ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XVII, 1 (1980)

ALGORITHMS 80-81

J. K. BAKSALARY, A. DOBEK and R. KALA (Poznań)

# CALCULATION OF PROJECTIONS

**1. Procedure declarations.** Given a real $(n \times p)$-matrix $A$ and a real $(n \times s)$-matrix $X$, the procedure *orproject* calculates the orthogonal (under the standard inner product) projections of the columns of $X$ on the subspace $\mathscr{C}(A)$, the column space of $A$, while the procedure *project* calculates projections also of the columns of $X$ and also on the subspace $\mathscr{C}(A)$ but without specifying the projection direction. In addition, the rank of $A$ is calculated.

The declarations of the two procedures are exactly the same.

Data:

$n$ — number of rows of $A$;
$p$ — number of columns of $A$;
$s$ — number of columns that are to be projected;
*eps* — smallest number for which $1 + eps > 1$ on the computer;
$A[1:n, 1:p]$ — array of elements of $A$;
$X[1:n, 1:s]$ — array of elements of $X$.

Results:

$r$ — rank of $A$;
$P[1:n, 1:s]$ — array formed by the column vectors being the required projections.

**2. Method used.** It is clear that the projections of the columns of $X$ on $\mathscr{C}(A)$ can be obtained by premultiplying $X$ by an appropriate projection operator which is expressible as $Q_1 = AA^+$ in the case of orthogonal projection, and as $Q_2 = AA^-$ in the case of oblique projection, where $A^+$ and $A^-$ are the Moore-Penrose inverse and a generalized inverse of $A$, respectively. A method for determining $Q_1$, which does not require the explicit calculation of $A^+$, has been given by Pyle [4] and Milliken [3], while a similar method for obtaining $Q_2$ — by Baksalary et al. [1].

The procedures of the present paper utilize the possibility (1) of calculating the projections of the columns of $X$ on $\mathscr{C}(A)$ without actually computing the matrix $Q_1$ or $Q_2$. Precisely, the procedure *orproject* uses the formulae

$$A_1 = A, \quad P_1 = 0,$$
$$A_{i+1} = (I - u_i u_i^+) U_i, \quad i = 1, \dots, p-1,$$
$$P_{i+1} = P_i + u_i u_i^+ X, \quad i = 1, \dots, p,$$

where, in each step, $u_i$ is the first column of the $[n \times (p-i+1)]$-matrix $A_i$, and $U_i$ consists of the remaining columns of $A_i$. The required projections are obtained as the columns of $P_{p+1}$.

The procedure *project* is based on similar formulae, namely

$$A_1 = A, \quad X_1 = X, \quad P_1 = 0,$$
$$A_{i+1} = (I - u_i u_i^-) U_i, \quad i = 1, \dots, p-1,$$
$$X_{i+1} = (I - u_i u_i^-) X_i, \quad i = 1, \dots, p-1,$$
$$P_{i+1} = P_i + u_i u_i^- X_i, \quad i = 1, \dots, p,$$

where $u_i$ and $U_i$ are related to the same partition of $A_i$ as described for the procedure *orproject*.

**3. Certification.** The procedures proposed have been tested on the ODRA 1204 computer which has a 37 binary digit mantissa. The projections of the columns of the matrices $H_n^* = (2n-1)!H_n$, where $H_n$ stands for the $n$-th leading principal minor of the Hilbert matrix, have been calculated for $n = 3, \dots, 8$ using three different methods. The last two of them (denoted by II and III in Table 1) use the procedures *orproject* and *project*, respectively, whereas the first method (denoted by I) consists in premultiplying $H_n^*$ by the operator provided by the procedure *ORPROPCOL* (see [2]). For every column of all tested matrices, Table 1 gives the values

$$rd_j = \max_{1 \leqslant i \leqslant n} |(\tilde{a}_{ij} - a_{ij})/a_{ij}|,$$

where $\tilde{a}_{ij}$ is the $i$-th element of the projection of the $j$-th column, and $a_{ij}$ is the $(i, j)$-th element of $H_n^*$. For all the methods here compared, the values $rd_j$ $(j = 1, \dots, n)$ are expected to be zeros.

It should be noted that fully comparable are the values $rd_j$ for methods I and II only, as they both provide the orthogonal projections. However, the values $rd_j$ for the method III compared with those for the method II indicate the advantage of the procedure *project* over the procedure *orproject*. Thus, the former should be recommended whenever the orthogonality of projections is not required.

---

(1) The authors are grateful to Dr. A. Otten from the Agricultural University at Wageningen for suggesting this possibility.

```
procedure orproject(n,p,s,eps,A,X,r,P);
 value n,p,s;
 integer n,p,s,r;
 real eps;
 array A,X,P;
 begin
  integer i,j,l;
  real x,y;
  array N[1:p],H[1:p+s];
  r:=0;
  eps:=eps↑2;
  for i:=1 step 1 until n do
   for j:=1 step 1 until s do
    P[i,j]:=.0;
  for j:=1 step 1 until p do
   begin
    x:=.0;
    for i:=1 step 1 until n do
     x:=x+A[i,j]↑2;
    N[j]:=x
   end j;
  for l:=1 step 1 until p do
   begin
    x:=.0;
    for i:=1 step 1 until n do
     x:=x+A[i,l]↑2;
    if x>eps×N[l]
     then
     begin
      r:=r+1;
```

```
        for j:=1 step 1 until s do
         begin
          y:=.0;
          for i:=1 step 1 until n do
           y:=y+X[i,j]×A[i,l];
          H[j]:=y
         end j;
        for j:=l+1 step 1 until p do
         begin
          y:=.0;
          for i:=1 step 1 until n do
           y:=y+A[i,j]×A[i,l];
          H[j+s]:=y
         end j;
        for i:=1 step 1 until n do
         begin
          y:=A[i,l]/x;
          for j:=1 step 1 until s do
           P[i,j]:=P[i,j]+H[j]×y;
          for j:=l+1 step 1 until p do
           A[i,j]:=A[i,j]-H[j+s]×y
         end i
      end x>eps×N[l]
    end l
  end orproject
```

```
procedure project(n,p,s,eps,A,X,r,P);
 value n,p,s;
 integer n,p,s,r;
 real eps;
 array A,X,P;
 begin
  integer i,j,l,pi;
  real x,y;
  array N[1:p];
  r:=0;
  for i:=1 step 1 until n do
   for j:=1 step 1 until s do
    P[i,j]:=.0;
  for j:=1 step 1 until p do
   begin
    x:=abs(A[1,j]);
    for i:=2 step 1 until n do
     begin
      y:=abs(A[i,j]);
      if y>x
       then x:=y
     end i;
    N[j]:=x
   end j;
  for l:=1 step 1 until p do
   begin
    x:=abs(A[1,l]);
    pi:=1;
    for j:=2 step 1 until n do
     begin
```

```
        y:=abs(A[j,l]);
        if y>x
         then
         begin
          x:=y;
          pi:=j
         end
       end j;
      if x>eps×N[l]
       then
       begin
        r:=r+1;
        x:=A[pi,l];
        for i:=1 step 1 until n do
         begin
          y:=A[i,l]:=-A[i,l]/x;
          for j:=1 step 1 until s do
           P[i,j]:=P[i,j]-y×X[pi,j]
         end i;
        for i:=1 step 1 until pi-1,pi+1 step 1 until n do
         begin
          y:=A[i,l];
          for j:=l+1 step 1 until p do
           A[i,j]:=A[i,j]+y×A[pi,j];
          for j:=1 step 1 until s do
           X[i,j]:=X[i,j]+y×X[pi,j]
         end i;
        for j:=l+1 step 1 until p do
         A[pi,j]:=.0;
        for j:=1 step 1 until s do
         X[pi,j]:=.0
       end x>eps×N[l]
     end l
   end project
```

TABLE 1

| Method | $rd_1$ | $rd_2$ | $rd_3$ | $rd_4$ | $rd_5$ | $rd_6$ | $rd_7$ | $rd_8$ |
|---|---|---|---|---|---|---|---|---|
| I | $268_{10}-12$ | $380_{10}-12$ | $417_{10}-12$ | | | | | |
| II | $501_{10}-12$ | $489_{10}-12$ | $475_{10}-12$ | | | | | |
| III | $776_{10}-14$ | $116_{10}-13$ | $776_{10}-14$ | | | | | |
| I | $117_{10}-09$ | $899_{10}-10$ | $808_{10}-10$ | $761_{10}-10$ | | | | |
| II | $261_{10}-10$ | $233_{10}-10$ | $223_{10}-10$ | $217_{10}-10$ | | | | |
| III | $118_{10}-13$ | $118_{10}-13$ | $118_{10}-13$ | $103_{10}-13$ | | | | |
| I | $171_{10}-08$ | $150_{10}-08$ | $141_{10}-08$ | $135_{10}-08$ | $132_{10}-08$ | | | |
| II | $146_{10}-08$ | $110_{10}-08$ | $967_{10}-09$ | $899_{10}-09$ | $857_{10}-09$ | | | |
| III | $131_{10}-13$ | $131_{10}-13$ | $131_{10}-13$ | $131_{10}-13$ | $118_{10}-13$ | | | |
| I | $856_{10}-05$ | $642_{10}-05$ | $561_{10}-05$ | $517_{10}-05$ | $489_{10}-05$ | $470_{10}-05$ | | |
| II | $636_{10}-06$ | $735_{10}-06$ | $745_{10}-06$ | $741_{10}-06$ | $736_{10}-06$ | $731_{10}-06$ | | |
| III | $144_{10}-13$ | $144_{10}-13$ | $144_{10}-13$ | $120_{10}-13$ | $144_{10}-13$ | $132_{10}-13$ | | |
| I | $298_{10}-07$ | $207_{10}-07$ | $175_{10}-07$ | $160_{10}-07$ | $150_{10}-07$ | $144_{10}-07$ | $139_{10}-07$ | |
| II | $351_{10}-07$ | $199_{10}-07$ | $148_{10}-07$ | $127_{10}-07$ | $118_{10}-07$ | $111_{10}-07$ | $107_{10}-07$ | |
| III | $143_{10}-13$ | $143_{10}-13$ | $143_{10}-13$ | $143_{10}-13$ | $143_{10}-13$ | $428_{10}-13$ | $257_{10}-13$ | |
| I | $367_{10}-05$ | $206_{10}-05$ | $152_{10}-05$ | $125_{10}-05$ | $108_{10}-05$ | $972_{10}-06$ | $894_{10}-06$ | $835_{10}-06$ |
| II | $465_{10}-06$ | $179_{10}-06$ | $883_{10}-07$ | $448_{10}-07$ | $255_{10}-07$ | $156_{10}-07$ | $104_{10}-07$ | $165_{10}-07$ |
| III | $137_{10}-13$ | $137_{10}-13$ | $137_{10}-13$ | $137_{10}-13$ | $137_{10}-13$ | $351_{10}-13$ | $137_{10}-13$ | $137_{10}-13$ |

## References


[1] J. K. Baksalary, A. Dobek and R. Kala, *A method for computing projectors*, Ž. Vyčisl. Mat. i Mat. Fiz. 16 (1976), p. 1038-1040.

[2] — *Wyznaczanie operatorów rzutowania ortogonalnego*, Alg. Biom. Statyst. 5 (1976), p. 187-194.

[3] G. A. Milliken, *New criteria for estimability for linear models*, Ann. Math. Statist. 42 (1971), p. 1588-1594.

[4] L. D. Pyle, *Generalized inverse computations using gradient projection method*, J. Assoc. Comput. Mach. 11 (1964), p. 422-428.



DEPARTMENT OF MATHEMATICAL AND STATISTICAL METHODS
ACADEMY OF AGRICULTURE
60-637 POZNAŃ


*Received on 16. 12. 1977*

---

**ALGORYTMY 80-81**

**J. K. BAKSALARY, A. DOBEK i R. KALA (Poznań)**

## WYZNACZANIE RZUTÓW

STRESZCZENIE


Dla danej $(n \times p)$-wymiarowej macierzy rzeczywistej $A$ oraz $(n \times s)$-wymiarowej macierzy rzeczywistej $X$ procedura *orproject* wyznacza rzuty ortogonalne (w sensie standardowego iloczynu skalarnego) kolumn macierzy $X$ na podprzestrzeń $\mathscr{C}(A)$ generowaną przez kolumny macierzy $A$, natomiast procedura *project* wyznacza rzuty tych samych wektorów na tę samą podprzestrzeń, lecz bez określenia kierunku rzutowania.