International Journal of Mining and Geo-Engineering

IJMGE

IJMGE 52-2 (2018) 119-123

DOI: 10.22059/IJMGE.2018.241271.594697

Computational and Programming Aspects of Transition Elements in a Three-dimensional Finite Element Program

Amin Chamani [a, *], Vamegh Rasouli [b]

[a] *Department of Mining Engineering, Faculty of Engineering, University of Birjand, Birjand, Iran*

[b] *Department of Petroleum Engineering, University of North Dakota, ND, USA*

Article History:

Received: 13 September 2017,

Revised: 22 July 2018,

Accepted: 24 July 2018.

ABSTRACT

The performance of any finite element (FE) structural analysis is directly related to the global number of nodes and degrees of freedom (DOF) of the discretized structure and mesh distribution attributes. It is obvious that the appropriate numerical analysis needs finer elements in the zone of interest, e.g. zone of high stress concentration and intensity, and coarser elements for farther portion of the structure. The transition element concept achieves this aim and with variable number of nodes of each element in the transition zone, it creates coarser elements in the outward zones of the discretized structure. These elements have larger size with variable number of nodes per element and their number of nodes is between the number of nodes per elements of the inner and outer zones. Despite the fact that the concept of transition element is not that new and dates back to the last few decades, but to the best knowledge of authors, an obvious and clear programming strategy and the method of implementation in a FE program have not been depicted in particular in the related literature. In this study, the main concept of transition element is completely presented with clear instances and the computational methodology of accounting for this subject is covered. Afterward, the programming strategy of the transition elements in a three-dimensional computer program of finite element method, together with the related computer program parts in FORTRAN programming language, are brought. Finally, a validating example is considered and the analogy between the results of the finite element program and analytical anticipation is made.

Keywords : *Finite element method, Programming aspects, Transition elements, Validating example*

1. Introduction

One of the main concepts of computational solid mechanics, especially Finite Element Method (FEM), is the mesh properties. Due to the computational cost related directly to the number of nodes per element, one is interested in having larger elements moving farther from the influence zone to decrease the global number of nodes and computational cost. Influence zone is the location over the model where the mechanical issues, e.g. stress concentration, are interested and finding out its stress magnitude is very important [1, 2].

Mesh characteristics include the size and aspect ratio of elements, the number of nodes per element and globally for the whole model (structure) and the way its nodes positioned relative to each other (connectivity). The less the number of nodes of the model, the less the computational cost will be. Therefore, experts seek for larger elements whenever they are moving farther from the influence zone [1, 2].

There are two main different approaches for enlarging elements throughout the model: (1) enlarging an element individually (if the configuration of neighboring element allows), and or (2) creating ***transition elements***. The first approach means one makes bigger elements as far as the shape of element and its adjacent elements configuration make it possible and this approach is the elementary one to think. The second approach lets the experts to have different element size with different nodes, which is related to the mesh of neighboring elements and helps them to pass from the fine elements to coarse elements when they move farther from the interested location [1-6].

1.1. Example I

As the first describing example, see Fig. 1 which is schematically showing a two-dimensional **Bossinesq problem** [8] in the elasticity theory. As it is seen, the elements near to the point load are finer and are getting coarser moving far from the point load. The shaded elements are ***transition elements,*** sometimes called "***variable-number-nodes elements***", because the number of the nodes per element in transition elements is variable [7, 9-11].

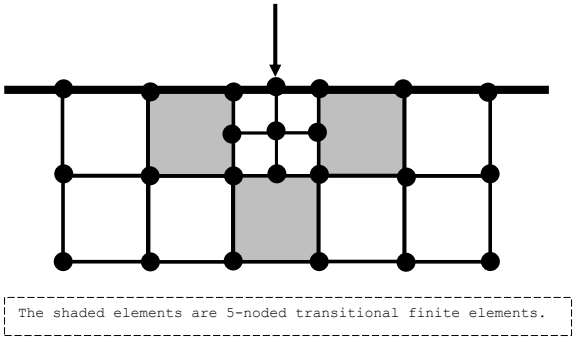

Fig. 1. Fine and coarse elements in finite element solution of Bossinesq's problem in a two-dimensional analysis (Example I).

1.2. Example II

The second example is shown in Fig. 2 which depicts the capability of remeshing of elements near the zone of interest, *crack tip*. It is a two-

* Corresponding author Tel: +98-56-32202049-50. *E-mail address:* aminchamani@birjand.ac.ir (A. Chamani).

dimensional example for showing more clearly and easily the concept of transition elements [12-17].

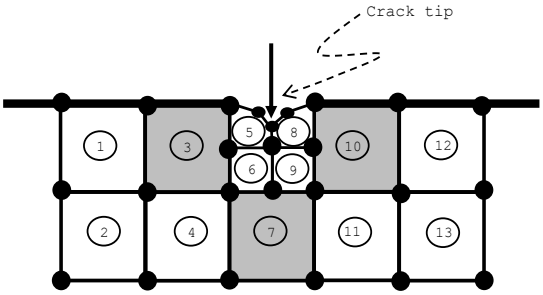

Fig. 2. Refinement of elements with variable nodes near to crack tip.

Fig. 2 shows the connectivity of elements with variable nodes for elements at different location relating to the crack tip. Elements number 1, 2, 4, 11, 12, and 13 as well as elements number 6, and 9 are isoparametric elements with four nodes per each element (corner nodes). Elements number 3, 7, and 10 are transition elements and the first layer of transition elements with 5 nodes per each element. These three elements have a middle-side nodes and each element totally has 5 nodes. Again, elements number 5, and 8 are isoparametric elements with 5 nodes per element.

In this study, after an introduction to the concept of transition or variable-number –nodes elements, the subject is clarified qualitatively using the discretization of Bossinesq problem and the second example of crack tip (fracture mechanics) using the modeling of transition elements. Afterward, the formulation of three-dimensional transition finite elements is developed, and finally, the finite element analysis of the 3D structure introduced in Section 2 is performed and the results show logical and reasonable values.

2. Example of three-dimensional transition finite elements

The examples I and II mentioned in the preceding Section were qualitative two-dimensional problems. Since our developed in-house finite element program is a three-dimensional one, we introduce a new example (Example III) designed for the three-dimensional mesh illustration and finite element analysis (Fig. 3). It is a 5*1*1 m^3 horizontal block, which is under 1MPa pressure on the upper side. The mesh consists of 3D hexagonal elements and it forms 8-noded, 20-noded and 8-20 variable node elements while the last one is referred to as the ***transition element*** [7, 10]. The block was discretized into five 1*1*1 m^3 adjacent cubic elements with different number of nodes per elements (Fig. 4). In this section, we describe the problem qualitatively, and in Appendix III, we will have an analogy between the results of its three-dimensional analysis by our in-house finite element program with anticipated rational values of the global *end force vector* for the sake of validation.

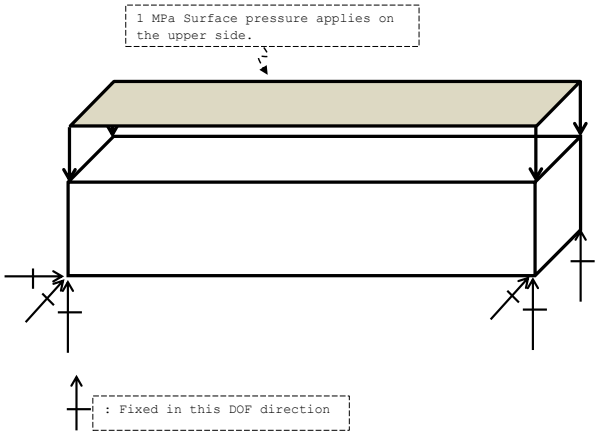

Fig. 3. Schematic diagram of Example II.

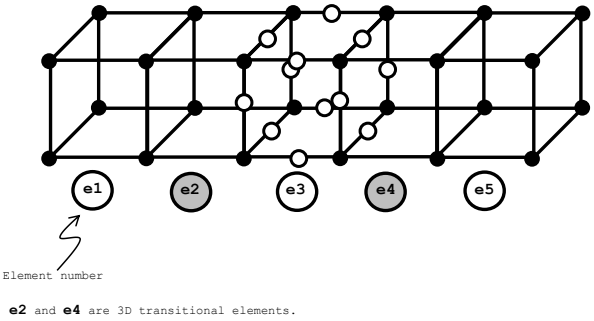

Fig. 4. Nodes and element connectivity of Example II.

To show more clearly the connectivity of the elements of this example we separated elements and their nodes and showed number of nodes (Fig. 5). There are two types of nodes: a corner node and an intermediate node located in the middle of each edge [1, 2]. We showed the corner nodes by black filled circles and intermediate nodes by white filled circles in these figures.

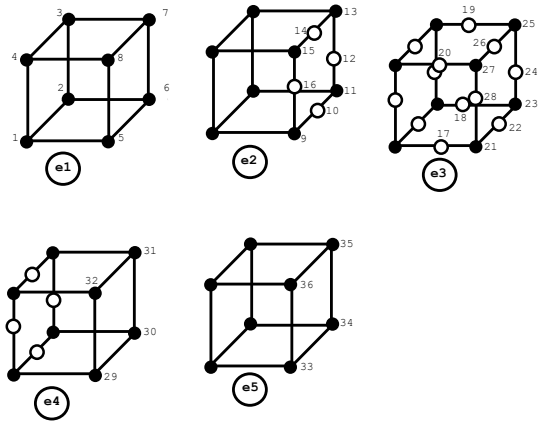

Fig. 5. Elements showed separately with node numbers.

3. Shape function formulation of three-dimensional transition element

Nodes 1-8 in the three-dimensional isoparametric hexagonal element essentially should exist; however, in this study, we focus on the three-dimensional hexagonal element with variable-number-nodes from 8 to 20 nodes per brick (hexagonal) element.

As a simple and basic example, suppose a cubic element where the corner nodes fundamentally exist and does the 9th node, as well (Fig. 6). The 9th node is an intermediate node and is located in the middle of nodes 1 and 2.

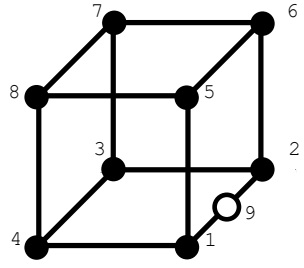

Fig. 6. Transition hexagonal element with one intermediate node (9th node).

Let us consider the intermediate ninth node in Fig. 6. If we use the shape function of the ninth node of 20-noded hexagonal element then it is zero at all nodes whether they exist or not except the ninth node where it is equal to unity. Since we used the corner node shape functions from an 8-noded source element, especially nodes 1 and 2, these shape automatically functions equal to zero at all other nodes; third to eighth.

Albeit shape functions of two corner adjacent nodes (nodes 1 and 2) have non-zero values, (but equal ½) at the location of the ninth node, and should be modified and corrected to be zero (Fig. 7), it is an intrinsic property of any shape function that should be zero in all other nodes except the considered node itself. Therefore, the shape function of the ninth node of a 20-noded source hexagonal element is helpful.

If node 9^{th} exists:

$$FN(1)^{8\text{-}20}=FN(1)^{8}- (1/2)FN(9)^{20}$$

$$FN(2)^{8\text{-}20}=FN(2)^{8}- (1/2)FN(9)^{20}$$

Fig. 7. Shape functions (FN) value modification while the ninth intermediate node exists.

4. Programming strategy

The concept of input file for the structures thath have transition elements needs modification rather than those elements that have constant numbers of nodes per element all over the model. We will consider a three-dimensional 8-20 hexagonal element henceforth. In such three-dimensional hexagonal elements, there are two types of nodes as mentioned before: 1) corner nodes and 2) intermediate nodes.

The first aim is determining the intermediate nodes of each element where their local positions in different elements vary and are not predefined. In our study, we put 20 empty positions and filled them with the global number of the nodes considering the relative local position in the element. If any node did not exist (nodes 9^{th} to 20^{th}), then a zero would be assigned for its position in this 20-positioned list. If there was any, an intermediate node at any location in the global node number would be given to the appropriate position in the list. It is clear that the first eight positions of each element would be filled necessarily because all elements in this in-house finite element (FE) program have at least eight different corner nodes. As an example, note the 9-noded element of Fig. 8. Intermediate 2017^{th} global node fills the 18^{th} position of the element configuration numbering list (Table 1). With the aim of this rearranging local nodes numbering for each element, the new subroutine, **Subroutine MODIFY**, was designed, developed and has been brought completely in **APPENDIX I**.

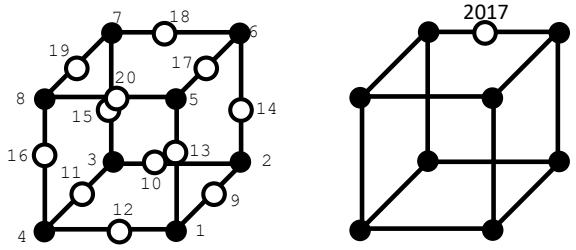

Fig. 8. A 20-noded 3D source element and variable-nodes-element (9-noded).

Table 1. List of local positions of nodes in sample element.

1	2	3	4	5	6	7	8	9	10
*	*	*	*	*	*	*	*	0	0
11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	2017	0	0

5. Conclusion

In this paper, the concept of transition element in a finite element analysis of a continuum was considered and the computational methodology as well as programing strategy were comprehensively discussed, and finally, a verification example was described (Appendix II). It was shown that the transition element enables experts to have a discretized structure with a lower size of global number of nodes with larger elements going farther of the influenced zone. Despite the fact that the concept of transition element is not that new and dates back to the late 1970s [1, 2], but to the best knowledge of authors, an obvious and clear programming strategy and the method of implementation in a FE program have not been depicted in particular in the related literature yet. The algorithm and the related part of the program for analyzing of a structure having variable-number-of-nodes per element were comprehensively brought in this manuscript. Finally, the validating example proved the right and proper capability of the programming strategy and the analogy between the program results and a rational anticipation showed a complete agreement.

REFRENCES

[1] Bathe, K. J. (2006). Finite Element Procedures. Prentice Hall, Pearson Education, Inc. 1037 P.

[2] Hughes, T. J. R (2000). The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover Publication. 682 P.

[3] Ergatoudis, I., Irons, B.M. and Zienkiewicz, O.C. (1968). Curved, Isoparametric, Quadrilateral Elements for Finite Element Analysis. International Journal of Solids and Structures, 4, 31-42.

[4] Irons and Zienkiewicz, (1968). The Isoparametric Finite Element System: A New Concept in Finite Element Analysis. Royal Aeronautical Society. 3-35.

[5] Irons. (1969). Economical computer techniques for numerically integrated finite elements. International Journal for Numerical Methods in Engineering. 1, 201-203.

[6] Irons B. M.. (2005) Quadrature rules for brick based finite elements, International Journal for Numerical Methods in Engineering, 3, 2, 293-294

[7] M. Calvin Mosher, (1985). A variable node finite element method. Journal of Computational Physics, Vol. 57, Issue 2, 157-187.

[8] Sadd, M. (2005). Elasticity: Theory, Applications, and Numerics. Elsevier Inc. 473 P.

[9] Gupta, A., & Mohraz, B. (1972). A method of computing numerically integrated stiffness matrices. International Journal for Numerical Methods in Engineering, Vol. 5, 83-89

[10] Gupta, A. K. (1978). A finite element for transition from a fine to a coarse grid. International Journal for Numerical Methods in Engineering, Vol. 12. 35-45

[11] Somervaille I. J. (1973). A technique for mesh grading applied to conforming plate bending finite elements. International Journal for Numerical Methods in Engineering. Vol. 6, Issue 2, 310-312.

[12] Belytschko T, Black T. 1999. Elastic crack growth in finite elements with minimal remeshing. International Journal for Numerical Methods in Engineering 45(5):601-620.

[13] Shih C, Asaro R 1988. Elastic-plastic analysis of cracks on bimaterial interfaces: part I—small scale yielding. Journal of Applied Mechanics 55:299-316.

[14] Ewalds H, Wanhill R. 1989, Fracture Mechanics. Edward Arnold: New York.

[15] Benzley S, 1974, Representation of singularities with isoparametric finite elements. International Journal for Numerical Methods in Engineering 8:537-545.

[16] Moesy N. Dolboz J. Belytschko T., 1999, A finite element method for crack growth without remeshing, Int. J. Numer. Meth. Engng. 46, 131-150

[17] Geralf Hutter and Lutz Zybell, 2016, Recent Trends in Fracture and Damage Mechanics, pp. 434.

APPENDIX I

In this appendix, we show the part of the code, named **SUBROUTINE MODIFY**, whose task is rearranging the element's nodes global number considering the non-existing or existing nodes.

```
C
C      >>
C      A ROUTINE FOR MODIFICATION OF CONNECTIVITY MATRIX

       SUBROUTINE MODIFY(NEL,NNODET,NOC,NON,MD)
       DIMENSION NOC(NEL,NNODET),NON(NEL),MD(NEL,12)

       DO 01 IE=1,NEL
       K=8
       K1=0
       DO 02 J=9,20
       IF(NOC(IE,J).GT.0)THEN
       K=K+1
       K1=K1+1
       NOC(IE,K)=NOC(IE,J)
       MD(IE,K1)=J
       ENDIF
02     CONTINUE
       NON(IE)=K
01     CONTINUE

       END SUBROUTINE
```

APPENDIX II

Here two subroutines are brought which determine the shape functions of the element with variable number of nodes. This part is the heart of any finite element analysis of continuum.

```
C
C      >>
C       A ROUTINE FOR SHAPE FUNCTIONS AND THEIR DERIVATIVES
C         RESPECT TO LOCAL COORDINATES (EXI) CALCULATION
C         X1-> EXI, X2 -> ETA, X3 -> SAI

       SUBROUTINE
SHAP(X1,X2,X3,NNODE,IE,NEL,NDOF,MD,FN,DFXI)

       REAL                                          (KIND=8)
GFN(20),GDFXI(20,3),FN(NNODE),DFXI(NNODE,NDOF)
       REAL (KIND=8) X1,X2,X3

       DIMENSION MD(NEL,12)
       DIMENSION NGH(20,2)

       FN(1)=(1+X1)*(1-X2)*(1-X3)/8
       DFXI(1,1)=(+1-X2)*(1-X3)/8
       DFXI(1,2)=(-1-X1)*(1-X3)/8
       DFXI(1,3)=(-1-X1)*(1-X2)/8

       FN(5)=(1+X1)*(1-X2)*(1+X3)/8
       DFXI(5,1)=(+1-X2)*(1+X3)/8
       DFXI(5,2)=(-1-X1)*(1+X3)/8
       DFXI(5,3)=(+1+X1)*(1-X2)/8

       FN(2)=(1+X1)*(1+X2)*(1-X3)/8
       DFXI(2,1)=(+1+X2)*(1-X3)/8
       DFXI(2,2)=(+1+X1)*(1-X3)/8
       DFXI(2,3)=(-1-X1)*(1+X2)/8

       FN(6)=(1+X1)*(1+X2)*(1+X3)/8
       DFXI(6,1)=(+1+X2)*(1+X3)/8
       DFXI(6,2)=(+1+X1)*(1+X3)/8
       DFXI(6,3)=(+1+X1)*(1+X2)/8

       FN(3)=(1-X1)*(1+X2)*(1-X3)/8
       DFXI(3,1)=(-1-X2)*(1-X3)/8
       DFXI(3,2)=(+1-X1)*(1-X3)/8
       DFXI(3,3)=(-1+X1)*(1+X2)/8

       FN(7)=(1-X1)*(1+X2)*(1+X3)/8
       DFXI(7,1)=(-1-X2)*(1+X3)/8
       DFXI(7,2)=(+1-X1)*(1+X3)/8
       DFXI(7,3)=(+1-X1)*(1+X2)/8

       FN(4)=(1-X1)*(1-X2)*(1-X3)/8
       DFXI(4,1)=(-1+X2)*(1-X3)/8
       DFXI(4,2)=(-1+X1)*(1-X3)/8
       DFXI(4,3)=(-1+X1)*(1-X2)/8

       FN(8)=(1-X1)*(1-X2)*(1+X3)/8
       DFXI(8,1)=(-1+X2)*(1+X3)/8
       DFXI(8,2)=(-1+X1)*(1+X3)/8
       DFXI(8,3)=(+1-X1)*(1-X2)/8

       GFN(10)=(1-(X1**2))*(1+X2)*(1-X3)/4
       GDFXI(10,1)=(-2*X1)*(1+X2)*(1-X3)/4
       GDFXI(10,2)=(+1)*(1-(X1**2))*(1-X3)/4
       GDFXI(10,3)=(-1)*(1-(X1**2))*(1+X2)/4

       GFN(12)=(1-(X1**2))*(1-X2)*(1-X3)/4
       GDFXI(12,1)=(-2*X1)*(1-X2)*(1-X3)/4
       GDFXI(12,2)=(-1)*(1-(X1**2))*(1-X3)/4
       GDFXI(12,3)=(-1)*(1-(X1**2))*(1-X2)/4

       GFN(18)=(1-(X1**2))*(1+X2)*(1+X3)/4
       GDFXI(18,1)=(-2*X1)*(1+X2)*(1+X3)/4
       GDFXI(18,2)=(+1)*(1-(X1**2))*(1+X3)/4
       GDFXI(18,3)=(+1)*(1-(X1**2))*(1+X2)/4

       GFN(20)=(1-(X1**2))*(1-X2)*(1+X3)/4
       GDFXI(20,1)=(-2*X1)*(1-X2)*(1+X3)/4
       GDFXI(20,2)=(-1)*(1-(X1**2))*(1+X3)/4
       GDFXI(20,3)=(+1)*(1-(X1**2))*(1-X2)/4

       GFN(9) =(1-(X2**2))*(1+X1)*(1-X3)/4
       GDFXI(9,1)=(+1)*(1-(X2**2))*(1-X3)/4
       GDFXI(9,2)=(-2*X2)*(1+X1)*(1-X3)/4
       GDFXI(9,3)=(-1)*(1-(X2**2))*(1+X1)/4

       GFN(17)=(1-(X2**2))*(1+X1)*(1+X3)/4
       GDFXI(17,1)=(+1)*(1-(X2**2))*(1+X3)/4
       GDFXI(17,2)=(-2*X2)*(1+X1)*(1+X3)/4
       GDFXI(17,3)=(+1)*(1-(X2**2))*(1+X1)/4

       GFN(19)=(1-(X2**2))*(1-X1)*(1+X3)/4
       GDFXI(19,1)=(-1)*(1-(X2**2))*(1+X3)/4
       GDFXI(19,2)=(-2*X2)*(1-X1)*(1+X3)/4
       GDFXI(19,3)=(+1)*(1-(X2**2))*(1-X1)/4

       GFN(11)=(1-(X2**2))*(1-X1)*(1-X3)/4
       GDFXI(11,1)=(-1)*(1-(X2**2))*(1-X3)/4
       GDFXI(11,2)=(-2*X2)*(1-X1)*(1-X3)/4
       GDFXI(11,3)=(-1)*(1-(X2**2))*(1-X1)/4

       GFN(13)=(1-(X3**2))*(1+X1)*(1-X2)/4
       GDFXI(13,1)=(+1)*(1-(X3**2))*(1-X2)/4
       GDFXI(13,2)=(-1)*(1-(X3**2))*(1+X1)/4
       GDFXI(13,3)=(-2*X3)*(1+X1)*(1-X2)/4

       GFN(14)=(1-(X3**2))*(1+X1)*(1+X2)/4
       GDFXI(14,1)=(+1)*(1-(X3**2))*(1+X2)/4
       GDFXI(14,2)=(+1)*(1-(X3**2))*(1+X1)/4
       GDFXI(14,3)=(-2*X3)*(1+X1)*(1+X2)/4

       GFN(15)=(1-(X3**2))*(1-X1)*(1+X2)/4
       GDFXI(15,1)=(-1)*(1-(X3**2))*(1+X2)/4
       GDFXI(15,2)=(+1)*(1-(X3**2))*(1-X1)/4
       GDFXI(15,3)=(-2*X3)*(1-X1)*(1+X2)/4

       GFN(16)=(1-(X3**2))*(1-X1)*(1-X2)/4
       GDFXI(16,1)=(-1)*(1-(X3**2))*(1-X2)/4
       GDFXI(16,2)=(-1)*(1-(X3**2))*(1-X1)/4
       GDFXI(16,3)=(-2*X3)*(1-X1)*(1-X2)/4

       CALL NEIGHBOR(NGH)
       IF(NNODE.GT.8)THEN
       J1=0
       DO 01 IN=9,NNODE
       J1=J1+1
       IOR=MD(IE,J1)
       FN(IN)=GFN(IOR)
       DO 04 J5=1,3
04     DFXI(IN,J5)=GDFXI(IOR,J5)
       DO 02 J2=1,2
       J3=NGH(IOR,J2)
       FN(J3)=FN(J3)-(GFN(IOR)/2)
       DO 03 J4=1,3
03     DFXI(J3,J4)=DFXI(J3,J4)-(GDFXI(IOR,J4)/2)
02     CONTINUE
```

```
01    CONTINUE
      ENDIF

      END SUBROUTINE


C
C     >>
C       A ROUTINE FOR NEIGHBOR NODE DETERMINATION

      SUBROUTINE NEIGHBOR(NGH)
      DIMENSION NGH(20,2)

      NGH(9,1)=1
      NGH(9,2)=2
      NGH(10,1)=2
      NGH(10,2)=3
      NGH(11,1)=3
      NGH(11,2)=4
      NGH(12,1)=1
      NGH(12,2)=4
      NGH(13,1)=1
      NGH(13,2)=5
      NGH(14,1)=2
      NGH(14,2)=6
      NGH(15,1)=3
      NGH(15,2)=7
      NGH(16,1)=4
      NGH(16,2)=8
      NGH(17,1)=5
      NGH(17,2)=6
      NGH(18,1)=6
      NGH(18,2)=7
      NGH(19,1)=7
      NGH(19,2)=8
      NGH(20,1)=5
      NGH(20,2)=8

      END SUBROUTINE
```

APPENDIX III

The values of end force vectors for nodes of the discretized block (Example III); the values of upward supports totally equal to the multiplication of surface pressure (1.0 MPa) by the upper area (5*1 m²). The results (bolded and underlined number in the Table 2.) are in complete agreement with the analytical anticipation.

Table 2. The end force vectors of the structure's nodes.

NODE #	END FORCE VECTORS FOR ALL NODES		
	X	Y	Z
1	-0.2443E+02	0.3537E+05	**0.1250E+07**
2	0.2092E+00	-0.3537E+05	**0.1250E+07**
3	-0.2455E+00	0.6678E+00	-0.2500E+06
4	0.8079E+00	0.7137E+00	-0.2500E+06
5	0.1438E+01	0.8828E+00	0.2906E+01
6	0.2938E+01	-0.2508E+01	-0.8750E+00
7	0.3375E+01	-0.1938E+01	-0.5000E+06
8	0.3938E+01	0.2133E+01	-0.5000E+06
9	0.2688E+01	0.3508E+01	0.2653E+02
10	-0.4500E+01	0.3311E+01	-0.2298E+02
11	-0.8438E+00	-0.4031E+01	-0.1075E+02
12	0.7461E+01	-0.4295E+01	-0.3066E+02
13	-0.1906E+01	-0.3281E+01	0.1291E+02
14	0.9625E+01	0.2938E+00	-0.6667E+06
15	-0.1191E+02	0.5344E+01	0.2362E+02
16	-0.8100E+01	-0.7795E+01	-0.3734E+02
17	-0.6354E+01	-0.1185E+00	0.5724E+01
18	-0.1279E+01	0.5648E+01	0.1447E+02
19	0.6766E+01	-0.3209E+01	-0.3333E+06
20	0.4661E+01	-0.5030E+01	-0.3333E+06
21	-0.2438E+01	-0.1026E+02	-0.1238E+02
22	-0.8375E+01	0.2397E+01	0.1209E+02
23	-0.2438E+01	-0.5469E+00	0.3164E+02
24	-0.1869E+01	-0.1122E+02	-0.1106E+02
25	-0.3656E+01	0.2320E+01	0.1602E+02
26	0.3000E+01	0.6993E+01	-0.6667E+06
27	-0.6250E-01	-0.7348E+01	0.1418E+02
28	0.3770E+00	0.1633E+02	0.3862E+02
29	-0.4375E+00	-0.1031E+01	-0.6375E+01
30	-0.2812E+01	0.2906E+01	-0.1875E+00
31	-0.1688E+01	0.4008E+01	-0.5000E+06
32	0.2312E+01	0.1086E+01	-0.5000E+06
33	0.3216E-01	0.7278E+01	**0.1250E+07**
34	-0.1991E+01	-0.4653E+00	**0.1250E+07**
35	-0.6340E+00	0.1538E+00	-0.2500E+06
36	0.3742E+00	-0.6833E+00	-0.2500E+06