frac{x - \bar{x}}{s_x}, \quad s_x = \max_i |x_i - \bar{x}|, \quad (4.2)$$

so that $t \in [-1, 1]$. With the fitted coefficients c_k of t^k , the coefficients of x^j follow from the binomial theorem,

$$a_j = \sum_{k=j}^m c_k \binom{k}{j} \frac{(-\bar{x})^{k-j}}{s_x^k}, \quad (4.3)$$

which is a linear map $\mathbf{a} = \mathbf{T}\mathbf{c}$; the covariance is transformed accordingly. The curve, its bands and derived quantities are evaluated in t .

4.1.4. Logarithmic

A straight line in $\ell = \ln x$: fitted as $y = A(\ell - \bar{\ell}) + B$, reported as $a = A$, $b = B - A\bar{\ell}$. All x_i must be positive.

4.2 Exponential, power and saturation models

4.2.1. Exponential

$$f = a e^{bx}, \quad \partial f / \partial a = e^{bx}, \quad \partial f / \partial b = a x e^{bx}.$$

The fit is a true nonlinear least-squares fit on the original scale of y . *Start*: let $s = +1$ if at least half of the y_i are positive and $s = -1$ otherwise. Using the points with $s y_i > 0$ (at least two are required), the straight line $\ln(s y_i) = \ln|a| + b x_i$ is fitted with weights $w_i y_i^2$, which accounts for $\text{Var}(\ln y) \approx \sigma^2 / y^2$. This gives $a_0 = s e^{\ln|a|}$, b_0 . LM is then run from (a_0, b_0) ; should it not improve S , the start is kept.

4.2.2. Power law

$$f = a x^b, \quad \partial f / \partial a = x^b, \quad \partial f / \partial b = a x^b \ln x \text{ (taken as 0 at } x = 0\text{)}.$$

All $x_i \geq 0$ are required. *Starts*: (i) a weighted straight line in log-log coordinates through the points with $x_i > 0$ and y_i of the majority sign, weights $w_i y_i^2$; (ii) $(a, b) = (\max y, 1)$.

4.2.3. Rational function

$$f = \frac{ax + b}{cx + 1}, \quad \frac{\partial f}{\partial a} = \frac{x}{cx + 1}, \quad \frac{\partial f}{\partial b} = \frac{1}{cx + 1}, \quad \frac{\partial f}{\partial c} = -\frac{(ax + b)x}{(cx + 1)^2}.$$

Starts: (i) the linearisation $y(cx + 1) = ax + b$, i.e. the linear least-squares problem $y_i = ax_i + b - c x_i y_i$ (with the fit weights); (ii) the straight line ($c = 0$). The better LM solution is kept. If the pole $x = -1/c$ lies within $[\min x_i, \max x_i]$, the fit is rejected with an explanation.

4.2.4. Michaelis–Menten

$$f = \frac{V_{\max} x}{K_m + x}, \quad \frac{\partial f}{\partial V_{\max}} = \frac{x}{K_m + x}, \quad \frac{\partial f}{\partial K_m} = -\frac{V_{\max} x}{(K_m + x)^2}.$$

Starts: (i) the Hanes–Woelf linearisation [12] $x/y = K_m/V_{\max} + x/V_{\max}$, fitted (unweighted) through the points with $x_i, y_i > 0$; if the slope $\beta_1 > 0$, then $V_{\max} = 1/\beta_1$, $K_m = \beta_0/\beta_1$; (ii) $V_{\max} = 1.2 \max y$ and K_m is the smallest x with $y \geq V_{\max}/2$ (at least 1/50 of the x range). The linearisation serves only as a start; the reported values come from the nonlinear fit [19].

4.2.5. Sigmoid (logistic)

$$f = y_0 + \frac{A}{1 + e^{-k(x-x_0)}}. \text{ With } E = e^{-k(x-x_0)}, D = 1 + E, G = AE/D^2:$$

$$\frac{\partial f}{\partial A} = \frac{1}{D}, \quad \frac{\partial f}{\partial k} = G(x - x_0), \quad \frac{\partial f}{\partial x_0} = -Gk, \quad \frac{\partial f}{\partial y_0} = 1.$$

The curve rises (or falls) from y_0 to $y_0 + A$ and is steepest at x_0 , where it passes through $y_0 + A/2$; the 25–75 % transition has width $2 \ln 3/k$.

Starts: the data are sorted by x ; the direction $d = \pm 1$ is the sign of the sample covariance of x and y . Let $L(q)$ be the first x at which the piecewise-linear data cross the level $y_{\min} + q(y_{\max} - y_{\min})$. With $x_0 = L(0.5)$ and $k_0 = d \cdot 2 \ln 3 / \max(|L(0.75) - L(0.25)|, \Delta x/20)$, the starts are $(A, k, x_0, y_0) = (y_{\max} - y_{\min}, \kappa k_0, x_0, y_{\min})$ for $\kappa \in \{1, 0.4, 2.5\}$ and one start with $k = 8d/\Delta x$ at the centre of the x range (Δx is the range of x).

Normalisation: the parameter sets (A, k, x_0, y_0) and $(-A, -k, x_0, y_0 + A)$ describe the same curve. The unrestricted fit is reported with $k > 0$, so that y_0 is always the left asymptote ($x \rightarrow -\infty$) and $y_0 + A$ the right one. With parameter limits no normalisation is applied.

4.3 Peak models

4.3.1. Gaussian peaks

For N peaks, $\beta = (A_1, \mu_1, \sigma_1, \dots, A_N, \mu_N, \sigma_N, c)$, and with $E_k = \exp(-(x - \mu_k)^2/(2\sigma_k^2))$

$$\frac{\partial f}{\partial A_k} = E_k, \quad \frac{\partial f}{\partial \mu_k} = A_k E_k \frac{x - \mu_k}{\sigma_k^2}, \quad \frac{\partial f}{\partial \sigma_k} = A_k E_k \frac{(x - \mu_k)^2}{\sigma_k^3}, \quad \frac{\partial f}{\partial c} = 1.$$

A_k is the height of peak k above the constant background c (negative for a dip), μ_k its position and σ_k its standard width. The full width at half maximum is

$$\text{FWHM}_k = 2\sqrt{2 \ln 2} \sigma_k \approx 2.35482 \sigma_k, \quad u(\text{FWHM}_k) = 2\sqrt{2 \ln 2} u(\sigma_k). \quad (4.4)$$

Starting values come from a greedy peak search on the data sorted by x . The y values are smoothed with the kernel $(1, 2, 1)/4$ (for $n \geq 5$) and multiplied by a polarity $\pi = \pm 1$ (peaks or dips). The baseline c_0 is the 10th percentile of the smoothed values. Then, N times: the largest remaining value A at position μ is found, the width is taken from the points where the residual falls below $A/2$ ($\sigma = \text{FWHM}/2.3548$), and the Gaussian is subtracted, so that overlapping peaks and shoulders are found one after another. Both polarities are tried. After each LM run $\sigma_k \leftarrow |\sigma_k|$, and the reported peaks are sorted by position.

4.3.2. Lorentzian peak

$f = A \frac{\gamma^2}{(x - x_0)^2 + \gamma^2} + c$. With $d = x - x_0$, $Q = d^2 + \gamma^2$:

$$\frac{\partial f}{\partial A} = \frac{\gamma^2}{Q}, \quad \frac{\partial f}{\partial x_0} = \frac{2Ad\gamma^2}{Q^2}, \quad \frac{\partial f}{\partial \gamma} = \frac{2A\gamma d^2}{Q^2}, \quad \frac{\partial f}{\partial c} = 1.$$

A is the peak height above c , x_0 the position and γ the half width at half maximum; $\text{FWHM} = 2|\gamma|$ with $u(\text{FWHM}) = 2u(\gamma)$. The starts reuse the Gaussian peak search (both polarities) with $\gamma = \sqrt{2 \ln 2} \sigma \approx 1.1774 \sigma$ and $\gamma = 0.6 \sigma$. The result is reported with $\gamma > 0$.

4.3.3. Sloped (linear) background

Spectra often sit on a tilted baseline (detector response, scattering, the tail of a neighbouring band). With the option **Background: Linear**, the constant c of all peak models is replaced by a straight line,

$$f(x; \beta) = P(x; \beta_{\text{peaks}}) + c + mx, \quad \frac{\partial f}{\partial c} = 1, \quad \frac{\partial f}{\partial m} = x, \quad (4.5)$$

where P is the sum of the Gaussians or the Lorentzian above; all other derivatives are unchanged. The line is fitted *together* with the peaks in one least-squares problem, not subtracted beforehand: the covariance matrix then contains the correlation between the background and the peak parameters, so that $u(A_k)$, $u(\mu_k)$ and $u(\sigma_k)$ correctly include the uncertainty of the background. (Subtracting a separately estimated line first and fitting the peaks afterwards would treat the baseline as exact and underestimate the uncertainties.)

Starting values. The data are sorted by x ; with $k = \max(2, \text{round}(0.12n))$, $(\bar{x}_1, \bar{y}_1)$ and $(\bar{x}_2, \bar{y}_2)$ are the medians of the first and of the last k points. The initial line $B_0(x) = c_0 + m_0x$ passes through these two points. The Gaussian peak search above is run on $y - B_0(x)$ (both polarities), and in addition the starts of the constant-background model with $m = 0$ are tried; the best LM result is kept.

Display. The fitted background $\hat{c} + \hat{m}x$ is drawn as a dashed line under the fit, in the chart and in the export. **Subtract background** creates a new dataset with

$$y'_i = y_i - (\hat{c} + \hat{m}x_i), \quad \sigma'_i = \sigma_i, \quad (4.6)$$

(percentage uncertainties are first converted to absolute ones); the original data are kept. The subtracted line is treated as exact in the new dataset — for the final uncertainties of the peak parameters use the joint fit above.

Note. If the data lie far from $x = 0$ (e.g. wavelengths 480–560 nm), $\hat{c}$ and $\hat{m}$ are strongly anticorrelated, and the fit check reports it. This is a property of the parametrisation, not a problem of the fit: the line itself and the peak parameters are well determined.

4.4 Fourier series

The model with H harmonics and angular frequency ω is

$$f(x) = a_0 + \sum_{k=1}^H (a_k \cos k\omega x + b_k \sin k\omega x), \quad T = 2\pi/\omega. \quad (4.7)$$

Known period. If the period T is set by the user, the model is linear in $(a_0, a_1, b_1, \dots, a_H, b_H)$ and is solved by QR. The period is then exact ($u(T) = 0$) and $p = 2H + 1$.

Unknown period. Otherwise ω is a fitted parameter ($p = 2H + 2$) and the fit proceeds in three stages.

1. *Period scan.* With $\Delta x = \max x - \min x$ and $M = \max(10, \min(60, \lfloor n/(2H) \rfloor))$, the candidate frequencies are

$$\omega_q = \frac{2\pi}{\Delta x} \cdot \frac{q}{4}, \quad q = 2, 3, \dots, 4M, \quad (4.8)$$

i.e. from half a period to M periods across the data, in steps of a quarter period. (A coarser grid of whole periods could miss the true frequency and lock onto a subharmonic.) For each ω_q the linear problem is solved on a subsample of at most about 400 points, and the weighted criterion $\sum w_i r_i^2$ is recorded. The unweighted R^2 of each candidate is shown as the *periodogram*.

2. *Refinement.* The three best local minima of the criterion are used as starts; from each, 15 LM iterations are run on all data with the Jacobian

$$\frac{\partial f}{\partial a_k} = \cos k\omega x, \quad \frac{\partial f}{\partial b_k} = \sin k\omega x, \quad \frac{\partial f}{\partial \omega} = \sum_{k=1}^H kx (-a_k \sin k\omega x + b_k \cos k\omega x).$$

Solutions with $\omega \leq 0$ (mirror images) are discarded. If two solutions have practically the same S (relative difference $\leq 10^{-9}$), the one with the *higher* frequency wins: it is the fundamental, whereas the lower one is a subharmonic in which a higher harmonic imitates the first.

3. *Convergence.* The winner is iterated to convergence (up to 200 iterations).

The sines and cosines of the harmonics are generated by the angle-addition recurrence $\cos(k+1)\theta = \cos k\theta \cos \theta - \sin k\theta \sin \theta$ (and likewise for the sine); for $k \leq 8$ its error is at the level of rounding.

Number of harmonics. With **Automatic harmonics**, models with $H = 1, 2, \dots, H_{\max}$, $H_{\max} = \min(8, \lfloor (n-3)/2 \rfloor)$, are fitted and compared by the Bayesian information criterion [23],

$$\text{BIC}(H) = n \ln \frac{S_H}{n} + p_H \ln n, \quad (4.9)$$

with S_H the weighted sum of squares and $p_H = 2H + 2$ ($2H + 1$ for a known period). The search stops when BIC has failed to improve twice in a row; the model with the smallest BIC is used. For $n > 6000$, H is selected on a subsample of every $\lceil n/2500 \rceil$ -th point and the final fit uses all points.

Amplitudes and phases. Each harmonic is also reported in amplitude–phase form $a_k \cos k\omega x + b_k \sin k\omega x = R_k \cos(k\omega x - \varphi_k)$ with

$$R_k = \sqrt{a_k^2 + b_k^2}, \quad \varphi_k = \text{atan2}(b_k, a_k), \quad (4.10)$$

$$u(R_k) = \frac{\sqrt{a_k^2 u_{a_k}^2 + b_k^2 u_{b_k}^2}}{R_k}, \quad u(\varphi_k) = \frac{\sqrt{b_k^2 u_{a_k}^2 + a_k^2 u_{b_k}^2}}{R_k^2}, \quad (4.11)$$

where the covariance between a_k and b_k is neglected. The period and its uncertainty are $T = 2\pi/\omega$, $u(T) = 2\pi u(\omega)/\omega^2$.

In the app. Parameter limits, σ_x and global fits are not available for the Fourier series, because the number of its parameters changes with H .

4.5 Custom equations

A custom equation is any expression in x , for example `A*exp(-x/tau)*cos(w*x+phi)+c`. It is parsed by `math.js` [14]. Every symbol other than x , the constants `pi` and `e`, and function names becomes a fitted parameter (so `tau` is a parameter, not 2π). Supported functions include `sin`, `cos`, `tan`, their inverses and hyperbolic versions, `exp`, `log` (natural; `log(x,b)` with base b), `ln`, `log10`, `log2`, `sqrt`, `cbt`, `abs`, `sign`, `min`, `max`, `floor`, `ceil`, `round` and `atan2`. For speed the expression is translated into a plain JavaScript function; anything outside this set is evaluated by `math.js` itself. Points where the expression is not finite make S infinite, so such parameter values are never accepted.

Jacobian. TrollFit forms the symbolic derivative of the expression with respect to every parameter. It is used if it agrees with central differences at three test points (relative tolerance about 10^{-5}); otherwise, and at any point where the analytic derivative is not finite, the Jacobian is computed by central differences with step $h_j = 10^{-6}|\beta_j|$ (10^{-6} for $\beta_j = 0$). Functions without a derivative rule (`min`, `floor`, ...) therefore always use differences.

Global search for a start. Because a general expression has no natural starting values, TrollFit searches for them. Typical magnitudes are derived from the data,

$$\mathcal{M} = \left\{ 1, \Delta x, \frac{1}{\Delta x}, |x|_{\max}, \frac{1}{|x|_{\max}}, \Delta y, |y|_{\max}, |\bar{y}|, \frac{2\pi}{\Delta x}, \frac{20\pi}{\Delta x}, 0.1, 10 \right\},$$

so that parameters such as a time constant $\tau = 300$ or a frequency of 1000 can be found, not only values near one. Then:

1. $N = \max(200, 60p)$ random candidates are drawn: each parameter is a magnitude $m \in \mathcal{M}$ times a factor $e^{1.4(U-0.5)} \in [0.5, 2]$, with a negative sign with probability 0.2. The all-ones vector is always included. The pseudo-random generator (`mulberry32`) has a fixed seed, so the result is reproducible.
2. All candidates are evaluated on a subsample of about 60 points; the ten best (plus the all-ones start) receive 12 LM iterations on all data.

3. The three best continue in batches of 25 iterations (at most 16 batches); a run that is clearly worse than the best one and no longer improving is stopped early.
4. Among solutions with practically equal S (relative difference $\leq 10^{-6}$) the most *natural* one is preferred: fewest negative parameters, then smallest magnitudes (measured by $\sum_j \log_{10}(1 + |\beta_j|)$). It is iterated to full convergence (up to 1000 iterations).
5. Equivalent solutions are normalised without knowledge of the formula: a parameter is replaced by its value wrapped into $(-\pi, \pi]$, or the sign of one parameter is flipped together with a shift of another by π (since $-A \sin \theta = A \sin(\theta + \pi)$). A change is accepted only if S stays the same to within 10^{-9} relative and the parameters become more natural, so non-periodic parameters are never affected.

User start values. Start values entered for some parameters are used directly in the first candidate; the random candidates vary them by $\pm 30\%$ (every third candidate keeps them exactly), while the remaining parameters are searched as above.

Note. For well-posed models the global search is very reliable, but it cannot guarantee the global minimum of an arbitrary function. If the result looks wrong, provide start values, or fix parameters you know (chapter 5).

Fixed parameters and parameter limits

In the **Parameter limits** dialog each parameter of a model can be

- *fixed* at a known value $\beta_j = v_j$ (e.g. $b = 0$ for a line through the origin), or
- restricted to an interval $\ell_j \leq \beta_j \leq h_j$, where either bound may be absent.

The settings are stored per dataset and per model. They are available for all models except the Fourier series.

5.1 Formulation

Let F be the index set of the free parameters and $p = |F|$. Fixed parameters are removed from the parameter vector; the remaining ones solve the bound-constrained problem

$$\min_{\beta_F} S(\beta) \quad \text{subject to} \quad \ell_j \leq \beta_j \leq h_j, \quad j \in F. \quad (5.1)$$

The fit is done in the original parameterisation of the model (for example $y = ax + b$, not the centred form), using the general Levenberg–Marquardt method with the model’s analytic Jacobian.

5.2 Algorithm

A plain projected LM method — clipping each trial step to the bounds — can stall at a worse point, because the step of the other parameters is computed as if the parameter on the bound were still free to move. TrollFit therefore uses an active-set strategy [20]:

1. *Start.* The unrestricted fit (if it succeeds) is projected onto the box. If both bounds of a parameter are finite, a second start at the interval midpoints is tried as well; the better result is kept.
2. *Inner fit.* The parameters not in the active set $\mathcal{A}$ are fitted by LM; every trial point is projected onto the box, $\beta_j \leftarrow \min(h_j, \max(\ell_j, \beta_j))$.
3. *Activation.* A parameter that ends on a bound (within $10^{-10} \max(1, |\beta_j|)$) is added to $\mathcal{A}$ and held fixed.
4. *Release (KKT check).* With the gradient $\mathbf{g} = \mathbf{J}^T \mathbf{W} \mathbf{r}$ (note $\partial S / \partial \beta_j = -2g_j$), a parameter at its lower bound with $g_j > 0$, or at its upper bound with $g_j < 0$, would decrease S by moving into the interior; it is released from $\mathcal{A}$.

5. Steps 2–4 are repeated until $\mathcal{A}$ no longer changes (at most $2p + 4$ rounds).

Afterwards the fit is checked to be finite at all data points; for the rational model the pole must not lie inside the data range.

5.3 Uncertainties and degrees of freedom

The covariance matrix is computed from eq. (3.4) with the Jacobian of the free parameters only, and with $\nu = n - p$ where p counts free parameters. Fixed parameters have zero uncertainty and are marked *fixed*. A parameter that lies on a bound (within $10^{-9} \max(1, |\text{bound}|)$) is marked *at limit*; its uncertainty is computed as if it were free and should be interpreted with care — the constraint, not the data, determines its value.

Note. Fixing a parameter changes the model: it reduces the uncertainty of the remaining parameters only because additional knowledge has been put in. Fix parameters only at values that are known independently of the data being fitted.

Uncertainties in x : orthogonal distance regression

6.1 The problem

If the abscissae are uncertain as well, minimising vertical distances is no longer correct: a point with a large σ_x on a steep part of the curve is penalised for a deviation that its x -uncertainty easily explains. Orthogonal distance regression (ODR) [5, 6] lets every point move in x by an unknown shift δ_i and minimises

$$S(\beta, \delta) = \sum_{i=1}^n \left[\frac{(y_i - f(x_i + \delta_i; \beta))^2}{\sigma_{y,i}^2} + \frac{\delta_i^2}{\sigma_{x,i}^2} \right] \quad (6.1)$$

over the parameters *and* the n shifts. Points with $\sigma_{x,i} = 0$ are exact in x ($\delta_i = 0$, the second term is absent). For a straight line this is the York (Deming) regression [26]; in general it is the problem solved by ODRPACK, available in Python as `scipy.odr`. Figure 6.1 illustrates the geometry.

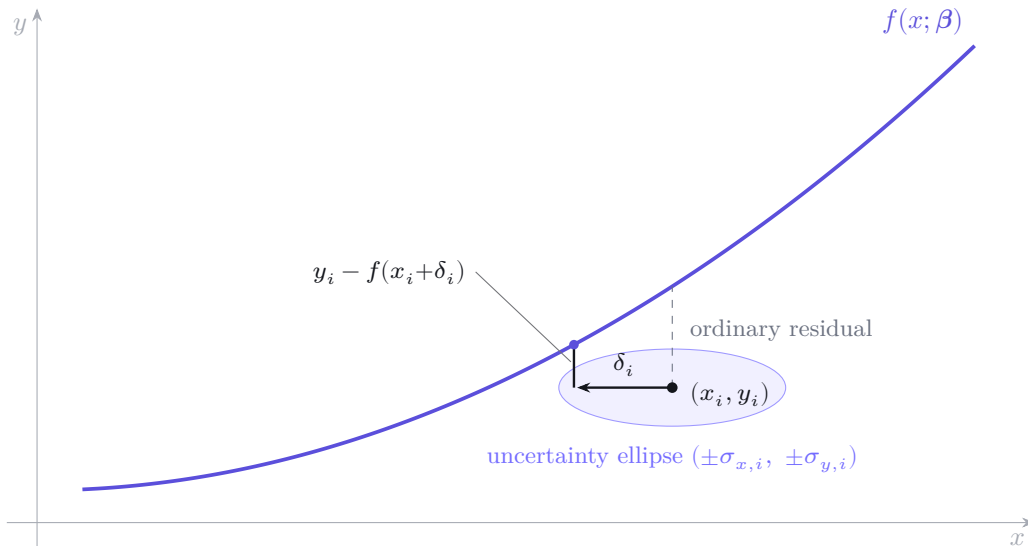


Figure 6.1.: ODR: the point may move along x by δ_i at a cost $\delta_i^2/\sigma_{x,i}^2$, in addition to the vertical misfit $(y_i - f)^2/\sigma_{y,i}^2$. The optimum balances the two according to the uncertainty ellipse of the point.

6.2 Algorithm

Problem (6.1) is a nonlinear least-squares problem in $p + n$ unknowns with residual vector $(\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)})$, $\rho_i^{(1)} = (y_i - f(x_i + \delta_i))/\sigma_{y,i}$, $\rho_i^{(2)} = \delta_i/\sigma_{x,i}$. It is solved by Levenberg–Marquardt over $(\boldsymbol{\beta}, \boldsymbol{\delta})$, exploiting the special structure of the system so that the cost per iteration is $O(np^2)$ instead of $O((n+p)^3)$.

Linearising at the current $(\boldsymbol{\beta}, \boldsymbol{\delta})$, let

$$\mathbf{G}_i = \frac{\nabla_{\boldsymbol{\beta}} f(x_i + \delta_i)}{\sigma_{y,i}}, \quad D_i = \frac{f'(x_i + \delta_i)}{\sigma_{y,i}}, \quad e_i = \frac{1}{\sigma_{x,i}},$$

where $f' = \partial f / \partial x$ is computed by a central difference with step $h = 10^{-6}(|x| + 0.01 \Delta x)$. The damped normal equations have an *arrowhead* structure: the δ -block is diagonal. Eliminating each $\Delta \delta_i$ (Schur complement) gives

$$\Delta \delta_i = \frac{D_i \rho_i^{(1)} - e_i \rho_i^{(2)} - D_i \mathbf{G}_i^T \Delta \boldsymbol{\beta}}{c_i}, \quad c_i = (D_i^2 + e_i^2)(1 + \lambda), \quad (6.2)$$

and the reduced $p \times p$ system for the parameter step

$$\left[\sum_i \kappa_i \mathbf{G}_i \mathbf{G}_i^T + \lambda \text{diag}(\mathbf{A}) \right] \Delta \boldsymbol{\beta} = \sum_i \mathbf{G}_i \left(\rho_i^{(1)} - \frac{D_i (D_i \rho_i^{(1)} - e_i \rho_i^{(2)})}{c_i} \right), \quad \kappa_i = 1 - \frac{D_i^2}{c_i}, \quad (6.3)$$

with $\mathbf{A} = \sum_i \mathbf{G}_i \mathbf{G}_i^T$. For points without x -uncertainty $\kappa_i = 1$ and the correction term vanishes. The reduced system is solved with diagonal scaling as in section 3.3; step acceptance and the update of λ are the same as there.

The iteration starts from the result of the ordinary fit ($\boldsymbol{\delta} = 0$), which respects fixed parameters and limits; limits are enforced by projection. It stops when the relative decrease of S is below 10^{-14} , or when both the largest relative parameter change is below 10^{-11} and the largest change of the shifts is below $10^{-9} \sigma_{x,i}$; or after 25 iterations with a total relative improvement below 10^{-10} ; or after 400 iterations. (The criterion on the shifts matters: starting from the ordinary fit, the first steps mainly move the δ_i while $\boldsymbol{\beta}$ barely changes.)

6.3 Uncertainties

At the solution, setting $\lambda = 0$ gives $\kappa_i = \sigma_{y,i}^2 / (\sigma_{y,i}^2 + f_i'^2 \sigma_{x,i}^2)$, so the reduced information matrix is

$$\mathbf{M} = \sum_{i=1}^n \frac{\nabla_{\boldsymbol{\beta}} f_i \nabla_{\boldsymbol{\beta}} f_i^T}{\sigma_{y,i}^2 + f_i'^2 \sigma_{x,i}^2}, \quad \mathbf{C} = s^2 \mathbf{M}^{-1}, \quad (6.4)$$

with $s^2 = 1$ for absolute uncertainties and $s^2 = S/\nu$ otherwise — the same estimate as `cov_beta` of `scipy.odr` (multiplied by `res_var` in the relative case). Each point thus has an *effective weight*

$$w_i^{\text{eff}} = \frac{1}{\sigma_{y,i}^2 + f_i'^2 \sigma_{x,i}^2}, \quad (6.5)$$

which is used wherever the subsequent statistics need weights: the prediction band, the standardised residuals and outlier test, and AICc. The reported χ^2 is the minimum of eq. (6.1).

In the app. With σ_x active, the results panel shows σ_x : *ODR fit* and the number of points with an x -uncertainty. The Fourier series and global fits ignore σ_x and say so.

Global fits of several datasets

A global fit applies one model to $K \geq 2$ datasets simultaneously. Each parameter is either *shared* — one common value for all datasets (for example a decay rate measured at several temperatures) — or *local*, fitted separately for every dataset. A shared parameter uses all the data at once, which is more precise and statistically more correct than averaging separate fits.

7.1 Formulation

Let dataset k have n_k points (x_{ki}, y_{ki}) with weights w_{ki} , and let the model have m parameters, $\mathcal{S}$ of them shared. The global parameter vector $\boldsymbol{\theta}$ contains each shared parameter once and each local parameter K times, so its length is

$$P = |\mathcal{S}| + K(m - |\mathcal{S}|). \quad (7.1)$$

With the map $\boldsymbol{\beta}^{(k)}(\boldsymbol{\theta})$ that assembles the parameters of dataset k , the fit minimises

$$S(\boldsymbol{\theta}) = \sum_{k=1}^K \sum_{i=1}^{n_k} w_{ki} (y_{ki} - f(x_{ki}; \boldsymbol{\beta}^{(k)}(\boldsymbol{\theta})))^2 \quad (7.2)$$

over the stacked data of $N = \sum_k n_k$ points by Levenberg–Marquardt (section 3.3). The Jacobian of a point of dataset k with respect to $\boldsymbol{\theta}$ is the model Jacobian placed in the columns of $\boldsymbol{\beta}^{(k)}$. The fit is weighted only if all selected datasets have valid σ_y ; otherwise all weights are one.

Starts. The datasets are first fitted separately. The first start uses the local values of these fits and, for every shared parameter, their mean; further starts take the shared values from each individual dataset in turn ($K + 1$ starts in total). The solution with the smallest S is kept.

Uncertainties. The covariance matrix is eq. (3.4) for the full problem, with $\nu = N - P$ and $s^2 = S/\nu$ (or 1 for absolute σ_y). The results of each dataset — parameters, bands, residual diagnostics — are reported with this common s^2 and ν .

7.2 Is sharing consistent with the data?

The separate fits form a larger model (Km parameters) that contains the global one as a special case, so the two can be compared by a nested-model test. With S_{sep} the sum of the separate minima, $\nu_{\text{sep}} = N - Km$ and $\Delta\nu = Km - P$:

$$F = \frac{(S_{\text{glob}} - S_{\text{sep}})/\Delta\nu}{S_{\text{sep}}/\nu_{\text{sep}}}, \quad p = 1 - F_{\Delta\nu, \nu_{\text{sep}}}(F) \quad (7.3)$$

(relative or no uncertainties), or, with absolute uncertainties,

$$\chi^2 = S_{\text{glob}} - S_{\text{sep}}, \quad p = 1 - F_{\chi^2_{\Delta\nu}}(\chi^2). \quad (7.4)$$

A small p -value means that the datasets do not support a common value of the shared parameters. The F and χ^2 distribution functions are computed from the regularised incomplete beta and gamma functions (chapter 12).

In the app. Changing the model switches the global fit off, because the set of parameters changes. σ_x is not used in global fits.

Checking a fit: residual diagnostics

After every fit TrollFit examines the residuals and summarises the result in plain words under the parameters (**Fit check**): are the residuals random, are they roughly normal, and is any point unusually far from the curve? The tests are deliberately simple and well established; they are aids to judgement, not proofs that a model is correct.

8.1 Standardised residuals

With the weights w_i (effective weights (6.5) for ODR) and s^2 from eq. (3.4), the standardised residuals are

$$z_i = \frac{\sqrt{w_i} r_i}{s}. \quad (8.1)$$

For a correct model with correct uncertainties they are approximately standard normal. The **Residual plot** shows the raw residuals $r_i = y_i - \hat{y}_i$ against x .

8.2 Outliers

A large outlier inflates s itself, so with an estimated s it can never exceed a fixed threshold in small samples ($|z_i| \leq \sqrt{\nu}$). TrollFit therefore uses *deleted* (externally studentised) residuals [8], in which the scale is estimated without the point in question:

$$t_i = \frac{\sqrt{w_i} r_i}{\sqrt{(S - w_i r_i^2)/(\nu - 1)}}. \quad (8.2)$$

(The leverage of the point is neglected.) A point is flagged as an outlier if $|t_i|$ exceeds

$$\tau = \max\left(3, t_{1-0.05/n}(\nu - 1)\right), \quad (8.3)$$

a Bonferroni-corrected bound [10] with a family-wise two-sided level of 10 %, so that large datasets do not report outliers merely by chance. With absolute uncertainties (or $\nu \leq 2$) the scale is known and the plain z_i is compared with $\max(3, z_{1-0.05/n})$. Outliers are only reported for $n \geq 6$.

In the app. Flagged points are circled in the chart. **Exclude outliers** unticks them so that the fit ignores them; they remain in the table and can be ticked again. Excluding points should always be justified by the experiment, not by the test alone.

8.3 Runs test: are the residuals random?

If the model misses a systematic trend, residuals of the same sign cluster together. With the residuals ordered by x and zero residuals omitted, let N be their number, n_+ and n_- the numbers of positive and negative ones, and R the number of runs (maximal blocks of equal sign). Under randomness [25]

$$\mu_R = \frac{2n_+n_-}{N} + 1, \quad \sigma_R^2 = \frac{(\mu_R - 1)(\mu_R - 2)}{N - 1}, \quad z = \frac{R - \mu_R + c}{\sigma_R}, \quad (8.4)$$

where $c = \mp \frac{1}{2}$ is a continuity correction towards zero, applied for $N < 50$. Too few runs indicate a systematic deviation, so the one-sided p -value is $p = \Phi(z)$. The test is performed when $N \geq 10$ and $n_+, n_- \geq 2$.

8.4 Normality of residuals: Anderson–Darling test

The standardised residuals are tested for normality with estimated mean and variance [3, 9]. With $u_{(1)} \leq \dots \leq u_{(n)}$ the sorted values of $(z_i - \bar{z})/s_z$, s_z being the sample standard deviation,

$$A^2 = -n - \frac{1}{n} \sum_{i=1}^n (2i-1) \left[\ln \Phi(u_{(i)}) + \ln(1 - \Phi(u_{(n+1-i)})) \right], \quad A^* = A^2 \left(1 + \frac{0.75}{n} + \frac{2.25}{n^2} \right), \quad (8.5)$$

and the p -value is taken from the approximation of D’Agostino and Stephens:

$$p = \begin{cases} \exp(1.2937 - 5.709A^* + 0.0186A^{*2}) & A^* \geq 0.6, \\ \exp(0.9177 - 4.279A^* - 1.38A^{*2}) & 0.34 < A^* < 0.6, \\ 1 - \exp(-8.318 + 42.796A^* - 59.938A^{*2}) & 0.2 < A^* \leq 0.34, \\ 1 - \exp(-13.436 + 101.14A^* - 223.73A^{*2}) & A^* \leq 0.2, \end{cases} \quad (8.6)$$

clipped to $[0, 1]$. The test is performed for $n \geq 8$.

Note. The residuals of a least-squares fit are not exactly independent (they satisfy p linear constraints), and the tests above ignore this, as is customary. With few points all tests have low power: *passing* them does not prove that a model is right.

Bands and derived quantities

All uncertainties in this chapter follow from the parameter covariance $\mathbf{C}$ by the first-order *delta method* [11]: if a quantity $q(\boldsymbol{\beta})$ depends on the parameters, then

$$u^2(q) = \mathbf{g}^T \mathbf{C} \mathbf{g}, \quad \mathbf{g} = \nabla_{\boldsymbol{\beta}} q(\hat{\boldsymbol{\beta}}). \quad (9.1)$$

95 % intervals are $q \pm k u(q)$ with k from eq. (3.6).

9.1 Confidence band

The value of the fitted curve at a point x has the uncertainty

$$u_f(x) = \sqrt{\mathbf{j}(x)^T \mathbf{C} \mathbf{j}(x)}, \quad \mathbf{j}(x) = \nabla_{\boldsymbol{\beta}} f(x; \hat{\boldsymbol{\beta}}), \quad (9.2)$$

and the 95 % confidence band is $f(x; \hat{\boldsymbol{\beta}}) \pm k u_f(x)$. It shows where the *true curve* lies, given the scatter of the data; it narrows as more data are added.

9.2 Prediction band

A *new* measurement at x scatters around the true curve by its own uncertainty $\sigma(x)$, so the 95 % prediction band is

$$f(x; \hat{\boldsymbol{\beta}}) \pm k \sqrt{u_f^2(x) + \sigma^2(x)}. \quad (9.3)$$

For the data points $\sigma_i^2 = s^2/w_i$, i.e. $\sigma_i = s$ for an unweighted fit, σ_i itself for absolute uncertainties and $s \sigma_i$ for relative ones (effective weights for ODR). Between data points $\sigma(x)$ is interpolated linearly in x ; outside the data range it is held constant at the value of the nearest end point. Figure 9.1 compares the two bands.

9.3 Evaluating the fit and inverse prediction

The **Evaluate fit** tool works in two directions.

$\mathbf{x} \rightarrow \mathbf{y}$. For given x the result is $y = f(x; \hat{\boldsymbol{\beta}})$ with $u(y) = u_f(x)$ and the 95 % interval $y \pm k u(y)$. Values outside the range of the data are marked as extrapolation.

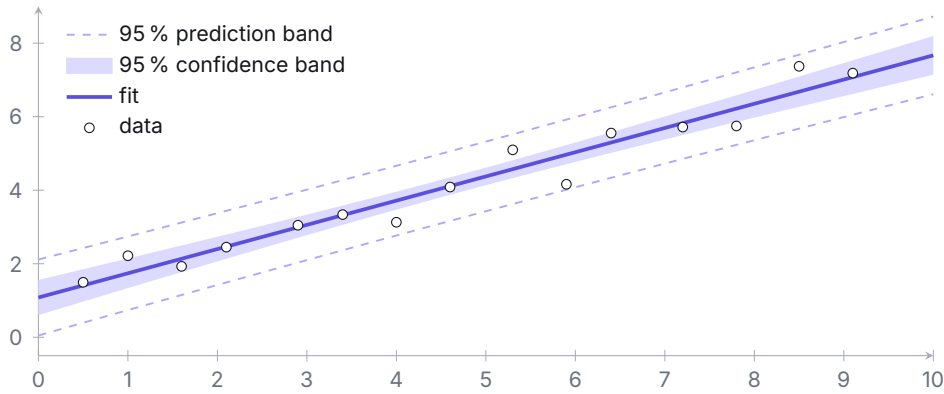


Figure 9.1.: Unweighted straight-line fit to 15 points. About 95 % of new measurements fall inside the prediction band; the confidence band contains the true line with 95 % confidence.

$y \rightarrow x$ (calibration). For a measured value y_0 with optional standard uncertainty $u(y_0)$, TrollFit finds *all* solutions of $f(x; \hat{\beta}) = y_0$ in the interval $[x_{\min} - 0.25\Delta x, x_{\max} + 0.25\Delta x]$. The interval is divided into 800 subintervals; every sign change of $f - y_0$ is refined by bisection (at most 80 halvings, tolerance $10^{-12} \times$ interval length), and roots closer than 10^{-7} of the interval are merged. For each root the uncertainty follows from linearising the curve,

$$u(x) = \frac{\sqrt{u_f^2(x) + u^2(y_0)}}{|f'(x)|}, \quad (9.4)$$

with f' from a central difference with step $10^{-6}\Delta x$. A root at which the curve is flat ($f' = 0$) has no finite uncertainty and is marked as such.

9.4 Integral

The **Integral** tool computes $I = \int_a^b f(x; \hat{\beta}) dx$ by the composite Simpson rule with 400 subintervals (if $a > b$ the sign is reversed; non-finite values of f are treated as zero). Its uncertainty follows from eq. (9.1) with

$$\mathbf{g} = \int_a^b \nabla_{\beta} f(x; \hat{\beta}) dx, \quad (9.5)$$

integrated by the same rule. (Integrating the upper and lower edges of the confidence band instead would overestimate the uncertainty systematically.)

9.5 Derivative

The **Derivative** tool shows the slope at x_0 as a tangent line. The slope is

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0 - h)}{2h}, \quad h = \max(10^{-4}\Delta x, 10^{-6}), \quad (9.6)$$

and its uncertainty follows from eq. (9.1) with $\mathbf{g} = \partial \mathbf{j} / \partial x$ at x_0 , computed by the same central difference. (Differentiating the edges of the confidence band would give the derivative of the uncertainty, not the uncertainty of the derivative — for a straight line at the centroid of the data it would wrongly give zero.)

9.6 Combining fitted functions

Two fitted curves can be added, subtracted or multiplied pointwise. The combined curve can be drawn, integrated and evaluated, but since the two fits are statistically independent and no joint covariance is formed, *no uncertainty* is reported for combined curves.

Comparing models

Compare models fits every built-in model (and the custom equation, if one is defined) to the active dataset, with the same weights, σ_x and parameter limits as a normal fit, and ranks them by the corrected Akaike information criterion AICc [1, 13, 7]. A better fit lowers AICc, every additional parameter raises it, so the criterion balances accuracy against complexity.

10.1 Definition

Let $S = \sum_i w_i r_i^2$ (effective weights for ODR) and let K be the number of estimated quantities. With absolute uncertainties the likelihood is known up to a constant, $-2 \ln L = \chi^2 + \text{const}$, and

$$\text{AIC} = S + 2K, \quad K = p. \quad (10.1)$$

Otherwise the variance of the errors is estimated as well, and

$$\text{AIC} = n \ln \frac{S}{n} + 2K, \quad K = p + 1. \quad (10.2)$$

The small-sample correction is

$$\text{AICc} = \text{AIC} + \frac{2K(K+1)}{n-K-1} \quad (10.3)$$

(infinite if $n \leq K + 1$, i.e. the model cannot be assessed on so few points). Here p counts free parameters only.

10.2 Interpretation

Only differences $\Delta_m = \text{AICc}_m - \min_j \text{AICc}_j$ matter. The Akaike weights

$$w_m = \frac{\exp(-\Delta_m/2)}{\sum_j \exp(-\Delta_j/2)} \quad (10.4)$$

estimate the relative support of the models by the data. As a rule of thumb, models with $\Delta_m < 2$ are about equally supported; $\Delta_m > 10$ means essentially no support [7]. A model that fits only slightly better but needs more parameters is not automatically preferred. AICc compares models fitted to the *same* data with the same weights; it says nothing about whether any of them is physically correct.

In the app. Models are listed from best to worst. Each row shows a thumbnail of the fit, the number of parameters, R^2 , χ^2/ν (for weighted fits), ΔAICc and the Akaike weight. The general polynomial is included with the degree currently set for the dataset. Models that cannot be fitted to the data (for example the logarithmic model with $x \leq 0$) are listed with the reason. A click on **Use** selects the model for the dataset.

Digitizing graphs

The digitizer reads data from an image of a graph — a screenshot, a figure from a paper, or a scan of an old printed plot. It is designed for the hardest case, old black-and-white scans with noise, blur, JPEG artefacts, slight rotation, dotted grids and text. All processing is local and works on the pixel array of the image.

11.1 Axis calibration

Four reference points are clicked: two ticks on the x axis, X_1 and X_2 , and two on the y axis, Y_1 and Y_2 , and their values are typed in. For a logarithmic axis the values must be positive and the calibration is done in $\log_{10}$ of the values.

The mapping from pixel coordinates (u, v) to data is a general *affine* map, so that rotated or skewed scans are handled correctly:

$$X(u, v) = c_1 u + c_2 v + c_3, \quad Y(u, v) = d_1 u + d_2 v + d_3, \quad (11.1)$$

where X and Y are the data values (or their $\log_{10}$). The three coefficients of X solve

$$X(X_1) = x_1, \quad X(X_2) = x_2, \quad c_1(u_{Y_2} - u_{Y_1}) + c_2(v_{Y_2} - v_{Y_1}) = 0, \quad (11.2)$$

the last condition expressing that the y axis is a line of constant x ; the coefficients of Y are obtained analogously with the roles of the axes exchanged. Each 3×3 system is solved directly; it is singular when the reference points of an axis coincide or all four points are collinear, and TrollFit then asks for different points.

Resolution and rounding. The data size of one pixel at a point is estimated from the map, $r_x = |X(u+1, v) - X(u, v)| + |X(u, v+1) - X(u, v)|$ (and similarly r_y , after undoing the logarithm). Each digitized value is rounded to the power of ten nearest to a quarter of a pixel, $10^{\text{round}(\log_{10}(r/4))}$: more digits would pretend a precision the image does not have.

Plot area. The intersection O of the two axes defines the side of each axis on which the data lie. Automatic tools search only inside this region (with a small margin), so axis lines, tick marks and tick labels are ignored. If an axis lies inside the plot (reference points on both sides of O), no restriction is applied in that direction.

11.2 Ink detection

The image is converted to luminance $L = 0.299R + 0.587G + 0.114B$ (transparent pixels count as white) and smoothed by a separable 3×3 box filter, which suppresses scan noise and JPEG artefacts. A global threshold T is chosen by Otsu's method [21], which maximises the between-class variance of the luminance histogram. When the user clicks, the darkest pixel within 3–4 px of the click decides what counts as “ink”:

- for a dark pixel (luminance $\leq T$) the ink is all pixels with $L \leq T$, weighted by $(T + 1 - L)/(T + 1)$ in centroid computations;
- for a saturated colour (max–min of the RGB channels above 60) the ink is all pixels within a Euclidean RGB distance of 85 of the clicked colour, so a red curve can be separated from a black grid.

11.3 Snapping a click

With snapping on, a click is moved to the centre of the nearest object within a window of ± 18 px:

1. The nearest ink pixel within 7 px is located.
2. *Hollow markers* ($\circ \square \diamond$) are filled: enclosed background regions of at most 22×22 px that fill at least 40 % of their bounding box and are not too elongated are treated as ink.
3. If the connected ink component is isolated (does not touch the window border) it is a *marker*, and the result is its centroid. For asymmetric markers ($\triangle \nabla$), whose centroid is displaced vertically from the centre used by plotting programs, the vertical centre of the bounding box of the marker's core (pixels with at least three ink neighbours) is used instead.
4. If the component continues beyond the window (a marker on a line, or a line), a chamfer (3–4) distance transform is computed and followed uphill to the thickest point. A local maximum of the distance of at least 2.4 px indicates a marker on a line: the result is the centroid of the ink inside the inscribed circle (with the same correction for asymmetric shapes). Otherwise the click is on a line, and the result is the centroid of the ink in a small circle around the nearest ink pixel, iterated twice — the centre of the line's thickness.

11.4 Finding similar markers

Similar finds all markers of the shape that was clicked. The clicked marker is snapped first; its neighbourhood within a disk of radius $r + 1.2$ px (r = half the marker size) forms a template

τ . The disk, rather than a square, keeps lines passing through the marker from influencing the match. Candidate positions are scored by the normalised cross-correlation [17]

$$\rho(u, v) = \frac{\sum_k (I_{(u,v)+k} - \bar{I}_{(u,v)}) (\tau_k - \bar{\tau})}{\sqrt{\sum_k (I_{(u,v)+k} - \bar{I}_{(u,v)})^2 \sum_k (\tau_k - \bar{\tau})^2}} \quad (11.3)$$

over the disk offsets k , where I is the ink-weight image. To make the search fast, an integral image of the ink mask rejects every position whose amount of ink differs too much from the template (less than $0.5\times$ or more than $2.6\times$). Positions with ρ at least the chosen similarity (70% by default) are kept, non-maximum suppression removes duplicates closer than $1.5r$, and each match is refined by snapping (section 11.3).

11.5 Tracing a curve

Trace follows a line from a clicked point in both directions. The tracker advances column by column along the direction of the calibrated x axis (so it also works on rotated scans), and in each column searches across the line for runs of ink:

- The position of the line is *predicted* by a quadratic least-squares fit to the last (up to 30) accepted points, with the curvature limited to $|c| \leq 0.02 \text{ px}^{-1}$; the initial slope comes from the principal direction of the ink around the click.
- The line thickness is the median length of the ink run across the line in 17 columns around the click.
- In each column the run whose centre is closest to the prediction is accepted, provided it is not longer than about 1.8 times the expected thickness (which rejects vertical grid lines and markers) and lies within a tolerance of the prediction. If two pieces of ink lie near the prediction (the edges of a hollow marker, a crossing grid line, text), the column is ambiguous and skipped.
- Gaps (dashed lines, crossings) are bridged along the prediction, up to $\max(30, 10 \times \text{thickness})$ px.

Finally the trace is cleaned in up to two passes: every point is compared with a local quadratic fit to its neighbours within ± 15 px, and points deviating by more than $\max(1 \text{ px}, 3.5 \times 1.4826 \times \text{median}|\text{deviation}|)$ — a robust 3.5-sigma rule based on the median absolute deviation — are removed.

The user then chooses the number of points m . If the traced curve is monotonic in x , the points are spaced evenly in the data value of x ; otherwise evenly along the arc length of the trace.

11.6 Accuracy

The digitizer was tested on synthetic graphs with known data that were deliberately “aged” by blur, noise, JPEG compression, rotation, dotted grids and legends. Snapped markers were found to within about 0.2% of the axis range and traced curves to within 0.6%. The accuracy

of the original plot (line width, printing, paper distortion) is of course a limit no digitizer can overcome; digitized data should be treated as having an uncertainty of at least the resolution of the image.

Numerical methods and validation

12.1 Special functions

TrollFit does not depend on external numerical libraries for statistics; the functions it needs are implemented as follows.

Function	Method
$\ln \Gamma(z)$	Lanczos approximation with $g = 7, 9$ coefficients [15]; reflection for $z < \frac{1}{2}$
$I_x(a, b)$, regularised incomplete beta	continued fraction (modified Lentz), relative accuracy 10^{-15} [22]
Student t distribution function $t_q(\nu)$, Student quantile	$F_\nu(t) = 1 - \frac{1}{2} I_{\nu/(\nu+t^2)}(\nu/2, \frac{1}{2})$ for $t \geq 0$ bisection on F_ν to relative accuracy 10^{-12} ; normal quantile for $\nu > 10^6$
$\Phi(z)$, normal distribution function	complementary error function after [22], relative error $< 1.2 \cdot 10^{-7}$
$\Phi^{-1}(q)$, normal quantile	rational approximation of Acklam [2], relative error $< 1.2 \cdot 10^{-9}$
$P(a, x)$, regularised incomplete gamma χ^2 and F survival functions	series for $x < a + 1$, continued fraction otherwise [22] $1 - P(k/2, x/2)$ and $1 - I_{d_1 f/(d_1 f + d_2)}(d_1/2, d_2/2)$

12.2 Linear algebra

The matrices in TrollFit are small (at most a few dozen parameters), so dense direct methods are used: Householder QR for linear least squares (section 3.2) and Gauss–Jordan elimination with partial pivoting for the LM systems and for covariance matrices. Every matrix is scaled to a unit diagonal before it is factorised and every singularity threshold is relative (10^{-12} for QR, 10^{-13} for covariance inversion, 10^{-15} for LM steps), in accordance with the principle of scale invariance.

12.3 Reproducibility

For a given input and version, every result is deterministic: there is no dependence on timing, on the order of browser events, or on the machine. The only random numbers (the start search for custom equations) come from the mulberry32 generator with the fixed seed 20260101.

12.4 Validation

The numerical core is verified by an automated test suite that is run after every change of the program. Reference values are produced independently by established Python software and compared with TrollFit’s results.

Area	Reference
All models: parameters and standard errors, unweighted, absolute and relative σ_y ; data in SI units (10^{-9}), years ($x \approx 2000$), tiny y	<code>scipy.optimize.curve_fit</code>
Weighted Fourier series with free frequency	<code>curve_fit</code>
Fixed parameters and bounds	<code>curve_fit</code> (bounds, method <code>trf</code>)
Errors in x (ODR), incl. mixed exact/uncertain x and fixed parameters	<code>scipy.odr</code> (ODRPACK)
Global fits and the sharing test	<code>curve_fit</code> on stacked data; F / χ^2 tests
Anderson–Darling statistic, runs test	<code>statsmodels</code> (<code>normal_ad</code> , <code>runstest_1samp</code>)
Student quantiles, normal distribution function	<code>scipy.stats</code>

Fitted parameters agree with the references to a relative tolerance of 10^{-6} (10^{-4} for multi-peak, Fourier and custom models, whose references are themselves iterative). Standard errors agree to $2 \cdot 10^{-3}$ relative ($2 \cdot 10^{-2}$ for custom equations); the remaining differences come from the finite-difference Jacobians used by the reference software, whereas TrollFit uses analytic derivatives where possible. In version 0.5.0 the numerical suite contains 513 checks, and a further 1645 automated checks test the user interface and the processing pipeline, including the digitizer on synthetic aged scans. All of them pass.

12.5 Limitations

- Uncertainties are first-order (linearised) estimates. For strongly nonlinear models with few points, the true confidence regions can be asymmetric; the reported $\pm u$ is then an approximation.
- All tests and intervals assume independent, normally distributed errors. Correlated errors (for example consecutive readings of a drifting instrument) are not modelled.
- A global minimum cannot be guaranteed for arbitrary custom equations (section 4.5).
- Numbers are IEEE double precision: about 15–16 significant digits.

How to cite TrollFit

If TrollFit contributed to your work — a fit in a paper, a thesis, a report or a poster — please cite it. This helps readers reproduce your analysis and helps the project continue. Citation is appreciated but never required, and the results you obtain with TrollFit are yours to use freely.

A.1 Citing the software

Please always state the *version* you used (it is shown next to the name in the app header).

Recommended citation

Svoboda, Š. (2026). *TrollFit: data analysis and curve fitting* (Version 0.5.0) [Computer software]. Troll Softworks. <https://trollsoftworks.com>

```
@software{trollfit,
  author    = {Svoboda, {\v S}imon},
  title     = {TrollFit: Data Analysis and Curve Fitting},
  version   = {0.5.0},
  year      = {2026},
  publisher = {Troll Softworks},
  url       = {https://trollsoftworks.com}
}
```

With classic BibTeX styles that do not know `@software`, use `@misc` with `howpublished = {Computer software, version 0.5.0, \url{https://trollsoftworks.com}}`.

A.2 Citing this manual

When you refer to a specific method (for example the ODR fit or the outlier test), you may also cite this manual, which documents it:

Manual

Svoboda, Š. (2026). *TrollFit Reference Manual: methods, models and numerics*, version 0.5.0. Troll Softworks. <https://trollsoftworks.com/trollfit-manual.pdf>

```
@manual{trollfit-manual,  
  author      = {Svoboda, {\v S}imon},  
  title       = {TrollFit Reference Manual: Methods, Models and Numerics},  
  edition     = {Version 0.5.0},  
  organization = {Troll Softworks},  
  year        = {2026},  
  url         = {https://trollsoftworks.com/trollfit-manual.pdf}  
}
```

A.3 Describing the analysis

A good methods section lets readers repeat the analysis. Besides the citation, state the model, how the uncertainties were treated and what the reported uncertainties mean. For example:

The data were fitted with the model $y = a e^{bx}$ by weighted nonlinear least squares (Levenberg–Marquardt) in TrollFit 0.5.0 [ref], with weights $1/\sigma_i^2$ from the measurement uncertainties taken as absolute. Uncertainties of the parameters are standard uncertainties obtained from the covariance matrix of the fit; $\chi^2/\nu = 1.07$ ($\nu = 18$).

Where relevant, also mention orthogonal distance regression for uncertainties in x (chapter 6), fixed parameters or limits (chapter 5), a global fit with the shared parameters (chapter 7), and points excluded from the fit with the reason.

Bibliography

- [1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19(6):716–723, 1974.
- [2] P. J. Acklam. An algorithm for computing the inverse normal cumulative distribution function. Technical note, 2003.
- [3] T. W. Anderson, D. A. Darling. A test of goodness of fit. *Journal of the American Statistical Association* 49(268):765–769, 1954.
- [4] P. R. Bevington, D. K. Robinson. *Data Reduction and Error Analysis for the Physical Sciences*, 3rd ed. McGraw-Hill, 2003.
- [5] P. T. Boggs, J. E. Rogers. Orthogonal distance regression. *Contemporary Mathematics* 112:183–194, 1990.
- [6] P. T. Boggs, R. H. Byrd, J. E. Rogers, R. B. Schnabel. *User’s Reference Guide for ODRPACK Version 2.01: Software for Weighted Orthogonal Distance Regression*. NISTIR 4834, National Institute of Standards and Technology, 1992.
- [7] K. P. Burnham, D. R. Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*, 2nd ed. Springer, 2002.
- [8] R. D. Cook, S. Weisberg. *Residuals and Influence in Regression*. Chapman & Hall, 1982.
- [9] R. B. D’Agostino, M. A. Stephens (eds.). *Goodness-of-Fit Techniques*. Marcel Dekker, 1986.
- [10] O. J. Dunn. Multiple comparisons among means. *Journal of the American Statistical Association* 56(293):52–64, 1961.
- [11] JCGM 100:2008. *Evaluation of measurement data — Guide to the expression of uncertainty in measurement* (GUM). Joint Committee for Guides in Metrology, 2008.
- [12] C. S. Hanes. Studies on plant amylases. I. The effect of starch concentration upon the velocity of hydrolysis by the amylase of germinated barley. *Biochemical Journal* 26(5):1406–1421, 1932.
- [13] C. M. Hurvich, C.-L. Tsai. Regression and time series model selection in small samples. *Biometrika* 76(2):297–307, 1989.
- [14] J. de Jong et al. *math.js: an extensive math library for JavaScript and Node.js*. <https://mathjs.org>.
- [15] C. Lanczos. A precision approximation of the gamma function. *SIAM Journal on Numerical Analysis, Series B* 1:86–96, 1964.
- [16] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics* 2(2):164–168, 1944.
- [17] J. P. Lewis. Fast normalized cross-correlation. In *Vision Interface*, pp. 120–123, 1995.
- [18] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics* 11(2):431–441, 1963.
- [19] L. Michaelis, M. L. Menten. Die Kinetik der Invertinwirkung. *Biochemische Zeitschrift* 49:333–369, 1913.

- [20] J. Nocedal, S. J. Wright. *Numerical Optimization*, 2nd ed. Springer, 2006.
- [21] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9(1):62–66, 1979.
- [22] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992.
- [23] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics* 6(2):461–464, 1978.
- [24] G. A. F. Seber, C. J. Wild. *Nonlinear Regression*. Wiley, 1989.
- [25] A. Wald, J. Wolfowitz. On a test whether two samples are from the same population. *The Annals of Mathematical Statistics* 11(2):147–162, 1940.
- [26] D. York, N. M. Evensen, M. López Martínez, J. De Basabe Delgado. Unified equations for the slope, intercept, and standard errors of the best straight line. *American Journal of Physics* 72(3):367–375, 2004.