ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XVI, 2 (1978)

ALGORITHM 61

Z. KASPERSKI (Opole)

THE SOLUTION OF A CERTAIN LINEAR EQUATION SYSTEM

1. Procedure declaration. Procedure *URP* solves a group of *lp* systems of linear equations with *lr* unknowns and with various right-hand members

(1) $$KX = T_r \quad (r = 1, 2, \ldots, lp),$$

where the global matrix K is treated as composed of submatrices $K_1, K_2, \ldots, K_{lm}$ of dimension $n \times n$ (n — an even number) in form as shown in Fig. 1. The marked elements are summarized.

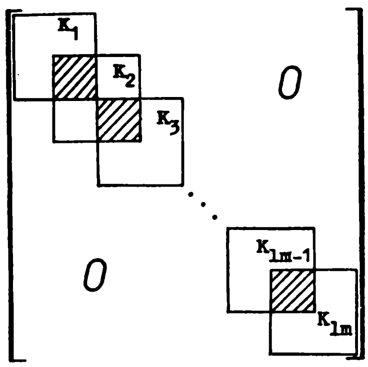

Fig. 1

The number of unknowns is $lr = .5 \times n \times (lm+1)$. It is easy to show that the indices of the elements k_{ij} (shown as 0 in Fig. 1) in the lower part of the matrix K satisfy the relation

$$1 \leqslant j \leqslant \frac{n}{2}\left(\left[\frac{i-1}{n/2}\right]-1\right) \quad (i = n+1, n+2, \ldots, lr),$$

while the indices of the zero elements in the upper part of the matrix K satisfy the relation

$$n+\left[\frac{i-1}{n/2}\right]\frac{n}{2} < j \leqslant lr \quad (i = 1, 2, \ldots, lr-n),$$

where the symbol $[\cdot]$ denotes the integer part. It is a special type of strip matrix. Such systems are found in many technical problems, usually with a great number of unknowns (for example, in shell analysis). The presented algorithm is more effective (mainly due to its storage-saving properties) than those treating the matrix K as of common strip-type.

Data:

n — degree of submatrices K_i (n must be even);

lm — number of submatrices K_i;

lp — number of systems processed;

$T[1\colon .5\times n\times(lm+1),\ 1\colon lp]$ — array of right-hand side of system (1) ($T[i, j]$ is the i-th free member of the j-th system of equations);

CMK — procedure with the following heading: **procedure** $CMK(i, n, K)$; **integer** i, n; **array** K; the procedure determines the elements of a submatrix K_i for $i = 1, 2, \ldots, lm$.

Results:

$T[1\colon .5\times n\times(lm+1),\ 1\colon lp]$ — array of results for system (1) ($T[i,j]$ is the i-th unknown for the j-th system).

Other parameters:

exit — label to which the program is switched in case of singularity of the matrix K.

2. Description of the method. We assume that the element k^i_{rs} is placed at the intersection of the row r and the column s of the matrix K_i ($p = n/2$). We analyze a part of the matrix K composed of submatrices K_i and K_{i+1} ($i = 1, 2, \ldots, lm-1$):

$$
\begin{array}{cccccccccc}
k^i_{11} & k^i_{12} & \ldots & k^i_{1p} & k^i_{1,p+1} & \ldots & k^i_{1n} & & & \\
k^i_{21} & k^i_{22} & \ldots & k^i_{2p} & k^i_{2,p+1} & \ldots & k^i_{2n} & & & \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & 0 & \\
k^i_{p1} & k^i_{p2} & \ldots & k^i_{pp} & k^i_{p,p+1} & \ldots & k^i_{pn} & & & \\
k^i_{p+1,1} & k^i_{p+1,2} & \ldots & k^i_{p+1,p} & d^i_{11} & \ldots & d^i_{1p} & k^{i+1}_{1,p+1} & \ldots & k^{i+1}_{1n} \\
k^i_{p+2,1} & k^i_{p+2,2} & \ldots & k^i_{p+2,p} & d^i_{21} & \ldots & d^i_{2p} & k^{i+1}_{2,p+1} & \ldots & k^{i+1}_{2n} \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
k^i_{n1} & k^i_{n2} & \ldots & k^i_{np} & d^i_{p1} & \ldots & d^i_{pp} & k^{i+1}_{p,p+1} & \ldots & k^{i+1}_{pn} \\
 & & & & k^{i+1}_{p+1,1} & \ldots & k^{i+1}_{p+1,p} & k^{i+1}_{p+1,p+1} & \ldots & k^{i+1}_{p+1,n} \\
 & & 0 & & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & & & & k^{i+1}_{n1} & \ldots & k^{i+1}_{np} & k^{i+1}_{n,p+1} & \ldots & k^{i+1}_{nn}
\end{array}
$$

where $d^i_{tq} = k^{i+1}_{tq} + k^i_{t+p,q+p}$ $(t, q = 1, 2, \ldots, p)$.

```
procedure URP(n,lm,lp,T,CMK,exit);
 value n,lm,lp;
 integer n,lm,lp;
 array T;
 procedure CMK;
 label exit;
 begin
  integer lr,lm1,np,t1,k2,r1,i,j,h,i1,j1,m,pi,pj,l;
  real g,w;
  np:=n+2;
  lr:=np×(lm+1);
  l:=n+lp;
  begin
   array A[1:lr,1:np],B[1:n,1:l],K[1:n,1:n];
   integer array P[1:l];
   t1:=k2:=np;
   r1:=0;
   CMK(1,n,K);
   for i:=1 step 1 until np do
    begin
     i1:=np+i;
     for j:=1 step 1 until np do
      B[i1,np+j]:=K[i,j];
     for j:=1 step 1 until lp do
      B[i1,n+j]:=T[i,j]
    end i;
   for i:=1 step 1 until l do
    P[i]:=i;
   lm1:=lm+1;
   for h:=2 step 1 until lm1 do
```

```
begin
 for i:=1 step 1 until np do
  begin
   i1:=np+i;
   for j:=1 step 1 until np do
    begin
     j1:=np+j;
     B[i,j]:=B[i1,j1];
     B[i,j1]:=K[i,j1];
     B[i1,j]:=K[i1,j];
     B[i1,j1]:=K[i1,j1];
    end j;
   for j:=1 step 1 until lp do
    begin
     j1:=n+j;
     B[i,j1]:=B[i1,j1];
     B[i1,j1]:=T[t1+i,j];
    end j
  end i;
 t1:=t1+np;
 if h=lm+1
  then
  begin
   k2:=n;
   go to E1
  end;
 CMK(h,n,K);
for i:=1 step 1 until np do
 begin
   i1:=np+i;
```

```
      for j:=1 step 1 until np do
       B[i1,np+j]:=B[i1,np+j]+K[i,j]
     end i;
E1:  for i:=1 step 1 until k2 do
      begin
       g:=.0;
       for j:=i step 1 until k2 do
        begin
         w:=abs(B[i,P[j]]);
         if w>g
          then
          begin
           g:=w;
           m:=j
          end w>g
        end j;
       if g=0
        then go to exit;
       pi:=P[m];
       P[m]:=P[i];
       P[i]:=pi;
       g:=1.0/B[i,pi];
       for j:=i+1 step 1 until 1 do
        begin
         pj:=P[j];
         w:=B[i,pj]:=B[i,pj]×g;
         for m:=1 step 1 until i-1,i+1 step 1 until n do
          B[m,pj]:=B[m,pj]-B[m,pi]×w
        end j
      end i;
```

```
        for i:=1 step 1 until k2 do
         begin
          i1:=P[i]+r1;
          for j:=np+1 step 1 until n do
           A[i1,j-np]:=B[i,j];
          for j:=1 step 1 until lp do
           T[i1,j]:=B[i,n+j]
         end i;
        r1:=r1+np
       end h;
      k2:=lr-n;
      r1:=0;
      for i:=k2 step -1 until 1 do
       begin
        for h:=1 step 1 until lp do
         begin
          g:=0;
          for j:=1 step 1 until np do
           g:=g+A[i,j]×T[i+r1+j,h];
          T[i,h]:=T[i,h]-g
         end h;
        r1:=r1+1;
        if r1<np
         then go to E;
        r1:=0;
E:     end i
     end
    end URP
```

Carrying out a proper elimination of the first p unknowns of this subsystem we obtain

$$\begin{array}{ccccccccccc}
1 & 0 & \ldots & 0 & k^{i*}_{1,p+1} & \ldots & k^{i*}_{1n} & & & & \\
0 & 1 & \ldots & 0 & k^{i*}_{2,p+1} & \ldots & k^{i*}_{2n} & & 0 & & \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & & & \\
0 & 0 & \ldots & 1 & k^{i*}_{p,p+1} & \ldots & k^{i*}_{pn} & & & & \\
0 & 0 & \ldots & 0 & d^{i*}_{11} & \ldots & d^{i*}_{1p} & k^{i+1}_{1,p+1} & \ldots & k^{i+1}_{1n} & \\
0 & 0 & \ldots & 0 & d^{i*}_{21} & \ldots & d^{i*}_{2p} & k^{i+1}_{2,p+1} & \ldots & k^{i+1}_{2n} & \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\
0 & 0 & \ldots & 0 & d^{i*}_{p1} & \ldots & d^{i*}_{pp} & k^{i+1}_{p,p+1} & \ldots & k^{i+1}_{pn} & \\
 & & & & k^{i+1}_{p+1,1} & \ldots & k^{i+1}_{p+1,p} & k^{i+1}_{p+1,p+1} & \ldots & k^{i+1}_{p+1,n} & \\
 & & 0 & & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\
 & & & & k^{i+1}_{n1} & \ldots & k^{i+1}_{np} & k^{i+1}_{n,p+1} & \ldots & k^{i+1}_{nn} &
\end{array}$$

where all elements are unchanged except those signed with *. We apply the same procedure for submatrices $K_1, K_2, \ldots, K_{lm}$. In the reverse procedure only elements k^{i*}_{tq} $(t = 1, 2, \ldots, p;\ q = p+1, p+2, \ldots, n)$ are necessary for calculation of the corresponding unknowns of the i-th step.

Conclusion. In the i-th step of the initial procedure only submatrices K_i and K_{i+1} are processed. In the i-th step of the reverse procedure only the matrix $[k_{rs}]$ of dimension $p \times p$ is processed.

In the procedure, the elimination process is carried out with selection of the main element.

3. Certification. Procedure *URP* was tested for some numerical examples. In all cases the results were the same as for the well-known Gauss procedure. The calculations were performed on the ODRA 1204 computer.

Remark. The assumption that the marked elements in Fig. 1 are summarized does not limit the application of the algorithm. If needed, the appropriate matrix elements can be set to be equal to zero.

COMPUTING CENTRE
HIGHER TECHNICAL SCHOOL OF OPOLE
55-950 OPOLE

*Received on 6. 1. 1976;
revised version on 10. 1. 1977*

ALGORYTM 61

Z. KASPERSKI (Opole)

ROZWIĄZANIE PEWNEGO UKŁADU RÓWNAŃ LINIOWYCH

STRESZCZENIE

Procedura *URP* rozwiązuje grupę układów równań liniowych (1) o wspólnej macierzy układu K, zbudowanej z podmacierzy kwadratowych $K_1, K_2, \ldots, K_{lm}$ o wymiarach $n \times n$ tak, jak to pokazano na rys. 1. Elementy zakreskowane na rysunku są dodawane, co nie stanowi istotnego ograniczenia w stosowaniu procedury. Jest to specjalny typ macierzy pasmowej. Tego typu układy występują w zastosowaniach technicznych (np. przy analizie powłok obrotowych) na ogół z dużą liczbą niewiadomych. Przedstawiony algorytm jest efektywniejszy (głównie ze względu na wykorzystanie pamięci maszyny cyfrowej) niż analogiczne algorytmy, traktujące K jako macierz pasmową.

Dane:

n — stopień podmacierzy K_i (n musi być liczbą parzystą);
lm — liczba podmacierzy K_i;
lp — liczba rozwiązywanych układów;
$T[1:.5\times n\times(lm+1), 1:lp]$ — tablica prawych stron układu (1) ($T[i,j]$ jest prawą stroną w i-tym równaniu j-tego układu);
CMK — procedura o nagłówku: **procedure** $CMK(i, n, K)$; **integer** i, n; **array** K; procedura oblicza elementy podmacierzy K_i dla $i = 1, 2, \ldots, lm$.

Wyniki:

$T[1:.5\times n\times(lm+1), 1:lp]$ — tablica rozwiązań układu (1) ($T[i,j]$ jest i-tą niewiadomą j-tego układu równań).

Inne parametry:

exit — etykieta poza treścią procedury, do której następuje skok, gdy macierz układu K jest osobliwa.