

A QR-DECOMPOSITION FOR MATRIX PENCILS

P. SPELLUCCI*, W. M. HARTMANN†

Abstract. In some applications there is a need for solving a linear least squares problem with a parametric matrix $\mathbf{A} + \lambda\mathbf{B}$ for several values of the parameter λ . This paper describes a modification of the QR-decomposition which allows to do this in an efficient and numerically stable way. The method is demonstrated on a typical application.

Keywords: linear least squares, matrix pencils

AMS(MOS) classification 65F20, 62J05

1. Introduction. In some applications, e.g. in SIMEX (SIMulated EXtrapolation) estimation of parameters [2], which is one form of linear *errors-in-variables regression*, there is a need for the solution of linear least squares problems

$$(1.1) \quad \|(\mathbf{A} + \lambda\mathbf{B})\mathbf{x} - \mathbf{f}\|_2^2 \stackrel{!}{=} \min_{\mathbf{x}}$$

for several values of λ . In this paper we deal with the situation where both \mathbf{A} and \mathbf{B} are $m \times n$ full matrices with $m \gg n$ which is the case in the above mentioned application.

Since $\lambda\mathbf{B}$ is not a low rank modification of \mathbf{A} , modification methods for QR- or SVD-decompositions of \mathbf{A} , see e.g. [8] and [1] are not useful here. For the case $m = n$ the QZ-decomposition of Moler and Stewart [6], see also [4] suggests itself. This would yield

$$\begin{aligned} \mathbf{QAZ} &= \mathbf{H} \text{ upper Hessenberg} \\ \mathbf{QBZ} &= \mathbf{R} \text{ upper triangular} \end{aligned}$$

and the solution of (1.1) could be finished by a sequence of $n - 1$ Givens-rotations transforming $\mathbf{H} + \lambda\mathbf{R}$ into final upper triangular form. Unfortunately, this transformation does not generalize to $m > n$ since trying to maintain the triangular structure of \mathbf{R} fails. There remains the possibility of solving (1.1) from scratch for every instance of λ , which however is quite costly. Therefore, in this paper we describe a modified QR-decomposition which takes advantage of the special structure of the problem as far as possible. This can be viewed as a generalization of the technique known from the implementation of the Levenberg-Marquardt algorithm [7]. In this algorithm a special instance of the problem (1.1) is encountered, namely

$$\mathbf{A} = \begin{pmatrix} \mathbf{J} \\ \mathbf{O} \end{pmatrix}, \mathbf{B} = \begin{pmatrix} \mathbf{O} \\ \mathbf{D} \end{pmatrix},$$

with \mathbf{O} denoting a matrix of zeroes and \mathbf{D} a diagonal nonsingular matrix. In this special situation, \mathbf{J} is transformed to upper triangular form first and then subsequently the matrix

$$\begin{pmatrix} \mathbf{R} \\ \mathbf{O} \\ \lambda\mathbf{D} \end{pmatrix}$$

is transformed to upper triangular form

$$\begin{pmatrix} \mathbf{R}_\lambda \\ \mathbf{O} \end{pmatrix}$$

using a series of $n(n + 1)/2$ Givens rotations combining elements $(i + k, i + k)$ and $(m + i + k, i + k)$ for $k = 0, \dots, n - 1$ and $i = k + 1, \dots, n - k$ in order to eliminate elements from the lower block, proceeding by diagonals.

¹TU Darmstadt, Dept. of Math. , Schloßgartenstraße 7, D 64289 Darmstadt, Germany.
spellucci@mathematik.tu-darmstadt.de

²SAS Institute, Inc., SAS Campus Drive, Cary, NC 27513, saswmh@unx.sas.com

2. QR-decomposition for Matrix Pencils. Our approach generalizes the technique known for the ordinary QR-decomposition, see e.g. [4]. We proceed by columns and in step i the algorithm is driven by the data from column i of the transformed matrices \mathbf{B} and \mathbf{A} in turn. Before stating the algorithm in formal terms we describe it by a picture. This gives the situation for $n = 3$ and $m = 8$. Here $\mathbf{U}_{i,A}$ denotes a Householder reflector computed from column i of the transformed matrix \mathbf{A} and similarly $\mathbf{U}_{i,B}$ is determined by column i of the transformed \mathbf{B}

$$\begin{array}{ccc}
\left(\begin{array}{ccc|ccc} + & + & + & + & + & + \\ + & + & + & + & + & + \\ + & + & + & + & + & + \\ + & + & + & + & + & + \\ + & + & + & + & + & + \\ + & + & + & + & + & + \\ + & + & + & + & + & + \\ + & + & + & + & + & + \end{array} \right) & \xrightarrow{\mathbf{U}_{1,B}} & \left(\begin{array}{ccc|ccc} + & + & + & + & + & + \\ + & + & + & 0 & + & + \\ + & + & + & 0 & + & + \\ + & + & + & 0 & + & + \\ + & + & + & 0 & + & + \\ + & + & + & 0 & + & + \\ + & + & + & 0 & + & + \\ + & + & + & 0 & + & + \end{array} \right) & \xrightarrow{\mathbf{U}_{1,A}} & \\
\left(\begin{array}{ccc|ccc} + & + & + & + & + & + \\ + & + & + & 0 & + & + \\ 0 & + & + & 0 & + & + \\ 0 & + & + & 0 & + & + \\ 0 & + & + & 0 & + & + \\ 0 & + & + & 0 & + & + \\ 0 & + & + & 0 & + & + \\ 0 & + & + & 0 & + & + \end{array} \right) & \xrightarrow{\mathbf{U}_{2,B}} & \left(\begin{array}{ccc|ccc} + & + & + & + & + & + \\ + & + & + & 0 & + & + \\ 0 & + & + & 0 & + & + \\ 0 & + & + & 0 & 0 & + \\ 0 & + & + & 0 & 0 & + \\ 0 & + & + & 0 & 0 & + \\ 0 & + & + & 0 & 0 & + \\ 0 & + & + & 0 & 0 & + \end{array} \right) & \xrightarrow{\mathbf{U}_{2,A}} & \\
\left(\begin{array}{ccc|ccc} + & + & + & + & + & + \\ + & + & + & 0 & + & + \\ 0 & + & + & 0 & 0 & + \\ 0 & 0 & + & 0 & 0 & + \\ 0 & 0 & + & 0 & 0 & + \\ 0 & 0 & + & 0 & 0 & + \\ 0 & 0 & + & 0 & 0 & + \\ 0 & 0 & + & 0 & 0 & + \end{array} \right) & \xrightarrow{\mathbf{U}_{3,B}} & \left(\begin{array}{ccc|ccc} + & + & + & + & + & + \\ + & + & + & 0 & + & + \\ 0 & + & + & 0 & + & + \\ 0 & 0 & + & 0 & 0 & + \\ 0 & 0 & + & 0 & 0 & + \\ 0 & 0 & + & 0 & 0 & 0 \\ 0 & 0 & + & 0 & 0 & 0 \\ 0 & 0 & + & 0 & 0 & 0 \end{array} \right) & \xrightarrow{\mathbf{U}_{3,A}} & \left(\begin{array}{ccc|ccc} + & + & + & + & + & + \\ + & + & + & 0 & + & + \\ 0 & + & + & 0 & + & + \\ 0 & 0 & + & 0 & 0 & + \\ 0 & 0 & + & 0 & 0 & 0 \\ 0 & 0 & + & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)
\end{array}$$

The resultant matrix $\tilde{\mathbf{A}} + \lambda \tilde{\mathbf{B}}$ is staircase triangular with $2i$ nonzeros at most in column i . For given value of λ it is then transformed into upper triangular form using Givens rotations which requires a total of $n^3/2 + \mathcal{O}(n^2)$ flops. (A flop is defined here as an operation of the form $a + b \# c \rightarrow d$ where $\#$ stands for \times or $/$). In column i , i rotations are to be applied on rows of length $n - i + 1$. Formally the reduction to the staircase triangular form is as follows:

Initialization: $(\mathbf{A}^{(0)}, \mathbf{B}^{(0)}) = (\mathbf{A}, \mathbf{B})$.

For $i = 1$ to $\min\{n, \lfloor \frac{m-1}{2} \rfloor\}$ do

$$\begin{aligned}
u_j^{(i)} &= 0, \quad j = 0, \dots, 2i - 2 \\
\gamma_i &= \left(\sum_{j=2i-1}^m (b_{j,i}^{(i-1)})^2 \right)^{1/2} \\
u_{2i-1}^{(i)} &= \text{sign}(b_{2i-1,i}^{(i-1)}) (|b_{2i-1,i}^{(i-1)}| + \gamma_i) \\
u_j^{(i)} &= b_{j,i}^{(i-1)}, \quad j = 2i, \dots, m \\
\beta_i &= 2 / \|\mathbf{u}^{(i)}\|_2^2 \\
(2.1) \quad (\tilde{\mathbf{A}}^{(i)}, \tilde{\mathbf{B}}^{(i)}) &= (\mathbf{A}^{(i-1)}, \mathbf{B}^{(i-1)}) - \beta_i \mathbf{u}^{(i)} ((\mathbf{u}^{(i)})^T (\mathbf{A}^{(i-1)}, \mathbf{B}^{(i-1)})) \\
w_j^{(i)} &= 0, \quad j = 0, \dots, 2i - 1
\end{aligned}$$

$$\begin{aligned}
\tilde{\gamma}_i &= \left(\sum_{j=2i}^m (\tilde{a}_{j,i}^{(i)})^2 \right)^{1/2} \\
w_{2i}^{(i)} &= \text{sign}(\tilde{a}_{2i,i}^{(i)}) (|\tilde{\gamma}_i| + \tilde{\gamma}_i) \\
w_j^{(i)} &= \tilde{a}_{j,i}^{(i)}, \quad j = 2i + 1, \dots, m \\
\tilde{\beta}_i &= 2 / \|\mathbf{w}^{(i)}\|_2^2, \\
(2.2) \quad (\mathbf{A}^{(i)}, \mathbf{B}^{(i)}) &= (\tilde{\mathbf{A}}^{(i)}, \tilde{\mathbf{B}}^{(i)}) - \tilde{\beta}_i \mathbf{w}^{(i)} ((\mathbf{w}^{(i)})^T (\tilde{\mathbf{A}}^{(i)}, \tilde{\mathbf{B}}^{(i)})).
\end{aligned}$$

Here the setting $\text{sign}(0) = 1$ is assumed. In the computation of (2.1) and (2.2) the special structure of the Householder matrices is to be exploited of course. The algorithm requires essentially $2mn^2 - \frac{4}{3}n^3$ flops for $m \geq 2n + 1$. It can be easily enhanced with simultaneous row and column interchanges in \mathbf{A} and \mathbf{B} to improve its stability properties even further in case of strongly varying row or column norms. Compared with a QR-decomposition of $\mathbf{A} + \lambda\mathbf{B}$ from scratch it is more efficient if three instances of λ are required at least. If we compare it with the normal equations solution combined with Cholesky-decomposition, which requires essentially $mn^2/2 + n^3/6$ flops, at least four instances of λ are required to justify the approach from the viewpoint of computational complexity. Of course, the normal equations approach suffers from the danger of numerical illconditioning which our solution does not.

3. Numerical Example. We tested our algorithm in two applications as follows: The first test considered the execution efficiency of the code. For given (m, n) a matrix \mathbf{A} and a solution vector \mathbf{x} were generated using the matrix generator of CMAT, [5]. Then \mathbf{f} was computed such that $\mathbf{A}\mathbf{x} = \mathbf{f}$. Then 100 instances of matrices \mathbf{B}_i with independent normally distributed entries with expectation value zero and variance one were generated and for values of $\lambda = i/30$, $i = 1, \dots, 30$ the linear least squares problems

$$\|(\mathbf{A} + \lambda\mathbf{B}_i)\mathbf{x}_{\lambda, B_i} - \mathbf{f}\|_2^2 \stackrel{!}{=} \min_{\mathbf{x}_{\lambda, B_i}}$$

were solved by our approach. The mean value

$$\mathbf{x}(\lambda) = \frac{1}{100} \sum_{i=1}^{100} \mathbf{x}_{\lambda, B_i}$$

was computed. This gave a set of 30 pairs $(\lambda, \mathbf{x}(\lambda))$. Following the theory of SIMEX we should have $\mathcal{E}(\mathbf{x}(0)) = \mathbf{x}$. In order to model $\mathbf{x}(\lambda)$ exactly we would have to use $4n$ values of λ , since each $x_i(\lambda)$ is a rational function of λ of numerator degree $2n - 1$ and denominator degree $2n$ at most. The data points were fitted componentwise by rational functions

$$f_i(\lambda) = \frac{a_{0,i} + a_{1,i}\lambda}{1 + a_{2,i}\lambda}, \quad i = 1, \dots, n$$

and finally $f_i(0)$ was taken as approximation for x_i . Therefore this approximation is a rather crude one, but this was not our concern here.

The timing results presented some surprise to us. From the operation counts given in the previous section we should expect timing relations like 1:7.5:15 for large m and n , comparing our approach with the use of the normal equations and the QR-decomposition from scratch, since 30 values of λ are used. This however was never obtained. The best values obtained were 1:6.4:13.1 and for very large m and n the relation was finally 1:2.56:7.1. A closer look at the details of our hardware revealed the reason why. We used a DELL PC with Intel Pentium II, 200MHz, with 256 KB cache and 128MB main memory. Simultaneous storage of A and one instance of B_i requires $16mn$ bytes of cache. Hence for $mn > 16000$ at least, cache failures occurred and this quite different for the different codes. In the normal equations approach the system matrix, requiring $4n(n + 1)$ bytes in cache only is accumulated from the outer products of the rows of $\mathbf{A} + \lambda\mathbf{B}_i$. Hence the large matrices \mathbf{A} and \mathbf{B}_i are accessed once only, and this rowwise. In our approach, were occur $n^2 + \mathcal{O}(n)$ accesses to individual columns of \mathbf{A} resp. \mathbf{B}_i , which are stored by rows as is natural in this application. This increases the number of cache failures dramatically. In the QR-decomposition from scratch this effect is halved. This is well reflected in the timing results. Therefore,

for large m and n the computing time for our algorithm may be dominated by transfer time from cache to memory depending on the specifics of the hardware. In the following some diagrams showing computing time are given. The key "neq" denotes the normal equations solution, "uqr" our algorithm and "fqr" the QR-decomposition computed from scratch. Time is cputime in seconds.

The second test concerned a simulated SIMEX extrapolation. In common least-squares regression it is assumed that the data matrix \mathbf{A} is errorfree, what is often not the case in practice. A number of errors-in-variables methods for linear regression were developed, like *instrumental variable estimation* [3] and *total least-squares* [9], but SIMEX [2] is obviously the most general approach that can easily be extended to nonlinear regression models including GLIM models and also provides techniques for the estimation of variance of parameter estimates, i.e. standard errors and confidence intervals. In a typical SIMEX application, the errorfree data matrix \mathbf{A} is not given but a perturbed $\mathbf{A} + \mathbf{B}_0$ with known distribution of \mathbf{B}_0 where $\mathbf{A} + \mathbf{B}_0$ corresponds to $\lambda = 0$. In this case, additive \mathbf{B}_i are chosen from the same distribution and $\mathbf{x}(\lambda)$ is extrapolated to $\lambda = -1$. Then the expectation value of $\mathbf{x}(-1)$ with respect to \mathbf{B}_0 is the true parameter vector.

Here in a small experiment we used a given matrix \mathbf{A}

$$\mathbf{A} = \begin{pmatrix} 6 & 28 \\ 12 & 40 \\ 10 & 32 \\ 8 & 36 \\ 9 & 34 \end{pmatrix}$$

and computed the right hand side $\mathbf{f} = \mathbf{A}\mathbf{x}$ with $\mathbf{x} = (0, 1, 1)$ for

$$\|\mathbf{A}\mathbf{x} - \mathbf{f}\|_2^2 \stackrel{!}{=} \min_{\mathbf{x}} \quad .$$

Generating large samples of error matrices \mathbf{B}_i with normal distributed $\mathcal{N}(0, .1)$ entries and using a fully parametrized rational extrapolation function

$$f(\lambda) = \frac{a_0 + a_1\lambda + a_2\lambda^2 + a_3\lambda^3 + a_4\lambda^4 + a_5\lambda^5}{1 + c_1\lambda + c_2\lambda^2 + c_3\lambda^3 + c_4\lambda^4 + c_5\lambda^5 + c_6\lambda^6}$$

with $\lambda_i = i/12$, $i = 1, \dots, 12$ we expected to obtain good estimates of \mathbf{x} by extrapolating toward $\lambda = 0$. If the meanvalues with respect to \mathbf{B}_i would be replaced by the true expectation values, then $\mathbf{x}(\lambda)$ would be exactly represented componentwise by such a function f . The following table contains the results for the sample sizes of 100, 1000, and 10000 of matrices \mathbf{B}_i :

Sample Size	Intercept x_0	x_1	x_2
100	0.0125025	1.0012001	0.9993136
1000	0.0026488	1.0007602	0.9997202
10000	-0.0016422	1.0000092	1.0000474

The table shows that the exact solution $x = (0, 1, 1)$ can be recovered for large samples.

4. Conclusion. We gave a QR-decomposition for matrix pencils which maintains the good stability properties while increasing the efficiency compared with the computation from scratch considerably. or large dimensional problems cache failures may however degrade this efficiency.

REFERENCES

- [1] Åke Björck: *Numerical Methods for Solving Least Squares Problems*. SIAM: Philadelphia 1996 .
- [2] R.J. Carroll, D. Ruppert, L.A. Stefanski: *Measurement errors in Nonlinear Models*. Chapman and Hall: Boston 1995.
- [3] W.A. Fuller: *Measurement Error Models*. J. Wiley: New York 1987.
- [4] G.H. Golub, Ch.F. van Loan: *Matrix Computations*. 3rd ed. John Hopkins: Baltimore 1996.
- [5] W. Hartmann: *CMAT: Extension of C Language . Matrix Algebra . Nonlinear Optimization and Estimation . Users Manual*. 1997.
- [6] C.B. Moler , G.W. Stewart: *An algorithm for generalized matrix eigenvalue problems*. SINUM 10, (1973), 241–256 .

- [7] J.J. Moré: *The Levenberg-Marquardt algorithm: implementation and theory*. pp 105–116 in Lecture Notes on Mathematics 630, (Numerical Analysis, G. Watson ed.). Springer: Heidelberg 1977 .
- [8] L. Reichel, W.B. Gragg: *Updating the QR-decomposition of a matrix*. ACM TOMS 16, (1990), 369–377 .
- [9] S. Van Huffel, J. Vandewalle: *The Total Least Squares Problem*. SIAM: Philadelphia 1991.