1.1 MITC4 Shell Element

Constructors:

NonlinearShellMITC4 Constructor for the Linear and Geometric Stiffness matrices and for the Internal Force vector.

Input: **path_in** (string), **xyz_in** (vector<vector<double>>), **vnor_in** (vector<vector<double>>), **thick_in** (vector<double>), **xlocal_in** (vector<vector<double>>), **ylocal_in** (vector<vector<double>>), **zetGPcoord_in** (vector<double>), **zetGPweigth_in** (vector<double>), **Stresses_in** (vector<double>), **DMatrix_in** (vector<double>)

path_in full path for the license file.

xyz_in nodal coordinates (e.g. xyz_in[2][0] is the x-coordinate of the third node of the element, xyz_in[0][1] is the y-coordinate of the first node of the element).

vnor_in nodal normal vector components (e.g. vnor_in[2][0] is the normal vector's x-component of the third node of the element, vnor_in[0][1] is the normal vector's y-component of the first node of the element).

thick_in thickness at the nodes of the element (e.g. thick_in[3] is the thickness value for the fourth node of the element).

xlocal_in x-local axis at the nodes. Together with “vnor_in” and ”ylocal_in” they form a local nodal coordinate system. The API constructs these vectors automatically if both “xlocal_in” and “ylocal_in” are passed through as empty vectors.

ylocal_in y-local axis at the nodes. Together with “vnor_in” and ”xlocal_in” they form a local nodal coordinate system. The API constructs these vectors automatically if both “xlocal_in” and “ylocal_in” are passed through as empty vectors.

zetGPcoord_in through the thickness Gauss integration coordinates defined in the natural coordinate system. A minimum of 2 integration points must be used. If this vector goes in empty then the API will consider 2 integration points along the thickness direction of the shell.

zetGPweigth_in through the thickness Gauss integration weights. A minimum of 2 integration points must be used. If this vector goes in empty then the API will consider 2 integration points along the thickness direction of the shell.

Stresses_in stresses at the Gauss integration points of the element. This vector is required as "non-empty" for the calculation of the internal force vector and the geometric (initial stress) stiffness matrix for nonlinear analysis.

DMatrix_in plane stress ($\sigma_{zz} = 0$) constitutive matrix in vector form for all integration points. Each integration point has 25 (5x5) positions in the D-Matrix vector.

NonlinearShellMITC4 Constructor for the Strains and Stresses.

Input: **path_in** (string), **xyz_in** (vector<vector<double>>), **vnor_in** (vector<vector<double>>), **thick_in** (vector<double>), **xlocal_in** (vector<vector<double>>), **ylocal_in** (vector<vector<double>>), **zetGPcoord_in** (vector<double>), **Displacements_in** (vector<double>), **DMatrix_in** (vector<double>)

path_in full path for the license file.

xyz_in nodal coordinates (e.g. xyz_in[2][0] is the x-coordinate of the third node of the element, xyz_in[0][1] is the y-coordinate of the first node of the element).

vnor_in nodal normal vector components (e.g. vnor_in[2][0] is the normal vector's x-component of the third node of the element, vnor_in[0][1] is the normal vector's y-component of the first node of the element).

thick_in thickness at the nodes of the element (e.g. thick_in[3] is the thickness value for the fourth node of the element).

xlocal_in x-local axis at the nodes. Together with “vnor_in” and ”ylocal_in” they form a local nodal coordinate system. The API constructs these vectors automatically if both “xlocal_in” and “ylocal_in” are passed through as empty vectors.

ylocal_in y-local axis at the nodes. Together with “vnor_in” and ”xlocal_in” they form a local nodal coordinate system. The API constructs these vectors automatically if both “xlocal_in” and “ylocal_in” are passed through as empty vectors.

zetGPcoord_in through the thickness Gauss integration coordinates defined in the natural coordinate system. A minimum of 2 integration points must be used. If this vector goes in empty then the API will consider 2 integration points along the thickness direction of the shell.

Displacements_in nodal displacement vector components (e.g. Displacements_in[2] is the z-displacement of node 1, Displacements_in[3] is the rotation of the ”xlocal_in” axis of node 1, Displacements_in[4] is the rotation of the ”ylocal_in” axis of node 1, and Displacements_in[5] is always 0.0. Displacements_in[12] is the x-displacement of node 3, Displacements_in[19] is the y-displacement of node 4).

DMatrix_in plane stress ($\sigma_{zz} = 0$) constitutive matrix in vector form for all integration points. Each integration point has 25 (5x5) positions in the D-Matrix vector.

Public Methods:

void **get_BMatrix** Strain-Displacement B-matrix in vector form (defined in the local coordinate system). The strain-displacement matrix is evaluated at each integration point of the element. There is a fixed number of in-plane integration points, 4 in-plane integration points, and a user-defined number of integration points along the thickness direction of the element. For instance, if the user defines 2 integration points along the thickness direction then the total number of integration points will be $2 \times 4 = 8$.

Arguments (Input): **igaus_in** (int), **zet_in** (double). **igaus_in** is the variable controlling the in-plane integration points (ranges from 0 to 3) with **igaus_in** = 0: $\xi = -1.0/\sqrt{}(3.0)$, $\eta = -1.0/\sqrt{}(3.0)$, **igaus_in** = 1: $\xi = 1.0/\sqrt{}(3.0)$, $\eta = -1.0/\sqrt{}(3.0)$, **igaus_in** = 2: $\xi = -1.0/\sqrt{}(3.0)$, $\eta = 1.0/\sqrt{}(3.0)$, and **igaus_in** = 3: $\xi = 1.0/\sqrt{}(3.0)$, $\eta = 1.0/\sqrt{}(3.0)$

Arguments (Output): ***BMat_out** (vector<double>). The strain-displacement B-matrix is calculated and stored in vectorial form with BMat_out[0:23] = B(1,1:24), BMat_out[24:47] = B(2,1:24), BMat_out[48:71] = B(3,1:24), BMat_out[72:95] = B(4,1:24) and BMat_out[96:119] = B(5,1:24). The rows of the strain-displacement matrix are consistent with the rows in the deformation tensor, i.e. row 1: ϵ_{xx}, row 2: ϵ_{yy}, row 3: $2\epsilon_{xy}$, row 4: $2\epsilon_{xz}$ and row 5: $2\epsilon_{yz}$.

void **get_BMatrix_Transposed()** Determines the transpose of the B-matrix at each integration point defined by **igaus_in** and **zet_in**.

Arguments (Input): —

Arguments (Output): ***BMatT** (vector<double>), transposed B-matrix.

void **set_LocalAxes_All()** Sets the local axes (co-rotational axes) at each in-plane integration point. The local axes are set for zet_ = 0.0

Arguments (Input): **axesgp_in** (vector<double>).
axesgp_in[igaus_*9+0:2] = x-local vector,
axesgp_in[igaus_*9+3:5] = y-local vector and
axesgp_in[igaus_*9+6:8] = z-local vector, with igaus_ representing the in-plane integration points $0 \leq$ igaus_ < 4.

Arguments (Output): —

vector<double> **get_LocalAxes_All()** Gets the local axes (co-rotational axes) for all in-plane integration points. The local axes are calculated for zet_ = 0.0

Arguments (Input): —

Arguments (Output): —

vector<double> **Calculate_StiffnessMITC4()** Calculates the linear (tangent) Stiffness matrix of the element.

Arguments (Input): —

Arguments (Output): —

vector<double> **get_StiffnessMITC4()** This is a getter for the linear (tangent) Stiffness matrix of the element. This getter should be called after the method "Calculate_StiffnessMITC4".

Arguments (Input): —

Arguments (Output): —

vector<double> **get_Stiffness_InitialStress()** Getter for the Geometric (Initial Stresses) Stiffness matrix of the element.

Arguments (Input): —

Arguments (Output): —

vector<double> **get_StrainsMITC4()** Get the Strain tensor in the local co-rotational coordinate system for all integration points of the element.

Arguments (Input): —

Arguments (Output): —

vector<double> **get_StrainsMITC4_Global()** Get the Strain tensor in the global coordinate system for all integration points of the element.

Arguments (Input): —