A weighted total least-squares algorithm for any fitting model with correlated variables

2013 Metrologia 50 654

(http://iopscience.iop.org/0026-1394/50/6/654)

Metrologia **50** (2013) 654–662 doi:10.1088/0026-1394/50/6/654

A weighted total least-squares algorithm for any fitting model with correlated variables

Andrea Malengo and Francesca Pennecchi

INRIM, Istituto Nazionale di Ricerca Metrologica, 10135 Torino, Italy

E-mail: f.pennecchi@inrim.it

Received 1 March 2013, in final form 18 October 2013
Published 27 November 2013
Online at stacks.iop.org/Met/50/654

Abstract

An algorithm able to deal with any desired fitting model was developed for regression problems with uncertain and correlated variables.

A typical application concerns the determination of calibration curves, especially (i) in those cases in which the uncertainties on the independent variables x_i cannot be considered negligible with respect to those associated with the dependent variables y_i, and (ii) when correlations exist among x_i and y_i. In the metrological field, several types of software have already been dedicated to the determination of calibration curves, some being focused just on problem (i) and a few others considering also problem (ii) but only for a straight-line fitting model. The proposed algorithm is able to deal with problems (i) and (ii) at the same time, for a generic fitting model. The tool was developed in the MATLAB® environment and validated on several benchmark data sets, fitted with linear and non-linear regression models.

A review of the most commonly applied approximations to the parameter uncertainty is also presented, together with a Monte Carlo method proposed for comparison purposes with the results provided by the formula for the uncertainty evaluation which is implemented in the software.

1. Introduction

The determination of a curve fitting a set of experimental data is a frequent problem in metrology, especially in relation to calibration procedures. The most widely known tools for such a problem are ordinary (OLS) and weighted (WLS) least-squares methods. In principle, when uncertainties appear on both the dependent and the independent variables, it is necessary to resort to more complex statistical techniques, such as the weighted total least-squares (WTLS) method [1]. The WTLS formulation addresses the case of heteroscedastic and/or correlated data. In the particular case of equal uncertainties and uncorrelated errors in the variables, the problem reduces to the so-called total least-squares (TLS) problem [2], also named as orthogonal regression, which has a unique solution given in terms of the singular value decomposition of the data matrix. In general, however, the WTLS problem does not have a closed-form solution and some iterative optimization algorithm needs to be applied.

Within the metrological area, many books and papers [3–10], reports [11] and standards [12–14] have been published on the topic, and several types of relevant software have been developed [15–18]. Most of the cited works only consider uncorrelated estimates x_i of the stimuli and uncorrelated data y_i. This is a strong assumption which may not apply to several measurement contexts, for example when the standards used for the calibration are traceable to a common reference standard or when the instrument responses need to be corrected for the estimate of a common quantity. Actually, it is important to take into account such correlations, both for estimating the fitting parameters and evaluating the associated uncertainty.

A general formulation of WTLS, taking into account the full covariance matrix of the independent variables and that of the dependent variables, is available in [6, 9] from NPL. However, the NPL software XLGENLINE [16] can actually work only with diagonal covariance matrices (see [6], section 5). A recently published standard [13] and the relevant software [18] address the case of correlation among x_i and among y_i, but considering only a straight-line model. The B-LEAST software [15] is able to work with different kinds of

regression models, including power and exponential models, but it requires diagonal covariance matrices only.

Within this context, we developed a MATLAB®-based tool for the WTLS method for calibration problems, able to work without any restriction on the form of the regression model nor on the structure of the covariance matrices, which generalizes the area of applicability of the above-mentioned software. Our algorithm is based on a MATLAB® minimization routine for non-linear unconstrained optimization. Notice that a different approach to the problem could be modelling it by introducing a certain number of constraints. Such a formulation could involve the use of Lagrange multipliers, which have been applied so far to several kinds of metrological problems [19, 20].

In section 2, we establish the minimization problem and give some details on the implemented software. Section 3 is dedicated to several methods for the evaluation of the uncertainty associated with the parameter estimates. Section 4 describes the software validation conducted on several benchmark data sets selected from the literature. In section 5, the software is applied to an example concerning the calibration of a gas chromatograph (GC).

2. Regression

2.1. The WTLS model

In order to fit n experimental data pairs (x_i, y_i), for $i = 1, \ldots, n$, whose components have uncertainties u_{x_i} and u_{y_i}, respectively, let us consider the following fitting model:

$$y = f(x, \boldsymbol{p}), \tag{1}$$

where $\boldsymbol{p} = [p_1, \ldots, p_k]$ is the vector of parameters to be estimated.

The function to be minimized is

$$\chi^2 = \sum_{i=1}^{n} \left[\frac{(x_i - X_i)^2}{u_{x_i}^2} + \frac{(y_i - f(X_i, \boldsymbol{p}))^2}{u_{y_i}^2} \right], \tag{2}$$

where X_i are n parameters to be adjusted jointly with $\boldsymbol{p}$. Parameters X_i do not appear in the framework of OLS nor in that of WLS, since in those models data x_i are not affected by uncertainties. Sum (2) is a function of $\boldsymbol{X} = [X_1, \ldots, X_n]$ and $\boldsymbol{p}$ and gets its minimum at the estimates $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{X}}$.

The above-presented approach does not take into account any possible covariances $\mathrm{cov}(x_i, x_j)$ and $\mathrm{cov}(y_i, y_j)$. A more general approach, involving also such terms, is expressed in the following matrix form:

$$\chi^2 = \boldsymbol{dx}\, \boldsymbol{U}_{\boldsymbol{x}}^{-1} \boldsymbol{dx}^{\mathrm{T}} + \boldsymbol{dy} \boldsymbol{U}_{\boldsymbol{y}}^{-1} \boldsymbol{dy}^{\mathrm{T}}, \tag{3}$$

where $\boldsymbol{dx} = [x_i - X_i, \ldots, x_n - X_n]$ and $\boldsymbol{dy} = [y_i - f(X_i, \boldsymbol{p}), \ldots, y_n - f(X_n, \boldsymbol{p})]$ are the residual vectors, and $\boldsymbol{U}_{\boldsymbol{x}}$ and $\boldsymbol{U}_{\boldsymbol{y}}$ are the covariance matrices of the experimental data. Note that, within expression (3), $\boldsymbol{U}_{\boldsymbol{x}}$ and $\boldsymbol{U}_{\boldsymbol{y}}$ can be covariance matrices of any form, not necessarily diagonal. When $\boldsymbol{U}_{\boldsymbol{x}}$ and $\boldsymbol{U}_{\boldsymbol{y}}$ are diagonal, (3) reduces to (2).

On the other hand, neither expression (2) nor (3) takes into account possible covariances $\mathrm{cov}(x_i, y_j)$; within the metrological literature, there exist some works taking them into account, but only for diagonal matrices $\boldsymbol{U}_{\boldsymbol{x}}$ and $\boldsymbol{U}_{\boldsymbol{y}}$ [5, 6]. We will not consider this case in this paper.

2.2. Minimization

In general, function (3) is non-linear in its parameters $\boldsymbol{X}$ and $\boldsymbol{p}$ and a numerical solution is necessary for its minimization. We implemented an algorithm based on the MATLAB® function `fminunc.m` (available within the MATLAB® Optimization Toolbox), which provides minimization for multidimensional and non-linear functions. This function uses descent direction methods, which are known as quasi-Newton methods, along with line-search strategies based on quadratic or cubic interpolations.

Function (3) is written in an m-file, which is passed to `fminunc.m` as an input, together with a starting (vector) point $[\boldsymbol{X}_0, \boldsymbol{p}_0]$ for the parameter estimates. Usually, but not mandatorily, $X_{0_i} = x_i$, for each i, and $\boldsymbol{p}_0$ is the vector of the OLS estimates of the parameters. See the appendix for details on how to implement function (3) as an m-file.

Among the outputs provided by `fminunc.m`, the most relevant are the estimates vector $[\hat{\boldsymbol{X}}, \hat{\boldsymbol{p}}]$, which is the optimal solution, and the value of the objective function χ^2 at the solution, that is, χ^2_{min}. Other outputs are available, such as the gradient vector and the Hessian matrix at the optimal solution, and the exitflag value, which describes the exit conditions of the algorithm.

In order to make the algorithm as efficient as possible, proper settings for the `fminunc.m` function can be applied and suitable data transformations implemented. For example, setting the 'GradObj' option of `fminunc.m` 'on' allows the application of a user-defined gradient of the objective function instead of letting `fminunc.m` estimate it by finite differences. To that aim, the m-file describing function (3) needs to be completed with the analytical expression for the gradient of (3) itself (see the appendix).

Moreover, before passing experimental data x_i and y_i to `fminunc.m`, it may be useful to standardize them into $(x_i - \bar{x})/s_x$ and $(y_i - \bar{y})/s_y$, respectively, where $\bar{x}$ and $\bar{y}$ are the sample means and s_x and s_y are the sample standard deviations of the input data sets $\boldsymbol{x}$ and $\boldsymbol{y}$. The reason for standardization is that it can avoid round-off errors typically occurring when inverting ill-conditioned matrices. Of course, also the covariance matrices of the input variables have to be scaled accordingly, that is, becoming $\boldsymbol{U}_{\boldsymbol{x}}/s_x^2$ and $\boldsymbol{U}_{\boldsymbol{y}}/s_y^2$. Note that χ^2_{min} is invariant for data standardization. After minimization, the obtained parameter estimates refer to the standardized data and need to be transformed back into the estimates of the original parameters[1]. In order to do that, the analytical expression of the original parameters $\boldsymbol{p}$ in terms of the standardized $\boldsymbol{p}_s$ is needed and it can be found from the equation $f(x, \boldsymbol{p}) = f((x - \bar{x})/s_x, \boldsymbol{p}_s)s_y + \bar{y}$ (obtained from the standardization of variables in expression (1)), by equating the coefficients of the corresponding terms in x. For the sake

[1] Accordingly, once the covariance matrix of the parameter estimates is determined (as will be described in section 3), it will also be suitably transformed.

of clarity, we describe such an approach for the straight-line model

$$y = ax + b. \tag{4}$$

Standardizing variables x and y, one gets the standardized model

$$\frac{y - \bar{y}}{s_y} = a_s \frac{x - \bar{x}}{s_x} + b_s, \tag{5}$$

where a_s and b_s are the corresponding standardized parameters. By rearranging equation (5), the following relation is obtained:

$$y = (a_s s_y / s_x)x - a_s s_y / s_x \bar{x} + b_s s_y + \bar{y}. \tag{6}$$

By equating the coefficients of x in equations (4) and (6), and the constant terms, one gets the relationships $a = a_s s_y / s_x$ and $b = -a_s s_y / s_x \bar{x} + b_s s_y + \bar{y}$, linking the original and the standardized parameters.

Analytical relations between original and standardized parameters are well known for the multiple regression model (see [21] for an example among many others). However, finding such a relationship for a general regression function is a model-dependent problem.

3. Uncertainty of parameter estimates

3.1. Approximating formulae

In general, the evaluation of the uncertainty associated with the parameter estimates of an implicit model is a difficult issue. In the considered case, the minimization problem does not have an analytical solution but needs to be solved iteratively [22]. Within metrological literature, several approximations to the uncertainty have been proposed:

(i) A first approximation [23] consists in applying to the fitting model (1) the typical WLS parameter uncertainty evaluation (see, e.g., [24]), after converting the variance of the independent variable x into an effective contribution to the variance of the dependent variable y (the so-called 'effective variance' method [25]). Actually, reducing a WTLS fitting to a WLS problem is a widespread habit, which may lead to reliable results provided that the heteroscedasticity of the dependent variable is negligible [26].

(ii) Another approximation corresponds to identifying the covariance matrix associated with the parameter estimates with $(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}$ [6, 13], where $\boldsymbol{J}$ is the Jacobian matrix of the x and y residuals, weighted by the triangular matrix of the Cholesky factorization of the covariance matrix associated with the experimental data.

(iii) A third approximation is given by $2\boldsymbol{H}^{-1}$ [4, 5], where $\boldsymbol{H}$ is the Hessian matrix of the cost function (3), i.e. the matrix of the second-order derivatives with respect to $\boldsymbol{X}$ and $\boldsymbol{p}$, calculated at the solution point $[\hat{\boldsymbol{X}}, \hat{\boldsymbol{p}}]$. In [22], it has been shown that such a formula gives an approximation to the estimate uncertainty better than that described in (ii). The two approximations coincide whenever the residuals $\boldsymbol{dy}$ are linear with respect to $\boldsymbol{X}$ and $\boldsymbol{p}$.

(iv) Some standards [12, 14], a paper [7] and the relevant software [15] refer to another approximation for the uncertainty of the parameter estimates, for which an analytical derivation can be found in [22]. The formula is given by

$$\boldsymbol{V}_{\mathrm{out}} = \boldsymbol{Q}\boldsymbol{V}_{\mathrm{in}}\boldsymbol{Q}^{\mathrm{T}}. \tag{7}$$

The matrix $\boldsymbol{V}_{\mathrm{in}}$ is the (block-diagonal) covariance matrix of the input variables x_i and y_i, that is

$$\boldsymbol{V}_{\mathrm{in}} = \begin{bmatrix} \boldsymbol{U}_x & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{U}_y \end{bmatrix}, \tag{8}$$

where $\boldsymbol{O}$ is the null matrix. The matrix $\boldsymbol{Q}$ in expression (7) is defined as

$$\boldsymbol{Q} = \boldsymbol{H}^{-1}\boldsymbol{D}_{\mathrm{in}}, \tag{9}$$

where $\boldsymbol{D}_{\mathrm{in}}$ is the matrix of the mixed second-order derivatives of function (3) with respect to parameters $[\boldsymbol{X}, \boldsymbol{p}]$ and input variables $[\boldsymbol{x}, \boldsymbol{y}]$.
Matrix $\boldsymbol{V}_{\mathrm{out}}$, given by (7), encloses the variances and the covariances of the whole set of the (output) parameter estimates $[\hat{\boldsymbol{X}}, \hat{\boldsymbol{p}}]$, from which it is easy to extract the submatrix relevant to the parameter estimates $\hat{\boldsymbol{p}}$ of the fitting model (1).

Expression (7) can be derived by applying the usual propagation of uncertainty to a first-order Taylor expansion of the cost function (3) [22] and considering that equation (3) implicitly defines parameters $[\boldsymbol{X}, \boldsymbol{p}]$ as a function of observations $[\boldsymbol{x}, \boldsymbol{y}]$. Hence, (7) is valid only when the non-linearity of the above-mentioned implicit function can be neglected.

For homoscedastic observations and for a straight-line fitting model, the problem reduces to a TLS problem, which has a solution in terms of the singular value decomposition of the data matrix [2]. In this case, since the parameter estimates result in a linear transformation of the data, formula (7) provides an exact expression for their uncertainty. For such a model, it was analytically demonstrated that both approximations (ii) and (iii) provide uncertainty values (in the sense of the norm of relevant covariance matrices) smaller than those provided by (7) [22]. Hence, at least for the considered TLS problem, it can be derived that approximations (ii) and (iii) underestimate the uncertainty. Therefore, within our software, we implemented formula (7) in order to calculate the parameter estimates' uncertainty (see the appendix).

3.2. Monte Carlo method for uncertainty evaluation

We describe here an implementation of MC simulations to evaluate the uncertainty associated with the WTLS parameter estimates. Once estimates $[\hat{\boldsymbol{X}}, \hat{\boldsymbol{p}}]$ of parameters $[\boldsymbol{X}, \boldsymbol{p}]$ are found, the corresponding adjusted points $(\hat{X}_i, \hat{Y}_i = f(\hat{X}_i, \hat{\boldsymbol{p}}))$ can be obtained, for $i = 1, \ldots, n$. Hence, M couples (x_{i_j}, y_{i_j}), for $j = 1, \ldots, M$, can be generated by means of a MC simulation. At each jth step of the simulation, n abscissae x_{i_j} (for $i = 1, \ldots, n$) are drawn from a multivariate normal distribution centred in $\hat{\boldsymbol{X}}$ and having covariance matrix equal to

$\boldsymbol{U}_x$, and n ordinates y_{ij} are drawn from a multivariate normal distribution centred in $\hat{\boldsymbol{Y}}$ and having covariance matrix $\boldsymbol{U}_y$. Hence, the WTLS method is applied to each generated data set, producing a vector estimate $\hat{\boldsymbol{p}}_j$[2]. Therefore, one can take the sample covariance matrix of the MC distribution of the $\hat{\boldsymbol{p}}_j$ values as the covariance matrix associated with $\hat{\boldsymbol{p}}$.

A similar MC approach was described and implemented also in [27]. It is to be noted that such a simulation consists of a method for repeatedly simulating potential data in an experiment where the fitted curve happens to be close to $f(\hat{X}_i, \hat{\boldsymbol{p}})$, which is a different concept from that of propagating the probability distributions of the model input quantities, an idea which underlies Supplements 1 and 2 to the GUM [28, 29]. However, considering the estimation of the parameter vector $\boldsymbol{p}$ as an implicit multivariate measurement problem, it would be interesting to apply the propagation of probability distributions as addressed in Supplement 2, but such an investigation is beyond the scope of this paper.

We tested the reliability of the used MC simulation in the case of the WLS method applied to a straight-line model: we obtained practically the same uncertainties as those provided by the well-known analytical expressions for the uncertainty of WLS parameter estimates. Such a result is expected for models which are linear in the parameters [30]. However, in the case of a general WTLS problem, which is non-linear in the parameters, MC simulations may perform worse than closed formulae for the uncertainty, such as (7), although they are based on linear approximations of the model [30]. One should be aware of such a possibility when comparing the uncertainties evaluated in the two ways.

4. Validation of regression results and associated uncertainty

4.1. Data and methods for validation

For most of the following validations, Pearson's data [31] with York's weights [32], reported in table 1, and unit weights have been used. These data are extensively used within the literature [4, 6, 7, 25, 32, 33] for validation of software implementing WTLS, because, for such data, exact solutions for some polynomial fitting models are available [34–36]. This data set shows a case of a non-negligible variability on x with respect to y and, when considering York's weights, it represents a case of a highly varying uncertainty (strong heteroscedasticity) in both variables.

It is to be noted that these data show an essentially linear trend, hence the fact that we will also address other models (such as the cubic and the exponential) may seem not very meaningful from a statistical point of view, but it is for purposes of comparison with the few other results we could find in the literature dedicated to WTLS regression models other than the straight line.

For validating our algorithm in those cases in which general covariance matrices $\boldsymbol{U}_x$ and $\boldsymbol{U}_y$, not necessarily diagonal, need to be considered, we used data from the ISO/TS 28037 ([13], example in section 10). Relevant data and corresponding covariance matrices are reported in tables 2 and 3.

[2] Because of the non-linearity of the regression model, it usually happens that the mean of values $\hat{\boldsymbol{p}}_j$ differs from $\hat{\boldsymbol{p}}$.

Table 1. Pearson's data with York's weights. Corresponding uncertainties are given by $w_{x_i}^{-1/2}$ and $w_{y_i}^{-1/2}$, respectively.

x_i	w_{x_i}	y_i	w_{y_i}
0.0	1000.0	5.9	1.0
0.9	1000.0	5.4	1.8
1.8	500.0	4.4	4.0
2.6	800.0	4.6	8.0
3.3	200.0	3.5	20.0
4.4	80.0	3.7	20.0
5.2	60.0	2.8	70.0
6.1	20.0	2.8	70.0
6.5	1.8	2.4	100.0
7.4	1.0	1.5	500.0

Table 2. Data from example in [13, section 10].

x_i	y_i
50.4	52.3
99.0	97.8
149.9	149.7
200.4	200.1
248.5	250.4
299.7	300.9
349.1	349.2

Note that function (3) is symmetric with respect to $\boldsymbol{x}$ and $\boldsymbol{y}$, hence a validation procedure for the algorithm could consist in the comparison between the parameter estimates obtained when fitting y on x and those obtained when fitting x on y (the latter estimates being made comparable to the former accordingly to the inverse model). This was verified for the straight-line model (see table 4).

4.2. Uncorrelated input variables

In this subsection, we consider diagonal matrices $\boldsymbol{U}_x$ and $\boldsymbol{U}_y$ only.

4.2.1. The straight-line model. The validation was performed on the fitting model $y = ax+b$, applied on Pearson's data with York's and unit weights.

- *Pearson's data with York's weights.* The second column of table 4 shows the fitting parameter estimates and the corresponding uncertainties relevant to the fitting of y on x. The uncertainties were calculated by applying formula (7). In the third column, the relative differences between the parameter estimates and the true values provided by [34] are shown, as well as the relative differences between the calculated uncertainties and those obtained by means of MC simulations ($M = 5 \times 10^5$), performed as described in section 3.2. Concerning the parameter estimates, the proposed algorithm provided the best approximation to the problem solution among several fitting procedures approximating the exact result (see table 2 in [34]). Concerning the uncertainty evaluation,

Table 3. Covariance matrices associated with data from example in [13, section 10].

$$U_x = \begin{pmatrix} 0.50 & 0.00 & 0.25 & 0.00 & 0.25 & 0.00 & 0.25 \\ 0.00 & 1.25 & 1.00 & 0.00 & 0.00 & 1.00 & 1.00 \\ 0.25 & 1.00 & 1.50 & 0.00 & 0.25 & 1.00 & 1.25 \\ 0.00 & 0.00 & 0.00 & 1.25 & 1.00 & 1.00 & 1.00 \\ 0.25 & 0.00 & 0.25 & 1.00 & 1.50 & 1.00 & 1.25 \\ 0.00 & 1.00 & 1.00 & 1.00 & 1.00 & 2.25 & 2.00 \\ 0.25 & 1.00 & 1.25 & 1.00 & 1.25 & 2.00 & 2.50 \end{pmatrix} \quad U_y = \begin{pmatrix} 5 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 5 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 5 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 5 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 5 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 5 \end{pmatrix}$$

Table 4. Straight-line model for Pearson's data with York's weights: validation of the fitting parameters and associated uncertainties.

	$y - x$		$x - y$	
	Value	% diff	Value	% diff
$\chi^2_{\min}$	11.866 353 194 1	0	11.866 353 194 1	0
a	−0.480 533 407 44	2.8×10^{-9}	−0.480 533 407 43	4.0×10^{9}
b	5.479 910 223 95	-1.4×10^{-9}	5.479 910 223 92	2.0×10^{-9}
u_a	5.76×10^{-2}	−1.3	—	—
u_b	2.92×10^{-1}	−1.5	—	—
$u_{a,b}$	-1.62×10^{-2}	2.9	—	—

the approximations provided by formula (7) were equal to those given in table 3 of [7]. The uncertainties obtained via MC simulations tend to be slightly larger than them.

The fourth column of table 4 reports the corresponding results for the fitting of x on y, where a and b were made comparable to those in the second column by appropriate transformation. Results in the two columns are equal up to the tenth decimal digit (the uncertainty values in the case of the fitting of x on y are not relevant to such a comparison).

- *Pearson's data with unit weights.* When considering unit weights, we obtained $\chi^2 = 0.618\,572\,759\,437\,045$, which is equal to the exact solution reported in [35, table III] up to the twelfth decimal digit (the exact values of the parameters to compare our estimates are not available in the literature).

4.2.2. The cubic model. The validation was performed on the fitting model $y = ax^3 + bx^2 + cx + d$, applied on Pearson's data with York's and unit weights.

- *Pearson's data with York's weights.* We obtained $\chi^2 = 10.486\,904\,057\,7079$, equal to the exact solution up to the tenth decimal digit, to which the exact solution is reported in [35, table III].

  The parameter estimates were compared with those provided in [7], which were obtained by means of the B_LEAST software [15], recommended by [12]. The relevant differences were calculated at the sixth decimal digit, since the results in [7] are reported with that numerical accuracy. The corresponding relative differences are shown in the third column of table 5. Actually, we saw that relative differences smaller than 3×10^{-6} could be obtained when running the B_LEAST software with different settings of the algorithm convergence. Concerning the uncertainty evaluation, the approximations provided by our software were equal to those provided by the B_LEAST software.

Table 5. Cubic model for Pearson's data with York's and unit weights: validation of the fitting parameters and associated uncertainties.

	York's weights		Unit weights	
	Value	% diff	Value	% diff
a	−0.011 556 565 379	0.03	−0.013 240 528 570	0.00
b	0.157 154 323 493	−0.02	0.152 471 601 429	0.00
c	−1.108 353 203 572	0.01	−0.999 835 346 653	0.00
d	6.142 329 401 915	0.00	6.015 263 733 009	0.00
u_a	1.00×10^{-2}	−44	4.05×10^{-2}	−81
u_b	1.36×10^{-1}	−33	4.72×10^{-1}	−80
u_c	5.83×10^{-1}	−20	1.55	−81
u_d	7.79×10^{-1}	−8	1.36	−83
$u_{a,b}$	-1.32×10^{-3}	63	-1.88×10^{-2}	96
$u_{a,c}$	5.15×10^{-3}	−55	5.67×10^{-2}	−96
$u_{a,d}$	-5.35×10^{-3}	43	-3.26×10^{-2}	97
$u_{b,c}$	-7.65×10^{-2}	46	-7.03×10^{-1}	96
$u_{b,d}$	8.58×10^{-2}	−34	4.40×10^{-1}	−97
$u_{c,d}$	-4.19×10^{-1}	24	−1.74	97

The third column of table 5 also reports the relative differences between the calculated uncertainties and those obtained by means of MC simulations ($M = 5 \times 10^5$). Here, the differences between the uncertainties obtained using equation (7) and those obtained via MC simulations are even more conspicuous than those reported in table 4 for the straight-line model, which may be due to the increased non-linearity of the model.

- *Pearson's data with unit weights.* We obtained $\chi^2 = 0.485\,152\,486\,927\,038$, equal to the exact solution up to the twelfth decimal digit, to which the exact solution is reported in [35, table 3].

  The parameter estimates were compared with those provided in [36]. The relevant differences were calculated at the eighth significant digit, since the results in [36] are reported with that numerical accuracy, and result in being all equal to zero.

Table 6. Exponential model for Pearson's data with York's weights: validation of the fitting parameters and associated uncertainties.

	York's weights	
	Value	% diff
a	95.690 719 390 590	−0.38
b	−90.234 033 497 515	0.41
c	0.005 200 660 678	0.40
u_a	1.27×10^{1}	—
u_b	1.26×10^{1}	—
u_c	7.12×10^{-4}	—
$u_{a,b}$	-1.60×10^{2}	—
$u_{a,c}$	-5.47×10^{-3}	—
$u_{b,c}$	5.53×10^{-3}	—

The fifth column of table 5 also reports the relative differences between the calculated uncertainties and those obtained by means of MC simulations ($M = 5 \times 10^5$). Here, such differences are even larger than those in the third column, because the unit weights used in the regression lead to higher uncertainties, which emphasize the non-linearity effects.

4.2.3. The exponential model. The validation was performed on the fitting model $y = a + be^{cx}$, applied on Pearson's data with York's weights. We obtained $\chi^2 = 11.863\,655\,879\,364$, equal to the solution reported in [7, table 2] up to the fourth decimal digit.

The parameter estimates were compared with those provided in [7] as well: the relevant differences were calculated at the fifth decimal digit, since the results in [7] are reported at that precision, and the corresponding relative differences are shown in the third column of table 6. Such values show a certain gap between the two fitting methods, still lying, however, within the associated uncertainties (reported in the same table). Within this validation example, unfortunately, we encountered serious numerical problems in applying the matrix inversions involved in equation (9). Note, in fact, that the exponential model can be approximated by the linear model when $cx \ll 1$; as a result, the least-squares algorithm will essentially try to fit three parameters to a straight-line model, and the associated covariance matrix will be essentially singular. Hence, the above-mentioned uncertainties could be obtained only by means of MC simulations ($M = 5 \times 10^5$). No uncertainty was provided by [7] for this model, hence a comparison is not possible.

4.3. Correlated input variables

In this subsection, we consider general matrices $\boldsymbol{U}_x$ and $\boldsymbol{U}_y$.

4.3.1. The straight-line model. The validation was performed to the fitting model $y = ax + b$, applied on data reported in table 2 with associated covariance matrices reported in table 3. We compared the results obtained with our algorithm with those produced by software [18] and reported in [13]. The second column of table 7 reports the $\chi^2_{\min}$ value of the fit and the parameter estimates together with the associated uncertainties and the covariance term. The third column shows the corresponding relative differences between such values and those given in [13].

Table 7. Straight-line model for data from the example in [13, section 10]: validation of fitting parameters and associated uncertainties.

	Value	% diff
$\chi^2_{\min}$	1.771 847 450 960	2.81×10^{-9}
a	1.001 230 760 542	-2.28×10^{-7}
b	0.342 395 888 828	-1.44×10^{-3}
u_a	9.01×10^{-3}	3.24×10^{-2}
u_b	2.06	-2.18×10^{-2}
$u_{a,b}$	-1.29×10^{-2}	1.27×10^{-2}

Table 8. Calibration data set: amount fractions and GC peak areas from [12, B.2.2 example 2, table B.7.]

Amount fraction		GC peak area	
y/mmol mol^{-1}	$u(y)$/mmol mol^{-1}	x/counts	$u(x)$/counts
1.5×10^{-3}	9.000×10^{-4}	6.000×10^{1}	3.500×10^{1}
1.888×10^{-1}	4.500×10^{-4}	7.786×10^{3}	1.357×10^{2}
1.990	4.000×10^{-3}	8.170×10^{4}	3.670×10^{1}
3.796	3.900×10^{-2}	1.562×10^{5}	2.232×10^{2}
5.677	1.250×10^{-2}	2.333×10^{5}	1.372×10^{2}
7.118	1.250×10^{-2}	2.930×10^{5}	2.455×10^{2}
9.210	2.000×10^{-2}	3.806×10^{5}	1.251×10^{2}
1.090×10^{1}	2.500×10^{-2}	4.497×10^{5}	3.218×10^{2}

Concerning the uncertainty and the covariance terms, we obtained relative differences of about 3×10^{-2} between our values and those obtained with a MC simulation ($M = 10^6$).

Neglecting correlation among x_i and y_i, i.e. taking the covariance matrices in table 3 as diagonal, we noted a relative variation of up to 10% in the parameter estimates and up to 38% in the uncertainty and covariance terms, which show the importance of properly taking correlations into account.

5. Example: application to calibration of GCs

We consider here an example about the calibration gas of a chromatograph (GC) reported in ISO 6143 [12, B.2.2 example 2]. In general, a GC is calibrated with respect to a set of gaseous certified reference materials (CRM), i.e. standard gas mixtures [6, 12]. The instrument responses are the peak areas of the chromatogram corresponding to each analysed gas component for each CRM.

In the present example, nitrogen (N_2) in a synthetic natural gas is analysed. The calibration data set consists of eight data points, reported in table 8 together with the associated uncertainties: y are the amount fractions of N_2 within the CRMs and x are the GC relevant peak areas.

We fitted the data with a second-order polynomial function $y = ax^2 + bx + c$ by means of our algorithm, in order to determine the GC analysis function for N_2.

The fitting results obtained when considering y and x values uncorrelated are reported in the second column of table 9 and can be compared with the results in [12], here reported in the third column.

Table 9. Results of WTLS fitting on data reported in table 8.

	Uncorrelated CRMs	ISO 6143	Correlated CRMs	% diff
$\chi^2_{\min}$	1.40	1.47	1.28	−8.24
a	-4.0373×10^{-13}	-4.1096×10^{-13}	-4.2247×10^{-13}	−4.64
b	2.4400×10^{-5}	2.4403×10^{-5}	2.4403×10^{-5}	0.01
c	-1.2895×10^{-4}	-1.4037×10^{-4}	-1.3538×10^{-4}	−4.99
u_a	1.895×10^{-13}	1.895×10^{-13}	1.804×10^{-13}	−4.81
u_b	5.901×10^{-8}	5.901×10^{-8}	5.644×10^{-8}	−4.35
u_c	1.175×10^{-3}	1.175×10^{-3}	1.174×10^{-3}	−0.13
$u_{a,b}$	-1.020×10^{-20}	-1.020×10^{-20}	-9.106×10^{-21}	10.72
$u_{a,c}$	4.667×10^{-17}	4.667×10^{-17}	4.304×10^{-17}	−7.79
$u_{b,c}$	-2.057×10^{-11}	-2.057×10^{-11}	-1.9609×10^{-11}	4.67

We also considered the case of correlated CRMs as described in [12], introducing a covariance term equal to 0.000 16 (in mmol mol^{-1} counts) between the fourth and the seventh CRMs (because the former is obtained by dilution from the latter) and another covariance term equal to 0.0001 (in mmol mol^{-1} counts) between the fifth and the eighth CRMs (because both are traceable back to a common set of primary standards). The corresponding fitting parameters are reported in the fourth column of table 9. Note that in [12], such a case was treated only for the straight-line model, hence a direct comparison between the method applied in the reference and the WTLS method is not possible. The fifth column of table 9 reports the relative differences between fitting results in the fourth and second columns, showing the error that one would make when neglecting the covariance terms between the CRMs.

6. Conclusions

Compared with the software available within the metrological area addressing the WTLS method for calibration curves, the presented algorithm has the advantage of dealing with both uncertain and correlated variables for any desired fitting model, even when this is strongly non-linear in the parameters. It is a very powerful and versatile tool for WTLS implementation, especially for calibration purposes, its main advantages being the following:

- no restriction on the choice of the fitting model;
- no restriction on the form of the covariance matrices associated with the dependent and independent variables;
- easy implementation;
- very good accuracy in the parameter estimation (verified by validation);
- implementation of an approximated formula for the uncertainty evaluation, appropriately derived for minimization problems.

A Monte Carlo method was also developed both for purposes of comparison with the formula for the uncertainty evaluation implemented within the software, and as an independent method for uncertainty evaluation. The comparison results showed that, for increasing non-linearity of the models, the uncertainties provided by the two methods can substantially depart one from the other, the MC uncertainties being larger than those provided by formula (7). Such a result deserves a more careful investigation; a proper implementation of Supplements 1 and 2 to the GUM could also be considered in order to evaluate the uncertainties associated with the WTLS parameter estimates.

It is planned to make the algorithm freely available at the File Exchange of the MATLAB® Central.

Acknowledgments

The research leading to the results described in this paper is part of the European Metrology Research Programme (EMRP), which is jointly funded by the EMRP participating countries within EURAMET and the European Union.

The authors wish to thank the two referees for valuable comments and suggestions for improving this paper.

Appendix. The WTLS algorithm

In the following, we give a short description of the algorithm:

(i) Input of data $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{U}_x$ and $\boldsymbol{U}_y$.
(ii) Standardization of the data.
(iii) Definition of the cost function χ^2 within a MATLAB® file `myfun.m`. As an example, in the following, we show the MATLAB® code relevant to a straight-line model fitting ten experimental points:

```
function [chi2, g] = myfun(p)
global x y Vxinv Vyinv dx dy;
dx   = x - p(1:end-2);
dy   = y - p(end-1)*p(1:end-2) - p(end);
chi2 = dx*Vxinv*dx' + dy*Vyinv*dy';

    if nargout > 1
      g(1) = -2*dx(1)*Vxinv(1,1)
        - 2*p(11)*dy(1)*Vyinv(1,1);
      g(2) = -2*dx(2)*Vxinv(2,2)
        - 2*p(11)*dy(2)*Vyinv(2,2);
      ⋮
      g(10) = -2*dx(10)*Vxinv(10,10)
        - 2*p(11)*dy(10)*Vyinv(10,10);
      g(11) = -2*(p(1)*dy(1)*Vyinv(1,1)
        + ... + p(10)*dy(10)*Vyinv(10,10));
      g(12) = -2*(dy(1)*Vyinv(1,1)
        + ... + dy(10)*Vyinv(10,10));
    end
```

In the above command lines, input global variables x, y, Vxinv and Vyinv are the standardized input data and relevant inverse covariance matrices; p is the parameter vector $[\boldsymbol{X}, \boldsymbol{p}]$, made of ten abscissae to be adjusted, i.e. p(1:end-2), and two fitting parameters pertaining to the straight-line model, i.e. p(end-1) and p(end) (end is equal to twelve, in this case); dx, dy and chi2 implement expression (3). Functions g(i) describe the analytical expression for the first derivative of χ^2 with respect to the ith parameter. Defining the gradient g in this way, it is possible to apply the GradObj option.

(iv) Calculation of the starting point $[\boldsymbol{X}_0, \boldsymbol{p}_0]$ for the parameter estimates, by means of the OLS method applied to the considered fitting model.

(v) WTLS estimation of the model parameters by means of the MATLAB® function fminunc:

```
[par_opt, fval, exitflag] = fminunc(@myfun,
x0,...  ...optimset('GradObj','on','TolX',
1e-9,'TolFun',1e-10,...  ...'MaxFunEvals',
3000, 'LargeScale','off'));
```

where

- par_opt is the solution $[\hat{\boldsymbol{X}}, \hat{\boldsymbol{p}}]$ of the minimization of the object function myfun,
- fval is the minimum value ($\chi^2_{\min}$) of myfun at the solution par_opt,
- exitflag is the value identifying the reason for the termination of the algorithm,
- myfun is the object function (3) to be minimized,
- x0 is the starting point $[\boldsymbol{X}_0, \boldsymbol{p}_0]$ of the iteration process,
- optimset creates optimization option structures. When a user-defined gradient of the objective function is provided, option GradObj is set to "on". The default "off" setting causes fminunc to estimate gradients using finite differences. TolX and TolFun are the termination tolerances on the function value and on the solution, respectively. MaxFunEvals is the maximum number of function evaluations allowed. When LargeScale is set "on", fminunc uses a minimization large-scale algorithm, when it is set "off", a medium-scale algorithm is implemented. For more details on fminunc, refer to the Help of the Optimization Toolbox.

(vi) Transformation of the obtained estimates (relevant to the standardized data) back into the estimates of the original parameter.

(vii) Evaluation of the covariance matrix, associated with the parameter estimates, by implementation of formula (7):

```
q0 = [par_opt, x, y];
H' = hessian(@chisquare_hessian, q0);
H = H'(1:npar, 1:npar);
Din = H'(1:npar, npar+1:npar+(2*N));
O = zeros(N);
Vin = [Vx, O; O, Vy];
Vout = inv(H)*Din*Vin*Din'*inv(H);
```

where x and y are the standardized input data, again, and par_opt is the solution of the minimization algorithm; q0 is the point at which the hessian H' of function chisquare_hessian is calculated by means of function hessian. The latter can be any MATLAB® function able to calculate the Hessian matrix of a scalar function (we used a function available within the File Exchange of the MATLAB® Central). Function chisquare_hessian is again an implementation of the cost function χ^2, which is now seen as a function not only of parameters $[\boldsymbol{X}, \boldsymbol{p}]$ but also of the input data $\boldsymbol{x}$ and $\boldsymbol{y}$, treated as variables. The chisquare_hessian.m function of the considered example, relevant to a straight-line model fitting ten experimental points, is shown in the following:

```
function chi2_hes = chisquare_hessian(q)
global npar Vxinv Vyinv dx dy;
dx = q(npar+1:2*npar-2) - q(1:npar-2);
dy = q(2*npar-1:end) - q(npar-1)*q(1:npar-2)
- q(npar);
chi2_hes = dx*Vxinv*dx' + dy*Vyinv*dy';
```

where npar, in the considered case, is equal to twelve and end is equal to thirty-two (ten abscissae and ten ordinates are added as parameters). Therefore, matrix H' contains all the second-order derivatives of the cost function. The relevant submatrices are H and Din, which are needed for calculating expression (9). Matrix Vin is constructed according to (8) and, finally, the covariance matrix Vout is obtained according to (7).

(viii) Extraction, from matrix Vout, of the submatrix relevant to the uncertainties and the covariances associated with estimates $\hat{\boldsymbol{p}}$.

(ix) Transformation of the obtained covariance matrix (relevant to the standardized data) back into the covariance matrix of the original parameter.

References

[1] Markovsky I and Van Huffel S 2007 *Signal Process.* **87** 2283–302

[2] Van Huffel S and Vandewalle J 1991 *The Total Least Squares Problem* (Philadelphia, PA: SIAM)

[3] Fuller W A 1987 *Measurement Error Models* (New York: Wiley)

[4] Krystek M and Anton M 2007 *Meas. Sci. Technol.* **18** 3438–42

[5] Krystek M and Anton M 2011 *Meas. Sci. Technol.* **22** 035101 (9pp)

[6] Milton M J T, Harris P M, Smith I M, Brown A S and Goody B A 2006 *Metrologia* **43** S291–8

[7] Bremser W and Hässelbarth W 1997 *Anal. Chim. Acta* **348** 61–9

[8] Forbes A B 1993 *Numer. Algorithms* **5** 523–34

[9] Forbes A B, Harris P M and Smith I M 2002 Generalised Gauss–Markov regression *Algorithms for Approximation IV* ed I Anderson *et al* (Huddersfield: University of Huddersfield) pp 270–7

[10] Bremser W, Viallon J and Wielgosz R I 2007 *Metrologia* **44** 495–504

[11] Cox M G, Forbes A B, Harris P M and Smith I 2004 The classification and solution of regression problems for calibration *NPL Report* CMSC 24/03, National Physical Laboratory, Teddington, UK; available from http://publications.npl.co.uk/npl_web/pdf/cmsc24.pdf

[12] ISO/TC 158 2001 ISO 6143:2001(E), Gas analysis—Comparison methods for determining and

checking the composition of calibration gas mixtures, 2nd edn
[13] ISO/TC 69 (SC 6) 2010 ISO/TS 28037:2010(E), Determination and use of straight line calibration functions, 1st edn
[14] DIN 1319-4 1985 Basic concepts of measurements; treatment of uncertainties in the evaluation of measurements Please note that the document has been replaced by DIN 1319-4:1999-02, Fundamentals of metrology: part 4. Evaluation of measurements; uncertainty of measurement.
[15] Bremser W 2001 *Calibration Tool B_LEAST* Software supporting implementation of ISO Standard 6143, developed by the Federal Institute for Materials Research and Testing (BAM), Berlin, Germany
[16] Smith I M and Onakunle F O 2007 SSfM-3 1.6.1—XLGENLINE, Software for generalised least-squares fitting, developed by the National Physical Laboratory (NPL), Teddington, UK, NPL document reference: CMSC/M/06/657
[17] Boggs P T, Byrd R H, Rogers J E and Schnabel R B 1992 ODRPACK, developed by National Institute of Standards and Technology, Boulder, CO, NIST document reference: NISTIR 4834, User's Reference Guide for ODRPACK Version 2.01 Software for Weighted Orthogonal Distance Regression
[18] 2010 Software to Support ISO/TS 28037:2010(E), developed by the National Physical Laboratory (NPL), Teddington, UK
[19] Nielsen L 1998 *Metrologia* **35** 115–8
[20] Nielsen L 2002 Evaluation of measurements by the method of least squares *Algorithms for Approximation IV* ed I Anderson *et al* (Huddersfield: University of Huddersfield) pp 170–86
[21] Schielzeth H 2010 *Methods Ecol. Evol.* **1** 103–13
[22] Balsamo A, Mana G and Pennecchi F 2006 *Metrologia* **43** 396–402
[23] Scarpa F 1998 *Metrologia* **35** 1–5
[24] Draper N R and Smith H 1966 *Applied Regression Analysis* (New York: Wiley)
[25] Cecchi G C 1991 *Meas. Sci. Technol.* **2** 1127–8
[26] Malengo A and Bich W 2012 *Metrologia* **49** 289–93
[27] York D and Evensen N M 2004 *Am. J. Phys.* **72** 367–75
[28] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP and OIML *Evaluation of Measurement Data—Supplement 1 to the 'Guide to the Expression of Uncertainty in Measurement'—Propagation of distributions using a Monte Carlo method* JCGM 101:2008 www.bipm.org/utils/common/documents/jcgm/JCGM_101_2008_E.pdf
[29] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP and OIML. *Evaluation of Measurement Data—Supplement 2 to the 'Guide to the Expression of Uncertainty in Measurement'—Extension to any number of output quantities* JCGM 102:2011 www.bipm.org/utils/common/documents/jcgm/JCGM_102_2011_E.pdf
[30] Willink R 2013 *Measurement Uncertainty and Probability* (Cambridge: Cambridge University Press)
[31] Pearson K 1901 *Phil. Mag.* **2** 559–72
[32] York D 1966 *Can. J. Phys.* **44** 1079–86
[33] Reed B C 1992 *Am. J. Phys.* **60** 59–62
[34] Cantrel C A 2008 *Atmos. Chem. Phys. Discuss.* **8** 6409–36
[35] Lybanon M 1984 *Am. J. Phys.* **52** 22–6
[36] Southwell W H 1969 *J. Comput. Phys.* **4** 465–74