STATIONARY VALUES OF THE RATIO OF QUADRATIC FORMS SUBJECT TO LINEAR CONSTRAINTS

BY

GENE H. GOLUB

RICHARD UNDERWOOD

TECHNICAL REPORT NO. CS 142

NOVEMBER, 1969

COMPUTER SCIENCE DEPARTMENT

School of Humanities and Sciences

STANFORD UNIVERSITY

STATIONARY VALUES OF THE RAT-IO OF QUADRATIC FORMS

SUBJECT TO LINEAR CONSTRAINTS

Gene H. Golub *

Richard Underwood

* This author was in part supported by Atomic Energy Commission.

Abstract

Let A be a real symmetric matrix of order n , B a real symmetric positive definite matrix of order n , and C and nxp matrix of rank r with $r < p < n$. We wish to determine vectors $\underset{\sim}{x}$ for which

$$\underset{\sim}{x}^T A \underset{\sim}{x} \,/\, \underset{\sim}{x}^T B \underset{\sim}{x}$$

is stationary and $C^T \underset{\sim}{x} = \underset{\sim}{\theta}$, the null vector. An algorithm is given for generating a symmetric eigensystem whose eigenvalues are the stationary values and for determining the vectors $\underset{\sim}{x}$. Several Algol procedures are included.

1. Introduction and Theoretical Background

Let A be a real symmetric matrix of order n, B a real symmetric positive definite matrix of order n, and C an $n \times p$ matrix of rank r with $r < p < n$. We wish to determine vectors x such that

$$x^T Ax / x^T Bx$$

is stationary and $C^T \underset{\sim}{x} = \underset{\sim}{\theta}$, the null vector.

By rearranging the columns of C, we may write

$$QC = \left[\begin{array}{c|c} \tilde{R}_r & S \\ \hline 0 & 0 \end{array}\right]$$

where $\tilde{R}_r$ is an upper triangular matrix of order r, S is $r \times (p-r)$, and $Q^T Q = I$. The matrix Q may be constructed as the product of r Householder transformations (cf. [3]).

Let

$$x = Q^T w = Q^T \left[\begin{array}{c} \underset{\sim}{y} \\ \hline \underset{\sim}{z} \end{array}\right]$$

where y is a vector of the first r components of w and $\underset{\sim}{z}$ consists of the last $(n-r)$ components of w. Thus

$$C^T x = \left[\begin{array}{c|c} \tilde{R}_r^T & 0 \\ \hline S^T & 0 \end{array}\right] \left[\begin{array}{c} \underset{\sim}{y} \\ \hline \underset{\sim}{z} \end{array}\right] = \underset{\sim}{\theta}$$

and hence

$$Y = \theta .$$

$$\tilde{w}^T Q A Q^T \tilde{w} \;/\; \tilde{w}^T Q B Q^T \tilde{w} \quad \text{subject to } w_1 = w_2 = \dots = w_r = 0 .$$

Let

$$G = Q A Q^T = \begin{bmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{bmatrix}, \quad H = Q B Q^T = \begin{bmatrix} H_{11} & H_{12} \\ H_{12}^T & H_{22} \end{bmatrix}$$

where G_{11}, H_{11} are $r \times r$ matrices, and G_{22}, H_{22} are $(n-r) \times (n-r)$ matrices. The matrices H and G are symmetric; H is positive definite, and H_{22} is positive definite. Indeed,

$$0 < \lambda_{min}(H) \le \lambda_{min}(H_{22}) \le \lambda_{max}(H_{22}) \le \lambda_{max}(H) \quad .$$

Thus the stationary values we seek, are the eigenvalues of the matrix equation

$$G_{22} \tilde{z} = \lambda H_{22} \tilde{z} \tag{1}$$

Since G_{22} and H_{22} are symmetric and H_{22} is positive definite, we may solve (1) by using standard algorithms (cf. [6]). Finally, if

$$G_{22} \tilde{z}_i = \lambda_i H_{22} \tilde{z}_i \qquad (i = 1,2,\dots,n-r) ,$$

then

$$\tilde{x}_i = Q^T \begin{bmatrix} 0 \\ \cdots\cdots \\ I_{n-r} \end{bmatrix} \tilde{z}_i \quad . \tag{2}$$

When $p = 1$, and $B = I$, a slightly different algorithm may be used for computing the stationary values. We assume

$$c^T c = 1 \quad . \tag{3}$$

Let

$$\varphi(x) = x^T A x - \lambda x^T x + 2 \mu x^T c \tag{4}$$

where (λ, μ) are Lagrange multipliers. Differentiating (4), we are led to the equation

$$Ax - \lambda x + \mu c = \Theta \quad . \tag{5}$$

Multiplying (5) on the left by c^T and using (3), we have

$$\mu = - c^T A x \tag{6}$$

Thus substituting (6) into (5), we have

$$P A x = \lambda x$$

where $P = I - cc^T$. Note $P^2 = P$ so that

$$\lambda(PA) = \lambda(P^2 A) = \lambda(PAP) \quad .$$

The matrix PAP is symmetric and consequently one of the standard methods may be used for computing its'eigenvalues.

It is easy to construct the matrix PAP using a device of Wilkinson [9]. Let

$$K = PAP = (I - cc^T)A(I - cc^T)$$
$$= A - cw^T - wc^T + \alpha cc^T \quad ,$$

where

$$\alpha = \mathbf{c}^T A \mathbf{c} \qquad \text{and} \qquad \mathbf{w} = A\mathbf{c} \quad .$$

Then if

$$\mathbf{u} = \frac{a}{2}\mathbf{c} - \mathbf{w} \quad ,$$

$$K = A + \mathbf{c}\mathbf{u}^T + \mathbf{u}\mathbf{c}^T$$

Therefore if

$$K\mathbf{z}_i = \lambda_i \mathbf{z}_i \quad ,$$

then

$$\mathbf{x}_i = P\mathbf{z}_i \qquad (i = 1,2,\ldots,n) \; .$$

The vector $\mathbf{c}$ is an eigenvector and the corresponding eigenvalue is zero.

2. Applicability

2.1 Testing for serial correlations

Let X be a given $n \times p$ matrix of rank r and y be a known vector. The vector b is the least squares estimate of regression vector so that

$$\|\underset{\sim}{y} - X\underset{\sim}{b}\|_2 = \min.$$

In many situations, it is desirable to consider the statistic

$$d = z^T A \underset{\sim}{z} / \underset{\sim}{z}^T \underset{\sim}{z}$$

where $\underset{\sim}{z} = \underset{\sim}{y} - X\underset{\sim}{b}$, the residual vector, and A is a given symmetric matrix. For

$$A = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}$$

the statistic d is the serial correlation of lag one. Note that $X^T z = \Theta$. We wish to consider the distribution of d over all possible z . Thus under a suitable transformation, we may write

$$d = \sum_{i=1}^{n-r} \lambda_i \xi_i^2 / \sum_{i=1}^{n-r} \xi_i^2$$

where $\{\lambda_i\}_{i=1}^{n-r}$ are the stationary values of $\underset{\sim}{z}^T A\underset{\sim}{z}$ over $\underset{\sim}{z}^T\underset{\sim}{z} = 1$ with $X^T\underset{\sim}{z} = \underset{\sim}{\theta}$. The distribution of d is discussed in special cases in [2].

2.2 Exponential fitting

In many situations, we observe a sequence $\{z_k\}_{k=1}^m$, and we wish to determine parameters $\{\alpha_i\}_{i=0}^q$, $\{\lambda_i\}_{i=1}^q$ so that

$$z_k \approx \alpha_o + \sum_{i=1}^{q} \alpha_i \lambda_i^k \quad (k = 1,2,\ldots,m) . \tag{7}$$

From (7), we note that $\{z_k\}_{k=1}^m$ satisfies a difference equation of the form

$$a_0 z_k + a_1 z_{k-1} + \cdots + a_{q+1} z_{k-q-1} = \epsilon_k \quad (k = q+1,\ldots,m)$$

where ϵ_k is a random perturbation. The coefficients $\{a_i\}_{i=0}^{q+1}$ determine the <u>characteristic polynomial</u>:

$$p(\lambda) = a_o\lambda^{q+1} + a_1\lambda^q + \ldots + a_{q+1}$$

Note $p(1) = 0$ by (7).

One procedure which may be used to estimate the coefficients of the characteristic polynomial is the determine $\{a_i\}_{i=0}^k$ so that

$$\sum_{k=q+1}^{m} \epsilon_k^2 = \min.$$

subject to the constraints $\sum_{i=0}^{q+1} a_i^2 = 1$ and $\sum_{i=0}^{q+1} a_1 = 0$. In matrix form, we have the problem of determining $\underset{\sim}{a}$ so that

$$\underset{\sim}{a}^T W^T W\underset{\sim}{a} = \min.$$

with

$$\underset{\sim}{a}^T\underset{\sim}{a} = 1 \quad \text{and} \quad \underset{\sim}{e}^T\underset{\sim}{a} = 0$$

where

$$W = \begin{bmatrix} z_{q+1} & ,\ldots,z_1,z_0 \\ z_{q+2} & ,\ldots,z_2,z_1 \\ \vdots & \vdots \\ z_m, z_{m-1} & ,\ldots,z_{m-q-1} \end{bmatrix}, \quad \underset{\sim}{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{q+1} \end{bmatrix}, \quad \underset{\sim}{e} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Thus the procedure outlined in Section 1 may be used for determining $\underset{\sim}{a}$. A more sophisticated statistical model for determining $\underset{\sim}{a}$ is given in [8] by Osborne.

2.3 Sloshing frequencies

In [5], Henrici _et al._ give a method for determining approximations (with rigorous error bounds) for the sloshing frequencies of an ideal fluid contained in a half-space with a circular or strip-like aperture. The stationary values may be obtainednumerically by the method described in Section 1.

3. Formal Parameter List

3.1 Input to Procedure REDUCE

n	number of rows of C.
p	number of columns of C .
tol	a machine dependent constant equal to eta/macheps, where eta is the smallest positive real number representable on the computer, and macheps is the machine precision, the smallest ϵ such that $1+\epsilon > 1$.
eps	a tolerance used in determining the rank of C .
c[1:n,1:p]	contains the matrix C to be reduced.

Output of procedure REDUCE

c[1:n,1:r]	together with d[1:r] , contains the details of the transformations which reduce C to upper triangular form.
d[1:r]	see above;
r	column rank of C .

3.2 Input to procedure APPLY

n	order of the matrix AB .
r	number of similarity transformations to be performed.

d[1:r]	see output of procedure REDUCE.
c[1:n,1:r]	see output of procedure REDUCE.
ab[1:n,1:n]	contains in its upper triangle the details--of the symmetric matrix AB .
gh[1:n-r,1:n-r]	contains in its upper triangle the details of the symmetric matrix GH , which is an n-r x n-r submatrix of 'the matrix obtained by applying the r similarity transformations contained in d and c to AB .

3.3 Input to'-'procedure BACKTRANSFORM

n	number of rows in C .
r	number of backtransformations to be performed.
d[1:r]	see output of procedure REDUCE.
c[1:n,1:r]	see output of procedure REDUCE.
z[1:n-r,1:n-r]	contains the matrix Z , the vectors to be transformed.

Output of procedure BACKTRANSFORM

x[1:n,1:n-r]	contains the matrix X obtained by applying the r transformations contained in d and c to the n x(n-r) matrix, the first r rows of which are zero, and the last n-r , Z .

4. Algol Programs

```
procedure reduce(n) data:(p,tol,eps) data and result:(c) result:(r,d);
value n,p,tol,eps; integer n,p,r;
real tol, eps; array c,d;
comment This procedure computes the sequence of r Householder
  transformations necessary to reduce the nxp matrix C  (n > p > 0)
  to upper triangular form.  On input, c[1:n,1:p] contains the
  columns of c.  On output, c[1:n,1:r] and d[1:r] contain the
  details of the transformations.  r is the column rank of C;
begin integer i,j,k,m;
  real h,f,g;
  array sumsq[1:p];
  comment Compute the lengths of the columns of C to be used in
    determining the necessary column interchanges in the reduction;
  for j:=1 step 1 until p do
  begin h:=0;
    for i:=1 step 1 until n do h:=h+c[i,j]×c[i,j];
    sumsq[j]:=h
  end;
  comment Now determine the transformations;
  for j:=1 step 1 until p do
  begin r:=j;
    h:=sumsq[j]; m:=j;
    for k:=j+1 step 1 until p do
    if sumsq[k]>h then
    begin h:=sumsq[k];
      m:=k
    end;
    if m ≠ j then
    begin
      comment Interchange columns m and j;
      sumsq[m]:=sumsq[j];
      for i:=j step 1 until n do
```

```
      begin g:=c[i,j];
         c[i,j]:=c[i,m];
         c[i,m]:=g
      end
   end;
   comment Compute the Householder transformation necessary to
      reduce the jth column of c;
   h:=0;
   for i:=j+1 step 1 until n do
   h:=h+c[i,j]×c[i,j];
   comment If the jth column of c is already essentially reduced,
      the transformation is skipped;
   if h < tol then
   begin d[j]:=0; go to skip end;
   f:=c[j,j]; h:=h+f×f;
   g:=if f ≥ 0 then sqrt(h) else -sqrt(h);
   d[j]:=h:=h+f×g;
   c[j,j]:=f+g;
   for i:=j+1 step 1 until p do
   begin g:=0;
      for k:=j step 1 until n do
      g:=g+c[k,j]×c[k,i];
      g:=g/h;
      for k:=j step 1 until n do
      c[k,i]:=c[k,i]-g×c[k,j]
   end i;
skip:
   h:=0;
   comment Update the values in sumsq and determine the modulus of
      the largest element in the remaining matrix;
   for i:=j+1 step 1 until p do
   begin sumsq[i]:=sumsq[i]-c[j,i]×c[j,i];
      for k:=j+1 step 1 until n do
      if abs(c[k,i]) > h then h:=abs(c[k,i])
   end i;
```

```
      if h < eps then go to exit
    end j;
exit:
end reduce;

procedure apply(n) data:(r,d,c,ab) result:(gh);
value n,r; integer n,r;
array d,c,ab,gh;
comment This procedure applies r orthogonal similarity transformations
    to the symmetric matrix AB.  GH is the (n-r) x (n-r) submatrix in the
    lower right hand corner of the resulting matrix. On input,
    ab[1:n,1:n] contains the upper triangle of AB, and c[1:n,1:r] and
    d[1:r], the details of the transformations. On output,
    gh[1:n-r,1:n-r] contains the upper triangle of GH. The strict
    lower triangles of ab and gh are not used. The actual parameters
    corresponding to ab and gh may be the same;
begin integer i,j,k; real f,g,h;
    array w[1:n];
    for j:=1 step 1 until r do
    begin h:=d[j];
        if h ≠ 0 then
        begin f:=0;
            for i:=j step 1 until n do
            begin g:=0;
                for k:=j step 1 until i do g:=g+ab[k,i] x c[k,j];
                for k:=i+1 step 1 until n do g:=g+ab[i,k] x c[k,j];
                w[i]:=g:=g/h;
                f:=f+c[i,j] x g
            end i;
            f:=f/(h+h);
            for i:=j+1 step 1 until n do
            begin w[i]:=w[i]-f x c[i,j];
                for k:=j+1 step 1 until i do
                ab[k,i]:=ab[k,i]-c[i,j] x w[k]-c[k,j] x w[i]
            end i
```

```
        end conditional
    end j;
    for i:=1 step 1 until n-r do.
    for j :=i step 1 until n-r do
    gh[i,j]:=ab[i+r,j+r]
end apply;

procedure backtransform(n) data:(r,d,c,z) result:(x);
value n,r; integer n,r; array d,c,z,x;
comment This procedure applies r orthogonal transformations to the
    n x n-r matrix, the first r rows of which are zero, and the last
    n-r, the matrix Z, to produce the matrix X. On input,
    z[1:n-r,1:n-r] contains Z, and d[1:r] and c[1:n,1:r], the details
    of the transformations.  On output, x[1:n,1:n-r] contains X. The
    actual parameters corresponding to x and z may be the same;
begin real h,s;
    integer i,j,k;
    for j:=1 step 1 until n-r do
    for i:=n step_ -1 until r+1 do
    x[i,j]:=z[i-r,j];
    for k:=r step -1 until 1 do
    begin h:=d[k];
        if h ≠ 0 then
        for j:=1 step 1 until n-r do
        begin s:=0;
            for i:=k+1 step 1 until n do'
            s:=s+c[i,k] x x[i,j];
            s:=s/h
            x[k,j]:=0;
            for i:=k step 1 until n do
            x[i,j]:=x[i,j]-s x c[i,k]
        end j
    end k
end back-transform;
```

5. Organizational and Notational Details

The matrix Q defined in Section 1 is constructed in REDUCE as the produce of r Householder transformations. Using the notation in [3], we have

$$C = C^{(1)}$$

$$C^{(k+1)} = P^{(k)} C^{(k)} \quad , \quad k = 1, \ldots, r ,$$

and

$$P^{(k)} = (I - \beta_k u^{(k)} u^{(k)^T}) \quad ,$$

where

$$s_k^2 = \sum_{i=k}^{n} (c_{ik}^{(k)})^2 \quad ,$$

$$\beta_k = (s_k(s_k + |c_{kk}^{(k)}|)) \quad ,$$

$$u_1^{(k)} = 0 \quad , \quad i < k \ ,$$

$$u_k^{(k)} = \mathrm{sgn}(c_{kk}^{(k)})(s_k + |c_{kk}^{(k)}|) \quad ,$$

$$u_1^{(k)} = c_{ik}^{(k)} \quad , \quad i > k \ .$$

We have, then, that

$$Q = P^{(r)} P^{(r-1)} \ldots P^{(1)} \quad .$$

To recover the $P^{(k)}$ for use in the procedures APPLY and RACKTRANSFORM, it is necessary merely to retain the vectors $u^{(k)}$ and the values β_k .

This is done in REDUCE by storing $u^{(k)}$ in the k-th column of the array c, and by retaining β_k^{-1} in the array element d[k].

In APPLY, it is necessary to form the matrix

$$QAQ^T$$

or

$$P^{(r)}P^{(r-1)} \ldots P^{(1)}AP^{(1)} \ldots P^{(r-1)}P^{(r)}$$

(since $(P^{(k)})^T = P^{(k)}$). This is done in r steps

$$A^{(1)} = A$$

$$A^{(k+1)} = P^{(k)}A^{(k)}P^{(k)} \quad , \quad k = 1,\ldots,r \ .$$

These similarity transformations are accomplished in the manner outlined at the end of Section 1.

The procedure BACKTRANSFORM performs the transformation of the eigenvectors of the eigenproblem (1) according to (2).

The use of the parameter tol in REDUCE is discussed in [7].

The problem of determining a good value for the parameter eps in REDUCE for the purpose of determining rank is rather difficult, (cf [4]).

6. Numerical Properties

The stability of the eigensystem of a matrix with respect to similarity transformations by elementary Hermitian matrices is discussed by Wilkinson in [10].

7. Test Results

These procedures were programmed and tested on the IBM System 360/67 at the Stanford Computation Center, Stanford, California.

Long floating point arithmetic was used (14 hexadecimal-digit fraction). Inner products were not accumulated in double precision.

To provide an example of the results produced by these procedures, the following matrices were used:

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix},$$

$$B = \begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 \\ 5 & 5 & 4 & 3 & 2 & 1 \\ 4 & 4 & 4 & 3 & 2 & 1 \\ 3 & 3 & 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 1 & 8 & 5 \\ 1 & -1 & 2 & 1 \\ 1 & 1 & 8 & 5 \\ 1 & -1 & 2 & 1 \\ 1 & 1 & 8 & 5 \\ 1 & -1 & 2 & 1 \end{bmatrix}.$$

With eps = $3_{10}-14$, REDUCE correctly determined that the rank of C was 2 .

The following stationary values and vectors were then determined by finding the eigensystem of the resulting generalized eigenproblem (1):

Stationary values:

$1.70039264847579_{10}-01$

$1.23788202328080_{10}+00$

$4.91760119261002_{10}+00$

$9.27447751926161_{10}+00$

Vectors:

$2.86085382484507_{10}-01$	$-4.89644700766029_{10}-01$
$2.82124288705312_{10}-01$	$2.21020749102174_{10}-02$
$1.55676307221979_{10}-02$	$5.72549998363964_{10}-01$
$-1.09686418150406_{10}-01$	$4.49859712956573_{10}-01$
$-3.01653013206705_{10}-01$	$-8.29052975979350_{10}-02$
$-1.72437870554907_{10}-01$	$-4.71961787866790_{10}-01$
$-4.95022659856411_{10}-01$	$4.83069132908663_{10}-01$
$3.95292112932390_{10}-01$	$-9.81662635257467_{10}-01$
$7.68429013103898_{10}-01$	$5.30528981364161_{10}-01$
$-8.92878392907869_{10}-01$	$4.34008414446343_{10}-01$
$-2.73406353247487_{10}-01$	$-1.01359811427282_{10}+00$
$4.97586279975478_{10}-01$	$5.47654220811123_{10}-01$

In addition, for each vector $\underset{\sim}{x}$ above, the vector $\underset{\sim}{x}^T C$ was computed. In each case, the value of the maximum element in this vector was less in modulus than $1.1_{10}-15$.

The eigensystems of the generalized eigenproblems arising in our work were found using the procedures reduc1 and rebaka [6], tred2 [7], and tqℓ2 [1].

Acknowledgement

The authors are very pleased to acknowledge the very helpful comments of Professors P. Henrici, W. Kahan, and I. Olkin.

References

[1] H. Bowdler, R. S. Martin, C. Reinsch, and J. H. Wilkinson, "The Q,R and QL algorithms for symmetric matrices", Numerische Mathematik 11, pp. 293-306 (1968).

[2] J. Durbin and G. S. Watson, "Testing for serial correlation in least squares regression. I", Biometrika, vol. 37 (1950), pp. h-09-428.

[3] G. H. Golub, "Numericalmethods for solving linear least squares problems", Numerische Mathematik 7, pp. 206-216 (1965).

[4] G. H. Golub and W. Kahan, "Calculating the singular values and pseudoinverse of a matrix", J. SIAM. Numer. Anal., Ser. B, pp. 205-224.

[5] P. Henrici, B. A. Troesch and L. Wuytack, "Sloshing frequencies for a half-space with circular or strip-like aperture", ZAMP (to appear).

[6] R. S. Martin and J. H. Wilkinson, "Reduction of the symmetric eigenproblem Ax = λBx and related problems to standard form", Numerische Mathematik 11, pp. 99-110 (1968).

[7] R. S. Martin, C. Reinsch, and J. H. Wilkinson, "Householder% tridiagonalization of a symmetric matrix", Numerische Mathematik 11, pp. 181-195 (1968).

[8] M. R. Osborne, "A method for fitting exponentials to data", Technical Report No. 31, May, 1969, The Australian National University, Canberra, A. C. T.

[9] J. H. Wilkinson, "Householder% method for the solution of the algebraic eigenproblem", Computer J., vol. 3 (1960), 'pp. 23-27.

[10] J. H. Wilkinson, "Error analysis of transformations based on the use of matrices of the form $I-2ww^H$ ", Error in Digital Computation, vol. ii, L. B. Rall, ed., John Wiley and Sons, Inc., New York, 1965, pp. 77-101.

Keywords

stationary values

eigenvalues

(Householder-transformations)

matrices