



HAL
open science

Hexahedral and prismatic solid-shell for nonlinear analysis of thin and medium-thick structures

Mouhamadou Dia

► **To cite this version:**

Mouhamadou Dia. Hexahedral and prismatic solid-shell for nonlinear analysis of thin and medium-thick structures. Mechanics [physics.med-ph]. Université de Lyon, 2020. English. NNT : 2020LY-SEI040 . tel-03178821

HAL Id: tel-03178821

<https://theses.hal.science/tel-03178821>

Submitted on 24 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSA

N° d'ordre NNT: 2020LYSEI040

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
l'Institut National des Sciences Appliquées de Lyon

École Doctorale 162
Mécanique, Énergétique, Génie civil, Acoustique

Spécialité de doctorat
MÉCANIQUE – GÉNIE MÉCANIQUE

Soutenue publiquement le 3 Juin 2020, par :

Mouhamadou DIA
Ingénieur

**HEXAHEDRAL AND PRISMATIC SOLID-SHELL FINITE
ELEMENTS FOR NONLINEAR ANALYSIS OF THIN AND
MEDIUM-THICK STRUCTURES.**

Devant le jury composé de :

E. ABISSET-CHAVANNE	Professeur (Arts et Métiers - ENSAM)	Présidente
F. ABED-MERAIM	Professeur (Arts et Métiers - ENSAM)	Rapporteur
H. NACEUR	Professeur (INSA Hauts-de-France)	Rapporteur
K. SAB	Professeur (ENPC)	Examinateur
B. TIE	Chargée de Recherche CNRS - HDR (CentraleSupélec)	Examinatrice
A. GRAVOUIL	Professeur (INSA de Lyon)	Directeur
N. HAMILA	Maître de conférences - HDR (INSA de Lyon)	Co-directeur
M. ABBAS	Ingénieur de recherche (EDF LAB)	Encadrant

Département FEDORA – INSA Lyon - Ecoles Doctorales – Quinquennal 2016-2020

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	CHIMIE DE LYON http://www.edchimie-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr INSA : R. GOURDON	M. Stéphane DANIELE Institut de recherches sur la catalyse et l'environnement de Lyon IRCELYON-UMR 5256 Équipe CDFA 2 Avenue Albert EINSTEIN 69 626 Villeurbanne CEDEX directeur@edchimie-lyon.fr
E.E.A.	ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE http://edeea.ec-lyon.fr Sec. : M.C. HAVGOUDOUKIAN ecole-doctorale.eea@ec-lyon.fr	M. Gérard SCORLETTI École Centrale de Lyon 36 Avenue Guy DE COLLONGUE 69 134 Écully Tél : 04.72.18.60.97 Fax 04.78.43.37.17 gerard.scorletti@ec-lyon.fr
E2M2	ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION http://e2m2.universite-lyon.fr Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 INSA : H. CHARLES secretariat.e2m2@univ-lyon1.fr	M. Philippe NORMAND UMR 5557 Lab. d'Ecologie Microbienne Université Claude Bernard Lyon 1 Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69 622 Villeurbanne CEDEX philippe.normand@univ-lyon1.fr
EDISS	INTERDISCIPLINAIRE SCIENCES-SANTÉ http://www.ediss-lyon.fr Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 INSA : M. LAGARDE secretariat.ediss@univ-lyon1.fr	Mme Emmanuelle CANET-SOULAS INSERM U1060, CarMeN lab, Univ. Lyon 1 Bâtiment IMBL 11 Avenue Jean CAPELLE INSA de Lyon 69 621 Villeurbanne Tél : 04.72.68.49.09 Fax : 04.72.68.49.16 emmanuelle.canet@univ-lyon1.fr
INFOMATHS	INFORMATIQUE ET MATHÉMATIQUES http://edinfomaths.universite-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	M. Luca ZAMBONI Bât. Braconnier 43 Boulevard du 11 novembre 1918 69 622 Villeurbanne CEDEX Tél : 04.26.23.45.52 zamboni@maths.univ-lyon1.fr
Matériaux	MATÉRIAUX DE LYON http://ed34.universite-lyon.fr Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bât. Direction ed.materiaux@insa-lyon.fr	M. Jean-Yves BUFFIÈRE INSA de Lyon MATEIS - Bât. Saint-Exupéry 7 Avenue Jean CAPELLE 69 621 Villeurbanne CEDEX Tél : 04.72.43.71.70 Fax : 04.72.43.85.28 jean-yves.buffiere@insa-lyon.fr
MEGA	MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE http://edmega.universite-lyon.fr Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bât. Direction mega@insa-lyon.fr	M. Jocelyn BONJOUR INSA de Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69 621 Villeurbanne CEDEX jocelyn.bonjour@insa-lyon.fr
ScSo	ScSo* http://ed483.univ-lyon2.fr Sec. : Véronique GUICHARD INSA : J.Y. TOUSSAINT Tél : 04.78.69.72.76 veronique.cervantes@univ-lyon2.fr	M. Christian MONTES Université Lyon 2 86 Rue Pasteur 69 365 Lyon CEDEX 07 christian.montes@univ-lyon2.fr

Éléments finis solide-coques hexaédrique et prismatique pour la modélisation non linéaire des structures minces et moyennement épaisses

Les structures à faibles ou moyennes épaisseurs sont naturellement présentes dans la plupart des installations de production d'énergie : bâtiment réacteur, tuyauteries sous pression, réservoirs métalliques ou bâches, cuve de réacteur, liners métalliques des enceintes de confinement pour ne citer que ceux-là. Un besoin actuellement exprimé par les unités d'ingénierie d'EDF est la modélisation des phénomènes de cloquage de liners métalliques des bâtiments réacteur. Un liner est une structure de type tôle métallique assurant la fonction d'étanchéité des centrales nucléaires. Sa modélisation nécessite la prise en compte d'un phénomène de contact-frottement engendrant du pincement sur la coque, de la plasticité sous l'effet de cloquage et de la non linéarité géométrique (instabilité de type flambement). Pour modéliser le comportement thermomécanique d'une structure pareille, les éléments finis de plaques et coques actuellement disponibles ne semblent pas être à la hauteur. Le premier verrou attribuable à ces éléments est l'hypothèse des contraintes planes qui empêche la prise en compte de certaines lois de comportement nativement tridimensionnelles. En deuxième lieu, du fait de leur formulation avec des degrés de liberté de rotations ces éléments n'offrent pas une facilité d'utilisation lorsqu'il s'agit de résoudre des problèmes prenant en compte les effets non-linéaires telles que les grandes transformations géométriques, le contact-frottement bi-facial, le flambement et les pressions suiveuses. Une alternative serait d'utiliser des éléments volumiques standards. Cependant le coût de calcul prohibitif de ces derniers est difficilement accessible pour de nombreuses applications industrielles. Le but de ces travaux est de proposer une solution à cette problématique. Nous avons proposé une formulation élément fini de type solide-coque enrichie en pincement et capable de reproduire les comportements des structures minces avec une précision satisfaisante. Ce nouvel élément fini fonctionne avec tout type de loi de comportement tridimensionnelle sans restriction sur les champs de contraintes. On peut également l'utiliser pour tous les types de problèmes mécaniques : linéaire et non linéaire, contact frottement, grande transformation, flambement, pression suiveuse etc. Les simulations numériques réalisées montrent des performances satisfaisantes.

MOTS CLÉS : Éléments fini solide-coque, Loi 3D, Champs de déformation enrichis, Intégration Réduite, Champ de déformation assumé, Contrainte normale enrichie

Hexahedral and prismatic solid-shell finite elements for nonlinear analysis of thin and medium-thick structures

Thin or medium-thick structures are naturally present in most power generation facilities: reactor building, pressurized pipelines, metal tanks or tarpaulins, reactor vessel, metal liners of containment chambers, to name but a few. A need currently expressed by EDF's engineering units is the modeling of the blistering phenomena of metal liners in reactor facilities. A liner is a metal sheet type structure that provides the impermeability function of nuclear power plants. Its modeling requires taking into account a contact-friction phenomenon causing pinching on the shell, plasticity under the effect of blistering and geometric nonlinearity (buckling type instability).

To model the thermo-mechanical behavior of such a structure, the finite elements of plates and shells currently available do not seem to be up to the task. The first limitation attributable to these elements is the assumption of plane stresses which prevents the consideration of some natively three-dimensional constitutive laws. Secondly, due to their formulation with rotational degrees of freedom these elements do not offer facility of use when solving problems that take into account non-linear effects such as large geometric transformations, bi-facial friction-contact, buckling and following pressures. An alternative would be to use standard volume elements. However, the prohibitive computing cost of the latter is difficult to access for many industrial applications.

The aim of this work is to propose a solution to this problem. We have proposed a solid-shell finite element formulation enriched in their pinching stress and strain and capable of reproducing accurately the behaviour of thin structures. This new finite element works with any type of three-dimensional behaviour law without restriction on stress fields. It can also be used for all types of mechanical problems: linear and nonlinear, frictional contact, large transformation, buckling, displacement-dependent pressure, etc. The numerical simulations carried out show satisfactory performances.

MOTS CLÉS : Solid-shell finite element, 3D constitutive Law, Enhanced assumed strain, Reduced Integration, Assumed natural strain, Enhanced pinching stress

Contents

Contents	i
List of Figures	v
List of Tables	ix
Introduction	1
1 Background on the modeling of shell like structures	7
1.1 Introduction	9
1.2 Nonlinear analysis of mechanical structures	12
1.2.1 Green Lagrange strain tensor	12
1.2.2 Stress tensor	12
1.2.3 Principle of Minimum Potential Energy	13
1.2.4 Linearization (Tangent Stiffness)	14
1.2.5 Finite Element Formulation	16
1.3 Conventional 2D shell finite element	17
1.3.1 Classical shell theories	18
1.3.2 Degenerated shell elements	19
1.4 The 3D standard finite element	22
1.4.1 Case study	22
1.4.2 Locking phenomena	23
1.5 The rise of solid-shell elements	30
1.5.1 Reduced integration (RI) method	31
1.5.2 Incompatible method	31
1.5.3 The Enhanced Assumed strain (EAS) method	33
1.5.4 The Assumed Natural Strain (ANS) method	38
1.5.5 Modified deformation gradient	40
1.6 Conclusion	40
2 Nine and seven nodes solid-shell elements	43
2.1 Introduction	45
2.2 Variational formulation	46
2.3 Nine nodes solid-shell element (SB9)	47

2.3.1	Kinematics	47
2.3.2	Mapping between the contravariant and the Cartesian basis	49
2.3.3	Strain Field	51
2.3.4	Assumed natural strain	53
2.3.5	Stabilization procedure	54
2.3.6	The ninth node	57
2.4	Seven nodes solid-shell element (SB7)	58
2.4.1	Local frame and objective stress rate	58
2.4.2	Strain tensor	61
2.4.3	In plane strain enhancement	62
2.4.4	Assumed transverse shear strain	67
2.4.5	Pinching Strain and seventh node	72
2.4.6	Stabilization of twist mode	72
2.5	Equivalent generalized nodal pressure forces	73
2.6	Static equilibrium	74
2.6.1	For SB9 finite element	75
2.6.2	For SB7 finite element	76
2.7	Elements Inertia Matrices	76
2.8	Conclusion	77
3	Numerical validations	79
3.1	Introduction	81
3.2	Linear elastic studies	82
3.2.1	Patch test	82
3.2.2	Clamped square plate	83
3.2.3	Circular clamped plate	84
3.2.4	Under pressure cylinder	86
3.2.5	Spherical shell with 18° hole	87
3.2.6	Pinched cylinder	89
3.3	Geometric nonlinear elastic studies	91
3.3.1	Bending of a cantilever beam	91
3.3.2	Spherical shell with 18 degree hole	93
3.3.3	Stretched cylinder with free edge	94
3.3.4	Pinched cylindrical shell mounted over rigid diaphragms	95
3.4	Plastic studies	96
3.4.1	Preprocessing of the logarithmic strain space	97
3.4.2	Constitutive model in the logarithmic strain space	98
3.4.3	Postprocessing	100
3.4.4	Simply supported plate	100
3.4.5	Cook's membrane	104
3.4.6	Stamping of a sheet by a cylindrical punch (Doc aster)	104
3.5	Modal analysis	106
3.5.1	Natural frequencies of a thick cylindrical ring (Doc aster)	106

3.5.2	Free vibration of a compression vane (Doc aster)	108
3.5.3	Free vibration of a corrugated metal sheet (Doc aster)	112
3.5.4	Free thin circular ring (Doc aster)	114
3.5.5	Vibration of a plate in transitory state (Doc aster)	117
3.6	Linear buckling analysis	117
3.6.1	Buckling of a free cylinder under external pressure (Doc aster)	119
3.7	Industrial application : study of a canopy joint	121
3.8	Conclusion	127
Conclusion and perspectives		129
Annexes		131
A Complement for SB9 formulation		133
A.1	Nine nodes solid-shell finite element SB9	133
B Difficulty related to the Discretization of SB9 and SB7 in thickness		137
B.1	The pinch	137
B.1.1	Generalized nodal pressure forces for two superposed elements	140
B.1.2	Assembly of the elementary matrices of the 2 interfacing elements	140
B.1.3	Interface continuity hypothesis of normal deformations	141
B.2	The transverse shear of the SB7	143
C Classical shell theory and associated finite elements		145
C.1	Kinematics	145
C.1.1	Displacement field	145
C.1.2	Strain field	145
C.1.3	Through-thickness integrated stress	146
C.1.4	Constitutive law	147
C.1.5	Resulting effort-strain relationship	147
C.1.6	Transverse shear correction	148
C.1.7	Principle of virtual works	149
C.1.8	Finite elements based on classical shell theory	149
D Recalls on the classic SB9γ25		157
D.1	Formulation of the SB9 γ 25	157
D.1.1	Variational formulation	157
D.1.2	Kinematics	158
D.1.3	Stabilization	166
D.1.4	Equivalent nodal pressure forces	171
D.2	Numerical validations SB9 γ 25	173
D.2.1	Study of a bending problem : SB9 γ 25 vs HEXA20	173
D.2.2	Study of a pinch problem : SB9 γ 25 vs Axis	175
D.3	Conclusion	177

E	Impacts on code_aster	179
E.1	Mesh reading command : LIRE_MAILLAGE	179
E.2	Mesh creation : CREA_MAILLAGE	181
E.3	Mesh modification : MODI_MAILLAGE	183
E.4	Model assignment : AFFE_MODELE	185
E.5	Conclusion	190
F	Matlab source code	191
F.1	SB9 Matlab Code	191
F.2	SB7 Matlab Code	220
	Bibliography	241

List of Figures

1	Example of power generation facilities.	2
2	Metal liner in the reactor containment vessel	2
1.1	Initial and final configurations of a transformation	12
1.2	plate theory kinematic	19
1.3	Ahmad degenerated shell finite element	20
1.4	Simple Bending of a plate with 3D standard finite elements	23
1.5	Hexahedral solid element	24
1.6	Flexion in continuum mechanics Vs flexion of low order element	24
1.7	Curved beam	28
1.8	Trapezoidal element	29
1.9	Full integration scheme to reduced integration scheme	31
2.1	Location of the Gauss-Lobatto integration points for the SB9 element	48
2.2	Configuration of the SB9 element	51
2.3	Positions of ANS tying points	54
2.4	Topology of the SB7 solid-shell element	59
2.5	Description of prismatic solid-shell element	63
2.6	Normal vectors and tangential vectors for each triangular side	64
2.7	Normal vectors and tangential vectors for each triangular side	66
2.8	Geometry of C^0 triangular element	68
2.9	Definition of the two covariant basis	69
2.10	projected view of the SB9 under pressure	73
3.1	Patch test geometry	82
3.2	Clamped square plate with concentrated load	83
3.3	Evolution of results with increasing distortion	84
3.4	Circular clamped plate.	85
3.5	Cylinder's mesh	87
3.6	Pinched hemisphere	88
3.7	Spherical shell with 18° hole, study of convergence.	89
3.8	The SB7 meshes used for this simulation	89
3.9	The SB7 meshes used for this simulation	90
3.10	Pinched cylinder, study of convergence.	91

List of Figures

3.11	Mesh form used for this simulation	92
3.12	Evolution of the cantilever beam's free-edge displacement	92
3.13	Pinched hemisphere with hole	93
3.14	Evolution of displacement of hemisphere	94
3.16	Evolution of displacement of stretched cylinder	95
3.15	Stretched cylinder with free edge	95
3.17	Pinched cylindrical shell mounted over rigid diaphragms	96
3.18	Evolution of displacement of pinched cylinder	96
3.19	Simply supported plate : geometric description	101
3.20	Evolution of the normal displacement of the plate center	101
3.21	Plastic variables	102
3.22	Stress Magnitude	103
3.23	Cook's membrane geometry.	104
3.24	Evolution of the cantilever beam's free-edge displacement	105
3.25	Stamping of a sheet by a cylindrical punch, tools geometry	106
3.26	Hardening law	106
3.27	Stamping of a sheet by a cylindrical punch, deformation state	107
3.28	Modes propres d'un anneau cylindrique épais	109
3.29	Geometry representation	110
3.30	Vibrations libres d'une aube de compression	111
3.31	Geometry representation	112
3.32	Free vibration of a corrugated metal sheet	114
3.33	Geometry representation	115
3.34	Free thin circular ring	116
3.35	Vibration of a plate in transitory state	117
3.36	Evolution of displacement of the Point P	118
3.37	Geometry of the cylinder	120
3.38	Buckling modes of a free cylinder under external pressure	121
3.39	Canopy Joint 3D mesh	121
3.40	Isotropic hardening diagram	122
3.41	The different modelizations	123
3.42	Displacements	124
3.43	Stress SXX	125
3.44	Stress SXY	126
3.45	Plastic deformations	127
B.1	SB9 finite element under pressure	138
B.2	2 superposed SB9 finite element under pressure	138
B.3	Shear profile	143
C.1	Allure des déformations.	146
C.2	Quadrangular and triangular plate finite elements	152
C.3	Enriched quadrangular shell elements	153

D.1	Reference geometry of the element and integration points	158
D.2	Local frame	167
D.3	Reference geometry of the element and integration points	171
D.4	Bending plate: thickness=0.1,HEXA20	174
D.5	Bending plate: thickness=0.01,HEXA20	174
E.1	Cube maillé avec 8 éléments HEXA8	182
E.2	Mesh cube with 8 elements HEXA9	182
E.3	Connectivité de l'élément	183
E.4	The two groups of meshes necessary for the reorientation of connectivity	184
E.5	Problem with the orientation of Gauss points for the SHB8	185
E.6	Gaussian point orientation corrected with Ori_SB9	185

List of Figures

List of Tables

1.1	Geometrical and material characteristics of the bent plate	22
1.2	Displacement of the eight-nodes standard element	23
1.3	Deformation modes of bilinear element	25
3.1	Center displacement and normal stress	85
3.2	Center displacement and normal stress	85
3.3	Pinching stress at Gauss Point : study 1	85
3.4	Pinching stress at Gauss Point : study 2	86
3.5	Geometric and material characteristics	86
3.6	Pinch stress on the 5 Gauss points	87
3.7	Displacements of the solicited nodes, with SB9 modelization	88
3.8	Displacements of the solicited nodes	89
3.9	Geometrical and material characteristics of the cylinder	90
3.10	Natural frequencies of a thick cylindrical ring	108
3.11	Geometrical and material characteristics of the cylinder	110
3.12	Frequencies in Hertz: comparison SB9 Vs Coque_3D	110
3.16	Mode 3	113
3.13	Geometrical and material characteristics of the sheet	113
3.14	Mode 1	113
3.15	Mode 2	113
3.17	Geometrical and material characteristics of the ring	115
3.18	Natural frequencies	115
3.19	Comparison of critical pressure load : SB9 vs reference	120
D.1	Geometrical and material characteristics of the plate	173
D.2	Normal displacement: plate of thickness 0.1	174
D.3	Normal displacement: plate of thickness 0.01	174
D.4	Geometrical and material characteristics of the cylinder	175
D.5	Observation of pinch stresses in the inner and outer walls of the cylinder	176

List of Tables

Introduction

The mission of the EDF Group (Electricité de France), as the operator of its power generation facilities (nuclear, hydroelectric, wind power, etc.), is to contribute to the supply of electricity in France and in some countries in Europe. To do this, it must maintain its generating facilities in good condition, carry out any repairs and justify to the authorities that they are in good working order both in operation and in the event of incidents. In particular, the Fukushima accident in 2011 led the French nuclear safety authority (ASN) to raise its requirements in terms of safety and security of production facilities. In addition, EDF wishes to extend the life of its nuclear fleet beyond 40 years (or even 50 years). In order to meet these new safety requirements, while controlling repair and compliance costs, EDF is relying increasingly on numerical simulation. This often proves to be the only tool available to EDF because justification by experimental means, in addition to being costly, is long and complex.

In this context, EDF R&D develops internally many numerical simulation codes according to the physical problems to be addressed. For structural mechanics, the ER-MES department is in charge of developing the finite element calculation code `code_aster`. `code_aster` is an open-source software developed since 1989 to meet and anticipate the needs of EDF's engineering departments as well as for the needs of the research carried out internally within EDF R&D for all that concerns the thermomechanics of non-linear structures. In addition, this code is developed with the quality assurance system required by the nuclear industry. For several years now, `code_aster` has been integrated into the numerical simulation platform `salome_meca` in order to have in open-source all the tools necessary for complete mechanical studies, including in particular pre- and post-treatments, and a user-friendly graphical interface for data entry.

The higher requirements in terms of operating safety of production units necessitate increasingly realistic simulations in order to justify the resistance of equipment to aging. A large majority of the equipment in the energy production facilities is composed of thin or medium-thick structures like: reactor building, pipelines, tanks, reactor vessel, metal liners, to name but a few. Numerical studies dedicated to the analysis of these thin structures involve the use of complex numerical models in order to obtain very precise results that can justify the long-term performance of the structures. A need currently expressed by the 'Direction Technique' (TD) is the modelling of blistering phenomena of the metal liners in reactor buildings. These liners provide a sealing function of nuclear power plants. The TD thus seeks to model structures of the metal sheet type with

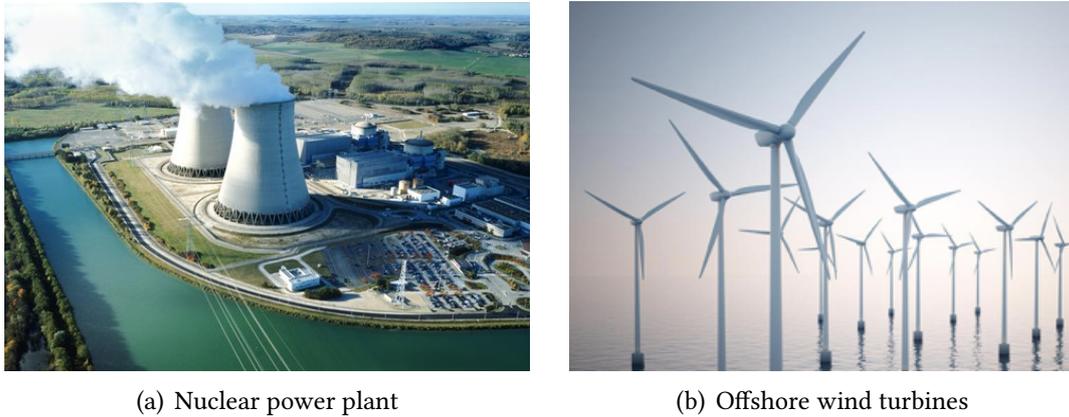


Figure 1: Example of power generation facilities.

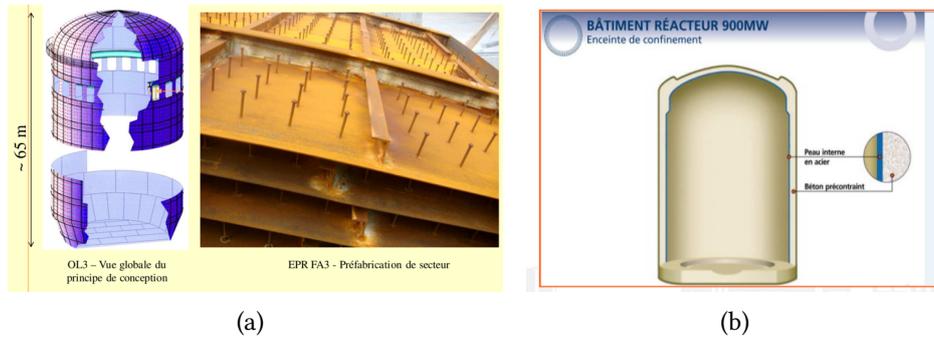


Figure 2: Metal liner in the reactor containment vessel

dimensional ratios (thickness/radius) that can be less than 0.5%, while taking into account strong non-linearity such as contact-friction causing pinching, the plasticity of sheet under the effect of blistering and the geometric non-linearity (like buckling type instability). To model the thermomechanical behavior of these thin structures, `code_aster` has several types of finite elements, DKT/DKQ (QUAD4, TRIA3, Kirchhoff-love), DST/Q4G (QUAD4, TRIA3, Mindlin-Reissner), COQUE_3D (QUAD9, Naghdi-Reissner) or SHB (HEXA8, solid-shell). The audit of the plate and shell modeling showed that these finite elements are not complete (either in their formulation or in their numerical implementation) to address EDF's current specific needs in dealing with non-linear mechanical problems on shells. The DKT, which is the most complete element currently in `code_aster`, is not usable (in `code_aster`) in large rotations and does not take into account the displacement-dependent pressures. The finite element COQUE_3D, on the other hand, does not take into account the continuous contact formulations and exhibits eccentricity problems. The SHB, which works under the hypothesis of plane stress (like DKT and COQUE_3D by the way), requires the modification of the constitutive laws by using the Deborst type algorithms which, very often, converge hardly. An alterna-

tive to shell-type finite elements would be to use classical isoparametric finite elements (3D, D_PLAN, AXIS). These elements work with purely three-dimensional behaviour laws and can handle several non-linearities such as contact/friction, fracture mechanics, elasto-plasticity or large geometrical transformations. Unfortunately they have the disadvantage of being too rigid to reproduce correctly the behaviour of thin structures. They require very fine discretization in the thickness direction (leading to discretization in other directions to avoid flattening the elements too much) to give good results. This is very expensive in terms of calculation time. Overall, advanced studies in numerical mechanics, in `code_aster`, on thin structures are limited by two obstacles. The first is the hypothesis of plane stresses usually adopted in finite elements of shells, which makes it difficult to carry out certain studies such as fracture mechanics or to take into account certain natively three-dimensional constitutive laws. The second limitation concerns non-linear effects (large geometrical transformations, elasto-plastic behaviour, contact/friction) which require reliable, robust and high-performance finite elements.

Objectifs of this thesis

EDF's need is to be able to carry out highly non-linear, robust and high-performance quasi-static calculations to demonstrate the mechanical strength of important nuclear components in critical situations. Recent work in the literature shows that the so-called `solid-shell` finite elements are very promising in addressing the issue just discussed. This is why their development is currently in full expansion. Indeed, both in the field of process simulation (sheet metal forming) and in the calculation of elongated structures (under extreme solicitations), the demand for the use of three-dimensional solid elements with behaviours close to the kinematics of shells is more and more important, in particular due to their ability to use non-linear behaviour laws with a large number of internal variables without the hypothesis of plane stress. The hypothesis of plane stress requires to completely reconsider the modeling of the nonlinear three-dimensional behaviour laws and can lead to numerical difficulties such as plastic incompressibility for example. In this context, the aim of this thesis is to develop a new class of solid-shell finite elements which allows the kinematics of Kirchhoff-Love and Reissner Mindlin, but without the hypothesis of plane stresses (in order to use three-dimensional behaviour laws). These elements must have only translational degrees of freedom, authorizing transverse shear as well as pinching stresses to be evaluated, but also contact and friction on the outer surfaces. This new class of solid-shell finite elements should also be used for large transformations. More generally, the objective of this thesis is to propose and develop a new solid-shell finite element that meets the following requirements:

- no restriction on the constitutive laws : indeed, the new finite elements will thus be a break with respect to the classical shell approaches and work with full 3D constitutive law.
- taking into account non-linear phenomena in thickness such as pinching or transverse shear.

- taking into account the large geometrical transformations involving displacement-dependent pressures, buckling etc.
- No restriction on the interface laws: contact mechanics essentially, but without restriction in the using of difficult interface laws such as de-cohesion or fracture mechanics. Due to the plane stress hypothesis, the classical shell elements do not deal with fracture mechanics.

In addition to these requirements we aim also to develop and implement solid-shell finite elements that perform accurately with :

1. No locking for incompressible materials
2. Good bending behavior
3. No locking in the limit of very thin elements
4. Distortion insensitivity
5. Good coarse mesh accuracy
6. Simple implementation of nonlinear constitutive laws
7. Automatic and robust stabilization in nonlinear situations
8. Easy mesh generation

The first aim is essential requirement to be able to model nearly incompressible materials. Such material behavior occurs in rubber like materials or in metal plasticity. The second and third aims are required for structural elements, such as plate, shell and beams elements. The fourth aim is important because in discretizing an arbitrary geometry the existence of distorted elements is inevitable. In addition, elements can get highly distorted during nonlinear simulations including finite deformations. The fifth aim results from the fact that many engineering problems have to be modeled as 3D problems. Due to computer limitations, quite coarse meshes have to be used often to solve these problems. Thus, an element which provides good coarse mesh accuracy is valuable in these situations. The sixth aim is associated with the fact that more and more nonlinear computations involving nonlinear constitutive models have to be performed to design engineering structures. Thus, an element formulation which allows a straightforward implementation of such constitutive equations is desirable. In many solid-shell formulations available in the literature, the stabilization techniques used relies on some coefficient that are often user-dependent. The seventh aim is to propose a more efficient way of stabilization, automatic and user-independent.

This thesis includes an introduction in which the objectives of the thesis are detailed, three chapters, a conclusion which sums up the work carried out, followed by a few perspectives and finally supporting documents (appendices) to provide additional information on subjects addressed but not detailed in the body of the document. The

first chapter recalls the different methods available to model the behaviour of shell-like structures, their evolution and their limits. The second chapter details the formulation of two new solid-shell finite elements. Finally, in the third chapter these two solid-shell finite elements are tested and validated on different types of mechanical problems.

Chapter 1

Background on the modeling of shell like structures

Contents

1.1	Introduction	9
1.2	Nonlinear analysis of mechanical structures	12
1.2.1	Green Lagrange strain tensor	12
1.2.2	Stress tensor	12
1.2.3	Principle of Minimum Potential Energy	13
1.2.4	Linearization (Tangent Stiffness)	14
1.2.5	Finite Element Formulation	16
1.3	Conventional 2D shell finite element	17
1.3.1	Classical shell theories	18
1.3.2	Degenerated shell elements	19
1.4	The 3D standard finite element	22
1.4.1	Case study	22
1.4.2	Locking phenomena	23
1.4.2.1	Transverse shear locking	23
1.4.2.2	Poisson thickness locking	26
1.4.2.3	Volumetric locking	27
1.4.2.4	Membrane locking	27
1.4.2.5	Trapezoidal locking	28
1.5	The rise of solid-shell elements	30

1.5.1	Reduced integration (RI) method	31
1.5.2	Incompatible method	31
1.5.3	The Enhanced Assumed strain (EAS) method	33
1.5.3.1	Three fields variational formulation	34
1.5.3.2	Linearization of discrete weak form	35
1.5.4	The Assumed Natural Strain (ANS) method	38
1.5.5	Modified deformation gradient	40
1.6	Conclusion	40

1.1 Introduction

Shell-like structures are largely present in most engineering design and process control. To model such structures, engineers generally use classical shell elements based on the degenerated shell concept or classical shell theories. These elements perform well for the simulation of bending problems ; in linear as well as for some nonlinear problems. However in certain engineering problems, like sheet metal forming or any structural problem where strong nonlinearity are in play, those elements can show some inadequacy related to the complexity of their formulations or the hypothesis they are embedded with. The main limiting hypothesis being the plane stress state, which literally means that the normal transverse stress is negligible. Under such assumption, one must modify the 3D nonlinear constitutive law to be numerically integrated. This is very often a laborious work particularly when the material law is intrinsically three dimensional. There are several approaches one can use to overcome the problem of plane stress state in shell element.

A first approach would be to use the so called ‘shell-solid’ element. These elements are based mainly upon Mindlin–Reissner shell elements with three displacements and two local rotational degrees of freedom at each node while the three-dimensional (3D) constitutive behavior is enforced via additional degrees of freedom giving the so-called 5, 6 or 7-parameters shell models, see for example, [SIM 90a, BRA 94, BET 96, BIS 97, KEM 98, BIS 00, ELA 00, CAR 02, BRA 02, CAR 05, BRA 05, SUS 13, KUL 08, KIM 08, KLI 99] among many others. The 5-parameter shell models have been enriched by a desired number of parameters at nodes to permit a representation of through-thickness stretching. Several shell elements have been developed which explicitly account for the thickness change as an additional degree of freedom leading to 6-parameter models. But due to the coupling with the Poisson ratio in bending dominated cases, the linear displacement field in thickness direction gives a constant strain which in turn causes an artificial normal stress. As a remedy, either the coupling terms have to be removed from the constitutive law by a plane stress state or the shell formulation has to be extended by a linear normal strain, leading to a 7-parameter model. This is achieved by a quadratic variation of the normal displacement field and the details can be found in References [SIM 90a, BRA 94, BET 96, BIS 97, BIS 00, ELA 00, CAR 02, CAR 05]. In a recently published paper [SAN 11], Sansalone and co-authors have proposed an approach in the same kind but a little different where an additional node is introduced in the center of three-node and four-node shell elements with only two through-thickness translational degrees of freedom of the upper and lower surfaces of the shell. Then a full 3D constitutive strain-stress behavior can be used. For triangles in bending state, either based on Kirchhoff’s or on Mindlin’s assumptions, it has been shown that the results are exactly the same as those given by the initial formulation of these elements using a plane stress hypothesis. For quadrilaterals, the results are slightly different but many numerical examples including nonlinear computations prove that those differences are not significant. While these elements work without the plane stress hypothesis, they still have a 2D topology and have rotational degrees of freedom which make their implemen-

tation in nonlinear finite element very complex. Also their 2D topology is limiting in a sense that it is hard to link them with 3D elements when a structure is composed of structural part and continuum part.

A second way to model shell like structure would be to use 3D standard isoparametric finite elements as an alternative to the plane stress hypothesis. These elements operate with purely three-dimensional behaviour laws and can handle several nonlinearities such as contact/friction, fracture mechanics, elastic-plastic or large geometrical transformations. Unfortunately they have the disadvantage of being too rigid to reproduce correctly the behaviour of thin structures. They require very fine discretization in the thickness direction (leading to a fine discretization in other directions as well, to avoid flattening the elements too much) to give good results. This is very expensive in terms of calculation time.

Faced with such an impasse, several works have been conducted over the last thirty years leading to the development of the so-called solid-shell finite elements (see [PAR 95, WRI 96a, HAU 98, HAU 00, SZE 00, VUQ 03, LEG 03, FON 05, KIM 05, REE 07, DES 07, CAR 08, SCH 09, MAS 00, ABE 02, ABE 09, KLI 06, CAR 05, REE 00, SOU 06, KLI 97, WRI 96c] to name just a few contributions). Solid-shell finite elements have the particularity of being able to model the behaviour of shell-like structures while giving an overview of the existing phenomena in the thickness. The so called solid-shell elements are very attractive and are still the subject of many researches. Like standard elements, solid-shell elements can incorporate the normal stress along thickness direction if the pinching is correctly handled. General 3D-constitutive material can be used without any assumption on the normal stress. These elements have only translational degrees of freedom (DOF) which make their formulation very simple. The difficulties associated with complex shell formulation with nodal rotation are then avoided. Moreover, since they have only translational DOF, solid-shell elements can easily be connected with standard solids elements when there is coexistence of three-dimensional and structural zones.

Now, for a solid-shell to perform accurately, it must avoid the amount of locking phenomena that standard low-order elements encounter when modeling shell structures. If not handled conveniently, solid-shell elements suffer from locking pathologies that lead the elements to give poor results, specially in case of out-of-plane bending load analysis or isochoric plasticity.

The first locking pathology that low order solid-shell element encounter is volumetric locking which arises when the material is incompressible or nearly-incompressible, as for rubber like material (see the work of Reese and Wriggers [REE 97]) or for metal plasticity (see Simo and Taylor [SIM 85], Miehe [MIE 04] to name just a few). In such material the non-vanishing volumetric strain makes the element stiffer, resulting in excessively small deformations. Several solutions have been proposed to solve volumetric locking. There are, among others, The reduced or selective integration technique proposed by Zienkiewicz [ZIE 77], the B-bar approach of Hughes [HUG 81a], applied for solid elements in the work of Belytschko [LIU 98] and the enhanced assumed strain initially proposed by Simo and co-authors [SIM 90a, SIM 92a, SIM 93]. These methods are largely applied in many solid-shell formulations. See for example the work of Massud et

al [MAS 00], Alves de Souza [SOU 03, SOU 05, SOU 06], Cardoso et al [CAR 05, CAR 08], Sze and Yao [SZE 00], Klinkel et al [KLI 97]. It's worth noting that the reduced integration or selective integration technique are not cost-free. Indeed appropriate hourglass stabilization techniques are necessary to prevent the spurious deformations modes that may arise and ensure a correct rank of the element stiffness matrix. It's well known that the spurious patterns or hourglass modes correspond to the kernel vectors of the stiffness matrix, aside from the rigid body modes. Different techniques have been proposed to deal with the rank deficiency due to selective or reduced integration technique. One can see the remarkable work of Belytschko and co-authors [BEL 83, BEL 84, BEL 93], Reese and co-author [REE 00, REE 07] as well as Combescure and co-authors [ABE 09, LEG 03].

Poisson thickness locking is due to resulting incorrect-constant distribution of the normal stress in the thickness direction. To avoid this locking phenomenon it is necessary to assure that the normal stress varies linearly along the thickness direction in bending situation. It can be done using a quadratic interpolation or the compatible normal displacement e.g. Parisch [PAR 95]. Another and most often used method is to enrich the transverse normal strain by mean of the enhanced-assumed stress method [SIM 92a, SOU 06, REE 07].

A further locking effect observed for solid-shell element is the phenomenon of a so called curvature locking or sometime trapezoidal locking. This phenomenon is found in structures where the out of plane element edges are not perpendicular to the mid-layer, which is the case for originally curved or heavily deformed structures. This locking effect can be overcome by using the naturally assumed interpolation of the normal strain as proposed by Bischoff and Ramm [BIS 97] or Betsch and Stein [BET 96].

Another commonly encountered locking pathology in solid-shell is the transverse shear locking. This happens because normal strains of linear element are coupled by shear strain. Low order elements do not have pure bending modes to behave correctly for pure bending load cases. Hence, there are parasitic shear strains that appear and become more important than normal strain. This makes the element stiffer than necessary resulting in a poor bending behavior. An interesting remedy is the Assumed Natural Strain (ANS) method first proposed by Hughes and Tezduyar [HUG 81b], followed by Wempner [WEM 82] then by Dvorkin and co-authors [DVO 84, DVO 95b, DVO 95a] for shell elements. The approach has since then been applied in solid-shell formulation by many authors. See e.g. [HAU 98, HAU 01, KLI 97, CAR 05, CAR 08, SCH 09].

The goal of this chapter is to give a short review about the different formulations used to model shell like structures (as discussed above). We will highlight their fundamentals assumptions and particularities as well as their advantages and limitations in the solving of industrial problems. Section 1.3 is all about classical faceted-shell elements with 2D topology. In section 1.4 we will discuss about the 3D standard element and why it's not convenient to use it for shell like structure due to many locking problems. And finally in section 1.5 we will see how the standard 3D element is modified and enriched to give solid-shell finite elements capable to efficiently model shell like structures.

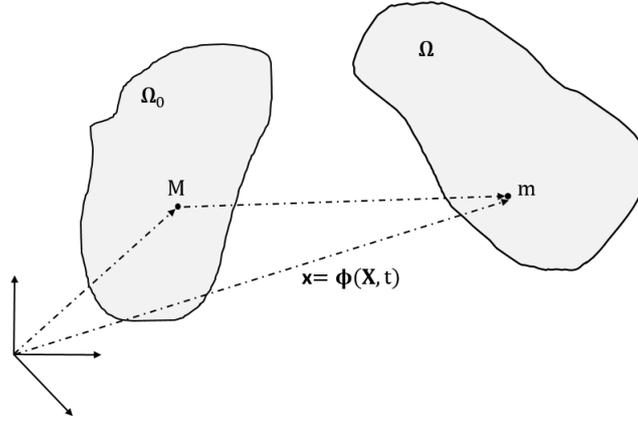


Figure 1.1: Initial and final configurations of a transformation

1.2 Nonlinear analysis of mechanical structures

Let's consider a continuous domain with a reference configuration Ω_0 which undergoes a transformation over time ϕ , making it pass into a current configuration Ω . Following this transformation, a point at the initial position $\mathbf{X} \in \Omega_0$ in the reference configuration is transformed into a point $\mathbf{x} \in \Omega$, with $\mathbf{x} = \phi(\mathbf{X}, t)$, in the current configuration. So the displacement field between the two positions is

$$\mathbf{u}(\mathbf{X}, t) = \mathbf{x} - \mathbf{X} = \phi(\mathbf{X}, t) - \mathbf{X} \quad (1.1)$$

To describe the motion of the solid between the current configuration and the reference configuration a gradient tensor of transformation \mathbf{F} is defined by

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbf{1} + \nabla \mathbf{u} \quad (1.2)$$

An infinitesimal vector $d\mathbf{x}$ of the deformed configuration can be expressed as a function of an infinitesimal vector $d\mathbf{X}$ of the reference configuration using the transformation gradient tensor \mathbf{F}

$$d\mathbf{x} = \mathbf{F} \cdot d\mathbf{X}. \quad (1.3)$$

1.2.1 Green Lagrange strain tensor

To evaluate the deformation of the solid, the Green-Lagrange deformation tensor \mathbf{E} is defined as follows:

$$\mathbf{E} = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{1}) = \frac{1}{2} (\nabla \mathbf{u} + \nabla^T \mathbf{u} + \nabla^T \mathbf{u} \nabla \mathbf{u}) \quad (1.4)$$

1.2.2 Stress tensor

To evaluate the stresses that apply to a section of a solid, the Cauchy tensor $\boldsymbol{\sigma}$ is generally calculated. It is the natural stress tensor since it directly measures the force per

unit area in the current configuration. Let's consider an oriented surface element $\mathbf{N}dS$ in the reference configuration, where dS is the surface of the element and \mathbf{N} is its unitary outgoing normal, and after transformation this surface element is deformed into a surface element $\mathbf{n}ds$ in the current configuration with ds in the scalar measurement of the surface and \mathbf{n} its outgoing normal. Assuming that on this surface element (current configuration) an elementary force $d\mathbf{f}$ is applied, the Cauchy tensor, is written as follows

$$d\mathbf{f} = \boldsymbol{\sigma}\mathbf{n}ds \quad (1.5)$$

The second Piola Kirchhoff stress tensor is then described as follows :

$$\mathbf{S} = J\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-T} \quad (1.6)$$

Where $J = \det\mathbf{F}$. If we assume the strain energy density to be W , the constitutive relation can be obtained by differentiating W with respect to the Lagrangian strain \mathbf{E} , to obtain

$$\mathbf{S} = \frac{\partial W(\mathbf{E})}{\partial \mathbf{E}} \quad (1.7)$$

1.2.3 Principle of Minimum Potential Energy

The weak form of a nonlinear elastic system can be obtained from the principle of minimum potential energy. The potential energy of an elastic system is the difference between the stored strain energy π_{int} and the work done by external forces π_{ext} . For simplicity, it is assumed that the applied forces are conservative, which means that the applied load is independent of deformation. Therefore, these forces can be transformed to the undeformed geometry. Using the strain energy and work done by applied forces, the potential energy of an elastic system can be obtained as

$$\begin{aligned} \pi(\mathbf{u}) &= \pi_{\text{int}}(\mathbf{u}) - \pi_{\text{ext}}(\mathbf{u}) \\ &= \int_{\Omega_0} W(\mathbf{E})d\Omega_0 - \int_{\partial\Omega_0} \mathbf{u}^T \cdot \mathbf{f}d(\partial\Omega_0) - \int_{\Omega_0} \mathbf{u}^T \cdot \mathbf{b}d\Omega_0 \end{aligned} \quad (1.8)$$

where \mathbf{f} represents a surface force and \mathbf{b} a body force. In order to find the displacement at the minimum potential energy, a perturbation method is often used. Let us assume that the displacement field \mathbf{u} is perturbed in the direction of $\delta\mathbf{u}$ (arbitrary) and τ is the parameter that controls the perturbation size. The perturbed displacement is denoted by

$$\mathbf{u}_\tau = \mathbf{u} + \tau\delta\mathbf{u} \quad (1.9)$$

$\delta\mathbf{u}$ then corresponds to the virtual displacement in the principle of virtual work. The first variation of the potential energy can be obtained by taking the first-order variation of $\pi(\mathbf{u})$ in the direction of $\delta\mathbf{u}$, as

$$\delta\pi(\mathbf{u}, \delta\mathbf{u}) = \frac{d}{d\tau}\pi(\mathbf{u} + \tau\delta\mathbf{u})|_{\tau=0} \quad (1.10)$$

where the symbol δ represents the first-order variation of a function. The process of variation is similar to the differentiation of a function. Using the potential energy in equation (1.8) and equating the first variation to zero, the following variational equation can be obtained

$$\delta\pi(\mathbf{u}, \delta\mathbf{u}) = \int_{\Omega_0} \frac{\partial W(\mathbf{E})}{\partial \mathbf{E}} : \delta\mathbf{E} \, d\Omega_0 - \int_{\partial\Omega_0} \delta\mathbf{u}^T \cdot \mathbf{f}d(\partial\Omega_0) - \int_{\Omega_0} \delta\mathbf{u}^T \cdot \mathbf{b}d\Omega_0 \quad (1.11)$$

For the variation of the strain energy, using the chain rule of differentiation, the strain energy density is differentiated with respect to the Lagrangian strain, and then the variation of the Lagrangian strain is taken from its definition in equation (1.4)

$$\begin{aligned} \delta\mathbf{E}(\mathbf{u}, \delta\mathbf{u}) &= \frac{1}{2} (\nabla\delta\mathbf{u} + \nabla^T\delta\mathbf{u} + \nabla^T\mathbf{u}\nabla\delta\mathbf{u} + \nabla^T\delta\mathbf{u}\nabla\mathbf{u}) \\ &= \text{sym}(\nabla\delta\mathbf{u}^T\mathbf{F}) \end{aligned} \quad (1.12)$$

where $\text{sym}(\cdot)$ denotes the symmetric part of a tensor. The principle of minimum potential energy says that if the system is in equilibrium, the variation in equation (1.11) must vanish for all $\delta\mathbf{u}$ that belongs to the space of kinematically admissible displacements $Z = \{\mathbf{v} \in (H^1(\Omega_0))^d \mid \mathbf{v} = 0 \text{ in } \partial\Omega_0\}$. The variational equation for the nonlinear elastic system can be written as

$$a(\mathbf{u}, \delta\mathbf{u}) = l(\delta\mathbf{u}), \forall \delta\mathbf{u} \in Z \quad (1.13)$$

where $a(\mathbf{u}, \delta\mathbf{u})$ is the energy form and $l(\delta\mathbf{u})$ is the load form, defined as

$$a(\mathbf{u}, \delta\mathbf{u}) = \int_{\Omega_0} \frac{\partial W(\mathbf{E})}{\partial \mathbf{E}} : \delta\mathbf{E} \, d\Omega_0, \quad (1.14)$$

and

$$l(\delta\mathbf{u}) = \int_{\partial\Omega_0} \delta\mathbf{u}^T \cdot \mathbf{f}d(\partial\Omega_0) + \int_{\Omega_0} \delta\mathbf{u}^T \cdot \mathbf{b}d\Omega_0 \quad (1.15)$$

Note that $a(\mathbf{u}, \delta\mathbf{u})$ and $l(\delta\mathbf{u})$ are linear with respect to $\delta\mathbf{u}$ but are nonlinear with respect to displacement \mathbf{u} . The nonlinearity comes from the fact that the stress and strain implicitly depend on \mathbf{u} .

1.2.4 Linearization (Tangent Stiffness)

In practice, the nonlinear variation equation (1.13) cannot be solved easily due to the nonlinearity involved in the displacement-strain relation, or in the constitutive relation. Newton-Raphson iterative method through a sequence of linearization is generally used to solve this nonlinear equation. If we assume that the equilibrium in equation (1.13) is not satisfied, then, the difference between the left- and right-hand sides is defined as a residual,

$$R = a(\mathbf{u}, \delta\mathbf{u}) - l(\delta\mathbf{u}) \quad (1.16)$$

In the Newton-Raphson method, the Jacobian of the residual is required in each iteration. Since the Jacobian in a one-dimensional problem is nothing but a tangent line

at the current solution, it is often called a tangent stiffness, and the process is called linearization. Let the linearization of a function $f(\mathbf{x})$ in the direction of $\Delta\mathbf{u}$ is denoted as

$$L[f] = \frac{d}{d\omega} f(\mathbf{x} + \omega\Delta\mathbf{u})|_{\omega=0} = \frac{\partial f^T}{\partial \mathbf{x}} \Delta\mathbf{u} \quad (1.17)$$

If right superscript k denotes the iteration counter, then the linear incremental solution procedure of the nonlinear equation $f(\mathbf{x}^{k+1}) = 0$ becomes

$$\begin{aligned} \frac{\partial f^T}{\partial \mathbf{x}^k} \Delta\mathbf{u}^k &= -f(\mathbf{x}^k) \\ \mathbf{u}^{k+1} &= \mathbf{u}^k + \Delta\mathbf{u}^k \\ \mathbf{x}^{k+1} &= \mathbf{X} + \mathbf{u}^{k+1} \end{aligned} \quad (1.18)$$

This way the nonlinear system (1.16) is solved iteratively until the residual term vanishes. The nonlinear equation (1.13) can be linearized following the same procedure. Since the load is assumed to be independent of displacement, it is not necessary to linearized it. Linearization of the energy form in equation (1.13) is written as follows

$$L[a(\mathbf{u}, \delta\mathbf{u})] = \int_{\Omega_0} \Delta\mathbf{S} : \delta\mathbf{E} + \mathbf{S} : \Delta(\delta\mathbf{E}) d\Omega \quad (1.19)$$

where $\Delta\mathbf{S}$ is the stress increment and $\Delta(\delta\mathbf{E})$ is the increment of strain variation. For simplicity, let's for example consider a St. Venant-Kirchhoff material, the stress-strain relation is linear, and thus, the increment of stress can be written as

$$\Delta\mathbf{S} = \frac{\partial \mathbf{S}}{\partial \mathbf{E}} : \Delta\mathbf{E} = \mathbf{D} : \Delta\mathbf{E} \quad (1.20)$$

where \mathbf{D} is a four-order constitutive tensor and $\Delta\mathbf{E}$ is the increment of Lagrangian strain. By noting that the increment of deformation gradient is $\Delta\mathbf{F} = \nabla_0\Delta\mathbf{u}$, the increment of Lagrangian strain and its variation can be obtained as follows

$$\Delta\mathbf{E}(\mathbf{u}, \Delta\mathbf{u}) = \text{sym}(\nabla_0\Delta\mathbf{u}^T \mathbf{F}) \quad (1.21)$$

and

$$\Delta(\delta\mathbf{E})(\Delta\mathbf{u}, \delta\mathbf{u}) = \text{sym}(\nabla_0\delta\mathbf{u}^T \nabla_0\Delta\mathbf{u}) \quad (1.22)$$

Thus, the linearization of the energy form can be explicitly derived with respect to displacement and its variation as

$$L[a(\mathbf{u}, \delta\mathbf{u})] = \int_{\Omega_0} \delta\mathbf{E} : \mathbf{D} : \Delta\mathbf{E} + \mathbf{S} : \Delta(\delta\mathbf{E}) d\Omega = a^*(\mathbf{u}; \Delta\mathbf{u}, \delta\mathbf{u}) \quad (1.23)$$

We use the notation $a^*(\mathbf{u}; \Delta\mathbf{u}, \delta\mathbf{u})$ to show that the equation implicitly depends on the total displacement \mathbf{u} and is bilinear with respect to $\Delta\mathbf{u}$ and $\delta\mathbf{u}$. The first integrand of equation (1.23) depends on the stress-strain relation. Since it is similar to the stiffness

term in linear systems, it is called the tangent stiffness. On the other hand, the second integrand does not exist in linear systems. It only appears in geometric nonlinear problems. It's the initial stiffness or geometric stiffness. Let's consider that the current load step is t_n and the current iteration counter be k . Assuming that the applied loads are independent of displacement, the linearized incremental equation (1.16) is obtained as follows

$$a^*({}^n\mathbf{u}^k; \Delta\mathbf{u}^k, \delta\mathbf{u}) = l(\delta\mathbf{u}) - a({}^n\mathbf{u}^k, \delta\mathbf{u}) \quad (1.24)$$

and the total displacement is updated using

$${}^n\mathbf{u}^{k+1} = {}^n\mathbf{u}^k + \Delta\mathbf{u}^k \quad (1.25)$$

Note that equation (1.24) is in the form of ${}^n\mathbf{K}^k \cdot \Delta\mathbf{u}^k = {}^n\mathbf{R}^k$ and is solved iteratively until the residual vanishes, which means that the original nonlinear equation is satisfied.

1.2.5 Finite Element Formulation

So far, formulations and solution procedures of nonlinear problems have been discussed in the continuum domain. In practice, the structure is discretized by finite elements and the equilibrium equations are applied to these elements. This section does not aim to recall all finite element theory but to provide elements for a better understanding of the following paragraphs. It will also allow us to have clear ideas on the steps of construction of a finite element and at the same time introduces some notations. A detailed presentation of the finite element method is made by Zienkiewicz [ZIE 77], Dhett [DHA 05], or Batoz [BAT 90b]. Finite element approximation is a move from the search for a continuous variable (the displacement at any point) to the search for discrete variables (nodal unknowns). The link between the two is made through the use of interpolation functions that are generally polynomial. In the iso-parametric mapping and displacement-based implementation of finite elements method, the approximation of the initial \mathbf{X} and current geometry vector \mathbf{x} and displacement vector \mathbf{u} at the element level are read as

$$\mathbf{X}^e = \sum_{I=1}^{no} N^I \mathbf{X}^I \quad ; \quad \mathbf{x}^e = \sum_{I=1}^{no} N^I \mathbf{x}^I \quad ; \quad \mathbf{u}^e = \sum_{I=1}^{no} N^I \mathbf{u}^I \quad (1.26)$$

where the superscript e refers to quantities at the element level and no being the number of nodes of the element type considered (for example : $no = 8$ for hexahedral element or $no = 6$ for prismatic element). N^I are the standard shape functions, \mathbf{X}^I , \mathbf{x}^I and \mathbf{u}^I are respectively the vectors of nodal coordinates and displacements. The displacement variation and increment required by the linearization of the variational are interpolated as follows

$$\delta\mathbf{u}^e = \sum_{I=1}^{no} N^I \delta\mathbf{u}^I \quad \text{and} \quad \Delta\mathbf{u}^e = \sum_{I=1}^{no} N^I \Delta\mathbf{u}^I \quad (1.27)$$

The variation $\delta\mathbf{E}$ and increment $\Delta\mathbf{E}$ of the Green-Lagrange strain at the element level are interpolated and written in Voigt form as follows

$$\delta\mathbf{E} = \mathbf{B}\delta\mathbf{U} \quad \text{and} \quad \Delta\mathbf{E} = \mathbf{B}\Delta\mathbf{U} \quad (1.28)$$

where \mathbf{B} is the standard strain-displacement matrix and is function of displacement \mathbf{u}^e and \mathbf{U} a larger vector containing the nodal displacements \mathbf{u}^l . The increment of strain variation $\Delta(\delta\mathbf{u})$ is written as follows, (see equation (1.22))

$$\Delta(\delta\mathbf{E}) = \delta\mathbf{U}^T \mathcal{B}\Delta\mathbf{U} \quad (1.29)$$

where \mathcal{B} is the standard nonlinear strain-displacement matrix. Equation (1.23) can then be discretized as follows

$$L[a(\mathbf{u}, \delta\mathbf{u})] = \delta\mathbf{U}^T \left(\int_{\Omega_0} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega + \int_{\Omega_0} \mathcal{B} \mathbf{S} d\Omega \right) \Delta\mathbf{U} = \delta\mathbf{U}^T \mathbf{K}_T \Delta\mathbf{U} \quad (1.30)$$

Where \mathbf{K}_T is the tangent stiffness matrix. In addition, the discrete external and internal force vector can be derived from the definition of the load form as

$$l(\delta\mathbf{u}) = \delta\mathbf{U}^T \mathbf{F}_{\text{ext}} \quad (1.31)$$

$$a(\mathbf{u}, \delta\mathbf{u}) = \delta\mathbf{U}^T \int_{\Omega_0} \mathbf{B}^T \mathbf{S} d\Omega = \delta\mathbf{U}^T \mathbf{F}_{\text{int}} \quad (1.32)$$

The discretized version of incremental equation (1.23) can now be written in the form of finite element matrix equation as

$$\delta\mathbf{U}^T \mathbf{K}_T \Delta\mathbf{U} = \delta\mathbf{U}^T (\mathbf{F}_{\text{ext}} - \mathbf{F}_{\text{int}}) \quad (1.33)$$

The above linear system of equations needs to be solved iteratively until the residual force (right-hand side) vanishes. Now that the basic fundamentals of continuum mechanics and finite elements method have been recalled, we can discuss about the different numerical methods used to model shell like structures.

1.3 Conventional 2D shell finite element

In this section classical and degenerated shell elements are briefly introduced. We will briefly discuss about the disadvantages of the conventional shell elements, such as modified material models, variation of thickness, contradictions of assumptions etc, without going deep in the equations (for more details about these formulations, the reader is referred to the appendix C). There are two approaches to formulate conventional shell finite elements. The first formulation uses the classical strain, displacement and momentum (equilibrium) equations for shell to develop a weak form of the equilibrium equations. These elements are called classical shell elements. The second formulation is directly derived from continuum element by structural assumptions on the weak form or on the discrete equations. The continuum finite element is modified so that it behaves like shell; this is called the continuum based approach. [AHM 70, DVO 84].

1.3.1 Classical shell theories

A plate is an area in which one dimension, called thickness, is smaller than the other two. It can be made of a homogeneous material or be obtained by the composition of several layers of different materials. A shell therefore is considered as a curved form of a plate and its structural action is a combination of stretching and bending [ZIE 00]. It is possible to perform a finite element analysis of a shell by using what is called a facet representation, meaning the shell surface is replaced by a flat triangular and/or quadrilateral plate elements in which a membrane stiffness (membrane element) is superposed on a bending stiffness (plate bending element). Such a model is understandably inaccurate in the sense that with very coarse meshes, they do not capture the bending-stretching coupling of thin shell behavior. There are two main theories to study plate-like structures:

- **Kirchhoff-love theory:** based on the assumption of preservation of normal. In other words, all material points located on a normal to the undeformed mid-surface remain on the same normal to the deformed mid-surface. In this model, which is well adapted to thin plates, the transverse shear is assumed to be zero. Kirchhoff-love assumptions are generally accepted if $\frac{L}{h} > 20$, L being a characteristic dimension of the plate in the mean plane.
- **Mindlin-Reissner theory:** based on the kinematic hypothesis of straight sections : the normal to the mid-surface remains straight. In this model, adapted to the study of thick plates, transverse shear is taken into account. The Mindlin-Reissner hypotheses are generally accepted if $5 < \frac{L}{h} \leq 20$, L being a characteristic dimension in the mean plane.

Shell theories, see [ZIE 00], proved that the Kirchhoff-Love assumptions are the most accurate in predicting the behavior of thin shells. For thick shells, the Mindlin-Reissner assumptions are more accurate because transverse shear effects become important. Generally speaking, Kirchhoff's theory can be interpreted as a special case of Reissner-Mindlin theory. Indeed, the straight section hypothesis includes the so-called normal conservation hypothesis. Also a good finite element model based on the Reissner-Mindlin theory should give results in accordance with the Kirchhoff-Love theory if the structure is thin (negligible CT) (see [BAT 90a]). In plate theory, contrary to the three-dimensional solid theory, the elementary mechanical entity is not a material particle (x_1, x_2, x_3) but a material segment of coordinates (x_1, x_2, z)

$$N(x_1, x_2, z) = \{(x_1, x_2, z), |z| \leq h(x_1, x_2)\} \quad (1.34)$$

A plate is therefore a family of $N(x_1, x_2, z)$ line segments which, in the reference configuration, are normal and centered on the mean plane (O, x_1, x_2) . The fundamental premise of plate theory is that each of these segments has a rigid body motion. The motion of any segment is then described by a six-component kinematic torsor. Nevertheless, the rotation of a segment around its axis being indifferent, it remains to consider five kinematic variables: the membrane displacements (u, v) , the transverse displacement w and the

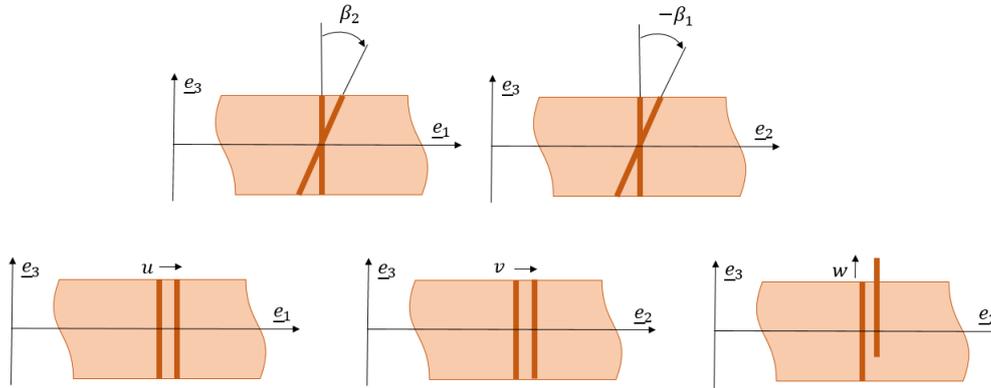


Figure 1.2: plate theory kinematic

two rotations around the axes of the mean plane (β_1, β_2) (see Figure 1.2). Shell elements based on Mindlin-Reissner or Kirchhoff-love theories assume the following kinematic

1. The length of the normal remains unchanged throughout deformation $\epsilon_{zz} = 0$.
2. Transverse normal stresses are negligible $\sigma_{zz} = 0$

When Poisson's ratio is not equal to zero the latter two assumptions are contradictory because the normal must stretch when $\sigma_{zz} = 0$. Furthermore, employing the second assumption requires modifications of the 3D-material law. This work is not simple in such an approach, especially for complicated material laws which are described only for 3D-continuum. Moreover, due to the presence of rotational degree of freedom, the equations for nonlinear shells are very complex and difficult to deal with. This makes this theory very difficult to use in the modeling of nonlinear shell problems. There is still a disagreement as to what are the best nonlinear classical shell equations [STO 95]. More details of the formulation of solid-shell are given in appendix C.

1.3.2 Degenerated shell elements

Among the first to attempt to provide an answer to the problems of shell were Ahmad and co-authors [AHM 70] for proposing a degenerated shell finite element. It is simply a matter of flattening a volume element into a shell element. An example for degenerating a hexahedral finite element is that each pair of nodes having the same ξ and η coordinates in the reference frame is replaced by a middle node of coordinates $(\xi, \eta, \zeta = 0)$.

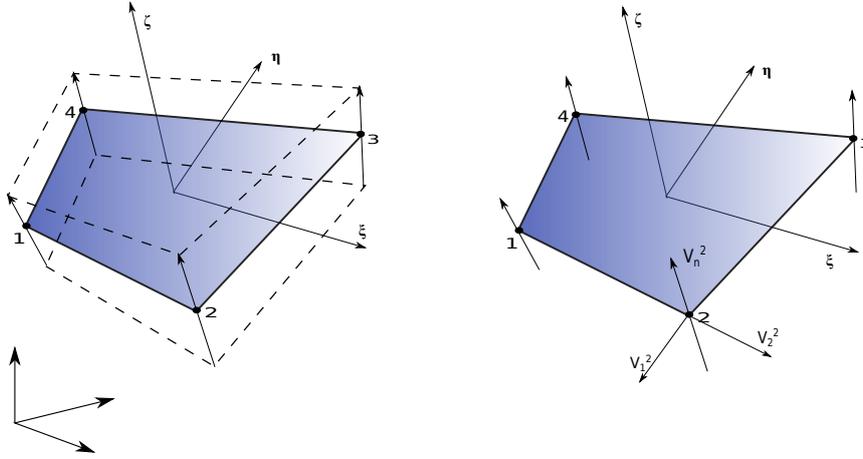


Figure 1.3: Ahmad degenerated shell finite element

The shell element is thus defined by the 4 (if the flattened volume element had 8 nodes) nodes delimiting its mid-surface. On each of these nodes, a vector is defined and directs the segment connecting the corresponding upper and lower vertices. In the hypothesis of small deformations a point of the shell element, at instant k , in the natural base (ξ, η, ζ) , is defined by its cartesian coordinates as follows

$${}^k\mathbf{x}(\xi, \eta, \zeta) = \sum_{i=1}^4 N_i(\xi, \eta) {}^k\mathbf{x}_i + \frac{\zeta}{2} \sum_{i=1}^4 h_i N_i(\xi, \eta) {}^k\mathbf{V}_n^i \quad (1.35)$$

where : N_i is the isoparametric interpolation function corresponding to node i

${}^k\mathbf{x}_i$ position of node i at instant k

h_i the thickness at node i (assumed invariant)

${}^k\mathbf{V}_n^i$ the vector linking the initially degenerated nodes i . $\|{}^k\mathbf{V}_n^i\| = 1$. Assuming small perturbations, the displacement at the moment k is written (for simplicity we omit the superscript k for the displacement) as follows

$$\mathbf{u}(\xi, \eta, \zeta) = \sum_{i=1}^4 N_i(\xi, \eta) \mathbf{u}_i + \frac{\zeta}{2} \sum_{i=1}^4 h_i N_i(\xi, \eta) ({}^k\mathbf{V}_n^i - {}^0\mathbf{V}_n^i) \quad (1.36)$$

If we define at the node i two vectors ${}^0\mathbf{V}_1^i$ and ${}^0\mathbf{V}_2^i$ such as the base $({}^0\mathbf{V}_1^i, {}^0\mathbf{V}_2^i, {}^0\mathbf{V}_n^i)$ is orthonormal, we can write, assuming the small rotations that :

$$\begin{aligned} {}^k\mathbf{V}_n^i &= {}^0\mathbf{V}_n^i + {}^k\theta_i \wedge {}^0\mathbf{V}_n^i \\ {}^k\theta_i &= \alpha_i^0 \mathbf{V}_1^i + \beta_i^0 \mathbf{V}_2^i \\ {}^k\mathbf{V}_n^i &= {}^0\mathbf{V}_n^i + \beta_i^0 \mathbf{V}_1^i - \alpha_i^0 \mathbf{V}_2^i \end{aligned}$$

Therefore the displacement of a point of the element can be written as follows

$$\mathbf{u}(\xi, \eta, \zeta) = \sum_{i=1}^4 N_i(\xi, \eta) \mathbf{u}_i + \frac{\zeta}{2} \sum_{i=1}^4 h_i N_i(\xi, \eta) (\beta_i^0 \mathbf{V}_1^i - \alpha_i^0 \mathbf{V}_2^i) \quad (1.37)$$

Thus each node of this element will also have 5 degrees of freedom (3 translations and the 2 rotations of the vector associated with this node). At first this may seem very similar to what is known for Reissner/Mindlin shell elements but there are fundamental distinctions : Ahmad's degenerate finite elements function under a purely three-dimensional formulation. Also quadrilaterals obtained by degeneracy of the 3D elements support warping (non-planarity of the nodes) much better. However, one can note similar hypotheses to those governing the classical plates :

1. the segments connecting two nodes (directed by a director vector) are straight through the transformation.
2. A segment initially normal to the mean surface remains inextensible during the transformation (which means that the deformation in the thickness is nil).
3. The stress in the thickness is nil. This necessitates modifying the 3D constitutive law to respect this hypothesis.

It is immediately noticeable that the second hypothesis necessarily implies to remain in the hypothesis of small deformations. Moreover, this hypothesis is in contradiction with the one stipulating that the stress in the thickness is nil (3rd hypothesis) but also with the fundamental hypothesis of mass conservation. If a plate element is extended horizontally, it should lose thickness to conserve its mass. Thus, in order to satisfy all these hypotheses at the same time, these elements operate with modified constitutive laws. Unlike the classical shell element mentioned above, the continuum-base approach is very straight forward. It is applicable to arbitrarily large deformations shell problems [BOI 94] and yield excellent results for many structural problems. Although Ahmad's formulation is innovative, in terms of plate modeling, the degenerated finite elements can suffer from severe locking problems. Several authors have proposed solutions to alleviate these locking problems. These include the work of Brendel and Ramm [BRE 80], Parisch [PAR 95], Hughes [HUG 81a], Bathe and Dvorkin [DVO 84, DVO 95a], Bischoff [BIS 97] among others. The main limitation of the two types of classical shell finite elements discussed above is the plane stress hypothesis. In addition to being in contradiction with the zero normal strain, it requires the modification of the constitutive law. For some constitutive laws, this is numerically feasible but for others, notably certain constitutive laws of natively three-dimensional concrete, plane stresses hypotheses are none-sense. Moreover, these flattened finite elements do not, to our knowledge, handle structures with variable thickness and their connection with other structures of thicker types is hardly manageable. As a result of all these difficulties, studies are increasingly turning to 3D type finite elements which are generally more robust, operate with three-dimensional tensors and handle several types of nonlinearities encountered in mechanics.

length	a	1m
width	a	1m
Thickness	t	0.1 then 0.01m
Young modulus	E	$21 \times 10^9 \text{Pa}$
Poisson ratio	ν	0.0
Applied force	F	100N

Table 1.1: Geometrical and material characteristics of the bent plate

1.4 The 3D standard finite element

Another way to model shell like structure would be to use 3D standard isoparametric finite elements as an alternative to the plane stress hypothesis. These elements operate with purely three-dimensional behaviour laws and can handle several nonlinearities such as contact/friction, fracture mechanics, elastic-plastic or large geometrical transformations. Unfortunately they have the disadvantage of being too rigid to reproduce correctly the behaviour of thin structures. They require very fine discretization to give good results, in the modeling of shell like structures. This is very expensive in terms of calculation time.

1.4.1 Case study

To illustrate the difficulty of standard 3D finite elements for modeling thin structures, a simple study is carried out. Let us consider a thin square plate with side length $a = 1m$ and thickness $e = 0.01m$, clamped on one side and undergoing a bending force on the opposite side. The material characteristics of the plate are given in the table 1.1. The plate is then meshed with standard 3D finite elements. A mesh convergence study is performed and compared with a reference solution given by a Discrete-Kirchhoff-quadrangle (DKQ) mesh. The CPU time of each study is also collected and introduced in the comparisons. The results are reported in the table 1.2. On the latter table we can see how difficult it is for the 3D standard finite element to give a good result for this simple bending problem. For the same number of elements, the 3D standard finite element gives a displacement of $9.5e^{-7}m$, where the DKQ element gives $1.9e^{-3}m$. One can refine by increasing the number of elements by $\times 3000$ and thus multiply the CPU time by a factor of 1000 and still not get the right result. This shows the difficulty of first order standard finite elements to model bending problems in thin structures. Due to the tri-linearity of their shape functions, these elements suffer from several locking phenomena when modeling bending problems of thin structures.

Mesh	Number of elements	CPU time	results	Ref
$5 \times 5 \times 1$	25	0.05s	$9.5e-7m$	$1.90e-3m$
$15 \times 15 \times 10$	2250	0.65s	$8.2e-6m$	$1.90e-3m$
$50 \times 50 \times 30$	75 000	65.54s	$6.3e-5m$	$1.90e-3m$

Table 1.2: Displacement of the eight-nodes standard element

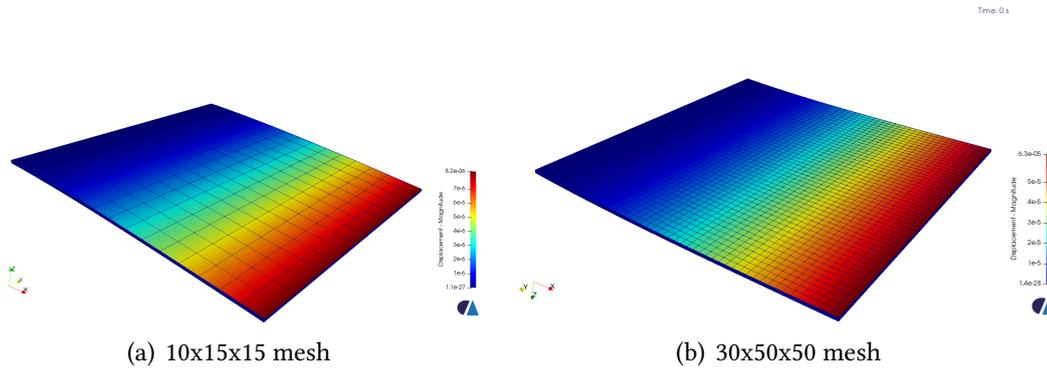


Figure 1.4: Simple Bending of a plate with 3D standard finite elements

1.4.2 Locking phenomena

To investigate the locking phenomena the low-order standard solid element goes trough, let's consider the following trilinear displacement field, which is conventionally employed for the eight-node standard element, see Figure 1.5. To facilitate understanding, the analysis is restricted to a cubic geometry element so that the physical system (x, y, z) and the isoparametric natural system (ξ, η, ζ) can be used interchangeably. In the frame (x, y, z) , the displacements (u, v, w) of any point can be written in the following trilinear form

$$\begin{aligned}
 u &= a_0 + a_1x + a_2y + a_3z + a_4xy + a_5yz + a_6xz + a_7xyz \\
 v &= b_0 + b_1x + b_2y + b_3z + b_4xy + b_5yz + b_6xz + b_7xyz \\
 w &= c_0 + c_1x + c_2y + c_3z + c_4xy + c_5yz + c_6xz + c_7xyz
 \end{aligned} \tag{1.38}$$

where a_i, b_i , and $c_i (i = 0, \dots, 7)$ are constant coefficients. For the sake of simplicity, locking effects are simply considered with infinitesimal strains. It means we consider infinitesimal strains ε instead of Green-Lagrange strain \mathbf{E} . All types of locking and their remedies for low-order solid elements will be mentioned in details in the following section.

1.4.2.1 Transverse shear locking

Transverse shear locking occurs when the EF discretization overestimates shear stiffness at the expense of bending stiffness, when modeling a thin or very thin structure. Shear

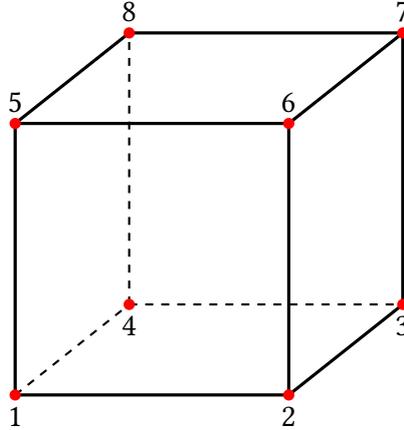


Figure 1.5: Hexahedral solid element

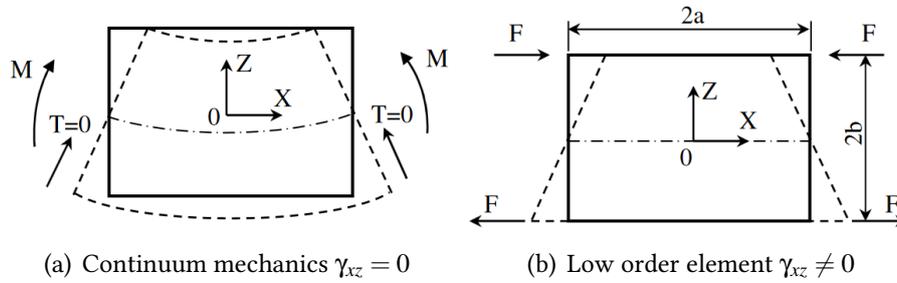


Figure 1.6: Flexion in continuum mechanics Vs flexion of low order element

locking is the most frequently encountered locking phenomenon but also the most severe locking. Not only does it slow down convergence but its numerical management requires considerable effort. In classical elasticity theory, when an element is subject to pure bending, the shear term should be zero. But this is not the case with the tri-linear form defined in the hexahedral element. the shear strain is written as follows

$$\gamma_{xz} = \frac{\partial u}{\partial z} + \frac{\partial v}{\partial x} = (a_1 + b_1) + (a_5 + b_6)z + a_4x + b_4y + a_7xz + b_7yx \quad (1.39)$$

In fact, the above shear strain can only be zero if all the coefficients of the linear shape are zero. Which is impossible since this would be opposed to the completeness of the shape function space. To better illustrate this phenomenon, it is simpler to consider the deformation modes of a beam in a bending problem. Let's consider a rectangular beam as in the figures (1.6.a) and (1.6.b). Assuming that the deformation of the beam is independent of the Y-direction, the displacements can be obtained in the following way

$$\begin{aligned} u &= c_1 + c_3x + c_5z + c_7xz \\ w &= c_2 + c_4x + c_6z + c_8xz \end{aligned} \quad (1.40)$$

Logically, this means that the hexahedral element with tri-linear form functions is reduced to a quadrangular element with bilinear functions. The coefficients c_1 and c_2 are

rigid body deformations, c_3 and c_4 are constant deformations, and c_7 and c_8 are linear deformation modes. During pure bending in the (x, z) plane (see figures 1.6) the shear mode c_7 remains active in the normal strain ϵ_x . In other words, with the 1st order shape functions, it is impossible to have a strain ϵ_x without it being accompanied by a parasitic shear term.

Modes	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
u	c_1	0	c_3x	0	c_5z	0	c_7xz	0
w	0	c_2	0	c_4x	0	c_6z	0	c_8xz
ϵ_x	0	0	c_3	0	0	0	c_7z	0
ϵ_z	0	0	0	0	0	c_6	0	c_8x
γ_{XZ}	0	0	0	c_4	c_5	0	c_7x	c_8z

Table 1.3: Deformation modes of bilinear element

The phenomenon of transverse shear locking then occurs when this term is preponderant. The example of the beam clearly shows this fact. When a beam is bent by a moment couple M (figure 1.6.a), continuum mechanic predicts the following stresses

$$\begin{cases} \sigma_z = \frac{M}{I}z \\ \sigma_x = \tau_{xz} = 0 \end{cases} \quad (1.41)$$

I being the moment of inertia of the beam. By integrating equation (1.41), it comes:

$$\begin{cases} u = \frac{M}{EI}zx + c_1z + c_2 \\ w = -\frac{M}{2EI}(x^2 + vz^2) - c_1x + c_3 \end{cases} \quad (1.42)$$

c_1 , c_2 and c_3 are integration constants. By imposing a symmetry boundary condition with respect to the plane $x=0$, the vertical displacements of the 4 nodes are zero. This results in the following displacements

$$\begin{cases} u = \frac{M}{EI}xz \\ w = \frac{M}{2EI}(a^2 - x^2) - \frac{M}{2EI}v(b^2 - z^2) \end{cases} \quad (1.43)$$

Which leads to the following strains

$$\begin{cases} \varepsilon_x = \frac{M}{EI}z \\ \varepsilon_z = -\frac{M\nu}{EI}z \quad \text{if } \nu \neq 0, \quad \varepsilon_z = 0 \text{ else} \\ \gamma_{xz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} = 0 \end{cases} \quad (1.44)$$

This solution satisfies the conditions of pure bending. however, for the finite element solution, when pure bending occurs, only the mode c_7 is active (see table 1.3), it means

$$\begin{cases} u = c_7xz \\ w = 0 \end{cases} \quad (1.45)$$

Therefore the finite element solution commits an error $\Delta\gamma_{xz} = c_7x$ which does not vanish and is likely to be preponderant with respect to the pure bending mode, when the thickness of the structure becomes small. This is the cause of shear locking. Note also that, when the Poisson ratio is not null, the finite element commits also an error in the normal strain $\Delta\varepsilon_z = -\frac{M\nu}{EI}z$, which lead to Poisson thickness locking (see next subsection).

In conclusion, note that transverse shear locking occurs because normal strains of linear elements are coupled by shear strains. Therefore, these elements do not have a pure bending mode that allows a pure bending behaviour to be correctly modelled. When the plate thickness becomes thin, shear modes become preponderant and absorb a large part of the bending deformation energy. This results in very small bending deformations compared to what would be obtained by analytical mechanics. Increasing the number of elements will allow a more accurate modeling of the curvature, and reduce the effects of shear locking. Shear locking can also be prevented through the use of higher-order finite elements.

1.4.2.2 Poisson thickness locking

Considering still the bending beam problem, the analytical solution gives a pinching strain worth: $\varepsilon_z = -\frac{M\nu}{EI}z$, whereas the linear finite element model gives

$$\varepsilon_z = \frac{\partial w}{\partial z} = c_6 + c_8x \quad (1.46)$$

This means that at a given position x , the pinching strain is constant in the thickness direction. However in a real structure, the analytical solution of the beam gives a normal displacement $\varepsilon_x = \frac{Mz}{EI}$ which by Poisson effect leads to a pinching strain depending on the thickness direction $\varepsilon_z = -\nu\varepsilon_x$. With low-order finite elements, this is not satisfied. Thus the pinching stress

$$\sigma_z = \frac{E}{(1+\nu)(1-2\nu)} ((1-2\nu)\varepsilon_z + \nu\varepsilon_x) \quad (1.47)$$

is different from what an analytical solution would yield. This causes Poisson thickness locking for low order elements.

1.4.2.3 Volumetric locking

Volumetric locking is a stability problem encountered when finite elements, formulated in displacement, are used to model the behaviour of incompressible or quasi-incompressible materials. This instability is primarily related to the mathematical formulation of the problem. Indeed, in mechanics, a material is said to be incompressible when its Poisson ratio ν is 0.5. If it is deformed, such a material does not undergo any variation in volume and should verify the following relationship

$$\epsilon_v = \text{tr}(\boldsymbol{\epsilon}) = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (1.48)$$

However if one consider, the interpolations form in the sections above, the strain ϵ_v is rather written as follows :

$$\epsilon_v = (a_1 + b_2 + c_3) + (b_4 + c_6)x + (a_4 + c_5)y + (a_6 + b_5)z + (c_7xy + a_7yz + b_7zx) \quad (1.49)$$

In other words, it's impossible to satisfy the $\epsilon_v = 0$ condition. Thus, for such a material, the first-order finite elements cause an overrigidity of the elementary stiffness matrix inducing a locking solution. Hooke's law for continuous and isotropic materials is

$$\boldsymbol{\sigma} = \frac{E}{1 + \nu} \left(\boldsymbol{\epsilon} + \frac{\nu}{1 - 2\nu} \epsilon_v \mathbf{I} \right) \quad (1.50)$$

One can see right away that when the $\nu \rightarrow 0.5$, the ratio $\frac{\nu}{1-2\nu}$ tends towards infinity. Therefore, the elasticity matrix in the product $\mathbf{B}^T \mathbf{D} \mathbf{B}$, see equation (1.30) is composed of elements tending towards infinity. The resulting stiffness matrix becomes an infinite wide number, providing infinitely small displacement values.

1.4.2.4 Membrane locking

Membrane locking is a locking phenomenon that usually occurs when beam and shell type structures with high curvature are subject to bending. It is often confused with trapezoidal or transverse shear locking because the latter affects membrane effects, however they are completely different types of locking. To better illustrate the membrane locking phenomenon, consider a curved beam with a length of $2l$ and a radius of curvature R (Figure 1.7). The circumferential displacement is referred to as u and radial displacement as w . The curvilinear abscissa carried by the mid-line of the beam is called s . Membrane and bending deformations can be written as follows (see Prathap [PRA 01])

$$\begin{aligned} \epsilon &= u_{,s} + \frac{w}{R} \\ \chi &= \frac{u_{,s}}{R} - w_{,ss} \end{aligned} \quad (1.51)$$

From these equations, we can immediately see that at least C^0 continuity is required for the circumferential displacement and C^2 for the radial displacement. It is then possible

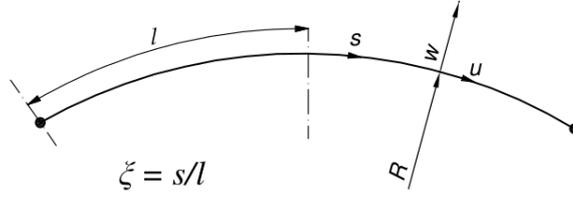


Figure 1.7: Curved beam

to write the polynomial form of the displacements

$$\begin{aligned} u &= a_0 + a_1 \xi \\ w &= b_0 + b_1 \xi + b_2 \xi^2 + b_3 \xi^3 \end{aligned} \quad (1.52)$$

where the coefficients a_i and b_i are generalized displacements corresponding to the degrees of freedom of the nodes. It is then possible to rewrite equations (1.51)

$$\begin{aligned} \varepsilon &= \left(\frac{a_1}{l} + \frac{b_0}{R} \right) + \frac{b_1}{R} \xi + \frac{b_2}{R} \xi^2 + \frac{b_3}{R} \xi^3 \\ \chi &= \left(\frac{a_1}{Rl} - \frac{2b_2}{l^2} \right) - \frac{6b_3}{l^2} \xi \end{aligned} \quad (1.53)$$

When this element is used to model a pure bending structure (without membrane extension) the membrane strain must vanish. This is only true if the following conditions are met

$$\begin{aligned} \left(\frac{a_1}{l} + \frac{b_0}{R} \right) &= 0 \\ \forall i = 1, 2, 3 \quad \frac{b_i}{R} &= 0 \end{aligned} \quad (1.54)$$

It's possible to arrange to physically meet the first condition. However, the other conditions $\frac{b_i}{R} = 0$ cannot be met without being in contradiction to the hypothesis of equation (1.52). The coefficients b_i , ($i = 1, 2, 3$) are then the cause of membrane locking.

In conclusion, membrane locking occurs if the interpolations are unable to model the in-extensional behavior in in-extensional bending problems. Consequently, a stiffening effect occurs when pure bending deformations are accompanied by parasitic membrane stresses.

1.4.2.5 Trapezoidal locking

Trapezoidal locking is a locking phenomenon encountered when the side faces of a 3D finite element are not perpendicular to the mid-plane of the element. It usually occurs when a mesh structure is curved (e.g. cylinder) or heavily deformed. However, curvature locking only occurs when the elements include thickness strains. In other words, degenerated shell elements and flat plate geometry for which the normal deformation

in thickness is assumed to be nil do not suffer from trapezoidal locking problems. As far as we know, this phenomenon was first illustrated by McNeal in 1987 [MAC 87]. To illustrate his reasoning, we consider a reference element transformed into a trapezoidal element of height $H = 2$ and width $W = H \times \Lambda$, Λ being the ratio between the reference width and the width of the trapezoid (see figure 1.8. The coordinates in the local frame can be written relative to the components in the reference base:

$$\begin{cases} x = \xi(1 - \alpha\zeta) \\ z = \zeta \\ \alpha\Lambda = \tan(\delta) \end{cases} \quad (1.55)$$

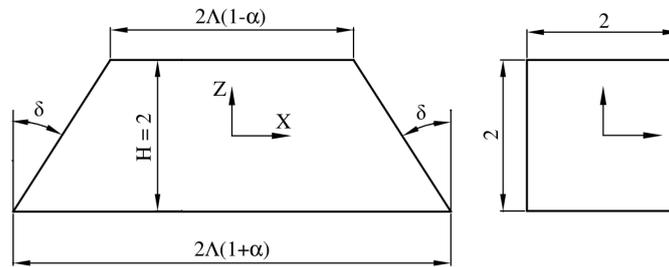


Figure 1.8: Trapezoidal element

To simplify, let's consider the case of a beam such that $\frac{M}{EI} = 1$ and $v = 0$. The analytical solution of the beam in bending (as in the preceding cases) gives:

$$\begin{aligned} u^a &= xz \\ w^a &= -\frac{x^2}{2} \end{aligned} \quad (1.56)$$

Or by using the iso-parametric coordinates

$$\begin{aligned} u^a &= \xi(\zeta - \alpha\zeta^2) \\ w^a &= -\frac{1}{2}\xi^2(1 - \alpha\zeta)^2 \end{aligned} \quad (1.57)$$

The displacements in terms of finite elements are written by (see McNeal [MAC 93])

$$\begin{aligned} u^{ef} &= \Lambda(\xi\zeta - \alpha\xi) \\ w^{ef} &= -\frac{1}{2}\Lambda^2(1 - \alpha\zeta + \alpha^2) \end{aligned} \quad (1.58)$$

And the corresponding strains are written as follows

$$\begin{aligned} \epsilon_x^{ef} &= \frac{(\zeta - \alpha)}{(1 - \alpha\zeta)} \\ \epsilon_z^{ef} &= \Lambda^2\alpha \\ 2\gamma_{xz}^{ef} &= \Lambda\xi \left[1 + \frac{\alpha(\zeta - \alpha)}{(1 - \alpha\zeta)} \right] \end{aligned} \quad (1.59)$$

While the analytical solutions are given as follows

$$\begin{aligned}\epsilon_x^a &= \zeta \\ \epsilon_z^a &= 0 \\ 2\gamma_{xz}^a &= 0\end{aligned}\tag{1.60}$$

From equations (1.59) and (1.60), it is visible that the finite element solution is different from the analytical solution when $\alpha \neq 0$. These parasitic terms that appear when α becomes large cause the trapezoidal locking. In conclusion, note that trapezoidal locking occurs when 2D or 3D finite elements with oblique edges (with respect to the mean line or surface) used to model the behavior of curved structures.

One way to avoid the locking problems of low-order 3D finite elements would be to use them in a very refined mesh. Another natural strategy to overcome these locking phenomenon would be to use 3D finite elements with higher degrees of interpolation. However, the latter is less attractive due to high computational cost, especially in non-linear applications. And both methods are of high computational cost. It is then preferable to use first order interpolations and to find ways to reduce locking phenomena. Shear locking of solid element can be alleviated by using the assumed natural strain (ANS) method [HUG 81b, WEM 82, DVO 84, DVO 95b, DVO 95a, KLI 97, HAU 98, HAU 01, CAR 05, CAR 08, SCH 09]. Also the ANS method applied to the transverse normal strain allows to eliminate trapezoidal locking as well, see Bischoff and Ramm [BIS 97] or Betsch and Stein [BET 96]. To avoid Poisson thickness locking additional terms with linear distribution in thickness direction is introduced for the transverse normal strain, see [SIM 92a, SOU 06, REE 07] to name just a few. This assures the normal stress to vary linearly along the thickness in bending situations. The reduced integration method allows to eliminate volumetric locking and some membrane locking. One can see the remarkable work of Belytschko and co-authors [BEL 83, BEL 84, BEL 93], Reese and co-author [REE 00, REE 07] as well as Combescure and co-authors [LEG 03, ABE 09]. 3D standard element with all this methods incorporated in their formulations are called solid-shell elements. They are naturally design to overcome the disadvantages of 3D low-order standard element while correctly reproducing the behavior of shell like structures.

1.5 The rise of solid-shell elements

Because of their simplicity, first order finite elements are widely used for structural modeling. However, as we have seen previously, when used to model the behavior of thin structures, these elements exhibit severe locking problems like **transverse shear locking, membrane, Poisson thickness, trapezoidal or volumetric locking**. Parasitic shear or membrane modes become predominant and engulf the bending modes resulting in membrane or shear locking. When incompressible or quasi-incompressible materials are used, volumetric locking occurs. Several methods have been proposed in recent years to construct 3D finite elements capable of modeling the behavior of shell like structures :

they are called solid-shell finite elements. The best known and used methods to alleviate locking phenomenon are the assumed strain methods (EAS, ANS), and the selective or reduced integration methods. In this section details about these methods are given as they are useful to understand the solid-shell elements that are going to be presented in the next chapter.

1.5.1 Reduced integration (RI) method

A very attractive method for removing locking phenomenon is the reduced integration technique. It consists of taking a number of Gauss integration points aligned in the thickness direction of the element and centred on the mid-plane. The main advantage is the reduction of calculation time while removing membrane and volumetric locking.

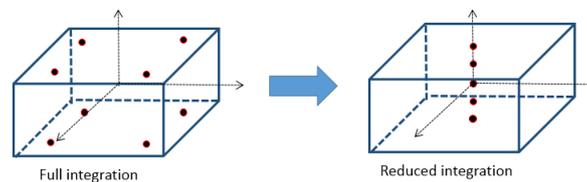


Figure 1.9: Full integration scheme to reduced integration scheme

However, reduced integration elements may generate zero energy modes associated with non-zero deformations. This causes strong distortions, creating areas where no stress accompanies the observed deformation of the element. This is known as the Hourglass (or hourglass) phenomenon. Several stabilization methods have been introduced by Belytschko and co-authors [BEL 83, BEL 84] or Reese and co-authors [REE 00, REE 07] in order to prevent the appearance of these Hourglass patterns. The selective integration technique is also often used and consists in cleverly combining the full and reduced integration techniques (e.g. full integration for bending stiffness matrix and reduced integration for shear stiffness matrix).

1.5.2 Incompatible method

Incompatible method, also called assumed displacement method, is derived from the potential energy variational principle. We here present the incompatible method as a reference to motivate for the EAS method presented in the next section. Hence, the incompatible method is briefly introduced and only considered in linear elastic theory. The standard solid elements pose the following difficulties: locking phenomena for bending and incompressible problems. By adding incompatible displacements to 3-D isoparametric nodes, the mentioned difficulties are canceled.

Consider a continuum body occupying a volume Ω_0 in a space of boundary surface $\partial\Omega_0$. Let's assume that the body force and the tension force are conservative and the object is in static state. Under the theory of linear elasticity, the principle of minimum

potential energy can be stated as :

$$\pi(\mathbf{u}) = \frac{1}{2} \int_{\Omega_0} (\boldsymbol{\varepsilon}^{com}(\mathbf{u}))^T \mathbf{D} \boldsymbol{\varepsilon}^{com}(\mathbf{u}) d\Omega_0 - \pi_{ext}(\mathbf{u}) \quad (1.61)$$

where $\boldsymbol{\varepsilon}^{com}$ is the compatible small deformation. When the solid continuum is discretized into a finite number of elements, the above variational is rewritten in the form as

$$\pi(\mathbf{u}) = \sum_{e=1}^{Nbr} \left\{ \frac{1}{2} \int_{\Omega_0} (\mathbf{B}\mathbf{u})_e^T \mathbf{D} (\mathbf{B}\mathbf{u})_e d\Omega_0 - \pi_{ext}^e(\mathbf{u}) \right\} \quad (1.62)$$

where Nbr is the number of elements. In the finite element formulation the element displacements $(\mathbf{u})_e$ are interpolated in terms of nodal displacements that may be at both boundary nodes (Serendipity elements) and internal nodes (Lagrange elements). Elements can also be formulated by adding to the original element displacements $(\mathbf{u})_e$, which are in terms of nodal displacements $(\mathbf{U})_e$ higher-order displacements $\boldsymbol{\alpha}$, which are not expressed in terms of nodal displacements of the boundary nodes. For example, the displacements u and v for the standard four-node quadrilateral element are based on bilinear interpolation functions, see subsection (1.4.2). They are incomplete in quadratic terms (see equation 1.40). Improvement of the performance of a four nodes element can be made by adding terms such that the displacements are complete in quadratic terms. Wilson et al [WIL 73] suggested the addition of incompatible displacements that vanish at all corner nodes. In these cases the element strain $\boldsymbol{\varepsilon} = (\mathbf{B}\mathbf{u})_e$ can be expressed as follows

$$\boldsymbol{\varepsilon} = [\mathbf{B}^u \quad \mathbf{B}^\alpha] \begin{bmatrix} \mathbf{U}_e \\ \boldsymbol{\alpha}_e \end{bmatrix} \quad (1.63)$$

where \mathbf{B}^u and \mathbf{B}^α are respectively the compatible and incompatible strain matrices. $\mathbf{U}_e = [U_1, V_1, W_1, \dots, U_{NbrNode}, V_{NbrNode}, W_{NbrNode}]$ and $\boldsymbol{\alpha}_e = [\alpha_1, \dots, \alpha_{NbrInmode}]$ are respectively nodal displacement vector and incompatible mode vector element, with $NbrNode$ the number of nodes per element, $NbrInmode$ the number of incompatible modes per element. The element stiffness matrix is then written as follows

$$\mathbf{K}_e = \begin{bmatrix} (\mathbf{B}^u)^T \mathbf{D} \mathbf{B}^u & (\mathbf{B}^u)^T \mathbf{D} \mathbf{B}^\alpha \\ (\mathbf{B}^\alpha)^T \mathbf{D} \mathbf{B}^u & (\mathbf{B}^\alpha)^T \mathbf{D} \mathbf{B}^\alpha \end{bmatrix} \quad (1.64)$$

where $(\mathbf{B}^u)^T \mathbf{D} \mathbf{B}^u$ is the standard stiffness matrix of the element, $(\mathbf{B}^\alpha)^T \mathbf{D} \mathbf{B}^\alpha$ the incompatible stiffness matrix of the element and $(\mathbf{B}^u)^T \mathbf{D} \mathbf{B}^\alpha$ the incompatible-standard stiffness matrix of the element. In the same way we can define the standard nodal force vector \mathbf{F}_{ext}^e and the incompatible nodal force vector \mathbf{F}_α^e as follows

$$\begin{aligned} \mathbf{F}_{ext}^e &= \int_{\partial\Omega_0} \mathbf{N}_u^T \cdot \mathbf{f} d\partial\Omega_0 + \int_{\Omega_0} \mathbf{N}_u^T \cdot \mathbf{b} d\Omega_0 \\ \mathbf{F}_\alpha^e &= \int_{\partial\Omega_0} \mathbf{N}_\alpha^T \cdot \mathbf{f} d\partial\Omega_0 + \int_{\Omega_0} \mathbf{N}_\alpha^T \cdot \mathbf{b} d\Omega_0 \end{aligned} \quad (1.65)$$

The incompatible vector α^e consists of internal variables. Hence it can be, in some cases, condensed out of the variational formulation by setting $\frac{\partial \pi}{\partial \alpha} = 0$ in order to get α_e in function of \mathbf{U}_e as

$$\alpha_e = -[(\mathbf{B}^\alpha)^T \mathbf{D}\mathbf{B}^\alpha]^{-1}((\mathbf{B}^\alpha)^T \mathbf{D}\mathbf{B}^u \cdot \mathbf{U}_e - \mathbf{f}_\alpha^e) \quad (1.66)$$

Doing so, the equivalent element stiffness matrix and the equivalent nodal force are written as

$$\mathbf{k}_e = (\mathbf{B}^\alpha)^T \mathbf{D}\mathbf{B}^\alpha - [(\mathbf{B}^\alpha)^T \mathbf{D}\mathbf{B}^u][(\mathbf{B}^\alpha)^T \mathbf{D}\mathbf{B}^\alpha]^{-1}[(\mathbf{B}^\alpha)^T \mathbf{D}\mathbf{B}^u]^T \quad (1.67)$$

$$\mathbf{f}_e = \mathbf{f}_{ext}^e - [(\mathbf{B}^\alpha)^T \mathbf{D}\mathbf{B}^u][(\mathbf{B}^\alpha)^T \mathbf{D}\mathbf{B}^\alpha]^{-1} \mathbf{f}_\alpha^e \quad (1.68)$$

Finally, assembling \mathbf{k}_e into a global matrix \mathbf{K} and \mathbf{f}_e into a global nodal vector \mathbf{F} , the total potential energy is

$$\pi(\mathbf{u}) = \frac{1}{2} \mathbf{U}^T \cdot \mathbf{K} \cdot \mathbf{U} - \mathbf{U}^T \cdot \mathbf{F} \quad (1.69)$$

Where \mathbf{U} is the global nodal displacement vector, $\mathbf{K} = \sum_{e=1}^{Nbr} \mathbf{k}_e$ and $\mathbf{F} = \sum_{e=1}^{Nbr} \mathbf{f}_e$. The variation of $\pi(\mathbf{u})$ with respect to the global displacement vector \mathbf{U} imposed equal to zero, gives the equation for displacement solution :

$$\mathbf{K} \cdot \mathbf{U} = \mathbf{F} \quad (1.70)$$

After solving the system (1.70), all the nodal displacements are known, and other variables like strains and stresses are obtained as in the standard manner. Note that for the seven and nine nodes solid-shell elements presented in the next chapter, the condensation technique mentioned in this section is not followed due to the pinching stress enhancement that will be detailed in chapter 2.

In the next section, the EAS method, based on the three field Hu-Washizu variational principle, is presented. The EAS method is considered as a generalized approach of the incompatible method as pointed by Simo and Rifai [SIM 90b].

1.5.3 The Enhanced Assumed strain (EAS) method

Due to their efficiency and simple geometry, low-order solid elements are often preferred in structural mechanics. But, as mentioned above, the low-order standard displacement elements exhibits, in many cases, severe stiffening effects known as locking. Shear locking occurs when simulating thin-walled structures by the low-order standard displacement elements, where pure bending modes are spoiled by parasitic shear strains. Membrane locking is encountered in high aspect ratio elements when bending modes cannot be separated from membrane strains and, thus, not allowing the verification of pure inextensional modes. For incompressible or nearly incompressible conditions, volumetric locking may also occur; in this case, deviatoric modes always come along undesirably with volumetric strains. The class of EAS elements presented below allows the systematic development of low-order elements with enhanced accuracy for coarse

meshes. This method whose formalism is based on the Hu-Washizu three-field formulation was initially proposed by Simo and Rifai [SIM 90a, SIM 90b] in small deformations, then extended to large deformations by Simo and co-authors [SIM 92b, SIM 93] before being applied to thermomechanical problems by Adam and Ponthot [ADA 05]. The idea of the EAS technique is to enhance the usual deformation of a finite element by enriching it with a deformation intelligently chosen to eliminate locking phenomena. Finite elements with the EAS approach have been applied to simulate geometrically and materially nonlinear problems due to the fact that they perform well in severe situations as the nearly incompressible limit and pure bending situations, when coupled with the reduced integration technique.

1.5.3.1 Three fields variational formulation

The EAS method aims to enhance elements displacement gradient and further improves volumetric and Poisson thickness locking. The Hu-Washizu principle is generally the starting point. As being proposed by Simo and Rifai [SIM 90b], the displacement field $\mathbf{u} \in \mathcal{V}$, the conjugated Green-Lagrange strain tensor $\mathbf{E} \in \mathcal{E}$ and the admissible second Piola-Kirchhoff stress $\mathbf{S} \in \mathcal{S}$ are treated as independent variables. \mathcal{V} , \mathcal{E} and \mathcal{S} being respectively the space of admissible displacement, strain and stress. The Hu-Washizu principle is derived as

$$\pi^{HW}(\mathbf{u}, \mathbf{E}, \mathbf{S}) = \int_{\Omega_0} W(\mathbf{E}) d\Omega_0 + \int_{\Omega_0} \mathbf{S} : \left[\frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}) - \mathbf{E} \right] d\Omega_0 - \pi_{\text{ext}}(\mathbf{u}) \quad (1.71)$$

Where (W) is the strain energy, \mathbf{F} the compatible deformation gradient depending on the displacement field \mathbf{u} and defined in section 1.2 and \mathbf{I} the metric tensor. According to the EAS method the independent strain field is given

$$\mathbf{E} = \mathbf{E}^{mod} = \mathbf{E}^u + \mathbf{E}^{eas} \quad \text{with} \quad \mathbf{E}^u = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}) \quad (1.72)$$

Where we call \mathbf{E}^u the compatible strain field and \mathbf{E}^{eas} the enhanced assumed strain field. Let's call \mathbf{E}^{mod} the modified strain field. Introducing the modified strain \mathbf{E}^{mod} into equation (1.71), the variational basis for the EAS method is written

$$\pi^{HW}(\mathbf{u}, \mathbf{E}^{mod}, \mathbf{S}) = \int_{\Omega_0} W(\mathbf{E}) d\Omega_0 + \int_{\Omega_0} \mathbf{S} : [\mathbf{E}^u - \mathbf{E}^{mod}] d\Omega_0 - \pi_{\text{ext}}(\mathbf{u}) \quad (1.73)$$

Once the modified strain tensor \mathbf{E}^{mod} is obtained, the gradient deformation tensor \mathbf{F}^{mod} can be consistently derived through the use of a polar decomposition, (see section 1.5.5). In general, as proposed by Simo [SIM 90a], the approximation of \mathbf{E}^{eas} and \mathbf{S} are chosen to satisfy the following orthogonality condition

$$\int_{\Omega_0} \mathbf{S} : \mathbf{E}^{eas} d\Omega_0 = 0 \quad (1.74)$$

Applying the orthogonality condition, the three field variational is reduced to a two field variational principle as follows :

$$\pi^{HW}(\mathbf{u}, \mathbf{E}^{eas}) = \int_{\Omega_0} W(\mathbf{E}^u + \mathbf{E}^{eas}) d\Omega_0 - \pi_{\text{ext}}(\mathbf{u}) \quad (1.75)$$

which immediately gives the first variation

$$\delta\pi^{HW}(\mathbf{u}, \mathbf{E}^{eas}) = \int_{\Omega_0} (\delta\mathbf{E}^u + \delta\mathbf{E}^{eas}) : \mathbf{S}^{mod} d\Omega_0 - \delta\pi_{\text{ext}}(\mathbf{u}) \quad (1.76)$$

Where the modified second Piola-Kirchhoff stress \mathbf{S}^{mod} is given by

$$\mathbf{S}^{mod} = \frac{\partial W}{\partial \mathbf{E}^{mod}} \quad (1.77)$$

Following the finite element method detailed in section 1.2, The variation $\delta\mathbf{E}^u$ and increment $\Delta\mathbf{E}^u$ of the compatible Green-Lagrange strain at the element level are interpolated as follows

$$\delta\mathbf{E}^u = \mathbf{B}^u \delta\mathbf{U} \quad \text{and} \quad \Delta\mathbf{E}^u = \mathbf{B}^u \Delta\mathbf{U} \quad (1.78)$$

where \mathbf{B}^u is the compatible standard strain-displacement matrix and is function of the compatible displacement \mathbf{u} . The enhanced strains is function of a enhanced strain interpolation matrix Γ and internal strain parameters α . The variation and increment of the enhanced strains are

$$\delta\mathbf{E}^{eas} = \Gamma \delta\alpha \quad \text{and} \quad \Delta\mathbf{E}^{eas} = \Gamma \Delta\alpha \quad (1.79)$$

Note that there are many ways to define the enhancement interpolation and the number of internal parameters (see [AND 93, KLI 97, TAY 76] among many others). The EAS method has two major roles: to reduce the volumetric locking (in the constant field) and also the Poisson locking deficiency. Recently, in the works of Alves de Sousa and co-authors [SOU 05, SOU 06], it has been shown that only a single enhancing parameter was enough to considerably reduce those locking pathologies on the constant strain matrices, thus giving rise to an element that can retain its computational efficiency, offering an almost perfect combination between EAS and reduced integration schemes. Therefore in the following sections and chapters we will only consider the EAS method with only one parameter.

1.5.3.2 Linearization of discrete weak form

Using equation (1.75), we formulate the linearization of the weak form $\delta\pi^{HW}(\mathbf{u}, \mathbf{E}^{eas})$ at the element level by employing the truncated Taylor series about the $(k+1)^{th}$ iteration

$$\begin{aligned} \delta\pi(\mathbf{u}_{k+1}, \mathbf{E}_{k+1}^{eas}) &= \delta\pi(\mathbf{u}_k, \mathbf{E}_k^{eas}) + \frac{\partial \delta\pi^e}{\partial (\mathbf{u}_k, \mathbf{E}_k^{eas})} \Big|_{(\mathbf{u}_k, \mathbf{E}_k^{eas})} \cdot (\Delta\mathbf{u}_k, \Delta\mathbf{E}_k^{eas}) \\ &= \delta\pi(\mathbf{u}_k, \mathbf{E}_k^{eas}) + D(\delta\pi^e) \Big|_{(\mathbf{u}_k, \mathbf{E}_k^{eas})} \cdot (\Delta\mathbf{u}_k, \Delta\mathbf{E}_k^{eas}) \end{aligned} \quad (1.80)$$

Where $D(\cdot)$ is the Gateaux derivative operator, see Simo and Hughes [SIM 98]. To alleviate the notation, the right subscript “k” designating the iterative index is omitted. To find $(\Delta \mathbf{u}, \Delta \mathbf{E}^{eas})$ we let the right hand side of equation (1.23) be nil, which gives

$$-\delta \pi_{int}^e - \delta \pi_{ext}^e = \frac{\partial \delta \pi^e}{\partial (\mathbf{U}^e, \alpha^e)} \Big|_{(\mathbf{U}, \alpha^e)} \cdot (\Delta \mathbf{U}^e, \Delta \alpha^e) \quad (1.81)$$

The internal virtual work is then written as follows

$$\begin{aligned} \delta \pi_{int}^e &= \int_{\Omega_0} (\mathbf{B} \delta \mathbf{U}^e)^T \mathbf{S}^{mod} d\Omega_0 + \int_{\Omega_0} (\Gamma \delta \alpha^e)^T \mathbf{S}^{mod} d\Omega_0 \\ &= \delta \mathbf{U}^T \mathbf{f}_{int}^e + \delta \alpha^T \mathbf{f}_{eas}^e \end{aligned} \quad (1.82)$$

With

$$\mathbf{f}_{int}^e = \int_{\Omega_0} \mathbf{B}^T \mathbf{S}^{mod} d\Omega_0 \quad \text{and} \quad \mathbf{f}_{eas}^e = \int_{\Omega_0} \Gamma^T \mathbf{S}^{mod} d\Omega_0 \quad (1.83)$$

Assuming that the external force is displacement-independent, the external virtual work is written as

$$\delta \pi_{ext}^e = -\delta \mathbf{U}^T \mathbf{f}_{ext}^e \quad (1.84)$$

where \mathbf{f}_{ext}^e is same as in section 1.2. Lets now derive the right hand side of equation (1.81)

$$\begin{aligned} D(\delta \pi^e) \cdot (\Delta \mathbf{u}, \Delta \mathbf{E}^{eas}) &= \frac{\partial (\delta \pi_{int}^e)}{\partial \mathbf{U}} \cdot \Delta \mathbf{U} + \frac{\partial (\delta \pi_{int}^e)}{\partial \alpha} \Delta \alpha \\ &= \delta \mathbf{U}^T \frac{\partial (\delta \mathbf{f}_{int}^e)}{\partial \mathbf{U}} \cdot \Delta \mathbf{U} + \delta \alpha^T \frac{\partial (\delta \mathbf{f}_{eas}^e)}{\partial \alpha} \Delta \alpha \\ &\quad + \delta \mathbf{U}^T \frac{\partial (\delta \mathbf{f}_{int}^e)}{\partial \alpha} \cdot \Delta \alpha + \delta \alpha^T \frac{\partial (\delta \mathbf{f}_{eas}^e)}{\partial \alpha} \Delta \alpha \\ &= [\delta \mathbf{U}^T \mathbf{k}_{uu}^e + \delta \alpha^T \mathbf{k}_{\alpha u}^e] \cdot \Delta \mathbf{U} + [\delta \mathbf{U}^T \mathbf{k}_{u\alpha}^e + \delta \alpha^T \mathbf{k}_{\alpha\alpha}^e] \cdot \Delta \alpha \end{aligned} \quad (1.85)$$

Where \mathbf{k}_{uu}^e is the compatible-standard stiffness matrix same as \mathbf{K}_T in section 1.2, $\mathbf{k}_{\alpha\alpha}^e$ the incompatible stiffness matrix and $\mathbf{k}_{\alpha u}^e$ the standard-incompatible stiffness matrix. The constitutive tensor in the physical space is now expressed through the following stress-strain relationship

$$\mathbf{D} = [D^{ijkl}] = \left[\frac{\partial \mathbf{S}^{(ij)mod}}{\partial \mathbf{E}_{kl}^{mod}} \right] \quad (1.86)$$

The standard stiffness matrix of the element includes the material part \mathbf{k}_{mat}^e and geometrical part \mathbf{k}_{geo}^e as

$$\mathbf{k}_{uu}^e = \frac{\partial \mathbf{f}_{int}^e}{\partial \mathbf{U}} = \mathbf{k}_{mat}^e + \mathbf{k}_{geo}^e = \int_{\Omega_0} \mathbf{B}^{uT} \mathbf{D} \mathbf{B}^u d\Omega + \int_{\Omega_0} \mathcal{B} \mathbf{S}^{mod} d\Omega \quad (1.87)$$

where, for a geometrical nonlinear theory the strain-displacement matrix \mathbf{B} , is a function of the displacements \mathbf{u} as in section 1.2. Furthermore, the strain-displacement matrix

contains the derivatives of the shape functions with respect to the global coordinates \mathbf{X} in the reference configuration. Components of the geometrical part \mathbf{k}_{geo}^e is defined as

$$\mathbf{k}_{geo}^e = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{11} & \dots & \mathbf{H}_{1no} \\ \mathbf{H}_{21} & \mathbf{H}_{21} & \dots & \mathbf{H}_{2no} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{no1} & \mathbf{H}_{no1} & \dots & \mathbf{H}_{nono} \end{bmatrix} \quad (1.88)$$

With no still being the number of nodes of the finite element considered. \mathbf{H}_{IJ} ($I, J = 1, 2, \dots, no$) is defined for a node combination I and J as $\mathbf{H}_{IJ} = H_{IJ}\mathbf{I}_3$; where \mathbf{I}_3 is the unit matrix of dimension 3 by 3 and

$$H_{IJ} = \int_{\Omega_0} N_{I,K} S_{KL} N_{J,L} d\Omega_0; \quad K, L = 1, 2, 3 \quad (1.89)$$

The enhanced-compatible stiffness matrix of the element is

$$\mathbf{k}_{\alpha u}^e = (\mathbf{k}_{\alpha u}^e)^T = \left(\frac{\partial \mathbf{f}_{int}^e}{\partial \alpha} \right)^T = \int_{\Omega_0} \Gamma^T \mathbf{D} \mathbf{B}^u d\Omega_0 \quad (1.90)$$

while the enhanced stiffness matrix of the element is as follows

$$\mathbf{k}_{\alpha\alpha}^e = \left(\frac{\partial \mathbf{f}_{eas}^e}{\partial \alpha} \right) = \int_{\Omega_0} \Gamma^T \mathbf{C} \Gamma d\Omega_0 \quad (1.91)$$

The combination of (1.83),(1.87),(1.90) and (1.91) gives the discrete linearized system of equations to solve for the increment $\Delta \mathbf{U}$ and $\Delta \alpha$ (see [KLI 97])

$$\begin{bmatrix} \mathbf{k}_{uu}^e & \mathbf{k}_{u\alpha}^e \\ \mathbf{k}_{\alpha u}^e & \mathbf{k}_{\alpha\alpha}^e \end{bmatrix} \begin{bmatrix} \Delta \mathbf{U} \\ \Delta \alpha \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{ext} - \mathbf{f}_{int} \\ -\mathbf{f}_{eas} \end{bmatrix} \quad (1.92)$$

The equation (1.92) is the elementary equilibrium equation of a solid-shell finite element operating with the EAS method. The latter is very effective in eliminating Poisson thickness and volumetric locking, see Andelfinger and Ramm [AND 93], Wriggers and Korelc [WRI 96c], Alves de Sousa and al [SOU 03]. However, although interesting in eliminating locking phenomena, solid-shell elements operating with the EAS method can be computationally time consuming due to the additional degrees of freedom. However, according to Alves de souza and co-authors [SOU 05, SOU 06], it is possible to reduce volumetric locking with just one EAS parameter. It has also been shown that there are signs of instability under certain loads such as in compression, see Wriggers and Reese [WRI 96b], De Sousa Neto et al [SOU 95]. Furthermore, EAS elements are effective in eliminating volumetric and Poisson thickness locking but not in completely eliminating transverse shear locking, especially for thin structures under bending situations. This is the reason why this method is very frequently associated with the assumed natural strain (ANS) method.

1.5.4 The Assumed Natural Strain (ANS) method

The EAS method presented above does not give satisfactory results in eliminating transverse shear locking phenomena, especially when modelling a thin structure subjected to bending loading. The method that is then widely used is the ANS or Assumed Natural Strain method, which has proven to be more effective in eliminating transverse shear lockings [BIS 97]. This new Assumed Natural Strain technique consists of replacing the transverse shear deformation with an assumed shear strain obtained by interpolation in the natural base of the element, hence the name Assumed Natural Strain. The most commonly used plate element currently used in design codes works with this technique to overcome transverse shear locking [DVO 84]. In addition to facilitating the management of shear locking, interpolation in the natural base gives elements working with this method the ability to be accurate even with a more or less distorted mesh. Therefore, it is necessary to define a convected description, which naturally preserves the objectivity (in the convected description, the material base vectors reflect the geometrical and kinematic aspects, hence, the corresponding components are indifferent with respect to their material base vectors). To this end, let's denote the position vectors of the reference configuration Ω_0 and the current configuration Ω_t in the local coordinates system by \mathbf{X} and \mathbf{x} , respectively. the convected basis vector \mathbf{G}_i and its components G_{ij} in the initial basis system are defined by

$$\mathbf{G}_i = \frac{\partial \mathbf{X}}{\partial \xi^i} \quad ; \quad G_{ij} = \mathbf{G}_i \cdot \mathbf{G}_j \quad ; \quad i = 1, 2, 3 \quad (1.93)$$

While the contravariant vector \mathbf{G}^j and its components G^{ij} are defined as follow

$$\mathbf{G}_i \cdot \mathbf{G}^j = \delta_j^i \quad ; \quad \mathbf{G}^j = G^{ij} \mathbf{G}_i \quad ; \quad i, j = 1, 2, 3 \quad (1.94)$$

In the same manner the convected basis vector \mathbf{g}_i and its components g^{ij} in the current basis system are defined as follows

$$\mathbf{g}_i = \frac{\partial \mathbf{x}}{\partial \xi^i} = \mathbf{G}_i + \frac{\partial \mathbf{u}}{\partial \xi^i} \quad ; \quad g_{ij} = \mathbf{g}_i \cdot \mathbf{g}_j \quad ; \quad i = 1, 2, 3 \quad (1.95)$$

While the contravariant vector \mathbf{g}^j and its components g^{ij} are defined as following

$$\mathbf{g}_i \cdot \mathbf{g}^j = \delta_j^i \quad ; \quad \mathbf{g}^j = g^{ij} \mathbf{g}_i \quad ; \quad i, j = 1, 2, 3 \quad (1.96)$$

The deformation gradient can then be written by combining the equations (1.93) and (1.95)

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \frac{\partial x^i}{\partial \xi^k} \mathbf{e}_i \otimes \frac{\partial \xi^k}{\partial X^j} \mathbf{e}_j = \mathbf{g}_i \otimes \mathbf{G}^i \quad (1.97)$$

The Green-Lagrange strain tensor defined in section 1.2 becomes

$$\begin{aligned}
 \mathbf{E} &= \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{I}) = E_{ij} \mathbf{e}_i \otimes \mathbf{e}_j \\
 &= \frac{1}{2}(g_{ij} - G_{ij}) \mathbf{G}^i \otimes \mathbf{G}^j = \hat{E}_{ij} \mathbf{G}^i \otimes \mathbf{G}^j \\
 &= \frac{1}{2} \left(\mathbf{G}_i \cdot \frac{\partial \mathbf{u}}{\partial \xi^j} + \frac{\partial \mathbf{u}}{\partial \xi^i} \cdot \mathbf{G}_j + \frac{\partial \mathbf{u}}{\partial \xi^j} \cdot \frac{\partial \mathbf{u}}{\partial \xi^i} \right) \mathbf{G}^i \otimes \mathbf{G}^j
 \end{aligned} \tag{1.98}$$

Now instead of using the standard computation of equation (1.98), which leads to shear locking, the transverse shear strains E_{13} and E_{23} are assumed, according to Dvorkin and Bathe [DVO 95a], to be interpolated through the use of certain sampling points as follows (using Voigt notations)

$$\mathbf{E} = \begin{bmatrix} E_{11} \\ E_{22} \\ E_{33} \\ 2E_{12} \\ 2E_{13} \\ 2E_{23} \end{bmatrix} = \mathbf{Q} \begin{bmatrix} 0.5(g_{11} - G_{11}) \\ 0.5(g_{22} - G_{22}) \\ 0.5(g_{33} - G_{33}) \\ (g_{12} - G_{12}) \\ \frac{1}{2}(1 - \eta)(g_{13} - G_{13})_{(A)} + \frac{1}{2}(1 + \eta)(g_{13} - G_{13})_{(C)} \\ \frac{1}{2}(1 - \xi)(g_{23} - G_{23})_{(D)} + \frac{1}{2}(1 + \xi)(g_{23} - G_{23})_{(B)} \end{bmatrix} \tag{1.99}$$

Where \mathbf{T} is a function of \mathbf{G}^i and convert the strain from the natural base to the initial base. These assumed strains can be implemented in the standard solid element in a straightforward manner. These assumptions allow the element to represent pure bending modes without any spurious shear effect. The modified shear strains lead to the new operator matrix \mathbf{B}^u in the natural space. However, the formulation can be slightly modified so that no explicit strain evaluation at the sampling points is necessary in the numerical calculation. The physical assumed strain-displacement matrix at node I of the eight-node solid element is

$$\mathbf{B}_I = \mathbf{Q} \begin{bmatrix} N_{I,\xi} \mathbf{g}_1^T \\ N_{I,\eta} \mathbf{g}_2^T \\ N_{I,\zeta} \mathbf{g}_3^T \\ N_{I,\xi} \mathbf{g}_2^T + N_{I,\eta} \mathbf{g}_1^T \\ \frac{1}{2}(1 - \eta)(N_{I,\xi} \mathbf{g}_3^T + N_{I,\zeta} \mathbf{g}_1^T)_{(A)} + \frac{1}{2}(1 + \eta)(N_{I,\xi} \mathbf{g}_3^T + N_{I,\zeta} \mathbf{g}_1^T)_{(C)} \\ \frac{1}{2}(1 - \xi)(N_{I,\eta} \mathbf{g}_3^T + N_{I,\zeta} \mathbf{g}_2^T)_{(D)} + \frac{1}{2}(1 + \xi)(N_{I,\eta} \mathbf{g}_3^T + N_{I,\zeta} \mathbf{g}_2^T)_{(B)} \end{bmatrix} \tag{1.100}$$

The ANS method has the advantage of being free of locking and is insensitive to small mesh distortions. The results also remain correct even when the mesh is relatively coarse

[PAR 86]. Initially the ANS method was a simple engineering intuition without a solid mathematical basis to support it. Formulated in the 1980s, the ANS method was quickly adopted by the research community: Hughes et Tezduyar [HUG 81b], Bathe et Dvorkin [DVO 84], Wempner et al [WEM 82], Hauptmann et al [HAU 98] to name just a few. A few years later, in two separate publications, Militello [MIL 90] provides a mathematical justification of the ANS technique based also on the three-field mixed formulation of Hu-Washizu. ANS elements are still widely used because of their simplicity but also because of their effectiveness in eliminating transverse shear locking.

1.5.5 Modified deformation gradient

The ANS and EAS methods modify the initial Green Lagrange strain \mathbf{E} and gives a new Green Lagrange strain \mathbf{E}^{mod} . The associated modified deformation gradient \mathbf{F}^{mod} is required if the element is implemented in a source code based on the updated Lagrange formulation or when a material algorithm for large elasto-plastic strains is needed. The original deformation gradient can be split into right-stretch tensor \mathbf{U} and rotation tensor \mathbf{R} as

$$\mathbf{F} = \mathbf{R}\mathbf{U} \quad (1.101)$$

Introducing \mathbf{U} and \mathbf{R} into the formulation for calculating Green-Lagrange strain tensor we have:

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{I}) \quad (1.102)$$

$$= \frac{1}{2}(\mathbf{U}^2 - \mathbf{I}) \quad (1.103)$$

Clearly, Green-Lagrange strain tensor depends only on the right-stretch tensor \mathbf{U} . Thus, from the modified strain \mathbf{E}^{mod} we can derive the associated modified right-stretch tensor \mathbf{U}^{mod} . According to Hauptmann et al [HAU 01], the modified deformation gradient \mathbf{F}^{mod} is calculated as

$$\mathbf{F}^{mod} = \mathbf{R}\mathbf{U}^{mod} \quad (1.104)$$

One can see that the computation of \mathbf{F}^{mod} requires twice a polar decomposition. The first step consists in the calculation of rotation tensor \mathbf{R} in (1.101). The second step is the calculation of the modified right-stretch tensor \mathbf{U}^{mod} from \mathbf{E}^{mod} . These calculations will increase a little bit the computational cost of the algorithm when the deformation gradient \mathbf{F}^{mod} is required.

1.6 Conclusion

First-order solid elements suffer from many locking phenomena. When the materials are incompressible, the rigidity matrix tends towards infinity, generating very small deformations: this is volumetric locking. This can be remedied either by using reduced integration techniques with additional use of stabilization techniques, or by using the EAS

method by adding incompatible deformations to satisfy the incompressibility condition [AND 93]. However, the EAS method can be time consuming due to the additional degrees of freedom. That's why it is recommended to use only one EAS parameter, since it is enough to reduce volumetric lockings. The reduced integration technique, in the other hand, is very advantageous in terms of time calculations. However, it is necessary to ensure that zero energy modes are well stabilized. It should also be noted that these two techniques also alleviate membrane and shear lockings. To eliminate transverse shear lockings, e.g. for a thin structure in bending, the ANS assumed method is very suitable. It is a method that has been proposed specifically to deal with transverse shear lockings. It can also be used to eliminate the trapezoidal locking phenomenon with some interpolation of the deformation in the thickness direction: see the work of Bischoff and Ramm [BIS 97], Betch and Stein [BET 96]. In order to overcome Poisson thickness locking, the transverse normal strain must be enriched in order to have at least a linear distribution. Which is also satisfied by the EAS approach. Generally speaking, a solid element becomes a solid-shell if it is able to overcome all the locking difficulties encountered by a standard solid element in modeling a thin structure in a bending-dominant situation [WIL 73]. In the next chapter we present a hexahedral and prismatic solid-shell elements using the methods presented in this chapter.

Chapter 2

Nine and seven nodes solid-shell elements

Contents

2.1	Introduction	45
2.2	Variational formulation	46
2.3	Nine nodes solid-shell element (SB9)	47
2.3.1	Kinematics	47
2.3.2	Mapping between the contravariant and the Cartesian basis	49
2.3.3	Strain Field	51
2.3.4	Assumed natural strain	53
2.3.4.1	Assumed shear strain	53
2.3.4.2	Assumed normal strain	54
2.3.5	Stabilization procedure	54
2.3.5.1	Green Lagrange strain stabilization	54
2.3.5.2	Second Piola Kirchhoff stress stabilization	56
2.3.6	The ninth node	57
2.4	Seven nodes solid-shell element (SB7)	58
2.4.1	Local frame and objective stress rate	58
2.4.2	Strain tensor	61
2.4.3	In plane strain enhancement	62
2.4.3.1	DKT6 Shell Finite element	63
2.4.3.2	Linking the triangle to the prism	65
2.4.4	Assumed transverse shear strain	67

2.4.4.1	C^0 shell element	67
2.4.4.2	Linking the triangle to the prism	70
2.4.5	Pinching Strain and seventh node	72
2.4.6	Stabilization of twist mode	72
2.5	Equivalent generalized nodal pressure forces	73
2.6	Static equilibrium	74
2.6.1	For SB9 finite element	75
2.6.2	For SB7 finite element	76
2.7	Elements Inertia Matrices	76
2.8	Conclusion	77

2.1 Introduction

The combination of the RI, ANS and EAS method in a standard solid element gives good performing solid-shell using 3D constitutive law. In fact the EAS parameter is added to solve Poisson thickness locking and volumetric locking by transforming a linear normal strain into a quadratic normal strain. However the value of this parameter is so low that it does not improve the pinching stress that much. In a problem with a shell structure under pressure in one or both side, the solid-shell element with EAS parameter alone gives a normal stress that is almost constant (see next chapter), and does not verify the Neumann boundary conditions. To overcome this, Sansalone and co-authors [SAN 11] add a additional node to a quadrangular and triangular shell element. The latter node is added to create a quadratic interpolation of the transverse displacement and consequently enrich the element pinching strain. This way the element can accurately work with a three dimensional constitutive law without the common plane stress hypothesis. Poisson thickness locking is then naturally avoided (like the EAS). But not only that, the additional dof allows to enrich the element pinching stress (enhancement of normal stress) and further match the upper and lower boundaries condition of the shell. This new method goes beyond the EAS approach. Therefore, the aim of this chapter is to extend this methodology by using an additional node at the center of an eight-node (respectively six-nodes) brick elements in a specific solid-shell formulation. Bassa and co-authors [BAS 12] have proposed a similar approach. Nevertheless the studies have shown a low convergence performance of that formulation in a implicit code, specially in nonlinear problems, due to the three-parameters stabilization procedure. Furthermore, the shear strain adopted in the SB9γ25 was interpolated following the work of Dvorking and Bathe [BAT 85, BAT 86, DVO 95b]. The Reissner multiplicative function was used as interpolation through the thickness direction to ensure a static admissibility condition. Such interpolation is only valid in linear isotropic situations.

It is then essential to realize that in this thesis, approach proposed by Sansalone [SAN 11] is here applied to a hexahedral and prismatic solid-shell elements. The Hexahedral solid-shell element is inspired by the work of Bassa [BAS 12] (see appendix D) but with a perfectly new formulation. The first difference is this new formulation is written in total Lagrangian. Secondly, the stabilization procedure that used to depend on the user (the three-parameters stabilization) is now completely automatic and depends only on the evolution of the problem. And finally, the assumed natural strain (ANS) is changed in a way that should allow the element to perform with friction contact, which was not the case with the initial formulation of Bassa. We also apply the ANS method to the pinching strain in order to eliminate trapezoidal locking.

For the prismatic solid-shell element we almost implement the formulation proposed by Hu and Hamila [XIO 18] with little improvement, due the lack of time, in the stabilization coefficient and in the elimination of trapezoidal locking.

An additional node is then introduced with only one through-thickness translational dof with a robust stabilization procedure that performs well in implicit code. This gives a nine (SB9) and seven (SB7) nodes solid-shell elements. The first eight (respectively

six) nodes are those of a classical hexahedral (respectively prismatic) element and have three translation degrees of freedom each. The ninth (respectively seventh) node is a additional node that has only one translation degree of freedom, in the element-thickness direction. Those central nodes act as EAS parameters, but not only. As been shown by Voldoire [VOL 93], attempting to improve shell modeling using only kinematic assumptions is less effective than working on both kinematic and static assumptions. Thus from those additional dof, a new distribution is made on the applied pressure forces, generating a volume contribution. This allows to go beyond the EAS method, allowing to improve considerably the pinch stresses. These two elements work with three-dimensional behaviour laws and are under integrated (reduced integration). Some stabilization procedures are necessary to handle hourglass modes resulting from the reduced integration method. The stabilization technique used are following the work of Belytschko [BEL 93] and Schwartz [SCH 09]. The ANS method is also applied in both formulation to reduce transverse shear locking. We adopt the ANS technique with 4 tying points for the shear strain as well as the normal transverse strain for the SB9 element while the 2 points ANS method from [BOI 94] is applied for the SB7 element. These new elements have a wide range of applications, showing very good convergence, robustness and accuracy in nonlinear problems. These elements have been implemented into the quasi-static and dynamic implicit software code_aster [Ele 20] developed by EDF.

2.2 Variational formulation

The Hu-Washizu principle as presented in the subsection 1.5.3.1 is the starting point for the elements formulations. It is written as follows

$$\pi^{HW}(\mathbf{u}, \mathbf{E}, \mathbf{S}) = \int_{\Omega_0} W(\mathbf{E}) d\Omega_0 + \int_{\Omega_0} \mathbf{S} : \left[\frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}) - \mathbf{E} \right] d\Omega_0 - \pi_{\text{ext}}(\mathbf{u}) \quad (2.1)$$

Where (W) is the strain energy, \mathbf{F} the deformation gradient depending on the displacement field \mathbf{u} , \mathbf{I} the metric tensor and $\pi^{\text{ext}}(\mathbf{u})$ the external force power. As being advocated first by Andelfinger and Ramm [AND 93] and further by Bischoff and Ramm [BIS 97] for large deformation with (EAS) method, the approach proposed herein is also based on an enrichment of the Green-Lagrange strain tensor.

$$\mathbf{E} = \mathbf{E}^u + \mathbf{E}^{\text{eas}} \quad \text{with} \quad \mathbf{E}^u = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}) \quad (2.2)$$

This means that the eight (respectively six) vertices nodes displacement-dependent strain tensor \mathbf{E}^u of the element is enriched by an additional enhanced assumed strain $\mathbf{E}^{\text{eas}} \in \mathcal{E}^{\text{eas}}$ thanks to the ninth node (respectively the seventh). The variation of the above functional is obtained from the directional derivative and leads to

$$\begin{aligned} \delta\pi^{HW}(\mathbf{u}, \mathbf{E}^{\text{eas}}, \mathbf{S}) = & \int_{\Omega_0} \left[\frac{\partial W}{\partial \mathbf{E}} : \delta \mathbf{E}^H + \frac{\partial W}{\partial \mathbf{E}} : \delta \mathbf{E}^{\text{eas}} \right] d\Omega_0 - \delta\pi_{\text{ext}} \\ & - \int_{\Omega_0} \mathbf{S} : \delta \mathbf{E}^{\text{eas}} d\Omega_0 - \int_{\Omega_0} \delta \mathbf{S} : \mathbf{E}^{\text{eas}} d\Omega_0 \end{aligned} \quad (2.3)$$

As in many researches, the three-field functional is reduced to a two field functional as suggested by Simo and Rifai [SIM 90b], by choosing the interpolation such that \mathbf{S} and \mathbf{E}^{eas} become orthogonal. A slightly different approach is applied in this new formulation. If the element is not pinched, in a sense that there is no pressure applied above or below the element, the formulation goes with the orthogonality condition. Otherwise, if the element is pinched, let's say by an upper pressure P_u and a lower pressure P_l , the second Piola Kirchhoff stress \mathbf{S} is chosen such that the corresponding Cauchy stress be linearly dependent to the applied pressure

$$\sigma_{33} = -\frac{1}{2}(1 - \zeta)P_l - \frac{1}{2}(1 + \zeta)P_u \quad (2.4)$$

ζ being the element thickness parameter, in the covariant frame. This way, the element normal stress is statically enhanced so that a correct pinching stress is derived.

Assuming a nil body force, the Euler-Lagrange equations [BIS 97] associated with equation (2.3) are the standard equilibrium equation of the domain

$$\begin{aligned} \operatorname{div} \left(\mathbf{F} \frac{\partial W}{\partial \mathbf{E}} \right) &= 0 \\ \mathbf{S} - \frac{\partial W}{\partial \mathbf{E}} &= 0 \\ \mathbf{E}^{eas} &= 0 \end{aligned} \quad (2.5)$$

Although $\mathbf{E}^{eas} = 0$ for the continuum problem, in general $\mathbf{E}_h^{eas} \neq 0$, when we introduce finite element approximations. By denoting \mathcal{D} the domain occupied by the body, we can observe that the space of enhanced strain field is in $[L_2(\mathcal{D})]^6$ (see Simo [SIM 90b]). Hence, no inter-element continuity on the \mathbf{E}^{eas} needs to be enforced when constructing the finite element approximation. Note also that the enhanced strain interpolation \mathcal{E}_h^{eas} and the standard strain interpolation defined by $\nabla^S \mathcal{V}_h$ are independent in the sense that $\mathcal{E}_h^{eas} \cap \nabla^S \mathcal{V}_h = \emptyset$

2.3 Nine nodes solid-shell element (SB9)

2.3.1 Kinematics

The SB9 element has nine nodes, one in-plane integration point and the ability to accommodate several integration points along the thickness direction of the element. Figure (2.1) represents the element topology with the order of node numbering (related to the isoparametric coordinate system of the element, defined by the natural coordinates ξ , η , ζ). Also, in the same figure, the distribution of Gauss-Lobatto integration points is given. Compared with the other eight-node 'solid-shell' bricks, the presence of a supplementary node has two main aims. First getting a linear normal strain component which, along with a full three-dimensional constitutive strain-stress behavior, allows to achieve similar results in bending cases as those obtained with the usual plane stress

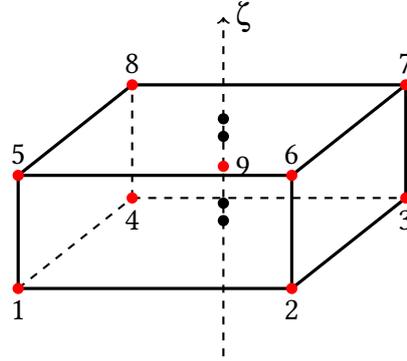


Figure 2.1: Location of the Gauss-Lobatto integration points for the SB9 element

state hypothesis. For that, the ninth node DOF plays the role of an extra parameter essential for a quadratic interpolation of the displacement in the thickness direction. The second advantage is that this DOF has a physical meaning and, for instance, a strength equivalent to a normal pressure can be prescribed to improve the normal stress when the shell structure is moderately thick. In this section we focus on the interpolation of the 8 vertices nodes, we will give more details about the ninth node in the subsection (2.3.6).

The location of nodes in the isoparametric coordinate system is given by the following isoparametric vectors:

$$\begin{aligned}
 a_1^I &= \{-1, +1, +1, -1, -1, +1, +1, -1\} \\
 a_2^I &= \{-1, -1, +1, +1, -1, -1, +1, +1\} \\
 a_3^I &= \{-1, -1, -1, -1, +1, +1, +1, +1\}
 \end{aligned} \tag{2.6}$$

where I means node number. Auxiliary vectors are defined from previous combinations of the isoparametric vectors:

$$\begin{aligned}
 h_1^I &= \{-1, -1, +1, -1, +1, -1, +1, -1\} \\
 h_2^I &= \{+1, +1, -1, -1, -1, -1, +1, +1\} \\
 h_3^I &= \{+1, -1, -1, +1, -1, +1, +1, -1\} \\
 h_4^I &= \{-1, +1, -1, +1, +1, -1, +1, -1\}
 \end{aligned} \tag{2.7}$$

These vectors form the basis for the interpolation field of the nodal displacements. The shape functions $N_I(\xi, \eta, \zeta)$ are obtained from a linear combination of the above-mentioned (equations 2.6 and 2.7) isoparametric vectors

$$N_I = \frac{1}{8} (1 + \xi a_1^I + \eta a_2^I + \zeta a_3^I + \xi \eta h_1^I + \eta \zeta h_2^I + \xi \zeta h_3^I + \xi \eta \zeta h_4^I) \tag{2.8}$$

The derivatives of the shape functions with respect to the isoparametric coordinates are

also spanned by the isoparametric vectors in the form :

$$\begin{aligned} N_{I,\xi} &= \frac{1}{8}(a_1^I + h_2^I \zeta + h_3^I \eta + h_4^I \eta \zeta) \\ N_{I,\eta} &= \frac{1}{8}(a_2^I + h_1^I \zeta + h_3^I \xi + h_4^I \xi \zeta) \\ N_{I,\zeta} &= \frac{1}{8}(a_3^I + h_1^I \eta + h_2^I \xi + h_4^I \xi \eta) \end{aligned} \quad (2.9)$$

One of the advantages of the SB9 element is the use of only translational degrees of freedom. Therefore the position of a point inside the element in the initial (\mathbf{X}) or the current (\mathbf{x}) state is obtained from the interpolation of the nodal translational degrees of freedom

$$\mathbf{X}(\xi, \eta, \zeta) = N_I(\xi, \eta, \zeta) \mathbf{X}_I \quad (2.10)$$

$$\mathbf{x}(\xi, \eta, \zeta) = N_I(\xi, \eta, \zeta) \mathbf{x}_I \quad (2.11)$$

where \mathbf{x}_I and \mathbf{X}_I are the nodal coordinates in the current and initial positions. The displacement field is also obtained from nodal values after a proper interpolation with shape functions

$$\mathbf{u}(\xi, \eta, \zeta) = N_I(\xi, \eta, \zeta) \mathbf{u}_I \quad (2.12)$$

where \mathbf{u}_I are the nodal displacement. Keep in mind that the effect of the ninth node will be discussed in a later section. The derivatives of displacements in the isoparametric coordinate system are interpolated from the shape functions derivatives as

$$\begin{aligned} \mathbf{u}_{,\xi}(\xi, \eta, \zeta) &= N_{I,\xi}(\xi, \eta, \zeta) \mathbf{u}_I \\ \mathbf{u}_{,\eta}(\xi, \eta, \zeta) &= N_{I,\eta}(\xi, \eta, \zeta) \mathbf{u}_I \\ \mathbf{u}_{,\zeta}(\xi, \eta, \zeta) &= N_{I,\zeta}(\xi, \eta, \zeta) \mathbf{u}_I \end{aligned} \quad (2.13)$$

2.3.2 Mapping between the contravariant and the Cartesian basis

From equation (2.9) the Jacobian matrices in the initial and the current configurations can be written as follows :

$$\mathbf{J} = \frac{\partial \mathbf{X}}{\partial \xi} \quad (2.14)$$

$$\mathbf{j} = \frac{\partial \mathbf{x}}{\partial \xi} = \frac{\partial \mathbf{X}}{\partial \xi} + \frac{\partial \mathbf{U}}{\partial \xi} = \mathbf{J} + \mathbf{D} \quad (2.15)$$

where $\xi = \{\xi, \eta, \zeta\}$ is the vector of natural coordinates and $\mathbf{D} = \frac{\partial \mathbf{U}}{\partial \xi}$ represents the derivative of the displacement vector \mathbf{U} with respect to the vector of natural coordinates ξ . \mathbf{J} and \mathbf{D} can be decomposed into constant, linear and bi-linear components depending on $\xi, \eta, \zeta, \xi\eta, \zeta\eta, \zeta\xi$ as follows

$$\mathbf{J} = \mathbf{J}^0 + \xi \mathbf{J}^\xi + \eta \mathbf{J}^\eta + \zeta \mathbf{J}^\zeta + \xi \eta \mathbf{J}^{\xi\eta} + \eta \zeta \mathbf{J}^{\eta\zeta} + \xi \zeta \mathbf{J}^{\xi\zeta} \quad (2.16)$$

$$\mathbf{D} = \mathbf{D}^0 + \xi \mathbf{D}^\xi + \eta \mathbf{D}^\eta + \zeta \mathbf{D}^\zeta + \xi \eta \mathbf{D}^{\xi\eta} + \eta \zeta \mathbf{D}^{\eta\zeta} + \xi \zeta \mathbf{D}^{\xi\zeta} \quad (2.17)$$

This procedure is required for the construction of the hourglass strain field that stabilizes the SB9 element. The components of equation (2.16 and 2.17) are given as follows, considering the Einstein convention for the repeated superscript

$$\mathbf{J}^0 = \frac{1}{8} \begin{bmatrix} a_1^I X_1^I & a_1^I X_2^I & a_1^I X_3^I \\ a_2^I X_1^I & a_2^I X_2^I & a_2^I X_3^I \\ a_3^I X_1^I & a_3^I X_2^I & a_3^I X_3^I \end{bmatrix} \quad \mathbf{D}^0 = \frac{1}{8} \begin{bmatrix} a_1^I U_1^I & a_1^I U_2^I & a_1^I U_3^I \\ a_2^I U_1^I & a_2^I U_2^I & a_2^I U_3^I \\ a_3^I U_1^I & a_3^I U_2^I & a_3^I U_3^I \end{bmatrix} \quad (2.18)$$

$$\mathbf{J}^\xi = \frac{1}{8} \begin{bmatrix} 0 & 0 & 0 \\ h_1^I X_1^I & h_1^I X_2^I & h_1^I X_3^I \\ h_3^I X_1^I & h_3^I X_2^I & h_3^I X_3^I \end{bmatrix} \quad \mathbf{D}^\xi = \frac{1}{8} \begin{bmatrix} 0 & 0 & 0 \\ h_1^I U_1^I & h_1^I U_2^I & h_1^I U_3^I \\ h_3^I U_1^I & h_3^I U_2^I & h_3^I U_3^I \end{bmatrix} \quad (2.19)$$

$$\mathbf{J}^\eta = \frac{1}{8} \begin{bmatrix} h_1^I X_1^I & h_1^I X_2^I & h_1^I X_3^I \\ 0 & 0 & 0 \\ h_2^I X_1^I & h_2^I X_2^I & h_2^I X_3^I \end{bmatrix} \quad \mathbf{D}^\eta = \frac{1}{8} \begin{bmatrix} h_1^I U_1^I & h_1^I U_2^I & h_1^I U_3^I \\ 0 & 0 & 0 \\ h_2^I U_1^I & h_2^I U_2^I & h_2^I U_3^I \end{bmatrix} \quad (2.20)$$

$$\mathbf{J}^\zeta = \frac{1}{8} \begin{bmatrix} h_3^I X_1^I & h_3^I X_2^I & h_3^I X_3^I \\ h_2^I X_1^I & h_2^I X_2^I & h_2^I X_3^I \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{D}^\zeta = \frac{1}{8} \begin{bmatrix} h_3^I U_1^I & h_3^I U_2^I & h_3^I U_3^I \\ h_2^I U_1^I & h_2^I U_2^I & h_2^I U_3^I \\ 0 & 0 & 0 \end{bmatrix} \quad (2.21)$$

$$\mathbf{J}^{\xi\eta} = \frac{1}{8} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ h_4^I X_1^I & h_4^I X_2^I & h_4^I X_3^I \end{bmatrix} \quad \mathbf{D}^{\xi\eta} = \frac{1}{8} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ h_4^I U_1^I & h_4^I U_2^I & h_4^I U_3^I \end{bmatrix} \quad (2.22)$$

$$\mathbf{J}^{\xi\zeta} = \frac{1}{8} \begin{bmatrix} 0 & 0 & 0 \\ h_4^I X_1^I & h_4^I X_2^I & h_4^I X_3^I \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{D}^{\xi\zeta} = \frac{1}{8} \begin{bmatrix} 0 & 0 & 0 \\ h_4^I U_1^I & h_4^I U_2^I & h_4^I U_3^I \\ 0 & 0 & 0 \end{bmatrix} \quad (2.23)$$

$$\mathbf{J}^{\eta\zeta} = \frac{1}{8} \begin{bmatrix} h_4^I X_1^I & h_4^I X_2^I & h_4^I X_3^I \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{D}^{\eta\zeta} = \frac{1}{8} \begin{bmatrix} h_4^I U_1^I & h_4^I U_2^I & h_4^I U_3^I \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.24)$$

An efficient hourglass stabilization of a reduced integration finite element should be integrated analytically over the element domain. The best way to do so is to work with integrands which are polynomials. For this reason it's interesting to seek for a polynomial approximation of the inverse Jacobian matrix \mathbf{J}^{-1} , which still allows to display arbitrary element shapes with sufficient accuracy. One possible way to obtain a polynomial form of the inverse Jacobian matrix is to follow Legay and Combescure [LEG 03] and Reese [REE 07] or many other authors, who evaluate the Jacobian matrix only in the center of the element. Doing so the element is represented by means of its equivalent parallelepiped. In this case the Jacobian as well as the inverse Jacobian matrix are constant within the element. However, this assumption is only true for very fine meshes. For this reason, we approximate the inverse Jacobian by means of a Taylor expansion with respect to the center of the element, following the work of Schwatze [SCH 09]. This formulation benefits from that in two important manners: on one hand we are able to

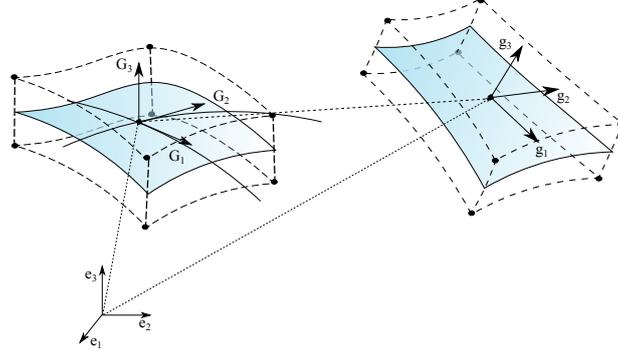


Figure 2.2: Configuration of the SB9 element

work with a polynomial form of the inverse Jacobian matrix, on the other hand arbitrary element shapes that deviate from the parallelepiped shape are more realistically taken into account.

From equation (2.16), the inverse of the Jacobian matrix is decomposed keeping only the constant and linear terms as follows

$$\mathbf{J}^{-1} \approx \mathbf{J}^{-1}|_{\xi=0} + \sum_{i=1}^3 \mathbf{J}_{,\xi_i}^{-1}|_{\xi=0} \xi_i \quad (2.25)$$

The constant term being easily determined, the work will be to determine the linear terms with very limited resources. To do so, equation (2.26) is simply derived with respect to the corresponding convective parameter and gives the equation (2.27).

$$\mathbf{J}\mathbf{J}^{-1} \approx (\mathbf{J}\mathbf{J}^{-1})|_{\xi=0} + \sum_{i=1}^3 (\mathbf{J}\mathbf{J}^{-1})_{,\xi_i}|_{\xi=0} \xi_i \quad (2.26)$$

$$0 = (\mathbf{J}\mathbf{J}^{-1})_{,\xi_i}|_{\xi=0} = \mathbf{J}_{,\xi_i}|_{\xi=0} \mathbf{J}^{-1}|_{\xi=0} + \mathbf{J}|_{\xi=0} \mathbf{J}_{,\xi_i}^{-1}|_{\xi=0} \quad (2.27)$$

From equation (2.27) one can easily determine the linear Jacobian terms, as follows, all terms being known or determined easily.

$$\mathbf{J}_{,\xi_i}^{-1}|_{\xi=0} = -(\mathbf{J}^0)^{-1} \mathbf{J}^{\xi_i} (\mathbf{J}^0)^{-1} \quad (2.28)$$

Hence, a good representation of the inverse Jacobian matrix is :

$$\mathbf{J}^{-1} \approx \mathbf{J}^{-1}|_{\xi=0} - \sum_{i=1}^3 (\mathbf{J}^0)^{-1} \mathbf{J}^{\xi_i} (\mathbf{J}^0)^{-1} \xi_i \quad (2.29)$$

2.3.3 Strain Field

The deformation gradient is written as follows

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \frac{\partial x^i}{\partial \xi^k} \mathbf{e}_i \otimes \frac{\partial \xi^k}{\partial X^j} \mathbf{e}_j = \mathbf{g}_i \otimes \mathbf{G}^i \quad (2.30)$$

\mathbf{F} is a tensor which maps the reference basis \mathbf{G}_i to the current one \mathbf{g}_i (Figure 2.2). \mathbf{e}_i is the Cartesian base vector. For simplicity, the superscript u will be omitted and \mathbf{E}^u will be written as \mathbf{E} . Further the Green Lagrange strain tensor is represented by it's Cartesian and covariant components as follows

$$\begin{aligned}\mathbf{E} &= \frac{1}{2}(\mathbf{g}_i \cdot \mathbf{g}_j - \mathbf{G}_i \cdot \mathbf{G}_j)\mathbf{G}^i \otimes \mathbf{G}^j \\ &= E_{ij}\mathbf{e}_i \otimes \mathbf{e}_j = \hat{E}_{ij}\mathbf{G}^i \otimes \mathbf{G}^j \\ &= \frac{1}{2} \left(\mathbf{G}_i \cdot \frac{\partial \mathbf{u}}{\partial \xi^j} + \frac{\partial \mathbf{u}}{\partial \xi^i} \cdot \mathbf{G}_j + \frac{\partial \mathbf{u}}{\partial \xi^j} \cdot \frac{\partial \mathbf{u}}{\partial \xi^i} \right) \mathbf{G}^i \otimes \mathbf{G}^j\end{aligned}\quad (2.31)$$

E_{ij} and \hat{E}_{ij} being respectively the components of the Green Lagrange tensor in the Cartesian and covariant frame. In Voigt notation the Cartesian and covariant Green-Lagrange strain are related as follows

$$\mathbf{E} = \mathbf{Q}\hat{\mathbf{E}} \quad (2.32)$$

with

$$\mathbf{E} = \{E_{11}, E_{22}, E_{33}, 2E_{12}, 2E_{23}, 2E_{13}\}^T \text{ and } \hat{\mathbf{E}} = \{\hat{E}_{\xi\xi}, \hat{E}_{\eta\eta}, \hat{E}_{\zeta\zeta}, 2\hat{E}_{\xi\eta}, 2\hat{E}_{\eta\zeta}, 2\hat{E}_{\xi\zeta}\}^T \quad (2.33)$$

and \mathbf{Q} a second-order matrix which contains the terms of the inverse of the jacobian \bar{J} matrix as follows

$$\mathbf{Q} = \begin{bmatrix} \bar{J}_{11}^2 & \bar{J}_{21}^2 & \bar{J}_{31}^2 & \bar{J}_{11}\bar{J}_{21} & \bar{J}_{21}\bar{J}_{31} & \bar{J}_{11}\bar{J}_{31} \\ \bar{J}_{12}^2 & \bar{J}_{22}^2 & \bar{J}_{32}^2 & \bar{J}_{12}\bar{J}_{22} & \bar{J}_{22}\bar{J}_{32} & \bar{J}_{12}\bar{J}_{32} \\ \bar{J}_{13}^2 & \bar{J}_{23}^2 & \bar{J}_{33}^2 & \bar{J}_{13}\bar{J}_{23} & \bar{J}_{23}\bar{J}_{33} & \bar{J}_{13}\bar{J}_{33} \\ 2\bar{J}_{11}\bar{J}_{12} & 2\bar{J}_{21}\bar{J}_{22} & 2\bar{J}_{31}\bar{J}_{32} & \bar{J}_{12}\bar{J}_{21} + \bar{J}_{11}\bar{J}_{22} & \bar{J}_{22}\bar{J}_{31} + \bar{J}_{21}\bar{J}_{32} & \bar{J}_{12}\bar{J}_{31} + \bar{J}_{12}\bar{J}_{32} \\ 2\bar{J}_{12}\bar{J}_{13} & 2\bar{J}_{22}\bar{J}_{23} & 2\bar{J}_{32}\bar{J}_{33} & \bar{J}_{13}\bar{J}_{22} + \bar{J}_{12}\bar{J}_{23} & \bar{J}_{23}\bar{J}_{32} + \bar{J}_{22}\bar{J}_{33} & \bar{J}_{13}\bar{J}_{32} + \bar{J}_{12}\bar{J}_{33} \\ 2\bar{J}_{11}\bar{J}_{13} & 2\bar{J}_{21}\bar{J}_{23} & 2\bar{J}_{31}\bar{J}_{33} & \bar{J}_{13}\bar{J}_{21} + \bar{J}_{11}\bar{J}_{23} & \bar{J}_{23}\bar{J}_{31} + \bar{J}_{21}\bar{J}_{33} & \bar{J}_{13}\bar{J}_{31} + \bar{J}_{11}\bar{J}_{33} \end{bmatrix} \quad (2.34)$$

By naming $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$ the row vectors of \mathbf{D} and $\mathbf{J}_1, \mathbf{J}_2, \mathbf{J}_3$ the row vectors of \mathbf{J} (see equation 2.16 and 2.17) the convective strain can be derived as follows

$$\hat{\mathbf{E}} = \begin{bmatrix} E_{c\xi\xi} \\ E_{c\eta\eta} \\ E_{c\zeta\zeta} \\ \Gamma_{c\xi\eta} \\ \Gamma_{c\eta\zeta} \\ \Gamma_{c\xi\zeta} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1^T \mathbf{D}_1 + 1/2\mathbf{D}_1^T \mathbf{D}_1 \\ \mathbf{J}_2^T \mathbf{D}_2 + 1/2\mathbf{D}_2^T \mathbf{D}_2 \\ \mathbf{J}_3^T \mathbf{D}_3 + 1/2\mathbf{D}_3^T \mathbf{D}_3 \\ \mathbf{J}_1^T \mathbf{D}_2 + \mathbf{J}_2^T \mathbf{D}_1 + \mathbf{D}_1^T \mathbf{D}_2 \\ \mathbf{J}_2^T \mathbf{D}_3 + \mathbf{J}_3^T \mathbf{D}_2 + \mathbf{D}_2^T \mathbf{D}_3 \\ \mathbf{J}_1^T \mathbf{D}_3 + \mathbf{J}_3^T \mathbf{D}_1 + \mathbf{D}_1^T \mathbf{D}_3 \end{bmatrix} \quad (2.35)$$

From the variation of the first line of equation (2.31) we can write the covariant strain-displacement $\hat{\mathbf{B}}_I$ at the node $I(I = 1, \dots, 8)$, such that $\delta\hat{E}_I = \hat{\mathbf{B}}_I \delta\mathbf{U}_I$ as follows

$$\hat{\mathbf{B}}_I = \begin{bmatrix} N_{I,\xi} \mathbf{g}_1^T \\ N_{I,\eta} \mathbf{g}_2^T \\ N_{I,\zeta} \mathbf{g}_3^T \\ N_{I,\xi} \mathbf{g}_2^T + N_{I,\eta} \mathbf{g}_1^T \\ N_{I,\xi} \mathbf{g}_3^T + N_{I,\zeta} \mathbf{g}_1^T \\ N_{I,\eta} \mathbf{g}_3^T + N_{I,\zeta} \mathbf{g}_2^T \end{bmatrix} = \begin{bmatrix} N_{I,\xi} \mathbf{j}_1^T \\ N_{I,\eta} \mathbf{j}_2^T \\ N_{I,\zeta} \mathbf{j}_3^T \\ N_{I,\xi} \mathbf{j}_2^T + N_{I,\eta} \mathbf{j}_1^T \\ N_{I,\xi} \mathbf{j}_3^T + N_{I,\zeta} \mathbf{j}_1^T \\ N_{I,\eta} \mathbf{j}_3^T + N_{I,\zeta} \mathbf{j}_2^T \end{bmatrix} \quad (2.36)$$

The SB9 element utilizes a reduced integration scheme with integration points aligned along the thickness direction $\xi = (0, 0, \zeta)$ (see Figure 2.1). Following this way, the cartesian compatible Green-Lagrange strain (equation 2.35) can be split into one part related to the integration and another part called hourglass strain

$$\hat{\mathbf{E}} = \hat{\mathbf{E}}^{\text{ri}} + \hat{\mathbf{E}}^{\text{stab}} \quad (2.37)$$

with

$$\begin{aligned} \hat{\mathbf{E}}^{\text{ri}} &= \hat{\mathbf{E}}^0 + \zeta \hat{\mathbf{E}}^\zeta \\ \hat{\mathbf{E}}^{\text{stab}} &= \xi \hat{\mathbf{E}}^\xi + \eta \hat{\mathbf{E}}^\eta + \xi \eta \hat{\mathbf{E}}^{\xi\eta} + \xi \zeta \hat{\mathbf{E}}^{\xi\zeta} + \eta \zeta \hat{\mathbf{E}}^{\eta\zeta} \end{aligned} \quad (2.38)$$

By considering equations (2.8) et (2.16), $\hat{\mathbf{B}}_I$ can also be decomposed into a reduced integrated part $\hat{\mathbf{B}}_I^{\text{ri}}$ and a stabilization part $\hat{\mathbf{B}}_I^{\text{stab}}$ as follows

$$\hat{\mathbf{B}}_I = \hat{\mathbf{B}}_I^{\text{ri}} + \hat{\mathbf{B}}_I^{\text{stab}} \quad (2.39)$$

with

$$\begin{aligned} \hat{\mathbf{B}}_I^{\text{ri}} &= \hat{\mathbf{B}}_I^0 + \zeta \hat{\mathbf{B}}_I^\zeta \\ \hat{\mathbf{B}}_I^{\text{stab}} &= \xi \hat{\mathbf{B}}_I^\xi + \eta \hat{\mathbf{B}}_I^\eta + \xi \eta \hat{\mathbf{B}}_I^{\xi\eta} + \eta \zeta \hat{\mathbf{B}}_I^{\eta\zeta} + \xi \zeta \hat{\mathbf{B}}_I^{\xi\zeta} \end{aligned} \quad (2.40)$$

The components of equations (2.38) and (2.40) are the basic building blocks of all solid-shell elements and are detailed in appendix A.

2.3.4 Assumed natural strain

2.3.4.1 Assumed shear strain

One way to reduce shear locking is to use the ANS concept following the work of Bathe and Dvorkin [BAT 85, DVO 95b] (see chapter 1). The transverse shear strain is evaluated in the four-middle edges points of the mid-surface of the solid-shell. To interpolate through the volume certain authors multiplied the mid-plane shear strain with the well known Reissner function [SAN 11, BAS 12]. This kind of interpolation ensures a systematic respect of the statical condition of shear stress in the element upper and lower faces. However in non-linear problem involving material non linearity and large deformation, that interpolation becomes obsolete since it does not guarantee the satisfaction of statical condition in the shell faces. Beside, such interpolation does not allow the element to naturally handle frictional contact problems since the shear deformation is assumed to be nil in the element upper and lower faces. Also it has been shown that such interpolation lead to a ill conditioned rigidity matrix. To avoid those inconveniences we applied the ANS technique following the work of Cardoso [CAR 08], Alves de Souza [SOU 05, SOU 06], Schwarze [REE 07]. In the covariant frame, each of the shear deformation $\hat{E}_{\eta\zeta}$ and $\hat{E}_{\xi\zeta}$ are interpolated using the equivalent shear deformation from four different tying points (figure 2.3). E(-1, 0, -1), F(1, 0, -1), G(1, 0, 1) and H(-1, 0, 1), for $\hat{E}_{\eta\zeta}$ and J(0, -1, -1), K(0, 1, -1), L(0, 1, 1) and M(0, -1, 1) for $\hat{E}_{\xi\zeta}$. Hence the assumed

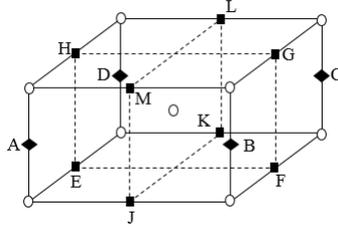


Figure 2.3: Positions of ANS tying points

shear deformations of the element is computed as follows

$$\begin{aligned}\hat{E}_{\xi\xi}^{\text{ANS}} &= N_J \hat{E}_{\xi\xi}^J + N_K \hat{E}_{\xi\xi}^K + N_L \hat{E}_{\xi\xi}^L + N_M \hat{E}_{\xi\xi}^M \\ \hat{E}_{\eta\xi}^{\text{ANS}} &= N_E \hat{E}_{\eta\xi}^E + N_F \hat{E}_{\eta\xi}^F + N_G \hat{E}_{\eta\xi}^G + N_H \hat{E}_{\eta\xi}^H\end{aligned}\quad (2.41)$$

with

$$\begin{aligned}N_I &= \frac{1}{4}(1 + \zeta_I \zeta)(1 + \eta_I \eta) \text{ for } I = J, \dots, M \\ N_J &= \frac{1}{4}(1 + \zeta_J \zeta)(1 + \xi_J \xi) \text{ for } J = E, \dots, H\end{aligned}\quad (2.42)$$

2.3.4.2 Assumed normal strain

In order to eliminate the curvature thickness locking, we follow the assumed pinching strain as suggested by Bischoff and Ramm [BIS 97], Betsch and Stein [BET 96], Schwatz and Reese [SCH 09].

$$\hat{E}_{\zeta\zeta}^{\text{ANS}} = N_A \hat{E}_{\zeta\zeta}^A + N_B \hat{E}_{\zeta\zeta}^B + N_C \hat{E}_{\zeta\zeta}^C + N_D \hat{E}_{\zeta\zeta}^D \quad (2.43)$$

$$N_k = \frac{1}{4}(1 + \xi_k \xi)(1 + \eta_k \eta) \text{ for } k = A, \dots, D \quad (2.44)$$

2.3.5 Stabilization procedure

2.3.5.1 Green Lagrange strain stabilization

The goal of the stabilization procedure is to correct the rank deficiency of the stiffness matrix coming from the adopted reduced integration scheme. The reader is referred to the important work in this subject by Liu, Belytschko and co-authors [LIU 98, BEL 83], Cardoso and co-authors [CAR 08, CAR 05], Alves de Souza [SOU 05, SOU 06], schwarze [SCH 09]. In some solid-shell formulation, the Jacobian matrix and it's inverse are only evaluated at the element center, see for example [BAS 12, ABE 02, ABE 09]. Such approximation assumes that the real element can be represented by it's equivalent parallelepiped. And this is quite accurate for thin and not very distorted meshes. However

for highly initially distorted meshes that assumption shows lack of accuracy [SCH 09]. In order to take into consideration the realistic shape of the element and stabilize properly the stiffness matrix, a polynomial decomposition of the inverse Jacobian matrix is derived as explicit in the subsection 2.3.2. Using equation (2.29) the matrix \mathbf{Q} of equation (2.34) is decomposed into constant and linear terms as follows

$$\mathbf{Q} \approx \mathbf{Q}^0 + \xi \mathbf{Q}^\xi + \eta \mathbf{Q}^\eta + \zeta \mathbf{Q}^\zeta \quad (2.45)$$

It has been shown in [SCH 09] that a very good accuracy is reached just using the constant and linear terms of \mathbf{Q} . Hence, the bilinear terms are ignored. In the same way, the covariant Green Lagrange strain tensor can be split into its constant, linear and bilinear terms and the Cartesian Green Lagrange strain can then be derived as follows

$$\begin{aligned} \mathbf{E} &\approx \left(\mathbf{Q}^0 + \xi \mathbf{Q}^\xi + \eta \mathbf{Q}^\eta + \zeta \mathbf{Q}^\zeta \right) \left(\hat{\mathbf{E}}^0 + \zeta \hat{\mathbf{E}}^\zeta + \zeta^2 \hat{\mathbf{E}}^{\zeta\zeta} + \xi \hat{\mathbf{E}}^\xi + \eta \hat{\mathbf{E}}^\eta + \xi \eta \hat{\mathbf{E}}^{\xi\eta} + \xi \zeta \hat{\mathbf{E}}^{\xi\zeta} + \eta \zeta \hat{\mathbf{E}}^{\eta\zeta} \right) \\ &\approx \underbrace{\mathbf{E}^0 + \zeta \mathbf{E}^\zeta + \zeta^2 \mathbf{E}^{\zeta\zeta}}_{\mathbf{E}^{\text{ri}}} + \underbrace{\xi \mathbf{E}^\xi + \eta \mathbf{E}^\eta + \xi \eta \mathbf{E}^{\xi\eta} + \xi \zeta \mathbf{E}^{\xi\zeta} + \eta \zeta \mathbf{E}^{\eta\zeta}}_{\mathbf{E}^{\text{stab}}} \end{aligned} \quad (2.46)$$

with

$$\mathbf{E}^0 = \mathbf{Q}^0 \hat{\mathbf{E}}^0 \quad (2.47)$$

$$\mathbf{E}^\zeta = \mathbf{Q}^0 \hat{\mathbf{E}}^\zeta + \mathbf{Q}^\zeta \hat{\mathbf{E}}^0 \quad (2.48)$$

$$\mathbf{E}^{\zeta\zeta} = \mathbf{Q}^0 \hat{\mathbf{E}}^{\zeta\zeta} + \mathbf{Q}^\zeta \hat{\mathbf{E}}^\zeta \quad (2.49)$$

$$\mathbf{E}^\xi = \mathbf{Q}^0 \hat{\mathbf{E}}^\xi + \mathbf{Q}^\xi \hat{\mathbf{E}}^0 \quad (2.50)$$

$$\mathbf{E}^\eta = \mathbf{Q}^0 \hat{\mathbf{E}}^\eta + \mathbf{Q}^\eta \hat{\mathbf{E}}^0 \quad (2.51)$$

$$\mathbf{E}^{\xi\eta} = \mathbf{Q}^0 \hat{\mathbf{E}}^{\xi\eta} + \mathbf{Q}^\xi \hat{\mathbf{E}}^\eta + \mathbf{Q}^\eta \hat{\mathbf{E}}^\xi \quad (2.52)$$

$$\mathbf{E}^{\eta\zeta} = \mathbf{Q}^0 \hat{\mathbf{E}}^{\eta\zeta} + \mathbf{Q}^\eta \hat{\mathbf{E}}^\zeta + \mathbf{Q}^\zeta \hat{\mathbf{E}}^\eta \quad (2.53)$$

$$\mathbf{E}^{\xi\zeta} = \mathbf{Q}^0 \hat{\mathbf{E}}^{\xi\zeta} + \mathbf{Q}^\xi \hat{\mathbf{E}}^\zeta + \mathbf{Q}^\zeta \hat{\mathbf{E}}^\xi \quad (2.54)$$

Thus

$$\mathbf{E} = \mathbf{E}^{\text{ri}} + \mathbf{E}^{\text{stab}} \quad (2.55)$$

The strain-displacement gradient can be split in the same way

$$\begin{aligned} \mathbf{B} &\approx \left(\mathbf{Q}^0 + \xi \mathbf{Q}^\xi + \eta \mathbf{Q}^\eta + \zeta \mathbf{Q}^\zeta \right) \left(\hat{\mathbf{B}}^0 + \zeta \hat{\mathbf{B}}^\zeta + \zeta^2 \hat{\mathbf{B}}^{\zeta\zeta} + \xi \hat{\mathbf{B}}^\xi + \eta \hat{\mathbf{B}}^\eta + \xi \eta \hat{\mathbf{B}}^{\xi\eta} + \xi \zeta \hat{\mathbf{B}}^{\xi\zeta} + \eta \zeta \hat{\mathbf{B}}^{\eta\zeta} \right) \\ &\approx \underbrace{\mathbf{B}^0 + \zeta \mathbf{B}^\zeta + \zeta^2 \mathbf{B}^{\zeta\zeta}}_{\mathbf{B}^{\text{ri}}} + \underbrace{\xi \mathbf{B}^\xi + \eta \mathbf{B}^\eta + \xi \eta \mathbf{B}^{\xi\eta} + \xi \zeta \mathbf{B}^{\xi\zeta} + \eta \zeta \mathbf{B}^{\eta\zeta}}_{\mathbf{B}^{\text{stab}}} \end{aligned} \quad (2.56)$$

with

$$\mathbf{B}^0 = \mathbf{Q}^0 \hat{\mathbf{B}}^0 \quad (2.57)$$

$$\mathbf{B}^\zeta = \mathbf{Q}^0 \hat{\mathbf{B}}^\zeta + \mathbf{Q}^\zeta \hat{\mathbf{B}}^0 \quad (2.58)$$

$$\mathbf{B}^{\zeta\zeta} = \mathbf{Q}^0 \hat{\mathbf{B}}^{\zeta\zeta} + \mathbf{Q}^\zeta \hat{\mathbf{B}}^\zeta \quad (2.59)$$

$$\mathbf{B}^\xi = \mathbf{Q}^0 \hat{\mathbf{B}}^\xi + \mathbf{T}^\xi \hat{\mathbf{B}}^0 \quad (2.60)$$

$$\mathbf{B}^\eta = \mathbf{Q}^0 \hat{\mathbf{B}}^\eta + \mathbf{Q}^\eta \hat{\mathbf{B}}^0 \quad (2.61)$$

$$\mathbf{B}^{\xi\eta} = \mathbf{Q}^0 \hat{\mathbf{B}}_\zeta^{\xi\eta} + \mathbf{Q}^\xi \hat{\mathbf{B}}^\eta + \mathbf{Q}^\eta \hat{\mathbf{B}}_\zeta^\xi \quad (2.62)$$

$$\mathbf{B}^{\eta\zeta} = \mathbf{Q}^0 \hat{\mathbf{B}}^{\eta\zeta} + \mathbf{Q}^\eta \hat{\mathbf{B}}^\zeta + \mathbf{Q}^\zeta \hat{\mathbf{B}}^\eta \quad (2.63)$$

$$\mathbf{B}^{\xi\zeta} = \mathbf{Q}^0 \hat{\mathbf{B}}^{\xi\zeta} + \mathbf{Q}^\xi \hat{\mathbf{B}}^\zeta + \mathbf{Q}^\zeta \hat{\mathbf{B}}^\xi \quad (2.64)$$

so that \mathbf{B} can be written as

$$\mathbf{B} = \mathbf{B}^{\text{ri}} + \mathbf{B}^{\text{stab}} \quad (2.65)$$

\mathbf{E}^{stab} and \mathbf{B}^{stab} represent the terms cancelled in the reduced integration. The process of stabilization consists in analytically restoring those terms into the stiffness matrix. Furthermore to eliminate volumetric locking, the B-bar approach [HUG 80] is adopted. Hence the hourglass counterpart of the stabilized strain end strain-displacement operators are split into there volumetric and deviatoric components, and only the deviatoric part are kept

$$\mathbf{B}^{\text{stab}} = \mathbf{B}_{\text{dev}}^{\text{stab}}(\xi, \eta, \zeta) + \mathbf{B}_{\text{vol}}^{\text{stab}}(0, 0, 0) \quad (2.66)$$

Since no constant term is present in the expansion of \mathbf{B}^{stab} (see equation 2.56)

$$\mathbf{B}_{\text{vol}}^{\text{stab}}(0, 0, 0) = 0 \quad (2.67)$$

and

$$\mathbf{B}^{\text{stab}} = \mathbf{B}_{\text{dev}}^{\text{stab}}(\xi, \eta, \zeta) \quad (2.68)$$

The stabilization components of the strain and strain-displacement matrices being well identified, a similar approach is used to identify the stabilization stress.

2.3.5.2 Second Piola Kirchhoff stress stabilization

From equation (2.55), since the two parts of the Green Lagrange strain tensor are orthogonal, the internal energy is split like

$$W(\mathbf{E}) = W^{\text{ri}}(\mathbf{E}^{\text{ri}}) + W^{\text{stab}}(\mathbf{E}^{\text{stab}}) \quad (2.69)$$

Since second Piola Kirchhoff tensor is the derivative of the internal energy with respect to Green-Lagrange strain, the stabilization counterpart can be identified by simply writing the derivative

$$\mathbf{S} = \frac{\partial W(\mathbf{E})}{\partial \mathbf{E}} \quad (2.70)$$

$$= \frac{\partial W^{\text{ri}}(\mathbf{E}^{\text{ri}})}{\partial \mathbf{E}^{\text{ri}}} \frac{\partial \mathbf{E}^{\text{ri}}}{\partial \mathbf{E}} + \frac{\partial W^{\text{stab}}(\mathbf{E}^{\text{stab}})}{\partial \mathbf{E}^{\text{stab}}} \frac{\partial \mathbf{E}^{\text{stab}}}{\partial \mathbf{E}} \quad (2.71)$$

$$= \mathbf{S}^{\text{ri}}(\mathbf{E}^{\text{ri}}) + \mathbf{S}^{\text{stab}}(\mathbf{E}^{\text{stab}}) \quad (2.72)$$

$\mathbf{S}^{\text{ri}}(\mathbf{E}^{\text{ri}})$ is Piola Kirchhoff obtained from the integration of constitutive law and $\mathbf{S}^{\text{stab}}(\mathbf{E}^{\text{stab}})$ Piola Kirchhoff stress for stabilization. The key point of the stabilization is to find an optimal way to evaluate that stabilization stress without integrating it so that the computation time is minimized. Now just as being explained before, to avoid volumetric locking, only the deviatoric part of the Piola Kirchhoff stabilization tensor is kept. Doing so, the stabilization of the stress is given as

$$\mathbf{S}^{\text{stab}} = \mathbf{C}^{\text{stab}} : \mathbf{E}^{\text{stab}} \quad (2.73)$$

\mathbf{C}^{stab} being the deviatoric part of St Venant Kirchhoff material given as follows

$$\mathbf{C}^{\text{stab}} = \mu^{\text{stab}} \begin{bmatrix} \frac{4}{3} & \frac{-2}{3} & \frac{-2}{3} & 0 & 0 & 0 \\ \frac{-2}{3} & \frac{4}{3} & \frac{-2}{3} & 0 & 0 & 0 \\ \frac{-2}{3} & \frac{-2}{3} & \frac{4}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.74)$$

With μ^{stab} the shear modulus evaluated, in elastic material behavior, as $\mu^{\text{stab}} = E/2(1 + \nu)$, in which E and ν are respectively the Young's modulus and the Poisson ratio. For inelastic materials, using the elastic shear modulus lead to an overestimation of the stabilization stress. To overcome such problem the secant modulus as defined in Belytschko and Bindeman [BEL 93] is used

$$2\mu^{\text{stab}} = \sqrt{\frac{\pi_{\mathbf{S}}}{\pi_{\mathbf{E}}}}, \quad (2.75)$$

where

$$\pi_{\mathbf{S}} = \frac{1}{2} \mathbf{S}_{\text{dev}} : \mathbf{S}_{\text{dev}} \quad , \quad \pi_{\mathbf{E}} = \frac{1}{2} \mathbf{E}_{\text{dev}} : \mathbf{E}_{\text{dev}} \quad (2.76)$$

These ways of evaluating the stabilization parameter μ^{stab} is very convenient because there is no need for consistent linearization of stabilization constitutive matrix, which is very interesting in terms of computation time. Moreover, the proposed definition of \mathbf{C}^{stab} ensures an efficient and robust stabilization as will be shown in chapter 3 in different numerical examples.

2.3.6 The ninth node

The additional central node is endowed with only one translation degree of freedom in the element thickness direction ζ . The enhanced normal displacement in the thickness direction is written as follows

$$u_{\zeta}(\xi, \eta, \zeta) = u_{\zeta}^H(\xi, \eta, \zeta) + (1 - \zeta^2)u_{\zeta}^9 \quad (2.77)$$

u_{ζ}^9 being the one and only relative displacement of the ninth node, in through the thickness direction \mathbf{g}_3 as proposed by ahmad [AHM 70], u_{ζ}^H the normal displacement of the

eight node element considered alone. A new column is then added to the covariant matrix $\hat{\mathbf{B}}^H$ making it a 6 by 25 matrix $\hat{\mathbf{B}}$

$$\hat{\mathbf{B}} = [\hat{\mathbf{B}}^H \quad \hat{\mathbf{B}}^9] \quad (2.78)$$

with

$$\hat{\mathbf{B}}^9 = [0 \quad 0 \quad -2\zeta \quad 0 \quad 0 \quad 0]^T$$

One can then write the variation of the pinching strain in the covariant basis as follows

$$\delta E_{\zeta\zeta} = \hat{\mathbf{B}}_3^H \delta \mathbf{u}_\zeta^H - 2\zeta \delta u_\zeta^9 \quad (2.79)$$

$\hat{\mathbf{B}}_3^H$ being the third line of matrix $\hat{\mathbf{B}}^H$.

2.4 Seven nodes solid-shell element (SB7)

The assumed strain method and reduced integration technique as applied in a hexahedral element does not eliminate the locking problems in a prismatic solid-shell finite element. Hence, in addition to using the assumed strain method, the reduced integration technique and the EAS method, the bending deformation of the DKT6 element is used to enrich the deformation of the SB7 finite element. Since DKT6 is a classical shell finite element with rotational degrees of freedom in small perturbation, it is then difficult to formulate SB7 in Total Lagrangian as is the case with SB9. An updated Lagrangian formulation with co-rotational frame is then adopted.

2.4.1 Local frame and objective stress rate

In the updated Lagrangian approach, it is useful to establish a set of configurations related to which each particle in the deforming body can be referred to. In this sense, we distinguish three configurations: the material (reference), the spatial (current) and the parametric (convective) configuration. Consider the natural coordinates $\mathbf{s} = (\xi, \eta, \zeta)$ representing the isoparametric prismatic domain (Figure 2.4). Without loss of generality, the reference configuration can be related to a converged state (n , the last increment), whereas the current configuration points corresponds to the next time increment ($n+1$). For the solid-shell topology treated in this part, any point in the reference configuration can be defined by the position vector $\mathbf{x}(s)$ as

$$\begin{aligned} {}^n\mathbf{x}(s) &= \frac{1}{2}(1+\zeta) \sum_{i=1}^3 N_i \mathbf{x}_t^i + \frac{1}{2}(1-\zeta) \sum_{i=1}^3 N_i \mathbf{x}_b^i \\ N_1 &= \xi \quad N_2 = \eta \quad N_3 = 1 - \xi - \eta \end{aligned} \quad (2.80)$$

where the subscripts t and b denote the projections of the variable onto the top ($\zeta = +1$) and bottom ($\zeta = -1$) surfaces, respectively and N_i the isoparametric shape function. The corresponding position after incremental deformation (current configuration) can

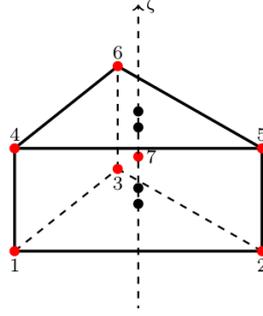


Figure 2.4: Topology of the SB7 solid-shell element

be defined by an analogous expression, referred to state $(n+1)$. The partial derivatives of the position vector ${}^n\mathbf{x}(\mathbf{s})$ allows for the definition of the covariant basis vector in the current and reference configurations, in the form

$$\begin{aligned} {}^n\mathbf{g}_a &= \frac{\partial {}^n\mathbf{x}}{\partial s_a}, \quad a = (1, 2, 3) \\ {}^{n+1}\mathbf{g}_a &= \frac{\partial {}^{n+1}\mathbf{x}}{\partial s_a} \end{aligned} \quad (2.81)$$

The incremental deformation gradient \mathbf{F} between configurations (n) and $(n+1)$ and the incremental displacement-based Green–Lagrange strain tensor (\mathbf{E}^u) can then be written as

$${}^{n+1}\mathbf{F} = \frac{\partial {}^{n+1}\mathbf{x}}{\partial {}^n\mathbf{x}} \quad (2.82)$$

$${}^{n+1}\mathbf{E} = \frac{1}{2}({}^{n+1}\mathbf{F}^T {}^{n+1}\mathbf{F} - \mathbf{I}) \quad (2.83)$$

$${}^{n+1}\mathbf{E}_{ab}^u = \frac{1}{2}({}^n\mathbf{g}_a \cdot \frac{\partial {}^{n+1}\mathbf{u}}{\partial s_b} + \frac{\partial {}^{n+1}\mathbf{u}}{\partial s_a} \cdot {}^n\mathbf{g}_b + \frac{\partial {}^{n+1}\mathbf{u}}{\partial s_a} \cdot \frac{\partial {}^{n+1}\mathbf{u}}{\partial s_b}) \quad (2.84)$$

with \mathbf{I} as the second-order identity tensor. The convective frame is selected as the departure point for the strain components and also for the enhancing strain field used in this element. As work conjugate measure for the Green–Lagrange strain tensor, the second Piola–Kirchhoff stress tensor \mathbf{S} is chosen

$$\mathbf{S} = S^{ab} {}^n\mathbf{g}_a \otimes {}^n\mathbf{g}_b \quad (2.85)$$

Due to the nonorthogonal character of the convective frame, a local orthonormal frame $\mathbf{r} = (\mathbf{r}^1, \mathbf{r}^2, \mathbf{r}^3)$ is defined at the element center ($\xi = 1/3, \eta = 1/3, \zeta = 0$) as

$$\begin{aligned} \mathbf{r}^1 &= \frac{{}^n\mathbf{g}_1}{\|{}^n\mathbf{g}_1\|} \\ \mathbf{r}^2 &= \frac{{}^n\mathbf{g}_2 + \mathbf{g}_c}{\|{}^n\mathbf{g}_2 + \mathbf{g}_c\|} \\ \mathbf{r}^3 &= \mathbf{r}^1 \wedge \mathbf{r}^2 \end{aligned} \quad (2.86)$$

where \mathbf{g}_c ensures orthogonality and is defined as follows

$$\mathbf{g}_c = -\frac{{}^n\mathbf{g}_1 \cdot {}^n\mathbf{g}_2}{{}^n\mathbf{g}_1 \cdot {}^n\mathbf{g}_1} {}^n\mathbf{g}_1 \quad (2.87)$$

After being defined in the undeformed configuration, the local frame is subjected to the rigid-body part of the relative deformation gradient, evaluated between configurations (n) and $(n+1)$. In fact, the polar decomposition applied to the relative deformation gradient of Equation (2.82) leads to

$${}^{n+1}\mathbf{F} = {}^{n+1}\mathbf{R} {}^n\mathbf{U} \quad (2.88)$$

where ${}^{n+1}\mathbf{R}$ and ${}^n\mathbf{U}$ are, respectively, the rotation tensor and the right-stretch tensor. Doing so, the local orthonormal coordinate system can be easily updated between stages (n) and $(n+1)$ in the form

$${}^{n+1}\mathbf{r} = {}^{n+1}\mathbf{R} {}^n\mathbf{r} \quad (2.89)$$

In this way, the local coordinate system for a given point is only affected by the rigidbody component of the total deformation, characterizing this frame as a corotational one, with material strain (and stress) tensors being rotated as if they were frozen into the deformed continuum [CAR 05]. The main advantage of the adoption of a corotational local frame is the simplified treatment of nonlinearities, either geometric or material, for solids subjected to finite rotations and displacements. A complete description of the constitutive behaviour of a given body can be obtained resorting to material stress and strain corotational tensors, continuously referred to the local frame and, consequently, following the deformation path from the beginning. Examples of such tensors are the rotated second Piola–Kirchhoff stress tensor and the rotated Green–Lagrange strain tensor, evaluated at the continuously rotated local reference system. This fact turns to be very useful, once the adopted EAS procedure is entirely designed resorting to the material frame. Additionally, the use of material tensors grants the objectivity within the formulation. In fact, the primary choice of a reference system rotating with the continuum body fulfills the objectivity requirements, for an observer situated at the body and affected by the rotation (but not the stretch) part of the deformation [SIM 98]. The update of the stress tensor and the internal force in the corotational framework is made under the assumption that the magnitude of the incremental strain is small, and is comparable to the order of strain in small deformation theory. The rotated stress rate $\dot{\boldsymbol{\sigma}}$ in an incremental form, and without loss of generality, assumes the rate of change of $\boldsymbol{\sigma}$ to be constant within the step [MAS 00]

$$\begin{aligned} {}^{n+1}\boldsymbol{\sigma} &= {}^n\boldsymbol{\sigma} + \Delta t \dot{\boldsymbol{\sigma}} \\ &= {}^n\boldsymbol{\sigma} + {}^{n+1}\boldsymbol{\sigma} \\ &= {}^n\boldsymbol{\sigma} + \frac{1}{J} {}^{n+1}\mathbf{F} {}^n\mathbf{S} {}^{n+1}\mathbf{F}^T \end{aligned} \quad (2.90)$$

where ${}^{n+1}\boldsymbol{\sigma}$ is the incremental Cauchy stress, ${}^n\mathbf{S}$ the incremental second Piola–Kirchhoff stress tensor in the rotated reference and $J = \det({}^{n+1}\mathbf{F})$. If the load incremental or the step size is kept small, the pure deformation part of the incremental motion

is a small quantity as compared to the element dimensions when measured in the local co-rotated frame. Consequently, it is reasonable to make the following two assumptions:

- The changes in element shapes are small in each individual local increment, meaning that ${}^{n+1}_n \mathbf{F} \approx \mathbf{I}$ and therefore $J = 1$.
- The gradients of pure deformational part of the incremental motion, when measured in the local co-rotated frame are of order of magnitude as the small strains. Therefore, quadratic terms in the Green-Lagrange strain tensor defined on the rotated frame can be neglected. (See Belytschko and Hsieh [BEL 73])

Based on these assumptions, the stress update (2.90) can be expressed in an iterative form as

$${}^{n+1} \boldsymbol{\sigma} = {}^n \boldsymbol{\sigma} + \mathbf{C} \mathbf{B} {}^{n+1} \mathbf{r}^T \Delta \mathbf{U} \quad (2.91)$$

with \mathbf{C} as the constitutive tensor and \mathbf{B} the standard strain–displacement operator that is defined in the next sections. This approach is equivalent to the adoption of a hypoelastic constitutive model representative of a Green–Naghdi objective stress rate, which will derive an additive constitutive update of the stress tensor also in the nonlinear material range.

2.4.2 Strain tensor

The conventional Jacobian matrix in the local frame is written as follows

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \quad (2.92)$$

The Green-Lagrange strain tensor can be interpolated all over the element’s domain as follows

$$\mathbf{E} = \mathbf{B}^u \mathbf{d} \quad (2.93)$$

where \mathbf{B}^u is the standard $[6 \times 18]$ strain–displacement operator, well-defined in the literature

$$\mathbf{B}^{uI} = \begin{bmatrix} \frac{\partial N^I}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N^I}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N^I}{\partial z} \\ \frac{\partial N^I}{\partial y} & \frac{\partial N^I}{\partial x} & 0 \\ \frac{\partial N^I}{\partial z} & 0 & \frac{\partial N^I}{\partial x} \\ 0 & \frac{\partial N^I}{\partial z} & \frac{\partial N^I}{\partial y} \end{bmatrix} \quad (2.94)$$

From the reduced integration scheme used in this work, the shape function derivative can be split as follows

$$\frac{\partial N^I}{\partial \mathbf{X}} \approx \frac{\partial N^I}{\partial \mathbf{X}} \Big|_{(\frac{1}{3}, \frac{1}{3}, 0)} + z \frac{\partial}{\partial z} \left(\frac{\partial N^I}{\partial \mathbf{X}} \right) \Big|_{(\frac{1}{3}, \frac{1}{3}, 0)} \quad (2.95)$$

where $\mathbf{X} = (x, y, z)$. Considering the equation (2.95) the terms of matrix \mathbf{B}'' can also be split into two parts as follows

$$\mathbf{B}'' = \mathbf{B}''|_{(\frac{1}{3}, \frac{1}{3}, 0)} + z\mathbf{B}''_z \quad (2.96)$$

The deformation defined in equation (2.93) can be separated into three parts : the in plane (IP) part \mathbf{E}_{IP} , the normal or pinching part \mathbf{E}_P and the transverse shear part \mathbf{E}_{SH}

$$\mathbf{E} = \mathbf{E}_{IP} + \mathbf{E}_P + \mathbf{E}_{SH} \quad (2.97)$$

For the SB7 to perform well in the simulation of bending shell like structures, some treatments have to be made to eliminate locking phenomena.

2.4.3 In plane strain enhancement

To treat properly the in-plane deformation \mathbf{E}_{IP} , it is splitted as the combination of two contributions : the first, constant, being a membrane contribution and the second, depending on z , viewed as a bending contribution (see equation 2.98).

$$\mathbf{E}_{IP} = \mathbf{E}_{IP}^m + z\mathbf{E}_{IP}^b \quad (2.98)$$

where

$$\mathbf{E}_{IP}^m = \begin{bmatrix} E_{11}^m \\ E_{22}^m \\ 2E_{12}^m \end{bmatrix} = \begin{bmatrix} \mathbf{m}_x^T & 0 \\ 0 & \mathbf{m}_y^T \\ \mathbf{m}_y^T & \mathbf{m}_x^T \end{bmatrix} \mathbf{d} \quad \text{and} \quad \mathbf{E}_{IP}^b = \begin{bmatrix} E_{11}^b \\ E_{22}^b \\ 2E_{12}^b \end{bmatrix} = \begin{bmatrix} \mathbf{b}_x^T & 0 \\ 0 & \mathbf{b}_y^T \\ \mathbf{b}_y^T & \mathbf{b}_x^T \end{bmatrix} \mathbf{d} \quad (2.99)$$

with

$$\begin{bmatrix} \mathbf{m}_x^T \\ \mathbf{m}_y^T \\ \mathbf{m}_z^T \end{bmatrix} = (\mathbf{J}^{-1} \frac{\partial \mathbf{N}}{\partial \xi})|_{(\frac{1}{3}, \frac{1}{3}, 0)} \quad \text{and} \quad \begin{bmatrix} \mathbf{b}_x^T \\ \mathbf{b}_y^T \\ \mathbf{b}_z^T \end{bmatrix} = \frac{\partial}{\partial z} (\mathbf{J}^{-1} \frac{\partial \mathbf{N}}{\partial \xi})|_{(\frac{1}{3}, \frac{1}{3}, 0)} \quad (2.100)$$

Now if the bending contribution is kept as in equation (2.98) the SB7 element would be very rigid and would show poor results in bending dominant problems. To circumvent this issue, \mathbf{E}_{IP}^b is replaced by a curvature κ directly derived from shell theory (Discrete Kirchhoff triangular element or DKT6, see [ROE 92])

$$\mathbf{E}_{IP} = \mathbf{E}_{IP}^m + z\kappa \quad (2.101)$$

For the prism described here, Figure 2.5.(a) shows the mean triangular area defined by points 1, 2 and 3 (midpoints of the edges i-l, j-m, k-n, respectively), the normal to this surface being \mathbf{e}_z . Figure 2.5.(b) shows this mid-surface as a 6-node triangular element whose degrees of freedom defining bending effects are three translations (w_1, w_2, w_3) in the \mathbf{e}_z direction and three rotations ($\theta_4, \theta_5, \theta_6$) in the direction of each side, at the nodes 4, 5 and 6 located at the midpoints of these sides. Finally Figure 2.5.(c) shows

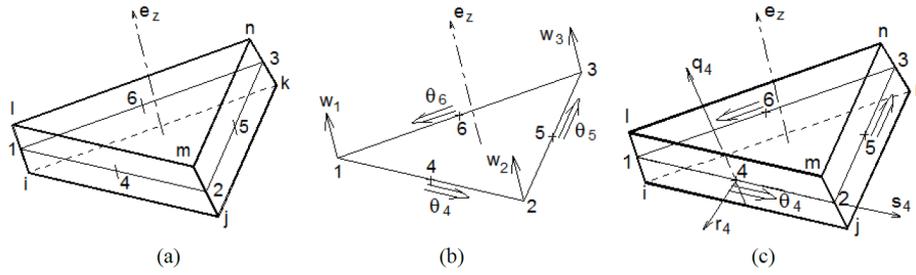


Figure 2.5: Description of prismatic solid-shell element

how to express, for example, θ_4 by using nodal displacement components (projection on the axis \mathbf{r}_4) of nodes i, j, m, l of the prism. And it would be the same for θ_5 by using translational dof of nodes j, k, n, m and for θ_6 with translations of nodes k, l, n of the prism. Finally, rotations $\theta_4, \theta_5, \theta_6$ are connected to the nodal displacements. Otherwise, w_1 is obtained with the translations of nodes i and l , w_2 with the translations of nodes j and m , and w_3 with the translations of nodes k and n .

2.4.3.1 DKT6 Shell Finite element

The DKT6 [ROE 92] element defines the bending of a plate with a Kirchhoff kinematic. It has three translational dof normal to the surface, located at the vertices, and three rotational dof, located at midpoints of the sides (Figure 2.5.(b)). Conventionally, interpolation functions use area coordinates (A : area of the triangle)

$$w(x, y) = N_1 \cdot w_1 + N_2 \cdot w_2 + N_3 \cdot w_3, \quad N_i(x, y) = A_i(x, y)/A \quad (2.102)$$

$$\theta_x(x, y) = (1 - 2N_3) \cdot \theta_{x4} + (1 - 2N_1) \cdot \theta_{x5} + (1 - 2N_2) \cdot \theta_{x6} \quad (2.103)$$

$$\theta_y(x, y) = (1 - 2N_3) \cdot \theta_{y4} + (1 - 2N_1) \cdot \theta_{y5} + (1 - 2N_2) \cdot \theta_{y6} \quad (2.104)$$

Hence at node 4 (middle of the side 1-2), $\theta_x = \theta_{x4}$, $\theta_y = \theta_{y4}$ and the same goes for the node 5 (middle of the side 2-3) and the node 6 (middle of the side 3-1). After derivations

$$N_{1,x}(x, y) = \frac{1}{2A}(y_2 - y_3); \quad N_{1,y}(x, y) = \frac{1}{2A}(x_2 - x_3) \quad (2.105)$$

$$N_{2,x}(x, y) = \frac{1}{2A}(y_3 - y_1); \quad N_{2,y}(x, y) = \frac{1}{2A}(x_1 - x_3) \quad (2.106)$$

$$N_{3,x}(x, y) = \frac{1}{2A}(y_1 - y_2); \quad N_{3,y}(x, y) = \frac{1}{2A}(x_2 - x_1) \quad (2.107)$$

For simplicity, the following notation is used : $x_{mn} = x_m - x_n$; $y_{mn} = y_m - y_n$ We can then write the curvature deformation as follows

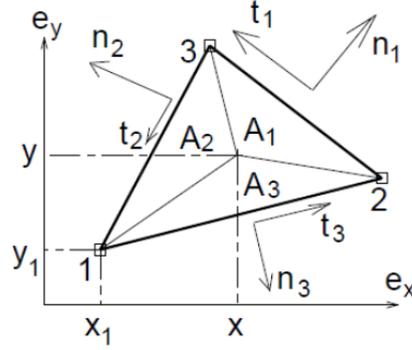


Figure 2.6: Normal vectors and tangential vectors for each triangular side

$$\begin{bmatrix} \kappa_{xx} \\ \kappa_{yy} \\ 2\kappa_{xy} \end{bmatrix} = \begin{bmatrix} \theta_{y,x} \\ -\theta_{x,y} \\ \theta_{y,y} - \theta_{x,x} \end{bmatrix} = \frac{1}{A} \begin{bmatrix} y_{21} & y_{32} & y_{13} \\ x_{21} & x_{32} & x_{13} \\ y_{12} & x_{12} & y_{23} & x_{23} & y_{31} & x_{31} \end{bmatrix} \begin{bmatrix} \theta_{x4} \\ \theta_{y4} \\ \theta_{x5} \\ \theta_{y5} \\ \theta_{x6} \\ \theta_{y6} \end{bmatrix} \quad (2.108)$$

It is useful to express the normal vectors and the tangential vectors in the local basis (see Figure 2.6)

$$\mathbf{n}_1 = c_1 \mathbf{e}_x + s_1 \mathbf{e}_y \quad (2.109)$$

$$\mathbf{t}_1 = -s_1 \mathbf{e}_x + c_1 \mathbf{e}_y \quad (2.110)$$

$$c_1 = \frac{y_{32}}{L_1} \text{ and } s_1 = -\frac{x_{32}}{L_1} \quad (2.111)$$

$$\mathbf{n}_2 = c_2 \mathbf{e}_x + s_2 \mathbf{e}_y \quad (2.112)$$

$$\mathbf{t}_2 = -s_2 \mathbf{e}_x + c_2 \mathbf{e}_y \quad (2.113)$$

$$c_2 = \frac{y_{13}}{L_2} \text{ and } s_2 = -\frac{x_{13}}{L_2} \quad (2.114)$$

$$\mathbf{n}_3 = c_3 \mathbf{e}_x + s_3 \mathbf{e}_y \quad (2.115)$$

$$\mathbf{t}_3 = -s_3 \mathbf{e}_x + c_3 \mathbf{e}_y \quad (2.116)$$

$$c_3 = \frac{y_{21}}{L_3} \text{ and } s_3 = -\frac{x_{21}}{L_3} \quad (2.117)$$

$L_1, L_2,$ and L_3 being respectively the side length 2-3, 3-1 and 2-1 (see Figure 2.6). The decomposition of the rotation vectors in their local basis can be given as

$$\theta_4 = \theta_{x4} \mathbf{e}_x + \theta_{y4} \mathbf{e}_y = \theta_4 \mathbf{t}_3 + \frac{w_2 - w_1}{L_3} \mathbf{n}_3 \quad (2.118)$$

$$\theta_{x4} = -s_3 \theta_4 + c_3 \frac{w_2 - w_1}{L_3}, \quad \theta_{y4} = c_3 \theta_4 + s_3 \frac{w_2 - w_1}{L_3} \quad (2.119)$$

$\theta_5, \theta_6, \theta_{x5}, \theta_{y5}, \theta_{x6}, \theta_{y6}$ are given by permuting the indices. The curvatures will be expressed in terms of the rotational degrees of freedom and the normal nodal displacement

$$\boldsymbol{\kappa} = \mathbf{B}_\theta \begin{bmatrix} \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix} + \mathbf{B}_w \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \quad (2.120)$$

where the matrices \mathbf{B}_θ and \mathbf{B}_w are given like this

$$\mathbf{B}_\theta = \frac{1}{A} \begin{bmatrix} c_3^2 L_3 & c_1^2 L_1 & c_2^2 L_2 \\ s_3^2 L_3 & s_1^2 L_1 & s_2^2 L_2 \\ 2c_3 s_3 L_3 & 2c_1 s_1 L_1 & 2c_2 s_2 L_2 \end{bmatrix} \quad (2.121)$$

$$\mathbf{B}_w = \frac{1}{A} \begin{bmatrix} c_2 s_2 - c_3 s_3 & c_3 s_3 - c_1 s_1 & c_1 s_1 - c_2 s_2 \\ c_3 s_3 - c_2 s_2 & c_1 s_1 - c_3 s_3 & c_2 s_2 - c_1 s_1 \\ 2(c_3^2 - c_2^2) & 2(c_1^2 - c_3^2) & 2(c_2^2 - c_1^2) \end{bmatrix} \quad (2.122)$$

2.4.3.2 Linking the triangle to the prism

In the local frame of the triangular finite element, which is also the frame of the associated prism, the displacement of the six nodes of the prism projected on \mathbf{e}_z are denoted : $w_i, w_j, w_k, w_l, w_m, w_n$. A trivial method gives directly w_1, w_2, w_3 as a function of $w_i, w_j, w_k, w_l, w_m, w_n$

$$w_1 = \frac{w_i + w_l}{2}; w_2 = \frac{w_j + w_m}{2}; w_3 = \frac{w_k + w_n}{2}; \quad (2.123)$$

Using \mathbf{B}_w (B_{wij} being the i^{th} row and the j^{th} column component) and performing a base change to make the nodal translations $U_{Xi}, U_{Yi}, U_{Zi}, U_{Xj}, \dots, U_{Zn}$ be expressed in the global frame of the structure (X, Y, Z), a matrix $\bar{\mathbf{B}}_w^b$ detailed in equation (2.124) is obtained. This matrix connects the part of the curvatures due to the normal translations to the global displacements of the six nodes.

$$\bar{\mathbf{B}}_w^b = [\mathbf{B}_w^b \quad \mathbf{B}_w^b] \quad (2.124)$$

with

$$\mathbf{B}_w^b = \begin{bmatrix} \mathbf{B}_{w(1,i)} \mathbf{e}_z^T \\ \mathbf{B}_{w(2,i)} \mathbf{e}_z^T \\ \mathbf{B}_{w(3,i)} \mathbf{e}_z^T \end{bmatrix}, \quad i = 1, 2, 3 \quad (2.125)$$

The last development for the expression of bending effects consists in linking $\theta_4, \theta_5, \theta_6$ to the 18 displacements $U_{Xi}, U_{Yi}, U_{Zi}, U_{Xj}, \dots, U_{Zn}$ of the prism's nodes. For example, for the rotation θ_4 , we proceed as follows

Two points 'a' and 'b' are located on the middle of the edge i-j and m-l (see Figure

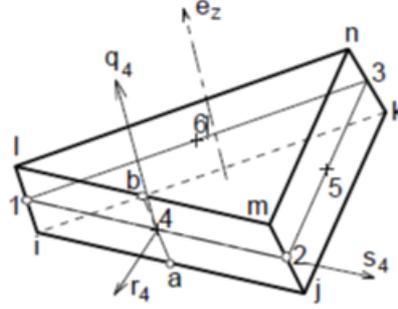


Figure 2.7: Normal vectors and tangential vectors for each triangular side

2.7). A local basis at each mid-side node (4,5 or 6) is expressed as

$$\mathbf{r}_4 = \frac{\mathbf{12} \wedge \mathbf{ab}}{\|\mathbf{12} \wedge \mathbf{ab}\|} \quad (2.126)$$

$$\mathbf{s}_4 = \frac{\mathbf{12}}{\|\mathbf{12}\|} \quad (2.127)$$

$$\mathbf{q}_4 = \mathbf{r}_4 \wedge \mathbf{s}_4 \quad (2.128)$$

with $\mathbf{12}$ is the vector connecting the middle point 1 to the middle point 2, see Figure 2.7. The area surface A_4 formed by the points i-j-m-l can be evaluated using their coordinates in the above defined frame

$$A_4 = ((s_m^{(4)} - s_i^{(4)})(q_l^{(4)} - q_j^{(4)}) + (s_j^{(4)} - s_l^{(4)})(q_m^{(4)} - q_i^{(4)}))/2 \quad (2.129)$$

We can then define the rotation θ_4 like

$$\theta_4 = w_{,q_4} \text{ with } \begin{bmatrix} w_{,s_4} \\ w_{,q_4} \end{bmatrix} = \frac{1}{2A_4} \begin{bmatrix} q_j^{(4)} - q_l^{(4)} & q_m^{(4)} - q_i^{(4)} & q_l^{(4)} - q_j^{(4)} & q_i^{(4)} - q_m^{(4)} \\ s_l^{(4)} - s_j^{(4)} & s_i^{(4)} - s_m^{(4)} & s_j^{(4)} - s_l^{(4)} & s_m^{(4)} - s_i^{(4)} \end{bmatrix} \begin{bmatrix} w_i^{(4)} \\ w_j^{(4)} \\ w_m^{(4)} \\ w_l^{(4)} \end{bmatrix} \quad (2.130)$$

Precisely, it is possible to find the latter relation by analogy with a conventional shape in the i-j-m-l quadrilateral : 's' takes the place of 'x' and 'q' of 'y' while 'r' corresponds to 'z' and 'w' is the translation in the normal surface defined by the nodes i,j,m,l. We are therefore interested in the derivatives $w_{,x}$ and $w_{,y}$ while the usual interpolation in the parent rectangular shape involves the parameter ξ and η ranging from -1 to +1

$$w = \frac{1}{4}(1-\xi)(1-\eta)w_i^{(4)} + \frac{1}{4}(1+\xi)(1-\eta)w_j^{(4)} + \frac{1}{4}(1+\xi)(1+\eta)w_m^{(4)} + \frac{1}{4}(1-\xi)(1+\eta)w_l^{(4)} \quad (2.131)$$

The derivatives $w_{,\xi}$ and $w_{,\eta}$ at $\xi = 0$ and $\eta = 0$ (position of the node '4' in the middle of the face i-j-m-l) are expressed by the nodal displacements of the nodes of the face.

$$\begin{bmatrix} w_{,\xi} \\ w_{,\eta} \end{bmatrix}_{0,0} = \frac{1}{4} \begin{bmatrix} -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} w_i^{(4)} \\ w_j^{(4)} \\ w_m^{(4)} \\ w_l^{(4)} \end{bmatrix}; \text{ and } \begin{bmatrix} w_{,\xi} \\ w_{,\eta} \end{bmatrix}_{(0,0)} = \mathbf{J}_0 \begin{bmatrix} w_{,s_4} \\ w_{,q_4} \end{bmatrix}_{(0,0)} \quad (2.132)$$

Using these derivatives and the local coordinates of the nodes i, j, m, l the 4 components of the matrix $J_{0(2 \times 2)}$ and those of \mathbf{J}_0^{-1} lead to the expression of θ_4 .

$$\theta_4 = P_4(\mathbf{r}_4 \cdot \mathbf{U}_i) + Q_4(\mathbf{r}_4 \cdot \mathbf{U}_j) - Q_4(\mathbf{r}_4 \cdot \mathbf{U}_m) - P_4(\mathbf{r}_4 \cdot \mathbf{U}_l) \quad (2.133)$$

where $P_4 = \frac{s_i^{(4)} - s_j^{(4)}}{2A_4}$ and $Q_4 = \frac{s_i^{(4)} - s_m^{(4)}}{2A_4}$. The expressions are similar for θ_5 and θ_6 with just some index permutations. The following relations between the three rotations and the translational displacements are then derived :

$$\begin{bmatrix} \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix} = \mathbf{T}_{3 \times 18} \cdot \mathbf{U}_{nodes} \quad (2.134)$$

with \mathbf{U}_{nodes} the nodal displacements of the element and $\mathbf{T}_{3 \times 18}$ a transformation matrix detailed like

$$\mathbf{T}_{3 \times 18} = \begin{bmatrix} P_4 \mathbf{r}_4^T & Q_4 \mathbf{r}_4^T & -P_4 \mathbf{r}_4^T & -Q_4 \mathbf{r}_4^T \\ P_5 \mathbf{r}_5^T & Q_5 \mathbf{r}_5^T & -P_5 \mathbf{r}_5^T & -Q_5 \mathbf{r}_5^T \\ Q_6 \mathbf{r}_6^T & P_6 \mathbf{r}_6^T & -Q_6 \mathbf{r}_6^T & -P_6 \mathbf{r}_6^T \end{bmatrix} \quad (2.135)$$

At this point we can finally write the expression of the curvature as follows

$$\boldsymbol{\kappa} = (\mathbf{B}_\theta \mathbf{T}_{3 \times 18} + \bar{\mathbf{B}}_w^b) \cdot \mathbf{U}_{nodes} \quad (2.136)$$

2.4.4 Assumed transverse shear strain

2.4.4.1 C^0 shell element

To avoid shear locking problems, a special treatment is needed. The adopted methodology is extracted from Boisse [BOI 94]. It consists in interpolating the transverse shear computed at the vertices of the triangle (1-2-3) of Figure 2.8. These transverse shear strains are obtained assuming that the transverse shear is constant in each side and is equal to the value taken at the center of each side. A similar solution was proposed in the paper of Bathe [LEE 04] C^0 shell element is based on the Mindlin kinematics. It is a 3 nodes triangle element 1, 2, 3, and for each node, there are three translations and two rotations. The position of an arbitrary point M is defined as

$$\mathbf{x} = \mathbf{x}_m + \mathbf{z} \quad (2.137)$$

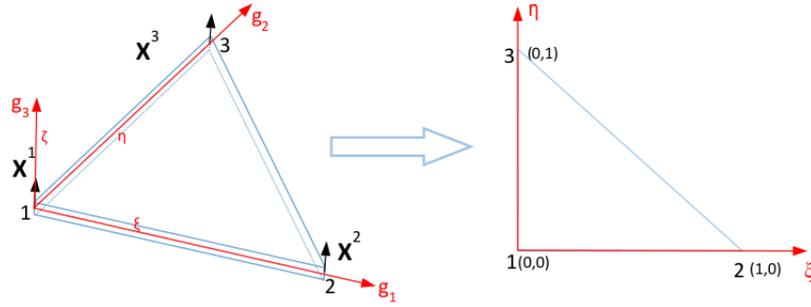


Figure 2.8: Geometry of C^0 triangular element

Where \mathbf{x}_m is the position of the associated point of the mid-surface, and \mathbf{z} is defined based on the pseudo-normal \mathbf{Z} as shown in the figure. \mathbf{x}_m and \mathbf{z} are interpolated as follows, the Einstein summation being applied

$$\begin{aligned}\mathbf{x}_m &= N^i(\xi, \eta)x_m^i \\ \mathbf{z} &= N^i(\xi, \eta)\frac{h^i}{2}\zeta\mathbf{Z}^i\end{aligned}\quad (2.138)$$

Where $N^i \in [1 - \xi - \eta, \xi, \eta]$ is the shape function associated to the i th DOF. The same interpolation is used for the displacement

$$\mathbf{u} = \mathbf{u}_m + \mathbf{u}_z \quad (2.139)$$

Where \mathbf{u}_m is the displacement at the associated point in the mid-surface and \mathbf{u}_z is the displacement given by the pseudo-normal rotation, with

$$\begin{aligned}\mathbf{u} &= N^i(\xi, \eta)(\mathbf{u}_m^i + \mathbf{u}_z^i) \\ \mathbf{u}_z^i &= \frac{h^i}{2}\zeta(\mathbf{x}^i - \mathbf{x}_0^i) = \frac{h^i}{2}\zeta(\mathbf{R}^i - \mathbf{I})\mathbf{x}_0^i\end{aligned}\quad (2.140)$$

where \mathbf{R}^i is the orthogonal tensor vector that transform \mathbf{x}_0^i to \mathbf{x}^i , and \mathbf{I} is the second-order identity tensor. An orthogonal frame $(\mathbf{V}_{10}^i, \mathbf{V}_{20}^i, \mathbf{V}_{30}^i)$ is defined at each node with $\mathbf{V}_{30}^i = \mathbf{x}_0^i$. Assuming that the rotations in the loading step are small enough and denoting by θ the rotation vector at node i ,

$$\begin{aligned}\theta &= \theta_1^i\mathbf{V}_{10}^i + \theta_2^i\mathbf{V}_{20}^i \\ (\mathbf{R}^i - \mathbf{I})\mathbf{x}_0^i &= \theta^i \wedge \mathbf{x}_0^i\end{aligned}\quad (2.141)$$

The displacement interpolation is given as follows

$$\mathbf{u} = N^i(\xi, \eta)\mathbf{u}_m^i + \zeta N^i(\xi, \eta)\frac{h^i}{2}(-\theta_1^i\mathbf{V}_{20}^i + \theta_2^i\mathbf{V}_{10}^i) \quad (2.142)$$

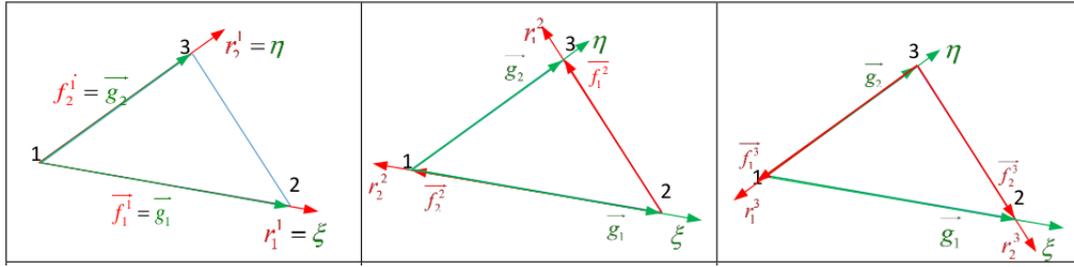


Figure 2.9: Definition of the two covariant basis

According to the interpolation of displacement, strain components can be obtained in different covariant frames. Here, the transverse shear is concerned, and is interpolated directly from the values of those components at nodes

$$\varepsilon_{\alpha 3} = N^1 \varepsilon_{\alpha 3}^1 + N^2 \varepsilon_{\alpha 3}^2 + N^3 \varepsilon_{\alpha 3}^3 ; \alpha = 1, 2 \quad (2.143)$$

In order to calculate the values of nodal transverse shear strains $\varepsilon_{\alpha 3}$, a covariant frame is built from the material coordinates along the element edges and the transverse shear strain $\varepsilon_{\alpha 3}^{i*}$ in the associated contravariant frame is assumed to be constant along the sides connected to node i . In the middle of each edge of the prism, we define a covariant basis $\mathbf{f}_1^i, \mathbf{f}_2^i, \mathbf{f}_3^i$ associated to the contravariant coordinates (r_1^i, r_2^i, r_3^i) using the natural basis of the prismatic element, see Figure (2.9)

$$\mathbf{g}_1 = \frac{\partial \mathbf{x}}{\partial \xi}, \mathbf{g}_2 = \frac{\partial \mathbf{x}}{\partial \eta}, \mathbf{g}_3 = \frac{\partial \mathbf{x}}{\partial \zeta}$$

- In node 1 : $\mathbf{f}_1^1 = \mathbf{g}_1, \mathbf{f}_2^1 = \mathbf{g}_2$ and $r_1^1 = \xi, r_2^1 = \eta$ which gives a transformation matrix between the two frames as $\mathbf{D}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- In node 2 : $\mathbf{f}_1^2 = -\mathbf{g}_1 + \mathbf{g}_2, \mathbf{f}_2^2 = \mathbf{g}_2$ and $r_1^2 = \eta, r_2^2 = 1 - \xi - \eta$ which gives a transformation matrix between the two frames as $\mathbf{D}^2 = \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix}$
- In node 3 : $\mathbf{f}_1^3 = -\mathbf{g}_2, \mathbf{f}_2^3 = -\mathbf{g}_2 + \mathbf{g}_1$ and $r_2^3 = \xi, r_1^3 = 1 - \xi - \eta$ which gives a transformation matrix between the two frames as $\mathbf{D}^3 = \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix}$

The transverse shear strain components $\varepsilon_{\alpha 3}^{i*}$ in $\mathbf{f}_1^i, \mathbf{f}_2^i, \mathbf{f}_3^i$ along the sides can be expressed as

$$\varepsilon_{\alpha 3}^{i*} = \frac{1}{2} \left(\frac{\partial \mathbf{u}}{\partial \zeta} \mathbf{f}_i^\alpha + \frac{\partial \mathbf{u}}{\partial r_\alpha^i} \mathbf{g}_3 \right) |_m \quad \alpha = 1, 2 \text{ and } m = 4, 5, 6 \quad (2.144)$$

The derivative $\frac{\partial \mathbf{u}}{\partial \zeta}$ in the middle node m is taken as the average of that derivative in nodes

i and $i + 1$

$$\begin{aligned}\frac{\partial \mathbf{u}}{\partial \zeta}|_m &= \frac{1}{2} \left(\frac{\partial \mathbf{u}}{\partial \zeta}|_i + \frac{\partial \mathbf{u}}{\partial \zeta}|_{i+1} \right) \\ &= \frac{1}{4} h^i (-\theta_1^i \mathbf{V}_{20}^i + \theta_2^i \mathbf{V}_{10}^i) + \frac{1}{4} h^{i+1} (-\theta_1^{i+1} \mathbf{V}_{20}^{i+1} + \theta_2^{i+1} \mathbf{V}_{10}^{i+1})\end{aligned}\quad (2.145)$$

and one can easily demonstrate that

$$\begin{aligned}\frac{\partial \mathbf{u}}{\partial r_1^i} &= \mathbf{u}^{i+1} - \mathbf{u}^i \\ \frac{\partial \mathbf{u}}{\partial r_2^i} &= \mathbf{u}^{i-1} - \mathbf{u}^i\end{aligned}\quad (2.146)$$

The nodal transverse shear strain in $\mathbf{f}_1^i, \mathbf{f}_2^i, \mathbf{f}_3^i$ can be transformed with the expressions detailed in [BOI 94] by the following form

$$2\boldsymbol{\varepsilon}_{\alpha 3}^{i*} = \mathbf{C}^i \cdot \mathbf{U}_{tria3} \quad \text{and} \quad 2\boldsymbol{\varepsilon}_{\alpha 3}^i = \mathbf{D}^i 2\boldsymbol{\varepsilon}_{\alpha 3}^{i*} \quad (2.147)$$

The shear strain in the covariant base is then given as the combinations of the three shear strain given in each node i

$$2\boldsymbol{\varepsilon}_{\alpha 3} = \sum N^i \mathbf{D}^i \mathbf{C}^i \cdot \mathbf{U}_{tria3} \quad (2.148)$$

The transverse shear strain components in the cartesian basis and at the barycenter of the triangle are interpolated and re-expressed in function of the dofs of the triangle

$$2\boldsymbol{\varepsilon}_{\alpha 3} = \mathbf{J}_{2 \times 2}^{-1} \sum N^i \mathbf{D}^i \mathbf{C}^i \cdot \mathbf{U}_{tria3}, \quad (2.149)$$

where $\mathbf{J}_{2 \times 2}^{-1}$ is the middle surface plane Jacobian matrix and

$$\mathbf{U}_{tria3} = [u^1, v^1, w^1, \theta_1^1, \theta_2^1, u^2, v^2, w^2, \theta_1^2, \theta_2^2, u^3, v^3, w^3, \theta_1^3, \theta_2^3]^T$$

From now on, the next step is to link \mathbf{U}_{tria3} to \mathbf{U}_{nodes}

2.4.4.2 Linking the triangle to the prism

Like in the in-plan shear part, the passage of this triangular element to the prismatic element can be expressed by the displacements of the nodes of the prism. The displacement of M (arbitrary point in the edge i of prism) $\mathbf{u}_M^i, i = 1, 2, 3$ can be interpolated by the displacements of the two nodes (top and bottom) with relation

$$\begin{aligned}\mathbf{u}_M^i &= \frac{1}{2} (1 - \zeta) \mathbf{u}_b^i + \frac{1}{2} (1 + \zeta) \mathbf{u}_t^i \\ &= \frac{1}{2} (\mathbf{u}_b^i + \mathbf{u}_t^i) + \frac{1}{2} (\mathbf{u}_t^i - \mathbf{u}_b^i) \zeta \\ &= \mathbf{u}^i + \mathbf{u}_R^i\end{aligned}\quad (2.150)$$

Where \mathbf{u}^i is the displacement of point i ($i = 1, 2, 3$), \mathbf{u}_R^i is the displacement due to the relative rotation of the prismatic edge which can also be described

$$\mathbf{u}_R^i = \frac{h^i}{2} \zeta \theta^i \wedge \mathbf{x}_0^i \text{ with } \theta^i = \theta_1^i \mathbf{V}_{10}^i + \theta_2^i \mathbf{V}_{20}^i \quad (2.151)$$

The rotations θ_1^i, θ_2^i of the points $i = 1, 2, 3$ can be expressed in function of nodal displacements of the prism with the equations

$$\begin{aligned} \theta_1^i &= -\frac{1}{2h^i} (\mathbf{u}_t^i - \mathbf{u}_b^i) \mathbf{V}_{20}^i \\ \theta_2^i &= -\frac{1}{2h^i} (\mathbf{u}_t^i - \mathbf{u}_b^i) \mathbf{V}_{10}^i \end{aligned} \quad (2.152)$$

with the vectors $\mathbf{V}_{10}^i, \mathbf{V}_{20}^i, \mathbf{V}_{30}^i$ defined as follows

$$\begin{aligned} \mathbf{V}_{30}^i &= \frac{\mathbf{x}_t^i - \mathbf{x}_b^i}{\|\mathbf{x}_t^i - \mathbf{x}_b^i\|} \\ \mathbf{V}_{20}^i &= \frac{(\mathbf{x}^{i+1} - \mathbf{x}^i) \wedge \mathbf{V}_{30}^i}{\|(\mathbf{x}^{i+1} - \mathbf{x}^i) \wedge \mathbf{V}_{30}^i\|} \\ \mathbf{V}_{10}^i &= \mathbf{V}_{20}^i \wedge \mathbf{V}_{30}^i \end{aligned} \quad (2.153)$$

Hence, the relation of the displacements of triangular element and the displacements of prism is linked with a transformation matrix \mathbf{T}_{ts} detailed like this

$$\mathbf{U}_{tria3} = \mathbf{T}_{ts} \cdot \mathbf{U}_{nodes} \quad (2.154)$$

where

$$\mathbf{T}_{ts} = \begin{bmatrix} \mathbf{I}_1 & \mathbf{I}_2 & \mathbf{I}_3 & \mathbf{I}_1 & \mathbf{I}_2 & \mathbf{I}_3 \\ \alpha_1 & 0 & 0 & -\alpha_1 & 0 & 0 \\ \mathbf{I}_1 & \mathbf{I}_2 & \mathbf{I}_3 & \mathbf{I}_1 & \mathbf{I}_2 & \mathbf{I}_3 \\ 0 & \alpha_2 & 0 & 0 & -\alpha_2 & 0 \\ \mathbf{I}_1 & \mathbf{I}_2 & \mathbf{I}_3 & \mathbf{I}_1 & \mathbf{I}_2 & \mathbf{I}_3 \\ 0 & 0 & \alpha_3 & 0 & 0 & -\alpha_3 \end{bmatrix} \quad (2.155)$$

with

$$\begin{aligned} \mathbf{I}_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \mathbf{I}_2 &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \mathbf{I}_3 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \\ \alpha_i &= \frac{1}{h_i} \begin{bmatrix} \mathbf{V}_{20}^{Ti} \\ -\mathbf{V}_{10}^{Ti} \end{bmatrix} \end{aligned} \quad (2.156)$$

The transverse shear stain matrix is then written as follows

$$\bar{\mathbf{E}}_{SH} = \mathbf{J}_{2 \times 2}^{-1} \sum N^i \mathbf{D}^i \mathbf{C}^i \cdot \mathbf{T}_{ts} \cdot \mathbf{U}_{nodes} \quad (2.157)$$

Now $\bar{\mathbf{E}}_{SH}$ is the shear strain at the element middle surface. To obtain the shear strain in all element thickness, the latter is multiplied by the well known Reissner function as follows

$$\mathbf{E}_{SH} = \frac{5}{6} (1 - \zeta^2) \bar{\mathbf{E}}_{SH} \quad (2.158)$$

2.4.5 Pinching Strain and seventh node

The normal strain is simply given as follows

$$\mathbf{E}_P = [0 \quad 0 \quad \mathbf{m}_z^T] \mathbf{U}_{nodes} \quad (2.159)$$

Combining the different equations (2.99), (2.136), (2.159) and (2.158) the deformation-displacement matrix \mathbf{B} of the SB7 element can be derived.

As with the SB9, the additional central node of the SB7 is endowed with only one translation degree of freedom in the element thickness direction ζ . The enhanced normal displacement in the thickness direction is written as follows

$$u_\zeta(\xi, \eta, \zeta) = u_\zeta^H(\xi, \eta, \zeta) + (1 - \zeta^2) u_\zeta^7 \quad (2.160)$$

u_ζ^7 being the one and only relative displacement of the seventh node, in through the thickness direction \mathbf{g}_3 , u_ζ^H the normal displacement of the six nodes element considered alone. A new column is then added to the deformation-displacement standard matrix \mathbf{B}^H making it a 6 by 19 matrix $\bar{\mathbf{B}}$

$$\bar{\mathbf{B}} = [\mathbf{B}^H \quad \mathbf{B}^7] \quad (2.161)$$

with

$$\mathbf{B}^7 = [0 \quad 0 \quad -\frac{4}{h}\zeta \quad 0 \quad 0 \quad 0]^T$$

Where h is the element average thickness. The pinching strain is then written as follows

$$E_{zz} = \bar{\mathbf{B}}_3^H \mathbf{u}_z^H - \frac{4}{h} \zeta u_\zeta^7 \quad (2.162)$$

$\bar{\mathbf{B}}_3^H = \mathbf{m}_z^T$ being the third line of matrix \mathbf{B}^H .

2.4.6 Stabilization of twist mode

Now the reduced integration technique utilized with the SB7 introduces a twist hourglass mode. To alleviate this mode it's important to introduce a stabilization term as it has been

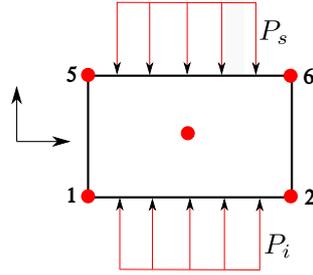


Figure 2.10: projected view of the SB9 under pressure

done with the SB9 element presented above.

$$\begin{aligned}
 \mathbf{h}_1 &= [0, 0, -1, 0, 0, -1]^T \\
 \mathbf{h}_2 &= [0, -1, 0, 0, -1, 0]^T \\
 \gamma_k &= \frac{1}{2}(\mathbf{h}_k - (\mathbf{h}_k^T \cdot \mathbf{x}_j) \cdot \mathbf{m}_j) \quad j = 1, 2, 3
 \end{aligned} \tag{2.163}$$

We can then define a stabilization matrix \mathbf{B}^{Stab} as follows

$$\mathbf{B}^{\text{Stab}} = J_{33}^{-1} \begin{bmatrix} \gamma_1^T & 0 \\ 0 & \gamma_2^T \end{bmatrix} \tag{2.164}$$

and a stabilization stress given like

$$\mathbf{S}^{\text{Stab}} = \mathbf{D}^{\text{Stab}} \cdot J_{33}^{-1} \begin{bmatrix} \gamma_1^T & 0 \\ 0 & \gamma_2^T \end{bmatrix} \mathbf{d}, \tag{2.165}$$

with

$$\mathbf{D}^{\text{Stab}} = \frac{1}{3} \alpha_{stab} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \tag{2.166}$$

with $\alpha_{stab} = 0.01 \sum_{i=1}^{npg} \sqrt{\frac{\sigma_{33}^2}{E_{33}^2} \bar{\omega}_i}$ a stabilization coefficient and $\bar{\omega}_i$ Gauss-lobatto weight.

2.5 Equivalent generalized nodal pressure forces

The advantage of having real extra dof instead of simple parameters is the possibility to physically act on them. For example, the nodal forces equivalent to a normal pressure are prescribed at the vertices but also on the extra node in order to get the proper normal stress distribution. This has been previously done for the element Q5TTS [SAN 11] and a similar method is presented here for the SB7 and SB9 solid-shell elements. To easily find the distribution of the required forces at each node, let us consider an element (Fig. 2.10) on which are defined two normal pressures P_s in the upper face and P_i in the lower face. From equation (2.79), assuming a small change of direction of covariant vector \mathbf{g}_3

we can deduce the variation of the actual normal strain, in the Ahmad base frame as follows

$$\delta\varepsilon_{33} = \frac{\delta u_{add}^+ - \delta u_{add}^-}{h} - 4\zeta \frac{\delta u_{add}}{h} \quad (2.167)$$

where δu_{add}^+ and δu_{add}^- are respectively the variation of the average normal displacement of the upper and lower face of the element, h the element average thickness and δu_{add} the variation of the ninth node displacement [BAS 12]. Assuming the pinching stress σ_{33} is linear through the element thickness

$$\sigma_{33} = \frac{1}{2}(\zeta - 1)P_i - \frac{1}{2}(\zeta + 1)P_s \quad (2.168)$$

the virtual equilibrium condition in the thickness direction can then be written and the equivalent nodal force distribution deduced as follows

$$\begin{aligned} \sum_{i=1}^8 F_i \delta u_i + F_{add} \delta u_{add} &= \int_{S_0} (P_i - P_s) \delta u_i dS_0 \\ &+ \int_{\Omega_0} \delta\varepsilon_{33} \cdot \sigma_{33} d\Omega_0 \end{aligned} \quad (2.169)$$

$$\begin{aligned} F_i &= \frac{1}{n} P_i A_i, & i &= 1, \dots, n \\ F_j &= \frac{1}{n} P_s A_s, & j &= n + 1, \dots, 2n \\ F_{add} &= \frac{2}{3} (P_s A_s - P_i A_i) \end{aligned} \quad (2.170)$$

with A_i and A_s being the element lower and upper surface area. Note that $n = 3$ for the prismatic element and $n = 4$ for the cube element. It is well known that solid-shells with only one layer of elements over the thickness are not able to reproduce a transverse normal stress state which is equal to the applied facial pressures. The new formulation herein (2.170) improves this lack, so the normal stress gives the accurate values corresponding to the applied pressures in the boundaries with only one element in the thickness direction.

2.6 Static equilibrium

In order to solve the nonlinear variational equation, we use a Newton-Raphson iterative method through a sequence of linearization. Assuming that the external energy is displacement independent, we only give details of the linearization of the internal energy.

$$L[\delta\pi_{\text{int}}] = \int_{\Omega_0} (\Delta \mathbf{S} : \delta \mathbf{E} + \mathbf{S} : \Delta(\delta \mathbf{E})) d\Omega_0 \quad (2.171)$$

Equation (2.171) can be split into a material stiffness and a geometrical stiffness.

$$\begin{aligned} \int_{\Omega_0} \Delta \mathbf{S} : \delta \mathbf{E} d\Omega_0 &= \delta \mathbf{U} \left(\int_{\Omega_0} (\mathbf{B}^{\text{ri}})^t \mathbf{D} \mathbf{B}^{\text{ri}} d\Omega_0 + \int_{\Omega_0} (\mathbf{B}^{\text{stab}})^t \mathbf{D} \mathbf{B}^{\text{stab}} d\Omega_0 \right) \Delta \mathbf{U} \\ &= \delta \mathbf{U}^t \left(\mathbf{K}_m^{\text{ri}} + \mathbf{K}_m^{\text{stab}} \right) \Delta \mathbf{U} \end{aligned} \quad (2.172)$$

and

$$\begin{aligned} \int_{\Omega_0} \mathbf{S} : \Delta(\delta \mathbf{E}) d\Omega_0 &= \delta \mathbf{U} \left(\int_{\Omega_0} \mathcal{B}^{\text{ri}} \mathbf{S}^{\text{ri}} d\Omega_0 + \int_{\Omega_0} \mathcal{B}^{\text{stab}} \mathbf{S}^{\text{stab}} d\Omega_0 \right) \Delta \mathbf{U} \\ &= \delta \mathbf{U}^t \left(\mathbf{K}_g^{\text{ri}} + \mathbf{K}_g^{\text{stab}} \right) \Delta \mathbf{U} \end{aligned} \quad (2.173)$$

2.6.1 For SB9 finite element

Material and geometric stiffness

$$\mathbf{K}_m^{\text{ri}} = 4J_0 \int_{-1}^1 (\mathbf{B}^0 + \zeta \mathbf{B}^\zeta + \zeta^2 \mathbf{B}^{\zeta^2})^t \mathbf{D} (\mathbf{B}^0 + \zeta \mathbf{B}^\zeta + \zeta^2 \mathbf{B}^{\zeta^2}) d\zeta \quad (2.174)$$

$$\mathbf{K}_m^{\text{stab}} = J_0 \left[\mathbf{B}^{\eta t} \mathbf{D} \mathbf{B}^\eta + \mathbf{B}^{\xi t} \mathbf{D} \mathbf{B}^\xi \right] \frac{8}{3} + J_0 \left[\mathbf{B}^{\xi \eta t} \mathbf{D} \mathbf{B}^{\xi \eta} + \mathbf{B}^{\xi \zeta t} \mathbf{D} \mathbf{B}^{\xi \zeta} + \mathbf{B}^{\eta \zeta t} \mathbf{D} \mathbf{B}^{\eta \zeta} \right] \frac{8}{9} \quad (2.175)$$

$$\mathbf{K}_g^{\text{ri}} = J_0 \int_{-1}^1 (\mathcal{B}^0 + \zeta \mathcal{B}^\zeta + \zeta^2 \mathcal{B}^{\zeta^2}) \mathbf{S}^{\text{ri}} d\zeta \quad (2.176)$$

$$\mathbf{K}_g^{\text{stab}} = J_0 \left[\mathcal{B}^\xi \mathbf{S}^\xi + \mathcal{B}^\eta \mathbf{S}^\eta \right] \frac{8}{3} + J_0 \left[\mathcal{B}^{\xi \eta} \mathbf{S}^{\xi \eta} + \mathcal{B}^{\eta \zeta} \mathbf{S}^{\eta \zeta} + \mathcal{B}^{\xi \zeta} \mathbf{S}^{\xi \zeta} \right] \frac{8}{9} \quad (2.177)$$

\mathbf{K}_m^{ri} and $\mathbf{K}_m^{\text{stab}}$ are respectively the integrated stiffness matrix and its stabilization counterpart, \mathbf{K}_g^{ri} and $\mathbf{K}_g^{\text{stab}}$ respectively the geometrical stiffness matrix and its counterpart. J_0 is the determinant of the Jacobian matrix. Note that the stabilization matrices are computed analytically without numerical integration. Details about the components of the geometric stiffness, \mathcal{B}^0 and similar matrices are given in appendix A.

Internal forces

In the same way, one can write the internal force and its stabilization counterpart as

$$\begin{aligned} \mathbf{F}_{int}^{\text{ri}} &= 4J_0 \int_{-1}^1 (\mathbf{B}^0 + \zeta \mathbf{B}^\zeta + \zeta^2 \mathbf{B}^{\zeta^2})^t \mathbf{S} d\zeta \\ \mathbf{F}_{int}^{\text{stab}} &= J_0 \left[\mathbf{B}^{\xi t} \mathbf{S}^\xi + \mathbf{B}^{\eta t} \mathbf{S}^\eta \right] \frac{8}{3} + J_0 \left[\mathbf{B}^{\xi \eta t} \mathbf{S}^{\xi \eta} + \mathbf{B}^{\xi \zeta t} \mathbf{S}^{\xi \zeta} + \mathbf{B}^{\eta \zeta t} \mathbf{S}^{\eta \zeta} \right] \frac{8}{9} \end{aligned} \quad (2.178)$$

2.6.2 For SB7 finite element

Material and geometric stiffness

$$\mathbf{K}_m^{\text{ri}} = J_0 \int_{-1}^1 (\mathbf{B}^{\text{ri}})^T \mathbf{D} \mathbf{B}^{\text{ri}} d\zeta \quad (2.179)$$

$$\mathbf{K}_m^{\text{stab}} = J_0 \int_{-1}^1 (\mathbf{B}^{\text{stab}})^T \mathbf{D} \mathbf{B}^{\text{stab}} d\zeta \quad (2.180)$$

$$\mathbf{K}_g^{\text{ri}} = J_0 \int_{-1}^1 (\mathbf{B}_G)^T \Sigma \mathbf{B}_G d\zeta \quad (2.181)$$

$$\mathbf{K}_g^{\text{stab}} = 0 \quad (2.182)$$

\mathbf{K}_m^{ri} and $\mathbf{K}_m^{\text{stab}}$ are respectively the integrated stiffness matrix and its stabilization counterpart, \mathbf{K}_g^{ri} and $\mathbf{K}_g^{\text{stab}}$ respectively the geometrical stiffness matrix and its counterpart. J_0 is the determinant of the Jacobian matrix.

Internal forces

In the same way, one can write the internal force and its stabilization counterpart as

$$\begin{aligned} \mathbf{F}_{\text{int}}^{\text{ri}} &= J_0 \int_{-1}^1 \mathbf{B}^{\text{ri}T} \mathbf{S} d\zeta \\ \mathbf{F}_{\text{int}}^{\text{stab}} &= J_0 \left(\int_{-1}^1 \mathbf{B}^{\text{stab}T} \mathbf{D}^{\text{stab}T} \mathbf{B}^{\text{stab}} d\zeta \right) \mathbf{U} \end{aligned} \quad (2.183)$$

2.7 Elements Inertia Matrices

For a finite element of n nodes, the consistent mass matrix is written like this

$$\mathbf{M} = \int_V \rho [\mathbf{N}]^T [\mathbf{N}] dV, \quad (2.184)$$

with

$$[\mathbf{N}] = \begin{bmatrix} N_1 & 0 & 0 & \dots & N_n & 0 & 0 \\ 0 & N_1 & 0 & \dots & 0 & N_n & 0 \\ 0 & 0 & N_1 & \dots & 0 & 0 & N_n \end{bmatrix} \quad (2.185)$$

For SB9 and SB7, since the last node has only one degree of freedom, we will write the $[\mathbf{N}]$ matrix as follows:

$$[\mathbf{N}] = \begin{bmatrix} N_1 & 0 & 0 & \dots & N_{n-1} & 0 & 0 & 0 \\ 0 & N_1 & 0 & \dots & 0 & N_{n-1} & 0 & 0 \\ 0 & 0 & N_1 & \dots & 0 & 0 & N_{n-1} & N_n \end{bmatrix} \quad (2.186)$$

where N_i , $i = 1, 2, \dots, n-1$ are the shape functions of the vertex nodes and $N_n = 1 - \zeta^2$ is the shape function associated with the additional node.

Element inertia of the SB9 element

The matrix $[\mathbf{N}]$ of equation (2.186) is applied into the equation (2.184) to give the element inertia matrix.

Element inertia of the SB7 element

It is the same as the SB9, but lets give some details.

- For $i, j \in \{1, 2 \dots 6\}$

A typical element \mathbf{M}_{ij} of the element inertia of the SB7 is given as follows

$$\begin{aligned}
 & \rho c \int_A \int_{-1}^{+1} N_i N_j d\zeta dA \\
 &= \frac{\rho c}{4} \int_{-1}^{+1} (1 + \zeta_i \zeta) (1 + \zeta_j \zeta) d\zeta \int_A L_i L_j dA \\
 &= \frac{\rho c}{4} 2 \left(1 + \frac{1}{3} \zeta_i \zeta_j\right) \times \begin{cases} \frac{A}{6} & L_i = L_j \\ \frac{A}{12} & L_i \neq L_j \end{cases}
 \end{aligned} \tag{2.187}$$

Where A is the area of the triangular cross-section, c the height of the prismatic element and $L_i \in \{\xi, \eta, 1 - \xi - \eta\}$. Using this results gives the following inertia matrix

- For $i = 7$ and $j \in \{1, 2 \dots 6, 7\}$

$$\begin{aligned}
 \mathbf{M}_{7j} &= \frac{\rho c}{2} \int_{-1}^{+1} (1 + \zeta_j \zeta) (1 - \zeta^2) d\zeta \int_A L_j dA, \quad j \in \{1, 2 \dots 6\} \\
 \mathbf{M}_{77} &= \frac{\rho c A}{2} \int_{-1}^{+1} (1 - \zeta^2) (1 - \zeta^2) d\zeta
 \end{aligned} \tag{2.188}$$

2.8 Conclusion

In this chapter we have formulated two new solid-shell finite elements called SB9 and SB7. Although inspired by the SB9 γ 25 developed by Bruno Bassa et al [BAS 12] in their way of adding an extra node in the middle of a hexahedral element, the SB9 is a completely different formulation from the SB9 γ 25. Indeed, the new finite element is written in total Lagrangian formulation, which eliminates the need for a local corotational frame to ensure objective modeling. This new element uses an assumed transverse shear strain with four control points per component instead of the 2 control points per component as used in the SB9 γ 25. This eliminates the need for the Reissner function in the transverse shear interpolation. In addition, the assumed natural strain method is also used in the pinch strain to eliminate trapezoidal locking. The polynomial decomposition of the inverse Jacobian is a plus, which allows to take more precisely the geometrical shape

of the element and thus to improve its accuracy. In addition, the new automatic and user-independent stabilization (depends only on the problem) is analytically integrated. The SB7, on the other hand, is a revision of the SB7 γ 19 with a slight modification of the stabilization coefficient. We have also used the assumed natural strain in the pinch strain to eliminate trapezoidal locking. However the SB7 is formulated in a co-rotational updated Lagrangian formulation due to the fact that this element incorporate some features from the DKT6 element. This is necessary to enhanced the element strain and avoid other locking phenomena.

In the next chapter, performance of the two solid-shell elements are investigated with various linear and nonlinear tests.

Chapter 3

Numerical validations

Contents

3.1 Introduction	81
3.2 Linear elastic studies	82
3.2.1 Patch test	82
3.2.1.1 Membrane patch test	82
3.2.1.2 Bending patch test	83
3.2.2 Clamped square plate	83
3.2.3 Circular clamped plate	84
3.2.4 Under pressure cylinder	86
3.2.5 Spherical shell with 18° hole	87
3.2.6 Pinched cylinder	89
3.3 Geometric nonlinear elastic studies	91
3.3.1 Bending of a cantilever beam	91
3.3.2 Spherical shell with 18 degree hole	93
3.3.3 Stretched cylinder with free edge	94
3.3.4 Pinched cylindrical shell mounted over rigid diaphragms	95
3.4 Plastic studies	96
3.4.1 Preprocessing of the logarithmic strain space	97
3.4.2 Constitutive model in the logarithmic strain space	98
3.4.2.1 Energy storage and elastic stress response	98
3.4.2.2 Dissipation and plastic flow response	99
3.4.3 Postprocessing	100
3.4.4 Simply supported plate	100
3.4.5 Cook's membrane	104

3.4.6	Stamping of a sheet by a cylindrical punch (Doc aster)	104
3.5	Modal analysis	106
3.5.1	Natural frequencies of a thick cylindrical ring (Doc aster)	106
3.5.2	Free vibration of a compression vane (Doc aster)	108
3.5.3	Free vibration of a corrugated metal sheet (Doc aster)	112
3.5.4	Free thin circular ring (Doc aster)	114
3.5.5	Vibration of a plate in transitory state (Doc aster)	117
3.6	Linear buckling analysis	117
3.6.1	Buckling of a free cylinder under external pressure (Doc aster) . . .	119
3.7	Industrial application : study of a canopy joint	121
3.8	Conclusion	127

3.1 Introduction

In Chapter 2 two solid-shell elements, SB9 and SB7, has been developed. In this chapter, various numerical tests are presented to demonstrate the capabilities of the proposed solid-shell elements. The SB9 elements is tested in linear as well as in nonlinear (geometric and material), but also in modal analysis and linear buckling. The SB7 element in the otherhand, implemented at the end of this thesis, has only been tested in linear and nonlinear (geometric) analysis, due to lack of time. As mentioned in the Introduction of this thesis, the aim of this work is to develop solid-shell element that perform accurately:

1. No locking for incompressible materials
2. Good bending behavior;
3. No locking in the limit of very thin elements
4. Distortion insensitivity
5. Good coarse mesh accuracy
6. Simple implementation of nonlinear constitutive laws
7. Automatic and robust stabilization in nonlinear situations when necessary.

These requirement are tested for both elements. The first two tests intend to show the importance of the enhancement of the normal stress (advanced EAS method), which is possible thank to the additional middle node. The following tests are classical tests available in the literature or in the official documentation of code_aster (Doc aster), see [Ele 20]. They show the performance of the elements in bending problems, in large deformation with or without material non-linearity and quasi-incompressibles materials. Several of these test have been run independently by Pierre Klag, intern at EDF Lab. The different finite elements used as reference for comparison are :

Q1STs : 8-nodes reduced integrated solid-shell element with EAS and ANS by Schwarze [SCH 09]

Q5TTS : 5-nodes quadrilateral with through thickness stress by Sansalone [SAN 11]

SHB8PS : 8-nodes reduced integrated solid-shell with ANS by Abed Meraim and Combescure [ABE 02]

S4R : ABAQUS's four-node shell element models [HIB 98]

MITC4 : 4-nodes degenerated shell element of Dvorking

[DVO 95b] RESS : 8-nodes reduced enhanced solid-shell element of Alves de Souza [DES 07]

REE07 : 8-nodes solid-shell concept based on reduced integration with hourglass stabilization [REE 07]

Gru05 : ANS mixed stress resultant element, Gruttmann and Wagner [WAG 05]

Kas00 : Mixed-enhanced strain element H1/ME9, Kasper and Taylor [KAS 00]

DKT : Discrete Kirchhoff shell element, see [BAT 92]

Coque_3D : 9 nodes, quadrilateral shell element, see [LAU 06] and www.code-aster.org

HEXA20 : 20-nodes hexahedral standard finite element.

3.2 Linear elastic studies

3.2.1 Patch test

The patch test has been originally proposed in the mid-sixties as a simple means to proof convergence of an element. Beside a numerical verification there is also the possibility of a theoretical analysis. The patch test checks, whether a constant distribution of any state variable within an arbitrary element patch (i.e. a distorted mesh) can be represented exactly. It is especially useful for finite element formulations which violate the compatibility condition (and thus cannot be proven to be consistent), such as the ANS, EAS, etc. Nature of the patch test is to verify an element's ability to represent a constant strain/stress field, and thus ensure completeness and an ability to converge in the limit as the element size decreases.

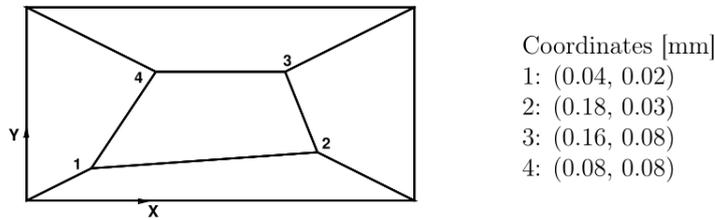


Figure 3.1: Patch test geometry

Figure 3.1 shows a patch of elements that has been extensively used to assess quadrilateral shell elements and hexahedral solid-shell elements. The dimensions are: length $a = 0.24$ mm, width $b = 0.12$ mm, and thickness $t = 0.001$ mm. The mechanical properties of the constitutive material are: elastic modulus $E = 106$ MPa and Poisson's ratio $\nu = 0.25$. As the problem is linear, only 5 integration points are used across the thickness, located at the usual positions of the Lobatto quadrature. Displacements are imposed at the nodes on the contour and computed at the inner nodes to assess the test. For solid-shell formulation to be competitive the membrane patch test must be strictly fulfilled, but if the goal is to have a robust and reliable element the bending patch test must also be passed.

3.2.1.1 Membrane patch test

The peripheral nodes of the plate are imposed a plane displacement such that :

$$u_x = \left(x + \frac{y}{2}\right) \times 10^{-3} \quad u_y = \left(y + \frac{x}{2}\right) \times 10^{-3} \quad (3.1)$$

To allow the contraction due to Poisson effect the lower face's nodes are fixed in the normal direction such that $u_z = 0$. The present element gives the correct results for both displacements of the inner nodes as defined in (Eq. 3.1) and the stress at all integration points are also correct ($\sigma_{xx} = \sigma_{yy} = 1333.3$ MPa and $\sigma_{xy} = 400$ MPa).

3.2.1.2 Bending patch test

The displacement field in this case is :

$$u_x = \pm \left(x + \frac{y}{2}\right) \frac{z}{2} \times 10^{-3} \quad u_y = \pm \left(y + \frac{x}{2}\right) \frac{z}{2} \times 10^{-3} \quad u_z = (x^2 + xy + y^2) \frac{1}{2} \times 10^{-3} \quad (3.2)$$

This displacement is imposed on the boundary nodes of both shell faces. Again the SB9 gives the expected values, for both the displacements applied and the stresses at the Gauss points ($\sigma_{xx} = \sigma_{yy} = \pm 0.667$ MPa and $\sigma_{xy} = \pm 0.200$ MPa).

3.2.2 Clamped square plate

In this example we study the fully clamped plate in the purpose of assessing the element in-plane distortion sensitivity. The length of the plate is $L = 100$ mm and its thickness $t = 1$ mm. The mechanical properties of the material are $E = 10^4$ MPa and $\nu = 0.3$. A central point load of value $P = 16.3527$ N is chosen so that the analytic displacement (Kirchhoff theory) of the plate center is $u_z = 1$ mm. For the sake of simplicity, only one-quarter of the geometry is analysed, using meshes of 2×2 elements. To assess distortion sensitivity the central node of the meshed domain is moved a distance inside the range $0 \leq d \leq 12$ mm as shown in the Figure 3.2. Figure 3.3 gives the evolution of results with the increasing distortion of the SB9 compared with the RESS element [SOU 05], the MITC4 element [DVO 95a] and the Q1ST element [SCH 09]. It can be seen that the distortion sensitivity is similar for all elements compared and that the present element provides results closer to the analytic solution. Now all these elements give sensibly the same results because they all use the ANS method. Now since the SB7 is prismatic, it is less sensitive to this test due to the prismatic form of the element. Therefore, for the most distorted case, the element still gives a good result.

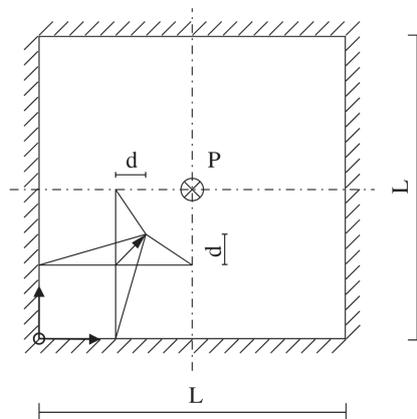


Figure 3.2: Clamped square plate with concentrated load

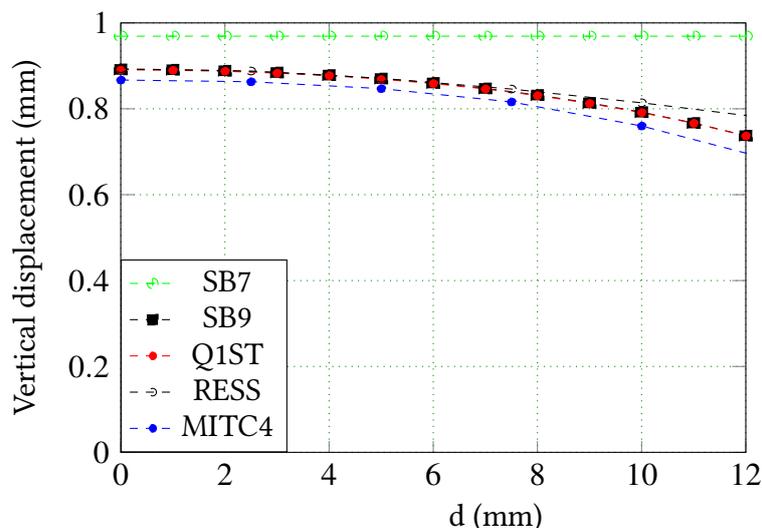
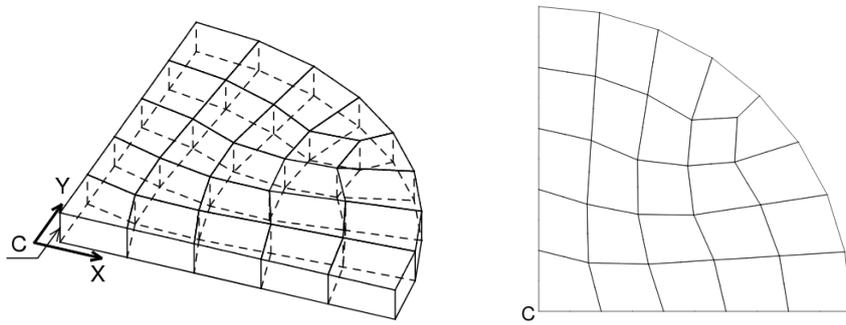


Figure 3.3: Evolution of results with increasing distortion

3.2.3 Circular clamped plate

The goal of this test is to show the main advantage of the ninth node compared to a EAS parameter alone. We consider a disk of radius $R = 100\text{mm}$ and of thickness $t = 1\text{mm}$. The material is isotropic. The Young modulus is $E = 2 \times 10^5\text{MPa}$ and the Poisson ratio $\nu = 0.3$. Two studies have been done. In the first study a uniform normal pressure $P_u = 0.01172\text{N/mm}^2$ is applied in the upper side of the plate. In the second study two normal pressures $P_u = 2 \times 0.01172\text{N/mm}^2$ and $P_l = 0.01172\text{N/mm}^2$ have been respectively applied in the upper and lower faces of the circular plate. The theoretical displacement of the plate center (C in Figure 3.4) is $(U)_C = 1\text{mm}$ according to Kirchhoff's theory. Table 3.1 and table 3.2 show respectively the results for study 1 and study 2 from different finite elements. It is interesting to notice that all different finite elements give good result if we look at the plate center displacement. However the interesting thing to notice is the pinching stress given by the different elements. The SHB8PS and the S4R of abaqus give a nil value for the pinching stress, which is normal since they work with plane stress state. The EAS-only (here used with 5 Gauss points), the SB7 and the SB9 elements are all using a full 3D constitutive law but we can see a non negligible difference in the pinching stress. In table 3.3 and 3.4 the pinching stress for the two elements are detailed in order to have a broad view of what is happening in the shell thickness. The pinching stress is almost constant in the thickness direction for the EAS solid-shell, while for the SB9 or the SB7 element we have a better normal stress which satisfy the boundary condition. This is the main benefice of utilizing a additional node compared to a simple EAS parameter. It allows to split the applied pressure so that the pinching stress is more accurate.

	Q5TTS	S4R	SHB8PS	SB9	SB7	EAS only
Uc	0.9874	0.995	0.995	0.995	0.995	0.995
σ ^{top}	0.0	0.0	0.0	0.0	0.0	-0.0058
σ ^{bottom}	-0.01172	0.0	0.0	-0.01172	-0.01172	-0.0058

Table 3.1: Center displacement and normal stress

Figure 3.4: Circular clamped plate.

	Q5TTS	S4R	SHB8PS	SB9	SB7	EAS only
Uc	0.9874	0.995	0.995	0.995	0.995	0.995
σ ^{top}	-0.01172	0.0	0.0	-0.01172	-0.01172	-0.01757
σ ^{bottom}	-0.02343	0.0	0.0	-0.02343	-0.02343	-0.01757

Table 3.2: Center displacement and normal stress

σ at Gauss points	EAS-only	SB9
1	-0.00585	-0.000000
2	-0.00585	-0.002021
3	-0.00585	-0.005857
4	-0.00585	-0.009693
5	-0.00585	-0.011717

Table 3.3: Pinching stress at Gauss Point : study 1

σ at Gauss points	EAS-only	SB9
1	-0.01757	-0.01171
2	-0.01757	-0.01373
3	-0.01757	-0.01757
4	-0.01757	-0.02140
5	-0.01757	-0.02343

Table 3.4: Pinching stress at Gauss Point : study 2

Thanks to their additional nodes the Q5TTS, the SB7 and the SB9 use a 3D constitutive law and give a pinching stress that is very accurate, on the contrary of the other elements that use the plan stress assumption or only the EAS parameter.

3.2.4 Under pressure cylinder

This test case, like the previous one, is designed to study the pinch response of the SB9 element but this time with a curved structure. We consider a cylinder under internal and external pressure and evaluate the normal stress (pinching stress). For symmetry reasons only one eighth of the cylinder is meshed, the figure 3.5 being obtained by several reflections relative to the axes of symmetries. The results are compared to an approximate analytical solution :

$$\left\{ \begin{array}{l} \sigma_{rr}(r) = \left(P \frac{R}{2t} - \bar{P} \right) - P \frac{R}{2t} \left(\frac{R}{r} \right)^2 \\ P = P_i - P_e \\ \bar{P} = \frac{1}{2} (P_i + P_e) \end{array} \right. \quad (3.3)$$

P_i and P_e are the pressures applied on the inner and outer walls of the cylinder. R and t represent the radius and thickness of the cylinder.

Radius	R	1
Thickness	t	0.01
Young Modulus	E	1×10^7
Poisson Ratio	ν	0.0
internal Pressure	P_i	2
external Pressure	P_e	1

Table 3.5: Geometric and material characteristics

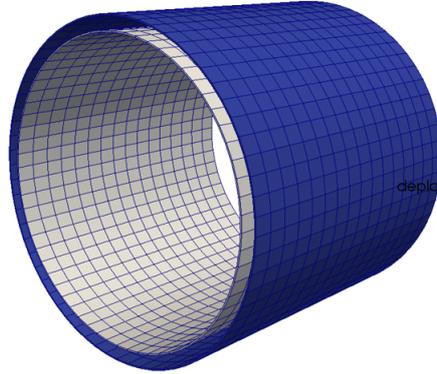


Figure 3.5: Cylinder's mesh

Pts Gauss	1	2	3	4	5
σ_{rr} Analyt	-2.00378	-1.82894	-1.5	-1.17427	-1.00373
σ_{rr} SB9	-1.99915	-1.82648	-1.49915	-1.17183	-0.99915
Error (%)	0.04	0.00	0.05	0.02	0.45

Table 3.6: Pinch stress on the 5 Gauss points

In this test also, we find the static admissibility $\sigma_{rr} = [\boldsymbol{\sigma} \cdot \mathbf{n}] \cdot \mathbf{n} = -p$, where $\boldsymbol{\sigma}$ is the stress field, \mathbf{n} the normal on the face and p the applied pressure. We can thus see the interest of the additional term compared with a EAS parameter alone without enhancing the pinching stress.

3.2.5 Spherical shell with 18° hole

This very popular test case is particularly severe due to the curvature of the geometry, which considerably accentuates the transverse shear and membrane locking. We consider a very thin hemispherical shell (radius $R = 10$, thickness $t = 0.04$) perforated at its top at an opening angle equal to 18° [Figure 3.6], which is subjected to two diametrically opposed pinching forces along the Y axis ($F_y = -2N$ and $F_y = 2N$) and two diametrically opposed extension forces along the X axis ($F_x = -2N$ and $F_x = 2N$). The useful characteristics of the material are Young's modulus ($E = 6.825 \times 10^7$) and Poisson ratio ($\nu = 0.3$). Using the (XZ) and (YZ) planes of symmetry, a quarter of the hemisphere is meshed with two unit forces in the X and Y directions. The boundary conditions are free, however, in order to avoid rigid body movements, the displacement in Z direction of one point is blocked in addition to the conditions relative to the two planes of symmetry. The reference solution given by many authors is **0.094** (see [SCH 09]). A convergence study has been carried out with SB9 in comparison with other types of finite elements of the solid-shell type, on different meshes. The results are reported in the table 3.7. The figure 3.7 also compares the normalized displacements as a function of the number of

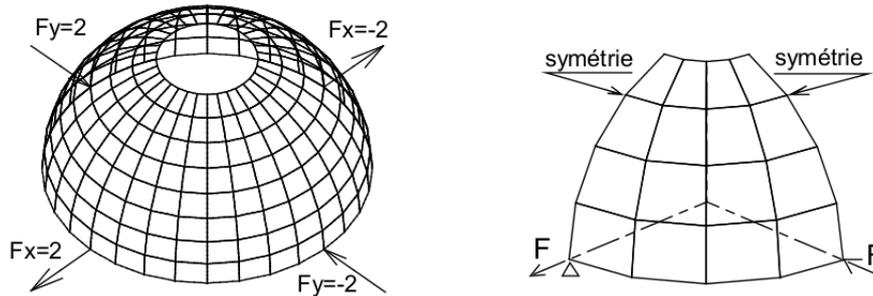


Figure 3.6: Pinched hemisphere

Mesh	4×4	8×8	16×16	32×32
SB9	1.043	1.002	0.993	0.994
Ree07	0.062	0.723	0.919	0.969
Gru05	1.032	0.999	0.993	0.995
Kas00	0.039	0.732	0.989	0.998

Table 3.7: Displacements of the solicited nodes, with SB9 modelization

elements per side, for the different solid-shell finite elements. It can be observed that SB9 and Q1st give the best results for this test. The fact that the SB9 element passes this very severe test demonstrates its ability to withstand the phenomena of membrane and transverse shear locking. This test exhibits the crucial importance of the ANS method to develop shear locking-free element formulations. Concerning the convergence behavior, the SB9 element formulation is among the best if one considers all element formulations here presented. A quick convergence study has also been carried with the SB7 element with two different meshes as one can see in figure 3.8. The results given in table 3.8 shows a good convergence for the SB7 too.

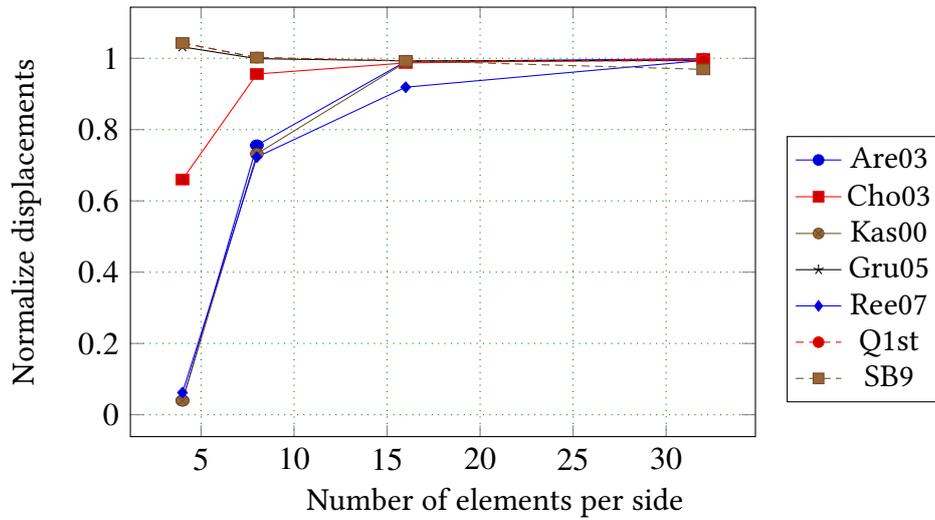


Figure 3.7: Spherical shell with 18° hole, study of convergence.

SB7	Mesh 1	Mesh 2	Ref
U	0.0929	0.0941	0.094
Error (%)	1.07	0.1	

Table 3.8: Displacements of the solicited nodes

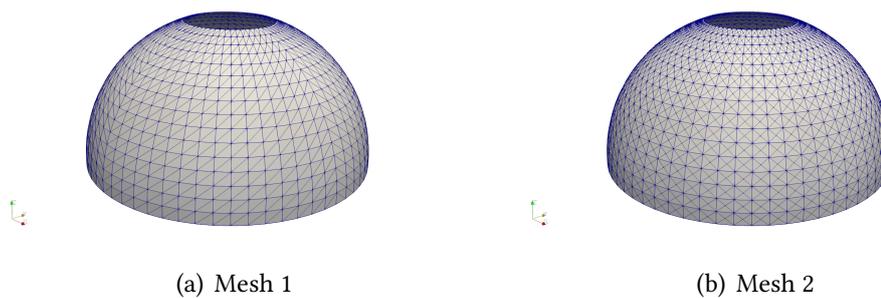


Figure 3.8: The SB7 meshes used for this simulation

3.2.6 Pinched cylinder

The purpose of this test case is almost the same as that of the pinched hemisphere: to confront the element to a well curved environment to see its capacity to resist the phenomena of membrane and transverse shear locking. With the help of symmetries, only one eighth of the pinched cylinder is meshed with different discretizations to see

Rayon	R	4.953
Longueur	L	10.35
Epaisseur	t	0.094
Module d'Young	E	10.5×10^6
Coefficient de poisson	ν	0.3125
Charge appliquée	F	100N

Table 3.9: Geometrical and material characteristics of the cylinder

the effect of refinement on the solution. By stressing the structure with two equal and opposite forces $F=100N$, we try to determine the displacement in the stress direction for the different discretizations and then compare it to the reference displacement: 0.1139 [MAC 85]. The geometrical and material characteristics for this test case are given in the table 3.9. The convergence diagram can be see in figure 3.10. Both the SB7 and SB9 elements converge with mesh refinement. One can also observe that the SB7 elements is a little bit soft compared to the SB9 for this test case which is due to the difference of the Assumed Natural strain used for each element.

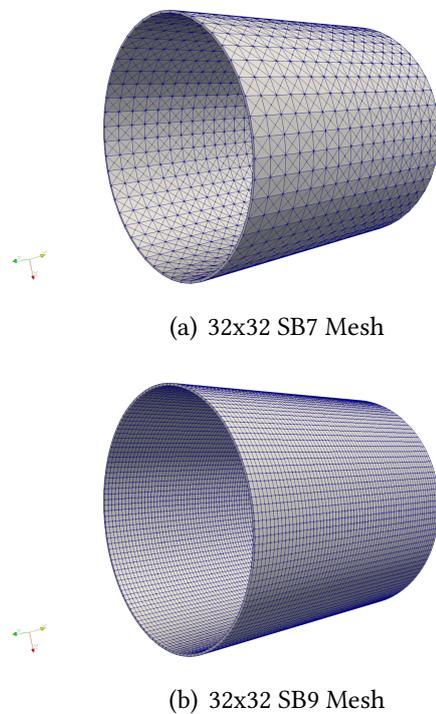


Figure 3.9: The SB7 meshes used for this simulation

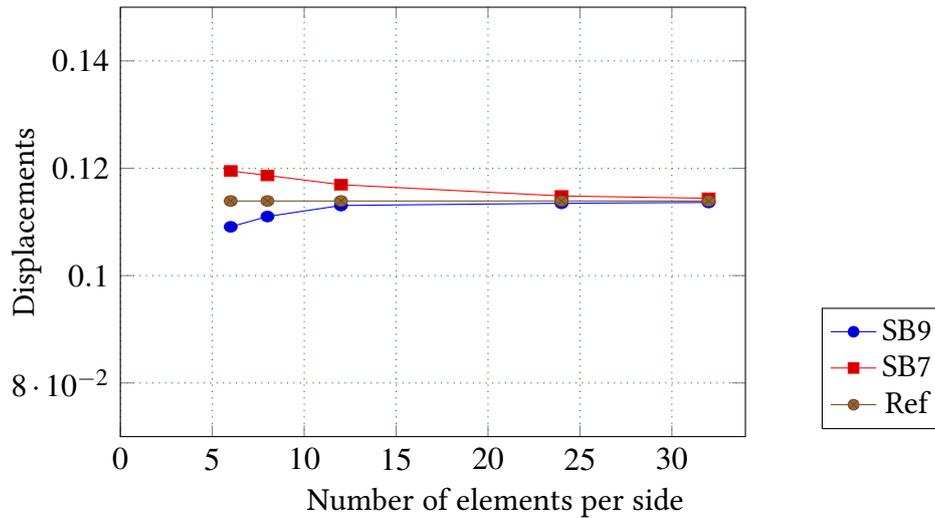


Figure 3.10: Pinched cylinder, study of convergence.

3.3 Geometric nonlinear elastic studies

3.3.1 Bending of a cantilever beam

In this test we investigate the out-of-plane bending of a cantilever beam under a tip load. The problem has been analyzed by many authors [SIM 92a, BÜC 94, MIE 98, HAU 01, FON 05, REE 07]. As shown in figure 3.11, the beam has a side length of $L = 10\text{mm}$ and a rectangular cross section of width $B = 1\text{mm}$ and a thickness $t = 0.1\text{mm}$. The material is isotropic and has a Young modulus $E = 10^7\text{N/mm}^2$ and a Poisson ratio $\nu = 0.3$. The total load is $F_{max} = 40\text{N}$ introduced in ten time increments for the SB9 modelization and 100 times for the SB7 modelization. Since the SB7 element is formulated in a co-rotational updated Lagrangian formulation, it necessitate small time increment to converge correctly. The Beam is discretized by one element along the width and thickness direction and 16 elements along the length. The numerical integration is performed with Gauss points through the thickness direction. A first study has been made to assess the convergence of the two finite elements for this test, using a regular mesh and then a highly distorted mesh. Figure 3.12-a gives the load displacement path for the two different meshes and for the two modelizations. The displacement at convergence is in agreement with the analytic solution $U_z = 7.083\text{mm}$ given in [FRI 62]. The load displacement path is identical for both the regular and distorted mesh for the SB9 and almost the same for the SB7 element, proving that these two solid-shell finite elements are insensitive to mesh distortion.

3. Numerical validations

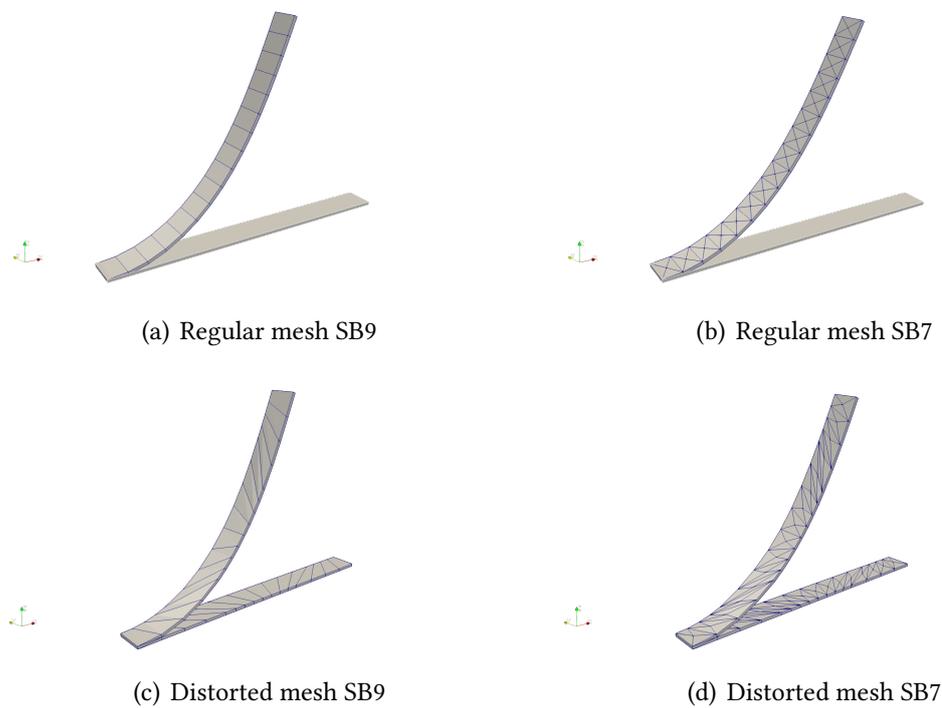


Figure 3.11: Mesh form used for this simulation

In a second time the behaviour of the two elements in the near quasi-incompressible situations is investigated. The regular mesh is kept but this time the Poisson ratio is varied between $\nu = 0.0$ to $\nu = 0.4999$. The elements give good results showing no sign of volumetric locking, see Figure 3.12-b.

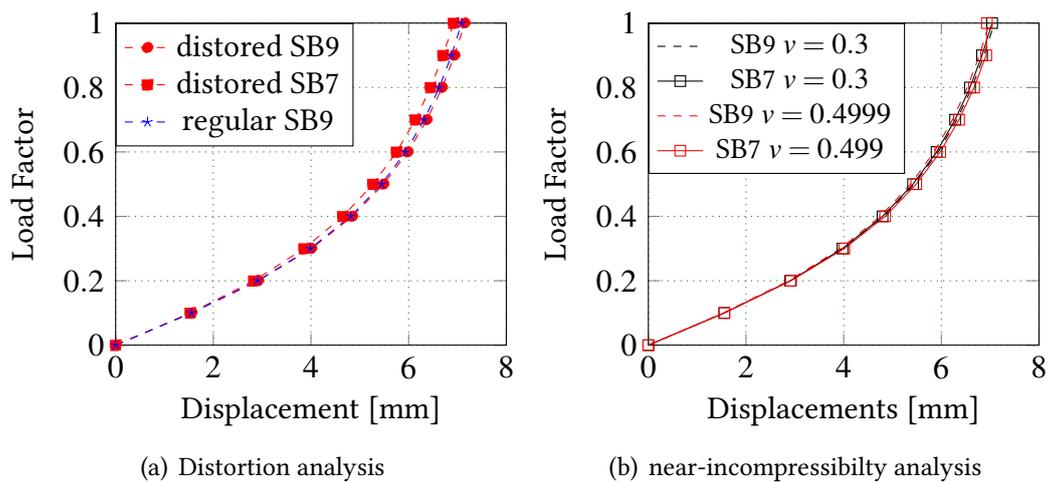
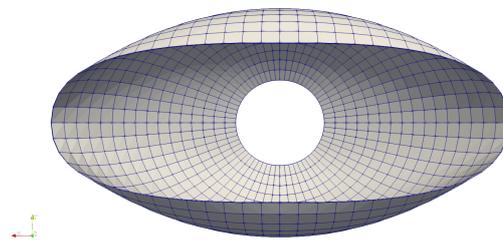
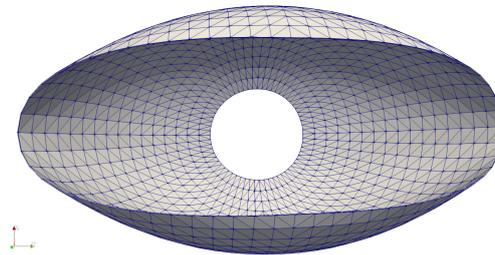


Figure 3.12: Evolution of the cantilever beam's free-edge displacement



(a) SB9 Mesh



(b) SB7 Mesh

Figure 3.13: Pinched hemisphere with hole

3.3.2 Spherical shell with 18 degree hole

In this example the hemispherical shell with an 18 degree circular cut out at it's pole is studied. The shell is loaded by an alternating radial point forces F at 90 degree intervals, see figure 3.13. This is a well known problem as it has been considered by many authors [SIM 90a, SAL 90, SAN 92, SZE 04, REE 07], to name just a few. Thanks to symmetry, only one-quarter of the shell is modeled with a 16×16 mesh. The geometrical parameters used are the same used by [SZE 04] with a radius of $R = 10.0mm$ and a thickness of $t = 0.04mm$ which give a ratio $R/t = 250$. So this is a relatively severe test more likely to exhibit locking effects. The structure is subdued to two concentrated force $F = 100N$ as shown in figure 3.13. This load is high enough to create a geometric nonlinear deformations. Therefore this test allows to observe the reaction of the SB7 and the SB9 in large geometric transformations. The material parameters are the Young modulus $E = 6.825 \times 10^7 N/mm^2$ and the Poisson ratio $\nu = 0.3$. The total number of increment to reach the maximum load is 10. In Figure 3.14, the deformation path of the points A and B are plotted and compared to the values given by Sze [SZE 04]. The results given by the SB9 and SB7 elements is in good agreement with the reference solutions.

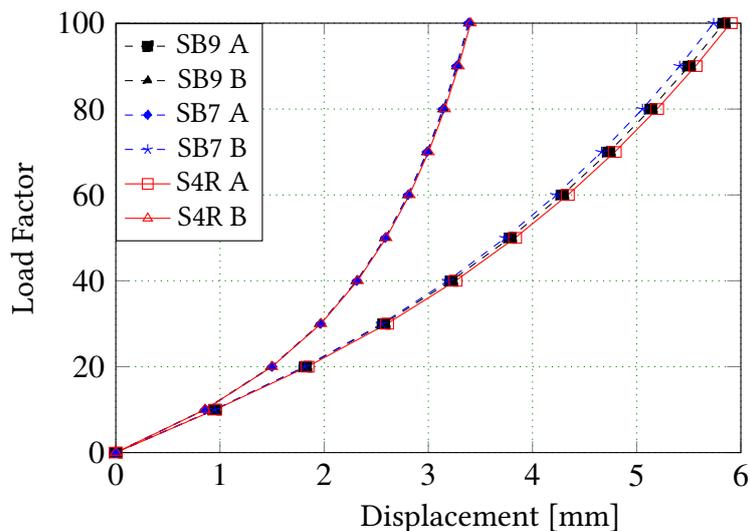


Figure 3.14: Evolution of displacement of hemisphere

3.3.3 Stretched cylinder with free edge

Loaded by two opposite single forces, see figure 3.9, the cylinder undergoes a significantly large rotation combining bending and membrane effects. The geometric is defined by a length $L = 10.35$, a radius $R = 4.953$ and a thickness $T = 0.094\text{mm}$. The material properties are given by the Young modulus $E = 10.5 \times 10^6 \text{N/mm}^2$ and the Poisson ratio $\nu = 0.3125$ and the applied load in each side is $F = 40\text{KN}$. Due to the symmetry of the problem, only one eighth of the system is discretized. In order to investigate the convergence several meshes refinements are tested including 8×12 , 16×24 and 20×30 , with only a single element through the thickness. Using the present formulation of the SB9 the problem is modeled using a 16×24 mesh and the results are compared to those tabulated in Sze et al [SZE 04]. Figure 3.16 shows the results in term of load versus radial displacements at points A, B and C. Point A corresponds to the point under the loading, while point B and C are on the side of the cylinder and undergo horizontal displacements see figure 3.16. As it can be seen in the load-displacement curves, the overall response exhibits two regimes : a first stage dominated by bending effect and characterized by large displacements and rotations, and a second phase dominated by membrane effects, which may cause locking. It's also important to note the snap-through phenomenon arising when the loading reaches the critical value of around 20KN . This can be seen through the displacement reversal that occurs on the load displacement curve of point C. This new formulation of the SB9 element gives results in very good agreement with the reference solution, with no locking effect.

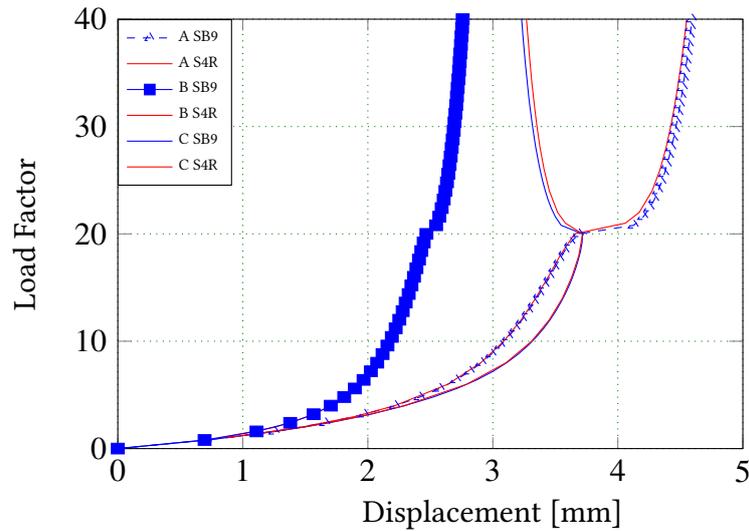


Figure 3.16: Evolution of displacement of stretched cylinder

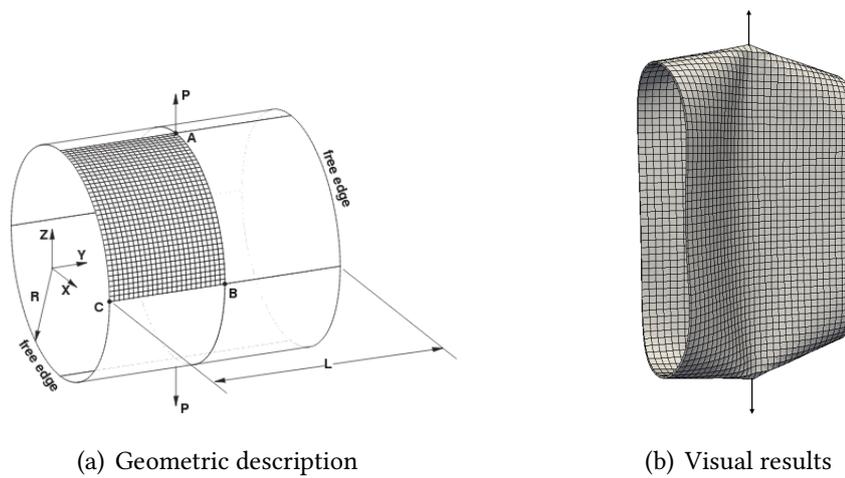


Figure 3.15: Stretched cylinder with free edge

3.3.4 Pinched cylindrical shell mounted over rigid diaphragms

This problem and its variations have been considered by many authors [SAL 90, SAN 92, HAU 98, SZE 04], among others. The cylinder shell is mounted on a rigid diaphragms and is subjected to two pinching forces load $P_{max} = 12000$ as shown is the Figure 3.17-a. The dimensions of the cylinder are a length of $L = 200$, a radius $R = 100$ and a thickness $h = 1$. The material properties are Young Modulus $E = 30 \times 10^3$ and the Poisson ratio $\nu = 0.3$. Thanks to symmetry, one-eighth of the shell is modeled. A commonly employed mesh of 40×40 solid-shell elements is used in this test. Figure 3.18 shows

the displacement of the points A and B, compared with those given in [SZE 04] with the Abaqus S4R element.

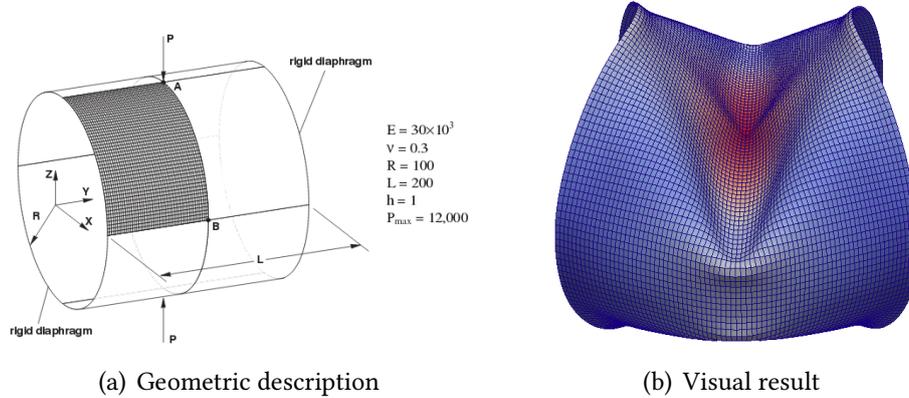


Figure 3.17: Pinched cylindrical shell mounted over rigid diaphragms

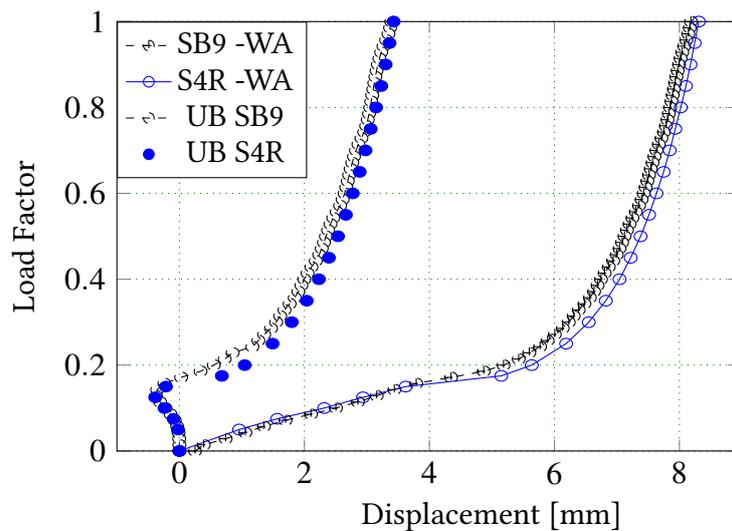


Figure 3.18: Evolution of displacement of pinched cylinder

3.4 Plastic studies

Most of Code_Aster's constitutive laws can be used under the hypothesis of small perturbations, which makes it possible to take the initial configuration as the current geometrical configurations. However, when the deformations become significant (the limit is usually set at 5%), this hypothesis is no longer verified. The notions of particle and partial derivatives are then different, and as a result the incrementally formulated constitutive laws lose their objectivity. In order to restore objectivity, which is therefore

essential to guarantee a good reliability of the result, a large deformation strategy is possible. This is the large deformation model based on a logarithmic measurement of the deformation, with a particular duality stress tensor. A complete presentation of this model can be found in C.Miehe, N.Apel and M.Lambrecht [MIE 04].

3.4.1 Preprocessing of the logarithmic strain space

A key point in the setting up of the logarithmic framework of finite plasticity is the definition of an elastic strain measure $\bar{\mathbf{E}}^e$. As pointed out in [Miehe, 36,37], there do exist several conceptual possibilities for the definition of the elastic strain measure. Following Miehe, we focus on the elementary additive form

$$\bar{\mathbf{E}}^e := \bar{\mathbf{E}} - \bar{\mathbf{E}}^p \quad (3.4)$$

consistent with the approach proposed by Miehe in term of the logarithmic Langrangian total and plastic strains

$$\bar{\mathbf{E}} := \frac{1}{2} \ln \mathbf{C}^{mod} \quad \text{and} \quad \bar{\mathbf{E}}^p := \frac{1}{2} \ln \mathbf{G}^p \quad (3.5)$$

respectively. The logarithmic strain measure $\bar{\mathbf{E}}$ is a function of the current metric \mathbf{C}^{mod} , which in turn is a function of the deformation \mathbf{F}^{mod} , see section 1.5.5 of chapter 1. The easiest way to compute \mathbf{C}^{mod} is

$$\mathbf{C}^{mod} = 2\mathbf{E}^{mod} + \mathbf{I}, \quad (3.6)$$

\mathbf{E}^{mod} being the Green Lagrange strain of the solid-shell element. Of key importance for the subsequent treatment is the sensitivity of the logarithmic strain measure with respect to a change of the deformation. To this end, we introduce the relationships

$$\dot{\bar{\mathbf{E}}} = \mathbb{P} : \dot{\mathbf{F}}^{mod} \quad \text{and} \quad \dot{\mathbb{P}} = \mathbb{L} : \dot{\mathbf{F}}^{mod} \quad (3.7)$$

where, in terms of the fourth- and sixth-order nominal transformation tensors

$$\mathbb{P} := \partial_{\mathbf{F}^{mod}} \bar{\mathbf{E}} \quad \text{and} \quad \mathbb{L} := \partial_{\mathbf{F}^{mod} \mathbf{F}^{mod}}^2 \bar{\mathbf{E}}, \quad (3.8)$$

respectively. These transformation tensors play a central role in the subsequent treatment. Closed-form algorithmic approaches to these types of tensors have recently been outlined in the more general context for the class of Seth–Hill strain measures in [45]. The stress power can then be represented like

$$\mathcal{P}(t) = \mathbf{T}(t) : \dot{\bar{\mathbf{E}}}(t) \quad \text{with} \quad \mathbf{T} := \mathbf{P} : \mathbb{P}^{-1}. \quad (3.9)$$

\mathbf{T} is the Lagrangian stress tensor work-conjugate to the logarithmic strain measure $\bar{\mathbf{E}}$ and \mathbf{P} the first Piola Kirchhoff stress. The symmetric Lagrangian tensors \mathbf{T} and $\bar{\mathbf{E}}$ provide a convenient pair of dual external variables of the local material element associated with the logarithmic strain space.

3.4.2 Constitutive model in the logarithmic strain space

Now we assume a constitutive model of plasticity that is exclusively restricted to the logarithmic strain space. This model is considered as a constitutive box, its input is given by the logarithmic strains $\bar{\mathbf{E}}$ and a set $\mathcal{T} := \{\bar{\mathbf{E}}^p, \dots\}$ of internal variables consisting of the logarithmic plastic strain tensor $\bar{\mathbf{E}}^p$ and some additional hardening variables. The output of the box is the current stress \mathbf{T} dual to the logarithmic strain and the associated elastic–plastic tangent moduli \mathbb{E}^{ep}

$$\{\bar{\mathbf{E}}, \mathcal{T}\} \longrightarrow \text{MODEL} \longrightarrow \{\mathbf{T}, \mathbb{E}^{\text{ep}}\} \quad (3.10)$$

In the continuous setting the tangent moduli exist for the rate-independent theory of plasticity and govern the rate of the stress with respect to rate of the logarithmic strain

$$\dot{\mathbf{T}} = \mathbb{E}^{\text{ep}} : \dot{\bar{\mathbf{E}}}. \quad (3.11)$$

The attractive feature of the constitutive model is that it can preserve the structure of plasticity models of the geometrically linear theory. Thus the model may adopt standard constitutive structures of the small strain theory. This section specifies a constitutive model in the logarithmic strain space with a structure adopted from the geometrically linear theory. We focus on a formulation suitable for the modelling of problems in metal plasticity.

3.4.2.1 Energy storage and elastic stress response

The stress-strain relation can be obtained by defining a local dissipation function with respect to unit volume and using the Clausius Planck inequality

$$\mathcal{D} := \mathbf{T} : \dot{\bar{\mathbf{E}}} - \dot{\psi} \geq 0 \quad (3.12)$$

with $\psi : \mathcal{R}^6 \times \mathcal{R}^6 \times \mathcal{R} \longrightarrow \mathcal{R}$ a free energy function defined like

$$\psi = \psi(\bar{\mathbf{E}} - \bar{\mathbf{E}}^p, \mathbf{A}, \alpha) \quad (3.13)$$

where $\bar{\mathbf{E}}^p$ is the logarithmic plastic strain, \mathbf{A} a symmetric second-order tensor for the description of the kinematic hardening and α a scalar variable that models isotropic hardening. Using the Coleman’s method, one can write the constitutive equation like

$$\begin{aligned} \mathbf{T} &= +\partial_{\bar{\mathbf{E}}} \psi(\bar{\mathbf{E}} - \bar{\mathbf{E}}^p, \mathbf{A}, \alpha) \\ \mathbf{B} &= -\partial_{\mathbf{A}} \psi(\bar{\mathbf{E}} - \bar{\mathbf{E}}^p, \mathbf{A}, \alpha) \\ \beta &= -\partial_{\alpha} \psi(\bar{\mathbf{E}} - \bar{\mathbf{E}}^p, \mathbf{A}, \alpha) \end{aligned} \quad (3.14)$$

The first equation is the constitutive expression for the stresses in the logarithmic strain space. The other equations define *internal forces* $\mathcal{F} := \{\mathbf{T}^p, \mathbf{B}, \beta\}$ dual to the above introduced internal variables. It can be shown that the internal force \mathbf{T}^p that derives the

plastic strain $\bar{\mathbf{E}}^p$ is identical to the stress \mathbf{T} . The reduced dissipation inequality is then rewritten like

$$\mathcal{D} := \mathbf{T} : \dot{\bar{\mathbf{E}}}^p + \mathbf{B} : \dot{\mathbf{A}} + \beta \dot{\alpha} \geq 0 \quad (3.15)$$

Assuming metal plasticity, see [58] for a micromechanical motivation, a fully decoupled representation of the energy function can be written as

$$\psi = \psi^e(\bar{\mathbf{E}} - \bar{\mathbf{E}}^p) + \psi^k(\mathbf{A}) + \psi^i(\alpha) \quad (3.16)$$

A typical example is the quadratic storage function

$$\psi = \frac{1}{2} \|\bar{\mathbf{E}} - \bar{\mathbf{E}}^p\|_{\mathbb{E}}^2 + \frac{k}{2} \|\mathbf{A}\|^2 + \frac{1}{2} \alpha^2 \quad (3.17)$$

where $\|\bar{\mathbf{E}}^e\|_{\mathbb{E}} := \sqrt{\bar{\mathbf{E}}^e : \mathbb{E} : \bar{\mathbf{E}}^e}$ is the norm of elastic strain measure $\bar{\mathbf{E}}^e$ with respect to a constant fourthorder tensor \mathbb{E} that characterizes the macroscopic elasticity moduli. $k \in \mathcal{R}_+$ and $h \in \mathcal{R}_+$ are elastic material parameters associated with the kinematic and isotropic hardening, respectively. For this quadratic function, the stresses and internal forces turn out to be linear functions of the strain-like variables

$$\begin{aligned} \mathbf{T} &= +\mathbb{E} : (\bar{\mathbf{E}} - \bar{\mathbf{E}}^p) \\ \mathbf{B} &= -k\mathbf{A} \\ \beta &= -h\alpha \end{aligned} \quad (3.18)$$

3.4.2.2 Dissipation and plastic flow response

Consider the plastic flow to be constrained by a convex elastic domain in the space of the internal forces

$$\mathcal{E} = \{(\mathbf{T}, \mathbf{B}, \beta) \in \mathcal{R} \mid f(\mathbf{T}, \mathbf{B}, \beta) \leq 0\} \quad (3.19)$$

where $f = f(\mathbf{T}, \mathbf{B}, \beta)$ is a convex level set function. Then a canonical form of the evolution equations for the internal variables is determined by a thermodynamic extremum principle, the well-known principle of maximum plastic dissipation. It yields the normality rules

$$\begin{aligned} \dot{\mathbf{T}}^p &= \gamma \partial_{\mathbf{T}} f(\mathbf{T}, \mathbf{B}, \beta) \\ \dot{\mathbf{A}} &= \gamma \partial_{\mathbf{B}} f(\mathbf{T}, \mathbf{B}, \beta) \\ \dot{\alpha} &= \gamma \partial_{\beta} f(\mathbf{T}, \mathbf{B}, \beta) \end{aligned} \quad (3.20)$$

which determine the evolution of the plastic flow and the hardening variables in terms of gradients of the yield criterion function. Here, the plastic parameter γ is in the rate-independent case determined by the Kuhn–Tucker-type loading–unloading conditions

$$\gamma \geq 0, \quad f \leq 0, \quad \gamma f = 0 \quad (3.21)$$

3.4.3 Postprocessing

Once the stresses and tangent moduli in the logarithmic strain space have been obtained from the constitutive model, equation (3.10), we map them to the nominal stresses and nominal moduli based on straightforward application of the transformation rules introduced in subsection 3.4.1. The combination of equations 3.7 and 3.9 gives

$$\mathbf{P} = \mathbf{T} : \mathbb{P} \quad \text{and} \quad \mathbb{C}^{\text{ep}} = \mathbb{P}^T : \mathbb{E}^{\text{ep}} : \mathbb{P} + \mathbf{T} : \mathbb{L} \quad (3.22)$$

Here, the fourth-order nominal elastic–plastic tangent moduli \mathbb{C}^{ep} govern the sensitivity of the nominal stresses

$$\dot{\mathbf{P}} = \mathbb{C}^{\text{ep}} : \dot{\bar{\mathbf{F}}} \quad (3.23)$$

with respect to the rate of deformation. The symmetric Lagrangian stresses $\mathbf{S} := \mathbf{F}^{-1}\mathbf{P}$ and their associated elastic–plastic tangent moduli \mathbb{C}_L^{ep} are obtained analogously to the above derivation of the nominal tensors. We get the formulations

$$\mathbf{S} = \mathbf{T} : \mathbb{P}_L \quad \text{and} \quad \mathbb{C}_L^{\text{ep}} = \mathbb{P}_L^T : \mathbb{E}^{\text{ep}} : \mathbb{P}_L + \mathbf{T} : \mathbb{L}_L \quad (3.24)$$

in terms of the fourth- and sixth-order Lagrangian transformation tensors \mathbb{P}_L and \mathbb{L}_L , respectively. They appear in the closed form

$$\mathbb{P}_L := 2\partial_{\mathbf{C}^{\text{mod}}}\bar{\mathbf{E}} \quad \text{and} \quad \mathbb{L}_L := 4\partial_{\mathbf{C}^{\text{mod}}}\partial_{\mathbf{C}^{\text{mod}}}\bar{\mathbf{E}}, \quad (3.25)$$

as derivatives of the logarithmic strain measure with respect to the convected current metric \mathbf{C}^{mod} . The Lagrangian elastic–plastic tangent moduli \mathbb{C}_L^{ep} govern the sensitivity of the symmetric second Piola stress

$$\dot{\mathbf{S}} = \mathbb{C}_L^{\text{ep}} : \frac{1}{2}\dot{\mathbf{C}} \quad (3.26)$$

with respect to the Lagrangian rate $\dot{\mathbf{C}}^{\text{mod}}/2$ of deformation.

3.4.4 Simply supported plate

In this example the simply supported plate under pressure is addressed. The problem has been treated by many authors [BET 99, EBE 99, FON 05, HAU 01, REE 07] among others. The geometric parameters of the plate are represented by the side length $L = 508\text{mm}$ and the thickness $t = 2.54$. Thanks to symmetry only one quarter of the plate is represented, Figure 3.19-b being obtained by reflexion. As boundary conditions, the outer lower side edge of the plate is fixed in the vertical direction ($u_3 = 0$), hence the rotation around these axes is possible. A uniform pressure $p = \lambda p_0$ with $p = 0.01\text{N/mm}^2$ is applied. The maximum of the load factor is $\lambda_{\text{max}} = 40$. The material is taken ideally elastic–plastic with $E = 6.9 \times 10^4\text{N/mm}^2$, $\nu = 0.3$ and the yield stress $\sigma_y = 248\text{N/mm}^2$. Figure 3.20 gives the evolution of the normal displacement of the plate center compared to the results given in [REE 07].

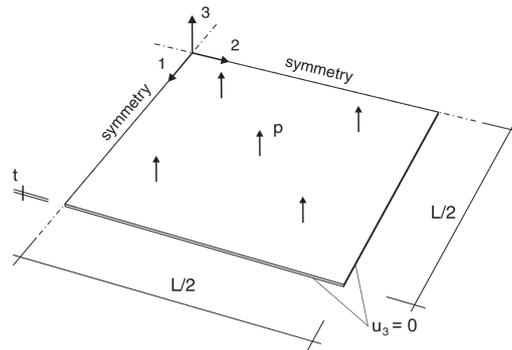


Figure 3.19: Simply supported plate : geometric description

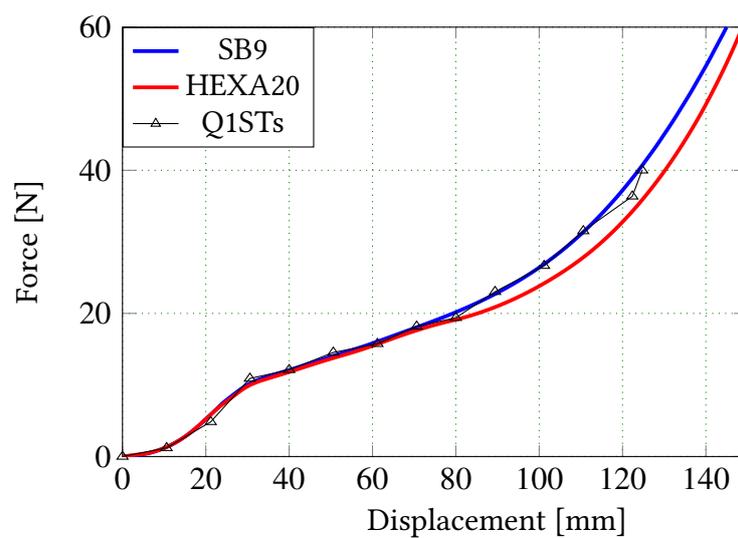
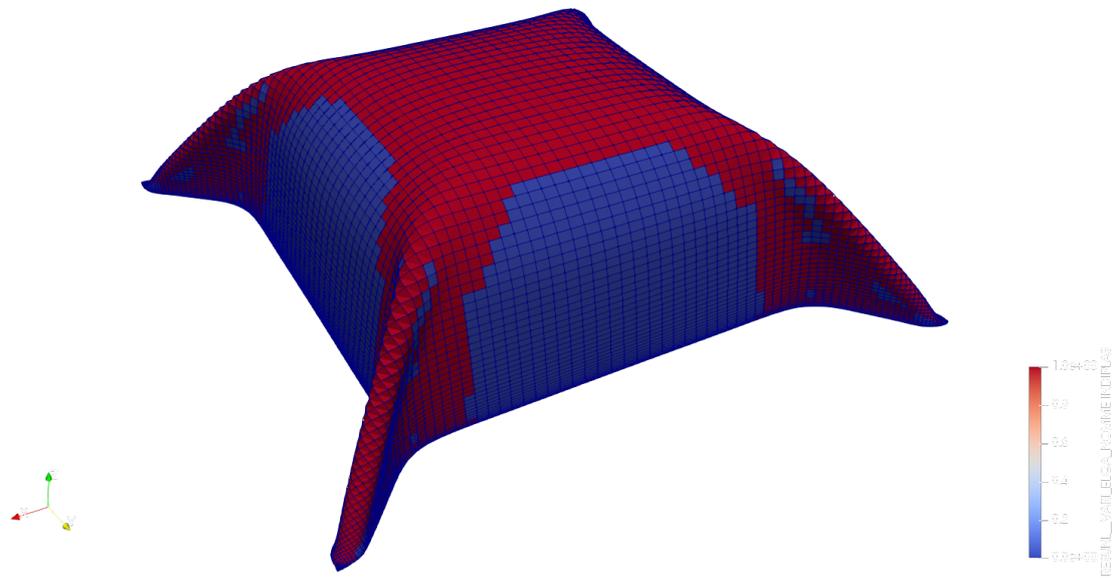
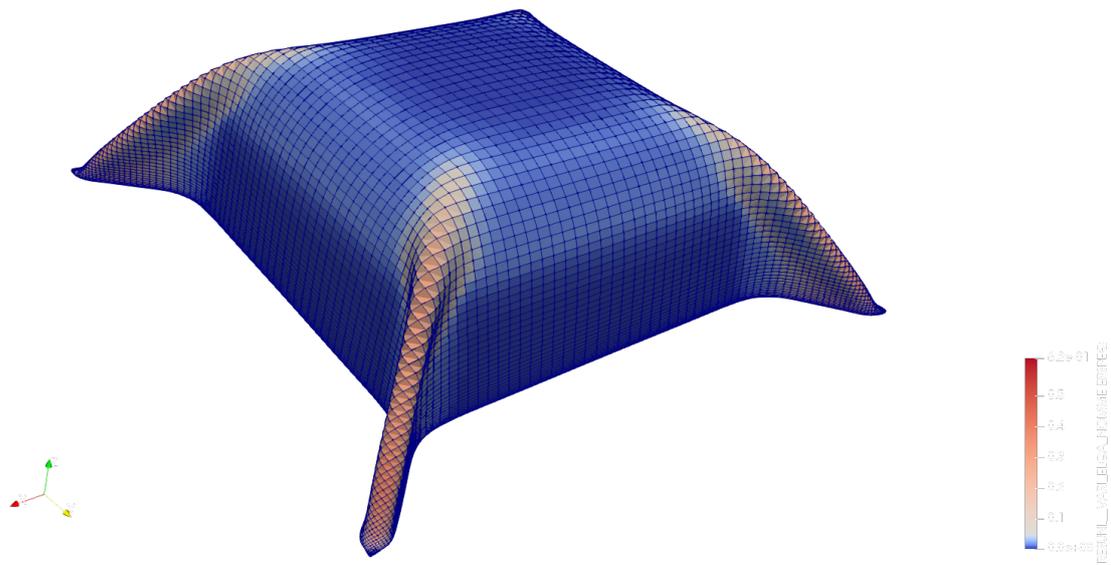


Figure 3.20: Evolution of the normal displacement of the plate center



(a) Index of plastification



(b) Plastic deformation

Figure 3.21: Plastic variables

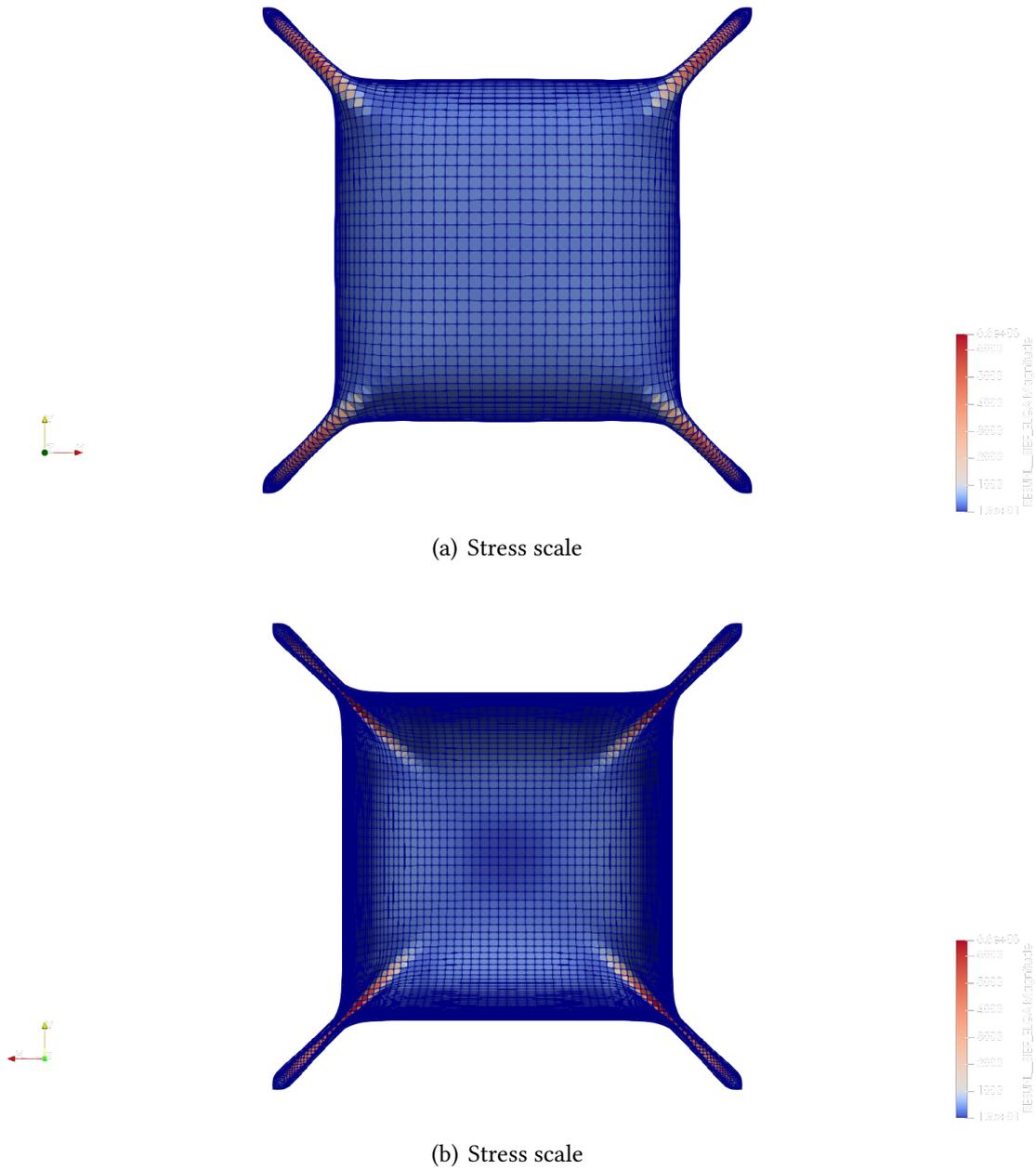


Figure 3.22: Stress Magnitude

3.4.5 Cook's membrane

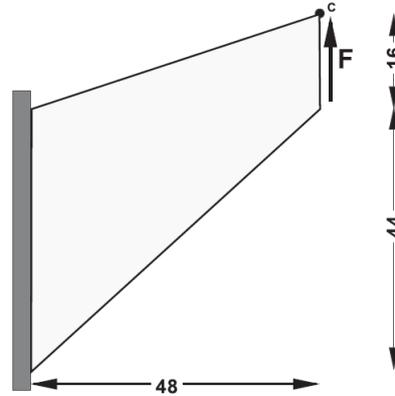
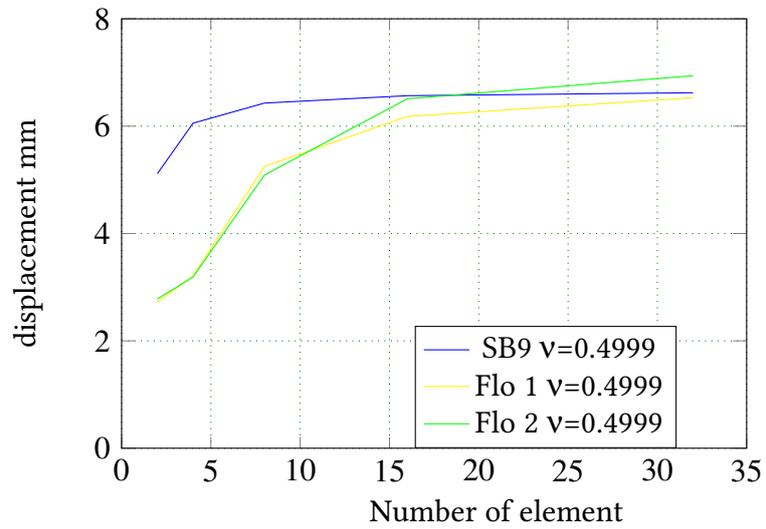


Figure 3.23: Cook's membrane geometry.

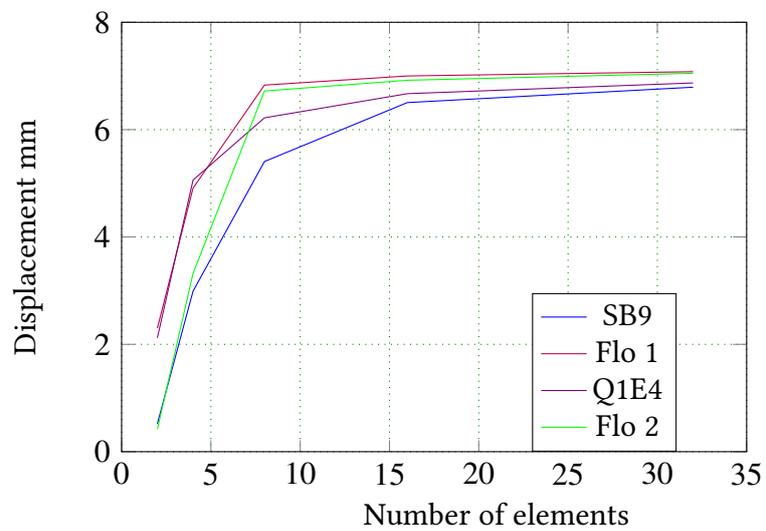
The Cook's membrane Figure 3.23 test simulates the deformation of a sheet metal of thickness $t=1\text{mm}$, clamped on one edge and subjected to a shear stress $F=6250\text{N}/\text{mm}^2$ in the opposite edge. Two tests are carried out. The first simulation is done with a quasi-incompressible elastic material, with a Young's modulus $E=2.4 \times 10^5\text{N}/\text{mm}^2$ and a Poisson's ratio $\nu=0.4999$ and the final displacement of the point C is evaluated as a function of the number of elements, so it is a convergence study. The applied load is 100kN for the elastic case and 5kN for the elastic-plastic material. The second simulation uses the same membrane as before, but this time with an elasto-plastic material of Young's modulus $E=2.069 \times 10^5\text{N}/\text{mm}^2$ and a Poisson's ratio $\nu=0.29$. The strain hardening law follows the formula $\sigma_y = 715 + 129.24\varepsilon^p - 265e^{-16.96\varepsilon^p}$, where ε^p corresponds to plastic deformation. Here also a convergence study was conducted. Figures 3.24.a and 3.24.b show a convergence analysis as the mesh is refined, where the vertical displacement of point C has been plotted vs. the number of divisions per side. The results have been compared with the reference Flores [MAT 18], and other results presented in the same article, where the results that we have obtained present a very good convergence in comparison.

3.4.6 Stamping of a sheet by a cylindrical punch (Doc aster)

This test represents a calculation of the drawing of a square sheet metal by a rigid cylindrical punch in the presence of large plastic deformations. This test is useful in the simulation of sheet metal forming. The sheet metal is modelled in SB9 elements. The punch and die are rigid elements. The contact between the different elements is modelled by a continuous formulation. The side length of the sheet is $L = 160\text{mm}$ and thickness $e = 2\text{mm}$. the die is $h = 35\text{mm}$ deep with a diameter of $D = 100\text{mm}$. The geometric details of the tools are visible in the Figure 3.25. The elastic parameters are the Young modulus $E = 200000\text{Mpa}$ and the Poisson ratio $\nu = 0.3$ and the hardening law is plotted in figure 3.26. To stamp the sheet, the punch is imposed with a vertical displacement



(a) Elastic quasi-incompressible



(b) Elastic-plastic

Figure 3.24: Evolution of the cantilever beam's free-edge displacement

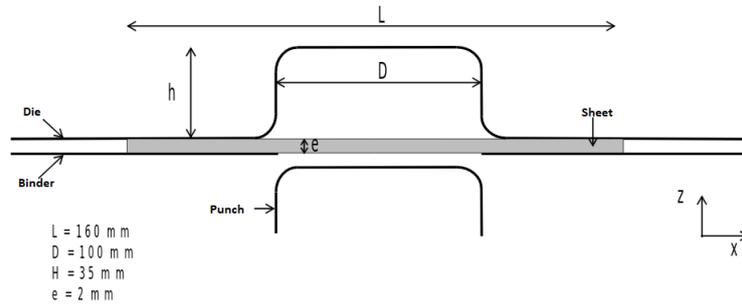


Figure 3.25: Stamping of a sheet by a cylindrical punch, tools geometry

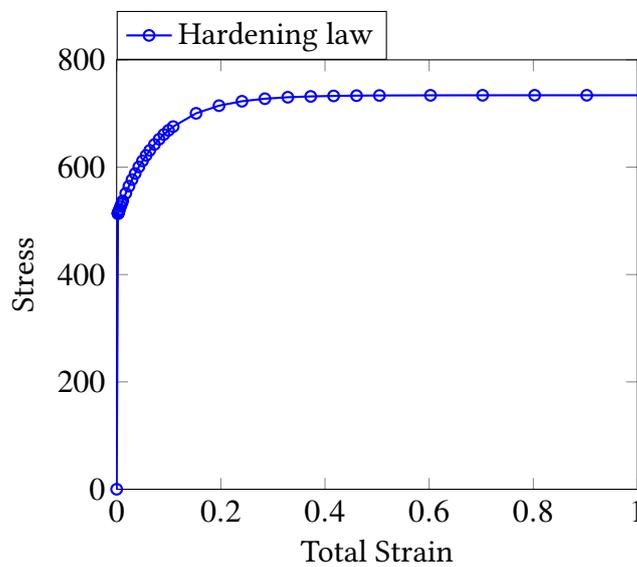


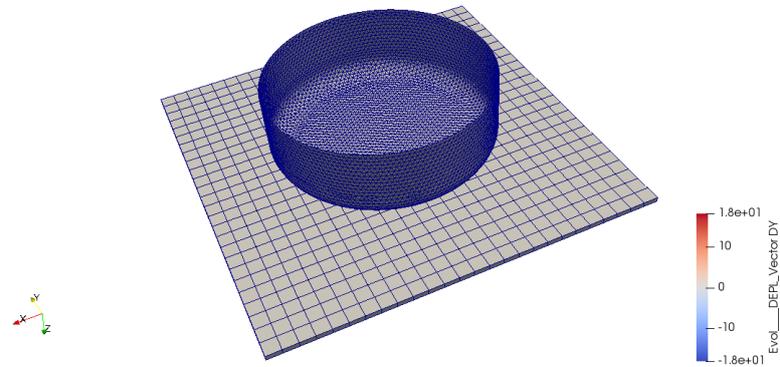
Figure 3.26: Hardening law

from 0 to 35mm. Figure 3.27 shows the deformation state of the sheet. This concludes that for these kind of problems the SB9 element is very robust, gives excellent results and is quite efficient.

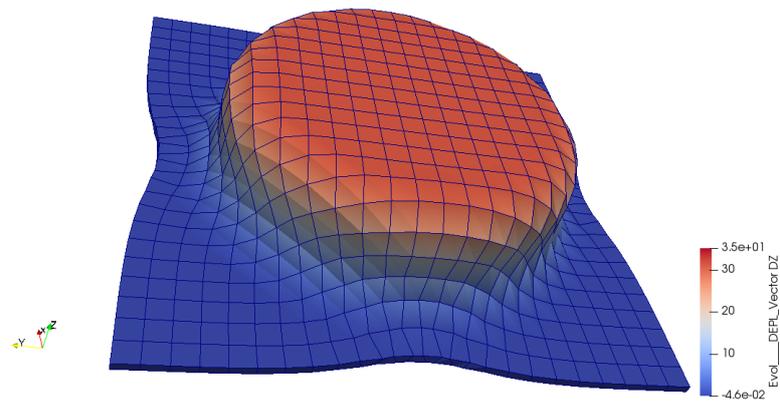
3.5 Modal analysis

3.5.1 Natural frequencies of a thick cylindrical ring (Doc aster)

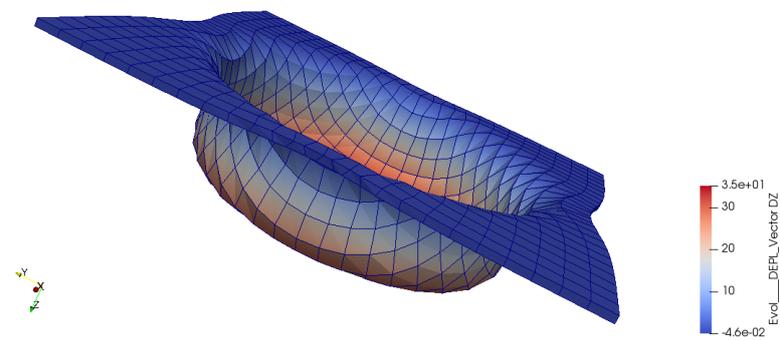
This test is inspired by a vibration study carried out on the VVP collector (steam evacuation systems between the primary and secondary circuits). This collector is medium-thick and has a maximum ratio of thickness to average radius of 0.13. This value, which may be typical of an industrial structure, is slightly higher than the limit of validity value usually recognized for plates and shells. In this study, the modeling of the shell collector is then evaluated by comparison with a volume model on a ring, showing the



(a) Punch and metal sheet



(b) Deformation after stamping



(c) Deformation after stamping

Figure 3.27: Stamping of a sheet by a cylindrical punch, deformation state

Modes	SB9 (Hz)	Ref	Errors SB9	Error DKQ	Error Coque_3D
Ovalization	210.16	210.55	0.18%	0.44%	0.3%
	210.16	210.55	0.18%	0.44%	0.3%
Trifoliated	587.86	587.92	0.01%	1.75%	0.24%
	587.86	587.92	0.01%	1.75%	0.24%
Out of plane	209.66	205.89	1.79%	13.99%	0.36%
	209.66	205.89	1.79%	13.99%	0.36%
	596.99	588.88	1.35%	9.75%	0.23%
	596.99	588.88	1.35%	9.75%	0.23%

Table 3.10: Natural frequencies of a thick cylindrical ring

Fourier modes of order 2 (ovalization) and 3 (trifoliate) as well as 2 out-of-plane modes. It is a cylindrical ring, with mean radius $R_m = 0.369m$, thickness $t = 0.048m$ and length $L = 0.05m$. The material is homogeneous, isotropic, linear elastic. The elastic coefficients are Young's modulus $E = 185000MPa$ and $\nu = 0.3$. The density is constant and is $\rho = 7800kgm^{-3}$. The structure is free in space. The eigenmodes searched correspond to the Fourier modes of order 2 and 3 of the ring. The frequencies of a ring can be estimated from an analytical Euler curved beam model [bib1] or numerically with a very fine mesh of 3D standard modelization. The search for clean modes is carried out in the 200-800Hz band. The frequencies associated with the different modes, (see figure 3.28), are listed in table 3.10.

The SB9 solid-shell element gives very good results for this modal analysis. This element shows more accurate results for the ovalization and the Trifoliated modes compared to the other elements (DKQ, Coque_3D) but a little less accurate results for the out of plane compared to the Coque_3D shell element. This is due to the fact that the Coque_3D shell element has quadratic interpolation functions, but not only that, the Coque_3D is based on Reissner hypothesis allowing it the model medium thick shell structures. The DKQ is based on Kirchhoff hypothesis and is limited to thin shell-like structures while the SB9 solid-shell element can be used to model thin shell structures as well as medium thick structures.

3.5.2 Free vibration of a compression vane (Doc aster)

This test is used to validate the calculation of the natural frequencies of a linear elastic compression vane. It is a cylindrical panel whose dimensions and material characteristics (homogeneous and isotropic) are given in the table 3.11. The structure is clamped on its BD end, see figure 3.29. The first six natural frequencies are measured in the interval [80., 570.] and compared with a reference solution corresponding to experimental measurements given in Batoz [BAT 92]. Comparisons are also made with another shell elements existing in code_aster called Coque_3D, which is a nine-node biquadratic classical shell finite element. The six different modes of vibration of the SB9 elements are given in figure 3.30 and the equivalent frequencies are listed in the table 3.12. The results

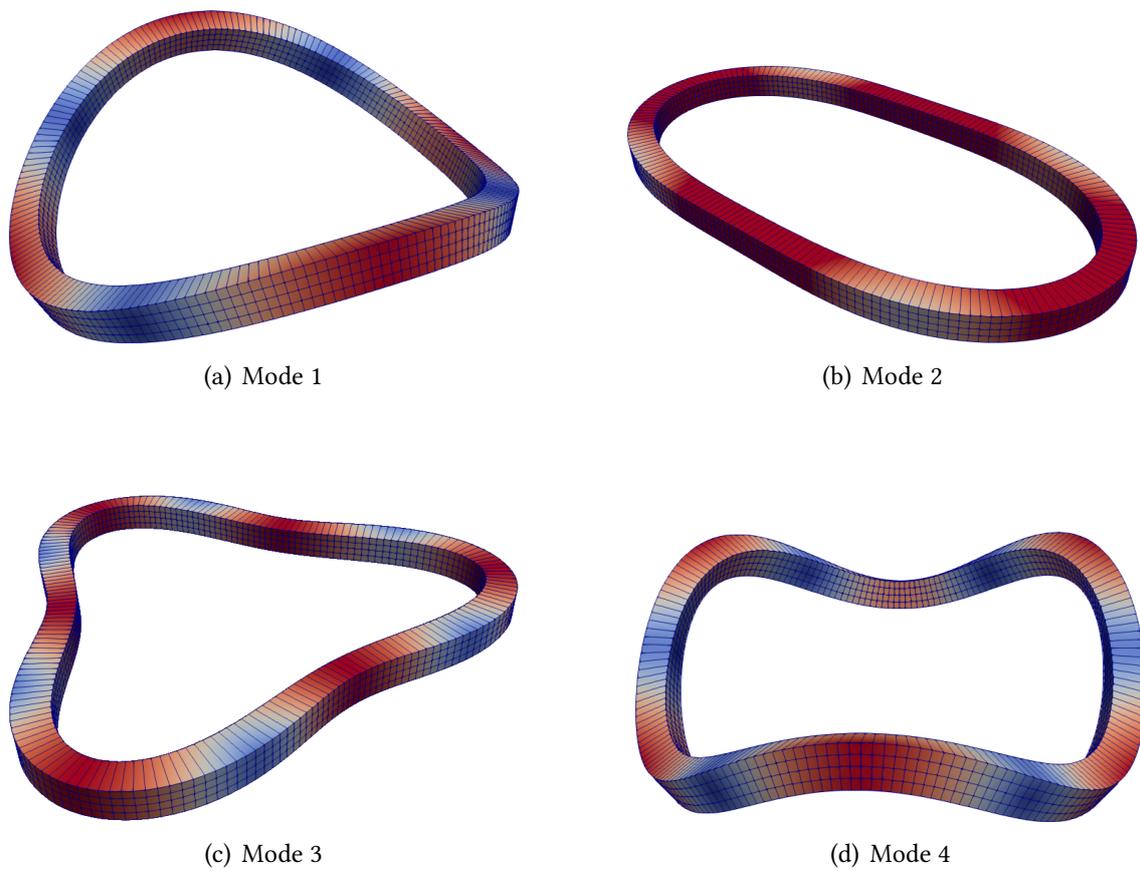


Figure 3.28: Modes propres d'un anneau cylindrique épais

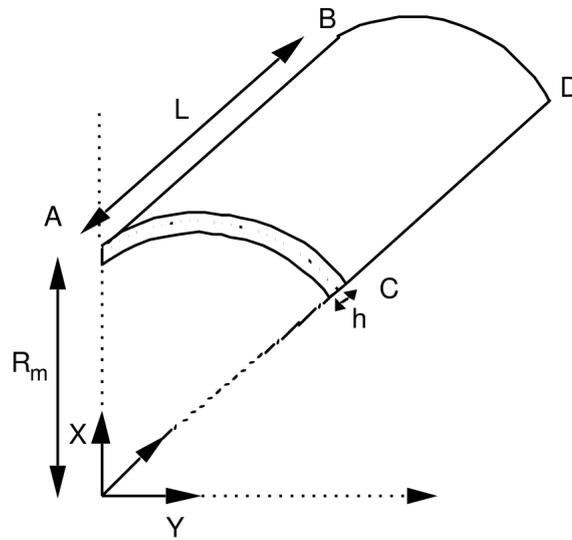


Figure 3.29: Geometry representation

Dimension	Valeur
E	$2.0685 \times 10^5 N/mm^2$
ν	0.3
ρ	$7857.2 kg.m^{-3}$
R_m	0.6096m
h	0.003048m
L	0.3048m
L_{arc}	0.3042m

Table 3.11: Geometrical and material characteristics of the cylinder

given by the SB9 elements are very satisfying.

mode	SB9	Coque 3D	Référence	Erreur relative SB9
1	85.9	85.85	85.6	0.35%
2	138.5	138.56	134.5	2.97%
3	247	246.92	259	4.63%
4	342	342.71	351	2.56%
5	386	386.66	395	2.28%
6	530	531.59	531	0.19%

Table 3.12: Frequencies in Hertz: comparison SB9 Vs Coque_3D

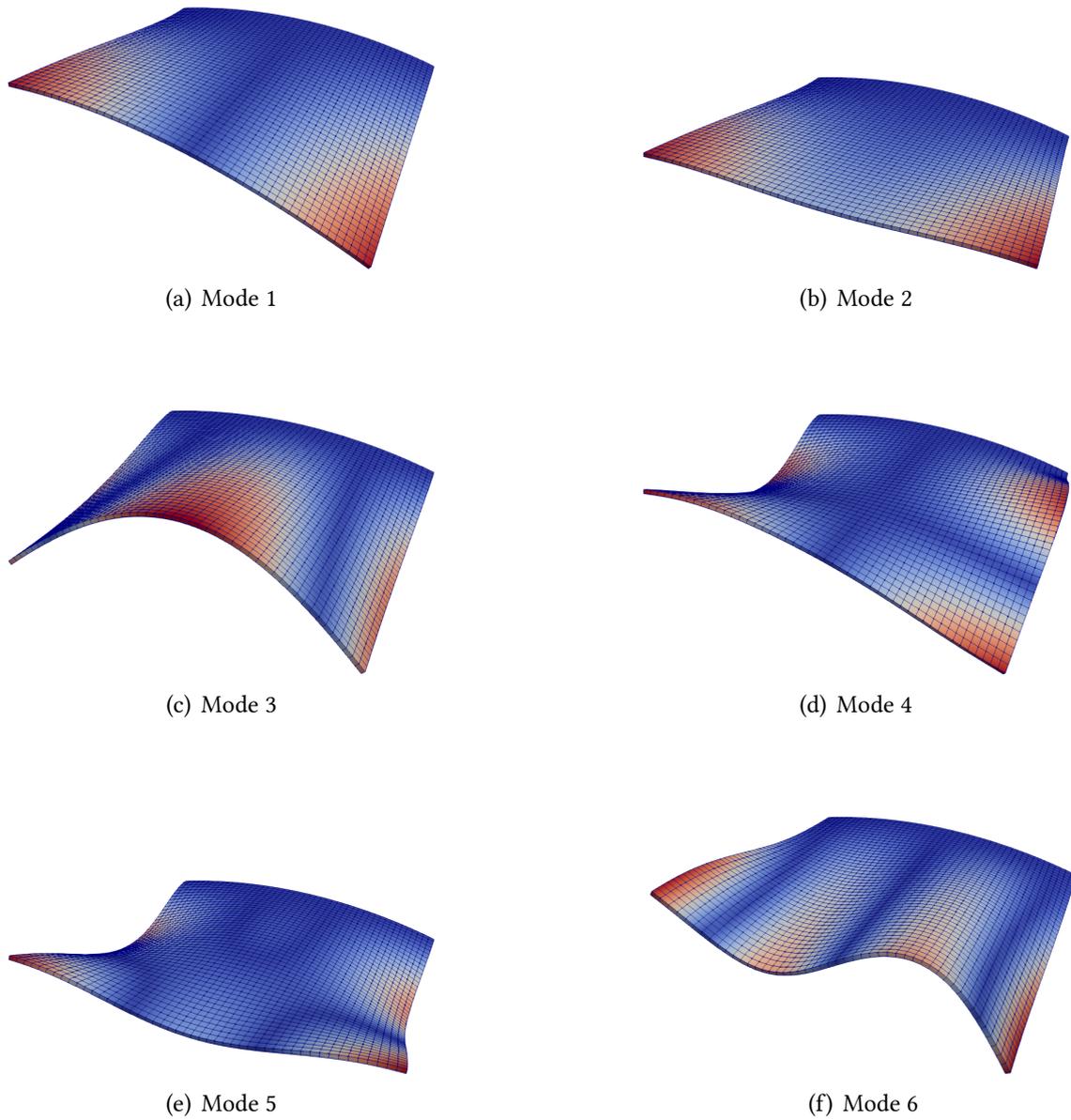


Figure 3.30: Vibrations libres d'une aube de compression

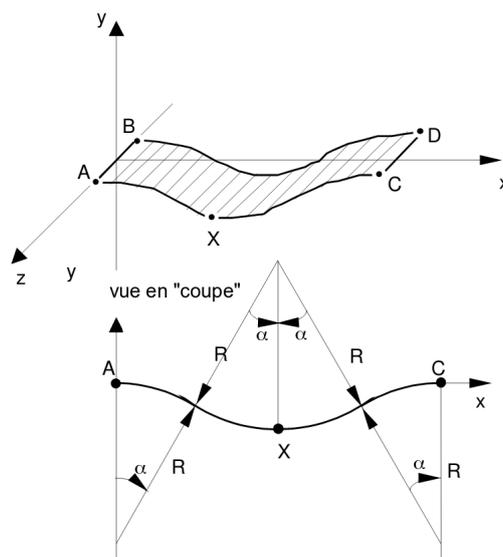


Figure 3.31: Geometry representation

3.5.3 Free vibration of a corrugated metal sheet (Doc aster)

This test represents a calculation in dynamic modal analysis of a corrugated sheet in free form, see figure 3.31. It is initially set up in doc_aster to validate the DKT finite element modelling on a non-planar plate modelled in quadrangles. The goal here is to see if the non-planarity has a consequence in the dynamic modal analysis of a SB9 model. The natural frequencies are then compared with a reference solution obtained with code_aster from a Coque_3D modeling. The geometrical and material characteristics of the corrugated shell are given in the table (3.13). The angle α is chosen so that the upper surface of the shell at point X is at ($y = 0$), i.e. aligned with A and C.

$$\cos(\alpha) = 1 - \frac{1}{4} \frac{h}{R} \quad (3.27)$$

The SB9 element gives very accurate results despite the geometry being initially non planar. The results are very close to those given by quadratic elements (Coque_3D and HEXA 27 finites elements) thanks to the enhancements embedded in this element. The DKT elements on the other hand is less accurate due it's linear interpolations.

	Coque_3D	SB9	Hexa27	DKT
Frequencies (Hz)	1686.02	1696.25	1688.28	1797.5
relative Error		0.61%	0.13%	6.61%

Table 3.16: Mode 3

Dimension	Valeur
E	$2 \times 10^5 N/mm^2$
ν	0.3
ρ	$7800 kg.m^{-3}$
R	1m
h	0.05m
AB et CD	0.1m
α	9.07°

Table 3.13: Geometrical and material characteristics of the sheet

	Coque_3D	SB9	Hexa27	DKT
Frequencies (Hz)	631.2	633.68	631.45	645.88
relative Error		0.39%	0.04%	2.33%

Table 3.14: Mode 1

	Coque_3D	SB9	Hexa27	DKT
Frequencies (Hz)	1236.14	1238.96	1236.39	1256.35
relative Error		0.23%	0.02%	1.63%

Table 3.15: Mode 2

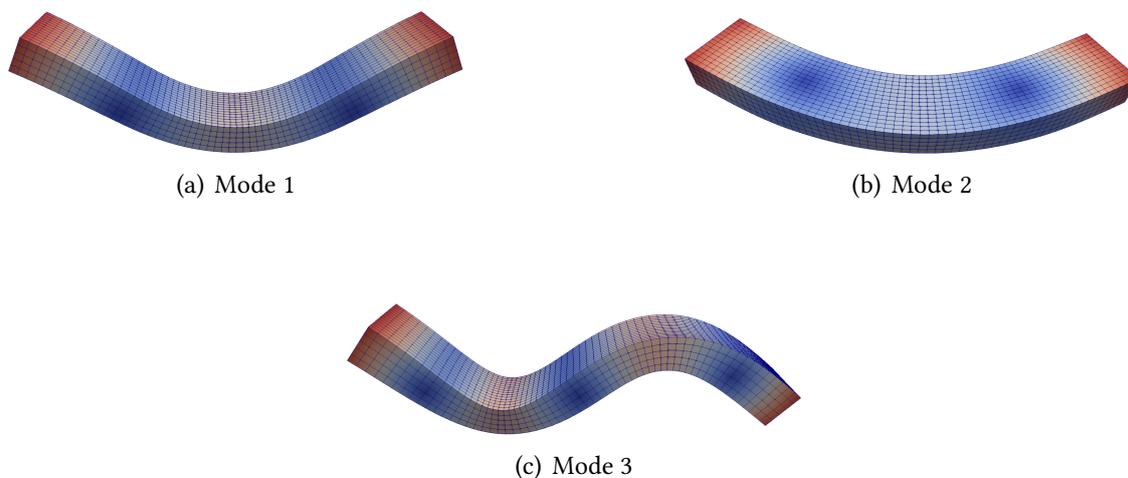


Figure 3.32: Free vibration of a corrugated metal sheet

3.5.4 Free thin circular ring (Doc aster)

This test represents a dynamic modal analysis calculation of a free-free circular ring, figure 3.33. The geometrical characteristics and material of the corrugated shell are given in the table 3.17. All points of the ring are free and no loading is applied. The calculation method used for the reference solution is:

- Vibration modes in the plane

For these modes of vibration, the bending equation for curved beams by V. Boussinesq (1883), without extension of the neutral fibre leads to

$$f_i = \frac{1}{2\pi} \sqrt{\frac{i^2(i^2 - 1)^2}{i^2 + 1} \frac{EI_z}{\rho AR^4}} \quad i = 0, 1, 2, \dots \quad (3.28)$$

The reference solution is established for thin archs such as

$$\alpha R \geq 100 \sqrt{\frac{I_z}{A}} \quad (3.29)$$

with α center angle in radians.

- Off-plane vibration modes

For transverse vibration modes with rectangular cross section, the solution was established from the results of two calculation codes, using different formulations, (see Guide de Validation des Progiciels de Calculs des Structures: SFM, AFNOR technique, ISBN: 2-12-486611-7). The frequencies associated to the vibration modes are given in the table 3.18, and compared to the references solutions. The results, see figure 3.34, given by the SB9 element are here also very accurate.

Dimension	Valeur
E	$2 \times 10^5 N/mm^2$
ν	0.3
R	2m
e	0.02m
L	2m

Table 3.17: Geometrical and material characteristics of the ring

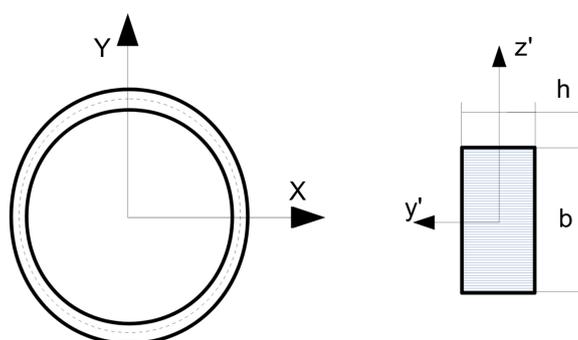
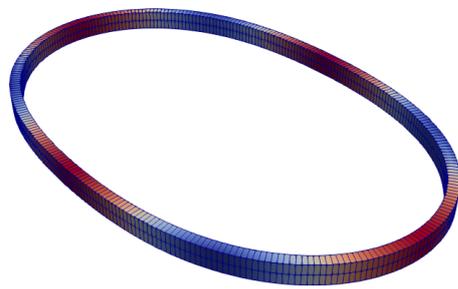


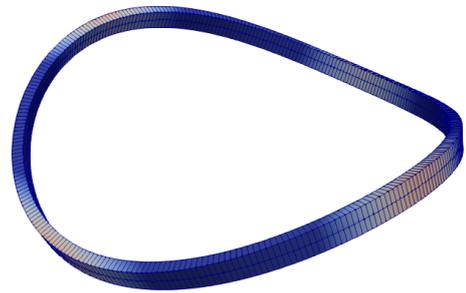
Figure 3.33: Geometry representation

Modes	SB9 (Hz)	Ref (Hz)	Relative error
1	318.05	318.36	0.10%
2	516.1	511	1.00%
3	898.31	900.46	0.24%
4	1566.28	1590	1.49%
5	1718.96	1726.55	0.44%
6	2772.73	2792.21	0.70%
7	3082.96	3184	3.17%

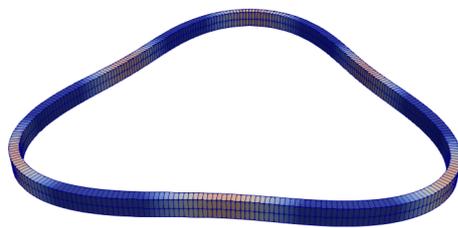
Table 3.18: Natural frequencies



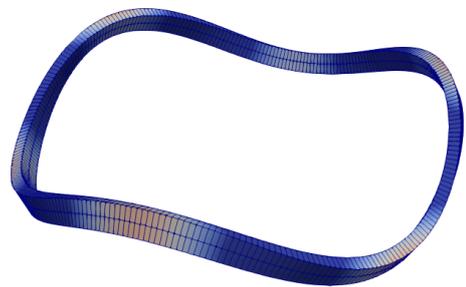
(a) Mode 1



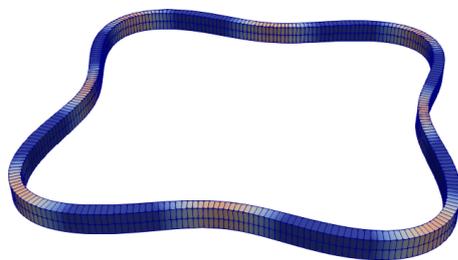
(b) Mode 2



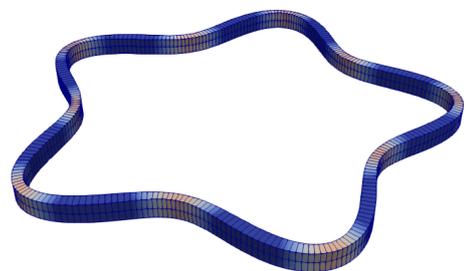
(c) Mode 3



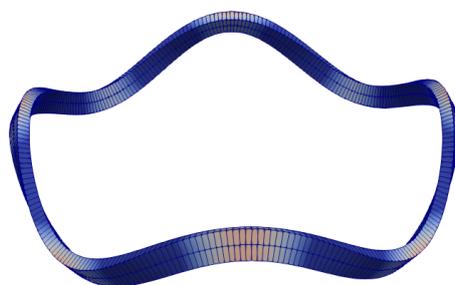
(d) Mode 4



(e) Mode 5



(f) Mode 6



(g) Mode 7

Figure 3.34: Free thin circular ring

3.5.5 Vibration of a plate in transitory state (Doc aster)

This test case uses SB9 for linear dynamic transient analysis. It deals with a plate subject to sinusoidal excitation on a vibrating table, see figure 3.35. The square plate is made of steel and its dimensions are width $L = 2m$ and thickness $t = 0.03m$. It is clamped on one edge and subjected to a sinusoidal volume force. The material used is elastic, with a Young's modulus $E = 2 \times 10^5 N/mm^2$, a Poisson's ratio $\nu = 0.3$ and a density $\rho = 8000 kg.m^{-3}$. The input signal is a simple sine with a frequency of 15Hz. The amplitude is 30g horizontally and 30g vertically down ($g = 10m/s^2$) and the duration of the signal is 0.5 seconds. We measure the displacement of the point P over time and compare it to the displacement given by a model with standard 3D quadratic elements (3 elements in the thickness direction vs 1 element in the thickness direction for the SB9). Figure 3.36 gives the displacement of the point P function of the time for the SB9 and the HEXA27. The SB9 element gives the same results as the HEXA27 element.

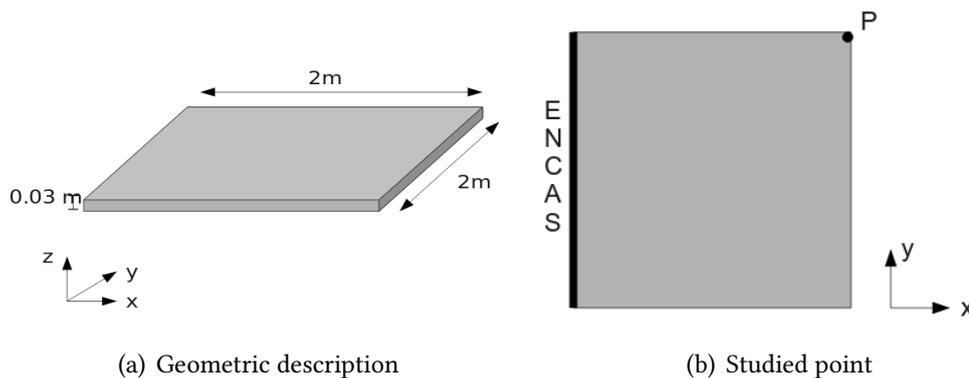


Figure 3.35: Vibration of a plate in transitory state

3.6 Linear buckling analysis

A complete non-linear calculation can be expensive in terms of the computer time that is needed. For stability problems it is desirable to have a simple method which gives an accurate estimate of the critical load at which loss of structural stability occurs. Such a method is known as linear buckling analysis. In this method the complete non-linear analysis in which the entire load–deflection path is followed up to, and possibly beyond the critical load level, is replaced by an eigenvalue analysis. A derivation of the method, starting from the complete set of the non-linear field equations and elucidating the assumptions that are made, is given below. The basic assumption is that prior to the point where loss of uniqueness occurs, the displacement gradients remain small. With this

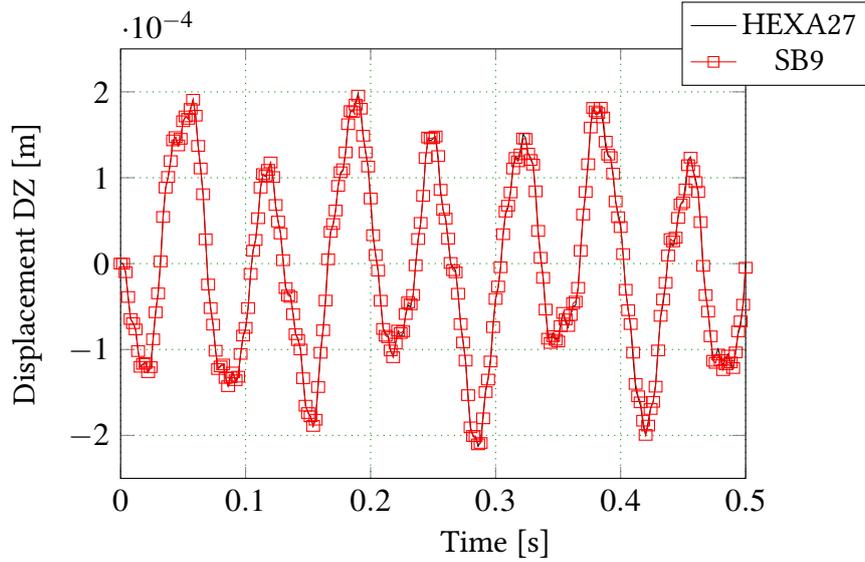


Figure 3.36: Evolution of displacement of the Point P

assumption the equilibrium equation can be written as follows :

$$\int_{\Omega_0} (\delta\Delta\mathbf{u})^T \mathbf{B}^T \mathbf{D} \mathbf{B} \Delta\mathbf{u} d\Omega + \int_{\Omega_0} (\delta\Delta\mathbf{u})^T \mathbf{B}_{NL}^T \Sigma \mathbf{B}_{NL} \Delta\mathbf{u} d\Omega = \int_{\delta\Omega_0} (\delta\Delta\mathbf{u})^T \mathbf{H}^T \mathbf{t} \Delta\mathbf{u} d\delta\Omega + \int_{\Omega_0} \rho (\delta\Delta\mathbf{u})^T \mathbf{H}^T \mathbf{g} \Delta\mathbf{u} d\Omega \quad (3.30)$$

Since the latter equation must hold for any virtual displacement increment $\delta\Delta\mathbf{u}$, the resulting set of equilibrium equations ensues :

$$\left[\int_{\Omega_0} \mathbf{B}^T \mathbf{D} \mathbf{B} \Delta\mathbf{u} d\Omega + \int_{\Omega_0} \mathbf{B}_{NL}^T \Sigma \mathbf{B}_{NL} d\Omega \right] \Delta\mathbf{u} = \int_{\delta\Omega_0} \mathbf{H}^T \mathbf{t} \Delta\mathbf{u} d\delta\Omega + \int_{\Omega_0} \rho \mathbf{H}^T \mathbf{g} d\Omega \quad (3.31)$$

At the critical load level at least two solutions exist which both satisfy incremental equilibrium [DEB 12]. If $\Delta\mathbf{u}_1$ denotes the incremental displacement field belonging to the first solution and if $\Delta\mathbf{u}_2$ is the incremental displacement field of the second solution, subtraction of the solutions, which must both satisfy Equation (3.31), yields:

$$\left[\int_{\Omega_0} \mathbf{B}^T \mathbf{D} \mathbf{B} \Delta\mathbf{u} d\Omega + \int_{\Omega_0} \mathbf{B}_{NL}^T \Sigma_c \mathbf{B}_{NL} d\Omega \right] (\Delta\mathbf{u}_1 - \Delta\mathbf{u}_2) = 0 \quad (3.32)$$

with Σ_c the stress matrix at the critical (buckling) load.

We now label Σ_e as the stress matrix that is obtained in a linear-elastic calculation for a unit load, and λ the load factor that sets the relation between this elastic solution for a unit load and the stress at the critical load Σ_c : $\Sigma_c = \lambda \Sigma_e$, so that λ is the multiplication factor for a unit load to obtain the critical load at which loss of uniqueness (buckling) occurs. Substitution of this identity in Equation (3.32) yields :

$$\left[\int_{\Omega_0} \mathbf{B}^T \mathbf{D} \mathbf{B} \Delta\mathbf{u} d\Omega + \lambda \int_{\Omega_0} \mathbf{B}_{NL}^T \Sigma_e \mathbf{B}_{NL} d\Omega \right] (\Delta\mathbf{u}_1 - \Delta\mathbf{u}_2) = 0 \quad (3.33)$$

Since by definition $\Delta \mathbf{u}_1 - \Delta \mathbf{u}_2 \neq 0$, his equation forms a linear eigenvalue problem for which a non-trivial solution exists if and only if the determinant of the characteristic equation vanishes:

$$\det(\mathbf{K}_0 + \lambda \mathbf{K}_{NL}^e) = 0 \quad (3.34)$$

where

$$\mathbf{K}_0 = \int_{\Omega_0} \mathbf{B}^T \mathbf{D} \mathbf{B} \Delta \mathbf{u} \quad \text{and} \quad \mathbf{K}_{NL}^e = \int_{\Omega_0} \mathbf{B}_{NL}^T \Sigma_e \mathbf{B}_{NL} \quad (3.35)$$

The solution of Equation (3.34) results in n eigenvalues, which belong to the load levels at which loss of uniqueness (bifurcation) can occur. The lowest eigenvalue corresponds to the lowest load level for which a bifurcation exists. Multiplication of the elastic solution with this eigenvalue therefore gives the critical load level at which bifurcation is first possible. It is noted that the load levels predicted by the higher eigenvalues can be so high that the assumptions made for the linear buckling analysis may be violated, and that these eigenvalues can therefore be unrealistic. In computations of structures that have a fine discretization, high eigenvalues may arise that are merely artifacts of the discretization, which is another possible source of unrealistic eigenvalues. In this section we investigate the SB9 and the SB7 through some linear buckling studies following the above mentioned assumptions.

3.6.1 Buckling of a free cylinder under external pressure (Doc aster)

This test represents a stability calculation of a thin cylindrical envelope free at its ends subjected to external pressure. The critical loads leading to Euler's elastic buckling are calculated. The matrix of geometrical stiffness used in solving the eigenvalue problem is that due to the initial constraints. The critical load and eigenmode obtained are compared with an analytical reference solution. The simulated cylinder has a mean radius $R = 2m$, a length $L = 2m$ and a thickness $t = 0.02m$. The material parameters are the Young modulus $E = 2 \times 10^5 N/mm^2$ and the Poisson's ratio $\nu = 0.3$. The symmetry of the problem makes it possible to model one eighth of the cylinder, with specific symmetry conditions. A uniformly distributed pressure of $P_c = 1Pa$ is applied to the cylindrical surface. The critical pressure found in the references [TIM 47, DON 75] are given as

$$P_{cr} = \frac{E}{12(1-\nu^2)} \left(\frac{e}{R}\right)^3 n^2 \quad (3.36)$$

n being the mode's number (with $n = 2, 4, 6$).

3. Numerical validations

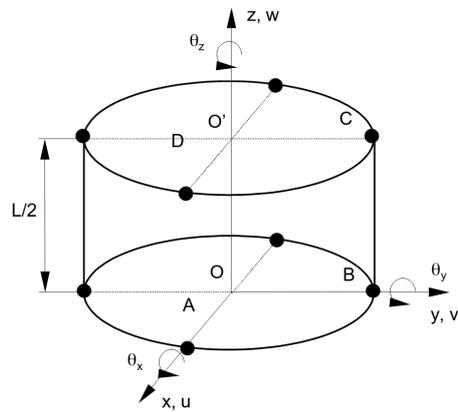


Figure 3.37: Geometry of the cylinder

mode	SB9	Reference	Relative Error
2	71974	73260	1.76%
4	291378	293040	0.57%
6	668816	659340	1.44%

Table 3.19: Comparison of critical pressure load : SB9 vs reference

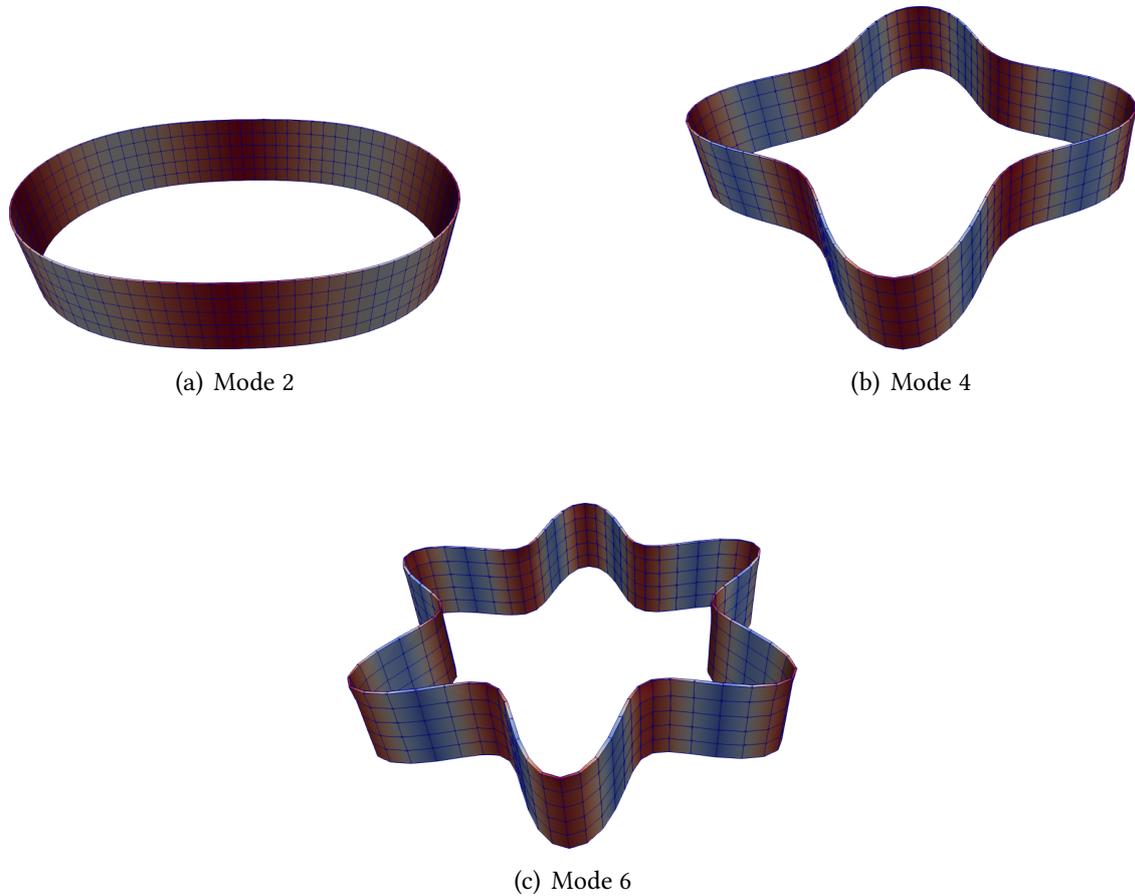


Figure 3.38: Buckling modes of a free cylinder under external pressure

3.7 Industrial application : study of a canopy joint

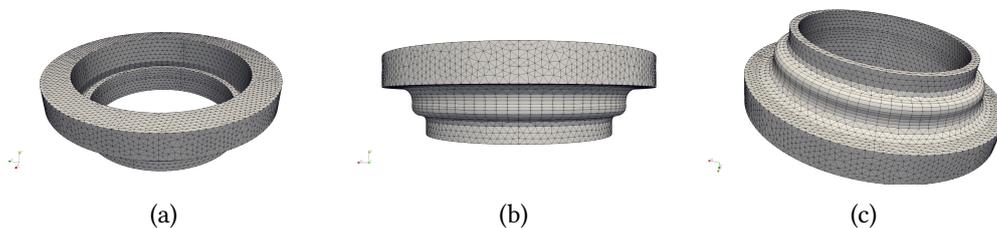


Figure 3.39: Canopy Joint 3D mesh

The covers of PWR reactor vessels support the nuclear reaction control equipment: the bundles, thermocouples, etc., which penetrate into the interior of the vessel via sealed feedthroughs in the covers. At this level, the mounting flanges of the parts are connected by welded parts used for their sealing. During repairs, it has been necessary to qualify the

3. Numerical validations

performance of this so-called canopy joint under the pressure exerted by the installation tool (approximately 30 MPa). We have carried out this study with an SB9 model to see the behavior of this new element for an industrial problem. We performed 2 elastic-plastic calculations (Von Mises isotropic hardening) in large deformations. We followed the hardening law in the diagram of figure 3.40. The first study is made with a combination of Axisymmetric quadratic elements (P2) with reduced integration (quadrangular elements) and fully integrated (triangular elements). The second study is a three-dimensional one and mixes an SB9 model (hexahedral elements) with classical 3D elements (Tetrahedrons and pyramids), see figure 3.41. The Young's modulus of the joint is $E=147001.07$ MPa with a Poisson's ratio: 0.30. The boundary conditions are : clamp of the edges 1,2,3,4,5 and a pressure load of 30 MPa is applied to the outer edge of the joint corresponding to the edge P, see figure 3.41.a. Figures 3.42, 3.43, 3.44 and 3.45 shows the different results : displacement, stress SXX, stress SXY, and the plastic deformation respectively. We can see that the results given by the SB9 modelization are very similar with the results given by the P2 axisymmetric modelization for all variables. This shows the applicability of the SB9 for many engineering problems. From this study we can retain three very interesting conclusions concerning the finite element SB9. The first is the quality of the results for a finite element of type P1. With a single element in the thickness direction, the results are quite close to those given by the under-integrated P2 elements. The second thing is the ease with which this element fits into a 3D model with a natural connection due to the fact that there are only translation degrees of freedom. The third important conclusion is the ease and manageability of thickness variation that conventional plate elements would be unable to achieve.

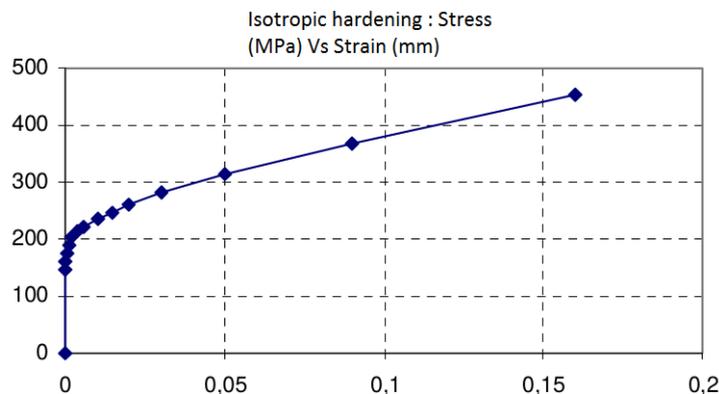
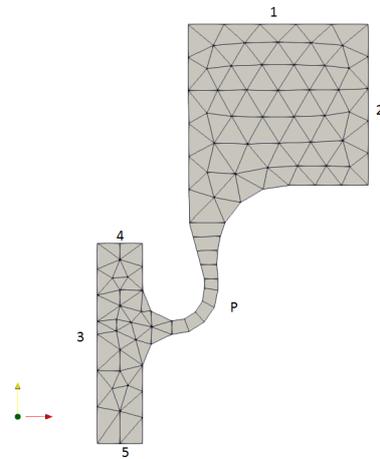
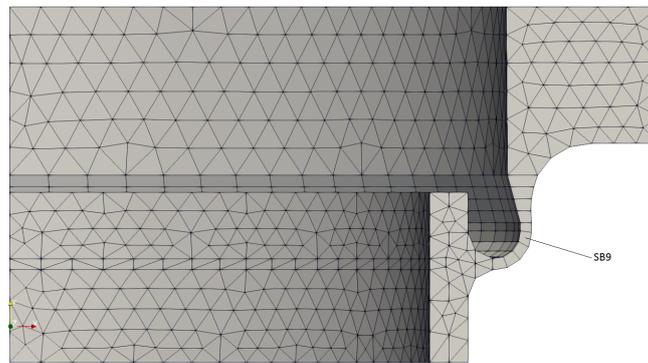


Figure 3.40: Isotropic hardening diagram

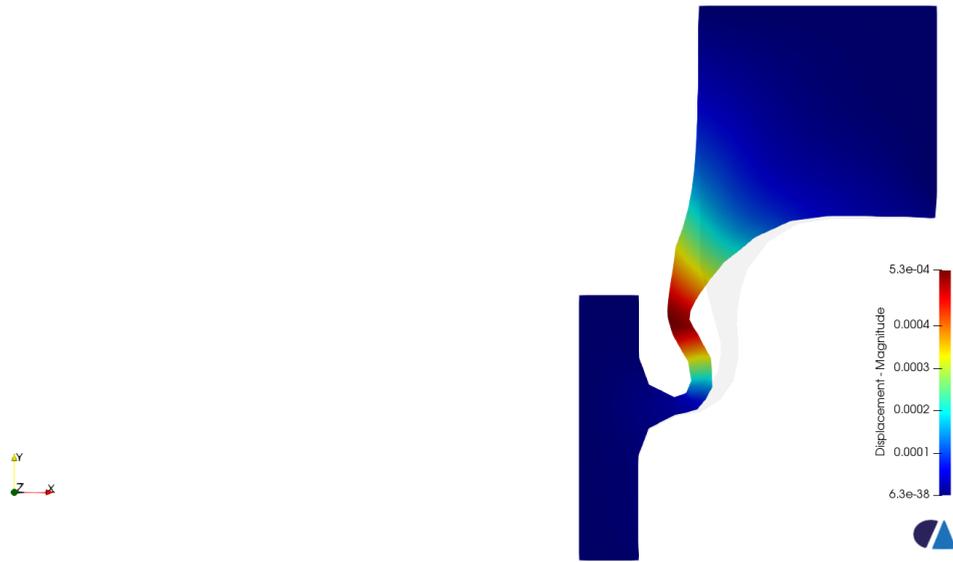


(a) AXI model

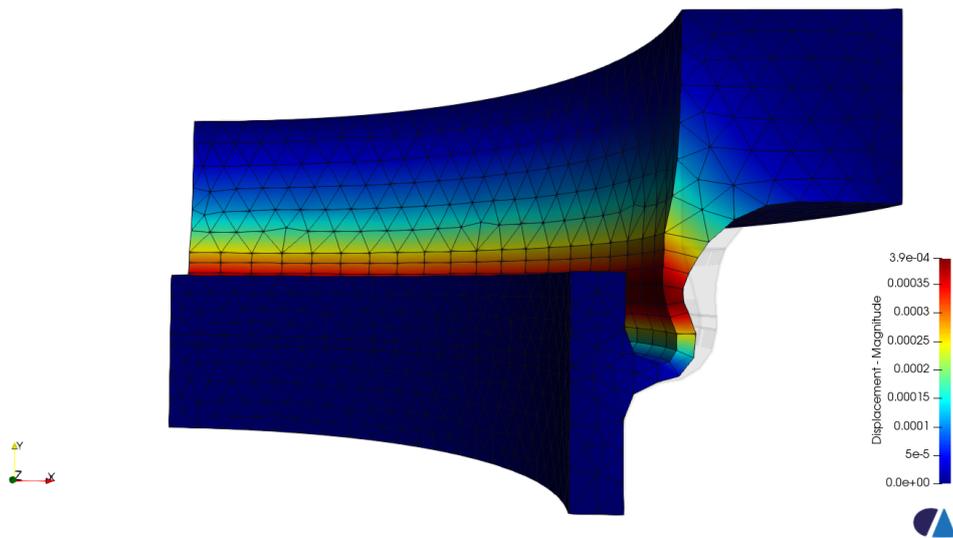


(b) 3D and SB9 model

Figure 3.41: The different modelizations

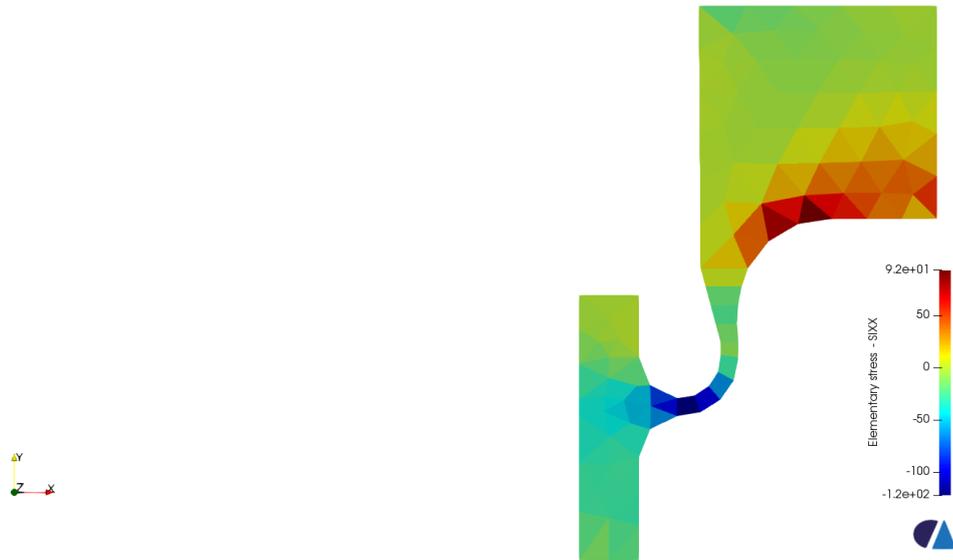


(a) AXI P2+ AXI_SI P2

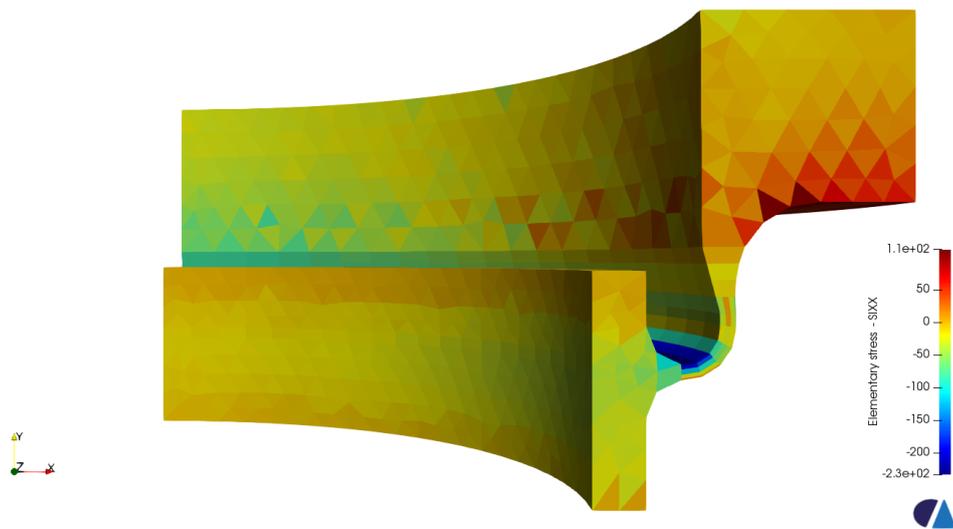


(b) SB9+3D P1

Figure 3.42: Displacements

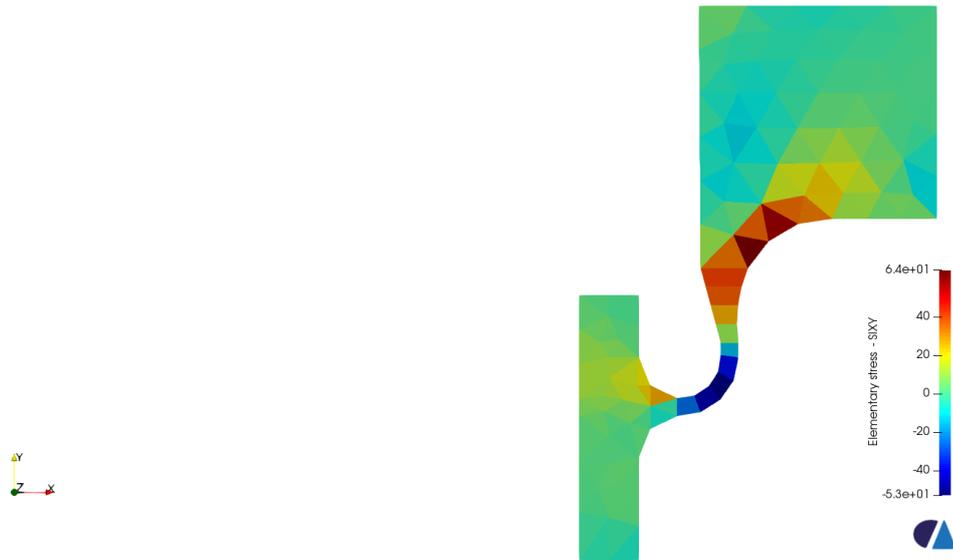


(a) AXI P2+ AXI_SI P2

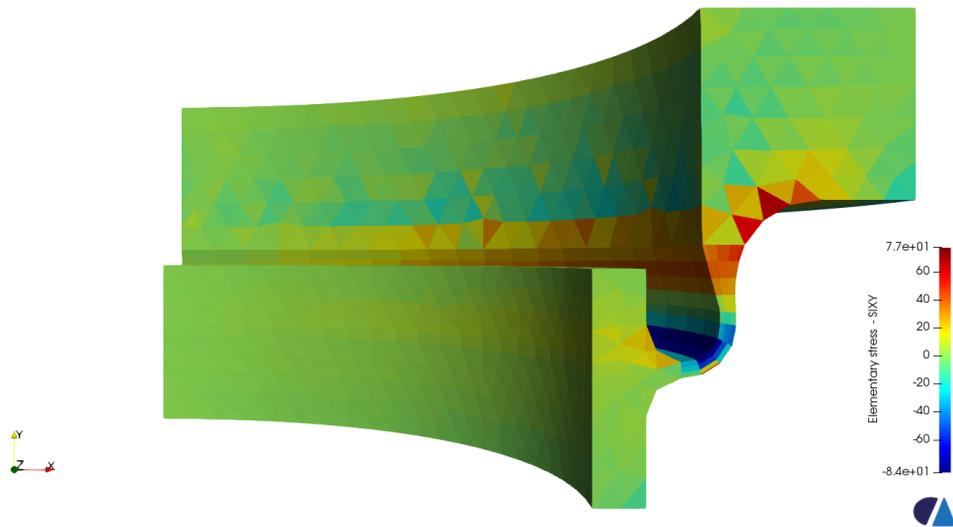


(b) SB9+3D

Figure 3.43: Stress SXX

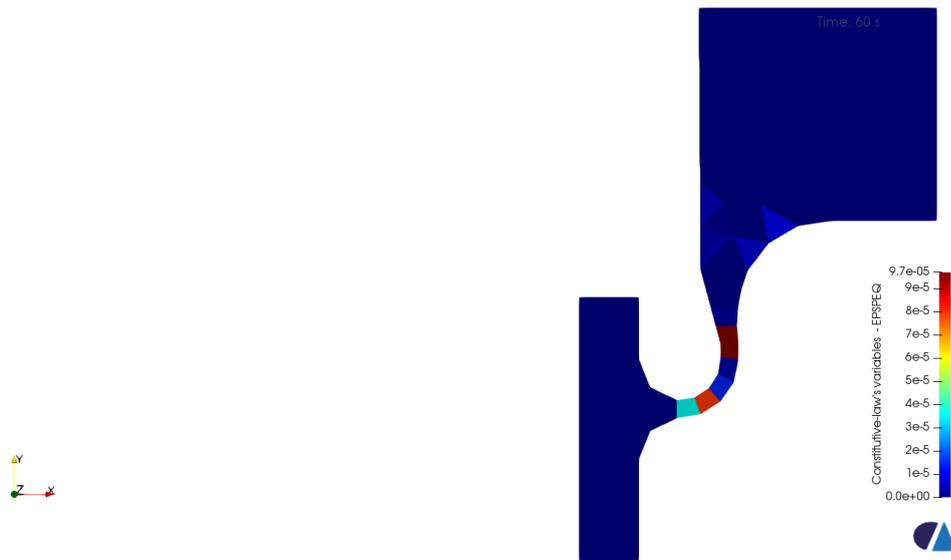


(a) AXI P2+ AXI_SI P2

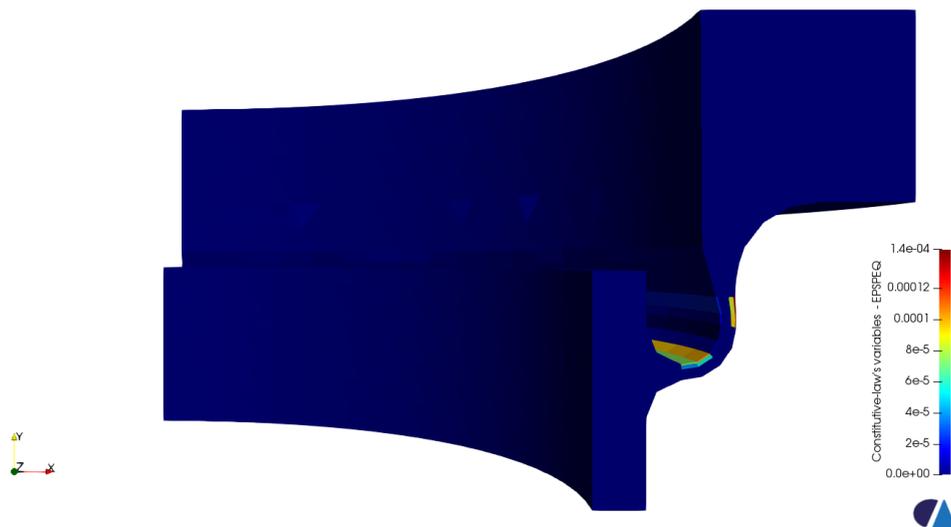


(b) SB9+3D P1

Figure 3.44: Stress S_{XY}



(a) AXI P2+ AXI_SI P2



(b) SB9+3D P1

Figure 3.45: Plastic deformations

3.8 Conclusion

The ninth or seventh node, with one translational degree of freedom through the elements thicknesses, allow the unrestricted use of 3D constitutive law without encountering Poisson thickness locking. Furthermore the adjunction of these middle nodes allow a redistribution of a pinching pressure force between a surface contribution and a volume contribution. This improves considerably the normal stress and gives better results than

elements with the EAS parameter alone, without pinching stress enrichment. In addition to eliminating pinching and volumetric locking thanks to the middle node, transverse and trapezoidal shear locking are reduced by the assumed strain method applied to both transverse shear and pinch strain. The two formulated solid-shell finite elements, SB9 and SB7, have been implemented in code_aster. In this chapter these two elements have been test with different types of mechanical problems available in the literature or in the documentation of code_aster. The results given by these two finite elements are very satisfying. The elements have been used in linear elastic as well as in large deformation bending problems including quasi-incompressible material. The results are very accurate for both elements. The SB9 solid-shell element have been further tested with material nonlinearity, for modal analysis as well as for buckling of shell like structures. In these case too, the SB9 finite element gives very accurate results in comparison with other classical shell finite elements.

Conclusion and perspectives

The aim of this work was to develop a finite element formulation to model the behaviour of thin structures without restrictive assumptions on the mechanical fields, i.e. stresses and strains. The objective is to propose to EDF a robust and performant formulation adapted to all the problems in terms of mechanical simulations that the company can face to dimension thin and medium thick structures present in the nuclear power plant. An advanced state of the art on finite element literature has shown the difficulty faced by engineers in modeling the behavior of thin structures. Conventional plate and shell finite elements are limited by several aspects of their formulation. First of all there is the plane stress hypothesis which states that the stress in the thickness is null. This hypothesis, which remains in contraction with the null pinch deformation, requires the laws of behaviour to be modified, often laboriously, in order to respect the conservation of mass and avoid pinch locking. Moreover, due to their bi-dimensional topology, these classical shell finite elements are difficult to use in the presence of finite elements with a tri-dimensional topology without developing techniques to connect the different elements. On the other hand, first order solid elements are very robust and almost usable for many mechanical modelling. However they perform very poorly when used for shell like structures. Indeed as shown in chapter 1 these elements suffer from several locking problems that disqualify them for thin structure modeling. furthermore in chapter one it has been shown that by using certain specific interpolations such as the assumed natural method or by enriching the deformations it is possible to “soften” first-order solid finite elements making them suitable for the modelling of shell like structures. This has given rise to a new type of finite element called solid-shell finite elements. These elements have the characteristic of being of three-dimensional topology and allow to model the bending of thin structures without restrictive assumptions on the mechanical fields. The goal of this thesis was to formulate a new all-terrain solid-shell finite element capable of modeling the behavior of shell like structures in all mechanical situations. The starting point of this work was the SB9 γ 25 formulated by Bassa [BAS 12]. Bassa’s idea was to add an additional node in the middle of a solid-shell element initially functioning under plane stress hypothesis, to enrich its normal strain and thus allow the use of a 3D constitutive law. Indeed, with a linear normal strain, pinch locking is eliminated and therefore there is no need to impose a plane stress hypothesis. In this thesis work we have borrowed Bassa’s idea but in a completely new formulation to propose the SB9 solid-shell finite element. First of all this new finite element is written in total Lagrangian formulation. Which eliminates the need to use a local corotational frame to guarantee an objective

modeling. Plus, the SB9 uses an assumed transverse shear strain (ANS) with four control points per component instead of the 2 control points per component used in the Bassa version. This eliminates the need for the Reissner function in transverse shear interpolation which is limiting for the SB9 γ 25 element. Further important locking effect, namely the curvature thickness locking, have been overcome utilizing also the ANS concept. To circumvent the volumetric locking and Poisson thickness locking, the EAS concept based on the Hu-Washizu variational principle has been implemented enhancing the Green–Lagrange strain tensor evaluated at the integration points. Herein the EAS concept is satisfied by the additional central node. It is then important to notice that the additional node effect goes beyond the usual EAS parameter since, since this additional node allows a distribution of the applied pressure in a way that enhances the pinching stress. The mapping of the compatible Green–Lagrange strain terms from the contravariant to the Cartesian basis is based on the Taylor expansion of the inverse Jacobian matrix. Hence at the one hand the element geometry is taken into account very accurately and on the other hand a polynomial form of the Cartesian Green–Lagrange strain is obtained which improves the efficiency of the element. Further crucial points are the Taylor expansion of the compatible Green Lagrange strain tensor and the second Piola–Kirchhoff stress tensor to derive an efficient adaptive hourglass stabilization concept. The corresponding hourglass residual vector and the hourglass stabilization matrix are integrated analytically. This new way of stabilization is automatic and user-independent. To ensure a volumetric locking-free hourglass stabilization, the B-Bar method is adopted in the so-called hourglass kernel. The SB7 finite element is a revision of the SB7 γ 19 [XIO 18] with a slight modification of the stabilization coefficient. We also used the assumed field method in the pinch to eliminate trapezoidal locking. These elements have been implemented in code_aster and tested on several test cases available in the literature or in the code_aster documentation. The numerical results show that these solid-shell finite elements are well adapted to most engineering problems in thin structure modeling. In particular, for bending problems, under quasi-incompressible conditions, elasto-plasticity, dynamics or for modeling buckling phenomenon of thin structures.

In perspective with the aim of having an all-terrain solid-shell finite element, it would be conceivable to extend the formulation on issues like anisotropy and thermal dilatation. The elements could also be tested on dynamic problems combining contact, material and geometric nonlinearities. To go further, it would be interesting also to use these solid-shell finite elements in formulations with the presence of cracks.

Annexes

Appendix A

Complement for SB9 formulation

A.1 Nine nodes solid-shell finite element SB9

$$\hat{\mathbf{E}}_c^0 = \begin{bmatrix} \mathbf{J}_1^{0T} \mathbf{D}_1^0 + 1/2 \mathbf{D}_1^{0T} \mathbf{D}_1^0 \\ \mathbf{J}_2^{0T} \mathbf{D}_2^0 + 1/2 \mathbf{D}_2^{0T} \mathbf{D}_2^0 \\ 1/4 \sum_{K=A}^D (\mathbf{J}_3^{KT} \mathbf{D}_3^K + 1/2 \mathbf{D}_3^{KT} \mathbf{D}_3^K) \\ \mathbf{J}_1^{0T} \mathbf{D}_2^0 + \mathbf{J}_2^{0T} \mathbf{D}_1^0 + \mathbf{D}_1^{0T} \mathbf{D}_2^0 \\ 1/4 \sum_{K=E}^H (\mathbf{J}_2^{KT} \mathbf{D}_3^K + \mathbf{J}_3^{KT} \mathbf{D}_2^K + \mathbf{D}_2^{KT} \mathbf{D}_3^K) \\ 1/4 \sum_{K=J}^M (\mathbf{J}_1^{KT} \mathbf{D}_3^K + \mathbf{J}_3^{KT} \mathbf{D}_1^K + \mathbf{D}_1^{KT} \mathbf{D}_3^K) \end{bmatrix} \quad (\text{A.1})$$

$$\hat{\mathbf{E}}_c^\zeta = \begin{bmatrix} \mathbf{J}_1^{0T} \mathbf{D}_1^\zeta + \mathbf{J}_1^{\zeta T} \mathbf{D}_1^0 + \mathbf{D}_1^{0T} \mathbf{D}_1^\zeta \\ \mathbf{J}_2^{0T} \mathbf{D}_2^\zeta + \mathbf{J}_2^{\zeta T} \mathbf{D}_2^0 + \mathbf{D}_2^{0T} \mathbf{D}_2^\zeta \\ 0 \\ \mathbf{J}_1^{0T} \mathbf{D}_2^\zeta + \mathbf{J}_1^{\zeta T} \mathbf{D}_2^0 + \mathbf{J}_2^{0T} \mathbf{D}_1^\zeta + \mathbf{J}_2^{\zeta T} \mathbf{D}_1^0 + \mathbf{D}_1^{0T} \mathbf{D}_2^\zeta + \mathbf{D}_1^{\zeta T} \mathbf{D}_2^0 \\ 1/4 \sum_{K=E}^H \zeta^K (\mathbf{J}_2^{KT} \mathbf{D}_3^K + \mathbf{J}_3^{KT} \mathbf{D}_2^K + \mathbf{D}_2^{KT} \mathbf{D}_3^K) \\ 1/4 \sum_{K=J}^M \zeta^K (\mathbf{J}_1^{KT} \mathbf{D}_3^K + \mathbf{J}_3^{KT} \mathbf{D}_1^K + \mathbf{D}_1^{KT} \mathbf{D}_3^K) \end{bmatrix} \quad (\text{A.2})$$

$$\hat{\mathbf{E}}_c^{\zeta\zeta} = \begin{bmatrix} \mathbf{J}_1^{\zeta T} \mathbf{D}_1^\zeta + 1/2 \mathbf{D}_1^{\zeta T} \mathbf{D}_1^\zeta \\ \mathbf{J}_2^{\zeta T} \mathbf{D}_2^\zeta + 1/2 \mathbf{D}_2^{\zeta T} \mathbf{D}_2^\zeta \\ 0 \\ \mathbf{J}_1^{\zeta T} \mathbf{D}_2^\zeta + \mathbf{J}_2^{\zeta T} \mathbf{D}_1^\zeta + \mathbf{D}_1^{\zeta T} \mathbf{D}_2^\zeta \\ 0 \\ 0 \end{bmatrix} \quad (\text{A.3})$$

$$\hat{\mathbf{E}}_c^\xi = \begin{bmatrix} 0 \\ \mathbf{J}_2^{0T} \mathbf{D}_2^\xi + \mathbf{J}_2^{\xi T} \mathbf{D}_2^0 + \mathbf{D}_2^{0T} \mathbf{D}_2^\xi \\ 1/4 \sum_{K=A}^D \xi^K (\mathbf{J}_3^{KT} \mathbf{D}_3^K + 1/2 \mathbf{D}_3^{KT} \mathbf{D}_3^K) \\ \mathbf{J}_1^{0T} \mathbf{D}_2^\xi + \mathbf{J}_2^{\xi T} \mathbf{D}_1^0 + \mathbf{D}_1^{0T} \mathbf{D}_2^\xi \\ 1/4 \sum_{K=E}^H \xi^K (\mathbf{J}_2^{KT} \mathbf{D}_3^K + \mathbf{J}_3^{KT} \mathbf{D}_2^K + \mathbf{D}_2^{KT} \mathbf{D}_3^K) \end{bmatrix} \quad (\text{A.4})$$

$$\hat{\mathbf{E}}_c^\eta = \begin{bmatrix} \mathbf{J}_1^{0T} \mathbf{D}_1^\eta + \mathbf{J}_1^{\eta T} \mathbf{D}_1^0 + \mathbf{D}_1^{0T} \mathbf{D}_1^\eta \\ 0 \\ 1/4 \sum_{K=A}^D \eta^K (\mathbf{J}_3^{KT} \mathbf{D}_3^K + 1/2 \mathbf{D}_3^{KT} \mathbf{D}_3^K) \\ \mathbf{J}_2^{0T} \mathbf{D}_1^\eta + \mathbf{J}_1^{\eta T} \mathbf{D}_2^0 + \mathbf{D}_2^{0T} \mathbf{D}_1^\eta \\ 0 \\ 1/4 \sum_{K=J}^M \eta^K (\mathbf{J}_1^{KT} \mathbf{D}_3^K + \mathbf{J}_3^{KT} \mathbf{D}_1^K + \mathbf{D}_1^{KT} \mathbf{D}_3^K) \end{bmatrix} \quad (\text{A.5})$$

$$\hat{\mathbf{E}}_c^{\eta\zeta} = \begin{bmatrix} \mathbf{J}_1^{0T} \mathbf{D}_1^{\eta\zeta} + \mathbf{J}_1^{\eta T} \mathbf{D}_1^\zeta + \mathbf{J}_1^{\zeta T} \mathbf{D}_1^\eta + \mathbf{J}_1^{\eta\zeta T} \mathbf{D}_1^0 + \mathbf{D}_1^{0T} \mathbf{D}_1^{\eta\zeta} + \mathbf{D}_1^{\eta T} \mathbf{D}_1^\zeta \\ 0 \\ 0 \\ \mathbf{J}_2^{0T} \mathbf{D}_1^{\eta\zeta} + \mathbf{J}_1^{\eta T} \mathbf{D}_2^\zeta + \mathbf{J}_2^{\zeta T} \mathbf{D}_1^\eta + \mathbf{J}_1^{\eta\zeta T} \mathbf{D}_2^0 + \mathbf{D}_2^{0T} \mathbf{D}_1^{\eta\zeta} + \mathbf{D}_1^{\eta T} \mathbf{D}_2^\zeta \\ 0 \\ 1/4 \sum_{K=J}^M \eta^K \zeta^K (\mathbf{J}_1^{KT} \mathbf{D}_3^K + \mathbf{J}_3^{KT} \mathbf{D}_1^K + \mathbf{D}_1^{KT} \mathbf{D}_3^K) \end{bmatrix} \quad (\text{A.6})$$

$$\hat{\mathbf{E}}_c^{\xi\zeta} = \begin{bmatrix} 0 \\ \mathbf{J}_2^{0T} \mathbf{D}_2^{\xi\zeta} + \mathbf{J}_2^{\xi T} \mathbf{D}_2^\zeta + \mathbf{J}_2^{\zeta T} \mathbf{D}_2^\xi + \mathbf{J}_2^{\xi\zeta T} \mathbf{D}_2^0 + \mathbf{D}_2^{0T} \mathbf{D}_2^{\xi\zeta} + \mathbf{D}_2^{\xi T} \mathbf{D}_2^\zeta \\ 0 \\ \mathbf{J}_1^{0T} \mathbf{D}_2^{\xi\zeta} + \mathbf{J}_2^{\eta T} \mathbf{D}_1^\zeta + \mathbf{J}_1^{\zeta T} \mathbf{D}_2^\xi + \mathbf{J}_2^{\xi\zeta T} \mathbf{D}_1^0 + \mathbf{D}_1^{0T} \mathbf{D}_2^{\xi\zeta} + \mathbf{D}_2^{\xi T} \mathbf{D}_1^\zeta \\ 1/4 \sum_{K=E}^H \xi^K \zeta^K (\mathbf{J}_2^{KT} \mathbf{D}_3^K + \mathbf{J}_3^{KT} \mathbf{D}_2^K + \mathbf{D}_2^{KT} \mathbf{D}_3^K) \\ 0 \end{bmatrix} \quad (\text{A.7})$$

$$\hat{\mathbf{B}}_{cl}^0 = \begin{bmatrix} g_{1I} \tilde{\mathbf{J}}_1^{0T} \\ g_{2I} \tilde{\mathbf{J}}_2^{0T} \\ 1/4 \sum_{K=A}^D N_{I,\zeta}^K \tilde{\mathbf{J}}_3^{KT} \\ g_{1I} \tilde{\mathbf{J}}_2^{0T} + g_{2I} \tilde{\mathbf{J}}_1^{0T} \\ 1/4 \sum_{K=E}^H (N_{I,\eta}^K \tilde{\mathbf{J}}_3^{KT} + N_{I,\zeta}^K \tilde{\mathbf{J}}_2^{KT}) \\ 1/4 \sum_{K=J}^M (N_{I,\xi}^K \tilde{\mathbf{J}}_3^{KT} + N_{I,\zeta}^K \tilde{\mathbf{J}}_1^{KT}) \end{bmatrix} \quad (\text{A.8})$$

$$\hat{\mathbf{B}}_{cl}^\zeta = \begin{bmatrix} h_{3I} \tilde{\mathbf{J}}_1^{0T} + g_{1I} \tilde{\mathbf{J}}_1^{\zeta T} \\ h_{2I} \tilde{\mathbf{J}}_2^{0T} + g_{2I} \tilde{\mathbf{J}}_2^{\zeta T} \\ 0 \\ h_{2I} \tilde{\mathbf{J}}_1^{0T} + h_{3I} \tilde{\mathbf{J}}_2^{0T} + g_{2I} \tilde{\mathbf{J}}_1^{\zeta T} + g_{1I} \tilde{\mathbf{J}}_2^{\zeta T} \\ 1/4 \sum_{K=E}^H \zeta^K (N_{I,\eta}^K \tilde{\mathbf{J}}_3^{KT} + N_{I,\zeta}^K \tilde{\mathbf{J}}_2^{KT}) \\ 1/4 \sum_{K=J}^M \zeta^K (N_{I,\xi}^K \tilde{\mathbf{J}}_3^{KT} + N_{I,\zeta}^K \tilde{\mathbf{J}}_1^{KT}) \end{bmatrix} \quad (\text{A.9})$$

$$\hat{\mathbf{B}}_{cl}^{\zeta\zeta} = \begin{bmatrix} h_{3I} \tilde{\mathbf{J}}_1^{\zeta T} \\ h_{2I} \tilde{\mathbf{J}}_2^{\zeta T} \\ 0 \\ h_{2I} \tilde{\mathbf{J}}_1^{\zeta T} + h_{3I} \tilde{\mathbf{J}}_2^{\zeta T} \\ 0 \\ 0 \end{bmatrix} \quad (\text{A.10})$$

$$\hat{\mathbf{B}}_{cl}^{\xi} = \begin{bmatrix} 0 \\ h_{1I}\tilde{\mathbf{J}}_2^{0T} + g_{2I}\tilde{\mathbf{J}}_2^{\zeta T} \\ 1/4\sum_{K=A}^D \xi^K N_{I,\eta}^K \tilde{\mathbf{J}}_3^{KT} \\ h_{1I}\tilde{\mathbf{J}}_2^{0T} + g_{1I}\tilde{\mathbf{J}}_2^{\zeta T} \\ 1/4\sum_{K=E}^H \xi^K (N_{I,\eta}^K \tilde{\mathbf{J}}_3^{KT} + N_{I,\zeta}^K \tilde{\mathbf{J}}_2^{KT}) \\ 0 \end{bmatrix} \quad (\text{A.11})$$

$$\hat{\mathbf{B}}_{cl}^{\eta} = \begin{bmatrix} h_{1I}\tilde{\mathbf{J}}_1^{0T} + g_{1I}\tilde{\mathbf{J}}_1^{\eta T} \\ 0 \\ 1/4\sum_{K=A}^D \eta^K N_{I,\zeta}^K \tilde{\mathbf{J}}_3^{KT} \\ h_{1I}\tilde{\mathbf{J}}_2^{0T} + g_{2I}\tilde{\mathbf{J}}_1^{\eta T} \\ 0 \\ 1/4\sum_{K=J}^M \eta^K (N_{I,\xi}^K \tilde{\mathbf{J}}_3^{KT} + N_{I,\zeta}^K \tilde{\mathbf{J}}_1^{KT}) \end{bmatrix} \quad (\text{A.12})$$

$$\hat{\mathbf{B}}_{cl}^{\eta\zeta} = \begin{bmatrix} h_{4I}\tilde{\mathbf{J}}_1^{0T} + h_{3I}\tilde{\mathbf{J}}_1^{\eta T} + h_{1I}\tilde{\mathbf{J}}_1^{\zeta T} + g_{1I}\tilde{\mathbf{J}}_1^{\eta\zeta T} \\ 0 \\ 0 \\ h_{4I}\tilde{\mathbf{J}}_2^{0T} + h_{2I}\tilde{\mathbf{J}}_1^{\eta T} + h_{1I}\tilde{\mathbf{J}}_2^{\zeta T} + g_{2I}\tilde{\mathbf{J}}_1^{\eta\zeta T} \\ 0 \\ 1/4\sum_{K=J}^M \eta^K \zeta^K (N_{I,\xi}^K \tilde{\mathbf{J}}_3^{KT} + N_{I,\zeta}^K \tilde{\mathbf{J}}_1^{KT}) \end{bmatrix} \quad (\text{A.13})$$

$$\hat{\mathbf{B}}_{cl}^{\xi\zeta} = \begin{bmatrix} 0 \\ h_{4I}\tilde{\mathbf{J}}_2^{0T} + h_{2I}\tilde{\mathbf{J}}_2^{\xi T} + h_{1I}\tilde{\mathbf{J}}_2^{\zeta T} + g_{2I}\tilde{\mathbf{J}}_2^{\xi\zeta T} \\ 0 \\ h_{4I}\tilde{\mathbf{J}}_1^{0T} + h_{3I}\tilde{\mathbf{J}}_2^{\xi T} + h_{1I}\tilde{\mathbf{J}}_1^{\zeta T} + g_{1I}\tilde{\mathbf{J}}_2^{\xi\zeta T} \\ 1/4\sum_{K=E}^H \xi^K \zeta^K (N_{I,\eta}^K \tilde{\mathbf{J}}_3^{KT} + N_{I,\zeta}^K \tilde{\mathbf{J}}_2^{KT}) \\ 0 \end{bmatrix} \quad (\text{A.14})$$

$$\mathcal{B}_c^* = \mathcal{B}_c^0 + \zeta \mathcal{B}_c^{\xi} + \zeta^2 \mathcal{B}_c^{\xi\zeta} \quad (\text{A.15})$$

$$\mathcal{B}_c^{\text{hg}} = \xi \mathcal{B}_c^{\xi} + \eta \mathcal{B}_c^{\eta} + \xi \eta \mathcal{B}_c^{\xi\eta} + \eta \zeta \mathcal{B}_c^{\eta\zeta} + \xi \zeta \mathcal{B}_c^{\xi\zeta} \quad (\text{A.16})$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \dots & \mathbf{H}_{18} \\ \mathbf{H}_{21} & \mathbf{H}_{22} & & \\ \vdots & & \ddots & \\ \mathbf{H}_{81} & & & \mathbf{H}_{88} \end{bmatrix} \quad (\text{A.17})$$

$$H_{IJ}^{0*} = \hat{\mathbf{S}}^{*T} \mathbf{T}^0 \begin{bmatrix} g_{1I}g_{1J} \\ g_{2I}g_{2J} \\ 1/4\sum_{K=A}^D N_{I,\zeta}^K N_{J,\zeta}^K \\ g_{1I}g_{2J} + g_{2I}g_{1J} \\ 1/4\sum_{K=E}^H (N_{I,\eta}^K N_{J,\zeta}^K + N_{I,\zeta}^K N_{J,\eta}^K) \\ 1/4\sum_{K=J}^M (N_{I,\xi}^K N_{J,\zeta}^K + N_{I,\zeta}^K N_{J,\xi}^K) \end{bmatrix} \quad (\text{A.18})$$

Appendix B

Difficulty related to the Discretization of SB9 and SB7 in thickness

B.1 The pinch

In the theoretical part on the formulation of SB9 and SB7 we have seen the writing of the generalized nodal forces when a SB9 or SB7 element is subjected to a pressure loading applied on its upper and/or lower faces (figure B.1). Indeed, in the linear framework with the quadratic interpolation of the transverse displacement, we can assume that the pinching strain is linear $\hat{\epsilon}_{33}$, which allows to assume also a linear σ_{33}^p pinching strain.

$$\hat{\epsilon}_{33} = \frac{\hat{w}_9^+ - \hat{w}_9^-}{h} + 2\zeta \frac{\hat{w}_9^+ - \hat{w}_9^- - 2\hat{w}_9}{h} \quad (\text{B.1})$$

$$\sigma_{33}^p = \frac{1}{2}(\zeta - 1)P_i - \frac{1}{2}(\zeta + 1)P_s \quad (\text{B.2})$$

Since the ninth node is only subject to translation in the thickness direction of the plate, the application of pressure causes the generalized forces to be written as follows, considering only the components in the thickness direction (which are non-zero):

$$W_{ext} - W_{int} = 0 \quad (\text{B.3})$$

$$\sum_{i=1}^8 F_i \hat{w}_i + F_9 \hat{w}_9 = \iint_S (P_i - P_s) \hat{w}_i ds + \iiint_V \hat{\epsilon}_{33} \cdot \sigma_{33}^p dV \quad (\text{B.4})$$

$$\begin{aligned} F_i &= +\frac{1}{4}P_i A_0, & i &= 1, 2, 3, 4 \\ F_j &= -\frac{1}{4}P_s A_0, & j &= 5, 6, 7, 8 \\ F_9 &= \frac{2}{3}(P_s - P_i) A_0 \end{aligned} \quad (\text{B.5})$$

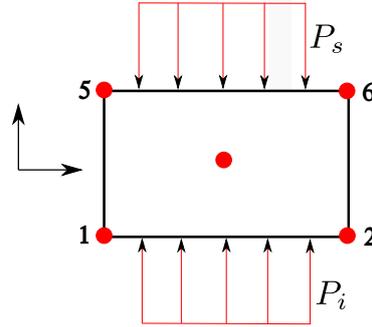


Figure B.1: SB9 finite element under pressure

The equations (B.5) are written considering the relative displacement of the ninth node. So far, there is nothing problematic. Knowing the applied pressures, we are able to determine the generalized nodal forces. The difficulty comes when one wishes to superpose two SB9 or SB7 elements. To illustrate that, let us consider two SB9 finite elements superposed in the direction of the thickness of a plate subjected to pressure on its upper and lower faces (P_s and P_i respectively, see figure B.2).

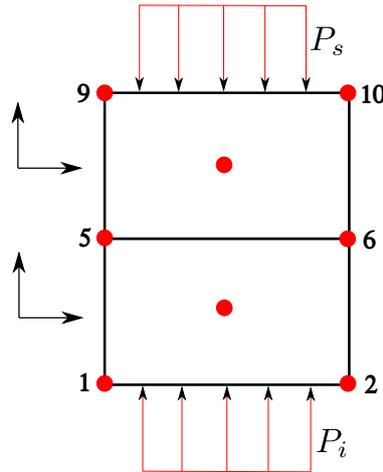


Figure B.2: 2 superposed SB9 finite element under pressure

If we call p , the pressure at the interface between the two elements, it is possible to write the generalized nodal forces for each of the SB9 elements. Considering only the direction where the components are non-zero, these efforts are written:

- Element 1

$$\begin{aligned}
 F_i &= +\frac{1}{4}P_i A_0, & i &= 1, 2, 3, 4 \\
 F_j &= -\frac{1}{4}p A_0, & j &= 5, 6, 7, 8 \\
 F_9^1 &= \frac{2}{3}(p - P_i) A_0
 \end{aligned} \tag{B.6}$$

- Element 2

$$\begin{aligned}
 F_j &= +\frac{1}{4}pA_0, & j &= 5, 6, 7, 8 \\
 F_k &= -\frac{1}{4}P_sA_0, & k &= 9, 10, 11, 12 \\
 F_9^2 &= \frac{2}{3}(P_s - p)A_0
 \end{aligned} \tag{B.7}$$

Note that we have assumed that the area A_0 is the same for both elements. However, this has no impact on the argumentation. What matters is the surface of the interface shared by the two elements. Now, if we want to put together these generalized nodal efforts, it looks like this:

$$\begin{aligned}
 F_i &= +\frac{1}{4}P_iA_0, & i &= 1, 2, 3, 4 \\
 F_j &= -\frac{1}{4}pA_0 + \frac{1}{4}pA_0 = 0, & j &= 5, 6, 7, 8 \\
 F_k &= -\frac{1}{4}P_sA_0, & k &= 9, 10, 11, 12 \\
 F_9^1 &= \frac{2}{3}(p - P_i)A_0 \\
 F_9^2 &= \frac{2}{3}(P_s - p)A_0
 \end{aligned} \tag{B.8}$$

We can see, in this assembly, that the pressure at the interface is always present, i.e. it does not disappear as it is the case with classical 3D finite elements (HEXA8 for example). Since we don't know the p pressure, we have an additional unknown in the equilibrium system which becomes *hypostatic*. This is an obstacle in the discretization of these elements in their thickness.

This difficulty can be solved by the following method. If we look at the equations more closely we realize that we are facing a problem of interface continuity. This can be solved if we find a way to impose the continuity of the pinch stress. One way to do this would be to add an additional equation to this *hypostatic* system to make it *isostatic*. This could be done by imposing a condition of C^0 continuity of the pinch deformations at the interface between two elements, i.e. C^1 continuity of the displacement $w(z)$:

$$\varepsilon_{zz}^1(\zeta = 1) = \varepsilon_{zz}^2(\zeta = -1) \tag{B.9}$$

This condition can be rewritten as the sum of two scalar products:

$$\langle M \rangle U_Z + \langle H \rangle W = 0 \tag{B.10}$$

So for each interface unknown p , we have an additional interface equation which then allows us to solve the system. Let's see in detail how we can do this.

B.1.1 Generalized nodal pressure forces for two superposed elements

The question that arises is what is the pressure p at the interface between the two elements. Considering the result of the previous section, the 25th component of the generalized effort of each element can be written as follows: To simplify the demonstration, we assume that the two elements have the same surface area and that this surface area is $A_0 = 1$:

- Element 1

$$\begin{aligned} F_i &= +\frac{1}{4}P_i A_0, & i &= 1, 2, 3, 4 \\ F_j &= -\frac{1}{4}p A_0, & j &= 5, 6, 7, 8 \\ F_9^1 &= \frac{2}{3}(p - P_i) A_0 \end{aligned} \quad (\text{B.11})$$

- Element 2

$$\begin{aligned} F_j &= +\frac{1}{4}p A_0, & j &= 5, 6, 7, 8 \\ F_k &= -\frac{1}{4}P_s A_0, & k &= 9, 10, 11, 12 \\ F_9^2 &= \frac{2}{3}(P_s - p) A_0 \end{aligned} \quad (\text{B.12})$$

So we have an unknown in the writing of generalized nodal pressure efforts. The objective of this part is to propose a method to calculate the interface pressure. For this we assemble the equations of these elements.

B.1.2 Assembly of the elementary matrices of the 2 interfacing elements

$$\begin{bmatrix} K_{XX} & K_{XY} & K_{XZ} & 0 \\ K_{YX} & K_{YY} & K_{YZ} & 0 \\ K_{ZX} & K_{ZY} & K_{ZZ} & K_{Zw} \\ 0 & 0 & K_{Zw} & K_{ww} \end{bmatrix} \begin{bmatrix} U_X \\ U_Y \\ U_Z \\ W \end{bmatrix} = \begin{bmatrix} F_X \\ F_Y \\ F_Z \\ F_W \end{bmatrix} \quad (\text{B.13})$$

We are here in the presence of a system of equations with 38 equations and 39 unknowns (38 kinematic unknowns and one static unknown p). To identify the static unknown p , we decompose the last line of the equation system (B.13) into two scalar equations:

$$K_{Zw}U_Z + K_{ww}W = F_W \quad (\text{B.14})$$

$$\begin{cases} K_{Zw}^1 U_Z + K_{ww}^1 W = F_W^1 \\ K_{Zw}^2 U_Z + K_{ww}^2 W = F_W^2 \end{cases} \quad (\text{B.15})$$

$$\begin{cases} K_{Zw}^1 U_Z + K_{ww}^1 W = F_9^1 = \frac{2}{3}(p - P_i) \\ K_{Zw}^2 U_Z + K_{ww}^2 W = F_9^1 = \frac{2}{3}(P_s - p) \end{cases} \quad (\text{B.16})$$

To make the unknown p disappear in one of the above equations, the first line is replaced by the sum of the two lines. The system becomes:

$$\begin{cases} \langle K_{Zw}^1 + K_{Zw}^2 \rangle U_Z + \langle K_{ww}^1 + K_{ww}^2 \rangle W = \frac{2}{3}(P_s - P_i) \\ \langle K_{Zw}^2 \rangle U_Z + \langle K_{ww}^2 \rangle W = \frac{2}{3}(P_s - p) \end{cases} \quad (\text{B.17})$$

The overall system can thus be written as follows:

$$\begin{bmatrix} K_{XX} & K_{XY} & K_{XZ} & 0 \\ K_{YX} & K_{YY} & K_{YZ} & 0 \\ K_{ZX} & K_{ZY} & K_{ZZ} & K_{Zw} \\ 0 & 0 & \langle K_{Zw}^1 + K_{Zw}^2 \rangle & \langle K_{ww}^1 + K_{ww}^2 \rangle \\ 0 & 0 & \langle K_{Zw}^2 \rangle & \langle K_{ww}^2 \rangle \end{bmatrix} \begin{bmatrix} U_X \\ U_Y \\ U_Z \\ W \end{bmatrix} = \begin{bmatrix} F_X \\ F_Y \\ F_Z \\ \frac{2}{3}(P_s - P_i) \\ \frac{2}{3}P_s \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{2}{3}p \end{bmatrix} \quad (\text{B.18})$$

By posing:

$$\bar{F}_W^1 = \frac{2}{3}(P_s - P_i) \quad \bar{F}_W^2 = \frac{2}{3}P_s \quad P = \frac{2}{3}p \quad (\text{B.19})$$

The system is written

$$\begin{bmatrix} K_{XX} & K_{XY} & K_{XZ} & 0 \\ K_{YX} & K_{YY} & K_{YZ} & 0 \\ K_{ZX} & K_{ZY} & K_{ZZ} & K_{Zw} \\ 0 & 0 & \langle K_{Zw}^1 + K_{Zw}^2 \rangle & \langle K_{ww}^1 + K_{ww}^2 \rangle \\ 0 & 0 & \langle K_{Zw}^2 \rangle & \langle K_{ww}^2 \rangle \end{bmatrix} \begin{bmatrix} U_X \\ U_Y \\ U_Z \\ W \end{bmatrix} = \begin{bmatrix} F_X \\ F_Y \\ F_Z \\ \bar{F}_W^1 \\ \bar{F}_W^2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ P \end{bmatrix} \quad (\text{B.20})$$

This system still has 38 equations for 39 unknown. To have an isostatic system, we need an additional equation.

B.1.3 Interface continuity hypothesis of normal deformations

To have an additional equation, let's impose a C^0 continuity condition of pinch deformations at the interface between two elements, i.e. a C^1 continuity of displacement $w(z)$:

$$\varepsilon_{zz}^1(\zeta = 1) = \varepsilon_{zz}^2(\zeta = -1) \quad (\text{B.21})$$

This condition can be rewritten as a scalar product:

$$\langle M \rangle U_Z + \langle H \rangle W = 0 \quad (\text{B.22})$$

By adding this additional equation to the system (B.24), we obtain a system with as many equations as there are unknowns (39-39). We can then pass the unknown P to the left

of the equation and rewrite the global system as follows:

$$\left[\begin{array}{cccc|c} K_{XX} & K_{XY} & K_{XZ} & 0 & 0 \\ K_{YX} & K_{YY} & K_{YZ} & 0 & 0 \\ K_{ZX} & K_{ZY} & K_{ZZ} & K_{Zw} & 0 \\ 0 & 0 & \langle K_{Zw}^1 + K_{Zw}^2 \rangle & \langle K_{ww}^1 + K_{ww}^2 \rangle & 0 \\ 0 & 0 & \langle K_{Zw}^2 \rangle & \langle K_{ww}^2 \rangle & 1 \\ \hline 0 & 0 & \langle M \rangle & \langle H \rangle & 0 \end{array} \right] \begin{bmatrix} U_X \\ U_Y \\ U_Z \\ w_9^1 \\ w_9^2 \\ P \end{bmatrix} = \begin{bmatrix} F_X \\ F_Y \\ F_Z \\ \bar{F}_W^1 \\ \bar{F}_W^2 \\ 0 \end{bmatrix} \quad (\text{B.23})$$

By posing:

$$\bar{U} = \begin{bmatrix} U_X \\ U_Y \\ U_Z \\ w_9^1 \\ w_9^2 \end{bmatrix}; \quad \bar{F} = \begin{bmatrix} F_X \\ F_Y \\ F_Z \\ \bar{F}_W^1 \\ \bar{F}_W^2 \end{bmatrix}; \quad \bar{C} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}; \quad \bar{L} = [0 \quad 0 \quad \langle M \rangle \quad \langle H \rangle] \quad (\text{B.24})$$

and

$$\bar{K} = \begin{bmatrix} K_{XX} & K_{XY} & K_{XZ} & 0 \\ K_{YX} & K_{YY} & K_{YZ} & 0 \\ K_{ZX} & K_{ZY} & K_{ZZ} & K_{Zw} \\ 0 & 0 & \langle K_{Zw}^1 + K_{Zw}^2 \rangle & \langle K_{ww}^1 + K_{ww}^2 \rangle \\ 0 & 0 & \langle K_{Zw}^2 \rangle & \langle K_{ww}^2 \rangle \end{bmatrix} \quad (\text{B.25})$$

Solving the system of equation (B.23) is equivalent to writing the next block system of equation:

$$\begin{bmatrix} \bar{K} & \bar{C} \\ \bar{L} & 0 \end{bmatrix} \begin{bmatrix} \bar{U} \\ \bar{P} \end{bmatrix} = \begin{bmatrix} \bar{F} \\ 0 \end{bmatrix} \quad (\text{B.26})$$

The interface pressure is then calculated using the following scalar equation:

$$\bar{a}P = \bar{b} \quad \text{avec} \quad \begin{cases} \bar{a} = \bar{L} \bar{K}^{-1} \bar{C} \\ \bar{b} = \bar{L} \bar{K}^{-1} \bar{F} \end{cases} \quad (\text{B.27})$$

This method works in theory but in practice it is very complicated to implement numerically, especially for software that already has an established architecture and often not simple to restructure as `code_aster`. This method also has the disadvantage of making the resolution matrix non-symmetric. This will require an adapted solver. It also requires to know the neighboring elements of each element (in thickness) which requires more resources in terms of data storage. It also makes the linear systems to be solved considerably heavier, which is costly in terms of computing time.

Another more attractive method in `code_aster`, would be to add additional nodes in the middle of the lower and upper faces of the element. Indeed, assuming that the pressure is uniform on the face of an element, we can consider it as an unknown and have it carried by a node in the middle of the face. This node will thus only be associated

with static unknowns, as its displacements do not enter into the system of equations. Since this node is coincident between two superposed elements, the two elements see the same pressure at the interface. This solves the problem of pressure continuity at the interface. Element SB9 would become, in a sense, an SB11.

In any case the solid-shell elements are implemented in order to model the behavior of thin structures with simple elements and a relatively coarse mesh compared to what would be required for modeling with standard 3D elements. Thus we recommend using SB9 and SB7 only in cases where these elements can work precisely, without the need to discretize in the thickness, that is in the case of thin and medium thick structures.

B.2 The transverse shear of the SB7

In the formulation of SB7 the transverse shear strain is written as follows:

$$\varepsilon^\perp = \frac{5}{4} (1 - \zeta^2) \left[\mathbf{B}_0^\perp + \xi \mathbf{B}_\xi^\perp + \eta \mathbf{B}_\eta^\perp \right] \cdot \mathbf{U}_e \quad (\text{B.28})$$

Where \mathbf{U}_e is the local displacement vector, $\left[\mathbf{B}_0^\perp + \xi \mathbf{B}_\xi^\perp + \eta \mathbf{B}_\eta^\perp \right]$ the assumed deformation gradient and $\frac{5}{4} (1 - \zeta^2)$ Reissner's function. Multiplying the deformation gradient assumed by the Reissner function makes it possible to respect the static admissibility of the upper and lower faces of the element when these are free of stress. This gives the parabolic form which is nullified at the free edges. However, such a formulation does not hold if the edges are not free, or if SB7 elements are to be superposed in the thickness direction. In fact, by superposing two SB elements, the shape of the transverse shear deformation field resembles the shape in the following figure B.3 :

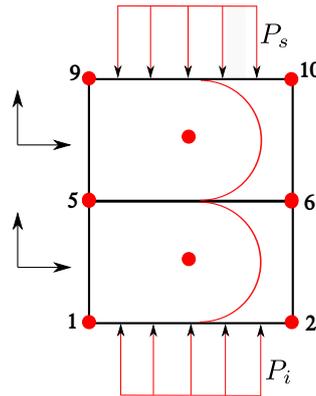


Figure B.3: Shear profile

We have two parabolic pieces that both nullify at the interface, indicating that the transverse shear at this location is zero. That's not correct. This transverse shear deformation will then have to be rewritten so that there are no inconsistencies during discretization in thickness. One could consider omitting the multiplication of the shear

B. Difficulty related to the Discretization of SB9 and SB7 in thickness

by the Reissner function. Indeed, SB7 is also a solid element and such an interpolation does not exist with prism type solid elements.

Appendix C

Classical shell theory and associated finite elements

C.1 Kinematics

C.1.1 Displacement field

Let's consider two points M and P of a segment of the plate such that M is in the mean plane $(O, \underline{e}_1, \underline{e}_2)$ and P has a z dimension with respect to this mean plane. The equiprojectivity hypothesis allows to express the displacement field of the point P with respect to the displacement field of the point M supposedly known :

$$\mathbf{u}_P = \begin{bmatrix} u_1 + z\beta_2 \\ u_2 - z\beta_1 \\ u_3 \end{bmatrix} \quad (\text{C.1})$$

$u_1, u_2, u_3, \beta_1, \beta_2$ being respectively the three displacements of the point M and the two rotations of the segment MP.

C.1.2 Strain field

As in the general case, the deformations are obtained by calculating the symmetrized gradient of the displacements, taking into account the kinematic assumptions mentioned in the previous paragraph. We also remain under the hypothesis of small perturbations. In Cartesian coordinates, the strain field is written in Cartesian coordinates :

$$\varepsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i}) \quad (\text{C.2})$$

For simplicity reason and taking into account the local frame, it is possible to dissociate in-plane deformations $(\underline{e}_1, \underline{e}_2)$ and transverse deformations (out-of-plane):

In plane strain

$$\boldsymbol{\varepsilon}^{2D} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix} = \begin{bmatrix} u_{1,1} \\ u_{2,2} \\ u_{1,2} + u_{2,1} \end{bmatrix} + z \begin{bmatrix} \beta_{2,1} \\ \beta_{1,2} \\ \beta_{2,2} - \beta_{1,1} \end{bmatrix} = \mathbf{e} + z\boldsymbol{\chi} \quad (\text{C.3})$$

with

$$\mathbf{e} = \begin{bmatrix} u_{1,1} \\ u_{2,2} \\ u_{1,2} + u_{2,1} \end{bmatrix} \quad \text{et} \quad \boldsymbol{\chi} = \begin{bmatrix} \beta_{2,1} \\ -\beta_{1,2} \\ \beta_{2,2} - \beta_{1,1} \end{bmatrix}$$

\mathbf{e} et $\boldsymbol{\chi}$ being the membrane and bending deformations, respectively.

Transverse shear strain

$$\boldsymbol{\gamma} = \begin{bmatrix} \beta_2 + w_{,1} \\ \beta_1 + w_{,2} \end{bmatrix} \quad (\text{C.4})$$

The $\boldsymbol{\varepsilon}^{2D}$ deformations are linear in z and the $\boldsymbol{\gamma}$ transverse shear deformations are constant across the thickness. Consequently, the shear stresses associated with these deformations will be constant across the thickness and will not meet the boundary conditions on the top and bottom faces of the plate. To remedy this a correction is applied to the distribution of these deformations to have the actual transverse shear deformations, respecting the static admissibility. The real deformations are then written as follows :

$$\boldsymbol{\varepsilon}^\perp = \begin{bmatrix} 2\varepsilon_{13} \\ 2\varepsilon_{23} \end{bmatrix} = f(z)\boldsymbol{\gamma} = f(z) \begin{bmatrix} \beta_2 + w_{,1} \\ \beta_1 + w_{,2} \end{bmatrix} \quad (\text{C.5})$$

$f(z)$ is the transverse shear correction function or Reissner function.

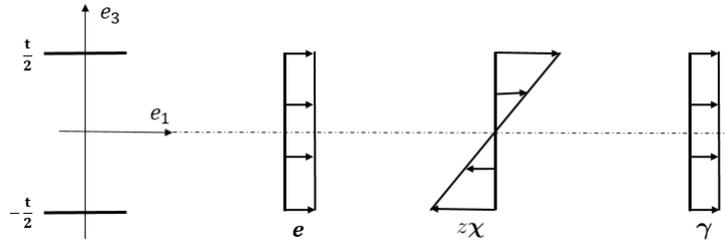


Figure C.1: Allure des déformations.

C.1.3 Through-thickness integrated stress

Under the assumption of plane constraints and in using Voigt notations, the stress tensor can be decomposed into two vectors: in-plane constraints $\boldsymbol{\sigma}^{2D}$ and out-of-plane (transverse) constraints $\boldsymbol{\sigma}^\perp$:

$$\boldsymbol{\sigma}^{2D} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} ; \boldsymbol{\sigma}^\perp = \begin{bmatrix} \sigma_{13} \\ \sigma_{23} \end{bmatrix} \quad (\text{C.6})$$

The integrated stresses are calculated by integrating the stresses on the thickness :

$$\begin{aligned}\mathbf{N} &= \begin{bmatrix} N_1 \\ N_2 \\ N_{12} \end{bmatrix} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} dz \\ \mathbf{M} &= \begin{bmatrix} M_1 \\ M_2 \\ M_{12} \end{bmatrix} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} z dz \\ \mathbf{T} &= \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \begin{bmatrix} \sigma_{13} \\ \sigma_{23} \end{bmatrix} dz\end{aligned}\quad (\text{C.7})$$

N_1, N_2, N_{12} : membrane forces (en N/m)

M_1, M_2, M_{12} : bending forces or moments (en $N \cdot m/m$)

T_1, T_2 : resultant shear forces or shearing forces (en N/m)

C.1.4 Constitutive law

The stress-strain relationship is written as follows :

$$\sigma^{2D} = H \cdot \varepsilon^{2D} + \sigma_0^{2D} ; \sigma^\perp = \mathbf{H}_\tau \cdot \varepsilon^\perp + \sigma_0^\perp \quad (\text{C.8})$$

If the material is homogeneous, isotropic and elastic, it comes:

$$\mathbf{H} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} ; \mathbf{H}_\tau = \frac{E}{2(1+\nu)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (\text{C.9})$$

C.1.5 Resulting effort-strain relationship

It is possible to link the integrated stresses directly to the deformation :

$$\begin{cases} \mathbf{N} = \mathbf{H}_m \cdot e + \mathbf{N}_0 \\ \mathbf{M} = \mathbf{H}_b \cdot \chi + \mathbf{M}_0 \\ \mathbf{T} = \mathbf{H}_s \cdot \gamma + \mathbf{T}_0 \end{cases} \quad (\text{C.10})$$

with

$$\mathbf{H}_m = \int_{-\frac{t}{2}}^{\frac{t}{2}} \mathbf{H} dz = \frac{Eh}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} ; \mathbf{H}_b = \int_{-\frac{t}{2}}^{\frac{t}{2}} \mathbf{H} z^2 dz = \frac{h^3}{12} \mathbf{H} \quad (\text{C.11})$$

and

$$\mathbf{N}_0 = \int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_0^{2D} dz ; \mathbf{M}_0 = \int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_0^{2D} z dz ; \mathbf{T}_0 = \int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_0^\perp dz \quad (\text{C.12})$$

The \mathbf{H}_s matrix takes into account the \mathbf{H}_τ components and a k coefficient called shear strain correction coefficient, obtained by integrating the Reissner function, explained in the following section.

C.1.6 Transverse shear correction

As noted above, the transverse shear deformations γ are constant across the thickness. This results in non-admissible shear stresses on the top and bottom faces of the plate. To remedy this, a correcting constant is applied to the distribution of these deformations in order to have the actual transverse shear deformations meet the static admissibility. Let's consider a homogeneous and isotropic plate. The constitutive matrix \mathbf{H}_τ linking the shear forces \mathbf{T} to the γ deformations is defined taking into account a continuous shear stress distribution σ^\perp respecting the equilibrium conditions. To simplify the reasoning, the plate is assumed to be free of initial stress and is placed in the context of pure bend/shear behavior. Thus the following equations are valid :

$$\mathbf{N} = \mathbf{N}_0 = \mathbf{M}_0 = 0 \quad (\text{C.13})$$

From the equations ((C.10)) we can write $\chi = \mathbf{H}_b^{-1} \cdot \mathbf{M}$, then considering the relation (C.9):

$$\begin{aligned} \sigma^{2D} &= \mathbf{H}(\mathbf{e} + z\chi) \\ &= \mathbf{H}(z\mathbf{H}_b^{-1} \cdot \mathbf{M}) \end{aligned} \quad (\text{C.14})$$

- **Calculation of the matrix \mathbf{H}_s**

The method used is based on static and energy equilibrium considerations so that the transverse shear stiffness of the plate model corresponds as closely as possible to that defined by the theory of three-dimensional elasticity. In other words, we define the matrix \mathbf{H}_s ((C.10)) so that the density of internal energy per unit area related to the shear strain obtained by a 3D (exact) stress σ_{13} et σ_{23} , noted U_1 , be identical to U_2 , associated with the plate model based on Reissner-Mindlin assumptions. The problem is then to find \mathbf{H}_s such that $U_1 = U_2$, with:

$$U_1 = \frac{1}{2} \int_{-\frac{t}{2}}^{\frac{t}{2}} (\sigma^\perp)^T \mathbf{H}_\tau^{-1} \sigma^\perp dz \quad \text{et} \quad U_2 = \frac{1}{2} \mathbf{T}^T \mathbf{H}_s^{-1} \mathbf{T} \quad (\text{C.15})$$

$$\mathbf{T} = \frac{1}{2} \int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma^\perp dz \quad (\text{C.16})$$

The calculation of \mathbf{H}_s also requires knowing the distribution of $\sigma^\perp(z)$ in the thickness direction. This is obtained by considering the homogeneous equilibrium equations associated with the free edge conditions :

$$\begin{cases} \sigma_{13} = - \int_{-\frac{t}{2}}^z (\sigma_{11,1} + \sigma_{12,2}) dz \\ \sigma_{23} = - \int_{-\frac{t}{2}}^z (\sigma_{12,1} + \sigma_{22,2}) dz \end{cases} \quad (\text{C.17})$$

with conditions at the edges being :

$$\sigma_{13} \left(z = -\frac{t}{2} \right) = \sigma_{23} \left(z = -\frac{t}{2} \right) = 0 \quad (\text{C.18})$$

Still in the case of pure bending/shear behaviour and using the equations ((C.14)) and ((C.11)) then by integrating over the thickness of the plate we find the distribution of $\sigma^\perp(z)$:

$$\sigma^\perp(z) = \frac{6}{t^3} \left(\frac{t^2}{4} - z^2 \right) \mathbf{T} \quad (\text{C.19})$$

from where, after integration on the thickness

$$U_1 = \frac{1}{2} \frac{6}{5t} \mathbf{T}^T \mathbf{H}_\tau^{-1} \mathbf{T} \quad (\text{C.20})$$

The equivalence $U_1 = U_2$ gives the transverse shear correction coefficient $k = \frac{5}{6}$. One can then write the stress-strain relationship in transverse shear and deduce the corrected shear deformations :

$$\sigma^\perp(z) = \frac{5}{4} \left(1 - \frac{4z^2}{t^2} \right) \mathbf{H}_\tau \gamma \quad \text{then} \quad \varepsilon^\perp(z) = \frac{5}{4} \left(1 - \frac{4z^2}{t^2} \right) \gamma \quad (\text{C.21})$$

Not that this interpolation is only valid in linear isotropic.

C.1.7 Principle of virtual works

Two types of external forces can be applied to a plate. A distinction is made between forces per unit length (tangential or normal to the mid-surface) applied to the plate contour and forces per unit area distributed over the mid-surface, again tangential or normal to the mid-surface. The equilibrium based on the deformation energy enables the internal forces to be related to the external forces. The principle of virtual work gives the equilibrium :

$$W = W_{int} - W_{ext} = 0 \quad (\text{C.22})$$

with

$$\begin{aligned} W_{int} &= \int_V \hat{\varepsilon}^{2D} \cdot \sigma_{2D} + \hat{\gamma} \cdot \sigma_\perp \, dV \\ W_{ext} &= \int_{S_m} \hat{\mathbf{u}} \cdot \mathbf{f} + \hat{\boldsymbol{\beta}} \cdot \mathbf{m} \, dV + \int_C \hat{\mathbf{u}} \cdot \mathbf{f}_s + \hat{\boldsymbol{\beta}} \cdot \mathbf{m}_s \, ds \end{aligned} \quad (\text{C.23})$$

\mathbf{f} and \mathbf{m} are the forces and moments per unit length applied on the mid-surface S_m and \mathbf{f}_s and \mathbf{m}_s are the forces and moments per unit length applied on the C contour of the plate. In general, it is impossible to find the analytical solutions to the equilibrium equation (C.22). In practice, to determine the displacements corresponding to the applied stresses, the finite element approximate is used.

C.1.8 Finite elements based on classical shell theory

Starting from the plate kinematics established in the section above, it is quite straightforward to propose an associated finite element formulation. The complexity of a plate finite element formulation is due to the rotational degrees of freedom considered at the

nodes of the elements and to the bending representation. An approximation of the bending effects can be done by considering elements based on standard kinematics, or by introducing additional degrees of freedom. Therefore, there are various possibilities for formulating finite plate elements. In this sub-section we focus on the main principles and will only detail the plate finite element method based on displacement formulation. Let's consider the general case of the equilibrium discretization of a plate based on Reissner/Mindlin kinematics. By defining the basic finite subspace of the shape functions $\varphi^h = Vect \{ \phi_1, \phi_2, \dots, \phi_n \}$, it is possible to write the displacement field of any point of a plate.

C.1.8.1 Discretized displacement field

The Reissner/Mindlin kinematics leads to elements with five degrees of freedom per node: two displacements in the plane u_1 and u_2 , one transverse displacement w and two rotations β_1 and β_2 . Considering a linear isoparametric plate element, the approximation of the elementary displacement is written as follows :

$$\left\{ \begin{array}{l} u = \sum_{j=1}^n \phi_j u_j = \Phi \mathbf{U}_1^e \\ v = \sum_{j=1}^n \phi_j v_j = \Phi \mathbf{U}_2^e \\ w = \sum_{j=1}^n \phi_j w_j = \Phi \mathbf{U}_3^e \\ \beta_1 = \sum_{j=1}^n \phi_j \beta_j^n = \Phi \beta_1^n \\ \beta_2 = \sum_{j=1}^n \phi_j \beta_j^n = \Phi \beta_2^n \end{array} \right. \quad (\text{C.24})$$

\mathbf{U}_i^e ($i = 1, 2, 3$) β_1^n and β_2^n are respectively the displacements and rotations at the nodes of the element.

C.1.8.2 The discretized deformations field

Membrane deformation:

$$\mathbf{e}(u) = \begin{bmatrix} u_{,1} \\ v_{,2} \\ u_{,2} + v_{,1} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix} \begin{bmatrix} \mathbf{U}_1^e \\ \mathbf{U}_2^e \end{bmatrix} \quad (\text{C.25})$$

meaning

$$\mathbf{e}(u) = \mathbf{B}_m \begin{bmatrix} \mathbf{U}_1^e \\ \mathbf{U}_2^e \end{bmatrix} \quad \text{and} \quad \mathbf{B}_m = \begin{bmatrix} \frac{\partial \Phi}{\partial x} & 0 \\ 0 & \frac{\partial \Phi}{\partial y} \\ \frac{\partial \Phi}{\partial y} & \frac{\partial \Phi}{\partial x} \end{bmatrix} \quad (\text{C.26})$$

\mathbf{B}_m is the discretized gradient operator for the membrane part.

Bending deformation

$$\chi(u) = \begin{bmatrix} \beta_{1,1} \\ \beta_{2,2} \\ \beta_{1,2} + \beta_{2,1} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix} \begin{bmatrix} \beta_1^n \\ \beta_2^n \end{bmatrix} \quad (\text{C.27})$$

meaning

$$\chi = \mathbf{B}_f \begin{bmatrix} \beta_1^n \\ \beta_2^n \end{bmatrix}$$

$$\mathbf{B}_f = \begin{bmatrix} \frac{\partial \Phi}{\partial x} & 0 \\ 0 & \frac{\partial \Phi}{\partial y} \\ \frac{\partial \Phi}{\partial y} & \frac{\partial \Phi}{\partial x} \end{bmatrix} \quad (\text{C.28})$$

\mathbf{B}_f is the discretized gradient operator for the curvature part (bending).

Transverse shear strain

$$\gamma(u) = \begin{bmatrix} \beta_1 + w_{,1} \\ \beta_2 + w_{,2} \end{bmatrix}$$

$$\gamma(u) = \begin{bmatrix} \Phi_{,1} & \Phi & 0 \\ \Phi_{,2} & 0 & \Phi \end{bmatrix} \begin{bmatrix} \mathbf{U}_3^e \\ \beta_1^n \\ \beta_2^n \end{bmatrix} = \mathbf{B}_{ct} \begin{bmatrix} \mathbf{U}_3^e \\ \beta_1^n \\ \beta_2^n \end{bmatrix} \quad (\text{C.29})$$

meaning

$$\mathbf{B}_{ct} = \begin{bmatrix} \Phi_{,1} & \Phi & 0 \\ \Phi_{,2} & 0 & \Phi \end{bmatrix} \quad (\text{C.30})$$

\mathbf{B}_{ct} is the discretized gradient operator for the transverse deformation part.

C.1.8.3 Stiffness matrix

The virtual work of the internal forces is then used to calculate the stiffness matrix of the element:

$$W_{int} = \frac{1}{2} [\mathbf{U}]^T \mathbf{K} [\mathbf{U}] \quad (\text{C.31})$$

By identification, and assuming that there is no membrane-bending coupling, it is possible to decompose the elementary stiffness matrix according to the three contributions: membrane, bending and shear.

$$\mathbf{K} = t\mathbf{K}_m + t^3\mathbf{K}_f + t\mathbf{K}_c \quad (\text{C.32})$$

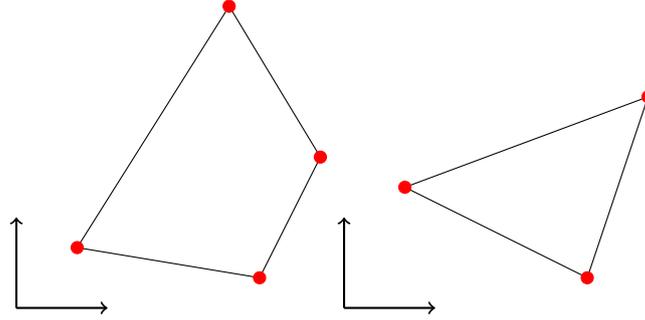


Figure C.2: Quadrangular and triangular plate finite elements

with

$$\begin{cases} \mathbf{K}_m = \mathbf{B}_m^T \mathbf{C}_m \mathbf{B}_m \\ \mathbf{K}_f = \mathbf{B}_f^T \mathbf{C}_f \mathbf{B}_f \\ \mathbf{K}_{ct} = \mathbf{B}_{ct}^T \mathbf{C}_{ct} \mathbf{B}_{ct} \end{cases} \quad (\text{C.33})$$

In other words, the overall stiffness matrix is a function of the matrices of membrane stiffness \mathbf{K}_m , bending \mathbf{K}_f and transverse shear \mathbf{K}_{ct} . t represents the thickness of the plate. The matrices \mathbf{C}_m , \mathbf{C}_f and \mathbf{C}_{ct} are behavioral matrices related to membrane, bending and transverse shear effects, respectively. For a homogeneous and isotropic material these matrices are written:

$$\mathbf{C}_m = \frac{E}{2(1+\nu)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1+\nu}{2} \end{bmatrix}, \quad \mathbf{C}_f = \frac{1}{12} \mathbf{C}_m, \quad \mathbf{C}_{ct} = \frac{E}{2(1+\nu)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (\text{C.34})$$

The generalized nodal efforts \mathbf{F}_n are then calculated by discretizing the work of the external efforts C.23. Finally, the algebraic system to be solved is the following:

$$\mathbf{K} \cdot \mathbf{U} = \mathbf{F}_{ext} \quad (\text{C.35})$$

The displacement formulation just detailed allows to define a variety of finite plate elements of triangular or quadrangular type. These elements are often referred to by their shape (Quadrangle or Triangle) and the number of nodes on which the approximation is based: for example Q4 and T3 for 4 node quadrangles and 3 node triangles (Figure C.2). In practice, the finite elements of plates formulated in this way are not directly usable to model the plates or shell type structures. Indeed, they suffer from many locking phenomena, and in particular the transverse shear locking which appears when the structure is very thin. To illustrate this transverse shear locking, let's consider a bending plate problem and the Mindlin/Reissner model. Thus, equation (C.32) becomes

$$\mathbf{K} = t \mathbf{K}_c + t^3 \mathbf{K}_f$$

It is then necessary that \mathbf{K}_c tends towards 0 when t (thickness) tends towards 0 in order to avoid that the transverse shear term becomes preponderant in the expression of the

stiffness. Otherwise, locking occurs due to the t dependence of the matrices. The element thus leads to a solution influenced solely by shear instead of being associated with the bending phenomenon. In this case there is less energy to deform the plate, resulting in an erroneous solution that is too rigid. It is the existence of this artificial stiffness that is called *shear locking*. This phenomenon appears significantly for plates that have a very low $\frac{t}{L}$ ratio.

C.1.8.4 Importance of higher order interpolation

Increasing the degrees of interpolation improves performance by reducing blockages. In Reissner/Mindlin plates a C^0 continuity is sufficient but it is possible to develop richer interpolations to improve the performance of plate finite elements, especially when modeling very thin structures in bending. The immediate approach is to increase the number of nodes to increase the number of degrees of freedom. Two families of elements are generally distinguished: elements with additional inner nodes (Lagrange) and elements with additional nodes just at the edges of the element (Serendip). The nodes inside the element, although they can intervene in the distribution of forces, are not of direct interest in the calculation codes because they do not contribute directly to the overall stiffness through the interaction with the related elements. Generally, static condensing methods are used to eliminate them. However, in Batoz [?], the authors showed that a simple increase in the number of degrees of freedom is not enough to have efficient elements. With the exact integration technique to model a square plate embedded on the contour ($L/t = 100$), it was shown that Q9 and Q16 elements can give satisfactory results. Q4, on the other hand, gives very poor results. Also when the L/t ratio becomes very low ($L/t < 100$ for example) elements Q4, Q8 and Q12 give very poor results. Elements Q9 and Q16, although a little better, also give false answers as soon as the mesh is a little distorted. Thus, despite a considerable number of degrees of freedom for some of these elements, they also suffer from the transverse shear locking phenomenon.

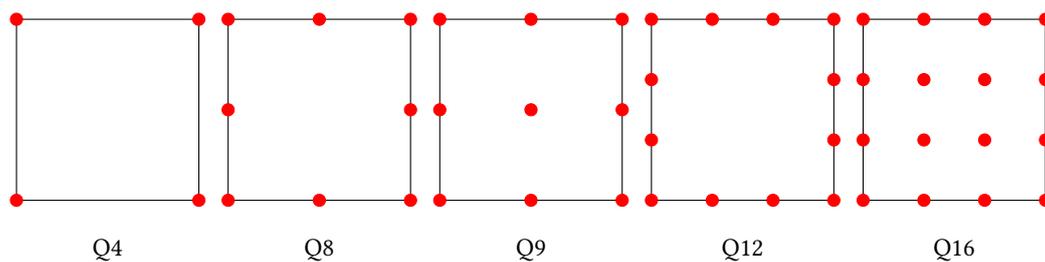


Figure C.3: Enriched quadrangular shell elements

C.1.8.5 Mixed formulations

The displacement variable formulation is subject to various locking phenomena. This is why it is often abandoned in favour of a richer formulation generally called mixed formulation. There are different classes of mixed models associated with plate problems but

generally a formulation is mixed when the equilibrium equation contains both kinematic unknowns \mathbf{u} , β of type C^0 and static variables \mathbf{N} , \mathbf{M} , \mathbf{T} of continuity C^{-1} . Thus, QM elements (Quadrangle with Mixed formulation) have been proposed. They give much better results than the classical moving quadrangles. The most noticeable improvement concerns the QM4 quadrangle and to a lesser extent the QM8 and QM12 elements. Indeed, it has been shown in [?] that QM8 and QM12 still suffer from the locking phenomenon even though it is less important than the locking of quadrangles in simple displacement formulation. The mixed formulation can also be extended to elements with triangular geometry. This is the case of Zienkiewicz's T6/3B3 element [ZIE 88].

C.1.8.6 Specific interpolation of the transverse shear

A solution to the locking problems induced by the consideration of transverse shear kinematics in Reissner/Mindlin plates consists in replacing the classical shear terms by substitution strain : this is the assumed strain method. There are several variants of the assumed strain method depending on how the condition between the actual strain γ and the substitution strain $\bar{\gamma}$ (collocation or integral form) is imposed. The most commonly used variant is the one based on an integral condition between actual and substitution deformations. This method is find in the formulation of the family of MITC elements (mixed interpolated tensoriel components) proposed in [BAN 87, BAR 81, ?]. This family consists of quadrilateral elements at 4, 8, 9 and 16 nodes, and triangular elements at 7 and 12 nodes. In the Q4 γ example a transverse shear strain energy is considered in this form :

$$U_c = \frac{1}{2} \int_A \bar{\gamma}^T \mathbf{H}_c \bar{\gamma} dA \quad (\text{C.36})$$

with

$$\bar{\gamma}_{xy} = \left\{ \begin{array}{c} \bar{\gamma}_{xz} \\ \bar{\gamma}_{yz} \end{array} \right\} = \mathbf{J} \bar{\gamma}_{\xi\eta} \quad (\text{C.37})$$

and

$$\bar{\gamma}_{\xi\eta} = \left\{ \begin{array}{c} \frac{1-\eta}{2} \gamma_{\xi}^a + \frac{1+\eta}{2} \gamma_{\xi}^c \\ \frac{1-\xi}{2} \gamma_{\eta}^b + \frac{1+\xi}{2} \gamma_{\eta}^d \end{array} \right\} \quad (\text{C.38})$$

Then along each side we connect γ and $\bar{\gamma}$ per:

$$\begin{aligned} \int_{-1}^1 (\bar{\gamma}_{\xi} - \gamma_{\xi}) d\xi &= 0 \\ \int_{-1}^1 (\bar{\gamma}_{\eta} - \gamma_{\eta}) d\eta &= 0 \end{aligned} \quad (\text{C.39})$$

The quadrangular elements whose transverse shear are thus defined are tested by several authors and give good results (patch-test, convergence, influence of distortion), see Batoz [?]. The technique of assumed strain, which is very efficient in eliminating the transverse shear locking, is also taken up by the family of DST (Discrete Shear Triangle) and DSQ (Discrete Shear Quadrangle) elements whose particularity is in the enrichment

of discrete rotational degree of freedom. However, despite the use of the assumed strain method, these elements can still show some locking issues when the modeled structure is very thin. In such a situation one can get around the locking problem by turning to the Kirchoff-Love hypothesis. The discretization of transverse shear deformations is then neglected. This is the case for the DKT and DKQ finite elements.

Appendix D

Recalls on the classic SB9 γ 25

D.1 Formulation of the SB9 γ 25

The SB9 γ 25 is a hexahedral element with nine nodes (eight vertex nodes and a central node), isoparametric and with linear interpolation. Its design is largely inspired by the solid-shell element named SHB8PS [ABE 02, ABE 09], the SB9 γ 25 uses the reduced integration method initially introduced by the work of Zienkiewicz and al [ZIE 77], Hughes and al [HUG 81a] to reduce membrane and shear locking and improve performance in terms of computing time. The resulting zero-energy modes are then stabilized by following Belytschko's numerous researches [BEL 83, BEL 84, BEL 93] on the stabilization of *hourglass* modes. To go further in the elimination of transverse shear locking, the authors [BAS 12] use the Assumed natural strain (ANS) method of transverse shear. In addition, the special and innovative feature in the design of the SB9 γ 25 was to add an extra node in the center of the solid-shell element at eight vertex nodes and 24 degrees of freedom (dof). This makes the SB9 γ 25 a nine-node and 25 dof solid-shell element. This central node has only one degree of freedom of translation normal to the mid-plane of the element. Thanks to this additional dof, a quadratic normal displacement is assumed which allows the assumption of a linear normal deformation. From then on it becomes possible to calculate a true normal stress (pinch) using a purely three-dimensional behaviour law: this is the end of the plane stress hypothesis.

In this chapter, we will detail the different phases of the design of the SB9 γ 25.

D.1.1 Variational formulation

The SB9 γ 25 element is nothing more than a classical hexahedral element enriched with an assumed natural strain (ANS) of transverse shear and pinch strain (EAS), which is accompanied by a assumed pinch stress. Therefore, the SB9 γ 25 formulation is based on the Hu-Washizu three-field formulation. Since the strain field is enriched, the new field can be written as

$$\mathbf{E} = \mathbf{E}'' + \tilde{\mathbf{E}} \quad (\text{D.1})$$

\mathbf{E}^u being the classical strain field of a hexahedral (derivation of the displacement vector) and \mathbf{E}^{eas} the enriched strain field (pinch and assumed transverse shear field). The three-field mixed variational formulation is written as follows

$$\pi^{HU}(\mathbf{u}, \mathbf{E}, \tilde{\mathbf{S}}) = \int_{\Omega_0} W(\mathbf{E}) d\Omega_0 + \int_{\Omega_0} \tilde{\mathbf{S}} : (\mathbf{E}^u - \mathbf{E}) d\Omega_0 - \int_{\Omega_0} \rho \mathbf{b} \cdot \mathbf{u} d\Omega_0 - \int_{\partial\Omega_0} \mathbf{f} \cdot \mathbf{u} d(\partial\Omega_0) \quad (\text{D.2})$$

or alternatively

$$\pi^{HU}(\mathbf{u}, \mathbf{E}^{eas}, \tilde{\mathbf{S}}) = \int_{\Omega_0} W(\mathbf{E}^u + \mathbf{E}^{eas}) d\Omega_0 - \int_{\Omega_0} \tilde{\mathbf{S}} : \mathbf{E}^{eas} d\Omega_0 - \int_{\Omega_0} \rho \mathbf{b} \cdot \mathbf{u} d\Omega_0 - \int_{\partial\Omega_0} \mathbf{f} \cdot \mathbf{u} d(\partial\Omega_0) \quad (\text{D.3})$$

where W is the constitutive law, $\tilde{\mathbf{S}}$ the assumed stress, \mathbf{b} being the body forces and \mathbf{f} surface forces. The variation in energy can then be written as follows

$$\begin{aligned} \delta\pi^{HU}(\mathbf{u}, \mathbf{E}^{eas}, \tilde{\mathbf{S}}) &= \int_{\Omega_0} \left[\frac{\partial W}{\partial \mathbf{E}} : \delta \mathbf{E}^u + \frac{\partial W}{\partial \mathbf{E}} : \delta \mathbf{E}^{eas} \right] d\Omega_0 - \int_{\Omega_0} \rho \mathbf{b} \cdot \delta \mathbf{u} d\Omega_0 \\ &\quad - \int_{\Omega_0} \tilde{\mathbf{S}} : \delta \mathbf{E}^{eas} d\Omega_0 - \int_{\Omega_0} \delta \tilde{\mathbf{S}} : \mathbf{E}^{eas} d\Omega_0 - \int_{\partial\Omega_0} \mathbf{f} \cdot \delta \mathbf{u} d(\partial\Omega_0) \end{aligned} \quad (\text{D.4})$$

D.1.2 Kinematics

D.1.2.1 Displacement field

The SB9γ25 finite element is an isoparametric element with reduced integration. The element therefore admits a privileged direction. This corresponds to the direction of its thickness along which the integration points are distributed (3 for elasticity, 5, 7 or more for plasticity).

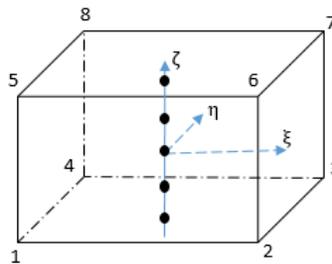


Figure D.1: Reference geometry of the element and integration points

The reduced integration gives the element zero-energy modes that will be eliminated by a stabilization stiffness matrix. In order to highlight these zero energy modes more easily, the usual shape functions are rewritten in vector form :

$$\mathbf{N} = \frac{1}{8} (\mathbf{s} + \mathbf{a}_1 \xi + \mathbf{a}_2 \eta + \mathbf{a}_3 \zeta + \mathbf{h}_1 \eta \zeta + \mathbf{h}_2 \xi \zeta + \mathbf{h}_3 \xi \eta + \mathbf{h}_4 \xi \eta \zeta) \quad (\text{D.5})$$

The vectors \mathbf{s} , \mathbf{a}_β ($\beta = 1, \dots, 3$) and \mathbf{h}_α ($\alpha = 1, \dots, 4$) are given by :

$$\begin{cases} \mathbf{s}^T = [1, 1, 1, 1, 1, 1, 1, 1] \\ \mathbf{a}_1^T = [-1, 1, 1, -1, -1, 1, 1, -1] \\ \mathbf{a}_2^T = [-1, -1, 1, 1, -1, -1, 1, 1] \\ \mathbf{a}_3^T = [-1, -1, -1, -1, 1, 1, 1, 1] \\ \mathbf{h}_1^T = [-1, -1, 1, 1, 1, 1, -1, -1] \\ \mathbf{h}_2^T = [1, -1, -1, 1, -1, 1, 1, -1] \\ \mathbf{h}_3^T = [1, -1, 1, -1, 1, -1, 1, -1] \\ \mathbf{h}_4^T = [-1, 1, -1, 1, 1, -1, 1, -1] \end{cases} \quad (\text{D.6})$$

The position and the displacement field at a point on the element are then written:

$$x_i = \sum_{l=1}^8 x_{il} N_l(\xi, \eta, \zeta) = x_{il} N_l(\xi, \eta, \zeta) \quad (\text{D.7})$$

$$u_i = \sum_{l=1}^8 u_{il} N_l(\xi, \eta, \zeta) = u_{il} N_l(\xi, \eta, \zeta) \quad (\text{D.8})$$

The small subscripts i range from one to three and represent the directions of the spatial coordinates while capital subscripts l range from one to eight corresponding to the nodes of the feature.

D.1.2.2 Deformation-displacement gradient

The interpolation of the displacement field (equation (D.8)) allows to write the relations linking the strain field to the nodal displacements. The linear part of the strain tensor is written:

$$\varepsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i}) \quad (\text{D.9})$$

Combining equations (D.5), (D.8) and (D.9) we are able to develop the displacement field as a linear combination of linear terms in x_i and terms involving functions h_α :

$$\begin{cases} \mathbf{u}_i = a_{0i} + a_{1i}x_1 + a_{2i}x_2 + a_{3i}x_3 + c_{1i}h_1 + c_{2i}h_2 + c_{3i}h_3 + c_{4i}h_4 \\ i = 1, 2, 3 \\ h_1 = \eta\zeta, h_2 = \zeta\xi, h_3 = \xi\eta, h_4 = \xi\eta\zeta \end{cases} \quad (\text{D.10})$$

The x_i being the coordinates of a point of the element. In addition, the above equation evaluated at the nodes of the element gives the following three systems of eight equations:

$$\mathbf{U}_i^e = a_{0i}\mathbf{s} + a_{1i}\mathbf{X}_1^e + a_{2i}\mathbf{X}_2^e + a_{3i}\mathbf{X}_3^e + c_{1i}\mathbf{h}_1 + c_{2i}\mathbf{h}_2 + c_{3i}\mathbf{h}_3 + c_{4i}\mathbf{h}_4 \quad (\text{D.11})$$

The constants a_{ji} and $c_{\alpha i}$ of the equations (D.10) and (D.11) are obtained by introducing the so-called Hallquist vectors [HAL 83] and taking into account the orthogonality relations :

- Hallquist vectors

$$\mathbf{b}_i^T = \mathbf{N}_{,i}(\xi = \eta = \zeta = 0), \quad i = 1, 2, 3 \quad (\text{D.12})$$

- Orthogonality relationship

$$\begin{aligned} \mathbf{b}_i^T \cdot \mathbf{U}_i^e &= 0 & \mathbf{b}_i^T \cdot \mathbf{s} &= 0 & \mathbf{b}_i^T \cdot \mathbf{X}_j^e &= \delta_{ij} \\ \mathbf{h}_\alpha^T \cdot \mathbf{s} &= 0 & \mathbf{h}_\alpha^T \cdot \mathbf{h}_\beta &= 8\delta_{\alpha\beta} & \alpha, \beta &= 1, 2, 3, 4 \\ & & i, j &= 1, 2, 3 & & \end{aligned} \quad (\text{D.13})$$

Provided with the three Hallquist vectors and the orthogonality relationships, the constants a_{ji} et $c_{\alpha i}$ are calculated by multiplying equation (D.11) by \mathbf{b}_i^T and \mathbf{h}_α^T respectively.

$$\begin{cases} a_{ji} = \mathbf{b}_i^T \cdot \mathbf{U}_i^e \\ c_{\alpha i} = \gamma_\alpha^T \cdot \mathbf{U}_i^e \\ \gamma_\alpha = \frac{1}{8} \left[\mathbf{h}_\alpha - \sum_{j=1}^3 (\mathbf{h}_\alpha^T \cdot \mathbf{X}_j) \mathbf{b}_j \right] \end{cases} \quad (\text{D.14})$$

The displacement field can then be written in a very convenient form:

$$\mathbf{u}_i = a_{0i} + (\mathbf{b}_j^T x_j + \gamma_\alpha^T h_{\alpha}) \mathbf{U}_i^e \quad i = 1, 2, 3 \quad (\text{D.15})$$

Deriving this last formula with respect to the three spatial variables x_j we find the deformation gradient:

$$u_{i,j} = (\mathbf{b}_j^T + \gamma_\alpha^T \cdot h_{\alpha,j}) \cdot \mathbf{U}_i^e \quad i = 1, 2, 3 \quad (\text{D.16})$$

which then allows the construction of the discretized gradient operator linking the deformation fields to the nodal coordinates:

$$\begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{13} \\ 2\varepsilon_{23} \end{pmatrix} = \mathbf{B} \begin{pmatrix} \mathbf{U}_1^e \\ \mathbf{U}_2^e \\ \mathbf{U}_3^e \end{pmatrix} \quad (\text{D.17})$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_1^T + \gamma_\alpha^T h_{\alpha,x_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_2^T + \gamma_\alpha^T h_{\alpha,x_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{b}_3^T + \gamma_\alpha^T h_{\alpha,x_3} \\ \mathbf{b}_2^T + \gamma_\alpha^T h_{\alpha,x_2} & \mathbf{b}_1^T + \gamma_\alpha^T h_{\alpha,x_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_3^T + \gamma_\alpha^T h_{\alpha,x_3} & \mathbf{b}_2^T + \gamma_\alpha^T h_{\alpha,x_2} \\ \mathbf{b}_3^T + \gamma_\alpha^T h_{\alpha,x_3} & \mathbf{0} & \mathbf{b}_1^T + \gamma_\alpha^T h_{\alpha,x_1} \end{bmatrix} \quad (\text{D.18})$$

D.1.2.3 Assumed natural shear strain

The Assumed natural shear strain method consists in interpolating the shear deformations from the calculated deformation in the middle of each face in order to avoid the locking phenomenon. The points at the midpoints of the four faces transverse to the mid-plane of the element are called P_i $\{i = 1, 2, 3, 4\}$. At each of these points the corresponding transverse shear strain is calculated and then interpolated over the entire element. The transverse shear strain at the point P_i is given by the following equation:

$$\bar{\boldsymbol{\varepsilon}}^\perp(P_i) = \mathbf{J}_{2 \times 2}(P_i) \cdot \mathbf{B}^\perp(P_i) \mathbf{U}_e \quad (\text{D.19})$$

$$\mathbf{B}^\perp = \begin{bmatrix} \mathbf{0} & \mathbf{b}_3^T + \gamma_\alpha^T h_{\alpha,x_3} & \mathbf{b}_2^T + \gamma_\alpha^T h_{\alpha,x_2} \\ \mathbf{b}_3^T + \gamma_\alpha^T h_{\alpha,x_3} & \mathbf{0} & \mathbf{b}_1^T + \gamma_\alpha^T h_{\alpha,x_1} \end{bmatrix} \quad (\text{D.20})$$

$\mathbf{J}_{2 \times 2}$ being the Jacobian matrix evaluated at the point P_i , considering only the in-plane terms ($\zeta = 0$) and $\mathbf{B}^\perp(P_i)$ representing the fifth and sixth line of the deformation gradient matrix Eq:(D.20) corresponding to the transverse shear terms. \mathbf{U}_e is the element displacement vector. It is then possible to write the interpolated transverse shear strain in any point of the mid-plane. ($\zeta = 0$):

$$\bar{\boldsymbol{\varepsilon}}^\perp = \bar{\mathbf{J}}_{2 \times 2} \cdot \mathbf{M}_{int} \cdot \mathbf{C}_e \cdot \mathbf{U}_e \quad (\text{D.21})$$

With $\bar{\mathbf{J}}_{2 \times 2}$ the opposite of the Jacobian matrix taken at the origin of the frame ($\xi = 0, \eta = 0, \zeta = 0$) and \mathbf{M}_{int} the interpolation matrix such as:

$$\mathbf{M}_{int} = \frac{1}{2} \begin{bmatrix} 1 - \eta & 0 & 0 & 0 & 1 + \eta & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 + \xi & 0 & 0 & 0 & 1 - \xi \end{bmatrix} \quad (\text{D.22})$$

$$\mathbf{C}_e = \begin{bmatrix} \mathbf{J}_{2 \times 2}(P_1) \cdot \mathbf{B}^\perp(P_1) \\ \mathbf{J}_{2 \times 2}(P_2) \cdot \mathbf{B}^\perp(P_2) \\ \mathbf{J}_{2 \times 2}(P_3) \cdot \mathbf{B}^\perp(P_3) \\ \mathbf{J}_{2 \times 2}(P_4) \cdot \mathbf{B}^\perp(P_4) \end{bmatrix} \quad (\text{D.23})$$

Then the plane strain is simply multiplied by the Reissner weight function to get the actual strain calculated at any given point on the element:

$$\boldsymbol{\varepsilon}^\perp = \frac{5}{4} (1 - \zeta^2) \bar{\mathbf{J}}_{2 \times 2} \cdot \mathbf{M}_{int} \cdot \mathbf{C}_e \cdot \mathbf{U}_e \quad (\text{D.24})$$

Note that the only matrix that contains the space variables in this formula is the \mathbf{M}_{int} matrix. Moreover, with the use of the reduced integration method, this matrix loses terms that it needs to be stabilized. Indeed, we can decompose the matrix \mathbf{M}_{int} as follows:

$$\mathbf{M}_{int} = (\mathbf{M}_0 + \xi\mathbf{M}_\xi + \eta\mathbf{M}_\eta) \quad (\text{D.25})$$

with

$$\begin{cases} \mathbf{M}_0 = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \\ \mathbf{M}_\xi = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \\ \mathbf{M}_\eta = \frac{1}{2} \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{cases} \quad (\text{D.26})$$

Hence, equation (D.24) can also be broken down into three terms:

$$\varepsilon^\perp = \frac{5}{4} (1 - \zeta^2) \left[\mathbf{B}_0^\perp + \xi\mathbf{B}_\xi^\perp + \eta\mathbf{B}_\eta^\perp \right] \cdot \mathbf{U}_e \quad (\text{D.27})$$

with

$$\begin{cases} \mathbf{B}_0^\perp = \bar{\mathbf{J}}_{2 \times 2} \cdot \mathbf{M}_0 \cdot \mathbf{C}_e \\ \mathbf{B}_\xi^\perp = \bar{\mathbf{J}}_{2 \times 2} \cdot \mathbf{M}_\xi \cdot \mathbf{C}_e \\ \mathbf{B}_\eta^\perp = \bar{\mathbf{J}}_{2 \times 2} \cdot \mathbf{M}_\eta \cdot \mathbf{C}_e \end{cases} \quad (\text{D.28})$$

The first of these three matrices \mathbf{B}_0^\perp represents the gradient matrix for the calculation of non-stabilized stiffness. Multiplied by the Reissner function $\frac{5}{4} (1 - \zeta^2)$ this matrix of shear effects (unstabilized) is used for the calculation of \mathbf{K}_0 while the matrices \mathbf{B}_ξ^\perp and \mathbf{B}_η^\perp allow, on the other hand, to obtain the stiffness matrices for stabilization of the transverse shear below, calculated in the case of an isotropic linear material:

$$\begin{aligned} \mathbf{K}_\xi^\perp &= \frac{5}{6} \left[\mathbf{B}_\xi^\perp \right]^T \begin{bmatrix} \mu & 0 \\ 0 & \mu \end{bmatrix} \left[\mathbf{B}_\xi^\perp \right] \frac{V}{3} \\ \mathbf{K}_\eta^\perp &= \frac{5}{6} \left[\mathbf{B}_\eta^\perp \right]^T \begin{bmatrix} \mu & 0 \\ 0 & \mu \end{bmatrix} \left[\mathbf{B}_\eta^\perp \right] \frac{V}{3} \end{aligned} \quad (\text{D.29})$$

Identification of membrane and bending contribution.

Since transverse shear effects have been addressed, it may be useful to isolate membrane and bending effects to better identify them. The first, second, and fourth lines of the deformation gradient matrix can be decomposed to highlight the membrane and bending parts. The pinch effect will be discussed in the following paragraphs.

$$\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{13} \end{Bmatrix} = \left[\mathbf{B}_0^m + \zeta\mathbf{B}_0^b \right] \begin{Bmatrix} \mathbf{U}_1^e \\ \mathbf{U}_2^e \\ \mathbf{U}_3^e \end{Bmatrix} \quad (\text{D.30})$$

Avec \mathbf{B}_0^m la matrice des effets de membranes et \mathbf{B}_0^b celle des effets de flexion:

$$\mathbf{B}_0^m = \begin{bmatrix} \mathbf{b}_1^T & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_2^T \\ \mathbf{b}_2^T & \mathbf{b}_1^T \end{bmatrix} \quad (\text{D.31})$$

$$\mathbf{B}_0^b = \begin{bmatrix} \bar{\mathbf{J}}_{11}\gamma_2^T & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{J}}_{22}\gamma_1^T + \bar{\mathbf{J}}_{21}\gamma_2^T \\ \bar{\mathbf{J}}_{22}\gamma_1^T + \bar{\mathbf{J}}_{21}\gamma_2^T & \bar{\mathbf{J}}_{11}\gamma_2^T \end{bmatrix} \quad (\text{D.32})$$

D.1.2.4 Assumed normal displacement

In classical shell theory the transverse displacement is constant in the thickness direction, which means that the transverse strain is zero. Furthermore, this theory generally works under the plane stress hypothesis which states that the stress in the thickness direction is negligible. The novelty of the solid shell element SB9 γ 25 is to free itself from this restrictive hypothesis of plane stress. It is well understood that in some studies, it is essential to be able to evaluate the stresses in the thickness direction even if they are relatively small. Thus, in order to be able to take into account the pinch stress and at the same time to get rid of the plane stress hypothesis, an additional node is added to the center of the hexahedral element constructed so far. This node is endowed with a single degree of freedom of translation w_9 in the ζ direction of the element's local frame. With this node, it becomes possible to assume a parabolic transverse displacement w .

By defining:

$$\begin{aligned} w_9^- &= \frac{1}{4}(u_{31} + u_{32} + u_{33} + u_{34}) \\ w_9^+ &= \frac{1}{4}(u_{35} + u_{36} + u_{37} + u_{38}) \end{aligned} \quad (\text{D.33})$$

The assumed normal displacement ' w ' field is written:

$$w(\xi, \eta, z) = w(\xi, \eta, 0) + \left(\frac{2z^2}{t^2} - \frac{z}{t}\right)w_9^- - \left(\frac{2z^2}{t^2} + \frac{z}{t}\right)w_9^+ - \frac{2z^2}{t^2}w_9 \quad (\text{D.34})$$

Then, by deriving, we obtain the pinch component ϵ_{33} :

$$\epsilon_{33} = \frac{\partial w}{\partial x_3} = \frac{w_9^+ - w_9^-}{t} + 2\zeta \frac{w_9^+ - w_9^- - 2w_9}{t} \quad (\text{D.35})$$

t being the thickness of the element given by the ratio between the volume of the element and the surface area of the mid-plane ($\zeta = 0$). Note that $z = \frac{t}{2}\zeta$, it then comes :

$$\begin{aligned} \frac{w_9^+ - w_9^-}{t} &= [\mathbf{B}_c^p] \cdot \mathbf{U}_3^e \\ 2 \frac{w_9^+ - w_9^- - 2w_9}{t} &= [\mathbf{B}_\zeta^p] \cdot \mathbf{U}_3^e + \mathbf{B}_w^p w_9 \end{aligned} \quad (\text{D.36})$$

With $\begin{cases} \mathbf{B}_c^p = \frac{1}{4t} [-1 & -1 & -1 & -1 & 1 & 1 & 1 & 1] \\ \mathbf{B}_\zeta^p = \frac{1}{2t} [1 & 1 & 1 & 1 & 1 & 1 & 1 & 1] \\ \mathbf{B}_w^p = -\frac{4}{t} \end{cases}$ the pinch strain is written:

$$\boldsymbol{\varepsilon}_{33} = \left[\mathbf{B}_c^p + \zeta \mathbf{B}_\zeta^p \mid \zeta \mathbf{B}_w^p \right] \begin{Bmatrix} \mathbf{U}_3^e \\ w_9 \end{Bmatrix} \quad (\text{D.37})$$

D.1.2.5 Stiffness matrix

The elementary stiffness matrix \mathbf{K}_0 is calculated by combining the effects of membrane, bending, transverse shear and pinch Equation (D.38). Furthermore, the behaviour law no longer includes a plane stress hypothesis because, with the ninth node, the stress is no longer assumed to be zero in the thickness direction. The calculation of stiffness is done with a purely three-dimensional behaviour matrix, without restrictions.

$$\begin{cases} \begin{Bmatrix} \boldsymbol{\varepsilon}_{11} \\ \boldsymbol{\varepsilon}_{22} \\ 2\boldsymbol{\varepsilon}_{12} \end{Bmatrix} = [\mathbf{B}_0^m + \zeta \mathbf{B}_0^b] \begin{Bmatrix} \mathbf{U}_1^e \\ \mathbf{U}_2^e \\ \mathbf{U}_3^e \end{Bmatrix} & \text{Membrane and bending} \\ \begin{Bmatrix} 2\boldsymbol{\varepsilon}_{13} \\ 2\boldsymbol{\varepsilon}_{23} \end{Bmatrix} = \frac{5}{4} (1 - \zeta^2) \mathbf{B}_0^\perp \begin{Bmatrix} \mathbf{U}_1^e \\ \mathbf{U}_2^e \\ \mathbf{U}_3^e \end{Bmatrix} & \text{Transverse shear} \\ \boldsymbol{\varepsilon}_{33} = \left[\mathbf{B}_c^p + \zeta \mathbf{B}_\zeta^p \mid \zeta \mathbf{B}_w^p \right] \begin{Bmatrix} \mathbf{U}_3^e \\ w_9 \end{Bmatrix} & \text{Pinching} \end{cases} \quad (\text{D.38})$$

$$\mathbf{K}_0 = \int_{\Omega} \mathbf{B}^T \cdot \mathbf{D} \cdot \mathbf{B} dV = 4 \sum_{i=1}^n \omega(\zeta_q) \cdot \mathbf{B}^T(\zeta_q) \cdot \mathbf{D} \cdot \mathbf{B}(\zeta_q) \cdot \det \mathbf{J}(0, 0, \zeta_q) \quad (\text{D.39})$$

\mathbf{B} being the gradient matrix, \mathbf{D} the 3D constitutive law, which is given in linear elasticity form by the equations D.40, n is the number of integration points in the thickness (at least 5 for elastic-plastic behaviour), $\omega(\zeta_q)$ the Gauss-lobatto weight associated to ζ_q along the thickness direction. The factor 4 compensates the fact that $\det \mathbf{J}(0, 0, \zeta_q)$ is evaluated in the position $\xi = \eta = 0$.

$$\mathbf{D} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \quad (\text{D.40})$$

The goal of reduced integration is to improve the performance of the element in terms of computing time and to eliminate shear and membrane locking.

D.1.2.6 Geometric rigidity matrix

σ being the stress tensor, we can define two matrices Σ and \mathbf{B}_G so that the geometric rigidity matrix is written as follows:

$$\mathbf{K}_\sigma = \int_{\Omega} \mathbf{B}_G^T \cdot \Sigma \cdot \mathbf{B}_G dV = 4 \sum_{i=1}^n \omega(\zeta_q) \cdot \mathbf{B}_G^T(\zeta_q) \cdot \Sigma \cdot \mathbf{B}_G(\zeta_q) \cdot \det \mathbf{J}(0, 0, \zeta_q) \quad (\text{D.41})$$

with

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \quad (\text{D.42})$$

$$\Sigma = \begin{bmatrix} [\sigma] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & [\sigma] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & [\sigma] \end{bmatrix} \quad (\text{D.43})$$

$$\mathbf{B}_G = \begin{bmatrix} \mathbf{b}_1^T + \gamma_\alpha^T h_{\alpha,x_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{b}_2^T + \gamma_\alpha^T h_{\alpha,x_2} & \mathbf{0} & \mathbf{0} \\ \mathbf{b}_3^T + \gamma_\alpha^T h_{\alpha,x_3} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_1^T + \gamma_\alpha^T h_{\alpha,x_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_2^T + \gamma_\alpha^T h_{\alpha,x_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_3^T + \gamma_\alpha^T h_{\alpha,x_3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{b}_1^T + \gamma_\alpha^T h_{\alpha,x_1} \\ \mathbf{0} & \mathbf{0} & \mathbf{b}_2^T + \gamma_\alpha^T h_{\alpha,x_2} \\ \mathbf{0} & \mathbf{0} & \mathbf{b}_3^T + \gamma_\alpha^T h_{\alpha,x_3} \end{bmatrix} \quad (\text{D.44})$$

D.1.2.7 Pressure matrix

Let's consider a p pressure, applied to a surface ∂V of normal \mathbf{n} . By neglecting the effects of gravity, the virtual work of the external forces for the finite element SB9γ25 is written :

$$\delta W_{ext} = - \int_{\Omega_0} \tilde{S}_{33} \delta \tilde{E}_{33} d\Omega_0 - \int_{\partial V_0} p J \mathbf{n}_0 \mathbf{F}^{-T} \cdot \delta \mathbf{u} d(\partial \Omega_0) \quad (\text{D.45})$$

Where \tilde{S}_{33} is the interpolated (assumed) stress written in a Piola Kirchhoff 2 form, \tilde{E}_{33} the assumed normal strain, \mathbf{n}_0 the normal vector in the reference configuration, \mathbf{F} the gradient of the transformation and $J = \det \mathbf{F}$. The assumption that has been made on the normal stress is that the pinch is linear over the thickness of the plate but statically admissible. For example if p is applied on the lower edge of the element and assuming that the upper edge is free, we have $\sigma_{33}^p = -\frac{1}{2}(1 - \zeta)p$. The virtual work of the pressure loads becomes linearized in this way:

$$\delta W_{ext}(\Delta \mathbf{u}) = -\delta \mathbf{U}_z^T \left(\int_{\Omega_0} \mathbf{N}^T \tilde{S}_{33} \mathbf{N} d\Omega \right) \Delta \mathbf{U}_z - L \left[\int_{\partial \Omega_0} p J \mathbf{n}_0 \mathbf{F}^{-T} \cdot \delta \mathbf{u} d(\partial \Omega) \right] \quad (\text{D.46})$$

$L[f]$ representing the linearized function f . The second term on the right side of the equation is the formula of the classical displacement-dependent pressure, while the first

term is the effect of the pinching assumed with the ninth node. This additional term is used to eliminate Poisson thickness locking in solid-shell finite elements.

D.1.3 Stabilization

The Reduced integration method of the SB9γ25 element (5 Gauss points only) aims to considerably reduce the computation time, but also to remove the various locking phenomenon encountered in the numerical implementation of finite elements. However, this reduced-integration has not only advantages: it unfortunately introduces parasitic modes associated with zero energy (hourglass mode). This deficiency of the stiffness matrix, due to the reduced-integration, must therefore be compensated by adding a stabilizing stiffness to the elementary stiffness. The kernel of the new stiffness, thus obtained, must be reduced to the only modes corresponding to rigid body movements.

D.1.3.1 Lost terms

At the 5 Gauss-lobatto coordinate points ($\xi_I = \eta_I = 0, \zeta_I \neq 0$), some terms from h_{α,x_i} ($\alpha = 3, 4; i = 1, 2, 3$) and h_{α,x_i} ($\alpha = 1, 2; i = 1, 2, 3$) becomes null. By elaborating the derivatives h_{α,x_i} , terms that do not enter into the construction of the stiffness matrix are clearly identified:

$$h_1 = \eta\zeta \Rightarrow \begin{Bmatrix} h_{1,x_1} \\ h_{1,x_2} \\ h_{1,x_3} \end{Bmatrix} = \zeta \begin{Bmatrix} \bar{J}_{12} \\ \bar{J}_{22} \\ \bar{J}_{32} \end{Bmatrix} + \eta \begin{Bmatrix} \bar{J}_{13} \\ \bar{J}_{23} \\ \bar{J}_{33} \end{Bmatrix} \quad (D.47)$$

$$h_2 = \xi\zeta \Rightarrow \begin{Bmatrix} h_{2,x_1} \\ h_{2,x_2} \\ h_{2,x_3} \end{Bmatrix} = \zeta \begin{Bmatrix} \bar{J}_{11} \\ \bar{J}_{21} \\ \bar{J}_{31} \end{Bmatrix} + \xi \begin{Bmatrix} \bar{J}_{13} \\ \bar{J}_{23} \\ \bar{J}_{33} \end{Bmatrix} \quad (D.48)$$

$$h_3 = \xi\eta \Rightarrow \begin{Bmatrix} h_{3,x_1} \\ h_{3,x_2} \\ h_{3,x_3} \end{Bmatrix} = \eta \begin{Bmatrix} \bar{J}_{11} \\ \bar{J}_{21} \\ \bar{J}_{31} \end{Bmatrix} + \xi \begin{Bmatrix} \bar{J}_{12} \\ \bar{J}_{22} \\ \bar{J}_{32} \end{Bmatrix} \quad (D.49)$$

$$h_4 = \xi\eta\zeta \Rightarrow \begin{Bmatrix} h_{4,x_1} \\ h_{4,x_2} \\ h_{4,x_3} \end{Bmatrix} = \zeta \begin{Bmatrix} \eta\bar{J}_{11} + \xi\bar{J}_{12} \\ \eta\bar{J}_{21} + \xi\bar{J}_{22} \\ \eta\bar{J}_{31} + \xi\bar{J}_{32} \end{Bmatrix} + \xi\eta \begin{Bmatrix} \bar{J}_{13} \\ \bar{J}_{23} \\ \bar{J}_{33} \end{Bmatrix} \quad (D.50)$$

Considering equations D.47 to D.50 it is easy to understand that the deformation gradient matrix is deficient in the five Gauss points of the reduced integration method.

By constructing a discretized displacement base, it is possible to show that the reduced integration reduces the rank of the discretized stiffness. In fact, with a complete (exact) integration we find six vectors in the kernel of the stiffness that correspond to the three translations and three rotations of rigid bodies. In addition to these six vectors, the reduced integration adds the six vectors [equation (D.51)] but also other vectors related to the elimination of some terms of the modes h_1 and h_2 .

$$\begin{pmatrix} \mathbf{h}_3 \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ \mathbf{h}_3 \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{h}_3 \end{pmatrix}, \begin{pmatrix} \mathbf{h}_4 \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ \mathbf{h}_4 \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{h}_4 \end{pmatrix} \quad (\text{D.51})$$

This expansion of the kernel of the stiffness matrix is manifested by the apparition of unrealistic deformations called *hourglass* modes that need to be stabilized.

D.1.3.2 Choosing the local stabilization reference frame

The SB9 γ 25 is a finite element built to simulate the behavior of thin structures, i.e. structures that have a dimension that is smaller than the two others. So to always identify this direction, the local frame is constructed so that its axis z always coincide with the thickness of the element ζ . Furthermore, the connectivity of the element is, at all times, written in the order 1, 2, 3, 4 (bottom side) and 5, 6, 7, 8 (top side) to take into account the thickness direction.

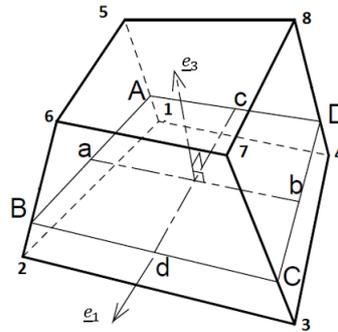


Figure D.2: Local frame

To define the local reference frame, it is more judicious to place oneself in a co-rotational reference frame, rotating and aligned with the reference frame [BEL 93]. This choice is justified by the rotation extracted from the decomposition of the gradient of the transformation [ABE 02]. This co-rotational formulation has advantages such as the simplification of the Jacobian of the element [BAS 12] but also of the formulas giving the stabilization matrix. In addition to facilitating the efficient processing of shear locking, this frame is best suited to geometric non-linearities. The construction of this local frame

(in $\xi = \eta = \zeta = 0$) starts with the definition of two vectors \mathbf{g}_1 and \mathbf{g}_2 such as:

$$\begin{aligned} g_{1i} &= \mathbf{a}_1 \cdot \mathbf{X}_i^G \\ g_{2i} &= \mathbf{a}_2 \cdot \mathbf{X}_i^G \\ i &= 1, 2, 3 \end{aligned} \quad (\text{D.52})$$

\mathbf{a}_1 and \mathbf{a}_2 are the vectors defined in the equations (D.6) and the \mathbf{X}_i^G are the coordinates, in the global frame, of the 8 nodes of the element. These two vectors must then be orthonormalized before calculating the co-rotational base vectors. The correction term \mathbf{g}_c is computed to ensure the orthogonality relationship: $\mathbf{g}_1 \cdot (\mathbf{g}_2 + \mathbf{g}_c) = 0$, meaning

$$\mathbf{g}_c = -\frac{\mathbf{g}_1 \cdot \mathbf{g}_2}{\mathbf{g}_1 \cdot \mathbf{g}_1} \mathbf{g}_1 \quad (\text{D.53})$$

Finally, the vectors of the co-rotational frame are written as follows:

$$\begin{aligned} e_1 &= \frac{\mathbf{g}_1}{\|\mathbf{g}_1\|} \\ e_2 &= \frac{\mathbf{g}_2 + \mathbf{g}_c}{\|\mathbf{g}_2 + \mathbf{g}_c\|} \\ e_3 &= e_1 \wedge e_2 \end{aligned} \quad (\text{D.54})$$

In the following we will call \mathbf{X}_i^e and \mathbf{U}_i^e respectively the coordinates and nodal displacements in the local coordinate system. If we call \mathbf{J} the Jacobian matrix such as:

$$\mathbf{J}_{ij} = \frac{\partial x_i}{\partial \xi_j}, \quad (\text{D.55})$$

the co-rotational frame allows the following simplifications of the terms of the Jacobian matrix:

$$\left\{ \begin{aligned} \frac{\partial x_i}{\partial \xi_i} &= \frac{1}{8} \mathbf{a}_i \cdot \mathbf{X}_j^e \\ \frac{\partial x_i}{\partial \xi_j} &= \frac{\partial \xi_j}{\partial x_i} = 0 \quad i \neq j \\ \det \mathbf{J} &= \frac{\mathbf{a}_1^T \cdot \mathbf{X}_1^e}{8} \frac{\mathbf{a}_2^T \cdot \mathbf{X}_2^e}{8} \frac{\mathbf{a}_3^T \cdot \mathbf{X}_2^e}{8} \end{aligned} \right. \quad (\text{D.56})$$

D.1.3.3 Stabilization of modes h_1 and h_2

As already shown the reduced integration cancels some terms of the h_1 and h_2 modes. These combined modes would give a kind of false torsional deformation whose impact can be neglected depending on the modeling. This is why these modes are not stabilized in the initial formulation of the SHB8PS element which was originally intended for the simulation of the forming of thin-thickness structures [BAS 12, ABE 01].

By combining the terms h_1 and h_2 removed during reduced integration, we can write as follows the deformation gradient matrix used to stabilize these modes.

$$\mathbf{B}_{12}^s = \begin{bmatrix} \bar{J}_{13} (\eta\gamma_1^T + \xi\gamma_2^T) & | & \mathbf{0} & | & \mathbf{0} \\ \mathbf{0} & | & \bar{J}_{23} (\eta\gamma_1^T + \xi\gamma_2^T) & | & \mathbf{0} \\ \mathbf{0} & | & \mathbf{0} & | & \bar{J}_{33} (\eta\gamma_1^T + \xi\gamma_2^T) \\ \bar{J}_{23} (\eta\gamma_1^T + \xi\gamma_2^T) & | & \bar{J}_{13} (\eta\gamma_1^T + \xi\gamma_2^T) & | & \mathbf{0} \\ \bar{J}_{33} (\eta\gamma_1^T + \xi\gamma_2^T) & | & 0 & | & \bar{J}_{13} (\eta\gamma_1^T + \xi\gamma_2^T) \\ 0 & | & \bar{J}_{33} (\eta\gamma_1^T + \xi\gamma_2^T) & | & \bar{J}_{23} (\eta\gamma_1^T + \xi\gamma_2^T) \end{bmatrix} \quad (\text{D.57})$$

It is then possible to calculate the stabilization stiffness matrix of the modes h_1 and h_2 :

$$\mathbf{K}_{12}^{Stab} = \int_{\Omega} (\mathbf{B}_{12}^s)^T \cdot \mathbf{D} \cdot \mathbf{B}_{12}^s d\Omega \quad (\text{D.58})$$

$$\mathbf{K}_{12}^{Stab} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{33} \end{bmatrix} \begin{cases} \mathbf{K}_{11} = \frac{1}{3}V\mu (\gamma_1\gamma_1^T + \gamma_2\gamma_2^T) \\ \mathbf{K}_{22} = \frac{1}{3}V\mu (\gamma_1\gamma_1^T + \gamma_2\gamma_2^T) \\ \mathbf{K}_{33} = \frac{1}{3}V(\lambda + 2\mu) (\gamma_1\gamma_1^T + \gamma_2\gamma_2^T) \end{cases} \quad (\text{D.59})$$

\mathbf{K}_{11} , \mathbf{K}_{22} and \mathbf{K}_{33} being 8×8 matrices.

D.1.3.4 Stabilization of modes h_3 et h_4

The stabilization matrix of the modes h_3 and h_4 is calculated in the co-rotational frame which gives the Jacobian \mathbf{J} the properties of a diagonal matrix. In addition, the simplification of the Jacobian leads to the following simplifications:

$$\begin{cases} h_{i,i} = \frac{\partial h_i}{\partial x_i} = 0 \\ h_{j,i} = \frac{8\xi_k}{\mathbf{a}_i \cdot \mathbf{X}_i^e} \\ h_{4,i} = \frac{8\xi_j\xi_k}{\mathbf{a}_i \cdot \mathbf{X}_i^e} \end{cases} \quad (\text{D.60})$$

$$\mathbf{B}_{34}^s = \begin{bmatrix} \gamma_3^T (\eta\bar{J}_{11} + \xi\bar{J}_{12}) + \bar{J}_{13}\gamma_4^T\xi\eta & | & \mathbf{0} & | & \mathbf{0} \\ \mathbf{0} & | & \gamma_3^T (\eta\bar{J}_{21} + \xi\bar{J}_{22}) + \bar{J}_{23}\gamma_4^T\xi\eta & | & \mathbf{0} \\ \mathbf{0} & | & \mathbf{0} & | & \gamma_3^T (\eta\bar{J}_{31} + \xi\bar{J}_{32}) + \bar{J}_{33}\gamma_4^T\xi\eta \\ \gamma_3^T (\eta\bar{J}_{21} + \xi\bar{J}_{22}) + \bar{J}_{23}\gamma_4^T\xi\eta & | & \gamma_3^T (\eta\bar{J}_{11} + \xi\bar{J}_{12}) + \bar{J}_{13}\gamma_4^T\xi\eta & | & \mathbf{0} \\ \gamma_3^T (\eta\bar{J}_{31} + \xi\bar{J}_{32}) + \bar{J}_{33}\gamma_4^T\xi\eta & | & 0 & | & \gamma_3^T (\eta\bar{J}_{11} + \xi\bar{J}_{12}) + \bar{J}_{13}\gamma_4^T\xi\eta \\ 0 & | & \gamma_3^T (\eta\bar{J}_{31} + \xi\bar{J}_{32}) + \bar{J}_{33}\gamma_4^T\xi\eta & | & \gamma_3^T (\eta\bar{J}_{21} + \xi\bar{J}_{22}) + \bar{J}_{23}\gamma_4^T\xi\eta \end{bmatrix} \quad (\text{D.61})$$

In order to simplify as much as possible the calculation of the stabilization stiffness matrix but also to deal more efficiently with transverse shear locking, the matrix \mathbf{B}_{34}^s is replaced by the simplified matrix Eq:(D.62). This simplified form is justified by the properties of the co-rotational frame which simplify the Jacobian ($\frac{\partial x_i}{\partial \xi_j} = 0$, si $i \neq j$).

$$\mathbf{B}_{34}^s = \begin{bmatrix} \eta \bar{J}_{11} (\gamma_3^T + \zeta \gamma_4^T) & | & \mathbf{0} & | & \mathbf{0} \\ \mathbf{0} & | & \xi \bar{J}_{22} (\gamma_3^T + \zeta \gamma_4^T) & | & \mathbf{0} \\ \mathbf{0} & | & \mathbf{0} & | & \mathbf{0} \\ \mathbf{0} & | & \mathbf{0} & | & \mathbf{0} \\ \mathbf{0} & | & \mathbf{0} & | & \eta \bar{J}_{11} (c \gamma_3^T + \zeta \gamma_4^T) \\ \mathbf{0} & | & \mathbf{0} & | & \xi \bar{J}_{22} (c \gamma_3^T + \zeta \gamma_4^T) \end{bmatrix} \quad (\text{D.62})$$

Thanks to the behaviour matrix we can calculate the stabilization stiffness matrix of the modes h_3 and h_4 from the formula:

$$\mathbf{K}_{34}^{Stab} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{33} \end{bmatrix} \begin{cases} \mathbf{K}_{11} = (\lambda + 2\mu) H_{11} [\gamma_3 \gamma_3^T + \frac{1}{3} \gamma_4 \gamma_4^T] \\ \mathbf{K}_{22} = (\lambda + 2\mu) H_{22} [\gamma_3 \gamma_3^T + \frac{1}{3} \gamma_4 \gamma_4^T] \\ \mathbf{K}_{33} = \mu (H_{22} + H_{22}) [c^2 \gamma_3 \gamma_3^T + \frac{1}{3} \gamma_4 \gamma_4^T] \end{cases} \quad (\text{D.63})$$

$$\text{with} \begin{cases} H_{ii} = \frac{1}{3} \frac{(\mathbf{a}_j^T \cdot \mathbf{X}_j^e) (\mathbf{a}_k^T \cdot \mathbf{X}_k^e)}{\mathbf{a}_i^T \cdot \mathbf{X}_i^e} \\ H_{ij} = \int_V h_{i,j} h_{j,i} dV \\ 0 = \int_V h_{i,j} dV \end{cases} \quad (\text{D.64})$$

The simplifications made so far with the co-rotational frame work if the element has a quite regular geometry with a quasi-constant thickness. In [ABE 09], the authors suggest to take the factor $c = 0.1$ in order to correctly stabilize the h_3 mode along the thickness if the thickness is not constant.

D.1.3.5 Consequence of assumed shear strain on stabilization

The Assumed strain method and its corresponding stabilization correct the deficiencies of the h_3 mode. In the same way, this stabilization also ensures the stabilization of the missing parts of the modes h_1 and h_2 , a combination of which corresponds to a twist mode around ζ . As a result, the stabilization gradient matrix can be rewritten in a more simplified way:

$$\mathbf{B}_S = \begin{bmatrix} \bar{\mathbf{J}}_{11} \eta (\gamma_3^T + \zeta \gamma_4^T) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{J}}_{22} \xi (\gamma_3^T + \zeta \gamma_4^T) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \bar{\mathbf{J}}_{33} (\eta \gamma_1^T + \xi \gamma_2^T + \xi \eta \gamma_4^T) \end{bmatrix} \quad (\text{D.65})$$

With the plane stress behaviour matrix, the stabilization stiffness matrix is rewritten as follows:

$$\mathbf{K}_S = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{33} \end{bmatrix} \begin{cases} \mathbf{K}_{11} = (\lambda + 2\mu) H_{11} \left[f_m \gamma_3 \gamma_3^T + \frac{1}{3} f_b \gamma_4 \gamma_4^T \right] \\ \mathbf{K}_{22} = (\lambda + 2\mu) H_{22} \left[f_m \gamma_3 \gamma_3^T + \frac{1}{3} f_b \gamma_4 \gamma_4^T \right] \\ \mathbf{K}_{33} = (\lambda + 2\mu) f_p H_{33} \left[\gamma_1 \gamma_1^T + \gamma_3 \gamma_3^T + \frac{1}{3} \gamma_4 \gamma_4^T \right] \end{cases} \quad (\text{D.66})$$

With $H_{ii} = \bar{\mathbf{J}}_{ii}^2 \frac{V}{3}$ in the co-rotational frame. λ and μ Lamé's constants. The coefficients f_m , f_b and f_p are three stabilization coefficients related to membrane, bending and pinch effects. If these coefficients are taken equal to 1, the results are close to those obtained with a fully integrated classical hexahedral element ($2 \times 2 \times 2$). In other words, locking phenomena appear. For the element to work well in solid-shell, these coefficients must be taken small, 10^{-4} for example. Recall that the matrix, \mathbf{K}_S , defined above must be added to the stabilization matrices of the transverse shears defined in the equation D.29

D.1.4 Equivalent nodal pressure forces

The advantage of the additional node is also to allow a redistribution of the pressure forces so that the normal pinch stress is more correctly distributed. In order to easily find the distribution of the latter, we consider a hexahedral element on which is applied a pressure P_i , which for a classical shell element would be directly applied on the mid-plane [figure D.3]. For an elastic material, it can be assumed that the pinching stress is zero on the free side and is the opposite of the pressure on the side where the pressure is applied. Moreover, in the elastic case, since the transverse displacement is assumed to be parabolic, it immediately follows that the pinching stress is linear; hence the following interpolation of σ_{zz} , the pressure being applied to the bottom face of the element:

$$\sigma_{33}^p = \frac{1}{2} (\zeta - 1) P_i \quad (\text{D.67})$$

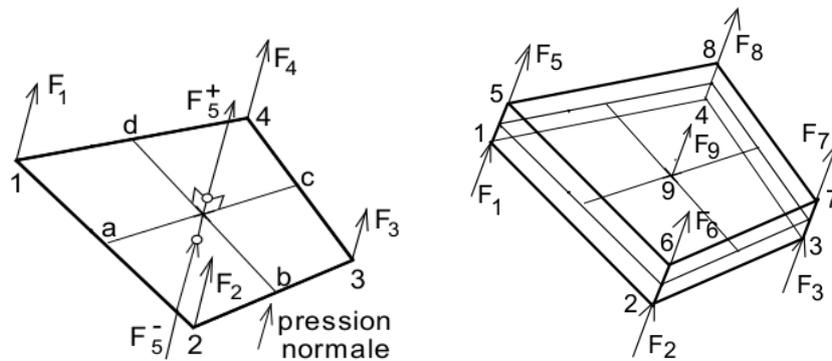


Figure D.3: Reference geometry of the element and integration points

By noting F_i the contribution to node i of the ζ component of the generalized nodal effort and \hat{w} the virtual displacement, the equilibrium of the virtual works gives the generalized nodal efforts:

$$\sum_{i=1}^8 F_i \hat{w}_i + F_9 \hat{w}_9 = \int_{\partial\Omega} P_i \hat{w}_i d(\partial\Omega) + \int_{\Omega} \hat{\epsilon}_{33} \cdot \sigma_{33}^p d\Omega \quad (\text{D.68})$$

$$\begin{cases} F_i = \frac{1}{3} P_i A_0, & i = 1, 2, 3, 4 \\ F_j = \frac{1}{12} P_i A_0, & j = 5, 6, 7, 8 \\ F_9 = -\frac{2}{3} P_i A_0 \end{cases} \quad (\text{D.69})$$

where A_0 is the area of the mid plane of the element. We notice in these formulas that the pressure applied on the lower face of the element is partly absorbed by the nodes of the upper face. For linear calculations this distribution can work very well [BAS 12] but in a perspective of carrying out non-linear calculations such as contact, this distribution of pressure forces can be a real problem. To remedy this drawback the authors in [BAS 12] recommend to consider the relative displacement of the ninth node w_9^* rather than the absolute displacement w_9 . The relationship between the two being:

$$w_9 = \frac{1}{2} (w_9^+ + w_9^-) + w_9^* \quad (\text{D.70})$$

Remember that w_9^+ and w_9^- are respectively the averages of the transverse displacements of the nodes of the upper and lower face. The use of the relative displacement for the 9th node then leads to a new writing of the transverse strain, and then of the generalized nodal forces:

$$\hat{\epsilon}_{33} = \frac{\hat{w}_9^+ - \hat{w}_9^-}{h} - 4\zeta \frac{\hat{w}_9^*}{h} \quad (\text{D.71})$$

$$\sum_{i=1}^8 F_i \hat{w}_i + F_9 \hat{w}_9 = \int_{\partial\Omega} P_i \hat{w}_i d(\partial\Omega) + \int_{\Omega} \hat{\epsilon}_{33} \cdot \sigma_{33}^p d\Omega \quad (\text{D.72})$$

$$\begin{cases} F_i = \frac{1}{4} P_i A_0, & i = 1, 2, 3, 4 \\ F_j = 0, & j = 5, 6, 7, 8 \\ F_9^* = \frac{2}{3} P_i A_0 \end{cases} \quad (\text{D.73})$$

This new distribution is closer to what is usually encountered; only the nodes on the pressure side take up efforts. It may seem strange at first glance that equilibrium is not reached in this configuration ($4F_i + 4F_j + F_9^* \neq P_i A_0$), as is the case with the first distribution Eq:(D.69) but remember that the displacement w_9^* is not a real (absolute) displacement. For a SB9γ25 element loaded in pressure P_s and P_i respectively on their

upper and lower faces, the generalized nodal forces can be obtained by superposition:

$$\sum_{i=1}^8 F_i \hat{w}_i + F_9 \hat{w}_9 = \int_{\partial\Omega} (P_i - P_s) \hat{w}_i d(\partial\Omega) + \int_{\Omega} \varepsilon_{33}^{\hat{}} \cdot \sigma_{33}^p d\Omega \quad (\text{D.74})$$

$$\begin{aligned} F_i &= +\frac{1}{4} P_i A_0, & i &= 1, 2, 3, 4 \\ F_j &= -\frac{1}{4} P_s A_0, & j &= 5, 6, 7, 8 \\ F_9^* &= \frac{2}{3} (P_s - P_i) A_0 \end{aligned} \quad (\text{D.75})$$

Note that these efforts are given in the element's local frame. It will therefore be necessary to put them back into the global frame before proceeding with the assembly. This is also valid for the elementary stiffness and stabilization matrices developed in this chapter. At this stage we have all the necessary elements for the implementation of this element. Before going into the details of the implementation, we will show, in the following section, the limitations encountered in the formulation of this element.

D.2 Numerical validations SB9γ25

D.2.1 Study of a bending problem : SB9γ25 vs HEXA20

The SB9γ25 element is a solid-shell element with quadratic interpolation of the transverse displacement. It is then natural to ask what interest can the finite element SB9γ25 have compared to the quadratic finite element HEXA20? The purpose of this test is to provide an answer to such a question. We consider a square plate of length **1**, clamped on one edge and subjected to a bending load applied on the opposite edge. Two studies were carried out: the first with a thickness of **0.1** and the second with a thickness of **0.01**. In both cases, a null Poisson ratio coefficient is considered in order to be able to compare the solutions to the analytical solution of a beam in bending. We also ran this test on using the two shell finite elements available in **code_aster**: DKT and Coque_3D. The mesh size in each model is : $5 \times 5 \times 1$

Length	L	1m
Width	l	1m
Thickness	t	0.1 then 0.01m
Young Modulus	E	$21 \times 10^9 \text{Pa}$
Poisson ratio	ν	0.0
Applied Load	Fz	100m

Table D.1: Geometrical and material characteristics of the plate

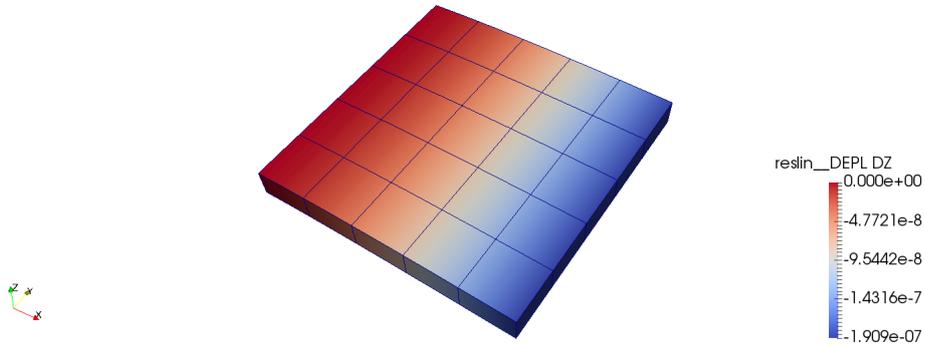


Figure D.4: Bending plate: thickness=0.1,HEXA20

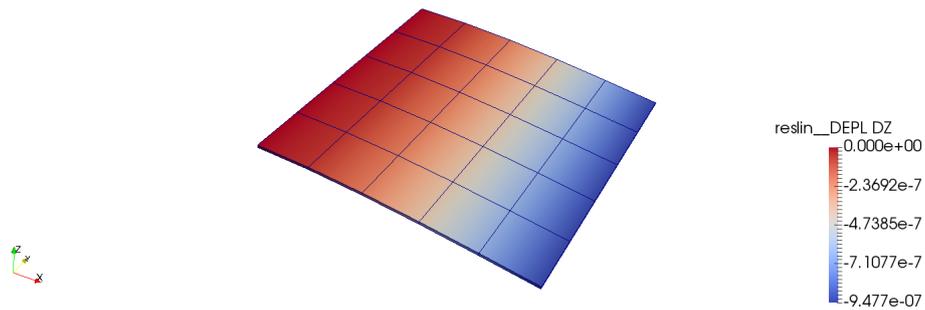


Figure D.5: Bending plate: thickness=0.01,HEXA20

	Hexa20	DKT	Coque 3D	SB9g25
$U_z(10^{-6}\text{m})$	1.90	1.90	1.91	1.89

Table D.2: Normal displacement: plate of thickness 0.1

the plate thickness is reduced to 0.01

	Hexa20	DKT	Coque 3D	SB9g25
$U_z(10^{-3}\text{m})$	0.0094	1.90	1.90	1.89

Table D.3: Normal displacement: plate of thickness 0.01

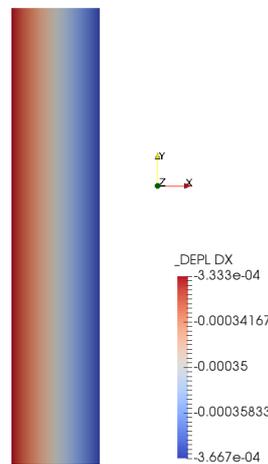
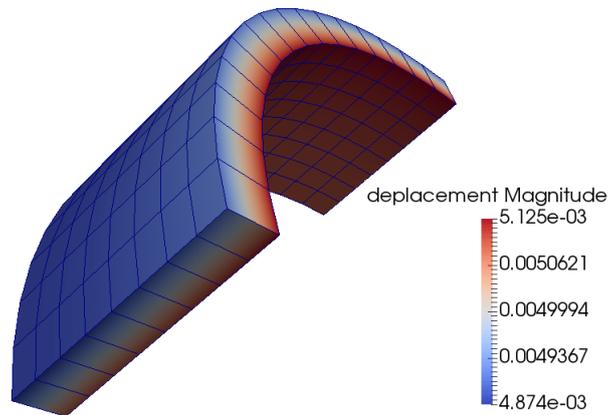
It can be seen that for a fairly large thickness the finite elements HEXA20 and SB9 give almost the same results; HEXA20 being slightly more precise than SB9γ25. On the other hand, for a thickness ten times smaller, HEXA20 remains far less accurate than SB9γ25. Indeed when the thickness is small, in a real plate configuration, the HEXA20 element loses its flexibility and shows a severe locking problem. The SB9γ25 element, on the other hand, seems to fit both cases. When the plate is thick, because of its solid character, SB9γ25 behaves quite well. When the plate becomes thinner in thickness, it also gives good results due to the many techniques for eliminating locking phenomena: reduced integration, assumed transverse shear fields. SB9γ25 is comparable to the HEXA20 solid finite element in terms of robustness. But in terms of the quality of the simulation of thin structure behavior, the SB9γ25 is far better than the HEXA20 element.

D.2.2 Study of a pinch problem : SB9γ25 vs Axis

In this test case a pressurized cylinder is studied in order to see the efficiency of the quadratic interpolation of the transverse displacement of the SB9γ25 element. A moderately thick cylinder is considered to observe the effect of thickness on pinch stress accuracy. We then consider a cylinder of radius $R = 1$ and thickness $e = 0.1$ subjected to an internal pressure $P_i = 200MPa$ and an external pressure $P_e = 100MPa$. The material parameters being Young's modulus $E = 210GPa$ and Poisson ratio $\nu = 0.3$. Two calculations have been performed: the first one with SB9γ25 elements and the second one being a modeling with quadratic Axi-symmetric elements. For symmetry reasons, only one eighth of the cylinder is meshed with SB9γ25 elements ($8 \times 8 \times 1$).

Radius	R	1m
thickness	t	0.1m
Young Modulus	E	$210 \times 10^9 Pa$
Poisson ratio	ν	0.3
Internal Pressure	P_i	$2 \times 10^8 Pa$
external Pressure	P_e	$1 \times 10^8 Pa$

Table D.4: Geometrical and material characteristics of the cylinder



	Axi (1 Elem)	Axi (5 Elem)	SB9	ref
$\sigma_{rr}^{int} (10^8 \text{Pa})$	1.97	1.99	2.08	2
$\sigma_{rr}^{ext} (10^8 \text{Pa})$	0.97	0.99	0.93	1

Table D.5: Observation of pinch stresses in the inner and outer walls of the cylinder

The table D.5 gives three results: the result of a calculation with a single quadratic axis element in the thickness of the plate, a calculation with a refinement (5 quadratic axis elements in the thickness) then the calculation with the SB9γ25 element. For a curved structure with a rather large thickness the SB9γ25 element gives less good results compared to 3D quadratic elements. This is not surprising. The SB9γ25 element is a solid-shell element, it can give fairly good results when the structures are medium thick but when you get closer to 3D structures its performance decreases. This element is not made to be used for 3D structure calculations. Moreover, due to facetization, the mapping effects intervene in the results, which is less the case with quadratic 3D elements, which

more or less match the cylindrical geometry. To improve this result, a refinement in the circumferential and radial directions would have been necessary. However, in its current formulation, SB9 γ 25 is not immediately discretizable in the thickness direction. The designers of this element were based on the idea that SB9 γ 25 is a finite element that can be used to model thin structures. Therefore they assumed that only one element was needed to discretize the thickness of the plates.

D.3 Conclusion

The purpose of this chapter was to present and evaluate the SB9 γ 25 element through a few test cases. The results given by this element are globally satisfactory. The assumed natural method and the reduced integration technique give this solid element a good plate behaviour. Thus, SB9 γ 25 gives very good results in the bending tests, without any sign of apparent transverse shear locking. This is not the case with the HEXA20 element, which suffers severely from locking phenomena when the thickness is small. Moreover, unlike many solid-shell elements, the SB9 finite element gives a normal stress. On the one hand, if we compare with the classic plate and shell elements DKT and COQUE_3D, we can see that the results provided by the SB9 γ 25 is in agreement with the kinematics specific to thin structures : this is one of the initial objectives which is satisfied with this element. On the other hand, comparing (with axi-symmetric) the pinching quality obtained by the SB9 γ 25, we find that the results are in agreement with the reference solution on a test case of a medium thick pinched cylinder. Compared to the AXI finite elements of code_aster, we realize that for the same fineness of mesh, the SB9 γ 25 gives quite comparable results. Nevertheless the SB9 γ 25 exhibits some weaknesses that deserve to be investigated: stabilization and refinement in thickness direction. The stabilization coefficient must be small to respect the shell kinematics. Moreover, the SB9 γ 25 is formalized in updated Lagrangian which is much slower in nonlinear calculation because it requires a very fine temporal discretization. Moreover, the assumed field method developed in this element is only valid for isotropic linear cases and does not support frictional contact because the transverse shear is considered null on the upper and lower faces. All these aspects motivated the development of a new solid-shell finite element, inspired by the SB9 γ 25 but with a more robust and automatic stabilization, without the need for user-entered stabilization coefficients. This new element is also written in total Lagrangian.

Appendix E

Impacts on `code_aster`

In this chapter we will briefly introduce the implementation process of the SB9 element in `code_aster`. Generally speaking, a finite element is composed of a geometrical support (its mesh), which is enriched with a physics that will drive its behavior during the transformation. We have also seen in chapter 2 that the finite element SB9 has a 9 node hexahedron as geometric support, in other words a classical hexahedron to which a central node is added. Unfortunately, the Mesh module of `code_aster` does not currently know how to produce HEXA9. The first implementation work then consisted in solving this problem. Since `SMesh` (the Mesh module in `Salome_meca`) knows how to produce HEXA8, the simplest solution is to start from this mesh and then to append it with an additional node directly in `code_aster`. This is what we will propose next. It should be noted that this impact study does not aim to go through the source code in the slightest details, but only to show the implementation process in a broad outline. Also we will not comment on the Fortran routines but only the impacted parts of the python catalogs.

E.1 Mesh reading command : LIRE_MAILLAGE

As seen in the theoretical part, the geometrical support of the finite element SB9 is a hexahedron with 9 nodes located as follows:

- Vertices (8 nodes)
- Element center (1 node)

Since `code_aster` did not know a hexahedral element with 9 nodes, it was necessary to start by creating this new (`type_mesh`) which is used as a geometrical (topological) support and to which one can directly associate the reference element SB9. (ELREFE).

The creation of the new mesh HEXA9 is then done by enriching the `mesh_types.py` catalog with the following code snippet:

```
1 # Definition of a new mesh type to which a reference ...
   element named SB9 is associated.
2 HEXA9 = MeshType(nbno=9, dim=3, code='SH9')
3 SH9 = Elrefe()
4 SH9.addLocation('NOEU', 9)
5 SH9.addLocation('NOEU_S', 8)
6 SH9.addLocation('SHB5', 5)
7 SH9.addLocation('SHB7', 7)
8 SH9.addLocation('FPG1', 1)
9 HEXA9.addElrefe(SH9)
```

Actually, this is not the creation of a mesh `HEXA9`, but just a way of defining in the code what a `HEXA9` is. With this piece of code, we teach `code_aster` to recognize a finite element whose mesh type is a `HEXA9`. In this particular case it is informed that a `HEXA9` is a type of mesh of dimension 3, composed of 9 nodes. This mesh is also associated with a reference element of 9 nodes (`NOEU`) of which 8 (`NOEU_S`) are located at the vertices. The reference element (`ELREFE`) works with integration schemes of types `SHB5`, `SHB7` (5 or 7 integration points in thickness). Once this knowledge has been acquired, it is now necessary to go further by specifying the geometrical characteristics of this element: shape functions, position of the gauss points in the natural reference frame of the element, etc. This is the reason why the following Fortran routines have been impacted:

- `elraca.F90` General characteristics of the reference element: number and names of the Gaussian point series, coordinates of the nodes, ...
- `elraga.F90`: Definition of weights and coordinates of Gaussian points
- `elrfvf.F90`: Definition of shape functions
- `elrfdF.F90`: Defining Derivatives of Shape Functions
- `elrfd2.F90`: Definition of the second derivatives of shape functions
- `nuelrf.F90`: Gives the number of the `jni00i` routine associated with a reference element.

In addition, the implementation of this new mesh has changed the number of mesh types known to `code_aster`. This number went from 69 to 71 (plus `SB7`) meshes `ntymax`. Since this number was coded as `hard`, it was necessary to go through a set of routines to update this change: `lrmmdi.F90`, `lrmdfa.F90`, `lrmmma.F90`, `lrmtyp.F90`, `iradhs.F90` ... (voir arbre `thg`).

E.2 Mesh creation : CREA_MAILLAGE

In the previous section, we taught code master to recognize a finite element whose geometrical support is a `HEXA9`. However, since the mesh module of **Salomé_Méca** is not able to provide a mesh of type `HEXA9` directly, we will also code a routine which allows to transform a mesh of type `HEXA8` into a mesh of type `HEXA9`, directly from **code_aster**. This action is done in the `CREA_MAILLAGE` routine.

It's then enriched with the following code :

```

1 from code_aster.Cata.Syntax import *
2 from code_aster.Cata.DataStructure import *
3 from code_aster.Cata.Commons import *
4
5 # added keyword HEXA8_9 as input parameter for mesh ...
6 CREA_MAILLAGE=OPER(nom="CREA_MAILLAGE",op= ...
7     167,sd_prod=maillage_sdaster,
8     reentrant='n',fr=tr("Cree un maillage a ...
9     partir d'un maillage existant"),
10    regles=(UN_PARMIS('COQU_VOLU', 'CREA_FISS', ...
11    'CREA_MAILLE', 'CREA_POI1',
12    'ECLA_PG', 'HEXA20_27', ...
13    'LINE_QUAD', 'MODI_MAILLE',
14    'QUAD_LINE', ...
15    'REPERE', 'RESTREINT', 'PENTA15_18', 'GEOM_FIBRE', ...
16    'DECOUPE_LAC', 'HEXA8_9'),),
17
18 # processing of the keyword factor HEXA8_9
19 HEXA8_9 =FACT(statut='f',fr=tr("Passage HEXA8 -> HEXA9"),
20 regles=(AU_MOINS_UN('TOUT', 'MAILLE', 'GROUP_MA'),),
21 TOUT =SIMP(statut='f',typ='TXM',into=("OUI",) ),
22 MAILLE ...
23     =SIMP(statut='c',typ=ma,validators=NoRepeat(),max='**'),
24 GROUP_MA =SIMP(statut='f',typ=grma ...
25     ,validators=NoRepeat(),max='**'),
26 PREF_NOEUD =SIMP(statut='f',typ='TXM',default="NS"),
27 PREF_NUME =SIMP(statut='f',typ='I',default= 1 ),
28 ),

```

This tells **code_aster** that the user wishes to transform the mesh group from `HEXA8` to `HEXA9`. On receipt of this information the Catalog `crea_mesh` calls the Fortran routine `op0167` which will call the routine `cm0809` programmed to process the keyword factor

HEXA8_9. This routine calls the next two routines:

- cm09nd.F90: generates the coordinates of the middle node of a HEXA9
- cm09ma.F90: takes care of the mesh update

From now on, it is possible to build a mesh of type HEXA9 directly in code_aster. The command looks like :

```
1 # creation of the HEXA9 mesh
2 MESH=CREA_MALLAGE (MAILLAGE=MESH0 , INFO=2 ,
3                   HEXA8_9=_F (TOUT= 'OUI ' , ) , ) ;
```

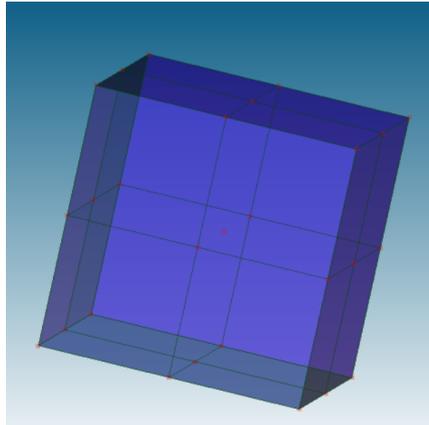


Figure E.1: Cube maillé avec 8 éléments HEXA8

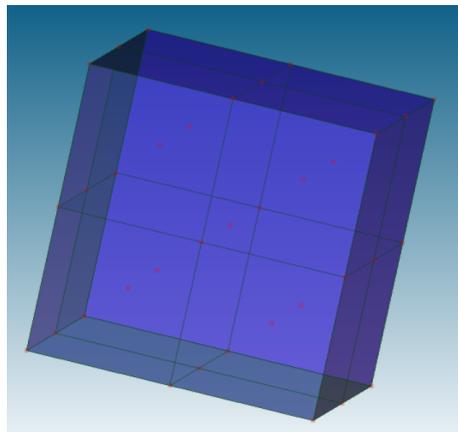


Figure E.2: Mesh cube with 8 elements HEXA9

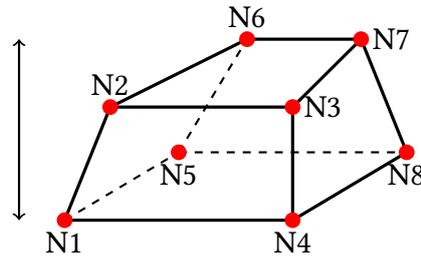


Figure E.3: Connectivité de l'élément

E.3 Mesh modification : MODI_MAILLAGE

Although solid, the SB9 finite element is also a shell element. Therefore its connectivity must be written so that the direction of the thickness of the shell it models is always identified. In other words, the reference element identifies the direction of its thickness from the connectivity of its geometric support HEXA8. To illustrate, let us consider the volume element of the figure E.3. If the connectivity of the element is written as follows: [N1 N2 N3 N4 N5 N6 N7 N8], the calculation will be made assuming that the thickness of the plate coincides with the direction from the [N1 N2 N3 N4] face to the [N5 N6 N7 N8] face or vice versa (note that the thickness is a direction, not a vector, so its sense is irrelevant). Now if the true direction of the thickness is that of the arrow (figure E.3), we will have to rewrite this connectivity as follows: [N1 N4 N8 N5 N2 N3 N7 N6 N2] so the element will know that the direction of the modeled plate is that of the drawn arrow.

As any HEXA8 element produced by a Mesh module such as SMesh (Salomé) has no predefined direction, i.e. its connectivity is written in a random order, it is then imperative that the user indicates the direction of the thickness of the studied plate by asking code_aster to rewrite the connectivity correctly. If this is not done, the calculation can only be correct if, by chance, the mesher has given a correctly oriented connectivity, i.e. coinciding with the right direction (a probability of 1/3). This was the problem with the SHB element. Because of the lack of correct orientation of the connectivity, the gauss points were not aligned in the thickness direction (reduced integration) which led to incorrect results. To avoid these errors, it is then necessary to rewrite the connectivity to orient the element in the direction of its thickness. To do this, the Catalog MODI_MAILLAGE is impacted by the keyword factor ORIE_SB9 which the user uses to indicate to Code_Aster that certain elements of the mesh are SB9 elements and that they must be oriented. The user then provides a group of volume elements corresponding to the SB9 elements to be reoriented and a group of skin elements indicating the thickness direction of these SB9 elements, see figure E.4.

```

1 from code_aster.Cata.Syntax import *
2 from code_aster.Cata.DataStructure import *
3 from code_aster.Cata.Commons import *

```

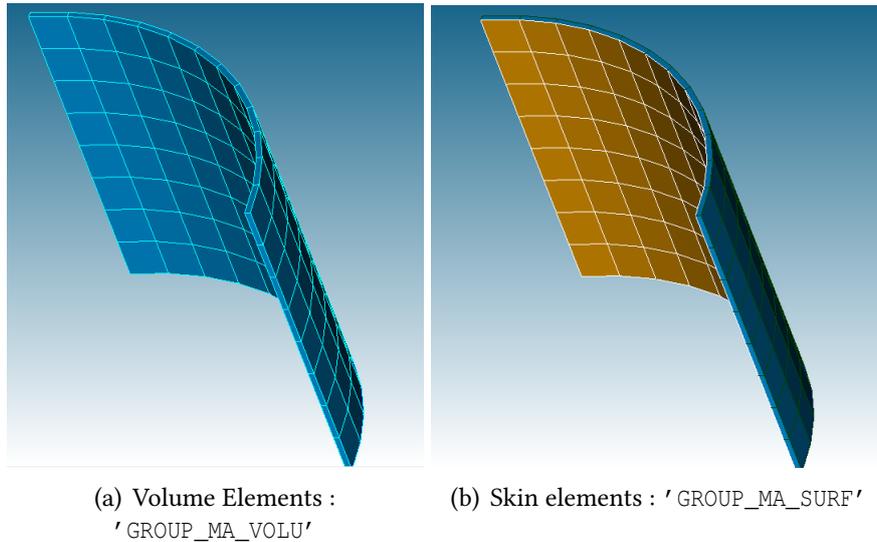


Figure E.4: The two groups of meshes necessary for the reorientation of connectivity

```

4 #
5 #
6 MODI_MAILLAGE=OPER(nom="MODI_MAILLAGE",op= ...
      154,sd_prod=maillage_sdaster,
7           fr=tr("...", " "..."),
8           reentrant='o:MAILLAGE',
9           regles=(AU_MOINS_UN('...', 'ORIE_SB9', '...'),
10 #
11 # Nouvelle implementation
12 ORIE_SB9 =FACT(statut='f',max='**',
13 # Volume group collection
14     GROUP_MA_VOLU ...
15     =SIMP(statut='f',typ=grma,validators=NoRepeat(),max='1'),
16 #Skin mesh group collection
17     GROUP_MA_SURF ...
18     =SIMP(statut='f',typ=grma,validators=NoRepeat(),max='1'),
19     ),

```

The orientation work is done in the Fortran routine `orish9.F90` which retrieves the volume mesh groups 'GROUP_MA_VOLU' to be oriented and the surface mesh groups 'GROUP_MA_SURF' which indicate the orientation directions.

The algorithm consists in going through all the groups of volume meshes and for each of the volume meshes we check if there is one of its faces in all the surface groups. If this is the case, the connectivity is rewritten so that the element is oriented in the

thickness direction.

With this algorithm it was possible to correct the orientation problems of the SHB8 finite element.

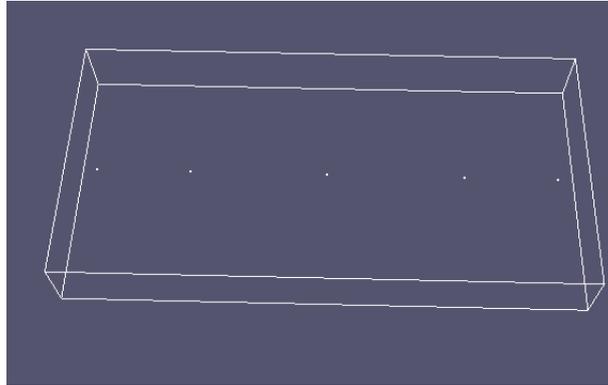


Figure E.5: Problem with the orientation of Gauss points for the SHB8

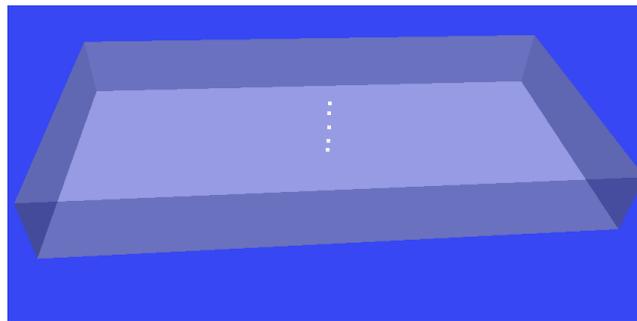


Figure E.6: Gaussian point orientation corrected with Ori_SB9

E.4 Model assignment : AFFE_MODELE

So far, what has been developed concerns only the mesh, the geometrical support of the SB9 finite element. Now we will be able to go into the details of the modeling itself, corresponding to the implementation of the physics that drives the finite element. Once the user has defined his mesh and made the necessary modifications (`modi_mesh`), he can proceed to the creation of the model by adding the following command lines in the command file :

```
1 MODELE = AFFE_MODELE(MAILLAGE = Nom_maillage ,  
2                       AFFE=_F(TOUT = 'OUI',  
3                               MODELISATION = 'SB9',  
4                               PHENOMENE = 'MECANIQUE'))
```

Thus, the user indicates to **code_aster** that he wishes to model a mechanical phenomenon using the physics of the finite element SB9. This choice of modeling is made through the operator `AFFE_MODELE` of **code_aster**. The catalogue `affe_modele.py` is then enriched by an additional modeling which is that of SB9 :

```

1 # ATTENTION THIS CONTENT IS TRUNCATED / REDUCED TO THE ...
  ESSENTIAL :
2 #####
3 AFFE_MODELE=OPER(nom="AFFE_MODELE",op=18,sd_prod=modele_sdaster,
4                 fr=tr("..."),reentrant='n',
5                 regles=(AU_MOINS_UN('AFFE','AFFE_SOUS_STRUC'),),
6                 MAILLAGE
7                 ...
8                 =SIMP(statut='o',typ=maillage_sdaster),
9                 INFO
10                ...
11                =SIMP(statut='f',typ='I',default=1,into=(1,2) ),
12 #
13 AFFE
14                 =FACT(statut='f',max='**',
15                 regles=(UN_PARMIS('TOUT','GROUP_MA','MAILLE'),),
16                 TOUT
17                 ...
18                 =SIMP(statut='f',typ='TXM',into=("OUI",) ),
19                 GROUP_MA
20                 ...
21                 =SIMP(statut='f',typ=grma,validators=NoRepeat(),max='**'),
22                 MAILLE
23                 =SIMP(statut='c',typ=ma
24                 ...
25                 ,validators=NoRepeat(),max='**'),
26                 PHENOMENE
27                 =SIMP(statut='o',typ='TXM',
28                 ...
29                 into=("MECANIQUE","THERMIQUE","ACOUSTIQUE") ),
30                 b_mecanique
31                 =BLOC( condition = ...
32                 ""equal_to("PHENOMENE", 'MECANIQUE') """,
33                 ...
34                 fr=tr("modelisations mecaniques"),
35                 MODELISATION
36                 ...
37                 =SIMP(statut='o',typ='TXM',validators=NoRepeat(),max=10,into=(
38                 "SB9", )

```

Then in the catalogue `phenomenons_modelisations.py` one defines more precisely what this new choice of modeling contains by clearly indicating the meshes involved in this modeling, the physical phenomenon at hand (mechanical), the geometrical and topological dimensions of the elements, the main elements and the edge elements, the attributes etc. The following code summarizes this information:

```
1 #####
2 # For the phenomenon: MECHANICS :
3 #####
4 MECANIQUE = Phenomenon(code='ME')
5 phen = MECANIQUE
6
7 phen.add('SB9', Modelisation(dim=(3,3), code='SB9',
8     attrs=(
9         (AT.NBSIGM, '6'),
10        (AT.TYPMOD, 'COMP3D'),
11    ),
12    elements=(
13        (MT.HEXA9      , EL.MECA_SB9),
14        (MT.QUAD4      , EL.MECA_FACE4),
15        (MT.SEG2       , EL.MECA_ARETE2),
16    )))
```

Given that the 9th node of the finite element SB9 has only one degree of freedom of translation in the thickness direction 'DZ' of the natural frame of the element, it was necessary to indicate this to **code_aster** through the local mode DDL_MECA in the class of the element SB9. (meca_sb9.py):

```
1
2 # local mode relative to SB9
3 DDL_MECA = LocatedComponents(phys=PHY.DEPL_R, ...
4     type='ELNO', diff=True,
5     components=(
6         ('EN1', ('DX', 'DY', 'DZ')),
7         ('EN2', ('DZ')),))
```

In this same catalog is defined the class of the finite element SB9. The class of the element contains in attribute the mesh type and the reference element with which this finite element works. The methods of the class are the set of elementary options. For **code_aster**, an elementary calculation option corresponds to a calculation leading to the production of one or more fields per element. For example, the option 'RIGI_MECA' calculates the stiffness of the element (elastic behavior).

```
1 #
2 # Class of the element followed by these different ...
3     electrical options
```

```

3 class MECA_SB9(Element):
4
5     meshType = MT.HEXA9
6     nodes = (
7         SetOfNodes('EN1', (1,2,3,4,5,6,7,8)),
8         SetOfNodes('EN2', (9,)),
9     )
10    elrefe = (
11        ElrefeLoc(
12            MT.SH9, gauss=(
13                'RIGI=SHB5', 'MASS=SHB5', 'FPG1=FPG1',), ...
14        mater=('RIGI', 'FPG1',),),
15        ElrefeLoc(MT.QU4, gauss = ('RIGI=FPG4', ...
16            'MASS=FPG4',),),
17    )
18    calculs = ( # Options elemntaires
19    )

```

```

1 # EXAMPLE OF A COMMAND FILE CONTAINING TESTED ELEMENTARY ...
2 # CALCULATIONS
3 # IN A MODELIZATION WITH THE ELEMENT SB9
4 #
5 DEBUT();
6
7 #Mesh : Reading a hexa8 mesh contained in a file 'MED'
8 MESH0=LIRE_MAILLAGE(FORMAT='MED',);
9 #modification of the mesh: orientation of the connectivity
10 MESH0=MODI_MAILLAGE(reuse=MESH0,
11                    MAILLAGE=MESH0,
12                    ORIE_SB9=_F(GROUP_MA_VOLU='cube',
13                                GROUP_MA_SURF='face_1',),
14                    );
15 # creation of the HEXA9 mesh
16 MESH=CREA_MAILLAGE(MAILLAGE=MESH0,INFO=2,
17                   HEXA8_9=_F(TOUT='OUI',),);
18 #MATERIAL: Definition and allocation of material
19 MAT=DEFI_MATERIAU(ELAS=_F(E=1,
20                           NU=0.3,),);
21 MATERIAL=AFFE_MATERIAU(MAILLAGE=MESH,

```

```

22             AFFE=(_F(GROUP_MA='cube',
23                     MATER=MAT, ), ), );
24 #Model creation
25 MODEL = AFFE_MODELE(MAILLAGE=MESH,
26                     AFFE=_F(TOUT='OUI',
27                             PHENOMENE='MECANIQUE',
28                             MODELISATION=('SB9', ), ),
29                             INFO=1,
30                             VERI_JACOBIEEN='NON',
31                             ...
32                     DISTRIBUTION=_F(PARTITIONNEUR='METIS',
33                                     ...
34                                     METHODE='SOUS_DOMAINE', ),
35                                     )
36 #IMPR_RESU(UNITE=8,FORMAT='RESULTAT',RESU=_F(MAILLAGE=MESH0))
37 # TEST of the COOR_ELGA computation option
38 COOR_G = CALC_CHAM_ELEM(MODELE=MODEL,
39                         GROUP_MA='cube',
40                         OPTION='COOR_ELGA',
41                         );
42 # TEST the creation of a field at the gauss points of ...
43     the SB9 element
44 SIGMELGA=CREA_CHAMP(TYPE_CHAM='ELGA_SIEF_R',
45                     OPERATION='AFFE',
46                     MODELE=MODEL,
47                     PROL_ZERO='OUI',
48                     INFO=2,
49                     AFFE=_F(TOUT='OUI',
50                             ...
51                             NOM_CMP=('SIXX', 'SIYY', 'SIZZ', ),
52                                     VALE=(1., 1.0, 1.0)));
53 # TEST the creation of a field at the nodes of the SB9 ...
54     element with given values
55 SIGMELNO = CREA_CHAMP(TYPE_CHAM='ELNO_SIEF_R',
56                       OPERATION='AFFE',
57                       MODELE=MODEL,
58                       AFFE=_F(NOM_CMP=('SIXX', 'SIYY', ...
59                                   'SIZZ'),
60                               VALE=(4.0, 5, 6.0),
61                               TOUT='OUI', ),
62                               PROL_ZERO='OUI',

```

```
59             INFO=2,)  
60 # TEST the creation of a field at the nodes of the SB9 ...  
    element with given values  
61 DEPELNO = CREA_CHAMP(TYPE_CHAM='NOEU_DEPL_R',  
62                     OPERATION='AFFE',  
63                     MODELE=MODEL,  
64                     AFFE=_F(NOM_CMP=('DX', 'DY', 'DZ'),  
65                             VALE=(1.0, 1.0, 10),  
66                             TOUT='OUI',),  
67                     INFO=2,)  
68 # LOAD ASSIGNMENT (clamp-on)  
69 IMPO=AFFE_CHAR_MECA(MODELE=MODEL,  
70                    DDL_IMPO=( _F(GROUP_MA = ('base'),  
71                                LIAISON = 'ENCASTRE',  
72                                ),  
73                                ),  
74  
75                    )  
76 # AFFECTATION D'UN CHARGEMENT (Pressure)  
77 CHAR=AFFE_CHAR_MECA(MODELE=MODEL,  
78                    PRES_REP=( _F(GROUP_MA=('face_1'), ...  
79                    PRES=1) ),),  
80  
81                    );  
82 # STATIC STUDY  
83 RESU=MECA_STATIQUE(MODELE=MODEL,  
84                    CHAM_MATER=MATERIAL,  
85                    ...  
86                    EXCIT=( _F(CHARGE=CHAR), _F(CHARGE=IMPO)))  
87  
88 FIN();
```

E.5 Conclusion

The approach proposed here is valid for SB7 except that SB7 does not need reorientation. The initial connectivity immediately gives the direction of the thickness.

Appendix F

Matlab source code

F.1 SB9 Matlab Code

```
1 function HYPER3D(PROP,UPDATE, LTAN, XYZ,Connec)
2 %*****
3 % MAIN PROGRAM COMPUTING GLOBAL STIFFNESS MATRIX RESIDUAL FORCE FOR
4 % Hyperelastic material model assembly
5 %*****
6 % initialisation
7 global DISPTD FORCE GKF SIGMA
8 [NUMNP, NDOF] = size(XYZ); % Analysis parameters
9 NE = size(Connec,1);
10 p1=NUMNP+1;
11 p2=NUMNP+NE;
12 DDLsup=p1:p2;
13 NEQ = NDOF*NUMNP;
14 LE=[Connec DDLsup'];
15 %
16 % Integration points and weights
17 ZG=[-1 -sqrt(3/7) 0 sqrt(3/7) 1];
18 WGT=[0.1 49.0/90.0 32.0/45.0 49.0/90.0 0.1];
19 %% matrice tangente de stabilisation(Deviatorique)
20 Cstab=mattddev;
21 %%
22 %
23 % Index for history variables (each integration pt)
24 INTN=0;
25 %
26 %LOOP OVER ELEMENTS, THIS IS MAIN LOOP TO COMPUTE K AND F
```

```

27 for IE=1:NE
28 % Nodal coordinates and incremental displacements
29 ELXY=XYZ(LE(IE,1:8),:);
30 % Local to global mapping
31 IDOF=zeros(1,25);
32 for I=1:8
33     II=(I-1)*NDOF+1;
34     IDOF(II:II+2)=(LE(IE,I)-1)*NDOF+1:(LE(IE,I)-1)*NDOF+3;
35 end
36 %%%%%%%%%%%
37 IDOF(25)=NEQ+IE;
38 %%%%%%%%%%%
39 DSP=DISPTD(IDOF(1:24));
40 w9=DISPTD(IDOF(25));
41 DSP=reshape(DSP,NDOF,8);
42 %%
43 ELxy=ELXY+DSP';
44 %LOOP OVER INTEGRATION POINTS
45 Ueff=0.0;
46 [B0,BZETA,BZZ,BXIdev,BETAdev,...
47 BXIETAdev,BETAZETAdev,BXIZETAdev]=matriceBcart(ELXY,ELxy);
48 J0=Jacobien(ELXY,[0 0 0]);
49 DET= det(J0);
50 %
51 for LZ=1:5
52     E3=ZG(LZ);
53     INTN = INTN + 1;
54     %
55     % Determinant and shape function derivatives
56     FAC=4*WGT(LZ)*DET;
57     %
58     % deformation de Gren Lagrange centrÃe
59     E0=deformEcart(DSP,ELXY,[0 0 E3]);
60     B9=calculB9(ELXY,[0 0 E3]);
61     E0=E0+B9*w9;
62     % Computer stress and tangent stiffness
63     DTAN= matrice_D(PROP(1), PROP(2));
64     % Loi de comportement de Saint Venant Kirchhoff
65     STRESS=DTAN*E0;
66     % Update plastic variables
67     if UPDATE
68         SIGMA(:,INTN)=STRESS;
69         continue;

```

```

70     end
71     % Calcul du module de cisaillement effectif
72     Ueff=DTAN(5,5);
73     % calcul des matrices de gradient BN et BG
74     Bm=B0+E3*BZETA+E3*E3*BZZ;
75     BN=[Bm B9];
76     %BG=BG0+E3*BGzeta;
77     %% pour force interne
78     %size(FORCE(IDOF))
79     FORCE (IDOF)=FORCE (IDOF)-FAC*BN'*STRESS;
80     % Tangent stiffness
81     if LTAN
82         Km=BN'*DTAN*BN*FAC;
83         Kg=rigigeo(STRESS,ELXY,[0 0 E3]);
84         Km(1:24,1:24)=Km(1:24,1:24)+FAC*Kg;
85         GKF (IDOF ,IDOF)=GKF (IDOF ,IDOF)+Km;
86     end
87 end;
88
89 if UPDATE==false
90     IDOFS=IDOF(1:24);
91     [SH1,SH2,SH12,SH23,SH13]=constraintstab(DSP,ELXY,Ueff*Cstab);
92     Fintstab=DET*(BXIdev'*SH1/3+...
93             BETAdev'*SH2/3+...
94             0*BXIETAdev'*SH12/9+...
95             BETAZETAdev'*SH23/9+...
96             BXIZETAdev'*SH13/9)*8;
97     FORCE (IDOFS)=FORCE (IDOFS)-Fintstab;
98 end
99 if LTAN
100     Kmstab=rigistab(BXIdev,BETAdev,BXIETAdev*0,...
101                   BETAZETAdev,BXIZETAdev,Ueff*Cstab);
102     Kgstab=rigigeostab1(ELXY,SH1,SH2,SH23,SH13,0*SH12);
103     GKF (IDOFS ,IDOFS)=GKF (IDOFS ,IDOFS)+DET*Kmstab+DET*Kgstab;
104 end
105 end
106
107 end

```

```

1 function [B0,BZETA,BZZ,BXIdev,...
2          BETAdev,BXIETAdev,BETAZETAdev,...

```

```

3  BXIZETAdev]=matriceBcart(XYZ,xyz)
4
5  % ICI IL FAUT RENTRER AVEC LES POSITIONS ACTUELLES xyz et Initiale ...
   XYZ
6  %
7  [Bc0,BcZETA,BcZZ,BcXI,BcETA,...
8  BcXIETA,BcETAZETA,BcXIZETA]=matriceBc(xyz);
9  [T0,TXI,TETA,TZETA]=matriceTdecomp(XYZ);
10 %
11 B0=T0*Bc0;
12 BZETA=T0*BcZETA+TZETA*Bc0;
13 BZZ=T0*BcZZ+TZETA*BcZETA;
14 %
15 BXI=T0*BcXI+TXI*Bc0;
16 BETA=T0*BcETA+TETA*Bc0;
17 BETAZETA=T0*BcETAZETA+TETA*BcZETA+TZETA*BcETA;
18 BXIETA=T0*BcXIETA+TXI*BcETA+TETA*BcXI;
19 BXIZETA=T0*BcXIZETA+TXI*BcZETA+TZETA*BcXI;
20 %
21 % Enlever les parties deviatoriques sur les B de stabilisation
22 BXIv=zeros(6,24);
23 %%
24 for i=1:8
25     alpha1=0;
26     alpha2=0;
27     alpha3=0;
28     col=3*(i-1)+1:3*(i-1)+3;
29     B=BXI(1:3,col);
30     for j=1:3
31         alpha1=alpha1+B(j,1);
32         alpha2=alpha2+B(j,2);
33         alpha3=alpha3+B(j,3);
34     end
35     BXIv(:,col)=[alpha1 alpha2 alpha3;
36                 alpha1 alpha2 alpha3;
37                 alpha1 alpha2 alpha3;
38                 0      0      0      ;
39                 0      0      0      ;
40                 0      0      0      ]/3;
41 end
42
43 BXIdev=BXI-BXIv;
44 %%

```

```

45 BETAv=zeros(6,24);
46 for i=1:8
47     alpha1=0;
48     alpha2=0;
49     alpha3=0;
50     col=3*(i-1)+1:3*(i-1)+3;
51     B=BETA(1:3,col);
52     for j=1:3
53         alpha1=alpha1+B(j,1);
54         alpha2=alpha2+B(j,2);
55         alpha3=alpha3+B(j,3);
56     end
57     BETAv(:,col)=[alpha1 alpha2 alpha3;
58                 alpha1 alpha2 alpha3;
59                 alpha1 alpha2 alpha3;
60                 0      0      0      ;
61                 0      0      0      ;
62                 0      0      0      ]/3;
63 end
64 BETAdev=BETA-BETAv;
65 %%
66 BXIETAv=zeros(6,24);
67 for i=1:8
68     alpha1=0;
69     alpha2=0;
70     alpha3=0;
71     col=3*(i-1)+1:3*(i-1)+3;
72     B=BXIETA(1:3,col);
73     for j=1:3
74         alpha1=alpha1+B(j,1);
75         alpha2=alpha2+B(j,2);
76         alpha3=alpha3+B(j,3);
77     end
78     BXIETAv(:,col)=[alpha1 alpha2 alpha3;
79                    alpha1 alpha2 alpha3;
80                    alpha1 alpha2 alpha3;
81                    0      0      0      ;
82                    0      0      0      ;
83                    0      0      0      ]/3;
84 end
85 BXIETAdev=BXIETA-BXIETAv;
86 %%
87 BETAZETAv=zeros(6,24);

```

```

88 for i=1:8
89 alpha1=0;
90 alpha2=0;
91 alpha3=0;
92     col=3*(i-1)+1:3*(i-1)+3;
93     B=BETAZETA(1:3,col);
94     for j=1:3
95         alpha1=alpha1+B(j,1);
96         alpha2=alpha2+B(j,2);
97         alpha3=alpha3+B(j,3);
98     end
99     BETAZETA(:,col)=[alpha1 alpha2 alpha3;
100                    alpha1 alpha2 alpha3;
101                    alpha1 alpha2 alpha3;
102                    0      0      0      ;
103                    0      0      0      ;
104                    0      0      0      ]/3;
105 end
106 BETAZETAdev=BETAZETA-BETAZETAdev;
107 %%
108 BXIZETAdev=zeros(6,24);
109 for i=1:8
110 alpha1=0;
111 alpha2=0;
112 alpha3=0;
113     col=3*(i-1)+1:3*(i-1)+3;
114     B=BXIZETA(1:3,col);
115     for j=1:3
116         alpha1=alpha1+B(j,1);
117         alpha2=alpha2+B(j,2);
118         alpha3=alpha3+B(j,3);
119     end
120     BXIZETA(:,col)=[alpha1 alpha2 alpha3;
121                    alpha1 alpha2 alpha3;
122                    alpha1 alpha2 alpha3;
123                    0      0      0      ;
124                    0      0      0      ;
125                    0      0      0      ]/3;
126 end
127 BXIZETAdev=BXIZETA-BXIZETAdev;
128
129 % BXIdev(5:6,:) = 0.0;
130 % BETAdev(5:6,:) = 0.0;

```

```

131 % BXIETAdev(5:6,:)=0.0;
132 % BETAZETAdev(5:6,:)=0.0;
133 % BXIZETAdev(5:6,:)=0.0;
134 end

```

```

1 function ...
   [Bc0,BcZETA,BcZZ,BcXI,BcETA,BcXIETA,BcETAZETA,BcXIZETA]=matriceBc(XYZ)
2
3 % ICI IL FAUT RENTRER AVEC LES POSITIONS ACTUELLES
4 [J10,J1ETA,J1ZETA,J1ETAZETA,...
5  J20,J2XI,J2ZETA,J2XIZETA,...
6  ~,~,~,~]=jacobienDecompose(XYZ);
7
8 %%
9 g1=[-1 1 1 -1 -1 1 1 -1]/8;
10 g2=[-1 -1 1 1 -1 -1 1 1]/8;
11 g3=[-1 -1 -1 -1 1 1 1 1]/8;
12 %
13 h1=[1 -1 1 -1 1 -1 1 -1]/8;
14 h2=[1 1 -1 -1 -1 -1 1 1]/8;
15 h3=[1 -1 -1 1 -1 1 1 -1]/8;
16 h4=[-1 1 -1 1 1 -1 1 -1]/8;
17 %
18 %% TYING POINTS
19 % A Ã D
20 XAD=[-1 -1 0
21       1 -1 0
22       1 1 0
23       -1 1 0];
24 % E TO H
25 XEH=[-1 0 -1
26       1 0 -1
27       1 0 1
28       -1 0 1];
29 % J TO M
30 XJM=[0 -1 -1
31       0 1 -1
32       0 1 1
33       0 -1 1];
34 %%
35 Bc0=zeros(6,24);

```

```

36 for I=1:8
37     aux33=zeros(3,1);
38     aux23=zeros(3,1);
39     aux13=zeros(3,1);
40     for AD=1:4
41         [~,~,J3]=colomnejacobien(XYZ,XAD(AD,:));
42         DSF = DSHAPE(XAD(AD,:));
43         aux33=aux33+DSF(3,I)*J3/4;
44     end
45     for EH=1:4
46         [~,J2,J3]=colomnejacobien(XYZ,XEH(EH,:));
47         DSF = DSHAPE(XEH(EH,:));
48         aux23=aux23+(DSF(2,I)*J3+DSF(3,I)*J2)/4;
49     end
50     for JM=1:4
51         [J1,~,J3]=colomnejacobien(XYZ,XJM(JM,:));
52         DSF = DSHAPE(XJM(JM,:));
53         aux13=aux13+(DSF(1,I)*J3+DSF(3,I)*J1)/4;
54     end
55     COL=3*(I-1)+1:3*(I-1)+3;
56     Bc0(:,COL)=[g1(I)*J10(1)  g1(I)*J10(2)  g1(I)*J10(3);
57                g2(I)*J20(1)  g2(I)*J20(2)  g2(I)*J20(3);
58                aux33(1)      aux33(2)      aux33(3);
59                g1(I)*J20(1)+g2(I)*J10(1)  ...
60                g1(I)*J20(2)+g2(I)*J10(2)  g1(I)*J20(3)+g2(I)*J10(3);
61                aux23(1)      aux23(2)      aux23(3);
62                aux13(1)      aux13(2)      aux13(3)];
63 end
64 %% BcZETA
65 BcZETA=zeros(6,24);
66 for I=1:8
67     aux23=zeros(3,1);
68     aux13=zeros(3,1);
69     for EH=1:4
70         [~,J2,J3]=colomnejacobien(XYZ,XEH(EH,:));
71         DSF = DSHAPE(XEH(EH,:));
72         aux23=aux23+XEH(EH,3)*(DSF(2,I)*J3+DSF(3,I)*J2)/4;
73     end
74     for JM=1:4
75         [J1,~,J3]=colomnejacobien(XYZ,XJM(JM,:));
76         DSF = DSHAPE(XJM(JM,:));
77         aux13=aux13+XJM(JM,3)*(DSF(1,I)*J3+DSF(3,I)*J1)/4;
78     end

```

```

78     COL=3*(I-1)+1:3*(I-1)+3;
79     AUX12=h2(I)*J10+h3(I)*J20+g2(I)*J1ZETA+g1(I)*J2ZETA ;
80     BcZETA(:,COL)=[ h3(I)*J10(1)+g1(I)*J1ZETA(1) ...
81                    h3(I)*J10(2)+g1(I)*J1ZETA(2)  h3(I)*J10(3)+g1(I)*J1ZETA(3);
                        h2(I)*J20(1)+g2(I)*J2ZETA(1) ...
82                    h2(I)*J20(2)+g2(I)*J2ZETA(2)  h2(I)*J20(3)+g2(I)*J2ZETA(3);
                        0      0      0;
83                    AUX12(1) AUX12(2) AUX12(3);
84                    aux23(1)   aux23(2)   aux23(3);
85                    aux13(1)   aux13(2)   aux13(3)];
86 end
87 %% ZETA ZETA
88 BcZZ=zeros(6,24);
89 for I=1:8
90     COL=3*(I-1)+1:3*(I-1)+3;
91     BcZZ(:,COL)=[h3(I)*J1ZETA(1)  h3(I)*J1ZETA(2)  h3(I)*J1ZETA(3);
92                  h2(I)*J2ZETA(1)  h2(I)*J2ZETA(2)  h2(I)*J2ZETA(3);
93                  0      0      0;
94                  h2(I)*J1ZETA(1)+h3(I)*J2ZETA(1) ...
95                  h2(I)*J1ZETA(2)+h3(I)*J2ZETA(2) h2(I)*J1ZETA(3)+h3(I)*J2ZETA(3);
96                  0      0      0;
97                  0      0      0];
98 end
99 %% BcXI
100 BcXI=zeros(6,24);
101 for I=1:8
102     AUX33=zeros(3,1);
103     aux23=zeros(3,1);
104     for AD=1:4
105         [~,~,J3]=colonnejacobien(XYZ,XAD(AD,:));
106         DSF = DSHAPE(XAD(AD,:));
107         AUX33=AUX33+XAD(AD,1)*(DSF(3,I)*J3)/4;
108     end
109     for EH=1:4
110         [~,J2,J3]=colonnejacobien(XYZ,XEH(EH,:));
111         DSF = DSHAPE(XEH(EH,:));
112         aux23=aux23+XEH(EH,1)*(DSF(2,I)*J3+DSF(3,I)*J2)/4;
113     end
114     COL=3*(I-1)+1:3*(I-1)+3;
115     BcXI(:,COL)=[ 0      0      0;
116                  h1(I)*J20(1)+g2(I)*J2XI(1) ...
117                  h1(I)*J20(2)+g2(I)*J2XI(2)  h1(I)*J20(3)+g2(I)*J2XI(3);
118                  AUX33(1)   AUX33(2)   AUX33(3)];

```

```

117             h1(I)*J10(1)+g1(I)*J2XI(1) ...
118 h1(I)*J10(2)+g1(I)*J2XI(2) h1(I)*J10(3)+g1(I)*J2XI(3);
119             aux23(1)      aux23(2)      aux23(3);
120             0      0      0];
121 end
122 %% ETA
123 BcETA=zeros(6,24);
124 for I=1:8
125     AUX33=zeros(3,1);
126     aux13=zeros(3,1);
127     for AD=1:4
128         [~,~,J3]=colomnejacobien(XYZ,XAD(AD,:));
129         DSF = DSHAPE(XAD(AD,:));
130         AUX33=AUX33+XAD(AD,2)*(DSF(3,I)*J3)/4;
131     end
132     for JM=1:4
133         [J1,~,J3]=colomnejacobien(XYZ,XJM(JM,:));
134         DSF = DSHAPE(XJM(JM,:));
135         aux13=aux13+XJM(JM,2)*(DSF(1,I)*J3+DSF(3,I)*J1)/4;
136     end
137     COL=3*(I-1)+1:3*(I-1)+3;
138     BcETA(:,COL)=[ h1(I)*J10(1)+g1(I)*J1ETA(1) ...
139 h1(I)*J10(2)+g1(I)*J1ETA(2) h1(I)*J10(3)+g1(I)*J1ETA(3);
140             0      0      0;
141             AUX33(1)      AUX33(2)      AUX33(3);
142             h1(I)*J20(1)+g2(I)*J1ETA(1) ...
143 h1(I)*J20(2)+g2(I)*J1ETA(2) h1(I)*J20(3)+g2(I)*J1ETA(3);
144             0      0      0;
145             aux13(1)      aux13(2)      aux13(3)];
146 end
147 %% XIETA
148 BcXIETA=zeros(6,24);
149 for I=1:8
150     AUX33=zeros(3,1);
151     for AD=1:4
152         [~,~,J3]=colomnejacobien(XYZ,XAD(AD,:));
153         DSF = DSHAPE(XAD(AD,:));
154         AUX33=AUX33+XAD(AD,2)*XAD(AD,1)*(DSF(3,I)*J3)/4;
155     end
156     COL=3*(I-1)+1:3*(I-1)+3;
157     BcXIETA(:,COL)=[0      0      0;
158             0      0      0;
159             AUX33(1)      AUX33(2)      AUX33(3);

```

```

157             h1(I)*J2XI(1)+h1(I)*J1ETA(1) ...
158             h1(I)*J2XI(2)+h1(I)*J1ETA(2) h1(I)*J2XI(3)+h1(I)*J1ETA(3);
159             0      0      0;
160             0      0      0];
161 end
162 %% ETA ZETA
163 BcETAZETA=zeros(6,24);
164 for I=1:8
165     AUX11=h4(I)*J10+h3(I)*J1ETA+h1(I)*J1ZETA+g1(I)*J1ETAZETA;
166     AUX12=h4(I)*J20+h2(I)*J1ETA+h1(I)*J2ZETA+g2(I)*J1ETAZETA;
167     AUX13=[0 0 0]';
168     for JM=1:4
169         [J1,~,J3]=colomnejacobien(XYZ,XJM(JM,:));
170         DSF = DSHAPE(XJM(JM,:));
171         AUX13=AUX13+XJM(JM,2)*XJM(JM,3)*(DSF(1,I)*J3+DSF(3,I)*J1)/4;
172     end
173     COL=3*(I-1)+1:3*(I-1)+3;
174     BcETAZETA(:,COL)=[ AUX11(1)  AUX11(2)  AUX11(3);
175                       0      0      0;
176                       0      0      0;
177                       AUX12(1)  AUX12(2)  AUX12(3);
178                       0      0      0;
179                       AUX13(1)   AUX13(2)   AUX13(3)];
180 end
181 %% XI ZETA
182 BcXIZETA=zeros(6,24);
183 for I=1:8
184     AUX22=h4(I)*J20+h2(I)*J2XI+h1(I)*J2ZETA+g2(I)*J2XIZETA;
185     AUX13=h4(I)*J10+h3(I)*J2XI+h1(I)*J1ZETA+g1(I)*J2XIZETA;
186     AUX23=[0 0 0]';
187     for EH=1:4
188         [~,J2,J3]=colomnejacobien(XYZ,XEH(EH,:));
189         DSF = DSHAPE(XEH(EH,:));
190         AUX23=AUX23+XEH(EH,1)*XEH(EH,3)*(DSF(2,I)*J3+DSF(3,I)*J2)/4;
191     end
192     COL=3*(I-1)+1:3*(I-1)+3;
193     BcXIZETA(:,COL)=[ 0      0      0;
194                       AUX22(1)  AUX22(2)  AUX22(3);
195                       0      0      0;
196                       AUX13(1)  AUX13(2)  AUX13(3);
197                       AUX23(1)  AUX23(2)  AUX23(3);
198                       0      0      0];

```

F. Matlab source code

```
199 end
200 end
```

```
1 function [J10,J1ETA,J1ZETA,J1ETAZETA,...
2           J20,J2XI, J2ZETA,J2XIZETA,...
3           J30,J3ETA,J3XI, J3XIETA]=jacobienDecompose(XYZ)
4
5 % XYZ == CoordonnÃes initiales
6 %
7 U=XYZ(:,1);
8 V=XYZ(:,2);
9 W=XYZ(:,3);
10 %
11 g1=[-1 1 1 -1 -1 1 1 -1]/8;
12 g2=[-1 -1 1 1 -1 -1 1 1]/8;
13 g3=[-1 -1 -1 -1 1 1 1 1]/8;
14 %
15 h1=[1 -1 1 -1 1 -1 1 -1]/8;
16 h2=[1 1 -1 -1 -1 -1 1 1]/8;
17 h3=[1 -1 -1 1 -1 1 1 -1]/8;
18 h4=[-1 1 -1 1 1 -1 1 -1]/8;
19 %
20 J10=[dot(g1,U) dot(g1,V) dot(g1,W)];
21 J1ETA=[dot(h1,U) dot(h1,V) dot(h1,W)];
22 J1ZETA=[dot(h3,U) dot(h3,V) dot(h3,W)];
23 J1ETAZETA=[dot(h4,U) dot(h4,V) dot(h4,W)];
24 %
25 J20=[dot(g2,U) dot(g2,V) dot(g2,W)];
26 J2XI=[dot(h1,U) dot(h1,V) dot(h1,W)];
27 J2ZETA=[dot(h2,U) dot(h2,V) dot(h2,W)];
28 J2XIZETA=[dot(h4,U) dot(h4,V) dot(h4,W)];
29 %
30 J30=[dot(g3,U) dot(g3,V) dot(g3,W)];
31 J3XI=[dot(h3,U) dot(h3,V) dot(h3,W)];
32 J3ETA=[dot(h2,U) dot(h2,V) dot(h2,W)];
33 J3XIETA=[dot(h4,U) dot(h4,V) dot(h4,W)];
34 end
```

```

1 function [J1,J2,J3]=colomnejacobien(XYZ,XI)
2 %
3 [J10,J1ETA,J1ZETA,J1ETAZETA,...
4  J20,J2XI, J2ZETA,J2XIZETA,...
5  J30,J3ETA,J3XI, J3XIETA]=jacobienDecompose(XYZ);
6 %
7 J1=J10+XI(2)*J1ETA+XI(3)*J1ZETA+XI(2)*XI(3)*J1ETAZETA ;
8 J2=J20+XI(1)*J2XI+XI(3)*J2ZETA+XI(1)*XI(3)*J2XIZETA ;
9 J3=J30+XI(2)*J3ETA+XI(1)*J3XI+XI(1)*XI(2)*J3XIETA;
10 %
11 J1=J1';
12 J2=J2';
13 J3=J3';
14 end

```

```

1 function [D1,D2,D3]=colomneDefconv(DSP,XI)
2
3 [D10,D1ETA,D1ZETA,D1ETAZETA,...
4  D20,D2XI,D2ZETA,D2XIZETA,...
5  D30,D3ETA,D3XI,D3XIETA]=deformaconva(DSP);
6
7 D1=D10+XI(2)*D1ETA+XI(3)*D1ZETA+XI(2)*XI(3)*D1ETAZETA ;
8 D2=D20+XI(1)*D2XI+XI(3)*D2ZETA+XI(1)*XI(3)*D2XIZETA ;
9 D3=D30+XI(2)*D3ETA+XI(1)*D3XI+XI(1)*XI(2)*D3XIETA;
10
11 D1=D1';
12 D2=D2';
13 D3=D3';
14 end

```

```

1 function [SH1,SH2,SH12,SH23,SH13]=constraintestab(DSP,XYZ,C)
2 % CALCUL DES CONTRAINTES DE STABILISATION
3 %
4 %
5 %[EH1,EH2,EH23,EH13]=deformEcartStab(DSP,XYZ);
6 [EH1,EH2,EH12,EH23,EH13]=deformEcartStab(DSP,XYZ);
7 % On considère la partie d'él'viation de ces deformations
8 %

```

F. Matlab source code

```
9 I=[1 1 1 0 0 0]';
10 EH1=EH1-(EH1(1)+EH1(2)+EH1(3))*I/3;
11 EH2=EH2-(EH2(1)+EH2(2)+EH2(3))*I/3;
12 EH13=EH13-(EH13(1)+EH13(2)+EH13(3))*I/3;
13 EH23=EH23-(EH23(1)+EH23(2)+EH23(3))*I/3;
14 EH12=EH12-(EH12(1)+EH12(2)+EH12(3))*I/3;
15 %
16
17 SH1=C*EH1;
18 SH2=C*EH2;
19 SH13=C*EH13;
20 SH23=C*EH23;
21 SH12=C*EH12;
22 end
```

```
1 function E0=deformEcart(DSP,XYZ,XI)
2
3 % XYZ == CoordonnÃles initiales
4 % DSP == DÃplacement total
5 [Ec0,Eczeta,Eczz,~,~,~,~]=deformatEc(DSP,XYZ);
6 [T0,~,~,TZETA]=matriceTdecomp(XYZ);
7
8 %% DEFORMATION DEPENDANT DE ZETA ET CONSTANCE
9 E0=T0*Ec0+XI(3)*(T0*Eczeta+TZETA*Ec0)+XI(3)*XI(3)*(T0*Eczz+TZETA*Eczeta);
10
11 end
```

```
1 function ...
2     [Ec0,Eczeta,Eczz,EcXI,EcETA,EcXIETA,EcETAZETA,EcXIZETA]=deformatEc(DSP,XYZ)
3
4 % iCI LES COORDONNÃLES SONT LES COORDONNÃLES INITIALES
5 % ALERTE ECRITURE VERIFIEE
6 %
7 [J10,J1ETA,J1ZETA,J1ETAZETA,...
8  J20,J2XI,J2ZETA,J2XIZETA,...
9  J30,J3ETA,J3XI,J3XIETA]=jacobienDecompose(XYZ);
10
11 [D10,D1ETA,D1ZETA,D1ETAZETA,...
```

```

11     D20,D2XI,D2ZETA,D2XIZETA,...
12     D30,D3ETA,D3XI,D3XIETA]=deformaconva(DSP);
13 %%
14 % A Ã D
15 XIAD=[-1 -1 0
16         1 -1 0
17         1 1 0
18        -1 1 0];
19 % E TO H
20 XIEH=[-1 0 -1
21         1 0 -1
22         1 0 1
23        -1 0 1];
24 % J TO M
25 XIJM=[0 -1 -1
26         0 1 -1
27         0 1 1
28         0 -1 1];
29 %%
30 Ec0=zeros(6,1);
31 Ec0(1)=dot(J10,D10)+0.5*dot(D10,D10);
32 Ec0(2)=dot(J20,D20)+0.5*dot(D20,D20);
33 Ec0(4)=dot(J10,D20)+dot(J20,D10)+dot(D10,D20);
34 %%
35 Eczeta=zeros(6,1);
36 Eczeta(1)=dot(J10,D1ZETA)+dot(D10,J1ZETA)+dot(D10,D1ZETA);
37 Eczeta(2)=dot(J20,D2ZETA)+dot(D20,J2ZETA)+dot(D20,D2ZETA);
38 Eczeta(4)=dot(J10,D2ZETA)+dot(J1ZETA,D20)+dot(J20,D1ZETA)+dot(J2ZETA,D10)...
39           +dot(D10,D2ZETA)+dot(D1ZETA,D20);
40
41 %%
42 Eczz=zeros(6,1);
43 Eczz(1)=dot(J1ZETA,D1ZETA)+0.5*dot(D1ZETA,D1ZETA);
44 Eczz(2)=dot(J2ZETA,D2ZETA)+0.5*dot(D2ZETA,D2ZETA);
45 Eczz(4)=dot(J1ZETA,D2ZETA)+dot(J2ZETA,D1ZETA)+dot(D2ZETA,D1ZETA);
46 %% STABILISATION XI
47 EcXI=zeros(6,1);
48 EcXI(2)=dot(J20,D2XI)+dot(J2XI,D20)+dot(D2XI,D20);
49 EcXI(4)=dot(J10,D2XI)+dot(J2XI,D10)+dot(D2XI,D10);
50 %% STABILISATION ETA
51 EcETA=zeros(6,1);
52 EcETA(1)=dot(J10,D1ETA)+dot(J1ETA,D10)+dot(D1ETA,D10);
53 EcETA(4)=dot(J20,D1ETA)+dot(J1ETA,D20)+dot(D1ETA,D20);

```

```

54 %% STABILISATION XIETA
55 EcXIETA=zeros(6,1);
56 EcXIETA(4)=dot(J1ETA,D2XI)+dot(J2XI,D1ETA)+dot(D2XI,D1ETA);
57 %% CHAMPS ASSUMES
58 for AD=1:4
59     [~,~,J3]=colonnejacobien(XYZ,XIAD(AD,:));
60     [~,~,D3]=colonneDefconv(DSP,XIAD(AD,:));
61     EcXIETA(3)= ...
        EcXIETA(3)+XIAD(AD,2)*XIAD(AD,1)*(dot(J3,D3)+0.5*dot(D3,D3))/4;
62 end
63 %% STABILISATION ETAZETA
64 EcETAZETA=zeros(6,1);
65 EcETAZETA(1)=dot(J10,D1ETAZETA)+dot(J1ETA,D1ZETA)+dot(J1ZETA,D1ETA)...
        +dot(J1ETAZETA,D10)+dot(D1ETAZETA,D10)+dot(D1ETA,D1ZETA);
66 EcETAZETA(4)=dot(J20,D1ETAZETA)+dot(J1ETA,D2ZETA)+dot(J2ZETA,D1ETA)...
        +dot(J1ETAZETA,D20)+dot(D1ETAZETA,D20)+dot(D1ETA,D2ZETA);
69 %% STABILISATION XIZETA
70 EcXIZETA=zeros(6,1);
71 EcXIZETA(2)=dot(J20,D2XIZETA)+dot(J2XI,D2ZETA)+dot(J2ZETA,D2XI)...
        +dot(J2XIZETA,D20)+dot(D2XIZETA,D20)+dot(D2XI,D2ZETA);
72 EcXIZETA(4)=dot(J10,D2XIZETA)+dot(J2XI,D1ZETA)+dot(J1ZETA,D2XI)...
        +dot(J2XIZETA,D10)+dot(D2XIZETA,D10)+dot(D2XI,D1ZETA);
75 %% CHAMPS ASSUMES
76 for AD=1:4
77     [~,~,J3]=colonnejacobien(XYZ,XIAD(AD,:));
78     [~,~,D3]=colonneDefconv(DSP,XIAD(AD,:));
79     Ec0(3)= Ec0(3)+(dot(J3,D3)+0.5*dot(D3,D3))/4;
80     EcXI(3)= EcXI(3)+XIAD(AD,1)*(dot(J3,D3)+0.5*dot(D3,D3))/4;
81     EcETA(3)= EcETA(3)+XIAD(AD,2)*(dot(J3,D3)+0.5*dot(D3,D3))/4;
82     EcXIETA(3)= ...
        EcXIETA(3)+XIAD(AD,2)*XIAD(AD,1)*(dot(J3,D3)+0.5*dot(D3,D3))/4;
83 end
84
85 for EH=1:4
86     [~,J2,J3]=colonnejacobien(XYZ,XIEH(EH,:));
87     [~,D2,D3]=colonneDefconv(DSP,XIEH(EH,:));
88     Ec0(5)= Ec0(5)+(dot(J3,D2)+dot(J2,D3)+dot(D3,D2))/4;
89     Eczeta(5)= ...
        Eczeta(5)+XIEH(EH,3)*(dot(J3,D2)+dot(J2,D3)+dot(D3,D2))/4;
90     EcXI(5)= EcXI(5)+XIEH(EH,1)*(dot(J2,D3)+dot(J3,D2)+dot(D2,D3))/4;
91     EcXIZETA(5)= EcXIZETA(5)+XIEH(EH,1)*XIEH(EH,3)*(dot(J2,D3)...
92         +dot(J3,D2)+dot(D2,D3))/4;
93 end

```

```

94
95 for JM=1:4
96     [J1,~,J3]=colonnejacobien(XYZ,XIJM(JM,:));
97     [D1,~,D3]=colonneDefconv(DSP,XIJM(JM,:));
98     Ec0(6)= Ec0(6)+(dot(J3,D1)+dot(J1,D3)+dot(D3,D1))/4;
99     Eczeta(6)= ...
100     Eczeta(6)+XIJM(JM,3)*(dot(J3,D1)+dot(J1,D3)+dot(D3,D1))/4;
101     EcETA(6)= ...
102     EcETA(6)+XIJM(JM,2)*(dot(J1,D3)+dot(J3,D1)+dot(D1,D3))/4;
103     EcETAZETA(6)= EcETAZETA(6)+XIJM(JM,2)*XIJM(JM,3)*(dot(J1,D3)+...
104     dot(J3,D1)+dot(D1,D3))/4;
105 end
end
end

```

```

1 function T=matriceT(XYZ,XI)
2 %
3 J=Jacobien(XYZ,XI);
4 J=inv(J);
5 J11=J(1,1);
6 J12=J(1,2);
7 J13=J(1,3);
8 J21=J(2,1);
9 J22=J(2,2);
10 J23=J(2,3);
11 J31=J(3,1);
12 J32=J(3,2);
13 J33=J(3,3);
14 %
15 T=[ J11*J11 J21*J21 J31*J31 J11*J21 J21*J31 ...
      J11*J31
16     J12*J12 J22*J22 J32*J32 J12*J22 J22*J32 ...
      J12*J32
17     J13*J13 J23*J23 J33*J33 J13*J23 J23*J33 ...
      J13*J33
18     2*J11*J12 2*J21*J22 2*J31*J32 J11*J22+J21*J12 J22*J31+J21*J32 ...
      J11*J32+J31*J12
19     2*J12*J13 2*J22*J23 2*J32*J33 J13*J22+J12*J23 J22*J33+J32*J23 ...
      J12*J33+J32*J13
20     2*J11*J13 2*J21*J23 2*J31*J33 J13*J21+J11*J23 J23*J31+J21*J33 ...
      J13*J31+J11*J33];

```

21 end

```
1 function [TXI,TETA,TZETA]=matricetxi(J0,A,B,C)
2 %
3 TXI=zeros(6,6);
4 TETA=zeros(6,6);
5 TZETA=zeros(6,6);
6 %
7 TXI(1,1)=2*J0(1,1)*A(1,1) ;
8 TXI(1,2)=2*J0(2,1)*A(2,1) ;
9 TXI(1,3)=2*J0(3,1)*A(3,1) ;
10 TXI(1,4)=J0(1,1)*A(2,1)+J0(2,1)*A(1,1);
11 TXI(1,5)=J0(2,1)*A(3,1)+J0(3,1)*A(2,1);
12 TXI(1,6)=J0(1,1)*A(3,1)+J0(3,1)*A(1,1);
13 %
14 TXI(2,1)=2*J0(1,2)*A(1,2) ;
15 TXI(2,2)=2*J0(2,2)*A(2,2) ;
16 TXI(2,3)=2*J0(3,2)*A(3,2) ;
17 TXI(2,4)=J0(1,2)*A(2,2)+J0(2,2)*A(1,2);
18 TXI(2,5)=J0(2,2)*A(3,2)+J0(3,2)*A(2,2);
19 TXI(2,6)=J0(1,2)*A(3,2)+J0(3,2)*A(1,2);
20 %
21 TXI(3,1)=2*J0(1,3)*A(1,3) ;
22 TXI(3,2)=2*J0(2,3)*A(2,3) ;
23 TXI(3,3)=2*J0(3,3)*A(3,3) ;
24 TXI(3,4)=J0(1,3)*A(2,3)+J0(2,3)*A(1,3);
25 TXI(3,5)=J0(2,3)*A(3,3)+J0(3,3)*A(2,3);
26 TXI(3,6)=J0(1,3)*A(3,3)+J0(3,3)*A(1,3);
27 %
28 TXI(4,1)=2*(J0(1,1)*A(1,2)+J0(1,2)*A(1,1));
29 TXI(4,2)=2*(J0(2,1)*A(2,2)+J0(2,2)*A(2,1));
30 TXI(4,3)=2*(J0(3,1)*A(3,2)+J0(3,2)*A(3,1));
31 TXI(4,4)=(J0(1,2)*A(2,1)+J0(2,1)*A(1,2))+(J0(1,1)*A(2,2)+J0(2,2)*A(1,1));
32 TXI(4,5)=(J0(2,2)*A(3,1)+J0(3,1)*A(2,2))+(J0(2,1)*A(3,2)+J0(3,2)*A(2,1));
33 TXI(4,6)=(J0(1,2)*A(3,1)+J0(3,1)*A(1,2))+(J0(1,1)*A(3,2)+J0(3,2)*A(1,1));
34 %
35 TXI(5,1)=2*(J0(1,2)*A(1,3)+J0(1,3)*A(1,2));
36 TXI(5,2)=2*(J0(2,2)*A(2,3)+J0(2,3)*A(2,2));
37 TXI(5,3)=2*(J0(3,2)*A(3,3)+J0(3,3)*A(3,2));
38 TXI(5,4)=(J0(1,3)*A(2,2)+J0(2,2)*A(1,3))+(J0(1,2)*A(2,3)+J0(2,3)*A(1,2));
39 TXI(5,5)=(J0(2,3)*A(3,2)+J0(3,2)*A(2,3))+(J0(2,2)*A(3,3)+J0(3,3)*A(2,2));
```

```

40 TXI(5,6)=(J0(1,3)*A(3,2)+J0(3,2)*A(1,3))+(J0(1,2)*A(3,3)+J0(3,3)*A(1,2));
41 %
42 TXI(6,1)=2*(J0(1,1)*A(1,3)+J0(1,3)*A(1,1));
43 TXI(6,2)=2*(J0(2,1)*A(2,3)+J0(2,3)*A(2,1));
44 TXI(6,3)=2*(J0(3,1)*A(3,3)+J0(3,3)*A(3,1));
45 TXI(6,4)=(J0(1,3)*A(2,1)+J0(2,1)*A(1,3))+(J0(1,1)*A(2,3)+J0(2,3)*A(1,1));
46 TXI(6,5)=(J0(2,3)*A(3,1)+J0(3,1)*A(2,3))+(J0(2,1)*A(3,3)+J0(3,3)*A(2,1));
47 TXI(6,6)=(J0(1,3)*A(3,1)+J0(3,1)*A(1,3))+(J0(1,1)*A(3,3)+J0(3,3)*A(1,1));
48 %
49 TETA(1,1)=2*J0(1,1)*B(1,1);
50 TETA(1,2)=2*J0(2,1)*B(2,1);
51 TETA(1,3)=2*J0(3,1)*B(3,1);
52 TETA(1,4)=J0(1,1)*B(2,1)+J0(2,1)*B(1,1);
53 TETA(1,5)=J0(2,1)*B(3,1)+J0(3,1)*B(2,1);
54 TETA(1,6)=J0(1,1)*B(3,1)+J0(3,1)*B(1,1);
55 %
56 TETA(2,1)=2*J0(1,2)*B(1,2);
57 TETA(2,2)=2*J0(2,2)*B(2,2);
58 TETA(2,3)=2*J0(3,2)*B(3,2);
59 TETA(2,4)=J0(1,2)*B(2,2)+J0(2,2)*B(1,2);
60 TETA(2,5)=J0(2,2)*B(3,2)+J0(3,2)*B(2,2);
61 TETA(2,6)=J0(1,2)*B(3,2)+J0(3,2)*B(1,2);
62 %
63 TETA(3,1)=2*J0(1,3)*B(1,3) ;
64 TETA(3,2)=2*J0(2,3)*B(2,3) ;
65 TETA(3,3)=2*J0(3,3)*B(3,3) ;
66 TETA(3,4)=J0(1,3)*B(2,3)+J0(2,3)*B(1,3);
67 TETA(3,5)=J0(2,3)*B(3,3)+J0(3,3)*B(2,3);
68 TETA(3,6)=J0(1,3)*B(3,3)+J0(3,3)*B(1,3);
69 %
70 TETA(4,1)=2*(J0(1,1)*B(1,2)+J0(1,2)*B(1,1));
71 TETA(4,2)=2*(J0(2,1)*B(2,2)+J0(2,2)*B(2,1));
72 TETA(4,3)=2*(J0(3,1)*B(3,2)+J0(3,2)*B(3,1));
73 TETA(4,4)=(J0(1,2)*B(2,1)+J0(2,1)*B(1,2))+(J0(1,1)*B(2,2)+J0(2,2)*B(1,1));
74 TETA(4,5)=(J0(2,2)*B(3,1)+J0(3,1)*B(2,2))+(J0(2,1)*B(3,2)+J0(3,2)*B(2,1));
75 TETA(4,6)=(J0(1,2)*B(3,1)+J0(3,1)*B(1,2))+(J0(1,1)*B(3,2)+J0(3,2)*B(1,1));
76 %
77 TETA(5,1)=2*(J0(1,2)*B(1,3)+J0(1,3)*B(1,2));
78 TETA(5,2)=2*(J0(2,2)*B(2,3)+J0(2,3)*B(2,2));
79 TETA(5,3)=2*(J0(3,2)*B(3,3)+J0(3,3)*B(3,2));
80 TETA(5,4)=(J0(1,3)*B(2,2)+J0(2,2)*B(1,3))+(J0(1,2)*B(2,3)+J0(2,3)*B(1,2));
81 TETA(5,5)=(J0(2,3)*B(3,2)+J0(3,2)*B(2,3))+(J0(2,2)*B(3,3)+J0(3,3)*B(2,2));
82 TETA(5,6)=(J0(1,3)*B(3,2)+J0(3,2)*B(1,3))+(J0(1,2)*B(3,3)+J0(3,3)*B(1,2));

```

```

83 %
84 TETA(6,1)=2*(J0(1,1)*B(1,3)+J0(1,3)*B(1,1));
85 TETA(6,2)=2*(J0(2,1)*B(2,3)+J0(2,3)*B(2,1));
86 TETA(6,3)=2*(J0(3,1)*B(3,3)+J0(3,3)*B(3,1));
87 TETA(6,4)=(J0(1,3)*B(2,1)+J0(2,1)*B(1,3))+(J0(1,1)*B(2,3)+J0(2,3)*B(1,1));
88 TETA(6,5)=(J0(2,3)*B(3,1)+J0(3,1)*B(2,3))+(J0(2,1)*B(3,3)+J0(3,3)*B(2,1));
89 TETA(6,6)=(J0(1,3)*B(3,1)+J0(3,1)*B(1,3))+(J0(1,1)*B(3,3)+J0(3,3)*B(1,1));
90 %
91 TZETA(1,1)=2*J0(1,1)*C(1,1);
92 TZETA(1,2)=2*J0(2,1)*C(2,1);
93 TZETA(1,3)=2*J0(3,1)*C(3,1);
94 TZETA(1,4)=J0(1,1)*C(2,1)+J0(2,1)*C(1,1);
95 TZETA(1,5)=J0(2,1)*C(3,1)+J0(3,1)*C(2,1);
96 TZETA(1,6)=J0(1,1)*C(3,1)+J0(3,1)*C(1,1);
97 %
98 TZETA(2,1)=2*J0(1,2)*C(1,2);
99 TZETA(2,2)=2*J0(2,2)*C(2,2);
100 TZETA(2,3)=2*J0(3,2)*C(3,2);
101 TZETA(2,4)=J0(1,2)*C(2,2)+J0(2,2)*C(1,2);
102 TZETA(2,5)=J0(2,2)*C(3,2)+J0(3,2)*C(2,2);
103 TZETA(2,6)=J0(1,2)*C(3,2)+J0(3,2)*C(1,2);
104 %
105 TZETA(3,1)=2*J0(1,3)*C(1,3);
106 TZETA(3,2)=2*J0(2,3)*C(2,3);
107 TZETA(3,3)=2*J0(3,3)*C(3,3);
108 TZETA(3,4)=J0(1,3)*C(2,3)+J0(2,3)*C(1,3);
109 TZETA(3,5)=J0(2,3)*C(3,3)+J0(3,3)*C(2,3);
110 TZETA(3,6)=J0(1,3)*C(3,3)+J0(3,3)*C(1,3);
111 %
112 TZETA(4,1)=2*(J0(1,1)*C(1,2)+J0(1,2)*C(1,1));
113 TZETA(4,2)=2*(J0(2,1)*C(2,2)+J0(2,2)*C(2,1));
114 TZETA(4,3)=2*(J0(3,1)*C(3,2)+J0(3,2)*C(3,1));
115 TZETA(4,4)=(J0(1,2)*C(2,1)+J0(2,1)*C(1,2))+(J0(1,1)*C(2,2)+J0(2,2)*C(1,1));
116 TZETA(4,5)=(J0(2,2)*C(3,1)+J0(3,1)*C(2,2))+(J0(2,1)*C(3,2)+J0(3,2)*C(2,1));
117 TZETA(4,6)=(J0(1,2)*C(3,1)+J0(3,1)*C(1,2))+(J0(1,1)*C(3,2)+J0(3,2)*C(1,1));
118 %
119 TZETA(5,1)=2*(J0(1,2)*C(1,3)+J0(1,3)*C(1,2));
120 TZETA(5,2)=2*(J0(2,2)*C(2,3)+J0(2,3)*C(2,2));
121 TZETA(5,3)=2*(J0(3,2)*C(3,3)+J0(3,3)*C(3,2));
122 TZETA(5,4)=(J0(1,3)*C(2,2)+J0(2,2)*C(1,3))+(J0(1,2)*C(2,3)+J0(2,3)*C(1,2));
123 TZETA(5,5)=(J0(2,3)*C(3,2)+J0(3,2)*C(2,3))+(J0(2,2)*C(3,3)+J0(3,3)*C(2,2));
124 TZETA(5,6)=(J0(1,3)*C(3,2)+J0(3,2)*C(1,3))+(J0(1,2)*C(3,3)+J0(3,3)*C(1,2));
125 %

```

```

126 TZETA(6,1)=2*(J0(1,1)*C(1,3)+J0(1,3)*C(1,1));
127 TZETA(6,2)=2*(J0(2,1)*C(2,3)+J0(2,3)*C(2,1));
128 TZETA(6,3)=2*(J0(3,1)*C(3,3)+J0(3,3)*C(3,1));
129 TZETA(6,4)=(J0(1,3)*C(2,1)+J0(2,1)*C(1,3))+(J0(1,1)*C(2,3)+J0(2,3)*C(1,1));
130 TZETA(6,5)=(J0(2,3)*C(3,1)+J0(3,1)*C(2,3))+(J0(2,1)*C(3,3)+J0(3,3)*C(2,1));
131 TZETA(6,6)=(J0(1,3)*C(3,1)+J0(3,1)*C(1,3))+(J0(1,1)*C(3,3)+J0(3,3)*C(1,1));
132 end

```

```

1
2 function [T0,TXI,TETA,TZETA]=matriceTdecomp(XYZ)
3 %
4 X=XYZ(:,1);
5 Y=XYZ(:,2);
6 Z=XYZ(:,3);
7 %
8 g1=[-1 1 1 -1 -1 1 1 -1]/8;
9 g2=[-1 -1 1 1 -1 -1 1 1]/8;
10 g3=[-1 -1 -1 -1 1 1 1 1]/8;
11 %
12 h1=[1 -1 1 -1 1 -1 1 -1]/8;
13 h2=[1 1 -1 -1 -1 -1 1 1]/8;
14 h3=[1 -1 -1 1 -1 1 1 -1]/8;
15 %
16 J0 =[dot(X,g1) dot(X,g2) dot(X,g3);
17      dot(Y,g1) dot(Y,g2) dot(Y,g3);
18      dot(Z,g1) dot(Z,g2) dot(Z,g3)];
19 J0=inv(J0);
20 %
21 JXI=[0 dot(X,h1) dot(X,h3);
22      0 dot(Y,h1) dot(Y,h3);
23      0 dot(Z,h1) dot(Z,h3)];
24 %
25 JETA=[dot(X,h1) 0 dot(X,h2);
26      dot(Y,h1) 0 dot(Y,h2);
27      dot(Z,h1) 0 dot(Z,h2)];
28 %
29 JZETA=[dot(X,h3) dot(X,h2) 0;
30      dot(Y,h3) dot(Y,h2) 0;
31      dot(Z,h3) dot(Z,h2) 0];
32 %
33 A=-J0*JXI*J0;

```

F. Matlab source code

```
34 B=-J0*JETA*J0;
35 C=-J0*JZETA*J0;
36 %%
37 T0=matriceT(XYZ,[0 0 0]);
38 [TXI,TETA,TZETA]=matricetxi(J0,A,B,C);
39
40 end
```

```
1 function [EH1,EH2,EH12,EH23,EH13]=deformEcartStab(DSP,XYZ)
2
3 %[Ec0,Eczeta,Eczz,EcXI,EcETA,EcETAZETA,EcXIZETA]=deformatEc(DSP,XYZ);
4 [Ec0,Eczeta,Eczz,EcXI,EcETA,EcXIETA,EcETAZETA,EcXIZETA]=deformatEc(DSP,XYZ);
5 [T0,TXI,TETA,TZETA]=matriceTdecomp(XYZ);
6
7 %% DEFORMATIONS DE STABILISATION
8 EH1=T0*EcXI+TXI*Ec0;
9 EH2=T0*EcETA+TETA*Ec0;
10 EH23=T0*EcETAZETA+TETA*Eczeta+TZETA*EcETA;
11 EH13=T0*EcXIZETA+TXI*Eczeta+TZETA*EcXI;
12 EH12=T0*EcXIETA+TXI*EcETA+TETA*EcXI;
13 end
```

```
1 function B9=calculB9(XYZ,XI)
2 % Calcul de la composante du neuvieme noeuds
3 [T0,~,~,~]=matriceTdecomp(XYZ);
4 Bc9=[0 0 -2*XI(3) 0 0 0]';
5 B9=T0*Bc9;
6 end
```

```
1 function Kg=rigigeo(S,XYZ,XI)
2
3 zeta=XI(3);
4 %CoordonnÃ'es initiale en entrÃ'e
5 % Cette matrice Kg est celle sur un points de Gauss
6 [T0,TXI,TETA,TZETA]=matriceTdecomp(XYZ);
7 %function ...
```

```

[Bc0,BcZETA,BcZZ,BcXI,BcETA,BcETAZETA,BcXIZETA]=matriceBc(XYZ)
8 %%
9 g1=[-1 1 1 -1 -1 1 1 -1]/8;
10 g2=[-1 -1 1 1 -1 -1 1 1]/8;
11 g3=[-1 -1 -1 -1 1 1 1 1]/8;
12 %
13
14 h1=[1 -1 1 -1 1 -1 1 -1]/8;
15 h2=[1 1 -1 -1 -1 -1 1 1]/8;
16 h3=[1 -1 -1 1 -1 1 1 -1]/8;
17 h4=[-1 1 -1 1 1 -1 1 -1]/8;
18 %
19 %% TYING POINTS
20 % A Ã D
21 XAD=[-1 -1 0
22      1 -1 0
23      1 1 0
24      -1 1 0];
25 % E TO H
26 XEH=[-1 0 -1
27      1 0 -1
28      1 0 1
29      -1 0 1];
30 % J TO M
31 XJM=[0 -1 -1
32      0 1 -1
33      0 1 1
34      0 -1 1];
35 %%
36 Kg=zeros(24,24);
37 for I=1:8
38     for J=1:8
39         LIN=3*(I-1)+1:3*(I-1)+3;
40         COL=3*(J-1)+1:3*(J-1)+3;
41
42         aux033=0;
43         aux023=0;
44         aux013=0;
45         auxZ23=0;
46         auxZ13=0;
47     %% B0
48         for AD=1:4
49             DSF = DSHAPE(XAD(AD, :));

```

```

50     aux033=aux033+DSF(3,I)*DSF(3,J)/4;
51 %
52     DSF = DSHAPE(XEH(AD, :));
53     aux023=aux023+(DSF(2,I)*DSF(3,J)+DSF(3,I)*DSF(2,J))/4;
54     ...
55     auxZ23=auxZ23+XEH(AD,3)*(DSF(2,I)*DSF(3,J)+DSF(3,I)*DSF(2,J))/4;
56 %
57     DSF = DSHAPE(XJM(AD, :));
58     aux013=aux013+(DSF(1,I)*DSF(3,J)+DSF(3,I)*DSF(1,J))/4;
59     ...
60     auxZ13=auxZ13+XJM(AD,3)*(DSF(1,I)*DSF(3,J)+DSF(3,I)*DSF(1,J))/4;
61 end
62 %mm
63 %aux033
64     GIJ0=[g1(I)*g1(J);
65           g2(I)*g2(J);
66           aux033;
67           g1(I)*g2(J)+g2(I)*g1(J);
68           aux023;
69           aux013];
70 %% BZETA
71     AUXZ12=h2(I)*g1(J)+h3(I)*g2(J)+g2(I)*h3(J)+g1(I)*h2(J) ;
72     GIJZ=[h3(I)*g1(J)+g1(I)*h3(J);
73           h2(I)*g2(J)+g2(I)*h2(J);
74           0 ;
75           AUXZ12;
76           auxZ23;
77           auxZ13];
78 %% BZZ
79     GIJZZ=[h3(I)*h3(J);
80            h2(I)*h2(J);
81            0 ;
82            h2(I)*h3(J)+h3(I)*h2(J);
83            0 ;
84            0 ];
85 %%
86     ...
87     alpha=T0*GIJ0+zeta*(T0*GIJZ+TZETA*GIJ0)+zeta*zeta*(T0*GIJZZ+TZETA*GIJZ);
88     Kg(LIN,COL)=dot(alpha,S)*eye(3);
89 end
end
end

```

```

1 function [SH1,SH2,SH12,SH23,SH13]=contraintestab(DSP,XYZ,C)
2 % CALCUL DES CONTRAINTES DE STABILISATION
3 %
4 %
5 %[EH1,EH2,EH23,EH13]=deformEcartStab(DSP,XYZ);
6 [EH1,EH2,EH12,EH23,EH13]=deformEcartStab(DSP,XYZ);
7 % On considère la partie déviatorique de ces deformations
8 %
9 I=[1 1 1 0 0 0]';
10 EH1=EH1-(EH1(1)+EH1(2)+EH1(3))*I/3;
11 EH2=EH2-(EH2(1)+EH2(2)+EH2(3))*I/3;
12 EH13=EH13-(EH13(1)+EH13(2)+EH13(3))*I/3;
13 EH23=EH23-(EH23(1)+EH23(2)+EH23(3))*I/3;
14 EH12=EH12-(EH12(1)+EH12(2)+EH12(3))*I/3;
15 %
16
17 SH1=C*EH1;
18 SH2=C*EH2;
19 SH13=C*EH13;
20 SH23=C*EH23;
21 SH12=C*EH12;
22 end

```

```

1 function ...
2     Kmstab=rigistab(BXIdev,BETAdev,BXIETAdev,BETAZETAdev,BXIZETAdev,C)
3
4 Kmstab=8*(BXIdev'*C*BXIdev+BETAdev'*C*BETAdev)/3+...
5     8*(BETAZETAdev'*C*BETAZETAdev+...
6     BXIETAdev'*C*BXIETAdev+...
7     BXIZETAdev'*C*BXIZETAdev)/9.0;
8 end

```

```

1 function Kgstab=rigigeostab1(XYZ,SH1,SH2,SH23,SH13,SH12)
2
3 %Coordonnées initiales en entrée
4 [T0,TXI,TETA,TZETA]=matriceTdecomp(XYZ);
5

```

```

6 %function ...
   [Bc0,BcZETA,BcZZ,BcXI,BcETA,BcETAZETA,BcXIZETA]=matriceBc(XYZ)
7 %%
8 g1=[-1 1 1 -1 -1 1 1 -1]/8;
9 g2=[-1 -1 1 1 -1 -1 1 1]/8;
10 g3=[-1 -1 -1 -1 1 1 1 1]/8;
11 %
12 h1=[1 -1 1 -1 1 -1 1 -1]/8;
13 h2=[1 1 -1 -1 -1 -1 1 1]/8;
14 h3=[1 -1 -1 1 -1 1 1 -1]/8;
15 h4=[-1 1 -1 1 1 -1 1 -1]/8;
16 IDEN=[1 1 1 0 0 0]';
17 %
18 %% TYING POINTS
19 % A Ã D
20 XAD=[-1 -1 0
21       1 -1 0
22       1 1 0
23       -1 1 0];
24 % E TO H
25 XEH=[-1 0 -1
26       1 0 -1
27       1 0 1
28       -1 0 1];
29 % J TO M
30 XJM=[0 -1 -1
31       0 1 -1
32       0 1 1
33       0 -1 1];
34 %%
35 Kgstab=zeros(24,24);
36 for I=1:8
37     for J=1:8
38         LIN=3*(I-1)+1:3*(I-1)+3;
39         COL=3*(J-1)+1:3*(J-1)+3;
40         %% B0
41         aux33=0;
42         aux23=0;
43         aux13=0;
44         for AD=1:4
45             DSF = DSHAPE(XAD(AD, :));
46             aux33=aux33+DSF(3, I)*DSF(3, J);
47         end

```

```

48     for EH=1:4
49         DSF = DSHAPE(XEH(EH, :));
50         aux23=aux23+(DSF(2,I)*DSF(3,J)+DSF(3,I)*DSF(2,J))/4;
51     end
52     for JM=1:4
53         DSF = DSHAPE(XJM(JM, :));
54         aux13=aux13+(DSF(1,I)*DSF(3,J)+DSF(3,I)*DSF(1,J))/4;
55     end
56
57         Gc0=[g1(I)*g1(J);
58             g2(I)*g2(J);
59             aux33;
60             g1(I)*g2(J)+g2(I)*g1(J);
61             aux23;
62             aux13];
63 %% BZETA
64 aux23=0;
65 aux13=0;
66 for EH=1:4
67     DSF = DSHAPE(XEH(EH, :));
68     aux23=aux23+XEH(EH, 3)*(DSF(2,I)*DSF(3,J)+DSF(3,I)*DSF(2,J))/4;
69 end
70 for JM=1:4
71     DSF = DSHAPE(XJM(JM, :));
72     aux13=aux13+XJM(JM, 3)*(DSF(1,I)*DSF(3,J)+DSF(3,I)*DSF(1,J))/4;
73 end
74 AUX12=h2(I)*g1(J)+h3(I)*g2(J)+g2(I)*h3(J)+g1(I)*h2(J) ;
75     GcZETA=[ h3(I)*g1(J)+g1(I)*h3(J);
76             h2(I)*g2(J)+g2(I)*h2(J);
77             0 ;
78             AUX12;
79             aux23;
80             aux13];
81 %% BcXI
82 AUX33=0;
83 aux23=0;
84 for AD=1:4
85     DSF = DSHAPE(XAD(AD, :));
86     AUX33=AUX33+XAD(AD, 1)*(DSF(3,I)*DSF(3,J))/4;
87 end
88 for EH=1:4
89     DSF = DSHAPE(XEH(EH, :));
90     aux23=aux23+XEH(EH, 1)*(DSF(2,I)*DSF(3,J)+DSF(3,I)*DSF(2,J))/4;

```

```

91     end
92         GcXI=[ 0 ;
93             h1(I)*g2(J)+g2(I)*h1(1);
94             AUX33;
95             h1(I)*g1(J)+g1(I)*h1(J);
96             aux23;
97             0];
98
99 %% ETA
100     AUX33=0;
101     aux13=0;
102     for AD=1:4
103         DSF = DSHAPE(XAD(AD, :));
104         AUX33=AUX33+XAD(AD, 2)*(DSF(3, I)*DSF(3, J))/4;
105     end
106     for JM=1:4
107         DSF = DSHAPE(XJM(JM, :));
108         aux13=aux13+XJM(JM, 2)*(DSF(1, I)*DSF(3, J)+DSF(3, I)*DSF(1, J))/4;
109     end
110     GcETA=[ h1(I)*g1(J)+g1(I)*h1(J);
111            0 ;
112            AUX33;
113            h1(I)*g2(J)+g2(I)*h1(J);
114            0 ;
115            aux13];
116 %% XIETA
117     for AD=1:4
118         DSF = DSHAPE(XAD(AD, :));
119         AUX33=AUX33+XAD(AD, 2)*XAD(AD, 1)*(DSF(3, I)*DSF(3, J))/4;
120     end
121     GcXIETA=[ 0;
122              0 ;
123              AUX33 ;
124              2*h1(I)*h1(J);
125              0 ;
126              0];
127 %% ETAZETA
128     AUX11=h4(I)*g1(J)+h3(I)*h1(J)+h1(I)*h3(J)+g1(I)*h4(J);
129     AUX12=h4(I)*g2(J)+h2(I)*h1(J)+h1(I)*h2(J)+g2(I)*h4(J);
130     AUX13=0;
131     for JM=1:4
132         DSF = DSHAPE(XJM(JM, :));
133         ...

```

```

134     AUX13=AUX13+XJM(JM,2)*XJM(JM,3)*(DSF(1,I)*DSF(3,J)+DSF(3,I)*DSF(1,J))/4;
135     end
136         GcETAZETA=[ AUX11;
137                   0   ;
138                   0   ;
139                   AUX12;
140                   0   ;
141                   AUX13];
142 %% XIZETA
143     AUX22=h4(I)*g2(J)+h2(I)*h1(J)+h1(I)*h2(J)+g2(I)*h4(J);
144     AUX12=h4(I)*g1(J)+h3(I)*h1(J)+h1(I)*h3(J)+g1(I)*h4(J);
145     AUX23=0;
146     for EH=1:4
147         DSF = DSHAPE(XEH(EH,:));
148         ...
149     AUX23=AUX23+XEH(EH,1)*XEH(EH,3)*(DSF(2,I)*DSF(3,J)+DSF(3,I)*DSF(2,J))/4;
150     end
151         GcXIZETA=[ 0 ;
152                   AUX22;
153                   0   ;
154                   AUX12;
155                   AUX23;
156                   0   ];
157 %%
158 % CALCUL DES DEVIATORIQUES
159 % GXIc=GXIc-(GXIc(1)+GXIc(2)+GXIc(3))*IDEN/3;
160 % GETAc=GETAc-(GETAc(1)+GETAc(2)+GETAc(3))*IDEN/3;
161 % ...
162 % GETAZETAc=GETAZETAc-(GETAZETAc(1)+GETAZETAc(2)+GETAZETAc(3))*IDEN/3;
163 % GXIZETAc=GXIZETAc-(GXIZETAc(1)+GXIZETAc(2)+GXIZETAc(3))*IDEN/3;
164 % GXIETAc=GXIETAc-(GXIETAc(1)+GXIETAc(2)+GXIETAc(3))*IDEN/3;
165 %
166 GXI=T0*GcXI+TXI*Gc0;
167 GETA=T0*GcETA+TETA*Gc0;
168 GETAZETA=T0*GcETAZETA+TETA*GcZETA+TZETA*GcETA;
169 GXIZETA=T0*GcXIZETA+TXI*GcZETA+TZETA*GcXI;
170 GXIETA=T0*GcXIETA+TXI*GcETA+TETA*GcXI;
171 GXI=GXI-(GXI(1)+GXI(2)+GXI(3))*IDEN/3;
172 GETA=GETA-(GETA(1)+GETA(2)+GETA(3))*IDEN/3;
173 GETAZETA=GETAZETA-(GETAZETA(1)+GETAZETA(2)+GETAZETA(3))*IDEN/3;
174 GXIZETA=GXIZETA-(GXIZETA(1)+GXIZETA(2)+GXIZETA(3))*IDEN/3;
175 GXIETA=GXIETA-(GXIETA(1)+GXIETA(2)+GXIETA(3))*IDEN/3;

```

```

174 % %
175 CONST=SH1 '*GXI*(8/3)+SH2 '*GETA*(8/3)+(SH23 '*GETAZETA+SH13 '*GXIZETA+SH12 '*GXIETA)*(8/
176 Kgstab(LIN,COL)=CONST*eye(3);
177     end
178 end
179 end

```

F.2 SB7 Matlab Code

```

1 %%
2 function [aire2D,h,volume,detJ,Bm0,Bb0,...
3         Bc0,Bc1,Bc2,Bp0,Bsz,Bgu,Bgv,Bgw]=MatricesBprisme(XYZ)
4
5 %% Repere local
6 Xi=XYZ(1,1);
7 Yi=XYZ(1,2);
8 Zi=XYZ(1,3);
9 Xj=XYZ(2,1);
10 Yj=XYZ(2,2);
11 Zj=XYZ(2,3);
12 Xk=XYZ(3,1);
13 Yk=XYZ(3,2);
14 Zk=XYZ(3,3);
15 Xl=XYZ(4,1);
16 Yl=XYZ(4,2);
17 Zl=XYZ(4,3);
18 Xm=XYZ(5,1);
19 Ym=XYZ(5,2);
20 Zm=XYZ(5,3);
21 Xn=XYZ(6,1);
22 Yn=XYZ(6,2);
23 Zn=XYZ(6,3);
24 %
25 X12=(Xj+Xm-Xi-Xl)/2.0;
26 Y12=(Yj+Ym-Yi-Yl)/2.0;
27 Z12=(Zj+Zm-Zi-Zl)/2.0;
28 X13=(Xk+Xn-Xi-Xl)/2.0;
29 Y13=(Yk+Yn-Yi-Yl)/2.0;
30 Z13=(Zk+Zn-Zi-Zl)/2.0;

```

```

31
32 xX=X12;
33 xY=Y12;
34 xZ=Z12;
35 Nx=sqrt(xX*xX+xY*Y12+xZ*Z12);
36 xX=xX/Nx;
37 xY=xY/Nx;
38 xZ=xZ/Nx;
39
40 zX=Y12*Z13-Z12*Y13;
41 zY=Z12*X13-X12*Z13;
42 zZ=X12*Y13-Y12*X13;
43 Nz=sqrt(zX*zX+zY*Y13+zZ*Z13);
44 zX=zX/Nz;
45 zY=zY/Nz;
46 zZ=zZ/Nz;
47
48 yX=zY*xZ-zZ*xY;
49 yY=zZ*xX-zX*xZ;
50 yZ=zX*xY-zY*xX;
51 Ny=sqrt(yX*yX+yY*Y13+yZ*Z13);
52 yX=yX/Ny;
53 yY=yY/Ny;
54 yZ=yZ/Ny;
55
56 %% CoordonnÃ'es locales
57
58     xi=xX*Xi+xY*Yi+xZ*Zi;
59     yi=yX*Xi+yY*Yi+yZ*Zi;
60     zi=zX*Xi+zY*Yi+zZ*Zi;
61     xj=xX*Xj+xY*Yj+xZ*Zj;
62     yj=yX*Xj+yY*Yj+yZ*Zj;
63     zj=zX*Xj+zY*Yj+zZ*Zj;
64     xk=xX*Xk+xY*Yk+xZ*Zk;
65     yk=yX*Xk+yY*Yk+yZ*Zk;
66     zk=zX*Xk+zY*Yk+zZ*Zk;
67
68     xl=xX*Xl+xY*Yl+xZ*Zl;
69     yl=yX*Xl+yY*Yl+yZ*Zl;
70     zl=zX*Xl+zY*Yl+zZ*Zl;
71     xm=xX*Xm+xY*Ym+xZ*Zm;
72     ym=yX*Xm+yY*Ym+yZ*Zm;
73     zm=zX*Xm+zY*Ym+zZ*Zm;

```

```

74     xn=xX*Xn+xY*Yn+xZ*Zn;
75     yn=yX*Xn+yY*Yn+yZ*Zn;
76     zn=zX*Xn+zY*Yn+zZ*Zn;
77
78     x1=(xi+xl)/2.0;
79     y1=(yi+yl)/2.0;
80
81     x2=(xj+xm)/2.0;
82     y2=(yj+ym)/2.0;
83
84     x3=(xk+xn)/2.0;
85     y3=(yk+yn)/2.0;
86
87     x3=x3-x1;
88     y3=y3-y1;
89
90     x2=x2-x1;
91     y2=0.0;
92
93     x1=0.0;
94     y1=0.0;
95
96     aire2D=abs(x2*y3-x3*y2)/2.0;
97     h=abs(zl+zm+zn-zi-zj-zk)/3.0;
98 %% volume
99 volume=aire2D*h;
100
101 %% Bb0=matriceflexion(XYZ,xX,xY,xZ,yX,yY,yZ,zX,zY,zZ,aire2D);
102 [Bt,Bw]=matricesBtBw(x1,y1,x2,y2,x3,y3,aire2D);
103 %%
104 %donne les trois rotations aux milieux des cotifs du DKT6 en ...
    fonction des
105 %translations dans le repire global des 6 noeuds
106 Tt=TransformerTransRot(XYZ);
107
108 %%
109 Bbt=Bt*Tt;
110 Bbw=zeros(3,18);
111 for p=1:3
112     Bbw(p,1)=0.5*Bw(p,1)*zX;
113     Bbw(p,2)=0.5*Bw(p,1)*zY;
114     Bbw(p,3)=0.5*Bw(p,1)*zZ;
115     Bbw(p,10)=Bbw(p,1);

```

```

116 Bbw(p,11)=Bbw(p,2);
117 Bbw(p,12)=Bbw(p,3);
118 Bbw(p,4)=0.5*Bw(p,2)*zX;
119 Bbw(p,5)=0.5*Bw(p,2)*zY;
120 Bbw(p,6)=0.5*Bw(p,2)*zZ;
121 Bbw(p,13)=Bbw(p,4);
122 Bbw(p,14)=Bbw(p,5);
123 Bbw(p,15)=Bbw(p,6);
124 Bbw(p,7)=0.5*Bw(p,3)*zX;
125 Bbw(p,8)=0.5*Bw(p,3)*zY;
126 Bbw(p,9)=0.5*Bw(p,3)*zZ;
127 Bbw(p,16)=Bbw(p,7);
128 Bbw(p,17)=Bbw(p,8);
129 Bbw(p,18)=Bbw(p,9);
130 end
131
132 Bb0=Bbt+Bbw;
133
134 %% Jacobien en ksi=1/3, eta=1/3, zeta=0
135 matJ(1,1)=(-xi+xj -xl+xm )/2.0;
136 matJ(1,2)=(-yi+yj -yl+ym )/2.0;
137 matJ(1,3)=(-zi+zj -zl+zm )/2.0;
138 matJ(2,1)=(-xi +xk-xl +xn)/2.0;
139 matJ(2,2)=(-yi +yk-yl +yn)/2.0;
140 matJ(2,3)=(-zi +zk-zl +zn)/2.0;
141 matJ(3,1)=(-xi-xj-xk+xl+xm+xn)/6.0;
142 matJ(3,2)=(-yi-yj-yk+yl+ym+yn)/6.0;
143 matJ(3,3)=(-zi-zj-zk+zl+zm+zn)/6.0;
144
145 %%determinant de J
146 detJ= (matJ(2,2)*matJ(3,3)-matJ(3,2)*matJ(2,3))*matJ(1,1);
147 detJ=detJ-(matJ(1,2)*matJ(3,3)-matJ(3,2)*matJ(1,3))*matJ(2,1);
148 detJ=detJ+(matJ(1,2)*matJ(2,3)-matJ(2,2)*matJ(1,3))*matJ(3,1);
149
150 invJls3=inv(matJ);
151 %% [bx,by,bz]=MatricesBxByBzauCentre(invJ);
152 b=1.0/2.0;
153 t=1.0/2.0;
154 u=1.0/6.0;
155 bx=zeros(1,6);
156 by=zeros(1,6);
157 bz=zeros(1,6);
158 %%<N,ksi> ; <N,eta> <N,zeta>

```

```

159   Bksi(1,1)=-b;
160   Bksi(2,1)=-b;
161   Bksi(3,1)=-u;
162   Bksi(1,2)= b;
163   Bksi(2,2)=0.0;
164   Bksi(3,2)=-u;
165   Bksi(1,3)=0.0;
166   Bksi(2,3)= b;
167   Bksi(3,3)=-u;
168   Bksi(1,4)=-t;
169   Bksi(2,4)=-t;
170   Bksi(3,4)= u;
171   Bksi(1,5)= t;
172   Bksi(2,5)=0.0;
173   Bksi(3,5)= u;
174   Bksi(1,6)=0.0;
175   Bksi(2,6)= t;
176   Bksi(3,6)= u;
177
178   %%ligne 1:<N,x>; ligne 2:<N,y>; ligne 3:<N,z> – en xi=1/3, ...
    eta=1/3, zeta=0 –
179   for j=1:6
180       bx(j)=invJ1s3(1,1)*Bksi(1,j)+invJ1s3(1,2)*Bksi(2,j)...
181           +invJ1s3(1,3)*Bksi(3,j);
182       by(j)=invJ1s3(2,1)*Bksi(1,j)+invJ1s3(2,2)*Bksi(2,j)...
183           +invJ1s3(2,3)*Bksi(3,j);
184       bz(j)=invJ1s3(3,1)*Bksi(1,j)+invJ1s3(3,2)*Bksi(2,j)...
185           +invJ1s3(3,3)*Bksi(3,j);
186   end
187   %% pour rigidite gÃľometrique
188   %%matrice B gifiomifitrique
189   [Bgu,Bgv,Bgw]=MatricesBg(xX,xY,xZ,yX,yY,yZ,zX,zY,zZ,bx,by,bz);
190   %% [Bm0]=MatriceBm0(xX,xY,xZ,yX,yY,yZ,bx,by);
191   Bm(1,1 )=bx(1)*xX; Bm(2,1 )=by(1)*yX; Bm(3,1 )=bx(1)*yX+by(1)*xX;
192   Bm(1,2 )=bx(1)*xY; Bm(2,2 )=by(1)*yY; Bm(3,2 )=bx(1)*yY+by(1)*xY;
193   Bm(1,3 )=bx(1)*xZ; Bm(2,3 )=by(1)*yZ; Bm(3,3 )=bx(1)*yZ+by(1)*xZ;
194   Bm(1,4 )=bx(2)*xX; Bm(2,4 )=by(2)*yX; Bm(3,4 )=bx(2)*yX+by(2)*xX;
195   Bm(1,5 )=bx(2)*xY; Bm(2,5 )=by(2)*yY; Bm(3,5 )=bx(2)*yY+by(2)*xY;
196   Bm(1,6 )=bx(2)*xZ; Bm(2,6 )=by(2)*yZ; Bm(3,6 )=bx(2)*yZ+by(2)*xZ;
197   Bm(1,7 )=bx(3)*xX; Bm(2,7 )=by(3)*yX; Bm(3,7 )=bx(3)*yX+by(3)*xX;
198   Bm(1,8 )=bx(3)*xY; Bm(2,8 )=by(3)*yY; Bm(3,8 )=bx(3)*yY+by(3)*xY;
199   Bm(1,9 )=bx(3)*xZ; Bm(2,9 )=by(3)*yZ; Bm(3,9 )=bx(3)*yZ+by(3)*xZ;
200   Bm(1,10)=bx(4)*xX; Bm(2,10)=by(4)*yX; Bm(3,10)=bx(4)*yX+by(4)*xX;

```

```

201 Bm(1,11)=bx(4)*xY; Bm(2,11)=by(4)*yY; Bm(3,11)=bx(4)*yY+by(4)*xY;
202 Bm(1,12)=bx(4)*xZ; Bm(2,12)=by(4)*yZ; Bm(3,12)=bx(4)*yZ+by(4)*xZ;
203 Bm(1,13)=bx(5)*xX; Bm(2,13)=by(5)*yX; Bm(3,13)=bx(5)*yX+by(5)*xX;
204 Bm(1,14)=bx(5)*xY; Bm(2,14)=by(5)*yY; Bm(3,14)=bx(5)*yY+by(5)*xY;
205 Bm(1,15)=bx(5)*xZ; Bm(2,15)=by(5)*yZ; Bm(3,15)=bx(5)*yZ+by(5)*xZ;
206 Bm(1,16)=bx(6)*xX; Bm(2,16)=by(6)*yX; Bm(3,16)=bx(6)*yX+by(6)*xX;
207 Bm(1,17)=bx(6)*xY; Bm(2,17)=by(6)*yY; Bm(3,17)=bx(6)*yY+by(6)*xY;
208 Bm(1,18)=bx(6)*xZ; Bm(2,18)=by(6)*yZ; Bm(3,18)=bx(6)*yZ+by(6)*xZ;
209 Bm0=Bm;
210 %% parail pour le pincement
211 %% [Bp0]=MatriceBp0(zX,zY,zZ,bz);
212     Bp0(1 )=bz(1)*zX;
213     Bp0(2 )=bz(1)*zY;
214     Bp0(3 )=bz(1)*zZ;
215     Bp0(4 )=bz(2)*zX;
216     Bp0(5 )=bz(2)*zY;
217     Bp0(6 )=bz(2)*zZ;
218     Bp0(7 )=bz(3)*zX;
219     Bp0(8 )=bz(3)*zY;
220     Bp0(9 )=bz(3)*zZ;
221     Bp0(10)=bz(4)*zX;
222     Bp0(11)=bz(4)*zY;
223     Bp0(12)=bz(4)*zZ;
224     Bp0(13)=bz(5)*zX;
225     Bp0(14)=bz(5)*zY;
226     Bp0(15)=bz(5)*zZ;
227     Bp0(16)=bz(6)*zX;
228     Bp0(17)=bz(6)*zY;
229     Bp0(18)=bz(6)*zZ;
230
231 %% [Bc0,Bc1,Bc2]=MatriceBc0(...)
232 %
233     h1=sqrt((xl-xi)*(xl-xi)+(yl-yi)*(yl-yi)+(zl-zi)*(zl-zi));
234     h2=sqrt((xm-xj)*(xm-xj)+(ym-yj)*(ym-yj)+(zm-zj)*(zm-zj));
235     h3=sqrt((xn-xk)*(xn-xk)+(yn-yk)*(yn-yk)+(zn-zk)*(zn-zk));
236     Bc0=zeros(2,18);
237     Bc1=zeros(2,18);
238     Bc2=zeros(2,18);
239 %% Jacobien
240 %%J en ksi=0, eta=0, zeta=0
241     matJ(1,1)=(-xi+xj   -xl+xm   )/2.0;
242     matJ(1,2)=(-yi+yj   -yl+ym   )/2.0;
243     matJ(1,3)=(-zi+zj   -zl+zm   )/2.0;

```

```

244     matJ(2,1)=(-xi   +xk-xl   +xn)/2.0;
245     matJ(2,2)=(-yi   +yk-yl   +yn)/2.0;
246     matJ(2,3)=(-zi   +zk-zl   +zn)/2.0;
247     matJ(3,1)=(-xi   +xl       )/2.0;
248     matJ(3,2)=(-yi   +yl       )/2.0;
249     matJ(3,3)=(-zi   +zl       )/2.0;
250
251 %%matrice J moins 1
252     invJo=inv(matJ);
253 %%
254 [g304,g305,g306,V1,V2,V3]=vecteurs_g30(xi,yi,zi,xj,yj,zj,xk,yk,zk,...
255                                     xl,yl,zl,xm,ym,zm,xn,yn,zn);
256 %%
257     z1=(zi+zl)/2.0;
258     z2=(zj+zm)/2.0;
259     z3=(zk+zn)/2.0;
260
261     g10(1)=x2-x1;
262     g10(2)=y2-y1;
263     g10(3)=z2-z1;
264
265     g20(1)=x3-x1;
266     g20(2)=y3-y1;
267     g20(3)=z3-z1;
268
269     g21(1)=x3-x2;
270     g21(2)=y3-y2;
271     g21(3)=z3-z2;
272 %% calcul de matrices C
273     C111(1)=-1.0*g304(1);
274     C111(2)=-1.0*g304(2);
275     C111(3)=-1.0*g304(3);
276     C111(4)=-0.25*h1*dot(V1(2,:),g10);
277     C111(5)= 0.25*h1*dot(V1(1,:),g10);
278
279     C112(1)=g304(1);
280     C112(2)=g304(2);
281     C112(3)=g304(3);
282     C112(4)=-0.25*h2*dot(V2(2,:),g10);
283     C112(5)=+0.25*h2*dot(V2(1,:),g10);
284
285     C212(1)=-1.0*g305(1);
286     C212(2)=-1.0*g305(2);

```

```

287     C212(3)=-1.0*g305(3);
288     C212(4)=-0.25*h2*dot(V2(2,:),g21);
289     C212(5)=+0.25*h2*dot(V2(1,:),g21);
290
291     C213(1)=g305(1);
292     C213(2)=g305(2);
293     C213(3)=g305(3);
294     C213(4)=-0.25*h3*dot(V3(2,:),g21);
295     C213(5)=+0.25*h3*dot(V3(1,:),g21);
296
297     C311(1)=g306(1);
298     C311(2)=g306(2);
299     C311(3)=g306(3);
300     C311(4)=+0.25*h1*dot(V1(2,:),g20);
301     C311(5)=-0.25*h1*dot(V1(1,:),g20);
302
303     C313(1)=-1.0*g306(1);
304     C313(2)=-1.0*g306(2);
305     C313(3)=-1.0*g306(3);
306     C313(4)=+0.25*h3*dot(V3(2,:),g20);
307     C313(5)=-0.25*h3*dot(V3(1,:),g20);
308
309     C113(1)=0.0; C113(2)=0.0; C113(3)=0.0; C113(4)=0.0; ...
    C113(5)=0.0;
310     C211(1)=0.0; C211(2)=0.0; C211(3)=0.0; C211(4)=0.0; ...
    C211(5)=0.0;
311     C312(1)=0.0; C312(2)=0.0; C312(3)=0.0; C312(4)=0.0; ...
    C312(5)=0.0;
312
313     for i=1:5
314         C121(i)=-C311(i);
315         C122(i)=-C312(i);
316         C123(i)=-C313(i);
317         C221(i)=-C111(i);
318         C222(i)=-C112(i);
319         C223(i)=-C113(i);
320         C321(i)=-C211(i);
321         C322(i)=-C212(i);
322         C323(i)=-C213(i);
323     end
324     for i=1:3
325         C1(1,i) =C111(i);
326         C1(1,i+3)=C112(i);

```

```

327     C1(1,i+6)=C113(i);
328     C1(2,i) =C121(i);
329     C1(2,i+3)=C122(i);
330     C1(2,i+6)=C123(i);
331
332     C2(1,i) =C211(i);
333     C2(1,i+3)=C212(i);
334     C2(1,i+6)=C213(i);
335     C2(2,i) =C221(i);
336     C2(2,i+3)=C222(i);
337     C2(2,i+6)=C223(i);
338
339     C3(1,i) =C311(i);
340     C3(1,i+3)=C312(i);
341     C3(1,i+6)=C313(i);
342     C3(2,i) =C321(i);
343     C3(2,i+3)=C322(i);
344     C3(2,i+6)=C323(i);
345     end
346
347     for i=4:5
348     C1(1,i+6) =C111(i);
349     C1(1,i+8) =C112(i);
350     C1(1,i+10)=C113(i);
351     C1(2,i+6) =C121(i);
352     C1(2,i+8) =C122(i);
353     C1(2,i+10)=C123(i);
354
355     C2(1,i+6) =C211(i);
356     C2(1,i+8) =C212(i);
357     C2(1,i+10)=C213(i);
358     C2(2,i+6) =C221(i);
359     C2(2,i+8) =C222(i);
360     C2(2,i+10)=C223(i);
361
362     C3(1,i+6) =C311(i);
363     C3(1,i+8) =C312(i);
364     C3(1,i+10)=C313(i);
365     C3(2,i+6) =C321(i);
366     C3(2,i+8) =C322(i);
367     C3(2,i+10)=C323(i);
368     end
369     D1(1,1)= 1.0;

```

```

370     D1(1,2)= 0.0;
371     D1(2,1)= 0.0;
372     D1(2,2)= 1.0;
373     D2(1,1)= 0.0;
374     D2(1,2)=-1.0;
375     D2(2,1)= 1.0;
376     D2(2,2)=-1.0;
377     D3(1,1)=-1.0;
378     D3(1,2)= 1.0;
379     D3(2,1)=-1.0;
380     D3(2,2)= 0.0;
381 %%calcul matrice Bd = D*C
382 Bd1=zeros(2,15);
383 Bd2=zeros(2,15);
384 Bd3=zeros(2,15);
385     for p=1:2
386         for q=1:15
387             aux1=0.0;
388             aux2=0.0;
389             aux3=0.0;
390             for r=1:2
391                 aux1=aux1+D1(p,r)*C1(r,q);
392                 aux2=aux2+D2(p,r)*C2(r,q);
393                 aux3=aux3+D3(p,r)*C3(r,q);
394             end
395             Bd1(p,q)=aux1;
396             Bd2(p,q)=aux2;
397             Bd3(p,q)=aux3;
398         end
399     end
400
401 %%calcul matrice Bdt dans la base global
402 %%calcul matrice transformation T
403
404     T=zeros(15,18);
405     for p=1:3
406         T(3*p-2,3*p-2)=0.5*xX;
407         T(3*p-2,3*p-1)=0.5*xY;
408         T(3*p-2,3*p)=0.5*xZ;
409         T(3*p-1,3*p-2)=0.5*yX;
410         T(3*p-1,3*p-1)=0.5*yY;
411         T(3*p-1,3*p)=0.5*yZ;
412         T(3*p,3*p-2)=0.5*zX;

```

```
413     T(3*p,3*p-1)=0.5*zY;
414     T(3*p,3*p)=0.5*zZ;
415
416     T(3*p-2,3*p+7)=0.5*xX;
417     T(3*p-2,3*p+8)=0.5*xY;
418     T(3*p-2,3*p+9)=0.5*xZ;
419     T(3*p-1,3*p+7)=0.5*yX;
420     T(3*p-1,3*p+8)=0.5*yY;
421     T(3*p-1,3*p+9)=0.5*yZ;
422     T(3*p,3*p+7)=0.5*zX;
423     T(3*p,3*p+8)=0.5*zY;
424     T(3*p,3*p+9)=0.5*zZ;
425     end
426
427 %% [Tr]=matrice_trans_rot
428     Tm=zeros(6,18);
429     Tr=zeros(6,18);
430 %
431     for i=1:3
432         Tm(1,i)=V1(2,i)/h1;
433         Tm(2,i)=-V1(1,i)/h1;
434         Tm(1,i+9)=-Tm(1,i);
435         Tm(2,i+9)=-Tm(2,i);
436
437         Tm(3,i+3)=V2(2,i)/h2;
438         Tm(4,i+3)=-V2(1,i)/h2;
439         Tm(3,i+12)=-Tm(3,i+3);
440         Tm(4,i+12)=-Tm(4,i+3);
441
442         Tm(5,i+6)=V3(2,i)/h3;
443         Tm(6,i+6)=-V3(1,i)/h3;
444         Tm(5,i+15)=-Tm(5,i+6);
445         Tm(6,i+15)=-Tm(6,i+6);
446     end
447
448     for i=1:2
449         Tr(i,1)=Tm(i,1)*xX+Tm(i,2)*yX+Tm(i,3)*zX;
450         Tr(i,2)=Tm(i,1)*xY+Tm(i,2)*yY+Tm(i,3)*zY;
451         Tr(i,3)=Tm(i,1)*xZ+Tm(i,2)*yZ+Tm(i,3)*zZ;
452         Tr(i,10)=-Tr(i,1);
453         Tr(i,11)=-Tr(i,2);
454         Tr(i,12)=-Tr(i,3);
455     end
```

```

456     for i=3:4
457         Tr(i,4)=Tm(i,4)*xX+Tm(i,5)*yX+Tm(i,6)*zX;
458         Tr(i,5)=Tm(i,4)*xY+Tm(i,5)*yY+Tm(i,6)*zY;
459         Tr(i,6)=Tm(i,4)*xZ+Tm(i,5)*yZ+Tm(i,6)*zZ;
460         Tr(i,13)=-Tr(i,4);
461         Tr(i,14)=-Tr(i,5);
462         Tr(i,15)=-Tr(i,6);
463     end
464     for i=5:6
465         Tr(i,7)=Tm(i,7)*xX+Tm(i,8)*yX+Tm(i,9)*zX;
466         Tr(i,8)=Tm(i,7)*xY+Tm(i,8)*yY+Tm(i,9)*zY;
467         Tr(i,9)=Tm(i,7)*xZ+Tm(i,8)*yZ+Tm(i,9)*zZ;
468         Tr(i,16)=-Tr(i,7);
469         Tr(i,17)=-Tr(i,8);
470         Tr(i,18)=-Tr(i,9);
471     end
472
473 %%
474     for p=1:6
475         for q=1:18
476             T(p+9,q)=Tr(p,q);
477         end
478     end
479     Bdt1=zeros(2,18);
480     Bdt2=zeros(2,18);
481     Bdt3=zeros(2,18);
482     for p=1:2
483         for q=1:18
484             aux1=0.0;
485             aux2=0.0;
486             aux3=0.0;
487             for r=1:15
488                 aux1=aux1+Bd1(p,r)*T(r,q);
489                 aux2=aux2+Bd2(p,r)*T(r,q);
490                 aux3=aux3+Bd3(p,r)*T(r,q);
491             end
492             Bdt1(p,q)=aux1;
493             Bdt2(p,q)=aux2;
494             Bdt3(p,q)=aux3;
495         end
496     end
497
498     for p=1:2

```

```
499     for q=1:18
500         aux1=0.0;
501         aux2=0.0;
502         aux3=0.0;
503         for r=1:2
504             aux1=aux1+invJo(p,r)*Bdt1(r,q);
505             aux2=aux2+invJo(p,r)*(Bdt3(r,q)-Bdt1(r,q));
506             aux3=aux3+invJo(p,r)*(Bdt2(r,q)-Bdt1(r,q));
507         end
508         Bc0(p,q)=aux1;
509         Bc1(p,q)=aux2;
510         Bc2(p,q)=aux3;
511     end
512 end
513
514 %% stabilisation
515 b=1.0/2.0;
516 t=1.0/2.0;
517 u=1.0/2.0;
518 %
519 Bksi(1,1)=-b;
520 Bksi(2,1)=-b;
521 Bksi(3,1)=-u;
522 Bksi(1,2)= b;
523 Bksi(2,2)=0.0;
524 Bksi(3,2)=0.0;
525 Bksi(1,3)=0.0;
526 Bksi(2,3)= b;
527 Bksi(3,3)=0.0;
528 Bksi(1,4)=-t;
529 Bksi(2,4)=-t;
530 Bksi(3,4)= u;
531 Bksi(1,5)= t;
532 Bksi(2,5)=0.0;
533 Bksi(3,5)=0.0;
534 Bksi(1,6)=0.0;
535 Bksi(2,6)= t;
536 Bksi(3,6)=0.0;
537
538 for j=1:6
539     ...
    Bx(j)=invJo(1,1)*Bksi(1,j)+invJo(1,2)*Bksi(2,j)+invJo(1,3)*Bksi(3,j);
```

```

540     ...
    By(j)=invJo(2,1)*Bksi(1,j)+invJo(2,2)*Bksi(2,j)+invJo(2,3)*Bksi(3,j);
541     ...
    Bz(j)=invJo(3,1)*Bksi(1,j)+invJo(3,2)*Bksi(2,j)+invJo(3,3)*Bksi(3,j);
542 end
543
544 %% [Vgamma]=VecteursGamma(xyz,Bx,By,Bz);
545 XYZe(1,1)=xi;
546 XYZe(1,2)=yi;
547 XYZe(1,3)=zi;
548 XYZe(2,1)=xj;
549 XYZe(2,2)=yj;
550 XYZe(2,3)=zj;
551 XYZe(3,1)=xk;
552 XYZe(3,2)=yk;
553 XYZe(3,3)=zk;
554 XYZe(4,1)=xl;
555 XYZe(4,2)=yl;
556 XYZe(4,3)=zl;
557 XYZe(5,1)=xm;
558 XYZe(5,2)=ym;
559 XYZe(5,3)=zm;
560 XYZe(6,1)=xn;
561 XYZe(6,2)=yn;
562 XYZe(6,3)=zn;
563
564 %% vecteurs {h}
565 mathH(1,1)= 0.0;
566 mathH(1,2)= 0.0;
567 mathH(2,1)= 0.0;
568 mathH(2,2)=-1.0;
569 mathH(3,1)=-1.0;
570 mathH(3,2)= 0.0;
571 mathH(4,1)= 0.0;
572 mathH(4,2)= 0.0;
573 mathH(5,1)= 0.0;
574 mathH(5,2)= 1.0;
575 mathH(6,1)= 1.0;
576 mathH(6,2)= 0.0;
577 Vgamma=zeros(6,2);
578 for j=1:2
579     hX=0.0;
580     hY=0.0;

```

```

581     hZ=0.0;
582     for i=1:6
583         hX=hX+matH(i,j)*XYZe(i,1);
584         hY=hY+matH(i,j)*XYZe(i,2);
585         hZ=hZ+matH(i,j)*XYZe(i,3);
586     end
587     for i=1:6
588         Vgamma(i,j)=(matH(i,j)-hX*Bx(i)-hY*By(i)-hZ*Bz(i))/2.0;
589     end
590 end
591 %%
592 %%matrice gradient pour la stabilisation stabilisation
593 Bsz=zeros(2,18);
594 for j=1:6
595     %%Bsx pour q1x,q1y; Bsy pour q1y,q2y; Bsz pour q1z,q2z
596     k=3*(j-1);
597     BsZ(1,k+1)=zX*Vgamma(j,1);
598     BsZ(1,k+2)=zY*Vgamma(j,1);
599     BsZ(1,k+3)=zZ*Vgamma(j,1);
600     BsZ(2,k+1)=zX*Vgamma(j,2);
601     BsZ(2,k+2)=zY*Vgamma(j,2);
602     BsZ(2,k+3)=zZ*Vgamma(j,2);
603 end
604 Bsz=Bsz*invJo(3,3);
605 end

```

```

1
2 %%
3 %%appel des deux procédures précalculées Bt_Bw
4 function [Bt,Bw]=matricesBtBw(x1,y1,x2,y2,x3,y3,aire)
5
6     x21=x2-x1;
7     y21=y2-y1;
8     L21=sqrt(x21*x21+y21*y21);
9     c4=x21/L21;
10    s4=y21/L21;
11    x32=x3-x2;
12    y32=y3-y2;
13    L32=sqrt(x32*x32+y32*y32);
14    c5=x32/L32;
15    s5=y32/L32;

```

```

16     x13=x1-x3;
17     y13=y1-y3;
18     L13=sqrt(x13*x13+y13*y13);
19     c6=x13/L13;
20     s6=y13/L13;
21
22 %%
23 %%matrice gradient faisant intervenir les rotations au milieu des ...
    cotÃl's
24 %%Bt du l'Ãl'lÃl'ment DKT6
25 Bt=matriceBt(x21,y21,x32,y32,x13,y13,aire,c4,s4,c5,s5,c6,s6);
26
27 %%
28 %%matrice gradient faisant intervenir les translations normales ...
    aux sommets
29 %%de l'Ãl'lÃl'ment DKT6
30 Bw=matriceBw(aire,c4,s4,c5,s5,c6,s6);
31
32 end

```

```

1 %%
2 %%matrice gradient faisant intervenir les rotations au milieu des ...
    cotÃl's
3 %%Bt du l'Ãl'lÃl'ment DKT6
4 function Bt=matriceBt(x21,y21,x32,y32,x13,y13,...
5     aire,c4,s4,c5,s5,c6,s6)
6
7     Bt(1,1)=(y21)*s4/aire;
8     Bt(1,2)=(y32)*s5/aire;
9     Bt(1,3)=(y13)*s6/aire;
10    Bt(2,1)=(x21)*c4/aire;
11    Bt(2,2)=(x32)*c5/aire;
12    Bt(2,3)=(x13)*c6/aire;
13    Bt(3,1)=((-y21)*c4+(-x21)*s4)/aire;
14    Bt(3,2)=((-y32)*c5+(-x32)*s5)/aire;
15    Bt(3,3)=((-y13)*c6+(-x13)*s6)/aire;
16 end

```

F. Matlab source code

```
1 %%
2 %%matrice gradient faisant intervenir les translations normales ...
   aux sommets
3 %%de l'Élément DKT6
4 function Bw=matriceBw(aire,c4,s4,c5,s5,c6,s6)
5
6     Bw(1,1)=(s4*c4-s6*c6)/aire;
7     Bw(1,2)=(s5*c5-s4*c4)/aire;
8     Bw(1,3)=(s6*c6-s5*c5)/aire;
9
10    %
11    Bw(2,1)=-Bw(1,1);
12    Bw(2,2)=-Bw(1,2);
13    Bw(2,3)=-Bw(1,3);
14
15    %
16    Bw(3,1)=(s4*s4-c4*c4+c6*c6-s6*s6)/aire;
17    Bw(3,2)=(s5*s5-c5*c5+c4*c4-s4*s4)/aire;
18    Bw(3,3)=(s6*s6-c6*c6+c5*c5-s5*s5)/aire;
19
20 end
```

```
1 %%Donne les trois rotations aux milieux des cotés du DKT6 en ...
   fonction des
2 %%translations dans le repère global des 6 noeuds
3 function T=TransformerTransRot(XYZ)
4     Xi=XYZ(1,1);
5     Yi=XYZ(1,2);
6     Zi=XYZ(1,3);
7     Xj=XYZ(2,1);
8     Yj=XYZ(2,2);
9     Zj=XYZ(2,3);
10    Xk=XYZ(3,1);
11    Yk=XYZ(3,2);
12    Zk=XYZ(3,3);
13    Xl=XYZ(4,1);
14    Yl=XYZ(4,2);
15    Zl=XYZ(4,3);
16    Xm=XYZ(5,1);
17    Ym=XYZ(5,2);
18    Zm=XYZ(5,3);
19    Xn=XYZ(6,1);
20    Yn=XYZ(6,2);
```

```

21     Zn=XYZ(6,3);
22
23     T=zeros(3,18);
24
25     %%theta 4
26     [xXf,xYf,xZf,yXf,yYf,yZf,zXf,zYf,zZf...
27      xif,yif,xjf,yjf,xlf,ylf,xfm,ymf,dAf...
28      ]=RepereFaceMilieuxij...
29      (Xi,Yi,Zi,Xj,Yj,Zj,Xm,Ym,Zm,Xl,Yl,Zl);
30
31     Pf=(xlf-xjf)/dAf/2;
32     T(1,1)=Pf*zXf;
33     T(1,2)=Pf*zYf;
34     T(1,3)=Pf*zZf;
35     T(1,13)=-Pf*zXf;
36     T(1,14)=-Pf*zYf;
37     T(1,15)=-Pf*zZf;
38     Qf=(xif-xfm)/dAf/2;
39     T(1,4)=Qf*zXf;
40     T(1,5)=Qf*zYf;
41     T(1,6)=Qf*zZf;
42     T(1,10)=-Qf*zXf;
43     T(1,11)=-Qf*zYf;
44     T(1,12)=-Qf*zZf;
45
46     %%theta 5
47     [xXf,xYf,xZf,yXf,yYf,yZf,zXf,zYf,zZf...
48      xjf,yjf,xkf,ykf,xfm,ymf,xnf,ynf,dAf...
49      ]=RepereFaceMilieuxij...
50      (Xj,Yj,Zj,Xk,Yk,Zk,Xn,Yn,Zn,Xm,Ym,Zm);
51
52     Pf=(xfm-xkf)/dAf/2;
53     T(2,4)=Pf*zXf;
54     T(2,5)=Pf*zYf;
55     T(2,6)=Pf*zZf;
56     T(2,16)=-Pf*zXf;
57     T(2,17)=-Pf*zYf;
58     T(2,18)=-Pf*zZf;
59     Qf=(xjf-xnf)/dAf/2;
60     T(2,7)=Qf*zXf;
61     T(2,8)=Qf*zYf;
62     T(2,9)=Qf*zZf;
63     T(2,13)=-Qf*zXf;

```

F. Matlab source code

```

64     T(2,14)=-Qf*zYf;
65     T(2,15)=-Qf*zZf;
66
67     %%theta 6
68     [xXf,xYf,xZf,yXf,yYf,yZf,zXf,zYf,zZf...
69      xkf,ykf,xif,yif,xfn,yfn,xlf,ylf,dAf...
70      ]=RepereFaceMilieuxij...
71      (Xk,Yk,Zk,Xi,Yi,Zi,Xl,Yl,Zl,Xn,Yn,Zn);
72
73     Pf=(xfn-xif)/dAf/2;
74     T(3,7)=Pf*zXf;
75     T(3,8)=Pf*zYf;
76     T(3,9)=Pf*zZf;
77     T(3,10)=-Pf*zXf;
78     T(3,11)=-Pf*zYf;
79     T(3,12)=-Pf*zZf;
80     Qf=(xkf-xlf)/dAf/2;
81     T(3,1)=Qf*zXf;
82     T(3,2)=Qf*zYf;
83     T(3,3)=Qf*zZf;
84     T(3,16)=-Qf*zXf;
85     T(3,17)=-Qf*zYf;
86     T(3,18)=-Qf*zZf;
87
88     end

```

```

1  function PLAST3D2(PROP, UPDATE,LTAN, XYZ, Connec)
2  %*****
3  % MAIN PROGRAM COMPUTING GLOBAL STIFFNESS MATRIX RESIDUAL FORCE FOR
4  % NON LINEAR MODELS
5  %*****
6  %%
7  global DISPDD DISPTD FORCE GKF SIGMA
8  %
9  % Integration points and weights (2-point integration)
10     [NUMNP, NDOF] = size(XYZ);           % Analysis parameters
11     NE = size(Connec,1);
12     p1=NUMNP+1;
13     p2=NUMNP+NE;
14     DDLsup=p1:p2;
15     NEQ = NDOF*NUMNP;

```

```

16  LE=[Connec DDLsup'];
17  %NEQ = NDOF*NUMNP;
18  %LE=Connec;
19  % Integration points and weights
20  ZG=[-1 -sqrt(3/7) 0 sqrt(3/7) 1];
21  WGT=[0.1 49.0/90.0 32.0/45.0 49.0/90.0 0.1];
22  %
23  %%
24  NDDL=19;
25  %
26  % Index for history variables (each integration pt)
27  INTN=0;
28  %
29  %LOOP OVER ELEMENTS, THIS IS MAIN LOOP TO COMPUTE K AND F
30  for IE=1:NE
31  % Nodal coordinates and incremental displacements
32  ELXY=XYZ(LE(IE,1:6),:);
33  % Local to global mapping
34  IDOF=zeros(1,NDDL);
35  for I=1:6
36  II=(I-1)*NDOF+1;
37  IDOF(II:II+2)=(LE(IE,I)-1)*NDOF+1:(LE(IE,I)-1)*NDOF+3;
38  end
39  IDOF(19)=NEQ+IE;
40  %%
41  DSP=DISPTD(IDOF);
42  DSPD=DISPDD(IDOF);
43  DSPDLigne=DSPD;
44  DSP=reshape(DSP(1:18),NDOF,6);
45  DSPD=reshape(DSPD(1:18),NDOF,6);
46  %%
47  %LOOP OVER INTEGRATION POINTS
48  for LZ=1:5
49  E1=1/3; E2=1/3; E3=ZG(LZ);
50  INTN = INTN + 1;
51  % Determinant and shape function derivatives
52  [~, DET] = SHAPEL([E1 E2 E3], ELXY+DSP');
53  FAC=WGT(LZ)*DET/2;
54  %
55  % Previous converged history variables
56  STRESSN=SIGMA(1:6,INTN);
57  %
58  %

```

```

59 [BM,BG,Bs]=Bprisme(ELXY+DSP',[E1 E2 E3]);
60 DDEPS=BM*DSPDLigne;
61 %
62 % Compute stress;
63 [DTAN,~]=matrice_D(PROP(1),PROP(2));
64 STRESS = STRESSN + DTAN*DDEPS;
65 %-----Stabilisation
66 Dsz=eye(2)*DTAN(3,3)/1200;
67 Kstab=(Bs')*Dsz*Bs;
68 %
69 % Update plastic variables
70 if UPDATE
71 SIGMA(1:6,INTN)=STRESS;
72 continue;
73 end
74 %
75 % Residual forces
76 FORCE(IDOF) = FORCE(IDOF) - FAC*BM'*STRESS-FAC*Kstab*DSPDLigne;
77 %
78 if LTAN
79 SIG=[STRESS(1) STRESS(4) STRESS(5);
80       STRESS(4) STRESS(2) STRESS(6);
81       STRESS(5) STRESS(6) STRESS(3)];
82 SHEAD=kron(eye(3),SIG);
83 %
84 EKF = BM'*(DTAN)*BM + BG'*SHEAD*BG+Kstab;
85 % Stiffness matrix
86 GKF(IDOF,IDOF)=GKF(IDOF,IDOF)+FAC*EKF;
87 end
88
89 end
90 end
91 end

```

Bibliography

- [ABE 01] ABED-MERAÏM F., COMBESURE A.
SHB8PS a new intelligent assumed strain continuum mechanics shell element ...
for impact analysis on a rotating body. *1st MIT Conference*, 2001.
- [ABE 02] ABED-MERAÏM F., COMBESURE A.
SHB8PS—a new adaptive, assumed-strain continuum mechanics shell element for
impact analysis. *Computers & Structures*, vol. 80, n° 9, 2002, p. 791–803, Elsevier.
- [ABE 09] ABED-MERAÏM F., COMBESURE A.
An improved assumed strain solid–shell element formulation with physical stabilization for geometric non-linear applications and elastic–plastic stability analysis. *International Journal for Numerical Methods in Engineering*, vol. 80, n° 13, 2009, p. 1640–1686, Wiley Online Library.
- [ADA 05] ADAM L., PONTHOT J.-P.
Thermomechanical modeling of metals at finite strains: first and mixed order finite elements. *International Journal of Solids and Structures*, vol. 42, n° 21-22, 2005, p. 5615–5655, Elsevier.
- [AHM 70] AHMAD S., IRONS B. M., ZIENKIEWICZ O.
Analysis of thick and thin shell structures by curved finite elements. *International Journal for Numerical Methods in Engineering*, vol. 2, n° 3, 1970, p. 419–451, Wiley Online Library.
- [AND 93] ANDELFINGER U., RAMM E.
EAS-elements for two-dimensional, three-dimensional, plate and shell structures and their equivalence to HR-elements. *International Journal for Numerical Methods in Engineering*, vol. 36, n° 8, 1993, p. 1311–1337, Wiley Online Library.
- [BAN 87] BANK L. C.
Shear coefficients for thin-walled composite beams. *Composite Structures*, vol. 8, n° 1, 1987, p. 47–61, Elsevier.
- [BAR 81] BARRETT D., SOLER A.
A finite element model for thick beams. *Computer Methods in Applied Mechanics and Engineering*, vol. 25, n° 3, 1981, p. 299–313, Elsevier.

- [BAS 12] BASSA B., SABOURIN F., BRUNET M.
A new nine-node solid-shell finite element using complete 3D constitutive laws. *International Journal for Numerical Methods in Engineering*, vol. 92, n° 7, 2012, p. 589–636, Wiley Online Library.
- [BAT 85] BATHE K.-J., DVORKIN E. N.
A four-node plate bending element based on Mindlin/Reissner plate theory and a mixed interpolation. *International Journal for Numerical Methods in Engineering*, vol. 21, n° 2, 1985, p. 367–383, Wiley Online Library.
- [BAT 86] BATHE K.-J., DVORKIN E. N.
A formulation of general shell elements—the use of mixed interpolation of tensorial components. *International Journal for Numerical Methods in Engineering*, vol. 22, n° 3, 1986, p. 697–722, Wiley Online Library.
- [BAT 90a] BATHOZ J., DHATT G.
. « Modélisation des structures par éléments finis, 2, poutres et plaques Hermes », 1990.
- [BAT 90b] BATOZ J., DHATT G.
Modélisation des structures par éléments finis. *Edition Hermès, Paris*, , 1990.
- [BAT 92] BATOZ J.-L., DHATT G.
Modélisation des structures par éléments finis, Volume 3. Solides élastiques. *Hermes pp 467 à 470, Paris*, , 1992.
- [BEL 73] BELYTSCHKO T., HSIEH B.
Non-linear transient finite element analysis with convected co-ordinates. *International Journal for Numerical Methods in Engineering*, vol. 7, n° 3, 1973, p. 255–271, Wiley Online Library.
- [BEL 83] BELYTSCHKO T., TSAY C.-S.
A stabilization procedure for the quadrilateral plate element with one-point quadrature. *International Journal for Numerical Methods in Engineering*, vol. 19, n° 3, 1983, p. 405–419, Wiley Online Library.
- [BEL 84] BELYTSCHKO T., ONG J. S.-J., LIU W. K., KENNEDY J. M.
Hourglass control in linear and nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, vol. 43, n° 3, 1984, p. 251–276, Elsevier.
- [BEL 93] BELYTSCHKO T., BINDEMAN L. P.
Assumed strain stabilization of the eight node hexahedral element. *Computer Methods in Applied Mechanics and Engineering*, vol. 105, n° 2, 1993, p. 225–260, Elsevier.
- [BET 96] BETSCH P., GRUTTMANN F., STEIN E.
A 4-node finite shell element for the implementation of general hyperelastic 3D-

elasticity at finite strains. *Computer Methods in Applied Mechanics and Engineering*, vol. 130, n° 1-2, 1996, p. 57–79, Elsevier.

[BET 99] BETSCH P., STEIN E.

Numerical implementation of multiplicative elasto-plasticity into assumed strain elements with application to shells at large strains. *Computer methods in applied mechanics and engineering*, vol. 179, n° 3-4, 1999, p. 215–245, Elsevier.

[BIS 97] BISCHOFF M., RAMM E.

Shear deformable shell elements for large strains and rotations. *International Journal for Numerical Methods in Engineering*, vol. 40, n° 23, 1997, p. 4427–4449, Wiley Online Library.

[BIS 00] BISCHOFF M., RAMM E.

On the physical significance of higher order kinematic and static variables in a three-dimensional shell formulation. *International Journal of Solids and Structures*, vol. 37, n° 46-47, 2000, p. 6933–6960, Elsevier.

[BOI 94] BOISSE P., DANIEL J., GELIN J.

A C0 three-node shell element for non-linear structural analysis. *International journal for numerical methods in engineering*, vol. 37, n° 14, 1994, p. 2339–2364, Wiley Online Library.

[BRA 94] BRAUN M., BISCHOFF M., RAMM E.

Nonlinear shell formulations for complete three-dimensional constitutive laws including composites and laminates. *Computational Mechanics*, vol. 15, n° 1, 1994, p. 1–18, Springer.

[BRA 02] BRANK B., KORELC J., IBRAHIMBEGOVIĆ A.

Nonlinear shell problem formulation accounting for through-the-thickness stretching and its finite element implementation. *Computers & structures*, vol. 80, n° 9-10, 2002, p. 699–717, Elsevier.

[BRA 05] BRANK B.

Nonlinear shell models with seven kinematic parameters. *Computer methods in applied mechanics and engineering*, vol. 194, n° 21-24, 2005, p. 2336–2362, Elsevier.

[BRE 80] BRENDEL B., RAMM E.

Linear and nonlinear stability analysis of cylindrical shells. *Computers & Structures*, vol. 12, n° 4, 1980, p. 549–558, Elsevier.

[BÜC 94] BÜCHTER N., RAMM E., ROEHL D.

Three-dimensional extension of non-linear shell formulation based on the enhanced assumed strain concept. *International journal for numerical methods in engineering*, vol. 37, n° 15, 1994, p. 2551–2568, Wiley Online Library.

- [CAR 02] CARDOSO R. P., YOON J.-W., GRÁCIO J. J., BARLAT F., DE SÁ J. M. C.
Development of a one point quadrature shell element for nonlinear applications with contact and anisotropy. *Computer Methods in Applied Mechanics and Engineering*, vol. 191, n^o 45, 2002, p. 5177–5206, Elsevier.
- [CAR 05] CARDOSO R. P., YOON J. W.
One point quadrature shell element with through-thickness stretch. *Computer Methods in Applied Mechanics and Engineering*, vol. 194, n^o 9, 2005, p. 1161–1199, Elsevier.
- [CAR 08] CARDOSO R. P., YOON J. W., MAHARDIKA M., CHOUDHRY S., ALVES DE SOUSA R., FONTES VALENTE R.
Enhanced assumed strain (EAS) and assumed natural strain (ANS) methods for one-point quadrature solid-shell elements. *International Journal for Numerical Methods in Engineering*, vol. 75, n^o 2, 2008, p. 156–187, Wiley Online Library.
- [DEB 12] DE BORST R., CRISFIELD M. A., REMMERS J. J., VERHOOSEL C. V.
Nonlinear finite element analysis of solids and structures. John Wiley & Sons, 2012.
- [DES 07] DE SOUSA R. A., YOON J. W., CARDOSO R., VALENTE R. F., GRACIO J.
On the use of a reduced enhanced solid-shell (RESS) element for sheet forming simulations. *International journal of plasticity*, vol. 23, n^o 3, 2007, p. 490–515, Elsevier.
- [DHA 05] DHATT G., TOUZOT G., LEFRANÇOIS E.
Méthode des éléments finis. Lavoisier, 2005.
- [DON 75] DON O., ALMROTH B. O.
Buckling of bars, plates, and shells. McGraw-Hill, 1975.
- [DVO 84] DVORKIN E. N., BATHE K.-J.
A continuum mechanics based four-node shell element for general non-linear analysis. *Engineering computations*, vol. 1, n^o 1, 1984, p. 77–88, MCB UP Ltd.
- [DVO 95a] DVORKIN E. N.
Nonlinear analysis of shells using the MITC formulation. *Archives of Computational Methods in Engineering*, vol. 2, n^o 2, 1995, Page 1, Springer.
- [DVO 95b] DVORKIN E. N., PANTUSO D., REPETTO E. A.
A formulation of the MITC4 shell element for finite strain elasto-plastic analysis. *Computer Methods in Applied Mechanics and Engineering*, vol. 125, n^o 1-4, 1995, p. 17–40, Elsevier.
- [EBE 99] EBERLEIN R., WRIGGERS P.
Finite element concepts for finite elastoplastic strains and isotropic stress response in shells: theoretical and computational analysis. *Computer Methods in Applied Mechanics and Engineering*, vol. 171, n^o 3-4, 1999, p. 243–279, Elsevier.

- [ELA 00] EL-ABBASI N., MEGUID S.
A new shell element accounting for through-thickness deformation. *Computer Methods in Applied Mechanics and Engineering*, vol. 189, n^o 3, 2000, p. 841–862, Elsevier.
- [Ele 20] ELECTRICITÉ DE FRANCE
. « Finite element *code_aster*, Structures and Thermomechanics Analysis for Studies and Research ». www.code-aster.org, 1989–2019, 1989–2020.
- [FON 05] FONTES VALENTE R., PARENTE M., NATAL JORGE R., CÉSAR DE SÁ J., GRÁCIO J.
Enhanced transverse shear strain shell formulation applied to large elasto-plastic deformation problems. *International Journal for Numerical Methods in Engineering*, vol. 62, n^o 10, 2005, p. 1360–1398, Wiley Online Library.
- [FRI 62] FRISCH-FAY R.
Flexible bars. Butterworths, 1962.
- [HAL 83] HALLQUIST J. O.
Theoretical manual for DYNA3D. rapport, 1983, Lawrence Livermore National Lab., CA (USA).
- [HAU 98] HAUPTMANN R., SCHWEIZERHOF K.
A systematic development of ‘solid-shell’ element formulations for linear and non-linear analyses employing only displacement degrees of freedom. *International Journal for Numerical Methods in Engineering*, vol. 42, n^o 1, 1998, p. 49–69, Wiley Online Library.
- [HAU 00] HAUPTMANN R., SCHWEIZERHOF K., DOLL S.
Extension of the ‘solid-shell’ concept for application to large elastic and large elasto-plastic deformations. *International Journal for Numerical Methods in Engineering*, vol. 49, n^o 9, 2000, p. 1121–1141, Wiley Online Library.
- [HAU 01] HAUPTMANN R., DOLL S., HARNAU M., SCHWEIZERHOF K.
Solid-shell’elements with linear and quadratic shape functions at large deformations with nearly incompressible materials. *Computers & Structures*, vol. 79, n^o 18, 2001, p. 1671–1685, Elsevier.
- [HIB 98] HIBBETT, KARLSSON, SORENSEN
ABAQUS/standard: User’s Manual, vol. 1. Hibbitt, Karlsson & Sorensen, 1998.
- [HUG 80] HUGHES T. J.
Generalization of selective integration procedures to anisotropic and nonlinear media. *International Journal for Numerical Methods in Engineering*, vol. 15, n^o 9, 1980, p. 1413–1418, Wiley Online Library.
- [HUG 81a] HUGHES T. J., LIU W. K.
Nonlinear finite element analysis of shells: Part I. Three-dimensional shells. *Computer methods in applied mechanics and engineering*, vol. 26, n^o 3, 1981, p. 331–362, Elsevier.

- [HUG 81b] HUGHES T. J., TEZDUYAR T.
Finite elements based upon Mindlin plate theory with particular reference to the four-node bilinear isoparametric element. *Journal of applied mechanics*, vol. 48, n° 3, 1981, p. 587–596, American Society of Mechanical Engineers.
- [KAS 00] KASPER E. P., TAYLOR R. L.
A mixed-enhanced strain method: Part I: Geometrically linear problems. *Computers & Structures*, vol. 75, n° 3, 2000, p. 237–250, Elsevier.
- [KEM 98] KEMP B. L., CHO C., LEE S. W.
A four-node solid shell element formulation with assumed strain. *International Journal for Numerical Methods in Engineering*, vol. 43, n° 5, 1998, p. 909–924, Wiley Online Library.
- [KIM 05] KIM K., LIU G., HAN S.
A resultant 8-node solid-shell element for geometrically nonlinear analysis. *Computational Mechanics*, vol. 35, n° 5, 2005, p. 315–331, Springer.
- [KIM 08] KIM D.-N., BATHE K.-J.
A 4-node 3D-shell element to model shell surface tractions and incompressible behavior. *Computers & Structures*, vol. 86, n° 21-22, 2008, p. 2027–2041, Elsevier.
- [KLI 97] KLINKEL S., WAGNER W.
A geometrical non-linear brick element based on the EAS-method. *International Journal for Numerical Methods in Engineering*, vol. 40, n° 24, 1997, p. 4529–4545, Wiley Online Library.
- [KLI 99] KLINKEL S., GRUTTMANN F., WAGNER W.
A continuum based three-dimensional shell element for laminated structures. *Computers & Structures*, vol. 71, n° 1, 1999, p. 43–62, Elsevier.
- [KLI 06] KLINKEL S., GRUTTMANN F., WAGNER W.
A robust non-linear solid shell element based on a mixed variational formulation. *Computer methods in applied mechanics and engineering*, vol. 195, n° 1-3, 2006, p. 179–201, Elsevier.
- [KUL 08] KULIKOV G., PLOTNIKOVA S.
Finite rotation geometrically exact four-node solid-shell element with seven displacement degrees of freedom. *Computer Modeling in Engineering & Sciences*, vol. 28, n° 1, 2008, p. 15–38.
- [LAU 06] LAULUSA A., BAUCHAU O., CHOI J., TAN V., LI L.
Evaluation of some shear deformable shell elements. *International Journal of Solids and Structures*, vol. 43, n° 17, 2006, p. 5033–5054, Elsevier.
- [LEE 04] LEE P.-S., BATHE K.-J.

-
- Development of MITC isotropic triangular shell finite elements. *Computers & Structures*, vol. 82, n° 11-12, 2004, p. 945–962, Elsevier.
- [LEG 03] LEGAY A., COMBESURE A.
Elastoplastic stability analysis of shells using the physically stabilized finite element SHB8PS. *International Journal for Numerical Methods in Engineering*, vol. 57, n° 9, 2003, p. 1299–1322, Wiley Online Library.
- [LIU 98] LIU W. K., GUO Y., TANG S., BELYTSCHKO T.
A multiple-quadrature eight-node hexahedral finite element for large deformation elastoplastic analysis. *Computer Methods in Applied Mechanics and Engineering*, vol. 154, n° 1-2, 1998, p. 69–132, Elsevier.
- [MAC 85] MACNEAL R. H., HARDER R. L.
A proposed standard set of problems to test finite element accuracy. *Finite elements in analysis and design*, vol. 1, n° 1, 1985, p. 3–20, Elsevier.
- [MAC 87] MACNEAL R. H.
A theorem regarding the locking of tapered four-noded membrane elements. *International Journal for Numerical Methods in Engineering*, vol. 24, n° 9, 1987, p. 1793–1799, Wiley Online Library.
- [MAC 93] MACNEAL R. H.
Finite Elements: Their Design and Performance, vol. 3. Marcel Dekker, Inc.; 1 edition (October 28, 1993), 1993.
- [MAS 00] MASUD A., THAM C. L.
Three-dimensional corotational framework for elasto-plastic analysis of multilayered composite shells. *AIAA journal*, vol. 38, n° 12, 2000, p. 2320–2327.
- [MAT 18] MATAIX V., FLORES F. G., ROSSI R., OÑATE E.
Triangular prismatic solid-shell element with generalised deformation description. *European Journal of Computational Mechanics*, vol. 27, n° 1, 2018, p. 1–32, Taylor & Francis.
- [MIE 98] MIEHE C.
A theoretical and computational model for isotropic elastoplastic stress analysis in shells at large strains. *Computer Methods in Applied Mechanics and Engineering*, vol. 155, n° 3-4, 1998, p. 193–233, Elsevier.
- [MIE 04] MIEHE C., APEL N.
Anisotropic elastic–plastic analysis of shells at large strains. A comparison of multiplicative and additive approaches to enhanced finite element design and constitutive modelling. *International Journal for Numerical Methods in Engineering*, vol. 61, n° 12, 2004, p. 2067–2113, Wiley Online Library.
-

- [MIL 90] MILITELLO C., FELIPPA C. A.
A variational justification of the assumed natural strain formulation of finite elements—I. Variational principles. *Computers & Structures*, vol. 34, n° 3, 1990, p. 431–438, Elsevier.
- [PAR 86] PARK K., STANLEY G.
A curved C0 shell element based on assumed natural-coordinate strains. *Journal of Applied Mechanics*, vol. 53, n° 2, 1986, p. 278–290, American Society of Mechanical Engineers.
- [PAR 95] PARISCH H.
A continuum-based shell theory for non-linear applications. *International Journal for Numerical Methods in Engineering*, vol. 38, n° 11, 1995, p. 1855–1883, Wiley Online Library.
- [PRA 01] PRATHAP G.
Finite element analysis as computation. *CSIR centre for mathematical modelling and computer simulation*, , 2001.
- [REE 97] REESE S., WRIGGERS P.
A material model for rubber-like polymers exhibiting plastic deformation: computational aspects and a comparison with experimental results. *Computer Methods in Applied Mechanics and Engineering*, vol. 148, n° 3-4, 1997, p. 279–298, Elsevier.
- [REE 00] REESE S., WRIGGERS P., REDDY B.
A new locking-free brick element technique for large deformation problems in elasticity. *Computers & Structures*, vol. 75, n° 3, 2000, p. 291–304, Elsevier.
- [REE 07] REESE S.
A large deformation solid-shell concept based on reduced integration with hourglass stabilization. *International Journal for Numerical Methods in Engineering*, vol. 69, n° 8, 2007, p. 1671–1716, Wiley Online Library.
- [ROE 92] ROELANDT J., BATOZ J.
Shell finite element for deep drawing problems: computational aspects and results. *Finite Inelastic Deformations—Theory and Applications*, p. 423–430 Springer, 1992.
- [SAL 90] SALEEB A., CHANG T., GRAF W., YINGYEUNYONG S.
A hybrid/mixed model for non-linear shell analysis and its applications to large-rotation problems. *International Journal for Numerical Methods in Engineering*, vol. 29, n° 2, 1990, p. 407–446, Wiley Online Library.
- [SAN 92] SANSOUR C., BUFLER H.
An exact finite rotation shell theory, its mixed variational formulation and its finite element implementation. *International Journal for Numerical Methods in Engineering*, vol. 34, n° 1, 1992, p. 73–115, Wiley Online Library.

- [SAN 11] SANSALONE M., SABOURIN F., BRUNET M.
A new shell formulation using complete 3D constitutive laws. *International Journal for Numerical Methods in Engineering*, vol. 86, n° 6, 2011, p. 688–716, Wiley Online Library.
- [SCH 09] SCHWARZE M., REESE S.
A reduced integration solid-shell finite element based on the EAS and the ANS concept—Geometrically linear problems. *International Journal for Numerical Methods in Engineering*, vol. 80, n° 10, 2009, p. 1322–1355, Wiley Online Library.
- [SIM 85] SIMO J. C., TAYLOR R. L.
Consistent tangent operators for rate-independent elastoplasticity. *Computer methods in applied mechanics and engineering*, vol. 48, n° 1, 1985, p. 101–118, Elsevier.
- [SIM 90a] SIMO J. C., FOX D. D., RIFAI M. S.
On a stress resultant geometrically exact shell model. Part III: Computational aspects of the nonlinear theory. *Computer Methods in Applied Mechanics and Engineering*, vol. 79, n° 1, 1990, p. 21–70, Elsevier.
- [SIM 90b] SIMO J. C., RIFAI M.
A class of mixed assumed strain methods and the method of incompatible modes. *International journal for numerical methods in engineering*, vol. 29, n° 8, 1990, p. 1595–1638, Wiley Online Library.
- [SIM 92a] SIMO J. C., LAURSEN T.
An augmented Lagrangian treatment of contact problems involving friction. *Computers & Structures*, vol. 42, n° 1, 1992, p. 97–116, Elsevier.
- [SIM 92b] SIMO J.-C., ARMERO F.
Geometrically non-linear enhanced strain mixed methods and the method of incompatible modes. *International Journal for Numerical Methods in Engineering*, vol. 33, n° 7, 1992, p. 1413–1449, Wiley Online Library.
- [SIM 93] SIMO J., ARMERO F., TAYLOR R.
Improved versions of assumed enhanced strain tri-linear elements for 3D finite deformation problems. *Computer methods in applied mechanics and engineering*, vol. 110, n° 3-4, 1993, p. 359–386, Elsevier.
- [SIM 98] SIMO J., HUGHES T.
Interdisciplinary Applied Mathematics. *Mechanics and Materials, Computational Inelasticity*. Springer-Verlag, New York, , 1998.
- [SOU 95] DE SOUZA NETO E., PERIĆ D., HUANG G., OWEN D.
Remarks on the stability of enhanced strain elements in finite elasticity and elastoplasticity. *International Journal for Numerical Methods in Biomedical Engineering*, vol. 11, n° 11, 1995, p. 951–961, Wiley Online Library.

- [SOU 03] ALVES DE SOUSA R., NATAL JORGE R., FONTES VALENTE R., CÉSAR DE SÁ J.
A new volumetric and shear locking-free 3D enhanced strain element. *Engineering Computations*, vol. 20, n° 7, 2003, p. 896–925, MCB UP Ltd.
- [SOU 05] ALVES DE SOUSA R. J., CARDOSO R. P., FONTES VALENTE R. A., YOON J.-W., GRÁCIO J. J., NATAL JORGE R. M.
A new one-point quadrature enhanced assumed strain (EAS) solid-shell element with multiple integration points along thickness: Part I—geometrically linear applications. *International journal for numerical methods in engineering*, vol. 62, n° 7, 2005, p. 952–977, Wiley Online Library.
- [SOU 06] ALVES DE SOUSA R. J., CARDOSO R. P., FONTES VALENTE R. A., YOON J.-W., GRÁCIO J. J., NATAL JORGE R. M.
A new one-point quadrature enhanced assumed strain (EAS) solid-shell element with multiple integration points along thickness—part II: nonlinear applications. *International Journal for Numerical Methods in Engineering*, vol. 67, n° 2, 2006, p. 160–188, Wiley Online Library.
- [STO 95] STOLARSKI H., BELYTSCHKO T., LEE S. H.
A review of shell finite elements and corotational theories. *Computational mechanics advances*, vol. 2, n° 2, 1995, p. 125–212.
- [SUS 13] SUSSMAN T., BATHE K.-J.
3D-shell elements for structures in large strains. *Computers & Structures*, vol. 122, 2013, p. 2–12, Elsevier.
- [SZE 00] SZE K., YAO L.
A hybrid stress ANS solid-shell element and its generalization for smart structure modelling. Part I—solid-shell element formulation. *International Journal for Numerical Methods in Engineering*, vol. 48, n° 4, 2000, p. 545–564, Wiley Online Library.
- [SZE 04] SZE K., LIU X., LO S.
Popular benchmark problems for geometric nonlinear analysis of shells. *Finite elements in analysis and design*, vol. 40, n° 11, 2004, p. 1551–1569, Elsevier.
- [TAY 76] TAYLOR R. L., BERESFORD P. J., WILSON E. L.
A non-conforming element for stress analysis. *International Journal for numerical methods in Engineering*, vol. 10, n° 6, 1976, p. 1211–1219, Wiley Online Library.
- [TIM 47] TIMOSHENKO S. P.
Théorie de la stabilité élastique. , 1947, Librairie Polytechnique.
- [VOL 93] VOLDOIRE F.
An elastic-plastic enriched shell model; formulation and numerical assessment. rapport, 1993, Collection de notes internes de la Direction des Études et Recherches,SSN 1161-059X, 1161-0611.

- [VUQ 03] VU-QUOC L., TAN X.
Optimal solid shells for non-linear analyses of multilayer composites. I. Statics. *Computer methods in applied mechanics and engineering*, vol. 192, n^o 9-10, 2003, p. 975–1016, Elsevier.
- [WAG 05] WAGNER W., GRUTTMANN F.
A robust non-linear mixed hybrid quadrilateral shell element. *International Journal for Numerical Methods in Engineering*, vol. 64, n^o 5, 2005, p. 635–666, Wiley Online Library.
- [WEM 82] WEMPNER G., TALASLIDIS D., HWANG C.-M.
A simple and efficient approximation of shells via finite quadrilateral elements. *Journal of Applied Mechanics*, vol. 49, n^o 1, 1982, p. 115–120, American Society of Mechanical Engineers.
- [WIL 73] WILSON E., TAYLOR R., DOHERTY W., GHABOUSSI J.
Incompatible displacement models. *Numerical and computer methods in structural mechanics*, p. 43–57 Elsevier, 1973.
- [WRI 96a] WRIGGERS P., EBERLEIN R., REESE S.
A comparison of three-dimensional continuum and shell elements for finite plasticity. *International Journal of Solids and Structures*, vol. 33, n^o 20-22, 1996, p. 3309–3326, Elsevier.
- [WRI 96b] WRIGGERS P., REESE S.
A note on enhanced strain methods for large deformations. *Computer Methods in Applied Mechanics and Engineering*, vol. 135, n^o 3-4, 1996, p. 201–209, Elsevier.
- [WRI 96c] WRIGGERS P., KORELC J.
On enhanced strain methods for small and finite deformations of solids. *Computational Mechanics*, vol. 18, n^o 6, 1996, p. 413–428, Springer.
- [XIO 18] XIONG H., MALDONADO E. G., HAMILA N., BOISSE P.
A prismatic solid-shell finite element based on a DKT approach with efficient calculation of through the thickness deformation. *Finite Elements in Analysis and Design*, vol. 151, 2018, p. 18–33, Elsevier.
- [ZIE 77] ZIENKIEWICZ O. C., TAYLOR R. L., ZIENKIEWICZ O. C., TAYLOR R. L.
The finite element method, vol. 3. McGraw-hill London, 1977.
- [ZIE 88] ZIENKIEWICZ O., LEFEBVRE D.
A robust triangular plate bending element of the Reissner–Mindlin type. *International journal for numerical methods in engineering*, vol. 26, n^o 5, 1988, p. 1169–1184, Wiley Online Library.
- [ZIE 00] ZIENKIEWICZ O. C., TAYLOR R. L., TAYLOR R. L., TAYLOR R. L.
The finite element method: solid mechanics, vol. 2. Butterworth-heinemann, 2000.



FOLIO ADMINISTRATIF

THÈSE DE L'UNIVERSITÉ DE LYON OPÉRÉE AU SEIN DE L'INSA DE LYON

NOM: DIA

DATE de SOUTENANCE: 9 Juin 2020

Prénoms: MOUHAMADOU

TITRE: Éléments finis solide-coques hexaédrique et prismatique pour la modélisation non linéaire des structures minces et moyennement épaisses

NATURE: Doctorat

Numéro d'ordre: 2020-LYSE-I098

École doctorale: MEGA

Spécialité: Mécanique - Génie Mécanique

RÉSUMÉ:

Les structures à faibles ou moyennes épaisseurs sont naturellement présentes dans la plupart des installations de production d'énergie : bâtiment réacteur, tuyauteries sous pression, réservoirs métalliques ou bâches, cuve de réacteur, liners métalliques des enceintes de confinement pour ne citer que ceux-là. Un besoin actuellement exprimé par les unités d'ingénierie d'EDF est la modélisation des phénomènes de cloquage de liners métalliques des bâtiments réacteur. Leur modélisation nécessite la prise en compte d'un phénomène de contact-frottement engendrant du pincement sur la coque, de la plasticité sous l'effet de cloquage et de la non linéarité géométrique. Pour modéliser le comportement thermomécanique d'une structure pareille, les éléments finis de plaques et coques actuellement disponible ne semblent pas être à la hauteur. Le premier verrou attribuable à ces éléments est l'hypothèse des contraintes planes qui empêche la prise en compte de certaines lois de comportement nativement tridimensionnelles. En deuxième lieu, du fait de leur formulation avec des degrés de liberté de rotations ces éléments n'offrent pas une facilité d'utilisation lorsqu'il s'agit de résoudre des problèmes prenant en compte les effets non-linéaires telles que les grande transformations géométriques, le contact-frottement, le flambement et les pressions suiveuses. Une alternative serait d'utiliser des éléments volumiques standards. Cependant le coût de calcul prohibitif des ces derniers est difficilement accessible pour de nombreuses applications industrielles. Le but de ces travaux est de proposer une solution à cette problématique. Nous avons proposé une formulation élément fini de type solide-coque enrichie en pincement et capable de reproduire les comportements des structures minces avec une précision satisfaisante. Ce nouvel éléments fini fonctionnent avec tout type de loi de comportement tridimensionnelle sans restriction sur les champs de contraintes.

MOTS-CLÉS: Éléments fini solide-coque, Loi de comportement 3D, Champs de déformation enrichis, Intégration Réduite, , Champ de déformation assumé, Contrainte normale enrichie

Laboratoire(s) de recherche: Laboratoire de Mécanique des Contacts et des Solides
UMR CNRS 5259 - INSA de Lyon
Bâtiment Sophie Germain, Avenue Jean Capelle O,
69100 Villeurbanne Cedex FRANCE

Directeur de thèse: Anthony GRAVOUIL

Président du jury:

Composition du jury: Emmanuelle ABISSET - CHAVANNE Abed MERAÏM-FARID
Karam SAB Hakim NACEUR
Bing TIE Anthony GRAVOUIL
Nahiène HAMILA Mickael ABBAS

