Computers & Structures Vol. 36, No. 6, pp. 1009–1012, 1990
Printed in Great Britain.

0045-7949/90 $3.00 + 0.00
Pergamon Press plc

TREATMENT OF SKEW BOUNDARY CONDITIONS IN FINITE ELEMENT ANALYSIS

D. V. GRIFFITHS
Department of Engineering, University of Manchester, Manchester M13 9PL, U.K.

(*Received* 6 *September* 1989)

Abstract—Finite element analyses in both structural and geotechnical applications sometimes require the freedom directions at certain nodes to differ from the global Cartesian directions. Examples of this include modelling of oblique interface behaviour and implementation of 'skew' boundary conditions. A simple method is described for performing the transformations at the element level, and a FORTRAN 77 subroutine is included to perform the operations.

1. INTRODUCTION

The boundary conditions incorporated in finite element analyses in both structural and geotechnical applications often require 'rollers' in the x- or y-directions (in two dimensions). These particular conditions usually imply smooth boundaries or lines of symmetry and are a standard feature in the majority of finite element software.

This paper is concerned with cases in which the freedom directions at certain nodes are to differ from the global Cartesian coordinate directions. Analyses where this is necessary include the implementation of 'skew' boundary conditions, such as the inclined 'roller' shown in Fig. 1. Another case is shown in Fig. 2 where a smooth interface is to be modelled between a stiff wedge and a soil stratum. In this case, the freedoms at the nodes along the interface are transformed so that they are parallel and perpendicular to the interface direction. By uncoupling the freedoms in the parallel directions on each side of the interface, smooth conditions can be reproduced.

In order to achieve these transformations, the stiffness matrix of the system must be modified to account for the new freedom directions at the required nodes. These operations can be performed at either the global or local levels. In this paper, the local level is chosen because it is more simple and more general. If the transformations are performed at the global level, the same principles will apply but account will have to be taken of the storage strategy (e.g. banding, skyline).

The present work is limited to two-dimensional analyses, but will be relevant to any applications using planar, axisymmetric, beam–column or truss elements (see e.g. [1]).

2. ELEMENT STIFFNESS MATRIX TRANSFORMATION

In order to develop the equations, the specific example of a four-node planar element is considered. In a regular two-dimensional analysis, this element has eight degrees of freedom (four x- and four y-displacements) and would yield an 8 by 8 element stiffness matrix. Figure 3a shows a four-node element with its local nodal numbering system which counts in a clockwise sense. The stiffness relationship for this element would have the form of eqn (1)

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} & k_{17} & k_{18} \\ & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} & k_{27} & k_{28} \\ & & k_{33} & k_{34} & k_{35} & k_{36} & k_{37} & k_{38} \\ & & & k_{44} & k_{45} & k_{46} & k_{47} & k_{48} \\ & & & & k_{55} & k_{56} & k_{57} & k_{58} \\ & \text{Symm.} & & & & k_{66} & k_{67} & k_{68} \\ & & & & & & k_{77} & k_{78} \\ & & & & & & & k_{88} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} = \begin{Bmatrix} F_{u1} \\ F_{v1} \\ F_{u2} \\ F_{v2} \\ F_{u3} \\ F_{v3} \\ F_{u4} \\ F_{v4} \end{Bmatrix}, \qquad (1)$$

where

k_{ij} = stiffness coefficients
F_{u1} = force in the x-direction at node 1
F_{v2} = force in the y-direction at node 2
u_1 = displacement in the x-direction at node 1
v_2 = displacement in the y-direction at node 2, etc.

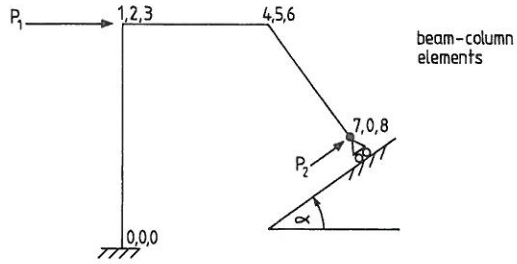

Fig. 1. Skew boundary condition.

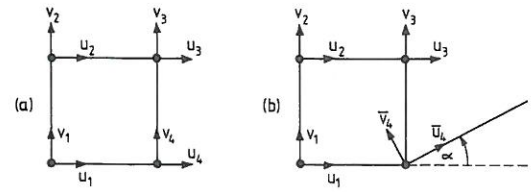

Fig. 3. Conventional (a) and modified (b) displacement directions.

Equation (1) relates forces to displacements at each node resolved into x- and y-components; however, the actual nodal force or displacement vector could be resolved in any two orthogonal components.

Consider (Fig. 3b) the same four-noded element, but with the displacement at node 4 resolved into inclined x'- and y'-components as shown. The notation used here signifies transformed displacements and forces with a bar over the symbol, e.g. $\bar{u}_4$, $\bar{F}_{u4}$, etc. Equation (1) may now be rewritten in the form

expressed in terms of the modified components $(\bar{u}_4, \bar{v}_4)$ and the transformation angle α.

From Fig. 4, it can be seen that transformed and untransformed displacement components are related through the expressions

$$u = \bar{u}\cos\alpha - \bar{v}\sin\alpha$$
$$v = \bar{u}\sin\alpha + \bar{v}\cos\alpha \qquad (3)$$

with a sign convention given by the arrowheads, which always point in the positive direction.

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} & k'_{17} & k'_{18} \\ & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} & k'_{27} & k'_{28} \\ & & k_{33} & k_{34} & k_{35} & k_{36} & k'_{37} & k'_{38} \\ & & & k_{44} & k_{45} & k_{46} & k'_{47} & k'_{48} \\ & & & & k_{55} & k_{56} & k'_{57} & k'_{58} \\ & \text{Symm.} & & & & k_{66} & k'_{67} & k'_{68} \\ & & & & & & k'_{77} & k'_{78} \\ & & & & & & & k'_{88} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ \bar{u}_4 \\ \bar{v}_4 \end{Bmatrix} = \begin{Bmatrix} F_{u1} \\ F_{v1} \\ F_{u2} \\ F_{v2} \\ F_{u3} \\ F_{v3} \\ \bar{F}_{u4} \\ \bar{F}_{v4} \end{Bmatrix}, \qquad (2)$$

where

k'_{ij} = modified stiffness coefficients
$\bar{F}_{u4}$ = force in the x'-direction at node 4
$\bar{F}_{v4}$ = force in the y'-direction at node 4, etc.

To compute the modified stiffness coefficients, the original displacement components (u_4, v_4) must be

By comparing eqns (1) and (2), and noting that a similar relationship to that given by eqns (3) exists for forces, it is readily shown that the transformed stiffness terms are given by

$$\left.\begin{aligned} k'_{i7} = k'_{7i} &= +k_{i7}\cos\alpha + k_{i8}\sin\alpha \\ k'_{i8} = k'_{8i} &= -k_{i7}\sin\alpha + k_{i8}\cos\alpha \end{aligned}\right\} \quad i = 1, 2, \ldots 6 \qquad (4)$$

and

$$k'_{78} = k'_{87} = \sin\alpha\cos\alpha(k_{88} - k_{77}) + k_{78}\cos 2\alpha \qquad (5)$$

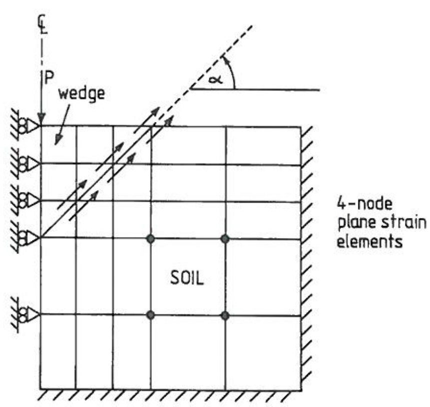

Fig. 2. Modelling of smooth inclined interface.

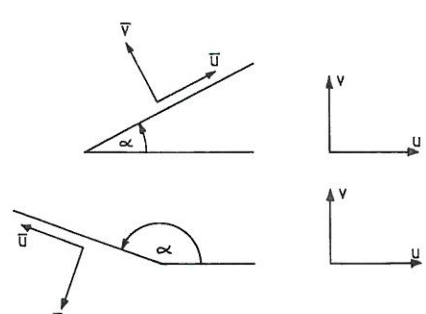

Fig. 4. Sign convention for nodal displacements and rotation.

$$k'_{77} = k_{77}\cos^2\alpha + k_{78}\sin 2\alpha + k_{88}\sin^2\alpha \quad (6)$$

$$k'_{88} = k_{77}\sin^2\alpha - k_{78}\sin 2\alpha + k_{88}\cos^2\alpha. \quad (7)$$

3. GENERAL FORMULATION

The example given above was for the specific case where the 7th and 8th displacements were transformed. In general we may wish the transformation to apply to other nodes, or even several nodes in the element.

Consider a finite element with n degrees of freedom in which consecutive freedoms k and l are to be rotated by a given angle α. The terms in the stiffness relationship that will be affected by this transformation are enclosed by the lines in eqn (8)

$$\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1k} & k_{1l} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2k} & k_{2l} & \cdots & k_{2n} \\ \vdots & & & & & & \vdots \\ k_{k1} & k_{k2} & \cdots & k_{kk} & k_{kl} & \cdots & k_{kn} \\ k_{l1} & k_{l2} & \cdots & k_{lk} & k_{ll} & \cdots & k_{ln} \\ \vdots & & & & & & \vdots \\ k_{n1} & k_{n2} & \cdots & k_{nk} & k_{nl} & \cdots & k_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \\ u_l \\ \vdots \\ u_n \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ \vdots \\ F_k \\ F_l \\ \vdots \\ F_n \end{Bmatrix}. \quad (8)$$

A general statement of the transformed stiffness terms can be written as

$$\left.\begin{aligned} k'_{ik} = k'_{ki} &= +k_{ik}\cos\alpha + k_{il}\sin\alpha \\ k'_{il} = k'_{li} &= -k_{ik}\sin\alpha + k_{il}\cos\alpha \end{aligned}\right\} \quad \begin{aligned} &i = 1, 2 \ldots n, \\ &i \neq k, i \neq l \end{aligned} \quad (9)$$

$$k'_{kk} = k_{kk}\cos^2\alpha + k_{kl}\sin 2\alpha + k_{ll}\sin^2\alpha \quad (10)$$

$$k'_{kl} = k'_{lk} = \sin\alpha\cos\alpha(k_{ll} - k_{kk}) + k_{kl}\cos 2\alpha \quad (11)$$

$$k'_{ll} = k_{kk}\sin^2\alpha - k_{kl}\sin 2\alpha + k_{ll}\cos^2\alpha. \quad (12)$$

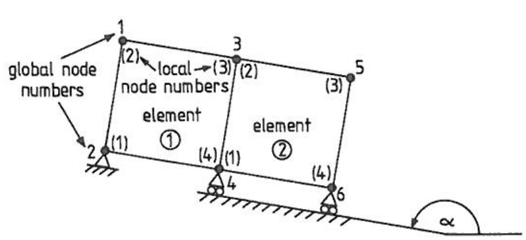

Fig. 5. Transformed node attached to more than one element.

4. ASSEMBLY OF TRANSFORMED ELEMENT MATRICES

Figure 5 shows two attached elements requiring skew boundary conditions with an inclination of α at nodes 4 and 6 in the global numbering system. Local level transformations are therefore required to node 4 in element 1, and to nodes 1 and 4 in element 2.

Figure 6 shows an element with skew boundary conditions on two sides. In this case the global and local node numbers are the same, so 30° transformations are required to both nodes 2 and 4. In cases where more than one node in an element requires transformation, it is not important in which order this is done.

After transformation of the element matrices, they are then assembled into the global stiffness matrix. If a skew boundary condition is required, it can be treated in exactly the same way as oridinary 'rollers' in the x- or y-directions, namely the terms corresponding to the zero displacement boundary condition are ignored in the assembly process. After solution of the equilibrium equations, the remaining non-zero displacement obtained at the skew 'roller' represents the distance moved up (or down) the incline.

5. TRANSFORMATION SUBROUTINE

The subroutine SKEW takes the untransformed element stiffness matrix and modifies those terms

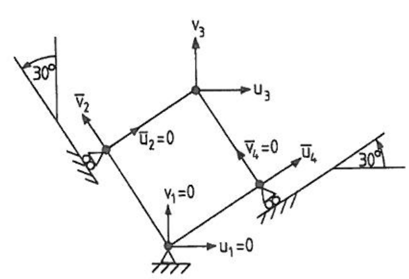

Fig. 6. Element with two nodal transformations.

affected by a displacement rotation at a given node through a given angle. The coding follows the derivations of eqns (9)–(12), and deals with one node at a time. Hence, if more than one node in a particular element requires transformation, the subroutine must be called a corresponding number of times.

Subroutine arguments:

KM	untransformed element stiffness matrix on input, overwritten by transformed matrix on output.
IKM	number of rows of KM as declared at the top of the main program.
NOD	number of nodes in the element.
NODOF	number of degrees of freedom per node.
NS	node number (at the element level) requiring transformation.
ALP	angle of rotation in degrees $0 \leqslant \alpha \leqslant 180°$.

6. CONCLUDING REMARKS

A simple method for transforming freedom directions at the element level has been described, and a FORTRAN 77 subroutine to perform the operations presented. The subroutine can be easily incorporated in two-dimensional finite element programs which seek to model skew boundary conditions or inclined interface behaviour. The subroutine is directly applicable to structural analysis involving truss or beam–column elements, and to any planar or axisymmetric analysis in solid mechanics.

REFERENCE

1. I. M. Smith and D. V. Griffiths, *Programming the Finite Element Method*, 2nd edn. John Wiley, Chichester (1988).

```
      SUBROUTINE SKEW(KM,IKM,NOD,NODOF,NS,ALP)
C
C      THIS SUBROUTINE TRANSFORMS THE ELEMENT STIFFNESS MATRIX
C      TO ACCOUNT FOR FREEDOM ROTATIONS
C
      REAL KM(IKM,*)
      PI=4.*ATAN(1.)
      XYZ=ALP*PI/180.
      CALP=COS(XYZ)
      SALP=SIN(XYZ)
      N1=(NS-1)*NODOF+1
      N2=N1+1
      IDOF=NOD*NODOF
      DO 1 I=1,IDOF
      IF(I.EQ.N1.OR.I.EQ.N2)GOTO 1
      KM(I,N1)=KM(I,N1)*CALP+KM(I,N2)*SALP
      KM(I,N2)=-KM(N1,I)*SALP+KM(I,N2)*CALP
      KM(N1,I)=KM(I,N1)
      KM(N2,I)=KM(I,N2)
    1 CONTINUE
      Z1=KM(N1,N1)
      Z2=KM(N1,N2)
      Z3=KM(N2,N2)
      Z4=2.*Z2*SALP*CALP
      KM(N1,N1)=Z1*CALP*CALP+Z4+Z3*SALP*SALP
      KM(N1,N2)=SALP*CALP*(-Z1+Z3)+Z2*(CALP*CALP-SALP*SALP)
      KM(N2,N2)=Z1*SALP*SALP-Z4+Z3*CALP*CALP
      KM(N2,N1)=KM(N1,N2)
      RETURN
      END
```