The following full text is a publisher's version.

For additional information about this publication click this link.
http://hdl.handle.net/2066/112310

Please be advised that this information was generated on 2017-07-26 and may be subject to change.
GENERALIZED RANK ANNIHILATION METHOD. III:
PRACTICAL IMPLEMENTATION

N. M. FABER, L. M. C. BUYDENS AND G. KATEMAN
Department of Analytical Chemistry, University of Nijmegen, Toernooiveld 1, NL-6525 ED Nijmegen,
Netherlands

SUMMARY

In this paper we discuss the practical implementation of the generalized rank annihilation method (GRAM). The practical implementation comes down to developing a computer program where two critical steps can be distinguished: the construction of the factor space and the oblique rotation of the factors. The construction of the factor space is a least-squares (LS) problem solved by singular value decomposition (SVD), whereas the rotation of the factors is brought about by solving an eigenvalue problem. In the past several formulations for GRAM have been published. The differences essentially come down to solving either a standard eigenvalue problem or a generalized eigenvalue problem. The first objective of this paper is to discuss the numerical stability of the algorithms resulting from these formulations. It is found that the generalized eigenvalue problem is only to be preferred if the construction of the factor space is not performed with maximum precision. This is demonstrated for the case where the dominant factors are calculated by the non-linear iterative partial least-squares (NIPALS) algorithm. Several performance measures are proposed to investigate the numerical accuracy of the computed solution. The previously derived bias and variance are proposed to estimate the number of physically significant digits in the computed solution. The second objective of this paper is to discuss the relevance of theoretical considerations for application of GRAM in the presence of model errors.

KEY WORDS
GRAM Least-squares problem Eigenvalue problem NIPALS Performance index Condition number

INTRODUCTION

The first contributions to the development of the method of rank annihilation are characterized by computational problems. For a long time these computational problems were thought to result from the iterative nature of the algorithm used: the solution of the problem demanded the minimization of the smallest significant eigenvalue of a residual matrix by means of trial and error. However, model errors caused several eigenvalues to vary simultaneously, thus leading to the difficult choice of the appropriate eigenvalue to be minimized. These computational difficulties were solved by the reformulation of the calibration problem as an eigenvalue equation by Lorber. Initially restricted to the one-component case, this reformulation was soon followed up by the generalization to the situation where both samples contain unique components. This generalization by Sánchez and Kowalski is known as the generalized rank annihilation method (GRAM). Many of the latest contributions to GRAM (see e.g. Reference 6), however, make use of an alternative algorithm that has been claimed by Wilson et al. to have better numerical properties.
In two previous papers we discussed the relationship between different formulations of GRAM and the effect of random measurement noise on the bias and variance in the estimated eigenvalues.\(^8,9\) (As a byproduct we also derived variance in the reconstructed profiles.) The primary result of that investigation was that the amount of bias and variance primarily depends on the specific combination of data matrices used for the construction of the factor space. It was shown that with respect to bias and variance the generalization of Sánchez and Kowalski compared favorably with the generalization of Wilson et al. In both papers we did not discuss the numerical properties of the different formulations. We also refrained from making speculations as to how the derived error estimates would behave in practice. In this paper we will pay attention to both aspects, since they determine how the practical implementation should be carried out and what may be expected if the method is to be used on real data.

One of the reviewers pointed out to us that the commercially available algorithms that would be used to solve the various forms of GRAM are stable\(^*\) and will produce all the accuracy necessary for getting reliable answers. We completely agree on this and want to stress once more that the statistical properties should be brought to bear on the problem of determining which method is ‘best’. However, some important additional background can be obtained by comparing different algorithms and presenting the solution in steps. The comparison will be made for the standard eigenvalue problem proposed by Sánchez and Kowalski\(^5\) (see References 8 and 9 for notational practice),

\[
(\bar{\Theta}^{-1}U^T\bar{M}V)T = T\Pi
\]

and a modified form of the generalized eigenvalue problem advanced by Wilson et al.,\(^7\)

\[
\bar{\mathbf{M}}_{uv}T = \bar{\mathbf{Q}}_{uv}T\Pi
\]

It should be noted that the method of Sánchez and Kowalski diagonalizes a slightly different matrix. However, equation (1) does not constitute a modification, since the results should be identical after changing the reconstruction formulae for the pure component profiles in a straightforward manner.\(^8\) The method of Wilson et al. is slightly modified in equation (2): instead of using the column and row augmented matrices to estimate the column and row space, we decompose the sum matrix \(Q\) according to the SVD. The results presented later show that it is not necessary to decompose the augmented matrices in order to obtain a numerically stable estimate of the factor space. Thus the only difference between (1) and (2) is the inversion of the diagonal matrix of singular values. According to numerical analysis, the conversion of a generalized to a standard eigenvalue problem constitutes no problem if the inverted matrix is stable (well conditioned) with respect to inversion. In that case it is even economic to solve the standard eigenvalue problem.\(^10\) Thus we are faced with the problem of ascertaining how instable the inverted matrix may be before we run into numerical problems when solving the standard eigenvalue problem. In a previous study we carried out calculations with data that were especially constructed in order to lead to a very instable problem.\(^9\) These calculations should therefore be ideally suited to test the conjecture that one should not be overly concerned about the stability of a particular algorithm as long as one solves the most stable problem. This will be illustrated by means of the performance measures. A performance index is proposed in order to monitor the size of the residuals for the critical steps, while the condition number is proposed to quantify the numerical stability. It is important to note that the condition

\* It is important to distinguish between the stability of the numerical problem and the stability of the algorithm. We will discuss this matter at length in later sections.
number is often used to predict (an upper bound) for the propagation of data error to the estimated parameters. With the availability of standard errors for the individual components the condition number is no longer useful for this purpose. Furthermore, bias is in no way diagnosed by the condition number.

The second objective of this paper is to discuss the relevance of theoretical considerations for application of GRAM in the presence of model errors. A notable effect of model errors is the possible occurrence of a complex solution. Li et al. introduced similarity transformations in order to convert the complex solution into a real one. This enabled the quantitation of a dilute compound in a complex matrix. For the acceptance of a method in a routine laboratory, however, it is necessary that together with the concentration also an estimate of its accuracy and precision is supplied. It is therefore imperative that the effect of the correction procedure on the final solution should be estimated, otherwise the proposed correction will not lead to a valid method. From this example it can be concluded that a discussion of the effect of model errors is essential for a method that is quite restrictive with respect to the assumed model.

In the following section we will describe which properties are essential for the correct performance of a computer program. Next we will introduce the performance measures that have been implemented to check the validity of the GRAM calculation. Finally the practical consequences of the theoretical considerations and the results for model data are discussed with respect to the analysis of data where model errors have to be taken into account.

**PROPERTIES OF A COMPUTER PROGRAM**

The solution of a numerical problem is preceded by the development of a computer program. A large number of aspects play a role in developing a computer program. Speed, efficient use of memory, simple structure, portability, friendly user interface, availability of a manual, precision, stability and reliability are all aspects that have to be considered. However, only the last three aspects are essential if the computer program is designed for scientific research. The other aspects become increasingly important if the program is to be converted into a commercial software package. We will therefore restrict ourselves to the last three concepts, since the solution of the problem is directly affected by them.

**Precision**

The precision of a solution refers to the number of exact figures. The simplest way to estimate the precision is to perform the calculation with a larger number of digits. The first digit that differs between the two solutions indicates that the remaining digits of the former solution are random. Figure 1 shows the relevance of the precision concept if the input data are of experimental origin. Because of measurement noise, only the first $p$ digits of the input data are physically significant. The next $q$ digits are also significant in the numerical sense. The last $r$ digits are also significant in the numerical sense. The last

\[
\begin{array}{c|c|c}
\text{p} & \text{q} & \text{r} \\
\hline
\text{input data} & \text{calculation} & \text{output solution}
\end{array}
\]

Figure 1. Schematic representation of the loss of physically significant and numerically significant digits for data with measurement error
digits constitute the representation error. (This error is caused by rounding the input decimal number to the nearest binary number.) During subsequent calculations rounding errors will affect the numerically significant digits \( p + q \). As a result the output number has become numerically significant in \( p' + q' \) digits. If the calculation is performed in double-precision floating point arithmetic, 14 digits, for example, may have survived the rounding process. However, this precision may be of little value if the number \( p' \) of physically significant digits has become too low for the practical application as a result of error propagation. It may turn out that we have calculated a meaningless number in 14 'exact figures'. If the problem itself is unstable, a different algorithm is not going to improve the situation. The only thing that may never happen is that the physically significant digits in the solution are affected by rounding errors. Stated differently: the solution to a possibly stable problem may never be ruined by an unstable algorithm. The precision of the solution is therefore intimately related to the following concepts.

**Stability and reliability**

A stable algorithm is insensitive to small changes in the input data. Reliability indicates that it is improbable that the program fails, or worse, that it gives the wrong answer without a warning. Reliability is obtained by building in controls to screen the input. There is a relation between stability and reliability: with a stable program the failure depends in a predictable way on small changes in the numerical problem.

A well-known example of an unstable algorithm is Gaussian elimination. This algorithm solves a system of linear equations by means of sweeping rows. If the element involved in the elimination process (the pivot) happens to be zero, the algorithm will return as answer that the input matrix is singular, even in cases where the input matrix is clearly invertible. This instability can be removed by interchanging rows and columns in such a way that the maximum element is used for the elimination (pivoting). The following example, taken from Reference 12, may be illustrative:

\[
\begin{pmatrix}
0 & 1 \\
1 & 0
\end{pmatrix}
\]

I: before pivoting

\[
\begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix}
\]

II: after pivoting

Both matrices are non-singular with ideal condition with respect to inversion (in contrast with the eigenvalue problem), but without pivoting, matrix I cannot be inverted by means of Gaussian elimination. Interchanging rows will immediately lead to the inverse. (It should be noted that Golub and van Loan define Gaussian elimination to use pivoting.)

Recently the stability of the NIPALS algorithm was discussed using the following example:

\[
A = \begin{pmatrix}
1 & 0.8 \\
-1 & 0.7
\end{pmatrix}, \quad A^T A = \begin{pmatrix}
2.13 & 0.12 \\
0.12 & 2.13
\end{pmatrix}
\]

The two eigenvalues of \( A^T A \) are \( \lambda_1 = 2.25 \) and \( \lambda_2 = 2.01 \) and the corresponding eigenvectors \( x_1^T = (1, 1) \) and \( x_2^T = (1, -1) \) respectively. We notice that this example is characterized by a very high symmetry. It is shown in Reference 16 that if a matrix is symmetrical around both diagonals, it will have eigenvectors that can be divided into two subsets. One subset contains the eigenvectors that are symmetric with respect to the midpoint, i.e. the point that divides the
components of the vector into two groups, \((x_1\) in this example), and the other subset contains the eigenvectors that are antisymmetric with respect to the midpoint \((x_2\) in this example). This symmetry is preserved by multiplication by the matrix and the outcome of the iteration procedure therefore depends on the particular choice of the starting vector. Starting exactly on an eigenvector will lead to convergence in one step. It is therefore essential to implement the proposed modifications in References 14 and 15 if this particular symmetry is pertinent to the problem at hand. It follows that NIPALS is an unstable algorithm, because small changes in the input* may lead to large changes in the output. It is also an unreliable algorithm, because the failure is not diagnosed in all cases by convergence in one step. It will, however, be clear that the examples referred to are very different from the data matrices usually encountered during a GRAM analysis.

**Performance Measures**

It is well known for the LS problem that small residuals in the data vector do not automatically correspond to small errors in the solution vector. The primary quantity for assessing the number of significant digits lost is given by the condition number of the inverted matrix. If the data are known to \(s = p + q + r\) in Figure 1) significant digits in base \(\beta\), if the condition number is of the order \(\beta^t\) and if a stable implementation of Gaussian elimination is used, then the number of significant digits in the computed solution will be \(s - t\) or zero, whichever is bigger. A similar reasoning holds for the eigenvalue problem, but now the invertibility of the eigenvector matrix has to be quantified. It follows that the numerical accuracy of the GRAM solution can be validated by monitoring both the residuals and the condition number of the critical steps of the calculation.

Some problems remain with respect to the practical evaluation of the following performance measures. First, the choice of a particular norm must be made. It is shown in Reference 13 that this choice is not critical (see also Reference 17). In order to obtain the smallest condition number, it is usually calculated by taking the largest singular value as matrix norm, i.e. the \(L_2\) matrix norm, but other norms are simpler to evaluate, e.g. the \(L_1\) and \(L_\infty\) matrix norms. Second, the value of the condition number (and therefore the estimated numerical precision of the solution) depends on the scaling of the matrix. General scaling strategies are unreliable and a problem-oriented approach should be followed considering measurement units and data error. The difficult matter of scaling will be further discussed in the 'Results and discussion' section. (See also Reference 18 where all calculations were performed on scaled data.)

**Performance Index**

The residuals of the SVD are monitored by calculating a performance index \((PI)\) for the individual factors. These indices are calculated from the size of the residual vector \(Qv_i - \theta_i u_i\) as

\[
(PI)_{SVD,i} = \frac{\|Qv_i - \theta_i u_i\|_1}{10n\epsilon \|Q\|_1 \|v_i\|_1}
\]

(3)

where \(\|\cdot\|_1\) denotes the \(L_1\)-norm, \(n\) is the length of the residual vector and \(\epsilon\) denotes the machine precision. The performance indices for the standard eigenvalue problem (SEP) are

* It is seen that convergence depends on the choice of the initial starting vector. Since the particular choice made is part of the input data, the algorithm is by definition instable.
calculated from the size of the residual vector $A\mathbf{x}_i - \lambda_i \mathbf{x}_i$ as

$$(PI)_{SEP,i} = \frac{\|A\mathbf{x}_i - \lambda_i \mathbf{x}_i\|_1}{10n\epsilon \|A\|_1 \|\mathbf{x}_i\|_1}$$

(4)

whereas the performance indices for the generalized eigenvalue problem (GEP) are calculated from the size of the residual vector $\beta_i A\mathbf{x}_i - \alpha_i B\mathbf{x}_i$ as

$$(PI)_{GEP,i} = \frac{\|\beta_i A\mathbf{x}_i - \alpha_i B\mathbf{x}_i\|_1}{10n\epsilon (|\beta_i| \|A\|_1 + |\alpha_i| \|B\|_1) \|\mathbf{x}_i\|_1}$$

(5)

where $|\cdot|$ denotes the modulus of a possibly complex number. The performance indices in (4) and (5) are adapted from the performance indices given in Reference 10 for the complete eigenvalue problem. (This reference was also the motivation for the use of the $L_1$ matrix norm.) This overall performance index is defined as the maximum of these indices. A value less than unity is considered to be excellent, a value between unity and 100 is considered good and a value larger than 100 is considered poor. The exact value is, however, machine-dependent. Since only the part of the eigensolution that corresponds to the calibrated components is interesting, we advise to calculate the performance index for each eigenvector.

It should be noted that the expression for the performance index of the generalized eigenvalue problem in Reference 10 is modified by adding the factor ‘$10n$’ to the denominator. Results presented later indicate that more consistent values are obtained this way. The performance index for the SVD is new to our knowledge and is presented here as the natural analogy to the performance indices for the eigenvalue problem.

**Condition number**

For the inversion in equation (1) the condition number is given by

$$\text{cond}(\Theta) = \| \Theta \| \| \Theta^{-1} \|$$

(6)

where $\| \cdot \|$ denotes a suitable matrix norm. Since $\Theta$ is a diagonal matrix, the usual matrix norms ($L_1, L_2, L_\infty$) lead to identical numerical values.

For the eigenvalue problem the condition number is given by

$$\text{cond}(T) = \| T \| \| T^{-1} \|$$

(7)

A small advantage of the condition number approach in combination with the eigenvalue problem is the fact that column scaling is automatically performed, since in contrast with the LS problem the matrix to scrutinize is a result of the calculation. However, another ambiguity arises because of the existence of equivalent representations of the eigenvalue problem. Although leading to identical eigenvalues, they yield eigenvector matrices that are related to each other by premultiplication by a diagonal matrix. Multiplication of the rows of the eigenvector matrix by a diagonal matrix may have a large influence on the numerical value of the condition number. This is not necessarily a serious problem, since only semiquantitative statements are possible on the basis of condition numbers.

**EXPERIMENTAL**

**Data**

Details about the generation of the data are given in Reference 9.
Calculations

The crucial steps in the GRAM calculation consist of the decomposition of a matrix and the solution of an eigenvalue problem. These steps are performed in part by black box routines from the IMSL library and in part by self-written or translated routines. An important advantage of not just using a black box routine is the possibility of studying the convergence properties of the algorithm. The complete SVD of $Q$ is calculated with the IMSL subroutine DLSVRR. The partial SVD is calculated with the NIPALS algorithm. The NIPALS algorithm is self-written and uses as starting vector the row of the data matrix with largest variance as recommended by Wold et al. The convergence criterion, i.e. the squared length of the difference of two subsequent normalized iterates, was set to the machine precision $\varepsilon = 2 \cdot 22 \times 10^{-16}$. The standard eigenvalue problem of equation (1) is solved with the modified Jacobi algorithm from Reference 20. The ALGOL procedure EIGEN was translated to FORTRAN77 for this purpose. The convergence criterion in EIGEN was adapted to the machine precision. The generalized eigenvalue problem of equation (2) is solved with the QZ algorithm implemented in the IMSL subroutine DGVCRG. Details about the algorithms used are summarized in Table 1. All calculations were performed on a HDS-EX60 mainframe computer. In Table 2 the calculation times for a complete GRAM analysis of the previously described simulated three-component system are compared with elementary floating point operations. These numbers will give an indication of the calculation time on another computer. It follows from Table 2 that calculating the first three factors with the NIPALS algorithm reduces the overall calculation time by a factor of two, but calculating an extra noise factor eliminates this advantage in speed because convergence is much slower for noise factors. The exact number of factors is usually not known in advance and it follows that calculation of the

<table>
<thead>
<tr>
<th>Calculation</th>
<th>Subroutine</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete SVD</td>
<td>DLSVRR</td>
<td>10</td>
</tr>
<tr>
<td>Partial SVD</td>
<td>NIPALS</td>
<td>19</td>
</tr>
<tr>
<td>SEP (equation (1))</td>
<td>EIGEN</td>
<td>20</td>
</tr>
<tr>
<td>GEP (equation (2))</td>
<td>DGVCRG</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Calculation</th>
<th>Execution time (CPU seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^7$ Empty DO-loops</td>
<td>1.7</td>
</tr>
<tr>
<td>$10^7$ Multiplications</td>
<td>0.5</td>
</tr>
<tr>
<td>$10^7$ Additions</td>
<td>0.5</td>
</tr>
<tr>
<td>$10^7$ Square roots</td>
<td>15.5</td>
</tr>
<tr>
<td>$10^4$ GRAM$^a$</td>
<td>2700</td>
</tr>
<tr>
<td>$10^4$ GRAM$^b$</td>
<td>1500</td>
</tr>
<tr>
<td>$10^4$ GRAM$^c$</td>
<td>5600</td>
</tr>
</tbody>
</table>

$^a$ Complete SVD by DLSVRR.  
$^b$ Three dominant factors by NIPALS.  
$^c$ Four dominant factors by NIPALS.
complete SVD will actually save time in many practical situations. A notable exception is provided by the method of cross-validation in principal component analysis (PCA). The basic method of cross-validation comes down to dividing the data matrix into a number of groups. Each group is deleted in turn from the data and a PCA performed on the reduced data set. The deleted values are then predicted from the PC model parameters. Since after the deletion of a group only the first PC has to be calculated, the computational gain may be considerable.

RESULTS AND DISCUSSION

We have divided this section into two parts. In the first part we give the results that were obtained calculating the full SVD of the sum data matrix \( Q \) with the IMSL subroutine DLSVRR. In the second part we give the results that were obtained calculating only the first three factors with the NIPALS algorithm.

Complete singular value decomposition of \( Q \) by IMSL subroutine DLSVRR

The matrices \( \bar{Q}_{UV} \) and \( \bar{M}_{UV} \) of equation (2) are given in Table 3. It is seen that the matrix \( \bar{Q}_{UV} \) is essentially a diagonal matrix. The diagonal elements would constitute the singular values if the calculations were exact. The effect of the rounding errors is reflected by the size of the off-diagonal elements. Now the question arises of whether it is permissible to convert the generalized eigenvalue problem into a standard one by simply inverting \( Q_{UV} \). In order to answer this question we must investigate the effect of measurement noise in the data and the effect of rounding errors on the computed solution separately (see Figure 1). In a previous paper we showed that the standard error in a singular value is equal to the standard deviation of the measurement noise.\(^\text{22}\) (Deterministic and therefore less efficient bounds for the perturbation of singular values are given by Lawson and Hanson.\(^\text{23}\) ) The standard deviation of the measurement noise is 0.05 for both matrices added in \( Q \).\(^\text{9}\) This means that the standard error in each singular value is 0.05,\( \delta = 0.07 \). It is seen that even the smallest singular value is about 50 times larger then its standard error and it follows that in spite of the large difference in scale the columns of the matrix \( \bar{Q}_{UV} \) (and its inverse) are orthogonal. This means that in order to assess the stability of the calculation, i.e. to predict the effect of rounding errors, the matrix is already perfectly ‘scaled’, because the uncertainty in every singular value is the same. A value of 1336 is obtained for the unscaled matrix. If, however, the condition number is to be used to predict that the numerical problem is stable and there is actually no propagation of data error on the basis of the condition number, one should scale the matrix \( \bar{Q}_{UV} \), giving the ideal value of one. (This lends credit to the approach of Otto and George to scaling the matrix.\(^\text{18}\) )

The results of the eigenanalysis with GRAM are summarized in Table 4. The eigenvalues are

| Table 3. Matrices \( \bar{Q}_{UV} \) and \( \bar{M}_{UV} \). Decomposition of \( Q \) with DLSVRR |
|---------------------------------|---------------------------------|---------------------------------|
| \( \bar{Q}_{UV} \) | \( \bar{M}_{UV} \) |
| \( 5 \cdot 2746 \times 10^1 \) | \( 1 \cdot 1 \times 10^{-13} \) | \( 1 \cdot 6 \times 10^{-12} \) | \( 1 \cdot 6410 \times 10^3 \) | \( -1 \cdot 0989 \times 10^1 \) | \( 1 \cdot 5014 \times 10^1 \) |
| \( -4 \cdot 2 \times 10^{-11} \) | \( 2 \cdot 0197 \times 10^1 \) | \( -6 \cdot 3 \times 10^{-15} \) | \( -1 \cdot 9356 \times 10^2 \) | \( 7 \cdot 4047 \) | \( -1 \cdot 0935 \) |
| \( -7 \cdot 1 \times 10^{-13} \) | \( -2 \cdot 0 \times 10^{-13} \) | \( 3 \cdot 9470 \) | \( 2 \cdot 2122 \times 10^1 \) | \( 8 \cdot 3195 \) | \( 0 \cdot 49980 \) |
divided into three parts according to Figure 1. The number of physically significant digits is estimated on the basis of the standard errors and bias. The number of numerically significant digits ('exact figures') is estimated by performing the same calculation on the data without measurement noise. (Actually this caused a floating point underflow in subroutine DLSVRR.) The solutions are seen to be very precise for both eigenvalue problems. This is satisfactorily predicted by the proposed performance measures. The performance indices indicate that the residuals are extremely small. This is particularly true for the SVD. This information should be combined with the value of the relevant condition number. The condition number for the eigenvector matrix is 72.86. (A value of 82.75 is found for the scaled spectrum matrix that was used to construct the data.) It should be noted that the loss of four or five decimal digits can easily be tolerated because we used double-precision arithmetic.

Partial singular value decomposition of Q by NIPALS

The matrices $\mathbf{Q}_{UV}$ and $\mathbf{M}_{UV}$ are given in Table 5. There are small differences with respect to the numbers in Table 3. It is seen that the off-diagonal elements of $\mathbf{Q}_{UV}$ are slightly larger. The elements of $\mathbf{M}_{UV}$ are identical to the precision given apart from pairwise changes of sign (that cancel in the SVD).

The results of the eigenanalysis with GRAM are summarized in Table 6. The influence of the loss of precision of the calculated SVD due to the use of NIPALS becomes clear on comparing Table 6 with Table 4. The performance index for the first factor is slightly better for the NIPALS algorithm, but the indices for the other factors show that three additional digits are lost. (Their value is no longer 'excellent' but 'good'.) Surprisingly, only the solution to the standard eigenvalue problem has seriously degraded. The reason is that when solving the generalized eigenvalue problem, both matrices are projected on the same, possibly slightly rotated, space. As a result the errors effectively cancel. The standard eigenvalue problem does

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\pi$</th>
<th>$(PI)_{SV}$</th>
<th>$(PI)_{EVP}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5001</td>
<td>6652058010</td>
<td>4648</td>
</tr>
<tr>
<td></td>
<td>0.5001</td>
<td>6652058013</td>
<td>6595</td>
</tr>
<tr>
<td>2</td>
<td>0.2530</td>
<td>1075726999</td>
<td>2989</td>
</tr>
<tr>
<td></td>
<td>0.2530</td>
<td>1075727002</td>
<td>9987</td>
</tr>
<tr>
<td>3</td>
<td>0.0511</td>
<td>9064208029</td>
<td>7126</td>
</tr>
<tr>
<td></td>
<td>0.0511</td>
<td>9064208027</td>
<td>6055</td>
</tr>
</tbody>
</table>

*Standard eigenvalue problem of equation (1).

*Generalized eigenvalue problem of equation (2).

<table>
<thead>
<tr>
<th>$\mathbf{Q}_{UV}$</th>
<th>$\mathbf{M}_{UV}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$5 \cdot 2746 \times 10^3$</td>
<td>$-2.1 \times 10^{-10}$</td>
</tr>
<tr>
<td>$1.5 \times 10^{-10}$</td>
<td>$2.0197 \times 10^1$</td>
</tr>
<tr>
<td>$7.7 \times 10^{-10}$</td>
<td>$-1.2 \times 10^{-9}$</td>
</tr>
</tbody>
</table>
Table 6. Results of eigenanalysis. Decomposition of $\mathbf{Q}$ with NIPALS

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\pi$</th>
<th>$(PI)_{SVD}$</th>
<th>$(PI)_{EVP}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$0.5001$</td>
<td>$6652052$</td>
<td>$0.003$</td>
</tr>
<tr>
<td></td>
<td>$0.5001$</td>
<td>$6652058025$</td>
<td>$2849^b$</td>
</tr>
<tr>
<td>2</td>
<td>$0.2530$</td>
<td>$1075732$</td>
<td>$1$</td>
</tr>
<tr>
<td></td>
<td>$0.2530$</td>
<td>$1075726986$</td>
<td>$0.045^b$</td>
</tr>
<tr>
<td>3</td>
<td>$0.0511$</td>
<td>$9064207$</td>
<td>$4$</td>
</tr>
<tr>
<td></td>
<td>$0.0511$</td>
<td>$9064207970$</td>
<td>$10615^b$</td>
</tr>
</tbody>
</table>

*a* Standard eigenvalue problem of equation (1).

*b* Generalized eigenvalue problem of equation (2).

not treat the two matrices symmetrically, since the projected matrix is replaced by its diagonal. Discarding the residuals thus leads to additional loss of significant digits. It follows that in combination with the NIPALS algorithm the solution of the generalized eigenvalue problem must be preferred. This conclusion presumably also holds if the partial least-squares (PLS) algorithm of Reference 6 is used to calculate the orthogonal base vectors. It is important to note that there is still some freedom of choice because the calculations were performed in double precision. If the calculations had been performed in single precision, part of the results in Table 6 would have been meaningless. The stability of a calculation depends on the algorithm as well as the precision of the arithmetic.13

CONSEQUENCES OF MODEL ERRORS

The preceding considerations about performance measures and results obtained for ideal bilinear data are perhaps interesting in theory but relatively useless in practice if model errors play a dominant role. Among the most frequently encountered model errors are matrix effects, interactions, non-linearities, heteroscedastic noise and retention time shifts (in chromatography). There are basically two ways to encounter model errors: a hardware and a software approach. The hardware approach comes down to improving the experiment in such a way that the assumed model for the data becomes essentially correct. This approach was followed by McCue and Malinowski, who optimized a liquid chromatograph with UV detection for rank annihilation and obtained excellent results.24 The alternative is to use more realistic models.

One may also simply ignore the fact that the assumptions about the model are violated and try to quantitate the effect in practice without doing anything about it. In order to facilitate the evaluation of the performance of multivariate methods on non-ideal data, so-called realistic simulations were developed in our department.25 These simulations come down to adding one-component data matrices with a specific weight (concentration) after background subtraction. The one-component data matrices are not approximated by the first factor as in the original work of Ho et al. to allow for all kinds of model errors that can give rise to additional significant factors. The results can, however, only be used to predict the errors for simple systems, since the number of factors to control by simulation increases rapidly with the number of overlapped components.
Matrix effects

Matrix effects can be corrected by the method of standard addition and this procedure is completely in the spirit of rank annihilation. Adding the analyte of interest to the unknown sample provides all the necessary data for the quantitation of this analyte. Departure from unity of the relative concentrations of the remaining components will give an indication about the quantitative aspects of the complete analytical procedure (i.e. starting from sample preparation until obtaining the raw data, e.g. injection volume irreproducibility in chromatography). Target testing\(^1\) of the added standards will provide useful information about the number of significant factors to use for the transformation. For data obtained by high-performance liquid chromatography with ultraviolet detection (HPLC–UV), target testing proved to give reasonable estimates of the number of absorbing species, while other well-known methods failed completely.\(^{25}\) Furthermore, if chromatographic data are used, standard addition will simplify the synchronization of the elution profiles.\(^6\)

Interaction effects

Interaction effects generate a cross-term between two concentrations. In contrast with matrix effects, this problem cannot be solved by, for example, standard additions. Neither can it be modelled within the framework of GRAM, since GRAM actually leads to an eigenvalue problem by excluding interactions. The influence on the calculated solution can, however, always be evaluated by means of realistic simulations.

Non-linearities

Non-linearities have been reported for data of entirely different sources. For fluorescence data Ho \textit{et al.} found that several of the largest eigenvalues varied linearly with concentration. Eigenvalues corresponding to dilute components were dominated by secondary eigenvalues of the strongly absorbing components. This introduced the problem of correctly interpreting the resulting eigenvalue plots. Lorber enabled the exact minima to be found by the solution of an eigenvalue problem. Since then, the eigenvector plots have no longer been reported in the literature. It is, however, clear that valuable information may be gained from these eigenvalue plots. We therefore suggest to construct these eigenvalue plots around the exact minima found by the solution of the eigenvalue problem, since the main disadvantage connected with the iterative method, i.e. locating the minimum, has already been solved. Dose and Guiochon\(^{26}\) have shown that non-linearities cannot be avoided for UV detectors. As a result more factors will be needed in order to describe the signal within the noise level.\(^{25}\) However, Li \textit{et al.}\(^6\) have shown that the stability of the problem is affected by the dimension of the transformation for real data. The dimension of the transformation matrix can be reduced to a minimum if non-linear factors\(^{27}\) are used for the projections. This could lead to a significant improvement of the stability of the problem (i.e. reduce standard errors).

Retention time shifts

Very recently Poe and Rutan\(^{28}\) have evaluated the sensitivity of the GRAM solution to retention time shifts by means of realistic simulations. Ironically, they find a negative bias in the estimated eigenvalues, while the theoretically predicted bias is positive. This illustrates both
the usefulness of these simulations as well as the inadequacy of predicted errors in the presence of model errors.

**Heteroscedastic noise**

It is well known that heteroscedastic noise can lead to additional significant eigenvalues in principal component analysis (PCA).\(^{29}\) In order to obtain a good fit with the 'correct' number of factors, weighted PCA must be carried out, i.e. the model must be adapted. Very recently Keller et al. have shown how the proper pretreatment of LC–UV data greatly improves the performance of evolving factor analysis (EFA).\(^{30}\) Since EFA assumes the same model as GRAM, this should be an important result for users of GRAM. Furthermore, LC–UV data is representative of the most important class of data nowadays being analysed with GRAM. It follows that in order to use GRAM on this kind of data, the same pretreatment should be used. It is perhaps interesting to note that the contribution of the separate modes to the total measurement error is not necessarily reflected in the generally asymmetric apportionment of error to the reconstructed profiles, since the error in the reconstructed profiles depends primarily on the overlap in the other mode.\(^9\)

**CONCLUSIONS**

The practical implementation of GRAM has been discussed with respect to the choice of the algorithms and the expected influence of model errors. A number of performance measures is proposed for the assessment of the reliability of the computed solution of GRAM. The performance indices are a measure of the size of the residuals that arise during the most important steps of the calculation, i.e. the construction of the factor space and the rotation of the factors. This information should be combined with the relevant condition number. The reliability of the computed solution is satisfactorily predicted in this way. It is shown that for the amount of overlap to be expected in practice, the number of numerically significant digits should greatly exceed the number of physically significant digits (predicted by the previously derived bias and variance expressions). Therefore the different algorithms that apply to this problem should all perform equivalently. This has been illustrated for the modified Jacobi algorithm (applied to the standard eigenvalue problem) and for the QZ algorithm (applied to the generalized eigenvalue problem). An exception must be made if the factor space used for the projections is not calculated with maximum numerical precision. It has been found that calculating the partial SVD by the NIPALS algorithm leads to an excessive loss of numerically significant digits if the standard eigenvalue problem is evaluated but has little effect on the solution of the generalized eigenvalue problem, since now both data matrices are projected on the same, less precise, factor space and the resulting errors tend to cancel out.

The effect of model errors on the outcome of the analysis has been discussed from a general point of view. Although good results are reported in the literature for the estimated concentrations and reconstructed profiles, the same cannot be expected for the predicted bias and variance. The reason for this unfortunate situation is that the concentration estimates are primarily determined by the adequacy of the model description of the data whereas the estimate of the confidence interval is primarily determined by the model description of the noise. The confidence levels predicted by theory may, however offer a reference point indicative of what could be possible in the absence of model errors. In the light of the fast development of theory we therefore agree with the prophetic statement of Sánchez and
Kowalski\textsuperscript{31} that ‘The greatest potential for GRAM, and in general second-order methods, is perhaps in future second-order instruments which are yet to be built’.

ACKNOWLEDGEMENTS

Dr Sijmen de Jong and the reviewers are acknowledged for rooting out errors and unclear passages.

REFERENCES