ASIAN JOURNAL OF CIVIL ENGINEERING (BUILDING AND HOUSING) VOL. 7, NO. 5 (2006)
PAGES 525-549

INTEGRATION SOLUTION ROUTINE TO EVALUATE THE ELEMENT STIFFNESS MATRIX FOR DISTORTED SHAPES

D.J. Hamadi[*a] and M.T. Belarbi[b]
[a]LESIA Laboratory, Civil Engineering Department, Biskra University, Algeria
[b]Civil Engineering Department, Biskra University, Algeria

ABSTRACT

The present paper attempts to develop a new analytical integration to evaluate the element stiffness matrix for the finite elements with irregular shapes. Most of the finite elements developed by Sabir are based on the strain rather than displacement approach. They are characterized by a regular form and appropriate coordinates with the form of the element. Together, they tend to decrease the elements utilization domain. Hence, for reasons of importance and particularity of these elements (higher order shape functions expressed in terms of independent strains); it is necessary to introduce irregular forms, which require a special integration technique, and a specific classification in programming level for different geometric forms. To overcome this geometrical inconvenience; the paper presents a new integration solution routine. This will help to know how the elements will behave when they have irregular form, and to extend their applications domain for the curved structures no matter what the geometrical shape of the element might be.

Keywords: Strain approach, analytical integration, rectangular element, new integration routine regular and irregular forms

1. INTRODUCTION

Most of the finite elements based on assumed strains have been developed since 1971 by many researchers like Sabir, Ashwell, Djoudi, Salhi, Belarbi and others. Many of them were undertaking their research work at Cardiff University in the U.K. These elements were characterized by a regular form and appropriate coordinates with the form of the element; these coordinates can be either Cartesian, polar, spherical, cylindrical or else conical.

A new finite element for cylindrical shells was developed by Ashwell and Sabir [1]. The effectiveness of this rectangular element was tested by applying it to the analysis of the familiar pinched cylinder and barrel vault problems. The results obtained were shown to converge rapidly for the displacements as well as stresses. Further investigations on this

[*] Email-address of the corresponding author: Dhamadiuk@yahoo.com

cylindrical shell element were carried out by analysing thin shells of the order of (r/t = 320). The results obtained were superior to those obtained from Cantin and Clough's [2] (24×24) element, and also to those drawn from the simplified (20x20) form suggested by Sabir and Lock [3].

The strain based approach was further applied by Sabir [4] to develop a new class of elements for general plane of elasticity problems in Cartesian coordinates. In the last reference a basic rectangular element having the only essential nodal degrees of freedom (2 d.o.f/node) and satisfying the requirements of the strain free rigid body modes is developed. The compatibility within the element is first established. Other elements meeting the above basic considerations together with equilibrium within the element are also developed. A simple an efficient rectangular element including the in-plane rotation is derived. This element was first applied to the simple problem of cantilevers and simply supported beams, where the results for deflections as well as stresses were satisfactory and converged to the exact solution. With the continuation of the development of the strain based approach many elements for general plane elasticity as well as shells have been derived by Sabir et al [5,6,7,8].

To model a structure which has complex geometrical shape in real problem, by a limited number of elements as cited above; is not sufficient at all. To overcome this geometrical inconvenience; the paper presents a new integration solution routine. This solution is adopted for two reasons. First, to know how these elements will behave when they have irregular forms. And second, in the positive case, to extend their applications domain for the curved structures whatever the geometrical shape of the structure could be. The performance of this new solution routine is tested by applying to the analysis of the problems used in previous publications and to obtain solutions for practical problems in engineering.

2. INTEGRATION METHOD

2.1 Numerical integration

The element stiffness matrix can be calculated using the following Eq. (1)

$$[K_e] = \left[A^{-1}\right]^T \iiint [Q]^T [D][Q] dv [A]^{-1} \tag{1}$$

To carry out the integral, we have to choose either numerical integration (e.g Gauss integration) or analytical integration. One of the disadvantages of the numerical integration is the high order of the monomials after the three multiplications of integral matrices Eq. (1), which would signify many integration points.

2.2 Sabir approach[6]

If we consider the triangular element shown in Figure 1.

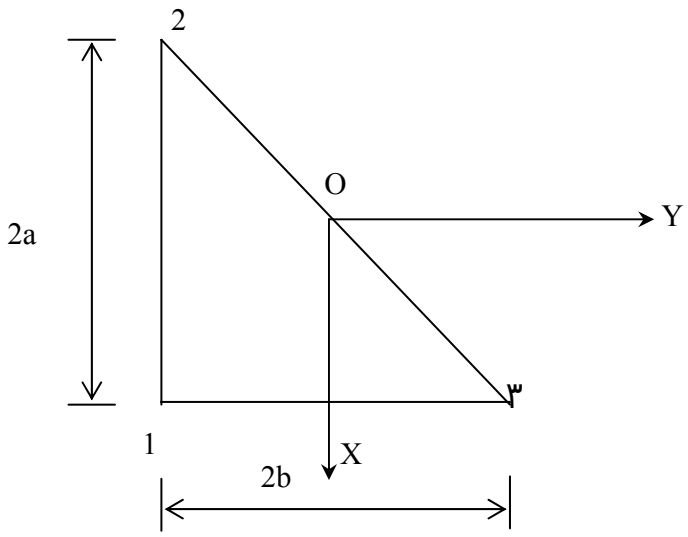
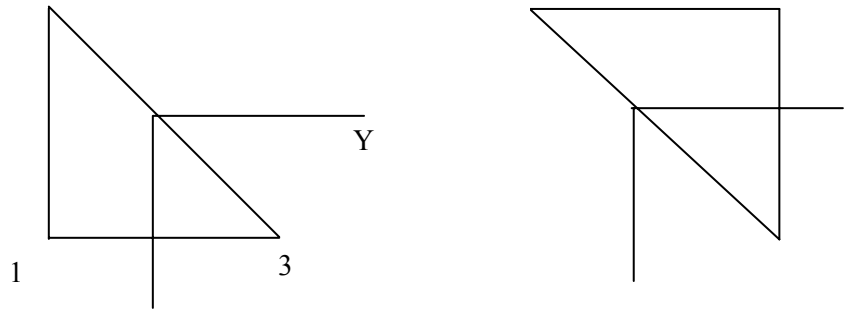


Figure 1. Triangular element, Coordinate axes

$$[K_e] = [A^{-1}]^T \left[\int_{-b}^{b} \int_{\frac{ay}{b}}^{a} [Q]^T [D][Q] dxdy \right] [A]^{-1} \tag{2}$$

The multiplication and integration of the terms within the brackets Eq. (2) are carried out explicitly. In order to use the nodal Sabir solution routine and to simplify the assembly of the finite elements, for the problem considered, Sabir used the following technique in which two triangles are combined together to form a rectangular element as shown in Fig 3. This was achieved by substituting the coefficients of each node from the element stiffness matrices of the two triangles into their corresponding place in the element stiffness matrix of the two combined elements as shown in Figures. 2 and 3. The stiffness matrix of the combined elements will then be used in the assembly of the overall stiffness matrix of the structure. Unfortunately, the above technique is suitable only for a rectangle triangular element (rectangular form) which decreases its utilization domain:

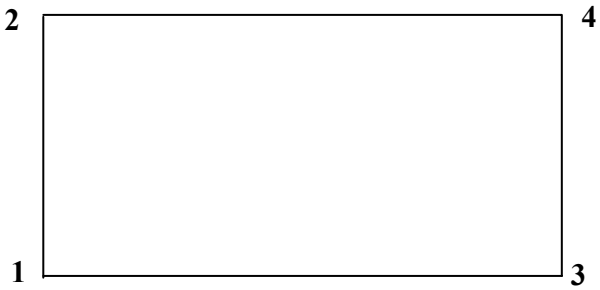
Firstly according to the integral limits, the obtained element has a simple shape which is a rectangle triangle. Secondly, according to quadrilateral shapes, the element obtained is a simple rectangle. Hence the applied domain will be limited.



A	B	C
D	E	F
G	H	I

a	b	**Case 1.c**
d	e	F
g	h	I

Figure 2. Stiffness matrix of each triangle element



	B	C	
A	E+a	F+b	C
G	H+d	I+e	F
	G	H	I

Figure 3. Stiffness matrix of the combined elements

2.3 A new approach

The evaluation of the element stiffness matrix is summarized with the evaluation of the following expression:

$$[K_e] = [A^{-1}]^T \left[\iint_S [Q]^T [D][Q].dx.dy \right] [A^{-1}] \tag{3a}$$

$$[K_e] = [A^{-1}]^T [K_0] [A^{-1}] \tag{3b}$$

With:

$$[K_0] = \iint_S [Q]^T [D][Q]..dx.dy \tag{3c}$$

Since [A] and its inverse can be evaluated numerically, the evaluation of the integral (3c) becomes the key of the problem.

In general, the multiplication $\mathbf{Q^T\ D\ Q}$ can be done manually, we will end up by calculating the double integrals of the form:

$$I = [K_0] = \iint_S C.x^{\alpha} y^{\beta}\, dx.dy \tag{4}$$

Knowing that, for certain elements, a too great distortion can lead to erroneous numerical results particularly in the calculation of the Jacobien, an expression that is general, and easy to implement numerically being formulated. It allows the evaluation of the matrix $[K_0]$ in an automatic way whatever the degree of the polynomial of the kinematic field and the distortion of the element (figure 4).

The calculation of integral **I** is the principal problem of the calculation of the element stiffness matrix [Ke].

In a very simple and effective manner, the integral is solved by the subroutine "INTEGRATION". To illustrate the step of calculation of the integral in detail, let us take the case of an arbitrary element as shown in figure 4. The integral is composed of three parts symbolized on the figure by Roman numerals $\mathbf{I_1}$ $\mathbf{II_2}$ and $\mathbf{III_3}$, each integral must be calculated separately.

The integral will be solved easily if one can determine the limits of the integral with precaution, which is far from being obvious. The fact that the limits can change with the geometry of the element raises difficulties, which make the programming enormously complex.

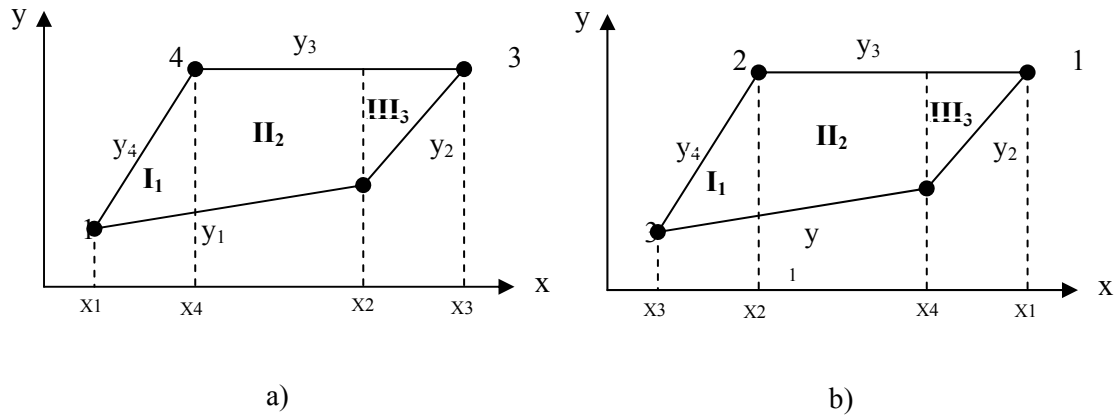


Figure 4. Quadrilateral element

$$I = I_1 + II_2 + III_3 \tag{5}$$

Where:

$$I_1 = \int_{x_1}^{x_4}\int_{y_1}^{y_4} x^{\alpha} y^{\beta} dxdy \tag{6a}$$

$$I_2 = \int_{x_4}^{x_2}\int_{y_1}^{y_3} x^{\alpha} y^{\beta} dxdy \tag{6b}$$

$$I_3 = \int_{x_2}^{x_3}\int_{y_2}^{y_3} x^{\alpha} y^{\beta} dxdy \tag{6c}$$

This means calculating the double integrals of the following form:

$$I = \iint_S C.x^{\alpha} y^{\beta}\, dx.dy \tag{7a}$$

Where
C: constant

y: the ordinate of the segment of equationy = ax + b (7b)

$$y^2 = (ax + b)^2 = \mathbf{1}a^2x^2 + \mathbf{2}abx + \mathbf{1}b^2 \tag{7c}$$

$$y^3 = (ax + b)(ax + b)^2 = \mathbf{1}\, a^3x^3 + \mathbf{3}a^2\, bx^2 + \mathbf{3}ab^2x + \mathbf{1}b^3 \tag{7d}$$

We will end up with the general form of y^β :

$$y^\beta = \sum_{k=1}^{\beta+1} C(k).a^{\beta+1-k}.b^{k-1}.x^{\beta+1-k} = \sum_{k=1}^{\beta+1} C(k).a^{k-1}.b^{\beta+1-k}.x^{k-1} \tag{8}$$

Where

C(k): Coefficients function of β (see table 1), is for example:
if β=1 we will have 2 coefficients (see (7b)).
if β=2 we will have 3 coefficients (see (7c)).
if β=3 we will have 4 coefficients (see (7d)).

Table 1. C(k) coefficients relating to the relation (8)

β	$C(k)_{K=1,6}$					
	C(1)	**C(2)**	**C(3)**	**C(4)**	**C(5)**	**C(6)**
0	1	-	-	-	-	-
1	1	1	-	-	-	-
2	1	2	1	-	-	-
3	1	3	3	1	-	-
4	1	4	6	4	1	-
5	1	5	10	10	5	1

In which

$$\int y^\beta dy = \frac{1}{\beta+1} y^{\beta+1} = \frac{1}{\beta+1}(ax+b)^{\beta+1} = \frac{1}{\beta+1}\sum_{k=1}^{\beta+2} C(k).a^{k-1}.b^{\beta+2-k}.x^{k-1} \tag{9}$$

Therefore

$$\int_{y_i}^{y_j} y^{\beta} dy = \frac{1}{\beta+1}\sum_{k=1}^{\beta+2} C(k).\left(a_j^{k-1}.b_j^{\beta+2-k} - a_i^{k-1}.b_i^{\beta+2-k}\right).x^{k-1} \tag{10}$$

$$\iint x^{\alpha} y^{\beta} dx.dy = \int_m^n \frac{1}{\beta+1}\sum_{k=1}^{\beta+2} C(k).\left(a_j^{k-1}.b_j^{\beta+2-k} - a_i^{k-1}.b_i^{\beta+2-k}\right).x^{k+\alpha-1}.dx \tag{11}$$

$$\iint x^{\alpha} y^{\beta} dx.dy = \frac{1}{\beta+1}\sum_{k=1}^{\beta+2} \frac{1}{k+\alpha} C(k).\left(a_j^{k-1}.b_j^{\beta+2-k} - a_i^{k-1}.b_i^{\beta+2-k}\right)\left(x_n^{k+\alpha} - x_m^{k+\alpha}\right) \tag{12}$$

In our case:

$$I = \sum_{P=1}^{3} I_P \tag{13}$$

The general expression of I_P for a quadrilateral would be:

$$I_P = \frac{C}{\beta+1}\sum_{k=1}^{\beta+2} \frac{1}{k+\alpha}.C(k)\left(a_j^{k-1}.b_j^{\beta+2-k} - a_i^{k-1}.b_i^{\beta+2-k}\right)\left(x_n^{k+\alpha} - x_m^{k+\alpha}\right) \tag{14}$$

That is to say the expression of I for a triangle is:

$$I = \sum_{P=1}^{2} I_P \tag{15}$$

3. PROGRAMMING THE EXPRESSION

3.1 Determination of the integral limits

The limits of the volumetric integral of the equation (4) depend on the element geometry. In the following figures (figures 5 to 12) all the possible cases that must be distinguished when calculating the integral are schematized. The different figures are characterized by their integration limits**.** Let us take for example figures 5 and 6. To calculate the integral of the first part, I1 should be solved by the following equation:

$$I_1 = \int_{x(A)}^{x(B)} \int_{y_1}^{y_4} x^{\alpha} y^{\beta} dxdy \quad \text{in the case of figure 5} \tag{16}$$

$$I_1 = \int_{x(A)}^{x(D)} \int_{y_1}^{y_4} x^{\alpha} y^{\beta} dxdy \quad \text{in the case of figure 6} \tag{17}$$

There is obviously a change of the limits of co-ordinates x. Figures 5 to 12 show all the possible cases: to form a distorted element, there are theoretically 6 possibilities (figures 5 to 10). As the distortion of the elements of Figures 11 and 12 is exaggerated, we can ignore the study of these two cases. We will accept only the use of the elements whose distortion remains moderate.

There remain only the 5 cases of a distorted element (Figures 5 to 9) and the particular case of a rectangular element, illustrated in Figure 10.

Let us examine initially the case of the distorted elements (Figures 5 to 9). Illustrated in the figures in Roman numerals, the integration is composed of three different parts. To calculate the integral of these elements, we need a routine which is able to make the distinction between the 4 possible cases, and which provide the limits of integration. The programming of such a routine is not obvious. The numbering of the nodes varies from 1 to 4 but *a priori* we do not know which node has which numbering. To illustrate the problems, let us look at figure 5. To calculate the integral $\mathbf{I_1}$ we should solve the following integral:

$$I_1 = \int_{x(A)}^{x(B)} \int_{y_1}^{y_4} x^{\alpha} y^{\beta} dxdy \tag{18}$$

Neither the lines y_1 and y_4 nor the limits x (A) and x (B) are easy to determine. The numbering of nodes A and D is unknown. We do not know which nodes are hidden behind the nodes A and B. We thus need a routine which determines the numbering and assigns it with the nodes A, B, C and D.

To simplify the problem, we introduce a convention to number the nodes in anticlockwise direction.

Although this convention was adopted by several authors; it does not solve the whole problem. We cannot still identify the various nodes.

To finally solve the problem, a subroutine **FORM_ICORD** is introduced into the programming. The purpose of this subroutine is to find the sequence of the nodes and to provide the order of the nodes of the element arranged by co-ordinates **x** (according to the ascending order).

Let us look at Figure (4) which shows an element having an arbitrary numbering. The subroutine **FORM_ICORD** introduces a Icord vector of dimension 4. In the example of the Figure (4), Icord stores the following values:

Icord	**Icord(1)**	**Icord(2)**	**Icord(3)**	**Icord(4)**
Number of the node	1	4	2	3

Icord(1) contains the node number with the lowest co-ordinate **x** .

Icord(4) contains the node number writh the highest co-ordinate **x**.

Using the Icord vector we can determine the limits of the integral easily. For example the integral $\mathbf{I_1}$ of the example of figure 5 is calculated in the following way:

$$I_1 = C \int_{x(\mathrm{Icord}(1))}^{x(\mathrm{Icord}(2))} \int_{y_1}^{y_4} x^{\alpha} y^{\beta} dxdy \tag{19a}$$

The key point of this step is to introduce into the limits of the co-ordinates of **x** the Icord vector. For the calculation of the above integral **I** , it is necessary to integrate the node with the lowest co-ordinates of **x** until the node which follows: x(Icord(1))→x(Icord(2)). The second integral $\mathbf{II_2}$ is calculated with the same method.

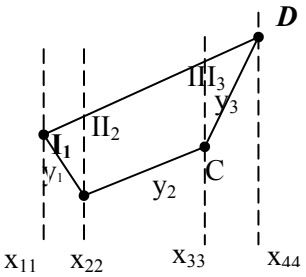


Figure. 5 Shape 1

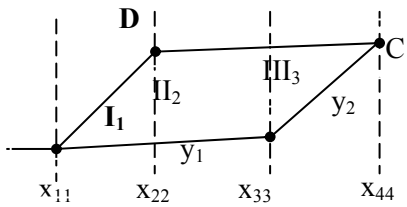


Figure 6. Shape 2

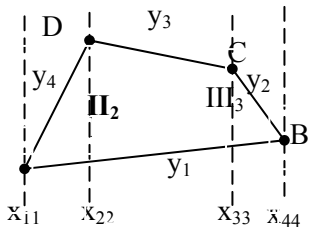


Figure 7. Shape 3

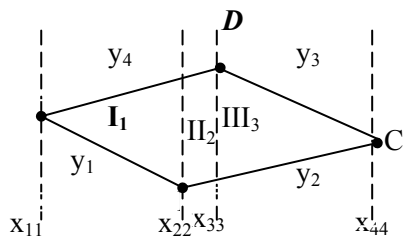
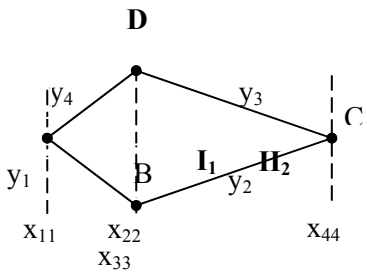


Figure 8. Shape 4



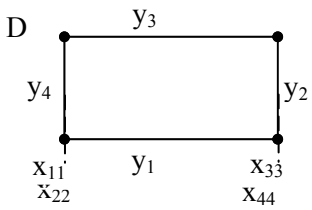
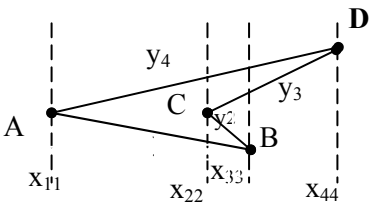


Figure 9. Shape 5

Figure 10. Shape 6



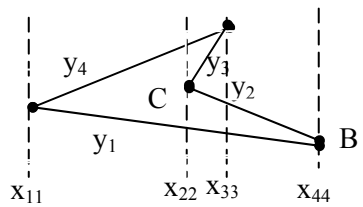
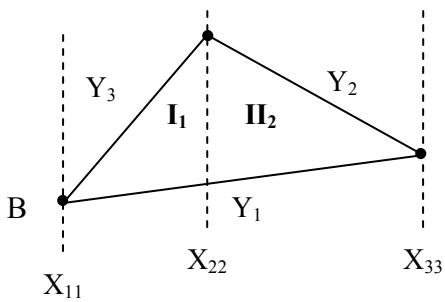


Figure 11. Shape 7

Figure 12. Shape 8

For the case of triangular shapes, we have the following Figures (13, 14).



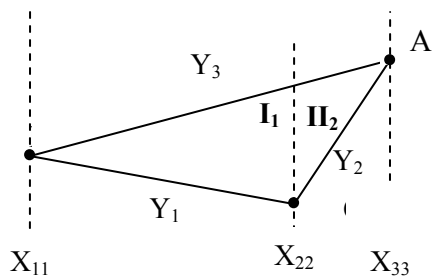


Figure 13. Shape 1

Figure 14. Shape 2

$$II_2 = C \int_{x(Icord(2))}^{x(Icord(3))} \int_{y_2}^{y_4} x^{\alpha} y^{\beta} dxdy \tag{19b}$$

The limits of the co-ordinates of x are replaced by x(Icord(2)) → x(Icord(3)). Likewise, it is necessary for the third integral **III₃** to replace the limits by x(Icord(3)) → x(Icord(4)).

$$III_3 = C \int_{x(Icord(3))}^{x(Icord(4))} \int_{y_3}^{y_4} x^{\alpha} y^{\beta} dxdy \tag{19c}$$

Now we know the limits of co-ordinates **x**, but we cannot still calculate the lines y_1 to y_4.

3.2 Determination of the lines y_1 to y_4 (y_3)

a) Case of quadrilateral shapes

Numbering the nodes in anticlockwise direction simplifies the determination of lines y_1 to y_4. Let us take the element of figure 6. In the drawing we can see the true numbering of the nodes and the numbering with the Icord vector. We can observe that the lines y_1 to y_4 do not change with the geometry of the element. The starting point of line y_1 is always the node stored in Icord(1). In the example of figure 4.a the value stored in Icord(1) is 1. We can easily calculate the second point of the line using the equation:

$$2^{2nd} \text{ node} = \text{Icord}(1) + 1 = 2$$

In the case of figure 4.b the value stored in Icord 1) is 3. The second point of the line can be calculated using the equation:

$$2^{2nd} \text{ node} = \text{Icord}(1) + 3 = 4$$

The other lines y_2 and y_3 are determined in the same way. Any handling of the Icord vector must hold account of which the node numbering is between 1 and 4. If for instance, the node 4 is hidden behind Icord(1), the complement Icord(1) +1 will be 5, which is obviously false. A correction is programmed easily with the order IF of FORTRAN77.

In the case of a rectangular element a subroutine must take account that the slope of the lines y_1 and y_2 is infinite. A value which does not exist in the programming languages.

The subroutine treating the calculation of the lines is called COEFF. The handling of the Icord vector is carried out by the subroutines ICORD_A1, ICORD_A2 and ICORD_A3 which take account of the corrections described above.

b) Case of triangular shapes

The triangular element is a similar to the quadrilateral in point view of numbering of nodes in Icord vector (Figure 13 and 14), within a minimum of geometric forms can be used.

4. CALCULATION OF THE INTEGRAL FOR THE DISTORTED ELEMENTS

With the explanations of the preceding paragraphs, it is now possible to determine the limits of the integral: the lines $y_1$to y_4 and the limits of the co-ordinates of **x.** We can then begin the programming of a routine which carries out integration ***(see appendix).***

5. SUB- PROGRAM OF THE STIFFNESS MATRIX [K_0] FOR THE CASE OF TRIANGULAR SHAPES

After the programming of the routines which calculate the integral, we can finally carry out the calculation of the element stiffness matrix:

```
      SUBROUTINE FORMK_T(TI, E, V, th,EKO)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION EKO(9,9),TI(15)

        DO 51 I=1, 9
        DO 51 J=1, 9
        EKO(I,J)=0.D0
51        CONTINUE
        D11=E/(1.D0-V**2)
        D12=E*V/(1.D0-V**2)
        D22=D11
        D33=E/(2*(1.D0+V))

       EKO(4,4)=D11*TI(1)
       EKO(4,5)=D11*TI(6)
       EKO(4,6)=D12*TI(1)
       EKO(4,7)=D12*TI(2)
       EKO(5,5)=D11*TI(10)
       EKO(5,6)=D12*TI(6)
       EKO(5,7)=D12*TI(7)
       EKO(6,6)=D22*TI(1)
       EKO(6,7)=D22*TI(2)
       EKO(7,7)=D22*TI(3)
       EKO(8,8)=D33*TI(1)
       EKO(8,9)=D33*(TI(2)+TI(6))
       EKO(9,9)=D33*(TI(10)+TI(3)+2.D0*TI(7))
        DO 15 I=1,9
        DO 15 J=1,9
        EKO(J,I)=EKO(I,J)
15        CONTINUE
        RETURN
        END
```

6. NUMERICAL APPLICATIONS

In order to illustrate the interest of the integration subroutine, **"INTEGRATION"** is thus developed. We have chosen to test the Sabir membrane element **SBRIEIR** [6] through three case tests of isotropic plane elasticity, taking into account the geometrical distortions. These tests are regarded as a tool to validate of the membrane elements.

The field of displacement for the element **"SBRIEIR"** is as follows[6]:

$$
\begin{aligned}
u &= a_1 - a_3 y + a_4 x + a_8 y/2 + a_5 xy + a_{10} y^2/2 + a_{11} x y^2 + a_{12} x^2 y^3 \\
v &= a_2 + a_3 x + a_6 y + a_8 x/2 + a_7 xy + a_9 x^2/2 - a_{11} x^2 y - a_{12} x^3 y^2 \\
\phi &= a_3 - a_5 x/2 + a_7 y/2 + a_9 x/2 - a_{10} y - 2a_{11} xy - 3 a_{12} x^2 y^2
\end{aligned}
\tag{20}
$$

Note: The distorted version of the element **"SBRIEIR"** will be baptized **"SBQIEIR"**

6.1 High Order Patch Test: Pure bending of a cantilever

The cantilever is modeled by two membrane rectangular elements (regular mesh) or trapezoidal (distorted mesh); various cases of boundary conditions [9] are shown in the Figures 15a, 15b and 15c.

The results obtained with the element **"SBQIEIR"** are compared with those obtained with other known quadrilateral elements (Q4, 07β MAQ, AQ and PS5β) (Figures 16 and 17).

Note Q4 and PS5β are elements without rotation dof.

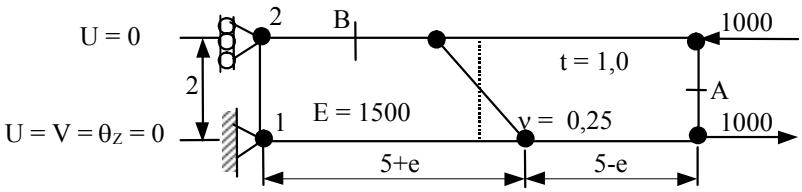


Figure 15a. Pure bending of a cantilever; Data and mesh. Rotation θ_Z is free at 2.

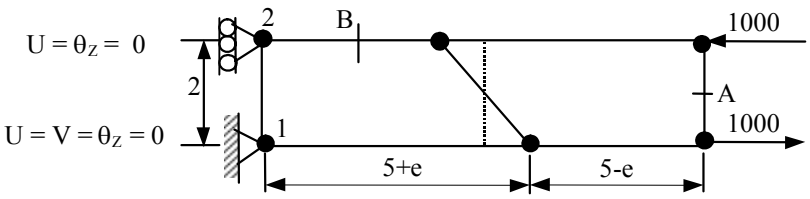


Figure 15b. Pure bending of a cantilever; Data and mesh. Rotation θ_Z is fixed at 1 and 2.

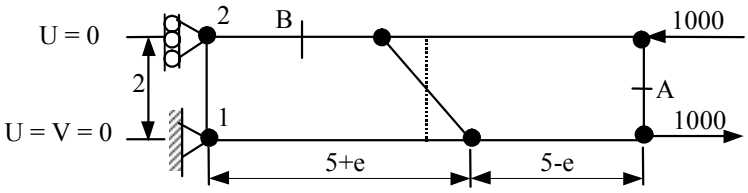


Figure 15c. Pure bending of a cantilever; Data and mesh. Rotation θ_Z is free at 1 and 2.

For the case of the regular mesh (Figure 15a; e=0), a good results are obtained for all the elements except for the standard element Q4 which gives unacceptable results. However, for the case of the distorted mesh characterized by the distance "e" (e> 0), the results of **SBQIEIR** are powerful and comparable with the robust element 07β. Elements AQ, PS5β and MAQ remain sensitive to the distortions of the mesh. For the standard element Q4, the precision is always largely insufficient (Figures 16a and 16b).

In the case of the figure 15b, the robustness of this element *via* the regular and distorted

mesh is confirmed. The figures 17a and 17b show the stability, the reliability and the good performance of **SBQIEIR** no matter what the geometrical distortion might be (only one element on h!). This is explained probably partly by the nature of analytical integration carried out. The distortion has a considerable influence on elements AQ and MAQ, while 07β element is not very sensitive to the geometrical distortions (Figure 17). These results confirm that the modified version of element **SBRIEIR** (**SBQIEIR)** satisfied the High Order Patch Test [10, 11] .

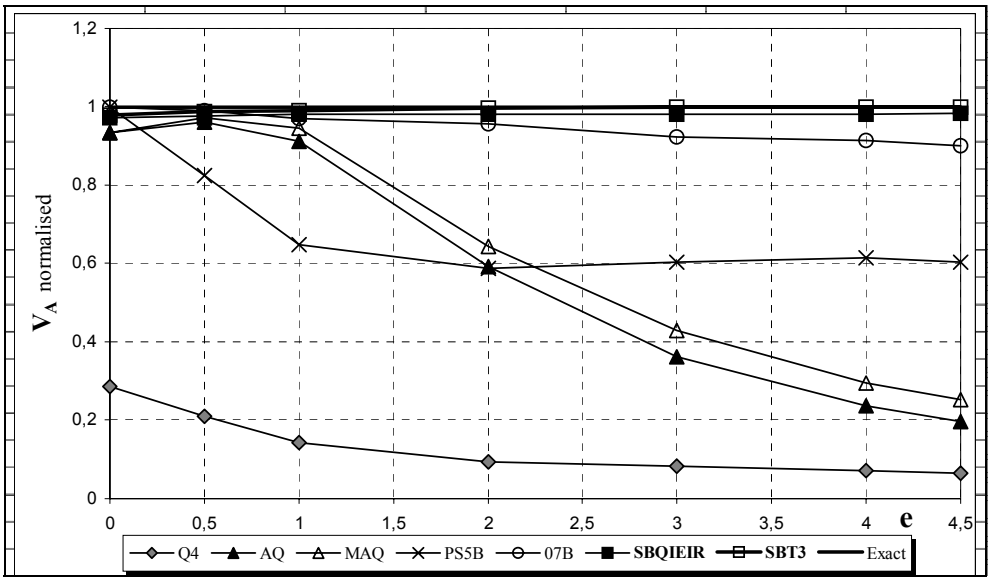


Figure 16a: Pure bending of a cantilever; Rotation θ_Z is free at 2. Vertical displacement at A.
Figure 15a

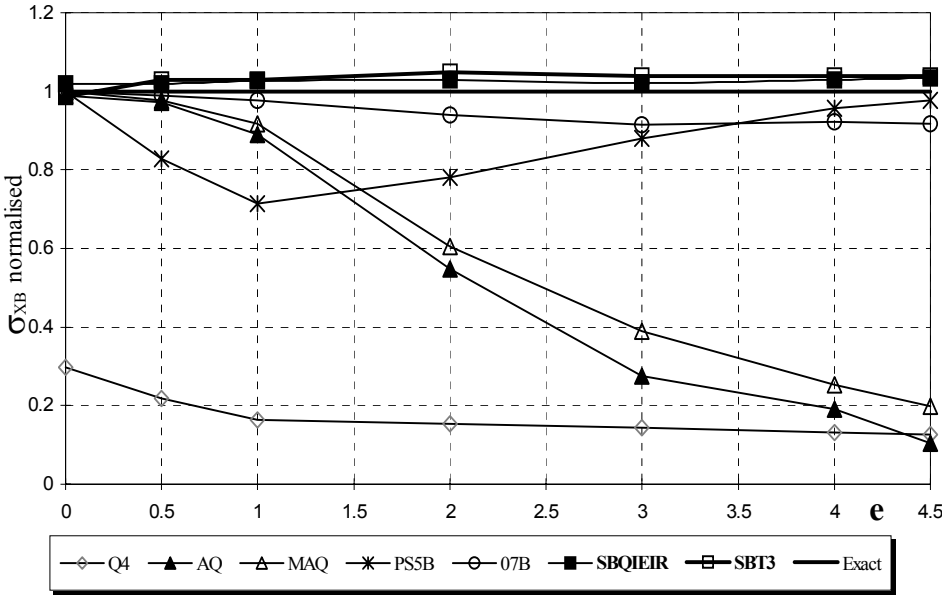


Figure 16b. Pure bending of a cantilever; Rotation θ_Z is free at 2. Normal stress at point B. Figure 15a

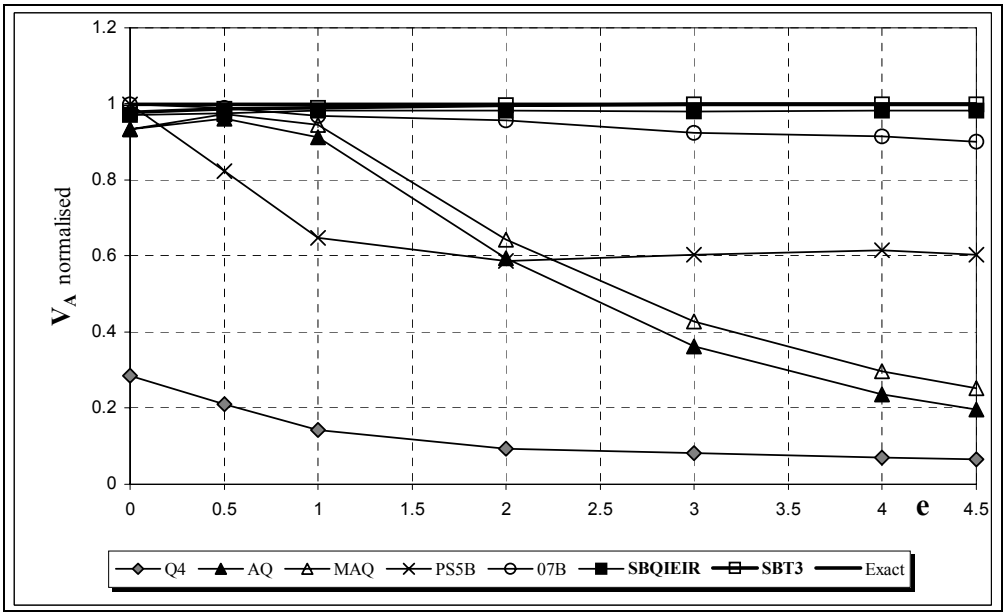


Figure 17a: Pure bending of a cantilever; Rotation θ_Z is free at 1 and 2. Vertical displacement at A. Figure 15b

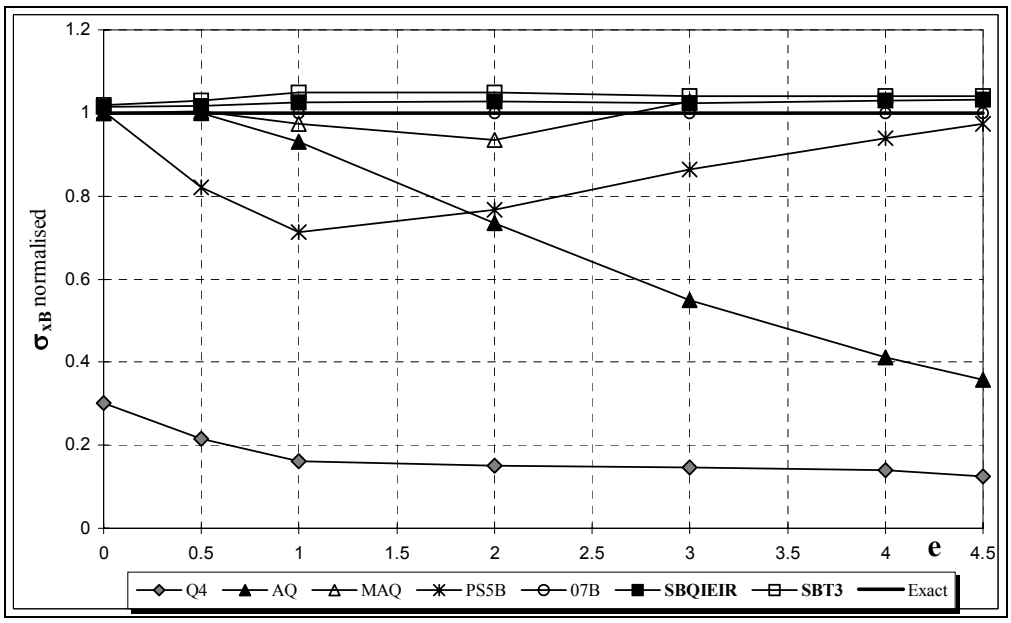


Figure 17b: Pure bending of a cantilever; Rotation θ_Z is fixed at 1 and 2. Normal stress at point B. Figure 15b

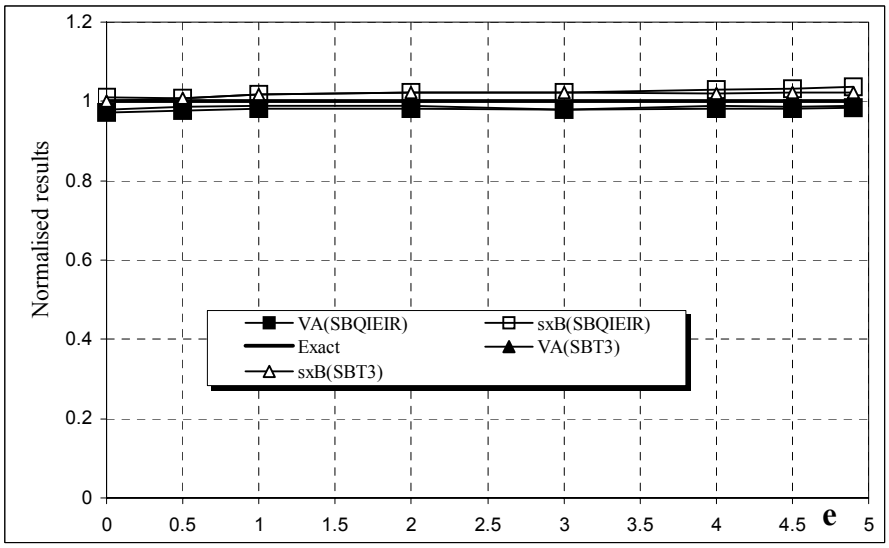


Figure 17c: Pure bending of a cantilever. Normalized results Rotation is free at 1 and 2 (Figure 15c)

The Figure 17c confirms the good performance and the stability of **SBQIEIR** element.

6.2 Allman's cantilever beam

In the following example, it is a question of evaluating the vertical displacement V_A at the free end of a short cantilever (Figure 18) subject to a uniform vertical load (resultant W).

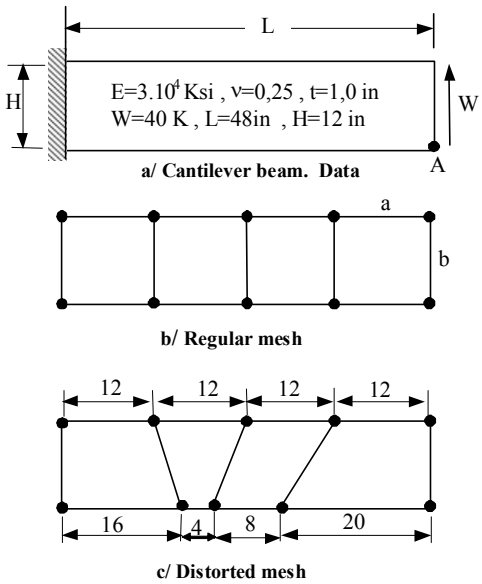


Figure 18. Allman's cantilever beam; Data and mesh

This test is considered by many researchers as a tool to validate the plane elements. It makes it possible to examine the aptitude of an element of the membrane type to simulate the problems dominated by bending.

The results obtained for the two cases of a mesh (regular and distorted) are listed on table 2.

In the case of the regular mesh (Figure 18b), the results obtained for **SBQIEIR** are powerful and comparable with the analytical solution given by the theory of the beams. For the case of the distorted mesh (Figure 18c), the very good performance of element **SBQIEIR** is confirmed. The corresponding results are more precise than the results of the other elements (see Table 2).

Table 2. Allman's short cantilever beam. Normalised vertical displacement at point A.

Formulation/ Element	Mesh	Normalized vertical displacement at A
Q4	Reg.	0,679
Q4	Dist.	0,596
PS5β	Reg.	0,978
PS5β	Dist.	0,925
AQ	Reg	0,918
AQ	Dist.	0,947

MAQ	Reg	0,918
MAQ	Dist.	0,952
QR4b	Reg.	0,978
QR4b	Dist.	0,977
Q4S	Reg	0,978
Q4S	Dist.	0,976
07β [9]	Reg	0,978
07β [9]	Dist.	0,978
Type-mixte [11]	Reg	0,969
Type-mixte [11]	Dist.	0,863
MacNeal et Harder [13]	Reg	0,959
MacNeal et Harder [13]	Dist.	0,838
Allman [14]	Reg	0,852
Allman [14]	Dist.	-
SBRIEIR [6]	Reg	0,987
SBQIEIR	Dist.	0,983
Exact solution [15]**:**		**1,000 (0,3553)**

6.3 MacNeal's elongated cantilever beam

Let us consider the example of the elongated cantilever beam of MacNeal and Harder [13], with rectangular section (6 x 0.2 x 1) deformed in pure bending by one moment at the end (M=10) and by a load applied at the free end (P=1).

The cantilever is modeled by six membrane elements rectangular (figure 19a), trapezoidal (Figure 19b) and parallelogram (Figure 19c).

MacNeal [16] affirms that the trapezoidal shape of the membrane finite elements with four nodes without degrees of freedom of rotation (with linear fields) generates a locking even if these elements pass the patch-test. This problem is known as "trapezoidal locking"

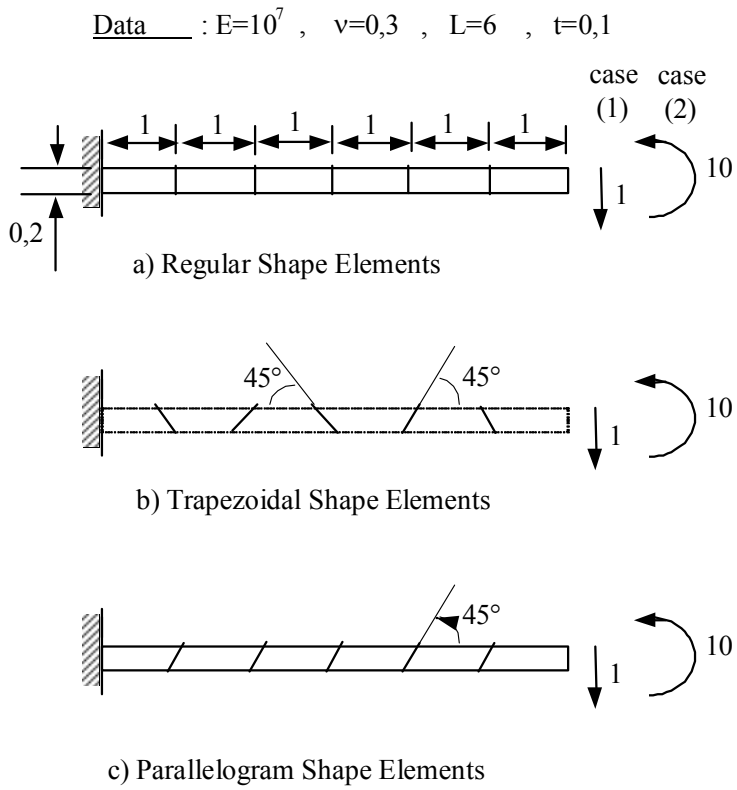


Figure 19. MacNeal's elongated beam subject to (1) end shear and (2) end bending.

NOTE: This rule does not apply to the finite elements based on assumed strain.

The results obtained for **SBQIEIR** are compared with those obtained with other known quadrilateral elements (Table 3).

Table 3. Normalized tip deflection for MacNeal's elongated beam

Element	Pure bending			End shear		
	Regular	**Trapezoidal**	**Parallel**	**Regular**	**Trapezoidal**	**Parallel**
Q4	0,093	0,022	0,031	0,093	0,027	0,034
PS5β [17]	1,000	0,046	0,726	0,993	0,052	0,632
AQ	0,910	0,817	0,881	0,904	0,806	0,873
MAQ	0,910	0,886	0,890	0,904	0,872	0,884
Q4S [13]	-	-	-	0,993	0,986	0,988
07β [9]	1,000	0,998	0,992	0,993	0,988	0,985
SBQIEIR	0,976	0,978	0,982	0,963	0,944	0,941
Theory of beams		**1,000** **(0,270)**			**1,000** **(0,1081)**	

The results obtained for elements Q4 and PS5β (table 3) show well the problem of trapezoidal locking announced by MacNeal et al. [16].

7. CONCLUSIONS

The interest of the subroutine of integration "**INTEGRATION**" was shown. The results demonstrate the stability of the element "**SBQIEIR**" whatever the value "e". This is partly explained probably by the nature of analytical integration carried out. Unfortunately, this last (**SBQIEIR)** does not pass the patch test.

REFERENCES

1. Ashwell D.G. and Sabir A.B. "A new cylindrical shell finite element based on simple independent strain functions". *Int. J. Mech. Sci*. **14**(1972) 171-183.
2. Cantin G. and Clough R.W. "A curved cylindrical shell finite element" *AIAA. Journal* **6**(1968) 1057-1062.
3. Sabirm A.B. and Lock A. "A curved cylindrical shell finite element" *Int. J. Mech. Sci*, **14**(1972) 125-135.

4. Sabir A.B. "A new class of finite elements for plane elasticity problems" CAFEM7 7th Int. Conf. Struct. Mech in reactor technology Chicago (1983).
5. Sabir A.B. and Salhi H.Y. "A strain based finite element for general plane elasticity in polar coordinates", Res. Mechanica 19, pp. 1-16. (1986).
6. Sabir A.B. "A rectangular and triangular plane elasticity element with drilling degrees of freedom" Chapter 9 in proceeding of the 2nd international conference on variational methods in engineering, Southampton University, Springer-Verlag, Berlin, pp. 17-25. 1985.
7. Sabir A.B. and Ramadhani F. "A shallow shell finite element for general shell analysis" Variational methods in engineering, Proceedings of the 2nd International Conference, University of Southampton England (1985).
8. Sabir A.B. and Sfendji A. Triangular and Rectangular plane elasticity finite elements. Thin-walled Structures 21. pp. 225-232, 1995.
9. Sze K.Y., Chen W. and Cheung Y.K. An efficient quadrilateral plane element with drilling degrees of freedom using orthogonal stress modes, C.S., No. 5, **42**(1992) 695-705.
10. Taylor R.L. Simo J.C. Zienkiewicz O.C. and Chan A.C. The patch test: A Condition for Assessing Finite Element Convergence, *International Journal For Numerical Methods In Engineering*, **22**(1986) 39-62.
11. Batoz J.L. and Dhatt G. Modélisation des structures par éléments finis, Solides élastiques, Eds Hermès, Paris, **1**(1990).
12. Ibrahimbegovic A, Taylor R.L. and Wilson E.L. A robust quadrilateral membrane finite element with drilling degrees of freedom, *International Journal For Numerical Methods In Engineering*, **30**(1990) 445-457.
13. MacNeal R.H. and Harder R.L. A refined four-noded membrane element with rotational degrees of freedom, C.S., **28**(1988) 75-84.
14. Allman D.J., A quadrilateral finite element including vertex rotations for plane elasticity analysis, *International Journal For Numerical Methods In Engineering*, **26**(1988) 717-730.
15. Timoshenko S. and Goodier J. N., Theory of Elasticity, Mc Graw-Hill, New York.
16. MacNeal R.H. A theorem regarding the locking of tapered four-noded membrane elements, *International Journal For Numerical Methods In Engineering*, **24**(1987) 1793-1799.
17. Pian T.H. and Sumihara K. Rational approach for assumed stress finite elements, *International Journal For Numerical Methods In Engineering*, **20**(1984) 1685-695.

APPENDIX

```
SUBROUTINE INTEGRATION(TI,X,Y)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (PETIT=1.0D-15)
      COMMON /COEFFICIENT/ A1,B1,A2,B2,A3,B3
      DIMENSION TI(15),ICORD(3),X(3),Y(3),dx(3),dy(3)
      DIMENSION TI1(15),TI2(15)
      do 87 iu=1,15
      ti1(iu)=0.
      ti2(iu)=0.
87    continue
      CALL FORM_ICORD(X,Y,ICORD)
```

```
      CALL COEFF(ICORD,X,Y,DX,DY)
c'***************     INTEGRAL No.1      ***************'
      if(abs(dx(1)).GT. PETIT)then
      CALL EXPRES (x(icord(1)),x(icord(2)),a1,b1,a3,b3,1,ti1)
      endif
c***************     INTEGRAL No.2      ****************
      if(abs(dx(1)).GT. PETIT)then
      CALL ICORD_A1(icord,ICO1)
      if(icord(3).EQ.ICO1)then
      CALL EXPRES (x(icord(2)),x(icord(3)),a1,b1,a2,b2,2,ti2)
      endif
      endif
      if(abs(dx(2)).GT. PETIT)then
      CALL ICORD_A3(icord,ICO1)
      if(icord(3).EQ.ICO1)then
      CALL EXPRES (x(icord(2)),x(icord(3)),a2,b2,a3,b3,2,ti2)
      endif
      endif
c***************     INTEGRAL No3      **************************
      if(abs(dx(3)).GT. PETIT)then
      CALL ICORD_A2(icord,ICO1)
      if(icord(3).EQ.ICO1)then
      CALL EXPRES (x(icord(2)),x(icord(3)),a2,b2,a3,b3,2,ti2)
      endif
      endif
      CALL TII(ti1,ti2,ti)
      RETURN
      END

      SUBROUTINE FORM_ICORD(X,Y,ICORD)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(3),Y(3),ICORD(3),XY(2,3)

      XY(1,1)=X(1)
      XY(1,2)=X(2)
      XY(1,3)=X(3)
      ICORD(1)=1
      ICORD(2)=2
      ICORD(3)=3
      DO 100 i=1,3
      DO 200 j=1,i
      IF(xy(1,i)  .LT. xy(1,j)) then
      temp=xy(1,i)
      xy(1,i)=xy(1,j)
      xy(1,j)=temp
      itemp=icord(i)
      icord(i)=icord(j)
      icord(j)=itemp
      endif
200   continue
```

```
100   continue
      return
      end

      SUBROUTINE ICORD_A1(ICORD,ICO1)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER ICORD(3)

      ICO1=ICORD(1)+1
      IF(ICO1 .EQ. 4) ICO1=1
      return
      end

      SUBROUTINE ICORD_A2(ICORD,ICO1)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER ICORD(3)

      ICO1=ICORD(1)+2
      IF(ICO1 .EQ. 4) ICO1=1
      IF(ICO1 .EQ. 5) ICO1=2
      return
      end

      SUBROUTINE ICORD_A3(ICORD,ICO1)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER ICORD(3)

      ICO1=ICORD(1)-1
      IF(ICO1 .EQ. 0) ICO1=3
      return
      end
      SUBROUTINE COEFF(ICORD,X,Y,DX,DY)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(3),Y(3),dx(3),dy(3)
      INTEGER ICORD(3)
       PARAMETER (PETIT =1.0D-15)
      COMMON /COEFFICIENT/ A1,B1,A2,B2,A3,B3
C     CALCUL Y1
      CALL ICORD_A1(icord,iCO1)
      dx(1)=x(icord(1))-x(iCO1)
      dy(1)=y(icord(1))-y(iCO1)
      IF(abs(dx(1)) .GT. PETIT) THEN
      a1=dy(1)/dx(1)
      b1=y(icord(1))-a1*x(icord(1))
      ELSE
      a1=0
      b1=0
      ENDIF
C     CALCUL Y2
      CALL ICORD_A2(icord,iCO2)
```

```
      dx(2)=x(ICO1)-x(iCO2)
      dy(2)=y(ICO1)-y(iCO2)
      IF(abs(dx(2)) .GT. PETIT) THEN
      a2=dy(2)/dx(2)
      b2=y(ICO1)-a2*x(ICO1)
      ELSE
      b2=0
      a2=0
      ENDIF
C     CALCUL Y3
      CALL ICORD_A3(icord,ICO1)
      dx(3)=x(icord(1))-x(ICO1)
      dy(3)=y(icord(1))-y(iCO1)
      IF(abs(dx(3)) .GT. PETIT) THEN
      a3=dy(3)/dx(3)
      b3=y(icord(1))-a3*x(icord(1))
      ELSE
      b3=0
      a3=0
      ENDIF
      RETURN
      END
      SUBROUTINE EXPRES(XZ,XXZ,A,B,C,D,IY,TI)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       COMMON /COEFFICIENT/ A1,B1,A2,B2,A3,B3
      DIMENSION TI(15),T(2)
      DIMENSION AX1(6),BX1(6)
      INTEGER IC(6)
      I=1
      DO 160 kK=1,15
160     TI(kK)=0.

      DO 1 IB=0,4
      II=IB+1
      GO TO (10,11,12,13,14),II
10     IC(1)=1
      IC(2)=1
      GO TO 100
11     IC(1)=1
      IC(2)=2
      IC(3)=1
      GO TO 100
12     IC(1)=1
      IC(2)=3
      IC(3)=3
      IC(4)=1
      GO TO 100
13     IC(1)=1
      IC(2)=4
      IC(3)=6
```

```
      IC(4)=4
      IC(5)=1
      GO TO 100
14     IC(1)=1
      IC(2)=5
      IC(3)=10
      IC(4)=10
      IC(5)=5
      IC(6)=1
100   CONTINUE
      DO 2 IA=0,4-IB
      T(iy)=0.
       DO 3 K=1,IB+2
       AX1(K)=C**(K-1)*D**(IB+2-K)
       BX1(K)=A**(K-1)*B**(IB+2-K)
       AA1=(IC(K)*(AX1(k)-BX1(k))*(XXZ**(IA+K)-XZ**(IA+K)))/(K+IA)
       AA2=AA1/(IB+1)
       T(IY)=T(IY)+AA2
3     CONTINUE
       TI(I)=T(IY)
      I=I+1
2     CONTINUE
1     CONTINUE
      RETURN
      END
      subroutine tii(ti1,ti2,ti)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      dimension ti(15),ti1(15),ti2(15)
      do 17 j=1,15
17       ti(j)=0.
      do 10 i=1,15
10     ti(i)=ti1(i)+ti2(i)
      return
      End
```