Computer Physics Communications 182 (2011) 419–434

Contents lists available at ScienceDirect

Computer Physics Communications

www.elsevier.com/locate/cpc

The program LOPT for least-squares optimization of energy levels ☆

A.E. Kramida

National Institute of Standards and Technology, 100 Bureau Dr., MS 8422, Gaithersburg, MD 20899, United States

ARTICLE INFO

Article history:
Received 14 June 2010
Received in revised form 29 September 2010
Accepted 30 September 2010
Available online 12 October 2010

Keywords:
Atomic energy levels
Ritz wavelengths
Least-squares optimization

ABSTRACT

The article describes a program that solves the least-squares optimization problem for finding the energy levels of a quantum-mechanical system based on a set of measured energy separations or wavelengths of transitions between those energy levels, as well as determining the Ritz wavelengths of transitions and their uncertainties. The energy levels are determined by solving the matrix equation of the problem, and the uncertainties of the Ritz wavenumbers are determined from the covariance matrix of the problem.

Program summary

Program title: LOPT
Catalogue identifier: AEHM_v1_0
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEHM_v1_0.html
Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland
Licensing provisions: Standard CPC licence, http://cpc.cs.qub.ac.uk/licence/licence.html
No. of lines in distributed program, including test data, etc.: 19 254
No. of bytes in distributed program, including test data, etc.: 427 839
Distribution format: tar.gz
Programming language: Perl v.5
Computer: PC, Mac, Unix workstations
Operating system: MS Windows (XP, Vista, 7), Mac OS X, Linux, Unix (AIX)
RAM: 3 Mwords or more
Word size: 32 or 64
Classification: 2.2
Nature of problem: The least-squares energy-level optimization problem, i.e., finding a set of energy level values that best fits the given set of transition intervals.
Solution method: The solution of the least-squares problem is found by solving the corresponding linear matrix equation, where the matrix is constructed using a new method with variable substitution.
Restrictions: A practical limitation on the size of the problem N is imposed by the execution time, which scales as N^3 and depends on the computer.
Unusual features: Properly rounds the resulting data and formats the output in a format suitable for viewing with spreadsheet editing software. Estimates numerical errors resulting from the limited machine precision.
Running time: 1 s for $N = 100$, or 60 s for $N = 400$ on a typical PC.

1. Introduction

The problem of energy level optimization consists of finding a set of energy level values that best fits the set of observed wavenumbers of transitions between these levels. The basic method of solution is a least squares fitting, in which the sum of weighted squares of deviations of observed wavenumbers from their Ritz values is minimized. This minimization can, in principle, be accomplished by different methods.

One of programs widely used for this purpose in atomic spectroscopy is ELCALC by Radziemski et al. [1]. This program is based on the "point-relaxation" iterative method [2] and has two major limitations. First, it requires a division of all energy levels into two sets of different parities and allows for inclusion of only "allowed" transitions, i.e. transitions between levels of different

☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

E-mail address: alexander.kramida@nist.gov.

0010-4655/$ – see front matter
doi:10.1016/j.cpc.2010.09.019

parities. Second, it cannot calculate the uncertainties of predicted (Ritz) wavenumbers of transitions between the optimized levels.

Another method of solution is to solve a set of linear differential equations by constructing and inverting a proper matrix of coefficients. This approach was implemented by Radziemski et al. [3]. Their program did not get much use, mainly because of an error in the expression for the level uncertainties; consequently, levels determined by only one transition always had zero uncertainties.

An interesting alternate approach for determination of the level uncertainties was suggested and described in detail by G.J. van het Hof in the appendix to his thesis [4]. He introduced a concept of "error current", replacing the graph of energy levels connected by a set of observed transitions with an analogous electric circuit. In this circuitry, transitions are replaced by resistances equal to squares of measurement uncertainties. In order to calculate relative uncertainty of any pair of levels, one should apply an arbitrary "voltage" between these levels and find the total "error current" between them. This would yield the total "resistance" between these two levels. The uncertainty of the energy difference (wavenumber of the transition) is equal to square root of this "resistance". Although this method gives a good intuitive understanding of the origin of the uncertainties of the Ritz wavenumbers, its implementation requires a repetitive solution of the problem of finding the "error currents" for each pair of levels involved, and thus leads to a large time for the solution of the complete problem. However, it was found that the solution is equivalent to finding the uncertainties of the Ritz wavenumbers from the covariance matrix of the least-squares problem [5], which can be done much faster.

A technically simple implementation of the matrix-inversion approach was recently described by Öberg [6]. He used the routines for matrix operations built in the MatLab package. His implementation includes one feature that was missing in all previously published level-optimization codes, namely, the determination of the χ^2 parameter of the problem. For the level optimization problem, this parameter provides a statistical measure of how reasonable the stated uncertainties of the input wavenumbers are. If the problem includes N excited energy levels and M observed transitions between them, the measured wavenumbers have normal statistical distribution (with no systematic errors), and the given measurement uncertainties for all observed transitions correspond to statistical standard deviations, then χ^2 should be close to the degrees of freedom of the problem, i.e. $M-N$. If it is significantly smaller, it usually means that the measurement uncertainties are strongly overestimated. If it is significantly greater, it may indicate significant flaws in the input transition data, such as incorrect identifications, underestimation of the measurement uncertainties, or presence of systematic shifts. It should be noted that using the concept of χ^2 implies a requirement for the input quantities (i.e. measured transition wavenumbers) to have a normal statistical distribution. It is better to use instead the residual sum of squares (RSS) of deviations of the measured wavenumbers from the corresponding Ritz values. This quantity is computed exactly in the same way as χ^2 and provides the same measure of quality of the data; however, it is independent of the statistical distribution of the input data.

The first version of the program LOPT was created at the National Institute of Standards and Technology in 1995. Its name LOPT stands for Level Optimization. The program was further developed during 1997–2006. The first brief description of the program and some of its features was given by Kramida et al. [5]. A more detailed description was given by Kramida and Nave [7]. In the latter article it was mentioned that the original code, written in the Turbo Pascal programming language, is not portable to operating systems other than MS Windows, and that the publication of the source code and its full description awaits the completion of porting the code to a programming language that would allow execution of the code under any operating system. This porting has now been completed, and the present article gives the full description of the program and includes the source code in Perl programming language.

The present version of the LOPT code includes many enhancements, such as determination of RSS, and is easy to use on any computer. It is written in the Perl language freely available for any operating system, and does not require any additional specialized software.

2. Program description

2.1. Basic method

The basic method is similar to the one described by Radziemski et al. [3]. However, the formalism employed here is different. The problem is formulated as it was by van het Hof [4]. The solution is obtained in two stages. In the first stage, the optimized energy levels are found by solving the matrix equation corresponding to a system of linear equations. This stage does not involve matrix inversion. In the second stage, the matrix equation is modified and the inverse matrix is found. Then the uncertainties of the Ritz wavenumbers are found as linear combinations of the elements of this inverse matrix.

The level-optimization problem can be formulated as minimization of the function F defined as

$$F=\sum_{i,j}(\Delta E_{ij}-s_{ij})^2 w_{ij}, \tag{1}$$

where s_{ij} is the measured wave number of the transition between the unknown energy levels E_i and E_j (s_{ij} is positive if $E_i<E_j$ and negative otherwise), $\Delta E_{ij}=E_j-E_i$, and the weight $w_{ij}=d_{ij}^{-2}$ is equal to the square of the reciprocal measurement uncertainty (dispersion). The sum includes M terms for all values of the indexes i and j corresponding to observed transitions (M is the number of observed transitions). The index i of the initial energy level E_i of a transition $E_i\to E_j$ varies from zero to the number of excited levels N, while the index j of the final level E_j varies from one to N, and the ground level is defined as $E_0=0$. Each observed transition is present in Eq. (1) only once (i.e., if there is a term corresponding to $E_i\to E_j$, there is no term corresponding to $E_j\to E_i$).

This leads to a set of N linear equations of the form

$$\sum_j E_j W_{ij}=S_i \quad (i,j=1\ldots N), \tag{2}$$

where W_{ij} is a matrix of coefficients,

$$W_{ij}=\delta_{ij}\sum_k w_{kj}-w_{ij} \quad (i,j=1\ldots N;\ k=0\ldots N), \tag{3}$$

where δ_{ij} is the Kronecker delta function, and S_i are linear combinations of s_{ij} and w_{ij}:

$$S_i=\sum_j w_{ij}s_{ij} \quad (i=1\ldots N;\ j=0\ldots N). \tag{4}$$

Among the weights w_{ij}, only those corresponding to observed transitions $E_i\to E_j$ are positive; the rest of them are zero. This implies $w_{ii}=0$ for all i. It should be noted that the terms in the sum in Eq. (4) have different signs depending on the ordering of the energy levels corresponding to the transition $E_i\to E_j$. They are positive if $E_i<E_j$ and negative otherwise. In fact, the exact level ordering need not be known in advance. However, some assumption about level ordering must be consistently followed in all

transition identifications. If after the level optimization procedure the sign of some of the resulting energy intervals ΔE_{ij} turns out to be different from the initially assumed, it would simply mean that the initial ordering was incorrect; it has no effect on the solution.

Solution of the equation system (2) can be found by inverting the matrix W_{ij}:

$$E_i = \sum_j W_{ij}^{(-1)} S_j. \tag{5}$$

This is the approach used by Radziemski et al. [3]. A faster and numerically more stable method is to decompose the matrix W_{ij} using, for example, the *LU* decomposition (see Demmel [8]):

$$W_{ij} = LU, \tag{6}$$

where L is a unit lower triangular matrix, and U is an upper triangular matrix, then solve

$$Ly = S \tag{7}$$

for vector y, and finally solve

$$UE = y \tag{8}$$

to find E. A similar approach using the Cholesky decomposition, which has better numerical properties, will be described in the following sections.

Although the solutions employing a matrix decomposition are faster than inverting the matrix, we will still have to find the inverse matrix W^{-1} in order to compute the uncertainties of the calculated transition wavenumbers. This is explained in the following section.

2.2. Ritz wavenumber uncertainties

In addition to finding the optimized level values, the program LOPT calculates uncertainties of the predicted (Ritz) wavenumbers. The method of this calculation, based on the use of the covariance matrix, was originally developed by Radziemski et al. [3]. Using statistical theory, it can be rigorously shown that, if the wavenumber measurements s_{ij} are uncorrelated, then the dispersions a_{ij} of the energy differences $E_j - E_i$ can be determined as a simple combination of elements of the inverse matrix W^{-1} [3,4]:

$$\begin{aligned} a_{ij}^2 &= \mathrm{Var}(E_i) + \mathrm{Var}(E_j) - 2\,\mathrm{Cov}(E_i, E_j) \\ &= W_{ii}^{(-1)} + W_{jj}^{(-1)} - 2W_{ij}^{(-1)}. \end{aligned} \tag{9}$$

It should be pointed out that, according to the statistical theory [9,10], the assumption of a normal statistical distribution of measurements s_{ij} is not necessary for the rigorous derivation of all relations given above, in contrast to statements in [3,4]. Only the absence of correlations between s_{ij} is required.

Thus, the validity of this approach is restricted to the case of absence of systematic errors in measured wavelengths. If some systematic errors are present, in certain cases they can result in progressive accumulation of systematic error in level values from the ground state to upper levels. To our knowledge, there is no rigorous method developed for the estimation of level uncertainties caused by systematic errors.

2.3. Variable substitution

Although the formal solution of the problem described by Eqs. (2)–(9) is fairly straightforward, in many cases it is numerically difficult, because the matrix W can be ill-formed (either in the sense of bad scaling or in the sense of a too large matrix condition number). The first observation is that from Eq. (3) it follows that the diagonal elements of this matrix involve summation of transition weights, which can include both very small and very large quantities, as the observed transitions can have vastly different measurement uncertainties. It is rather common in atomic spectroscopy that small separations between excited levels are measured with much greater accuracy than wavenumbers of transitions connecting these levels to the ground level or another relatively distant level. Then the summation in Eq. (3) can lead to a significant loss of precision due to machine rounding of floating-point numbers. The elements of the right-hand side vector S given by Eq. (4) are computed as sums of quantities having different signs, which can involve numerical errors due to cancellation. In addition to that, the matrix W can be ill-conditioned, so that computation of the inverse matrix W^{-1} can involve large numerical errors. This is illustrated by the following simple example of a three-level system depicted in Fig. 1.

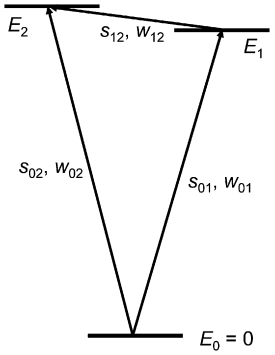

Fig. 1. An example of a system with three observed transitions defining two unknown levels E_1 and E_2. The measured transition wavenumbers s_{01}, s_{02}, and s_{12} have uncertainties d_{01}, d_{02}, and d_{12}, corresponding to weights w_{01}, w_{02}, and w_{12} (see text).

The levels $E_0 = 0$, E_1, and E_2 are connected by three lines with measured wavenumbers s_{01}, s_{02}, and s_{12} with uncertainties d_{01}, d_{02}, and d_{12} such that $d_{01} \approx d_{02}$ and $d_{12} \ll d_{01}, d_{02}$. We solve the problem of finding the two unknown levels E_1 and E_2 by minimizing the function F in Eq. (1) with weights $w_{01} = 1/d_{01}^2$, $w_{02} = 1/d_{02}^2$, and $w_{12} = 1/d_{12}^2$, such that $w_{01} \approx w_{02}$ and $w_{12} \gg w_{01}, w_{02}$. Then the system of equations (2) has the following form:

$$\begin{bmatrix} w_{01}+w_{12} & -w_{12} \\ -w_{12} & w_{02}+w_{12} \end{bmatrix} \begin{pmatrix} E_1 \\ E_2 \end{pmatrix} = \begin{pmatrix} w_{01}s_{01} - w_{12}s_{12} \\ w_{02}s_{02} + w_{12}s_{12} \end{pmatrix}. \tag{10}$$

Dividing both rows by w_{12} and defining $\alpha_1 = w_{01}/w_{12}$, $\alpha_2 = w_{02}/w_{12}$, we obtain

$$W = \begin{bmatrix} 1+\alpha_1 & -1 \\ -1 & 1+\alpha_2 \end{bmatrix}, \qquad S = \begin{pmatrix} \alpha_1 s_{01} - s_{12} \\ \alpha_2 s_{02} + s_{12} \end{pmatrix} \tag{11}$$

and

$$W^{-1} = \frac{1}{\alpha_1+\alpha_2+\alpha_1\alpha_2} \begin{bmatrix} 1+\alpha_2 & 1 \\ 1 & 1+\alpha_1 \end{bmatrix}. \tag{12}$$

If the matrix is inverted using *LU* decomposition with row pivoting, the magnitude of numerical errors in the solution is mainly determined by the condition number κ of the matrix W [8,11,12]:

$$\begin{aligned} &\frac{\|\dot{E} - E\|}{\|E\|} \lesssim \kappa \cdot \mathbf{eps}, \quad \kappa = \|W\|\,\|W^{-1}\| = \frac{(1+\alpha_{\max})^2}{\alpha_1+\alpha_2+\alpha_1\alpha_2}, \\ &\alpha_{\max} = \max(\alpha_1, \alpha_2), \end{aligned} \tag{13}$$

where E is the exact solution, $\dot{E}$ is the approximate numerical solution, **eps** is the machine precision of digital representation of floating-point numbers (**eps** $\approx 2.2 \times 10^{-16}$ in the case of 64-bit double precision).

Thus, if $\alpha_1 \approx \alpha_2$ and $\alpha_1 \ll 1$, κ can be large, and so can be the relative error in the approximate solution $\dot{E}$. It is not uncommon to have $d_{12}/d_{01} \sim 10^{-6}$. Then $\alpha_1 \sim 10^{-12}$ and $\kappa\,\mathbf{eps} \sim 10^{-4}$, which is still much smaller than unity, ensuring that Eqs. (13) are valid; however, such accuracy may well be unacceptably poor for the determination of the level positions.

The solution can be greatly improved if we use the following variable substitution: $E_1 = x_1$ and $E_2 = x_1 + x_2$. Then, by differentiating Eq. (1) over x_1 and x_2, instead of the system (10) we obtain the following equation system:

$$\begin{bmatrix} w_{01}+w_{02} & w_{02} \\ w_{02} & w_{02}+w_{12} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} w_{01}s_{01}+w_{02}s_{02} \\ w_{02}s_{02}+w_{12}s_{12} \end{pmatrix}. \quad (14)$$

Dividing the first row by w_{02} and the second row by w_{12}, and defining $\alpha = w_{01}/w_{02}$ and $\beta = w_{02}/w_{12}$, we obtain

$$W = \begin{bmatrix} 1+\alpha & 1 \\ \beta & 1+\beta \end{bmatrix}, \qquad S = \begin{pmatrix} \alpha s_{01}+s_{02} \\ s_{12}+\beta s_{02} \end{pmatrix} \quad (15)$$

and

$$W^{-1} = \frac{1}{1+\alpha+\alpha\beta} \begin{bmatrix} 1+\beta & -1 \\ -\beta & 1+\alpha \end{bmatrix}. \quad (16)$$

Then, since $\beta \ll 1$ and $\alpha \sim 1$, the condition number becomes

$$\kappa = \frac{(1+\alpha)^2}{1+\alpha+\alpha\beta} \sim 1, \quad (17)$$

and the numerical accuracy of the solution approaches the machine precision **eps**.

The discussion below Eq. (13) does not prove that the solution of Eq. (10) will necessarily have large errors, because Eqs. (13) give only an estimate of the upper bound of the errors. For some matrices, the actual errors may be small despite the large condition number κ. However, qualitatively one can easily see that in Eq. (10) the large quantity w_{12} wipes out the information about the small quantities w_{01} and w_{02}, while this does not happen in Eq. (14).

The method of variable substitution described above can be generalized for an arbitrarily large system of equations. We start with substituting some of the unknown energy levels E_i in Eq. (1) with their separations x_i from some of the other levels E_{si}:

$$E_i = x_i + \sum_{k_i=1}^{k_i^{\max}} x_{sk_i}; \qquad x_{sk_i^{\max}} = E_{sk_i^{\max}};$$

$$x_{sk_i} \neq 0, \quad k_j \neq i \quad \text{for } i \in Subst \equiv (i_{s1}, i_{s2}, \ldots, i_{sn_s}),\ n_s < N;$$

$$E_i = x_i, \quad k_i^{\max} = 0 \quad \text{for } i \notin Subst; \quad (18)$$

$$F = \sum_{i,j} \left(x_j + \sum_{k_j=1}^{k_j^{\max}} x_{sk_j} - x_i - \sum_{k_i=1}^{k_i^{\max}} x_{sk_i} - s_{ij} \right)^2 w_{ij}. \quad (19)$$

The substitutions should be done in such a way that the terms with the largest weights w_{ij} in Eq. (19) have $x_{sk_j} = x_i$ or $x_{sk_i} = x_j$, respectively, which then cancel out. Note that the substitution levels x_{sk_i} may themselves be substituted; hence the expression (18) for E_i contains a sum over the index k_i of the substitution levels. For all i, the level corresponding to the last term in this sum is not substituted.

When choosing the levels to be substituted, the basic idea is that, if there is a group of levels connected by very precisely measured transitions, one of these levels is selected as the main one, and for the rest of the group the level energies are substituted by sums of the energy of the main level and the energies of the connecting transitions. The strategy we adopted for choosing the substitution levels consists (in a somewhat simplified form) of the following:

1) For each unknown level E_i in the system, cycle through all observed transitions originating from the level E_i or terminating on it and find the minimum and maximum weights $w_i^{\min}$ and $w_i^{\max}$ of these observed transitions.
   For each of these transitions ($E_i \to E_j$ or $E_j \to E_i$), do the following:
2) If the ratio $r_i = w_i^{\max}/w_i^{\min}$ is smaller than a threshold value $r_{\text{thresh}} = 10^5$, or if the connecting level E_j of the transition having weight $w_i^{\max}$ is the ground level or a virtually fixed level (see Section 2.5) skip the level E_i (do not substitute it).
   Otherwise, if E_i has not been substituted or if its substitution level has been set using a transition with a lower weight than w_{ij}, set the substitution level for E_i to E_j.
   Then cycle through all transitions $E_i \to E_j$ in which E_i is the lower level and compare their weights w_{ij} with $w_i^{\min}$. If $w_{ij}/w_i^{\min} > r_{\text{thresh}}$, and the substitution level for E_j either has not already been set or has been set to a connecting level with a smaller transition weight, set the substitution level for E_j to E_i.

The next step is to build the system of equations similar to Eq. (2) by differentiating Eq. (19) over the new variables x_i. It is difficult to write explicit formulas similar to Eqs. (3)–(4) for this new system of equations. However, algorithmically it is not too difficult to build the elements of the matrix W_{ij} and the right-hand side vector S by cycling through all observed transitions $x_i \to x_j$ and adding or subtracting relevant terms to the corresponding variables. In a somewhat simplified form, the algorithm is as follows:

1) Initialize all elements of W and S to zero values.
2) For each observed transition $E_i \to E_j$ having observed wavenumber s_{ij} and weight w_{ij}, do the following:
   2.1) Build a list L^+ of variables x_{j2} that contribute with a plus sign to the term with weight w_{ij} in Eq. (19). This list includes x_j and a recursively built list of substitution levels for it.
   2.2) Build a list L^- of variables x_{i1} that contribute with a minus sign to the term with weight w_{ij} in Eq. (19). This list includes x_i and a recursively built list of substitution levels for it.
   2.3) Delete from L^+ and L^- all variables that are common in both lists. This emulates the cancellation of variables in Eq. (19).
   2.4) For all levels x_{j1} in the list L^+, if x_{j1} is not the ground level, do the following:
      2.4.1) Add w_{ij} to the diagonal element $W_{j_1 j_1}$ and $s_{ij}w_{ij}$ to S_{j_1}.
      2.4.2) For all other levels with indexes $j_m > j_1$ in the list L^+, if x_{j_m} is not the ground level, add w_{ij} to the off-diagonal elements $W_{j_1 j_m}$ and $W_{j_m j_1}$.
      2.4.3) For all levels x_{i_1} in the list L^-, if x_{i_1} is not the ground level,
         - Subtract w_{ij} from the off-diagonal elements $W_{i_1 j_1}$ and $W_{j_1 i_1}$.
         - If step 2.4.1 has not been executed at least once:
           * Add w_{ij} to the diagonal element $W_{i_1 i_1}$ and subtract $s_{ij}w_{ij}$ from S_{i_1}.
           * For all other levels with indexes $i_m > i_1$ in the list L^-, if x_{i_m} is not the ground level, add w_{ij} to the off-diagonal elements $W_{i_1 i_m}$ and $W_{i_m i_1}$.
   2.5) If step 2.4.1 has not been executed at least once, execute step 2.4.3.

This algorithm was tested on the case of level optimization for hydrogen [13] and was found to effectively eliminate numerical

errors in the solution. For the hydrogen spectrum, where some transitions were measured with unprecedented small uncertainties of 3×10^{-5} MHz, while for other transitions the measurement uncertainties were as large as 1.2×10^5 MHz, this method reduces the matrix condition number κ from 4×10^6 to 1.6×10^4 and the maximum componentwise numerical error (see further sections) in the solution for energy levels from 0.6 MHz to 0.002 MHz. The numerical error δE_{num} in each of the energy levels becomes significantly lower than the smallest measurement uncertainty of any connected transition d_{min}, while without the variable substitution the ratio $\delta E_{\mathrm{num}}/d_{\mathrm{min}}$ exceeded a factor of 10 for some of the levels.

We note that the matrix W built using the variable substitution method is different from the one built using Eqs. (3)–(5). The latter matrix is necessary for the determination of the uncertainties of Ritz wavenumbers (see Eq. (9)). Therefore, after finding the solution for the energy levels using the variable substitution and solving Eqs. (6)–(8), the program LOPT makes an additional step involving matrix inversion for the determination of the uncertainties of the Ritz wavenumbers. In this second step, the matrix is built without variable substitution. For this step, the influence of numerical errors is always insignificant, because the accuracy needed for the determination of the uncertainties is much lower than that for the energy values.

2.4. Inverting the matrix and finding the solution of the matrix equation

Inversion of the matrix W and solving the matrix equation (2) can be accomplished using several different methods. The routines for these procedures are included in the separate program module InvertMatrix.pl. One of the methods used is the *LU* decomposition with row pivoting and matrix scaling, and the other one uses the Cholesky decomposition. Although there exist other methods suitable for the task, such as *QR* and singular value decompositions [8,11,12], which have similar or better numerical properties, the methods we chose are much faster and proved to provide sufficiently precise results for the type of matrices arising in atomic spectroscopy problems. As noted in the textbooks quoted above, row pivoting is essential for achieving numerical stability with the *LU* decomposition. The necessity for matrix scaling to minimize the matrix condition number is discussed in detail by Higham [12] in Chapter 7.3. He showed that, if a matrix A is scaled by left-multiplying it by a diagonal matrix D_R whose diagonal elements are reciprocals of row norms of A ($A_{sc} = D_R A$, $D_{Rii} = \|A_i\|^{-1}$),[1] the condition number of A_{sc} is close to a minimum for a set of all possible row scaling matrices D_R. This type of scaling proved to drastically decrease the condition number for the matrices W formed by Eqs. (3)–(5) or Eq. (19) with variable substitution. Column scaling, also discussed by Higham [12], did not decrease the matrix condition number, nor did it decrease the values of the residuals (see below) for the type of problems solved here. Therefore, in the program LOPT we use only the row scaling. The rows of W are scaled by factors equal to reciprocals of row norms of W; then the scaled matrix W_{sc} is inverted, after which W^{-1} is rescaled back by multiplying its columns with the stored row norms of W.

As discussed by Demmel [8] in Chapter 2.5 and by several other authors, the solution of the linear system (2) obtained with the *LU* decomposition can be improved by applying an iterative correction using Newton's method. If a numerical solution of the system $Ax = b$ yields the solution vector $\dot{x}_0$, it can be improved (made closer to x) by applying the following iterative procedure:

1) Find the residual vector $r = A\dot{x}_i - b$.
2) Solve $Ad = r$ for d (similar to Eqs. (7), (8), first solve $Ly = r$ for y, then solve $Ud = y$ for d).
3) Find the next iteration of the solution: $\dot{x}_{i+1} = \dot{x}_i - d$.

As recommended by Moler [14], Golub and Van Loan [11], and other authors, a significant improvement in the accuracy of the solution can be achieved if step 1 of this procedure is executed with twice as high numerical precision as all the other steps (including the matrix inversion). However, this recipe turned out to be inapplicable in our case, because the versions of Perl interpreters currently available for most platforms are built so that they support only double precision floating point numbers stored in 64 bits. Doubling the precision would imply 128-bit floating point numbers, which are not currently available in Perl even for 64-bit operating systems. Since one of our main objectives is to make the program portable to any platform, we have to use the same (double) precision in all steps.

The convergence properties of the iterative improvement process performed with the same precision in all steps were investigated by many authors, and a brief summary is given by Demmel [8] in Chapter 2.5.1. His Theorem 2.8 states that, if r is computed in the same precision as in all other steps of the solution, and

$$\left\|A^{-1}\right\| \cdot \|A\| \cdot \frac{\max_i(|A| \cdot |x|)_i}{\min_i(|A| \cdot |x|)_i} \cdot \mathbf{eps} < 1, \tag{20}$$

then one step of iterative refinement yields $\dot{x}_1$ for which the componentwise relative backward error is as small as possible, of the order of **eps**. Further iterations would not improve the solution.

In numerical experiments we verified that the condition in Eq. (20) is satisfied in all cases (provided that the variable substitution is used). Therefore, the iterative improvement with one iteration is always done in LOPT if the *LU* decomposition is chosen. It indeed reduces the values of the residuals d, in many cases by an order of magnitude or more.

The matrix equation (2) is a system of normal equations for the least-squares problem (1). It was shown by Seber [10] that the matrix of such a system is always symmetric positive definite (provided that it is non-singular; this condition is always true in our case, which is ensured by special procedures described in the further sections). Therefore, the solution can be found by using the Cholesky decomposition $W = CC^T$, where C is lower triangular. This method is numerically much more stable than the *LU* decomposition (see, e.g., Demmel [8] or Higham [12]). Therefore, by default the program LOPT uses this method to solve Eq. (2). In this case, Eqs. (7), (8) are modified so that C is used instead of L, and C^T instead of U. There are several different implementations of the Cholesky method. The one used in LOPT is adapted from the JAMA package [15] and optimized for Perl. This implementation corresponds to the so-called "sdot" form (see Algorithm 10.2 in Higham [12]). Higham [12] in Chapter 10.3 also describes another form with an outer product and complete pivoting, which is numerically even more stable. However, it is computationally more expensive. For all tested atomic-physics problems the stability of the "sdot" algorithm proved to be quite sufficient (provided that the variable substitution is used to form the matrix). The Cholesky decomposition should nominally be faster than the *LU* decomposition. This was, indeed, confirmed for our Perl implementation. However, in this implementation the *LU* decomposition turned out to work faster for finding the matrix inverse.

The matrix inversion procedure is the most computationally intensive part of the program. It requires $O(N^3)$ floating point operations and can take significant time for large matrices. For the vast majority of atomic spectroscopy problems that we encountered so far the number of levels N was less than 400 and required

[1] Throughout this article we use symbol $\|\bullet\|$ to designate the row norm $\|\bullet\|_\infty$.

an execution time of less than one minute on a typical PC. However, for very complex spectra such as Fe II the number of levels can be as large as 1000, which leads to execution times of the order of 10 minutes. Although still acceptable, this may be inconvenient, especially taking into account that for each spectrum the level optimization usually needs to be performed many times with adjustments of transition assignments, weights, and other parameters between runs. If large problems have to be solved often, the task of inverting the matrix can be delegated to a separate program written in a more efficient language such as Java or Fortran. In this case the module InvertMatrix.pl should be modified so that it would execute this external program instead of doing the matrix inversion in Perl. Data exchange can be easily arranged through temporary files.

2.5. Virtually fixed levels

The formula (9) was derived under the assumption that there is only one fixed level (the ground state) in the level system. Observed atomic spectra are often divided into two or more independent sub-systems of levels. For example, in the case of the Ne III spectrum [7], the quintet levels are not connected with triplet and singlet levels by any observed transition. In such cases, the problem of finding the energy levels is usually solved by fixing some additional levels in order to eliminate degeneracy of the equation system (2). Such fixing decreases the summation range in the left-hand side of Eq. (2) and modifies its right-hand side. As a consequence, although the levels can still be found from Eqs. (5)–(8), the rigorous derivation that led to Eq. (9) cannot be followed in the same way. Thus, Eq. (9) becomes inapplicable. Besides that, uncertainties of predicted intersystem lines cannot be derived with this approach, since the fixed levels are assumed to have zero uncertainties.

To avoid the problems described above, a concept of *virtually fixed levels* was introduced in the LOPT code. Instead of ultimately fixing the level, a *virtual transition* is added to the initial set of spectral lines. This virtual transition connects the level to be fixed with the ground state (or with another chosen level). The wavenumber of this transition is assumed to have a finite uncertainty. The problem of level optimization is then solved in two stages. In the first stage, the optimized level values are found, assuming the uncertainties of the virtually fixed levels to be very small (ten times smaller than the smallest of the measurement uncertainties of the observed lines). This effectively fixes these levels in the solution of the equation system (2) (or an equivalent equation system resulting from Eq. (19) if the variable substitution is used). In the second stage, the equation system (2) is re-built assuming some realistic (user-defined) uncertainties of the virtually fixed levels, and the uncertainties of predicted wavenumbers are found from the matrix of this modified equation system. In this stage, Eqs. (5)–(8) are skipped, as there is no need to find the level values; only the elements of the inverse matrix W^{-1} are calculated, and then the uncertainties a_{ij} are determined using Eq. (9).

It should be noted that the matrix W is non-singular if for any excited level E_i $(i > 0)$ there exists a sequence of transitions to lower levels ultimately connecting this level to the ground state E_0. The program LOPT verifies that there is indeed such a connection. If for any level such connection is missing, this level is automatically "fixed" at zero value using the approach of virtually fixed levels described above. In the output file, such levels are marked as "fixed by program". This fixing ensures non-singularity of the matrix equation (2).

2.6. Weights of components of unresolved blends

Another feature of the LOPT code is a possibility to correctly derive the level values associated with unresolved blends of several transitions. This is done by multiplication of the weights of the components of an unresolved blend B by factors proportional to the calculated intensities:

$$w_{ij}^B = \left(d^B\right)^{-2} I_{ij}^B / I_{tot}^B, \tag{21}$$

where I_{tot}^B is the sum of the intensities of all components of the blend. This is equivalent to dividing the measurement uncertainty of the center of gravity of the blend d^B by the square root of the relative intensity factors.

Our numerical experiments showed that Eq. (21) results in correct optimized level values. Some limitations of its applicability were discussed by Kramida and Nave [7].

The relative intensities of the blend components can usually be estimated using some approximate model. Errors of such approximations may affect the optimized level values. Therefore, even if the center of gravity of the blend is measured with a very small uncertainty, the uncertainties d^B assigned to the components of the blend in the equation system (2) should be assumed large enough to encompass the uncertainties of the model.

2.7. Treatment of systematic shifts

As was already pointed out, it is not possible to rigorously account for systematic errors in the measured wavelengths. Nevertheless, some estimate of the possible effects of such shifts can be found. The approach implemented in the LOPT program is the following.

In the input set of observed wavelengths, the lines must be divided into several groups. Within each group K, all lines are supposed to be mutually correlated, i.e. it is assumed that all measured wavenumbers s_{ijK} of transitions between levels E_i and E_j in the group K are affected by the same value of the systematic shift δ_K:

$$s_{ijK} = s_{ijK}^* + \delta_K, \tag{22}$$

where s_{ijK}^* represents the unshifted wave number.

The values of shifts δ_K for different groups are assumed to be independent of each other.

The smallest uncertainty of the measured wavenumber within the group is adopted as an estimate of the possible systematic shift δ_K. The effect of this shift on the energy level E_j resulting from the solution of the least-squares problem can be estimated by differentiating Eq. (5) with respect to the group-shift variable δ_K:

$$\delta E_{jK} = \delta_K \sum_{i,m} W_{ij}^{-1} w_{imK}, \tag{23}$$

and the total shift of E_j is

$$\delta E_j = \sum_K \delta E_{jK}. \tag{24}$$

Thus, apart from the standard dispersion of the calculated energy E_j, we have an additional possible error δE_j due to the systematic shifts of correlated lines. The possible shift of the Ritz wavenumber of a transition between E_i and E_j is simply the difference $\delta E_i - \delta E_j$. This possible shift should be combined with the dispersion a_{ij} (see Eq. (9)) in order to obtain a more confident estimate of the possible error:

$$b_{ij}^2 = a_{ij}^2 + \sum_K (\delta E_{iK} - \delta E_{jK})^2, \tag{25}$$

where b_{ij} is the total estimated uncertainty of the Ritz wavenumber $E_i - E_j$, and a_{ij} is defined by Eq. (9).

The absolute uncertainty of the energy level E_i is, by definition, the uncertainty of the wavenumber of the transition from this level to the ground state:

$$D_i \equiv b_{i0} = \left(W_{ii}^{-1} + \sum_K \delta E_{iK}^2 \right)^{1/2}. \tag{26}$$

2.8. Treatment of rounding errors

The term "rounding errors" is applied to two different types of rounding. The first type is the rounding intrinsic to digital arithmetical operations. The real numbers, which generally can be represented by floating-point numbers with an infinite number of digits (or by infinite sums of decreasing powers of 2), are approximated by discrete values stored in finite number of bits. The second type of rounding occurs when a floating-point number calculated with a very high (computer-limited) digital precision is rounded off in order to make its representation consistent with its actual uncertainty (determined by measurement uncertainties of the input quantities). Both types of rounding introduce additional errors in the computed quantities. They are discussed below.

2.8.1. Rounding errors due to finite precision of digital arithmetic

This type of error occurs because the initial system of equations (2) is solved not exactly but with approximate values of each floating point number involved in the procedure. Digital representation of the floating point numbers has a relative uncertainty of $\pm\mathbf{eps}/2$, where $\mathbf{eps} \approx 2.2 \times 10^{-16}$ in the case of 64-bit double precision representation. Numerical solution of the equation system (2) can introduce significant computational errors if the matrix is ill-formed. The magnitude of these errors is rather difficult to estimate analytically. To estimate these numerical errors, we use a method of random trials. We introduce an artificial random numerical noise $\delta s_{ij} = \pm 1.1\,\mathbf{eps}\, s_{ij}$ and $\delta d_{ij} = \pm 1.1\,\mathbf{eps}\, d_{ij}$ in the initial values of the observed transition wavenumbers s_{ij} and their uncertainties d_{ij}, which leads to small random modifications δS_i and δW_{ij} of the values of the right-hand side vector S_i and of the matrix elements W_{ij} computed with Eqs. (3), (4). The coefficient 1.1 is introduced in the size of the random noise in order to ensure that the actual modification of s_{ij} and d_{ij} is not zero due to numerical rounding. Then we find the corresponding modified solution $E_i + \delta E_i$ by solving the same Eqs. (7), (8) and find the deviations from the initial solution δE_i. These random tests are repeated several times, and the sums of squares of δE_i are accumulated. Then the root-of-mean-square (rms) estimates of errors δE_i^{rms} are easy to compute. In all tested cases, ten random trials proved to be sufficient to obtain stable and sufficiently accurate estimates of δE_i^{rms}. These estimates are compared for each E_i with the corresponding minimum uncertainty value $D_i^{\min} = \min(D_{1i}, D_{2i}, D_{3i})$ (see Section 2.9). If any of δE_i^{rms} is found to exceed half of $D_i^{\min}$, then the solution is deemed to suffer from unacceptably large numerical errors, and the corresponding diagnostics are printed to the output screen.

The solution employing the Cholesky decomposition method and variable substitution proved to yield negligibly small numerical errors even in the most sensitive case of the hydrogen spectrum. Therefore, it is chosen as the default method of solution. The *LU* decomposition with row scaling and one-step iterative improvement yielded a somewhat greater but still negligibly small estimated maximum numerical error, and the numerical values of the resulting energy levels were identical to those obtained with the Cholesky decomposition method. If other methods are used, such as the original Radziemski method of the matrix construction (without the variable substitution) or *LU* decomposition without row scaling, the numerical errors turned out to be unacceptably large. They were successfully detected by the procedure described above, and the resulting values of the energy levels were indeed different from the most accurate solution, with deviations far exceeding the estimated uncertainties $D_i^{\min}$ for many of the levels.

As mentioned in Section 2.3, the case of the hydrogen spectrum, where the ratio of the maximum and minimum wavenumber uncertainties exceeds 4×10^9, is rather exotic. In most experimental cases, this ratio does not exceed 10^5, and the numerical errors of the solution are negligibly small. For this reason, and because the random trials method requires a substantial time, by default the LOPT code does not estimate the numerical errors. This estimation can be turned on by a corresponding setting in the input parameter file. Then the values of the estimated numerical errors are printed in an additional column in the levels output file. If these errors are determined to be significant, a corresponding warning message is displayed, and relevant diagnostics are printed to the output screen. The diagnostics include the maximum ratio $\delta E_i^{rms}/D_i^{\min}$, machine precision **eps**, and normwise relative error bound. If this happens, then the program should be executed on a computer having a more precise representation of floating-point numbers (for example, 128 bits instead of 64 bits). This should effectively eliminate machine-limited rounding errors. Another possible solution is to implement a more numerically stable algorithm such as the outer-product Cholesky decomposition with complete pivoting [12].

2.8.2. Errors due to proper rounding of values according to their uncertainties

The problem of proper rounding of measurement results was already addressed in a vast number of publications. It has two parts. The first part is a purely mathematical problem of rounding a floating-point decimal number to a given number of significant figures. This problem is discussed, for example, by Higham [12]. Its most important aspect is the symmetry of the statistical distribution of the results of rounding relative to the initial (unrounded) randomly selected numbers. In our case, all computations are made using double precision floating point arithmetic, and the output values need to be properly rounded according to their estimated uncertainties. The relative uncertainties of the output values are always much greater than the machine precision $\mathbf{eps} \approx 2.2 \times 10^{-16}$. Therefore, the influence of asymmetry of the floating-point rounding errors on the solution is negligible, and we use the simple "round-half-up" method implemented in the Perl's sprintf( ) function [16].

The second part of the problem is finding a proper number of significant figures to represent a result of measurement or computation having a known statistical uncertainty. This problem was also widely discussed in the literature. However, there is no strictly defined standard solution. One of the best solutions is described in an on-line article by Lindberg [17]. According to his recommendations, if the leading figure in the uncertainty is a one, we use two significant figures in the uncertainty value; otherwise we use one significant figure. Then the reported value is rounded off to the same number of places after the decimal point as the uncertainty value.

A similar approach is implemented by the so-called "rule of 20", which specifies that the least significant digit of the value should be dropped if the uncertainty, expressed in the units of that least significant digit, is greater than 20. Sometimes a "rule of 24" or a "rule of 15" is used in place of the "rule of 20". In general, all similar rounding rules may be called "rule-of-m rounding" with the parameter m characterizing a particular rule. Farther below, we call the quantity m characterizing the rule-of-m rounding the *round-off threshold*.

The ISO 1995 standard rule for rounding [18] is formulated as follows: The standard uncertainty s of a value x always ought to be rounded to two significant digits. The value x ought to be rounded to the same order as is the standard uncertainty s. This means, for example, that, if the quantity x is measured with a standard uncertainty of ± 0.99, then the value of x ought to be rounded to two places after the decimal point. This standard is equivalent to the rule-of-99.4999... rounding. Despite the existence of this standard, most scientific publications do not observe it, because it is only a recommendation rather than a strict rule, and in many cases it requires one too many significant figures than is intuitively felt adequate.

An important aspect of the rounding problem, which is not well understood or explained in the literature, is the influence of the rounding errors on statistical properties of measured or calculated quantities. The most comprehensive treatment of this aspect of rounding, to our knowledge, was reported by Wimmer et al. [19]. These authors introduced a concept of *ε-proper rounding*, in which the quantity ε denotes the accepted tolerance for the confidence level value. In Appendix A we describe a practical implementation of this concept and use it to obtain an estimate of applicability of the rule-of-m rounding in representation of scientific data, depending on the parameter m. The basic conclusions are as follows:

1) Rule-of-m rounding with $m < 10$ (in which the value of uncertainty is always rounded to one significant decimal figure) should never be used to represent scientific data, since it distorts the statistical distribution of data to an unacceptably large degree.
2) Rule-of-m rounding with $m = 20$ to 25 is quite acceptable for representing the data in the cases where statistics of outliers with deviations exceeding 2σ is not important. This is the case for most atomic-spectroscopy problems.
3) Rule-of-m rounding with $m \geqslant 70$, as well as the ISO 1995 standard [18] recommended rule for rounding, preserves most of the statistical properties of the data, including the probability distribution of outliers with deviations up to 3σ. The ISO 1995 standard rounding should be used in the cases where statistical distribution of strongly deviating outliers is significant.

The program LOPT implements the rule-of-m rounding and allows for a user-defined value of the round-off threshold m. Allowed values of m are between 10 and 99.9999.

When a quantity is measured or calculated with a known standard uncertainty, its value is often obtained with an excessive precision and needs to be properly rounded. In such case, for each rounded value, the original, more precise value is known and can be directly compared with the rounded value. This comparison yields the value of the rounding error. In general, this rounding error increases the variance of the rounded values compared to the original, unrounded ones. Therefore, the rounding errors contribute to the uncertainties of the rounded values. The greater the rounding error, the greater is the increase in the uncertainty. In Appendix B we analyze this dependence in detail and show that it can be approximated by a sum in quadrature. The program LOPT takes into account this increase of the uncertainties of the reported values due to rounding.

2.9. Presentation of level uncertainties

The concept of "absolute" level uncertainty is quite convenient for the cases where only the excitation energy is used, e.g., in formulas for ground-state-absorption oscillator strengths or cross sections of processes involving the ground state and one excited state. However, it is inadequate for estimation of uncertainties of transition wavelengths or any other quantities involving the energy differences between excited states. This is due to the fact that the optimized level values are not statistically independent even if the measured wavelengths were such. This becomes obvious if we consider a simple three-level system ($E_1 = 0$, $E_2, E_3 > 0$) connected by three imaginary "observed" lines ($\lambda_1, \lambda_2, \lambda_3$). Let us assume that the two excited levels have a very small relative separation compared to their excitation energy. If the long-wavelength transition λ_3 between the two excited levels is measured with the same relative uncertainty $d = \Delta\lambda/\lambda$ as the short-wavelength transitions λ_1 and λ_2, then the "absolute" uncertainty of E_2 and E_3 will be much greater than the uncertainty of their separation:

$$\delta(E_2) = E_2 d, \tag{27}$$

$$\delta(E_3) = E_3 d, \tag{28}$$

$$\delta(E_3 - E_2) = |E_3 - E_2| d, \qquad |E_3 - E_2| \ll E_2, E_3. \tag{29}$$

Thus, in a general case, for adequate treatment of energy-interval uncertainties, one needs to know the entire set of individual uncertainty values b_{ij} (see Eqs. (9) and (25)) for each energy difference $E_i - E_j$. Such data can hardly be made available because they would take too much storage space while most of them will never be needed. In the LOPT program, a compromise solution is chosen: the program calculates the uncertainty values only for those energy differences that correspond to observed or predicted transitions included in the input file, and also three sets of uncertainty estimates for individual level values.

These three different values of level uncertainty, denoted as D_1, D_2 and D_3 in the levels output file, have the following meaning:

D_1 is close to the minimum estimated dispersion relative to any other term. It is derived by a method similar to the one suggested by Radziemski and Kaufman [20]:

$$D_{1i} = \frac{(\sum w_{ij}^2 [d_{ij}^2 + (s_{ij} - E_i + E_j)^2])^{1/2}}{\sum w_{ij}}, \tag{30}$$

which is equivalent to

$$D_{1i} = \frac{(\sum d_{ij}^{-2} [1 + d_{ij}^{-2} (s_{ij} - E_i + E_j)^2])^{1/2}}{\sum d_{ij}^{-2}}, \tag{31}$$

where summation goes over all observed lines. The value of D_{1i} determines the number of significant figures given in the output energy levels list for the energy E_i.

D_2 is the uncertainty of the level value relative to the ground level, as defined by Eq. (26).

D_3 is the estimated uncertainty relative to the lowest fixed level f in the isolated level system (here, "isolated" means "isolated from the ground level"), determined in the same way as D_2:

$$D_{3i} \equiv b_{if} = \left(a_{if}^2 + \sum_K \delta E_{iK}^2 \right)^{1/2}. \tag{32}$$

In addition to these three uncertainty values, it is possible to include in the output level list uncertainties of level separations from any chosen levels. We call such reference levels *base levels*. Uncertainties relative to base levels are calculated according to Eq. (23).

2.10. Calculation of RSS

The value of the residual sum of squares, *RSS*, is calculated from the following formula:

$$RSS = \sum \left(\frac{s_{ij} - E_i + E_j}{d_{ij}} \right)^2. \tag{33}$$

As explained in the Introduction, the value of *RSS* provides a statistical measure of how reasonable the stated uncertainties of the

```
h1_lin3.txt          ; Transitions input file name
h1_fix2.lev          ; Fixed levels input file name
H1_opt3.lev          ; Levels output file name
H1_opt3.lin          ; Transitions output file name
Y                    ; OMIT calc. wavelengths in ANGSTROMS in the output (Y/N)?
N                    ; TUNE single-line levels (Y/N)?
N                    ; Print listing of correlated-lines uncert. effects (Y/N)?
Y       ; Calculate predicted line uncertainties (Y/N)?
Y       ; Use Cholesky decomposition (Y/N)?
Y       ; Use variable substitution (Y/N)?
10      ; Number of trials for estimating numerical instability effects (0-100)
Y       ; Write virtual lines (Y/N)?
N       ; Divide levels into independent groups?
0       ; Min. wavenumber for air wavelength (0 for no conversion)
0       ; Max. wavenumber for air wavelength (0 for no conversion)
24.5    ; Round-off threshold for output levels
24.5    ; Round-off threshold for output lines
   69   ; first column of line intensity in the transitions-input file
   70   ; last column of line intensity in the transitions-input file
19      ; first column of wavelength in the transitions-input file
38      ; last column of wavelength in the transitions-input file
48      ; 1st column of wavelength UNITS in the transitions-input file (width=4 chars)
  39    ; first column of wavelength uncertainty in the transitions-input file
  47    ; last column of wavelength uncertainty in the transitions-input file
   53   ; first column of line weight in the transitions-input file
   56   ; last column of line weight in the transitions-input file
1       ; first column of lower level label in the transitions-input file
7       ; last column of lower level label in the transitions-input file
10      ; first column of upper level label in the transitions-input file
63 ; first column for line flags (B=blend,Q=quest.,M=masked,P=predicted,A=air,V=vacuum)
Y       ;tab-delimited output [Y/N]?
```

Fig. 2. Sample parameter file for the hydrogen spectrum. Some parameters are mandatory (e.g., file names), others may be omitted (see details in the User's Guide).

input wavenumbers are. If the measured wavenumbers have normal statistical distribution (with no systematic shifts), and the given measurement uncertainties for all observed transitions correspond to statistical standard deviations, then *RSS* is equal to χ^2 and should be close to the number of degrees of freedom of the problem, i.e.

$$N_{df} = M - N, \tag{34}$$

where M is the total number of observed transitions, and N is the total number of levels involving observed transitions (excluding the ground level).

2.11. Description of the program modules

The program consists of three separate files, Lopt.pl, InvertMatrix.pl, and vacair.pl. The main program is Lopt.pl. The functions for the *LU* and Cholesky decompositions, the corresponding solutions of the matrix equation, and for the matrix inversion are separated in the file InvertMatrix.pl. If a particular application frequently involves tasks with more than 500 energy levels, this Perl implementation of these functions can become prohibitively slow. In such cases, the module InvertMatrix.pl should be replaced by another one that would delegate the task of matrix inversion to a separate program written in a more computationally efficient programming language. For example, a similar code written in Java [15] works approximately 10 times faster. The only requirement for the replacement module is that it should provide the above-mentioned functions with the same interfaces.

The module vacair.pl contains functions related to conversion of air wavelengths to vacuum and vice versa. These functions (Lair( ) and Lvac( )) can be used in other atomic spectroscopy problems and therefore are separated from the task-specific code.

2.12. Description of the input files

The program requires three input files: (1) the parameter file, (2) the transition input file, and (3) the fixed-levels input file. Detailed description of the contents and formats of these files is given in the User's Guide included with the program package. Here we give only a brief description.

The parameter file governs the execution flow of the program. All different modes of the program action, as well as main input/output file names and formats, are specified here. The contents of a sample file for the hydrogen spectrum are given in Fig. 2.

The transition input file is the main input file. It contains the following data:

1) W (wavelengths, wavenumbers, transition frequencies, or energy intervals);
2) Measurement units for W (optional);
3) Measurement uncertainties for W (mandatory for the first line);
4) Lower and upper level designations for each transition (mandatory);
5) Weight of the transition in level optimization (only for multiply-assigned lines; optional);
6) Flag specifying a special status of the transition (optional);
7) Line intensity and/or any other auxiliary information (optional).

Two important features of the input file should be mentioned. First, each line in the file must correspond to a unique transition. If multiple measurements of the same transition are available, they must be properly averaged before inserting the line in the input file. This is required by the algorithm of line counting and by the algorithm of assigning weights to blended transitions. Second, the lower and upper levels for each transition must be given in

```
1S  1/2  2S  1/2  2466061413.187074   0.000034 MHz
1S  1/2  2P  1/2  2466067546          4057            1
1S  1/2  2P  3/2  2466067546          4057            2
1S  1/2  3P  1/2  2922728618          8548            1
1S  1/2  3S  1/2  2922743277.97       0.22                    P
1S  1/2  3P  3/2  2922728618          8548            2
1S  1/2  4P  1/2  3082572007          63392           1
1S  1/2  4S  1/2  3082581563.823      0.010                   P
1S  1/2  4P  3/2  3082572007          63392           2
1S  1/2  5P  1/2  3156567341          13294           1
1S  1/2  5S  1/2  3156563685.1        1.2
1S  1/2  5P  3/2  3156567341          13294           2
1S  1/2  5D  5/2  3156564549.6        0.7
1S  1/2  6P  1/2  3196763254          68176           1
1S  1/2  6S  1/2  3196751430.284      0.021                   P
1S  1/2  6P  3/2  3196763254          68176           2
1S  1/2  7P  1/2  3220977255          69213           1
```

Fig. 3. Portion of a sample transition input file for the hydrogen spectrum. Essential parts of the input data: lower and upper level labels, the value of W, and its uncertainty.

```
1S  1/2  0              0
;2P  3/2  82259.2850014  999999[1]     ; Lines starting with a semicolon are ignored.
```

Fig. 4. Contents of a sample fixed-levels input file for the hydrogen spectrum. In this typical example, only the ground level label, its energy (zero), and uncertainty (zero) are specified.

the correct columns specified in the parameter file. Interchange of lower and upper levels for some of the transitions would destroy the correct order of levels determined by the graph of transitions.

A sample of the input file for the hydrogen spectrum is given in Fig. 3.

Note that the transitions marked with the flag "P" (meaning "predicted") are excluded from the level optimization. However, their calculated Ritz wavenumbers and their uncertainties will be included in the output. The weights are essential only for transitions having the same W value (in this example, two transitions with frequency 2 466 067 546 MHz). For other transitions they are ignored (assumed to be unity). Blank weights are interpreted as the default unity ("1.0").

The fixed-levels input file contains the definition of the ground level and other fixed levels (if any). The level designations must be the same as in the transition input file. For each fixed level, the energy value and its uncertainty (in cm^{-1}) must be specified. Some other properties of the fixed levels may also be specified, as described in the User's Guide. A sample fixed-levels file for the hydrogen spectrum is given in Fig. 4. Lines starting with a semicolon are comments.

Several sets of complete working input files are included in the Examples section of the package.

2.13. Description of the output

During the program execution, the following diagnostics are printed to the standard output:

1) Messages identifying each program step and execution time for the most time-consuming tasks such as calculation of the energy levels (solving the matrix equation) and calculating dispersions for Ritz wavelengths.
2) Number of levels in each isolated group.
3) The value of *RSS* and the number of degrees of freedom for each isolated group of levels.
4) Warnings and error messages, if any.

A sample screen output is presented in Fig. 5.

There are two main output files, one for levels and one for transitions. Detailed description of the contents and formats of these files is given in the User's Guide included with the program package.

The format of the levels output file is illustrated by the example given in Table 1. The file header defines the columns present in the output. They are as follows:

1) Level designation (the level label read from the transition input file).
2) Energy in units of cm^{-1}.
3) Three values of dispersion (uncertainty): D_1, D_2 and D_3 (see Section 2.9).
4) Estimate of the numerical uncertainty d due to floating-point computation errors (see Section 2.8.1). This column appears in the levels output file only if the number of random trials is set to a value greater than zero (see the parameter "Number of trials" in the parameter file sample given above).
5) Number of lines defining the level. The value starts with the total number of combinations significant for the optimization of the level value. After that, there may follow some numbers followed by letter symbols: L – number of lines with large deviation of the observed wavelength from the calculated one (greater than 1.2 times the standard deviation), D – number of doubly (or multiply) assigned lines, Q – number of questionable lines, B – number of blended lines.
6) Comments. The values can be "fixed by user" or "fixed by program" for the fixed levels, and blank for all other levels.

In addition to these columns, if required by specifications in the input file, there may be additional columns of uncertainties relative to base levels (see Section 2.9) and columns containing estimates of contributions of correlated groups of lines to the total uncertainty (see Section 2.7).

The structure of the transition output file is illustrated by an example in Table 2. The columns shown in this example contain the following quantities:

1) The input measured quantity W (in this case transition frequency).
2) Measurement uncertainty of W.
3) Single-line level flag (a star in this column indicates that one of the levels involved is determined solely by this transition).

```
C:\Work\LOPT\LOPT_IM>perl lopt.pl H1_LOP3.PAR
Reading parameters...Done.

Using Cholesky decomposition to solve the matrix equation.

Reading lines...Done.
Filling groups...Done.
Counting assigned lines...Done.
Reading fixed levels...Done.

Group 1 of 1: 77 levels.
Pass 1: finding energies...
Done. (spent 1 sec)
Number of substituted variables = 29
Pass 2: finding dispersions...
Done. (spent 0 sec)
Adjusting weights...Done.
Adjusting dispersions...Done.

RSS/degrees_of_freedom = 0.41 (41 degrees_of_freedom)

Finding Ritz uncertainties for lines...
Done. (spent 0 sec)

Rounding levels...Done.
Adjusting single-line levels...Done.
Rounding lines...Done.
Sorting levels...Done.
Writing levels...Done.
Writing lines...Done.
Ok.
Total time spent: 1 sec
```

Fig. 5. Sample screen output for the calculation of the hydrogen spectrum.

Table 1
Portion of a sample levels output file for the hydrogen spectrum.

Designation	Energy	D_1	D_2	D_3	d	N_lines	Comments
1S 1/2	0.0000000000	0.0000000011	0	–	0.0000000000	30, 1L, 22D	fixed by user
2P 1/2	82258.9191133	0.0000003	0.0000003	–	0.0000000	4, 1D	
2S 1/2	82258.9543992823	0.0000000011	0.0000000011	–	0.0000000000	11	
2P 3/2	82259.2850014	0.0000004	0.0000004	–	0.0000000	4, 1D	
3P 1/2	97492.2112001	0.0000008	0.000007	–	0.0000000	5, 1D	
3S 1/2	97492.2217014	0.0000011	0.000007	–	0.0000000	4, 1L	
3D 3/2	97492.319432	0.000018	0.000019	–	0.000000	5	

4) Ritz wavenumber (i.e. the wavenumber calculated as the difference between the optimized level values) in cm^{-1}.
5) Uncertainty of the Ritz wavenumber.
6) The "obs. − calc." difference (observed minus Ritz wavenumber) in cm^{-1}.
7) Lower level label.
8) Upper level label.
9) Lower level energy.
10) Upper level energy.
11) Transition flag ("P" signifies a predicted transition; see the next section).
12) Weight of the transition (differs from 1 only for blended or predicted lines; see Section 2.6 and the User's Guide).
13) Measurement units for W.

Several other columns have been omitted from this example for brevity. These include observed line intensity, Ritz wavelengths in air and in vacuum, uncertainties of the latter, etc.

2.14. Special features

It is often desirable to obtain predicted wavelengths (as well as their uncertainties) for unobserved transitions. With the LOPT program, this can be done easily. Those unobserved transitions can be included in the main transition input file along with the observed transitions, but they must be excluded from the level-optimization procedure. This is enforced by setting a corresponding flag ("M" for "masked" or "P" for "predicted") in the input file. For such transitions, the program finds the Ritz wavelengths and their uncertainties.

The questionable lines, i.e. observed lines with uncertain identification, should normally be excluded from the optimization procedure. Nevertheless, while the identification of the lines is not yet finalized, it may be worthwhile to experiment with different variants of line assignments of questionable lines, assigning a reduced weight for questionable lines. In such cases, weights of the questionable lines may be reduced by increasing their uncertainty values. The questionable lines should be marked by a flag "Q", so that the number of questionable lines could be counted for each level that involves such lines.

Sometimes there is a need to include in the transition list not only the measured wavelengths, but also level separations obtained by other means, e.g. measured ejected electron energies of Auger transitions. For such purposes, the program is able not only to understand different measurement units of wavelength, but also wavenumbers in different units. Therefore, the measurement units have to be specified in a separate column in the transition input file. If the unit is not specified, the program assumes that it is an

Table 2
Portion of a sample transition output file for the hydrogen spectrum.

W_obs	uncW_o	S	Wn_c	uncWnC	dEO-C	L_1	L_2	E_1	E_2	F	Weight	Unit
2466061413.18707	0.00003		82258.9543992823	0.0000000011	0.0000000000	1S 1/2	2S 1/2	0.0000000000	82258.9543992823		1.000	MHz
2466068000	7000		82258.9191133	0.0000003	0.24	1S 1/2	2P 1/2	0.0000000000	82258.9191133		0.333	MHz
2466068000	5000		82259.2850014	0.0000004	−0.13	1S 1/2	2P 3/2	0.0000000000	82259.2850014		0.667	MHz
2922729000	15000		97492.211200	0.000007	−0.5	1S 1/2	3P 1/2	0.0000000000	97492.2112001		0.333	MHz
–	–		97492.221701	0.000007	–	1S 1/2	3S 1/2	0.0000000000	97492.2217014	P	0.000	MHz
2922729000	10000		97492.319611	0.000007	−0.6	1S 1/2	3P 3/2	0.0000000000	97492.3196114		0.667	MHz
3082570000	110000		102823.8485825	0.0000005	0	1S 1/2	4P 1/2	0.0000000000	102823.8485825		0.333	MHz
–	–		102823.8530211	0.0000003	–	1S 1/2	4S 1/2	0.0000000000	102823.8530211	P	0.000	MHz
3220983655.4	0.7	*	107440.449866	0.000023	0.000000	1S 1/2	7D 5/2	0.0000000000	107440.449866		1.000	MHz

angstrom, until it reads a line where a different unit is specified. Then the default unit becomes the last one read from the transition input file.

The quantities actually used in the calculations are wavenumbers. Therefore, all wavelengths and/or energies are converted to wavenumbers before the optimization procedure. Conversion from air wavelengths to wavenumbers is made by using the five-parameter formula of Peck and Reeder [21]. Conversion from electron-volts to cm^{-1} utilizes the conversion factor $1.239841875 \times 10^{-4}$ eV/cm^{-1} recommended by CODATA 2006 [22].

2.15. A typical scenario of solving a level optimization problem

Solution of a typical level optimization problem usually involves the following steps:

1) Preliminary level optimization;
2) Exclusion or correction of outliers;
3) Weighting of multiply assigned lines;
4) Final level optimization.

In the first step, the three input files are constructed: the transition input file, the fixed levels file, and the parameter file. The most time-consuming part is the construction of the transition input file. At this stage, the fixed levels file usually contains just one level, the ground state. Running the LOPT program with the newly constructed parameter file usually reveals some errors, e.g., wrongly specified column numbers in the parameter file, typos in the level designations in the transition input file, or unconnected groups of levels. The latter problem arises in the cases when several excited levels are connected by some observed transitions with each other, but not with the ground level. In most cases, the program detects such unconnected level groups, fixes the lowest level in each group at zero energy, and marks it in the level output file with the "Fixed by program" comment. If such levels are present, they should be added to the fixed levels input file with a proper value of energy and uncertainty relative to the ground state. In some cases, it is necessary to include more complex specifications in the fixed levels input file, which involves fixing precisely known splittings between some of the levels (see the User's Guide for the recipe). In this stage of the process, it is best to use the option in the parameter file to produce the tab-delimited output files. Such files can be viewed with a spreadsheet program, which makes it easier to detect errors.

In the second step, the errors in the transition input file are detected and corrected. The level output file should be checked first for levels having large fraction of highly deviating connecting transitions (outliers). The numbers of connecting transitions and the numbers of outliers are given in the column "N_lines" of the level output file. The number of outliers, if any, is given after the total number of connecting transition with a letter code "L". If a level has a large number of outliers, all connecting transitions involving this level should be checked in the transition output file. After correction of obvious errors, the level optimization is repeated, and the transition output file should be checked for high "Obs.–Ritz" deviations. A good way of doing it is to insert a new column in this file and set the values in this column to absolute values of normalized deviations $\Delta = |W_{\mathrm{obs}} - W_{\mathrm{Ritz}}|/\delta W_{\mathrm{obs}}$, where W_{obs} is the value of the measured quantity (e.g., wavelength or wavenumber), W_{Ritz} is the Ritz value of this quantity, and δW_{obs} is the uncertainty of the measured value. Then it is easy to filter out only the highly deviating transitions (e.g., $\Delta > 2$). If the level-optimization problem is posed correctly, there should normally be only a few of such high deviations. Each of them should be carefully examined, as they are the prime candidates for errors in the line identification or measurement flaws. For example, the measured wavelength

may have been affected by blending; in this case the measurement uncertainty should be properly increased.

In the third step, if there are any multiply assigned lines (i.e., the same spectral line is assigned to more than one transition), each transition associated with them should be properly weighted. As explained in Section 2.6, the best way of weighting the multiply-assigned (blended) lines is to assign weights proportional to intensities of the unresolved components of the blend. In most cases, estimates of relative intensities can be obtained from calculations of transition rates or oscillator strengths. In some cases, more complicated (e.g., collisional-radiative) modeling may be necessary. Usually, the blended lines should be given with relatively large measurement uncertainties to decrease their effect on the derived energy levels.

At this stage of the process, if the set of measured transitions involves several sets of measurements of different nature (e.g., by different methods or in widely separated wavelength regions), the entire set of transitions can be divided into subsets of similar measurements, and the effects of possible systematic shifts in each subset can be investigated (see Section 2.7). If such systematic shifts are detected, it may be possible to remove them from the measured wavelengths. In addition, dividing the transitions list into subsets allows one to investigate the normalized deviations Δ (see above) for each subset, which may reveal errors in assigning measurement uncertainties for some of the subsets. Generally, the number of high values of Δ in each subset should be consistent with statistics for normally distributed measurements.

In the final step of the process, special attention should be paid to proper rounding of the output quantities. The round-off thresholds for levels and transitions (see Section 2.8.2) should normally be set to values between 20 and 25. The overall quality of the fitting is given by the ratio of RSS to degrees of freedom reported by the program. This ratio should be close to unity. Numerical errors due to limited machine precision can be investigated at this stage (see Section 2.8.1).

Most of the sample input files provided in the Examples in the program package correspond to the final stage of the level optimization. For W I and W II, the input files correspond to step 2 (no weighting of multiply assigned lines was made). The example for B III illustrates an extensive use of advanced options in the fixed-level input file.

3. Contents of the program package

The LOPT package consists of eight files:

Lopt.pl – the main program;
InvertMatrix.pl – the matrix inversion and related functions;
vacair.pl – functions for conversion of wavelengths from vacuum to air and vice versa;
lopt.par – sample parameter file;
test_lin.txt – sample transitions input file;
test.fix – sample fixed-levels file;
LOPT_Users_Guide.pdf – user's guide;
README – text file with general information about the program.

Several practical examples of level-optimization problems are given in the Examples directory. Each of these sets consists of three input files defining the problem.

Acknowledgements

The author is grateful to Prof. C. Froese Fischer and Dr. G.W. Stewart for useful comments and suggestions.

Appendix A. Implementation of the concept of ε-proper rounding and investigation of statistical properties of the rule-of-m rounding

The concept of *ε-proper rounding* was introduced by Wimmer et al. [19]. The quantity ε is a parameter of rounding. It denotes the accepted tolerance for the confidence level value. Its meaning can be understood from the following example.

If the measured value of a quantity x is $x_m \pm \Delta x$ with confidence level 0.95, and we want to represent x_m by a properly rounded decimal number $x_m^{\text{rounded}} \pm \Delta x^{\text{rounded}}$, so that its confidence level is guaranteed to be not less than $0.95 - \varepsilon$, the algorithm of Wimmer et al. [19] gives a recipe for obtaining the needed number of significant figures in x_m^{rounded}.

However, that algorithm is rather cumbersome and not easy to use. Furthermore, it does not strictly define the number of significant figures necessary in the rounded value of the uncertainty. Moreover, the rounding order of the measured value depends on the number of significant figures in the rounded value of its uncertainty, and the resulting numbers of places after the decimal point in the measured value and its uncertainty may be different. For example, let the measured value x be 0.04501, its uncertainty (one standard deviation) σ be 0.10101, and the required value of the confidence-level tolerance ε be 0.005. If we start from rounding σ to two significant figures, we arrive at the 0.005-proper rounded value of $x^{\text{rounded}} = 0.045 \pm 0.10$. If we start from rounding σ to three significant figures, we arrive at $x^{\text{rounded}} = 0.05 \pm 0.101$. To produce more sensible rounding, in which both the rounded values of the measured quantity and its uncertainty always have the same number of figures after the decimal point, we modified the algorithm of Wimmer et al. [19] as follows:

1) We always start from rounding the uncertainty value σ to two significant figures.
2) If the value of $\gamma \equiv \sigma^{\text{rounded}}/\sigma$ is lower than the threshold value $\gamma_{\text{threshold}}$ (computed as specified by Wimmer et al. [19]), increase the number of significant figures in σ^{rounded} until γ becomes greater than or equal to $\gamma_{\text{threshold}}$ (which depends on ε).
3) Compute the necessary rounding order for the measured value x as prescribed by Wimmer et al. [19] for the values of σ^{rounded} and γ resulting from the previous step and obtain the corresponding rounded value x^{rounded}.
4) Obtain a new value of the rounded uncertainty $\sigma^{\text{rounded}*}$ by rounding σ to the same order as x^{rounded}.
5) If the resulting number of required places after the decimal point in $\sigma^{\text{rounded}*}$ is different from that in σ^{rounded}, compute the new value of $\gamma^* = \sigma^{\text{rounded}*}/\sigma$. If γ^* is not less than $\gamma_{\text{threshold}}$, repeat step 3 with the values of $\sigma^{\text{rounded}*}$ and γ^* in place of σ^{rounded} and γ.

This algorithm does not always work with the approximate formulas for the required rounding order given by Wimmer et al. [19], since their approximation involves a square root of a polynomial containing powers of γ and ε, and this polynomial can be negative if values of ε are very small (< 0.005). To obtain estimates of the performance of rounding algorithms at small values of ε, we assumed that the approximation of Wimmer et al. [19] is valid for $\varepsilon < 0.005$ and skipped the results if the above-mentioned polynomial is zero or negative. By making numerical experiments with randomly selected numbers, we found that the approximation of Wimmer et al. works in at least 95% cases even with the smallest $\varepsilon = 0.001$, which was sufficient for our testing.

When we round the measured values using the rule-of-m rounding, the rounded values have a different distribution compared to the original (non-rounded) values. If the measured val-

Table A.1
Tolerance ε on the confidence levels imposed by the rule-of-m rounding with different values of m.

	m								
	72	43	27	19	15	11	8	5	3
ε	0.001	0.003	0.01	0.02	0.03	0.05	0.1	0.2	0.5
% difference	23.4	17.4	7.4	4.9	4.2	4.6	7.8	4.0	4.0
% errors	1.2	0.6	0.1	0.1	0.1	0.0	0.0	0.0	0.0

ues are normally distributed, and the one-standard-deviation measurement uncertainty is σ, we can say that the value of uncertainty equal to σ corresponds to the confidence level 0.68, the value of 2σ corresponds to the confidence level 0.9545, and the value of 3σ corresponds to the confidence level 99.7300 [23]. The rounded values do not necessarily have the same confidence levels. Instead, the confidence levels are decreased by a tolerance value, which depends on the roughness of rounding, i.e., on the parameter m of the rule-of-m rounding. The purpose of our testing was to establish this dependence. The testing was done by rounding off an arbitrarily selected "measured" value of 0.04501 with a set of 11 000 random uncertainty values between 0.0001 and 1.0000. The rounding was made using the two methods described above, the rule-of-m rounding and the ε-proper rounding. The values rounded using the two different methods were compared to each other, and the fraction of cases where the rounding order was different in the two methods was counted. For each given value of ε, it was possible to find the value of m that produced the best correspondence between the results of the rule-of-m and ε-proper rounding. With such choice of m, the rule-of-m rounding produced exactly the same rounded values as the ε-proper rounding in at least 80% of cases. Thus, we can say that for any given value of m we approximately determined the tolerance ε imposed by such rounding on the confidence levels of the uncertainties of the data. Table A.1 shows the results of this numerical experiment.

This dependence can be approximated by the following simple formula:

$$\varepsilon(m) = \left(0.00105m^2 + 0.363m + 0.251\right)^{-2}. \tag{A.1}$$

Extrapolating Eq. (A.1) to $m = 99.4999$, we find that the rule-of-99.4999 (or ISO 1995 standard) rounding results in the tolerance on the confidence level of approximately 0.0005 (on average). This decreases the 99.7% confidence level (corresponding to a three-standard-deviations uncertainty) by a relatively small amount of 0.05%, while the influence of this tolerance on the one- and two-standard-deviations confidence levels is negligible. Thus we conclude that the ISO 1995 standard (or rule-of-99.4999) rounding should be used for presentation of scientific data if statistical properties of the presented quantities, including the confidence levels for up to three-standard-deviations uncertainties, must be preserved with a reasonable degree of accuracy.

For the rule-of-9 rounding (implying that only one significant figure is given in the uncertainty value), the tolerance value is 0.08. This means that even the one-standard-deviation uncertainty value no more guarantees the confidence level of 68%; it is 60% instead, which can be interpreted as a possible increase of the uncertainty value by a factor of 1.14 (the actual increase depends on the value and its uncertainty being rounded). The confidence level of the two-standard-deviations uncertainty decreases from 95% to 87%, which can be interpreted as a possible increase of the uncertainty value by a factor of 1.32. The confidence level of the three-standard-deviations uncertainty decreases from 99.7% to 92%, which can be interpreted as a possible increase of the uncertainty value by a factor of 1.74. The general conclusion is that the rule-of-m rounding with $m < 10$ results in unacceptably large distortions of the statistical distribution function of the measured values and therefore should not be used for presentation of scientific data.

An intermediate case of the rule-of-20 rounding corresponds to a tolerance level of 1.6%. This decreases the guaranteed 68%, 95%, and 99.7% confidence levels to 67%, 94%, and 98.1%, respectively, which implies a possible increase of the uncertainty values by 2.6%, 7%, and 27%, respectively. We consider this small distortion of the statistical distribution as quite acceptable unless the distribution of highly deviating (by more than two standard deviations) outlying values is important for the particular investigation.

It should be noted that the main effect of the distortion of the probability distribution function by rounding is a decrease of confidence levels rather than an increase of variance. As explained below, variance of the rounded values is in most cases only slightly greater than variance of initial (unrounded) values.

Let us assume that measured values of a quantity x are described by a continuous probability distribution function with variance σ^2. As explained by Schwartz [24], if the measured values of x are quantized by rounding, their probability distribution function becomes discrete, and variance of the discrete rounded values V_d increases compared to σ^2 according to the following formula:

$$V_d = \sigma^2 + q^2/12, \tag{A.2}$$

where q is the value of the least significant digit in the rounded values (e.g., if the values are rounded to two places after the decimal point, then $q = 0.01$). For example, if the measurement uncertainty (standard deviation) is 0.05, and we are rounding all measurements to two places after the decimal point, the increase of the standard deviation due to rounding is 0.00008, which is negligibly small.

All the above discussion did not take into account the particular values of x, because it concerned only the overall statistical properties of large sets of random data. However, in the least-squares optimization we encounter a problem of another type. Namely, the procedure results in a set of numerical (floating-point) values, each calculated with a virtually infinite (machine-limited) precision, and each having an estimate of its uncertainty. When we round them off according to a chosen rule, for each of those values we know exactly how large is the effect of the rounding on each value. For example, if the calculated value is 0.25000001 ± 0.21, and we round it off according to the "rule of 20", it becomes 0.3 ± 0.2. However, we know that the rounding error in this case was approximately 0.05, and a reasonable question is: Do we have to increase the uncertainty of this value from 0.21 to 0.3 instead of decreasing it to 0.2?

To obtain an answer to this type of question, we made a numerical experiment with a large set of random computer-generated floating-point numbers. Description of the experiment is given in Appendix B. The main conclusion is as follows: If a measured (or calculated) value X having a known standard deviation D is rounded to X_r with an arbitrarily chosen rule-of-m, and the rounding error $d = X - X_r$ is known, then the standard deviation of X_r increases approximately as the sum in quadrature of D and d:

$$D_r^2 \approx D^2 + d^2. \tag{A.3}$$

The approximation error of the formula (A.3) is less than 2% for $m \geqslant 5$, and less than 15% for $m > 2$.

This approximation is used in the rounding algorithm implemented in the program LOPT.

Now we can answer the question posted above about the proper rounding of the value 0.25000001 ± 0.21 according to the "rule of 20". The rounded value should be 0.3 ± 0.2, since its standard deviation is $(0.21^2 + 0.05^2)^{1/2} = 0.216$. The influence of the rounding error on the rounding procedure is significant only for a small fraction of values with standard deviations that are very close to the rounding threshold (parameter m of the rule-of-m rounding). The size of this fraction depends on the range of uncertainty values and on m. If $m \geqslant 20$, this fraction is (on average) smaller than 0.003%. For $m = 10$, it is smaller than 0.05%.

Appendix B. Numerical investigation of the influence of rounding errors on the standard deviation of measured values from their expected mean

The numerical experiment was as follows:

1) For each of the arbitrarily chosen dispersion values D from the set (0.021, 0.03, 0.05, 0.07, 0.1, 0.15, 0.2, 0.25, 0.3, 0.5, 0.7, 0.9, 0.99), 10000 random values of mean expectation values μ were generated in the range $[0.0, 1.0)$ (including zero but excluding 1.0). To generate these random numbers, we used Perl's built-in function *rand*( ). Each of the values of μ was rounded to four places after the decimal point. For each combination (D, μ), the following steps were repeated 5000 times:
   1.1) A pair of random uniformly distributed numbers in the interval $(0, 1]$ (excluding zero but including 1) was generated using Perl's built-in function *rand*( ).
   1.2) This pair of random uniformly distributed numbers was transformed to a pair of normally distributed numbers (i.e. having a standard normal distribution function with zero mean value and standard deviation 1.0) by using the Box–Muller transformation [25].
   1.3) Each of the two random numbers generated in the previous step was scaled so that the resulting numbers a and b conform to a normal distribution function with the given dispersion D and expected mean value μ.
   1.4) Both a and b were rounded to one place after the decimal point, producing rounded values a_r and b_r. The differences $a - a_r$ and $b - b_r$ were rounded to two places after the decimal point, producing two values of deviation δ_a and δ_b. There are 11 possible equally spaced values of δ_a and δ_b ranging from -0.05 to $+0.05$ with an increment of 0.01. For each of these possible values d, the following sums were accumulated:

$$S(d, D, \mu) = \sum x, \tag{B.1}$$

$$S_2(d, D, \mu) = \sum x^2, \tag{B.2}$$

$$S_r(d, D, \mu) = \sum x_{\text{rounded}}, \tag{B.3}$$

$$S_{2r}(d, D, \mu) = \sum (x_{\text{rounded}})^2, \tag{B.4}$$

   where x corresponds to the values a and b. The counts of occurrence of each combination (d, D, μ) were collected in the variable $N(d, D, \mu)$.
2) The results of the step 1 were analyzed as follows:
   2.1) For each combination (d, D, μ), the following mean values over all possible values of μ were computed:

   mean of actual values of x,

$$M = S(d, D, \mu)/N(d, D, \mu), \tag{B.5}$$

   mean of squares of actual values of x,

$$M_2 = S_2(d, D, \mu)/N(d, D, \mu), \tag{B.6}$$

   mean of rounded values x_r,

$$M_r = S_r(d, D, \mu)/N(d, D, \mu), \tag{B.7}$$

   mean of squares of rounded values x_r,

$$M_{2r} = S_{2r}(d, D, \mu)/N(d, D, \mu). \tag{B.8}$$

   The standard deviation $D_r(d, D, \mu)$ of the rounded values x_r from the mean of the actual values x was calculated as follows:

$$D_r(d, D, \mu)^2 = \frac{\sum (x_r - M)^2}{N(d, D, \mu)} = M_{2r} - 2M_r M + M^2, \tag{B.9}$$

   and the standard deviation of the sample of the actual values of x for each d, D, and μ was calculated as well:

$$D_s(d, D, \mu)^2 = \frac{\sum (x - M)^2}{N(d, D, \mu)} = M_2 - M^2. \tag{B.10}$$

   The difference $\delta(d, D, \mu) = D_r(d, D, \mu) - D_s(d, D, \mu)$ was stored for each combination (d, D, μ).
   2.2) For each combination (d, D), means of $\delta(d, D, \mu)$ over all values of μ were calculated and stored in the variable $\delta_{\text{mean}}(d, D)$.
   2.3) The statistics for negative and positive values of d was averaged and stored in the variable $\delta^*_{\text{mean}}(d, D)$, where d assumed only non-negative values 0.00 through 0.05.

It should be noted that the values $\delta_{\text{mean}}(d, D)$ from step 2.2 were highly asymmetric in regards to the different sign of d for values of D smaller than 0.05. This leads to the conclusion that the rule-of-m rounding with $m < 5$ may produce a significant bias in the rounded values (which depends on the actual values of μ and D) and therefore must never be used in treatment of scientific data.

The resulting values of $\delta^*_{\text{mean}}(d, D)$ can be interpreted as an increase of the standard deviation due to rounding. It turns out that in most cases they can be approximated by the sum in quadrature:

$$\delta_{\text{quad}}(d, D) = \left(d^2 + D^2\right)^{1/2}. \tag{B.11}$$

This means that the normalized differences

$$\delta_{\text{norm}}(d, D) = \delta^*_{\text{mean}}(d, D)/\delta_{\text{quad}}(d, D) - 1 \tag{B.12}$$

are close to zero.

The values of $\delta_{\text{norm}}(d, D)$ resulting from the test are given in Table B.1. The small values of $\delta_{\text{norm}}(d, D)$ in Table B.1 indeed confirm that the sum in quadrature is a good approximation for the increase of the standard deviation of rounded values as a function of the rounding error d, especially for the cases when D is greater than 0.05, which correspond to the rule-of-m rounding with $m > 5$. For $m \leqslant 3$, the standard deviation of rounded values exceeds the sum-in-quadrature approximation by a noticeable fraction. In particular, for $m = 2.1$ (roughly corresponding to the rule-of-2 rounding), this excess amounts to 8 to 15% for non-zero values of d.

If we have a set of rounded values for which the values of d are unknown, we should expect that the standard deviation of these values increases compared to the standard deviation of unrounded values as described by the formula (A.2) [24] (see Appendix A). To verify it, we computed the averages of the increase of the standard deviation $\delta_{\text{ave}}(D)$ (using the values from Table B.1) over all possible values of d, and compared them to the formula (A.2). (Note that, for a set of random data, the number of cases with $|d| = 0.05$ and $|d| = 0$ is smaller by a factor of two than the num-

Table B.1
Normalized differences $\delta_{\mathrm{norm}}(d, D)$ of the mean standard deviation of rounded numbers from the sum-in-quadrature approximation.

$\lvert d\rvert$	D												
	0.021	0.03	0.05	0.07	0.1	0.15	0.2	0.25	0.3	0.5	0.7	0.9	0.99
0	−0.0404	0.0008	0.0017	0.0008	0.0004	0.0002	0.0001	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
0.01	0.1052	0.0243	0.0017	0.0009	0.0004	0.0002	0.0001	0.0001	0.0001	0.0000	0.0000	0.0000	0.0000
0.02	0.1533	0.0508	0.0018	0.0008	0.0004	0.0002	0.0001	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
0.03	0.1464	0.0593	0.0017	0.0008	0.0004	0.0002	0.0001	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
0.04	0.1292	0.0582	0.0017	0.0007	0.0004	0.0002	0.0001	0.0001	0.0001	0.0000	0.0000	0.0000	0.0000
0.05	0.0819	0.0181	−0.0236	−0.0162	−0.0096	−0.0048	−0.0028	−0.0018	−0.0013	−0.0005	−0.0002	−0.0001	−0.0001

Table B.2
Comparison of the increase of the standard deviation due to rounding, $\delta_{\mathrm{ave}}(D)$ with the approximation of Schwartz [24] (Eq. (A.2)).

	D												
	0.021	0.03	0.05	0.07	0.1	0.15	0.2	0.25	0.3	0.5	0.7	0.9	0.99
	$\delta_{\mathrm{ave}}(D) \times 100$												
From Table B.1	1.7312	1.2564	0.7427	0.5575	0.4030	0.2737	0.2067	0.1659	0.1385	0.0833	0.0595	0.0463	0.0421
From Eq. (A.2)	1.4698	1.1633	0.7735	0.5719	0.4083	0.2753	0.2073	0.1661	0.1386	0.0833	0.0595	0.0463	0.0421

ber of cases with any other value of $|d|$. Therefore, the values with $|d| = 0.05$ and $|d| = 0$ in Table B.1 were given a weight of 0.5 in the averaging.) This comparison is given in Table B.2. According to Schwartz [24], the formula (A.2) should be valid for the cases in our test where $D/0.1 > 1.0$. Indeed, Table B.2 confirms this assertion.

For the limiting case of infinitely precise data, $D = 0$, when the values are rounded so that the least significant decimal figure has a value of q, the mean rounding error is equal to

$$\left[\left(0.5^2/2 + 0.4^2 + 0.3^2 + 0.2^2 + 0.1^2\right)/5\right]^{1/2} q \approx 0.29q, \qquad \text{(B.13)}$$

which almost exactly coincides with the value resulting from Eq. (A.2).

References

[1] L.J. Radziemski Jr., K.J. Fisher, D.W. Steinhaus, Calculation of atomic-energy-level values, Los Alamos Scientific Lab., Univ. California Report LA-4402, Los Alamos, NM, 1970, 79 pp.
[2] R.S. Varga, Matrix Iterative Analysis, Prentice–Hall Inc., Englewood Cliffs, NJ, 1962.
[3] L.J. Radziemski Jr., K.J. Fisher, D.W. Steinhaus, A.S. Goldman, Comput. Phys. Comm. 3 (1972) 9.
[4] G.J. van het Hof, Orthogonal operators in atomic 3d systems, Ph.D. thesis, Amsterdam University, The Netherlands, 1990.
[5] A.E. Kramida, T. Bastin, E. Biémont, P.-D. Dumont, H.-P. Garnir, Eur. Phys. J. D 7 (1999) 547.
[6] K.J. Öberg, Eur. Phys. J. D 41 (2007) 25.
[7] A.E. Kramida, G. Nave, Eur. Phys. J. D 37 (2006) 1.
[8] J.W. Demmel, Applied Numerical Linear Algebra, Society for Industrial and Applied Mathematics, 1997.
[9] V.N. Vapnik, T.G. Glazkova, V.A. Koshsheyev, A.I. Mikhalskii, A.Ya. Chervonenkis, Algorithms and Programs for Reconstruction of Dependencies, Nauka, Moscow, 1984.
[10] G.A.F. Seber, Linear Regression Analysis, John Wiley & Sons, Inc., 2003.
[11] G.H. Golub, C.F. Van Loan, Matrix Computations, 3rd edition, Johns Hopkins University Press, 1996.
[12] N.J. Higham, Accuracy and Stability of Numerical Algorithms, 2nd edition, Society for Industrial and Applied Mathematics, 2002.
[13] A.E. Kramida, At. Data Nucl. Data Tables 96 (2010) 586.
[14] C.B. Moler, J. ACM 14 (1967) 316.
[15] J. Hicklin, C. Moler, P. Webb, R.F. Boisvert, B. Miller, R. Pozo, K. Remington, JAMA: A Java Matrix Package, online at http://math.nist.gov/javanumerics/jama/, 2005.
[16] L. Wall, T. Christiansen, J. Orwant, Programming Perl, 3rd edition, O'Reilly Media, 2000.
[17] V. Lindberg, Guide to Uncertainties and Error Propagation, online at http://www.rit.edu/cos/uphysics/uncertainties/Uncertainties.html, 2003.
[18] ISO 1995 Guide to the Expression of Uncertainty in Measurement, 1st edition, ISO, Geneva, 1995.
[19] G. Wimmer, V. Witkovský, T. Duby, Meas. Sci. Technol. 11 (2000) 1659.
[20] L.J. Radziemski Jr., V. Kaufman, J. Opt. Soc. Am. 59 (1969) 424.
[21] E.R. Peck, K. Reeder, J. Opt. Soc. Am. 62 (1972) 958.
[22] P.J. Mohr, B.N. Taylor, D.B. Newell, Rev. Mod. Phys. 80 (2008) 633, online at http://physics.nist.gov/constants.
[23] M. Abramowitz, I.A. Stegun (Eds.), Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, Dover Publications, New York, 1972.
[24] L.M. Schwartz, Anal. Chem. 52 (1980) 1141.
[25] G.E.P. Box, M.E. Muller, Ann. Math. Statist. 29 (1958) 610.