COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 50 (1985) 25–69
NORTH-HOLLAND

A TRIANGULAR MEMBRANE ELEMENT WITH ROTATIONAL DEGREES OF FREEDOM

P.G. BERGAN

Division of Structural Mechanics, The Norwegian Institute of Technology, N-7034 Trondheim-NTH, Norway

C.A. FELIPPA

Applied Mechanics Laboratory, Lockheed Palo Alto Research Laboratory, Palo Alto, CA 94304, U.S.A.

Received 9 August 1984
Revised manuscript received 7 January 1985

A new plane-stress triangular element is derived using the free formulation of Bergan and Nygård. The triangle possesses nine degrees of freedom: six corner translations and three corner normal rotations. The element is coordinate-invariant and passes the patch test for any geometry. Two free parameters in the formulation may be adjusted to optimize the behavior for in-plane bending patterns. With the recommended parameter choices the element performance is significantly better than that of the constant-strain triangle. Because of the presence of the rotational freedoms, this new element appears especially suitable as membrane component of a flat triangular element for modelling general shell structures.

1. Introduction

Plane-stress elements with translational degrees of freedom at corner points were among the earliest displacement models developed for the finite element method. The linear (constant-strain) triangle and the bilinear rectangle were developed by Turner et al. [1] and the general bilinear quadrilateral by Taig [2]. Both have been extensively used as plane-stress, plane-strain and axisymmetric-solid models for two-dimensional structures, and as ‘membrane components’ for more general three-dimensional structures.

Computational experience soon showed that these elements are excessively stiff for problems in which the response is dominated by linear strain gradients. Moreover, the overstiffness increases rapidly as the aspect ratio worsens. A classical example is that of a thin rectangular beam modelled by membrane elements and bending in its own plane. The phenomenon occurs also in rocket casing, long bent pipes, cylindrical shell roofs, etc. For conciseness we refer to this type of behavior as *in-plane bending*, although in practice it is often related to more complex deformation patterns.

To improve the in-plane bending characteristics of these elements, several approaches have been pursued within the framework of the *displacement* formulation:

(1) *Selective integration of shear energy*. This is restricted to quadrilateral elements and is not coordinate-invariant.

(2) *Addition of bubble modes.* This modification of the interior displacement field is restricted to four-node elements (quadrilaterals or triangle assemblies). It does not improve in-plane bending performance significantly. The static-condensation process becomes awkward in geometrically nonlinear problems.

(3) *Addition of 'nodeless' incompatible modes.* This method is restricted to four-node bilinear elements. As in the case of bubble modes, a static-condensation process is required. The in-plane bending performance is, however, significantly improved.

(4) *Midside freedoms.* Higher-order displacement modes are linked to deviation-from-linearity translations at midpoint nodes. Presently the most popular elements of this type are the versatile quadratic isoparametric and Lagrangian elements with 8 and 9 nodal points.

(5) *Rotational degrees of freedom.* Higher-order displacement modes are linked with corner rotations normal to the plane of the element; these are also called the *drilling freedoms.*

In summary, the first three approaches are inapplicable to triangular shapes, whereas the fourth one requires the definition of additional nodal points and thus increases the programming and data-preparation burden. The last approach, which introduces rotational degrees of freedom, is pursued here for triangular elements.

1.1. Advantages

A key advantage of using corner rotations as additional degrees of freedom in three-dimensional analysis is that such freedoms are often available 'for free'. Most general-purpose finite element programs carry six degrees of freedom at corner nodes: three displacement and three rotations. Projecting the latter onto normals to element planes provides a drilling freedom at no extra cost.

As a second advantage, the rotational freedom automatically takes care of the problem of singularities that affects coplanar and nearly coplanar shell elements when the only rotational stiffness is that associated with out-of-plane bending deformations. This problem has no satisfactory solution if six degrees of freedom are maintained at corners. The simplest and most commonly used cure is the addition of a fictitious torsional-spring stiffness matrix as described by Zienkiewicz [3]. More elaborate variations on the theme involve the addition of fictitious in-plane beams along the edges of the element. The additional stiffness, however, does not improve the in-plane bending behavior but rather makes the 'parent' element (e.g., constant-strain triangle or bilinear quadrilateral) even stiffer.

1.2. Historical background

Rectangles and quadrilaterals. Among the earliest attempts at deriving membrane elements with rotational degrees of freedom were those of Scordelis and coworkers [4–6]. Envisioned applications were folded-plate concrete roofs and box-girder bridges. The most successful studies were made by Willam [6], who derived two quadrilateral membrane elements with 12 and 16 degrees of freedom (dof). In addition to the 8 translational freedoms, the 12-dof element has 4 corner rotations whereas the 16-dof element has 4 rotations and 4 shears. The derivation makes use of averaged element rotations and decaying beam shape functions associated with the corner rotations. Both elements are nonconforming.

Variations of the Willam elements have been incorporated in the STAGS nonlinear shell analysis program [7]. Experience to date indicates that the performance is erratic, with good results generally associated with regular rectangular grids. Willam [6] in fact noted that the element formulation was non-invariant and that certain shapes resulted in rank deficiency. Undoubtedly none of these elements pass Irons' patch test [8–9] for arbitrary geometries.

Triangles. A different approach pursued at about the same time [10–11] started with the 20-dof, 10-node complete cubic expansion for the plane-stress triangle. Twelve freedoms are moved from midside nodes to the corners and expressed as the four displacement gradients $\partial u/\partial x$, $\partial u/\partial y$, $\partial v/\partial x$, $\partial v/\partial y$ at each corner. Cross-derivatives are then linearly combined to form the rotation and shear. The element derived in [10] was applied by Carr [11] to thin-shell analysis. A similar development of a higher-order triangle was reported in [12–13]. Furthermore, Tocher and Hartz [14] reported a 'hyperconforming' plane-stress triangle with displacement gradient freedoms based on the Clough–Tocher C^1-conforming plate-bending displacement functions.

These higher-order elements preserve full conformity (at least for coplanar elements) but their use leads to complex models with strain-type degrees of freedom. Matching strains enforces C^1-continuity at corner points, which is not a good idea if adjacent elements are of different material or thickness. (Condensing the strain freedoms at the element level, however, destroys conformity and reintroduces the problems noted for the incompatible quadrilaterals.) Such elements may give good results in the hand of specialists, but are not suitable for general-purpose programs.

1.3. Abandoning the traditional displacement formulation

The main difficulty with past derivations is that the inclusion of beam-type modes associated with corner rotations violates inter-element C^0-compatibility unless all displacement gradients are retained as corner freedoms, which is practically undesirable. It follows that elements with rotational freedoms fail when they are implemented by way of the traditional displacement formulation based on potential energy. In the book of Irons and Ahmad [9, p. 289] it is stated that

> "If [the element developer] intends to incorporate the third rotation into the plane stress problem, then Fig. 17.9 should help to prevent company money from being wasted. It seems to us unlikely that a recent convert can expect to be more successful in this quest than the pioneers of many years ago."

The warning is sound as far as the traditional potential-energy formulation goes. It will be shown below, however, that it is possible to construct membrane elements with rotational freedoms that satisfy the patch test and that have improved stiffness properties in in-plane bending. This may be done within the framework of the so-called *free formulation* of the finite element [15]. Using this technique we are no longer required to impose conditions of full inter-element continuity in order to satisfy convergence conditions.

2. The free formulation

2.1. The individual-element test

The standard way of testing whether a nonconforming element satisfies the convergence requirements is by way of Irons' patch test [8–9]. This test verifies whether a patch of elements can reproduce exactly the limiting conditions when it is subjected to a state of rigid-body and constant-strain modes. An equivalent way of verifying these properties is through the *individual-element test* proposed by Bergan and Hanssen [16–18]. This test is formulated as a set of linear constraint conditions that apply directly to the stiffness matrix of an individual element.

To apply this test, one begins by expanding the *limit* element displacement field as

$$\boldsymbol{u} = \boldsymbol{N}_{\mathrm{r}}\boldsymbol{q}_{\mathrm{r}} + \boldsymbol{N}_{\mathrm{c}}\boldsymbol{q}_{\mathrm{c}}, \tag{2.1}$$

where $\boldsymbol{N}_{\mathrm{r}}$ and $\boldsymbol{N}_{\mathrm{c}}$ represent a *complete* set of linearly independent rigid-body and constant-strain modes, respectively, and $\boldsymbol{q}_{\mathrm{r}}$ and $\boldsymbol{q}_{\mathrm{c}}$ are the corresponding coefficients (generalized coordinates). The relation between these coefficients and the element nodal degrees of freedom $\boldsymbol{v}$ is given by the transformation

$$\boldsymbol{v} = \boldsymbol{G}_{\mathrm{rc}}\boldsymbol{q}_{\mathrm{rc}} = \boldsymbol{G}_{\mathrm{r}}\boldsymbol{q}_{\mathrm{r}} + \boldsymbol{G}_{\mathrm{c}}\boldsymbol{q}_{\mathrm{c}}, \tag{2.2}$$

where the $\boldsymbol{G}_{\mathrm{rc}}$ matrix is easily obtained from the $\boldsymbol{N}_{\mathrm{r}}$ and $\boldsymbol{N}_{\mathrm{c}}$ modes through substitution of nodal coordinates.

The first part of the test requires that the element stiffness matrix $\boldsymbol{k}$ produce no nodal forces in a rigid-body motion:

$$\boldsymbol{k}\boldsymbol{G}_{\mathrm{r}}\boldsymbol{q}_{\mathrm{r}} = \boldsymbol{P}_{\mathrm{r}}\boldsymbol{q}_{\mathrm{r}} = \boldsymbol{0}, \tag{2.3}$$

where $\boldsymbol{P}_{\mathrm{r}}$ are the generalized forces associated with $\boldsymbol{q}_{\mathrm{r}}$. Since the latter are arbitrary,

$$\boldsymbol{k}\boldsymbol{G}_{\mathrm{r}} = \boldsymbol{P}_{\mathrm{r}} = \boldsymbol{0}. \tag{2.4}$$

On the other hand, the constant-strain patterns create stresses and tractions on the element boundaries as illustrated in Fig. 1. These tractions may be lumped to the nodal points where they create concentrated forces $\boldsymbol{t}_{\mathrm{c}}$. At an inter-element boundary, the tractions are equal and opposite for the two adjoining elements. It is clear that when the same energy-consistent lumping of boundary tractions is used for adjoining elements, the corresponding nodal forces will appear in equal and opposite pairs (see Fig. 1). Therefore, the resulting force at an interior node in a patch of elements subjected to constant-strain modes will cancel, as required by the

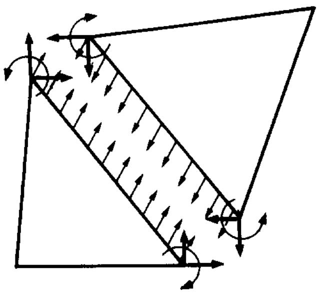

Fig. 1. Equal and opposite nodal reaction forces created by c-mode tractions along element side.

patch test. This property may be stated as a constraint condition on the the stiffness matrix of a single element:

$$\boldsymbol{k}\boldsymbol{G}_{\mathrm{c}}\boldsymbol{q}_{\mathrm{c}} = \boldsymbol{t}_{\mathrm{c}} = \boldsymbol{P}_{\mathrm{c}}\boldsymbol{q}_{\mathrm{c}}\,, \tag{2.5}$$

where $\boldsymbol{P}_{\mathrm{c}}$ are generalized forces corresponding to $\boldsymbol{q}_{\mathrm{c}}$. Since this equation must be satisfied by any $\boldsymbol{q}_{\mathrm{c}}$,

$$\boldsymbol{k}\boldsymbol{G}_{\mathrm{c}} = \boldsymbol{P}_{\mathrm{c}}\,. \tag{2.6}$$

Equations (2.4) and (2.6) make up the individual-element test.

The element nodal forces $\boldsymbol{t}_{\mathrm{c}}$ produced by constant-strain modes may be also expressed as

$$\boldsymbol{t}_{\mathrm{c}} = \boldsymbol{L}\boldsymbol{\sigma}_{\mathrm{c}}\,, \tag{2.7}$$

where $\boldsymbol{\sigma}_{\mathrm{c}}$ is an array of constant-stress components. Assuming that material properties do not vary within the element, these stressed may be calculated from the constitutive matrix $\boldsymbol{C}$ and the 'modal' strain matrix $\boldsymbol{B}_{\mathrm{c}}$ corresponding to the N_{c} modes, as

$$\boldsymbol{\sigma}_{\mathrm{c}} = \boldsymbol{C}\boldsymbol{B}_{\mathrm{c}}\boldsymbol{q}_{\mathrm{c}}\,. \tag{2.8}$$

Note that $\boldsymbol{B}_{\mathrm{c}}$ is constant over the element. By comparing (2.5), (2.6) and (2.7) it is seen that

$$\boldsymbol{P}_{\mathrm{c}} = \boldsymbol{L}\boldsymbol{C}\boldsymbol{B}_{\mathrm{c}}\,. \tag{2.9}$$

The matrix $\boldsymbol{L}$ is called the *lumping matrix*, because it transfers the constant-stress field $\boldsymbol{\sigma}_{\mathrm{c}}$ to nodal forces. This matrix plays an important role in the free formulation described below.

2.2. The free formulation

Bergan and Nygård [15] have presented a simple derivation of the stiffness matrix that implicitly satisfies the individual-element test of equations (2.4) and (2.6) even though the assumed shape functions may be nonconforming. In this *free formulation* the element stiffness matrix $\boldsymbol{k}$ is constructed as the sum of a basic stiffness $\boldsymbol{k}_{\mathrm{b}}$ and a higher-order stiffness $\boldsymbol{k}_{\mathrm{h}}$:

$$\boldsymbol{k} = \boldsymbol{k}_{\mathrm{b}} + \boldsymbol{k}_{\mathrm{h}}\,. \tag{2.10}$$

The basic stiffness is derived directly from the constitutive matrix and the lumping matrix:

$$\boldsymbol{k}_{\mathrm{b}} = \frac{1}{V}\boldsymbol{L}\boldsymbol{C}\boldsymbol{L}^{\mathrm{t}}\,, \tag{2.11}$$

where V is the volume of the element and superscript t denotes transposition. Note that $\boldsymbol{k}_{\mathrm{b}}$ is only related to the basic modes and is identical for all elements of the same type with the same freedoms. The rank of $\boldsymbol{k}_{\mathrm{b}}$ is equal to the dimension of $\boldsymbol{C}$.

The higher-order stiffness is derived from a set of higher-order modes $\boldsymbol{N}_{\mathrm{h}}$ which complete the displacement expansion

$$\boldsymbol{u} = \boldsymbol{N}_{\mathrm{r}}\boldsymbol{q}_{\mathrm{r}} + \boldsymbol{N}_{\mathrm{c}}\boldsymbol{q}_{\mathrm{c}} + \boldsymbol{N}_{\mathrm{h}}\boldsymbol{q}_{\mathrm{h}}\,, \tag{2.12}$$

so that the total number of modes becomes equal to the number of element freedoms. The element nodal displacements are related to the complete set of modes through

$$\boldsymbol{v} = \boldsymbol{G}_{\mathrm{r}}\boldsymbol{q}_{\mathrm{r}} + \boldsymbol{G}_{\mathrm{c}}\boldsymbol{q}_{\mathrm{c}} + \boldsymbol{G}_{\mathrm{h}}\boldsymbol{q}_{\mathrm{h}} = \boldsymbol{G}_{\mathrm{rc}}\boldsymbol{q}_{\mathrm{rc}} + \boldsymbol{G}_{\mathrm{h}}\boldsymbol{q}_{\mathrm{h}} = \boldsymbol{G}\boldsymbol{q}\,. \tag{2.13}$$

The total set of modes must be linearly independent with respect to the nodal freedoms. If this condition holds, $\boldsymbol{G}$ is nonsingular and may be inverted:

$$\boldsymbol{q} = \begin{Bmatrix} \boldsymbol{q}_{\mathrm{r}} \\ \boldsymbol{q}_{\mathrm{c}} \\ \boldsymbol{q}_{\mathrm{h}} \end{Bmatrix} = \boldsymbol{G}^{-1}\boldsymbol{v} = \boldsymbol{H}\boldsymbol{v} = \begin{bmatrix} \boldsymbol{H}_{\mathrm{r}} \\ \boldsymbol{H}_{\mathrm{c}} \\ \boldsymbol{H}_{\mathrm{h}} \end{bmatrix} \boldsymbol{v}\,. \tag{2.14}$$

The higher-order stiffness now follows as

$$\boldsymbol{k}_{\mathrm{h}} = \boldsymbol{H}_{\mathrm{h}}^{\mathrm{t}}\boldsymbol{k}_{q\mathrm{h}}\boldsymbol{H}_{\mathrm{h}}\,. \tag{2.15}$$

Here $\boldsymbol{H}_{\mathrm{h}}$ is the submatrix of $\boldsymbol{H} = \boldsymbol{G}^{-1}$ that relates $\boldsymbol{q}_{\mathrm{h}}$ to $\boldsymbol{v}$, and $\boldsymbol{k}_{q\mathrm{h}}$ is the generalized stiffness matrix for the higher-order modes

$$\boldsymbol{k}_{q\mathrm{h}} = \int_V \boldsymbol{B}_{\mathrm{h}}^{\mathrm{t}}\boldsymbol{C}\boldsymbol{B}_{\mathrm{h}}\,\mathrm{d}V\,, \tag{2.16}$$

where $\boldsymbol{B}_{\mathrm{h}}$ is the strain matrix for the higher-order modes. In [15] it is shown that a stiffness matrix formed by adding (2.11) and (2.15) automatically satisfies the individual-element test conditions (2.4) and (2.6).

REMARK 2.1. It is also demonstrated in [15] that the free formulation is related to a standard potential-energy (PE) formulation, the main difference being that the coupling stiffness between the basic and higher-order modes is modified in such a way that the convergence requirements are met. In fact, both formulations lead to identical results when the higher-order modes satisfy conditions of energy and force orthogonality.

REMARK 2.2. The free formulation always produces a positive-semidefinite stiffness matrix of proper rank.

REMARK 2.3. The higher-order stiffness may be scaled by a positive coefficient β without destroying the basic convergence properties. This property is exploited in Section 5 for 'tuning up' the element response to selected deformation patterns.

2.3. Relation to hybrid methods

There exists an interesting relation between the basic stiffness of the free formulation and the stress-hybrid formulation [19]. Let $\boldsymbol{\varepsilon}_c$ denote the array of constant element strains conjugate to $\boldsymbol{\sigma}_c$. The compatibility conditions may be written

$$\boldsymbol{\varepsilon}_c = \boldsymbol{C}^{-1}\boldsymbol{\sigma}_c = \boldsymbol{B}_c\boldsymbol{q}_c = \frac{1}{V}\boldsymbol{L}^{t}\boldsymbol{v}_c , \tag{2.17}$$

where the last transformation is derived from virtual-work arguments in [15]. Equation (2.17) may be combined with the force-equilibrium equation (2.7) to form the system

$$\begin{bmatrix} -V\boldsymbol{C}^{-1} & \boldsymbol{L}^{t} \\ \boldsymbol{L} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\sigma}_c \\ \boldsymbol{v}_c \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{0} \\ \boldsymbol{t}_c \end{Bmatrix} \tag{2.18}$$

in which both stresses $\boldsymbol{\sigma}_c$ and displacements $\boldsymbol{v}_c$ are retained as unknowns. This is the element equation form produced by the stress/hybrid formulation based on the Hellinger–Reissner principle when an interior constant-stress field $\boldsymbol{\sigma}_c$, which is an equilibrium field if element properties are constant and body forces are absent, is assumed. In the hybrid formulation, $\boldsymbol{L}$ is called the boundary-connection matrix (also the 'leverage' matrix [9]) and is obtained by evaluating the work of boundary tractions on an assumed boundary-displacement field. The assumed boundary displacements must satisfy inter-element compatibility but need not be related to the *interior* displacements associated with the $\boldsymbol{\sigma}_c$-stress field.

Elimination of the stress variables in (2.18) yields the stiffness equation

$$\left(\frac{1}{V}\boldsymbol{L}\boldsymbol{C}\boldsymbol{L}^{t}\right)\boldsymbol{v}_c = \boldsymbol{t}_c , \tag{2.19}$$

where the matrix in parentheses can be recognized as the basic stiffness (2.11). Thus, $\boldsymbol{k}_b$ corresponds to the stiffness produced by a constant-stress/hybrid method if the assumed boundary-displacement field produces the same $\boldsymbol{L}$.

The basic stiffness, however, is generally rank-deficient. At this point the two formulations take different paths. In the hybrid formulation, the rank deficiency is treated by injecting additional stress modes (in the conventional hybrid method these additional modes also satisfy the equilibrium equations but this requirement may be relaxed). As a result, the stress equations in (2.18) expand; instead of $V\boldsymbol{C}^{-1}$ and $\boldsymbol{L}$ a generalized flexibility matrix and an augmented boundary-connection matrix appear. To produce $\boldsymbol{k}$ the flexibility matrix, which depends on constitutive properties, must be inverted.

In the free formulation the rank deficiency is corrected by adding the higher-order stiffness (2.15) based on assumed deformation modes. The required inversion is that of the matrix $\boldsymbol{G}$, which depends only on the element geometry and not on constitutive properties. This approach is a pure displacement formulation which is more easily extendible to geometrically nonlinear problems.

3. A triangular element with rotational freedoms

3.1. A closer look at rotational freedoms

Investigators that have developed membrane elements with rotational degrees of freedom have traditionally regarded them as average rotations of the element corner sector, as illustrated in Fig. 2. In turn, this sector rotation can be interpreted as the mean value of the corner-side rotations θ_a and θ_b. In the simplest element discussed in [6, 7] it is assumed that θ_a and θ_b are both equal to the average rotation θ. Because this assumption eliminates the possibility of angular distortions, an alternative version was proposed by Willam [6] in which shear-distortion freedoms are introduced. Unfortunately, these destroy the invariance of the formulation.

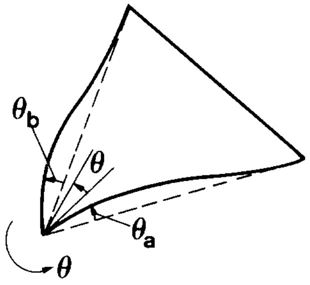

Fig. 2. Interpretation of θ as average element side rotation.

The inherent difficulties in linking the nodal rotations to the element side rotations have been clearly illustrated by Irons and Ahmad [9] for the case of the axial deformation of an element shown in Fig. 3(a). Stretching of the element gives an average corner rotation θ. However, a pure extension should not give rotations; hence this rotation is adjusted as shown in Fig. 3(b). It is seen that this correction 'bends' the element. It follows that it is not possible to satisfy both zero-rotation and no bending for a pure extensional mode.

Fig. 4 illustrates an analogous problem in the representation of pure shear. The two shaded areas in Fig. 4(a) represent two elements with sharply pointed corners meeting at a node. A shear mode is introduced in Fig. 4(b). Element I rotates with the shear angle γ whereas element II does not rotate. It is clear that if we impose the condition of same rotation for all elements meeting at a node an inconsistency results.

From the preceding discussion it follows that conventional finite elements cannot be based upon the concept of element rotations. This does not mean, however, that the concept has to be abandoned. It is still possible to use the continuum-mechanics definition of rotation

$$\theta_z = \frac{1}{2}\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right), \tag{3.1}$$

which is invariant with respect to the choice of x and y (the time derivative of θ_z is the spin). The invariance condition means that the condition $\theta \equiv \theta_z$ can be introduced even if the elements joining at a node are formed in different coordinate systems. But it is important to

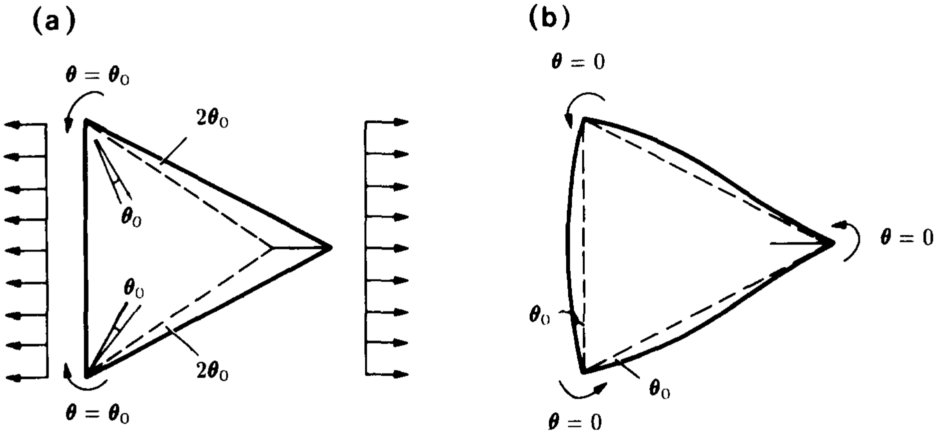

Fig. 3. Uniaxial stretching of an element. (a) Straight element sides but with corner rotatıons; (b) no rotatıons but with curved element sides.

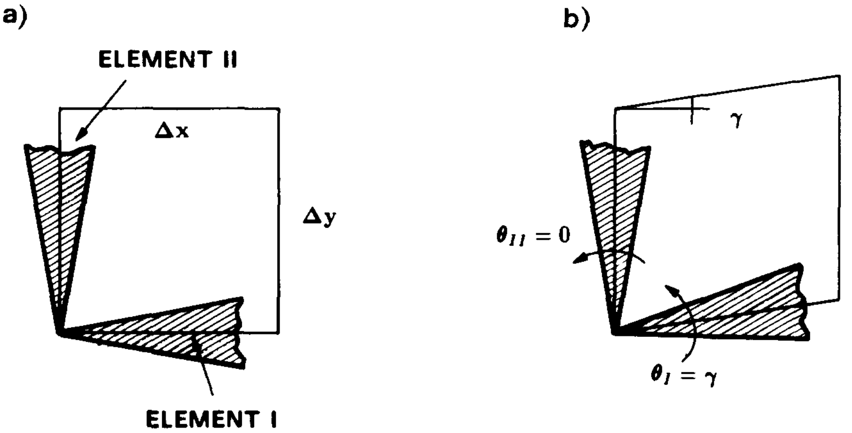

Fig. 4. Pure shear mode at a node. (a) Initıal configuratıon of two sharply pointed element corners; (b) rotation of the two element corners.

note that θ_z is *not* directly related to the element side rotations. For example, θ_z vanishes for the pure stretching mode of Fig. 3, and is the same for the two elements of Fig. 4.

REMARK 3.1. In addition to θ_z other invariants of strain type may be formed with the displacement gradients; for example $I_1 = \frac{1}{2}(\partial u/\partial x + \partial v/\partial y)$. These invariants may be used as corner freedoms of more refined nonconforming elements based on the free formulation. To avoid excessive inter-element continuity, however, such freedoms should be condensed out at the element level if they can be decoupled from the basic stiffness.

3.2. Triangle geometry

The geometry of an individual element is illustrated in Fig. 5. The triangle is referred to a local Cartesian system (x, y). Dimensionless coordinates (ξ, η) are defined by

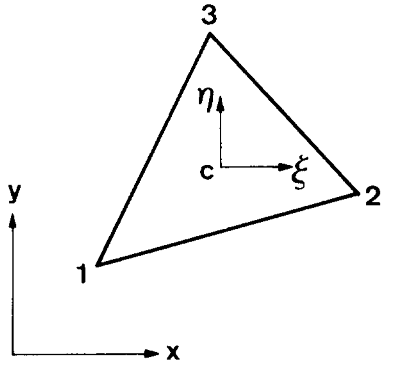

Fig. 5. Triangular element and its coordinate systems.

$$\xi = \frac{x - x_c}{\sqrt{A}} = \lambda (x - x_c), \qquad \eta = \frac{y - y_c}{\sqrt{A}} = \lambda (y - y_c) \tag{3.2}$$

where x_c and y_c are the coordinates of the centroid, A is the triangle area,

$$A = \tfrac{1}{2}(x_2 y_3 - x_3 y_2 + x_3 y_1 - x_1 y_3 + x_1 y_2 - x_2 y_1) \tag{3.3}$$

and $\lambda = 1/\sqrt{A}$ is a scaling factor with dimension of inverse length. The use of ξ and η avoids dimensional dependencies. Note that

$$\frac{\partial \xi}{\partial x} = \lambda, \qquad \frac{\partial \xi}{\partial y} = 0, \qquad \frac{\partial \eta}{\partial x} = 0, \qquad \frac{\partial \eta}{\partial y} = \lambda\,, \tag{3.4}$$

and that

$$\xi_1 + \xi_2 + \xi_3 = 0, \qquad \eta_1 + \eta_2 + \eta_3 = 0\,. \tag{3.5}$$

3.3. Basic displacement modes

The in-plane displacement field $\boldsymbol{u}$ of the element is defined by the two components u and v along x and y, respectively. The assumed field is expressed as a linear combination of nine displacement modes N_i with amplitudes q_i:

$$\boldsymbol{u} = \begin{Bmatrix} u \\ v \end{Bmatrix} = \sum_{i=1}^{9} N_i q_i = \boldsymbol{N}\boldsymbol{q}\,. \tag{3.6}$$

In accordance with the free formulation of Section 2, $\boldsymbol{u}$ is decomposed into two sets $\boldsymbol{u}_b$ and $\boldsymbol{u}_h$, which are associated with basic and higher-order modes, respectively. The basic modes for a membrane element are simply three in-plane rigid-body modes and three in-plane constant-strain modes. Using the coordinates ξ and η, the basic modes may be written

$$\boldsymbol{u}_b = \boldsymbol{u}_r + \boldsymbol{u}_c = \boldsymbol{N}_r \boldsymbol{q}_r + \boldsymbol{N}_c \boldsymbol{q}_c = \begin{bmatrix} 1 & 0 & -\eta \\ 0 & 1 & \xi \end{bmatrix} \boldsymbol{q}_r + \begin{bmatrix} \xi & 0 & \eta \\ 0 & \eta & \xi \end{bmatrix} \boldsymbol{q}_c\,. \tag{3.7}$$

From left to right, these modes are: rigid translations along the x and y directions, a positive-counterclockwise rigid-body rotation about z, the uniform stretchings $\varepsilon_x = \partial u/\partial x =$ const, $\varepsilon_y = \partial v/\partial y =$ const, and the uniform shear strain $\gamma_{xy} = \partial u/\partial y + \partial v/\partial x =$ const.

3.4. Higher-order modes

For a nine-degree-of-freedom element three higher-order modes are required. The obvious candidate functions are quadratic polynomial terms in x and y, which produce linear strain variations. There are a total of six functions for u and v, and a careful selection of three such modes has to be made. One possible choice is

$$\boldsymbol{u}_{\mathrm{h}} = \begin{Bmatrix} u_{\mathrm{h}} \\ v_{\mathrm{h}} \end{Bmatrix} = \begin{bmatrix} \xi\eta & -\frac{1}{2}\eta^2 & \xi^2 \\ -\frac{1}{2}\xi^2 & \xi\eta & \eta^2 \end{bmatrix} \boldsymbol{q}_{\mathrm{h}} . \tag{3.8}$$

The first two of these functions express shear-free pure bending along the x and y axes, whereas the last function is a shear-free 'volumetric' extension mode. Unfortunately it is clear that these higher-order modes are not invariant with respect to the choice of the reference axes. The two in-plane bending modes would be clearly beneficial for element performance as witness the effect of such modes on Wilson's incompatible rectangle [20]. The usefulness of the last mode in (3.8) appears dubious, however.

The use of three *coordinate-invariant* pure bending modes requires a deeper study of the triangle geometry. Let us assume that the modes are chosen in a rotated coordinate system $\bar{x}$, $\bar{y}$ (with corresponding dimensionless versions $\bar{\xi}$ and $\bar{\eta}$) with origin at the triangle centroid, as illustrated in Fig. 6. A pure bending mode in the $\bar{x}$, $\bar{y}$ system is

$$\bar{u} = \bar{\xi}\bar{\eta}, \qquad \bar{v} = -\tfrac{1}{2}\bar{\xi}^2 . \tag{3.9}$$

REMARK 3.2. Note the shear strain in the local system cancels:

$$\bar{\gamma}_{xy} = \frac{\partial \bar{u}}{\partial \bar{y}} + \frac{\partial \bar{v}}{\partial \bar{x}} = 0 \tag{3.10}$$

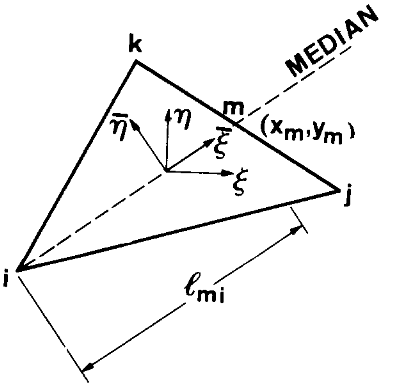

Fig. 6. Local coordinate system for the higher-order mode.

and so does $\bar{\varepsilon}_y$. The only nonvanishing strain is $\bar{\varepsilon}_x$, which is zero along the $\bar{x}$ axis. The rotation about the z axis is

$$\theta = \frac{1}{2}\left(\frac{\partial \bar{v}}{\partial \bar{x}} - \frac{\partial \bar{u}}{\partial \bar{y}}\right) = -\lambda \bar{\xi}, \tag{3.11}$$

which vanishes along $\bar{y}$.

REMARK 3.3. Since the mean value over the triangle area of the associated strains is zero, it follows that modes of the form (3.9) are *energy-orthogonal* with respect to the basic modes (3.7).

3.5. Transformation to global system

Let ψ be the angle between the ξ and $\bar{\xi}$ axes. Having

$$\begin{Bmatrix} \bar{\xi} \\ \bar{\eta} \end{Bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{Bmatrix} \xi \\ \eta \end{Bmatrix} \tag{3.12}$$

the local in-plane bending mode (3.9) may be transformed to global coordinates:

$$\begin{Bmatrix} \bar{u} \\ \bar{v} \end{Bmatrix} = \begin{bmatrix} -\sin\psi\cos\psi & \cos^2\psi - \sin^2\psi & \sin\psi\cos\psi \\ -\frac{1}{2}\cos^2\psi & -\sin\psi\cos\psi & -\frac{1}{2}\sin^2\psi \end{bmatrix} \begin{Bmatrix} \xi^2 \\ \xi\eta \\ \eta^2 \end{Bmatrix}. \tag{3.13}$$

The transformation between local and global displacement components is

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{Bmatrix} \bar{u} \\ \bar{v} \end{Bmatrix}, \tag{3.14}$$

Substitution of (3.14) into (3.13) gives the mode expressed directly in the global system:

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} -\frac{1}{2}\sin\psi\cos^2\psi & \cos^3\psi & \frac{1}{2}\sin^3\psi + \sin\psi\cos^2\psi \\ -\sin^2\psi\cos\psi - \frac{1}{2}\cos^3\psi & -\sin^3\psi & \frac{1}{2}\sin^2\psi\cos\psi \end{bmatrix} \begin{Bmatrix} \xi^2 \\ \xi\eta \\ \eta^2 \end{Bmatrix}. \tag{3.15}$$

3.6. Choosing the local systems

To get three higher-order modes of the type (3.9) we have to select three values of ψ: ψ_1, ψ_2 and ψ_3, which should be intrinsically related to the triangle geometry. The three element side directions furnish the most obvious choice, but unfortunately this leads to a singular $\boldsymbol{G}$ matrix. As the next choice we select the directions of the three median lines as local $\bar{x}$ axes, and this choice furnishes a linearly independent mode set. (Choosing the median lines as $\bar{y}$ axes also leads to a singular $\boldsymbol{G}$ matrix.)

The three higher-order modes are then

$$\boldsymbol{u}_\mathrm{h} = \boldsymbol{N}_\mathrm{h}\boldsymbol{q}_\mathrm{h} = \boldsymbol{N}_{\mathrm{h}1}q_7 + \boldsymbol{N}_{\mathrm{h}2}q_8 + \boldsymbol{N}_{\mathrm{h}3}q_9 \,. \tag{3.16}$$

According to (3.15) a typical mode may be written as

$$\boldsymbol{N}_{\mathrm{h}i} = \begin{Bmatrix} u_{\mathrm{h}i} \\ v_{\mathrm{h}i} \end{Bmatrix} = \begin{bmatrix} a_{1i} & a_{2i} & a_{3i} \\ b_{1i} & b_{2i} & b_{3i} \end{bmatrix} \begin{Bmatrix} \xi^2 \\ \xi\eta \\ \eta^2 \end{Bmatrix}, \quad i = 1, 2, 3, \tag{3.17}$$

where

$$\begin{aligned} &a_{1i} = -\tfrac{1}{2}s_i c_i^2, \qquad && a_{2i} = c_i^3, \qquad && a_{3i} = \tfrac{1}{2}s_i^3 + s_i c_i^2 \,, \\ &b_{1i} = -s_i^2 c_i - \tfrac{1}{2}c_i^3, \qquad && b_{2i} = -s_i^3, \qquad && b_{3i} = \tfrac{1}{2}s_i^2 c_i \,, \end{aligned} \tag{3.18}$$

$$s_i = \sin\psi_i = (y_\mathrm{m} - x_i)/l_{\mathrm{m}i}, \qquad c_i = \cos\psi_i = (x_\mathrm{m} - x_i)/l_{\mathrm{m}i} \,. \tag{3.19}$$

Here x_m, y_m and $l_{\mathrm{m}i}$ are defined by (cf. Fig. 6)

$$x_\mathrm{m} = \tfrac{1}{2}(x_j + x_k), \qquad y_\mathrm{m} = \tfrac{1}{2}(y_j + y_k), \qquad l_{\mathrm{m}i} = \sqrt{(x_\mathrm{m} - x_i)^2 + (y_\mathrm{m} - y_i)^2} \tag{3.20}$$

in which i, j, k denote the positive cyclic permutations of 1, 2, 3; for example $i = 2$, $j = 3$, $k = 1$.

3.7. *Displacement derivatives*

The gradients of the rc-modes, which are constant over the triangle, are easily derived from (3.7):

$$\begin{Bmatrix} u_{,x} \\ u_{,y} \\ v_{,x} \\ v_{,y} \\ \gamma_{xy} \\ \theta_z \end{Bmatrix} = \lambda \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{q}_\mathrm{rc} \,. \tag{3.21}$$

On the other hand, the gradients of the h-modes vary linearly over the triangle and may be expressed as

$$\begin{Bmatrix} u_{,x} \\ u_{,y} \\ v_{,x} \\ v_{,y} \\ \gamma_{xy} \\ \theta_z \end{Bmatrix} = \lambda \begin{bmatrix} 2a_{11}\xi + a_{21}\eta & 2a_{12}\xi + a_{22}\eta & 2a_{13}\xi + a_{23}\eta \\ a_{21}\xi + 2a_{31}\eta & a_{22}\xi + 2a_{32}\eta & a_{23}\xi + 2a_{33}\eta \\ 2b_{11}\xi + b_{21}\eta & 2b_{12}\xi + b_{22}\eta & 2b_{13}\xi + b_{23}\eta \\ b_{21}\xi + 2b_{31}\eta & b_{22}\xi + 2b_{32}\eta & b_{23}\xi + 2b_{33}\eta \\ -4b_{31}\xi - 4a_{11}\eta & -4b_{32}\xi - 4a_{12}\eta & -4b_{33}\xi - 4a_{13}\eta \\ -c_1\xi - s_1\eta & -c_2\xi - s_2\eta & -c_3\xi - s_3\eta \end{bmatrix} \boldsymbol{q}_\mathrm{h} \,, \tag{3.22}$$

where the coefficients a and b are defined by (3.18). Note that even though the higher-order modes are of pure bending type in the local systems, they generally produce nonzero shear strains in the global system.

3.8. Transformation to nodal degrees of freedom

The nodal degrees of freedom of the membrane element are

$$v^{\mathrm{t}} = [u_1, v_1, \theta_1, u_2, v_2, \theta_2, u_3, v_3, \theta_3]\,, \tag{3.23}$$

where we use θ as abbreviation for θ_z. As shown in Section 2, the relationship between the nodal freedoms (3.23) and the generalized degrees of freedom can be expressed as the matrix transformation (2.13), where G is a 9×9 matrix. The 9×6 submatrix G_{rc} is associated with the rc-modes. If these are written in terms of (3.7), substitution of nodal coordinates ξ_i and η_i yields

$$G_{\mathrm{rc}} = \begin{bmatrix} 1 & 0 & -\eta_1 & \xi_1 & 0 & \eta_1 \\ 0 & 1 & \xi_1 & 0 & \eta_1 & \xi_1 \\ 0 & 0 & \lambda & 0 & 0 & 0 \\ 1 & 0 & -\eta_2 & \xi_2 & 0 & \eta_2 \\ 0 & 1 & \xi_2 & 0 & \eta_2 & \xi_2 \\ 0 & 0 & \lambda & 0 & 0 & 0 \\ 1 & 0 & -\eta_3 & \xi_3 & 0 & \eta_3 \\ 0 & 1 & \xi_3 & 0 & \eta_3 & \xi_3 \\ 0 & 0 & \lambda & 0 & 0 & 0 \end{bmatrix}. \tag{3.24}$$

Let the 9×3 matrix G_{h} be partitioned as follows:

$$G_{\mathrm{h}} = \begin{bmatrix} G_{\mathrm{h}11} & G_{\mathrm{h}12} & G_{\mathrm{h}13} \\ G_{\mathrm{h}21} & G_{\mathrm{h}22} & G_{\mathrm{h}23} \\ G_{\mathrm{h}31} & G_{\mathrm{h}32} & G_{\mathrm{h}33} \end{bmatrix}. \tag{3.25}$$

Then a typical 3×1 submatrix can be written as

$$G_{\mathrm{h}ij} = \begin{bmatrix} a_{1j}\xi_i^2 + a_{2j}\xi_i\eta_i + a_{3j}\eta_i^2 \\ b_{1j}\xi_i^2 + b_{2j}\xi_i\eta_i + b_{3j}\eta_i^2 \\ -\lambda(c_j\xi_i + s_j\eta_i) \end{bmatrix}. \tag{3.26}$$

The free formulation requires that all modes be linearly independent in terms of the nodal degrees of freedom (3.23). Therefore the inverse relation to (2.13) must exist. The inverse matrix $H = G^{-1}$ may conveniently be partitioned into submatrices as shown in (2.14). Only the submatrix H_{h} appears in the derivation of the material stiffness discussed in the following section. This submatrix is evaluated numerically.

4. The stiffness matrix

In accordance with the free formulation described in Section 2, the element stiffness matrix is formed by the superposition of basic and higher-order stiffness:

$$\boldsymbol{k} = \boldsymbol{k}_\mathrm{b} + \boldsymbol{k}_\mathrm{h} \,. \tag{4.1}$$

The basic stiffness is given by (2.11), which specialized to the two-dimensional case reads

$$\boldsymbol{k}_\mathrm{b} = \frac{1}{A} \boldsymbol{L}\boldsymbol{D}_\mathrm{m}\boldsymbol{L}^\mathrm{t} \,, \tag{4.2}$$

where $\boldsymbol{D}_\mathrm{m}$ is a constitutive matrix that relates membrane forces per unit-length to membrane strains. (It is essentially $\boldsymbol{C}$ of (2.11) integrated through the plate thickness; this integration process also reduces V to A, the area of the element.) The higher-order stiffness is given by

$$\boldsymbol{k}_\mathrm{h} = \boldsymbol{H}_\mathrm{h}^\mathrm{t} \boldsymbol{k}_{q\mathrm{h}} \boldsymbol{H}_\mathrm{h} \,, \tag{4.3}$$

where $\boldsymbol{H}_\mathrm{h}$ is defined by (2.14) and $\boldsymbol{k}_{q\mathrm{h}}$ is derived below.

4.1. Membrane equations

The membrane strains are found by appropriate differentiation of the assumed displacement modes:

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \boldsymbol{\Delta u} = \boldsymbol{\Delta N q} = \boldsymbol{B}_\mathrm{rc}\boldsymbol{q}_\mathrm{rc} + \boldsymbol{B}_\mathrm{h}\boldsymbol{q}_\mathrm{h} \,, \tag{4.4}$$

where the differentiation operator for membrane strains is

$$\boldsymbol{\Delta} = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix} . \tag{4.5}$$

From (3.21) and (3.22) one readily finds that

$$\boldsymbol{B}_\mathrm{rc} = \boldsymbol{\Delta N}_\mathrm{rc} = \lambda \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} , \tag{4.6}$$

$$\boldsymbol{B}_\mathrm{h} = \boldsymbol{\Delta N}_\mathrm{h} = \lambda \begin{bmatrix} 2a_{11}\xi + a_{21}\eta & 2a_{12}\xi + a_{22}\eta & 2a_{13}\xi + a_{23}\eta \\ b_{21}\xi + 2b_{31}\eta & b_{22}\xi + 2b_{32}\eta & b_{23}\xi + 2b_{33}\eta \\ -4b_{31}\xi - 4a_{11}\eta & -4b_{32}\xi - 4a_{12}\eta & -4b_{33}\xi - 4a_{13}\eta \end{bmatrix} . \tag{4.7}$$

The membrane forces $\boldsymbol{p}$ are related to the strains through the elastic constitutive equation

$$\boldsymbol{p} = \begin{Bmatrix} p_{xx} \\ p_{yy} \\ p_{xy} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \boldsymbol{D}_{\mathrm{m}} \boldsymbol{\varepsilon} . \tag{4.8}$$

The constitutive coefficients in (4.8) are assumed to be constant over the element. For an isotropic plate of thickness t, elastic modulus E and Poisson's ratio ν, $\boldsymbol{D}_{\mathrm{m}}$ becomes

$$\boldsymbol{D}_{\mathrm{m}} = \frac{Et}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \tfrac{1}{2}(1-\nu) \end{bmatrix} . \tag{4.9}$$

4.2. The lumping matrix

The lumping matrix defined in (2.7) for a general three-dimensional element expresses the relationship between the nodal forces $\boldsymbol{t}_{\mathrm{c}}$ and an arbitrary constant-stress field $\boldsymbol{\sigma}_{\mathrm{c}}$. Similarly, for the present case

$$\boldsymbol{t}_{\mathrm{c}} = \boldsymbol{L}\boldsymbol{p}_{\mathrm{c}} , \tag{4.10}$$

where $\boldsymbol{t}_{\mathrm{c}}$ are the nodal forces consistent with the nodal freedoms (3.23) and $\boldsymbol{p}_{\mathrm{c}}$ are constant membrane forces in accordance with (4.8). The membrane forces along a side of the element are

$$\begin{Bmatrix} \tilde{p}_{xx} \\ \tilde{p}_{xy} \end{Bmatrix} = \begin{bmatrix} \cos^2\varphi & \sin^2\varphi & 2\sin\varphi\cos\varphi \\ -\sin\varphi\cos\varphi & \sin\varphi\cos\varphi & \cos^2\varphi - \sin^2\varphi \end{bmatrix} \begin{Bmatrix} p_{xx} \\ p_{yy} \\ p_{xy} \end{Bmatrix} , \tag{4.11}$$

in which φ defines the orientation of the element side; see Fig. 7. The displacements along side i–j in the side coordinate system $(\tilde{x}, \tilde{y})$ are

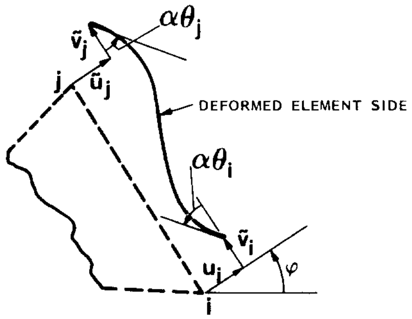

Fig. 7. Deformation of triangle side i–j assumed for the derivation of the lumping matrix

$$\begin{Bmatrix} \tilde{u} \\ \tilde{v} \end{Bmatrix} = \begin{bmatrix} \tilde{N}_{ui} & 0 & \alpha\tilde{N}_{\theta i} & \tilde{N}_{uj} & 0 & \alpha\tilde{N}_{\theta j} \\ 0 & \tilde{N}_{vi} & 0 & 0 & \tilde{N}_{vj} & 0 \end{bmatrix} \begin{Bmatrix} \tilde{u}_i \\ \tilde{v}_i \\ \theta_i \\ \tilde{u}_j \\ \tilde{v}_j \\ \theta_j \end{Bmatrix} . \tag{4.12}$$

$\tilde{N}_{ui}$, $\tilde{N}_{uj}$, $\tilde{N}_{\theta i}$ and $\tilde{N}_{\theta j}$ are cubic, beam-type interpolation functions that define the outward displacement $\tilde{u}$ in terms of the nodal displacements at i and j. Similarly, $\tilde{N}_{vi}$ and $\tilde{N}_{vj}$ are linear interpolation functions for the tangential component $\tilde{v}$.

Observe that a scaling factor α has been introduced on the shape functions that relate boundary-normal displacements to the corner rotations. As discussed in Sections 1 and 3 there exists no unique relationship between the element side rotations and the continuum-mechanics rotation (3.1). The factor α parametrizes the linkage between the two concepts of rotation. As further discussed in Section 5, the element performance in bending may be improved significantly by choosing a value of α greater than unity. However, the same α should be used for all elements in an assembly; otherwise the individual-element test described in Section 2 would be violated.

Note also that the assumed boundary displacements do not correspond to the internal displacements (3.7). As discussed in Section 2, this freedom of choice is a characteristic of the present formulation.

The virtual work along i–j due to virtual displacements at node j is

$$\delta W = \int_{i-j} (\delta\tilde{u}\,\tilde{p}_{xx} + \delta\tilde{v}\,\tilde{p}_{xy})\,\mathrm{d}\tilde{y} = \int_{i-j} ((\delta\tilde{u}_j\tilde{N}_{uj} + \alpha\delta\theta_j\tilde{N}_{\theta j})\tilde{p}_{xx} + \delta\tilde{v}_j\tilde{N}_{vj}p_{xy})\,\mathrm{d}\tilde{y} . \tag{4.13}$$

This work may be transformed to global coordinates by substituting the membrane force transformation (4.11) and the displacement transformation (3.14) with φ instead of ψ:

$$\delta W = \delta u_j t_{uj} + \delta v_j t_{vj} + \delta\theta_j t_{\theta j} . \tag{4.14}$$

The force components in this equation are

$$\begin{aligned} t_{uj} &= \tfrac{1}{2} l_{ij} (p_{xx}\cos\varphi + p_{xy}\sin\varphi) = \tfrac{1}{2}(y_{ji}p_{xx} + x_{ij}p_{xy}) , \\ t_{vj} &= \tfrac{1}{2} l_{ij} (p_{xx}\sin\varphi + p_{xy}\cos\varphi) = \tfrac{1}{2}(x_{ij}p_{yy} + y_{ji}p_{xy}) , \\ t_{\theta j} &= \tfrac{1}{12}\alpha l_{ij}^2 (p_{xx}\cos^2\varphi + p_{yy}\sin^2\varphi + 2p_{xy}\sin\varphi\cos\varphi) = \tfrac{1}{12}\alpha(y_{ji}^2 p_{xx} + x_{ij}^2 p_{yy} + 2x_{ij}y_{ji}p_{xy}) , \end{aligned} \tag{4.15}$$

where $x_{ij} = x_i - x_j$, $y_{ji} = y_j - y_i$, l_{ij} is the element side length, and the following side integrals have been used:

$$\int_{i-j} \tilde{N}_{uj}\,\mathrm{d}\tilde{y} = \tfrac{1}{2} l_{ij} , \qquad \int_{i-j} \tilde{N}_{vj}\,\mathrm{d}\tilde{y} = \tfrac{1}{2} l_{ij} , \qquad \int_{i-j} \tilde{N}_{\theta j}\,\mathrm{d}\tilde{y} = \tfrac{1}{12} l_{ij}^2 . \tag{4.16}$$

The virtual-work contribution from side j–k to node j is obtained following a similar procedure, and the sum of the two contributions gives the total force at node j. The lumped forces at the two remaining nodes are obtained by simple index permutations. The complete matrix may be written

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_1 \\ \boldsymbol{L}_2 \\ \boldsymbol{L}_3 \end{bmatrix}, \tag{4.17}$$

where

$$\boldsymbol{L}_j = \frac{1}{2} \begin{bmatrix} y_{ki} & 0 & x_{ik} \\ 0 & x_{ik} & y_{ki} \\ \frac{1}{6}\alpha(y_{ji}^2 - y_{kj}^2) & \frac{1}{6}\alpha(x_{ij}^2 - x_{jk}^2) & \frac{1}{3}\alpha(x_{ij}y_{ji} - x_{jk}y_{kj}) \end{bmatrix}. \tag{4.18}$$

If $\alpha = 0$, the lumping matrix of the constant-strain triangle (CST) results. For this element all nodal forces are associated with translations only.

Once $\boldsymbol{L}$ is available, it is a simple matter to form the basic stiffness (4.2).

4.3. Higher-order stiffness

The higher-order stiffness matrix is given by (4.3). The computation of $\boldsymbol{H}_\mathrm{h}$ has been discussed in Section 3, and it only remains to consider the generalized higher-order stiffness:

$$\boldsymbol{k}_{q\mathrm{h}} = \int_A \boldsymbol{B}_\mathrm{h}^\mathrm{t} \boldsymbol{D}_\mathrm{m} \boldsymbol{B}_\mathrm{h} \, \mathrm{d}A\,. \tag{4.19}$$

Matrix $\boldsymbol{B}_\mathrm{h}$ contains only linear functions and is given by (4.7). Denote the ith column of $\boldsymbol{B}_\mathrm{h}$ by

$$\boldsymbol{B}_{\mathrm{h}i} = \xi \boldsymbol{B}_{\xi i} + \eta \boldsymbol{B}_{\eta i}\,, \tag{4.20}$$

in which $\boldsymbol{B}_\xi$ and $\boldsymbol{B}_\eta$ are the coefficient terms of $\boldsymbol{B}_\mathrm{h}$ associated with ξ and η, respectively. Then, assuming that $\boldsymbol{D}_\mathrm{m}$ is constant over the element, the (i, j)th term of $\boldsymbol{k}_{q\mathrm{h}}$ may be expressed as

$$k_{q\mathrm{h}}(i,j) = J_{\xi\xi} \boldsymbol{B}_{\xi i}^\mathrm{t} \boldsymbol{D}_m \boldsymbol{B}_{\xi j} + J_{\xi\eta}(\boldsymbol{B}_{\xi i}^\mathrm{t} \boldsymbol{D}_m \boldsymbol{B}_{\eta j} + \boldsymbol{B}_{\eta i}^\mathrm{t} \boldsymbol{D}_m \boldsymbol{B}_{\xi j}) + J_{\eta\eta} \boldsymbol{B}_{\eta i}^\mathrm{t} \boldsymbol{D}_m \boldsymbol{B}_{\eta j}\,, \tag{4.21}$$

where $J_{\xi\xi}$, $J_{\xi\eta}$ and $J_{\eta\eta}$ are momentum integrals:

$$\begin{aligned} J_{\xi\xi} &= \int_A \xi^2 \, \mathrm{d}A = -\tfrac{1}{6}A(\xi_1\xi_2 + \xi_2\xi_3 + \xi_3\xi_1)\,, \\ J_{\xi\eta} &= \int_A \xi\eta \, \mathrm{d}A = \tfrac{1}{12}A(\xi_1\eta_1 + \xi_2\eta_2 + \xi_3\eta_3)\,, \\ J_{\eta\eta} &= \int_A \eta^2 \, \mathrm{d}A = -\tfrac{1}{6}A(\eta_1\eta_2 + \eta_2\eta_3 + \eta_3\eta_1)\,. \end{aligned} \tag{4.22}$$

(Despite appearances, the right-hand sides of the first and last equations yield strictly positive values.)

4.4. *Recovery of strains and stresses*

Once the nodal displacements are determined, the element strains can easily be calculated as

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_{\mathrm{c}} + \boldsymbol{\varepsilon}_{\mathrm{h}} = \boldsymbol{B}_{\mathrm{rc}}\boldsymbol{q}_{\mathrm{rc}} + \boldsymbol{B}_{\mathrm{h}}\boldsymbol{q}_{\mathrm{h}} = (\boldsymbol{B}_{\mathrm{rc}}\boldsymbol{H}_{\mathrm{rc}} + \boldsymbol{B}_{\mathrm{h}}\boldsymbol{H}_{\mathrm{h}})\boldsymbol{v}\,, \tag{4.23}$$

in which $\boldsymbol{B}_{\mathrm{rc}}$ and $\boldsymbol{B}_{\mathrm{h}}$ are given by (4.6) and (4.7), respectively. Note that the r-part of $\boldsymbol{B}_{\mathrm{rc}}$ is null since only the constant-strain patterns contribute. The constant-strain part $\boldsymbol{\varepsilon}_{\mathrm{c}}$ may also be computed by taking advantage of the lumping-matrix relationship (2.17), which for a two-dimensional element reduces to

$$\boldsymbol{\varepsilon}_{\mathrm{c}} = \frac{1}{A}\boldsymbol{L}^{\mathrm{t}}\boldsymbol{v}\,. \tag{4.24}$$

Use of this form avoids the calculation of $\boldsymbol{H}_{\mathrm{rc}}$.

The higher-order strains $\boldsymbol{\varepsilon}_{\mathrm{h}}$, which are associated with the h-modes, vanish at the centroid and vary linearly over the element. Their corner values may be evaluated by substituting appropriate corner coordinates in $\boldsymbol{B}_{\mathrm{h}}$.

Once the strains are available, the corresponding stresses and membrane forces are easily obtained from the constitutive equation (4.8).

5. Adjusting the stiffness matrix

5.1. *The free parameters*

In Section 4 a parameter $\alpha \geqslant 0$ was introduced in the derivation of the lumping matrix $\boldsymbol{L}$. This parameter appears as a scaling factor on the boundary shape functions associated with the rotational freedoms. The value of α affects both the basic stiffness matrix and the mean (centroidal) strains calculated through (4.23).

A second parameter, $\beta > 0$, may be introduced as a scaling factor on the higher-order stiffness so that

$$\boldsymbol{k} = \boldsymbol{k}_{\mathrm{b}} + \beta\boldsymbol{k}_{\mathrm{h}}\,. \tag{5.1}$$

This may be justified as follows. The stiffness matrix produced by the free formulation passes the individual-element test described in Section 2 for *any* higher-order generalized stiffness that has the proper rank [15]. Consequently, $\boldsymbol{k}_{\mathrm{h}}$ may be scaled by a positive coefficient without destroying the fundamental convergence properties.

In the present section we discuss the use of α and β to improve the element stiffness properties. Inasmuch as α affects the basic stiffness, it has an important effect on element performance and has to be chosen with care. On the other hand, β affects only the higher-order stiffness and thus plays a less important role.

5.2. *The parametrized lumping matrix*

The role of α in the lumping matrix (4.17) may be displayed more clearly by writing

$$\boldsymbol{L} = \boldsymbol{L}_0 + \alpha \boldsymbol{L}_1 . \tag{5.2}$$

$\boldsymbol{L}_0$ is the lumping matrix of the constant-strain triangle (CST) expanded with zero-entries in rows associated with rotational freedoms. Conversely, $\boldsymbol{L}_1$ has nonzero-entries associated with the corner rotations and is uncoupled from the translational degrees of freedom. The corresponding decomposition of the basic stiffness (4.2) is

$$\boldsymbol{k}_{\mathrm{b}} = \frac{1}{A}\boldsymbol{L}_0\boldsymbol{D}_{\mathrm{m}}\boldsymbol{L}_0^{\mathrm{t}} + \frac{\alpha}{A}(\boldsymbol{L}_0\boldsymbol{D}_{\mathrm{m}}\boldsymbol{L}_1^{\mathrm{t}} + \boldsymbol{L}_1\boldsymbol{D}_{\mathrm{m}}\boldsymbol{L}_0^{\mathrm{t}}) + \frac{\alpha^2}{A}\boldsymbol{L}_1\boldsymbol{D}_{\mathrm{m}}\boldsymbol{L}_1^{\mathrm{t}} = \boldsymbol{k}_{\mathrm{b}00} + \alpha\boldsymbol{k}_{\mathrm{b}01} + \alpha^2\boldsymbol{k}_{\mathrm{b}11} . \tag{5.3}$$

If $\alpha = 0$, $\boldsymbol{k}_{\mathrm{b}}$ reduces to $\boldsymbol{k}_{\mathrm{b}00}$, which is the CST stiffness matrix augmented with zero-rows and columns. The CST element is notoriously overstiff in bending patterns. At the other extreme, if $\alpha \gg 1$, $\boldsymbol{k}_{\mathrm{b}}$ is eventually dominated by $\boldsymbol{k}_{\mathrm{b}11}$, which injects very large rotational stiffnesses. The effect of intermediate α on the basic stiffness is studied in more detail below.

5.3. *Energy minimization*

The basic stiffness strain energy corresponding to a nodal displacement vector $\boldsymbol{v}$ is

$$U_{\mathrm{b}} = \tfrac{1}{2}\boldsymbol{v}^{\mathrm{t}}\boldsymbol{k}_{\mathrm{h}}\boldsymbol{v} = U_{\mathrm{b}00} + \alpha U_{\mathrm{b}01} + \alpha^2 U_{\mathrm{b}11} . \tag{5.4}$$

If $U_{\mathrm{b}11} > 0$, the energy is minimized with respect to α for

$$\alpha_{\mathrm{b}} = -U_{\mathrm{b}01}/2U_{\mathrm{b}11} . \tag{5.5}$$

If $\boldsymbol{v}$ is a rigid-body motion, a constant-strain mode or a linear combination of these, $\boldsymbol{L}_1\boldsymbol{v} = \boldsymbol{0}$; both $U_{\mathrm{b}01}$ and $U_{\mathrm{b}11}$ vanish, and the value of α has no significance. The parameter α only becomes significant if $\boldsymbol{v}$ contains components of the higher-order modes (3.17). A general combination of the h-modes is

$$\boldsymbol{v} = \boldsymbol{G}_{\mathrm{h}}\boldsymbol{q}_{\mathrm{h}} . \tag{5.6}$$

Consider the effect of applying one higher-order mode at a time, so that

$$\boldsymbol{v}_i = \boldsymbol{G}_{\mathrm{h}i} , \tag{5.7}$$

where $\boldsymbol{G}_{\mathrm{h}i}$ denotes the ith column of $\boldsymbol{G}_{\mathrm{h}}$. This $\boldsymbol{v}$ corresponds to a pure bending mode about the median that passes through corner i. The energy associated with this mode can be expressed in terms of the constitutive matrix $\boldsymbol{D}_{\mathrm{m}}$ and of the triangle dimensions h_i, e_i and $l_{\mathrm{m}i}$

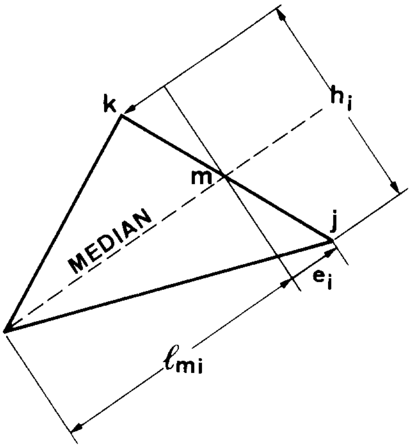

Fig. 8. Geometric quantities that enter in the definition of α_{b_i}.

defined in Fig. 8. For an isotropic material with zero Poisson's ratio, (5.5) gives

$$\alpha_{b_i} = 1 + \frac{\rho_i^2(3\kappa_i^2 - 1)^2 + 9\kappa_i^2}{9\rho_i^4\kappa_i^2(\kappa_i^2 - 1)^2 + 2\rho_i^2(3\kappa_i^2 - 1)^2 + 9\kappa_i^2}, \tag{5.8}$$

where

$$\rho_i = l_{mi}/(2h_i), \quad \kappa_i = e_i/l_{mi}. \tag{5.9}$$

The value of α_{b_i} is always in the range 1 through 2, and is not sensitive to the choice of material properties. If the median is normal to the side j–k, $\kappa_i = 0$ and $\alpha_{b_i} = 1.5$ for all ρ_i. This happens for the three medians of an equilateral triangle. For an isosceles right-angled triangle, $\alpha_{h1} = 1.50$, $\alpha_{h2} = \alpha_{h3} = 1.226$, where corner 1 is at the 90° angle. If $\rho \to \infty$ or $\rho \to 0$ and the special cases $\kappa = 0$ and $\kappa = 1$ are excluded, $\alpha \to 1$ and $\alpha \to 2$, respectively; these being of interest for elements of high aspect ratio.

REMARK 5.1. If we consider an arbitrary combination of higher-order modes as in (5.6), α_b does not necessarily remain in the range (1, 2). To illustrate, if $\boldsymbol{q}_h^t = [0, 1, 1]$ for the isosceles right-angled triangle (with corner 1 at the right angle), then $\alpha_b = 2.4$. On the other hand, if $\boldsymbol{q}_h^t = [0, 1, -1]$, which is more practically significant, $\alpha_b = 1.2$.

REMARK 5.2. There is a linear combination $\hat{\boldsymbol{q}}_h$ of the three higher-order modes that makes all rotations vanish; e.g., for the equilateral triangle $\hat{\boldsymbol{q}}_h^t = [1, 1, 1]$. For this combination both $\boldsymbol{L}_0\boldsymbol{v}$ and $\boldsymbol{L}_1\boldsymbol{v}$ vanish, and $U_b = 0$ for any α. This particular mode is *force-orthogonal* to the basic modes. If a test $\boldsymbol{q}_h$ close to $\hat{\boldsymbol{q}}_h$ is selected in (5.6) one is liable to find widely varying values of α_b since (5.5) approaches 0/0. But this has no practical significance because U_b is then small for any α.

The bending response of an element in practical applications is generally dominated by a single higher-order bending mode. Since the best α for the latter lies between 1 and 2, the average value $\alpha = 1.5$ is recommended for general use in arbitrary grids. Numerical experiments show that sensitivity is low as long as α is near 1.5. This recommendation can be backed up by an important property of the centroidal strains.

5.4. Centroidal strains

To manifest the effect of α on centroidal strains, rewrite (4.23) as

$$\boldsymbol{\varepsilon}_c = \boldsymbol{\varepsilon}_c^0 + \alpha\boldsymbol{\varepsilon}_c^1 = \frac{1}{A}(\boldsymbol{L}_0^t + \alpha\boldsymbol{L}_1^t)\boldsymbol{v}\,. \tag{5.10}$$

Any strain $\boldsymbol{\varepsilon}$ may be referred to a rotated coordinate system $(\bar{x}, \bar{y})$ through the transformation

$$\bar{\boldsymbol{\varepsilon}} = \boldsymbol{T}(\psi)\boldsymbol{\varepsilon}, \qquad \psi = \text{angle}(\bar{x}, x)\,, \tag{5.11}$$

which with $c = \cos\psi$ and $s = \sin\psi$ may be written down as

$$\begin{Bmatrix} \bar{\varepsilon}_{xx} \\ \bar{\varepsilon}_{yy} \\ \bar{\gamma}_{xy} \end{Bmatrix} = \begin{bmatrix} c^2 & s^2 & sc \\ s^2 & c^2 & -sc \\ -2sc & 2sc & c^2 - s^2 \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix}. \tag{5.12}$$

Select $\bar{x}$, $\bar{y}$ as one of the three triangle-median local systems so that $\psi = \psi_i$ $(i = 1, 2, 3)$, and apply the transformation (5.11) to the centroidal strains (5.10):

$$\bar{\boldsymbol{\varepsilon}}_{ci} = \frac{1}{A}\boldsymbol{T}(\psi_i)(\boldsymbol{L}_0^t + \alpha\boldsymbol{L}_1^t)\boldsymbol{v}\,. \tag{5.13}$$

Choose for $\boldsymbol{v}$ the higher-order nodal displacement $\boldsymbol{v}_i$ of (5.7). The exact strains associated with this mode are zero at the centroid (cf. Remark 3.2). On the other hand, $\bar{\boldsymbol{\varepsilon}}_c$ as provided by (5.7) will not generally vanish (it certainly will not for $\alpha = 0$). A symbolic calculation (with MACSYMA) shows that

component $\bar{\varepsilon}_{xx}$ vanishes if $\alpha = 2$,
component $\bar{\varepsilon}_{yy}$ vanishes if $\alpha = 1$,
component $\bar{\gamma}_{xy}$ vanishes if $\alpha = 1.5$.

This result holds for *any* triangle geometry and may be expressed in matrix form as

$$\bar{\boldsymbol{\varepsilon}}_{ci} = \boldsymbol{T}(\psi_i)\boldsymbol{\varepsilon}_{ci}^0 + \boldsymbol{S}\boldsymbol{T}(\psi_i)\boldsymbol{\varepsilon}_{ci}^1 = \boldsymbol{0} \quad \text{for } i = 1, 2, 3\,, \tag{5.14}$$

in which

$$\boldsymbol{S} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1.5 \end{bmatrix}, \qquad \boldsymbol{\varepsilon}_{ci}^0 = \frac{1}{A}\boldsymbol{L}_0^t\boldsymbol{v}_i, \qquad \boldsymbol{\varepsilon}_{ci}^1 = \frac{1}{A}\boldsymbol{L}_1^t\boldsymbol{v}_i\,. \tag{5.15}$$

Selecting $\alpha = 1.5$ decouples the parasitic shear strain $\bar{\gamma}_{xy}$ from the three higher-order modes. This property is especially important for high-aspect-ratio triangles. A similar shear-decoupling condition appears in the derivation of C^0-bending elements [21].

5.5. Scaling the higher-order stiffness

The higher-order stiffness scaling factor β defined by (5.1) may be used to adjust the total stiffness $\boldsymbol{k}$ so that the element response to certain deformation patterns is optimized. Unlike α, we could select a different β for each element, although this would be admittedly a refinement (see Remark 5.3).

The analytical derivation of 'best' β-values for arbitrary geometries poses severe difficulties because of the complexity of the matrices that make up $\boldsymbol{k}_h$. In the present paper a simple energy-balancing technique is used for improving the response of triangle assemblies of the type shown in Fig. 9 to pure bending. A more refined study was not felt to be justified at this time.

The rectangular mesh assemblies of Fig. 9 consist of linearly isotropic material of elastic modulus E, Poisson's ratio ν, and unit-thickness. They are subjected to the pure bending displacement field

$$u = xy, \qquad v = -\tfrac{1}{2}x^2, \tag{5.16}$$

where x and y are referred to the quadrilateral center. Let $\boldsymbol{v}$ be the nodal displacement vector associated with this mode. The strain energy taken up by the assembly is

$$U_b(\beta) = \tfrac{1}{2}\boldsymbol{v}^t(\boldsymbol{K}_b + \beta\boldsymbol{K}_h)\boldsymbol{v}, \tag{5.17}$$

where $\boldsymbol{K}_b$ and $\boldsymbol{K}_h$ denote the basic and higher-order stiffness matrices, respectively, of the assembly. The basic stiffness is formed with $\alpha = 1.5$ as recommended. U_b is a linearly increasing function of the coefficient β, which is assumed to be the same for all elements. The exact strain energy taken up by the continuum body is

$$U_c = \frac{E}{24(1-\nu^2)}\,ab^3\,. \tag{5.18}$$

If $U_b(0) < U_c$, there is a value

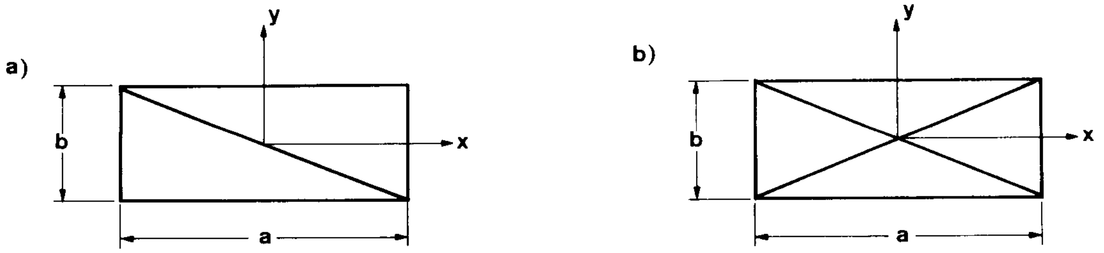

Fig. 9. Triangle assemblies considered in the optimization of β.

$$\beta_b = \frac{U_c - U_b(0)}{U_b(1) - U_b(0)} \tag{5.19}$$

for which $U_b(\beta_b) = U_c$; if $U_b(0) \geqslant U$, then $\boldsymbol{K}_b$ is overstiff and no energy balancing is possible.

The results for the two-triangle mesh unit are listed in Table 1 for three values of ν and a wide range of aspect ratios $r = a/b$. For this configuration the dependence on the aspect ratio r is surprisingly mild. The dependence of Poisson's ratio is of minor importance.

Table 1
Values of β_b for the two-triangle assembly of Fig. 9(a)

Aspect ratio	Balancing β for		
$r = a/b$	$\nu = 0.0$	$\nu = 0.25$	$\nu = 0.50$
0 01	0.376	0 376	0.376
0.10	0.387	0.383	0.381
0.25	0.437	0.419	0 404
0.50	0 529	0.479	0.439
1 00	0.521	0 475	0.437
2.00	0.440	0.422	0.405
4.00	0.396	0.390	0.386
10.00	0.379	0.378	0.377
100 00	0 376	0 376	0.376
∞	0 375	0 375	0.375

For the four-triangle 'union-jack' assembly of Fig. 9(b), a different type of aspect-ratio dependence is observed. The results for $\alpha = 1.5$ and $\nu = 0$ are shown in Table 2.

Table 2
Values of β_b for the four-triangle assembly of Fig. 9(b)

$r = a/b$	β_b
0 01	1.595
0.10	1.636
0.25	1.726
0.50	1 544
1.00	1.354
1.25	0.927
1 50	0.173

For $r > 1.541$ the basic stiffness matrix is overstiff in the mode (5.15) and there is no balancing β. Because of this high variability, the union-jack configuration as a repeating mesh unit appears less desirable than the two-triangle configuration, although the latter displays more discretization anisotropy.

REMARK 5.3. One complication left unexplored here is the fact that the higher-order stiffness may in fact embody *three* scaling factors, one for each higher-order mode, with each factor being individually adjustable according to the element geometry.

5.6. Recommendations for choosing β

The two-triangle assembly of Fig. 10(a) is quite common in modelling plane and shell structures of quadrilateral planform. The value $\beta = 0.5$ in conjunction with $\alpha = 1.5$ emerges as a good choice.

6. Numerical experiments

6.1. Cantilever beam problem

The shear-loaded cantilever beam shown in Fig. 10(a) has been selected as a test problem for plane-stress elements by several investigators [6, 10, 11]. The problem is designed to assess the performance of plane-stress elements subjected to dominant in-plane bending behavior. Most of the analyses reported here were performed with regular meshes typified by that

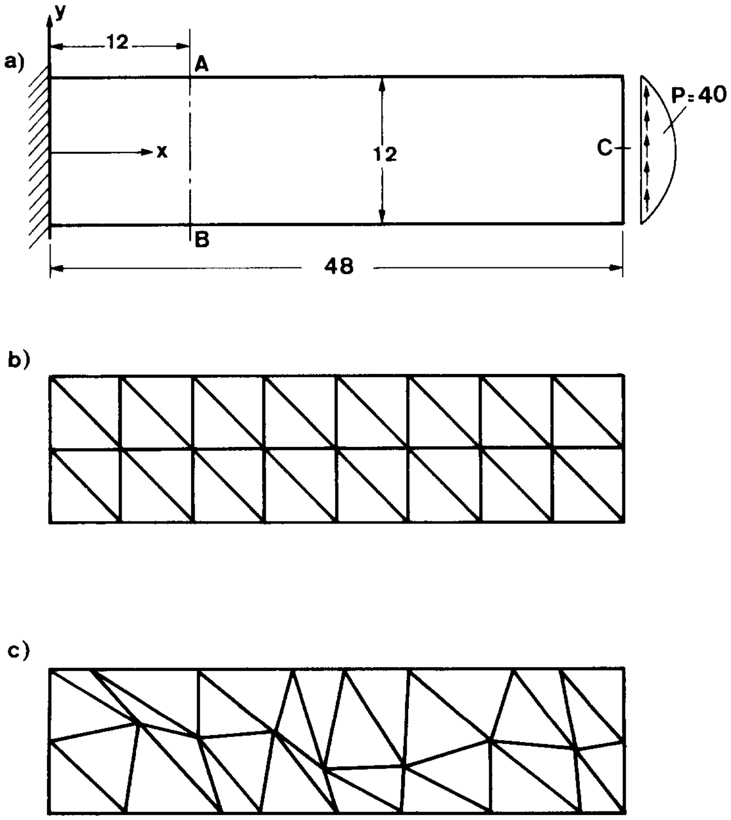

Fig. 10. (a) Cantilever beam with load P parabolically distributed along the right-hand edge (elastic modulus $E = 30000$, Poisson's ratio $\nu = 1/4$); (b) regular 8×2 mesh; (c) distorted 8×2 mesh generated for $\delta = 0.5$.

shown in Fig. 10(b). These meshes are built of two-triangle rectangular mesh units and are identified as $N_x \times N_y$, which denote the number of subdivisions in the x and y dimensions, respectively. The clamping condition was represented by constraining both displacement components to zero at nodes located on the $x = 0$ section. Nodal rotations were not constrained at the root because the term $\partial v/\partial x$ in the continuum-mechanics definition (3.1) is nonzero there. The applied shear load varies parabolically over the end section.

6.2. Deflections

The tip deflection v_C is directly related to the strain energy and suffices to describe the accuracy of the energy approximation. A theoretical value derived from 2-D elasticity is

$$v_C^{th} = \frac{PL^3}{3EI} + \frac{(4+5\nu)PL}{2EH} = 0.35583\,,$$

where L and H are the beam span and height, respectively, and $I = \frac{1}{12}H^3$. The first term comes from the linearly varying bending-moment field and the second one from the shear field. The shear-term coefficient results from assuming a root-clamping condition somewhat different from that imposed on the FE model. Extrapolated fine-grid results for the latter give $v_C^{FE} \approx 0.35587$, which is taken as 5-digit converged ('exact') value.

Tables 3, 4 and 5 give computed deflections for mesh units with aspect ratios of 1:1, 2:1 and 4:1, respectively. For reporting purposes the total end-load P was scaled by $100/v_C^{FE}$ so that the theoretical deflection becomes 100.00. Calculations were performed with the free-formulation (FF) elements for various values of α and β. The particular case $\alpha = \beta = 0$ is labelled 'CST' because the FF stiffness matrix then reduces to that of the constant-strain triangle if the corner rotations are ignored. Also reported under the label 'PE' are results obtained by using the conventional potential-energy formulation in conjunction with the shape functions of the free formulation. The following conclusions may be drawn from the tables:

(1) All elements except the PE show ultimate quadratic convergence. This rate is characteristic of nonconforming elements that pass the linear-displacement patch test but not superlinear patch tests [22].

Table 3
Tip deflection of shear-loaded cantilever beam for square mesh units

	Deflection v_C (exact = 100) for subdivision				
Element	4×1	8×2	16×4	32×8	64×16
CST	25.48	55.24	82.66	94 96	98.71
PE	60.20	89.63	101.97	105.82	106.93
FF ($\alpha = 1, \beta = 1$)	58.88	79.48	93.48	98.20	99.55
FF ($\alpha = 1.5, \beta = 1$)	75.23	92.12	97.69	99.36	99 86
FF ($\alpha = 1.5, \beta = 0.5$)[a]	96.88	99.58	99.86	99.96	99.99
FF ($\alpha = 1.5, \beta = 0.475$)[b]	98.58	100.12	99.97	100.00	100.00

[a] Recommended values of α and β.

[b] Pure bending optimized value of β; see Table 1

Table 4
Tip deflection of shear-loaded cantilever beam for 2:1 mesh unıts

	Deflectıon v_C (exact = 100) for subdivision				
Element	2×1	4×2	8×4	16×8	32×16
CST	15.20	38.04	69.95	90.10	97.30
PE	77 15	99.34	102.26	102.24	102 26
FF ($\alpha = 1, \beta = 1$)	47.50	74.44	90.80	97.23	99.25
FF ($\alpha = 1.5, \beta = 1$)	70.99	88.96	96.23	98.80	99 66
FF ($\alpha = 1.5, \beta = 0.5$)[a]	90.30	95.28	98.08	99.31	99.80
FF ($\alpha = 1.5, \beta = 0.422$)[b]	94.50	96.38	98.38	99.39	99 83

[a] Recommended values of α and β.
[b] Pure bending optimized value of β; see Table 1.

Table 5
Tip deflection of shear-loaded cantilever beam for 4·1 mesh units

	Deflection v_C (exact = 100) for subdivision					
Element	1×1	2×2	4×4	8×8	16×16	32×32
CST	7.05	17.95	43.92	75.07	92.18	97.91
PE	70.38	104.99	106 92	103.32	101.52	101.04
FF ($\alpha = 1, \beta = 1$)	29.58	59.56	83.54	94.56	98.39	99 57
FF ($\alpha = 1.5, \beta = 1$)	58.79	85.04	94.89	98.18	99.38	99 83
FF ($\alpha = 1.5, \beta = 0.5$)[a]	73.56	91.09	96.83	98.79	99.56	99.88
FF ($\alpha = 1.5, \beta = 0.390$)[b]	78.22	92.58	97.30	98.94	99.61	99.90

[a] Recommended values of α and β.
[b] Pure bending optimized value of β; see Table 1.

(2) The PE-formulated elements give erratic results. The deflections are deceptively good for coarse meshes but the finer meshes converge to an overflexible answer. This behavior illustrates the effect of energy loss for nonconforming elements that fail the patch test.

(3) The FF elements are dramatically more accurate than the CST element. The accuracy is consistently improved by more of an order of magnitude for the recommended FF element ($\alpha = 1.5$, $\beta = 0.5$). Note that this element gives reasonably good deflection values even when only one element is used through the beam height. Table 6 demonstrates some typical tip-deflection error ratios between the CST element and the FF elements. It is evident that the inclusion of rotational freedoms has paid off handsomely.

(4) Adjustment of the parameters α and β improves the results significantly over the basic values $\alpha = \beta = 1$. On the other hand, fine tuning by optimization of β pays off little over the generally recommended values $\alpha = 1.5$, $\beta = 0.5$, which give good results throughout.

(5) The FF elements are less sensitive to bad aspect ratios than the CST element.

Table 6
Some typical tip-deflection error-ratios between CST and FF elements

	1:1 (32×8)	2:1 (16×8)	4:1 (8×8)
FF ($\alpha = 1, \beta = 1$)	2.8	3.6	4 6
FF ($\alpha = 1.5, \beta = 1$)	7.8	8.3	13 7
FF ($\alpha = 1.5, \beta = 0.5$)	101 0	14.4	17 7

6.3. Stresses

Table 7 gives computed stresses at nodes of the A–B section located at $x = 12$ for the 16×4 mesh. At this section the beam-theory stresses can be used for comparison purposes since the effects of root clamping are highly localized. The beam values correspond to a nonscaled force $P = 40$. The reported values are obtained by averaging the corner stresses of all elements meeting at a node.

The stresses at interior points obtained from the best (α, β) combinations are fairly accurate, and noticeably better than the CST stresses. The simple stress averaging scheme gives relatively high residual values of τ_{xy} at boundary nodes, and underestimates the normal stress σ_{xx}. A similar pattern is observed for other meshes.

Table 7
Selected nodal stresses in cantilever beam for the 16×4 mesh

Stress component	Location x	Location y	Beam theory	CST	$\alpha = 1$, $\beta = 1$	$\alpha = 1$ 5, $\beta = 0.5$
σ_{xx}	12.0	6 0	−60.00	−44.10	−55.06	−54.57
σ_{xx}	12.0	3.0	−30.00	−25 08	−28.41	−30.09
σ_{xx}	12.0	0.0	0.00	−0.41	−0.09	−0.15
σ_{xx}	12.0	−3.0	30.00	24.38	28.10	29.75
σ_{xx}	12 0	−6.0	60.00	41.88	55 83	55 58
τ_{xy}	12.0	6.0	0.00	0.47	4.43	3.04
τ_{xy}	12.0	3.0	3.75	3 20	3.26	3.55
τ_{xy}	12.0	0.0	5.00	4.31	4 99	4.91
τ_{xy}	12.0	−3.0	3.75	3.34	3.87	3.62
τ_{xy}	12.0	−6.0	0.00	3.91	−1.32	0 15

6.4. Distorted meshes

To assess the sensitivity of the new elements to mesh distortion, the regular meshes used in the previous analyses were systematically 'perturbed' as follows. A 'nodal imperfection' parameter δ is read as data. As the regular grids are generated, the (x, y) nodal coordinates are displaced by the amounts

$$s\delta\Delta x, \qquad s\delta\Delta y,$$

where Δx, Δy are the x, y dimensions of a regular mesh unit, and the scalar $-1 \leqslant s \leqslant 1$ is produced by a random number generator. At the $y = \pm 6$ boundary points this displacement is only applied tangentially. At δ-values over 0.4–0.5 'negative area triangles' begin to appear in which case calculations are aborted. A 'perturbed' 8×2 mesh for $\delta = 0.5$ is shown in Fig. 10(c).

Deflection results for various irregular meshes and element types are given in Table 8. The reported 'deterioration' is with respect to the regular-mesh result. It can be seen that the FF elements are fairly insensitive to element layout; much less so than the fully conforming CST.

Table 8
Solution deterioration induced by mesh distortion

Mesh	Element	δ	v_C	% Deterioration
8×2	CST	0.000	55.23	0.00
		0.125	54.34	1.61
		0.250	49.90	9.95
		0.500	47.64	13.74
8×2	FF ($\alpha = 1, \beta = 1$)	0.000	79.48	0.00
		0.125	78.98	0.63
		0 250	77.88	2.05
		0.500	74.76	6.31
8×2	FF ($\alpha = 1.5, \beta = 0.5$)	0.000	99.58	0.00
		0.125	99.37	0.21
		0.250	98.40	1.20
		0.500	96.12	3.48
8×8	FF ($\alpha = 1, \beta = 1$)	0.000	94.56	0.00
		0.125	94.41	0.15
		0.250	93.92	0.66
		0.500	92.01	2 70
16×8	FF ($\alpha = 1.5, \beta = 1$)	0.000	98.80	0.00
		0.125	98.67	0.13
		0.250	98.21	0.60
		0.400	97.41	1 41

6.5. Cook's problem

Tables 9–11 gives results computed for the plane-stress problem defined in Fig. 11. This problem was proposed in [23] as a test case for general quadrilateral elements. There is no known analytical solution but the results for the 32×32 mesh may be used for comparison purposes. Results for the HL, HG and Q4 elements are taken from [23] whereas the results for Q6 and QM6 are taken from [24]. Further data on other elements are provided in [25]. The accuracy of the new triangle for the recommended α and β is generally comparable to that of the hybrid (HL, HG) and nonconforming (Q6, QM6) quadrilateral elements, and somewhat better than that of the conforming bilinear quadrilateral Q4.

Table 9
Results for the plane-stress problem of Fig. 11

	Vertical deflection at C for subdivision				
Element	2×2	4×4	8×8	16×16	32×32
HL	18.17	22.03		23.81	
HG	22.32	23.23		23.91	
Q4	11.85	18.30		23.43	
Q6	22.94	23 48			
QM6	21 05	23 02			
CST	11.92	18.28	22.02	23 41	
FF ($\alpha = 1.5, \beta = 0.5$)	20.36	22.42	23.41	23.79	23.91

Table 10
Results for the plane-stress problem of Fig. 11

	Minimum stress at B for subdivision				
Element	2×2	4×4	8×8	16×16	32×32
HL	−0.1335	−0.1700		−0 2005	
HG	−0 1394	−0.1566		−0.1952	
Q4	−0.0916	−0.1510		−0.2002	
Q6	−0.1936	−0.1936			
QM6	−0.1565	−0.1853			
CST	−0.0360	−0.1002	−0.1567	−0.1844	
FF ($\alpha = 1.5, \beta = 0.5$)	−0.1804	−0 1706	−0.1902	−0.1981	−0.2012

Table 11
Results for the plane-stress problem of Fig 11

	Maximum stress at A for subdivision				
Element	2×2	4×4	8×8	16×16	32×32
HL	0.1582	0.1980		0.2294	
HG	0.0933	0.1628		0.2225	
Q4	0.1281	0.1905		0 2358	
Q6	0.1866	0.2210			
QM6	0.1900	0.2239			
CST	0.0760	0.1498	0.1999	0.2217	
FF ($\alpha = 1.5, \beta = 0.5$)	0.1700	0.2129	0.2309	0.2333	0.2359

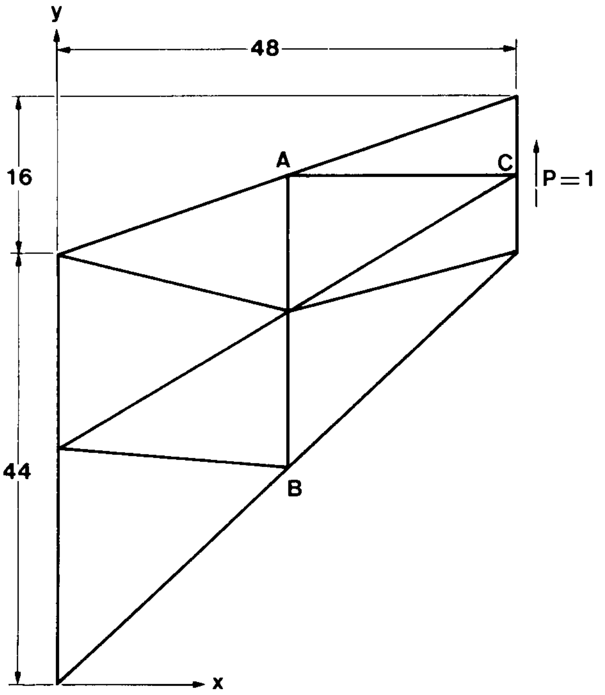

Fig. 11. Plane-stress structure (after [24]) with unit-load uniformly distributed along right-hand edge (elastic modulus $E = 1$, Poisson's ratio $\nu = 1/3$); 2×2 mesh shown.

7. Conclusions

Previous attempts at including rotational degrees of freedom in membrane finite elements have been only partially successful. The derivation of invariant and convergent elements of arbitrary geometry has proven particularly difficult. This record has caused prominent investigators to argue against their use [9]. It is demonstrated here, however, that invariant formulations may be constructed by using the continuum-mechanics definition of rotation instead of the geometrically more intuitive concept of element side rotations. A triangular element with six translations and three rotational degrees of freedom at the corner nodes has been derived using nonconforming shape functions within the framework of the free formulation [15]. The triangle passes the individual-element test of Bergan and Hanssen on all linear displacement motions, and consequently Irons' patch test for any geometry.

It is described how the bending performance of the element may be greatly improved by adjusting the transfer of boundary stresses to corner moments and by scaling the stiffness associated with higher-order deformation patterns. Numerical experiments on a problem dominated by in-plane bending modes show that the accuracy of the new element is in most cases more than an order of magnitude better than that obtainable with the constant-strain triangle.

The new element appears particularly suitable as the membrane constituent of triangular shell elements. For this class of problems the rotational degrees of freedom merge naturally with those carried along by general-purpose finite element programs.

The present formulation can be applied without difficulty for deriving rectangular and quadrilateral membrane elements with rotational degrees of freedom. Finally, since the formulation is entirely based on displacement modes, it may be readily extended to nonlinear problems [26].

Acknowledgment

This study was sponsored by the Discrete Element Technology project of the Independent Research Program of Lockheed Missiles & Space Company, Inc. The authors thank Dr. K.C. Park for many stimulating discussions, and the reviewers for many helpful suggestions for improving the exposition.

References

[1] M.J. Turner, R.W. Clough, H.C. Martin and L.J. Topp, Stiffness and deflection analysis of complex structures, J. Aero. Sci. 23, 1956.

[2] I C. Taig and R I. Kerr, Some problems in the discrete element representation of aircraft structures, in: B.M. Fraeijs de Veubeke, ed., Matrix Methods of Structural Analysis (Pergamon Press, London, 1964).

[3] O.C. Zienkiewicz, The Finite Element Method in Engineering Science (McGraw-Hill, New York, 3rd ed., 1977).

[4] B.N. Abu-Gazaleh, Analysis of plate-type prismatic structures, Ph.D. Dissertation, Dept of Civil Engineering, University of California, Berkeley, CA, 1965.

[5] A.C. Scordelis, Analysis of continuum box girder bridges, SESM Report 67-2, Dept. of Civil Engineering, University of California, Berkeley, CA, 1967.

[6] K J Willam, Finite element analysis of cellular structures, Ph.D. Dissertation, Dept. of Civil Engineering, University of California, Berkeley, CA, 1969.

[7] B O. Almroth and F.A. Brogan, Numerical procedures for analysis of structural shells, Technical Rept. AFWAL-TR-80-3129, Lockheed Palo Alto Research Laboratory, Palo Alto, CA, 1981.

[8] B.M. Irons and A. Razzaque, Experiences with the patch test for convergence of finite elements, in: K. Aziz, ed., Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations (Academic Press, New York, 1972).

[9] B.M. Irons and S. Ahmad, Techniques of Finite Elements (Ellis Horwood, Chichester, 1980).

[10] C.A. Felippa, Refined finite element analysis of linear and nonlinear two-dimensional structures, Ph.D. Dissertation, SESM Rept. 66-26, Dept. of Civil Engineering, University of California, Berkeley, CA, 1966.

[11] A.J. Carr, Refined finite element analysis of thin shell structures including dynamic loadings, Ph.D. Dissertation, SESM Rept. 67-9, Dept. of Civil Engineering, University of California, Berkeley, CA, 1967.

[12] P.G. Bergan, Plane stress analysis using the finite element method. Triangular element with 6 parameters at each node, Division of Structural Mechanics, The Norwegian Institute of Technology, Trondheim, Norway, 1967.

[13] I. Holand and P.G. Bergan, Higher order finite element for plane stress, Discussion, Proc. ASCE, J. Engrg. Mech. Div. 94 (EM2) (1968) 698–702.

[14] J.L. Tocher and B. Hartz, Higher order finite element for plane stress, Proc. ASCE, J. Engrg. Mech. Div. 93 (EM4) (1967) 149–174.

[15] P.G. Bergan and M.K. Nygård, Finite elements with increased freedom in choosing shape functions, Internat. J Numer. Meth. Engrg. 20 (1984) 643–664.

[16] P.G. Bergan and L. Hanssen, A new approach for deriving "good" finite elements, MAFELAP II

Conference, Brunel University, 1975, in: J.R. Whiteman, ed., The Mathematics of Finite Elements and Applicatıons (Academıc Press, London, 1976) 483–498.

[17] L Hanssen, Finite elements based on a new procedure for computing stiffness coefficıents, Dr. Ing Dissertation Rept. 76-1, Division of Structural Mechanics, The Norwegian Instıtute of Technology, Trondheım, Norway, 1976.

[18] P.G. Bergan, Finıte elements based on energy orthogonal functions, Internat. J. Numer. Meths Engrg. 15 (1980) 1541–1555. Addendum 17 (1981) 154–155.

[19] T.H.H. Pıan and P. Tong, Basis of finite element methods for solıd contınua, Internat. J. Numer Meths. Engrg. 1 (1969) 3–29.

[20] E.L. Wilson, R.L. Taylor, W.P. Doherty and J. Ghaboussi, Incompatible displacement models, in: S.J. Fenves, ed., Numerical and Computer Methods in Structural Mechanıcs (Academıc Press, New York, 1973) 43–57.

[21] R.H. MacNeal, A sımple quadrilateral shell element, Comput. & Structures 8 (1978) 175–183.

[22] G. Strang and G. Fix, An Analysıs of the Finite Element Method (Prentıce-Hall, Englewood Cliffs, NJ, 1972).

[23] R.D. Cook, Improved two-dimensional finıte element, J. Structural Div. ASCE 100 (ST6) (1974) 1851–1863.

[24] R.L. Taylor, P.J. Beresford and E.L. Wilson, A non-conformıng element for stress analysıs, Internat. J. Numer. Meths. Engrg. 10 (1976) 1211–1219.

[25] R.D. Cook, Ways to improve the bendıng response of finite elements, Internat. J. Numer. Meths. Engrg. 11 (1977) 1029–1039.

[26] P.G. Bergan and M.K. Nygård, Nonlinear analysis of shells usıng free formulation finite elements, in: Proceedıngs European-U.S. Symposium on Finite Element Methods for Nonlinear Problems, Trondheim, August, 1985.

Appendix A. Element stiffness subroutines

Listings of five Fortran-77 subroutines that compute the stiffness matrix of the 9-dof membrane element are given below. These subroutines are SM4M, 2M3MB, SM3MH, LUFACT and LUSOLV. A brief usage overview suffices since the programs are self-documented.

Subroutine SM4M is a "driver" that is usually called to assemble the stiffness of a quadrilateral mesh unit made up of two triangles such as the one depicted in Fig. 9(a); it can also form a single triangle or a four-triangle assembly. SM4M calls SM3MB and SM3MH, which compute the basic and higher order stiffness, respectively, of an individual triangle. SM3MH calls LUFACT and LUSOLV to numerically evaluate the (3×9) matrix $\boldsymbol{H}_{\mathrm{h}}$ defined by (2.14). This matrix appears in array HH. These two routines may be replaced by equivalent unsymmetric linear-equation solvers. Should an explicit expression of $\boldsymbol{H}_{\mathrm{h}}$ be derived in the future (for instance, through a symbolic manipulation program such as MACSYMA or SMP), these calculations would be eliminated.

None of the subroutines initializes the stiffness array; this is supposed to be done by the caller. This technique, plus the provision for a pointer array, simplifies extension to nonlinear problems in which other stiffness contributions (e.g. the geometric stiffness) may have to be added in, or use of these subroutines in conjunction with plate-bending subroutines to form shell elements.

The version listed here works in double precision. Statements that begin with C= are used to maintain the program in a source-library system and are ignored by the Fortran compiler.

```
C=DECK SM4M
C=PURPOSE Form material membrane stiffness of quadrilateral assembly
C=AUTHOR C. A. Felippa, June 1984
C=VERSION June 1984
C=EQUIPMENT Machine independent
C=KEYWORDS finite element membrane plane stress
C=KEYWORDS quadrilateral material stiffness matrix
C=BLOCK ABSTRACT
C
C     SM4M computes the material stiffness matrix of a membrane
C     quadrilateral built by 2 or 4 triangles (also a single triangle)
C     formed by subroutines SM3MB and SM3MH.  These elements are
C     based on the "free formulation" of Bergan and Nygard.
C
C=END ABSTRACT
C=BLOCK USAGE
C     The calling sequence is
C
C       CALL  SM4M (X, Y, DM, ALPHA, BETA, IAT, LS, SM, M, STATUS)
C
C     The input arguments are
C
C       X           (4 x 1) array of x coordinates of quad nodes.
C                   (only first 3 used if IAT=0).
C       Y           (4 x 1) array of y coordinates of quad nodes.
C                   (only first 3 used if IAT=0).
C       DM          (3 x 3) constitutive matrix
C       ALPHA       Rotational lumping factor (0 = CST lumping)
C       BETA        Higher-order stiffness scaling factor
C       IAT         Identifies assembly type.
C                     0    Single triangle
C                     1    2-triangles: 1-2-3 and 3-4-1 (diagonal 1-3)
C                     2    2-triangles: 1-2-4 and 2-3-4 (diagonal 1-4)
C                     3    4 triangles: (IAT=1 + IAT=2) /2
C       LS          Array of stiffness location pointers (see output SM).
C       SM          Incoming element stiffness array.
C       M           First dimension of SM in calling program.
C
C     The outputs are
C
C       SM          Output stiffness array with bending stiffness
C                   coefficients added in.  The (i,j)-th entry of the
C                   element membrane stiffness is added to SM(K,L),
C                   where K=LS(I) and L=LS(J).  If LS = (1,2,3, ... )
C                   the freedom order is  u1,v1,theta1,u2,v2, ...
C       STATUS      Status character variable.  Blank if no error found.
C
C=END USAGE
C=BLOCK FORTRAN
      subroutine    SM4M
     $       (x, y, dm, alpha, beta, iat, ls, sm, m, status)
C
C                    T Y P E   &   D I M E N S I O N
C
      character*(*)  status
      integer        iat, m, ls(12)
      double precision   x(3), y(3), dm(3,3), alpha, beta, sm(m,m)
      double precision   xt(3), yt(3), dmt(3,3), f
      integer        i, ias, j, n, t
      integer        ntrigs(0:3), tnodes(3,4, 0.3), lst(12)
      data           ntrigs /1,2,2,4/
      data           tnodes /1,2,3, 9*0,
     $                     1,2,3, 3,4,1, 6*0,
     $                     1,2,4, 2,3,4, 6*0,
     $                     1,2,3, 3,4,1, 1,2,4, 2,3,4/
C
C                  L O G I C
C
      status =    ' '
C
      do 2200  i = 1,3
        do 2200  j = 1,3
          dmt(i,j) = dm(ı,j)
 2200     continue
C
      ias =   max(0,min(iat,3))
      f =     1.0
      if (ias  eq. 3)             f = 0 5
      do 3000  t = 1,ntrigs(ias)
        do 2500  i = 1,3
          n =    tnodes(i,t,ias)
          xt(i) =  x(n)
          yt(i) =  y(n)
          lst(2*ı-1) =   ls(3*n-2)
          lst(2*i  ) =   ls(3*n-1)
          lst( i+6) =   ls(3*n  )
 2500     continue
C
        call   SM3MB (xt, yt, dmt, alpha, f, lst, sm, m, status)
        if (beta .ne. 0.0)      then
          call   SM3MH (xt, yt, dmt, f*beta, lst, sm, m, status)
        end if
        if (status(1:1) .ne  ' ')     return
 3000   continue
      return
      end
C=END FORTRAN
C=DECK SM3MB
C=PURPOSE Form basic membrane stıffness of 9-dof triangle
C=AUTHOR C. A. Felippa, June 1984
C=VERSION June 1984
C=EQUIPMENT Machine independent
C=KEYWORDS finite element membrane plane stress
C=KEYWORDS basic triangle material stiffness matrix
C=BLOCK ABSTRACT
C
C     SM3MB forms the materıal element stiffness matrix associated with
C     the basic displacement modes (rigid body & constant strain) of
C     a 9-dof plane-stress triangle based on the free formulation.
C
C=END ABSTRACT
C=BLOCK USAGE
C
C     The calling sequence is
C
C       CALL      SM3MB (X, Y, DM, ALPHA, F, LS, SM, M, STATUS)
C
C     The input arguments are
C
C       X         (3 x 1) array of x coordinates of triangle nodes
C       Y         (3 x 1) array of y coordinates of triangle nodes.
C       DM        (3 x 3) matrix relating in-plane forces to strains.
C       ALPHA     Rotational lumping factor (0 = CST lumping).
C       F         Factor by which stiffness entries wıll be multiplied
```

```
C       LS        (9 x 1) array of stiffness location pointers.
C       SM        Incoming element stiffness array.
C       M         First dimension of SM in calling program.
C
C     The outputs are
C
C       SM        Output stiffness array with basic stiffness
C                 coefficients added in  The (i,j)-th entry of the
C                 triangle stiffness is added to SM(K,L), where K=LS(I)
C                 and L=LS(J)  If LS = (1,2,3, ... ) the freedom order
C                 will be   u1,v1,u2,v2,u3,v3,theta1,theta2,theta3
C       STATUS    Status character variable.  Blank if no error found.
C
C=END USAGE
C=BLOCK TEST
C
C     For the inputs X=(1,3,2), Y=(1,1,2), DM=(8,2,0, 2,8,0, 0,0,3)
C     (isotropic material with E=7.5 and NU=.25), ALPHA=1.5, F=1, and
C     LS=(1,2,4,5,7,8,3,6,9), the computed basic stiffness should be
C
C      2 750  1 250  0 250 -1.250 -0 250  0.500 -1 500 -1.000 -0.750
C      1 250  2 750  1.750  0 250  1.250 -1 000 -1 500 -4.000 -0.750
C      0 250  1.750  1 250  0.500  1 000 -0 875 -0 750 -2 750 -0.375
C     -1 250  0 250  0.500  2.750 -1.250  0.250 -1 500  1.000 -0 750
C     -0.250  1.250  1 000 -1.250  2 750 -1 750  1.500 -4 000  0.750
C      0 500 -1.000 -0 875  0.250 -1 750  1 250 -0.750  2 750 -0.375
C     -1 500 -1.500 -0 750 -1.500  1.500 -0 750  3.000  0 000  1.500
C     -1.000 -4 000 -2 750  1 000 -4.000  2 750  0.000  8.000  0.000
C     -0.750 -0 750 -0 375 -0 750  0.750 -0.375  1.500  0.000  0.750
C
C    in which the freedom order is  u1,v1,theta1,u2, ... theta3
C=END TEST
C=BLOCK FORTRAN
      subroutine    SM3MB
     $              (x, y, dm, alpha, f, ls, sm, m, status)
C
C                   T Y P E   &   D I M E N S I O N
C
      character*(*)  status
      integer        m, ls(9)
      double precision   x(3), y(3), dm(3,3), alpha, f, p(9,3), sm(m,m)
      double precision   area2, c
      double precision   d11, d12, d13, d22, d23, d33
      double precision   x21, x32, x13, y21, y32, y13
      double precision   x12, x23, x31, y12, y23, y31
      double precision   s1, s2, s3
      integer            i, j, k, l, n
C
C                   L O G I C
C
      status =    ' '
      x21 =      x(2) - x(1)
      x12 =     -x21
      x32 =      x(3) - x(2)
      x23 =     -x32
      x13 =      x(1) - x(3)
      x31 =     -x13
      y21 =      y(2) - y(1)
      y12 =     -y21
      y32 =      y(3) - y(2)
      y23 =     -y32
      y13 =      y(1) - y(3)
      y31 =     -y13
      area2 =    y21*x13 - x21*y13
      if (area2 .le. 0.0)      then
        status = 'NEGA_AREA'
        if (area2  eq. 0.0)   status = 'ZERO_AREA'
        return
      end if
      p(1,1) =   y23
      p(2,1) =   0 0
      p(3,1) =   y31
      p(4,1) =   0.0
      p(5,1) =   y12
      p(6,1) =   0 0
      p(1,2) =   0.0
      p(2,2) =   x32
      p(3,2) =   0 0
      p(4,2) =   x13
      p(5,2) =   0 0
      p(6,2) =   x21
      p(1,3) =   x32
      p(2,3) =   y23
      p(3,3) =   x13
      p(4,3) =   y31
      p(5,3) =   x21
      p(6,3) =   y12
      n =        6
      if (alpha .ne. 0 0)        then
        p(7,1) =  y23*(y13-y21)*alpha/6.
        p(7,2) =  x32*(x31-x12)*alpha/6
        p(7,3) =  (x31*y13-x12*y21)*alpha/3.
        p(8,1) =  y31*(y21-y32)*alpha/6.
        p(8,2) =  x13*(x12-x23)*alpha/6.
        p(8,3) =  (x12*y21-x23*y32)*alpha/3.
        p(9,1) =  y12*(y32-y13)*alpha/6.
        p(9,2) =  x21*(x23-x31)*alpha/6.
        p(9,3) =  (x23*y32-x31*y13)*alpha/3
        n =        9
      end if
      c =       0.5*f/area2
      d11 =     c * dm(1,1)
      d22 =     c * dm(2,2)
      d33 =     c * dm(3,3)
      d12 =     c * dm(1,2)
      d13 =     c * dm(1,3)
      d23 =     c * dm(2,3)
      do 3000  j = 1,n
        l =     ls(j)
        s1 =    d11*p(j,1) + d12*p(j,2) + d13*p(j,3)
        s2 =    d12*p(j,1) + d22*p(j,2) + d23*p(j,3)
        s3 =    d13*p(j,1) + d23*p(j,2) + d33*p(j,3)
        do 2500  i = 1,j
          k =     ls(i)
          sm(k,l) =  sm(k,l) + (s1*p(i,1) + s2*p(i,2) + s3*p(i,3))
          sm(l,k) =  sm(k,l)
 2500     continue
 3000   continue
      return
      end
C=END FORTRAN
C=DECK SM3MH
C=PURPOSE Form high-order materıal stiffness of 9-dof membrane triangle
C=AUTHOR C. A. Felippa, June 1984
C=VERSION June 1984
C=EQUIPMENT Machine independent
```

```
C=KEYWORDS finite element membrane plane stress
C=KEYWORDS higher order material stiffness matrix
C=BLOCK ABSTRACT
C
C     SM3MH forms the material stiffness matrix associated with 3 higher
C     order modes (pure bending about medians) of a 9-dof membrane
C     triangle based on the free formulation
C
C=END ABSTRACT
C=BLOCK USAGE
C
C     The calling sequence is
C
C       CALL      SM3MH (X, Y, DM, F, LS, SM, M, STATUS)
C
C     The input arguments are
C
C       X         (3 x 1) array of x coordinates of triangle nodes
C       Y         (3 x 1) array of y coordinates of triangle nodes
C       DM        (3 x 3) matrix relating in-plane forces to strains
C       F         Factor by which stiffness entries will be multiplied
C       LS        (9 x 1) array of stiffness location pointers.
C       SM        Incoming material stiffness array
C       M         First dimension of SM in calling program
C
C     The outputs are
C
C       SM        Output stiffness array with higher order stiffness
C                 coefficients added in.  The (i,j)-th entry of the
C                 triangle stiffness is added to SM(K,L), where K=LS(I)
C                 and L=LS(J).  If LS = (1,2,3,  .  ) the freedom order
C                 will be  u1,v1,u2,v2,u3,v3,theta1,theta2,theta3.
C       STATUS    Status character variable.  Blank if no error found.
C
C=END USAGE
C=BLOCK TEST
C
C     For the inputs X=(1,3,2), Y=(1,1,2), DM=(8,2,0, 2,8,0, 0,0,3)
C     (isotropic material with E=7.5 and NU=.25), F=1, and
C     LS=(1,2,4,5,7,8,3,6,9), the computed H.O. stiffness should be
C
C      0 667 -0.667 -0.750  0.667  0.667 -0 750 -1.333  0.000 -1.167
C     -0 667  0.667  0.750 -0 667 -0.667  0.750  1 333  0 000  1 167
C     -0.750  0.750  1.184 -0.750 -0.750  0.982  1 500  0.000  0 833
C      0.667 -0.667 -0.750  0.667  0.667 -0.750 -1 333  0 000 -1.167
C      0 667 -0.667 -0.750  0.667  0.667 -0.750 -1 333  0 000 -1.167
C     -0.750  0.750  0.982 -0.750 -0 750  1.184  1 500  0 000  0 833
C     -1.333  1 333  1.500 -1.333 -1 333  1 500  2.667  0.000  2 333
C      0 000  0 000  0.000  0 000  0.000  0 000  0.000  0.000  0 000
C     -1 167  1.167  0.833 -1.167 -1.167  0.833  2.333  0.000  3 000
C
C    in which the freedom order is  u1,v1,theta1,u2,     theta3.
C
C=END TEST
C=BLOCK FORTRAN
      subroutine    SM3MH
     $              (x, y, dm, f, ls, sm, m, status)
C
C                   T Y P E   &   D I M E N S I O N
C
      character*(*)  status
      integer        ls(9), m
      double precision     x(3), y(3), dm(3,3), f, sm(m,m)
      double precision     xc(3), yc(3), xm(3), ym(3)
      double precision     bh(3,3), gt(9,9), hh(3,9), sqh(3,3), t(9)
      double precision     qx(3,3), qy(3,3)
      double precision     area, area2
      double precision     a1j, a2j, a3j, b1j, b2j, b3j
      double precision     c, cj, sj, dl, dx, dy
      double precision     d11, d12, d13, d22, d23, d33, jxx, jxy, jyy
      double precision     s1, s2, s3, s4, s5, s6
      double precision     x0, y0, xi, yi
      integer       i, j, k, l, ising, iperm(9)
C
C                    L O G I C
C
      status =  ' '
      area2 =    (y(2)-y(1))*(x(1)-x(3)) - (x(2)-x(1))*(y(1)-y(3))
      if (area2 .le. 0.0)      then
        status = 'NEGA_AREA'
        if (area2 .eq. 0.0)   status = 'ZERO_AREA'
        return
      end if
      x0 =        (x(1)+x(2)+x(3))/3.0
      y0 =        (y(1)+y(2)+y(3))/3.0
      area =       0.5*area2
      c =          1./sqrt(area)
      xc(1) =      c * (x(1)-x0)
      xc(2) =      c * (x(2)-x0)
      xc(3) =      c * (x(3)-x0)
      yc(1) =      c * (y(1)-y0)
      yc(2) =      c * (y(2)-y0)
      yc(3) =      c * (y(3)-y0)
      xm(1) =      0.5 * (xc(2)+xc(3))
      xm(2) =      0.5 * (xc(3)+xc(1))
      xm(3) =      0.5 * (xc(1)+xc(2))
      ym(1) =      0.5 * (yc(2)+yc(3))
      ym(2) =      0.5 * (yc(3)+yc(1))
      ym(3) =      0.5 * (yc(1)+yc(2))
C
C         Form G' (G transposed) in GT and initialize HH
C
      do 1300  i = 1,9
        do 1200  j = 1,6
          gt(j,i) =  0
 1200     continue
          hh(1,i) = 0.
          hh(2,i) = 0.
          hh(3,i) = 0.
 1300   continue
C
      d11 =      f * dm(1,1)
      d22 =      f * dm(2,2)
      d33 =      f * dm(3,3)
      d12 =      f * dm(1,2)
      d13 =      f * dm(1,3)
      d23 =      f * dm(2,3)
      jxx =    -2.*(xc(1)*xc(2)+xc(2)*xc(3)+xc(3)*xc(1))/3.
      jxy =        (xc(1)*yc(1)+xc(2)*yc(2)+xc(3)*yc(3))/3.
      jyy =    -2.*(yc(1)*yc(2)+yc(2)*yc(3)+yc(3)*yc(1))/3.
      do 2500  j = 1,3
        dx =      xm(j) - xc(j)
        dy =      ym(j) - yc(j)
        dl =      sqrt (dx**2 + dy**2)
        cj =      dx/dl
        sj =      dy/dl
```

```
          a1j =    -0.5*sj*cj**2
          a2j =     0.5*cj**3
          b2j =    -0 5*sj**3
          b3j =     0.5*sj**2*cj
          a3j =   -(b2j + a1j + a1j)
          b1j =   -(b3j + b3j + a2j)
          gt(1,2*j-1) =   1.
          gt(2,2*j  ) =   1.
          gt(3,2*j-1) =  -yc(j)
          gt(3,2*j  ) =   xc(j)
          gt(3,  j+6) =   c
          gt(4,2*j-1) =   xc(j)
          gt(6,2*j-1) =   yc(j)
          gt(5,2*j  ) =   yc(j)
          gt(6,2*j  ) =   xc(j)
          hh(j,j+6) =     1
          qx(j,1) =      a1j
          qx(j,2) =      b2j
          qx(j,3) =     -2 *b3j
          qy(j,1) =      a2j
          qy(j,2) =      b3j
          qy(j,3) =     -2 *a1j
          s1 =      d11*qx(j,1) + d12*qx(j,2) + d13*qx(j,3)
          s2 =      d12*qx(j,1) + d22*qx(j,2) + d23*qx(j,3)
          s3 =      d13*qx(j,1) + d23*qx(j,2) + d33*qx(j,3)
          s4 =      d11*qy(j,1) + d12*qy(j,2) + d13*qy(j,3)
          s5 =      d12*qy(j,1) + d22*qy(j,2) + d23*qy(j,3)
          s6 =      d13*qy(j,1) + d23*qy(j,2) + d33*qy(j,3)
          do 2200  i = 1,3
            xi =     xc(i)
            yi =     yc(i)
            gt(j+6,2*i-1) =    a1j*xi*xi + 2 *a2j*xi*yi + a3j*yi*yi
            gt(j+6,2*i)   =    b1j*xi*xi + 2 *b2j*xi*yi + b3j*yi*yi
            gt(j+6,i+6) =     -c*(cj*xi+sj*yi)
 2200       continue
          do 2400  i = 1,j
            sqh(i,j) = jxx * (qx(i,1)*s1+qx(i,2)*s2+qx(i,3)*s3)
     $               + jxy * (qx(i,1)*s4+qx(i,2)*s5+qx(i,3)*s6
     $                       +qy(i,1)*s1+qy(i,2)*s2+qy(i,3)*s3)
     $               + jyy * (qy(i,1)*s4+qy(i,2)*s5+qy(i,3)*s6)
 2400       continue
 2500     continue
C
C               Factor G' and backsolve to obtain H
C                                                        h
        call   LUFACT (gt, 9, 9, iperm, t, ising)
        if (ising .ne 0)        then
          status = 'SINGULAR_G'
          return
        end if
        call   LUSOLV (gt, 9, 9, iperm, hh, 3, -3)
C
C              Form physical stiffness and add to incoming SM
C
        do 4000  j = 1,9
          l =      ls(j)
          s1 =   sqh(1,1)*hh(1,j) + sqh(1,2)*hh(2,j) + sqh(1,3)*hh(3,j)
          s2 =   sqh(1,2)*hh(1,j) + sqh(2,2)*hh(2,j) + sqh(2,3)*hh(3,j)
          s3 =   sqh(1,3)*hh(1,j) + sqh(2,3)*hh(2,j) + sqh(3,3)*hh(3,j)
          do 3500  i = 1,j
            k =      ls(i)
            sm(k,l) =  sm(k,l) + (s1*hh(1,i) + s2*hh(2,i) + s3*hh(3,i))
            sm(l,k) =  sm(k,l)
 3500      continue
 4000    continue
      return
      end
C=END FORTRAN

C=DECK  LUFACT
C=PURPOSE LU factor a real unsymmetric matrix with partial pivoting
C=AUTHOR C. A. Felippa, April 1973
C=VERSION June 1984
C=KEYWORDS unsymmetric matrix triangularization LU decomposition
C=BLOCK ABSTRACT
C
C     This subroutine is a Fortran adaptation of the ALGOL procedure
C     'unsymdet' published in:
C
C     H.J. Bowdler, R.S  Martin, G.Peters and J.H Wilkinson,
C     'Solution of real and complex systems of linear equations',
C     Handbook for automatic computation, Vol 2, ed  by J.H. Wilkinson
c     and C  Reinsch, Springer-Verlag, 1971
C
C     The unsymmetric matrix, [A],  is stored in the array A(I,J),
C     I=1(1)N, J=1(1)N.  The triangular decomposition
C
C                    [A] = [L] [U]
C
C     where [L] is a lower triangular matrix and [U] is a unit
C     upper triangular matrix, is performed and overwritten
C     on [A], omitting the unit diagonal of [U].  The reciprocals
C     of the diagonal of [L] replace the original diagonal.  All-zero
C     rows are ignored.  The decomposition is carried out with
C     partial pivoting and row equilibration (i.e., rows are
C     implicitly scaled to unit length).  A record of any interchanges
C     made to the rows of [A] is kept in array IPERM(I),I=1(1)N,
C     such that the I-th row and the IPERM(I)th row were
C     interchanged at the I-th step.  The decomposition will fail
C     if [A], modified by the rounding errors (and excluding null
C     rows), is singular or almost singular.  Singularity is
C     indicated by ISING, which returns the apparent rank
C     deficiency of [A].
C
C=END       ABSTRACT
C=BLOCK     USAGE
C     The calling sequence is
C
C           CALL   LUFACT  (A, NA, N. IPERM, V, ISING)
C
C       A           Array containing matrix to be factored on input.
C                   On exit, A has the computed factors (cf. abstract)
C       NA          First dimension of array A in calling program.
C       N           Matrix order (N .le. NA)
C       IPERM       Integer array of length N where LUFACT records
C                   pivotal row interchanges (cf. ABSTRACT).
C       V           Scratch floating-point array of length ge N
C       ISING       Singularity indicator.  ISING = 0  marks a
C                   successful factorization.  Else the value of
C                   ISING indicates the apparent rank deficiency.
C
C=END       USAGE
C=BLOCK  FORTRAN
        subroutine      LUFACT  (a, na, nn, iperm, v, ising)
C
```

```
C                    T Y P E   A N D   D I M E N S I O N
C
      integer        na, nn, iperm(*), ising
      double precision  a(na,*), v(*), mactol, maceps, x, y
      integer        i, j, k, l, n
      data           maceps /2 5d-17/
C
C                    L O G I C
C
C               Initialization
C
      mactol =   8 *maceps
      ising =    0
      n =        iabs(nn)
      if (n .le. 0)                  return
C
C               Calculate the Euclidean norm of each row of A
C
      do 1500  i = 1,n
        y =        0.0
        do 1200  j = 1,n
          y =        y  +  a(i,j)**2
 1200     continue
        v(i) =     0.0
        if (y  ne. 0.0)                v(i) = sqrt(1 0/y)
 1500   continue
C
C               Main loop
C
      do 4000  k = 1,n
        if (v(k) .eq. 0.0)             then
          iperm(k) =  k
          go to 4000
        end if
        l =       k
        x =       0.0
        do 2000  i = k,n
          y =       0 0
          do 1600  j = 1,k-1
            y =       y +  a(i,j) * a(j,k)
 1600       continue
          a(i,k) =    a(i,k) - y
          y =      abs(v(i)*a(i,k))
          if (y .gt.  x)               then
            x =       y
            l =       i
          end if
 2000     continue
C
C               Interchange rows
C
        if (l  ne. k)                  then
          do 2100  j = 1,n
            y =       a(k,j)
            a(k,j) =  a(l,j)
            a(l,j) =  y
 2100       continue
        end if
        v(l) =     v(k)
        iperm(k) = l
C
C               Test for singularity
C
```

```
        if (x  le   mactol)           then
          ising =    n - (k-1)
          return
        end if
        x =    1.0/a(k.k)
        a(k.k) =   x
C
C              Calculate elements of strict upper trıangle
C
        do 3600  j = k+1.n
          y =        0.0
          do 3400  ı = 1.k-1
            y =        y  +  a(k.i) * a(i.j)
 3400       continue
          a(k.j) =   (a(k.j)-y) * x
 3600     contınue
 4000   contınue
C
      return
      end
C=END  FORTRAN
C=DECK  LUSOLV
C=PURPOSE Solve linear system with LUFACT-factored coefficıent matrıx
C=AUTHOR C. A  Felippa. April 1973
C=VERSION June 1984
C=BLOCK     ABSTRACT
C
C     This subroutıne complements LUFACT  It solves two forms
C     of the general unsymmetric linear algebraıc system
C
C               [A] [X] = [B]      or    ([A] [X])' = [B]'
C
C     where [A] is a real unsymmetric matrix prefactored by LUFACT.
C     [B] ıs a given right-hand side matrıx. [X] the corresponding
C     solution matrix. and a prime denotes transposıtion
C
C     Remark - aside from its obvious use as linear solver. this
C     procedure. ın conjunction with LUFACT. fınds application in
C     the computation of the matrix forms
C
C                                -1
C               (1)  [X] =  [A]   [B]
C                                      -1
C               (2)  [X]' =  [B]' [A]
C
C     where [A] is (N by N) and [B] is (N by M).  For (1). factor
C     [A] with LUFACT and substitute for M columns of [B]
C     For (2). factor [A]' wıth LUFACT. and solve for M rows of [B]
C     as rhs vectors (setting the last argument negative).
C
C=END      ABSTRACT
C=BLOCK    USAGE
C     The calling sequence is
C
C              CALL  LUSOLV  (A. NA. N. IPERM. B. NB. M)
C
C       A           Array containıng coefficient matrix prefactored
C                   by LUFACT.
C       NA          First dimension of array A in calling program
C       N           System order.
C       IPERM       Pivotal row ınterchange record array produced
C                   by LUFACT.
C       B           Array containing rıght hand sıdes on input
```

```
C                   On exit, B contains solution vectors.
C       NB          First dimension of array B
C       M           The absolute value of M, NRHS=IABS(M), is the
C                   number of right-hand sides to be processed.
C                   The sign of M indicates type of lhs/rhs storage
C                   GT 0,  rhs/sol vectors stored as columns of B
C                   LT 0,  rhs/sol vectors stored as rows of B.
C                   The last setting is useful in the computation
C                   of certain matrix forms (see remark in ABSTRACT)
C
C=END       USAGE
C=BLOCK  FORTRAN
        subroutine      LUSOLV  (a, na, nn, iperm, b, nb, m)
C
C                   T Y P E   A N D   D I M E N S I O N
C
      integer           na, nn, iperm(*), nb, m
      double precision  a(na,*), b(nb,*), t, sum
      logical       transp
      integer       i, j, k, n, nrhs
C
C                   L O G I C
C
C              Initialization
C
      nrhs =     iabs(m)
      transp =    false
      if (m  lt  0)                    transp =  true.
      n =        iabs(nn)
C
C              Interchange elements of rhs matrix B
C
      do 1800  i = 1,n
        k =        iperm(i)
        if (k  ne  i)                  then
          if ( not  transp)            then
            do 1400  j = 1,nrhs
            t =        b(i,j)
            b(i,j) =   b(k,j)
            b(k,j) =   t
 1400       continue
          else
            do 1600  j = 1,nrhs
            t =        b(j,i)
            b(j,i) =   b(j,k)
            b(j,k) =   t
 1600       continue
          end if
        end if
 1800   continue
C
C              Cycle over right hand sides
C
      do 4000  j = 1,nrhs
        if (.not  transp)              then
C
C              Solve for rhs  column storage
C
          b(1,j) =     b(1,j) * a(1,1)
          do 2400  i = 1,n-1
            sum =      0.0
            do 2300  k = 1,i
              sum =      sum  -  a(i+1,k) * b(k,j)
 2300          continue
             b(i+1,j) = (b(i+1,j) + sum) * a(ı+1,i+1)
 2400        contınue
           do 2800  ı = n-1,1,-1
             sum =     0 0
             do 2600  k = 1,n-i
               sum =     sum  -  a(i,i+k) * b(i+k,j)
 2600          continue
             b(ı,ȷ) =   b(ı,ȷ) + sum
 2800        continue
         else
C
C              Solve for rhs· row storage
C
           b(j,1) =    b(j,1) * a(1,1)
           do 3400  i = 1,n-1
             sum =     0.0
             do 3200  k = 1,i
               sum =     sum  -  a(i+1,k) * b(j,k)
 3200          continue
             b(j,i+1) = (b(j,i+1) + sum) * a(i+1,i+1)
 3400        contınue
           do 3800  i = n-1,1,-1
             sum =     0 0
             do 3600  k = 1,n-i
               sum =     sum  -  a(i,i+k) * b(j,i+k)
 3600          continue
             b(j,i) =   b(j,i) + sum
 3800        contınue
         end if
 4000    continue
C
        return
        end
C=END  FORTRAN
```