Numerically Stable Cointegration Analysis

Jurgen A. Doornik[*]
Nuffield College, University of Oxford, Oxford OX1 1NF
R.J. O'Brien
Department of Economics University of Southampton

November 3, 2001

Abstract

Cointegration analysis involves the solution of a generalized eigenproblem involving moment matrices and inverted moment matrices. These formulae are unsuitable for actual computations because the condition numbers of the resulting matrices are unnecessarily increased. Our note discusses how to use the structure of the problem to achieve numerically stable computations, based on QR and singular value decompositions.

Keywords: Cointegration, QR decomposition, Reduced rank, Singular-value decomposition.

JEL classification: C32, C63

1 Introduction

Many observed time series appear to be non-stationary. A simple form of non-stationarity arises when the first differences of a series are white noise. In that case we say that the series, say y_t, is integrated of order one, $I(1)$, and for the first differences: $\Delta y_t \sim I(0)$. An important step in the statistical analysis of integrated series was the realization that it is also possible for a linear combination of $I(1)$ variables to be $I(0)$. These variables are then said to cointegrate.

A simple framework for the statistical analysis is the vector autoregression (VAR). For example, when using one lag:

$$y_t = \pi y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathrm{N}_n\left[0, \Omega\right], \tag{1}$$

where y_t is an $n \times 1$ vector and π an $n \times n$ matrix. The disturbances ε_t are independent and identically distributed, with N_n denoting the n-dimensional normal distribution This system can be written as a vector equilibrium-correction model (VECM), by subtracting y_{t-1} from both sides:

$$\Delta y_t = \left(\pi - I_n\right) y_{t-1} + \varepsilon_t,$$

using I_n for the $n \times n$ identity matrix. Alternatively, writing $\Pi = \pi - I_n$:

$$\Delta y_t = \Pi y_{t-1} + \varepsilon_t.$$

This shows that the matrix Π determines how the level of the process y enters the system: for example, when $\Pi = 0$, the dynamic evolution does not depend on the levels of any of the variables. The statistical hypothesis of cointegration is:

$$\mathsf{H}(r)\colon \text{ rank } (\Pi) \leq r.$$

[*] Correspondence to: jurgen.doornik@nuffield.ox.ac.uk

Under this hypothesis, Π can be written as the product of two matrices:

$$\Pi = \alpha\beta',$$

where α and β have dimension $n \times r$, and vary freely. As suggested by Johansen (1988) (also see Johansen, 1995), such a restriction can be analyzed by maximum likelihood methods.

Maximum likelihood estimation requires solving a generalized eigenproblem. The purpose of this note is to show that the standard method for solving this eigenproblem, as proposed in the statistical literature and discussed in the next section, is numerically undesirable. We then consider several alternatives which are numerically more stable. We also consider an example to illustrate the impact of the different implementations. But, before we continue, we summarize the two orthogonal decompositions that are used.

The singular value decomposition (SVD) of an $m \times n$ matrix A, rank$A = r$ is $A = U\Sigma V'$. Assuming $m \geq n$ gives the 'thin' version: U is $m \times n$, V is $n \times n$, $U'U = V'V = I_n$, and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$. The singular values are real, non-negative, and ordered: $\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0$. So the rank of A corresponds to the number of non-zero singular values.

The QR decomposition of a full rank $m \times n$ matrix A, $m \geq n$, is $A = \mathbf{QR}$. Here $\mathbf{Q}$ is $m \times m$, $\mathbf{Q}'\mathbf{Q} = I_m$, and $\mathbf{R}$ is an upper triangular $m \times n$ matrix with positive diagonal elements. Since only the top n rows of $\mathbf{R}$ are non-zero, it is customary to use the thin version where R consists of the first n rows of $\mathbf{R}$ and Q of the first n columns of $\mathbf{Q}$, so $\mathbf{Q} = (Q : *)$, $\mathbf{R}' = (R' : 0)$ and $A = QR$. Instead of storing $\mathbf{Q}$, only the $m \times n$ matrix of Householder transformations is stored. $R'R$ is the Choleski decomposition of $A'A$. Normally, the QR decomposition is augmented with column pivoting to handle rank deficient A, see Golub and Van Loan (1996, Ch.5).

2 Standard algorithm for cointegration analysis

A more general version of (1) allows for longer lags, and deterministic variables such as constant and trend. From a statistical point of view, the treatment of the constant and trend is very important (see, e.g. Doornik, Hendry, and Nielsen, 1998). The trend, if present, is usually restricted to lie inside the cointegration space, leading to the general VECM:

$$\Delta x_t = \Pi x^*_{t-1} + \sum_{j=1}^{k-1} \Gamma_j \Delta x_{t-j} + \Phi q_t + \varepsilon_t, \qquad t = 1, \ldots, T, \tag{2}$$

where $\{x_t\}$ is an n-vector time series, the starting values $(x_{1-k}, \ldots, x_0)$ are fixed, $\{\varepsilon_t\}$ is i.i.d. $\mathsf{N}_n[0, \Omega]$, and $x^*_{t-1} = (x'_{t-1}, d'_t)'$, where d_t and q_t are deterministic regressors. T is the sample size after allowing for lags and differences.

The maximum likelihood method for estimating the cointegration space, proposed by Johansen (1988), writes $\Pi = \alpha\beta'$, where α is $n \times r$, and β is $n_1 \times r$. In this case, there are $n_1 - n$ variables d_t restricted to lie in the cointegrating space. The unrestricted deterministic variables q_t and lagged differences (and possibly additional non-modelled variables) can be partialled out (i.e. removed through a prior regression) from Δx_t and x^*_{t-1}, giving y_{0t} and y_{1t}:

$$y_{0t} = \alpha\beta' y_{1t} + \varepsilon_t, \qquad t = 1, \ldots, T. \tag{3}$$

Stacking the partialled data, $Y'_0 = (y_{01} \cdots y_{0T})$, and $Y'_1 = (y_{11} \cdots y_{1T})$ gives the matrix equivalent of (3):

$$Y_0 = Y_1\beta\alpha' + E. \tag{4}$$

The likelihood is maximized by solving the generalized eigenvalue problem:

$$|\lambda S_{11} - S_{10}S_{00}^{-1}S_{01}| = 0, \tag{5}$$

writing $S_{ij} = T^{-1}Y_i'Y_j$, for $i, j = 0, 1$. This can be translated in a standard eigenvalue problem through pre and post multiplication by $S_{11}^{-1/2}$, see Johansen (1995, p.95), giving n eigenvalues $1 > \widehat{\lambda}_1 > \ldots > \widehat{\lambda}_n > 0$. The remaining (if any) $n_1 - n$ eigenvalues are zero. The hats on the λs indicate that these have been estimated. The 'trace' statistic tests whether Π has rank $\leq r$, given that it has rank n or less: $-T\sum_{i=r+1}^{n}\log(1-\widehat{\lambda}_i)$. Under the null hypothesis, the trace test has a distribution which is a functional of multivariate Brownian motions. Full details on the statistical analysis are given in Johansen (1995).

Simple restrictions of the form $\alpha = B\theta$ and $\beta = H\phi$ can be incorporated with only small adjustments (as can calculation of Wald tests for different r). Let $B = Q_bR_b$ be the QR decomposition of B. Then (3) can be written as:

$$Q_b'y_{0t} = R_b\theta\phi'H'y_{1t} + Q_b'\varepsilon_t.$$

The following algorithm has often been used to implement the Johansen procedure:

Algorithm 1 *If Y_0, Y_1 have full column rank, then ML estimation of (4) can be achieved by:*

(a) Compute second moment matrices S_{00}, S_{01}, S_{11} :	S_{ij}	$\leftarrow$	$T^{-1}Y_i'Y_j,$
(b) Choleski decomposition of S_{11} :	S_{11}	$=$	$PP',$
(c) Solve symmetric eigenproblem :	$P^{-1}S_{01}'S_{00}^{-1}S_{01}P^{-1\prime}$	$=$	$H\Lambda H'.$

Λ is a diagonal matrix with the eigenvalues $\widehat{\lambda}_i$. The remaining quantities are:

$$\widehat{\beta} = P^{-1\prime}H, \quad \widehat{\alpha} = S_{01}\widehat{\beta}.$$

Given β, α can be computed from $\widehat{\alpha}(\beta) = S_{01}\beta(\beta'S_{11}\beta)^{-1}$ (Johansen, 1995, p.91). Since $\widehat{\beta}'S_{11}\widehat{\beta} = I$, $\widehat{\alpha}$ follows as given. Note that α and β are not uniquely determined: $\alpha\beta' = \alpha V^{-1}V\beta'$ for any non-singular V.

Solving a generalized eigenproblem $|\lambda A - B| = 0$ with A, B symmetric and positive definite is analyzed in Golub and Van Loan (1996, §8.7). However, in the cointegration case there is more structure in A and B. Consequently, Algorithm 1 is numerically undesirable, in the same way that regression is not implemented by inverting a second moment matrix. The next three sections present alternative algorithms which are numerically more stable.

In the remainder we concentrate on (4). Note that it is essential that the move from (2) to (3) is done in a numerically stable way, for example using the QR decomposition with column pivoting (this is the default procedure for OLS in most software packages).

3 Algorithm 2: QR decomposition

We first consider an alternative procedure for the Johansen procedure that uses the QR algorithm with pivoting on both data matrices.

Algorithm 2 *If Y_0, Y_1 have full column rank, then ML estimation of (4) can be achieved by:*

$$\begin{array}{lrcl}
\textit{(a) Thin QR decomposition of } Y_0: & Y_0P_0 & = & Q_0R_0, \\
\textit{(b) Computation of } T \times n_1 \textit{ matrix } W & W & \leftarrow & \mathbf{Q}_0'Y_1, \\
\textit{(c) QR decomposition of } W: & WP_2 & = & Q_2R_2, \\
\textit{(d) Thin SVD of } Z = Q_2(1:n,:)': & Z & = & U\Sigma V'.
\end{array}$$

The required quantities are:

$$\widehat{\lambda}_i = \sigma_i^2, \quad R_2\widetilde{\beta} = T^{1/2}U, \quad \widehat{\beta} = P_2\widetilde{\beta} \quad \widetilde{\alpha} = T^{-1/2}R_0'Z'U, \quad \widehat{\alpha} = P_0\widetilde{\alpha},$$

where $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$. V is not required. P_i are permutation matrices.

Proof. Note that the algorithm is identical to computing the thin QR of Y_i: $Y_iP_i = Q_iR_i$, and then the SVD of $Q_1'Q_0$. We present the algorithm in an alternate format because most QR implementations do not return Q, but the sequences of Householder transformations instead, complemented by a procedure to compute $\mathbf{Q}'A$.

To see that the algorithm solves (5) without pivoting (i.e. setting P_i to the identity matrix), write $Y_i = Q_iR_i$ and substitute into (5):

$$|T^{-1/2}R_1||\lambda I - Q_1'Q_0Q_0'Q_1||T^{-1/2}R_1'| = 0.$$

Using $ZZ' = U\Sigma^2U'$, we find that the squared singular values of $Q_1'Q_0$ are the required eigenvalues. Next, $T^{-1/2}R_1\beta = U$ gives the cointegrating vectors. Note that this does not require inversion, and can be solved by simple backsubstitution, yielding $\widetilde{\beta}$. Applying the permutation gives $\widehat{\beta}$. Finally, substitution in $\widehat{\alpha}(\widehat{\beta}) = S_{01}\widehat{\beta}$ yields $\widehat{\alpha}$. □

The algorithm is similar to that given in Björck and Golub (1973) and analyzed in Golub and Zha (1994) (also see Golub and Van Loan, 1996, §12.4.3). The adjustment for pivoting to (4) is:

$$Y_0P_0 = Y_1P_1P_1'\beta\alpha'P_0 + EP_0.$$

Using the Cauchy–Schwartz inequality: $\sigma_1^2 = ||Z||_2^2 \le ||Q_1||_2^2||Q_0||_2^2 \le 1$. So all eigenvalues are between zero and one.

4 Algorithm 3: QR decomposition

The next alternative procedure is also QR based, but works on the combined data matrix.

Algorithm 3 *If Y_0, Y_1 have full column rank, then ML estimation of (4) can be achieved by:*

$$\begin{array}{ll}
\textit{(a) Thin QR decomposition of the} & \\
\quad T \times (n_1 + n) \textit{ matrix } Y = (Y_1\ Y_0): & Y = QR = Q\begin{pmatrix} R_{11} & R_{10} \\ 0 & R_{00} \end{pmatrix}, \\
\textit{(b) Computation of } n_1 \times n \textit{ matrix Z} & Z \leftarrow R_{10}R_{00}^{-1}, \\
\textit{(c) Thin SVD of } Z: & Z = U\Sigma V'.
\end{array}$$

The required quantities are:

$$\widehat{\lambda}_i = \frac{\sigma_i^2}{1+\sigma_i^2}, \quad R_{11}\widehat{\beta} = T^{1/2}U, \quad \widehat{\alpha} = T^{-1/2}R_{10}'U,$$

where $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$; Q and V are not required.

Proof. The second-moment matrix of Y using the QR decomposition without pivoting from step (a) contains the S_{ij}:

$$T^{-1}Y'Y = T^{-1}\begin{pmatrix} R_{11}'R_{11} & R_{11}'R_{10} \\ R_{10}'R_{11} & R_{10}'R_{10} + R_{00}'R_{00} \end{pmatrix} = \begin{pmatrix} S_{11} & S_{10} \\ S_{01} & S_{00} \end{pmatrix}.$$

We need a standard result for symmetric partitioned matrices. Define

$$\begin{pmatrix} A & B \\ B' & D \end{pmatrix},$$

where A and D have full rank. Then $(A - BD^{-1}B')^{-1} = A^{-1} + A^{-1}B(D - B'A^{-1}B)^{-1}B'A^{-1}$. When $A = I, D = -R_{00}'R_{00}$ and $B = R_{10}$:

$$\left[I + R_{10}(R_{00}'R_{00})^{-1}R_{10}'\right]^{-1} = I - R_{10}(R_{10}'R_{10} + R_{00}'R_{00})^{-1}R_{10}'.$$

Substituting this into (5) gives an alternative symmetric eigenproblem:

$$\begin{aligned}
0 &= |\lambda S_{11} - S_{10}S_{00}^{-1}S_{01}| \\
&= |T^{-1/2}R_{11}'||\lambda I - R_{10}\left(R_{10}'R_{10} + R_{00}R_{00}\right)^{-1}R_{10}'||T^{-1/2}R_{11}| \\
&= |T^{-1/2}R_{11}'||(\lambda - 1)I + [I + R_{10}(R_{00}'R_{00})^{-1}R_{10}']^{-1}||T^{-1/2}R_{11}| \\
&= |T^{-1/2}R_{11}'||(\lambda - 1)^{-1}I + I + R_{10}(R_{00}'R_{00})^{-1}R_{10}'||T^{-1/2}R_{11}| \\
&= |T^{-1/2}R_{11}'||\lambda(\lambda - 1)^{-1}I + R_{10}(R_{00}'R_{00})^{-1}R_{10}'||T^{-1/2}R_{11}| \\
&= (-1)^{n_1}|T^{-1/2}R_{11}'|| - \lambda(\lambda - 1)^{-1}I - R_{10}(R_{00}'R_{00})^{-1}R_{10}'||T^{-1/2}R_{11}|.
\end{aligned}$$

In the fourth line we use the fact that the eigenvalues of an inverted symmetric matrix are the reciprocal eigenvalues of the original matrix. If μ_i are the eigenvalues solving the term in the middle, then $\lambda_i = \mu_i/(1 + \mu_i)$ (and hence $0 \leq \lambda_i < 1$). Also see Johansen (1995, p.94). R_{11} is again an upper-triangular matrix, but R_{10} is not. □

Like algorithm 2, this algorithm uses a QR decomposition for the first step, and the singular value decomposition to solve the symmetric eigenvalue problem. The main difference is that Algorithm 2 only uses Q from the QR step, whereas Algorithm 3 only uses R.

Implementations of the QR decomposition would normally signal rank deficiencies in Y_i. In that case, both algorithms should abort with an error message. Algorithm 2 will produce an answer for the singular case, but this does not solve the original problem. Algorithm 3 is numerically less reliable, because it does not use pivoting, and requires the inverse of R_{00}.

5 Algorithm 4: the singular case

If either Y_0 or Y_1 has reduced rank, the previous algorithms break down. A more reliable solution is to use the SVD throughout:

Algorithm 4 *ML estimation of (4) can be achieved by:*

$$\begin{array}{lrcl}
\textit{(a) Thin SVD decomposition of } Y_i,\ i = 0,1: & Y_i & = & U_i\Sigma_i V_i', \\
\textit{(b) Computation of } n_1 \times n \textit{ matrix Z} & Z & \leftarrow & U_1'U_0, \\
\textit{(z) Thin SVD of } Z: & Z & = & U_z\Sigma_z V_z'.
\end{array}$$

The required quantities are:

$$\widehat{\lambda}_i = \sigma_i^2, \quad \widehat{\beta} = T^{1/2}V_1\widetilde{\Sigma}_1^{-1}U_z \quad \widehat{\alpha} = T^{-1/2}V_0\Sigma_0 Z'U_z \ (\equiv T^{-1/2}Y_0'U_1U_z),$$

$\widetilde{\Sigma}_1^{-1} = \mathrm{diag}(\widetilde{\sigma}_1^{-1} \cdots \widetilde{\sigma}_n^{-1})$, *where* $\widetilde{\sigma}_i^{-1} = \sigma_i^{-1}$ *when* $\sigma_i > \epsilon_y$, *and* $\widetilde{\sigma}_i^{-1} = 0$ *otherwise.*

Proof. This is similar to Algorithm 3:

$$|T^{-1/2}V_1\Sigma_1||\lambda I - U_1'U_0U_0'U_1||T^{-1/2}\Sigma_1V_1'| = 0.$$

$\square$

The choice of ϵ_y would depend on the machine precision. We suggest using $\epsilon_y = 10^{-9}||Y_1||_\infty$, which is appropriate for double precision (8-byte) arithmetic.

6 Operation count

We follow Golub and Van Loan (1996) in the definition of floating point operation (flop) count, counting a scalar addition, multiplication, or division as one flop. Golub and Van Loan (1996, p.225) give the flop count of the thin Householder QR decomposition of an $m \times n$ matrix as $2mn^2 - 2n^3/3$. In this note, m corresponds to the sample size T, and is assumed to be significantly larger than n. Therefore, we take the number of flops of the thin QR as $2Tn^2$. Application of the Householder transformation to compute $\mathbf{Q}'A$ when A is $T \times q$ uses $4Tnq$ flops (*op.cit.* p.212–213). In some cases we also require Q, the first q columns of $\mathbf{Q}$. This is most efficiently done through backward accumulation (*op.cit.* p.213), and can be shown to take approximately $4qmn - 2(m+n)q^2 + \frac{4}{3}q^3$ operations. Therefore, when $m = T$ dominates, and $q = n$, this reduces to $2Tn^2$, the same as for the QR decomposition itself. Using the same large T argument, we assume that the thin SVD takes $6Tn^2$ flops (*op.cit.* p.254), regardless whether V is required or not, because the term with T is assumed to dominate. Computation of $A'B$ when A, B are $T \times n$ matrix takes $2Tn^2$ flops, and $A'A$ half that.

Assume that $n_1 = n$ and that the sample size T is sufficiently large to make Tn^2 the dominant term. Then algorithm 1 forms three S_{ij} moment matrices, taking $Tn^2(1$ for S_{11} + 2 for S_{10} + 1 for $S_{00}) = 4Tn^2$.

Algorithm 2 takes two QR decompositions at about $2Tn^2$ flops each, as well as a multiplication with Q_0 at $4Tn^2$ and extraction of Q_2 at another $2Tn^2$, resulting in a total of $10Tn^2$. This can be reduced to $8Tn^2$ if Q_2 is found from $(WP_2)(1:n,:) = Q_2(1:n,:)R_2$, but this has some negative effect on the numerical stability, making it comparable to Algorithm 3. If the algorithm is implemented as the QR of Y_i: $Y_iP_i = Q_iR_i$, and then the SVD of $Q_1'Q_0$ it would also take $10Tn^2$ flops.

Algorithm 3 does one QR of a $T \times 2n$ matrix, which takes $2T(2n)^2$ flops. Finally, algorithm 4 requires two SVD decomposition and the computation of Z.

The approximate operation count for the three algorithms under those assumptions equals:

Algorithm	flops when $T \gg n$
1	$4Tn^2$
2	$10Tn^2$
3	$8Tn^2$
4	$14Tn^2$

7 Illustration

To illustrate the different properties of the algorithms, we estimate a two-equation VAR(2) which has both equations asymptotically identical:

$$\mathbf{y}_t = \left(y_t, y_t + u_t10^{-m}\right) \text{ where } u_t \sim \mathsf{N}(0,1).$$

The estimated ECM has no deterministic terms:

$$\Delta \mathbf{y}_t = \alpha\beta' \mathbf{y}_{t-1} + \Delta \mathbf{y}_{t-1} + \varepsilon_t.$$

We select y_t as the *CONS* variable from a tutorial data set of Ox (Doornik, 2001). Ox was used for all computations, and we created u_t using the default random number generator. For m not too large, this problem will yield the same cointegration eigenvalues as (y_t, u_t), namely 0.35091938503 and 0.00450356266. As m gets very large, the system reduces to (y_t, y_t). The single equation VAR(2) involving y_t has eigenvalue 0.00487801748.

Table 1: Largest eigenvalue for VAR(2) $(y_t, y_t + u_t 10^{-m})$. Incorrect digits in bold.

	$m = 1$	$m = 3$	$m = 5$	$m = 10$
Algorithm 1a	0.3509194**0555**	0.35089**570155**	**0.05949553867**	failed
Algorithm 1b	0.3509194**0557**	0.35089**589051**	**0.13273076746**	failed
Algorithm 2	0.35091938503	0.35091938503	0.350919385**42**	0**.01335798065***
Algorithm 3	0.35091938503	0.35091938503	0.350919385**40**	failed
Algorithm 4	0.35091938503	0.35091938503	0.350919384**94**	0.00487801748*

$*$ A warning signalling singularity is also printed.

The results are in Table 1 for selected values of m, obtained with Ox version 2.20 (the rounding errors accumulate slightly differently in Ox 3.00). Algorithm 1a and 1b only differ in the order of evaluation: the former has $P^{-1}[S_{01}' S_{00}^{-1} S_{01}] P^{-1\prime}$, while the latter uses $[P^{-1} S_{01}'] S_{00}^{-1} [S_{01} P^{-1\prime}]$. Both algorithms are clearly much less stable than those using QR and SV decompositions. At $m = 5$ they have broken down, with completely wrong eigenvalues. In this case there is no indication of failure: only at $m = 6$ does the Choleski decomposition break down. Note that with Algorithm 1 it is actually possible for badly conditioned problems to find eigenvalues which are negative or greater than one! The fact that the algorithm is so sensitive to the ordering of the computations is also a sign of instability.

Algorithm 2 signals problems from $m = 6$ onwards, warning that Y_0 has reduced column rank, but keeps producing an answer. Algorithm 3 also signals problems from $m = 6$ onwards, but is unable to provide a solution in that case.

Algorithm 4 is designed to handle singularities. From $m = 10$ onwards it gives the correct answer as if there was only one y. Between $m = 7$ and $m = 10$ it moves to that value: for $m = 6$ the largest eigenvalue is 0.350917 but for $m = 7$ it is: 0.004914. The value for ϵ_y in this example is 2.1×10^{-6}; from $m = 7$ onwards, the algorithm signals the singularity.

Algorithms 2–4 are clearly superior to Algorithm 1, still producing essentially the correct answer at $m = 5$.

8 Conclusion

We presented and analyzed several algorithms for maximum likelihood estimation of cointegration which are numerically more stable than the standard form which uses second moment matrices. For a general purpose implementation, where singularity could be an issue, we recommend using the SVD-based algorithm 4. Otherwise, for example in Monte Carlo experiments, Algorithm 3 could be used as the fastest algorithm that can detect non-singularity, or 2 as the more stable algorithm. Most importantly, all alternative algorithms are an improvement on the standard method, because they will will signal problems instead of reporting potentially spurious results.

Acknowledgements

Financial support from the UK Economic and Social Research Council (grant R000237500) is gratefully acknowledged by JAD. We also wish to thank two anonymous referees for their comments.

References

Björck, Å. and G. H. Golub (1973). Numerical methods for computing angles between linear subspaces. *Mathematics of Computation 27*, 579–594.

Doornik, J. A. (2001). *Object-Oriented Matrix Programming using Ox* (4th ed.). London: Timberlake Consultants Press.

Doornik, J. A., D. F. Hendry, and B. Nielsen (1998). Inference in cointegrated models: UK M1 revisited. *Journal of Economic Surveys 12*, 533–572.

Golub, G. H. and C. F. Van Loan (1996). *Matrix Computations* (3rd ed.). Baltimore: The Johns Hopkins University Press.

Golub, G. H. and H. Zha (1994). Perturbation analysis of the canonical correlations of matrix pairs. *Linear Algebra and Its Applications 210*, 3–28.

Johansen, S. (1988). Statistical analysis of cointegration vectors. *Journal of Economic Dynamics and Control 12*, 231–254.

Johansen, S. (1995). *Likelihood-based Inference in Cointegrated Vector Autoregressive Models*. Oxford: Oxford University Press.