

Nodally Integrated Finite Element Formulation for Mindlin-Reissner Plates

D. A. Simoes and T. A. Jadhav

Abstract— This work describes a nodally integrated finite element formulation for plates under the Mindlin-Reissner theory. The formulation makes use of the weighted residual method and nodal integration to derive the assumed strain relations. An element formulation for four-node quadrilateral elements is implemented in the nonlinear finite element solver Abaqus using the UEL user element subroutine. Numerical tests are carried out on the new element and the results are presented.

Index Terms— Abaqus, CAE, element formulation, finite element analysis, four-node quadrilateral element, nodal integration, UEL

1 INTRODUCTION

IN nodal integration, the numerical integration is carried out at the corner nodes of the element, rather than at quadrature points inside the element. Research on finite element formulations emerged in the beginning of the 1970s, while nodal integration in FEA is a relatively recent development, with the first works beginning in the year 2000 describing its application to tetrahedral elements. Wolff [1] discussed nodally integrated finite elements, focussing on solid elements. It was shown that nodally integrated elements show good convergence compared to full integration when applied to incompressible media. Nodal integration is also attractive because of its applicability to meshfree methods as described by Quak et al [2], who compared gauss integration and nodal integration for meshless analyses. This is because, as described in this formulation, the method focuses on a nodal patch, rather than on an elemental area, as conventional elements do. In large deformation processes, for example in extrusion and injection moulding, finite elements can suffer from excessive mesh deformation. Meshless methods are well suited to avoid these problems. However, the obstacle that must be overcome in developing nodally integrated elements is that the quantities at nodal points are not continuous, and the nodes are shared among multiple elements. These elements also suffer from shear locking, being based on the Mindlin-Reissner theory for thick plates.

Wang and Chen [3] described a Mindlin-Reissner plate formulation with nodal integration. Castelazzi and Krysl [4] introduced Reissner-Mindlin plate elements with nodal integration in which the nodal integration is derived from the a priori satisfaction of the weighted residuals. They have applied the same formulation, called the NIPE technique to a nine-node quadrilateral plate element [5]. However, in this case, a variational energy method is used instead of the weighted residual method in their previous work. The resulting element is found to be an improvement over the earlier one. Giner, E. et al [6] and Park, K. et al [7], have implemented special elements for fracture mechanics in Abaqus UEL.

Abaqus is an extremely capable nonlinear solver with a large element library that allows analysis of even the most complex structural problems. The default element formulations in Abaqus are accurate, robust and reliable enough, having been extensively tested. However, situations arise in which the Abaqus element library would not serve the purpose, and the user would like to define their own elements, such as:

- Modelling non-structural physical processes that are coupled to structural behaviour
- Applying solution-dependent loads
- Modelling active control mechanisms
- Studying the behaviour of proposed formulations

For example, elements can be developed to function as control or feedback mechanisms in an analysis that consists of regular elements. Moreover, it is easier to maintain and port a subroutine than to do the same for a complete finite element program.

The following are some desirable reliability criteria for finite element formulations [8]:

1. The element formulation must not incorporate numerically adjusted factors which can be adjusted to make the element accurate for one class of problems, but not for other types of problems.
2. The element formulation must not contain spurious rigid body modes.
3. The element should not lock in thin plate / shell analyses.
4. The element formulation must satisfy patch tests.
5. The element accuracy should be insensitive to element distortions and changes in material properties.

In this paper, we describe an element formulation which makes use of nodal integration and the weighted residual method to derive the assumed strain operators. This is applied to two elements: a four-node quadrilateral element and a three-node triangular element. A four-node quadrilateral element is then implemented in the nonlinear finite element solver Abaqus by writing a user element subroutine (UEL). In sections 3 and 4, we follow the same procedure described in [4].

D. A. Simoes and T. A. Jadhav,
Department of Mechanical Engineering, Sinhgad College of Engineering,
University of Pune, India

2 BASIC GOVERNING EQUATIONS

In this section, the basic equations of a Reissner-Mindlin plate theory are summarised. We denote the domain of the plate, which is its volume by V and the thickness by t .

$$V = \{(x, y, z) \in \mathbb{R}^3 \mid z \in [-t/2; t/2], (x, y) \in A \in \mathbb{R}^2\} \quad (1)$$

where A is the area of the plate and $C = \partial A$ is its boundary. The Cartesian components of the displacement in three-dimensional space are

$$u_x = z\theta_y, \quad u_y = -z\theta_x, \quad u_z = w \quad (2)$$

θ_y and θ_x are the rotations of the midsurface normal about the Cartesian x and y axes and w is the displacement in z . The functions θ_y, θ_x, u_z are the unknowns that we would like to determine. This is shown in Figs. 1 and 2. θ_z is the drilling degree of freedom.

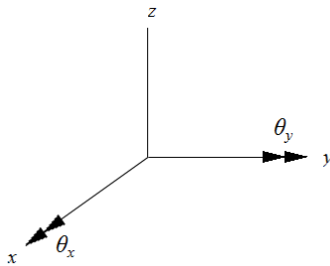


Fig. 1. Notation for rotation components of a midsurface normal

Mindlin plate theory assumes that transverse shear deformation occurs, and the strains are given as follows:

$$\begin{aligned} \epsilon_x &= z \frac{\partial \theta_y}{\partial x} \\ \epsilon_y &= -z \frac{\partial \theta_x}{\partial y} \\ \gamma_{xy} &= z \left(\frac{\partial \theta_y}{\partial y} - \frac{\partial \theta_x}{\partial x} \right) \\ \gamma_{yz} &= \partial w / \partial y - \theta_x \\ \gamma_{zx} &= \partial w / \partial x - \theta_y \end{aligned} \quad (3)$$

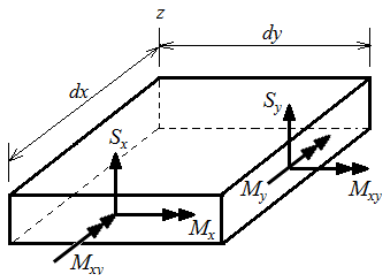


Fig. 2. Moments and transverse shear forces on a differential element of a plate

We express the three-dimensional displacement vector in terms of the mixed component vector of the generalised displacements \tilde{u}

$$\tilde{u} = [w, \theta_x, \theta_y]^T \quad (4)$$

The curvatures and transverse shear strains are given by

$$\eta^b = \begin{bmatrix} \theta_{y,x} \\ \theta_{x,y} \\ -\theta_{y,y} + \theta_{x,x} \end{bmatrix} \quad (5)$$

$$\eta^s = \begin{bmatrix} -\theta_y - w_{,x} \\ \theta_x - w_{,y} \end{bmatrix} \quad (6)$$

The compatibility equations can be written as bending strains,

$$\epsilon^b = z\beta^b \tilde{u} \quad (7)$$

curvatures,

$$\eta^b = \beta^b \tilde{u} \Rightarrow \epsilon^b = z\eta^b \quad (8)$$

shear strains,

$$\epsilon^s = \beta^s \tilde{u} = \gamma = \eta^s \quad (9)$$

where β^b and β^s are the gradient matrices

The constitutive equations for plane stress for bending,

$$\sigma^b = D^b \epsilon^b = z D^b \eta^b \quad (10)$$

for shear,

$$\sigma^s = D^s \epsilon^s = kG\eta^s \quad (11)$$

For a homogeneous, linear elastic and isotropic material,

$$D^b = \begin{bmatrix} D & \nu D & 0 \\ \nu D & D & 0 \\ 0 & 0 & \frac{(1-\nu)D}{2} \end{bmatrix} \quad (12)$$

Here, the bending rigidity

$$D = \frac{Et^3}{12(1-\nu^2)} \quad (13)$$

E is the Young's modulus and ν is the Poisson's ratio,

$$D^s = kGt \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (14)$$

G is the shear modulus. k is the shear correction factor, which accounts for the parabolic z -direction variation of the transverse shear stress, and kt can be regarded as the effective thickness for transverse shear deformation. It is taken to be $5/6$ throughout.

The plate is loaded by boundary load $\bar{b} = [0, 0, \bar{b}_2(x, y)]^T$

The balance equation obtained after integrating through the thickness of the plate is

$$\beta^{bT} M + \beta^{sT} S + t\bar{b} = 0 \quad (15)$$

and the resultants,

$$M = \int_{-t/2}^{t/2} z\sigma^b dz = D^b \eta^b \quad (16)$$

$$S = \int_{-t/2}^{t/2} \sigma^s dz = tkG\eta^s = D^s \eta^s \quad (17)$$

3 WEIGHTED RESIDUAL FORMULATION

The method of weighted residuals is applied to obtain the weak form of the Mindlin-Reissner governing equations. The weak form will be constructed in a manner that facilitates a

locking-free FE discretisation.

Nodal quadrature cannot directly process the integrands from the individual elements to which the node is common, because each contributing basis function derivative will be multi-valued at the node (discontinuous across the element edges). For the process of nodal quadrature, the basis functions derivatives, or, equivalently, the strains should be computed using a special strain-displacement operator, so we use an assumed-strain method.

A separate residual statement is kept for each weakly enforced boundary condition. The three residual equations are as follows:

- The balance equation residual:

$$r_B = \int_A \tilde{\boldsymbol{\eta}}^T (\boldsymbol{\beta}^{bT} \mathbf{M} + \boldsymbol{\beta}^{sT} \mathbf{S} + t\bar{\mathbf{b}}) dA = 0 \quad (18)$$

- The natural boundary condition residual :

$$r_c = \int_{C,1} \tilde{\eta}_1 (S_{nz} - \bar{S}_{nz}) dC + \int_{C,2} \tilde{\eta}_2 (M_{nn} - \bar{M}_{nn}) dC + \int_{C,3} \tilde{\eta}_3 (M_{ns} - \bar{M}_{ns}) dC = 0 \quad (19)$$

- The kinematic equation residuals:

$$r_{kb} = \int_A (\bar{\boldsymbol{\beta}}^b \tilde{\boldsymbol{\eta}})^T \mathbf{D}^b (\bar{\boldsymbol{\beta}}^b \tilde{\mathbf{u}} - \boldsymbol{\beta}^b \tilde{\mathbf{u}}) dA = 0 \quad (20)$$

$$r_{ks} = \int_A (\bar{\boldsymbol{\beta}}^s \tilde{\boldsymbol{\eta}})^T \mathbf{D}^s (\bar{\boldsymbol{\beta}}^s \tilde{\mathbf{u}} - \boldsymbol{\beta}^s \tilde{\mathbf{u}}) dA = 0 \quad (21)$$

Here $\tilde{\boldsymbol{\eta}}$ is the test function (generalised displacement, as indicated by the tilde '~'), which is assumed to vanish along the portions of the boundary where essential boundary conditions are prescribed, in order to eliminate the unknown reactions:

$[\tilde{\boldsymbol{\eta}}]_i = 0$, where the i th component of the generalised displacement is prescribed on C .

Next, the assumed strain operators $\bar{\boldsymbol{\beta}}^b$ and $\bar{\boldsymbol{\beta}}^s$ will be derived from condition that they will make the kinematic residuals (20) and (21) identically zero. The process will be the same for both $\bar{\boldsymbol{\beta}}^b$ and $\bar{\boldsymbol{\beta}}^s$. Hence, we will describe the derivation of $\bar{\boldsymbol{\beta}}^s$ only.

4 PATCH AVERAGED ASSUMED STRAIN

We now derive the strain-displacement assumed-strain matrices for meshes of isoparametric three-node triangles, four-node quadrilaterals, or possibly a combination of both element types.

We introduce the finite element approximation $\tilde{\mathbf{u}} = \sum_I N_I \tilde{\mathbf{u}}_I$ and $\tilde{\boldsymbol{\eta}} = \sum_I N_I \tilde{\boldsymbol{\eta}}_I$.

We have introduced matrices $\mathbf{B}_j^s = \boldsymbol{\beta}^s(N_j)$ and the strain-displacement matrices $\bar{\mathbf{B}}_i^s$ (not yet specified), that are used to obtain the assumed strains as

$$\boldsymbol{\gamma} = \sum_I \bar{\mathbf{B}}_i^s \tilde{\mathbf{u}}_I \quad (22)$$

We obtain

$$\int_A \bar{\mathbf{B}}_i^{sT} \mathbf{D}^s (\bar{\mathbf{B}}_j^s - \mathbf{B}_j^s) dA = 0 \quad \forall J \quad (23)$$

Fig. 3 explains the indices. For a particular I , we note that in order to formulate strictly local operations, we should only consider the strain-displacement matrices \mathbf{B}_j^s defined within the elements e_r , $r = 1, \dots, 5$ connected to node I . The index J then ranges over $J = J_q$, $q = 1, \dots, 6$ and $J = I$. Therefore, we replace $\forall J$ in (23) with the limited range $J \in \text{nodes}(\text{elems}(I))$; the term $\text{nodes}(e)$ refers to nodes of the element e , the term $\text{elems}(I)$ refers to the elements connected to the node I , and the term $\text{nodes}(\text{elems}(I))$ refers to the union of the nodes of the elements connected to the node I . Therefore we use the integrals

$$\int_A \bar{\mathbf{B}}_i^{sT} \mathbf{D}^s (\bar{\mathbf{B}}_j^s - \mathbf{B}_j^s) dA = 0 \quad (24)$$

$J \in \text{nodes}(\text{elems}(I))$

for any particular node I .

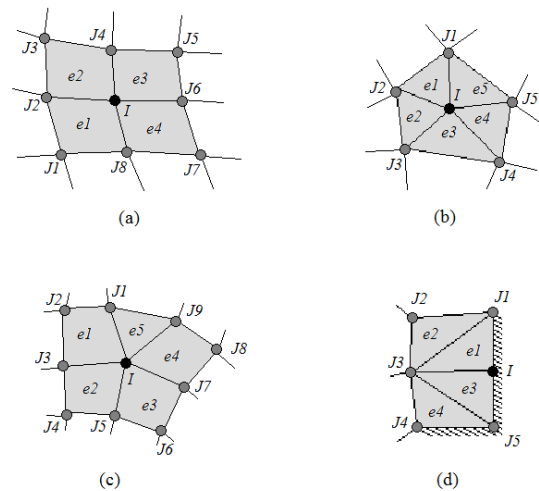


Fig. 3. Nodal patch illustrated on (a) four-node quadrilateral mesh; (b) three-node triangular mesh; (c) mixed mesh; and (d) a three-node triangular mesh with boundary conditions

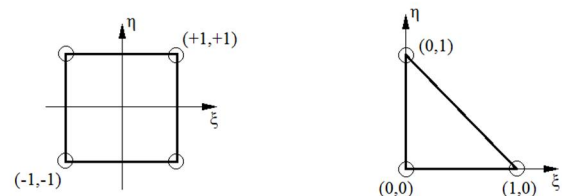


Fig. 4. Nodal quadrature locations for the triangular and quadrilateral elements

Fig. 4 shows the quadrature rules for two of the simplest element formulations discussed. The weights are $1/3$ for each node of a triangular element and 1 for each node of quadrilateral elements. Using numerical quadrature, the integral (24) is replaced by the following double sum:

$$\sum_e \sum_{K \in \text{nodes}(e)} J(\mathbf{x}_K) W_K [\bar{\mathbf{B}}_j^s \mathbf{D}^s(\mathbf{x}_K)] = \mathbf{0} \quad (25)$$

$J \in \text{nodes}(\text{elems}(l))$

where e ranges over all the elements in the mesh, K ranges over all the quadrature points in the element. Here, the quadrature points coincide with the nodes. \mathbf{x}_K is the location of the quadrature point (node), $J(\mathbf{x}_K)$ is the Jacobian of the isoparametric mapping, and W_K is the weight of the quadrature point.

Finally we are led to the definition of the assumed-strain nodal matrix as a weighted average of the elemental strain displacement matrices.

$$\bar{\mathbf{B}}_j^s = \frac{\sum_{e \in \text{elems}(K)} J(\mathbf{x}_K) W_K \mathbf{B}_j^s(\mathbf{x}_K)}{\sum_{e \in \text{elems}(K)} J(\mathbf{x}_K) W_K} \quad (26)$$

Thus, constructing the nodal strain-displacement matrices as averages of the strain-displacement matrices from the connected elements will satisfy the kinematic residual statement, enabling nodal quadrature in the process. If the integration point K lies at a multi-material interface, the above derivation is not valid since the material stiffness matrix \mathbf{D}^s is multi-valued at \mathbf{x}_K . Under these circumstances, one must split the sum of the elements into two or more groups, one for each material. Then for each group, the material stiffness matrix will be single-valued.

Since the element is integrated at the nodal points, the same function can be used to interpolate the quantities within the element as well as to describe the deflection. Thus, the element is an isoparametric element.

5 ABAQUS IMPLEMENTATION

In this section, the main features related to the Abaqus implementation through the user subroutine UEL are discussed. The following information must be passed on to Abaqus from the user element code [9]:

- Strain – displacement matrix B
- Constitutive law matrix D
- Element stiffness matrix k

$$k = \int_V \mathbf{B}^T \mathbf{D} \mathbf{B} t dV \quad (27)$$

- Element internal force vector F_i

$$\mathbf{F}_i = \int_V \mathbf{B}^T \boldsymbol{\sigma} dV \quad (28)$$

- Numerical integration

These have to be coded in FORTRAN. In the subroutine, nodal coordinates (COORDS), nodal displacements in the global coordinates (\mathbf{u}) and the material properties defined in the input file (PROPS) are available, while the right-hand-side vector (RHS) and the Jacobian matrix (AMATRIX) of an element need to be defined. In addition, the state dependent variables (SVARS) can be updated at the end of each iteration.

Each element has four nodes, and each node has three degrees of freedom (w, θ_x, θ_y). First, the global nodal displacements are converted into local displacements.

Through the *USERELEMENT command, we define 4-node user elements. We also provide three real-valued user properties: the element thickness t , Young's modulus E and Poisson's

ratio ν . This is done through the command *UEL PROPERTY [11]. Since there is no way to display user elements in post-processors, all the user elements are duplicated with overlay elements with a very low stiffness. This enables the displacement to be visualised in the post-processor.

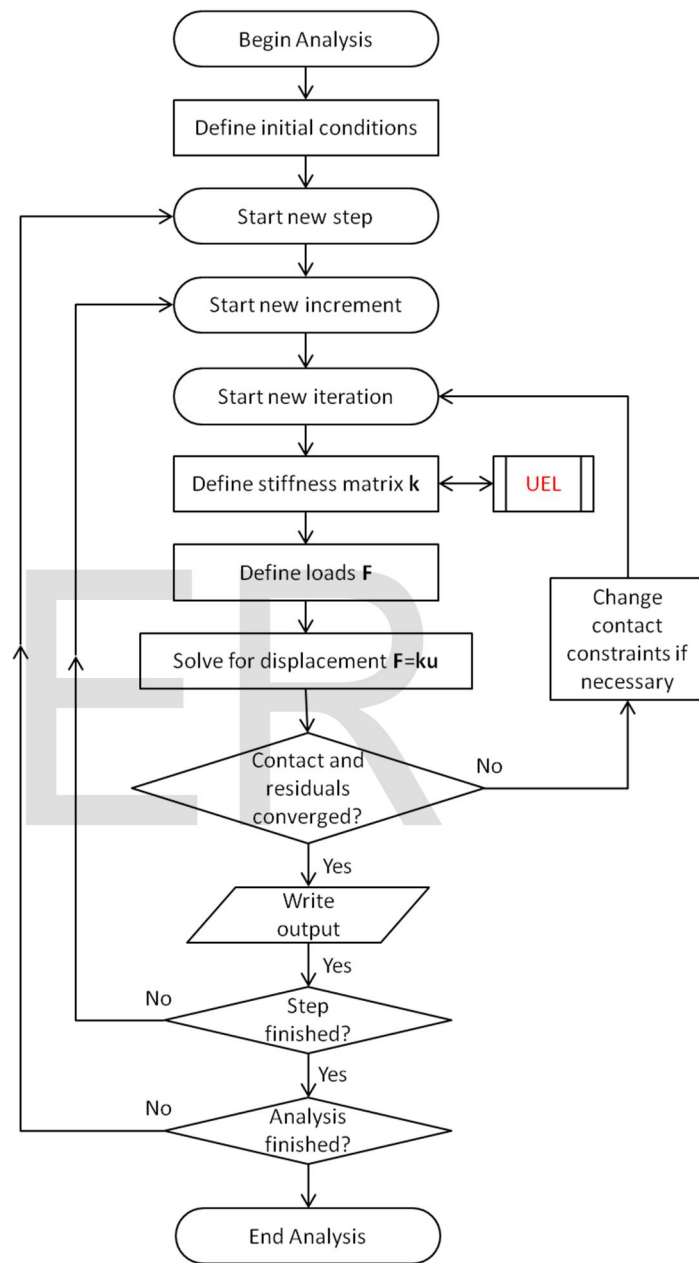


Fig. 5. Global flow in Abaqus

The name of the FORTRAN file containing the UEL code must be added to the normal command for launching an Abaqus job as follows:

```
abaqus job=<job-id> user=<name-of-file>
```

Example:

```
abaqus job=clamped_strip.inp user=NIP-Q4.for
```

User elements in Abaqus cannot form part of a contact surface. However, this problem can again be solved by making use of the duplicate overlay elements to define the contact

surface. In addition, it is necessary to make some simplifications to the formulation in order to implement it in UEL. Fig. 5 shows the flow of the algorithm used in Abaqus for solving nonlinear equations, and the interaction of the UEL with it.

6 NUMERICAL TESTS

6.1 Simply supported square plate with concentrated load

In this test, a simply supported square plate subjected to a concentrated load at the centre is simulated as shown in Fig. 6. Poisson's coefficient value is assigned in this example to make the plate incompressible and check for the occurrence of volumetric locking. The length of the plate is $L=10$ and the thickness is $t=0.2$. The material properties are given as $E=3.0 \times 10^6$ and $\nu=0.3$. The concentrated force $F=400$. One quarter of the plate is discretised with 4×4 regular mesh by taking advantage of the symmetry of the plate and applying the symmetric boundary condition available in Abaqus. Table 1 lists the vertical displacement of the centre point for various elements comparing to the analytical results. The result is compared with a series solution based on Kirchhoff's theory, considering it to be a thin plate. The displacement at the centre of the plate is given by:

$$\delta = \frac{4P}{\pi^4 D L^6} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{\sin^2\left(\frac{m\pi}{2}\right) \sin^2\left(\frac{n\pi}{2}\right)}{(m^2 + n^2)^2} \quad (29)$$

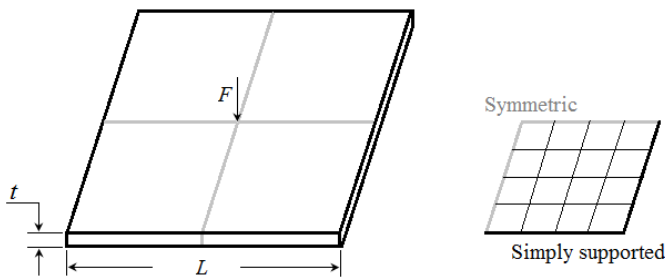


Fig. 6. Simply supported square plate with concentrated load

6.2 Clamped strip

In this example, the dimensions of the plate were 220mm x 70mm and its thickness was 3mm. The Young's modulus was 1195. The load applied at the centre was 100N. The strip is shown in Fig. 7. Considering the strip to be a beam, the analytical solution of the displacement at the centre is given by:

$$\delta = \frac{PL^3}{192EI} \quad (30)$$

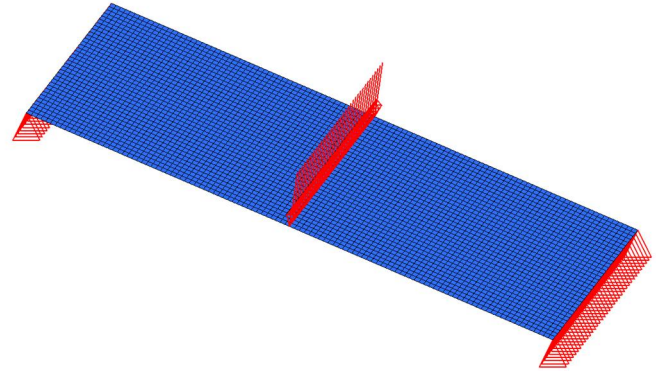


Fig. 7. Clamped strip

The results are given in Table 1.

7 RESULTS

Fig. 8 shows the displacement contours for the simply supported plate, while Fig. 9 shows the results for the clamped strip. Table 1 gives the results of displacements obtained in the numerical tests using the analytical equations, Abaqus S4 general shell elements, and the nodally integrated new formulation for plates.

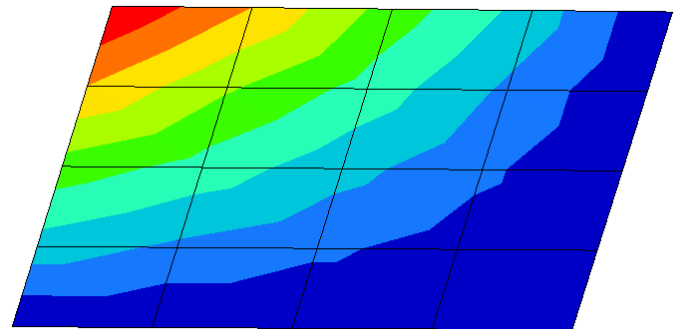


Fig. 8. Contour plot of one quarter of simply supported plate

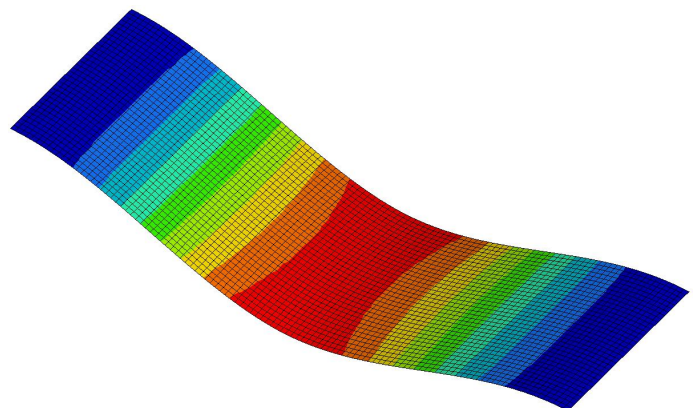


Fig. 9. Contour plot of clamped strip

TABLE 1
DISPLACEMENT RESULTS OF NUMERICAL CASES

Test	Analytical	S4 elements	New formulation
Simply supported plate	0.021138	0.021502	0.014779
Clamped strip	29.468	29.459	24.224

Based On Mindlin/Reissner Plate Theory And A Mixed Interpolation", *International Journal For Numerical Methods in Engineering*, Volume 21, (1985) pp. 367-383.

- [9] *Abaqus User Subroutines Reference Manual*, ver 6.10.
 [10] R.D. Cook, D.S. Malkus, M.E. Plesha, and R.J. Witt, *Concepts and Applications of Finite Element Analysis*, John Wiley and Sons, 2003.
 [11] *Abaqus Analysis User's Manual*, ver 6.10.

8 CONCLUSION

A new nodally integrated element formulation was described. This formulation makes use of the weighted residual method to derive the assumed strain relations. The nodal integration was performed by averaging over the nodal patch. It was found that Abaqus UELs are not suitable for implementing element formulations where the properties of more than one element are needed at a time for calculation. A new element formulation for four-node quadrilateral elements was implemented in Abaqus, and two examples were used to test its behaviour. However, the simulated results have a limited correlation with the analytical results, and cannot match the performance of the inbuilt elements, as seen in Table 1. This could be attributed to the assumptions made in order to implement in Abaqus UEL, and due to the research being at an initial stage. Improvements in the accuracy of the formulation and in the implementation in Abaqus will help to improve the results.

ACKNOWLEDGMENT

The authors would like to thank the automotive supplier Faurecia for permitting the use of their resources in order to run some of the simulations.

REFERENCES

- [1] S. Wolff, "Nodally Integrated Finite Elements", *18th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering*, K. Gurlebeck and C. Konke (eds.), Weimar, Germany, 07-09 July 2009.
 [2] W. Quak, A.H. van den Boogaard, D. González, and E. Cueto, "A Comparative Study On The Performance Of Meshless Approximations And Their Integration", *Computational Mechanics*, 48, no. 2 (2011) pp. 121-137.
 [3] D. Wang, and J. Chen, "Locking-Free Stabilized Conforming Nodal Integration For Meshfree Mindlin-Reissner Plate Formulation", *Computer Methods in Applied Mechanics and Engineering*, 193.12 (2004) pp. 1065-1083.
 [4] G. Castellazzi, and P. Krysl, "Displacement-Based Finite Elements With Nodal Integration For Reissner-Mindlin Plates", *International Journal For Numerical Methods In Engineering*, 80.2 (2009) pp. 135-162.
 [5] G. Castellazzi, and P. Krysl, "A Nine-Node Displacement-Based Finite Element for Reissner-Mindlin Plates Based on an Improved Formulation of the NIPE Approach", *Finite Elements in Analysis and Design*, 58 (2012) pp. 31-43.
 [6] E. Giner, N. Sukumar, J.E. Tarancon, and F.J. Fuenmayor, "An Abaqus Implementation Of The Extended Finite Element Method", *Engineering Fracture Mechanics*, 76, no. 3 (2009) pp. 347-368.
 [7] K. Park, and G.H. Paulino, "Computational Implementation Of The PPR Potential-Based Cohesive Model In ABAQUS: Educational Perspective", *Engineering Fracture Mechanics* (2012).
 [8] K.J. Bathe, E.N. Dvorkin, "A Four-Node Plate Bending Element